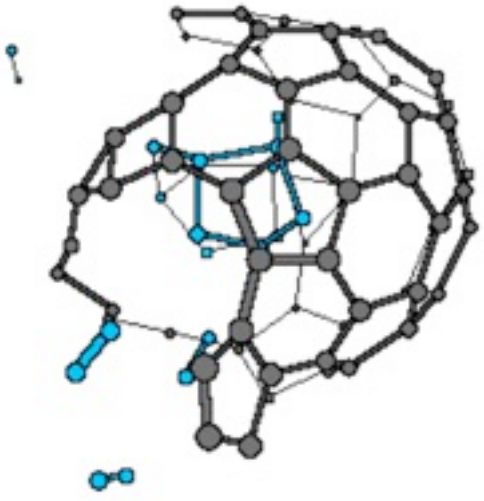
ReaxFF User Manual

Written by Adri van Duin,
Original December 2002
Updated and expanded March 2013
Expanded June 2017 (e-ReaxFF)

Only relevant for the Standalone, single-processor, fortran-77 ReaxFF program, as written by Adri van Duin, 1999-2013, at the University of Newcastle upon Tyne (1999-2002), the California Institute of Technology (2002-2008) and the Pennsylvania State University (2008-2017)

E-mail: acv13@psu.edu
Department of Mechanical and Nuclear Engineering
Pennsylvania State University
University Park, PA 16802
USA



Contents

1. General overview

1.1 Concept. ReaxFF was developed to bridge the gap between quantum chemical (QC) and empirical force field (EFF) based computational chemical methods (Figure 1.1). Where QC methods are, in general, applicable to all chemical systems, regardless of connectivity, their computational expense makes them inapplicable for large (say, more than100 atoms) systems. Their computational expense also makes QC methods primarily applicable for single point or local energy minimization; high-temperature molecular dynamics (MD) simulations are generally too time-consuming.

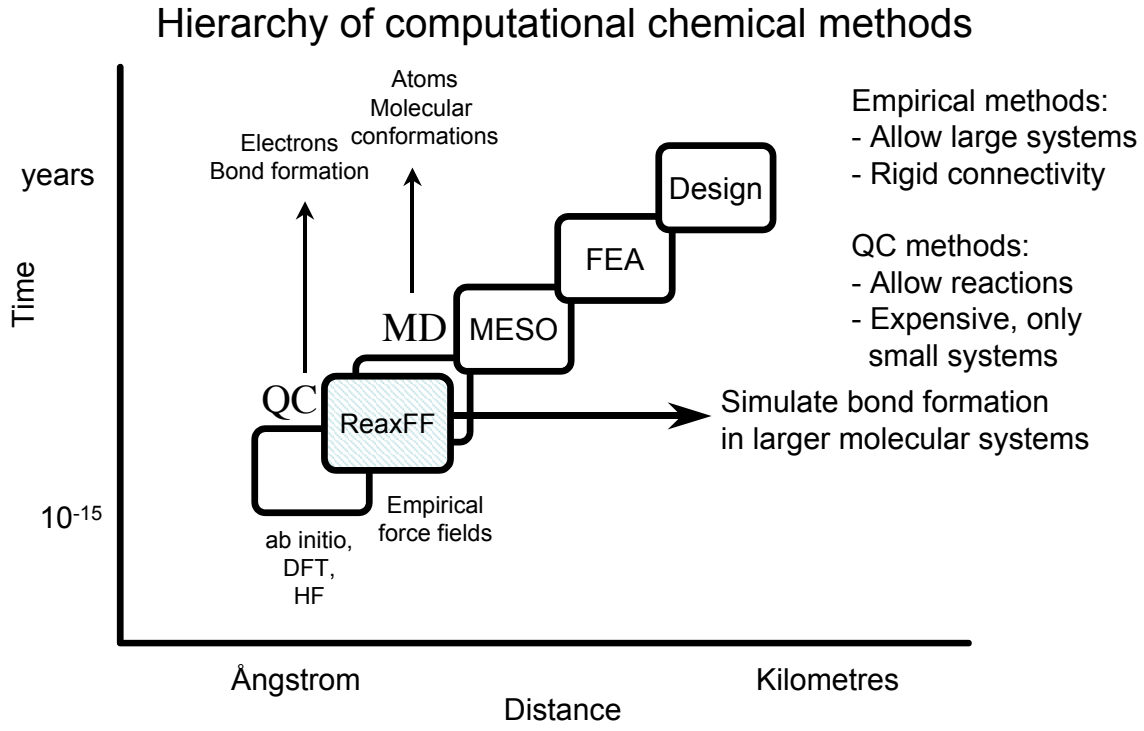


Figure 1.1: Position of ReaxFF in the computational chemical hierarchy.

EFF methods describe the relationship between energy and geometry with a set of relatively simple potential functions. In their most simple form, EFF methods describe molecular or condensed phase systems by simple harmonic equations that describe the stretching and compression of bonds and the bending of bond angles, usually augmented by van der Waals potential functions and Coulomb interactions to describe non-bonded interactions. Their relative simplicity allows EFF methods to be applied to much larger systems than QC systems (thousands of atoms on single processors; millions of atoms on multiprocessors). EFF methods have been very successful in describing physical interactions in and between molecules and condensed phase systems, and EFF methods haven been developed for a wide variety of chemical environments, including hydrocarbons, proteins and many inorganic systems. However, EFF methods are mainly applicable for systems at or around their equilibrium configuration. Due to their empirical nature, EFF methods require that the parameters used in their potential functions are

fitted against a suite of data, which can be gathered from experimental and/or from QC-sources (a.k.a. training set). The force field resulting from this fitting procedure can obviously be no more reliable than the data used in its training set. Furthermore, as the force field describes the system in an empirical rather than fundamental fashion, it should only be applied to systems similar to the ones present in the training set. As such, the quality and diversity of the training set define the transferability of the EFF method. With a few exceptions, current EFF methods are only trained for systems in which the bonds remain within about 75% of their equilibrium value. For this reason, these EFF methods cannot describe reactive systems, and in most cases the shape of the potential functions applied in these methods, like the aforementioned harmonic description of the bond length/bond energy relationship, would make it impossible to find parameter values that accurately describe bond energy towards the dissociation limit.

The concept of bond order/bond energy relation, as first formulated by Tersoff [lit], allows for the construction of EFF methods that can, in principle, handle connectivity changes. This concept was used by Brenner (Brenner, 1990) to construct the REBO-potential, an EFF method for hydrocarbon systems, allowing, for the first time, dynamical simulations of reactions in large (>>100 atoms) systems. Over the years, REBO has enjoyed widespread application, but its transferability is limited as it is based on a relatively small training set and because of its exclusion of all non-bonded interactions.

As with the Brenner potential, a bond order/bond energy relationship lies at the center of the ReaxFF-potential. Bond orders are obtained from interatomic distances (Figure 2) and are continually updated at every MD or energy minimization (MM) iteration, thus allowing for connectivity changes. These bond orders are incorporated in all valence terms (i.e. energy contributions dependent on connectivity, like valence angle and torsion angle energy) ensuring that energies and forces associated with these terms go to zero upon dissociation. Furthermore, ReaxFF describes non-bonded interactions between all atoms, irrespective of connectivity. Excessive short-range repulsive/attractive non-bonded interactions are circumvented by inclusion of a shielding term in the van der Waals and Coulomb interaction. For a more detailed description of the ReaxFF energy description see (van Duin et al., 2001) Chapter 4 of this manual and the recent ReaxFF review by Senftle et al. (Senftle et al., 2016).

Chapter 9 of this manual describes a recent extension of ReaxFF to include explicit electrons and holes (e-ReaxFF) – thus improving the ReaxFF capability to reproduce molecular electron affinities and ionization potentials – specifically enabling applications to reactions at electrochemical interfaces like batteries and fuel cells. This chapter specifically focuses on the practical simulation aspects associated with e-ReaxFF – input and output file structures. For a detailed description of the e-ReaxFF theory – and the affiliated ACKS2 charge calculation method – we refer to (Islam et al., 2016; Islam and van Duin, 2016) for e-ReaxFF and (Verstraelen et al., 2013) for the ACKS2-method charge calculation method.

ReaxFF aims to provide a transferable potential, applicable to a wide range of chemical environments. To ensure its transferability, the following general guidelines have been adopted in its development:

- No discontinuities in energy or forces, even during reactions.

- Each element is described by just one force field atom type. The ReaxFF metal oxide oxygen is described by the same parameters as the ReaxFF oxygen in organic molecules. ReaxFF does not have separate sp^2 and sp^3 atoms for carbon, the method determines the atoms hybridization from its chemical environment.
- No pre-definition of reactive sites is necessary using ReaxFF. Although it is possible to drive reactions using restraints (see Input files section) this is not required; given the right temperature and chemical environment reactions will happen automatically.

1.2 Features. At the moment of writing this manual, the ReaxFF-program supports the following features:

- NVT , NVE and NPT dynamics for molecular and periodic systems. Velocity and system volume scaling are performed using the Berendsen method. Velocity scaling can be performed on the entire system, on individual molecules or on individual atoms. Different temperature regimes can be applied to different parts of the system using the **tregime.in** file. This input file can also be used to increase and decrease system temperature during an MD-simulation and can be used to set op annealing runs. The **vregime.in**-file can be used to compress/expand and shear a periodic system.
- ReaxFF incorporates an external electric field. Using the **eregime.in**-file elaborate electric field regimes can be imposed on the system.
- Steepest descent, conjugate gradient and MD-based minimization methods.
- Numerical optimization of cell parameters. Default setting is for cubic optimization, but a/b/c parameters can be optimized separately. Also, c/a ratios can be varied separately (see sections on control and input geometry files). Also, low-temperature MD/NPT simulations can be used for cell parameter optimization.
- Numerical calculation of second derivatives. For molecular systems, these second derivatives can be used to calculate vibrational frequencies and modes. Using a harmonic approximation to populating these modes, ReaxFF can calculate entropy and related thermodynamic properties (see **thermo.out**-file).
- ReaxFF supports interatomic distance, angle, torsion angle and centre-of-mass restraints. These restraints can be used to drive reactions and can be defined in the **geo**-file (in .bgf-format). Sliding interatomic distance, angle and torsion restraints can be used in MD-simulations.
- ReaxFF can perform simulations on crystal unit cells, keeping track of bonds and valence angles between periodic images of atoms. Currently, this feature cannot be applied to systems requiring torsion angles across periodic boundaries (e.g. carbon crystals). None of the inorganic ReaxFF force fields developed to date have included torsion energy terms, and as such ReaxFF can do unit cell calculations for these systems. ReaxFF can also be used to create supercell structures (**fort.85**-output file).
- ReaxFF contains a force field optimization module. See input and output file sections for a further description.
- ReaxFF generates connection tables (see **fort.7** and **fort.8** output files) and performs a fragment analysis using a bond order cutoff (see **molfra.out**-file).
- Using the optional **addmol.bgf**-file, ReaxFF allows for the addition of a user-specified molecular fragment to the system at regular intervals. The initial position and velocity of this fragment can be fully defined by the user.

- ReaxFF has been developed around the EEM charge derivation method (Mortier et al., 1986), allowing calculation of geometry dependent charge distributions. As default, ReaxFF equilibrates the charges over the entire system, however, it can also equilibrate charges within each molecule or run with fixed charges (see **control**-file and **charges**-file in the input file section).
- ReaxFF generates. bgf, .geo, .xyz, .MOP, z-matrix (for molecules) and .pdf output files and can read .geo, .bgf, .xyz, .pdf and z-matrix files. Trajectories are saved in .xyz-format, with optional velocities. Restart files are generated at user-specified intervals.
- Optionally, ReaxFF can generate elaborate output on user-specified bond order/bond lengths and valency angles. See **control**-file, keyword ianalysis.

1.4 Current force fields. Figure 1.2 shows for which systems ReaxFF has currently been parameterized and at which stage of development these parameters sets are. Note that this figure only indicates the elements that have been visited by ReaxFF – it does not describe the actual materials. For example, while a ReaxFF description is available for Fe-metal and Fe/C/H/O interactions (Zou et al., 2012), we do not currently have parameters for the Fe/B or Fe/Si system available.

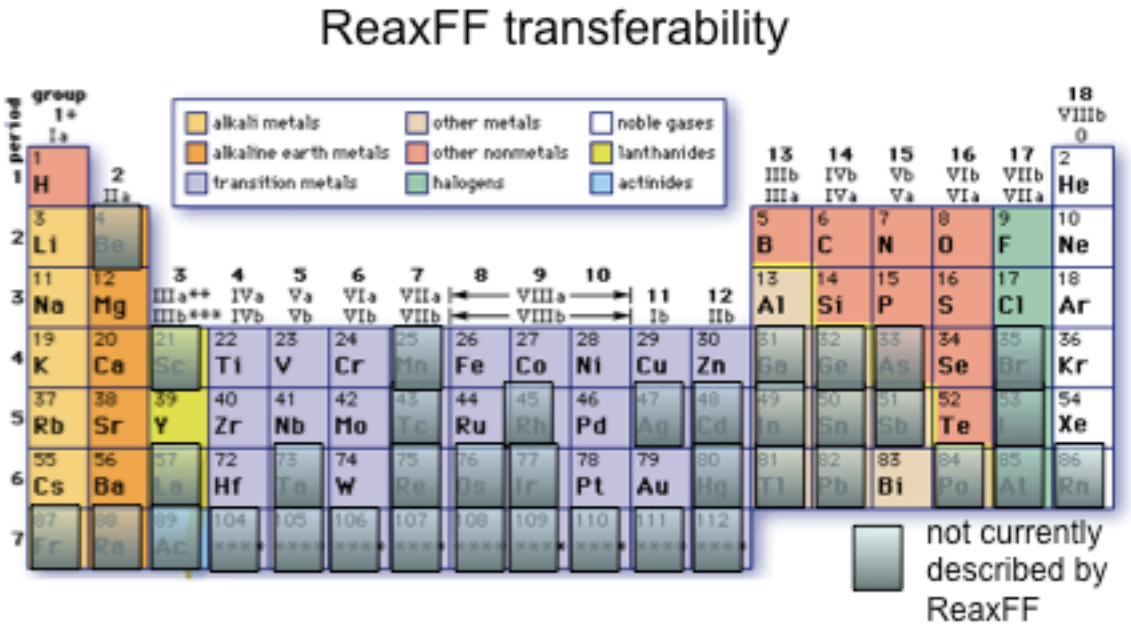


Figure 1.2. Overview of current elements supported by ReaxFF.

Training sets for these force fields primarily, and for many systems exclusively, consist of QC data on clusters and condensed phases. Although some of these reactive potentials have been applied successfully neither of these parameter sets are considered final. Our approach is that during an application we will continuously scrutinize the ReaxFF results by checking against QC data (probably by performing targeted QC-simulations on representative small systems). When major discrepancies occur the QC

data can be added to the appropriate training set and the parameters can be re-optimized. By this continuous communication between QC and ReaxFF we should obtain increasingly reliable and transferable reactive force field descriptions.

The currently available ReaxFF parameter sets have been mainly developed along two major development branch – the water-development branch and the combustion branch. Force field sharing a branch are fully transferable – they share general parameters and, for example, all contain the same first-row element (H/C/N/O) parameters. Figure 1.3 gives and overview of these development branches.

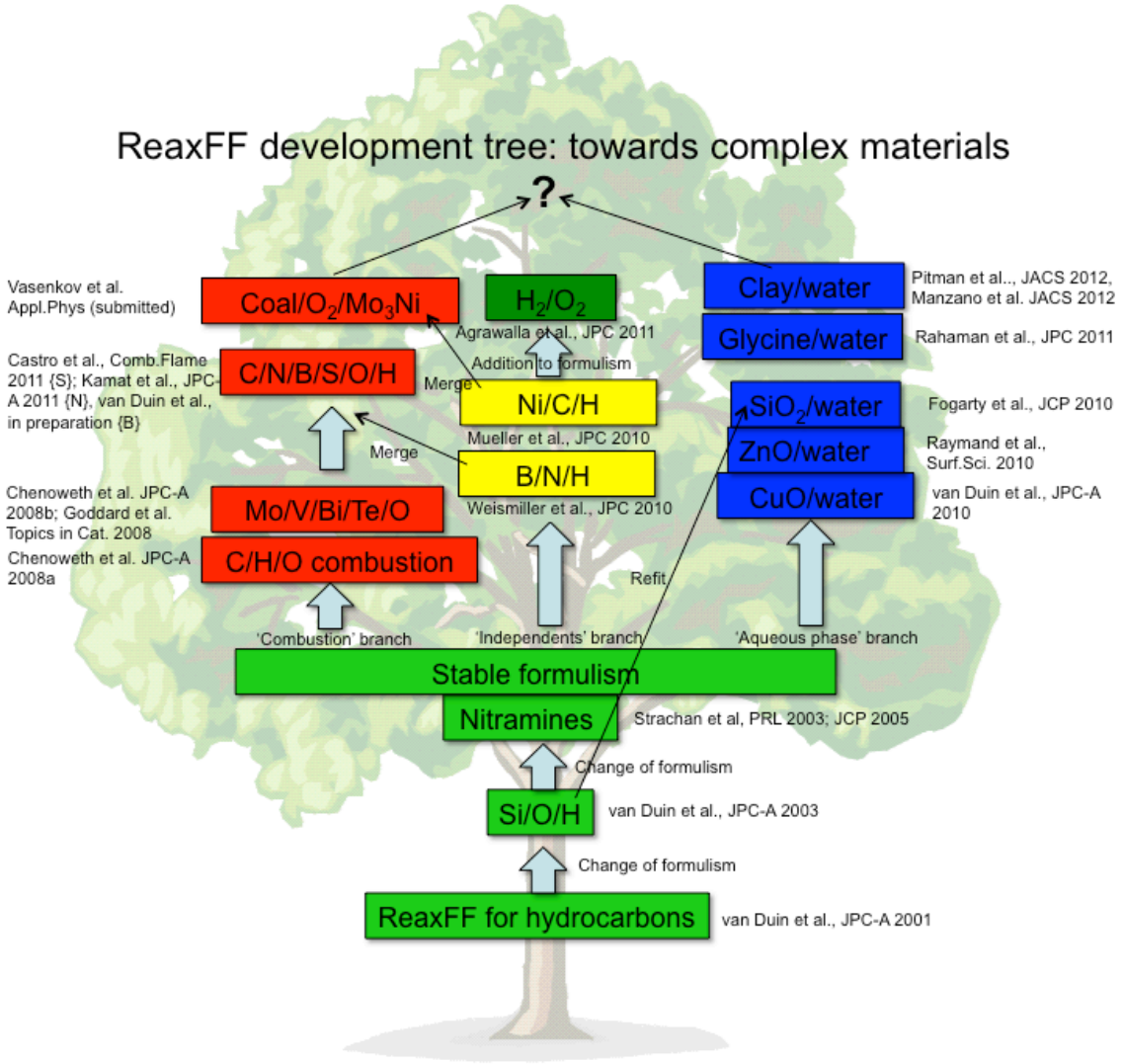


Figure 1.3 Historical overview of ReaxFF parameter development and transferability of published parameter sets on development branches.

Both of the two major development branches have been used for a wide range of materials – the water-branch contains a series of metal oxides, but also a protein and DNA description, while the combustion branch contains all first row elements, as well as a smaller number of oxides. The water-branch has been applied in the following publications: (Abolfath et al., 2011; Aryanpour et al., 2010; Bharati et al., 2012; Fogarty et al., 2010; Gale et al., 2011; Goken et al., 2011; Huang et al., 2013; Jeon et al., 2011; Jeon et al., 2012; Joshi and van Duin, 2013; Kim et al., 2012a; Kim et al., 2012b; Manzano et al., 2012a; Manzano et al., 2012b; Monti et al., 2013; Monti et al., 2012; Nomura et al., 2012; Pitman and van Duin, 2012; Rahaman et al., 2010; Rahaman et al., 2011; Raju et al., 2013; Raymand et al., 2011; Raymand et al., 2010; Russo et al., 2011; van Duin et al., 2010; van Duin et al., 2012; Vedadi et al., 2010; Yusupov et al., 2012).

The combustion branch has been used in the following publications:(Castro-Marcano et al., 2012; Chenoweth et al., 2009a; Chenoweth et al., 2008a, 2009b; Chenoweth et al., 2008b; Ganesh et al., 2011; Goddard et al., 2010; Jiang et al., 2009; Khalilov et al.,

2013a, b; Khalilov et al., 2012a; Khalilov et al., 2012b; Neyts et al., 2011; Salmon et al., 2009a, b; Srinivasan and van Duin, 2011). As Figure 1.3 indicates, there are also several 'Independent' force fields, not belonging to the two major development branches – however, most of these force fields have later been merged with the combustion branch. Several elements – like carbon, hydrogen, oxygen and silicon – are present on both the combustion and the water branch – once a training set is available we have found that it is fairly straightforward to transfer a parameter set from either the combustion branch to the water-branch, or vice versa. As such, we hope that in the future all these force fields can be combined into a single, fully transferable, ReaxFF description covering most of the periodic table.

2. Input Files

2.1 General. The ReaxFF input files can be divided into mandatory files, which are necessary for the program to run, and optional files, which are only needed for specific applications. Table 2.1 gives an overview of the input files.

Table 2.1: Overview of ReaxFF input files.

File name	Section	Short description
geo	Mandatory	system geometry
ffield	Mandatory	force field parameters
control	Mandatory	run control switches
exe	Mandatory	executable script
models.in	Optional	geometry file location
tregime.in	Optional	temperature regime definition
vregime.in	Optional	volume regime definition
eregime.in	Optional	electric field definition
iopt	Optional	normal run/force field optimization toggle
addmol.bgf	Optional	molecular fragment definition. At specific intervals, this fragment will be added to the system
charges	Optional	fixed charges
vels	Optional	restart-file
trainset.in	Force field optimization	training set definition
params	Force field optimization	optimizable force field parameters
koppel2	Force field optimization	links values force field parameters

2.2 Mandatory files.

geo-file. This file describes the system geometry. Currently, ReaxFF supports the .geo, .bgf, .xyz and z-matrix input formats (see examples below). By default, ReaxFF assumes **geo** contains a .geo or a z-matrix format; to use other formats the igeofor-switch in the **control**-file should be given a value of 1 for .bgf format and a value of 2 for .xyz-format. Future developments will be primarily centered around the .bgf-format and will move away from the .geo-format. As such, only a brief description of the .geo-format will be given in this manual.

All **geo**-input formats are format sensitive. Below follows a discussion of several examples of **geo**-input files.

Example 2.1: Non-periodic .bgf-input file.

```
BIOGRF 200
DESCRP  Ethane_radical.
REMARK Example
RUTYPE NORMAL RUN
# THIS LINE IS IGNORED
FORMAT ATOM   (a6,1x,i5,1x,a5,1x,a3,1x,a1,1x,a5,3f10.5,1x,a5,i3,i2,1x,f8.5)
HETATM    1   C          39.53649  39.80304  39.57992   C   1 1  -0.2894
HETATM    2   H          39.96404  38.93497  39.07954   H   1 1   0.1031
HETATM    3   C          40.55862  40.34907  40.60075   C   1 1  -0.2330
HETATM    4   H          39.30695  40.55630  38.82721   H   1 1  0.1048
HETATM    5   H          38.62048  39.49467  40.08241   H   1 1  0.1048
HETATM    6   H          40.65314  39.88418  41.56157   H   1 1  0.1048
HETATM    7   H          41.36027  40.97776  40.26860   H   1 1  0.1048
FORMAT CONECT (a6,12i6)
CONECT    1    2    3    4    5
CONECT    2    1
CONECT    3    1    6    7
CONECT    4    1
CONECT    5    1
CONECT    6    3
CONECT    7    3
END
```

Example 2.1 shows a basic, non-periodic .bgf-input file. .bgf is a keyword-driven input format; each line starts with a keyword followed by information associated with that keyword. The .bgf-format is used by various molecular simulation software packages (i.e. Cerius2, Jaguar). As these programs should ignore lines starting with unrecognized keywords the .bgf-format should be easily portable between different applications. ReaxFF recognizes the following .bgf-keywords:

B IOGRF [VERSION NUMBER]: defines .bgf-version number. ReaxFF reads version number 200 (and will stop if other version numbers are provided).

DESCRP [NAME]: System description. This description can be used in **trainset.in** to define a force field training set.

REMARK: Remarks. Multiple REMARK lines are allowed.

RUTYPE [KEYWORD ;NUMBER]: Defines run parameters. If KEYWORD is NORMAL RUN (as in Example 2.1) ReaxFF uses the switches defined in the **control**-file. Alternatively, switches in the **control**-file can be overridden by the KEYWORD options listed in Table 2.2. This options is useful in force field optimizations as different MM-methods can be used for different training set geometries. If no RUTYPE-line is supplied, ReaxFF uses the **control**-file definitions

Table 2.2: RUTYPE keywords supported by ReaxFF.

KEYWORD	Description
NORMAL RUN	Use switches in control-file
MAXIT [NUMBER]	Stop MM-run after NUMBER iterations
ENDPO [NUMBER]	Stop MM-run when RMSG drops below NUMBER
MAXMOV [NUMBER]	Maximum atom movement (in 10^{-6} Å) during steepest descent MM minimization. With NUMBER=0 a conjugate gradient method is used.
SINGLE POINT	Stop MM-run after first point
DOUBLE PRECISION	Double MM maximum number of iterations and half RMSG end point criterion as defined in control-file
LOW PRECISION	Half MM maximum number of iterations as defined in control-file
CELL OPT [NUMBER]	Perform numerical cell optimization in MM. See control-file for options defined by NUMBER
NO CELL OPT	Do not perform numerical cell optimization

FORMAT [STRING]: Supplies format-information. ReaxFF ignores these FORMAT lines; they cannot be used to modify input formatting.

