



OldReaxFF Manual

Amsterdam Modeling Suite 2022.1

www.scm.com

Feb 24, 2022

CONTENTS

1	General	1
1.1	What's new in ReaxFF 2019	1
2	Ensembles	3
2.1	Nose-Hoover chains NVT ensemble	3
2.2	AH and PRH NPT ensemble with NHC	4
2.3	Berendsen NVT with separate damping for translational, rotational, and vibrational degrees of freedom	4
3	eReaxff: classical treatment of the explicit electron	5
4	Properties and Analysis	7
4.1	Vibrational Frequencies	7
4.2	Per-atoms Stress Tensor	7
4.3	NEMD Methods for Thermal Conductivity	7
4.4	Temperature Profile along Coordinate Axis	8
4.5	Local Atomic Temperature	9
4.6	Adsorption Analysis	9
4.7	On-the-fly Analysis of Reaction Events	10
4.8	Trajectory Analysis	11
4.8.1	ChemTraYzer	11
	General information	11
	Command line execution	11
	Results	12
	Very large trajectories	12
4.8.2	TRAVIS: A Trajectory Analyzer and Visualizer	13
5	FF Optimizer: framework for force field parameter optimization	15
5.1	Input files common for all optimizers	15
5.2	Control parameters common for all optimizers	16
5.3	MCFF Optimizer: Monte Carlo force field parameter optimizer	17
5.3.1	Input files	17
5.3.2	Control parameters	17
5.3.3	Results	18
5.3.4	Run-time control	19
5.4	CMA-ES Force-field Optimizer	19
5.4.1	Input files	19
5.4.2	Control parameters	20
5.4.3	Results	20
6	Accelerating Molecule Dynamics and Other Related Methods	23
6.1	Grand Canonical Monte Carlo (GCMC)	23

6.1.1	General info	23
6.1.2	Input	24
6.1.3	Output	27
6.1.4	Code Details	28
6.2	Force-bias Monte Carlo (fbMC)	29
6.3	CVHD: Collective Variable-Driven Hyperdynamics	30
6.4	The Molecule Gun	31
6.5	Bond Boost Method	32
7	Other new features and fixes	35
7.1	Repeating the unit cells	35
7.2	Molecular charges	35
7.3	External electric fields	35
7.4	Elastic wall restraint	36
7.5	ACKS2: Atom-condensed Kohn-Sham DFT approximated to second order	37
7.6	Correction for small bond orders	37
7.7	Correction for torsion angles asymptotics	37
7.8	LG dispersion	37
7.9	The rxffutil utility	38
7.9.1	reformat	38
7.9.2	minmax	38
7.9.3	pdist	39
7.9.4	params	39
8	Troubleshooting	41
8.1	Geometry optimization	41
9	ReaxFF as an AMS engine	43
10	Technical Topics	45
10.1	Hybrid parallelism	45
11	Warnings	47
12	Included Forcefields	49
13	Included Forcefields (development)	63
14	Reparametrizing ReaxFF with the CMA-ES optimizer	77
14.1	Overview of the workflow	77
14.1.1	Assigning weights	79
14.2	How to run the optimizer	80
14.3	How to monitor a running optimization	81
14.4	How to change optimizer settings	81
14.5	How to cross-validate a fitted force field	82
14.6	Running the optimizer	83
14.6.1	Errors and Cross-validation	83
14.6.2	Refine the training set	85
15	Appendices	87
15.1	The Force Field File	87
15.1.1	Format	87
15.1.2	Equation Reference	88
15.2	trainset.in	91
15.2.1	Description of the trainset.in file	91

15.2.2	Sections format	92
15.2.3	Format description	92
16	Required Citations	95
16.1	General References	95
16.2	Feature References	95
16.3	External programs and Libraries	96

1.1 What's new in ReaxFF 2019

ReaxFF is available in the [AMS driver](#) as one of the engines, which means that all tasks that can be performed with the AMS driver are also available for ReaxFF. However, not all tasks available in the classic ReaxFF standalone program have already been ported to AMS, so in the 2019 release of the Amsterdam Modeling Suite, the standalone ReaxFF program and the ReaxFF engine coexist. This manual mostly describes the standalone ReaxFF program. For documentation on how to use ReaxFF in AMS, see the special page on [ReaxFF as an AMS engine](#) (page 43) as well as the [AMS driver manual](#).

Development of the *standalone* ReaxFF program has been stopped, so there are no new features in the 2019 version.

ENSEMBLES

Currently the following ensembles are implemented in the ReaxFF program:

- NVE
- NVT with a Berendsen thermostat (see the original reaxff pdf manual for details)
- NVT with a Nose-Hoover chain (NHC) thermostat
- NPT with a Berendsen baro- and thermostat (see the original reaxff pdf manual for details)
- Anderson-Hoover (AH) or Parrinello-Rahman-Hoover (PRH) NPT with NHC thermostats

The ensemble is selected based on the `imdmet` value in the control file. Currently supported `imdmet` values are:

- 1 - Berendsen NVT
- 2 - Nose-Hoover chains NVT
- 3 - Microcanonical ensemble (NVE)
- 4 - Berendsen NPT
- 9 - Isotropic NPT with MTTK NHCP
- 10 - Anisotropic NPT with only the length of the unit cell vectors changing and MTTK NHCP
- 11 - Anisotropic NPT with full cell fluctuations and MTTK NHCP

The following `imdmet` values are also recognized by the program but the corresponding functionality has not been tested by SCM and is thus not supported (they come directly from the Adri van Duin's code):

- 5 - NVE with a switch to NVT when the average system's temperature exceeds a certain target
- 6 - Hugonostat
- 7 - NVE with shear force
- 8 - non-equilibrium NVE with the *ab* plane moving every step by a fixed amount

The selected ensemble can further be modified using other control keys: `itdmet`, `inpt` (for anisotropic NPT), `mdtemp` (target temperature, K), `tdamp1` (T damping constant, fs), `mdpres` (target pressure, MPa), `pdamp1` (P damping constant, fs), and some other. Furthermore, the NVT ensembles can further be modified using the so-called temperature regimes (T-regimes) specified via a `tregime.in` file.

2.1 Nose-Hoover chains NVT ensemble

Both the NHC-NVT and the AH-NPT are implemented following the paper by G.J. Martyna et al. [G J Martyna, M E Tuckerman, D J Tobias, M L Klein, Mol. Phys. 87 (1996) 1117]. The following parameters are used: `N_c=5`, `N_ys=5`, `Nchains=10`. The NHC-NVT ensemble is invoked by setting `imdmet=2` in the control file. The `tdamp` control

parameter (τ_t) is used to determine the first thermostat mass as $Q = N_{free}kT_{set}\tau_t^2$. It should be noted that the τ_t in the Nose-Hoover method corresponds to the period of the characteristic oscillations of the system, which is different from the τ in the Berendsen thermostat where it determines the relaxation time. Relaxation of the system with a Nose-Hoover thermostat usually takes longer than with the Berendsen one with the same τ .

The temperature regimes and zones specified via the `tregime.in` file can also be used with the NHC thermostat the same way as with the Berendsen one.

2.2 AH and PRH NPT ensemble with NHC

The Anderson-Hoover NPT ensemble is invoked by setting `imdm=9` in the control file. Everything written above about the NHC thermostat applies also to the thermostat part of the AH-NPT. The particle thermostat and the barostat are controlled by two different chains. The masses in the barostat thermostat chain are determined by the `tdamp` parameter as $Q_p = kT_{set}\tau_t^2$. The `pdamp` control parameter τ_p defines the barostat mass as $W = (N_{free} + 3)kT_{set}\tau_p^2$. Similar to the NHC thermostat, the τ_p here corresponds to the period of the characteristic pressure fluctuations of the system.

For an anisotropic (Parrinello-Rahman-Hoover) NPT dynamics one should specify `imdm=10` (fixed cell angles) or `imdm=11` (full cell fluctuations). It should be noted that the anisotropic NPT dynamics has not been well tested yet and thus it should be considered experimental. For `imdm=10` one can additionally freeze one or more axes using the `inpt` control parameter. The supported values are identical to those used for the Berendsen NPT: 1, 2, 3 to keep a, b, or c fixed, respectively, and 4 to keep both a and b fixed.

2.3 Berendsen NVT with separate damping for translational, rotational, and vibrational degrees of freedom

It is possible to equilibrate translational, rotational, and vibrational degrees of freedom separately from each other. This may be necessary, for example, in the case where the initial state of the system contains many quickly spinning molecules with a relatively small translational velocity. The usual equilibration of such a system would take a lot of time due to the low translational velocity of the molecules. This mode is invoked by setting `imdm=1` and `itdm=7` in the control file.

EREAXFF: CLASSICAL TREATMENT OF THE EXPLICIT ELECTRON

The eReaxFF implementation in the SCM ReaxFF program is based on the paper [J. Chem.Theory Comput 12, 3463 (2016) (<https://doi.org/10.1021/acs.jctc.6b00432>)] and on the prototype code obtained from the group of Adri van Duin. It was, however, largely rewritten by SCM to meet our coding standards and to maximize performance.

An eReaxFF calculation requires the following additional force-field parameters to be defined (equation numbers are referring to the eReaxFF paper):

- The header of the force-field file must begin with the “[ereaxff acks2]” string,
- General parameter 27: the currently reserved p_{elho} parameter in the unpublished electron-hole interaction equation (which is still highly experimental and is subject to change),
- General parameter 37: the Gauss exponent parameter p_{val} describing the free electron, Eq(3),
- General parameter 41: a different taper radius for electron and hole interactions can be specified here,
- Atomic parameter 24: the α parameter in Eq(2),
- Atomic parameter 27: the β parameter in Eq(2),
- Atomic parameter 13: the p_{ij}^{xel2} parameter in Eq(4d),
- Bond parameter 16: the p_{ij}^{xel1} parameter in Eq(4d).

Additionally, since the eReaxFF method is usually coupled to the ACKS2 charge equilibration method, the ACKS2-related parameters should also be defined.

In the input geometry file, the explicit electrons are called ‘El’ and the holes are ‘Ho’ (yes we are aware that the latter clashes with holmium but this is what the original authors are using).

It should be noted that the eReaxFF implementation is considered experimental so the users are advised to contact Adri van Duin regarding its use.

PROPERTIES AND ANALYSIS

4.1 Vibrational Frequencies

See [AMS documentation](#)

4.2 Per-atoms Stress Tensor

The per-atom stress tensor is calculated according to [Thompson, Plimpton, Mattson, J Chem Phys, 131, 154107 \(2009\)](#) (<https://doi.org/10.1063/1.3245303>) . when the `lstres` control parameter is set to a non-zero value.

The calculated stress tensor is stored in the `rxkf` file in the “History%Atomic stress xxxx” variable, where `xxxx` corresponds to a step number, in units of MPa or MPa * A3, depending on the `lstres` value in the ReaxFF control file:

- 1 `lstres` (GUI: *Stress energy*) the stress times volume value per atom $S_{\alpha\beta}$ is calculated as a sum of the per-atom virial and the kinetic energy term: $S_{\alpha\beta} = mv_{\alpha\nu}\nu_{\beta} + F_{\alpha}\nu_{\beta}$.
- 2 `lstres` (GUI: *Stress*) the result is equal to $S_{\alpha\beta}/V$, where V is atomic volume calculated using the Voronoi partitioning scheme.

The temporal average atomic stress values are stored in “History%Average atomic stress xxxx”. The tensors are saved as follows: six components (`xx`, `yy`, `zz`, `yx`, `zx`, `zy`) for the 1st atom followed those for the 2nd one, and so on.

See the ReaxFF [stress strain tutorial](#) for an application of the per-atom stress tensor.

4.3 NEMD Methods for Thermal Conductivity

There are different methods to study thermal conductivity using non-equilibrium molecular dynamics (NEMD). In ReaxFF, two methods are implemented. A common feature of these methods is that they require the system to be divided into three or more zones, each with its own thermostat and/or other properties. Both methods make use of the `tregime.in` file to define the zones and their properties. The first method maintains the temperature of the heat source and the heat sink zones at the given temperature using two different thermostats and measures the amount of heat transferred. This method is invoked by setting the `itdmet` control parameter to 2, which is the default value. The format of the `tregime.in` file for this method is as follows:

```
# A hash sign starts comment
# First equilibrate the system at 400K
#start_step nzones zone1: start end Tset damp dT
0 1 1 7840 400 100 0.0
# Switch to NEMD starting from step 10000
```

(continues on next page)

(continued from previous page)

```
#start_step nzones zone1: start end Tset damp dT | zone2: start end Tset damp dT |
↪zone3: start end Tset damp dT
  10000      3          1 7520  0  0 0.0          7521 7680  500 100 0.0
↪          7681 7840  300 100 0.0
```

In this example, atoms 1-7520 form the heat conduction zone, zone1. This zone is not thermostatted (the NVE ensemble is used), which is specified by a zero value for the temperature damping parameter damp. The zone2 (atoms 7521-7680) is a heat source and its temperature, 500K in this case, is maintained by the thermostat specified by the *imdm* control parameter (1 for Berendsen and 2 for NHC). The zone3 (atoms 7681-7840) is the heat sink whose temperature is maintained at 300K. In the steady state, a temperature gradient will evolve through the conduction zone and the amount of heat transferred through it per unit of time should be equal to the amount of heat added to the heat source zone, and to the heat removed from the heat sink zone. The total heat added or removed for each zone by the corresponding thermostat is written to the fort.75 file.

The second method (which is available in two variants) is invoked by setting *itdmet* to 8 or 9. There is no thermostat as such involved here, which means that the *imdm* control parameter must be set to 3 (NVE ensemble). Instead of controlling the zone's kinetic energy based on the temperature, the heat source zone is added (and the heat sink is subtracted) a fixed amount of kinetic energy each step. Currently there are two variants of this method implemented: a simple heat pump (*itdmet* = 8), where atom velocities are simply scaled when adding or removing the kinetic energy, and the HEX method by Ikeshoji and Hafskjold in which the velocities are scaled in such a way as to keep the velocity of the center of mass of the zone constant. It should be noted that the HEX introduces a small energy drift. The format of the *tregime.in* file for these two methods is almost the same as for the first method with some differences: the Tset and damp parameters are ignored and the dT value becomes dQ (in kcal/mol per MD step):

```
# NEMD heat flow parameters starting from the very first step.
# In practice this should be restarted from an equilibrated system
# nzones conduction zone:start end | heat source: start end dQ | heat
↪sink: start end dQ
0 3          1 7520 0 0 0.0          7521 7680 0 0 0.01
↪          7681 7840 0 0 -0.01
```

The zone assignment in this example is the same as in the previous one. One should be careful when choosing values for dQ because a too large positive value may lead to pyrolysis of the heat source and a too large negative value may lead to abnormal termination when all kinetic energy of the heat sink has been drained.

For all these methods, in order to calculate the thermal conductivity value one needs to divide the heat flow rate by the temperature gradient and by the flow cross-section area: $k = W / (S \cdot \text{grad}(T))$. In the first method the heat flow rate can be calculated from the thermostat energy values found in the fort.75 file (or rather their derivative w.r.t. the time). In the heat pump and the HEX methods the heat flow rate is defined by the dQ value and by the time step parameter directly: $W[\text{Watt}] = 6.94122 \cdot 10^{-6} dQ/dt$, where dQ is given in kcal/mol/step and dt is the time-step in fs. See below on how to determine the temperature gradient along an axis.

4.4 Temperature Profile along Coordinate Axis

The temperature profile along the simulation box axes will be calculated when *ianaly=2* is set in the control file. In the profile.out file found in the results directory, the average temperatures along each box axis are listed in the following format:

```
Temperature profiles along unit cell coordinate axes, averaged over time per axis. MD
↪step 1000
Cell coord| N(sample) T(ave) | N(sample) T(ave) | N(sample) T(ave) |
          | a          | b          | c          |
```

(continues on next page)

(continued from previous page)

0.0050	22809	326.57722	14475	237.18337	24784	260.64395
...						

Here, “Cell coord” is the coordinate of the middle point of the corresponding length interval (the bin), N(sample) is the number of samples used in the averaging (which is done over the time and the atoms in the bin), and T(ave) is the average temperature in the bin over the time interval. The time interval and the number of bins are determined by the iout7 and tprof control parameters, respectively. It should be noted that the binning of atoms is done in fractional coordinates, which means that the plane separating two bins, for example, along the *c* crystal axis is always parallel to the (**ab**) plane and may thus not be orthogonal to the *c* axis itself.

The profile is also saved to the KF result file. The corresponding variables in the History section are “T profile X YYYY”, when “X” specifies the profile axis and is one of “a”, “b”, or “c” and “YYYY” is the MD step number.

4.5 Local Atomic Temperature

ReaxFF can calculate atomic temperatures and spatial averages thereof. Individual atomic temperatures are saved with the trajectory by default and these can be visualized by AMSmovie. However temperatures of individual atoms can vary significantly, which may result in a not so clear picture. To see trends in the spacial distribution of the temperature across the system it may be useful to display an average temperature of the atom and its neighbors instead. The spacial averaging is activated by specifying the following keywords in the control file:

1	localt	Request local temperature averaging between atoms
10.0	localr	Set the awareness radius to 10.0 angstrom (Default 5.0)

See also: [How to visualize local atomic temperatures in the GUI \(https://www.scm.com/news/visualization-local-temperatures-atomic-properties-amsmovie/\)](https://www.scm.com/news/visualization-local-temperatures-atomic-properties-amsmovie/).

4.6 Adsorption Analysis

Analysis of the molecular composition of the system under study normally includes all atoms. In the original ReaxFF, it was already possible to ignore a certain type of atoms when performing the analysis, which may help analyze surface dissociation reactions. For example, in an Al-H₂O system, one can distinguish a physisorbed water molecule from a surface-dissociated one by excluding all aluminum atoms from the molecular analysis. However, this feature cannot be used for surfaces consisting of atoms of more than one type and it will also not distinguish a surface reaction from the same reaction taking place off the surface.

The new type of analysis is intended to fill this gap. This analysis is triggered by specifying a range of atoms as a “support” in the input geometry (in the BGF format):

```
SUPPORT startAtom endAtom
```

Here, startAtom and endAtom are indices of the first and the last atoms of the range constituting the support (surface+bulk) part of the model system. Multiple SUPPORT keys are allowed in the input file, in which case they will all be merged into one “support” entity. If this key is present, the molfra.out file will contain something like this (taking again the Al-H₂O system as an example and treating the Al slab as a support):

Iteration	Freq.	Molecular formula	Molecular mass
0	120 x	Al	26.9820
0	61 x	H2O	18.0150
0	14 x	H2O(ads)	18.0150

The “H2O(ads)” must be understood as a water molecule adsorbed on a surface via its oxygen atom. During dynamics one may also see molecules such as HH(ads)O(ads), where both H and O atoms interact with the support. In other words, the “(ads)” suffix applies to the atom preceding it and, through this atom, to the whole molecule. As usual in this kind of analysis in ReaxFF, two atoms are considered connected if the order of the bond between them is greater than the `cutof3` control parameter.

4.7 On-the-fly Analysis of Reaction Events

Built-in reaction detection and analysis in `reaxff` is enabled by setting the `ibocha` control parameter to a non-zero value. This method is based on keeping track of the number of bonds of atoms in the system at set intervals, which guarantees a very low overhead in almost all cases. When the number of bonds around an atom changes this atom becomes an atom of interest and the output is written to `bondchange.out` and possibly `reactions.xyz`. The molecular composition analysis is performed every `iout7` steps and results are both saved as part of the trajectory and used for the analysis described here. For this reason the `kf-file` sampling interval parameter `iout7` also defines the frequency of the on-the-fly reaction detection.

Two `ibocha` values are currently supported: 1 and 2. The `ibocha=1` does the same as in the original `reaxff` version: printing to `bondchange.out` a one-line message for each atom with changed number of bonds, like this:

```
MD-step:      11780 Atom N          15 has lost bond(s): 3->2, moved from C5H8O12N4 to
↔O2N
```

The `ibocha=2` option is an extension to `ibocha=1` and it tells `reaxff` to also dump the `xyz` coordinates of molecules mentioned above to the `reactions.xyz` file. For the example above, two molecules would be written:

```
29
Step 11780 reactant C5H8O12N4
 1  N          2.69656838          5.12589261          0.97180852
 2  O          4.15730086          4.99047544          1.27822949
 3  O          2.22159806          4.44324905          0.03576820
 4  O          1.96584976          5.93167910          1.72665822
 5  H          3.95163594          6.25753755          3.15184024
 6  H          5.44777765          6.49477803          2.46494613
 7  C          6.41416826          3.90121980          2.65835254
 8  N          8.24612020          4.30823059          1.11141846
 9  O          7.66154742          4.55735751          2.58464023
10  O          8.79263162          5.29595371          0.49362056
11  O          8.80963036          3.13719696          1.11070541
12  H          6.40826369          3.18493699          3.51002413
13  H          5.74405149          3.41353259          1.73612946
14  C          4.42001493          3.68879837          4.30184825
15  N          4.37449815          1.82746861          6.38489303
16  O          5.14915771          2.89762593          5.05416466
17  O          4.99604438          1.65830747          0.48544462
18  O          3.20940517          1.45602335          6.22444281
19  H          3.86848640          2.95960033          3.60488579
20  H          3.39984332          4.11318335          4.66083432
21  C          5.87896652          5.70360041          4.73680165
22  N          5.55108399          7.52510417          6.10594010
23  O          4.91172061          6.76799249          4.93853789
24  O          4.75311340          7.58235081          0.10430559
25  O          6.80961357          7.76560552          6.07934918
26  H          6.91649734          6.15363498          4.61919911
27  H          6.21346007          4.91150861          5.55644182
28  C          4.60495584          5.65630445          2.50946344
```

(continues on next page)

(continued from previous page)

29	C	5.45265424	4.77567020	3.49485932
3				
Step	11780	product	O2N	
15	N	4.36602590	1.82312814	6.39657912
17	O	4.99790591	1.64883276	0.48915935
18	O	3.22220332	1.46348009	6.23717031

In the example above, the first molecule, C₅H₈O₁₂N₄ denoted as “reactant”, is the molecule that contained atom N 15 at the previous sampling step. The second molecule, NO₂ denoted as “product”, is the molecule that contains the atom at the current step. Thus, the xyz coordinates of the “reactant” molecules is always iout7 steps behind the coordinates given for the “products”.

Note: The choice of the iout7 sampling frequency is very important for a correct analysis. Too small iout7 value (frequent sampling) may lead to many forth-and-back “reactions” detected, which are merely stretched bonds due to some very hot vibrations. A too large iout7 value (rare sampling) may lead to some elementary reactions being missed and merged into “collective” events.

Note: The low-overhead bonding analysis method used here may miss events that do not result in the change of the number of bonds on an atom. For example, concerted reaction of proton transfer in a ring of water molecules will likely go unnoticed because the number of bonds remains the same on every atom, even though the bonds do change.

4.8 Trajectory Analysis

4.8.1 ChemTraYzer

General information

The ChemTraYzer scripts, developed in the [Leonhard group](http://www.ltt.rwth-aachen.de/forschung/molekulare_thermodynamik/molekulare_thermodynamik/project/ChemTraYzer_Modellierung_chemi/) (http://www.ltt.rwth-aachen.de/forschung/molekulare_thermodynamik/molekulare_thermodynamik/project/ChemTraYzer_Modellierung_chemi/) at Aachen University, provide a methodology for deducing quantitative reaction models from reactive molecular dynamics simulations by identifying, quantifying, and evaluating elementary reactions of ReaxFF trajectories.

For usage with ADF’s ReaxFF the scripts have been slightly modified and use RDKit instead of the openbabel-bindings. A GUI-tutorial is available [here](#). The original paper, Automated Discovery of Reaction Pathways, Rate Constants, and Transition States Using Reactive Molecular Dynamics Simulations, can be found [here](http://dx.doi.org/10.1021/acs.jctc.5b00201) (<http://dx.doi.org/10.1021/acs.jctc.5b00201>).

Command line execution

1. Open a terminal:

- on Linux: Ctrl+Alt+T usually works, otherwise browse your application menus
- on Windows: double click `adf_command_line.bat` in your ADF installation directory (e.g. `C:/ADF2016.106`), enter ‘sh’ to go to a shell

2. cd into the directory where your trajectories are located with the terminal

3. Assuming your trajectory file is called `traj.rxkf`, running ChemTraYzer is a two step procedure

- run the processing step (with default values)

```
$AMSBIN/amspython $AMSHOME/scripting/standalone/chemtrayzer/processing_scm.py  
traj.rxkf
```

- run the analysis step (with default values)

```
$AMSBIN/amspython $AMSHOME/scripting/standalone/chemtrayzer/analyzing_scm.py  
traj.rxkf
```

You will probably want to adjust the settings. A list of all available options is shown by calling the scripts without arguments, e.g. `$AMSBIN/amspython $AMSHOME/scripting/standalone/chemtrayzer/analyzing_scm.py`

Results

ChemTraYzer creates several output files, the most relevant ones being:

reac.spec.tab

1st line: indices of the species, e.g. S1;S2;S3;etc...

2nd line: SMILES of the species, e.g. ;C;CC;CCO;CCOO;CO;O;[H];etc...

3rd - end: timestep, counts of the species, e.g. 150;0;0;2;... // step 150, current count of S3 == 2

reac.rate.tab

1st line: indices of the reactions, e.g. t [steps];R0;R0*;R1;R1*; // a "*" marks the back reaction

2nd line: the reactions formulated with the species indices, e.g. S0 + S0 + S0 + S13 -> S65; // see reac.spec.tab

3rd line: the reactions formulated with sum formulas, e.g. CH3 + H -> CH4, etc...

4th - ...: timestep; rate constant of the reaction; rate constant of the back reaction; etc...

reac.pic/

folder containing the 2D structures (if obabel was able to create them) reac.pic/xyz/ xyz files of the species as they first appeared in the trajectory

Very large trajectories

When a long simulation has been run, the files handled and generated by ChemTraYzer can become very large. Sometimes even too large to handle. In such a situation, you can request the information required by ChemTraYzer to be written to a separate file at a separate frame rate. The advantage of this approach is that you can drastically increase the frame rate for the rxkf file which will speed up the calculation and save a lot of disc space.

To apply the two different frame rates, add a new command to the control file in the run script:

```
100 lmpsbo LAMMPS bonding table!
```

ReaxFF will now generate a file called **tab** with all required bonding information written every 100th iteration. You can adjust the number as needed. Next, decrease the frame rate for the rxkf file by altering these existing parameters in the control file:

```
5000 iout1 Output frequency to unit 71 and unit 73 5000 iout2 Save coordinates  
(xmout, moldyn.vel, Molfra) 5000 iout7 Frequency of reaxout.kf writing
```

Once the calculation has finished, you will find a file called **tab** in the **results** folder. This file has to be renamed to the same filename as the rxkf trajectory file and placed in the same location as that file. For example, if your rxkf file is called traj.rxkf:

Rename the file from **tatb** to **traj.tatb**

```
cp tatb traj.tatb
```

and place it in the same folder as the file traj.rxkf. Run the ChemTraYzer processing and analysis as explained above. To double check if the file tatb file has been read, check for the following confirmation log message of the processing script:

```
!-----!
! RXKF: Using Existing Bonding Table: traj.tatb      !
!-----!
```

4.8.2 TRAVIS: A Trajectory Analyzer and Visualizer

TRAVIS is a command line C++ program for analyzing and visualizing trajectories from all kinds of Molecular Dynamics or Monte Carlo simulations developed by M. Brehm (*J. Chem. Inf. Model.* 51(8) (2011) (<https://doi.org/10.1021/ci200217w>)). TRAVIS is shipped with the Amsterdam Modeling Suite from on ADF2017.

The supported input formats of TRAVIS are:

- *xyz*
- *pdb*
- *lmp* (Lammps)
- *HISTORY* (DLPOLY)
- *prmtop/mdcrd* (Amber)

With ADF2017 the analysis of DFTB and ReaxFF trajectories is supported.

Note: TRAVIS was designed with non-reactive trajectories in mind, thus one needs to make sure that:

- *either* the composition of the target species does not change, e.g. by analyzing only a suitable time-interval or running non-reactive dynamics
 - *or* the analysis does not depend on the detection of a particular species, e.g. the diffusion of (inert) atoms
-

In order to run a TRAVIS analysis on your trajectory open a shell and navigate to the folder you ran your calculation in. Use the python script **kf_to_xyz.py** to convert your trajectory into a TRAVIS-compatible XYZ-format.

Assuming the trajectory is called **traj.rxkf**, run:

```
$AMSBIN/amspython $AMSHOME/scripting/trajectory_analysis/kf_to_xyz.py
traj.rxkf
```

This step will create a file called **traj_travis.xyz** that contains the frames in XYZ format plus an additional 5th column containing the ReaxFF atomic charges.

Next, start TRAVIS to enter an interactive dialog:

```
$AMSBIN/travis.exe -p traj_travis.xyz
```

An example of how to use TRAVIS with ReaxFF can be found in the [advanced Li-Batteries ReaxFF tutorial](https://www.scm.com/news/tutorial-li-ion-diffusion-coefficients-reaxff/) (<https://www.scm.com/news/tutorial-li-ion-diffusion-coefficients-reaxff/>).

FF OPTIMIZER: FRAMEWORK FOR FORCE FIELD PARAMETER OPTIMIZATION

The FF Optimizer is a base module that can be extended with other modules that implement specific optimization methods. The base FFOptimizer module offers methods to compute the cost function (also known as the error or the fitness function) and its first and second derivatives with respect to the force-field parameter values. Besides that, there are two extensions available that implement global optimization methods: Monte-Carlo (MCFE) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

5.1 Input files common for all optimizers

In order to optimize a ReaxFF forcefield, the files listed below must be present in the directory where the reaxff program is executed.

- *iopt* - file containing a line of text with a single number on it. To use the basic FFOptimizer features this file must contain 6. See the *ffoact* description below for more details.
- *ffield* (page 87)- the initial force-field file.
- *control* - in addition to general ReaxFF control parameters it may also contain the FFOptimizer-related ones explained in the corresponding sections.
- *trainset.in* (page 91) - file with test values from the training set, the same as in the original reaxff force-field optimization, see page 27 of the ReaxFF User Manual.
- *params* - file that describes variable forcefield parameters, one line per parameter (see example below).

This format is a generalization of the original params file format.

The hash symbol '#' starts a comment. Each non-comment line must begin with three integer numbers, the section-block-item coordinates of the corresponding parameter. The first coordinate specifies a section: 1 - general parameters, 2 - atomic parameters, 3 - bonds, 4 - off-diagonal terms, 5 - valence angles, 6 - torsion angles, 7 - hydrogen bonds. The second coordinate specifies a block within the given section or an item index for the general parameters section (section=1). The third coordinate specifies an item index within the block, except for section 1 for which the third coordinate is ignored. The coordinates can optionally be followed by a one or more real number. The number and the meaning of the reals depends on the selected task. For some force-field optimization methods the first three real number have a special meaning referred to as *delta* (or *sigma*), *ffmin*, and *ffmax* in the following sections, respectively. A params file can be generated using the *rxffutil utility* (page 38).

