|  |
| --- |
| Direct Forcefield  (Version 8.0)  User Guide  Rev. 2024.07.20    Copyright 2000-2024 © Aeon Technology  Usage of the Direct Forcefield software including this documentation is restricted to purposes in accordance with the Direct Forcefield license agreement. Direct Forcefield is a trademark of Aeon Technology, Inc.  *GROMACS* is software licensed under the GNU General Public License. The main distribution site is at [www.gromacs.org](http://www.gromacs.org). *LAMMPS* is distributed as an open source code under the terms of the GPL. *LAMMPS* is distributed by Sandia National Laboratories, a US Department of Energy laboratory, at <http://lammps.sandia.gov>. |

Contents

[CHAPTER 1 - INTRODUCTION 8](#_Toc172367536)

[1-1. Overview 8](#_Toc172367537)

[1-2. Feature List 9](#_Toc172367538)

[1-3. Typical Use Cases 10](#_Toc172367539)

[Using TEAM forcefields 10](#_Toc172367540)

[Generating Additional TEAMFF Forcefields 12](#_Toc172367541)

[Making On-the-fly Forcefield (OTF-FF) 13](#_Toc172367542)

[1-4. Getting Start 14](#_Toc172367543)

[System Requirements 15](#_Toc172367544)

[Installation 15](#_Toc172367545)

[Uninstalling 15](#_Toc172367546)

[Starting 15](#_Toc172367547)

[Licensing 16](#_Toc172367548)

[Local License 17](#_Toc172367549)

[Floating License 17](#_Toc172367550)

[More Options of dfflsv.exe 18](#_Toc172367551)

[Install External Software 19](#_Toc172367552)

[CHAPTER 2 – TUTORIALS 20](#_Toc172367553)

[2-1. Section A - Essential Functionality 20](#_Toc172367554)

[Tutorial A1 - Building a Model of Molecule 20](#_Toc172367555)

[Tutorial A2 - Working With a Liquid Model 26](#_Toc172367556)

[Tutorial A3 - Working with Polymers 30](#_Toc172367557)

[Tutorial A4 - Working with Proteins 34](#_Toc172367558)

[Tutorial A5 - Mixing Two Forcefields 36](#_Toc172367559)

[2-2. Section B - Parameterization 39](#_Toc172367560)

[Tutorial B1 - Fitting Parameters for a Small Molecule 39](#_Toc172367561)

[Tutorial B2 - Fitting Parameters Of A FLexible Molecule 43](#_Toc172367562)

[Tutorial B3 - Fitting a Molecular Dimer 47](#_Toc172367563)

[Tutorial B4 - Fitting Liquid Data 50](#_Toc172367564)

[2-3. Section C - Developing Force Fileds 53](#_Toc172367565)

[Tutorial C1 - Make A Training Set 53](#_Toc172367566)

[Tutorial C2 - Make a New Forcefield 54](#_Toc172367567)

[Tutorial C3 - Import A Forcefield 57](#_Toc172367568)

[Tutorial C4 - Hypervalence Molecules 63](#_Toc172367569)

[2-4. Section D – Quantum Mechanics Data Forcefield 64](#_Toc172367570)

[Tutorial D1 – Use QMDFF Database 64](#_Toc172367571)

[Tutorial D2 – Make new Forcefields 66](#_Toc172367572)

[CHAPTER 3 – CONCEPTS, METHODS AND ALGORITHMS 69](#_Toc172367573)

[3-1. Project and Files 69](#_Toc172367574)

[Project 69](#_Toc172367575)

[Project Navigator 69](#_Toc172367576)

[File Types 69](#_Toc172367577)

[3-2. Forcefields 70](#_Toc172367578)

[Forcefield type 70](#_Toc172367579)

[Typing Rule 71](#_Toc172367580)

[Parameters 72](#_Toc172367581)

[3-3. Molecular Structure 72](#_Toc172367582)

[Connectivity 72](#_Toc172367583)

[Essential Properties of Atoms 73](#_Toc172367584)

[Bond Orders 73](#_Toc172367585)

[Hydrogen Atoms 73](#_Toc172367586)

[Formal Charges 73](#_Toc172367587)

[Aromaticity 74](#_Toc172367588)

[Assumptions and Recomendations 74](#_Toc172367589)

[3-4. Atom Type Definitions 74](#_Toc172367590)

[Hierarchical Atom Type (HAT) Scheme 75](#_Toc172367591)

[Default Definition 75](#_Toc172367592)

[Extended Default Definition 76](#_Toc172367593)

[Complet Scripting Definition 77](#_Toc172367594)

[Atom types of hypervalence molecules 78](#_Toc172367595)

[Apparent and Equivalent Atom Types 79](#_Toc172367596)

[3-5. Overview of Parameterization 79](#_Toc172367597)

[Forcefield type and Atom Tying Rules 79](#_Toc172367598)

[Training Set and Data 80](#_Toc172367599)

[Parameterization Methods 80](#_Toc172367600)

[Systematic Procedure 81](#_Toc172367601)

[3-6. Automatic Parameters 82](#_Toc172367602)

[3-7. Quantum Mechanics Data 84](#_Toc172367603)

[Sampling Conformational Energy Space 84](#_Toc172367604)

[Sampling Intermolecualr Energy Curve 86](#_Toc172367605)

[QMD File 86](#_Toc172367606)

[3-8. Fit QM Data 87](#_Toc172367607)

[Methods 87](#_Toc172367608)

[Charge Parameters 88](#_Toc172367609)

[Valence Parameters 89](#_Toc172367610)

[Molecular Dimers 90](#_Toc172367611)

[Specific Properties 90](#_Toc172367612)

[3-9. Fit Properties of Liquid 91](#_Toc172367613)

[Perturbation Theory 91](#_Toc172367614)

[Temperature Dependent Parameters 93](#_Toc172367615)

[3-10. Simulations 94](#_Toc172367616)

[Assigning a Forcefield 94](#_Toc172367617)

[Internal Simulation Engines 95](#_Toc172367618)

[External Simulation Engines 97](#_Toc172367619)

[3-11. TEAMFF Database 98](#_Toc172367620)

[Direct Forcefield Database 98](#_Toc172367621)

[Customized Forcefield Tables 101](#_Toc172367622)

[Internal Extensibility of Atom Types 102](#_Toc172367623)

[External Extensibility using TEAMFF 103](#_Toc172367624)

[3-12. Quantum Mechanical Data Forcefield 104](#_Toc172367625)

[Cutomized vs. Centralzied Parametirzation 104](#_Toc172367626)

[What is QMDFF 104](#_Toc172367627)

[How to Use QMDFF 106](#_Toc172367628)

[CHAPTER 4 – REFERENCE OF COMMANDS 108](#_Toc172367629)

[4-1. File Menu - Manage Project and Files 108](#_Toc172367630)

[Commands for Projects 108](#_Toc172367631)

[Commands for Files 109](#_Toc172367632)

[4-2. Edit Menu – Sketching, Editing and Modifying Models 109](#_Toc172367633)

[Menu Commands 109](#_Toc172367634)

[4-3. Build Menu - Building More Complicated Models 110](#_Toc172367635)

[Group and Subset 110](#_Toc172367636)

[Bulk Models 111](#_Toc172367637)

[Interphase Models 112](#_Toc172367638)

[Super Cells 113](#_Toc172367639)

[Fragments and Sampling 113](#_Toc172367640)

[4-4. View menu - Managing How Models Are Displayed 115](#_Toc172367641)

[View Commands 115](#_Toc172367642)

[Toolbar Commands 116](#_Toc172367643)

[4-5. TEAMFF menu – the TEAM Forcefield Database 117](#_Toc172367644)

[Managing the Database 117](#_Toc172367645)

[Use the Database 119](#_Toc172367646)

[Expand the Database 120](#_Toc172367647)

[Make a New Database 120](#_Toc172367648)

[4-6. Simulation Menu - Internal and External Simulation Methods 120](#_Toc172367649)

[Internal Simulations 120](#_Toc172367650)

[External Simulations 122](#_Toc172367651)

[4-7. QMDFF Menu – quantum Mechanics Data Forcefields 123](#_Toc172367652)

[Manage the Database 124](#_Toc172367653)

[Automatic Procedures 124](#_Toc172367654)

[Toolkits 126](#_Toc172367655)

[4-8. Forcefield menu - Managing Forcefield 128](#_Toc172367656)

[Managing Forcefield 129](#_Toc172367657)

[Forcefield Editor 129](#_Toc172367658)

[Using a Forcefield 130](#_Toc172367659)

[Advanced Topics on Forcefield 131](#_Toc172367660)

[4-9. Parameters Menu - Fit and validate forcefield parameters 133](#_Toc172367661)

[Fit QM Data 133](#_Toc172367662)

[Fit Liquid Properties 135](#_Toc172367663)

[Fit Molecular Properties 135](#_Toc172367664)

[Additional Tools 137](#_Toc172367665)

[4-10. QMD Menu - Compute and Organize Quantum Mechanics Data 138](#_Toc172367666)

[QM Calculations 138](#_Toc172367667)

[Quantum Mechanics Data (QMD) 139](#_Toc172367668)

[4-11. Help Menu – Information, License, and Document 140](#_Toc172367669)

[4-12. Toolbar Commands 140](#_Toc172367670)

[Toolbar File Commands 140](#_Toc172367671)

[Toolbar Sketching Commands 140](#_Toc172367672)

[Additional Shor-cut Keys on Atoms 142](#_Toc172367673)

[4-13. Mouse Commands on Project Navigator 143](#_Toc172367674)

[CHAPTER 5 – Executable PROGRAM COMMANDS 145](#_Toc172367675)

[5-1. Model Commands 145](#_Toc172367676)

[5-2. Force Filed Commands 147](#_Toc172367677)

[5-3. Export Commands 147](#_Toc172367678)

[5-4. TEAMFF Database Commands 148](#_Toc172367679)

[5-5. Fit Force Field Commands 149](#_Toc172367680)

[5-6. MM Computation Commands 151](#_Toc172367681)

[5-7. QM Computation Commands 151](#_Toc172367682)

[CHAPTER 6 – FUNCTIONAL FORMS 152](#_Toc172367683)

[6.1 Forcefield types 152](#_Toc172367684)

[TEAM 153](#_Toc172367685)

[CHARMM 155](#_Toc172367686)

[AMBER 156](#_Toc172367687)

[CFF 157](#_Toc172367688)

[DREIDING 158](#_Toc172367689)

[GROMACS 158](#_Toc172367690)

[6.2 Functions 160](#_Toc172367691)

[A) Valence Functions 160](#_Toc172367692)

[B) The Nonbond Functions 164](#_Toc172367693)

[C) The Cross Coupling Terms 166](#_Toc172367694)

[6.3 Functions and Parameters 166](#_Toc172367695)

[CHAPTER 7 – File Formats 172](#_Toc172367696)

[7.1 Molecular Structural Data (MSD) File 172](#_Toc172367697)

[7.2 Potential Parameter File (PPF) File 173](#_Toc172367698)

[7.3 Quantum Mechanics Data (QMD) File 175](#_Toc172367699)

[7.4 Atom Type Definition (DEF, EXT) File 179](#_Toc172367700)

[REFERENCES 181](#_Toc172367701)

# CHAPTER 1 - INTRODUCTION

## 1-1. Overview

Direct Forcefield (DFF) is specialized software designed for the development and deployment of forcefields. It comes in two versions: Standard and Professional. The Standard version is intended for the deployment of forcefields, while the Professional version offers additional functionalities for forcefield development.

Within the DFF environment, forcefield management is facilitated through computer database technology. The Direct Forcefield Database (TEAMFF) is a crucial component that supports multiple database tables, each representing an independent forcefield. These forcefields are defined by their functional characteristics, typing rules, and parameters. The modular nature of TEAMFF allows for the independent development of each forcefield. This extensibility is achieved without compromising accuracy, even when multiple forcefields within the same functional category are used concurrently.

One notable contribution of DFF is the creation of a forcefield database named TEAMFF, included as part of the DFF package. TEAMFF encompasses a range of forcefields categorized under common functional types, including AMBER, CHARMM, CFF, DREDING, and TEAM. Notably, the TEAM and AMBER forcefield types are consistently updated. These diverse forcefields collectively cover a wide array of organic molecules, polymers, drug-like compounds, zeolites, ionic liquids, and metal oxides. Moreover, TEAMFF seamlessly interfaces with prominent molecular dynamics packages, GROMACS and LAMMPS, directly from the DFF platform.

A pivotal aspect of DFF is its capacity to facilitate the development of new parameters and forcefields. A suite of parameterization tools has been meticulously crafted to aid in various tasks: assembling training sets, generating molecular fragments and quantum mechanics data (QMD), deriving forcefield parameters through QMD and physical data fitting, and validating the derived parameters. This comprehensive toolkit collectively empowers users with the ability to intricately refine and substantiate forcefield parameters. Furthermore, an automated workflow streamlines the utilization of these tools, enabling the creation and validation of forcefields in a dynamic manner. This innovative approach diverges from conventional practices, as DFF enables the generation of on-the-fly forcefield (OTF-FF) directly from QMD data of molecular fragments. This paradigm shift empowers precision over breadth, resulting in the creation of highly accurate forcefields tailored to specific target molecules.

DFF is delivered with an intuitive window-based graphical user interface, ensuring easy accessibility to its functionalities. It is compatible with multiple operating systems, including Windows and Linux. While the graphical interface offers convenience, many functions can also be executed through command-line operations. This flexibility enables seamless integration of the software with other simulation environments.

## 1-2. Feature List

**(*\* indicates functions of the Professional version*)**

Model Builder

* Sketch, edit and modify molecular structures.
* Supports all elements on the periodic table.
* Automatically edit molecular topologies.
* Group atoms by charges or by connectivity.
* Pack and unpack periodic simulation boxes.
* Edit simulation cell parameters.
* Build liquid and interface models.
* Reads popular molecular structure data files - MOL, MOL2, PDB, CAR
* Supports solid and surface structures.
* Build super cells from 3-D or 2-D structures.

Visualization

* Project Navigator allows easy access.
* Portable, self-contained project files facilitate sharing of results with colleagues.
* Mouse and short-cut key operations for convenience.
* Display atomic and molecular properties.
* Measure molecular properties.
* Customize Look-and-Feel, including adjustable font sizes.

TEAMFF Database

* Integration of different independent forcefields in one database
* Supports common forcefield types - TEAM, AMBER, CHARMM, CFF, DREIDING, provide parameters in each forcefield type as the base for continue development.
* Provides high quality parameters with great coverage in common organic molecules, synthetic polymers, solid state materials, transition metal complexes and drug-like compounds in TEAM and AMBER forcefield types.
* Graphic interface for managing and using database, automatically assign atom types using multiple forcefields, with options to handle missing parameters.

TEAMFF Expansion \*

* Full functionality for managing the database contents.
* Create customized forcefields to be used with the provided forcefields.
* Check out original atom types and parameters for examination and extended development.
* Check in new parameters, automatically set version numbers.
* Functionality for importing common forcefields into database.
* Advanced handling of missing parameters, automatically build molecular fragments for parameterization.

Forcefield Development\*

* Support popular forcefield types: TEAM, AMBER, CHARMM, CFF, DREIDING and customized forcefield types using common functional forms.
* Support preset (default) and customized atom-typing rules.
* Automatic generation of molecular fragments for parameterization.
* Automatic generation quantum mechanics data (QMD) by sampling potential energy surfaces.
* Automatic fit charge and valence parameters using QMD.
* Automatic optimizing the Lennard-Jones parameters by fitting molecular simulation data against experimental liquid data.
* Database storage of QMD and parameterized forcefield files (QMDFF).
* Automatic workflow enables on-the-fly forcefield (OTF-FF) parameterization for target molecular systems, which can be drug molecules, liquids, and macromolecules, pure or mixed components.

Simulation Capability

* Energy calculation, normal mode analysis, structure optimization and molecular dynamics simulation for validation of forcefields.
* Interface to quantum mechanics software, GAUSSIAN.
* Integrated with GROMACS and LAMMPS.

User-Friendly Features

* Toolbars for quick and easy access to common commands.
* Project Navigator for convenient organization.
* Integrated project and disk file management.
* Online documentation and tutorials.
* Supports Windows and Linux platforms.
* Local and floating licensing.
* User guide document, on-line help and tutorials.

## 1-3. Typical Use Cases

### Using TEAM forcefields

TEAMFF encompasses a comprehensive assortment of developed forcefields catering to organic molecules, polymers, solid materials, transition metal complexes, and drug-like molecules. Among these forcefields, two primary families within the TEAM and AMBER forcefield types are continually updated based on user input and advancements in simulation technology.

TEAMFF functions much like conventional forcefields, but its most significant advantage lies in its capacity to seamlessly combine multiple forcefields within the same functional. This characteristic significantly expands coverage while upholding the forcefield's high precision. The general procedure for employing TEAMFF forcefields is outlined below:

* 1. **Construct or Import a Molecular Model:**

The creation of straightforward models, like isolated molecules or molecular liquid representations using DFF's provided functions, is straightforward. Complex systems can be imported through popular file formats. Upon importing or creating models, DFF automatically checks or reconstructs molecular topologies, priming them for atom type and parameter assignments.

* 1. **Select a Forcefield Type:**

Opt for a specific forcefield type, which consists of distinct functional forms. DFF supports various forcefield types, including TEAM, AMBER, CHARMM, CFF, DREIDING, and more. The choice of functional is largely driven by intended applications and simulation engines, with TEAMFF forcefields predominantly designed for TEAM and AMBER forcefield types.

* 1. **Choose Forcefields:**

Within the chosen forcefield type, select one or multiple forcefields for application. If multiple forcefields are employed, they can be either used in the specified order or allocated to distinct subsets. For models containing multiple molecules, parameters for different molecules might originate from various forcefields. Interactions between different molecules are often represented solely through nonbond parameters. Utilizing subsets to assign forcefields offers greater precision and flexibility but necessitates predefined subset definitions. By default, forcefields are employed in the order they are chosen, where a forcefield must encompass all parameters for a given molecule; otherwise, the model proceeds to the next forcefield. Typically, order is arranged based on generality, with the most specific and accurate forcefield being applied first, followed by more generic ones.

* 1. **Parameter Assignment and Saving:**

Upon locating all necessary parameters, the model is attributed with atom types and charges, and subsequently saved as a molecular structural data (.msd) file. Additionally, a forcefield file (.ppf) containing all required parameters with accurate atom types is generated. These two files are utilized in tandem. An alternative exists to merge these two files into one (.spf).

* 1. **Addressing Missing Parameters:**

In cases where parameters are absent, options include transferring parameters or generating automatic parameters alongside database-acquired ones. A more rigorous approach involves the development of new parameters (outlined below).

### Generating Additional TEAMFF Forcefields

New forcefields can be developed from the ground up or built upon existing parameters. Regardless of the approach, these new forcefields can coexist alongside pre-existing ones within the same forcefield type. The general procedure for creating a new forcefield is outlined as follows:

* 1. **Select a Forcefield type and Typing Rule:**

Choose a forcefield type and an atom typing rule. Since each forcefield operates independently, typing rules can be established autonomously. The choice of typing rule is influenced by the complexity of parameterization scope. Generally, it's advisable to opt for simpler typing rules. For those seeking highly flexible typing rules, custom scripts can be employed. The recommended is the programmed Default typing rules.

* 1. **Create a New Forcefield Table**:

Once the forcefield type and typing rule are decided, initiate the creation of a fresh forcefield table. This table can be appended to the existing TEAMFF database or introduced into a newly established database. In most scenarios, integrating the new table into TEAMFF is preferable, allowing seamless collaboration with other forcefields.

* 1. **Select Molecule(s) for Parameterization:**

Start with selecting the molecule(s) to be parameterized. Initialize a new forcefield (.ppf) file using the same typing rule employed during forcefield creation; this is necessary for successful check-in. Utilize the empty forcefield to generate fragment molecules that are suited for parametrization using the fragmentation tool.

* 1. **Perform Gaussian Calculations for Training Set:**

Execute Gaussian calculations to generate quantum mechanics data (QMD) files for the training set. For each molecule, it's important to sample not only the global minimum energy but also the potential energy surface within the conformational space. DFF offers several tools to facilitate this process, with each molecule being stored in a separate QMD file.

* 1. **Global Fitting of QMD Files:**

Unless dealing with an excessively large training set, perform a comprehensive fitting of all QMD files to account for correlations. If the training set exceeds DFF's capacity, divide it into subsets. Start the parameterization process from common functional groups to enable parameter transfer to more complex subsets.

* 1. **Parameter Fitting and Validation:**

Conduct fitting of QMD files to derive atomic partial charges and valence parameters within the .ppf file. At this stage, retain the Lennard-Jones (L-J) parameters as suggested by DFF. Validate the fit outcomes to ensure success. Default L-J parameters, sourced from prior work, are reasonable but might not be optimal for your system. If experimental liquid density and heat of vaporization data are available, optimization of nonbond parameters (L-J and charge parameters) is possible. Adjusting charge parameters is crucial, considering that they're derived from QMD of isolated molecules, which might not suit molecules in condensed phases with high polarization. Modifying any nonbond parameters necessitates repeating the valence parameter fit due to the interconnected nature of nonbond and torsional parameters.

* 1. **Finalize Parameterization and Integration:**

Upon completing the parameterization process, check the .ppf file into the designated forcefield table. This newly generated forcefield is now ready for individual or collaborative use within the same forcefield type.

For optimizing a few parameters within an existing forcefield, an alternative involves checking out pertinent parameters into a .ppf file from TEAMFF. Refine the .ppf file until all parameters meet requirements, then create a new table for checking in the updated .ppf file."

### Making On-the-fly Forcefield (OTF-FF)

Despite the convenience offered by traditional forcefields, such as those stored in the TEAMFF database, they consistently face challenges in terms of coverage and accuracy. This stems from the process of constructing forcefields through fitting extensive datasets meant to represent the intended chemical space. The assumption of transferability among molecules within this space, on which forcefield parameters rely, doesn't always yield optimal results. In essence, any data model derived from regressing a diverse and large dataset, regardless of the regression methodology employed, inherently involves a trade-off between coverage and accuracy.

Nevertheless, these conventional forcefields are designed for broad usability among various users, fulfilling their purpose. However, from an individual user's perspective, this broad coverage may not always be the ideal solution. Given that it's possible to construct a forcefield directly from pertinent quantum mechanical data, a superior approach involves representing a smaller, specific subspace of interest. This approach ensures the most precise representation of the underlying quantum mechanical description (QMD). This is where Directed Forcefield (DFF) comes into play, offering the opportunity for such specialization. The typical procedure unfolds as follows:

* 1. **Fragment and Identify Molecular System:**

Identify all molecular fragments that signify local (intra-molecular) interactions within a target molecule system. This could involve a complex of drag molecules and a bio-macromolecule or a collection of pure or mixed molecules, small or large, contained with a periodic condition. DFF functions aid in this fragment identification process. The unique collection of smiles strings representing the fragments is referred to as MID (Molecular Identifier).

* 1. **Search QMDFF Database for Forcefield:**

Use MID to search the QMDFF database for forcefield. If the forcefield has been previously parameterized, retrieve it, assign it to the molecular system and proceed to simulation. If MID lacks a stored forcefield, a new one must be created as follows.

* 1. **Search or Make QMD files:**

This begins with a search in the QMDFF database to check if any of the necessary fragments have existing QMD files. If yes, these files are downloaded, if not, DFF will launch quantum mechanical calculations to generate new QMD files and upload them to the database.

* 1. **Make Forcefield:**

With all QMD files prepared, initiate an automatic fitting procedure to generate a comprehensive forcefield encompassing these fragments. This procedure autonomously validates the fit, highlights issues if they arise, and if successful, uploads the forcefield to QMDFF, assigning it to the target molecular system.

A streamlined batch workflow encapsulates all the above steps, enabling execution with a single command. In scenarios involving a large assortment of molecules needing coverage, it's expedient to generate all required fragments (many of which can be shared across different molecules) and perform batch QM calculations. Once these calculations conclude, the resultant QMD files are uploaded to the QMDFF database. During usage, only steps 1 and 2 are necessary.

In this process, the Lenard-Jones parameters are preset to default, optimized values derived from fitting experimental liquid data of molecules with similar atom types. While these parameters can be fine-tuned using the DFF toolkits and available experimental data, the provided values serve as a starting point.

## 1-4. Getting Start

### System Requirements

Linux or Windows operating systems with at least 1 GB of free disk space and 1 GB of memory are required.

1. *Windows* – versions after XP are supported.
2. *Linux* - kernel version 2.6.9 and above are supported.

### Installation

DFF requires Java Runtime Environment (JRE). The DFF installation package for *Windows* includes JRE, which will not replace any JRE installed on your computer. On *Linux* platform, ensure that JRE has been installed, which can be verified by using *"java -version"* in the command prompt. For example:

*$ java -version*

*openjdk version "1.8.0\_201"*

*OpenJDK Runtime Environment (build 1.8.0\_201-b09)*

*OpenJDK 64-Bit Server VM (build 25.201-b09, mixed mode)*

On Windows, run the installer executable such as “dff8001-WinJre.exe” to start the installation process and follow the instructions of the installer. You must have written permission to the installation directory.

On Linux, unpack the release package by typing “$tar zxvf dff8.tar.gz”, the installation directory should be placed in the current directory.

The installation directory is referred to **<dff-root>** in this document. It contains several subdirectories such as: /bin64x, /data, /database, /forcefields, /images, /jar, /License, etc., and some script files.

### Uninstalling

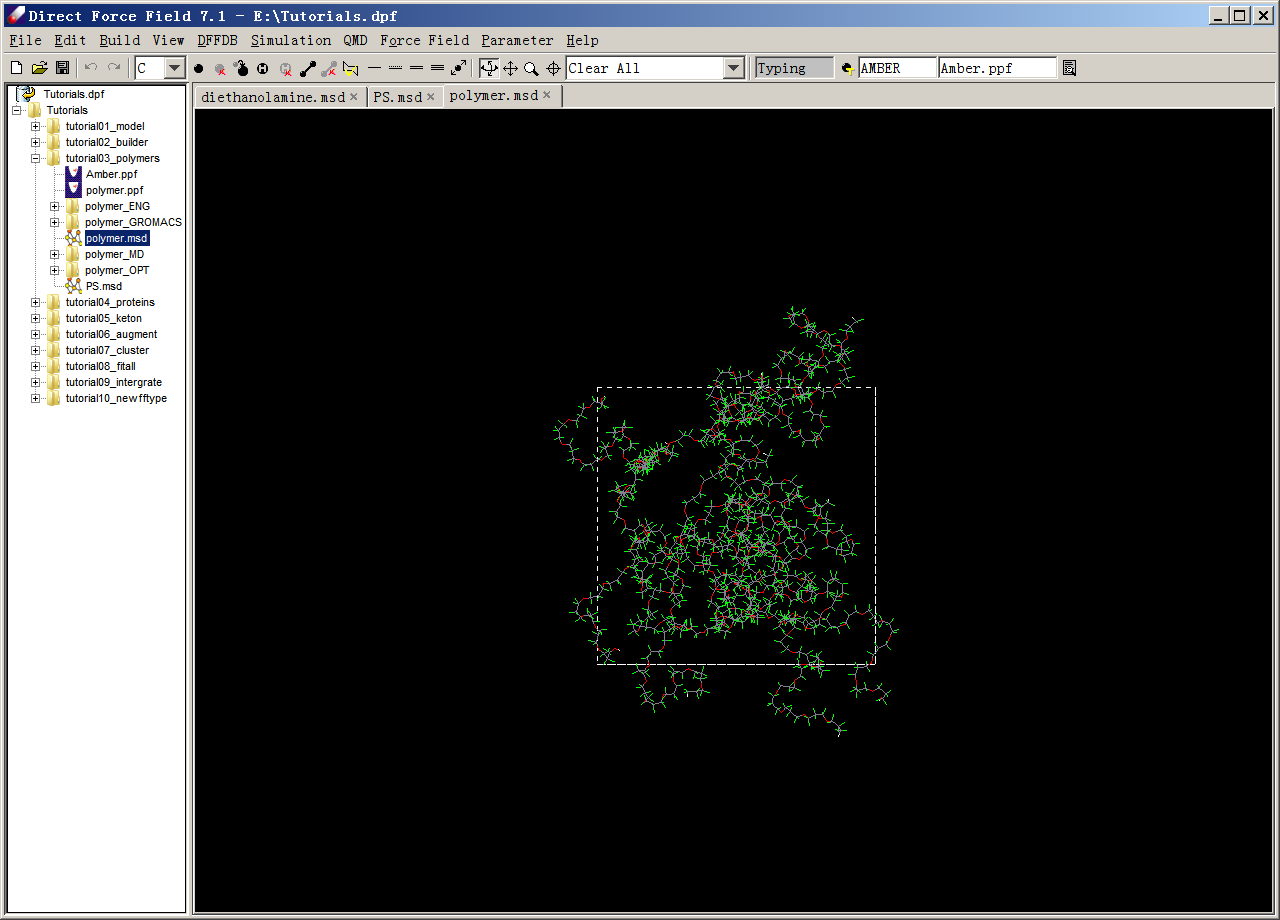
To uninstall DFF from Windows systems, run *Unistall.exe* which is in the ***<dff-root>*** directory. On Linux platform, delete the DFF installation directory.

### Starting

To start the software, double click on the DFF icon on Windows. On Linux, DFF can be started by running the shell script *dff8.sh*:

        $sh dff8.sh

The DFF user interface is divided into three areas. At the top are the *Command Menu* and *Tool Bar*, at the left is the *Project Navigator* which lists files and sub folders in current project, and in the center of the window are *Main Panels* for displaying molecular models.

.

**Main Panels**

**Project Navigator**

**Command Menu & Toolbar**

### Licensing

Two license modes, *local* or *floating*, can be used with DFF. The local mode allows usage of DFF on the same computer that is licensed. The floating mode allows usage on any computers (clients) that access the licensed computer (server). In both cases the license is node-locked to a specific computer. The license data, including products, computer, and number of concurrent usages, etc., are specified by a license file. The license file, License.txt, must be saved in <dff-root>/License/folder of the licensed computer.

To use the local mode is straightforward. Essentially a valid license file needs to be saved as <dff-root>/License/License.txt. To use the floating mode, the license server must be set up and then the client computers need to be configured to use the license server.

### Local License

There is an interface that can be used to configure license, either as a local mode or floating mode.

Install and start DFF. If the license is not installed, an error message will appear, close the error message, the "License Manager" dialog will appear:

This dialog can also be opened by using **Help→License** command from DFF user interface. There are four command buttons on this dialog.

**Request** is used to generate a license request file. Click this button and follow the instructions to generate a license request form; send this form to your DFF distributor to obtain a license file.

**Install** is used to install the license file. Click this button to allocate the license file and install it. The license file is copied to *<dff-root>/License/License.txt.* The license file must be issued for the same computer.

**Configure** is used to set up a floating license. The default values are "***localhost***" and "***null***" which means to use local license.

**Ok** to close this dialog.

### Floating License

**On the server side**

If DFF is installed on the server computer, the computer can be used as both client and server. Alternatively, only two files are required to run the license server:

* + - 1. The executable: <dff-root>/bin64x/dfflsv.exe (/bin64w on Widows)
      2. The license file: <dff-root>/License/License.txt

Note that the license file must be authorized to the server computer.

The simplest way to start the server is run the server program from command line: such as:

$dfflsv.exe start &

which puts the program running in the background. The server can run automatically on Windows or Linux. Refer to the system guidance for instructions.

The program dfflsv.exehas several options as explained below.

**On the client side**, follow the same instructions to install DFF and open the *License Manager*, then use the **Configure** command to set the IP address and port number of the licensed server. For example:

This assumes the license server has been set up on a computer identified by the IP address and accessible through the port numbered 3583.

Click **OK**, the data is saved in <dff-root>/License/License.ini file. There is no need for License.txt, which is used on the server.

### More Options of dfflsv.exe

This program can be used on server and client depending on the task from the command line.

On both client and server:

Dfflsv.exe help - to show the options as explained below

Dfflsv.exe request [file] - to make and save a license request file.

Dfflsv.exe install file - to install the license file.

Dfflsv.exe getmac file - to get and save the MAC address in a file.

Dfflsv.exe test [host [port]] - to test the connection to the server.

Dfflsv.exe status [host [port]] - to list the license information.

Dfflsv.exe reset [host [port]] - to clear the list of computers registered.

On server only:

Dfflsv.exe start [port] - to start a server with optional port (default 3583).

Dfflsv.exe stop [port] - to stop the server, the port must be the same as start.

### Install External Software

The user interface of DFF directly integrates with the quantum mechanics software suites *GAUSSIAN*.  This software package must be installed and configured on the same computer, and the environment variables must be properly set. Please refer to the documentation of Gaussian packages for more information. Once the software is installed and configured, it can be started directly from DFF interface.

The DFF installation package includes software packages of *GROMACS*[*1*](#_ENREF_1)and *LAMMPS*[*2*](#_ENREF_2), which can be called directly from DFF interface.

# CHAPTER 2 – TUTORIALS

The tutorials are divided into three parts according to common usages: 1) Essential functionality of deployment of TEAMFF forcefields; 2) Developing new forcefields; 3) Advanced topics for conventional forcefield developers; and 4) The new on-the-fly forcefield (OTF-FF) developments.

All user files are saved in a project folder, which is associated with a project file, The project file is named ***“.dpf”*** and created in the project folder.

For convenience, molecular models used in the tutorials are saved in a project folder named *Tutorials* under the *<dff root>*. For example, in Windows the default installation path for the Tutorials folder may be:

C:\Program Files\ATI\Direct Forcefield 8.0\Tutorials

It is recommended to copy the tutorials folder to a new location so that changes can be isolated from the original files. In this tutorial, we assume the exercise folder is:

E:\Tutorials\

*Note: You need written permission to the project file and folder. If you are using a different location, replace "E:\*Tutorials*\" with the path name you have chosen in the rest of this Tutorial* document*.*

## 2-1. Section A - Essential Functionality

In this section we learn some of the essential functions of DFF. We focus on how to create or import molecular models, in terms of small molecules, liquids, polymers, proteins, and mixtures. We also practice how to use TEAMFF forcefields to prepare input files for external software of molecular simulation.

The tutorials illustrate some concerts, algorithms and methods, which are further explained in [Chapter 3 - Concepts, Methods and Algorithms](#_CHAPTER_3_–). More information about available commands is explained in [Chapter 4 - Reference of Commands](#_CHAPTER_4_–).

### Tutorial A1 - Building a Model of Molecule

In this lesson, we will practice some of the basic operations of DFF. The following topics are covered:

* Creating a project
* Creating a model
* Assigning a forcefield
* Performing a simple calculation

#### Creating a New Project

We will create a new project called *MyWork*. In DFF, a project consists of a *project file* and a *project folder*, such as:

*E:\MyWork  
       E:\MyWork.dpf*

denoting a project “*MyWork*” in the *E* driver.

1. Start DFF and click File → New Project command*.* *A* dialog named "*Create Project*" will pop up. Select a drive, enter "MyWork" as the “*File Name”*, and click the Create Project button. A file of "MyWork.dpf" and a folder of "MyWork" will be created in working directory. In the *Project Navigator*, the root node named *“MyWork.dpf”* and a project node named as “*MyWork*” will appear.

2. Point to the root node in the *Project Navigator and* right-click the mouse, a pop-up options of “Larger Font” and “Smaller Font”, which can be used to set the size of font in the *Project Navigator*.

3. Hover over the project node and right-click. A pop-up menu with several commands: **New**, **Sort**, **Refresh** and **Open in Explore** appear. Select **New,** a sub-menu appears, select **Folder**. A *"New Folder"* dialog window will appear. Enter "***test*** " and click**.** A sub-folder will be created in the project folder and listed in the *Project Navigator*. Right-click on subfolder to add a new molecular model and name it as "**Aspirin**".

*Note: A model is a molecular system which can be one molecule or a set of molecules. Depending on the purpose, we may have multiple models for one of one molecular system, which represent different configurations or conformations.*

#### Creating a New Model

In this tutorial, we will make a new aspirin molecule as shown below and then estimate forcefield parameters for this molecule.

1. Double-click on the newly created empty model "Aspirin" in the *Project Navigator*.

2. Click the Add Atom button () in the toolbar.  Select “**C**” in the element pull-down menu (  ) on the toolbar.

3. Left-click on the main screen to add a carbon atom. Move the mouse away from the first atom and click again to make another atom.  The single bond between them is automatically added.

4. Repeat until a six-member ring of carbons is formed. To close the ring, click on the starting atom.

*Note: A single click on an existing atom toggles connectivity. A yellow dot indicates connectivity is on. When connectivity is on, any atom added will be bonded to the selected atom. If you make a mistake, use the undo or redo button  to alter the operations.*

5. Continue drawing until the entire skeleton (carbon atoms and connectivity) is sketched out. At this stage, don't worry about the hydrogen atoms, elements and bond orders.  You should have a structure that looks like this:

6. Select "**O**" in the pull-down element menu of the toolbar.  Click the replace-atom button (). Replace carbon with oxygen as needed by clicking on the atom to be replaced.  You should end up with the following structure:

7. Select the double-bond button (). Click on a bond to change the bond order as needed.  When you are done with this step, the model should look like this:

8. Click the add hydrogen button () to add hydrogen atoms automatically.  Hydrogen atoms are added to match the available and existing valence numbers for each atom. Make sure the bond orders are correctly set before clicking the **add hydrogen** button.

9. Clean the structure by clicking the refine structure button ().

10. The coordinates are adjusted from the points clicked on the screen. For convenience in viewing and operation, recenter the molecule by clicking the **Edit→ Reset COM** command. The molecule will be put in the center of the screen. Select the rotate button in the command bar panel , and then hold the right mouse-button and drag to see how the model can be viewed from different angles. Other commands are translation, zoom in/out and recenter (not changing COM).

11. Click on the Save button to save the model. DFF uses the “.msd” (Model Structure Data) extension to identify model files.

#### Mouse and Short-cut Key Operations

This would be a good time to practice mouse and keyboard editing functions.

1. Click the selection (arrow) button in the command bar panel (  ) to activate the mouse selection function. Once it is activated, click on an atom will pick up the atom. You can also press the left button and drag the mouse to select multiple atoms. You can cut, copy, and paste the selected atoms, to the same model but different location (where the mouse point is) and to a different model.
2. The common short-cut keys are supported, the combination keys Ctrl + C, Ctrl +X and Ctrl + V are used for cutting, copy and paste respectively. In addition, Ctrl + V and Ctrl + Y are used for undo and redo, which are the same as the undo and redo buttons in the toolbar.

#### Save, Close and Open Project

At any time, you can save the project by using **File → Save Project** command, you can also save the project into different name by using **File → Save Project As** command. To close the project, use **File → Close Project** command. DFF will prompt you to save before close. The saved project can be reloaded by using **File → Open Project** command.

*Note: The same model is provided in the Tutorial project. In the following section, we will use the Tutorial project, but the same operations can be applied to the project you have just created.*

#### Assign Parameters using TEAMFF

1. Open the *Tutorial* project using **File → Open Project** command. A list of subfolders is shown in the *Project Navigator*, click on the first one, **“tutorialA1\_edit*”***, which has the *Aspirin.msd* model listed. Click on this node to have the molecule shown on the main screen.Click the **TEAMFF → Admin** command. Browse and select TEAMFF database at *“***<DFF installation>/database/TEAMFF.TEAMFF**”. This will open a TEAMFF Admin dialog. Set *"Forcefield type"* to “**AMBER**”, all forcefield tables in *AMBER* type are listed. Select *“***AMBER-General**” *forcefield and* click **OK**.

2. Make sure the model “**Aspirin.msd***”* is selected in the *Project Navigator and* click **TEAM-FF → Assign** command. This will open a dialog listing the selected forcefield table in *“Input”*. The *“Output”* is the extracted forcefield (.ppf) parameter file. Leave other options unchanged, click **OK** to launch the job.

3. When the job is done, a forcefield table appears. The molecule will have also been assigned atom types (C1, C2, ...) and charges. To display atomic properties such as atomic types, charges, isotope, mass, formal charges, etc., select corresponding item in the pulldown command of atomic properties on the toolbar  . The *“Apparent Atom Types”* assigned to the model must be consistent with the atom types listed in the forcefield ppf file.

