Linear scaling conjugate gradient density matrix search: Implementation, validation, and application with semiempirical molecular orbital methods.

Inaugural Dissertation

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

vorgelegt von
Rouslan Kevorkiants
aus Russland

Düsseldorf 2003

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: 11.06.2003

Contents

Chapter 1

Introduction

Over the past decade it has been a common goal in theoretical chemistry to perform calculations on ever larger molecules. The improvements in hardware performance (by about one order of magnitude every five years) are not sufficient to reach this objective since the available methods in their conventional implementations show a steep scaling of the computational effort with molecular size. If the latter is characterized by a parameter N (e.g., the number of atoms or the number of electrons), the computation times of standard quantum-chemical calculations scale formally as $O(N^3)$ for semiempirical molecular orbital (MO) methods, $O(N^4)$ for ab initio MO or density functional methods, and at least $O(N^5)$ for correlated ab initio methods. To extend quantum-chemical treatments to large molecules with thousands of atoms, it is therefore mandatory to develop new methods and techniques that overcome this prohibitive scaling behavior.

One such approach involves the combination of quantum mechanics (QM) and molecular mechanics (MM). The QM/MM methods [1,2,3] (for reviews see [4,5,6,7]) partition a large system into a small active center where electronic events such as chemical reactions or electronic excitations take place, and an environment that influences these events. The active center is then treated at an appropriate QM level (as accurately as needed) whereas the environment is described at the classical MM level. This pragmatic strategy allows computations for large systems because the expensive QM calculation with an adverse scaling is restricted to a relatively small region while modelling the effects of the environment through an efficient MM treatment.

An alternative approach is the attempt to achieve linear scaling of the computational effort in pure QM calculations. This concept rests on the principle of locality or "nearsightedness" [8]: most of the chemically relevant interactions (e.g., covalent bonding) are short-ranged, and it must therefore be possible to describe them by O(N) algorithms which exploit this locality through suitable

truncation schemes, while the long-range electrostatic interactions can be handled separately by special techniques.

Linear scaling algorithms have been proposed for most standard quantum-chemical methods (for reviews see [9,10,11,12,13]). Generally speaking, linear scaling code is required both for the computation of the Hamiltonian matrix (including integral evaluation) and the determination of the wave function or the density from a given Hamiltonian matrix. In ab initio methods and density functional theory (DFT) both these issues need to be addressed, whereas linear scaling developments for semiempirical and tight-binding MO schemes normally focus on the second issue (since integral evaluation is very efficient and usually not a limiting factor in practice).

The present work describes the implementation and application of linear scaling techniques in semiempirical quantum-chemical methods. At the outset of this project, three such linear scaling approaches had already been discussed in the literature: the divide-and-conquer (DC) method originally introduced by Yang [14] and later extended to semiempirical methods [15,16], the localized molecular orbital (LMO) of Stewart [17], and the direct minimization of the density matrix originally proposed for tight-binding (TB) methods by Li, Nunes and Vanderbilt [18] and later implemented for semiempirical methods by Scuseria [19]. Each of these approaches has its merits, but the latter seems most attractive: the conjugate gradient density matrix search (CG-DMS) employs reliable and well-established minimization procedures and offers a transparent route towards linear scaling through the use of cutoffs in the density matrix and the Fock matrix. We have therefore chosen this approach. Our main goal was to provide an efficient and robust implementation of CG-DMS in our semiempirical code [20] such that linear scaling MO calculations can be done for all semiempirical Hamiltonians available in this code (including new methods with explicit orthogonalization corrections [21,22,23]). From an application-oriented point of view, this development was motivated by the desire to treat enzymes and enzymatic reactions both at the QM/MM and the pure QM level.

This thesis is organized as follows. After reviewing the theoretical background of semiempirical MO methods (section 2) the linear scaling algorithms for obtaining self-consistent-field (SCF) solutions are described in some detail (section 3). Our CG-DMS implementation is reported in section 4 which addresses a number of technical issues such as sparse matrix handling and SCF convergence. Test calculations on small molecules have been used to establish the correctness of our implementation and to determine optimum CG-DMS control parameters (section 5). After validating the code and checking its performance for several series of larger test molecules (section

6) it has been applied to study minima and transition states in two enzymatic reactions (section 7). The thesis concludes with a brief summary and outlook (section 8).

Chapter 2

Theoretical background

2.1 Ab initio molecular orbital theory

We start from the nonrelativistic time-independent Schrödinger equation and introduce the Born-Oppenheimer approximation to separate electronic and nuclear motion. The resulting electronic Schrödinger equation (Hamiltonian $\mathbf{H}_{el}$, wave function Ψ_{el}, electronic energy E_{el})

$$\mathbf{H}_{el}\,\Psi_{el} = E_{el}\,\Psi_{el} \tag{1}$$

needs to be solved at given nuclear coordinates. In orbital approximation, the n-electron wave function Ψ_{el} is represented as an antisymmetric Slater determinant built from one-electron orbitals which are determined variationally (Hartree-Fock method). In the case of molecules, these orbitals ψ_i are normally written as linear combination of atomic orbitals φ_μ (LCAO-MO ansatz):

$$\psi_i = \Sigma_\mu\, C_{\mu i}\, \varphi_\mu \tag{2}$$

Variational minimization of the electronic energy E_{el} with respect to the coefficients $C_{\mu i}$ leads to the Roothaan-Hall-equations:

$$\mathbf{F\,C = S\,C\,E} \tag{3}$$

where **C** denotes the coefficient matrix, **E** is a diagonal matrix containing the orbital energies ε_i, and **S** is the overlap matrix with the elements

$$S_{\mu\nu} = \langle\varphi_\mu|\varphi_\nu\rangle \tag{4}$$

Using standard notation and atomic units, the Fock matrix elements for closed-shell systems are given by

$$F_{\mu\nu} = H_{\mu\nu} + G_{\mu\nu} \tag{5}$$

$$G_{\mu\nu} = \Sigma_\lambda \Sigma_\sigma P_{\lambda\sigma} [<\mu\nu|\lambda\sigma> - 1/2 <\mu\lambda|\nu\sigma>] \tag{6}$$

The one-electron integrals $H_{\mu\nu}$ include the kinetic energy term and the nucleus-electron attractions, while the two-electron integrals $<\mu\nu|\lambda\sigma>$ represent the repulsion between the corresponding charge distributions. The density matrix elements $P_{\lambda\sigma}$ are defined by the following summation over occupied orbitals:

$$P_{\lambda\sigma} = 2 \Sigma_i^{Nocc} C_{\lambda i} C_{\sigma i} \tag{7}$$

The electronic energy is given by:

$$E_{el} = 1/2 \Sigma_\mu \Sigma_\nu P_{\mu\nu} (H_{\mu\nu} + F_{\mu\nu}) \tag{8}$$

The total energy is obtained by adding the Coulomb repulsion energy between the nuclei to the electronic energy.

Two points are obvious from this brief overview:

(a) For N basis functions, there are $O(N^4)$ two-electron integrals, and the computational effort will thus formally scale as $O(N^4)$.

(b) An iterative solution of the secular equations (3) is required since the computation of the Fock matrix according to eqs. (5)-(7) makes use of the LCAO coefficients that are determined from eq. (3). Each such SCF iteration involves the solution of a generalized eigenvalue problem, typically through diagonalization, which is known to scale as $O(N^3)$ in computational effort.

In conventional implementations of ab initio MO theory, the most demanding steps are the computation and the handling of the $O(N^4)$ two-electron integrals.

2.2 Semiempirical molecular orbital methods

Semiempirical MO methods introduce integral approximations to neglect most of the small integrals that appear in the ab initio MO formalism. To compensate for the errors caused by these approximations, the remaining integrals are described by parametric expressions and then calibrated against reliable experimental or theoretical reference data.

All semiempirical methods are based on the zero-differential-overlap (ZDO) approximation which is invoked to a different extent in different schemes. We shall only consider the most advanced NDDO variant (neglect of diatomic differential overlap) where products of basis functions depending on the same electronic coordinates are neglected if they are located on different atoms. This has the following consequences:

- The overlap matrix **S** is a unit matrix.
- All three-center one-electron integrals vanish.
- All three-center and four-center two-electron integrals are neglected.

The secular equations thus assume a simplified form

$$\mathbf{F\,C = C\,E} \tag{9}$$

but their iterative solution still requires $O(N^3)$ steps. On the other hand, only one-center and two-center integrals are retained at the NDDO level, and therefore integral evaluation and integral processing during the formation of the Fock matrix scales as $O(N^2)$, and is no longer the computational bottleneck. Using the convention that the basis functions μ and ν are assigned to atom A, and basis functions λ and σ are assigned to atom B, the closed-shell NDDO Fock matrix elements can be written as:

$$\begin{aligned} F_{\mu\mu} &= H_{\mu\mu} && \text{one-electron part} \\ &+ \Sigma_\nu \left[P_{\nu\nu} \langle\mu\mu|\nu\nu\rangle - 1/2\, P_{\nu\nu} \langle\mu\nu|\mu\nu\rangle\right] && \text{two-electron part} \\ &+ \Sigma_{B\neq A} \Sigma_\lambda \Sigma_\sigma P_{\lambda\sigma} \langle\mu\mu|\lambda\sigma\rangle \end{aligned} \tag{10}$$

$$\begin{aligned} F_{\mu\nu} &= H_{\mu\nu} && \text{one-electron part} \\ &+ 1/2\, P_{\mu\nu} \left[\, 3 \langle\mu\nu|\mu\nu\rangle - \langle\mu\mu|\nu\nu\rangle\right] && \text{two-electron part} \\ &+ \Sigma_{B\neq A} \Sigma_\lambda \Sigma_\sigma P_{\lambda\sigma} \langle\mu\nu|\lambda\sigma\rangle \end{aligned} \tag{11}$$

$$F_{\mu\lambda} = H_{\mu\lambda} \qquad \text{one-electron part}$$
$$-1/2 \, \Sigma_\nu \Sigma_\sigma P_{\mu\lambda} \langle \mu\nu|\lambda\sigma \rangle \qquad \text{two-electron part} \qquad (12)$$

Eqs. (9)-(12) specify the theoretical model that underlies the established semiempirical valence-electron methods MNDO [24], AM1 [25] and PM3 [26,27]. These methods employ the same strategies and the same parametric expressions to represent the integrals in eqs. (10)-(12). Briefly, the one-center terms are treated as adjustable parameters or are determined empirically from atomic spectra. The two-center two-electron integrals are evaluated from a semiempirical multipole-multipole interaction model with the correct asymptotic behavior at small and large distances. The two-center core-core repulsions and the two-center one-electron attraction integrals are expressed in terms of certain two-center two-electron integrals, and the two-center one-electron resonance integrals are parametric functions that are taken to be proportional to the overlap integrals. Hence, MNDO, AM1 and PM3 share the same theoretical model: they differ only in the parametric expressions for the core-core repulsions (more flexible in AM1 and PM3), and of course in the actual values of the optimized parameters.

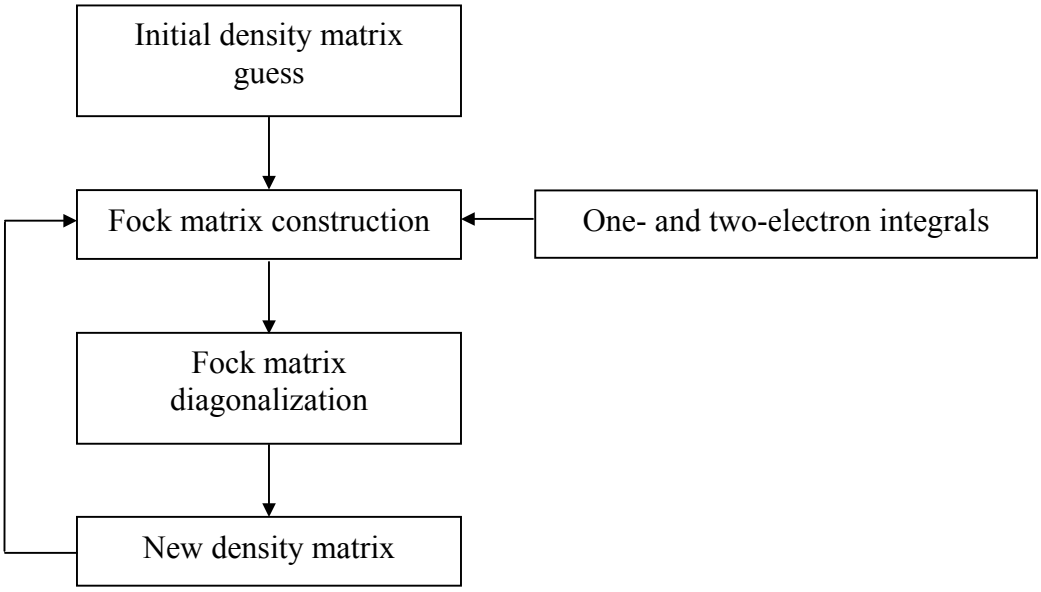
The more recent semiempirical methods OM1 [28], OM2 [22], and OM3 [23] also obey eqs. (9)-(12), but they employ different representations for the terms in eqs. (10)-(12) and incorporate explicit orthogonalization corrections to the one-electron integrals.

In the context of linear scaling, the solution of the secular equations needs to be addressed first since this is the dominating $O(N^3)$ step for all of the above NDDO-based methods. In the following sections we shall focus on implementing an alternative approach (CG-DMS) to replace the $O(N^3)$ diagonalization procedure. However, to arrive at truly linear scaling, it is also necessary to avoid the $O(N^2)$ operations during integral evaluation and formation of the Fock matrix, particularly with regard to the two-center two-electron integrals. Corresponding algorithms such as fast multipole methods or tree codes are known (for reviews see [9,11b]) which are based on a hierarchical partition of the electron density and the use of multipole expansions for the interactions between well-separated partitions. We have not yet implemented these algorithms in our semiempirical code, but have included the option of using simple cutoffs for the two-center two-electron integrals (see also [15,17,19,29]) which has been sufficient for the molecules studied presently.

2.3 Conventional SCF-MO approach

As mentioned above, the secular equations must be solved iteratively both in ab initio and semiempirical MO theory, see eqs. (3)-(7) and (9)-(12). Since the one-electron and two-electron integrals remain unchanged during these iterations, they can be precomputed. The conventional implementation of a semiempirical MO calculation thus involves the following steps (see also Scheme 1):

a) Calculation of one-electron and two-electron integrals.
b) Formation of an initial guess for the density matrix.
c) Formation of the Fock matrix, eqs. (10)-(12).
d) Fock matrix diagonalization to solve the eigenvalue problem, eq. (9).
e) Formation of a new density matrix, eq. (7).
f) Check for convergence, otherwise start next iteration with step (c).



Scheme 1. Iterative solution of Hartree-Fock LCAO-MO equations.

There is no guarantee that the fixed-point iterations outlined above will converge to a self-consistent solution. Experience indicates that convergence is normally not a problem in semiempirical MO calculations which employ minimal valence-electron basis sets with limited flexibility. Convergence is usually fast for closed-shell molecules in undistorted geometries, but problems may occasionally arise for open-shell species or transition metal compounds. Since SCF convergence will be an issue in CG-DMS (see section 4) we briefly address some procedures that are used in conventional SCF-MO calculations to assist and accelerate SCF convergence [30,31].

Extrapolation: The idea is to form a new Fock matrix, during selected SCF steps, not just from the current density matrix, but from a modified density matrix obtained by extrapolating those from the preceding SCF iterations. If this extrapolation is done properly, the number of required SCF iterations will be reduced. A common practice is to use up to three previous density matrices for extrapolation.

Damping: This technique aims at eliminating problems due to oscillations. If the density matrix from iteration k tends to be very close to that of iteration k+2, and quite different from that of iteration k+1, the variations can be damped by employing (during Fock matrix formation) the linear combination $\alpha*P_k+ (1-\alpha)*P_{k+1}$ instead of the current density matrix. This may overcome convergence problems. The mixing parameter α can either be constant or changed dynamically (decreased) during the SCF procedure.

Level shifting [32]: Between two SCF iterations, there is orbital mixing between occupied and virtual orbitals. Convergence problems may arise if this mixing is too pronounced and causes oscillations. A remedy is to artificially increase the energy of the virtual orbitals and thereby reduce this mixing. This will make the SCF iterations smoother and thus assist convergence in difficult cases, but may also reduce the rate of convergence in uncritical cases.

Direct inversion in the iterative subspace (DIIS) [33,34,35]: This is an extrapolation technique which accelerates convergence and moreover may lead to convergence even in difficult cases where other procedures fail. It focuses on the sequence of Fock matrices that are formed during the SCF treatment, and computes an error matrix at each iteration (FPS-SPF in the ab initio case, FP-PF in the semiempirical case). It proceeds to determine the linear combination of previously available Fock matrices that minimizes the error function (via the solution of an appropriate linear system of equations) and then uses this linear combination as the next Fock matrix in the SCF procedure.

The relative cost for the DIIS extrapolation is negligible in ab initio MO calculations, but substantial in the semiempirical case. Therefore, semiempirical MO calculations normally employ the simpler techniques for SCF convergence acceleration first (see above) and turn to DIIS only if these simpler techniques do not work.

Another important issue for SCF convergence is the initial guess of the density matrix. Obviously, a high-quality initial guess will facilitate and speed up convergence. This is particularly relevant for ab initio and density functional calculations with large basis sets where it is established practice to

derive a realistic initial density matrix from lower-level calculations. Since semiempirical MO methods are normally rather robust with regard to SCF convergence, one usually starts from a diagonal density matrix guess (with the correct number of electrons distributed such that the atoms tend to be neutral). A slightly more sophisticated guess can be obtained by diagonalizing the one-electron core Hamiltonian matrix. In the context of the linear scaling LMO approach [17] it has been suggested to start from localized bonding, nonbonding, and antibonding orbitals generated from the classical Lewis structures.

Chapter 3

Linear scaling SCF approaches

3.1 Overview

In conventional semiempirical SCF-MO treatments, the solution of the secular equations through matrix diagonalization is the computational bottleneck with an $O(N^3)$ scaling. As already mentioned, several linear scaling algorithms have been proposed in recent years as a replacement for matrix diagonalization (for reviews see [10,11b,12]). In this section we shall first give an overview over the available methods and then describe those in more detail that are of particular relevance for semiempirical SCF-MO treatments.

The linear scaling SCF approaches focus on the density matrix. Due to the short-range nature of quantum interactions, the density matrix is known [10,13,36] to decay exponentially for systems with a HOMO-LUMO gap E_{gap} (molecules, insulators):

$$\rho(\mathbf{r},\mathbf{r'}) \sim \exp (-\sqrt{E_{gap}}) \mid \mathbf{r}-\mathbf{r'} \mid) \qquad (13)$$

Consequently, the discrete representation $P_{\mu\nu}$ of the density matrix in a basis of local functions ϕ has similar localization properties and must form a sparse matrix for large systems.

$$\rho(\mathbf{r},\mathbf{r'}) = \Sigma_{\mu\nu} P_{\mu\nu} \phi_\mu (\mathbf{r}) \phi_\nu (\mathbf{r'}) \qquad (14)$$

For an approximate description of such systems, we thus need to determine only the non-negligible density matrix elements $P_{\mu\nu}$ whose number will scale linearly for a sufficiently large system. Broadly speaking, there are two general strategies available [11b]: local Hamiltonian and variational principle approaches.

In the first case, local Hamiltonians and local properties are defined and determined within a local region, and the global properties are then derived therefrom. The best example for this approach is the divide-and-conquer method [14,15,16,29,37,38,39,40] which constructs the global density of the system from local densities that are determined for the chosen subsystems by standard separate calculations for each subsystem and its immediate surroundings (buffer regions). There are other such methods [11b], most notably the Fermi operator expansion method [10]: in this case, the local Hamiltonian is constructed by truncation in the atomic orbital space, and the density matrix is obtained by an iterative diagonalization based on the Chebyshev polynomial approximation [10].

Variational principle approaches determine the density matrix directly by minimizing a suitable energy functional with regard to the density matrix (while imposing the required constraints of normalization and idempotency); linear scaling is achieved by truncating the density matrix (i.e., computing only the non-negligible elements above a certain threshold). Such direct minimization of the density matrix has first been suggested and implemented for simple tight-binding approaches [18,41,42,43,44] and has later been extended to self-consistent DFT methods [45,46], semiempirical MO methods [19,47] and ab initio MO methods [46,48,49]. These minimizations usually employ steepest descent or conjugate gradient algorithms, but a quadratically convergent scheme has also been introduced [48]. The idempotency constraint is normally imposed through a McWeeny purification transformation [50] (see also [51,52,53,54] for possible generalizations) or alternatively through the use of penalty functions [8,55]. Another recent method for the direct optimization of the atomic-orbital density matrix makes use of an exponential parameterization in the framework of Hartree-Fock and Kohn-Sham theory [56,57,58]. Finally, it should be noted that some of the variational principle approaches are formulated not in terms of the density matrix, but in terms of localized orbitals (see the reviews [10,11b,12,13] for further details).

In the following we shall describe three representative linear scaling approaches for semiempirical MO methods in more detail: the divide-and-conquer technique [15,29,59] and the conjugate gradient density matrix search [19,47] are examples for the two general strategies outlined above, while the local molecular orbital approach [17] is essentially a linear scaling variant of the pseudodiagonalization scheme [60] that is commonly used in semiempirical codes. Comparisons between different approaches are found in the literature [47,61,62,63].

3.2 Divide-and-conquer method

The basic idea in the divide-and-conquer method is to divide a system into disjoint subsystems α which contain only a small number of atoms. A partition matrix $p^{\alpha}_{\mu\nu}$ is defined in the space of atomic orbitals as follows [11b]:

$$
\begin{array}{llllll}
p^{\alpha}_{\mu\nu} = 1 & \text{if} & \mu \in \alpha & \text{and} & \nu \in \alpha & \\
p^{\alpha}_{\mu\nu} = 1/2 & \text{if} & \mu \in \alpha & \text{and} & \nu \notin \alpha & (15) \\
p^{\alpha}_{\mu\nu} = 1/2 & \text{if} & \mu \notin \alpha & \text{and} & \nu \in \alpha & \\
p^{\alpha}_{\mu\nu} = 0 & \text{if} & \mu \notin \alpha & \text{and} & \nu \notin \alpha &
\end{array}
$$

where $\mu \in \alpha$ means that the atomic orbital φ_μ is located at an atom in subsystem α. The partition matrix is normalized,

$$\Sigma_\alpha \, p^{\alpha}_{\mu\nu} = 1, \tag{16}$$

and connects the subsystem density matrix P^α to the global density matrix P:

$$P^{\alpha}_{\mu\nu} = p^{\alpha}_{\mu\nu} \, P_{\mu\nu} \tag{17}$$

$$P_{\mu\nu} = \Sigma_\alpha \, P^{\alpha}_{\mu\nu} \tag{18}$$

The subsystem density matrices are not computed from the global eigenvectors of the system, but from the local eigenvectors C^{α}_{i} of the subsystem and the surrounding buffer region (local SCF region, see Figure 1).

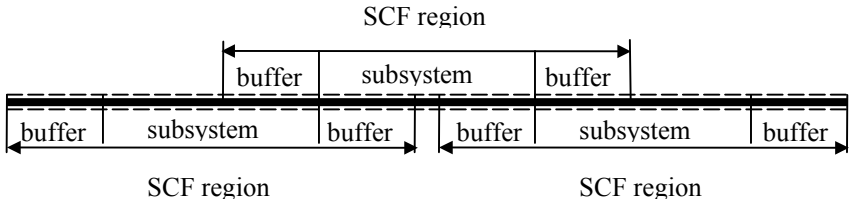


Figure 1. Example of system subdivision in divide-and-conquer method.

They are given by

$$P^{\alpha}_{\mu\nu} = p^{\alpha}_{\mu\nu} \, \Sigma_i \, n^{\alpha}_{i} \, C^{\alpha}_{\mu i} \, C^{\alpha}_{\nu i} \tag{19}$$

The local eigenvectors C^{α}_{i} and the associate local eigenvalues ε^{α}_{i} are obtained from a standard MO calculation for the local SCF region (subsystem plus buffer) by solving secular equations of the type:

$$\mathbf{F}^{\alpha}\,\mathbf{C}^{\alpha} = \mathbf{C}^{\alpha}\,\mathbf{E}^{\alpha} \tag{20}$$

The occupation number n^{α}_{i} for each localized eigenstate is approximated by a Fermi function with a low but finite temperature T:

$$n^{\alpha}_{i} = 2 / [\, 1 + \exp (-x / k\,T) \,] \tag{21}$$

$$x = \varepsilon_{F} - \varepsilon^{\alpha}_{i} \tag{22}$$

where ε_{F} is the Fermi energy that is common to all of the subsystems. It is determined such that the global density matrix yields the correct number of electrons for the entire system. The requirement of a common Fermi energy enables electron flow between different subsystems and ensures the proper normalization of the electron density.

Linear scaling is achieved in this approach by the trivial fact that an enlargement of the system leads to a linear increase in the number of subsystems (assuming a consistent partitioning), and therefore the effort for computing the subsystem density matrices will also increase linearly. The overhead for assembling the global density matrix and for computing the total energy and gradient is small and can likewise be handled in a linear scaling fashion [11].

It is obvious that the divide-and-conquer approach neglects all density matrix elements that do not simultaneously belong to any local SCF region (see Figure 1) so that the global density matrix is sparse. In a given iteration of the divide-and-conquer SCF procedure, the following steps are performed (see [11a] for a flowchart):

(a) use the currently available sparse global density matrix to construct a corresponding sparse global Fock matrix;
(b) compute the electronic energy and check for convergence;
(c) extract the appropriate elements to form local Fock matrices for the local SCF regions;
(d) solve the Roothaan-Hall equations for these local SCF regions, eq. (20);
(e) construct the subsystem density matrices, eq. (19), from the computed local eigenvectors and eigenvalues using the Fermi energy of the previous iteration;

(f) generate a new sparse global density matrix, eq. (18);

(g) determine the new Fermi energy from the normalization requirement.