An example of the params file:

```
# i j k  x1  x2  x3  ...
1 1 0  0.1 -1.0  1.0  # The 1st general parameter
```

(continues on next page)

(continued from previous page)

```
2 3 4 0.1 -1.0 1.0 # The 4th parameter of the 3rd atoms block
...
```

- *ffield_bool* - an alternative way to specify variable parameters if the params file is not present. This file has the same format as *ffield* but instead of parameter values it contains 1.0 or 0.0 as a flag whether the corresponding value is to be variable or not, respectively.
- *ffield_min* - an alternative way to specify minimum values (*ffmin*) for the variable parameters if the *params* file is not present. This file has the same format as *ffield*.
- *ffield_max* - an alternative way to specify maximum values (*ffmax*) for the variable parameters if the *params* file is not present. This file has the same format as *ffield*.
- *geo* - file with the training set geometries in the BGF format, the same as in the original reaxff force-field optimization. Geometries of different molecules in this file must be concatenated.
- *models.in* - as an alternative to a single *geo* file one can save each geometry in its own file and list the files that are part of the training set here. The molecule name specified in the *models.in* file must match that in the *trainset.in* file and it takes precedence over any molecule name specified inside the geometry file (filename1, etc.). The format of the file is as follows:

```
filename1 MoleculeName1
filename2 MoleculeName2
...
```

5.2 Control parameters common for all optimizers

The following control parameters are related to the FFOptimizer. The default value for each parameter is given in parentheses.

- ***ffoact* - the task ID to be performed by the base FFOptimizer module (i.e. when the *iopt* file contains 6):**
 - 1 - calculate gradient of the error function with respect to the variable FF parameters by finite differences. This option requires $2*N + 1$ error function evaluations, where N is the number of variable force-field parameters.
 - 2 - calculate the second derivatives (Hessian) matrix of the error function with respect to the variable FF parameters by finite differences. The eigenvalues and eigenvectors of the obtained Hessian matrix will also be computed. In this case, $N^2 + N + 1$ function evaluations will be done. Note: a calculation of derivatives can be very slow so make sure you run it on as many processors as possible.
 - 3 - calculate an error function value for each column-vector of parameters specified in the *params* file. If the params file does not contain any column-vectors (i.e. there are no real numbers at all in the file) then the error function for the current *ffield* file is calculated. This feature can be used, for example, by external force-field optimizers. The result per *trainset.in* entry is written to the *fort.99* file.
 - 4 - find *replic* random vectors of parameter values that result in a valid (non-NaN) error function value. The random values are distributed uniformly in the allowed parameter space.
- *ffostp* - the delta used for calculating first and second derivatives by finite differences (0.01).

The following options may apply to any FF optimization type.

- *ffdedi* - If running in parallel and *ffdedi* is set to 1 then the master process will run as a dedicated dispatcher and will not perform any computations. The default value depends on the number shared-memory nodes used in the calculation (0 for single-node calculation, 1 otherwise).

- *replic* - number of parameter sets to calculate at once for *ffoact* = 4 (1 by default). This control parameter may have different meaning and defaults for other force-field optimizers.
- *fort99* - if this parameters is set to 0 then the fort.99 files will not be written. By default, every fitness function evaluation produces a fort.99.xxx.yyy file, where xxx is an iteration number and yyy is a replica index at the iteration. This may potentially lead to a large number of fort.99.* files so setting fort99 to 0 can be useful for saving space in long production runs.
- *fort90* - MCFFOptimizer will create a geo file with optimized geometries corresponding to the current best parameter set.

5.3 MCFF Optimizer: Monte Carlo force field parameter optimizer

The MCFF Optimizer uses a Monte Carlo approach to finding the best fit force field for the given training set. It is based on the paper by E. Iype et. al. (<https://doi.org/10.1002/jcc.23246>) In the following sections, the input and output of the MCFF Optimizer are described.

A worked out example on building a training set and optimizing a ReaxFF force field with MCFF is on our [workshop pages](https://www.scm.com/adf-modeling-suite/adf-hands-on-workshops/advanced-2-day-reaxff-workshop/) (<https://www.scm.com/adf-modeling-suite/adf-hands-on-workshops/advanced-2-day-reaxff-workshop/>).

5.3.1 Input files

In addition to files specified in the FF Optimizer section the following files must be present in the directory where the reaxff program is executed.

- *iopt* - file containing a single text line with 4 on it. This will instruct reaxff to perform a Monte-Carlo force-field optimization.

The *mccfopt_water* example in the *examples/reaxff* directory demonstrates the use of the MCFFOptimizer. Note however that this example is not physically meaningful. For example, many atomic force-field parameters are allowed to vary in a very broad range, which is the same for all elements. In practice, you will want to set the range for each element separately.

5.3.2 Control parameters

The following control parameters are related to MCFFOptimizer. The default value for each parameter is given in parentheses.

- *mccfit* - number of MC iterations (10000). Since the Monte-Carlo method does not have any notion of convergence the optimization is stopped after *mccfit* iterations.
- *mcbeta* - initial MC beta parameter in the acceptance probability calculation $P = \exp(-\beta\Delta E)$ (0.0). Here, ΔE is a difference in the error function between the current and the best step so far. If the current step is the best, it is always accepted. Otherwise, the acceptance probability is calculated using the formula above and it is calculated with a random number from the (0,1) range. The optimal value of beta depends on values of the error function. If absent or set to 0, it will be calculated after the first step as $\beta_0 = \sqrt{\frac{N_{par}}{2} \frac{1}{E_0}}$, where N_{par} and E_0 are the number of variable parameters and the initial error function value, respectively.
- *mcdbet* - simulated annealing: increase the beta parameter by this value at each step (0.0). A positive *mcdbet* value means that the probability to take a step that increases the error function will decrease over time. This has the same effect as decreasing the temperature in the classical molecular Monte-Carlo method.

- *mcbsca* - simulated annealing: divide the beta parameter by this value at each step (1.0). A value of *mcbsca* less than 1.0 has an effect similar to positive *mcdbet*. If *mcbsca* is set to 0, it will be calculated after the first step as $\frac{\beta_n}{\beta_{n+1}} = N_{step} \sqrt{\frac{5}{E_0}}$, where N_{step} and E_0 are the number of steps and the initial error function value, respectively.
- *mcacpf* - probability to vary a variable at each step (0.2). To avoid taking very large steps only some of the variables are varied at each step (selected randomly).
- *mcrxdd* - number of intervals to divide the parameter range between *ffield_min* and *ffield_max* into (100).
- *mcstep* - initial max step size in units of *range/mcrxdd*, where *range* = difference between *ffield_max* and *ffield_min* values (1.0).
- *mcmxst* - maximum allowed value of max step size (100);
- *mcscps* - factor to scale max step size to satisfy acceptance tolerance (1.1);
- *mctart* - target acceptance rate, percent (30.0);
- *mcmart* - max acceptance rate, percent (70.0);
- *mcmimi* - if not 0, minimize the best force-field parameter set after so many iterations (0). The optimization is performed only if the best set has changed since the previous minimization. The minimization employed here is gradient-free and relatively slow so it should not be used too frequently.
- *replic* - number of replicas to try at each step (1). At each step, *replic* Monte-Carlo steps are done and the one with the lowest error is selected for the next iteration.

Relation between *mcrxdd*, *mcstep*, *mcmxst*, and *mcscps*. When the parameter ranges are given via *ffield_max* and *ffield_min* files, the range for each parameter is divided into *mcrxdd* intervals. The step taken is a uniformly distributed random value in the range $(-\Delta x, \Delta x)$, where $\Delta x = mcstep * (Xmax - Xmin) / mcrxdd$. Here, *Xmax* and *Xmin* are values of this parameter from *ffield_max* and *ffield_min*, respectively. When the parameter range is given via a *params* file then $\Delta x = mcstep * delta$, where *delta* corresponds the first real number in the *params* file. When performing optimization, the program keeps track of the average acceptance rate and adjusts *mcstep* up or down by the *mcscps* factor to keep the acceptance rate close to *mctart*. If the acceptance rate is too low the step size is decreased to allow searching for a smaller parameter space. The *mcstep* value can never be larger than *mcmxst*.

It should be noted that the value of the MC step size (and thus all the parameters discussed in this section) applies to all force-field parameters to the same extent, which means that it is very important to select the min and max parameter values very carefully. The rule of thumb here is that the range should be as small as possible covering only the physically meaningful values.

5.3.3 Results

Main results of the MCFFOptimizer are saved in the following files:

- *ffield_best* - force-field file corresponding to the lowest error value
- *ffield_last* - the most recent accepted force-filed
- *MCFFOptimizer.log* - summary of iterations including the error function value, number of changed and bounded force-field parameters, cumulative number of accepted and rejected steps at each step. Also the current MC parameters such as the β value and the acceptance rate, are shown, as well as the elapsed time in seconds.
- *fort.99* - the error function breakdown for the latest step
- *fort.90* - if *fort90* is set in the control file, geo file with optimized geometries corresponding to the current best parameter set

5.3.4 Run-time control

The progress of the force-field optimization can be controlled by changing parameters in the *istop* file present in the calculation directory. The file is read every 10 iterations. The parameters are explained below:

- StopKey - replace 0 with 1 to stop the calculation
- Beta - the current β value corresponding to the *mcbeta* control parameter
- Command - one of: NONE, WALK, JUMPWALK, MINIMIZE, JUMPMINI:
- NONE - no change in the procedure
- WALK - switch to Monte-Carlo steps (the default);
- MINIMIZE - switch to gradient-free minimization of the latest accepted force-field;
- JUMPWALK - take the best force-field so far and start the Monte-Carlo procedure from there;
- JUMPMINI - take the best force-field so far and minimize it.
- ScaleFactor - change the current step size corresponding to the *mcstep* control parameter
- ActiveParameterFraction - set the fraction of force-field parameters changed at each step (*mcacpf*)
- deltaBeta - change the current value of the *mcdbet* control parameter
- BetaScaling - change the current value of the *mbsca* control parameter
- ScaleParSpace - change the current value of the *mcscps* control parameter

5.4 CMA-ES Force-field Optimizer

The CMA-ES FFOptimizer uses the Covariance Matrix Adaptation Evolution Strategy to find the best fit force field for the given training set. The current implementation is using the *c-cmaes* library (<https://github.com/CMA-ES/c-cmaes>) by Nikolaus Hansen. In the following sections, the input and output of the CMA-ES FFOptimizer are described.

See also: The *advanced tutorial on ReaxFF reparametrization* (page 77) .

5.4.1 Input files

In addition to files specified in the FF Optimizer section the following files must be present in the directory where the reaxff program is executed.

- *iopt* - file containing a single text line with 7 on it. This will instruct reaxff to perform a CMA-ES force-field optimization.

The following files are optional:

- *cmaes_restart.txt* - a restart file. It should be a copy of the *cmaes_resume.txt* file from another calculation. If this file is present the optimization procedure will be restarted where it was left off.
- *params* - If this file is present, it will define the variable reaxff parameters, one line per parameters, as follows:

```
# i j k   sigma   min   max   # The 1st general parameter
  1 1 0     0.1   -1.0  1.0   # The 4th parameter of the 3rd atoms block
  2 3 4     0.1   -1.0  1.0
...

```

The first three number on a line specify the parameter coordinates (the i,j,k indices). The fourth number is the initial sigma value (the step size) for the given parameter. The fifth and sixth numbers define the range of allowed values for this parameter.

If sigma is less than or equal to zero then it is determined by dividing the range between *min* and *max* by the *mcrxdd* parameter as described below.

Whenever the CMA-ES algorithm makes a trial step outside the given interval, a very large number for the fitness function is returned without performing the calculation. This should “teach” the CMA-ES algorithm to avoid the forbidden part of the parameter space. If at some point the CMA-ES procedure leads to a mean value (the center of the distribution) outside the allowed interval then the optimization stops.

5.4.2 Control parameters

For compatibility with the MCFFOptimizer the following control parameters are also applicable to the CMA-ES Optimizer. The default value for each parameter is given in parentheses.

- *ffotol* - CMA-ES convergence parameter (1.0e-6). The optimization is stopped when the step-size becomes smaller than *ffotol* for every parameter.
- *mcffit* - max number of iterations (10000). An optimization is stopped when it has converged or after *mcffit* iterations.
- *mcrxdd* - number of steps to divide the parameter range between *ffield_min* and *ffield_max* into if the *params* file is not present (100).
- *replic* - the CMA-ES sample size. The CMA-ES algorithm is designed to use multiple fitness function values at once, called a sample. The sample size should be at least as large as the default calculated as $M = 4 + 3 * \ln(N)$, where N is the number of variables.

Relation between *mcrxdd* and the step size sigma. When the parameter ranges are given via *ffield_max* and *ffield_min* files, the initial sigma parameter for CMA-ES is calculated as $\sigma = (X_{max} - X_{min}) / mcrxdd$. Here, X_{max} and X_{min} are values of this parameter from *ffield_max* and *ffield_min*, respectively. When the parameter range is given via a *params* file then σ corresponds the first real number in the *params* file.

5.4.3 Results

Main results of the CMA-ES Force-Field Optimizer are saved in the following files:

- *ffield_best* - force-field file corresponding to the lowest error value
- *cmaes_progress.txt* - a progress file written by the *c-cmaes* library containing some useful information about the current state of the CMA-ES model.
- *fort.73* - a log file containing some general output from the *ReaxCMAESFFOptimizer* module.
- *cmaes_resume.txt* - a file containing the complete set of CMA-ES model parameters needed to restart the optimization. This file is updated only when a new best-fit force-field is found.
- *fort.99* - the error function breakdown for the latest step

Tip: See also: The [advanced tutorial on ReaxFF reparametrization](#) (page 77) .

Note: With the 2020 release of the Amsterdam Modeling Suite, we have introduced [ParAMS](#), a Python based toolkit integrated with the [AMS driver](#) and [PLAMS](#) for the optimization of parameterized engines, such as [ReaxFF](#) and [DFTB](#). [ParAMS](#) provides similar functionality to the parameter optimizers included in the standalone [ReaxFF](#) program, which

are documented in this manual. We encourage users interested in parameterization to consider using ParAMS. It is currently in an alpha state, but will be developed further in the future and will at some point become the default parameterization tool in the Amsterdam Modeling Suite.

ACCELERATING MOLECULE DYNAMICS AND OTHER RELATED METHODS

6.1 Grand Canonical Monte Carlo (GCMC)

6.1.1 General info

Tip: Take a look at the [GCMC tutorial](#) and learn how to setup a GCMC calculation.

About Monte Carlo / the Grand Canonical Ensemble

It is best to read a bit about Monte Carlo and ensembles before working with the GCMC code. Almost every book or review text on molecular simulations will do, for example: Frenkel D, Smit B. *Understanding molecular simulation: from algorithms to applications*. Academic Press; 2002. 672 p.

Wikipedia also has some pages of interest:

- http://en.wikipedia.org/wiki/Monte_Carlo_method
- http://en.wikipedia.org/wiki/Grand_canonical_ensemble

It is important to note that this method heavily relies on random numbers, and simulations are thus non-repeatable in detail, but should converge to the same answer.

About the Reaxff GCMC code

The GCMC code for reaxff was originally developed by Thomas Senftle, working as a Graduate Student at Penn State University under the supervision of Dr. Adri van Duin. The original version was a wrapper code that called an external executable to perform the reaxff minimization step and energy calculation, and relied on file modification and parsing to steer the reaxff code and get the results back.

A rewrite of the code, made by Hans van Schoot (SCM) in close collaboration with Thomas Senftle, is now available in the ADF package. The rewrite directly integrates into the ADF-ReaxFF code, solving performance issues of the original code by removing the calling overhead of the reaxff executable and the relatively slow file management. It also merged several modifications of the original code to support the usage of whole molecules for Monte Carlo moves, and supports the usage of multiple atom/molecule types during the simulation. Other improvements were made on the input options, the accessible volume calculation, the MC acceptance prefactor calculation and the writing of logfiles.

The relevant papers are:

Thomas P. Senftle, Randall J. Meyer, Michael J. Janik and Adri C.T. van Duin, *Development of a ReaxFF potential for Pd/O and application to palladium oxide formation*, *J. Chem. Phys.* 139, 044109 (2013) (<https://doi.org/10.1063/1.4815820>)

Thomas P. Senftle, Adri C.T. van Duin, Michael J. Janik, *Determining in situ phases of a nanoparticle catalyst via grand canonical Monte Carlo simulations with the ReaxFF potential*, *Catalysis Communications* volume 52, 5 July 2014, Pages 72–77 (<https://doi.org/10.1016/j.catcom.2013.12.001>)

6.1.2 Input

A worked out GCMC exercise is available from a [2-day ReaxFF workshop](https://www.scm.com/adf-modeling-suite/adf-hands-on-workshops/advanced-2-day-reaxff-workshop/) (<https://www.scm.com/adf-modeling-suite/adf-hands-on-workshops/advanced-2-day-reaxff-workshop/>).

Overview

The GCMC code in ADF-ReaxFF needs the following input files to run:

- control_MC : The GCMC control file, which holds MC settings and the atoms/molecules to insert/move/delete
- control : A reaxff control file, in which only a small number of parameters is of interest
- ffield : A standard reaxff forcefield file
- geo : A geometry file, preferably in biograph format (code not yet tested with xyz)
- iopt : Text file that should only contain a “5” (without the quotes)
- insertData_MC : (optional file) Table used when restarting GCMC simulations

the control_MC file

Lines in the control_MC file that start with ! or # will be ignored, so those can be used for comments. Empty lines are also ignored, so feel free to leave some in the file. Lines with keywords should have their value in the first 8 columns, followed by a couple of spaces (at least 1), followed by the 6-character keyword. The order is free, except for the nmols keyword, which should be the last one. The nmols keyword signals the parser that the next section of control_MC will define X new MC Molecule Types.

This is an example for the control_MC file:

```
! GCMC control file example
  0  iensmb !select MC ensemble (0=Mu-NVT with fixed volume, 1=Mu-NPT with
↳variable volume)
 5000 niter !number of MC iterations to do this simulation
  0  nstart !start the iteration counter with an offset, useful for restarts to
↳avoid double files
 300.0 mctemp !Temperature of the simulation, affects acceptance rate for steps
↳that increase the energy
  0.0 mcpres !NPT pressure in GPa (set to zero for incompressible solid systems
↳unless at very high pressures)
  3.0 rmaxpl !Max radius for atom placement on insert/displace move
  1.2 rminpl !Min radius for atom placement on insert/displace move
 2000 nmctry !Maximum number of trials allowed when inserting or moving a
↳molecule. If the
!           ! rmaxpl and rminpl variables are very strict, this number needs to
↳be large
  1  igcfac !include GC prefactor in probabilities? 0 = no 1 = yes
  0  ivol   !select MC volume calculation technique:
!           ! 0: vvacu needed! volume = total volume - occupied volume -
↳specified vacuum volume (vvacu)
!           ! 1:           volume = total cell volume
!           ! 2: vacc needed! volume = specified accessible volume (vacc)
!           ! 3:           volume = total cell volume - occupied volume
!           ! 4: vacc needed! volume = specified accessible volume (vacc) -
↳occupied volume
```

(continues on next page)

(continued from previous page)

```

435.0  vacc  !if ivol=2 or ivol=4, specify Vacc in angstroms^3
      0.0  vvacu !if ivol=0 specify non-accessible (vacuum) volume Vvacu in angstroms^
↪3
      0.25 ivlim !volume change limit (value between between 0 and 1, Vnew =
↪((1+ivlim)*Vl)
      1  resopt !write restart files: 0=no, 1=yes
      1  resfrq !frequency of writing restart files (MC code only writes files if
↪the MC move is accepted)
      0  debug  !print debug output if set to 1, print even more debug output when
↪set to 2
      5  nmols !Number of MC molecule types, must match the number of molecule
↪blocks that follow!
!
! Molecule Specific Data: C2H2 example
!   This part is fixed format!
!   We need cmpot on line 1,
!   possibly followed by the noinsr on line two,
!   and forced to be ended with nmatom on line 2 or 3, followed by nmatom lines of
↪coordinates.
! the coordinates are FIXED FORMAT! (24d.15,1x,A2) x,y,z (24 wide, 15 after decimal),
↪1 space, 2chars symbol)
-75.00  cmpot  !chemical potential of molecule
      4  nmatom !number of atoms in molecule
      12.1804800000000000      0.4216960000000000      1.3166890000000000 C
      13.1247310000000000      0.3769020000000000      0.5683600000000000 C
      11.3494750000000000      0.4595600000000000      1.9882080000000000 H
      13.9573140000000000      0.3358430000000000      -0.1010000000000000 H

!Molecule Specific Data: C2H4 example
-75.00  cmpot  !chemical potential of molecule
      6  nmatom !number of atoms in molecule
      13.9892220000000000      0.4053910000000000      1.0001500000000000 C
      13.3167840000000000      0.3996460000000000      0.8857950000000000 C
      11.4945130000000000      0.4618370000000000      1.9706120000000000 H
      11.3352190000000000      0.3535770000000000      0.1295810000000000 H
      13.8117010000000000      0.3402240000000000      -0.0840000000000000 H
      13.9705610000000000      0.4533250000000000      1.7562360000000000 H

!Molecule Specific Data: H2O Example
-75.00  cmpot  !chemical potential of molecule
      1  noinsr !setting this to 1 disables insert/deletion moves. If it is set to
↪1 for all types, the ensemble becomes NVT/NPT
      3  nmatom !number of atoms in molecule
      39.9967200000000000      40.7476600000000000      40.5122100000000000 H
      40.0002100000000000      39.9995200000000000      39.9347300000000000 O
      40.0000300000000000      39.2598800000000000      40.5237000000000000 H

!Molecule Specific Data: H2 Example
-75.00  cmpot  !chemical potential of molecule
      2  nmatom !number of atoms in molecule
      5.0258120000000000      0.0000000000000000      0.0000000000000000 H
      5.7741880000000000      0.0000000000000000      0.0000000000000000 H

!Molecule Specific Data: Single atom example
-75.00  cmpot  !chemical potential of molecule
      1  nmatom !number of atoms in molecule
      0.0000000000000000      0.0000000000000000      0.0000000000000000 H

```

The Molecule Specific Data blocks define the molecules (or atoms) that can be inserted/moved/deleted with the MC code. The atoms named here should of course be in the forcefield files, and the coordinates should form a reasonable structure. The MC code uses these coordinates during the insertion step by giving them a random rotation, followed by a random translation to generate a random position of the molecule inside the box. Currently, there is no check to make sure the molecule stays inside the boundaries of the box, the code only checks that the rmaxpl/rminpl values are satisfied. If you plan on inserting large molecules, make sure there is enough room in the rmaxpl value, otherwise the code will stop with an error message.

The chemical potential (cmpot) keyword

The cmpot keyword sets the chemical potential of the molecule (or atom) reservoir, and is employed when calculating the Boltzmann accept/reject criteria after a MC move is executed. This value can be derived from first principles using statistical mechanics, or equivalently, it can be determined from thermochemical tables available in literature sources.

For example, the proper chemical potential for a GCMC simulation in which single oxygen atoms are exchanged with a reservoir of O2 gas, should equal 1/2 the chemical potential of O2 at the temperature and pressure of the reservoir:

$$\text{cmpot} = \text{Mu}_O(T,P) = 1/2 * \text{Mu}_{O_2}(T,P) = 1/2 * [\text{Mu}_{\text{ref}}(T,P_{\text{ref}}) + kT * \text{Log}(P/P_{\text{ref}}) - E_{\text{diss}}]$$

where the reference chemical potential [$\text{Mu}_{\text{ref}}(T,P_{\text{ref}})$] is the experimentally determined chemical potential of O2 at T and Pref, $kT * \text{Log}(P/P_{\text{ref}})$ is the pressure correction to the free energy, and E_{diss} is the dissociation energy of the O2 molecule.

The no insert (noinstr) keyword

The noinstr keyword tells the GCMC code to keep the number of molecules/atoms of this type fixed. It will thus disable Insert/Delete moves on this type, meaning it can only do a displacement move, or volume change move (if the iensmb keyword is set to 1).

the control file

The control file is a regular reaxff control file and it influences the minimization step after an MC trial move. Because of this, only a small number of the reaxff keywords are used during the GCMC simulation.

An example of the control file:

```
# some of the parameters that influence the minimization step in the GCMC code
  1 icentr      Put the center of mass at the center of the cube
  1 igeofo      0:xyz-input geometry 1: Biograf input geometry 2: xmol-input
↪geometry
2.50000 endmm  End point criterion for MM energy minimization
  500 imaxit    Maximum number of iterations
  0 icelop      Optimize cell parameters 0=no 1=yes
1.00050 celopt Cell parameter change
  0 imaxmo      In this case: 0: POLAK_RIBIERE Conj.Grad method, 1: Limited-memory
↪BFGS method
```

The code has been tested with various imaxit and endmm values, the other options have not been fully tested. Other reaxff keywords might also influence the minimization procedure, but those are best left to their default settings

theffield file

Theffield file should be a normal reaxff forcefield file, as described in the reaxff documentation by A. van Duin (visit the documentation section on the SCM website to obtain this document).

thegeo file

The GCMC code has been tested with biograph input files, but other input formats might work. The details of this file are also described in the original reaxff documentation by van Duin.

theiopt file

The `iopt` file is a text file with a single digit inside that selects the execution mode of the `reaxff` code. To run the GCMC code, this file should contain a “5” (without the quotes).

the `insertData_MC` file

The GCMC code can insert multiple atom/molecule types in a single simulation, so it needs to keep track of what atom belongs to which insert. This information is automatically stored and updated when insertion/deletion/moving of atoms or molecules during the simulation, but is by default unknown of the atoms of the starting geometry. The GCMC code will therefore by default not modify the atoms in the original input in the MC trial moves (keep in mind that they can move around during the minimization step). the `insertData_MC` file can be used to tell the GCMC code what atoms in the `geo` file belong to which molecule.

An example of the `insertData_MC` file:

#	<i>atomNumber</i>	<i>MCInsMolType</i>	<i>MCInsertNmbr</i>
	30	1	1
	40	2	1
	46	2	1
	47	1	2
	48	1	3

This example specifies 4 molecules/atoms that are modifiable by the GCMC code, belonging to 2 different GCMC molecules/atoms that are defined in the `control_MC` file. The first “molecule” in the `control_MC` file should thus consist of a single atom (if this doesn’t match, the code will most likely crash!). It was inserted three times (atom 30, 47 and 48) The second molecule has two atoms, and was inserted once.

The atoms do not have a fixed order, and not all atoms have to be defined. If an atom is not appointed to a certain `MCInsMolType` and `MCInsertNmbr`, it will simply not be modified during the MC moves. The `insertData_MCXXXXXX` files generated by the restart option of the code can be directly used as valid `insertData_MC` files, just remove the digits from the filename and replace the `geo` file with the corresponding `geo_MCXXXXXX` file.

6.1.3 Output

Overview

The GCMC code writes a couple of output files, each described in this section. It also produces a number of `reaxff` output files, and some of these are described in the original `reaxff` documentation by van Duin. Keep in mind that these files might not provide a complete or correct picture of the simulation, as they could also contain data originating from rejected MC trial moves.

`geo_MCXXXXXX`

This file is generated every X accepted MC moves and contains the current geometry of the system in biograph format (X is set with the `resfrq` keyword in the `control_MC` file).

`insertData_MCXXXXXX`

This file contains a table of all the atoms in the system with their MC Molecule Type and MC Insert Number. This data can be used to map atoms to an inserted molecule, and is needed if you want to restart your calculation from an accepted MC step. The table contains -1 values for atoms that were in the original input and did not get a manually assigned MCInsert Molecule Type and MC Insert Number, the GCMC code will not modify these atoms during the MC steps.

Also see the section on `insertData_MC` file.

MCstats

The `MCstats` file is a logfile that contains the statistics of the MC simulation. The GCMC code writes a single line to it after every MC step, containing the number of: Tried MC moves (tried), Accepted MC moves (accept), Rejected MC

moves (reject), Accepted Insertion/Deletion/Moving/Volume change MC moves (addAcc/delAcc/mvAcc/volAcc), Rejected Insertion/Deletion/Moving/Volume change MC moves (addRej/delRej/mvRej/volRej)

An example of the MCstats file:

tried	accept	reject	addAcc	delAcc	mvAcc	volAcc	addRej	delRej	mvRej	volRej
0	1	0	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	1	0
2	1	2	1	0	0	0	0	0	2	0
3	1	3	1	0	0	0	0	1	2	0
4	1	4	1	0	0	0	1	1	2	0
5	1	5	1	0	0	0	1	1	3	0
6	1	6	1	0	0	0	1	2	3	0
7	1	7	1	0	0	0	1	3	3	0
8	1	8	1	0	0	0	1	3	4	0
9	1	9	1	0	0	0	1	4	4	0
10	1	10	1	0	0	0	1	4	5	0
11	2	10	2	0	0	0	1	4	5	0
12	3	10	3	0	0	0	1	4	5	0

Elog

The Elog file contains the Volume and energies of the accepted MC steps. The energies in this logfile are the pure ReaxFF energy of the system (RxFFEnergy) and the MC corrected energy, which is used in determining if the step should be accepted or not (see the section on calculating energies for details).

An example of the Elog file:

Iteration	Naccepted	Volume	MC Energy	RxFFEnergy
0	1	15625.00	-3098.88	-3179.88
2	2	15625.00	-3107.92	-3269.92
3	3	15625.00	-3130.13	-3373.13
4	4	15625.00	-3160.05	-3484.05
6	5	15625.00	-3169.77	-3493.77
11	6	15625.00	-3200.13	-3605.13

reaxout.kf

This is the binary logfile generated by the GCMC code. Its contents can be viewed with the KFBrowser utility, or it can be loaded into AMSmovie to view the geometries in the file. Only the data of successful MC moves is written to this file.

6.1.4 Code Details

Overview

The GCMC code will perform niter (control_MC file option) Grand Canonical Monte Carlo trial moves, and accept or reject them based on the Energy produced by the ReaxFF minimization step of the trial geometry. The Monte Carlo algorithm will always accept a step if it results in a decrease of the energy, and accept steps that go up in energy with a probability. This section will give some details about how the code works.

MC Moves (Insert/Delete/Move/Volume)

The GCMC code currently supports 4 types of MC Moves: Insert, Delete, Move (displace), Volume change. The first three moves always change a whole “molecule” of the system, as defined in the control_MC file (a molecule can of course contain only a single atom). Every MC iteration selects one MC Molecule Type from the defined molecules in control_MC at random, followed by a random MC Move (unless there are no molecules of the type in the system, in that case it will do an insert move).

The Insert and Displace (move) MC Moves will generate a random rotation and position for the molecule, and then check if the random positions are within the “RminPI” and “RmaxPI” boundaries (this means no atom in the molecule can be closer to any atom currently in the system than “RminPI”, and it should be within “RmaxPI” distance to an atom in the system). If the conditions are not satisfied, a new set of coordinates is generated and the code checks again. This is repeated a maximum number of “nmctry” times before stopping with an error.

The volume change is controlled by the “ivlim” variable in control_MC. The ivlim sets the volume change limit, and it should be a value between 0 and 1. The new volume will be calculated like this: $V_{\text{new}} = (1 + \text{ivlim}) * V_{\text{old}}$.