HETATM [ATOM INFO]: Defines atom type and atom position. In this order, the ATOM INFO consists of the atom number, the atom type, the x, y and z-coordinates of the atom in Å, the force field type (same as atom type for ReaxFF), two switches not used by ReaxFF and the atom partial charge. ReaxFF does not use these partial charges.

CONECT [ATOM NUMBER:CONNECTED ATOM NUMBERS]: Connection table. ReaxFF ignores these lines and calculates its own connections.

END: Last line of .bgf-input geometry.

Line starting with a # are ignored by ReaxFF and can be used to supply additional comment.

Example 2.2: Periodic .bgf-input geometry file.

```
XTLGRF 200
DESCRP fcc_1
REMARK Platinum fcc-structure
RUTYPE NORMAL RUN
CRYSTX    4.50640   4.50640   4.50640  90.00000  90.00000  90.00000
FORMAT ATOM   (a6,1x,i5,1x,a5,1x,a3,1x,a1,1x,a5,3f10.5,1x,a5,i3,i2,1x,f8.5)
HETATM    1   Pt          0.00000   0.00000   0.00000   Pt  1 1  0.00000
HETATM    2   Pt          2.25138   2.25138   0.00000   Pt  1 1  0.00000
HETATM    3   Pt          2.25138   0.00000   2.25138   Pt  1 1  0.00000
HETATM    4   Pt          0.00000   2.25138   2.25138   Pt  1 1  0.00000
FORMAT CONECT (a6,12i6)
END
```

Example 2.2 shows a periodic .bgf-input file. In principle, ReaxFF treats every system as periodic; in non-periodic systems it simply uses large values for the cell parameters (these values are defined by the axis1, axis2 and axis3-switches in the **control**-file). However, these **control**-file definitions are ignored if cell parameters are defined in the **geo**-file. Example 2.2 features the following keywords in addition to Example 2.1:

XTLGRF [VERSION NUMBER]: As BIOGRF-keyword, but tells ReaxFF that user-specified cell parameters are supplied with this geometry.

CRYSTX [A B C Alpha Beta Gamma]: Defines cell lengths (in Å) and cell angles (in degrees) for periodic system.

ReaxFF always uses Cartesian (not fractional) coordinates to define atom positions.

The .bgf-input format can also be used to define interatomic, valence angle, torsion angle and center-of-mass restraints used during MM or MD simulations. Such restraints can be used to drive reactions or to force conformational changes. Example 2.3 shows the

Example 2.3: .bgf-input file with restraints.

```
BIOGRF 200
DESCRP Hshift11
RUTYPE NORMAL RUN
FORMAT BOND RESTRAINT (15x,2i4,f8.4,f8.2,f8.5,f10.7)
#                          At1 At2  R12    Force1  Force2  dR12/dIteration(MD only)
BOND RESTRAINT    1   2  1.0900 7500.00 0.25000 0.0000000
FORMAT ANGLE RESTRAINT (16x,3i4,2f8.2,f8.4,f9.6)
#                             At1 At2 At3   Angle  Force1  Force2  dAngle/dIteration
(MD)
ANGLE RESTRAINT    1    2    3   120.00  250.00 1.00000  0.0000
FORMAT TORSION RESTRAINT (18x,4i4,2f8.2,f8.4,f9.6)
#                                At1 At2 At3 At4   Angle    Force1  Force2    dAngle/dIt
TORSION RESTRAINT    1    2    3    4     45.00     250.00  1.00000   0.0000
FORMAT MASCEN RESTRAINT FREE FORMAT
#                               x/y/z At1 At2   R     At3 At4  Force1  Force2
MASCEN RESTRAINT     x      1      3   1.50     4    7      50.00     0.25
FORMAT ATOM   (a6,1x,i5,1x,a5,1x,a3,1x,a1,1x,a5,3f10.5,1x,a5,i3,i2,1x,f8.5)
HETATM    1    C            39.53692  39.80281  39.57996    C   1 1  0.00000
HETATM    2    H            39.96200  38.93424  39.07781    H   1 1  0.00000
HETATM    3    C            40.55717  40.34771  40.59881    C   1 1  0.00000
HETATM    4    H            39.30845  40.55556  38.82947    H   1 1  0.00000
HETATM    5    H            38.62310  39.49566  40.08262    H   1 1  0.00000
HETATM    6    H            40.65332  39.88631  41.56086    H   1 1  0.00000
HETATM    7    H            41.35903  40.97771  40.27048    H   1 1  0.00000
FORMAT CONECT (a6,12i6)
END
```

available restraint input options. The FORMAT lines give the required input format for these restraints. Example 2.3 features the following keywords:

BOND RESTRAINT [At1 At2 R12 Force1 Force2 dR12/dIteration]: Defines bond (i.e. interatomic) restraint between atoms At1 and At2. An additional force is added to the ReaxFF potential, aiming to keep the distance between At1 and At2 restraint at value R12. Force1 and Force2 are the force constants used for this additional force, which is defined in Equation 2.1:

$$E_{restraint} = Force1*(1.0-\exp(Force2*(R_{ij}-R12)^2)) \qquad \textbf{Equation 2.1}$$

During MD-simulations the value of R12 will be modified by dR12/dIteration every iteration. By using this feature, ReaxFF can drive reactions during MD simulations. This option is not available during MM minimizations.

ANGLE RESTRAINT [At1 At2 At3 Angle Force1 Force2 dAngle/dIteration]: Defines angle restraint between atoms At1, At2 and At3. Any angle in the system can be restrained in this way, independent of connectivity. An additional force, similar to Equation 2.1, is added to the ReaxFF potential. As with the bond restraint, dAngle/dIteration can be used to drive angles during an MD simulation.

TORSION RESTRAINT [At1 At2 At3 At4 Angle Force1 Force2 dAngle/dIteration]: Defines torsion angle restraint between atoms At1, At2, At3 and At4. At this moment, this restraint can ONLY be used between connected atoms. In defining the torsion angle At2 should be smaller than At3. Force1 and Force2 define an additional force, similar to that described by Equation 1.2, that is added to the ReaxFF potential. dAngle/dIteration can be used to drive torsion angles during an MD simulation. Driving a torsion angle through 0º or through 180 º might cause problems.

MASCEN RESTRAINT [x/y/z At1 At2 R At3 At4 Force1 Force2]: Defines center-of-mass restraints between atoms At1 to At2 and atoms At3 to At4 in either the x, y or z-direction. An additional force, similar to that described in Equation 2.1, is added to the ReaxFF potential, restraining the center-of-mass of atoms At1 to At2 at distance R in the x/y/z directions from the center-of-mass of atoms At3 to At4.

To facilitate building training sets with multiple geometries, ReaxFF can run multiple simulations from one **geo**-file. This is done by putting these geometries in one **geo**-file, separated by one empty line after each END-keyword. Alternatively, the **models.in**-file can be used to define the paths to these multiple **geo**-files.

When combining multiple geometries in one **geo**-file, ReaxFF can repeat simulations on the previous input geometry with modified cell volume. This is useful for obtaining equation of state information for condensed phase systems. Example 2.4 shows an .bgf-input file from which ReaxFF first runs a single point simulation on a Pt-fcc unit cell, followed by a simulation on the same structure in which the cell volume is reduced by 80% (and the atomic positions are rescaled accordingly). This is done using the VCHANGE keyword:

VCHANGE [NUMBER]: repeat the previous simulation with rescaled cell volume and atomic coordinates. The rescaling factor is NUMBER*100%. ReaxFF will automatically use the RUTYPE NO CELL OPT (Table 2.2) option for structure specified with VCHANGE.

As mentioned, future ReaxFF input file developments will primarily focus around the .bgf-format. Anticipated developments include:

- Making the .bgf-input format free.
- Making more **control**-file options available through .bgf-file input, thus allowing all run-input information to be supplied in only one input file.
- Allowing addition of force field training set information (charges, energies) to the .bgf-file. This information should be automatically picked up by ReaxFF and compared to ReaxFF output.

Example 2.4: Periodic .bgf-input geometry file followed by VCHANGE option.

```
XTLGRF 200
DESCRP fcc_1
REMARK Platinum fcc-structure
RUTYPE SINGLE POINT
CRYSTX    4.50640   4.50640   4.50640   90.00000   90.00000   90.00000
FORMAT ATOM   (a6,1x,i5,1x,a5,1x,a3,1x,a1,1x,a5,3f10.5,1x,a5,i3,i2,1x,f8.5)
HETATM    1   Pt            0.00000   0.00000   0.00000    Pt  1 1  0.00000
HETATM    2   Pt            2.25138   2.25138   0.00000    Pt  1 1  0.00000
HETATM    3   Pt            2.25138   0.00000   2.25138    Pt  1 1  0.00000
HETATM    4   Pt            0.00000   2.25138   2.25138    Pt  1 1  0.00000
FORMAT CONECT (a6,12i6)
END

XTLGRF 200
DESCRP fcc_2
REMARK Rerun fcc_1 at 80% volume
VCHANGE 0.80
END
```

As an alternative to the .bgf-input format, ReaxFF can also read .geo-, .xyz- and z-matrix formats. Example 2.5 shows an ReaxFF .geo input file. The first line of the .geo-format contains a control-character (C in Example 2.5) and the name of the structure. The next lines contain the atom number, atom type and the x,y and z-coordinates. By means of the control-character, the .geo-format provides most of the options available with the .bgf-format. As we are moving away from supporting the .geo-format we only summarize these options in Table 2.3; more information can be obtained from the author.

Example 2.5: .geo-input file.

```
C   Ethyl_radical
 1 C    0.39536921883140E+02  0.39802812097390E+02  0.39579964797377E+02
 2 H    0.39962000508882E+02  0.38934237894502E+02  0.39077807973956E+02
 3 C    0.40557168455097E+02  0.40347711311539E+02  0.40598809008712E+02
 4 H    0.39308447421804E+02  0.40555557250666E+02  0.38829468433362E+02
 5 H    0.38623101361336E+02  0.39495660102816E+02  0.40082615760111E+02
 6 H    0.40653318128573E+02  0.39886313307789E+02  0.41560863579234E+02
 7 H    0.41359032241168E+02  0.40977708035298E+02  0.40270480449231E+02
```

Table 2.3 Control character options available with .geo-format.

Control character	Effect
C	Normal run as defined in control-file
F	Use cell parameters (defined on lines 2+3 of .geo-format)
1	Single point
D	Double precision run
H	Low precision run
5	Use cell parameters; single point

Example 2.6: z-matrix input file.

```
I  Ethyl_radical
         1 C
      1  2 C                          1.08962
   1  2  3 C              42.38253    2.15999
2  3  1  4 H -119.14400 110.52474    1.08723
2  3  1  5 H  119.14417 110.52466    1.08723
5  1  3  6 H   38.20847 120.04903    1.07130
5  1  3  7 H -159.92026 120.04917    1.07130
```

Example 2.6 demonstrates the ReaxFF z-matrix input format. The 'I' on position 3 on the first line is required as a format-type identifier and is followed by the structure name. The following lines have the 4i3,1x,a2,3f10.5 format and contain at_l, at_k, at_j, at_i, atype, $tors_{ijkl}$, $angle_{ijk}$ and R_{ij}, where $tors_{ijkl}$ is the at_i-at_j-at_k-at_l torsion angle, $angle_{ijk}$ the at_i-at_j-at_k angle, R_{ij} the at_i-at_j interatomic distance and atype the atom type of at_i. By using internal instead of Cartesian coordinates, the z-matrix-input format facilitates building molecules. The z-matrix does not provide room to define cell parameters and, as such, can only be used in concert with the default cell parameters axis1, axis2 and axis3 from the **control**-file.

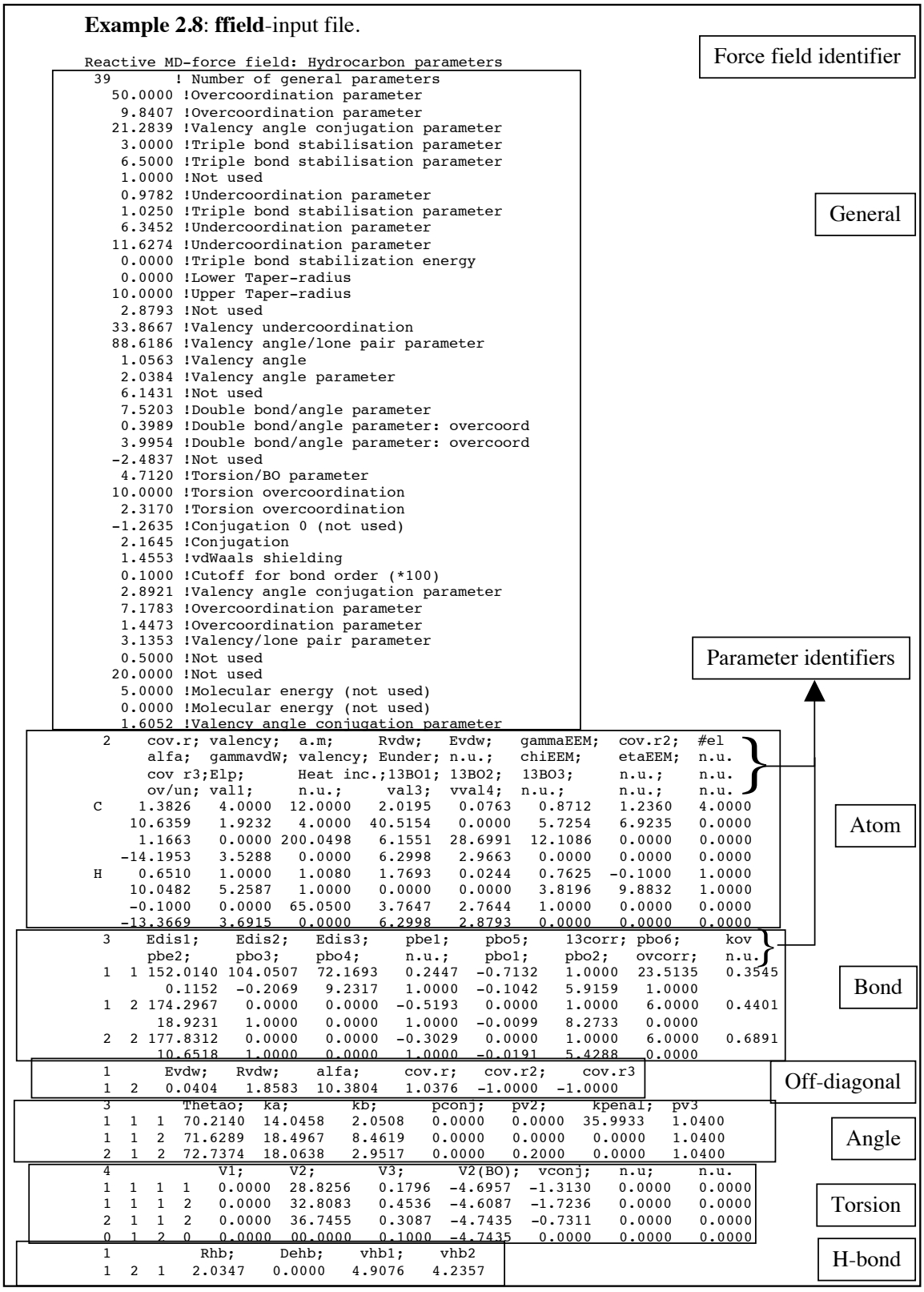
Example 2.7: .xyz-input file.

```
7
Ethyl_radical
C   39.53692  39.80281  39.57996
H   39.96200  38.93424  39.07781
C   40.55717  40.34771  40.59881
H   39.30845  40.55556  38.82947
H   38.62310  39.49566  40.08262
H   40.65332  39.88631  41.56086
H   41.35903  40.97771  40.27048
```

ReaxFF supports the .xyz-format in Example 2.7 because its simplicity facilitates communication with other simulation software and molecular viewers (e.g Icarus, Molden and Xmol). The first line of the .xyz-format contains the number of atoms, the second line the structure name. The other lines contain the type and x,y,z Cartesian coordinates for each of the atoms. The .xyz format cannot communicate cell parameters and, as such, can only be used with the default cell parameters axis1, axis2 and axis3 defined in the **control**-file.

ffield-file. The **ffield** input file contains the force field parameters. The first line of the **ffield**-input file contains a force field identifier (Example 2.8). Thereafter, the force field is divided into 7 sections, containing the general, atom, bond, off-diagonal, valence angle, torsion angle and hydrogen bond parameters. Below follows a description of the format and meaning of the force field parameters in each of these sections.

Example 2.8: **ffield**-input file.




```
Reactive MD-force field: Hydrocarbon parameters
 39       ! Number of general parameters
   50.0000 !Overcoordination parameter
    9.8407 !Overcoordination parameter
   21.2839 !Valency angle conjugation parameter
    3.0000 !Triple bond stabilisation parameter
    6.5000 !Triple bond stabilisation parameter
    1.0000 !Not used
    0.9782 !Undercoordination parameter
    1.0250 !Triple bond stabilisation parameter
    6.3452 !Undercoordination parameter
   11.6274 !Undercoordination parameter
    0.0000 !Triple bond stabilization energy
    0.0000 !Lower Taper-radius
   10.0000 !Upper Taper-radius
    2.8793 !Not used
   33.8667 !Valency undercoordination
   88.6186 !Valency angle/lone pair parameter
    1.0563 !Valency angle
    2.0384 !Valency angle parameter
    6.1431 !Not used
    7.5203 !Double bond/angle parameter
    0.3989 !Double bond/angle parameter: overcoord
    3.9954 !Double bond/angle parameter: overcoord
   -2.4837 !Not used
    4.7120 !Torsion/BO parameter
   10.0000 !Torsion overcoordination
    2.3170 !Torsion overcoordination
   -1.2635 !Conjugation 0 (not used)
    2.1645 !Conjugation
    1.4553 !vdWaals shielding
    0.1000 !Cutoff for bond order (*100)
    2.8921 !Valency angle conjugation parameter
    7.1783 !Overcoordination parameter
    1.4473 !Overcoordination parameter
    3.1353 !Valency/lone pair parameter
    0.5000 !Not used
   20.0000 !Not used
    5.0000 !Molecular energy (not used)
    0.0000 !Molecular energy (not used)
    1.6052 !Valency angle conjugation parameter
 2    cov.r; valency;  a.m;     Rvdw;   Evdw;   gammaEEM;  cov.r2;  #el
      alfa;  gammavdW; valency; Eunder; n.u.;   chiEEM;    etaEEM;  n.u.
      cov r3;Elp;      Heat inc.;13BO1; 13BO2;  13BO3;     n.u.;    n.u.
      ov/un; val1;     n.u.;     val3;  vval4;  n.u.;      n.u.;    n.u.
C    1.3826   4.0000  12.0000   2.0195   0.0763   0.8712   1.2360   4.0000
    10.6359   1.9232   4.0000  40.5154   0.0000   5.7254   6.9235   0.0000
     1.1663   0.0000 200.0498   6.1551  28.6991  12.1086   0.0000   0.0000
   -14.1953   3.5288   0.0000   6.2998   2.9663   0.0000   0.0000   0.0000
H    0.6510   1.0000   1.0080   1.7693   0.0244   0.7625  -0.1000   1.0000
    10.0482   5.2587   1.0000   0.0000   0.0000   3.8196   9.8832   1.0000
    -0.1000   0.0000  65.0500   3.7647   2.7644   1.0000   0.0000   0.0000
   -13.3669   3.6915   0.0000   6.2998   2.8793   0.0000   0.0000   0.0000
 3    Edis1;    Edis2;   Edis3;    pbe1;    pbo5;    13corr; pbo6;     kov
      pbe2;     pbo3;    pbo4;     n.u.;    pbo1;    pbo2;   ovcorr;   n.u.
 1  1 152.0140 104.0507  72.1693   0.2447  -0.7132   1.0000  23.5135   0.3545
       0.1152  -0.2069   9.2317   1.0000  -0.1042   5.9159   1.0000
 1  2 174.2967   0.0000   0.0000  -0.5193   0.0000   1.0000   6.0000   0.4401
      18.9231   1.0000   0.0000   1.0000  -0.0099   8.2733   0.0000
 2  2 177.8312   0.0000   0.0000  -0.3029   0.0000   1.0000   6.0000   0.6891
      10.6518   1.0000   0.0000   1.0000  -0.0191   5.4288   0.0000
 1       Evdw;   Rvdw;    alfa;     cov.r;   cov.r2;    cov.r3
 1  2    0.0404   1.8583  10.3804   1.0376  -1.0000  -1.0000
 3       Thetao;  ka;       kb;       pconj;   pv2;     kpenal;  pv3
 1  1  1  70.2140  14.0458   2.0508   0.0000   0.0000  35.9933   1.0400
 1  1  2  71.6289  18.4967   8.4619   0.0000   0.0000   0.0000   1.0400
 2  1  2  72.7374  18.0638   2.9517   0.0000   0.2000   0.0000   1.0400
 4          V1;      V2;      V3;      V2(BO);  vconj;   n.u;     n.u.
 1  1  1  1  0.0000  28.8256   0.1796  -4.6957  -1.3130   0.0000   0.0000
 1  1  1  2  0.0000  32.8083   0.4536  -4.6087  -1.7236   0.0000   0.0000
 2  1  1  2  0.0000  36.7455   0.3087  -4.7435  -0.7311   0.0000   0.0000
 0  1  2  0  0.0000  00.0000   0.1000  -4.7435   0.0000   0.0000   0.0000
 1          Rhb;     Dehb;    vhb1;    vhb2
 1  2  1   2.0347   0.0000   4.9076   4.2357
```

General parameters. The first section of the force field contains the general parameters, which affect all interactions regardless of atom type. The first line of this

section contain the npar, the number of general parameters present in the force field file (format i3), followed by npar lines each containing a parameter value followed by a parameter identified. Of particular interest are the Upper Taper radius parameters, which describes the non-bonded cutoff radius, and the Cutoff for bond orders, which describes the bond order threshold, above which atoms are considered connected. Both these parameters have major impact on the ReaxFF calculation speed; decreasing the Upper Taper radius parameter or increasing the bond order cutoff parameter will can make ReaxFF run considerably faster. These parameters, however, have a major impact on the force description and can, as such, not be changed without re-parameterization of other parts of the force field.

Table 2.4: Description of the Carbon ('C') parameters in the **ffield**-file in Example 2.8. n.u. identifiers in the **ffield**-file signify 'not used'; these parameters are left out of the description in this table.

Parameter value	Identifier	Description
1.3826	cov.r	sigma bond covalent radius
4.0000	valency	Valency
12.0000	a.m.	Atomic mass
2.0195	Rvdw	van der Waals radius
0.0763	Evdw	van der Waals dissociation energy
0.8712	gammaEEM	EEM shielding
1.2360	cov. r2	pi bond covalent radius
4.0000	#el.	Number of valence electrons
10.6359	alfa	van der Waals parameter
1.9232	gammavdW	van der Waals shielding
4.0000	valency	valency for 1,3-BO correction
40.5154	Eunder	Undercoordination energy
5.7524	chiEEm	EEM electronegativity
6.9235	etaEEM	EEM hardness
1.1663	cov. r3	double pi bond covalent radius
0.0000	Elp	Lone pair energy
200.049	Heat inc.	Heat of formation increment
6.1551	13BO1	Bond order correction
28.6991	13BO2	Bond order correction
12.1086	13BO3	Bond order correction
-14.1953	ov/un	Over/undercoordination
3.5288	vval1	Valence angle energy
6.2998	vval2	Valence angle energy
2.9663	vval3	Valence angle energy

Atom parameters. The second section of the force field contains the atom parameters. The section starts with the number of atom types present in the force field, followed by four lines of parameter identifiers. Thereafter follow four lines for each atom, starting with a line containing the atom name an 8 parameter values (format 1x,a2,8f9.4) followed by three lines with 8 parameter values (format 3x, 8f9.4). Table 2.4 gives a short

description of the meaning of each parameter for Carbon ('C' atom name) in the atom parameter section. If negative values are given to either of the three bond radii (sigma, pi and double pi) bond order contributions are ignored for that atom. In Example 2.8, only the sigma bond radius for H has a positive value (0.6510).

Table 2.5 Description of carbon-carbon bond parameters in the **ffield**-file in Example 2.8. n.u. identifiers in the **ffield**-file signify 'not used'; these parameters are left out of the description in this table.

