



UFF Manual

ADF Modeling Suite 2018

www.scm.com

Sep 10, 2018

CONTENTS

1	Introduction	1
1.1	UFF-GUI	1
1.2	What's new in UFF 2018	1
1.2.1	New features	1
1.2.2	AMS: a new driver program	1
2	Input and Examples	3
2.1	UFF input example	3
3	Periodic UFF	7
4	Parameter files	9
4.1	User-modified force fields (expert option)	9
5	Atom typing behavior	15
6	Required Citations	17
6.1	External programs and Libraries	17
7	References	19

INTRODUCTION

UFF (Universal Forcefield) is a full periodic table forcefield that can be used to calculate single point energies, do geometry optimizations, calculate frequencies, etc... 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](#) (page 19)].

1.1 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.

1.2 What's new in UFF 2018

1.2.1 New features

- Elastic tensor and related properties (e.g. Bulk modulus) (via AMS)
- Linear transit and PES scan (via AMS)
- Lattice optimization (via AMS)
- Geometry optimization under pressure (via AMS)
- More Molecular Dynamics options (via AMS)

1.2.2 AMS: a new driver program

Important: In the 2018 release of the ADF Modeling Suite we introduced a new driver program call **AMS**. We recommend you to first read the General section of the AMS Manual

If you use UFF exclusively via the Graphical User Interface (GUI), this change should not create any issues. If, on the other hand, you create input files *by hand* (or you use UFF via PLAMS), then you should be aware that **shell scripts for UFF-2017 and previous versions are not compatible with UFF-2018 and have to be adjusted to the new setup.**

The example below shows how a shell script for UFF-2017 is converted to UFF-2018.

UFF-2017 shell script (obsolete):

```
#!/bin/sh

# This is a shell script for UFF-2017 which will not work for UFF-2018

$ADFBIN/uff << eor

Task
  RunType GO
End

System
  Atoms
    H 0.0 0.0 0.0
    H 0.9 0.0 0.0
  End
End

Geometry
End

eor
```

UFF-2018 shell script:

```
#!/bin/sh

# This is a shell script for UFF-2018

# The executable '$ADFBIN/UFF' is no longer present.
# You should use '$ADFBIN/ams' instead.

$ADFBIN/ams <<eor
  # Input options for the AMS driver:

  Task GeometryOptimization

  System
    Atoms
      H 0.0 0.0 0.0
      H 0.9 0.0 0.0
    End
  End

  # The input options for UFF, which are described in this manual,
  # should be specified in the 'Engine UFF' block:

  Engine UFF
  EndEngine
eor
```

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

2.1 UFF input example

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

```
$ADFBIN/ams << eor

Task GeometryOptimization

System
  Atoms
    C   1.36012328  -0.14520095   0.60144543   Type=C_3   Charge=0.000000
    C   0.00000000   0.00000000   0.00000000   Type=C_2   Charge=0.000000
    H   2.09833847  -0.46327872  -0.16560721   Type=H_    Charge=0.000000
    H   1.32657807  -0.90546800   1.40917410   Type=H_    Charge=0.000000
    H   1.67935140   0.82750664   1.02977296   Type=H_    Charge=0.000000
    H  -0.83486863   0.30434056   0.62258487   Type=H_    Charge=0.000000
    O  -0.18030374  -0.22462371  -1.18585739   Type=O_2   Charge=0.000000
  End
  BondOrders
    1 5 1.0
    1 4 1.0
    1 3 1.0
    1 2 1.0
    2 6 1.0
    2 7 2.0
  End
End

Engine UFF
EndEngine
```

```
eor
```

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

Specify Elements, coordinates, and bonds:

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/ams << eor

Task GeometryOptimization

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
  BondOrders
    1 5 1.0
    1 4 1.0
    1 3 1.0
    1 2 1.0
    2 6 1.0
    2 7 2.0
  End
End

Engine UFF
EndEngine

eor
```

Specify Elements and coordinates:

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

```
$ADFBIN/ams << eor

Task GeometryOptimization

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
```



```
Engine UFF
EndEngine

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 behavior.

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.
- 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 parameter set used by UFF can be changed via the `Library` key:

```
Library [UFF | UFF4MOF | UFF4MOF-II]
```

Library

Type Multiple Choice

Default value UFF

Options [UFF, UFF4MOF, UFF4MOF-II]

Description Selects the used parameter library.

Aside from the standard UFF forcefield, we ship two parameter sets for Metal-Organic Frameworks:

UFF4MOF parameters

We ship the extended parameter set for Metal-Organic Frameworks created by M.A. Addicoat et al. (2013). Select the `UFF4MOF_general_db`, `UFF4MOF_elements_db` and `UFF4MOF_mmatomtypes_db` files to use these parameters, and check that the proper atom types are detected for your system or set them manually. Please see [7 (page 19)] for details on the parameters.