Calculating energies

Because the GCMC simulation adds and deletes atoms or molecules during the runtime, it cannot directly compare the ReaxFF energies for the MC acceptance criteria: inserting a molecule will usually lower the total energy of the system, causing the MC to always accept it, and always reject a deletion. To balance this out, the GCMC code calculates a “corrected” MC energy to compare the trial reaxFF energy with, consisting of the previously accepted ReaxFF energy + the chemical potential (cmpot in control_MC) for the inserted molecule, or the ReaxFF energy - the chemical potential for the deleted molecule. The volume change energy is also corrected, using the following formula:

$$E_{\text{MC_Corr}} = E_{\text{reaxff_last_accept}} - (\text{Pressure} * 0.1439 * (\text{newV} - \text{oldV})) + ((1.0/\beta) * n_{\text{InsertedMols}} * \log(\text{newVavail}/\text{oldVavail}))$$

where newVavail and oldVavail are calculated from the MC available volume (see the section calculating volumes).

Calculating volumes

The GCMC code can calculate the available volume in a couple of different ways, depending on the ivol setting in control_MC:

- ivol = 0: volume = total volume - occupied volume - specified vacuum volume (vvacu)
- ivol = 1: volume = total cell volume
- ivol = 2: volume = specified accessible volume (vacc)
- ivol = 3: volume = total cell volume - occupied volume
- ivol = 4: volume = specified accessible volume (vacc) - occupied volume

Where the occupied volume is calculated by summing up the volumes of the atoms in the geo file that are not specified to be part of an MC type molecule. The volume of an atom is calculated using the average of the covalent atomic radius and the vd Waals radius of the atom, which are found in the reaxff forcefield file (ffield).

the vacc and vvacu options can be specified in the control_MC file to get a more accurate available volume.

Acceptance criteria

An MC move is always accepted if the reaxff energy is lower than the corrected MC energy of the last accepted MC move, or if the energy increase is small enough. If the new energy is higher, the code generates a random number between 0 and 1, and accepts the move if the random number is bigger than:

$$\text{prob} = \text{preFactor} * \exp(-\beta * \Delta E)$$

Where the prefactor is calculated (for insert and delete moves) using the deBroglie wavelength of the inserted molecules, the number of inserted molecules and the available MC volume of the system.

6.2 Force-bias Monte Carlo (fbMC)

The Uniform-acceptance force-bias Monte Carlo (fbMC) method (Phys. Rev. B 85, 134301 (2012) (<https://doi.org/10.1103/PhysRevB.85.134301>)) has been implemented in ADF ReaxFF by the group of Erik Neyts.

Usually, ReaxFF optimizations are alternated with fbMC steps to accelerate events such as reactions. The acceleration (MC step) is controlled by the temperature and maximum displacement, which should be balanced to keep close to the real dynamics. fbMC does not scale time uniformly (*J. Chem. Phys.* 141, 204104 (2014) (<https://doi.org/10.1063/1.4902136>)). Diffusion coefficients and reaction rates should probably use quartic mass scaling.

fbMC mass scaling can be set. It defaults to quadratic scaling as in the original fbMC implementation. Setting *imcroo* to 4 would mean quartic mass scaling.

Input keywords

These keywords should be set at the end of the *control* file. fbMC control parameters can also be set in the GUI.

- *drmax* – maximum displacement
- *imcfrq* – frequency of fbMC steps
- *imcstp* – number of fbMC steps
- *imcroo* – mass scaling (defaults to 2, 4 is recommended as per the Bal & Neyts paper)

A worked out example for fbMC versus unaccelerated ReaxFF MD for healing graphene has been made for the [2-day ReaxFF workshop](https://www.scm.com/adf-modeling-suite/adf-hands-on-workshops/advanced-2-day-reaxff-workshop/) (<https://www.scm.com/adf-modeling-suite/adf-hands-on-workshops/advanced-2-day-reaxff-workshop/>).

6.3 CVHD: Collective Variable-Driven Hyperdynamics

Tip: See also the advanced [cvhd tutorial](#) .

The CVHD implementation in Reaxff follows the algorithm described in K.M. Bal, E.C. Neyts, *JCTC*, 11 (2015) (<https://doi.org/10.1021/acs.jctc.5b00597>)

Currently only the bond-breaking collective variable is implemented. The CVHD takes a *cvhd.in* file as the only additional input. The file describes both the hyperdynamics and the collective variables and it has the following format:

```
# Comments may be present, starting with a hash sign.
# CVHD bias potential parameters
startIter  freqIter  waitIter  gaussHeight  gaussHalfWidth  [wellTemperedT]
# Bond collective variable components (one per line, up to 10)
type  AtomName1  AtomName2  rmin  rmax  BOcutoff  pExponent
[type  AtomName1  AtomName2  rmin  rmax  BOcutoff]
```

The bias potential parameters are: *startIter* - the first iteration at which the bias potential is deposited, *freqIter* - bias potential deposition rate in steps, *waitIter* - number of steps to wait before removing the bias after CV becomes equal to 1, *gaussHeight* - height of a single gaussian deposited at a time in kcal/mol, *gaussHalfWidth* - gaussian delta of the deposited bias, *wellTemperedT* - the bias temperature as defined in [*PhysRevLett* 100, 020603 (2008)] (optional, the full bias is applied if *wellTemperedT* is absent).

The bond-breaking CV parameters: *type* = bond, *AtomName1* and *AtomName2* - atom names specified in the second atom name entry (the one after the coordinates) in the BGF-type geo file, which may be different from the element name specified before the coordinates; *rmin* and *rmax* - *rmin* and *rmax* from Eq 5 in the paper above, *BOcutoff* - bond order cutoff, bonds with BO below this value are not considered even if the bond distance is within the (*rmin*,*rmax*) interval; *pExponent* - corresponds to *p* in Eq 3 in the paper. The same *pExponent* value is used for all bond types but the *rmin*, *rmax*, and *BOcutoff* may (and should) be different.

An example `cvhd.in` file that one could use to reproduce the NVT pyrolysis modeling results from [this paper](https://doi.org/10.1039/c6sc00498a) (<https://doi.org/10.1039/c6sc00498a>) is given below:

```
# CVHD bias potential parameters
# startIter  freqIter  waitIter  gaussHeight  gaussHalfWidth
  10000      2000      10000      0.250        0.025
# Bond-breaking collective variable
# type  AtomName1  AtomName2  rmin  rmax  BOcutoff  pExponent
bond    C          H          1.05  1.65  0.5      6
bond    C          C          1.55  2.2   0.5
```

6.4 The Molecule Gun

The molecule gun allows you to “shoot” (add with velocity) a molecule into the simulation box. Molecules can be continuously added to the simulation or only once. The initial position as well as the initial velocity can be pre-set or random w.r.t. to all or a selection of X,Y,Z-components. Possible applications of the molecule gun include e.g. the simulation of enforced collisions or deposition processes on surfaces.

See also:

[Mol. Gun GUI tutorial](#)

The molecule gun requires an additional input file called **addmol.bgf**. It contains the geometry of the particle to be shot in BGF format and some additional keywords.

The following keywords may be specified in the **addmol.bgf** file:

Input Keywords

- *FREQADD* – Add new molecule every N-th step but not before the iteration specified in *FIRSTADD*
- *FIRSTADD* – Add the first molecule at iteration N
- *VELADD* – Velocity of the added molecule: 1 - random, 2 - specified in the `addmol.vel` file, 3 - via a temperature specified with *TADDMOL*
- *STARTX* – The initial X coordinate of the added molecule, random if not specified
- *STARTY* – The initial Y coordinate of the added molecule, random if not specified
- *STARTZ* – The initial Z coordinate of the added molecule, random if not specified
- *ADDIST* – Minimum distance from the molecule to other atoms for successful addition
- *NATTEMPT* – The program will try to add a molecule N times until the *ADDIST* criterion is met
- *TADDMOL* – for *VELADD*=1|3: temperature for the velocity of the molecule as a whole
- *ROTATE* – The molecule is rotated randomly before being added
- *SIGMAX* – sigma parameter for the Gauss distribution of the initial X coordinate around *STARTX*
- *SIGMAY* – sigma parameter for the Gauss distribution of the initial Y coordinate around *STARTY*
- *SIGMAZ* – sigma parameter for the Gauss distribution of the initial Z coordinate around *STARTZ*
- *VELXYZ* ($V_x V_y V_z$) – for *VELADD*=3: the direction vector for the initial velocity of the added molecule
- *SIGMAT* – for *VELADD*=1|3: standard deviation for *TADDMOL*
- *TDISPER* – for *VELADD*=3: each atom’s velocity is modified with $V_{\{x,y,z\}} = \sqrt{RT/m} * \text{random_normal}(0,1)$

Example

The following example **addmol.bgf** file will create one carbon atom every 1000th iteration starting from iteration 500. The added atom will have an initial temperature of 800K and will initially move in the positive direction along the Z axis.

```
BIOGRF 200
DESCRP C
REMARK Created by AMSinput
VELADD 3
FIRSTADD 500
FREQADD 1000
STARTX 1.671997104
STARTY -4.00486678
STARTZ -0.3259654382
ROTATE 1
TADDMOL 800
VELXYZ 0.0 0.0 5.0
HETATM 1 C 1.67200 -4.00487 -0.32597 C 1 1 0.0
END
```

Tip: Use the GUI to setup a molecule gun calculation and edit the .run file created by the GUI to minimize typos. The section *addmol.bgf* is found at the end of the run file just before the *control* section.

Changes with respect to the original reac program

It is possible to add some randomness both to the pre-set position and to the velocity: the SIGMA{X,Y,Z} keywords specify a standard deviation value for the Gauss distribution around the START{X,Y,Z} values, respectively. If "ROTATE 1" is specified, the molecule will be rotated randomly around its geometric center before being added to the system.

A VELADD=3 option has been added. With this option, it is possible to specify the velocity vector's direction via VELXYZ. This vector is normalized so the exact magnitude of its components does not matter. The magnitude of the velocity along this vector is determined by the TADDMOL and SIGMAT parameters as $\sqrt{3nRT/M}$, where $T=TADDMOL + SIGMAT * \text{random_normal}(0,1)$, n - number of atoms in the molecule, R - gas constant, M - molar mass. Additionally, the velocity of each atom may be modified with a random component based on the temperature specified by the TDISPER keyword. On average, the total temperature of the added molecules should be close to TADDMOL+TDISPER.

Molecules will never be added before the step specified by the FIRSTADD keyword. Thus, a molecule will always be added at the FIRSTADD step and then at every iteration divisible by FREQADD. If, for example, FIRSTADD=1999 and FREQADD=1000 then the first molecule will be added at iteration 1999, the second at iteration 2000 and then every 1000-th iteration after that.

6.5 Bond Boost Method

Tip: See also the advanced [bond boost tutorial](#) .

The bond boost method implemented here is described in [A. Vashisth, C. Ashraf, W. Zhang, C.E. Bakis, and A.C.T. van Duin, *Accelerated ReaxFF simulations for describing the reactive cross-linking of polymers*, J. Phys. Chem. A, 2018, 122 (32), 6633-6642]. In this method, the distances between atoms that are relevant to the reaction of interest are calculated to determine the orientation of the reactant molecules. If a suitable initial configuration is recognized, an additional restraint energy (possibly consisting of more than one term) is added to the system that is intended to

stretch or compress bonds at a pre-defined rate such that this additional energy can help achieve the energy to cross the reaction barrier. A single term of the restraint energy is given by

$$E_{ij}^{rest} = F_1 * (1 - e^{-F_2(R_{ij}-R_0)^2})$$

Where E_{ij}^{rest} - restraint energy between atoms i and j in kcal/mol; F_1 and F_2 - parameters for restraint energy in units of kcal/mol and \AA^{-2} , respectively; R_0 - target distance (in \AA) between two atoms according to the restraint; R_{ij} - actual distance (in \AA) between the two atoms under consideration.

The input for this method is given via the tracking.in file, which has the following format:

```
#Distance regimes
#T pair1: at1 at2 min max pair2: at1 at2 min max pair3: at1 at2 min max pair4: at1
↪at2 min max pair5: at1 at2 min max
5 Si O 1.5 1.8 O H 0.9 1.3 H O 0.9 1.3 O
↪Si 1.5 1.8 Si O 1.5 1.8
#Restraint settings
#iter nr. r1: at1 at2 R0 F1 F2 r2: at1 at2 R0 F1 F2 r3: at1 at2 R0 F1
↪ F2 r4: at1 at2 R0 F1 F2
10000 4 2 5 1.80 500.0 1.0 2 3 3.00 100.0 0.10 3 6 1.00 500.
↪0 0.25 5 6 3.50 250.0 0.1
```

A hash sign in the first position on a line begins a comment. The first non-comment line contains atom symbols and min/max distances for detection of the initial configuration. The first number (5 in the example above) specifies the number of atom pairs. For each pair, the minimum and maximum distances are specified and the first atom in each pair after the first corresponds to the second atom in the previous one. This means that the first element symbol in pair 2, 3, etc. is actually ignored. Thus, the example above defines a chain of atoms Si-O-H-O-Si-O with R(Si-O) in the (1.5,1.8) range and R(O-H) in the (0.9,1.3) range. It should be noted that when detecting the initial configuration, an atom may enter the chain more than once. This allows ring configurations to be specified although the ring must consist of at least 4 atoms.

Note: When detecting coordinates, the program uses alternative atom names specified after the atomic coordinates in the *geo* file. Since this name is generally not used to detect the ReaxFF element name, one can allow only a subset of A-B bonds to be boosted by giving certain atoms different alternative names. If more than one suitable configuration is found the one with the smallest sum of distances is used to create the restraints.

When a configuration matching these criteria is found, parameters specified on the second non-comment line are used to add restraints. The first number specifies the number of time-steps the restraint must remain active for (10000 in this case, which should be long enough for the reaction to take place). During this time no configuration detection is performed. In other words, only one set of restraints may be active at a given time. The second number on the line specifies the number of restraints to add (4 in this case). For each restraint, two indices and three parameters are specified. The indices correspond to positions of the atoms in the chain specified on the first line. In the example above, the restraint is set in such a way that the O2-H3 and Si5-O6 bonds are broken and Si5-O2 and O6-H3 bonds are formed. The results of the detection are written to the fort.84 file together with the imposed restraints.

OTHER NEW FEATURES AND FIXES

7.1 Repeating the unit cells

A simple way to increase the size of the simulation box is by multiplying it a number of times in the a, b, and c directions. In the ReaxFF program this can be achieved by specifying the *reptx*, *repty*, and *reptz* keys (for the a, b, and c direction, respectively) in the control file. The specified value must be strictly positive and the default is 1 (one) for each of them.

7.2 Molecular charges

In order to freeze the total charge of a group of atoms (for example, a molecule) one can add a MOLCHARGE key to the geo file in the BGF format. The format of the key is as follows:

```
MOLCHARGE  n1 n2 q
```

where *n1* and *n2* specify indices of the first and the last atoms of the molecule and *q* is the charge. All atoms between *n1* and *n2* will be included in the molecule. When specifying the MOLCHARGE key *all* atoms present in the system must be assigned to one of the the molecules. For example, if your system consists of 10 water molecules and the charge of the first of them is constrained to be +1 and the total charge of the system is zero then the following two MOLCHARGE keys must be specified:

```
MOLCHARGE  1  3  1.0  
MOLCHARGE  4 30 -1.0
```

Note: The MOLCHARGE is implemented as a simple constraint in the charge equilibration procedure and it does not take any bonding information into account. This means that specifying molecular charges using the MOLCHARGE key makes sense only if you are sure that the constitution of the molecule will not change during calculation. This is useful, for example, when fitting EEM-related parameters in a force-field.

7.3 External electric fields

External electric field regimes are specified in an external file `eregime.in`. The presence of this file in the ReaxFF run directory will activate the electric fields.

The format of this file is as follows:

```
#Electric field regimes
#start #V direction1 Magnitude1 (V/A) direction2 Magnitude2 (V/A)
0000 1 x 0.010000
1000 1 x -0.010000
2000 1 y 0.010000
3000 1 y -0.010000
4000 2 x -0.010000 y -0.0100
5000 2 x 0.010000 y 0.0100
```

Note: The electrostatic potential is sawtooth-like, which requires a vacuum zone (we suggest 10Å) on the box plane perpendicular to the efield vector. Otherwise one risks abnormal polarization for atoms on two sides of the plane. A good strategy to ensure the vacuum zone is to define an elastic wall restraint (see below).

Example:

Adding the following lines to your run script:

```
cat > eregime.in <<eor
#Electric field regimes
#start #V direction Magnitude (V/Angstrom)
0000 1 x 0.010000
1000 1 x -0.010000
eor
```

will overlay an external electric field with magnitude 0.01 V/Å in x-direction for the duration of 1000 iterations, then the magnitude is switched to -0.01 V/Å and kept for the rest of the simulation.

7.4 Elastic wall restraint

The wall is implemented as a $\cos^2(x)$ function (one period) between Coord-HalfWidth and Coord+HalfWidth with a maximum at Coord. The Height value is given in kcal/mole. At this moment, the elastic wall restraint can only be used with an orthorhombic unit cell.

The wall is specified in the geo file as:

```
EWALL RESTRAINT AxisIndex Coord HalfWidth Height
```

For example, for a wall that crosses the Z axis at z=7 Angstrom, is 2.0 Angstrom thick and 100 kcal/mole high, the input would look like this:

```
EWALL RESTRAINT 3 7.0 1.0 100.0
```

Note: We recommended to set the icentr option in the control file to 0 to avoid inconsistencies between the initial atomic coordinates and the wall position.

7.5 ACKS2: Atom-condensed Kohn-Sham DFT approximated to second order

The ACKS2 charge equilibration scheme has been implemented following the original paper “ACKS2: Atom-condensed Kohn-Sham DFT approximated to second order” by T. Verstraelen et al. *J. Chem. Phys.* 138 (2013) 074108.

Using the ACKS2 scheme requires a suitable force-field, which is recognized by “[acks2]” at the start of the first line of force field file (note: the spaces around “acks2” are important!). Besides, the *icharg* parameter in the control file must be set to 7. In addition to the general EEM parameters the ACKS2 scheme needs the general force-field parameter #35 (“Xamp”) and the atomic cut-off parameter #23 (“softcut”).

7.6 Correction for small bond orders

There is a discontinuity problem for small bond orders in the expressions for covalent and hydrogen bonds, valence and torsion angles. The discontinuity and the correction for it are described in detail in *J. Phys. Chem. Lett.* 10 (2019) 7215 (<https://doi.org/10.1021/acs.jpcclett.9b02810>).

The correction can be switched on by setting the taperb control flag to 1.

7.7 Correction for torsion angles asymptotics

The discontinuity at small bond orders in the expression for torsion angles and conjugation contributions can alternatively be corrected for using the tors13 correction. The corresponding terms are given by f_{10} (eq. 10b) and f_{12} (eq. 11b) in the [original ReaxFF paper](https://doi.org/10.1021/jp004368u) (<https://doi.org/10.1021/jp004368u>). The new expression for each term in f_{10} is $(1 - e^{-2\lambda_{23}BO^2})$ and in f_{12} the new expression is $\sin(\frac{\pi}{3}BO)^4$. The new expressions ensure correct asymptotic behavior for the dE/dBO for $BO \rightarrow 0$.

The correction can be enabled by setting the tors13 flag in the control file to 1.

Another discontinuity in the torsion angle term is when one or both valence angles approach 180 degrees. It is described in detail in *J. Chem. Phys.* 153 (2020) 021102 (<https://doi.org/10.1063/5.0013906>).

The corresponding correction can be enabled by setting the tors19 flag in the control file to 1.

Using the above corrections can make geometry optimizations more stable and improve conservation of energy during MD.

7.8 LG dispersion

The LG dispersion correction was implemented following the paper Liu et al., *ReaxFF-Ig: Correction of the ReaxFF Reactive Force Field for London Dispersion, with Applications to the Equations of State for Energetic Materials*, *J. Phys. Chem. A*, 2011, 115 (40), pp 11016–11022 (<https://doi.org/10.1021/jp201599t>)

The LG dispersion correction is turned on when using a suitable forcefield, which is recognized by the “[IgDispersion=1]” key in the file header.

7.9 The rxffutil utility

Usage:

```
rxffutil command [arguments]
```

7.9.1 reformat

The **reformat** command tells rxffutil to read a (possibly mis-formatted) force-field file from standard input and write it in the format required by the ReaxFF program:

```
rxffutil reformat < input_ffield_file
```

The result is written as a ‘ffield’ file in the current directory. The input can be almost free-format but the following points need to be taken into account:

- If you copy-paste force-field parameters from a pdf document (for example, from a publication’s supporting info), then make sure to remove any page numbers that may have been copied together.
- The atom types section header (with the number of atom types in it) must have three comment lines following the line with the number of atom types.
- Likewise, the bond types section must have one comment line following the line with the number of bond types. Any comments provided on input are ignored and the standard comments are written instead.
- Numbers must be separated by at least one space

7.9.2 minmax

The **minmax** command is intended for analyzing existing forcefield files. It will print the number of entries, as well as the minimum, maximum and average values for the given parameters:

```
rxffutil minmax [--ffdir=/path/to/ff/directory] key ...
```

Command arguments:

- - *ffdir=/path/to/ff/directory* – scan the specified directory for ff files when computing parameter ranges. The \$AMSRRESOURCES/ForceFields/ReaxFF directory is used if this option is absent.
- *key* (mandatory) – specifies which parameters to gather data for. Multiple keys may be specified. The syntax of the keys is as follows:
 - *gen:i* – the i-th general parameter;
 - *A:i* – the i-th item of the A atom type block;
 - *b%AB:i* – the i-th item of the A-B bonds block;
 - *o%AB:i* – the i-th item of the A-B off-diagonal block;
 - *v%ABC:i* – the i-th item of the A-B-C valence angle block;
 - *t%ABCD:i* – the i-th item of the A-B-C-D torsion angle block;
 - *h%XHY:i* – the i-th item of the X-H-Y hydrogen bond block.

The index part of the key (:i) may be omitted, in which case the whole block is considered. In order to match *wildcard* torsions (indicated by 0 in the ff file) a question mark “?” instead of the atom type may be used. Examples of valid keys:

```
v%HPtH:1 - first parameter of the H-Pt-H valence angle block
v%HPtH   - all parameters of the H-Pt-H valence angle block
t%?CC?:4 - fourth parameter of the *-C-C-* torsion angle block
```

Note that some angles may have more than one optimal value (for example 90 and 180 degrees), in which case ReaxFF may have more than one block per key in a force-field file. For such a key, the number of entries may be larger than the number of scanned files. In the following example, the number of entries for a C-O-O-C torsion is larger than that for the C-O bond because twelve force-fields contain two C-O-O-C torsion angles each:

```
$ rxfutil minmax b%CO:1 t%COOC:1
Key          N          min          max          average
t%COOC:1     64       -2.0324        3.4682        0.8679
b%CO:1       52        95.6228       197.3588       151.8473
```

One can see which files contain the given keys using the `-vv` verbosity option. There are three verbosity levels:

- `-v` – log ff files as they are opened, read in and parsed;
- `-vv` – as `-v` plus print a parameter value for each case of the found key together with the corresponding file and block index within the section;
- `-vvv` – as `-vv` plus print a key-block_index pair for every block found when parsing ff files.

For example:

```
$ rxfutil minmax -vv v%HPtH:1
... [snip] ...
Key          N          min          max          average
Found v%HPtH:1     95.0000 in c:/adf2018.105/atomicdata/ForceFields/ReaxFF/
↳CHONSFPtClNi.ff block 69
Found v%HPtH:1     57.3916 in c:/adf2018.105/atomicdata/ForceFields/ReaxFF/
↳PtCH.ff block 14
v%HPtH:1         2         57.3916        95.0000        76.1958
```

7.9.3 pdist

The `pdist` command can be used in addition to `minmax` to get a better idea about the parameter values distribution:

```
rxffutil pdist [--ffdir=/path/to/ff/directory] [--npoints=int] key ...
```

Keys used with the `pdist` command are required to have a non-empty index part. That is, `v%HPtH` is not a valid key while `v%HPtH:1` is.

The number of points (bins) in the distribution is set by the `-npoints` option (20 by default).

7.9.4 params

The `params` command can be used to prepare a `params` file used by ReaxFF for force-field fitting:

```
rxffutil params [options] key ...
```

In additions to `-ffdir`, the following options are available:

- `-ff=/path/to/ffield` – specify the `ffield` file for which the `params` file is to be created. If this option is omitted, the `ffield` file in the current directory is used. If there is no such file the program will quit.

- `--delta=real` – specify a value for the *delta* (or *sigma*) column of the *params file* (page 15) as a fraction of the corresponding min-max interval. The default value is 0.01.
- The min-max interval for each parameter is determined from parameter values (having the same key) found in the `ffdir` directory. This range is initially identical to the one printed by the `minmax` command but it can be modified using the following options:
 - `--scalebound=real` – change the parameter range by scaling both boundaries by the specified factor. The boundaries are scaled in such a way that the range is expanded if `scalebound` is greater than 1, or reduced otherwise. In other words, a negative lower bound is multiplied by the factor, while a positive one is divided. Likewise, a positive upper bound is multiplied by the factor, while a negative one is divided by it. This way, the bounds do not change their sign when extending or shrinking the parameter range (the `scalebound` must be positive);
 - `--scalerange=real` may be specified when the `--scalebound` option is not used (otherwise it is ignored). If this option is present then the initial range is multiplied by the given factor and the lower and upper bounds are shifted symmetrically. Using this option may change the sign of the parameter boundary, which may lead to undesired consequences;
 - if none of the above options is specified then the lower and upper bounds may be scaled independently using the `--scalemin=real` and `--scalemax=real` options. When using this option the range can be expanded or shrunk depending on the sign of the boundary value.

For example, to get a `params` file for optimization of the C-O-O-C torsion angle parameters from the CHO-2016 force-field using MCFFOptimizer one can use the following commands:

```
$ cp $AMSRESOURCES/ForceFields/ReaxFF/CHO-2016.ff ./ffield
$ rxffutil params --scalebound=1.2 t%COOC:1 t%COOC:2 t%COOC:3 t%COOC:4 t%COOC:5
#      Key      delta      min      max      comment
6  16  1      0.066007    -2.4389    4.1618    # t%COOC:1 V1
6  16  2      1.168364   -20.8364   96.0000   # t%COOC:2 V2
6  16  3      0.030000   -1.8000    1.2000    # t%COOC:3 V3
6  16  4      0.061995   -8.2829   -2.0833   # t%COOC:4 p_tor1
6  16  5      0.034717   -3.4800   -0.0083   # t%COOC:5 p_cot1
```

In this example, the range is expanded by scaling the lower and upper bounds by 20% outwards. Exercise: run this command with and without the `--scalebound` option and compare min max values.

When preparing a `params` file for the CMA-ES optimizer one should use the `--delta` option to specify a suitably large initial sigma (typically in the order of 0.1 – 0.25).

Note: When optimizing valence and torsion angle parameters, care should be taken when selecting starting parameters for a parameters that are present more than once in the forcefield in question, e.g. the C-O-O-C torsion that has two optimal values in many force-fields. The average printed by the program is based on all entries so it will be somewhere in the middle between the optimal values, i.e. highly non-optimal.

Note: The fields in the output of the `params` command are sorted by the key (actually by its integer equivalent), while the output from the `minmax` command has an arbitrary order.

Note: When working on a real force-field fitting project one should not use the default `ffdir` value (`$AMSRESOURCES/ForceFields/ReaxFF`). Instead, one should copy related `ff` files from there to a separate folder and use it. By related files one should understand, for example, the ones corresponding to the same branch (combustion or water) and/or having the same kind of non-bonding parameters (shielding/no-shielding, inner-core/no-inner-core).

TROUBLESHOOTING

8.1 Geometry optimization

Geometry optimization issues with reaxff are usually caused by the fact that the derivative of the ReaxFF energy function has discontinuities. These discontinuities are often related to the bond order cutoff (cutof2 in the control file, 0.001 by default). This cutoff determines whether a valence or a torsion angle is included in the potential energy evaluation or not, i.e. when at least one of the bonds forming an angle is of order lower than the cutoff then the angle is discarded. Thus, when the order of a particular bond crosses the cutoff value between two optimization steps, the energy derivative (the force) experiences a sudden change. The magnitude of the jump depends on the cutoff value. The default value of 0.001 is normally not a problem during MD but it may break the geometry optimization convergence.

There are a two ways to reduce the discontinuity and thus to improve the stability of the geometry optimization. One way is to switch on the 2013 formula for the torsion angles (the tors13=1 key in the control file). This will make the torsion angles change more smoothly at lower bond orders but it will not affect the valence angles. You should be aware, however, that using this option changes the bond order dependence of the ReaxFF 4-center term.

The other way is to decrease the cutof2 value in the control file. This will significantly reduce the discontinuity in the valence angles and somewhat in the torsion angles but will not remove it. This will also make the calculation somewhat slower because more angles will have to be included in the computation.

REAXFF AS AN AMS ENGINE

The ReaxFF AMS engine enables efficient parallel calculations using this parametrized reactive molecular mechanics potential. The engine shares all core routines with the *standalone ReaxFF program* (page 1), guaranteeing identical energies and forces between these two codes.

For more information, see the [ReaxFF Engine manual](#).

TECHNICAL TOPICS

10.1 Hybrid parallelism

Reaxff, both as a program and as an AMS engine, has been parallelized using both MPI and OpenMP. MPI uses multiple processes to share the work, while OpenMP uses multiple threads within the same process. Each parallelization method has its pluses and minuses. The OpenMP usually has a lower overhead than MPI but MPI can be used on a distributed-memory machine (compute cluster). It is possible to combine both methods in a single job, which may lead to a lower performance because of the double overhead. For small systems pure OpenMP can be much faster than hybrid OpenMP+MPI or pure MPI because of the relatively high cost of MPI communication.

Number of OpenMP threads

The number of OpenMP threads is controlled by the `OMP_NUM_THREADS` environment variable. If `OMP_NUM_THREADS` is not defined the program will set the number of threads to the minimum of the default number of threads from the OpenMP library and the number of physical processor cores in the machine divided by the number of MPI processes running on it. The default number of threads returned by the OpenMP library depends on different factors: the number of the machine's virtual processors, the MPI and the OpenMP library used and the batch system, if any. For example, when running Reaxff on a quad-core desktop computer with hyperthreading (8 virtual processors) with IntelMPI using 4 MPI processes, the default number returned by the Intel OpenMP library will usually be 2 because the library is aware of the fact that it is running under MPI. The same OpenMP library running under a different MPI run-time, for example OpenMPI, will probably set the default number of threads to 8. In both cases, ReaxFF will use one OpenMP thread per process because it counts only physical cores and is aware of the job's MPI configuration.

When running under a batch system the default number of OpenMP threads will depend on the quality of integration of the batch system with the MPI runtime and the OpenMP library. For example, when running an IntelMPI version under SLURM, the number of threads should be set using the `-cpus-per-task` option to set the number of threads.

The `OMP_NUM_THREADS` environment variable's value overrides any defaults or heuristics when present.

Recommendations

For single-node calculations we recommend using pure OpenMP. To this end, set the `NSCM` environment variable to 1 and set `OMP_NUM_THREADS` to the number of processors to use or leave it undefined to use all cores.

When running on multiple nodes we recommend using pure MPI. To this end, set the `OMP_NUM_THREADS` environment variable to 1 and set other MPI-related options as you would do for any other AMS job. This is especially important if you are using an MPI runtime environment loosely integrated with the cluster's batch system.