The divide-and-conquer method is an approximate method since it neglects the density matrix elements between distant atoms and truncates other matrix elements in an analogous manner [11]. Its accuracy can be improved systematically by enlarging the subsystems and/or the buffer regions. For better control of the accuracy, one may also use dual buffer regions [11a,29]. The convergence of the DC results towards the exact results from full matrix diagonalization and the influence of various computational parameters (subsystem and buffer, cutoffs, temperature) have been investigated in several papers [11,15,16,29,40,59] so that divide-and-conquer semiempirical MO calculations have now become a well-established technique. Recent applications include geometry optimizations for proteins [40], investigations of the active site in enzymatic reactions [64,65], solvation energies and electrostatic potentials in DNA [66], charge fluctuations in DNA and RNA [67], and studies of charge transfer between biomolecules and solvent [68].

3.3 Conjugate gradient density matrix search

It has long been recognized [50,69] that the eigensolution of the Roothaan-Hall equations can be avoided by a direct minimization of the electronic energy with respect to the density matrix (under certain constraints). For the sake of convenience, we formulate the following derivations in terms of the one-electron atomic-orbital density matrix **P** that does not contain the closed-shell occupancy factor of 2, see eq. (7), and that is thus related to the occupied LCAO-MO coefficients $\mathbf{C}_{occ}$ by

$$\mathbf{P} = \mathbf{C}_{occ}\, \mathbf{C}^{T}_{occ} \tag{23}$$

where a superscript T denotes the transpose of a matrix. This density matrix is symmetric, normalized to the number of electrons (N_e), and idempotent:

$$\mathbf{P} = \mathbf{P}^{T} \tag{24}$$

$$\mathrm{Tr}\,(\,\mathbf{P}\,) = N_e / 2 \tag{25}$$

$$\mathbf{P}\,\mathbf{P} = \mathbf{P} \tag{26}$$

The trace Tr (**P**) of matrix **P** is the sum of its diagonal elements. The idempotency condition, eq. (26), follows from the orthonormality of the LCAO-MO eigenvectors:

$$\mathbf{P}\,\mathbf{P} = \mathbf{C}_{occ}\,(\,\mathbf{C}^{T}_{occ}\,\mathbf{C}_{occ}\,)\,\mathbf{C}^{T}_{occ} = \mathbf{C}_{occ}\,\mathbf{C}^{T}_{occ} = \mathbf{P} \qquad (27)$$

The three conditions, eqs. (24)-(26), must be fulfilled by any density matrix that represents a closed-shell determinant [50,69], and will therefore serve as constraints.

The electronic energy, eq. (8), can be written as the trace of the following matrix product:

$$E_{el} = Tr\,[\,(\,\mathbf{h} + \mathbf{F}\,)\,\mathbf{P}\,] \qquad (28)$$

involving the one-electron core Hamiltonian matrix **h** and the Fock matrix **F** (the latter depending on **P**). First-order variation of **P** [50] leads to the stationarity condition

$$\varepsilon = 2\ Tr\,(\,\mathbf{F}\,\mathbf{P}\,) = \text{stationary} \qquad (29)$$

subject to the constraints of eqs. (24)-(26), with the Fock matrix **F** being regarded as fixed during the variation. This implies [50,69] that the solution of the Roothaan-Hall generalized eigenvalue problem, eq. (3), is equivalent to the optimization of the sum of the orbital energies of the occupied MOs (subject to the orthonormality constraints on the occupied MOs).

The normalization constraint, eq. (25), can easily be incorporated into eq. (29) through a Lagrangian term. The idempotency constraint, eq. (26), is harder to satisfy. The first practical solution to this problem was achieved in direct minimizations at the tight-binding level [18,41] which imposed the idempotency constraint implicitly through substitution of the McWeeny purification [50]

$$\mathbf{P} \rightarrow 3\,\mathbf{PP} - 2\,\mathbf{PPP} \quad \text{(for orthogonal basis functions)} \qquad (30)$$

into eq. (29). This purification transform brings an approximately idempotent matrix closer to idempotency through a process that is quadratically convergent upon fixed-point iteration [50]. The form of eq. (30) is derived from a steepest-descent minimization of the trace Tr[(**PP**-**P**)(**PP**-**P**)] which must become zero [50]. The convergence properties of this transform are best studied [69] by considering the scalar fixed-point iteration

$$x_{n+1} = f(x_n) \quad \text{with } f(x) = 3 x^2 - 2 x^3. \tag{31}$$

This function has a minimum at x=0 and a maximum at x=1, and it is easily verified that the sequence (31) will convergence to x=0 from above for starting values between $(1 - \sqrt{3})/2$ and 1/2, and to x=1 from below for starting values between 1/2 and $(1 + \sqrt{3})/2$. The purification transform (30) is therefore expected to converge as desired if the eigenvalues of the current non-idempotent density matrix are sufficiently close to 0 and 1 (see the intervals above). For an idempotent density matrix **P**, the eigenvalues must be either 0 or 1, of course.

Including the Lagrangian normalization constraint and the McWeeny purification transformation into eq. (29) yields the functional that needs to be minimized:

$$\Omega (\mathbf{P}) = \mathrm{Tr} [(3 \mathbf{PP} - 2 \mathbf{PPP}) \mathbf{F}] + \mu (\mathrm{Tr} (\mathbf{P}) - N_e/2) \tag{32}$$

where μ is a Lagrangian multiplier. To be consistent with previous formulations [46,49], we have made use of the identity Tr(**AB**) = Tr(**BA**) in the first term. The functional (32) differs slightly from the one originally proposed for tight-binding methods [18,41]. The present form is more suitable for semiempirical and ab initio MO methods, for reasons that are discussed elsewhere [46].

The minimization of the functional (32) needs to be carried out during each SCF iteration after forming the new Fock matrix. Any established minimization technique can be applied for this purpose. The simplest choice is the steepest descent (SD) method where the minimization follows the negative gradient **G** of the functional:

$$\mathbf{G} (\mathbf{P}) = - \nabla \Omega (\mathbf{P}) \tag{33}$$

A preferred alternative is the conjugate gradient (CG) method which implicitly employs properties of the Hessian to generate a sequence of orthogonal search directions $\mathbf{H}_k$ [70]. Line searches are performed along $\mathbf{H}_k$ to determine an optimum step length λ_k such that the updated density matrix

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \lambda_k \mathbf{H}_k \tag{34}$$

minimizes the given functional. When using exact line searches, each gradient in the CG sequence will be orthogonal to the others which leads to an efficient minimization [70]; however, numerical inaccuracies and deviations of the functional from quadratic behavior can impair conjugacy and the

rate of convergence [70]. The initial CG search direction H_0 is taken to be the steepest descent direction $\mathbf{G}_0(\mathbf{P}_0)$, and subsequent search directions $\mathbf{H}_k$ are obtained from

$$\mathbf{H}_{k+1} = \mathbf{G}_{k+1} (\mathbf{P}_{k+1}) + \gamma_k \mathbf{H}_k \tag{35}$$

The factor γ_k is determined from the current and the previous gradient using one of several alternative update formulas [70] (see section 4).

Both the SD and CG methods as well as more refined quadratically convergent schemes [48] require the gradient of the functional which can be derived using trace algebra [46,49]:

$$\nabla\Omega = 3 (\mathbf{PF} + \mathbf{FP}) - 2 (\mathbf{PPF} + \mathbf{PFP} + \mathbf{FPP}) + \mu \mathbf{I} \tag{36}$$

where $\mathbf{I}$ is the NxN unit matrix (N basis functions). The Lagrange multiplier μ is chosen such that the gradient, eq. (36), is traceless [43,46]:

$$\mu = - \mathrm{Tr} [3 (\mathbf{PF} + \mathbf{FP}) - 2 (\mathbf{PPF} + \mathbf{PFP} + \mathbf{FPP})] / N \tag{37}$$

This ensures that the search direction derived from the gradient (see above) is traceless. As a consequence, the line search according to eq. (34) will preserve the number of electrons. This line search can be carried out analytically: since the functional (32) has a cubic dependence on the density matrix, its value during the line search (34) can be expressed as a cubic polynomial of the step length λ_k:

$$\Omega (\lambda_k) = a + b \lambda_k + c \lambda^2_k + d \lambda^3_k \tag{38}$$

The coefficients can be derived by inserting eq. (34) into eq. (32):

$$a = - \mathbf{G}_k (\mathbf{P}_k) \tag{39}$$

$$b = - \mathrm{Tr} [\mathbf{H}_k \mathbf{G}_k (\mathbf{P}_k)] \tag{40}$$

$$c = 3 \mathrm{Tr} [\mathbf{H}_k \mathbf{H}_k \mathbf{F}] - 2 \mathrm{Tr} [\mathbf{H}_k \mathbf{H}_k \mathbf{P}_k \mathbf{F} + \mathbf{H}_k \mathbf{P} \mathbf{H}_k \mathbf{F} + \mathbf{P} \mathbf{H}_k \mathbf{H}_k \mathbf{F}] \tag{41}$$

$$d = - 2 \mathrm{Tr} [\mathbf{H}_k \mathbf{H}_k \mathbf{H}_k \mathbf{F}] \tag{42}$$

The minimum of eq. (38) with respect to λ_k is found by taking the corresponding derivative and setting it to zero:

$$b + 2\,c\,\lambda_k + 3\,d\,\lambda^2_k = 0 \qquad (43)$$

Solution of this quadratic equation yields two roots, and the root that corresponds to the minimum provides the optimum step size λ_k for the line search, eq. (34). The idempotency of the density matrix is not preserved exactly during the line search, and it is therefore necessary to restore idempotency of the updated density matrix, eq. (34), to the desired accuracy through a sequence of purification transforms, eq. (30).

In summary, a given iteration of the SCF procedure with CG-DMS will thus involve the following steps:

(a) use the currently available density matrix to construct a corresponding Fock matrix;
(b) compute the electronic energy and check for convergence;
(c) apply an iterative CG search to determine a new density matrix that minimizes the functional (32) by performing the following operations in each CG cycle:
- compute the current gradient, eq. (33);
- define the current search direction, eq. (35);
- update the density through a line search, eq. (34), which requires the determination of the optimum step length, eqs. (40)-(43);
- purify the resulting density matrix, eq. (30).

If no further approximations are introduced, the CG-DMS approach outlined above will yield the correct solution to the SCF-MO treatment (i.e., the same electronic energy and the same density matrix as obtained from a conventional solution of the Roothaan-Hall equations through matrix diagonalization). Essentially all the computational work in this approach consists of matrix operations (mostly matrix multiplications). In large systems, most elements of the relevant matrices (**P**, **F**, **h**) are very small and can be neglected (see section 3.1). By introducing suitable cutoffs, one obtains sparse matrices which give rise to linear scaling of the computational cost in the CG-DMS approach.

A number of modifications can be made to the basic CG-DMS algorithm, e.g., concerning preconditioning [46], DIIS convergence acceleration [19], and simplifications due to a special choice of the initial density matrix [49]. Some of these topics and other technical issues will be

discussed in more detail later (see section 4). It has been established that the CG-DMS results converge to the exact results from matrix diagonalization when the cutoffs tend to zero, and suitable values for these cutoffs have been recommended [19,46,47,49]. Linear scaling CG-DMS calculations have been reported in a number of solid-state studies, mostly at the tight-binding level (for reviews see [10,13,71]). The relatively few applications to molecules include geometry optimizations for giant fullerenes at the tight-binding level [44] and for a small protein at the semiempirical PM3 level [72].

3.4 Localized molecular orbital approach

The idea behind this linear scaling technique is best appreciated by first considering the pseudodiagonalization scheme [60] that is commonly used in standard semiempirical MO calculations as a replacement for full matrix diagonalization. In a given SCF iteration, this scheme builds the Fock matrix **F** in the AO basis from the currently available density matrix, eqs. (10)-(12), and then generates the Fock matrix $\mathbf{F}^{MO}$ in the MO basis by transforming with the currently available MO eigenvectors **C**. More precisely, only the occupied-virtual block of $\mathbf{F}^{MO}$ is constructed using the occupied and virtual MO eigenvectors ($\mathbf{C}_{occ}$ and $\mathbf{C}_{virt}$, respectively):

$$\mathbf{F}^{MO}\,(\text{occ-virt}) = \mathbf{C}^{T}_{occ}\,\mathbf{F}\,\mathbf{C}_{virt} \tag{44}$$

To obtain the solution of the Roothaan-Hall equations, it is sufficient to annihilate all Fock matrix elements F_{ia} in eq. (44) connecting the occupied MOs $\mathbf{C}_i$ and the virtual MOs $\mathbf{C}_a$. An approximate elimination of these matrix elements is achieved by a series of 2x2 unitary transformations that diagonalize the corresponding 2x2 secular problem:

$$\mathbf{C}_i\,(\text{new}) = \alpha\,\mathbf{C}_i + \beta\,\mathbf{C}_a \tag{45}$$

$$\mathbf{C}_a\,(\text{new}) = -\,\beta\,\mathbf{C}_i + \alpha\,\mathbf{C}_a \tag{46}$$

Explicit formulas for the coefficients α and β are given in the literature [17,60]. This procedure is essentially a series of Jacobi transformations which, however, are restricted to the occupied-virtual block and are not iterated. The resulting new eigenvectors are used to build a new density matrix, eq. (7), and the next SCF cycle is entered. It has been established that the combined use of SCF iterations and non-iterative 2x2 Jacobi transformations for the occupied-virtual block is normally sufficient to reach SCF convergence in semiempirical MO calculations and that the number of

required SCF iterations is similar as in calculations with full matrix diagonalization [60]. A prerequisite is, however, that this pseudodiagonalization scheme starts from realistic initial MOs that are usually obtained by matrix diagonalization during the first SCF iteration(s).

Standard pseudodiagonalization schemes employ delocalized canonical MOs. Since the ground-state determinant is invariant to unitary transformations between the occupied MOs, it can equally well be described by localized molecular orbitals (LMOs). For large molecules, the use of LMOs in the pseudodiagonalization scheme outlined above is particularly attractive, for the following reasons [17]:

- The annihilation of all occupied-virtual interactions becomes much easier because most of them will automatically be zero (i.e., all those separated by a large distance). The computations according to eqs. (44)-(46) can be restricted to those pairs of occupied and virtual LMOs that are close enough to each other (according to some cutoff criterion).

- For each occupied LMO, the calculation of the contributions to the density matrix can be limited to a small number of matrix elements.

These and related arguments [17] show that the computational effort for the pseudodiagonalization scheme and some other time-consuming steps in the SCF procedure will scale linearly with the size of the system when LMOs are used. An overall linear scaling can only be achieved, however, if this is true for all steps of the calculation, particularly also for the generation of the LMOs.

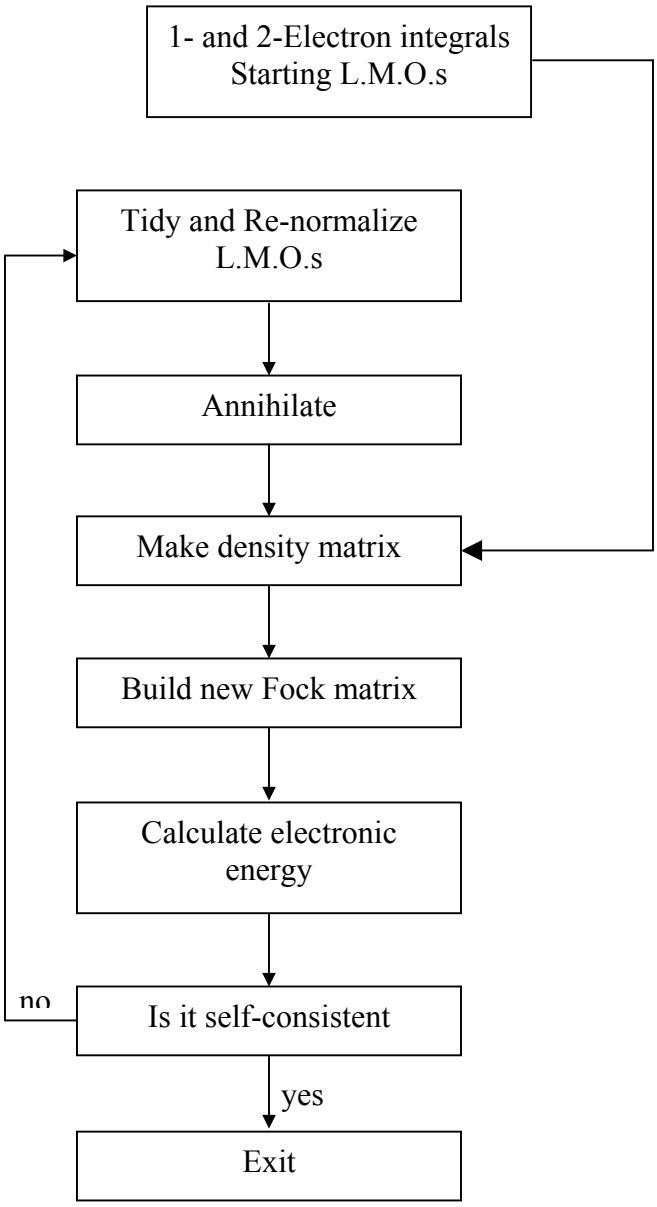
To avoid the initial full diagonalizations (see above) a special procedure has been developed that provides starting LMOs based on the classical Lewis structure of the system studied [17]. These starting LMOs are built from suitable hybrid orbitals at each atom that are combined with those from the neighboring atoms in a bonding or antibonding fashion to give (diatomic) occupied and virtual LMOs, respectively; hybrid orbitals not used in this process are considered to be nonbonding (monoatomic) LMOs. This algorithm [17] will produce a set of orthonormal starting LMOs for all systems that can be represented by Lewis structures (but may fail in more complicated electronic situations).

When these starting LMOs are subjected to the SCF iterations in the pseudodiagonalization LMO scheme described above, the LMOs spread over more than two atoms during the SCF cycles, but they tend to remain localized for large molecules in a region that is small compared with the whole molecule [17]. Linear scaling can be achieved by "tidying" the LMOs (neglecting small coefficients

at distant atoms) with subsequent renormalization [17]. The overall LMO approach has been found to converge well [17].

Each SCF iteration that employs the pseudodiagonalization LMO scheme requires following steps (see also the flowchart [17] in Scheme 2):

(a) use the currently available density matrix to construct a corresponding Fock matrix in the AO basis;
(b) compute the electronic energy and check for convergence;
(c) tidy and renormalize the currently available LMOs;
(d) use these LMOs to transform the Fock matrix to the MO basis, eq. (44);
(e) annihilate occupied-virtual interaction matrix elements and generate improved LMOs, eqs. (45)-(46);
(f) compute a new density matrix from these LMOs.



Scheme 2. SCF calculation using localized molecular orbitals.

The consistent use of LMOs leads to an essentially linear scaling of the computational effort as demonstrated by single-point calculations on selected proteins [17]. It has also been shown that the results from the LMO approach converge to those from full matrix diagonalization when the relevant cutoffs are tightened [17]. Applications of the LMO approach at the semiempirical level include a geometry optimization of a small protein (crambin) [73], comparisons of QM/MM and QM energy profiles in enzymatic reactions (hydride transfer in dihydrofolate reductase) [74,75], and an analysis of the electrostatic potential in the potassium ion channel [76].

3.5 Conclusion

At the outset of the project, the three linear scaling approaches for semiempirical MO calculations described above had already been proposed, and it had become clear that they could be used to treat large molecules. All of these methods attempt to achieve linear scaling by neglecting small matrix elements so that their results will generally show some deviations from the exact results which, however, can be controlled through the choice of the relevant cutoffs. All of them require some overhead so that conventional calculations with full diagonalization remain faster for small molecules; the crossover point beyond which the linear scaling approaches become faster depends on a number of factors (such as the chosen cutoffs). Looking at the underlying algorithms it is not obvious which of the three approaches is expected to perform best in general, and even though there have been a number of benchmark studies published more recently (see above) this question has not yet settled.

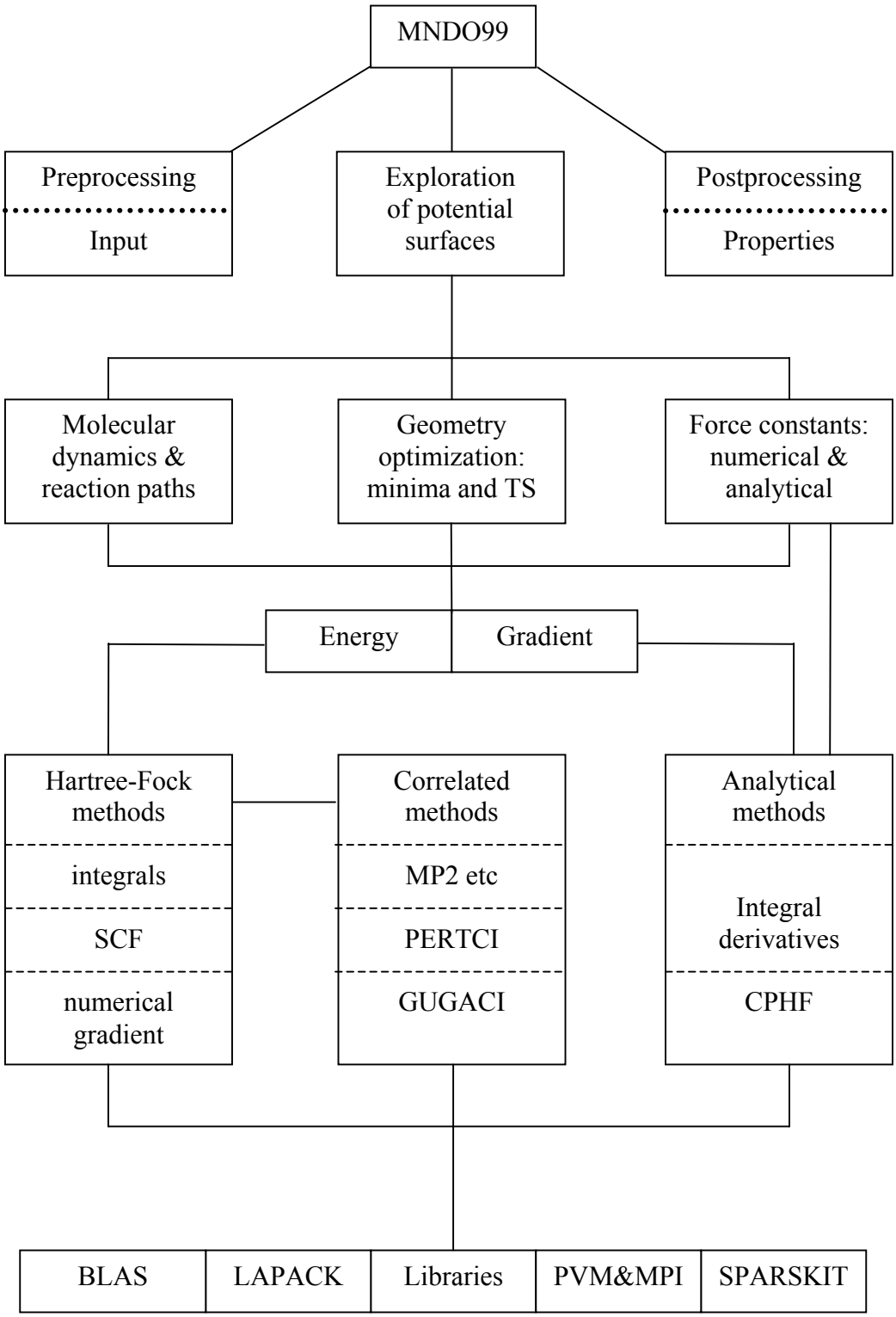
Given this situation, we decided at the outset of the project to implement the conjugate gradient density matrix search. The main reason was that it is the most straightforward approach from a conceptual point of view. It is based on the direct minimization of the electronic energy with regard to the density matrix (under suitable constraints) and achieves linear scaling simply by truncating matrix elements below a user-chosen threshold. By contrast, the divide-and-conquer approach has to introduce a partitioning of the system into subsystems (with associated buffer regions), and the LMO approach requires the validity of Lewis structures during the initial LMO generation. Such additional constructs are not needed in the conjugate gradient density matrix search which is therefore conceptually the simplest of the three approaches. Its implementation is discussed in the next section.

Chapter 4

Implementation

4.1 Overview

The program package MNDO99 [20] performs semiempirical quantum-chemical calculations of molecular properties. It provides all the standard semiempirical methods (e.g., MNDO [24], AM1 [25], PM3 [26,27], and MNDO/d [77,78]) as well as the recently developed methods with orthogonalization corrections (OM1 [28], OM2 [22], and OM3 [23]). Electron correlation can be treated explicitly by perturbation theory or various forms of configuration interaction (CI) up to full CI within a given active space (GUGACI formalism). Analytic gradients are available for most methods both at the SCF and the CI level, while an analytic Hessian can be computed only for MNDO-type methods at the SCF level. Potential energy surfaces can be explored by a variety of techniques including geometry optimization for minima or transition states and force constant analysis. Scheme 3 shows the overall program structure.



Scheme 3: The modular structure of the MNDO99 code.

