



Scientific Computing & Modelling

UFF Manual

**ADF Program System
Release 2013**

Scientific Computing & Modelling NV
Vrije Universiteit, Theoretical Chemistry
De Boelelaan 1083; 1081 HV Amsterdam; The Netherlands
WWW: www.scm.com
E-mail: support@scm.com

Copyright © 1993-2013: SCM / Vrije Universiteit, Theoretical Chemistry, Amsterdam, The Netherlands
All rights reserved

Table of Contents

UFF Manual	1
Table of Contents	2
Introduction.....	3
Release 2013	3
UFF-GUI	3
Input and Examples.....	4
UFF input example	4
Periodic UFF	7
Parameter files.....	8
General parameters file	8
Elements file	10
MM Atom Types file.....	11
Parameter file keywords (summary).....	11
Atom typing behaviour	12
References	13

Introduction

UFF (Universal Forcefield) is a full periodic table forcefield that can be used to calculate single point energies, do geometry optimizations, calculate frequencies and to do transition state searches. It is also the default pre-optimizer in the Graphical User interface (ADF GUI) to clean up manually drawn structures. UFF can also do calculations with chain, slab or bulk periodicity. For details on the inner workings of the forcefield, see the 1992 paper by Rappe et al. [1].

Release 2013

The 2013 release of our UFF program features the following major updates:

- The bonds drawn in the GUI are now taken as input, making UFF a more robust pre-optimizer
- All UFF parameters and atom types can now be edited, making it possible to tune UFF or extend the set of atom types

UFF-GUI

Note that the graphical user interface UFF-GUI enables all users to set up UFF calculations with a few mouse clicks. When using UFF, also for pre-optimizing, it is important to remember that any forcefield method relies on a set of parameters. This means that even though UFF supports all elements up to Z=103, it might not generate the desired structures for uncommon oxidation states in metallic structures. If this is the case, you could add new parameters to UFF or attach dummy hydrogen atoms to the metal atom.

Input and Examples

The input style for the UFF program is similar to the input structure of the other parts in the ADF package. Some helpful examples of UFF calculations can be found in the \$ADFHOME/examples/uff directory. The input can also be studied by setting up a job with the UFF tab in the GUI, followed by clicking on "Details" -> "Run Script". The UFF input can have multiple forms in the ADF package. Depending on what information is given, UFF fills in the rest. The main options are:

- Specify Elements, coordinates, MM Atom Types, Charges, and bonds
- Specify Elements, coordinates, and bonds
- Specify Elements and coordinates

UFF input example

The input should look something like this:

```
$ADFBIN/uff << eor

Units
  length Angstrom
End

System
  Atoms
    C  1.36012328  -0.14520095   0.60144543   C_3   0.000000
    C  0.00000000   0.00000000   0.00000000   C_2   0.000000
    H  2.09833847  -0.46327872  -0.16560721   H_    0.000000
    H  1.32657807  -0.90546800   1.40917410   H_    0.000000
    H  1.67935140   0.82750664   1.02977296   H_    0.000000
    H  -0.83486863   0.30434056   0.62258487   H_    0.000000
    O  -0.18030374  -0.22462371  -1.18585739   O_2   0.000000
  End

  Bonds
    1    2    1.000000000000
    1    3    1.000000000000
    1    5    1.000000000000
    1    4    1.000000000000
    2    7    2.000000000000
    2    6    1.000000000000
  End
End

Task
  RunType GO
END

Geometry
  Optim Cartesian
End

eor
```

The format in the Atoms section is: Element, x-coord, y-coord, z-coord, MM Atom Type, Charge.

The format in the bonds section is: atom A, atom B, bond order.