WARNINGS

WARNING: Inconsistent vdWaals-parameters in forcefield. See output All atom types in a force-field file should normally have consistent Van der Waals screening and short-range repulsion (a.k.a. inner core) parameters. This warning is printed if two atom types have inconsistent parameters.

WARNING: Suspicious force-field EEM parameters for ... For every atom type, the eta and gamma parameters for the electronegativity equalization method (EEM) should satisfy the relation: $\eta > 7.2 \cdot \gamma$. Otherwise a polarization catastrophe is likely to occur at a short interatomic distance, the critical distance being dependent on the eta/gamma ratio. The smaller the ratio the larger the distance at which this occurs. When the polarisation catastrophe occurs the amount of charge that flows from one atom to the other can become very large. The EEM routine checks that the resulting charge for each atom lies within the $[-10, Z]$ interval, where Z is the number of electrons in the neutral atom, and throws an error if the check fails.

INCLUDED FORCEFIELDS

See also [Included Forcefields \(development version\)](#)

Description of ReaxFF force fields

Disclaimer: Using these forcefields for systems they have not been explicitly trained against may produce unrealistic results. Please see the full manuscripts for more detailed information.

The force field files used by the SCM version of ReaxFF are compatible with those used by the original ReaxFF code. So if you have the force field information from somewhere else you can just use it (save it in a text file, and select it in AMSinput via the Other... option).

There are currently two major groupings (i.e., the ReaxFF branches) of parameter sets that are intra-transferable with one another: (1) the combustion branch and (2) the aqueous (water) branch. The major difference between these two branches is in the O/H parameters, where the combustion branch focuses on accurately describing water as a gas-phase molecule, and the water branch is targeted at aqueous chemistry.

AB.ff: (H/O/N/B) Ammonia Borane M.R. Weismiller, A.C.T. van Duin, J. Lee, R.A. Yetter *ReaxFF Reactive Force Field Development and Applications for Molecular Dynamics Simulations of Ammonia Borane Dehydrogenation and Combustion* *J. Phys. Chem. A* 2010, 114, 5485-5492 (<http://dx.doi.org/10.1021/jp100136c>)

- QM data were generated describing the single and (if relevant) double and triple bond dissociation for all B/N/O/H combinations. These data were used to derive initial ReaxFF bond parameters, and all calculations were performed using DFT with the B3LYP functional and the Pople 6-311G** basis set.
- The training set was then extended with QM data describing angular distortions in a set of small AB-related (AB H₃N-BH₃) molecules. These data were used to derive the initial ReaxFF angular parameters.
- The training set was extended with reaction barriers for key reaction steps such as H₂ release from AB, dimerization of H₂B-NH₂ and reaction energies associated with H₂ release from AB and with AB oxidation.
- Branch: combustion.

AuCSOH.ff: (Au/C/S/O/H) J.A. Keith, D. Fantauzzi, T. Jacob, and A.C.T. van Duin *Reactive forcefield for simulating gold surfaces and nanoparticles* *Physical Review B* (2010) 81, 235404-1/235404-8 (<http://dx.doi.org/10.1103/PhysRevB.81.235404>)

- The original Au-Au parameters were extended by three publications:
- Au/O: K. Joshi, A.C.T. van Duin, and T. Jacob Development of a ReaxFF description of gold oxides and initial application to cold welding of partially oxidized gold surfaces *Journal of Materials Chemistry* 20, (2010), 10431-10437 <http://dx.doi.org/10.1039/C0JM01556C>
- Au/C/S/H: T.T. Jarvi, A.C.T. van Duin, K. Nordlund, and W.A. Goddard Development of interatomic ReaxFF potentials for Au-S-C-H systems *Journal of Physical Chemistry C* 115, (2011), 10315-10322 <http://dx.doi.org/10.1021/jp201496x>

- C/O/H/S: Rahaman, O., van Duin, A. C. T., Goddard, W. A., III, and Doren, D. J. Development of a ReaxFF reactive force field for glycine and application to solvent effect and tautomerization *Journal of Physical Chemistry B* 115 (2011), 249-261 <http://dx.doi.org/10.1021/jp108642r>
- The forcefield does not include Au/N parameters
- Branch: water.

CHO.ff: (C/H/O) Hydrocarbon oxidation K. Chenoweth, A.C.T. van Duin, W. A. Goddard *ReaxFF Reactive Force Field for Molecular Dynamics Simulations of Hydrocarbon Oxidation* *J. Phys. Chem. A* 2008, 112, 1040-1053 (<http://dx.doi.org/10.1021/jp709896w>)

- To obtain the H/C/O compound data required to extend the hydrocarbon-training set, DFT calculations were performed on the following systems: (a) dissociation energies for various bonds containing carbon, oxygen, and hydrogen. The ground state structure was obtained through full geometry optimization. Dissociation curves were calculated by constraining only the bond length of interest and re-optimization of the remaining internal coordinates. Optimization was also performed for the various angles and torsions associated with C/H/O interactions.
- Branch: combustion.

HCONSB.ff: (H/C/O/N/S/B) M.R. Weismiller, A.C.T. van Duin, J. Lee, and R.A. Yetter, *ReaxFF Reactive Force Field Development and Applications for Molecular Dynamics Simulations of Ammonia Borane Dehydrogenation and Combustion* *J. Phys. Chem. A* (2010), 114, 5485-5492. (<http://dx.doi.org/10.1021/jp100136c>)

- The parameters in this forcefield were extended/improved by two other publications:
- A.M. Kamat, A.C.T. van Duin, and A. Yakovlev *Molecular Dynamics Simulations of Laser-Induced Incandescence of Soot Using an Extended ReaxFF Reactive Force Field*. *Journal of Physical Chemistry A* (2010), 114, 12561-1257 <http://dx.doi.org/10.1021/jp1080302>
- F. Castro-Marciano, A.M. Kamat, M.F. Russo, A.C.T. van Duin, and J.P. Mathews *Combustion of an Illinois No. 6 Coal Char Simulated Using an Atomistic Char Representation and the ReaxFF Reactive Force Field*. *Combustion and Flame* (2012), 159, 23273-1285 <http://dx.doi.org/10.1016/j.combustflame.2011.10.022>
- The C/H/O parameters are the same as in the CHO forcefield, with added S/C, S/H and S/O descriptions. This force field was used in Castro et al, *Combustion and Flame* 2011
- The Boron and Nitrogen parameters are based on (but not identical to) the parameters used in Weismiller et al, *JPC-A* 2010.
- Branch: combustion.

CuCl-H2O.ff: (Cu/Cl/H/O) O. Rahaman, A.C.T. van Duin, V.S. Bryantsev, J.E. Mueller, S.D. Solares, W.A. Goddard III, and D.J. Doren *Development of a ReaxFF Reactive Force Field for Aqueous Chloride and Copper Chloride* *J. Phys. Chem. A* 114 (2010), 3556-3568 (<http://dx.doi.org/10.1021/jp9090415>)

- This forcefield is an extension of: A.C.T. van Duin, V.S. Bryantsev, M.S. Diallo, W.A. Goddard, O. Rahaman, D.J. Doren, D. Raymand, and K. Hermansson *Development and validation of a ReaxFF reactive force field for Cu cation/water interactions and copper metal/metal oxide/metal hydroxide condensed phases* *Journal of Physical Chemistry A* 2010, 114, 9507-9514 <http://dx.doi.org/10.1021/jp102272z>
- Branch: water.

FeOCHCl.ff: (Fe/O/C/H/Cl) M. Aryanpour, A.C.T. van Duin, J.D. Kubicki *Development of a Reactive Force Field for Iron-Oxyhydroxide Systems* *J. Phys. Chem. A* 2010, 114, 6298-6307 (<http://dx.doi.org/10.1021/jp101332k>)

- The Cl parameters were published by: O. Rahaman, A.C.T. van Duin, V.S. Bryantsev, J.E. Mueller, S.D. Solares, W.A. Goddard III, and D.J. Doren *Development of a ReaxFF Reactive Force Field for Aqueous Chloride and Copper Chloride* *J. Phys. Chem. A* 114 (2010), 3556-3568 <http://dx.doi.org/10.1021/jp9090415>

- The initial force field parameters for the Fe-Fe parameters were taken from an earlier force field development project on bulk-iron metal, based on DFT-calculations on antiferromagnetic BCC and FCC. The ReaxFF parameters have not been published yet, however the DFT data can be found in ref 31 of the above mentioned manuscript. The O/H parameters were taken from the ReaxFF bulk water description. The Fe/Fe and O/H parameters were kept fixed to these initial values, whereas the Fe/O parameters were reoptimized against the quantum mechanical results presented in the above mentioned manuscript.
- Detailed information on the force field parameters is given in the supporting information of the above mentioned manuscript.
- Branch: water.

HE.ff: (C/H/O/N) RDX/High Energy L.Z. Zhang, A.C.T. van Duin, S.V. Zybin, and W.A. Goddard *Thermal Decomposition of Hydrazines from Reactive Dynamics Using the ReaxFF Reactive Force Field* *Journal of Physical Chemistry B* (2009) 113, 10770-10778 (<http://dx.doi.org/10.1021/jp900194d>)

- Part of this forcefield is also published in: L.Z. Zhang, S.V. Zybin, A.C.T. van Duin, S. Dasgupta, W.A. Goddard, and E.M. Kober *Carbon Cluster Formation during Thermal Decomposition of Octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine and 1,3,5-Triamino-2,4,6-trinitrobenzene High Explosives from ReaxFF Reactive Molecular Dynamics Simulations* *Journal of Physical Chemistry A* (2009) 113, 10619-10640 <http://dx.doi.org/10.1021/jp901353a>
- The parameters of the nitramine ReaxFF are based on a large number of ab initio QM calculations. Over 40 reactions and over 1600 equilibrated molecules have been used; they are designed to characterize the atomic interactions under various environments likely and unlikely high energy each atom can encounter. The training set contains bond breaking and compression curves for all possible bonds, angle and torsion bending data for all possible cases, as well as crystal data.
- Please see the supplemental material from *Phys. Rev. Lett.* 2003, 91, 098301 for a detailed description of the parameterization of this force field.
- Branch: combustion.

HE2.ff: (C/H/O/N/S/Si) RDX/High Energy L.Z. Zhang, S.V. Zybin, A.C.T. van Duin, S. Dasgupta, W.A. Goddard, and E.M. Kober *Carbon Cluster Formation during Thermal Decomposition of Octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine and 1,3,5-Triamino-2,4,6-trinitrobenzene High Explosives from ReaxFF Reactive Molecular Dynamics Simulations* *Journal of Physical Chemistry A* (2009) 113, 10619-10640 (<http://dx.doi.org/10.1021/jp901353a>)

- Similar to HE.ff with additional parameters fitted for the TATB explosive
- Branch: combustion.

NaH.ff: (Na/H) J.G.O. Ojwang, R. Van Santen, G.J. Kramer, A.C.T. van Duin, and W.A. Goddard *Modeling the sorption dynamics of NaH using a reactive force field* *Journal of Chemical Physics* 2008, 128, 164714 (<http://dx.doi.org/10.1063/1.2908737>)

- This forcefield originally had a typo, defining the H-Na-Na angle twice. The same typo is in table 4 of the publication, but the text clearly mentions that the second line should define Na-H-Na instead.
- Branch: combustion.

NiCH.ff: (Ni/C/H) J.E. Mueller, A.C.T. van Duin, and W.A. Goddard III *Development and Validation of ReaxFF Reactive Force Field for Hydrocarbon Chemistry Catalyzed by Nickel* *J. Phys. Chem. C* 2010, 114, 4939-4949 (<http://dx.doi.org/10.1021/jp9035056>)

- Branch: combustion.

SiOH.ff: (Si/O/H) J.C. Fogarty, H.M. Aktulga, A.Y. Grama, A.C.T. van Duin, S.A. Pandit *A reactive molecular dynamics simulation of the silica-water interface* *J. Chem. Phys.* 2010, 132, 174704 (<http://dx.doi.org/10.1063/1.3407433>)

- This force field was trained to model the interaction of water at the SiO₂ surface, with specific emphasis on proton-transfer reactions. Updated parameters were fitted for all Si/O/H bond, angle, and torsion interactions as well, in addition to the dissociation of a water molecule from a single Si(OH)₄ molecule and reaction energies for the polymerization of Si(OH)₄
- Branch: water.

SiC.ff: (Si/C/O/H/N/S) D. Newsome, D. Sengupta, H. Foroutan, M.F. Russo, and A.C.T. van Duin *Oxidation of Silicon Carbide by O₂ and H₂O: A ReaxFF Reactive Molecular Dynamics Study, Part I* *Journal of Physical Chemistry* 2012 116, 16111-16121 (<http://dx.doi.org/10.1021/jp306391p>)

- The included forcefield is based on the Newsome reference, with slightly improved parameters by van Duin et al.
- Branch: combustion.

VOCH.ff: (V/O/C/H) K. Chenoweth, A.C.T. van Duin, P. Persson, M.J. Cheng, J. Oxgaard, W.A. Goddard *Development and Application of a ReaxFF Reactive Force Field for Oxidative Dehydrogenation on Vanadium Oxide Catalysts* *J. Phys. Chem. C*, 2008, 112, 14645-14654 (<http://dx.doi.org/10.1021/jp802134x>)

- The ReaxFF force field parameters have been fit to a large quantum mechanics (QM) training set containing over 700 structures and energetics related to bond dissociations, angle and dihedral distortions, and reactions between hydrocarbons and vanadium oxide clusters. In addition, the training set contains charge distributions for small vanadium oxide clusters and the stabilities of condensed-phase systems including V₂O₅, VO₂, and V₂O₃ in addition to metallic V (V₀).
- Branch: combustion.

ZnOH.ff: (Zn/O/H) D. Raymand, A.C.T. van Duin, M. Baudin, K. Hermansson *A reactive force field (ReaxFF) for zinc oxide* *Surface Science* 2008, 602, 1020-1031 (<http://dx.doi.org/10.1016/j.susc.2007.12.023>)

- updated version published by: D. Raymand, A.C.T. van Duin, D. Spangberg, W.A. Goddard, K. Hermansson *Water adsorption on stepped ZnO surfaces from MD simulation* *Surface Science* 2010, 604, 9-10, 741-752 <http://dx.doi.org/10.1016/j.susc.2009.12.012>
- Based on QM calculations for Zn(s), ZnO(s), and Zn hydroxide clusters [Zn(OH)₂ and O(ZnOH)₂], ReaxFF parameters were generated for Zn-O and Zn-Zn bond energies and for Zn-O-Zn, O-Zn-O, O-Zn-Zn and Zn-O-H valence angle energies.
- QM calculations were performed for the four crystal polymorphs of the wurtzite, zincblende, rocksalt and caesium chloride structures (the structures are also referred to as h-ZnS, c-ZnS, NaCl and CsCl, respectively).
- Branch: water.

Al-H₂O.ff: (Al/H/O) M. Russo, R. Li, M. Mench, and A.C.T. van Duin *Molecular Dynamic Simulation of Aluminum-Water Reactions Using the ReaxFF Reactive Force Field* *International Journal of Hydrogen Energy* 36 (2011), 5828-5835 (<http://dx.doi.org/10.1016/j.ijhydene.2011.02.035>)

- Branch: water.

CaSiAlO.ff: (C/H/O/Fe/Cl/Si/Al/Ca) M.C. Pitman and A.C.T. van Duin *Dynamics of Confined Reactive Water in Smectite Clay-Zeolite Composites* *J. Am. Chem. Soc.*, 2012, 134 (6), 3042-3053 (<http://dx.doi.org/10.1021/ja208894m>)

- Branch: water.

dispersion/CHONSSi-Ig.ff: (C/H/O/N/S/Si) L. Liu, Y. Liu, S.V. Zybin, H. Sun, and W.A. Goddard III *ReaxFF-Ig: Correction of the ReaxFF Reactive Force Field for London Dispersion, with Applications to the Equations of State for Energetic Materials* *The Journal of Physical Chemistry A*, 2011, 115(40), 11016-11022 (<http://dx.doi.org/10.1021/jp201599t>)

- This forcefield adds London dispersion correction terms to reaxFF, and is optimized for the energetic materials RDX, PETN, TATB, and NM plus graphite, polyethylene, solid carbon dioxide, and solid N₂, using the low temperature crystal structures to determine the lg correction parameters.
- Branch: combustion.

CHOFeAlNiCuS.ff: (C/H/O/Fe/Al/Ni/Cu/S) O. Rahaman, A.C.T. van Duin, W.A. Goddard III, and D.J. Doren *Development of a ReaxFF reactive force field for glycine and application to solvent effect and tautomerization* *Journal of Physical Chemistry B* 115 (2011), 249-261. (<http://dx.doi.org/10.1021/jp204894m>)

- C/O/H parameters only. The Cu/Fe/Al/Ni parameters are from: Y.K. Shin, H. Kwak, C. Zou, A.V. Vasenkov, and A.C.T. van Duin *Development and Validation of a ReaxFF Reactive Force Field for Fe/Al/Ni Alloys: Molecular Dynamics Study of Elastic Constants, Diffusion, and Segregation* *J. Phys. Chem. A*, 2012, 116 (49), pp 12163–12174 <http://dx.doi.org/10.1021/jp308507x>
- Not all cross-terms between the two forcefield files are defined, which might cause problems if the system has (for example) C-Cu interactions.
- Branch: water.

AuSCH_2011.ff: (Au/S/C/H) T.T. Jarvi, A.C.T. van Duin, K. Nordlund, and W.A. Goddard III, *Development of Interatomic ReaxFF Potentials for Au-S-C-H Systems* *J. Phys. Chem. A*, 115, 10315-10322 (2011) (<http://dx.doi.org/10.1021/jp201496x>)

- Branch: combustion.

AuSCH_2013.ff: (Au/S/C/H) Gyun-Tack Bae and Christine M. Aikens *Improved ReaxFF Force Field Parameters for Au-S-C-H Systems* *Journal of Physical Chemistry A* 2013 117 (40), 10438-10446 (<http://dx.doi.org/10.1021/jp405992m>)

- Based upon: T.T. Jarvi, A.C.T. van Duin, K. Nordlund, and W.A. Goddard *Development of interatomic ReaxFF potentials for Au-S-C-H systems*. *Journal of Physical Chemistry C* 115, (2011), 10315-10322.
- yields improvements for bond bending potential energy surfaces
- aims to agree with DFT geometries of small clusters and gold-thiolate nanoparticles
- Branch: combustion.

PDMSDecomp.ff: (C/H/O/Si) K. Chenoweth, S. Cheung, A.C.T. van Duin, W.A. Goddard III, and E.M. Kober *Simulations on the Thermal Decomposition of a Poly(dimethylsiloxane) Polymer Using the ReaxFF Reactive Force Field* *J. Am. Chem. Soc.*, 2005, 127 (19), pp 7192-7202 (<http://dx.doi.org/10.1021/ja050980t>)

- Specialized forcefield, designed to “investigate the failure of the poly(dimethylsiloxane) polymer (PDMS) at high temperatures and pressures and in the presence of various additives”
- Line from the torsion block was referring to non-existent atoms from the atomic block and thus was removed.
- Branch: combustion.

TiOCHNCl.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F) S.Y. Kim, A.C.T. van Duin, and J.D. Kubicki *Molecular dynamics simulations of the interactions between TiO₂ nanoparticles and water with Na⁺ and Cl⁻, methanol, and formic acid using a reactive force field* *Journal of Materials Research / Volume 28 / Issue 03 / 2013*, pp 513-520 (<http://dx.doi.org/10.1557/jmr.2012.367>)

- used for simulating TiO₂(both rutile and anatase) nanoparticles with water, methanol, and formic acid
- The force field was validated by comparing water dissociative adsorption percentage and bond length between Na-O with density functional theory (DFT) and experimental results
- Branch: water.

PtCH.ff: (C/H/Pt) C.F. Sanz-Navarro, P. Astrand, De Chen, M. Ronning, A.C.T. van Duin, T. Jacob, and W.A. Goddard III *Molecular Dynamics Simulations of the Interactions between Platinum Clusters and Carbon Platelets* J. Phys. Chem. A 112, 1392-1402 (2008) (<http://dx.doi.org/10.1021/jp074806y>)

- Branch: combustion.

BaYZrCHO.ff: (C/H/O/Ba/Zr/Y) A.C.T. van Duin, B.V. Merinov, S.S. Jang, and W.A. Goddard III *ReaxFF Reactive Force Field for Solid Oxide Fuel Cell Systems with Application to Oxygen Ion Transport in Yttria-Stabilized Zirconia* J. Phys. Chem. A, 112, 3133-3140 (2008) (<http://dx.doi.org/10.1021/jp076775c>)

- Branch: combustion.

CHONSSiPtZrNiCuCo.ff: (C/H/O/N/S/Si/Pt/Zr/Ni/Cu/Co) K.D. Nielson, A.C.T. van Duin, J. Oxgaard, W.Q. Deng, and W.A. Goddard III *Development of the ReaxFF Reactive Force Field for Describing Transition Metal Catalyzed Reactions, with Application to the Initial Stages of the Catalytic Formation of Carbon Nanotubes* J. Phys. Chem. A, 109, 493-499 (2005) (<http://dx.doi.org/10.1021/jp046244d>)

- Branch: combustion.

Glycine.ff: (C/H/O/N) O. Rahaman, A.C.T. van Duin, W.A. Goddard III, and D.J. Doren, *Development of a ReaxFF Reactive Force Field for Glycine and Application to Solvent Effect and Tautomerization* J. Phys. Chem. B, 115, 249-261 (2011) (<http://dx.doi.org/10.1021/jp108642r>)

- Line from the valence angle block was referring to non-existent atoms from the atomic block and thus was removed.
- Branch: water.

SiONH.ff: (C/H/O/N/Si/S) A.D. Kulkarni, D.G. Truhlar, S.G. Srinivasan, A.C.T. van Duin, P. Norman, and T.E. Schwartzentruber *Oxygen Interactions with Silica Surfaces: Coupled Cluster and Density Functional Investigation and the Development of a New ReaxFF Potential* J. Phys. Chem. C, 2013, 117 (1), pp 258-269 (<http://dx.doi.org/10.1021/jp3086649>)

- Aimed at oxygen interactions with realistic silica surfaces
- Lines from the valence angle block was referring to non-existent atoms from the atomic block and thus was removed.
- Branch: combustion.

CHOFe.ff: (C/H/O/Fe/Cl/Si/Al) Chenyu Zou, A.C.T. Van Duin *Investigation of Complex Iron Surface Catalytic Chemistry Using the ReaxFF Reactive Force Field Method* JOM, December 2012, Volume 64, Issue 12, pp 1426-1437 (<http://dx.doi.org/10.1007/s11837-012-0463-5>)

- only the parameters for Fe (and crossterms) differ from the CHOAlSi.ff forcefield
- Branch: water.

CHOAlSi.ff: (C/H/O/Fe/Cl/Si/Al) F. Castro-Marcanao, A.C.T. van Duin *Comparison of thermal and catalytic cracking of 1-heptene from ReaxFF reactive molecular dynamics simulations* Combustion and Flame, Volume 160, Issue 4, April 2013, Pages 766-775 (<http://dx.doi.org/10.1016/j.combustflame.2012.12.007>)

- only the parameters for Fe (and crossterms) differ from the CHOFe.ff forcefield
- Branch: water.

CHOLi.ff: (C/H/O/N/S/Mg/P/Na/Li) D. Bedrov, G.D. Smith, A.C.T. van Duin *Reactions of Singly-Reduced Ethylene Carbonate in Lithium Battery Electrolytes: A Molecular Dynamics Simulation Study Using the ReaxFF* Journal of Physical Chemistry A, 2012, 116 (11), pp 2978-2985 (<http://dx.doi.org/10.1021/jp210345b>)

- specifically generated for simulating Lithium battery electrolytes
- must be used in combination with the MOLCHARGE keyword to set a charge restraint on Li and CO3!
- Branch: water.

SiOAlLi.ff: (H/O/Si/Al/Li) B. Narayanan, A.C.T. van Duin, B.B. Kappes, I.E. Reimanis and C.V. Ciobanu *A reactive force field for lithium-aluminum silicates with applications to eucryptite phases* *Modelling and Simulation in Materials Science and Engineering* 2012 20 015002 (<http://dx.doi.org/10.1088/0965-0393/20/1/015002>)

- Branch: water.

PdO.ff: (Pd/O) T.P. Senftle, R.J. Meyer, M.J. Janik and A.C.T. van Duin *Development of a ReaxFF potential for Pd/O and application to palladium oxide formation* *The Journal of Chemical Physics* 139, 044109 (2013) (<http://dx.doi.org/10.1063/1.4815820>)

- used for studying Oxidation states of Pd nanoparticles, surfaces and bulk configurations with a GCMC method
- Branch: combustion.

PdH.ff: (Pd/H) T.P. Senftle, M.J. Janik and A.C.T. van Duin *A ReaxFF Investigation of Hydride Formation in Palladium Nanoclusters via Monte Carlo and Molecular Dynamics Simulations* *The Journal of Physical Chemistry C*, 2014, 118 (9), pp 4967-4981 (<http://dx.doi.org/10.1021/jp411015a>)

- used in combination with a GCMC method
- Branch: combustion.

Co.ff: (Co) Xue-Qing Zhang, E. Iype, S.V. Nedeia, A.P.J. Jansen, B.M. Szyja, E.J.M. Hensen, and R.A. van Santen *Site Stability on Cobalt Nanoparticles: A Molecular Dynamics ReaxFF Reactive Force Field Study* *The Journal of Physical Chemistry C*, 2014, 118 (13), pp 6882-6886 (<http://dx.doi.org/10.1021/jp500053u>)

- forcefield was generated using a recently developed Monte Carlo algorithm with simulated annealing.
- Branch: combustion.

CHONSMgPNaCuCl.ff: (C/H/O/N/S/Mg/P/Na/Cu/Cl) Susanna Monti, Cui Li, and Vincenzo Carravetta *Reactive Dynamics Simulation of Monolayer and Multilayer Adsorption of Glycine on Cu(110)* *J. Phys. Chem. C*, 2013, 117 (10), pp 5221-5228 (<http://dx.doi.org/10.1021/jp312828d>)

- Reactive MD-force field for amino acids on copper
- Branch: water.

CHOSMoNiLiBFPN.ff: (C/H/O/S/Mo/Ni/Li/B/F/P/N) Md M. Islam, V.S. Bryantsev, A.C.T. van Duin *ReaxFF Reactive Force Field Simulations on the Influence of Teflon on Electrolyte Decomposition during Li/SWCNT Anode Discharge in Lithium-Sulfur Batteries* *J. Electrochem. Soc.* 2014 volume 161, issue 8, E3009-E3014 (<http://dx.doi.org/10.1149/2.005408jes>)

- forcefield for Electrochemistry in Li-S batteries
- Branch: combustion.

CHONSSiNaFZr.ff: (C/H/O/N/S/Si/Na/F/Zr) A. Rahnamoun and A.C.T. van Duin *Reactive Molecular Dynamics Simulation on the Disintegration of Kapton, POSS Polyimide, Amorphous Silica, and Teflon during Atomic Oxygen Impact Using the Reaxff Reactive Force-Field Method* *J. Phys. Chem. A*, 2014, 118 (15), pp 2780-2787 (<http://dx.doi.org/10.1021/jp4121029>)

- comments in the forcefield file: interactions with water and Na+ Fogarty et al. JCP-2010 ; with glycine + C/H/F parameters; Si-F bond/offdiag/angle parameters; Si-S dummy parameters + S-O-H parameters(Yun 2012 Oct8) + H-F bond/offdiag (Jan14 2013 Joon) Jan31: added Zr/O/H/C
- Branch: water.

TiClOH.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F) Sung-Yup Kim and A.C.T. van Duin *Simulation of Titanium Metal/Titanium Dioxide Etching with Chlorine and Hydrogen Chloride Gases Using the ReaxFF Reactive Force Field* *J. Phys. Chem. A*, 2013, 117 (27), pp 5655-5663 (<http://dx.doi.org/10.1021/jp4031943>)

- adaptation/evolution of the TiOCHNCl.ff forcefield by Kim, S.-Y. et al.

- Branch: water.

CHONSSiNaAl.ff: (C/H/O/N/S/Si/Na/Al) C. Bai, L. Liu, and H. Sun *Molecular Dynamics Simulations of Methanol to Olefin Reactions in HZSM-5 Zeolite Using a ReaxFF Force Field* *J. Phys. Chem. C*, 2012, 116 (12), pp 7029-7039 (<http://dx.doi.org/10.1021/jp300221j>)

- used to simulate methanol to olefin (MTO) reactions in H-ZSM-5 zeolite
- Branch: water.

undocumented/NiCH.ff: (Ni/C/H/O/N/S/F/Pt/Cl) Unknown *No Name* Not published (<http://>)

- This forcefield is from april 2009, and is not published
- Branch: combustion.

LiS.ff: (Li/S) Md M. Islam, A. Ostadhossein, O. Borodin, A.T. Yeates, W.W. Tipton, R.G. Hennig, N. Kumar, and A.C.T. van Duin *ReaxFF molecular dynamics simulations on lithiated sulfur cathode materials* *Phys. Chem. Chem. Phys.*, 2015, 17, 3383-3393 (<http://dx.doi.org/10.1039/C4CP04532G>)

- Developed for and used to study Sulfur cathode behavior in Li battery cells
- Branch: combustion.

CHONSSiPtNiCuCoZrYBa.ff: (C/H/O/N/S/Si/Pt/Ni/Cu/Co/Zr/Y/Ba) B.V. Merinov, J.E. Mueller, A.C.T. van Duin, Qi An, and W.A. Goddard III *ReaxFF Reactive Force-Field Modeling of the Triple-Phase Boundary in a Solid Oxide Fuel Cell* *J. Phys. Chem. Lett.*, 2014, 5 (22), pp 4039-4043 (<http://dx.doi.org/10.1021/jz501891y>)

- Force field was developed by combining the YSZ and Ni/C/H descriptions
- From the summary: “The products obtained in our simulations are the same as those in experiment, which indicates that the developed ReaxFF potential properly describes complex physicochemical processes, such as the oxide-ion diffusion, fuel conversion, water formation reaction, coking, and delamination, occurring at the TPB and can be recommended for further computational studies of the fuel/electrode/electrolyte interfaces in a SOFC”
- The Aa atom type is a rename of the Zr atom type, which was defined twice.
- Branch: combustion.

CHONSSiNaP.ff: (C/H/O/N/S/Si/Na/P) Bo Zhang, A.C.T. van Duin and J.K. Johnson *Development of a ReaxFF Reactive Force Field for Tetrabutylphosphonium Glycinate/CO₂ Mixtures* *J. Phys. Chem. B*, 2014, 118 (41), pp 12008-12016 (<http://dx.doi.org/10.1021/jp5054277>)

- The forcefield file mentions “Gly with dummy N-P parameters”
- It was developed for studying Carbon dioxide interactions with the ionic liquid tetrabutylphosphonium glycinate, both physical and chemical absorption.
- Several lines from valence angle block were referring to non-existent atoms from the atomic block and thus were removed.
- Branch: combustion.