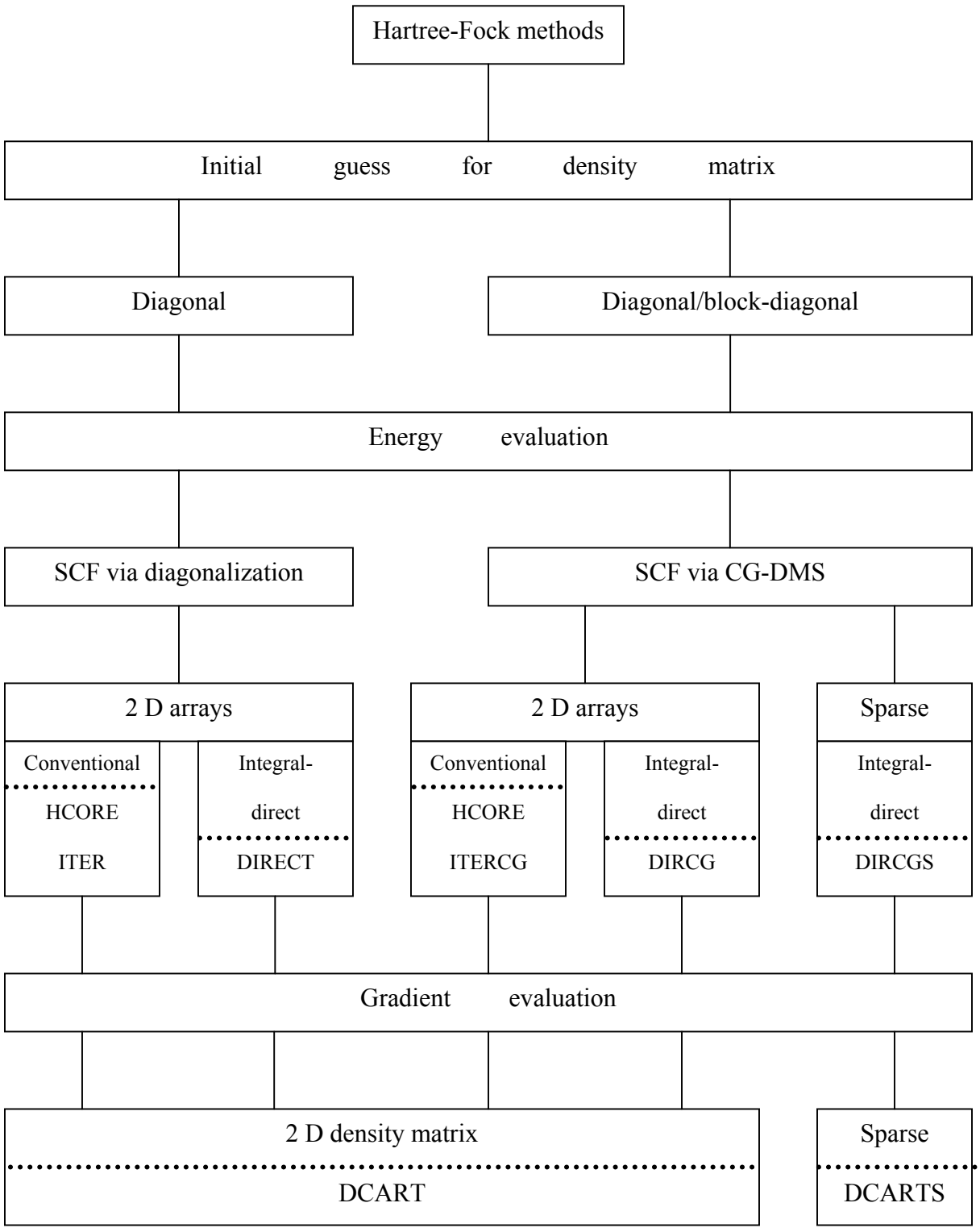
The task of the present work was to implement a linear scaling approach for the evaluation of the energy and gradient in semiempirical SCF-MO methods. For reasons discussed above (section 3) the conjugate gradient density matrix search (CG-DMS) was chosen for this purpose. The module

for SCF-MO calculations (labeled "Hartree-Fock methods" in Scheme 3) thus had to be extended by a CG-DMS implementation.

The previously available SCF-MO code in MNDO99 offered the choice between a conventional calculation with precomputation of the required integrals (HCORE) followed by the SCF iterations (ITER) and an integral-direct calculation with on-the-fly integral evaluation during the SCF iterations (DIRECT). In the present work, three CG-DMS variants were implemented. The first two employ a conventional strategy (HCORE followed by ITERCG) and an integral-direct strategy (DIRCG), respectively, in combination with full two-dimensional arrays for all relevant matrices, in order to allow for direct comparisons with the previously existing versions with matrix diagonalization. The third CG-DMS variant (DIRCGS) is again integral-direct, but makes use of sparse matrix technology in order to achieve linear scaling. The first two CG-DMS implementations may thus be regarded as test versions, since they are analogous to the previously available code, except for replacing matrix diagonalization by direct minimization of the density matrix. The third implementation is designed as linear scaling production version.

In any case, an initial guess of the density matrix is required at the start of the SCF iterations. The default in the existing code is a simple diagonal guess (GUESSP) which can also be generated in sparse matrix format (SPARSP). In the course of this work, it became evident that SCF convergence is less robust when using CG-DMS instead of matrix diagonalization (see below), and therefore an alternative block-diagonal guess was developed for CG-DMS that provides an improved initial density matrix either in square matrix or sparse matrix format (FRAGMT).

A successful SCF-MO treatment yields a converged density matrix that may be used to compute molecular properties. The first two CG-DMS implementations (HCORE/ITERCG and DIRCG) provide the results in the same format as usual so that the standard routines can be used for postprocessing without any changes. The third variant (DIRCGS) produces the density matrix as a sparse matrix which makes it necessary to adapt the routines for postprocessing correspondingly (e.g., for the dipole moment and other properties). The most important changes concern the gradient of the energy with respect to the nuclear coordinates which is required for the efficient exploration of potential surfaces. The simplest evaluation of the Cartesian gradient for closed-shell SCF-MO calculations in MNDO99 is numerical and involves a finite-difference computation of all relevant integral derivatives which are then contracted with the corresponding density matrix elements to accumulate the Cartesian gradient. The corresponding standard routine (DCART) was modified such that the gradient can be computed with a sparse input density matrix (DCARTS).



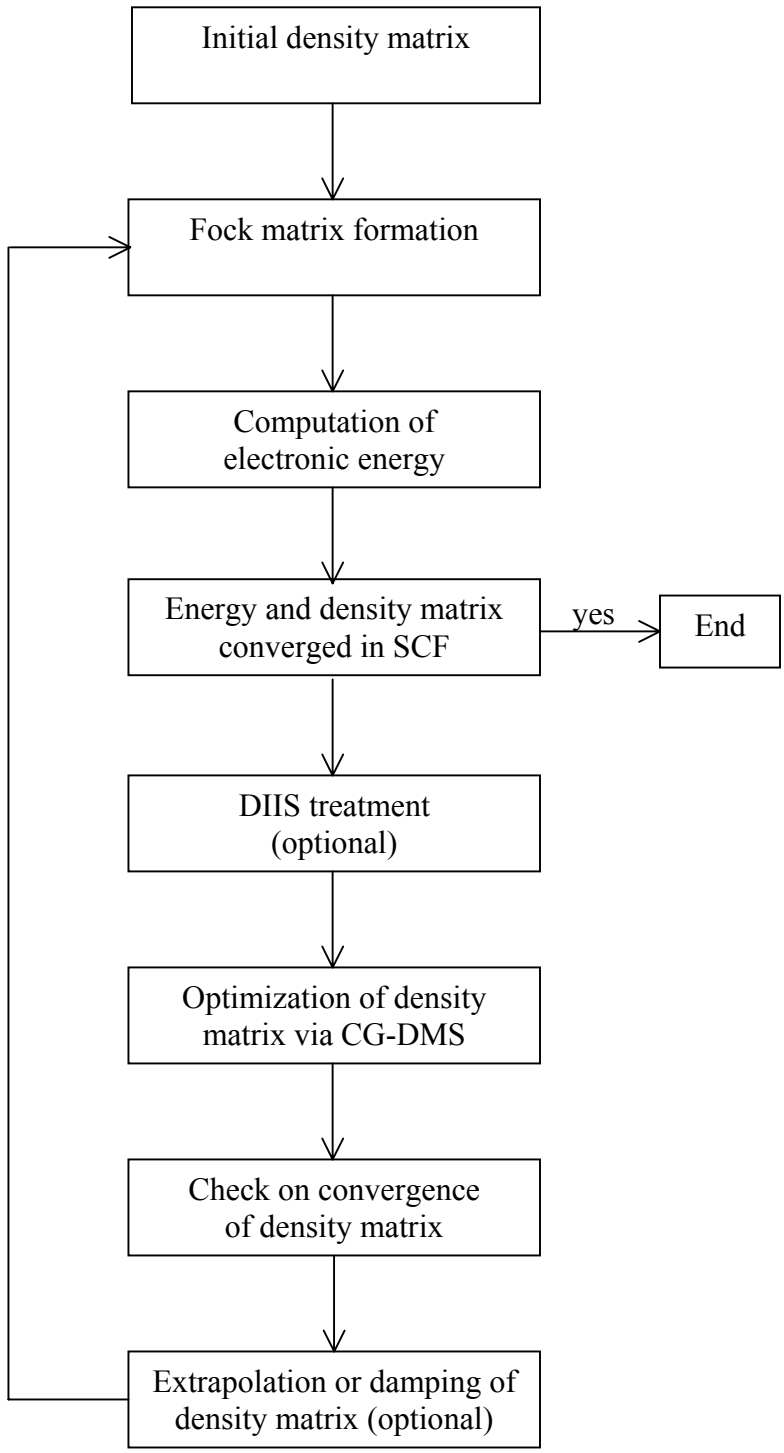
Scheme 4: Structure of the SCF-MO module (see text).

Scheme 4 shows the program structure of the SCF-MO module after including the new CG-DMS routines. In the following, we shall first discuss the CG-DMS implementation as such (common to ITERCG, DIRCG, and DIRCGS), before we address the sparse matrix version in more detail (DIRCGS).

4.2 Conjugate gradient density matrix search

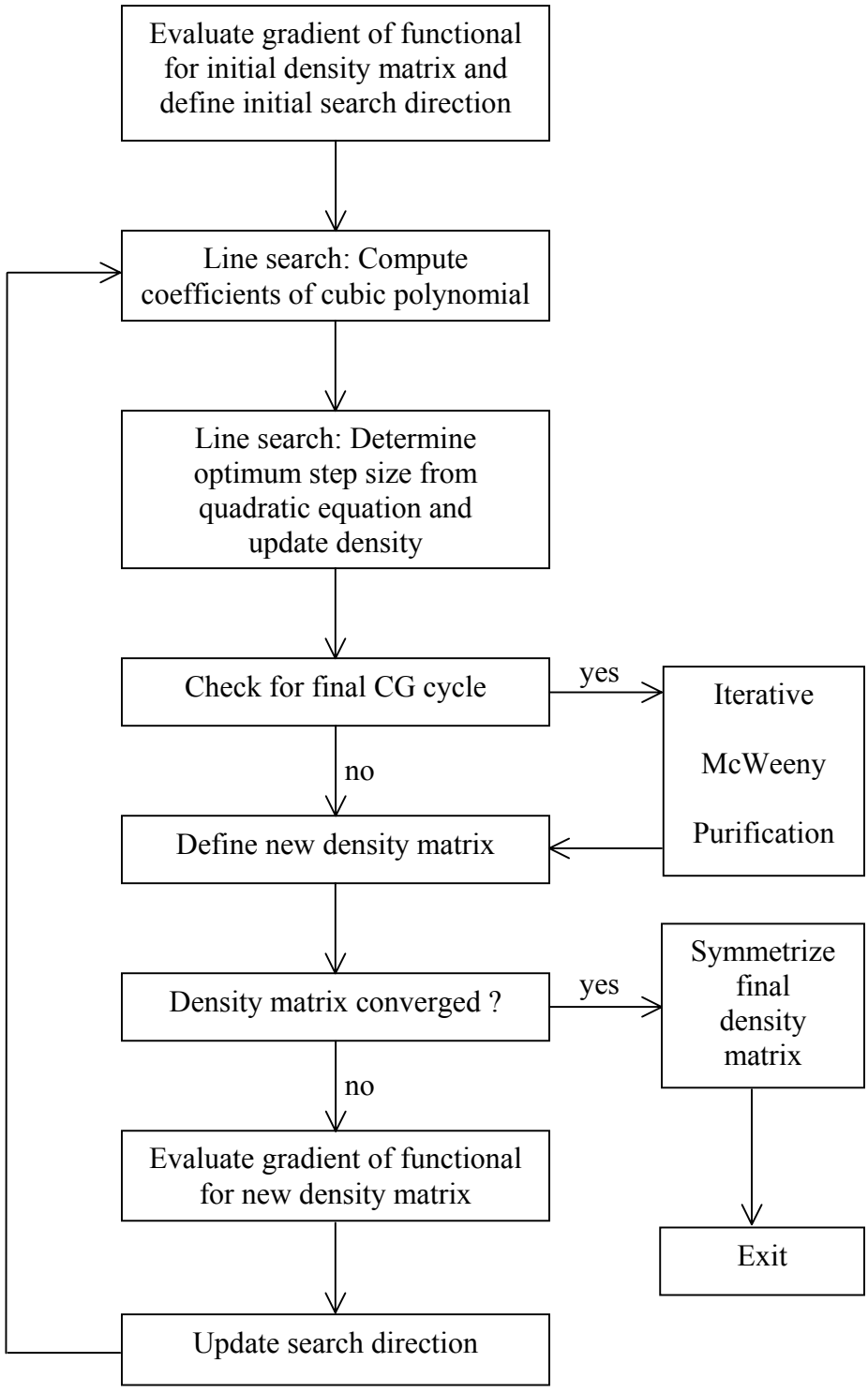
In section 3.3, the SCF procedure with direct minimization of the density matrix has been described in general terms, and the CG-DMS approach has been specified through eqs. (23)-(43). The corresponding implementation is presented in this section (see Schemes 5 and 6).

At the beginning of each SCF iteration, the currently available density matrix (either from the initial guess or the preceding iteration) is used to build a new Fock matrix. In the conventional approach (ITERCG), the Fock matrix is constructed from the density matrix and precomputed integrals using standard MNDO99 routines, followed by the calculation of the electronic energy. In the integral-direct approach (DIRCG), the calculation of the Fock matrix is done in a loop over atom pairs with on-the-fly integral evaluation and concomitant calculation of the corresponding contributions to the electronic energy. Thereafter, both implementations (ITERCG, DIRCG) follow the same course. SCF convergence is checked both with regard to the electronic energy and the density matrix (using data obtained from the latest density update, see below). If convergence has not yet been reached, there is an option to perform a DIIS extrapolation for convergence acceleration (normally not done). The Fock matrix is then brought into the form of a square symmetric matrix before entering the CG-DMS procedure (see below) which generates an updated density matrix. This new density matrix is then compared with the previous one to extract convergence data (maximum and rms deviations), and there is the option to modify the density matrix through extrapolation or damping. Thereafter, the next SCF iteration starts.



Scheme 5: SCF procedure with CG-DMS (see text).

This SCF procedure is analogous to the standard one, except that the diagonalization of the Fock matrix and the subsequent computation of the density matrix from the eigenvectors are replaced by the CG-DMS procedure. Both implementations (ITERCG, DIRCG) use the same routine (CGDMS) for this procedure (see Scheme 6).



Scheme 6: Conjugate gradient density matrix search (see text).

At the outset of CGDMS, the gradient of the functional Ω, eq. (32), is evaluated (DMSGRD) according to eqs. (36)-(37). Exploiting the fact that the Fock matrix **F** and the density matrix **P** are symmetric, the gradient formula (36) can be implemented as follows:

$$\mathbf{U} = \mathbf{F}\,\mathbf{P} \text{ (intermediate array)} \tag{47}$$

$$\mathbf{V} = \mathbf{F}\,\mathbf{P} + \mathbf{P}\,\mathbf{F} = \mathbf{U} + \mathbf{U}^T \quad \text{(intermediate array)} \tag{48}$$

$$\nabla\Omega = 3\,\mathbf{V} - 2\,\mathbf{P}\,\mathbf{V} - 2\,\mathbf{U}\,\mathbf{P} + \mu\,\mathbf{I} \tag{49}$$

Hence, only three (expensive) matrix multiplications are needed to evaluate the gradient from eqs. (36)-(37), along with some other (cheaper) matrix operations (e.g., addition and calculation of the trace). Under special circumstances, further simplifications are possible [49] which have been implemented, but are normally not used in practice. The negative gradient of the functional defines the initial steepest-descent search direction.

The conjugate gradient method is an iterative procedure to minimize the functional Ω, eq. (32), with respect to all density matrix elements. In each CG cycle, the density matrix $\mathbf{P}_k$ is updated by a line search along the search direction $\mathbf{H}_k$, see eq. (34). The procedure is terminated after N_{cg} cycles, either if the density matrix has converged to the required accuracy or if the maximum number of allowed CG cycles (maxcg) has been exceeded.

Each CG cycle starts with an analytic line search, eqs. (38)-(43), using the currently available matrices $\mathbf{P}_k$, $\mathbf{H}_k$, and $\mathbf{G}_k$ (negative gradient). The latter two matrices are obtained either from the initialization (see above) or from the preceding CG cycle (see below). The line search requires determination of the coefficients b, c, and d in the cubic polynomial representation of Ω (DMSCOF): exploiting the symmetry of the matrices involved and introducing some auxiliary matrices, this can be achieved essentially through four matrix multiplications and four evaluations of the trace of a product matrix. Dropping the cycle index k, we have:

$$\mathbf{W} = \mathbf{H}\,\mathbf{F} \tag{50}$$

$$\mathbf{X} = \mathbf{H}\,\mathbf{W} = \mathbf{H}\,\mathbf{H}\,\mathbf{F} \tag{51}$$

$$\mathbf{Y} = \mathbf{H}\,\mathbf{P} \tag{52}$$

$$\mathbf{Z} = (\mathbf{Y}+\mathbf{Y}^T)\,\mathbf{F} = (\mathbf{H}\,\mathbf{P} + \mathbf{P}\,\mathbf{H})\,\mathbf{F} \tag{53}$$

$$b = -\,\mathrm{Tr}\,(\mathbf{H}\,\mathbf{G}) \tag{54}$$

$$c = 3\ \mathrm{Tr}\,(\mathbf{W}) - 2\ \mathrm{Tr}\,(\mathbf{H}\,\mathbf{Z}) - 2\ \mathrm{Tr}\,(\mathbf{P}\,\mathbf{X}) \tag{55}$$

$$d = -\,2\ \mathrm{Tr}\,(\mathbf{H}\,\mathbf{X}) \tag{56}$$

Using the computed coefficients, the optimum step size λ in the line search can be found by solving a simple quadratic equation, see eq. (43). The corresponding routine (CGROOT) first checks whether the linear term dominates (coefficient d vanishingly small) such that the solution of the corresponding linear equation can be adopted; if this is not the case, it rejects any physically unacceptable root of the quadratic equation that would yield unphysical diagonal density matrix elements; if both roots are acceptable, it computes the corresponding values of the functional, eq. (32), and adopts the root with the lower value. The latter (general) case requires four matrix multiplications so that each line search will typically involve eight matrix multiplications. The root-finding routine (CGROOT) returns the updated density matrix, eq. (34), and also the corresponding purified version 3 **PP** - 2 **PPP**, eq. (30), which is automatically obtained when computing the value of the functional, eq. (32).

After the line search, the maximum and rms deviations between the updated and the previous density matrix are determined (MAXDEV) to see whether the iterative CG procedure has converged. If this is the case or if the maximum number of CG cycles has been reached, an iterative McWeeny purification of the current density matrix is carried out, eq. (30), in order to restore the idempotency of the density matrix that is not preserved exactly during the CG update; such purification can be enforced in every CG cycle via input, but this is normally not done. Each McWeeny purification requires two matrix multiplications (PURIFY). The iterative procedure starts from the purified matrix that is available after the CG update (CGROOT, see above) and terminates after N_{pur} cycles, either if the convergence criteria for the purification are satisfied (concerning the maximum allowed change in the density matrix, mcmax, and the maximum allowed violation of idempotency, midemp) or if the maximum number of allowed McWeeny cycles (maxpur) has been exceeded. The McWeeny purification thus involves a total of 2 N_{pur} matrix multiplications.

The currently available purified density matrix (from CGROOT or PURIFY) is adopted as the density matrix $\mathbf{P}_{k+1}$ resulting from the current CG cycle. The maximum and rms deviations from the previous density matrix $\mathbf{P}_k$ are determined (MAXDEV) to check for CG convergence, applying the same convergence criteria as for the overall SCF procedure (iplscf). If convergence has not yet been attained and if the maximum number of allowed CG cycles have not yet been completed, the next CG cycle is prepared: the gradient $\mathbf{G}_{k+1}$ is computed for the new density matrix $\mathbf{P}_{k+1}$ according

to eqs. (32)-(33) and (36)-(37), which requires three matrix multiplications (DMSGRD, see above). In addition, the new search direction vector $\mathbf{H}_{k+1}$ is determined from $\mathbf{G}_{k+1}$ and $\mathbf{H}_k$ through an update, eq. (35), where the factor γ is taken either from the Polak-Ribiere or the Fletcher-Reeves formula [70] which are given in terms of dot products of gradients:

$$\gamma = [\ \mathbf{G}_{k+1}\ \mathbf{G}_{k+1} - \mathbf{G}_k\ \mathbf{G}_{k+1}\ ] / \mathbf{G}_k\ \mathbf{G}_k \quad \text{(Polak-Ribiere)} \tag{54}$$

$$\gamma = \mathbf{G}_{k+1}\ \mathbf{G}_{k+1} / \mathbf{G}_k\ \mathbf{G}_k \quad \text{(Fletcher-Reeves)} \tag{55}$$

The two formulas are mathematically equivalent if the CG minimization is applied to exact quadratic forms [70]. Our functional, eq. (32), will certainly not show an exact quadratic dependence on the density matrix elements, and therefore the performance of these two updates may be different in practice. Systematic tests have shown, however, that there is no significant difference in our case. We have adopted the Polak-Ribiere update as default based on the general recommendations in the literature [70]. At this point, the k-th CG cycle is completed, and all information is available ($\mathbf{P}_{k+1}$, $\mathbf{G}_{k+1}$, $\mathbf{H}_{k+1}$) to enter the next CG cycle.

If CG convergence has been reached or if the maximum number of allowed CG cycles have been carried out, the program returns to the calling routine. Before doing so, the density matrix is symmetrized in order to remove numerical "noise" that may have appeared due to the finite numerical precision of the computation. Experience has shown that numerical noise may otherwise accumulate during the SCF/CG-DMS iterations and cause problems. There is also an option to scale the final density matrix in order to impose the correct trace in case of numerical inaccuracies, but this is normally not needed.

Looking at the overall procedure, the SCF/CG-DMS approach involves three nested loops (see Schemes 5 and 6): the outer SCF iterations, the CG cycles within each SCF iteration, and the McWeeny purification cycles at the end of each CG cycle. It would seem inefficient to converge the CG and McWeeny cycles tightly when the SCF iterations are still far away from convergence. On the other hand, if the overall SCF procedure is close to convergence, only very few CG and McWeeny cycles will be needed since the density matrix will then change only very slightly and both approaches converge very fast in this case (quadratic convergence for the McWeeny iterations [50,69]). Under these circumstances, it makes sense to allow only a small maximum number of CG cycles (maxng) and McWeeny cycles (maxpur) during the whole calculation: convergence may then be incomplete during initial SCF iterations, but will be reached later on. Recommended values of these input options (maxng, maxpur) can be derived from systematic tests (see below).

As already mentioned, the computational effort in the CG-DMS approach is dominated by matrix multiplications. In the current implementation, the total number of such matrix multiplications during each SCF iteration is typically N_{cg} (11 + 2 N_{pur}); for $N_{cg} = N_{pur} = 2$ (see below) this amounts to typically 30 matrix multiplications per SCF iteration. This has to be contrasted with the standard SCF approach using matrix diagonalization: the usual semiempirical implementations employ the pseudodiagonalization scheme (see section 3.4) which requires two matrix multiplications for the transformation of the Fock matrix, eq. (44), roughly the equivalent of one matrix multiplication for the Jacobi-type transformations, eqs. (45)-(46), and another matrix multiplication for the subsequent formation of the density matrix, eq. (7). Hence, the computational effort of the pseudodiagonalization scheme amounts to approximately only four matrix multiplications. It is therefore about an order of magnitude smaller than in the CG-DMS approach, especially when considering that the matrix operations in CG-DMS always involve full NxN matrices (N basis functions) whereas those in the pseudodiagonalization mostly work with smaller matrices of dimension NxN_{occ} or NxN_{virt} (N_{occ} occupied MOs, N_{virt} virtual MOs). To be fair, one also has to take into account that the standard SCF approach needs some full matrix diagonalizations (e.g., at the outset, see section 3.4) which are more expensive. In an overall assessment, however, the fact remains that the standard SCF approach is much less demanding computationally than the SCF/CG-DMS approach: with regard to the most time-consuming $O(N^3)$ matrix operations, there is about one order of magnitude difference in the computational effort.

It is therefore not surprising that the direct minimization of the density matrix is normally not employed in standard semiempirical SCF-MO codes, even though the basic theory behind this approach has long been known [50]. The recent interest in the SCF/CG-DMS approach arises from the fact, of course, that it enables us to exploit matrix sparsity rather easily (see section 3). It should also be clear from the preceding discussion, however, that the relevant matrices must become very sparse before SCF/CG-DMS becomes more efficient than the standard SCF approach: the latter requires intrinsically much less computation and can make use of highly optimized library routines (BLAS, LAPACK) which will always be more efficient than sparse matrix routines. In spite of these caveats, it is also clear that an O(N) SCF/CG-DMS code must win over a standard $O(N^3)$ SCF code at some point when the molecules become large enough.

