University of Namur
Laboratoire de Physico-Chimie Informatique

Daniel P. Vercauteren
Director, Professor at the Sciences Faculty

Rue de Bruxelles, 61
B-5000 Namur, Belgium
Phone +32 (0)81 72 45 34
Fax +32 (0)81 72 54 66
Daniel.vercauteren@fundp.ac.be

Namur, December 3rd, 2007

Professor **J. W. Ponder**
Washington University School of
Medicine
Biochemistry, Campus Box 8231
660 South Euclid Avenue
Saint Louis, MO 63110
USA

Dear Professor Ponder,

Sure, you remember I wrote you, in September 2006, a letter about a partial implementation of the MMFF94[1] force field in Tinker.

In your last mail, you were wondering about the work left to fully implement MMFF94 in Tinker. We are pleased to announce you that since then we added most of the missing features, delivering an almost complete implementation of MMFF94 in Tinker.

This implementation has been validated thanks to a validation suite[2] available on the Computational Chemistry List website[3], containing 761 molecular structures, with geometry optimised at a convergence degree of 0.000001 kcal/mol (details in Annex 1). The validation was carried on 740 molecules. The remaining 21 molecules contain ionic aromatic rings, which are still not supported by our implementation.

Something that we omitted to talk about in our previous mails is the definition of the Bond Types. This feature of MMFF94 is explained in Annex 2.

There remains two features that we didn't implemented because we believe it would take too much programming and validation time: the **second derivatives of the energy terms** and parameters for **ionic aromatic rings**. We provide more information about these features in Annex 3.

Regarding the implementation of the other features, i.e., support for small rings (3-5 membered), angle bending energy expression for linear and near-linear valence angles (sp-hybridised atoms), parameters for ions and default parameters assignment, we noted two major inconsistencies between MMFF94's literature and the results reported in the validation suite, as developed in Annex 4.

The modified sources have been compiled on a Sun Fire V20z powered by two 64 bits AMD® Opteron™ 248 processors running at 2.2 GHz, with 4 GB of total memory, by the Intel® Fortran Compiler for Linux, under Debian Linux 3.1 (Sarge), kernel 2.6.18.

The *MMFF94.tar.gz* archive file including the modified sources (*source_bis*) and MMFF94's parameter file in Tinker's format (*params_bis/mmff94.prm*) can be downloaded at http://perso.fundp.ac.be/~nstaelen/MMFF94.

Please, let me know what you think about our work. Prof D.P. Vercauteren, Promoter of my Ph. D. Thesis thinks it could be interesting if we could meet, either at your place or here, to discuss it more deeply. What do you think about visiting us? If you are coming to Europe in a near future, it would be our pleasure to invite you. You could, if you wish, present a talk and we would intervene in part of the expenses (part of travel, hotel, …).

Thanking you in advance for your interest in our work and remaining at your disposal for any question or remark you would have.

With best regards,

Yours sincerely.

Nicolas Staelens
Ph. D. Student
PCI Laboratory
University of Namur

cc / Prof. D. P. Vercauteren, Director of the Chemistry Department and Director of the PCI Laboratory, University of Namur, Belgium

In order to validate our implementation, we proceeded in 4 steps:
1. We used the *analyze* function to get the energy and its components for each structure;
2. We then re-optimised the structure with the *minimize* function (convergence degree of 0.000001 kcal/mol, or 0.00001 kcal/mol when the convergence could not be reached);
3. The energy was then re-calculated with *analyze*, and;
4. The two structures, before and after re-optimisation, were superposed by the *superpose* function, ensuring the correctness of the first derivative equations of the energy.

The energy differences between the values given in the validation suite (file *MMFF94_opti.log*) and our calculations were all below 0.01 kcal/mol, and the RMS deviation was 0.002291 $Å^2$ at most. We just noted one exception, the structure referred as "CYGUAN01", i.e., 4,6-diamino-1-(p-chlorophenyl)-1,2-dihydro-2,2-dimethyl-s-triazine hydrochloride, for which we disagree with the values reported in the *MMFF94_opti.log* file: the 3-2-7-6 angle has been assigned torsion parameters corresponding to Torsion Type 0, tough, according to MMFF94's literature[1], that torsion angle is of Torsion Type 2 (see the *CYGUAN01.jpg* picture[*]). The input files (.sh script files, .xyz structure files, .key and BOND files) and output files (.out files) are available in the *validation_suite_MMFF94.tar.gz*[*] archive file.

---

[*]   Available at http://perso.fundp.ac.be/~nstaelen/MMFF94

MMFF94 makes usage of a Bond Type system which allows, for instance, to make the difference between single and double bonds in an alternating sequence like C=C-C=C-C=C. As all of those carbon atoms would have the same atom type, the parameters would be the same for the single and double bounds. Though, the Bond Type system resolves that problem: for each bond of any molecular system, we have to define whether it can or cannot be a multiple bond. If it is a single bond that could be a multiple bond, then it has a Bond Type of 1. Otherwise, it is 0.

To make it clearer, if for example you have two sp2-hybridized carbons linked by a single bond, it has a Bond Type of 1 because there **could** be a double bond between those two atoms. This is illustrated by the picture hereunder – the Bond Types are given in white.
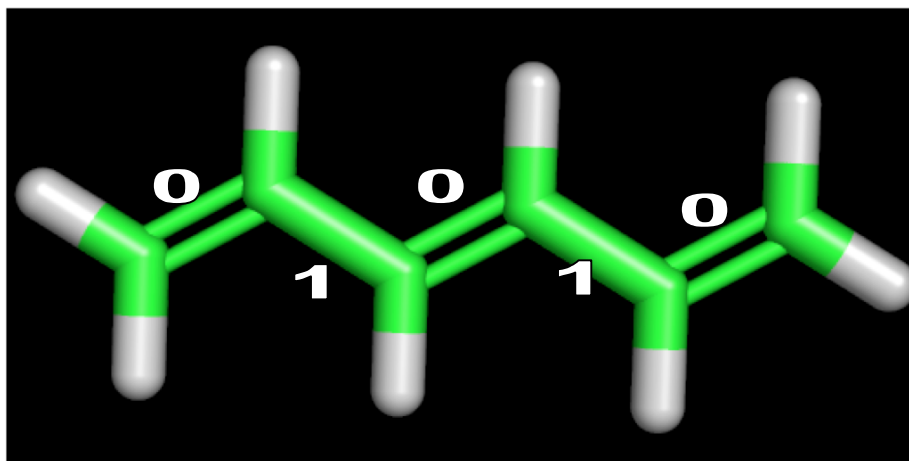
In our implementation, the bond type has to be defined manually in an additional file named BOND, in the same folder as the .xyz and .key files. On the first line, the user has to write the number of bonds that have a Bond Type of 1 in the .xyz structure file, and then the numbers of the atoms of each pair having a Bond Type of 1, with the numbers given from the lowest to the highest. Examples can be found in *validation_suite_TINKER.tar.gz*[*] .

Concerning the second derivatives, I just avoided it because it sounded more complex and not essential to my present Ph. D. thesis.

While we were validating the parameters implementation for ions, we found out that the atomic charges in aromatic rings were not constant. Actually, this is a particularity that has not been reported in MMFF94's literature, but we found some explanation on the Open Eye Scientific Software website[4]. Actually, the assignment of the atom types has to be done in two stages: first, a "non-aromatic" assignment of the atom types is performed for all the atoms, and then, the aromaticity is taken into account to switch the atom types of the atoms involved in aromatic rings to an "aromatic" atom type. The parameters used to assign the charges on the aromatic atoms depend on the "non-aromatic" atom type, the "aromatic" atom type and the number of occurrence of that atom type in the aromatic ring. This procedure thus requires a program aimed at assigning the MMFF94 atom types ("non-aromatic" and "aromatic") to each atom, which, considering the large number of atom types (212), would be a far too demanding task in the context of my Ph. D. thesis.

When validating the implementation of the empirical rule for supplying missing bond stretching parameters (*cf. kbond.f*), we realized that the results obtained with the rule described in MMFF94's literature did not match the values reported in the validation suite. After carefully ensuring the absence of any bug, we noted that the author (Thomas A. Halgren) was referring to an older (simpler) rule[5] as a basis for the "new" rule. We tried that older rule instead of the "new" one, and all the results matched with the validation suite. We thus kept the "old" rule in the *kbond.f* file, but we left the possibility to switch easily to the "new" one by simply un-commenting some lines in the *kbond.f* file.

Another feature is the Torsion Type assignment (*cf. ktors.f*). This is, according to MMFF94's literature, only dependant of the Bond Types of the three bonds forming the torsion angle. However, during the validation, we remarked that the Torsion Type 2 (e. g., in a sequence of atoms 1-2-3-4, having a Bond Type of 0 for the central 2-3 bond, and a Bond Type of 1 for the 1-2 and/or 3-4 bonds) needed a sharper definition to match the values of the validation suite. As a corollary of the Bond Types definition, a Bond Type of 0 means "a multiple bond OR the impossibility of the presence of a multiple bond". In order to have a good correspondence with the validation suite, we had to restrain the conditions on the central bond, so that the criteria now involves the fact that, if there is a multiple bond, it cannot be delocalised.

1.   - T. A. Halgren, "Merck Molecular Force Field. I. Basis, Form, Scope, Parametrization, and Performance of MMFF94", J. Comput. Chem. 17, pp. 490-519 (1996)

        - T. A. Halgren, "Merck Molecular Force Field. II. MMFF94 van der Waals ans Electrostatic Parameters for Intermolecular Interactions", J. Comput. Chem. 17, pp. 520-552 (1996)

        - T. A. Halgren, "Merck Molecular Force Field. III. Molecular Geometries and Vibrational Frequencies for MMFF94", J. Comput. Chem. 17, pp. 553-586 (1996)

        - T. A. Halgren and R. B. Nachbar, "Merck Molecular Force Field. IV. Conformational Energies and Geometries for MMFF94", J. Comput. Chem. 17, pp. 87-615 (1996)

        - T. A. Halgren, "Merck Molecular Force Field. V. Extension of MMFF94 Using Experimental Data, additional Computational Data, and Empirical Rules", J. Comput. Chem. 17, pp. 616-641 (1996)

2.   HTTP://SERVER.CCL.NET/CCA/DATA/MMFF94/INDEX.SHTML

3.   HTTP://WWW.CCL.NET

4.   HTTP://WWW.EYESOPEN.COM/DOCS/HTML/QUACPAC_0_9_B/NODE30.HTML

5.   R. Blom and A. Haaland, "A Modification of the Schomaker-Stevenson Rule for Prediction of Single Bond Distances", J. Mol. Struct. 128, pp.21-27 (1985)