Parameter value	Identifier	Description
152.0140	Edis1	Sigma-bond dissociation energy
104.0507	Edis2	Pi-bond dissociation energy
72.1693	Edis3	Double pi-bond dissociation energy
0.2447	pbe1	Bond energy
-0.7132	pbo5	Double pi bond order
1.000	13corr	1,3-Bond order correction
23.5135	pbo6	Double pi bond order
0.3545	kov	Overcoordination penalty
0.1152	pbe2	Bond energy
-0.2069	pbo3	Pi bond order
9.2317	pbo4	Pi bond order
-0.1042	pbo1	Sigma bond order
5.9159	pbo2	Sigma bond order
1.0000	ovcorr	Overcoordination BO correction

Bond parameters. The second section of the force field contains the bond parameters. This section starts with the number of bond types defined in the force field, followed by two lines of parameter identifiers. Then follow two lines for each bond type, the first of which (format 2i3,8f9.4) contains two atom type identifiers, followed by 8 parameter values. The atom type identifiers define the bond;'1 1', for example, describes the bond between atom #1 and atom #1 (the C-C bond in case of Example 2.8, as C is the first atom type defined in the Atom section). The second line (format 6x, 8f9.4) contains another 8 parameter values. ReaxFF will terminate if at *any* stage of the simulation a bond type is found that is not defined in the **ffield**-file. Table 2.5 gives a short description of the C-C bond parameters in the force field from Example 2.8.

Off-diagonal terms. 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.

The off-diagonal section starts with the number of off-diagonal types defined in the force field, followed on the same line by parameter identifiers. Then follow one line each for each off-diagonal type (format 2i3,6f9.4), beginning with the type identifier ('1 2' in Example 2.8 defines the C-H off diagonal parameters) followed by six parameters. Table 2.6 gives a short description of the C-H off-diagonal parameters from Example 2.8.

Table 2.6 Description of C-H off diagonal parameters in the **ffield**-input file in Example 2.8. The negative values for cov.r2 and cov. r3 signify that pi and double pi bond orders are not calculated for C-H pairs.

Parameter value	Identifier	Short description
0.0404	Ediss	vdWaals dissociation energy
1.8583	Rvdw	vdWaals radius
10.3804	alfa	vdWaals parameter
1.0376	cov. r	sigma bond covalent radius
-1.0	cov. r2	pi bond covalent radius
-1.0	cov. r3	double pi bond covalent radius

Valence angle parameters. The fifth section in the **ffield**-file contains the valence angle parameters. This section starts with a line containing the number of valence angles defined in the force field, followed with the parameter identifiers. Thereafter follow one line for each valence angle type. Each of these lines (format 3i3,7f9.4) starts with the valence angle identifier (e.g. '1 1 1' in Example 2.8, which defines the C-C-C valence angle) followed by seven parameters. ReaxFF will ignore valence angles in the simulation that are not defined in the **ffield**-file. Table 2.7 gives a short description of the C-C-C valence angle parameters from Example 2.8.

Torsion angle parameters. The sixth section in the ffield-file contains the torsion angle parameters. This sections starts with a line containing the number of torsion angles defined in the force field and the parameter identifiers. Thereafter follows one line for each torsion angle type. Each of these lines (format 4i3,7f9.4) starts with the torsion angle identifier, followed by seven parameters. Two types of torsion angle parameters identifiers are available: (1) identification by only the central bond or (2) identification by all four atoms in the torsion angle. For example, the '1 1 1 1' identifier on the second line of the torsion angle section in Example 2.8 defines the C-C-C-C torsion angle parameters, while the '0 1 2 0' identifier defined all X-C-H-X torsion angles. The four-atom identifier overrules the central bond identifier, thus, it is possible to define all carbon-carbon torsion angles with '0 1 1 0' and define a special case for C-C-C-H torsion angles with '1 1 1 2'.

Table 2.7 Description of C-C-C valence angle parameters in the **ffield**-input file in Example 2.8.

Parameter value	Identifier	Short description
70.2140	Thetao	180°-(equilibrium angle)[a]
14.0458	ka	1st force constant
2.0508	kb	2nd force constant
0.0000	pconj	Valence conjugation
0.0000	pv2	Undercoordination
35.9933	kpenal	Penalty energy
1.0400	pv3	Energy/bond order

[a]: This lead to an equilibrium angle of 109.7860° for the C-C-C sigma-bond angle.

ReaxFF will ignore torsion angles not defined in the **ffield**-file. Table 2.8 gives a short description of the C-C-C-C torsion angle parameters from Example 2.8.

Table 2.8 Description of the C-C-C-C torsion angle parameters in the **ffield**-input file in Example 2.8. n.u. identifiers mean that this parameter field is currently not used.

Parameter value	Identifier	Short description
0.0000	V1	V1-torsion barrier
28.8256	V2	V2-torsion barrier
0.1796	V3	V3-torsion barrier
-4.6957	V2(BO)	V2/bond order
-1.3130	vconj	Torsion angle conjugation

Hydrogen bond parameters. The seventh section of the force field contains the hydrogen bond parameters. This section starts with a line containing the number of hydrogen bond types described in the force field and the parameter identifiers. Thereafter follows one line for each hydrogen bond type. This line (format 3i3,4f9.4) first defines the hydrogen bond type, followed by four parameter values. The identifier '1 2 1' in line 2 of the hydrogen bond section in Example 2.8 refers to the C-H - - C hydrogen bond. ReaxFF will ignore hydrogen bonds not defined in the **ffield**-input file. Table 2.9 gives a short description of the C-H - - C hydrogen bond parameters from Example 2.8.

Table 2.9 Description of the C-H - - C hydrogen bond parameters in the **ffield**-input file in Example 2.8.

Parameter value	Identifier	Short description
2.0347	Rhb	Hydrogen bond equilibrium distance
0.0000	Dehb	Hydrogen bond dissociation energy
4.9076	vhb1	Hydrogen bond/bond order
4.2357	vhb2	Hydrogen bond parameter

control-file. This file contains the run-control parameters for ReaxFF. Example 2.9 shows a **control**-file used for an NVT MD-simulation. Apart from the comment-lines (lines starting with a #) each line contains a number, a keyword and an optional keyword description. The **control**-file is format free and keywords can be arranged in any order. ReaxFF will re-check **control**-file during an MD-simulation of a force field optimization run, giving the user the option to modify a ongoing simulation. If a keyword is left out of the **control**-file ReaxFF will use a default value. Tables 2.10-2.13 describe the name, default value and function of the general, MD, MM and force field optimization keywords currently recognized by ReaxFF. This division in general, MD, MM and force field optimization parameters is used here for convenience only and bears no further significance.

Example 2.9. control-file for an NVT MD-simulation.

```
# General parameters
      0 imetho     0: Normal MD-run 1: Energy minimisation 2:MD-energy minimisation
      1 igeofo     0:xyz-input 1: Biograf input 2: xmol-input geometry
 80.000 axis1      a (for non-periodical systems)
 80.000 axis2      b (for non-periodical systems)
 80.000 axis3      c (for non-periodical systems)
 0.0010 cutof2     BO-cutoff for valency angles and torsion angles
  0.300 cutof3     BO-cutoff for bond order for graphs
      3 icharg     Charges. 1:EEM 2:- 3: Shielded EEM  4: Full system EEM 5: Fixed
      1 ichaen     Charges. 1:include charge energy 0: Do not include charge energy
     25 irecon     Frequency of reading control-file
# MD-parameters
      1 imdmet     MD-method. 1:Velocity Verlet+Berendsen 2:-;3:NVE 4: NPT
  0.250 tstep      MD-time step (fs)
0050.00 mdtemp     1st MD-temperature
      2 itdmet     0: T-damp atoms 1: - 2:System 3: Mols 4: Anderson
   25.0 tdamp1     1st Berendsen/Anderson temperature damping constant (fs)
0001000 nmdit      Number of MD-iterations
    005 iout1      Output to unit 71 and unit 73
   0050 iout2      Save coordinates
 000025 irten      Frequency of removal of rotational and translational energy
  00025 itrafr     Frequency of trarot-calls
      1 iout3      0: create moldyn.xxxx-files 1: do not create moldyn.xxxx-files
      1 iravel     1: Random initial velocities
 000500 iout6      Save velocity file
 02.50 range       Range for back-translation of atoms
```

Table 2.10. Name, default value and function of the general keywords in the **control**-file.

Name	Default	Function
imetho	0	0: MD-run; 1: MM-run; 2: MD-energy minimization
igeofo	0	0: .geo- or z-matrix input; 1: .bgf-input; 2: .xyz-input
axis1	200.0	a-cell parameter for non-periodic systems
axis2	200.0	b-cell parameter for non-periodic systems
axis3	200.0	c-cell parameter for non-periodic systems
cutof2	0.001	Bond order cutoff for valence and torsion angles
cutof3	0.300	Bond order cutoff for graphs, molfra.out and **fort.7**
icharg	3	Charge calculation. 1: EEM; 2 -; 3: Shielded EEM; 4: Full-system EEM; 5: fixed charges from charges-file 7: ACKS2 charges
ichaen	1	0: Do not include charge energy; 1: include charge energy
iappen	0	1: Append **fort.7** and **fort.8**-connection tables 0: Overwrite
isurpr	0	0: Full output 1: Suppress some output; 2: Suppress most output and read in all geometries at once (for force field optimization)
icheck	0	0: Normal run; 1: Single point + 1st derivatives check;2: Single point
idebug	0	0: Normal run; 1: Output subroutine names in **fort.65**
ixmolo	0	0: Only x,y,z-coordinates in **xmolout**; 1: add velocities to **xmolout;** 2: x,y,z coordinates and molecule number; 3: x,y,z-coordinates, molecule number and strain energy; 4: x,y,z-coordinates, molecule number and total bond order
itrout	0	1: generate an diff_traj.xyz-file and a translate.out output file. The diff_traj.xyz contains the unfolded coordinates, allowing diffusion analysis. The translate.out-file keeps track of how often, and in which direction, the atoms migrate out of the central periodic cell.
iexx	1	Defines the number of additional periodic cells provided in the **fort.85** output file in the x-direction; can be used to generate supercells.
iexy	1	Same as iexx, only in the y-direction.
iexz	1	Same as iexz, only in the z-direction
cutmo1	0.0	If > 0.00, provide an additional output (molfra2.out) file like molfra.out, only with a different cutoff than cutof3
cutmo2	0.0	If > 0.00, provide an additional output (molfra3.out) file like molfra.out, only with a different cutoff than cutof3
ignore	0	Generate a new output file (molfra_ig.out), similar to molfra.out, in which all bond related to atom type [ignore] are ignored. The atom type number is defined in the ffield-file (for example, in **Example 2.8** carbon would be atom type 1, hydrogen atom type 2.

Table 2.10 (continued). Name, default value and function of the general keywords in the **control**-file.

Name	Default	Function
ireflx	0	1: reflective boundary in x-direction; only works for simple molecules
irefly	0	1: reflective boundary in y-direction; only works for simple molecules
ireflz	0	1: reflective boundary in z-direction; only works for simple molecules

Table 2.11. Name, default value and function of the MD-related keywords in the **control**-file. Frequency refers to how often (in MD-iterations) a certain function is performed. Refer to the Output-section for more detail on output files.

Name	Default	Function
imdmet	3	MD-method. 1: NVT; 2: -; 3: NVE; 4: NPT
tstep	0.25	MD time step (fs)
mdtemp	298.0	MD temperature
itdmet	2	Systems used in temperature control method. 0: atoms; 1: -; 2: whole system; 3: molecules
tdamp1	2.5	Berendsen temperature damping constant (fs)
mdpres	0.0	MD pressure (GPa)
pdamp1	500.0	Berendsen pressure damping constant (fs)
inpt	0	0: Change all cell parameters in NPT; 1/2/3: fixed a/b/c
nmdit	1000	Number of MD-iterations
ichupd	1	Charge update frequency
iout1	5	Output frequency to **fort.71** and **fort.73**
iout2	50	Output frequency to **xmolout** and **moldyn.vel**
ivels	0	0: Use velocities from **vels** restart-file; 1: Zero initial velocities
iout3	0	0: create **moldyn.xxxx** .xyz-trajectory files; 1: Not.
iravel	0	0: Zero initial velocities; 1: Random initial velocities. Ignored if **vels**-restart file is present.
endmd	1.0	RMSG endpoint criterium for MD energy minimization
iout6	1000	Frequency of **molsav.xxxx** restart file creation
irten	25	Rotational and translational energy removal frequency
npreit	0	Number of MD-iterations in previous runs (for restarts)
range	2.50	Range for back-translation of atoms outside periodic box (Å)
irecon	25	Frequency of re-checking **control**-file
iremov	0	Every iremov iterations, remove molecules with a mass between vrami2 and vramin from the simulation box; removed molecules coordinates are output in remove.### output file, where ### is the iteration number
vramin	0.1	Upper mass limit for molecule removal
vrami2	0.01	Lower mass limit for molecule removal

Table 2.12. Name, default value and function of the MM-related keywords in the **control**-file. Frequency refers to how often (in MM-iterations) a certain function is performed. Refer to the Output-section for more detail on output files.

Name	Default	Function
endmm	1.0	RMSG MM end point criterium
imaxmo	50	0: Conjugate gradient; >0: Maximum move ($1/10^6$ Å) in steepest descent
imaxit	50	Maximum number of MM iterations
iout4	50	Frequency of structure output in **xmolout**
iout5	0	1: remove **fort.57** and **fort.58**-files
icelop	0	0: No cell optimization 1: Numerical cell optimization
celopt	1.0005	Cell parameter optimization stepsize
icelo2	0	0: Cubic cell optimization; 1/2/3: only a/b/c; 4: c/a ratio

Table 2.13. Name, default value and function of the force field optimization related keywords in the control-file. For a more detailed discussion of these parameters see the Force field optimization input and output sections.

Name	Default	Function
parsca	1.0	Scale parameter step
parext	0.001	Parameter extrapolation
igeopt	1	0: Always use same start geometries; 1: Update start geometries
iincop	0	Heat increment optimization. 0: No; 1: Yes
accerr	2.50	Accepted increase error force field

exe-file. The **exe**-file contains the script that copies the input files to the right location, calls the ReaxFF-executable and, optionally, copies output files. The **exe**-script gets called from the UNIX command line. Example 2.10 shows the **exe**-script currently used by the author; users with a knowledge of UNIX scripting will probably want to personalize this file. Please note that the some of the input file names, like **geo** and **ffield**, used in this manual are defined in this script; the ReaxFF program only recognizes these input files as **fort.3** (**geo**) and **fort.4** (**ffield**). Other files (**control**, **models.in**, **trainset.in** and **tregime.in**) have their names hard coded in the ReaxFF source and need not to be moved by the **exe**-script to a fort.#-unit.

Example 2.10: **exe**-script used for running ReaxFF from the UNIX command line.

```
#
if (-f xmolout) rm xmolout
if (-f moldyn.vel) mv moldyn.vel vels23
if (-f fort.58) cp fort.58 58s
if (-f fort.57) cp fort.57 57s
if (-f fort.71) cp fort.71 71s
if (-f fort.73) cp fort.73 73s
if (-f fort.13) cp fort.13 13s
if (-f fort.99) cp fort.99 99s
if (-f fort.7) cp fort.7 7s
if (-f fort.8) cp fort.8 8s
touch fort.5a
touch moldyn.0a
touch molsav.0a
rm fort.*
rm moldyn.0*
rm molsav.0*
if  (-f control )  then
#
else
echo 'Missing control'
exit
endif
if  (-f geo )  then
cp geo fort.3
else
echo 'Missing geo'
exit
endif
if  (-f ffield )  then
cp ffield fort.4
else
echo 'Missing ffield'
exit
endif
if  (-f ranfile )  then
cp ranfile fort.35
else
echo 'Created ranfile in unit 35'
echo '234535.1' > fort.35
endif
if (-f iopt ) then
cp iopt fort.20
else
echo 'Created iopt in unit 20; assume normal run'
echo '  0   0: Normal run   1: Force field optimization' > fort.20
endif
touch fort.9
if (-f params) cp params fort.21
if (-f koppel2) cp koppel2 fort.23
if (-f charges) cp charges fort.26
if (-f vels) cp vels moldyn.vel
reac > run.log
exit
```

Save output files from previous run

Check presence mandatory files; stop if not present.
Copy **geo** to **fort.3** and **ffield** to **fort.4**

Create default **ranfile** and **iopt** if not provided.
If provided, copy these files to units 35 and 20.

Copy optional files to their units.

Call executable; redirect output to run.log

2.3 Optional files.

tregime.in-file. This file can be used to define temperature regimes during an NVT of NPT MD-simulation. Example 2.12 demonstrates the format used in this file.

Example 2.12. **tregime.in** input file .

```
#Start #Zones At1 At2 Tset1 Tdamp2 dT1/dIt At3 At4 Tset2 Tdamp2 dT2/dIt
0       2     1   20  200.0 50.0   0.05    21   40  10.0  1.0     0.00
10000   2     1   20  700.0 50.0   0.00    21   40  10.0  1.0     0.00
20000   2     1   20  700.0 50.0  -0.10    21   40  10.0  1.0     0.00
```

This example splits the system into 2 zones (#Zones), atoms 1-20 (At1-At2) and atoms 21-40 (At3-At4). Atoms 1-20 get heated up at a rate of 0.05K/iteration during the first 10,000 MD-iterations, taking their temperature from 200.0 to 700.0 K. During the second stage (iteration 10,000-20,000; line 3) atoms 1-20 remain constant at 700K after which they get cooled down at a rate of 0.10K/iteration during the last stage (iteration 20,000-end). Meanwhile, atoms 21-40 are kept at a temperature of 10.0K during the entire simulation. In principle, unlimited amounts of temperature zones and temperature stages can be defined in this way, thus allowing for intricate annealing simulations.

When a tregime.in-file is used the ReaxFF program generates a new output file (fort.75) providing temperature information for every temperature zone.

iopt-file. As the force field optimization routines are written as a shell around the rest of the ReaxFF program, the **control**-file cannot be used to switch between a normal and a force field optimization run. For this reason, the **iopt**-file, which gets copied by the **exe**-script to **fort.20**, gets read by this outside shell. The **iopt**-file only contains 1 integer, format i3. If a value of 0 is given a normal run is performed, while a 1 results in a force field optimization run. Officially, this file in mandatory, but the **exe**-script (Example 2.10) will create a default **fort.20** with a value of 0 if this file is not present in the directory from which it is invoked.

addmol.bgf-file. This optional input file allows the user to add a particular molecule with a particular temperature or velocity at various intervals during an MD-simulations, essentially creating a Grand-Canonical Molecular Dynamics ensemble.

Example 2.13: addmol.bgf file, used to add an O_2 molecule every 1000 steps at a temperature of 250K at a random location in the simulation box.

```
BIOGRF 200
DESCRP O2
FREQADD  1000
VELADD  1
STARTX -9000.0
STARTY -9000.0
STARTZ -9000.0
ADDIST 3.0
NATTEMPT 050
TADDMOL 250.0
FORMAT ATOM
(a6,1x,i5,1x,a5,1x,a3,1x,a1,1x,a5,3f10.5,1x,a5,i3,i2,1x,f8.5)
HETATM     1 O              0.00000   0.00000   0.00000 O     1 3  0.00000
HETATM     2 O              1.24500   0.00000   0.00000 O     1 3  0.00000
END
```

The keywords in the addmol.bgf file function as follows:

FREQADD: Indicates how often the molecule should be added to the simulation box. The first addition always happens at iteration 5, in example 2.13 the second addition would happen at iteration 1000, the third at iteration 2000 etc.

VELADD: 1: random initial velocities, according to TADDMOL; 2: read in velocities from addmol.vel (see **Example 2.14**)

STARTX: x-position of the centre-of-mass of the added molecule. If < -5000.0: random position.

STARTY: y-position of the centre-of-mass of the added molecule. If < -5000.0: random position.

STARTZ: z-position of the centre-of-mass of the added molecule. If < -5000.0: random position.

ADDIST: minimal distance between the added molecule and the other atoms in the simulation box.

NATTEMPT: number of attempts at placing the molecule; if after NATTEMPT attempts no position is found that is at least ADDIST Angstrom away from the other atoms the placement is skipped.

TADDMOL: Temperature of added molecule; only used with VELADD=1.

Example 2.14: Example of an addmol.vel-file associated with the addmol.bgf file from **Example 2.13**. This file has the same format as the velocity-fields in the vels-restart file (**Example 2.16**).

```
Atom velocities (Angstrom/s):
  0.676920600871422E+13   0.250389491659681E+13   0.385204179579294E+03
  0.638733784812228E+13   0.125025292253893E+13  -0.372288199177400E+03
```

The ReaxFF program generates output to the standard terminal regarding the placement of the molecules, as defined in the **addmol.bgf** optional input file.

charges-file. When a value of 5 is given for icharge keyword in the **control**-file (Table 2.10) ReaxFF will run with fixed charges, which it will read in from the **charges**-input file. **Example 2.15** demonstrates the format of this file.

Example 2.15. **charges**-file for a system containing 5 methane (molecule 1 to 5) and 5 water molecules (molecule 6 to 10).

```
   2              ! Number of molecule types (format i4)
   1   5   6  10  ! Molecule type definition (format 20i4)
Methane           ! Molecule 1 identifier
   5              ! Number of atoms in molecule 1
   1   -0.4800    ! Atom number; charge (format (i4,f10.6) )
   2    0.1200
   3    0.1200
   4    0.1200
   5    0.1200
Water             ! Molecule 2 identifier
   3              ! Number of atoms in molecule 2
   1   -0.8200    ! Atom number; charge
   2    0.4100
   3    0.4100
```

The **charges**-file input format facilitates defining fixed charges for systems containing multiple copies of the same molecule. With a bit of editing, this file can be straightforwardly generated from the ReaxFF partial charges output in **fort.56**. Running with fixed instead of geometry-dependent charges will make ReaxFF run faster and may help at the initial stages of equilibration, as the flexible charges may make a unequilibrated system unstable. However, in many cases partial charge distributions will be greatly modified by reactions. For that reason, running a reactive simulation with fixed charges might lead to unrealistic results.

The **charges**-file gets copied to **fort.26** by the **exe**-script, from which it is accessed by ReaxFF.

vels-file. The **vels**-file contains the atom positions, velocities and accelerations and can be used to restart an MD-simulation. During an MD-simulation ReaxFF will, at intervals defined by **control**-keywords iout2 and iout6 (Table 2.11), generate **moldyn.vel** and **molsav.xxxx** restart files. **moldyn.vel** contains the most recent system information. By copying one of these restart files to **vels** and re-running the **exe**-script ReaxFF will continue the MD-simulation. The geometry in **vels** will override any geometry given in the **geo**-file. The **geo**-file is still required, however, and ReaxFF will check whether the **geo** and the **vels**-file contain the same number of atoms. Example 2.16 demonstrates the format of the **vels**-file.

The **vels**-file gets (somewhat confusingly) copied to **moldyn.vel** by the **exe**-script. ReaxFF reads this **moldyn.vel**-file and subsequently overwrites it, at MD-iteration intervals defined by the iout2-keyword (Table 2.11) in the **control**-file, with the latest coordinates, velocities and accelerations.

Example 2.16: **vels**-restart file generated from an MD-simulation on an ethyl radical. The previous acceleration data is outdated and is of no influence to the ReaxFF-run.

```
 Lattice parameters:
    80.00000000    80.00000000    80.00000000
    90.00000000    90.00000000    90.00000000
   7 Atom coordinates (Angstrom):
   0.392027787892917E+02   0.400902383055691E+02   0.396569719222545E+02    C
   0.408076194536754E+02   0.407017914536647E+02   0.409641189620557E+02    H
   0.406156358473423E+02   0.399085739070730E+02   0.402446784332288E+02    C
   0.389320157142440E+02   0.409742263417328E+02   0.391261065022365E+02    H
   0.384743625215192E+02   0.392966529997087E+02   0.397356676283833E+02    H
   0.413546771765171E+02   0.399886339278518E+02   0.394389532581619E+02    H
   0.407016447482017E+02   0.389398236842467E+02   0.407560383741949E+02    H
 Atom velocities (Angstrom/s):
  -0.639976248783438E+12   0.451200007339684E+11  -0.216754312886992E+12
  -0.703141792567835E+11   0.421377644135005E+13   0.824472191800807E+13
   0.447898432906652E+12  -0.105077751499844E+13  -0.658290512224369E+11
   0.195067105934229E+13   0.289176812368303E+13  -0.140045800761676E+14
   0.476126532155786E+13  -0.152785617001799E+13   0.169491694508477E+14
   0.456467329116016E+13   0.118352257917078E+14  -0.396100285863348E+12
  -0.891965482760371E+13  -0.544080092166964E+13  -0.742912333885548E+13
 Atom accelerations (Angstrom/s**2):
  -0.810436027147472E+27  -0.981147717871880E+27  -0.313616213868668E+27
   0.116753745428499E+29   0.109570643908876E+29   0.135954922986376E+29
  -0.244853315374905E+27  -0.599286401168541E+27  -0.831858387309253E+27
   0.105361267406127E+28   0.134697085751095E+28  -0.115819691864393E+27
   0.208416152961380E+28   0.364332842399750E+28  -0.673228416751219E+27
  -0.149559883359616E+28  -0.125042567119739E+28   0.468080992010752E+28
  -0.754581549567212E+27   0.411775389213967E+28  -0.385065171515426E+28
 Previous atom accelerations:
   0.000000000000000E+00   0.000000000000000E+00   0.000000000000000E+00
   0.000000000000000E+00   0.000000000000000E+00   0.000000000000000E+00
   0.000000000000000E+00   0.000000000000000E+00   0.000000000000000E+00
   0.000000000000000E+00   0.000000000000000E+00   0.000000000000000E+00
   0.000000000000000E+00   0.000000000000000E+00   0.000000000000000E+00
   0.000000000000000E+00   0.000000000000000E+00   0.000000000000000E+00
   0.000000000000000E+00   0.000000000000000E+00   0.000000000000000E+00
 MD-temperature (K):
   0.550040311117348E+02
```

vregime.in-file. The vregime.in-optional input file allows the user to manipulate the cell parameters – very similar in format to the tregime.in-file (**Example 2.12**). **Example 2.17** shows an input format example – this file is format-free. When a vregime.in-file is provided, the ReaxFF program will generate a new output file (fort.77) that contains information regarding the cell parameters targeted in the vregime.in-file. One of the most common applications of this file is to strain condensed-phase systems for crack propagation simulations (e.g. (Buehler et al., 2007; Buehler et al., 2006) and to bring complex amorphous materials to their correct density (e.g. (Salmon et al., 2009b)).