CHOLi_2.ff: (C/H/O/Li) M. Raju, P. Ganesh, P.R.C. Kent, and A.C.T. van Duin *Reactive Force Field Study of Li/C Systems for Electrical Energy Storage* *J. Chem. Theory Comput.*, 2015, 11 (5), pp 2156-2166 (<http://dx.doi.org/10.1021/ct501027v>)

- Used for studying Li/C systems with GCMC and MD
- Also contains parameters for “Vi” and “GI”, but these virtual elements are not mentioned in the paper
- Parameters were fitted using a training set consisting, amongst others, of Li binding energies on pristine graphene and graphite, Li migration pathways in graphite and Li dissociation pathways in different hydrocarbons

- The forcefield was validated by a side-by-side comparison of ReaxFF and DFT energies for Li binding on a divacancy, and ReaxFF and DFT ground-state configurations for stage II and stage I stacking in graphite obtained using GCMC simulations
- The forcefield reproduces the in-plane Li ordering as well as the graphite stacking sequence for stage II and stage I compounds
- Branch: water.

CHONSiCaCsKSrNaMgAlCu.ff: (C/H/O/N/S/Si/Ca/Cs/K/Sr/Na/Mg/Al/Cu) G.M. Psfogiannakis, J.F. McCleerey, E. Jaramillo and A.C.T. van Duin *ReaxFF Reactive Molecular Dynamics Simulation of the Hydration of Cu-SSZ-13 Zeolite and the Formation of Cu Dimers* *J. Phys. Chem. C*, 2015, 119 (12), pp 6678-6686 (<http://dx.doi.org/10.1021/acs.jpcc.5b00699>)

- This Cu/Si/Al/O/H forcefield was developed for and used in MD simulations of the hydration of Cu-exchanged SSZ-13 catalyst.
- The force field was developed by merging and expanding previously tested and published ReaxFF force fields for Si/Al/O/H systems and Cu/O/H systems (citations 19-26 of the publication)
- This forcefield is an extension of K.L. Joshi et al 2014, <http://dx.doi.org/10.1039/C4CP02612H>, all non-Cu parameters are the same.
- Branch: water.

HOSMg.ff: (H/O/S/Mg) E. Iype, M. Hütter, A.P.J. Jansen, S.V. Nedeia and C.C.M. Rindt *Parameterization of a reactive force field using a Monte Carlo algorithm* *J. Comput. Chem.*, 2013, 34: 1143-1154 (<http://dx.doi.org/10.1002/jcc.23246>)

- This forcefield is optimized with a metropolis Monte-Carlo algorithm with simulated annealing to search for the optimum parameters for the ReaxFF force field in a high- dimensional parameter space.
- The optimization is done against a set of quantum chemical data for MgSO₄ hydrates.
- Branch: combustion.

CHONSMgPNaCuCl_v2.ff: (C/H/O/N/S/Mg/P/Na/Cu/Cl) S. Monti, A. Corozzi, P. Fristrup, K.L. Joshi, Yun Kyung Shin, P. Oelschlaeger, A.C.T. van Duin and V. Baronee *Exploring the conformational and reactive dynamics of biomolecules in solution using an extended version of the glycine reactive force field* *Phys. Chem. Chem. Phys.*, 2013,15, 15062-15077 (<http://dx.doi.org/10.1039/C3CP51931G>)

- This forcefield is an extension of CHONSMgPNaCuCl.ff by S. Monti et al., *J. Phys. Chem. C*, 2013, 117 (10), pp 5221-5228, <http://dx.doi.org/10.1021/jp312828d>
- Developed for peptide and protein simulations
- Branch: water.

OPt.ff: (O/Pt) D. Fantauzzi, J. Bandlow, L. Sabo, J.E. Mueller, A.C.T. van Duin and T. Jacob *Development of a ReaxFF potential for Pt-O systems describing the energetics and dynamics of Pt-oxide formation* *Phys. Chem. Chem. Phys.*, 2014,16, 23118-23133 (<http://dx.doi.org/10.1039/C4CP03111C>)

- Pt-Pt parameters for bulk platinum phases, low & high-index platinum surfaces and nanoclusters.
- O-Pt parameters for bulk platinum oxides, as well as oxygen adsorption and oxide formation on Pt(111) terraces and the {111} and {100} steps connecting them.
- Branch: combustion.

CHONSMgPNaTiClF.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F) Stijn Huygh, Annemie Bogaerts, Adri C.T. van Duin and Erik C. Neyts *Development of a ReaxFF reactive force field for intrinsic point defects in titanium dioxide* *Computational Materials Science*, Volume 95, December 2014, Pages 579-591, ISSN 0927-0256 (<http://dx.doi.org/10.1016/j.commatsci.2014.07.056>)

- Forcefield developed for studying the influence of intrinsic point defects on the chemistry with TiO₂ condensed phases.
- Forcefield seems to be based on TiOCHNCl.ff, <http://dx.doi.org/10.1557/jmr.2012.367>
- Branch: water.

LiSi.ff: (Li/Si) A. Ostadhosseini, E.D. Cubuk, G.A. Tritsarlis, E. Kaxiras, S. Zhanga and A.C.T. van Duin *Stress effects on the initial lithiation of crystalline silicon nanowires: reactive molecular dynamics simulations using ReaxFF* Phys. Chem. Chem. Phys., 2015, 17, 3832-3840 (<http://dx.doi.org/10.1039/C4CP05198J>)

- Branch: combustion.

CHOFeAlNiCuSCr.ff: (C/H/O/Fe/Al/Ni/Cu/S/Cr) Y.K. Shin, H. Kwak, A.V. Vasenkov, D. Sengupta and A.C.T. van Duin *Development of a ReaxFF Reactive Force Field for Fe/Cr/O/S and Application to Oxidation of Butane over a Pyrite-Covered Cr₂O₃ Catalyst* ACS Catalysis, 2015, 5 (12), pp 7226-7236 (<http://dx.doi.org/10.1021/acscatal.5b01766>)

- Forcefield optimized for Fe/Cr/O/S
- Branch: water.

CHOFeAlNiCuSCr_v2.ff: (C/H/O/Fe/Al/Ni/Cu/S/Cr) Ni-O-vacancy C. Zou et al. *Molecular dynamics simulations of the effects of vacancies on nickel self-diffusion, oxygen diffusion and oxidation initiation in nickel, using the ReaxFF reactive force field* Acta Materialia, Volume 83, 15 January 2015, Pages 102-112 (<http://dx.doi.org/10.1016/j.actamat.2014.09.047>)

- Forcefield optimized for Ni/O, trained with a QM data on Ni, NiO and vacancies
- The non-carbon parameters are the same as in CHOFeAlNiCuSCr_v3.ff
- Branch: water.

CHOFeAlNiCuSCr_v3.ff: (C/H/O/Fe/Al/Ni/Cu/S/Cr) F. Tavazza, T.P. Senftle, C. Zou, C.A. Becker and A.C.T. van Duin *Molecular Dynamics Investigation of the Effects of Tip-Substrate Interactions during Nanoindentation* J. Phys. Chem. C, 2015, 119 (24), pp 13580-13589 (<http://dx.doi.org/10.1021/acs.jpcc.5b01275>)

- Combination of the C.ff (condensed carbon) forcefield with Ni/C/O/H parameters
- The non-carbon parameters are the same as in CHOFeAlNiCuSCr_v2.ff
- Branch: water.

C.ff: (C) C-2013 S.G. Srinivasan, A.C.T. van Duin and P. Ganesh *Development of a ReaxFF Potential for Carbon Condensed Phases and Its Application to the Thermal Fragmentation of a Large Fullerene* J. Phys. Chem. A, 2015, 119 (4), pp 571-580 (<http://dx.doi.org/10.1021/jp510274e>)

- Forcefield designed for modeling Carbon condensed phases
- Branch: combustion.

CHONSSiGe.ff: (C/H/O/N/S/Si/Ge) G. Psogiannakis and A.C.T. van Duin *Development of a ReaxFF reactive force field for Si/Ge/H systems and application to atomic hydrogen bombardment of Si, Ge, and SiGe (100) surfaces* Surface Science 2016, Vol. 646, pp 253-260 (<http://dx.doi.org/10.1016/j.susc.2015.08.019>)

- Forcefield designed for bombardment of Si, Ge and SiGe surfaces with atomic hydrogen.
- Branch: combustion.

CHONSFPtCINi.ff: (C/H/O/N/S/F/Pt/Cl/Ni) Water-Pt-Ni-Nafion D. Fantauzzi, J.E. Mueller, L. Sabo, A.C.T. van Duin and T. Jacob *Surface Buckling and Subsurface Oxygen: Atomistic Insights into the Surface Oxidation of Pt(111)* ChemPhysChem, 2015, Vol.16: 2797-2802 (<http://dx.doi.org/10.1002/cphc.201500527>)

- extension of the OPt.ff forcefield parameters
- Branch: water.

CHONSSiPtZrNiCuCoHeNeArKrXe.ff: (C/H/O/N/S/Si/Pt/Zr/Ni/Cu/Co/He/Ne/Ar/Kr/Xe) A.M. Kamat, A.C.T. van Duin and A. Yakovlev *Molecular Dynamics Simulations of Laser-Induced Incandescence of Soot Using an Extended ReaxFF Reactive Force Field* *J. Phys. Chem. A*, 2010, 114 (48), pp 12561-12572 (<http://dx.doi.org/10.1021/jp1080302>)

- forcefield designed for the study of Laser-induced incandescence of soot
- Branch: combustion.

CHOSFCIN.ff: (C/H/O/S/F/Cl/N) M.A. Wood, A.C.T. van Duin and A. Strachan *Coupled Thermal and Electromagnetic Induced Decomposition in the Molecular Explosive alpha-HMX; A Reactive Molecular Dynamics Study* *J. Phys. Chem. A*, 2014, 118 (5), pp 885-895 (<http://dx.doi.org/10.1021/jp406248m>)

- forcefield designed for studying the combustion of the high-energy material a-HMX
- Branch: combustion.

Mue2016.ff: (C/H/O/S) Julian Mueller and Bernd Hartke *ReaxFF Reactive Force Field for Disulfide Mechanochemistry, Fitted to Multireference ab Initio Data;* *J. Chem. Theory Comput.*, 2016, xxx, xxx (<http://dx.doi.org/10.1021/acs.jctc.6b00461>)

- forcefield for studying S-S bond ruptures in mechanophores upon mechanical stress. Ambient conditions, both in gas phase and toluene solvent.
- Branch: combustion.

CBN.ff: (C/H/B/N) Sung Jin Pai, Byung Chul Yea and Sang Soo Han *Reactive force field for the improved design of liquid CBN hydrogen storage materials* *Phys. Chem. Chem. Phys.*, 2016, 18, pp 1818-1827 (<http://dx.doi.org/10.1039/C5CP05486A>)

- forcefield for studying liquid CBN (carbon-boron-nitrogen) hydrogen-storage materials.
- Branch: combustion.

AgZnO.ff: (C/H/O/N/Si/Cu/Ag/Zn) A. Lloyd, D. Cornil, A.C.T. van Duin, D. van Duin, R. Smith, S.D. Kenny, J. Cornil and D. Beljonne *Development of a ReaxFF potential for Ag/Zn/O and application to Ag deposition on ZnO* *Surface Science*, 2016, 645, pp 67-73 (<http://dx.doi.org/10.1016/j.susc.2015.11.009>)

- ReaxFF potential for Ag/Zn/O used to study Ag deposition on ZnO.
- Branch: water.

AlCHO.ff: (Al/C/H/O) Sungwook Hong and Adri C. T. van Duin *Atomistic-Scale Analysis of Carbon Coating and Its Effect on the Oxidation of Aluminum Nanoparticles by ReaxFF-Molecular Dynamics Simulations* *J. Phys. Chem. C*, 2016, 120 (17), pp 9464-9474 (<http://dx.doi.org/10.1021/acs.jpcc.6b00786>)

- ReaxFF potential for Al/C interactions. Used to investigate carbon coating and its effect on the oxidation of aluminum nanoparticles (ANPs)
- Branch: water.

CHNa.ff: (C/H/Na) E. Hjertenaes, A.Q. Nguyen and H. Koch *A ReaxFF force field for sodium intrusion in graphitic cathodes* *Phys. Chem. Chem. Phys.*, 2016, 18, pp 31431-31440 (<http://dx.doi.org/10.1039/C6CP06774C>)

- The force field is applied in hybrid grand canonical Monte Carlo-molecular dynamics (GC-MC/MD) simulations of model systems representative of sodium intrusion in graphitic carbon cathodes used in aluminium electrolysis.
- Branch: combustion.

CuBTC.ff: (C/H/O/N/S/Mg/P/Na/Cu) L. Huang, T. Bandosz, K. L. Joshi, A. C. T. van Duin and K. E. Gubbins *Reactive adsorption of ammonia and ammonia/water on CuBTC metal-organic framework: A ReaxFF molecular dynamics simulation* *J. Chem. Phys.*, 2013, 138, 034102 (<http://dx.doi.org/10.1063/1.4774332>)

- The force field was used to study reactive adsorption of NH₃ on the dehydrated CuBTC metal-organic framework.
- Branch: water.

CHONSMgPNaTiClFAu.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F/Au) Susanna Monti, Vincenzo Carravetta, and Hans Ågren *Simulation of Gold Functionalization with Cysteine by Reactive Molecular Dynamics* *J. Phys. Chem. Lett.*, 2016, 7 (2), pp 272-276 (<http://dx.doi.org/10.1021/acs.jpcclett.5b02769>)

- The force field was designed to study gold-protein interactions in water.

HOSiAILi.ff: (H/O/Si/Al/Li) Alireza Ostadhossein, Sung-Yup Kim, Ekin D. Cubuk, Yue Qi, and Adri C. T. van Duin *Atomic Insight into the Lithium Storage and Diffusion Mechanism of SiO₂/Al₂O₃ Electrodes of Lithium Ion Batteries: ReaxFF Reactive Force Field Modeling* *J. Phys. Chem. A*, 2016, 120 (13), pp 2114-2127 (<http://dx.doi.org/10.1021/acs.jpca.5b11908>)

- Developed for studying the energetics and kinetics of lithiation, as well as Li transportation within the crystalline/amorphous silica and alumina phases.

CHArHeNeKr.ff: (C/H/Ar/He/Ne/Kr) Kichul Yoon, A. Rahnamoun, J.L. Swett, V. Iberi, D.A. Cullen, I.V. Vlas-siouk, A. Belianinov, S. Jesse, X. Sang, O.S. Ovchinnikova, A.J. Rondinone, R.R. Unocic, and A.C.T. van Duin *Atomistic-Scale Simulations of Defect Formation in Graphene under Noble Gas Ion Irradiation* *ACS Nano*, 2016, 10 (9), pp 8376-8384 (<http://dx.doi.org/10.1021/acs.nano.6b03036>)

- Developed for studying noble gas ion irradiation of graphene and the subsequent effects of annealing. Lattice defects including nanopores were generated.

CHO-radiation.ff: (C/H/O) Roger Smith, K. Jolley, C. Latham, M. Heggie, A van Duin, D. van Duin, Houzheng Wu *A ReaxFF carbon potential for radiation damage studies* *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, Volume 393, 15 February 2017, Pages 49-53, (<http://dx.doi.org/10.1016/j.nimb.2016.11.007>)

- Developed forcefield reproduces the formation energies of many of the defects predicted by the ab initio calculations of energetic impacts and collision cascades in graphite.
- Forcefield reproduces the formation energies of many of the defects predicted by the ab initio calculations and the energy pathways between different defect states, which are important for investigating long term defect evolution.

HOTiPd.ff: (H/O/Ti/Pd) Rafik Addou, T.P. Senftle, N. O'Connor, M.J. Janik, A.C.T. van Duin, and M. Batzill *Influence of Hydroxyls on Pd Atom Mobility and Clustering on Rutile TiO₂(011)-2 x 1* *ACS Nano*, 2014, 8 (6), pp 6321-6333 (<http://dx.doi.org/10.1021/nn501817w>)

- Developed for MC simulations of Pd on TiO₂ surfaces.

CHONSMgPNaFBLi-e.ff: (C/H/O/N/S/Mg/P/Na/F/B/Li/El/Ho) M. Islam and A.C.T van Duin *Reductive Decomposition Reactions of Ethylene Carbonate by Explicit Electron Transfer from Lithium: An eReaxFF Molecular Dynamics Study* *J. Phys. Chem. C*, 2016, 120 (48), pp 27128-27134 (<http://dx.doi.org/10.1021/acs.jpcc.6b08688>)

- This forcefield uses the ACKS2 charge model
- This forcefield is an eReaxFF forcefield
- Developed for the study of lithium-ion batteries
- NOTE: This ff uses vpar(38) for e-reaxff taper radius instead of vpar(41) that ADF-reaxff expects. This will generate a warning but should still work.

CHOFeAlNiCuScrSiGe.ff: (C/H/O/Fe/Al/Ni/Cu/S/Cr/Si/Ge) Y. Zheng, S. Hong, G. Psfogiannakis, S. Datta, B. Rayner, A.C.T. van Duin and R. Engel-Herbert *Modeling and In-situ Probing of Surface Reactions in Atomic Layer Deposition* *ACS Appl. Mater. Interfaces*, 2017, 9 (18), pp 15848-15856 (<http://dx.doi.org/10.1021/acsami.7b01618>)

- Used for studying the ALD process of Al₂O₃ from trimethylaluminum and water on hydrogenated and oxidized Ge(100) surfaces

Water2017.ff: (H/O/X) W. Zhang, A.C.T. van Duin *Second-Generation ReaxFF Water Force Field: Improvements in the Description of Water Density and OH-Anion Diffusion* *J. Phys. Chem. B*, 2017, 121 (24), pp 6021-6032 (<http://dx.doi.org/10.1021/acs.jpcc.7b02548>)

- Better description of liquid water

HSMo.ff: (H/S/Mo) A. Ostadhosseini, A. Rahnamoun, Y. Wang, P. Zhao, S. Zhang, V.H. Crespi, and A.C.T. van Duin *ReaxFF Reactive Force-Field Study of Molybdenum Disulfide (MoS₂)* *J. Phys. Chem. Lett.*, 2017, 8 (3), pp 631-640 (<http://dx.doi.org/10.1021/acs.jpclett.6b02902>)

- MoS₂ training set in supporting info
- strain-stress analysis

CHON2017_weak.ff: (C/H/O/N/S/Mg/P/Na/Cu/Cl/X) Weiwei Zhang and Adri C. T. van Duin *Improvement of the ReaxFF Description for Functionalized Hydrocarbon/Water Weak Interactions in the Condensed Phase* *J. Phys. Chem. B*, 2018, 122, 4083-4092 (<http://dx.doi.org/10.1021/acs.jpcc.8b01127>)

- retraining of the protein-2013 force-field (C, H, O, and N parameters for weak interactions)
- reproduces well the density of liquid water and hydrocarbons

CaSiOH.ff: (C/H/O/Ca/Si/X) Hegoi Manzano, Roland J. M. Pellenq, Franz-Josef Ulm, Markus J Buehler, and A.C.T. van Duin *Hydration of Calcium Oxide Surface Predicted by Reactive Force Field Molecular Dynamics* *Langmuir*, 2012, 28 (9), 4187-4197 (<http://dx.doi.org/10.1021/la204338m>)

- hydration of calcium oxide surface
- fitted using density functional theory calculations on gas phase calcium-water clusters, calcium oxide bulk and surface properties, calcium hydroxide, bcc and fcc Ca, and proton transfer reactions in the presence of calcium.

CHO-2016.ff: (C/H/O) Chowdhury Ashraf and Adri C.T. van Duin* *Extension of the ReaxFF Combustion Force Field toward Syngas Combustion and Initial Oxidation Kinetics* *J. Phys. Chem. A* 121, 5, 1051-1068 (<https://dx.doi.org/10.1021/acs.jpca.6b12429>)

- improved description of oxidation of small hydrocarbons and syngas reaction

CHON2017_weak_bb.ff: (C/H/O/N/S/Mg/P/Na/Cu/Cl/X) Vashisth A., Ashraf C., Zhang W., Bakis C.E., van Duin A.C.T. *Accelerated ReaxFF Simulations for Describing the Reactive Cross-Linking of Polymers* *J Phys Chem A*. 2018 Aug 1. (<https://doi.org/10.1021/acs.jpca.8b03826>)

- reparametrized version of CHON2017_weak.ff for bond-boost application

CH_aromatics.ff: (C/H) Qian Mao, Yihua Ren, K. H. Luo, Adri C. T. van Duin *Dynamics and kinetics of reversible homo-molecular dimerization of polycyclic aromatic hydrocarbons* *The Journal of Chemical Physics* 147, 244305 (2017) (<https://doi.org/10.1063/1.5000534>)

- ff for polycyclic aromatic hydrocarbons

CuSCH.ff: (C/H/O/S/Cu/Cl/X) Jejoon Yeon, H.L. Adams, C.E. Junkermeier, A.C.T. van Duin, W.T. Tysoe, and A. Martini *Development of a ReaxFF Force Field for Cu/S/C/H and Reactive MD Simulations of Methyl Thiolate Decomposition on Cu (100)* *J. Phys. Chem. B*, 2018, 122 (2), pp 888-896 (<https://doi.org/10.1021/acs.jpcc.7b06976>)

- combination of CuCl-H₂O.ff and AuSCH_2011.ff
- reparametrized Cu-S parameters with copper sulfides data

TiO₂bio.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F/X) Susanna Monti, Mariachiara Pastore, Cui Li, Filippo De Angelis, and Vincenzo Carravetta *Theoretical Investigation of Adsorption, Dynamics, Self-Aggregation, and Spectroscopic*

Properties of the D102 Indoline Dye on an Anatase (101) Substrate J. Phys. Chem. C, 2016, 120 (5), pp 2787–2796 (<https://doi.org/10.1021/acs.jpcc.5b11332>)

CHFe.ff: (C/H/Fe) Md Mahbulul Islam, Chenyu Zou, Adri C. T. van Duin and Sumathy Raman *Interactions of hydrogen with the iron and iron carbide interfaces: a ReaxFF molecular dynamics study* Phys. Chem. Chem. Phys., 2016, 18, 761-771 (<https://doi.org/10.1039/C5CP06108C>)

CHGe.ff: (C/H/O/Ge) Nadire Nayir, Adri C.T. van Duin and Sakir Erkoç *Development of a ReaxFF Reactive Force Field for Interstitial Oxygen in Germanium and Its Application to GeO₂/Ge Interfaces* J. Phys. Chem. C, 2019, 123 (2), pp 1208–1218 (<http://dx.doi.org/10.1021/acs.jpcc.8b08862>)

- Created by extending the training set from CHOFeAlNiCuSCrSiGe.ff by Zheng et al. with additional crystal data
- Some general parameters were missing in the Supporting Information

CHONSSi.ff: (C/H/O/N/S/Si) F.A. Soria, W. Zhang, P.A. Paredes-Olivera, A.C.T. van Duin and E.M. Patrito *Si/C/H ReaxFF Reactive Potential for Silicon Surfaces Grafted with Organic Molecules* J. Phys. Chem. C, 2018, 122 (41), pp 23515-23527 (<http://dx.doi.org/10.1021/acs.jpcc.8b07075>)

- developed for the study of the functionalization and decomposition of alkyl monolayers on silicon surface

CHOSiNa.ff: (C/H/O/Si/Na) S.H. Hahn, J. Rimsza, L. Criscenti, Wei Sun, Lu Deng, Jincheng Du, Tao Liang, S.B. Sinnott, and A.C.T. van Duin *Development of a ReaxFF Reactive Force Field for NaSiO_x/Water Systems and Its Application to Sodium and Proton Self-Diffusion* J. Phys. Chem. C, 2018, 122 (34), pp 19613-19624 (<https://dx.doi.org/10.1021/acs.jpcc.8b05852>)

- developed for reactive MD simulation of the sodium silicate-water interfaces
- validated for sodium silicate crystal structures and glasses, and transport properties of sodium ions and protons within the amorphous structures

CHOCsKNaClIFLi.ff: (C/H/O/Cs/K/Na/Cl/I/F/Li) M.V. Fedkin, Y.K. Shin, N. Dasgupta, J. Yeon, W. Zhang, D. van Duin, A.C.T. van Duin, K. Mori, A. Fujiwara, M. Machida, H. Nakamura, and M. Okumura *Development of the ReaxFF Methodology for Electrolyte-Water Systems* J. Phys. Chem. A, 2019, 123 (10), pp 2125-2141 (<http://dx.doi.org/10.1021/acs.jpca.8b10453>)

- developed for water-electrolyte systems with Li⁺, Na⁺, K⁺, Cs⁺, F⁻, Cl⁻, and I⁻
- trained against (QM) calculations related to water binding energies, hydration energies and energies of proton transfer

INCLUDED FORCEFIELDS (DEVELOPMENT)

See also [Included Forcefields \(released version\)](#)

Description of ReaxFF force fields

Disclaimer: Using these forcefields for systems they have not been explicitly trained against may produce unrealistic results. Please see the full manuscripts for more detailed information.

The force field files used by the SCM version of ReaxFF are compatible with those used by the original ReaxFF code. So if you have the force field information from somewhere else you can just use it (save it in a text file, and select it in AMSinput via the Other... option).

There are currently two major groupings (i.e., the ReaxFF branches) of parameter sets that are intra-transferable with one another: (1) the combustion branch and (2) the aqueous (water) branch. The major difference between these two branches is in the O/H parameters, where the combustion branch focuses on accurately describing water as a gas-phase molecule, and the water branch is targeted at aqueous chemistry.

AB.ff: (H/O/N/B) Ammonia Borane M.R. Weismiller, A.C.T. van Duin, J. Lee, R.A. Yetter *ReaxFF Reactive Force Field Development and Applications for Molecular Dynamics Simulations of Ammonia Borane Dehydrogenation and Combustion* *J. Phys. Chem. A* 2010, 114, 5485-5492 (<http://dx.doi.org/10.1021/jp100136c>)

- QM data were generated describing the single and (if relevant) double and triple bond dissociation for all B/N/O/H combinations. These data were used to derive initial ReaxFF bond parameters, and all calculations were performed using DFT with the B3LYP functional and the Pople 6-311G** basis set.
- The training set was then extended with QM data describing angular distortions in a set of small AB-related (AB H₃N-BH₃) molecules. These data were used to derive the initial ReaxFF angular parameters.
- The training set was extended with reaction barriers for key reaction steps such as H₂ release from AB, dimerization of H₂B-NH₂ and reaction energies associated with H₂ release from AB and with AB oxidation.
- Branch: combustion.

AuCSOH.ff: (Au/C/S/O/H) J. A. Keith, D. Fantauzzi, T. Jacob, and A.C.T. van Duin *Reactive force-field for simulating gold surfaces and nanoparticles* *Physical Review B* (2010) 81, 235404-1/235404-8 (<http://dx.doi.org/10.1103/PhysRevB.81.235404>)

- The original Au-Au parameters were extended by three publications:
- Au/O: K. Joshi, A.C.T. van Duin, and T. Jacob Development of a ReaxFF description of gold oxides and initial application to cold welding of partially oxidized gold surfaces *Journal of Materials Chemistry* 20, (2010), 10431-10437 <http://dx.doi.org/10.1039/C0JM01556C>
- Au/C/S/H: T.T. Jarvi, A.C.T. van Duin, K. Nordlund, and W.A. Goddard Development of interatomic ReaxFF potentials for Au-S-C-H systems *Journal of Physical Chemistry C* 115, (2011), 10315-10322 <http://dx.doi.org/10.1021/jp201496x>

- C/O/H/S: Rahaman, O., van Duin, A. C. T., Goddard, W. A., III, and Doren, D. J. Development of a ReaxFF reactive force field for glycine and application to solvent effect and tautomerization *Journal of Physical Chemistry B* 115 (2011), 249-261 <http://dx.doi.org/10.1021/jp108642r>
- The forcefield does not include Au/N parameters
- Branch: water.

CHO.ff: (C/H/O) Hydrocarbon oxidation K. Chenoweth, A.C.T. van Duin, W. A. Goddard *ReaxFF Reactive Force Field for Molecular Dynamics Simulations of Hydrocarbon Oxidation* *J. Phys. Chem. A* 2008, 112, 1040-1053 (<http://dx.doi.org/10.1021/jp709896w>)

- To obtain the H/C/O compound data required to extend the hydrocarbon-training set, DFT calculations were performed on the following systems: (a) dissociation energies for various bonds containing carbon, oxygen, and hydrogen. The ground state structure was obtained through full geometry optimization. Dissociation curves were calculated by constraining only the bond length of interest and re-optimization of the remaining internal coordinates. Optimization was also performed for the various angles and torsions associated with C/H/O interactions.
- Branch: combustion.

HCONSB.ff: (H/C/O/N/S/B) M.R. Weismiller, A.C.T. van Duin, J. Lee, and R.A. Yetter, *ReaxFF Reactive Force Field Development and Applications for Molecular Dynamics Simulations of Ammonia Borane Dehydrogenation and Combustion* *J. Phys. Chem. A* (2010), 114, 5485-5492. (<http://dx.doi.org/10.1021/jp100136c>)

- The parameters in this forcefield were extended/improved by two other publications:
- A.M. Kamat, A.C.T. van Duin, and A. Yakovlev *Molecular Dynamics Simulations of Laser-Induced Incandescence of Soot Using an Extended ReaxFF Reactive Force Field*. *Journal of Physical Chemistry A* (2010), 114, 12561-1257 <http://dx.doi.org/10.1021/jp1080302>
- F. Castro-Marcano, A.M. Kamat, M.F. Russo, A.C.T. van Duin, and J.P. Mathews *Combustion of an Illinois No. 6 Coal Char Simulated Using an Atomistic Char Representation and the ReaxFF Reactive Force Field*. *Combustion and Flame* (2012), 159, 23273-1285 <http://dx.doi.org/10.1016/j.combustflame.2011.10.022>
- The C/H/O parameters are the same as in the CHO forcefield, with added S/C, S/H and S/O descriptions. This force field was used in Castro et al, *Combustion and Flame* 2011
- The Boron and Nitrogen parameters are based on (but not identical to) the parameters used in Weismiller et al, *JPC-A* 2010.
- Branch: combustion.

CuCl-H2O.ff: (Cu/Cl/H/O) O. Rahaman, A.C.T. van Duin, V.S. Bryantsev, J.E. Mueller, S.D. Solares, W.A. Goddard III, and D.J. Doren *Development of a ReaxFF Reactive Force Field for Aqueous Chloride and Copper Chloride* *J. Phys. Chem. A* 114 (2010), 3556-3568 (<http://dx.doi.org/10.1021/jp9090415>)

- This forcefield is an extension of: A.C.T. van Duin, V.S. Bryantsev, M.S. Diallo, W.A. Goddard, O. Rahaman, D.J. Doren, D. Raymand, and K. Hermansson *Development and validation of a ReaxFF reactive force field for Cu cation/water interactions and copper metal/metal oxide/metal hydroxide condensed phases* *Journal of Physical Chemistry A* 2010, 114, 9507-9514 <http://dx.doi.org/10.1021/jp102272z>
- Branch: water.