*Note that sequential atom types are used as the apparent atom types. The actual atom types used in the database are made by the typing rules associated with the forcefield table. The conversion is necessary for mixing multiple forcefields, although in this tutorial it is not essential.*

#### Test the Assigned forcefield

Now that forcefield parameters have been prepared and the atom types and charges have been assigned, the model is ready for simulation.

1. Double-click on *“Aspirin”* in the *Project Navigator*; make sure it has a forcefield assigned as indicated by the *"Forcefield"* option in toolbar.

2. Click Simulation **→** Optimization command to open the *"Optimization"* dialog. Examine the job name, number of steps and convergence criterion (energy in kcal/mol). Select the **“Variable matrix (BFGS)”** method and the **“Screen view”** option, leave other options unchanged, and click Execute to start the job. The structure is automatically updated as the job runs.

3. Similarly, you can perform a molecular dynamics simulation using **Simulation →** Molecular Dynamics command.

### Tutorial A2 - Working With a Liquid Model

In this lesson, we will build a liquid model and use TEAMFF to launch simulation jobs.

#### Loading Models and Assigning Forcefield

1. Start DFF*.* If the project is not fully loaded, right-click on the “*Tutorials”* node in *Project Navigator*, right-click mouse to select **Refresh** to update the list. The command **Sort** can be used to arrange the subfolders in the order as specified.

2. In the *“Tutorials”*folder, there is a “*tutorialA2\_liquid”*subfolder. There are four molecular models built in this subfolder: “*water”, “phenol”, “octane”,* and*“SOS” (Sodium octyl sulfate)*. We will use these models in this lesson.

3. Select all four models. Click **TEAMFF → Admin** to ensure the database *“*TEAMFF*.TEAMFF”* is selected. Set "*Forcefield type*" to “**TEAM”**, select **“TEAM-General*”*** forcefield from the list, and click **OK** to close the dialog.

4. Select **TEAMFF → Assign** command to open a dialog. Rename the *"Output"* forcefield to **“team.ppf”**. Four molecular models are listed under “*Available Models”*. Click **OK** to assign forcefield parameters for these molecules.

5. When the job is complete, a forcefield table appears. Examine the parameters and close the forcefield table. Note that each of the models is associated with the newly assigned forcefield “*team.ppf*”, which can be verified by clicking on the model in the *Project Navigator* to open the *Forcefield Editor*.

6. Select a model in the *Project Navigator* and then select the **Build → Charge Group** command to set the charge groups.  Leave the default option unchanged, click **Execute** to assign charge groups. The central panel of the dialog shows the charge group properties. Repeat this operation for each of models listed in the *Project Navigator*. Water contains one charge group, phenol has 6, octane has 10, and surfactant has 8, among these groups two are charged species: sodium cation and SO4- anion.

#### Building a Liquid Mixture Model

1. Let's make a mixture of water and phenol. Select water and phenol from the *Project Navigator*, and then open **Build → Bulk Liquid** command.  This brings up a dialog to build a liquid model. select **“300”** water molecules and **“100”** phenol molecules, set density to be **“0.8”**, and click **Execute**.

DFF *will now build a box with low density and then compress the box to the target density. The compression is a series of quick NPT simulation without sufficient relaxation. For bulky molecules, this could cause a very high energy configuration in the end. To avoid such problem, try to make an initial configuration with low density and gradually compress the box by using normal NPT MD simulations with high pressure.*

2. Make sure this model is associated with the forcefield assigned earlier. If not, use **Forcefield → Associate** command to do so. Alternatively, you can select the model from the *Project Navigator* and right-click mouse button to associate the forcefield.

This model is now ready for simulation. Open **Simulation → Molecular Dynamics** command. Since the model is not optimized, let's relax the configuration by setting a small-time step of 0.2 fs and run a short period of time, for example:

The most expensive calculation in atomistic simulation is about the nonbond energy. With periodic boundary condition, two options of cutoff, group-based and atom-based, can be selected. This can be done by clicking the **Nonbond Energy** button. The group-based cutoff is recommended.

*If the energy of initial configuration is too high, MD simulation will stop and return an error message. Reduce the time step to smaller value (such as 0.1 - 0.001 fs) can quickly reduce the tensions in the system, then continue the simulation with normal time step.*

3. Click **Execute** to launch a MD simulation. The screen will update as the simulation progresses. Set *“Label”* to **“Total Energy”** to see how the energy changes. When the job is done, a new liquid box is saved in a subfolder.

#### Building Liquid-Vapor Interface

DFF can be used to build liquid-vapor interfaces, which are useful for forcefield parameterization and validation because the experimental data of vapor-liquid-equilibrium and surface tension are often available. We will build a water liquid-vapor interface first.

1. Open the *"Bilayer Interface"* dialog by clicking on **Build → Liquid** Interface. Enter a *“Job Name”* if desired, and then make sure the box dimensions in X and Y are acceptable, make the values larger or smaller will change the box sizes.
2. The *“Phase A”* is the central part of the model to be built, which has two surfaces. We will treat it as water bulk. Select **“water”** from the pulldown, enter **“1.0”** as the *“Density”*, and **“500”** as the *“Count”* of molecules. Note the count will change the Z-length, make sure the value is reasonable.
3. The “Phase B” defines the two phases above and below the central part. Let’s treat it as water vapor. Set *“Density”* to be **“0.1”**, “Count” to be **“50”**, so that the Z-length is like that of the bulk liquid.
4. Do not select anything for the *“Interface”*, and then click **OK** button to start the builder job.
5. When the job is finished, a bilayer model is built. Turn it around you can see the model looks like this:

A picture containing text

Description automatically generated

1. Using **Simulation → Molecular Dynamics** to relax the system, and then using **Simulation → LAMMPS** to submit a simulation.

*The vapor-liquid interface model can be used to calculate vapor-liquid equilibrium directly by using MD simulations. It is generally an expensive simulation because of slow convergence, especially at low temperatures when the number of molecules in the vapor phase is low.*

#### Building Liquid-Surfactant-Liquid Interface

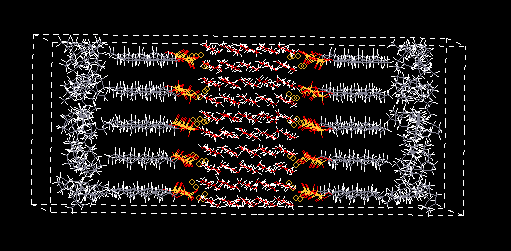
DFF can be used to build liquid-liquid interfaces with or without the third molecule in the interface. We will use water, octane, and SOS to build an interface with surfactant.

1. Follow the same procedure as above, set **“water”** in Phase A, and **“octant”** in Phase B. The density of water is **1.0**, the density of octant is **0.7**. Control the *“Count”* so that the Z-lengths are reasonable.
2. Select **“SOS”** as the interface. Enter “25” as the number of molecules, which defines the surface concentration. The Dialog looks like this:

Graphical user interface, application

Description automatically generated

1. Click OK to launch a series of DFF background jobs. When the jobs are finished, a new model is loaded automatically. Rotate the model, it should look like:



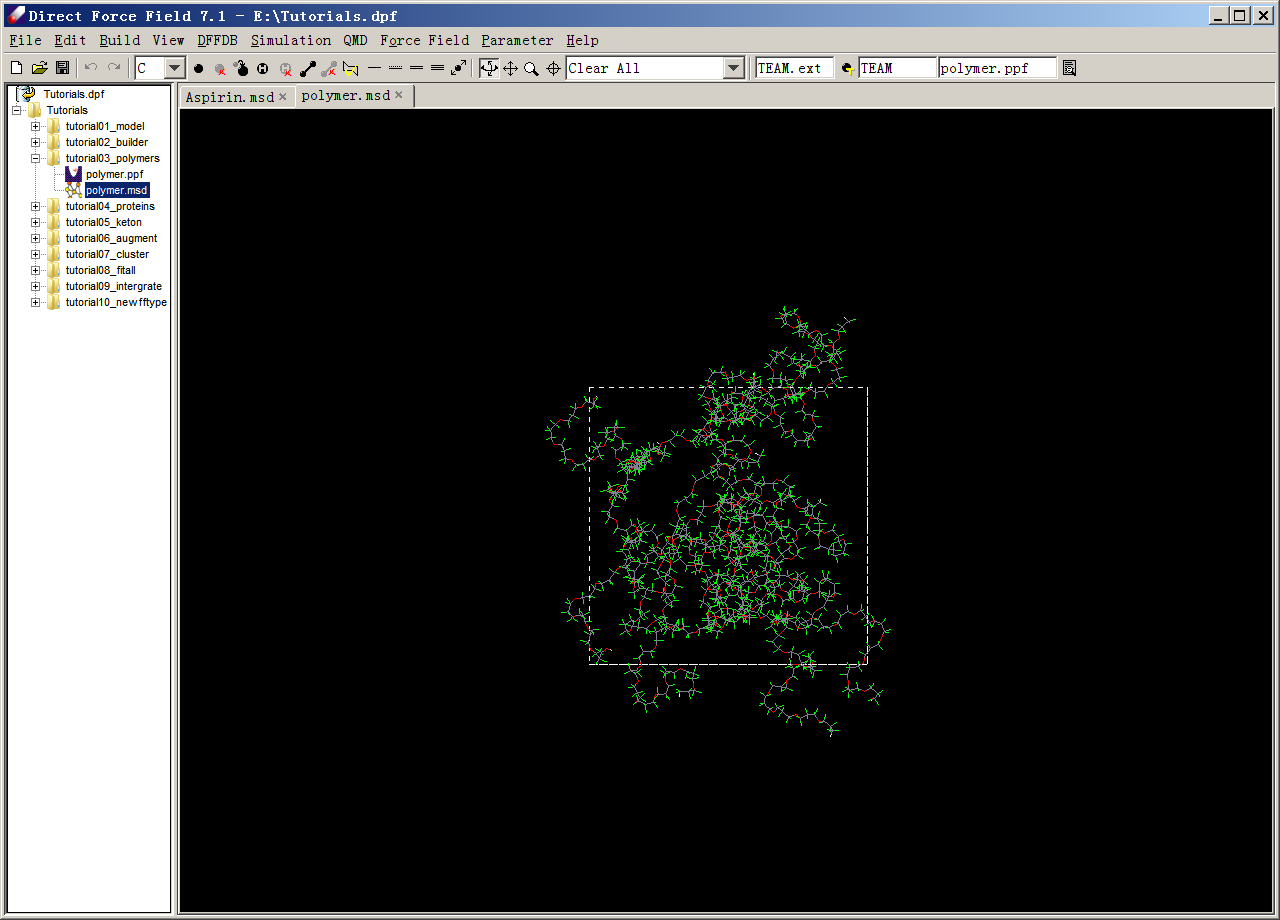
### Tutorial A3 - Working with Polymers

In this lesson, we will work with a polymer model.

1. Start DFF and use **File → Recent Project** to open the *tutorials* project. Navigate to the project root folder, right-click, and select ***Refresh.*** This will reload all files in this project.

#### Assigning a Forcefield

2. In the subfolder ***tutorialA3\_Polymers***, there is a polymer mode that has been prebuilt and converted to the *MSD* format. Double-click on this model to load the structure:



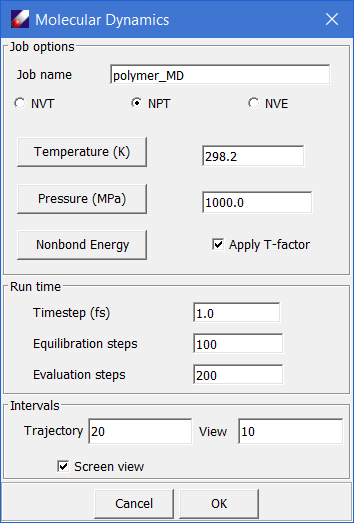
*Note that this model has a low density (0.4587), which can be displayed by clicking* ***“Property”*** *in the toolbar. We will compress this model to a density closer to the experimental value.*

3. Open **TEAMFF → Assign** command. This dialog lists the database and forcefield selected (TEAM-General). The *"Output"* field lists the file name to be used for the output forcefield. Unselect Click **OK** to assign forcefield parameters. When the job is done, a forcefield table appears. Review and close the forcefield table.

4. Select this model, open **Build → Charge Group** to assign charge groups, and click **Go** to assign the charge groups automatically. After the job is completed, the central section shows 505 charge groups (scroll the display area to the top to view). The largest group contains 7 atoms, the smallest group contains 3 atoms, and all groups have a total charge of zero charge.

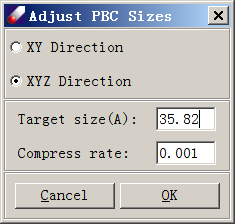
#### Preparing the Model

5. We will now compress this model to a higher density state. Select the model and open **Simulation → Molecular Dynamics**. Select **“*NPT”***, set “*Pressure”* to be **“*1000.0”*** MPa (very high for a rapid compression), set the “*Equilibration steps”* to be ***“100”*** and “*Evaluation steps”*to be “**200**”. Make sure “*Screen View”* is “**On**” and click ***OK***.



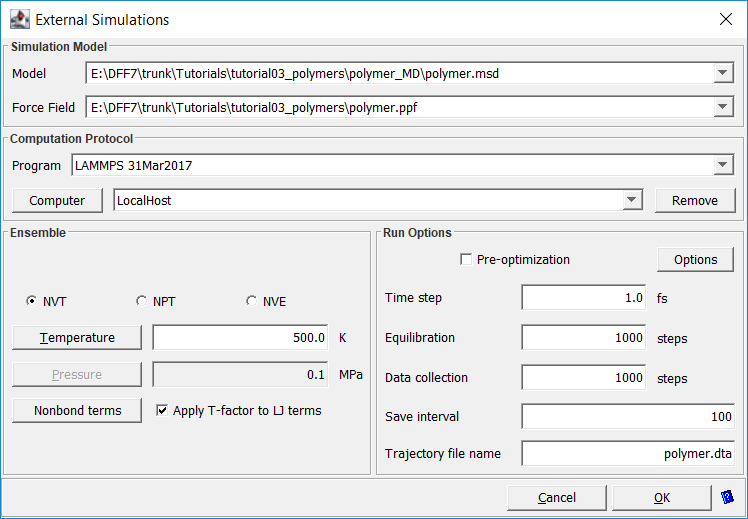
6. DFFnow displays density changes during compression. Step 5 can be repeated as needed. Every time a new MD job starts, a subfolder is created for that job. When the system approaches the targeted density (1.5), stop the job. The instantaneous configuration should approximate but not match the target density. To make an exact density of 1.5, we would need to reset the cell edge parameters. The required edge size X1 can be calculated from current edge size X0, current density D0 and target density D1 using the following equation:

For this system, the target edge size should be 35.82 for a density of 1.5. Select the model, open **Edit → Adjust PBC**, enter the cell edge size, and click **OK**.



#### Running LAMMPS

7. The model is now ready for simulation. Repeat a NVT simulation in DFF to relax the simulation box and verify that the density is indeed correct, and then submit the MD simulation to *LAMMPS* by selecting the prepared model and clicking **Simulation → LAMMPS**.



The model and forcefield files are loaded automatically. Note that the option of “*Apply T-factor to LJ terms*” is selected, a new forcefield in which the LJ parameters are scaled according to the applied temperature will be made and used for this job. Select **“NVT”**, set *“Steps”* to “**1,000**” in order to see the results quickly, and note that the “*Trajectory file name*” is ***“*polymer.dta*”***. Click **OK**.

8. When the job is finished, a subfolder named as “*polymer\_LAMMPS”* will be created in the *Project Navigator*. This folder contains the input and output files of this LAMMPS job. Examine the input and output files. Select “*polymer.dta”* and “*polymer.msd”* files from the *Project Navigator*, then click **Simulation → Trajectory Viewer** to bring up the following dialog box:

Graphical user interface, text, application

Description automatically generated

Click **OK** to start a new window that replays the trajectory. The trajectory is played in a new window, you can use the same viewing options of DFF to translate, rotate, and zoom the models on the trajectory window.

### Tutorial A4 - Working with Proteins

Models built using external software may need to be modified before assigning a forcefield. In this lesson, we will work on one example. In *“tutorialsA4\_Proteins”*, a *MOL2* file is provided. The model does not have hydrogen atoms, so we will assign hydrogen atoms first.

#### Import Models and Assign forcefield

1. Start DFF and open the *“***tutorials***”* project. Right-click on *“***tutorialA4\_Protein***”* and select **Add → Models** command. In the *"Open"* dialog, set *"Files of type"* to **“MOL2 Files (\*.mol2)”** and select “**protein1.mol2***”*. Click **Open** to load both models.

*2.* In the Project Tree, click the newly added model to view it in the main screen. The model does not have hydrogen atoms. Click **Add Hydrogen** button ( ) on the tool bar or use **Edit → Add Hydrogen** to add hydrogen atoms automatically.

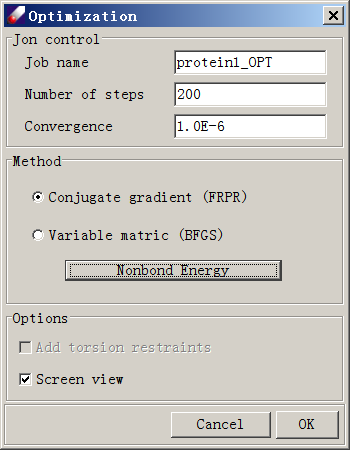
3. Click **TEAMFF → Admin** command. Select the “**AMBER”** forcefield type and the “**AMBER-General”** forcefield. Select protein1 and click **TEAMFF *→* Assign**. Note that the selected forcefield, model (protein1), and output forcefield are listed. Click **OK** to assign forcefield parameters. When the job is complete, a forcefield table appears indicating the forcefield type is AMBER and the typing rule is “Composed”, which means atom types have been successfully assigned.

#### Prepare the Model

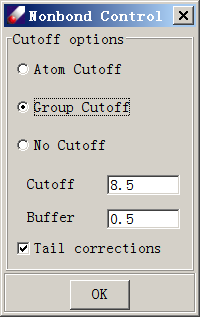
5. Select **“Protein1.msd”** from the *Project Navigator* and click **Build *→* Charge Group**. Set “*Allow charge crossing a bond*” parameter to **“0.1”** and click **Execute**. The text area shows the results indicating the largest group has 7 atoms.

*To define small charge group is essential. The groups do not have to be charge neutral, but the size may be an issue. For example, if the group is too large, GROMACS may fails.*

6. Because hydrogen atoms were automatically added, the structure is not relaxed. If the configuration is submitted to *GROMACS* or LAMMPS, the simulation may fail. Therefore, we should relax this configuration first. To do so, select **“Protein1.msd”**and open **Simulation *→* Optimization**:



Click on **“Nonbond Energy”**, which opens the following dialog:



Select **“Group Cutoff”** and leave other options unchanged. Click **OK** to close this dialog. Click **OK** again in the “Optimization” dialog and watch as the energy value drops. When the job is done, the optimized model will be saved to a subfolder named “*protein1\_OPT*”.

#### Run *GROMACS*

7. Now that the optimization job is done, we are ready to submit a job to *GROMACS*. Click **“protein1.msd”** from the “*protein1\_OPT”* folder. Make sure the forcefield is still associated and select **Simulation *→* GROMACS** command.

8. Using the default options, set the simulation steps to small numbers (e.g. 1000 steps), and click **OK** to start the job. The trajectory file will be saved as “*protein1.dta”*.

9. When the job is complete, a notice will pop up and a subfolder named “*protein1\_GROMACS”* will be placed under the folder in which the job is started. Double-click on the files in this subfolder to see the results of the simulation. Open the **“protein1\_GROMACS”** folder, right-click select **Import**, set the file type to **“GROMACS Trajectory File”**, and select **“protein1.dta”** to load the trajectory file into protein1.dta.

10. Select “**protein1.msd”** and **protein1.dta**, then click **Simulation → Trajectory Viewer** to watch the trajectory.

### Tutorial A5 - Mixing Two Forcefields

In this tutorial, we will practice how to use two forcefields in one simulation. This example demonstrates one of the unique features of TEAMFF. Because each forcefield is independently developed, the forcefields can be parameterized accurately. Upon use, different forcefields can be applied to different molecules depending on which provides a better coverage. Interactions between the different molecules are described by nonbond energy terms only, for which the regular combination rules are applied.

In this lesson, we will use a model consisting of zeolite (MOR) interacting with two xylene molecules. For convenience, a sample file “MOR\_2Xylene.msd” is provided.

1. Start DFF and open the project **"Tutorials.dpf"** if it has been closed. Open “**tutorials/tutorialA5\_Zeolite”** folder, a unit cell containing two xylene molecules caged in zeolite is listed. Open the model.

*The model contains two different kinds of materials, zeolite, and organic molecules, which are described using different forcefields in TEAMFF. There are two ways to use these forcefields correctly, the simplest way is set the order of forcefield in TEAMFF so that the forcefields are used in the order as specified. Another is to define subsets and use different forcefields for different subsets.*

*We will first use the order method, and then the subset method which gives us more specifically control.*

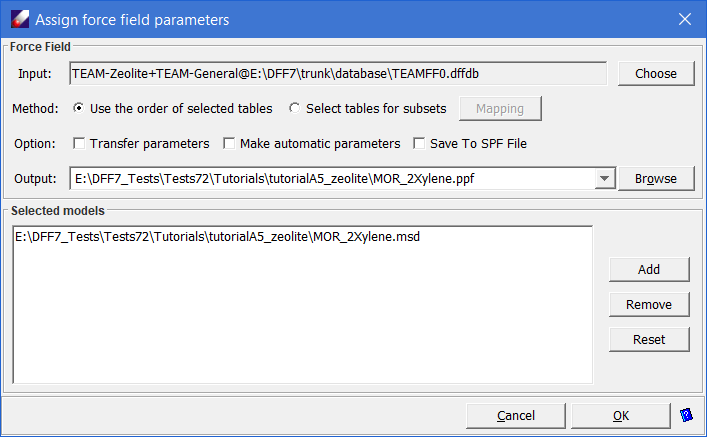
#### Using The Order OF Tables

2. Click **TEAMFF → Admin**, load **“**TEAMFF**.TEAMFF”**, select **TEAM** as the forcefield type, and select two forcefields, **“TEAM-General”** and **“TEAM-Zeolite”**, which is indicated by the check mark in the first column. Since we will use the order to get parameters, make sure the more specific forcefield, TEAM-Zeolite, is on top of TEAM-General. If the order is different, select a line and use the buttons **Up** and **Down** to adjust its position.

Graphical user interface, application

Description automatically generated

3. With the forcefields are selected, click **TEAMFF → Assign**. Make sure the “Use the order of selected tables” is selected, leave other options as given, and then click **OK** to launch the job. A new forcefield will be generated and associated with the model.



4. To verify the forcefield is set properly. Use **Simulation → LAMMPS** command to launch a short MD simulation. When the job is done, using **Simulation → Trajectory** to see the simulation trajectory or examining *LAMMPS* output file to verify the simulation is carried out successfully.

*Note: DFF does not support simulations of crystals.*

#### Using Subsets

*To use subsets to assign forcefield for different part of model, we first set the subsets.*

1. Click **Build → Subsets**. In the pop-up dialog, select the “**automatic – to set subsets using connectivity**” option and click **Execute**. This will create two subsets based on connectivity – one for the zeolite and another for the xylene molecules.

Graphical user interface, text, application

Description automatically generated

2. Click **Rename** to rename the subsets as shown below. Click **OK** to close both dialogs. Click **“Save”** command to save the model.

Graphical user interface, text, application

Description automatically generated

3. Click **TEAMFF → Admin**, load **“**TEAMFF**.TEAMFF”**, select **TEAM** as the forcefield type, and select two forcefields, **“TEAM-General”** and **“TEAM-Zeolite”**. Close the dialog.

*Since we will use subsets to assign different forcefield, the order does not matter in this case.*

4. Click **TEAMFF → Assign**. In the “Assign forcefield parameters” dialog, select “**Select table for subset**” and click **Mapping** to open another dialog to choose forcefield table for each subset as shown below:

Graphical user interface, text, application

Description automatically generated

Click **OK** to close the dialogs. Leave other options in “Assign forcefield parameters” dialog unchanged, and then click **Ok** to begin the assignment process. When the job is completed, a forcefield will be loaded showing parameters taken from two tables.

5. To verify the forcefield has been set properly. Use **Simulation → LAMMPS** command to launch a short MD simulation. When the job is done, using **Simulation → Trajectory** to see the simulation trajectory or examining *LAMMPS* output file to verify the simulation is carried out successfully.

## 2-2. Section B - Parameterization

In this section we focus on the central functionality of DFF*,* making forcefield parameters for molecules. The tutorials cover topics on how to prepare quantum mechanics (QM) data and how to fit parameters using the QM data. We also illustrate how to optimize the van der Waals parameters by using molecular dynamics simulation data of liquid.

### Tutorial B1 - Fitting Parameters for a Small Molecule

In this tutorial, we will learn how to use DFF to fit forcefield parameters from quantum mechanics data (QMD) for a simple molecule. The molecule is dimethyl ketone in which the four heavy atoms (3 carbons and 1 oxygen) form a rigid plane with two methyl groups and are rotatable. To fit this kind of molecule is not difficult. We will learn how to make a forcefield that not only fits minimum energy structure, but also reproduces the conformational energy profile calculated using QM methods.

A general procedure is to obtain the QMD for an energy minimum structure as well as conformational isomers. The QMD includes energy, energy first and second order derivatives. We then fit QMD to derive valence and charge parameters.

For convenience, files used in this tutorial have been pre-calculated and provided in the **TutorialB1\_fitKetone** folder of the tutorials project.

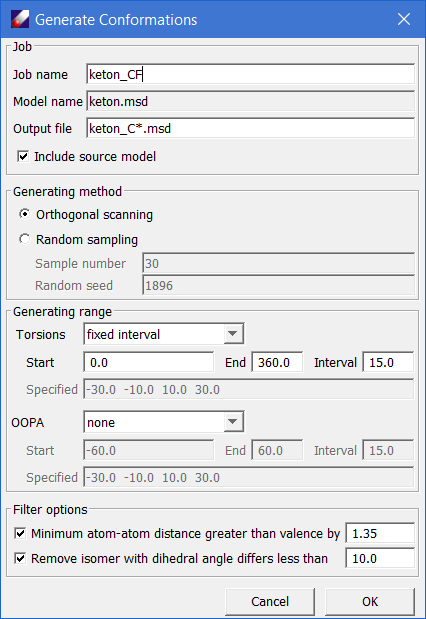
#### Preparing the Quantum Mechanics Data

1. Start DFF and open project *"Tutorials"* if it is not yet. Open **“TutorialB1\_fitKetone”** folder. Double-click on the **“ketone”** model and click QMD ***→*** Compute command. The name of the selected model is listed in the dialog. Set the **Computation** to "OPT FREQ" to optimize the structure and get the Hessians in one step. The computation options as follows:

Method: B3LYP   
Basic Set: 6-31G\*  
Total Charge: 0  
Multiplicity: 1  
CPU time limit: 300 (default)   
Memory Limit: 800 (default)

If you have GAUSSIAN software installed locally, select "Run background job" to run the calculations from *DFF*. When the job is done, there will be a notice prompting the next step. To run the job outside of *DFF* environment, select “Make input files only”. Click **“OK”** to continue.

1. When the Gaussian job is done, locate the output file (*keton.gauout*) in *Project Navigator*, right-click and select **“Create QMD”** to make a QMD (*keton.qmd*) file. A check mark of the QMD file icon () indicates that this QMD data contains optimized structure with the first and second derivatives calculated. Open the QMD file will show the same information.
2. Point to the QMD file, right-click to select “**Create MSD**” command, the optimized structure is saved in a new MSD (*keton.msd*) file. This is the optimized structure, will be used to generate conformers. Select the MSD file from the *Project Navigator,* click Build **→** Conformational Isomers, which brings up the "Generate Conformations" dialog box.

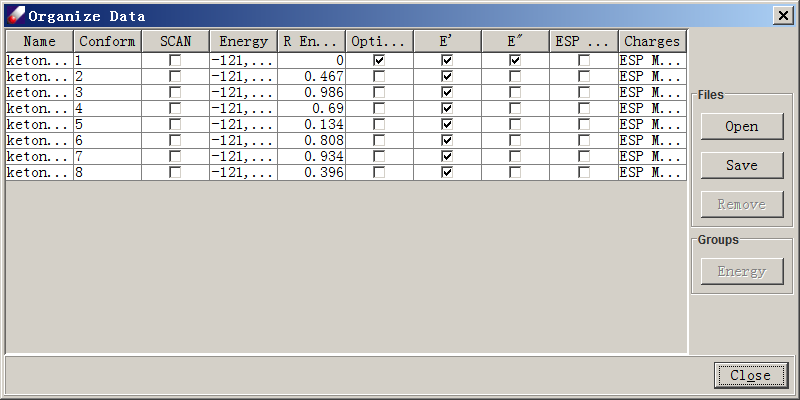


*This is a dialog that launches the automatic conformation generator (ACG). The software selects all dihedral angles defined by heavy atoms in the molecule, and then starts from the original structure, which should be the one of minimum energy, then scan each of the dihedral angles either systematically or randomly.*

1. Using the default options, click **OK** should generate 8 isomers in a sub-folder. Select all the conformers generated and click **QMD →** Compute. The selected conformers should be listed in the *"Models to Be Calculated"* text field. Set the Computation options to "***GRADIENT***", keep other options unchanged, and click **OK** to run.

*Since the isomers are not energy minimized, it is not helpful to calculate the second derivatives, but the energy and gradients are useful for fitting forcefield parameters.*

1. When the jobs are done, double click on the QMD file obtained above (*keton.qmd*) to open the QMD spreadsheet. Click the **Add** button to locate and load the newly obtained Gaussian output files. The spreadsheet will show entries loaded. Select all entries, and click Energy button to calculate the relative energies, some entries may have the same relative energy, which indicates that some structures (e.g. optimized) are calculated twice. You may select and delete the redundant isomer that has no second derivate. The final QMD list should look like the following figure. It shows a high-level summary of the data: name, conformation (if calculated), is it scanned systematically, the total energy, the relative energy, is it optimized, does it contain first and second energy derivatives, and electrostatic potential and atomic charges.

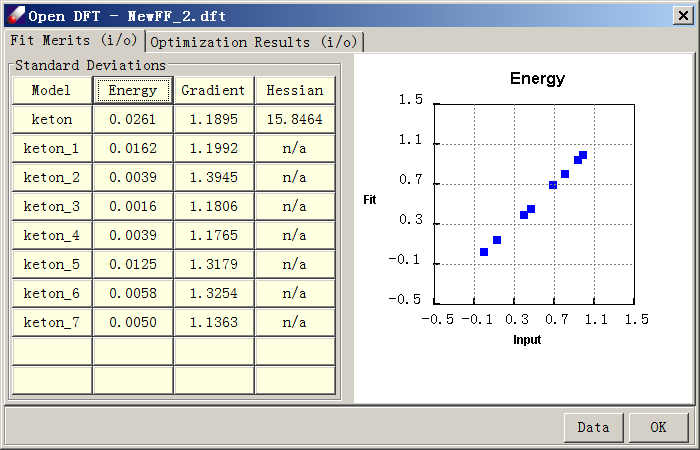


#### Fitting and Analyzing

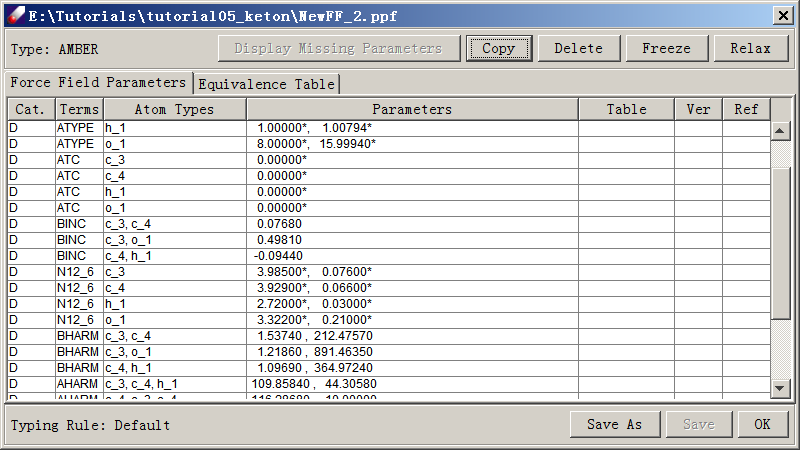
6. Refresh the “*tutorialB1\_ketone”* folder and select the prepared QMD file. Create a new forcefield using **Forcefield → New Forcefield**. Enter "***keton***" as the new forcefield name and select **“AMBER”** as the forcefield type. Select "**Default**" typing rules. Make sure ***"*Generate EQT table*"*** is checked. Click **OK**. Examine ***keton.ppf*** tables. Note that atom types are added into the parameter table and a default equivalence table is created.

7. Make sure **"ketone.qmd"** is selected, click **Parameter → Fit Charges** command. Select “**ESP”**as *"Data Source"*, make sure the **“Bond type based”** is checked, and click **Execute**. The charge parameters are calculated and assigned to the model.

9. Select **Parameter → Fit Energy Data** to fit valence parameters. The QMD model should be listed in the "Data to fit" box. Leave everything as default and click **OK**. When the job is finished, several files will be loaded into the *Project Navigator*. These files are the input (.dfi), output (.dfo), result forcefield (.ppf) and fit result table (.dft) files. The result file shows fit merits and validation results, which are structural and vibrational frequency data calculated by the forcefield using comparisons to QM data.



The parameters can be viewed by double-clicking on the *PPF* file. This dialog shows two panels, “Forcefield Parameters” and Equivalence Table. At the upper left corner, the forcefield type is indicated to be AMBER. At the lower left corner, the typing rule for this forcefield is shown as “Default”. This information is also displayed in the toolbar of the DFF interface.

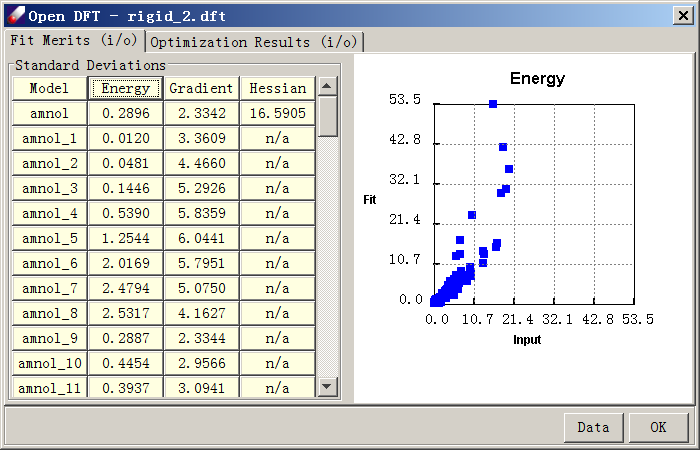


### Tutorial B2 - Fitting Parameters Of A FLexible Molecule

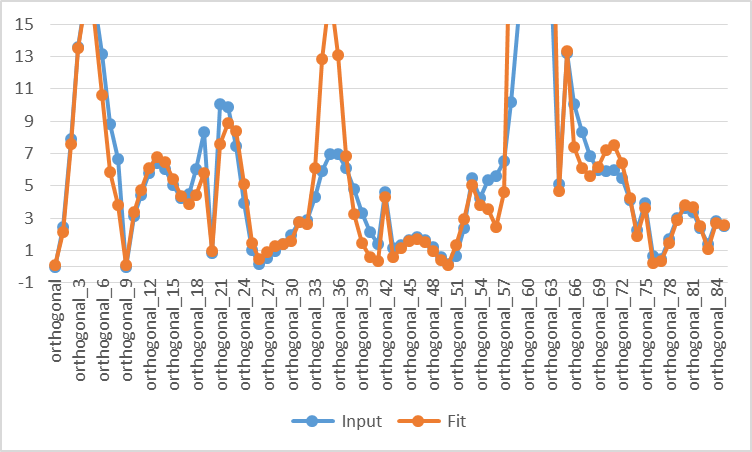
In this lesson we will learn how to fit a forcefield of 2-methylamino-ethanol (C3H9NO), a molecule with several rotatable bonds. Fitting flexible molecules is more difficult than fitting a rigid molecule. We will compare two sampling strategies and demonstrate the impact of different samplings on the quality of developed forcefield. The first is rigid sampling, in which all internal coordinates are fixed at selected values, as we used in the previous lesson. The second sampling is flexible; we sample the minimized energy curves along selected internal coordinates. For convenience, pre-calculated QMD files are included in the “tutorialB2\_fitAmnol” folder.

#### Rigid Sampling

1. Start DFF and open the *"Tutorials"* project. Open the **“TutorialB2\_fitAmnol”** folder. Double-click on the ***C3H9NO.msd*** model in *Project Navigator*. Click **Build → Conformational Isomers** to open a dialog of sampling. The amino group is not planar, therefore, we shall sample the out-off-plane (OOP) angle as well. Select **“Fixed interval”** for “OOPA”. Leave all other parameters as default, then click on **OK** to launch the job. A total of 85 conformational isomers will be generated.
2. Run single point energy and gradient calculations for these structures using the **B3LYP/6-31G\*** protocol. When the calculations are done, combine all data including the fully optimized structure with Hessian calculated (86 data in total) into one QMD file, which is named **“rigid.qmd”** in this Tutorial. Open **“rigid.qmd”**, select all entries and click on the **“Energy”** button to see the relative energy. The highest value is about 20 kcal/mol.
3. Fit the QMD file using the procedure explained in the last tutorial. Since some of the conformers are in high conformational energy, select “**Apply Boltzmann Factor”** in the **Job & Data** panel.
4. When the job is done, the fit quality is not very good. However, if we focus on the low energy range from 0 to 10 kcal/mol, the agreement between QM and FF is not bad.



1. To see the deviations in detail, click “**Data”** to export to a csv file and graph the data below 15 kcal/mol using software such as Microsoft Excel:



*As seen in this chart, the agreement at low energy region is not perfect, but acceptable. As the example above demonstrates, the rigid sampling is fast, but does not yield very accurate torsional energy curves.*

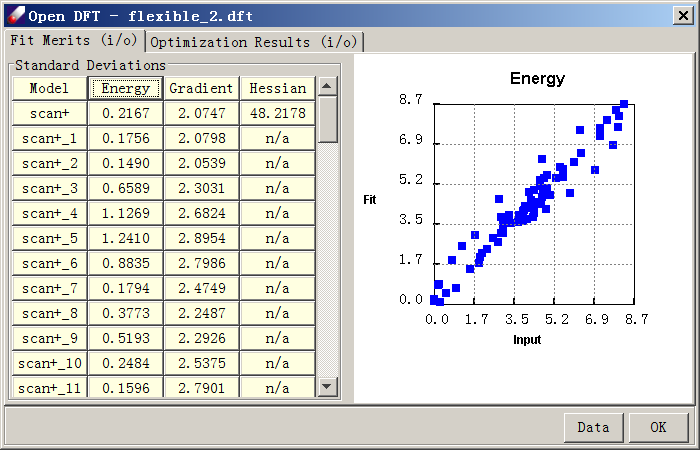
#### Relaxed Samplings

Alternatively, DFF provides a functionality that samples internal rotation energy curves using constrained minimization. This is done by conducting the *Scan* calculations using *Gaussian*.

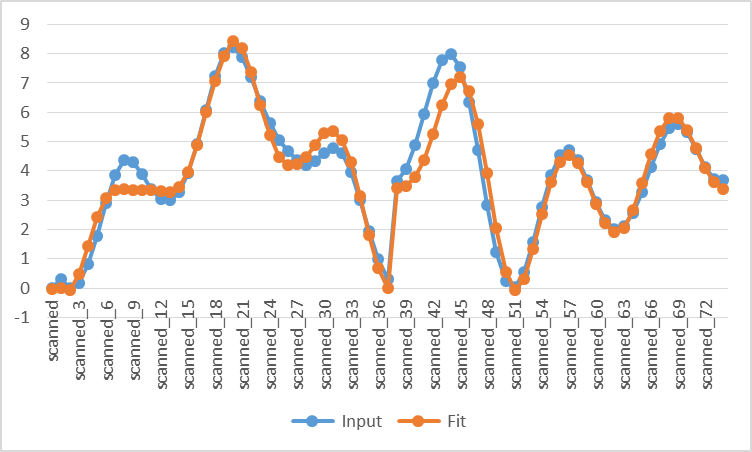
1. Click **QMD → Compute** to optimize and calculate Hessians of the molecule using the DFT B3LYP/6-31G\* method. Import the result into a QMD file. Make sure a MSD file is obtained from the optimized structure, which will be used to as the starting point to sample the conformational space.
2. Open the MSD file, click **QM → Compute** to bring up the QMD dialog. Select the option of “Dihedral constraints”, a text field is activated. Click **Add** constraints button to open a “Torsion” dialog for selecting the dihedral angles. The molecules with atom names are displayed on the screen. Select four atoms that defines a dihedral angle **O14-C1-C2-N13**, as shown in the “Torsion” dialog, the angle is “63.845”. Enter “**10**” as the sampling increment, “**36**” as the scan times, which means that we will completely scan 360 degrees for the dihedral angle. Click **OK** to close the “Torsion” dialog. The Dihedral constraints in terms of atom indexes are listed such as “D 14 1 2 13 S 36 10.0”. Make sure the Computation type is “OPTIMIZE” then click **OK** to submit the Gaussian job, which will generate 37 data points. If the option of “Run background job” is selected, it is convenient to select “Reload” and “Append data to a QMD file”.
3. Repeat the above calculation for another dihedral angle, **C1-C2-N13-C9**, which will generate another 37 data points.

*In the above procedure, one dihedral angle of selection is fixed while all other internal coordinates are relaxed. The sampling is incomplete but hopefully sufficient for deriving the forcefield parameters. Alternatively, we may completely scan the two-dimensional space by adding the second constraint before submitting the Gaussian job. If we use 10 degrees as the sampling increment, it will require (37x37) energy minimizations.*

1. When the QM jobs are done, import all Gaussian output files to the QMD file. The optimized entry with Hessian calculated must be included and it must show lowest energy. The scanned data points should show various relative energies. In the tutorial, “*relaxed.qmd*” has been pre-calculated. There are 75 data points in this file; one of them has the minimum energy with Hessian calculated.
2. Fit the *“relaxed.qmd”* using the previously demonstrated procedure. Because energies are not too high in this case, there is noneed to use *Boltzmann Factor*. When the fit is over, the energy dialog should look like this:



1. To examine quality of fit, export the data to a .csv file and graph the input vs. fit energies. Note the excellent agreement between forcefield and QM curves.



*The relaxed sampling is more time-consuming than the rigid sampling, but the internal energy curve is predicted more accurately.*

### Tutorial B3 - Fitting a Molecular Dimer

In this tutorial, we will learn how to fit QM data of a molecular dimer to determine VDW parameters, demonstrated by methane molecules. To fit a VDW dimer, three models are required: dimer and two monomers. The two monomers may be the same as shown in this example. In practice, valence parameters should be determined before the VDW parameters to be adjusted. In this tutorial we will checkout parameters from TEAMFF.

*The initial parameter PPF file and QMD files for methane monomers and dimer can be obtained following the steps below. For convenience, these files have been prepared and provided in the “*tutorialB3\_fitCluster” *directory.*

#### Preparing Forcefield

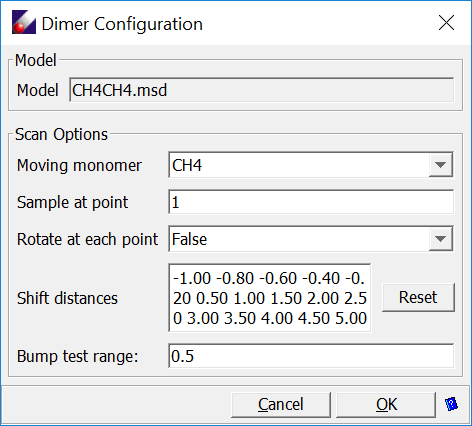
1. Start DFF and open the project of *"Tutorials.dpf"*. In the **“tutorialB3\_fitCluster”** folder, double-click **"CH4.msd"** and **"CH4CH4.msd"** to open the molecular models.
2. Select both models in *Project Navigator* and use **TEAMFF→Checkout** command to get a PPF file for methane from ***“AMBER-General”*** forcefield table. The result PPF file will be loaded to the main screen. All parameters are fix indicated by the asterisk (\*) sign. Select the ***“N12\_6”*** parameters and click **Relax** button in the forcefield spreadsheet, the VDW parameters are relaxed. Click **Save** to save the changes, and then **OK** to close the window.

Graphical user interface, text, application, email

Description automatically generated

#### Preparing QMD

1. Click **QMD→Compute** command to optimize and calculated Hessians for both models at MP2/6-311++G (2d,2p) level of theory. After the jobs are finished, make a new MSD model file by selecting the Gaussian output file, **CH4CH4.log**, and tight-click “**Create MSD**” command in *Project Navigator*.
2. Double-click **“CH4CH4.msd”** in the *Project Navigator* to open it. Then select **Build → Dimer Configurations**to open the following dialog:



Leave all options as given, click **OK** to generate models in a sub-folder.

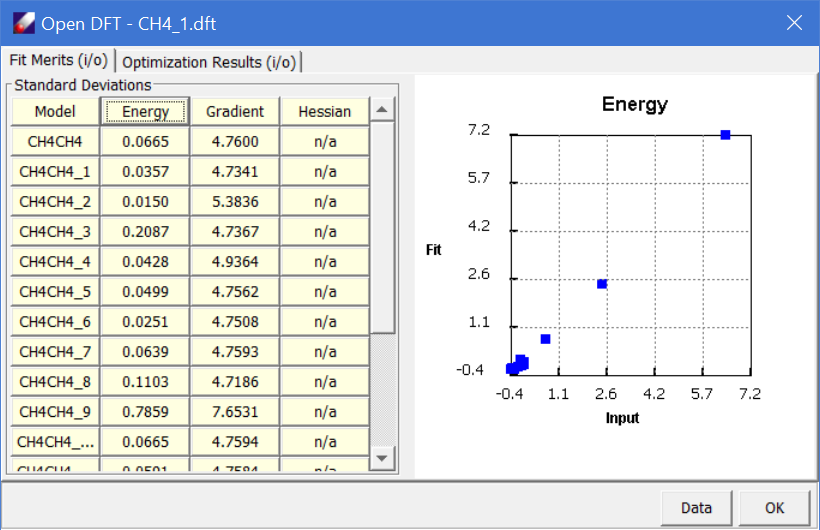
*This dialog sets the sampling points by shifting and optionally rotating one monomer relative to another. The shift distances in Angstroms are listed as the “Shift distance”. The values can be manually edited. At each point, the number of random rotations can be changed. This provides a way to sample the intermolecular interactions much more thoroughly. However, limited by the simple functional forms in common forcefield, fitting the potential energy surfaces might not be satisfactory.*

1. Compute QM data at the same level of theory, MP2/6-311++G (2d,2p), for each of the sampling configurations (There should be 15 sampling points). Select all generated sampling models from the *Project Navigator* and click **QM → Compute**, set the Computation Type to “**GRAIENT**” and carry out the calculations.
2. When all Gaussian jobs are done, make two QMD files: “CH4.QMD” from the optimized with Hessian calculated CH4.log and “CH4CH4.qmd” from the optimized-hessian and all sampling Gaussian output files. This can be done by using **QMD → Organize QMD** command to open a new QMD dialog, and then use the **Add** button to browse and load the Gaussian output files.

#### Fit VDW Dimer

1. Select both **“CH4CH4.qmd”** and **“CH4CH4.qmd”** models, which are associated with their MSD files, the MSD files are associated with the PPF file, Open **“CH4.ppf”** to list the parameters in *Forcefield Editor* spreadsheet. Make sure the **N12\_6** parameters are relaxed (without the asterisk \* mark).
2. Use **Parameter → Fit VDW Dimer** to open *"Fit Cluster Configuration"* dialog. Both models are listed in the *“Data to fit”* list. We need to identify which one is cluster, which ones are monomers. Select **“CH4CH4.QMD”** and click **Cluster**, select **“CH4.QMD”** and click **Monomer,** twice. In the end, the dialog looks like:

Keep other parameters unchanged and click **Execute** to start fitting. When the job is done, a validation panel will appear:



*It should be noted that the fit of isolated molecular dimer is only useful for describing intermolecular interaction in gas phase. For molecules in condensed phases, the VDW parameters should be optimized using the simulation data as shown in next tutorial.*

### Tutorial B4 - Fitting Liquid Data

In this lesson, we will learn how to optimize VDW parameters by fitting liquid properties using liquid benzene. We will check out the parameters from a forcefield of TEAMFF, fit the L-J parameters using simulation data of the liquid.

Because the NVT simulation is more stable than NPT simulation, we use NVT simulation to calculate pressure (P) and heat of vaporization (Hv) as the target properties to fit, and the temperature and density are fixed as input.

In this tutorial, we use the experimental data of one temperature (298 K), there are only two target data to fit, therefore we must reduce the adjustable parameters. After all valence and charge parameters are fixed, there are still have 4 adjustable parameters. Therefore, we choose to use two scaling factors for LJ well-depth and radium respectively, to optimize.

#### Preparing A Liquid Model

*An equilibrated liquid simulation box has been prepared and is included in the tutorial directory. You can skip this section.*

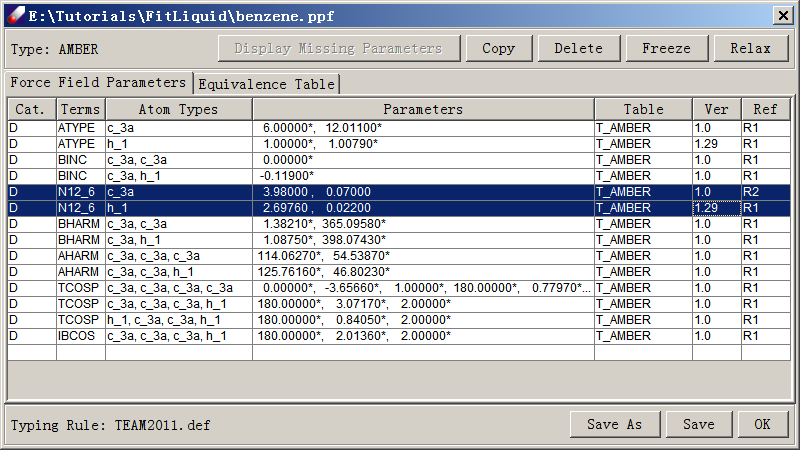
1. Sketch a benzene molecule as illustrated in Tutorial 1. Check out forcefield parameters using **TEAMFF → Check out**. This will generate a forcefield in the *Project Navigator*. We are using the **AMBER-Basic** forcefield as an example.
2. Assign charge groups for benzene molecule using **Build → Charge Group**. The charge groups, atom types, and associated forcefields for this model will be carried over when building the liquid box.
3. Select **Build →** Bulk Liquid to build a liquid box. Select the molecular model from the pull-down menu and enter "100" as the **Count**. This will update several variables in the dialog window. Enter a **Density** of "0.872" and leave the **Compress Rate** as given. Click Execute to launch a background job.
4. When the job is finished, a liquid box with periodic boundary condition will be created. Optimize the constructed liquid cell by using **Simulation → Optimization**. A few dozen steps (~60) should be sufficient. Use **Nonbond Energy** to set nonbond calculation options. Select **Group Cutoff**, set the **Cutoff** value to "9.5", and make sure "Tail corrections" is checked. Close the "Nonbond Energy" dialog by clicking **OK**. Launch the optimization job. The total energy should decrease rapidly.
5. Equilibrate the liquid model by running MD simulations repeatedly. Consider using high temperatures (~3000 K) with NVT MD simulations, and then gradually cool the system down to 298K over several (2~3) steps. Each step requires a run time of ~10 ps. Finally, run a NVT MD simulation with a 9.5Å cutoff for 50-100 ps to equilibrate the system.

#### Optimizing VDW Parameters

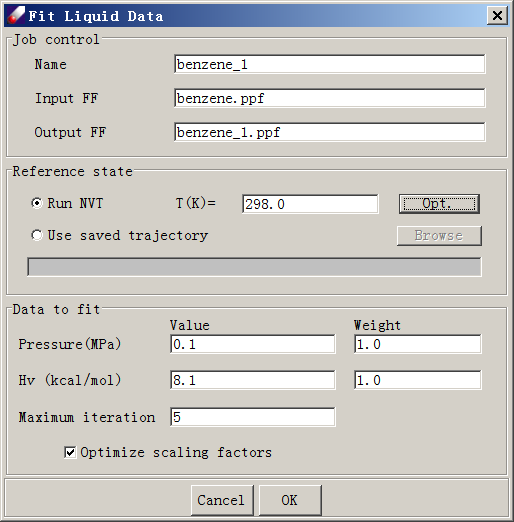
*To fit the L-J parameters, we use experimental data for the density and heat of vaporization for liquid benzene:*

Temperature = 298 K  
Density = 0.872 g/ml  
Pressure = 0.1 MPa  
Hv = 8.1 kcal/mol

1. Load the equilibrated liquid model. Make sure that the forcefield is associated with the liquid model. The L-J parameters must be relaxed.



1. Open Parameter **→** Fit Liquid Properties. Select **Run NVT**. Enter 298.0 as the temperature T (K), 0.1 as the Pressure (MPa), and 20.1 as the heat of vaporization Hv (kcal/mol). Click **Opt to set simulation conditions**. By default, the following values are given: ***1*** fs for **Time step**, ***10,000*** steps for equilibration and ***20,000*** steps for evaluation. The simulation trajectory must be saved and the default interval is every 20 steps. Make sure **nonbond options** are set as before. In this case, a group-based method with a cutoff value of 9.5 A and **tail correction** is adequate. Leave other options as illustrated in the following figure and click OK to launch a job.



*If you did not select* ***Screen View*** *in the* ***NVT simulation control panel*** *(activated by clicking* ***Opt*** *in the* ***Fit Liquid Data*** *window), the job will run in background. You can now close the Direct Forcefield interface. However, if you are running* DFF *on Windows platforms, a command prompt window will remain on the screen.*

1. When the job is finished, examine the results in the output file (<jobname>.dfo). At the end of the output file, a section of summary of the fitting merit is given.
2. Repeat the calculation until calculated values or parameters are converged. Note that convergence is subject to the statistic fluctuations. In a system of about 1000 atoms, evaluated pressures are normally associated with uncertainties of several thousand bars (hundreds of MPa). Normally, values within 10-20 MPa of the experimental value are considered acceptable. (With this magnitude of deviation, the percentage errors in densities obtained using NPT simulations are normally within 1%.)

*After the parameterization, validation should be carried out to test the parameters. For example, calculating densities at different temperatures using NPT simulations with LAMMPS are usually a good test of the parameters obtained.*

## 2-3. Section C - Developing Force Fileds

In this section we practice the advanced tasks for making forcefield in TEAMFF environment. We will learn how to prepare a training set and how to make a new forcefield in TEAMFF database. Another important task is to import existing forcefields to TEAMFF so that available resources can be utilized. Finally a special topic is given to hyper valence molecules.

### Tutorial C1 - Make A Training Set

In this lesson, we practice how to prepare a training set, one of the most fundamental tasks in forcefield development. The molecules of the training set should a) represent all interactions of the target molecules, and b) as small as possible to avoid unnecessary correlations among the forcefield terms. Both conditions means that a fragmentation must depend on a forcefield to be developed.

1. In *Project Navigator*, click on the node of ***“TutorialC1\_fragment”*** to open the folder. There are dozens of drug-like molecules saved in this folder. Many of these molecules are too large to be parameterized directly, smaller fragment molecules are required. Open at least one of these molecules to activate the commands.
2. Make sure all models are selected, use **Forcefield → New Forcefield** command to initialize a forcefield for these molecules. Select ***“AMBER”*** as the forcefield type and “***Default typing rule”***. Click **OK** to create an empty forcefield with atom types and equivalence table created.
3. With the new forcefield created and atom types assigned, select all models in the folder, and **Build → Fragment** command to launch a background job to make fragments. In the end, 29 molecules will be created and saved in the folder “*Required\_Fragments*”, which is the training set representing those drug-like molecules. If the training set is parameterized, the result forcefield is adequate to describe the target molecules.
4. To illustrate how it works with a quick turnaround, let’s estimate a forcefield. First create another forcefield in the folder of “*Required\_Fragments* by using **Forcefield → New Forcefield**, name the new forcefield as “*frag.ppf*”. Then, click **Parameter → Estimate Charges** and **Parameter → Estimate Parameters** commands to make an estimated forcefield. By default, the result forcefield is named as **“*frag\_EC\_EP.ppf*”**. This forcefield covers all parameters for the training set. The coverage is greater than that required for the drug-like molecules.
5. To verify the drug-like molecules are covered, copy ***“frag\_EC\_EP.ppf”*** to upper directory where the drug-like molecules are saved. Select all the drug-like molecules from *Project Navigator*, and right-click the mouse button, select **Associate PPF** command to make the selected molecules associated with **“*frag\_EC\_EP.ppf*”**, which assigns the forcefield to the selected molecules. *DFF* should not report any missing parameters.
6. We can further test the assigned forcefield by carrying out a calculation. Select one of the drug-like molecules from *Project Navigator*, use **Simulation → Optimization** or **Molecular Dynamics** to run a quick job.

*The fragments depend on how atom types are defined. In this lesson, we use “Default” typing rule. If different typing rules or different equivalence tables are used, the result fragments could be different. The fragmentation method assumes that the atom types are defined using the first circle of neighbor atoms of the atom to be typed, which is sufficient in most cases. Therefore, if you define the typing rules using information of atoms beyond the first circle of neighbor atoms, the fragments generated may not be adequate, iterations might be necessary to make sure the training set has all energy terms represented.*

### Tutorial C2 - Make a New Forcefield

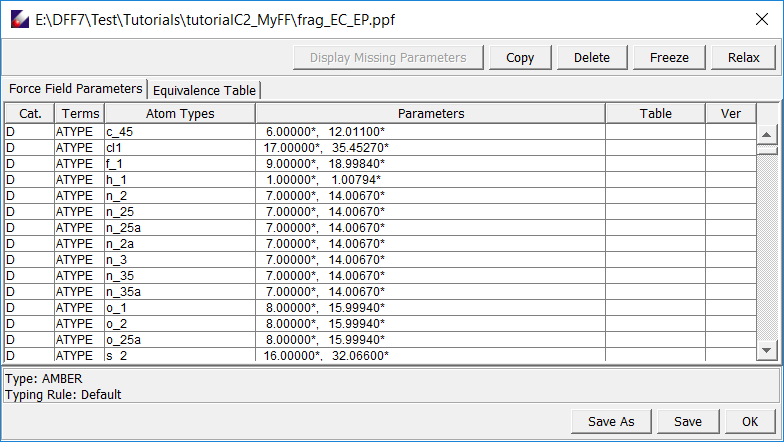
In this lesson, we make a new forcefield table in TEAMFF database. Let’s use the ppf file we obtained (estimated) in previous lesson, “*frag\_EC\_EP.ppf”,* make a new forcefield table in TEAMFF. We will use “*hydroxychloroquine.msd*”, also copied from the last tutorial to test the new forcefield.

*The TEAMFF database is saved in <DFF Installation>/database folder. If this folder is protected, the database file may not be writable. You must get the write permission before running this tutorial. It is also good practice to make a copy of the database so that the original database is protected. The database includes one binary file (TEAMFF.TEAMFF) and a folder in the same root-name (TEAMFF) in the database folder.*

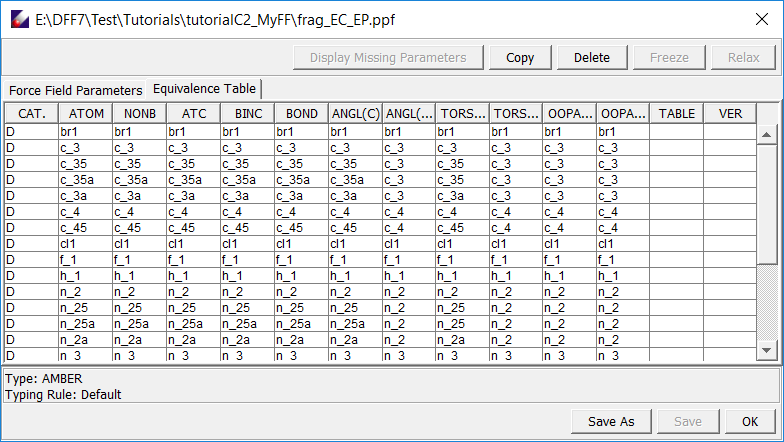
1. We first created a new table in TEAMFF database. Click TEAMFF → Admin command to open *“TEAMFF.TEAMFF”* database. Select *AMBER* as the forcefield type, the dialog lists several force-field tables in AMBER type. Click Add button on the right panel to open a new dialog for adding a table. Enter *MyFF* as the forcefield name and select *AMBER* as the forcefield type. Select *Use default typing rules*, which must be the same as those used for forcefield to be checked in. In the *Description*, enter a brief note for the forcefield. Clicking OK to close this dialog. A new forcefield table is listed in the TEAMFF Admin dialog.

*Before checking in parameters,* DFF *prompts another test to make sure the parameters are ready for checking in. Only entries marked as “D” denoting “developed” can be checked in. If the parameters are obtained by fitting, the entries will be marked as “D”. Since “frag\_EC\_EP.ppf” is estimated, the entries are either unmarked or marked as “A” standing for “Automatic”. Let’s fool the system by making all entries as “D”.*

1. Click **“*frag\_EC\_EP.ppf*”** to open the *Forcefield editor*. Delete entries of “E0”, which are not energy parameters but the values to set relative energies, then select all entries, right-click mouse button to select “*Label as “D”*” command. Repeat the operation for the “*Equivalence Table”*. The *Parameter* page should look like this:



And the ***Equivalence Table*** should look like this

Save the change.

1. Now we proceed to check in the forcefield. Select the forcefield from the *Project* *Navigator* and click TEAMFF → Check in command. Make sure the forcefield table, *“MyFF”,* is selected. Enter a *Descriptio*n for this task, and then click OK. A pop-up window will display the number of parameters (491 entries) will be checked in. Click OK to continue.
2. The parameters should be checked successfully. To verify it, open TEAMFF Admin page again. Make sure the forcefield type is AMBER, you may need to Refresh the list. Select the *“MyFF”* table, we will test this forcefield.
3. Select *“hydroxychloroquine.msd”* in the *Project Navigator* and use TEAMFF → Check out command to check out parameters. Make sure the forcefield is selected correctly, click OK to execute. The parameters checked out are associated with the molecule, which can be tested by running a quick simulation.

### Tutorial C3 - Import A Forcefield

In this tutorial, we will practice how to integrate an external forcefield into TEAMFFas a new forcefield table, which is another way to extend the coverage of TEAMFF. External forcefields can be taken from literature or from other developers. As long as the forcefield type is consistent, the imported forcefield can be used with other forcefields in TEAMFF.

Two text file must be prepared in order to import a forcefield: a “*.ppf”* file containing the parameters and a *“.def”* file of the typing rule. DFF has tools to prepare the *“.def”* and *“.ppf”* interactively.

To illustrate how this works, let's use the molecule (“*diethanolamine.msd”*) used in previous lesson. We assume a forcefield is in AMBER type, and the atom types used in this forcefield are:

C1 - for carbon  
O1 - for oxygen  
N1 - for nitrogen  
HO - hydrogen bonded to oxygen  
HN - hydrogen bonded to nitrogen  
HC - hydrogen bonded to carbon

Because the forcefield is independently stored and used, the atom types can be defined as simple as possible. As shown above, only hydrogen atoms need to be differentiated, for other atoms – carbon, oxygen and nitrogen, the atom types can be as general as the element symbols.

In the following steps, we will first see how to prepare the definition file using DFF.

#### Defining Typing Rules

1. Using **Forcefield → Define Typing Rules** command to open “*Atom Type Definition”* dialog. Click **New/Select** button to create a new typing file entitled ***"MyFF.def"***. Note that all typing rules must be saved in <dff-root>/data/TypingScripts/ folder. A definition file, *“Tutorial.def”*, which is the result of this tutorial, has been saved in the directory for comparison.

*Since the typing rule is essential data, all typing rules are saved in a special folder under the DFF installation directory* <dff-root>/data/TypingScripts/*. You need to have access to this folder in order to save and use the typing rules you created.*

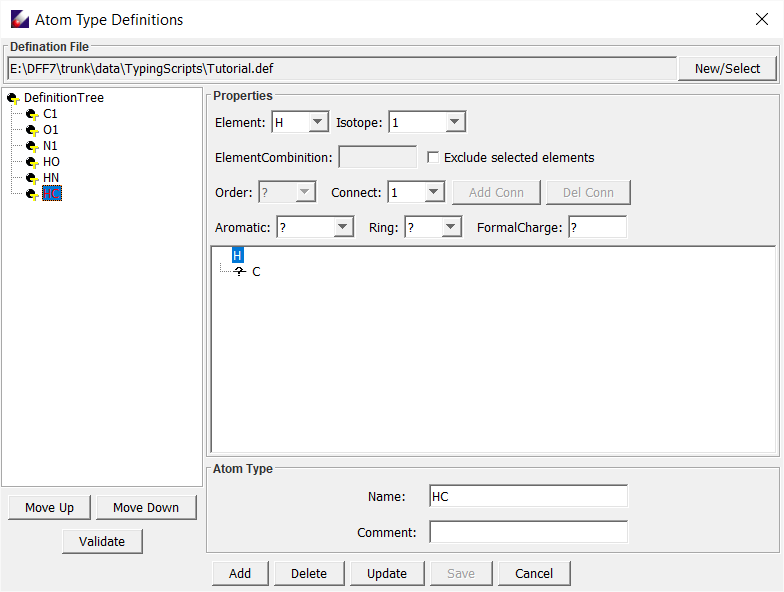
2. Select the root node (*"****DefinitionTree****"*) in the left panel of this dialog and select ***C*** as "Element" in the right panel. Since we will only handle one carbon atom type, there is no need to specify any additional attributes to distinguish a given carbon atom in the molecule. Enter ***"C"*** in the editable field as seen below and click **Add**. A new atom type ***"C1"*** is added into the ***DefinitionTree*** on the left side.

*Any symbol can be used for representing atom types. It is a good practice and useful for parameterization to name the atom types in a consistent and self-explanatory way so that the atom types can be easily recognized.*

3. Repeat the above steps for adding the atom types for oxygen and nitrogen. Note that each of these atom types is distinct with no inheritance relationship (*i.e.*, not subtypes of one another), so each type should be added to the base node ("***Definition Tree***".) The result should look like this:

4. Let's make the hydrogen atom types now. There are three different hydrogen atoms in the molecule depending on whether the hydrogen is bonded to oxygen, nitrogen, or carbon. We will need to differentiate them. First, let's define hydrogen bonded to oxygen. Click on the "***DefinitionTree****"* node and select element ***H*** in the right panel. Set Connect as ***1*** (to indicate one bond). In the editable field, click on the new node under H marked "**?**". The top section above this field will now refresh. Select element ***O*** to indicate the hydrogen is bonded to oxygen. Then enter ***"HO"*** as the atom type name.

5. Repeat the process for the other two hydrogen atom types. When all done, it should like this:



6. Click **Save** to save the definition file, which looks like this:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! User atom types defination for DFF (Wed Jan 31 17:10:35 CST 2018)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Define C1

C

atom 1: coor=?

End

Define O1

O

atom 1: coor=?

End

Define N1

N

atom 1: coor=?

End

Define HO

H [&O]

atom 1: coor=1

atom 2: coor=?

End

Define HN

H [&N]

atom 1: coor=1

atom 2: coor=?

End

Define HC

H [&C]

atom 1: coor=1

atom 2: coor=?

End

RelationTree:

(C1

)

(O1

)

(N1

)

(HO

)

(HN

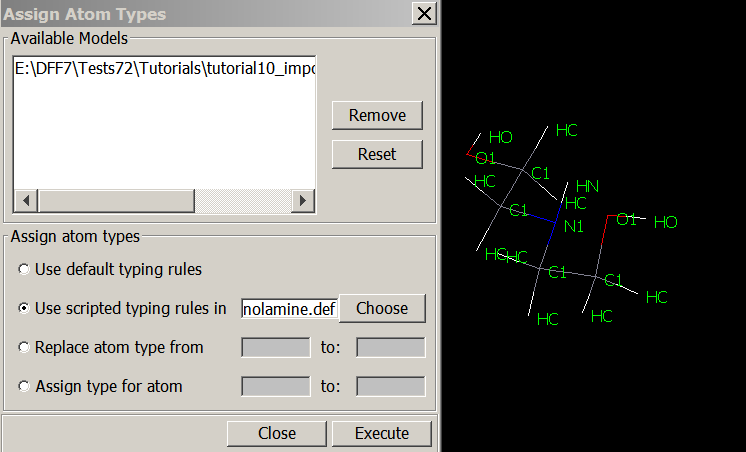
)

(HC

)

End

7. Test the new typing rule. Select ***diethanolamine.msd***, click **Forcefield → Assign Atom** Types, and click **Choose** to select a typing rule. Browse and find the definition file you created, and click **Execute** to get the atom types assigned.



Click **Execute** to assign atom types. Atom types should now be correctly assigned to the molecule.

#### Preparinge a PPF File

8. We will now prepare a PPF file. The PPF format is explained in Chapter 6. In this case, the file should read as follows:

#DFF:PPF

#PROTOCOL = AMBER

#TYPINGRULE = MyFF.def

ATYPE: C1 : 6.00000,12.01100 : C D

ATYPE: HC : 1.00000, 1.00790 : C D

ATYPE: HN : 1.00000, 1.00790 : C D

ATYPE: HO : 1.00000, 1.00790 : C D

ATYPE: N1 : 7.00000,14.00670 : C D

ATYPE: O1 : 8.00000,15.99940 : C D

BINC: C1 , C1 : 0.00000 : C D

BINC: C1 , HC : 0.00150 : C D

BINC: C1 , N1 : 0.18840 : C D

BINC: C1 , O1 : 0.19890 : C D

BINC: HN , N1 : 0.34380 : C D

BINC: HO , O1 : 0.41360 : C D

N12\_6: C1 : 3.93000, 0.06600 : C D

N12\_6: HC : 2.72000, 0.03000 : C D

N12\_6: HN : 2.72000, 0.03000 : C D

N12\_6: HO : 2.72000, 0.03000 : C D

N12\_6: N1 : 3.69300, 0.14500 : C D

N12\_6: O1 : 3.37000, 0.17000 : C D

BHARM: C1, C1 : 1.50740, 257.60920 : C D

BHARM: C1, HC : 1.10030, 337.96220 : C D

BHARM: C1, N1 : 1.45050, 269.52610 : C D

BHARM: C1, O1 : 1.40890, 260.18510 : C D

BHARM: HN, N1 : 1.01940, 494.86480 : C D

BHARM: HO, O1 : 0.97090, 576.31990 : C D

AHARM: C1, C1, HC : 114.51930, 35.72660 : C D

AHARM: C1, C1, N1 : 112.94380, 55.26240 : C D

AHARM: C1, C1, O1 : 110.63270, 65.44320 : C D

AHARM: C1, N1, C1 : 111.58130, 93.59760 : C D

AHARM: C1, N1, HN : 107.62090, 48.64750 : C D

AHARM: C1, O1, HO : 105.31840, 56.30250 : C D

AHARM: HC, C1, HC : 113.08930, 37.03450 : C D

AHARM: HC, C1, N1 : 116.67120, 35.28640 : C D

AHARM: HC, C1, O1 : 113.03220, 57.01090 : C D

TCOSP: C1, C1, N1, C1 : 0.0, 0.40790, 3.0, 0.0, -2.92360, 1.0, 180.0, 0.78360, 2.0 : C D

TCOSP: C1, C1, N1, HN : 0.0, -0.11820, 3.0, 0.0, -0.82030, 1.0, 180.0, -0.71270, 2.0 : C D

TCOSP: C1, C1, O1, HO : 0.0, 0.15760, 3.0, 0.0, -1.08540, 1.0, 180.0, -0.51830, 2.0 : C D

TCOSP: C1, N1, C1, HC : 0.0, 0.39130, 3.0, 0.0, -0.74250, 1.0, 180.0, -0.15020, 2.0 : C D

TCOSP: HC, C1, C1, HC : 0.0, 0.10950, 3.0, 0.0, 9.90490, 1.0, 180.0, -0.98030, 2.0 : C D

TCOSP: HC, C1, C1, N1 : 0.0, 0.04500, 3.0, 0.0, 10.64810, 1.0, 180.0, -0.65010, 2.0 : C D

TCOSP: HC, C1, C1, O1 : 0.0, -0.18490, 3.0, 0.0, 9.79680, 1.0, 180.0, -1.54950, 2.0 : C D

TCOSP: HC, C1, N1, HN : 0.0, 0.31100, 3.0, 0.0, 0.07450, 1.0, 180.0, -0.58790, 2.0 : C D

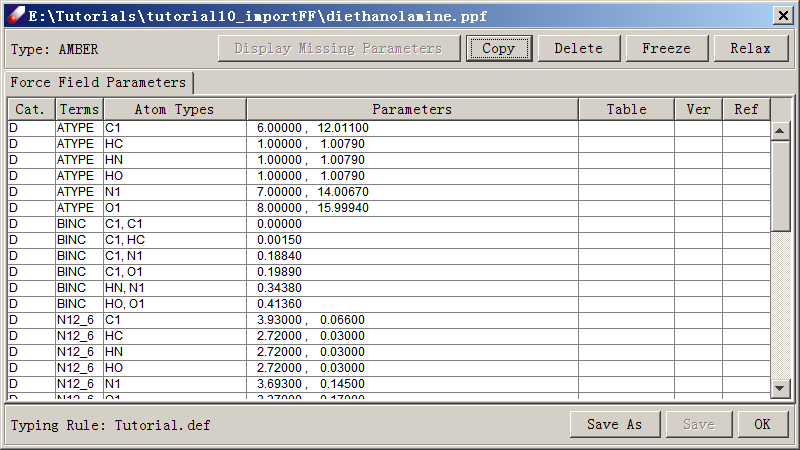
TCOSP: HC, C1, O1, HO : 0.0, 0.35220, 3.0, 0.0, 0.87650, 1.0, 180.0, -0.82550, 2.0 : C D

TCOSP: N1, C1, C1, O1 : 0.0, 1.24760, 3.0, 0.0, 9.61000, 1.0, 180.0, -1.01000, 2.0 : C D

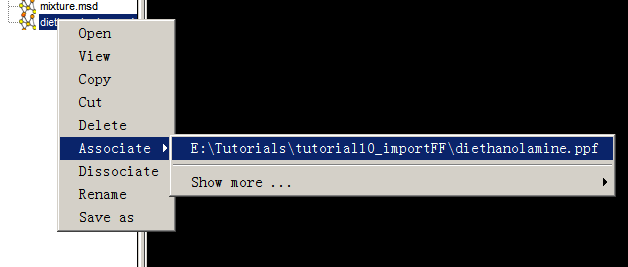
IBCOS: C1, C1, N1, HN : 180.0, 0.00000, 2.0 : C D

The 2nd and 3rd lines indicate the forcefield type and typing rules. The rest are parameters from different sections. Note that there is no equivalence table for this forcefield, which means the apparent atom types are the same as the atom types used in identifying parameters, which is acceptable.

Double-click to open the diethanolamine.ppf file:



9. Assign this forcefield to ***diethanolamine.msd*** to test if atom types are assigned correctly and parameters are complete. This can be done by selecting the model, right-clicking, selecting **Associate**, and selecting the appropriate forcefield:



10. Run simulations using **Simulation → Optimization** or **Simulation → Molecular Dynamics** to verify that these parameters work properly.

11. You may create a new forcefield table using the *AMBER* format and check in parameters as explained in previous lesson. This forcefield can be used with other *AMBER* type forcefields in TEAMFF.

### Tutorial C4 - Hypervalence Molecules

In this lesson, we learn a special treatment of coordination bonds of molecules with hypervalence. Hypervalence is common in many transition-metal complexes. Normal forcefield function forms are capable to describe the variation in bond angles. By using the hypervalence modifier as explained in Chapter 3, we can distinguish different bond angles using simple forcefield type form.

1. In *Project Navigator*, click on the node of **“TutorialC4\_hypervalence”** to open the folder. There are several molecules listed. Open and view these molecules on the screen, they represent typical coordination structures: octahedral (AsF6), trigonal bi-pyramidal (AsF5), T-Shape (IF3), square pyramidal (IF5), See-Saw (SF4) and square planar (XeF4).

*Note that these bonds are treated as covalent bonds. The assumption is that these bonds are stable under normal condition of simulation.*

1. Select one or more of these molecules from the *Project Navigator*, then use **Forcefield → New Forcefield** command to initialize a forcefield. In the dialog, select *“Type”*, and choose a typing rule using ***“Default typing rule”*** or any typing scripts. Select “Hyper valence modifier”, which works with any typing rules. Click **“OK”** to go.
2. A new forcefield is created, although the parameters are empty, atom types and equivalence tables are prepared. Examine the atom types, you will see the atom types of ligands are appended with ‘@’ or ‘%’ to differentiate the equatorial and axial positions. Check the Equivalence Table, only ANGLE-S (side) terms keep the special characters.
3. Use **Parameter → Estimate Charge** and **Parameter → Estimate Charge** commands to make automatic parameters. Examine the parameters obtained, the parameters of angles terms are identified by the atom types with the special characters, the reference angles are differentiated depending the relative positions of the atoms, in terms of 180, 90, and 120 degrees, as shown in following figure.

Graphical user interface, text

Description automatically generated

1. Run quick MD simulations for any of these molecules to see how the structures are shown.

*Although in this tutorial we did not fit QM data, the concept is demonstrated. The QM calculation can be carried out to test if the optimized structures are as illustrated, and a rigorous forcefield can be obtained by fitting the QM data, which will yield reasonable structures, energies as well vibrational frequencies.*

## 2-4. Section D – Quantum Mechanics Data Forcefield

### Tutorial D1 – Use QMDFF Database

In this lesson, we practice how to use QMDFF method to make forcefield for simulation models. The concepts are explained in [a section of Chapter 3](#_3-12._Quantum_Mechanical) and an overview of the commands are explained in [a section of Chapter 4](#_4-9._QMDFF_Menu).

#### Database

The functions of QMDFF are listed under the QMDFF menu. First, let us have a look at the database.

1. Select **QMDFF → DB Admin**,the currently active database is displayed in a dialog. The top section lists where the database file is located. **Browse/New** command is used to select or create a database. List command update the contents. There are three tabbed panels each showing a repository table: Fragment QMD, Fragment TODO and Molecule PPF.
2. Click **Browse/New** to browse folders. The default is with DFF release is located at <DFF root>/database folder, and the file name is QMDFF.db. The database can be placed any accessible directories on your computer. Select a file and click Open will load the database. Enter a file name and then click **Open** will create and load an empty database.

Under the QMDFF menu there are several commands, we will practice the semi-automatic procedures: **Enquiry, Collect QMD, Smart Fit,** and **Assign & Upload PPF**. These commands are designed to be executed in order.

#### Use Existing Forcefileds

We first use **Enquiry** to search existing forcefield. There are several MSD files provided in the *tutorialD1\_QMDFF* folder: *boxMix.msd* is a mixture of water and phenol, *PFMO.msd* is the same polymer used in previous tutorials, and *R4Q.msd* is a small-molecule inhibitor (Nature, volume 588, page 712–716, 2020). The molecules in these models have been parameterized so that we can use them directly. We will use DFF to find molecules and search QMDFF database for molecule forcefields. If succeed, a merged forcefield will be saved and assigned to the simulation model.

1. Use **QMDFF → DB Admin** to switch back to the default database, “*QMDFF.db*” for the following steps.
2. In *Project Navigator*, click on the node of **“TutorialD1\_useQMDFF”** to open the project. Let’s start with the mixture, “**boxMix.msd**”, which is 3D periodic box of water and phenol molecules.
3. Click **QMDFF → Enquiry**, in the dialog select a forcefield type and then click **OK** to start the background job. When the job is finished, read the content in the Job Dialog, and then click the dialog by clicking on OK. Since both water and phenol forcefields are in the database, the results, “boxMix\_mrg.msd” and “boxMix\_mrg.ppf”, will be made and loaded to the *Project Navigator*. This model is ready for simulations, either internal or using external simulation engine GROMACS or LAMMPS.
4. Repeat the same operation for other models, in each case the result PPF and MSD files will be prepared, and they are ready for simulation.

### Tutorial D2 – Make new Forcefields

*For molecules that have not been parameterized, using* **Enquiry** *command will prepare the molecules and their fragments for parameterization. This is best illustrated by creating a new database.*

#### Enquiry

1. Use **QMDFF → DB Admin** to open a dialog, click **Browse/New** button to make a new database, we use **“tutorial.db”** in the following steps.
2. Select *PFMO.msd* from the folder “*TutorialD2\_makeQMDFF*” in the *Project Navigator*. We use this model as an example; the same procedure can be applied to any other models. Then click **QMDFF → Enquiry** command. Select a forcefield type, and then click OK, a background job starts, and a job monitor window named “Enquiry” should appear.
3. When the job is done, we should read something like “One unique molecule is found, there is no forcefield in the database and 3 fragments are found and saved”. Close the task dialog, we find a new model, M1.msd, together with a new folder M1. M1 is a unique molecule found by DFF. Inside of M1 subfolder, there are three new models *F1.msd, F2.msd and F3.msd* which are fragments made from M1.

*The molecule is one of those in the simulation model. Others, regardless of the molecular weights, are equivalent to DFF. The fragments are standardized, and they represent the polymer, again regardless of the number and molecular wights in the simulation model. Our attentions are now on these fragments and make a forcefield for the polymer.*

#### Collect QMD

1. Select the three fragments, ***F1.msd, F2.msd,******F3.msd***, and then click on **QMDFF → Collect QMD** command. DFF will search the current folder and the database (Tutorial.db) for QMD files of these fragments. Since nothing exists, a “DFFQMD Job” dialog appears. This dialog lists the fragments to be calculated, the quantum mechanics calculation options, etc. The defaults are used for building up the QMDFF.db. Keep the default and click **OK** to continue.

*Feel free to use different options as you are making a new database. Note the options for bases set “6-31+G(d,p)/DGDZVP” means the first for elements up to Kr (#36) and the second for the rest. The options for Total charge and Multiplicity are COMPUTE means they are calculated depending on the formal charges. We can make input files only or run the jobs directly. The option in Task shows that DFFQMD will do three types of jobs: optimization, scan dihedral angles and normal mode analysis, all automatically.*

1. Since we select to “Make input files only”, the job finished by saving DFFQMD input files in a new sub-folder “qmd”. Refresh the folder to see these files. Since the DFFQMD jobs are time consuming, to run on the same computer is okay, but it is preferable to run in batches. A batch file is saved in this folder. To see the folder, select **“qmd”** folder in the *Project Navigator*, right-click mouse button, and then select **Open in Explorer**. Meanwhile, the three fragment molecules are saved in the Fragment TODO table. Click **QMDFF → DB Admin** to confirm. When their QMD files are uploaded, these files will be removed.

*When the DFFQMD jobs are finished, the QMD files should be saved in the “M1” subfolder. For convenience of this tutorials, the QMD files are prepared and provided in the tutorial’s subfolder “qmdPFMO”. Copy them to “M1” subfolder to continue.*

#### Smart Fit

Select *all three* QMD files in the “M1” subfolder, and then **QMDFF → Smart Fit** to open a dialog. The selected three QMD files should be listed in the dialog. Select a force field type, then click OK to launch the job. The Job monitor window shows information of the background job. When it is done, click OK to close the window. Examine the DFT file to see the validation results.

*It is important to note that we must fit all fragment molecules together so that the molecule is covered by the result forcefield. It is instructive to open the PPF file from File Explore to see there a line in the PPF file that shows the combined SMILES indicating that this forcefield is for a molecular system represented by three (fragment) molecules.*

#### Assign and Update

1. To test if the result forcefield works on the target molecule, select the results PPF file, and open **QMDFF → Assign and Update PPF** to open a dialog. The dialog shows the molecule (M1.msd) and the forcefield made from fragments and assigned to the molecule. A short molecular dynamics job is prepared to test if the forcefield is stable. When the job is done, examine the molecule on the screen. If there is no obvious distortion, agree to upload the result PPF.

*The PPF uploaded can be overwritten (updated) repeatedly if any revision is made.*

1. We also want to upload the QMD files so that they can be repeatedly used. To do this, select the QMD files and then click **QMDFF → Update QMD** command. When the job is done, an information window appears indicating that the QMD files are loaded, and the TODO repository is cleared. Use **QMDFF → DB Admin** to confirm.
2. Now we can repeat step 2 above to use the **QMDFF → Enquiry** command on the simulation model, PDMO.msd. Two new files, PDMO\_mrg.msd and PDMO\_mrg.ppf should be made and saved in the *Project*. They are ready for simulations.

This ends this this lesson. The procedure can be applied with the system database *QMDFF.db*. The difference is that new QM jobs or fitting jobs won’t be called unless it is necessary to do so.

# CHAPTER 3 – CONCEPTS, METHODS AND ALGORITHMS

## 3-1. Project and Files

### Project

DFF organizes user files by project. A project consists of a project folder (directory) named "<project name>" and a project file named "<project name>.dpf". The "<project name>" can be any string, but must be identical for both the project folder and the project file. Both the project folder and the project file must also be placed in one folder, which can be anywhere on your computer file system that you have read and write access rights.

The project is potable, which means you can copy or share a project. Make sure to bundle the project folder with the associated project file.

### Project Navigator

The file structure of an opened project is displayed on the left side of the DFF user interface using a computer tree structure. This area is called the *Project Navigator*, andis an interface between your computer files and DFF. However, files relevant to DFF are displayed. You can get into the computer file system from DFF by using one key command **Open in Explorer**.

Operations performed on files or folders in the *Project Navigator* are applied to real files. For example, deleting a file or a folder from the *Project Navigator* also immediately deletes it from your computer. *Be careful that files deleted in the Project Navigator will not be stored in the Recycle Bin.* New files or new folders can be added to the *Project Navigator* from DFF*.* Since any sub-folders under the project folder are considered as a part of the project, if a folder or file is added to the project folder or its sub-folders in the operating system, it will be treated as a part of the project. The **Refresh** command searches out and lists all sub-folders and relevant files in the *Project Navigator*. Another useful function is **Sort**, which can be used to sort by file name, file type, or date. Files in the Project Navigator can be deleted, copied, moved, added, and renamed. Files in other folders on your computer can be imported into the project through DFF, which will copy the selected files to the project folder on your disk drive.

### File Types

Files used by DFF include molecular structure data (.msd), potential parameters (.ppf), validation result tables (.dft), graphic tables (.plot), quantum mechanics data (.qmd), and several types of text files. Each of these files types has a unique icon in the Project Navigator. The file types and their icons in the Project Navigator are listed as follows:

|  |  |  |
| --- | --- | --- |
| **File Type** | **Icon** | **Description** |
| *Project* | E:\DFF7\Release7.1\UserGuide7\References\Images\project.gifE:\DFF7\Release7.1\UserGuide7\References\Images\folder.gif | The root directory of the project. |
| *Folder* | A folder containing project data. |
| *Molecular Structure Data(msd)* | E:\DFF7\Release7.1\UserGuide7\References\Images\msd.gif | A model file containing structural data. |
| *Structure and Potential File (spf)* | E:\DFF7\Release7.1\UserGuide7\References\Images\spf.gif | A model file containing both molecular structural data and potential parameters. |
| *Quantum Mechanics Data (qmd)* | E:\DFF7\Release7.1\UserGuide7\References\Images\qm.gif | A quantum mechanics data file. |
| *Quantum Mechanics Data (qmd)* | E:\DFF7\Release7.1\UserGuide7\References\Images\ok.gif | A quantum mechanics data file containing Hessian matrix. |
| *Potential Parameter File (ppf)* | E:\DFF7\Release7.1\UserGuide7\References\Images\ppf.gif | A forcefield potential parameter file. |
| *Text* | E:\DFF7\Release7.1\UserGuide7\References\Images\dfi.gif | A text file. |
| DFF *Fit Table (dft)* | E:\DFF7\Release7.1\UserGuide7\References\Images\dft.gif | A table of validation results. |

Many commands of DFF are performed on the type of selected files or folders. The commands are usually accessible by using the command menu or by right-clicking files.

## 3-2. Forcefields

A forcefield is a set of analytic functions that represent Born-Oppenheimer potential energy surfaces for molecules. The complete definition of a forcefield includes three components: functions in terms of coordinates, atom types, and parameters. The coordinates are usually redundant internal coordinates: bond lengths, bond angles, out-of-plane angles, dihedral angles, non-bonded distances, etc. The functions are generally empiric functions originated from chemical intuitions, atom types are used to distinguish atoms of the same element but in different environments, the parameters are associated with atom types and embedded in the functions. All together a forcefield describes how energy changes as the coordinates change.

### Forcefield type

In this document, we refer a combination of functional forms as a **Forcefield type** . DFF supports several common forcefield types including AMBER[3](#_ENREF_3),[4](#_ENREF_4), CHARMM[5-8](#_ENREF_5), OPLS[9-16](#_ENREF_9) CFF[17-22](#_ENREF_17), COMPASS[23-30](#_ENREF_23) and DREIDING[31](#_ENREF_31). In addition, a specially designed forcefield type, TEAM[32-38](#_ENREF_32) is implemented.

In TEAM forcefield type, the bond and angle terms are expressed in polynomial expansions representing anharmonicity. In addition, the bond-bond and bond-angle coupling terms are introduced. The anharmonicity, as well as the coupling terms useful for accurately describe the energy change of molecules with high constrains. To be compatible with popular forcefields, the van der Waals (VDW) terms are modeled by Lennard-Jones-12-6 (LJ-12-6) function. However, one important feature is that the dispersion part if LJ-12-6 function is temperature dependent[35](#_ENREF_35).

The functional forms of TEAM forcefield type are written as:

Additionally, the AMBER forcefield type is commonly used:

More forcefield types are described in the [Chapter 6](#_6.1_Force_Field).

In the above functions, total energy is expressed as a summation of multiple energy terms; each of these energy terms is expressed as a function of one of the internal coordinates: bond lengths, bond angles, torsion (dihedral) angles, improper dihedral angles and nonbond atom-atom distances. In addition to internal coordinates, each energy term contains one or more adjustable parameters. Because atoms in function terms are classified by atom types that represent atoms in certain chemical environments (see next section), atom types are another part of a forcefield. To develop a forcefield requires determination of functional forms, atom types, and parameters.

### Typing Rule

Forcefield parameters are identified by atom types, which are alphanumerical strings representing atoms of the same element in different chemical environments. For example, a carbon atom in a benzene molecule behaves very differently from a carbon atom in an ethane molecule. To distinguish these two carbon atoms, different atom types are used. The chemical environments are characterized by the atomic attributes of element, connectivity (covalence bonds and bond orders), ring size, aromaticity, and formal charge.

Atom types are assigned using typing rules. Different typing rules assign atom types differently, so a specific set of typing rules must be associated with a forcefield. Three categories of typing rules are available in DFF – *Default, Extended Default,* and *Free Defined*. The *Default* typing rules are hard-coded in the software; they are systematic and complete cover all elements in any environments. The *Extended Default* are made based on the default typing rules written as scripts in a file with extension “.ext”. *Free Defined* typing rules are written as scripts in “.def” files.

### Parameters

A forcefield is written as a PPF file with extension of “.ppf”, which stands for potential parameter file. It contains two parts: an equivalence table of atom types and a table of parameters. The file format is further explained in the [Chapter 6](#_6.4_Potential_Parameter). An equivalence table is a map between assigned atom types and atom types used in the forcefield, for the purpose of reducing the number of parameters. Equivalence table is optional, if equivalence table is not provided, a one-to-one mapping is used. For more information about atom type definition and equivalence table usage, see the next section.

There are several ways to add new forcefields to DFF projects. It can be created, imported or checked out from the DFF database. A PPF file can be associated with one or more MSD (molecular structure data) files with extension of “.msd”. When a forcefield is associated with models, DFF assigns the atom types and scans the forcefield for required parameters.

## 3-3. Molecular Structure

Assigning atom types properly is critically important for getting parameters consistently with a force field as designed for. DFF uses molecular structures to define atom types, understanding how the molecular structures are defined is useful for assigning atom types. The molecular structure refers to molecular connectivity, essential atom properties, hydrogen atom state, and bond orders.

### Connectivity

Connectivity indexes are atom numbers that describe how each atom is bonded with others. The indexes can be established automatically by DFF using atom-atom distances, which must be provided with reasonable precision. The algorithm is straightforward: for any pair of atoms, if the distance between the two atoms is shorter than a threshold value defined as the sum of the tabulated valence radii[39](#_ENREF_39) saved in Element.dat file, and a tolerance value (fixed as ±0.2 Å), the atom pair is considered to be bonded.

### Essential Properties of Atoms

From the connectivity indexes, essential atom property such as coordination number and ring sizes are calculated via a complete search algorithm[40](#_ENREF_40) that finds neighboring atoms, rings and ring sizes of each atom. If an atom bridges two rings (e.g., in a fused ring), the smallest ring size is assigned to the atom.

### Bond Orders

Bond orders are estimated based on the octet rule of classical valence bond theory. The algorithm uses the number of available valence electrons, which is calculated as the total number of valence electrons, including formal charge, minus the number of bonding electrons. Each bond is initially assigned as a single bond, if the number of available valence electrons is not zero, the atom is marked as a potential bonding acceptor. The bond order of two adjacent acceptors is then increased by one. This process is repeated until the above condition fails. Note that the octet rule may not be always valid, and the bond orders are calculated for the most common scenarios.

### Hydrogen Atoms

The hydrogen atoms must be explicitly given. In some model files such as PDB and MOL2 the hydrogen atoms may be omitted (implicit). You may use other modeling software or DFF to add hydrogen atoms.

In DFF, hydrogen atoms are added to an atom using the most common number of hydrogen atoms of the atom, minus the number of bonds and formal charge assigned to the atom.

### Formal Charges

Formal charges are used for distinguishing ions from neutral molecules. If hydrogen atoms are omitted, the formal charges must be explicitly specified. An example is acetic acid and acetate ion: if the hydrogen atom is not explicitly given, DFF will treat it as acetic acid. In ring molecules, formal charges are counted towards calculating the aromaticity, the formal charges must be assigned correctly. The formal charges are also used to calculate distributed atomic charges, if the species is charged, the net charge may be distributed among the same atoms but drawn differently (by different bond orders). Typical examples are aromatic rings such as imidazolium (IM), pyridinium (Py), ammonium (NH4+), tetrafluoroborate (BF4+) hexafluorophosphate (PF6-). Moreover, the formal charges are used to build bond orders, add hydrogen atoms, and calculate aromaticity of rings.

The formal charges can be assigned or edited by using DFF user interface. An automatic program is implemented to enable quick assignments. In this program, a *hard-coded procedure is implemented to assign the formal charges using bonding information including number of bonds, ring sizes, atoms in the ring, directly attached elements and numbers to assign the formal charges for special cases. For atoms that are not defined by the hard-coded program, the number of bond electrons (depending on bond orders) and valence electrons are used to calculate the formal charges.*

If a model has been assigned formal charges, the program can preserve the values unless instructed to reset all values. A command can be used to remove all formal charges. These functions provide convenience for editing the formal charges by users.

### Aromaticity

With the ring sizes and bond orders, the aromaticity of atoms is calculated by the 4N+2 rule.

### Assumptions and Recomendations

As discussed above, the formal charges, bond orders, hydrogen atoms are dependent. There is no simple way to fix the order of procedure.

If a model is built from scratch by using DFF, the connectivity and bond orders can be drawn and edited first, and then the hydrogen atoms. With the molecular topology established, the formal charges can be assigned automatically.

For a model imported from other software, try to get the data with full connectivity (with hydrogen atoms) and formal charges. If the connectivity is not given or incomplete, use DFF to build the bonds, which will add hydrogen atoms and calculate bond orders. If full connectivity is given, use DFF to compute bond orders only. This will not change the connectivity but recalculate the bond orders. With the bond orders fixed, assign formal charges.

Note that the formal charge assignment is designed for the most probable (often the most stable) cases, which may not be appropriate for your purposes. Examine the formal charges carefully and edit the DefaultFormalCharge.dat file to customize the assignments.

## 3-4. Atom Type Definitions

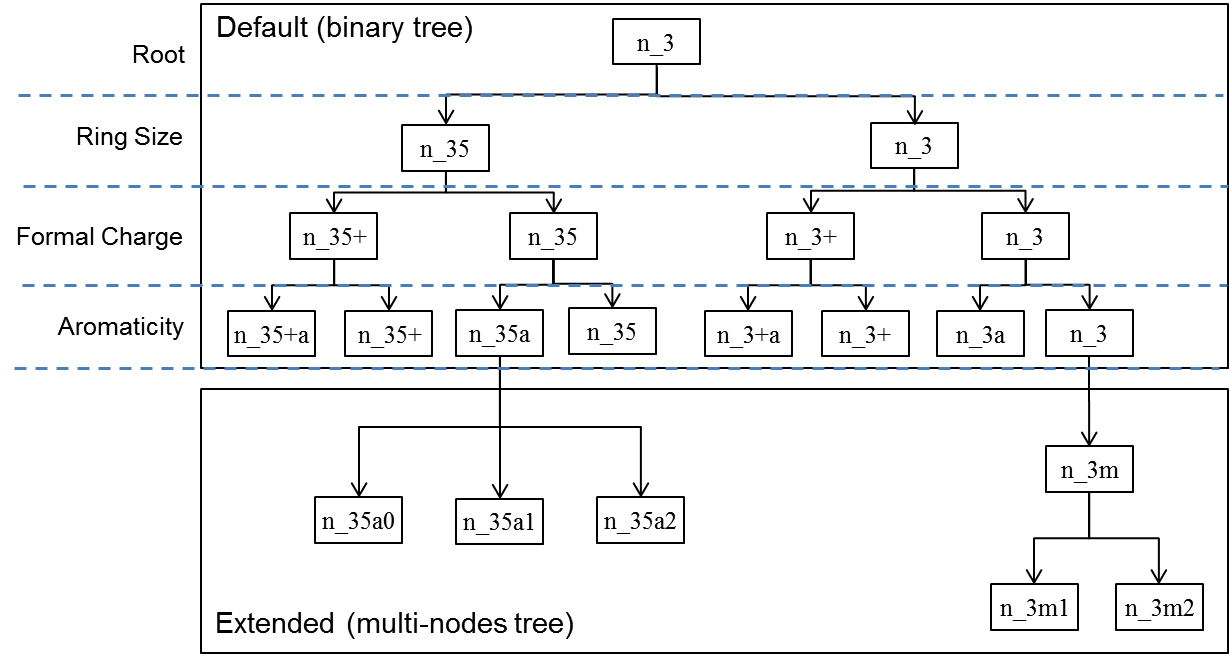
Atom type represents a subspace in the chemical space. The more specifically defined an atom type, the smaller the subspace it represents. The applicable scope is related to the accuracy of the parameters. Generally speaking, broader coverage implies less accurate predictions, and vice versa. Therefore, an optimal balance between applicable scope and accuracy can be adjusted by defining atom types properly.

Definition of atom types also impacts the extensibility of the forcefield. Traditionally, atom types are defined rather arbitrarily. Different developers may define atom types differently, and there is no global rule. This makes it very difficult to extend a forcefield, as adding a new atom type may jeopardize existing definitions. The most common problem is that a new atom type may conflict or overlap with existing ones, and parameters defined under previously atom type sets may be lost.

In order to make extensible and transferable forcefields, atom types must be defined properly. For this purpose, a Hierarchical Atom Type (HAT) scheme is designed.[32](#_ENREF_32)

### Hierarchical Atom Type (HAT) Scheme

HAT is a two-level typing scheme consisting of a default definition and an extended definition.[32](#_ENREF_32) The default definition for an atom considers the attributes of the atom only. The extended definition contains more information by considering the attributes of its nested neighboring atoms. Figure 1 illustrates the hierarchy of this typing scheme using a nitrogen atom as an example.



### Default Definition

A default definition starts with a three-character root type in which the first two characters are reserved for a given atom's element symbol. If the element symbol is a single character, it is followed by an underscore. The third character is an integer that corresponds to the coordination number or the number of chemical bonds connected to the given atom. For example, n\_3 (Figure 1) represents a nitrogen atom with three chemical bonds.

From the root type, a binary tree data structure is used to classify atom types in three levels:

1)     Small ring size. If an atom is in a small ring (<6), the ring size is assigned. Therefore, an integer at the fourth character is a flag indicating the atom is in a small ring (e.g., n\_35).

2)     Formal Charge. If a formal charge is assigned to an atom, it is flagged by a sign (+ or –) followed by an integer (1 is omitted) or a character representing a fractional formal charge:

*d:Double(1/2)   
t:Triple(1/3)   
q:Quadruple(1/4)  
p:Pentuple(1/5)  
h:Hextuple (1/6)*

3)     Aromaticity. If an atom is in an aromatic ring, the reserved character “a” is appended.

For example, the atom type c\_35+pa represents a carbon atom with three bonds in a five-member aromatic ring bearing a +1/5 formal charge.

The default definition provides the most essential information for an atom and has the following properties: (1) the definition is complete; that is, any atom in any environment has an assigned atom type; (2) each atom type is uniquely defined because only two child nodes are present at each node of the tree, and the choices are mutually exclusive; and (3) the atom types are defined inheritably, that is, each atom subtype has only one parent atom type, and each atom subtype represents a subset of the chemical space represented by the parent atom type.

The default definition is implemented in DFF and identified as “Default”. It is hard-coded and cannot be changed by users.

### Extended Default Definition

The extended definition starts from a default definition. The attributes of neighboring atoms are then collected, and the search for neighbors can be nested. A multi-node tree data structure is used for extended definitions because the number of attributes is sufficiently large to differentiate atom types at a given level. An extended definition is specified by a set of scripts, which can be prepared using a text editor. For example, the following script defines a given amide nitrogen with two hydrogen atoms as n\_3mh2, which is extended from the default atom type n\_3:

Define n\_3m2h  
    N [–\*] [–H] [–C[=O]]      
    atom 1: co=3, ar=no, rs>5, fc=0  
    atom 2: co=?, elem = C, Si  
    atom 3: co=?  
    atom 4: co=3  
    atom 5: co=1|  
End

In this example, the first line indicates the atom type name. The second line defines the topologic substructure of the atom type to be defined. The substructure is defined using element symbols and bond orders. Element symbols can be a wild card symbol (\* or ?). Bond orders are defined as single, double, partial double, triple and/or any bond order. In this example, the atom element (N) maps out the connectivity for the amide nitrogen atom, which has three single bonds: [–\*], [–H], and [–C[=O]]. Brackets indicate substructures that can be nested. From lines 3 to 7, attributes for each atom in line 2 are provided in order of appearance. The coordination number (co) must be listed, but “?” indicates that the coordination number is ignored. Other characters include aromaticity (ar), ring size (rs), and formal charge (fc).

The same exclusive and inheritable properties of default definitions are implemented for extended definitions. Each child node has only one parent node, and sibling nodes are exclusively defined. The software tests the following two conditions to enforce exclusive and inheritable properties: a) each atom type must represent a subset of the chemical space that its parent atom type represents, and b) the chemical spaces represented by sibling atom types must not overlap.

Extended definitions must be prepared as a definition file named **<file name>.ext**. For each extended definition for an atom type, a block of scripts as in the above example should be included. A complete description of the definition file format (.ext) is given in [Chapter 6](file:///C:\Users\huai\AppData\Roaming\Microsoft\Word\UserGuide71.docx#6.5 Atom Type DEFinition (DEF, EXT) File Format) of this document. This file must be placed in the **<DFF Installation Folder>/data** folder. DFF can help user prepare definition scripts and check if definitions satisfy the exclusive and inheritable property.

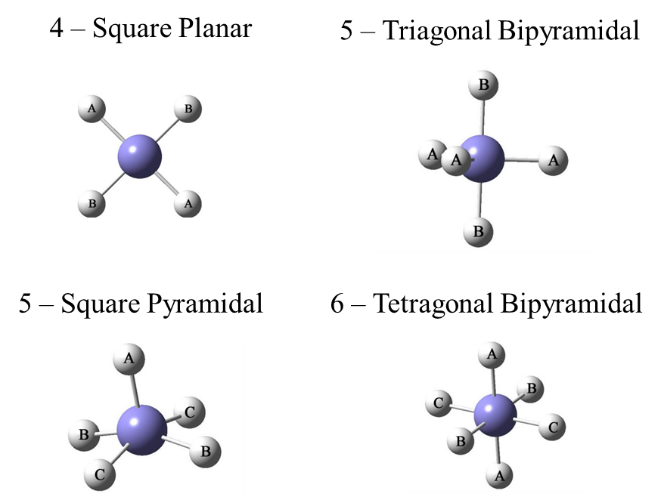
### Complet Scripting Definition

In addition, scripts can be used to prepare a set of definitions without extending the default definitions. This definition is completely flexible, it does not rely on any predefined rules, can be defined at any level of sophistication. The definition is written as scripts that uses molecular topologic information and atomic properties. The definition is particularly useful when importing a forcefield. The extension name for a Free Definition file is \*.def.

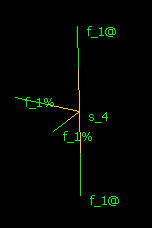
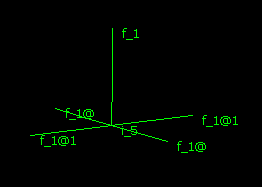
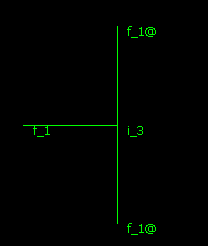
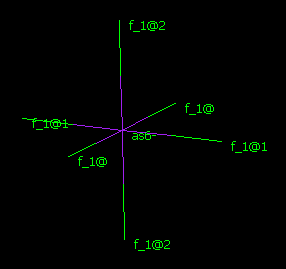
### Atom types of hypervalence molecules

Hypervalence means some atoms in the third and above period on the periodic table do not obey the octet rule. This is common in metal-ligand complexes and metal oxides. In classical forcefields, the chemical bonds are assumed to be stable. Although the chemistry of bond breaking and formation cannot be described by normal forcefield, it is nevertheless significant to study the physical properties of metal-ligand complex or metal oxides.

The hypervalence structure imposes a difficulty for classic forcefield method. As shown in the following picture. A complex of an atom (M) bonded with a number of the same ligand atoms (L), the bond angles are different depending on the relative positions. The angles are 90, 180 degrees for T-shaped (not shown), Square-planar, Square-pyramidal and Octahedral structures, and 90, 120 and 180 degrees for the See-Saw (not shown) and Trigonal-bipyramidal structures. However, since the ligand atoms bear the same atom types, a normal bond angle function, either harmonic, cosine, or polynomial, cannot describe these bond angles as minimum energy. Special function must be introduced, which imposes a requirement for the simulation engines.



In order to use common function forms of forcefield, an alternative solution is to modify the atom types using structural information. The modifier is a special character optionally plus a numerical index. Two special characters, “@” and “%”, are found to be sufficient for common uses. Each modifier represents a set of structural equivalent atoms. Therefore, combination of the same modifiers represents an angle near 180 or 120 degrees, combination of different modifiers represents an angle near 90 degrees, as shown by the examples below.



Octahedral See-Saw Square-pyramidal T-Shaped

### Apparent and Equivalent Atom Types

An equivalence table (EQT) is used to reduce the number of atom types (and consequently the number of parameters). Apparent atom types (AATs) are mapped to equivalent atom types (EATs) to identify parameters in forcefield files. This mapping is done by using an equivalence table.



Energy terms in a forcefield are defined as functions of coordinates of one, two, three or four atoms, with each atom represented by an atom type. Because each atom type carries environmental information for other atoms in the same energy term, the required information for each atom type depends on the number of atoms in an energy term and the position of the atom in an energy term. The more atom types used in an energy term, the less specific the atom types.; For three-atom and four-atom energy terms, the atom type definitions of the central atoms are more specific than those of the peripheral atoms.

EQTs are constructed during the process of parameterization, with more specific atom types used only when necessary. The most specific atom types are those defined for one-atom energy terms (nonbond), with two-atom (bond), three-atom (angle) and four-atom (dihedral) terms becoming less specific. The least specific atom types are those used for peripheral atoms in three-atom and four-atom energy terms.

## 3-5. Overview of Parameterization

### Forcefield type and Atom Tying Rules

To develop a forcefield, the first task is to choose a forcefield type and a set of typing rules. This is usually decided based on intended applications and availability of simulation software. The following forcefield types are supported by different simulation engines.

|  |  |
| --- | --- |
| AMBER | LAMMPS, *GROMACS* |
| CHARMM | LAMMPS, *GROMACS* |
| DREIDING | LAMMPS |
| TEAM | LAMMPS |
| CFF | LAMMPS |

Because the atom typing rules determine the transferability and accuracy of a forcefield, selection of a set of typing rules is a key factor in forcefield development. Typing rules can be set either using the programmed *Default* typing rules or by preparing a script file of typing rules. The *Default* typing rules are modestly fine defined, suitable for most needs, when the definition of atom types is not fine enough, indicated by parameters are not transferrable, new atom types should be introduced.

### Training Set and Data

Forcefield works on assumption of energy additivity. Therefore, fragments of a molecule are used to parameterize forcefields. The fragment molecules must satisfy two criterions: a) contain all required energy terms, and b) as small as possible to avoid unnecessary correlations and computational expenses. An algorithm is developed and implemented in DFF to automatically find all fragments according to the energy terms required. Given the forcefield type and atom typing rules determined, the required energy terms are fixed for a set of target molecules. From the energy terms, the target molecules are scanned to find the minimum fragments. The fragments are cut from the target molecules and saturated by adding hydrogen atoms. The obtained fragments are filtered to remove redundant ones. If the typing rules are changed, the fragmentation process needs to be repeated, the new fragments need to be added to the training set.

Given the training set, next task is to generate data for developing forcefield. Quantum mechanics calculations are powerful to produce large amount of information in terms of energy, energy derivatives, electrostatic potentials and electron densities, atomic partial charges, etc., which can be used to derive forcefield parameters. However, how to sample the molecular conformations is a critical issue. Only with sufficient sampling the conformational spaces, the forcefield derived can be expected to represent the conformers well.

Using the small fragments implies that long-range (nonbond) interactions are to be considered separately. In addition, the nonbond interactions cannot be represented well in QM calculations. Even with the electron correlations are considered, the polarization and multibody effects are very difficult to be captured by using QM calculations. A common practice is to use experimental data of molecular liquids to optimize the nonbond interaction parameters. Therefore, another training set of liquid molecules, which often overlaps with the training set of fragments, is required. The experimental data usually includes equilibrium data such as liquid density at given temperature and pressure, heat of vaporization energy, vapor-liquid-equilibrium curve, as well as dynamic data such as diffusion coefficients and viscosity.

### Parameterization Methods

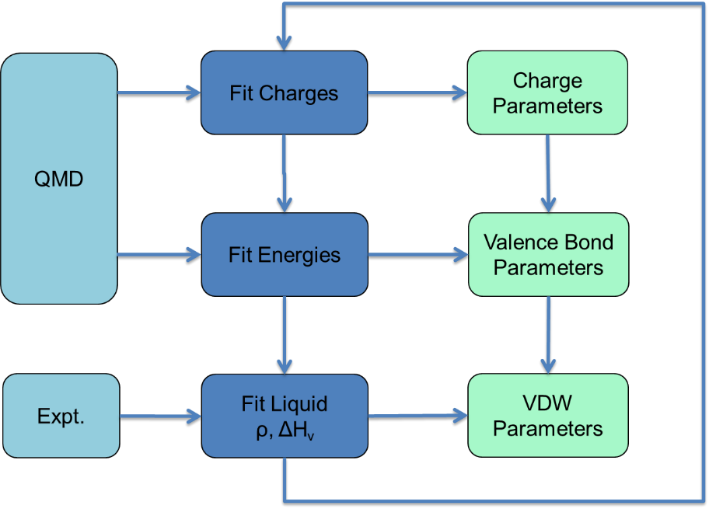
A common method of parameterization is to empirically fit a set of observable properties in a trial-and-error manner. Because the number of adjustable parameters is usually small, a common problem in the empirical method is *under fit*, the parameters work well for the targeted properties, but fail to predict properties outside of the scope. The solution is try to fit as many properties as possible so that the parameters are likely to be transferrable. This method has been used for developing general forcefields for common small molecules,[41-44](#_ENREF_41) and it is still very useful for deriving nonbond parameters[9](#_ENREF_9),[10](#_ENREF_10),[16](#_ENREF_16),[23](#_ENREF_23). However, for general parameterization purposes, it is practically infeasible.

To fit hundreds or even thousands parameters is possible by using QM calculations to generate large number of data.[17-23](#_ENREF_17),[43](#_ENREF_43),[45](#_ENREF_45) In one sense, the “observable” properties such as bond lengths, angles and vibrational frequencies, can be obtained by QM calculations. A fundamental approach is to fit the potential energy surfaces in terms of energies, the first and second derivatives of energies of different configurations or conformations. By adjusting the parameters to fit the potential energy surfaces, the resulting forcefield can be used to predict the “observable” properties and the parameters are naturally transferrable.

However, to fit the potential energy surfaces is a difficult task. One cause is the empirical functions used in forcefield, are often inadequate to accurately describe the complex energy surfaces; another problem is incomplete sampling of the potential energy surfaces, which results in a very small region; the third reason is that the number of parameters may be too many due to use of redundant internal coordinates, therefore, over-fit could be a problem.

### Systematic Procedure

Forcefield parameters can be divided into three general parts: charge parameters (electrostatics), valence bond parameters, and van der Waals (VDW) parameters. The combination of electrostatics and VDW terms are also called non-valence or non-bond energies because they described interactions between atoms that are not directly bonded. Because all three parts are coupled, it difficult to fit them simultaneously. A practical method is to fit these three parts iteratively until all parameters converge, as illustrated in the following chart:



As shown in the figure, charge and valence terms are derived based on QM data, and VDW parameters are derived from experimental data. Data is calculated or obtained for a set of molecules called the training set. In the following sections, we will explain how to collect a training set, prepare QM and experimental data, and make initial parameters, as well as how to derive charge, valence and VDW parameters.

## 3-6. Automatic Parameters

DFF has the functionality to make automatic parameters for two purposes: to provide initial parameters for fitting and/or to provide default parameters used to approximate forcefields. Similar to the methods used in UFF forcefields[46](#_ENREF_46), automatic parameters are calculated based on a set of base parameters specified for default atom types. Forcefield parameters for bond stretches, bond angle and torsion angle distortions, out-of-plane angle distortions, and VDW parameters are calculated based on a set of rules.

Base parameters are stored in the file AtomType.dat, which is located in the Data directory. You can modify parameters through the user interface or by directly editing the data files.

Bond Stretch

Two variables - reference bond length and force constant - are calculated:

where *ri* is the bond radii of atom i and *Z\*i* is the effective charge of atom *i*. Both are basic atomic parameters.

Bond Angle Distortion (i-j-k)

Three variables are calculated: reference equilibrium angle, force constant and multiplicity of energy minima. The force constants are:

where

The multiplicity n and the reference equilibrium angle are fixed depending on the preferred geometry of the central atom type j:

|  |  |  |
| --- | --- | --- |
| Pref. Geom. | n |  |
| SQUARE | 2 | 90, 180 |
| SQPYRAMIDAL | 2 | 90, 180 |
| BIPYRAMIDAL | 6 | 90, 120, 180 |
| OCTAHEDRAL | 2 | 90, 180 |
| OTHERS | 1 | various |

Dihedral Angle Torsion (i-j-k-l)

Three variables are calculated: reference torsion angle, energy barrier height and multiplicity of energy minima. The torsion energy barrier height is derived by:

where , are intrinsic torsion barrier parameters for atoms j and k and is the total number of torsion angles about bond (j-k).

The multiplicity n and the reference torsion angle are determined by the preferred geometries of center atoms j and k:

|  |  |  |  |
| --- | --- | --- | --- |
| J | k | n | phi0 |
| TETRAHEDRAL | TETRAHEDRAL | 3 | 180.0 |
| TETRAHEDRAL | PYRAMIDAL | 6 | 0.0 |
| TETRAHEDRAL | PLANAR | 6 | 0.0 |
| TETRAHEDRAL | BENT | 3 | 180.0 |
| PYRAMIDAL | PYRAMIDAL | 6 | 180.0 |
| PYRAMIDAL | PLANAR | 6 | 180.0 |
| PYRAMIDAL | BENT | 6 | 180.0 |
| PLANAR | PLANAR | 2 | 0.0 |
| PLANAR | BENT | 6 | 0.0 |
| BENT | BENT | 3 | 180.0 |

Out-of-Plane Distortion (i-j-k-l, where j is the central atom)

Out-of-Plane distortion is defined only when the preferred geometry of atom j is PYRAMIDAL or PLANAR. The reference angle is always 0.0, and force constant is 5.0 or 0.0, respectively.

VDW Interactions

Two parameters are provided: the VDW well depth and diameters in terms of the Lennard-Jones 12-6 function:

Both and  are basic atomic parameters.

Atomic charge parameters are estimated using charges calculated with the QEQ[47](#_ENREF_47) method.

## 3-7. Quantum Mechanics Data

The importance of preparing quality quantum mechanics data (QMD) cannot be overstated. Generally speaking, if the fit itself is successful - which can be verified by checking the fit merits (squares of displacements, variances of parameters) - the quality of resulting forcefield is mostly determined by the quality of the underlying data. It is therefore very important to provide complete, quality data in order to develop a successful forcefield.

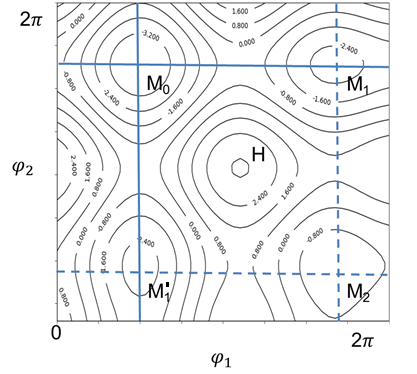
### Sampling Conformational Energy Space

To derive forcefield parameters for a molecule using QM method, at least one optimized structure must be provided, and the calculations should include total energy, gradients, the Hessian matrix (the second derivatives of energy), and atomic charges.

Different structures of a molecule in addition to the minimum energy structure can be included to sample potential energy surfaces. In most cases, the interested structures are conformational isomers, which can be generated by rotating bonds measured by dihedral angles. Although structures generated by distorting bond lengths and angles could also be included, it is not necessary in forcefield development because the anharmonic behavior is not significant under conditions of classic simulations. Many forcefield types do not support the anharmonic terms. Even for forcefield type such as CFF that includes the anharmonic terms, those terms can be well estimated from the harmonic term (see TEAM type). Therefore, the distortion sampling usually does not include bond lengths and angles.

The conformers can be calculated by rigid or relaxed samplings. In rigid sampling the correlations among different internal coordinates are ignored, the dihedral angles are changed while all other internal coordinates are fixed to their equilibrium values. In relaxed sampling, the correlations are taken into consideration, for a set of dihedral angles specified, all other internal coordinates are relaxed. Although more computational resources are required in the relaxed sampling, the obtained potential energies are near minimum energy states, the forcefield parameters obtained by fitting the potential energies are more reliable.

To sample the conformational space efficiently is not trivial for a molecule with multiple rotatable bonds. A systematical scan of entire surface quickly becomes unmanageable. For example, for a molecule with 3 rotatable bonds, if each rotation needs to calculate 10 points in corresponding dihedral angle, a total of 1000 minimizations are required. One may use the random sampling to pick up certain number of points and then find the local minima by energy minimization. Another approach, called orthogonal sampling, may find not only the minimal energy points, but also the saddle points. In the orthogonal sampling, the rotatable bonds are *scanned one after another, not simultaneously*, *from identified minimum energy points.* Although the conformational space is not completely sampled, the data points will likely cover the important lower energy regions. As illustrated by the following figure of a 2-D energy surface. Starting from a minimum energy point M0, varying φ1 reaches minimum energy point M1 and varying φ2 reaches another minimum energy point M’1 (solid lines). Moreover, varying either φ1 from M’1 or φ2 from M1 both reaches M2 (dash lines). By doing so, the local minima and vicinity of saddle points are scanned, but not the high-energy region (H).



### Sampling Intermolecualr Energy Curve

The QM method can be used to probe intermolecular interactions by calculating structures and energies of molecular dimers. DFF supports this type of calculation and provides a method to generate dimer structures automatically. However, it must be noted that using QM methods to characterize weak VDW interactions is a challenging task. Even if calculations can be performed accurately with high levels of correlation and a complete basis set, results still represent interactions between two molecules in a vacuum. To get accurate van der Waals parameters for molecules in condensed phases, condensed phase properties must be used to optimize the forcefield.

### QMD File

QM calculations can be done by using any quantum mechanics software package. In order to support different software packages, a standardized file format was designed for storage of QM data. This file format is the QMD file format (more details in **Chapter 6**) and uses the extension \*.qmd. QMD files are written in simple text file format, and is the only interface between QM calculations and DFF software. DFF can also convert output files generated by Gaussian to QMD files.

Each QMD file corresponds to one molecule, must contain one optimized structure, and may or may not contain conformational isomers. The optimized structure, contains essential data: energy, first and second derivatives of energy, atomic charges (Mulliken or ESP), and electrostatic multiple moments. Conformational isomers, if included, do not require second derivatives. Calculation protocols, methods and basis sets are embedded in the QMD file. The QMD file also contains molecular topology information (connectivity, formal charges, atom types, etc.) which are not required for input, but are generated and used by DFF for parameterization.

There are no particular requirements on the QM methods (HF, DFT, MP2, etc.) and the basis sets used in calculations. Since the ultimate criterion is agreement with experimental data, any QM method that can produce good agreement with experimental data can be used to generate QMD. The DFT method with B3LYP functionality and a modest 6-31G(d) basis set is adequate for describing structural and energetic properties of most organic molecules,[48](#_ENREF_48) and is used as the default QM calculation protocol in DFF.

Before QMD data is used for parameterization, it is a good practice to check data validity. Usually this can be done by comparing relative energies of different structural isomers in a QMD file. The lowest relative energy should correspond to the global minimum with a Hessian matrix calculated. Relative energies should not be too large, and charge parameters should be present in the data set.

## 3-8. Fit QM Data

### Methods

Atomic charges must be calculated and assigned first. Atomic charges can be calculated using QM methods (ESP charges or Mulliken Charges) or using the charge equilibrium method. Direct Forcefield uses a charge equilibrium method similar to the published QEq method. The basic parameters used to calculate charge distribution are the ionization potentials (IP) and electron affinities (EA) of relevant atoms, which are collected from literature and included in the *Elements.dat* file.

Charge parameters are derived from QM atomic charges by solving a set of linear equations. Suppose a set of atomic charges are given in a vector Q for M atoms:

. (1)

The charge parameters can then be calculated by:

, (2)

where P is the charge parameter vector in N order:

(3)

and A is an assignment matrix:

(4)

for each atom i and nearest neighbor j:

, (5)

and where all of the other elements are zero. Assignment matrix elements are determined using ATC and BIC parameters as expressed in Eq. (1).

In general, the number of atoms (M) is greater than the number of charge parameters (N). Therefore, Eq. (3) does not have a precise solution. The singular value decomposition (SVD) method[49](#_ENREF_49) is used to find the best solution as determined by the least-squares approach.

#### Nonlinear Terms

The Levenberg-Marquardt method[49](#_ENREF_49) with constrained minimization technique is used to fit QMD in order to derive valence terms. At this stage, charge and VDW parameters are frozen, as charge parameters should have already been derived from QMD, while VDW parameters are initialized using a set of default parameters provided by DFF.

The least squares problem is written as:

                (6)

The first term is the normal least squares expression of input data and parameter-dependent calculated data. The weighting factor σi differs for different data and can be related to the Boltzmann factor for energy data. The second term is a penalty term in which parameters are restrained to preferred values. The penalty factor *Pj* is an adjustable factor that controls the constraints. The above equation is reduced to the normal least-squares equation by setting all penalty factors to zero.

#### Linear Terms

In addition, the SVD method[49](#_ENREF_49) is used to reduce redundant problems for linear terms. By limiting variables to linear terms only, the least-squares equation is reduced to a linear equation:

                (7)

If some parameters are linear combinations of others, matrix A is singular or numerically very close to singular, and the SVD method is suitable for solving this problem.

### Charge Parameters

A common way to derive charge parameters from QM data is to convert QM atomic partial charges to charge parameters. Atomic partial charges are calculated from QM wave functions. ESP charges are obtained by fitting to electrostatic potentials, and are thus dependent on how the fit was conducted. ESP charges for atoms in a bulky molecule are typically difficult to obtain as they may be insufficiently sampled. On the other hand, atomic partial charges determined using population analyses such as the Mulliken method are stable. Both types of atomic partial charges have been used in forcefield development.

There are three types of charge parameters: atom based, atom-type based and bond-type based. Atom based charge parameters are identified by atom names (usually element symbol plus sequence number) assigned by the program. Atom-type based parameters are defined by atom types.  Each atom type has a charge parameter, and all atoms that have the same atom type share one parameter. The bond-type based parameter is also called bond charge increment, which is specified by atom types of two bonded atoms (δt(i),at(j)). This parameter represents charge relocation from atom j to atom i. For example, if δat(i),at(j) = 0.5, that means atom i gains 0.5 electron, and atom j loses 0.5 electron. For any atom i, the net charge is a sum of all relevant bond charge increments.

(8)

Atom based charge parameters are difficult to transfer; bond-type based charge parameters reflect the interactions of nearest neighbors and ensure charge neutrality of the entire molecule; and atom-type based charges provides the flexibility to assign net charges. DFF uses atom-type and bond-type based charge parameters in developed forcefields. This scheme is flexible to represent any molecules, ionic or non-ionic.

### Valence Parameters

Valence parameters are the largest component of a forcefield, making up a majority of all parameters. For a forcefield with N atom types, the number of valence parameters are on the order of O(N2), O(N3) and O(N4) for bond, angle and dihedral angle terms respectively. Although using equivalence tables can reduce the number of atom types and parameters effectively, the number of valence parameters remains very large.

Therefore, optimization of these parameters to fit QM data is a high-dimension optimization problem. In principle, optimization can be achieved using the least-square approach, but in practice it is usually a very difficult task. The fit can be both over-determined (more parameters than data) for some properties and under-determined (more data than parameters) for others. In addition, the correlations and dependencies among parameters make the problem worse.

DFF has been designed to solve these problems using various techniques. An expert system that examines the data, decides an appropriate fitting procedure, monitors the fitting process and adjusts fitting options has been implemented to guide the automated parameterization process.

It may be necessary to adjust the fitting options or initial parameters in order to get a better fit for very complicated systems. Adding controls to the initial parameters is usually very helpful, especially when data is limited. For example, freezing some parameters to reduce the number of adjustable parameters is usually an effective approach to improving fit quality.

### Molecular Dimers

DFF can determine nonbond parameters by fitting molecular dimers in which inter-molecular interactions are represented by VDW and electrostatic terms only. To fit VDW parameters, a number of configurations of the dimer should be generated and calculated with relatively high-level QM calculations (MP2 with sufficiently large basis set). DFF automatically freezes other parameters and fits potential energies of dimers in different configurations using the Levenberg-Marquardt method[49](#_ENREF_49).

It should be noted that this approach might not work well for predicting physical properties of molecules in condensed phases. Although in principle the VDW parameters can be derived by fitting to QM data calculated for molecular dimers, this approach has the following limitations. a) it is very difficult to accurately describe weak VDW interactions using QM method, high level of theory and an extensive basis set must be used. b) the dimer is too small to represent molecular interactions in condensed phases. The result energetic data only represents molecular interactions in gas phase, which in many cases can be significantly different from those found in the condensed phases.

### Specific Properties

Fitting a few specific properties is an alternative way to optimize a forcefield. This method is useful when the number of adjustable parameters is small such as in the situation a forcefield is extended to cover new molecules while most parameters are transferred and fixed.

The least-square Levenberg-Marquardt method requires the first derivatives of the merit function with respect to the adjustable parameters. For a given property A, it is:

The property A may or may not be an analytical function of the parameters. Therefore, a numerical differentiation can be calculated, using e.g.a two-point formula:

In this formula, the first-order errors cancel so the slope of the secant lines differs from the slope of the tangent line by an amount that is approximately proportional to δa*2*.

With the first derivatives, fitting can be automated. This method can be used to optimize parameters of a forcefield for any properties that can be calculated using the forcefield. However, the computational expenses can be very high. A critical step in employing this method is to select a few, most-sensitive parameters. For some properties (*e.g.* conformational energy profiles and structural parameters), the most sensitive parameters are usually obvious: the parameters of the dihedral angle, bond length and bond angle. DFF has the functionality to fit structures (bond lengths and angles) and conformational energy profiles using the numerical derivative method.

For example, torsion profiles are calculated for selected internal rotations using constrained minimizations in which scanned dihedral angles are fixed at certain values while other degrees of freedom are relaxed. QM calculated values are compared with forcefield calculated ones, and then a least-squares problem is solved to optimize the relevant torsion parameters. The following figure shows the internal rotation energy profiles for diethyl ether (C–C–O–C). The forcefield results (dots) are in excellent agreement with DFT data (curve).

## 3-9. Fit Properties of Liquid

DFF can be used to optimize VDW parameters automatically. This is done by combining a molecular dynamics simulation with a least-squares fit procedure.

### Perturbation Theory

The statistic mechanical perturbation theory is used to enable the automation.[36](#_ENREF_36),[38](#_ENREF_38),[50](#_ENREF_50) Assume a small change to the adjustable parameters:

                               (1)

The resultant change to the potential energy can be regarded as a perturbation:

                                                                                 (2)

Based on the perturbation theory, the configuration integral of canonical ensemble consisting of N particles can be written as:

(3)

where the brackets indicate unperturbed (reference state) ensemble average. For any physical property A, we have:

(4)

For example, the internal energy and pressure of ensemble average are given by:

(5)

(6)

The above equations indicate that the change in potential energy and pressure with the perturbation can be estimated by averaging the perturbation energy and virial in the reference state obtained by NVT simulation. The derivatives of an ensemble averaged property <A> (energy or pressure) with respect to the parameter *ak* is then:

(7)

If the quantity can be analytically evaluated, such as the pressure and internal energy in canonic ensemble, the derivative can be obtained in one simulation. Subsequently, the parameters can be optimized by the least-square method automatically. Notice that in the fluctuation framework[51](#_ENREF_51), similar requirement is needed.

In order to obtain transferable parameters, several molecules containing the same atom types should be parameterized simultaneously. For example, the following table shows molecular liquids that can be used to parameterize non-valence parameters for alkanes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **T** | **HOV** | **Den** |
| cyclohexane | 293.2 | 8.0 | 0.779 |
| 2\_2\_4\_trimethylpentane | 303.2 | 8.4 | 0.684 |
| isopentane | 293.2 | 6.0 | 0.620 |
| ethane | 184.5 | 3.5 | 0.546 |
| propane | 231.1 | 4.5 | 0.581 |
| 2\_2\_dimethylhexane | 303.2 | 8.9 | 0.687 |
| 2\_5\_dimethylhexane | 303.2 | 8.9 | 0.685 |
| pentane | 293.2 | 6.4 | 0.626 |
| 2\_methylheptane | 303.2 | 9.5 | 0.690 |
| butane | 298.2 | 5.3 | 0.573 |
| octane | 298.2 | 10.2 | 0.699 |
| methane | 111.1 | 2.0 | 0.424 |

The table lists temperature, heat of vaporization and equilibrium density of the relevant molecular liquids.

The uncertainty in ensemble average has a direct impact to the fitting quality. Large uncertainties associated with short simulation time lead to oscillation in the fitting parameters. In such situations, extending the simulation time is necessary.

For polar molecules with relatively large atomic partial charges, the electrostatic terms dominate the nonbond interactions. If the VDW parameters cannot be optimized to fit empirical data, the atomic charges may not be appropriate, even if the charges are derived by fitting QM electrostatic potentials. The reason is that the separation of electrostatic and VDW parameters is for convenience but not scientifically solid. As in principle, all intermolecular interactions are electrostatic in nature. Comparing the charge values with those in successful forcefields might be helpful. If necessary, consider using restraints in the ESP fit and re-parameterizing the charge parameters.

Similarly, if the valence parameters are incorrect, it could be very difficult to optimize the VDW parameters. Even if a fit seems to be successful, the VDW parameters might not be physically meaningful. Careful examination of the intramolecular properties such as bond lengths, angles and dihedral angles is a key to solve this problem.

### Temperature Dependent Parameters

Strictly speaking, forcefield is not potential energy function because of the empirical (non-polarizable) functions deployed. The nonbond interaction is found to be dependent on temperature because of the polarization effects.[35](#_ENREF_35) One solution is to scale the dispersion coefficient using the following form of [Lennard-Jones (LJ-12-6) function](#_1._Lennard-Jones_(12-6))

(8)

(9)

where the scaling factor is temperature dependent, which can be written as a linear function,

. (10)

The reference temperature is conveniently fixed at 298 K.

Using the expression of [Lennard-Jones (LJ-12-6) function](#_1._Lennard-Jones_(12-6)) used in DFF, the well-depth and radius parameters are expressed as:

(11)

(12)

It is found that the scaling factor is rather generic, it takes only two values depending on the hybridization state of the atom type:

|  |  |  |
| --- | --- | --- |
| Atom type | (/K) | Description |
| c\_3a | 5.0E-05 | C, aromatic |
| c\_3ac | 5.0E-05 | C, aromatic, bond to C |
| c\_3aco | 5.0E-05 | C, aromatic, with carbonyl |
| c\_3o | 5.0E-05 | C, sp2, in ketone |
| c\_3oh | 5.0E-05 | C, sp2, in aldehyde |
| c\_3oh2 | 5.0E-05 | C, sp2, in formaldehyde |
| c\_4 | 1.4E-04 | C, sp3 |
| c\_4h | 1.4E-04 | C, sp3, with 1 H |
| c\_4h2 | 1.4E-04 | C, sp3, with 2 H |
| c\_4h3 | 1.4E-04 | C, sp3, with 3 H |
| c\_4o2 | 1.4E-04 | C, sp3, bond to 2 O |
| c\_4oh2 | 1.4E-04 | C, sp3, bond to O, with 2 H |
| c\_4oh3 | 1.4E-04 | C, sp3, bond to O, with 3 H |
| h\_1 | 1.4E-04 | H |
| o\_1 | 5.0E-05 | O, sp2, in carbonyl |
| o\_2 | 1.4E-04 | O, sp3, in ether |

The scaling factors are specified by a global parameter file: *<dff-root>/data TemperatureFactor.dat,* which subject to further optimizations.

## 3-10. Simulations

A general procedure of using a forcefield made by DFF is as follows.

### Assigning a Forcefield

Prepare or import a simulation model. DFF can be used to build 3-dimensional cubic liquid boxes and liquid interfaces, but this functionality is intended primarily for parameterizations and validations. For complex molecular systems, an external software package can be used to build the simulation models. DFF reads most popular file formats such as .pdb, .mol2, and .car files.

It is important to have a consistent assignment of topologies and formal charges in order to assign atom types. If the coordinates of atoms are given with reasonable precision, DFF can automatically assign topologies and formal charges. This method works well for common organic molecules.

With topologies and formal charges assigned, the next step is to assign atom types and forcefield parameters using TEAMFF. Multiple forcefields in the same forcefield type can be used together. However, the multiple forcefields are called in the order as specified. It is important to know that all valence parameters for one molecule must be taken from one forcefield, which means that there is no mixing of intra-molecular parameters from different forcefields, only the inter-molecular parameters (the nonbond parameters) may be taken from different forcefield and used by the combinational rule. In operation, DFF scan all molecules again the selected forcefield, if any parameters are missing for a molecule in the forcefield, the molecule is not assigned and passed to the next forcefield assignment.

When all parameters are found, the atom types and parameters, which may be taken from different forcefield tables, are unified to one forcefield for the simulation model. A new set of atom types which corresponds to the unified forcefield table, is made automatically. The atom types are assigned to the simulation model. The model and the forcefield table can be used directly in simulations.

If any parameters are missing, TEAMFF will report the missing terms, which then need to be parameterized.

### Internal Simulation Engines

DFF contains a built-in simulation engine that can be used to optimize and validate parameters. Simulation jobs can be run interactively so that structures and energies are dynamically updated on the screen.

Two different algorithms, the conjugate gradient and variable metric methods, are implemented in DFF for energy minimization. The conjugate gradient method is based on the Polak-Ribiere method, which is similar to the Fletcher-Reeve method. The variable metric method is also called the quasi-Newton method. Both methods involve calculating the derivatives of the potential energy. Both methods are iterative and make use of the previous history of minimization steps as well as the current gradient to determine the next step. For a quadratic functional form in N dimensions, these methods lead to the exact minimum in N steps. The two algorithms have computer memory different requirements. The conjugate gradient needs to store intermediate data on the order of N, while the variable metric method requires storage on the order of N x N.

Restrained energy minimization can be performed to explore energy profiles. This is done by adding energy terms to the total energy function on selected internal coordinates. The restraint function is a harmonic function in which reference values and force constants can be specified. The restraint energy value is subtracted from the total energy in reported results.

The velocity verlet algorithm is implemented in DFF. The choice of integration step size is important. One must weigh the increased accuracy of using a small step size against the greater length of simulated real time per given execution time when using a larger step size. In most molecular systems, the highest vibrational frequency is that of X-H bond stretching, whose period is approximately 10-14 s. The integration time step should therefore be approximately 1 femtosecond (fs).

A dynamic run consists of three parts: initialization, equilibration, and simulation. Initialization provides an initial position and velocity for all atoms. This step can be skipped for a continuous simulation. Equilibration is used to attains the most probable configuration for a system consistent with a given target temperature and pressure. When equilibrium has been achieved, the average properties of the system (temperature, potential energies and kinetic energies) must be constant. After equilibration, a simulation can be conducted and collected data can be analyzed.

The available simulation types are:

*NVE (micro-canonical) Simulation.* This ensemble is obtained by solving Newton's equation without temperature or pressure controls. Excepting rounding and truncation errors during the integration process, energy is conserved..

*NVT (canonical) Simulation.* This ensemble is obtained by controlling temperature either through direct temperature scaling or via temperature-bath. The volume is kept constant throughout the run.

*NPT (isothermal-isobaric) Simulation.* This ensemble (NPT) allows control over both temperature and pressure. Unit cell vectors are allowed to change, and pressure is adjusted by adjusting volume. This is the ensemble of choice when correct pressure, volume, and densities are important in the simulation. This ensemble can also be used during equilibration to achieve the desired temperature and pressure before changing to a constant-volume or constant-energy ensemble when data collection starts.

Temperature can be modularized using two methods:

*Direct Velocity Scaling*. Temperature can be modified by scaling atom velocities so that the target temperature can is exactly matched. Velocities of all atoms are scaled uniformly. This is controlled by a quantity called a temperature window. When instantaneous temperature is outside of the window, scaling takes place.

*Andersen Stochastic Method*. At intervals proportional to *N2/3*, where *N* is the number of atoms in the system, the velocity of each atom is set from the Maxwell-Boltzmann distribution. This corresponds to a collision with an imaginary heat-bath. The system moves through phase space on a constant energy surface until the velocities are changed. Between these “massive stochastic collisions,” time correlation functions may be calculated as usual.

Pressure is calculated using the virial theorem. Pressure changes can be accomplished by changing particle coordinates and unit cell size in the periodic boundary conditions. Berendsen's method couples the system to a pressure "bath" to maintain a target pressure. At each step, the *x*, *y*, and *z* coordinates of each atom and the cell edges are scaled by a scaling factor. Strength of the coupling is determined by normalized compressibility of the system (user-defined variable). Note that this method changes the cell uniformly so that the size but not the shape of the cell is changed.

DFF uses period boundary conditions in simulating of molecules in condensed phases. A minimum-image model is used. Atom-based or charge-group-based cutoffs can be used in evaluating nonbond interactions. Both methods can be supplemented with cutoff tail corrections. The charge-group based cutoff scheme works very well for organic molecules in which small neutral charge-groups can be defined. As illustrated in the following table, this method yields essentially same results as those obtained using Ewald summation with greatly reduced computational times:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **system** | **cutoff** | **energy** | **density** | **sec./step** |
| Propane N=1320 | 6.5 | -292.5±19.5 | 0.567±0.008 | 0.131 |
| 8.5 | -296.4±19.9 | 0.572±0.009 | 0.200 |
| 10.5 | -298.4±19.2 | 0.572±0.008 | 0.331 |
| 12.5 | -302.2±22.8 | 0.574±0.009 | 0.506 |
| 14.5 | -301.6±20.6 | 0.573±0.009 | 0.759 |
| Ewald | -301.3±18.3 | 0.573±0.007 | 7.566 |
| Ethanol (N=900) | 6.5 | -1156.8±26.8 | 0.734±0.015 | 0.088 |
| 8.5 | -1214.0±26.0 | 0.776±0.015 | 0.142 |
| 10.5 | -1210.4±23.9 | 0.785±0.014 | 0.229 |
| 12.5 | -1215.6±23.6 | 0.785±0.011 | 0.352 |
| 14.5 | -1214.8±23.9 | 0.782±0.013 | 0.524 |
| Ewald | -1207.7±24.3 | 0.780±0.013 | 4.379 |

### External Simulation Engines

DFF is designed to work with different simulation packages. The prepared simulation model and forcefield parameters can be exported to popular simulation engines including LAMMPS, TOWHEE, DISCOVER, AMBER and CHARMM. Because different conventions are used in different programs, DFF translates the atom types to those recognized by other software packages. It should be noted that this translation is simply a one-to-one mapping from the atom types used by DFF to a set of unique string symbols.

To facilitate the simulation setup, *GROMACS* and *LAMMPS* jobs can be launched directly from the DFF after the forcefield parameters are assigned. Energy minimization and NVT, NPT and NVE molecular dynamics can be carried out directly from the user interface.

#### Special Conversions

Because potential functions implemented in *GROMACS* and LAMMPS may be different from published forcefield types, some parameters must be converted to the closest functional form upon export of parameters. The following conversions are used:

#### DREIDING to LAMMPS

Dihedral terms in DREIDING is based on the following function:

The closest function used in LAMMPS is the CHARMM dihedral\_style:

Therefore, parameters are converted as:

For out-of-plane, DREIDING uses two different functions. For non-planar structure:



Otherwise:



The harmonic function is used in LAMMPS (class2 improper style):

Therefore, to get the same force constant at, the following conversions can be used:

## 3-11. TEAMFF Database

Direct Forcefield Database (TEAMFF) is a software that implements and manages the forcefields in DFF. A forcefield database called TEAMFF is created using TEAMFF.

### Direct Forcefield Database

A relational database model is implemented to store and manage forcefields and their parameters. This relational model is illustrated in the following figure:

A blue arrow pointing to a black background

Description automatically generated

Two types of tables are used: the forcefield table (FFT, upper block) and parameter tables (PPTs, lower block). A FFT lists the forcefields stored in the database; its headings are identifiers (FF\_1, FF\_2, …), and its tuples include forcefield names, functional forms, typing rules, references, and flags. Each forcefield in FFT is linked with a PPT that lists the forcefield parameters. In each PPT, the headings are identifiers for parameter entries (entry\_1, entry\_2, …); its tuples include version number, energy terms, atom types, parameters and parameterization notes. The database is specifically designed for forcefield management, and is implemented in-house and operated by scripting commands.

Using a database to manage forcefields provides a standardized interface to add, delete, copy, and extract parameters. Operations can be easily automated, thereby avoiding unintentional human errors. Only new entries are accepted because each entry in the database is unique, and new entries are automatically assigned to a new version number. This version number can be used as a search criterion to recall a certain range of parameters upon retrieval, and is also useful when calculations are repeated using older version parameters.

#### TEAMFF Basic

TEAMFF contains multiple forcefields, each forcefield is a table in the database, completely defined with specific forcefield type (AMBER, CHARMM, CFF, etc.), atom typing rules, and parameters. The independence of forcefields in the TEAMFF database allows developments of high quality forcefields without being constrained by pre-existing parameters. Multiple forcefields in the same forcefield type can be used together. This greatly enhances the extensibility of TEAMFF.

TEAMFF contains a set of basic forcefields as listed in the table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Typing Rule** | **Coverage** |
| TEAM-Basic | TEAM | DFF2012.def | common organics and drug-like molecules |
| AMBER-Basic | AMBER |
| CHARMM-Basic | CHARMM |
| CFF-Basic | CFF |
| CFF12-Basic | CFF(LJ12-6) |
| DREIDING-Basic | DREIDING |

These forcefields were developed *automatically* by fitting a large set of QM data. Hundreds of small organic molecules, synthetic polymer fragments and drug-like molecules were selected as the training set. Quantum mechanics calculations were carried out at the level of B3LYP/6-31G(d,p). The calculated energies and the first and second derivatives of the energy were used to fit the forcefield parameters. The default equivalence table was employed to reduce the number of parameters. Atomic partial charges were derived from ESP charges. Parameterization was validated by comparing forcefield predicted data (*e.g.*, vibrational frequencies; conformational energies; structural properties such as bond lengths, angles, dihedral angles, *etc*) to QM data. However, these parameters were not validated for specific molecules nor for condense phase properties.

The purpose of these forcefields is for providing a starting point of further developments and optimizations. For example, if you find the quality of the basic forcefield of your choice is not optimal for a set of molecules of interest, you can check out parameters from the basic forcefield, optimize the parameters by using additional computational or experimental data, and save the revised parameters in a new table of TEAMFF database, and then use the customized forcefield together with the basic one.

#### TEAMFF General

In addition, TEAMFF contains two default forcefields: TEAM-Default and AMBER-Default. Essential characteristics of these two forcefields are summarized in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Typing Rule** | **Coverage** |
| TEAM-General | TEAM | General.ext | common organics and drug-like molecules |
| AMBER-General | AMBER |

These forcefields are developed using the same typing rules, against same training set of molecules, and have very similar coverage. The difference from the Basic forcefields is that these forcefields are continually improved depending on users’ feedbacks and advances of research. The parameters were obtained by fitting not only QM data but also experimental data.

#### TEAMFF Special

In addition, TEAMFF contains forcefields specially designed for particular applications. These are also high quality, off-shelf type of products, that will be continually improved. Some of the special forcefields are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Typing Rule** | **Coverage** |
| TEAM-IonLiq | TEAM | IonLiq.ext | Ionic liquids |
| TEAM-Zeolite | TEAM | Zeolite.def | Zeolites |
| AMBER-IonLiq | AMBER | IonLiq.ext | Ionic liquids |

These forcefields, like the General forcefields, are developed in two forcefield types, TEAM and AMBER. Depending on the coverage scope, one or both types may be developed. The forcefields can be used alone for the specific coverage, or to be used together with the General forcefield to provide a coverage of mixtures.

#### Usage of TEAMFF

TEAMFF can be used for two purposes. One is to use it for application, another is to use it for development.

To use TEAMFF for application is similarly to use a traditional forcefield. The most important advantage of TEAMFF is that it allows multiple forcefields to be used seamlessly together to achieve high quality and coverage. The forcefields of TEAMFF are selected and applied in an order of precedence. DFF searches for the best set of parameters for a target molecular system, combines the parameters into a unified set of parameters, creates a new set of atom types, assigns the atom types to the molecular system, and finally prepares the forcefield and model files for simulation.

To use TEAMFF for further development is a unique feature. One possible approach is as follows. If a forcefield of TEAMFF is found to be not optimal for specific applications, you can either check out the original parameters and optimized them, or develop new parameters independently, using the parameterization tools. When the optimization or parameterization is done, the new parameters can be check into a **new, customized** forcefield table of TEAMFF. The new forcefield table can be used together with other existing TEAMFF tables in the same forcefield type, as explained above.

### Customized Forcefield Tables

A customized forcefield is like other TEAMFF forcefields, has its forcefield type, atom typing rules and parameters. The customized forcefields can be added to or removed from TEAMFF without affect other forcefields. Since each forcefield is independent and usually focus on a small set of target molecules, to make or optimize a customized forcefield is relatively easier than making a general forcefield that covers a large set of molecules.

The combination of different forcefields is based on the combination rules of nonbond parameters, which are columbic and Lennard-Jones terms. Therefore, in developing a customized forcefield the compatibility of Lennard-Jones and charge parameters should be considered. In principle, these parameters should be derived using the same parameterization protocol.

### Internal Extensibility of Atom Types

Because of the inheritance property of atom types, the forcefield made based on the HAT is internally extensible.[32](#_ENREF_32) Only when existing parameters are inaccurate or insufficient, the extensibility becomes a challenge. The simplified chart below illustrates the relationship between atom types, parameters and chemical space.

A black background with a black square

Description automatically generated with medium confidence

Two levels of atom types, old (To) and new (Tn), are shown in this chart. The new atom types (child nodes) are derived from the old ones (parent nodes). Parameters Po(To), Po(Tn), Pn(To), and Pn(Tn) represent old or new parameters associated with old or new atom types, respectively. The chemical space is divided in four regions, I, II, III, and IV. In the HAD scheme, a higher level of atom-type definitions represents a broader chemical space than a lower level does. For example, Figure 2 shows that To, at a higher level, represents four regions while Tn, at the lower level, represents only regions II and III. Note that forcefield parameters associated with atom types are not necessarily complete. We assume that the old parameters with old atom types Po(To) cover regions I, II, and III, but not region IV. We further assume that region I is the training set used to develop Po(To) so that region I is covered accurately; regions II and III are thus covered based on the transferability of parameters. Given that atom types are completely defined, one of the following two scenarios may sometimes occur:

1) One of the scenarios is represented by region IV. In this case, parameters are insufficient to cover this region in a chemical space represented by the old atom types (To). Thus, new parameters must be derived, and this derivation can be performed by selecting a training set from region IV. This procedure is relatively easy because the same atom types are used and the new parameters do not affect the coverage of other regions.

2) The other scenario is represented by regions II and III. In this situation, the parameters Po(To) are sufficient but not accurate. New atom types (Tn) must be introduced and new parameters based on the new atom types Pn(Tn) must be developed. As the forcefield parameters associated with the new atom types are not necessarily complete, we assume region II is covered by the new parameters but region III is not. Consequently, region III is not covered by any parameters because the new atom types (Tn) replaced the old atom types (To) that represented this region. This phenomenon is a common failure when a forcefield is extended: coverage is reduced as new atom types are introduced. However, this problem can be solved naturally by using the HAD method. Because new atom types (Tn) are derived from old atom types (To), we can copy old parameters associated with old atom types to new parameters associated with the new atom types for region III, as denoted by Po(Tn). Thus, region III retains coverage by the old parameters while gaining coverage under the new atom types.

Forcefield transferability is implemented during the parameterization process. Old parameters are often transferred and fixed as new parameters are derived because molecules share common chemical subunits. In other words, new parameters are derived within the constraints of old parameters. This process itself ascertains that old parameters are transferable, and that new parameters work consistently with the old parameters.

### External Extensibility using TEAMFF

The TEAMFF technology can be used to integrate different forcefields. Each forcefield in the database is modularized: functional forms, typing rules and parameters, which are inseparable parts of a forcefield, are bundled together in the database. Because of this modularization, each forcefield is independent and can be developed or imported separately.

When importing forcefields, functional forms and parameters can be loaded using text files according to the format specified in [Chapter 6](#_CHAPTER_6_–). The atom typing rules can be implemented using the script language. A definition file with the extension \*.def is a complete definition, while a definition file with the extension \*.ext is an extension of existing definitions.  The format of these definition files is given in Chapter 6.

Parameters from different forcefields with the same functional forms can be used together under the combination rule for nonbond interactions. This makes the forcefields externally extensible, allowing the user to further expand the coverage of a general forcefield. For example, one may use AMBER or CHARMM for proteins alongside a newly developed forcefield for small drug molecules.

## 3-12. Quantum Mechanical Data Forcefield

### Cutomized vs. Centralzied Parametirzation

The chemical space is too large to cover by using a single forcefield. On the other hand, only a handful number of molecules are interested in individual needs. Therefore, trying to make a forcefield to cover everything is not only impractical, but also unnecessary in most scenarios.

A new customized approach, as implemented in DFF, is to make forcefield parameters from quantum mechanical data QMD on-the-demand. This functionality is called QMDFF (Quantum Mechanical Data Forcefield).

The advantage of using QMDFF is multifold. First, the quality of forcefield is higher than traditional forcefields [1] because the parameters are made specifically for the target molecular systems; There is no compromise of accuracy and coverage as commonly seen in “general” forcefield. Secondly, it is economic; the customized parameterizations enable users to focus on their needs without wasting resources on unneeded functionality. Thirdly, it is efficient; The underlying QM data can be stored and shared without repeating the expensive QM calculations.

### What is QMDFF

QMDFF is a collection of computer databases and programs that enable the usages and developments of forcefields from quantum mechanics data (QMD) directly. In QMDFF, forcefields are made and used for molecules. Each molecule is represented by several fragments and the forcefield is made by fitting the QMDs of fragments.

**Molecule Forcefields**

Unlike classical forcefield in which parameters are identified by atom types, QMDFF forcefield parameters are identified by molecules. A molecule is a covalently bonded species with a limited number of chemical bonds (vs. unlimited in solids). A molecule can be as small as an atom, or as large as a synthetic or biological macromolecule.

To uniquely identify molecules is fundamental to the QMDFF methodology. A molecule can be partitioned into fragments, even for macromolecules the number of fragments is limited. Therefore, a molecule can be uniquely identified by its fragments. In QMDFF, a molecule is identified by a combination of SMILES strings of its fragments.

**Standard Fragment**

Fragments of a molecule are obtained by partitioning the molecule into small compounds that can be easily calculated by using QM methods, and the entire molecule can be completely described by the union of all fragment forcefield terms. The atom and bond descriptions such as formal charges and bond orders of a fragment are inherited from its parent molecule, which impacts the calculations of SMILES strings.

A potential ambiguity of using SMILES strings to represent fragments is that the same molecule may be drawn differently, e.g., phenol ring can be drawn in three ways, two with alternating double and single bonds, and one with partial double bonds. Therefore, the fragments must be standardized. To do this, first the structure of parent molecule is optimized using estimated forcefield so that the structure is reasonably close to reality. Secondly, the fragments are rebuilt using atomic coordinates optimized structures. The optimizations are done by using the estimated forcefield and the QM optimization to test if any errors exist in the estimated forcefields. The estimated parameters can be revised by [modifying the base parameters](#_Additional_Tools).

**About Formal Charge**

Formal charges are useful for ionic species. Using formal charges helps to calculate the total molecule charge for QM computation, and to define atom types more effectively from otherwise charge-neutral species. Formal charges are not necessary for charge-neutral molecules. Since formal charge calculations are entangled with bond order calculations, formal charges should be fixed before the bond orders are calculated.

**Quantum Mechanics Date of Fragment**

Quantum mechanics methods are used to obtain the minimum energy structures, and to sample the potential energy surfaces in conformational spaces of fragments. The calculated data includes atomic charges, energies, and energy derivatives of optimized and conformationally distorted structures. The data are collected in QMD (with extension of “.qmd”) files. Although any level of theory can be used for the QM calculations, a standard QM calculation protocol is suggested and used to build up a default repository. The default repository contains QMD for common fragments. If a different protocol is used, the data can be saved in a new repository to keep the data consistency.

To track the QM computation tasks, the fragments to be computed will be saved in the *F-MSD* repository as computational candidates, when their QMD files are uploaded, the candidates will be removed.

**Parameterization**

Using the QMD data of fragments, molecule forcefield can be parameterized using DFF. The parameterization is on the charge and valence parameters which are the most complicated terms in a forcefield. Since the QMD does not have sufficient description of intermolecular interactions, the parameterization does not include Lennard-Jones parameters, which are transferred from TEAMFF and fixed. The result forcefield can be saved in QMDFF database.

QMDFF is not designed to provide complete coverage of molecule forcefields because the chemical space is too large to do so. Instead, it is designed for customer parameterizations. The contents in the *M-PPF* repositories are made for common molecules, or molecules that are difficult to fit by using the automatic procedures.

### How to Use QMDFF

**Find Molecule Types**

Often, the number of molecule types of a simulation model is small. For example: a box of water/methanol mixture has only two molecule types, H2O and CH3OH; and a biological complex may have a couple of protein molecule types, one inhibitor, and one molecule type of water. DFF has a functionality to automatically identify the molecule types.

**Use Molecule Forcefields**

For a simulation model, if all molecule types are parameterized, a forcefield for the model can be created by simply merging the molecule forcefields. DFF can be used to search the required molecule force fields and construct a forcefield for the simulation model. Since each molecule is independently described by its own forcefield, only the inter-molecular interactions, namely the nonbonded interaction (LJ and Columbic) terms, are added to the model, which is done by using the well-established combination rules.

**Prepare QMD Files**

To parameterize a forcefield for a molecule, the first step is to find the standard fragments of the molecule. DFF can be used to make and save the fragments and save them. Next, DFF can be used to search if the database has required fragment QMD files for downloading. If not, DFF can be instructed to start new QMD computations, upload the QMD files when the QM jobs are done.

**Parametrize Molecule Forcefield**

When all required QMD files are collected, DFF can be used to fit a forcefield for the molecule. A least-squares fitting procedure is recommended for general fitting purposes. For difficult fitting tasks, a programable fit procedure can be deployed to seek the overall best result. The validation is done by comparing forcefield and QM computation data. When the parameterizations are done, DFF can be used to upload the molecule forcefields to the database, then circle is closed.

**User Input Commands**

The above tasks are integrated into several [automatic procedures](#_Automatic_Procedures) to simplify the usage. Meanwhile, a collection of [toolkits](#_Toolkits_of_QMDFF) is provided for specific tasks.

# CHAPTER 4 – REFERENCE OF COMMANDS

This chapter explains commands in DFF user interface, in three areas of the *DFF* window:

* 1. *Command Menus*
  2. *Toolbars*
  3. *Project Navigator*

The *Command Menus* are the main entries of commands, each of the command menus is explained in the sections below. The section of [toolbars](#_4-12._Toolbar_Commands) commands explains the quick accessible commands, which is followed by the section of [Project Navigator Commands](#_4-13._Mouse_Commands).

## 4-1. File Menu - Manage Project and Files

*File* menu provides commands used for managing projects and files. Any folder can be used as a *Project* by using File menu to create a project as explained later. Once a project is created, a project file named as “*.pjf*” is saved in the folder.

### Commands for Projects

The following commands found in the *File* menu are used for managing *Projects*. A project is a folder and its sub-folders on disk, with relevant files listed in the *Project Navigator*. The location of the project is listed as the title bar of DFF main window.

**Recent Projects** lists projects that have been opened recently. Click one on the list will open it. This is a convenient method to access existing projects.

**Open Project** opens a dialog to reload or create a project from disk. Browse to a folder and click **Open**. If this folder contains .pjf file, the files and folders used by previous session will be restored.

**New Project** opens a dialog to create a new folder and opens it as a *Project*. Browse to where to place the work, enter a name for the project and click **Create Project** command in the dialog. A new folder will be created and loaded.

**Close Project** closes the current project. The project folder and its files and sub-folders are saved. If the project content has changed since the last save, you will be prompted to save again before closing the project.

**Refresh Project** reloads all contents of the current project folder. All subfolders and DFF-relevant files will be listed in the *Project Navigator.*

**Save Project** saves the current project folder under the current name and disk location.

**Save Project As** saves the current project under a specific name or location. You can select a different name in the same folder or a different folder altogether.

### Commands for Files

**New** is a command to make a new model or folder to the project. The model can be created by drawing and saved in MSD format.

**Import** is a command to load existing files to the project. The files can be one of those supported file formats. The imported file will be saved in the project folder and listed in the *Project Navigator*.

**Open** can be used to select and load MSD files to the interface. This function can be done by double-clicking on the model nodes in Project Navigator.

**Save** will save the selected model to disk file if the model has been changed.

**Save All** will save all opened and modified models to the disk.

**Close All** closes all models. For any models that have been changed, a warning will be given. You can select either save or abandon any changes.

**Exit** closes DFF.

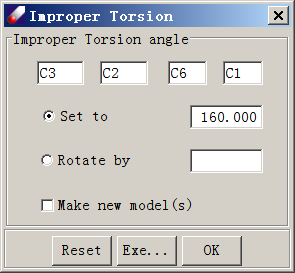
## 4-2. Edit Menu – Sketching, Editing and Modifying Models

The following commands are used to editing and modifying molecular models. They are listed in the **Edit** menu.

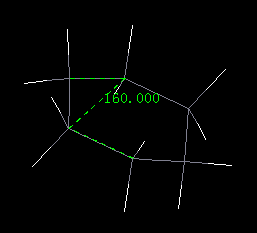
### Menu Commands

**Isotope** sets atomic masses in order to make isotopes. This command can be applied to change a single atom or all atoms of a given type in a selected model.

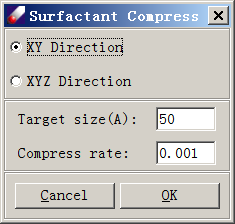
**Guess Bond Order** sets the bond orders of the selected model. If the model already has bond orders set, this operation resets all bond orders. This is useful for imported models, in which bond orders can be reset to ensure that atom types are assigned correctly. Bond orders are recalculated based on the method explained in Section 3-3.

**Formal Charge** assigns formal charges automatically or manually. The automatic option uses the method explained in last chapter, Section 3-3. The manual option is useful for assigning special formal charges.

**Torsion** sets or modifies torsion angles and opens a **Torsion** dialog. With this dialog open, selecting four atoms defines a dihedral angle. A new angle can then be set, or an existing one can be rotated by a specified value. Clicking **Execute** performs the specified function. If **Make new model(s)** is selected, each **Execute** click will generate a new model with the specified bond rotation.

**Improper Torsion** is similar to the **Torsion** command, but is used to set or modify improper torsion angles. In the popup dialog, selecting four atoms defines an improper dihedral angle for which the bond angle may be set. Improper torsion can be defined for four atoms that may not be consecutively bonded. In the example at right, this command is used to set the out-of-plane angle of cyclohexane.

**Reset COM** resets the center of mass for a model. Relative positions of atoms are not altered and energy calculations are not impacted. **Reset COM** is useful to obtain a better view of the model, especially when the model needs to be rotated on the screen.

**Repack PBC** has two sub-commands - **Repack** and **Undo Repack**. **Repack** moves molecules into a box with periodic boundary conditions (PBC) using the minimum image rule, while **Undo Repack** reverses the move. The position of a molecule is determined by its center of mass.

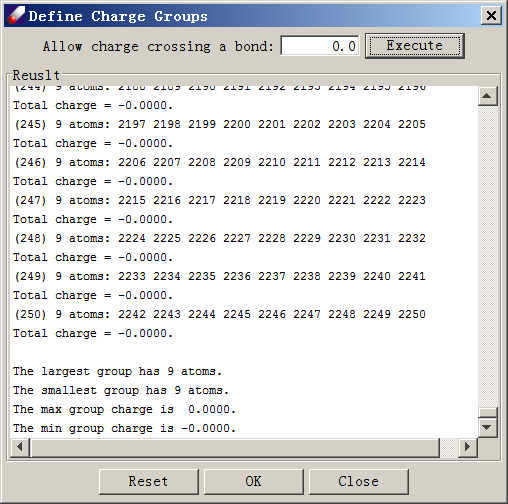
**Adjust PBC** compresses or expands the PBC box in two or three directions. This is done by adjusting the dimensions in small steps with quick MD simulations. This command is useful for preparing interface models.

**Set PBC** is used to **Add**, **Edit** or **Delete** unit cell parameters for a model. Select one of the available options, specify the cell dimensions as necessary, and click **OK** to launch the job.

## 4-3. Build Menu - Building More Complicated Models

### Group and Subset

**Charge Group** defines charge groups in a molecule model as is often required by simulation engines. This method can be used only when a forcefield has been assigned to the model. It works by querying the charge separation between any two bonded atoms. If the charge crossing a bond is smaller than a threshold that can be adjusted, then the bond is used as a division of two charge groups. **Execute** begins the calculation, and results are displayed in the center panel.



**Subsets** defines a group of atoms or molecules as a subset of the molecular model. It opens a **Define Subsets** dialog, which provides two options, **Manual Define** and **Auto Define**. The **Manual Define** option offers the flexibility of defining subsets based on atom or molecule. In either case, atoms or molecules can be selected individually. Alternatively, all atoms or molecules identical to a selected atom/molecule can be selected at once. **Define** finalizes one subset definition. The **Auto Define** option identifies subsets and defines them as molecules automatically. **Execute** runs the definition algorithm. Defined subsets are marked with different colors. If necessary, **Reset** restarts the process.

### Bulk Models

**Bulk Liquid** builds a molecular liquid model in a cube (A=B=C) or a slab (A=B≠C) simulation box. Molecules are placed with random orientations in the box at low density initially and then compressed to a targeted density.  Clicking **Bulk Liquid** opens a **Make Liquid Model** dialog. Use the pull-down menu under ***Model*** to specify the molecular composition of the liquid. Mixtures can be built by selecting different molecules with different numbers (counts). ***Counts*** should be entered as number of the selected molecule. Pressing **Enter** will instruct DFF to calculate the percentage of each molecule in the mixture, the total number of atoms and the number of molecules in the box. ***Density*** allows the user to specify a target density, and ***Compress Rate*** allows the user to specify a rate at which the system will be compressed to the target density. Clicking ***Build Cubic Box*** or ***Build Slab Box*** will begin the process. For a cubic box, all three edge lengths (A=B=C) are calculated based on the given number of molecules and density. For a slab box, the box height is specified by the user (C) and the other two edge lengths (A=B) are calculated. **Execute** starts the background job. When the job finished, the initial bulk liquid model will be loaded into the *Project Navigator*.

### Interphase Models

**Liquid Interface** combines two or more bulk-liquid boxes to build an interface of liquids. It opens a **Liquid Interface** dialog. In the **Choose Boxes** portion of this dialog, select your liquid models of choice by browsing to your prepared models (.msd file). The distance between adjacent liquid boxes can be set using **Interval**. Specify an **output file name** and click **OK** to launch the job. The combined box will be loaded automatically when the job is finished.

**Bilayer Interface** builds water-surfactant-air or water-surfactant-oil interfaces. The algorithm will build the water box first and place surfactant molecules on either side of the water phase with hydrophilic groups oriented towards the water phase. If an oil phase has been chosen, this phase will then be placed outside the surfactant phase. Within the **Bilayer Interface** dialog, ***Box Name*** specifies the name of the model to be built. ***Box Width (X)*** and ***Box Height (Y)*** set the dimension of the surfaces and thus determine the area of the surfaces. **Build Options** specifies steps of energy minimization or/and molecular dynamics after the box is built, and defaults to no simulations.

To build a bilayer interface, select **Water** in the **Liquid A** section using the pull-down menu and specify the ***Thickness*** of the water phase.  **Add** can be used to add additional components into the water phase. **Delete** can be used to remove added components. **Update** calculates the number of water molecules within the constraint of a density close to 1.0. Surfactant molecules can be specified in the **Surfactant B** section using the ***Surfactant*** pull-down menu. Surfactant models must have been prepared and saved in the same folder as the current bilayer interface project. Once a surfactant molecule is selected, a number of surfactant molecules per side must be chosen. It is recommended to place N^2 (25, 36, 48 ...) molecules in order to build an evenly distributed surface. Note that because the number refers molecules per side, the total number of surfactants is double your chosen number. **Add** can be used to add an additional surfactant species (without changing the total number of surfactant molecules), and **Delete** can be used to remove surfactant species. Finally, **Liquid C** can used to describe either a vacuum when building water-surfactant-air interfaces or an oil when building water-surfactant-oil interfaces. In the latter case, select the oil molecule and specify its density and count-per-side as described above. **Update** can then be used to calculate the thickness of this phase.  
When you are satisfied with your bilayer composition, **OK** launches a background job. A bilayer box without minimization or simulation steps can usually be built in a few minutes.

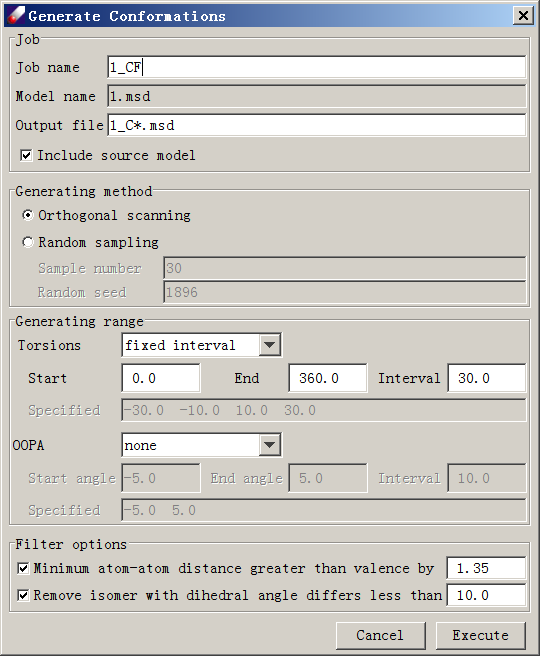
### Super Cells

**Super Cell**builds super cells by extending cell lengths into a two- or three-dimensional periodic box. For a 2-D box, only the A and B edges can be extended. The extension is accomplished by multiplying the cell edge lengths by integers.

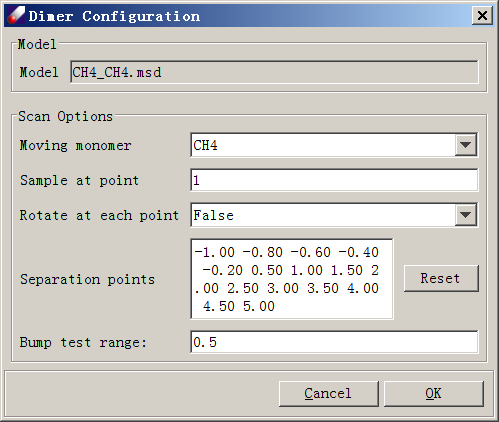
### Fragments and Sampling

**Fragments** makes molecular fragments for parameterization. This function works by constructing molecular fragments based on the parameterization needs of a selected model. The software automatically builds fragments and saves the results to a subdirectory.

**Conformational Isomers** is useful for building conformational isomers for a molecule in order to sample its energy surfaces. This command should only be used on small molecules, and the model should first be optimized. User-input parameters are used to generate conformational isomers. There are essentially two methods: orthogonal scanning or random sampling. If orthogonal scanning is chosen, the software automatically searches for rotatable bonds and generates isomers by rotating those bonds. If random sampling is chosen, the user sets the number of samples, the random seed, and the range and interval of sampling. Two types of internal coordinates, the torsion dihedral angle and the out-of-plane angle, can be sampled. after After sampling, the software will check generated conformers and remove structures in which atoms are too close to one another. This threshold is set by the **Bump factor**, which is set as a multiplier of the atomic radii. A user-determined **Symmetry factor** can be used to remove structures that are identical due to rotational symmetry. Upon **Execute**, this command will create a sub-folder in the project and save output configurations in the sub-folder.



**Dimer Configurations** generates different configurations of a molecular dimer by moving one monomer along the line of the two centers of mass. To use this command, the dimer should be optimized first. The software generates configurations based on user-specified parameters that can be edited by using the **Reset** button. The configurations are **Separation points** between the two molecules as given in the text area. The monomers may be rotated randomly at each separation point. **Bump test range** sets a threshold of atom-atom distance between two monomers, below which the configuration will be removed.



## 4-4. View menu - Managing How Models Are Displayed

### View Commands

These addition commands are available through the **View** Menu.

**Measure** displays measured molecular properties of the selected molecule. To change what is measured and displayed, click **View → Measure**. The following properties can be measured:

Angle  
Bond   
Torsion angle  
OOP angle  
Dipole moment

Measurements are taken using mouse clicks. Once a measurement property has been selected, click on atoms to measure the bonds, angles, or moments between them. Selected atoms are labeled by colored dots, and measured values are displayed next to selected atoms.

**Pack View** displays all atoms inside of a periodic box. It does not change the coordinates.

**Job Monitor** toggles a display of background DFF jobs.

IDE Optionschanges the **Look-and-Feel** style of DFF on the Windows platform. Four different **Look-and-Feel** styles can be selected: Metal, CDE/Motif, Windows and Windows Classic.

**Display Style** changes the molecular rendering style:

**Line** - the default style in which colored lines indicate atoms and bonds. Nonbonded atoms are displayed as a circle.  
**Stick** - a style in which thick lines indicate for atoms and bonds.  
**Stick and Ball** - a style in which thick lines indicate bonds and balls indicate atoms.  
**CPK** - the space-filling Corey, Pauling, and Koltun (CPK) model frequently used in industry.

### Toolbar Commands

These commands are available both through the toolbar, and are used to alter the behavior of mouse clicks while viewing a model.

**Rotate** () allows the user to **Left Click-and-Drag** to rotates the model around a central axis.

*Tip: It may take a bit of experimentation to make the model do what you want on the screen in Rotate mode. The easiest way to visualize its movement is to imagine the structure as being confined within a globe. Moving the mouse left will turn the globe toward west along a north-south axis, whereas moving it right will turn the globe east. Moving the mouse up will turn the globe north along an east-west axis, while moving the mouse down will turn it south.*

**Translate** () allows the user to **Left Click-and-Drag** to slide the model up, down, left, or right in the working area.

**Zoom** ()allows the user to zoom in by **pushing left-button and drag right** or zoom out by **pushing left button and dragging left**.

**Center** () moves the selected model back to center of viewing area.

**Label ( )** is a pull-down menu that sets display modes for various properties of the selected model. Label options are:

**Clear All** - removes all labels  
**Atom Names** - atom names as given in .msd file.  
**Atom Types** - the Apparent Atom Types.

**Mulliken Charges** - as given in QMD files.  
**ESP Charges** - as given in QMD files.  
**Forcefield Charges** - assigned by forcefield   
**FF/ESP Charges** - comparison of forcefield and ESP charges  
**Formal Charges** - formal charges  
**Charge Groups** - charge groups   
**Total Energy** - the total energy and cell dimensions if available  
**Hybridization** - the hybridization states.  
**Bond Lengths** - measures and displays average bond lengths.   
**Measurement** - labels the measured properties (see below)   
**Subsets** - color-codes defined subsets.

## 4-5. TEAMFF menu – the TEAM Forcefield Database

TEAMFF is a database of premade forcefields, which are managed by Direct Forcefield Database (DFFDB). These commands manage the database and its content forcefields. This menu provides the following commands:

* Admin
* Assign
* Check-out
* Check-in
* New Database

### Managing the Database

**Admin** is the method of managing database contents. It opens the DFFDB Manager dialog, for example:

A screenshot of a computer

Description automatically generated

*Database file* lists where the database data file is located. Each database is stored as a binary file with a “dffdb” extension. Normally one database is sufficient because each database can contain multiple forcefields. Database files can be placed anywhere on your computer. Use **Browse** to locate and load a database file. The default location is the <DFF installation>/database folder.

*Forcefield type* indicates what forcefield type will be used. Choose a forcefield type using the pull-down menu. Available forcefield types for the selected database are listed. After selecting a database and a forcefield type, DFF will search and list all forcefields of the same forcefield type in the **Table** field.

Each table can be selected by checking its **checkbox**. To the right of the checkbox is the ***Name*** of the table, its highest ***Version*** number, the number of unique and total ***Entries***, its ***Typing Rules***, and brief description of the table. The **Version** number can be selected using a pop-up pull-down menu that appears when the version cell is selected. This sets a specific version to be used, but does not change the table.

Two hidden commands, **Show log** and **Revert**, can be accessed by **right-clicking** a table entry. **Show log** displays a log of parameter check-in times and dates. **Revert** resets the highest version number to a user-specified version. Note that the **Revert** function, unlike **Version**, will actually remove entries after the specified version and should be used with caution.

The right this dialog contains four commands for ordering and managing database tables: **Add**, **Remove**, **Up** and **Down**. The order in which tables are listed is important. When the database is used, DFF reads parameters from the forcefield at the top of the list first and moves down as necessary. To reorder the list, select a table and click **Up** or **Down**. To delete a table, use **Remove**. *Note that this operation deletes the table and its contents from the database.* To add a new table, click **Add** to open an **Add New Table** dialog.

A screenshot of a computer

Description automatically generated

In the **Add New Table** dialog, enter a ***Name*** for the forcefield to be created and select a ***Type***, which must be match that of the checked-in forcefield. Similarly, the **Typing Rule** must also match that of the checked-in forcefield. Note that once typing rules are decided, they are permanently attached to the forcefield and can only be modified but not replaced. **Forcefield** allows you to enter or browse to the .ppf file to be checked in. However, it is not necessary to add parameters at this stage, as they can be added later. Finally, enter a brief **description** of the table and click **OK** to create a new table.

### Use the Database

**Assign** retrieves parameters from the database and assigns them to the model(s) selected. To use this command, select a model(s) from the *Project Navigator* and click **Assign** to open an “Assign forcefield parameters” dialog:

A screenshot of a computer

Description automatically generated

***Input*** lists the tables and database selected. The selected input can be changed by clicking **Choose**. ***Output*** specifies where the output forcefield file will be saved. Use **Browse** to change the location.

Three options can be selected. If parameters are missing for a given atom type, *Transfer parameters* will instruct DFF to find parameters similar atom types. The retrieved parameters are scored in the output file on basis of similarity to the required atom type(s). *Make automatic parameters* is an alternative method to automatically obtain missing parameters. This option allows DFF to automatically generate parameters using the rules explained in section 3-9. The last option, **Save to SPF File,** makes an integrated file (.spf) from the model (.msd) and the forcefield parameter (.ppf) files.

**Available Models** lists all models to be assigned in this operation. Forcefield terms required by these models will be retrieved from the database. You can **Add** or **Remove** any models using **Reset** searches current folder to load all models available. Click **OK** to start the operation. If all parameters are found, retrieved parameters are saved and loaded into the *Project Navigator* and models are assigned atom types and charges. If some parameters are missing, a sub-folder is created and molecules with missing parameters are saved in this folder.

### Expand the Database

**Check Out** retrieves parameters from the database, but the operation is different from that used by **Assign**. The main difference is that this operation retrieves data from database tables not only the same forcefield type, but also the same typing rules. Therefore, atom types must be consistent between the forcefield from which parameters will be checked out and the forcefield to which parameters will be checked in. The **Check out Parameters** dialog is like the **Assign forcefield parameters** dialog, but the **Make automatic parameters** and **Make SPF file** options are not available.

The resulting forcefield file is mainly used to continue parameterization. If some parameters are missing, this forcefield will be used as the initial input, and existing parameters are frozen so that only missing parameters can be added.

**Check In** to add new parameters into the database table. It opens the **Check in Parameter** Dialog. ***Database*** shows the database selected. ***Table*** lists the table to which new parameters will be added, which can be selected using a pull-down menu. The ***Type*** is fixed since the database table has already been selected. The ***Version*** number is automatically updated in minor revision increments. A major version increment can be manually set via the pull-down menu.

In the **Forcefield** section, **Browse** to the .ppf file to be checked in. ***Description*** shows the development log, which can be viewed in the **Database Manager** dialog.  ***Reference*** lists citation for this check in. **Organize** opens a dialog to manage references in which references can be added or removed. References can be viewed directly from the forcefield table. Clicking **OK** will insert newly checked-in parameters into the table. The database engine will automatically check for existing parameters in the input forcefield file such that only new terms and/or changed terms will be saved into database.

### Make a New Database

**New Database** opens the “Create TEAM Database” dialog to create a new database. Browse to a folder and enter a file name for the new database. An empty database will be created and loaded into the "Database Manager".

## 4-6. Simulation Menu - Internal and External Simulation Methods

### Internal Simulations

**Energy** opens an **Energy** dialog used to set up single-point energy calculations. ***Job Name*** identifies the running job and its output file. **Nonbond Energy** button opens a **Nonbond Control** dialog to set up options for nonbond calculations. Computation options include ***Gradient*** and ***Hessian***. After the job finishes, the input, output file and result structure will be loaded into a new folder of the Project Navigator. The **Nonbond Control** dialog also allows the user to select a cutoff method. Three options can be selected: ***Atom Cutoff***, ***Group Cutoff*** and ***No Cutoff***. A ***Cutoff*** value and a ***Buffer*** value can then be set. The **Buffer** value is used to calculate the neighbor list efficiently. Both are specified in Angstroms. If using **Group Cutoff**, charge groups must be defined using **Build → Group by Charge**. If ***No Cutoff*** is selected, all nonbond interaction pairs are calculated. For systems with periodic boundary conditions, ***Tail Corrections*** can be added to compensate for energy and pressure loses due to cutoff.

**Optimization** performs structure optimizations by minimizing energy. It opens an **Optimization** dialog that allows the user to set a ***Job Name***, ***Number of Steps***, and a ***Convergence*** criterion. Two optimization algorithms can be selected: ***Conjugate gradient*** and ***Variable metric***. Options for nonbond energy calculations are accessible by clicking **Nonbond Energy**, as explained above. To add torsion restraints during optimization, click the checkbox to open a new **Add Restraints** dialog to set up the restraints. In the **Add Restraints** dialog, the top section displays input restraints. To add a restraint, click and highlight the first cell labeled **a1**. This allows you to select atoms from the display window by **Left-Clicking**. Click and select four atoms to fill the four atom cells (a1, a2, a3, a4). Enter a restraining force constant (100 is the default), a starting angle *x1*, increments *dx*,and a total number of points *n*. Clicking **Add** will add one restraint.  To add another, repeat the above procedure. Click **OK** to quit this dialog. To see your changes to the structure and energy of the molecule, select the **Screen view** option.  Clicking "Execute" launches the background optimization job. When the job finishes, the input, output file and resulting structure will be loaded into a new folder in the **Project Navigator**

**Molecular Dynamics** performs molecular dynamics simulation. It opens a **Molecular Dynamics** dialog. In this dialog, a ***Job name*** must first be specified before a simulation ensemble can be chosen - ***"NVT", "NPT"*** or ***"NVE"***. To use a temperature control method, click **Temperature (K)** and select either the ***"Stochastic*** ***(Anderson)"*** or the ***"Scale Velocity"*** method. To use a pressure control method, click **Pressure (MPa)**. The ***Berensdsen*** method is implemented. **Nonbond Energy** can be used to set up nonbond calculations as explained above. In the **Run time** section, ***Timestep (fs)***, ***Equilibrium Steps*** and ***Evaluation Steps*** are user-set parameters that control how the simulation is to be performed. The **Intervals** section is used to for control how frequently (in number of steps) the ***Trajectory*** will be saved and how frequently the **Screen View**, if selected, will be updated if ***Screen View*** is selected. **Execute** launches the background job. When the job finished, the input, output file and resultant structure will be loaded into a new folder in the **Project Navigator**.

### External Simulations

**GROMACS** or **LAMMPS** opens an interface that allows the user to set up simulation jobs using external *GROMACS* or LAMMPS software. The interfaces are very similar. We will use LAMMPS as an example.

***Model*** lists the molecular model file (.msd) to be used for the simulation. The model should have atom types, charges, and forcefield parameters assigned. ***Forcefield*** indicates the forcefield file (.ppf) assigned to the model. ***Program*** allows the user to select either *GROMACS* or LAMMPS from a pull-down menu. By default, ***LocalHost*** is the computer that will run the simulation task. The ***Computer*** button can be used to add new and remote computers, which can subsequently be selected using the pull-down menu. Clicking **Computer** starts up a new **Connection** dialog. Enter the ***IP address***, ***Port***, ***User name***, ***Password*** and ***Directory*** to connect to a remote computer. The software, *GROMACS*/LAMMPS must also have been installed and configured for the remote computer.

The ***Ensemble*** section lists three usable ensembles, ***NVT, NPT*** and ***NVE***. ***Temperature*** sets the simulation to run under temperature control methods, the choice of which depends on the software selected.  ***Pressure*** sets the simulation to run under pressure control method. ***Nonbond interactions*** sets options for nonbond evaluations. An option of **Apply T-factor to LJ term** scale the nonbond parameters using temperature scaling factor. A new PPF file will be generated before the forcefield is exported to the simulation software. ***Constraints*** allows the user to set up certain constraints supported by the simulation software. ***Trajectory*** specifies how the simulation is to be conducted. ***Pre Optimization*** allows the software to perform an energy minimization before MD simulations. ***Time Step*** specifies the integration time step in fs. ***Equilibrium*** sets the number of steps to simulate before collecting data for analysis. ***Trajectory collect*** sets the number of steps to simulate during data collection and after the equilibrium steps. ***Save Interval*** sets the interval in steps between saves of the trajectory. ***Trajectory file name*** sets the file name for the generated trajectory files. Click **OK** to start the simulation job.

**Job List** opens a **Job Monitor** dialog used to view external simulation jobs. Jobs that have been submitted are listed here. More information about each job can be viewed by **double-clicking** a record. **Right-clicking** on a record allows the user to perform four functions. ***Refresh*** rechecks the status of this job. *If the job is finished, it will be marked as* (). ***View Log*** displays a log file of the job. ***View Result*** displays simulation results and should only be used when the job has finished. ***Delete*** removes the job record from the list but preserves the saved files.

A screenshot of a computer

Description automatically generated

**Trajectory View** displays the simulation trajectory on DFF screen. Paired topology (.msd) and trajectory (.dta) files must be selected before clicking this command. A dialog will then show the selected files with options to either ***Save the last frame*** or ***Show the trajectory***.

A screenshot of a computer

Description automatically generated

**Export** allows the user to prepare input files for different simulation engines. Exported files are saved in a folder with the default name of the forcefield type used. Resultant forcefield parameters and model structural data are written into output files with the following extension names:

|  |  |  |
| --- | --- | --- |
| **Forcefield type** | **File Extensions** | **Explanation** |
| *GROMACS* | *mdp, gro, top, itp* | *mdp=input, gro, top=coordinates, itp=forcefield* |
| *LAMPPS* | *in, data* | *in=input, data=coordinates and forcefield* |
| *TOWHEE* | *input, coord, ff* | *input=input, coord= coordinatse, ff=forcefield* |
| *AMBER* | *dat, prep* | *dat = forcefield, prep = coordinates* |
| *CHARMM* | *par, rtf* | *par = forcefield, rtf = coordinates* |
| *DISCOVER* | *car, mdf, frc* | *car, mdf = coordinates, frc = forcefield* |
| *DREIDING* | *drg* | *drg = forcefield file* |

## 4-7. QMDFF Menu – quantum Mechanics Data Forcefields

This menu provides information of the commands that utilize the functionality of [quantum mechanics data forcefields](#_3-12._Quantum_Mechanical).

### Manage the Database

**DB Admin**

This command opens a dialog that shows the content of the database. Using **Browse/New** to open or create a database. The default location is <DFFRoot>\database\QDFF.db, but the database can be saved in any place. Click **List** to show the contents. There are three tables, *“F-QMD”, “F-MSD”,* and *“M-PPF”*, for fragment QMD, fragment MSD and molecule forcefield PPF respectively. Here F means fragment and M means molecule, to signify the purpose of the repository, although in many small molecules there is no difference between a molecule and a fragment.

*F-QMD* is the main repository, it contains many QMD files. This is the database for parameterization of molecular forcefields. Each fragment is identified by its SMILES, the molecular formula, theoretical level of computation and size of data points are listed. The data points are obtained by the automatic QM computations, each datapoint is on calculation on the conformation space. For rigid molecules, the number of data points is small.

*F-MSD* is a working repository, which is used for QM calculations. Since fragments may be found in different molecules and the QM computation is time consuming, saving molecules to be computed in the repository helps to eliminate redundant computations and it is convenient to use the repository for large scale batch-job runs. In this table, select molecules and right-click mouse button will prompt to submit a QM job.

*M-PPF* is a repository of parameterized molecular forcefields. Molecules are identified by c-SMILES which is the same as SMILES for a small molecule that is a fragment, or a combined SMILES for molecules constitutes several fragments. The typing rule and functional form are labeled. This repository contains forcefields that have been parameterized.

### Automatic Procedures

**Enquiry**

This automatic procedure is used for getting a forcefield for a simulation model. A simulation model can be a single molecule, a collection of many molecules, a mixture of different molecules, etc.

Select a model node, and then click this command to open a dialog. The user input is simple, only need to specify the forcefield type, either **AMBER** or **TEAM**. Then click the **OK** button to start a background job.

The background job identifies the unique molecules first, and then searches molecular forcefields for the molecules.

If all enquiry forcefields are found, this command will make a merged PPF and update the MSD files for atom types and charges, which are ready for simulation.

If any molecular forcefield is missing, the background job will run to partition each of the molecules into fragments and save the fragments in *subfolders* *named after the molecule*. For easy to read, the molecules are named as a letter ‘M’ followed by numbers, the fragments are named as a letter ‘F’ followed by numbers.

*The following commands are used for the parametrization, which should be done for each of the molecules. When all molecules are parameterized, repeat the* ***Enquiry*** command to get the final forcefield.

**Collect QMD**

This is a procedure preparing QMD files for parameterization. This should be done for each molecule to be parameterized. To use this command, move mouse point to a subfolder of molecule in the *Project Navigator*, select *all* fragments in the folder, and click this command. A background job will be launched, it searches QMD files that exist either in the current folder or in the database and prompts user to make choices. Depending on the user’s selections, it may skip new computations or repeat the computations so that the existing QMD files can be updated.

If the QM jobs are launched directly, the result QMD files will be saved in the same molecule folder. If the QM jobs are done externally, either move them back to the folder, or upload them to the QMDFF database and then use this command to download them.

**Smart Fit**

This command uses an optimized fitting procedure to fit forcefield automatically. Make sure all QMD files are prepared and selected from the *Project Navigator*, and then click this command to start a dialog. The interface is simple, only needs to specify the forcefield type. Click **OK** to start the background *DFF* job, which fits and validates the forcefield automatically. A job dialog will appear on the screen, which shows the fitting and validation output. In the end, check the fit result table (\*.dft) to examine if the results are acceptable.

If the fit quality is poor, consider using the **Program Fit** command, which provides more flexible options to improve the fit quality, or discover the cause of poor fitting.

**Assign & Upload PPF**

This command runs a short MD simulation to test if the forcefield is stable, and prompt upload the result PPF to the database. To use this command, select the result ppf file in last step from the *project navigator*, a dialog will prompt selection of the molecule to be assigned. *The molecule must be the same one used for the parameterization.* Once the molecule is selected, a dialog appears to launch a short MD run. If the simulation appears to be stable, a dialog will appear to prompt upload the .ppf file to *M-PPF* repository.

**Automatic Workflow**

This command prepares a complete procedure in Python script to run the above tasks for a list of target simulation models without attendance. The input is a list of target simulation models, the output is updated target models and associated forcefields, and the produced molecular forcefields are saved in *M-PPF* repository. This workflow is designed for large scale parameterization.

### Toolkits

*The following commands perform specific tasks, which are useful for analyzing problems when the automatic procedure fails.*

**Get Molecules**

This command is used to split a simulation model to unique molecules. Molecules are covalently bonded species; unique molecule means the kind of molecule that may be repeated multiple times in the simulation model. The unique molecules identified are saved in a subfolder named after the simulation model.

**Get Fragments**

This command is used to make standard fragments. DFF will make fragments by cutting from a molecule, and then make standard fragments. If the standard fragments are different from originally cut from the molecule, an error is returned. The cause must be investigated. Since the standard fragments are obtained by using estimated forcefield, it is helpful to carry out QM calculations and compare the standard fragment obtained after the QM optimization.

**Compute QMD**

This command is used for *standalone* QM calculations, which does not check if the molecules have been calculated and saved in the database. This command is useful for independently making new QMD files.

The dialog is like that explained in the [QMD section](#_QM_Calculations), however, there are several important options. The total charge and multiplicity are set to be COMPUTER, DFF calculate these quantities depending on the formal charge information. The default basis set is 6-31+G(d,p)/DGDZVP, which means for elements the first three period of the periodic table (up to Kr), the basis set is 6-31+G(d,p). Beyond the third period, the basis set is DGDZVP. The most important difference is that this dialog prepares a complete process of automatic QM computations, which is specified in the “Task” section near the bottom. Three options: “***Optimization***”, “***Scan Dihedral***” and “***Normal mode***” indicate the computation types, among them the *Scan Dihedral* can be very expensive, depending on how flexible the fragment molecule is. For forcefield parameterization purposes, the “***1-Dimension***” scan is more appropriate. The QM jobs can be launched directly or saved in batch-job scripts. To avoid redundant computations, the molecules to be calculated are saved in the *F-MSD* repository, which can be used to prepare batch-jobs as well as explained above.

**Download QMD**

This command is used to search and download QMD files in the database. Select *MSD* files from the *Project Navigator* and click this command to open a dialog. Note that the repository saves QMD files of fragments, make sure to select MSD files are fragments. *DFF* will make SMILES strings, search, and download QMD files.

**Update QMD**

This command *updates* the QMD repository, which means it will *replace* the *QMD* files if they exist or insert new QMD files. Select QMD files from the *Project Navigator* and click this command to open a dialog.

**Program Fit**

This is a command useful for dealing with difficult fitting tasks. It is flexible for setting single or multiple jobs, and options for each of the jobs. Depending on the setting, the jobs can be run sequentially or independently. Each job produces a set of result *DFI, DFO,* *PPF* and *DFT* files which will be loaded to the *Project Navigator*. By comparing the outputs, the best result and setting can be identified. The setting will be saved and reloaded next time, so that the same setting can be repeated used.

To use this command, select QMD files from the *Project Navigator*, optionally pick up an initial PPF file, and then click this command to open a dialog. The dialog contains several tabbed panels, each tabbed panel describes a job. The forcefield type, number of jobs and fitting variables of each job can be edited. Use **Execute** command to start the background jobs.

The fit quality depends on the quality of data and the functional forms used in the forcefield. In most cases, the data should be sufficient. If in doubt, inspect the QMD data using [QMD table](#_Quantum_Mechanics_Data). Using less complex functional forms may cause underfitting, but using more complex functional forms may lead to overfitting. The default setting of DFF is for general purposes and overall reasonable fit quality can be obtained.

**Download PPF**

This command is used to search *PPF* files of molecules. Select one or more MSD files from the *Project Navigator* and click this command opens a dialog that lists the models selected. Select a forcefield type, then click **OK** to launch a background job. *DFF* will partition each molecule to standard fragments and calculate its c-SMILES and use the c-SMILES to search the *M-PPF* repository and download if found.

**Update PPF** replaces or inserts selected PPF to the database. Select a forcefield (PPF), which must have c-SMILES embedded, if not, associate this forcefield with the target molecule to calculate and embed them and then click this command. DFF will update the forcefield table using the C-SMILES.

## 4-8. Forcefield menu - Managing Forcefield

### Managing Forcefield

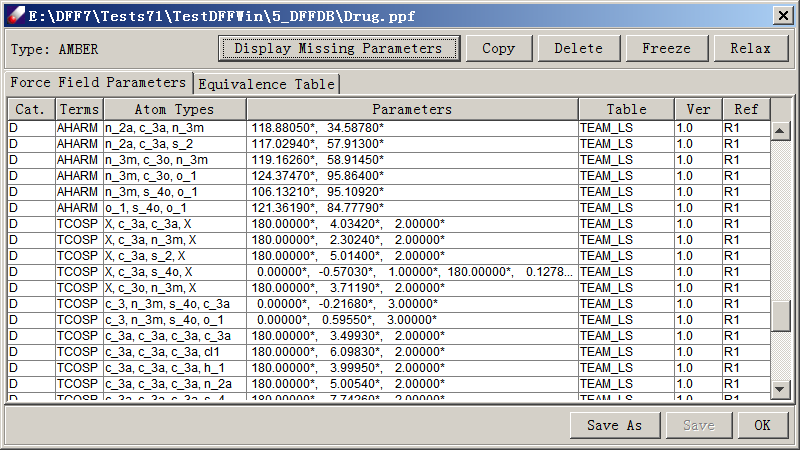
These commands are used to manage forcefield files. In DFF*,* forcefields are written as .ppf files.

**New Forcefield** opens a dialog for creating a new forcefield for the selected models. The forcefield can be empty or optionally filled with estimated parameters. First, select a ***Forcefield type*** from a pull-down menu, which fixed the functional forms. The ***Typing rule*** determines how atoms are represented in the forcefield. Other than the “Default” option which uses computed atom types, you can also use definitions (\*.def and \*.ext files) found in the <dff-root>/data/TypingScripts folder. The ***Level of atom type equivalence*** sets how the apparent atom types are mapped to equivalent atom types, which impacts the real number of atom types used in the forcefield. Next, enter a name for the forcefield. Finally, select or deselect the ***Estimate parameters*** option. If selected, both charge and valence parameters will be estimated using the UFF algorithm. Click **OK** to lunch a background job. When the job is done, the new forcefield should be displayed.

**Open PPF** browses and loads a PPF file into current project.

### Forcefield Editor

**Open a Forcefield Table** opens a table used to edit and organize forcefield parameters, and can also be accessed by **double-clicking** on a forcefield file in the *Project Navigator*.



There are three sections in the table. At the top are commands that can be performed on parameters. The middle section displays the parameter table and the optional equivalence table. The bottom portion contains commands for the forcefield file.

Commands that can be performed on parameters are **Display Missing Parameters**, **Copy**, **Delete**, **Freeze** and **Relax**. To use these commands, select one or more lines in the parameter spreadsheet and click on a button. **Display Missing Parameters** is activated if parameters are missing (with empty lines), and will shows atom associated with the missing terms. **Copy** makes copy of the selected lines and appends them to the table. **Delete** removes selected lines from the table. **Freeze** prevents parameters from being adjusted and marks them with an asterisk (\*). **Relax** allows frozen parameters to be adjusted and removes the asterisk.

The parameter spreadsheet has six (6) columns - ***Table, Version, Terms, Atom Types, Parameters and Reference.***  ***Table*** shows the database table from which parameters are taken. ***Version*** displays the parameter version number, and ***Reference*** indexes the reference list. These terms may be empty if a forcefield was not taken from a TEAM-FF database. The central three columns, ***Terms***, ***Atom*** ***Types*** and ***Parameters****,* are essential for any forcefield.

Clicking on the heads of columns sorts the lines accordingly, which is helpful for finding terms. ***Atom Types*** and ***Parameters*** cells are editable. **Double click** on a cell to change its values. **Right-clicking** a line brings up three commands - ***Freeze force constants, Reverse to original,*** and ***Set to zero***. These commands provide additional ways to control parameters, which is useful for difficult fitting tasks.

Commands that can be performed on forcefield file are **Save As, Save** and **OK** to close the dialog.

### Using a Forcefield

When the forcefield files are ready, the following commands can be applied:

**Assign Atom Type** is used to assign atom types to models using an **Assign Atom Types** dialog. In this dialog, four methods can be used to assign atom types: **Use Default typing rules** call the program to calculate atom types; **Use scripted typing rules** let selection of the typing scripts in the pull-down menu; **Replace atom type from**, in which all atoms can be switched from one type to another, and ***Assign type for Atom***, in which atom types can be changed for a particular atom. **Execute** assigns atom types using the selected rules.

**Associate Forcefield** assumes atom types have been assigned to the selected models and associates a forcefield with the models. It brings up a list of forcefields available for selection in a given project, checks if parameters are complete in the selected forcefield, and assigns atom charges accordingly. This command is also accessible from the **Project Navigator** by selecting models and right-clicking.

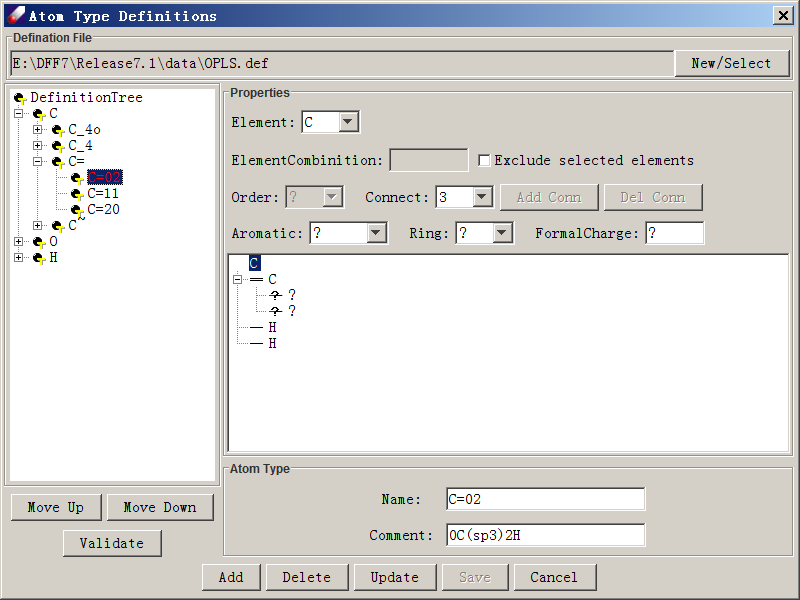
**Assign Atom Charges** assigns or recalculates atom charges using charge parameters in a selected forcefield. It is active only when charge parameters are available in the selected forcefield.

### Advanced Topics on Forcefield

Importing an external forcefield to DFF involves two steps: 1) reading in parameters and writing them to a PPF file, and 2) creating typing rules for the external forcefield. After parameters and typing rules are prepared, the forcefield can be made into a new table in TEAM-FF.

**Import Forcefield** converts forcefield files in other formats to the DFF PPF file format. It reads AMBER, CHARMM, CFF and DREIDING formats. In the **Importing Forcefield** Dialog, **Forcefield type** lists supported file formats. **Import from** lists the file location, and **Browse** is used to navigate to the file. When **Protect imported parameters** is selected, imported parameters will be frozen. When **Active parameters only** is selected, only required parameters of the selected model will be imported. **Save to** indicates where imported parameters are saved.

**Define Typing Rules** opens an **Atom Type Definitions** dialog to create or modify a typing rule file.



The typing rule file (.def) must be placed in the <dff-root>/data folder.  **New → Select** is used to open or create a .def file. The **Atom Type Definitions** dialog is divided into two main regions. The definition tree on the left shows the relationships between all defined atom types. The child nodes are derived from a parent node and they are more specifically defined than their parent is. The siblings are exclusively defined so that there is no ambiguity. Clicking any node of the definition tree displays properties of selected atom type on the right side of the dialog.

The definition tree also indicates how atom types are searched and compared. Each atom type is a node on the tree, and the order shown in the tree sets the precedence order for assigning atom types, with types at lower-order nodes taking precedence over higher-order nodes. **Move Up** and **Move Down** moves an atom type up or down at the same level. A **Validate** button is used to check for conflicts between the selected definition and its sub-definitions.

An atom type definition must specify properties of the center atom and its neighboring atoms. These properties are also displayed in a small tree-structure at the right side of the **Atom Type Definitions** dialog.  The center atom is the root node, while neighboring atoms are subnodes. Clicking on the root node or any subnode to review its properties. The **Element** pull-down displays the element of selected node, which can be changed. Once an element is selected, checking **Exclude selected element** sets the selected element as exclusive (every element but this one). **Order** sets the bond order of a subnode to its super-node. For the root node, **Order** is grayed out. **Connect** indicates how many subnodes a selected node has. **Add Conn** or **Del Conn** increases or decreases available subnodes. **Aromatic** can be set to **Yes, No or?** (Unknown); **Ring** indicates ring number, and **FormalCharge** sets the formal charge. The **Atom Type** section lists the **Name** of the defined atom type, and **Comment** provides a description of the definition to be saved in the .def file.

After a definition is specified, **Add** is used to add a new type to the definition tree. **Delete** removes a type from the definition tree. **Update** applies any changes to a selected node’s attributes. **Save** writes the revised definitions to the .def file. **Close** quits the dialog without saving.

**Define Forcefield type** opens a dialog to define a new force type type. In this dialog, **Type Name** can be any string representing the type to be defined. Functional terms can be specified using **Select** button, which open a **Select Function Term** dialog listing all available terms. There are two panels, **Basic Term** and **Cross Term**. By default, none of the checkboxes are selected. A function term can be set by clicking its checkbox and selecting a functional form from the pull-down list. After selecting terms, close the dialog to save the new type definition. You may use it by creating a new **USERDEFINE**-type forcefield.

**Modify Forcefield type** opens a dialog to modify user defined forcefield types. **Modify** changes the definition. **Delete** deletes the definition.

## 4-9. Parameters Menu - Fit and validate forcefield parameters

### Fit QM Data

To use the following parameter fitting functions, QMD files should first be loaded and associated with corresponding MSD files. Note that all forcefield related operations, such as assigning atom types, charges, etc., are performed on the MSD files. Make sure the forcefield is associated with appropriate MSD files.

**Fit Charge Data** is used to fit charge parameters from QM atomic charges. It opens the **Make Charge Parameters** dialog. **Data source** indicates what data is used for deriving charge parameters. **Scaling factor** scales the data. **Available models** lists selected models. **Reset** reloads the available models, and **Remove** removes models from the list. **Show comparison of charges** displays fit results with original data. **Parameter model** fits charges based on atom types (**Atom type based**) or bond increment parameters (**Bond type based**). **Execute** starts the fitting procedure.

**Fit Energy Data** is used to fit valence parameters from QM data via a dialog with three tabbed panels. Prior to using **Fit Energy Data,** charge parameters should be prepared and atom charges should be assigned.

In the first panel, **Job & Data**, **Data to fit** lists QMD models to be used. **Reset** and **Remove** can be used to modify data models to fit. **Weighting factors** are applied to **Energy, Gradient**and**Hessian** respectively. The default values are 100, 10 and 1. **Apply Boltzmann Factor** applies only to energy. **FF Type** is determined by the associated forcefield to be parameterized. **Job Name** assigns a name to background jobs. The input forcefield should be entered into **Initial FF**. The fit result will be written into **Result FF**. The job, initial forcefield, and result forcefield names are numbered sequentially; with each iterative fit, the suffix number increases by one, and previous result forcefield becomes the new initial forcefield. Thus, the fit may be repeated multiple times. If a new fit needs to be started, click **Start Over** to reset the job sequence. By default, the initial forcefield is the forcefield associated with the QMD files to be fit, which is listed in the **Forcefield** text field in the toolbar and can also be viewed by clicking **View Forcefield**.

The second panel, **Procedure**, provides options to set fitting procedure and relevant parameters. **Auto Procedure**, which uses DFF expert system to fit the data, is selected by default. To disable automatic fitting, select either **Levenberg-Marquardt** or **SVD**. Note the SVD method only fits linear terms. For the Levenberg-Marquardtmethod, the **Maximum iteration** number and the **Convergence** threshold can be specified. For the SVD method, the **Filter threshold**, which controls redundancy in the parameter space, can be adjusted.

The third panel, **Parameters**, allows the user to set adjustable parameters used during the fitting procedure. **Constraints** contains options used to place constraints on the parameters, which are useful especially when data is inadequate for deriving parameters. **Auto-reduce coupling terms** reduces redundant torsion terms in a forcefield. **Freeze nonbonded terms** is the default for fitting intramolecular properties. **Freeze out-of-plane terms**reduces adjustable parameters which may be redundantly used in a forcefield. **Use wildcards in torsion terms** is another way to reduce the number of adjustable parameters in the torsion terms. Three options are available for the **Torsional terms:** **Default, Maximum,**and **Minimum**. The **Default** option allows the system to automatically decide the number of torsion parameters depending on the number of data points. The **Anharmonic terms** can be **Frozen to zeros, Relaxed,** or **Derived.** The **Derived** option, anharmonic parameters are derived (not fitted) from harmonic force constants.  A **Penalty factor** is used by the implemented conditional minimizer to control parameters related to bond lengths and angles. If the **Check and fix parameter** option in the **Parameter correction** section is selected, parameters will be checked against two criteria: estimated variance and specified range. If the test fails, the corresponding parameter will be fixed. On the right side, **Additional Energy Terms** may be added by clicking on corresponding terms. Note that the terms available are related to the forcefield type selected.

**Fit VDW Dimer** derives VDW parameters by fitting QM data for molecular dimers. The data requirement for this function is slightly different from that for fitting molecular conformers. The QM data must be prepared for the dimer and for all monomers. For dimers, monomers configurations in different separations should be included in addition to optimized structures. This function only uses energies. Relationships between loaded models must be specified before fitting. To use this command, select the QM models, associate an initial forcefield, relax at least one of the LJ parameters, and open the **Fit Cluster Configuration** dialog.

This dialog contains two tabbed panels. The **Job&Data** tab has two parts. On the left side, **Data to fit** lists selected models. Select the dimer model and click **Cluster**. Then select the monomer models and click **Monomer**. If the cluster contains only one monomer type, repeat the operation twice. A number after an asterisk (\*) will tell you how many times the monomer used. **Reset** clears previous settings, and­ **Remove** clears any models that are not required. The right side is very similar to that used in the **Fit Energy Data** dialog, and the **Procedure** panel is the same as that used in the **Fit Energy Data** dialog*.*

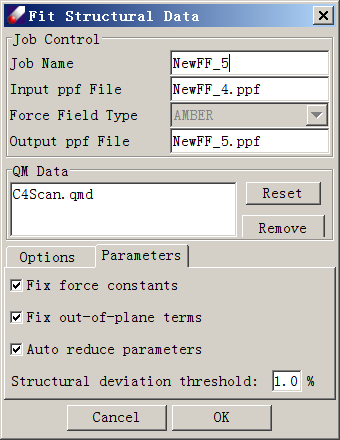
### Fit Liquid Properties

**Fit Liquid Properties** fits VDW parameters fields from liquid data. A liquid model is required instead of a single molecule. Use the **Build → Liquid Bulk** command to build a liquid box and ensure it has correct valence parameters. Then relax some nonbond parameters in the forcefield editor. During the fitting of valence parameters, nonbond parameters are normally fixed. Then click **Fit Liquid Properties** to open a **Fit Liquid Data** dialog. **Name, Input FF**and**Output FF** are used to identify files used in the background job. **Run NVT** generates the reference state using MD simulation. If you want to use a saved trajectory file from a previous simulation instead, click **Use saved trajectory**. If **Run NVT** is selected, enter a target temperature and click Opt. to examine and modify simulation conditions. If using a saved trajectory, use **Browse** to find the trajectory file. In the **Data to fit** section, select liquid data to fit into **Pressure(MPa)** and **Hv(kcal/mol)**. The temperature and pressure should correspond to the condition at which the density and heat of vaporization are to be measured.

Since only two observable data forms are used to optimize VDW parameters, the number of adjustable variables can be reduced either by freezing some VDW parameters (for example, freezing some parameters to previously optimized parameters) or by using two scaling factors (for Lennard-Johns well depth and radii, respectively). To use the latter method, select **Optimize scaling factor**. Execute starts the job.

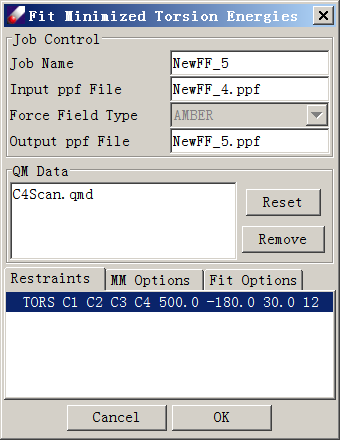
### Fit Molecular Properties

**Fit Structures** fits molecular structural data, equilibrium bond lengths and angles. This command mimics an empirical fit in which force constants are frozen and structures are reproduced by adjusting the most sensitive parameters. This method uses a finite differentiation method to calculate the derivatives of structural properties with respect to adjustable parameters and fits structural properties using a least squares method. In order to use this command, parameters for bonds and angles must first be relaxed. This command opens the following **Fit Structure Data** dialog:



In the dialog, **Job Name, Input ppf File, Forcefield type**and**Output ppf File**are used to identify files for the background job. **QM data** lists selected QM model. **Options** are maximum iteration and step size which controls how the fit is performed. In the **Parameters** panel is used to set adjustable parameters. In this panel, **Fix force constants, Fix out-of-plane terms,**and**Auto-reduce parameters**are useful to reduce the adjustable parameters. For the **Auto-reduce parameters** option, the ***Structural deviation threshold*** is used to eliminate the adjustable parameters for bond lengths and angles with deviations lower than the threshold.

**Fit Torsion Energies** fits minimized torsional energy profiles only. This command uses numerical differentiation to calculate derivatives of the minimized torsional energy profile with respect to the adjustable parameters, and then uses the derivatives in a least squares fit. **Job Name, Input ppf File, Forcefield type**and**Output ppf File**are used to select files for the background job. **QM data** displays the selected QMD file, which must contain the minimized conformational energy data.



There are three tabbed panels. **Restraints** lists the scanned torsion angle, identified by atom names, force constant, starting value, interval, and data point. Excepting the force constant, which is set to a default of 500, these values are taken from the QMD file and are consistent with QM calculations. **MM Options** are parameters for constrained energy minimization. **Fit Options** specify differentiation step size and fit maximum iteration, and includes an option to reduce the coupling torsional parameters.

### Additional Tools

**Analyze Fit** opens a **Fit Analysis** dialog used to view the .dft file, which records fit results. The **Fit Analysis** dialog contains two tabbed panels. The first, **Fit Merits (i/o)** shows deviations between input and fitted energies, gradients, and Hessian matrix elements. The second, **Optimized Results (i/o)**, displays comparisons between optimized structures and vibrational frequencies. The result forcefield is used to minimize molecules and compare results against input data for optimized structures. Two components are displayed in each of the panels. On the left, a table lists standard deviations; on the right, the correlation of input to fitted data is displayed graphically. Clicking column headers will display corresponding correlation charts for each property. Clicking any cell in the table will display the property correlation chart for that specific model. When input and fitted data agree well, all data points should be located on the diagonal line of the correlation chart. Clicking on **Data** opens another window to display the data numerically. It also displays the difference between the **Input** and **Fit** results. From this window, the data can be saved as a comma delimited (.csv) file for export to other spreadsheet applications such as Microsoft Excel.

**Estimate Charge** applies the QEq method to estimate charge parameters for selected models. To use this command, a forcefield must first be created and associated with the models. After selecting the models from the **Project Navigator** and clicking **Estimate Charge**, a dialog shows the input and output PPF files. After execution, the output PPF file will populate with estimated charge parameters, and partial charges are assigned to models.

**Estimate Parameters** opens an **Estimate Forcefield** dialog to estimate valence and nonbond parameters for a given molecule. In this dialog, you may select the models you want to use to estimate the parameters in the **Model** panel and assign output forcefield names. If **Save active parameters only** is checked, the output forcefield will not contain any parameters not used by the selected models even if they exist in the input forcefield. If **Protect existing parameters** is checked, the input parameters will not be changed and only missing parameters will be created. Otherwise input parameters may be overwritten by estimated parameters. Check **Use wildcard in torsion terms** to create torsion parameters with wildcards on its end atoms. Check **Use anharmonic terms** to estimate anharmonic parameters.

**Base Parameters** opens a **Default Atom Type Properties** to modify the base parameters of atom types. Input the atom type to be modified in **Default Atom Type** and click **Load** to view the base parameters. You may change the data in the dialog and click **Save** to save the parameters.

## 4-10. QMD Menu - Compute and Organize Quantum Mechanics Data

### QM Calculations

**Z-Matrix** opens a **Make Z-matrix** dialog for *Gaussian* calculations. With this dialog open, **Left-Click** to select a ***Starting atom*** and click **Make** to make a new Z-matrix. The Z-matrix can then be edited as necessary. Click **OK** to save it.

**Compute** opens an interface to prepare input files for QMcalculations. If Gaussian is installed on the same computer, it can be launched directly from the interface.

The top section lists the models selected. "Computation" provides common options used in the preparation of parameterization data.  The ***Job name*** will used to identify this background job. The ***Use Z-Matrix*** option is useful for dealing with difficult structures, and ***Check Z-Matrix*** can be used to modify the Z-matrix. ***Scan Dihedral*** can be used to calculate a minimized conformational energy profile. ***Set*** specifies which dihedral angle will be rotated. With this option, a series of minimization jobs will be launched. Note that this option should be used with the minimized structure.

In the **Options** section, computation variables including method, basis set, total charge, spin multiplicity, CPU number (only for shared computers), maximum CPU time allowed, and memory limit can be specified. If the molecule to be calculated is an ion, the charge and spin multiplicity will be assigned accordingly.

In the **Job** section, computation logistics are specified. ***Make input files only*** generates input files without running calculations, while ***Run background job*** starts the calculation and loads results onto the same computer. If a background job is begun, there are three options for results. ***Append data to a QMD file*** converts and saves all calculated data into one QMD file. ***Make a QMD file for each calculation*** makes a QMD file for each model calculated. ***Do not generate QMD file*** does not convert the QM output to QMD. As default, original QM output files are saved. ***Package*** is used to set the path to the *Gaussian* executable, which can be located using **Browse**.

### Quantum Mechanics Data (QMD)

**Organize QMD** starts an **Organize Data** dialog allowing the user to view, manage, and output QM files *(.qmd)*. From version 7 onward, DFF uses the .*qmd* file format for fitting forcefield parameters and does not directly use *GAUSSIAN* output files. **Organize QMD** can also be used to examine and organize QM data.

A screenshot of a computer

Description automatically generated

In the **Organize Data** dialog, data for each structure is listed in spreadsheet form. **Open** is used to load GAUSSIAN or DFF output files in .qmd format. Files of the .bdf format can also be loaded. Files already in the Project Navigator can be selected and directly loaded by clicking **QMD Organizer**. ***Name*** displays the file name of each structure. Note that each file may contain multiple conformers, which are labeled in the ***Conform*** column. ***Energy*** lists the absolute energy and ***R Energy*** lists the relative energy. ***Optimized*** indicates whether a structure is fully optimized, and ***E' , E"***  are first and second derivatives. ***ESP Field*** indicates whether potential values have been calculated. ***Charges*** lists atomic charges available in the QM data files. In order to derive parameters, at least one conformer must be optimized with E', E" and one set of charges calculated. Selecting one or more lines activates several command buttons in this dialog. **Save** writes .qmd files, **Remove** deletes selected line from the list, and **Energy** calculates and displays relative energies.

## 4-11. Help Menu – Information, License, and Document

**About** displays the software's version and copyright information.

**Instant Help** provides additional help on some dialogs. This is complementary functionality, it is not shown on every dialog, but some of them that are not very intuitive to see the imput parameters. When you see this icon (), click it to toggle the display page.

**User Guide** will load and display this document in pdf format.

**References** lists the publications directly relevant to DFF and TEAMFF.

**License** allows the user to view and manage license information. **Agreement** shows software instructions. A license file may be requested using **Request** and installed using **Install**. Server opens a connection to a license server.

## 4-12. Toolbar Commands

The toolbar is right below the command menus.

### Toolbar File Commands

The New (E:\DFF7\Release7.1\images\newFile.gif), Add ( E:\DFF7\Release7.1\images\openFile.gif) and Save (E:\DFF7\Release7.1\images\saveFile.gif) commands are used to make a new model, add models from existing files, and save models.

### Toolbar Sketching Commands

*Some of editing functions are provided via the tool bars.*

**Undo (**E:\DFF7\Release7.1\images\undoFile.gif**)** can reverse most editing actions.

**Redo** (E:\DFF7\Release7.1\images\redoFile.gif) restores actions that have been undone.

**Select Element** (  ) is a pull-down menu listing elements that can be added to the selected model. Only the most common elements are listed explicitly, but clicking "..." will open a periodic table dialog, from which any element can be selected and added.

On the periodic table, the selected element is marked in yellow. The **Property** command opens the **Element Properties** dialog, where an element's properties are displayed and can be edited. **Default** reverts an element's properties to those preloaded in **DFF**. *Note that once the new values are saved, they become the default value.*

A screenshot of a computer

Description automatically generated

*The following buttons toggles the* ***Mouse Left-Click*** *function. Once it is activated, left-click on the main windows performs the task. To deactivate the function, click the button again.*

**Add Atom** () sets the **add** atom or bead function. Clicking in the window will add an atom or a bead.

**Delete Atom** () sets the mouse to **delete**. Clicking on an atom will delete it from the model.

**Replace Atom** () sets the mouse to **replace** function. Clicking an existing atom will replace that atom with the selected element.

**Move Atom** () sets the mouse to **move** function. **Clicking and dragging** on an atom will move it. Note that the move is a two-dimensional transformation.

**Add Hydrogen** () automatically adds hydrogen atoms to the model. The number of hydrogen required is calculated based on the most probable chemical structure for each atom.

**Del Hydrogen** () automatically removes all hydrogen atoms.

**Add Bond** () sets the mouse function to **add bond(s)**. Click any two atoms to add a bond between them. Repeating this operation changes the bond order.

**Delete Bond** () sets the mouse function to **delete bond**. Select a bond to delete it.

**Refine Structure** () is used to optimize the model. This command approximates a forcefield for the model and optimizes the structures automatically.

**Set Bond Order** () sets the mouse function to **set bond order** status. Clicking on a bond change its bond order to selected bond order.

**Select** E:\DFF7\trunk\images\select.gifsets the mouse function as a selector. If this button set on, left-click on an atom will pick up the atom.

**Rotate** E:\DFF7\trunk\images\rotate.gif sets the mouse to rotate an object on the screen.

**Translate** E:\DFF7\trunk\images\translate.gifsets the mouse to translate an object on the screen

**Zoom** E:\DFF7\trunk\images\zoom.gif sets the mouse to zoom in or zoom out

**Reset** E:\DFF7\trunk\images\center.gif restores the original position of an object

### Additional Shor-cut Keys on Atoms

*In addition, when the* **Select** *button is activated, mouse has more functionality as follows.*

**Select one or more atoms** – Press and hold the **Left** button and drag over a region on the screen will select all atoms in the region.

**Move selected atoms** – While the atoms are selected (marked by dots), press and hold the **Right** button and drag to move the atoms selected.

**Copy** – Ctrl+C collects the selected atoms

**Delete** – push the Delete key on keyboard

**Paste** – Ctrl+V puts down the copied atoms

**Undo** – Ctrl+Z reverse last action

**Redo** – Ctrl+Y reverse undo

## 4-13. Mouse Commands on Project Navigator

These commands are available by **right-clicking** on nodes of the *Project Navigator*. The available commands depend on the node type. Some commands operate identically to those in the command menus or toolbar menus, some are only accessible here.

One important function is to select nodes for subsequent operation. To select a single node, left click on the node, the node is colored. To select multiple adjacent nodes, push **Shift** key on the keyboard and clickon the beginning and end nodes. To select multiple non-adjacent nodes, use **Ctrl** key instead.

#### On Root Node

The root node is the project file “.pjf”. It has only two functions to make the Project Navigator easy to read.

**Larger Font** increases the font size of Project Navigator

**Smaller Font** decreases the font size of Project Navigator

*Note that the font size only applies to the Project Navigator. To change the font size for all components, use* ***View→Larger Font*** *and* ***View → Smaller Font*** *in command menus.*

#### On Project Node

The project node corresponds to the project folder on computer file system. It has four basic functions to manage folders and files in the project.

**New** creates a new molecule model or a new folder

**Sort** rearranges the order of files and folders by name, date and type.

**Refresh** reload all folders and files under this folder.

**Open in Explorer** open the file folder in Windows explorer.

#### On Folder Node

The folder node is where the jobs are conducted, it has more functions to manage the files and sub-folder.

**New** creates a new model or new folder

**Open** shows the selected file in MSD, PPF, QMD, LOG, etc. format.

**Cut** puts this folder into clipboard for copy, will remove this folder.

**Copy** puts this folder into clipboard for copy, will not remove this folder.

**Paste** puts a model or file to current folder.

**Delete** remove this folder.

**Rename** changes the name of the folder.

**Sort** rearranges the order of files and sub-folders in the selected folder.

**Property** list folder location, etc.

**Refresh** reload all folders and files under this folder.

**Open in Explorer** open the folder in Windows Explorer.

#### On Model Node

The model node is relevant to a file, the functions are.

**New** creates a new model or new folder

**Open** loads various files, MSD, PPF, QMD, LOG, etc.

**View** display the selected models in a separate window

**Cut** puts this folder into clipboard for copy, will remove this folder.

**Copy** puts this folder into clipboard for copy, will not remove this folder.

**Paste** puts a model or file to current folder.

**Delete** remove this model.

**Rename** changes the name of the model.

**Save as** to save the model in a new name.

**Associate PPF** and **Dissociate PPF** are used to associate or dissociate a forcefield with the selected model(s). Available forcefields in current or other folders are listed. Once a forcefield is chosen, it is assigned to the selected model(s).

**Associate MSD** and **Dissociate MSD** are commands used to associate or dissociate a topological data file (MSD) with the selected QMD file.

**Create MSD** is activated only for QMD file and adds a MSD file to be associated with the selected QMD file.

# CHAPTER 5 – Executable PROGRAM COMMANDS

DFF package includes eight (8) executable programs that perform various tasks.

DFFDB – database operations on force field database TEAMFF.

DFFEXP – export force field and prepared models to external simulation engines

DFFQMD – operations on QM computations.

DFFFIT – fit engine for fit parameters of QM and liquid data.

DFFMD – MD engine for validation.

DFFMM – MM engine for validation.

DFFJOB – background engine working with JAVA and Python interfaces.

DFFLSV – License management

These programs can be executed as background jobs. The programs are operated by using three types of commands: single [S], array [A] and dictionary [D].

1. Single [S] command requires no parameter.
2. Array [A] command requires a list of parameters.
3. Dictionary [D] command requires a list of parameters as KEYWORD=VALUE.

Syntax:

1. Brackets (**<…>**) indicates contents to be filled.
2. A vertical divider (**|**) indicates alternative values.
3. Bold face indicates mandatory entries. Normal face indicates optional entries. Optional entries may have default values.

The commands are object oriented: they are designed for objects: model, forcefield, database, and computations including QM, fit, and validation. Each program uses some of the objects, which are indicated as follows.

## 5-1. Model Commands

Model commands are used for managing model files, including read, write, and modification, which are used in all programs except DFFLSV.

**$MODEL=$OPEN [A]**

**<list if model files>**

**$End**

The command opens models to the program. Each line is a model file, which can be in any of the supported file formats: **msd, pdb, mol2.**

**$MODEL=$SAVE [A]**

**<list of msd files>**

**$END**

This command saves models in MSD format.

The list of msd files is for the models in the order as opened. The file base names can be different from the original files.

**$MODEL=$SAVEALL [S]**

This command saves all models to MSD files. The file names share the same base name as loaded, but with extension of “.msd”. Note that if the original file is in MSD format, it will be overwritten.

**$MODEL=$CLOSE [S]**

Close all opened models (any changes made to the models will be lost unless saved).

**$MODEL=$SETTYPE: [D]**

**TYPINGRULE = <typing rule>**

**$END**

Assign atom types of opened models using a specified typing rule.

<typing rules> can be reserved words: DEFAULT or saved script files. DEFAULT is recommended, more flexible typing rules are scripted documents saved in /data/TypingScripts folder. If the definition of script file is an .ext file, the DEFAULT is applied first, followed by the extension definitions.

For example:

$MODEL=$SETTYPE

RULER=TEAM.ext

$END

resets the atom types of models in memory according to AMBER rules.

**$MODEL=$SETFC: [D]**

REMOVE = FALSE | TRUE

RESET = TRUE | FALSE

**$END**

This command sets the formal charges of opened models.

Both parameters are optional.

REMOVE – to remove all formal charge values and stop. The default is FALSE.

RESET - This is an option that deals with existing values. If TRUE, all existing values will be erased and then recalculated. If FALSE, the existing values will be kept, only atoms without formal charge values will be calculated. The default is TRUE.

**$MODEL=$GUESSBOND: [S]**

Build or rebuild connectivity and then calculate bond orders. The existing connectivity will be erased and recalculated first and then recalculated using the atom-atom distance.

**$MODEL=$GUESSBONDORDER: [S]**

Calculate bond orders only, the existing connectivity is kept.

## 5-2. Force Filed Commands

Force field module is used in most forcefield related programs: DFFDB, DFFJOB, DFFFIT, DFFMM and DFFMD.

**$FORCEFIELD=$OPEN [A]**

**<list of ppf files>**

**$END**

Loads a list of forcefield ppf files. Each line represents a forcefield file. The $Open command is applied as default so it can be omitted.

Although multiple force fields can be loaded, some programs limit the operations on one force field only.

**$FORCEFIELD=$CLOSE: [S]**

Closes the opened forcefields.

**$FORCEFIELD=$SETCHARGE: [S]**

Using the force field to set the partial charges of models. The model must be open.

**$FORCEFIEL=$TEMPADJ [D]**

**TEMP=<K>**

**$END**

Adjust the LJ-12-6 parameters using temperature. The only parameter is temperature in Kalven.

## 5-3. Export Commands

These commands are used for exporting models and force fields to external simulation engines. To use these commands, the models and force fields must be loaded first.

**$EXPORT=$GROMACS [A]**

**TEMPLATE** = <template file>

**BASENAME** = <file name>

**$END**

TEMPLATE – The template is used to generate GROMACS input files. The file is saved in the /data folder and can be customized for repeat uses.

BASENAME – The base name is used for generating files for GROMACS.

**$EXPORT=$LAMMPS [A]**

Export model and force field to LAMMPS. The same parameters as explained above:

**TEMPLATE** = <template file>

**BASENAME** = <file name>

## 5-4. TEAMFF Database Commands

**$DATABASE=$CREATE: [D]**

**FILENAME=<database file>**

**TABLE=<table name>**

**TYPE =<force field type>**

**TYPINGRULE=<typing rule>**

**NOTE=<description>**

**$END**

This command creates a new database with one table. Once the database is created, additional tables can be added.

The parameters are:

FILENAME – the full path to which the database file is saved

TABLE – the name of the created table. You can add new tables after the database is created.

TYPE – is the force field type, which can be AMBER, TEAM, CHARMM, CFF, DREIDING, etc.

TYPINGRULE – is the file of typing rules used for the force field table created.

NOTE – is appended to the table.

**$DATABASE=$OPEN: [D]**

**FILENAME=<path to the database file>**

**TABLE=<list of tables to be opened>**

**$END**

Opens the database and one or multiple tables.

FILENAME – is the full path to the database file.

TABLE – a list of the tabled to be opened, separated by comma “,”. The order of the tables is used for Checkout and Assign commands.

**$DATABASE=$CLOSE: [S]**

Closes all tables.

**$DATABASE=$CHECKOUT: [D]**

**OUTPUT=<ppf file name>**

**PAIRWISE=< FALSE|TRUE >**

**TRANS=<FALSE|TRUE>**

**$END**

This command checks out parameters from the tables opened for the model opened.

All parameters are optional.

* OUTPUT – save the ppf file, if not given, the force field will not be saved, but kept in the memory.
* TRANS – if parameters are not found, search similar parameters as transferred. The default is FALSE.
* PAIRWISE – use explicit pairwise parameters if exist. The default is FALSE.

**$DATABASE=$ASSIGN: [D]**

OUTPUT=<ppf file>

TRANS=<FALSE|TRUE>

AUTO=<FALSE|TRUE>

PAIRWISE=< FALSE|TRUE >

ATMAP=<atm file >

FRAGMENT=<path to the folder of generated fragments>

**$End**

This command uses one or multiple forcefield tables to assign force field parameters for models opened. Note that the order of tables opened is sensitive, parameters will be taken from the tables as specified, only when the parameters are not complete the next table will be called.

All parameters are optional.

* OUTPUT – save the ppf file, if not given, the force field will not be saved, but kept in the memory.
* TRANS – if parameters are not found search and transfer similar parameters. The default is FALSE.
* AUTO – if parameters are missing, generate automatic parameters. The default is FALSE.
* PAIRWISE – use explicit pairwise parameters if exist. The default is FALSE.
* ATMAP – save the mapping scheme of atom types between the unified atom types and atom types used in each table. If not specified, the file will not be saved.
* FRAGMENTS – the folder that saves the generated fragments which can be used to augment the parameters.

## 5-5. Fit Commands

[To be completed]

**$COMPUTE** sets what is to be calculated. Valid entries are as follows:

**FIT** specifies fit tasks

**FIT = AUTO | CLUSter | CONFormer | LINEar | NONLinear | TORSion | STRUcture | OPTImized | VALIdation**

AUTO - use the automatic procedure,

CLUS - fit molecular clusters (dimers)

CONF - fit conformers

LINE -use linear fit routine (SVD)

NONL - use nonlinear fit routine (LMM)

TORS - fit minimum torsion profiles

STRU - fit structures only

OPTI - fit optimized structures only

VALI -validate the fit

## 5-6. MM Computation Commands

[to be completed]

**CALCULATE** is used for energy calculations only

**CALC = ENERgy | GRADient | HESSian**

ENER - energies only

GRAD - include gradients

HESS - include Hessian matrix elements

**OPTIMIZE** is used for energy minimization

**OPTI = YES | NO**

**INITIALIZATION** estimates initial parameters without doing any calculations

**INIT = YES | NO**

**SETTYPE** is used to assign atom types only.

**SETTypes = NO | AUTO | <definition file >**

NO - do not assign types

AUTO - assign atom types using default typing rules.

<definition file> - a definition file with an extension of ext or .def, which must be saved in the <dff-root>/data folder.

**SETCHARGE** is used to set atomic charges only

**SETCharges = YES | NO**

The **COMPUTE** command must be ended with **$END**

**$END**

**$MODEL**

**E:\DFF\_DEV\QMDFF\_Methods\SnMcCl2\SnNcCl2\DFF.msd**

**$END**

**$FORCEFIELD**

**TYPE = AMBER**

**INPUT = none**

**USE\_CUTOFF = YES**

**NBMETHOD = ATOM\_BASED**

**CUTOFF = 6.0**

**BUFFER = 0.5**

**TAIL\_CORRECTION = NO**

**$END**

**$MMOPTIONS**

**CONVERGENCE = 1.0E-10**

**MAXSTEPS = 100**

**METHOD = BFGS**

**$END**

**$PRINT**

**CONSOLEXYZ = NO**

**$END**

**$RESTRICTION** is an array command, and each entry specifies a torsion dihedral angle. The syntax for each entry is

**TORS <atom\_1> <atom\_2> <atom\_3> <atom\_4> <fc> <t0> <dt> <n>**

This command is used only with **OPTI = YES** in the **$COMPUTE** command. **<atom\_i>** is an atom name of the model, and **<fc>** is a force constant used to fix the dihedral angle, usually 100-500 kcal/Mol/A2. **<t0>** is the initial dihedral angle, **<dt>** is the increment for each step, and **<n>** is the number of steps.

**$END**

$SYSTEM

ROOT = E:\DFFX\trunk

$END

$MODEL

P902993.msd

$END

$FORCEFIELD

TYPE = AMBER

INPUT = none

EQUI = DEFAULT

USE\_CUTOFF = YES

NBMETHOD = ATOM\_BASED

CUTOFF = 5.0

BUFFER = 0.0

TAIL\_CORRECTION = NO

$END

$MDJOB

VRGSTEPS = 100

ENSEMBLE = NVE

TIMESTEP = 1.0

RUNSTEPS = 100

TEMPERATURE = 1.0

TCONTROL = VELOCITY\_SCALE

TWINDOW = 10.0

PEEK = 1000

$END

$PRINT

COORDINATE = NO

CONSOLEXYZ = NO

$END

## 5-7. QM Computation Commands

[to be completed]

$MODEL

INPUT = methanol.msd

OUTPUT = methanol.msd

$END

$TOOL

EXE = C:\G03W\g03.exe

BASIS = 6-31G\*

METHOD =B3LYP

NPROC = 8

MEMORY = 800MB

CHARGE = COMPUTE

MULTIP = COMPUTE

$END

$OPT

TASK = EXECUTE

SAVE = methanol.qmd

$END

$SCAN

TASK = INPUT

LOAD = methanol.qmd

SAVE = methanol.qmd

STEPSIZE = 10.0

$END

$FREQ

TASK = INPUT

LOAD = methanol.qmd

SAVE = methanol.qmd

$END

# CHAPTER 6 – FUNCTIONAL FORMS

## 6.1 Forcefield types

The forcefield type refers to a fixed combination of forcefield functions, each describes a particular interaction term. It does not include any parameters and atom types. DFF supports common forcefield types including AMBER, CHARMM, CFF, DREIDING. In addition, a new forcefield type, TEAM, is created and tested for better performance.

Electrostatic interactions are calculated from atomic charges.

For each atom, the charge is calculated by:

There are two parameters, atom type charge (ATC) **,** and bond increment (BINC) ATC is particularly useful for ionic species, BINC describes a charge separation of a chemical bond between two atoms i and j.

Two types of LJ functions, LJ-12-6 and LJ-9-6 are used in forcefield types, LJ-9-6 is used in CFF only, LJ-12-6 is the most popular function.

The 1-4 nonbond terms are scaled as follows:

*AMBER: LJ is scaled by 1/2; electrostatic is scaled by 1/1.2*

*TEAM: electrostatic is scaled by 1/2.*

The following table lists functional terms used in each forcefield type along with sample parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| **FF Type** | **Term** | **Atom Types** | **Parameters** |
| All | ATC | o\_1-/2 | -0.5 |
|  | BINC | c\_4, o\_1 | 0.2360 |
|  |  |  |  |
| TEAM | BDQUA | c\_2, c\_2 |  |
|  | ADQUA | c\_2, c\_2, c\_4 | 179.9511, 53.8431 |
|  | TBCOS | c\_3, c\_3, c\_4, h\_1 | 1, -0.1978, 2\*, 1.0728, 3\*, -0.0947 |
|  | TBCOS | c\_4, c\_3, c\_3, c\_4 | 2\*, 5.4661 |
|  | OHARM | c\_3a, c\_3o, c\_4, o\_1 | 0.0000, 44.0651 |
|  | N12\_6 | c\_3 | 3.9800, 0.0760 |
|  | C\_B\_B | c\_2, c\_2, c\_4 | 8.6266 |
|  | C\_B\_A | c\_3, c\_4, o\_2 | 51.9669 |
|  |  |  |  |
| CFF | BQUAR | c\_2, c\_2 | 1.2107, 1212.6459, -2425.2919, 2829.5072 |
|  | AQUAR | c\_2, c\_2, c\_4 | 179.8087, 53.8431, -10741.8351, -2569.8855 |
|  | TCS3M | c\_2, c\_2, c\_4, c\_4 | 0.1000, -0.2500\*, 0.1200 |
|  | OHARM | c\_3, c\_3, c\_4, c\_4 | 0.0000\*, 2.3488\* |
|  | N\_9\_6 | c\_2 | 3.9015\*, 0.0819\* |
|  | C\_B\_B | c\_2, c\_2, c\_4 | -4.6381 |
|  | C\_B\_A | c\_34, c\_44, h\_1 | 21.4453\* |
|  | BINC | h\_1p, o\_2 | 0.3361\* |
|  |  |  |  |
| AMBER | BHARM | c\_2, c\_2 | 1.2057\*, 1276.7505\* |
|  | AHARM | c\_2, c\_2, c\_4 | 179.9975\*, 21.7689\* |
|  | TCOSP | c\_3, c\_3, c\_3a, c\_3a | 180.0\*, 0.9349\*, 2\*, 0.0\*, 0.0968\*, 3\* |
|  | IBCOS | c\_3, c\_4o, c\_3, h\_1 | 180.0000\*, 0.0000 , 2.0000\* |
|  | N12\_6 | CC2H2 | 3.9300\*, 0.0660\* |
|  |  |  |  |
| CHARMM | BHARM | c\_2, c\_2 | 1.2057\*, 1276.7505\* |
|  | AHARM | c\_2, c\_2, c\_4 | 179.9975\*, 21.7689\* |
|  | TCOSP | c\_3, c\_3, c\_3a, c\_3a | 180.0\*, 0.9349\*, 2\*, 0.0\*, 0.0968\*, 3\* |
|  | IHARM | c\_34, c\_34, c\_44, h\_1 | 0.0000\*, 73.6481\* |
|  | N12\_6 | n\_3 | 3.6930\*, 0.1450\* |
|  | V14LJ | c\_2 | 3.8250\*, 0.0910\* |
|  |  |  |  |
| DREIDING | N12\_6 | c\_4 | 3.9300\*, 0.0660\* |
|  | BHARM | c\_4, c\_4 | 1.5283 , 213.6099 |
|  | ACOSH | c\_4, c\_4, h\_1 | 107.1210 , 54.6774 |
|  | TCOSM | h\_1, c\_4, c\_4, h\_1 | 0.0000\*, -0.1405 , 3.0000\* |
|  | OCOSH | c\_3, n\_3, c\_4, h\_1p | 0.0000\*, 10.0000\* |

### TEAM

The essential functions are:

Each term represents an energy contribution as function of bond length, bond angle, torsion dihedral angle, out-of-plane angle and nonbond atom-atom distance. The bond length is in Å; the bond angle is in *radian*; the dihedral angle is in *radian*. The averaged Wilson out-of-plane angle defined by four atoms (i-j-k-l) where i is the center atom, in *radian*.

The anharmonic parameters in bond and angle terms () can be optionally set to be:

* Zeros, the function is a harmonic function.
* Free variables.
* Derived from the harmonic parameter:

In the torsion term, the reference dihedral angle is fixed so that the energy is always 0 at :

The expansion is up to three terms with n fixed, therefore, there are three free parameters. Any integers are allowed for n, but usually n = 1, 2, or 3, in this case, the function becomes:

The nonbond parameters are given for like-atom types, for unlike atom types, the combination rule is:

The nonbond interaction functions apply to *intermolecular* atom pairs that belong to different molecules, or *intramolecular* atom pairs that belong to the same molecule but are separated by at least three consecutive bonds. In the case of only three consecutive bonds (1-4 interaction) between the intramolecular atom pairs, the electrostatic interaction is scaled by 0.5.

In addition, the CFF style cross-coupling terms are optionally used.

For the two bonds that form a bond angle (BB):

For each of the two bonds and the angle (BA):

For two angles with a common atom (AA):

For the center bond in a dihedral angle (BT)

For each of the two bonds in a dihedral angle (BBT)

For each of the two angles in a dihedral angle (AT)

For two angles in a dihedral angle (AAT)

### CHARMM

Functional form:



Combination rules for VDW parameters:



**Coordinates:**

*b* – bond length in Å.

** – bond angle in *rad*.

** – dihedral angle of four atoms bonded sequentially (i-j-k-l), in *rad*.

** – improper dihedral angle measuring out-of-plane motion for atoms (i-j-k-l) where i is the center atom, in *rad*.

*rij* – Nonbonded distance in Å. The atoms are either in different molecules or in the same molecule but separated by at least three consecutive bonds (1-4 and beyond interactions).

**Notes:**

1. May use separate 1,4 parameters for VDW

2. May use specific (ij and Rij, no combination rule) nonbonded parameters

3. Hydrogen bond term is not supported.

### AMBER

Functional form:



Combination rules for VDW parameters:

**Coordinates:**

*b* – bond length in Å.

** – bond angle in *rad*.

** – dihedral angle of four atoms bonded sequentially (i-j-k-l), in *rad*.

** – improper dihedral angle measuring out-of-plane motion for atoms (i-j-k-l) where i is the center atom, in *rad*.

*rij* – Nonbonded distance in Å. The atoms are either in different molecules or in the same molecule but separated by at least three consecutive bonds (1-4 and beyond interactions).

**Notes:**

Torsion parameters are divided by a factor of 2.

1-4 VDW terms is scaled by 1/2

1-4 electrostatic is scaled by 1/1.2

### CFF

Functional form:



Combination rules for VDW parameters:



**Coordinates:**

*b* – bond length in Å.

** – bond angle in *rad*.

** – dihedral angle of four atoms bonded sequentially (i-j-k-l), in *rad*.

** – improper dihedral angle measuring out-of-plane motion for atoms (i-j-k-l) where i is the center atom, in *rad*.

*rij* – Nonbonded distance in Å. The atoms are either in different molecules or in the same molecule but separated by at least three consecutive bonds (1-4 and beyond interactions).

**Note:**

1-4 VDW and electrostatic terms are not scaled.

### DREIDING

Functional form



Where the angle term depends on the geometry of the out-of-plane center. For non-planar geometry:



For planar geometry



Combination rules for VDW parameters:



**Coordinates:**

*b* – bond length in Å.

** – bond angle in *rad*.

** – dihedral angle of four atoms bonded sequentially (i-j-k-l), in *rad*.

** – improper dihedral angle measuring out-of-plane motion for atoms (i-j-k-l) where i is the center atom, in *rad*.

*rij* – Nonbonded distance in Å. The atoms are either in different molecules or in the same molecule but separated by at least three consecutive bonds (1-4 and beyond interactions).

**Note**: The Exp-6 form  and special hydrogen bond terms are not supported.

### GROMACS

Functional form:

Combination rules for VDW parameters:

**Coordinates:**

*b* – bond length in Å.

** – bond angle in *rad*.

** – dihedral angle of four atoms bonded sequentially (i-j-k-l), in *rad*.

** – improper dihedral angle measuring out-of-plane motion for atoms (i-j-k-l) where i is the center atom, in *rad*.

*rij* – Nonbonded distance in Å. The atoms are either in different molecules or in the same molecule but separated by at least three consecutive bonds (1-4 and beyond interactions).

## 6.2 Functions

The following potential functions are supported in Direct Forcefield:

### A) Valence Functions

Valence terms are defined by internal chemical coordinates - bond lengths, bond angles, dihedral angles, improper dihedral angles and out-of-plane angles. These terms are also referred as "bond terms" because the internal coordinates are defined by atoms that are all directly bonded.

#### Bond Potential Functions

The bond distance between atoms *i* and *j* is variable in these functions.

Harmonic (BHARM):

Quartic (BQUAR):

Derived Quartic (BDQUA):

Morse (BMORS):

Restrained Harmonic (BRHAR):

Where rc is a fixed (not adjustable) parameter specified by user.

#### Angle Potential Functions

In these functions, the angle between two adjacent bonds is the variable:

Harmonic (AHARM):

Quartic (AQUAR):

Derived Quartic (ADQUA):

Truncated Harmonic (ATRNH):

Where  is a fixed (not adjustable) parameter specified by user.

Screened Harmonic (ASCRH):

        Where *1* and *2* are user-provided fixed parameters.

Screened Vessel (ASCRV):

        Where r1 and r2 are user-provided fixed parameters.

Cosine (ACOS\_):

Cosine Harmonic (ACOSH):

#### Dihedral Angle (Torsion) Potential Functions

    This variable is the dihedral angle between two surfaces made by bonds i-j, j-k and bonds j-k, j-n, respectively:

Harmonic (THARM):

Cosine Harmonic (TCOSH):

Cosine (P) (TCOSP):

Cosine (M) (TCOSM):

Triple Cosine (P) (TCS3P):

Triple Cosine (M) (TCS3M):

Biased Cosine (TBCOS):

 Effective, it is

**IV. Improper Dihedral Angle Potential Functions**

The improper dihedral angle is measured as the dihedral angle of two surfaces made by bond pairs of j-i/i-k and n-i/i-k respectively.

Harmonic (IHARM):

Cosine Harmonic (ICOSH):

Cosine (P) (ICOSP):

Cosine (M) (ICOSM):

Triple Cosine (P) (ICS3P):

Triple Cosine (M) (ICS3M):

Biased Cosine (IBCOS):

#### Out of Plane Angle (OOPA) Potential Functions

    This variable is defined as the angle between a surface made by bond pair of j-j/j-n and the bond i-j:

Harmonic (OHARM):

Cosine Harmonic (OCOSH):

Cosine (OCOS\_):

### B) The Nonbond Functions

The nonbond terms refer to potential energy terms that describe interactions between two atoms that are not directly bonded, which include the intermolecular interactions and "1-4 and beyond" intramolecular interactions. "1-4 and beyond" means the atoms are at least separated by three connective bonds.

As default, the parameters of these nonbond energy functions are given for two atoms separated by distance (). The forcefield parameters are given in terms of atom types. By default, the parameters are given as like-atom types () in a forcefield, and the parameters of unlike atom types () are calculated using the combination rules.

The following functions and their combination rules are supported by DFF.

#### 1. Lennard-Jones (12-6) function

The LJ-12-6 function has three popular expressions. The first one which is used by DFF is written as:

Where corresponds to the minimum energy of -. Parameters are usually given by like-atom types, the Lorentz−Berthelot (LB) combination rule is used to calculate the parameters of unlike atom pairs:

The second expression of this function is

Where is the cross point where the energy equals zero, clearly we have

The third form is

Clearly, we have

#### 2. Lennard-Jones (9-6) Function

#### 3. general LJ-n-m function

#### 4. Buckingham potential

#### 5. Morse Function:

### C) The Cross Coupling Terms

Bond-bond (C\_B\_B)

Bond-angle (C\_B\_A)

Angle-angle (C\_A\_A)

Bond-Torsion (C\_B\_T)

Angle-Torsion (C\_A\_T)

Bond-Bond-Torsion(C\_BBT)

Angle-Angle-Torsion(C\_AAT)

Urey-Bradley 1-3 Coupling (UB\_13)

## 6.3 Functions and Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cat.** | **Key** | **Discription** | **Coordinate** | **Functional Form** | **Parameters** |
| Charge | ATC | Atom type charge parameter, a single value assigned to atom i | Atom  (i) |  | qi |
| BINC | Bond increment of charge from atom j to i (a positive value means atom i gets a positive charge) | Bond  (i, j) |  |  |
| VDW | N12\_6 | LENNARD\_JONES 12-6 function | Atom  (i) |  |  |
| N\_9\_6 | LENNARD\_JONES 9-6 function | Atom  (i) |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bond | BHARM | HARMONIC\_BOND | Bond  (i-j) |  |  |
| BQUAR | QUARTIC\_BOND | Bond  (i-j) |  |  |
| BDQUA | DERIVED QUARTIC\_BOND | Bond  (i-j) |  |  |
| Angle | AHARM | HARMONIC\_ANGLE | Bond angle  (i-j-k) |  |  |
| AQUAR | QUARTIC\_ANGLE | Bond angle  (i-j-k) |  |  |
| ADQUA | DERIVED QUARTIC\_ANGLE | Bond angle  (i-j-k) |  |  |
| ACOSH | COSINE\_SQUARED\_ANGLE | Bond angle  (i-j-k) |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Torsion | TBCOS | Biased (reference angles are fixed) Fourier expansion of cosine with plus sign (up to three terms in one line) | Dihedral angle  (i-j-k-l) |  |  |
| TCOSM | Fourier expansion of cosine with minus sign (up to three terms in one line) | Dihedral angle  (i-j-k-l) |  |  |
| TCOSP | Fourier expansion of cosine with plus sign (up to three terms in one line) | Dihedral angle  (i-j-k-l) |  |  |
| TCS3M | Fixed three-term Fourier expansion of cosine with minus sign. | Dihedral angle  (i-j-k-l) |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Out-of-plane | OCOSH | Harmonic function of cosine for out-of-plane angle | Out-of-plane  angle  (i, j, k, l)  The second (j) is the center atom |  |  |
| OHARM | Harmonic function for out-of-plane angle | Out-of-plane  angle  (i, j, k, l)  The second (j) is the center atom |  |  |
| Improper Torsion | IBCOS | Biased (reference angles are fixed) Fourier expansion of cosine of improper dihedral angle with plus sign (up to three terms in one line) | Improper  dihedral  angle (1)  (i, j, k, l)  The third (k) is the center atom |  |  |
| IHARM | Harmonic of improper dihedral angle | Improper  dihedral  angle (2)  (i, j, k, l)  The first (i) is the center atom |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cross-terms | C\_B\_A | Bond-angle coupling; reference values are taken from corresponding bond or angle terms | Bond angle  (i, j, k) |  |  |
| C\_B\_B | Bond-bond coupling, reference values are taken from corresponding bond terms. | Bond angle  (i, j, k) |  |  |
|  | C\_A\_A | Angle-angle coupling | Out-of-plane  angle  (i, j, k, l)  The second (j) is the center atom |  |  |
|  | C\_B\_T | Central bond and torsion coupling | Dihedral angle  (i-j-k-l) |  |  |
|  | C\_A\_T | Angle-torsion coupling | Dihedral angle  (i-j-k-l) |  |  |
|  | C\_BBT | End bonds-torsion coupling | Dihedral angle  (i-j-k-l) |  |  |
|  | C\_AAT | End Angles-torsion coupling | Dihedral angle  (i-j-k-l) |  |  |
|  | C\_B2B | Endbond-endbond | Dihedral angle  (i-j-k-l) |  |  |
|  | UB\_13 | Urey-Bradley 1-3 coupling | Bond angle  (i-j-k) |  |  |

# CHAPTER 7 – File Formats

## 7.1 Molecular Structural Data (MSD) File

MSD files are in free format, columns are not fixed and separated by space. The keywords are prefixed by ‘$’, which will be read by the program so that they cannot be changed. The comments or labels are prefixed by ‘#’, which are not read by the program, and can be edited.

**Section 1 - header**

Several keywords may be given to specify the model.

$ForceField = < the path to ppf assigned to this model >

$TypingRule = < the typing rule used for assign atom type >

For example:

#DFF:MSD 2.0  
#ForceField = E:\DFF8\MyFF.ppf

#TypingRule = Default

**Section 2 - Atoms**

This section starts with keyword: $NumAtom = < the number of atoms > and followed by lines of atomic data. In each line, the order of data is:

index, atomic number, atom type, atomic charge, coordinate x, coordinate y, coordinate z, molecule id, residue id, and group id.

For example,

$NumAtom = 21

1 6 c\_3 0.3165 -4.1082 -0.9024 -0.1570 1 UNK 0

2 6 c\_3a 0.0770 -2.6080 -1.1963 -0.1397 1 UNK 0

3 6 c\_3a 0.1824 -1.4993 -0.1274 -0.1188 1 UNK 0

**Section 3 - Connectivity**

This section starts with keyword $NumBond = < the number of bonds > and followed by data of each bond: atom 1 index, atom 2 index, and bond order. Valid bond orders are 1, 2, 3 and -2 for single, double, triple and partial double bonds.

$NUMBOND = 3  
1    2    1  
1   3   -2  
2    3   -2

**Section 4 (optional) – other information**

$FORMAL\_CHARGE starts this section, with d representing the total

$FORMAL\_CHARGE = 0  
3  
15   -1/2      
16    1      
17   -1/2

$SUBSET = NS starts this section, where NS is the number of subsets. The following NS definition sections each contains the name of the subset and the number of subset members, followed by lines each listing atom indexes in each subset member.

$SUBSET = 2  
SurfactantBox\_subset\_1 3  
$1: 1, 2, 3  
$2: 4, 5, 6  
$3: 9, 10, 12, 13

SurfactantBox\_subset\_0 2  
$1: 8, 20  
$2: 21, 25

## 7.2 Potential Parameter File (PPF) File

PPF files use free formatting, in which columns are not fixed, and data is separated by a space (or a comma). Section 1, the equivalence table, is optional.

**Section 1 - Equivalence Table**

Line 1: file header fixed as *#DFF:EQT*

Line 2: description of terms. There are 11 terms:

AAT - apparent atom type  
NB - nonbond  
ATC - atom type charge  
BINC - bond increment  
A/C - angle center atom  
A/S - angle side atom  
T/C - torsion center atoms  
T/S - torsion side atoms  
O/C - out-of-plane center atom  
O/S - out-of-plane side atoms

Line 3 & up: equivalence table, from the apparent type to actual type used in each terms. For example:

#DFF:EQT  
#AAT : NB ATC BINC Bond A/C A/S T/C T/S O/C O/S  
c\_4n: c\_4n c\_4n c\_4n c\_4n c\_4n c\_4n c\_4n c\_4n c\_4n c\_4n : A  
c\_4o: c\_4o c\_4o c\_4o c\_4o c\_4o c\_4o c\_4o c\_4o c\_4o c\_4o : A  
h\_1: h\_1 h\_1 h\_1 h\_1 h\_1 h\_1 h\_1 h\_1 h\_1 h\_1 : A  
h\_1n: h\_1n h\_1n h\_1n h\_1n h\_1n h\_1n h\_1n h\_1n h\_1n h\_1n : A  
h\_1o: h\_1o h\_1o h\_1o h\_1o h\_1o h\_1o h\_1o h\_1o h\_1o h\_1o : A  
n\_3h1: n\_3h1 n\_3h1 n\_3h1 n\_3h1 n\_3h1 n\_3h1 n\_3h1 n\_3h1 n\_3h1 n\_3h1 : A  
o\_2: o\_2 o\_2 o\_2 o\_2 o\_2 o\_2 o\_2 o\_2 o\_2 o\_2 : A

**Section 2 - Parameter list**

Line 1: file header fixed as **#DFF:PPF**  
Line 2: forcefield type, **#Protocol = CFF | AMBER | CHARMM | TEAM | DREIDING**  
Line 3 & up: parameter terms, generally with four sections separated by colons (:) as follows: **term : atom types : parameters : flags**

For example,

#DFF:PPF  
#PROTOCOL = CFF  
ATYPE: c\_4n: 6.0000\*, 12.0110\*: E 0  
ATYPE: c\_4o: 6.0000\*, 12.0110\*: E 0  
BINC: c\_4n, c\_4o: 0.0351 : E 0  
BINC: c\_4n, h\_1: -0.0007 : E 0  
OHARM: c\_4n, n\_3h1, c\_4n, h\_1n: 0.0000\*, 0.0000\*: E 43  
C\_B\_B: c\_4n, c\_4o, h\_1: 6.9965 : E 51  
C\_B\_A: c\_4n, c\_4o, h\_1: 23.8853 : E 62  
N\_9\_6: c\_4n: 4.0086 , 0.0594 : E 45  
N\_9\_6: c\_4o: 4.0086 , 0.0594 : E 44  
BQUAR: c\_4n, c\_4o: 1.5031 , 330.6238 , -661.2477 , 771.4556 : E 15  
BQUAR: c\_4n, h\_1: 1.0946 , 356.2721 , -712.5442 , 831.3016 : E 18  
AQUAR: c\_4n, c\_4o, h\_1: 110.9436 , 54.6945 , -11.4222 , -11.7180 : E 22  
AQUAR: c\_4n, c\_4o, o\_2: 107.4400 , 104.3870 , -17.8398 , -21.9785 : E 23  
TCS3M: c\_4n, c\_4o, o\_2, h\_1o: -0.9319 , -0.2864 , -0.2246 : E 37  
TCS3M: c\_4n, n\_3h1, c\_4n, c\_4o: 2.7786 , 0.0744 , -0.5756 : E 40

The following is a complete example of a PPF file for a water molecule:

#DFF:EQT  
#AAT : NB ATC BINC Bond A/C A/S T/C T/S O/C O/S  
h\_1w: h\_1w h\_1w h\_1w h\_1w h\_1w h\_1w h\_1w h\_1w h\_1w h\_1w : A, V 1.0  
o\_2w: o\_2w o\_2w o\_2w o\_2w o\_2w o\_2w o\_2w o\_2w o\_2w o\_2w : A, V 1.19  
#DFF:PPF  
#PROTOCOL = AMBER  
BINC: h\_1w, o\_2w: 0.4238\*: E 0, V 1.0, S T\_AMBER, R R1  
N12\_6: h\_1w: 0.0001\*, 0.0001\*: E 0, V 1.0, S T\_AMBER, R R5  
N12\_6: o\_2w: 3.5532\*, 0.1553\*: E 0, V 1.0, S T\_AMBER, R R5  
BHARM: h\_1w, o\_2w: 1.0000\*, 450.0000\*: E 0, V 1.0, S T\_AMBER, R R5  
AHARM: h\_1w, o\_2w, h\_1w: 109.4700\*, 55.0000\*: E 0, V 1.0, S T\_AMBER, R R5

## 7.3 Quantum Mechanics Data (QMD) File

The QMD file contains quantum data. In older versions of DFF, this file also contains topologic information for the molecule by including a section at the header of the file in MSD format. For example:

#DFF:MSD  
#Model Structure Data File Energy = -47947.382101  
3  
1 O0 8 o\_2w o\_2w 0.000000 -1.000000 -0.875000 -0.017310 1 UNK 0  
2 H1 1 h\_1w h\_1w 0.000000 -1.000000 -0.113702 0.582231 1 UNK 0  
3 H2 1 h\_1w h\_1w 0.000000 -1.000000 -1.636298 0.582231 1 UNK 0  
2  
1 2 1   
1 3 1

From release 7.1.4 onward, this topologic data is not included in the QMD file. This is because the topologic data can be changed by DFF functions, while QMD data should be unchanged. In order for DFF to perform calculations using topologic data, a link to a MSD file is included in the QMD file. This link can be added or changed by using the **Associate** command. This is recorded in the first line of a QMD file. For example:

#Associated MSD = keton.msd

By default, the MSD file should be placed in the same folder of the QMD file. When the QMD file is moved, the associated MSD file should be moved with it.

Quantum data information format is freely formatted. Columns are not fixed and data are separated by spaces. Only sections 1, 2, and 8 are required, while others are optional. One .qmd file can contain several structures. Each structure should have the required sections 1, 2, and 8. After the last structure, a section 9 is required.

**Section 1, header**

Part 1: the associated MSD file (can be empty)  
Part 2: file header fixed **! Aeon Quantum Mechanic Data Version 0.1**  
Part 3: mark of **ID** label  
Part 4: mark of **METHOD** label (optional)

#Associated MSD = keton.msd

! Aeon Quantum Mechanic Data Version 0.1  
$ID  
    0  
$END

$METHOD  
    B3LYP/6-31G\* Force Pop=(CHelpG,Regular)  
$END

**Section 2, Basic Data**

!------------------------------------------------------------------------------

! Basic Data

!------------------------------------------------------------------------------

$NUMBER OF ATOMS

10

$END !NUMBER OF ATOMS

$ATOM

!Index Symbol Number X Y Z

1 C 6 -1.295735 -0.118523 0.054397

2 C 6 0.029560 0.613385 0.192253

3 C 6 1.281459 -0.179593 -0.146886

4 O 8 0.085192 1.773221 0.552857

5 H 1 -1.306202 -1.021219 0.677911

6 H 1 -2.114783 0.540259 0.348486

7 H 1 -1.444815 -0.447479 -0.981899

8 H 1 2.163001 0.453922 -0.033472

9 H 1 1.371165 -1.053349 0.510930

10 H 1 1.231159 -0.560623 -1.174577

$END

**Section 3, (optional) Energy Data**

!----------------------------------------------------------  
!                         Energy Data  
!----------------------------------------------------------  
$ENERGY  
    -84812.887268 kcal/mol  
$END !ENERGY  
$FORCES !(kcal/mol/A)  
    -0.716913     -5.987174     -2.555379  
     2.752115     -0.662026      0.665939  
    -1.625517     -2.123094     -1.871374  
     0.045693     -0.226217     -0.191085  
    -0.051677     -0.803779      0.290121  
     0.643164     -1.016603     -0.639069  
    -2.261426      6.214656      1.616661  
    -0.738376      1.817246      2.189279  
    -2.518061      3.433639      1.752931  
     4.470998     -0.646648     -1.258025  
$END !FORCES

**Section 4, (optional) Charge Data**

!----------------------------------------------------------   
!                         Charge Data  
!----------------------------------------------------------   
$TOTAL CHARGE  
    -0.000000 e  
$END !TOTAL CHARGE  
$ESP CHARGES  
    -0.739042  
     0.340667  
     0.158486  
     0.007208  
     0.033324  
    -0.025768  
     0.355330  
    -0.020934  
    -0.049300  
    -0.059971  
$END !ESP CHARGES  
$MULLIKEN CHARGES  
    -0.526655  
     0.294600  
    -0.305418  
     0.146243  
     0.152488  
     0.116706  
    -0.304082  
     0.145300  
     0.153126  
     0.127692  
$END !MULLIKEN CHARGES

**Section 5, (Optional) Electrostatic Properties**

!----------------------------------------------------------   
!          Electrostatic Properties (Atomic Units)  
!----------------------------------------------------------   
$ESP  
!        X           Y            Z        Potential  
     0.000316     0.566741    -0.152380   -18.370141  
     0.000132     1.330423     0.520541    -1.071977  
    -1.212072    -0.222454     0.021839   -14.725686  
    -2.089652     0.427595    -0.060418    -1.127493  
    -1.275350    -0.969811    -0.778655    -1.130773  
    -1.271076    -0.764719     0.985643    -1.136620  
     1.214571    -0.220036     0.019766   -14.726146  
     0.979593    -1.282681    -0.117743    -1.124138  
     1.946472     0.060080    -0.745303    -1.138253  
     1.692673    -0.113130     1.012963    -1.127335  
$END !ESP

**Section 6, (Optional) Poles**

!----------------------------------------------------------   
!                         Poles  
!----------------------------------------------------------  
$DIPLE !(Debye)  
    -0.001000     -0.256600     1.107300 (TOTAL=1.136643)  
$END !DIPLE  
$QUADRUPOLE  
    -19.824400     -20.116900     -21.402600  
     -0.126300       0.107500       1.912300  
$END !QUADRUPOLE

**Section 7, (Optional) Miscellaneous Options**

!----------------------------------------------------------   
!                   Miscellaneous Options  
!----------------------------------------------------------   
$OPTIONS  
B3LYP/6-31G(d)  
$END !OPTIONS  
$DATE  
22-Oct-2007  
$END !DATE  
$PACKAGE  
x86-Linux-G03RevB.03  
$END !PACKAGE

**Section 8, Structure End-mark**

$$$$$$$$$$$$$$$$$$$$$$$$$ -- End Of Record -- $$$$$$$$$$$$$$$$$$$$$$$$$

**Section 9, End-mark**

#DFF:END

## 7.4 Atom Type Definition (DEF, EXT) File

DEF or EXT files contain a number of blocks, and each block defines one atom type. The difference between .def and .ext files is that the former is a complete definition *de novo*, while the latter is an extended definition based on Default definitions. The .ext definition is an optional extension to the Default definition. For default atom types that do not have extended definitions, the default definition is used.

Each definition block follows the general format:

Define <atom type> !comments  
<Substructure Specification >  
[<Atom attribute string>]  
[…]  
End

Line 1, the definition line, starts with **Define**. **<atom type>** is a string of any length representing the atom type symbol. The **comment** is separated from the definition by **!** and should be kept within the same line.

Line 2 is a substructure specification, written as a SMILE string consisting of element symbols, bonds and bond orders. The nested substructure, which can be of unlimited width and depth, is represented by square brackets **[ ]**. The more structures defined, the more specific a match will be. The elements in the substructure specification are:

1. Element Symbol – any element on the periodic table in capital case. **?** is a wildcard indicating any element. \* indicates multiple elements are allowed or not allowed. A list must be given in the atom attribute section using the keyword **element = < >**.
2. Bond orders: - for single, **=** for double, **#** for triple, **~** for partial double, **&** for any.

Line 3 onward lists atom attributes for each atom in line 2. Each line corresponds to one atom and the order must match that in line 2. The attribute string must follow the syntax below:

**Atom id: [coordination=n] [,aromatic = yes|no] [,ringsize=r][,formalcharge=c] [,element = [!]list]**

where **coordination=n** is an integer indicating the coordination number (how many bonds are connected). If an integer is given, all bonds should be specified. It can be a question mark **?** indicating any coordination number is allowed. An integer following the question mark, **?n**, indicates the coordination is unlimited but **n** bonds must be specified.

**ringsize = r** is an integer indicating ring size. **>=** is allowed in addition to **=**.

**c** indicates formal charge using a string such as **+q, -2, 1,** etc.

**element = list** indicates allowed elements, which should be separated by spaces. A **!** placed in front of the list indicates disallowed elements.

Note that the first 4 characters of the above attributes are used in the definition, while the rest are optional.

**End** in the last line closes this block.

For example:

Define n\_samle ! sample for definition

N [-C[=O][-C[-\*][&\*]]][-C]

atom 1: coor=3

atom 2: coor=3

atom 3: coor=?

atom 4: coor=4

atom 5: coor=?, elem=F O

atom 6: coor=?, elem=!P Si

atom 7: coor=?, arom=no, ring=>5, form=1

End

And the first nitrogen is in a non-aromatic 5-member ring, the second atom (carbon) has three bonds, also in a 5-member ring; the third (nitrogen) and last (carbon) atoms are in 5-member ring.

# REFERENCES

(1) GROMACS available under the GNU Lesser General Public License (LGPL), version 2.1. <http://www.gromacs.org/>.

(2) LAMMPS distributed as an open source code under the terms of the GPL, distributed by Sandia National Laboratories, a US Department of Energy laboratory. <http://lammps.sandia.gov>.

(3) Asensio, J. L.; Jimenez‐Barbero, J. The use of the AMBER forcefield in conformational analysis of carbohydrate molecules: Determination of the solution conformation of methyl α‐lactoside by NMR spectroscopy, assisted by molecular mechanics and dynamics calculations. *Biopolymers* **1995**, *35*, 55-73.

(4) Wang, J. M.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and testing of a general amber forcefield. *Journal of Computational Chemistry* **2004**, *25*, 1157-1174.

(5) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. CHARMM: The biomolecular simulation program. *Journal of Computational Chemistry* **2009**, *30*, 1545-1614.

(6) Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry* **1983**, *4*, 187-217.

(7) Guvench, O.; Mallajosyula, S. S.; Raman, E. P.; Hatcher, E.; Vanommeslaeghe, K.; Foster, T. J.; Jamison, F. W.; MacKerell Jr, A. D. CHARMM additive all-atom forcefield for carbohydrate derivatives and its utility in polysaccharide and carbohydrate–protein modeling. *Journal of chemical theory and computation* **2011**, *7*, 3162-3180.

(8) Huang, J.; Rauscher, S.; Nawrocki, G.; Ran, T.; Feig, M.; de Groot, B. L.; Grubmüller, H.; MacKerell Jr, A. D. CHARMM36m: an improved forcefield for folded and intrinsically disordered proteins. *Nature methods* **2017**, *14*, 71.

(9) Jorgensen, W. L.; Tiradorives, J. THE OPLS POTENTIAL FUNCTIONS FOR PROTEINS - ENERGY MINIMIZATIONS FOR CRYSTALS OF CYCLIC-PEPTIDES AND CRAMBIN. *Journal of the American Chemical Society* **1988**, *110*, 1657-1666.

(10) Jorgensen, W. L.; Tirado-Rives, J. The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *Journal of the American Chemical Society* **1988**, *110*, 1657-1666.

(11) Damm, W.; Frontera, A.; Tirado–Rives, J.; Jorgensen, W. L. OPLS all-atom forcefield for carbohydrates. *Journal of Computational Chemistry* **1997**, *18*, 1955-1970.

(12) Kony, D.; Damm, W.; Stoll, S.; Van Gunsteren, W. F. An improved OPLS–AA forcefield for carbohydrates. *Journal of Computational Chemistry* **2002**, *23*, 1416-1429.

(13) MacCallum, J. L.; Tieleman, D. P. Calculation of the water–cyclohexane transfer free energies of neutral amino acid side-chain analogs using the OPLS all-atom forcefield. *Journal of Computational Chemistry* **2003**, *24*, 1930-1935.

(14) Zubillaga, R. A.; Labastida, A.; Cruz, B.; Martínez, J. C.; Sánchez, E.; Alejandre, J. Surface Tension of Organic Liquids Using the OPLS/AA Forcefield. *Journal of Chemical Theory and Computation* **2013**, *9*, 1611-1615.

(15) Bernardes, C. E. S.; Joseph, A. Evaluation of the OPLS-AA Forcefield for the Study of Structural and Energetic Aspects of Molecular Organic Crystals. *The Journal of Physical Chemistry A* **2015**, *119*, 3023-3034.

(16) Harder, E.; Damm, W.; Maple, J.; Wu, C.; Reboul, M.; Xiang, J. Y.; Wang, L.; Lupyan, D.; Dahlgren, M. K.; Knight, J. L. OPLS3: a forcefield providing broad coverage of drug-like small molecules and proteins. *Journal of chemical theory and computation* **2015**, *12*, 281-296.

(17) Hwang, M. J.; Stockfisch, T. P.; Hagler, A. T. Derivation of Class II Forcefields. 2. Derivation and Characterization of a Class II Forcefield, CFF93, for the Alkyl Functional Group and Alkane Molecules. *Journal of the American Chemical Society* **1994**, *116*, 2515-2525.

(18) Maple, J. R.; Hwang, M. J.; Stockfisch, T. P.; Dinur, U.; Waldman, M.; Ewig, C. S.; Hagler, A. T. Derivation of class II forcefields. I. Methodology and quantum forcefield for the alkyl functional group and alkane molecules. *Journal of Computational Chemistry* **1994**, *15*, 162-182.

(19) Sun, H.; Mumby, S. J.; Maple, J. R.; Hagler, A. T. An ab Initio CFF93 All-Atom Forcefield for Polycarbonates. *Journal of the American Chemical Society* **1994**, *116*, 2978-2987.

(20) Sun, H.; Mumby, S. J.; Maple, J. R.; Hagler, A. T. Ab Initio Calculations on Small Molecule Analogs of Polycarbonates. *The Journal of Physical Chemistry* **1995**, *99*, 5873-5882.

(21) Hwang, M. J.; Ni, X.; Waldman, M.; Ewig, C.; Hagler, A. Derivation of class II forcefields. VI. Carbohydrate compounds and anomeric effects. *Biopolymers* **1998**, *45*, 435-468.

(22) Ewig, C. S.; Thacher, T. S.; Hagler, A. T. Derivation of Class II Forcefields. 7. Nonbonded Forcefield Parameters for Organic Compounds. *The Journal of Physical Chemistry B* **1999**, *103*, 6998-7014.

(23) Sun, H. COMPASS: An ab Initio Force-Field Optimized for Condensed-Phase ApplicationsOverview with Details on Alkane and Benzene Compounds. *The Journal of Physical Chemistry B* **1998**, *102*, 7338-7364.

(24) Sun, H.; Ren, P.; Fried, J. The COMPASS forcefield: parameterization and validation for phosphazenes. *Computational and Theoretical Polymer Science* **1998**, *8*, 229-246.

(25) Bunte, S. W.; Sun, H. Molecular modeling of energetic materials: the parameterization and validation of nitrate esters in the COMPASS forcefield. *The Journal of Physical Chemistry B* **2000**, *104*, 2477-2489.

(26) Bunte, S. W.; Sun, H. Molecular Modeling of Energetic Materials:  The Parameterization and Validation of Nitrate Esters in the COMPASS Forcefield. *The Journal of Physical Chemistry B* **2000**, *104*, 2477-2489.

(27) Yang, J.; Ren, Y.; Tian, A.-m.; Sun, H. COMPASS Forcefield for 14 Inorganic Molecules, He, Ne, Ar, Kr, Xe, H2, O2, N2, NO, CO, CO2, NO2, CS2, and SO2, in Liquid Phases. *The Journal of Physical Chemistry B* **2000**, *104*, 4951-4957.

(28) McQuaid, M. J.; Sun, H.; Rigby, D. Development and validation of COMPASS forcefield parameters for molecules with aliphatic azide chains. *Journal of computational chemistry* **2003**, *25*, 61-71.

(29) McQuaid, M. J.; Sun, H.; Rigby, D. Development and validation of COMPASS forcefield parameters for molecules with aliphatic azide chains. *Journal of Computational Chemistry* **2004**, *25*, 61-71.

(30) Sun, H.; Jin, Z.; Yang, C.; Akkermans, R. L. C.; Robertson, S. H.; Spenley, N. A.; Miller, S.; Todd, S. M. COMPASS II: extended coverage for polymer and drug-like molecule databases. *Journal of Molecular Modeling* **2016**, *22*, 1-10.

(31) Mayo, S. L.; Olafson, B. D.; Goddard, W. A. DREIDING: a generic forcefield for molecular simulations. *The Journal of Physical Chemistry* **1990**, *94*, 8897-8909.

(32) Jin, Z.; Yang, C.; Cao, F.; Li, F.; Jing, Z.; Chen, L.; Shen, Z.; Xin, L.; Tong, S.; Sun, H. Hierarchical atom type definitions and extensible all-atom forcefields. *Journal of Computational Chemistry* **2016**, *37*, 653-664.

(33) Cao, F.; Gong, Z.; Wu, Y.; Sun, H. A high-throughput computing procedure for predicting vapor-liquid equilibria of binary mixtures – Using carbon dioxide and n-alkanes as examples. *Fluid Phase Equilibria* **2017**, *452*, 58-68.

(34) Huang, H.; Cao, F.; Wu, L.; Sun, H. All-atom and coarse-grained forcefields for polydimethylsiloxane. *Molecular Simulation* **2017**, 1-10.

(35) Gong, Z.; Sun, H.; Eichinger, B. E. On the Temperature Transferability of Forcefield Parameters for Dispersion Interactions. *Journal of Chemical Theory and Computation* **2018**.

(36) Gong, Z.; Wu, Y.; Wu, L.; Sun, H. Predicting Thermodynamic Properties of Alkanes by High-Throughput Forcefield Simulation and Machine Learning. *Journal of Chemical Information and Modeling* **2018**, *58*, 2502-2516.

(37) Gong, Z.; Sun, H. Pressure-viscosity relation of 2,2,4-trimethylhexane predicted using all-atom TEAM forcefield. *Fluid Phase Equilibria* **2019**, *497*, 64-70.

(38) Gong, Z.; Sun, H. Extension of TEAM Force-Field Database to Ionic Liquids. *Journal of Chemical & Engineering Data* **2019**.

(39) Lide, D. R.; Baysinger, G.; Berger, L. I.; Goldberg, R. N.; Kehiaian, H. V.; Kuchitsu, K.; Rosenblatt, G.; Roth, D. L.; Zwillinger, D. CRC Handbook of Chemistry and Physics.

(40) Figueras, J. Ring Perception Using Breadth-First Search. *Journal of Chemical Information and Computer Sciences* **1996**, *36*, 986-991.

(41) Allinger, N. L. Conformational analysis. 130. MM2. A hydrocarbon forcefield utilizing V1 and V2 torsional terms. *Journal of the American Chemical Society* **1977**, *99*, 8127-8134.

(42) Allinger, N. L.; Yuh, Y. H.; Lii, J. H. Molecular mechanics. The MM3 forcefield for hydrocarbons. 1. *Journal of the American Chemical Society* **1989**, *111*, 8551-8566.

(43) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A Second Generation Forcefield for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *Journal of the American Chemical Society* **1995**, *117*, 5179-5197.

(44) MacKerell, A. D.; Banavali, N.; Foloppe, N. Development and current status of the CHARMM forcefield for nucleic acids. *Biopolymers* **2000**, *56*, 257-265.

(45) Dasgupta, S.; Yamasaki, T.; III, W. A. G. The Hessian biased singular value decomposition method for optimization and analysis of forcefields. *The Journal of Chemical Physics* **1996**, *104*, 2898-2920.

(46) Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M. UFF, a full periodic table forcefield for molecular mechanics and molecular dynamics simulations. *Journal of the American Chemical Society* **1992**, *114*, 10024-10035.

(47) Rappe, A. K.; Goddard III, W. A. Charge equilibration for molecular dynamics simulations. *The Journal of Physical Chemistry* **1991**, *95*, 3358-3363.

(48) Riley, K. E.; Op't Holt, B. T.; Merz, K. M. Critical Assessment of the Performance of Density Functional Methods for Several Atomic and Molecular Properties. *Journal of Chemical Theory and Computation* **2007**, *3*, 407-433.

(49) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical recipes 3rd edition: The art of scientific computing*; Cambridge University Press, 2007.

(50) Sun, H. Prediction of fluid densities using automatically derived VDW parameters. *Fluid Phase Equilibria* **2004**, *217*, 59-76.

(51) Wang, L.-P.; Chen, J.; Van Voorhis, T. Systematic Parametrization of Polarizable Forcefields from Quantum Chemistry Data. *Journal of Chemical Theory and Computation* **2013**, *9*, 452-460.