Example 2.17. Example of a vregime.in file used to manipulate the cell parameters. If rescale=y then all atom coordinates get rescaled, if rescale=n only the cell parameters are changed, effectively putting all strain on the edge of the periodic cell.

```
#Volume regimes
#start #V type1 change/it rescale type 2  change/it  rescale
0000    2  alfa  0.050000   y     beta    -0.05        y
0100    2  beta  0.050000   y     alfa    -0.05        y
0200    2  a     0.010000   y     b       -0.010       y
0300    2  a    -0.010000   y     b        0.010       y
0400    4  a    -0.010000   y     alfa     0.050       y     b  0.01    y  beta  0.05    y
```

eregime.in-file. This optional input file allows the ReaxFF program to impose an electric field over the system (e.g. (Assowe et al., 2012; Neyts et al., 2012). This electric field is fully integrated with the EEM-charges – as such, the electric field will polarize the system. The eregime.in file is format free and allows the user to switch on/switch off the field, give it a direction (x/y/z) or even employ multiple fields (for example, in both the x- and y-direction. Beware: the electric field will not function correctly if a molecule moves across the periodic boundary in the field direction – this will lead to an energy discontinuity. As such, the electric field is best used in the presence of a large vacuum-layer that crosses the periodic boundary.

When an eregime.in file is provided the ReaxFF program will generate a new output file in fort.78, containing information regarding the electric field(s) strength in the x/y/z directions and its associated energy term.

Example 2.18. Example of a eregime.in file used to manipulate an electric field.

```
#Electric field regimes
#start #V direction Magnitude(V/Angstrom)
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
```

2.3 Force field optimization files.

trainset.in-file. This file allows the user to generate a training set or cost function, which can be used to optimize the force field parameters. **trainset.in** uses the identifiers defined in the DESCRP-field in the geo-file (.bgf-format) or in the models.in file to compare force field derived geometries and energy differences to literature or QC-values. **trainset.in** is format-free, although it does require that fields are space-separated. Also, the -, + and / symbols have special meaning in the **trainset.in**-file and should not be used in identifiers. Example 2.15 demonstrates how this file can be used to set up a cost function. The **trainset.in**-file is divided into 5 sections, each of which communicates a particular data type to ReaxFF. Each of these sections starts with a keyword (CHARGE, HEATFO, GEOMETRY, CELL PARAMETER and ENERGY) followed by data lines and ending by a END [keyword] line. In the CHARGE, HEATFO, GEOMETRY and CELL PARAMETER sections the data lines start with the structure identifier, followed by the weight of the data point. This is followed by a type identifier (the atom number for the CHARGE section, the bond/valence angle/torsion angle definition for the GEOMETRY section and a/b /c/alpha/beta/gamma for the CELL PARAMETER section) and, finally, the literature/QC value with which the ReaxFF-data is to be compared. The HEATFO section, which compares ReaxFF heat of formation data to literature data, does not require a type identifier. If an identifier is not provided in the GEOMETRY section ReaxFF is going to compare the ReaxFF RMSG of the forces.

In the ENERGY section **trainset.in** allows comparison of ReaxFF energy differences

Example 2.15: **trainset.in**-file with CHARGE, HEATFO, GEOMETRY, CELL PARAMETER and ENERGY sections. Lines commencing with a # are ignored by ReaxFF.

```
CHARGE
#Iden Weight Atom  Lit
chexane 0.1    1   -0.15
ENDCHARGE
HEATFO
#Iden   Weight Lit
methane  2.00 -17.80                    !Heat of formation
chexane  2.00
ENDHEATFO
GEOMETRY
#Iden   Weight At1 At2 At3 At4 Lit
chexane  0.01   1  2           1.54     !bond
chexane  1.00   1  2  3        111.0    !valence angle
chexane  1.00   1  2  3  4     56.0     !torsion angle
chexane  1.00                  0.01     !RMSG
ENDGEOMETRY
CELL PARAMETERS
#Iden       Weight Type Lit
chex_cryst  0.01    a  11.20
END CELL PARAMETERS
ENERGY
#Weigh op1 Ide1  n1 op2 Ide2    n2    Lit
#alfa vs. beta vs. gamma cleavage in butylbenzene
 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
#cyclohexane heat of vaporization
 1.0   + chex_cryst/16 — chexane/1  -11.83
ENDENERGY
```

between structures to literature/QC data. In this case each data line starts with the weight of the data point, followed by up to five operator/identifier/divider parts and finishes with the literature/QC value. The operator is either '+' or '-' ('+' is the default). The energy associated with the identifier is divided by the divider, allowing comparison of condensed phase structures to monomers, as exemplified in the last data line of the ENERGY section in which the ReaxFF heat of vaporization for cyclohexane is compared to a value of –11.83 kcal/mol by dividing the energy of a crystal by its number of monomers (16) and subtracting the energy of the gas molecule. The '/' character in the ENERGY section data lines is optional.

After running ReaxFF on the structures associated with the keywords in **trainset.in** the program will automatically compare the ReaxFF-data with the literature/QC data provided in the training set and produce **fort.13** and **fort.99** output file (see Output section).

BUGS:

- Literature/QC values of zero (0.00) cannot be used and will cause unpredictable results. This can be avoided by comparing with a very small value (i.e. 0.0001) instead.

- Structure identifiers should always refer to only one structure; if the same structure identifier is used twice ReaxFF will get confused when that identifier is mentioned in the ENERGY section. As of yet, ReaxFF does not stop or give a warning of this, so users should make certain that they use unique identifiers.

params-file. The **params**-file tells ReaxFF which force field parameters are to be incorporated in the force field optimization scheme. Example 2.16 demonstrates the format of the **params**-file.

Example 2.16: **params**-file, describing the parameters used in the force field optimization procedure.

```
  1   1   1    0.100   75.000   25.000   !1st general parameter
  1   2   1    0.010   10.000    9.00    !2nd general parameter
  2   1   1    0.001    1.600    1.350   !2nd section,1st type,1st parameter
  2   2   1    0.001    0.700    0.600   !2nd section,2nd type,1st parameter
  2   2   5    0.010    0.020    0.030   !2nd section,2nd type,5th parameter
  6   3   4    0.050   -3.000   -5.000   !6th section,3rd type,4th parameter
```

Each line (format 3i3,3f8.4) starts with three integers defining the force field parameter. The first of these integer defines the force field section (general, atom, bond, off-diagonal, valence angle, torsion angle, hydrogen bond; see Example 2.8), the second the type and the third the parameter. For example, the last line of **params** in Example 2.16 identifies parameter '6 3 4', which is the 4th parameter of the H-C-C-H (2-1-1-2) torsion angle for the **ffield**-file of Example 2.8. The value following these three integers defines the interval within which the parameter value is to be searched during the force field optimization procedure. This value is followed by the maximum and minimum allowed for that particular force field parameter.

As such, the last line of Example 2.16 instructs ReaxFF to perform a parameter search for the 4th parameter of the H-C-C-H torsion angle. This torsion angle parameter starts at a value of –4.7435 (Example 2.8); ReaxFF is going to increase and decrease its value by a factor 0.050, recalculate the cost function, and determine an optimal value based on a fit to the training set, using a parabolic extrapolation constrained by a maximum value of –3.000 and a minimum value of –5.000.

ReaxFF simply starts at the first line of **params** and works its way down, until it reaches the end of the file after which it terminates. For each parameter the constrained parabolic extrapolation procedure described previously is followed. For each parameter thus optimized information is written to **fort.79** regarding the results from the parabolic extrapolation. The parext-keyword in the **control**-file regulates how far outside the search interval ReaxFF is allowed to extrapolate, while the parsca-keyword can be used to decrease and increase the search interval (as the **control**-file is read continuously this can be useful to modify long-running force field optimizations).

The **params**-file gets copied to **fort.21** by the **exe**-script, from which it is accessed by ReaxFF.

koppel2-file. The **koppel2**-file allows the user to establish links between different force field parameters during the force field optimization procedure. Example 2.17 demonstrates the format of the **koppel2**-file.

Example 2.17: **koppel2**-file, used for linking force field parameter values.

```
5   1   7   5       !Parameter identifier; Nr. of links
5   2   7
5   3   7
5   4   7           !Parameters linked to parameter 5   1   7
5   5   7
5   6   7
```

The **koppel2**-file starts with a parameter identifier, which is formulated similar as in the **params**-file (‘5 1 7’ in Example 2.17 indicating section 5 (valence angles), type 1, parameter 7 {1.0400 for the **ffield**-file in Example 2.8}). Thereafter follows how many other force field parameters are to retain the same value as parameter ‘5 1 7’. The subsequent lines in **koppel2** define these other parameters, after which another link can be defined. The **koppel2**-file requires a 4i3-format.

The **koppel2**-file gets copied to **fort.23** by the **exe**-script, from which it is accessed by ReaxFF.

3. Output Files

3.1 General. Table 3.1 categorizes and gives a short description of the ReaxFF output files.

Table 3.1: ReaxFF output files.

Name	Related to	Description
NAME.bgf[a]	all runs	Output geometry in .bgf-format
NAME.geo	all runs	Output geometry in .geo-format
output.pdf	MM/MD runs	Output geometry in .pdf-format
output.MOP	MM/MD runs	Output z-matrix in MOPAC format
xmolout	all runs	Trajectory in .xyz-format
moldyn.vel	MD runs	restart file (see vels-file)
molsav.####[b]	MD runs	restart file, controlled by iout6[c]
moldyn.####	MD runs	trajectories, controlled by iout2[c]
fort.7	all runs	Connection table using cutof3 bond order cutoff[c]
fort.8	all runs	Connection table+charges
fort.13	FF optimization	Total error force field
fort.24	FF optimization	Values for heat increments (iincop=1)
fort.40	all runs	Forces (icheck>0[c])
fort.41-fort.50	all runs	Partial forces (icheck>0[c])
fort.56	MM/MD runs	ReaxFF charge distribution
fort.57	MM-runs	Energy minimization report
fort.58	MM-runs	Partial energy contribution report
fort.59	MD-runs	NPT: pressure report
fort.65	all runs	idebug=1[c]: subroutine names
fort.71	MD-runs	Energy/temperature/pressure report
fort.73	MD-runs	Partial energy contribution report
fort.74	all runs	Heat of formation+volume
fort.76	all restraint runs	Restraint report
fort.79	FF-optimization	Parabolic extrapolation report (see params-file)
fort.83	FF-optimization	Force field files from FF-optimization
fort.90	MM/MD runs	.bgf-output
fort.91	MM/MD runs	z-matrix output
fort.98	MM/MD runs	.geo-output
fort.99	FF-optimization	Detailed cost-function report

[a]: NAME=identifier in DESCRP field. [b]: #### is number of iterations/iout6. [c]: see **control**-file section.

Some of these output files are fairly straightforward or are, format-wise, the same as files discussed in the Input-files section. The sections below give a more elaborate description for the output-files that do not fall into either of these two categories.

3.2 MM/MD output files.

xmolout-file. The **xmolout**-file contains the trajectory from the MD-run or the MM-minimization. It is the generally the most useful file for interaction with molecular viewing programs. Programs like Icarus, Xmol, Molden and Jmol can directly read in **xmolout** and can provide graphical animations. In MD-runs, the frequency of structure output to **xmolout** is controlled by the **control**-file keyword iout2; a frame is saved every iout2 MD-iterations. In a similar way, iout4 controls the frame output frequency during MM simulations. **xmolout** always gets appended (i.e. never overwritten) during ReaxFF-simulations. The **exe**-script (see Example 2.10) can remove **xmolout**-files from the run directory; if this is not done the **xmolout**-output is appended. When ReaxFF is run on multiple geometries (either by putting them in one **geo**-input file or using the **models.in**-file **xmolout** will contain output for all these geometries. Example 3.1 demonstrates the format of the **xmolout**-file.

Example 3.1: **xmolout**-file, containing 3 frames from an MD-simulation on an ethyl radical.

```
7                                    !Number of atoms
Ethyl_radical                        !Structure identifier
C    39.19924  40.08333  39.66585 !Atom type;x/y/z
H    40.88239  40.79608  40.81249 !(format a2,3f10.5)
C    40.61413  39.91227  40.24250
H    38.86032  41.06389  39.38266
H    38.59716  39.25081  39.41763
H    41.33155  39.78117  39.45850
H    40.65892  39.04746  40.86980
   7
Ethyl_radical
C    39.20278  40.09024  39.65697
H    40.80762  40.70179  40.96412
C    40.61564  39.90857  40.24468
H    38.93202  40.97423  39.12611
H    38.47436  39.29665  39.73567
H    41.35468  39.98863  39.43895
H    40.70164  38.93982  40.75604
   7
Ethyl_radical
C    39.20278  40.09024  39.65697
H    40.80762  40.70179  40.96412
C    40.61564  39.90857  40.24468
H    38.93202  40.97423  39.12611
H    38.47436  39.29665  39.73567
H    41.35468  39.98863  39.43895
H    40.70164  38.93982  40.75604
```

If the ixmolo control-switch has a value of 1 additional velocity and molecule number output is generated in the **xmolout**-file.

fort.7/fort.8-files. These files contain the ReaxFF-generated connection table and charge distribution. **fort.8** contains information on all bonds, **fort.7** contains only the

bonds with a bond order greater then **control**-file keyword cutof2. These files get updated with the same frequency as the **xmolout**-file; every iout2 iterations during MD-simulations or every iout4 iterations during an MM-run. Example 3.2 demonstrates the format of these files.

Example 3.2: **fort.7**-connection table for an ethyl radical.

```
7 Ethyl_radical
  1  1  2  3  4  5  0  1 0.985  1.024  0.986  0.986  0.000  3.981  0.000 -0.289
  2  2  1  0  0  0  0  1 0.985  0.000  0.000  0.000  0.000  0.985  0.000  0.103
  3  1  1  6  7  0  0  1 1.024  0.987  0.987  0.000  0.000  2.999  0.000 -0.233
  4  2  1  0  0  0  0  1 0.986  0.000  0.000  0.000  0.000  0.986  0.000  0.105
  5  2  1  0  0  0  0  1 0.986  0.000  0.000  0.000  0.000  0.986  0.000  0.105
  6  2  3  0  0  0  0  1 0.987  0.000  0.000  0.000  0.000  0.987  0.000  0.105
  7  2  3  0  0  0  0  1 0.987  0.000  0.000  0.000  0.000  0.987  0.000  0.105
```

The first line of the **fort.7** and **fort.8**-files contain the number of atoms and the structure identifier. After that follows a line for each atom, starting with the atom number and the atom type number (as defined in the **ffield**-file). Then follow a number of integers, indicating the connectivity of this atom. The last integer gives the molecule number. Then follow the bond orders for each bond defined by the connectivity, followed by the sum of the bond orders, the number of lone pairs and finally the partial charge.

Normally, **fort.7** and **fort.8** get overwritten by the most recent information. Setting control-file keyword iappen to 1, though, causes output to these files to get appended.

fort.57-file. This file contains a report from the MM-energy minimization. - Example 3.3 demonstrates the format of this file.

Example 3.3: **fort.57**-output file generated from an MM conjugate gradient minimization (RMSG endpoint criterion 0.500) on an ethyl radical.

```
Ethyl_radical
 Iter.        Epot              Max.move     Factor       RMSG       nfc
    0    -664.0048203101      0.000000     0.500000     4.969706     0
    1    -664.0170216413     15.795587     0.089117     3.360058     0
    2    -664.0394728751      9.339616     0.135649     6.279516     0
    3    -664.0765683298      0.001387     1.957694     3.047916     0
    4    -664.0956555161      0.000342     3.000000     2.270724     0
    5    -664.1009627083      0.001933     0.585262     1.026026     0
    6    -664.1025221864      0.000851     1.000000     0.465193     0
```

Of main interest in the **fort.57** file are the columns related to the potential energy (Epot, in kcal/mol) and the RMSG of the forces on the structure (RMSG). The other columns show data related to the conjugate gradient minimizer performance. When RMSG drops below the value of **control**-file keyword endmm the MM run terminates.

Alternatively, the run terminates when the maximum number of iterations (**control**-file keyword maxit) has been reached.

fort.58-file. This file contains a partial energy report from the MM-energy minimization. -Example 3.4 demonstrates the format of this file.

Example 3.4: **fort.58**-output file generated from the same MM-run as used for Example 3.3.

Ethyl_radical

Iter.	Eatom	Elopa	Ebond	Emol	Eval	Ecoa	Ehb	Etor	Econj	Evdw	Ecoul
0	-6.922	0.000	-907.050	0.000	0.237	0.000	0.000	4.715	-0.508	257.018	-22.970
1	-6.741	0.000	-907.442	0.000	0.244	0.000	0.000	4.801	-0.512	257.123	-22.966
2	-6.559	0.000	-908.118	0.000	0.242	0.000	0.000	4.876	-0.516	257.524	-22.965
3	-6.196	0.000	-909.827	0.000	0.233	0.000	0.000	5.013	-0.523	258.720	-22.972
4	-6.283	0.000	-910.220	0.000	0.230	0.000	0.000	4.954	-0.521	259.253	-22.985
5	-6.305	0.000	-910.733	0.000	0.234	0.000	0.000	4.924	-0.520	259.819	-22.996
6	-6.340	0.000	-911.121	0.000	0.234	0.000	0.000	4.896	-0.519	260.277	-23.006

Energy contributions reported in **fort.58** include over+undercoordination energy (Eatom), lone-pair energy (Elopa), bond energy (Ebond), molecular energy (Emol; not used in current force fields), valence angle energy (Eval), valence angle conjugation energy (Ecoa), hydrogen bond energy (Ehb), torsion angle energy (Etors), van der Waals energy (Evdw) and Coulomb energy (Ecoul). Currently, it does not contain the energy required for generating the charges (Echarge, see **fort.73**-output file) which means that the sum of the partial energies in **fort.58** does not add up to the potential energy in **fort.57**.

fort.71-file. This file contains the energy, temperature and pressure information from an MD-simulation. Example 3.5 demonstrates the format of this file.

Example 3.5: **fort.71**-output file generated from a 45-iteration MD simulation.

Iter.	Nmol		Epot	Ekin	Etot	T(K)	Eaver(b)	Eaver(tot)	Taver	Tmax	Pres	sdev1	sdev2	Tset	Tstep	RMSG
5	1	1	-663.7	0.8	-662.9	38.3	-663.8	-663.8	41.3	46.0	0.0	0.1	0.1	50.0	0.5	22.2
10	1	1	-663.9	0.9	-663.0	42.4	-663.8	-663.8	42.2	44.5	0.0	0.0	0.0	50.0	0.5	28.7
15	1	1	-663.8	0.9	-662.9	43.9	-663.8	-663.8	42.5	43.9	0.0	0.0	0.0	50.0	0.5	18.9
20	1	1	-663.9	1.0	-662.9	49.9	-663.9	-663.8	48.6	49.9	0.0	0.0	0.0	50.0	0.5	7.4
25	1	1	-663.7	0.8	-662.9	37.1	-663.7	-663.8	40.5	45.4	0.0	0.1	0.0	50.0	0.5	25.0
30	1	1	-663.8	0.5	-663.3	24.6	-663.8	-663.8	21.5	25.6	0.0	0.1	0.0	50.0	0.5	31.1
35	1	1	-663.8	0.5	-663.3	25.0	-663.8	-663.8	25.4	27.0	0.0	0.0	0.0	50.0	0.5	15.0
40	1	1	-663.7	0.5	-663.2	25.3	-663.7	-663.8	24.3	25.4	0.0	0.0	0.0	50.0	0.5	13.1
45	1	1	-663.7	0.6	-663.2	26.9	-663.7	-663.8	24.9	26.9	0.0	0.0	0.0	50.0	0.5	30.0

control-file keyword iout1 regulates the output-frequency to **fort.71** (5 in case of Example 3.5). The **fort.71**-file has 16 columns, which contain the following information:

Column header Information

- Iter: Number of MD-iterations.
- Nmol: Number of molecules; number of molecules using **control**-file keyword cutof2 as bond order criterion (comparable with information in **fort.8** and **fort.7**).
- Epot: Total potential energy.
- Ekin: Total kinetic energy.

- Etot: Epot+Ekin.
- T(K): MD-temperature.
- Eaver(b): Block average potential energy over the last iout1-iterations.
- Eaver(tot): Average potential energy over entire run.
- Taver: Average temperature over the last iout1 iterations.
- Tmax: Maximum temperature in the last iout1 iterations.
- Pres: MD-pressure (in MPa) based on intermolecular interactions (cannot yet be used to evaluate pressures in condensed phase materials).
- sdev1: Standard deviation in potential energy over last iout1 iterations.
- sdev2: Standard deviation in average potential energy over entire run.
- Tset: Set temperature.
- Tstep: MD time step (on fs).
- RMSG: Root mean square of forces. In an MD-energy minimization run (**control**-file keyword imetho=2) ReaxFF will terminate if the RMSG drops below the value of **control**-file keyword endmd.

fort.73-file. This file contains the partial energy contribution information from an MD-run. Example 3.6 demonstrates the format of this file.

Example 3.6: **fort.73-**output file generated from the same MD-simulation as that used for Example 3.5.

Iter.	Ebond	Eatom	Elp	Emol	Eval	Ecoa	Ehbo	Etors	Econj	Evdw	Ecoul	Echarge
5	-905.00	-7.39	0.00	0.00	0.30	0.00	0.00	4.57	-0.49	255.78	-22.93	11.45
10	-913.27	-6.45	0.00	0.00	0.34	0.00	0.00	4.87	-0.51	262.72	-23.33	11.76
15	-915.74	-5.42	0.00	0.00	0.20	0.00	0.00	5.38	-0.53	263.86	-23.29	11.74
20	-910.80	-5.68	0.00	0.00	0.23	0.00	0.00	5.42	-0.53	258.94	-23.05	11.54
25	-907.76	-6.64	0.00	0.00	0.44	0.00	0.00	4.97	-0.51	257.35	-23.13	11.60
30	-910.68	-6.80	0.00	0.00	0.37	0.00	0.00	4.74	-0.51	260.62	-23.32	11.76
35	-914.36	-6.12	0.00	0.00	0.37	0.00	0.00	4.95	-0.53	263.46	-23.23	11.68
40	-910.62	-6.00	0.00	0.00	0.47	0.00	0.00	5.17	-0.53	259.26	-22.80	11.34
45	-906.77	-6.74	0.00	0.00	0.31	0.00	0.00	4.98	-0.51	256.49	-22.88	11.40

The output-frequency to **fort.73** is regulated by **control**-file keyword iout1 (5 in case of Example 3.6). **fort.73** contains all energy contributions, including bond energy (Ebond), over+undercoordination energy (Eatom), lone-pair energy (Elp), molecular energy (Emol; not use in recent force fields), valence angle energy (Eval), angle conjugation energy (Ecoa), hydrogen bond energy (Ehbo), torsion angle energy (Etors), van der Waals energy (Evdw), Coulomb energy (Ecoul) and charge polarization energy (Echarge). The sum of these terms should be the same as the number in the Epot-column in **fort.71**.

3.3 Force field optimization output files.

fort.13-file. This file contains the total error of the force field, calculated from the cost function defined in **trainset.in**. Example 3.7 shows the output to **fort.13** that could get generated in conjunction with the **params**-input file from Example 2.16 and the **ffield**-input file from Example 2.8.

Example 3.7: **fort.13**-output file generated using the **ffield**-, **params-** and **trainset.in**-input files from Examples 2.15, 2.16 and 2.18. See also the **fort.79**-example in the next section.

```
 91.5209      !Total FF error using 45.00 for parameter 1  1  1
 91.5893      !Total FF error using 55.00 for parameter 1  1  1
 91.6449      !Total FF error using 50.00 for parameter 1  1  1
 91.5209      !Total FF error using optimized value for parameter 1  1  1 (see fort.79)
 91.6038      !Total FF error using 9.7423 for parameter 1  2  1
 91.4330      !Total FF error using 9.9391 for parameter 1  2  1
 91.5175      !Total FF error using 9.8407 for parameter 1  2  1
 91.3896      !Total FF error using optimized value for parameter 1  2  1 (see fort.79)
```

As Example 2.8 demonstrates, for each parameter named in **params** four separate ReaxFF runs are performed on the input geometries associated with the training set, creating four entries in **fort.13**. In the first two of these runs the parameter value is decreased and increased according to the search interval defined in the **params**-file. The third run is performed with the unmodified force field. The total force field errors (see **fort.99** for a discussion on how this total error is determined) from these 3 runs are subsequently entered in a parabolic extrapolation procedure (see **fort.79** for a detailed discussion of this procedure) from which an optimal value for that parameter is calculated. The fourth run is performed with this optimal value, after which the next parameter in the **params**-file is tackled.

fort.79-file. This file contains the report from the parabolic extrapolation procedure used by ReaxFF to optimize the force field parameters. Example 2.9 demonstrates the format of this file. This example is linked to the discussion of the **fort.13**-file in the previous section. In the first section of the **fort.79**-example, concerning the 1 1 1-parameter, the parabolic search identifies a hill parabol (a<0). This means that extrapolation is not possible, so ReaxFF simply sticks with the best of the three parameter values. In the second section, concerning the 1 2 1-parameter, it identifies 7.4 as the optimal parameter value, but as this lies outside the parext extrapolation limit it chooses 8.6 instead as the new parameter value.