4.3 Sparse matrix version

The algorithms discussed in the preceding section form the basis of all three CG-DMS implementations that we have developed. This section will address the issues that are specific to the sparse matrix version (DIRCGS).

The sparsity of a matrix is defined as the percentage of zero elements. For large molecules, density matrix elements connecting distant atoms will become vanishingly small (see section 3), and it is therefore a reasonable approximation to neglect them if their absolute value is smaller than a user-defined cutoff (mcutp). In the limit of very large molecules, this leads to sparse density matrices for which the number of non-zero elements increases linearly with system size. The Fock matrix shows a similar behavior and can therefore also be truncated by applying another user-defined cutoff (mcutf). Moreover, the distribution of nonvanishing elements is similar in the density matrix and the Fock matrix.

To take advantage of such truncations it is obvious that one should only store and process the nonzero elements of these sparse matrices. This makes it necessary to define the corresponding data structures and to choose standard procedures for sparse matrix operations. While there are general books on sparse matrix technology [79,80,81] and specific studies on sparse matrix multiplication [82,83,84,85], this field is much less developed than that of standard matrix operations, and there is no generally accepted sparse matrix software that would be comparable to the highly optimized BLAS and LAPACK libraries for standard matrix operations.

A variety of data structures have been suggested for sparse matrices [79,80,81,86] which include the compressed sparse row (CSR) format, the compressed sparse column (CSC) format, the coordinate format, and the linked list storage format. We have adopted the CSR scheme which seems to be the most common format for sparse matrices that do not have a special regular structure. In this scheme three one-dimensional arrays are used to define a sparse matrix: all nonzero matrix elements are stored row by row in a real (or double precision) array A, their column indices are collected in an integer array JA, and the pointers to the beginning of each row in A and JA are given in another integer array IA. The dimension of A and JA is equal to the total number of nonzero matrix elements (NNZ), and the length of IA is equal to N+1 if there are N rows. IA(i) defines the position in A and JA where the i-th row starts (for i=1,...,N), and IA(N+1) contains IA(1)+NNZ such that IA(N+1)-1 is the address of the last entry in A and JA. It should be stressed that the data in A and JA must match perfectly. The order of the elements within a given row does

not matter: the i-th row is stored between A(i) and A(i+1)-1, and the corresponding column indices are found between JA(i) and JA(i+1)-1.

An example of CSR storage for a square matrix is shown in Figure 2.

$$\mathrm{A} = \begin{matrix} 1 & 0 & 2 & 0 \\ 4 & 0 & 0 & 3 \\ 0 & 3 & 6 & 0 \\ 0 & 0 & 5 & 0 \end{matrix} \qquad \begin{matrix} A & 1,2,4,3,3,6,5 \\ JA & 1,3,1,4,2,3,3 \\ IA & 1,3,5,7,8 \end{matrix}$$

Figure 2: Square matrix A (left) in CSR format (right)

As already mentioned, the standardization and optimization of sparse matrix software is not yet very advanced. One of the first and probably most successful non-commercial attempts to build a library of sparse matrix routines is SPARSKIT [86]. Although still not complete with regard to the mathematical operations available, the freely available version SPARSKIT2 [86] provides sparse matrix versions of many BLAS routines. It is written in Fortran77 and uses the CSR format for the internal computations. We have decided to adopt SPARSKIT2 as our sparse matrix library. More specifically, we use three modules from SPARSKIT2 [86]: FORMATS converts matrices from one format to another. UNARY performs basic non-algebraic operations on sparse matrices (such as extracting certain elements or blocks, filtering out elements according to their magnitude, sorting elements within rows, forming the transpose, copying matrices, doing certain permutations, computing the number of nonzero matrix elements in the sum or product of two sparse matrices, etc). Finally, BLASSM carries out basic linear algebra operations on sparse matrices (such as matrix addition **A**+**B** and matrix multiplication **A***B**)**.

The CG-DMS algorithm requires a large number of different matrices (see section 4.2). Although their size scales linearly with the size of the molecule when cutoffs are applied, the memory requirements can still become quite substantial. We have therefore decided to program the sparse matrix CG-DMS implementation in Fortran90 to take advantage of the possibility to allocate and deallocate memory for temporary arrays and also of the additional data structures offered in Fortran90 (such as pointers). To accommodate these Fortran90 data structures, some of the routines in SPARSKIT2 were modified correspondingly and added to the library. This posed no problems since source code is available for SPARSKIT2.

After carrying out a sparse matrix operation such as an addition or multiplication, the resulting matrix will generally tend to have more nonzero elements than the two original matrices. After several such operations, matrix sparsity may be lowered significantly, and eventually the linear scaling behavior may be lost if no countermeasures are taken. There are two possible counterstrategies [19] that have been termed "let it grow" (LIG) and "fixed format" (FF). The LIG approach allows matrix elements to appear in the result matrix "wherever they like": initially the matrix operation is carried out only symbolically to determine the number of resulting nonzero matrix elements so that memory can be allocated; thereafter, the actual operation is performed, and finally the result matrix is subjected to filtration, i.e., the elements that are smaller than a suitable user-defined cutoff are neglected. This produces a matrix whose sparsity is compatible with that of the original matrices. The FF approach, on the other hand, analyzes the structure of the original two matrices beforehand; in our case of CG-DMS, for example, the density matrix **P** and the Fock matrix **F** have a similar structure (zero blocks for distant atoms) so that the structure of the product matrix **FP** can be fixed. Extending this idea one may define a fixed format of all relevant matrices for a certain portion of a CG-DMS calculation, e.g., within one CG cycle, before allowing the structures to change again when entering the next SCF iteration [19]. It is obvious that the FF scheme will be faster than the LIG scheme in general because the use of fixed format obviates the need for the initial symbolic operations. On the other hand, the FF scheme is more difficult to implement and introduces extra (small) numerical errors that need to be controlled in an appropriate manner. Since the gain in speed is less than a factor of 2 [19], we have decided against the FF approach and in favor of the LIG scheme, with filtration at the level of individual matrix elements. An alternative is to apply the filtration at the level of atom-atom blocks where a whole block is disregarded if its Frobenius norm is less than the chosen cutoff [49,82].

Based on these general considerations, a sparse matrix version of the CG-DMS code has been implemented. It is essentially a translation of the integral-direct Fortran77 version with full two-dimensional arrays (DIRCG) into a corresponding Fortran90 code (DIRCGS) with sparse matrices in CSR format using SPARSKIT2 library routines when possible. For the sake of documentation, the routines that implement the CG-DMS approach with full two-dimensional arrays and with sparse matrices are listed in Appendix 1 and Appendix 2, respectively, along with a brief description of their functionality.

4.4 Initial density matrix

The quality of the initial guess for the density matrix will affect the number of SCF iterations that are needed until convergence. In the extreme case of a very poor guess, the SCF procedure may even fail to converge. Fortunately, standard diagonalization methods for solving the eigenvalue problem in semiempirical SCF-MO approaches are quite robust so that a simple diagonal guess is normally sufficient. The following variants have been considered in this work.

D1: In the case of neutral molecules, each single atom is assumed to be neutral. For atoms with an s or sp basis set, the valence electrons are distributed evenly over the atomic orbitals (i.e., over the corresponding diagonal density matrix elements). For atoms with an spd basis in MNDO/d, d-orbitals (p-orbitals) are initially not populated for main-group elements (transition metals), and the valence electrons are evenly distributed over the remaining atomic orbitals. In the case of charged systems, the molecular charge is evenly distributed over all orbitals. These conventions specify the default initial guess in MNDO99 [20]. This diagonal density matrix has the correct number of electrons, but is generally not idempotent.

D2: Following a recent suggestion [49] for a simplified density matrix minimization, we have also implemented an initial diagonal guess where all electrons are evenly distributed over all atomic orbitals. Hence, the initial density matrix is the unit matrix scaled by an overall factor (N_e / N).

D3: In the case of neutral molecules, each single atom is assumed to be neutral. For each atom, the valence electrons are distributed evenly over the atomic orbitals. In the case of charged systems, the molecular charge is evenly distributed over all orbitals. This is obviously a modification of D1 and differs from D1 only in the treatment of atoms with an spd basis.

In the course of our work, it became apparent that the SCF/CG-DMS approach converges less reliably than the standard diagonalization approach (see section 5). It would thus be desirable to use an initial guess for the density matrix that is not merely diagonal, but includes some nondiagonal elements that reflect the bonding situation around a given atom. This basic idea is in the spirit of the LMO approach [17] which starts the SCF procedure from initial LMOs that are derived from the classical Lewis structures and represent diatomic bonding and antibonding interactions (see section 3.4). Moreover, the success of the divide-and-conquer method indicates that it is possible to assemble the density of the whole system from subsystem density matrices (see section 3.2).

We introduce a related concept and build an initial block-diagonal density matrix for the whole system from non-converged fragment densities that are obtained as follows in the case of closed-shell molecules: The system is partitioned into user-defined fragments, and each fragment is assigned an even number of electrons such that each fragment is neutral or at least approximately neutral (details see below) subject to the condition that the total number of all electrons is preserved. For each (small) fragment, a standard closed-shell SCF-MO calculation with matrix diagonalization is started and terminated after a few (linfrg) SCF iterations (typically linfrg = 1 or 2). The resulting non-converged density matrix is copied into the corresponding block of the initial density matrix for the whole system.

Some remarks are appropriate. First, the partitioning might yield rather unphysical fragments that might be hard to converge; this is no problem since the idea is not to reach SCF convergence in the fragment, but just do one or two diagonalizations to generate a local fragment density with a sensible topology. Second, the computational effort for this procedure scales linearly in the same sense as the divide-and-conquer method, since an increase in the system size will lead to a linear increase in the number of fragments. Third, the effort is small on an absolute scale since only small matrices need to be diagonalized in the fragments. Fourth, by construction, the resulting block-diagonal guess for the density matrix is sparse, normalized to the correct number of electrons, and idempotent. The latter property follows from the fact that each of the disjoint blocks is derived from matrix diagonalization of a fragment, see eqs. (9) and (27).

Technically, the generation of the block-diagonal initial guess has been implemented in a separate module (FRAGMT). Some input is needed to define the fragments (see below). Thereafter, the program loops over all fragments and performs one or few iterations of an RHF-SCF calculation (SCFFRG) which makes use of many existing standard routines from MNDO99. The fragment densities are then copied into the molecular density array which can be generated as a two-dimensional array (ITERCG, DIRCG) or as a sparse matrix in CSR format (DIRCGS). For the sake of documentation, Appendix 3 lists and describes the corresponding new routines.

In the remainder of this section, we address some issues related to the fragmentation. We cover clusters of isolated molecules which can be partitioned in a natural manner into fragments, and polymers where the partitioning is more difficult since covalent bonds must be cut.

In the first case, the block-diagonal guess outlined above works easily. If we have, for instance, a cluster of water molecules we can perform one or several SCF iterations for each molecule at the given coordinates. Combining the resulting densities of all water molecules trivially yields the

initial density matrix for the cluster, with each block corresponding to a particular isolated water molecule. The only open issue is then the number of SCF iterations to be performed on each fragment. It seems reasonable to use the same number (linfrg) on all fragments. We found in our tests that the SCF convergence for water clusters of different size is about equally fast for linfrg=1 and linfrg=2, and that nothing significant could be gained by converging the separate SCF calculations for the isolated water molecules (which is easy here, but might be problematic in other cases, see above). As a result of these tests, we have adopted the option linfrg=1 for all SCF/CG-DMS calculations on water clusters (see section 6).

The situation becomes more complicated when dealing with large covalent systems which need to be divided into subsystems. In general, the choice of suitable subsystems is not trivial and must be left to the user who has to define the subsystems via input (see below). In the case of polymers, the obvious choice is to partition the system into its natural subunits - the monomers. Homolytic cuts through covalent bonds in a long-chain polymer will formally give rise to two radicals at the two ends, and many biradicals in between. The latter might create problems if we insist on converging the fragment SCF calculations, but since we perform only a few SCF iterations we can treat these biradicals (with an even number of electrons) as closed-shell species when generating an initial density matrix guess. We find that this procedure works well in practice (better and simpler than an alternative open-shell treatment). The two terminal fragments are radicals in the case of homolytic cuts, but for the purpose of the initial guess, they can formally also be treated as closed-shell cation/anion or a anion/cation pairs (by shifting one electron from one terminus to the other). In the case of peptides, we have studied these three possibilities in detail, and we find it best to employ charged closed-shell terminal fragments, i.e., a cation at the N-terminus and an anion at the C-terminus. The block-diagonal initial density matrix generated in this manner leads to reliable and fast SCF/CG-DMS convergence for peptides, whereas the two alternative choices normally converge a bit more slowly and may occasionally even fail to converge. Therefore, we have adopted the closed-shell residue-based partitioning with a cationic N-terminal group and an anionic C-terminal group as our default fragmentation for peptides and proteins (see also Figure 3).

subsystems:

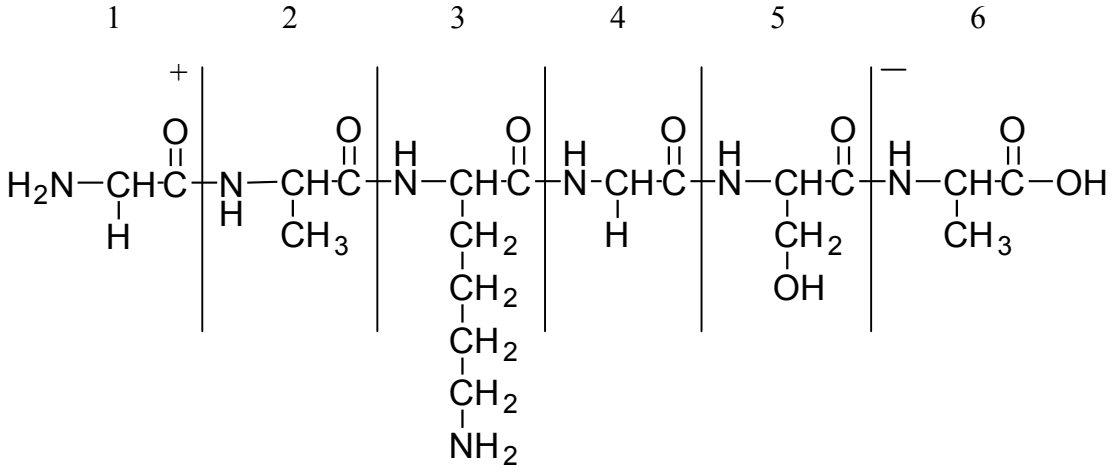


Figure 3: Subdivision of a peptide chain into residue-based fragments.

Some additional conventions have been established for calculations on peptides and proteins. First, when there is more than one chain of amino acids, it is obvious that the subdivision needs to be done for each chain separately. Second, the correct protonation state of each residue must be chosen, both for the overall calculation and for the generation of the initial guess: at pH 7, arginine (ARG) and lysine (LYS) normally occur in the protonated form, whereas glutamic acid (GLU) and aspartic acid (ASP) are usually deprotonated (see Figure 4). Third, some proteins contain disulfide bridges connecting two cystein residues [H2N-CH(CH2SH)-COOH]; in this case, it is advantageous for the initial guess to put the two cystein molecules with the disulfide bridge into the same fragment (see Figure 5). Fourth, if there is a metal ion (e.g., the zinc dication coordinated to the sulfur atom of cystein or the nitrogen atom of histidine), it is best to define a fragment that includes both the metal ion and the residue to which it is coordinated. Finally, any solvent molecules in the protein (e.g., water) are treated as separate subsystems, of course.

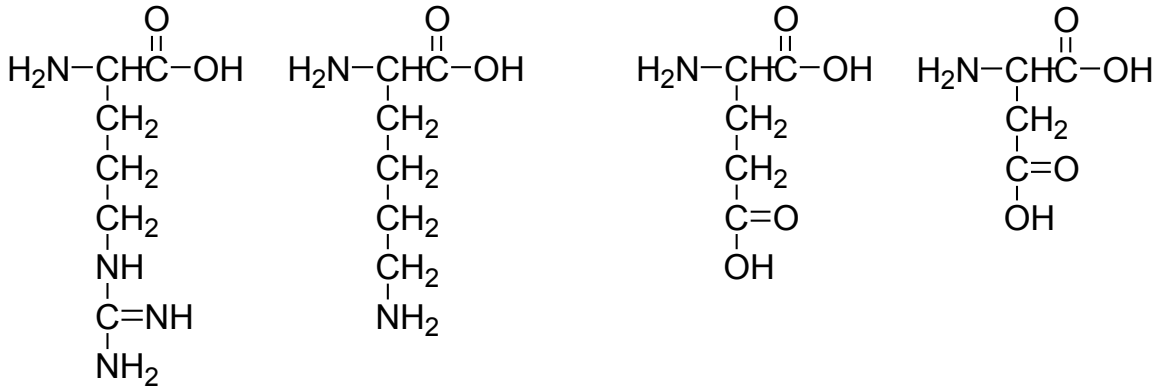


Figure 4: Formulas of selected amino acids (see text).

The conventions outlined above have been adopted for the SCF/CG-DMS calculations on peptides and proteins (see section 6). Starting from the resulting block-diagonal initial density matrices, SCF convergence has regularly been achieved without the need for additional measures (such as DIIS, extrapolation, or damping). We have not studied other large covalent systems as extensively as proteins, but we expect that an analogous block-diagonal guess will also work elsewhere - at least there is no evidence to the contrary.

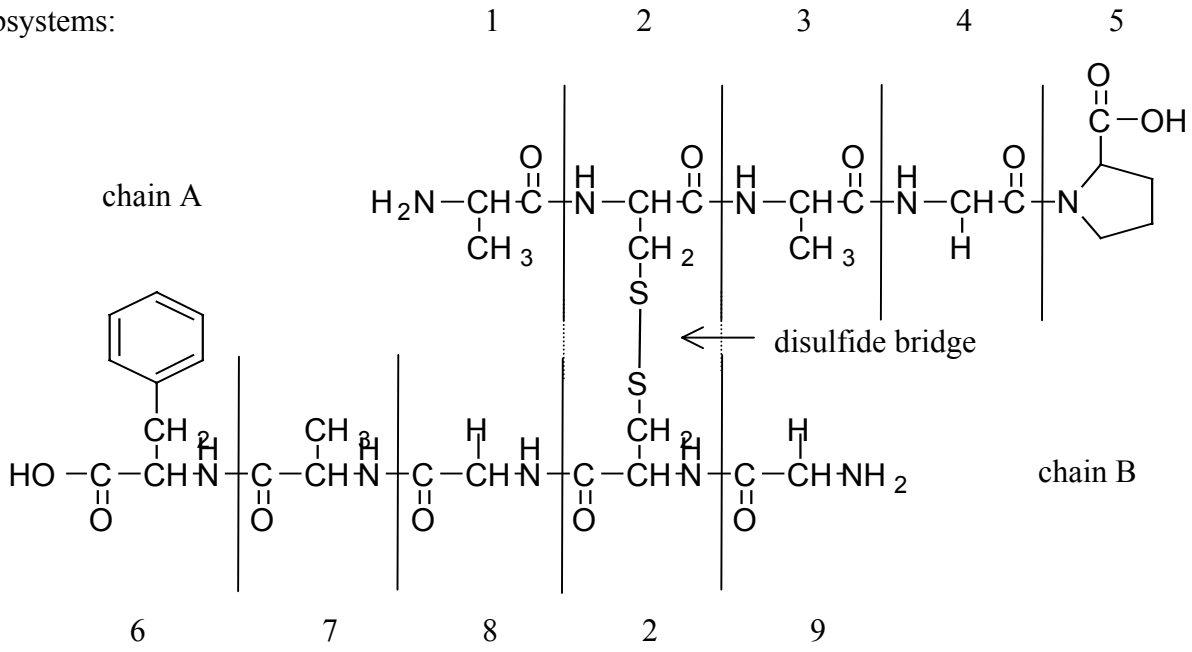


Figure 5: Recommended subdivision in bridged systems (see text).

In order to generate the block-diagonal guess, the user has to assign each atom in the molecule to a particular fragment. In general, this input can be provided manually, but in the case of biomacromolecules such as proteins, the assignment of atoms to residues (fragments) is included in structural databases (e.g., PDB) and can be retrieved by automatic tools that provide the required input information in a suitable format. In addition, the user has to specify the number of electrons assigned to each fragment. Since most fragments are neutral, the actual input contains only any nonzero charges of the fragments. Finally, the user has to choose the number (linfrg) of SCF iterations that will be performed for each fragment.

For the sake of documentation, Appendix 4 summarizes all input options that are relevant for SCF/CG-DMS calculations.

Chapter 5

SCF convergence and CG-DMS parameters

5.1 Overview

This section describes the tests of our CG-DMS implementation with regard to its correctness and its performance for small molecules. Two test sets were employed.

a) The CHNO set consists of 218 closed-shell molecules containing the elements hydrogen, carbon, nitrogen, and oxygen. It is derived from the original MNDO validation set [24] and is essentially identical to the CHNO-File used in the recent OM3 validation [23]. The size of the test molecules ranges from hydrogen H_2 to adamantane $C_{10}H_{16}$ (see [23] for a detailed list). MNDO single-point calculations were carried out using a standard sp minimal basis.

b) The MNDO/d set is comprised of 366 closed-shell molecules containing at least one heavier atom (P, S, Cl, Br, or I). It is a union of the sets used during the evaluation of the MNDO/d method for these elements [78,87]. The size of the test molecules ranges from hydrogen chloride HCl to triphenylphosphine oxide $(C_6H_5)_3PO$ (see [78,87] for a detailed list). The tests involved MNDO/d single-point calculations with d orbitals included at the heavier atoms.

The correctness of the SCF/CG-DMS results was judged by comparison against standard SCF results obtained by matrix diagonalization or pseudodiagonalization (see section 2). All calculations were done with the same general SCF convergence criteria: 10^{-6} eV for the electronic energy (iscf=6) and 10^{-6} for the diagonal elements of the density matrix (iplscf=6). Two separate calculations were considered to agree with each other if the deviations were smaller than these convergence criteria.

SCF/CG-DMS computations are controlled by a large number of options (see Appendix 4), and optimum choices for these options need to be established. In the following we shall discuss how the results depend on the choice of the initial density matrix (linfrg, ktrial), on the maximum number of conjugate gradient cycles per SCF iteration (maxcg), on the maximum number of McWeeny purification transformations per conjugate gradient cycle (maxpur), and on the conditions when these transformations are applied during a conjugate gradient minimization (mpurif). We have also studied the influence of general SCF options such as extrapolation of the density matrix between SCF iterations, but have not considered damping, level shifting, or DIIS convergence acceleration (see section 2.3).

5.2 Initial density matrix from diagonal guess

Starting from the default diagonal guess for the density matrix (D1, linfrg=0, ktrial=0; see section 4.4) and using standard CG-DMS options (maxcg=2, maxpur=2, mpurif=0), the SCF/CG-DMS calculations converge for all 218 molecules from the CHNO set. However, the computed heats of formation are correct only in 207 cases, and incorrect in the remaining 11 cases where they are much higher than the MNDO reference values obtained from matrix diagonalization. Closer inspection shows that the SCF/CG-DMS procedure converges to the dianion in these problem cases (ozone, several peroxides, p-quinone, dinitrogen tetroxide, and nitric acid).

Taking ozone as an example, we describe the course of such faulty SCF/CG-DMS calculations in more detail. The default initial diagonal density matrix (D1) for ozone is the unit matrix multiplied by 0.75 (closed-shell occupation factor of 2 not included; the trace of 9 thus corresponds to half the number of valence electrons). This initial density matrix has the correct trace, by construction, but it is far from idempotent: all its eigenvalues are 0.75 rather than 0 or 1 as in a proper idempotent density matrix. The trace is retained during the first CG update, by definition, but not during the subsequent first McWeeny purification: allowing for a single McWeeny transformation after the first CG update (default) yields a trace of 9.843 and eigenvalues of the density matrix in the range of 0.651-1.000. During the first three SCF iterations with the chosen CG-DMS options (maxcg=2, maxpur=2, mpurif=0; see Appendix 4), the trace settles to values around 10 while the minimum and maximum eigenvalues of the density matrix approach 0 and 1, respectively. From the fourth SCF iteration onwards, there is smooth convergence, with the trace stable at 10 and the eigenvalues stable at 0 and 1, and after 22 SCF iterations the chosen convergence criteria are satisfied. The SCF solution with a trace of 10 represents the ozone dianion with 20 valence electrons.

The transition from ozone to the ozone dianion is thus caused by the very first McWeeny purification: a single such transformation increases the trace from 9 to around 10, and the iterative SCF/CG-DMS procedure does not recover from this initial change. It should be stressed that under our default criteria (maxcg=2, maxpur=2, mpurif=0) the first purification is incomplete: when completing this first McWeeny purification in 8 steps one obtains an idempotent density matrix with a trace of 12 and eigenvalues of 0 and 1 (yielding an ozone hexaanion that is converged after 3 SCF iterations). Hence, one may arrive at SCF/CG-DMS solutions with different numbers of electrons depending on how the McWeeny purification is invoked (remember that the CG update preserves the number of electrons).

The problems encountered arise from the fact that our default diagonal guess for the initial density matrix is far from idempotent. The McWeeny purification transformation is designed for "almost idempotent" matrices which are then rendered "more idempotent". It should therefore be applied only with caution in the initial stages of the SCF/CG-DMS minimization with a diagonal initial estimate of the density matrix, but it cannot be completely avoided since idempotency needs to be imposed in order to get a physically valid density matrix.