If we leave out the MM atom types and charges, UFF will determine the MM atom types automatically from the bond information. Except for leaving out MM atom types and charges, the second input format is similar to the first input format:

```
$ADFBIN/uff << eor

Units
  length Angstrom
End

System
  Atoms
    C  1.36012328  -0.14520095   0.60144543
    C  0.00000000   0.00000000   0.00000000
    H  2.09833847  -0.46327872  -0.16560721
    H  1.32657807  -0.90546800   1.40917410
    H  1.67935140   0.82750664   1.02977296
    H -0.83486863   0.30434056   0.62258487
    O -0.18030374  -0.22462371  -1.18585739
  End

  Bonds
    1    2    1.000000000000
    1    3    1.000000000000
    1    5    1.000000000000
    1    4    1.000000000000
    2    7    2.000000000000
    2    6    1.000000000000
  End
End

Task
  RunType GO
END

Geometry
  Optim Cartesian
End

eor
```

The third input format is similar to the second, but without a Bonds section in System:

```
$ADFBIN/uff << eor

Units
  length Angstrom
End

System
  Atoms
    C  1.36012328  -0.14520095   0.60144543
    C  0.00000000   0.00000000   0.00000000
    H  2.09833847  -0.46327872  -0.16560721
    H  1.32657807  -0.90546800   1.40917410
    H  1.67935140   0.82750664   1.02977296
```

```
H -0.83486863 0.30434056 0.62258487
O -0.18030374 -0.22462371 -1.18585739
End
End
Task
  RunType GO
END
Geometry
  Optim Cartesian
End
eor
```

The GUI generates inputs of the second or third type, depending on the "Use existing bonds" setting in the UFF main tab.

Note that to specify the MM Atom Types, the charges also need to be set. UFF will not assign charges to atoms on its own, so fixing the charges to 0.000 will not change behaviour.

Periodic UFF

UFF can be used to optimize the geometries of periodic systems. However, some limitations apply:

- UFF cannot handle systems with unit cells that are too small, each lattice direction should be at least 6 Angstrom. If the unit cells are too small, UFF will confuse bonds through periodic boundaries with internal bonds. If you need to use UFF for a very small system, You can create a supercell to increase the sizes of the unit cell.
- UFF is currently unable to optimize lattice vectors. The size and shape of the unit cell will thus remain fixed during the optimization.
- When building a periodic system, one should also specify the bonds that pass through a periodic boundary. These have the same format as the other UFF bonds and there is no distinction between normal bonds and bonds passing through periodic boundaries in the UFF input file. For an example on how this works: open ADF GUI, go to the UFF interface and use the magnifying glass to find 'zeolite ABW'. The UFF input file can now be seen by going to details > Run Script.

If your system shows strange geometries when optimizing with UFF, it is probably one of the above problems. To fix problems with bonds, you can try to disable `Use existing bonds` on the main UFF tab. This will enable automatic bond detection by UFF, which might work if your system has a reasonable geometry. If your unit cell is too small, use the structure tool (the snowflake) to generate a supercell of 2x2x2, and try again.

Parameter files

The UFF implementation in ADF uses three parameter files:

- `general_db`: Parameters per MM atom type
- `elements_db`: Defines the elements and the possible valences and bond orders
- `mmatomtypes_db`: Defines the MM Atom Types known to UFF and their matching settings

These parameter files are kept in the directory `$ADFHOME/atomicdata/UFF` (or `$ADFRESOURCES/UFF` if the installation is modified for specific systems).

General parameters file

The `general_db` file (`$ADFHOME/atomicdata/UFF/general_db`) contains all the parameters used to calculate the forces and energies. The format is:

`MMAtomType`, `ri`, `phi`, `xi`, `di`, `psi`, `zmm`, `vsp3`, `vsp2`, `chi`, `nc`.