After calculating the total force field error for three parameter values ReaxFF tries to draw a parabol (ax^2+bx+c) through these three points. If a>0 than ReaxFF will use extrapolation (within the limits set by the parext **control**-file parameter) or interpolation to find the optimum parameter value.

Example 2.9: **fort.79**-output generated using the **ffield**-, **params**-and **trainset.in**-input files from Examples 2,15, 2.16 and 2.18. See also the **fort.13**-example in the previous section.

```
Values used for parameter  1  1  1
   0.4500000000E+02    0.5500000000E+02    0.5000000000E+02
Differences found
   0.9152086304E+02    0.9158925608E+02    0.9164489675E+02
Parabol: a=   -0.3593487269E-02  b=    0.3661880309E+00  c=    0.8231921337E+02
Minimum of the parabol    0.5095162492E+02
Difference belonging to minimum of parabol    0.9164814892E+02
New parameter value    0.4500000000E+02
Difference belonging to new parameter value    0.9152086100E+02

Values used for parameter  1  2  1
   0.9742293000E+01    0.9939107000E+01    0.9840700000E+01
Differences found
   0.9160380360E+02    0.9143296226E+02    0.9151752206E+02
Parabol: a=    0.8889634892E-01  b=   -0.2617639062E+01  c=    0.1086682558E+03
Minimum of the parabol    0.1472298410E+02
Difference belonging to minimum of parabol    0.8939852465E+02
New parameter value    0.9988802535E+01
Difference belonging to new parameter value    0.9139091180E+02
```

This one-parameter search procedure has the advantage of being robust. Furthermore, it is relatively straightforward to retrace one or more steps if the force field optimization has gone astray (for example, if a parameter reaches a physically unrealistic value). More sophisticated multiparameter search methods may be more economical, but do run the risk of finding themselves trapped in the strong ReaxFF inter-parameter correlations.

fort.99-file. This file contains a detailed report of the ReaxFF reproduction of the data in the training set (as defined in the **trainset.in**-file). Example 3.10 demonstrates the format of the **fort.99**-file. This example was generated from the trainset.in-file from Example 2.15.

Example 3.10: **fort.99**-output file as generated using the **trainset.in-file** from Example 2.15. For cross-reference see the **fort.13** and **fort.79**-examples (Example 3.8 and 3.9).

		FField value	QM/Lit value	Weight	Error	Total error
methane	Heat of formation:	-17.8000	-17.8000	2.0000	0.0000	0.0000
chexane	Charge atom: 1	-0.1604	-0.1500	0.1000	0.0109	0.0109
Heat of formation:		-29.4900	-29.4900	2.0000	0.0000	0.0109
Bond distance:	1 2	1.5586	1.5400	0.0100	3.4571	3.4679
Bond distance:	1 7	1.1696	1.1000	0.0200	12.1227	15.5906
Bond distance:	1 8	1.1713	1.1000	0.0200	12.7203	28.3109
Valence angle:	1 2 3	110.8117	111.0000	1.0000	0.0354	28.3463
Valence angle:	7 1 8	104.3207	107.0000	1.0000	7.1788	35.5251
chex_cryst	a:	11.8448	11.2000	0.4000	2.5987	38.1238
Energy +butbenz/ 1 -butbenz_a/ 1		-96.6941	-90.0000	1.5000	19.9158	58.0396
Energy +butbenz/ 1 -butbenz_b/ 1		-63.4751	-71.0000	1.5000	25.1663	83.2060
Energy +butbenz/ 1 -butbenz_c/ 1		-77.1805	-78.0000	1.5000	0.2985	83.5045

The force field error, i.e. the deviation between the ReaxFF and QC/Literature values, is calculated by Equation 3.1:

$$Error^{ReaxFF}=\{[v^{ReaxFF}-v^{QC/Lit}]/weight\}^2 \qquad \text{Equation (3.1)}$$

The sum of these errors, at the bottom of the last column in **fort.99**, is the value used in **fort.13** and **fort.79** to optimize the force field.

4. Potential functions

This section contains all the general ReaxFF-potential functions. In the current ReaxFF code all the energy contributions in this document are calculated regardless of system composition. All parameters that do not bear a direct physical meaning are named after the partial energy contribution that they appear in. For example, p_{val1} and p_{val2} are parameters in the valence angle potential function. Parameters with a more direct physical meaning, like the torsional rotational barriers (V_1, V_2, V_3) bear their more recognizable names. These potential functions were published in (Chenoweth et al., 2008a)

1. Overall system energy

Equation (1) describes the ReaxFF overall system energy.

$$E_{system} = E_{bond} + E_{lp} + E_{over} + E_{under} + E_{val} + E_{pen} + E_{coa} + E_{C2} + E_{triple} + E_{tors} + E_{conj} + E_{H-bond} + E_{vdWaals} + E_{Coulomb} \quad (1)$$

Below follows a description of the partial energies introduced in equation (1).

2. Bond Order and Bond Energy

A fundamental assumption of ReaxFF is that the bond order BO'$_{ij}$ between a pair of atoms can be obtained directly from the interatomic distance r_{ij} as given in Equation (2). In calculating the bond orders, ReaxFF distinguishes between contributions from sigma bonds, pi-bonds and double pi bonds.

$$BO'_{ij} = BO'^{\sigma}_{ij} + BO'^{\pi}_{ij} + BO'^{\pi\pi}_{ij} = \exp\left[p_{bo1}\cdot\left(\frac{r_{ij}}{r_o^{\sigma}}\right)^{p_{bo2}}\right] + \exp\left[p_{bo3}\cdot\left(\frac{r_{ij}}{r_o^{\pi}}\right)^{p_{bo4}}\right] + \exp\left[p_{bo5}\cdot\left(\frac{r_{ij}}{r_o^{\pi\pi}}\right)^{p_{bo6}}\right] \quad (2)$$

Based on the uncorrected bond orders BO', derived from Equation 1, an uncorrected overcoordination Δ' can be defined for the atoms as the difference between the total bond order around the atom and the number of its bonding electrons *Val*.

$$\Delta'_i = -Val_i + \sum_{j=1}^{neighbour(i)} BO'_{ij} \quad (3a)$$

ReaxFF then uses these uncorrected overcoordination definitions to correct the bond orders BO'$_{ij}$ using the scheme described in Equations (4a-f). To soften the correction for atoms bearing lone electron pairs a second overcoordination definition Δ'^{boc} (equation 3b) is used in equations 4e and 4f. This allows atoms like nitrogen and oxygen, which bear lone electron pairs after filling their valence, to break up these electron pairs and involve them in bonding without obtaining a full bond order correction.

$$\Delta_i^{'boc} = -Val_i^{boc} + \sum_{j=1}^{neighbours(i)} BO_{ij}^{'} \quad (3b)$$

$$\begin{aligned} BO_{ij}^{\sigma} &= BO_{ij}^{'\sigma} \cdot f_1(\Delta_i^{'},\Delta_j^{'}) \cdot f_4(\Delta_i^{'},BO_{ij}^{'}) \cdot f_5(\Delta_j^{'},BO_{ij}^{'}) \\ BO_{ij}^{\pi} &= BO_{ij}^{'\pi} \cdot f_1(\Delta_i^{'},\Delta_j^{'}) \cdot f_1(\Delta_i^{'},\Delta_j^{'}) \cdot f_4(\Delta_i^{'},BO_{ij}^{'}) \cdot f_5(\Delta_j^{'},BO_{ij}^{'}) \\ BO_{ij}^{\pi\pi} &= BO_{ij}^{'\pi\pi} \cdot f_1(\Delta_i^{'},\Delta_j^{'}) \cdot f_1(\Delta_i^{'},\Delta_j^{'}) \cdot f_4(\Delta_i^{'},BO_{ij}^{'}) \cdot f_5(\Delta_j^{'},BO_{ij}^{'}) \\ BO_{ij} &= BO_{ij}^{\sigma} + BO_{ij}^{\pi} + BO_{ij}^{\pi\pi} \end{aligned} \quad (4a)$$

$$f_1(\Delta_i,\Delta_j) = \frac{1}{2} \cdot \left(\frac{Val_i + f_2(\Delta_i^{'},\Delta_j^{'})}{Val_i + f_2(\Delta_i^{'},\Delta_j^{'}) + f_3(\Delta_i^{'},\Delta_j^{'})} + \frac{Val_j + f_2(\Delta_i^{'},\Delta_j^{'})}{Val_j + f_2(\Delta_i^{'},\Delta_j^{'}) + f_3(\Delta_i^{'},\Delta_j^{'})} \right) \quad (4b)$$

$$f_2(\Delta_i^{'},\Delta_j^{'}) = \exp(-p_{boc1} \cdot \Delta_i^{'}) + \exp(-p_{boc1} \cdot \Delta_j^{'}) \quad (4c)$$

$$f_3(\Delta_i^{'},\Delta_j^{'}) = -\frac{1}{p_{boc2}} \cdot \ln\left\{ \frac{1}{2} \cdot \left[\exp\left(-p_{boc2} \cdot \Delta_i^{'}\right) + \exp\left(-p_{boc2} \cdot \Delta_j^{'}\right) \right] \right\} \quad (4d)$$

$$f_4(\Delta_i^{'},BO_{ij}^{'}) = \frac{1}{1 + \exp(-p_{boc3} \cdot (p_{boc4} \cdot BO_{ij}^{'} \cdot BO_{ij}^{'} - \Delta_i^{'boc}) + p_{boc5})} \quad (4e)$$

$$f_5(\Delta_j^{'},BO_{ij}^{'}) = \frac{1}{1 + \exp(-p_{boc3} \cdot (p_{boc4} \cdot BO_{ij}^{'} \cdot BO_{ij}^{'} - \Delta_j^{'boc}) + p_{boc5})} \quad (4f)$$

A corrected overcoordination Δ_i can be derived from the corrected bond orders using equation (5).

$$\Delta_i = -Val_i + \sum_{j=1}^{neighbours(i)} BO_{ij} \quad (5)$$

Equation (6) is used to calculate the bond energies from the corrected bond orders $BO_{ij.}$

$$E_{bond} = -D_e^{\sigma} \cdot BO_{ij}^{\sigma} \cdot \exp\left[p_{be1}\left(1 - \left(BO_{ij}^{\sigma}\right)^{p_{be2}}\right)\right] - D_e^{\pi} \cdot BO_{ij}^{\pi} - D_e^{\pi\pi} \cdot BO_{ij}^{\pi\pi} \quad (6)$$

3. Lone pair energy

Equation (8) is used to determine the number of lone pairs around an atom. Δ_i^e is determined in Equation (7) and describes the difference between the total number of outer shell electrons (6 for oxygen, 4 for silicon, 1 for hydrogen) and the sum of bond orders around an atomic center.

$$\Delta_i^e = -Val_i^e + \sum_{j=1}^{neighbours(i)} BO_{ij} \qquad (7)$$

$$n_{lp,i} = \mathrm{int}\left(\frac{\Delta_i^e}{2}\right) + \exp\left[-p_{lp1}\cdot\left(2 + \Delta_i^e - 2\cdot\mathrm{int}\left\{\frac{\Delta_i^e}{2}\right\}\right)^2\right] \qquad (8)$$

For oxygen with normal coordination (total bond order=2, Δ_i^e=4), equation (8) leads to 2 lone pairs. As the total bond order associated with a particular O starts to exceed 2, equation (8) causes a lone pair to gradually break up, causing a deviation Δ_i^{lp}, defined in equation (9), from the optimal number of lone pairs $n_{lp,opt}$ (e.g. 2 for oxygen, 0 for silicon and hydrogen).

$$\Delta_i^{lp} = n_{lp,opt} - n_{lp,i} \qquad (9)$$

This is accompanied by an energy penalty, as calculated by equation (10).

$$E_{lp} = \frac{p_{lp2}\cdot\Delta_i^{lp}}{1+\exp\left(-75\cdot\Delta_i^{lp}\right)} \qquad (10)$$

4. Overcoordination

For an overcoordinated atom (Δ_i>0), equations (11a-b) impose an energy penalty on the system. The degree of overcoordination Δ is decreased if the atom contains a broken-up lone electron pair. This is done by calculating a corrected overcoordination (equation 11b), taking the deviation from the optimal number of lone pairs, as calculated in equation (9), into account.

$$E_{over} = \frac{\sum_{j=1}^{nbond} p_{ovun1}\cdot D_e^\sigma \cdot BO_{ij}}{\Delta_i^{lpcorr} + Val_i}\cdot\Delta_i^{lpcorr}\cdot\left[\frac{1}{1+\exp\left(p_{ovun2}\cdot\Delta_i^{lpcorr}\right)}\right] \qquad (11a)$$

$$\Delta_i^{lpcorr} = \Delta_i - \frac{\Delta_i^{lp}}{1+p_{ovun3}\cdot\exp\left(p_{ovun4}\cdot\left\{\sum_{j=1}^{neighbours(i)}\left(\Delta_j - \Delta_j^{lp}\right)\cdot\left(BO_{ij}^{\pi} + BO_{ij}^{\pi\pi}\right)\right\}\right)} \qquad (11b)$$

5. Undercoordination

For an undercoordinated atom (Δ_i<0), we want to take into account the energy contribution for the resonance of the π-electron between attached under-coordinated atomic centers. This is done by equations 12 where E_{under} is only important if the bonds between under-coordinated atom *i* and its under-coordinated neighbors *j* partly have π-bond character.

$$E_{under} = -p_{ovun5} \cdot \frac{1-\exp\left(p_{ovun6} \cdot \Delta_i^{lpcor}\right)}{1+\exp(-p_{ovun2} \cdot \Delta_i^{lpcor})} \cdot \frac{1}{1+p_{ovun7} \cdot \exp\left[p_{ovun8} \cdot \left\{\sum_{j=1}^{neighbours(i)} \left(\Delta_j - \Delta_j^{lp}\right) \cdot \left(BO_{ij}^{\pi} + BO_{ij}^{\pi\pi}\right)\right\}\right]} \quad (12)$$

6. Valence Angle Terms

6.1 Angle energy. Just as for bond terms, it is important that the energy contribution from valence angle terms goes to zero as the bond orders in the valence angle goes to zero. Equations (13a-g) are used to calculate the valence angle energy contribution. The equilibrium angle Θ_o for Θ_{ijk} depends on the sum of π-bond orders (*SBO*) around the central atom *j* as described in Equation (13d). Thus, the equilibrium angle changes from around 109.47 for sp^3 hybridization (π-bond=0) to 120 for sp^2 (π-bond=1) to 180 for sp (π-bond=2) based on the geometry of the central atom j and its neighbors. In addition to including the effects of π-bonds on the central atom j, Equation (13d) also takes into account the effects of over- and under-coordination in central atom j, as determined by equation (13e), on the equilibrium valency angle, including the influence of a lone electron pair. Val^{angle} is the valency of the atom used in the valency and torsion angle evaluation. Val^{angle} is the same as Val^{boc} used in equation (3c) for non-metals. The functional form of Equation (13f) is designed to avoid singularities when *SBO*=0 and *SBO*=2. The angles in Equations (13a)-(13g) are in radians.

$$E_{val} = f_7(BO_{ij}) \cdot f_7(BO_{jk}) \cdot f_8(\Delta_j) \cdot \left\{p_{val1} - p_{val1} \exp\left[-p_{val2}\left(\Theta_o(BO) - \Theta_{ijk}\right)^2\right]\right\} \quad (13a)$$

$$f_7(BO_{ij}) = 1 - \exp\left(-p_{val3} \cdot BO_{ij}^{p_{val4}}\right) \quad (13b)$$

$$f_8(\Delta_j) = p_{val5} - (p_{val5} - 1) \cdot \frac{2+\exp\left(p_{val6} \cdot \Delta_j^{angle}\right)}{1+\exp\left(p_{val6} \cdot \Delta_j^{angle}\right)+\exp\left(-p_{val7} \cdot \Delta_j^{angle}\right)} \quad (13c)$$

$$SBO = \sum_{n=1}^{neighbors(j)} \left(BO_{jn}^{\pi} + BO_{jn}^{\pi\pi}\right) + \left[1 - \prod_{n=1}^{neighbours(j)} \exp\left(-BO_{jn}^{8}\right)\right] \cdot \left(-\Delta_j^{angle} - p_{val8} \cdot n_{lp,j}\right) \quad (13d)$$

$$\Delta_j^{angle} = -Val_j^{angle} + \sum_{n=1}^{neighbours(j)} BO_{jn} \quad (13e)$$

$$\begin{aligned} &SBO2 = 0 \text{ if } SBO \le 0 \\ &SBO2 = SBO^{p_{val9}} \text{ if } 0 < SBO < 1 \\ &SBO2 = 2 - (2 - SBO)^{p_{val9}} \text{ if } 1 < SBO < 2 \\ &SBO2 = 2 \text{ if } SBO > 2 \end{aligned} \quad (13f)$$

$$\Theta_o(BO) = \pi - \Theta_{0,0} \cdot \left\{1 - \exp\left[-p_{val10} \cdot (2 - SBO2)\right]\right\} \quad (13g)$$

6.2 Penalty energy. To reproduce the stability of systems with two double bonds sharing an atom in a valency angle, like allene, an additional energy penalty, as described in Equations (14a) and

(14b), is imposed for such systems. Equation (9b) deals with the effects of over/undercoordination in central atom *j* on the penalty energy.

$$E_{pen} = p_{pen1} \cdot f_9(\Delta_j) \cdot \exp\left[-p_{pen2} \cdot \left(BO_{ij} - 2\right)^2\right] \cdot \exp\left[-p_{pen2} \cdot \left(BO_{jk} - 2\right)^2\right] \quad (14a)$$

$$f_9(\Delta_j) = \frac{2 + \exp\left(-p_{pen3} \cdot \Delta_j\right)}{1 + \exp\left(-p_{pen3} \cdot \Delta_j\right) + \exp\left(p_{pen4} \cdot \Delta_j\right)} \quad (14b)$$

6.3 Three-body conjugation term. The hydrocarbon ReaxFF potential contained only a four-body conjugation term (see section 7.2), which was sufficient to describe most conjugated hydrocarbon systems. However, this term failed to describe the stability obtained from conjugation by the $–NO_2$-group. To describe the stability of such groups a three-body conjugation term is included (equation 15).

$$E_{coa} = p_{coa1} \cdot \frac{1}{1 + \exp\left(p_{coa2} \cdot \Delta_j^{val}\right)} \cdot \exp\left[-p_{coa3} \cdot \left(-BO_{ij} + \sum_{n=1}^{neighbours(i)} BO_{in}\right)^2\right] \cdot \exp\left[-p_{coa3} \cdot \left(-BO_{jk} + \sum_{n=1}^{neighbours(i)} BO_{kn}\right)^2\right] \cdot$$
$$\exp\left[-p_{coa4} \cdot \left(BO_{ij} - 1.5\right)^2\right] \cdot \exp\left[-p_{coa4} \cdot \left(BO_{jk} - 1.5\right)^2\right] \quad (15)$$

7. Torsion angle terms

7.1 Torsion rotation barriers. Just as with angle terms we need to ensure that dependence of the energy of torsion angle ω_{ijkl} accounts properly for BO $\rightarrow$ 0 and for BO greater than 1. This is done by Equations (16a)-(16c).

$$E_{tors} = f_{10}(BO_{ij}, BO_{jk}, BO_{kl}) \cdot \sin\Theta_{ijk} \cdot \sin\Theta_{jkl} \cdot$$
$$\left[\frac{1}{2}V_1 \cdot \left(1 + \cos\omega_{ijkl}\right) + \frac{1}{2}V_2 \cdot \exp\left\{p_{tor1} \cdot \left(BO_{jk}^{\pi} - 1 + f_{11}(\Delta_j, \Delta_k)\right)^2\right\} \cdot \left(1 - \cos 2\omega_{ijkl}\right) + \frac{1}{2}V_3 \cdot \left(1 + \cos 3\omega_{ijkl}\right)\right] \quad (16a)$$

$$f_{10}(BO_{ij}, BO_{jk}, BO_{kl}) = \left[1 - \exp\left(-p_{tor2} \cdot BO_{ij}\right)\right] \cdot \left[1 - \exp\left(-p_{tor2} \cdot BO_{jk}\right)\right] \cdot \left[1 - \exp\left(-p_{tor2} \cdot BO_{kl}\right)\right] \quad (16b)$$

$$f_{11}(\Delta_j, \Delta_k) = \frac{2 + \exp\left[-p_{tor3} \cdot \left(\Delta_j^{angle} + \Delta_k^{angle}\right)\right]}{1 + \exp\left[-p_{tor3} \cdot \left(\Delta_j^{angle} + \Delta_k^{angle}\right)\right] + \exp\left[p_{tor4} \cdot \left(\Delta_j^{angle} + \Delta_k^{angle}\right)\right]} \quad (16c)$$

7.2 Four body conjugation term. Equations (17a-b) describe the contribution of conjugation effects to the molecular energy. A maximum contribution of conjugation energy is obtained when successive bonds have bond order values of 1.5 as in benzene and other aromatics.

$$E_{conj} = f_{12}(BO_{ij}, BO_{jk}, BO_{kl}) \cdot p_{cot1} \cdot \left[1 + \left(\cos^2\omega_{ijkl} - 1\right) \cdot \sin\Theta_{ijk} \cdot \sin\Theta_{jkl}\right] \quad (17a)$$

$$f_{12}(BO_{ij}, BO_{jk}, BO_{kl}) = \exp\left[-p_{cot2} \cdot \left(BO_{ij} - 1\frac{1}{2}\right)^2\right] \cdot \exp\left[-p_{cot2} \cdot \left(BO_{jk} - 1\frac{1}{2}\right)^2\right] \cdot \exp\left[-p_{cot2} \cdot \left(BO_{kl} - 1\frac{1}{2}\right)^2\right] \quad (17b)$$

8. Hydrogen bond interactions

Equation (18) described the bond-order dependent hydrogen bond term for a X-H—Z system as incorporated in ReaxFF.

$$E_{Hbond} = p_{hb1} \cdot \left[1 - \exp\left(p_{hb2} \cdot BO_{XH}\right)\right] \cdot \exp\left[p_{hb3}\left(\frac{r_{hb}^{o}}{r_{HZ}} + \frac{r_{HZ}}{r_{hb}^{o}} - 2\right)\right] \cdot \sin^{8}\left(\frac{\Theta_{XHZ}}{2}\right) \quad (18)$$

9. Correction for C_2

ReaxFF erroneously predicts that two carbons in the C_2-molecule form a very strong (triple) bond, while in fact the triple bond would get de-stabilized by terminal radical electrons, and for that reason the carbon-carbon bond is not any stronger than a double bond. To capture the stability of C_2 we introduced a new partial energy contribution (E_{C2}). Equation (19) shows the potential function used to de-stabilize the C_2 molecule:

$$\begin{aligned} E_{C2} &= k_{c2} \cdot \left(BO_{ij} - \Delta_i - 0.04 \cdot \Delta_i^4 - 3\right)^2 \quad \text{if } BO_{ij} - \Delta_i - 0.04 \cdot \Delta_i^4 > 3 \\ E_{C2} &= 0 \qquad\qquad\qquad\qquad\qquad\quad \text{if } BO_{ij} - \Delta_i - 0.04 \cdot \Delta_i^4 \le 3 \end{aligned} \quad (19)$$

where Δ_i is the level of under/overcoordination on atom i as obtained from subtracting the valency of the atom (4 for carbon) from the sum of the bond orders around that atom and k_{c2} the force field parameter associated with this partial energy contribution.

11. Triple bond energy correction.

To describe the triple bond in carbon monoxide a triple bond stabilization energy is used, making CO both stable and inert. This energy term only affects C-O bonded pairs. Equation (20) shows the energy function used to describe the triple bond stabilization energy.

$$E_{trip} = p_{trip1} \exp\left[-p_{trip2}\left(BO_{ij} - 2.5\right)^2\right] \cdot \frac{\exp\left[-p_{trip4} \cdot \left(\sum_{k=1}^{neighbours(i)} BO_{ik} - BO_{ij}\right)\right] + \exp\left[-p_{trip4} \cdot \left(\sum_{k=1}^{neighbours(j)} BO_{jk} - BO_{ij}\right)\right]}{1 + 25 \cdot \exp\left[p_{trip3}\left(\Delta_i + \Delta_j\right)\right]}$$

12. Nonbonded interactions

In addition to valence interactions which depend on overlap, there are repulsive interactions at short interatomic distances due to Pauli principle orthogonalization and attraction energies at long distances due to dispersion. These interactions, comprised of van der Waals and Coulomb forces, are included for *all* atom pairs, thus avoiding awkward alterations in the energy description during bond dissociation.

12.1 Taper correction. To avoid energy discontinuities when charged species move in and out of the non-bonded cutoff radius ReaxFF employs a Taper correction, as developed by de Vos Burchart (1995). Each nonbonded energy and derivative is multiplied by a Taper-term, which is taken from a distance-dependent 7th order polynomial (equation 21)).

$$Tap = Tap_7 \cdot r_{ij}^7 + Tap_6 \cdot r_{ij}^6 + Tap_5 \cdot r_{ij}^5 + Tap_4 \cdot r_{ij}^4 + Tap_3 \cdot r_{ij}^3 + Tap_2 \cdot r_{ij}^2 + Tap_1 \cdot r_{ij} + Tap_0 \quad (21)$$

The terms in this polynomal are chosen to ensure that all 1st, 2nd and 3rd derivatives of the non-bonded interactions to the distance are continuous and go to zero at the cutoff boundary. To that end, the terms Tap_0 to Tap_7 in equation (21) are calculated by the scheme in equation (22), where R_{cut} is the non-bonded cutoff radius.

$$
\begin{aligned}
Tap_7 &= 20/R_{cut}^7 \\
Tap_6 &= -70/R_{cut}^6 \\
Tap_5 &= 84/R_{cut}^5 \\
Tap_4 &= -35/R_{cut}^4 \\
Tap_3 &= 0 \\
Tap_2 &= 0 \\
Tap_1 &= 0 \\
Tap_0 &= 1
\end{aligned}
\qquad (22)
$$

12.2 van der Waals interactions. To account for the van der Waals interactions we use a distance-corrected Morse-potential (Equations. 23a-b). By including a shielded interaction (Equation 23b) excessively high repulsions between bonded atoms (1-2 interactions) and atoms sharing a valence angle (1-3 interactions) are avoided.

$$
E_{vdWaals} = Tap \cdot D_{ij} \cdot \left\{ \exp\left[\alpha_{ij} \cdot \left(1 - \frac{f_{13}(r_{ij})}{r_{vdW}} \right) \right] - 2 \cdot \exp\left[\frac{1}{2} \cdot \alpha_{ij} \cdot \left(1 - \frac{f_{13}(r_{ij})}{r_{vdW}} \right) \right] \right\} \qquad (23a)
$$

$$
f_{13}(r_{ij}) = \left[r_{ij}^{p_{vdW}} + \left(\frac{1}{\gamma_w} \right)^{p_{vdW}} \right]^{\frac{1}{p_{vdW}}} \qquad (23b)
$$

12.3 Coulomb Interactions

As with the van der Waals-interactions, Coulomb interactions are taken into account between *all* atom pairs. To adjust for orbital overlap between atoms at close distances a shielded Coulomb-potential is used (Equation 24).

$$
E_{coulomb} = Tap \cdot C \cdot \frac{q_i \cdot q_j}{\left[r_{ij}^3 + \left(1/\gamma_{ij} \right)^3 \right]^{1/3}} \qquad (24)
$$

Atomic charges are calculated using the Electron Equilibration Method (EEM)-approach (Mortier et al., 1986). The EEM charge derivation method is similar to the QEq-scheme; the only differences, apart from parameter definitions, are that EEM does not use an iterative scheme for hydrogen charges (as in QEq) and that QEq uses a more rigorous Slater orbital approach to account for charge overlap.

5. Program structure

Figure 5.1 shows the general flow diagram of a typical molecular dynamics simulation using the ReaxFF code, which contains three main sections (ffopt.f, reac.f and poten.f). This flow diagram does not include the force field optimization section, which is essentially a shell written around the general ReaxFF code – this shell is predominantly located in the ffopt.f section.

For external codes aiming to link into the ReaxFF code – for example, a Monte Carlo external shell – the encalc subroutine is essentially at the core – this routine, which some minor initiation, can essentially be called as a force our energy library.

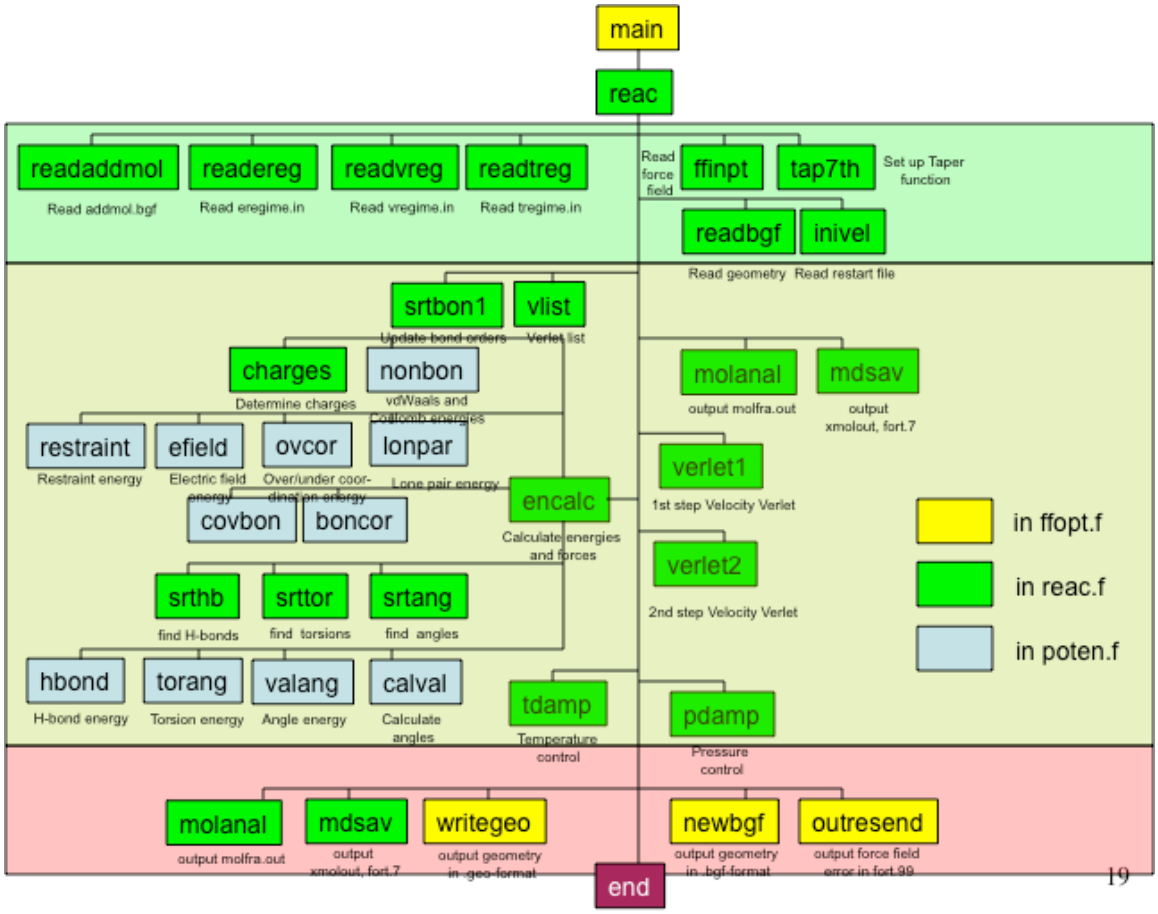


Figure 5.1. Flow diagram for a molecular dynamics simulation using the Standalone ReaxFF program.

6. Performance

Calculation speed for ReaxFF greatly depends on the atom connectivity. Slowest calculation speed is obtained for high-number density crystal systems requiring valence angle and torsion potentials (e.g. diamond). Furthermore, covalent materials that employ the bond order correction (Equations 3-4, Chapter 4), like carbon, are far more computationally demanding then materials that only use an uncorrected 2-body bond order (Equation 2, Chapter 4), like most metals.

Table 6.1 shows the ReaxFF calculation speed for some benchmark systems; these numbers are for the standalone code using a gnu-fortran 77-compiler without any optimization options. The Intel-compiler provides about 40% speed-up. To date, the largest system the standalone, non-parallel ReaxFF has been applied for contained about 5000 atoms. Ongoing developments in parallelizing the ReaxFF-code have greatly increase the feasible system size in parallel environments like LAMMPS and ADF (Aktulga et al., 2012; Shan et al., 2013; Zybin et al., 2010). Vashishta and co-workers have reported very large (>>1,000,000 parallel MD-simulations using their optimized USC/ReaxFF parallel code (Chen et al., 2010; Nakano et al., 2007; Nakano et al., 2008; Nomura et al., 2012; Nomura et al., 2007; Vashishta et al., 2006; Vedadi et al., 2010).

At low and medium temperatures (0-1500K) ReaxFF can run with time-steps of up to 0.25 femtoseconds and retain reasonable energy conservation. At higher temperatures smaller time-steps are required to retain energy conservation. This is also dependent on

Table 6.1: Benchmark runs for some crystal and molecular systems. Simulations were performed using NVE-molecular dynamics on a single-processor Dell T3600-workstation using a XEON E5-1620, 3.6 GHz processor with 16GB memory; EEM charges were updated every step. The Diamond crystal calculations involved valence and torsion angle energies, the Si(α) calculations did only involve valence angle energies.

System	#atoms	Time/MD iteration (s)
Diamond crystal	64	0.08
Diamond crystal	512	0.7
Si(α) crystal	64	0.012
Si(α) crystal	512	0.108
13 Methane molecules	65	0.022
104 Methane molecules	520	0.033
Fe-crystal	64	0.025
Fe-crystal	512	0.27

atom type – for systems with only heavy atoms (e.g. pure metal) larger time-steps may be feasible – this can be evaluated by performing an NVE-simulation at the temperature target; good energy conservation means that the selected time-step is acceptable.

7. Currently available execution environments

Besides the standalone ReaxFF program, which is described in this manual, there are a number of commercial and open-source codes available that contain a ReaxFF force engine. Most of these can read in the ReaxFF force field format, as described in this manual. We recommend that the energy output of any of these codes is first always compared to the standalone code – in case of discrepancy, the standalone code is always correct. Small energy deviations, for example due to different EEM-solvers used in parallel programs, are acceptable. Below follows a brief description of the execution environments that we are familiar with.

LAMMPS/ReaxFF. LAMMPS is an open-source, free, simulation environment that incorporates a large number of simulation methods, amongst which a ReaxFF force engine. LAMMPS also contains metadynamics tools – some of which can be used in conjunction with ReaxFF. LAMMPS enables massively parallel ReaxFF simulations (Shan et al., 2013; Zybin et al., 2010). The current ReaxFF implementation in LAMMPS is written in C++; this rewrite was performed by Metin Aktulga (Aktulga et al., 2012). We have used LAMMPS/ReaxFF for a number of systems – overall its integration runs very well, but we strongly recommend comparison between LAMMPS and standalone ReaxFF before running any large-scale simulations. LAMMPS can be downloaded from http://lammps.sandia.gov/.

ADF/ReaxFF. ADF is licensed by Scientific Computing & Modeling (SCM Amsterdam, the Netherlands, http://www.scm.com/). While originally focused on DFT calculations, recently SCM has integrated ReaxFF into ADF and is continuously expanding its functionality. The core of the ReaxFF engine in ADF is based on the standalone, fortran-77 code described here, which was re-mapped into a Fortran-95 environment to make the method more memory efficient; furthermore a significant optimization of the code was performed, making ADF/ReaxFF about 10 times faster than the standalone code described in this manual. ADF/ReaxFF also supports parallel molecular dynamics and incorporates various metadynamics tools.

Material Studio. Recently, Material Studio, licensed by Accelrys, started supporting ReaxFF, via the GULP module (http://accelrys.com/products/materials-studio/). This ReaxFF/GULP module was developed by Julian Gale – we believe this module to provide correct ReaxFF energies and forces, but recommend comparison with the standalone. At this moment, Material Studio only supports single-processor ReaxFF simulations.

8. Simulation examples and analysis tools

This section describes the use of three programs external to the ReaxFF standalone code that can be use to analyze the ReaxFF results. These three programs are the msd-t_list program (freeware, available by request from Adri van Duin) for diffusion analysis, the VMD graphical program (freeware, downloadable from http://www.ks.uiuc.edu/Research/vmd/) for calculating a radial distribution function and the molfraanal-program (freeware, available by request from Adri van Duin) for analyzing reaction kinetics.

Diffusion constant. To calculate the diffusion constant, we employ the msd-t_list program, which does the statistical analysis and provides a mean square displacement (MSD) versus time report for selected atoms. Below are the steps required to calculate a diffusion constant with the msd-t_list program.

1) perform a MD-simulation at the desired system composition and temperature/pressure settings. We recommend to only calculate a diffusion constant using the NVE ensemble on a system pre-equilibrated at the desired temperature and pressure using NPT-ensemble simulations. Use itrout=1 in the **control**-file to generate a unfolded trajectory in diff_traj.xyz; do not use the normal **xmolout** trajectory, since atoms moving across the periodic boundary will significantly affect the MSD.

2) Make a new directory for the diffusion analysis; copy the diff_traj.xyz into this directory and rename it input.xyz

3) Generate a compute-msd.list input file indicating the atoms to be included in the MSD-analysis (see **Example 8.1**).

Example 8.1: compute-msd.list file, defining the atom numbers of the atoms to be included in the MSD-analysis.

```
6      !Number of atoms
1
50
99
158
217
512
```

- Call the msd-t_list program – provide the number of frames, the number of frames to be included in the MSD-analysis and the time between frames (see **Example 8.2**).

Example 8.2: Example of normal communication with the msd_t_list program. This example concerns a input file with 512 atoms, from which 6 atoms are selected for the MSD analysis (see **Example 8.1**). The input file contains 1602 frames, which were saved every 250 steps, with a stepsize of 0.25 fs., making delta t 0.001*250*0.25=0.0625.

```
$ msd-t_list
 number of times to read:
1602
 maximum lag (in number of frames
1602
 delta t between frames (in ps):
0.0625
1 2
 2 3
.....
.....
249 510
 250 511

movable indexes read 250
 calculation will start

 points read:  500
 points read:  1000
 points read:  1500
```

The msd-t_list program subsequently produces a out.msd-t output file, which provides the MSD (in Angstrom*Angstrom vs. time (in picoseconds). **Figure 8.1** shows a generated plot from the out.msd-t file. These in- and output files are available in the MoNa_1000K.tar simulation example.

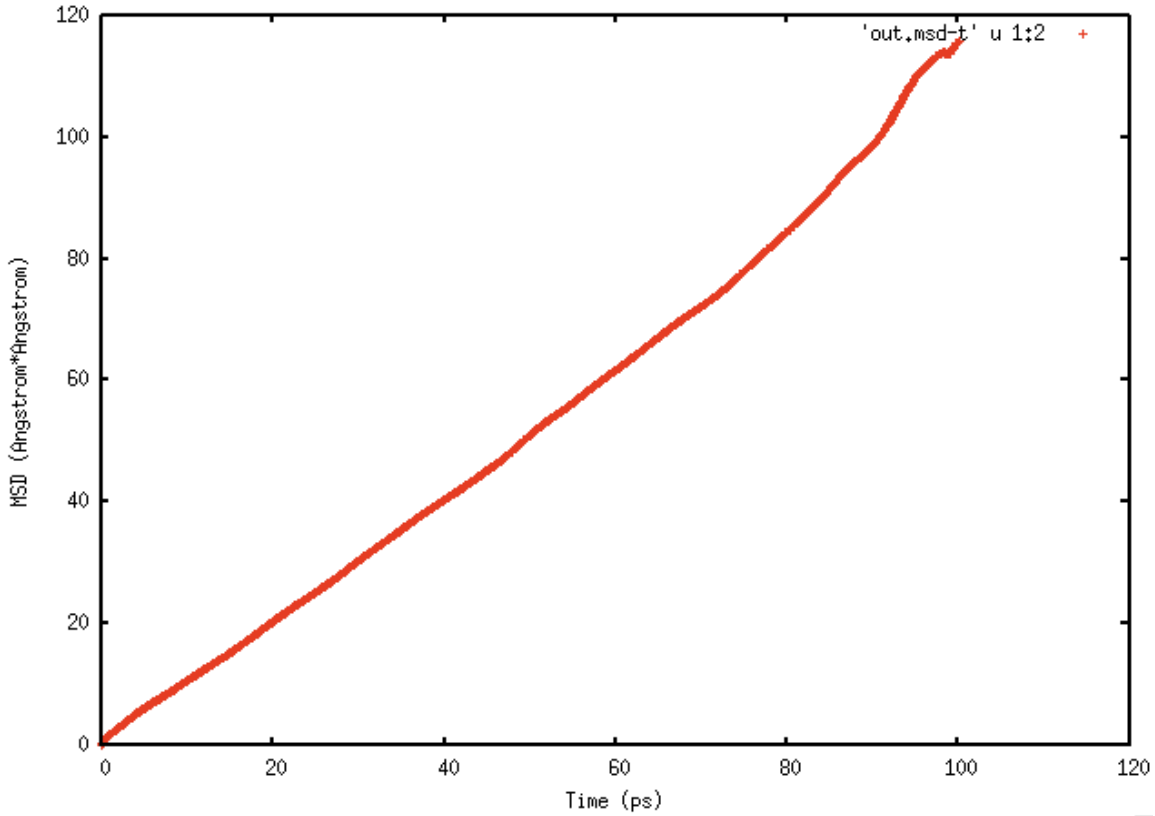


Figure 8.1: MSD versus time for Mo-atoms in a Na/Fe matrix, as calculated using the msd-t_list program.

The slope of the line in **Figure 8.1** is directly related to the diffusion constant – for 3-dimensional diffusion the diffusion constant is the slope of this line divided by 6. Please note that a long simulation is required to filter out short-time vibrational effects or ballistic diffusion effects. Also, beyond time=0.5(simulation time), where (simulation time) is 100 ps. in **Figure 8.1** the accuracy of the diffusion analysis diminishes due to poorer statistical sampling – as such, it is advised to only consider the first half of the MSD versus time plot for the diffusion constant calculation.

Radial distribution function. The radial distribution function g(r) has many uses in the analysis of MD-simulations – for example, it can provide information regarding the physical state (solid/liquid/gas) of the system and it can directly provide neighbour information. There are many freeware programs available that can be used to calculate g(r); in this example we describe the use of the VMD program.

Using the same simulation example as for the diffusion (files in MoNiFe_1000K.tar) here is the procedure for calculating g(r) in VMD.

- Give the 'vmd –xyz xmolout' command to open the trajectory in VMD
- VMD should open in a graphical window, see **Figure 8.2**

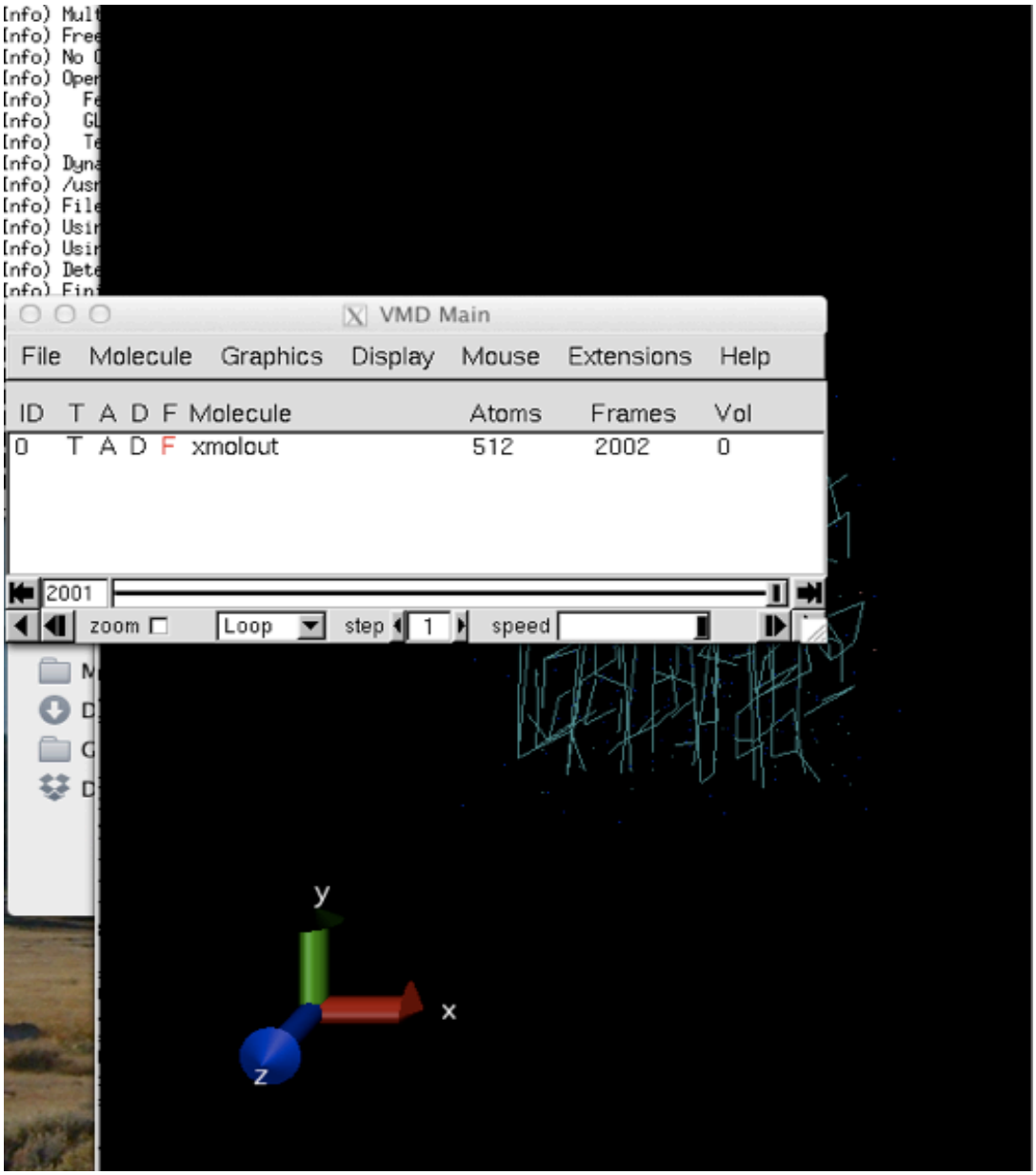


Figure 8.2: Screen image after opening VMD

- In the VMD-Main window, go to 'Extensions/Analysis/Radial Pair Distribution Function g(r)', see **Figure 8.3**.
- Define in the Utilities-menu the cell parameters of the system (in this case, a=18.60, b=19.86, c=40.23; alpha=beta=gamma=90). We have only used VMD for orthogonal systems and do not know its accuracy for non-orthogonal ReaxFF trajectories.
- Select the atom types using the 'name At' VMD concention, where At is the atom name (see **Figure 8.3** where we use a Na—Na pair).
- Select use PBC if the cell parameters are defined
- Select the output options (in this case, only a file is saved, no graphical display is asked for)
- Press 'Computer g(r)'; if the calculation is successfull, a file-save window will open (**Figure 8.4**).

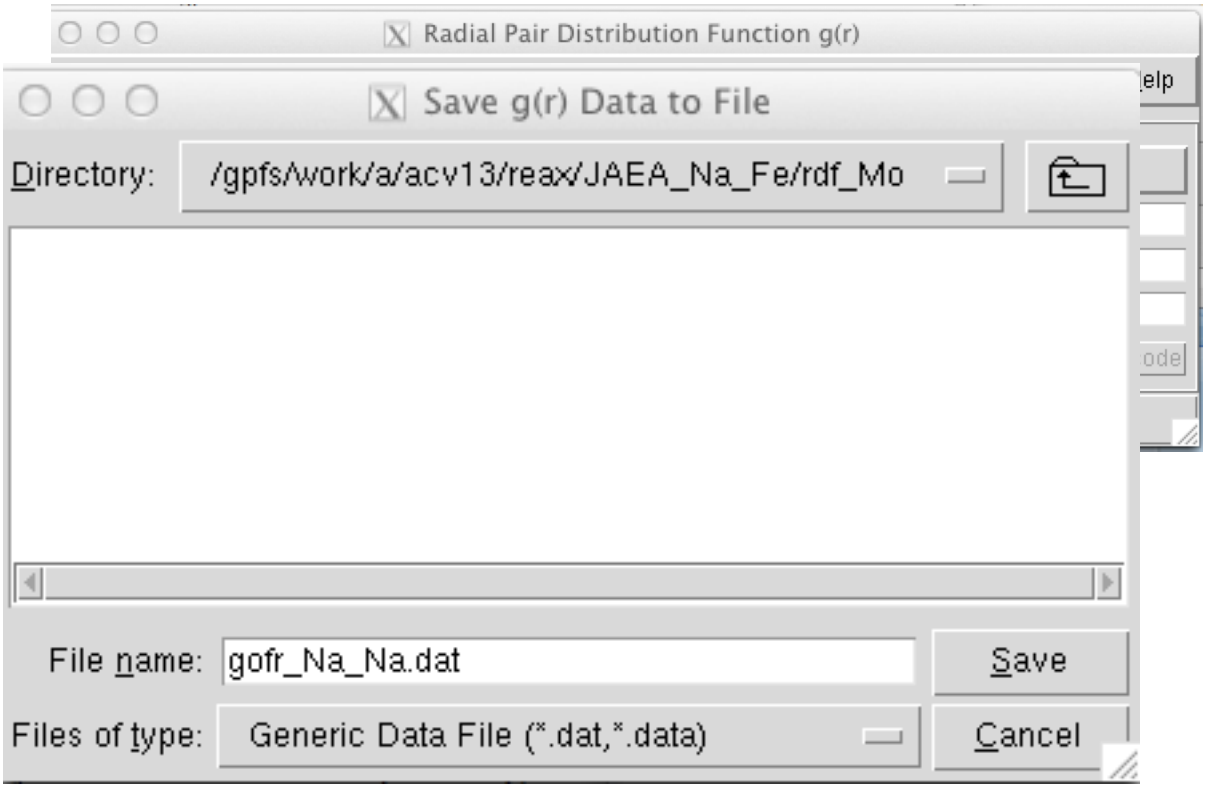


Figure 8.3: Radial distribution calculation menu in VMD.