We have carried out systematic test calculations to find out whether there is a reliable procedure for converging SCF/CG-DMS calculations starting from a diagonal initial guess. The following options were investigated both for the CHNO set and the MNDO/d set:

- different initial guesses D1, D2, and D3 (see section 4.4; note that D1 and D3 are identical for the CHNO set with an sp basis);
- maximum number of allowed CG cycles per SCF iteration (maxcg=1-5);
- maximum number of allowed McWeeny transformations per CG cycle (maxpur=1-3);
- performing maxpur McWeeny transformation in each CG cycle (mpurif=1), or doing so only in the last CG cycle while performing a single McWeeny transformation otherwise (mpurif=0), or allowing McWeeny transformations only when the CG search is essentially converged (mpurif=-1);
- calculations without or with density matrix extrapolation between SCF iterations ("std" or "ext", respectively).

Tables 1-4 list the results for the CHNO set with 218 molecules. They specify the number of molecules that were computed successfully for a given combination of options. It is immediately obvious that it has not been possible to converge all 218 molecules simultaneously in any such case. There are some trends: The initial guess D1 performs slightly worse than D2 for maxcg=1,

but appears to be better for maxcg>1. The number of successful calculations tends to decrease for higher values of maxcg and maxpur. Performing McWeeny transformations mainly in the last CG cycle (mpurif=0) seems to be better than doing them always (mpurif=1), and much better than doing them almost never (mpurif=-1, data not shown). Finally, density matrix extrapolation may be helpful in some cases, but the overall benefits are certainly small. The main finding from these tests is, however, that there is no robust and reliable procedure for SCF/CG-DMS convergence when starting from a diagonal density matrix guess.

This conclusion is corroborated by the results for the MNDO/d set with 366 molecules (see Tables 5-10). Due to the presence of d orbitals at the heavier elements, these molecules are more difficult to converge: the electrons can be distributed over more basis orbitals, and the populations are normally less uniform for an spd basis compared with an sp basis. Consequently, the proportion of successful calculations is significantly smaller. The unsuccessful cases again involve a change of the trace during McWeeny transformations which then usually leads to dianions or even more highly charged polyanions. The performance of the initial guess improves in the sequence D1 < D2 < D3. The relative merits of the other options are similar as in the case of the CHNO set and thus need not be discussed again.

In summary, the systematic tests in Tables 1-10 clearly indicate that the diagonal guesses for the initial density matrix that are commonly used in conventional semiempirical SCF calculations are not sufficient for the SCF/CG-DMS approach. A better initial guess is required to provide robust and reliable convergence.

Table 1: Number of successful MNDO calculations for the CHNO set with 218 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D1, McWeeny transformations in the last CG cycle (mpurif=0).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	214	214	205	205	205	205
2	208	208	207	209	203	205
3	206	208	202	203	201	202
4	192	194	193	195	190	192
5	169	172	155	159	148	152

Table 2: Number of successful MNDO calculations for the CHNO set with 218 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D1, McWeeny transformations in each CG cycle (mpurif=1).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	214	214	205	205	205	205
2	204	205	197	197	164	164
3	195	195	193	193	159	159
4	185	185	188	188	169	169
5	174	174	187	187	169	169

Table 3: Number of successful MNDO calculations for the CHNO set with 218 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D2, McWeeny transformations in the last CG cycle (mpurif=0).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	217	217	212	212	190	190
2	210	210	185	189	174	175
3	183	189	178	186	168	179
4	150	168	141	158	138	155
5	97	123	95	110	98	109

Table 4: Number of successful MNDO calculations for the CHNO set with 218 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D2, McWeeny transformations in each CG cycle (mpurif=1).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	217	217	212	212	190	190
2	181	186	177	178	142	145
3	153	158	140	145	112	116
4	101	108	111	110	111	111
5	89	93	105	107	110	112

Table 5: Number of successful MNDO/d calculations for the MNDO/d set with 366 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D1, McWeeny transformations in the last CG cycle (mpurif=0).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	246	248	200	201	143	143
2	199	177	142	144	111	116
3	121	132	105	113	80	94
4	81	97	46	71	25	55
5	11	44	13	41	7	34

Table 6: Number of successful MNDO/d calculations for the MNDO/d set with 366 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D1, McWeeny transformations in each CG cycle (mpurif=1).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	246	248	200	201	143	143
2	137	141	94	97	64	65
3	59	62	29	31	33	37
4	6	18	9	15	26	32
5	6	12	9	14	24	29

Table 7: Number of successful MNDO/d calculations for the MNDO/d set with 366 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D2, McWeeny transformations in the last CG cycle (mpurif=0).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	293	298	235	240	153	157
2	224	230	170	173	134	142
3	137	148	120	140	89	113
4	78	104	69	100	69	97
5	61	100	47	87	39	85

Table 8: Number of successful MNDO/d calculations for the MNDO/d set with 366 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D2, McWeeny transformations in each CG cycle (mpurif=1).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	293	298	235	240	153	157
2	153	163	84	91	71	78
3	72	88	60	60	47	49
4	47	58	38	41	39	39
5	28	40	31	36	40	44

Table 9: Number of successful MNDO/d calculations for the MNDO/d set with 366 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D3, McWeeny transformations in the last CG cycle (mpurif=0).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	303	307	231	237	185	186
2	274	275	225	227	187	188
3	197	206	183	186	173	174
4	161	169	134	145	118	136
5	99	116	75	90	54	73

Table 10: Number of successful MNDO/d calculations for the MNDO/d set with 366 molecules for different options (maxcg, maxpur): CG-DMS approach without and with extrapolation, initial guess D3, McWeeny transformations in each CG cycle (mpurif=1).

maxpur	1		2		3	
maxcg	std	ext	std	ext	std	ext
1	303	307	231	237	185	186
2	215	218	144	146	112	112
3	149	153	100	100	91	92
4	97	103	81	84	81	86
5	61	69	72	78	81	83

5.3 Initial density matrix from diagonalization

As discussed previously (see section 4.4) we have implemented another alternative option to construct an initial density matrix for large molecules: a block-diagonal guess is assembled from non-converged fragment density matrices which are obtained by performing one or more conventional SCF iterations for user-defined fragments. In the case of small molecules, fragmentation is not sensible, and the conventional SCF iterations are carried out for the whole molecule to generate the initial density matrix for the SCF/CG-DMS procedure by Fock matrix diagonalization. This is done for testing purposes only, of course, since there is no practical need for SCF/CG-DMS when a conventional SCF calculation can be performed for the whole molecule.

We have investigated the convergence of the SCF/CG-DMS procedure when starting from such initial density matrices that are generated through linfrg conventional SCF iterations. The full range of options presented in Tables 1-10 (maxcg, maxpur, mpurif, extrapolation) were tested both for the CHNO set and the MNDO/d set. In all cases, all SCF/CG-DMS runs converged and provided the correct results. This is true already for linfrg=1 and also holds for linfrg>1, of course. Hence, we find robust and reliable convergence for this initial guess which is therefore adopted as our standard. In an overall assessment, the convergence behaviour is similar for different values of linfrg, and it is thus recommended to use linfrg=1 as the default value.

The reasons for this safe convergence become clear if we look again at the example of ozone (see section 5.2). The initial guess obtained by a single Fock matrix diagonalization is idempotent and has the correct trace, by construction. The density matrix updates in the CG search perturb the idempotency only slightly since the eigenvalues of the resulting density matrix deviate from the proper values of 0 and 1 only slightly (always by less than 0.01). Consequently, the McWeeny purification can be applied with confidence, and the resulting trace remains essentially at the correct value of 9 (deviations of less than 0.000001 throughout). The problems encountered with the diagonal guess (see section 5.2) are thus avoided from the very beginning because the initial guess from diagonalization is properly idempotent.

All further SCF/CG-DMS results in this thesis are based on initial density matrices obtained through matrix diagonalization.

5.4 Comparison of CG-DMS implementations and options

Having established reliable convergence, we now turn to a comparison of the different CG-DMS implementations in our code (see section 4).

a) ITERCG: Full-matrix version with precomputation of all integrals.
b) DIRCG : Integral-direct full-matrix version.
c) DIRCGS: Integral-direct sparse-matrix version.

Using a variety of CG-DMS options (see above) and applying no cutoffs, all three implementations yield identical results for all 218 molecules of the CHNO set and for all 366 molecules of the MNDO/d set. The computed heats of formation agree with each other and with those from conventional SCF calculations (within the limits of the SCF convergence criteria). Moreover, for a given choice of CG-DMS options, the three implementations require essentially the same overall number of SCF iterations, CG cycles, and matrix multiplications for the whole CHNO and MNDO/d set, respectively, apart from very rare exceptions due to different rounding.

Tables 11 and 12 document these numbers as obtained from the ITERCG code, to indicate how they depend on the CG-DMS options; those from DIRCG and DIRCGS are mostly the same (with occasional differences of the order of 1). Increasing the allowed number of CG cycles per SCF iteration (maxcg) generally reduces the overall number of SCF iterations needed, both for the CHNO set (Table 11) and the MNDO/d set (Table 12), but also tends to increase the overall number of CG cycles and matrix multiplications. An exception to the latter trend is the transition from maxcg=1 to maxcg=2 in the MNDO/d set where all these numbers decrease: in this case, the overall SCF/CG-DMS convergence is obviously improved by allowing more CG cycles in the inner loop, whereas a further increase (maxcg>2) may be counterproductive as far as the overall workload is concerned (cf. the overall number of matrix multiplications). The maximum number of allowed McWeeny transformations (maxpur) has almost no influence on the overall number of SCF and CG steps that are required, and it also affects the overall number of matrix multiplications only to a small extent.

Table 11: Total number of SCF iterations, CG cycles, and matrix multiplications (MM) for all 218 molecules of the CHNO set for different CG-DMS options (maxcg=1-5, maxpur=1-3, mpurif=0): MNDO single-point energy calculations, initial guess from matrix diagonalization, standard SCF convergence criteria (iscf=6, iplscf=6), no density matrix extrapolation, no cutoffs, ITERCG code.

maxcg	maxpur = 1			maxpur = 2			maxpur = 3		
	SCF	CG	MM	SCF	CG	MM	SCF	CG	MM
1	6731	6731	74147	6731	6731	74839	6731	6731	74879
2	4354	8289	91295	4354	8289	92211	4354	8289	92251
3	4001	10317	113591	4001	10317	114377	4001	10317	114417
4	3917	12514	137760	3917	12514	138536	3917	12514	138576
5	3901	14439	158935	3901	14439	159727	3901	14439	159727

Table 12: Total number of SCF iterations, CG cycles, and matrix multiplications (MM) for all 366 molecules of the MNDO/d set for different CG-DMS options (maxcg=1-5, maxpur=1-3, mpurif=0): MNDO/d single-point energy calculations, initial guess from matrix diagonalization, standard SCF convergence criteria (iscf=6, iplscf=6), no density matrix extrapolation, no cutoffs, ITERCG code.

maxcg	maxpur = 1			maxpur = 2			maxpur = 3		
	SCF	CG	MM	SCF	CG	MM	SCF	CG	MM
1	19579	19579	215587	19579	19579	217423	19579	19579	217457
2	9824	18977	209009	9825	18983	211483	9825	18983	211517
3	8496	21870	240828	8495	21867	242899	8495	21867	242933
4	8203	26545	292237	8203	26545	294437	8203	26545	294471
5	8116	30954	340744	8116	30954	343002	8116	30954	343036

In a triply iterative scheme with nested SCF, CG, and McWeeny cycles (see section 4.2, schemes 5 and 6) it is obviously important to establish an optimum approach towards convergence. As discussed before (see section 4.2) it may be efficient not to insist on convergence in the inner loops during the initial stage of the overall SCF procedure and thus allow only few CG and McWeeny cycles in general (which will be sufficient for convergence in the final SCF stage). The data in Tables 11 and 12 are useful for an optimum choice of the relevant CG-DMS options (maxcg,

maxpur), but a more direct assessment is possible on the basis of the actual computation times which are listed in Tables 13 and 14.

In the case of the CHNO set (Table 13), the cpu times tend to increase slightly between maxcg=1 and maxcg=2 (in spite of the overall reduction in the number of SCF cycles, Table 11), and they increase further for larger values of maxcg. In the case of the MNDO/d set (Table 14), the overall cpu times decrease when going from maxcg=1 to maxcg=2, and then start to increase again, particularly for maxcg=4 and maxcg=5. On the basis of these results, we adopt maxcg=2 as default value. The MNDO/d calculations are generally somewhat harder to converge than the MNDO calculations, and one is probably on the safe side when choosing the default value to reflect the more difficult cases. As anticipated from the earlier discussion, the choice of maxpur affects the cpu times only to a very minor extent, and its exact value thus seems less important. We adopt maxpur=2 as our standard.

Comparing the three different CG-DMS implementations, the cpu times increase in the order ITERCG < DIRCG << DIRCGS. The higher workload in DIRCG relative to ITERCG is due to the repeated integral evaluations in the integral-direct mode. The large increase between DIRCG and DIRCGS is caused by the different handling of matrix operations: DIRCG employs highly optimized BLAS routines (e.g., DGEMM for matrix multiplication) that work with full square matrices, whereas DIRCGS uses sparse-matrix techniques with considerable overhead and unoptimized routines from the SPARSKIT2 library [86] (see section 4.3). These differences increase the overall cpu times roughly by an order of magnitude (Tables 13 and 14).

Table 13: Cpu times (sec) for all 218 molecules of the CHNO set for different CG-DMS options (maxcg=1-5, maxpur=1-3, mpurif=0): MNDO single-point energy calculations, initial guess from matrix diagonalization, standard SCF convergence criteria (iscf=6, iplscf=6), no density matrix extrapolation, no cutoffs, ITERCG vs DIRCG vs DIRCGS implementation (see text). The cpu times were measured on a Compaq XP1000 workstation (ath4, 667 MHz).

maxcg	maxpur = 1			maxpur = 2			maxpur = 3		
	ITERCG	DIRCG	DIRCGS	ITERCG	DIRCG	DIRCGS	ITERCG	DIRCG	DIRCGS
1	5.2	8.3	67.4	5.3	7.4	67.9	5.2	7.3	67.9
2	6.3	7.3	73.9	6.4	7.6	74.3	6.4	7.6	74.4
3	6.7	8.0	89.4	6.7	7.9	89.8	6.7	8.0	89.8
4	9.1	10.3	107.2	9.2	10.4	107.5	9.2	10.4	107.7
5	9.2	10.4	122.3	9.2	10.4	123.1	9.2	10.5	123.1

Table 14: Cpu times (sec) for all 366 molecules of the MNDO/d set for different CG-DMS options (maxcg=1-5, maxpur=1-3, mpurif=0): MNDO/d single-point energy calculations, initial guess from matrix diagonalization, standard SCF convergence criteria (iscf=6, iplscf=6), no density matrix extrapolation, no cutoffs, ITERCG vs DIRCG vs DIRCGS implementation (see text). The cpu times were measured on a Compaq XP1000 workstation (ath4, 667 MHz).

maxcg	maxpur = 1			maxpur = 2			maxpur = 3		
	ITERCG	DIRCG	DIRCGS	ITERCG	DIRCG	DIRCGS	ITERCG	DIRCG	DIRCGS
1	49.3	101.4	734.0	49.6	100.8	737.5	49.6	101.0	733.0
2	41.9	68.3	618.3	42.3	68.4	625.8	42.2	68.8	627.3
3	46.1	67.5	683.7	46.3	67.8	692.9	46.3	68.0	689.3
4	54.1	74.4	817.7	54.5	74.9	821.3	54.4	75.0	823.7
5	62.0	83.7	941.9	62.5	82.8	951.2	62.3	80.7	944.8

Finally it is instructive to compare the performance of our three CG-DMS implementations also with that of conventional SCF implementations using Fock matrix diagonalization or pseudodiagonalization, i.e., the standard version ITER and the integral-direct version DIRECT (see section 4.1). Some relevant data are collected in Table 15. The total number of required SCF iterations is somewhat higher in SCF/CG-DMS than in the conventional SCF approach, by almost

10 % for the CHNO test set and by around 20 % for the MNDO/d test set. The computational effort increases more strongly, as expected from the inherent workload of the two approaches, e.g., with regard to the number of matrix multiplications (see section 4.2). Comparing the two most efficient variants, ITER and ITERCG, the former is faster by a factor of about 4 for both test sets. On the other hand, the linear scaling SCF/CG-DMS version, DIRCGS, is intrinsically slower than ITER by about a factor of about 50, and therefore a high degree of sparsity is expected to be necessary before the linear scaling implementation becomes most efficient.

Table 15: Total number of SCF cycles and cpu times (sec) for all molecules of the CHNO and MNDO/d sets for different SCF procedures: ITER vs DIRECT vs ITERCG vs DIRCG vs DIRCGS (see text): Single-point energy calculations, standard options (iscf=6, iplscf=6, linfrg=1, maxcg=2, maxpur=2, mpurif=0), no density matrix extrapolation, no cutoffs. The cpu times were measured on a Compaq XP1000 workstation (ath1, 667 MHz).

	Set	ITER	DIRECT	ITERCG	DIRCG	DIRCGS
SCF	CHNO	4020	4020	4354	4354	4354
	MNDO/d	8055	8056	9823	9823	9823
cpu	CHNO	1.3	2.5	5.2	6.7	68.4
	MNDO/d	9.1	29.3	40.7	66.9	567.8

Chapter 6

Validation and performance for large systems

6.1 Overview

This section reports SCF/CG-DMS calculations on polyglycines, water clusters, proteins, and DNA molecules which were carried out to study the performance of the code for large molecules, particularly with regard to its scaling behaviour. Unless noted otherwise in the following subsections, we used the following conventions and options (see section 5 and Appendix 4).

Single-point AM1 computations were done at fixed input geometries that were mostly taken from the literature (details see below). Standard SCF convergence criteria were normally applied (iscf=6, iplscf=6). The initial density matrix was obtained from a block-diagonal guess (usually with linfrg=1). The SCF/CG-DMS calculations employed the sparse-matrix integral-direct version of the code (DIRCGS). In the CG-DMS procedure, the maximum number of allowed GC cycles and McWeeny transformations was set to maxcg=4 and maxpur=2, respectively. In addition to the maxpur McWeeny transformations at the end of each CG cycle, one such purification step was included at each GC step (mpurif=0). The chosen options correspond to those recommended in section 5, except for maxcg=4, which was originally adopted to be on the safe side with regard to convergence. The experience gained in later timing studies suggests (see section 5.4) that the use of maxcg=4 instead of maxcg=2 slows down DIRCGS calculations by about 30 % (see Tables 13 and 14). This should be taken into account when judging the cpu times reported below.

The validation work on small molecules (section 5) did not employ any cutoffs. The use of cutoffs is essential in large systems, of course, in order to achieve linear scaling (section 3). We have chosen a single cutoff parameter (cutoff=X) which is applied to the density matrix P, the Fock matrix F, and other intermediate matrices such as the product matrix FP. Density matrix elements are neglected if their absolute value is smaller than the cutoff X. Likewise, Fock matrix elements

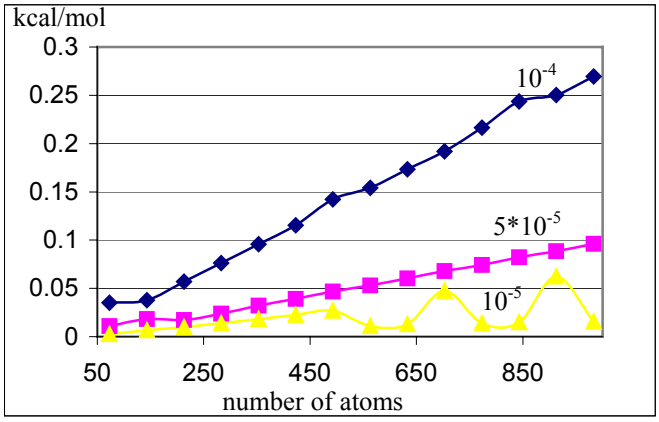
are discarded if their absolute value in eV is smaller than X. In intermediates such as FP, the matrix elements are removed if their absolute value is smaller than X/10 (i.e., the cutoff is tighter in this case, and there is also an option to completely avoid the filtering of intermediates). Typical values of X range between 10^{-4} and 10^{-8}.

6.2 Polyglycines

The input files for linear-chain polyglycines of the general formula NH_2-CH_2-CO-(NH-CH_2-CO)$_n$-NH-CH_2-COOH (7n+17 atoms) were generated by an automatic program written for this purpose. AM1 calculations were carried out for the systems up to n = 138 (983 atoms) using both the SCF/CG-DMS approach (DIRCGS, see above) with cutoffs (X = 10^{-4}, $5*10^{-5}$, 10^{-5}) and the conventional SCF procedure (ITER). The results are plotted in Figures 6 and 7.

The use of cutoffs invariably causes some error. Figure 6 shows that, for the computed heats of formation, the deviations from the reference results without cutoffs (ITER) increase with the chosen value of the cutoff, as expected. They also increase with molecular size, more or less in a linear fashion (with some fluctuations for X=10^{-5}). On an absolute scale, the deviations seem tolerable: for the largest cutoff of X=10^{-4}, they are of the order of 0.03 kcal/mol per 100 atoms.

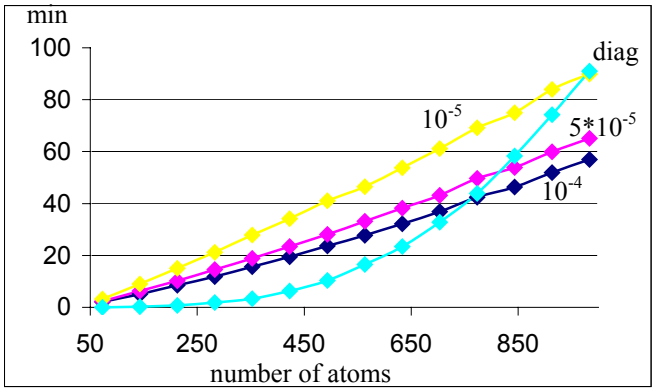
Figure 6: Differences between computed heats of formation (kcal/mol), DIRCGS vs ITER (see text), as a function of system size (number of atoms in polyglycines). Three different cutoffs were employed.



The cpu times (Figure 7) for the SCF/CG-DMS calculations scale linearly with molecular size, for each of the three cutoffs. Tightening the cutoff from X=10^{-4} to X=10^{-5} increases the cpu time for the polyglycines by a factor of about 1.5. For the sake of comparison, Figure 7 also includes the

measured cpu times for corresponding conventional SCF calculations which scale more steeply (cubically, see section 2.2). They are faster than the SCF/CG-DMS calculations for small polyglycines and become slower only beyond a certain molecular size which depends on the chosen cutoff: for $X=10^{-4}$, this crossover occurs around 750 atoms.

Figure 7: Cpu times (min) for one SCF iteration as a function of system size (number of atoms in polyglycines). The conventional SCF treatment (ITER) is compared to CG-DMS calculations (DIRCGS) with three different cutoffs.



In another published CG-DMS implementation [19] significantly lower crossover points have been reported for polyglycines. A direct comparison between our results and these published results [19] is difficult due to the differences in hardware, software, input geometries, and computational options. We have decided to attempt such a comparison nevertheless, using glycine decapeptide as an example (73 atoms, third data point in Figure 4 of ref. 19, cpu time of about 15 sec both for conventional SCF and for CG-DMS with the higher threshold). The different hardware (Compaq XP 1000 in our case vs IBM RS/6000 3CT [19]) is taken into account through the SPEC CPU95 benchmark data [88] which indicate that the Compaq workstation is faster than the IBM workstation by a factor of 6.4. All our measured cpu times were corrected by this factor and normalized to a total of 10 SCF iterations [19]. Under these conventions, the conventional SCF calculations (ITER) run about five times faster with our program than with the Gaussian version used in the literature [19]. Our most efficient CG-DMS approach (ITERCG, full-matrix version with precomputed integrals, no cutoffs, recommended options: maxcg=2, maxpur=2, mpurif=0, linfrg=1) is about as fast as the published CG-DMS approach [19]. However, this comparison is not quite fair, since the latter employs an integral-direct sparse-matrix code (with cutoffs) that can also be applied to much larger systems. We have therefore run the closest analogue to the published work that we can access with our code (DIRCGS, integral-direct sparse-matrix version; same cutoffs [19]: $X=5*10^{-5}$ au for the Fock matrix, $X=5*10^{-5}$ for the density matrix, hard atom-atom

cutoff of 10 Å for the integrals; same CG-DMS options [19]: maxcg=4, maxpur=2, mpurif=0; linfrg=1). In this CG-DMS calculation, our implementation is about five times slower. Profiling shows that in this case more than 85% of the cpu time are spent on sparse matrix multiplications (including the symbolic multiplications needed for allocation purposes in the LIG scheme, see section 4.3). The corresponding routines (amubp, amubdgp) have been taken from the SPARSKIT2 library [86] without change. It would thus seem possible to improve the performance of the DIRCGS code significantly by optimizing or rewriting these routines using techniques recently suggested in the literature for sparse matrix multiplication [82,83]. The implementation of the FF scheme may yield further improvements (see section 4.3).

In summary, the lower crossover points reported for polyglycines [19] are due to the fact that the published work employs a slower conventional SCF treatment and a faster SCF/CG-DMS treatment. It should also be noted that the chosen cutoffs [19] (particularly the hard atom-atom integral cutoff) cause a significant deviation (12 kcal/mol) of the computed heat of formation of glycine decapeptide from the reference value. This deviation can be reduced by tighter cutoffs, of course, which however lead to higher cpu times for SCF/CG-DMS and thus to later crossover.