The items in the list are:

- `MMAtomType`: name, max 5 characters
- `ri`: Valence Bond [Å]
- `phi`: Valence Angle [Degree]
- `xi`: Nonbond Distance [Å]
- `di`: Nonbond Energy [kcal/mol]
- `psi`: Nonbond scale [Number]
- `zmm`: Effective Charge [Charge]
- `vsp3`: sp3 Torsional Barrier [kcal/mol]
- `vsp2`: sp2 Torsional Barrier [kcal/mol]
- `chi`: Electronegativity
- `nc`: Number of directly attached atoms, aka coordination number. This is required for counting the number of possible dihedrals, and is defined only for the sp2 and sp3 centres (types 2, R, and 3)

The current set of parameters comes from the `deMonNano` program, and is a combination of published parameters and fitted data to fill in the gaps. The `deMonNano` documentation says the following about the parameters:

Implementation of the Universal Force Field (UFF) in `deMonNano`

As far as possible, UFF molecular mechanics forcefield in `deMon` follows the published forcefield definition in [1]. In several cases, the definitions and expressions in [1] are not consistent with the published applications of the forcefield [1,5,6]. In those cases, an attempt was made to correct the errors and omissions, using information from [2-4].

The following changes were made, compared to the published UFF forcefield description (all equation and page numbers refer to [1]).

1. Sign error in Eq. 2 (equilibrium bond length) was corrected
- electronegativity correction must be negative!
2. Equilibrium valence angle for `O_3_z` was corrected from 146.0

degree to 145.45 degree.

3. Bending periodicity (Eq. 10) for linear coordination was corrected from 1 to 2.
4. Sign errors were corrected in eqs. 13 and and unnumbered equation for the beta parameter (between eqs. 13 and 14).
5. The reference value of the UFF amide force constant, of 105.5 kcal/mol/rad**2 (p. 10028) is wrong. The results are consistent with the force constant of 211.0 kcal/mol/rad**2.
6. Equilibrium torsional angle for a bond between a group-6A atom (oxygen ...) and an sp2 atom (90 degree) is wrong (p. 10028). It should be 0 degree.
7. The conditional for the special-case sp2-sp3 torsion (p. 10029) is wrong, and should be inverted - see [4].
8. The overall shape of the UFF torsional potential degenerates to a Heavyside function when one of the bond angles becomes linear, leading to failures in geometry optimization and force constant evaluation. The UFF torsional term was augmented with a smooth masking function, to avoid this. See "uff_4centre.f90" for details.
9. UFF inversion potential is not defined in [1] for group 5A elements (from phosphorus down). Taking the equilibrium inversion coordinate of 87 degree, and the suitable expressions for the cosine weights (see uff_get_inversion_shape in "uff_database.f90") appears to reproduce published UFF structures and energetics.

The following atom types have been fully tested, and are believed to reproduce published UFF forcefield results exactly. The examples refer to the \$deMon/examples/test.mm directory.