FeOCHCl.ff: (Fe/O/C/H/Cl) M. Aryanpour, A.C.T. van Duin, J.D. Kubicki *Development of a Reactive Force Field for Iron-Oxyhydroxide Systems* *J. Phys. Chem. A* 2010, 114, 6298-6307 (<http://dx.doi.org/10.1021/jp101332k>)

- The Cl parameters were published by: O. Rahaman, A.C.T. van Duin, V.S. Bryantsev, J.E. Mueller, S.D. Solares, W.A. Goddard III, and D.J. Doren *Development of a ReaxFF Reactive Force Field for Aqueous Chloride and Copper Chloride* *J. Phys. Chem. A* 114 (2010), 3556-3568 <http://dx.doi.org/10.1021/jp9090415>

- The initial force field parameters for the Fe-Fe parameters were taken from an earlier force field development project on bulk-iron metal, based on DFT-calculations on antiferromagnetic BCC and FCC. The ReaxFF parameters have not been published yet, however the DFT data can be found in ref 31 of the above mentioned manuscript. The O/H parameters were taken from the ReaxFF bulk water description. The Fe/Fe and O/H parameters were kept fixed to these initial values, whereas the Fe/O parameters were reoptimized against the quantum mechanical results presented in the above mentioned manuscript.
- Detailed information on the force field parameters is given in the supporting information of the above mentioned manuscript.
- Branch: water.

HE.ff: (C/H/O/N) RDX/High Energy L.Z. Zhang, A.C.T. van Duin, S.V. Zybin, and W.A. Goddard *Thermal Decomposition of Hydrazines from Reactive Dynamics Using the ReaxFF Reactive Force Field* *Journal of Physical Chemistry B* (2009) 113, 10770-10778 (<http://dx.doi.org/10.1021/jp900194d>)

- Part of this forcefield is also published in: L.Z. Zhang, S.V. Zybin, A.C.T. van Duin, S. Dasgupta, W.A. Goddard, and E.M. Kober *Carbon Cluster Formation during Thermal Decomposition of Octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine and 1,3,5-Triamino-2,4,6-trinitrobenzene High Explosives from ReaxFF Reactive Molecular Dynamics Simulations* *Journal of Physical Chemistry A* (2009) 113, 10619-10640 <http://dx.doi.org/10.1021/jp901353a>
- The parameters of the nitramine ReaxFF are based on a large number of ab initio QM calculations. Over 40 reactions and over 1600 equilibrated molecules have been used; they are designed to characterize the atomic interactions under various environments likely and unlikely high energy each atom can encounter. The training set contains bond breaking and compression curves for all possible bonds, angle and torsion bending data for all possible cases, as well as crystal data.
- Please see the supplemental material from *Phys. Rev. Lett.* 2003, 91, 098301 for a detailed description of the parameterization of this force field.
- Branch: combustion.

HE2.ff: (C/H/O/N/S/Si) RDX/High Energy L.Z. Zhang, S.V. Zybin, A.C.T. van Duin, S. Dasgupta, W.A. Goddard, and E.M. Kober *Carbon Cluster Formation during Thermal Decomposition of Octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine and 1,3,5-Triamino-2,4,6-trinitrobenzene High Explosives from ReaxFF Reactive Molecular Dynamics Simulations* *Journal of Physical Chemistry A* (2009) 113, 10619-10640 (<http://dx.doi.org/10.1021/jp901353a>)

- Similar to HE.ff with additional parameters fitted for the TATB explosive
- Branch: combustion.

NaH.ff: (Na/H) J.G.O. Ojwang, R. Van Santen, G.J. Kramer, A.C.T. van Duin, and W.A. Goddard *Modeling the sorption dynamics of NaH using a reactive force field* *Journal of Chemical Physics* 2008, 128, 164714 (<http://dx.doi.org/10.1063/1.2908737>)

- This forcefield originally had a typo, defining the H-Na-Na angle twice. The same typo is in table 4 of the publication, but the text clearly mentions that the second line should define Na-H-Na instead.
- Branch: combustion.

NiCH.ff: (Ni/C/H) J.E. Mueller, A.C.T. van Duin, and W.A. Goddard III *Development and Validation of ReaxFF Reactive Force Field for Hydrocarbon Chemistry Catalyzed by Nickel* *J. Phys. Chem. C* 2010, 114, 4939-4949 (<http://dx.doi.org/10.1021/jp9035056>)

- Branch: combustion.

SiOH.ff: (Si/O/H) J.C. Fogarty, H.M. Aktulga, A.Y. Grama, A.C.T. van Duin, S.A. Pandit *A reactive molecular dynamics simulation of the silica-water interface* *J. Chem. Phys.* 2010, 132, 174704 (<http://dx.doi.org/10.1063/1.3407433>)

- This force field was trained to model the interaction of water at the SiO₂ surface, with specific emphasis on proton-transfer reactions. Updated parameters were fitted for all Si/O/H bond, angle, and torsion interactions as well, in addition to the dissociation of a water molecule from a single Si(OH)₄ molecule and reaction energies for the polymerization of Si(OH)₄
- Branch: water.

SiC.ff: (Si/C/O/H/N/S) D. Newsome, D. Sengupta, H. Foroutan, M.F. Russo, and A.C.T. van Duin *Oxidation of Silicon Carbide by O₂ and H₂O: A ReaxFF Reactive Molecular Dynamics Study, Part I* *Journal of Physical Chemistry* 2012 116, 16111-16121 (<http://dx.doi.org/10.1021/jp306391p>)

- The included forcefield is based on the Newsome reference, with slightly improved parameters by van Duin et al.
- Branch: combustion.

VOCH.ff: (V/O/C/H) K. Chenoweth, A.C.T. van Duin, P. Persson, M.J. Cheng, J. Oxgaard, W.A. Goddard *Development and Application of a ReaxFF Reactive Force Field for Oxidative Dehydrogenation on Vanadium Oxide Catalysts* *J. Phys. Chem. C*, 2008, 112, 14645-14654 (<http://dx.doi.org/10.1021/jp802134x>)

- The ReaxFF force field parameters have been fit to a large quantum mechanics (QM) training set containing over 700 structures and energetics related to bond dissociations, angle and dihedral distortions, and reactions between hydrocarbons and vanadium oxide clusters. In addition, the training set contains charge distributions for small vanadium oxide clusters and the stabilities of condensed-phase systems including V₂O₅, VO₂, and V₂O₃ in addition to metallic V (V₀).
- Branch: combustion.

ZnOH.ff: (Zn/O/H) D. Raymand, A.C.T. van Duin, M. Baudin, K. Hermansson *A reactive force field (ReaxFF) for zinc oxide* *Surface Science* 2008, 602, 1020-1031 (<http://dx.doi.org/10.1016/j.susc.2007.12.023>)

- updated version published by: D. Raymand, A.C.T. van Duin, D. Spangberg, W.A. Goddard, K. Hermansson *Water adsorption on stepped ZnO surfaces from MD simulation* *Surface Science* 2010, 604, 9-10, 741-752 <http://dx.doi.org/10.1016/j.susc.2009.12.012>
- Based on QM calculations for Zn(s), ZnO(s), and Zn hydroxide clusters [Zn(OH)₂ and O(ZnOH)₂], ReaxFF parameters were generated for Zn-O and Zn-Zn bond energies and for Zn-O-Zn, O-Zn-O, O-Zn-Zn and Zn-O-H valence angle energies.
- QM calculations were performed for the four crystal polymorphs of the wurtzite, zincblende, rocksalt and caesium chloride structures (the structures are also referred to as h-ZnS, c-ZnS, NaCl and CsCl, respectively).
- Branch: water.

Al-H₂O.ff: (Al/H/O) M. Russo, R. Li, M. Mench, and A.C.T. van Duin *Molecular Dynamic Simulation of Aluminum-Water Reactions Using the ReaxFF Reactive Force Field* *International Journal of Hydrogen Energy* 36 (2011), 5828-5835 (<http://dx.doi.org/10.1016/j.ijhydene.2011.02.035>)

- Branch: water.

CaSiAlO.ff: (C/H/O/Fe/Cl/Si/Al/Ca) M.C. Pitman and A.C.T. van Duin *Dynamics of Confined Reactive Water in Smectite Clay-Zeolite Composites* *J. Am. Chem. Soc.*, 2012, 134 (6), 3042-3053 (<http://dx.doi.org/10.1021/ja208894m>)

- Branch: water.

dispersion/CHONSSi-Ig.ff: (C/H/O/N/S/Si) L. Liu, Y. Liu, S.V. Zybin, H. Sun, and W.A. Goddard III *ReaxFF-Ig: Correction of the ReaxFF Reactive Force Field for London Dispersion, with Applications to the Equations of State for Energetic Materials* *The Journal of Physical Chemistry A*, 2011, 115(40), 11016-11022 (<http://dx.doi.org/10.1021/jp201599t>)

- This forcefield adds London dispersion correction terms to reaxFF, and is optimized for the energetic materials RDX, PETN, TATB, and NM plus graphite, polyethylene, solid carbon dioxide, and solid N₂, using the low temperature crystal structures to determine the lg correction parameters.
- Branch: combustion.

CHOFeAlNiCuS.ff: (C/H/O/Fe/Al/Ni/Cu/S) O. Rahaman, A.C.T. van Duin, W.A. Goddard III, and D.J. Doren *Development of a ReaxFF reactive force field for glycine and application to solvent effect and tautomerization* *Journal of Physical Chemistry B* 115 (2011), 249-261. (<http://dx.doi.org/10.1021/jp204894m>)

- C/O/H parameters only. The Cu/Fe/Al/Ni parameters are from: Y.K. Shin, H. Kwak, C. Zou, A.V. Vasenkov, and A.C.T. van Duin *Development and Validation of a ReaxFF Reactive Force Field for Fe/Al/Ni Alloys: Molecular Dynamics Study of Elastic Constants, Diffusion, and Segregation* *J. Phys. Chem. A*, 2012, 116 (49), pp 12163–12174 <http://dx.doi.org/10.1021/jp308507x>
- Not all cross-terms between the two forcefield files are defined, which might cause problems if the system has (for example) C-Cu interactions.
- Branch: water.

AuSCH_2011.ff: (Au/S/C/H) T.T. Jarvi, A.C.T. van Duin, K. Nordlund, and W.A. Goddard III, *Development of Interatomic ReaxFF Potentials for Au-S-C-H Systems* *J. Phys. Chem. A*, 115, 10315-10322 (2011) (<http://dx.doi.org/10.1021/jp201496x>)

- Branch: combustion.

AuSCH_2013.ff: (Au/S/C/H) Gyun-Tack Bae and Christine M. Aikens *Improved ReaxFF Force Field Parameters for Au-S-C-H Systems* *Journal of Physical Chemistry A* 2013 117 (40), 10438-10446 (<http://dx.doi.org/10.1021/jp405992m>)

- Based upon: T.T. Jarvi, A.C.T. van Duin, K. Nordlund, and W.A. Goddard *Development of interatomic ReaxFF potentials for Au-S-C-H systems*. *Journal of Physical Chemistry C* 115, (2011), 10315-10322.
- yields improvements for bond bending potential energy surfaces
- aims to agree with DFT geometries of small clusters and gold-thiolate nanoparticles
- Branch: combustion.

PDMSDecomp.ff: (C/H/O/Si) K. Chenoweth, S. Cheung, A.C.T. van Duin, W.A. Goddard III, and E.M. Kober *Simulations on the Thermal Decomposition of a Poly(dimethylsiloxane) Polymer Using the ReaxFF Reactive Force Field* *J. Am. Chem. Soc.*, 2005, 127 (19), pp 7192-7202 (<http://dx.doi.org/10.1021/ja050980t>)

- Specialized forcefield, designed to “investigate the failure of the poly(dimethylsiloxane) polymer (PDMS) at high temperatures and pressures and in the presence of various additives”
- Line from the torsion block was referring to non-existent atoms from the atomic block and thus was removed.
- Branch: combustion.

TiOCHNCl.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F) S.Y. Kim, A.C.T. van Duin, and J.D. Kubicki *Molecular dynamics simulations of the interactions between TiO₂ nanoparticles and water with Na⁺ and Cl⁻, methanol, and formic acid using a reactive force field* *Journal of Materials Research / Volume 28 / Issue 03 / 2013*, pp 513-520 (<http://dx.doi.org/10.1557/jmr.2012.367>)

- used for simulating TiO₂(both rutile and anatase) nanoparticles with water, methanol, and formic acid
- The force field was validated by comparing water dissociative adsorption percentage and bond length between Na-O with density functional theory (DFT) and experimental results
- Branch: water.

PtCH.ff: (C/H/Pt) C.F. Sanz-Navarro, P. Astrand, De Chen, M. Ronning, A.C.T. van Duin, T. Jacob, and W.A. Goddard III *Molecular Dynamics Simulations of the Interactions between Platinum Clusters and Carbon Platelets* J. Phys. Chem. A 112, 1392-1402 (2008) (<http://dx.doi.org/10.1021/jp074806y>)

- Branch: combustion.

BaYZrCHO.ff: (C/H/O/Ba/Zr/Y) A.C.T. van Duin, B.V. Merinov, S.S. Jang, and W.A. Goddard III *ReaxFF Reactive Force Field for Solid Oxide Fuel Cell Systems with Application to Oxygen Ion Transport in Yttria-Stabilized Zirconia* J. Phys. Chem. A, 112, 3133-3140 (2008) (<http://dx.doi.org/10.1021/jp076775c>)

- Branch: combustion.

CHONSSiPtZrNiCuCo.ff: (C/H/O/N/S/Si/Pt/Zr/Ni/Cu/Co) K.D. Nielson, A.C.T. van Duin, J. Oxgaard, W.Q. Deng, and W.A. Goddard III *Development of the ReaxFF Reactive Force Field for Describing Transition Metal Catalyzed Reactions, with Application to the Initial Stages of the Catalytic Formation of Carbon Nanotubes* J. Phys. Chem. A, 109, 493-499 (2005) (<http://dx.doi.org/10.1021/jp046244d>)

- Branch: combustion.

Glycine.ff: (C/H/O/N) O. Rahaman, A.C.T. van Duin, W.A. Goddard III, and D.J. Doren, *Development of a ReaxFF Reactive Force Field for Glycine and Application to Solvent Effect and Tautomerization* J. Phys. Chem. B, 115, 249-261 (2011) (<http://dx.doi.org/10.1021/jp108642r>)

- Line from the valence angle block was referring to non-existent atoms from the atomic block and thus was removed.
- Branch: water.

SiONH.ff: (C/H/O/N/Si/S) A.D. Kulkarni, D.G. Truhlar, S.G. Srinivasan, A.C.T. van Duin, P. Norman, and T.E. Schwartzentruber *Oxygen Interactions with Silica Surfaces: Coupled Cluster and Density Functional Investigation and the Development of a New ReaxFF Potential* J. Phys. Chem. C, 2013, 117 (1), pp 258-269 (<http://dx.doi.org/10.1021/jp3086649>)

- Aimed at oxygen interactions with realistic silica surfaces
- Lines from the valence angle block was referring to non-existent atoms from the atomic block and thus was removed.
- Branch: combustion.

CHOFe.ff: (C/H/O/Fe/Cl/Si/Al) Chenyu Zou, A.C.T. Van Duin *Investigation of Complex Iron Surface Catalytic Chemistry Using the ReaxFF Reactive Force Field Method* JOM, December 2012, Volume 64, Issue 12, pp 1426-1437 (<http://dx.doi.org/10.1007/s11837-012-0463-5>)

- only the parameters for Fe (and crossterms) differ from the CHOAlSi.ff forcefield
- Branch: water.

CHOAlSi.ff: (C/H/O/Fe/Cl/Si/Al) F. Castro-Marcanao, A.C.T. van Duin *Comparison of thermal and catalytic cracking of 1-heptene from ReaxFF reactive molecular dynamics simulations* Combustion and Flame, Volume 160, Issue 4, April 2013, Pages 766-775 (<http://dx.doi.org/10.1016/j.combustflame.2012.12.007>)

- only the parameters for Fe (and crossterms) differ from the CHOFe.ff forcefield
- Branch: water.

CHOLi.ff: (C/H/O/N/S/Mg/P/Na/Li) D. Bedrov, G.D. Smith, A.C.T. van Duin *Reactions of Singly-Reduced Ethylene Carbonate in Lithium Battery Electrolytes: A Molecular Dynamics Simulation Study Using the ReaxFF* Journal of Physical Chemistry A, 2012, 116 (11), pp 2978-2985 (<http://dx.doi.org/10.1021/jp210345b>)

- specifically generated for simulating Lithium battery electrolytes
- must be used in combination with the MOLCHARGE keyword to set a charge restraint on Li and CO3!
- Branch: water.

SiOAlLi.ff: (H/O/Si/Al/Li) B. Narayanan, A.C.T. van Duin, B.B. Kappes, I.E. Reimanis and C.V. Ciobanu *A reactive force field for lithium-aluminum silicates with applications to eucryptite phases* *Modelling and Simulation in Materials Science and Engineering* 2012 20 015002 (<http://dx.doi.org/10.1088/0965-0393/20/1/015002>)

- Branch: water.

PdO.ff: (Pd/O) T.P. Senftle, R.J. Meyer, M.J. Janik and A.C.T. van Duin *Development of a ReaxFF potential for Pd/O and application to palladium oxide formation* *The Journal of Chemical Physics* 139, 044109 (2013) (<http://dx.doi.org/10.1063/1.4815820>)

- used for studying Oxidation states of Pd nanoparticles, surfaces and bulk configurations with a GCMC method
- Branch: combustion.

PdH.ff: (Pd/H) T.P. Senftle, M.J. Janik and A.C.T. van Duin *A ReaxFF Investigation of Hydride Formation in Palladium Nanoclusters via Monte Carlo and Molecular Dynamics Simulations* *The Journal of Physical Chemistry C*, 2014, 118 (9), pp 4967-4981 (<http://dx.doi.org/10.1021/jp411015a>)

- used in combination with a GCMC method
- Branch: combustion.

Co.ff: (Co) Xue-Qing Zhang, E. Iype, S.V. Nedeia, A.P.J. Jansen, B.M. Szyja, E.J.M. Hensen, and R.A. van Santen *Site Stability on Cobalt Nanoparticles: A Molecular Dynamics ReaxFF Reactive Force Field Study* *The Journal of Physical Chemistry C*, 2014, 118 (13), pp 6882-6886 (<http://dx.doi.org/10.1021/jp500053u>)

- forcefield was generated using a recently developed Monte Carlo algorithm with simulated annealing.
- Branch: combustion.

CHONSMgPNaCuCl.ff: (C/H/O/N/S/Mg/P/Na/Cu/Cl) Susanna Monti, Cui Li, and Vincenzo Carravetta *Reactive Dynamics Simulation of Monolayer and Multilayer Adsorption of Glycine on Cu(110)* *J. Phys. Chem. C*, 2013, 117 (10), pp 5221-5228 (<http://dx.doi.org/10.1021/jp312828d>)

- Reactive MD-force field for amino acids on copper
- Branch: water.

CHOSMoNiLiBFPN.ff: (C/H/O/S/Mo/Ni/Li/B/F/P/N) Md M. Islam, V.S. Bryantsev, A.C.T. van Duin *ReaxFF Reactive Force Field Simulations on the Influence of Teflon on Electrolyte Decomposition during Li/SWCNT Anode Discharge in Lithium-Sulfur Batteries* *J. Electrochem. Soc.* 2014 volume 161, issue 8, E3009-E3014 (<http://dx.doi.org/10.1149/2.005408jes>)

- forcefield for Electrochemistry in Li-S batteries
- Branch: combustion.

CHONSSiNaFZr.ff: (C/H/O/N/S/Si/Na/F/Zr) A. Rahnamoun and A.C.T. van Duin *Reactive Molecular Dynamics Simulation on the Disintegration of Kapton, POSS Polyimide, Amorphous Silica, and Teflon during Atomic Oxygen Impact Using the Reaxff Reactive Force-Field Method* *J. Phys. Chem. A*, 2014, 118 (15), pp 2780-2787 (<http://dx.doi.org/10.1021/jp4121029>)

- comments in the forcefield file: interactions with water and Na+ Fogarty et al. JCP-2010 ; with glycine + C/H/F parameters; Si-F bond/offdiag/angle parameters; Si-S dummy parameters + S-O-H parameters(Yun 2012 Oct8) + H-F bond/offdiag (Jan14 2013 Joon) Jan31: added Zr/O/H/C
- Branch: water.

TiClOH.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F) Sung-Yup Kim and A.C.T. van Duin *Simulation of Titanium Metal/Titanium Dioxide Etching with Chlorine and Hydrogen Chloride Gases Using the ReaxFF Reactive Force Field* *J. Phys. Chem. A*, 2013, 117 (27), pp 5655-5663 (<http://dx.doi.org/10.1021/jp4031943>)

- adaptation/evolution of the TiOCHNCl.ff forcefield by Kim, S.-Y. et al.

- Branch: water.

CHONSSiNaAl.ff: (C/H/O/N/S/Si/Na/Al) C. Bai, L. Liu, and H. Sun *Molecular Dynamics Simulations of Methanol to Olefin Reactions in HZSM-5 Zeolite Using a ReaxFF Force Field* *J. Phys. Chem. C*, 2012, 116 (12), pp 7029-7039 (<http://dx.doi.org/10.1021/jp300221j>)

- used to simulate methanol to olefin (MTO) reactions in H-ZSM-5 zeolite
- Branch: water.

undocumented/NiCH.ff: (Ni/C/H/O/N/S/F/Pt/Cl) Unknown *No Name* Not published (<http://>)

- This forcefield is from april 2009, and is not published
- Branch: combustion.

LiS.ff: (Li/S) Md M. Islam, A. Ostadhossein, O. Borodin, A.T. Yeates, W.W. Tipton, R.G. Hennig, N. Kumar, and A.C.T. van Duin *ReaxFF molecular dynamics simulations on lithiated sulfur cathode materials* *Phys. Chem. Chem. Phys.*, 2015, 17, 3383-3393 (<http://dx.doi.org/10.1039/C4CP04532G>)

- Developed for and used to study Sulfur cathode behavior in Li battery cells
- Branch: combustion.

CHONSSiPtNiCuCoZrYBa.ff: (C/H/O/N/S/Si/Pt/Ni/Cu/Co/Zr/Y/Ba) B.V. Merinov, J.E. Mueller, A.C.T. van Duin, Qi An, and W.A. Goddard III *ReaxFF Reactive Force-Field Modeling of the Triple-Phase Boundary in a Solid Oxide Fuel Cell* *J. Phys. Chem. Lett.*, 2014, 5 (22), pp 4039-4043 (<http://dx.doi.org/10.1021/jz501891y>)

- Force field was developed by combining the YSZ and Ni/C/H descriptions
- From the summary: “The products obtained in our simulations are the same as those in experiment, which indicates that the developed ReaxFF potential properly describes complex physicochemical processes, such as the oxide-ion diffusion, fuel conversion, water formation reaction, coking, and delamination, occurring at the TPB and can be recommended for further computational studies of the fuel/electrode/electrolyte interfaces in a SOFC”
- The Aa atom type is a rename of the Zr atom type, which was defined twice.
- Branch: combustion.

CHONSSiNaP.ff: (C/H/O/N/S/Si/Na/P) Bo Zhang, A.C.T. van Duin and J.K. Johnson *Development of a ReaxFF Reactive Force Field for Tetrabutylphosphonium Glycinate/CO₂ Mixtures* *J. Phys. Chem. B*, 2014, 118 (41), pp 12008-12016 (<http://dx.doi.org/10.1021/jp5054277>)

- The forcefield file mentions “Gly with dummy N-P parameters”
- It was developed for studying Carbon dioxide interactions with the ionic liquid tetrabutylphosphonium glycinate, both physical and chemical absorption.
- Several lines from valence angle block were referring to non-existent atoms from the atomic block and thus were removed.
- Branch: combustion.

CHOLi_2.ff: (C/H/O/Li) M. Raju, P. Ganesh, P.R.C. Kent, and A.C.T. van Duin *Reactive Force Field Study of Li/C Systems for Electrical Energy Storage* *J. Chem. Theory Comput.*, 2015, 11 (5), pp 2156-2166 (<http://dx.doi.org/10.1021/ct501027v>)

- Used for studying Li/C systems with GCMC and MD
- Also contains parameters for “Vi” and “GI”, but these virtual elements are not mentioned in the paper
- Parameters were fitted using a training set consisting, amongst others, of Li binding energies on pristine graphene and graphite, Li migration pathways in graphite and Li dissociation pathways in different hydrocarbons

- The forcefield was validated by a side-by-side comparison of ReaxFF and DFT energies for Li binding on a divacancy, and ReaxFF and DFT ground-state configurations for stage II and stage I stacking in graphite obtained using GCMC simulations
- The forcefield reproduces the in-plane Li ordering as well as the graphite stacking sequence for stage II and stage I compounds
- Branch: water.

CHONSiCaCsKSrNaMgAlCu.ff: (C/H/O/N/S/Si/Ca/Cs/K/Sr/Na/Mg/Al/Cu) G.M. Psfogiannakis, J.F. McCleerey, E. Jaramillo and A.C.T. van Duin *ReaxFF Reactive Molecular Dynamics Simulation of the Hydration of Cu-SSZ-13 Zeolite and the Formation of Cu Dimers* *J. Phys. Chem. C*, 2015, 119 (12), pp 6678-6686 (<http://dx.doi.org/10.1021/acs.jpcc.5b00699>)

- This Cu/Si/Al/O/H forcefield was developed for and used in MD simulations of the hydration of Cu-exchanged SSZ-13 catalyst.
- The force field was developed by merging and expanding previously tested and published ReaxFF force fields for Si/Al/O/H systems and Cu/O/H systems (citations 19-26 of the publication)
- This forcefield is an extension of K.L. Joshi et al 2014, <http://dx.doi.org/10.1039/C4CP02612H>, all non-Cu parameters are the same.
- Branch: water.

HOSMg.ff: (H/O/S/Mg) E. Iype, M. Hütter, A.P.J. Jansen, S.V. Nedeia and C.C.M. Rindt *Parameterization of a reactive force field using a Monte Carlo algorithm* *J. Comput. Chem.*, 2013, 34: 1143-1154 (<http://dx.doi.org/10.1002/jcc.23246>)

- This forcefield is optimized with a metropolis Monte-Carlo algorithm with simulated annealing to search for the optimum parameters for the ReaxFF force field in a high-dimensional parameter space.
- The optimization is done against a set of quantum chemical data for MgSO₄ hydrates.
- Branch: combustion.

CHONSMgPNaCuCl_v2.ff: (C/H/O/N/S/Mg/P/Na/Cu/Cl) S. Monti, A. Corozzi, P. Fristrup, K.L. Joshi, Yun Kyung Shin, P. Oelschlaeger, A.C.T. van Duin and V. Barone *Exploring the conformational and reactive dynamics of biomolecules in solution using an extended version of the glycine reactive force field* *Phys. Chem. Chem. Phys.*, 2013, 15, 15062-15077 (<http://dx.doi.org/10.1039/C3CP51931G>)

- This forcefield is an extension of CHONSMgPNaCuCl.ff by S. Monti et al., *J. Phys. Chem. C*, 2013, 117 (10), pp 5221-5228, <http://dx.doi.org/10.1021/jp312828d>
- Developed for peptide and protein simulations
- Branch: water.

OPt.ff: (O/Pt) D. Fantauzzi, J. Bandlow, L. Sabo, J.E. Mueller, A.C.T. van Duin and T. Jacob *Development of a ReaxFF potential for Pt-O systems describing the energetics and dynamics of Pt-oxide formation* *Phys. Chem. Chem. Phys.*, 2014, 16, 23118-23133 (<http://dx.doi.org/10.1039/C4CP03111C>)

- Pt-Pt parameters for bulk platinum phases, low & high-index platinum surfaces and nanoclusters.
- O-Pt parameters for bulk platinum oxides, as well as oxygen adsorption and oxide formation on Pt(111) terraces and the {111} and {100} steps connecting them.
- Branch: combustion.

CHONSMgPNaTiClF.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F) Stijn Huygh, Annemie Bogaerts, Adri C.T. van Duin and Erik C. Neyts *Development of a ReaxFF reactive force field for intrinsic point defects in titanium dioxide* *Computational Materials Science*, Volume 95, December 2014, Pages 579-591, ISSN 0927-0256 (<http://dx.doi.org/10.1016/j.commatsci.2014.07.056>)

- Forcefield developed for studying the influence of intrinsic point defects on the chemistry with TiO₂ condensed phases.
- Forcefield seems to be based on TiOCHNCl.ff, <http://dx.doi.org/10.1557/jmr.2012.367>
- Branch: water.

LiSi.ff: (Li/Si) A. Ostadhosseini, E.D. Cubuk, G.A. Tritsarlis, E. Kaxiras, S. Zhanga and A.C.T. van Duin *Stress effects on the initial lithiation of crystalline silicon nanowires: reactive molecular dynamics simulations using ReaxFF* Phys. Chem. Chem. Phys., 2015, 17, 3832-3840 (<http://dx.doi.org/10.1039/C4CP05198J>)

- Branch: combustion.

CHOFAlNiCuSCr.ff: (C/H/O/Fe/Al/Ni/Cu/S/Cr) Y.K. Shin, H. Kwak, A.V. Vasenkov, D. Sengupta and A.C.T. van Duin *Development of a ReaxFF Reactive Force Field for Fe/Cr/O/S and Application to Oxidation of Butane over a Pyrite-Covered Cr₂O₃ Catalyst* ACS Catalysis, 2015, 5 (12), pp 7226-7236 (<http://dx.doi.org/10.1021/acscatal.5b01766>)

- Forcefield optimized for Fe/Cr/O/S
- Branch: water.

CHOFAlNiCuSCr_v2.ff: (C/H/O/Fe/Al/Ni/Cu/S/Cr) Ni-O-vacancy C. Zou et al. *Molecular dynamics simulations of the effects of vacancies on nickel self-diffusion, oxygen diffusion and oxidation initiation in nickel, using the ReaxFF reactive force field* Acta Materialia, Volume 83, 15 January 2015, Pages 102-112 (<http://dx.doi.org/10.1016/j.actamat.2014.09.047>)

- Forcefield optimized for Ni/O, trained with a QM data on Ni, NiO and vacancies
- The non-carbon parameters are the same as in CHOFAlNiCuSCr_v3.ff
- Branch: water.

CHOFAlNiCuSCr_v3.ff: (C/H/O/Fe/Al/Ni/Cu/S/Cr) F. Tavazza, T.P. Senftle, C. Zou, C.A. Becker and A.C.T. van Duin *Molecular Dynamics Investigation of the Effects of Tip-Substrate Interactions during Nanoindentation* J. Phys. Chem. C, 2015, 119 (24), pp 13580-13589 (<http://dx.doi.org/10.1021/acs.jpcc.5b01275>)

- Combination of the C.ff (condensed carbon) forcefield with Ni/C/O/H parameters
- The non-carbon parameters are the same as in CHOFAlNiCuSCr_v2.ff
- Branch: water.

C.ff: (C) C-2013 S.G. Srinivasan, A.C.T. van Duin and P. Ganesh *Development of a ReaxFF Potential for Carbon Condensed Phases and Its Application to the Thermal Fragmentation of a Large Fullerene* J. Phys. Chem. A, 2015, 119 (4), pp 571-580 (<http://dx.doi.org/10.1021/jp510274e>)

- Forcefield designed for modeling Carbon condensed phases
- Branch: combustion.

CHONSSiGe.ff: (C/H/O/N/S/Si/Ge) G. Psogiannakis and A.C.T. van Duin *Development of a ReaxFF reactive force field for Si/Ge/H systems and application to atomic hydrogen bombardment of Si, Ge, and SiGe (100) surfaces* Surface Science 2016, Vol. 646, pp 253-260 (<http://dx.doi.org/10.1016/j.susc.2015.08.019>)

- Forcefield designed for bombardment of Si, Ge and SiGe surfaces with atomic hydrogen.
- Branch: combustion.

