Effective Fragment Potential Method in Q-CHEM: A Guide for Users and Developers

Debashree Ghosh,[a,b] Dmytro Kosenkov,[c,d] Vitalii Vanovschi,[a] Joanna Flick,[c] Ilya Kaliman,[a,c] Yihan Shao,[e] Andrew T. B. Gilbert,[f] Anna I. Krylov,*[a] and Lyudmila V. Slipchenko*[c]

A detailed description of the implementation of the effective fragment potential (EFP) method in the Q-CHEM electronic structure package is presented. The Q-CHEM implementation interfaces EFP with standard quantum mechanical (QM) methods such as Hartree–Fock, density functional theory, perturbation theory, and coupled-cluster methods, as well as with methods for electronically excited and open-shell species, for example, configuration interaction, time-dependent density functional theory, and equation-of-motion coupled-cluster models. In addition to the QM/EFP functionality, a "fragment-only" feature is also available (when the system is described by effective fragments only). To aid further developments of the EFP methodology, a detailed description of the C++ classes and EFP module's workflow is presented. The EFP input structure and EFP job options are described. To assist setting up and performing EFP calculations, a collection of Perl service scripts is provided. The precomputed EFP parameters for standard fragments such as common solvents are stored in Q-CHEM's auxiliary library; they can be easily invoked, similar to specifying standard basis sets. The instructions for generating user-defined EFP parameters are given. Fragments positions can be specified by their center of mass coordinates and Euler angles. The interface with the IQMOL and WEBMO software is also described. © 2013 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23223

Introduction

The effective fragment potential (EFP) method is a systematic approach to describing intermolecular interactions in an efficient yet rigorous way.†[1–3] EFP is a quantum mechanical (QM) based potential; it can be described as a nonempirical polarizable force field. The parameters used to compute interactions between the effective fragments with each other and with the QM part are obtained from auxiliary *ab initio* calculations, without any fitting and/or empirical parameters. Thus, EFP elevates a popular QM/MM (quantum-mechanics/molecular mechanics) scheme to the class of predictive *ab initio* methods. The EFP Hamiltonian is pairwise, however, leading many-body effects are included through a self-consistent treatment of polarization.

The absence of fitted parameters and a natural partitioning of the interaction energy into Coulomb, polarization, dispersion, exchange repulsion, and optional charge-transfer terms make EFP an attractive choice for analysis and interpretation of intermolecular forces. Moreover, by construction, it is free from the basis set superposition error that plagues *ab initio* calculations of weakly bound systems.

Recently, EFP has been extended to excited-states methods.[4–8] An important feature of the excited-state EFP implementation is that, unlike standard QM/MM approaches, it includes polarization response of the environment to the electronic excitation/ionization of the QM part,[4–7,9] which is important for quantitative accuracy.

In contrast to the continuum solvation models,[10–13] EFP is an explicit solvent model and allows one to describe solvent–solute specific interactions such as hydrogen bonding.

† Here, we consider a more mature version of EFP, so-called EFP2. The original EFP implementation, EFP1, involved rigorous treatment of electrostatic and polarization, but relied on fitting the remaining contributions (consisting mainly of exchange-repulsion and charge-transfer).

[a] *D. Ghosh, V. Vanovschi, I. Kaliman, A. I. Krylov*
Department of Chemistry, University of Southern California, Los Angeles, California 90089-0482
E-mail: krylov@usc.edu

[b] *D. Ghosh*
Physical Chemistry Division, National Chemical Laboratory (CSIR), Pune 411008, India

[c] *D. Kosenkov, J. Flick, I. Kaliman, L. V. Slipchenko*
Department of Chemistry, Purdue University, West Lafayette, Indiana 47907
E-mail: slipchenko@purdue.edu

[d] *D. Kosenkov*
Department of Chemistry, Med. Tech. and Physics, Monmouth University, West Long Branch, New Jersey 07764

[e] *Y. Shao*
Q-Chem Inc., 5001 Baum Boulevard, Suite 690, Pittsburgh, Pennsylvania 15213

[f] *A. T. B. Gilbert*
Research School of Chemistry, Australian National University, Canberra, Australian Capital Territory 0200, Australia

Contract/grant sponsor: NIH-SBIR grant with Q-Chem, Inc. (to A.I.K. and L.V.S.); Contract/grant sponsor: National Science Foundation; contract/grant number: CHE-0951634 (to A.I.K.); Contract/grant sponsor: Humboldt Research Foundation (Bessel Award) (to A.I.K.); Contract/grant sponsor: National Science Foundation CAREER grant; contract/grant numbers: CHE-0955419 (to L.V.S.); Contract/grant sponsor: Purdue University (to L.V.S.).

EFP (in particular, its early incarnation, EFP1) is similar to the so-called polarizable embedding (PE) approach.[14,15] The description of electrostatics and polarization is nearly identical in EFP and PE; however, EFP features more rigorous treatment of dispersion and exchange repulsion.

The performance of EFP for noncovalent interactions has been benchmarked against high-level *ab initio* methods for a variety of systems, including those dominated by hydrogen bonding, $\pi-\pi$, or mixed interactions.[16–26] Recently, a systematic benchmark study of EFP using a popular S22 dataset for noncovalent interactions has been published.[27] The reported[27] mean absolute deviation (MAD) of EFP relative to the high-level *ab initio* results, CCSD(T) (coupled-cluster with single and double substitutions and perturbative inclusion of triple excitations) extrapolated to the complete basis set limit, was 0.9 kcal/mol, which is comparable to the accuracy of MP2 and outperforms classical force fields.

The extension of EFP to methods targeting electronically excited and open-shell species[4,6,7,9,28] has enabled accurate calculations of electronically excited and ionized states in bulk solvents.[5,7,29–31]

The EFP was originally developed and implemented in GAMESS-US.[32,33] Recently, it was implemented[23] in the Q-CHEM electronic structure program,[34,35] which extended the scope of possible applications by combining EFP with state-of-the-art electronic structure methods. The unique QM/EFP features available in Q-CHEM include a suite of coupled-cluster (CC) and equation-of-motion CC (EOM-CC) methods[36–38] (for excitation energies, ionization potentials, electron affinities, spin-flip,[39] double spin-flip,[40] and double ionization potential[41,42]), SOS-CIS(D) (scaled-opposite spin configuration interaction singles with perturbative account of double excitations, Ref. [43]), as well as time-dependent density functional theory (TD-DFT) with long-range and dispersion corrected functionals,[44–50] collinear and noncollinear spin-flip TD-DFT method,[51,52] and so on.

In this article, we describe technical aspects of the EFP implementation in Q-CHEM to aid further EFP developments in Q-CHEM and other electronic structure packages. We also provide detailed instructions for users of how to set up and execute EFP calculations. In addition, we provide a set of auxiliary scripts to aid setting up and performing EFP calculations. Finally, the interface with the IQMOL and WEBMO software[53,54] enabling visualization of EFP calculations is described.

The EFP Formalism and Implementation

The detailed description of the EFP formalism is available in several original and review articles.[3,55,56] The derivation and the programmable expressions of all EFP terms can be found in Refs. [1, 2, 18, 22, 57–60]. The details of the present implementation are given in Refs. [4, 7, 23]. Here, we provide only a brief summary of the steps involved in a EFP calculation to accompany the EFP classes description.

The total energy of a system containing effective fragments and a QM part consists of the interactions between the effective fragments ($E^{\mathrm{ef-ef}}$) and the energy of the QM region in the field of the fragments. The former includes Coulomb, polarization, dispersion, and exchange-repulsion contributions:

$$E^{\mathrm{ef-ef}} = E_{\mathrm{Coul}} + E_{\mathrm{pol}} + E_{\mathrm{disp}} + E_{\mathrm{ex-rep}}. \tag{1}$$

The Coulomb interactions between the effective fragments are described by using distributed multipoles (up to octopoles) located at each atom and bond mid-points. Polarization term is computed using anisotropic polarizability tensors positioned at the centroids of the localized molecular orbitals. Induced dipoles appearing at the polarizability centers are iterated until self-consistency. Dispersion interactions are expressed as the first $n = 6$ term in the London series:

$$E_{\mathrm{disp}} = \sum_{n=6} \frac{C_n}{R^n} \tag{2}$$

that corresponds to the instantaneous dipole induced dipole interactions. Effects of the higher-order terms are estimated as 1/3 of the C_6 term.[58] Distributed C_6 coefficients are obtained on-the-fly using the dynamic polarizabilities integrated over the imaginary frequency range.