UFF4MOFII parameters

We ship a second extended parameter set for Metal-Organic Frameworks created by D.E. Coupry et al. (2016). Select the `UFF4MOFII_general_db`, `UFF4MOFII_elements_db` and `UFF4MOFII_mmatomtypes_db` files to use these parameters, and check that the proper atom types are detected for your system or set them manually. Please see [8 (page 19)] for details on the parameters.

4.1 User-modified force fields (expert option)

Finding good UFF parameters 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.

General parameters file

```
Database string
```

Database

Type String

Default value `general_db`

Description Expert option: Select the file that defines the UFF parameters per atom type

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 centers (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].

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 an 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.mmm directory.

Atom type	Example	Description
-----	-----	-----
Al3	alme3tma	Trivalent aluminum
As3+3	asf3	Trivalent arsenic
B_2	bcl3	Planar (sp ²) boron
B_3	b2h5nme2	Tetrahedral (sp ²) boron, including charge transfer adducts and borohydrates
Br	bbr3	Univalent bromine
C_1	c2h2, co	Linear (sp) carbon
C_2	acetone	Planar tricoordinated (sp ²) carbon
C_3	c2h6	Tetrahedral (sp ³) carbon
C_R	c4h6	Resonant, variable bond order (sp ²) carbon.
Cl	socl2	Univalent chlorine
F_	sof2ncl	Univalent fluorine
Ge3	geh3ogeh3	Tetrahedral (sp ³) 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 (sp ²) nitrogen, single-double bond
N_3	ch3nh2	Amine (sp ³) nitrogen, three single bonds
N_3+4	b2h5nme2	Charged amine (sp ³) nitrogen, four single bonds (THIS IS NOT A STANDARD UFF TYPE!)
N_R	c5h5n	Resonant planar (sp ²) 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 (sp ²) oxygen, one double bond.
O_3	h2o	Two-coordinated (sp ³) oxygen, two single bonds
O_3_z	sih3osih3	Special two-coordinated oxygen, for use in Si-O bonds
O_R	c4h4o	Resonant planar (sp ²) 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 be complete, and may be expected 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`.

Elements file

```
ElementsFile string
```

ElementsFile

Type String

Default value `elements_db`

Description Expert option: Select the file that defines the elements known to UFF

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.

MM Atom Types file

```
AtomTypesFile string
```

AtomTypesFile

Type String

Default value `mmatomtypes_db`

Description Expert option: Select the file that defines how UFF determines the atom types

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 forth 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 B2Hs 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 behavior of UFF.

ATOM TYPING BEHAVIOR

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 behavior can be modified in the MMStructureInfo module.

REQUIRED CITATIONS

When you publish results in the scientific literature that were obtained with programs of the ADF package, you are required to include references to the program package with the appropriate release number, and a few key publications.

For calculations with the UFF4MOF parameters: M.A. Addicoat, N. Vankova, I.F. Akter, and T. Heine, *An extension of the Universal Force Field to Metal-Organic Frameworks*, *J. Chem. Theory Comput.* **10**, 880-891 (2013) (<https://doi.org/10.1021/ct400952t>)

For calculations with the UFF4MOFII parameters: D.E. Coupry, M.A. Addicoat, and T. Heine, *An Extension of the Universal Force Field for Metal-Organic Frameworks*, *J. Chem. Theory Comput.* **12**, 5215-5225 (2016) (<https://doi.org/10.1021/acs.jctc.6b00664>)

6.1 External programs and Libraries

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

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). (<https://doi.org/10.1021/ja00051a040>)
2. A.K. Rappe, W.A. Goddard III, *Charge Equilibration for Molecular Dynamics Simulations*, *The Journal of Physical Chemistry* 95, 3358-3363 (1991). (<https://doi.org/10.1021/j100161a070>)
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). (<https://doi.org/10.1021/ja00009a002>)
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). (<https://doi.org/10.1021/j100389a010>)
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). (<https://doi.org/10.1021/ja00051a042>)
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). (<https://doi.org/10.1021/ja00051a041>)

For calculations with the UFF4MOF parameters:

7. M.A. Addicoat, N. Vankova, I.F. Akter, and T. Heine, *An extension of the Universal Force Field to Metal-Organic Frameworks*, *J. Chem. Theory Comput.* 10, 880-891 (2013) (<https://doi.org/10.1021/ct400952t>)

For calculations with the UFF4MOFII parameters:

8. D.E. Coupry, M.A. Addicoat, and T. Heine, *An Extension of the Universal Force Field for Metal-Organic Frameworks*, *J. Chem. Theory Comput.* 12, 5215-5225 (2016) (<https://doi.org/10.1021/acs.jctc.6b00664>)