Atom type	Example	Description
-----	-----	-----
Al3	alme3tma	Trivalent aluminum
As3+3	asf3	Trivalent arsenic
B_2	bcl3	Planar (sp2) boron
B_3	b2h5nme2	Tetrahedral (sp2) boron, including charge transfer adducts and borohydrates
Br	bbr3	Univalent bromine
C_1	c2h2, co	Linear (sp) carbon
C_2	acetone	Planar tricoordinated (sp2) carbon
C_3	c2h6	Tetrahedral (sp3) carbon
C_R	c4h6	Resonant, variable bond order (sp2) carbon.
Cl	socl2	Univalent chlorine
F_	sof2ncl	Univalent fluorine
Ge3	geh3ogeh3	Tetrahedral (sp3) germanium
H_	h2o	Normal, non-bridging hydrogen
H_b	b2h5nme2	Bridging hydrogen, for use in boranes (NOT SUITABLE FOR H-BONDS!)
I_	bi3	Univalent iodine
N_1	ch3cn	Monocoordinated (sp) nitrogen, triple bond

```

N_2  ch3n2ch3  Dicoordinated (sp2) nitrogen, single-double bond
N_3  ch3nh2    Amine (sp3) nitrogen, three single bonds
N_3+4 b2h5nme2 Charged amine (sp3) nitrogen, four single bonds
      (THIS IS NOT A STANDARD UFF TYPE!)
N_R   c5h5n    Resonant planar (sp2) nitrogen, for use in
      aromatics and amides. For amides, use 1.41 bond
      order!
O_1   co      Special "co" type, one triple bond.
O_2   acetone One-coordinated (sp2) oxygen, one double bond.
O_3   h2o     Two-coordinated (sp3) oxygen, two single bonds
O_3_z sih3osih3 Special two-coordinated oxygen, for use in
      Si-O bonds
O_R   c4h4o   Resonant planar (sp2) oxygen, also for use in
      nitro groups and such.
P_3+3 ph3     Pyramidal (sp3) phosphorus, three single bonds
P_3+5 p4o7    Tetrahedral hypervalent phosphorus
P_3+q bh3ph3  Dative tetrahedral (sp3) phosphorus, watch for
      the bond order!
S_3+2 ch3sch3 Bent two-coordinated sulfur (sp3), two single bonds
S_3+4 socl2   Pyramidal three-coordinated hypervalent sulfur
S_3+6 so2cl2  Tetrahedral four-coordinated hypervalent sulfur
Se3+2 h2se    Bent two-coordinated (sp3) selenium
Si3   si4o4h8 Tetrahedral silicon

```

Additionally, parameter sets for the following atom types are believed to be complete, and may be expected to produce results identical to the published UFF data: Li, Na, K, Rb, Cs (Note that UFF does not specify atomic charges - it is your responsibility to assign those, if charges are needed!)

For the remaining atom types, UFF definition [1] relies on an unpublished set of electronegativities [2]. In deMon, these values were replaced by Pauling electronegativities, scaled to fit published UFF electronegativities. This can be expected to produce small deviations in bond lengths and bond angles, compared to published UFF results.

If you wish to use other parameters, you should copy the `general_db` file, and rename it. This new file can also be placed outside of `$ADFHOME/atomicdata/UFF`. To use this new parameter file in UFF, specify the following keyword in your input:

```

| UFF
|   database /path/to/the/new/file/mygeneral_db
| End

```

Elements file

The `elements_db` file holds all the elements known to UFF. Keep in mind that these are not the `MMAtomTypes`, but pure chemical elements. The table contains for every element: atomic number, symbol, minimal valence number, maximum valence number, minimal bond order, maximal bond order. The data in the `elements_db` is mainly used for cleaning up the Pauling bond orders guessed by UFF, and will probably not need to be modified. However, if you want to, you can change the elements file used by UFF in your input:

```

| UFF
|   elementsfile /path/to/the/new/file/myelements_db
| End

```

MM Atom Types file

The `mmatomtypes_db` file contains the matching rules for assigning MM atom types to chemical elements, based on their valence number, and the number of neighbor (bonded) atoms. The current implementation of UFF is limited to 6 MM atom types per element. The table contains for every MM atom type: Number of the element it belongs to, the *i*-th type of this element, the valence number corresponding to this MM atom type, number of neighbors this MM atom type has, the name of this MM atom type. The naming convention follows the original UFF paper [1]:

"A five-character mnemonic label is used to describe the atom types. The first two characters correspond to the chemical symbol; an underscore appears in the second column if the symbol has one letter (e.g., N_ is nitrogen, Rh is rhodium). The third column describes the hybridization or geometry: 1 = linear, 2 = trigonal, R = resonant, 3 = tetrahedral, 4 = square planar, 5 = trigonal bipyramidal, 6 = octahedral. Thus N_3 is tetrahedral nitrogen, while Rh6 is octahedral rhodium. The fourth and fifth columns are used as indicators of alternate parameters such as formal oxidation state: Rh6+3 indicates an octahedral rhodium formally in the +3 oxidation state, e.g., Rh(NH3)_6^3+. H_b indicates a bridging hydrogen as in B2H6. O_3_z is an oxygen suited for framework oxygens of a zeolite lattice. P_3_q is a tetrahedral four-coordinate phosphorus used to describe organo-metallic coordinated phosphines."