The classical multipolar expansion diverges when fragments approach each other too closely, because then the actual electron density of the fragments is not represented well by point multipoles. To improve the description of interactions at short interfragment separations, the Coulomb, polarization, and dispersion terms are modulated by damping (or screening) functions.[61]

The exchange repulsion is the only repulsive term in the EFP potential. Uniquely to EFP, this nonclassical interaction is expressed as an expansion in the intermolecular overlap truncated at the quadratic term.[62,63] The overlap integrals for each interacting pair of fragments are evaluated on-the-fly using a basis set and a localized wave function stored as fragment parameters.

The QM-EF interactions are computed using electronic embedding approach as described below. The Coulomb and polarization parts of the EFP potential contribute to the quantum Hamiltonian H via one-electron terms

$$\hat{H} = \hat{H}_0 + \langle p|\hat{V}^{\mathrm{Coul}} + \hat{V}^{\mathrm{pol}}|q\rangle, \tag{3}$$

where H_0 is an unperturbed Hamiltonian of the QM part, $\hat{V}^{\mathrm{Coul}}$ and $\hat{V}^{\mathrm{pol}}$ are electrostatic and polarization perturbations, respectively; $|p\rangle$,$|q\rangle$ denote molecular orbitals in the QM part. Both terms represent classical electrostatic interactions; $\hat{V}^{\mathrm{Coul}}$ is a Coulomb potential due to fragments nuclear charges and electron density represented by multipole expansion, whereas $\hat{V}^{\mathrm{pol}}$ describes electrostatic field due to induced dipoles (thus, it depends on the polarizabilites of the fragments). The induced dipoles are iterated until self-consistency with each other and with the electronic wave function (QM part). In our implementation, the dispersion and exchange-repulsion QM-EF interactions are treated as additive corrections to the total energy, as in the fragment–fragment interactions. Thus, the total ground-state (or, more precisely, reference-state) energy of the QM/EFP system is:

$$\begin{aligned} E_{\mathrm{gr}}^{\mathrm{QM/EFP}} &= \langle \Phi_{\mathrm{gr}}|\hat{H}_0 + \hat{V}^{\mathrm{Coul}} + \hat{V}_{\mathrm{gr}}^{\mathrm{pol}}|\Phi_{\mathrm{gr}}\rangle + E_{\mathrm{Coul}} + E_{\mathrm{pol,gr}} \\ &\quad + E_{\mathrm{disp}} + E_{\mathrm{ex-rep}} \end{aligned} \tag{4}$$

where Φ_{gr} is the reference-state wave function, $\hat{V}^{\mathrm{Coul}}$ and $\hat{V}_{\mathrm{gr}}^{\mathrm{pol}}$ are Coulomb and polarization EFP contributions to the

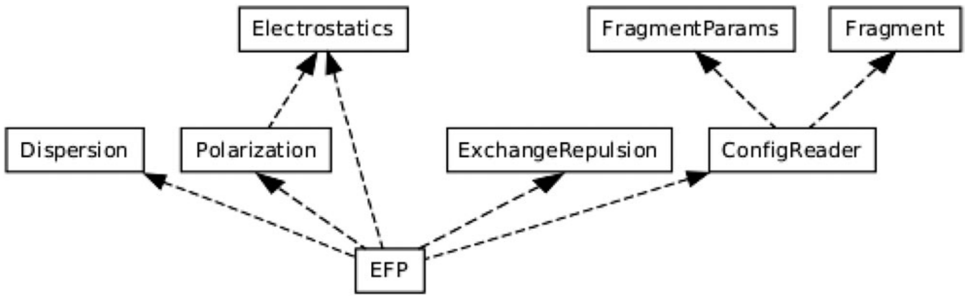


Figure 1. Class diagram for the EFP energy part of the implementation. Dashed arrows represent a dependence.

Hamiltonian (subscript “gr” means that the induced dipoles corresponding to the electronic density of the reference state, which is often referred to as “ground state,” are used), respectively. E_{Coul} is the electrostatic fragment–fragment interaction energy; $E_{\text{ex-rep}}$ and E_{disp} are the exchange repulsion and dispersion energies, respectively, of both the fragment–fragment and the QM-EFP interactions. $E_{\text{pol,gr}}$ is the self-consistent ground-state polarization energy of the QM/EFP system; it is computed using converged induced dipoles of the fragments and the fields due to the static fragment multipoles and the nuclei of the quantum region. Note that the polarization contributions appear both in the quantum Hamiltonian through $\hat{V}^{\text{pol}}$ and in the EFP energy as $E_{\text{pol,gr}}$; this is because size-consistency makes it impossible to completely separate QM-EFP and EFP-EFP polarization contributions.[1]

In correlated QM/EFP calculations, the induced dipoles are found for the Hartree–Fock (HF) density of the QM part; they are frozen in post-HF calculations. Finally, in multistate calculations such as EOM-CC, CI (configuration interaction), or TD-DFT, the polarization response of the fragments to changes of electronic density of the QM part are computed using first-order perturbation theory[4,7]:

$$\Delta E_k = -\int \mathrm{d}r(\hat{V}_k^{\text{pol}} - \hat{V}_{gr}^{\text{pol}})\rho^k(r), \qquad (5)$$

where ΔE_k is a state-specific correction to the kth target state energy, $\rho^k(r)$ is electronic density of the kth target state, and $\hat{V}_k^{\text{pol}}$ is a field due to induced dipoles on the fragments corresponding to the density of this state.

EFP class structure

The Q-CHEM implementation has a modular structure facilitating the interface between EFP and various QM methods in the hybrid QM/EFP schemes, as well as linking EFP to molecular dynamics and optimizer modules. The EFP module is implemented using C++ programming language.

Figure 1 shows the basic class structure of the EFP module. The main class *EFP* is responsible for managing the interactions between other classes and providing the interface to the rest of the Q-CHEM package. To interface EFP to another electronic structure package, this class needs to be rewritten using a package-specific interface. Classes *Electrostatics*, *Polarization*, *Dispersion*, and *Exchange-repulsion* encapsulate the functionality required for the corresponding individual energy and gradient components. Class *Polarization* has the dependence on the class *Electrostatics*, because induced dipoles depend on the electrostatic field of the multipoles. The *ConfigReader* class is responsible for reading the effective fragment parameters and the configuration of the system from the input file, and storing them in the internal representation using classes *FragmentParams* and *Fragment*.

The internal structure of *FragmentParams* class is shown in Figure 2. Labels above the arrows indicate names of the data member. This class encapsulates all the parameters for a given EFP fragment type: names and positions of atoms, positions and values of multipoles, positions and values of polarizability tensors, positions of dispersion points and values of direction-averaged isotropic dynamic polarizabilities at 12 imaginary frequencies, and parameters for exchange repulsion (which start with er_prefix) including positions of localized molecular orbital (LMO) centroids, atomic basis set, wave function, and Fock matrix of the fragment. Positions of all elements are defined in the internal coordinate system of the fragment. The system may contain more than one fragment type. For

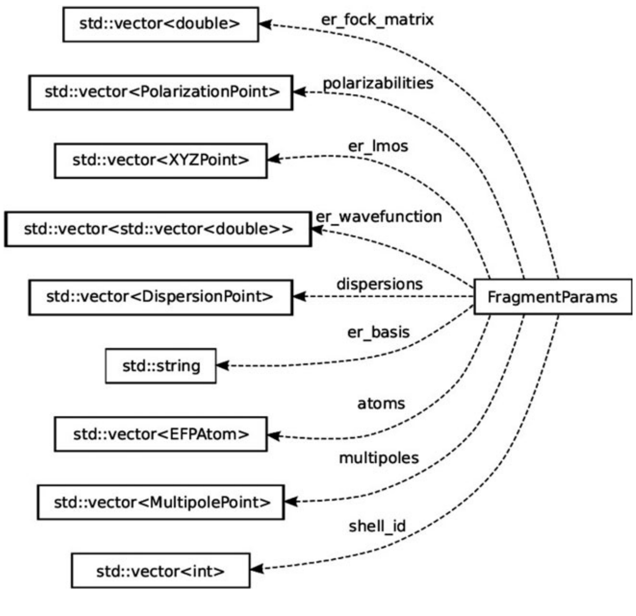


Figure 2. The internal structure of the *FragmentParams* class.

