AFMM: A Molecular Mechanics Force Field Parametrization Program

A.C. Vaiana[*], Z.Cournia, I.B. Costescu, J.C. Smith

Computational Molecular Biophysics, Interdisciplinary Center for Scientific Computing (IWR), Im Neuenheimer Feld 368, Universität Heidelberg, 69120 Heidelberg, Germany

[*]Present Address:Institut de Biologie Molculaire et Cellulaire CNRS (UPR 9002 - SMBMR) 15, rue Ren Descartes - 67084 STRASBOURG Cedex

Contents

1 Introduction

AFMM (Automated Frequency Matching Method) is a program package for CHARMM [1] force field parametrization. The method includes fitting the molecular mechanics potential to both vibrational frequencies and eigenvector projections derived from quantum chemical calculations and is benchmarked on a series of already parametrized molecules [2].
The program is written in Python and is provided as one Python source file (afmm.py). Through a simple configuration file, the program is directed to import and then match normal modes from a quantum mechanical (QM) program output file and a CHARMM output file. The program optimizes the missing CHARMM parameters by iteratively changing them in a random fashion in order to get the best fit with the reference set (QM data). By implementing a Monte Carlo-like algorithm, the tedious task of manual parametrization is replaced by an efficient automated procedure.
The method is best suited for optimization of small rigid molecules with a well-defined energy minimum, for which the harmonic approximation to the energy surface is appropriate to describe their intra-molecular degrees of freedom. It is also of particular use in deriving parameters for molecules for which experimental data are scarce.
Final testing of a parameter set should be performed against experimental or theoretical data that are as far as possible independent of the data used in the optimization phase. For example, molecular dynamics of the crystal structure (if present) using the set of optimized parameters.

2 Theoretical Background

Molecular Dynamics (MD) aims to reproduce the time-dependent motional behavior, structures and energies of molecular systems by integrating Newton's equations of motion. The potential energy of the system is described as a function of the atomic positions. This function or "force field" describes how the energy changes when the system moves from one conformation to another for example when bonds are rotated or atoms are displaced. The functional form of the force field generally includes a set of empirical parameters which are system dependent and must be tuned prior to performing simulations on a new system or molecule. This tuning step is generally referred to as parametrization of the force field. The reliability of a molecular mechanics calculation is dependent on both the functional form of the force field and on the numerical values of the parameters implemented in the force field itself. Thus, the first necessary step for reliable MD simulations is the parametrization procedure.

AFMM is mainly intended for optimization of the force constants of the CHARMM force field although optimization of all other CHARMM parameters is, in principle, possible. In CHARMM the empirical potential energy function is given by Eq.1:

$$
\begin{aligned}
V(\mathbf{r}^N) \quad = \quad & \sum_{bonds} K_b(b-b_0)^2 + \sum_{ub} K_{ub}(s-s_0)^2 + \sum_{angles} K_\theta(\theta-\theta_0)^2 + \\
& \sum_{dihedrals} K_\chi(1+cos(n\chi-\chi_0)) + \sum_{impropers} K\psi(\psi-\psi_0)^2 + \\
& \sum_{nonbond} \epsilon_{ij}\left[\left(\frac{R_{ij}^{min}}{r_{ij}}\right)^{12} + \left(\frac{R_{ij}^{min}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{D r_{ij}} \qquad (1)
\end{aligned}
$$

where K_b, K_{ub}, K_θ, K_χ, K_ϕ are, respectively, the bond, Urey-Bradley, angle, dihedral and improper dihedral constants. b, s, θ, χ, and ϕ represent, respectively, bond length, Urey-Bradley 1-3 distance, bond angle, dihedral angle and improper torsion angle (the subscript zero is used to represent the corresponding equilibrium value). Nonbonded interactions between pairs of atoms (labeled i and j) at a relative distance r_{ij} are described by the Lennard-Jones 6-12 (LJ) and Coulomb interaction terms. R_{ij}^{min} and ϵ_{ij} are, respectively, the distance between atoms i and j at which the LJ potential is minimum and the depth of the LJ potential well for the same pair of atoms. D is the effective dielectric constant and q_i the partial atomic charge on atom i.