6.3 Water clusters

The input geometries for the water clusters $(H_2O)_n$ were taken from the internet [89]. These geometries are three-dimensional (3D) in the sense that they are globular and extend into all directions, in contrast to the linear-chain polyglycines which are essentially one-dimensional (1D) in their shape.

We have carried out AM1 calculations for the clusters up to n = 1195 (3585 atoms) using again both the SCF/CG-DMS approach (DIRCGS) with cutoffs (X = 10^{-4}, $5*10^{-5}$, 10^{-5}) and the conventional SCF procedure (ITER). The results are plotted in Figures 8 and 9. In most aspects, they are qualitatively similar to those for the polyglycines so that the discussion can be brief.

The deviations of the computed heats of formation from the reference results (Figure 8) again increases with the chosen value of the cutoff and with molecular size (as before typically by 0.03 kcal/mol per 100 atoms for X=10^{-4}). The cpu times (Figure 9) for the SCF/CG-DMS again exhibit approximately linear scaling with molecular size. The gains from exploiting sparsity are less pronounced in compact 3D systems than in extended 1D systems, and it is therefore not surprising that the crossover point between the SCF and SCF/CG-DMS calculations occurs considerably later

in the water clusters than in the polyglycine chains (see Figure 9, around 2800 vs 750 atoms for $X=10^{-4}$). For the same reason, tightening the cutoff from $X=10^{-4}$ to $X=10^{-5}$ increases the cpu time for the water clusters more strongly, by a factor of about 2.5 (compared with 1.5 for the polyglycines).

Figure 8: Differences between computed heats of formation (kcal/mol), DIRCGS vs ITER (see text), as a function of system size (number of atoms in water clusters). Three different cutoffs were employed.

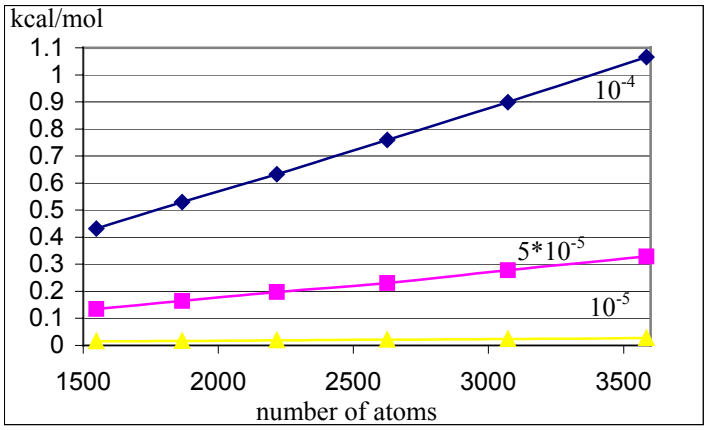
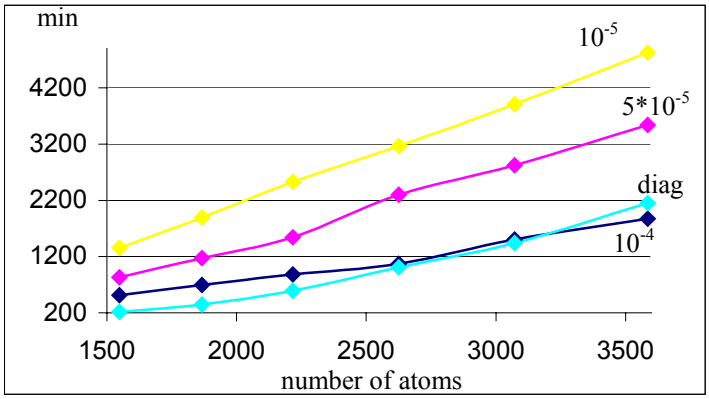


Figure 9: Cpu times (min) for one SCF iteration as a function of system size (number of atoms in water clusters). The conventional SCF treatment (ITER) is compared with CG-DMS calculations (DIRCGS) with three different cutoffs.



6.4 Proteins and DNA molecules

Large biochemical molecules are a natural application area for linear scaling semiempirical SCF methods. We have selected a set of 47 such molecules for further testing and validation which includes peptides, proteins, and DNA molecules as well as protein-metal and protein-DNA complexes. The size of the selected systems ranges from 137 to 9595 atoms. Geometries were taken from the Protein Data Bank (PDB) [90,91] and labeled by the corresponding PDB entry. Missing hydrogen atoms were added to these structures using the Insight II modelling software [92] (assuming a pH value of 7). The adopted SCF convergence criteria (iscf=4, iplscf=4) were less tight than usual. Otherwise standard CG-DMS options were employed throughout (see sections 4.4 and 6.1), except for the specifications given below.

Three series of AM1 single-point calculations were carried out that differ in the chosen cutoffs (X) and the number of conventional diagonalizations during the formation of the initial block-diagonal guess for the density matrix (linfrg).

a) Cutoff $X=10^{-4}$, linfrg=1 (see Table 16).
b) Cutoff $X=10^{-4}$, linfrg=2 (see Table 17).
c) Cutoff $X=10^{-5}$, linfrg=1 (see Table 18).

In each series, the SCF/CG-DMS calculations were done both without and with density matrix extrapolation (in the latter case starting at the second SCF iteration, nstart=2). Tables 16-18 list the cpu time (t), the final heat of formation (H), and the number of required SCF iterations (IT) for each of these calculations. At the beginning of each entry, the corresponding molecule is identified, usually by its PDB label, the number of atoms (N), and the total charge (Z) (see footnotes for further details). Appendix 5 specifies the chemical names for all entries in Tables 16-18.

Table 16: Single-point SCF/CG-DMS calculations on 47 biochemical molecules: AM1, first series, $X=10^{-4}$, linfrg=1 (see text).

Identification			without extrapolation			with extrapolation		
PDB	N	Z	t(min)	H(kcal/mol)	IT	t(min)	H(kcal/mol)	IT
1B32	9595	-16	5895.8	-44011.47602	31	5292.6	-44011.53534	28
2QWE	7151	-4	3232.6	-36229.21710	19	3054.3	-36229.28753	27
1IXH	4855	0	1466.8	6002.05321	29	985.6	6002.16781	16

subt[a]	3854	-2	1216.8	-11549.42858	18	1341.7	-11549.57321	33
1IAU	3607	+14	3068.0	-4856.93852	29	3545.6	-4857.11190	49
1E3B	2582	+3	912.5	-6349.56898	28	772.1	-6349.57486	27
lyso[a]	1960	+8	610.9	-3622.13671	16	579.4	-3622.31184	19
1EV3	1776	-9	387.8	-8273.80984	20	334.6	-8273.84748	20
1A3K	1577	+3	256.0	-3464.58875	15	219.0	-3464.73527	17
1CC8	1550	+3	435.2	-8378.18116	42	321.9	-8378.18824	23
ribo[a]	1470	+6	215.7	-1877.09172	15	206.4	-1877.29931	19
1AOY	1239	+3	237.1	-2837.82796	32	164.8	-2837.81213	16
1IKL	1149	+3	209.7	-1341.75040	22	163.4	-1341.72862	21
1A43	1145	-2	201.2	-3232.17277	22	164.5	-3232.20431	19
1AFJ	1071	+3	273.0	-1619.67309	16	258.1	-1619.80575	19
1IRN	1042	-11	166.2	-2612.49743	19	150.9	-2612.50792	21
cspa[a]	1010	0	201.7	-2581.11459	16	179.7	-2581.12059	20
2R63	989	+2	170.4	-2678.84633	32	118.3	-2678.84397	17
5EBX	916	+1	153.9	-2339.14029	14	130.3	-2339.25655	15
2OVO	904	-1	145.4	-3757.62790	18	127.2	-3757.64477	20
bpti[a]	892	+6	177.4	-458.05219	15	206.8	-458.18736	31
1A4T	836	-9	442.8	-3881.30483	31	413.9	-3881.29924	29
2REL	829	+2	78.4	-1081.26337	13	73.7	-1081.43196	21
1A7F	751	-5	97.1	-1815.69770	19	86.7	-1815.69465	21
2SH1	695	-1	120.0	-1708.31576	18	98.5	-1708.34752	16
1B8W	668	+4	118.9	54.91029	25	102.9	54.90974	24
cram[a]	642	0	91.9	750.78709	22	68.7	750.80002	17
1ATO	605	-18	199.7	-2332.30406	23	190.1	-2332.31462	20
1AML	597	-4	35.3	-901.41865	15	30.6	-901.44081	16
1OKA	574	-16	107.7	-2006.86354	20	107.6	-2006.87050	20
2PTA	550	+6	97.1	295.36607	25	89.6	295.35973	24
1AXH	523	-2	68.6	-1640.88420	16	59.3	-1640.90060	17
1BH0	469	0	32.6	-889.73692	17	26.7	-889.73671	19
1ATF	450	-2	23.7	-1735.54803	14	17.8	-1735.64312	10
1BXJ	438	-1	42.0	-513.20568	17	36.7	-513.21782	19
toxi[a]	410	-3	57.4	-1202.16611	23	45.7	-1202.16507	19

1AFX	387	-11	143.4	-1598.95171	29	134.1	-1598.94879	24
1BQF	371	+1	28.3	-451.22706	24	24.5	-451.23441	23
2MAG	359	+4	22.5	171.11162	13	18.4	171.08777	12
8TFV	351	+6	19.9	680.06031	11	17.0	680.12373	11
2NR1	339	-2	21.4	-842.66298	20	14.7	-842.65950	13
1ALE	293	-2	14.1	-613.93705	14	12.1	-613.94101	15
1BCV	281	0	6.1	-709.71559	14	4.3	-709.70113	9
1A3J	265	-2	14.8	-634.38842	13	12.2	-634.40678	12
3CMH	260	-3	12.9	-477.68942	15	10.5	-477.69742	13
1CB3	193	+1	8.6	61.59949	14	7.4	61.58593	15
2SOC	137	0	4.7	-209.29484	12	3.9	-209.31967	12

(a) Input geometries obtained from K. M. Merz, private communication (1999).

Table 17: Single-point SCF/CG-DMS calculations on 40 biochemical molecules: AM1, second series, $X=10^{-4}$, linfrg=2 (see text).

Identification			without extrapolation			with extrapolation		
PDB	N	Z	t(min)	H(kcal/mol)	IT	t(min)	H(kcal/mol)	IT
1IXH	4855	0	1584.4	6002.11883	27	1278.4	6002.06409	23
subt[a]	3854	-2	1334.8	-11549.34639	19	1336.0	-11549.62743	24
1E3B	2582	+3	1049.7	-6349.50583	24	900.0	-6349.52766	25
lyso[a]	1960	+8	852.8	-3622.30156	37	697.1	-3622.29495	21
1EV3	1776	-9	444.6	-8273.79906	18	410.2	-8273.85179	21
1A3K	1577	+3	325.1	-3464.83812	18	271.7	-3464.88205	15
1CC8	1550	+3	411.5	-8378.18191	21	337.7	-8378.16961	16
ribo[a]	1470	+6	279.8	-1877.26340	23	217.4	-1877.30811	17
1AOY	1239	+3	242.6	-2837.85040	20	209.1	-2837.88547	21
1IKL	1149	+3	237.4	-1341.70509	19	200.9	-1341.72428	17
1A43	1145	-2	218.8	-3232.15887	15	186.3	-3232.20706	16
1AFJ	1071	+3	358.9	-1619.80115	21	307.2	-1619.77826	19
1IRN	1042	-11	206.1	-2612.49345	22	183.2	-2612.49535	24

cspa[a]	1010	0	253.1	-2581.07893	18	208.5	-2581.12627	15
2R63	989	+2	181.7	-2678.86926	19	133.5	-2678.80975	13
5EBX	916	+1	196.0	-2339.29464	19	165.7	-2339.31506	16
2OVO	904	-1	161.6	-3757.61476	17	138.2	-3757.63881	16
bpti[a]	892	+6	237.0	-458.16681	19	206.2	-458.09570	15
2REL	829	+2	101.6	-1081.44675	17	77.3	-1081.36056	12
1A7F	751	-5	99.2	-1815.69296	17	95.0	-1815.70656	17
2SH1	695	-1	215.1	-1708.19315	27	198.5	-1708.17520	24
1B8W	668	+4	145.2	54.97109	14	149.8	54.98757	16
cram[a]	642	0	91.6	750.80572	16	80.4	750.77152	17
1AML	597	-4	40.4	-901.40324	13	37.8	-901.43211	19
2PTA	550	+6	128.6	295.50053	16	140.6	295.33284	23
1AXH	523	-2	73.8	-1640.86745	16	64.4	-1640.88841	16
1BH0	469	0	32.9	-889.72785	15	27.9	-889.75832	15
1ATF	450	-2	31.7	-1735.35199	13	44.0	-1735.60185	43
1BXJ	438	-1	44.1	-513.19526	15	37.6	-513.21206	15
toxi[a]	410	-3	51.7	-1202.15704	17	44.3	-1202.17198	17
1BQF	371	+1	25.9	-451.18744	15	27.7	-451.23034	20
2MAG	359	+4	24.3	171.20760	12	19.9	171.10895	12
8TFV	351	+6	23.1	680.10140	15	21.7	680.09050	20
2NR1	339	-2	20.8	-842.65434	16	14.9	-842.63631	11
1ALE	293	-2	14.2	-613.92437	14	12.4	-613.93567	15
1BCV	281	0	6.5	-709.71047	13	5.7	-709.72480	13
1A3J	265	-2	16.6	-634.40924	20	12.2	-634.39805	11
3CMH	260	-3	30.0	-477.68029	29	22.4	-477.69194	21
1CB3	193	+1	9.8	61.57804	15	7.5	61.58661	12
2SOC	137	0	5.3	-209.32218	14	4.0	-209.31519	12

(a) See footnote of Table 16.

Table 18: Single-point SCF/CG-DMS calculations on 44 biochemical molecules: AM1, third series, $X=10^{-5}$, linfrg=1 (see text).

Identification			without extrapolation			with extrapolation		
PDB	N	Z	t(min)	H(kcal/mol)	IT	t(min)	H(kcal/mol)	IT
1IXH	4855	0	6062.0	5999.46580	34	4178.3	5999.43803	12
subt[a]	3854	-2	5309.8	-11551.54931	32	4048.9	-11551.50290	15
1E3B	2582	+3	3724.4	-6351.09644	31	2804.4	-6351.10051	24
lyso[a]	1960	+8	2748.8	-3623.68993	40	1703.0	-3623.64307	15
1EV3	1776	-9	1642.4	-8274.47860	26	1167.4	-8274.48256	12
1A3K	1577	+3	870.8	-3465.58896	25	696.9	-3465.60293	24
1CC8	1550	+3	1404.1	-8378.98312	23	1132.0	-8378.98386	21
ribo[a]	1470	+6	742.2	-1878.05133	25	571.3	-1878.05255	21
1AOY	1239	+3	801.0	-2838.45964	24	598.2	-2838.45967	19
1IKL	1149	+3	737.8	-1342.28685	24	554.7	-1342.28302	18
1A43	1145	-2	800.1	-3232.69960	24	612.4	-3232.69365	15
1AFJ	1071	+3	1096.3	-1620.43986	29	815.2	-1620.42957	15
1IRN	1042	-11	574.7	-2612.87587	22	465.9	-2612.87988	16
cspa[a]	1010	0	742.7	-2581.58479	22	605.6	-2581.58546	15
2R63	989	+2	515.8	-2679.35535	23	392.2	-2679.35682	18
5EBX	916	+1	577.1	-2339.76626	22	476.6	-2339.76438	15
2OVO	904	-1	505.1	-3758.01972	22	409.8	-3758.02170	17
bpti[a]	892	+6	613.1	-458.67513	30	466.7	-458.67518	19
1A4T	836	-9	1159.4	-3881.59514	22	1117.3	-3881.59572	20
2REL	829	+2	263.1	-1081.78179	19	211.3	-1081.79087	15
1A7F	751	-5	265.1	-1815.92983	14	221.7	-1815.93466	12
2SH1	695	-1	390.0	-1708.66664	22	307.8	-1708.66300	15
1B8W	668	+4	357.4	54.61136	21	262.6	54.61720	11
cram[a]	642	0	291.0	750.48985	20	221.6	750.49287	11
1ATO	605	-18	622.8	-2332.53739	17	609.6	-2332.55132	16
1AML	597	-4	86.2	-901.62306	19	68.7	-901.62157	13
1OKA	574	-16	306.9	-2007.08546	16	311.6	-2007.07874	16
2PTA	550	+6	259.6	295.05907	16	206.8	295.05233	11

1AXH	523	-2	218.6	-1641.12152	20	171.0	-1641.11741	13
1BH0	469	0	65.3	-889.90236	17	49.8	-889.90282	11
1ATF	450	-2	56.3	-1735.82037	19	42.9	-1735.82598	16
1BXJ	438	-1	119.8	-513.35274	16	99.9	-513.35486	13
toxi[a]	410	-3	115.8	-1202.29724	14	98.0	-1202.29784	12
1AFX	387	-11	303.5	-1599.08199	19	295.8	-1599.07968	16
1BQF	371	+1	53.4	-451.34266	14	45.8	-451.34410	12
2MAG	359	+4	53.7	170.90675	21	42.7	170.89922	16
8TFV	351	+6	44.1	679.90767	18	33.9	679.90604	12
2NR1	339	-2	39.9	-842.76738	16	32.8	-842.77537	16
1ALE	293	-2	26.6	-614.00378	14	21.5	-614.00727	11
1BCV	281	0	11.1	-709.77000	15	8.7	-709.77115	11
1A3J	265	-2	31.3	-634.50105	13	25.6	-634.50221	11
3CMH	260	-3	23.3	-477.75952	14	18.6	-477.76188	11
1CB3	193	+1	15.5	61.52620	13	12.0	61.53008	10
2SOC	137	0	8.0	-209.35889	13	7.0	-209.36157	11

(a) See footnote of Table 16.

Comparing the results without and with extrapolation, it is obvious that the final heats of formation generally differ by more than the adopted SCF convergence criterion for the energy (0.0001 eV = 0.0023 kcal/mol). When using relatively large cutoffs (Tables 16 and 17), these deviations are typically in the order of 0.01 kcal/mol for the smaller systems and 0.1 kcal/mol for the larger systems. With tighter cutoffs (Table 18) these deviations are generally smaller, often by factor of about 10, such that the results for the smaller systems now often agree within the SCF convergence criteria. To avoid premature SCF convergence, it is thus advisable to tighten either the CG-DMS cutoffs or the SCF convergence criteria (or both).

In the majority of cases (34 out of 47 in Table 16, 28 out of 40 in Table 17, and 30 out of 44 in Table 18) the calculations with extrapolation yield a lower total energy. The overall number of required SCF iterations is either lower or similar compared to calculations without extrapolation (871 vs 868 in Table 16, 800 vs 811 in Table 17, and 657 vs 980 in Table 18). Both these findings support the use of density matrix extrapolation in SCF/CG-DMS calculations on large molecules.

Comparing the results for different initial block-diagonal density matrices (linfrg=1 vs linfrg=2, Tables 16 and 17) it seems generally somewhat better to use two initial matrix diagonalizations (linfrg=2) rather than only one (linfrg=1) since the total number of required SCF iterations decreases when performing a second initial diagonalization, from 868/871 (Table 16) to 811/800 (Table 17) in calculations without/with extrapolation (for the 40 molecules that appear in both tables). Refinement of the initial intra-fragment density thus tends to facilitate convergence, but the gains are not dramatic.

Comparing the results for different cutoffs (Tables 16 and 18) it is obvious that smaller cutoffs lead to lower energies and that this lowering becomes more pronounced with increasing molecular size. This is consistent with our previous results for polyglycines (Figure 6) and water clusters (Figure 8). The price to be paid for this better precision is a higher cpu time and a higher memory demand: when tightening the cutoffs from $X=10^{-4}$ to $X=10^{-5}$, for example, the cpu times per SCF iteration typically rise by a factor of 2-3 for the systems studied (Tables 16 and 18) which is again compatible with analogous comparisons for 3D water clusters (Figure 9).

For further analysis of cutoff effects, we have carried out conventional SCF reference calculations for five peptides with 137-450 atoms (Table 19). Comparing the resulting heats of formation with the corresponding values for a cutoff of $X=10^{-4}$ (Table 16) confirms the rule of thumb that the latter are higher by about 0.03 kcal/mol per 100 atoms (see section 6.2 and 6.3). For three of these peptides, additional SCF/CG-DMS calculations were done for cutoffs ranging from $X=10^{-4}$ to $X=10^{-8}$, all other options being standard. The results are shown in Figures 10-12. It is obvious that a change of the cutoff from $X=10^{-4}$ to $X=5*10^{-5}$ reduces the error in the heat of formation by 60-75 % (at the expense of an increase in the cpu time by 19-63 % per SCF iteration). A further tightening of the cutoff beyond $X=10^{-5}$ yields heats of formation that would seem precise enough for most practical purposes. The choice $X=5*10^{-5}$ appears to be a reasonable compromise between precision and computational effort in semiempirical SCF/CG-DMS studies of large biomolecules.

Table 19: Conventional single-point AM1 calculations on peptides (see text and Tables 16-18 for notation).

PDB	N	Z	H(kcal/mol)	IT
1ATF	450	-2	-1735.83222	26
1BCV	286	0	-709.78068	20
1A3J	265	-2	-634.51243	18
1CB3	193	+1	61.51952	22
2SOC	137	0	-209.36601	20

Figure 10: SCF/CG-DMS heat of formation (kcal/mol) as function of the cutoff X: Peptide 2SOC (see text). The exponent m of the cutoffs $X=10^{-m}$ is given at the top.

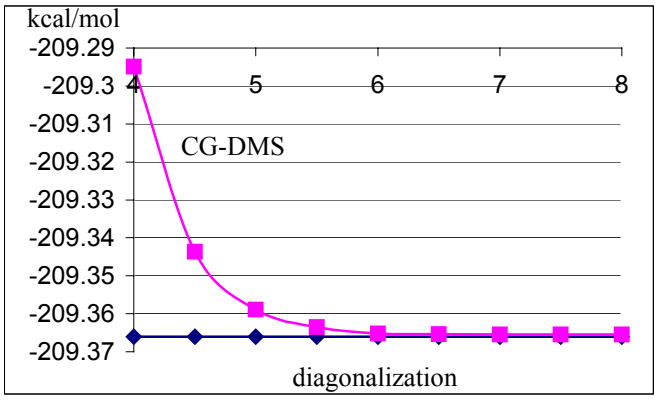


Figure 11: SCF/CG-DMS heat of formation (kcal/mol) as function of the cutoff X: Peptide 1A3J (see text). The exponent m of the cutoffs $X=10^{-m}$ is given at the top.

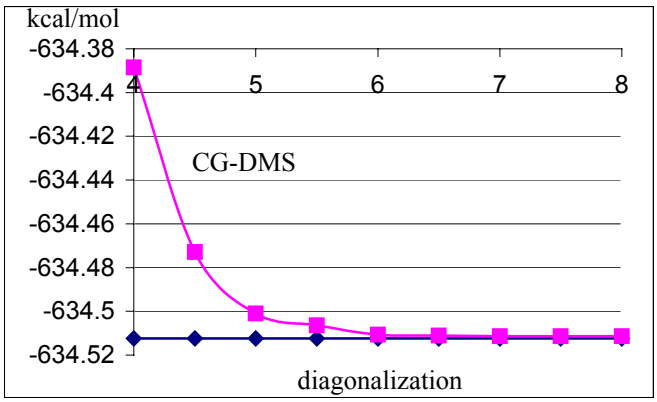
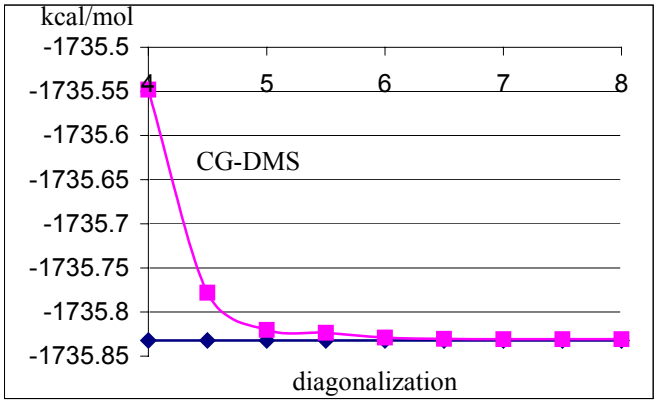


Figure 12: SCF/CG-DMS heat of formation (kcal/mol) as function of the cutoff X: Peptide 1ATF (see text). The exponent m of the cutoffs $X=10^{-m}$ is given at the top.



Biomolecular simulations often employ classical force fields such as CHARMM [93,94] or GROMOS [95,96] which normally assume fixed atomic charges that sum to an integer value for any given amino acid residue (i.e., to zero for neutral residues). The single-point SCF/CG-DMS calculations on peptides and proteins reported in this subsection provide self-consistent quantum-chemical charge distributions that can be used to check this assumption. Hence, for all peptides and proteins in Table 16, the net atomic charges from an AM1 Mulliken population analysis were summed up over each residue and subjected to a statistical evaluation. The results are collected in Table 20. They cover all 20 naturally occurring amino acids, because each of them appears in our validation set (between 56 and 353 times). It is obvious that the mean total charges per residue from the AM1 calculations are indeed close to the expected values from standard force fields: the deviations are at most 0.02 for neutral residues and 0.05-0.10 for charged residues. The root-mean-square deviations from the expected values are typically around 0.05 which indicates moderate charge fluctuations between the different residues. The maximum deviations encountered are normally of the order of 0.20 for a given residue (usually somewhat higher for charged residues, up to 0.32 for Asp). The maximum fluctuations (i.e., the differences between the corresponding maximum and minimum total charges in Table 20) are usually around 0.30-0.40; these values represent extreme individual cases, of course, but nevertheless they serve as a reminder that charge fluctuations and charge transfers do occur in proteins which are not captured by the standard force fields. It is one of the merits of linear scaling semiempirical SCF methods that such effects can be studied if needed in any specific application. The importance of these effects has already been demonstrated in several cases [67,68,75,76].