Figure 8.4: g(r) data save-to-fie menu in VMD

- Provide an output name (here gofr_Na_Na.dat). This output file contains 3 columns, first the distance (r), second the g(r) and the third column the integral of g(r), which indicates the total number of neighbors from 0 to r.
- **Figure 8.5** shows that results from this analysis, indicating that the Na-phase is a liquid, as there is a non-zero g(r) between the first and the second peak, and both peaks are very broad. Furthermore, the integral over the first peak indicates 12 neighbors, indicating a fcc- or hcp-like close-packed liquid structure.

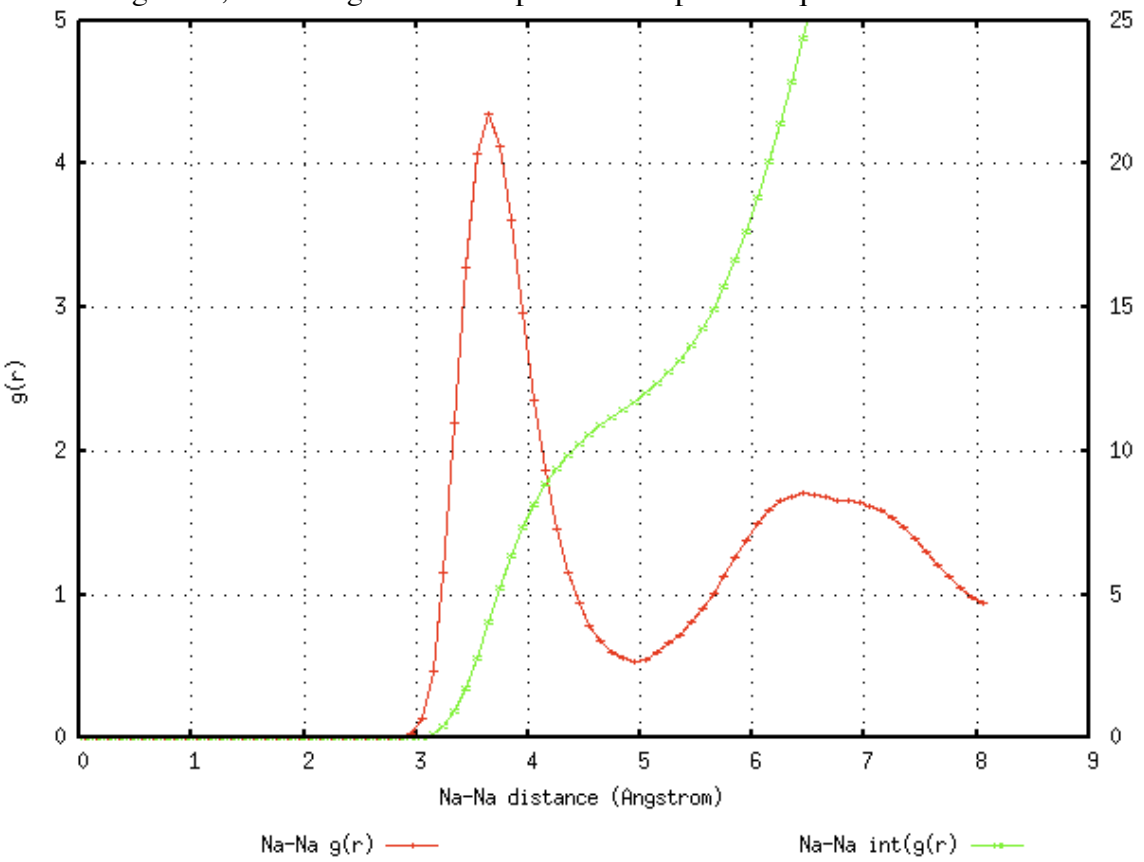


Figure 8.5. Results from the radial distribution analysis for the Na-Na pair, indicating a molten, fcc/hcp-like sodium phase.

Molecular analysis. While the ReaxFF program performs its own molecular analysis, the format of the **molfra.out**-file makes it not straightforward to produce a graphical representation of the molecular composition versus time – which is what is usually required to analyse the reaction kinetics. For that reason, the molfraanal-program was developed. This is a relatively straightforward script, that essentially converts the **molfra.out**-file into a column-format, where the major species can be tracked as a function of time. The output from the molfraanal-program (**fr1.dat**) can be easily plotted using freeware tools like gnuplot or can be imported into spreadsheet programs.

In the following example we use the **molfra.out** file from a short simulation on a system containing 20 acetylene molecules (C2H2) and 20 O-atoms to demonstrate the use of molfraanal. **Example 8.3** contains the **molfra.out**-file, as obtained using the ReaxFF program:

Example 8.3. Sample molfra.out-file from a ReaxFF simulation on acetylene reactions with atomic oxygen at T=1000K, density 0.09 kg/dm^3.

```
Bond order cutoff:0.3000
Iteration Freq. Molecular formula          Molecular mass
      0  20 x  C2H2                      26.0160
      0  20 x  O                         15.9990
 Total number of molecules: 40
 Total number of atoms: 100
 Total system mass:  840.3
Iteration Freq. Molecular formula          Molecular mass
   2500  17 x  C2H2                      26.0160
   2500   2 x  C2H2O                      42.0150
   2500   1 x  C2H2O2                      58.0140
   2500  16 x  O                         15.9990
 Total number of molecules: 36
 Total number of atoms: 100
 Total system mass:  840.3
Iteration Freq. Molecular formula          Molecular mass
   5000  16 x  C2H2                      26.0160
   5000   3 x  C2H2O                      42.0150
   5000   1 x  CH2O                      30.0150
   5000   1 x  CO                        27.9990
   5000  15 x  O                         15.9990
 Total number of molecules: 36
 Total number of atoms: 100
 Total system mass:  840.3
```

Using the following command-line sequence (**Example 8.4**), molfraanal can convert this molfra.out file into a columnal format, while filtering out the minor products (in this case, CO and CH2O, which have a maximum frequency smaller than the cutoff).

Example 8.4: Command-line sequence for running the molfraanal-program.

```
$ ls
molfra.out
$ molfraanal
 Input file (probably molfra.out) ?
molfra.out
 Cutoff for compound frequency ?
2
STOP Normal end of program statement executed

$ cat fr1.dat
      Iteration             C2H2           O              C2H2O          Others
         0                  20             20              0              0
      2500                  17             16              2              1
      5000                  16             15              3              2
```

As **Example 8.4** indicates, the **fr1.dat** file format is quite easy to turn into a graph – this analysis tool has been used in a large number of ReaxFF publications (e.g. (Chenoweth et al., 2005; Chenoweth et al., 2008a). **Example 8.5** shows how gnuplot (freeware) can be used to make a graphical representation of the reaction kinetics, **Figure 8.6** shows the gnuplot output.

For systems with a large amount of products, the molfraanal automatically creates a **fr2.dat** and **fr3.dat** output file – thus allowing tracking of up to 30 different molecules.

Example 8.5: Using gnuplot to plot the **fr1.dat**-output from the molfraanal-program.

```
$ gnuplot
        G N U P L O T
        Version 4.0 patchlevel 0
        last modified Thu Apr 15 14:44:22 CEST 2004
        System: Linux 2.6.18-128.el5

        Copyright (C) 1986 - 1993, 1998, 2004
        Thomas Williams, Colin Kelley and many others

        This is gnuplot version 4.0.  Please refer to the documentation
        for command syntax changes.  The old syntax will be accepted
        throughout the 4.0 series, but all save files use the new syntax.

        Type `help` to access the on-line reference manual.
        The gnuplot FAQ is available from
                http://www.gnuplot.info/faq/

        Send comments and requests for help to
                <gnuplot-info@lists.sourceforge.net>
        Send bugs, suggestions and mods to
                <gnuplot-bugs@lists.sourceforge.net>
Terminal type set to 'x11'
gnuplot> plot 'fr1.dat' u 1:2 t'C2H2','fr1.dat' u 1:3 t'O','fr1.dat' u 1:4 t'C2H2O','fr1.dat' u 1:5 t'Others'
gnuplot> set style data lp
gnuplot> replot
gnuplot> set ylabel 'Number of molecules'
gnuplot> set xlabel 'MD-iteration (0.25fs)'
gnuplot> replot
```

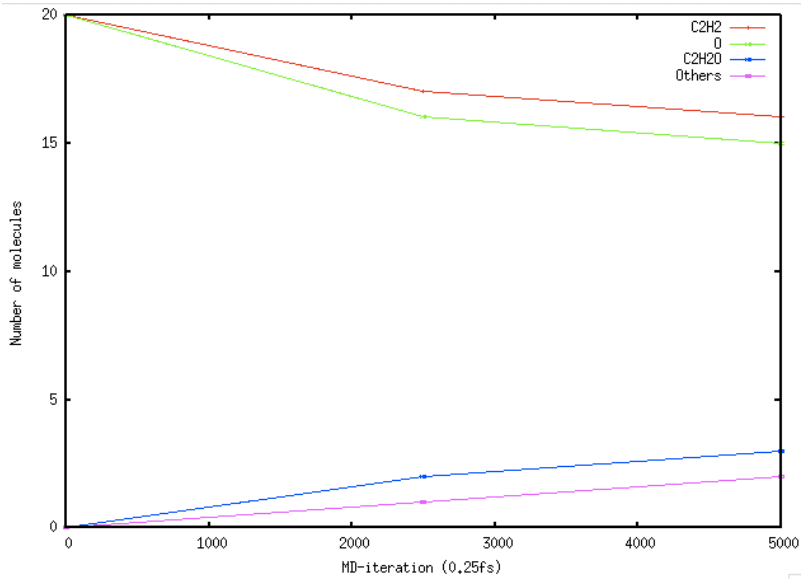


Figure 8.5: Gnuplot graph based on the fr1.dat output file generated in **Example 8.4**

Chapter 9. The e-ReaxFF method – inclusion of explicit electrons or holes

Introduction. As discussed in the previous sections of this manual, the ReaxFF method features a polarizable, geometry dependent charge calculation method – the EEM method. This enables ReaxFF to give a realistic description of the charge distribution within a molecular or condensed phase system. However – this charge calculation method is not accurate for the calculation of ionization potential and/or electron affinity – where the system fully accepts an electron or a hole. The main reason for this inaccuracy – which limits the reliability of the ReaxFF method for applications to electrochemically active interfaces, like batteries and fuel cells – is that the bond orders in ReaxFF are fully de-coupled from the charges – as such, a hydrogen atom that completely loses its electron through an ionization (H^+) is still considered capable of forming bonds. Similarly, a methane molecule in the ReaxFF description can be ionized and still retain a valency of 4 – thus creating a stable CH_4^+ radical cation. This is incorrect – the CH_4^+ radical cation only has 3 remaining valence electrons, and as such it should be unstable and form a CH_3^+ carbocation and a H-radical with zero or a small barrier. Similarly, ReaxFF predicts both H_2O and OH-radical to have positive electron affinities – i.e. both molecules can accept an additional electron forming, respectively, a H_2O^- radical anion and a OH^- anion. This is incorrect – while the OH-radical indeed has a strongly positive electron affinity, H_2O has a negative electron affinity, since the H_2O^- radical anion is highly unstable.
In order to remediate this incorrect description of electron affinity the ReaxFF concept was extended with the ability to describe explicit electrons or holes – this extension was named e-ReaxFF and is described in (Islam et al., 2016; Islam and van Duin, 2016). These electrons/holes carry a fixed -1/+1 charge. If these electrons or holes are captured by an atom then this results in a change in the number of valence electrons – following periodic table rules. For example – a carbon atom that capture a hole obtains an electron configuration of boron – which gives it a valency of 3. If the same carbon captures an electron it obtains a nitrogen electron configuration – giving it 5 valence electrons, of which 2 typically pair up as a lone pair, yielding an effective valency of 3 for the ReaxFF over- and undercoordination energy terms.

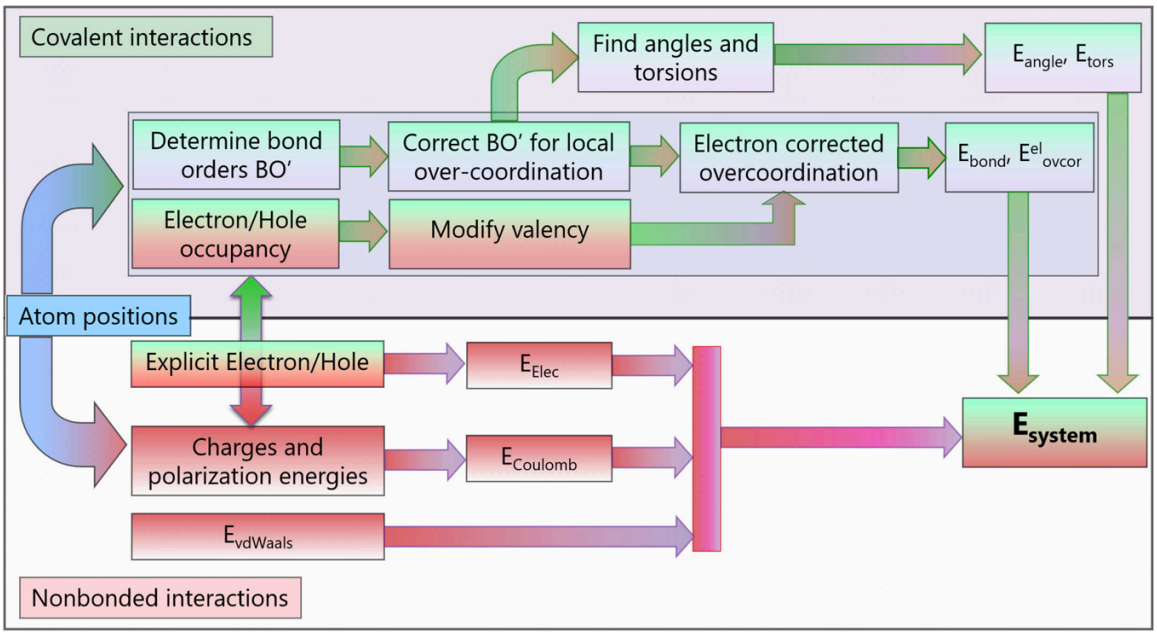


Figure 9.1; Schematic diagram indicating the e-ReaxFF concept.

Figure 9.1 schematically shows the e-ReaxFF concept. Note that, due to the fixed electron/hole charge, e-ReaxFF still keeps the bond-order based terms separate from the non-bonded terms – which is highly desirable from a computational expense perspective. As Figure 9.2 indicates – e-ReaxFF enables a far more accurate description of electron affinities for molecules – e-ReaxFF clearly identifies the low- or negative electron affinity for closed-shell molecules while recognizing the high electron affinity for radicals – see, in particular, the e-ReaxFF performance for methane, OH-radical and water.

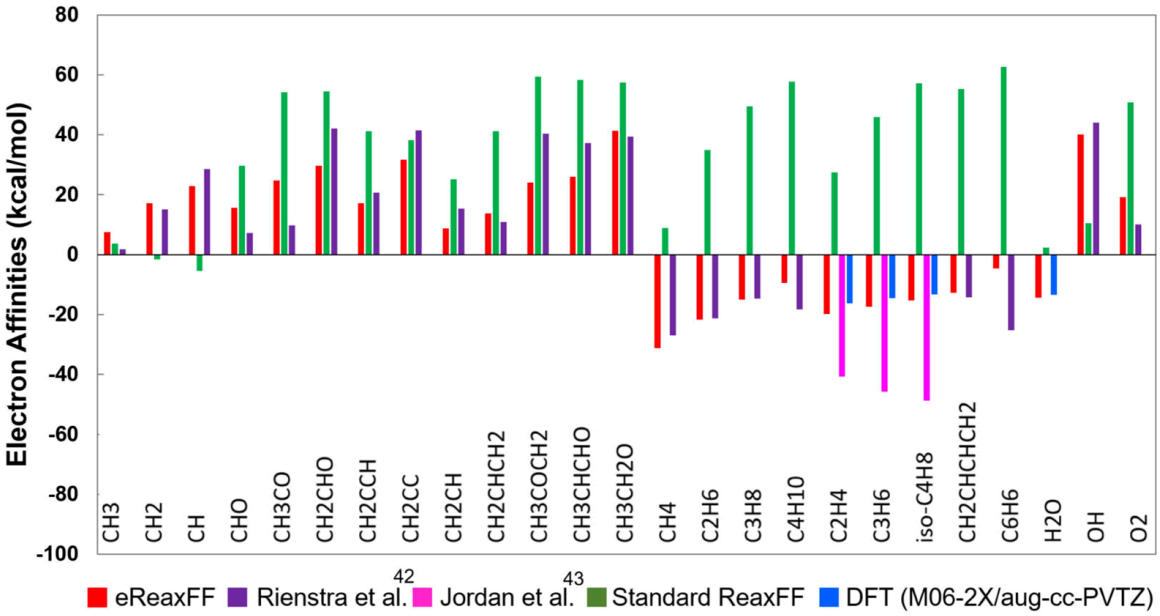


Figure 9.2 Comparison of ReaxFF and e-ReaxFF electron affinities to DFT-results.

It is important to note that in e-ReaxFF the electron/hole is essentially a particle – however, the electron/hole particle does not have to reside on a single atom – it can distribute itself over multiple atoms. In the current e-ReaxFF force fields, the mass of the electron/hole particles is 1 amu – this is mainly done to enable normal size MD-steps (typically 0.25 fs). The mass of the electron/hole particle can be straightforwardly modified in the force field file (see Example 2.8 and associated discussion). Furthermore, while e-ReaxFF can, in principle, work with EEM-charges, significant artificial charge transfer is typically observed when an explicit electron or hole is added to a molecule. EEM is not aware of chemical bond structure – as such, we often observe that a significant amount of the charge of a molecule/electron or molecule/hole pair is transferred to a neighboring molecule – which is physically suspect. As such, we recommend using the ACKS2 charge calculation – an extension of the EEM concept developed by Toon Verstraelen (Senftle et al., 2016; Verstraelen et al., 2013) instead of EEM. ACKS2 takes bonding structure into account during the charge calculation – resulting in effectively zero charge transfer between remote molecules. ACKS2 is currently supported in the standalone ReaxFF code and in ADF/ReaxFF.
e-ReaxFF is a relatively new method – and as such its capabilities and limitations are not yet fully established.

Equations Equations 9.1-9.3 describe the current e-ReaxFF formulation. These are straightforwardly added up to the normal ReaxFF equations.

In e-ReaxFF the electron/hole is described as a Gaussian wave (Equation 9.1)

$$\psi \infty exp(-\alpha(r-r')^2) \qquad (1)$$

This Gaussian wave interacts with the nuclei through a pairwise interaction, represented in Equation 9.2:

$$E_{nucl(i)-elec(j)} = -\frac{1}{4\pi\varepsilon_o}\beta \sum_{i,j} \frac{Z_i}{R_{ij}} erf\left(\sqrt{2\alpha}R_{ij}\right) \qquad (2)$$

Furthermore, the electron/hole carries a formal -1/+1 charge, which interacts with the other atomic charges through the regular ReaxFF shielded Coulomb interaction (Equation 8.24) which is also subject to the 7th order Taper function (Equation 8.22).

In addition to equations 9.2 and the Coulomb interaction, the electron/hole also affect the number of valence electrons of their neighboring atoms through Equation 9.3:

$$n_{el}=exp(-p_{val}*R_{ij}^2) \qquad (3)$$

This represents the amount of the hole/electron that belongs to a particular atom – note that this means that multiple atoms can own the electron/hole and can thus be affected. The number of valence electrons of an atom (Val_i equation 9.4) is subsequently modified by n_{el} from equation 9.3 using atom-specific rules (e.g. Carbon loses a valence electron both by hole/electron capture, Nitrogen loses a valence electron by electron capture but

gains a valence electron through hole capture) to get the amount of over/undercoordination Δ_i through equation 9.4.

$$\Delta_i = -\mathrm{Val}_i + \sum_{j=1}^{\mathrm{neighbor}(i)} \mathrm{BO}_{ij} \qquad (4)$$

Δ , derived from Equation 4, is subsequently modified to Δ_i^{xel} through a bond-type and atom-type dependent equation – resulting in an electron/hole corrected valence electron count for the atom (Equation 9.5) which subsequently enters the normal ReaxFF over/undercoordination terms (Equations 4.11a, 4.11b and 4.12)

$$\Delta_i^{\mathrm{xel}} = \Delta_i \cdot \exp\left(-p_i^{\mathrm{xel2}} \cdot n_{\mathrm{el}} \cdot \frac{\sum_{j=1}^{\mathrm{nbond}} \mathrm{BO}_{ij} \cdot p_{ij}^{\mathrm{xel1}}}{\sum_{j=1}^{\mathrm{neighbor}(i)} \mathrm{BO}_{ij}} \right) \qquad (5)$$

At this moment, only the under/overcoordination terms in ReaxFF are affected by the electron/hole modified valence electron count – the angle, dihedral and hydrogen bond terms still use the non-modified electron count.

Input/output files.
Geometry file. The explicit electrons and holes are added in the geo-file, as depicted in Example 9.1. Currently, the explicit electron is indicated by 'El' and the hole by 'Ho' – we acknowledge that the 'Ho' label can be confusing, as it might be mistaken for a Holbium atom (for which, currently, no ReaxFF parameters exist). The explicit charge of the electron or hole is set with MOLCHARGE keywords – note that it is required to also define the non-electron/hole atoms in MOLCHARGE keywords. We strongly recommend to put all the holes/electrons at the beginning or end of the geo-information, so that the rest of the non-electron/hole atoms can be captured in a single MOLCHARGE line and as such can freely exchange charges when molecular compositions change.

```
BIOGRF 200
DESCRP 08_w_el
REMARK w file created by Babel 1.6
MOLCHARGE   1   1 -1.00
MOLCHARGE   2  25  0.00
FORMAT ATOM   (a6,1x,i5,1x,a5,1x,a3,1x,a1,1x,a5,3f10.5,1x,a5,i3,i2,1x,f8.5)
HETATM     1 El                  39.99614  40.00419  40.00024    El  2 0  0.00000
HETATM     2 O                   39.11043  40.36875  37.76037    O   2 0  0.00000
HETATM     3 H                   38.54676  39.90381  38.44284    H   1 0  0.00000
HETATM     4 H                   39.98293  39.89636  37.88051    H   1 0  0.00000
HETATM     5 O                   41.51247  38.94035  38.32860    O   2 0  0.00000
HETATM     6 H                   41.73438  38.28709  37.66000    H   1 0  0.00000
HETATM     7 H                   41.05415  38.38735  39.04565    H   1 0  0.00000
HETATM     8 O                   39.75264  42.41341  39.51496    O   2 0  0.00000
HETATM     9 H                   39.27947  43.18404  39.19200    H   1 0  0.00000
HETATM    10 H                   39.51031  41.69227  38.83669    H   1 0  0.00000
HETATM    11 O                   37.70709  39.03190  39.85086    O   2 0  0.00000
HETATM    12 H                   37.96246  39.74552  40.53246    H   1 0  0.00000
HETATM    13 H                   36.75335  39.12421  39.78742    H   1 0  0.00000
HETATM    14 O                   42.28956  41.06839  40.27309    O   2 0  0.00000
HETATM    15 H                   42.17141  40.34847  39.61669    H   1 0  0.00000
HETATM    16 H                   41.51341  41.63285  40.06592    H   1 0  0.00000
HETATM    17 O                   40.21988  37.50300  40.35104    O   2 0  0.00000
HETATM    18 H                   40.52191  38.11673  41.05430    H   1 0  0.00000
HETATM    19 H                   39.33995  37.88268  40.14335    H   1 0  0.00000
HETATM    20 O                   38.50063  40.98387  41.64997    O   2 0  0.00000
HETATM    21 H                   38.87735  41.55897  40.92569    H   1 0  0.00000
HETATM    22 H                   39.31865  40.49040  41.94512    H   1 0  0.00000
HETATM    23 O                   40.94725  39.66444  42.28547    O   2 0  0.00000
HETATM    24 H                   41.45763  40.18520  41.57947    H   1 0  0.00000
HETATM    25 H                   41.33577  39.97570  43.10707    H   1 0  0.00000
FORMAT CONECT (a6,12i6)
UNIT ENERGY   kcal
ENERGY       -1923.014984
END
```

Example 9.1: Biograf input file for an 8-water, 1-explicit electron system. See Figure 9.3 for an image of this system.

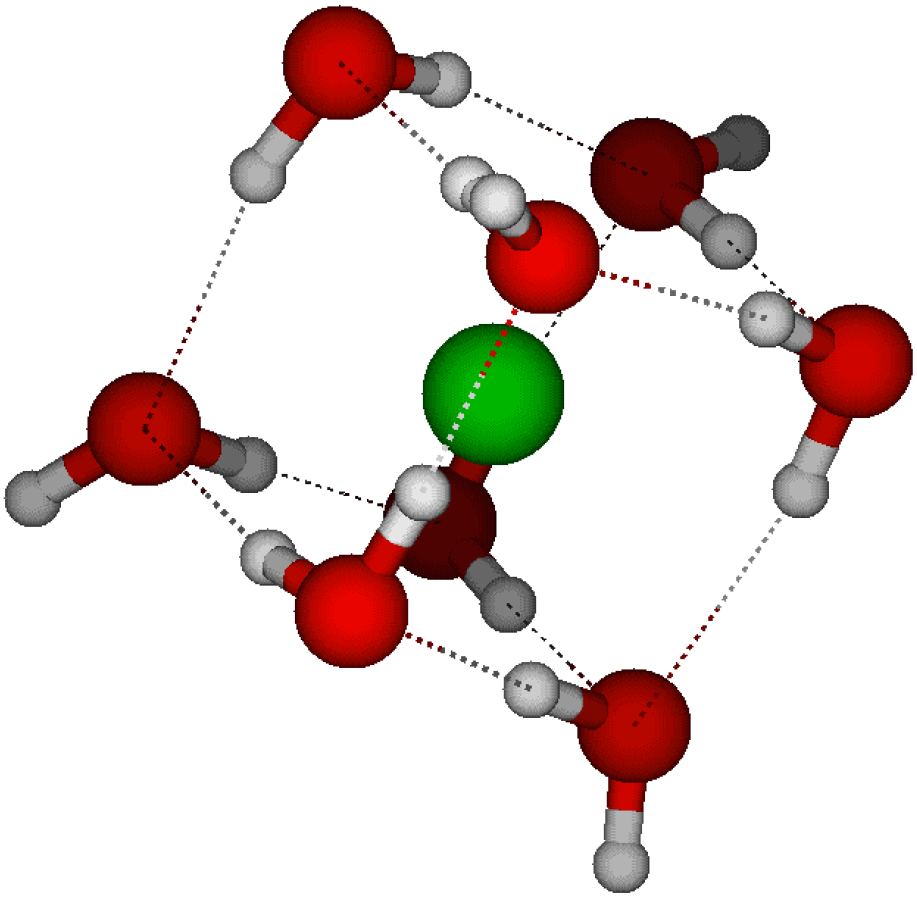


Figure 9.3. Structure of the 8-water, 1-electron system described in Example 9.1.

Note that for the visualization of the e-ReaxFF trajectories we typically change the ‘El’ atom type in the xmolout-trajectory to ‘Cl’ – since most visualization software can not handle the ‘El’ atom type.
When a second electron is added to the system from Example 9.1 and Figure 9.3 the input file will look like Example 9.2:

```
BIOGRF 200
DESCRP 8H2O_2el
MOLCHARGE   1   1 -1.00
MOLCHARGE   2   2 -1.00
MOLCHARGE   3  26  0.00
REMARK .bgf-file generated by xtob-script
HETATM     1 El                39.99614  40.00419  40.00024    Cl  0 0  0.00000
HETATM     2 El                38.04596  37.78193  36.99883    Cl  0 0  0.00000
HETATM     3 O                 39.11043  40.36875  37.76037    O   0 0  0.00000
HETATM     4 O                 38.50063  40.98387  41.64997    O   0 0  0.00000
HETATM     5 O                 39.75264  42.41341  39.51496    O   0 0  0.00000
HETATM     6 O                 37.70709  39.03190  39.85086    O   0 0  0.00000
HETATM     7 O                 41.51247  38.94035  38.32860    O   0 0  0.00000
HETATM     8 O                 40.94725  39.66444  42.28547    O   0 0  0.00000
HETATM     9 O                 40.21988  37.50300  40.35104    O   0 0  0.00000
HETATM    10 O                 42.28956  41.06839  40.27309    O   0 0  0.00000
HETATM    11 H                 38.54676  39.90381  38.44284    H   0 0  0.00000
HETATM    12 H                 39.98293  39.89636  37.88051    H   0 0  0.00000
HETATM    13 H                 41.73438  38.28709  37.66000    H   0 0  0.00000
HETATM    14 H                 41.05415  38.38735  39.04565    H   0 0  0.00000
HETATM    15 H                 39.27947  43.18404  39.19200    H   0 0  0.00000
HETATM    16 H                 39.51031  41.69227  38.83669    H   0 0  0.00000
HETATM    17 H                 37.96246  39.74552  40.53246    H   0 0  0.00000
HETATM    18 H                 36.75335  39.12421  39.78742    H   0 0  0.00000
HETATM    19 H                 42.17141  40.34847  39.61669    H   0 0  0.00000
HETATM    20 H                 41.51341  41.63285  40.06592    H   0 0  0.00000
HETATM    21 H                 40.52191  38.11673  41.05430    H   0 0  0.00000
HETATM    22 H                 39.33995  37.88268  40.14335    H   0 0  0.00000
HETATM    23 H                 38.87735  41.55897  40.92569    H   0 0  0.00000
HETATM    24 H                 39.31865  40.49040  41.94512    H   0 0  0.00000
HETATM    25 H                 41.45763  40.18520  41.57947    H   0 0  0.00000
HETATM    26 H                 41.33577  39.97570  43.10707    H   0 0  0.00000
END
```

Example 9.2: Biograf input file for an 8-water, 2-explicit electron system. See Figure 9.4 for an image of this system.

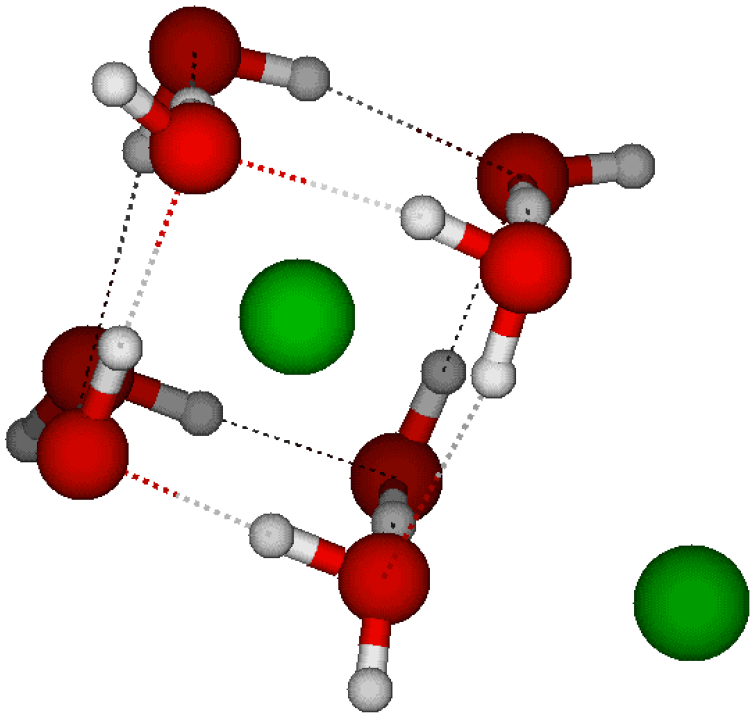


Figure 9.3. Structure of the 8-water, 1-electron system described in Example 9.2.

As indicated in Example 9.2, the electrons/holes need to be grouped at the top (or the bottom) of the geometry file, so that the non-electron/hole particles can be targeted with a single MOLCHARGE line (MOLCHARGE 3 26 0.00, Example 9.2) enabling normal charge transfer during reactions between the regular atoms.

Also, while electrons/holes can be put very close to the nuclei, they should not be put *exactly* on top of the atom – since this will results in a NaN results. However, even with a very small displacement from the atom (e.g. 0.001 Angstrom) the atom/electron/hole configuration will provide a meaningful result. As such – for building initial structures, when it is desired to put the electron/hole close to a particular atom it is advisable to copy the target atom location, rename the atom label to 'El' or 'Ho' for electron/holes, respectively, and then provide a small translation.

Control-file. Two control-keywords are of particular relevance for the e-ReaxFF method. First, and most important, the icharg-keyword (see Table 2.10) should be set to 7 to invoke the ACKS2 charge calculation method – which is essential to e-ReaxFF. Second, a new control-keyword, ielec, is available. Setting ielec=1 will create an additional trajectory output file (xmolout_el) in which the electrons or holes standard atom labels (El/Ho) are replaced with X – this enables direct visualization with Molden and VMD.

Force field file. A number of e-ReaxFF specific parameters are added to the ReaxFF force field file. Below is a list of these parameters and their general purpose. For a further explanation of the ACKS2 parameters, please see (Verstraelen et al., 2013)

General parameters (section 1 of the force field, see Example 2.8)
Parameter 27: Electron/hole repulsion (under development)
Parameter 35: ACSK2 softness parameter
Parameter 37: p_{val}, equation 9.3

Atom parameters (section 2 of the force field)
Parameter 13: p_{xel2}, equation 9.5
Parameter 23: ACKS2 softcut
Parameter 24: alfa, equation 9.2
Parameter 27: beta, equation 9.3

Bond parameters (section 3 of the force field)
Parameter 16: p_{xel1}, equation 9.5

10. Literature

Abolfath, R.M., van Duin, A.C.T., Biswas, P. and Brabec, T. (2011) Reactive Molecular Dynamics study on the first steps of DNA-damage by free hydroxyl radicals. Journal of Physical Chemistry A 115, 11045-11049.

Aktulga, H.M., Grama, A.Y., Pandit, S.A. and van Duin, A.C.T. (2012) REACTIVE MOLECULAR DYNAMICS: NUMERICAL METHODS AND ALGORITHMIC TECHNIQUES. SIAM Journal on Scientific Computing 34, C1-C23.

Aryanpour, M., van Duin, A.C.T. and Kubicki, J.D. (2010) Development of a Reactive Force Field for Iron-Oxyhydroxide Systems. Journal of Physical Chemistry A 114, 6298-6307.

Assowe, O., Politano, O., Vignal, V., Arnoux, P., Diawara, B., Verners, O. and van Duin, A.C.T. (2012) Reactive molecular dynamics simulation of the initial oxidation stages of Ni (111) in pure water: Effect of an applied electric field. Journal of Physical Chemistry 116, 11796-11805.

Bharati, A.K., Kamat, A.M. and van Duin, A.C.T. (2012) Study of effect of strain on the thermal conductivity of Zinc Oxide using the ReaxFF reactive force field. Computational and Theoretical Chemistry 987, 71-76.

Brenner, D.W. (1990) Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films. Physical Review B 42, 9458-9471.

Buehler, M.J., Tang, H., van Duin, A.C.T. and Goddard, W.A. (2007) Threshold crack speed controls dynamical fracture of silicon single crystals. Physical Review Letters 99, 165502/165501-165502/165504.

Buehler, M.J., van Duin, A.C.T. and Goddard, W.A. (2006) Multiparadigm modeling of dynamical crack propagation in silicon using a reactive force field. Physical Review Letters 96, 095505.

Castro-Marcano, F., Kamat, A.M., Russo, M.F., van Duin, A.C.T. and Mathews, J.P. (2012) Combustion of an Illinois No. 6 Coal Char Simulated Using an Atomistic Char Representation and the ReaxFF Reactive Force Field. Combustion and Flame 159, 1272-1285.

Chen, H.-P., Kalia, R.K., Kaxiras, E., Lu, G., Nakano, A., Nomura, K.-i., van Duin, A.C.T., Vashishta, P. and Yuan, Z. (2010) Embrittlement of Metal by Solute Segregation-Induced Amorphization. Physical Review Letters 104, 155502/155501-155502/155504.
Chenoweth, K., Cheung, S., van Duin, A.C.T., Goddard, W.A. and Kober, E.M. (2005) Simulations on the thermal decomposition of a poly(dimethylsiloxane) polymer using the ReaxFF reactive force field. Journal of the American Chemical Society 127, 7192-7202.
Chenoweth, K., van Duin, A.C.T., Dasgupta, S. and Goddard, W.A. (2009a) Initiation Mechanisms and Kinetics of Pyrolysis and Combustion of JP-10 Hydrocarbon Jet Fuel. Journal of Physical Chemistry A 113, 1740-1746.
Chenoweth, K., van Duin, A.C.T. and Goddard, W.A. (2008a) ReaxFF reactive force field for molecular dynamics simulations of hydrocarbon oxidation. Journal of Physical Chemistry A 112, 1040-1053.
Chenoweth, K., van Duin, A.C.T. and Goddard, W.A. (2009b) The ReaxFF Monte Carlo Reactive Dynamics Method for Predicting Atomistic Structures of Disordered Ceramics: Application to the Mo3VOx Catalyst. Angewandte Chemie-International Edition 48, 7630-7634.
Chenoweth, K., van Duin, A.C.T., Persson, P., Cheng, M.J., Oxgaard, J. and Goddard, W.A. (2008b) Development and application of a ReaxFF reactive force field for oxidative dehydrogenation on vanadium oxide catalysts. Journal of Physical Chemistry C 112, 14645-14654.
Fogarty, J.C., Aktulga, H.M., Grama, A.Y., van Duin, A.C.T. and Pandit, S.A. (2010) A reactive molecular dynamics simulation of the silica-water interface. Journal of Chemical Physics 132, 174704/174701-174704/174710.
Gale, J.D., Ratieri, P. and van Duin, A.C.T. (2011) A Reactive Force Field for Aqueous-Calcium Carbonate Systems. Phys. Chem. Chem. Phys. 13, 16666-16679.
Ganesh, P., Kent, P.R.C. and V., M. (2011) Formation, Characterization and Dynamics of Onion-like Carbon Structures from Nanodiamonds Using Reactive Force-Fields for Electrical Energy Storage. Journal of Applied Physics 110, 073506.
Goddard, W.A., III, Mueller, J.E., Chenoweth, K. and van Duin, A.C.T. (2010) ReaxFF Monte Carlo reactive dynamics: Application to resolving the partial occupations of the M1 phase of the MoVNbTeO catalyst. Catal. Today 157, 71-76.
Goken, E., Joshi, K., Russo, M., van Duin, A.C.T. and Castleman, A.W. (2011) The effect of formic acid addition on water cluster stability and structure. Journal of Physical Chemistry A 115, 4657-4664.
Huang, L., Bandosz, T., Joshi, K., van Duin, A.C.T. and Gubbins, K.E. (2013) Reactive adsorption of ammonia and ammonia/water on CuBTC metal-organic framework: a ReaxFF molecular dynamics simulation. Journal of Chemical Physics 138, 034102.
Islam, M., Kolesov, G., Verstraelen, T., Kaxiras, E. and van Duin, A.C.T. (2016) eReaxFF: A Pseudo-Classical Treatment of Explicit Electrons in ReaxFF Reactive Force Field Simulations. Journal of Chemical Theory and Computation 12, 3463-3472.
Islam, M. and van Duin, A.C.T. (2016) Reductive Decomposition Reactions of Ethylene Carbonate via Explicit Electrons: An eReaxFF Molecular Dynamics Study. Journal of Physical Chemistry 120, 27128-27134.
Jeon, B., Sankaranarayanan, S., van Duin, A.C.T. and Ramanathan, S. (2011) Atomistic insights into aqueous corrosion of copper. Journal of Chemical Physics 134, 234706-234701-234706-234709.

Jeon, B., Sankaranarayanan, S., van Duin, A.C.T. and Ramanathan, S. (2012) Reactive molecular dynamics study of chloride ion interaction with copper oxide surfaces in aqueous media. ACS Applied Materials and Interfaces 4, 1225-1232.
Jiang, D.E., van Duin, A.C.T., Goddard, W.A. and Dai, S. (2009) Simulating the Initial Stage of Phenolic Resin Carbonization via the ReaxFF Reactive Force Field. Journal of Physical Chemistry A 113, 6891-6894.
Joshi, K. and van Duin, A.C.T. (2013) Molecular dynamics study on the influence of additives on the high temperature structural and acidic properties of ZSM-5 zeolite. Energy and Fuels 27, 4481-4488.
Khalilov, U., Pourtois, G., Bogaerts, A., van Duin, A.C.T. and Neyts, E.C. (2013a) A new mechanism for oxidation of native silicon oxide. Journal of Physical Chemistry C 117, 9819-9825.
Khalilov, U., Pourtois, G., Bogaerts, A., van Duin, A.C.T. and Neyts, E.C. (2013b) Reactive Molecular Dynamics Simulations on SiO2-Coated Ultra-Small Si-Nanowires. Nanoscale 5, 719-725.
Khalilov, U., Pourtois, G., van Duin, A. and Neyts, E.C. (2012a) On the c-Si/a-SiO2Interface in Hyperthermal Si Oxidation at Room Temperature. Journal of Physical Chemistry C 116, 21856-21863.
Khalilov, U., Pourtois, G., van Duin, A.C.T. and Neyts, E.C. (2012b) Self-limiting Oxidation in Small Diameter Si Nanowires. Chemistry of Materials 24, 2141-2147.
Kim, S.-Y., Kumar, N., Persson, P., Sofo, J., van Duin, A.C.T. and Kubicki, J.D. (2012a) Development of a ReaxFF reactive force field for Titanium dioxide/water systems. Langmuir submitted.
Kim, S.-Y., van Duin, A.C.T. and Kubicki, J.D. (2012b) 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 Accepted for publication.
Manzano, H., Pellenq, R., Ulm, F.-J., Buehler, M.J. and van Duin, A.C.T. (2012a) Hydration of calcium oxide predicted by reactive force field molecular dynamics. Langmuir 28, 4187-4197.
Manzano, H., Ulm, F.-J., van Duin, A., Pellenq, R., Marinelli, F. and Moeni, S. (2012b) Water polarization and dissociation in confined nanopores: mechanism, dipole distribution, and impact on the substrate properties. Journal of the American Chemical Society 134, 2208-2215.
Monti, S., Corozzi, A., Fristrup, P., Joshi, K., Shin, Y.K., Oelschlager, P., van Duin, A.C.T. and Barone, V. (2013) Exploring the conformational and reactive dynamics of biomolecules in solution using an extended version of the glycine reactive force field. Phys. Chem. Chem. Phys. 15, 15062-15077.
Monti, S., van Duin, A.C.T., Kim, S.-Y. and Barone, V. (2012) Exploration of the Conformational and Reactive Dynamics of Glycine During and After its Adsorption onto Titania: Computational Investigations in the Gas Phase and in Solution. Journal of Physical Chemistry C 116, 5141-5150.
Mortier, W.J., Ghosh, S.K. and Shankar, S. (1986) Electronegativity Equalization Method for the Calculation of Atomic Charges in Molecules. Journal of the American Chemical Society 108, 4315-4320.
Nakano, A., Kalia, R.K., Nomura, K., Sharma, A., Vashishta, P., Shimojo, F., van Duin, A.C.T., Goddard, W.A., Biswas, R. and Srivastava, D. (2007) A divide-and-

conquer/cellular-decomposition framework for million-to-billion atom simulations of chemical reactions. Computational Materials Science 38, 642-652.
Nakano, A., Kalia, R.K., Nomura, K.I., Sharma, A., Vashishta, P., Shimojo, F., Van Duin, A.C.T., Goddard, W.A., Biswas, R., Srivastava, D. and Yang, L.H. (2008) De novo ultrascale atomistic simulations on high-end parallel supercomputers. International Journal of High Performance Computing Applications 22, 113-128.
Neyts, E.C., Khalilov, U., Portois, G. and van Duin, A.C.T. (2011) Hyperthermal Oxygen Interacting with Silicon Surfaces: Adsorption, Implantation and Damage Creation. Journal of Physical Chemistry C 115, 4818-4823.
Neyts, E.C., van Duin, A.C.T. and Bogaerts, A. (2012) Insights in the Plasma Assisted Growth of Carbon Nanotubes through Atomic Scale Simulations: Effect of Electric Field. Journal of the American Chemical Society 134, 1256-1260.
Nomura, K., Kalia, R.K., Nakano, A., Vashishta, P. and van Duin, A.C.T. (2012) Mechanochemistry of Nanobubble Collapse near Silica in Water. Applied Physics Letters 101, 073108/073101-073108/073104.
Nomura, K.I., Kalia, R.K., Nakano, A., Vashishta, P., van Duin, A.C.T. and Goddard, W.A. (2007) Dynamic transition in the structure of an energetic crystal during chemical reactions at shock front prior to detonation. Physical Review Letters 99, 148303.
Pitman, M.C. and van Duin, A.C.T. (2012) Dynamics of Confined Reactive Water in Smectic Clay–Zeolite Composites. Journal of the American Chemical Society 134, 3042-3053.
Rahaman, O., van Duin, A.C.T., Bryantsev, V.S., Mueller, J.E., Solares, S.D., Goddard, W.A., III and Doren, D.J. (2010) Development of a ReaxFF Reactive Force Field for Aqueous Chloride and Copper Chloride. J. Phys. Chem. A 114, 3556-3568.
Rahaman, O., van Duin, A.C.T., Goddard, W.A., III and Doren, D.J. (2011) Development of a ReaxFF reactive force field for glycine and application to solvent effect and tautomerization. Journal of Physical Chemistry B 115, 249-261.
Raju, M., Fichthorn, K., Kim, S.-Y. and van Duin, A.C.T. (2013) ReaxFF Reactive Force Field Study of the Dissociation of Water on Titania Surfaces. Journal of Physical Chemistry C 117, 10558-10572.
Raymand, D., van Duin, A.C.T., Goddard, W.A., Hermansson, K. and Spangberg, D. (2011) Hydroxylation Structure and Proton Transfer Reactivity at the Zinc Oxide-Water Interface. Journal of Physical Chemistry A 115, 8573-8579.
Raymand, D., van Duin, A.C.T., Spangberg, D., Goddard, W.A. and Hermansson, K. (2010) Water adsorption on stepped ZnO surfaces from MD simulation. Surface Science 604, 741-752.
Russo, M., Li, R., Mench, M. and van Duin, A.C.T. (2011) Molecular Dynamic Simulation of Aluminum-Water Reactions Using the ReaxFF Reactive Force Field. International Journal of Hydrogen Energy 36, 5828-5835.
Salmon, E., van Duin, A.C.T., Lorant, F., Marquaire, P.M. and Goddard, W.A. (2009a) Early maturation processes in coal. Part 2: Reactive dynamics simulations using the ReaxFF reactive force field on Morwell Brown coal structures. Organic Geochemistry 40, 1195-1209.
Salmon, E., van Duin, A.C.T., Lorant, F., Marquaire, P.M. and Goddard, W.A. (2009b) Thermal decomposition process in algaenan of Botryococcus braunii race L. Part 2:

Molecular dynamics simulations using the ReaxFF reactive force field. Organic Geochemistry 40, 416-427.
Senftle, T., Hong, S., Islam, M., Kylasa, S.B., Zheng, Y., Shin, Y.K., Junkermeier, C., Engel-Herbert, R., Janik, M., Aktulga, H.M., Verstraelen, T., Grama, A.Y. and van Duin, A.C.T. (2016) The ReaxFF Reactive Force-field: Development, Applications, and Future Directions. Nature Computational Materials 2, 15011.
Shan, T.-R., Wixom, R.R., Mattson, A.E. and Thompson, A.P. (2013) Atomistic Simulation of Orientation Dependence in Shock-Induced Initiation of Pentaerythritol Tetranitrate. Journa of Physical Chemistry B 117, 928-936.
Srinivasan, S.G. and van Duin, A.C.T. (2011) A molecular dynamics based study of the collisions of hyperthermal atomic oxygen with graphene using the ReaxFF reactive force field. Journal of Physical Chemistry A 115, 13269-13280.
van Duin, A.C.T., Bryantsev, V.S., Diallo, M.S., Goddard, W.A., Rahaman, O., Doren, D.J., Raymand, D. and Hermansson, K. (2010) 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 114, 9507-9514.
van Duin, A.C.T., Dasgupta, S., Lorant, F. and Goddard, W.A. (2001) ReaxFF: A reactive force field for hydrocarbons. Journal of Physical Chemistry A 105, 9396-9409.
van Duin, A.C.T., Zou, C., Joshi, K., Bryantsev, V.S. and Goddard, W.A. (2012) A ReaxFF reactive force field for proton transfer reactions in bulk water and its applications to heterogeneous catalysis.
Vashishta, P., Kalia, R.J. and Nakano, A. (2006) Multimillion Atom Simulations of Dynamics of Oxidation of an Aluminum Nanoparticle and Nanoindentation on Ceramics. Journal of Physical Chemistry B 110, 3727-3733.
Vedadi, M., Choubey, A., Nomura, K., Kalia, R.K., Nakano, A., Vashishta, P. and van Duin, A.C.T. (2010) Structure and Dynamics of Shock-Induced Nanobubble Collapse in Water. Physical Review Letters 105, 014503/014501-014503/014504.
Verstraelen, T., Ayers, P.W., Van Speybroeck, V. and Waroquier, M. (2013) ACKS2: atom-condensed Kohn-Sham DFT approximated to second order. Journal of Chemical Physics 138, 074108.
Yusupov, M., Khalilov, U., Neyts, E.C., Snoeckx, R., van Duin, A.C.T. and Bogaerts, A. (2012) Atomic scale simulations of plasma species interacting with bacteria cell walls. New Journal of Physics 14, 093043.
Zou, C., van Duin, A.C.T. and Sorescu, D. (2012) Theoretical investigation of hydrogen adsorption and dissociation on iron and iron carbide surfaces using the ReaxFF reactive force field method. Topics in Catalysis 55, 391-401.
Zybin, S.V., Goddard, W.A., III, Xu, P., van Duin, A.C.T. and Thompson, A.P. (2010) Physical mechanism of anisotropic sensitivity in pentaerythritol tetranitrate from compressive-shear reaction dynamics simulations. Applied Physics Letters 96, 081918/081911-081918/081913.