General procedures for automated parametrization of molecular mechanics (MM) force fields based on fitting to any given set of reference data involve the following main steps[3, 4]:

- Definition of a merit function based on the available reference data
- Choice of atom types to be used and definition of new atom types when needed
- Careful choice of initial parameter values
- Refinement of Parameters (optimization of the merit function)
- Testing and validation of the final parameter set

2.1 Computational Methods

The determination of the actual values of the various force constants in Eq.1 is a demanding job. One major difficulty in the development of molecular force fields is that these parameters cannot be directly determined from experiments. Nonetheless, they are more directly related to quantities that are well defined quantum mechanically. Experimental data pertaining to force field calculations, such as normal modes, infrared frequencies or crystal lattice constants cannot be expressed as simple functions of the force field parameters. Furthermore, often availability of sufficient experimental data for parametrization is rather scarce.

On the other hand, the second derivatives of the energy with respect to coordinates (i.e. the Hessian matrix elements) are much more directly connected to the force constants of the force field. The point charges of the system can also be readily computed. These quantities are therefore available through *ab initio* calculations, which in this context are invaluable.

AFMM provides an efficient automated way to generate intra-molecular force field parameters using normal modes. The method can be, in principle, used with any atom-based molecular mechanics program which has the facility of calculating normal modes and the corresponding eigenvectors.

The basic principle behind the AFMM method is that it is iteratively tuning an initial MM parameter set in order to reproduce the normal modes generated from a QM program. The reference quantum mechanical normal modes can be calculated with various QM programs (e.g. Gaussian 94/98[5, 6], NWChem[7], ADF[8]) and using various levels of theory (e.g. Hartree-Fock, DFT). The choice of reference data upon which to base the new parametrization is a critical step in the parametrization procedure. The reliability and accuracy of the new parameters in reproducing various properties of the molecule depend on the quality of the reference data.

Equilibrium values for bonds b_0, angles, θ_0 and dihedrals χ_0 can be derived from the quantum chemical ground state structure or from experimental X-ray or NMR structures. A set of partial atomic charges can be computed from these packages using various methods as well. Use of AFMM for optimizing van der Waals parameters is not reccomended. The van der Waals constants ϵ_{ij} and R_{ij}^{min} depend mostly on atomic properties and are relatively insensitive to changes in the molecular environment. Therefore, they can be copied from existing CHARMM values and should not be modified during refinement.

2.2 Description of the Method

Automated refinement methods are mostly based on optimizing a "merit function", which usually corresponds to minimizing a weighted sum of square deviations from a set of reference values. Refinement of parameter sets always involves exploring a high dimensional space in search of an optimal set. As for any multidimensional search method, in parameter optimization there is always a substantial risk that the search will get stuck in a high local minimum. Initializing the procedure from a good initial guess can help reducing this risk.

2.2.1 Choice of atom types and definition of new atom types

A major requirement in MM force fields is the portability of the parameter set, that is, the possibility to transfer large groups of parameters from one molecule to another. In this respect, addition of new atom types to the force field when designing the new parameter set should be limited only to specific cases in which existing types cannot be used.

2.2.2 Choice of initial parameter values

The initial guess has to be based on analogy to other existing CHARMM parameters and on chemical intuition. Equilibrium values and hybridization of the atoms involved should be carefully taken into account when designing a set of initial parameters. The second term in Eq.1 is the Urey-Bradley term which is not present in most force fields and within the CHARMM model its use is limited to a few cases. The initial parameter set is then used for energy minimization and calculation of normal modes (frequencies and eigenvectors) with CHARMM.

These can then be directly compared with the results of the "reference" normal modes calculated by means of a quantum chemical program. Parameters are thus refined iteratively to fit the results of the quantum chemical normal mode calculation.

Another way to ensure the good choice of the initial parameter set involves checking it by visual inspection of the motions involved in exchanged eigenvector modes, using the Molden program[9] for example, and manually adjusting the parameters concerned. This approach is particularly useful for critical torsion parameters. In some cases it is necessary to derive initial parameters from rotational potential energy profiles (single point energy calculations from QM programs) before achieving good optimization.

2.2.3 The Merit Function

One of the major problems of parametrization methods that fit to vibrational frequencies is identifying a calculated mode with the corresponding reference mode. It is possible to obtain good matching of the frequencies for a molecule while exchanging the corresponding eigenvectors. The resulting model would then reproduce well the vibrational frequencies (and the energy) of the reference molecule. However, it may not reproduce the distribution of energy among the inter-molecular modes, and thus the dynamical properties of the molecule. It is therefore important to use a merit function that takes into account both the frequencies and all the corresponding eigenvectors to avoid this problem.

The fitting method proposed here minimizes the above frequency exchange effect. In the "ideal" case of a perfect molecular mechanics model, the vibrational properties of the molecule, as calculated by molecular mechanics, should perfectly match those resulting from the quantum ab initio calculation. For this to occur not only must the frequencies coincide but also the two sets of eigenvectors (resulting from the two different calculations) should coincide. Each eigenvector from the set calculated by molecular mechanics would therefore be orthonormal to all but one (it s corresponding eigenvector) of the vectors from the reference set (calculated using quantum chemical methods).

An efficient way to check simultaneously for both orthonormality and frequency matching is to project each of the CHARMM eigenvectors $\left\{\overline{\chi_i^C}\right\}$ (where the subscript i indicates the normal mode number and the superscript C indicates that the modes are calculated with CHARMM) onto the reference set of eigenvectors $\left\{\overline{\chi_i^Q}\right\}$ (the superscript Q indicates that these modes are calculated with a QM program, to find the frequency ν_j^{max} corresponding to the highest projection ($j : \overline{\chi_i^C} \cdot \overline{\chi_i^Q}$=max) and to plot this frequency against the corresponding frequency, ν_i. In the ideal case mentioned above, this plot would give a one-to-one relationship: $\nu_i = \nu_j^{max}$ and $\overline{\chi_i^C} \cdot \overline{\chi_i^Q} = \delta_{ij}$ where δ_{ij} is the Kroenecker delta. Points that deviate from the ideal plot may indicate exchanged eigenvectors or mismatched frequencies.

AFMM is based on minimizing the weighted sum-of-squares, Y^2 of the deviations from the ideal situation as follows:

$$Y^2 = \sum_{3N-6} \Omega_i^2 (\nu_i - \nu_j^{max})^2 \tag{2}$$

$$\Omega_i^{(1)} = \frac{1}{max_j(\overline{\chi_i^C} \cdot \overline{\chi_i^Q})} \tag{3}$$

$$\Omega_i^{(2)} = \frac{1}{\omega_i^C} \tag{4}$$

where N is the number of atoms in the molecule and there are 3N-6 independent vibrational frequencies. In AFMM there are three possibilities to weigh the merit function:

1. The weights Ω_i are chosen to be the inverse of the highest eigenvector projection. This has the effect of biasing the merit function, even in the case of a good frequency assignment, such that minimization of Y^2 leads to an improved eigenvector projection distribution (Eq.3).

2. The weights Ω_i are chosen to be the inverse of the MD frequency. This has the effect of biasing the merit function towards better fitting of the lowest frequencies, which are biologically more relevant (Eq.4).

3. No weights.

2.2.4 Parameter Refinement

For the automatic optimization of the chosen subset of parameters a standard Monte Carlo (MC) scheme is used to minimize Y^2. Although the subset of parameters to be optimized can be chosen at wish by the user, it is advisable to perform optimizations separately on bond, angle, and torsion constants. At each step i, all chosen parameters are iteratively varied in the MC algorithm with a uniform distribution within a fixed range, Y_i^2 is evaluated, and, if $Y_i^2 < Y_{i-1}^2$, the new parameter set is used in the next step, i+1. The optimisation algorithm is illustrated in Fig.1

When comparing results for different molecules, normalization of Y^2 can be rather tedious due to the different weights Ω_i. For comparison purposes, then, after minimization of Y^2 the root-mean-square deviation, σ from the reference case is calculated:

$$\sigma = \sqrt{\frac{\sum_{3N-6}(\nu_i - \nu_j^{max})^2}{3N-6}} \tag{5}$$

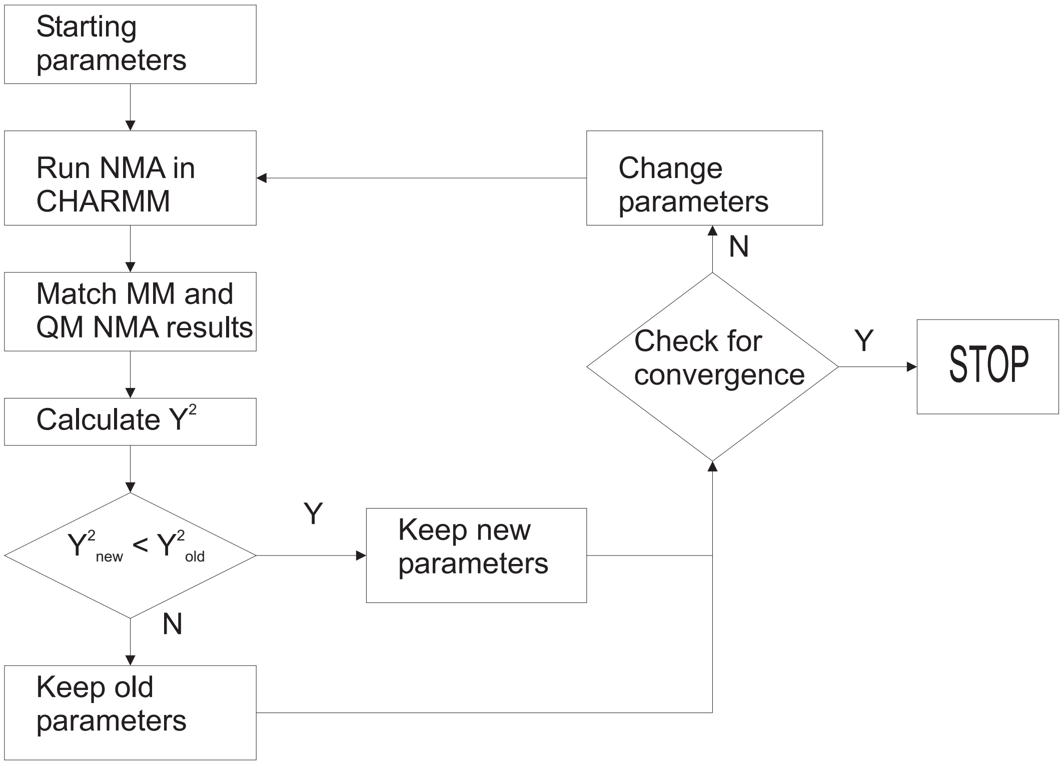

Figure 1: Schematic representation of the optimization algorithm used in the AFMM method. The method iteratively changes the parameters and matches both frequencies and eigenvector projections from the molecular mechanics (CHARMM in this case) normal mode analysis (NMA) with reference QM NMA.

3 Description of the Program

The current version of the program requires Python version 2 or newer and does not require any non-standard Python modules. It contains definitions of two classes and the normal modes import functions.
The `param` class contains information about the parameter to be optimized and a Monte Carlo-like method to generate a new random value that is different from both the starting and the current value.
The names of the normal modes import functions are composed of `read_` followed by the program name. For the QM output files, there are also functions that identify the type of file, their name being composed of `is_` followed by the program name. All the normal modes import functions return lists of non-zero frequencies and corresponding eigenvectors.
The `afmm` class is the core of the program and contains the following methods:

- `ReadConfig` - read the configuration file
- `WriteNewParams` - write new parameters in the CHARMM stream file
- `WriteStreamedInput` - write a new CHARMM input file

- `RunMD` - run MD program, checking for normal termination
- `DotProduct` - calculates the dot product between the eigenvectors
- `TooLow` - checks if a value is too close to zero (before division)
- `Compute` - matches the modes and computes the merit function
- `Optimize` - main routine that assigns new values to parameters and minimizes the merit function
- `OutputResults` - writes out the minimum weighted sigma, the corresponding non-weighted sigma, the optimized parameters and the frequency matching file

The main program consists of only 3 calls: reading the configuration file, calling `Optimize` for computation then calling `OutputResults`.

4 Needed files

AFMM needs the following files to start the parametrization:

4.1 CHARMM Input File

A CHARMM input file whose name has to have an extension ".inp" is required. The CHARMM input file should be organized as follows:

- Read in the topology/parameter/structure files. When defining files you have to use the absolute paths or run AFMM in the directory where these files are to be found. In the parameter file you have to replace the parameter you want to optimize (force constant) with a variable e.g. "@p1" for the first, "@p2" for the second and so on.
- Minimize the structure e.g. `mini abnr nstep 50000 tolgrd 0.00001`
- Run the normal modes calculation using the "vibran" module. You shouldn't output the mass-weighted eigevectors which is the default in CHARMM, so you have to use the `nomass` option within the vibran module. You can find an example input file within the distribution (opt.inp). The output CHARMM file will be automatically created with extension .out (opt.out).

4.2 QM Output File

Currently 3 types of output files are supported for optimization in AFMM: NWChem 4.5 and older, Gaussian 94/98 and Molden format. In principle, any normal mode output can be transformed in the Molden format which contains the frequencies, coordinates and eigenvectors. A number of people have created scripts or programs that output Molden format files starting from other programs output files.
See: http://www.cmbi.kun.nl/∼schaft/molden/others.html.

Important: In order for the parametrization algorithm to be able to compare the same normal modes and eigenvectors two things should be taken care of:

- Firstly, the order of the atoms in the coordinate set used in CHARMM and QM programs respectively should be the same. We suggest that a minimization of the molecule (using initial guesses for the parameters) should be run first with CHARMM and the output coordinates to be given to the QM program for further optimization and normal mode analysis.
- Second, the orientation of the two molecules should be the same when comparing the normal modes. We suggest that when the QM calculation is finished, the optimized coordinates should be converted in a CHARMM .pdb or .crd file and give it as input to CHARMM before the parametrization starts.

5 Structure of the AFMM configuration file

The AFMM configuration file has to be called "afmm.cfg" and to be in the current directory. An example configuration file is distributed with the program.

The configuration file is composed of two sections. The first section is called `[parameters]` and contains information on the parameters that you want to optimize. The general syntax is:

```
parameter_name = min_value max_value start_value
```

The AFMM parameter names (`parameter_name`) should correspond to the variable names (e.g. p1) that you defined in the CHARMM parameter file. The first number is the minimum value that the parameter can take during the optimization. The second number is the maximum value that the parameter can take during the optimization. The third number is the starting value (the initial guess). The numbers can be specified as integers or reals. The items can be separated by any number of whitespaces or tabs. In the following example:

```
[parameters]
p1 = 200.0 1000.0 500
```

the parameter `p1` will be optimized in the range of 200.0-1000.0 with a starting value of 500.0.

The second section is called `[general]` and contains settings that control the flow of the program. The settings that are recognized by the current version are listed below; the values given are just an example:

- `maxsteps=10000`
  The maximum number of optimization steps for each parameter.
- `maxsigmasteps = 2000`
  The maximum number of steps after which, if sigma remains constant, you want the optimization to stop (Convergence Criterion).

- `mdexec = /usr/local/charmm/c28b1/cmf`
  Path of the CHARMM executable.

- `mdinp = /home/cournia/propene/propene.inp`
  Path of the CHARMM input file you want to run.

- `qmout = /home/cournia/propene/propene_freq.out`
  Path of the output QM file that contains the frequencies and the eigenvectors.

- `qmfactor = 0.89`
  Empirical scaling factor for the QM frequencies[10].

- `afmmfile = freq.dat`
  AFMM frequency matching output file to be written at the end of the optimization.

- `weighting = frequency`
  Choose between `frequency`, `projection` or `none` for the way you want your parametrization to be weighted (see section 2).

6 Running the program

The program can be run as follows:

```
python afmm.py
```

If any errors occur in the configuration file, they will be printed. Errors in the `[parameters]` section lead to ignoring the parameter for which they occured. Errors in the `[general]` section lead to the termination of the program.
While running, the program will print on standard output the values of weighted sigma that result during the optimization. When the program exits after the convergence criterion is fulfilled or maximum number of optimization steps is reached, the minimum weighted sigma, the corresponding non-weighted sigma and the final parameter set is printed on standard output and the frequency matching file is created. The frequency matching file will contain 2 columns, the first containing the scaled QM frequencies (if a scaling factor was given to the program) and the second containing corresponding MM frequency values.
For comparison between different molecules or optimizations with different weights, the non-weighted sigma should be used.
Based on previous benchmark studies on test molecules, a good optimization is reached when the value of σ is within 0-100cm^{-1}. This is normally achieved with a `maxsteps` value of 100000 optimization cycles and a convergence criterion (`maxsigmasteps`) of 10000 cycles for each parameter.

6.1 Citing the Program

Please use the following citations when publishing results obtained with AFMM:

- A.C. Vaiana, A.Schulz, J. Wolfrum, M. Sauer and J.C. Smith, "*Molecular Mechanics Force Field Parameterization of the Fluorescent Probe Rhodamine 6G Using Automated Frequency Matching*", J Comput Chem 24: 632-639, 2003
- A.C. Vaiana, Z. Cournia, I.B. Costescu and J.C. Smith, "*AFMM: A Molecular MEchanics Force Field Vibrational Parametrization Program"*, Computer Physics Communications, accepted for publication

For additional applications see:

- Z. Cournia, A.C. Vaiana, G.M. Ullmann and J.C. Smith, "*Derivation of a Molecular Mechanics Force Field for Cholesterol*", Pure and Applied Chemistry, 76(1):189-196, 2004

References

[1] B. R. Brooks, Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: A Program for Macromolecular Energy, Minimization and Dynamics Calculations. *J. Comp. Biol.*, 4:187, 1983.

[2] A.C. Vaiana, A. Schulz, J. Worfrum, M.Sauer, and J.C. Smith. Molecular mechanics force field parametrization of the fluorescent probe rhodamine 6G using automated frequency matching. *J. Comp. Chem.*, 24:632, 2002.

[3] P.O. Norrby and T.Liljefors. Automated molecular mechanics parametrization with simultaneous utilization of experimental and quantum mechanical data. *J. Comp. Chem.*, 10:1146, 1998.

[4] A.J. Hopfinger and R.A. Pearlstein. Molecular Mechanics Force-Field Parameterization Procedures. *J. Comp. Chem.*, 5:486, 1984.

[5] M. J. Frisch, G. W. Trucks, H. B. Schlegel, P. M. W. Gill, B. G. Johnson, M. A. Robb, J. R. Cheeseman, T. Keith, G. A. Petersson, J. A. Montgomery, K. Raghavachari, M. A. Al-Laham, V. G. Zakrzewski, J. V. Ortiz, J. B. Foresman, J. Cioslowski, B. B. Stefanov, A. Nanayakkara, M. Challacombe, C. Y. Peng, P. Y. Ayala, W. Chen, M. W. Wong, J. L. Andres, E. S. Replogle, R. Gomperts, R. L. Martin, D. J. Fox, J. S. Binkley, D. J. Defrees, J. Baker, J. P. Stewart, M. Head-Gordon, C. Gonzalez, and J. A. Pople. Gaussian 94, Revision E.1. *Gaussian, Inc., Pittsburgh PA*, 1995.

[6] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, Jr. J. A. Montgomery, R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui, K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J. V. Ortiz, A. G. Baboul, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, J. L. Andres, C. Gonzalez, M. Head-Gordon, E. S. Replogle, and J. A. Pople. Gaussian 98, Revision A.9. *Gaussian, Inc., Pittsburgh PA*, 1998.

[7] R. J. Harrison, J. A. Nichols, T. P. Straatsma, M. Dupuis, E. J. Bylaska, G. I. Fann, T. L. Windus, E. Apra, J. Anchell, D. Bernholdt, P. Borowski, T. Clark, D. Clerc, H. Dachsel, B. de Jong, M. Deegan, K. Dyall, D. Elwood, H. Fruchtl, E. Glendenning, M. Gutowski, A. Hess, J. Jaffe, B. Johnson, J. Ju, R. Kendall, R. Kobayashi, R. Kutteh, Z. lin, R. Littlefield, X. Long, B. Meng, J. Nieplocha, S. Niu, M. Rosing, G. Sandrone, M. Stave, H. Taylor, G. Thomas, J. van Lenthe, K. Wolinski, A. Wong, and Z. Zhang. NWChem, A Computational Chemistry Package for Parallel Computers, Version 4.0.1. *Pacific Northwest National Laboratory, Richland, Washington, USA*, 2001.

[8] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, and T. Ziegler. Chemistry with ADF. *J. Comp. Chem.*, 22:931, 2001.

[9] G.Schaftenaar and J.H. Noordik. Molden: a pre- and post-processing program for molecular and electronic structures. *J. Comput.-Aided Mol. Design*, 14:123–134, 2000.

[10] J.B. Foresman and AE. Frisch. Exploring Chemistry with Electronic Structure Methods. *Gaussian, Inc., Pittsburgh PA*, page 2nd ed., 1993.