Table 20: Sum of net atomic charges per residue from AM1 calculations on peptides and proteins (see Table 16).

Label(a)	N(b)	max(c)	min(d)	mean(e)	rms(f)
Gly	353	0.15	-0.19	0.00	0.0543
Ala	340	0.15	-0.22	0.00	0.0543
Asn	244	0.15	-0.21	-0.02	0.0636
Ile	222	0.24	-0.16	-0.01	0.0565
Leu	311	0.14	-0.15	0.00	0.0560
Phe	181	0.11	-0.16	0.00	0.0423
Val	322	0.16	-0.15	0.00	0.0470
Pro	253	0.11	-0.23	-0.02	0.0636
His	94	0.15	-0.11	0.00	0.0548
Gln	146	0.13	-0.15	-0.01	0.0517
Trp	80	0.09	-0.11	0.00	0.0482
Thr	293	0.22	-0.15	-0.01	0.0540
Met	56	0.09	-0.11	-0.01	0.0475
Ser	352	0.18	-0.18	-0.01	0.0595
Tyr	172	0.23	-0.16	0.00	0.0585
Arg	176	1.10	0.76	0.94	0.0549
Cys (g)	135	0.24	-0.38	-0.03	0.0908
Lys	305	1.08	0.73	0.95	0.0502
Asp	269	-0.68	-1.11	-0.90	0.0645
Glu	224	-0.76	-1.12	-0.92	0.0571

(a) Standard symbols for amino acids [97].

(b) Number of occurrencies of the given residue in the molecules studied (Table 16).

(c) Maximum total charge.

(d) Minimum total charge.

(e) Mean total charge.

(f) Root-mean-square deviation of the computed total charges from the values expected for the given residue (0 in most cases; 1 for Arg and Lys; -1 for Asp and Glu).

(g) For a bridged fragment consisting of two cystein residues (see section 4.4).

Chapter 7

Biochemical applications

7.1 Introduction

Combined quantum mechanical / molecular mechanical (QM/MM) studies have recently been carried out in our group for several enzymatic reactions. This work has provided optimized QM/MM geometries of the relevant minima and transition states, e.g., for the oxygenation reaction catalyzed by p-hydroxybenzoate hydroxylase [98] and for the proton transfers catalyzed by triosephosphate isomerase [99]. It is of interest to perform single-point QM calculations at these optimized QM/MM geometries in order to check the validity of the QM/MM partitioning (e.g., with regard to the number of electrons assigned to the QM region) and to compare the computed QM/MM energy profiles with the corresponding pure QM results.

The QM/MM investigations on p-hydroxybenzoate hydroxylase and triosephosphate isomerase employed AM1/GROMOS and AM1/CHARMM, respectively. Pure AM1 data are thus needed for direct comparisons. Conventional SCF calculations are no longer affordable for these systems with 7004 and 8326 atoms, respectively, and therefore the linear scaling SCF/CG-DMS approach was applied for this purpose as implemented. Unless noted otherwise below, standard SCF/CG-DMS options were used (see sections 4.4, 6.1, and 6.4; iscf=4, iplscf=4, maxcg=4, maxpur=2, mpurif=0, cutoff $X=10^{-4}$, block-diagonal initial guess with linfrg=1).

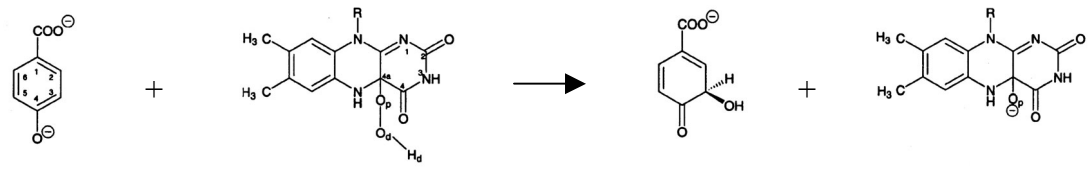
7.2 p-Hydroxybenzoate hydroxylase

p-Hydroxybenzoate hydroxylase (PHBH) is a flavoprotein involved in the degradation of aromatic compounds [100]. It catalyzes the monooxygenation of p-hydroxybenzoate (p-OHB) into 3,4-dihydroxybenzoate (3,4-DOHB). The catalytic cycle consists of reductive and oxidative half-

reactions [101,102]. During the reductive phase, p-OHB and reduced nicotinamide adenine dinucleotide (NADPH) bind to the enzyme, and NADPH reduces the cofactor (flavin adenine dinucleotide, FAD). In the subsequent oxidative phase, the reduced flavin reacts with oxygen to form flavin-4a-hydroperoxide (FADHOOH) which then oxygenates p-OHB. The oxygen transfer to p-OHB yields flavin-4a-hydroxide (FADHOH) and an intermediate which tautomerizes to the product 3,4-DOHB. The catalytic cycle is then completed by the release of the product and the elimination of water from FADHOH to regenerate the oxidized enzyme.

The oxygen transfer from FADHOOH to p-OHB is the rate-determining step in the cycle. The accepted mechanism is electrophilic substitution at the aromatic ring of the substrate, with heterolytic cleavage of the peroxide bond [100,103], although alternative pathways have been considered in the literature (see [98] for references). It is generally believed that p-OHB reacts as a deprotonated dianion during hydroxylation since the reaction is much slower if the deprotonation is suppressed [101]. We have therefore studied the reaction between the p-OHB dianion and FADHOOH (see Figure 13).

Figure 13: PHBH-catalyzed monooxygenation reaction between the p-OHB dianion and FADHOOH (R = ribityl side chain).



The previous AM1/GROMOS calculations [98] employed a QM region with 102 atoms (p-OHB + FADHOOH) and an MM region with 6902 atoms (protein environment + crystal water). The QM region was assigned a total charge of -4 arising from the two extra electrons on the p-OHB substrate and the doubly charged phosphate group in the ribityl side chain of FADHOOH. Charge transfer between the QM and MM regions is ruled out by definition in the standard QM/MM treatments. It is thus desirable to check whether this total charge of -4 is actually found in pure QM calculations that allow for such charge transfer. Table 21 lists the corresponding results from single-point AM1 calculations.

Starting with the data for the isolated QM region (102 atoms) it is obvious that conventional SCF calculations yield essentially the same charge distributions as SCF/CG-DMS calculations with standard cutoffs, see (c) vs (d) in Table 21 (maximum differences of 0.001 e in reactant and

product, and 0.004 e in the transition state TS). Addition of the external MM point charges leads to relatively minor modifications in the subsystem charges, see (c) vs (e) (up to 0.010 e). Performing an SCF/CG-DMS calculation for the full system (7004 atoms) causes somewhat larger changes, see (e) vs (f), which are however not excessive (up to 0.042 e). The total charges of the QM region remain close to -4 in the full SCF/CG-DMS treatment (f) and vary only slightly during the reaction (from -3.958 to -3.972 e). This offers a convincing justification (a posteriori) for the chosen QM/MM approach [98].

The subsystem charges in Table 21 clearly show that the reaction under study is indeed an electrophilic substitution. Going from reactant to product, the substrate p-OHB loses a charge of about 0.94 e while the cofactor FADHOOH gains about the same amount of charge. At the transition state, there is an intermediate charge transfer of about 0.28 e (more reactant-like). Both the AM1/GROMOS and the pure AM1 calculation yield essentially the same electronic characterization of the reaction, see (e) vs (f).

Table 21: Charge distributions (au) from AM1 calculations (see text).

System (a)	Subsystem (b)	Sum of atomic charges (c)	(d)	(e)	(f)
Reactant	p-OHB	-1.952	-1.952	-1.957	-1.931
	FADHOOH	-2.048	-2.048	-2.043	-2.027
	Op	-0.174	-0.173	-0.166	-0.163
	Od-Hd	0.085	0.085	0.084	0.084
	QM region	-4.000	-4.000	-4.000	-3.958
TS	p-OHB	-1.673	-1.669	-1.681	-1.639
	FADHOOH	-2.327	-2.331	-2.319	-2.323
	Op	-0.361	-0.362	-0.352	-0.355
	Od-Hd	0.080	0.079	0.082	0.079
	QM region	-4.000	-4.000	-4.000	-3.963
Product	3,4-DOHB	-1.012	-1.012	-1.011	-1.003
	FADHO(-)	-2.988	-2.988	-2.990	-2.969
	Op	-0.636	-0.635	-0.626	-0.615
	Od-Hd	-0.079	-0.079	-0.077	-0.079
	QM region	-4.000	-4.000	-4.000	-3.972

(a) Geometries taken from optimizations of the complete system (7004 atoms) at the AM1/GROMOS level.

(b) p-OHB substrate (p-oxybenzoate dianion); FADHOOH cofactor; Op proximal oxygen atom of OOH group; Od-Hd distal oxygen atom plus bound H atom of OOH group; QM region consisting of substrate plus cofactor (102 atoms).

(c) Net subsystem charges, AM1 calculation of QM region (102 atoms), conventional SCF.

(d) Net subsystem charges, AM1 calculation of QM region (102 atoms), SCF/CG-DMS with cutoff.

(e) Net subsystem charges, AM1 calculation of QM region (102 atoms) with external MM charges (GROMOS) included, conventional SCF.

(f) Net subsystem charges, AM1 calculation of the complete system (7004 atoms), SCF/CG-DMS with cutoff.

Table 22 compares the barriers obtained for the monooxygenation reaction (Figure 13). The experimental ΔG value [104] corresponds most closely to the theoretical free-energy barrier (ΔF) that has been determined by molecular dynamics simulations with thermodynamic integration at the AM1/GROMOS level [98]. The excellent agreement between these two values (11.7 vs 11.8 kcal/mol) must be considered fortuitous. The energy barrier (ΔE) of 21.3 kcal/mol from AM1/GROMOS geometry optimizations can be compared directly with the single-point AM1 value of 15.0 kcal/mol obtained with standard cutoffs; tightening the cutoffs from $X=10^{-4}$ to $X=10^{-5}$ leaves the barrier essentially unchanged (15.1 kcal/mol). In an overall assessment, the pure AM1 value for the barrier is of the same order as the AM1/GROMOS values and thus confirms the previous AM1/GROMOS results. Given the limited accuracy of the applied methods and inherent technical limitations [98,99] we believe that we cannot prefer one theoretical value over the other and that it is more appropriate to stress the internal consistency of the AM1/GROMOS and pure AM1 results for PHBH.

Table 22: Barriers (kcal/mol) for the PHBH-catalyzed reaction (see Figure 13).

Method	Quantity(a)	Barrier	Reference
Experiment	ΔG	11.7	104
AM1/GROMOS	ΔF, MD	11.8	98
AM1/GROMOS	ΔE, opt	21.3	98
AM1	ΔE, s-p	15.0	b
AM1	ΔE, s-p	15.1	c

(a) MD molecular dynamics, opt geometry optimization, s-p single-point calculations (see text).

(b) SCF/CG-DMS with cutoff $X=10^{-4}$; heats of formation for reactant, TS, and product: -31139.74, -31124.73, and -31206.81 kcal/mol.

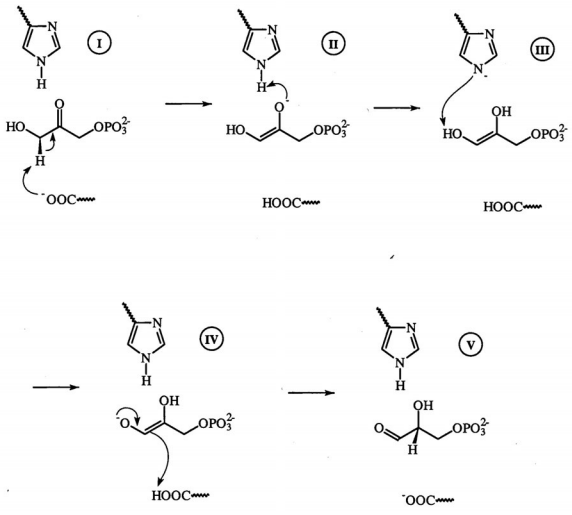
(c) SCF/CG-DMS with cutoff $X=10^{-5}$.

7.3 Triosephosphate isomerase

Triosephosphate isomerase (TIM) catalyzes the conversion of dihydroxyacetone phosphate (DHAP) to glycerinaldehyde-3-phosphate (GAP). This conversion is a key step on the glycolytic pathway [97]. It is accelerated in TIM by a factor of 10^{10} compared with the rate in aqueous solution under acetate ion catalysis [97]. The corresponding reactions have been the subject of several reviews and many detailed experimental and theoretical studies (see the references cited in [99]). It is generally accepted that the conversion proceeds by a proton shuttle mechanism [97] although there is still some debate over the detailed nature of these proton transfers [105,106].

The most widely accepted mechanism [97,107] is shown in Figure 14. The first step involves the abstraction of the pro-R proton of DHAP (I) by the Glu 165 side chain carboxylic group to form an enediolate intermediate (II). A subsequent proton transfer from the pyrrolic nitrogen of the imidazole ring in the side chain of the His 95 residue then gives an enediol intermediate (III). The imidazolate can be reprotonated by the terminal hydroxyl group of DHAP which produces another enediolate intermediate (IV) with the oxy and hydroxy positions interchanged compared to (II). In the final step, the enediolate (IV) is reprotonated by the glutamic acid side chain to form the product GAP (V).

Figure 14: TIM-catalyzed conversion of DHAP to GAP (see text).



The reactions depicted in Figure 14 have been investigated through QM/MM calculations at different levels [99,105-107] addressing both mechanistic and methodological issues. In the work from our group [99], the main objective has been to study the sensitivity of the QM/MM results towards variations in the QM/MM model, including the choice of the QM method (semiempirical

vs. density functional vs. ab initio methods, i.e., AM1 vs. BP86 / B3LYP vs. MP2), the size of the QM region, and the treatment of the QM/MM boundary. In the following, we extend this methodological work by comparing the previous AM1/CHARMM results [99] with pure AM1 results from single-point SCF/CG-DMS calculations at the available optimized AM1/CHARMM geometries. These comparisons will again focus on charge distributions and relative energies.

The reference AM1/CHARMM calculations [99] employed a QM region with 37 atoms including the substrate, the His 95 side chain, the Glu 165 side chain, and two hydrogen link atoms to saturate the dangling bonds at each side chain (i.e., for the reactant I: DHAP, 2-methyl imidazole, and propionate). Due to the successive proton transfers, the number of atoms in the components of the QM region and the corresponding formal charges do not remain constant in the intermediates I-V. These data are collected in Table 23 for easy reference.

Table 23: Number of atoms (N) and formal charges (Z) for QM region (a): Intermediates I-V (see Figure 14 and text).

Label	Intermediate	Substrate		His 95		Glu 165		Sum	
		N	Z	N	Z	N	Z	N	Z
I	educt	15	-2	11	0	9	-1	35	-3
II	enediolate	14	-3	11	0	10	0	35	-3
III	enediol	15	-2	12	-1	10	0	35	-3
IV	enediolate	14	-3	11	0	10	0	35	-3
V	product	15	-2	11	0	9	-1	35	-3

(a) Note that the two link atoms are not counted. Including these link atoms will increase N by 1 for His 95 and Glu 165, and by 2 for the sum.

For each of the minima and transition states in the reaction scheme (see Figure 14), charge distributions have been determined from two separate single-point AM1 calculations. The first one was a conventional SCF calculation for the QM region plus the two link atoms (for a total of 37 atoms) whereas the second one used an SCF/CG-DMS treatment of the full enzyme (8326 atoms). In both cases, the Cartesian input coordinates were taken from the available optimized AM1/CHARMM geometries for the full enzyme. In the calculations for the QM region, the two link atoms were put into the corresponding C-C frontier bonds being cut, at a fixed C-H distance of 1.1 Å.

The resulting charge distributions are given in Tables 24 and 25, respectively. Since the link atoms are only present in the model calculations for the QM region, but not in the calculations for the enzyme, our evaluation will cover only the 35 atoms from the QM region that are included in both cases. The link atoms carry a small positive charge, and therefore, in the first case, the sum of the net atomic charges deviates somewhat from the formal charge of -3, by 0.08-0.10 e (Table 24). In the second case, this deviation is much larger and amounts to 0.34-0.37 e (Table 25). Evidently the protein environment withdraws considerable electron density from the QM region, at least at the AM1 level, which is not reflected by the model calculations for the isolated QM region. Adding external MM point charges will not remedy this situation because all QM electrons are constrained to stay within the QM region in the standard QM/MM approach, by definition, so that the external MM point charges can only lead to polarization within the QM region (and not to charge transfer outside this region).

While the large amount of charge transfer to the protein environment in the pure AM1 calculations may seem alarming at first sight, it should also be pointed out that this effect appears to be fairly uniform for all nine species studied. Taking the differences between the data in Tables 24 and 25 as a measure, the total charge transfer out of the QM region ranges from 0.42-0.47 e (mostly around 0.43 e). The charge transfers out of the individual subsystems vary somewhat more, but still tend to be rather similar (substrate 0.08-0.20 e, often around 0.08-0.12, higher for I and II; His 0.14-0.24 e, at the lower end for I and TS I->II; Glu 0.09-0.14 e). As long as the overall charge transfers remain uniform, there is a chance that the differences between the QM/MM and the pure QM calculations may also be uniform which can then lead to error compensation (e.g., for relative energies).

Finally it should be noted that the computed charges for the subsystems (Tables 24 and 25) resemble the corresponding formal charges (Table 23) qualitatively, but may deviate quantitatively. For example, the computed charges for the substrate are indeed reasonably close to -2 for I, III, and V, but they do not reach -3 for II and IV (the deviations from -3 being larger in Table 25 than in Table 24 due to higher charge transfer in the enzyme, see above). Similar remarks apply to the histidine subsystem (see III with a formal charge of -1).

Table 24: Charge distributions (au) from AM1 calculations (see text): Conventional SCF treatment for the QM region (a).

Label	Species	Substrate	His	Glu	Sum
I	educt	-2.01	-0.07	-1.00	-3.08
TS	educt to enediolate	-2.23	-0.08	-0.77	-3.08
II	enediolate	-2.60	-0.10	-0.40	-3.10
TS	enediolate to enediol	-2.25	-0.47	-0.37	-3.09
III	enediol	-1.94	-0.79	-0.35	-3.08
TS	enediol to enediolate	-2.10	-0.62	-0.36	-3.08
IV	enediolate	-2.41	-0.30	-0.39	-3.10
TS	enediolate to product	-2.05	-0.23	-0.80	-3.08
V	product	-1.89	-0.20	-1.00	-3.09

(a) For the definition of the subsystems see Table 23. In the case of transition states, the migrating hydrogen atom is assigned to the subsystem from where it comes.

Table 25: Charge distributions (au) from AM1 calculations (see text): SCF/CG-DMS treatment for the entire enzyme (a).

Label	Species	Substrate	His	Glu	Sum
I	educt	-1.81	0.07	-0.91	-2.65
TS	educt to enediolate	-2.08	0.06	-0.64	-2.66
II	enediolate	-2.41	0.06	-0.28	-2.63
TS	enediolate to enediol	-2.13	-0.26	-0.26	-2.65
III	enediol	-1.83	-0.58	-0.24	-2.65
TS	enediol to enediolate	-2.02	-0.39	-0.24	-2.65
IV	enediolate	-2.33	-0.06	-0.27	-2.66
TS	enediolate to product	-1.99	0.00	-0.66	-2.65
V	product	-1.79	0.03	-0.89	-2.65

(a) See footnote of the Table 24.

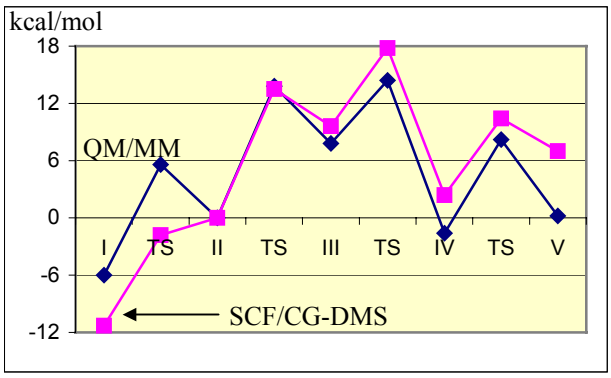
The heats of formation and the corresponding relative energies from the single-point AM1 calculations for the full enzyme are given in Table 26. The relative energies from the AM1/GROMOS calculations [99] are included for comparison. The resulting energy profiles are shown in Figure 15 using intermediate II to define the common origin of the energy scale. The AM1 and AM1/GROMOS energy profiles for the three last proton transfers (II -> III -> IV -> V) are surprisingly similar, with typical deviations on the order of 3 kcal/mol. Larger discrepancies

occur for the first proton transfer (I -> II) where the relative stability of the educt I is more pronounced in AM1 than in AM1/GROMOS. In an overall assessment, the two energy profiles in Figure 15 appear qualitatively similar, in spite of the considerable charge transfer to the protein environment observed in the pure AM1 calculations. This would seem to suggest that the results may indeed benefit from error compensation (see above) which might be less effective for the first proton transfer (I -> II) with less uniform charge transfers (see Tables 24 and 25).

Table 26: AM1 heats of formation ΔH_f (kcal/mol) and relative energies E_{rel} (kcal/mol) from single-point SCF/CG-DMS calculations on the entire enzyme (see text). QM/MM (AM1/GROMOS) relative energies (kcal/mol) are given for comparison.

Label	Species	ΔH_f AM1	E_{rel} AM1	E_{rel} QM/MM
I	educt	-34913.94	-11.3	-6.0
TS	educt to enediolate	-34904.45	-1.8	5.6
II	enediolate	-34902.62	0.0	0.0
TS	enediolate to enediol	-34889.13	13.5	13.8
III	enediol	-34893.06	9.6	7.8
TS	enediol to enediolate	-34884.79	17.8	14.4
IV	enediolate	-34902.01	2.4	-1.6
TS	enediolate to product	-34892.20	10.4	8.2
V	product	-34895.66	7.0	0.2

Figure 15: Computed energy profiles for TIM-catalyzed reactions (see text).



Chapter 8

Conclusions and outlook

Given the need to extend current quantum-chemical treatments to large molecules with thousands of atoms, the main goal of this work was the implementation of linear scaling approaches in the context of semiempirical molecular orbital theory. At the outset of this project, three such approaches had already been suggested in the literature. We have chosen the conjugate gradient density matrix search (CG-DMS) because it employs reliable and well-established minimization procedures and offers a transparent route towards linear scaling through the use of cutoffs in the density matrix and the Fock matrix.

Three versions of the CG-DMS code have been implemented. The full-matrix versions with precomputation (ITERCG) and on-the-fly computation (DIRCG) of the required integrals serve mainly for testing purposes, whereas the sparse-matrix integral-direct version (DIRCGS) is designed for linear scaling production work on large molecules. DIRCGS employs the compressed sparse row format and subroutines for matrix operations from the public-domain SPARSKIT2 library. In systematic test calculations on small molecules without cutoffs, the three versions of the code yield identical results, which are in full agreement with the results from conventional SCF calculations using matrix diagonalization.

When starting from a diagonal initial density matrix (as commonly done in conventional semiempirical calculations), the SCF/CG-DMS approach does not converge reliably. This failure can be traced to the fact that such an initial density matrix is far from idempotent, which may cause the McWeeny idempotency transformation to behave erratically such that the number of electrons is not conserved. To circumvent such problems, an alternative option to generate an initial density matrix for large molecules has been implemented: a block-diagonal guess is assembled from non-converged fragment density matrices which are obtained by performing typically one conventional

SCF iteration for user-defined fragments. This initial guess is idempotent and normalized, by construction, and leads to reliable and robust convergence in all cases considered so far.

The SCF/CG-DMS procedure involves a triply nested loop of iterations (SCF, CG, McWeeny) that are controlled by a large number of parameters. To establish recommended default values for these control parameters, systematic tests have been carried out using three sets of calculations: MNDO for 218 small organic molecules, MNDO/d for 366 small inorganic molecules, and AM1 for 47 large biochemical molecules. Default values for all options have been chosen on this basis. The tests have shown in particular that it is not efficient to converge the inner CG and McWeeny cycles tightly when the outer SCF cycles are still far from convergence: imposing a maximum number of two CG and McWeeny cycles is found to be the best choice.

For validation purposes, the sparse-matrix integral-direct code has been applied to several series of large molecules including polyglycines, water clusters, and proteins. For suitable cutoffs, the computational effort for SCF/CG-DMS calculations is found to scale linearly with molecular size, as expected. The crossover points with conventional SCF calculations occur later than in other published implementations which is probably at least partly due to the use of non-optimized sparse matrix routines from the SPARSKIT2 library. In the case of the AM1 calculations on proteins, rms charge fluctuations of the order of 0.05 e per residue are found which cannot be captured by the usual classical force fields with fixed charges.