CHONSFPtCINi.ff: (C/H/O/N/S/F/Pt/Cl/Ni) Water-Pt-Ni-Nafion D. Fantauzzi, J.E. Mueller, L. Sabo, A.C.T. van Duin and T. Jacob *Surface Buckling and Subsurface Oxygen: Atomistic Insights into the Surface Oxidation of Pt(111)* ChemPhysChem, 2015, Vol.16: 2797-2802 (<http://dx.doi.org/10.1002/cphc.201500527>)

- extension of the OPt.ff forcefield parameters
- Branch: water.

CHONSSiPtZrNiCuCoHeNeArKrXe.ff: (C/H/O/N/S/Si/Pt/Zr/Ni/Cu/Co/He/Ne/Ar/Kr/Xe) A.M. Kamat, A.C.T. van Duin and A. Yakovlev *Molecular Dynamics Simulations of Laser-Induced Incandescence of Soot Using an Extended ReaxFF Reactive Force Field* *J. Phys. Chem. A*, 2010, 114 (48), pp 12561-12572 (<http://dx.doi.org/10.1021/jp1080302>)

- forcefield designed for the study of Laser-induced incandescence of soot
- Branch: combustion.

CHOSFCIN.ff: (C/H/O/S/F/Cl/N) M.A. Wood, A.C.T. van Duin and A. Strachan *Coupled Thermal and Electromagnetic Induced Decomposition in the Molecular Explosive alpha-HMX; A Reactive Molecular Dynamics Study* *J. Phys. Chem. A*, 2014, 118 (5), pp 885-895 (<http://dx.doi.org/10.1021/jp406248m>)

- forcefield designed for studying the combustion of the high-energy material a-HMX
- Branch: combustion.

Mue2016.ff: (C/H/O/S) Julian Mueller and Bernd Hartke *ReaxFF Reactive Force Field for Disulfide Mechanochemistry, Fitted to Multireference ab Initio Data;* *J. Chem. Theory Comput.*, 2016, xxx, xxx (<http://dx.doi.org/10.1021/acs.jctc.6b00461>)

- forcefield for studying S-S bond ruptures in mechanophores upon mechanical stress. Ambient conditions, both in gas phase and toluene solvent.
- Branch: combustion.

CBN.ff: (C/H/B/N) Sung Jin Pai, Byung Chul Yea and Sang Soo Han *Reactive force field for the improved design of liquid CBN hydrogen storage materials* *Phys. Chem. Chem. Phys.*, 2016, 18, pp 1818-1827 (<http://dx.doi.org/10.1039/C5CP05486A>)

- forcefield for studying liquid CBN (carbon-boron-nitrogen) hydrogen-storage materials.
- Branch: combustion.

AgZnO.ff: (C/H/O/N/Si/Cu/Ag/Zn) A. Lloyd, D. Cornil, A.C.T. van Duin, D. van Duin, R. Smith, S.D. Kenny, J. Cornil and D. Beljonne *Development of a ReaxFF potential for Ag/Zn/O and application to Ag deposition on ZnO* *Surface Science*, 2016, 645, pp 67-73 (<http://dx.doi.org/10.1016/j.susc.2015.11.009>)

- ReaxFF potential for Ag/Zn/O used to study Ag deposition on ZnO.
- Branch: water.

AlCHO.ff: (Al/C/H/O) Sungwook Hong and Adri C. T. van Duin *Atomistic-Scale Analysis of Carbon Coating and Its Effect on the Oxidation of Aluminum Nanoparticles by ReaxFF-Molecular Dynamics Simulations* *J. Phys. Chem. C*, 2016, 120 (17), pp 9464-9474 (<http://dx.doi.org/10.1021/acs.jpcc.6b00786>)

- ReaxFF potential for Al/C interactions. Used to investigate carbon coating and its effect on the oxidation of aluminum nanoparticles (ANPs)
- Branch: water.

CHNa.ff: (C/H/Na) E. Hjertenaes, A.Q. Nguyen and H. Koch *A ReaxFF force field for sodium intrusion in graphitic cathodes* *Phys. Chem. Chem. Phys.*, 2016, 18, pp 31431-31440 (<http://dx.doi.org/10.1039/C6CP06774C>)

- The force field is applied in hybrid grand canonical Monte Carlo-molecular dynamics (GC-MC/MD) simulations of model systems representative of sodium intrusion in graphitic carbon cathodes used in aluminium electrolysis.
- Branch: combustion.

CuBTC.ff: (C/H/O/N/S/Mg/P/Na/Cu) L. Huang, T. Bandosz, K. L. Joshi, A. C. T. van Duin and K. E. Gubbins *Reactive adsorption of ammonia and ammonia/water on CuBTC metal-organic framework: A ReaxFF molecular dynamics simulation* *J. Chem. Phys.*, 2013, 138, 034102 (<http://dx.doi.org/10.1063/1.4774332>)

- The force field was used to study reactive adsorption of NH₃ on the dehydrated CuBTC metal-organic framework.
- Branch: water.

CHONSMgPNaTiClFAu.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F/Au) Susanna Monti, Vincenzo Carravetta, and Hans Ågren *Simulation of Gold Functionalization with Cysteine by Reactive Molecular Dynamics* *J. Phys. Chem. Lett.*, 2016, 7 (2), pp 272-276 (<http://dx.doi.org/10.1021/acs.jpcclett.5b02769>)

- The force field was designed to study gold-protein interactions in water.

HOSiAILi.ff: (H/O/Si/Al/Li) Alireza Ostadhossein, Sung-Yup Kim, Ekin D. Cubuk, Yue Qi, and Adri C. T. van Duin *Atomic Insight into the Lithium Storage and Diffusion Mechanism of SiO₂/Al₂O₃ Electrodes of Lithium Ion Batteries: ReaxFF Reactive Force Field Modeling* *J. Phys. Chem. A*, 2016, 120 (13), pp 2114-2127 (<http://dx.doi.org/10.1021/acs.jpca.5b11908>)

- Developed for studying the energetics and kinetics of lithiation, as well as Li transportation within the crystalline/amorphous silica and alumina phases.

CHArHeNeKr.ff: (C/H/Ar/He/Ne/Kr) Kichul Yoon, A. Rahnamoun, J.L. Swett, V. Iberi, D.A. Cullen, I.V. Vlas-siouk, A. Belianinov, S. Jesse, X. Sang, O.S. Ovchinnikova, A.J. Rondinone, R.R. Unocic, and A.C.T. van Duin *Atomistic-Scale Simulations of Defect Formation in Graphene under Noble Gas Ion Irradiation* *ACS Nano*, 2016, 10 (9), pp 8376-8384 (<http://dx.doi.org/10.1021/acs.nano.6b03036>)

- Developed for studying noble gas ion irradiation of graphene and the subsequent effects of annealing. Lattice defects including nanopores were generated.

CHO-radiation.ff: (C/H/O) Roger Smith, K. Jolley, C. Latham, M. Heggie, A van Duin, D. van Duin, Houzheng Wu *A ReaxFF carbon potential for radiation damage studies* *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, Volume 393, 15 February 2017, Pages 49-53, (<http://dx.doi.org/10.1016/j.nimb.2016.11.007>)

- Developed forcefield reproduces the formation energies of many of the defects predicted by the ab initio calculations of energetic impacts and collision cascades in graphite.
- Forcefield reproduces the formation energies of many of the defects predicted by the ab initio calculations and the energy pathways between different defect states, which are important for investigating long term defect evolution.

HOTiPd.ff: (H/O/Ti/Pd) Rafik Addou, T.P. Senftle, N. O'Connor, M.J. Janik, A.C.T. van Duin, and M. Batzill *Influence of Hydroxyls on Pd Atom Mobility and Clustering on Rutile TiO₂(011)-2 x 1* *ACS Nano*, 2014, 8 (6), pp 6321-6333 (<http://dx.doi.org/10.1021/nn501817w>)

- Developed for MC simulations of Pd on TiO₂ surfaces.

CHONSMgPNaFBLi-e.ff: (C/H/O/N/S/Mg/P/Na/F/B/Li/El/Ho) M. Islam and A.C.T van Duin *Reductive Decomposition Reactions of Ethylene Carbonate by Explicit Electron Transfer from Lithium: An eReaxFF Molecular Dynamics Study* *J. Phys. Chem. C*, 2016, 120 (48), pp 27128-27134 (<http://dx.doi.org/10.1021/acs.jpcc.6b08688>)

- This forcefield uses the ACKS2 charge model
- This forcefield is an eReaxFF forcefield
- Developed for the study of lithium-ion batteries
- NOTE: This ff uses vpar(38) for e-reaxff taper radius instead of vpar(41) that ADF-reaxff expects. This will generate a warning but should still work.

CHOFeAlNiCuScrSiGe.ff: (C/H/O/Fe/Al/Ni/Cu/S/Cr/Si/Ge) Y. Zheng, S. Hong, G. Psfogiannakis, S. Datta, B. Rayner, A.C.T. van Duin and R. Engel-Herbert *Modeling and In-situ Probing of Surface Reactions in Atomic Layer Deposition* *ACS Appl. Mater. Interfaces*, 2017, 9 (18), pp 15848-15856 (<http://dx.doi.org/10.1021/acsami.7b01618>)

- Used for studying the ALD process of Al₂O₃ from trimethylaluminum and water on hydrogenated and oxidized Ge(100) surfaces

Water2017.ff: (H/O/X) W. Zhang, A.C.T. van Duin *Second-Generation ReaxFF Water Force Field: Improvements in the Description of Water Density and OH-Anion Diffusion* *J. Phys. Chem. B*, 2017, 121 (24), pp 6021-6032 (<http://dx.doi.org/10.1021/acs.jpcc.7b02548>)

- Better description of liquid water

HSMo.ff: (H/S/Mo) A. Ostadhosseini, A. Rahnamoun, Y. Wang, P. Zhao, S. Zhang, V.H. Crespi, and A.C.T. van Duin *ReaxFF Reactive Force-Field Study of Molybdenum Disulfide (MoS₂)* *J. Phys. Chem. Lett.*, 2017, 8 (3), pp 631-640 (<http://dx.doi.org/10.1021/acs.jpclett.6b02902>)

- MoS₂ training set in supporting info
- strain-stress analysis

CHON2017_weak.ff: (C/H/O/N/S/Mg/P/Na/Cu/Cl/X) Weiwei Zhang and Adri C. T. van Duin *Improvement of the ReaxFF Description for Functionalized Hydrocarbon/Water Weak Interactions in the Condensed Phase* *J. Phys. Chem. B*, 2018, 122, 4083-4092 (<http://dx.doi.org/10.1021/acs.jpcc.8b01127>)

- retraining of the protein-2013 force-field (C, H, O, and N parameters for weak interactions)
- reproduces well the density of liquid water and hydrocarbons

CaSiOH.ff: (C/H/O/Ca/Si/X) Hegoi Manzano, Roland J. M. Pellenq, Franz-Josef Ulm, Markus J Buehler, and A.C.T. van Duin *Hydration of Calcium Oxide Surface Predicted by Reactive Force Field Molecular Dynamics* *Langmuir*, 2012, 28 (9), 4187-4197 (<http://dx.doi.org/10.1021/la204338m>)

- hydration of calcium oxide surface
- fitted using density functional theory calculations on gas phase calcium-water clusters, calcium oxide bulk and surface properties, calcium hydroxide, bcc and fcc Ca, and proton transfer reactions in the presence of calcium.

CHO-2016.ff: (C/H/O) Chowdhury Ashraf and Adri C.T. van Duin* *Extension of the ReaxFF Combustion Force Field toward Syngas Combustion and Initial Oxidation Kinetics* *J. Phys. Chem. A* 121, 5, 1051-1068 (<https://dx.doi.org/10.1021/acs.jpca.6b12429>)

- improved description of oxidation of small hydrocarbons and syngas reaction

CHON2017_weak_bb.ff: (C/H/O/N/S/Mg/P/Na/Cu/Cl/X) Vashisth A., Ashraf C., Zhang W., Bakis C.E., van Duin A.C.T. *Accelerated ReaxFF Simulations for Describing the Reactive Cross-Linking of Polymers* *J Phys Chem A*. 2018 Aug 1. (<https://doi.org/10.1021/acs.jpca.8b03826>)

- reparametrized version of CHON2017_weak.ff for bond-boost application

CH_aromatics.ff: (C/H) Qian Mao, Yihua Ren, K. H. Luo, Adri C. T. van Duin *Dynamics and kinetics of reversible homo-molecular dimerization of polycyclic aromatic hydrocarbons* *The Journal of Chemical Physics* 147, 244305 (2017) (<https://doi.org/10.1063/1.5000534>)

- ff for polycyclic aromatic hydrocarbons

CuSCH.ff: (C/H/O/S/Cu/Cl/X) Jejoon Yeon, Heather L. Adams, Chad E. Junkermeier, Adri C. T. van Duin, Wilfred T. Tysoe, and Ashlie Martini *Development of a ReaxFF Force Field for Cu/S/C/H and Reactive MD Simulations of Methyl Thiolate Decomposition on Cu (100)* *J. Phys. Chem. B*, 2018, 122 (2), pp 888-896 (<https://doi.org/10.1021/acs.jpcc.7b06976>)

- combination of CuCl-H₂O.ff and AuSCH_2011.ff
- reparametrized Cu-S parameters with copper sulfides data

TiO₂bio.ff: (C/H/O/N/S/Mg/P/Na/Ti/Cl/F/X) Susanna Monti, Mariachiara Pastore, Cui Li, Filippo De Angelis, and Vincenzo Carravetta *Theoretical Investigation of Adsorption, Dynamics, Self-Aggregation, and Spectroscopic*

Properties of the D102 Indoline Dye on an Anatase (101) Substrate J. Phys. Chem. C, 2016, 120 (5), pp 2787-2796 (<https://doi.org/10.1021/acs.jpcc.5b11332>)

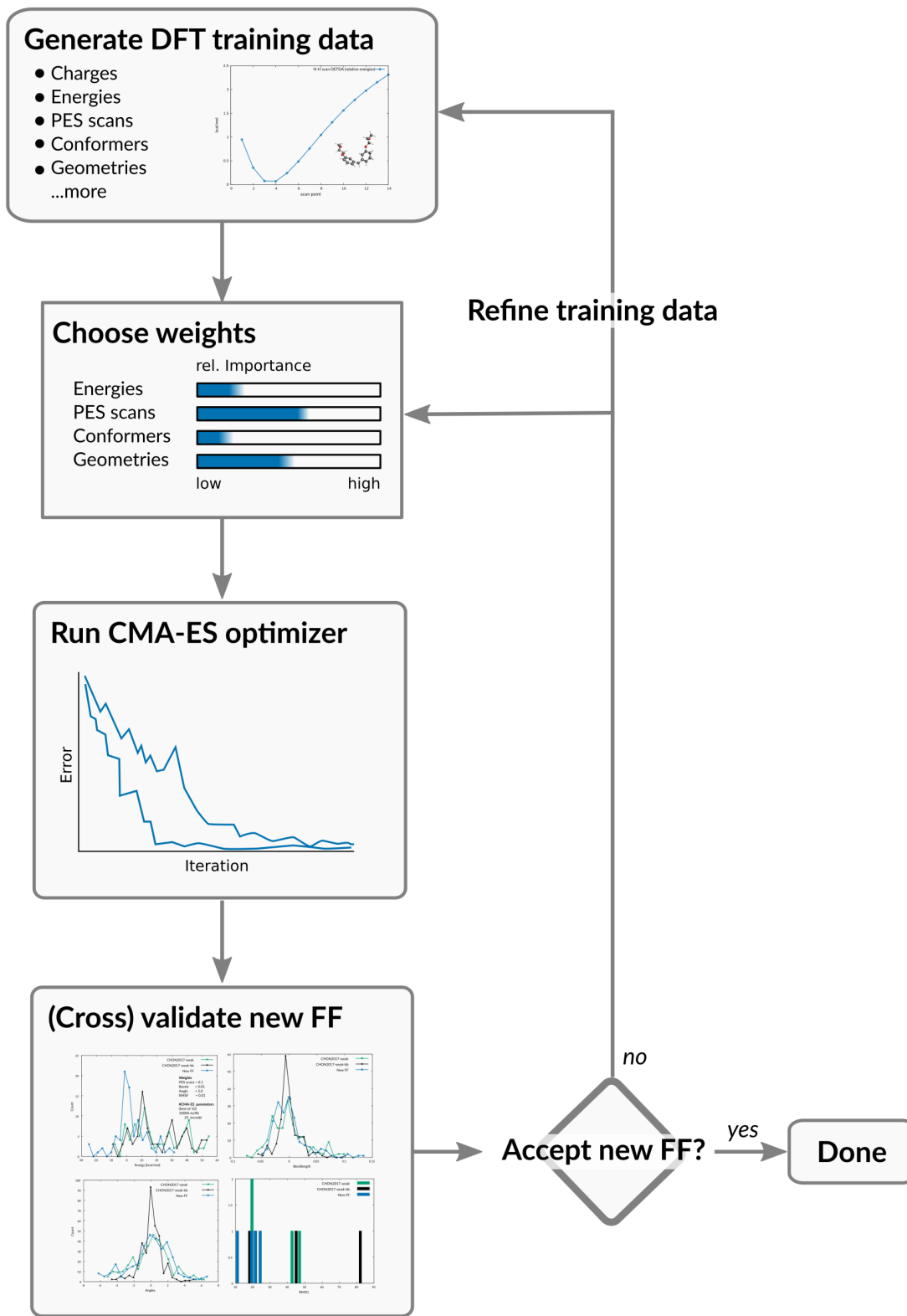
CHFe.ff: (C/H/Fe) Md Mahbubul Islam, Chenyu Zou, Adri C. T. van Duin and Sumathy Raman *Interactions of hydrogen with the iron and iron carbide interfaces: a ReaxFF molecular dynamics study* Phys. Chem. Chem. Phys., 2016, 18, 761-771 (<https://doi.org/10.1039/C5CP06108C>)

REPARAMETRIZING REAXFF WITH THE CMA-ES OPTIMIZER

Important: This tutorial uses the old standalone ReaxFF program. From AMS2022, we recommend that you instead follow the [ParAMS tutorials](#).

14.1 Overview of the workflow

This advanced ReaxFF tutorial will demonstrate how to fit ReaxFF parameters to create a new force field.



The reparametrization workflow consists of the following steps:

- Fit a new force field to the reference data
- **Cross-validate** ([https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))) the new force field
- If needed: refine training set (add entries, adjust weights...) and re-fit, cross-validate,...

The force field optimization algorithm requires the evaluation of an objective function (aka cost function). For ReaxFF the objective function takes the form:

$$\text{Error} = \sum_{i=1}^n \left[\frac{x_{i,ref} - x_{i,ReaxFF}}{\sigma_i} \right]^2$$

where the sum runs over all training set entries. Each difference between the reference property ($x_{i,ref}$) and the ReaxFF value - calculated with your new force field - ($x_{i,ReaxFF}$) is weighted individually via the weightings σ_i . The weightings, σ_i , can be set automatically but will most likely need to be refined manually during the fitting process.

Once the objective function has been set up, the CMA-ES optimizer can be used to fit a set of new parameters. Since the force field optimizers included in AMS, *MCFF* <*MCFFOptimizer*> and *CMA-ES* <*cmaes*>, are stochastic optimization algorithms, it is advised to run them multiple times, possibly starting from different/random starting conditions. The force fields with the lowest overall value of the objective function can then be tested in a cross-validation scheme.

It's advised to validate a newly fitted force field by testing it on a set of references that were *not used during the training* of the force field. This serves as a simple measure against **overfitting** (<https://en.wikipedia.org/wiki/Overfitting>) and provides some direct information on the transferability of the new parameters.

The tutorial is inspired by the work done by A. Vashisth and coworkers on re-fitting an existing ReaxFF force field for usage with the novel bond boost acceleration method :

A. Vashisth, C. Ashraf, W. Zhang, C. E. Bakis, and Adri C. T. van Duin, *J. Phys. Chem. A* 122 (2018) 6633 (<http://dx.doi.org/10.1021/acs.jpca.8b03826>)

Like in the paper, the parameters related to C-O, C-N and N-H in the force field **CHON-2017_weak.ff** are going to be refitted. Both the initial force field and the refitted force field from the paper (**CHON-2017_weak_bb.ff**) are shipped with AMS2019, so you can compare both with the force field you are going to create in this tutorial.

14.1.1 Assigning weights

Some remarks on weights

The force field optimization algorithm requires the evaluation of an objective function. For ReaxFF the objective function takes the form:

$$\text{Error} = \sum_{i=1}^n \left[\frac{x_{i,ref} - x_{i,ReaxFF}}{\sigma_i} \right]^2$$

where the sum runs over all training set entries. Each difference between the reference property ($x_{i,ref}$) and the ReaxFF value - calculated with your new force field - ($x_{i,ReaxFF}$) is weighted individually via the weightings σ_i . The weightings, σ_i , can be set automatically but will most likely need to be refined manually during the fitting process.

If you look at the weights in the above function more closely, you might spot that they can be interpreted in terms of an accuracy or rather a *desired* accuracy as you don't know beforehand which accuracy the fitting process will yield in the end. When coming up with weights this view can guide your initial guesses. However, it should also be kept in mind that the relative values of the weights are important too as discussed in the below example.

Let's assume you have a medium sized organic molecule and you assign a weight of 0.01 to all C-H and all C-O bonds. This scheme will probably not result in a very high accuracy of the C-O bond energies. Why? Simply because there will be much, much more C-H bonds in the system than C-O bonds. Since all errors are going to be summed up, even

small changes in the C-H bond energies will affect the value of the objective function more than a medium change in the C-O bond. The optimizer “sees” only a single number, so it’s important to make sure that the objective function is balanced (hence the usage of the terms *weight*) - or - if it’s biased, it should be biased towards the entries you consider important for your system.

If this left you puzzled, why not take a look at the weights that were used in the published [Co trainingset](#) to gain some further inspiration?

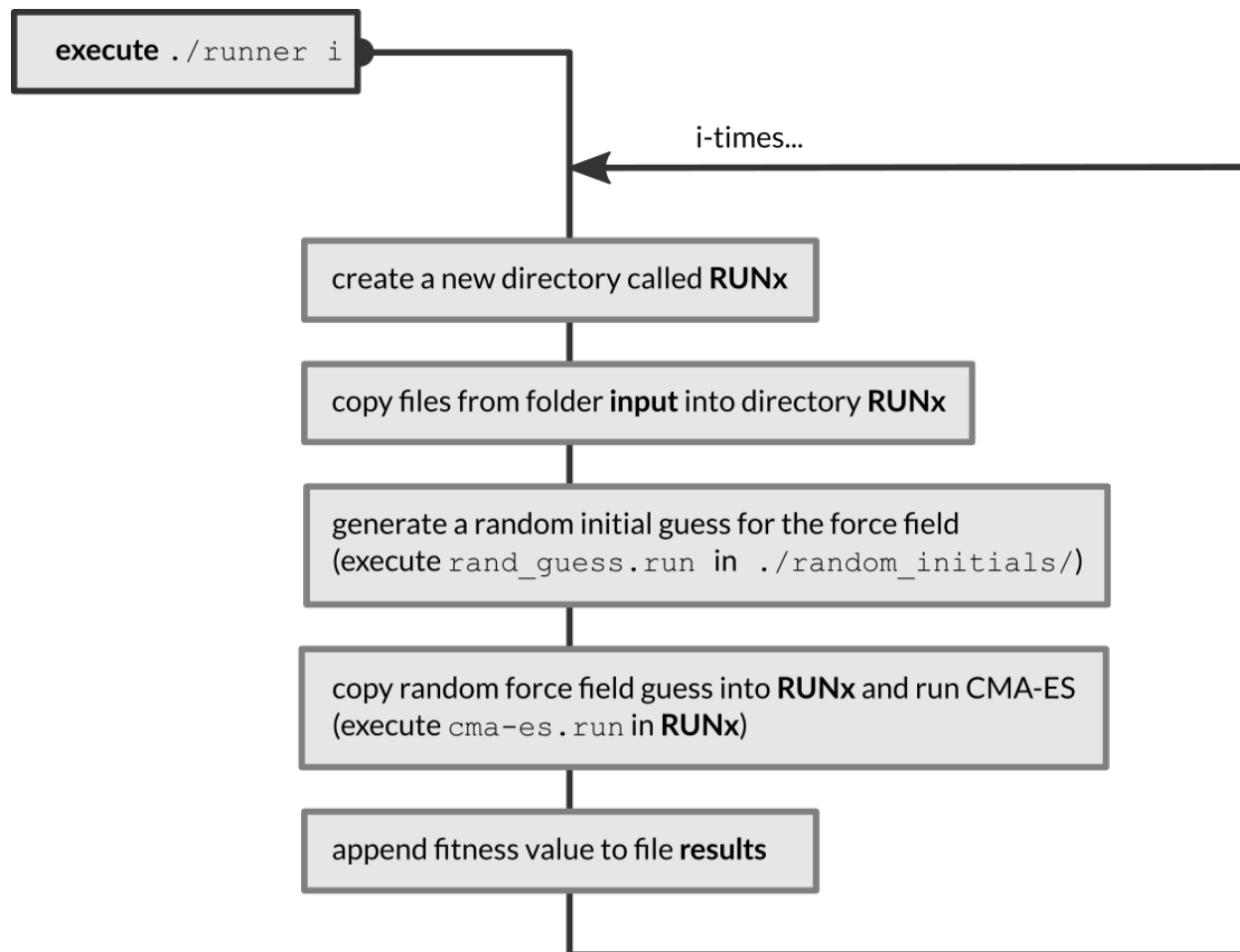
14.2 How to run the optimizer

At the time of writing there exists no GUI support for the CMA-ES optimizer, instead we use the command line to execute the calculations. If you are unfamiliar with using the command line with AMS, then take a look at the [Getting Started](#) page of the scripting docs.

Inside the folder CMA-ES-FIT are three subdirectories, called **error_function**, **input** and **random_initials** as well as a shell script, **runner**, to start the optimization. Before the script can be used, it has to be declared executable:

```
chmod +x runner
```

When run, the shell script will execute the following workflow



The fitness values of the force fields are appended to the results file, located in the same folder as the runner script. The fitted force fields are found in `./RUNx/ffield_best`.

To execute the runner script, provide the number of cma-es runs as an argument. For example, if you want to run 5 CMA-ES runs:

```
./runner 5
```

Per default, the optimizations start from a random initial force field. In case you have a reason to believe that your current force field is already a good starting point, it can make sense to not start from a random guess, but start the CMA-ES optimizations from the force field located in `training_data/ffield` instead. This can be easily done:

```
./runner 5 norand
```

Before starting the optimization process the following files need to be present:

File	Description	In folder
trainset.in, geo	The training set . As exported by AMStrain.	training_data
trainset.in, geo	The validation set . As exported by AMStrain.	error_function
ffield	A ReaxFF force field file. Serves as initial guess.	training_data
params	Defines ranges and active parameters during optimization.	training_data

For the course of this tutorial you can leave the **params** and **ffield** file untouched.

14.3 How to monitor a running optimization

A running CMA-ES calculation can be monitored from a second command line window by executing the command:

```
opt_convergence
```

inside the folder where a CMA-ES calculation is running. This will print a list of fitness values as function of the iteration index to the screen and create a file called `errors.csv` at the same time. The fitness values should decrease with increasing number of iterations.

To visualize the convergence, you can call the `AMSgraphs` module from the command line:

```
amsgraphs errors.csv
```

The CMA-ES run typically reaches somewhat asymptotic behavior within just a couple of thousand steps.

14.4 How to change optimizer settings

The settings for the CMA-ES optimizer are documented in the *CMA-ES manual* `<cmaes>`. In the current setup the control file is created by executing the script `cma-es.run` located in the folder **input**. The file can be viewed with any editor.

The relevant commands are found at the end of the file. For the CMA-ES optimization these are:

```
[...snip...]
#CMA-ES settings
5000      mcffit      Max. Number of iterations.
0.00001   ffotol        Convergence criteria.
50        replic       The CMA-ES sample size.
25        mcxddd       width of the distribution.
           The smaller this value, the more
           random (non-local) the search will be.
0         fort99     Do not write fort.99 files
```

(continues on next page)

(continued from previous page)

```
cat > iopt <<eor
7
eor
```

14.5 How to cross-validate a fitted force field

To cross validate a newly fitted force field, supply the path to the force as console argument to the script **calculate_errors.run** found in the folder **error_function**. For example, to test the force field fitted in the first CMA-ES run, use the following command (from within the folder **error_function**):

```
chmod +x calculate_errors.run
./calculate_errors.run ../RUN1/ffield_best
```

The result of this calculation is a detailed breakdown of the error function, written to a **fort.99** file. Note the root mean square errors reported at the end of the text written to the commandline:

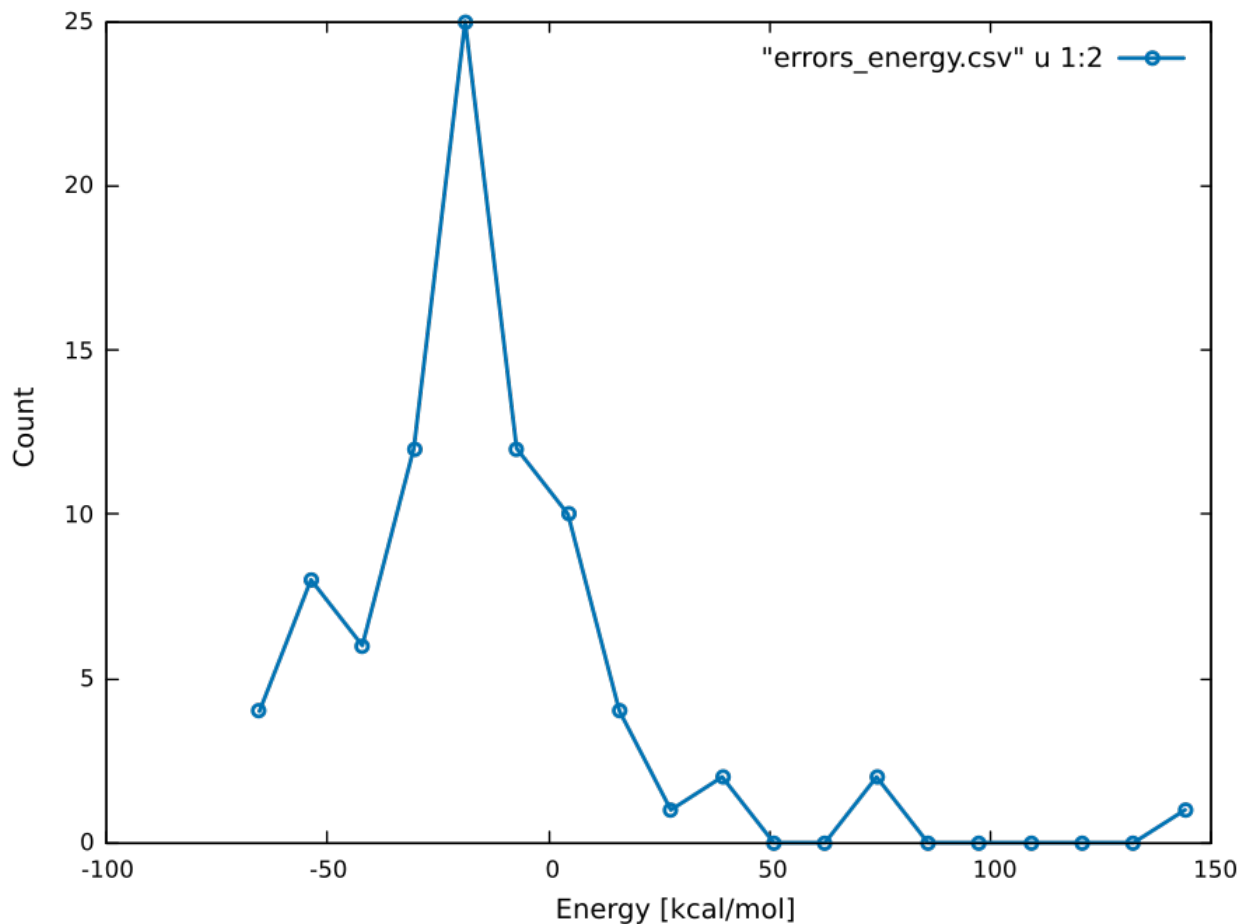
```
RMSD (Charge) : 0.1448 (errors_charge.csv)
RMSD (Bond) : 8.5617 (errors_bond.csv)
RMSD (RMSG) : 10.9749 (errors_rmsg.csv)
RMSD (Energy) : 14.8500 (errors_energy.csv)
```

further the differences between predicted and target values are binned and written to error files (filenames are given in parentheses, behind the RMSD) that can be opened with AMSgraphs.

For example:

```
amsgraphs errors_energies.csv
```

will show a distribution of the relative error in the energies prediction of your new force field



14.6 Running the optimizer

Begin with just two CMA-ES optimization runs, using the default CMA-ES settings:

```
./runner 2 norand
```

Note that we are just running two optimizations since the optimization can take a long time, depending on your hardware and training set. Inside the command line window you can always end a running calculation by pressing CTRL+C. And finally, since we are trying to re-fit an existing force field we can assume the existing force field is a good starting guess which is why the optimization is run **without a random starting guess**.

14.6.1 Errors and Cross-validation

Once the calculation has finished, take a look at the results file to find out which optimization run yielded the best fitness value. Keep in mind that the outcome of the optimization runs are non-deterministic and hence your results will differ from the below, but the overall trend should be similar.

For example, a run with 4 independent CMA-ES optimizations, yielded the following results

```
/home/ole/Workspace/CMA-ES-FIT/RUN1
xbestever found after 1314 evaluations, function value 1.13854e+08
/home/ole/Workspace/CMA-ES-FIT/RUN2
```

(continues on next page)

(continued from previous page)

```
xbestever found after 1116 evaluations, function value 4.77308e+07
/home/ole/Workspace/CMA-ES-FIT/RUN3
xbestever found after 1314 evaluations, function value 5.85587e+07
/home/ole/Workspace/CMA-ES-FIT/RUN4
xbestever found after 1260 evaluations, function value 7.06017e+08
```

So in this case CMA-ES run #2 yielded the best overall fitness value. Let's see how well this force field does for the validation test set. From inside the folder error_function run the following command to generate a **fort.99** file:

```
./calculate_errors.run ../RUN2/ffield_best
```

Next, inspect the resulting RMSD errors:

```
RMSD (Bond) : 0.0673 (errors_bond.csv)
RMSD (angle) : 3.4729 (errors_angle.csv)
RMSD (RMSG) : 33.6747 (errors_rmsg.csv)
RMSD (Energy) : 10.9002 (errors_energy.csv)
```

These errors reflect the weighting in the training set. By assigning a weight of 0.01 to the bond scans in the training data we have put quite some emphasis on the energies and thus it is not surprising that these are reproduced and predicted fairly well. The fit results become even more obvious when comparing with the two force fields CHON2017_weak.ff (our starting guess!) and a refitted version of our starting guess, CHON2017_weak_bb.ff (refitted for usage with the bond boost method).

To calculate the objective function for these force fields, run:

```
./calculate_errors.run $AMSHOME/atomicdata/ForceFields/ReaxFF/CHON2017_weak.ff
```

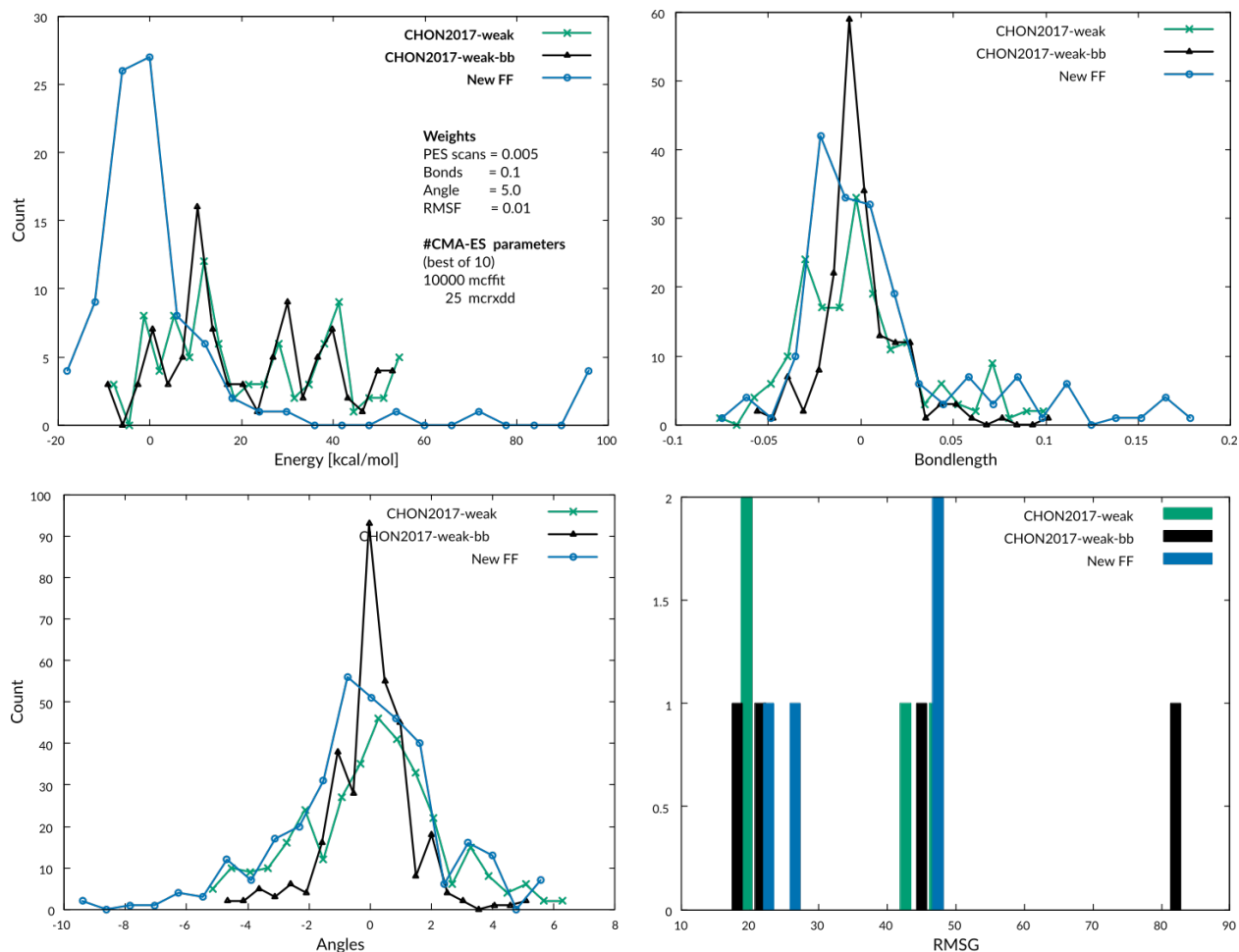
for the force field we used as a starting guess, and

```
./calculate_errors.run $AMSHOME/atomicdata/ForceFields/ReaxFF/
CHON2017_weak_bb.ff
```

for a force field that was refitted from our starting guess to proper DFT training set. If compare against the refitted force field:

Value in training set	New FF	CHON2017_weak_bb.ff
Bonds	0.0673	0.0212
Angles	3.4729	1.3347
RMSG	33.6747	49.4250
Energy	10.9002	26.7171

When comparing the error files generated by analysis of the errors you see that the geometries seem to be much better described by the CHON2017_weak_bb.ff. This is due to the fact that we put so much emphasis on the PES scans in our training set.



14.6.2 Refine the training set

Let's try to adjust the weights in our training data to get better geometries:

- Increase the number of steps in the geometry optimizations to 15
- Assign a weight of 0.01 to every bond distance in the training.in
- Assign a weight of 0.1 to all the energies in trainset.in

and run the CMA-ES optimizer again, followed by a calculation of the objective function for the best force field of this run. The new results should look similar to these finding

Value in training set	New FF	CHON2017_weak_bb.ff
Bonds	0.0275	0.0212
Angles	1.7123	1.3347
RMSG	28.0047	49.4250
Energy	27.4686	26.7171

The resulting force field is already much closer to the published force field. Again, taking closer look at the error files with AMSgraphs reveals that this weighting scheme seems to result in a much more balanced force field

Keep in mind that the original force field - of which we have re-used quite some parameters - was fitted against DFT data. In other words, the force field has been trained to reproduce DFT geometries and DFT energies. This means

deviations between our DFTB force field and the ‘real’ force fields are to be expected since the comparison isn’t really fair. After all this approach serves for illustrating the concepts and workflow of parameter fitting and not the generation of a production ready force field. If you want to optimize your force field even further, try some of the following:

Training set / Optimization tweaks:

- More tests: Apply new force field to MD and geometry optimizations
- Try to locate problematic trainset entries and change (decrease) their weights
- Increase max. steps for the geometry optimizer
- Add more entries
- Run more (~10) CMA-ES runs
- Check params file in folder input, maybe more parameters need to be fitted?
- Check parameter ranges in params file: should a range be increased (compare to your fitted FF)?

APPENDICES

15.1 The Force Field File

Each force field file consist of following sections:

Section name	N of params	N of header lines	N of block keys	block keys
General	41	1		
Atoms	32	4	1	atom type name
Bonds	16	2	2	atom type index
Off-diagonal	6	1	2	atom type index
Angles	7	1	3	atom type index
Torsions	7	1	4	atom type index
Hydrogen bonds	4	1	2	atom type index

15.1.1 Format

The force field file begins with a description line that, in turn, may optionally begin with a list of keywords between square brackets, for example (for an ACKS2+eReaxff force-field):

```
[ ereaxff acks2 ] Reactive MD-force field for Ethylene Carbonate and Li
```

Each section starts with one or more header line containing, on the first line, the number of blocks in the section, possibly followed by description of the parameters. The number of header lines is supposed to match the number lines in a block of the corresponding section (4 in atoms, 2 in bonds and 1 in every other section). The additional header lines after the first are skipped when reading the force field file.

General parameters

The header of this section starts with *npar*, the number of general parameters present in the force field file. The header is followed by *npar* lines each containing a parameter value followed by a comment, for example:

```
39      ! Number of general parameters  
50.0000 !Overcoordination parameter
```

Atoms

The atomic parameters section starts with the number of atom types present in the force field, followed by three additional header lines and the blocks of parameters, one block per atom type. Each block consists of 4 lines starting with a line containing the atom name and 8 parameter values with the (1x,a2,8f9.4) format followed by three lines with 8 parameter values each, with the (3x,8f9.4) format, for example:

```

3      ! Nr of atoms; cov.r; valency;a.m;Rvdw;Evdw;gammaEEM;cov.r2;#
      alfa;gammavdW;valency;Eunder;Eover;chiEEM;etaEEM;n.u.
      cov r3;Elp;Heat inc.;n.u.;n.u.;n.u.;n.u.
      ov/un;vall;n.u.;val3,vval4
C     1.3817   4.0000  12.0000   1.8903   0.1838   0.   9000   1.1341   4.0000
      9.7559   2.1346   4.0000  34.9350  79.5548   5.   9666   7.0000   0.0000
      1.2114   0.0000 202.5551   8.9539  34.9289 13.   5366   0.8563   0.0000
      -2.8983   2.5000   1.0564   4.0000   2.9663   0.   0000   0.0000   0.0000
H     0.7853   1.0000   1.0080   1.5904   0.0419   1.0206 -0.1000   1.0000
      9.3557   5.0518   1.0000   0.0000 121.1250  5.3200   7.4366   1.0000
      -0.1000   0.0000  62.4879   1.9771   3.3517   0.7571   1.0698   0.0000
      -15.7683  2.1488   1.0338   1.0000   2.8793   0.0000   0.0000   0.0000
... one more block ...

```

Bonds, angles, etc.

In the remaining sections, a block key consists of two or more integer numbers, each of them referring to the atomic block with this index. The number of integers in the key depends on the block type (two for bonds, three for valence angles, etc.). For instance, the bond parameters block below corresponds to the C-H bond for the atoms block shown above.

The bond parameters section starts with the number of bond types followed by one additional comment line. The first line of the block has the (2i3,8f9.4) format and the second (6x,8f9.4):

```

6      ! Nr of bonds; Edis1; LPpen;n.u.;pbe1;pbo5;13corr;pbo6
      pbe2;pbo3;pbo4;n.u.;pbo1;pbo2;ovc orr
1  1 156.5953 100.0397 80.0000 -0.8157 -0.4591  1.0000 37.7369  0.4235
      0.4527 -0.1000  9.2605  1.0000 -0.0750  6.8316  1.0000  0.0000
... five more blocks ...

```

For the rest of the sections, the format remains similar to the bonds section, except that they do not have additional header lines and the Fortran format may be slightly different: (2i3,6f9.4) for off-diagonal, (3i3,7f9.4) for valence angles, (4i3,7f9.4) for torsion angles, and (3i3,4f9.4) for hydrogen bonds.

15.1.2 Equation Reference

In the tables below the ReaxFF parameters are listed with their corresponding equation numbers from the SCM developer notes which have mostly technical relevance. For a good introduction to the meaning of ReaxFF parameters we advise the initial ReaxFF publication *ReaxFF: A Reactive Force Field for Hydrocarbons*, A.C.T. van Duin, S. Dasgupta, F. Lorant, W.A. Goddard, *J. Phys. Chem. A*. 2001 105 41 9396-9409. (<https://doi.org/10.1021/jp004368u>)

General

Of particular interest are the upper taper radius parameter (#13), which describes the non-bonded cutoff radius, and the bond order cutoff (#30), which describes the bond order threshold, above which atoms are considered connected. Both these parameters may have a major impact on the ReaxFF calculation speed; decreasing the taper radius or increasing the bond order cutoff can make ReaxFF run considerably faster. These parameters, however, have a significant impact on the force description and should not be changed without re-parameterization of other parts of the force field.

Index	Name in Eq	Equation	Comment
1	p_boc1	4c	Overcoordination parameter
2	p_boc2	4d	Overcoordination parameter
3	-p_coa2	15	Valency angle conjugation parameter
4	p_trip4	20	Triple bond stabilization parameter
5	p_trip3	20	Triple bond stabilization parameter

Continued on next page

Table 15.1 – continued from previous page

6	k_c2	19	C2-correction
7	p_ovun6	12	Undercoordination parameter
8	p_trip2	20	Triple bond stabilization parameter
9	p_ovun7	12	Undercoordination parameter
10	p_ovun8	12	Undercoordination parameter
11	p_trip1	20	Triple bond stabilization energy
12	n/a	21	Lower Taper-radius
13	R_cut	21	Upper Taper-radius
14	p_fe1	6a	Fe dimer correction
15	p_val6	13c	Valency undercoordination
16	p_lp1	8	Valency angle/lone pair parameter
17	p_val9	13f	Valency angle parameter
18	p_val10	13g	Valency angle parameter
19	p_fe2	6a	Fe dimer correction
20	p_pen2	14a	Double bond/angle parameter
21	p_pen3	14b	Double bond/angle parameter: overcoord
22	p_pen4	14b	Double bond/angle parameter: overcoord
23	p_fe3	6a	Fe dimer correction
24	p_tor2	16b	Torsion/BO parameter
25	p_tor3	16c	Torsion overcoordination
26	p_tor4	16c	Torsion overcoordination
27	p_elho	26a	eReaxFF
28	p_cot2	17b	Conjugation
29	p_vdW1	23b	VdW shielding
30	cutoff * 100	3a,b	Cutoff for bond order (* 100)
31	p_coa4	15	Valency angle conjugation parameter
32	p_ovun4	11b	Overcoordination parameter
33	p_ovun3	11b	Overcoordination parameter
34	p_val8	13d	Valency/lone pair parameter
35	X_soft	25	ACKS2 softness parameter
36	unused	n/a	n/a
37	p_val	27 via n_el	eReaxFF
38	n/a	13d	if 1: remove delta_j term for non-C-C-C angles and where none of the atoms is N
39	p_coa3	15	Valency angle conjugation parameter
40	n/a	20	Condition to turn triple bond option: vpar(40) == 1
41	n/a	26 via Tap(R)	eReax-specific taper radius for interactions with/between electrons and holes

Atoms

If negative values are provided to either of the three bond radii (**sigma**, **pi**, and **double pi**) the bond order contributions are ignored for that atom.

Index	Name in Eq	Equation	Comment
1	r_0^sigma	2	Sigma bond covalent radius
2	Val_i	3a, 4b, 5, 9a	Valency
3	n/a	9a	Atomic mass
4	r_vdW	23a	van der Waals radius
5	D_ij	23a	van der Waals dissociation energy

Continued on next page

Table 15.2 – continued from previous page

6	gamma_i	24	gammaEEM; EEM shielding
7	r_0^pi	2	Pi bond covalent radius
8	Val_i^e	7, 8, 9	Number of valence electrons
9	alpha_ij	23b	van der Waals parameter
10	1/gamma_w	23b	van der Waals shielding
11	Val_j^angle	16c, 13c	Valency for 1,3-BO correction
12	p_ovun5	12	Undercoordination energy
13	p_i^xel2	26	eReaxFF, atom type parameter
14	chi_i	24, 25	EEM electronegativity
15	eta_i	24, 25	EEM hardness
16	n/a	n/a	Donor or acceptor switch in H-bonds
17	r_0^pi	2	Double pi bond covalent radius
18	p_lp2	10	Lone pair energy
19	n/a	n/a	Atomic heat of formation
20	p_boc4	4e,f	Bond order correction
21	p_boc3	4e,f	Bond order correction
22	p_boc5	4e,f	Bond order correction
23	C_i	25	Atomic softness cutoff parameter
24	alpha, alpha_i	26, 26a	eReaxFF, constant, dependent on atom type
25	p_ovun2	12	Valence angle parameter
26	p_val3	13b -> 13a	Valence angle parameter
27	beta, beta_i	26a	eReaxFF, constant, dependent on atom type
28	Val_i'^boc	3b	Number of lone pairs
29	p_val5	13b	Valence angle parameter
30	p_c1	23c	Inner wall vdW repulsion parameter
31	p_c2	23c	Inner wall vdW repulsion parameter
32	p_c3	23c	Inner wall vdW repulsion parameter
33	C_i	23d	Lg dispersion parameter
34	R_eij	23d	VdW Radius for Lg dispersion correction

Bonds

1	D_e^sigma	6, 11a	Sigma-bond dissociation energy
2	D_e^pi	6	Pi-bond dissociation energy
3	D_e^pipi	6	Double pi-bond dissociation energy
4	p_be1	6	Bond energy parameter
5	p_bo5	2	Double pi bond parameter
6	Val'_i^boc	3b	1,3-Bond order correction
7	p_bo6	2	Double pi bond order
8	p_ovun1	11a	Overcoordination penalty
9	p_be2	6	Bond energy parameter
10	p_bo3	2	Pi bond order parameter
11	p_bo4	2	Pi bond order parameter
12	unused	n/a	n/a
13	p_bo1	2	Sigma bond order
14	p_bo2	2	Sigma bond order
15	delta'_i	3a	Uncorrected BO overcoordination
16	p_ij^xel1	27	e ReaxFF param; for adjusting number of electrons available to host atom

Off-diagonal

This section allows for the definition of off-diagonal values for both bond order and van der Waals pair interactions. By default, ReaxFF calculates these terms from the combination rules and the atom parameters (i.e. the default C-H van der Waals radius is $(R_{vdW}[C]*R_{vdW}[H])^{0.5}$), but the off-diagonal section allows for the definition of different values. Any value given in the off-diagonal section overrules that obtained from the combination rules.

1	D_ij	23a	VdW energy
2	r_vdW	23a	VdW radius
3	alpha_ij	23a	VdW parameter
4	r_0^sigma	2	Sigma bond length
5	r_0^pi	2	Pi bond length
6	r_0^pipi	2	PiPi bond length
7	C_i, C_lg,ij	23d	Lg dispersion parameter

Angles

1	Theta_0,0	13g	180o-(equilibrium angle)
2	p_val1	13a	Valence angle parameter
3	p_val2	13a	Valence angle parameter
4	p_coa1	15	Valence conjugation
5	p_val7	13c	Undercoordination
6	p_pen1	14b -> 14a	Penalty energy
7	p_val4	13b	Valence angle parameter

Torsions

1	V_1	16a	V1-torsion barrier
2	V_2	16a	V2-torsion barrier
3	V_3	16a	V3-torsion barrier
4	p_tor1	16a	Torsion angle parameter
5	p_cot1	17a	Conjugation energy
6	unused	n/a	n/a
7	unused	n/a	n/a

Hydrogen bonds

1	r_hb^0	18	Hydrogen bond equilibrium distance
2	p_hb1	18	Hydrogen bond energy
3	-p_hb2	18	Hydrogen bond/bond order
4	-p_hb3	18	Hydrogen bond parameter

15.2 trainset.in

15.2.1 Description of the trainset.in file

The trainset.in file contains the training set data and tells the program how to calculate the cost function $F = \sum((y_i - y_i^{ref})/acc_i)^2$, which can be used to optimize the *force field* (page 87) parameters. The **trainset.in** uses molecule identifiers, or keys, defined in the DESCRP field of the geo file (in BGF format), or in the models.in file, to compare force field derived geometries and energy differences to the reference values. The **trainset.in** has a free format as far as numbers concerned, although it does require that fields are space-separated. Besides, the “-”, “+” and “/” symbols

have a special meaning in the **trainset.in** file and should not be used in identifiers. The **trainset.in** file is divided into 5 sections listed below. Each section begins with a start keyword and ends with the corresponding end keyword. The words in “CELL PARAMETERS” and “ENDCELL PARAMETERS” must be separated by exactly one space.

15.2.2 Sections format

Block name	Start keyword	End keyword	Format
charges	CHARGE	ENDCHARGE	Key Acc Atom Ref
geometries	GEOMETRY	ENDGEOMETRY	Key Acc [Atom1 [Atom2 [Atom3 [Atom4]]] Ref]
forces	FORCES	ENDFORCES	Key Acc Atom Ref
cell parameters	CELL PARAMETERS	ENDCELL PARAMETERS	Key Acc Type Ref
energy differences	ENERGY	ENDENERGY	Acc [+/-] Key1/n1 ... [+/-] Key5/n5 Ref
heat of formation	HEATFO	ENDHEATFO	Key Acc Ref

“**Key**” is the molecule name from the geo file. “**Atom**” is an atom index in the corresponding molecule. “**Acc**” is a value of the target accuracy desired for the given error function contribution. This value is often called “**weight**” although in practice it is 1/weight. “**Ref**” is the reference value.

15.2.3 Format description

In the all sections except “ENERGY” each data line starts with the structure identifier (the **Key**), followed by the “**Acc**” of the data point. This is followed by a type identifier. Each section contains following data entries:

- **CHARGE** In the CHARGE section the type identifier is the number of the atom in the molecule and the reference value is its charge. Example:

```
CHARGE
#Key      Acc Atom  Ref
chexane  0.1     1    -0.15
ENDCHARGE
```

- **GEOMETRY** In the GEOMETRY section the type ID is the list of atoms defining an internal coordinate (two for an interatomic distance, three for a valence and four for a torsion angle). When there is only one atom index specified, then the Euclidian distance for the given atom between the two geometries is calculated. When the index is -1 then an average Euclidian distance quantity between the two geometries is used instead. Please note that any reference value different from zero for the Euclidian distances does not make much sense. Besides, since these distances are computed in the Cartesian coordinates, which means that a simple translation of the molecule as a result of energy minimization may result in large Euclidian distances for otherwise similar geometries. If there is no identifier provided then it means that the ReaxFF RMS force will be compared with the reference (which should probably be zero in most cases). Example:

```
GEOMETRY
#Key      Acc At1 At2 At3 At4  Ref
chexane  0.01   1      0.0   # Euclidian distance between atom in_
↳the reference and the trial structure
chexane  0.01  -1      0.0   # Average Euclidian distance between_
↳atoms in the two structures
chexane  0.01   1  2    1.5   # Interatomic distance
chexane  1.00   1  2  3   120.0 # Valence angle
```

(continues on next page)

(continued from previous page)

```

chexane 1.00 1 2 3 4 180.0 # Torsion angle
chexane 1.00                0.0  # RMS force
ENDGEOMETRY

```

- **CELL PARAMETERS** In the CELL PARAMETERS section the type IDs are names of the corresponding lattice parameters. Example:

```

CELL PARAMETERS
#Key      Acc   Type   Ref
chex_cryst 0.01  a     11.20
chex_cryst 0.01  b     11.20
chex_cryst 0.01  c     11.20
chex_cryst 0.01  alpha 90.00
chex_cryst 0.01  beta  90.00
chex_cryst 0.01  gamma 90.00
ENDCELL PARAMETERS

```

- **HEATFO** The HEATFO section does not require a type ID as compares the ReaxFF heat of formation with the reference value. Example:

```

HEATFO
#Key      Acc   Ref
methane  2.00 -17.80
ENDHEATFO

```

- **ENERGY** This section allows comparison of ReaxFF energy differences between structures to the reference data. In this case, each data line starts with the **Acc** of the data point, followed by up to five operator/identifier/divider parts and finishes with the reference value. The operator is either '+' or '-' ('+' is the default). The energy associated with the identifier is divided by the divider, allowing comparison of condensed structures to monomers. The '/' character in the ENERGY section data lines is optional. Example:

```

ENERGY
#Acc op1 Key1 n1 op2 Key2 n2      DeltaE
1.5  + butbenz/1 - butbenz_a/1 -90.00
1.5  + butbenz/1 - butbenz_b/1 -71.00
1.5  + butbenz/1 - butbenz_c/1 -78.00
ENDENERGY

```


REQUIRED CITATIONS

When you publish results in the scientific literature that were obtained with programs of the ADF package, you are required to include references to the program package with the appropriate release number, and a few key publications. In addition to these general references, references to special features are mandatory, in case you have used them.

16.1 General References

The ReaxFF software that SCM makes available is based on the ReaxFF program developed by Adri van Duin.

For calculations with ReaxFF:

1. 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) (<https://doi.org/10.1021/jp004368u>)
2. K. 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) (<https://doi.org/10.1021/jp709896w>)
3. ReaxFF 2022.1, SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands, <http://www.scm.com> Optionally, you may add the following list of authors and contributors: A.C.T. van Duin, W.A. Goddard, M.M. Islam, H. van Schoot, T. Trnka, A.L. Yakovlev

The ReaxFF GUI (ReaxFFinput and AMSmovie) has been developed within SCM (with O. Visser as primary developer).

The ReaxFF program has been parallelized, optimized, and extended by SCM (with A.L. Yakovlev as primary developer).

If you use one of the included force fields you must also add the *proper reference for it* (page 49).

Many examples of ReaxFF applications can be found on Prof. van Duin's publication list (<http://www.engr.psu.edu/adri/Publications.aspx>).

16.2 Feature References

When you have used force fields or special features, you should include the reference(s) to the implementation.

Collective Variable-Driven Hyperdynamics (CVHD) K.M. Bal and E.C. Neyts, *Merging Metadynamics into Hyperdynamics: Accelerated Molecular Simulations Reaching Time Scales from Microseconds to Seconds*, *J. Chem. Theory Comput.*, 11, 4545-4554 (2015) (<https://doi.org/10.1021/acs.jctc.5b00597>)

Force-bias Monte Carlo (fbMC) M.J. Mees, G. Pourtois, E.C. Neyts, B.J. Thijsse, A. Stesmans, *Uniform-acceptance force-bias Monte Carlo method with time scale to study solid-state diffusion*, *Physical Review B* **85**, 134301 (2012) (<https://doi.org/10.1103/PhysRevB.85.134301>)

Grand Canonical Monte Carlo (GCMC) Th.P. Senftle, R.J. Meyer, M.J. Janik and A.C.T. van Duin, *Development of a ReaxFF potential for Pd/O and application to palladium oxide formation*, *J. Chem. Phys.* **139**, 044109 (2013) (<https://doi.org/10.1063/1.4815820>)

Th.P. Senftle, A.C.T. van Duin, M.J. Janik, *Determining in situ phases of a nanoparticle catalyst via grand canonical Monte Carlo simulations with the ReaxFF potential*, *Catalysis Communications* ****volume 52****, 5 July 2014, Pages 72–77 (<https://doi.org/10.1016/j.catcom.2013.12.001>)

Monte-Carlo force-field optimizer (MCFFOptimizer)

E. Iype, M. Huetter, A.P.J. Jansen, S.V. Nedeia, C.C.M. Rindt, *Parameterization of a Reactive Force Field Using a Monte Carlo Algorithm*, *J. Comp. Chem.* **34**, 1143-1154 (2013) (<https://doi.org/10.1002/jcc.23246>)

ACKS2

T. Verstraelen, P.W. Ayers, V. Van Speybroeck, M. Waroquier, *ACKS2: Atom-Condensed Kohn-Sham DFT Approximated to Second Order* *J. Chem. Phys.* **138**, 074108 (2013) (<https://doi.org/10.1063/1.4791569>)

eReaxFF M.M. Islam, G. Kolesov, T. Verstraelen, E. Kaxiras, A.C.T. van Duin, *eReaxFF: A Pseudoclassical Treatment of Explicit Electrons within Reactive Force Field Simulations*, *J. Chem.Theory Comput* **12**, 3463 (2016) (<https://doi.org/10.1021/acs.jctc.6b00432>)

ReaxFF Force Field References When you publish results in the scientific literature that were obtained with one of the included force fields for ReaxFF, including the proper reference for the force field used is mandatory.

- *Forcefields included in the latest release* (page 49)
- *Forcefields included in the development snapshots* (page 63)

Reaction Event Detection (ChemTraYzer) M. Döntgen, M.-D. Przybylski-Freund, L.C. Kröger, W.A. Kopp, A.E. Ismail, K. Leonhard *Automated Discovery of Reaction Pathways, Rate Constants, and Transition States Using Reactive Molecular Dynamics Simulations* (<https://doi.org/10.1021/acs.jctc.5b00201>) *J. Chem.Theory Comput.*, **11** (6), 2517–2524, 2015

16.3 External programs and Libraries

[Click here](#) for the list of programs and/or libraries used in the ADF package. On some platforms optimized libraries have been used and/or vendor specific MPI implementations.