You can copy the `mmatomtypes_db` and change it if you need to modify the atom typing behaviour of UFF. To use your new file in UFF, specify the following in your input:

```
UFF
  atomtypesfile /path/to/the/new/file/mymmatomtypes_db
End
```

Parameter file keywords (summary)

As detailed above, the UFF program can use modified parameters. To use these modified files, the input needs one or more of the following keywords:

```
UFF
  database      /path/to/the/new/file/mygeneral_db
  atomtypesfile /path/to/the/new/file/mymmatomtypes_db
  elementsfile  /path/to/the/new/file/myelements_db
End
```

Finding good parameters for these files can be a challenging task, and any results with modified parameters should be checked very carefully. SCM has no experience with this, and the parameters supplied for UFF have not been generated by us. Feel free to test new parameters, and feel free to let us know if you have a good working set for a specialized situation.

Atom typing behaviour

Atom typing is the process of matching MM atom types to elements. For example, a Carbon atom becomes a C_1, C_2, C_3 or C_R, depending on the number (and type) of bonds it has to neighboring atoms. UFF is capable of finding a matching MM atom type on its own, but might not always succeed in doing so. When doing calculations with UFF, it is important to check the beginning of the output file, as the program will print the detected MM atom types there. You can also take matters into your own hands, and tell UFF what MM atom types you want it to use (see the section on Input and examples).

The atom typing in UFF is mostly controlled by the elements and mmatomtypes parameter files, however, some part of it is hidden in the code itself and is (at the moment) not accessible to users. This is done by UFF to differentiate between:

- Carbon: double-bonded sp² carbon (C_2) vs aromatic carbon (C_R), if any of the orders are close to 2, it's C_2.
- Oxygen: having a bond to silicon gives O_3_z; otherwise, having bond order of 1.44 or higher gives O_R, provided that the partner is not a hydrogen. Otherwise, it's just an O_3.
- Sulfur: having a bond order greater than 1.3 gives aromatic S_R.

If you have source-code access, this behaviour can be modified in the MMStructureInfo module.

References

1. A.K. Rappe, C.J. Casewit, K.S. Colwell, W.A. Goddard III, and W. M. Skiff, "*UFF, a Full Periodic Table Force Field for Molecular Dynamics Simulations*", *Journal of the American Chemical Society* **114**, 10024-10035 (1992).
2. A.K. Rappe, W.A. Goddard III, "*Charge Equilibration for Molecular Dynamics Simulations*", *The Journal of Physical Chemistry* **95**, 3358-3363 (1991).
3. M. O'Keeffe and N.E. Brese, "*Atom sizes and bond lengths in Molecules and Crystals*", *Journal of the American Chemical Society* **113**, 3226-3229 (1991).
4. S.L. Mayo, B.D. Olafson, W.A. Goddard III, "*DREIDING: A Generic Force Field for Molecular Simulations*", *The Journal of Physical Chemistry* **94**, 8897-8909 (1990).
5. C.J. Casewit, K.S. Colwell, A.K. Rappe, "*Applications of a Universal Force Field to Main Group Compounds*", *Journal of the American Chemical Society* **114**, 10046-10053 (1992).
6. C.J. Casewit, K.S. Colwell, A.K. Rappe, "*Application of a Universal Force Field to Organic Molecules*", *Journal of the American Chemical Society* **114**, 10035-10046 (1992).

For calculations with the UFF4MOF parameters:

7. Matthew A. Addicoat, Nina Vankova, Ismot Farjana Akter, and Thomas Heine, "*An extension of the Universal Force Field to Metal-Organic Frameworks*", *J. Chem. Theory Comput.* **10** (2), 880-891 (2013)