The new linear scaling code has been applied to study reactions in the enzymes p-hydroxybenzoate hydroxylase (PHBH) and triosephosphate isomerase (TIM) which had previously been investigated in our group by combined quantum mechanical / molecular mechanical (QM/MM) methods. These single-point AM1 calculations at the available optimized QM/MM geometries would not have been possible with the conventional SCF code due to the size of these systems (7004 and 8326 atoms, respectively). The AM1 relative energies and activation barriers are consistent with the previous AM1/GROMOS and AM1/CHARMM results. An analysis of the charge distributions justifies the chosen QM/MM approach for PHBH since the total charge of the QM region remains close to the formal value of -4 e throughout the reaction. In the case of TIM, the AM1 charges from the full enzyme calculations indicate a considerable charge transfer from the QM region to the protein environment (about 0.4 e) which is however rather uniform for all species involved so that the QM/MM results may benefit from error compensation.

As a result of this thesis, a validated and working semiempirical SCF/CG-DMS code is available that exhibits linear scaling in the computational effort for large molecules. Further optimization and

tuning of this code seems possible, in particular with regard to the sparse matrix library routines (especially for sparse matrix multiplication). In its present form, the code can already be used in studies on enzymes to complement corresponding QM/MM work.

Appendix 1 Subroutines for conjugate gradient density matrix search: test version with full two-dimensional matrices.

All matrices are handled as two-dimensional arrays to allow for full calculations without neglecting any matrix elements (for comparison with calculations employing matrix diagonalization). There is an option to neglect matrix elements below user-defined cutoffs (for comparison with the sparse-matrix version of the code).

The one-electron and two-electron integrals can either be precomputed and used as stored (ITERCG) or calculated on-the-fly as needed in an integral-direct manner (DIRCG). The latter approach forms the basis for the sparse-matrix version (see appendix 2).

Routine	Brief description
ITERCG	Control routine for SCF iterations using the conjugate gradient density matrix search (CG-DMS), conventional version with precomputed one- and two-electron integrals.
DIRCG	Control routine for SCF iterations using the conjugate gradient density matrix search (CG-DMS), integral-direct version with on-the-fly calculation of the integrals.
CGDMS	Control routine for conjugate gradient density matrix search. Standard Fortran code using full square matrices.
DMSGRD	Compute gradient of the functional used in CG-DMS.
PRECO	Generate diagonal preconditioner for CG-DMS.
DMSCOF	Compute coefficients for analytic line search.
CGROOT	Select proper root in line search for CG update. Choose root with lower value of the functional.
PURIFY	McWeeny purification transformation: **R** = 3 **PP** - 2 **PPP**.
PSCAL	Scale density matrix such that it has the correct trace.
BORDCG	Determine changes in the density matrix for later convergence checks and perform extrapolation or damping (optional).
COMMUT	Find the maximum absolute value of commutator matrix elements of two symmetric matrices: [**B**,**C**] = **BC** - **CB**.

DOTMAT	Compute dot product between two vectors containing the matrix elements of two symmetric matrices that are stored either as full two-dimensional arrays or as upper triangles.
MATCOP	Matrix copy: **B** = **A** or **B** = -**A**.
MATCUT	Set matrix elements below a given cutoff to zero.
MATDEB	Debug print for a two-dimensional matrix array.
MATDEV	Evaluate deviations between two matrices **A** and **B**.
MATDIF	Difference of two matrices: **C** = **A** - **B**.
MATSCL	Scale matrix by a factor.
MATUPD	Perform matrix update: **C** = **A** + FACTOR***B**.
MATUPM	Perform matrix update: **C** = **A** + FACTOR***B**, for diagonal elements only, and return minimum and maximum diagonal elements.
TRACE1	Compute trace of matrix **A**.
TRACE2	Compute trace of matrix product **AB**.
TRMSUM	Sum of a square matrix and its transpose: **B** = **A** + $\mathbf{A}^T$.

There are about 2100 lines of code in these new routines.

Appendix 2 Subroutines for conjugate gradient density matrix search: linear scaling version with sparse matrices.

All sparse matrices are handled in the CSR format. All one-electron and two-electron integrals are computed on-the-fly as needed. The sparse-matrix code is derived from the integral-direct test version (see appendix 1).

The new sparse-matrix routines can be grouped as follows:

(a) Routines that are essentially equivalent to corresponding CG-DMS routines in the test version (appendix 1).
(b) Routines that are introduced for convenience in connection with sparse-matrix operations.
(c) Routines for integral-direct calculations and for postprocessing that are essentially equivalent to existing standard routines.
(d) Modified versions of routines from the SPARSKIT2 library to allow for the use of Fortran90 data types.

Routine	Brief description
(a)	Sparse-matrix CG-DMS routines.
DIRCGS	Control routine for SCF iterations using the conjugate gradient density matrix search (CG-DMS), integral-direct version with on-the-fly calculation of the integrals.
CGDMSS	Control routine for conjugate gradient density matrix search.
DMSGRDS	Compute gradient of the functional used in CG-DMS.
DIAPRC	Generate diagonal preconditioner for CG-DMS.
DMSCOFS	Compute coefficients for analytic line search.
CGROOTS	Select proper root in line search for CG update. Choose root with lower value of the functional.
PURIFYN	McWeeny purification transformation: **R** = 3 **PP** - 2 **PPP**. Input matrix **P** overwritten by output matrix **R**.

PURIFYP McWeeny purification transformation: **R** = 3 **PP** - 2 **PPP**. Input matrix **P** kept, separate output matrix **R**.

BORDS1 Determine changes in the density matrix for later convergence checks and perform extrapolation or damping (optional).

DDOTS1 Compute dot product **A**·**A** for a sparse symmetric matrix.

DDOTS2 Compute dot product **A**·**B** for two sparse symmetric matrices.

MATDEVS Evaluate deviations between two sparse matrices **A** and **B**.

SPAUPM Perform sparse matrix update: **C** = **A** + FACTOR***B**, for diagonal elements only, and return minimum and maximum diagonal elements.

(b) Sparse-matrix utility routines.

CGUPD Update of density matrix after CG line search (combining code from DIRCG and MATUPD, see appendix 1).

IDAMAXP Find the index of an element in a pointer array with the largest maximum value (equivalent to the BLAS1 routine IDAMAX).

SPAPRT Print a sparse matrix.

UTIL Count non-zero entries per column in a sparse matrix.

XERALL Error handling for ALLOCATE and DEALLOCATE.

XERSPA Error handling for sparse-matrix operations.

module1 Fortran90 interface definition, first set of routines.

module2 Fortran90 interface definition, second set of routines.

module3 Fortran90 interface definition, third set of routines.

(c) Sparse-matrix versions of routines from the standard program.

HFOCKS Calculate a block of the core Hamiltonian and Fock matrix and evaluate the corresponding energy contributions in an integral-direct fashion.

PONES Extract one-center density matrix elements from the sparse density matrix.

PSORTS Extract two-center density matrix elements for a given atom pair from the sparse density matrix.

DCARTS Evaluate the Cartesian gradient using the sparse density matrix.

DIPOLS Evaluate the dipole moment using the sparse density matrix.

PRTSCFS Print SCF results available in sparse-matrix format.

SPARSP Generate initial diagonal density matrix.

(d) Modified versions of routines from the SPARSKIT2 library.

amubdgp	Perform symbolic multiplication and count the number of nonzero elements in the product.
amubp	Perform actual multiplication, **AB**.
aplbdgp	Perform symbolic summation and count the number of nonzero elements in the sum.
aplbp	Perform actual summation, **A**+**B**.
coicsrp	Convert sparse matrix from coordinate to CSR format.
copmatp	Copy matrix.
csrcscp	Convert sparse matrix from CSR to CSC (transpose) format.
filterp	Perform numerical thresholding on matrix elements.
traceap	Calculate trace of matrix **A**.
tracenp	Calculate trace of matrix product **AB** without computing the product itself.
transpp	Transpose matrix.

There are about 5300 lines of code in these new routines.

Appendix 3 Subroutines for generating an initial block-diagonal density matrix from fragment RHF-SCF calculations.

The fragment RHF-SCF calculations make use of many existing standard routines. New code is required for setting up and controlling these calculations, and for assembling a suitable molecular density matrix. The latter can be generated either as a standard two-dimensional array (for use in ITERCG and DIRCG) or as a sparse matrix in CSR format (for use in DIRCGS).

Routine	Brief description
FRAGMT	Control routine for fragment RHF-SCF calculations.
INFRG	Read data to define the fragments.
DEFFRG	Extract fragment data from the available molecular data.
INPUTS	Define control variables from fragment input data.
DYNFRG	Dynamic memory allocation for fragment RHF-SCF calculation.
SCFFRG	Perform RHF-SCF calculation for a given fragment.
ITFRG	RHF-SCF iterations for a given fragment.
COPFRG	Copy fragment density into molecular density array.
FILFRG	Fill sparse density matrix with fragment density.
SRTFRG	Sort sparse density matrix by reordering columns.

There are about 1300 lines of code in these new routines.

Appendix 4 Input options for linear scaling SCF-MO calculations.

This appendix documents those input options that are particularly relevant for linear scaling SCF-MO calculations. The MNDO99 input is generally keyword-oriented, but it also supports fixed-format input. The first table specifies the keywords for the input options, the unique internal number of each option, the Fortran format, and a short description. The second table gives a full description for each option.

Table A.4.1: Overview over available options.

Option	No.	Format	Short description
intdir	55	i2	Integral direct SCF procedure.
lindms	56	i2	Linear scaling CG-DMS approach.
lindia	57	i2	Choice of full diagonalization after CG-DMS.
linfrg	58	i2	Initial density from fragment calculations.
inpfrg	59	i2	Read extra input for fragments.
inp24	63	i2	Read extra input for options (171-186).
maxcg	171	i5	Maximum number of CG cycles during DMS.
maxpur	172	i5	Maximum number of McWeeny purifications.
mcmax	173	i5	Convergence criterion for purification (P).
midemp	174	i5	Convergence criterion for purification (PP).
mpurif	175	i5	CG cycle where purification starts.
mlroot	176	i5	Choice of root for the CG density update.
mcgpre	177	i5	Preconditioning of CG gradient matrix.
mcgupd	178	i5	Choice of update for search direction.
mpscal	180	i5	Scaling of intermediate density matrices.
mcuth	181	i5	Cutoff for core Hamiltonian matrix (eV).
mcutf	182	i5	Cutoff for Fock matrix (eV).
mcutp	183	i5	Cutoff for density matrix.
mcut1	184	i5	Cutoff for one-electron integrals (eV).
mcut2	185	i5	Cutoff for two-electron integrals (eV).

mcutr	186	i5	Cutoff for interatomic distances (A).

Table A.4.2: Specification of the available options.

Option	Full description
intdir	Choice of integral-direct SCF procedure. = 0 Conventional integral handling. = 1 Direct approach, without thresholds. = 2 Direct approach, default thresholds. = 3 Direct approach, special thresholds from input. = 4 Sparse approach, default thresholds. = 5 Sparse approach, special thresholds from input.
lindms	Choice of linear scaling CG-DMS approach: conjugate gradient density matrix search. = 0 Conventional SCF treatment. = 1 CG-DMS approach, without thresholds. = 2 CG-DMS approach, default thresholds. = 3 CG-DMS approach, special thresholds from input. = 4 Sparse approach, default thresholds. = 5 Sparse approach, special thresholds from input.
lindia	One conventional diagonalization after CG-DMS convergence to obtain MO eigenvalues and eigenvectors. = 0 No such diagonalization. = 1 Allow such diagonalization. *** Note that this will not work for really large molecules. *** Useful for any postprocessing that requires MOs.
linfrg	Block-diagonal initial density matrix from separate RHF-SCF calculations on user-defined fragments using the same convergence criteria as in the molecular case. = 0 Do not build such an initial density matrix. = n Number of SCF iterations allowed for each fragment. Recommended values are n=1 and n=2 to obtain a sufficiently accurate initial guess for CG-DMS. Normally (unless inpfrg=-1) the definition of the fragments requires some extra input (section 3.14).
inpfrg	Input to define the fragments for CG-DMS in section 3.14.

=-1 Do not read such input, treat the whole molecule as a single fragment (only useful for testing purposes).

= 0 Read such extra input using the default format.

= n Read such extra input using other formats.

inp24 One line of extra input for options 171-186. Special options for linear scaling and direct SCF methods.

= 0 Do not read such extra input, use default options.

= 1 Read such extra input.

*** Set internally for intdir=3,5 or lindms=3,5 (see above).

maxcg Maximum number of conjugate gradient (CG) cycles during density matrix search. Default 2.

maxpur Maximum number of McWeeny purifications during one CG cycle. Default 2.

mcmax Convergence criterion for purification. Maximum allowed change of diagonal density matrix elements: pmcmax=10**(-mcmax). Default for mcmax.le.0: Ignore criterion, use pmcmax=pcgmax/2 where pcgmax is the corresponding global criterion.

midemp Convergence criterion for purification. Maximum allowed violation of idempotency for diagonal density matrix elements: pidemp=10**(-midemp). Default for midemp.le.0: pidemp=1, i.e. the criterion is effectively ignored.

mpurif CG cycle where purification starts.

= n Purification starting at CG cycle n.

= 0 Use default value of mpurif=99.

=-1 Purification turned off, activated automatically only when the CG search approaches convergence (as measured by the magnitude of the CG update).

*** For mpurif.ge.0, a purified density matrix is always used, available either from a single transformation (before CG cycle n) or from repeated transformations (thereafter).

*** For mpurif.gt.maxcg, purification will be turned on in the last CG cycle (maxcg) even if CG convergence is not reached.

*** For mpurif=-1, the linear CG update for the density matrix is used as long as the purification has not been activated. Option mpurif=-1 is NOT recommended.

mlroot Choice of root for the CG density update.

The step size for the CG density update is given by the root of a quadratic equation. The root-finding algorithm is as follows:

(a) Check whether linear term dominates such that the solution of the linear equation can be adopted.

(b) Reject any physically unacceptable root of the quadratic equation with any diagonal density matrix element below 0 or above 1.

(c) Compute the functional value for both roots of the quadratic equation and adopt the lower root.

Option mlroot controls the first step.

= n Check (a) is done, and the solution of the linear equation is adopted if the absolute value of x in the term $\sqrt{1+x}$ of the quadratic equation is smaller than the threshold: xsqmax = 10**(-mlroot). Errors will then be of the order x**2.

= 0 Use default value of mlroot=5.

=-1 Check (a) is not done. Step size determined from (b)-(c).

mcgpre Preconditioning of CG gradient matrix.

= 0 Not used.

= 1 Diagonal preconditioning applied.

mcgupd Choice of update for search direction.

= 0 Polak-Ribiere formula for CG.

= 1 Fletcher-Reeves formula for CG.

mpscal Scaling of intermediate density matrices to enforce normalization which may be lost due to purification.

= 0 No such scaling.

= 1 Restore correct trace of the density matrix after each CG search by adding a constant to each diagonal element.

= 2 Restore correct trace of the density matrix after each CG search by scaling each diagonal element.

= 3 Analogous to mpscale=2, but apply the scaling to the complete matrix.

mcuth Cutoff for core Hamiltonian matrix (eV).

=-1 No such cutoff.

= 0 Use default value of mcuth=20.

= n Cutoff 10**(-n) eV.

mcutf Cutoff for Fock matrix (eV).

=-1 No such cutoff.

= 0 Use default value of mcutf=20.

= n Cutoff 10**(-n) eV.

mcutp Cutoff for density matrix.

	=-1	No such cutoff.
	= 0	Use default value of mcutp=20.
	= n	Cutoff 10**(-n).
mcut1	Cutoff for one-electron integrals (eV).	
	=-1	No such cutoff.
	= 0	Use default value of mcut1=20.
	= n	Cutoff 10**(-n) eV.
mcut2	Cutoff for two-electron integrals (eV).	
	=-1	No such cutoff.
	= 0	Use default value of mcut2=20.
	= n	Cutoff 10**(-n) eV.
mcutr	Cutoff for interatomic distances (Angstrom).	
	=-1	No such cutoff.
	= 0	Use default value of mcutr=10000.
	= n	Cutoff of 0.1*n Angstrom.
		Two-center integrals are not computed if the corresponding distance exceeds the cutoff.

Explicit definition of fragments for block-diagonal density matrix guess.

Note: Only the general default input is described here (inpfrg=0).

Note: Other input formats are available for special cases (inpfrg>0).

- First and following lines (to assign atoms to fragments)

nfrags(i)	10i5	Number of fragment containing atom i. Use as many lines as necessary.

- Subsequent lines (to assign nonzero charges)

i	i5	Number of atom bearing a formal charge.
		= 0 End of this section of input.
ndum	i5	Formal charge of atom i, nchrgs(i)=ndum.
		The array nchrgs is initialized to zero. Only nonzero values are needed from input. Fragment charges are computed from nchrgs(i).

Appendix 5 List of biomolecules studied.

The biomolecules studied in section 6.4 were identified only by their PDB entry or another four-character symbol. The corresponding names are given in the following.

PDB	Name
1B32	Oligo-peptide binding protein (OPPA) complexed with KMK.
2QWE	A complex of 4-Guanidino-NEU5AC2EN and A drug resistant variant R292K of TERN N9 influenza virus neuraminidase.
1IXH	Phosphate-binding protein (PBP) complexed with phosphate.
subt	Subtilisim E. Wild type.
1IAU	Human Granzyme B in complex with AC-IEPD-CHO.
1E3B	Cyclophilin 3 from *C. Elegans* complexed with AUP(ET)3.
lyso	Lysozyme.
1EV3	Rhombohedral form of the m-cresol/Insulin.
1A3K	Human Galectin-3 carbohydrate recognition domain (CRD).
1CC8	ATX1 Metallochaperone protein.
ribo	Ribose.
1AOY	N-terminal domain of *Escherichia Coli* arginine repressor.
1IKL	Monomeric human Interleukin-8.
1A43	HIV-1 capsid protein dimerization domain.
1AFJ	Mercury-bound form of MERP, the periplasmic protein from the bacterial mercury detoxification system.
1IRN	Rubredoxin (Zn-substituted).
cspa	CspA, the major cold shock protein of *Escherichia Coli*.
2R63	Buried salt bridge in the 434 repressor DNA-binding domain.
5EBX	Erabutoxin.

2OVO	Third domain of silver pheasant Ovomucoid (OMSVP3).
bpti	Bovine pancreatic trypsin inhibitor.
1A4T	Phage P22 N peptide-box B RNA complex.
2REL	R-Elafin, A specific inhibitor of Elastase.
1A7F	Insulin mutant B16 Glu, B24 Gly, Des-B30.
2SH1	Neurotoxin I from the sea anemone *Stichodactyla Helianthus*.
1B8W	Defensin-like peptide 1.
cram	Crambin.
1ATO	Isolated, Central hairpin of the Hdv antigenomic ribozyme.
1AML	The Alzheimer`s disease amyloid A4 peptide (residues 1-40).
1OKA	RNA/DNA chimera.
2PTA	Pandinus toxin K-A (Pitx-Ka) from *Pandinus Imperator.*
1AXH	Atracotoxin-Hvi from *Hadronyche Versuta.*
1BH0	Glucagon analog.
1ATF	Transactivation domain of Cre-Bp1/Atf-2.
1BXJ	M8L mutant of squash trypsin inhibitor Cmti-I.
toxi	Toxin.
1AFX	Ugaa eukaryotic ribosomal RNA tetraloop.
1BQF	Growth-blocking peptide (Gbp) from *Pseudaletia Separata.*
2MAG	Magainin 2 in Dpc micelles.
8TFV	Insect defense peptide.
2NR1	Transmembrane segment 2 of Nmda receptor Nr1.
1ALE	Conformation of two peptides corresponding to human apolipoprotein C-I residues 7-24 and 35-53 in the presence of sodium dodecyl sulfate.
1BCV	Synthetic peptide corresponding to the major immunogen site of Fmd virus.
1A3J	Collagen-like peptide with the repeating sequence (Pro-Pro-Gly).
3CMH	Synthetic linear truncated Endothelin-1 agonist.
1CB3	Non-native structure in the denatured state of human α-Lactalbumin.

2SOC Backbone conformational equilibria of Sandostatin.

References

1 A. Warshel and M. Levitt, J. Mol. Biol. 103, 227-249 (1976).
Theoretical Studies of Enzymic Reactions: Dielectric, Electrostatic, and Steric Stabilization of the Carbonium Ion in the Reaction of Lysozyme.

2 M. J. Field, P. A. Bash and M. Karplus, J. Comp. Chem. 11, 700-733 (1990).
A Combined Quantum Mechanical and Molecular Mechanical Potential for Molecular Dynamics.

3 D. Bakowies and W. Thiel, J. Phys. Chem. 100, 10580-10594 (1996).
Hybrid Models for Combined Quantum Mechanical and Molecular Mechanical Approaches.

4 J. Gao, in: Reviews in Computational Chemistry (eds. K. B. Lipkowitz and D. B. Boyd), VCH Publishers, New York, 1996, vol. 7, pp. 119-185.
Methods and Applications of Combined Quantum Mechanical and Molecular Mechanical Approaches.

5 G. Monard and K. M. Merz, Acc. Chem. Res. 32, 904-911 (1998).
Combined Quantum Mechanical/Molecular Mechanical Methodologies Applied to Biomolecular Systems.

6 *Encyclopedia of Computational Chemistry* (eds. P. v. R. Schleyer, N. L. Allinger, T. Clark, P. A. Kollman, H. F. Schaefer, and P. R. Schreiner), Wiley, Chicester, 1998.
(a) P. Amara and M. J. Field, vol. 1, pp. 431-437.
Combined Quantum Mechanical and Molecular Mechanical Potentials.
(b) M. F. Ruiz-Lopez and J.-L. Rivail, vol. 1, pp. 437-448.
Combined Quantum Mechanics and Molecular Mechanics Approaches to Chemical and Biochemical Reactivity.
(c) R. D. J. Froese and K. Morokuma, vol. 2, pp. 1244-1257.
Hybrid Methods.

(d) J. Gao, vol. 2, pp. 1257-1263.
Hybrid Quantum Mechanical/Molecular Mechanical (QM/MM) Methods.
(e) K. M. Merz and R. V. Stanton, vol. 4, pp. 2330-2343.
Quantum Mechanical/Molecular Mechanical (QM/MM) Coupled Potentials.
(f) J. Tomasi and C. S. Pomelli, vol. 4, pp. 2343-2350.
Quantum Mechanics/Molecular Mechanics (QM/MM).

7 P. Sherwood, in: *Modern Methods and Algorithms of Quantum Chemistry* (ed. J. Grotendorst), NIC Series, Jülich, 2000, vol. 3, pp. 285-305.
Hybrid Quantum Mechanics/Molecular Mechanics Approaches.

8 W. Kohn, Phys. Rev. Lett. 76, 3168-3171 (1996).
Density Functional and Density Matrix Method Scaling Linearly with the Number of Atoms.

9 G. E. Scuseria, J. Phys. Chem. A 103, 4782-4790 (1999).
Linear Scaling Density Functional Calculations with Gaussian Orbitals.

10 S. Goedecker, Rev. Mod. Phys. 71, 1085-1123 (1999).
Linear Scaling Electronic Structure Methods.

11 *Encyclopedia of Computational Chemistry* (eds. P. v. R. Schleyer, N. L. Allinger, T. Clark, P. A. Kollman, H. F. Schaefer, and P. R. Schreiner), Wiley, Chicester, 1998.
(a) S. L. Dixon and K. M. Merz, vol. 1, pp. 761-776.
Divide and Conquer for Semiempirical MO Methods.
(b) W. Yang and J. M. Perez-Jorda, vol. 2, pp. 1496-1513.
Linear Scaling Methods for Electronic Structure Calculations.

12 P. Ordejon, Comp. Mat. Sci. 12, 157-191 (1998).
Order-N Tight-binding Methods for Electronic Structure and Molecular Dynamics.

13 G. Galli, Phys. Stat. Sol. B 217, 231-249 (2000).
Large-scale Electronic Structure Calculations Using Linear Scaling Methods.

14 W. Yang, Phys. Rev. Lett. 66, 1438-1441 (1991).
Direct Calculation of Electron Density in Density Functional Theory.

15 S. L. Dixon and K. M. Merz, J. Chem. Phys. 104, 6643-6649 (1996).
Semiempirical molecular orbital calculations with linear system size scaling.

16 T.-S. Lee, D. M. York and W. Yang, J. Chem. Phys. 105, 2744-2750 (1996).
Linear-scaling semiempirical quantum calculations for macromolecules.

17 J. J. P. Stewart, Int. J. Quantum Chem. 58, 133-146 (1996).
Application of Localized Molecular Orbitals to the Solution of Semiempirical Self-Consistent Field Equations.

18 X. P. Li, W. Nunes and D. Vanderbilt, Phys. Rev. A 47, 10891-10894 (1993).
Density-matrix electronic-structure method with linear system-size scaling.

19 A. D. Daniels, J. M. Millam and G. E. Scuseria, J. Chem. Phys. 107, 425-431 (1997).
Semiempirical methods with conjugate gradient density matrix search to replace diagonalization for molecular systems with thousands of atoms.

20 W. Thiel, Program MNDO99, Max-Planck-Institut für Kohlenforschung, Mülheim, 1999.

21 W. Thiel, in: *Modern Methods and Algorithms of Quantum Chemistry* (ed. J. Grotendorst), NIC Series, Jülich, 2000, vol. 3, pp. 261-283.
Semiempirical Methods.

22 W. Weber and W. Thiel, Theor. Chem. Acc. 103, 495-506 (2000).
Orthogonalization corrections for semiempirical methods.

23 M. Scholten, Dissertation, Universität Düsseldorf, 2003.
Semiempirische Verfahren mit Orthogonalisierungskorrekturen: Die OM3 Methode.

24 M. J. S. Dewar and W. Thiel, J. Am. Chem. Soc. 99, 4899-4907 (1977).
Ground States of Molecules. 38. The MNDO Method. Approximations and Parameters.

25 M. J. S. Dewar, E. Zoebisch, E. F. Healy and J. J. P. Stewart, J. Am. