Semiempirical Quantum Chemistry on High-Performance Heterogeneous Computers

Inaugural-Dissertation

zur
Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von
Xin Wu
aus der Volksrepublik China

Mülheim an der Ruhr, 2013

Gedruckt mit der Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Heinrich-Heine-Universität Düsseldorf

Referent: Herr Prof. Dr. Walter Thiel
Korreferentin: Frau Prof. Dr. Christel Marian
Tag der mündlichen Prüfung:

Hiermit versichere ich, die hier vorgelegte Arbeit eigenständig und ohne unerlaubte Hilfe angefertigt zu haben. Die Dissertation wurde in der vorgelegten oder in ähnlicher Form noch bei keiner Institution eingereicht. Ich habe keine erfolglosen Promotionsversuche unternommen.
Mülheim an der Ruhr, den 11. Dezember 2013

Xin Wu

(Xin Wu)

Abstract

Two essential issues of semiempirical quantum chemical methods are addressed in this dissertation: performance and accuracy. First, semiempirical program code was developed for heterogeneous high-performance CPU-GPU computing platforms. In systematic test calculations on large molecules, the overall performance could be improved by one order of magnitude, which is unattainable on CPU-only parallel computers due to intrinsic constraints of the hardware architecture. Second, both local and global optimization algorithms for the parameters of semiempirical methods were implemented from scratch. The efficiency of parameterization was increased by taking advantage of coarse-grained parallelism on symmetric multiprocessors, which enables more comprehensive explorations of parameter space. This was demonstrated by reparameterization of OM2 and OM3 using dispersion corrections and by specific parameterizations for an enzymatic reaction, the hydride transfer catalyzed by dihydrofolate reductase, and for hydrogen bonding and proton transfer in water. The optimized CPU-GPU code was used in a systematic benchmark with full geometry optimization for a set of 28 proteins using 10 different semiempirical quantum chemical methods. These extensive computations unveiled some limitations of the currently available semiempirical methods that need to be addressed in future work.

Zusammenfassung

Zwei wesentliche Aspekte semiempirischer quantenchemischer Methoden werden in dieser Dissertation behandelt: Effizienz und Genauigkeit. Erstens wurde semiempirischer Programmcode für heterogene CPU-GPU-Systeme mit hoher Rechenleistung entwickelt. Bei systematischen Testrechnungen an großen Molekülen konnten dadurch Leistungssteigerungen bis zu einer Größenordnung erzielt werden, was durch Parallelisierung des seriellen Codes auf rein CPU-basierten Systemen problem- und hardwarebedingt nicht möglich ist. Zweitens wurden Algorithmen zur lokalen und globalen Optimierung von Parametern semiempirischer Methoden von Grund auf neu implementiert. Die Effizienz der Parametrisierung konnte durch Nutzung grobkörniger Parallelität auf symmetrischen Multiprozessorsystemen gesteigert werden, wodurch eine weitergehende Erkundung der Parameterfläche ermöglicht wurde. Dies wurde demonstriert durch Reparametrisierung von OM2 und OM3 unter Einbeziehung von Dispersionskorrekturen und durch spezifische Parametrisierungen für eine enzymatische Reaktion, den durch Dihydrofolatreductase katalysierten Hydridtransfer, und für Wasserstoffbrückenbindungen und Protonentransfer in Wasser. Mit dem optimierten CPU-GPU Code konnte ein systematisches Benchmarking mit vollständiger Geometrieoptimierung für einen Satz von 28 Proteinen mit zehn verschiedenen semiempirischen quantenchemischen Methoden durchgeführt werden. Diese ausgedehnten Rechnungen deckten einige Schwächen aktueller semiempirischer Methoden auf, die in zukünftigen Arbeiten behoben werden sollen.

Acknowledgements

I thank Prof. Dr. Walter Thiel for his open mind, risk-taking, and encouragement for my work on the GPU-accelerated semiempirical quantum chemistry program and other developments. It was worthwhile to learn the classic MNDO method by following the creator. Last but not least, I am grateful for the continuous financial support.

I would like to acknowledge Prof. Dr. Christel Marian for kindly accepting to act as a co-referee for my dissertation.

I am indebted to Dr. Axel Koslowski for his tireless help on parallel programming, code optimization, and translation of the abstract of this dissertation into German, to Horst Lenk for the support on specific hardware and software installations, to Ursula Scheifhacken for her assistance in management issues, and to Prof. Hai Lin for inspiration and fruitful collaboration.

I am thankful to all my colleagues at the Max-Planck-Institut für Kohlenforschung for many interesting scientific discussions.

Finally I am appreciative of the patience of my wife and our parents in China.

Recent work has brought chemistry closer to its ultimate role as a branch of applied mathematics.

Michael J. S. Dewar

Contents

Chapter 1

Introduction

Semiempirical quantum chemical methods are fast electronic structure approaches for studying molecular geometry, stability, spectroscopy, and chemical reactions [1]. The development of semiempirical methods has a long history. Soon after the advent of quantum mechanics [2], Dirac remarked the following in his paper [3]:

> The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.

A recommendation how to approach these difficulties was given immediately after this statement:

> It therefore becomes desirable that *approximate practical methods* of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation.

Hence it was anticipated already at that time that methods, which entail suitable approximations in the complicated mathematical formalism, would be able to provide feasible solutions to problems in chemistry.

One of the earliest semiempirical methods was a contribution of Erich Hückel. In 1931, he proposed a molecular orbital (MO) method for the energies of the π-electrons in planar conjugated hydrocarbons using linear combinations of atomic orbitals (LCAO) [4]. In spite of its simplicity, the Hückel method was able to rationalize and predict the properties and reactivities of conjugated compounds on a qualitative level, and it was widely used as a guide. However, there are two major limitations of the Hückel method. First, electron repulsion is explicitly neglected in the Hamiltonian. Second, since it is based on the approximation of the σ-π separation of electrons, only the π-electrons of planar molecules can be treated. The Pariser-Parr-Pople (PPP) method [5,6] introduces electron repulsion into the semiempirical π-electronic Hamiltonian and solves the secular equations in an iterative manner but it is still restricted to planar systems. Extensions of the Hückel method considering all valence electrons were developed later by Hoffmann [7,8] and others [9] for nonplanar molecules.

The corresponding Hamiltonians, however, are made up of one-electron terms only, still neglecting electron repulsion. General valence-electron semiempirical quantum chemical methods overcoming both limitations were initially developed by Pople and coworkers with three different levels of approximation for the two-electron integrals, namely CNDO (complete neglect of differential overlap), INDO (intermediate neglect of differential overlap) and NDDO (neglect of diatomic differential overlap) [10, 11]. The NDDO model fulfills the requirement that the energy must be invariant with respect to a rotation of the molecule in the coordinate system and for an orthogonal transformation of the occupied orbitals. The model can be formally justified [12–16] in terms of orthogonalized Löwdin atomic orbitals [17]. The MNDO (modified neglect of diatomic overlap) method is the first successful NDDO-based approach for ground-state molecular properties, such as heats of formation, molecular geometries, ionization potentials, and dipole moments [18–20]. Moreover, the MNDO model constitutes the basis for the most popular semiempirical methods today, in particular AM1 (Austin Model 1) with empirical Gaussian functions for core-core interactions [21] and PM3 (Parametric Method 3) with more thoroughly optimized parameters [22, 23].

Some underlying deficiencies of the NDDO model prohibit further improvement of the accuracy merely through more complicated core repulsion functions or refinement of the parameters [1, 24–29]. The OMx (Orthogonalization Method x, $x = 1$, 2, and 3) methods were developed to go beyond the NDDO model by means of explicit inclusion of orthogonalization corrections in the Fock matrix to account for Pauli exchange repulsion [30–32]. Persistent improvements were obtained over MNDO-type methods for ground-state molecular properties. Furthermore, noteworthy enhancements are also observed in some particular areas, for example, for vertical excitation energies, rotational barriers, relative energies for different conformers, and hydrogen bonding [33–35].

The development of quantum chemistry has always been tightly coupled with the available computing technologies. In fact, the demand for a semiempirical treatment of molecules was dictated by the limited computational capabilities in the early days of quantum chemistry. For example, the first all-electron *ab initio* calculation of N_2 was done in 1955 by Scherr and two assistants with hand-operated mechanical calculators, which took two years to complete [36]. After the introduction of electronic computers into quantum chemistry, some groups started to develop computer programs so that approximate solutions of the non-relativistic Schrödinger equation for a molecular system could be calculated automatically [37–41]. These programs were written for completely sequential execution on one general-purpose central processing unit (CPU). Due to increasing clock speed and architectural advances, the performance of a single-core CPU followed Moore's law [42] which predicts doubling every 12–24 months.[1] Thus quantum-chemical calculations became faster and more complex algorithms could be implemented with every new processor generation. But as some physical limits associated with the design of silicon chips were approached about a decade ago, it became more and more difficult to increase the clock frequency of a single processor. Therefore many efforts were devoted to the parallelism of multi-processing units in order to keep the exponential growth of the computing power [43]. Since the 1990s, quantum chemists have been

[1] In its original formulation, Moore's law refers to the number of transistors on a silicon chip doubling every 12–24 months, but the exponential growth seems to apply for the resulting performance as well.

inspired by this tendency to parallelize their serial codes on high-performance computers with multiple CPUs for boosting performance [44]. Even more recently, high-performance computers with heterogeneous architectures have become available, e.g. computing nodes consisting of symmetrically parallel CPUs and multiple graphics processing units (GPUs) as accelerators for compute-intensive tasks, which benefit from the substantial peak arithmetic performance and memory bandwidth provided by the co-processors [45–47]. This has motivated many theoretical groups [48,49] to develop dedicated programs for such high-performance heterogeneous computers in the areas of quantum Monte Carlo methods [50,51], two-electron integral evaluation [52–57], density functional theory (DFT) [58–65], high-level correlated *ab initio* methods [66–73], and semiempirical quantum chemistry [74,75].

The accuracy of a semiempirical method is essentially determined by the theoretical model and the optimization of the involved parameters. This thesis does not address the important and challenging task to advance the available semiempirical quantum chemical models, but focuses on the second and more technical issue of parameterization. There have been numerous attempts to improve a given semiempirical method to some extent, and even to approach the limit of accuracy inherent to the underlying model, by careful parameterizations. Examples are RM1 [76], which is a reparameterization of AM1 [21], and the series of PMx ($x = 3$, 5, 6, and 7) methods [22,77–79]. Due to insufficient computation power, searches for the best parameters were usually confined to the vicinity of the initial estimates in the early days of semiempirical quantum chemistry [19]. Global optimizations of the parameters inside some predefined intervals became feasible later [80–83] with progress in computer hardware and software. Since an enormous number of function evaluations is generally required both in local and global optimization procedures, it is vital to take advantage of parallel computing technologies and to distribute independent function evaluations among multiple processors to speed up the parameterization. Nonetheless, experience has shown that it is necessary to carry out careful checks by human beings during the whole process to make sure that the obtained parameters are consistent with their physical meaning.

This dissertation is organized as follows: Chapter 2 gives a brief overview over the theory of semiempirical quantum chemistry. Chapter 3 describes the development of modules for the MNDO99 program [84] that enables accelerated semiempirical calculations on heterogeneous CPU-GPU computing platforms. The results, especially the speedups compared with CPU-only systems, are reported in Appendices A and B. The development and the technical details of the parallel parameterization program on multiple CPUs are presented in Chapter 4 together with an illustrative application, i.e. the reparameterization of the OM2 and OM3 methods with dispersion corrections. Two specific parameterizations for an enzymatic reaction and for hydrogen bonding and proton transfer in water clusters are described in Appendices C and D, respectively. Chapter 5 summarizes systematic geometry optimizations for 28 proteins using 10 different semiempirical methods as a benchmark application. Finally, a short conclusion is offered in Chapter 6.

Chapter 2

Theoretical Background

Semiempirical quantum chemistry has a long history starting from 1931 [4]. This chapter does not intend a complete survey of its development in the past decades. Instead it will present the main concepts and the working formulas of the methods covered in the thesis, for the sake of coherence and for establishing notation. The underlying theory of the semiempirical methods and the numerous applications are described in several books [85–87] and reviews [1, 24–29, 88–91].

2.1 Nightmare of integrals

A non-relativistic time-independent quantum mechanical description of a molecular system is provided by the Schrödinger equation:

$$\mathscr{H}\Psi = E\Psi \tag{2.1}$$

where $\mathscr{H}$ is the Hamiltonian operator, E is the total energy of a particular quantum state described by the wave function Ψ. The exact solution to Equation (2.1) is limited to the simplest systems consisting of one nucleus and a single electron, i.e. the hydrogen atom and hydrogen-like ions. If each electron is assumed to move independently in an effective field generated by the fixed nuclei and the other electrons, the Hartree-Fock (HF) equations for single-electron wave functions ψ_i can be derived as

$$\mathscr{F}_i\psi_i = \varepsilon_i\psi_i. \tag{2.2}$$

The Fock operator $\mathscr{F}_i$ represents the average potential experienced by the ith electron. $\{\psi_i\}$ forms a set of orthonormal molecular orbitals (MOs) and ε_i is the orbital energy. It is, however, impossible to solve Equation (2.2) analytically for a molecule. A strategy to express ψ_i as a linear combination of atomic orbitals (LCAO), namely

$$\psi_i = \sum_{\mu=1}^{K} C_{\mu i}\phi_\mu, \tag{2.3}$$

is commonly adopted. ϕ_μ are called basis functions that are normally centered at the nuclei of a molecule. The coefficients $C_{\mu i}$ can be varied such that the total energy is minimized. Since Equation (2.3) is a finite expansion in practice, the integro-differential HF equations can be rewritten in matrix form (the Roothaan-Hall equations):

$$\mathbf{FC} = \mathbf{SCE} \tag{2.4}$$

where $\mathbf{F}$ is the matrix representation of the Fock operator in the basis of $\{\phi_\mu : \mu = 1, 2, \ldots, K\}$ and each entry of $\mathbf{S}$ describes the overlap between two atomic orbitals (AOs),

$$S_{\mu\nu} = \int \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)\, \mathrm{d}\mathbf{r}_1. \tag{2.5}$$

The vector $\mathbf{r}_1$ represents the spatial coordinates of an electron. The basis set may be composed of Slater-type or Gaussian-type functions at each atom. The basis functions are orthogonal only in a local environment, i.e. $S_{\mu\nu} = 0$ if both ϕ_μ and ϕ_ν are centered on the same atom, but the overlap of AOs belonging to different atoms in a molecule may not vanish. $\mathbf{S}$ is thus not an identity matrix. Therefore Equation (2.4) is a pseudo-eigenvalue problem and the eigenvalues and eigenvectors are the MO energies and coefficients.

The Fock matrix elements $F_{\mu\nu}$ for a closed-shell molecule of $2n$ electrons are listed below in atomic units.

$$F_{\mu\nu} = H^{\text{core}}_{\mu\nu} + G_{\mu\nu} \tag{2.6}$$

$$H^{\text{core}}_{\mu\nu} = \int \phi_\mu(\mathbf{r}_1)[\mathscr{T}(\mathbf{r}_1) + \mathscr{V}(\mathbf{r}_1)]\phi_\nu(\mathbf{r}_1)\, \mathrm{d}\mathbf{r}_1 \tag{2.7}$$

$$G_{\mu\nu} = \sum_{\lambda,\sigma} P_{\lambda\sigma}[(\mu\nu|\lambda\sigma) - \frac{1}{2}(\mu\lambda|\nu\sigma)] \tag{2.8}$$

$$P_{\mu\nu} = 2\sum_{i}^{n} C_{\mu i}C_{\nu i} \tag{2.9}$$

$$(\mu\nu|\lambda\sigma) = \iint \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}\phi_\lambda(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2)\, \mathrm{d}\mathbf{r}_1\, \mathrm{d}\mathbf{r}_2 \tag{2.10}$$

$F_{\mu\nu}$ is the sum of $H^{\text{core}}_{\mu\nu}$ and $G_{\mu\nu}$. $H^{\text{core}}_{\mu\nu}$ is the one-electron core-Hamiltonian element that includes the kinetic energy ($\mathscr{T}$) of an electron and the attraction ($\mathscr{V}$) between the electron and the nuclei. Once a basis set $\{\phi_\mu\}$ is specified, the core-Hamiltonian matrix will not change in a calculation.

$G_{\mu\nu}$ is the two-electron part of the Fock matrix. It features two characteristics. First, it is not fixed in a calculation. The construction of $G_{\mu\nu}$ depends on the density matrix $\mathbf{P}$, which itself depends on the doubly occupied MOs (see Equation (2.9)). Because these orbitals are the solution to Equation (2.4), an initial guess must be provided for the density matrix. Thereby, the Roothaan-Hall equations have to be tackled by a self-consistent-field (SCF) procedure, during which $\mathbf{P}$ is iteratively refined until certain convergence criteria are satisfied. Second, the computation of $G_{\mu\nu}$ requires the evaluation of a huge number of two-electron integrals (see Equation (2.10)). Since every $(\mu\nu|\lambda\sigma)$ involves four AOs, their number scales in proportion to the fourth power of the number of basis functions. For example, the number of two-electron integrals for a calculation with 1000 basis functions is one trillion (1×10^{12}) if orbital and integral symmetry are completely discarded. The computer memory required is 8 terabytes, i.e. 8×10^{12} bytes, for double precision floating-point numbers. Even today (in 2013) this kind of calculation is not feasible on most current computers. Direct SCF techniques, in which the two-electron integrals necessary for the Fock matrix update are recomputed on the fly in every SCF iteration, were invented to surmount this difficulty [92]. While this circumvents the storage problems, the intrinsic number of floating-point operations for integral evaluation is actually increased in accurate direct SCF calculations, which in essence trades floating-point operations for memory space. In practice, the computational costs are reduced by applying screening techniques during on-the-fly integral evaluation in direct SCF methods, which makes it possible to compute molecules with up to several hundred atoms by using *ab initio* HF methods or density functional theory (DFT).

2.2 Neglect of differential overlap

To alleviate the burden of two-electron integral evaluation, the zero-differential overlap (ZDO) approximation,

$$\varphi^{\mathrm{A}}(\mathbf{r}_1)\varphi^{\mathrm{B}}(\mathbf{r}_1)\,\mathrm{d}\mathbf{r}_1 = 0, \tag{2.11}$$

was introduced in the early days of quantum chemistry. It was first used in the PPP (Pariser-Parr-Pople) method [5, 6, 93] for the π-electrons of planar conjugated organic molecules, where φ^{A} and φ^{B} denote the p_π orbitals on atoms A and B, respectively.

An effective valence-shell Hamiltonian for both σ- and π-electrons [94] is employed in most of the semiempirical methods today. The most sophisticated neglect of differential overlap (NDO) approximation is assumed by the NDDO (Neglect of Diatomic Differential Overlap) model [10],

$$\phi_\mu^{\mathrm{A}}(\mathbf{r}_1)\phi_\nu^{\mathrm{B}}(\mathbf{r}_1)\,\mathrm{d}\mathbf{r}_1 = \delta_{\mathrm{AB}}\phi_\mu^{\mathrm{A}}(\mathbf{r}_1)\phi_\nu^{\mathrm{B}}(\mathbf{r}_1)\,\mathrm{d}\mathbf{r}_1, \tag{2.12}$$

where δ_{AB} is the Kronecker delta, i.e. $\delta_{\mathrm{AB}} = 1$ if A and B are the same atom, otherwise $\delta_{\mathrm{AB}} = 0$, and ϕ_μ^{A} and ϕ_ν^{B} are arbitrary AOs.

There are two consequences of the NDO approximation. First, the overlap matrix $\mathbf{S}$ in the Roothaan-Hall equations becomes an identity matrix. The pseudo-eigenvalue problem of Equation (2.4) is thus simplified to a standard eigenvalue problem

$$\mathbf{FC} = \mathbf{CE}. \tag{2.13}$$

These NDDO secular equations are easier and faster to solve than Equation (2.4) from a computational point of view, since one additional diagonalization of $\mathbf{S}$ and several extra matrix multiplications, which are necessary in the full Roothaan-Hall treatment, are bypassed in Equation (2.13).

Second, the NDDO model eliminates a substantial number of two-electron integrals and the only leftovers are the one- and two-center electron repulsion integrals,

$$(\mu^{\mathrm{A}}\nu^{\mathrm{B}}|\lambda^{\mathrm{C}}\sigma^{\mathrm{D}}) = \delta_{\mathrm{AB}}\delta_{\mathrm{CD}}(\mu^{\mathrm{A}}\nu^{\mathrm{A}}|\lambda^{\mathrm{C}}\sigma^{\mathrm{C}}). \tag{2.14}$$

Thus the number of two-electron integrals in semiempirical quantum chemical methods formally scales as $O(K^2)$ instead of $O(K^4)$ in *ab initio* HF (or DFT) methods. Equation (2.14) is the major source for the reduction in floating-point arithmetic operations and memory usage. Hence, in contrast to large-scale *ab initio* HF and DFT calculations that make use of the direct SCF technique, it is possible in semiempirical methods to compute the relevant integrals only once and to store them in fast memory for the subsequent Fock matrix formation.

The NDDO integral approximation (in Equation (2.14)) is seemingly rather drastic at first sight. It can be rationalized by considering the transformation of the non-orthogonal AOs $\{\phi_\mu\}$ into the symmetrically orthogonalized AOs (or Löwdin orbitals) $\{^\lambda\phi_\nu\}$:

$$^\lambda\phi_\nu = \sum_{\mu=1}^{K} T_{\mu\nu}\phi_\mu \qquad (\mathbf{T} = \mathbf{S}^{-1/2}) \tag{2.15}$$

where the transformation matrix $\mathbf{T}$ is the inverse square root of the overlap matrix in the basis of $\{\phi_\mu\}$. Since the Löwdin orbitals are orthonormal, it is straightforward to conceive $\mathbf{S}$ as an identity matrix and to retrieve the NDDO secular equations (2.13). Even though the orthogonalized basis set $\{^\lambda\phi_\nu\}$ consists of linear combinations of the original AOs, the Löwdin orbitals are deliberately optimized to minimize the overall differences between $\{\psi_\mu\}$ and $\{^\lambda\psi_\mu\}$ so that every symmetrically orthogonalized AO is dominated by the corresponding localized AO. The NDDO model for the two-electron integral evaluation can be motivated in this way, and supported by numerical computations as well [12–16, 95]. For example, the neglected multi-center and certain two-center integrals are very small in the basis of the Löwdin orbitals. The remaining two-electron repulsion integrals do not undergo serious variations if the basis functions are changed from $\{\psi_\mu\}$ to $\{^\lambda\psi_\mu\}$.

Finally, it must be stressed that the one-electron part of the Fock matrix, $H^{\mathrm{core}}_{\mu\nu}$ in Equation (2.7), accounts for the electron-core attraction and represents the major cause of covalent bonding. If Equation (2.12) were applied to the integrals in $H^{\mathrm{core}}_{\mu\nu}$, the whole molecule would be unbound. Therefore $H^{\mathrm{core}}_{\mu\nu}$ in the NDDO model is not subject to the NDO approximation.

2.3 MNDO method

The MNDO (Modified Neglect of Diatomic Overlap) method [19] is based on the NDDO model. The Fock matrix elements are defined as

$$\begin{aligned} F_{\mu\nu} = H^{\mathrm{core}}_{\mu\nu} &+ \sum_{\lambda,\sigma}^{\mathrm{A}} P_{\lambda\sigma}(G^{\mathrm{A}}_{\mu\nu,\lambda\sigma} - \frac{1}{2}G^{\mathrm{A}}_{\mu\lambda,\nu\sigma}) \\ &+ \sum_{\lambda,\sigma}^{\mathrm{B}} P_{\lambda\sigma}G^{\mathrm{AB}}_{\mu\nu,\lambda\sigma} \qquad (\mu \in \mathrm{A} \text{ and } \nu \in \mathrm{A}) \end{aligned} \tag{2.16}$$

$$F_{\mu\nu} = H_{\mu\nu}^{\text{core}} - \frac{1}{2}\sum_{\sigma}^{\text{B}}\sum_{\lambda}^{\text{A}} P_{\lambda\sigma} G_{\mu\lambda,\nu\sigma}^{\text{AB}} \qquad (\mu \in \text{A and } \nu \in \text{B}), \tag{2.17}$$

where G^{A} and G^{AB} are the one- and two-center two-electron repulsion integrals, respectively (see Equation (2.18)).

$$G_{\mu\nu,\lambda\sigma}^{\text{A}} = (\mu^{\text{A}}\nu^{\text{A}}|\lambda^{\text{A}}\sigma^{\text{A}}), \tag{2.18a}$$

$$G_{\mu\nu,\lambda\sigma}^{\text{AB}} = (\mu^{\text{A}}\nu^{\text{A}}|\lambda^{\text{B}}\sigma^{\text{B}}). \tag{2.18b}$$

The values of G^{A} adopted in the MNDO method are derived from experimental data. In later general-purpose parameterizations of the MNDO model, they have been freely optimized, for example in PM3 [22]. In the current thesis, this has also been done in specific parameterizations for an enzyme-catalyzed reaction and for hydrogen bonding in water (see Appendices C and D).

A more challenging task is to devise an appropriate treatment of the two-center two-electron integrals $G_{\mu\nu,\lambda\sigma}^{\text{AB}}$. These integrals were computed analytically in terms of Slater-type orbitals in early attempts to parameterize the NDDO model [96–98]. But the results were not satisfactory. One of the distinctions between MNDO and its predecessors lies in the evaluation of the two-center two-electron integrals. In the MNDO method, they are expanded as a series of multipole-multipole interactions

$$G_{\mu\nu,\lambda\sigma}^{\text{AB}} = \sum_{l_1}\sum_{l_2}\sum_{m}[M_{\text{A}}^{l_1 m}, M_{\text{B}}^{l_2 m}], \tag{2.19}$$

where the superscripts l and m are the order and the orientation of the multipoles, respectively. The multipoles of two interacting charge distributions are modeled by suitable configurations of point charges. The integrals evaluated in this manner exhibit the correct asymptotic behavior for large interatomic separations, i.e. they converge to the classical point charge interactions. If atoms A and B are superposed, the G^{AB} integrals are designed to reproduce the numeric values of the one-center integrals. For intermediate distances, the semiempirical G^{AB} integrals are somewhat smaller than their analytic *ab initio* counterparts, which is considered to account (to some extent) for dynamic electron correlation effects in an average manner.

The total energy of a molecule is the sum of the electronic energy and the core-core interactions. The latter consist of electrostatic terms and additional core-repulsion functions (CRFs). In the framework of the MNDO model, there have been a number of developments that primarily focused on refinements of the empirical CRFs, with the hope that some limitations of the original MNDO method could be (at least partially) remedied [21, 22, 99–104]. AM1 [21] and PM3 [22] are the most popular variants of this kind. The accuracy limits of the MNDO model were explored by using ever more sophisticated CRFs in a sequence of reparameterizations in the PMx ($x = 3$, 5, 6, and 7) methods [22, 77–79]. The results of these efforts indicate that the accuracy of the MNDO-type methods cannot be improved dramatically without advances in the underlying theoretical model. Hence it seems crucial for further progress to go beyond the MNDO model.

2.4 OMx methods

The *ab initio* HF method makes use of localized non-orthogonal basis functions $\{\phi_\mu\}$ in the Roothaan-Hall equations (see Equation (2.4)). The transformations to the Löwdin orbitals $\{^\lambda\phi_\mu\}$ and to the orthogonalized Fock matrix $^\lambda\mathbf{F}$ are performed explicitly by

$$^\lambda\phi = \mathbf{S}^{1/2}\phi \tag{2.20}$$

$$^\lambda\mathbf{F} = \mathbf{S}^{-1/2}\mathbf{F}\mathbf{S}^{-1/2}. \tag{2.21}$$

Because of the aforementioned NDO approximation (see Equation (2.12)), the basis functions in the MNDO method are presumed to be the orthogonalized AOs. Therefore, the semiempirical NDDO Fock matrix $^{\mathrm{NDDO}}\mathbf{F}$ is supposed to mimic the transformed Fock matrix $^\lambda\mathbf{F}$ in the *ab initio* HF method. The MNDO-type methods try to recover the effects of these transformations indirectly through flexible CRFs and continual reparameterizations. The OMx (Orthogonalization Method, x = 1, 2, and 3) methods, on the other hand, strive to incorporate the orthogonalization effects into the semiempirical Fock matrix via explicit corrections guided by the *ab initio* formulas and numerical results [30–32]. As already discussed, the two-electron integral approximations in the NDDO model can be motivated in terms of the Löwdin orbitals [12–16]. Hence, the semiempirical two-electron integrals implicitly incorporate the orthogonalization effects to some degree. By contrast, the one-electron core Hamiltonian is significantly affected by changes of the basis from $\{\phi\}$ to $\{^\lambda\phi\}$, i.e. $\mathbf{H}^{\mathrm{core}}$ and $^\lambda\mathbf{H}^{\mathrm{core}}$ can be quite different. Thus it is important to capture this effect for the semiempirical core Hamiltonian at the electronic level by going beyond the classic MNDO method [19].

The semiempirical orthogonalization correction to the core Hamiltonian matrix elements in the OMx methods is given by

$$\begin{aligned} V_{\mu\nu}^{\mathrm{ORT}} = &-\frac{1}{2}A_{\mu\nu}\sum_{\lambda}^{\mathrm{C}}(S_{\mu\lambda}\beta_{\lambda\nu} + \beta_{\mu\lambda}S_{\lambda\nu}) \\ &+\frac{1}{8}B_{\mu\nu}\sum_{\lambda}^{\mathrm{C}}S_{\mu\lambda}S_{\lambda\nu}(H_{\mu\mu}^{\mathrm{core}} + H_{\nu\nu}^{\mathrm{core}} - 2H_{\lambda\lambda}^{\mathrm{core}}), \end{aligned} \tag{2.22}$$

where $\beta_{\mu\lambda}$ are the two-center one-electron resonance integrals (ϕ_μ and ϕ_λ are located on different atoms) and $A_{\mu\nu}$ and $B_{\mu\nu}$ are adjustable parameters. $V_{\mu\nu}^{\mathrm{ORT}}$ in Equation (2.22) is a generic formula. The first OMx method, OM1, only contains the corrections for the one-center core Hamiltonian, i.e. both ϕ_μ and ϕ_ν are on the same atom [30]. Corrections to all elements of the core Hamiltonian are included in the OM2 and OM3 methods [31,32]. For computational efficiency, $V_{\mu\nu}^{\mathrm{ORT}}$ in OM3 is truncated at the first-order expansion level, i.e. $B_{\mu\nu}$ is set to zero, which causes only a very minor loss in accuracy.

Benchmark calculations [33–35] have demonstrated that the OMx methods can slightly improve the calculated ground-state properties, e.g. heat of formations and molecular geometries, compared with MNDO-type methods, whereas more significant enhancements are observed in several other areas, including vertical excitation energies, rotational barriers, and hydrogen bonding.

2.5 Dispersion corrections

Noncovalent forces, such as hydrogen bonds and dispersion interactions that are much weaker than conventional covalent bonds, are very important in biomolecules [105]. They are essential for driving the spontaneous folding of proteins and nucleic acids [106]. The MNDO method [19] tends to overestimate the repulsion between non-bonded atoms and can thus not properly predict the energy and structure of a system with dominant noncovalent interactions. A commonly adopted remedy of this problem is to use empirical CRFs to improve calculated energies and optimized geometries. A number of CRFs has been proposed for various MNDO-type methods in the past [21,22,99–104], particularly in order to improve the description of hydrogen bonding. Dispersion is a rather weak attractive interaction that acts between any pair of atoms (mainly within distances around the sum of their van der Waals radii) due to instantaneous electron fluctuations [107–109]. Formally, the dispersion interaction is not included in any of the current standard semiempirical methods [1], and the long-range dispersion attractions are not taken into account at the HF and DFT levels of theory either [110–114]. On the other hand, this kind of interaction is often well modeled by molecular mechanics force fields, e.g. AMBER [115] and CHARMM [116], with an empirical Lennard-Jones potential [117]. Thus, one can model dispersion by an additional energy correction term

$$E_{\mathrm{disp}} = -s_6 \sum \sum_{\mathrm{B>A}} f_{\mathrm{d}} \frac{C_{\mathrm{AB}}}{R_{\mathrm{AB}}^6}, \tag{2.23}$$

which is proportional to the sixth power of the inverse distance between atoms A and B. Such corrections have been incorporated into the self-consistent-charge density-functional tight-binding (SCC-DFTB) method [118], the HF [119–121], a variety of DFT [122–127], and some semiempirical methods [128–133] to better represent the van der Waals interaction in weakly bound complexes of molecules.

Chapter 3

GPU-Accelerated Semiempirical Quantum Chemical Calculations

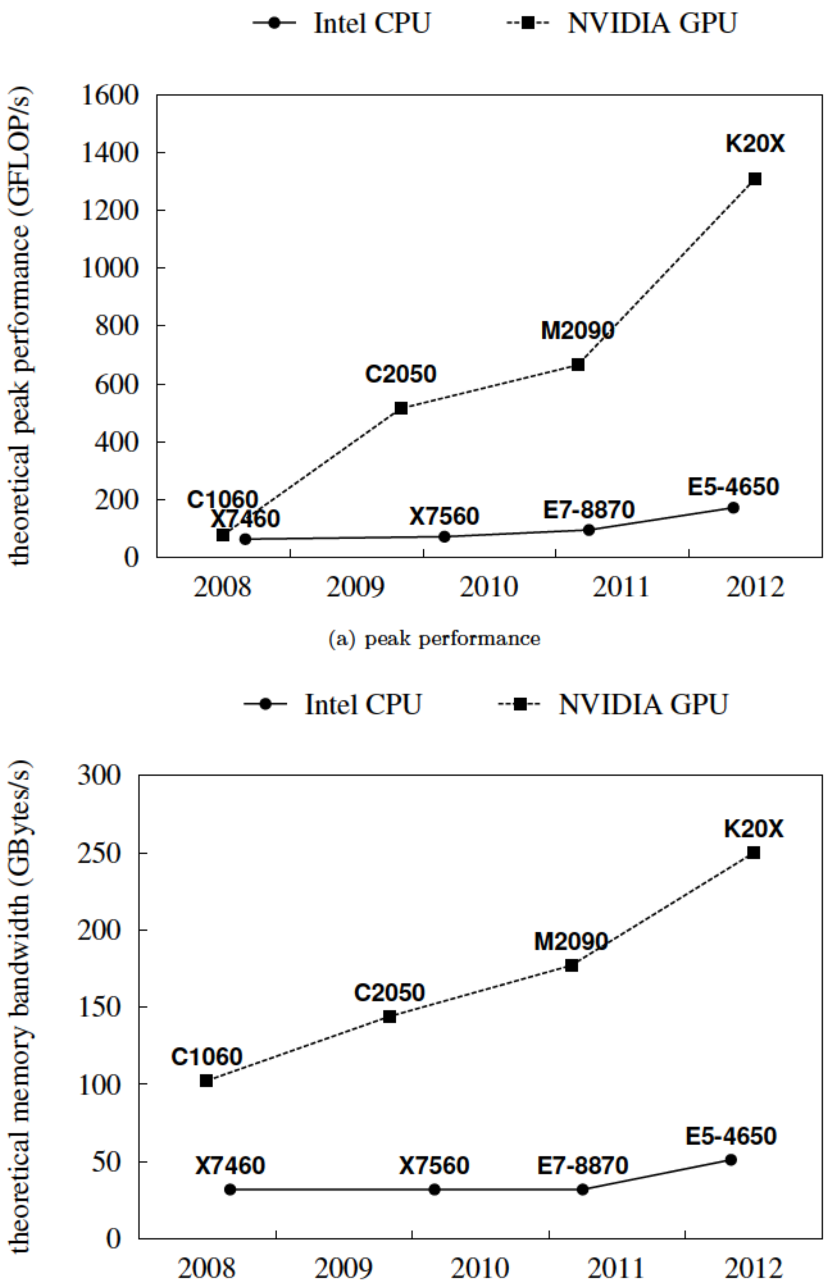
3.1 Introduction

The graphics processing unit (GPU) has evolved into a many-core processor with massive parallelism for tremendously intensive computations over the last few years [46, 47]. Both the peak arithmetic performance and memory bandwidth of a GPU have significantly exceeded those of a multi-core central processing unit (CPU) as illustrated in Figure 3.1. Thus it becomes appealing to develop programs in computational chemistry that can harness the GPU as a powerful computing device [48, 49, 61].

The reason for the advanced performance of a GPU lies in its distinct architectural design. The CPU and GPU dies are schematically represented in Figure 3.2. Since a CPU is a processor for general-purpose computing, many transistors are needed as control units for sophisticated workflow. In contrast, a GPU can be regarded as a specific co-processor. It devotes more transistors to arithmetic and logic units for highly parallel floating-point computations. Moreover, due to the von Neumann bottleneck [134], more area on a CPU die is used for caches to facilitate data access to main memory. On the other hand, a GPU is optimized for the execution of tens of thousands of parallel threads that can efficiently hide memory latency. Consequently, the architecture of a GPU has progressed in a very different direction than that for a CPU.

Even though semiempirical quantum chemical calculations are about three orders of magnitude faster than *ab initio* Hartree-Fock and DFT calculations [26], they can be rather time-consuming when it comes to large-scale applications, e.g. for proteins and long-time dynamics simulations. Therefore the best strategy is to deploy a heterogeneous CPU-GPU computer and move the most demanding parts of semiempirical calculations to the GPUs.

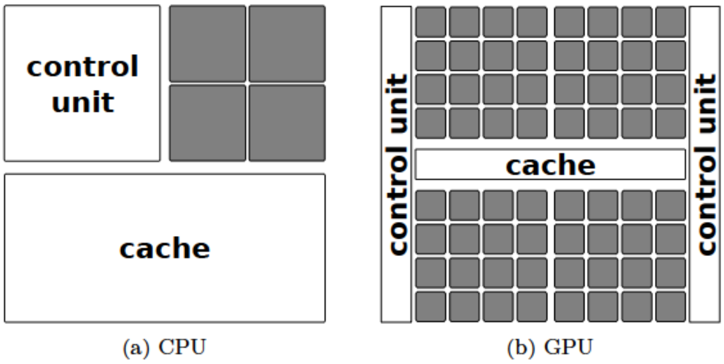
One major project in the dissertation was to port the MNDO99 code [84] to a hybrid CPU-GPU platform and to optimize the performance of GPU-accelerated semiempirical quantum chemical calculations. This has been accomplished both



(a) peak performance

(b) memory bandwidth

Figure 3.1: Comparison of peak arithmetic performance and memory bandwidth of different Intel CPUs and NVIDIA GPUs.



(a) CPU (b) GPU

Figure 3.2: Schematic diagrams of the CPU and GPU dies. The processing cores are depicted by gray squares.

for MNDO-type and OMx methods, with overall speedups of typically one order of magnitude for SCF calculations and geometry optimizations of large molecular systems. This work has been published [74], and copies of these articles are included in Appendices A and B. Rather than summarizing these published papers, the aim of this chapter is to document the technical implementation in greater detail so that the intricate GPU code can be more easily understood. In the following, we first describe the hardware specifications of the heterogeneous CPU-GPU computing node used for the development. Thereafter we present the dedicated optimizations on multiple GPUs for the fast pseudodiagonalization and the orthogonalization corrections.

3.2 Heterogeneous CPU-GPU computer

The development was conducted on a CVS version of the MNDO99 package [84]. The GPU acceleration is exclusively based on the CUDA (Compute Unified Device Architecture) framework that allows developers to program the NVIDIA GPU as a discrete computational device in terms of a C-like language [135].

Figure 3.3 presents a diagram for the heterogeneous CPU-GPU computer used in the development. It consists of two parts, host and device, which are connected by a 16-lane PCIE (Peripheral Component Interconnect Express) interface with a theoretical bandwidth of 8.0 GB/s. The host has two Intel Xeon X5690 CPUs which are linked via QPI (Intel QuickPath Interconnect). To each CPU a part of the system memory is connected. Using the QPI link the memory is shared among all CPUs. Because of the distiction between locally connected and remote memory this is called a NUMA (Non-Uniform Memory Access) architecture. The device is composed of two NVIDIA Tesla M2090 GPUs with disjoint DRAM (Dynamic Random-Access Memory). The multi-GPU part may also be regarded to have a kind of NUMA architecture as each GPU may access the memory connected to the other GPU (peer-to-peer direct access). Data transfer between both GPUs (and between host and device) takes place over the PCIE links. PCIE bandwidth represents a significant constraint for the transfer rate. Thus it is crucial to avoid the peer-to-peer data access for the GPUs in the code optimization.

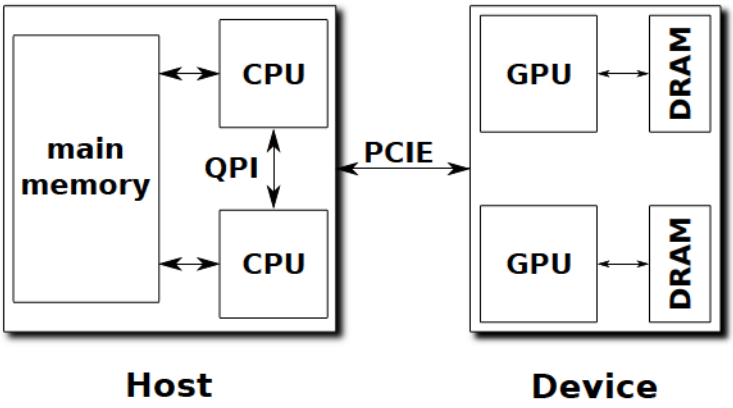


Figure 3.3: A schematic representation of the heterogeneous CPU-GPU computer.

Table 3.1: Some hardware specifications of the Intel Xeon X5690 CPU with the Turbo Boost Technology switched off and the NVIDIA Tesla M2090 GPU with ECC (error-correcting code) memory.

	CPU	GPU
clock frequency (GHz)	3.46	1.3
number of cores	6	–
number of CUDA cores	–	512
double precision performance (GFLOP/s)	83.0	665.6
thermal design power (W)	130	250
performance per watt	0.6	2.7
maximum memory size (GB)	288	5.25
memory bandwidth (GB/s)	32	155
cache size (MB)	12	0.768

The hardware architectures of the Intel Xeon X5690 CPU and the NVIDIA Tesla M2090 GPU are quite different (see Table 3.1). Each CPU core is much more powerful than a CUDA core of the GPU. The clock frequency alone is 3.46 GHz versus 1.3 GHz, and a CPU core is a lot more complex. The GPU, however, possesses many more processing cores than the CPU, i.e. 512 versus 6. Therefore the Tesla M2090 has a much higher double-precision floating-point performance than the Xeon X5690. The energy consumed by the GPU is roughly twice as high as that of the CPU. However, Tesla M2090 is more energy efficient than Xeon X5690 because the former can deliver 8 times the arithmetic performance of the latter.

The GPU memory size is the primary restriction for a large-scale calculation. Our current workaround is to move the required data arrays back and forth on-the-fly for the bottleneck computations. This approach minimizes the usage of GPU memory, whereas the host-device data transfer represents a compromise. Typically, there are 30 iterations to achieve a self-consistent solution to the secular equations in a semiempirical Hartree-Fock calculation. In addition, only a few data transfers must take place for each iteration. Thus it seems reasonable to make such a trade-off for the GPU memory.

Each GPU can normally maintain a higher speed for accessing the data in its own DRAM. Moreover, the GPU is designed to execute computing threads with massive parallelization to hide memory latency. Hence a cache was not present in earlier generations of the NVIDIA GPU until the recent Fermi architecture. In contrast to the cache size on a CPU, the GPU cache is fairly small, e.g. 12 MB versus 768 KB. Nonetheless, some algorithms that cannot make use of the GPU shared memory may benefit from the GPU cache to some extent.

3.3 Fast pseudodiagonalization

The fast pseudodiagonalization has been described in detail in the original paper [136]. Because the optimized GPU implementation can be quite complicated, its mathematical formulas are briefly repeated here.

Basically the fast pseudodiagonalization entails two operations. First, the Fock matrix in the basis of the atomic orbitals (AOs) is transformed to molecular orbitals (MOs) by means of a triple matrix multiplication

$$\mathscr{F} = \mathbf{C}_o^\top \mathbf{F} \mathbf{C}_v, \tag{3.1}$$

where $\mathbf{C}_o$ and $\mathbf{C}_v$ are the matrices of the occupied and virtual MO vectors. The second step begins with the evaluation of the rotation coefficients, c and s,

$$u = \left(\frac{\mathscr{F}_{ia}}{\epsilon_a - \epsilon_i}\right)^2 \tag{3.2}$$

$$c = 1 - \frac{u}{2} \tag{3.3}$$

$$s = -\operatorname{sgn}(\mathscr{F}_{ia})\sqrt{u - \frac{u^2}{4}}, \tag{3.4}$$

where i and a are the indices for the occupied and virtual MOs, respectively. Noniterative Jacobi-like 2 × 2 rotations are then applied for appropriate pairs of occupied ($\mathbf{c}_o$) and virtual ($\mathbf{c}_v$) vectors

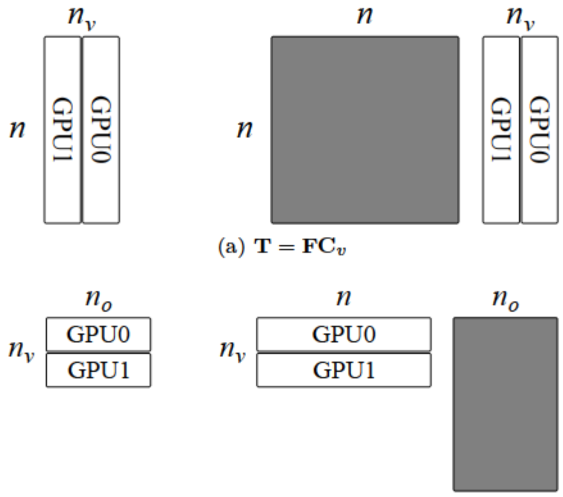
$$\mathbf{c}_o' = c\mathbf{c}_o - s\mathbf{c}_v \tag{3.5}$$

$$\mathbf{c}_v' = s\mathbf{c}_o + c\mathbf{c}_v. \tag{3.6}$$

$\mathbf{c}_o'$ and $\mathbf{c}_v'$ are the new MOs. The values of $\mathscr{F}_{ia}$ and $\epsilon_a - \epsilon_i$ are used to determine whether the associated rotation will be done or not, so that only a sparse set of orbitals is rotated in practice.

In the following, we begin with the description for the multi-GPU implementation of the triple matrix multiplication (see Equation (3.1)). Then the memory management and data storage for the subsequent Jacobi-like rotations on one single GPU device are demonstrated using an example of 8 basis functions and 5 doubly occupied MOs. A description of how to utilize multiple GPU devices for performing the rotations is given at the end of this section.

Let us consider a semiempirical calculation with n basis functions, with the number of occupied and virtual orbitals being n_o and n_v, respectively. In



(a) $\mathbf{T} = \mathbf{F}\mathbf{C}_v$

(b) $\mathscr{F}^\top = \mathbf{T}^\top \mathbf{C}_o$

Figure 3.4: Schematic illustrations for the implementation of $\mathscr{F} = \mathbf{C}_o^\top \mathbf{F}\mathbf{C}_v$ on two GPUs (GPU0 and GPU1). The arrays that reside exclusively in the memory of one GPU are marked by GPU0 (or GPU1), whereas the arrays in the memory of both GPUs are colored in gray. $\mathbf{T}$ is a temporary matrix.

addition, we assume that n_o is greater than or equal to n_v, which will often be the case in semiempirical calculations. Then it can be readily proved that the number of floating-point operations for computing $\mathscr{F}$ is minimized if the associativity of $\mathbf{C}_o^\top(\mathbf{F}\mathbf{C}_v)$ is exploited in Equation (3.1). Therefore, the implementation performs the matrix multiplications $\mathbf{T} = \mathbf{F}\mathbf{C}_v$ and $\mathscr{F}^\top = \mathbf{T}^\top \mathbf{C}_o$, in this order (see Figure 3.4), where $\mathbf{T}$ is a temporary array. Because $\mathscr{F}$ is symmetric it does not matter that these two multiplications result in the transposed matrix.

The basic principle of the multi-GPU implementation is to partition the relevant matrices into smaller tiles for each GPU. Initially, both $\mathbf{F}$ and $\mathbf{C}$ reside in the host main memory. For the first matrix multiplication, $\mathbf{C}_v$ is evenly divided and the resulting smaller arrays are transferred to the corresponding GPU memory. Meanwhile the complete Fock matrix (the gray square in Figure 3.4(a)) is moved to both GPUs. Thus the two independent computations may run on two GPUs at the same time. The second matrix multiplication can be executed in a similar manner, provided the matrix $\mathbf{C}_o$ resides in both GPU memories (see Figure 3.4(b)). It should be noted that the two matrix multiplications can run asynchronously on multi-GPU devices, i.e. the calculation on one GPU does not need to wait for the others for finishing the multiplications at any stage. It is enough to have a final barrier that transfers the related tiles of $\mathscr{F}$ back to the main memory on the host. Our numerical tests have shown that the triple matrix multiplications on two Tesla M2090 GPUs are ~ 1.6 times faster than those on one single GPU.

The GPU-dedicated algorithm of the noniterative Jacobi-like rotations has been described in the published paper [74] (see Appendix A). The internal memory management is documented here by using an example calculation with 8 basis functions, in which 5 occupied orbitals and 3 virtual orbitals are presumed.

All the Jacobi-like rotations are depicted in Figure 3.5 for the example

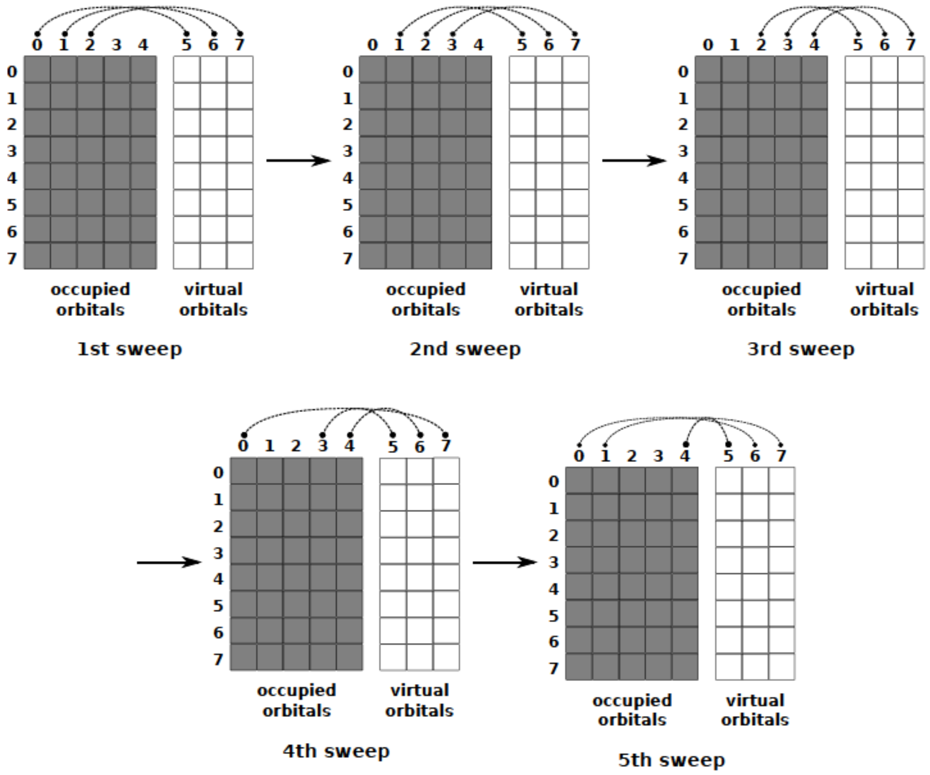
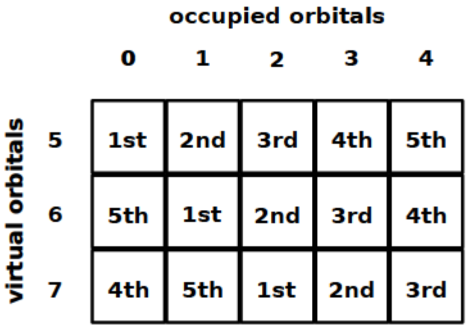


Figure 3.5: All noniterative Jacobi-like 2 × 2 rotations for an example calculation with 8 basis functions. The 5 occupied orbitals and 3 virtual orbitals are represented by gray and white squares, respectively. Rotations in a sweep are indicated by dashed lines. The thresholds for skipping small rotations are not considered.

calculation. There are a total of n_o sweeps and n_v rotations for each sweep (if all rotations are performed disregarding the thresholds for skipping small rotations). Moreover, the rotations belonging to the same sweep can be executed out-of-order, but the sweeps must be performed in a sequential manner. Thus a local data synchronization is imposed for every block of GPU threads prior to an execution of the next sweep.

All rotations and the related pairs of occupied and virtual orbitals are summarized in Figure 3.6. There is a one-to-one mapping between the rotations and the matrix entries $\mathscr{F}_{ia}$. Because certain thresholds on the Fock matrix elements for skipping small rotations are always used in a calculation, not all pairs of orbitals are actually rotated. The ratio between rotated and skipped pairs, however, can vary a lot from one SCF cycle to another, and thereby be unpredictable. Hence we may assume that two pairs of orbitals, $4 \leftrightarrow 5$ and $1 \leftrightarrow 7$, of the 5th sweep in the example fulfill the thresholds for rotation, whereas the $0 \leftrightarrow 6$ combination will not be rotated (see Figure 3.7).

Two supplementary integer arrays, `npair` and `imo`, are employed to store the counters for the Jacobi-like rotations. `npair` is one-dimensional. Its length equals the number of sweeps. Each entry of `npair` contains the number of



occupied orbitals

virtual orbitals	0	1	2	3	4
5	1st	2nd	3rd	4th	5th
6	5th	1st	2nd	3rd	4th
7	4th	5th	1st	2nd	3rd

Figure 3.6: Mapping of the sweeps (given by ordinal numbers in the boxes) to the orbital pairs and corresponding elements of the Fock matrix $\mathscr{F}$ stored in memory as a two-dimensional array.

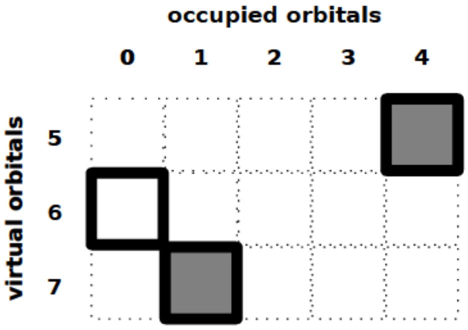


Figure 3.7: All the Jacobi-like rotations in the example calculation of 5 occupied and 3 virtual orbitals. The 5th sweep of rotations is highlighted by bold squares, whereas the others are represented by dashed squares. The rotations of the 5th sweep, which can satisfy the thresholds, are filled with gray.

orbital rotations to be performed in the corresponding sweep, e.g. 2 for the 5th sweep. `imo` is an $n_v \times n_o$ matrix. Its columns correspond to the sweeps. The occupied orbitals of the rotations for which the thresholds are fulfilled are recorded in the column vectors, whose actual lengths (without trailing unused elements) are given by `npair`. The corresponding virtual orbitals can be easily deduced from the occupied orbitals using an offset determined by the sweep (see Figure 3.6). The occupied orbitals in the sweep vectors are always definite, but their sequential order in a column is completely unspecified. For example, orbital 1 may be followed by orbital 4 for the 5th sweep (see Figure 3.8). But it does make sense to have a sequence of 1 coming after 4 for the same sweep in another independent run of the program. The reason for this uncertainty is that the GPU computing threads are executed completely out-of-order. Therefore atomic operations are applied to find the next unused element of the current column vector in matrix `imo` into which the index of the occupied orbital will be stored. Before storing any index into the current column of `imo`, the corresponding element of `npair` will be initialized with 0 indicating an empty column and pointing to the first element in the column (0-based index). A GPU thread that finds a non-negligible rotation will use an atomic operation to increment this pointer, i.e. reading the old value, adding 1 and storing the new value will not be interrupted by another thread. Using the pointer obtained this way the occupied orbital index can be safely stored into the column vector. The coefficients of the Jacobi-like rotations (see Equation (3.2)) are stored in a $(2n_v) \times n_o$ double precision matrix `a`. There is a direct mapping between the orbitals listed in `imo` and the coefficients given in `a`, i.e. the arrangement of the occupied orbitals for the 5th sweep in `imo` is consistent with the coefficients saved in `a` (see Figure 3.8).

The multi-GPU implementation of the noniterative Jacobi-like rotations is illustrated in Figure 3.9 on two GPU devices. The basic principle is rather simple: just partition the relevant arrays of the occupied and virtual orbitals on the GPUs as balanced as possible. Our test calculations demonstrate that the multi-GPU implementation for the rotations is very efficient, i.e. the speedup is 1.9 on 2 GPUs (see Appendix B). It should be stressed that the actual dimension of the GPU threads in a block, the mapping of the threads to the rotations, and the local synchronization within a block of threads are not shown in Figure 3.9 explicitly. These complicated technical details are beyond the scope of this thesis and can be checked in the code itself.

3.4 Orthogonalization corrections

Our benchmark calculations have shown that the orthogonalization corrections for the OM2 and OM3 methods may consume a very significant share of the total computation time, i.e. $\sim 10\%$ of the wall clock time in a single-core CPU-only calculation and $\sim 30\%$ for a GPU-accelerated calculation [74]. Instead of a straightforward port of the original CPU code onto the GPU devices, we devised a new algorithm for the orthogonalization corrections, which is entirely equivalent to the old one formally, but much more efficient for numerical computations on both CPU and GPU architectures. The new algorithm is presented first in this section. Thereafter it is tested through OM3 calculations on a set of proteins, showing that speedups of ~ 400 can be readily achieved by the implementation of the new algorithm on 2 GPU devices (compared with the original serial

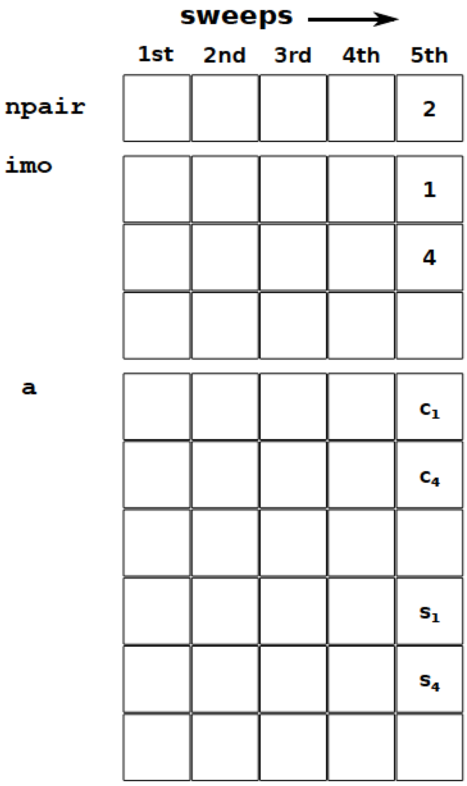


Figure 3.8: A diagram of the internal memory management for the GPU implementation of the noniterative Jacobi-like rotations. See the text for more details on the 5th sweep. **npair** and **imo** are integer arrays, whereas **a** is an array of double-precision floating-point numbers.

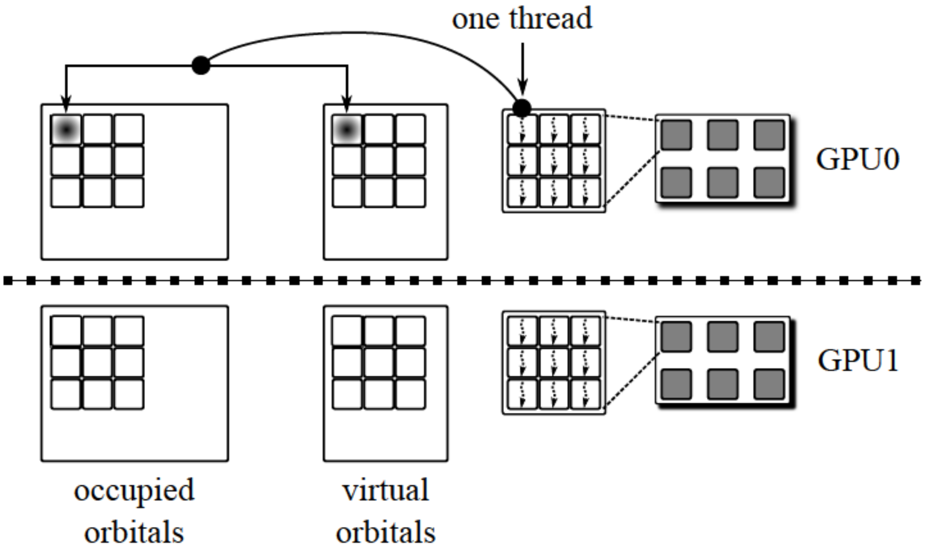


Figure 3.9: The multi-GPU implementation of the noniterative Jacobi-like rotations. The arrays of orbitals are divided between two GPU devices (GPU0 and GPU1). A single GPU thread, which computes one allowed rotation, is represented by a dashed arrow. The gray squares denote blocks of GPU threads. The grids of thread blocks are organized in shadow rectangles on the GPUs.

CPU-only implementation).

In the OM2 method, the orthogonalization corrections are in their most general form, and hence they are used for deriving the new algorithm. The general formula for the OM2 corrections is

$$\begin{aligned} V_{\mu\nu}^{\mathrm{OM2}} &= -\frac{1}{2}A_{\mu\nu}\sum_{\lambda}^{\mathrm{C}}(S_{\mu\lambda}\beta_{\lambda\nu}+\beta_{\mu\lambda}S_{\lambda\nu}) \\ &+\frac{1}{8}B_{\mu\nu}\sum_{\lambda}^{\mathrm{C}}S_{\mu\lambda}S_{\lambda\nu}(H_{\mu\mu}^{\mathrm{core}}+H_{\nu\nu}^{\mathrm{core}}-2H_{\lambda\lambda}^{\mathrm{core}}), \end{aligned} \tag{3.7}$$

where μ and ν are atomic orbitals on arbitrary atoms A and B, respectively. The summation in Equation (3.7) ranges over atomic orbitals λ at atoms C that differ from A and B. We now formally extend these summations to cover all atoms including A and B:

$$\begin{aligned} V_{\mu\nu}^{\mathrm{OM2}} &= -\frac{1}{2}A_{\mu\nu}\sum_{k}(S_{\mu k}\beta_{k\nu}+\beta_{\mu k}S_{k\nu}) \\ &+\frac{1}{8}B_{\mu\nu}\sum_{k}S_{\mu k}S_{k\nu}(H_{\mu\mu}^{\mathrm{core}}+H_{\nu\nu}^{\mathrm{core}}-2H_{kk}^{\mathrm{core}}) \\ &+\frac{1}{2}A_{\mu\nu}\sum_{\rho}^{\mathrm{A}}(S_{\mu\rho}\beta_{\rho\nu}+\beta_{\mu\rho}S_{\rho\nu}) \end{aligned}$$

$$
\begin{aligned}
&- \frac{1}{8} B_{\mu\nu} \sum_{\rho}^{\mathrm{A}} S_{\mu\rho} S_{\rho\nu} (H_{\mu\mu}^{\mathrm{core}} + H_{\nu\nu}^{\mathrm{core}} - 2H_{\rho\rho}^{\mathrm{core}}) \\
&+ \frac{1}{2} A_{\mu\nu} \sum_{\sigma}^{\mathrm{B}} (S_{\mu\sigma}\beta_{\sigma\nu} + \beta_{\mu\sigma} S_{\sigma\nu}) \\
&- \frac{1}{8} B_{\mu\nu} \sum_{\sigma}^{\mathrm{B}} S_{\mu\sigma} S_{\sigma\nu} (H_{\mu\mu}^{\mathrm{core}} + H_{\nu\nu}^{\mathrm{core}} - 2H_{\sigma\sigma}^{\mathrm{core}}), \qquad (3.8)
\end{aligned}
$$

where k runs over all atomic orbitals, and ρ and σ symbolize the orbitals on atoms A and B, respectively. The extra terms in the last four lines of Equation (3.8) exactly cancel the extra terms in the first two lines that arise from extending the sums over k to all orbitals. The extra terms in the last four lines are all zero, because the the basis functions on the same atom are orthonormal to each other

$$S_{\mu\nu} = \delta_{\mu\nu}, \qquad (3.9)$$

and the related resonance integrals are also zero by definition

$$\beta_{\mu\nu} = 0, \qquad (3.10)$$

Equation (3.8) can thus be simplified as

$$V_{\mu\nu}^{\mathrm{OM2}} = -\frac{1}{2} A_{\mu\nu} X_{\mu\nu} + \frac{1}{8} B_{\mu\nu} Y_{\mu\nu}, \qquad (3.11)$$

where $X_{\mu\nu}$ and $Y_{\mu\nu}$ are given below.

$$X_{\mu\nu} = \left[\sum_{k} S_{\mu k}\beta_{k\nu} - \beta_{\mu\nu} \right] + \left[\sum_{k} \beta_{\mu k} S_{k\nu} - \beta_{\mu\nu} \right] \qquad (3.12)$$

$$Y_{\mu\nu} = (H_{\mu\mu}^{\mathrm{core}} + H_{\nu\nu}^{\mathrm{core}}) \sum_{k} S_{\mu k} S_{k\nu} - 2 \sum_{k} S_{\mu k} H_{kk}^{\mathrm{core}} S_{k\nu} \qquad (3.13)$$

Equation (3.12) can be written as

$$\mathbf{X} = (\mathbf{S}\beta - \beta) + (\beta\mathbf{S} - \beta) \qquad (3.14)$$

in matrix form. If we define $\mathbf{Q} = \mathbf{S} - \mathbf{I}$, where $\mathbf{I}$ is an identity matrix, then Equation (3.14) becomes $\mathbf{X} = \mathbf{Q}\beta + \beta\mathbf{Q}$. Because $\mathbf{Q}$ and β are symmetric matrices, we have

$$\mathbf{X} = (\mathbf{Q}\beta) + (\mathbf{Q}\beta)^{\top}. \qquad (3.15)$$

If we further define $G_{\mu\nu} = -\frac{1}{2} A_{\mu\nu}$, the first term of Equation (3.7), which is in fact the orthogonalization correction in the OM3 method, is

$$V_{\mu\nu}^{\mathrm{OM3}} = G_{\mu\nu} X_{\mu\nu}. \qquad (3.16)$$

Equation (3.16) in matrix form is

$$\mathbf{V}^{\mathrm{OM3}} = \mathbf{G} \circ \mathbf{X}, \qquad (3.17)$$

where $\circ$ denotes the element-wise product.

Table 3.2: The proteins used for testing the orthogonalization corrections in the OM3 method. N_a and N_f are the numbers of atoms and basis functions, respectively.

notation	P_{020}	P_{063}	P_{086}	P_{100}	P_{125}	P_{156}	P_{166}	P_{221}
PDB ID	1BTQ	1K50	2HXX	3K6F	1ACF	2A4V	4A02	3AQO
N_a	307	1 097	1 495	1 842	2 004	2 969	3 415	3 558
N_f	754	2 699	3 655	4 446	4 920	7 157	8 173	8 727

Unfortunately, there is no similar matrix expression for Equation (3.13), since its second term, i.e. $\sum_k S_{\mu k} H_{kk}^{\text{core}} S_{k\nu}$, involves three matrix elements. Experience shows that the contribution of Equation (3.13) to the orthogonalization corrections in the OM2 method is normally rather small, mainly because the first and second sums in Equation (3.13) are of opposite sign and may thus partially compensate each other in numerical computations. The orthogonalization corrections in OM2 and OM3 are thus fairly similar, and benchmark calculations have indeed demonstrated that the accuracy of OM3 is closely comparable to that of the OM2 method [31, 32, 34, 35].

A new algorithm for computing the orthogonalization corrections in the OM3 method is outlined in Algorithm 3.1. The main computational effort is the matrix-matrix multiplication in line 6. In addition, as $\mathbf{Q}$, β, $\mathbf{G}$, $\mathbf{X}$, and $\mathbf{V}^{\text{OM3}}$ are symmetric, only the lower (or upper) triangular entries of the matrices are considered in practice (see lines 3, 7, and 8).

Algorithm 3.1 A matrix multiplication algorithm for computing the orthogonalization corrections in the OM3 method.

```
1: for $i = 1$ to $N_f$ do
2:     for $j = 1$ to $i$ do
3:         compute $Q_{j,i}$, $\beta_{j,i}$ and $G_{j,i}$.        ▷ All in linear arrays.
4:     end for
5: end for
6: $\mathbf{T} \leftarrow \mathbf{Q}\beta$                        ▷ $\mathbf{T}$ is a temporary workspace.
7: $\mathbf{X} \leftarrow \mathbf{T} + \mathbf{T}^\top$
8: $\mathbf{V}^{\text{OM3}} \leftarrow \mathbf{G} \circ \mathbf{X}$
```

This algorithm has been implemented in the MNDO99 package [84] for both multi-CPU and multi-GPU architectures. Its performance was compared with that of the original code by using a set of 8 proteins [137–144]. Table 3.2 lists these proteins that are labeled as P_x, where x is the number of residues. The notations for the computing configurations and the employed algorithms are collected in Table 3.3. The computation times and speedups are summarized in Tables 3.4 and 3.5, respectively.

The new algorithm offers significant advantages in both computational efficiency and numerical accuracy. The original code without any cutoff for the orthogonalization corrections to matrix elements involving distant atoms is comparatively slow on one single CPU core ($\mathrm{O}_{1\mathrm{C}}$). By contrast, the new algorithm is about 10 times faster when run on the same hardware ($\mathrm{N}_{1\mathrm{C}}$). Furthermore, the GPU-oriented implementation can amplify the performance gain to more

Table 3.3: Description of the notations for the different algorithms for the orthogonalization corrections in the OM3 method on a variety of computing platforms.

notation	description
O_{1C}	the original algorithm on a single CPU core
O_{6C}	the original algorithm on a single CPU using all 6 cores
O_{12C}	the original algorithm on two CPUs using all 12 cores
O'_{1C}	the original algorithm with the default cutoffs on a single CPU core
O'_{6C}	the original algorithm with the default cutoffs on a single CPU using all 6 cores
O'_{12C}	the original algorithm with the default cutoffs on two CPUs using all 12 cores
N_{1C}	the new algorithm on a single CPU core
N_{6C}	the new algorithm on a single CPU using all 6 cores
N_{12C}	the new algorithm on two CPUs using all 12 cores
N_{1G}	the new algorithm on a single GPU device
N_{2G}	the new algorithm on two GPU devices

Table 3.4: Computation time (in seconds) for the subroutine calculating the orthogonalization corrections in the OM3 method.

	P_{020}	P_{063}	P_{086}	P_{100}
O_{1C}	0.560	39.470	84.384	153.225
O_{6C}	0.100	5.969	15.711	28.686
O_{12C}	0.054	3.710	10.045	19.076
O'_{1C}	0.071	2.331	5.629	10.471
O'_{6C}	0.014	0.396	0.952	1.779
O'_{12C}	0.008	0.201	0.479	0.900
N_{1C}	0.075	3.583	8.147	14.170
N_{6C}	0.023	1.219	1.563	2.774
N_{12C}	0.020	1.113	1.766	1.701
N_{1G}	0.007	0.143	0.310	0.530
N_{2G}	0.008	0.119	0.247	0.396
	P_{125}	P_{156}	P_{166}	P_{221}
O_{1C}	243.246	639.672	968.720	1156.657
O_{6C}	38.403	124.616	194.525	233.753
O_{12C}	25.818	102.241	176.604	214.191
O'_{1C}	13.517	44.965	68.980	78.165
O'_{6C}	2.299	7.573	11.650	13.202
O'_{12C}	1.150	3.897	6.125	7.037
N_{1C}	18.499	56.778	84.360	130.444
N_{6C}	3.471	10.997	15.760	20.819
N_{12C}	2.655	6.223	8.415	11.501
N_{1G}	0.694	2.660	4.017	4.799
N_{2G}	0.509	1.601	2.437	2.730

Table 3.5: Speedups of the subroutine computing the orthogonalization corrections in the OM3 method. The wall clock times of O_{1C} are used as reference.

	P_{020}	P_{063}	P_{086}	P_{100}
O_{6C}	5.6	6.6	5.4	5.3
O_{12C}	10.4	10.6	8.4	8.0
O'_{1C}	7.9	16.9	15.0	14.6
O'_{6C}	40.0	99.7	88.6	86.1
O'_{12C}	70.0	196.4	176.2	170.2
N_{1C}	7.5	11.0	10.4	10.8
N_{6C}	24.3	32.4	54.0	55.2
N_{12C}	28.0	35.5	47.8	90.1
N_{1G}	80.0	276.0	272.2	289.1
N_{2G}	70.0	331.7	341.6	386.9
	P_{125}	P_{156}	P_{166}	P_{221}
O_{6C}	6.3	5.1	5.0	4.9
O_{12C}	9.4	6.3	5.5	5.4
O'_{1C}	18.0	14.2	14.0	14.8
O'_{6C}	105.8	84.5	83.2	87.6
O'_{12C}	211.5	164.1	158.2	164.4
N_{1C}	13.1	11.3	11.5	8.9
N_{6C}	70.1	58.2	61.5	55.6
N_{12C}	91.6	102.8	115.1	100.6
N_{1G}	350.5	240.5	241.2	241.0
N_{2G}	477.9	399.5	397.5	423.7

Table 3.6: The maximum numerical error (in eV) for the matrix elements of the orthogonalization corrections in the OM3 method. The reference values are those computed with the O_{1C} setup.

	P_{020}	P_{063}	P_{086}	P_{100}
O_{6C}	0.00	0.00	0.00	0.00
O_{12C}	0.00	0.00	0.00	0.00
O'_{1C}	0.92×10^{-4}	0.14×10^{-3}	0.11×10^{-3}	0.11×10^{-3}
O'_{6C}	0.92×10^{-4}	0.14×10^{-3}	0.11×10^{-3}	0.11×10^{-3}
O'_{12C}	0.92×10^{-4}	0.14×10^{-3}	0.11×10^{-3}	0.11×10^{-3}
N_{1C}	0.17×10^{-14}	0.18×10^{-14}	0.22×10^{-14}	0.22×10^{-14}
N_{6C}	0.17×10^{-14}	0.18×10^{-14}	0.22×10^{-14}	0.22×10^{-14}
N_{12C}	0.17×10^{-14}	0.18×10^{-14}	0.22×10^{-14}	0.22×10^{-14}
N_{1G}	0.18×10^{-14}	0.17×10^{-14}	0.22×10^{-14}	0.23×10^{-14}
N_{2G}	0.18×10^{-14}	0.17×10^{-14}	0.22×10^{-14}	0.23×10^{-14}

	P_{125}	P_{156}	P_{166}	P_{221}
O_{6C}	0.00	0.00	0.00	0.00
O_{12C}	0.00	0.00	0.00	0.00
O'_{1C}	0.12×10^{-3}	0.13×10^{-3}	0.12×10^{-3}	0.12×10^{-3}
O'_{6C}	0.12×10^{-3}	0.13×10^{-3}	0.12×10^{-3}	0.12×10^{-3}
O'_{12C}	0.12×10^{-3}	0.13×10^{-3}	0.12×10^{-3}	0.12×10^{-3}
N_{1C}	0.24×10^{-14}	0.22×10^{-14}	0.36×10^{-14}	0.17×10^{-14}
N_{6C}	0.24×10^{-14}	0.22×10^{-14}	0.36×10^{-14}	0.17×10^{-14}
N_{12C}	0.24×10^{-14}	0.22×10^{-14}	0.36×10^{-14}	0.17×10^{-14}
N_{1G}	0.26×10^{-14}	0.19×10^{-14}	0.43×10^{-14}	0.19×10^{-14}
N_{2G}	0.26×10^{-14}	0.19×10^{-14}	0.43×10^{-14}	0.19×10^{-14}

than 400 times on 2 GPU devices.

The original code provides the option to skip the evaluation of the exponentially decaying orthogonalization corrections if certain distance-based and/or overlap-based criteria are satisfied. When using the default cutoffs in the original code, the computations are significantly accelerated. For a large system, the speedup can be one order of magnitude on one CPU core and can reach 200 times with a total of 12 CPU cores. However, these default cutoffs cause numerical errors of around 0.1×10^{-3} eV in the matrix elements $V_{\mu\nu}^{\text{OM3}}$ (see Table 3.6). On the other hand, the new algorithm is able to reproduce reference values of $\mathbf{V}^{\text{OM3}}$ from O_{1C} calculations with very high accuracy, i.e. the errors are always less than 0.5×10^{-14} eV (i.e. of the order of the finite precision of the floating-point numbers as represented in the computer). Therefore the new algorithm is error-free in principle.

No matter what algorithm is employed in the computation, the parallelization on the CPU-only platforms scales well with respect to the number of processing cores in the test. A superlinear scaling may happen occasionally, e.g. a speedup of 6.6 times for P_{063} is observed with the O_{6C} computing setup. This can be attributed to the presence of the CPU cache. Data that does not fit into the cache of a single CPU core may fit if distributed among multiple cores causing a greater speedup. The GPU-dedicated implementation of the new algorithm affords the best performance. If multiple GPU devices are utilized, some extra acceleration that is proportional to the number of GPUs should be possible, provided the investigated system is large enough. For example, the computation of P_{221} on two GPUs takes 2.7 sec, which is 1.8 times faster than that (4.8 sec) on a single GPU.

Chapter 4

Parallel Parameterization Program

4.1 Introduction

Semiempirical quantum chemical methods are quite unique varieties of electronic structure theory in that the parameterization with regard to experimental or high-level theoretical reference data plays a much more important role than for *ab initio* and density functional methods during method development [26]. The accuracy of a semiempirical method is mostly governed by the underlying model and the quality of the optimized parameters. Although it has been widely recognized that the former determines the intrinsic accuracy of a semiempirical method, the latter is of great practical importance. Extended parameterization can improve the overall quality of an established semiempirical model to approach the inherent limit of accuracy for general applications. For example, the continuous parameterization endeavors for MNDO-type semiempirical methods [18,19] gave rise to AM1 [21], RM1 [76] and PMx ($x = 3$, 5, 6, and 7) [22,77–79] with progressively enhanced accuracy and capabilities. Moreover, a specific reparameterization of a semiempirical method may be useful for a targeted system of interest [80–83,145,146]. It must be stressed, however, that parameterization should not be overused, because breakthroughs in the development of semiempirical quantum chemical methods will mainly come from advances in the underlying theoretical model, rather than the mere reoptimization of the parameters. It should also be noted that the numerical values of certain parameters need to be confined to a certain range to retain their physical significance [26].

The parameterization of a semiempirical method aims at finding the optimum values for a set of parameters by fitting calculated molecular properties x^{calc} to reference data x^{ref} so that the error function is minimized. Because a definite range of allowed values is usually imposed on each parameter, this procedure can be viewed as a constrained optimization mathematically. A single evaluation of the error function may involve semiempirical calculations on hundreds of molecules. Furthermore, the optimization of the parameters will normally require many iterative cycles, and hence a huge number of evaluations of the error function, to ensure a comprehensive exploration of parameter space. Thus the parameterization of a semiempirical method can be an exceedingly time-consuming task.

One major project in this thesis was the development of a parallel parameterization program (PPP) that is able to take advantage of symmetric multiprocessing (SMP) computers and parallel optimization algorithms, in order to solve the constrained optimization problem in a more efficient way.[1] In the following, we first describe the mathematical foundations of the parameterization problem and the basic principles employed in PPP. We then outline the algorithms for local and global optimizations that have been implemented in PPP. Finally, an application is presented, namely the optimization of parameters in empirical dispersion corrections for the OM2 and OM3 methods.

During the thesis, specific semiempirical parameterizations were performed in two cases, in order to obtain an optimum semiempirical description of the enzymatic reaction catalyzed by dihydrofolate reductase [145] and of hydrogen bonding and proton transfer in water [146]. These studies will not be summarized here, since they have been published. The corresponding publications are attached in Appendices C and D.

4.2 Theory

Consider a semiempirical method that depends on n parameters

$$\mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}, \quad \forall p_i \in \mathbb{R} \tag{4.1}$$

that are optimized by calibrating against a set of m $(m \geq n)$ reference properties

$$\mathbf{x}^{\mathrm{ref}} = \begin{pmatrix} x_1^{\mathrm{ref}} \\ x_2^{\mathrm{ref}} \\ \vdots \\ x_m^{\mathrm{ref}} \end{pmatrix} \tag{4.2}$$

obtained from either experiment or high-level theoretical calculations. The residuals f_i are defined as the weighted differences between the calculated and reference values of the molecular properties,

$$\mathbf{f} = \begin{bmatrix} w_1\left(x_1^{\mathrm{calc}}(\mathbf{p}) - x_1^{\mathrm{ref}}\right) \\ w_2\left(x_2^{\mathrm{calc}}(\mathbf{p}) - x_2^{\mathrm{ref}}\right) \\ \vdots \\ w_m\left(x_m^{\mathrm{calc}}(\mathbf{p}) - x_m^{\mathrm{ref}}\right) \end{bmatrix}, \tag{4.3}$$

where w_i is the ith weighting factor whose unit is the inverse of the dimension of the reference property so that f_i is dimensionless. Equation (4.3) is a vector function of $\mathbf{p}$, i.e. $\mathbf{f}\colon \mathbf{R}^n \mapsto \mathbf{R}^m$. The objective of the parameterization of a semiempirical method is to find the minimizer

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} s(\mathbf{p}), \tag{4.4}$$

[1] In this chapter, PPP does not refer to the Pariser-Parr-Pople method. [5, 6]

where $s(\mathbf{p})$ is the sum of squared residuals,

$$s(\mathbf{p}) = \sum_{i=1}^{m} f_i^2. \tag{4.5}$$

The residual function (see Equation (4.3)) can be approximated by a first-order Taylor series expansion in the vicinity of an initial set of parameters $\mathbf{p}_0$

$$\mathbf{f}(\mathbf{p}_0 + \mathbf{h}) \approx \mathbf{f}(\mathbf{p}_0) + \mathbf{J}(\mathbf{p}_0)\mathbf{h}, \tag{4.6}$$

where $\mathbf{h}$ represents an infinitesimal shift and $\mathbf{J}$ is the Jacobian at the point $\mathbf{p}_0$. The Jacobian contains the first-order partial derivatives of the residual components with respect to the parameters organized in an $m \times n$ matrix

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial p_1} & \dots & \frac{\partial f_1}{\partial p_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial p_1} & \dots & \frac{\partial f_m}{\partial p_n} \end{pmatrix}. \tag{4.7}$$

After applying the linear approximation in Equation (4.6) to the sum of squares given by Equation (4.5), we have

$$s(\mathbf{p}) \approx (\mathbf{f}_0 + \mathbf{J}_0\mathbf{h})^\top (\mathbf{f}_0 + \mathbf{J}_0\mathbf{h}), \tag{4.8}$$

where $\mathbf{f}_0$ and $\mathbf{J}_0$ are shorthand notations for $\mathbf{f}(\mathbf{p}_0)$ and $\mathbf{J}(\mathbf{p}_0)$, respectively. The approximate gradient of $s(\mathbf{p})$ can also be derived from Equation (4.8)

$$\mathbf{g}(\mathbf{p}) \approx 2\mathbf{J}_0^\top \mathbf{f}_0 + 2\mathbf{J}_0^\top \mathbf{J}_0\mathbf{h}. \tag{4.9}$$

It is set to zero when Equation (4.5) is minimized. Upon rearrangement the normal equations written in matrix form become

$$\mathbf{J}_0^\top \mathbf{J}_0\mathbf{h} = -\mathbf{J}_0^\top \mathbf{f}_0. \tag{4.10}$$

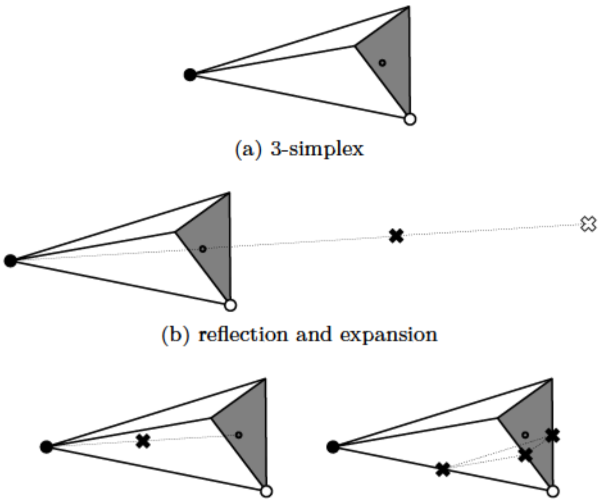
Equation (4.10) constitutes the theoretical basis for the gradient-based algorithms.

Since the analytical expressions for the components of the residual function are normally not available, the Jacobian has to be constructed by means of numerical differentiation,

$$J_{ij} = \frac{f_i(\mathbf{p}_0 + \mathbf{e}_j\delta) - f_i(\mathbf{p}_0 - \mathbf{e}_j\delta)}{2\delta}, \tag{4.11}$$

which can be a very demanding computational task. In Equation (4.11), $\mathbf{e}_j$ is the jth column of an identity matrix and δ denotes the small variation of the jth parameter.

The parameterization of a semiempirical method is a typical non-linear least squares problem that can be solved by various general optimization methods. Derivative-free optimization techniques would seem to be particularly attractive, because the imprecise and time-consuming gradient evaluation can be omitted completely. To prepare for the description of the derivative-free algorithms in the next section, we will first introduce the concept of a simplex and the elementary operations associated with it [147].



(a) 3-simplex

(b) reflection and expansion

(c) contraction

(d) reduction

Figure 4.1: A 3-simplex and the relevant operations for derivative-free optimization. The vertices of the highest and lowest values are designated by black and white circles, respectively. The gray triangles represent the reflection plane and the centroids are denoted by smaller circles. The points of reflection, contraction, and reduction are marked by black crosses, respectively, whereas the white cross indicates the expansion point.

A set of parameters, i.e. $\mathbf{p}$ in Equation (4.1), can be considered as a point in an n-dimensional space. A simplex of parameters is a special polytope of $(n+1)$ vertices in n dimensions. Each vertex is defined by a parameter set $\mathbf{p}_i$ and its value is given by the sum of squared residuals (see Equation (4.5)). For example, a 3-simplex in Figure 4.1(a) is a tetrahedron. It can represent a set of 3 parameters that need to be optimized.

There are four basic operations on a simplex in numerical optimization, namely reflection, expansion, contraction, and reduction. After initial evaluation of all vertices of a simplex, the vertices of the highest and lowest values are sorted and labeled as $\mathbf{p}_{\mathrm{h}}$ and $\mathbf{p}_{\mathrm{l}}$, respectively. The vertices except $\mathbf{p}_{\mathrm{h}}$ then form an abstract plane, where $\bar{\mathbf{p}}$ denotes its centroid

$$\bar{\mathbf{p}} = \frac{1}{n+1}\sum_{i=1}^{n+1}\mathbf{p}_i \qquad (i \neq \mathrm{h}). \tag{4.12}$$

The reflection operation is to reflect the vertex $\mathbf{p}_{\mathrm{h}}$ through the opposite side of the plane and obtain a new point $\mathbf{p}^{\mathrm{r}}$

$$\mathbf{p}^{\mathrm{r}} = (1+\alpha)\bar{\mathbf{p}} - \alpha\mathbf{p}_{\mathrm{h}}, \tag{4.13}$$

where α is a positive constant named the reflection coefficient. This point can be further expanded along the line joining $\mathbf{p}_{\mathrm{h}}$ and $\bar{\mathbf{p}}$ in the same direction giving $\mathbf{p}^{\mathrm{e}}$ in the second operation

$$\mathbf{p}^{\mathrm{e}} = \gamma\mathbf{p}_{\mathrm{r}} + (1-\gamma)\bar{\mathbf{p}}, \tag{4.14}$$

where γ is greater than one and called the expansion coefficient. The contraction operation is to move the highest vertex towards the centroid of the plane which results in

$$\mathbf{p}^{\mathrm{c}} = \beta \mathbf{p}_{\mathrm{h}} + (1 - \beta)\bar{\mathbf{p}}, \tag{4.15}$$

where the coefficient β lies between 0 and 1. The last operation displaces all vertices to the lowest $\mathbf{p}_{\mathrm{l}}$

$$\mathbf{p}_i = \mathbf{p}_{\mathrm{l}} + \delta(\mathbf{p}_i - \mathbf{p}_{\mathrm{l}}) \qquad (i \neq \mathrm{l}), \tag{4.16}$$

so that the size of the simplex is shrunk. The allowed interval of δ is between 0 and 1. All four operations are demonstrated for the example of a 3-simplex in Figure 4.1.

4.3 Technical details

In the parameterization of a semiempirical method, the aim is to find an optimal set of parameters that minimize the objective function (see Equation (4.5)). In numerical analysis, there are a large number of algorithms available for such minimization problems. These algorithms can be classified into two types, local optimization and global optimization. Local optimization finds the minimizer that gives the lowest value of a function in the neighborhood of the starting point. For example, p_1 may be reached from p_0 by a local optimization (see Figure 4.2). On the other hand, the overall lowest function value in the search domain is targeted in global optimization, e.g. the point p_{g} in Figure 4.2. Both local and global approaches can be utilized for parameterization. In both cases, the comprehensive exploration of parameter space will generally require an excessive number of function evaluations, and it should thus be preferable to take advantage of algorithms devised for parallel computers. Some representative algorithms for both local and global optimizations implemented in PPP are described in the next subsections.

4.3.1 Local optimization

Algorithms for local optimization usually lead to a minimum in the vicinity of the initial point. If a function is differentiable and its derivatives can be analytically computed or accurately estimated by numerical means, then a wide variety of gradient-based methods is available for searching the best parameters. Otherwise, a direct derivative-free method may be useful, especially if it is inconvenient or too demanding to evaluate the function derivatives.

4.3.1.1 Gradient-based algorithms

Two of the most widely used gradient-based algorithms are the Gauss-Newton method [147,148] and the Levenberg-Marquardt method [149,150]. The basic theory has been briefly reviewed in the previous section, Algorithms 4.1 and 4.2 present the pseudocode for both methods.

p_0 gives an initial guess for the parameters and *check_conv* verifies whether the stopping criteria are satisfied in the end. The vector-valued residual function (Equation (4.3)) and the Jacobian (Equation (4.7)) need to be evaluated in

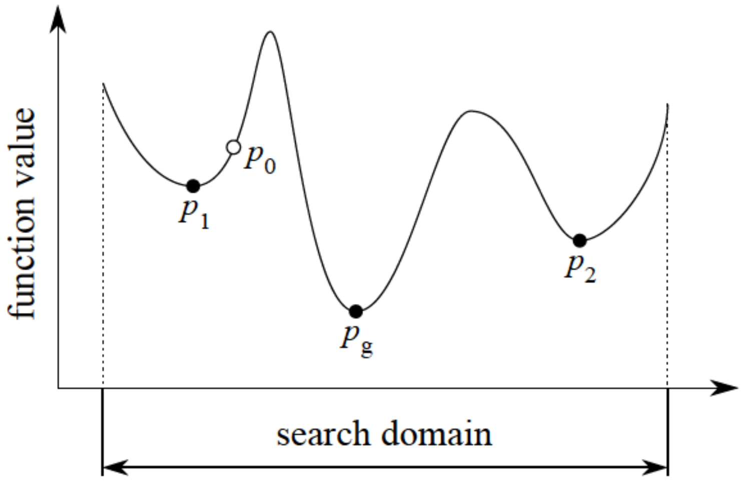


Figure 4.2: The minima of a function in an interval. The starting point p_0 for optimization is indicated by a white circle. All minima are represented by black circles. p_1 and p_2 are local minima, whereas p_g denotes the global minimum within the search domain.

Algorithm 4.1 The Gauss-Newton method.

```
$\mathbf{p} \leftarrow \mathbf{p}_0$
repeat
    $\mathbf{f} \leftarrow \mathbf{f}(\mathbf{p})$
    $\mathbf{J} \leftarrow \mathbf{J}(\mathbf{p})$
    solve $\mathbf{J}^\top \mathbf{J}\mathbf{h} = -\mathbf{J}^\top \mathbf{f}$
    search $\alpha^* \leftarrow \arg\min s(\mathbf{p} + \alpha\mathbf{h})$
    $\{\mathbf{p}, s\} \leftarrow \{\mathbf{p} + \alpha^*\mathbf{h}, s(\mathbf{p})\}$
    $conv \leftarrow check\_conv()$
until $conv$
```

Algorithm 4.2 The Levenberg-Marquardt method.

```
$\mathbf{p} \leftarrow \mathbf{p}_0$
repeat
    $\mathbf{f} \leftarrow \mathbf{f}(\mathbf{p})$
    $\mathbf{J} \leftarrow \mathbf{J}(\mathbf{p})$
    $\lambda \leftarrow update()$
    solve $(\mathbf{J}^\top \mathbf{J}\mathbf{h} + \lambda\mathbf{M}) = -\mathbf{J}^\top \mathbf{f}$
    $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{h}$
    $conv \leftarrow check\_conv()$
until $conv$
```

both methods at the beginning of the loops. In the Gauss-Newton method, the normal equations are solved immediately for the direction of steepest descent. This is followed by a line search for an optimal step size α^* in this direction. Then a new set of parameters is generated to test of convergence. On the other hand, the Levenberg-Marquardt method introduces a damping factor λ into the normal equations which needs to be updated in each iteration (see lines 5 and 6 in Algorithm 4.2). For small values of λ the Levenberg-Marquardt method approaches the Gauss-Newton method. The matrix $\mathbf{M}$ in the modified normal equations is a positive diagonal matrix. It can be either an identity matrix $\mathbf{I}$ or $\mathrm{diag}(\mathbf{J}^\top\mathbf{J})$, which might be helpful to converge the procedure in some circumstances.

There may be tens of parameters in a semiempirical method and thousands of reference properties for a parameterization. In addition, the analytical derivatives with respect to the parameters are unknown in most cases. Hence the numerical evaluation of the Jacobian by using a finite difference method can become the computational bottleneck (see line 4 in Algorithms 4.1 and 4.2). It can thus be beneficial to make use of multiple processors, e.g. all CPU cores, for this task. The parallel pseudocode for the computation of the Jacobian is given in Algorithm 4.3.

Algorithm 4.3 The computation of the Jacobian by a finite difference method on multiple processors.

```
for $i_p \leftarrow 1$ to $n_p$ do
    $\{i_s, i_e\} \leftarrow \{set\_start(i_p, n_p, n), set\_end(i_p, n_p, n)\}$
    for $i \leftarrow i_s$ to $i_e$ do
        $\mathrm{f}_+ \leftarrow \mathrm{f}(\mathrm{p}_0 + \mathrm{e}_i\delta)$
        $\mathrm{f}_- \leftarrow \mathrm{f}(\mathrm{p}_0 - \mathrm{e}_i\delta)$
        $\mathbf{J}_i \leftarrow (\mathrm{f}_+ - \mathrm{f}_-)/2\delta$
    end for
end for
```

The parallelism in Algorithm 4.3 is achieved by a balanced distribution of the computation of the Jacobian matrix for n parameters on n_p processors. The column ranges of the Jacobian, i.e. from i_s to i_e, assigned to each processor are determined by the functions *set_start* and *set_end*. The speedup will trivially scale proportional to the number of processors in general.

4.3.1.2 Derivative-free algorithms

There are at least two reasons for employing a derivative-free optimization algorithm for the parameterization. First, the numerical evaluation of the Jacobian can be very expensive. Second, the residual functions may be somewhat noisy with respect to variations of the parameters in some calculations so that one may not always be able to trust the derivatives approximated by finite differences.

The Nelder-Mead method [151] is a well-established numerical technique that requires no knowledge of derivatives for a general nonlinear optimization problem in a multidimensional space. This method, however, usually demands a very large number of function evaluations to converge a minimization, and therefore a generalization of the Nelder-Mead method for parallel processors was developed [152]. The pseudocode for the parallel Nelder-Mead method is given in Algorithm 4.4, where 1-based indexing is used throughout.

It should not be difficult to parallelize the construction of an initial simplex on n_p processors (lines 1 to 3). Unlike the primitive algorithm that tries to replace the highest vertex of the simplex with a better point, the parallel approach attempts to simultaneously improve n_p vertices of the highest values. The computation corresponding to one vertex is assigned to a single processor so that a coarse-grained parallelization of the Nelder-Mead method can be realized for parameterization. As can be seen in Algorithm 4.4, the four elementary operations on a simplex, i.e. reflection, expansion, contraction, and reduction, are all encompassed in the parallel procedure. Moreover, if there is only one processor in use, i.e. $n_p \leftarrow 1$, the parallel Nelder-Mead algorithm collapses to the original version. Finally, it is crucial to notice that the starred variables, i.e. $\mathbf{p}_i^*$ and s_i^*, must be independent on each processor to avoid data race conditions.

4.3.2 Global optimization

Global optimization appears to be attractive since it promises to locate the best point in the variable space. However, it is not really that appealing as far as the parameterization of a semiempirical method is concerned. First, the essential physical meaning of some parameters must be preserved, which implies a certain locality on the search domain. Second, an overuse of the global optimization technique should be avoided as the reference data is usually limited to a set of representative molecules.

Global optimization is very difficult in general, because the objective function may have many local minima, and criteria are normally lacking to determine whether a local minimum is truly global. There is a large number of monographs on global optimization. For the purpose of parameterizing a semiempirical quantum chemical method, a parallel controlled random search (CRS) was implemented in the PPP program.

The CRS algorithm belongs to the stochastic direct search methods of global optimization. It is robust for optimizing a function with noise.[2] The original version was proposed by Price in the 1970s [153–156]. Its effectiveness was improved by many others later on [157–159]. Nevertheless, it is critical to parallelize the algorithm for more efficiency, because it requires far more function evaluations to adequately sample the points in the search domain [160]. The parallel pseudocode is given in Algorithm 4.5.

First, a configuration of N trial points ($N \gg n$) is generated at random over the search domain $\mathbb{V}$, which can be easily parallelized on multiple processors. Then each working processor randomly samples $(n+1)$ out of N points in the configuration and produces an updated point according to some combination rules. There is no need to synchronize the computations on the processors at the end of the loop (line 12). Once a processor finishes the computation for a new point, the maximum of the N points is found and compared with the new point. If the new point has a smaller value, then the maximum will be replaced. It should be emphasized that the modification of the maximum is enclosed in a critical region (lines 13 to 18), which permits execution by only one processor at

[2] There is no guarantee for the smoothness of the residual functions with respect to the parameters of a semiempirical method.

Algorithm 4.4 The parallel Nelder-Mead method.

```
for $i \leftarrow 1$ to $(n+1)$ do                                ▷ create an initial simplex
    $s_i \leftarrow s(\mathbf{p}_i)$
end for
repeat
    $sort(s_i, \mathbf{p}_i)$                           ▷ $s_1(\mathbf{p}_1) < s_2(\mathbf{p}_2) < \cdots < s_{n+1}(\mathbf{p}_{n+1})$
    $\bar{\mathbf{p}} \leftarrow \frac{1}{n+1-n_p} \sum_{i=1}^{n+1-n_p} \mathbf{p}_i$
    $shrink \leftarrow$ true
    for $i \leftarrow (n+2-n_p)$ to $(n+1)$ do    ▷ loop through the $n_p$ worst points
        $\{\mathbf{p}^r_i, s^r_i\} \leftarrow \{reflect(\mathbf{p}_i), s(\mathbf{p}^r_i)\}$                  ▷ reflection
        if $s^r_i < s_1$ then
            $\{\mathbf{p}^e_i, s^e_i\} \leftarrow \{expand(\mathbf{p}^r_i), s(\mathbf{p}^e_i)\}$              ▷ expansion
            if $s^e_i < s_1$ then
                $\{\mathbf{p}^*_i, s^*_i\} \leftarrow \{\mathbf{p}^e_i, s^e_i\}$
            else
                $\{\mathbf{p}^*_i, s^*_i\} \leftarrow \{\mathbf{p}^r_i, s^r_i\}$
            end if
            $strink \leftarrow$ false
        else if $s_1 \leq s^r_i$ and $s^r_i < s_{i-1}$ then
            $\{\mathbf{p}^*_i, s^*_i\} \leftarrow \{\mathbf{p}^r_i, s^r_i\}$
            $strink \leftarrow$ false
        else if $s_{i-1} \leq s^r_i$ then
            $\{\mathbf{p}^c_i, s^c_i\} \leftarrow \{contract(\mathbf{p}_i), s(\mathbf{p}^c_i)\}$           ▷ contraction
            $\{\tilde{\mathbf{p}}_i, \tilde{s}_i\} \leftarrow min(\{\mathbf{p}_i, s_i\}, \{\mathbf{p}^r_i, s^r_i\})$
            if $s^c_i < \tilde{s}_i$ then
                $\{\mathbf{p}^*_i, s^*_i\} \leftarrow \{\mathbf{p}^c_i, s^c_i\}$
                $strink \leftarrow$ false
            else
                $\{\mathbf{p}^*_i, s^*_i\} \leftarrow \{\tilde{\mathbf{p}}_i, \tilde{s}_i\}$
            end if
        end if
    end for
    if $shrink$ then                                               ▷ reduction
        for $i = 2$ to $(n+1-n_p)$ do
            $\{\mathbf{p}_i, s_i\} \leftarrow \{reduce(\mathbf{p}_i), s(\mathbf{p}_i)\}$
        end for
        for $i = (n+2-n_p)$ to $(n+1)$ do
            $\{\mathbf{p}_i, s_i\} \leftarrow \{reduce(\mathbf{p}^*_i), s(\mathbf{p}_i)\}$
        end for
    else
        for $i = (n+2-n_p)$ to $(n+1)$ do
            $\{\mathbf{p}_i, s_i\} \leftarrow \{\mathbf{p}^*_i, s^*_i\}$
        end for
    end if
    $conv \leftarrow check\_conv()$
until $conv$
```

a time, so that a corresponding data race condition can be completely avoided. The final step is to check for convergence. The outermost loop is terminated if some suitable convergence criterion is satisfied.

Algorithm 4.5 The parallel controlled random search method.

```
for $i = 1$ to $N$ do                                  ▷ $N$ trial points
    $\mathbf{p}_i \leftarrow random(\mathbb{V})$
    $s_i \leftarrow s(\mathbf{p}_i)$
end for
repeat
    for all processors do
        $\mathbf{r} \leftarrow sample(N, n+1)$
        for $i = 1$ to $(n+1)$ do
            $\mathbf{p}^*_i \leftarrow \mathbf{p}_{r_i}$
        end for
        $\{\mathbf{p}^{\mathrm{u}}, s^{\mathrm{u}}\} \leftarrow \{update(\mathbf{p}^*), s(\mathbf{p}^{\mathrm{u}})\}$      ▷ a new point
    end for
    Begin Critical
    $\{\mathbf{p}_m, s_m\} \leftarrow max(\{\mathbf{p}_i, s_i\}, N)$
    if $s^{\mathrm{u}} < s_m$ then
        $\{\mathbf{p}_m, s_m\} \leftarrow \{\mathbf{p}^{\mathrm{u}}, s^{\mathrm{u}}\}$
    end if
    End Critical
    $conv \leftarrow check\_conv()$
until $conv$
```

4.3.3 Implementation

No matter whether gradient information is used or not in the aforementioned optimization algorithms, the primary computational task for parameterizing a semiempirical method is the evaluation of the sum of squared residual functions (see Equation (4.5)). Since the function evaluations are mostly independent of each other, they can be assigned to the processors of a symmetric multiprocessing computer and executed in parallel (see Figure 4.3). The coarse-grained parallelization is organized in a master-worker pattern. PPP, as a master program, is responsible for generating new trials of parameters, distributing the computations and accumulating the final sum-of-squares, whereas the function evaluations are invoked via system calls to the MNDO99 program as workers on parallel processors. Please note that the MNDO99 program should be executed in a sequential mode for this particular situation to maximize the parallelism of the function evaluations on multiple processors.

4.4 Demonstrations

Three studies on the parameterization of semiempirical methods were carried out as demonstrations. The first one is a specific parameterization for an enzyme-catalyzed hydride transfer based on the AM1 Hamiltonian [145]. The second one is an extensive parameterization of MNDO-type (MNDO, AM1, and PM3) and

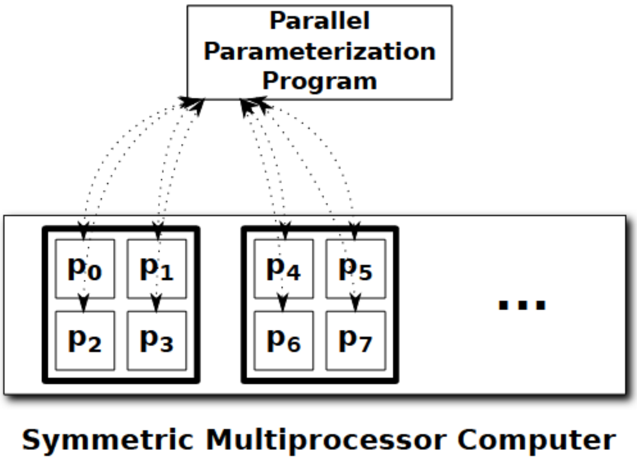

Parallel Parameterization Program
p_0
p_1
p_2
p_3
p_4
p_5
p_6
p_7
...
Symmetric Multiprocessor Computer

Figure 4.3: A diagram for the coarse-grained parallelization employed in the PPP program. The multi-core CPUs are represented by bold squares and the smaller squares indicate the processing cores.

OMx ($x = 1$, 2, and 3) methods for hydrogen bonding and proton transfer in water clusters [146]. Both studies have been published, the papers are attached in Appendices C and D. The third investigation concerns the parameterization of dispersion corrections in the OM2 and OM3 framework for weakly bound complexes. The corresponding results are presented in this section.

Noncovalent intermolecular interactions are weak by nature, but they are ubiquitous in biological systems and of particular prominence for protein folding and nucleobase stacking in DNA (deoxyribonucleic acid) and RNA (ribonucleic acid) [106]. One of the noncovalent interactions is dispersion which is attractive for any pair of atoms due to instantaneous fluctuations of the electrons [108,109]. A proper theoretical description of dispersion is difficult [105,161]. Some commonly used quantum chemical methods, e.g. *ab initio* MO theory and DFT with standard functionals, do not account for dispersion effects properly [113, 114]. High-level correlated *ab initio* methods, e.g. MP2 or CCSD(T), with large basis sets are required for an accurate treatment of dispersion interactions accurately [110–112]. Such calculations are practical only for rather small molecules. However, unlike other effects in electronic structure theory, dispersion interactions can be reasonably modeled by molecular mechanics force fields in terms of empirical formulas with suitable parameters [115, 116]. In order to retrieve the dispersion energy for methods based on quantum mechanics with an affordable computational effort, dispersion terms from empirical Lennard-Jones potentials [117] have been added to the energies from the Hartree-Fock method [119–121, 162], some DFT methods [122–127], the self-consistent-charge density-functional tight-binding (SCC-DFTB) method [118], and several semiempirical methods, e.g. AM1, PM3, and PM6 [128, 130–133, 163]. Dispersion corrected OMx methods have been explored in our group [129]. The current study aims at testing alternative dispersion formulas, searching the parameter space more thoroughly, and reaching the best accuracy.

The empirical dispersion energy for the OM2 and OM3 methods is given by

$$E_{\mathrm{d}} = -s_6 \sum\sum_{\mathrm{B>A}} f_{\mathrm{d}} \frac{C_{\mathrm{AB}}}{R_{\mathrm{AB}}^6}, \tag{4.17}$$

where s_6 is a global scaling factor, R_{AB} is the distance between atoms A and B, and $C_{\mathrm{AB}}/R_{\mathrm{AB}}^6$ is an analogue of the second term for the Lennard-Jones potential. The C_{AB} coefficient is computed as the geometric mean of the C_6 parameters for a pair of atoms

$$C_{\mathrm{AB}} = \sqrt{C_6^{\mathrm{A}} C_6^{\mathrm{B}}}. \tag{4.18}$$

A damping function f_{d} must be used for every term in the sum to prevent the dispersion energy from assuming unphysically large negative values when two atoms, A and B, are coming very close to each other. Several functional forms have been proposed for f_{d} in the literature. Two different damping functions, denoted as $f_{\mathrm{D_1}}$ and $f_{\mathrm{D_2}}$, are studied in this work, namely

$$f_{\mathrm{D_1}} = \left[1 - \exp\left(-c_{\mathrm{d}} \frac{R_{\mathrm{AB}}^7}{R_{\mathrm{v}}^7}\right)\right]^4 \tag{4.19}$$

and

$$f_{\mathrm{D_2}} = \frac{1}{1 + \exp\left[-c_{\mathrm{d}}\left(R_{\mathrm{AB}}/R_{\mathrm{v}} - 1\right)\right]}. \tag{4.20}$$

c_{d} are the damping coefficients and R_{v} is the sum of the van der Waals radii

$$R_{\mathrm{v}} = R_0^{\mathrm{A}} + R_0^{\mathrm{B}}. \tag{4.21}$$

The numerical values of c_{d} can be quite different for the two functions, e.g. 3.0 in Equation (4.19) and 20.0 in Equation (4.20) were used in previous studies [118, 122, 123]. When R_{AB} becomes infinite, both sigmoid functions approach unity, and hence the correct R_{AB}^{-6} asymptotic behavior of the dispersion interaction will be fulfilled. However, there is a fundamental difference between these two functions at short distances. If R_{AB} is zero, Equation (4.19) is zero precisely, whereas Equation (4.20) is not zero (albeit very tiny). In the latter case, $-C_{\mathrm{AB}}/R_{\mathrm{AB}}^6$ will diverge when the two atoms are very close; however, this happens at unrealistically short distances that are generally not encountered in molecular systems.

The S22 set of noncovalent complexes [164] was employed for the parameterization of the dispersion-corrected OM2 and OM3 methods (denoted as OM2-D and OM3-D, respectively). This training set includes 7 hydrogen bonded complexes, 8 dispersion force dominated complexes, and 7 complexes of mixed interactions (see Figure 4.4). The reference properties for the parameterization, i.e. binding energies, hydrogen bond lengths and angles, and some relevant atomic distances mainly governed by dispersion forces, are listed in Table 4.1 together with the associated weighting factors. There are 141 reference data in total. Since the potential energy surface for a weakly bound system may be rather flat, stringent convergence criteria were employed for both the SCF calculations, in which the variation of the electronic energy for successive iterations must be less than 1.0×10^{-9} eV and the maximum change of the density matrix

Table 4.1: The number (No.) of reference data and the associated weighting factors (w) used in the parameterization of the OM2-D and OM3-D methods. The molecular properties include binding energy (ΔE), hydrogen bond length (r_H) and angle (a_H), important atomic distances dominated by dispersion interactions involving one hydrogen atom (r_HD) and heavy atoms only (r_D).

	ΔE	r_H	a_H	r_HD	r_D
No.	22	14	14	32	59
w	10.0 mol/kcal	100.0 Å^{-1}	1.0 degree^{-1}	100.0 Å^{-1}	100.0 Å^{-1}

elements must be less than 1.0×10^{-9}, and for the geometry optimizations of single fragments and whole complexes, in which the gradient norm must be less than 0.01 kcal/(mol · Å).

The parameters to be optimized are s_6, c_d, C_6, and R_0. The first two are general parameters not associated with any atoms, whereas the others possess distinct values for different atoms. Two damping functions, f_{D_1} and f_{D_2} (see Equations (4.19) and (4.20)), were investigated and compared in this work and the relevant methods are denoted by suffixes "-D$_1$" and "-D$_2$", respectively.

Because there are two types of parameters, the parameterizations were carried out in two steps. First, the optimal values for s_6 and c_d were found by a grid search, in which s_6 was varied from 0.4 to 1.0 and c_d was varied in the interval of 0.5 to 4.0 for f_{D_1} and 12.0 to 24.0 for f_{D_2}, respectively. The granularity of the grid was 0.1. The top 5 combinations of s_6 and c_d sorted by the root-mean-square deviation (RMSD) of the binding energy are given in Table 4.2 for the dispersion-corrected OM2 and OM3 methods. It can be seen that all reference properties are fairly insensitive to small variations of s_6 and c_d in the regions that exhibit the smallest deviations of the binding energy. We have adopted the combinations of s_6 and c_d listed in the first row of each method in Table 4.2 as suitable parameters for the empirical dispersion corrections. Keeping those s_6 and c_d fixed for each method, the atomic parameters, C_6 and R_0, for hydrogen, carbon, nitrogen, and oxygen were optimized in the second step. There is a connection between the C_6 coefficients and the atomic polarizabilities [107,165]. In addition, the R_0 parameters are the van der Waals radii of the atoms [166]. Hence the underlying physical significance of both parameters must be preserved during parameterization. Therefore the initial values for C_6 and R_0 were drawn from reference [123] and further optimizations were confined by allowing variations of at most ±10% from the original values. The initial and final values of C_6 and R_0 are given in Table 4.3 for the OM2-D and OM3-D methods. Evidently, the second parameterization step (see Table 4.4) achieves only minor improvements of the reference properties compared with the first step (Table 4.2). Overall, there are substantial enhancements relative to the results from the original OM2 and OM3 methods, especially in the case of OM3 (see Table 4.4).

A comparison of the molecular properties for the S22 set calculated by using the standard OM2 and OM3 methods and the dispersion-corrected OM2-D and OM3-D methods is given in Table 4.4. The OM2-based methods, i.e. OM2, OM2-D$_1$, and OM2-D$_2$, always produce smaller errors for r_H than the OM3 counterparts, but the RMSDs for r_H are almost unchanged for a given method with or without dispersion corrections. a_H and r_HD are very slightly improved by

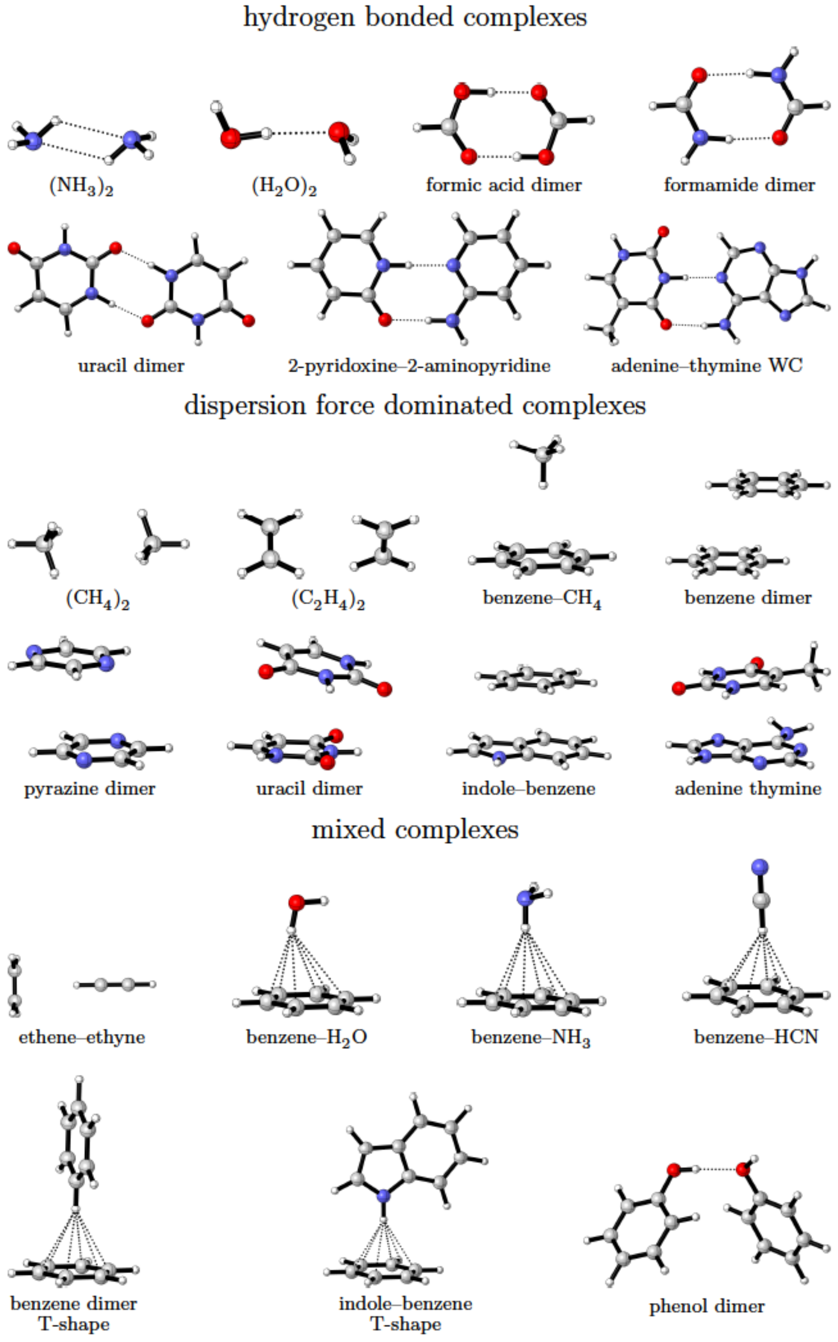


Figure 4.4: The S22 training set. The carbon, nitrogen, and oxygen atoms are represented by gray, blue, and red balls, respectively. The hydrogen atoms are shown as smaller white balls. The noncovalent interactions involving hydrogen atoms are indicated by dashed lines.

Table 4.2: Root-mean-square deviations of the OM2-D and OM3-D binding energies (ΔE, in kcal/mol), hydrogen bond lengths (r_{H}, in Å) and angles (a_{H}, in degree), and some important atomic distances involving dispersion interactions (r_{HD} and r_{D}, both in Å) for the S22 set in the grid search for s_6 and c_d.

(a) OM2-D$_1$

s_6	c_d	ΔE	r_{H}	a_{H}	r_{HD}	r_{D}
0.8	0.6	1.26	0.18	3.65	0.34	0.37
0.7	0.8	1.27	0.18	3.24	0.31	0.34
0.9	0.5	1.27	0.17	3.79	0.40	0.37
0.7	0.7	1.27	0.18	3.42	0.32	0.36
0.6	1.0	1.27	0.18	2.90	0.29	0.37

(b) OM2-D$_2$

s_6	c_d	ΔE	r_{H}	a_{H}	r_{HD}	r_{D}
0.5	12.0	1.24	0.18	3.01	0.30	0.39
0.5	12.1	1.24	0.18	3.01	0.30	0.39
0.5	12.2	1.24	0.18	3.02	0.30	0.39
0.5	12.3	1.24	0.18	3.01	0.30	0.39
0.5	12.4	1.24	0.18	3.02	0.30	0.39

(c) OM3-D$_1$

s_6	c_d	ΔE	r_{H}	a_{H}	r_{HD}	r_{D}
0.6	1.4	0.80	0.35	1.51	0.42	0.30
0.6	1.5	0.80	0.35	1.57	0.42	0.30
0.6	1.6	0.80	0.35	1.64	0.45	0.30
0.6	1.3	0.81	0.35	1.44	0.42	0.30
0.7	1.0	0.81	0.35	1.23	0.46	0.29

(d) OM3-D$_2$

s_6	c_d	ΔE	r_{H}	a_{H}	r_{HD}	r_{D}
0.6	18.6	0.83	0.35	1.72	0.44	0.32
0.6	18.5	0.83	0.35	1.72	0.44	0.33
0.6	18.7	0.83	0.35	1.72	0.44	0.33
0.6	18.8	0.83	0.35	1.73	0.44	0.33
0.6	18.4	0.83	0.35	1.72	0.44	0.33

Table 4.3: The initial values of the C_6 (in $\mathrm{J} \cdot \mathrm{nm}^6 \cdot \mathrm{mol}^{-1}$) and R_0 (in nm) parameters for hydrogen, carbon, nitrogen, and oxygen and the optimized parameters for the OM2-D and OM3-D methods.

(a) initial values

	hydrogen	carbon	nitrogen	oxygen
C_6	0.14000000	1.75000000	1.23000000	0.70000000
R_0	0.10010000	0.14520000	0.13970000	0.13420000

(b) OM2-D$_1$

	hydrogen	carbon	nitrogen	oxygen
C_6	0.13836140	1.74837313	1.22660965	0.70235574
R_0	0.09195488	0.14057505	0.13120977	0.13943710

(c) OM2-D$_2$

	hydrogen	carbon	nitrogen	oxygen
C_6	0.13349720	1.74649564	1.22649173	0.68974720
R_0	0.10685589	0.14169369	0.14069819	0.13069564

(d) OM3-D$_1$

	hydrogen	carbon	nitrogen	oxygen
C_6	0.13699517	1.76051239	1.23050050	0.70000000
R_0	0.09359344	0.14369850	0.13970000	0.13420000

(e) OM3-D$_2$

	hydrogen	carbon	nitrogen	oxygen
C_6	0.14146973	1.75148586	1.23148586	0.70136073
R_0	0.10158635	0.15275442	0.14115458	0.13573278

Table 4.4: Root-mean-square deviations of the binding energy (ΔE, in kcal/mol), hydrogen bond length (r_{H}, in Å) and angle (a_{H}, in degree), and some important atomic distances involving dispersion interactions (r_{HD} and r_{D}, both in Å) for the S22 set calculated by using the standard and the dispersion-corrected OM2 and OM3 methods.

	ΔE	r_{H}	a_{H}	r_{HD}	r_{D}
OM2	2.69	0.17	3.24	0.42	1.54
OM2-D$_1$	1.36	0.18	2.73	0.27	0.35
OM2-D$_2$	1.26	0.18	3.24	0.30	0.38
OM3	2.59	0.34	3.08	0.44	6.82
OM3-D$_1$	0.81	0.35	1.97	0.41	0.27
OM3-D$_2$	0.81	0.35	1.64	0.41	0.31

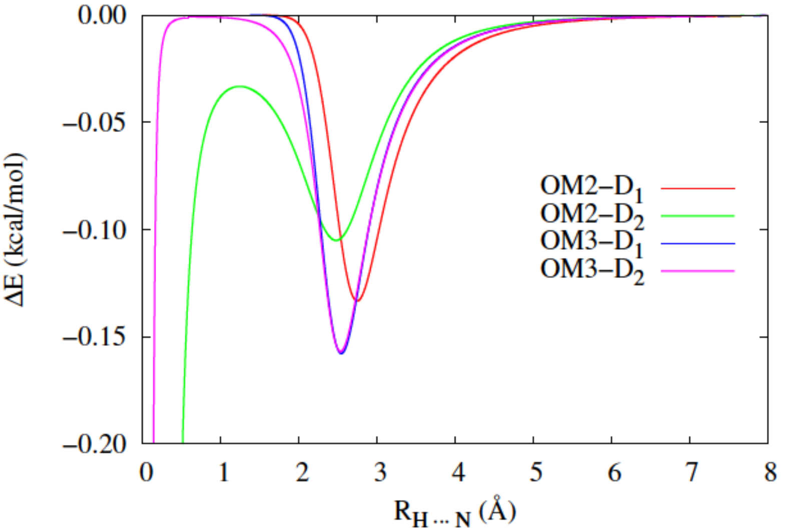


Figure 4.5: The empirical dispersion interaction energies between hydrogen and nitrogen atoms in the OM2-D_1, OM2-D_2, OM3-D_1, and OM3-D_2 methods.

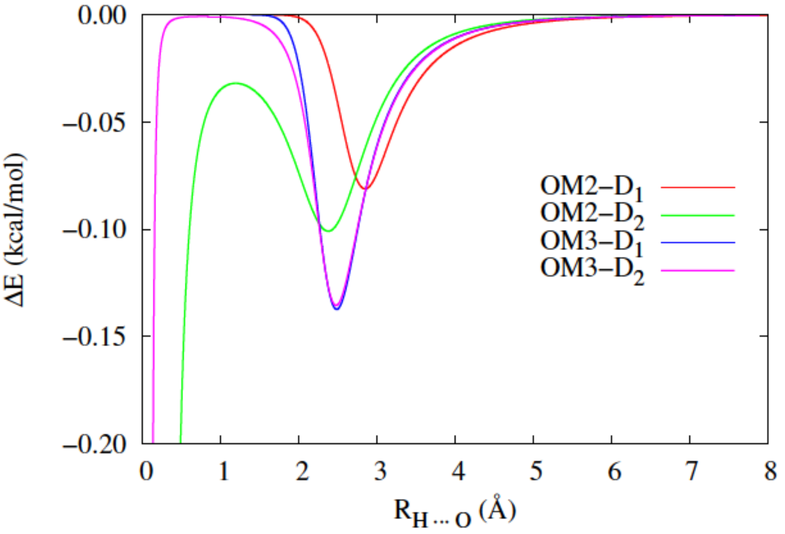


Figure 4.6: The empirical dispersion interaction energies between hydrogen and oxygen atoms in the OM2-D_1, OM2-D_2, OM3-D_1, and OM3-D_2 methods.

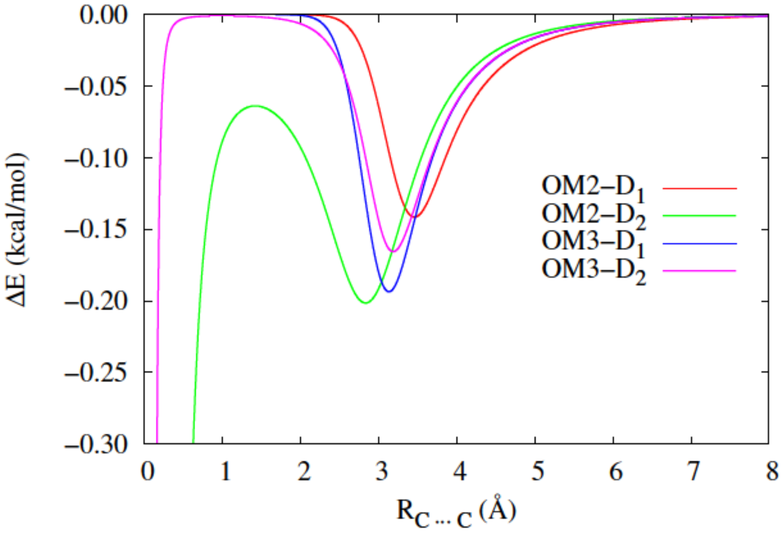


Figure 4.7: The empirical dispersion interaction energies between two carbon atoms in the OM2-D_1, OM2-D_2, OM3-D_1, and OM3-D_2 methods.

adding the empirical dispersion corrections. Overall, however, the impact of the dispersion corrections on the geometries of hydrogen bonds is very small. This can be easily understood. For one thing, hydrogen bonds are treated reasonably well already by the standard OM2 and OM3 methods. The strength of common hydrogen bonds is at least $\sim$ 2 kcal/mol [167], while the interaction energy contributed by the dispersion corrections is rather small, e.g. ≤ 0.15 kcal/mol between hydrogen and an electronegative atom (see Figures 4.5 and 4.6 for the potential curves of $H \cdots N$ and $H \cdots O$). Hence, the hydrogen bonding interactions are primarily electrostatic attractions. On the other hand, the dispersion corrections significantly improve the interatomic distances dominated by dispersion forces. Since dispersion effects are formally missing in the standard OM2 and OM3 methods, the errors of r_D are unacceptably large, e.g. 1.54 Å for OM2 and 6.82 Å for OM3. The original OM2 and OM3 methods thus do not produce realistic structures for complexes dominated by dispersion interactions, and the corresponding binding energies are meaningless. It is obvious that the predicted r_D values are greatly improved if the methods are augmented with dispersion corrections (see Table 4.4). Moreover, it seems that the results are not much affected by the choice of the damping function. The accuracy of OM2-D_1 and OM2-D_2 (also of OM3-D_1 and OM3-D_2) with respect to r_D is quite similar. The OM3-D methods have smaller errors than OM2-D with regard to the binding energy, e.g. 0.8 kcal/mol for the former versus $\sim$ 1.3 kcal/mol for the latter, but both perform better than the standard OM2 and OM3 methods. To summarize, the empirical dispersion corrections enhance the description of the van der Waals complexes, and the quality of the four methods, i.e. OM2-D_1, OM2-D_2, OM3-D_1, and OM3-D_2, for noncovalent systems is comparable in general.

Plots of the OM2-D and OM3-D dispersion energies for two carbon atoms are illustrated in Figure 4.7. Obvious wells are found at distances of 3.0 Å to 3.5 Å, which roughly correspond to the sum of the van der Waals radii of two carbon atoms.[3] In all four semiempirical methods, the well depths due to dispersion between a pair of carbon atoms vary from −0.15 to −0.20 kcal/mol. These values are reasonable and in good agreement with the dispersion energies for various DFT-D methods [124, 125]. Because the damping function f_{D_2} of Equation (4.20) does not go to zero when two atoms approach each other closely, unphysical attractions are witnessed when the distance between two carbon atoms drops below 1.0 Å (see the green and magenta curves in Figure 4.7). The steepness of the attractive curve in the covalent region is governed by the damping coefficient c_d. The optimized c_d value for OM2-D$_2$ is less than that for the OM3-D$_2$ method, i.e. 12.0 versus 18.6. Consequently, the curve for OM2-D$_2$ has a broader well that extends to ~ 1.5 Å and then drops off again for a pair of carbon atoms. By contrast, the damping function from Equation (4.19) is almost exactly zero for short distances, e.g. for $R_{\mathrm{C}\cdots\mathrm{C}} \leq 2.0$ Å, and the empirical dispersion corrections affect essentially only the van der Waals region, if f_{D_1} is employed as the damping function. For large separations, all empirical dispersion corrections fall off according to the asymptotic R_{AB}^{-6} law, by construction. In conclusion, even though the four dispersion corrected methods are of similar accuracy for noncovalent complexes, the OM2-D$_1$ and OM3-D$_1$ methods are recommended because the damping function f_{D_1} behaves more properly than f_{D_2} over the full range of interatomic distances.

[3] The van der Waals radius of a carbon atom is 1.70 Å [166]. Note that this value is subject to debate [168].

Chapter 5

Benchmark of Semiempirical Methods on Protein Structures

Because of the improved performance of the MNDO99 program [84] on hybrid CPU-GPU computing architectures [74, 169], full geometry optimizations were carried out for a set of 28 proteins characterized by different secondary structures using 10 semiempirical quantum chemical methods, namely MNDO [19], AM1 [21], PM3 [22], OM1 [30], OM2 [31], OM2-D$_1$, OM2-D$_2$, OM3 [32], OM3-D$_1$, and OM3-D$_2$. The 28 proteins were selected from a benchmark set used in previous work [170] by eliminating those containing sulfur, since parameters for this element are not available for the OMx-type methods at present. The 28 proteins can be classified according to their dominant secondary structural element, namely α-helix, β-strand, or random coil (see Figure 5.1). However, there are only two proteins of β-strand type. In order to have sufficient samples for the statistics of the benchmark calculations, the α-helix and β-strand proteins are treated as a single group in the following discussion. The primary noncovalent interaction among the residues in this first group with dominant α-helix or β-strand structures is hydrogen bonding, whereas it is dispersion in the second group with somewhat unfolded main chains.

The quality of the protein structures predicted by the semiempirical calculations was evaluated in terms of the structural criteria for the backbone and the side chains by using the PROCHECK program [171]. As depicted in Figure 5.2, the criteria cover all covalent bond lengths and bond angles for the residues along the main chains [172]. Moreover, all the dihedral angles (ϕ for $\mathrm{C_{i\text{-}1}{-}N_i{-}C_i^\alpha{-}C_i}$, ψ for $\mathrm{N_i{-}C_i^\alpha{-}C_i{-}N_{i+1}}$, and ω for $\mathrm{C_i^\alpha{-}C_i{-}N_{i+1}{-}C_{i+1}^\alpha}$) of the backbone and the first side chain dihedral angle (χ_1 for $\mathrm{N_i{-}C_i^\alpha{-}C_i^\beta{-}C_i^\gamma}$) were also included [173, 174]. It must be emphasized that the default geometric data employed in PROCHECK, instead of the experimental structures, were used as references to evaluate the quality and errors of the protein structures optimized by the semiempirical calculations. It should be noted in this context that the structures of the 28 proteins in this set were determined by several different experimental techniques, e.g. X-ray diffraction, solution NMR (nuclear magnetic resonance) and solid state NMR, and hence the quality of the experimental struc-

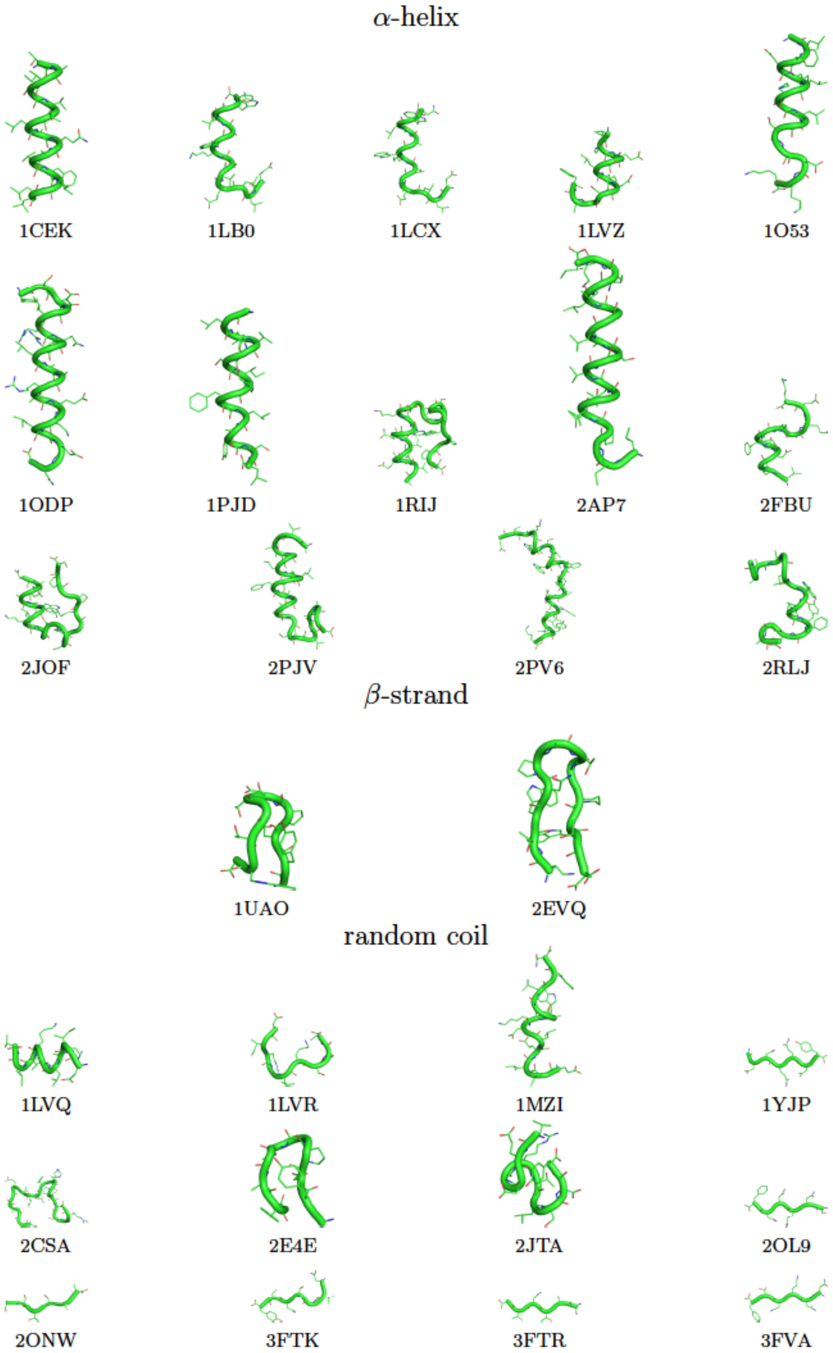


Figure 5.1: The benchmark set of proteins.

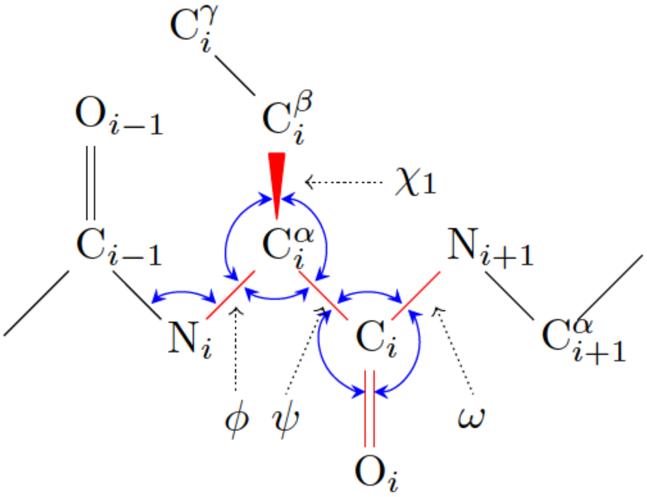


Figure 5.2: The structural criteria for assessing the quality of the protein structures. The covalent bonds are colored in red. The bond angles are denoted by blue arc arrows. The backbone dihedral angles are indicated by ϕ, ψ, and ω. χ_1 is the first dihedral angle of the side chain.

tures is not guaranteed to be the same for all proteins, e.g. the structure resolved by one experiment might be more accurate than others [106]. By contrast, the stereochemical parameters, i.e. bond lengths, bond angles, and dihedral angles, used in the PROCHECK program are derived from statistics on protein coordinates obtained from high-resolution X-ray crystallography [172,174]. Therefore it seems better to use the reference values from PROCHECK as the “standards”.

The experimental structures of our protein set were also compared with the PROCHECK reference values so that their quality can be assessed with respect to other protein structures solved by more reliable experiments. Last but not least, we also considered the protein structures optimized by other theoretical methods, i.e. the AMBER ff03 force field (AMBER) [115], the restricted Hartree-Fock (HF) method, and the ωPBEh density functional theory [175], using the 6-31G basis set in the latter two cases. The optmized structures were extracted from the original paper [170] and the relevant stereochemical quantities were determined using the PROCHECK program to allow for a comprehensive comparison of the protein structures obtained from the semiempirical quantum chemical methods and the other theoretical approaches.

Usually, there are 5 relevant covalent bonds and 7 relevant bond angles per residue in a protein (see Figure 5.2). The bond lengths and bond angles are classified in PROCHECK according to the type of amino acids. The reference data and their standard deviations are given in Tables 5.1 and 5.2. The mean absolute errors (MAEs) for the bond lengths and the bond angles in the backbone determined by experiments and different theoretical calculations averaged over each group of the protein test set are listed in Table 5.3. In general, the experimental values show the best agreement with the PROCHECK reference data in both protein groups. They are followed by the AMBER, HF, and DFT calculations, for which the deviations are roughly similar, i.e. 0.01 Å for the bond lengths and ~ 2 degree for the bond angles. The bond angles predicted by

Table 5.1: The classification of the main-chain covalent bonds in a protein, the reference values (x_0, in Å), and the standard deviations (σ, in Å) employed in the PROCHECK program.

notation	bond	description		x_0	σ
b_1	C–N	C-NH1	(except Pro)	1.33	0.01
b_2		C-N	(Pro)	1.34	0.02
b_3	C–O	C-O		1.23	0.02
b_4	CA–C	CH1E-C	(except Gly)	1.52	0.02
b_5		CH2G*-C	(Gly)	1.52	0.02
b_6	CA–CB	CH1E-CH3E	(Ala)	1.52	0.03
b_7		CH1E-CH1E	(Ile, Thr, Val)	1.54	0.03
b_8		CH1E-CH2E	(the rest)	1.53	0.02
b_9	N–CA	NH1-CH1E	(except Gly, Pro)	1.46	0.02
b_{10}		NH1-CH2G*	(Gly)	1.45	0.02
b_{11}		N-CH1E	(Pro)	1.47	0.01

all semiempirical methods are of about the same quality as the other theoretical methods, but the deviations for the bond lengths are somewhat larger. On average, the deviations for the OMx-based methods are slightly smaller, by ~ 0.01 Å, than those obtained with the MNDO-type methods. The MAEs of all types of covalent bonds for the experimental and theoretical methods are summarized in Tables 5.4 and 5.5. The worst errors for all semiempirical methods are due to the elongated (by about 0.05 to 0.08 Å) amide C−N bonds. This bond seems quite sensitive to the environment of the molecule. For example, the experimental peptide bond length of N-methylacetamide is 1.386 Å in the gas phase [176], but is decreased to 1.33 Å in the crystalline state [177]. The default values in PROCHECK are taken from the X-ray crystal structures of proteins. By contrast, semiempirical methods are mostly parameterized with respect to experimental gas-phase data, and the geometry optimizations were carried out for the isolated proteins. This complicates the assessment of the quality of the theoretical predictions for the amide bond length.

The Ramachandran map of $[\phi, \psi]$ combinations in proteins [178] is divided into four regions by the PROCHECK program: most favored ($P^1_{\phi,\psi}$), additionally allowed ($P^2_{\phi,\psi}$), generously allowed ($P^3_{\phi,\psi}$), and disallowed ($P^4_{\phi,\psi}$). The average distributions of the regions for the benchmark proteins are given in Table 5.6. In general, much higher $P^1_{\phi,\psi}$ values are observed for the first group of proteins with more regular secondary structures than for the second group of disordered polypeptide chains. As required by the PROCHECK program, the value of $P^1_{\phi,\psi}$ is expected to be higher than 90% for a well-resolved protein structure. This is a very strict requirement, which is not satisfied on average by the protein structures determined either from experiment or from theoretical calculations (see Table 5.6). The experimental structures show the best $P^1_{\phi,\psi}$ values, i.e. 81% and 68% for the first and second group, respectively. In most cases, the AMBER, HF, and DFT calculations retain higher $P^1_{\phi,\psi}$ and lower $P^4_{\phi,\psi}$ values than the semiempirical methods, which tend to predict higher distributions in the additionally allowed region. Overall MNDO and OM1 seem to be the best among the semiempirical methods for the $[\phi, \psi]$-distribution.

Table 5.2: The classification of the main-chain bond angles in a protein, the reference values (x_0, in degree), and the standard deviations (σ, in degree) employed in the PROCHECK program.

notation	angle	description		x_0	σ
a_1	CA–C–N	CH1E-C-NH1	(except Gly, Pro)	116.2	2.0
a_2		CH2G*-C-NH1	(Gly)	116.4	2.1
a_3		CH1E-C-N	(Pro)	116.9	1.5
a_4	O–C–N	O-C-NH1	(except Pro)	123.0	1.6
a_5		O-C-N	(Pro)	122.0	1.4
a_6	C–N–CA	C-NH1-CH1E	(except Gly, Pro)	121.7	1.8
a_7		C-NH1-CH2G*	(Gly)	120.6	1.7
a_8		C-N-CH1E	(Pro)	122.6	5.0
a_9	CA–C–O	CH1E-C-O	(except Gly)	120.8	1.7
a_{10}		CH2G*-C-O	(Gly)	120.8	2.1
a_{11}	CB–CA–C	CH3E-CH1E-C	(Ala)	110.5	1.5
a_{12}		CH1E-CH1E-C	(Ile, Thr, Val)	109.1	2.2
a_{13}		CH2E-CH1E-C	(the rest)	110.1	1.9
a_{14}	N–CA–C	NH1-CH1E-C	(except Gly, Pro)	111.2	2.8
a_{15}		NH1-CH2G*-C	(Gly)	112.5	2.9
a_{16}		N-CH1E-C	(Pro)	111.8	2.5
a_{17}	N–CA–CB	NH1-CH1E-CH3E	(Ala)	110.4	1.5
a_{18}		NH1-CH1E-CH1E	(Ile, Thr, Val)	111.5	1.7
a_{19}		N-CH1E-CH2E	(Pro)	103.0	1.1
a_{20}		NH1-CH1E-CH2E	(the rest)	110.5	1.7

Table 5.3: Mean absolute errors averaged over all bond lengths (MAE$_b$, in Å) and bond angles (MAE$_a$, in degree), respectively, in the protein main chains determined by experiments (Expt.) and various theoretical methods.

method	α-helix and β-strand		random coil	
	MAE$_b$	MAE$_a$	MAE$_b$	MAE$_a$
Expt.	0.008	1.03	0.013	1.26
AMBER	0.011	1.84	0.011	1.92
HF	0.008	1.63	0.010	2.09
DFT	0.010	1.58	0.013	2.24
MNDO	0.033	2.44	0.036	2.59
AM1	0.020	1.98	0.020	2.01
PM3	0.030	2.05	0.032	1.98
OM1	0.022	2.01	0.027	2.23
OM2	0.020	2.18	0.020	2.28
OM2-D$_1$	0.019	2.25	0.021	2.63
OM2-D$_2$	0.020	2.23	0.020	2.50
OM3	0.026	1.92	0.028	2.11
OM3-D$_1$	0.025	1.91	0.028	2.25
OM3-D$_2$	0.025	1.86	0.029	2.23

Table 5.4: Mean absolute errors (in Å) for all types of bond lengths determined by experiments (Expt.) and different levels of theoretical calculations. The statistics refer to the first group of proteins with dominant α-helix and β-strand structures in the main chain.

	Expt.	AMBER	HF	DFT	MNDO	AM1	PM3	OM1	OM2	OM2-D$_1$	OM2-D$_2$	OM3	OM3-D$_1$	OM3-D$_2$
b_1	0.008	0.009	0.014	0.020	0.080	0.046	0.088	0.044	0.039	0.039	0.038	0.064	0.061	0.061
b_2	0.010	0.010	0.004	0.009	0.081	0.037	0.085	0.036	0.035	0.035	0.035	0.061	0.057	0.058
b_3	0.006	0.004	0.010	0.027	0.005	0.021	0.009	0.022	0.021	0.020	0.020	0.027	0.028	0.027
b_4	0.006	0.014	0.005	0.006	0.035	0.025	0.016	0.040	0.023	0.022	0.020	0.038	0.036	0.036
b_5	0.009	0.013	0.005	0.007	0.031	0.023	0.012	0.035	0.023	0.022	0.023	0.034	0.035	0.034
b_6	0.007	0.015	0.010	0.007	0.028	0.006	0.003	0.008	0.004	0.004	0.003	0.006	0.006	0.004
b_7	0.008	0.008	0.011	0.007	0.039	0.009	0.011	0.021	0.009	0.009	0.010	0.019	0.016	0.016
b_8	0.006	0.010	0.011	0.009	0.033	0.010	0.009	0.015	0.007	0.007	0.007	0.012	0.011	0.011
b_9	0.010	0.013	0.008	0.008	0.010	0.016	0.033	0.009	0.018	0.019	0.018	0.010	0.010	0.010
b_{10}	0.014	0.018	0.005	0.007	0.009	0.019	0.030	0.010	0.021	0.020	0.022	0.006	0.006	0.006
b_{11}	0.004	0.004	0.005	0.005	0.013	0.008	0.032	0.003	0.015	0.018	0.018	0.010	0.011	0.011

Table 5.5: Mean absolute errors (in Å) for all types of bond lengths determined by experiments (Expt.) and different levels of theoretical calculations. The statistics refer to the second group of proteins with dominant random coil structures in the main chain.

	Expt.	AMBER	HF	DFT	MNDO	AM1	PM3	OM1	OM2	OM2-D$_1$	OM2-D$_2$	OM3	OM3-D$_1$	OM3-D$_2$
b_1	0.010	0.007	0.013	0.018	0.078	0.047	0.090	0.042	0.038	0.047	0.038	0.063	0.061	0.061
b_2	0.004	0.010	0.006	0.013	0.086	0.035	0.088	0.040	0.030	0.035	0.038	0.055	0.053	0.063
b_3	0.004	0.005	0.010	0.028	0.006	0.020	0.012	0.020	0.021	0.022	0.022	0.027	0.028	0.030
b_4	0.015	0.012	0.006	0.009	0.037	0.028	0.018	0.044	0.025	0.023	0.023	0.040	0.038	0.038
b_5	0.004	0.014	0.009	0.014	0.036	0.023	0.014	0.041	0.024	0.028	0.032	0.034	0.034	0.035
b_6	0.039	0.017	0.011	0.007	0.027	0.006	0.004	0.008	0.004	0.005	0.004	0.007	0.009	0.008
b_7	0.023	0.007	0.011	0.011	0.043	0.013	0.016	0.024	0.013	0.013	0.013	0.023	0.020	0.022
b_8	0.014	0.012	0.011	0.010	0.034	0.009	0.009	0.018	0.008	0.009	0.008	0.015	0.015	0.015
b_9	0.019	0.014	0.010	0.009	0.015	0.015	0.033	0.034	0.017	0.018	0.017	0.028	0.031	0.031
b_{10}	0.014	0.020	0.021	0.018	0.018	0.019	0.034	0.016	0.018	0.019	0.016	0.008	0.009	0.008
b_{11}	0.001	0.004	0.002	0.001	0.015	0.008	0.033	0.006	0.015	0.015	0.015	0.008	0.010	0.004

Table 5.6: Statistics (in %) of the $[\phi, \psi]$-distribution in the most favored ($P^1_{\phi,\psi}$), additionally allowed ($P^2_{\phi,\psi}$), generously allowed ($P^3_{\phi,\psi}$), and disallowed ($P^4_{\phi,\psi}$) regions of the Ramachandran maps for the protein structures determined by experiments (Expt.) and theoretical calculations at different levels.

(a) proteins dominated by α-helix or β-strand

method	$P^1_{\phi,\psi}$	$P^2_{\phi,\psi}$	$P^3_{\phi,\psi}$	$P^4_{\phi,\psi}$
Expt.	80.7	16.8	1.7	0.8
AMBER	77.9	17.6	2.8	1.7
HF	81.5	15.6	2.3	0.7
DFT	78.2	19.1	2.7	0.0
MNDO	69.5	28.2	0.8	1.5
AM1	60.2	35.7	3.1	1.0
PM3	59.9	36.5	2.6	1.0
OM1	72.7	19.7	4.8	2.8
OM2	63.5	29.3	4.5	2.8
OM2-D$_1$	66.4	25.5	5.8	2.2
OM2-D$_2$	65.2	29.0	3.6	2.2
OM3	65.1	29.3	3.3	2.3
OM3-D$_1$	69.6	22.5	3.9	4.0
OM3-D$_2$	67.8	25.1	3.7	3.5

(b) proteins dominated by random coil

method	$P^1_{\phi,\psi}$	$P^2_{\phi,\psi}$	$P^3_{\phi,\psi}$	$P^4_{\phi,\psi}$
Expt.	67.5	25.8	4.7	2.1
AMBER	57.5	32.4	7.5	2.5
HF	67.4	23.4	6.4	2.7
DFT	61.8	26.7	10.0	1.5
MNDO	60.7	34.7	2.5	2.1
AM1	44.3	42.8	9.4	3.5
PM3	48.1	41.9	8.1	1.9
OM1	52.5	39.5	3.7	4.2
OM2	42.0	46.1	6.4	5.5
OM2-D$_1$	42.4	46.1	5.4	6.1
OM2-D$_2$	31.2	49.8	13.4	5.5
OM3	46.2	43.4	4.9	5.5
OM3-D$_1$	42.1	41.2	11.5	5.2
OM3-D$_2$	36.7	50.2	9.4	3.6

Table 5.7: The default values ($\bar{x}$, in degree) of the dihedral angles in proteins and their standard deviations (σ_x, in degree) defined in the PROCHECK program.

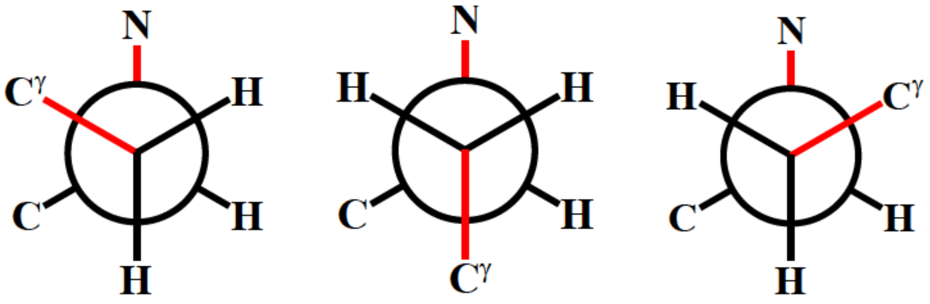
	ω	χ_1 (g^-)	χ_1 $(trans)$	χ_1 (g^+)
$\bar{x}$	180.0	64.1	183.6	-66.7
σ_x	5.8	15.7	16.8	15.0

Table 5.8: Average value ($\bar{\omega}$, in degree), mean absolute error (MAE, in degree), and root-mean-square deviation (RMSD, in degree) for the peptide bond dihedral angle of the protein structures determined by experiments (Expt.) and different theoretical calculations.

method	α-helix and β-strand			random coil		
	$\bar{\omega}$	MAE	RMSD	$\bar{\omega}$	MAE	RMSD
Expt.	179.5	1.6	3.3	179.0	6.4	12.7
AMBER	175.4	7.6	10.5	179.4	11.8	15.7
HF	179.3	6.1	8.4	180.6	9.1	12.4
DFT	178.7	6.1	8.7	179.8	10.5	13.6
MNDO	183.0	12.6	19.7	177.5	15.0	18.0
AM1	176.4	9.8	16.3	175.0	14.8	26.5
PM3	177.0	16.0	21.7	177.5	22.7	34.4
OM1	180.8	8.2	11.8	179.9	12.8	16.2
OM2	179.0	8.8	12.0	177.1	13.7	18.5
OM2-D$_1$	177.1	11.0	14.8	176.9	15.8	20.5
OM2-D$_2$	177.6	10.6	14.0	177.6	14.4	18.9
OM3	178.4	7.7	11.0	177.1	11.8	16.8
OM3-D$_1$	178.8	9.1	12.5	178.3	12.5	17.5
OM3-D$_2$	177.1	9.4	12.6	177.1	12.5	16.5

The peptide bonds are expected to be planar owing to the partial double bond character arising from two dominant resonance structures [179]. Moreover, the *trans* configuration is overwhelmingly preferred due to the much smaller steric hindrance between the groups attached to the C$^\alpha$ atoms. Hence the dihedral angle of the peptide bond (ω) in proteins is mostly 180°, but small deviations may occasionally be encountered according to the statistics [174]. The average values of ω and the errors for the experimental structures and the theoretically optimized structures are given in Table 5.8. Evidently, the results from experiments are generally closest to the predefined PROCHECK reference values. The dihedral angles predicted by the AMBER, HF, and DFT calculations are less accurate, and the RMSDs exceed the standard deviations given in PROCHECK (see Table 5.7). All semiempirical methods produce still larger deviations from planarity. This problem has already been reported in earlier studies [180,181]. Nevertheless, slight improvements can be found for the OMx methods over MNDO, AM1, and PM3. Since the molecular mechanics corrections for the peptide bond in the MNDO-type methods were deliberately switched off, PM3 gives the worst deviations [22,23]. Finally we note that the dispersion corrections included in the OM2-D and OM3-D methods (see Section 4.4) may deteriorate the calculated ω values because of the added small dispersive attractions between the two nearest C$^\alpha$ atoms surrounding the peptide bond.

Figure 5.3 depicts the three rotational conformers, two gauche rotamers (g^- and g^+) and one *trans* isomer, for the C$^\alpha$–C$^\beta$ bond in proteins. The different conformations are distinguished by the dihedral angle χ_1 of N$_i$–C$_i^\alpha$–C$_i^\beta$–C$_i^\gamma$ (see Figure 5.2). The expected values of χ_1 for the three configurations and their



(a) g^- (b) *trans* (c) g^+

Figure 5.3: The Newman projection for the three conformations of the $C^{\alpha}-C^{\beta}$ bond in protein. The front $C^{\beta}-C^{\gamma}$ bond and the back $C^{\alpha}-N$ bond of the dihedral angle χ_1 are colored in red.

standard deviations obtained from statistics of the X-ray protein structures are given in Table 5.7. In contrast to ω, large deviations, e.g. 15° to 17°, are often found for χ_1 in the experimental structures due to the diversity of the vicinal groups and the small rotational barrier, which implies rather shallow potential energy surfaces [173]. The average χ_1 and the corresponding errors for the three conformations in the protein set are summarized for all methods in Tables 5.9 to 5.11. In most cases, the χ_1 values for the first group of proteins are more clustered around the predefined values than the second group. The RMSDs of χ_1 for all the methods are more or less in the same range as the standard deviations specified in the PROCHECK program (see Table 5.7). In this regard the semiempirical methods are of similar quality as the other theoretical models.

In summary, geometry optimizations were performed for 28 proteins characterized by α-helix, β-sheet, and random coil secondary structures using 10 semiempirical methods. The quality of the predicted protein structures was assessed in terms of the geometric criteria defined in the PROCHECK program. Furthermore, the semiempirical results were also compared with the protein structures determined by experiments and other theoretical methods, i.e. molecular mechanics force field, Hartree-Fock, and DFT calculations. Overall the semiempirical methods produce qualitatively reasonable geometries, especially for the proteins with more regular secondary structures like α-helix and β-strand. There are, however, some shortcomings of the protein structures obtained from the semiempirical methods. For example, the $[\phi, \psi]$-distributions are less apt to the most favored region of the Ramachandran plot and the peptide bonds tend to be slightly nonplanar. We are confident that these quantitative deficiencies can be alleviated in the future development of improved semiempirical quantum chemical methods.

Table 5.9: Average value ($\bar{\chi}_1$, in degree), mean absolute error (MAE, in degree), and root-mean-square deviation (RMSD, in degree) for the side-chain dihedral angle of the g^- conformation in the protein structures determined by experiments (Expt.) and different theoretical calculations.

method	α-helix and β-strand			random coil		
	$\bar{\chi}_1$	MAE	RMSD	$\bar{\chi}_1$	MAE	RMSD
Expt.	64.2	12.9	18.5	60.7	16.7	22.9
AMBER	67.1	10.3	13.3	56.1	10.9	12.8
HF	65.9	10.3	14.7	54.9	13.5	17.7
DFT	62.4	10.4	12.7	54.3	15.4	18.4
MNDO	61.7	9.0	11.4	54.2	15.0	17.5
AM1	57.2	15.5	19.8	48.3	17.3	21.8
PM3	61.0	14.2	18.1	53.4	11.5	15.6
OM1	61.1	11.6	17.5	58.2	19.1	22.6
OM2	64.6	11.5	15.6	51.6	15.7	18.7
OM2-D$_1$	62.7	10.1	12.2	47.8	22.7	25.7
OM2-D$_2$	59.7	11.9	16.7	50.5	18.8	23.9
OM3	59.2	12.1	16.4	57.2	12.2	14.1
OM3-D$_1$	64.2	13.9	19.2	54.8	18.0	20.8
OM3-D$_2$	65.4	12.4	17.1	56.3	15.7	17.5

Table 5.10: Average value ($\bar{\chi}_1$, in degree), mean absolute error (MAE, in degree), and root-mean-square deviation (RMSD, in degree) for the side-chain dihedral angle of the *trans* conformation in the protein structures determined by experiments (Expt.) and different theoretical calculations.

method	α-helix and β-strand			random coil		
	$\bar{\chi}_1$	MAE	RMSD	$\bar{\chi}_1$	MAE	RMSD
Expt.	192.7	15.6	21.8	190.9	12.7	17.3
AMBER	188.6	8.8	13.3	185.6	9.8	14.0
HF	189.4	11.5	16.0	189.3	11.0	13.3
DFT	189.0	14.0	19.2	189.1	12.0	14.4
MNDO	189.9	10.1	13.7	191.0	12.4	15.6
AM1	194.6	15.0	19.6	196.3	13.5	18.2
PM3	194.0	13.9	17.3	200.7	20.2	24.2
OM1	187.6	11.4	15.5	187.2	16.1	21.0
OM2	187.3	11.8	15.4	187.7	13.1	16.4
OM2-D$_1$	181.1	12.5	16.8	181.8	13.4	18.4
OM2-D$_2$	181.0	13.0	16.9	182.5	14.9	20.0
OM3	190.7	12.9	16.8	191.6	14.6	19.3
OM3-D$_1$	187.7	13.4	17.4	187.2	11.8	15.7
OM3-D$_2$	184.4	13.0	17.3	185.1	14.9	18.8

Table 5.11: Average value ($\bar{\chi}_1$, in degree), mean absolute error (MAE, in degree), and root-mean-square deviation (RMSD, in degree) for the-side chain dihedral angle of the g^+ conformation in the protein structures determined by experiments (Expt.) and different theoretical calculations.

method	α-helix and β-strand			random coil		
	$\bar{\chi}_1$	MAE	RMSD	$\bar{\chi}_1$	MAE	RMSD
Expt.	−72.2	17.1	21.4	−64.2	15.4	21.9
AMBER	−66.8	13.1	17.6	−57.9	11.6	16.1
HF	−69.7	10.0	15.3	−59.1	10.6	12.7
DFT	−68.3	10.2	14.7	−58.8	12.5	15.3
MNDO	−70.7	8.1	11.6	−65.7	10.9	13.4
AM1	−67.4	9.3	13.2	−65.1	15.3	20.4
PM3	−72.2	12.0	15.7	−69.2	14.3	18.9
OM1	−64.6	7.3	10.5	−61.8	10.8	13.5
OM2	−71.0	9.9	14.0	−66.2	14.4	18.2
OM2-D_1	−64.1	13.0	16.9	−63.8	16.5	19.6
OM2-D_2	−69.0	12.3	17.1	−61.4	16.2	19.6
OM3	−72.6	10.2	14.4	−62.1	13.2	17.3
OM3-D_1	−67.6	9.2	12.5	−60.5	15.9	19.7
OM3-D_2	−68.0	9.7	12.8	−61.6	15.2	19.1

Chapter 6

Concluding Remarks

The central goal of the investigations conducted in this dissertation is rather simple: to make semiempirical quantum chemistry faster and more accurate (at least for some specific applications) by taking advantage of high-performance computers.

First, a profile-guided optimization of the MNDO99 program was carried out on heterogeneous CPU-GPU computing platforms (see Chapter 3 and Appendices A and B). Systematic tests on fullerenes, water balls, and proteins showed that semiempirical MNDO, AM1, PM3, and OMx ($x = 1$, 2, and 3) calculations on multiple CPU-GPU platforms can be accelerated by one order of magnitude over the serial executions on a single CPU core for large systems, e.g. a protein with 100 residues, 1842 atoms, and 4446 basis functions. By contrast, the speedups were always less than fourfold in the CPU-only calculations, no matter how many parallel processors were deployed. A detailed analysis indicated that this limited gain in performance can be attributed to the inadequate memory bandwidth of the CPU-only architecture. In other words, the parallel CPUs have to wait for the data due to the "traffic jam" from memory to processors. Hence the arithmetic power of the CPUs is highly underutilized in the CPU-only cases.

Second, a parallel parameterization program (PPP) was developed for symmetric multiprocessor computers (see Chapter 4). Because a large number of function evaluations is usually involved in the parameterization of a semiempirical method, the parallel implementation on multiple CPUs permits more efficient and more thorough optimization of the parameters within given boundaries, e.g. ±10% of the initial values. The use of PPP was demonstrated by three studies: the specific parameterization of the AM1 Hamiltonian for the reactant, transition state, and product of an enzyme-catalyzed hydride transfer reaction (see Appendix C); the reparameterization of MNDO, AM1, PM3, and OMx ($x = 1$, 2, and 3) for hydrogen bonding and proton transfer in water clusters (see Appendix D); and the optimization of the parameters of the dispersion corrections in the OM2-D and OM3-D methods for noncovalent complexes (see Section 4.4).

Finally, benchmark calculations with geometry optimizations for a set of 28 proteins were performed by using 10 semiempirical methods, i.e. MNDO, AM1, PM3, OMx ($x = 1$, 2, and 3), and OM2-D and OM3-D with two types of damping functions (see Chapter 5). The optimized protein structures were compared with those determined by experiments and by other theoretical approaches, with regard to the geometries and stereochemical properties of the backbone and side chains. In general, the semiempirical methods were found to produce qualitatively reasonable geometries, especially for proteins with dominant α-helix and β-strand secondary structures. However, the protein structures obtained at the semiempirical level are not as good as those obtained from experiments, from state-of-the-art molecular mechanics force fields, or from first-principles Hartree-Fock and DFT calculations using the 6-31G basis set. For instance, the semiempirical protein structures are less prominently clustered in the most favored $[\psi, \phi]$-region of the Ramachandran diagram, and there is some slight pyramidalization at the nitrogen atom of the peptide bond. We believe that these quantitative deficiencies can be alleviated in future developments and that semiempirical quantum chemical methods will then become even more valuable tools for studying complicated biochemical molecules and their reactions.

Bibliography

[1] W. Thiel. Semiempirical quantum-chemical methods. *WIREs Comput. Mol. Sci.* **2013**, DOI: 10.1002/wcms.1161.

[2] E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Phys. Rev.* **1926**, *28*, 1049 – 1070.

[3] P. A. M. Dirac. Quantum mechanics of many-electron systems. *Proc. R. Soc. Lond., A* **1929**, *123*, 714 – 733.

[4] E. Hückel. Quantentheoretische Beiträge zum Benzolproblem. *Z. Phys.* **1931**, *70*, 204 – 286.

[5] R. Pariser, R. G. Parr. A semi-empirical theory of the electronic spectra and electronic structure of complex unsaturated molecules. I. *J. Chem. Phys.* **1953**, *21*, 466 – 471.

[6] J. A. Pople. Electron interaction in unsaturated hydrocarbons. *Trans. Faraday Soc.* **1953**, *49*, 1375 – 1385.

[7] R. Hoffmann. An extended Hückel theory. I. Hydrocarbons. *J. Chem. Phys.* **1963**, *39*, 1397 – 1412.

[8] R. Hoffmann. Extended Hückel theory. II. σ orbitals in the azines. *J. Chem. Phys.* **1964**, *40*, 2745 – 2745.

[9] M. Wolfsberg, L. Helmholz. The spectra and electronic structure of the tetrahedral ions MnO_4^-, CrO_4^{--}, and ClO_4^-. *J. Chem. Phys.* **1952**, *20*, 837 – 843.

[10] J. A. Pople, D. P. Santry, G. A. Segal. Approximate self-consistent molecular orbital theory. I. Invariant procedures. *J. Chem. Phys.* **1965**, *43*, S129 – S135.

[11] J. A. Pople, D. L. Beveridge, P. A. Dobosh. Approximate self-consistent molecular-orbital theory. V. Intermediate neglect of differential overlap. *J. Chem. Phys.* **1967**, *47*, 2026 – 2033.

[12] R. D. Brown, K. R. Roby. Approximate molecular orbital theory for inorganic molecules. *Theor. Chim. Acta* **1970**, *16*, 175 – 193.

[13] K. R. Roby. On the justifiability of neglect of differential overlap molecular orbital methods. *Chem. Phys. Lett.* **1971**, *11*, 6 – 10.

[14] K. R. Roby. Fundamentals of an orthonormal basis set molecular orbital theory. *Chem. Phys. Lett.* **1972**, *12*, 579 – 582.

[15] D. B. Cook, P. C. Hollis, R. McWeeny. Approximate ab initio calculations on polyatomic molecules. *Mol. Phys.* **1967**, *13*, 553 – 571.

[16] G. S. Chandler, F. E. Grader. A re-examination of the justification of neglect of differential overlap approximations in terms of a power series expansion in S. *Theor. Chim. Acta* **1980**, *54*, 131 – 144.

[17] P.-O. Löwdin. On the non-orthogonality problem connected with the use of atomic wave functions in the theory of molecules and crystals. *J. Chem. Phys.* **1950**, *18*, 365 – 375.

[18] M. J. S. Dewar, W. Thiel. A semiempirical model for the two-center repulsion integrals in the NDDO approximation. *Theor. Chim. Acta* **1977**, *46*, 89 – 104.

[19] M. J. S. Dewar, W. Thiel. Ground states of molecules. 38. The MNDO method. Approximations and parameters. *J. Am. Chem. Soc.* **1977**, *99*, 4899 – 4907.

[20] M. J. S. Dewar, W. Thiel. Ground states of molecules. 39. MNDO results for molecules containing hydrogen, carbon, nitrogen, and oxygen. *J. Am. Chem. Soc.* **1977**, *99*, 4907 – 4917.

[21] M. J. S. Dewar, E. G. Zoebisch, E. F. Healy, J. J. P. Stewart. Development and use of quantum mechanical molecular models. 76. AM1: A new general purpose quantum mechanical molecular model. *J. Am. Chem. Soc.* **1985**, *107*, 3902 – 3909.

[22] J. J. P. Stewart. Optimization of parameters for semiempirical methods I. Method. *J. Comput. Chem.* **1989**, *10*, 209 – 220.

[23] J. J. P. Stewart. Optimization of parameters for semiempirical methods II. Applications. *J. Comput. Chem.* **1989**, *10*, 221 – 264.

[24] W. Thiel. Semiempirical methods: Current status and perspectives. *Tetrahedron* **1988**, *44*, 7393 – 7408.

[25] W. Thiel, D. G. Green, *The MNDO94 Code: Parallelization of a Semiempirical Qauntum-chemical Program* in *Methods and Techniques in Computational Chemistry: METECC-95*, E. Clementi, G. Corongiu (Eds.), STEF, Cagliari, **1995**, pp. 141 – 168.

[26] W. Thiel. Perspectives on semiempirical molecular orbital theory. *Adv. Chem. Phys.* **1996**, *93*, 703 – 757.

[27] W. Thiel. Computational methods for large molecules. *J. Mol. Str. Theochem* **1997**, *398 – 399*, 1 – 6.

[28] W. Thiel, *Semiempirical Methods* in *Modern Methods and Algorithms of Quantum Chemistry Proceedings, Second Edition*, J. Grotendorst (Ed.), John von Neumann Institute for Computing, Jülich, **2000**, pp. 261 – 283.

[29] W. Thiel, *Semiempirical Quantum-Chemical Methods in Computational Chemistry* in *Theory and Applications of Computational Chemistry: The First Forty Years*, C. E. Dykstra, G. Frenking, K. S. Kim, G. E. Scuseria (Eds.), Elsevier, Amsterdam, **2005**, pp. 559 – 580.

[30] M. Kolb, W. Thiel. Beyond the MNDO model: Methodical considerations and numerical results. *J. Comput. Chem.* **1993**, *14*, 775 – 789.

[31] W. Weber, W. Thiel. Orthogonalization corrections for semiempirical methods. *Theor. Chem. Acc.* **2000**, *103*, 495 – 506.

[32] M. Scholten, *Semiempirische Verfahren mit Orthogonalisierungskorrekturen: Die OM3 Methode*, Ph.D. thesis, Universität Düsseldorf, Düsseldorf, **2003**.

[33] N. Otte, M. Scholten, W. Thiel. Looking at self-consistent-charge density functional tight binding from a semiempirical perspective. *J. Phys. Chem. A* **2007**, *111*, 5751 – 5755.

[34] M. Korth, W. Thiel. Benchmarking semiempirical methods for thermochemistry, kinetics, and noncovalent interactions: OMx methods are almost as accurate and robust as DFT-GGA methods for organic molecules. *J. Chem. Theory Comput.* **2011**, *7*, 2929 – 2936.

[35] M. R. Silva-Junior, W. Thiel. Benchmark of electronically excited states for semiempirical methods: MNDO, AM1, PM3, OM1, OM2, OM3, INDO/S, and INDO/S2. *J. Chem. Theory Comput.* **2010**, *6*, 1546 – 1564.

[36] J. D. Bolcer, R. B. Hermann, *The Development of Computational Chemistry in the United States* in *Reviews in Computational Chemistry, Vol. 5*, K. B. Lipkowitz, D. B. Boyd (Eds.), John Wiley & Sons, Inc., **1994**, pp. 1 – 63.

[37] S. F. Boys, G. B. Cook, C. M. Reeves, I. Shavitt. Automatic fundamental calculations of molecular structure. *Nature* **1956**, *178*, 1207–1209.

[38] J. A. Pople, *Quantum Chemical Models* in *Nobel Lectures in Chemistry (1996-2000)*, I. Grenthe (Ed.), World Scientific Publishing Co., Singapore, **2003**, pp. 246 – 260.

[39] D. B. Boyd, *Appendix: Compendium of Software for Molecular Modeling* in *Reviews in Computational Chemistry*, K. B. Lipkowitz, D. B. Boyd (Eds.), John Wiley & Sons, Inc., **1990**, pp. 383 – 392.

[40] P. K. Pearson, R. R. Lucchese, W. H. Miller, H. F. Schaefer III, *Theoretical Chemistry via Minicomputer* in *Minicomputers and Large Scale Computations*, P. Lykos (Ed.), American Chemical Society, **1977**, chapter 14, pp. 171 – 190.

[41] D. B. Boyd, *Quantum Chemistry Program Exchange, Facilitator of Theoretical and Computational Chemistry in Pre-Internet History* in *Pioneers of Quantum Chemistry*, T. Strom, A. K. Wilson (Eds.), American Chemical Society, **2013**, chapter 9, pp. 221 – 273.

[42] G. E. Moore. Cramming more components onto integrated circuits. *Electronics* **1965**, *38*, 114 – 117.

[43] J. J. Dongarra, A. J. van der Steen. High-performance computing systems: Status and outlook. *Acta Numer.* **2012**, *21*, 379 – 474.

[44] E. Clementi, G. Corongiu (Eds.), *Methods and Techniques in Computational Chemistry: METECC-95*, STEF, Cagliari, **1995**.

[45] A. R. Brodtkorb, C. Dyken, T. R. Hagen, J. M. Hjelmervik, O. O. Storaasli. State-of-the-art in heterogeneous computing. *Sci. Program.* **2010**, *18*, 1 – 33.

[46] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, T. Purcell. A survey of general-purpose computation on graphics hardware. *Comput. Graph. Forum* **2007**, *26*, 80 – 113.

[47] D. B. Kirk, W. M. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann, **2012**.

[48] J. E. Stone, D. J. Hardy, I. S. Ufimtsev, K. Schulten. GPU-accelerated molecular modeling coming of age. *J. Mol. Graph. Model.* **2010**, *29*, 116 – 125.

[49] R. M. Farber. Topical perspective on massive threading and parallelism. *J. Mol. Graph. Model.* **2011**, *30*, 82 – 89.

[50] A. G. Anderson, W. A. Goddard III, P. Schröder. Quantum Monte Carlo on graphical processing units. *Comput. Phys. Commun.* **2007**, *177*, 298 – 306.

[51] J. Kim, J. M. Rodgers, M. Athènes, B. Smit. Molecular Monte Carlo simulations using graphics processing units: To waste recycle or not? *J. Chem. Theory Comput.* **2011**, *7*, 3208 – 3222.

[52] K. Yasuda. Two-electron integral evaluation on the graphics processor unit. *J. Comput. Chem.* **2008**, *29*, 334 – 342.

[53] I. S. Ufimtsev, T. J. Martínez. Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. *J. Chem. Theory Comput.* **2008**, *4*, 222 – 231.

[54] A. Asadchev, V. Allada, J. Felder, B. M. Bode, M. S. Gordon, T. L. Windus. Uncontracted Rys quadrature implementation of up to g functions on graphical processing units. *J. Chem. Theory Comput.* **2010**, *6*, 696 – 704.

[55] K. A. Wilkinson, P. Sherwood, M. F. Guest, K. J. Naidoo. Acceleration of the GAMESS-UK electronic structure package on graphical processing units. *J. Comput. Chem.* **2011**, *32*, 2313 – 2318.

[56] Y. Miao, K. M. Merz. Acceleration of electron repulsion integral evaluation on graphics processing units via use of recurrence relations. *J. Chem. Theory Comput.* **2013**, *9*, 965 – 976.

[57] A. V. Titov, I. S. Ufimtsev, N. Luehr, T. J. Martínez. Generating efficient quantum chemistry codes for novel architectures. *J. Chem. Theory Comput.* **2013**, *9*, 213 – 221.

[58] K. Yasuda. Accelerating density functional calculations with graphics processing unit. *J. Chem. Theory Comput.* **2008**, *4*, 1230 – 1236.

[59] I. S. Ufimtsev, T. J. Martínez. Quantum chemistry on graphical processing units. 2. Direct self-consistent-field implementation. *J. Chem. Theory Comput.* **2009**, *5*, 1004 – 1015.

[60] I. S. Ufimtsev, T. J. Martínez. Quantum chemistry on graphical processing units. 3. Analytical energy gradients, geometry optimization, and first principles molecular dynamics. *J. Chem. Theory Comput.* **2009**, *5*, 2619 – 2628.

[61] I. S. Ufimtsev, T. J. Martínez. Graphical processing units for quantum chemistry. *Comput. Sci. Eng.* **2008**, *10*, 26 – 34.

[62] L. Genovese, M. Ospici, T. Deutsch, J.-F. Méhaut, A. Neelov, S. Goedecker. Density functional theory calculation on many-cores hybrid central processing unit-graphic processing unit architectures. *J. Chem. Phys.* **2009**, *131*, 034103 (8 pages).

[63] N. Luehr, I. S. Ufimtsev, T. J. Martínez. Dynamic precision for electron repulsion integral evaluation on graphical processing units (GPUs). *J. Chem. Theory Comput.* **2011**, *7*, 949 – 954.

[64] X. Andrade, L. Genovese, *Harnessing the Power of Graphic Processing Units* in *Fundamentals of Time-Dependent Density Functional Theory*, Vol. 837 in *Lecture Notes in Physics*, M. A. Marques, N. T. Maitra, F. M. Nogueira, E. Gross, A. Rubio (Eds.), Springer, **2012**, pp. 401 – 413.

[65] X. Andrade, A. Aspuru-Guzik. Real-space density functional theory on graphical processing units: Computational approach and comparison to Gaussian basis set methods. *J. Chem. Theory Comput.* **2013**, *9*, 4360 – 4373.

[66] C. M. Isborn, N. Luehr, I. S. Ufimtsev, T. J. Martínez. Excited-state electronic structure with configuration interaction singles and Tamm-Dancoff time-dependent density functional theory on graphical processing units. *J. Chem. Theory Comput.* **2011**, *7*, 1814 – 1823.

[67] L. Vogt, R. Olivares-Amaya, S. Kermes, Y. Shao, C. Amador-Bedolla, A. Aspuru-Guzik. Accelerating resolution-of-the-identity second-order Møller-Plesset quantum chemistry calculations with graphical processing units. *J. Phys. Chem. A* **2008**, *112*, 2049 – 2057.

[68] R. Olivares-Amaya, M. A. Watson, R. G. Edgar, L. Vogt, Y. Shao, A. Aspuru-Guzik. Accelerating correlated quantum chemistry calculations using graphical processing units and a mixed precision matrix multiplication library. *J. Chem. Theory Comput.* **2010**, *6*, 135 – 144.

[69] M. Watson, R. Olivares-Amaya, R. G. Edgar, A. Aspuru-Guzik. Accelerating correlated quantum chemistry calculations using graphical processing units. *Comput. Sci. Eng.* **2010**, *12*, 40 – 51.

[70] A. E. DePrince, J. R. Hammond. Coupled cluster theory on graphics processing units I. The coupled cluster doubles method. *J. Chem. Theory Comput.* **2011**, *7*, 1287 – 1295.

[71] W. Ma, S. Krishnamoorthy, O. Villa, K. Kowalski. GPU-based implementations of the noniterative regularized-CCSD(T) corrections: Applications to strongly correlated systems. *J. Chem. Theory Comput.* **2011**, *7*, 1316 – 1327.

[72] K. Bhaskaran-Nair, W. Ma, S. Krishnamoorthy, O. Villa, H. J. J. van Dam, E. Aprà, K. Kowalski. Noniterative multireference coupled cluster methods on heterogeneous CPU-GPU systems. *J. Chem. Theory Comput.* **2013**, *9*, 1949 – 1957.

[73] A. Asadchev, M. S. Gordon. Fast and flexible coupled cluster implementation. *J. Chem. Theory Comput.* **2013**, *9*, 3385 – 3392.

[74] X. Wu, A. Koslowski, W. Thiel. Semiempirical quantum chemical calculations accelerated on a hybrid multicore CPU-GPU computing platform. *J. Chem. Theory Comput.* **2012**, *8*, 2272 – 2281.

[75] J. D. C. Maia, G. A. Urquiza Carvalho, C. P. Mangueira, S. R. Santana, L. A. F. Cabral, G. B. Rocha. GPU linear algebra libraries and GPGPU programming for accelerating MOPAC semiempirical quantum chemistry calculations. *J. Chem. Theory Comput.* **2012**, *8*, 3072 – 3081.

[76] G. B. Rocha, R. O. Freire, A. M. Simas, J. J. P. Stewart. RM1: A reparameterization of AM1 for H, C, N, O, P, S, F, Cl, Br, and I. *J. Comp. Chem.* **2006**, *27*, 1101 – 1111.

[77] J. J. P. Stewart. Comparison of the accuracy of semiempirical and some DFT methods for predicting heats of formation. *J. Mol. Model.* **2004**, *10*, 6 – 12.

[78] J. J. P. Stewart. Optimization of parameters for semiempirical methods V: Modification of NDDO approximations and application to 70 elements. *J. Mol. Model.* **2007**, *13*, 1173 – 1213.

[79] J. J. P. Stewart. Optimization of parameters for semiempirical methods VI: More modifications to the NDDO approximations and re-optimization of parameters. *J. Mol. Model.* **2013**, *19*, 1 – 32.

[80] A. Gonzalez-Lafont, T. N. Truong, D. G. Truhlar. Direct dynamics calculations with NDDO (neglect of diatomic differential overlap) molecular orbital theory with specific reaction parameters. *J. Phys. Chem.* **1991**, *95*, 4618 – 4627.

[81] I. Rossi, D. G. Truhlar. Parameterization of NDDO wavefunctions using genetic algorithms. An evolutionary approach to parameterizing potential energy surfaces and direct dynamics calculations for organic reactions. *Chem. Phys. Lett.* **1995**, *233*, 231 – 236.

[82] M. C. Hutter, J. R. Reimers, N. S. Hush. Modeling the bacterial photosynthetic reaction center. 1. Magnesium parameters for the semiempirical am1 method developed using a genetic algorithm. *J. Phys. Chem. B* **1998**, *102*, 8080 – 8090.

[83] E. N. Brothers, K. M. Merz. Sodium parameters for AM1 and PM3 optimized using a modified genetic algorithm. *J. Phys. Chem. B* **2002**, *106*, 2779 – 2785.

[84] W. Thiel, *MNDO99 CVS Development Version*, Tech. Rep., Max-Planck-Institut für Kohlenforschung, Mülheim an der Ruhr, Germany, **2012**.

[85] M. J. S. Dewar, *The Molecular Orbital Theory of Organic Chemistry*, in *McGraw-Hill Series in Advanced Chemistry*, McGraw-Hill, **1969**.

[86] J. A. Pople, D. L. Beveridge, *Approximate Molecular Orbital Theory*, in *McGraw-Hill Series in Advanced Chemistry*, McGraw-Hill, **1970**.

[87] J. N. Murrell, A. J. Harget, *Semi-empirical Self-consistent-field Molecular Orbital Theory of Molecules*, Wiley-Interscience, **1972**.

[88] M. J. S. Dewar. Quantum organic chemistry. *Science* **1975**, *187*, 1037 – 1044.

[89] J. J. P. Stewart, *Semiempirical Molecular Orbital Methods* in *Reviews in Computational Chemistry*, K. B. Lipkowitz, D. B. Boyd (Eds.), John Wiley & Sons, Inc., **1990**, pp. 45 – 81.

[90] M. C. Zerner, *Semiempirical Molecular Orbital Methods* in *Reviews in Computational Chemistry*, K. B. Lipkowitz, D. B. Boyd (Eds.), John Wiley & Sons, Inc., **1991**, pp. 313 – 365.

[91] J. J. P. Stewart. MOPAC: A semiempirical molecular orbital program. *J. Comput. Aid. Mol. Des.* **1990**, *4*, 1 – 103.

[92] J. Almlöf, K. Faegri, K. Korsell. Principles for a direct SCF approach to LICAO-MO *ab-initio* calculations. *J. Comput. Chem.* **1982**, *3*, 385 – 399.

[93] R. G. Parr. A method for estimating electronic repulsion integrals over LCAO MO's in complex unsaturated molecules. *J. Chem. Phys.* **1952**, *20*, 1499.

[94] K. F. Freed. Is there a bridge between ab initio and semiempirical theories of valence? *Acc. Chem. Res.* **1983**, *16*, 137 – 144.

[95] P. Habitz, E. Clementi. A general program to compute two electron repulsion integrals. *Comput. Phys. Commun.* **1983**, *29*, 301 – 306.

[96] R. Sustmann, J. E. Williams, M. J. S. Dewar, L. C. Allen, P. v. R. Schleyer. Molecular orbital calculations on carbonium ions. II. Methyl, ethyl, and vinyl cations. The series $C_3H_7^+$. *J. Am. Chem. Soc.* **1969**, *91*, 5350 – 5357.

[97] R. B. Davidson, W. L. Jorgensen, L. C. Allen. Structural and energetic predictions for simple hydrocarbons from the NDDO [neglect of diatomic differential overlap] and CNDO [complete neglect of differential overlap] semiempirical molecular orbital methods. *J. Am. Chem. Soc.* **1970**, *92*, 749 – 753.

[98] P. Birner, H.-J. Köhler, C. Weiss. C–H acidity comparative CNDO/2 and NDDO calculations on the reactivity of azabenzenes. *Chem. Phys. Lett.* **1974**, *27*, 347 – 350.

[99] K. Y. Burstein, A. N. Isaev. MNDO calculations on hydrogen bonds. Modified function for core-core repulsion. *Theor. Chim. Acta* **1984**, *64*, 397 – 401.

[100] A. Goldblum. Improvement of the hydrogen bonding correction to MNDO for calculations of biochemical interest. *J. Comput. Chem.* **1987**, *8*, 835 – 849.

[101] M. I. Bernal-Uruchurtu, M. T. C. Martins-Costa, C. Millot, M. F. Ruiz-López. Improving description of hydrogen bonds at the semiempirical level: Water-water interactions as test case. *J. Comput. Chem.* **2000**, *21*, 572 – 581.

[102] M. I. Bernal-Uruchurtu, M. F. Ruiz-López. Basic ideas for the correction of semiempirical methods describing H-bonded systems. *Chem. Phys. Lett.* **2000**, *330*, 118 – 124.

[103] M. P. Repasky, J. Chandrasekhar, W. L. Jorgensen. PDDG/PM3 and PDDG/MNDO: Improved semiempirical methods. *J. Comput. Chem.* **2002**, *23*, 1601 – 1622.

[104] I. Tubert-Brohman, C. R. W. Guimarães, M. P. Repasky, W. L. Jorgensen. Extension of the PDDG/PM3 and PDDG/MNDO semiempirical molecular orbital methods to the halogens. *J. Comput. Chem.* **2004**, *25*, 138 – 150.

[105] K. Müller-Dethlefs, P. Hobza. Noncovalent interactions: A challenge for experiment and theory. *Chem. Rev.* **2000**, *100*, 143 – 168.

[106] G. E. Schulz, R. H. Schirmer, *Principles of Protein Structure*, in *Springer Advanced Texts in Chemistry*, Springer-Verlag, New York, **1979**.

[107] R. Eisenschitz, F. London. Über das Verhältnis der van der Waalsschen Kräfte zu den homöopolaren Bindungskräften. *Z. Phys.* **1930**, *60*, 491 – 527.

[108] A. J. Stone, *The Theory of Intermolecular Forces*, Clarendon Press, Oxford, **1997**.

[109] A. J. Stone. Intermolecular potentials. *Science* **2008**, *321*, 787 – 789.

[110] P. Hobza, H. L. Selzle, E. W. Schlag. Structure and properties of benzene-containing molecular clusters: Nonempirical *ab initio* calculations and experiments. *Chem. Rev.* **1994**, *94*, 1767 – 1785.

[111] P. Hobza, J. Šponer. Structure, energetics, and dynamics of the nucleic acid base pairs: Nonempirical *ab initio* calculations. *Chem. Rev.* **1999**, *99*, 3247 – 3276.

[112] K. E. Riley, M. Pitoňák, P. Jurečka, P. Hobza. Stabilization and structure calculations for noncovalent interactions in extended molecular systems based on wave function and density functional theories. *Chem. Rev.* **2010**, *110*, 5023 – 5063.

[113] P. Hobza, J. Šponer, T. Reschel. Density functional theory and molecular clusters. *J. Comput. Chem.* **1995**, *16*, 1315 – 1325.

[114] J. Černý, P. Hobza. The X3LYP extended density functional accurately describes H-bonding but fails completely for stacking. *Phys. Chem. Chem. Phys.* **2005**, *7*, 1624 – 1626.

[115] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, P. Kollman. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Comput. Phys. Commun.* **1995**, *91*, 1 – 41.

[116] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* **1983**, *4*, 187 – 217.

[117] J. E. Lennard-Jones. On the determination of molecular fields. II. From the equation of state of a gas. *Proc. R. Soc. Lond., A* **1924**, *106*, 463 – 477.

[118] M. Elstner, P. Hobza, T. Frauenheim, S. Suhai, E. Kaxiras. Hydrogen bonding and stacking interactions of nucleic acid base pairs: A density-functional-theory based treatment. *J. Chem. Phys.* **2001**, *114*, 5149 – 5155.

[119] J. Hepburn, G. Scoles, R. Penco. A simple but reliable method for the prediction of intermolecular potentials. *Chem. Phys. Lett.* **1975**, *36*, 451 – 456.

[120] R. Ahlrichs, R. Penco, G. Scoles. Intermolecular forces in simple systems. *Chem. Phys.* **1977**, *19*, 119 – 130.

[121] C. Gonzalez, E. C. Lim. Evaluation of the Hartree-Fock dispersion (HFD) model as a practical tool for probing intermolecular potentials of small aromatic clusters: Comparison of the HFD and MP2 intermolecular potentials. *J. Phys. Chem. A* **2003**, *107*, 10105 – 10110.

[122] S. Grimme. Accurate description of van der Waals complexes by density functional theory including empirical corrections. *J. Comput. Chem.* **2004**, *25*, 1463 – 1473.

[123] S. Grimme. Semiempirical GGA-type density functional constructed with a long-range dispersion correction. *J. Comput. Chem.* **2006**, *27*, 1787 – 1799.

[124] S. Grimme, J. Antony, T. Schwabe, C. Mück-Lichtenfeld. Density functional theory with dispersion corrections for supramolecular structures, aggregates, and complexes of (bio)organic molecules. *Org. Biomol. Chem.* **2007**, *5*, 741 – 758.

[125] S. Grimme, J. Antony, S. Ehrlich, H. Krieg. A consistent and accurate *ab initio* parametrization of density functional dispersion correction (DFT-D) for the 94 elements H–Pu. *J. Chem. Phys.* **2010**, *132*, 154104.

[126] Q. Wu, W. Yang. Empirical correction to density functional theory for van der Waals interactions. *J. Chem. Phys.* **2002**, *116*, 515 – 524.

[127] P. Jurečka, J. Černý, P. Hobza, D. R. Salahub. Density functional theory augmented with an empirical dispersion term. Interaction energies and geometries of 80 noncovalent complexes compared with *ab initio* quantum mechanics calculations. *J. Comput. Chem.* **2007**, *28*, 555 – 569.

[128] J. P. McNamara, I. H. Hillier. Semi-empirical molecular orbital methods including dispersion corrections for the accurate prediction of the full range of intermolecular interactions in biomolecules. *Phys. Chem. Chem. Phys.* **2007**, *9*, 2362 – 2370.

[129] T. Tuttle, W. Thiel. OMx-D: semiempirical methods with orthogonalization and dispersion corrections. Implementation and biochemical application. *Phys. Chem. Chem. Phys.* **2008**, *10*, 2159 – 2166.

[130] J. Řezáč, J. Fanfrlík, D. Salahub, P. Hobza. Semiempirical quantum chemical PM6 method augmented by dispersion and H-bonding correction terms reliably describes various types of noncovalent complexes. *J. Chem. Theory Comput.* **2009**, *5*, 1749 – 1760.

[131] M. Korth, M. Pitoňák, J. Řezáč, P. Hobza. A transferable H-bonding correction for semiempirical quantum-chemical methods. *J. Chem. Theory Comput.* **2010**, *6*, 344 – 352.

[132] M. Korth. Third-generation hydrogen-bonding corrections for semiempirical QM methods and force fields. *J. Chem. Theory Comput.* **2010**, *6*, 3808 – 3816.

[133] J. Řezáč, P. Hobza. Advanced corrections of hydrogen bonding and dispersion for semiempirical quantum mechanical methods. *J. Chem. Theory Comput.* **2012**, *8*, 141 – 151.

[134] J. Backus. Can programming be liberated from the von Neumann style?: A functional style and its algebra of programs. *Commun. ACM* **1978**, *21*, 613 – 641.

[135] NVIDIA Corporation, Santa Clara, CA, *CUDA C Programming Guide*, **2012**.

[136] J. J. P. Stewart, P. Császár, P. Pulay. Fast semiempirical calculations. *J. Comput. Chem.* **1982**, *3*, 227 – 228.

[137] A. R. Gargaro, G. B. Bloomberg, C. E. Dempsey, M. Murray, M. J. A. Tanner. The solution structures of the first and second transmembrane-spanning segments of band 3. *Eur. J. Biochem.* **1994**, *221*, 445 – 454, PDB ID: 1BTQ.

[138] J. W. O'Neill, D. E. Kim, K. Johnsen, D. Baker, K. Y. Zhang. Single-site mutations induce 3D domain swapping in the B1 domain of protein L from Peptostreptococcus magnus. *Structure* **2001**, *9*, 1017 – 1027, PDB ID: 1K50.

[139] M. Rubini, S. Lepthien, R. Golbik, N. Budisa. Aminotryptophan-containing barstar: structure-function tradeoff in protein design and engineering with an expanded genetic code. *Biochim. Biophys. Acta* **2006**, *1764*, 1147 – 1158, PDB ID: 2HXX.

[140] C. Ciatto, F. Bahna, N. Zampieri, H. C. VanSteenhouse, P. S. Katsamba, G. Ahlsen, O. J. Harrison, J. Brasch, X. Jin, S. Posy, J. Vendome, B. Ranscht, T. M. Jessell, B. Honig, L. Shapiro. T-cadherin structures reveal a novel adhesive binding mechanism. *Nat. Struct. Mol. Biol.* **2010**, *17*, 339 – 347, PDB ID: 3K6F.

[141] A. A. Fedorov, K. A. Magnus, M. H. Graupe, E. E. Lattman, T. D. Pollard, S. C. Almo. X-ray structures of isoforms of the actin-binding protein profilin that differ in their affinity for phosphatidylinositol phosphates. *Proc. Natl. Acad. Sci. USA* **1994**, *91*, 8636 – 8640, PDB ID: 1ACF.

[142] J. Choi, S. Choi, J. K. Chon, J. Choi, M.-K. Cha, I.-H. Kim, W. Shin. Crystal structure of the C107S/C112S mutant of yeast nuclear 2-Cys peroxiredoxin. *Proteins* **2005**, *61*, 1146 – 1149, PDB ID: 2A4V.

[143] G. Vaaje-Kolstad, L. A. Bøhle, S. Gåseidnes, B. Dalhus, M. Bjørås, G. Mathiesen, V. G. Eijsink. Characterization of the chitinolytic machinery of Enterococcus faecalis V583 and high-resolution structure of its oxidative CBM33 enzyme. *J. Mol. Biol.* **2012**, *416*, 239 – 254, PDB ID: 4A02.

[144] T. Tsukazaki, H. Mori, Y. Echizen, R. Ishitani, S. Fukai, T. Tanaka, A. Perederina, D. G. Vassylyev, T. Kohno, A. D. Maturana, K. Ito, O. Nureki. Structure and function of a membrane component SecDF that enhances protein export. *Nature* **2011**, *474*, 235 – 238, PDB ID: 3AQO.

[145] D. Doron, D. T. Major, A. Kohen, W. Thiel, X. Wu. Hybrid quantum and classical simulations of the dihydrofolate reductase catalyzed hydride transfer reaction on an accurate semi-empirical potential energy surface. *J. Chem. Theory Comput.* **2011**, *7*, 3420 – 3437.

[146] X. Wu, W. Thiel, S. Pezeshki, H. Lin. Specific reaction path hamiltonian for proton transfer in water: Reparameterized semiempirical models. *J. Chem. Theory Comput.* **2013**, *9*, 2672 – 2686.

[147] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, Cambridge University Press, New York, **1996**.

[148] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, New York, **1987**.

[149] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.* **1944**, *2*, 164 – 168.

[150] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.* **1963**, *11*, 431 – 441.

[151] J. A. Nelder, R. Mead. A simplex method for function minimization. *Comput. J.* **1965**, *7*, 308 – 313.

[152] D. Lee, M. Wiswall. A parallel implementation of the simplex function minimization routine. *Comput. Econ.* **2007**, *30*, 171 – 187.

[153] W. L. Price. A controlled random search procedure for global optimisation. *Comput. J.* **1977**, *20*, 367 – 370.

[154] W. L. Price, *A Controlled Random Search Procedure for Global Optimization* in *Towards Global Optimisation 2*, L. C. W. Dixon, G. P. Szegö (Eds.), Amsterdam: North-Holland Pub. Co., **1978**, pp. 71 – 84.

[155] W. L. Price. Global optimization by controlled random search. *J. Optim. Theory Appl.* **1983**, *40*, 333–348.

[156] W. Price. Global optimization algorithms for a CAD workstation. *J. Optim. Theory Appl.* **1987**, *55*, 133–146.

[157] M. M. Ali, C. Storey. Modified controlled random search algorithms. *Int. J. Comput. Math.* **1994**, *53*, 229 – 235.

[158] M. M. Ali, A. Törn, S. Viitanen. A numerical comparison of some modified controlled random search algorithms. *J. Global Optim.* **1997**, *11*, 377 – 385.

[159] P. Kaelo, M. M. Ali. Some variants of the controlled random search algorithm for global optimization. *J. Optim. Theory Appl.* **2006**, *130*, 253 – 264.

[160] I. Garcia, P. M. Ortigosa, C. L. G., G. T. Herman, S. Matej, *Multidimensional Optimization in Image Reconstruction From Projections* in *Developments in Global Optimization*, L. M. Bomze, T. Csendes, R. Horst, P. M. Pardalos (Eds.), Dordrecht: Kluwer Academic Publishers, **1997**, pp. 289 – 300.

[161] P. Hobza, R. Zahradník. Van der Waals molecules: Quantum chemistry, physical properties, and reactivity. *Int. J. Quant. Chem.* **1983**, *23*, 325 – 338.

[162] E. Clementi, G. Corongiu. Van der Waals interaction energies of helium, neon, and argon with naphthalene. *J. Phys. Chem. A* **2001**, *105*, 10379 – 10383.

[163] R. Sedlak, T. Janowski, M. Pitoňák, J. Řezáč, P. Pulay, P. Hobza. Accuracy of quantum chemical methods for large noncovalent complexes. *J. Chem. Theory Comput.* **2013**, *9*, 3364 – 3374.

[164] P. Jurečka, J. Šponer, J. Černý, P. Hobza. Benchmark database of accurate (MP2 and CCSD(T) complete basis set limit) interaction energies of small model complexes, DNA base pairs, and amino acid pairs. *Phys. Chem. Chem. Phys.* **2006**, *8*, 1985 – 1993.

[165] T. A. Halgren. The representation of van der Waals (vdW) interactions in molecular mechanics force fields: Potential form, combination rules, and vdW parameters. *J. Am. Chem. Soc.* **1992**, *114*, 7827 – 7843.

[166] A. Bondi. Van der Waals volumes and radii. *J. Phys. Chem.* **1964**, *68*, 441 – 451.

[167] G. A. Jeffrey, *An Introduction to Hydrogen Bonding*, Oxford University Press, **1997**.

[168] S. S. Batsanov. Van der Waals radii of elements. *Inorg. Mater.* **2001**, *37*, 871 – 885.

[169] X. Wu, A. Koslowski, W. Thiel, *Semiempirical Quantum Chemistry* in *Electronic Structure Calculations on Graphics Processing Units*, R. C. Walker, A. W. Goetz (Eds.), John Wiley & Sons, Inc., **2014**, accepted for publication.

[170] H. J. Kulik, N. Luehr, I. S. Ufimtsev, T. J. Martínez. *Ab initio* quantum chemistry for protein structures. *J. Phys. Chem. B* **2012**, *116*, 12501 – 12509.

[171] R. A. Laskowski, M. W. MacArthur, D. S. Moss, J. M. Thornton. PROCHECK: A program to check the stereochemical quality of protein structures. *J. Appl. Crystallogr.* **1993**, *26*, 283 – 291.

[172] R. A. Engh, R. Huber. Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Cryst.* **1991**, *A47*, 392 – 400.

[173] J. Janin, S. Wodak, M. Levitt, B. Maigret. Conformation of amino acid side-chains in proteins. *J. Mol. Biol.* **1978**, *125*, 357 – 386.

[174] A. L. Morris, M. W. MacArthur, E. G. Hutchinson, J. M. Thornton. Stereochemical quality of protein structure coordinates. *Proteins.* **1992**, *12*, 345 – 364.

[175] T. M. Henderson, A. F. Izmaylov, G. Scalmani, G. E. Scuseria. Can short-range hybrids describe long-range-dependent properties? *J. Chem. Phys.* **2009**, *131*, 044108.

[176] M. Kitano, T. Fukuyama, K. Kuchitsu. Molecular structure of *N*-methylacetamide as studied by gas electron diffraction. *Bull. Chem. Soc. Jpn.* **1973**, *46*, 384 – 387.

[177] F. H. Allen, S. Bellard, M. D. Brice, B. A. Cartwright, A. Doubleday, H. Higgs, T. Hummelink, B. G. Hummelink-Peters, O. Kennard, W. D. S. Motherwell, J. R. Rodgers, D. G. Watson. The Cambridge Crystallographic Data Centre: Computer-based search, retrieval, analysis and display of information. *Acta Cryst.* **1979**, *B35*, 2331 – 2339.

[178] G. N. Ramachandran, C. Ramakrishnan, V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.* **1963**, *7*, 95 – 99.

[179] L. Pauling, R. B. Corey, H. R. Branson. The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. USA* **1951**, *37*, 205 – 211.

[180] G. de M. Seabra, R. C. Walker, A. E. Roitberg. Are current semiempirical methods better than force fields? A study from the thermodynamics perspective. *J. Phys. Chem. A* **2009**, *113*, 11938 – 11948.

[181] K. Möhle, H.-J. Hofmann, W. Thiel. Description of peptide and protein secondary structures employing semiempirical methods. *J. Comput. Chem.* **2001**, *22*, 509 – 520.

Appendix A

Semiempirical Quantum Chemical Calculations Accelerated on a Hybrid Multicore CPU-GPU Computing Platform

Xin Wu, Axel Koslowski, and Walter Thiel

Journal of Chemical Theory and Computation, **2012**, 8, 2272 – 2281.

I developed the GPU module (with code revision by Axel Koslowski) of the MNDO package, implemented part of the OpenMP parallelization, carried out all the calculations, analyzed the results, and drafted the manuscript and supporting information.

JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

Semiempirical Quantum Chemical Calculations Accelerated on a Hybrid Multicore CPU−GPU Computing Platform

Xin Wu, Axel Koslowski, and Walter Thiel*

Max-Planck-Institut für Kohlenforschung, Kaiser-Wilheim-Platz 1, 45470 Mülheim an der Ruhr, Germany

Supporting Information

ABSTRACT: In this work, we demonstrate that semiempirical quantum chemical calculations can be accelerated significantly by leveraging the graphics processing unit (GPU) as a coprocessor on a hybrid multicore CPU−GPU computing platform. Semiempirical calculations using the MNDO, AM1, PM3, OM1, OM2, and OM3 model Hamiltonians were systematically profiled for three types of test systems (fullerenes, water clusters, and solvated crambin) to identify the most time-consuming sections of the code. The corresponding routines were ported to the GPU and optimized employing both existing library functions and a GPU kernel that carries out a sequence of noniterative Jacobi transformations during pseudodiagonalization. The overall computation times for single-point energy calculations and geometry optimizations of large molecules were reduced by one order of magnitude for all methods, as compared to runs on a single CPU core.

1. INTRODUCTION

Semiempirical molecular orbital (MO) methods are widely used in quantum chemical studies of large molecules.[1−8] They provide qualitative insights into chemical problems at low cost, because they are much faster than the more accurate ab initio and density functional methods, typically by at least 3 orders of magnitude. This efficiency makes them especially useful for initial explorations before employing more expensive calculations, for correlating large sets of experimental and theoretical data to establish trends, and for studying the properties and dynamical behavior of complex systems.[1,2]

Semiempirical and ab initio MO approaches share the same conceptual framework, but a number of drastic approximations are introduced at the semiempirical level to speed up the calculations. The most important approximation is that many of the smaller one- and two-electron integrals are neglected, and the remaining ones are either determined directly from experiment or calculated from suitable parametric functions.[4,9] As a consequence, integral evaluation in semiempirical calculations formally scales as $O(N^2)$ for N basis functions, in contrast to $O(N^4)$ in generic ab initio MO calculations. The dominant computational effort is therefore shifted from integral evaluation to linear algebra operations, for example, matrix multiplication and matrix diagonalization, which formally scale as $O(N^3)$.

To further extend the scope of the current quantum chemical work, it is essential to take advantage of new powerful computer architectures. The traditional hardware is based on the general-purpose central processing unit (CPU), which is capable of executing multiple threads simultaneously due to various multicore designs. On the other hand, the graphics processing unit (GPU) is a many-core architecture originally designed for the rapid processing of images. To the present day, GPUs have evolved to being able to perform on the order of 10^{12} floating-point operations per second (FLOP/s) using thousands of threads and high memory bandwidth.[10] These advances have made the GPU become particularly suitable for highly parallel arithmetic-intensive computations, in particular also in the field of molecular modeling (for reviews, see refs 11−13). Recently, many authors have implemented GPU-oriented algorithms in their quantum chemistry codes and reported considerable speedups, up to about 100 times as compared to CPU-only implementations.[14−20] At the semiempirical level, we are only aware of some unpublished work in this area.[21]

In this article, we describe a comprehensive optimization of the self-consistent-field (SCF) code in our semiempirical quantum chemistry program MNDO[22] on a hybrid multicore CPU−GPU computing platform. Six semiempirical methods, MNDO (Modified Neglect of Differential Overlap),[23] AM1 (Austin Model 1),[24] PM3 (Parameterized Model 3),[25] and OM*x* (Orthogonalization-corrected Model *x*, *x* = 1, 2, and 3),[26] are considered. We first report profiles of our original code on a single CPU core using three representative types of test systems, fullerenes,[27,28] water clusters,[29] and crambin[15] solvated in water spheres of increasing size. The most time-consuming routines identified in this manner are ported to the GPU by utilizing both vendor-optimized linear algebra library functions[30,31] as well as a manually tuned GPU kernel. Finally, the performance of the resulting code is checked through single-point energy evaluations and geometry optimizations for all test cases to establish the speedups that can be achieved for semiempirical calculations on a hybrid multicore CPU−GPU platform.

2. COMPUTATIONAL DETAILS

This work covers the semiempirical MNDO,[23] AM1,[24] PM3,[25] and OM*x* (*x* = 1, 2, and 3)[26] methods, which are all based on the NDDO (Neglect of Diatomic Differential Overlap) integral approximation.[32] The standard MNDO-type methods (MNDO, AM1, and PM3) have served for decades as widely

Received: March 2, 2012
Published: May 25, 2012

dx.doi.org/10.1021/ct3001798 | J. Chem. Theory Comput. 2012, 8, 2272−2281

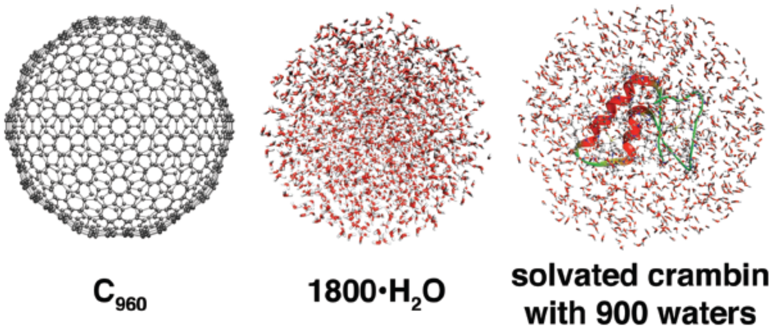


Figure 1. The largest system in each test set.

used computational tools. The more recently developed OM*x* methods[26] add orthogonalization corrections to the MNDO model and offer significant improvements in systematic benchmarks for ground-state molecules[33,34] and especially for excited-state properties.[35] The major computational burdens during SCF calculations are expected to be similar for all of these methods. We employ three sets of test molecules, fullerenes (C*i*), water clusters (W*j*), and the protein crambin solvated in water balls (P*k*).[36] These systems represent typical application areas of semiempirical methods, ranging from regularly shaped highly symmetric molecules to molecular assemblies with many degrees of freedom and complex biochemical systems.

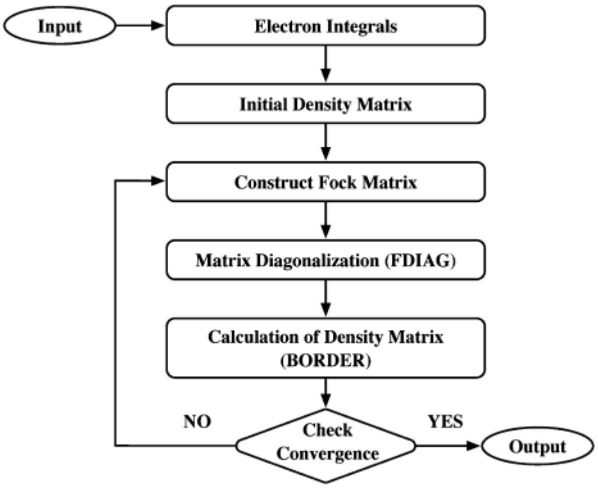
All present calculations have been carried out on a workstation featuring one Intel Xeon X5670 processor (6 cores @ 2.93 GHz),[37] 12 GB of main memory, and one NVidia Tesla C2070 GPU (448 cores @ 1.15 GHz) with 6 GB of global memory. Figure 1 shows the largest test systems in each set: C_{960}, 1800·H_2O, and crambin·900H_2O. The latter two are close to the size that can be handled in the current setup with the memory available in the workstation.[38] All floating point arithmetic operations were performed in double precision to avoid numerical inaccuracies in the computation of the large molecules considered (>1000 heavy atoms). The SCF convergence criteria were 1.0×10^{-6} eV for the electronic energy and 1.0×10^{-6} for the maximum change in the density matrix. When using the DIIS (direct inversion of iterative subspace)[39] procedure in the SCF iterations, the maximum error matrix element was required to be less than 1.0×10^{-6} eV. The norm of the gradient vector ($\|g\| < 1.0$ kcal/(mol·Å)) served as the convergence criterion in geometry optimizations.

The FORTRAN compiler from the development suite "Intel Composer XE 12.0" and the NVidia CUDA (Compute Unified Device Architecture) Toolkit 3.2 were used during the development of the CPU and GPU code in a CVS version of MNDO99.[22] In the CPU-only implementation, the Intel Math Kernel Library (MKL) was dynamically linked to provide the required BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra Package) functions. Alternatively, on the hybrid multicore CPU–GPU platform, the corresponding GPU-accelerated routines were taken from the libraries CUBLAS[30] and MAGMA.[31] On our workstation, the CPU–GPU communication is limited by a 16-lane PCIe (Peripheral Component Interconnect Express) connection with a theoretical bandwidth of 8.0 GB/s. The actually available bandwidth was observed to be greater than 4.0 GB/s in our tests. Having sufficient bandwidth is essential because all data to be processed by the GPU and the results obtained must be transferred over this connection. In our implementation, the communication overhead was carefully minimized such that the data transfer only needs a very small portion of the total computation time, for example, ~0.5% for the MNDO calculation of 1800·H_2O on the hybrid CPU–GPU platform.

3. SEMIEMPIRICAL CALCULATIONS ON THE CPU-ONLY PLATFORM

A simplified workflow of a semiempirical single-point SCF calculation is sketched in Chart 1. The tasks in this workflow are generic and will appear in virtually any semiempirical SCF program. The profiling results reported below for these tasks are thus of general relevance for semiempirical code development.

Chart 1. Workflow of a Single-Point Semiempirical SCF Calculation



Because of the NDDO approximation, the evaluation of the two-electron integrals formally requires only $O(N^2)$ operations. Therefore, the diagonalization of the Fock matrix (FDIAG) and the computation of the density matrix (BORDER) are the $O(N^3)$ steps that dominate the calculations in practice.[40] Two additional optional procedures, DIIS[39] and the pseudodiagonalization (PDIAG),[41] may also consume a significant part of the computation time and hence deserve special attention. The DIIS scheme makes use of the results from previous SCF iterations to construct a new Fock matrix such that SCF convergence is accelerated; this requires the computation of an error matrix that scales as $O(N^3)$. The full diagonalization (FDIAG) can often be replaced by a less expensive pseudodiagonalization (PDIAG), which involves a triple matrix product (MMM) and noniterative Jacobi transformations (JACOBI); normally, PDIAG can be applied throughout the SCF process except for the first and the last cycles.

We have used the existing and already sufficiently optimized CPU code to profile the semiempirical calculations for all test molecules on a single CPU. Two typical computational protocols were chosen: (A) pseudodiagonalization without DIIS, and (B) full diagonalization with DIIS. In the case of the MNDO-type methods, the fullerenes and the water clusters were run using (A), and the crambin set was computed using (B). All MNDO-type methods behave similarly in this analysis, so we show only the profiles for AM1 in Figure 2a. For the OM*x* methods, we only examined the fullerenes (B) and the water clusters W1000 and W1200 (A); crambin could not be handled because of the missing sulfur parameters, while for the larger water clusters the memory capacity was not sufficient to process the orthogonalization corrections with the existing setup. We present only the OM2 profiles in Figure 2b because all three OM*x* methods show similar behavior (except for the integral routines because there are no orthogonalization corrections to the two-center terms in OM1). The profiling results for the other methods (MNDO, PM3, OM1, and OM3) are documented in the Supporting Information.

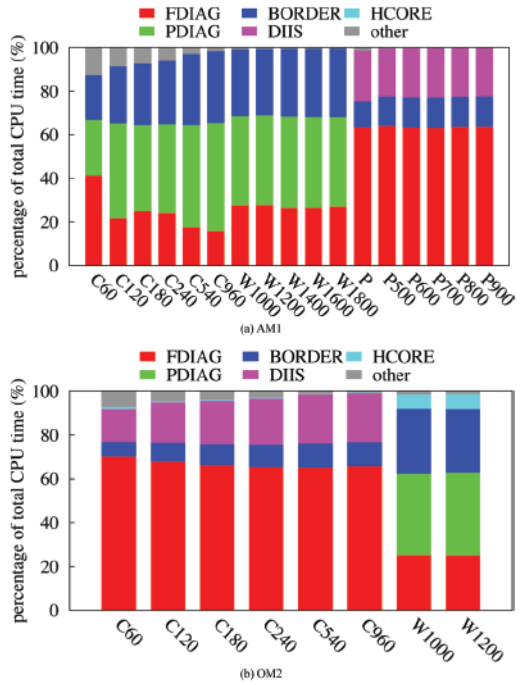


Figure 2. Profiles of the AM1 and OM2 calculations of the test molecules on a single CPU core.

Table 1 reports the computation times as well as the number of SCF iterations and full diagonalizations during the SCF procedure for all six semiempirical methods, taking the water cluster W1000 (1000·H_2O) as a typical example.[42] It is obvious that the number of SCF iterations (~20) and the computation times per SCF cycle are roughly in the same range for all methods; none of them is much more expensive or much cheaper than the others. AM1 and PM3 are slightly more costly than the original MNDO method, presumably at least partly due to the more complex core-repulsion function.[24,25] Despite the orthogonalization corrections to the one-center terms, OM1 turns out to be the fastest of all six methods. OM2 and OM3 are somewhat slower than OM1 because of the additional three-center terms arising from the orthogonalization corrections to the two-center resonance integrals.[26]

The MNDO-type calculations (Figure 2a) are completely dominated by the four routines FDIAG, PDIAG, BORDER, and DIIS (see above). The percentage of the CPU time consumed by other tasks (i.e., mainly integral evaluation and Fock matrix construction) decreases dramatically with increasing system size, for example, from 12.1% to 1.6% in the MNDO calculations on C_{60} and C_{960}, respectively. In small molecules, the other tasks may require ~10% of the time, but the total computation time is negligible in these cases; for example, the MNDO calculations on C_{60} and C_{120} just take 0.1 and 1.0 s, respectively. In larger molecules with several hundred atoms, the four dominant routines together usually account for >98% of the total time of an MNDO-type calculation. The situation is slightly different for the OM*x* methods because of the additional orthogonalization corrections to the one-electron core Hamiltonian (HCORE). The share of HCORE becomes more pronounced especially in OM2 and OM3 due to the inclusion of three-center terms in the orthogonalization corrections. For the water cluster W1000, for instance, the fraction of the computer time for HCORE amounts to 6.5% in OM2 and 6.2% in OM3. Nevertheless, FDIAG, PDIAG, BORDER, and DIIS remain the four most CPU-intensive routines also in these cases and together consume more than 90% of the total computation time.

By default, PDIAG rather than FDIAG is commonly applied whenever possible. According to the profiling, PDIAG will then normally contribute more than one-third to the total CPU time, which calls for a more detailed analysis of this routine. The breakdown of the computational effort in PDIAG is shown in Figure 3 for MNDO calculations on fullerenes and water clusters that use the default threshold for deciding whether the Jacobi transformation is done for a given pair of eigenvectors.[41] The triple matrix product MMM takes 50−60% of the CPU time within PDIAG in the case of the water clusters, and

Table 1. Computation Times (in seconds) of MNDO, AM1, PM3, OM1, OM2, and OM3 Single-Point Energy Calculations of 1000·H_2O on a Single CPU Core[a]

	MNDO-based methods			OM*x* methods		
	MNDO	AM1	PM3	OM1	OM2	OM3
N_{SCF}	18	19	23	18	20	22
N_{diag}	4	4	4	4	4	4
time	1433.17	1595.66	2045.83	1316.51	1740.69	1818.91
time/N_{SCF}	79.62	83.98	88.95	73.14	87.03	82.68

[a] The number of SCF iterations and full diagonalizations is denoted by N_{SCF} and N_{diag}, respectively.

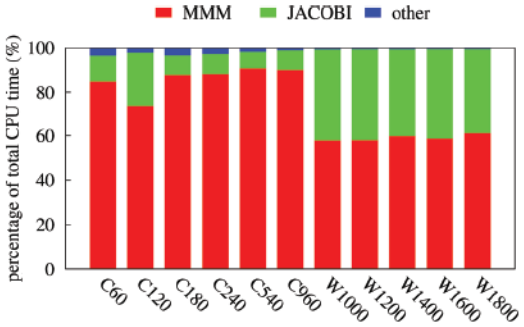


Figure 3. Profiles of PDIAG in MNDO calculations of fullerenes and water clusters on a single CPU core.

typically more than 80% in the case of the fullerenes, while most of the rest is consumed by the noniterative Jacobi transformations. The difference between the fullerenes and the water clusters arises from the thresholding: in the latter case, more Jacobi rotations are found to be needed, which leads to a significant increase in the computational effort (see Figure 3).

To summarize, the profiling has identified four bottlenecks that together consume more than 90% of the CPU time in semiempirical calculations on large molecules. The GPU adaptation of the code will focus exclusively on the corresponding four routines FDIAG, PDIAG, BORDER, and DIIS.

4. PROFILE-GUIDED OPTIMIZATION ON THE HYBRID PLATFORM

4.1. Survey of Computational Tasks. To establish notation, we briefly summarize the main computational tasks. FDIAG solves the eigenvalue equation:

$$\mathbf{FC} = \mathbf{CE} \tag{1}$$

where **F** denotes the Fock matrix, **C** is the matrix of MO coefficients, and **E** is a diagonal matrix containing the MO energies. The triple matrix multiplication in PDIAG is

$$\mathbf{F}_{\mathrm{MO}} = \mathbf{C}_{\mathrm{o}}^{\mathrm{T}}\mathbf{F}\mathbf{C}_{\mathrm{v}} \tag{2}$$

where $\mathbf{F}_{\mathrm{MO}}$ is the occupied-virtual block of the Fock matrix. $\mathbf{C}_{\mathrm{o}}$ and $\mathbf{C}_{\mathrm{v}}$ are the occupied and virtual vectors, respectively. BORDER performs one dominant matrix multiplication (>99% CPU time, eq 3) to determine the density matrix **P**.

$$\mathbf{P} = 2\mathbf{C}_{\mathrm{o}}\mathbf{C}_{\mathrm{o}}^{\mathrm{T}} \tag{3}$$

DIIS evaluates the error matrix, which requires matrix multiplications (eq 4) that are computationally much more demanding (~98% CPU time) than the subsequent matrix–vector multiplications (eqs 5 and 6).

$$\mathbf{e}_k = \mathbf{FP} - \mathbf{PF} \tag{4}$$

$$\mathbf{b}_k = \tilde{\mathbf{E}}^{\mathrm{T}}\mathbf{e}_k \tag{5}$$

$$\mathbf{F}' = \tilde{\mathbf{F}}\mathbf{x} \tag{6}$$

$\mathbf{e}_k$ is the error matrix of the kth DIIS step, which is stored in linearly packed form as the kth column of the composite matrix $\tilde{\mathbf{E}}$. $\mathbf{b}_k$ collects the scalar products of error matrices evaluated in the kth DIIS step. The new Fock matrix $\mathbf{F}'$ is obtained as a linear combination of previous Fock matrices stored in composite form ($\tilde{\mathbf{F}}$), using the coefficient vector **x** determined in the DIIS procedure.[39]

4.2. GPU-Accelerated Library Functions. As documented above, matrix diagonalization, matrix–matrix multiplication, and matrix–vector multiplication are the most important linear algebra operations in semiempirical SCF calculations. To identify the routines in the MKL, CUBLAS, and MAGMA libraries best suited for our purposes, we have performed test calculations. In the following, we briefly summarize the results of these tests (for detailed data, see the Supporting Information).

Matrix Diagonalization. We have applied four different LAPACK routines (SYEV, SYEVX, SYEVD, and SYEVR)[43] for Fock matrix diagonalization (eq 1). The MKL library contains an implementation of all four routines, and multithreaded versions are available for use with symmetric multiprocessing systems. MAGMA offers SYEVD only, taking advantage of the hybrid CPU–GPU architecture: one of the steps in the algorithm is executed on the CPU and the other two on the GPU.[44]

The tests on one single CPU core indicate that SYEV is always slowest while SYEVR is a little faster than SYEVD. All CPU-only calculations are accelerated when run in parallel on 6 CPU cores, but the speedups are moderate (at most by a factor of 3 for large matrices) because these routines are bandwidth-bound; that is, the performance of the algorithm is limited by the memory bandwidth the hardware can provide. On the hybrid CPU–GPU platform, there are larger gains in performance when all six CPU cores are used in combination with the GPU: the hybrid SYEVD (H6CG) routine is always the fastest one for any matrix size, with a speedup over SYEVD (1C) of about 5-fold. Because SYEVR is not available in MAGMA, SYEVD was used for Fock matrix diagonalization in the following semiempirical calculations to allow for a fair comparison between CPU-only and hybrid CPU–GPU computing platforms.

Matrix–Matrix Multiplication. Depending on the type of the matrices involved, multiplications can be carried out with the BLAS routines GEMM, SYMM, or SYRK. GEMM performs the generic matrix–matrix multiplication, whereas SYMM is more specific because at least one of the matrices is required to be symmetric. SYRK can be applied if the product matrix is known to be symmetric so that only the upper or lower triangular entries need to be computed, which can lead to a speedup of almost 2 by skipping almost one-half of the floating-point operations as compared to the general case. SYRK will thus be an attractive choice for computing the density matrix, which is symmetric by definition (eq 3).

To assess the performance of GEMM, SYMM, and SYRK for our purposes, we evaluated the product $\mathbf{P} = \mathbf{AA}^{\mathrm{T}}$, where **A** is a random real symmetric matrix. On the CPU-only platform, the parallel runs with 6 cores show a speedup of almost 6 over those on one core (6C vs 1C). On the hybrid CPU–GPU platform, GEMM and SYRK are much faster still, with a peak performance close to 300 GFLOP/s. Hence, the SYRK routine was selected for computing the density matrix in BORDER (eq 3), while GEMM was chosen for carrying out the other matrix–matrix multiplications in PDIAG (eq 2) and in DIIS (eq 4). The use of these GPU-accelerated routines on the hybrid platform offers a ~25× boost in performance as compared to a single CPU core.

Matrix–Vector Multiplication. Matrix–vector multiplications are required in DIIS to compute $\mathbf{b}_k$ (eq 5, operation $\mathbf{p} = \mathbf{M}^{\mathrm{T}}\mathbf{v}$) and $\mathbf{F}'$ (eq 6, operation $\mathbf{p} = \mathbf{Mv}$). These are rather special cases because the matrices in these products have a very large number of rows (of the order of 10^7 for the biggest molecules in this work) and only a very small number of columns (of the order of 10). Three different library routines are available: GEMV is the generic routine for matrix–vector multiplications and can be applied in both situations; DOTL and AXPYL are alternatives. DOTL loops over the dot product (DOT) of each row in **M** and **v** to produce $\mathbf{p} = \mathbf{M}^T\mathbf{v}$, whereas AXPYL sequentially sweeps each column in **M** with **v** and assembles the results for **Mv** in **p**. In DIIS, the use of DOTL and AXPYL ensures coalesced memory access in the computation of $\mathbf{b}_k$ and **F**′, respectively, because the two-dimensional arrays $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{F}}$ are stored in column major order. The row dimension of these arrays is given by the number of iterations in the DIIS procedure and will thus be much smaller than their column dimension that is equal to the number of elements in the linearly packed Fock matrix.

Because of the inherently low ratio of compute-to-memory accesses, the routines GEMV, DOTL, and AXPYL are all bandwidth-bound, and using more CPU cores thus hardly helps to increase the speed. The DOTL function from the GPU-based CUBLAS library is best suited for $\mathbf{p} = \mathbf{M}^T\mathbf{v}$, being up to about 5 times faster than the CPU-based counterpart. GEMV is always superior to AXPYL for **p** = **Mv** on the same platform: in this case, the GPU-based GEMV routine can be roughly 10 times faster than the CPU-based version. In view of these comparisons, DOTL and GEMV were chosen for the computation of $\mathbf{b}_k$ and **F**′ on the hybrid computing platform, respectively.

4.3. Jacobi Transformation on GPU.

There is no ready-to-use library function available for the noniterative Jacobi transformation in PDIAG. The underlying algorithm[41] involves several approximations, including the neglect of changes in the matrix elements of the secular determinant upon rotation. This substantially simplifies parallelization of the algorithm. For further background information, the interested reader is referred to other recent work related to implementing Jacobi rotations on GPUs[45] and to earlier work concerning parallel Jacobi rotations on CPUs.[46−48] In this section, we first describe the basic algorithms considered in our implementation and then discuss the results.

The original CPU algorithm[41] in PDIAG performs sweeps of independent 2 × 2 rotations of one occupied and one virtual molecular orbital trial vector from the last SCF iteration, C_i and C_a:

$$C_i' = cC_i - sC_a; \; C_a' = sC_i + cC_a \tag{7}$$

where c and s denote the rotation coefficients, and the new MO vectors are indicated by primes. The rotation coefficients are calculated from the approximate MO energies ε_i and ε_a of the previous SCF iteration and from the matrix element $\mathcal{F}_{ia}$ of the Fock matrix in the MO basis connecting both orbitals.

$$c = 1 - \frac{u}{2}; \; s = \pm\sqrt{u - \frac{u^2}{4}}; \; u = \left(\frac{\mathcal{F}_{ia}}{\varepsilon_a - \varepsilon_i}\right)^2 \tag{8}$$

The sign of s is opposite of that of the Fock matrix element.

During one sweep, all rotations on a set of independent occupied-virtual orbital pairs are performed that satisfy a predefined threshold criterion.[41] If, respectively, n_o and n_v are the number of occupied and virtual molecular orbitals, and n_{small} and n_{large} are the smaller and the larger number of n_o and n_v, there are n_{large} sweeps and n_{small} orbital pairs examined per sweep.

The rotation of one pair of vectors in eq 7 can be carried out by the BLAS level 1 routine ROT. In semiempirical calculations, the number of basis functions N is usually less than 10^4. In this range, the ROT (G) routine from the GPU-based CUBLAS library is not yet efficient, due to the low ratio (<3) of compute-to-memory accesses, and is outperformed by the single-core CPU variant ROT (1C); this situation is reversed only for huge values of N (>10^6) that are not reached in practice. Semiempirical calculations will thus not benefit from the naive use of ROT (G) for the Jacobi transformation in PDIAG. This calls for the development of a dedicated version of PDIAG that exploits the potential of the GPU for speeding up this transformation, which may represent a significant part of the overall computational effort (see above).

Four algorithms were devised for the Jacobi transformation on the GPU whose architecture is designed for executing large numbers of parallel threads. The algorithms differ in the way how the work is distributed among the threads, how these are organized in blocks, how the threads are synchronized, and, most importantly, where the decision is made as to which orbital pairs to rotate. A detailed description of all algorithms is given in the Supporting Information.

Algorithms I−III have in common that the decision whether to rotate an orbital pair is made in the threads just before the rotation would be performed. These algorithms come in three variants depending on the treatment of the rotation coefficients in eq 7 that may be recomputed as needed (first variant) or precomputed by a separate routine and stored in global memory. For efficient access to these quantities in global memory, the kernels may rely on the caching mechanisms of the GPU (second variant) or may explicitly buffer them using shared memory, which is much faster than global memory (third variant). These combinations give rise to nine GPU kernels altogether.

For these algorithms, there is the possibility that the threads that skip a rotation just remain idle waiting for other threads to complete a rotation. This will reduce the workload on the GPU, which will ultimately lead to lower performance. Therefore, another algorithm IV was designed in which the decision on which orbital pairs to rotate is made in a separate initial step: the rotation coefficients are precomputed, and only those are stored that correspond to rotations actually to be performed later. This complicates the logistics of precomputing the rotation coefficients in the parallel environment of the GPU. The implementation of the actual rotations in algorithm IV was based on our experience gained with the other algorithms.

For all algorithms and their variants, we tested a large number of different execution configurations, that is, the dimensions of the thread blocks, a parameter specific to GPU programming. The detailed results for the fastest algorithm (algorithm IV) are documented in the Supporting Information. In the following, we only summarize the essential findings.

We find in our extensive search that algorithm III with cached coefficients is the fastest among the first three algorithms. For C_{960}, it is about twice as fast as the second-ranked one (algorithm I with shared memory). Therefore, the rotations needed in algorithm IV were carried out in analogy to algorithm III. Systematic tests show that the time for precomputing and storing the rotation coefficients in algorithm IV is negligible and that statistically the configuration 64 × 16 performs best on average (see the Supporting Information). We have therefore adopted this configuration as our default choice. With this setup, algorithm IV consistently outperforms the other three algorithms by a large margin. For example, in the case of C_{960}, where the benefits from skipping rotations are largest, algorithm IV is more than 5 times faster than any of the three competitors.

Table 2 shows the execution times and speedups of the Jacobi transformation for selected test molecules, on a single CPU (1C) and on the GPU (algorithm IV). The acceleration of the Jacobi transformation on the GPU is expected to be influenced by the size and symmetry of the molecule. Because the architecture of the GPU is designed for massively parallel computations, the benefits for larger molecules should be higher. This is borne out by the data for 1000·H_2O and crambin: in these two unsymmetrical molecules, the percentage (R_{rot}) of required Jacobi rotations is similar (13% vs 11%), and hence the speedup is higher in the case of the larger system (6.6 vs 4.6). In the case of a highly symmetric molecule, many Fock matrix elements between occupied and virtual MOs will be zero by symmetry, which will lower R_{rot}, for example, to 2% in the icosahedral C_{960} molecule; algorithm IV is designed to exploit this reduced workload and thus leads to a maximum speedup of 7.1 relative to the time on a single CPU.

Table 2. Computation Times (in seconds) for the Jacobi Transformation in MNDO Calculations on a Single CPU (1C) and on the GPU[a]

	C_{960} (N = 3840)		1000·H_2O (N = 6000)		crambin (N = 1623)	
	time	speedup	time	speedup	time	speedup
1C	26.96		241.34		12.39	
GPU	3.82	7.1	36.52	6.6	2.71	4.6

[a] N denotes the number of basis functions.

5. SEMIEMPIRICAL CALCULATIONS ON THE CPU–GPU PLATFORM

5.1. Performance. The workstation used in this study provides several kinds of computing environments: one single CPU core (1C), multiple CPU cores (2C, 6C), one single CPU core with one GPU (H1CG), and all CPU cores plus one GPU (H6CG). Our code also allows for OpenMP parallel execution of the most demanding tasks not yet ported to the GPU, in particular, integral evaluation and Fock matrix formation. To reach the best-effort performance on the currently used hybrid CPU–GPU platform, OpenMP parallelization was turned on using all 6 CPU cores in the calculations denoted by H6CG*. In this section, we report computation times and speedups obtained in semiempirical calculations on the test molecules in all of these environments, both for MNDO-type and for OM*x* methods.

The full list of performance data is given in the Supporting Information. As a representative example, the results for 1000·H_2O are summarized in Table 3. For a given environment, the computation times are quite similar for the six semiempirical methods considered presently, especially when taking the slight differences in the number of SCF iterations into account. For instance, the computation times per SCF cycle range between 73.14 and 88.95 s in the slowest environment (1C) and between 7.43 and 9.97 s in the fastest one (H6CG*). The minimum and maximum times for the full SCF calculation differ at most by factors of 1.55 (1C) and 1.72

Table 3. Computation Times (in seconds) of the MNDO, AM1, PM3, OM1, OM2, and OM3 Single-Point Energy Evaluations for 1000·H_2O on CPU-Only (*n*C) and Hybrid CPU–GPU Platforms (H*n*CG)[a]

	1C	2C	6C	H1CG	H6CG	H6CG*
MNDO: N_{SCF} = 18, N_{diag} = 4						
time	1433.17	858.18	491.57	217.84	160.60	154.59
time/N_{SCF}	79.62	47.68	27.31	12.10	8.92	8.59
speedup		1.7	2.9	6.6	8.9	9.3
AM1: N_{SCF} = 19, N_{diag} = 4						
time	1595.66	993.67	612.29	241.02	182.97	176.74
time/N_{SCF}	83.98	52.30	32.23	12.69	9.63	9.30
speedup		1.6	2.6	6.6	8.7	9.0
PM3: N_{SCF} = 23, N_{diag} = 4						
time	2045.83	1345.98	902.66	291.88	236.51	229.36
time/N_{SCF}	88.95	58.52	39.25	12.69	10.28	9.97
speedup		1.5	2.3	7.0	8.6	8.9
OM1: N_{SCF} = 18, N_{diag} = 4						
time	1316.51	745.44	379.78	208.55	151.75	133.68
time/N_{SCF}	73.14	41.41	21.10	11.59	8.43	7.43
speedup		1.8	3.5	6.3	8.7	9.8
OM2: N_{SCF} = 20, N_{diag} = 4						
time	1740.69	1115.49	720.15	341.31	284.65	190.67
time/N_{SCF}	87.03	55.77	36.01	17.07	14.23	9.53
speedup		1.6	2.4	5.1	6.1	9.1
OM3: N_{SCF} = 22, N_{diag} = 4						
time	1818.91	1148.49	720.07	342.92	287.18	190.59
time/N_{SCF}	82.68	52.20	32.73	15.59	13.05	8.66
speedup		1.6	2.5	5.3	6.3	9.5

[a] *n* is the number of CPU cores in use. The number of SCF iterations and full diagonalizations is denoted by N_{SCF} and N_{diag}, respectively. OpenMP parallelization was turned on using 6 CPU cores in H6CG* (our best effort).

(H6CG*), indicating again a similar computational burden for the different semiempirical methods in these environments.

In the CPU-only case, the speedups of the multicore calculations do not increase in proportion to the number of cores, reaching factors of about 1.5 on two cores (2C) and only around 3 on six cores (6C) regardless of the type of test molecule (see Table 3 and the Supporting Information). This is because the computation is bandwidth-bound on the present hardware for two of the major bottlenecks, FDIAG and the JACOBI part of PDIAG; adding more processing units hardly helps under these circumstances. It should be noted in this context that this limitation arises from the chosen hardware and not from the software: early tests of the MNDO program on a Cray Y-MP8 computer with eight CPU cores and comparatively fast shared memory gave speedups up to 7.7 in the MNDO geometry optimization of the fullerene C_{540}.[27] Repeating this calculation with identical input options on the current hardware (and with OpenMP parallelization being turned on) yields a smaller speedup of 4.0 on six CPU cores, because the higher clock frequency of the current processor (3 GHz vs 80 MHz on the Cray Y-MP) is not matched by a corresponding increase in memory bandwidth. The MNDO code will thus offer higher speedups in multicore CPU-only calculations on large molecules when the hardware performance is compute-bound (and not bandwidth-bound as in the present case).

For the chosen representative example of $1000{\cdot}H_2O$, the hybrid CPU–GPU platform with only one CPU core (H1CG) offers speedups of 5.1–7.0 relative to the CPU-only case (1C) due to the higher FLOP rate and bandwidth of the GPU (Table 3). H1CG outperforms the best CPU-only platform (6C) by a wide margin (speedup factors of 2.3–3.5). Adding more CPU cores to H1CG further accelerates the SYEVD routine (see Figure 3 in the Supporting Information) and leads to speedups of 6.1–8.9 for H6CG relative to 1C. Turning on OpenMP parallelization (H6CG*) for the other parts of the code leads to further (rather small) gains in the case of the MNDO-type calculations where integral evaluation is fast anyway, but to substantial accelerations especially for OM2 and OM3 where the computation of the three-center orthogonalization corrections strongly benefits from this kind of parallelization. As a consequence, HC6G* provides an excellent overall performance for all semiempirical methods considered presently, with speedup factors for $1000{\cdot}H_2O$ of 8.9–9.8 (Table 3).

Average speedups (H6CG and H6CG* vs 1C) for large molecules (N > 1000) are summarized in Table 4. Those for the MNDO-type methods are quite uniform because of the similar underlying theoretical framework, both for H6CG (8.1–8.3) and for H6CG* (8.4–8.7). In the case of the OMx methods, the average speedups for OM2 and OM3 are again relatively low (5.6–5.7) because the orthogonalization corrections are computed using single-threaded CPU code, which can consume up to 7% of the computation time in the CPU-only case (Figure 2). After turning on OpenMP parallelization (H6CG*), the average speedups for the OMx methods increase to a rather uniform level (7.3–7.7) but remain slightly below those for the MNDO-type methods.

Table 4. Average Speedups of Semiempirical Single-Point Energy Evaluations on the Hybrid Multicore CPU–GPU Platform (H6CG and H6CG*) Relative to the Single-CPU Case (1C), for Large Molecules with More than 1000 Basis Functions[a]

	MNDO-type methods			OMx methods		
	MNDO	AM1	PM3	OM1	OM2	OM3
H6CG	8.3	8.2	8.1	7.1	5.6	5.7
H6CG*	8.7	8.6	8.4	7.7	7.3	7.5

[a] OpenMP parallelization was turned on using 6 CPU cores in H6CG* (our best effort).

Figures 4–6 show the speedups obtained in the MNDO, AM1, PM3, and OMx calculations for all test molecules on the H6CG and H6CG* platforms as compared to the single CPU case (1C); the corresponding numerical results are available in the Supporting Information.

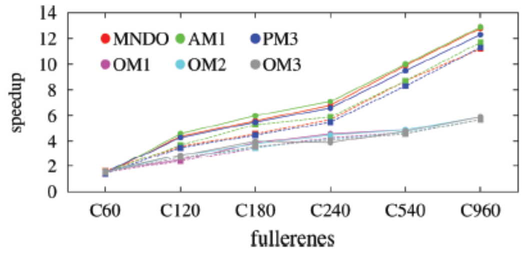


Figure 4. Speedup of the semiempirical single-point energy calculations of fullerenes on the hybrid multicore CPU–GPU platform (H6CG, dashed lines with squares; and H6CG*, solid lines with circles) with respect to one single CPU core (1C). The MNDO-type and OMx calculations employ pseudodiagonalization (without DIIS) and full diagonalization (with DIIS), respectively.

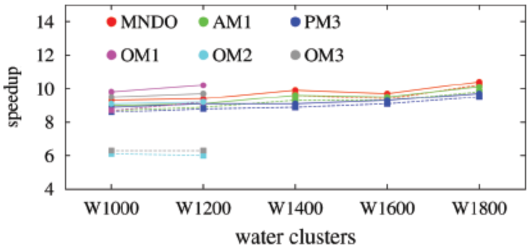


Figure 5. Speedup of the semiempirical single-point energy calculations of water clusters on the hybrid multicore CPU–GPU platform (H6CG, dashed lines with squares; and H6CG*, solid lines with circles) with respect to one single CPU core (1C). The calculations make use of pseudodiagonalization (without DIIS).

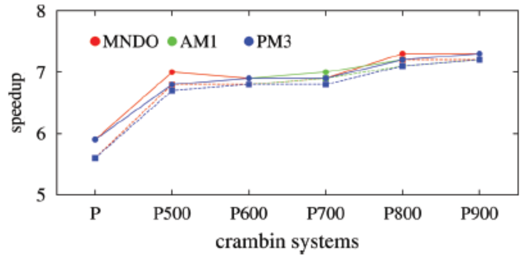


Figure 6. Speedup of the semiempirical single-point energy calculations of solvated crambin systems on the hybrid multicore CPU–GPU platform (H6CG, dashed lines with squares; and H6CG*, solid lines with circles) with respect to one single CPU core (1C). The calculations employ full diagonalization and DIIS extrapolation (no use of pseudodiagonalization).

The overall speedups generally increase with the size of the test molecules and the number of basis functions (N) ranging from 240 to 10 800. Small molecules like C_{60} benefit only little from the GPU. For example, relative to 1C, MNDO-type and OM*x* calculations on C_{60} are merely ~1.5 times faster on H6CG and H6CG*. Several factors restrict the performance for smaller molecules. First, routines other than those ported to the GPU take a higher percentage of the computation time, for example, 12.1% for C_{60} versus 1.6% for C_{960} (see Figure 2a). Second, the CPU-only computations for small molecules are really fast, for example, 0.1 s for C_{60}, so that the overhead for CPU−GPU communication on the hybrid platform is no longer negligible. Hence, there is not much advantage of using the GPU for semiempirical calculations on small molecules (say, less than 100 atoms).

In the fullerene and water cluster series, the MNDO-type calculations for the largest systems reach maximum speedups on the hybrid multicore CPU−GPU platform of around 13 (C_{960}) and 10 (1800·H_2O). Because of the memory limitations on our workstation, OM*x* calculations are possible for systems up to the size of 1200·H_2O, for which speedups of 9−10 can be achieved (Figure 5).

In the MNDO-type calculations on the solvated crambin series, the observed speedups are generally somewhat lower (around 7, see Figure 6). This is mostly due to the deliberate decision to turn off pseudodiagonalizations (PDIAG) and use only full diagonalizations (FDIAG) in these calculations. FDIAG, which dominates the computation time in this case (>60%, see Figure 2), is accelerated typically only 5-fold on the GPU (see Figure 3 in the Supporting Information), much less so than PDIAG.

Finally, it should be pointed out that the number of required SCF iterations may occasionally differ on different platforms, despite using double-precision floating point arithmetic throughout (see the Supporting Information). These differences arise from roundoff errors and are unavoidable. In such cases, the calculated properties of the molecules, for example, heats of formation, ionization energies, etc., are the same on the different platforms (within the limits imposed by the SCF convergence criteria).

5.2. Profiles on the Hybrid Multicore CPU−GPU Platform.

In this section, we examine the hotspots in semiempirical calculations on the hybrid CPU−GPU platform (H6CG). For the reasons outlined in section 3, we only show the H6CG profiles of the AM1 and OM2 calculations for all test systems in Figure 7. The data for the other methods are documented in the Supporting Information.

It is obvious that Fock matrix diagonalization (FDIAG, in red) and pseudodiagonalization (PDIAG, in green) demand a larger share of the computation time than on the CPU-only platform (Figure 2). For example, the percentage of time consumed by FDIAG grows from 64% to 82% in the MNDO calculation of unsolvated crambin. The reason for this larger share is that both SYEVD in FDIAG and the Jacobi transformation in PDIAG are bandwidth-bound and are thus not accelerated too much on the GPU, that is, typically by a factor of ~5; see Figure 3 in the Supporting Information. By contrast, the major computations in BORDER and DIIS, that is, the matrix multiplications in SYRK and GEMM, are dramatically accelerated on the GPU (by a factor of about 25, see Figure 4 in the Supporting Information), and they are thus no longer bottlenecks in semiempirical calculations on H6CG. The other routines, for example, integral evaluation and construction of the Fock matrix, have not yet been ported to the GPU in our present work. Their share of the computation time starts to rise on H6CG especially for small molecules, but remains rather low for large molecules in MNDO-type calculations.

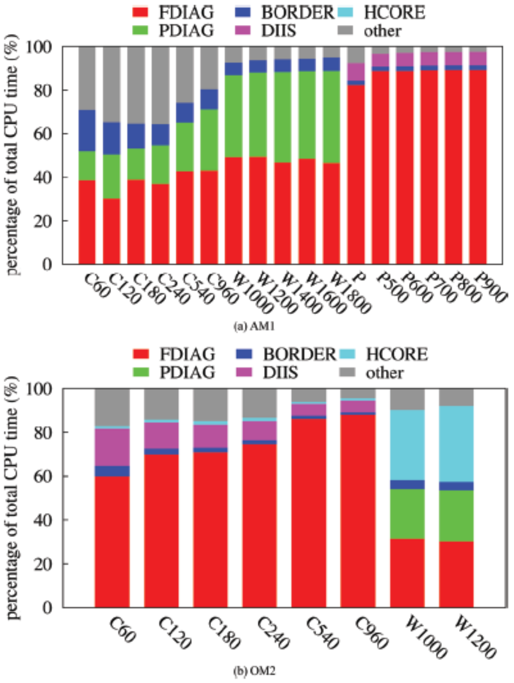


Figure 7. Profiles of the semiempirical AM1 and OM2 calculations of the test molecules on the hybrid multicore CPU−GPU computing platform.

Figure 8 presents the profiles of PDIAG in the MNDO calculations on H6CG. Unlike the situation on a single CPU core (Figure 3) where the triple matrix product (MMM) consumes more time than the Jacobi transformation (JACOBI), the latter takes most of the computation time on the GPU (~80%). This is because the matrix−matrix multiplication is far more accelerated on the GPU than the Jacobi transformation, with speedups of about 25 (see Figure 4 in the Supporting Information) versus 4−7 (Table 2).

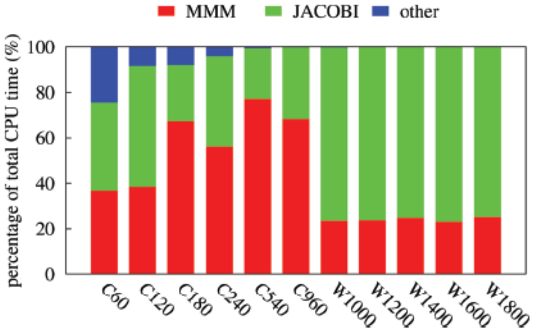


Figure 8. The profiles of PDIAG in the MNDO calculations of fullerenes and water clusters on the hybrid CPU−GPU platform.

The H6CG profiles of the OM*x* calculations on water clusters look quite different from the other profiles, particularly in the case of OM2 and OM3; see Figure 7b. The computations in HCORE become quite demanding, taking about 32% and

33% of the time for OM2 and OM3, respectively. This latter increase is caused by the three-center orthogonalization corrections to the resonance integrals,[26] which are presently still computed in HCORE by a single-threaded CPU code. These corrections are crucial for the quality of the OM*x* results.[33−35] One way to accelerate their computation is to turn on OpenMP parallelization (H6CG*, see above), but it would also seem worthwhile to develop a dedicated GPU kernel and find out whether this offers an even more efficient solution on hybrid CPU−GPU platforms.

5.3. Geometry Optimizations. As a further test, geometry optimizations of the fullerenes were carried out at the MNDO level using the 1C, H6CG, and H6CG* platforms. In the MNDO program, the required gradients are computed by default using a simple finite-difference procedure with recalculated integrals and a constant density matrix. This procedure scales as $O(N^2)$ and thus does not constitute a bottleneck in the present approach, but will benefit from the existing OpenMP parallelization (in full analogy to integral evaluation). Therefore, the geometry optimizations were performed without any additional GPU-oriented modification of the code using both H6CG and H6CG*. They were done in internal coordinates to exploit the symmetry of the fullerenes. The performance data are shown in Table 5. The speedups increase with the size of the fullerene being optimized, as expected, and they are generally somewhat higher than those obtained in the single-point calculations (see the Supporting Information). The improved performance arises from the fact that the relatively slow full diagonalizations (FDIAG) at the beginning of an SCF calculation can normally be avoided in the course of a geometry optimization because the density matrix from the preceding point usually provides a sufficiently good starting guess for the SCF procedure of the next point. For the largest fullerene considered (C_{960}), the MNDO geometry optimizations on the hybrid multicore CPU−GPU platform thus achieve record speedups of 15.5 (H6CG) and 19.2 (H6CG*).

Table 5. Computation Times (in seconds) for MNDO Geometry Optimization of Fullerenes on a Single CPU Core (1C) and on the Hybrid Multicore CPU−GPU Platform (H6CG and H6CG*)[a]

	C_{60}	C_{120}	C_{180}	C_{240}	C_{540}	C_{960}
N_{opt}	4	49[b]	14	28	34	40[c]
symmetry	I_h	T_d	I_h	I_h	I_h	I_h
N_{var}	2	17	6	7	15	26
1C	0.46	59.41	42.68	274.49	4927.51	57 651.94
H6CG	0.29	14.76	8.27	39.88	398.28	3731.21
speedup	1.5	4.0	5.2	6.9	12.4	15.5
H6CG*	0.24	10.37	5.82	28.54	309.63	3007.62
speedup*	1.9	5.7	7.3	9.6	15.9	19.2

[a] N_{opt} and N_{var} denote the number of the optimization cycles and geometric variables, respectively. OpenMP parallelization was turned on using 6 CPU cores in H6CG* (our best effort). [b] The geometry optimization of C_{120} finishes ($||g||$ = 0.05 kcal/(mol·Å)) in 44 cycles on H6CG. [c] An incomplete optimization for benchmark purposes with 40 cycles.

6. CONCLUSIONS

In this work, GPU-accelerated routines for semiempirical quantum chemical calculations were implemented on a hybrid multicore CPU−GPU platform to enable more efficient computations with MNDO-type methods (for MNDO, AM1, PM3) and with orthogonalization-corrected methods (OM1, OM2, OM3). Systematic calculations on fullerenes, water clusters, and solvated crambin systems were performed with all of these methods on a variety of computing environments encompassing one single CPU core, multiple CPU cores, a single CPU core with GPU, and a multicore CPU with GPU.

For all methods considered, the computational bottlenecks on one single CPU core were identified to be full diagonalization (FDIAG), pseudodiagonalization (PDIAG), and matrix multiplications during density matrix formation (BORDER) and SCF convergence acceleration (DIIS). The computational effort in other routines, for example, integral evaluation and Fock matrix construction, turned out to be small for large molecules. Vendor-optimized library functions and a manually tuned GPU kernel for Jacobi transformations (PDIAG) were employed in the profile-guided code optimization on the hybrid multicore CPU−GPU platform. The least accelerated routines on the GPU were found to be FDIAG and PDIAG, which both contain parts that are memory-bandwidth bound. Therefore, FDIAG and PDIAG govern the overall speedup that can be achieved in semiempirical calculations on the hybrid CPU−GPU platform.

The performance gains in the semiempirical calculations increase with the size of the molecule. On average, speedups close to 10 are achieved in MNDO-type and OM*x* single-point calculations for large molecules, on the multicore CPU−GPU platform relative to one single CPU core, with a maximum of 12.9 for the AM1 energy evaluation of the largest fullerene (C_{960}, 3840 basis functions). MNDO geometry optimizations for the largest fullerenes on the hybrid multicore CPU−GPU platform reach an even higher performance (with speedups up to 19.2). The OM*x* calculations are slightly less accelerated than the MNDO-type calculations because of the orthogonalization corrections to the core Hamiltonian. Their computation requires a significant share of the overall time on the hybrid platform (especially for the three-center terms in OM2 and OM3). While their evaluation benefits from the available OpenMP parallelization, it would still seem desirable to develop a carefully optimized GPU kernel for this task in future work.

ASSOCIATED CONTENT

Supporting Information

Further profiles of semiempirical SCF calculations on CPU and hybrid CPU-GPU platforms (MNDO, PM3, OM1, and OM3), performance tests of standard library routines, detailed description of the GPU algorithms designed for the Jacobi transformation, benchmarks of the optimized GPU kernel for this transformation, and computation times for the semiempirical calculations of the test molecules on all computing platforms. This material is available free of charge via the Internet at http://pubs.acs.org.

AUTHOR INFORMATION

Corresponding Author

*E-mail: thiel@mpi-muelheim.mpg.de.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We would like to thank Wolfgang Angenendt for implementing parts of the OpenMP parallelization.

■ REFERENCES

(1) Clark, T.; Stewart, J. J. P. In *Computational Methods for Large Systems*; Reimers, J. R., Ed.; John Wiley & Sons, Inc.: Hoboken, NJ, 2011; pp 259−286.

(2) Thiel, W. In *Theory and Applications of Computational Chemistry*; Dykstra, C. E., Frenking, G., Kim, K. S., Scuseria, G. E., Eds.; Elsevier: Amsterdam, 2005; pp 559−580.

(3) Bredow, T.; Jug, K. *Theor. Chem. Acc.* **2005**, *113*, 1−14.

(4) Thiel, W. In *Modern Methods and Algorithms of Quantum Chemistry*; Grotendorst, J., Ed.; John von Neumann Institute for Computing: Jülich, 2000; pp 261−283.

(5) Clark, T. *J. Mol. Struct. (THEOCHEM)* **2000**, *530*, 1−10.

(6) Thiel, W. *Advances in Chemical Physics*; John Wiley & Sons, Inc.: New York, 1996; pp 703−757.

(7) Stewart, J. J. P. *Reviews in Computational Chemistry*; John Wiley & Sons, Inc.: New York, 1990; pp 45−81.

(8) Thiel, W. *Tetrahedron* **1988**, *44*, 7393−7408.

(9) Dewar, M. J. S.; Thiel, W. *Theor. Chim. Acta* **1977**, *46*, 89−104.

(10) *NVIDIA CUDA C Programming Guide*; NVIDIA Corp.: Santa Clara, CA, 2010.

(11) Stone, J. E.; Hardy, D. J.; Ufimtsev, I. S.; Schulten, K. *J. Mol. Graphics Modell.* **2010**, *29*, 116−125.

(12) Farber, R. M. *J. Mol. Graphics Modell.* **2011**, *30*, 82−89.

(13) van der Spoel, D.; Hess, B. *WIREs Comput. Mol. Sci.* **2011**, *1*, 710−715.

(14) Anderson, A. G.; Goddard, W. A., III; Schröder, P. *Comput. Phys. Commun.* **2007**, *177*, 298−306.

(15) (a) Ufimtsev, I. S.; Martínez, T. J. *Comput. Sci. Eng.* **2008**, *10*, 26−34. (b) Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2008**, *4*, 222−231. (c) Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2009**, *5*, 1004−1015. (d) Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2009**, *5*, 2619−2628. (e) Luehr, N.; Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2011**, *7*, 949−954. (f) Isborn, C. M.; Luehr, N.; Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2011**, *7*, 1814−1823.

(16) (a) Yasuda, K. *J. Comput. Chem.* **2008**, *29*, 334−342. (b) Yasuda, K. *J. Chem. Theory Comput.* **2008**, *4*, 1230−1236.

(17) Asadchev, A.; Allada, V.; Felder, J.; Bode, B. M.; Gordon, M. S.; Windus, T. L. *J. Chem. Theory Comput.* **2010**, *6*, 696−704.

(18) (a) Vogt, L.; Olivares-Amaya, R.; Kermes, S.; Shao, Y.; Amador-Bedolla, C.; Aspuru-Guzik, A. *J. Phys. Chem. A* **2008**, *112*, 2049−2057. (b) Olivares-Amaya, R.; Watson, M. A.; Edgar, R. G.; Vogt, L.; Shao, Y.; Aspuru-Guzik, A. *J. Chem. Theory Comput.* **2010**, *6*, 135−144.

(19) (a) DePrince, A. E.; Hammond, J. R. *J. Chem. Theory Comput.* **2011**, *7*, 1287−1295. (b) Ma, W.; Krishnamoorthy, S.; Villa, O.; Kowalski, K. *J. Chem. Theory Comput.* **2011**, *7*, 1316−1327.

(20) Wilkinson, K. A.; Sherwood, P.; Guest, M. F.; Naidoo, K. J. *J. Comput. Chem.* **2011**, *32*, 2313−2318.

(21) Mangueira, C. P., Jr.; Carvalho, J. D.; Cabral, L. A. F.; Rocha, G. B. *XVI Brasilian Symposium of Theoretical Chemistry*; Ouro Preto, 21 November, 2011; Poster P377.

(22) Thiel, W. *MNDO99 CVS Development Version*; Max-Planck-Institut für Kohlenforschung: Mülheim an der Ruhr, Germany, 2012.

(23) (a) Dewar, M. J. S.; Thiel, W. *J. Am. Chem. Soc.* **1977**, *99*, 4899−4907. (b) Dewar, M. J. S.; Thiel, W. *J. Am. Chem. Soc.* **1977**, *99*, 4907−4917.

(24) Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. *J. Am. Chem. Soc.* **1985**, *107*, 3902−3909.

(25) (a) Stewart, J. J. P. *J. Comput. Chem.* **1989**, *10*, 209−220. (b) Stewart, J. J. P. *J. Comput. Chem.* **1989**, *10*, 221−264.

(26) (a) Kolb, M.; Thiel, W. *J. Comput. Chem.* **1993**, *14*, 775−789. (b) Weber, W.; Thiel, W. *Theor. Chem. Acc.* **2000**, *103*, 495−506. (c) Scholten, M. Semiempirische Verfahren mit Orthogonalisierungskorrekturen: Die OM3 Methode. Ph.D. thesis, University of Düsseldorf, 2003.

(27) Bakowies, D.; Thiel, W. *J. Am. Chem. Soc.* **1991**, *113*, 3704−3714.

(28) Bakowies, D.; Bühl, M.; Thiel, W. *J. Am. Chem. Soc.* **1995**, *117*, 10113−10118.

(29) Martínez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M. *J. Comput. Chem.* **2009**, *30*, 2157−2164.

(30) *CUDA CUBLAS Library*; NVIDIA Corp.: Santa Clara, CA, 2010.

(31) Tomov, S.; Nath, R.; Du, P.; Dongarra, J. *MAGMA Users' Guide*; Innovative Computing Laboratory, University of Tennessee: Knoxville, TN, 2010.

(32) Pople, J. A.; Santry, D. P.; Segal, G. A. *J. Chem. Phys.* **1965**, *43*, S129−S135.

(33) Otte, N.; Scholten, M.; Thiel, W. *J. Phys. Chem. A* **2007**, *111*, 5751−5755.

(34) Korth, M.; Thiel, W. *J. Chem. Theory Comput.* **2011**, *7*, 2929−2936.

(35) Silva-Junior, M. R.; Thiel, W. *J. Chem. Theory Comput.* **2010**, *6*, 1546−1564.

(36) i (i = 60, 120, 180, 240, 540, 960) is the number of carbon atoms in the fullerenes. j (j = 1000, 1200, 1400, 1600, 1800) and k (k = 500, 600, 700, 800, 900) specify the number of water molecules in the water clusters and in the solvation spheres around crambin, respectively. An isolated crambin (642 atoms) is denoted as P.

(37) Intel Turbo Boost Technology, which increases the CPU clock depending on the workload, has been switched off to obtain consistent benchmark timings.

(38) For example, in the case of crambin·900H_2O, 6 DIIS cycles in double precision demand about 3 GB memory on the GPU, which combined with the requirements by other subroutines approaches the memory capacity of the GPU.

(39) Pulay, P. *J. Comput. Chem.* **1982**, *3*, 556−560.

(40) Thiel, W.; Green, D. G. In *Methods and Techniques in Computational Chemistry: METECC-95*; Clementi, E., Corongiu, G., Eds.; METECC (Series); STEF: Cagliari, 1995; pp 141−168.

(41) Stewart, J. J. P.; Császár, P.; Pulay, P. *J. Comput. Chem.* **1982**, *3*, 227−228.

(42) Detailed information on all tests is given in the Supporting Information.

(43) Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Demmel, J.; Dongarra, J.; Croz, J. D.; Greenbaum, A.; Hammarling, S.; McKenney, A.; Sorensen, D. *LAPACK Users' Guide*, 3rd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, 1999.

(44) The three steps in the MAGMA SYEVD implementation are (i) reduce a real symmetric matrix to tridiagonal form (SYTRD on GPU), (ii) compute all eigenvalues and eigenvectors of a symmetric tridiagonal matrix using the divide and conquer method (STEDC on CPU), and (iii) multiply a real matrix by a previously determined orthogonal matrix (DORMTR on GPU).

(45) Novakovic, V.; Singer, S., arXiv:1008.1371v2.

(46) Modi, J. J.; Parkinson, D. *Comput. Phys. Commun.* **1982**, *26*, 317−320.

(47) Berry, M.; Sameh, A. *J. Comp. Appl. Math.* **1989**, *27*, 191−213.

(48) Eberlein, P. J.; Park, H. *J. Par. Dist. Comp.* **1990**, *8*, 358−366.

Supporting information for: Semiempirical Quantum Chemical Calculations Accelerated on a Hybrid Multi-core CPU-GPU Computing Platform

Xin Wu, Axel Koslowski, and Walter Thiel*

Max-Planck-Institut für Kohlenforschung, Kaiser-Wilheim-Platz 1,
45470 Mülheim an der Ruhr, Germany

E-mail: thiel@mpi-muelheim.mpg.de

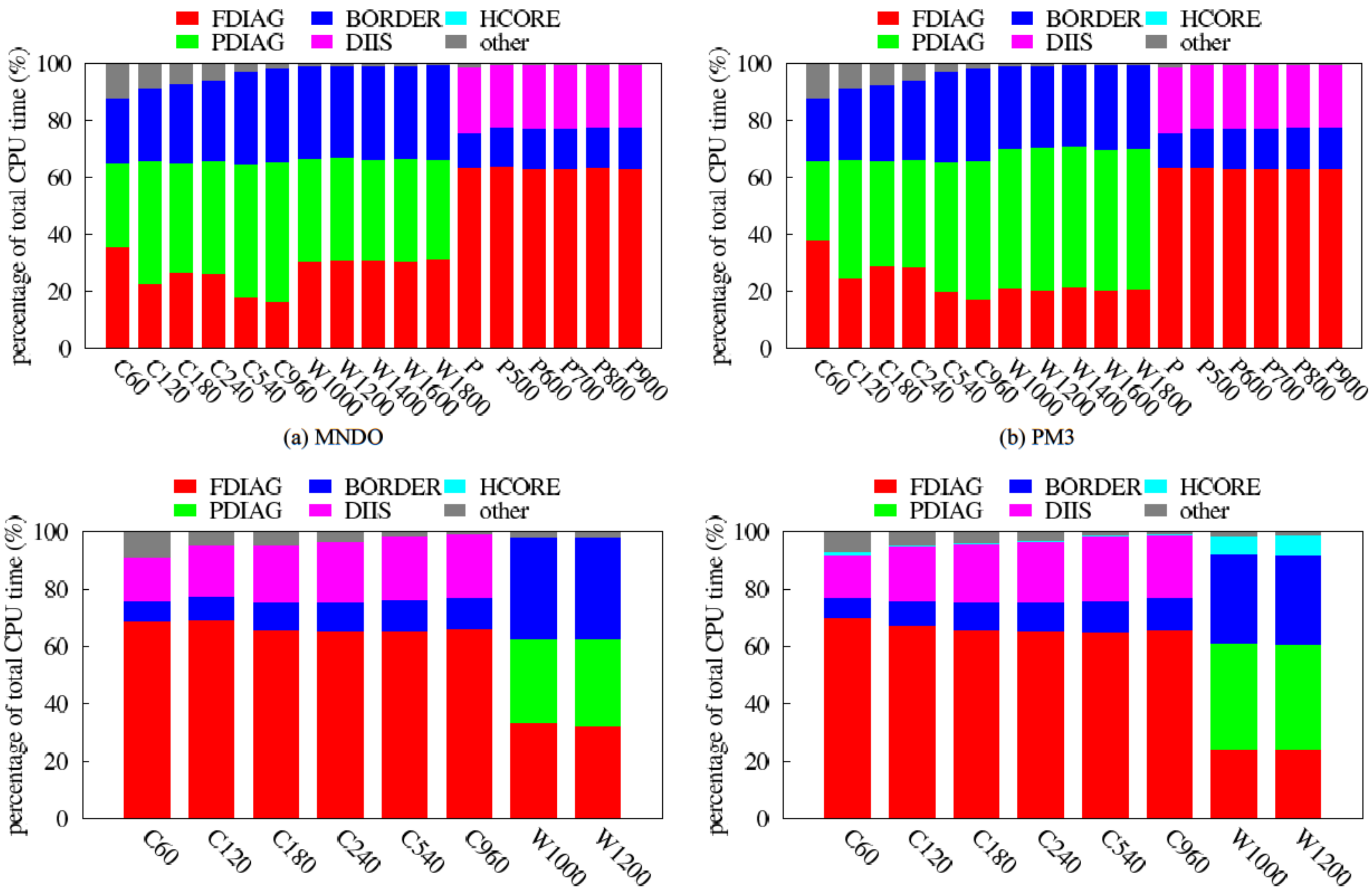
*To whom correspondence should be addressed

1 Profiles of Semiempirical SCF Calculations

Figure 1 shows the profiles of single-point SCF calculations for the methods MNDO, PM3, OM1, and OM3 obtained with the MNDO program running on one CPU core (1C). Figure 2 shows the corresponding profiles obtained on the hybrid CPU-GPU platform using all six CPU cores and the GPU (H6CG).

2 Performance Tests of Standard Library Routines

Figure 3 compares the performance of matrix diagonalization routines from the MKL and MAGMA libraries. Figures 4 and 5 show analogous comparisons of routines for matrix-matrix and matrix-vector multiplication from the MKL and CUBLAS libraries, respectively.



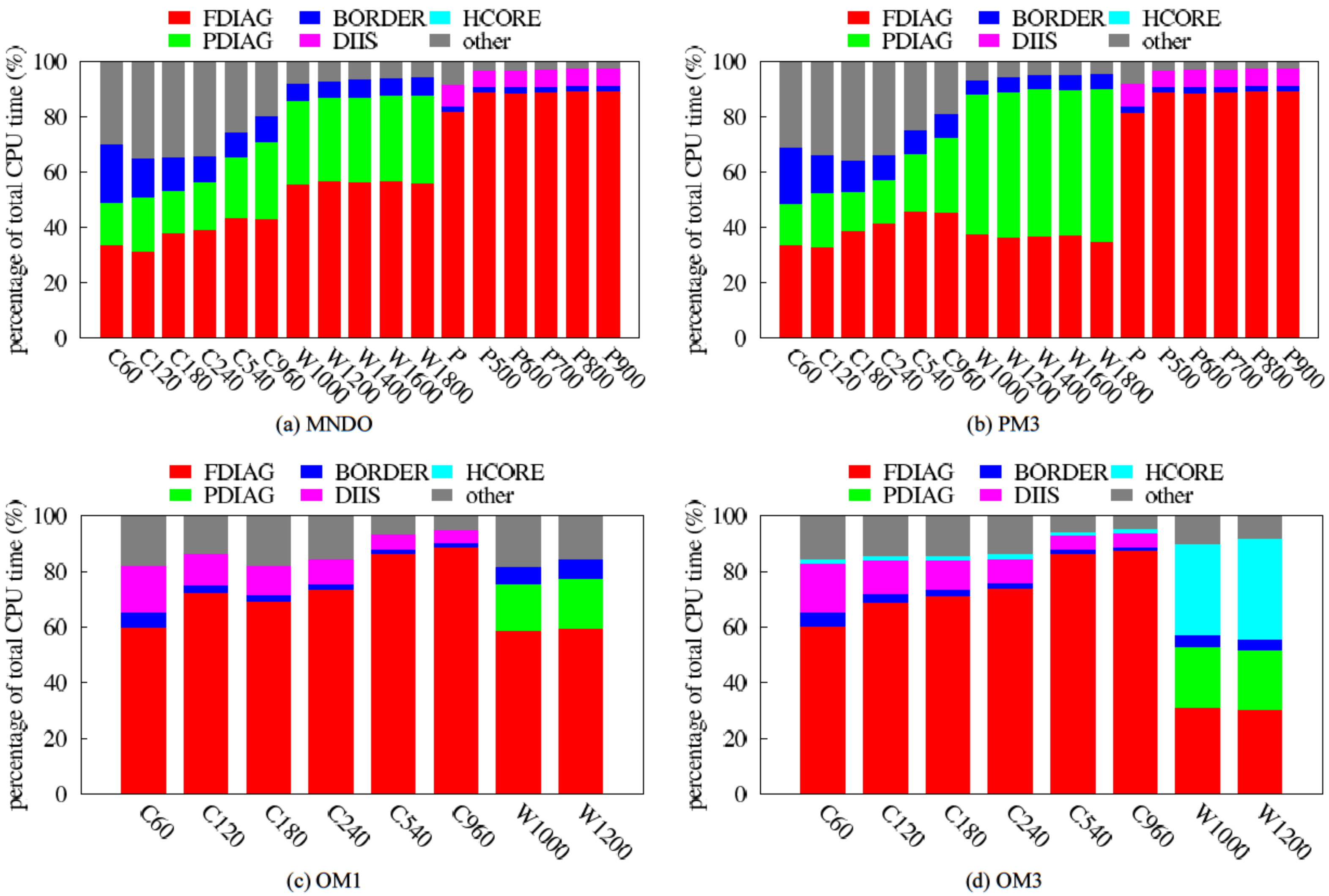
(a) MNDO

(b) PM3

(c) OM1

(d) OM3

Figure 1: Profiles of semiempirical calculations of the test molecules on a single CPU core.



(a) MNDO

(b) PM3

(c) OM1

(d) OM3

Figure 2: Profiles of semiempirical calculations of the test molecules on the hybrid multi-core CPU-GPU computing platform.

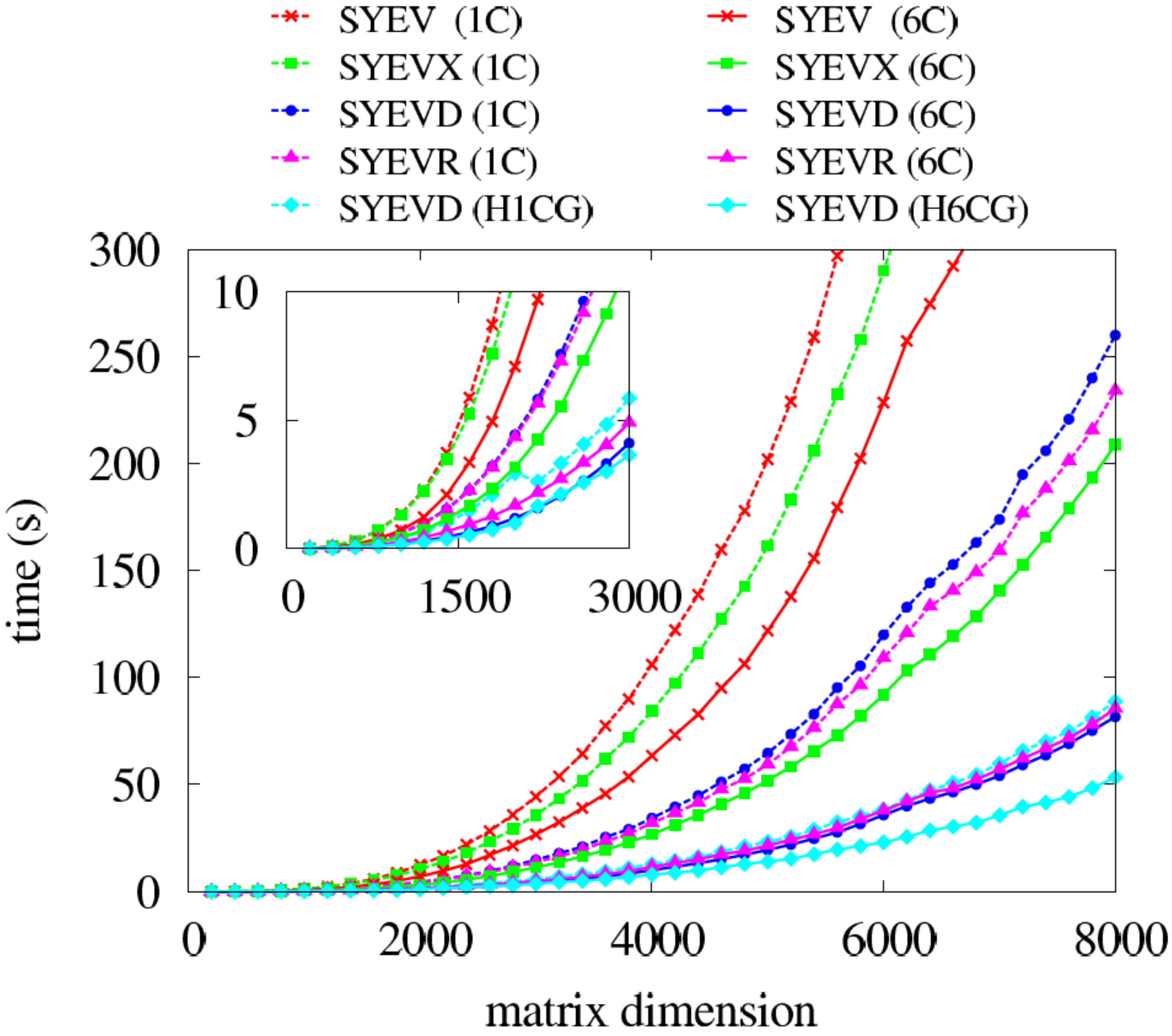


Figure 3: Computation times (in seconds) for the diagonalization of a random real symmetric matrix using SYEV, SYEVX, SYEVD and SYEVR from Intel MKL on CPU-only platforms (*n*C) and SYEVD from MAGMA on hybrid CPU-GPU platforms (H*n*CG). *n* denotes the number of CPU cores. An enlarged plot for smaller matrices (dimension < 3000) is also shown. The floating-point arithmetic was performed in double precision.

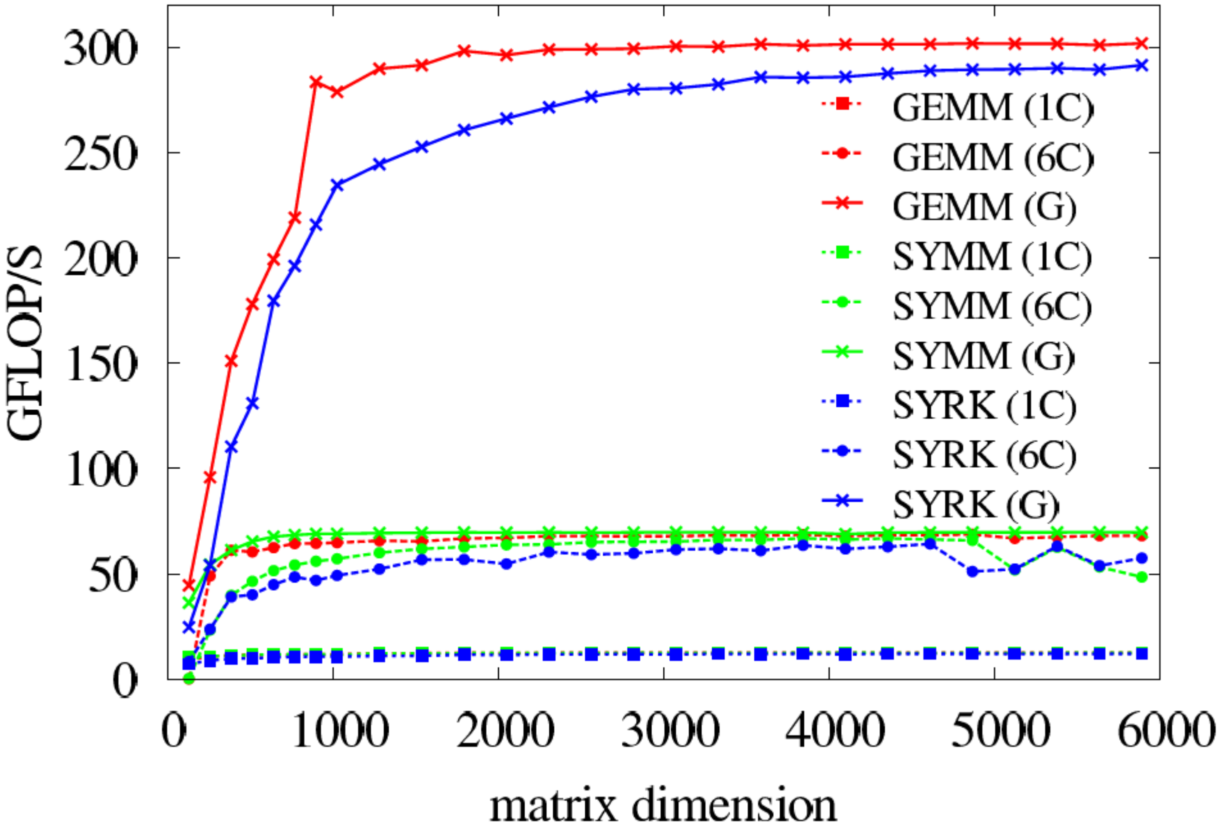
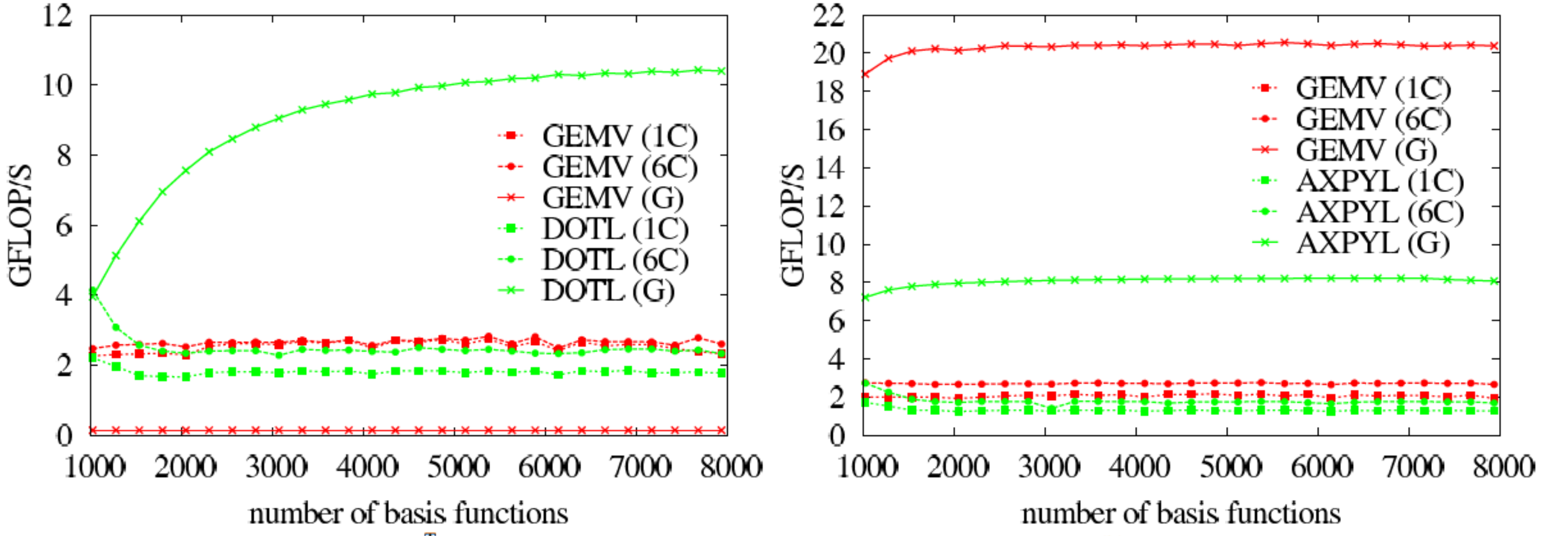


Figure 4: Performance of matrix-matrix multiplication ($\mathbf{P} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$) using GEMM, SYMM, and SYRK from Intel MKL on CPU-only platforms (nC) and the corresponding routines from NVidia CUBLAS on the hybrid CPU-GPU platforms (G). n denotes the number of CPU cores. **A** is a random real symmetric matrix. The floating-point arithmetic was performed in double precision.



(a) $\mathbf{p} = \mathbf{M}^{\mathrm{T}}\mathbf{v}$

(b) $\mathbf{p} = \mathbf{M}\mathbf{v}$

Figure 5: Performance of matrix-vector multiplications using GEMV, DOT, and AXPY from Intel MKL on CPU-only platforms (nC) and the corresponding routines from NVidia CUBLAS on the hybrid CPU-GPU platforms (G). n denotes the number of CPU cores. DOTL and AXPYL are used to compute $\mathbf{p} = \mathbf{M}^{\mathrm{T}}\mathbf{v}$ and $\mathbf{p} = \mathbf{M}\mathbf{v}$, respectively. $\mathbf{M}$ and $\mathbf{v}$ are random real matrix and vector, respectively. The row dimension of the matrix is $\frac{1}{2}N(N+1)$ for N basis functions, the column dimension is equal to 10. The floating-point arithmetic was performed in double precision.

3 Jacobi Transformation on the GPU

As stated in the manuscript, the original CPU algorithm in PDIAG performs sweeps of independent 2×2 rotations of one occupied and one virtual molecular orbital (MO) trial vector from the last SCF iteration, C_i and C_a,

$$C_i' = cC_i - sC_a \qquad C_a' = sC_i + cC_a \tag{1}$$

where c and s denote the rotation coefficients and the new MO vectors are indicated by primes. The rotation coefficients are calculated from the approximate MO energies ε_i and ε_a of the previous SCF iteration and from the matrix element $\mathscr{F}_{ia}$ of the Fock matrix in the MO basis connecting both orbitals.

$$c = 1 - \frac{u}{2} \qquad s = \pm\sqrt{u - \frac{u^2}{4}} \qquad u = \left(\frac{\mathscr{F}_{ia}}{\varepsilon_a - \varepsilon_i}\right)^2 \tag{2}$$

The sign of s is opposite to that of the Fock matrix element.

During one sweep all rotations on a set of independent occupied-virtual orbital pairs are performed that satisfy a predefined threshold criterion (see Ref. (41) in the manuscript). If, respectively, n_o and n_v are the number of occupied and virtual molecular orbitals, and n_small and n_large are the smaller and the larger number of n_o and n_v, there are n_large sweeps and n_small orbital pairs examined per sweep. In the first sweep, the first pair is formed by the lowest occupied and the lowest virtual orbital, the second pair by the second-lowest occupied and the second-lowest virtual orbital, and so on. In the second sweep, the first pair is formed by the second-lowest occupied and the lowest virtual orbital, the second pair by the third-lowest occupied and the second-lowest virtual orbital and so on, provided that the number of virtual orbitals is less than the number of occupied orbitals, which is usually the case for organic molecules. Otherwise, the first pair is formed by the lowest occupied and the second-lowest virtual orbital, and so on. Using this algorithm every occupied-virtual orbital pair is examined exactly once and rotated where appropriate.

This algorithm is to be ported to the GPU, taking into account that the architecture of a GPU is very different from a traditional CPU. One fundamental difference concerns the cost of creating, deleting, and switching between threads. While on a CPU these actions are relatively expensive, so that a programmer is advised to reuse a once created thread as often as possible, and elaborate algorithms have been developed to efficiently schedule different processes and threads by the operating system kernel, all information about the threads on a GPU are stored in hardware registers so that creating, deleting, and switching between threads is very cheap or involves no cost at all. This allows GPU threads to be extremely numerous and short, like millions of threads containing a single arithmetic operation only.

On a GPU, threads are grouped in blocks and blocks are grouped in a grid, concepts totally absent on a traditional CPU. Each block and each thread have up to three indices uniquely identifying a thread in a block and a block in the grid. The dimensions of all indices (all blocks being equivalent) are called the execution configuration.[1] The dimensions and indices can be used to decide which thread operates on which matrix element, for instance. These quantities also determine in which order threads are created and executed.

It is, however, impossible for the programmer to know in advance which threads will run concurrently unless specific measures are taken to synchronize the threads. The threads of one block (which all execute on one core of one multiprocessor of the GPU) can be easily synchronized by a simple (and cheap) function call implementing a so-called barrier which makes the threads wait until all threads have reached this point. Then the function returns and all threads (at least potentially) continue simultaneously. This behaviour is called local synchronization. Threads in different blocks can presently only be synchronized by relaunching the kernel after it has completed a certain subtask and stopped (global synchronization).

We have designed three algorithms differing in the way how the work is distributed among the threads, how these are organized in blocks and how the threads are synchronized.

Algorithm I uses a two-dimensional grid of one-dimensional blocks. Every thread checks the threshold criterion and performs the rotation of two MO coefficients where appropriate. The first index of a block in the grid and the index of a thread in the block determine the row in

[1] The number of bytes of dynamically allocated shared memory per block is also subsumed under the execution configuration, as well as an associated stream.

the MO coefficient matrix, and the second block index identifies the virtual orbital.[2] Execution configurations with 32, 64, 128, 256, 512, 768, and 1024 threads per block have been tested. (1024 threads per block is the hardware limit of our GPU.) The kernel uses global synchronization of the threads by performing one sweep per launch.

Algorithm II completely avoids the synchronization problem using exactly one thread per row of the trial vectors, so each thread does all the work for this row very much like the original CPU algorithm (except that the original algorithm rotates entire vectors instead of two elements). Execution configurations with the same number of threads per block as in algorithm I have been tested.

Algorithm III uses a one-dimensional grid of two-dimensional blocks. The row index of the MO coefficient matrix is determined by the index of the block in the grid and the first thread index. Each thread performs all sweeps for every D_y-th virtual orbital of two subsequent rows, D_y being the second block dimension, starting with the second thread index in each block. The threads use local synchronization to make sure that one sweep on two rows has finished before the next sweep on these rows is started.

There are actually three distinct kernels for each of the above algorithms which differ in the treatment of the rotation coefficients. One option is to recompute the rotation coefficients in every thread as needed. Alternatively, the rotation coeffients are precomputed and stored in global memory before the first sweep, and for efficient access the kernel either relies on caching of these values (second option) or explicitly copies them to shared memory (third option).

For every of the above algorithms there is the possibility that the GPU will not run at full efficiency for the orbital pairs not rotated due to the threshold criterion. This is in part because some of the above kernels are designed such that the threads not performing a rotation will wait for other threads to have finished a rotation. Another reason is that the threads do not run completely independent from each other, but are executed in so-called warps, i.e. batches of 32 threads executed simultaneously by one GPU core. The core always executes the same instruction of the kernel

[2] The occupied orbital is selected using an offset to the virtual orbital which is incremented at every sweep.

for all threads in the warp. Therefore, while some rotations are performed, other threads may idle automatically.

Algorithm IV was designed with the goal to gain efficiency by avoiding to have threads waiting for others to finish rotations. In this algorithm the rotation coefficients are precomputed by a separate kernel, with only those being stored (together with the corresponding orbital indices) that satisfy the threshold criterion. This step requires a strict linearization, i.e. the rotation coefficients must be stored one after the other. This is achieved by an atomic increment of the array index pointing to the location where to store the next pair of rotation coefficients and orbital indices. The atomic increment consists of reading, modifying and writing the array index without interference of other threads.

The dimensions of the blocks and the grid and the synchronization method (local synchronization) of algorithm IV correspond to algorithm III, except that in algorithm IV each thread works on one row of the MO coefficient matrix only. This algorithm will rely on the caching mechanisms of the GPU exclusively, there are no other variants.

With this setup, algorithm IV consistently outperforms the other three algorithms by a large margin in systematic test calculations. For example, in the case of C_{960}, where the benefits from skipping rotations are largest, algorithm IV is more than five times faster than any of the three competitors. Therefore, we present representative timings only for algorithm IV. Tables 1 and 2 contain the timings of the two steps of algorithm IV for different execution configurations.

Table 1: Computation times (in seconds, sorted in ascending order) for generating the rotation coefficients for the Jacobi transformation in MNDO calculations of C_{960}, $1000 \cdot H_2O$, and solvated crambin using GPU algorithm IV. blockDim denotes the dimension of the thread blocks.

C_{960}		$1000 \cdot H_2O$		crambin	
blockDim	time	blockDim	time	blockDim	time
16×8	0.03	16×8	0.06	16×8	0.01
8×16	0.03	8×16	0.06	8×16	0.01
8×32	0.03	32×8	0.06	32×8	0.02
32×8	0.03	128×7	0.07	8×8	0.02
16×16	0.03	8×8	0.07	128×7	0.02
8×8	0.04	16×16	0.07	16×16	0.02
256×3	0.04	256×3	0.07	256×3	0.02
128×7	0.04	8×112	0.07	8×32	0.02
64×14	0.04	8×32	0.07	64×14	0.02
16×56	0.05	64×14	0.08	32×28	0.02
32×28	0.05	16×56	0.08	8×112	0.02
8×112	0.05	32×28	0.08	16×56	0.02
8×64	0.06	64×8	0.09	64×8	0.02
16×32	0.06	32×16	0.10	32×16	0.02
64×8	0.06	8×64	0.10	16×32	0.03
32×16	0.06	16×32	0.11	8×64	0.03

Table 2: Computation times (in seconds, sorted in ascending order) for the Jacobi transformation step in MNDO calculations of C_{960}, $1000 \cdot H_2O$, and solvated crambin using GPU algorithm IV. blockDim denotes the dimension of the thread blocks.

C_{960}		$1000 \cdot H_2O$		crambin	
blockDim	time	blockDim	time	blockDim	time
32×32	3.74	128×8	36.13	256×4	2.61
64×16	3.79	64×16	36.48	128×8	2.67
16×32	3.88	64×8	36.48	64×16	2.68
32×8	3.93	256×4	36.65	64×8	2.80
16×64	4.15	32×32	36.94	32×32	2.87
32×16	4.18	32×16	36.95	32×16	3.10
256×4	4.34	32×8	37.10	16×64	3.13
64×8	4.36	16×64	38.38	32×8	3.18
128×8	4.37	16×32	38.55	8×128	3.51
16×16	4.37	16×16	38.56	16×32	3.91
16×8	4.54	16×8	38.80	16×16	4.07
8×8	4.66	8×8	49.28	16×8	4.27
8×64	5.07	8×128	57.27	8×64	5.26
8×32	5.22	8×64	62.20	8×8	5.36
8×16	6.89	8×32	67.82	8×32	5.91
8×128	7.10	8×16	69.53	8×16	6.31

4 MNDO Calculations

Fullerenes

Table 3: Computation times (in seconds) for MNDO single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). There are 3 full diagonalizations in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
C_{60} : $N_f = 240$						
N_{SCF}	18	16	18	16	16	18
time	0.12	0.07	0.06	0.08	0.07	0.08
speedup	—	1.6	2.0	1.4	1.6	1.4
C_{120} : $N_f = 480$						
N_{SCF}	24	24	24	24	24	24
time	1.02	0.64	0.41	0.36	0.28	0.23
speedup	—	1.6	2.5	2.8	3.6	4.4
C_{180} : $N_f = 720$						
N_{SCF}	20	20	20	20	20	20
time	2.59	1.51	0.85	0.81	0.56	0.46
speedup	—	1.7	3.0	3.2	4.6	5.6
C_{240} : $N_f = 960$						
N_{SCF}	20	20	20	20	20	20
time	5.89	3.35	1.80	1.62	1.03	0.86
speedup	—	1.8	3.3	3.6	5.7	6.8
C_{540} : $N_f = 2160$						
N_{SCF}	30	30	30	30	30	30
time	86.58	47.39	22.21	12.11	9.94	8.74
speedup	—	1.8	3.9	7.2	8.7	9.9
C_{960} : $N_f = 3840$						
N_{SCF}	34	34	34	36	36	34
time	534.21	292.19	136.10	59.15	47.59	41.84
speedup	—	1.8	3.9	9.0	11.2	12.8

Water Clusters

Table 4: Computation times (in seconds) for MNDO single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). There are 4 full diagonalizations in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
$1000 \cdot H_2O : N_f = 6000$						
N_{SCF}	18	18	18	18	18	18
time	1433.17	858.18	491.57	217.84	160.60	154.59
speedup	—	1.7	2.9	6.6	8.9	9.3
$1200 \cdot H_2O : N_f = 7200$						
N_{SCF}	18	18	18	18	18	18
time	2476.15	1478.91	845.54	365.84	270.84	262.51
speedup	—	1.7	2.9	6.8	9.1	9.4
$1400 \cdot H_2O : N_f = 8400$						
N_{SCF}	18	18	18	18	18	18
time	3859.75	2302.46	1295.02	550.02	402.20	390.56
speedup	—	1.7	3.0	7.0	9.6	9.9
$1600 \cdot H_2O : N_f = 9600$						
N_{SCF}	18	18	18	18	18	18
time	5782.25	3462.94	1956.33	832.70	613.97	597.08
speedup	—	1.7	3.0	6.9	9.4	9.7
$1800 \cdot H_2O : N_f = 10800$						
N_{SCF}	18	18	18	18	18	18
time	8123.96	4815.66	2658.12	1116.80	797.56	777.87
speedup	—	1.7	3.1	7.3	10.2	10.4

Solvated Crambin

Table 5: Computation times (in seconds) for MNDO single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). The number of the full diagonalizations equals N_{SCF} in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
crambin : $N_f = 1623$						
N_{SCF}	24	24	24	24	24	24
time	87.56	47.92	23.66	38.82	15.62	14.96
speedup	—	1.8	3.7	2.3	5.6	5.9
crambin in $500 \cdot H_2O$: $N_f = 4623$						
N_{SCF}	30	29	30	30	29	30
time	2441.81	1274.88	629.00	553.62	356.59	350.81
speedup	—	1.9	3.9	4.4	6.8	7.0
crambin in $600 \cdot H_2O$: $N_f = 5223$						
N_{SCF}	30	30	30	30	30	30
time	3426.76	1870.37	900.35	769.59	502.23	495.30
speedup	—	1.8	3.8	4.5	6.8	6.9
crambin in $700 \cdot H_2O$: $N_f = 5823$						
N_{SCF}	29	29	29	29	29	29
time	4565.55	2506.35	1202.36	1012.93	665.17	657.62
speedup	—	1.8	3.8	4.5	6.9	6.9
crambin in $800 \cdot H_2O$: $N_f = 6423$						
N_{SCF}	36	36	36	36	36	36
time	7673.80	4189.18	2004.12	1639.71	1069.81	1056.54
speedup	—	1.8	3.8	4.7	7.2	7.3
crambin in $900 \cdot H_2O$: $N_f = 7023$						
N_{SCF}	36	36	36	36	36	36
time	9983.93	5490.60	2632.89	2135.51	1379.79	1367.24
speedup	—	1.8	3.8	4.7	7.2	7.3

5 AM1 Calculations

Fullerenes

Table 6: Computation times (in seconds) for AM1 single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). There are 3 full diagonalizations in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
$C_{60} : N_f = 240$						
N_{SCF}	14	14	15	14	14	14
time	0.10	0.07	0.05	0.08	0.07	0.07
speedup	—	1.5	1.9	1.3	1.5	1.4
$C_{120} : N_f = 480$						
N_{SCF}	26	26	26	24	26	24
time	1.08	0.68	0.44	0.36	0.29	0.24
speedup	—	1.6	2.5	3.0	3.7	4.6
$C_{180} : N_f = 720$						
N_{SCF}	22	22	22	18	18	20
time	2.79	1.63	0.91	0.79	0.52	0.46
speedup	—	1.7	3.1	3.5	5.3	6.0
$C_{240} : N_f = 960$						
N_{SCF}	22	22	22	22	22	22
time	6.34	3.61	1.93	1.67	1.08	0.90
speedup	—	1.8	3.3	3.8	5.9	7.1
$C_{540} : N_f = 2160$						
N_{SCF}	31	31	31	31	31	31
time	89.17	48.93	23.07	12.35	10.20	8.94
speedup	—	1.8	3.9	7.2	8.7	10.0
$C_{960} : N_f = 3840$						
N_{SCF}	36	36	36	36	36	36
time	563.63	308.60	141.59	59.58	47.98	43.66
speedup	—	1.8	4.0	9.5	11.7	12.9

Water Clusters

Table 7: Computation times (in seconds) for AM1 single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). There are 4 full diagonalizations in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
$1000 \cdot H_2O : N_f = 6000$						
N_{SCF}	19	19	19	19	19	19
time	1595.66	993.67	612.29	241.02	182.97	176.74
speedup	—	1.6	2.6	6.6	8.7	9.0
$1200 \cdot H_2O : N_f = 7200$						
N_{SCF}	19	19	19	19	19	19
time	2774.96	1730.01	1067.60	408.75	312.25	303.43
speedup	—	1.6	2.6	6.8	8.9	9.1
$1400 \cdot H_2O : N_f = 8400$						
N_{SCF}	20	20	20	20	20	20
time	4531.02	2841.73	1743.17	634.53	486.23	472.98
speedup	—	1.6	2.6	7.1	9.3	9.6
$1600 \cdot H_2O : N_f = 9600$						
N_{SCF}	20	20	20	20	20	20
time	6706.42	4188.18	2545.57	944.75	724.53	705.41
speedup	—	1.6	2.6	7.1	9.3	9.5
$1800 \cdot H_2O : N_f = 10800$						
N_{SCF}	20	20	20	20	20	20
time	9510.36	5901.32	3552.85	1292.06	965.67	943.63
speedup	—	1.6	2.7	7.4	9.8	10.1

Solvated Crambin

Table 8: Computation times (in seconds) for AM1 single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). The number of the full diagonalizations equals N_{SCF} in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
crambin : $N_f = 1623$						
N_{SCF}	33	33	33	33	33	33
time	120.83	66.78	33.22	53.60	21.47	20.52
speedup	—	1.8	3.6	2.3	5.6	5.9
crambin in $500 \cdot H_2O$: $N_f = 4623$						
N_{SCF}	31	31	31	31	31	31
time	2527.47	1363.24	651.13	576.57	377.10	371.21
speedup	—	1.9	3.9	4.4	6.7	6.8
crambin in $600 \cdot H_2O$: $N_f = 5223$						
N_{SCF}	41	41	41	41	41	41
time	4710.21	2564.69	1236.09	1061.84	690.92	680.98
speedup	—	1.8	3.8	4.4	6.8	6.9
crambin in $700 \cdot H_2O$: $N_f = 5823$						
N_{SCF}	54	54	54	54	54	54
time	8557.76	4689.35	2251.66	1915.85	1245.62	1231.29
speedup	—	1.8	3.8	4.5	6.9	7.0
crambin in $800 \cdot H_2O$: $N_f = 6423$						
N_{SCF}	40	40	40	40	40	40
time	8555.17	4661.00	2241.68	1845.91	1196.62	1183.40
speedup	—	1.8	3.8	4.6	7.1	7.2
crambin in $900 \cdot H_2O$: $N_f = 7023$						
N_{SCF}	32	32	32	32	32	32
time	8905.18	4896.37	2347.41	1926.49	1236.16	1221.88
speedup	—	1.8	3.8	4.6	7.2	7.3

6 PM3 Calculations

Fullerenes

Table 9: Computation times (in seconds) for PM3 single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). There are 3 full diagonalizations in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
C_{60} : $N_f = 240$						
N_{SCF}	16	16	16	16	16	16
time	0.11	0.07	0.05	0.08	0.08	0.07
speedup	—	1.5	2.1	1.3	1.4	1.5
C_{120} : $N_f = 480$						
N_{SCF}	22	23	23	22	23	22
time	0.96	0.63	0.41	0.35	0.28	0.22
speedup	—	1.5	2.4	2.7	3.5	4.3
C_{180} : $N_f = 720$						
N_{SCF}	18	18	18	18	18	18
time	2.40	1.40	0.79	0.79	0.53	0.44
speedup	—	1.7	3.0	3.0	4.5	5.5
C_{240} : $N_f = 960$						
N_{SCF}	18	18	18	18	18	18
time	5.43	3.10	1.66	1.59	0.98	0.82
speedup	—	1.8	3.3	3.4	5.5	6.6
C_{540} : $N_f = 2160$						
N_{SCF}	27	27	27	27	27	27
time	79.21	43.53	20.63	11.76	9.49	8.36
speedup	—	1.8	3.8	6.7	8.3	9.5
C_{960} : $N_f = 3840$						
N_{SCF}	32	32	32	32	32	32
time	509.75	279.42	129.62	57.29	45.24	41.36
speedup	—	1.8	3.9	8.9	11.3	12.3

Water Clusters

Table 10: Computation times (in seconds) for PM3 single-point energy calculations in CPU-only (*n*C) and hybrid CPU-GPU (H*n*CG and H*n*CG*) environments (*n* is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). There are 4 full diagonalizations in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
$1000 \cdot \mathrm{H_2O} : N_f = 6000$						
N_{SCF}	23	23	23	23	23	23
time	2045.83	1345.98	902.66	291.88	236.51	229.36
speedup	—	1.5	2.3	7.0	8.6	8.9
$1200 \cdot \mathrm{H_2O} : N_f = 7200$						
N_{SCF}	24	24	24	24	24	24
time	3701.09	2448.78	1656.62	510.69	418.72	407.61
speedup	—	1.5	2.2	7.2	8.8	9.1
$1400 \cdot \mathrm{H_2O} : N_f = 8400$						
N_{SCF}	22	22	22	22	22	22
time	5445.46	3613.87	2438.36	755.31	612.85	598.85
speedup	—	1.5	2.2	7.2	8.9	9.1
$1600 \cdot \mathrm{H_2O} : N_f = 9600$						
N_{SCF}	24	24	24	24	24	24
time	8531.11	5607.69	3721.23	1149.17	938.75	916.80
speedup	—	1.5	2.3	7.4	9.1	9.3
$1800 \cdot \mathrm{H_2O} : N_f = 10800$						
N_{SCF}	24	24	24	24	24	24
time	12169.84	8016.51	5300.55	1597.48	1285.98	1260.80
speedup	—	1.5	2.3	7.6	9.5	9.7

Solvated Crambin

Table 11: Computation times (in seconds) for PM3 single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} denotes the number of SCF iterations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort). The number of the full diagonalizations equals N_{SCF} in each calculation.

	1C	2C	6C	H1CG	H6CG	H6CG*
crambin : $N_f = 1623$						
N_{SCF}	31	31	31	31	31	31
time	114.12	62.69	30.98	50.20	20.23	19.40
speedup	—	1.8	3.7	2.3	5.6	5.9
crambin in $500 \cdot H_2O$: $N_f = 4623$						
N_{SCF}	35	35	35	35	35	35
time	2830.34	1526.66	734.56	640.36	424.15	418.38
speedup	—	1.9	3.9	4.4	6.7	6.8
crambin in $600 \cdot H_2O$: $N_f = 5223$						
N_{SCF}	34	34	34	34	34	34
time	3874.09	2116.22	1020.04	867.27	571.87	563.24
speedup	—	1.8	3.8	4.5	6.8	6.9
crambin in $700 \cdot H_2O$: $N_f = 5823$						
N_{SCF}	35	35	35	35	35	35
time	5507.13	3017.66	1453.20	1219.83	805.46	795.09
speedup	—	1.8	3.8	4.5	6.8	6.9
crambin in $800 \cdot H_2O$: $N_f = 6423$						
N_{SCF}	32	32	32	32	32	32
time	6808.15	3712.72	1788.26	1459.84	959.15	947.13
speedup	—	1.8	3.8	4.7	7.1	7.2
crambin in $900 \cdot H_2O$: $N_f = 7023$						
N_{SCF}	33	33	33	33	33	33
time	9127.94	5034.41	2397.42	1955.34	1273.37	1258.21
speedup	—	1.8	3.8	4.7	7.2	7.3

7 OM1 Calculations

Table 12: Computation times (in seconds) for OM1 single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} and N_{diag} denote the numbers of SCF iterations and full diagonalizations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort).

	1C	2C	6C	H1CG	H6CG	H6CG*
$C_{60} : N_f = 240, N_{SCF} = 13, N_{diag} = 13$						
time	0.26	0.17	0.12	0.23	0.17	0.16
speedup	—	1.5	2.1	1.1	1.5	1.6
$C_{120} : N_f = 480, N_{SCF} = 18, N_{diag} = 18$						
time	2.31	1.43	0.89	1.63[a]	0.98[a]	0.92[a]
speedup	—	1.6	2.6	1.4	2.4	2.5
$C_{180} : N_f = 720, N_{SCF} = 13, N_{diag} = 13$						
time	4.76	2.77	1.58	2.50	1.35	1.23
speedup	—	1.7	3.0	1.9	3.5	3.9
$C_{240} : N_f = 960, N_{SCF} = 15, N_{diag} = 15$						
time	12.18	6.87	3.74	5.86	2.86	2.66
speedup	—	1.8	3.3	2.1	4.3	4.6
$C_{540} : N_f = 2160, N_{SCF} = 24, N_{diag} = 24$						
time	202.67	108.20	53.21	63.62	42.92	41.38
speedup	—	1.9	3.8	3.2	4.7	4.9
$C_{960} : N_f = 3840, N_{SCF} = 21, N_{diag} = 21$						
time	968.54	523.53	256.93	257.05	168.66	164.31
speedup	—	1.9	3.8	3.8	5.7	5.9
$1000 \cdot H_2O : N_f = 6000, N_{SCF} = 18, N_{diag} = 4$						
time	1316.51	745.44	379.78	208.55	151.75	133.68
speedup	—	1.8	3.5	6.3	8.7	9.8
$1200 \cdot H_2O : N_f = 7200, N_{SCF} = 19, N_{diag} = 4$						
time	2376.83	1345.03	684.17	353.66	258.91	232.30
speedup	—	1.8	3.5	6.7	9.2	10.2

[a] N_{SCF} and N_{diag} are 20.

8 OM2 Calculations

Table 13: Computation times (in seconds) for OM2 single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} and N_{diag} denote the numbers of SCF iterations and full diagonalizations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort).

	1C	2C	6C	H1CG	H6CG	H6CG*
C_{60} : $N_f = 240$, $N_{SCF} = 14$, $N_{diag} = 14$						
time	0.28	0.19	0.14	0.25	0.19	0.19
speedup	—	1.5	2.1	1.1	1.5	1.5
C_{120} : $N_f = 480$, $N_{SCF} = 18$, $N_{diag} = 18$						
time	2.22	1.36	0.83	1.37	0.82	0.76
speedup	—	1.6	2.7	1.6	2.7	2.9
C_{180} : $N_f = 720$, $N_{SCF} = 15$, $N_{diag} = 15$						
time	5.54	3.25	1.86	2.95	1.57	1.45
speedup	—	1.7	3.0	1.9	3.5	3.8
C_{240} : $N_f = 960$, $N_{SCF} = 17$, $N_{diag} = 17$						
time	13.78	7.85	4.26	6.79	3.29	3.09
speedup	—	1.8	3.2	2.0	4.2	4.5
C_{540} : $N_f = 2160$, $N_{SCF} = 24$, $N_{diag} = 24$						
time	203.45	108.79	53.42	64.81	43.52	41.80
speedup	—	1.9	3.8	3.1	4.7	4.9
C_{960} : $N_f = 3840$, $N_{SCF} = 25$, $N_{diag} = 25$						
time	1156.48	624.59	308.59	308.19	201.99	195.95
speedup	—	1.9	3.7	3.8	5.7	5.9
$1000 \cdot H_2O$: $N_f = 6000$, $N_{SCF} = 20$, $N_{diag} = 4$						
time	1740.69	1115.49	720.15	341.31	284.65	190.67
speedup	—	1.6	2.4	5.1	6.1	9.1
$1200 \cdot H_2O$: $N_f = 7200$, $N_{SCF} = 20$, $N_{diag} = 4$						
time	3061.91	1977.35	1289.87	599.30	506.24	333.82
speedup	—	1.5	2.4	5.1	6.0	9.2

9 OM3 Calculations

Table 14: Computation times (in seconds) for OM3 single-point energy calculations in CPU-only (nC) and hybrid CPU-GPU (HnCG and HnCG*) environments (n is the number of the CPU cores). N_{SCF} and N_{diag} denote the numbers of SCF iterations and full diagonalizations. An asterisk indicates OpenMP parallelization using 6 CPU cores (our best effort).

	1C	2C	6C	H1CG	H6CG	H6CG*
$C_{60} : N_f = 240, N_{SCF} = 14, N_{diag} = 14$						
time	0.28	0.18	0.13	0.25	0.18	0.18
speedup	—	1.5	2.1	1.1	1.5	1.6
$C_{120} : N_f = 480, N_{SCF} = 18, N_{diag} = 18$						
time	2.18	1.34	0.81	1.34	0.80	0.75
speedup	—	1.6	2.7	1.6	2.7	2.9
$C_{180} : N_f = 720, N_{SCF} = 17, N_{diag} = 17$						
time	6.24	3.65	2.07	3.29	1.74	1.57
speedup	—	1.7	3.0	1.9	3.6	4.0
$C_{240} : N_f = 960, N_{SCF} = 15, N_{diag} = 15$						
time	12.23	6.94	3.80	6.04	3.01	3.16
speedup	—	1.8	3.2	2.0	4.1	3.9
$C_{540} : N_f = 2160, N_{SCF} = 26, N_{diag} = 26$						
time	220.40	119.01	58.44	70.26	47.51	45.65
speedup	—	1.9	3.8	3.1	4.6	4.8
$C_{960} : N_f = 3840, N_{SCF} = 21, N_{diag} = 21$						
time	973.13	527.42	259.81	261.92	171.36	165.75
speedup	—	1.8	3.7	3.7	5.7	5.9
$1000 \cdot H_2O : N_f = 6000, N_{SCF} = 22, N_{diag} = 4$						
time	1818.91	1148.49	720.07	342.92	287.18	190.59
speedup	—	1.6	2.5	5.3	6.3	9.5
$1200 \cdot H_2O : N_f = 7200, N_{SCF} = 22, N_{diag} = 4$						
time	3148.18	1982.86	1244.15	593.63	501.18	324.96
speedup	—	1.6	2.5	5.3	6.3	9.7

Appendix B

Semiempirical Quantum Chemistry

Xin Wu, Axel Koslowski, and Walter Thiel

Electronic Structure Calculations on Graphics Processing Units, Ross C. Walker and Andreas W. Goetz (Eds.), **2014**, John Wiley & Sons Ltd.

I further extended the GPU module of the MNDO package for supporting multiple GPU devices, performed all the calculations, analyzed the results, and drafted the manuscript.

1

Semiempirical Quantum Chemistry

Xin Wu, Axel Koslowski, and Walter Thiel

Max-Planck-Institut für Kohlenforschung, Mülheim an der Ruhr, Germany

1.1 Introduction

Semiempirical quantum-chemical methods are cost-effective tools for chemists to study the structure, stability, and spectroscopy of molecules as well as chemical reactions [1]. They are based on the Hartree-Fock method commonly used in *ab initio* molecular orbital theory [2]. The different semiempirical models simplify the Hartree-Fock procedure by introducing distinct approximations to the Hamiltonian, neglecting many integrals to speed up computations by several orders of magnitude [3]. The remaining integrals are modeled using empirical functions with adjustable parameters that are calibrated against a large number of accurate experimental or high-level theoretical reference data to make semiempirical methods as reliable and general as possible. These features make semiempirical models well-suited to many research areas in chemistry and enabled a large number of semiempirical applications already in the 1970s and 1980s. Since the 1990s, density functional theory (DFT) has become the major workhorse in computational chemistry [4]. However, considering that semiempirical methods are approximately 1000 times faster than standard DFT approaches [5], they are still valuable computational tools nowadays, e.g., for screening large numbers of drug candidates [6], for calculations on proteins [7], for long-time ground-state molecular dynamics simulations [8], and for nonadiabatic excited-state dynamics of large chromophores [9].

The development of computational chemistry is intimately tied to the evolution of computer technology. Originally, computational chemistry programs had been exclusively written for sequential execution on a single central processing unit (CPU) since the 1950s [10]. With the widespread advent of parallel computing in the 1990s, many quantum-chemical codes were parallelized to take advantage of the new architectures, including semiempirical programs [11]. The most recent wave of hardware-driven code development was triggered by the rise of graphics processing units (GPUs). A GPU is a specially designed integrated circuit with powerful, but fixed-function pipelines for faster image rendering and video games. Until 2006, implementing algorithms for general numeric calculations on a

This is a Book Title Name of the Author/Editor

GPU was tediously difficult because the problem had to be cast into graphics operations by resorting to a specific (graphics) API (Application Programming Interface). Programming purely computational tasks on a GPU was considerably simplified by the introduction of the CUDA (Compute Unified Device Architecture) and OpenCL (Open Computing Language) frameworks. In this chapter, we will focus exclusively on the CUDA framework that allows developers to employ the C programming language, with CUDA-specific extensions, to use a CUDA-capable GPU as co-processor of the CPU for computations [12]. As of 2012, the raw hardware peak performance and memory bandwidth of a many-core GPU had significantly outpaced a multi-core CPU. For example, the maximum floating-point performance and theoretical memory bandwidth of an Intel Xeon E5-4650 CPU (eight cores with a base clock of 2.7 GHz and a maximum boost clock of 3.3 GHz with the Intel Turbo Boost Technology, four-channel DDR-1600) are 0.17–0.21 TFlop/s (floating-point operations per second) and 51.2 GB/s, respectively. By contrast, the flagship Tesla K20x by NVIDIA (2688 CUDA cores @ 732 MHz) has a peak of 1.31 TFlop/s for double-precision arithmetic and a memory bandwidth of 250 GB/s with ECC (error-correcting code) off. Hence many groups decided to develop GPU-accelerated programs [13, 14] to take advantage of this promising device for quantum Monte Carlo computations [15, 16], the evaluation of two-electron integrals [17–22], DFT calculations [23–30], high-level correlated *ab initio* methods [31–38], and semiempirical quantum chemistry [39, 40].

In this chapter, we begin with a brief review of semiempirical quantum chemistry, referring readers interested in the detailed formalism and the numerical results to available books [41–43] and reviews [5, 11, 44–50]. We then examine the computational bottlenecks by performing systematic calculations on a set of eight proteins with up to 3558 atoms and 8727 basis functions. Thereafter, we outline how the hotspots identified in this manner are ported to a GPU (making use of multiple devices where possible), and how the remaining code is parallelized CPU-only using symmetric multiprocessing (SMP) capabilities via OpenMP. Next, we analyze the overall performance of our code on the hybrid CPU-GPU platform and compare it with the CPU-only case. Finally, as an illustrative application, we use our CPU-GPU hybrid program to optimize the geometries of three small proteins, each consisting predominantly of one type of secondary structure, namely α-helix, β-strand, and random coil, employing six different semiempirical methods.

1.2 Overview of Semiempirical Methods

Nonrelativistic quantum chemistry aims at finding sufficiently accurate approximate solutions to the Schrödinger equation. In the early days of quantum chemistry, the zero-differential-overlap (ZDO) approximation [51, 52] was introduced to deal with "the nightmare of the integrals" [10], i. e. the difficulty of evaluating the large number of three- and four-center integrals in *ab initio* methods. As a consequence, the integral problem could be tackled at different levels of approximation. Currently, the most accurate semiempirical methods are based on the NDDO (Neglect of Diatomic Differential Overlap) model [3], which retains all one- and two-center two-electron repulsion integrals in the Fock matrix. The first successful and widely adopted NDDO-based parameterization was the MNDO (Modified Neglect of Diatomic Overlap) method [53–55]. The MNDO model also serves as the basis for later parametrizations that have been widely applied, including AM1 (Austin Model 1) [56], PMx (Parametric Methods, $x = 3$, 5, 6, and 7) [57–60] as well

Table 1.1 Proteins in the test set for the OM3 calculations. N_a and N_f denote the number of atoms and basis functions, respectively.

notation	P_{020}	P_{063}	P_{086}	P_{100}	P_{125}	P_{156}	P_{166}	P_{221}
PDB ID	1BTQ	1K50	2HXX	3K6F	1ACF	2A4V	4A02	3AQO
N_a	307	1 097	1 495	1 842	2 004	2 969	3 415	3 558
N_f	754	2 699	3 655	4 446	4 920	7 157	8 173	8 727

as PDDG/MNDO and PDDG/PM3 (MNDO and PM3 augmented with Pairwise Distance Directed Gaussian functions) [61].

Conceptual deficiencies in the established MNDO-type methods include the lacking representation of Pauli exchange repulsion in the Fock matrix. One possible remedy is to introduce orthogonalization corrections into the Fock matrix to account for Pauli exchange repulsion. This can be done through truncated and parametrized series expansions in terms of overlap, which provide corrections to the one-electron core Hamiltonian. These corrections are applied to the one-center matrix elements in OM1 (Orthogonalization Method 1) [62] and to all one- and two-center matrix elements in OM2 [63] and OM3 [64]. Benchmark calculations demonstrate that the OMx methods, especially OM2 and OM3, are superior to AM1 and PM3 for both ground-state and excited-state molecular properties [65–67]. The costs of OMx calculation are roughly the same as those for MNDO-type calculations [39], especially when using suitable cutoffs to neglect the exponentially decreasing three-center orthogonalization corrections to matrix elements involving distant atoms.

1.3 Computational Bottlenecks

In the present work, the OM3 method is taken as an example to illustrate the general strategy of optimizing a semiempirical quantum-chemical program on a hybrid CPU-GPU platform. We have selected a set of eight proteins that are denoted as P_x (x being the number of residues) and listed in Table 1.1, for the purpose of profiling OM3 calculations in a systematic manner [68–75]. Timings for the OMx methods are also representative for MNDO-type methods, because the most time-consuming parts of the calculations are the same in both cases. Consequently, similar wall clock times are obtained: for example, one SCF (self-consistent-field) iteration in MNDO, AM1, PM3, OM1, OM2, and OM3 calculations of $1000 \cdot \mathrm{H_2O}$ takes 80, 84, 89, 73, 87 and 83 seconds, respectively, on a single Intel Xeon X5670 CPU core [39]. Hence, it is sufficient to consider only OM3 in the following.

The OM3 calculations on our test proteins were performed on a server with two Intel Xeon X5690 CPUs (6 cores @ 3.46 GHz per chip), 48 GiB host memory (24 GiB of triple-channel DDR-1333 per chip) with a total theoretical bandwidth[1] of 64 GB/s, and two NVIDIA Tesla M2090 GPUs (512 CUDA cores @ 1.3 GHz per device) with 5.25 GiB ECC memory and a bandwidth of 155 GB/s per device. Intel Turbo Boost Technology (which may automatically increase the CPU frequency above the base clock in accord with the workload in order to exhaust the allowed thermal envelope of the CPU) was intentionally turned off to ensure consistent timings. Three criteria were adopted for SCF convergence in our single-point energy calculations, namely i) a variation of the electronic energy in

[1] If one CPU needs to access memory connected to the other CPU, the theoretical bandwidth is lower.

successive SCF iterations of at most 1.0×10^{-6} eV, ii) a maximum change of the density matrix elements of 1.0×10^{-6}, and iii) a maximum entry in the error matrix of 1.0×10^{-6} in the DIIS (Direct Inversion of Iterative Subspace) extrapolation [76]. To speed up the calculations, the full diagonalization was automatically replaced in the SCF procedure by fast pseudodiagonalization [77] whenever possible.

The code development was conducted on a CVS version of the MNDO99 package [78]. Intel Composer XE 13.1 and NVIDIA CUDA Toolkit 5.0 were used for compiling the FORTRAN subroutines on the CPU and the C-like functions on the GPU, respectively. The final executable was dynamically linked against Intel Math Kernel Library (MKL) 11.0, CUBLAS from the NVIDIA Toolkit, and MAGMA version 1.3.0 [79]. The latter includes a subset of LAPACK routines ported to the GPU; it has been modified locally to conform to the ILP64 (64-bit integers, long integers, and pointers) data model, which is needed to access arrays with 2^{32} or more elements.[2] Before the inclusion of dynamic memory allocation in the FORTRAN standard, the early versions of the MNDO program emulated dynamic memory by passing sections of a fixed-size array in the unnamed COMMON block as arguments to subroutines. The current version of the MNDO code uses essentially the same mechanism, but with a dynamically allocated array instead of the fixed-size array. For larger proteins, the indices of this array may exceed the 32-bit integer range – this is why 64-bit integers are needed.

The computing setup for the OM3 benchmark calculations is denoted as $\mathrm{C}_{[x\mathrm{C}\text{-}y\mathrm{G}]}$, where the subscripts x and y are the number of CPU cores and GPU devices in use, respectively. The wall clock time of an OM3 calculation on $\mathrm{C}_{[1\mathrm{C}]}$ is the reference for calculations with the other compute configurations and the basis for assessing the corresponding speedups. Timings for $\mathrm{C}_{[1\mathrm{G}]}$ and $\mathrm{C}_{[2\mathrm{G}]}$ refer to subroutines executed exlusively on 1 GPU or 2 GPUs, respectively, including the associated and generally negligible CPU-GPU communication. All floating-point operations were done in double precision, both on the CPUs and GPUs, and therefore the numerical results produced on all hardware setups are essentially the same. Deviations in the computed heat of formation (total energy) were occasionally encountered, but remained below 1.0×10^{-5} kcal/mol. Such tiny discrepancies can be attributed to the different order, in which the floating-point operations are performed on the CPU and GPU architectures. Since many operations are performed in parallel, the execution order may not even be fixed, i. e. there might be small deviations between different runs of the same calculation on the same computing setup. The execution order matters because fixed-precision arithmetics is not associative.

The general form of a two-electron repulsion integral (ERI) in *ab initio* and DFT methods is

$$(\mu\nu|\lambda\sigma) \;=\; \int_1\int_2 \frac{\mu(1)\,\nu(1)\,\lambda(2)\,\sigma(2)}{r_{12}}\,dV_1\,dV_2,$$

where the Greek letters represent basis functions or atomic orbitals (AOs). The complexity of the two-electron integral evaluation formally scales as $O(N_f^4)$ for N_f basis functions, but the actual scaling may be more favorable due to the application of screening techniques [80]. The currently applied semiempirical methods make use of the NDDO approximation [3] for ERI evaluation:

$$(\mu_\mathrm{A}\nu_\mathrm{B}|\lambda_\mathrm{C}\sigma_\mathrm{D}) \;=\; \delta_\mathrm{AB}\,\delta_\mathrm{CD}\,(\mu_\mathrm{A}\nu_\mathrm{B}|\lambda_\mathrm{C}\sigma_\mathrm{D}),$$

[2] Starting with version 1.4, MAGMA supports both 32-bit and 64-bit integers out of the box.

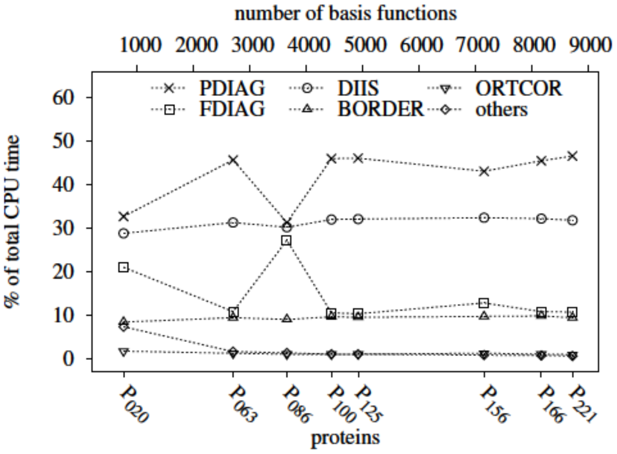


Figure 1.1 Profiles of the OM3 calculations for the test proteins for the $C_{[1C]}$ computing setup.

where atomic centers are denoted by capital letters and δ_{AB} (or δ_{CD}) will vanish unless A and B (or C and D) are the same atom. This rather drastic approximation reduces the formal scaling of the ERI computation in semiempirical methods to $O(N_f^2)$ and makes it possible to simulate complex systems with thousands of atoms. The solution of the secular equations

$$\sum_{\nu}(F_{\mu\nu} - \delta_{\mu\nu}\varepsilon_i)C_{\nu i} \;=\; 0 \tag{1.1}$$

scales as $O(N_f^3)$ and thus becomes the primary computational task in semiempirical methods. ε_i is the energy of the ith molecular orbital (MO). Because the Fock matrix elements $F_{\mu\nu}$ depend on the elements $C_{\nu i}$ of the eigenvectors, Equation (1.1) has to be solved by an iterative SCF procedure that requires $O(N_f^3)$ dense linear algebraic operations.

Figure 1.1 depicts the profiles of OM3 calculations for the $C_{[1C]}$ setup. The pseudodiagonalization procedure (PDIAG) is roughly twice as fast as a full diagonalization (FDIAG), and it is thus preferable to replace FDIAG by PDIAG as often as possible. Applying the default criteria of the MNDO code for the choice between FDIAG and PDIAG, it is normally sufficient to call FDIAG in four of the SCF iterations (i. e. the first three and the last one) during single-point energy evaluation and to call PDIAG in the other SCF iterations (typically 25).[3] Hence most OM3 calculations are dominated by PDIAG with 42.1% of the wall clock time on average. FDIAG and PDIAG complement each other, they collectively contribute $\sim 55\%$ of the total CPU time and are thus the first two bottlenecks.

DIIS is the third hotspot that consumes $\sim 30\%$ of the computation time (see Figure 1.1). Although the DIIS extrapolation may be omitted for small systems (with less than 100 atoms), it is in our experience imperative to apply DIIS to reliably converge the SCF procedure for larger molecules such as proteins. We will thus also investigate the option of leveraging multiple GPUs for the DIIS treatment (see next section).

The last two bottlenecks are the calculation of the density matrix (also called bond order matrix, subroutine BORDER) and the orthogonalization corrections (subroutine ORTCOR

[3] An exception is P_{086} with 11 calls to FDIAG.

in the case of OM3). We spent considerable effort on both routines to achieve optimum performance with the MNDO99 program [78], especially for ORTCOR, where we obtained a huge speedup by formulating all operations as standard matrix-matrix multiplications. After code optimization, BORDER and ORTCOR take 9.4% and 1.1% of the wall clock time on average, respectively, on the $C_{[1C]}$ setup.

Other computational tasks in an OM3 calculation include integral evaluation, formation of the Fock matrix, and initial density matrix generation, which all scale as $O(N_f^2)$. Cumulatively, they require 7% of the CPU time in a serial calculation for a small protein such as P_{020} with 307 atoms and 754 orbitals, but this portion quickly diminishes with increasing system size, to $\sim 0.5\%$ for the largest proteins in our test set, which are the main targets of our code development. Therefore, these other tasks are not considered to be real bottlenecks, and the corresponding subroutines are thus only subjected to an OpenMP parallelization to take advantage of multiple CPUs.

In summary, we have identified five subroutines (FDIAG, PDIAG, DIIS, BORDER and ORTCOR) as computational bottlenecks by systematic analysis of OM3 calculations on a set of proteins. We describe the optimization of these hotspots on a hybrid CPU-GPU platform in the following.

1.4 Profile-Guided Optimization for the Hybrid Platform

1.4.1 Full Diagonalization, Density Matrix, and DIIS

The GPU-accelerated full diagonalization, density matrix construction, and DIIS extrapolation are jointly described here because they heavily rely on the standard routines in the BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra Package) libraries.

Equation (1.1) is an eigenvalue problem that can be solved by diagonalizing the Fock matrix $\mathbf{F}$, which yields the ith MO energy ε_i and the coefficient vector $\mathbf{c}_i$:

$$\mathbf{F}\mathbf{c}_i = \varepsilon_i \mathbf{c}_i$$

This task can be carried out by the LAPACK function DSYEVD that computes all eigenvalues and eigenvectors of a real symmetric matrix using the divide-and-conquer algorithm. DSYEVD of the Intel MKL library makes use of all processor cores on a CPU-only platform, whereas the DSYEVD implementation in MAGMA is a hybrid that utilizes both multi-core CPUs and GPU(s)[4] for the diagonalization [81]. In Figure 1.2 the speedups of FDIAG are plotted as obtained in the OM3 calculations on the proteins in our test set. The scalability on CPU-only setups is evidently rather poor: for instance, the best speedups are observed in the calculations on P_{063}, which are 4.3 on $C_{[6C]}$ and 5.4 on $C_{[12C]}$. Hence, the symmetric parallel processors are highly under-utilized in the FDIAG subroutine, and the efficiency[5] is merely 0.72 and 0.45, respectively. This becomes even worse for larger systems: for example, the speedup of FDIAG for P_{221} on $C_{[6C]}$ is 3.3 and barely increases to 3.8 on $C_{[12C]}$, with corresponding efficiencies of 0.55 and 0.32, respectively. On the contrary, the speedup of the hybrid FDIAG is constantly rising until P_{166} (up to more than 8000 basis

[4] The hybrid DSYEVD function in MAGMA version 1.3 does not support multiple GPUs. This feature is available starting with MAGMA version 1.4.

[5] Processor efficiency is defined as the speedup devided by the number of parallel processing units.

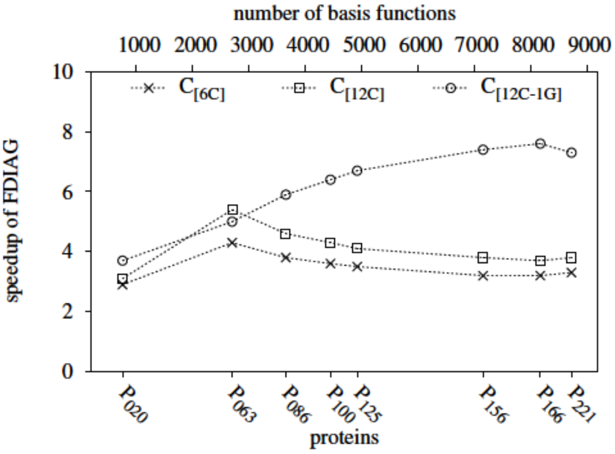


Figure 1.2 Speedups of the FDIAG subroutine in the OM3 calculations on the multi-CPU $C_{[6C]}$, $C_{[12C]}$, and hybrid CPU-GPU $C_{[12C\text{-}1G]}$ computing setups over the serial configuration.

functions). Moreover, it is always superior to its CPU-only counterpart with the exception of P_{063} on the $C_{[12C]}$ setup. For the larger calculations, the hybrid FDIAG subroutine tends to be more than 7 times faster than the serial version and at least two times faster than the parallel CPU-only version.

The primary computational task ($> 99\%$ of the CPU time) in BORDER is a matrix-matrix multiplication, $\mathbf{P} = 2\mathbf{C}_o\mathbf{C}_o^T$, where $\mathbf{P}$ is the density matrix and $\mathbf{C}_o$ is the coefficient matrix of the occupied MOs. A general DGEMM routine could be used to perform this task. Because $\mathbf{P}$ is symmetric, and only the lower triangle is stored as a linear array in the MNDO99 package, we employ a more specific function, namely DSYRK, which only calculates the lower part of a symmetric matrix and thus avoids unnecessary floating-point operations. The CPU-only DSYRK routine has no difficulty to fully load all processors, and the performance scales almost ideally with respect to the number of CPU cores (see Figure 1.3). For example, the speedups for P_{166} are 5.8 on $C_{[6C]}$ and 9.9 on $C_{[12C]}$. At present, no multi-GPU enabled version of DSYRK is available in either CUBLAS or MAGMA. On the other hand, DSYRK on a single GPU may be more than 20 times faster than a single-threaded CPU routine. Thus, we will stick to DSYRK in our development, hoping that multi-GPU support will be added by the vendors in the future.

The DIIS procedure is composed of several different kinds of algebraic operations, in which the calculation of the error matrix ($\mathbf{\Delta} = \mathbf{FP} - \mathbf{PF}$) usually consumes more than 98% of the CPU time [39]. Because the product of $\mathbf{F}$ and $\mathbf{P}$ is a general matrix, the standard DGEMM function is chosen for the DIIS subroutine. The number of floating-point operations and memory accesses in DGEMM scale as $O(N^3)$ and $O(N^2)$ (N being the matrix dimension), respectively. This implies that the number of compute operations per memory access is proportional to N in DGEMM. Thus DGEMM is a compute-bound routine that should be well suited to parallelization. The observed speedups on the CPU-only setups are ~ 5.5 on $C_{[6C]}$ and ~ 10.0 on $C_{[12C]}$. Moreover, a call to DIIS accelerated by a single GPU ($C_{[1G]}$) can be up to 20 times faster than for the baseline setup $C_{[1C]}$. However, the speedup

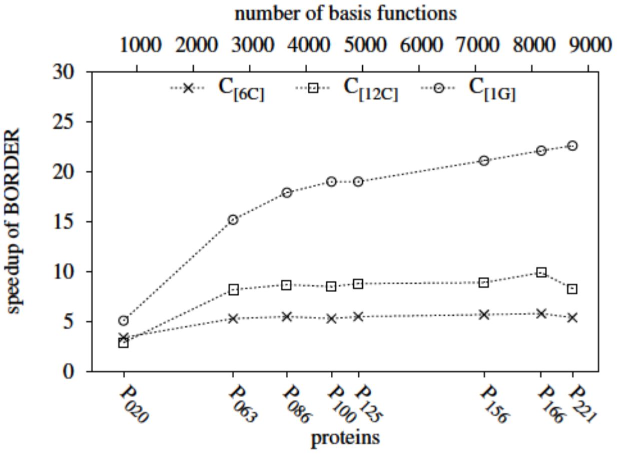


Figure 1.3 Speedups of the BORDER subroutine in the OM3 calculations on the multi-CPU $C_{[6C]}$, $C_{[12C]}$, and GPU-only $C_{[1G]}$ computing setups over the serial configuration.

for $C_{[1G]}$ turns out not to be monotonous with increasing system size: it is highest for P_{125} with $\sim$ 20 and then drops again for the next-larger protein P_{156} to $\sim$ 18.

In order to make the best use of our dual-GPU equipped hardware, we designed a block-matrix scheme for the matrix-matrix multiplication aimed at multiple GPU devices based on the standard DGEMM routine [82]. There are of course more sophisticated multi-GPU DGEMM implementations reported in the literature [83, 84], but the performance of our homemade multi-GPU DGEMM is virtually doubled on two GPUs ($C_{[2G]}$ compared to $C_{[1G]}$) with a peak around 0.7 TFlop/s.

The overall speedup for the DIIS procedure with the multi-GPU DGEMM routine on the $C_{[2G]}$ setup is plotted in Figure 1.4. We find a monotonous increase in performance up to a factor of 30 compared with the $C_{[1C]}$ setup. The use of two GPU devices ($C_{[2G]}$) results in a 1.6-fold speedup over the setup with one single GPU ($C_{[1G]}$).

1.4.2 Pseudodiagonalization

As mentioned in the previous section, pseudodiagonalization will be approximately two times faster than the conventional diagonalization in a given SCF iteration. Thus PDIAG is used instead of FDIAG whenever possible. However, an efficient implementation of PDIAG on multiple GPUs can be challenging. Here, we first analyze the computations involved in pseudodiagonalization, and then report the individual and overall speedups that have been achieved.

The details of pseudodiagonalization have been described in the original paper [77]. From a computational point of view, it is basically comprised of two tasks. First, the Fock matrix is transformed from the AO basis to the MO basis by a triple matrix multiplication (FMO):

$$\mathbf{F}_{\mathrm{MO}} = \mathbf{C}_{\mathrm{o}}^{\mathrm{T}}\mathbf{F}\mathbf{C}_{\mathrm{v}},$$

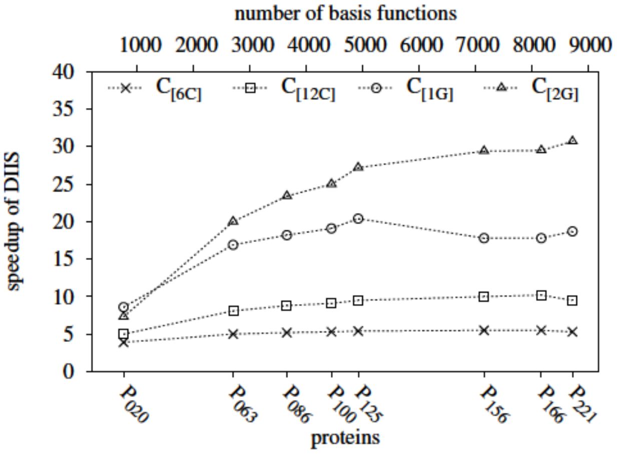


Figure 1.4 Speedups of the DIIS subroutine in the OM3 calculations on the multi-CPU $C_{[6C]}$, $C_{[12C]}$, and GPU-only $C_{[1G]}$ and $C_{[2G]}$ computing setups over the serial configuration.

Table 1.2 Percentages (%) of computation time in the PDIAG subroutine consumed by FMO, JACOBI, and other tasks in OM3 calculations on a single CPU core.

notation	P_{020}	P_{063}	P_{086}	P_{100}	P_{125}	P_{156}	P_{166}	P_{221}
FMO	47.2	40.8	45.1	41.8	42.5	44.7	42.2	42.3
JACOBI	51.6	58.8	54.5	57.9	57.2	55.1	57.6	57.5
others	1.3	0.5	0.4	0.3	0.2	0.2	0.2	0.2

where $\mathbf{C}_o$ and $\mathbf{C}_v$ denote the matrices of the occupied and virtual MO vectors, respectively. Then noniterative Jacobi-like 2 × 2 rotations (JACOBI) between pairs of occupied ($\mathbf{c}_o$) and virtual ($\mathbf{c}_v$) vectors are executed:

$$\mathbf{c}'_o = a\mathbf{c}_o - b\mathbf{c}_v \qquad \text{and} \qquad \mathbf{c}'_v = b\mathbf{c}_o + a\mathbf{c}_v \tag{1.2}$$

where a and b are the elements of the rotation matrix, and the new MO vectors, $\mathbf{c}'_o$ and $\mathbf{c}'_v$, are denoted by primes.

The profiles of the serial PDIAG version for the OM3 calculations on the proteins in our test set are given in Table 1.2. On average, FMO and JACOBI consume $\sim 45\%$ and $\sim 55\%$ of the CPU time, respectively. The other operations are negligible ($< 1\%$) and can be safely excluded from optimization.

The FMO step only contains the DGEMM calls for the matrix multiplications. The relevant speedups with different computing configurations are summarized in Table 1.3. Since DGEMM is compute-bound, FMO scales well with respect to the number of parallel processors in the CPU-only setups. One single GPU-accelerated FMO step can be as much as 20 times faster than on one CPU core. The setup with two GPU devices may further increase the speedup to more than 30-fold, being about 1.6 times faster than on $C_{[1G]}$. The best performance for a small protein like P_{020} is achieved with the CPU-only setup of 12

Table 1.3 Speedups of the FMO and JACOBI steps in the PDIAG subroutine on the multi-CPU $C_{[6C]}$, $C_{[12C]}$, and GPU-only $C_{[1G]}$ and $C_{[2G]}$ computing setups over the serial setup.

	FMO				JACOBI			
	$C_{[6C]}$	$C_{[12C]}$	$C_{[1G]}$	$C_{[2G]}$	$C_{[6C]}$	$C_{[12C]}$	$C_{[1G]}$	$C_{[2G]}$
P_{020}	5.2	7.8	5.9	5.2	3.4	5.1	2.6	3.6
P_{063}	5.7	10.2	16.2	18.6	1.6	1.6	4.4	7.9
P_{086}	5.8	10.6	19.6	22.4	1.4	1.4	4.6	8.6
P_{100}	5.8	10.7	20.0	23.1	1.3	1.2	4.5	8.1
P_{125}	5.8	10.8	20.3	25.4	1.2	1.2	5.0	9.4
P_{156}	5.8	11.3	21.0	30.4	1.2	1.2	4.4	8.6
P_{166}	5.8	11.3	20.6	31.9	1.2	1.2	4.4	8.6
P_{221}	5.5	10.6	20.8	32.9	1.4	1.4	5.1	9.9

cores, however. This is because a GPU is designed for massively parallel tasks that a small system will not fully exploit, and some inevitable overhead such as CPU-GPU data transfer may hurt the overall performance of a smaller calculation.

The GPU-oriented optimization of the JACOBI step is demanding. The technical details can be found in our paper [39]. The resulting speedups are shown in Table 1.3. As one 2×2 rotation given in Equations (1.2) involves six memory accesses (four reads and two writes) and six floating-point operations, the performance of JACOBI is fully determined by the memory bandwidth. In the case of P_{020}, the MO coefficient matrix is small enough (4.3 MiB) to completely fit into the CPU cache (12 MiB per chip). Modest speedups of 3.4 and 5.1 are therefore achieved on the $C_{[6C]}$ and $C_{[12C]}$ setups, respectively. On the other hand, numerous cache misses can occur for larger proteins starting from P_{063}. The performance on the CPU-only platform will then be determined entirely by the available memory bandwidth. The obtained speedup rapidly falls down to 1.2, no matter how many CPU cores are in use for parallelization. On the contrary, JACOBI on a single GPU benefits from the enhanced memory bandwidth (155 GB/s versus 64 GB/s for two CPUs) and speedups of around 4.5-fold are consistently achieved in the benchmarks except for the smallest case, P_{020}. Addition of a second GPU doubles the total memory bandwidth, and the equal distribution of horizontal blocks of the coefficient matrix among the available devices enables the rotations to be carried out independently on each device [82]. The overall speedup on the $C_{[2G]}$ setup for P_{221} is 10, which is 1.9 times higher than that on a single GPU ($C_{[1G]}$).

Since the JACOBI step consumes a slightly higher fraction of the CPU time (between 55 and 60% for most proteins in our test set) than FMO in the PDIAG subroutine for the serial configuration, and since JACOBI benefits less from parallelization than PDIAG on all computing setups, the overall speedups of PDIAG shown in Figure 1.5 resemble those of JACOBI (see Table 1.3), but with some additional performance benefits from the FMO step. The highest speedup is 13.7 for P_{221} on $C_{[2G]}$, which is again 1.9 times higher than that on a single GPU ($C_{[1G]}$).

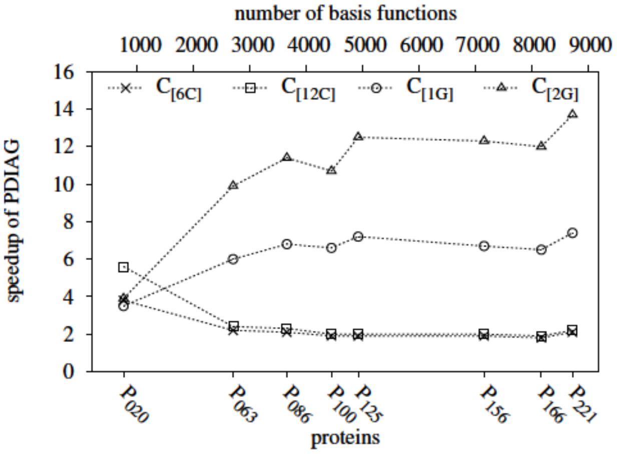


Figure 1.5 Speedups of the PDIAG subroutine in the OM3 calculations on the multi-CPU $C_{[6C]}$, $C_{[12C]}$, and GPU-only $C_{[1G]}$ and $C_{[2G]}$ computing setups over the serial configuration.

1.4.3 Orthogonalization Correction in OM3

The OM3 method [64] accounts for Pauli exchange repulsion by explicitly adding the orthogonalization corrections ($V^{\mathrm{ORT}}_{\mu_{\mathrm{A}}\nu_{\mathrm{B}}}$) to the core Hamiltonian of the Fock matrix:

$$V^{\mathrm{ORT}}_{\mu_{\mathrm{A}}\nu_{\mathrm{B}}} = -\frac{1}{2}G\sum_{\lambda_{\mathrm{C}}}(S_{\mu_{\mathrm{A}}\lambda_{\mathrm{C}}}\beta_{\lambda_{\mathrm{C}}\nu_{\mathrm{B}}} + \beta_{\mu_{\mathrm{A}}\lambda_{\mathrm{C}}}S_{\lambda_{\mathrm{C}}\nu_{\mathrm{B}}}) \qquad (\mathrm{C} \neq \mathrm{A} \text{ and } \mathrm{C} \neq \mathrm{B})$$

where S and β denote elements of the overlap and resonance matrices, respectively, and G is defined in terms of a few parameters that can be adjusted to fit reference data. μ_{A}, ν_{B}, and λ_{C} are AOs at atoms A, B, and C, respectively. If A and B are the same atom, $V^{\mathrm{ORT}}_{\mu_{\mathrm{A}}\nu_{\mathrm{B}}}$ is a correction to a one-center term, otherwise it refers to a two-center element. Inclusion of the latter three-center contributions leads to qualitative improvements over the MNDO-type methods for the calculated molecular properties, such as rotational barriers, relative energies of isomers, hydrogen bonds, and vertical excitation energies [1, 65–67].

Even though the ORTCOR subroutine consumes only $\sim 1\%$ of the wall clock time for the $C_{[1C]}$ setup, we implemented a dedicated algorithm utilizing multiple GPUs in an attempt to harness all available computing power. The ORTCOR performance for various setups is depicted in Figure 1.6. The technical details will be presented elsewhere.

The speedup of the ORTCOR subroutine scales reasonably well on the symmetric multi-CPU setups. For example, 5.5- and 10.1-fold performance boosts are feasible on the $C_{[6C]}$ and $C_{[12C]}$ setups, respectively. ORTCOR is accelerated up to 28-fold for medium-sized proteins like P_{063} on a single GPU ($C_{[1G]}$ setup), but thereafter the speedup decreases again with increasing system size to ~ 20 for the largest proteins in our test set. The speedup on the $C_{[2G]}$ setup can reach 35-fold for a moderately sized protein, and there is no performance deterioration for larger proteins. Moreover, the multi-GPU ORTCOR scales well compared to a single GPU device for sufficiently large proteins. For example, ORTCOR is 1.7 times faster on $C_{[2G]}$ than on $C_{[1G]}$ for P_{221}.

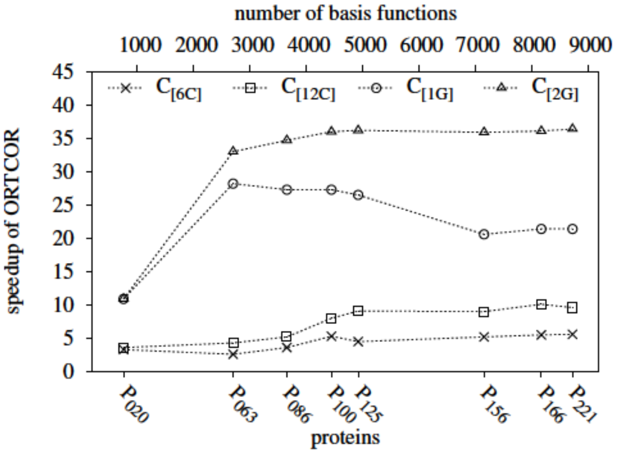


Figure 1.6 Speedups of the ORTCOR subroutine in the OM3 calculations on the multi-CPU $C_{[6C]}$, $C_{[12C]}$, and GPU-only $C_{[1G]}$ and $C_{[2G]}$ computing setups over the serial configuration.

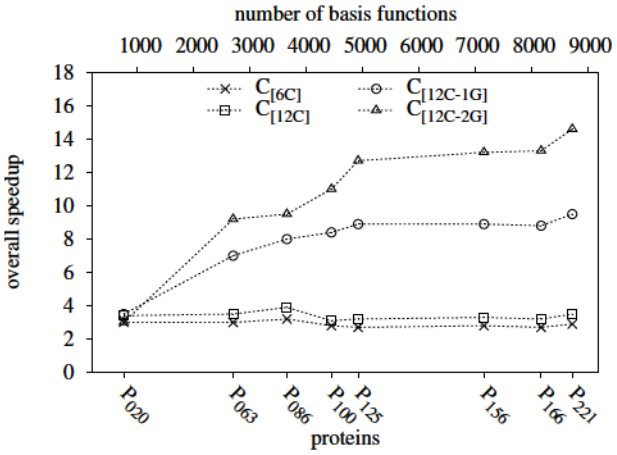


Figure 1.7 Overall speedups of the OM3 calculations of test proteins on the multi-CPU $C_{[6C]}$, $C_{[12C]}$, and hybrid CPU-GPU $C_{[12C\text{-}1G]}$ and $C_{[12C\text{-}2G]}$ computing setups over the serial configuration.

1.5 Performance

Since a user will of course never run an individual subroutine by itself, the overall speedups for the OM3 calculations on proteins are more relevant in practice. They are presented in Figure 1.7.

The performance of the OM3 calculations on the CPU-only platform can hardly be improved by using more processor cores. The speedups quickly reach a saturation point and never exceed 4. The mean values averaged over the proteins in the test set are 2.9 and

3.4 on $C_{[6C]}$ and $C_{[12C]}$, respectively. Moreover, the individual speedups seem to be almost invariant with respect to the size of the protein. Thus neither using more CPU cores nor increasing the system size yields higher speedups on the CPU-only setup. At first glance, this conclusion seems to contradict our previous result that the speedup of an MNDO calculation on fullerene C_{540} could reach 7.7 on Cray Y-MP with eight vector processors [85]. This apparent discrepancy can be resolved by considering the relevant arithmetic operations and the differences in the computer architectures. Concerning the computational bottlenecks mentioned in the preceding section, only three subroutines (BORDER, DIIS, and ORTCOR) of the five hotspots in the OM3 calculations can be well accelerated on current hardware by using additional CPU cores (see Figure 1.3, 1.4 and 1.6), whereas neither FDIAG nor PDIAG, which consume $\sim 65\%$ of the wall clock time, scale favorably with the number of cores (see Figure 1.2 and 1.5). This is because the former three are primarily dominated by compute-bound routines, which demand more arithmetic power than memory bandwidth. On the other hand, both diagonalization subroutines are composed of bandwidth-bound operations that would parallelize well on more CPU cores if and only if the demand for memory bandwidth could be satisfied in the first place. The theoretical floating-point peak performance of the two Xeon X5690 CPUs (a total of 166 GFlop/s) exceeds that of the Cray Y-MP (2.6 GFlop/s) by a factor of 64. The theoretical memory bandwidth of our current Xeon server (64 GB/s), however, is merely two times greater than that of the 25-year-old Cray Y-MP (32 GB/s). Therefore, a tremendously inadequate memory bandwidth prevents the performance boost on a computer system including only parallel superscalar CPUs.

Because of the advantages of GPUs with regard to floating-point peak performance and memory bandwidth, the speedups achieved for the OM3 calculations on GPUs are monotonously growing with the size of the proteins and the number of GPUs (see Figure 1.7). Although the hybrid CPU-GPU platform provides higher speedups than the CPU-only platform for most bottlenecks, there may be exceptions in the case of calculations on small proteins like P_{020}. This may be caused by the CPU-GPU communication overhead, by the unfavorable behavior of certain subroutines for small systems on a hybrid platform compared to a CPU-only setup (especially PDIAG, see Figure 1.5), or by less optimized non-GPU routines becoming more dominant. For example, the CPU-only computation on P_{020} takes only 41% of the wall clock time for the $C_{[12C\text{-}2G]}$ setup (see Figure 1.8). Thus the overall performance of the OM3 calculations for P_{020} is rather similar on all computing setups. On the other hand, the acceleration on the hybrid CPU-GPU and CPU-only platforms is quite different for large calculations. The speedups of the OM3 calculations for P_{221} reach 9.5 and 14.6 on the $C_{[12C\text{-}1G]}$ and $C_{[12C\text{-}2G]}$ setups, respectively. The relative speedup of $C_{[12C\text{-}2G]}$ over $C_{[12C\text{-}1G]}$ is ~ 1.5 for the OM3 calculations of large proteins. Further performance increases are thus very likely when more GPU devices are employed in even larger semiempirical quantum-chemical calculations.

Finally, we inspect the profiles of the OM3 calculations on the hybrid $C_{[12C\text{-}2G]}$ setup (see Figure 1.8). DIIS, BORDER, and ORTCOR are the three subroutines most accelerated on the GPU, thus their combined share of the wall clock time is just about half of that on $C_{[1C]}$. On average, the shares of DIIS, BORDER, and ORTCOR amount to 31.4%, 9.4%, 1.1% and 14.0%, 5.7%, 0.4% on the $C_{[1C]}$ and $C_{[12C\text{-}2G]}$ setups, respectively. The speedups for FDIAG and PDIAG are not as good as those for the former three routines, and hence their combined share on the $C_{[12C\text{-}2G]}$ setup is increased to 64.4% on average. The remaining subroutines (e. g., for integral evaluation and Fock matrix formation) have not yet been ported to a

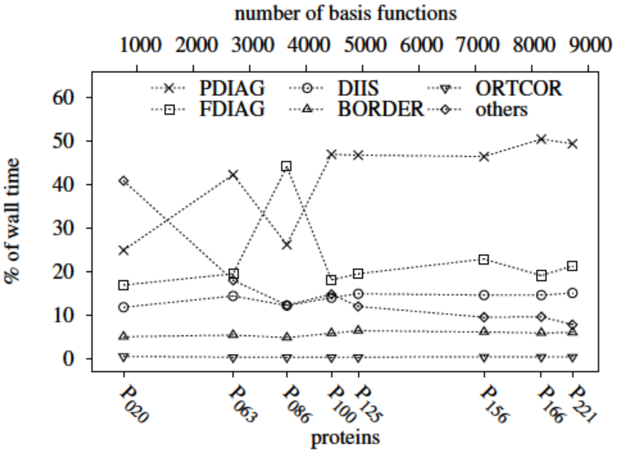
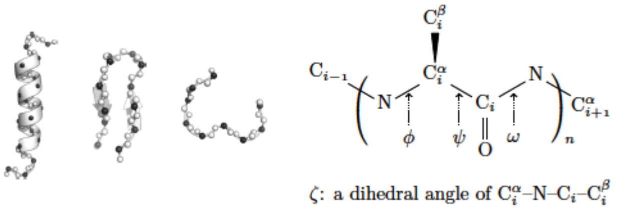


Figure 1.8 Profiles of the OM3 calculations of test proteins on the $C_{[12C\text{-}2G]}$ computing setup.



(a) P_A (b) P_B (c) P_C (d)

Figure 1.9 Experimental structures of (a) P_A (PDB ID: 2AP7, 80% α-helix), (b) P_B (PDB ID: 2EVQ, 50% β-strands) and (c) P_C (PDB ID: 1LVR, 100% random coil). Only the backbone atoms are shown, with the C^α atoms represented by black balls. Four dihedral angles (ϕ, ψ, ω, and ζ) in a residue serve as stereochemical metrics, see the schematic sketch in (d).

GPU, but are executed in parallel using multiple CPU cores (via OpenMP). They become the bottlenecks for small protein calculations with a time share of 40.9% in P_{020}, which gradually decreases with system size, down to 7.8% for a large protein like P_{221}. We may thus anticipate some further improvement of the overall performance with dedicated multi-GPU kernels for the semiempirical integral evaluation and Fock matrix construction.

1.6 Applications

Given the code developments outlined above, it has now become a routine task to carry out semiempirical quantum-chemical calculations for large biomolecules, such as proteins, on a hybrid CPU-GPU computing platform. We have carried out full geometry optimizations of three proteins with α-helix, β-sheet, and random coil structures (see Figure 1.9), which were chosen from a collection of proteins used in previous work [86]. Six different semiempirical methods were applied, namely MNDO, AM1, PM3, and OMx ($x = 1$, 2, and 3). The

Table 1.4 Statistics (%) for $[\phi, \psi]$ in the most favored ($P^0_{\phi,\psi}$) and additionally allowed ($P^1_{\phi,\psi}$) regions of the Ramachandran plot and standard deviations (°) of ω and ζ. Results for the experimental structures of $\mathrm{P_A}$, $\mathrm{P_B}$ and $\mathrm{P_C}$ are compared with those calculated by semiempirical quantum-chemical methods.

	$\mathrm{P_A}$				$\mathrm{P_B}$				$\mathrm{P_C}$			
	$P^0_{\phi,\psi}$	$P^1_{\phi,\psi}$	σ_ω	σ_ζ	$P^0_{\phi,\psi}$	$P^1_{\phi,\psi}$	σ_ω	σ_ζ	$P^0_{\phi,\psi}$	$P^1_{\phi,\psi}$	σ_ω	σ_ζ
Expt.	100.0	0.0	0.8	0.6	87.5	12.5	2.6	0.7	42.9	42.9	2.3	1.8
MNDO	91.7	8.3	9.0	1.3	87.5	12.5	14.9	0.5	28.6	57.1	17.0	2.5
AM1	75.0	16.7	9.7	1.0	100.0	0.0	15.8	1.0	28.6	71.4	7.0	1.9
PM3	75.0	25.0	15.3	1.3	87.5	12.5	19.7	1.0	42.9	57.1	22.8	1.7
OM1	81.8	18.2	12.4	1.0	87.5	12.5	9.4	0.9	28.6	71.4	12.5	2.5
OM2	83.3	16.7	8.6	1.1	87.5	12.5	10.4	1.1	42.9	42.9	10.5	2.1
OM3	83.3	16.7	7.6	1.1	87.5	12.5	15.7	1.2	42.9	42.9	14.3	2.5

optimizations were terminated when the gradient vector norm dropped below a preselected threshold value ($|\mathrm{g}| \leq 1.0\,\mathrm{kcal} \cdot \mathrm{mol}^{-1} \cdot \text{Å}^{-1}$). The quality of the computed structures was assessed in terms of the conformation of the main chain by using the PROCHECK package [87], in comparison with the structures determined in aqueous solution by nuclear magnetic resonance (NMR) experiments. It should be stressed that the results given here are just for demonstration, since more realistic simulations would require more elaborate approches (e.g., including explicit solvent).

The backbones of the proteins are shown in Figure 1.9. Highly regular local structures imposed by hydrogen bonds are found in $\mathrm{P_A}$ and $\mathrm{P_B}$, whereas $\mathrm{P_C}$ possesses an unfolded polypeptide chain. The backbone conformation of a protein is determined by a pair of less rigid dihedral angles $[\phi, \psi]$ at the C^α-atom [88] and a stiff torsion angle ω of the peptide bond. ω is usually restricted to be around 180° for an energetically more favorable $trans$ conformer due to the partial double bond character of the amide bond that prevents facile rotation. In addition, a virtual dihedral angle ζ is defined between C^α_i–N and noncovalently bound $\mathrm{C}_i \cdots \mathrm{C}^\beta_i$ as a measure of chirality at the central C^α_i atom of the amino acid [89].

PROCHECK divides a Ramachandran map into four regions: most favored, additionally allowed, generously allowed, and disallowed. The shares of the first two distributions, $P^0_{\phi,\psi}$ and $P^1_{\phi,\psi}$, for $\mathrm{P_A}$, $\mathrm{P_B}$ and $\mathrm{P_C}$ are listed in Table 1.4. In most cases, $P^0_{\phi,\psi}$ and $P^1_{\phi,\psi}$ add up to the total population. Neither experimental nor theoretically optimized protein structures are spoiled by disallowed $[\phi, \psi]$ combinations. Since more regular secondary structures exist in $\mathrm{P_A}$ and $\mathrm{P_B}$ than in the disordered $\mathrm{P_C}$, significantly higher values for $P^0_{\phi,\psi}$ are obtained for the former two proteins. MNDO, AM1, PM3, and OM1 predict a higher $[\phi, \psi]$ population in the additionally allowed region for $\mathrm{P_C}$, whereas OM2, OM3, and the NMR experiment give equal values for $P^0_{\phi,\psi}$ and $P^1_{\phi,\psi}$. Although the deficiencies of the original MNDO method for the description of hydrogen bonds are known from early studies [90, 91], its actual performance for the proteins in the test set seems rather satisfactory. Both the α-helix (in $\mathrm{P_A}$) and β-strand (in $\mathrm{P_B}$) structures are found, and reasonable $[\phi, \psi]$ distributions are retained in the optimized structures.

All semiempirical methods predict greater deviations from planarity around the peptide bond than deduced from experiment (see the σ_ω values). Such deviations from planarity in the peptide group have already been reported in earlier theoretical studies [7, 92, 93]: the sp^2-hybrid nitrogen in a peptide bond should be planar, but it tends to be pyramidalized in semiempirical calculations. The average value of ζ for L-amino acids is $33.81 \pm 4.17°$ [89]. The σ_ζ values from experiment and from semiempirical calculations are rather small and of similar quality, indicating a good description of the local environment of the sp^3-C_i^α atoms in the main chains.

1.7 Conclusion

In this chapter, we have presented a profile-guided optimization of the semiempirical quantum-chemical MNDO program on a hybrid CPU-GPU platform. OM3 calculations on a set of eight proteins were used to guide the code development and to assess the performance. The computational bottlenecks on one single CPU core were identified as the diagonalization of the Fock matrix (FDIAG), fast pseudodiagonalization (PDIAG), SCF acceleration (DIIS), density matrix formation (BORDER), and computation of the orthogonalization corrections in OM3 (ORTCOR), which cover altogether $\sim 99\%$ of the wall clock time in the test runs. Standard library routines and special finely tuned kernels targeting multiple GPU devices were employed to accelerate these toutines, whereas the relevant remaining subroutines ($\sim 1\%$ of the computation time) were run in parallel using multiple CPU cores (via OpenMP) to achieve optimum performance on the hybrid CPU-GPU platform.

We have identified severe restraints to parallelize the semiempirical calculations on currently available CPU-only computing architectures. No matter how many processor cores are utilized in a calculation, a ceiling of the overall acceleration is reached rapidly due to the limitations imposed by the hardware memory bandwidth. On the other hand, the speedup of the calculations on the hybrid CPU-GPU platform rises continuously with increasing system size and reaches one order of magnitude in large protein calculations. The overall performance can be further improved through the use of multiple GPUs.

As an illustrative application, geometry optimizations of three typical proteins with α-helix, β-sheet, and random coil structures were carried out by means of the MNDO, AM1, PM3, and OMx ($x =$ 1, 2, and 3) methods. These calculations produced qualitatively reasonable conformations of the main chains (with regard to the usual metrics for assessing protein backbone structures) but showed some deviation from experiment by giving slightly nonplanar peptide bonds. We are confident that such quantitative deficiencies can be ameliorated in future semiempirical method development. This will enhance the impact of the current code development work on hybrid CPU-GPU platforms, which has enabled semiempirical quantum-chemical calculations on large systems like proteins with thousands of atoms.

References

[1] W. Thiel. Semiempirical quantum-chemical methods. *WIREs Comput. Mol. Sci.* **2013**, DOI:10.1002/wcms.1161.

[2] W. J. Hehre, L. Radom, P. v. R. Schleyer, J. A. Pople, *Ab initio molecular orbital theory*, John Wiley & Sons, New York, **1986**.

[3] J. A. Pople, D. P. Santry, G. A. Segal. Approximate self-consistent molecular orbital theory. I. Invariant procedures. *J. Chem. Phys.* **1965**, *43*, S129 – S135.

[4] R. G. Parr, W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford University Press, USA, **1989**.

[5] W. Thiel. Perspectives on semiempirical molecular orbital theory. *Adv. Chem. Phys.* **1996**, *93*, 703 – 757.

[6] B. Ehresmann, M. J. de Groot, A. Alex, T. Clark. New molecular descriptors based on local properties at the molecular surface and a boiling-point model derived from them. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 658 – 668.

[7] J. J. P. Stewart. Application of the PM6 method to modeling proteins. *J. Mol. Model.* **2009**, *15*, 765 – 805.

[8] X. Wu, W. Thiel, S. Pezeshki, H. Lin. Specific reaction path hamiltonian for proton transfer in water: Reparameterized semiempirical models. *J. Chem. Theory Comput.* **2013**, *9*, 2672 – 2686.

[9] E. Fabiano, Z. Lan, Y. Lu, W. Thiel, *Nonadiabatic Trajectory Calculations with Ab Initio and Semiempirical Methods* in *Conical Intersections: Theory, Computation and Experiment*, W. Domcke, D. R. Yarkony, H. Köppel (Eds.), World Scientific Publishing Company, **2011**, pp. 463 – 496.

[10] J. A. Pople, *Quantum Chemical Models* in *Nobel Lectures in Chemistry (1996-2000)*, I. Grenthe (Ed.), World Scientific Publishing Co., Singapore, **2003**, pp. 246 – 260.

[11] W. Thiel, D. G. Green, *The MNDO94 Code: Parallelization of a Semiempirical Qauntum-chemical Program* in *Methods and Techniques in Computational Chemistry: METECC-95*, E. Clementi, G. Corongiu (Eds.), STEF, Cagliari, **1995**, pp. 141 – 168.

[12] D. B. Kirk, W. M. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann, **2012**.

[13] J. E. Stone, D. J. Hardy, I. S. Ufimtsev, K. Schulten. GPU-accelerated molecular modeling coming of age. *J. Mol. Graph. Model.* **2010**, *29*, 116 – 125.

[14] R. M. Farber. Topical perspective on massive threading and parallelism. *J. Mol. Graph. Model.* **2011**, *30*, 82 – 89.

[15] A. G. Anderson, W. A. G. III, P. Schröder. Quantum Monte Carlo on graphical processing units. *Comput. Phys. Commun.* **2007**, *177*, 298 – 306.

[16] J. Kim, J. M. Rodgers, M. Athènes, B. Smit. Molecular Monte Carlo simulations using graphics processing units: To waste recycle or not? *J. Chem. Theory Comput.* **2011**, *7*, 3208 – 3222.

[17] K. Yasuda. Two-electron integral evaluation on the graphics processor unit. *J. Comput. Chem.* **2008**, *29*, 334 – 342.

[18] I. S. Ufimtsev, T. J. Martínez. Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. *J. Chem. Theory Comput.* **2008**, *4*, 222 – 231.

[19] A. Asadchev, V. Allada, J. Felder, B. M. Bode, M. S. Gordon, T. L. Windus. Uncontracted Rys quadrature implementation of up to G functions on graphical processing units. *J. Chem. Theory Comput.* **2010**, *6*, 696 – 704.

[20] K. A. Wilkinson, P. Sherwood, M. F. Guest, K. J. Naidoo. Acceleration of the GAMESS-UK electronic structure package on graphical processing units. *J. Comput. Chem.* **2011**, *32*, 2313 – 2318.

[21] Y. Miao, K. M. Merz. Acceleration of electron repulsion integral evaluation on graphics processing units via use of recurrence relations. *J. Chem. Theory Comput.* **2013**, *9*, 965 – 976.

[22] A. V. Titov, I. S. Ufimtsev, N. Luehr, T. J. Martinez. Generating efficient quantum chemistry codes for novel architectures. *J. Chem. Theory Comput.* **2013**, *9*, 213 – 221.

[23] K. Yasuda. Accelerating density functional calculations with graphics processing unit. *J. Chem. Theory Comput.* **2008**, *4*, 1230 – 1236.

[24] I. S. Ufimtsev, T. J. Martínez. Quantum chemistry on graphical processing units. 2. Direct self-consistent-field implementation. *J. Chem. Theory Comput.* **2009**, *5*, 1004 – 1015.

[25] I. S. Ufimtsev, T. J. Martínez. Quantum chemistry on graphical processing units. 3. Analytical energy gradients, geometry optimization, and first principles molecular dynamics. *J. Chem. Theory Comput.* **2009**, *5*, 2619 – 2628.

[26] I. S. Ufimtsev, T. J. Martínez. Graphical processing units for quantum chemistry. *Comput. Sci. Eng.* **2008**, *10*, 26 – 34.

[27] L. Genovese, M. Ospici, T. Deutsch, J.-F. Méhaut, A. Neelov, S. Goedecker. Density functional theory calculation on many-cores hybrid central processing unit-graphic processing unit architectures. *J. Chem. Phys.* **2009**, *131*, 034103 (8 pages).

[28] N. Luehr, I. S. Ufimtsev, T. J. Martínez. Dynamic precision for electron repulsion integral evaluation on graphical processing units (GPUs). *J. Chem. Theory Comput.* **2011**, *7*, 949 – 954.

[29] X. Andrade, L. Genovese, *Harnessing the Power of Graphic Processing Units* in *Fundamentals of Time-Dependent Density Functional Theory*, Vol. 837 of *Lecture Notes in Physics*, M. A. Marques, N. T. Maitra, F. M. Nogueira, E. Gross, A. Rubio (Eds.), **2012**, pp. 401 – 413.

[30] X. Andrade, A. Aspuru-Guzik. Real-space density functional theory on graphical processing units: computational approach and comparison to Gaussian basis set methods. *J. Chem. Theory Comput.* **2013**, *9*, 4360 – 4373.

[31] C. M. Isborn, N. Luehr, I. S. Ufimtsev, T. J. Martínez. Excited-state electronic structure with configuration interaction singles and Tamm-Dancoff time-dependent density functional theory on graphical processing units. *J. Chem. Theory Comput.* **2011**, *7*, 1814 – 1823.

[32] L. Vogt, R. Olivares-Amaya, S. Kermes, Y. Shao, C. Amador-Bedolla, A. Aspuru-Guzik. Accelerating resolution-of-the-identity second-order Møller-Plesset quantum chemistry calculations with graphical processing units. *J. Phys. Chem. A* **2008**, *112*, 2049 – 2057.

[33] R. Olivares-Amaya, M. A. Watson, R. G. Edgar, L. Vogt, Y. Shao, A. Aspuru-Guzik. Accelerating correlated quantum chemistry calculations using graphical processing units and a mixed precision matrix multiplication library. *J. Chem. Theory Comput.* **2010**, *6*, 135 – 144.

[34] M. Watson, R. Olivares-Amaya, R. G. Edgar, A. Aspuru-Guzik. Accelerating correlated quantum chemistry calculations using graphical processing units. *Comput. Sci. Eng.* **2010**, *12*, 40 – 51.

[35] A. E. DePrince, J. R. Hammond. Coupled cluster theory on graphics processing units I. The coupled cluster doubles method. *J. Chem. Theory Comput.* **2011**, *7*, 1287 – 1295.

[36] W. Ma, S. Krishnamoorthy, O. Villa, K. Kowalski. GPU-based implementations of the noniterative regularized-CCSD(T) corrections: Applications to strongly correlated systems. *J. Chem. Theory Comput.* **2011**, *7*, 1316 – 1327.

[37] K. Bhaskaran-Nair, W. Ma, S. Krishnamoorthy, O. Villa, H. J. J. van Dam, E. Aprà, K. Kowalski. Noniterative multireference coupled cluster methods on heterogeneous CPU-GPU systems. *J. Chem. Theory Comput.* **2013**, *9*, 1949 – 1957.

[38] A. Asadchev, M. S. Gordon. Fast and flexible coupled cluster implementation. *J. Chem. Theory Comput.* **2013**, *9*, 3385 – 3392.

[39] X. Wu, A. Koslowski, W. Thiel. Semiempirical quantum chemical calculations accelerated on a hybrid multicore CPU-GPU computing platform. *J. Chem. Theory Comput.* **2012**, *8*, 2272 – 2281.

[40] J. D. C. Maia, G. A. Urquiza Carvalho, C. P. Mangueira, S. R. Santana, L. A. F. Cabral, G. B. Rocha. GPU linear algebra libraries and GPGPU programming for accelerating MOPAC semiempirical quantum chemistry calculations. *J. Chem. Theory Comput.* **2012**, *8*, 3072 – 3081.

[41] M. J. S. Dewar, *The Molecular Orbital Theory of Organic Chemistry*, of *McGraw-Hill series in advanced chemistry*, McGraw-Hill, **1969**.

[42] J. A. Pople, D. L. Beveridge, *Approximate Molecular Orbital Theory*, of *McGraw-Hill series in advanced chemistry*, McGraw-Hill, **1970**.

[43] J. N. Murrell, A. J. Harget, *Semi-empirical Self-consistent-field Molecular Orbital Theory of Molecules*, Wiley-Interscience, **1972**.

[44] W. Thiel. Semiempirical methods: current status and perspectives. *Tetrahedron* **1988**, *44*, 7393 – 7408.

[45] W. Thiel. Computational methods for large molecules. *J. Mol. Str. Theochem* **1997**, *398 – 399*, 1 – 6.

[46] W. Thiel, *Semiempirical Methods* in *Modern Methods and Algorithms of Quantum Chemistry Proceedings, Second Edition*, J. Grotendorst (Ed.), John von Neumann Institute for Computing, Jülich, **2000**, pp. 261 – 283.

[47] W. Thiel, *Semiempirical Quantum-Chemical Methods in Computational Chemistry* in *Theory and Applications of Computational Chemistry: The First Forty Years*, C. E. Dykstra, G. Frenking, K. S. Kim, G. E. Scuseria (Eds.), Elsevier, Amsterdam, **2005**, pp. 559 – 580.

[48] J. J. P. Stewart, *Semiempirical Molecular Orbital Methods* in *Reviews in Computational Chemistry*, K. B. Lipkowitz, D. B. Boyd (Eds.), John Wiley & Sons, Inc., **1990**, pp. 45 – 81.

[49] M. C. Zerner, *Semiempirical Molecular Orbital Methods* in *Reviews in Computational Chemistry*, K. B. Lipkowitz, D. B. Boyd (Eds.), John Wiley & Sons, Inc., **1991**, pp. 313 – 365.

[50] J. J. P. Stewart. MOPAC: A semiempirical molecular orbital program. *J. Comput. Aid. Mol. Des.* **1990**, *4*, 1 – 103.

[51] R. G. Parr. A method for estimating electronic repulsion integrals over LCAO MO's in complex unsaturated molecules. *J. Chem. Phys.* **1952**, *20*, 1499.

[52] J. A. Pople. Electron interaction in unsaturated hydrocarbons. *Trans. Faraday Soc.* **1953**, *49*, 1375 – 1385.

[53] M. J. S. Dewar, W. Thiel. A semiempirical model for the two-center repulsion integrals in the NDDO approximation. *Theor. Chim. Acta* **1977**, *46*, 89 – 104.

[54] M. J. S. Dewar, W. Thiel. Ground states of molecules. 38. The MNDO method. Approximations and parameters. *J. Am. Chem. Soc.* **1977**, *99*, 4899 – 4907.

[55] M. J. S. Dewar, W. Thiel. Ground states of molecules. 39. MNDO results for molecules containing hydrogen, carbon, nitrogen, and oxygen. *J. Am. Chem. Soc.* **1977**, *99*, 4907 – 4917.

[56] M. J. S. Dewar, E. G. Zoebisch, E. F. Healy, J. J. P. Stewart. Development and use of quantum mechanical molecular models. 76. AM1: A new general purpose quantum mechanical molecular model. *J. Am. Chem. Soc.* **1985**, *107*, 3902 – 3909.

[57] J. J. P. Stewart. Optimization of parameters for semiempirical methods I. Method. *J. Comput. Chem.* **1989**, *10*, 209 – 220.

[58] J. J. P. Stewart. Comparison of the accuracy of semiempirical and some DFT methods for predicting heats of formation. *J. Mol. Model.* **2004**, *10*, 6 – 12.

[59] J. J. P. Stewart. Optimization of parameters for semiempirical methods V: Modification of NDDO approximations and application to 70 elements. *J. Mol. Model.* **2007**, *13*, 1173 – 1213.

[60] J. J. P. Stewart. Optimization of parameters for semiempirical methods VI: More modifications to the NDDO approximations and re-optimization of parameters. *J. Mol. Model.* **2013**, *19*, 1 – 32.

[61] M. P. Repasky, J. Chandrasekhar, W. L. Jorgensen. PDDG/PM3 and PDDG/MNDO: Improved semiempirical methods. *J. Comput. Chem.* **2002**, *23*, 1601 – 1622.

[62] M. Kolb, W. Thiel. Beyond the MNDO model: Methodical considerations and numerical results. *J. Comput. Chem.* **1993**, *14*, 775 – 789.

[63] W. Weber, W. Thiel. Orthogonalization corrections for semiempirical methods. *Theor. Chem. Acc.* **2000**, *103*, 495 – 506.

[64] M. Scholten, *Semiempirische Verfahren mit Orthogonalisierungskorrekturen: Die OM3 Methode*, Ph.D. thesis, Universität Düsseldorf, Düsseldorf, **2003**.

[65] N. Otte, M. Scholten, W. Thiel. Looking at self-consistent-charge density functional tight binding from a semiempirical perspective. *J. Phys. Chem. A* **2007**, *111*, 5751 – 5755.

[66] M. Korth, W. Thiel. Benchmarking semiempirical methods for thermochemistry, kinetics, and noncovalent interactions: OMx methods are almost as accurate and robust as DFT-GGA methods for organic molecules. *J. Chem. Theory Comput.* **2011**, *7*, 2929 – 2936.

[67] M. R. Silva-Junior, W. Thiel. Benchmark of electronically excited states for semiempirical methods: MNDO, AM1, PM3, OM1, OM2, OM3, INDO/S, and INDO/S2. *J. Chem. Theory Comput.* **2010**, *6*, 1546 – 1564.

[68] A. R. Gargaro, G. B. Bloomberg, C. E. Dempsey, M. Murray, M. J. A. Tanner. The solution structures of the first and second transmembrane-spanning segments of band 3. *Eur. J. Biochem.* **1994**, *221*, 445 – 454, PDB ID: 1BTQ.

[69] J. W. O'Neill, D. E. Kim, K. Johnsen, D. Baker, K. Y. Zhang. Single-site mutations induce 3D domain swapping in the B1 domain of protein L from Peptostreptococcus magnus. *Structure* **2001**, *9*, 1017 – 1027, PDB ID: 1K50.

[70] M. Rubini, S. Lepthien, R. Golbik, N. Budisa. Aminotryptophan-containing barstar: structure-function tradeoff in protein design and engineering with an expanded genetic code. *Biochim. Biophys. Acta* **2006**, *1764*, 1147 – 1158, PDB ID: 2HXX.

[71] C. Ciatto, F. Bahna, N. Zampieri, H. C. VanSteenhouse, P. S. Katsamba, G. Ahlsen, O. J. Harrison, J. Brasch, X. Jin, S. Posy, J. Vendome, B. Ranscht, T. M. Jessell, B. Honig, L. Shapiro. T-cadherin structures reveal a novel adhesive binding mechanism. *Nat. Struct. Mol. Biol.* **2010**, *17*, 339 – 347, PDB ID: 3K6F.

[72] A. A. Fedorov, K. A. Magnus, M. H. Graupe, E. E. Lattman, T. D. Pollard, S. C. Almo. X-ray structures of isoforms of the actin-binding protein profilin that differ in their affinity for phosphatidylinositol phosphates. *Proc. Natl. Acad. Sci. USA* **1994**, *91*, 8636 – 8640, PDB ID: 1ACF.

[73] J. Choi, S. Choi, J. K. Chon, J. Choi, M.-K. Cha, I.-H. Kim, W. Shin. Crystal structure of the C107S/C112S mutant of yeast nuclear 2-Cys peroxiredoxin. *Proteins* **2005**, *61*, 1146 – 1149, PDB ID: 2A4V.

[74] G. Vaaje-Kolstad, L. A. Bøhle, S. Gåseidnes, B. Dalhus, M. Bjørås, G. Mathiesen, V. G. Eijsink. Characterization of the chitinolytic machinery of Enterococcus faecalis V583 and high-resolution structure of its oxidative CBM33 enzyme. *J. Mol. Biol.* **2012**, *416*, 239 – 254, PDB ID: 4A02.

[75] T. Tsukazaki, H. Mori, Y. Echizen, R. Ishitani, S. Fukai, T. Tanaka, A. Perederina, D. G. Vassylyev, T. Kohno, A. D. Maturana, K. Ito, O. Nureki. Structure and function of a membrane component SecDF that enhances protein export. *Nature* **2011**, *474*, 235 – 238, PDB ID: 3AQO.

[76] P. Pulay. Improved SCF convergence acceleration. *J. Comput. Chem.* **1982**, *3*, 556 – 560.

[77] J. J. P. Stewart, P. Császár, P. Pulay. Fast semiempirical calculations. *J. Comput. Chem.* **1982**, *3*, 227 – 228.

[78] W. Thiel, *MNDO99 CVS Development Version*, Tech. Rep., Mülheim an der Ruhr, Germany, **2012**.

[79] J. Dongarra, T. Dong, M. Gates, A. Haidar, S. Tomov, I. Yamazaki, *MAGMA: A new generation of linear algebra libraries for GPU and multicore architectures*, **2012**.

[80] M. Häser, R. Ahlrichs. Improvements on the direct SCF method. *J. Comput. Chem.* **1989**, *10*, 104 – 111.

[81] A. Haidar, R. Solcà, M. Gates, S. Tomov, T. Schulthess, J. Dongarra, *Leading Edge Hybrid Multi-GPU Algorithms for Generalized Eigenproblems in Electronic Structure Calculations* in *Supercomputing*, Vol. 7905 of *Lecture Notes in Computer Science*, J. Kunkel, T. Ludwig, H. Meuer (Eds.), Springer Berlin Heidelberg, **2013**, pp. 67 – 80.

[82] X. Wu, A. Koslowski, W. Thiel. unpublished results.

[83] D. Rohr, M. Bach, M. Kretz, V. Lindenstruth. Multi-GPU DGEMM and high performance Linpack on highly energy-efficient clusters. *Micro, IEEE* **2011**, *31*, 18 – 27.

[84] F. Spiga, I. Girotto, *phiGEMM: A CPU-GPU library for porting quantum ESPRESSO on hybrid systems* in *Proceeding of 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP2012)*, pp. 368 – 375.

[85] D. Bakowies, W. Thiel. MNDO study of large carbon clusters. *J. Am. Chem. Soc.* **1991**, *113*, 3704 – 3714.

[86] H. J. Kulik, N. Luehr, I. S. Ufimtsev, T. J. Martínez. *Ab initio* quantum chemistry for protein structures. *J. Phys. Chem. B* **2012**, *116*, 12501 – 12509.

[87] R. A. Laskowski, M. W. MacArthur, D. S. Moss, J. M. Thornton. PROCHECK: A program to check the stereochemical quality of protein structures. *J. Appl. Crystallogr.* **1993**, *26*, 283 – 291.

[88] G. N. Ramachandran, C. Ramakrishnan, V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.* **1963**, *7*, 95 – 99.

[89] A. L. Morris, M. W. MacArthur, E. G. Hutchinson, J. M. Thornton. Stereochemical quality of protein structure coordinates. *Proteins.* **1992**, *12*, 345 – 364.

[90] K. Y. Burstein, A. N. Isaev. MNDO calculations on hydrogen bonds. Modified function for core-core repulsion. *Theor. Chim. Acta* **1984**, *64*, 397 – 401.

[91] A. Goldblum. Improvement of the hydrogen bonding correction to MNDO for calculations of biochemical interest. *J. Comput. Chem.* **1987**, *8*, 835 – 849.

[92] K. Möhle, H.-J. Hofmann, W. Thiel. Description of peptide and protein secondary structures employing semiempirical methods. *J. Comput. Chem.* **2001**, *22*, 509 – 520.

[93] G. de M. Seabra, R. C. Walker, A. E. Roitberg. Are current semiempirical methods better than force fields? A study from the thermodynamics perspective. *J. Phys. Chem. A* **2009**, *113*, 11938 – 11948.

Appendix C

Hybrid Quantum and Classical Simulations of the Dihydrofolate Reductase Catalyzed Hydride Transfer Reaction on an Accurate Semi-Empirical Potential Energy Surface

Dvir Doron, Dan Thomas Major, Amnon Kohen, Walter Thiel, and Xin Wu

Journal of Chemical Theory and Computation, **2011**, 7, 3420 – 3437.

I parameterized the AM1-SRP(D) Hamiltonian for the hydride transfer reaction, produced some of the figures, and drafted the appendix of the paper.

JCTC Journal of Chemical Theory and Computation ARTICLE
pubs.acs.org/JCTC

Hybrid Quantum and Classical Simulations of the Dihydrofolate Reductase Catalyzed Hydride Transfer Reaction on an Accurate Semi-Empirical Potential Energy Surface

Dvir Doron,[†] Dan Thomas Major,[*,†] Amnon Kohen,[‡] Walter Thiel,[§] and Xin Wu[§]

[†]Department of Chemistry, The Lise Meitner-Minerva Center of Computational Quantum Chemistry, Bar-Ilan University, Ramat-Gan 52900, Israel

[‡]Department of Chemistry, University of Iowa, Iowa City, Iowa 52242, United States

[§]Max-Planck-Institut für Kohlenforschung, Kaiser-Wilhelm-Platz 1, D-45470 Mülheim an der Ruhr, Germany

S Supporting Information

ABSTRACT: Dihydrofolate reductase (DHFR) catalyzes the reduction of 7,8-dihydrofolate by nicotinamide adenine dinucleotide phosphate hydride (NADPH) to form 5,6,7,8-tetrahydrofolate and oxidized nicotinamide. DHFR is a small, flexible, monomeric protein with no metals or SS bonds and serves as one of the enzymes commonly used to examine basic aspects in enzymology. In the current work, we present extensive benchmark calculations for several model reactions in the gas phase that are relevant to the DHFR catalyzed hydride transfer. To this end, we employ G4MP2 and CBS-QB3 ab initio calculations as well as numerous density functional theory methods. Using these results, we develop two specific reaction parameter (SRP) Hamiltonians based on the semiempirical AM1 method. The first generation SRP Hamiltonian does not account for dispersion, while the second generation SRP accounts for dispersion implicitly via the AM1 core-repulsion functions. These SRP semiempirical Hamiltonians are subsequently used in hybrid quantum mechanics/molecular mechanics simulations of the DHFR catalyzed reaction. Finally, kinetic isotope effects are computed using a mass-perturbation-based path-integral approach.

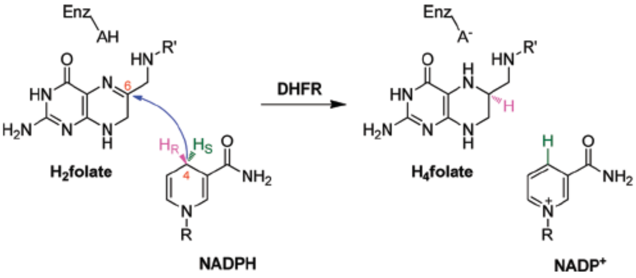
INTRODUCTION

Dihydrofolate reductase (DHFR; EC 1.5.1.3) catalyzes the reduction of 7,8-dihydrofolate (H_2folate) by nicotinamide adenine dinucleotide phosphate hydride (NADPH) to form *S*-5,6,7,8-tetrahydrofolate (H_4folate) and NADP$^+$. Its principal function is to maintain intracellular pools of H_4folate, which in turn serves as a cofactor in one-carbon metabolic processes and is essential for the biosynthesis of purines, thymine nucleotides, and several amino acids. DHFR has long been recognized as an important target for various therapeutic purposes, in particular the development of anticancer and antibacterial drugs, such as methotrexate and trimethoprim, respectively.[1,2] The clinical importance of DHFR, along with its relatively modest size (159 aa in *E. coli* DHFR), has led many researchers to study, both experimentally and theoretically, the catalytic mechanism and kinetics of the NADPH-dependent hydride transfer reaction.[1]

The key chemical step in the catalytic cycle of DHFR involves a stereospecific transfer of the *pro-R* hydrogen at the C4 position of the nicotinamide ring in NADPH to the *si*-face of the C6 atom of the pterin ring in H_2folate, with concomitant protonation at the adjacent N5 position (Scheme 1).[3,4]

Early kinetic studies of *E. coli* DHFR by Fierke et al. established a catalytic pathway cycling between five intermediates, including the E·NADPH, the Michaelis complex E·NADPH·H_2folate, the ternary product complex E·NADP$^+$·H_4folate, the binary product complex E·H_4folate, and the product release complex E·NADPH·H_4folate.[5] At neutral pH, the hydride-transfer rate in the wild-type enzyme is 220 s^{-1} and the rate-determining step is the release of the product H_4folate (12 s^{-1}), whereas at high pH, the hydride transfer step becomes more rate limiting, suggesting that the protonated substrate (henceforth H_3folate$^+$) is the reactive species for the hydride transfer reaction.[5] Thus, the traditional view is that the protonation step precedes hydride transfer, thereby generating a resonance-stabilized iminium intermediate that serves as a more efficient acceptor toward the negatively charged hydride ion.[4,6−8] Indeed, vibrational spectroscopic studies by Callender et al. have demonstrated that the N5 pK_a of H_2folate is raised from 2.6 in solution to 6.5 in the ternary product complex with *E. coli* DHFR.[9−12] The elevation of the N5 pK_a by four units upon complex formation is likely an enzymatic strategy for a substantial rate enhancement

Scheme 1. Hydride Transfer Reaction Catalyzed by DHFR[a]



[a] R: adenine dinucleotide 2′-phosphate. R′: *p*-aminobenzoyl-glutamate.

Received: July 10, 2011
Published: August 26, 2011

over the uncatalyzed reaction in solution. It is likely that the hydrophobic nature of the active site pocket maintains a relatively low permittivity, thereby lowering H_3folate$^+$ acidity. Nevertheless, the protonation source is a rather controversial issue: originally, it was believed to be Asp27, being the only ionizable residue within the binding site of *E. coli* DHFR.[10,13,14] However, the studies of Callender and co-workers suggested that the N5 atom is responsible for the pH dependency of the reaction and that the Asp27 residue exists in a deprotonated form at physiological pH and does not donate a proton to the substrate during enzymatic catalysis. It has been suggested instead that the negative charge of the carboxylate of Asp27 could stabilize the protonated substrate even though this group is on the other side of the bound substrate relative to N5.[10,11,15] This notion was supported by computational studies of Brooks and Rod, arguing that the protonation of the substrate's N5 position comes directly from the bulk solvent.[16]

From a structural point of view, DHFR is a small monomeric protein (~18 kD, 159 aa for the *E. coli* DHFR).[17−20] The substrate and coenzyme bind in a deep hydrophobic cleft at the juncture of the adenosine binding subdomain and the major ("loop") subdomain. In its "closed" conformation, the Met20 loop (residues 9−24) lies directly over the active site, shielding the reactants from the solvent, and is primarily responsible for determining the active site architecture. X-ray studies in numerous ligand-bound states show that the Met20 loop assumes four characteristic conformations with respect to the active site.[20] In particular, the movement of Met20 loop and the cycling of the βF−βG and βG−βH loops between the *closed* and *occluded* conformations are coordinated with the stages of the catalytic cycle.[20]

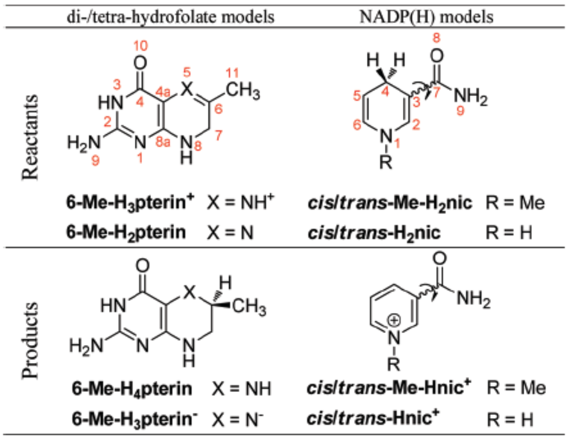
Early computational studies on the hydride transfer reaction catalyzed by DHFR were typically performed for small model systems *in vacuo*.[21−24] In these studies, the potential energy surface (PES) was traditionally explored at the semiempirical or *ab initio* levels of theory. Subsequently, hybrid quantum-mechanical/molecular-mechanical (QM/MM) studies of DHFR have enabled the examination of environmental effects.[4,25−27] Indeed, Moliner and co-workers in their pioneering QM/MM study on DHFR[4] illustrated the importance of incorporating the enzyme's degrees of freedom in the study of transition state structures, compared to those calculated for the gas phase models.[4] However, the calculated free energies of reaction and activation barriers by this approach were unrealistic compared to the experimental data, overestimating the free energy barrier for the enzymatic reaction, due to the inaccurate representation of the atoms in the QM region by a semiempirical method.

Nevertheless, numerous research groups took advantage of the computational efficiency of the hybrid semiempirical QM/MM approach combined with molecular dynamics (MD) simulations, to explore the hydride transfer reaction in DHFR[7,28−30] and other related dehydrogenases.[31−34] Some of these works[7,28,30] attempted to compensate for the inherent errors in the semiempirical models by introducing correction terms. In particular, Gready and co-workers derived correction terms for selected configurations estimated from analogue cluster calculations at the DFT and post-Hartree−Fock theoretical levels, and these were applied to the activation free energy.[28] However, the authors did not manage to obtain quantitative agreement with the experimentally observed free energy barrier. On the other hand, Garcia-Viloca et al.,[7] and more recently Brooks and Thorpe,[30] enhanced the semiempirical Hamiltonian with a simple valence bond (SVB) correction term,[35] which was parametrized to fit two quantities: the activation free energy of 13.4 kcal/mol, which is predicted by transition-state theory expression taking the experimental pH-independent hydride transfer rate constant (950 s^{-1} at 25 °C)[5] into account, and the reaction free energy of -4.4 kcal/mol, calculated from the reported equilibrium constant (1700).[5] The disadvantage of such an approach is that for systems where the original QM method performs poorly, the SVB term is necessarily large. This may introduce artifacts into the vibrational frequency related to the SVB (reaction) coordinate, as well as an artificial increase in energy at the end points of the reaction coordinate.[7] Such QM/MM potentials have been employed in combination with ensemble averaged variational transition state theory with multidimensional tunneling (EA-VTST/MT) calculations to obtain kinetic isotope effects.[36]

A different QM/MM approach to studying hydride transfer reactions is the empirical valence bond (EVB) potential for the QM region, which was developed by Warshel and Weiss[37,38] and parametrized for DHFR by Hammes-Schiffer et al.[39,40] and Warshel and Liu.[41] In this method, the hydride transfer reaction is represented by means of two empirical valence bond states, namely, reactants and products. The matrix elements between these diabatic states are represented as MM terms which are parametrized to reproduce experimental free energies of reaction and activation (dictated by either the pH-independent[39] or -dependent[40] intrinsic rate constants). An advantage of the EVB approach is that the reaction coordinate can naturally be expressed in terms of a collective entity (i.e., the energy gap between the reactant and product diabatic states). Additionally, the EVB method is computationally very efficient. On the other hand, due to its simplistic form and depending on the parametrization strategy, the fine details of the potential energy surface may not be described correctly. Furthermore, the experimental assessment of intrinsic rates in a complex kinetic cascade is limited, and the presumed experimental rate constant is commonly a complex kinetic term with many microscopic rate constants that cannot be distinguished experimentally. Therefore, using experimentally determined rate constants to parametrize the EVB terms might be problematic. The EVB potential has been employed to incorporate nuclear quantum effects by representing the transferring hydrogen nucleus as a 3-D vibrational wave function,[39] as well as in Feynman path-integral (PI) simulations.[42] For a tabulated summary of some prominent computational studies on the DHFR hydride transfer reaction published in the past decade, the reader is referred to the Supporting Information.

To the best of our knowledge, no simulation of DHFR has gone beyond the EVB level or the semiempirical (AM1/PM3) Hamiltonian levels, using the standard parameters. In this paper, we first present gas-phase model calculations for the hydride transfer reaction between H_2folate and NADPH, using high-level *ab initio* and density functional theory methods. Although these models lack the contribution of the enzymatic environment, the calculations shed light on some key thermodynamic aspects related to the intrinsic thermochemistry of the reaction. Second, we present an accurate potential energy surface for the hydride transfer reaction in the enzyme *E. coli* DHFR, taking advantage of the comprehensive gas phase calculations presented herein. This potential energy surface is described by a hybrid quantum mechanics/molecular mechanics (QM/MM) potential, where the QM subsystem is treated by a semiempirical model that has been specifically parametrized to reproduce *ab initio* and DFT data. Consequently, the quality of such a specific reaction parameter (SRP) model[43−45] is comparable with calculations at the *ab initio* and DFT levels, but at a considerably lower cost, hence

Chart 1. Main Models Subjected to Gas-Phase Calculations in This Study



allowing us to perform long MD simulations with the full solvated enzyme. The calibration of the AM1 Hamiltonian was performed only for the reaction in the gas phase (i.e., the intrinsic performance of the QM model), while the interactions with the environment are captured via the QM/MM interaction terms. Subsequently, the performance of the QM/MM potential was validated for the enzymatic reaction. This is in contrast to models employed in earlier studies, which parametrize empirical QM models such as EVB and SVB to the experimental free energies of reaction and activation in solution or in the enzyme. Two principle SRP models were explored, differing in the way they treat dispersion interactions. Nuclear quantum effects (NQE) are described using a coupled free-energy mass-perturbation and umbrella sampling simulation technique employing Feynman centroid path integral calculations (PI-FEP/UM).[46] Thus, both the electronic structure of the reacting system and the nuclear dynamics are treated quantum mechanically. This method has been demonstrated in a series of studies of chemical reactions in solution and in enzymes.[44−48]

■ METHODOLOGY

Gas Phase QM Calculations. *Model Reactions.* The molecules depicted in Chart 1 represent chemical analogues of the reacting ligands and their corresponding products involved in the DHFR enzymatic reaction. The geometries were fully optimized in the gas phase using the Gaussian 09 program[49] and seven different density functionals, including one generalized gradient approximation (GGA) functional [PBEPBE (PBE)[50]], two hybrid GGA functionals [PBE1PBE (PBE0)[51] and B3LYP[52−54]], and four hybrid meta-GGA functionals (B98,[55] BB1K,[56] MPWB1K,[56] and M06[57]). For any functional, the 6-31+G(d,p) basis set was found to give adequately converged geometries and reaction energies (results are not shown for other basis sets). The same geometries were recalculated with the semiempirical potential Austin Model 1 (AM1),[58] using the standard parameter scheme within Gaussian 09. We also carried out high-level calculations with the complete basis set method CBS-QB3,[59,60] as well as the Gaussian theory method G4MP2.[61,62] The latter method provided most of the target values for reparametrization of the AM1 Hamiltonian (*vide infra*).

The H_2folate substrate and its corresponding H_4folate product are represented by derivatives of 6-methyl-7,8-dihydropterin and 6-methyl-5,6,7,8-tetrahydropterin. These molecular models are the substrate and product of the DHFR catalyzed reaction, and it is assumed that the mechanism of reduction of the 7,8-dihydropterin is similar to that of 7,8-dihydrofolate.[63] To investigate the effect of protonation of the pyrazine's N5 atom on thermodynamics, two model reactions were inspected: the "protonated" model, which involves the protonated form of the dihydropterin reactant (6-Me-H_3pterin$^+$), ending up in a neutral reduced tetrahydropterin product (6-Me-H_4pterin), and the "unprotonated" model, in which the unprotonated dihydropterin (6-Me-H_2pterin) is reduced to a negatively charged species (6-Me-H_3pterin$^-$).

Two conformers of the nicotinamide derivative representing the NADP(H) cofactor were considered, distinguished by the orientation of the carboxamide with respect to the (dihydro)-pyridine ring: the *cisoid* conformer, in which the carbonyl and the C2=C3 bond are quasi-synperiplanar, and the *transoid* conformer, where these two are quasi-antiperiplanar. As the two types of conformers were taken into account for both the reduced and oxidized nicotinamide species (Me-H_2nic and Me-Hnic$^+$, respectively), four thermodynamic pathways were computed for each model reaction separately, as depicted in Scheme 2.

In both model reactions, the changes in electronic energy (ΔE_{el}), enthalpy (ΔH), and Gibbs free energy (ΔG) at 298 K were computed for each pathway with the aforementioned methods, according to the following general equations:

$$\Delta X_r^{prot} = X[\text{6-Me-H}_4\text{pterin}] + X[\textit{cis/trans}\text{-Me-Hnic}^+] - X[\text{6-Me-H}_3\text{pterin}^+] - X[\textit{cis/trans}\text{-Me-H}_2\text{nic}] \quad (1a)$$

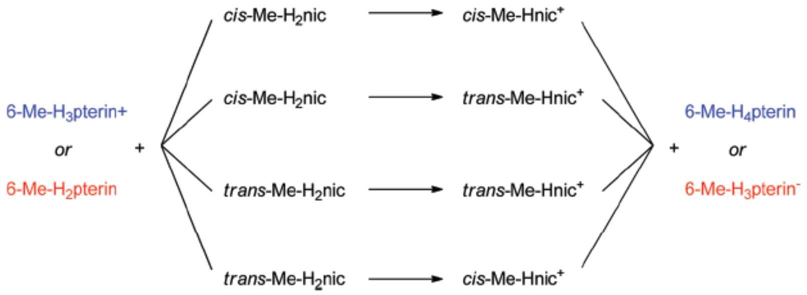
$$\Delta X_r^{unprot} = X[\text{6-Me-H}_3\text{pterin}^-] + X[\textit{cis/trans}\text{-Me-Hnic}^+] - X[\text{6-Me-H}_2\text{pterin}] - X[\textit{cis/trans}\text{-Me-H}_2\text{nic}] \quad (1b)$$

where X is a general notation for E_{el}, H, or G, and the superscripts X^{prot} and X^{unprot} refer to the protonated and unprotonated model reactions, respectively.

Modeling the Reactant, Transition, and Product State Complexes. The optimized model structure for the transition state complex was found using the synchronous transit-guided quasi-Newton (STQN) method[64] implemented in the Gaussian 09 program,[49] with all seven density functionals and the standard semiempirical AM1 potential (the G4MP2 method turned out to be too costly for transition state optimization of the bimolecular complex). In the case of the B3LYP functional, an empirical Grimme-type dispersion correction was also added (denoted B3LYP-D).[65,66] The *transoid* conformer of the Me-H_2nic subunit was chosen, because it was found to be the most prevalent conformer identified in X-ray crystal structures of most enzyme active sites,[67,68] in particular *E. coli* DHFR.[20] The saddle point was identified by a single imaginary vibrational frequency corresponding to the normal mode of transferring the hydride between the donor (C4 in the nicotinamide subunit) and acceptor (C6 in the pterin subunit) carbons.

The reactant and product complexes, [6-Me-H_3pterin$^+$·*trans*-Me-H_2nic] and [6-Me-H_4folate·*trans*-Me-Hnic$^+$], respectively, were obtained by intrinsic reaction coordinate (IRC) calculations in the direction of reactants and products. The steepest descent path in mass-scaled coordinates was followed using 100 steps of 0.1 Bohr in each direction of the reaction path

Scheme 2. Possible Thermodynamic Pathways for the Protonated (Blue) and Unprotonated (Red) Model Reactions of the Hydride Transfer in the Gas Phase



down to the reactant and product complex wells, where the root-mean-square gradient norms at the end points were no higher than 0.2 kcal·mol^{-1}·Å^{-1} and 0.3 kcal·mol^{-1}·Å^{-1} in the forward and reverse directions, respectively. Wherever methods accounting for dispersion (B3LYP-D and M06) were used, the IRC calculations were performed further until the default convergence criteria were reached in each IRC direction. The IRC end points were further geometry optimized as was done for the individual molecules.

Development of Semiempirical Specific Reaction Parameters. In the SRP approach,[43] the semiempirical parameters are optimized for a given system, i.e., here to treat the hydride transfer reaction in DHFR. More specifically, AM1-SRP parameters were developed to reproduce electronic and thermodynamic properties obtained from high-level QM calculations on representative molecular models in the gas phase (*vide supra*). For the majority of the modeled species, the reference method for the AM1 parametrization was the composite Gaussian method G4MP2,[61,62] whereas target quantities associated with models of the reactant, transition, and product state bimolecular complexes were calculated using the B3LYP, B3LYP-D, and M06 hybrid functionals[52−54] with the 6-31+G(d,p) basis set. The observables used as target values were enthalpies of formation, reaction energies, geometries, dipole moments, Mulliken charges, and vibrational frequencies.

Briefly, the current SRP strategy aims at two major goals: (1) an electronic PES which is of high accuracy, comparable to a high-level *ab initio* or DFT PES, and (2) absolute atomization energies which give rise to accurate heats of formation. The former goal allows the SRP model to be employed in MD simulations where all classical thermal effects are included directly via the propagation of Newton's equations of motion, while all quantum thermal effects are included via path-integral simulations or similar methods. The latter objective serves to limit deviations from the original AM1 parameter set (which has been optimized to reproduce heats of formation) to a minimum, as many basic molecular properties are treated well by AM1, although energy values are often not sufficiently accurate for quantitative comparison with experimental results. The SRP parameters are obtained by a nonlinear optimization, starting with the original AM1 parameters as the initial input.[69] The following general sequential optimization scheme is adopted: (1) First, optimize the one-center energies (U_{ss} and U_{pp}) and the resonance integrals (β_s and β_p), followed by the α parameters in the core-repulsion function (CRF). (2) Adjust the orbital exponents (ζ_s and ζ_p) together with the previously optimized parameters. (3) Adjust the one-center two-electron repulsion integrals G_{ss}, G_{sp}, G_{pp}, G_{p2} (or $G_{pp'}$), and H_{sp} together with the previously optimized parameters. (4) Include the Gaussian CRF parameters L, M, and K together with the previously optimized parameters.

The training set for the optimizations consisted of the 12 individual molecules in Chart 1, plus the neutral nicotinamides, *trans*-nic and *cis*-nic. Additionally, the reactant, product, and transition states and selected structures along the IRC reaction paths were also included. All individual molecules were fully geometry optimized during the SRP parametrization, while the complexes were either partially or fully optimized. Single point calculations were carried out for the structures along the IRC.

In order to assess the importance of dispersion interactions, we attempted to implicitly include dispersion effects into the AM1-SRP model via the AM1 CRFs. In this approach, an improved SRP Hamiltonian was developed which implicitly accounts for dispersion. The motivation for including dispersion was provided by initial attempts to optimize the geometry of the reactant and product complexes using standard AM1. This led to unrealistic gas-phase structures, presumably due to the lack of dispersion interactions in standard AM1; for example, the two fragments in the product complex moved far apart (see Results and Discussion section). In order to capture the dispersion interactions implicitly, an empirical Grimme-type dispersion correction was added to the CRF in standard AM1, and thereafter all parameters in the Gaussian AM1 terms were reoptimized to best fit this dispersion-corrected CRF by using a nonlinear least-squares procedure (see Appendix A for technical details). The original CRF, the added dispersion corrections, the sum of these two terms, and the best fit Gaussian terms for the H−H, C−C, N−N, and O−O atom pairs are plotted in Figure 1. The sum and fitted curves for the first three pairs are visually almost indistinguishable, and there are merely minor variations of the standard AM1 CRF parameters (Figure 1). In the case of the O−O pair, the fitted Gaussian terms for oxygen, however, are not able to reproduce the attractive well in the van der Waals region due to the presence of two positive (repulsion) Gaussian functions ($K > 0$). Since oxygen is chemically less important for the DHFR reaction, the standard AM1 parameters for oxygen in the Gaussian terms were retained except in the very last step (4) of the SRP parametrization. The dispersion corrected CRF parameters for hydrogen, carbon, and nitrogen were used in the subsequent optimization procedure described above (fixed in steps 1−3, and adjusted in step 4).

Following the strategy outlined above, two SRP Hamiltonians were developed and employed in the simulations on DHFR. Specifically, an initial model termed AM1-SRP was designed, where the target complex structures and properties employed in the SRP development process were obtained from B3LYP.

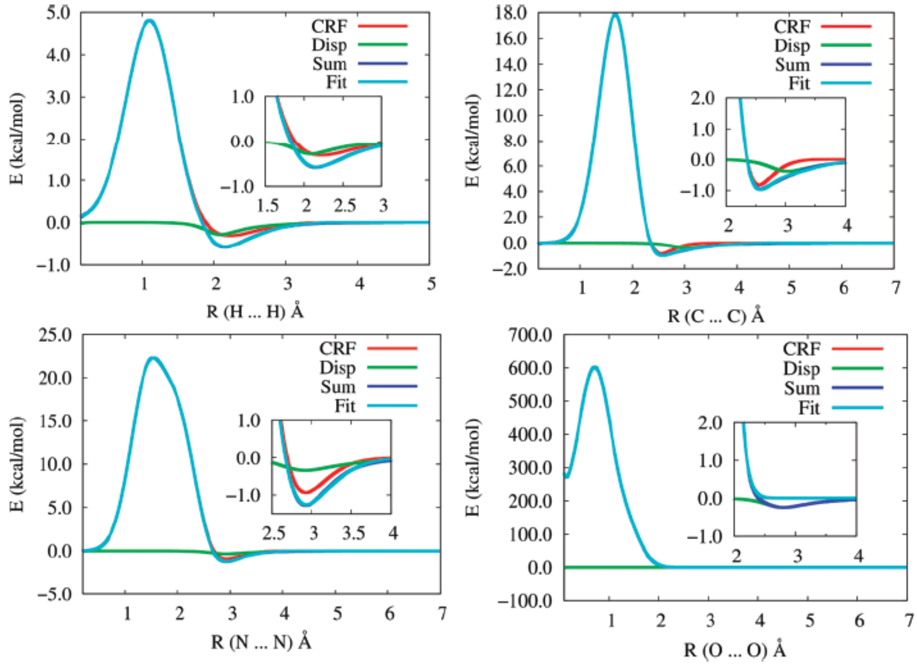


Figure 1. The nonlinear least-squares curve fittings for pairs of hydrogen, carbon, nitrogen, and oxygen atoms. CRF is the standard AM1 core repulsion function. Disp is the dispersion energy correction calculated using Grimme's formula. Sum is the total energy of CRF and Disp. Fit is the Gaussian term fitted with respect to Sum. A magnified subplot of the van der Waals region is also shown for each atom pair.

In this AM1-SRP model, intermolecular dispersion effects are not included during the parametrization process. Subsequently, a SRP Hamiltonian which implicitly includes dispersion was devised [termed AM1-SRP(D)]. Here, all complex structures were obtained from B3LYP-D calculations. In both approaches, the target data for the individual molecules were taken from G4MP2 calculations (accounting for intramolecular dispersion) at B3LYP optimized geometries.

Target values are presented in the Supporting Information, along with a detailed explanation of their determination, validation of the AM1-SRP performance, and a complete list of the modified parameters. The AM1-SRP results presented in the Supporting Information were obtained using the Gaussian 09 program.

Modeling QM/MM Interactions. To investigate the ability of the developed AM1-SRP Hamiltonians to accurately model QM/MM interactions, we computed the complexation energies between selected QM molecules and a TIP3P water molecule. The QM molecules are 6-Me-H_3pterin$^+$, *trans*-Me-H_2nic, 6-Me-H_4pterin, and *trans*-Me-Hnic$^+$. For each of these QM moieties, a single TIP3P water molecule was placed at different hydrogen bonding positions around the molecule, for a total of up to four complexes per QM molecule. The QM/MM complexation energies were compared to complexation energies computed using various high-level methods.

QM/MM Simulations of *E. coli* DHFR. *Model of the Ternary Complex of E. coli DHFR.* The crystal structure of *E. coli* DHFR with folate and the oxidized cofactor NADP$^+$ (PDB ID code: 1rx2), originally reported by Sawaya and Kraut,[20] was used to construct the initial configuration for the present study. This structure corresponds to the Michaelis complex of *E. coli* DHFR with the Met20 loop in the closed conformation. The X-ray crystal structure contains a total of 159 amino acid residues, 153 crystallographic waters, and the folate and NADP$^+$ ligands,[20] which were replaced by H_2folate and NADPH, respectively, for the simulation. The N5 atom on the substrate's pyrazine ring was protonated, as this form is thought to be the active species for the hydride transfer reaction.[11,70]

On the basis of the results of Callender and co-workers,[10,11,15] Asp27 was assumed to be deprotonated at neutral pH, while the protonation states for all other ionizable residues were set corresponding to pH 7.[5,71,72] The hydrogen bonding patterns of the ionizable residues with the surrounding environment were visually inspected to verify that the protonation states are reasonable. The coordinates of hydrogen atoms of the protein, water, and coenzyme were determined using the HBUILD facility in the program CHARMM.[73,74] The possible protonation states of histidine residues (proton on Nε, proton on Nδ, or doubly protonated form) were determined by examination of the hydrogen bonding interactions. Peripheral/surface His residues were generally assumed to be positively charged. In NADPH, the 2′-phosphate moiety on the adenosine ribose was treated as a dianion, based on a pK_a of ~5.9 reported for NADP$^+$ bound in cytochrome P-450 oxidoreductase (P-450R).[75] The coordinates of NE2 and OE1 atoms of the carboxamide moiety in the Gln102 residue were swapped, with respect to the original solved crystal structure, in order to form plausible hydrogen bond interactions with the adenine group of NADPH (a similar analysis was carried out by Brooks et al.[76]). Residue 37 was built as Asp37 to be consistent with the 1RX2 PDB entry and previous work on DHFR, although sequencing data suggest that this residue might be Asn37.[77,78] We note that the DHFR function is insensitive to the nature of the amino acid at position 37.[79] The resulting negatively charged enzyme (−14) has dimensions of ca. 34 × 42 × 50 Å^3. To this system, we added 14 sodium ions in random positions outside the protein to obtain a net-neutral system, a prerequisite for convergence of the Ewald summation method (*vide infra*).[80] Subsequently, the protein, ligands, crystal waters, and counterions were embedded in a water box as detailed below.

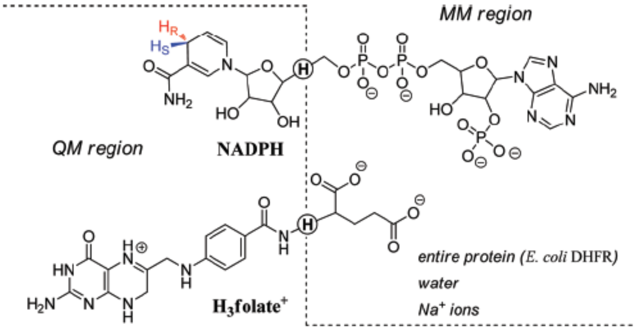


Figure 2. QM/MM partitioning scheme. The dashed line divides the QM and MM regions, and the quantum hydrogen link atoms are circled.

Hybrid QM(SRP)/MM Potential Energy Surface. The hydride transfer reaction in *E. coli* DHFR was described using a hybrid QM/MM potential energy surface.

$$\hat{H} = \hat{H}_{QM} + \hat{H}_{MM} + \hat{H}_{QM/MM} \tag{2}$$

The system was partitioned into a QM region consisting of 69 atoms and a MM region containing the rest of the system. The QM subsystem includes 38 atoms from the $H_3folate^+$ substrate (the pterin ring, the *N*-methylene-substituted *p*-aminobenzoyl (pABA) moiety, and the NH atoms of the glutamate moiety) and 29 atoms from the NADPH coenzyme (the dihydronicotinamide and ribose rings). In addition, two hydrogen link atoms were introduced along the covalent bonds crossing the boundary between the QM and the MM regions, to satisfy the valence requirements of the QM fragments. A schematic representation is depicted in Figure 2, where the quantum link atoms are circled.

The QM region was treated by the AM1-SRP or the AM1-SRP(D) Hamiltonians described above. The all-atom CHARMM22 force field[81] with grid-based energy correction maps (CMAP)[82] for peptide dihedral angles was employed to treat the entire protein, the substrate, and the ions, while the CHARMM27 force field[81] was used for the coenzyme. The water molecules were represented by the three-point charge TIP3P model.[83] Atom types and parameters for the substrate were assigned by analogy with existing functional groups, as reported by Garcia-Viloca et al.[7] For structure minimization and initial equilibration at the pure MM level, partial atomic charges for the substrate were assigned using the CHARMm force field (Momany-Rone) integrated in Discovery Studio 2.5 (Accelrys Software Inc., San-Diego, CA).

QM/MM interactions were treated by electrostatic embedding wherein the MM partial atomic charges are included in the one-electron Hamiltonian. To fine-tune the PES, QM/MM interaction energies between the reacting fragments (QM) and the protein (MM) were modified. The van der Waals (vdW) parameters of the QM hydrogen atoms were changed to those that reproduce the interaction energies for hydrogen bonded complexes in the gas phase obtained from *ab initio* calculations at the MP2/6-31G(d,p) level.[84]

Free-Energy Simulations—General Approach.[46] We follow a two-step procedure[85] in which we first carry out Newtonian MD simulations to determine the classical mechanical potential of mean force (PMF) along the reaction coordinate for the hydride transfer reaction between $H_3folate^+$ and NADPH in the fully solvated enzyme. Then, atoms that are directly involved in the hydride transfer are quantized, and the configurations sampled in MD simulations are used in path-integral simulations by constraining the centroid positions of the quantized particles to the classical coordinates. This double (quantum and classical) averaging scheme is formally rigorous[46,85−88] and yields the QM-PMF as a function of the centroid reaction path.[89,90] In PI-FEP/UM, the ratio of the quantum partition functions for different isotopes, which yields the kinetic isotope effects (KIEs), is obtained by free-energy perturbation from a light isotope mass into a heavier one within the same centroid path-integral simulation,[46] avoiding the difference between two free-energy barriers with greater fluctuations than the difference itself for the two isotopic reactions. Consequently, the PI-FEP/UM method is unique in that it yields accurate results for computed KIEs, including secondary KIEs.[45,46]

MD Simulations. MD simulations were conducted under periodic boundary conditions (PBC), with Ewald summation for electrostatic interactions.[91] The solute was soaked in a pre-equilibrated 65 × 65 × 65 $Å^3$ cubic box of 9461 water molecules, with its longest axis lying along the space diagonal of the box to ensure that all protein atoms are at least 10 Å away from the edges of the box. The final model contained 27 986 atoms. For van der Waals and electrostatic interactions, a 13.0 Å group-based cutoff was used. The Ewald method was employed for reciprocal space summations between MM sites as well as for the QM/MM interactions using a 64 × 64 × 64 FFT grid.[91] The κ value was set to 0.340 $Å^{-1}$.

All water molecules were relaxed using the adopted-basis set Newton−Raphson (ABNR) minimization method (30 steps), while the crystal water oxygens were harmonically restrained to their original positions. This was followed by a 100 ps MD equilibration of the water molecules, which were thereafter minimized again (30 steps ABNR). Afterward, all atoms were subjected to minimization in a stepwise fashion, to remove close contacts in the initial protein−ligand−solvent system: (a) The substrate and coenzyme molecules were first minimized (30 step ABNR) while placing harmonic restraints on heavy atoms and keeping the rest of the system fixed. The restraints were gradually decreased to zero, while the ligands were further minimized (5 × 30 steps ABNR). (b) The water molecules and protein molecules were minimized (this time the ligands were held fixed) while the harmonic restraints on their heavy atoms were gradually diminished (4 × 10 steps ABNR). (c) Eventually, the whole system was minimized (30 steps ABNR) without any restraints.

The isothermal−isobaric ensemble (NPT) was employed at 1 atm and 298 K using the extended system pressure/temperature (CPT) algorithm of Andersen[92] with an effective mass of 500 amu and a Hoover thermostat[93] with an effective mass of 1000 kcal/mol·ps^2. The SHAKE algorithm[94] was applied to constrain all MM bonds involving hydrogen atoms, allowing a time-step of 1 fs. The system was gradually heated up from 48 to 298 K during five sessions of 5 ps for a total of 25 ps and thereafter equilibrated at the target temperature (298 K) over the course of 1 ns at the MM level of theory, with a further 200 ps of equilibration using the QM(SRP)/MM potential.

In light of the flexibility of the protein and the structural manipulation of the original ligands bound in the crystal structure, some issues emerged during the equilibration phase which required intervention. This included transient introduction of nuclear Overhauser effect (NOE) harmonic restraints on the distance between hydride donor and acceptor carbons (C4N in NADPH and C6 in $H_3folate^+$, respectively), as well as on other distances between donors and acceptors of selected hydrogen bonds within the protein which are characteristic of the closed conformation (for details, see the Supporting Information). All restraints were removed 100 ps prior to commencing the production phase.

Potential of Mean Force. The classical-mechanical potential of mean force (CM-PMF)[95] was determined using the umbrella sampling technique, in order to sample the high-energy regions of the potential energy surface.[96] The reaction coordinate (ζ) was defined geometrically as the difference between the lengths of the breaking ($C4N_{NADPH}-H4N$) and forming ($H4N-C6_{H_3folate^+}$) bonds. A total of 13 discrete regions along the reaction coordinate ("windows") were defined with a uniform spacing of 0.25 Å. Each simulation was performed with the addition of a biasing potential (roughly the negative of the computed PMF), and a harmonic restraint centered at each window. The harmonic force constants, k, ranged between 20.0 and 60.0 kcal·mol^{-1}·Å^{-2} [$E_{harm} = k(\zeta - \zeta_{ref})^2$]. Each window was equilibrated for 2 ps, followed by a 100 ps production simulation that collected the probability densities of configurations (ρ) along the reaction coordinate (ζ) and sorted them into bins of width 0.01 Å. The coordinates were saved every 0.5 ps, and the velocities and positions of the last configuration generated in each window were used to initiate the next window. The PMF curve was obtained using the weighted histogram analysis method (WHAM).[97] To ensure convergence of the PMF, the simulations were run until the difference between sequential PMF profiles was less than ±1 kcal/mol. The QM-PMF was obtained using a double averaging procedure by centroid path-integral simulations on configurations saved during the umbrella sampling.[46,85,87] In essence, the centroid path-integral simulations yield the free energy difference between the classical mechanical and the quantum mechanical PMFs.[46,85,87] For each isotope, a quantized correction curve was fit to the PI simulation data using an inverted Eckart function. The curve fitting was done using the Levenberg–Marquardt algorithm, and the inverted Eckart potential was added to the CM-PMF to obtain the QM-PMF.

Kinetic Isotope Effects. For the primary KIE (k_H^H/k_D^H), the *pro-R* hydrogen (the donated hydride, H4N) is substituted with deuterium, whereas the secondary KIE (k_H^H/k_H^D) involves the geminal *pro-S* hydrogen (H42N). To evaluate the KIEs, the centroid path-integral simulations were carried out for the light isotopic reaction, and the ratio of the partition functions between two isotopic reactions was determined by free-energy perturbation theory from the light mass into a heavier one.[46] In the present study, we quantized the donor ($C4N_{NADPH}$) and acceptor ($C6_{H_3folate^+}$) carbons, in addition to the *pro-R* (H4N) and *pro-S* (H42N) hydrogens connected to the donor carbon (in the reactant state). Each quantized particle was represented by 32 beads. We used a bisection sampling technique[88] in all centroid path-integral simulations, and 10 free-particle configurations were sampled for each of 10 200 classical configurations, yielding a total of 102 000 path-integral sampling steps.

■ RESULTS AND DISCUSSION

Gas Phase QM Calculations. *Optimized Geometries.* According to the X-ray crystal structure of synthetically prepared 6-methyl-7,8-dihydropterin-monohydrochloride-monohydrate (6-Me-H_2pterin·HCl, which is essentially 6-Me-H_3pterin$^+$), the heterocyclic ring members form a perfectly planar structure.[98] However, our gas phase DFT calculations, in particular at the B3LYP/6-31G(2df,p) level, suggest that the framework of the pyrazine ring in the dihydropterins is not completely planar, but rather a pucker is observed as the C7 methylene carbon projects above or below the plane formed by all other ring atoms (Figure 3). The magnitude of deviation from the plane varies with the selection of the method and basis set. MP2/6-311+G(2d,p) optimization supports the nonplanar nature of 6-Me-H_3pterin$^+$ predicted by DFT. The discrepancy between experimental results and theory could be due to crystal forces in the experimental structure. Conversely, the observed crystal structure could be the average of two structures puckered in opposite directions. As for the reduced equilibrium structures (6-Me-H_4pterin and 6-Me-H_3pterin$^-$), the pyrazine ring adopts a half-chair conformation, where C6 and C7 lie below and above the plane, respectively, in a staggered conformation with respect to each other, and the methyl group attached to C6 takes a pseudo-equatorial position. These findings are in good agreement with ^{1}H NMR studies on tetrahydropterin derivatives (spin–spin coupling constants measurements).[99–102] A comparison of the crystal structure of 6-Me-H_3pterin$^+$ and the computed geometry at the B3LYP/6-31G(2df,p) level is available in the Supporting Information.

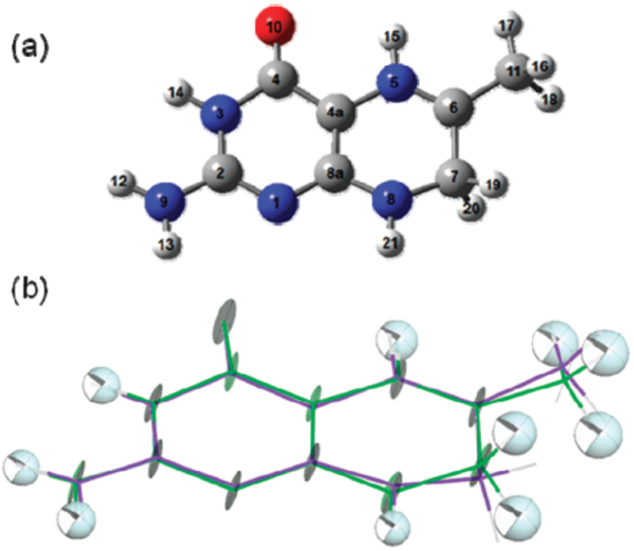


Figure 3. (a) Optimized geometry of 6-Me-H_3pterin$^+$ calculated at the B3LYP/6-31G(2df,p) level of theory. (b) Superimposition of the X-ray crystal structure of 6-methyl-7,8-dihydropterin-monohydrochloride-monohydrate (green) with the computed geometry of 6-Me-H_3pterin$^+$ (purple). Isotropic and anisotropic temperature factors corresponding to atoms in the crystal structure are represented by thermal ellipsoids.

The degree of pyramidality of the C2-exocyclic amino group can be defined by its torsion angles and tilting from the plane. While in the 6-Me-H_3pterin$^+$ structure, this group is nearly coplanar with the 4-oxopyrimidine ring, it exhibits a substantial pyramidalization in the other pterin derivatives examined. As this amino group can be seen as a fragment of a guanidine-like moiety, it is relevant to mention that the solid-state structure of free base guanidine, recently determined by X-ray diffraction,[103] indicates a nonplanar geometry with pyramidal amino groups, in accordance with earlier *ab initio* calculations.[104] A pyramidal geometry is also exhibited at the N8 position in 6-Me-H_3pterin$^-$ (but not in the oxidized form), as well as at the N5 site in 6-Me-H_4pterin.

As for the nicotinamides, it is noteworthy that the *cisoid* conformer was found to be slightly more stable *in vacuo* (with the exception of the unsubstituted neutral molecules, *cis*/*trans*-nic), while the conformer identified in X-ray crystal structures of many

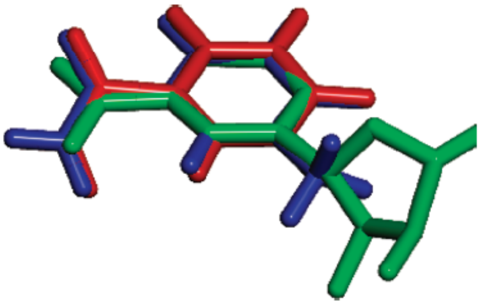


Figure 4. Superimposition of the oxidized nicotinamide portion in three models: segment from the NADP$^+$ cofactor (green), coordinates taken from the crystal structure of *E. coli* DHFR ternary complex (PDB ID: 1rx2), *trans*-Me-Hnic$^+$ (blue), and *trans*-Hnic$^+$ (red) computed with B3LYP/6-31G(2df,p). Note the differences in the orientation of the carboxamide relative to the pyridine ring.

enzyme active sites is the *transoid* form.[20,67,68] One may attribute the preference of the *transoid* species in the enzyme to specific interactions with the environment (hydrophilic/hydrophobic, polar/nonpolar), which may offset the relatively small energy difference between the two conformers.[105] There are some noticeable differences between the geometries of the *cisoid* and the *transoid* conformers. The degree of distortion of the carboxamide group with respect to the pyridine plane can be expressed in terms of the dihedral angle C2−C3−C7═O8 (Chart 1), as the difference between a "perfect" antiperiplanar angle (180°, for the *transoid* conformer) or synperiplanar angle (0°, for the *cisoid* conformer) and the actual dihedral. Calculations with B3LYP/6-31G(2df,p) suggest that in the *transoid* species, the carboxamide is rotated substantially out-of-plane, with distortions of 18.4−19.7° and 27.5−28.4° for the dihydronicotinamides and their corresponding oxidized counterparts. However, the out-of-plane rotations observed in the *cisoid* conformers are considerably smaller, with values of ca. 2.1° and 13.0−13.2° for H_2nic and Hnic$^+$ derivatives, respectively. These observations are in agreement with previous theoretical studies.[105−108] In the X-ray crystal structure of the ternary complex *ec*DHFR:folate:NADP$^+$ (PDB code: 1rx2) used in the QM/MM simulations in this study, the nicotinamide subunit is found in its *transoid* conformer, with out-of-plane distortion of only 6° (Figure 4). Moreover, in most of the *transoid* conformers, there is a considerable pyramidalization of the amide nitrogen. These two geometric features of the carboxamide —the out-of-plane distortion and the *N*-pyramidalization—help to relieve unfavorable steric interactions between the NH_2 group and C2−H bond in the pyridine ring.

The 1,4-dihydropyridine ring skeleton in both unsubstituted and *N*-Me-substituted dihydronicotinamides is nearly planar, and no apparent puckering into a boat conformation is observed. This finding is in accord with X-ray data for some synthetically produced and isolated *N*-substituted nicotinamides, which are essentially planar in the dihydropyridine ring system.[109,110]

Model Reactions. In Table 1, the calculated energies, enthalpies, and free energies at room temperature are given for the "protonated" (a) and "unprotonated" (b) model reactions involving *N*-methyl substituted nicotinamides. The results are reported separately for each of the four thermodynamic pathways described in the Methodology section (eqs 1a, 1b).

All methods except AM1 predict an exothermic (and exergonic) reaction for the "protonated" model. The reaction energies can be ordered in absolute magnitude as follows: *cisoid* → *transoid* < *cisoid* → *cisoid* ≈ *transoid* → *transoid* < *transoid* → *cisoid* (this trend is opposite for the "unprotonated" model). The variation between these pathways stems from the slightly higher energy of the *transoid* conformer of the nicotinamide species relative to its *cisoid* counterpart, either in the reduced or oxidized form. For the density functionals, the relative order of ΔE_{el} (and generally also of ΔH_r and ΔG_r) in absolute values is PBEPBE < M06 < PBE1PBE < BB1K ≈ MPWB1K < B3LYP ≈ B98. The results obtained with CBS-QB3 and G4MP2 are almost identical yet considerably more exothermic than those obtained from DFT (ΔE_{el} being ~3 kcal/mol more negative than in the case of B3LYP, which is closest to the *ab initio* target data).

On the other hand, the "unprotonated" model reaction was found to be highly endothermic (and endergonic) with most of the computed ΔG_r values ranging between 120 and 134 kcal/mol. This emphasizes the crucial role of N5-protonation at the pyrazine ring of pterin in facilitating a thermodynamically feasible hydride transfer from dihydronicotinamide to dihydropterin in the gas phase.

Reaction, Transition, and Product Complexes. The diagram in Figure 5 compares the relative energies of the reactant, transition, and product states (RS, TS, and PS), for the reaction 6-Me-H_3pterin$^+$ + *trans*-Me-H_2nic → 6-Me-H_4pterin + *trans*-Me-Hnic$^+$, using AM1 and various density functionals.

Focusing on the bimolecular complexes, we define the energy barrier, $\Delta E^{\ddagger}$, as the energy gap between the TS and RS complexes, and the reaction energy ΔE_r as the difference between the PS and RS complexes. These quantities are poorly predicted by the standard AM1 method, as demonstrated by a high barrier ($\Delta E^{\ddagger}$ = 27.8 kcal/mol) and a slightly endothermic reaction (ΔE_r = 1.9 kcal/mol). The DFT methods suggest a much lower barrier and an exothermic reaction. The gradient corrected functional PBEPBE, which does not contain Hartree−Fock exchange, yields the lowest barrier, $\Delta E^{\ddagger}$ = 3.5 kcal/mol. There is good agreement between the functionals B3LYP, BB1K, MPWB1K, and M06, while B3LYP-D gives a slightly lower barrier. It is interesting to note that the computed gas-phase barriers (e.g., using M06, $\Delta E^{\ddagger}$ = 12.2 kcal/mol) are quite similar to the experimental free-energy barrier in the enzyme, 13.4 kcal/mol.[5]

The effect of dispersion is clearly seen by inspecting the energies of the fully separated molecules relative to the RS/PS complexes. Using the M06 and B3LYP-D functionals, which both account for dispersion interactions, there is a considerable additional stabilization of the complexes. At the B3LYP-D level, this dispersion effect may be estimated directly by comparing with B3LYP, yielding ca. 11 kcal/mol for the RS and PS. We note that the PS complexes presented in Figure 5 correspond to a nearly T-stacked conformation between the pterin and nicotinamide rings, similar to that found in DHFR. For methods that include dispersion, an additional minimum configuration corresponding to planar stacking between the rings is found. This point is discussed further below.

Considering the close agreement between B3LYP and M06 for the current system, we employ B3LYP (and B3LYP-D) as target data for the complexes in the SRP parametrization process. This choice is consistent with the target data for the individual molecules being taken from the G4MP2 approach, which employs B3LYP geometries and thermodynamic data.

SRP Development. The AM1 parameters were reoptimized against high-level QM results, obtained from G4MP2 calculations as well as B3LYP and B3LYP-D calculations with the 6-31+G(d,p) basis set. The properties used as target values

Table 1. Energetics Calculated for the "Protonated" (a) and "Unprotonated" (b) Model Reactions in the Gas Phase[a]

	Part a											
	cisoid → *cisoid*[b]			*cisoid* → *transoid*[c]			*transoid* → *transoid*[d]			*transoid* → *cisoid*[e]		
method	ΔE_{el}	ΔH_r	ΔG_r	ΔE_{el}	ΔH_r	ΔG_r	ΔE_{el}	ΔH_r	ΔG_r	ΔE_{el}	ΔH_r	ΔG_r
AM1		7.4	8.5		9.0	9.8		7.1	7.6		5.5	6.4
B3LYP/6-31+G(d,p)	−7.5	−6.9	−6.6	−5.6	−5.0	−4.6	−7.7	−7.0	−7.0	−9.6	−9.0	−9.0
PBE1PBE/6-31+G(d,p)	−6.7	−6.1	−5.6	−4.7	−4.0	−3.8	−6.9	−6.3	−6.5	−8.8	−8.3	−8.3
PBEPBE/6-31+G(d,p)	−5.3	−4.6	−4.2	−3.5	−2.7	−2.8	−5.2	−4.7	−4.8	−7.0	−6.6	−6.2
B98/6-31+G(d,p)	−7.6	−7.0	−6.5	−5.7	−5.1	−5.5	−7.7	−7.2	−7.8	−9.6	−9.1	−8.7
BB1K/6-31+G(d,p)	−7.0	−6.5	−5.7	−4.9	−4.5	−4.5	−7.3	−6.8	−7.2	−9.4	−8.9	−8.5
MPWB1K/6-31+G(d,p)	−7.1	−6.7	−5.5	−5.0	−4.5	−5.1	−7.4	−6.9	−7.8	−9.5	−9.1	−8.2
M06/6-31+G(d,p)	−6.3	−5.8	−4.8	−4.5	−4.0	−2.2	−6.6	−6.0	−4.8	−8.4	−7.9	−7.4
G4MP2	−10.7	−10.1	−9.5	−8.7	−8.1	−7.6	−10.0	−9.5	−9.0	−12.0	−11.5	−10.9
CBS-QB3	−10.8	−10.1	−9.8	−8.7	−8.0	−7.7	−10.2	−9.6	−9.5	−12.2	−11.7	−11.6

	Part b											
	cisoid → *cisoid*[f]			*cisoid* → *transoid*[g]			*transoid* → *transoid*[h]			*transoid* → *cisoid*[i]		
method	ΔE_{el}	ΔH_r	ΔG_r	ΔE_{el}	ΔH_r	ΔG_r	ΔE_{el}	ΔH_r	ΔG_r	ΔE_{el}	ΔH_r	ΔG_r
AM1		129.6	130.7		131.1	132.0		129.2	129.8		127.7	128.6
B3LYP/6-31+G(d,p)	125.1	124.9	125.1	127.0	126.8	127.1	124.9	124.8	124.7	123.0	122.8	122.7
PBE1PBE/6-31+G(d,p)	127.2	126.8	127.2	129.2	128.8	129.0	127.0	126.5	126.3	125.1	124.5	124.5
PBEPBE/6-31+G(d,p)	121.7	121.1	121.2	123.5	122.9	122.6	121.8	120.9	120.5	120.0	119.1	119.1
B98/6-31+G(d,p)	125.5	124.9	125.4	127.4	126.9	126.3	125.4	124.8	124.0	123.6	122.9	123.1
BB1K/6-31+G(d,p)	131.4	131.0	131.8	133.6	133.0	133.0	131.2	130.7	130.3	129.1	128.6	129.1
MPWB1K/6-31+G(d,p)	131.7	131.2	132.5	133.9	133.3	132.9	131.5	131.0	130.2	129.3	128.8	129.8
M06/6-31+G(d,p)	129.0	128.2	129.0	130.8	130.1	131.6	128.7	128.0	129.0	126.9	126.2	126.4
G4MP2	124.9	124.3	124.7	126.9	126.3	126.5	125.6	125.0	125.1	123.6	123.0	123.2
CBS-QB3	125.2	124.6	124.7	127.2	126.7	126.8	125.8	125.1	125.0	123.7	123.0	122.9

[a] ΔE_{el} is the change in electronic energy; ΔH_r and ΔG_r are the respective enthalpy and free energy changes at 298.15 K and 1 atm (including zero-point contributions). [b] 6-Me-H_3pterin$^+$ + *cis*-Me-H_2nic $\rightleftharpoons$ 6-Me-H_4pterin + *cis*-Me-Hnic$^+$. [c] 6-Me-H_3pterin$^+$ + *cis*-Me-H_2nic $\rightleftharpoons$ 6-Me-H_4pterin + *trans*-Me-Hnic$^+$. [d] 6-Me-H_3pterin$^+$ + *trans*-Me-H_2nic $\rightleftharpoons$ 6-Me-H_4pterin + *trans*-Me-Hnic$^+$. [e] 6-Me-H_3pterin$^+$ + *trans*-Me-H_2nic $\rightleftharpoons$ 6-Me-H_4pterin + *cis*-Me-Hnic$^+$. [f] 6-Me-H_2pterin + *cis*-Me-H_2nic $\rightleftharpoons$ 6-Me-H_3pterin$^-$ + *cis*-Me-Hnic$^+$. [g] 6-Me-H_2pterin + *cis*-Me-H_2nic $\rightleftharpoons$ 6-Me-H_3pterin$^-$ + *trans*-Me-Hnic$^+$. [h] 6-Me-H_2pterin + *trans*-Me-H_2nic $\rightleftharpoons$ 6-Me-H_3pterin$^-$ + *trans*-Me-Hnic$^+$. [i] 6-Me-H_2pterin + *trans*-Me-H_2nic $\rightleftharpoons$ 6-Me-H_3pterin$^-$ + *cis*-Me-Hnic$^+$.

included heats of formation, dipole moments, Mulliken charges, and vibrational frequencies of the individual species. Furthermore, the reaction energies for the model complexes also served as reference data that contributed to the fitness function of the SRP model (*vide infra*), in order to improve the accuracy of the resulting PES.

Determination of the target heats of formation required some caution, as we are interested in a purely electronic structure description of the potential energy surface, while nuclear classical and quantum effects are added only at a later stage during enzyme simulations. Therefore, any thermal contribution should be excluded. On the other hand, the AM1 energies are originally interpreted in terms of heats of formations.[58] In the present AM1-SRP reparametrization, we mainly use ab initio and DFT energies as target data, and therefore the resulting AM1-SRP energies will mimic electronic energies. This allows the AM1-SRP models to be employed in molecular dynamics simulations where all classical thermal effects are included directly via the propagation of Newton's equations of motion, while all quantum thermal effects are included via path-integral simulations.

Table 2 compares the root-mean-square deviations (RMSD) from the target data for various properties calculated at the optimized geometries in three parametrization schemes: standard AM1, AM1-SRP, and AM1-SRP(D). The target data for the 14 molecules are the same for the three parametrization schemes. The AM1-SRP parametrization did not involve the heats of formation of the pterin molecules due to the large gap between the corresponding target and AM1 values, an obstacle which was circumvented by minimizing the errors associated with related relative energies (see Table 3). Indeed, AM1-SRP produced the smallest RMSD errors in relative energies among the parametrization schemes, and the quality of the individual heats of formations was considerably improved by ca. 8 kcal/mol when going from standard AM1 to AM1-SRP. There were significant changes in the AM1-SRP(D) parameters compared with standard AM1, in particular in the Gaussian terms. This further reduced the RMSD error in the heats of formation to ca. 1 kcal/mol, including the pterin species (whose heats of formation were incorporated in the training set of AM1-SRP(D)). At the same time, the error in the relative energies was increased only slightly relative to AM1-SRP, making its energetic accuracy comparable to the latter. The RMSD error for the vibrational wavenumbers was reduced by 29 cm^{-1} and 55 cm^{-1} in AM1-SRP and AM1-SRP(D), respectively, relative to AM1. On the other hand, the

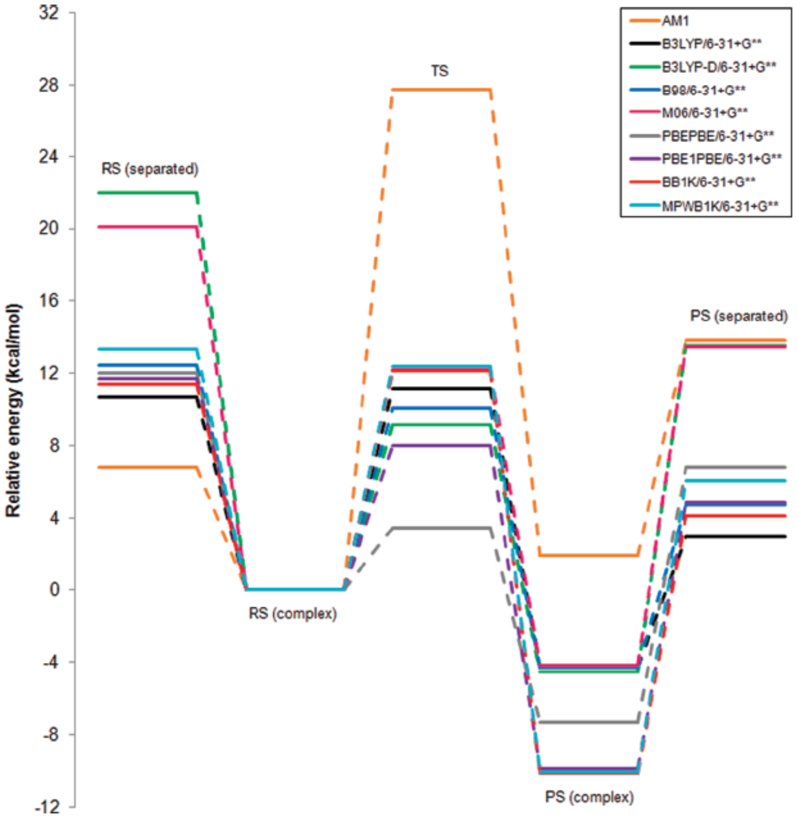


Figure 5. The relative energies (kcal/mol) of the reactant, transition, and product states associated with the reaction 6-Me-H$_3$pterin$^+$ + *trans*-Me-H$_2$nic → 6-Me-H$_4$pterin + *trans*-Me-Hnic$^+$. The fully separated reactants and products are represented by RS (separated) and PS (separated), respectively, where RS (complex) and PS (complex) refer to the corresponding bimolecular complexes.

Table 2. RMSD Errors for Properties Calculated at Optimized Geometries Using AM1, AM1-SRP, and AM1-SRP(D), with Respect to Target Values (G4MP2)

properties	AM1	AM1-SRP	AM1-SRP(D)
heats of formation[a] (kcal/mol)	12.1	3.8	1.3
relative energies[b] (kcal/mol)	9.5	1.9	2.2
bond lengths[c] (Å)	0.02	0.03	0.04
bond angles[c] (deg)	1.8	3.4	1.9
vibrational wavenumbers[d] (cm^{-1})	108	78	52
Mulliken atomic charges[c] (*e*)	0.16	0.27	0.20
dipole moments[e] (D)	1.27	1.08	(1.47)

[a] The statistics refer to nicotinamide derivatives only. The target heats of formation for the pterin species were far off the AM1 values and were thus omitted from the calibration set for AM1-SRP [but included in the AM1-SRP(D) parametrization]. [b] The statistics refer to the reactions described in Table 3. [c] The statistics refer to all 14 individual species. [d] The statistics refer to wavenumbers larger than 2500 cm^{-1} for all 14 individual species. [e] The statistics refer to eight neutral species out of the 14 individual species. Dipole moments were not included as reference data in the parametrization of AM1-SRP(D), hence the parentheses.

RMSD of the bond lengths, bond angles, and atomic charges calculated with AM1-SRP were slightly increased relative to AM1 by 0.01 Å, 1.6°, and 0.11*e*, whereas AM1-SRP(D) did somewhat better with 0.02 Å, 0.1°, and 0.04*e*, respectively. However, their overall quality was considered to be acceptable in light of the large improvements for the energies and vibrational frequencies. The dipole moments calculated with all three schemes were of similar quality.

Table 3 provides further insight into the quality of selected relative energies obtained with each Hamiltonian. It lists the deviations for various reactions involving four and two species (classes I and II, respectively), with respect to the reference entries. As demonstrated by the total RMSD error for each class, AM1-SRP performs better for reactions of class I, while generally AM1-SRP(D) treats class II transitions more accurately (though to a less pronounced degree).

Having the tailored semiempirical Hamiltonians in hand, some structural and thermodynamic features of the reactant, transition, and product state complexes were analyzed and compared against the target DFT methods (Table 4). The analysis includes geometric relations between the hydride-donating and accepting carbons (C4′ and C6, respectively) and the transferring hydride itself, as well as the imaginary vibrational frequency and energetic relations. The three complexes are depicted in Figure 6, comparing geometries that were obtained using DFT and SRP Hamiltonians.

Overall, there is good qualitative agreement between the computed semiempirical structures of the reactant and transition state complexes and the corresponding DFT reference structures, and the improvements of the SRP Hamiltonians over standard AM1 are substantial. In particular, the AM1-SRP Hamiltonian is able to reproduce the imaginary vibrational frequency of the reactive normal mode in the transition state as

Table 3. Signed Errors of Relative Energies (kcal/mol) Calculated with the AM1-SRP and AM1-SRP(D) Parameters, with Respect to the Target Data (G4MP2)[a]

		signed error (kcal/mol)		
class	reaction	AM1	AM1-SRP	AM1-SRP(D)
I	6-Me-H_3pterin$^+$ + *trans*-Me-H_2nic → 6-Me-H_4pterin + *trans*-Me-Hnic$^+$	17.1	1.0	3.1
	6-Me-H_3pterin$^+$ + *cis*-Me-H_2nic → 6-Me-H_4pterin + *cis*-Me-Hnic$^+$	18.2	1.9	4.2
	6-Me-H_3pterin$^+$ + *trans*-H_2nic → 6-Me-H_4pterin + *trans*-Hnic$^+$	14.2	−2.3	−1.1
	6-Me-H_3pterin$^+$ + *cis*-H_2nic → 6-Me-H_4pterin + *cis*-Hnic$^+$	15.1	−1.4	−0.2
	6-Me-H_2pterin + *trans*-Me-H_2nic → 6-Me-H_3pterin$^-$ + *trans*-Me-Hnic$^+$	3.8	0.6	2.0
	6-Me-H_2pterin + *cis*-Me-H_2nic → 6-Me-H_3pterin$^-$ + *cis*-Me-Hnic$^+$	4.8	1.6	3.1
	6-Me-H_2pterin + *trans*-H_2nic → 6-Me-H_3pterin$^-$ + *trans*-Hnic$^+$	0.9	−2.7	−2.2
	6-Me-H_2pterin + *cis*-H_2nic → 6-Me-H_3pterin$^-$ + *cis*-Hnic$^+$	1.8	−1.7	−1.3
RMSD		11.7	1.8	2.5
II	6-Me-H_3pterin$^+$ $\xrightarrow{H^-}$ 6-Me-H_4pterin	15.7	1.6	1.3
	6-Me-H_2pterin $\xrightarrow{H^-}$ 6-Me-H_3pterin$^-$	2.4	1.3	0.2
	trans-Me-H_2nic $\xrightarrow[H^-]{}$ *trans*-Me-Hnic$^+$	1.4	−0.6	1.8
	cis-Me-H_2nic $\xrightarrow[H^-]{}$ *cis*-Me-Hnic$^+$	2.5	0.3	3.0
	trans-H_2nic $\xrightarrow[-H^-]{}$ *trans*-Hnic$^+$	−1.5	−3.9	−2.4
	cis-H_2nic $\xrightarrow[-H^-]{}$ *cis*-Hnic$^+$	−0.6	−3.0	−1.5
	cis-nic → *trans*-nic	0.5	0.3	0.6
RMSD		6.1	2.1	1.8

[a] Classes I and II assemble relative energies involving four and two species, respectively.

predicted by B3LYP/6-31+G(d,p) calculations (i.e., −880 cm^{-1} vs −838 cm^{-1}). On the other hand, the calculated imaginary frequency of the TS complex with AM1-SRP(D), −960 cm^{-1}, resembles more that predicted by M06, −953 cm^{-1}, than the corresponding B3LYP-D value, −716 cm^{-1}.

The energetic profiles obtained at different theoretical levels are presented in Figure 7. As expected, the AM1-SRP complexation energies are fairly close to those of B3LYP (−9.3 and −10.7 kcal/mol for RS, −3.0 and −7.2 kcal/mol for PS, respectively), while the AM1-SRP(D) results are similar to those of B3LYP-D (−22.8 and −22.0 kcal/mol for RS, −25.2 and −27.0 kcal/mol for PS; the counterpoise corrections for basis set superposition error (BSSE) with B3LYP-D are 1.5 and 2.1 kcal/mol for RS and PS, respectively). The barrier height obtained with AM1-SRP is 10.7 kcal/mol, comparable to 11.2 kcal/mol with B3LYP, while that obtained with AM1-SRP(D) is 13.6 kcal/mol, which is somewhat higher than the target value 9.2 kcal/mol obtained with B3LYP-D. In comparison, the M06 barrier height is 12.2 kcal/mol. An in-depth analysis of the RS and PS complexes will be presented below.

A detailed inspection of the geometry optimizations of the RS and PS complexes reveals that the PES in these regions is very flat with several plausible minima, particularly in the PS region. Using B3LYP and the AM1-SRP Hamiltonian, we could attain reasonable geometries of the RS and PS complexes via the IRC path from the TS only down to a certain point away from the TS in either direction (these structures are presented in Figure 6). Full geometry optimization of the RS and PS complexes in the gas phase yielded highly distorted minimum structures, with no stacking, which do not resemble the configuration in the DHFR active site. As the two molecular segments in these complexes are weakly bound in the van der Waals region (>3.5 Å), it is necessary to include dispersion interactions (which are missing in the AM1 formalism) to accurately reproduce these geometries. Furthermore, the target complexes for AM1-SRP were derived from B3LYP calculations which do not include much dispersion. Indeed, the introduction of dispersion by means of B3LYP-D enabled us to obtain plausible minimum RS and PS geometries.

B3LYP-D and M06 optimizations of the PS both reveal two possible minimum configurations, with regard to the orientation of the nicotinamide ring toward the pterin fragment. (a) First is a "T-stacked" configuration, where the nicotinamide ring lies in a quasi-perpendicular plane with respect to the pterin ring. With this orientation, the complexation energy obtained with B3LYP-D is −18.1 kcal/mol, while M06 affords a related minimum geometry with a complexation energy of −17.6 kcal/mol. (b) Next is a fully "stacked" configuration, in which the rings are stacked one on top of the other, which corresponds to the global minimum of the PES of the PS complex (Figures 6 and 7). The B3LYP-D structure (complexation energy: −27.0 kcal/mol) is almost identical to the fully stacked complex obtained with M06 (−23.3 kcal/mol) and resembles that obtained with AM1-SRP(D) (−25.2 kcal/mol). With AM1-SRP(D), we could not find a partially T-stacked configuration, suggesting that this

Table 4. Geometric and Thermodynamic Properties of the Reactant, Transition and Product State Complexes in the Gas Phase, Calculated with Different Hamiltonians[a]

		R (Å)					
complex	method	C4′−H	C6−H	C4′−C6	∠C4′−H−C6 (deg)	IFreq (cm^{-1})	ΔE (kcal/mol)
reactant state	AM1[b]	1.128	2.640	3.636	146.61		0.0
	B3LYP/6-31+G(d,p)[b]	1.110	2.996	3.863	135.95		0.0
	AM1-SRP[b]	1.110	2.551	3.593	155.91		0.0
	B3LYP-D/6-31+G(d,p)[c]	1.103	2.751	3.500	124.85		0.0
	AM1-SRP(D)[c]	1.087	2.283	3.170	137.49		0.0
	M06[c]	1.104	3.301	3.944	118.15		0.0
transition state	AM1	1.411	1.363	2.763	169.81	−1385	27.8
	B3LYP/6-31+G(d,p)	1.309	1.409	2.715	174.61	−838	11.2
	AM1-SRP	1.317	1.348	2.633	162.43	−880	10.7
	B3LYP-D/6-31+G(d,p)	1.280	1.396	2.664	169.09	−716	9.2
	AM1-SRP(D)	1.308	1.322	2.602	163.18	−960	13.6
	M06	1.313	1.362	2.667	171.01	−953	12.2
product state	AM1[b]	2.512	1.136	3.507	145.37		1.9
	B3LYP/6-31+G(d,p)[b]	2.533	1.110	3.613	163.96		−4.3
	AM1-SRP[b]	2.514	1.116	3.503	147.07		−2.8
	B3LYP-D/6-31+G(d,p)[c,d]	2.679	1.110	3.632	143.52		−4.6
		(5.069)	(1.100)	(5.394)	(101.32)		(−13.5)
	AM1-SRP(D)[c]	4.248	1.092	4.748	110.82		−9.3
	M06[c,d]	2.499	1.107	3.443	142.35		−4.2
		(5.034)	(1.107)	(5.706)	(107.44)		(−9.8)

[a] C4′ is the donor carbon in the nicotinamide subunit, C6 is the acceptor carbon in the pterin subunit, and H denotes the transferring hydride. IFreq is the imaginary frequency, and ΔE is the relative energy with respect to the reactant complex. [b] The final structure was obtained using IRC calculations. [c] The final structure was obtained using geometry optimization. [d] Values without parentheses refer to the local minimum structure with a "T-stacked" configuration. Values in parentheses refer to the fully stacked configuration, which corresponds to the global minimum on the potential energy surface.

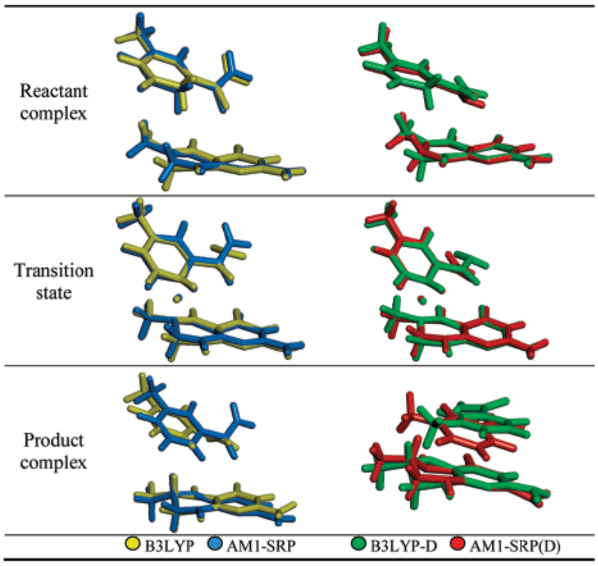


Figure 6. Structures of the reactant, transition, and product state complexes in the gas phase. The superimposed geometries were obtained by calculations at a DFT and a related semiempirical level. Structures are shown for two pairs of methods: B3LYP/6-31+G(d,p) and AM1-SRP and B3LYP-D/6-31+G(d,p) and AM1-SRP(D). The product state complexes represent the T-stacked configuration for B3LYP and AM1-SRP optimized structures and the fully stacked configuration for the B3LYP-D and AM1-SRP(D) optimized structures.

method might "over-stack" the PS complex. The greater stability of the fully stacked complexes may be attributed to the presence of dispersion interactions, hydrogen bonding interactions within the bimolecular complex, and close contacts between the π systems in the nicotinamide and pterin rings.

The gradient norm of the product complex during AM1-SRP(D) geometry optimization is plotted in Figure S1 (Supporting Information). The gradient is greatly reduced in the first steps. The PES then becomes quite flat and shallow. Thus, the structure of the product complex seems to be highly flexible, and many arrangements of the two molecular segments have relatively small gradients. The use of less stringent convergence criteria in the AM1-SRP(D) geometry optimization would have led to termination after about 100 cycles, at a structure close to the T-stacked configuration obtained with B3LYP-D (Figure S1).

QM/MM Interactions. To investigate the ability of the AM1-SRP Hamiltonians to accurately model QM/MM interactions, we computed the complexation energies between selected QM moieties (6-Me-H$_3$pterin$^+$, *trans*-Me-H$_2$nic, 6-Me-H$_4$pterin, and *trans*-Me-Hnic$^+$) and a TIP3P water molecule. For each of these QM moieties, a single water molecule was placed at different hydrogen bonding positions around the molecule, for a total of 14 QM/MM complexes (Table S8a, Supporting Information). The QM/MM interaction energies were computed using AM1, AM1-SRP, or AM1-SRP(D) and a TIP3P water molecule. These interaction energies were compared with data from M06, B3LYP-D, B3LYP, and HF, all in conjunction with the

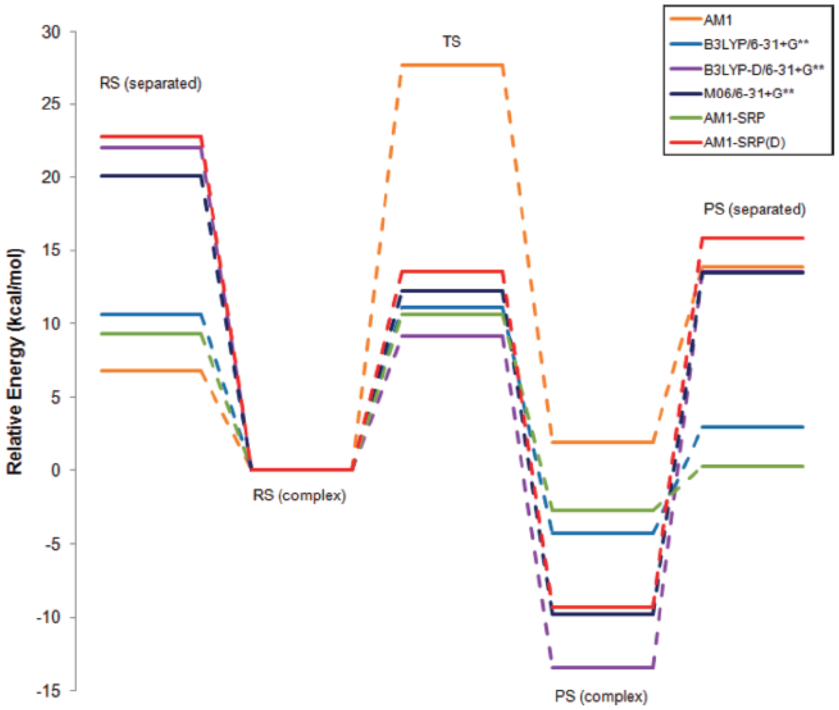


Figure 7. The relative energies of the reactant, transition and product states associated with the reaction 6-Me-H_3pterin$^+$ + *trans*-Me-H_2nic → 6-Me-H_4pterin + *trans*-Me-Hnic$^+$, for standard AM1, B3LYP/6-31+G**, B3LYP-D/6-31+G**, M06/6-31+G**, AM1-SRP, and AM1-SRP(D). The sums of the energies of the individual subunits for the reactants and products are represented by RS (separated) and PS (separated), respectively, where RS (complex) and PS (complex) refer to the corresponding bimolecular complexes.

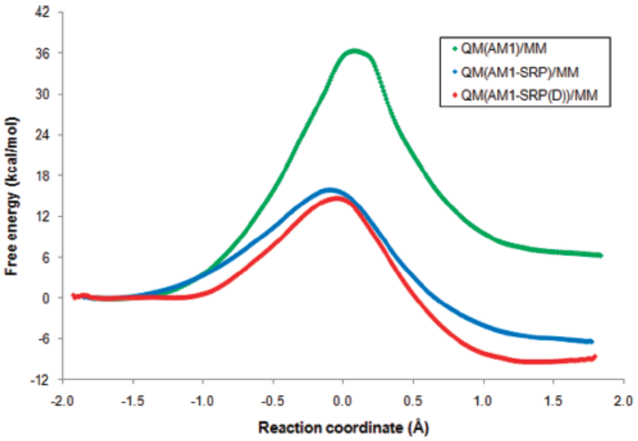


Figure 8. Computed classical potentials of mean force for the hydride transfer reaction catalyzed by *E. coli* DHFR at 298 K, obtained with three different QM/MM schemes distinguished by the semiempirical treatment of the QM region: AM1/MM (green), AM1-SRP/MM (blue), and AM1-SRP(D)/MM (red). The reaction coordinate is defined as the difference between the distances of the transferring hydride and the donor and acceptor carbon atoms.

6-31+G(d,p) basis set. BSSE corrections were included for the DFT methods but not for HF. The different semiempirical QM/MM methods give similar interaction energies, indicating that the SRP optimization process did not introduce artificial polarization of the molecules (Table S8b and Figure S2, Supporting Information). The QM/MM methods predict slightly weaker complexes than M06 and B3LYP-D (RMSDs of 2.6−4.4 kcal/mol), while they are in good agreement with B3LYP and HF (RMSDs between 1.2 and 2.4 kcal/mol).

Enzyme Simulations. *Classical Potential of Mean Force.* Figure 8 shows the classical mechanical PMF (CM-PMF) for the hydride transfer reaction in DHFR obtained from free energy MD simulations, using the AM1, AM1-SRP, and AM1-SRP(D) QM/MM Hamiltonians. We note that these data are not directly comparable to experimental results as NQEs are not yet included at this stage. The AM1/MM Hamiltonian substantially overestimates the free energy barrier ($\Delta G^{\ddagger}$ = 36.3 kcal/mol) and predicts an endothermic reaction (ΔG^{r} = 6.3 kcal/mol). The transition state (the free energy bottleneck of the PMF) for the QM(AM1-SRP)/MM Hamiltonian is located at $\zeta^{\ddagger} = -0.089$ Å, whereas that of the QM(AM1-SRP(D))/MM Hamiltonian is placed at $\zeta^{\ddagger} = -0.045$ Å. Both are relatively late compared to that reported by Gao and co-workers in recent QM/MM studies ($\zeta^{\ddagger} = -0.145$ Å) on *ec*DHFR, which utilized the AM1 Hamiltonian with SVB correction (*vide supra*).[7,8] We further note that the end points of the AM1-SRP and AM1-SRP(D) QM/MM free energy profiles are smooth and show no artificial free-energy increase. The classical free energies of reaction and activation extracted from the QM(AM1-SRP)/MM-based free energy profile are −6.4 and 15.9 kcal/mol. The CM-PMF obtained by the enhanced Hamiltonian, QM(AM1-SRP(D))/MM, predicts a classical mechanical free energy barrier which is narrower and slightly lower ($\Delta G^{\ddagger}$ = 14.7 kcal/mol), and the reaction is more exergonic (ΔG^{r} = −9.3 kcal/mol) by 3 kcal/mol. The considerable stabilization of the product state by ca. 5 kcal/mol compared with the experimental data is possibly a manifestation of the "over-stacking" phenomenon already encountered with the AM1-SRP(D) in the gas phase (*vide supra*). Finally, the differences

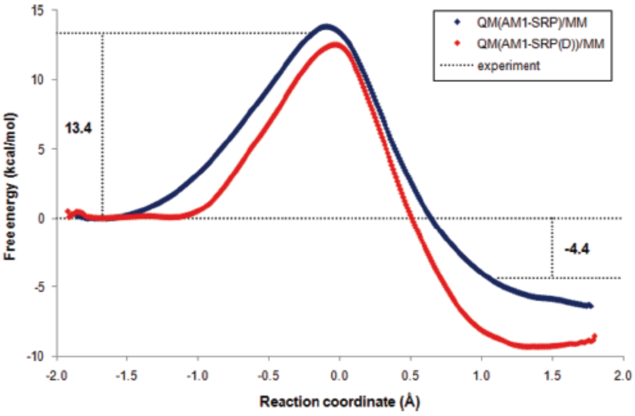


Figure 9. The quantum mechanical potentials of mean force for the hydride transfer reaction in *E. coli* DHFR. The centroid coordinates are used in path-integral simulations. Each quantized particle was represented by 32 beads. The experimental free energies of reaction and activation are denoted near the dashed lines.

in the free energy barrier width among the three QM/MM Hamiltonians are consistent with the trends in the imaginary vibrational frequency of the transition state in the gas phase obtained from the corresponding semiempirical schemes. According to Table 4, the imaginary wavenumbers in absolute values are ordered as follows: AM1 > AM1-SRP(D) > AM1-SRP. Indeed, the QM(AM1)/MM free energy profile has the narrowest barrier, that of QM(AM1-SRP(D))/MM is wider, and the QM(AM1-SRP)/MM Hamiltonian yields a PMF with the widest barrier.

Quantum Potential of Mean Force. The QM-PMF is obtained from Feynman path-integral calculations,[46,86,88] in which the centroid positions of the discrete paths of quantized particles are used to specify the reaction coordinate.[46,89,90] The "quantum" free energy profiles displayed in Figure 9 describe the hydride transfer reaction with the two SRP QM/MM Hamiltonians. Using QM(AM1-SRP)/MM, the inclusion of NQE in the simulations[46] lowers the computed free energies of activation for the hydride and deuteride transfer by 2.1 and 1.4 kcal/mol, respectively, relative to the "classical" free energy barrier. The resulting quantum free energies of reaction and activation for the hydride transfer, −6.4 and 13.8 kcal/mol, are in good accord with the corresponding experimental results (−4.4 and 13.4 kcal/mol).[5] The quantum corrections for the AM1-SRP(D)/MM CM-PMF are very similar, 2.2 and 1.5 kcal/mol for hydride and deuteride, so that the predicted free energy barrier (12.5 kcal/mol) is in close agreement with experimental results. The free-energy results demonstrate that the present QM/MM and path integral methods can provide an adequate description of the hydride transfer reaction in DHFR.

Kinetic Isotope Effects. The computed primary (k_H^H/k_D^H) and secondary (k_H^H/k_H^D) KIEs for the hydride transfer reaction in DHFR at 298 K are 3.51 ± 0.14 and 1.18 ± 0.06 with QM(AM1-SRP)/MM and 3.49 ± 0.16 and 1.11 ± 0.04 with QM(AM1-SRP(D))/MM, respectively. These values are in good agreement with the experimental intrinsic KIEs measured by Kohen and coworkers (k_H^H/k_D^H = 3.55 ± 0.17; k_H^H/k_H^D = 1.13 ± 0.01),[72] providing additional evidence for the accuracy of our computational treatments. These KIEs are also in good agreement with various QM/MM calculations (e.g., refs 7 (2° KIEs = 1.13) and 39 (1° KIEs = 3.4 ± 0.6)), but here the same method is within experimental error for both 1° and 2° KIEs.

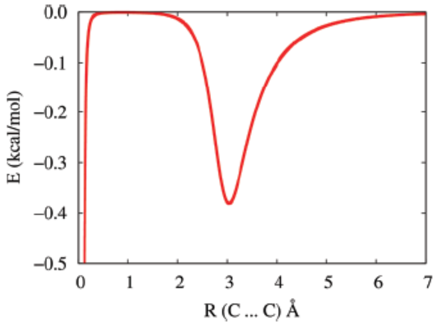


Figure 10. The dispersion energy between two carbon atoms calculated using Grimme's formula.

SUMMARY

In the current work, we presented extensive benchmark calculations for several model reactions in the gas phase that are relevant to the DHFR catalyzed hydride transfer. We employed G4MP2 and CBS-QB3 ab initio calculations as well as numerous density functional methods. Using these results as target data, we developed two specific reaction parameter (SRP) Hamiltonians by reparametrization of the semiempirical AM1 method. The first generation SRP Hamiltonian does not account for dispersion, while the second generation SRP includes dispersion implicitly via the AM1 core-repulsion functions. These SRP semiempirical Hamiltonians were subsequently used in hybrid quantum mechanics/molecular mechanics simulations of the DHFR catalyzed reaction. The classical PMFs were computed using the standard AM1 method as well as the AM1-SRP and AM1-SRP(D) models. Nuclear quantum effects were included using a Feynman path-integral method. Finally, kinetic isotope effects were computed using a mass-perturbation-based path-integral approach. The quantum PMFs predict free energy barriers and reaction free energies in good agreement with available experimental kinetic data.

We conclude that the resulting PESs yield accuracies comparable to those obtained at the G4MP2 and DFT levels, with a computational cost that is several orders of magnitude less. This will allow us to perform long MD simulations of the solvated enzyme, while providing a realistic description of the kinetics and thermodynamic properties in the DHFR catalyzed reaction.

APPENDIX A

The dispersion energies were estimated using the formula introduced by Grimme for density functional methods.

$$E_{\text{disp}}(r_{ij}) = -s_6 \sum_{i=1}^{N} \sum_{j=i+1}^{N-1} \frac{C_{6,ij}}{r_{ij}^6} f_{\text{d}}(r_{ij})$$

$$f_{\text{d}}(r_{ij}) = \frac{1}{1 + \exp\{-d[(r_{ij}/r_0) - 1]\}}$$

where r_{ij} is the distance between atoms i and j. s_6 is a global scaling factor for the dispersion energy, with numerical values ranging from 0.75 to 1.2 for different density functionals. Herein, we chose the value 1.0 for simplicity. d is a parameter for damping function f_d. It was chosen to be 20.0 in line with the value used in DFT-D2. The atomic C_6 coefficients and van der Waals radii r_0 were taken directly from the original publication. Figure 10 plots the calculated dispersion energy for two carbon atoms.

The sum of the standard AM1 core repulsion function and the dispersion energy correction was scanned for each atom pair. A total of 396 points for the carbon, nitrogen, and oxygen pairs and 346 points for the hydrogen pair were collected from a distance of 0.1 Å with a step size of 0.02 Å. Finally, the parameters in the Gaussian terms were fitted to these data points using a nonlinear least-squares procedure implemented in gnuplot with the standard AM1 parameters as an initial guess. The convergence criterion was 1.0×10^{-6}.

■ ASSOCIATED CONTENT

S Supporting Information. Comprehensive list of the gas phase molecular models; summary of recent computational studies on the DHFR-catalyzed hydride transfer reaction; vibrational frequency scale factors used for gas phase thermochemistry calculations; details of the development of AM1-SRP, calibration, and related target values; selected gas phase vibrational frequencies; list of the semiempirical parameters of AM1-SRP and AM1-SRP(D); coordinates of geometries calculated at the target levels; comparison between the crystal structure and computed geometry of 6-Me-H_3pterin$^+$; ligand−water QM/MM complexation energies; and further details with regard to the MD simulations of the solvated enzyme. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: majort@biu.ac.il.

■ ACKNOWLEDGMENT

This work has been supported by the Israel Science Foundation and the United States−Israel Binational Science Foundation (Grant # 2007256).

■ REFERENCES

(1) Schnell, J. R.; Dyson, H. J.; Wright, P. E. Structure, dynamics, and catalytic function of dihydrofolate reductase. *Annu. Rev. Biophys. Biomol. Struct.* **2004**, *33*, 119–140.

(2) Huennekens, F. In search of dihydrofolate reductase. *Protein Sci.* **1996**, *5*, 1201.

(3) Frey, P. A.; Hegeman, A. D. *Enzymatic reaction mechanisms*; Oxford University Press: New York, 2007.

(4) Castillo, R.; Andrés, J.; Moliner, V. Catalytic Mechanism of Dihydrofolate Reductase Enzyme. A Combined Quantum-Mechanical/Molecular-Mechanical Characterization of Transition State Structure for the Hydride Transfer Step. *J. Am. Chem. Soc.* **1999**, *121*, 12140–12147.

(5) Fierke, C. A.; Johnson, K. A.; Benkovic, S. J. Construction and evaluation of the kinetic scheme associated with dihydrofolate reductase from Escherichia coli. *Biochemistry* **1987**, *26*, 4085–4092.

(6) Morrison, J. F.; Stone, S. R. Mechanism of the reaction catalyzed by dihydrofolate reductase from *Escherichia coli*: pH and deuterium isotope effects with NADPH as the variable substrate. *Biochemistry* **1988**, *27*, 5499–5506.

(7) Garcia-Viloca, M.; Truhlar, D. G.; Gao, J. Reaction-Path Energetics and Kinetics of the Hydride Transfer Reaction Catalyzed by Dihydrofolate Reductase. *Biochemistry* **2003**, *42*, 13558–13575.

(8) Pu, J.; Ma, S.; Garcia-Viloca, M.; Gao, J.; Truhlar, D. G.; Kohen, A. Nonperfect Synchronization of Reaction Center Rehybridization in the Transition State of the Hydride Transfer Catalyzed by Dihydrofolate Reductase. *J. Am. Chem. Soc.* **2005**, *127*, 14879–14886.

(9) Maharaj, G.; Selinsky, B.; Appleman, J.; Perlman, M.; London, R.; Blakley, R. Dissociation constants for dihydrofolic acid and dihydrobiopterin and implications for mechanistic models for dihydrofolate reductase. *Biochemistry* **1990**, *29*, 4554.

(10) Chen, Y. Q.; Kraut, J.; Blakley, R. L.; Callender, R. Determination by Raman Spectroscopy of the pKa of N5 of Dihydrofolate Bound to Dihydrofolate Reductase: Mechanistic Implications. *Biochemistry* **1994**, *33*, 7021–7026.

(11) Deng, H.; Callender, R. Structure of Dihydrofolate When Bound to Dihydrofolate Reductase. *J. Am. Chem. Soc.* **1998**, *120*, 7730–7737.

(12) Deng, H.; Callender, R.; Howell, E. Vibrational Structure of Dihydrofolate Bound to R67 Dihydrofolate Reductase. *J. Biol. Chem.* **2001**, *276*, 48956–48960.

(13) Howell, E.; Villafranca, J.; Warren, M.; Oatley, S.; Kraut, J. Functional role of aspartic acid-27 in dihydrofolate reductase revealed by mutagenesis. *Science* **1986**, *231*, 1123–1128.

(14) David, C. L.; Howell, E. E.; Farnum, M. F.; Villafranca, J. E.; Oatley, S. J.; Kraut, J. Structure and function of alternative proton-relay mutants of dihydrofolate reductase. *Biochemistry* **1992**, *31*, 9813–9822.

(15) Chen, Y.; Kraut, J.; Callender, R. pH-dependent conformational changes in *Escherichia coli* dihydrofolate reductase revealed by Raman difference spectroscopy. *Biophys. J.* **1997**, *72*, 936–941.

(16) Rod, T. H.; Brooks, C. L., III. How Dihydrofolate Reductase Facilitates Protonation of Dihydrofolate. *J. Am. Chem. Soc.* **2003**, *125*, 8718–8719.

(17) Bolin, J.; Filman, D.; Matthews, D.; Hamlin, R.; Kraut, J. Crystal structures of *Escherichia coli* and *Lactobacillus casei* dihydrofolate reductase refined at 1.7 Å resolution. I. General features and binding of methotrexate. *J. Biol. Chem.* **1982**, *257*, 13650.

(18) Bystroff, C.; Oatley, S.; Kraut, J. Crystal structures of *Escherichia coli* dihydrofolate reductase: The NADP$^+$ holoenzyme and the folate·NADP$^+$ ternary complex. Substrate binding and a model for the transition state. *Biochemistry* **1990**, *29*, 3263–3277.

(19) Bystroff, C.; Kraut, J. Crystal structure of unliganded *Escherichia coli* dihydrofolate reductase. Ligand-induced conformational changes and cooperativity in binding. *Biochemistry* **1991**, *30*, 2227–2239.

(20) Sawaya, M.; Kraut, J. Loop and Subdomain Movements in the Mechanism of *Escherichia coli* Dihydrofolate Reductase: Crystallographic Evidence. *Biochemistry* **1997**, *36*, 586–603.

(21) Donkersloot, M. C. A.; Buck, H. M. The hydride-donation reaction of reduced nicotinamide adenine dinucleotide. 2. MINDO/3 and STO-3G calculations on the role of the carbamoyl group in enzymic reactions. *J. Am. Chem. Soc.* **1981**, *103*, 6554–6558.

(22) Cummins, P. L.; Gready, J. E. Mechanistic aspects of biological redox reactions involving NADH 2: A combined semiempirical and *ab initio* study of hydride-ion transfer between the NADH analogue, 1-methyldihydronicotinamide, and folate and dihydrofolate analogue substrates of dihydrofolate reductase. *J. Comput. Chem.* **1990**, *11*, 791–804.

(23) Andrés, J.; Safont, V. S.; Martins, J. B. L.; Beltrán, A.; Moliner, V. AM1 and PM3 transition structure for the hydride transfer. A model of reaction catalyzed by dihydrofolate reductase. *THEOCHEM* **1995**, *330*, 411–416.

(24) Andrés, J.; Moliner, V.; Safont, V. S.; Domingo, L. R.; Picher, M. T.; Krechl, J. On Transition Structures for Hydride Transfer Step: A Theoretical Study of the Reaction Catalyzed by Dihydrofolate Reductase Enzyme. *Bioorg. Chem.* **1996**, *24*, 10–18.

(25) Cummins, P. L.; Ramnarayan, K.; Singh, U. C.; Gready, J. E. Molecular dynamics/free energy perturbation study on the relative affinities of the binding of reduced and oxidized NADP to dihydrofolate reductase. *J. Am. Chem. Soc.* **1991**, *113*, 8247–8256.

(26) Cummins, P. L.; Gready, J. E. Molecular dynamics and free energy perturbation study of hydride-ion transfer step in dihydrofolate reductase using combined quantum and molecular mechanical model. *J. Comput. Chem.* **1998**, *19*, 977–988.

(27) Ferrer, S.; Silla, E.; Tuñón, I.; Martí, S.; Moliner, V. Catalytic Mechanism of Dihydrofolate Reductase Enzyme. A Combined Quantum-Mechanical/Molecular-Mechanical Characterization of the N5 Protonation Step. *J. Phys. Chem. B* **2003**, *107*, 14036–14041.

(28) Cummins, P. L.; Greatbanks, S. P.; Rendell, A. P.; Gready, J. E. Computational Methods for the Study of Enzymic Reaction Mechanisms. 1. Application to the Hydride Transfer Step in the Catalysis of Dihydrofolate Reductase. *J. Phys. Chem. B* **2002**, *106*, 9934–9944.

(29) Thorpe, I. F.; Brooks, C. L. Barriers to Hydride Transfer in Wild Type and Mutant Dihydrofolate Reductase from E. coli. *J. Phys. Chem. B* **2003**, *107*, 14042–14051.

(30) Thorpe, I. F.; Brooks, C. L. Conformational Substates Modulate Hydride Transfer in Dihydrofolate Reductase. *J. Am. Chem. Soc.* **2005**, *127*, 12997–13006.

(31) Ranganathan, S.; Gready, J. E. Hybrid Quantum and Molecular Mechanical (QM/MM) Studies on the Pyruvate to l-Lactate Interconversion in l-Lactate Dehydrogenase. *J. Phys. Chem. B* **1997**, *101*, 5614–5618.

(32) Turner, A.; Moliner, V.; Williams, I. Transition-state structural refinement with GRACE and CHARMM: Flexible QM/MM modelling for lactate dehydrogenase. *Phys. Chem. Chem. Phys.* **1999**, *1*, 1323–1331.

(33) Faulder, P. F.; Tresadern, G.; Chohan, K. K.; Scrutton, N. S.; Sutcliffe, M. J.; Hillier, I. H.; Burton, N. A. QM/MM Studies Show Substantial Tunneling for the Hydrogen-Transfer Reaction in Methylamine Dehydrogenase. *J. Am. Chem. Soc.* **2001**, *123*, 8604–8605.

(34) Ferrer, S.; Ruiz-Pernía, J. J.; Tuñón, I.; Moliner, V.; Garcia-Viloca, M.; González-Lafont, A.; Lluch, J. M. A QM/MM Exploration of the Potential Energy Surface of Pyruvate to Lactate Transformation Catalyzed by LDH. Improving the Accuracy of Semiempirical Descriptions. *J. Chem. Theory Comput.* **2005**, *1*, 750–761.

(35) Devi-Kesavan, L. S.; Garcia-Viloca, M.; Gao, J. Semiempirical QM/MM potential with simple valence bond (SVB) for enzyme reactions. Application to the nucleophilic addition reaction in haloalkane dehalogenase. *Theor. Chem. Acc.* **2003**, *109*, 133–139.

(36) Truhlar, D. G.; Gao, J.; Alhambra, C.; Garcia-Viloca, M.; Corchado, J.; Sánchez, M. L.; Villà, J. The Incorporation of Quantum Effects in Enzyme Kinetics Modeling. *Acc. Chem. Res.* **2002**, *35*, 341–349.

(37) Warshel, A.; Weiss, R. M. An empirical valence bond approach for comparing reactions in solutions and in enzymes. *J. Am. Chem. Soc.* **1980**, *102*, 6218–6226.

(38) Warshel, A. *Computer Modeling of Chemical Reactions in Enzymes*; Wiley: New York, 1991.

(39) Agarwal, P. K.; Billeter, S. R.; Hammes-Schiffer, S. Nuclear quantum effects and enzyme dynamics in dihydrofolate reductase catalysis. *J. Phys. Chem. B* **2002**, *106*, 3283–3293.

(40) Wong, K. F.; Selzer, T.; Benkovic, S. J.; Hammes-Schiffer, S. Chemical theory and computation special feature: impact of distal mutations on the network of coupled motions correlated to hydride transfer in dihydrofolate reductase. *Proc. Nat. Acad. Sci. U.S.A.* **2005**, *102*, 6807–6812.

(41) Liu, H.; Warshel, A. The Catalytic Effect of Dihydrofolate Reductase and Its Mutants Is Determined by Reorganization Energies. *Biochemistry* **2007**, *46*, 6011–6025.

(42) Hwang, J. K.; Warshel, A. A quantized classical path approach for calculations of quantum mechanical rate constants. *J. Phys. Chem.* **1993**, *97*, 10053–10058.

(43) Rossi, I.; Truhlar, D. G. Parameterization of NDDO wavefunctions using genetic algorithms. An evolutionary approach to parameterizing potential energy surfaces and direct dynamics calculations for organic reactions. *Chem. Phys. Lett.* **1995**, *233*, 231–236.

(44) Major, D. T.; York, D. M.; Gao, J. L. Solvent polarization and kinetic isotope effects in nitroethane deprotonation and implications to the nitroalkane oxidase reaction. *J. Am. Chem. Soc.* **2005**, *127*, 16374–16375.

(45) Gao, J. L.; Wong, K. Y.; Major, D. T. Combined QM/MM and path integral simulations of kinetic isotope effects in the proton transfer reaction between nitroethane and acetate ion in water. *J. Comput. Chem.* **2008**, *29*, 514–522.

(46) Major, D. T.; Gao, J. L. An integrated path integral and free-energy perturbation-umbrella sampling method for computing kinetic isotope effects of chemical reactions in solution and in enzymes. *J. Chem. Theory Comput.* **2007**, *3*, 949–960.

(47) Major, D. T.; Gao, J. L. A combined quantum mechanical and molecular mechanical study of the reaction mechanism and alpha-amino acidity in alanine racemase. *J. Am. Chem. Soc.* **2006**, *128*, 16345–16357.

(48) Major, D. T.; Heroux, A.; Orville, A. M.; Valley, M. P.; Fitzpatrick, P. F.; Gao, J. Differential quantum tunneling contributions in nitroalkane oxidase catalyzed and the uncatalyzed proton transfer reaction. *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *106*, 20734–20739.

(49) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, J. A., Jr.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, N. J.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. *Gaussian 09*, revision B.01; Gaussian, Inc.: Wallingford, CT, 2009.

(50) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865.

(51) Adamo, C.; Cossi, M.; Barone, V. An accurate density functional method for the study of magnetic properties: the PBE0 model. *THEOCHEM* **1999**, *493*, 145–157.

(52) Becke, A. D. Density-functional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* **1993**, *98*, 5648–5652.

(53) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B: Condens. Matter* **1988**, *37*, 785.

(54) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. *Ab Initio* Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *J. Phys. Chem.* **1994**, *98*, 11623–11627.

(55) Schmider, H. L.; Becke, A. D. Optimized density functionals from the extended G2 test set. *J. Chem. Phys.* **1998**, *108*, 9624–9631.

(56) Zhao, Y.; Lynch, B. J.; Truhlar, D. G. Development and Assessment of a New Hybrid Density Functional Model for Thermochemical Kinetics. *J. Phys. Chem. A* **2004**, *108*, 2715–2719.

(57) Zhao, Y.; Truhlar, D. The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other functionals. *Theor. Chem. Acc.* **2008**, *120*, 215–241.

(58) Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. AM1: a new general purpose quantum mechanical molecular model. *J. Am. Chem. Soc.* **1985**, *107*, 3902–3909.

(59) Montgomery, J. J. A.; Frisch, M. J.; Ochterski, J. W.; Petersson, G. A. A complete basis set model chemistry. VII. Use of the minimum population localization method. *J. Chem. Phys.* **2000**, *112*, 6532–6542.

(60) Montgomery, J. J. A.; Frisch, M. J.; Ochterski, J. W.; Petersson, G. A. A complete basis set model chemistry. VI. Use of density functional geometries and frequencies. *J. Chem. Phys.* **1999**, *110*, 2822–2827.

(61) Curtiss, L. A.; Redfern, P. C.; Raghavachari, K. Gaussian-4 theory. *J. Chem. Phys.* **2007**, *126*, 084108–12.

(62) Curtiss, L. A.; Redfern, P. C.; Raghavachari, K. Gaussian-4 theory using reduced order perturbation theory. *J. Chem. Phys.* **2007**, *127*, 124105.

(63) Armarego, W. L. F.; Waring, P.; Williams, J. W. Absolute configuration of 6-methyl-5,6,7,8-tetrahydropterin produced by enzymic reduction (dihydrofolate reductase and NADPH) of 6-methyl-7,8-dihydropterin. *J. Chem. Soc., Chem. Commun.* **1980**, 334–336.

(64) Peng, C.; Schlegel, H. B. Combining synchronous transit and quasi-Newton methods to find transition states. *Isr. J. Chem.* **1994**, *33*, 449–54.

(65) Grimme, S. Semiempirical GGA-type density functional constructed with a long-range dispersion correction. *J. Comput. Chem.* **2006**, *27*, 1787–1799.

(66) Schwabe, T.; Grimme, S. Double-hybrid density functionals with long-range dispersion corrections: higher accuracy and extended applicability. *Phys. Chem. Chem. Phys.* **2007**, *9*, 3397–3406.

(67) Benach, J.; Atrian, S.; Gonzàlez-Duarte, R.; Ladenstein, R. The catalytic reaction and inhibition mechanism of Drosophila alcohol dehydrogenase: observation of an enzyme-bound NAD-ketone adduct at 1.4 Å resolution by X-ray crystallography. *J. Mol. Biol.* **1999**, *289*, 335–355.

(68) Kavanagh, K. L.; Klimacek, M.; Nidetzky, B.; Wilson, D. K. Crystal Structure of Pseudomonas fluorescens Mannitol 2-Dehydrogenase Binary and Ternary Complexes. *J. Biol. Chem.* **2002**, *277*, 43433–43442.

(69) Major, D. T.; Nam, K.; Gao, J. L. Transition state stabilization and alpha-amino carbon acidity in alanine racemase. *J. Am. Chem. Soc.* **2006**, *128*, 8114–8115.

(70) Gready, J. E. Theoretical studies on the activation of the pterin cofactor in the catalytic mechanism of dihydrofolate reductase. *Biochemistry* **1985**, *24*, 4761–4766.

(71) Rajagopalan, P.; Lutz, S.; Benkovic, S. Coupling interactions of distal residues enhance dihydrofolate reductase catalysis: mutational effects on hydride transfer rates. *Biochemistry* **2002**, *41*, 12618–12628.

(72) Sikorski, R. S.; Wang, L.; Markham, K. A.; Rajagopalan, P. T. R.; Benkovic, S. J.; Kohen, A. Tunneling and coupled motion in the E. coli dihydrofolate reductase catalysis. *J. Am. Chem. Soc.* **2004**, *126*, 4778–4779.

(73) 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. *J. Comput. Chem.* **1983**, *4*, 187–217.

(74) Brooks, B. R., III; C. L., B., Jr.; 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. *J. Comput. Chem.* **2009**, *30*, 1545–1614.

(75) Sem, D. S.; Kasper, C. B. Enzyme-substrate binding interactions of NADPH-cytochrome P-450 oxidoreductase characterized with pH and alternate substrate/inhibitor studies. *Biochemistry* **1993**, *32*, 11539–11547.

(76) Khavrutskii, I.; Price, D.; Lee, J.; Brooks, C., III. Conformational change of the methionine 20 loop of *Escherichia coli* dihydrofolate reductase modulates pKa of the bound dihydrofolate. *Protein Sci.* **2007**, *16*, 1087.

(77) Stone, D.; Paterson, S. J.; Raper, J. H.; Phillips, A. W. The amino acid sequence of dihydrofolate reductase from the mouse lymphoma L1210. *J. Biol. Chem.* **1979**, *254*, 480–488.

(78) Smith, D. R.; Calvo, J. M. Nucleotide sequence of the E. coli gene coding for dihydrofolate reductase. *Nucleic Acids Res.* **1980**, *8*, 2255–2274.

(79) Murzina, N. V.; Gudkov, A. T. Invariant amino acid replacement affects the dihydrofolate reductase function and its gene expression. *Protein Eng.* **1990**, *3*, 709–712.

(80) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Oxford University Press: Oxford, U. K., 1987; pp 156—162.

(81) MacKerell, A. D.; Bashford, D.; Bellott; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, W. E.; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiorkiewicz-Kuczera, J.; Yin, D.; Karplus, M. All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins. *J. Phys. Chem. B* **1998**, *102*, 3586–3616.

(82) MacKerell, A. D.; Feig, M.; Brooks, C. L., III. Extending the treatment of backbone energetics in protein force fields: Limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations. *J. Comput. Chem.* **2004**, *25*, 1400–1415.

(83) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **1983**, *79*, 926–935.

(84) Gao, J. Toward a Molecular Orbital Derived Empirical Potential for Liquid Simulations. *J. Phys. Chem. B* **1997**, *101*, 657–663.

(85) Sprik, M.; Klein, M. L.; Chandler, D. Staging - a sampling technique for the Monte-Carlo evaluation of path-integrals. *Phys. Rev. B: Condens. Matter* **1985**, *31*, 4234–4244.

(86) Major, D. T.; Garcia-Viloca, M.; Gao, J. L. Path integral simulations of proton transfer reactions in aqueous solution using combined QM/MM potentials. *J. Chem. Theory Comput.* **2006**, *2*, 236–245.

(87) Hwang, J. K.; Chu, Z. T.; Yadav, A.; Warshel, A. Simulations of quantum mechanical corrections for rate constants of hydride-transfer reactions in enzymes and solutions. *J. Phys. Chem.* **1991**, *95*, 8445–8448.

(88) Major, D. T.; Gao, J. L. Implementation of the bisection sampling method in path integral simulations. *J. Mol. Graphics Modell.* **2005**, *24*, 121–127.

(89) Cao, J.; Voth, G. A. A unified framework for quantum activated rate processes. I. General theory. *J. Chem. Phys.* **1996**, *105*, 6856–6870.

(90) Gillan, M. J. Quantum simulation of hydrogen in metals. *Phys. Rev. Lett.* **1987**, *58*, 563.

(91) Nam, K.; Gao, J.; York, D. M. An Efficient Linear-Scaling Ewald Method for Long-Range Electrostatic Interactions in Combined QM/MM Calculations. *J. Chem. Theory Comput.* **2005**, *1*, 2–13.

(92) Andersen, H. C. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.* **1980**, *72*, 2384–2393.

(93) Hoover, W. G. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A: At. Mol. Opt. Phys.* **1985**, *31*, 1695.

(94) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. C. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J. Comput. Phys.* **1977**, *23*, 327–341.

(95) Pu, J.; Gao, J.; Truhlar, D. G. Multidimensional Tunneling, Recrossing, and the Transmission Coefficient for Enzymatic Reactions. *Chem. Rev.* **2006**, *106*, 3140–3169.

(96) Torrie, G. M.; Valleau, J. P. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.* **1977**, *23*, 187–199.

(97) Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *J. Comput. Chem.* **1992**, *13*, 1011–1021.

(98) Bieri, J. The crystal structure of 6-methyl-7,8-dihydropterine-monohydrochloride-monohydrate. *Helv. Chim. Acta* **1977**, *60*, 2303–2308.

(99) Weber, R.; Viscontini, M. Über Pterinchemie 53. Mitteilung: Zur Konformation von 6-Methyl- und 5,6-Dimethyl-5,6,7,8-terahydropterin in wässeriger Lösung. *Helv. Chim. Acta* **1975**, *58*, 1772–1780.

(100) Poe, M.; Hoogsteen, K. 5,6,7,8-Tetrahydrofolic acid. Conformation of the tetrahydropyrazine ring. *J. Biol. Chem.* **1978**, *253*, 543–546.

(101) Ganguly, A. N.; Bieri, J. H.; Viscontini, M. Über Pterinchemie. 77. Mitteilung: Das (6*R*,*S*)-5-Formyl-6-methyl-5,6,7,8-tetrahydropterin: Synthese, chemische und physikalisch-chemische Eigenschaften. *Helv. Chim. Acta* **1981**, *64*, 367–372.

(102) Williams, T. C.; Strom, C. B. Tetrahydrobiopterin analogs: the solution conformations of 6-methyltetrahydropterin, 7-methyltetrahydropterin, and cis- and trans-6,7-dimethyltetrahydropterins as determined by proton nuclear magnetic resonance. *Biochemistry* **1985**, *24*, 458–466.

(103) Yamada, T.; Liu, X.; Englert, U.; Yamane, H.; Dronskowski, R. Solid-State Structure of Free Base Guanidine Achieved at Last. *Chem.—Eur. J.* **2009**, *15*, 5651–5655.

(104) Caminiti, R.; Pieretti, A.; Bencivenni, L.; Ramondo, F.; Sanna, N. Amidine N−C(N)−N Skeleton: Its Structure in Isolated and Hydrogen-Bonded Guanidines from ab Initio Calculations. *J. Phys. Chem.* **1996**, *100*, 10928–10935.

(105) Zhong, H.; Bowen, J. P. Theoretical study of stereoselective reduction controlled by NADH analogs. *J. Mol. Graphics Modell.* **2005**, *24*, 1–9.

(106) Cummins, P. L.; Gready, J. E. Mechanistic aspects of biological redox reactions involving NADH 1: Ab initio quantum chemical structures of the 1-methyl-nicotinamide and 1-methyl-dihydronicotinamide coenzyme analogues. *THEOCHEM* **1989**, *183*, 161–174.

(107) Wu, Y. D.; Houk, K. N. Theoretical study of conformational features of NAD$^+$ and NADH analogs: protonated nicotinamide and 1,4-dihydronicotinamide. *J. Org. Chem.* **1993**, *58*, 2043–2045.

(108) Almarsson, Ö.; Bruice, T. C. Evaluation of the factors influencing reactivity and stereospecificity in NAD(P)H dependent dehydrogenase enzymes. *J. Am. Chem. Soc.* **2002**, *115*, 2125–2138.

(109) Karle, I. The crystal structure of N-benzyl-1,4-dihydronicotinamide. *Acta Crystallogr.* **1961**, *14*, 497–502.

(110) Glasfeld, A.; Zbinden, P.; Dobler, M.; Benner, S. A.; Dunitz, J. D. Crystal structures of two simple N-substituted dihydronicotinamides: possible implications for stereoelectronic arguments in enzymology. *J. Am. Chem. Soc.* **1988**, *110*, 5152–5157.

■ NOTE ADDED AFTER ASAP PUBLICATION

This article was published ASAP on September 14, 2011. A change has been made to the caption of Figure 4. The correct version was published on September 21, 2011.

Supporting Information for Hybrid Quantum and Classical Simulations of the Dihydrofolate Reductase Catalyzed Hydride Transfer Reaction on an Accurate Semi-Empirical Potential Energy Surface

Dvir Doron,[1] Dan Thomas Major[*1], Amnon Kohen,[2] Walter Thiel,[3] and Xin Wu[3]

[1] *Department of Chemistry and the Lise Meitner-Minerva Center of Computational Quantum Chemistry, Bar-Ilan University, Ramat-Gan 52900, Israel*

[2] *Department of Chemistry, University of Iowa, Iowa City, Iowa 52242*

[3] *Max-Planck-Institut für Kohlenforschung, Kaiser-Wilhelm-Platz 1, D-45470 Mülheim an der Ruhr, Germany*

[*] Electronic mail: Dan-Thomas.Major@biu.ac.il

[1] *Bar-Ilan University*

[2] *University of Iowa*

[3] *Max-Planck-Institut für Kohlenforschung*

Received:

Chart S1. The ligands involved in the redox reaction catalyzed by DHFR, and the analogous models that were subjected to gas-phase calculations in this study. The letters in parentheses (**A**, **B**, **C**, **D**) denote the groups according to which the compounds were classified and are discussed throughout this text.

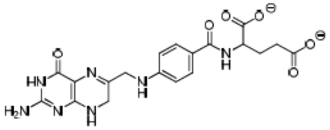
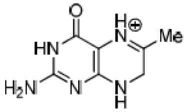
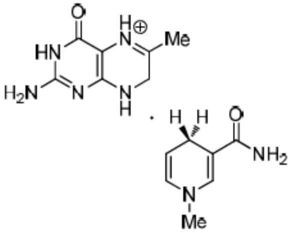
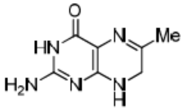
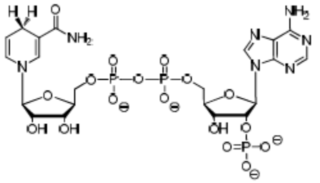
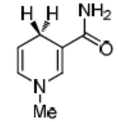
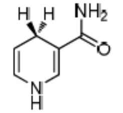
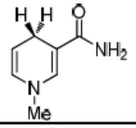
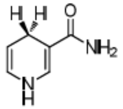
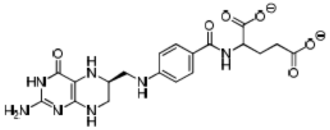
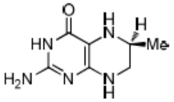
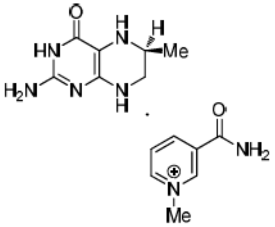
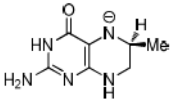
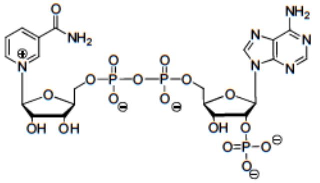
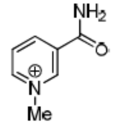
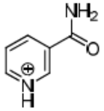
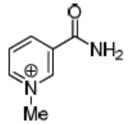
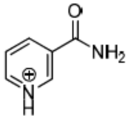
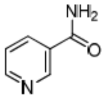
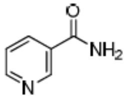
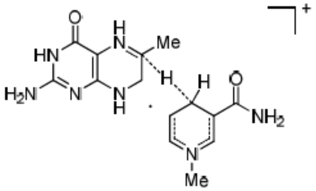
	Original Ligands		**Analogue Models**					
	Name (Label)	**Structure**	**Label (Group)**	**Structure**	**Label (Group)**	**Structure**	**Label (Group)**	**Structure**
Reactants	7,8-dihydrofolate (**H_2folate**)		**6-Me-H_3pterin$^+$** **(A)**				RS complex **[6-Me-H_3pterin$^+$ · *trans*-Me-H_2nic]** **(D)**	
			6-Me-H_2pterin **(A)**					
	Nicotinamide Adenine Dinucleotide Phosphate Hydride (**NADPH**)		***cis*-Me-H_2nic** **(B)**		***cis*-H_2nic** **(C)**			
			***trans*-Me-H_2nic** **(B)**		***trans*-H_2nic** **(C)**			
Products	5,6,7,8-tetrahydrofolate (**H_4folate**)		**6-Me-H_4pterin** **(A)**				PS complex **[6-Me-H_4pterin · *trans*-Me-Hnic$^+$]** **(D)**	
			6-Me-H_3pterin$^-$ **(A)**					
	Nicotinamide Adenine Dinucleotide Phosphate (**NADP$^+$**)		***cis*-Me-Hnic$^+$** **(B)**		***cis*-Hnic$^+$** **(C)**			
			***trans*-Me-Hnic$^+$** **(B)**		***trans*-Hnic$^+$** **(C)**			
Misc.	-	-	***cis*-nic** **(C)**		***trans*-nic** **(C)**		TS complex **(D)**	

Table S1. Key computational studies on the DHFR-catalyzed hydride transfer reaction in the past decade.

Ref	Year of publication	PES Description	Nuclear Quantum Effects	Reaction coordinate	Solvation model
1	2002	QM(PM3)/MM (with DFT-level correction)	-	Distance between the hydride to acceptor carbon, *r*(H4-C6)	SBC
2	2002	A parametrized two-state EVB model	Incorporated by representing the transferring hydrogen nucleus as a 3-D vibrational wavefunction	The energy gap between the reactant and product diabatic states	Truncated octahedral PBC
3	2003	QM(PM3)/MM	-	The distance between the donor and acceptor atoms (DAD), *r*(C4-C6)	Truncated octahedral PBC
4	2003	QM(AM1-SVB)/MM	Incorporated by ensemble-averaged variational transition state theory with multidimensional tunneling (EA-VTST/MT), based on optimized tunneling paths	Difference between the hydrogen-to-donor-carbon distance and the hydrogen-to-acceptor-carbon distance, *r*(H4-C4) - *r*(H4-C6)	Cubic PBC
5	2005	A parametrized two-state EVB model	Incorporated by representing the transferring hydrogen nucleus as a 3-D vibrational wavefunction	The difference between the energies of two valence bond states averaged over the ground state vibrational wavefunction of the transferring hydrogen	Truncated octahedral PBC
6	2005	QM(PM3-SVB)/MM		Difference between the hydrogen-to-acceptor-carbon distance and the hydrogen-to-donor-carbon distance, *r*(H4-C6) - *r*(H4-C4)	SBC
7	2007	EVB-QCP	Quantized Classical Path (QCP)	The energy gap between the reactant and product diabatic states	Surface constraint all-atom solvent (SCAAS) model
8	2010	A parametrized two-state EVB model		The energy gap between the reactant and product diabatic states	PBC

Geometry Optimizations and Reaction Energetics in the Gas Phase. All structures (with the exception of the reactant, transition and product state complexes) were initially set up using the builder tool in the GaussView 4.1.2 program.[9] These geometries were thereafter fully optimized by various methods using the Gaussian03[10] and Gaussian09[11] programs. The vibrational normal modes of the molecules at their stationary points on the potential energy surface (PES) were calculated using the same basis set employed in the optimization, to verify true local minima (*i.e.* all frequencies are real). The vibrational frequency scale factors are summarized in Table S2.

Table S2. Frequency scaling factors for fundamental vibrations for the various levels of theory used in the calculation of the thermodynamic properties of the models.

Method	**Scaling factor**
AM1	0.9532[12]
B3LYP/6-31+G(d,p)	0.9648[13]
PBE1PBE/6-31+G(d,p)	0.9547[13]
PBEPBE/6-31+G(d,p)	0.9904[13]
B98/6-31+G(d,p)	0.9635[13]
BB1K/6-31+G(d,p)	0.9359[13]
MPWB1K/6-31+G(d,p)	0.9335[13]
M06/6-31+G(d,p)	0.950[14]
G4MP2	0.9854[15]
CBS-QB3	0.9900[16]

Calibration of the AM1 Hamiltonian. The general approach for the re-parametrization of the AM1 method[17] has been outlined in the main paper (see the section on SRP Development). Here we provide further details.

All target species were grouped into four categories: pterins (**A**, 4 molecules), *N*-methyl nicotinamides (**B**, 4 molecules), unsubstituted and neutral nicotinamides (**C**, 6 molecules) and the reactant, transition and product state complexes (**D**, 3 molecules). For each of the first three categories **A**-**C**, an internal reference molecule was determined (6-Me-H_3pterin$^+$, *cis*-Me-H_2nic and *trans*-nic, respectively) and its reference energy was chosen as its enthalpy of formation in the gaseous state at 298 K. The heats of formation for 6-Me-H_3pterin$^+$ and *cis*-Me-H_2nic were calculated at the G4MP2 level of theory, according to the procedure described by Ochterski;[18] whereas the entry for the *transoid* conformer of neutral nicotinamide (*trans*-nic) was set to the corresponding experimental enthalpy of formation, -6.9 kcal/mol.[19] All other species in groups **A**-**C** were energetically interrelated to the reference molecule in their group by addressing different types of thermodynamic relations: (a) conformational rearrangements (groups **B** and **C**); (b) proton affinities, *i.e.* protonated vs. unprotonated species (groups **A** and **C**); (c) hydride affinities (groups **A**,**B**,**C**). According to this scheme, the target values for the reference molecules are actual heats of formation. The target values for all other molecules represent the electronic energy difference relative to the reference molecule of their group. In this approach, we obtain an accurate electronic energy surface wherein the absolute energies of the reference molecules are their actual heats of formation (obtained from experiment or computation) whereas the absolute energies of the remaining molecules are similar to their heats of formation, but without thermal and zero-point energy effects.

In reactions of type (a), the entries of the nicotinamides in question were simply determined by adding or subtracting the *cisoid-transoid* electronic energy difference relative to the reference molecule. In reactions of type (b), we considered the proton affinity (PA) of an unprotonated species R:

$$PA^{\mathrm{SE}}(\mathrm{R}) = \Delta H_f^{\mathrm{SE}}(\mathrm{R}) + \Delta H_f^{\mathrm{SE}}(\mathrm{H}^+) - \Delta H_f^{\mathrm{SE}}(\mathrm{RH}^+) \quad (1a)$$

$$PA^{\mathrm{AI}}(\mathrm{R}) = H^{\mathrm{AI}}(\mathrm{R}) + H^{\mathrm{AI}}(\mathrm{H}^+) - H^{\mathrm{AI}}(\mathrm{RH}^+) \quad (1b)$$

where Eq. 1a represents PA in terms of heats of formation, which applies to data obtained from semiempirical (SE) calculations, while Eq. 2a expresses the PA in the sense of absolute total enthalpies, which is the kind of values resulting from *ab-initio* (AI) and DFT calculations. The goal of the SRP development of the SE method is to correctly reproduce the PA from AI calculations. Therefore Eq. 1a and 1b should be equal, that is

$$\Delta H_f^{\mathrm{SE}}(\mathrm{R}) + \Delta H_f^{\mathrm{SE}}(\mathrm{H}^+) - \Delta H_f^{\mathrm{SE}}(\mathrm{RH}^+) = H^{\mathrm{AI}}(\mathrm{R}) + H^{\mathrm{AI}}(\mathrm{H}^+) - H^{\mathrm{AI}}(\mathrm{RH}^+) \quad (2)$$

A slight rearrangement of Eq. 2 yields the explicit expression for the difference between the heats of formation for the unprotonated and protonated species:

$$\Delta H_f^{\mathrm{SE}}(\mathrm{R}) - \Delta H_f^{\mathrm{SE}}(\mathrm{RH}^+) = H^{\mathrm{AI}}(\mathrm{R}) + H^{\mathrm{AI}}(\mathrm{H}^+) - H^{\mathrm{AI}}(\mathrm{RH}^+) - \Delta H_f^{\mathrm{SE}}(\mathrm{H}^+) \tag{3}$$

The semi-empirical energy is in essence described in terms of "heats of formation" computed by the semi-empirical code on the one hand, but the result is required to reproduce electronic energies on the other hand, excluding any thermal contribution. We shall therefore impose the following equality requirement:

$$\Delta H_f^{\mathrm{SE}}(\mathrm{R}) - \Delta H_f^{\mathrm{SE}}(\mathrm{RH}^+) \leftrightarrow E_{el}^{\mathrm{AI}}(\mathrm{R}) + E_{el}^{\mathrm{AI}}(\mathrm{H}^+) - E_{el}^{\mathrm{AI}}(\mathrm{RH}^+) - \Delta H_f^{\mathrm{SE}}(\mathrm{H}^+) \tag{4}$$

where each E_{el}^{AI} term represents the electronic energy contribution to the corresponding total enthalpy term H^{AI}. By definition, $E_{el}^{\mathrm{AI}}(\mathrm{H}^+) = 0$, and indicating the source of each term in the equation, we can rewrite Eq. 4 as

$$E^{\mathrm{SRP}}(\mathrm{R}) - E^{\mathrm{SRP}}(\mathrm{RH}^+) = E_{el}^{\mathrm{G4MP2}}(\mathrm{R}) - E_{el}^{\mathrm{G4MP2}}(\mathrm{RH}^+) - \Delta H_f^{\mathrm{exp}}(\mathrm{H}^+) \tag{5}$$

where E^{SRP} stands for the target semi-empirical energy (analogous to the "heat of formation", ΔH_f^{SE} in standard semi-empirical terms), E_{el}^{G4MP2} is the electronic energy calculated at the G4MP2 level of theory, and $\Delta H_f^{\mathrm{exp}}(\mathrm{H}^+)$ is the experimental heat of formation for the proton, 365.7 kcal/mol.[20]

In reactions of type (c), a development analogous to (b) was applied, starting from two possible definitions of the hydride affinity (HA) of species R, readily obtained from the enthalpies of formation or the absolute total enthalpies of RH$^-$ and R:

$$HA^{\mathrm{SE}}(\mathrm{R}) = \Delta H_f^{\mathrm{SE}}(\mathrm{R}) + \Delta H_f^{\mathrm{SE}}(\mathrm{H}^-) - \Delta H_f^{\mathrm{SE}}(\mathrm{RH}^-) \tag{6a}$$

$$HA^{\mathrm{AI}}(\mathrm{R}) = H^{\mathrm{AI}}(\mathrm{R}) + H^{\mathrm{AI}}(\mathrm{H}^-) - H^{\mathrm{AI}}(\mathrm{RH}^-) \tag{6b}$$

Equating Eqs. 6a and 6b, followed by rearrangement gives

$$\Delta H_f^{\mathrm{SE}}(\mathrm{R}) - \Delta H_f^{\mathrm{SE}}(\mathrm{RH}^-) = H^{\mathrm{AI}}(\mathrm{R}) + H^{\mathrm{AI}}(\mathrm{H}^-) - H^{\mathrm{AI}}(\mathrm{RH}^-) - \Delta H_f^{\mathrm{SE}}(\mathrm{H}^-) \tag{7}$$

which can be converted into the corresponding "electronic energy formulation" as

$$\Delta H_f^{\mathrm{SE}}(\mathrm{R}) - \Delta H_f^{\mathrm{SE}}(\mathrm{RH}^-) \leftrightarrow E_{el}^{\mathrm{AI}}(\mathrm{R}) + E_{el}^{\mathrm{AI}}(\mathrm{H}^-) - E_{el}^{\mathrm{AI}}(\mathrm{RH}^-) - \Delta H_f^{\mathrm{SE}}(\mathrm{H}^-) \qquad (8)$$

Or alternatively:

$$E^{\mathrm{SRP}}(\mathrm{R}) - E^{\mathrm{SRP}}(\mathrm{RH}^-) = E_{el}^{\mathrm{G4MP2}}(\mathrm{R}) + E_{el}^{\mathrm{G4MP2}}(\mathrm{H}^-) - E_{el}^{\mathrm{G4MP2}}(\mathrm{RH}^-) - \Delta H_f^{\mathrm{exp}}(\mathrm{H}^-) \qquad (9)$$

Here not only the electronic energies of the coupled species are calculated with G4MP2, but also the energy of a hydride ion $E_{el}^{\mathrm{G4MP2}}(\mathrm{H}^-)$, while the experimental enthalpy of formation for hydride $\Delta H_f^{\mathrm{exp}}(\mathrm{H}^-)$ is 34.7 kcal/mol.[21]

Table S3 lists the relative target energies, E^{SRP}, and the associated molecular dipole moments for neutral species, μ^{SRP} (dipole moments for charged species were not considered as they are dependent upon the choice of the origin of the coordinate system) calculated with G4MP2. Other properties included in the optimization procedure were the geometries of the model structures, Mulliken charges, vibrational frequencies attributed to C-H stretches (at the G4MP2 level), and the transition state imaginary frequencies (at the B3LYP/6-31+G(d,p) or B3LYP-D/6-31+G(d,p) level).

Table S3. Target values for the semiempirical models. E^{SRP} and μ^{SRP} are the target energy and dipole moment

Group		Compound Label	E^{SRP} (kcal/mol)	μ^{SRP} (D)
A	**Pterins**	6-Me-H_3pterin$^+$	136.4[a]	-
		6-Me-H_2pterin	14.7	6.3508
		6-Me-H_4pterin	-33.4	4.2581
		6-Me-H_3pterin$^-$	-19.6	-
B	***Methyl Nicotinamides***	*cis*-Me-H_2nic	-11.6[a]	4.0039
		cis-Me-Hnic$^+$	147.6	-
		trans-Me-H_2nic	-10.3	5.5978
		trans-Me-Hnic$^+$	149.5	-
C	***Unsubstituted Nicotinamides***	*cis*-H_2nic	-33.7	3.9881
		cis-Hnic$^+$	131.8	-
		trans-H_2nic	-32.4	5.2943
		trans-Hnic$^+$	133.4	-
	Neutral Nicotinamides	*cis*-nic	-6.1	5.3038
		trans-nic	-6.9[b]	2.0730

[a] Internal reference, calculated as the enthalpy of formation in the gaseous state at 298 K at the G4MP2 level.
[b] Internal reference, the experimental enthalpy of formation of nicotinamide in the gaseous state at 298 K.[19]

The individual energies were used to construct a set of reaction energies as target values *per se* to be weighted in the fitness function, using the general formula

$$\Delta E_r^{\mathrm{SRP}} = \sum_{\text{products}} E^{\mathrm{SRP}} - \sum_{\text{reactants}} E^{\mathrm{SRP}} . \qquad (10)$$

Other entries included were the relative energies between the reactant, transition and product state complexes (group **D**) obtained by geometry optimization and IRC calculations (*i.e.* reaction energy and barrier), computed either at the B3LYP/6-31+G(d,p) or B3LYP-D/6-31+G(d,p) level of theory.

Figure S1. The gradient norm of the product complex during AM1-SRP(D) geometry optimization. Note the logarithmic scale for the gradient norm.

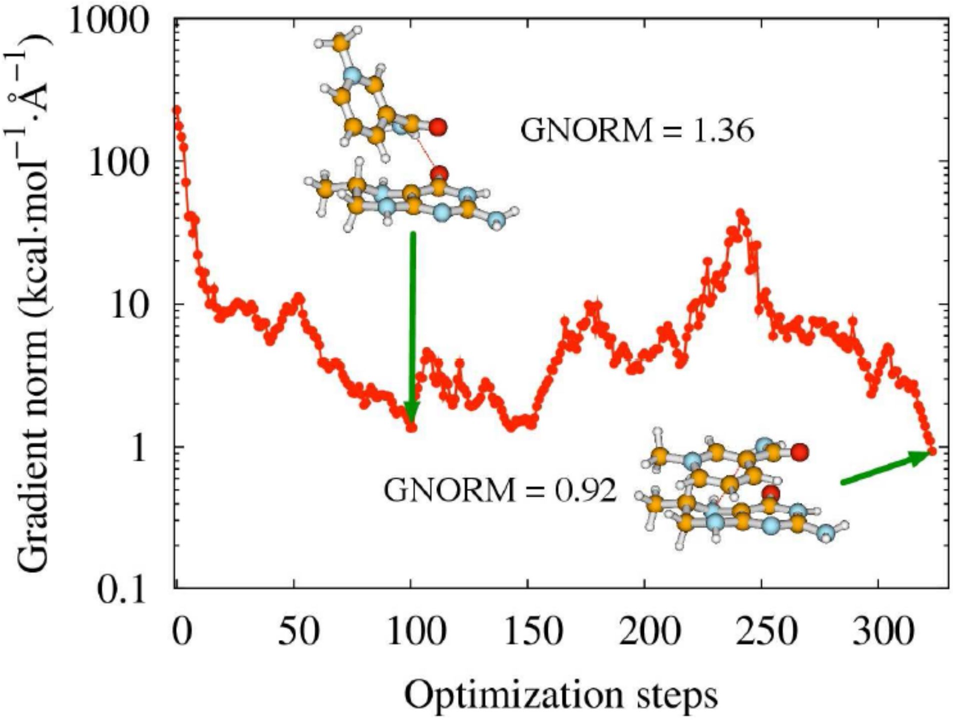


Table S4. Selected unscaled vibrational frequencies (as wavenumbers) calculated with different levels of theory: (i) G4MP2; (ii) AM1; (iii) AM1-SRP; (iv) AM1-SRP(D). The RMSD values with respect to G4MP2 are also given.

	level	$\tilde{\nu}$ (cm^{-1})											RMSD
6-Me-H$_3$pterin$^+$	i	3738.35	3618.35	3611.51	3589.16	3457.38	3139.44	3085.58	3071.31	3030.89	2945.80		-
	ii	3484.82	3447.09	3419.35	3407.35	3280.76	3130.77	3039.88	3023.25	2997.18	2924.64		141.72
	iii	3552.90	3464.22	3460.46	3454.76	3370.56	3148.63	3094.10	3062.96	3053.80	3037.22		107.55
	iv	3635.34	3600.97	3590.29	3585.68	3483.51	3140.78	3093.48	3070.07	3049.19	3001.59		39.46
6-Me-H$_2$pterin	i	3697.79	3630.28	3603.32	3584.14	3147.07	3055.03	3031.35	3009.93	2904.62			-
	ii	3493.03	3485.01	3462.41	3450.69	3148.65	3059.55	3050.76	3015.67	2941.31			106.71
	iii	3624.35	3574.87	3542.82	3537.91	3177.45	3117.92	3107.60	3092.42	3064.78			79.87
	iv	3631.48	3628.42	3622.20	3603.19	315..67	3098.71	3074.24	3053.42	2992.00			45.26
6-Me-H$_4$pterin	i	3667.09	3641.22	3608.17	3560.67	3547.89	3114.99	3104.93	3065.88	3036.36	2988.60	2934.60	-
	ii	3472.76	3458.10	3454.61	3438.33	3373.99	3158.93	3068.83	3065.03	3037.56	2987.50	2968.06	114.59
	iii	3629.84	3625.83	3587.57	3545.05	3543.20	3189.34	3121.53	3117.52	3108.70	3085.01	3028.13	55.65
	iv	3632.34	3615.64	3612.14	3596.33	3531.79	3153.21	3095.08	3091.86	3069.26	3048.03	3018.83	39.72
6-Me-H$_3$pterin$^-$	i	3604.17	3553.42	3483.45	3395.38	3092.39	3063.56	3019.30	2997.13	2981.09	2770.53		-
	ii	3497.95	3438.26	3386.31	3365.41	3157.37	3070.01	3062.85	3038.15	2963.56	2941.01		84.91
	iii	3703.90	3689.17	3631.90	3614.31	3186.05	3122.34	3094.23	3079.46	3038.85	3014.21		136.26
	iv	3655.33	3584.29	3579.27	3576.75	3145.66	3089.93	3051.53	3036.53	2987.87	2968.23		95.34
***cis*-Me-H$_2$nic**	i	3720.42	3592.34	3208.56	3204.90	3183.35	3131.21	3075.29	3005.48	2951.84	2909.83		-
	ii	3559.99	3534.75	3201.47	3150.47	3109.47	3100.59	3054.45	3019.30	2996.92	2974.14		67.27
	iii	3690.03	3634.17	3215.31	3199.79	3175.77	3162.96	3125.97	3122.36	3092.05	3031.00		73.74
	iv	3692.53	3673.87	3236.45	3205.99	3177.19	3114.21	3073.26	3071.60	3040.03	3002.14		54.04
***trans*-Me-H$_2$nic**	i	3686.76	3570.08	3204.30	3189.38	3181.53	3119.96	3075.54	2999.19	2997.91	2982.81		-
	ii	3551.76	3523.47	3199.55	3150.23	3133.18	3102.79	3065.01	3021.67	2997.29	2983.48		50.20
	iii	3705.79	3631.69	3215.42	3199.10	3188.87	3162.78	3124.37	3121.98	3100.12	3039.44		61.14
	iv	3690.55	3665.06	3235.44	3205.30	3184.94	3114.79	3087.02	3073.16	3038.97	3021.44		43.68
***cis*-Me-Hnic$^+$**	i	3722.35	3592.17	3237.38	3234.06	3225.05	3205.64	3189.41	3162.55	3076.58			-
	ii	3539.74	3472.43	3155.16	3142.86	3095.56	3086.03	3042.87	3003.03	2997.13			127.83
	iii	3661.24	3533.91	3164.85	3155.88	3148.81	3143.10	3134.67	3129.59	3103.48			60.58
	iv	3693.99	3628.22	3216.24	3201.18	3175.95	3126.42	3116.44	3085.99	3080.01			51.12
***trans*-Me-Hnic$^+$**	i	3700.85	3576.86	3237.37	3227.18	3226.17	3216.16	3183.77	3164.90	3075.69			-
	ii	3535.48	3468.85	3150.56	3122.40	3093.56	3087.55	3086.49	3006.52	2994.20			121.49
	iii	3667.12	3535.77	3160.80	3156.86	3141.75	3136.61	3134.30	3130.22	3119.35			60.23
	iv	3680.23	3618.23	3214.32	3174.80	3170.38	3165.21	3116.87	3086.12	3078.76			49.23
***cis*-H$_2$nic**	i	3723.42	3691.56	3594.47	3217.52	3213.20	3187.84	2954.39	2910.81				-
	ii	3559.45	3534.64	3511.48	3203.81	3153.78	3113.62	3053.95	2973.60				100.92
	iii	3689.62	3633.54	3622.40	3223.00	3204.45	3189.77	3091.61	3030.94				69.52
	iv	3691.73	3673.26	3640.98	3238.20	3209.68	3180.91	3071.53	3001.89				57.02
***trans*-H$_2$nic**	i	3688.66	3668.38	3571.18	3211.81	3197.39	3185.66	3017.43	2969.46				-
	ii	3551.17	3522.83	3462.02	3201.28	3152.73	3136.65	3065.35	2984.04				85.89
	iii	3706.59	3631.62	3629.01	3223.67	3203.57	3205.15	3099.71	3039.33				46.31
	iv	3689.77	3664.41	3623.03	3237.19	3209.15	3190.35	3087.03	3021.40				37.17
***cis*-Hnic$^+$**	i	3720.53	3590.34	3558.49	3244.02	3235.53	3229.91	3204.29					-
	ii	3538.43	3469.20	3329.46	3151.95	3138.68	3088.45	3037.61					153.97
	iii	3659.20	3529.70	3402.04	3161.56	3143.72	3141.91	3107.42					95.82
	iv	3692.83	3625.15	3531.69	3213.52	3198.45	3173.26	3125.25					45.45
***trans*-Hnic$^+$**	i	3701.07	3577.01	3562.99	3241.41	3231.47	3229.56	3212.67					-
	ii	3534.69	3465.96	3335.79	3147.74	3115.77	3087.79	3082.39					146.81
	iii	3664.75	3530.87	3411.88	3157.96	3142.60	3139.63	3113.07					91.93
	iv	3679.28	3615.80	3537.88	3211.73	3171.53	3169.73	3166.04					42.83
***cis*-nic**	i	3714.06	3587.68	3201.55	3188.06	3178.32	3156.86						-
	ii	3554.96	3524.08	3206.45	3184.76	3145.52	3126.95						72.30
	iii	3696.50	3618.84	3220.60	3200.90	3188.93	3184.18						21.08
	iv	3706.07	3673.44	3249.08	3219.23	3199.69	3180.35						44.08
***trans*-nic**	i	3723.14	3594.68	3213.24	3195.05	3158.12	3145.66						-
	ii	3553.03	3524.49	3204.83	3173.20	3146.25	3142.00						75.90
	iii	3691.34	3617.66	3218.92	3203.83	3199.61	3171.56						25.95
	iv	3704.18	3673.96	3249.02	3209.97	3199.04	3195.61						45.31

Table S5. Comparison of the semi-empirical parameters of the canonical AM1 Hamiltonian vs. the re-parametrized schemes AM1-SRP and AM1-SRP(D).

Hydrogen					
Parameter (units)	AM1	AM1-SRP	Δ	AM1-SRP(D)	Δ
U_{ss} (eV)	-11.39642700	-10.68984024	-0.70658676	-11.43733166	0.04090466
G_{ss} (eV)	12.84800000	12.84709466	0.00090534	12.60364621	0.24435379
α (Å^{-1})	2.88232400	2.98632722	-0.10400322	3.40130304	-0.51897904
β_s (eV)	-6.17378700	-6.33105768	0.15727068	-5.64837035	-0.52541665
ζ_s (a.u.)	1.18807800	1.24851395	-0.06043595	1.17345275	0.01462525
K_1 (eV)	0.12279600	0.10477938	0.01801662	0.13056000	-0.00776400
L_1 (1/Å^2)	5.00000000	5.11512463	-0.11512463	4.80257197	0.19742803
M_1 (Å)	1.20000000	1.15101043	0.04898957	1.21380391	-0.01380391
K_2 (eV)	0.00509000	-0.01289865	0.01798865	0.00888975	-0.00379975
L_2 (1/Å^2)	5.00000000	5.18465935	-0.18465935	14.28487074	-9.28487074
M_2 (Å)	1.80000000	1.75423483	0.04576517	1.75118838	0.04881162
K_3 (eV)	-0.01833600	-0.01178407	-0.00655193	-0.03032460	0.01198860
L_3 (1/Å^2)	2.00000000	1.98528456	0.01471544	1.60610364	0.39389636
M_3 (Å)	2.10000000	2.19234758	-0.09234758	2.01192128	0.08807872

Carbon					
Parameter (units)	AM1	AM1-SRP	Δ	AM1-SRP(D)	Δ
U_{ss} (eV)	-52.02865800	-52.24737422	0.21871622	-52.12801676	0.09935876
U_{pp} (eV)	-39.61423900	-38.46838755	-1.14585145	-38.86576821	-0.74847079
G_{ss} (eV)	12.23000000	12.17894654	0.05105346	12.05774032	0.17225968
G_{sp} (eV)	11.47000000	11.42197994	0.04802006	11.71252941	-0.24252941
G_{pp} (eV)	11.08000000	11.28414421	-0.20414421	11.13199843	-0.05199843
$G_{pp'}$ (eV)	9.84000000	9.87970858	-0.03970858	9.86125443	-0.02125443
H_{sp} (eV)	2.43000000	2.40762387	0.02237613	2.40118582	0.02881418
α (Å^{-1})	2.64827400	2.64566311	0.00261089	2.70195063	-0.05367663
β_s (eV)	-15.71578300	-15.72667726	0.01089426	-15.85249007	0.13670707
β_p (eV)	-7.71928300	-7.80162672	0.08234372	-7.41730443	-0.30197857
ζ_s (a.u.)	1.80866500	1.83025672	-0.02159172	2.02999558	-0.22133058
ζ_p (a.u.)	1.68511600	1.69607884	-0.01096284	1.61950090	0.06561510
K_1 (eV)	0.01135500	0.00301604	0.00833896	0.03207184	-0.02071684
L_1 (1/Å^2)	5.00000000	4.94955509	0.05044491	4.61337571	0.38662429
M_1 (Å)	1.60000000	1.58636330	0.01363670	1.70743023	-0.10743023
K_2 (eV)	0.04592400	0.05629335	-0.01036935	0.02309988	0.02282412
L_2 (1/Å^2)	5.00000000	5.04827121	-0.04827121	5.87793752	-0.87793752
M_2 (Å)	1.85000000	1.87582178	-0.02582178	1.83819070	0.01180930
K_3 (eV)	-0.02006100	-0.02474245	0.00468145	-0.01421778	-0.00584322
L_3 (1/Å^2)	5.00000000	5.00959715	-0.00959715	3.84436254	1.15563746
M_3 (Å)	2.05000000	2.02202225	0.02797775	1.95422526	0.09577474
K_4 (eV)	-0.00126000	-0.00722588	0.00596588	-0.00219237	0.00093237
L_4 (1/Å^2)	5.00000000	5.13550628	-0.13550628	1.03264951	3.96735049
M_4 (Å)	2.65000000	2.68246146	-0.03246146	2.75960465	-0.10960465

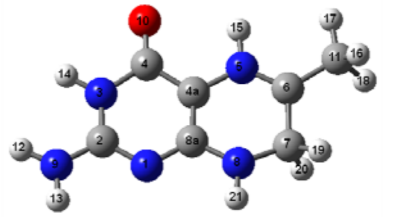
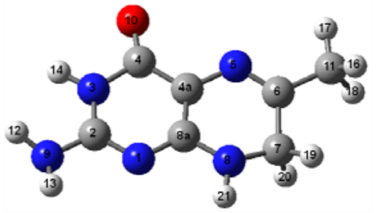
Nitrogen					
Parameter (units)	AM1	AM1-SRP	Δ	AM1-SRP(D)	Δ
U_{ss} (eV)	-71.86000000	-72.82069585	0.96069585	-72.01619964	0.15619964
U_{pp} (eV)	-57.16758100	-59.49215630	2.32457530	-57.55693452	0.38935352
G_{ss} (eV)	13.59000000	13.65461868	-0.06461868	13.54840912	0.04159088

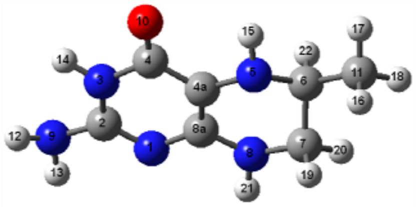
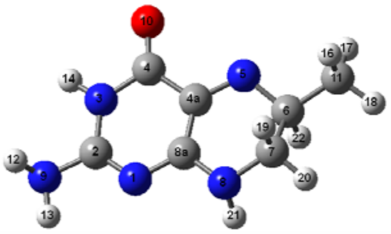
G_{sp} (eV)	12.66000000	12.79898733	-0.13898733	12.54471362	0.11528638
G_{pp} (eV)	12.98000000	12.88438212	0.09561788	13.04037236	-0.06037236
$G_{pp'}$ (eV)	11.59000000	11.47612859	0.11387141	11.66657487	-0.07657487
H_{sp} (eV)	3.14000000	3.18707623	-0.04707623	3.15191830	-0.01191830
α (Å^{-1})	2.94728600	3.11091376	-0.16362776	2.86730265	0.07998335
β_s (eV)	-20.29911000	-19.14256652	-1.15654348	-20.15951726	-0.13959274
β_p (eV)	-18.23866600	-19.28044417	1.04177817	-18.30236349	0.06369749
ζ_s (a.u.)	2.31541000	2.45754818	-0.14213818	2.36751669	-0.05210669
ζ_p (a.u.)	2.15794000	2.29720493	-0.13926493	2.14340234	0.01453766
K_1 (eV)	0.02525100	0.01584487	0.00940613	0.02577954	-0.00052854
L_1 (1/Å^2)	5.00000000	5.01639898	-0.01639898	4.93484582	0.06515418
M_1 (Å)	1.50000000	1.40990729	0.09009271	1.50396975	-0.00396975
K_2 (eV)	0.02895300	0.02413205	0.00482095	0.02888160	0.00007140
L_2 (1/Å^2)	5.00000000	4.95264284	0.04735716	5.03773737	-0.03773737
M_2 (Å)	2.10000000	2.08955372	0.01044628	2.10230179	-0.00230179
K_3 (eV)	-0.00580600	0.00231665	-0.00812265	-0.00608690	0.00028090
L_3 (1/Å^2)	2.00000000	2.02251051	-0.02251051	1.57427110	0.42572890
M_3 (Å)	2.40000000	2.53289116	-0.13289116	2.43317594	-0.03317594

			Oxygen		
Parameter (units)	AM1	AM1-SRP	Δ	AM1-SRP(D)	Δ
U_{ss} (eV)	-97.83000000	-97.18740347	-0.64259653	-97.82978333	-0.00021667
U_{pp} (eV)	-78.26238000	-78.15419188	-0.10818812	-78.26194049	-0.00043951
G_{ss} (eV)	15.42000000	15.58420000	-0.16420000	15.41584115	0.00415885
G_{sp} (eV)	14.48000000	14.53118414	-0.05118414	14.48088177	-0.00088177
G_{pp} (eV)	14.52000000	14.52057396	-0.00057396	14.51015532	0.00984468
$G_{pp'}$ (eV)	12.98000000	13.09474870	-0.11474870	12.98948394	-0.00948394
H_{sp} (eV)	3.94000000	3.98430700	-0.04430700	3.80637580	0.13362420
α (Å^{-1})	4.45537100	4.45864569	-0.00327469	4.51366550	-0.05829450
β_s (eV)	-29.27277300	-29.03471808	-0.23805492	-29.27065291	-0.00212009
β_p (eV)	-29.27277300	-28.97637174	-0.29640126	-29.28300296	0.01022996
ζ_s (a.u.)	3.10803200	3.11536491	-0.00733291	3.01595571	0.09207629
ζ_p (a.u.)	2.52403900	2.49978900	0.02425000	2.38139341	0.14264559
K_1 (eV)	0.28096200	0.28934086	-0.00837886	0.28091667	0.00004533
L_1 (1/Å^2)	5.00000000	4.98540050	0.01459950	5.00000144	-0.00000144
M_1 (Å)	0.84791800	0.83276255	0.01515545	0.84783749	0.00008051
K_2 (eV)	0.08143000	0.07785336	0.00357664	0.08133050	0.00009950
L_2 (1/Å^2)	7.00000000	7.01317050	-0.01317050	7.00009001	-0.00009001
M_2 (Å)	1.44507100	1.42486880	0.02020220	1.44514698	-0.00007598

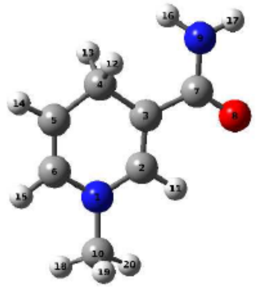
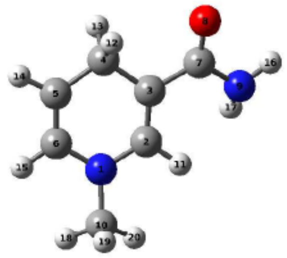
Table S6. Cartesian coordinates of the molecular models optimized geometries at the target levels of theory, calculated with B3LYP/6-31G(2df, p) (individual molecules), B3LYP/6-31+G(d,p) or B3LYP-D/6-31+G(d,p)) (bimolecular complexes).

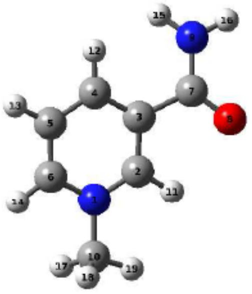
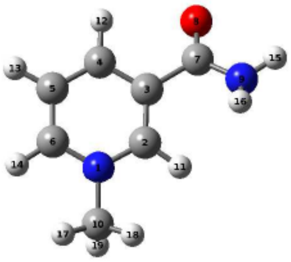
Group A

	6-Me-H_3pterin$^+$			**6-Me-H_2pterin**		
						
Atom	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	1.4545190	-1.3496800	-0.0485110	1.4002650	-1.3505480	-0.0689770
C(2)	2.4102640	-0.4282010	-0.0219930	2.3675560	-0.4697350	-0.0209990
N(3)	2.1631920	0.9130410	0.0364140	2.1524010	0.8639100	0.0485900
C(4)	0.8613050	1.4653150	0.0500000	0.8563910	1.4851510	0.0452790
C(4a)	-0.1351230	0.4457600	0.0629390	-0.1963050	0.5091620	0.0025960
N(5)	-1.4682590	0.8251900	-0.0247830	-1.5218310	0.9332070	-0.1001300
C(6)	-2.4792700	0.0133790	-0.0068400	-2.4645510	0.0759180	-0.0144710
C(7)	-2.2009660	-1.4546960	0.2239790	-2.2270880	-1.4027840	0.2759900
N(8)	-0.8198260	-1.8025960	-0.0913200	-0.8672580	-1.7578100	-0.0885430
C(8a)	0.1908820	-0.9086240	-0.0087520	0.1359650	-0.8424620	-0.0302050
N(9)	3.6830690	-0.8388020	-0.0528620	3.6661510	-0.9112140	0.0134860
O(10)	0.6626200	2.6645450	0.0400310	0.8025980	2.6982270	0.0844360
C(11)	-3.8766560	0.4954790	-0.1407190	-3.8955040	0.5053480	-0.1442440
H(12)	4.4664000	-0.2089370	-0.0256300	4.3801140	-0.3070630	-0.3624610
H(13)	3.8568270	-1.8300880	-0.0953350	3.7638260	-1.8845090	-0.2328930
H(14)	2.9168120	1.5890150	0.0333980	2.9182400	1.5184480	0.1284070
H(15)	-1.6011540	1.8349100	-0.1213150	-	-	-
H(16)	-4.3327190	0.0827950	-1.0490970	-4.3780780	0.0219830	-1.0039490
H(17)	-3.9336960	1.5850800	-0.1829390	-3.9479770	1.5877780	-0.2660450
H(18)	-4.4808130	0.1408010	0.7032920	-4.4748680	0.2144020	0.7429590
H(19)	-2.8795250	-2.0474660	-0.3982590	-2.9304960	-2.0194220	-0.2970710
H(20)	-2.4594770	-1.6843970	1.2735620	-2.4396570	-1.6011960	1.3444230
H(21)	-0.5650890	-2.7786200	-0.1421740	-0.5787820	-2.7226270	-0.0435150

	6-Me-H_4pterin			**6-Me-H_3pterin$^-$**		
						
Atom	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	1.4741170	-1.3427950	0.0114120	-1.4438670	-1.3327070	-0.0256140
C(2)	2.4261560	-0.4556780	0.0309780	-2.4005970	-0.4653360	-0.0459660
N(3)	2.1913770	0.8834950	0.0093270	-2.1528820	0.8815590	0.0275320
C(4)	0.8949430	1.4552300	-0.0431320	-0.8779760	1.4720960	0.0806010
C(4a)	-0.1361920	0.4739610	-0.0940310	0.2295390	0.4997090	-0.0118200
N(5)	-1.4767470	0.8926120	-0.2078450	1.4741400	0.9806910	-0.1028520
C(6)	-2.4553200	-0.0236850	0.3655920	2.4583770	-0.0305610	-0.3749020
C(7)	-2.1825280	-1.4219630	-0.2200570	2.1737840	-1.3423110	0.4030220
N(8)	-0.8052640	-1.8044450	0.0351490	0.8437910	-1.8710740	0.1250050
C(8a)	0.1913140	-0.8642720	-0.0210140	-0.1598020	-0.8611760	-0.0007530
N(9)	3.7450550	-0.8614900	0.1375470	-3.7710350	-0.8421480	-0.1843530
O(10)	0.7568680	2.6741030	-0.0314170	-0.8045150	2.6959290	0.1525070
C(11)	-3.8687730	0.4512090	0.0426250	3.8566640	0.5145500	-0.0595220
H(12)	4.4191380	-0.3156150	-0.3805010	-4.2899390	-0.6371310	0.6710530
H(13)	3.8413900	-1.8542380	-0.0231650	-3.7700140	-1.8530010	-0.2838690
H(14)	2.9448090	1.5493500	0.1078460	-2.9129030	1.5434390	-0.0335160
H(15)	-1.5754320	1.8562170	0.0909510	-	-	-
H(16)	-4.0148880	0.5182930	-1.0397790	3.9468060	0.7251060	1.0132990
H(17)	-4.0481980	1.4436630	0.4687200	4.0100470	1.4596880	-0.5900250
H(18)	-4.6161200	-0.2307710	0.4600760	4.6533640	-0.1867820	-0.3464220
H(19)	-2.4157850	-1.4043520	-1.2954100	2.2495160	-1.1182950	1.4773840
H(20)	-2.8457950	-2.1519070	0.2560580	2.9143980	-2.1240630	0.1833140
H(21)	-0.5175720	-2.7570690	-0.1196860	0.8462170	-2.4865380	-0.6812920
H(22)	-2.3438470	-0.1068360	1.4613340	2.4576650	-0.3259380	-1.4519620

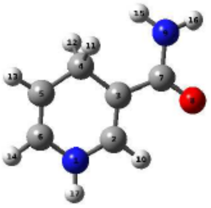
Group B

	cis-Me-H_2nic  16 17 9 13 12 4 7 14 5 3 8 6 2 15 1 11 10 18 19 20			*trans*-Me-H_2nic  8 13 12 4 7 16 14 5 3 9 17 6 2 15 1 11 10 18 19 20		
Atom	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	-1.7398450	-0.4389690	-0.0830370	-1.7102370	-0.5039830	-0.0732330
C(2)	-0.4161810	-0.7863080	-0.0498650	-0.3666940	-0.7833600	-0.0073380
C(3)	0.6023270	0.0989740	-0.0128850	0.6041480	0.1551000	-0.0303770
C(4)	0.3531440	1.5970210	0.0128640	0.2778730	1.6373650	-0.0462430
C(5)	-1.1304170	1.8794460	0.0360000	-1.2146570	1.8466010	-0.1080650
C(6)	-2.0562400	0.9187300	-0.0092780	-2.0932840	0.8428590	-0.1110680
C(7)	1.9652660	-0.4789460	0.0077280	2.0350980	-0.1957910	0.0440030
O(8)	2.1950530	-1.6812160	0.0309210	2.8803480	0.6306850	0.3522220
N(9)	2.9948300	0.4379160	0.0399050	2.3866010	-1.5180110	-0.2112580
C(10)	-2.7864870	-1.4347250	0.0432320	-2.6984280	-1.5218850	0.2235780
H(11)	-0.1883750	-1.8459600	-0.0591170	-0.1303610	-1.8380310	0.0821960
H(12)	0.8438760	2.0625320	0.8838950	0.7152890	2.1278850	0.8352610
H(13)	0.8102260	2.0927300	-0.8637230	0.7742730	2.1259530	-0.8983630
H(14)	-1.4582610	2.9119940	0.0797940	-1.5923080	2.8619320	-0.1601900
H(15)	-3.1181460	1.1358860	0.0024360	-3.1644650	1.0024030	-0.1563620
H(16)	2.8472560	1.3889110	-0.2527310	3.3858690	-1.6483000	-0.2791350
H(17)	3.9163320	0.0555480	-0.1024810	1.8557510	-2.0154340	-0.9089610
H(18)	-3.6335230	-1.1754740	-0.5999260	-3.6057180	-1.3415450	-0.3610160
H(19)	-3.1516360	-1.5290270	1.0752320	-2.9733810	-1.5483370	1.2877300
H(20)	-2.4015400	-2.4051920	-0.2756050	-2.3066110	-2.5033840	-0.0544400

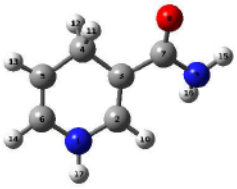
	cis-Me-Hnic$^+$  15 16 9 12 4 7 13 5 3 8 6 2 14 1 11 10 17 18 19			*trans*-Me-Hnic$^+$  8 12 4 7 15 13 5 3 9 16 6 2 14 1 11 10 17 19 18		
Atom	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	1.7032190	-0.3902260	0.0366930	1.7081140	-0.4248470	0.0258840
C(2)	0.4126260	-0.7605940	0.0445880	0.4044690	-0.7876960	0.0367410
C(3)	-0.6114670	0.1778720	0.0005850	-0.6007360	0.1580360	-0.0498910
C(4)	-0.2676000	1.5250600	-0.0759200	-0.2406480	1.5071280	-0.1332130
C(5)	1.0791990	1.8946510	-0.0903870	1.1024640	1.8625490	-0.1668060
C(6)	2.0492600	0.9188420	-0.0282050	2.0639190	0.8730770	-0.0824920
C(7)	-2.0101850	-0.4172330	-0.0196880	-2.0806780	-0.1696980	0.0484020
O(8)	-2.1312140	-1.6057970	-0.2367940	-2.8315730	0.6775860	0.4768510
N(9)	-3.0390850	0.4357660	0.1829240	-2.4491860	-1.4287190	-0.3092420
C(10)	2.7748790	-1.4126400	0.0926840	2.7564610	-1.4588600	0.1910560
H(11)	0.1780900	-1.8172000	0.0727600	0.2094950	-1.8463140	0.1501350
H(12)	-1.0281180	2.2945940	-0.1466560	-1.0310850	2.2491550	-0.1588150
H(13)	1.3777000	2.9334410	-0.1539000	1.4140410	2.8963150	-0.2480580
H(14)	3.1098550	1.1352040	-0.0333300	3.1261390	1.0807590	-0.0893510
H(15)	-2.9255400	1.3640120	0.5520340	-3.4407240	-1.6240710	-0.2879390
H(16)	-3.9670180	0.0382630	0.2058930	-1.8997340	-1.9817980	-0.9466780
H(17)	3.3807690	-1.2463590	0.9848920	3.6562070	-1.1410220	-0.3343010
H(18)	3.3946060	-1.3297650	-0.8016300	2.4024010	-2.3976370	-0.2330210
H(19)	2.3201500	-2.4003400	0.1350400	2.9718410	-1.5883320	1.2539450

Group C

***cis*-H_2nic**

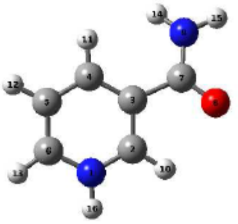


***trans*-H_2nic**

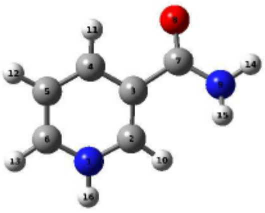


Atom	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	1.8162500	-1.1962800	0.0478900	1.7521170	-1.2644050	-0.0471120
C(2)	0.4485600	-1.1675270	0.0300790	0.3821090	-1.1527720	-0.0907700
C(3)	-0.2653470	-0.0235760	0.0128730	-0.2649590	0.0236710	0.0351490
C(4)	0.4148460	1.3368510	0.0077500	0.4911520	1.3349550	0.1717760
C(5)	1.9192110	1.1840410	-0.0413500	1.9810910	1.1113880	0.0678740
C(6)	2.5301320	-0.0024680	-0.0188970	2.5287120	-0.1001260	-0.0333650
C(7)	-1.7389970	-0.1727270	-0.0101080	-1.7369540	0.1214360	-0.0381250
O(8)	-2.3119420	-1.2535290	-0.0508510	-2.2984240	1.1791020	-0.2775950
N(9)	-2.4512880	1.0067130	-0.0198350	-2.4631170	-1.0512210	0.1382060
H(10)	-0.0679770	-2.1197910	0.0295960	-0.1475400	-2.0846900	-0.2519940
H(11)	0.0687470	1.9407910	-0.8473360	0.1394150	2.0402840	-0.5930110
H(12)	0.1312170	1.9257940	0.8997090	0.2377800	1.8226820	1.1264420
H(13)	2.5255070	2.0811870	-0.0919370	2.6338800	1.9770360	0.0833810
H(14)	3.6075600	-0.1102160	-0.0490890	3.5973330	-0.2647280	-0.0996160
H(15)	-2.0282050	1.8701720	0.2748110	-3.4569900	-0.8892150	0.2146930
H(16)	-3.4455340	0.9141450	0.1152200	-2.1010060	-1.7343020	0.7848470
H(17)	2.2990600	-2.0744390	-0.0026410	2.1746150	-2.1218250	-0.3568780

***cis*-$Hnic^+$**

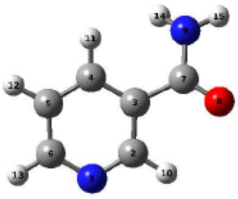


***trans*-$Hnic^+$**

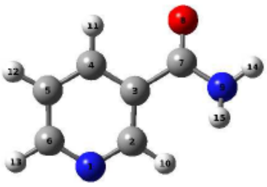


Atom	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	1.8126600	-1.1034740	0.0969070	-1.8004140	-1.1450080	-0.1516010
C(2)	0.4708470	-1.1333570	0.0957860	-0.4496170	-1.1634410	-0.1385350
C(3)	-0.2495760	0.0503570	0.0125520	0.2484550	0.0173600	0.0315260
C(4)	0.4634340	1.2457520	-0.0924960	-0.4827400	1.2052270	0.1673270
C(5)	1.8612340	1.2399890	-0.0970730	-1.8746940	1.1858580	0.1707390
C(6)	2.5297310	0.0389200	0.0048010	-2.5274080	-0.0206770	0.0035560
C(7)	-1.7594170	-0.1290850	-0.0200270	1.7628690	0.1317630	-0.0299460
O(8)	-2.2005320	-1.2424390	-0.2178860	2.2441610	1.1887190	-0.3709950
N(9)	-2.5113750	0.9809490	0.1477000	2.4689740	-0.9924720	0.2564540
C(10)	-0.0294820	-2.0925910	0.1473770	0.0168840	-2.1270170	-0.2979830
H(11)	-0.0557670	2.1924660	-0.1941150	0.0724550	2.1333090	0.2545110
H(12)	2.4267120	2.1590400	-0.1834400	-2.4528960	2.0936080	0.2878270
H(13)	3.6073440	-0.0588430	0.0113810	-3.6039330	-0.1303240	-0.0212940
H(14)	-2.1501670	1.8488270	0.5045670	3.4755680	-0.8993900	0.2552720
H(15)	-3.5137560	0.8577470	0.1596260	2.0899040	-1.7329290	0.8240130
H(16)	2.3128620	-1.9849220	0.1641830	-2.2923930	-2.0211880	-0.2963570

***cis*-nic**

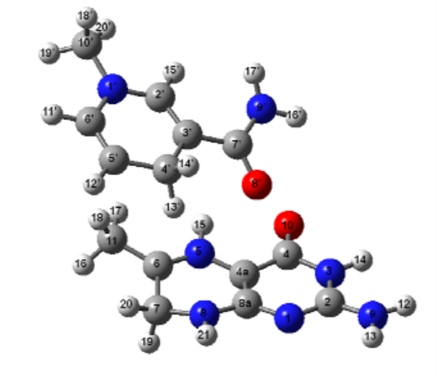
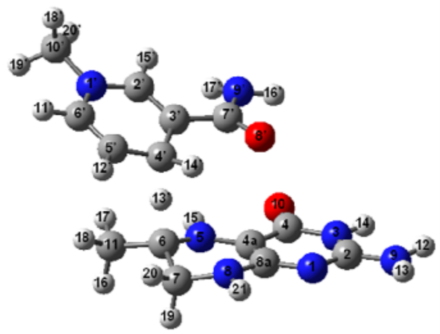
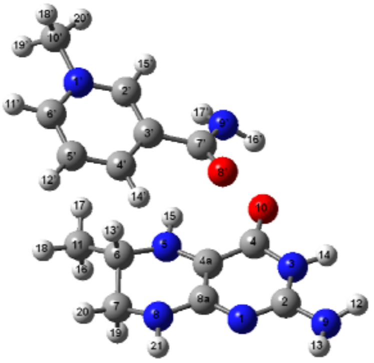


***trans*-nic**



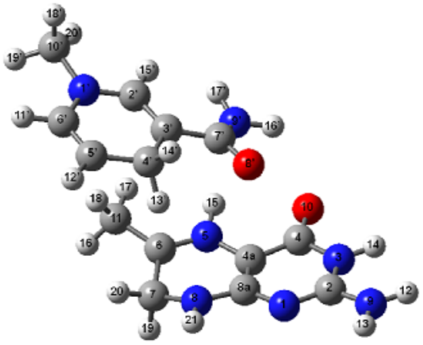
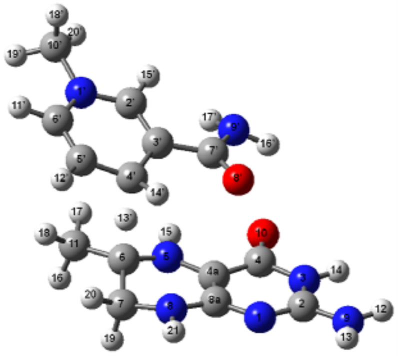
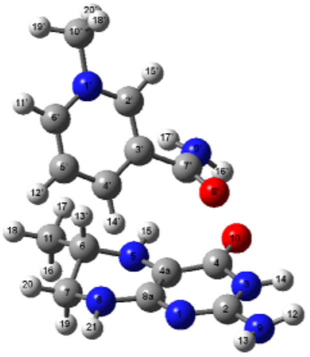
Atom	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	1.8777390	-1.2089270	0.1550700	-1.8250300	-1.2671730	-0.1191470
C(2)	0.5458880	-1.1775860	0.1229320	-0.4941880	-1.1721920	-0.1314720
C(3)	-0.2075620	-0.0061600	-0.0167010	0.2065730	0.0308900	-0.0102670
C(4)	0.4850670	1.1976530	-0.1528570	-0.5389810	1.2060910	0.1031720
C(5)	1.8743840	1.1802930	-0.1296810	-1.9231360	1.1250500	0.1203230
C(6)	2.5197580	-0.0427870	0.0311320	-2.5161360	-0.1320520	0.0131960
C(7)	-1.7022950	-0.1320300	-0.0361530	1.7034050	0.1449900	-0.0268480
O(8)	-2.2560220	-1.1869470	-0.2823340	2.2595610	1.2086110	-0.2313390
N(9)	-2.4052450	1.0220720	0.2122830	2.4016000	-1.0163440	0.1759500
C(10)	0.0229980	-2.1248140	0.2121620	0.0377150	-2.1106810	-0.2732650
H(11)	-0.0460560	2.1317080	-0.3054280	-0.0116710	2.1505350	0.1723620
H(12)	2.4485050	2.0931560	-0.2412370	-2.5372170	2.0138020	0.2130800
H(13)	3.6059980	-0.0901100	0.0553390	-3.5988700	-0.2320760	0.0287700
H(14)	-1.9812240	1.7727700	0.7311640	3.3996690	-0.9163460	0.2764150
H(15)	-3.4009330	0.9045590	0.3231570	1.9726850	-1.8061510	0.6271000

Group D: B3LYP/6-31+G(d,p)

	Reactant State complex			Transition State complex			Product State complex		
	 18' 20' 19' 10' 1' 15' 2' 17' 9' 16' 11' 6' 3' 7' 14' 5' 12' 4' 8' 13' 18 17 11 15 10 16 6 5 4a 4 3 14 20 7 8 8a 1 2 9 12 19 21 13			 18 20' 10' 19' 15' 1' 2' 17' 9' 16' 11' 6' 3' 7' 8' 5' 4' 14' 12' 13 17 15 10 14 18 11 6 5 4a 4 3 2 12 16 20 7 8 8a 1 9 13 21 19			 18' 20' 10' 19' 1' 15' 2' 11' 6' 3' 17' 9' 16' 7' 5' 4' 12' 8' 14' 17 13 15 10 18 11 6 5 4a 4 3 14 16 20 7 8 8a 1 2 12 9 19 21 13		
Atom	*x*	*y*	*z*	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	3.3619270	0.8094190	-0.9939910	3.3565400	0.7471030	-1.0146290	3.3658390	0.8397480	-0.9397950
C(2)	3.9423860	-0.3366890	-0.6604950	3.9954720	-0.3373850	-0.6155610	3.9168580	-0.3174820	-0.6564010
N(3)	5.2337880	-0.2237440	-2.2148760	3.5814990	-1.0977650	0.4449140	3.4229750	-1.1469030	0.3119170
C(4)	2.4671520	-0.7563650	1.2444400	2.4120350	-0.8225660	1.1842810	2.2747660	-0.8532310	1.0693440
C(4a)	1.9109460	0.5156500	0.9105180	1.7752470	0.3837410	0.7759590	1.7358180	0.4387050	0.8168280
N(5)	0.6895180	0.8615140	1.4804220	0.5772380	0.7521010	1.3891210	0.6573530	0.9091370	1.5699770
C(6)	0.0285420	1.9501570	1.2154330	-0.2259790	1.7339600	0.8995540	-0.1485910	1.9889260	1.0410250
C(7)	0.7287590	2.9724110	0.3475410	0.5317710	2.7893530	0.0896090	0.8001540	3.0463700	0.4428820
N(8)	5.4568820	-1.6370450	-1.2298140	1.5237030	2.1614190	-0.7751940	1.7056800	2.4182070	-0.5146660
C(8a)	2.3368490	1.2137040	-0.2250300	2.2445630	1.0910020	-0.3286590	2.2893030	1.2155260	-0.1993970
N(9)	4.9687510	-0.7742840	-1.4112610	5.1300710	-0.7010690	-1.2558710	5.0582720	-0.7000500	-1.3027280
O(10)	2.0729440	-1.5091070	2.1295040	2.0243800	-1.5812690	2.0777260	1.8207290	-1.6953660	1.8692860
C(11)	-1.3216150	2.2133420	1.7775150	-1.3058420	2.2477240	1.8220040	-1.0398110	2.5798780	2.1322240
H(12)	5.4568820	-1.6370450	-1.2298140	5.5800670	-1.5883830	-1.0960150	5.2976740	-1.6784950	-1.3684450
H(13)	5.2337880	-0.2237440	-2.2148760	5.3791270	-0.1818460	-2.0847370	5.3149640	-0.1278020	-2.0947250
H(14)	3.9941990	-1.9843290	0.6160360	4.0805670	-1.9348790	0.7238290	3.8672130	-2.0317520	0.5279050
H(15)	0.2960400	0.1407070	2.0919990	0.2673670	0.1398400	2.1386660	0.2391140	0.2329590	2.1950010
H(16)	-1.3592120	3.2118840	2.2288850	-0.8798640	2.9432150	2.5552430	-0.4356140	2.9909300	2.9467740
H(17)	-1.6022550	1.4730890	2.5302890	-1.7826690	1.4275410	2.3670930	-1.6976080	1.8128480	2.5574950
H(18)	-2.0703100	2.1900390	0.9731290	-2.0775260	2.7801920	1.2594160	-1.6718100	3.3779260	1.7291510
H(19)	1.2590890	3.6817600	1.0099600	0.9931530	3.4963420	0.8010310	1.3481800	3.5348940	1.2611290
H(20)	-0.0133990	3.5464700	-0.2125590	-0.1728660	3.3554720	-0.5257800	0.2152710	3.8092120	-0.0793660
H(21)	1.9297640	2.8004470	-1.4261190	1.9577220	2.7161530	-1.5018040	2.2572890	3.0152720	-1.1174650
N(1')	-4.4230930	-0.5048470	-0.1651750	-4.2314360	-0.4552060	-0.1655920	-4.4038950	-0.5939110	-0.2756300
C(2')	-3.2844450	-1.2635990	-0.0803520	-3.1228050	-1.2480590	-0.0756430	-3.2826800	-1.3387180	-0.1000430
C(3')	-2.1144630	-0.9493110	-0.7016550	-1.8840180	-0.8443350	-0.5009470	-2.0763890	-0.9682900	-0.6735040
C(4')	-2.0020950	0.2685660	-1.6074840	-1.6987080	0.5158270	-1.0281390	-2.0360630	0.1956340	-1.4636590
C(5')	-3.2637810	1.0971870	-1.5380670	-2.9448310	1.2382620	-1.2649570	-3.1980820	0.9543410	-1.6271710
C(6')	-4.3678530	0.6930450	-0.8899460	-4.1314070	0.7684310	-0.8035050	-4.3715560	0.5456480	-1.0220780
C(7')	-0.8853340	-1.7407800	-0.5017710	-0.7059970	-1.7645810	-0.5329650	-0.7956610	-1.7660630	-0.5175260
O(8')	0.2267240	-1.2652330	-0.7821230	0.2571810	-1.5021490	-1.2524230	0.0680260	-1.6571880	-1.3769270
N(9')	-0.9831270	-2.9964280	0.0531260	-0.7584480	-2.9092560	0.2210460	-0.6969320	-2.5722640	0.5748560
C(10')	-5.6993340	-0.9762920	0.3640960	-5.5588770	-0.9643380	0.2092340	-5.6921250	-1.0251610	0.3153900
H(11')	-5.2817270	1.2768200	-0.8769870	-5.0662350	1.2974470	-0.9478280	-5.3060940	1.0858470	-1.1136560
H(12')	-3.2981670	2.0441160	-2.0683160	-2.9348780	2.1731530	-1.8147580	-3.2005060	1.8587540	-2.2248720
H(13')[†]	-1.1195740	0.8623130	-1.3291490	-1.0182970	1.1579300	-0.1125810	-0.7920700	1.6345400	0.2083690
H(14')	-1.7977860	-0.0413930	-2.6447080	-0.9244320	0.5982260	-1.7941650	-1.1042880	0.4579380	-1.9533040
H(15')	-3.3691840	-2.1267470	0.5720140	-3.3055890	-2.2509940	0.2930080	-3.4013730	-2.2430680	0.4844270
H(16')	-0.1164440	-3.5129970	0.1133950	0.1045980	-3.4365490	0.2556140	0.2528910	-2.8555800	0.8125670
H(17')	-1.8200270	-3.5480280	-0.0624740	-1.3148500	-2.9489160	1.0622520	-1.2571190	-2.3756350	1.3923830
H(18')	-6.3338470	-1.3949720	-0.4267300	-6.1282380	-1.2485410	-0.6812940	-6.3802040	-1.3123850	-0.4826320
H(19')	-6.2336820	-0.1517030	0.8446260	-6.1053890	-0.1953070	0.7593010	-6.1154310	-0.2030970	0.8950550
H(20')	-5.5192670	-1.7495620	1.1137900	-5.4421560	-1.8362750	0.8529060	-5.5184870	-1.8774930	0.9702910

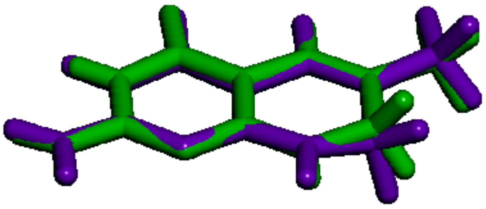
[†] The transferring hydride

Group D: B3LYP-D/6-31+G(d,p)

	Reactant State complex			Transition State complex			Product State complex		
									
Atom	*x*	*y*	*z*	*x*	*y*	*z*	*x*	*y*	*z*
N(1)	3.3252290	1.1518350	-0.7667330	3.3062810	0.8847160	-0.8943060	2.4520150	-0.4326970	-1.6479740
C(2)	4.0150990	0.0194930	-0.8007680	3.9219590	-0.2479930	-0.6096810	2.9958110	-1.4175790	-0.9679990
N(3)	3.7105200	-1.0700950	-0.0305400	5.0599770	-0.5603340	-1.2675000	3.0189210	-1.4383260	0.3965160
C(4)	2.5821870	-1.1307430	0.8149580	2.2927370	-0.9160310	1.0823550	2.3988680	-0.4632630	1.1897300
C(4a)	1.8993470	0.1222860	0.8947110	1.6849470	0.3426370	0.8150730	1.8849480	0.6427960	0.4562360
N(5)	0.6298440	0.1465360	1.4732980	0.4475410	0.6263320	1.3981900	1.3503740	1.7494850	1.1324840
C(6)	-0.1339030	1.1973430	1.5209800	-0.3324750	1.6487060	0.9667220	0.3743200	2.5408130	0.4048530
C(7)	0.4831850	2.5079490	1.0907150	0.4496880	2.8000080	0.3398590	0.9621630	2.8427380	-0.9853420
N(8)	1.4576700	2.2753100	0.0295830	1.4752550	2.2742340	-0.5525420	1.2909530	1.5868600	-1.6666860
C(8a)	2.2626690	1.1862780	0.0552190	2.1821860	1.1606560	-0.1990910	1.9168330	0.5914910	-0.9331610
N(9)	5.0796780	-0.0617720	-1.6174100	5.0599770	-0.5603340	-1.2675000	3.6122620	-2.4399370	-1.6283730
O(10)	2.2695490	-2.1738010	1.3829420	1.8637300	-1.7808920	1.8544540	2.3088330	-0.6221930	2.4241160
C(11)	-0.1339030	1.1973430	1.5209800	-1.5274500	2.0044930	1.8088770	0.0629090	3.8265300	1.1631590
H(12)	5.6367220	-0.8973260	-1.7067170	5.5050690	-1.4598920	-1.1758160	3.7258170	-3.3329430	-1.1716820
H(13)	5.2924530	0.7346070	-2.2008580	5.3443240	0.0458210	-2.0230090	3.4667300	-2.4628280	-2.6277000
H(14)	4.2233420	-1.9399330	-0.1249270	3.9509380	-1.9974090	0.5224690	3.3924480	-2.2325740	0.9029950
H(15)	0.2583570	-0.7761650	1.7360620	0.0863830	-0.0790080	2.0330730	1.1745420	1.5816360	2.1158890
H(16)	-1.6948420	1.8916700	2.7860530	-1.2306620	2.6589260	2.6380370	0.9723760	4.4262860	1.2738540
H(17)	-1.7886650	0.1481760	2.3886860	-1.9892960	1.1016550	2.2226760	-0.3219730	3.6019860	2.1649050
H(18)	-2.2122410	1.3753720	1.1696520	-2.2725100	2.5289240	1.2019740	-0.6912710	4.4162730	0.6306810
H(19)	0.9438460	2.9833240	1.9764290	0.8785890	3.4073180	1.1560110	1.8465890	3.4815010	-0.8550210
H(20)	-0.3008970	3.1742110	0.7226590	-0.2358770	3.4322640	-0.2312500	0.2248350	3.3836710	-1.5868110
H(21)	1.7066520	3.0282760	-0.5983570	1.9255430	2.9024360	-1.2051390	1.6181550	1.6559320	-2.6231100
N(1’)	-4.2654400	-0.2921300	-0.2534750	-4.1087640	-0.4932410	-0.2280910	-4.1870840	-0.5772570	-0.1450430
C(2’)	-3.1362200	-1.0632790	-0.2160420	-2.9746330	-1.2476730	-0.1309260	-3.1202740	-1.0304010	0.5658820
C(3’)	-1.9385480	-0.6791020	-0.7368600	-1.7514340	-0.8052370	-0.5591400	-1.8272550	-0.8151440	0.1204800
C(4’)	-1.7920100	0.6256830	-1.5080930	-1.6154450	0.5577600	-1.0971190	-1.6230360	-0.1065760	-1.0753010
C(5’)	-3.0618660	1.4406490	-1.4150050	-2.8842650	1.2428210	-1.3319120	-2.7339000	0.3705080	-1.7760050
C(6’)	-4.1873370	0.9759450	-0.8504340	-4.0528460	0.7332740	-0.8662320	-4.0092110	0.1150200	-1.2969910
C(7’)	-0.7059690	-1.4594350	-0.5801000	-0.5298800	-1.6606660	-0.5327440	-0.6406090	-1.4091380	0.8494870
O(8’)	0.3180740	-1.1540120	-1.2026200	0.4464220	-1.3603940	-1.2208530	0.0901960	-2.1758350	0.2368500
N(9’)	-0.6391130	-2.4354260	0.4098570	-0.5467650	-2.7806220	0.2576580	-0.5075540	-1.0560620	2.1499190
C(10’)	-5.5748890	-0.8482480	0.0800900	-5.4196190	-1.0579510	0.1232470	-5.5565630	-0.9040650	0.3189350
H(11’)	-5.1049690	1.5523120	-0.8113160	-5.0062930	1.2300770	-1.0033100	-4.9088220	0.4425230	-1.8041010
H(12’)	-3.0782230	2.4356290	-1.8486170	-2.9029830	2.1813090	-1.8748750	-2.6187050	0.9318710	-2.6962240
H(13’)[†]	-0.9316550	1.1902250	-1.1119220	-0.9932400	1.1891610	-0.1734780	-0.5645200	1.9714880	0.2426200
H(14’)	-1.5208200	0.4283380	-2.5558020	-0.8344370	0.6715250	-1.8528290	-0.6156120	0.0701080	-1.4424100
H(15’)	-3.2730270	-2.0178480	0.2832260	-3.1216810	-2.2514170	0.2522850	-3.3427190	-1.5775150	1.4750360
H(16’)	0.2003930	-3.0051880	0.3871580	0.3654830	-3.1885520	0.4255300	0.4116250	-1.2131890	2.5774610
H(17’)	-1.4751790	-2.9135890	0.7147470	-1.1657040	-2.8187960	1.0543760	-1.0250410	-0.2627500	2.5020380
H(18’)	-6.0953190	-1.2122750	-0.8152940	-5.9390590	-1.4069310	-0.7761590	-5.8259540	-1.9028690	-0.0355590
H(19’)	-6.1869240	-0.0778820	0.5587070	-6.0225250	-0.2927490	0.6188210	-6.2526590	-0.1651910	-0.0798910
H(20’)	-5.4474550	-1.6783980	0.7801990	-5.2774360	-1.8960770	0.8080030	-5.5773060	-0.8760430	1.4096270

[†] The transferring hydride

Table S7. Comparison between the X-ray crystal structure of 6-methyl-7,8-dihydropterin-monohydrochloride-monohydrate[22] (green, isotropic/anisotropic temperature factors are not shown) and the computed geometry of 6-Me-H_3pterin$^+$ at the B3LYP/6-31G(2df,p) level (purple).



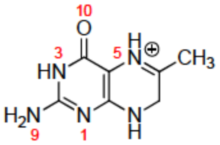
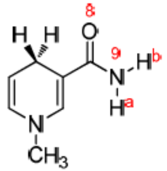
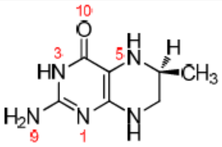
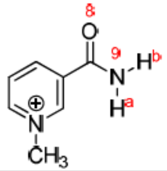
	exp.	calc.		exp.	calc.
Bond Distances (Å)[a]					
N(1)-C(2)	1.333 ± 0.003	1.328	N(5)-C(6)	1.294 ± 0.003	1.297
N(1)-C(8a)	1.355 ± 0.003	1.339	N(5)-H(15)	0.950 ± 0.030	1.023
C(2)-N(3)	1.360 ± 0.003	1.365	C(6)-C(7)	1.491 ± 0.004	1.512
C(2)-N(9)	1.329 ± 0.004	1.338	C(6)-C(11)	1.483 ± 0.004	1.484
N(9)-H(12)	0.890 ± 0.028	1.006	C(11)-H(16)	0.902 ± 0.034	1.097
N(9)-H(13)	0.925 ± 0.030	1.007	C(11)-H(17)	1.005 ± 0.033	1.092
N(3)-C(4)	1.389 ± 0.003	1.414	C(11)-H(18)	0.988 ± 0.033	1.097
N(3)-H(14)	0.840 ± 0.030	1.012	C(7)-N(8)	1.451 ± 0.004	1.459
C(4)-C(4a)	1.402 ± 0.003	1.426	C(7)-H(19)	0.957 ± 0.032	1.095
C(4)-O(10)	1.241 ± 0.003	1.216	C(7)-H(20)	0.972 ± 0.032	1.105
C(4a)-C(8a)	1.394 ± 0.003	1.395	N(8)-C(8a)	1.337 ± 0.003	1.352
C(4a)-N(5)	1.412 ± 0.003	1.389	N(8)-H(21)	0.906 ± 0.028	1.010
Bond Angles (deg)[a]					
C(2)-N(1)-C(8a)	115.89 ± 0.21	116.75	N(5)-C(6)-C(11)	121.52 ± 0.25	121.97
N(1)-C(2)-N(3)	123.11 ± 0.22	123.53	C(7)-C(6)-C(11)	118.59 ± 0.25	120.16
N(1)-C(2)-N(9)	119.01 ± 0.23	118.12	C(6)-C(7)-N(8)	113.99 ± 0.24	111.89
N(3)-C(2)-N(9)	117.88 ± 0.23	118.35	C(6)-C(7)-H(19)	107.18 ± 1.90	108.96
C(2)-N(9)-H(12)	122.46 ± 1.82	123.25	C(6)-C(7)-H(20)	107.91 ± 1.92	107.68
C(2)-N(9)-H(13)	118.61 ± 1.82	117.85	N(8)-C(7)-H(19)	111.94 ± 1.90	109.56
H(12)-N(9)-H(13)	118.54 ± 2.56	118.90	N(8)-C(7)-H(20)	110.61 ± 1.92	112.16
C(2)-N(3)-C(4)	123.58 ± 0.22	123.42	H(19)-C(7)-H(20)	104.70 ± 2.69	106.39
C(2)-N(3)-H(14)	117.02 ± 2.07	121.41	C(7)-N(8)-C(8a)	123.75 ± 0.23	122.48
C(4)-N(3)-H(14)	119.36 ± 2.07	115.12	C(7)-N(8)-H(21)	114.64 ± 1.72	118.70
N(3)-C(4)-C(4a)	113.32 ± 0.21	111.36	C(8a)-N(8)-H(21)	121.27 ± 1.72	116.97
N(3)-C(4)-O(10)	119.78 ± 0.22	122.39	N(8)-C(8a)-C(4a)	119.14 ± 0.22	118.07
C(4a)-C(4)-O(10)	126.89 ± 0.23	126.25	N(8)-C(8a)-N(l)	117.68 ± 0.22	119.07
C(4)-C(4a)-C(8a)	120.89 ± 0.22	122.04	N(1)-C(8a)-C(4a)	123.18 ± 0.22	122.81
C(4)-C(4a)-N(5)	119.80 ± 0.21	118.36	C(6)-C(11)-H(16)	114.33 ± 2.19	110.11
N(5)-C(4a)-C(8a)	119.25 ± 0.21	119.10	C(6)-C(11)-H(17)	113.24 ± 1.90	112.14
C(4a)-N(5)-C(6)	123.69 ± 0.21	125.20	C(6)-C(11)-H(18)	110.53 ± 1.93	110.13
C(4a)-N(5)-H(15)	116.71 ± 1.76	113.60	H(16)-C(11)-H(17)	110.47 ± 2.89	108.76
C(6)-N(5)-H(15)	119.59 ± 1.77	121.19	H(16)-C(11)-H(18)	102.49 ± 2.91	106.65
N(5)-C(6)-C(7)	119.89 ± 0.24	117.81	H(17)-C(11)-H(18)	104.83 ± 2.69	108.88
Dihedral Angles (deg)[b]					
N(1)-C(2)-N(3)-H(14)	-178.71	-178.55	N(5)-C(6)-C(11)-H(18)	-121.03	117.36
N(1)-C(2)-N(9)-H(12)	-173.38	-179.28	N(5)-C(6)-C(7)-H(19)	-118.89	-142.39
N(1)-C(2)-N(9)-H(13)	-0.59	0.03	N(5)-C(6)-C(7)-H(20)	128.85	102.61
N(3)-C(2)-N(9)-H(12)	7.39	0.65	N(5)-C(4a)-C(8a)-N(1)	-179.13	174.08
N(3)-C(2)-N(9)-H(13)	-179.82	179.96	C(7)-C(6)-C(11)-H(16)	-55.29	-65.62
C(2)-N(3)-C(4)-O(10)	179.14	-176.52	C(7)-C(6)-C(11)-H(17)	177.10	173.13
H(14)-N(3)-C(4)-O(10)	-3.38	0.98	C(7)-C(6)-C(11)-H(18)	59.78	51.72
N(3)-C(4)-C(4a)-N(5)	178.45	-175.22	C(11)-C(6)-C(7)-H(19)	60.32	40.47
O(10)-C(4)-C(4a)-N(5)	-0.35	4.20	C(11)-C(6)-C(7)-H(20)	-51.94	-74.53
C(4)-C(4a)-N(5)-H(15)	0.62	1.78	C(11)-C(6)-C(7)-N(8)	-175.17	161.77
C(4)-C(4a)-N(5)-C(6)	-178.72	-179.02	C(6)-C(7)-N(8)-H(21)	-179.61	-168.66
C(4)-C(4a)-C(8a)-N(8)	178.33	-175.14	C(6)-C(7)-N(8)-C(8a)	-6.18	27.34
C(4a)-N(5)-C(6)-C(11)	178.63	-178.39	C(7)-N(8)-C(8a)-C(4a)	3.03	-15.72
C(4a)-N(5)-C(6)-C(7)	-2.19	4.52	C(7)-N(8)-C(8a)-N(1)	-176.84	166.71
H(15)-N(5)-C(6)-C(11)	-0.69	0.75	H(21)-N(8)-C(8a)-C(4a)	176.04	-179.99
N(5)-C(6)-C(11)-H(16)	123.90	-125.30	H(21)-N(8)-C(8a)-N(1)	-3.83	2.44
N(5)-C(6)-C(11)-H(17)	-3.71	-3.89	N(8)-C(8a)-N(1)-C(2)	-179.24	177.18

[a] Bond lengths and angles are given as originally reported in ref. 22.

[b] The torsion angles were computationally measured based on the crystal structure reported in ref. 22.

Table S8. QM/MM interactions in ligand-water complexes

(a) List of complexes

Model	Structure	Hydrogen bonding position with H_2O
6-Me-H_3pterin$^+$		N^1,N^9 N^3,N^9 N^3,O^{10} N^5,O^{10}
trans-Me-H_2nic		$N^9(H^a)$ $N^9(H^b),O^8$ § O^8
6-Me-H_4pterin		N^1,N^9 N^3,N^9 N^3,O^{10} N^5,O^{10}
trans-Me-Hnic$^+$		$N^9(H^a)$ $N^9(H^b),O^8$ § O^8 †

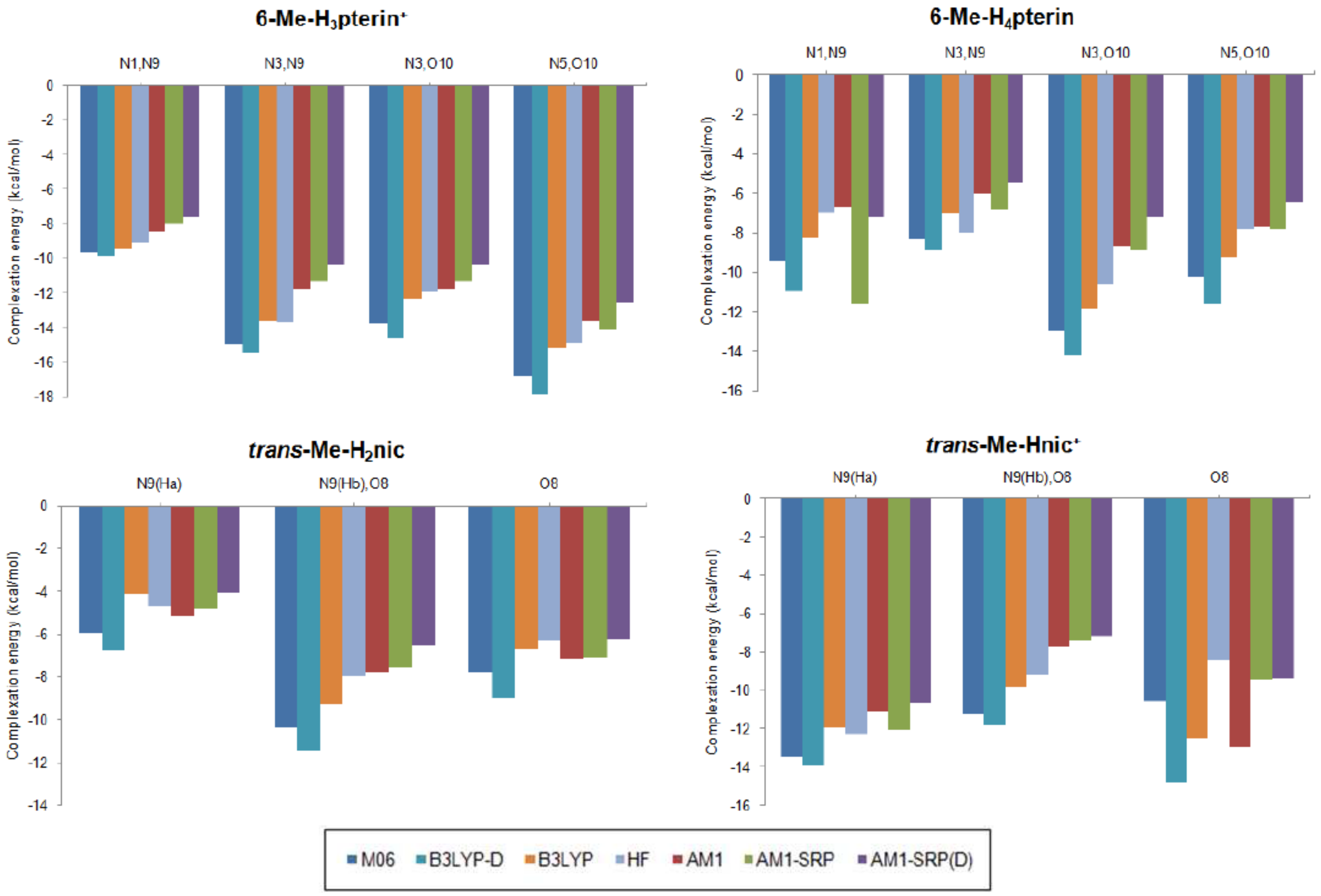
§ The water molecule forms a H-bonded bridge between $N^9(H^b)$ and O^8 in the optimized complex, regardless of whether the water is initially directed only towards $N^9(H^b)$ or between $N^9(H^b)$ and O^8. With both initial structures, the complexes optimize to the same local minimum.

† For these complexes, the potential energy minimum is very shallow, yielding considerably different optimized geometries with different methods, therefore their corresponding complexation energies were not included in the RMSD calculation (though shown in Figure S2, see below).

(b) Root mean square deviations (in kcal/mol) of the QM/MM complexation energies with respect to *ab initio* / DFT calculations

	M06/6-31+G(d,p)	B3LYP-D/6-31+G(d,p)	B3LYP/6-31+G(d,p)	HF/6-31+G(d,p)
QM(AM1)/MM	2.6	3.5	1.6	1.2
QM(AM1-SRP)/MM	2.6	3.3	1.8	1.7
QM(AM1-SRP(D))/MM	3.5	4.4	2.4	2.0

Figure S2. Complexation energies for the ligand-water complexes described in Table S8, calculated with different methods.



Technical details concerning the equilibration phase during molecular dynamics simulations

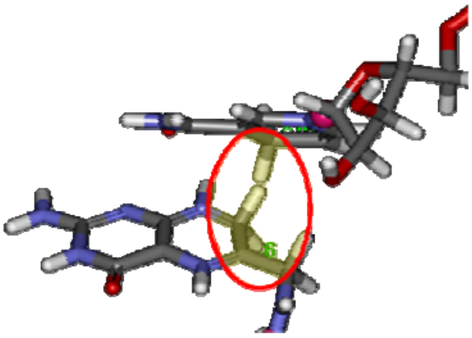


Figure S3. Streic clash between the *pro*-S hydrogen on C7 (of H_3folate$^+$) and the *pro*-R hydrogen on C4 (of NADPH)

Some dynamical issues emerged during the equilibration phase which required some intervention. First and foremost among these are the structural differences between the ligands originally bound in the crystal structure of *E. coli* DHFR (folate and NADP$^+$) and those modeled in this study (H_3folate$^+$ and NADPH): While folate consists of a fully oxidized pyrazine ring with no tetrahedral centers, H_3folate$^+$ contains a *sp^3*-hybridized C7 methylene carbon. In particular, the *pro*-S hydrogen is located in van der Waals contact with the donated hydride (*pro-R*) on the C4 carbon of the nicotinamide moiety in NADPH (Figure S3). This steric congestion inevitably results in moderate repulsion between the substrate and the coenzyme during the minimization and equilibration stage, prohibiting the donated hydride from approaching the acceptor carbon. This issue was addressed by imposing a Nuclear Overhauser Effect (NOE) harmonic restraint on the distance between the donor and acceptor carbons (C4N in NADPH and C6 in H_3folate$^+$, respectively).

Secondly, in light of the flexibility of the protein, some specific hydrogen bond interactions within the protein (which are characteristic of the closed conformation), as well as key interactions between the ligands and the protein residues, were distorted or lost during MD heating or equilibration. Therefore, NOE harmonic restraints were occasionally placed as well on the distances between the donors and acceptors of selected hydrogen bonds, such as that between the carboxamide group of the nicotinamide moiety in NADPH and residues Ala7 and Ile14, and that between Asp27 and the proximate nitrogens of the 4-oxopyrimidine moiety in the substrate's pterin ring.

References

1. Cummins, P. L.; Greatbanks, S. P.; Rendell, A. P.; Gready, J. E., Computational Methods for the Study of Enzymic Reaction Mechanisms. 1. Application to the Hydride Transfer Step in the Catalysis of Dihydrofolate Reductase. *J. Phys. Chem. B* **2002,** *106*, 9934-9944.

2. Agarwal, P. K.; Billeter, S. R.; Hammes-Schiffer, S., Nuclear quantum effects and enzyme dynamics in dihydrofolate reductase catalysis. *J. Phys. Chem. B* **2002,** *106*, 3283-3293.

3. Thorpe, I. F.; Brooks, C. L., Barriers to Hydride Transfer in Wild Type and Mutant Dihydrofolate Reductase from E. coli. *J. Phys. Chem. B* **2003,** *107*, 14042-14051.

4. Garcia-Viloca, M.; Truhlar, D. G.; Gao, J., Reaction-Path Energetics and Kinetics of the Hydride Transfer Reaction Catalyzed by Dihydrofolate Reductase. *Biochemistry* **2003,** *42*, 13558-13575.

5. Wong, K. F.; Selzer, T.; Benkovic, S. J.; Hammes-Schiffer, S., Chemical theory and computation special feature: impact of distal mutations on the network of coupled motions correlated to hydride transfer in dihydrofolate reductase. *Proc. Nat. Acad. Sci. U.S.A.* **2005,** *102*, 6807-6812.

6. Thorpe, I. F.; Brooks, C. L., Conformational Substates Modulate Hydride Transfer in Dihydrofolate Reductase. *J. Am. Chem. Soc.* **2005,** *127*, 12997-13006.

7. Liu, H.; Warshel, A., The Catalytic Effect of Dihydrofolate Reductase and Its Mutants Is Determined by Reorganization Energies. *Biochemistry* **2007,** *46*, 6011-6025.

8. Kamath, G.; Howell, E. E.; Agarwal, P. K., The Tail Wagging the Dog: Insights into Catalysis in R67 Dihydrofolate Reductase. *Biochemistry* **2010,** *49*, 9078-9088.

9. GaussView 4.1.2, Gaussian Inc.; Wallingford, CT, 2004.

10. Gaussian 03, Revision C.02, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, 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, C. Gonzalez, and J. A. Pople, Gaussian, Inc., Wallingford CT, 2004.

11. Gaussian 09, Revision B.01, Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, N. J.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. Gaussian, Inc., Wallingford CT, 2009.

12. Scott, A. P.; Radom, L., Harmonic Vibrational Frequencies:□ An Evaluation of Hartree–Fock, Møller–Plesset, Quadratic Configuration Interaction, Density Functional Theory, and Semiempirical Scale Factors. *J. Phys. Chem.* **1996,** *100*, 16502-16513.

13. Merrick, J. P.; Moran, D.; Radom, L., An Evaluation of Harmonic Vibrational Frequency Scale Factors. *J. Phys. Chem. A* **2007,** *111*, 11683-11700.

14. Alecu, I. M.; Zheng, J.; Zhao, Y.; Truhlar, D. G., Computational Thermochemistry: Scale Factor Databases and Scale Factors for Vibrational Frequencies Obtained from Electronic Model Chemistries. *J. Chem. Theory Comput.* **2010,** *6*, 2872-2887.

15. Curtiss, L. A.; Redfern, P. C.; Raghavachari, K., Gaussian-4 theory using reduced order perturbation theory. *J. Chem. Phys.* **2007,** *127*, 124105.

16. Casasnovas, R.; Frau, J.; Ortega-Castro, J.; Salvà, A.; Donoso, J.; Muñoz, F., Simplification of the CBS-QB3 method for predicting gas-phase deprotonation free energies. *Int. J. Quantum Chem* **2009,** *110*, 323-330.

17. Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P., AM1: a new general purpose quantum mechanical molecular model. *J. Am. Chem. Soc.* **1985,** *107*, 3902-3909.

18. http://www.gaussian.com/g_whitepap/thermo/thermo.pdf

19. Ribeiro da Silva, M. D. M. C.; Gonçalves, J. M.; Ferreira, S. C. C.; da Silva, L. C. M.; Sottomayor, M. J.; Pilcher, G.; Acree Jr, W. E.; Roy, L. E., Experimental thermochemical study of the enthalpies of formation and sublimation of isonicotinamide, picolinamide, nicotinamide, isonicotinamide N-oxide, and nicotinamide N-oxide. The dissociation enthalpies of the N-O bonds. *J. Chem. Thermodyn.* **2001,** *33*, 1263-1275.

20. Lias, S.; Bartmess, J.; Liebman, J.; Holmes, J.; Levin, R.; Mallard, W., Gas-phase ion and neutral thermochemistry. *J. Phys. Chem. Ref. Data* **1988,** *17*, 1-861.

21. Goebbert, D. J.; Wenthold, P. G., Gas-phase hydride affinities of neutral molecules. *Int. J. Mass spectrom.* **2006,** *257*, 1-11.

22. Bieri, J., The crystal structure of 6-methyl-7,8-dihydropterine-monohydrochloride-monohydrate. *Helv. Chim. Acta* **1977,** *60*, 2303-2308.

Appendix D

Specific Reaction Path Hamiltonian for Proton Transfer in Water: Reparameterized Semiempirical Models

Xin Wu, Walter Thiel, Soroosh Pezeshki, and Hai Lin

Journal of Chemical Theory and Computation, **2013**, 9, 2672 – 2686.

I optimized the parameters of the semiempirical methods for the water clusters (denoted as MNDOn, AM1n, PM3n, OM1n, OM2n, and OM3n in the paper), produced some of the figures, and drafted the paragraphs of parameterization.

JCTC Journal of Chemical Theory and Computation
Article
pubs.acs.org/JCTC

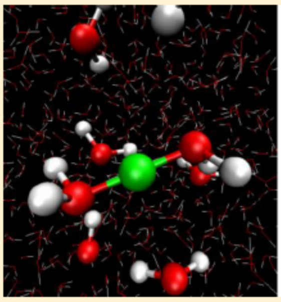
Specific Reaction Path Hamiltonian for Proton Transfer in Water: Reparameterized Semiempirical Models

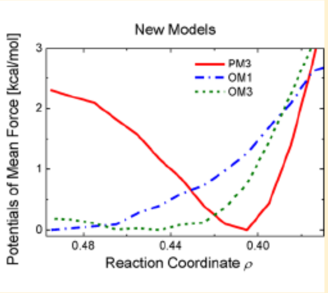
Xin Wu,[†] Walter Thiel,[*,†] Soroosh Pezeshki,[‡] and Hai Lin[*,‡]

[†]Max-Planck-Institut für Kohlenforschung, 45470 Mülheim an der Ruhr, Germany

[‡]Chemistry Department, University of Colorado, Denver, Denver, Colorado 80217, United States

Ⓢ *Supporting Information*





ABSTRACT: The semiempirical MNDO-based AM1 and PM3 methods and the orthogonalization-corrected OM1, OM2, and OM3 models were reparameterized to improve their description of bulk water and of proton transfer in water. Reference data included the gas-phase geometries and energies of the water molecule, small water clusters, the hydronium ion, and small hydronium ion−water clusters, as well as the gas-phase potential energy surface for proton transfer between the two water molecules in a Zundel ion, all calculated at the MP2/aug-cc-pVTZ level of theory. Combined QM/MM molecular dynamics simulations were carried out for bulk water and for a proton solvated in water using large cluster models. Both the authentic and reparameterized semiempirical models were employed in the simulations. The reparameterization led to significantly better results in all cases. The new set of OM3 parameters gave the best overall results for the structural and dynamic properties of water and the hydrated proton, with a small but finite barrier of 0.1−0.2 kcal/mol in the potential of mean force for proton transfer, in agreement with ab initio path-integral molecular dynamics simulations. The reparameterized OM3 model is expected to be useful for efficient modeling of proton transfer in aqueous solution.

1. INTRODUCTION

Transfer of the hydrated proton in bulk water is very important in many chemical and biological processes.[1] The proposed Grotthuss shuttling mechanism,[2] which involves reorganization of the covalent and hydrogen bonds over time, presents a challenge in molecular dynamics (MD) simulations. The published studies can be largely divided into four categories, according to how the potential energies were determined. The first category of studies employed classical force fields that were specially designed to describe water dissociation, for example, the central force field by Stillinger and David[3,4] and its modification by Halley et al.,[5] the protonizable water model by Billeter and van Gunsteren,[6] the polarizable and dissociable water potential by Lussetti et al.,[7] the dissociative water potential by Mahadevan and Garofalini,[8] the ReaxFF reactive force field by van Duin and co-workers,[9] the reactive molecular dynamics (RMD) approach by Selvan et al.,[10] and the LEWIS reaction force field by Herzfeld and co-workers.[11] The second group of studies were based on the so-called multistate empirical valence bond (MS-EVB) model,[12] which is an extension of Warshel's EVB model.[13] Several parametrized two-state EVB and MS-EVB models for proton transfer in water were proposed by Voth and co-workers,[12,14−17] by Vuilleumier and Borgis,[18,19] by Tuckerman and co-workers,[20,21] by Kornyshev et al.,[22] and by Paesani and co-workers.[23] The third approach made use of ab initio molecular dynamics (AIMD), in particular, Car−Parrinello molecular dynamics (CPMD),[24] simulations using density functional theory (DFT) to compute the potential on the fly. The first such studies by Tuckerman et al.[25,26] in 1994 were followed by many similar publications.[1,16,17,27−34] The fourth type of simulations employed semiempirical methods, in particular, the self-consistent-charge density functional tight-binding (SCC-DFTB)[35] approach; this includes the work by Choi and Jordan,[36] by Maupin et al.,[37] and by Cui and co-workers.[38,39] All these studies have provided valuable insight into the solvation structure and the transport mechanism of the hydrated proton. It is commonly agreed that the proton transfer occurs via an Eigen−Zundel−Eigen mechanism, where the Eigen ion is the resting state and the Zundel ion is the transition state. However, some groups have also advanced different views (see, for example, refs 27 and 33, as well as the discussion in ref 17). Furthermore, based on the analysis of ab initio MD simulations, Berkerbach et al.[34] proposed a concerted mechanism, in which the rate-limiting step to proton diffusion is the loss of an acceptor hydrogen bond at the proton-receiving water molecule with concomitant formation of a hydrogen bond at the lone-pair site of the hydronium ion.

Semiempirical quantum-chemical methods are computationally much more efficient than ab initio or DFT methods, making them attractive choices for MD simulations. However,

Received: March 20, 2013
Published: May 3, 2013

the existing standard semiempirical models are generally not accurate enough in describing interactions between water molecules, or proton transfer barriers, or both. This calls for a revision of these models as a prerequisite for using them in simulations of water and aqueous solutions. One option is to add empirical dispersion corrections and/or hydrogen-bonding corrections, which has been attempted, for example, by McNamara and Hillier,[40] Hobza and co-workers,[41] Maupin et al.,[37] and Korth.[42,43] Hydrogen-bonding corrections have recently also been included by Cui and co-workers[39] into a new SCC-DFTB model that features third-order terms in the Taylor expansion of the charge fluctuations.[44]

In this paper, we report a specific reaction parametrization (SRP)[45] of two sets of established semiempirical methods that targets proton transfer in water. The first set includes the popular Austin Model 1 (AM1) by Dewar et al.[46] and the Parameterized Model 3 (PM3) by Stewart,[47,48] which are based on the Modified Neglect of Differential Overlap (MNDO) approximation by Dewar and Thiel.[49] The second set covers the family of orthogonalization-corrected model OMx (x = 1, 2, and 3) developed by Thiel and co-workers.[50−54] In all cases, the theoretical framework of the underlying semiempirical method was retained (i.e., no additional terms were introduced). Our goal was to develop specific semiempirical models that can reproduce ab initio calculated potential surfaces with reasonable accuracy, in order to enable efficient simulations of proton transfer in water. The reference data and the parametrization process are described in Section 2. Section 3 discusses the results, and conclusions are drawn in Section 4.

2. COMPUTATIONAL DETAILS

2.A. Parameterization. For the parametrization, we employed two groups of reference data calculated at the MP2 level of theory[55] with the aug-cc-pVTZ[56,57] basis set. The first group consisted of the fully optimized gas-phase geometries and the associated energies for water H_2O, water clusters $H_{2n}O_n$ (n = 2−5), the hydronium ion H_3O^+, and protonated water clusters $H_{2m+1}O_m^+$ (m = 2−4). The model systems are displayed in Figure 1; entries (a)−(g) are included in the training set and entries (h)−(j) are part of the validation set. For large clusters such as the water tetramer and pentamer, which have many close-lying local minima, we only selected the global minimum, with the initial geometries for optimization being taken from refs 58 and 59. We also computed the vibrational frequencies for these model systems as part of our assessment of the accuracy, but we did not attempt to reproduce these frequencies during the parametrization. The second group of reference data consisted of 99 data points on the two-dimensional potential energy surface for the transfer of one proton between two water molecules within the Zundel ion in the gas phase, computed by relaxed surface scans (Figure 2). In these surface scans, the distance between the oxygen atoms (R_{OO}) and the distance between the migrating proton and one oxygen atom were kept fixed in partial geometry optimizations. The 53 data points obtained for R_{OO} = 2.2, 2.4, 2.6, and 2.8 Å were included in the training set, whereas the other 46 data points for R_{OO} = 2.3, 2.5, 2.7, and 2.9 Å were employed for validation. Please note that the 99 data points are not uniformly distributed.

Semiempirical methods have traditionally been parametrized to reproduce experimental heats (enthalpies) of formation. The experimental heat of formation for water (−57.80 kcal/mol)[60] thus serves as the reference energy for water, i.e., entry (a) in Figure 1. The target energies for all other systems are the corresponding ab initio binding energies. In the case of the water dimer (H_4O_2), the semiempirically computed energies (E^{SE}) should thus reproduce as closely as possible the binding energy calculated from their ab initio counterparts (E^{AI}), with the goal being

$$E^{SE}(H_4O_2) - 2E^{SE}(H_2O) = E^{AI}(H_4O_2) - 2E^{AI}(H_2O) \quad (1)$$

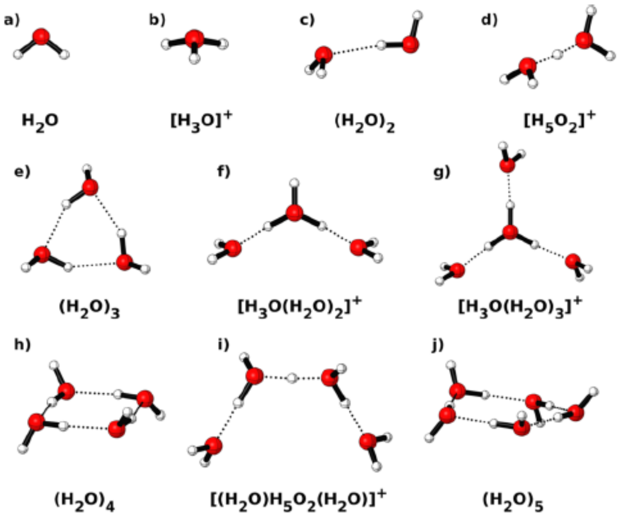


Figure 1. Neutral water clusters and protonated water clusters used for parametrization; entries (a)−(g) belong to the training set, and entries (h)−(j) are part of the validation set. Geometries were optimized at the MP2/aug-cc-pVTZ level.

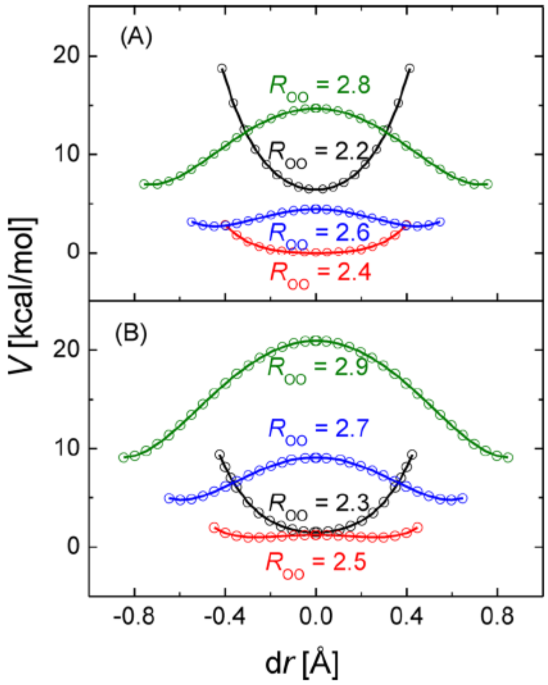


Figure 2. Reference MP2/aug-cc-pVTZ gas-phase energies (V) for proton transfer in the Zundel ion at fixed distances R_{OO} (in Ångstroms) obtained from relaxed surface scans ((A) training set and (B) validation set). Here, dr is the difference in the distances between the migrating proton and the two O atoms.

This equation can trivially be generalized to water *n*-mers. For the protonated water clusters, the semiempirical energy of the proton [i.e., experimental (or accurate) heat of formation for proton $\Delta H(H^+)$] is needed to compute the relative energies. In the case of the hydronium ion (H_3O^+), the target is

$$E^{SE}(H_3O^+) - E^{SE}(H_2O) - \Delta H(H^+) = E^{AI}(H_3O^+) - E^{AI}(H_2O) - E^{AI}(H^+) \quad (2)$$

where the electronic energy of an isolated proton $E^{AI}(H^+)$ is, by definition, zero. As is common in semiempirical work, we have used the value of 365.7 kcal/mol from ref 61 for $\Delta H(H^+)$. The target energies for the other protonated water clusters can be derived in the same way.

The ab initio reference calculations were carried out by using the *Gaussian09* package.[62] The semiempirical calculations and parametrizations were performed by using the MNDO package,[63] except for the PM6 calculations,[64] which were determined using *Gaussian09*. Reparameterizations were attempted for AM1, PM3, OM1, OM2, and OM3, and the resulting new models are denoted AM1*n*, PM3*n*, OM1*n*, OM2*n*, and OM3*n*, respectively. The following weights were employed in evaluating the error function for the parametrization: 10.0 $(\text{kcal/mol})^{-1}$ for energy, 100.0 $Å^{-1}$ for distances, and 5.0 deg^{-1} for angles. We did not perform any extensive search for global minima in parameter space, because we wanted to keep the new parameters as close to the authentic parameters as possible.

2.B. Tests and Validation. The reparameterized semiempirical models were tested by doing calculations on a series of model systems. First, we examined the gas-phase binding energy profiles for two water molecules as a function of the distance between the two O atoms. The ab initio reference curve for the binding energy was computed at the MP2/aug-cc-pVTZ level by relaxed surface scans using *Gaussian09*,[62] in which the distance between the O atoms was changed by 0.1 Å from the previous geometry and then fixed during geometry optimization. The relaxed surface scan started from the fully optimized water dimer geometry and moved in both directions, toward longer and shorter O—O distances. The MP2 optimized geometries along this path were then adopted as input geometries for the semiempirical calculations using the MNDO program.[63]

Second, we carried out MD simulations for bulk water and for a hydrated proton in bulk water using large cluster models. Figure 3 depicts the model system for the hydrated proton. The entire system had a radius of R_{system} = 22 Å, containing 1482 water molecules and one hydronium ion initially placed at the center. The density was 1.00 g/mL. The system was divided into two regions: an inner core ($R_{inner} \leq 12$ Å) containing 240 water molecules plus the hydronium ion and an outer layer. The water molecules in the outer layer were subjected to harmonic potentials with force constants of 10 kcal mol^{-1} $Å^{-2}$ that restrained the atoms to their original positions and effectively prevented the water molecules from evaporating into the vacuum. The cluster model for neutral bulk water simulations was obtained by deprotonation of the hydronium ion.

Regarding the update of the hydronium O atom index, we have adopted the scheme proposed by Hofer et al.[65] As illustrated in Figure 4A, the donor (hydronium oxygen) is labeled O0. Any water oxygen atom ($O1_i$) within 2.9 Å from O0 will be enlisted as a potential acceptor; the number of potential acceptors varies over time during the MD simulations and can be larger than 3. For each potential acceptor $O1_i$ and each hydronium hydrogen H_j (j = 1, 2, and 3), a parallel vector component $\mathbf{R}^{//i}_{O0Hj}$ is computed by projecting the donor-

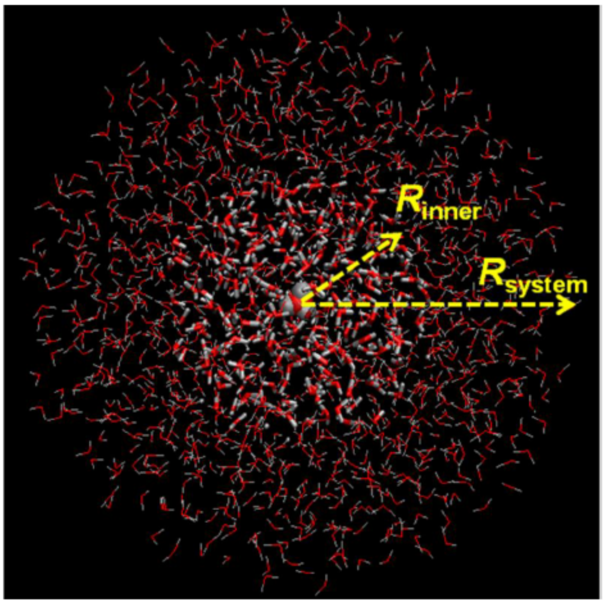


Figure 3. Cluster model for dynamics simulations of protonated water. The entire system has a radius of R_{system} = 22 Å, containing 1482 water molecules and one hydronium ion initially placed at the center. It is divided into an inner core ($R_{inner} \leq 12$ Å) containing 213 water molecules (licorice) plus the hydronium ion (van der Waals balls) and an outer layer with the other water molecules (lines). (See Section 2.B for further details.) The cluster model for the charge-neutral water simulations was generated by deprotonating the hydronium ion.

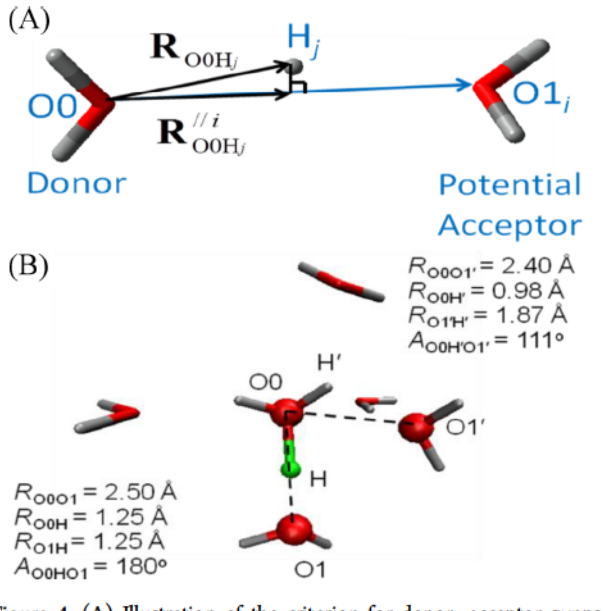


Figure 4. (A) Illustration of the criterion for donor–acceptor swaps during proton migration. (See Section 2.B for a detailed description.) (B) A snapshot of a donor–acceptor swap from an OM3*n* trajectory. H (in green) is the proton being transferred between O0 and O1, while H′ is the hydrogen between O0 and the nearest potential acceptor O1′. Note that the acceptor O1 is further away from O0 than O1′ is from O0.

Table 1. Parameters Re-optimized in This Work for the MNDO-Based Methods AM1 and PM3[a]

	AM1		PM3	
parameter	H	O	H	O
U_{ss} (eV)	−11.396 248 28	−97.830 051 00	−13.072 844 90	−86.946 179 86
U_{pp} (eV)		−78.262 408 38		−71.908 402 68
ζ_s (au)	1.188 187 56	3.107 792 05	0.954 530 51	3.757 723 92
ζ_p (au)		2.523 560 51		2.404 027 00
β_s (eV)	−6.173 824 58	−29.272 769 70	−5.618 505 74	−45.206 662 11
β_p (eV)		−29.272 857 32		−24.733 236 26
α (Å^{-1})	2.882 439 11	4.455 399 50	3.374 889 68	3.219 112 53
G_{ss} (eV)	12.847 854 58	15.419 913 68	14.809 518 87	15.777 238 66
G_{pp} (eV)		14.520 099 32		13.598 867 86
G_{sp} (eV)		14.479 690 53		10.621 326 89
G_{p^2} (eV)		12.979 865 09		12.449 747 02
H_{sp} (eV)		3.939 834 42		0.578 426 14
K_1	0.122 538 40	0.280 798 02	1.047 892 68	−1.074 471 74
L_1	4.999 771 05	5.000 022 09	5.142 074 74	5.949 950 70
M_1	1.199 850 35	0.847 737 05	1.556 607 86	1.600 454 10
K_2	0.004 892 58	0.081 222 13	−1.022 154 81	1.109 274 09
L_2	4.999 919 96	7.000 037 71	5.984 287 97	5.926 025 97
M_2	1.800 222 36	1.444 973 83	1.563 919 61	1.598 900 68
K_3	−0.018 442 33			
L_3	1.999 842 70			
M_3	2.100 114 10			

[a] See Table S1 in the Supporting Information for the description of the parameters.

hydrogen vector $\mathrm{R}_{\mathrm{O0H}j}$ onto the donor-potential acceptor vector $\mathrm{R}_{\mathrm{O0O1}i}$. The ratio ρ_{ji} is defined as follows:

$$\rho_{ji} = \begin{cases} \dfrac{|\mathbf{R}^{//i}_{\mathrm{O0H}_j}|}{|\mathbf{R}_{\mathrm{O0O1}_i}|} & \text{if } \mathbf{R}^{//i}_{\mathrm{O0H}_j} \cdot \mathbf{R}_{\mathrm{O0O1}_i} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The ratio ρ_{ji} is a measure of how likely H_j is going to be transferred to $\mathrm{O1}_i$ (the larger positive the ρ_{ji} value, the more likely the transfer will be). At any given time step during the simulation, O1 is identified as the potential acceptor with the largest ρ_{ji} value, which is labeled as ρ. The donor and acceptor are swapped if $\rho > 0.5$. Please note that O1 is not necessarily the potential acceptor that is the closest to the donor O0, which is labeled as O1′, although O1 and O1′ are identical most of the time (>97%) during our MD simulations. Figure 4B shows a snapshot taken from an OM3*n* trajectory where O1 and O1′ have different identities. In this case, the acceptor O1 is further away from O0 than the nearest potential acceptor O1′, but the geometry clearly favors the proton transfer between O0 and O1.

Because of the large number of atoms in the above cluster models, it is computationally quite expensive to simulate the entire system even at the semiempirical level. We thus carried out the MD simulations at the combined quantum-mechanics/molecular-mechanics (QM/MM)[66−73] level. The inner core was described by semiempirical quantum chemistry, and the outer layer by the SPC[74] water potential. For the sake of simplicity, the mechanical embedding scheme[75] was adopted. Both the authentic and reparameterized semiempirical models were applied to the simulations of the charge-neutral water system, but only the reparameterized models were employed for simulating the hydrated proton. In the OM2 and OM2*n* simulations, we encountered self-consistent-field (SCF) convergence problems quite often, so we decided not to pursue them further; hence, we do not present OM2 and OM2*n* results for the bulk systems. The MD simulations were performed using the QMMM[76] program, which calls the MNDO[63] program for QM calculations and TINKER[77] for MM calculations. They were carried out in the canonical (*NVT*) ensemble at a temperature of 298.15 K, using a Nosé-Hoover thermostat.[78,79] For a given semiempirical model, 10 trajectories were propagated independently with a step size of 1 fs, each consisting of 2 ps equilibration followed by a 20 ps productive run. The total time length of the productive trajectories was 200 ps for each semiempirical model. For the bulk water simulations, the geometry of the entire system was saved every 50 steps. In the hydrated proton case, the geometry was saved every 10 steps, and the information about the donor and potential acceptors was recorded every step.

To limit the boundary effects due to the finite size of the model systems, the radial distribution functions (RDFs) for the water molecules were computed for an "elite" group of selected water molecules, more specifically, the 21 water molecules that were initially located within 5 Å from the center of the cluster. If an elite water molecule drifted away from the center of mass of the entire system by more than 5 Å at a given time step, the saved geometry of that snapshot was excluded from the corresponding RDF analysis. This RDF analysis procedure was first tested in two calculations at the MM level, where the entire system was described by the SPC[74] and TIP3P[80] water potentials, respectively. In either test, the RDF computed for the above cluster model was compared with the RDF obtained from 200 ps *NVT* simulations at the same temperature using periodic boundary conditions (PBC) with 216 water molecules in a cubic box of 18.63 Å length. Inspection of the cluster model-based RDF and the PBC-based RDF (see Figures S1 and S2 in the Supporting Information) revealed only minor and insignificant differences, which suggests that our approach is sufficiently accurate. For the RDF calculations of the hydrated proton, the criterion to include a snapshot was that the

Table 2. Parameters Re-optimized in This Work for the Orthogonalization-Corrected Models OM1, OM2, and OM3[a]

	OM1		OM2		OM3	
parameter	H	O	H	O	H	O
U_{ss} (eV)	−12.838 184 53	−93.042 206 09	−12.560 935 10	−101.852 535 97	−12.484 287 97	−105.829 748 09
U_{pp} (eV)		−77.598 439 87		−79.043 912 84		−78.874 514 70
ζ (au)	1.209 193 33	1.102 381 11	1.558 958 16	1.396 722 13	1.285 158 34	1.313 975 69
β_s (eV bohr$^{-1/2}$)	−4.892 972 58	−6.222 354 84	−3.313 643 01	−10.761 940 01	−3.358 940 34	−14.461 600 14
β_p (eV bohr$^{-1/2}$)		−9.940 330 68		−8.570 870 01		−8.741 752 81
β_π (eV bohr$^{-1/2}$)		−11.293 709 59		−9.434 719 22		−12.911 417 67
β_s(X−H) (eV bohr$^{-1/2}$)		−6.459 788 22		−6.640 584 53		−13.583 127 26
β_p(X−H) (eV bohr$^{-1/2}$)		−12.474 169 74		−10.192 763 80		−9.409 924 53
α_s (au)	0.096 316 41	0.108 573 45	0.098 526 80	0.111 786 91	0.060 830 99	0.084 727 80
α_p (au)		0.096 425 42		0.085 136 94		0.078 303 72
α_π (au)		0.152 820 61		0.166 179 91		0.143 615 89
α_s(X−H) (au)		0.079 434 49		0.075 948 50		0.173 009 39
α_p(X−H) (au)		0.129 291 89		0.083 188 46		0.117 783 94
G_{ss} (eV)	12.847 829 62	15.420 763 38	12.803 449 02	15.385 277 00	12.845 336 07	15.369 303 75
G_{pp} (eV)		14.520 494 26		14.547 376 40		14.578 254 22
G_{sp} (eV)		14.480 064 08		14.512 376 78		14.445 520 26
G_{p^2} (eV)		12.980 302 07		12.867 807 46		13.015 101 93
H_{sp} (eV)		3.940 777 43		3.992 884 32		3.950 34628
F_1	0.541 034 13	0.681 430 71	0.389 765 28	1.156 680 58	0.349 155 26	0.535 902 15
F_2	0.846 677 03	0.476 399 46	1.527 086 00	1.089 211 84		
G_1			0.660 531 90	0.262 691 92	0.333 525 83	0.052 674 67
G_2			0.915 091 13	0.869 410 78		

[a] See Table S1 in the Supporting Information for the description of the parameters.

hydronium O atom must be within 5 Å of the center of the cluster; it turned out that all saved snapshots satisfied this requirement and could be used. The integrated coordination number (ICN) for two types of atoms A and B: $n_{AB}(r) = 4\pi\int_0^r g_{AB}(r')r'^2\, dr'$ was computed in a similar way. Again, as can be seen from Figures S16(A) and S17(A) in the Supporting Information, we have obtained essentially identical plots of the oxygen−oxygen ICN $n_{OO}(r)$ and the oxygen−hydrogen ICN $n_{OH}(r)$ for water molecules in the simulations with PBC and with cluster models.

In this work, the nuclear degrees of freedom were treated classically. Therefore, quantum effects on the nuclear motion were not included. The zero-point vibrational corrections would effectively lower the free-energy barrier for proton transfer, but the change is known to be small (~0.5 kcal/mol),[1] smaller than the root-mean-square deviation (RMSD) between the potential energy surfaces from the semiempirical and ab initio calculations. Tunneling effects are insignificant for proton transfer in bulk water,[1,17] as expected from the low free-energy barriers. Therefore, a classical treatment of nuclear motion was considered adequate for our purposes.

3. RESULTS AND DISCUSSION

3.A. Parameters. The OM*x* Hamiltonian differs from the MNDO-type Hamiltonian used in AM1 and PM3 by including orthogonalization corrections that account for Pauli exchange repulsions.[81] These corrections are applied to the one-center Fock matrix elements in OM1, and to all Fock matrix elements in OM2 and OM3. The corrections in OM2 are truncated in OM3 for computational efficiency without losing much of the accuracy. Moreover, while AM1 and PM3 attempt to use flexible core repulsion functions to model the interactions in weakly bound systems, the OM*x* methods put emphasis on the fine-tuning of the resonance integrals that represent the major bonding interactions. The optimized new parameters for hydrogen and oxygen are tabulated in Table 1 for the MNDO-based methods AM1 and PM3, and in Table 2 for the OM*x* models. A brief description of these parameters is given in Table S1 in the Supporting Information. The new values of the parameters are generally quite close to the original values: the maximum relative changes are <8% for AM1*n*, PM3*n*, and OM1*n*, and they mostly remain below 10% for OM2*n* and OM3*n* (except for a few larger changes in the parameters entering the resonance integrals and the prefactors of the orthogonalization terms). Therefore, the reparameterization should be regarded as a fine tuning of the investigated semiempirical methods (rather than a drastic modification).

3.B. Training and Validation Sets. The overall performance of the authentic parameters is documented in Table 3 by the RMSDs of the energies, geometries, and vibrational frequencies for the training set, using the ab initio results as the reference. We are particularly interested in the cluster binding energy (V), the two-dimensional potential surface for proton transfer (V^{PES}), and the hydrogen-bonding angle (θ_H) (which is important because it indicates the orientation of the water molecules in the hydrogen-bonding network). Obviously, the orthogonalization-corrected models outperform the MNDO-based methods in nearly all aspects. While the OM*x* models reproduce the reference data for the cluster binding energies V within 6 kcal/mol, the MNDO-based methods yield significantly larger errors ranging from 12 kcal/mol to 33 kcal/mol. A similar performance is observed for V^{PES}. In the case of θ_H, AM1 and PM6 show very large deviations (RMSDs of 48° and 33°, respectively), implying an erroneous orientation of the water molecules in the hydrogen-bonding network. OM1 (RMSD = 21°) and PM3 (RMSD = 13°) perform somewhat better, although not satisfactorily. The OM2 and OM3 results are best (RMSD < 5°). To exemplify the significance of θ_H, we superimpose the reference MP2 geometry for the Eigen cation with those obtained from different semiempirical methods in

Table 3. Root-Mean-Square Deviations (RMSDs) of the Standard Semiempirical Methods for the Training Set[a]

	MNDO-based			Orthogonalization-corrected		
	AM1	PM3	PM6	OM1	OM2	OM3
V (kcal/mol)	12.5	32.5	16.6	5.5	5.8	4.1
r (Å)	0.01	0.01	0.03	0.01	0.01	0.05
r_H (Å)	0.36	0.15	0.10	0.22	0.15	0.23
θ (deg)	3.6	2.9	7.0	1.6	2.2	1.5
θ_H (deg)	47.6	13.1	32.8	20.9	4.6	4.2
$\nu(\text{cm}^{-1})$[b]	235	204	695	116	372	229
$n_{i\nu}$	3	1	1	3	1	1
V^{PES} (kcal/mol)	19.7	39.2	15.8	3.9	8.5	5.4

[a] The training set consists of 7 potential energies (V), 35 bond lengths (r), 11 hydrogen-bond lengths (r_H), 22 bond angles (θ), 16 hydrogen-bond angles (θ_H), 114 frequencies (ν), and the number of imaginary frequencies ($n_{i\nu}$) in the water clusters and charged water clusters [model systems (a)−(g) in Figure 1]. It also includes 53 data points of the two-dimensional potential energy surface (V^{PES}) for proton transfer in the Zundel ion at fixed R_{OO} distances (see Figure 2A). See Section 2 for computational details. [b] The total number of reference frequencies is 114; any imaginary frequencies are excluded from the statistics.

Figure S3 of the Supporting Information. The MP2 calculations yield θ_H = 175.2°. The MNDO-based methods, AM1 (θ_H = 100.7°), PM3 (θ_H = 150.2°), and PM6 (θ_H = 136.7°), produce qualitatively wrong orientations of the coordinating water molecules with respect to the central hydronium ion, whereas the OM*x* methods predict much more realistic angles θ_H of 172.9°, 168.9°, and 175.0° for OM1, OM2, and OM3, respectively.

How do the reparameterized methods perform? First, we look at the training set (rows 4−11 of Table 4). The reparameterization leads to substantial improvements in the energies for PM3*n*, OM2*n*, and OM3*n*, but only slight changes for AM1*n* or OM1*n*. The most impressive advances are observed for PM3*n*. Overall, OM2*n* and OM3*n* clearly show the best performance, achieving excellent agreement with the ab initio reference data for V (RMSD = 0.5 and 2.0 kcal/mol, respectively) and V^{PES} (RMSD = 0.5 and 0.7 kcal/mol, respectively). In terms of the hydrogen-binding geometries, OM2*n* and OM3*n* are again superior with small RMSD values for θ_H (2.8° and 4.1°, respectively). Again using the Eigen cation as an example (see Figure S3 in the Supporting Information), AM1*n* still gives an erroneous hydrogen bonding angle (θ_H = 101.8°), the PM3*n* value (θ_H = 167.4°) underestimates the MP2 value by 7.8°, and the reparameterized OM*x* models reproduce the MP2 angle within 2.6°.

Next, we turn to the validation set (see rows 13−18 in Table 4; optimized geometries for entries (h)−(j) are given in Figures S4−S6 in the Supporting Information). Overall, each of the reparameterized semiempirical models shows a rather similar performance for the validation set and the training set. However, two exceptions should be noted. First, PM3*n* achieves a very small RMSD value for θ_H (3.7°), indicating excellent hydrogen-bonding geometries in the validation set. Second, OM1*n* gives much larger RMSD values for V and θ_H due to poor geometries for the water tetramer and pentamer, which maintain the cyclic structure, but adopt hydrogen-bond orientations that deviate substantially from the MP2 geometry. The OM1*n* geometry for the Zundel ion complex with two water molecules is still in good agreement with the ab initio result.

Table 4. Root-Mean-Square Deviations (RMSD) of the Reparameterized Semiempirical Methods for the Training and Validation Sets[a]

	MNDO-based		Orthogonalization-corrected		
	AM1*n*	PM3*n*	OM1*n*	OM2*n*	OM3*n*
		Training Set[b]			
V (kcal/mol)	11.6	7.4	5.1	0.5	2.0
r (Å)	0.01	0.01	0.01	0.01	0.02
r_H (Å)	0.36	0.38	0.21	0.04	0.04
θ (deg)	3.7	3.8	1.5	0.9	1.3
θ_H (deg)	47.4	15.4	13.6	2.8	4.1
ν (cm^{-1})[c]	243	145	119	383	245
n_ν	4	3	5	2	2
V^{PES} (kcal/mol)	18.4	1.0	3.9	0.5	0.7
		Validation Set[d]			
V (kcal/mol)	7.1	8.3	13.5	0.2	1.6
r (Å)	0.02	0.01	0.02	0.01	0.02
r_H (Å)	0.66	0.11	0.41	0.02	0.01
θ (deg)	2.1	1.3	2.8	0.7	0.4
θ_H (deg)	83.7	3.7	49.7	7.3	6.4
V^{PES} (kcal/mol)	19.3	0.9	3.3	0.8	0.6

[a] See Section 2 for computational details. [b] See footnote (a) of Table 3. [c] See footnote (b) of Table 3. [d] The validation set consists of 3 potential energies (V), 26 bond lengths (r), 13 hydrogen-bond lengths (r_H), 13 bond angles (θ), 12 hydrogen-bond angles (θ_H) of the water clusters and charged water clusters (model systems (h)−(j) in Figure 1). It also includes 46 data points of the two-dimensional potential energy surface (V^{PES}) for proton transfer in the Zundel ion at fixed R_{OO} distances (see Figure 2B).

Finally, we take a look at the two-dimensional (2D) potential surfaces for proton transfer within a Zundel ion in the gas phase computed by the reparameterized methods (Figure 5). The reference MP2 surface clearly shows a well at (R_{OO}, dr) ≈ (2.4 Å, 0), which corresponds to the symmetric Zundel ion. Here, dr is the difference in the distances from the migrating proton to either O atom. The AM1*n* surface differs from the reference MP2 surface significantly, displaying a well in the upper right corner that is deeper than the central well of the symmetric Zundel ion; this suggests that the lowest-energy AM1n structure of the Zundel ion is very asymmetric (also see Figure S5(B) in the Supporting Information), and should perhaps better be regarded as a water molecule that is hydrogen-bonded to a hydronium ion. Because hydrogen moves much faster than oxygen, due to the much smaller mass, proton transfers can be approximated by movements along vertical lines on the two-dimensional surfaces shown in Figure 5. Since the asymmetric AM1*n* well is located at R_{OO} ≥ 2.8 Å, the barrier for proton transfer will be quite high (>6 kcal/mol). The PM3*n* surface shows a narrow valley extending from (R_{OO}, dr) ≈ (2.36 Å, 0) to (R_{OO}, dr) ≈ (2.44 Å, 0.3 Å). Such a surface is not optimal for proton transfer, although the RMSD value in V^{PES} is rather small (<1 kcal/mol). By contrast, all OM*xn* surfaces resemble the reference MP2 surface in giving a central well for the symmetric Zundel ion. The OM2*n* surface seems to agree best with the MP2 surface, followed by the OM3*n* surface; both show a central well near (R_{OO}, dr) ≈ (2.41 Å, 0). The OM1*n* surface has a central well that is somewhat displaced at (R_{OO}, dr) ≈ (2.34 Å, 0). Interestingly, the OM3*n* surface, and to a slightly lesser extent also the OM2*n* surface, resemble the

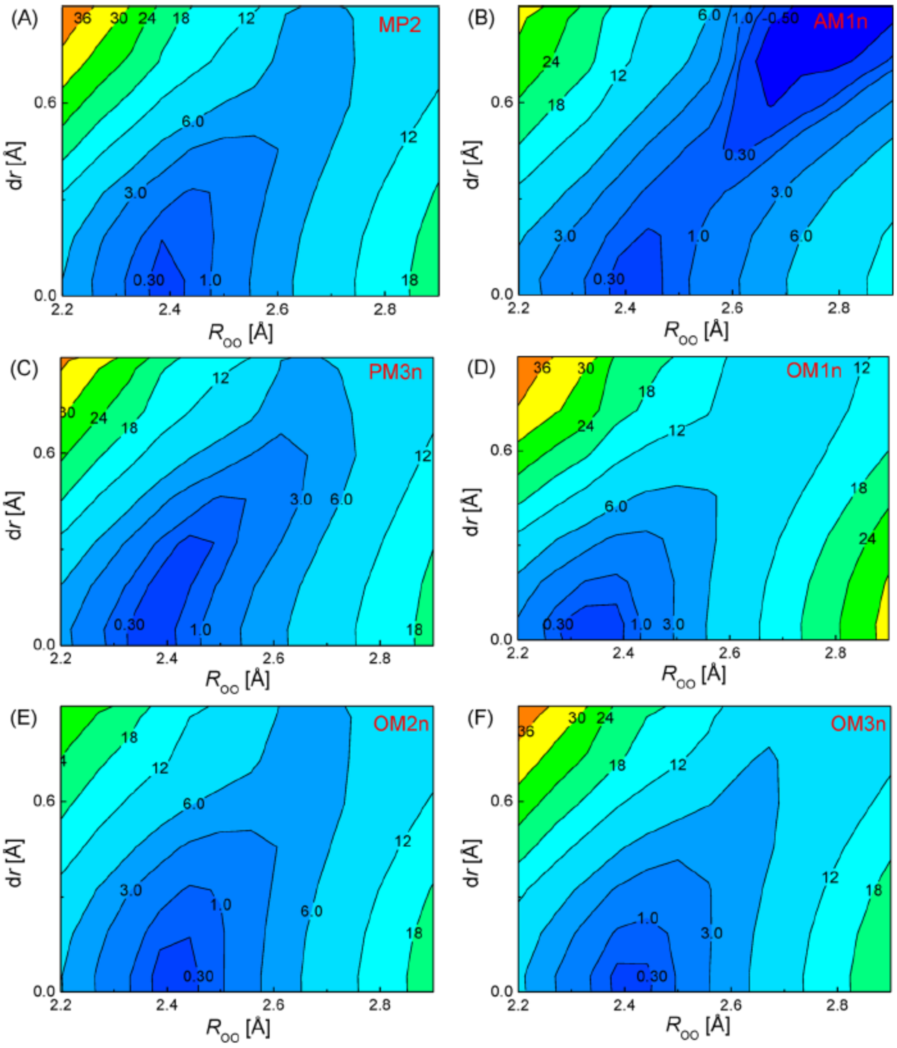


Figure 5. Two-dimensional gas-phase energy surfaces for proton transfer in the Zundel ion at fixed distances R_{OO} (in Ångstroms) obtained from (A) MP2/aug-cc-pVTZ and (B−F) the reparameterized semiempirical methods. Here, *dr* is the difference in the distances between the migrating proton and the two O atoms.

effective ab initio path-integral surface for the Zundel ion calculated by Brancato and Tuckerman (see Figure 1B in ref 21), who have parametrized an MS-EVB model on the basis of the path-integral surface, so that quantum effects such as the zero-point energy are implicitly included.

3.C. Binding Energy Profile of the Water Dimer. The gas-phase binding energy curves of the water dimer are plotted in Figure 6. Panel (A) shows those computed with the authentic parameters. First, we note that, at short distances R_{OO}, all methods including MP2 experience difficulties in getting smooth curves, because of changes in the relative orientation of the water molecules. Taking MP2 as an example, at $R_{OO} \geq 2.6$ Å, the first water molecule lies in a plane that bisects the HOH angle of the second water molecule, whereas at $R_{OO} \leq 2.5$ Å, both molecules are in the same plane (Figure S12 in the Supporting Information). This change in relative orientation only gives rise to a small bump in the MP2 energy profile, but leads to larger kinks in the curves computed semiempirically. Second, we note that all semiempirical methods except AM1 predict water dimers that are underbound, and the optimal distances between the two O atoms are shorter than the MP2 value. At large distances R_{OO}, all semiempirical models, including AM1, underestimate the binding energy, presumably because of the lack of explicit dispersion terms. Interestingly, the OM*x* curves seem to converge faster to the MP2 curve than the MNDO-based curves do when the water molecules are gradually taken apart. Adding empirical dispersion corrections may help to improve the agreement at large distances.[54,82,83] However, dispersion interactions were not deemed to be the main problem in the parametrization of the energy surface for proton transfer, where the two water molecules within the Zundel ion are in close proximity, so that electrostatic interactions are expected to be dominant. Therefore, we have not included empirical dispersion corrections in this work.

The energy profiles computed with the reparameterized methods are depicted in panel (B) in Figure 6. While the AM1*n* curve has changed little from AM1, the PM3*n* curve has changed remarkably from PM3. The binding well around R_{OO} = 2.8 Å in PM3 moves to R_{OO} = 3.1 Å in PM3*n*, and moreover becomes very shallow, with the binding energy decreased from 4 kcal/mol to 2 kcal/mol. The OM1*n* curve looks very similar

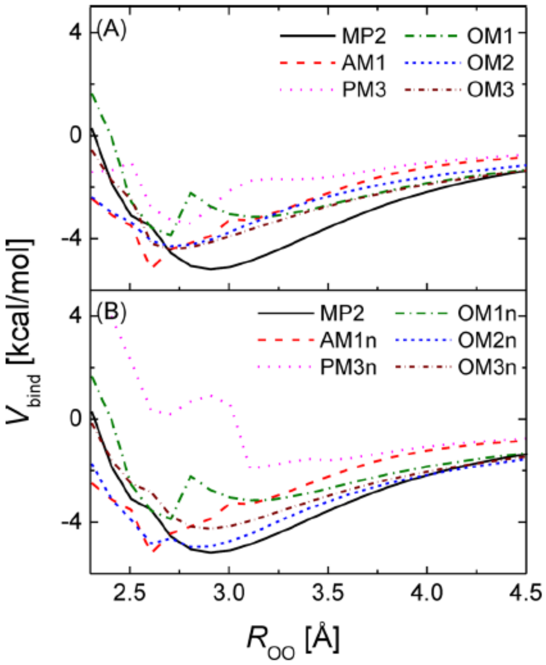


Figure 6. Gas-phase binding energy profiles of two water molecules computed with the (A) authentic and (B) reparameterized semi-empirical methods. The solid reference curve was obtained at the MP2/aug-cc-pVTZ level from relaxed surface scans for the water dimer as a function of distance between the two O atoms (R_{OO}), in steps of 0.1 Å. The zero of energy corresponds to two water molecules at infinite distance: $V_{bind} = V_{(H_2O)_2} - 2V_{H_2O}$.

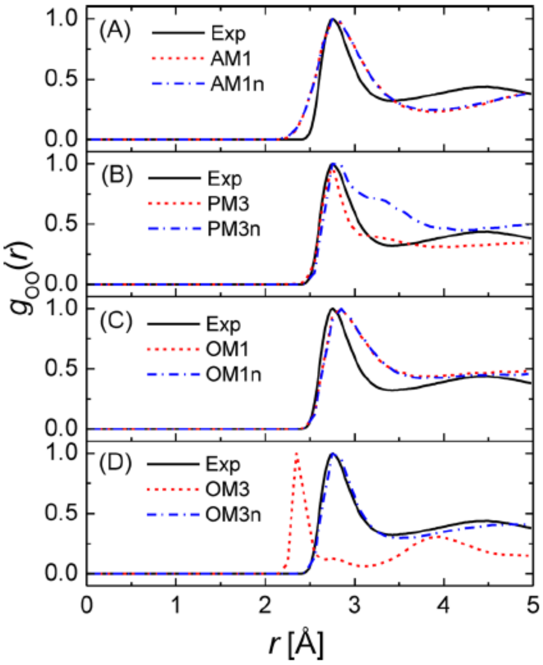


Figure 7. Radial distribution functions $g_{OO}(r)$ for water computed for (A) AM1 and AM1*n*, (B) PM3 and PM3*n*, (C) OM1 and OM1*n*, and (D) OM3 and OM3*n*. The experimental data (denoted by "Exp") is plotted in all panels for comparison. The AM1 and OM1 curves are directly beneath the AM1*n* and OM1*n* curves, respectively. All curves have been scaled such that the height of the first peak is 1.00.

to the OM1 curve. The OM2*n* curve shows the best agreement with the MP2 curve, followed by the OM3*n* curve. Compared with OM2 and OM3, the improvements in OM2*n* and OM3*n* are most encouraging at larger distances (R_{OO} > 2.8 Å), which further alleviates the need to include empirical dispersion corrections.

3.D. Properties of Water. The oxygen−oxygen and oxygen−hydrogen RDF obtained from the bulk water simulations are plotted in Figures 7 and 8, respectively, together with the experimental curves.[84,85] First, let us look at the oxygen−oxygen RDF. Both AM1 and AM1*n* give a first (broad) peak near *r* = 2.8 Å, but fail to predict the second peak at *r* ≈ 4.5 Å; the minimum between the first and second peaks occurs at *r* ≈ 3.9 Å, which is a larger distance than that found experimentally (*r* = 3.4 Å). The PM3 curve shows the first peak near *r* = 2.8 Å, with a small shoulder between *r* = 3.0 Å and *r* = 3.5 Å. In the PM3*n* curve, the shoulder becomes more prominent. Neither PM3 nor PM3*n* predict the second peak correctly. The OM1 and OM1*n* RDF are similar, with the first peak at *r* = 2.9 Å and the second peaks not seen. The OM3 curve has a first sharp peak at a distance that is much too short (*r* = 2.4 Å), but the reparameterization improves the RDF dramatically: the OM3*n* curve shows the best agreement with the experimental RDF, although the second peak is still slightly off (by <0.3 Å). The difficulties in reproducing the second peak with AM1, PM3, and OM1 (and their reparameterized counterparts) are likely due to the problem that they all underestimate the binding energies between water molecules (see Figure 6).

The first-solvation-shell coordination numbers were estimated by $n_{OO}(r)$ at the *r* values that correspond to the first

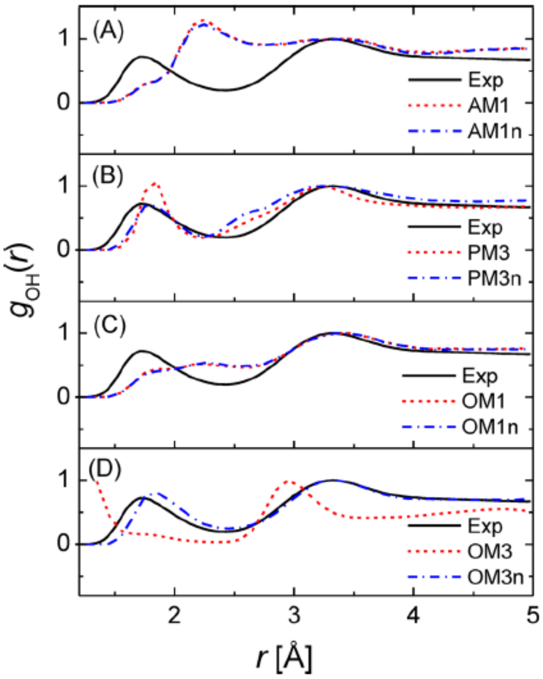


Figure 8. Radial distribution functions $g_{OH}(r)$ for water computed for (A) AM1 and AM1*n*, (B) PM3 and PM3*n*, (C) OM1 and OM1*n*, and (D) OM3 and OM3*n*. The experimental data (denoted by "Exp") is plotted in all panels for comparison. The AM1 and OM1 curves are right beneath the AM1*n* and OM1*n* curves, respectively. All curves have been scaled such that the height of the peak at ~3 Å is 1.00.

minima of $g_{OO}(r)$, and the results echo the above RDF analysis. The first-solvation-shell coordination numbers are tabulated in the second column of Table 5, and the $n_{OO}(r)$ plots are given in panels (B) and (C) of Figure S16 in the Supporting Information. Compared with the coordination number value of 4.7 determined experimentally,[84−86] it is apparent that the MNDO-based methods overestimate the number of coordinating water molecules due to the erroneous hydrogen-bonding pattern. Better performance is achieved by the orthogonalization-corrected methods, because of their improved hydrogen-bonding description. In particular, OM3*n* yields a coordination number of 4.9 that is in excellent agreement with the experiments.

Table 5. Integrated Coordination Numbers of the First Solvation Shell for Water and for Hydronium from Semiempirical MD Simulations[a]

	Integrated Coordination Number, ICN	
	water[b]	hydronium[c]
AM1	8.7	n/a
AM1*n*	9.4	10.4
PM3	8.0	n/a
PM3*n*	7.7	7.9[d]
OM1	6.4	n/a
OM1*n*	6.7	7.8[e]
OM3	3.7	n/a
OM3*n*	4.9	6.5

[a] ICNs at the distances corresponding to the first minima in $g_{OO}(r)$ for water and in $g_{O0O}(r)$ for hydronium, unless otherwise indicated. [b] 4.7 in the experiments[84−86] and 4.1 in ab initio MD simulations.[37] [c] 3.0 in ab initio MD simulations.[37] [d] Including the peak at r = 3.0 Å (otherwise, ICN = 3.3). [e] Integration over both peaks in the bimodal distribution.

Turning to the oxygen−hydrogen RDF, we find that none of original semiempirical methods with authentic parameters reproduces the experimental curve correctly. PM3 performs best, predicting two peaks at r = 1.9 Å and 3.3 Å that are rather close to the experimental ones at r = 1.7 Å and 3.3 Å, respectively; however, the first PM3 peak is much too narrow. Reparameterization again leads to dramatic improvements in the OM3 case: the OM3*n* RDF matches the experimental curve very well, although the position of the first peak deviates slightly, by ∼0.1 Å. There are also improvements in the PM3 case, leading to lower and broader peaks in the PM3n curve, but the agreement is not as good as for OM3*n*. The AM1 and OM1 RDF are not affected much by the reparameterization. Taken together, the above results clearly show that OM3*n* offers the best structural properties for bulk water.

3.E. Structural Properties of the Hydrated Proton. The RDF for water O atoms around the hydronium O atom (O0) are plotted in Figure 9. A bimodal distribution is found with OM1*n*: a narrower peak near r = 2.4 Å and a wider peak at $r \approx$ 2.6 Å. Maupin et al.[37] observed similar bimodal distributions in their SCC-DFTB simulations and interpreted them in terms of a prominent Zundel character for the solvated proton, with one of the three coordinating water molecules sustainably closer to the hydronium ion than the other two. Indeed, integration over the first peak in our OM1*n* plot yielded a coordination number of 1.4. This bimodal distribution is not seen in experiments or ab initio MD simulations. The recently reparameterized SCC-DFTB model by Cui and co-workers[39] gave a better overlap between the distributions of the three coordinating water O atoms, which removed the bimodal distribution. Our OM3*n* model also predicts a single peak near r = 2.5 Å, in good agreement with the experiment and ab initio MD simulations. The AM1*n* RDF displays a single peak at r = 2.5 Å, with a shoulder near r = 2.9 Å. This shoulder becomes another lower and wider peak at $r \approx$ 3.0 Å in the PM3*n* curve.

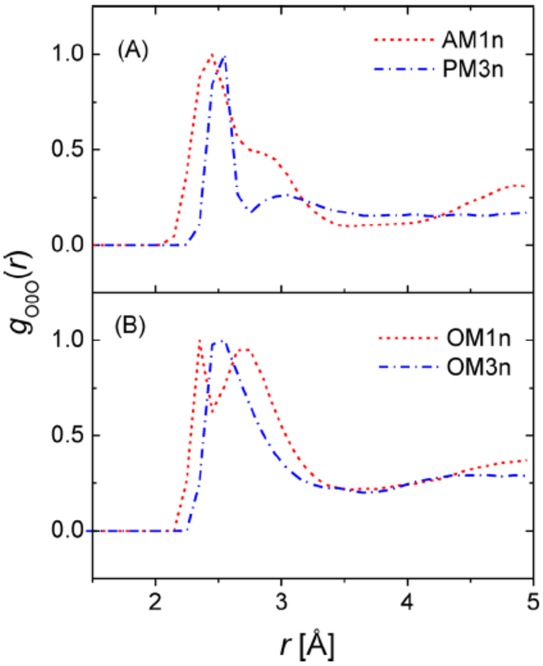


Figure 9. Radial distribution functions $g_{O0O}(r)$ for an excess proton in water computed with the reparameterized semiempirical methods; O0 represents the hydronium O atom. All curves have been scaled such that the height of the first peak at $r \approx$ 2.5 Å is 1.00.

The first-solvation-shell coordination numbers are given in the third column of Table 5. Compared with the coordination number of 3.0 from the ab initio MD simulations,[37] all coordination numbers from the semiempirical methods in Table 5 are much larger, including the OM3*n* result (6.5). We note that previous SCC-DFTB simulations[37,39] produced coordination numbers in the range of 4.5−5.2, which are closer to, but still larger than, the ab initio MD value. The larger coordination numbers in the semiempirical simulations indicate more fluidic structures around the hydronium ion, as also implied by the less-prominent second peaks and the minima at larger r values in the $g_{OO}(r)$ curves.

The RDF for hydrogen atoms around O0 is displayed in Figure 10. The first peak at ∼1 Å is due to the covalently bound H atoms in the hydronium ion. Near r = 1.8 Å, a small peak can be found in the PM3*n*, OM1*n*, and OM3*n* curves. This peak is caused by the H atom of the water molecule that is hydrogen-bonding to the hydronium ion oxygen. AM1*n* does not yield this small peak, because it does not properly describe the hydrogen-bonding network (see the earlier discussion on the hydrogen-bonding angles). All reparameterized models successfully predict the peak at 3.1 Å. On the basis of this evidence (Figures 9 and 10), we conclude that it is again OM3*n* which best reproduces the structural properties.

Traditionally, the structures of the solvated proton are classified as Eigen- or Zundel-like. However, the distinction between distorted Eigen and distorted Zundel structures is not clear-cut. There are various ways to define the Eigen-like and

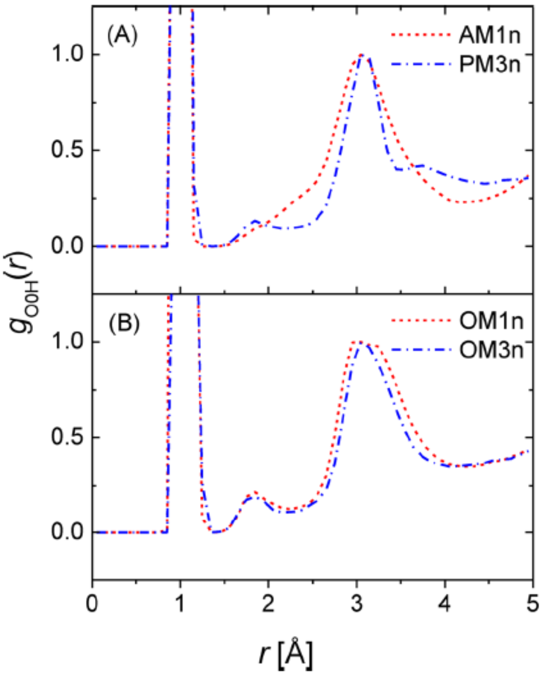


Figure 10. Radial distribution functions $g_{O0H}(r)$ for an excess proton in water computed with the reparameterized semiempirical methods; O0 represents the hydronium O atom. All curves have been scaled such that the height of the peak at $r \approx 3.0$ Å is 1.00.

Zundel-like geometries.[17,27,36,37] In this work, we tested three definitions based on different geometric parameters, the distributions of which are illustrated by histograms in Figure S13 in the Supporting Information. The first definition (Def-1) used by Choi and Jordan[36] employs the distance between the hydronium oxygen O0 and the oxygen of the nearest potential acceptor O1′ ($R_{O0O1'}$) as the criterion: Zundel for $R_{O0O1'} < 2.46$ Å, intermediate for 2.46 Å $\leq R_{O0O1'} \leq$ 2.49 Å, and Eigen for 2.49 Å $< R_{O0O1'}$. The second definition (Def-2) is similar to that used by Marx et al.[27] and relies on the position of H′, the hydrogen located between O0 and O1′; here, the criterion is the distance difference $dr = |R_{O0H'} - R_{O1'H'}|$: Zundel for $dr <$ 0.1 Å, intermediate for 0.1 Å $\leq dr \leq$ 0.2 Å, and Eigen for 0.2 Å $< dr$, where $R_{O0H'}$ and $R_{O1'H'}$ denote the distances between O0 and H′ and between O1′ and H′, respectively; the currently chosen Eigen dr value of 0.2 Å (instead of 0.3 Å in ref 33) has been suggested by Markovitch et al.[17] The third definition (Def-3) is based on the ratio ρ (see Figure 4): Zundel for $\rho >$ 0.46, intermediate for 0.45 $\leq \rho \leq$ 0.46 Å, and Eigen for $\rho <$ 0.45. We adopted ρ values of 0.45 and 0.46 Å for the following reason. The MP2-optimized Eigen structure has R_{O0O1} = 2.554 Å and R_{O0H} = 1.011 Å. To reduce the noise from the fast OH stretching vibrations of the hydronium ion, we (arbitrarily) took R_{O0H} = 1.05 Å as the starting point of proton transfer, which gave ρ_{start} = 0.41; we note that the value of 1.05 Å is close to the average O0H bond length of 1.057 Å from ab initio path integral calculations.[21,27] Because of the donor−acceptor swap at ρ_{end} = 0.5, we chose ρ_{mid} = 0.455 (with 0.005 at each side as the intermediate buffer) to be the boundary separating Eigen and Zundel. The average values of the relevant geometric parameters and the percentages of the Eigen, intermediate, and Zundel structures for the saved trajectories are listed in rows 6−17 of Table 6.

Table 6. Statistics of the Observed Donor−Acceptor Swaps, Grotthuss Shuttling Rates (Forward Hop), and Zundel-Eigen Structural Classifications for the Trajectories Obtained with the Reparameterized Semiempirical Methods[a]

	AM1*n*	PM3*n*	OM1*n*	OM3*n*
number of swaps	0	306	9364	4678
average time between swaps	n/a	0.65 ps	0.02 ps	0.04 ps
forward hops	0	11	91	105
average time between forward hops	n/a	18 ps	2.2 ps	1.9 ps
Def-1				
$\langle R_{O0O1'} \rangle$	2.31 Å	2.46 Å	2.35 Å	2.44 Å
Zundel	97.7%	51.2%	96.9%	64.0%
Eigen	0.8%	24.9%	1.3%	18.1%
Intermediate	1.5%	23.9%	1.8%	17.9%
Def-2				
$\langle dr = \vert R_{O0H'} - R_{O1'H'} \vert \rangle$	1.03 Å	0.45 Å	0.19 Å	0.28 Å
Zundel	0	1.4%	36.1%	19.6%
Eigen	100.0%	96.4%	35.6%	59.3%
Intermediate	0	2.2%	28.4%	21.1%
Def-3				
$\langle \rho \rangle$	0.309	0.416	0.464	0.453
Zundel	0	3.4%	61.9%	41.4%
Eigen	100.0%	94.6%	27.4%	46.5%
Intermediate	0	2.0%	10.7%	12.1%

[a] Based on 10 trajectories for each semiempirical model, each consisting of 2 ps equilibration and a 20 ps productive run. The criterion for donor−acceptor swaps is illustrated in Figure 4. The forward hop rate is defined in eqs 4 and 5 of the text. Three ways to define Zundel and Eigen structures have been used here. Def-1 is based on $R_{O0O1'}$: Zundel for $R_{O0O1'} < 2.46$ Å, intermediate for 2.46 Å $\leq R_{O0O1'} \leq 2.49$ Å, and Eigen for 2.49 Å $< R_{O0O1'}$. Def-2 is based on $dr = |R_{O0H'} - R_{O1'H'}|$: Zundel for $dr < 0.1$ Å, intermediate for 0.1 Å $\leq dr \leq$ 0.2 Å, and Eigen for 0.2 Å $< dr$. Def.-3 is based on ρ (see Figure 4): Zundel for $\rho > 0.46$, Intermediate for $0.45 \leq \rho \leq 0.46$ Å, and Eigen for $\rho < 0.45$. See Section 3.E for further discussion.

It is remarkable that these different definitions lead to rather different classifications of the geometries of the hydrated proton in the saved trajectories. While most of the geometries are characterized as Zundel-like by Def-1, the Eigen "flavor" is more prominent according to Def-2 and Def-3. Most noticeably, the AM1*n* geometries are considered almost exclusively Zundel-like by Def-1 but 100% Eigen-like by Def-2 and Def-3. The classifications by Def-2 and Def-3 are qualitatively similar, albeit with more geometries in the OM1*n* and OM3*n* trajectories being identified as distorted Eigen structures by Def-3. Such similarity is not surprising, because both Def-2 and Def-3 characterize the solvated proton structures by assessing how symmetrical the location of the migrating proton is. The AM1*n* and PM3*n* geometries appear to be very asymmetric on average, and, therefore, identifying them as being Eigen-dominated seems appropriate. Both OM1*n* and OM3*n* simulations yield mixed Zundel and Eigen structures in comparable amounts. However, the OM1*n* geometries are generally more symmetric than the OM3*n* geometries, implying that the Zundel population is higher in the OM1*n* simulations.

3.F. Dynamic Properties of Proton Transfer. Inspection of the trajectories revealed that no proton transfer ever occurred in the AM1*n* simulations. Proton transfer was observed occasionally in the PM3*n* simulations, and was found more frequently in the OM1n and OM3n simulations. The average time intervals between the proton shuttling were 0.65, 0.02, and 0.04 ps for PM3n, OM1n, and OM3n, respectively (row 3 in Table 6). These time intervals do not distinguish between the fast subpicosecond back-and-forth proton exchange within a donor–acceptor pair of a Zundel structure and the true proton transfer that results in a localized distorted Eigen structure, the latter of which occurring at a longer time scale. Similar to what has been done previously,[16] we computed the Grotthuss shuttling rate, or the forward hop rate, using an accumulation function:

$$h(t) = h(t - \Delta t) + \Delta h(\Delta t)$$
$$h(0) = 0 \tag{4}$$

where Δt is the time step, and $\Delta h(\Delta t)$ is given by

$$\Delta h(\Delta t) = \begin{cases} 0 & \text{if no proton hops} \\ 0 & \text{if a proton hops back to the last donor} \\ 1 & \text{if a proton hops to a new donor} \end{cases} \tag{5}$$

Note that $\Delta h(\Delta t)$ is defined here in a different way than previously defined.[16] The average time intervals between forward hops are found to be 18, 2.2, and 1.9 ps for the PM3n, OM1n, and OM3n simulations, respectively (row 5 in Table 6), indicating corresponding average rates of proton hopping of 0.06, 0.45, and 0.53 ps^{-1}, respectively. Interestingly, although proton transfer occurs more frequently in the OM1n simulations, there are more successful Grotthuss shuttling events for OM3n. The OM1n and OM3n hopping rates are close to the estimated experimental effective hopping rate (0.71 ps^{-1})[16] and to the rates obtained in the SCC-DFTB (~0.5 ps^{-1}) and CPMD (0.4 ps^{-1}) simulations by Maupin et al.[37] For comparison, we note that MS-EVB simulations[16] yielded a rate of 0.16 ps^{-1}.

We next look at the potentials of mean force (PMF) for proton transfer shown in Figure 11, where both $dr = |R_{O0H'} - R_{O1'H'}|$ and ρ are employed as distinguished reaction coordinates. Both yield a qualitatively similar picture of proton transfer for any given semiempirical method. The PMF from PM3n clearly indicates that the resting Eigen state ($dr \approx 0.5$ Å or $\rho \approx 0.41$) is more stable than the transition Zundel state ($dr \approx 0$ or $\rho \approx 0.5$) by ~2 kcal/mol. The barrier is quite high, compared with the thermal energy at temperatures near 300 K ($k_BT \approx 0.6$ kcal/mol), so that there are only occasional proton hops in the PM3n simulations. The OM3n model also predicts the Eigen–Zundel–Eigen proton transfer mechanism but with much lower barriers: 0.06 kcal/mol in the dr plot and 0.19 kcal/mol in the ρ plot. These OM3n effective barriers are lower than the classical CPMD barriers of 0.5 kcal/mol,[27,37] but, incidentally, are very close to the 0.15 kcal/mol barrier from ab initio path integral calculations,[27,33] which include the quantum fluctuations of all nuclei. A very shallow OM3n minimum corresponding to the Eigen structure is found near $dr = 0.25$ Å, again in excellent agreement with the ab initio path integral calculations.[21,27,33] By contrast, the PMFs from OM1n yield minima at the Zundel structure and thus indicate a Zundel–Zundel mechanism, which is consistent with the results of Maupin et al. from standard SCC-DFTB and from a specially parametrized SCC-DFTB version with a hydrogen-bonding damping function.[37] Cui and co-workers[39] also found the Zundel–Zundel mechanism in their standard and hydrogen-bonding-corrected SCC-DFTB simulations, while their third-order SCC-DFTB model with hydrogen-bonding corrections and a modified O–H repulsion potential gave the Eigen–Zundel–Eigen mechanism (albeit with a rather-high 0.9 kcal/mol free-energy barrier for proton transfer).

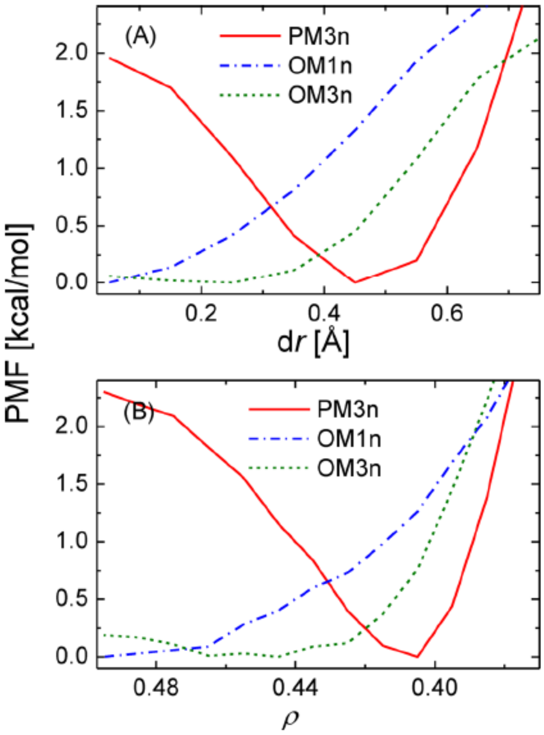


Figure 11. Potential of mean force (PMF) for the solvated proton: (A) The reaction coordinate is the distance difference $dr = |R_{O0H'} - R_{O1'H'}|$, where O0 is the hydronium oxygen atom, O1′ is the nearest potential acceptor, and H′ is the hydrogen between them ($R_{O0H'}$ ($R_{O1'H'}$) is the distance between O0 (O1′) and H′); (B) the reaction coordinate is the ratio ρ (see Figure 4 for definition).

Examples for the evolution of the hydronium O atom index in a representative sample trajectory are provided in Figure 12A for the PM3n, OM1n, and OM3n simulations. Also shown is the percentage of time when the given atom is identified as the hydronium-ion O atom. The sample trajectories were selected by requiring that the number of hydronium oxygen switches in the 20 ps trajectory should be approximately the same as the average number observed in all 10 trajectories for a given semiempirical model. In the PM3n sample trajectory, only two O atoms were ever identified as the donor, with O220 being the dominant one (>99% of the time). By contrast, five and four O atoms served as donor in the OM1n and OM3n sample trajectories, respectively, and in either case, three of these atoms were associated with the hydronium ion for a long time (>9%), which is consistent with the picture of frequent proton transfer previously discussed. For instance, in the OM3n sample trajectory, the donor switched from O529 to O109 shortly after the productive run began, remained at O109 for ~10 ps with many short visits to O55 and O394 of up to ~50 fs, and then migrated to O106 after a period of ~3 ps that featured many rapid rattles between O109 and O106. Such dynamics was found to be typical of the OM3n simulations. The forward hop rates $h(t)$ are plotted in Figure 12B for the sample trajectories. The flat plateaus represent the resting Eigen state of no proton transfer with oscillatory shuttling in the Zundel structure, while the often-clustered steps indicate Grotthuss

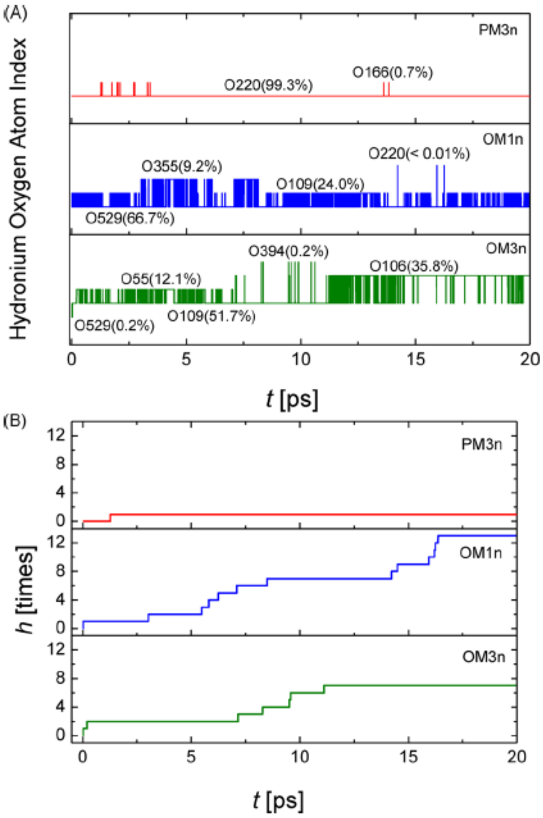


Figure 12. Evolution of (A) the hydronium O atom index and (B) the forward hop rate in sample trajectories from the PM3*n*, OM1*n*, and OM3*n* simulations. The percentage of time when the given atom was identified as the hydronium ion O atom is shown in panel (A) in parentheses. In the sample trajectories, the hydronium O atom index changed 30 times for PM3*n*, 1207 times for OM1*n*, and 576 times for OM3*n*. The average numbers of such changes in all trajectories are 31 for PM3n, 936 for OM1*n*, and 468 for OM3*n*. In panel (B), the forward hop rate $h(t)$ is defined in eqs 4 and 5 of the text. No donor–acceptor swapping occurred in the AM1 (AM1*n*) simulations.

proton transfer events. The length of the plateaus varies, but is roughly in the range of 5−10 ps for the OM1*n* and OM3*n* simulations (generally somewhat longer for OM3*n*). The length of the plateaus is determined by the fluctuation-induced breakage and reorganization of hydrogen bonds in the solvation shells, which are the rate-limiting steps of proton diffusion.[1,17]

Figure 13 displays the geometries of the sample trajectories projected onto the $R_{OO1'}$−d*r* plane, as well as the corresponding gas-phase two-dimensional potential surfaces V^{PES} for proton transfer. While V^{PES} differs from the free-energy surface for proton transfer in bulk water, the plots are instructive in connecting the simulated trajectories with the potential surfaces used in the parametrization (see Section 3.B). As discussed previously, the gas-phase PM3*n* surface yields a narrow valley, which differs noticeably in shape from the reference MP2 surface. Here, we find that the PM3*n* sample trajectory spends the majority of time wandering in the area centered at $R_{OO1'}$ = 2.45 Å and d*r* = 0.45 Å, roughly corresponding to one end of the narrow valley. Compared with the MP2 surface, the well in the OM1*n* surface is wider in the R_{OO} direction but narrower in the d*r* direction; this is consistent with the concentration of the OM1*n* geometries at small values of d*r*. The well in the OM3*n* surface resembles the well in the MP2 surface more closely; the OM3*n* distribution is more diffusive in the d*r* direction and more condensed in the R_{OO} direction. As previously discussed, OM3*n* is the most accurate semiempirical model developed in this work, and, hence, the OM3*n* distributions should be the most reliable. The results shown in Figure 13 suggest that the quality of reproducing V^{PES} will play a critical role for the successful semiempirical modeling of proton transfer in bulk water.

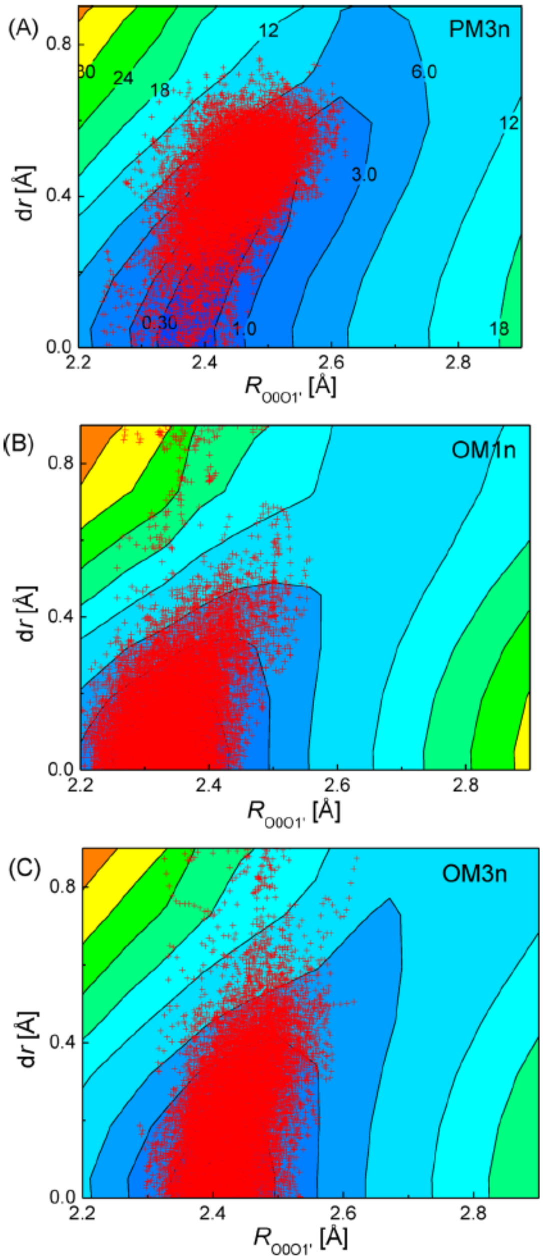


Figure 13. Projections of the geometries in the sample trajectories in Figure 11 for (A) PM3*n*, (B) OM1*n*, and (C) OM3*n*. The *x*-axis represents the distance between the hydronium oxygen O0 and the oxygen of the nearest potential acceptor O1′; the *y* axis represents the distance difference d*r* = $|R_{O0H'} - R_{O1'H'}|$, where H′ is the hydrogen between O0 and O1′. For comparison, the gas-phase potential for proton transfer between two water molecules is also shown for each reparameterized semiempirical method.

Finally, we comment on the algorithm used here to update the hydronium oxygen index (Figure 4). The acceptor O1 was the oxygen atom O1′ of the closest water molecule most of the time, but not always. Figure S14 in the Supporting Information shows plots of the distances $R_{\mathrm{OO1'}}$ vs R_{OO1} at the donor−acceptor swap moments. We find $R_{\mathrm{OO1'}} \neq R_{\mathrm{OO}}$ only for a small fraction of the swaps (up to 2.7%). As such, the "special pair dancing" dynamics discussed in ref 17 can be applied here, e.g., to interpret the bimodal distribution in the solvated proton RDF $g_{\mathrm{OO}}(r)$ obtained from OM1*n* (see Section 3.E). The donor−acceptor distances at the swapping event display normal distributions centered at 2.37, 2.33, and 2.41 Å for the PM3*n*, OM1*n*, and OM3*n* simulations, respectively (see Figure S15 in the Supporting Information); these values match the respective locations of the well centers in V^{PES} for the corresponding semiempirical models.

4. CONCLUSION

In this paper, we have reparameterized several well-established semiempirical methods, with the objective to improve the semiempirical modeling of proton transfer in bulk water. Overall, the orthogonalization-corrected OM1*n* and OM3*n* models outperform the MNDO-based AM1*n* and PM3*n* methods in reproducing the ab initio reference geometries and energies, as well as in describing proton transfer. The OM3*n* model seems to be the most successful overall, providing radial distribution functions and potentials of mean force that are the most accurate among all methods considered. The small but finite (0.1−0.2 kcal/mol) OM3*n* free-energy barrier for proton transfer agrees with the barrier from ab initio path-integral studies.[21,27,33] The OM3*n* simulations support the Eigen−Zundel−Eigen mechanism, where a proton is transferred between resting Eigen-like structures via transitional Zundel-like structures. The success of the OM3*n* model is likely due to the resemblance between the OM3*n* potential energy surface and the effective ab initio path-integral energy surface for the Zundel ion.[21] This good match is, to some extent, fortuitous, because our reparameterization was done with regard to MP2/aug-cc-pVTZ reference data (and not with respect to the path-integral surface[21]). Future refinements of the current semiempirical surfaces are conceivable, for example, by including explicit empirical corrections for dispersion interactions.

The present work demonstrates that the semiempirical framework is robust and flexible enough to allow for realistic reparameterizations, which provide a much improved description of water and proton transfer in water. Not surprisingly, it has been easier to obtain more accurate results by starting from the improved OM*x* models with orthogonalization corrections rather than from the MNDO-based AM1 or PM3 methods. Given our choice of MP2/aug-cc-pVTZ reference data, our favored OM3*n* model is expected to provide results of MP2-like quality for water and proton transfer in water, and it should thus be useful in practice. We anticipate that it should also be possible, in principle, to reparameterize the OM*x* methods with regard to even more-accurate ab initio data (e.g., computed at the coupled-cluster level with larger basis sets).

Finally, we note that the new semiempirical models are specifically designed for water and proton transfer in water. They may or may not be more accurate than the original methods in describing other molecules or reactions. The specific reaction parametrization[45] offers a convenient way to develop computationally efficient tools with only a handful of adjustable parameters for simulations of specific systems that are too large and too expensive for density functional theory and ab initio methods to handle. Another recent example[87] is the reparameterization of AM1 for modeling the reduction of 7,8-dihydrofolate by nicotinamide adenine dinucleotide phosphate hydride in dihydrofolate reductase (DHFR). The corresponding SRP AM1 Hamiltonian was used in QM/MM simulations of the DHFR-catalyzed reaction to compute kinetic isotope effects (using a mass-perturbation-based path-integral approach), in excellent agreement with the experiment.

■ ASSOCIATED CONTENT

Supporting Information

Table S1 describes the optimized parameters for the currently investigated semiempirical methods. Table S2 summarizes the maxima and minima in the RDF for bulk water. Figures S1 and S2 compare the RDF computed at the MM level using the cluster model and periodic boundary conditions. The optimized geometries of the Eigen cation, the water tetramer, the Zundel ion in complex with two water molecules, and the water pentamer are shown in Figures S3−S6. Cross sections of two-dimensional potential surfaces for gas-phase proton transfer between two water molecules within a Zundel ion are illustrated in Figures S7−S11. Water dimer geometries obtained in relaxed surface scans are exemplified in Figure S12. Analyses of sample trajectories are presented in Figure S13. The distribution of R_{O1O2} vs $R_{\mathrm{O1O2'}}$ at donor−acceptor swap events and the corresponding histogram of R_{O1O2} are displayed in Figures S14 and S15, respectively. Integrated coordination numbers are plotted in Figures S16 and S17 for water and in Figure S18 for the hydronium ion. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*E-mails: thiel@mpi-muelheim.mpg.de (W.T.), hai.lin@ucdenver.edu (H.L.).

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by grants of the National Science Foundation (No. CHE0952337) and the Alexander von Humboldt Foundation to H.L.

■ REFERENCES

(1) Marx, D. *ChemPhysChem* **2006**, *7*, 1848−1870.

(2) Agmon, N. *Chem. Phys. Lett.* **1995**, *244*, 456−462.

(3) Stillinger, F. H.; David, C. W. *J. Chem. Phys.* **1978**, *69*, 1473−1484.

(4) David, C. W. *J. Chem. Phys.* **1996**, *104*, 7255−7260.

(5) Halley, J. W.; Rustad, J. R.; Rahman, A. *J. Chem. Phys.* **1993**, *98*, 4110−4119.

(6) Billeter, S. R.; van Gunsteren, W. F. *J. Phys. Chem. A* **1998**, *102*, 4669−4678.

(7) Lussetti, E.; Pastore, G.; Smargiassi, E. *Chem. Phys. Lett.* **2003**, *381*, 287−291.

(8) Mahadevan, T. S.; Garofalini, S. H. *J. Phys. Chem. B* **2007**, *111*, 8919−8927.

(9) Fogarty, J. C.; Aktulga, H. M.; Grama, A. Y.; van Duin, A. C. T.; Pandit, S. A. *J. Chem. Phys.* **2010**, *132*, 174704/1−14.

(10) Selvan, M. E.; Keffer, D. J.; Cui, S.; Paddison, S. J. *J. Phys. Chem. C* **2010**, *114*, 11965−11976.

(11) Kale, S.; Herzfeld, J.; Dai, S.; Blank, M. *J. Biol. Phys.* **2012**, *38*, 49−59.

(12) Schmitt, U. W.; Voth, G. A. *J. Phys. Chem. B* **1998**, *102*, 5547−5551.

(13) Warshel, A.; Weiss, R. M. *J. Am. Chem. Soc.* **1980**, *102*, 6218−6226.

(14) Lefohn, A. E.; Ovchinnikov, M.; Voth, G. A. *J. Phys. Chem. B* **2001**, *105*, 6628−6637.

(15) Swanson, J. M. J.; Maupin, C. M.; Chen, H.; Petersen, M. K.; Xu, J.; Wu, Y.; Voth, G. A. *J. Phys. Chem. B* **2007**, *111*, 4300−4314.

(16) Wu, Y. J.; Chen, H. N.; Wang, F.; Paesani, F.; Voth, G. A. *J. Phys. Chem. B* **2008**, *112*, 467−482.

(17) Markovitch, O.; Chen, H.; Izvekov, S.; Paesani, F.; Voth, G. A.; Agmon, N. *J. Phys. Chem. B* **2008**, *112*, 9456−9466.

(18) Vuilleumier, R.; Borgis, D. *Chem. Phys. Lett.* **1998**, *284*, 71−77.

(19) Vuilleumier, R.; Borgis, D. *J. Chem. Phys.* **1999**, *111*, 4251−4266.

(20) Sagnella, D. E.; Tuckerman, M. E. *J. Chem. Phys.* **1998**, *108*, 2073/1−2073/11.

(21) Brancato, G.; Tuckerman, M. E. *J. Chem. Phys.* **2005**, *122*, 224507/1−224507/11.

(22) Kornyshev, A. A.; Kuznetsov, A. M.; Spohr, E.; Ulstrup, J. *J. Phys. Chem. B* **2003**, *107*, 3351−3366.

(23) Park, K.; Lin, W.; Paesani, F. *J. Phys. Chem. B* **2012**, *116*, 343−352.

(24) Car, R.; Parrinello, M. *Phys. Rev. Lett.* **1985**, *55*, 2471−2474.

(25) Tuckerman, M. E.; Laasonen, K.; Sprik, M.; Parrinello, M. *J. Phys.: Condens. Matter* **1994**, *6*, A93−A100.

(26) Tuckerman, M.; Laasonen, K.; Sprik, M.; Parrinello, M. *J. Phys. Chem.* **1995**, *99*, 5749−5752.

(27) Marx, D.; Tuckerman, M. E.; Hutter, J.; Parrinello, M. *Nature* **1999**, *397*, 601−604.

(28) Asthagiri, D.; Pratt, L. R.; Kress, J. D. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 6704−6708.

(29) Chandra, A.; Tuckerman, M. E.; Marx, D. *Phys. Rev. Lett.* **2007**, *99*, 145901.

(30) Swanson, J. M. J.; Simons, J. *J. Phys. Chem. B* **2009**, *113*, 5149−5161.

(31) Tuckerman, M. E.; Chandra, A.; Marx, D. *J. Chem. Phys.* **2010**, *133*, 124108/1−22.

(32) Marsalek, O.; Elles, C. G.; Pieniazek, P. A.; Pluharova, E.; VandeVondele, J.; Bradforth, S. E.; Jungwirth, P. *J. Chem. Phys.* **2011**, *135*, 224510/1−224510/14.

(33) Marx, D.; Tuckerman, M. E.; Parrinello, M. *J. Phys.: Condens. Matter* **2000**, *12*, A153−A159.

(34) Berkelbach, T. C.; Lee, H.-S.; Tuckerman, M. E. *Phys. Rev. Lett.* **2009**, *103*, 238302/1−238302/4.

(35) Elstner, M.; Hobza, P.; Frauenheim, T.; Suhai, S.; Kaxiras, E. *J. Chem. Phys.* **2001**, *114*, 5149−5155.

(36) Choi, T. H.; Jordan, K. D. *J. Phys. Chem. B* **2010**, *114*, 6932−6936.

(37) Maupin, C. M.; Aradi, B. l.; Voth, G. A. *J. Phys. Chem. B* **2010**, *114*, 6922−6931.

(38) Riccardi, D.; Konig, P.; Prat-Resina, X.; Yu, H. B.; Elstner, M.; Frauenheim, T.; Cui, Q. *J. Am. Chem. Soc.* **2006**, *128*, 16302−16311.

(39) Goyal, P.; Elstner, M.; Cui, Q. *J. Phys. Chem. B* **2011**, *115*, 6790−6805.

(40) McNamara, J. P.; Hillier, I. H. *Phys. Chem. Chem. Phys.* **2007**, *9*, 2362−2370.

(41) Řezáč, J.; Fanfrlík, J. i.; Salahub, D.; Hobza, P. *J. Chem. Theory Comput.* **2009**, *5*, 1749−1760.

(42) Korth, M. *J. Chem. Theory Comput.* **2010**, *6*, 3808−3816.

(43) Korth, M. *ChemPhysChem* **2011**, *12*, 3131−3142.

(44) Gaus, M.; Cui, Q.; Elstner, M. *J. Chem. Theory Comput.* **2011**, *7*, 931−948.

(45) Gonzalez-Lafont, A.; Truong, T. N.; Truhlar, D. G. *J. Phys. Chem.* **1991**, *95*, 4618−4627.

(46) Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. *J. Am. Chem. Soc.* **1985**, *107*, 3902−3909.

(47) Stewart, J. J. P. *J. Comput. Chem.* **1989**, *10*, 209−220.

(48) Stewart, J. J. P. *J. Comput. Chem.* **1989**, *10*, 221−264.

(49) Dewar, M. J. S.; Thiel, W. *J. Am. Chem. Soc.* **1977**, *99*, 4899−4907.

(50) Kolb, M.; Thiel, W. *J. Comput. Chem.* **1993**, *14*, 775−789.

(51) Weber, W.; Thiel, W. *Theor. Chem. Acc.* **2000**, *103*, 495−506.

(52) Scholten, M. Doctoral Thesis, University of Düsseldorf, Düsseldorf, Germany, 2003.

(53) Otte, N.; Scholten, M.; Thiel, W. *J. Phys. Chem. A* **2007**, *111*, 5751−5755.

(54) Korth, M.; Thiel, W. *J. Chem. Theory Comput.* **2011**, *7*, 2929−2936.

(55) Møller, C. M. S.; Plesset, M. S. *Phys. Rev.* **1934**, *46*, 618−622.

(56) Dunning, J. T. H. *J. Chem. Phys.* **1989**, *90*, 1007−1023.

(57) Woon, D. E.; Thom H. Dunning, J. *J. Chem. Phys.* **1994**, *100*, 2975−2988.

(58) Ramírez, F.; Hadad, C. Z.; Guerra, D.; David, J.; Restrepo, A. *Chem. Phys. Lett.* **2011**, *507*, 229−233.

(59) Pérez, J. F.; Hadad, C. Z.; Restrepo, A. *Int. J. Quantum Chem.* **2008**, *108*, 1653−1659.

(60) Stull, D. R.; Prophet, H. In *National Standard Reference Data Series (United States, National Bureau of Standards)*; National Bureau of Standards (NBS): Washington, DC, 1971; Vol. 37.

(61) Lias, S. G.; Liebman, J. F.; Levin, R. D. *J. Phys. Chem. Ref. Data* **1984**, *13*, 695−808.

(62) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Keith, T.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, J. M.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, O.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J.; Version B.01. Gaussian, Inc.: Wallingford, CT, 2010.

(63) Thiel, W.; Version 7.0. Max-Planck-Institut für Kohlenforschung: Mülheim an der Ruhr, Germany, 2005.

(64) Stewart, J. P. *J. Mol. Model.* **2007**, *13*, 1173−1213.

(65) Hofer, T. S.; Hitzenberger, M.; Randolf, B. R. *J. Chem. Theory Comput.* **2012**, *8*, 3586−3595.

(66) Warshel, A.; Levitt, M. *J. Mol. Biol.* **1976**, *103*, 227−249.

(67) Field, M. J.; Bash, P. A.; Karplus, M. *J. Comput. Chem.* **1990**, *11*, 700−733.

(68) Gao, J., Thompson, M. A., Eds. *Combined Quantum Mechanical and Molecular Mechanical Methods*; ACS Symposium Series 712; American Chemical Society: Washington, DC, 1998.

(69) Zhang, Y.; Lee, T.-S.; Yang, W. *J. Chem. Phys.* **1999**, *110*, 46−54.

(70) Gogonea, V.; Westerhoff, L. M.; Merz, K. M., Jr. *J. Chem. Phys.* **2000**, *113*, 5604−5613.

(71) Sherwood, P. In *Modern Methods and Algorithms of Quantum Chemistry*; Grotendorst, J., Ed.; John von Neumann-Instituts: Jülich, Germany, 2000; Vol. 3, pp 285−305.

(72) Lin, H.; Truhlar, D. G. *Theor. Chem. Acc.* **2007**, *117*, 185−199.

(73) Senn, H. M.; Thiel, W. *Top. Curr. Chem.* **2007**, *268*, 173−290.

(74) Berendsen, H. J. C.; Postma, J. P. M.; von Gunsteren, W. F.; Hermans, J. In *Intermolecular Forces*; Pullman, B., Ed.; D. Reidel Publishing Company: Dordrecht, The Netherlands, 1981; pp 331−342.

(75) Bakowies, D.; Thiel, W. *J. Phys. Chem.* **1996**, *100*, 10580−10594.

(76) Lin, H.; Zhang, Y.; Pezeshki, S.; Truhlar, D. G. QMMM, Version 1.4.0.CO; University of Minnesota: Minneapolis, MN, 2012.

(77) Ponder, J. W. TINKER, Version 5.1; Washington University: St. Louis, MO, 2010.

(78) Nosé, S. *Mol. Phys.* **1982**, *52*, 255−268.
(79) Hoover, W. G. *Phys. Rev. A* **1985**, *31*, 1695−1697.
(80) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926−935.
(81) Thiel, W. In *Theory and Applications of Computational Chemistry*; Dykstra, C. E., Frenking, G., Kim, K. S., Scuseria, G. E., Eds.; Elsevier: Amsterdam, 2005; pp 559−580.
(82) Tuttle, T.; Thiel, W. *Phys. Chem. Chem. Phys.* **2008**, *10*, 2159−2166.
(83) Grimme, S. *J. Comput. Chem.* **2006**, *27*, 1787−1799.
(84) Soper, A. K. *Chem. Phys.* **2000**, *258*, 121−137.
(85) Soper, A. K.; Benmore, C. J. *Phys. Rev. Lett.* **2008**, *101*, 065502/1−065502/4.
(86) Head-Gordon, T.; Johnson, M. E. *Proc. Natl. Acad. Sci. U.S.A.* **2006**, *103*, 7973−7977.
(87) Doron, D.; Major, D. T.; Kohen, A.; Thiel, W.; Wu, X. *J. Chem. Theory Comput.* **2011**, *7*, 3420−3437.

May 2, 2013

Revised for *JCTC*

Supporting Information for

Specific Reaction Path Hamiltonian for Proton Transfer in Water: Re-parameterized Semi-Empirical Models

Xin Wu[1], Walter Thiel[1]*, Soroosh Pezeshki[2], and Hai Lin[2]*

[1] *Max-Planck-Institut für Kohlenforschung, 45470 Mülheim an der Ruhr, Germany*

[2] *Chemistry Department, University of Colorado Denver, Denver, Colorado 80217 USA*

Received Date: 2013

* Email: thiel@mpi-muelheim.mpg.de, hai.lin@ucdenver.edu

List of Contents

.

Table S1. Description of the optimized parameters for the semi-empirical methods.

Parameters [a]	MNDO-based (AM1 and PM3)	OMx (x = 1, 2 and 3)
Slate-type orbital exponents ζ_s, ζ_p	Y	N
Scale factor for the Gaussian-type orbitals ζ	N	Y
One-electron one-center integrals U_{ss}, U_{pp}	Y	Y
One-electron two-center integral parameters β_s, β_p	Y	N
One-electron two-center integral parameters β_s, β_p, β_π, α_s, α_p, α_π	N	Y
Two-electron one-center integrals G_{ss}, G_{pp}, G_{sp}, $G_{p}2$, H_{sp}	Y	Y
Atomic core-core repulsion term parameters α	Y	N
Gaussian core repulsion function parameters K, L, M	Y	N
Pre-factors for orthogonalization corrections F, G	N	Y

[a] Parameters are optimized for oxygen and hydrogen, for which the total number of parameters is 32 for AM1, 29 for PM3, 27 for OM1, 31 for OM2, and 27 for OM3, respectively. Note that the parameters β_s and β_p are defined differently in the MNDO-based and orthogonalization-corrected methods.

Table S2. Location of minima and maxima in the radial distribution functions for bulk water.

	$g_{OO}(r)$ [a]			$g_{OH}(r)$ [a]		
	1st Max.	1st Min.	2nd Max.	1st Max.	1st Min.	2nd Max.
Exp.[b]	2.76	3.42	4.47	1.71	2.43	3.33
AM1	2.75	3.85	n/a	2.25	2.85	3.45
AM1n	2.75	3.90	n/a	2.25	2.75	3.45
PM3	2.75	4.05	n/a	1.85	2.25	3.25
PM3n	2.75	4.15	n/a	1.75	2.25	3.25
OM1	2.85	3.85	n/a	2.25	2.60	3.45
OM1n	2.85	3.85	n/a	2.25	2.55	3.35
OM3	2.35	3.15	3.95	n/a	2.30	2.95
OM3n	2.75	3.45	4.75	1.85	2.45	3.35

[a] In Å. The maximum corresponding to the covalently bound hydrogen is excluded from $g_{OH}(r)$. See Fig. 7 and Fig. 8 in the text for the RDF curves.

[b] Soper, A. K. *Chem. Phys.* **2000**, *258*, 121-137; Soper, A. K.; Benmore, C. J. *Phys. Rev. Lett.* **2008**, *101*, 065502/1-4.

Figure S1: Radial distribution functions $g_{OO}(r)$ for water at the MM level

Radial distribution functions $g_{OO}(r)$ for water computed for the (A) SPC and (B) TIP3P water models. PBC indicates simulations employing periodic boundary conditions. The experimental data (Exp) is plotted in both panels for comparison. Note that all curves have been scaled such that the height of the first peak is 1.00.

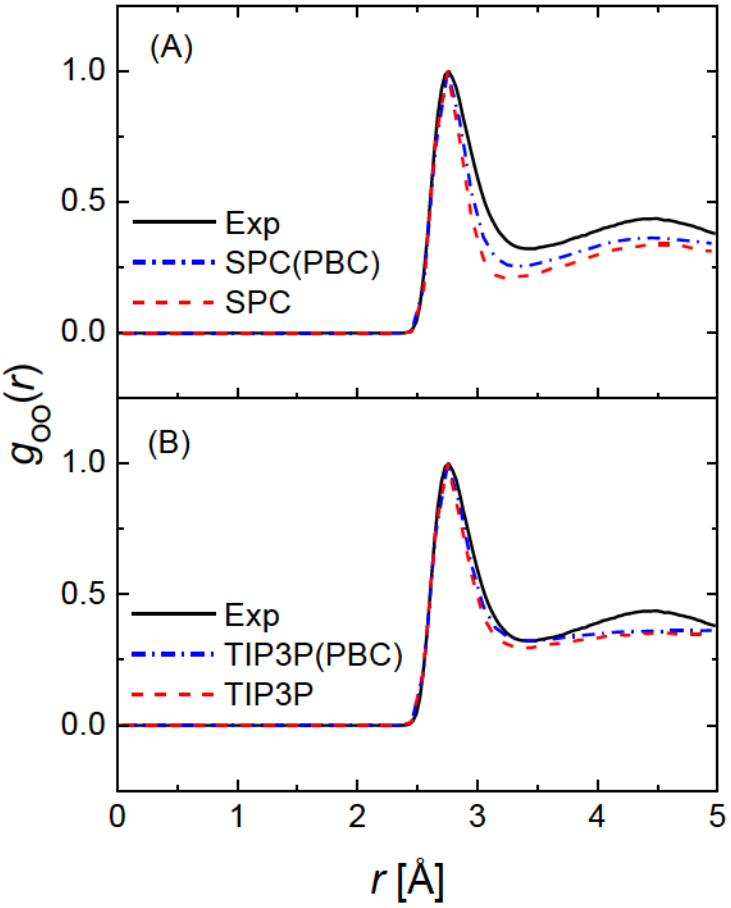


Figure S2: Radial distribution functions $g_{OH}(r)$ for water at the MM level

Radial distribution functions $g_{OH}(r)$ for water computed for the (A) SPC and (B) TIP3P water models. PBC indicates simulations employing periodic boundary conditions. The experimental data (Exp) is plotted in both panels for comparisons. Note that all curves have been scaled such that the height of the second peak in the graph (near 3.3 Å) is 1.00.

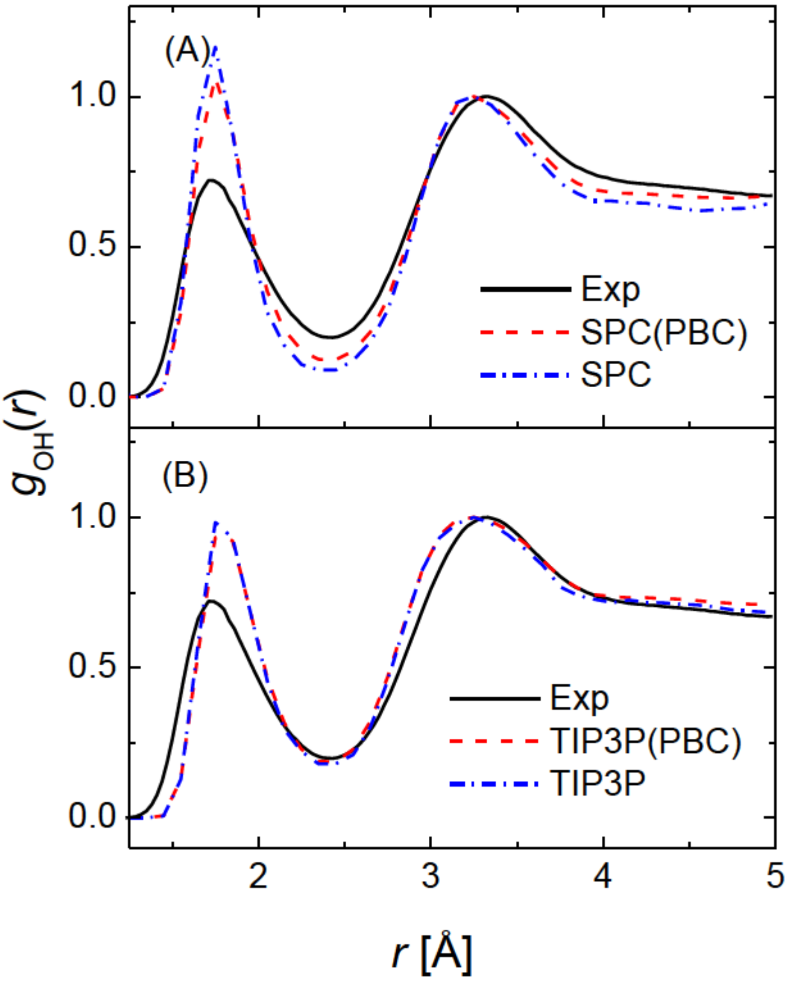
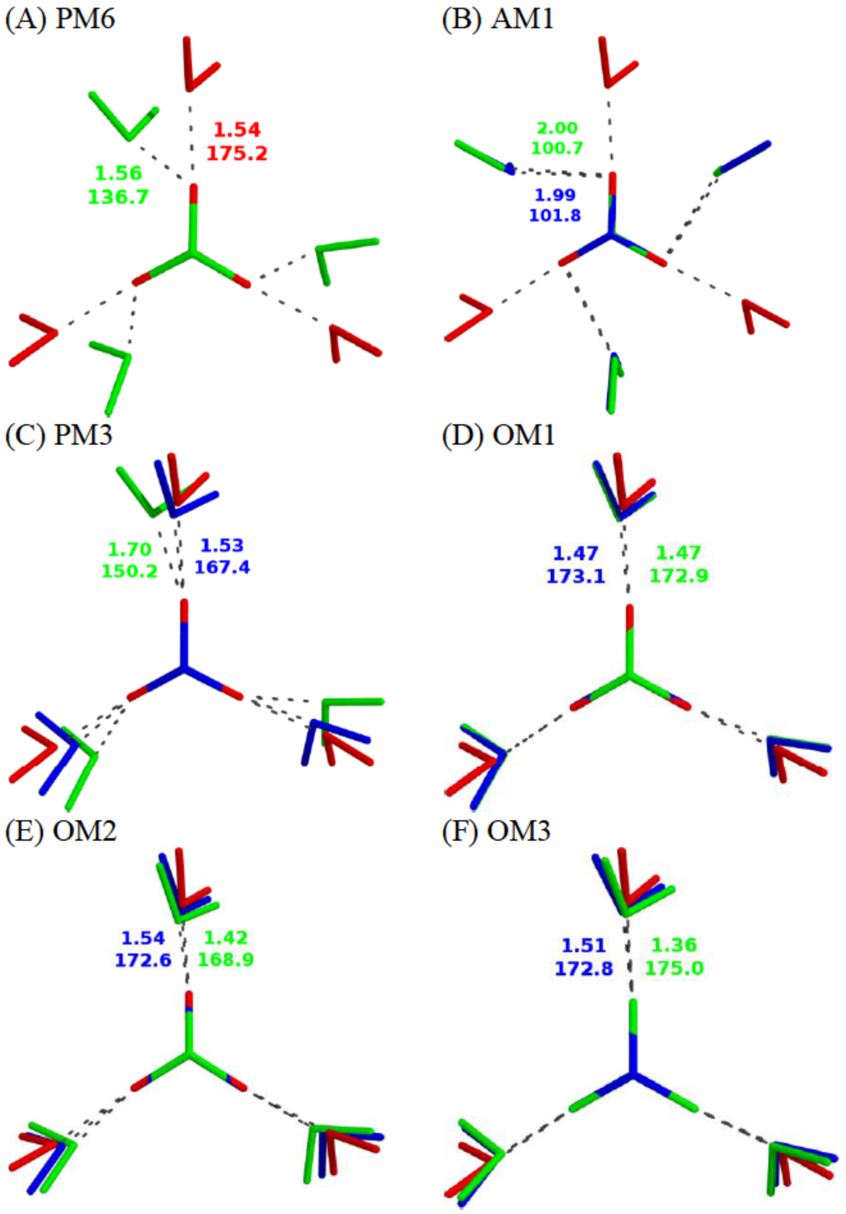


Figure S3: Optimized Eigen cation

Superimposition of the optimized geometries of the Eigen cation (Entry (g) in Fig. 1). The alignment is based on the hydronium ion coordinates. Color code for the geometries: MP2 in red, authentic semi-empirical methods in green, and re-parameterized semi-empirical models in blue. Distances in Å and angles in degree.



(A) PM6

(B) AM1

(C) PM3

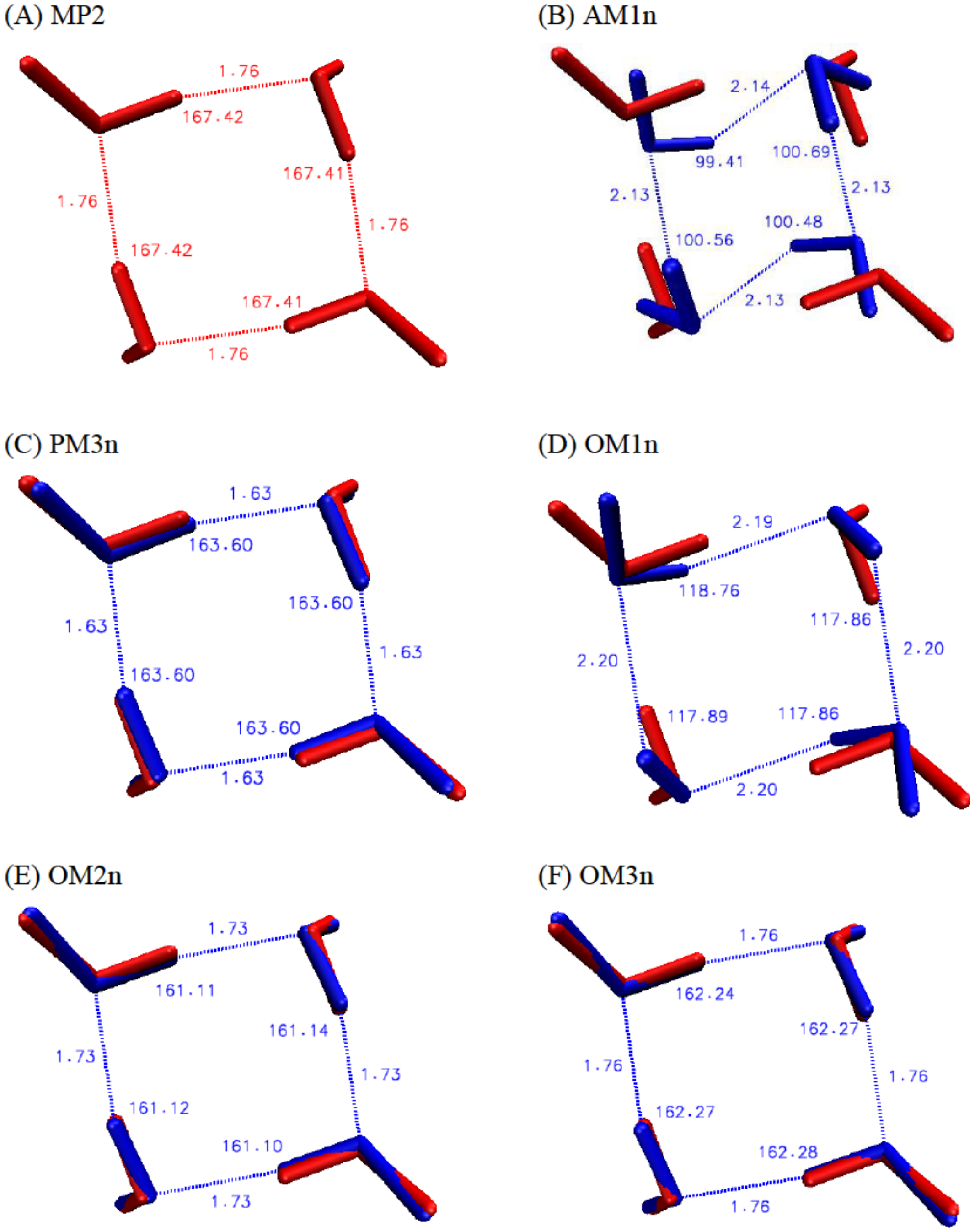
(D) OM1

(E) OM2

(F) OM3

Figure S4: Optimized water tetramer

Superimposition of the optimized geometries of the water tetramer (Entry (h) in Fig. 1). The alignment is based on all atomic coordinates. See caption of Fig. S3 for conventions.



(A) MP2

(B) AM1n

(C) PM3n

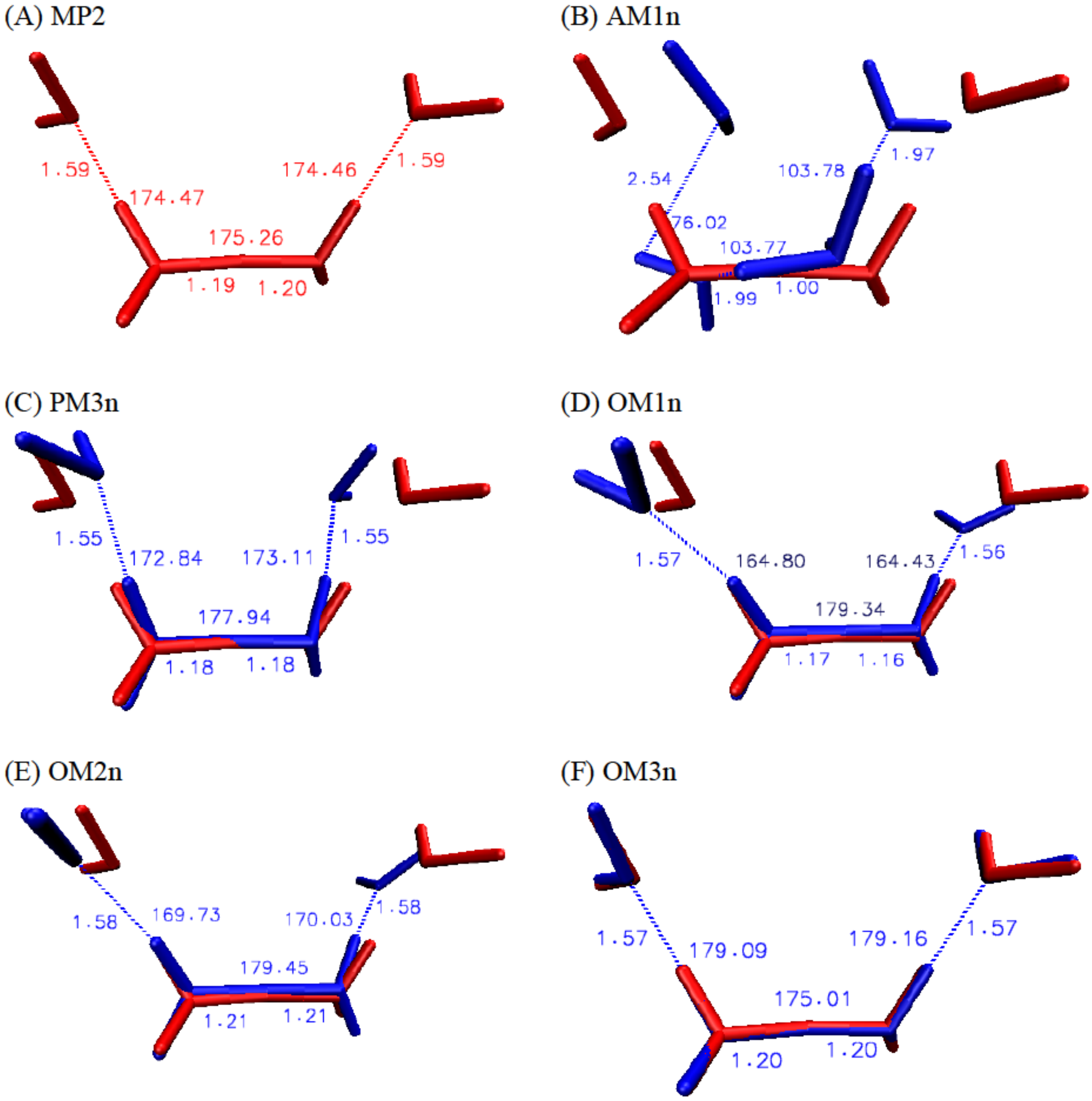
(D) OM1n

(E) OM2n

(F) OM3n

Figure S5: Optimized Zundel ion in complex with two water molecules

Superimposition of the optimized Zundel ion in complex with two water molecules (Entry (i) in Fig. 1). The alignment is based on the Zundel ion coordinates except in AM1n for which the alignment is based on all atomic coordinates. See caption of Fig. S3 for conventions.


(A) MP2

(B) AM1n

(C) PM3n

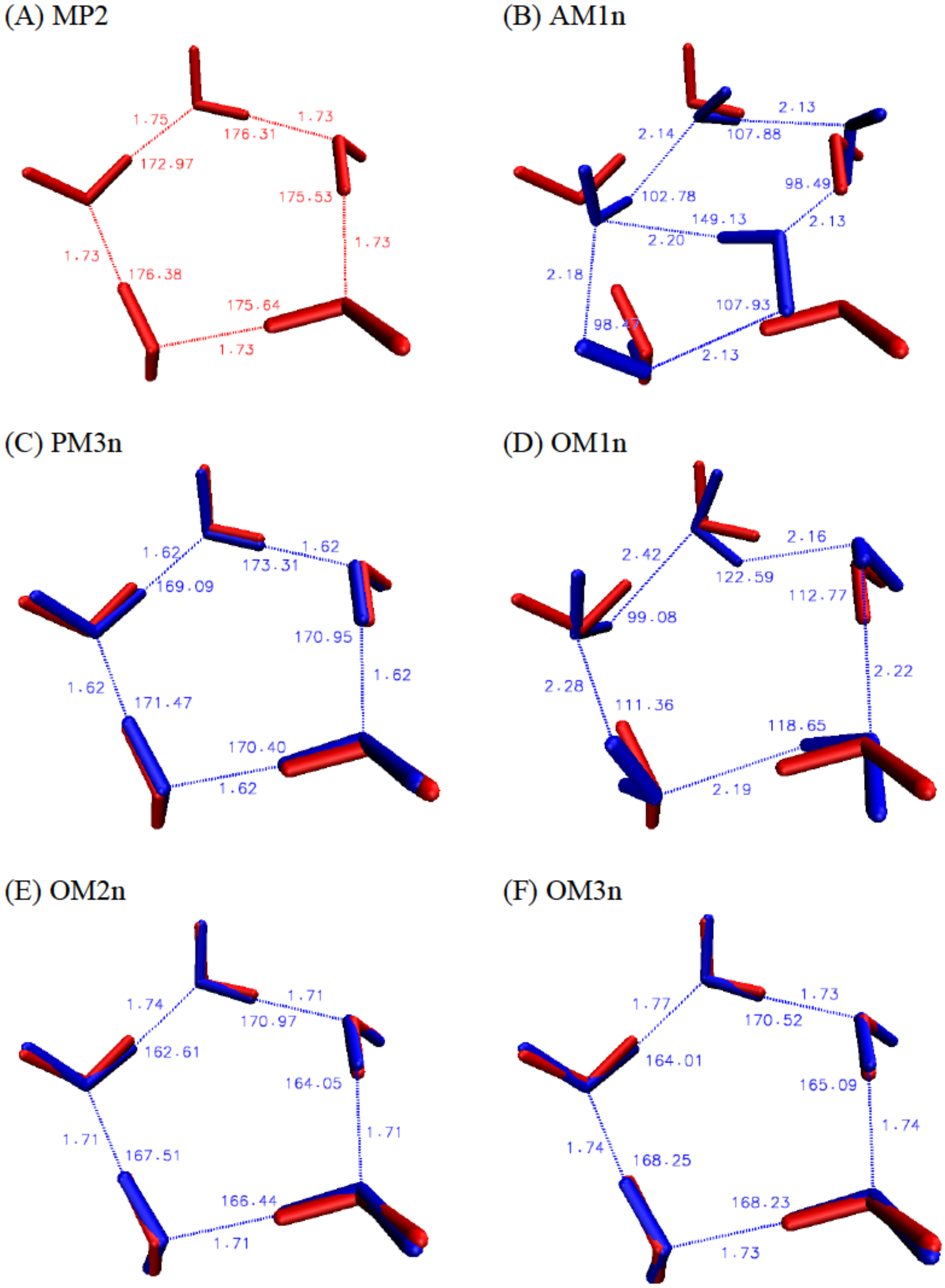
(D) OM1n

(E) OM2n

(F) OM3n

Figure S6: Optimized water pentamer

Superimposition of the optimized geometries of the water pentamer (Entry (j) in Fig. 1). The alignment is based on all atomic coordinates. See caption of Fig. S3 for conventions.



(A) MP2

(B) AM1n

(C) PM3n

(D) OM1n

(E) OM2n

(F) OM3n

Figure S7: Gas-phase energy profiles for proton transfer: AM1 and AM1n

Comparison of MP2 (solid line), AM1 (dashed lines), and AM1n (dashed-dotted lines) gas-phase energy profiles for proton transfer between two water molecules in the Zundel ion for given oxygen-oxygen distances (R_{OO} in Å). Here, dr is the difference in the distances between the migrating proton and the two oxygen atoms. The MP2/aug-cc-pVTZ data obtained from relaxed surface scans in panel (A) were included in the training set, and those in panel (B) in the validation set. The AM1 curve is right beneath the AM1n curve.

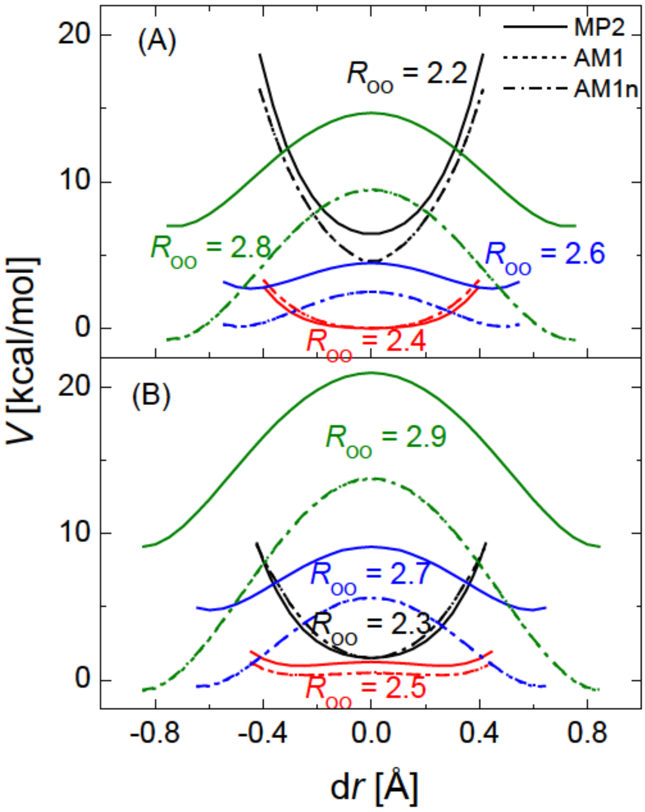


Figure S8: Gas-phase energy profiles for proton transfer: PM3 and PM3n

Comparison of MP2 (solid line), PM3 (dashed lines), and PM3n (dashed-dotted lines) gas-phase energy profiles for proton transfer between two water molecules in the Zundel ion for given oxygen-oxygen distances (R_{OO} in Å). See caption of Fig. S7 for conventions.

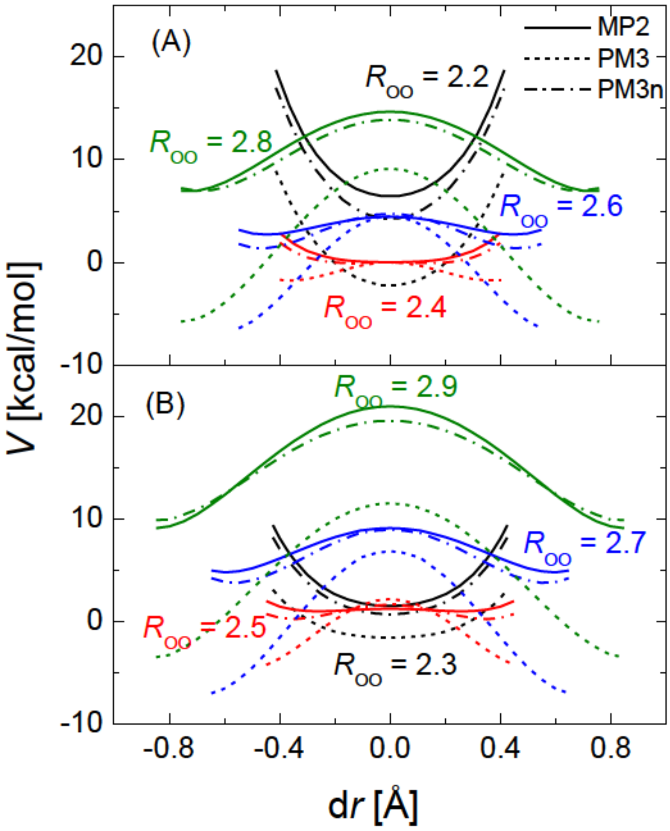


Figure S9: Gas-phase energy profiles for proton transfer: OM1 and OM1n

Comparison of MP2 (solid line), OM1 (dashed lines), and OM1n (dashed-dotted lines) gas-phase energy profiles for proton transfer between two water molecules in the Zundel ion for given oxygen-oxygen distances (R_{OO} in Å). See caption of Fig. S7 for conventions. The OM1 curve is right beneath the OM1n curve.

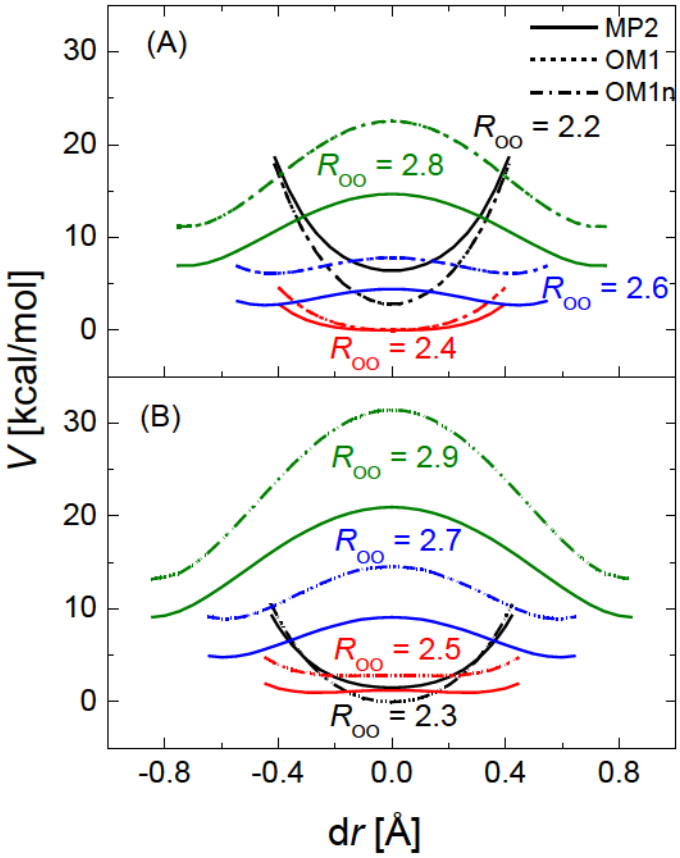


Figure S10: Gas-phase energy profiles for proton transfer: OM2 and OM2n

Comparison of MP2 (solid line), OM2 (dashed lines), and OM2n (dashed-dotted lines) gas-phase energy profiles for proton transfer between two water molecules in the Zundel ion for given oxygen-oxygen distances (R_{OO} in Å). See caption of Fig. S7 for conventions.

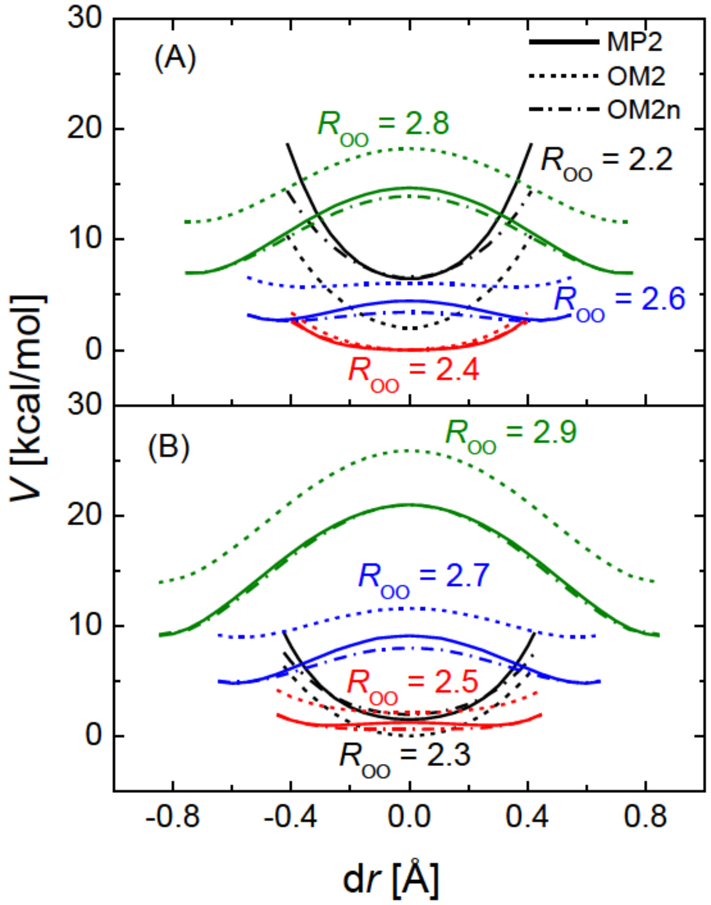


Figure S11: Gas-phase energy profiles for proton transfer: OM3 and OM3n

Comparison of MP2 (solid line), OM3 (dashed lines), and OM3n (dashed-dotted lines) gas-phase energy profiles for proton transfer between two water molecules in the Zundel ion for given oxygen-oxygen distances (R_{OO} in Å). See caption of Fig. S7 for conventions.

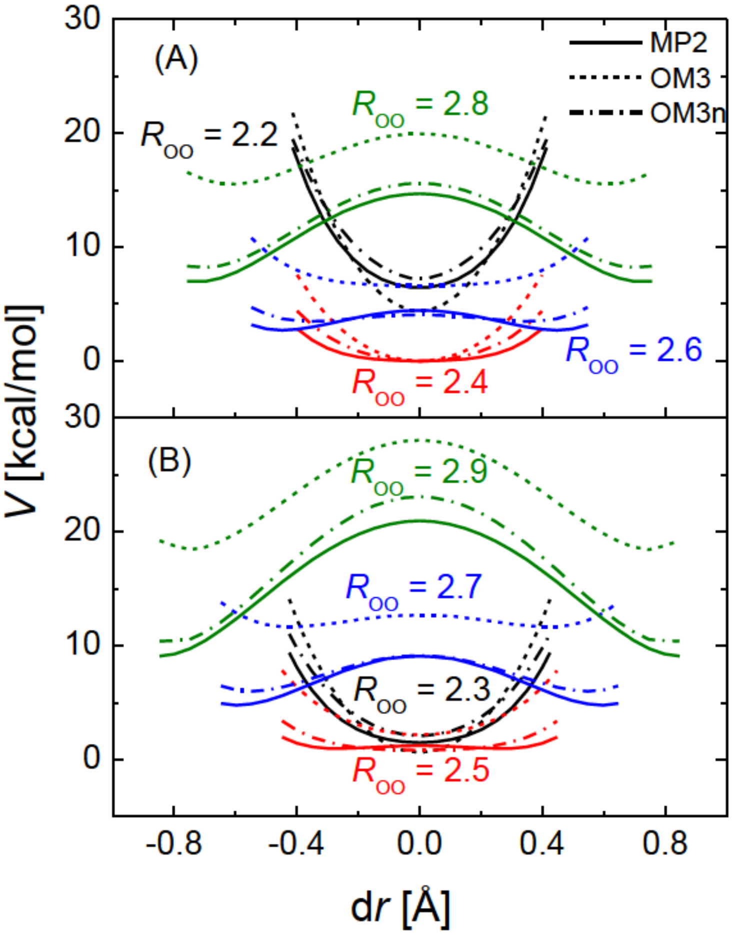


Figure S12: Water dimer geometries from relaxed surface scans

Water dimer geometries obtained in relaxed surface scans at the MP2/aug-cc-pVTZ level for (A) long ($R_{\mathrm{O1O2}} \geq 2.6$ Å) and (B) short ($R_{\mathrm{O1O2}} \leq 2.5$ Å) distances between the oxygen atoms. Different relative orientations were observed: (A) the H11-O1-H12 plane bisects the angle H21-O2-H22 with dihedral angle H11-O1-H12-O2 ~ 0°; (B) both water molecules are in the same plane with dihedral angle H11-H12-H21-H22 ~ 0°.

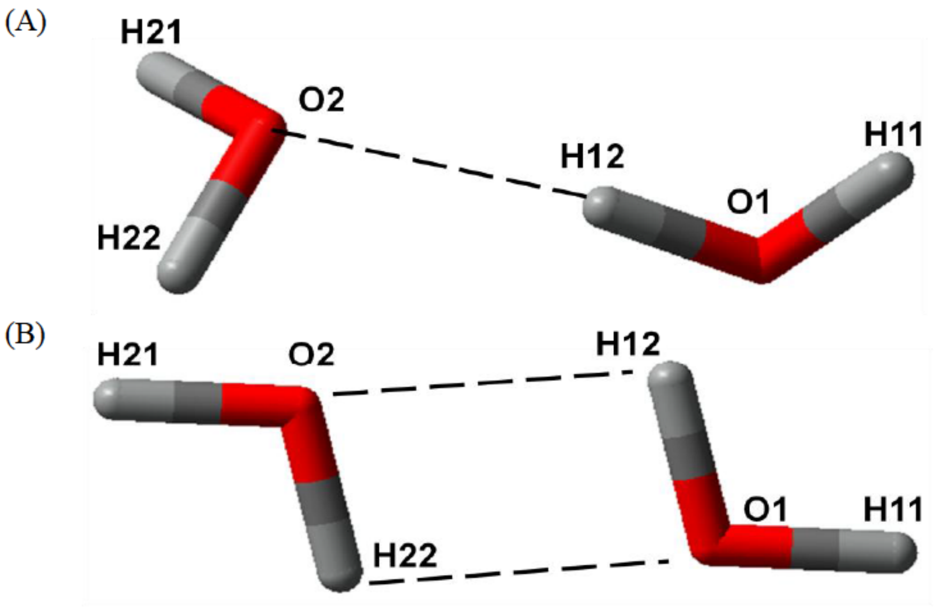


Figure S13: Distributions of the sampled geometries

Distributions of the sampled geometries as functions of (A) the distance $R_{\mathrm{O0O1'}}$ between the hydronium oxygen O0 and the nearest potential acceptor O1′; (B) the distance difference $\mathrm{d}r = \left|R_{\mathrm{O0H'}} - R_{\mathrm{O1'H'}}\right|$, where H′ denotes the hydrogen atom between O0 and O1′, and $R_{\mathrm{O0H'}}$ ($R_{\mathrm{O1'H'}}$) are the distances between O0 (O1′) and H′; and (C) the ratio ρ (see Fig. 4 for definition).

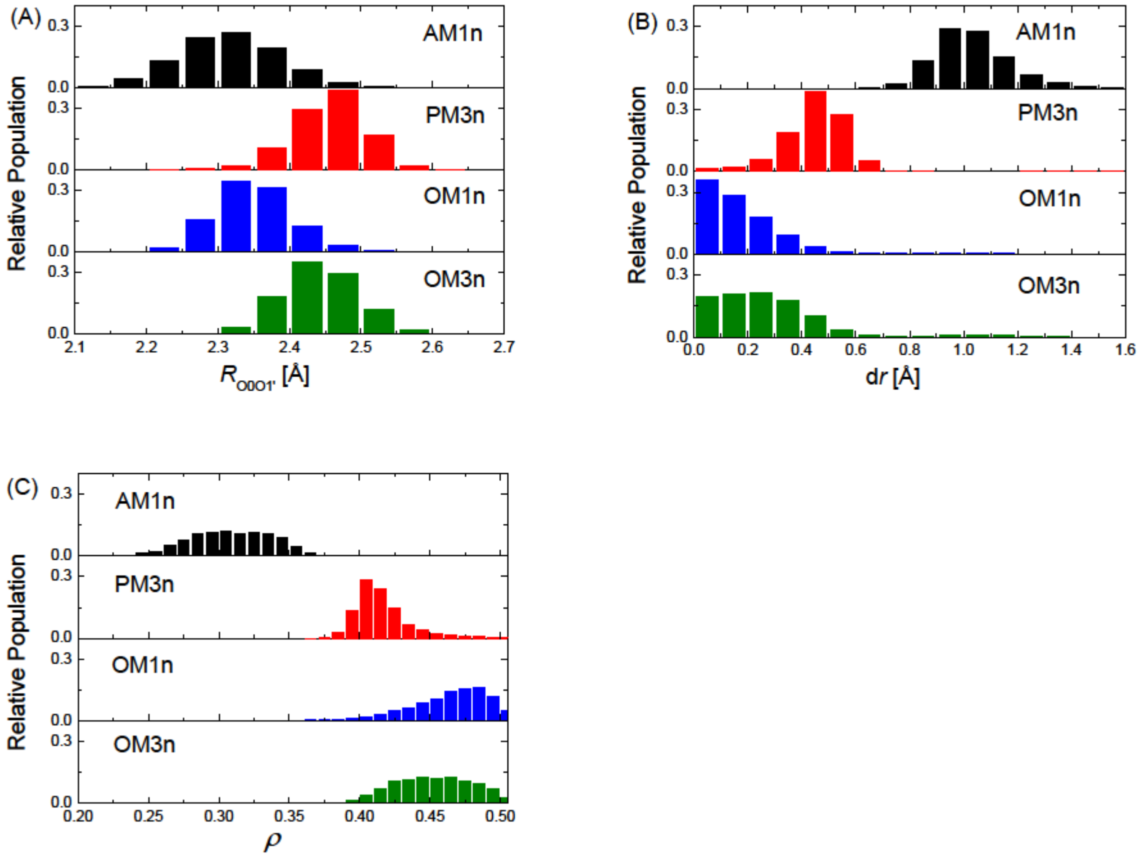


Figure S14: R_{O0O1} and $R_{O0O1'}$ at donor-acceptor swaps

Comparison of R_{O0O1} and $R_{O0O1'}$ when the donor and acceptor atoms swap, from (A) PM3n, (B) OM1n, and (C) OM3n simulations. Here, R_{O0O1} ($R_{O0O1'}$) is the distance between the donor O0 and the acceptor O1 (the nearest potential acceptor O1′). Usually O1 and O1′ are identical at the time of swapping: O1′ differs from O1 only in 1.6% of the swaps (5 out of 305) for PM3n, in 0.5% (50 out of 9364) for OM1n, and 2.7% (128 out of 4678) for OM3n.

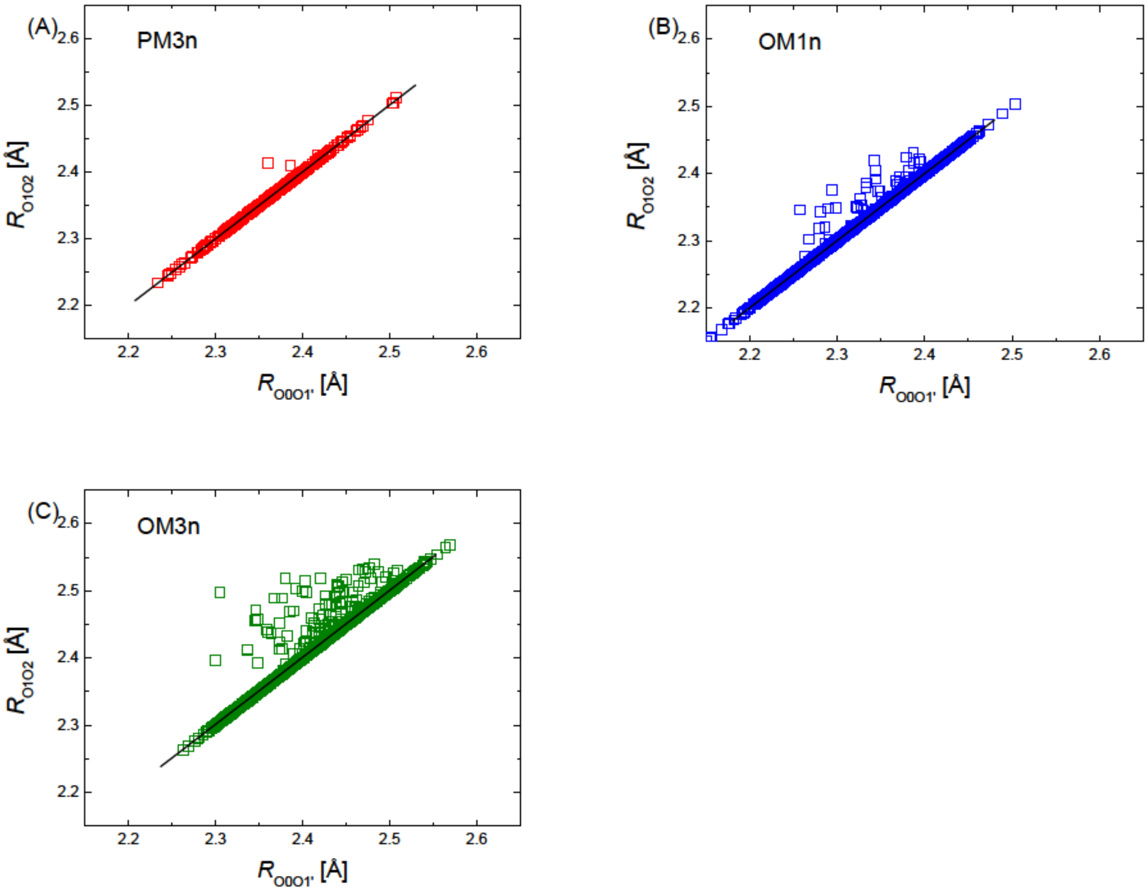


Figure S15: Distribution of R_{O0O1} at donor-acceptor swaps

Distribution of the donor-acceptor distance R_{O0O1} at the moment when the donor and acceptor swap. The average R_{O0O1} values are 2.37 Å, 2.33 Å, and 2.41 Å for PM3n, OM1n, and OM3n, respectively.

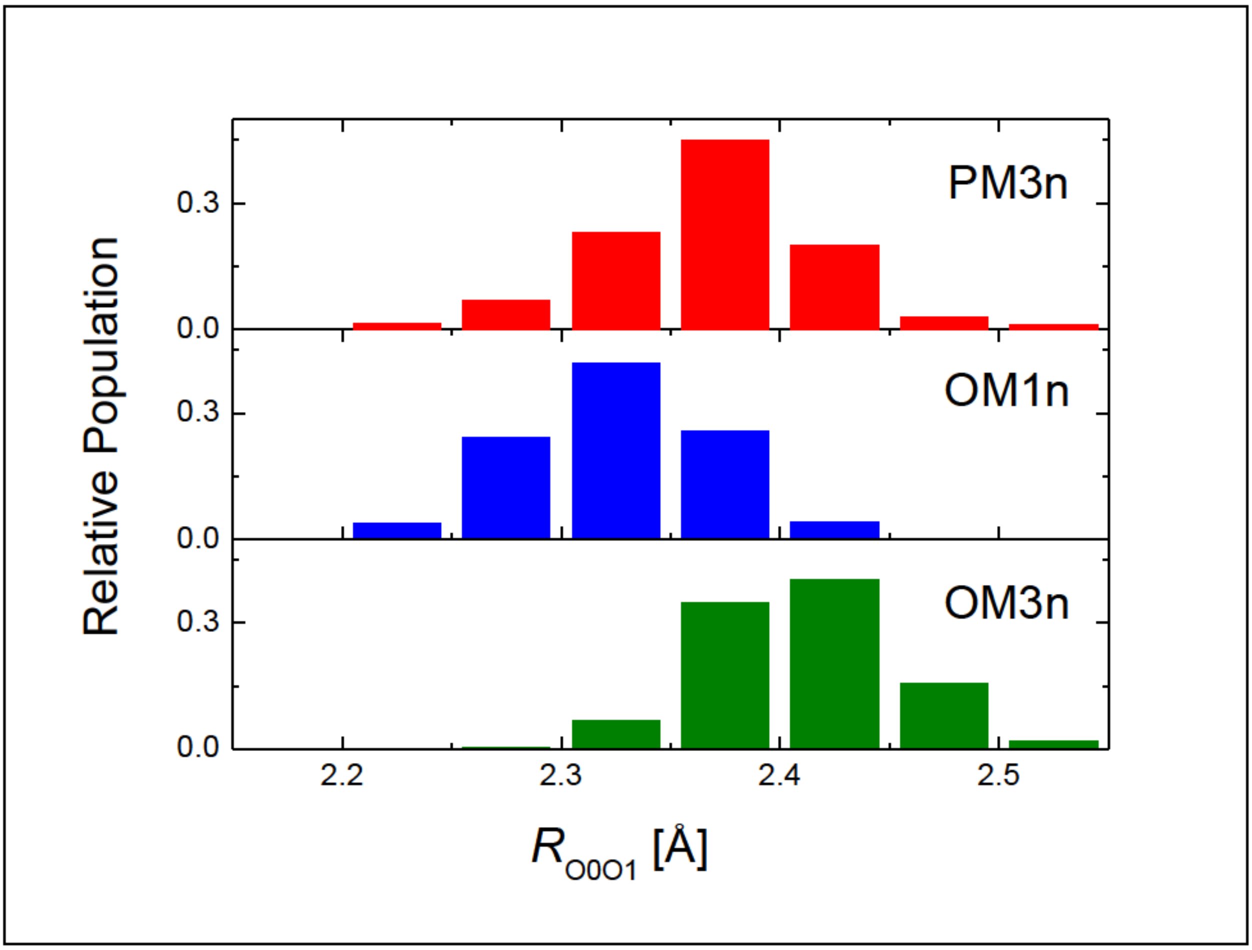


Figure S16: Integrated coordination number $n_{OO}(r)$ of water

Integrated coordination number $n_{OO}(r)$ for bulk water by the (A) SPC and TIP3P water models, (B) MNDO-based AM1 and PM3 models, and (C) orthogonalization-corrected OM1 and OM3 models. Results are shown for the authentic and the re-parameterized semi-empirical methods.

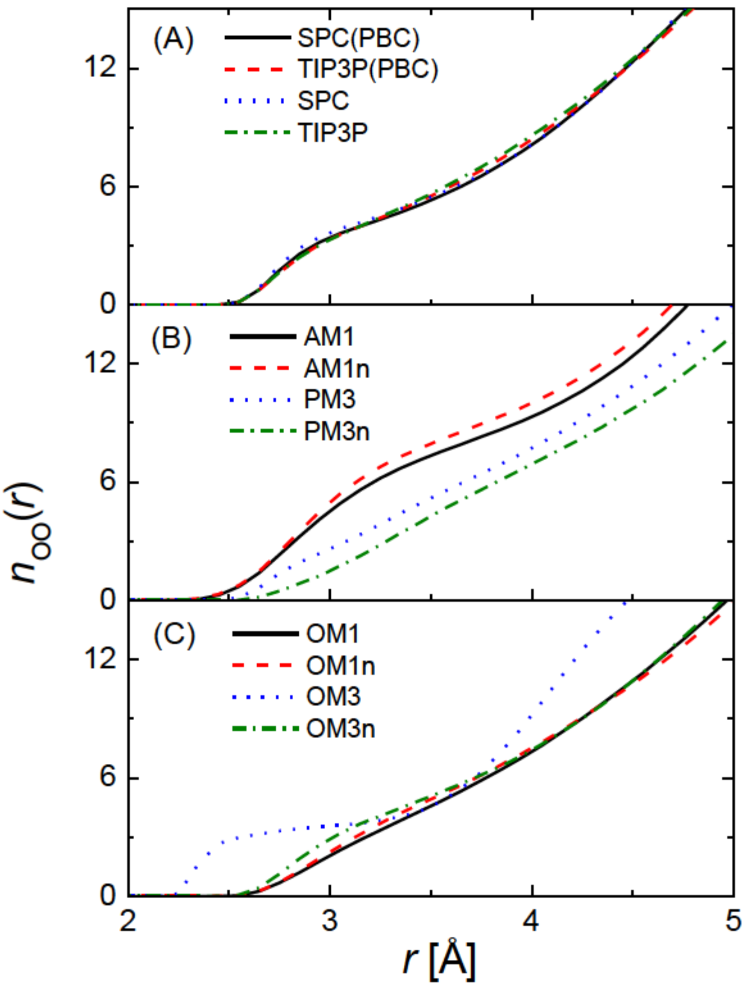


Figure S17: Integrated coordination number $n_{OH}(r)$ of water

Integrated coordination number $n_{OH}(r)$ for bulk water by the (A) SPC and TIP3P water models, (B) MNDO-based AM1 and PM3 models, and (C) orthogonalization-corrected OM1 and OM3 models. Results are shown for the authentic and the re-parameterized semi-empirical methods.

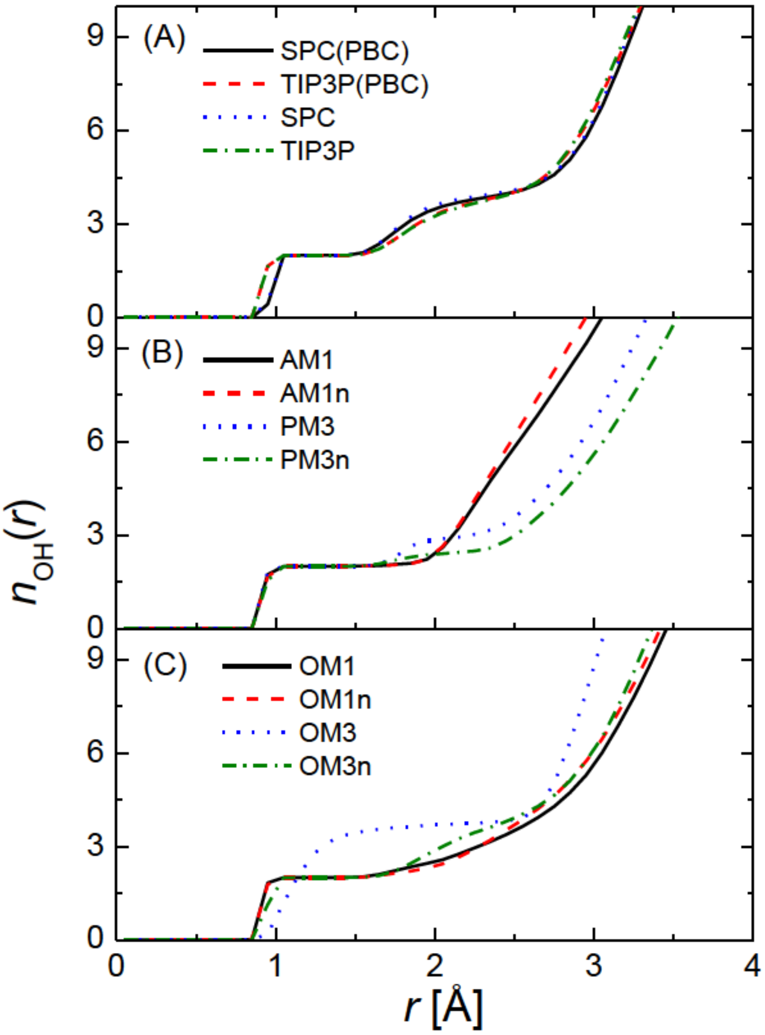
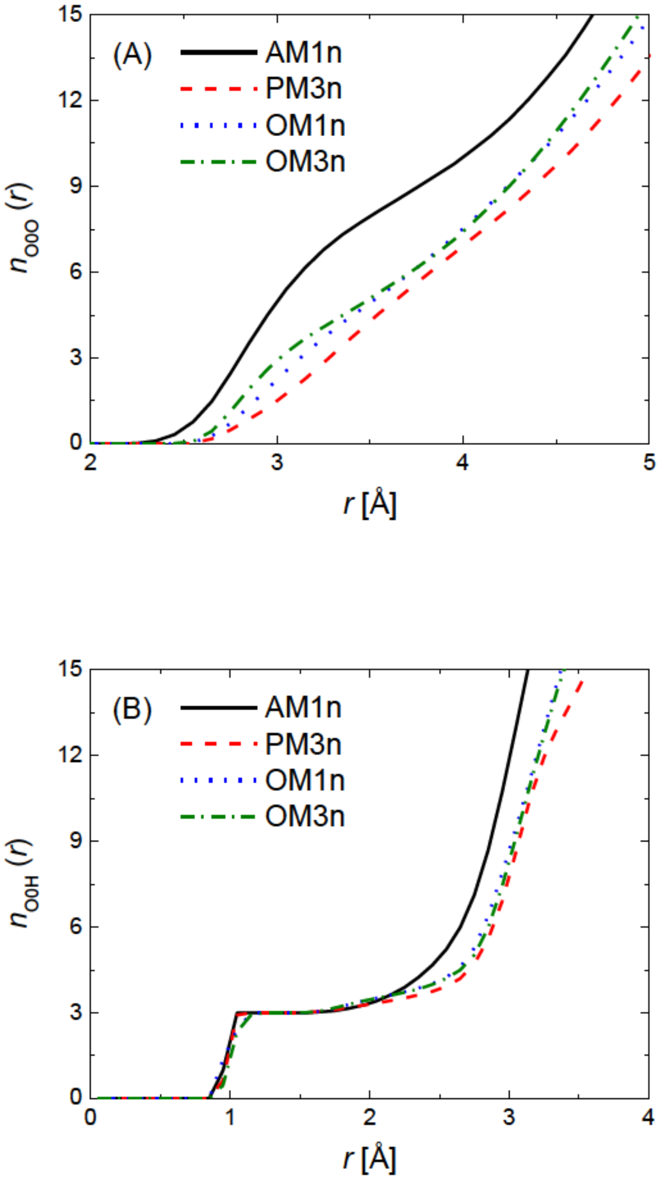


Figure S18: Integrated coordination numbers $n_{O0O}(r)$ and $n_{O0H}(r)$ of solvated proton

Integrated coordination numbers (A) $n_{O0O}(r)$ and (B) $n_{O0H}(r)$ for solvated proton obtained from the re-parameterized semi-empirical methods.


(A)
AM1n
PM3n
OM1n
OM3n
n_{O0O} (r)
r [Å]
(B)
AM1n
PM3n
OM1n
OM3n
n_{O0H} (r)
r [Å]