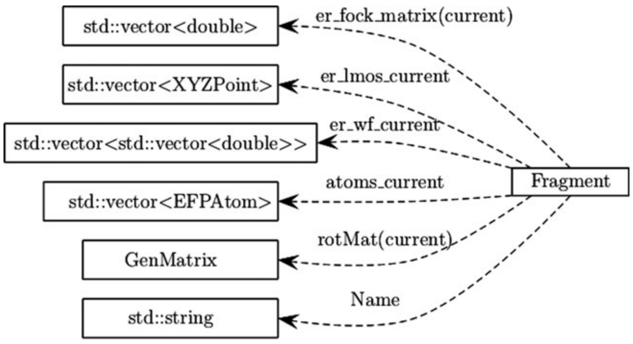


Figure 3. The internal structure of the *Fragment* class.

example, in a pure EFP simulation ("fragments-only") of aqueous solution of sodium chloride, there would be three fragment types: water, Na^+, and Cl^-.

The information in the *FragmentParams* class is filled in the beginning of the program execution upon reading and analyzing the EFP input section, and it is not modified after that.

The class *Fragment* keeps information about each individual effective fragment (see Fig. 3). It stores the fragment type (data member "Name") and its position and orientation in the coordinate frame of the system. It also stores the properties needed for exchange-repulsion calculations (denoted by "er") such as Fock matrix, coordinates of localized MOs, and the wave functions. This class is updated every time the position of the fragment is changed, for example, during optimization or molecular dynamics run. The details of the position and orientation of the fragment as well as the fragment parameters are described in "EFP Input Format."

The class *Electrostatics* comprises pointers to: (i) coordinates of the multipoles, (ii) number of multipole points, *mN*, (iii) charges, (iv) dipoles, (v) quadrupoles, (vi) octopoles, and (vii) damping parameters. It also contains pointers to objects of *Fragment* and *FragmentParams* classes. The important functions associated with the *Electrostatics* class are (i) UpdateElectrostatics, which updates the position of the multipole points and orientation of multipoles according to the rotation matrix, (ii) Hamiltonian, which calculates the perturbation to the electronic (quantum) Hamiltonian [$\hat{V}^{\text{Coul}}$ of eq. (3)] due to the electric field of multipoles, (iii) AddMultipoleField, which computes the electric field due to multipoles used in the evaluation of polarization energy, (iv) MakePolGradTerm, which assists in the gradient computation of the polarization, (v) Energy, which calculates the electrostatic energy of the EFP subsystem, and (vi) Gradient, which provides the electrostatic gradient in the EFP subsystem.

The classes *Polarization* and *Dispersion* are very similar to *Electrostatics*. *Polarization* consists of pointers to (i) coordinates of the polarization points, (ii) the number of polarization points, *pN*, (iii) the field due to external multipoles, extElectField, and (iv) induced dipoles. The *Dispersion* class consists of (i) the coordinates of dispersion points, (ii) the number of dispersion points, *dN*, and (iii) direction-averaged trace of dynamic polarizability tensor. Both classes have pointers to objects of *Fragment* and *FragmentParams* classes. The important functions associated with the classes are: (i) UpdateRepresentation, (ii) Energy, and (iii) Gradient. In addition, *Polarization* has the Hamiltonian function that calculates the perturbation to the quantum Hamiltonian [$\hat{V}^{\text{pol}}$ of eq. (3)] due to the field of induced dipoles.

EFP job sequence

Figure 4 shows the flow sequence of the EFP module and the Q-CHEM main structure. The *ReadParams* and *ReadFragments* calls instruct the EFP module to read the definitions of the fragment types and the system configuration from Q-CHEM's standard input (see section "EFP Input Format"). *ReadStandardParams* enables reading of standard EFP fragment types from the library (see section "The EFP library of standard fragments"). The *Init* call checks the effective fragment parameters and the configuration for consistency, creates and initializes the instances of the *Electrostatics*, *Polarization*, *Dispersion*, and *Exchange-Repulsion* classes. *UpdateRepresentation* is called every time when the geometry of the system changes. It performs rotation and translation of atoms, multipoles, polarization tensors, dispersion points, and LMOs in accordance with the coordinates and orientation for each individual fragment. The *NuclearEnergy* function computes and returns to Q-CHEM the EFP energy components that are independent of the wave function of the QM system, that is, interfragment electrostatics, dispersion, and exchange repulsion.

If the QM region is present, then at every iteration of the SCF procedure Q-CHEM calls the *Hamiltonian* and *WFDependentEnergy* functions. *Hamiltonian* updates the one-electron term of the *ab initio* Hamiltonian by the electrostatic and polarization EFP contributions. After updating the wave function accordingly, Q-CHEM calls *WFDependentEnergy* to compute the wave function-dependent part of the EFP energy (polarization).

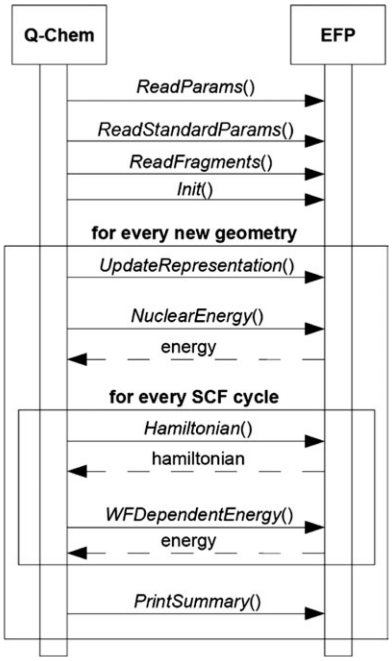


Figure 4. The flow of control between Q-CHEM and the EFP energy module.

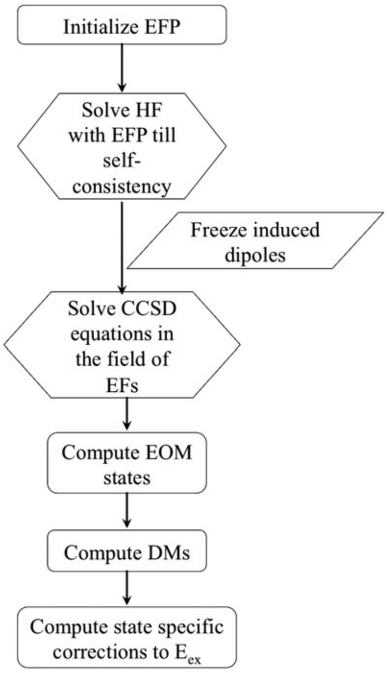

Initialize EFP
Solve HF with EFP till self-consistency
Freeze induced dipoles
Solve CCSD equations in the field of EFs
Compute EOM states
Compute DMs
Compute state specific corrections to E_{ex}

Figure 5. The work flow of a typical excited/ionized state calculation using EOM-CCSD/EFP. DM stands for one-particle density matrices.

WFDependentEnergy takes the electronic density of the *ab initio* region as an argument. When there is no QM region, Q-CHEM calls *WFDependentEnergy* only once with a zero electronic density as an argument.

Finally, *PrintSummary* is called to print the individual energy components for the interfragment EFP energy.

The current implementation also includes EFP only (no QM region) optimization and gradient computation. Similar to energy calculations, it starts with *ReadParams* and *ReadFragments* calls to read the fragment types and store the information. It is followed by an *Init* call which checks for consistency and initializes the instances of the *Electrostatics*, *Polarization*, *Dispersion*, and *Exchange-Repulsion* classes. The *NuclearEnergy* function overloaded (double EFP::NuclearEnergy(std::vector <double > &grad)) is called for optimization or gradient jobs. The gradient contains a vector of double precision numbers of dimension $6* <$ number of fragments $>$. They are Cartesian components of the gradient vector (Grad_x, Grad_y, Grad_z) and torque (Torque_x, Torque_y, Torque_z) for each fragment. After each step of optimization, *UpdateRepresentation* is called to update geometries of the fragments. Thus, the rotation matrix is updated along with the position of the atoms, multipoles, polarization tensor, dispersion points, and LMOs.

Performing EFP Calculations

Performing EFP calculations in Q-CHEM is easy, especially when using the library of standard fragments. Let us consider calculations of solvated thymine as an example. The user should prepare an input file specifying what type of calculation is requested (single point, etc.), type of the QM method (correlation level and basis set), and input a structure of the QM system (thymine) as well as coordinates of the fragments (solvated waters) and their library names (WATER_L). If the structure is available in a PDB format, a script is provided for converting it to the Q-CHEM input format.

For calculations of excited/ionized states and/or open-shell species, a number (and type) of target states needs to be specified (e.g., using EOM_EE_STATES keyword for an EOM-EE calculation of excited states).

The EFP related keywords are described later. The GUI (IQMOL and WEBMO) can be used for preparing EFP inputs. For more sophisticated calculations (such as using nonstandard fragments or user-defined parameters), several service scripts are provided.

The workflow of an EOM-EE-CCSD calculation of solvated thymine is shown in Figure 5.

EFP input format

The following sections of the Q-CHEM input contain EFP-related information: (i) $*molecule*, (ii) $ *rem*, (iii) $*efp_fragments*, and (iv) $*efp_params*. There are two formats for $*molecule* and $*efp_fragments* sections: one used for jobs containing EFP fragments only and the second is used for QM/EFP jobs.

When only EFP region is present (no QM part), the fragment positions are provided in the $*molecule* section. The first line defines the charge and multiplicity of the QM part; they are set to 0 and 1, respectively, as there is no QM part. The information for each fragment contains two lines, one providing its charge and multiplicity and the other specifying the position vector of fragment's center of mass (COM) and its Euler angles. Information for each fragment is separated by a line containing "–". A typical input for the $*molecule* section for the EFP-only job with two fragments looks like:

```
$molecule
0 1
--
0 1
-0.30448173 -2.24210052 -0.29383131 -0.642499 1.534222
-0.568147
--
0 1 -0.60075437 1.36443336 0.78647823 3.137879 1.557344
-2.568550
$end
```

The $*efp_fragments* section should contain (in the same order) the names of the fragments specified in the $*molecule* section. For example:

```
$efp_fragments
BENZENE_L
BENZENE_L
$end
```

In the case of a hybrid QM/EFP job, the $*molecule* section should contain the description of the QM part only (which is identical to the standard Q-CHEM format for QM jobs), whereas the *$efp_fragments* section contains the names and positions of the effective fragments. For example:

```
$molecule
0 1
O1 0.47586 0.56326 0.53843
H2 0.77272 1.00240 1.33762
H3 0.04955 -0.23147 0.86452
$end
$efp_fragments
WATER_L -2.12417561 1.22597097 -0.95332054 -2.902133 1.734999 -1.953647
AMMONIA_L 1.04358758 1.90477190 2.88279926 -1.105309 2.033306 -1.488582
AMMONIA_L -4.16795656 -0.98129149 -1.27785935 2.526442 1.658262 -2.742084
$end
```

The Cartesian coordinates of the fragments specified in the *$molecule* or *$efp_fragments* sections are the coordinates of its COM in the same coordinate system as the QM part. The orientation of the fragment is specified by the Euler angles (α, β, γ) in terms of the three consequent rotations, as explained in Figure 6. The rotation matrix corresponding to these angles is:

$$\mathbf{R}=\begin{pmatrix} \cos\alpha\cos\gamma-\sin\alpha\cos\beta\sin\gamma & -\cos\alpha\sin\gamma-\sin\alpha\cos\beta\cos\gamma & \sin\beta\sin\alpha \\ \sin\alpha\cos\gamma+\cos\alpha\cos\beta\sin\gamma & -\sin\alpha\sin\gamma+\cos\alpha\cos\beta\cos\gamma & -\sin\beta\cos\alpha \\ \sin\beta\sin\gamma & \sin\beta\cos\gamma & \cos\beta \end{pmatrix} \tag{6}$$

The position x_i', y_i', z_i' of an atom i of an effective fragment can be computed as:

$$\begin{bmatrix} x_i' \\ y_i' \\ z_i' \end{bmatrix} = \mathbf{R}\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{7}$$

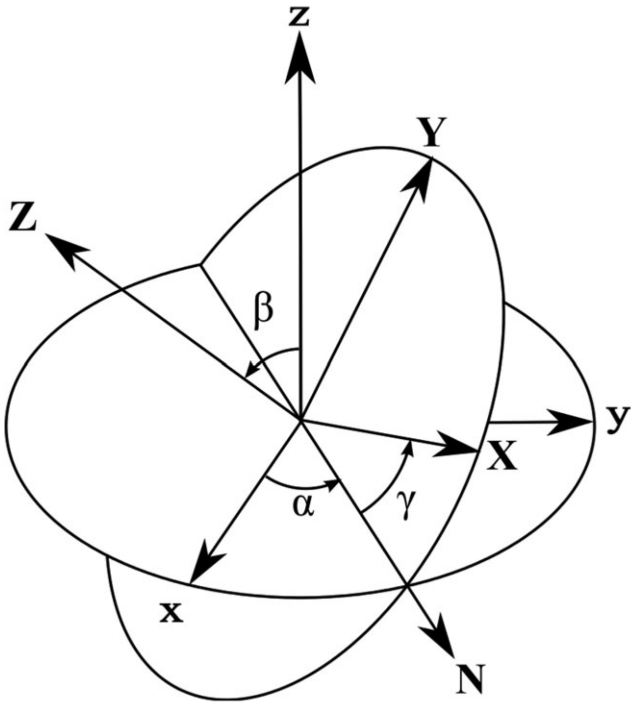


Figure 6. Representing the orientation of a fragment with the Euler angles. *xyz* is the fixed system, *XYZ* is the rotated system, *N* is the intersection between the *xy* and *XY* planes called the *line of nodes*. Transformation between the *xyz* and *XYZ* systems is given by α rotation in the *xy* plane, followed by β rotation in the new *y′z′* plane (around the *N* axis), followed by γ rotation in the new *x″y″* plane.

where x_i, y_i, z_i are the coordinates of the atom in the fragment frame, **R** is the rotation matrix defined in eq. (6), and *x, y, z,* are fragment's COM coordinates.

In the *$rem* section, (i) "EFP = TRUE" specifies that an EFP calculation has been requested, (ii) "efp_fragments_only = FALSE/TRUE" distinguishes between a hybrid QM/EFP or a "fragments-only" jobs, and (iii) "efp_input = FALSE/TRUE" differentiates between the QM/EFP type of input (i.e., coordinates of EFP fragments are provided in *$efp_fragments* section) or a pure EFP input (EFP coordinates are given in the *$molecule* section).

In the case of the hybrid QM/EFP jobs all the usual variables for the required QM job have to be included (see Q-CHEM's user manual, http://www.q-chem.com).

There are also boolean keywords that allow users to turn on and off various EFP and QM/EFP interaction terms. A list of them with the current default settings is provided below:

- EFP_ELEC (TRUE)
- EFP_POL (TRUE)
- EFP_EXREP (TRUE)
- EFP_DISP (TRUE)
- EFP_QM_ELEC (TRUE)
- EFP_QM_POL (TRUE)
- EFP_QM_EXREP (FALSE)
- EFP_ELEC_DAMP (TRUE)
- EFP_DISP_DAMP (TRUE)
- EFP_QM_ELEC_DAMP (FALSE)

The *$efp_params* section is optional. If the *$efp_fragments* section contains the fragments that are included in the library of fragments (see Section "The EFP library of standard fragments") *$efp_params* may be omitted. However, if one wishes to use user-defined parameters or fragments that are not present in the fragment library, this section should be included. *$efp_params* provides the EFP potential for a particular type of the fragment, including information about multipoles, static and dynamic polarizabilities, parameters for exchange repulsion, and so on. A typical *$efp_params* section looks like:

```
$efp_ params
fragment <Name of fragment>
C -0.1908012267 -0.5247948651 0.0010972008
N -1.2026293848 -1.4577534979 -0.0079312871
C -2.2987002719 -0.7184880661 -0.0057519465
…
…
mult -0.1908012267 -0.5247948651 0.0010972008
cdamp 1.6654966670
-3.5077487676
-0.0425432014 0.1062122796 -0.0016925508
-0.4755552531 -0.6270138980 -1.6987483387 0.0252821834
0.0008336815 0.0050990162
0.0232752673 0.2799122420 0.0042146764 0.0019159258 ….
…
…
pol -2.1869866728 0.0807135416 0.1052729275
1.7644811062 2.3392508114 0.5873152422 -0.0422766110 ….
…
…
disp -2.1869866728 0.0807135416 0.1052729275
1.5637589095 1.5641364941 1.5662032257 1.5716934174 ….
…
…
er_basis <basis set>
er_wavefunction
1 4.08900539E-03 7.50961882E-03 7.85021979E-03 ….
…
er_fock_matrix
-0.9414556247 0.0010490578 -0.9148477223 -0.0013149275 ….
…
er_lmos
-2.1869866728 0.0807135416 0.1052729275
…
labels
A01O1 O 0.0000000000 0.0000000000 0.0664326840
…
$end
```

The first line consists of the name of the fragment. In the following lines, the Cartesian coordinates of a standard fragment in XYZ format in Å are provided. These coordinates are the reference point for the real fragments' positions and orientations and used to form the rotation matrix.

Then, a section containing information on electrostatic multipoles follows. If the fragment possesses more than one multipole point (which is typically the case), a separate entry for each multipole point appears. Each entry starts with a line "mult X Y Z" that contains coordinates of a multipole point. This line is followed by several (optional) lines containing parameters of this multipole point, namely its (i) electrostatic damping parameter (in a format "cdamp C"), (ii) charge (one real value), (iii) dipole (three real values), (iv) quadrupole (six real values), and (v) octopole (ten real values).

Each polarizability point requires two lines, the first line "pol X Y Z" containing Cartesian coordinates of the polarizability point (in a coordinate system of the standard fragment), followed by the line with a nine-component polarizability tensor.

Similarly, each dispersion point is defined by two lines, one (in a format "disp X Y Z") containing the Cartesian coordinates of the dispersion point, and the second line providing direction-averaged isotropic dynamic polarizabilities at 12 imaginary frequencies.

Exchange-repulsion parameters should include the following sections: (i) er_basis, which specifies the name of the basis set, (ii) er_wavefunction, followed by the LMO coefficients of the electronic wave function corresponding to the basis set defined above, (iii) er_fock_matrix, with $N*(N-1)/2$ symmetry-unique elements of a Fock matrix (upper triagonal) of the fragment, and (iv) er_lmos, followed by the Cartesian coordinates of the LMO centroids.

Finally, optional "labels" section provides specific names (labels) for the fragment's atoms. This section is ignored by Q-CHEM's parser but is used for converting the PDB coordinates into the EFP format (see Supporting Information).

Computing EFP parameters

The EFP parameters can be generated in GAMESS using MAKEFP job (RUNTYPE = MAKEFP). The recommended basis set for the parameter generation is 6-311++G(3df,2p). An example of a GAMESS MAKEFP input file is provided in Supporting Information.

The EFP parameters should be computed once for every new fragment type and then can be reused in all subsequent EFP computations.

To facilitate the conversion from the GAMESS format to the Q-CHEM format, converter scripts ("efp_g2qinp.pl" and "efp_g2qlib.pl") have been developed in Perl. The detailed instructions on using these scripts are provided in Supporting Information. These scripts can be easily adapted to convert the parameters to another format.

The EFP library of standard fragments

To make EFP calculations simpler and facilitate inter-operability of the EFP jobs performed in the Q-CHEM and GAMESS packages, the library of standard fragments with precomputed EFP parameters was designed and implemented.[23] Currently, in addition to the common organic solvents (acetone, carbon tetrachloride, dichloromethane, methane, methanol, ammonia, acetonitrile, water dimethyl sulfoxide, benzene, phenol, and toluene), the library contains five nucleic acid bases (adenine, thymine, cytosine, guanine, and uracil). The details on generating the EFP parameters for these fragments can be found in Ref. [23]. The most recent information about the library can be found in the Q-CHEM user manual (http://www.q-chem.com).

The workflow of the EFP library creation and usage is shown in Figure 7. The geometry of the fragment is used as input for generating the EFP parameters in GAMESS. The EFP parameters are converted to the Q-CHEM EFP library format using "efp_g2qinp.pl" script (see Section "Computing EFP parameters" and

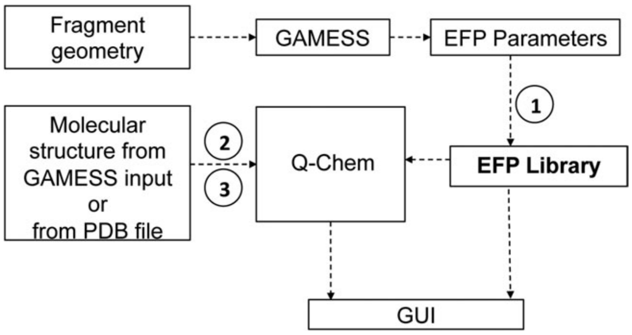


Figure 7. Workflow of the EFP library utilization. The EFP parameters are generated in GAMESS based on the geometry of the effective fragment provided. Using script (1) "efp_g2qinp.pl" the EFP parameters are converted into the Q-CHEM EFP library format. Coordinates of a molecular system in the GAMESS input format or in the PDB format are converted to Q-CHEM's *$efp_fragments* format using scripts (2) and (3), respectively. The EFP calculations are performed in Q-CHEM. IQMOL or WEBMO, can be used to visualize Q-CHEM's input and output files.

Supporting Information for detail). The library can be directly accessed from Q-CHEM. Coordinates of the system can be converted from the Cartesian or PDB formats to the Q-CHEM-compatible format (as in the *$efp_fragments* section) using the scripts provided in Supporting Information. The IQMOL and WEBMO visualization software[53,54] can be used for visualizing Q-CHEM's input and output files. The format of the Q-CHEM EFP library files is also compatible with the WEBMO and IQMOL packages.

The support of the EFP library is implemented through the Q-CHEM classes that provide input parsing (in ParseQInput.C) and main EFP declarations (in EFP.h). The call stack is shown in Figure 8. If the fragment parameters are found in the input file, then the basic EFP method, ReadStandardParams, is called which in turn calls the LoadParamsFromLib method of the *ConfigReader* class if the EFP parameters should be loaded from the library.

The EFP operation workflow relies on a set of service scripts facilitating information exchange between the computational packages supporting common interfaces and file formats. The scripts are written in Perl and distributed as a part of standard Q-CHEM distribution. The details of using these scripts as well as scripts themselves are provided in Supporting Information.

Interface with IQMOL

IQMOL is a fully integrated molecular builder and viewer[53] for the Q-CHEM package.[34] It is an open-source software available for the Windows, Linux, and Mac OS X platforms. We extended IQMOL functionality to handle visualization of EFP calculations as well as creating inputs using the standard fragment library.

The appearance of the atoms and bonds is fully customizable, but by default the QM region appears as balls and sticks, and the fragments appear as tubes to easily distinguish between them. As illustrated in Fig. 9, fragments can be added by clicking the "Add Fragment" button that appears in the tool bar at the top of the window. This opens a small popup fragment dialog that allows the user to select either EFP fragments or molecules from the existing internal library. The parameters for the fragments are stored in a local directory structure (the location of which is configurable), and the tree structure that appears in the dialog reflects this directory structure. This makes it easy for the user to extend the list of available EFP fragments by simply adding the parameter file to the appropriate directory, and the fragment will appear in the fragment dialog the next time IQMOL is opened. A small image file with the same base name can also be added to the directory, and this will appear in the fragment dialog when the new fragment is selected. If the name of the fragment does not end with "_L," then it is assumed that the fragment does not exist in the Q-CHEM library and the contents of the file will be entered into the $efp_params section when generating an input file.

The content of the EFP library looks very similar to the *$efp_params* section (see example of a typical *$efp_params* section in Section "EFP input format"). The first line contains "fragment <name of fragment>", followed by the atoms and their positions. This is followed by the multipoles (positions and moments), static and dynamic polarizabilities (position and tensor elements), basis in which exchange repulsion have been calculated, localized molecular orbitals, elements of the Fock matrices, and so forth.

IQMOL can optimize structures using the built-in force fields available in OPEN BABEL and, using these, it is possible to preoptimize QM, QM/EFP, and fragment-only structures before submitting the calculation to Q-CHEM. Systems with effective fragments are first allowed to fully relax using the selected force field, and then the fragment positions and orientations are recovered by minimizing the RMSD between the optimized coordinates and the fragment geometry for each fragment in the system.

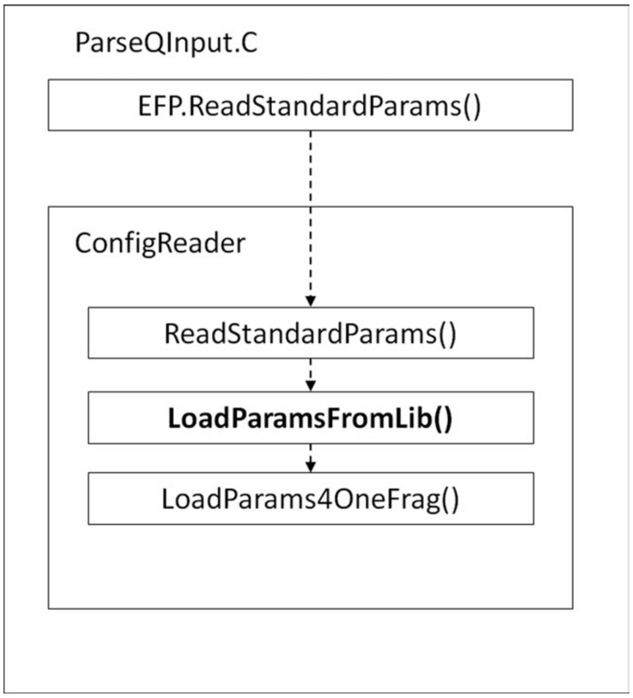


Figure 8. Call stack of Q-CHEM's EFP library: the ReadStandardParams function of the EFP class calls the ReadStandardParams, LoadParamsFromLib, and LoadParams4OneFrag functions of the parent class, *ConfigReader*. The LoadParamsFromLib function provides interface to the EFP library files.

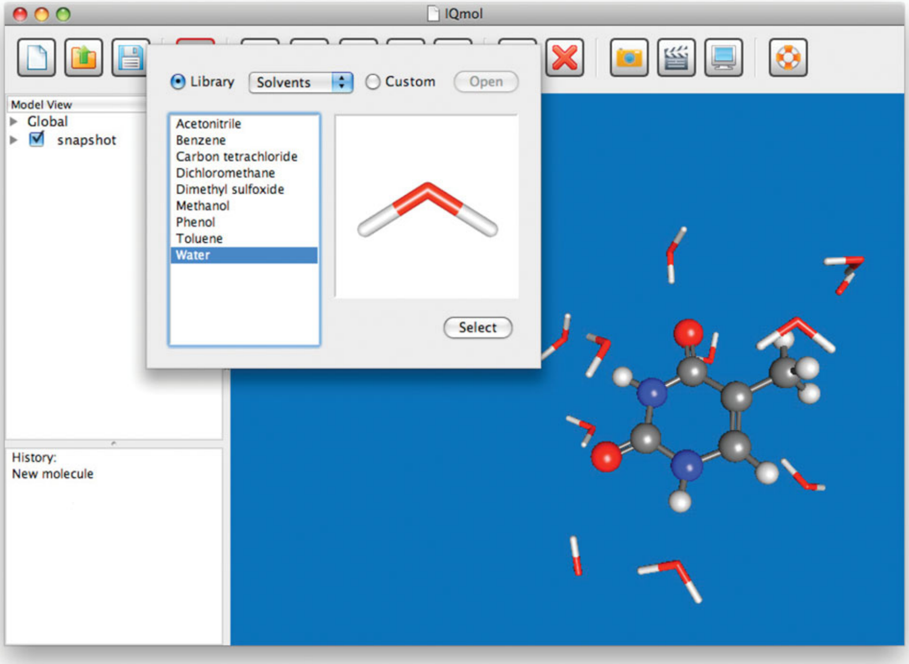
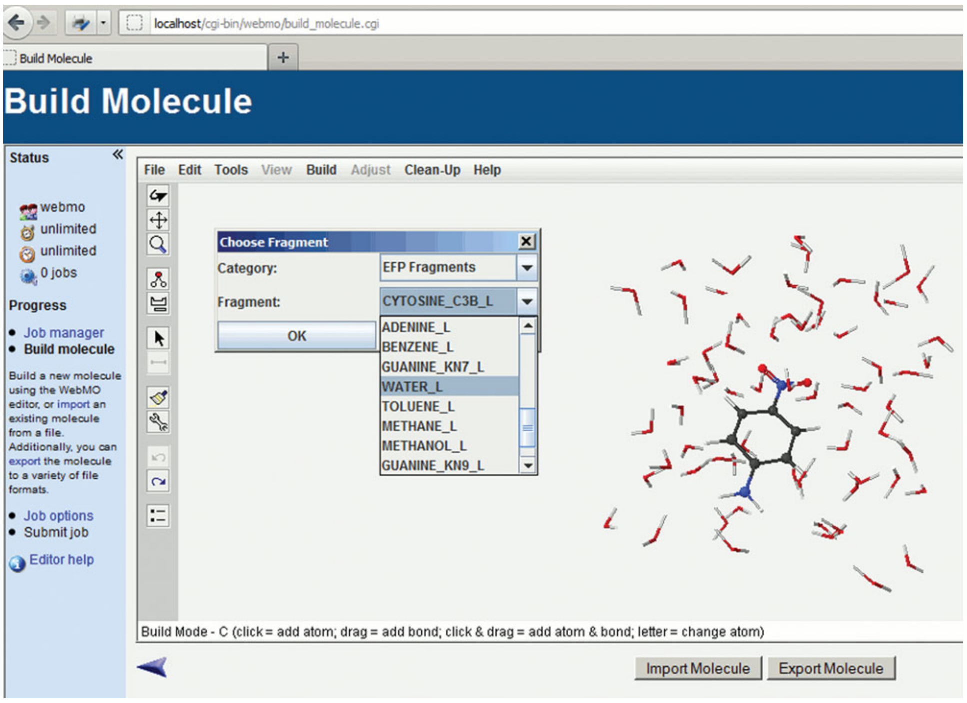


Figure 9. IQMOL can visualize Q-CHEM's EFP outputs and includes 'Add Fragment' feature that allows users to build a QM/EFP input by adding standard fragments from the library or importing custom fragments. The screenshot shows a snapshot of the first solvation shell of thymine.

Interface with WEBMO

To extend visualization options, we also interfaced Q-CHEM's EFP library with the WEBMO package[54] (see Fig. 10). WEBMO uses a client–server computing model in which some tasks are executed on the server and other tasks—on the client machine. The server part consists of a set of CGI (common gateway interface) scripts written in Perl. These server scripts are responsible, among other tasks, for parsing Q-CHEM's output files. The client part of WEBMO is written in Java and is responsible for processing input files as well as visualization of input and output files.


localhost/cgi-bin/webmo/build_molecule.cgi

Build Molecule

Build a new molecule using the WebMO editor, or import an existing molecule from a file. Additionally, you can export the molecule to a variety of file formats.

Category: EFP Fragments

Fragment: CYTOSINE_C3B_L

Build Mode - C (click = add atom; drag = add bond; click & drag = add atom & bond; letter = change atom)

Figure 10. WEBMO can be used for visualization of Q-CHEM's EFP inputs and outputs, as in this EFP input file containing one *p*-nitroaniline molecule (QM region) solvated by the EFP water molecules. The fragment manager ('Choose Fragment') panel can be used to add new EFP fragments from the EFP library to the input.

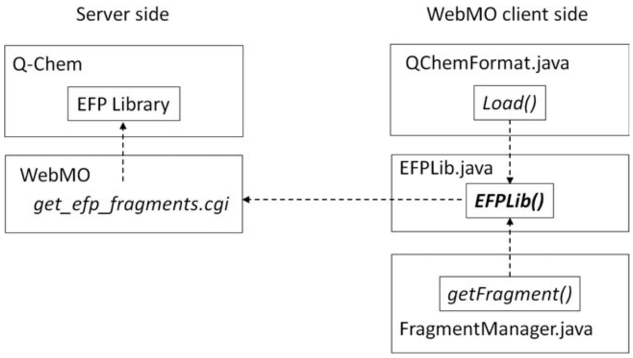


Figure 11. WEBMO employs client–server architecture. On the server side, EFP data is obtained from Q-CHEM and sent to the client. On the client side, EFP data is processed by Java code in the *EFPLib* and *FragmentManager* classes to read, parse and create EFP inputs.

The scheme of the WEBMO operation is shown in Figure 11. On the server side, WEBMO uses *get_efp_fragments.cgi* script to obtain data from Q-CHEM's EFP library and sends it to the client. On the client side, Java code of the *EFPLib* class is used by the *QChemFormat* and *FragmentManager* classes to process EFP input files. The *QChemFormat* class is responsible for reading and parsing Q-CHEM's input files, whereas the *FragmentManager* class is used to build new input files.

As follows from Figure 11, the server side of WEBMO needs information about location of the Q-CHEM directories and availability of the EFP library on the server. This information is provided by using the WEBMO built-in mechanism for managing external program packages *InterfaceManager*.

To make the client part of WEBMO more stable with respect to working with the EFP library, the latter was integrated into the Java code (i.e., the EFP library was made a part of the Java archive file JAR). In this case, even if the EFP library is unavailable on a server or client–server connection is interrupted, the client is still able to use basic functionality of the EFP library.

The WEBMO/Q-CHEM interface can be easily extended to EFP calculations using other ab initio packages.

Conclusions and Outlook

We presented technical details of the EFP implementation in the Q-CHEM electronic structure package and described the interface with the IQMOL and WEBMO visualization software. The implementation opens access to state-of-the-art electronic structure methods available in Q-CHEM, in particular, methods for electronically excited and open-shell species. The EFP implementation in Q-CHEM includes a library of standard effective fragments populated with validated parameters for commonly used fragments, such as standard solvents and nucleobases.

Detailed description of the class structure and the EFP module workflow will aid further EFP developments in Q-CHEM and other electronic structure packages. We also provide detailed instructions for users and a set of auxiliary scripts to aid setting up and performing EFP calculations.

Acknowledgments

The authors are grateful to Prof. Mark S. Gordon and Dr. Michael S. Schmidt for their generous help with the original implementation. Contributions from Prof. John M. Herbert, Dr. Chris Williams, Dr. Bernie Brooks, Prof. Rolin King, and Prof. J.R. Schmidt are gratefully acknowledged.

Keywords: effective fragment potential · hybrid QM/MM · nonempirical forcefield · Q-Chem · WebMO · IQmol · EFP library · modularcode

How to cite this article: D. Ghosh, D. Kosenkov, V. Vanovschi, J. Flick, I. Kaliman, Y. Shao, A. T. B. Gilbert, A. I. Krylov, L. V. Slipchenko, *J. Comput. Chem.* **2013,** *34,* 1060–1070. DOI: 10.1002/jcc.23223

Additional Supporting Information may be found in the online version of this article.

[1] P. N. Day, J. H. Jensen, M. S. Gordon, S. P. Webb, W. J. Stevens, M. Krauss, D. Garmer, H. Basch, D. Cohen, *J. Chem. Phys.* **1996**, *105,* 1968.
[2] M. S. Gordon, L. Slipchenko, H. Li, J. H. Jensen, In Annual Reports in Computational Chemistry, Vol. 3; D. C. Spellmeyer, R. Wheeler, Eds.; Elsevier, **2007**; Chapter 10, pp. 177–193. doi: 10.1016/S1574-1400(07)03010-1.
[3] M. S. Gordon, M. A. Freitag, P. Bandyopadhyay, J. H. Jensen, V. Kairys, W. J. Stevens, *J. Phys. Chem. A* **2001**, *105,* 293.
[4] L. V. Slipchenko, *J. Phys. Chem. A* **2010**, *114,* 8824.
[5] D. Kosenkov, L. V. Slipchenko, *J. Phys. Chem. A* **2011**, *115,* 392.
[6] A. DeFusco, N. Minezawa, L. V. Slipchenko, F. Zahariev, M. S. Gordon, *J. Phys. Chem. Lett.* **2011**, *2,* 2184.
[7] D. Ghosh, O. Isayev, L. V. Slipchenko, A. I. Krylov, *J. Phys. Chem. A* **2011**, *115,* 6028.
[8] S. Yoo, F. Zahariev, S. Sok, M. S. Gordon, *J. Chem. Phys.* **2008**, *129,* 8.
[9] P. Arora, L. V. Slipchenko, S. P. Webb, A. Defusco, M. S. Gordon, *J. Phys. Chem. A* **2010**, *114,* 6742.
[10] M. Cossi, V. Barone, *J. Chem. Phys.* **1998**, *109,* 6246.
[11] V. Barone, M. Cossi, *J. Phys. Chem. A* **1998**, *102,* 1995.
[12] J. Tomasi, B. Mennucci, R. Cammi, *Chem. Rev.* **2005**, *105,* 2999.
[13] A. W. Lange, J. M. Herbert, *J. Chem. Phys.* **2010**, *133,* 24411.
[14] K. Sneskov, T. Schwabe, O. Christiansen, J. Kongsted, *Phys. Chem. Chem. Phys.* **2011**, *13,* 18551.
[15] K. Sneskov, T. Schwabe, J. Kongsted, O. Christiansen, *J. Chem. Phys.* **2011**, *134,* 104108.
[16] I. Adamovic, M. S. Gordon, *J. Phys. Chem. A* **2006**, *110, 10267.*
[17] I. Adamovic, H. Li, M. H. Lamm, M. S. Gordon, *J. Phys. Chem. A* **2006**, *110,* 519.
[18] L. V. Slipchenko, M. S. Gordon, *J. Comput. Chem.* **2007**, *28,* 276.
[19] T. Smith, L. V. Slipchenko, M. S. Gordon, *J. Phys. Chem. A* **2008**, *112,* 5286.
[20] J. M. Mullin, M. S. Gordon, *J. Phys. Chem. B* **2009**, *113,* 8657.
[21] J. M. Mullin, M. S. Gordon, *J. Phys. Chem. B* **2009**, *113,* 14413.
[22] L. V. Slipchenko, M. S. Gordon, *J. Phys. Chem. A* **2009**, *113,* 2092.
[23] D. Ghosh, D. Kosenkov, V. Vanovschi, C. Williams, J. Herbert, M. S. Gordon, M. Schmidt, L. V. Slipchenko, A. I. Krylov, *J. Phys. Chem. A* **2010**, *114,* 12739.
[24] T. Smith, M. S. Gordon, L. V. Slipchenko, *J. Phys. Chem. A* **2011**, *115,* 4598.
[25] Q. A. Smith, M. S. Gordon, L. V. Slipchenko, *J. Phys. Chem. A* **2011**, *115,* 11269.
[26] M. D. Hands, L. V. Slipchenko, *J. Phys. Chem. B* **2012**, *116,* 2775.
[27] J. C. Flick, D. Kosenkov, E. G. Hohenstein, C. D. Sherrill, L. V. Slipchenko, *J. Chem. Theory Comput.* **2012**, *8,* 2835.
[28] N. Minezawa, N. D. Silva, F. Zahariev, M. S. Gordon, *J. Chem. Phys.* **2011**, *134,* 054111.

[29] D. Ghosh, A. Roy, R. Seidel, B. Winter, S. Bradforth, A. I. Krylov, *J. Phys. Chem. B* **2012**, *116,* 72697280.

[30] A. DeFusco, J. Ivanic, M. W. Schmidt, M. S. Gordon, *J. Phys. Chem. A* **2011**, *115,* 4574.

[31] S. Sok, S. Y. Willow, F. Zahariev, M. S. Gordon, *J. Phys. Chem. A* **2011**, *115,* 9801.

[32] M. W. Schmidt, K. K. Baldridge, S. T. Elbert J. A. Boatz, M. S. Gordon, S. Koseki J. H. Jensen, N. Mastunaga, K. A. Nguyen, T. L. Windus S. Su, M. Dupuis, J. A. Montgomery, *J. Comput. Chem.* **1993**, *14,* 1347.

[33] M. S. Gordon, M. W. Schmidt, Theory and Applications of Computational Chemistry. *Advances in Electronic Structure Theory: GAMESS a Decade Later*; Elsevier, 2005; Chapter 41.

[34] Y. Shao, L. Fusti-Molnar, Y. Jung, J. Kussmann, C. Ochsenfeld, S. Brown, A. T. B. Gilbert, L. V. Slipchenko, S. V. Levchenko, D. P. O'Neill, R. A. Distasio, Jr., R. C. Lochan, T. Wang, G. J. O. Beran, N. A. Besley, J. M. Herbert, C. Y. Lin, T. Van Voorhis, S. H. Chien, A. Sodt, R. P. Steele, V. A. Rassolov, P. Maslen, P. P. Korambath, R. D. Adamson, B. Austin, J. Baker, E. F. C. Bird, H. Daschel, R. J. Doerksen, A. Dreuw, B. D. Dunietz, A. D. Dutoi, T. R. Furlani, S. R. Gwaltney, A. Heyden, S. Hirata, C.-P. Hsu, G. S. Kedziora, R. Z. Khalliulin, P. Klunziger, A. M. Lee, W. Z. Liang, I. Lotan, N. Nair, B. Peters, E. I. Proynov, P. A. Pieniazek, Y. M. Rhee, J. Ritchie, E. Rosta, C. D. Sherrill, A. C. Simmonett, J. E. Subotnik, H. L. Woodcock, III, W. Zhang, A. T. Bell, A. K. Chakraborty, D. M. Chipman, F. J. Keil, A. Warshel, W. J. Hehre, H. F. Schaefer, III, J. Kong, A. I. Krylov, P. M. W. Gill, M. Head-Gordon, *Phys. Chem. Chem. Phys.* **2006**, *8,* 3172.

[35] A. I. Krylov, P. M. W. Gill, *WIREs Comput. Mol. Sci.* **2012**. doi: 10.1002/wcms.1122.

[36] A. I. Krylov, *Annu. Rev. Phys. Chem.* **2008**, *59,* 433.

[37] R. J. Bartlett, *Mol. Phys.* **2010**, *108,* 2905.

[38] K. Sneskov, O. Christiansen, Wiley Interdiscip. Rev. *Comput. Mol. Sci.* **2011**, *2,* 566.

[39] A. I. Krylov, *Acc. Chem. Res.* **2006**, *39,* 83.

[40] D. Casanova, L. V. Slipchenko, A. I. Krylov, M. Head-Gordon, *J. Chem. Phys.* **2009**, *130,* 044103.

[41] M. Nooijen, R. J. Bartlett, *J. Chem. Phys.* **1997**, *107,* 6812.

[42] T. Kuś, A. I. Krylov, *J. Chem. Phys.* **2011**, *135,* 084109.

[43] Y. M. Rhee, M. Head-Gordon, *J. Phys. Chem. A* **2007**, *111,* 5314.

[44] P. M. W. Gill, R. D. Adamson, J. A. Pople, *Mol. Phys.* **1996**, *88,* 1005.

[45] H. Iikura, T. Tsuneda, T. Yanai, K. Hirao, *J. Chem. Phys.* **2001**, *115,* 3540.

[46] J.-D. Chai, M. Head-Gordon, *J. Chem. Phys.* **2008**, *128,* 084106.

[47] S. Grimme, *J. Comput. Chem.* **2004**, *25,* 1463.

[48] J.-D. Chai, M. Head-Gordon, *Phys. Chem. Chem. Phys.* **2008**, *10,* 6615.

[49] M. A. Rohrdanz, K. M. Martins, J. M. Herbert, *J. Chem. Phys.* **2009**, *130,* 054112.

[50] R. Baer, E. Livshits, U. Salzner, *Annu. Rev. Phys. Chem.* **2010**, *61,* 85.

[51] Y. Shao, M. Head-Gordon, A. I. Krylov, *J. Chem. Phys.* **2003**, *118,* 4807.

[52] Y. A. Bernard, Y. Shao, A. I. Krylov, *J. Chem. Phys.* **2012**, *136,* 204103.

[53] A. T. B. Gilbert, IQmol molecular viewer, Available at: http://iqmol.org. Last accessed December 12, 2012.

[54] Webmo 10.0, Available at: www.webmo.net. Last accessed December 12, 2012.

[55] M. S. Gordon, J. M. Mullin, S. R. Pruitt, L. B. Roskop, L. V. Slipchenko, J. A. Boatz, *J. Phys. Chem. B* **2009**, *113,* 9646.

[56] M. S. Gordon, D. G. Fedorov, S. R. Pruitt, L. V. Slipchenko, *Chem. Rev.* **2012**, *112,* 632.

[57] J. H. Jensen, *J. Chem. Phys.* **1996**, *104,* 7795.

[58] I. Adamovic, M. S. Gordon, *Mol. Phys.* **2005**, *103,* 379.

[59] H. Li, M. S. Gordon, J. H. Jensen, *J. Chem. Phys.* **2006**, *124, 214108.*

[60] Q. A. Smith, K. Ruedenberg, M. S. Gordon, L. V. Slipchenko, *J. Chem. Phys.* **2012**, *136,* 244107.

[61] L. V. Slipchenko, M. S. Gordon, *Mol. Phys.* **2009**, *107,* 999.

[62] J. H. Jensen, M. S. Gordon, *Mol. Phys.* **1996**, *89,* 1313.

[63] J. H. Jensen, M. S. Gordon, *J. Chem. Phys.* **1998**, *108,* 4772.

Received: 21 September 2012
Revised: 5 December 2012
Accepted: 8 December 2012
Published online on 15 January 2013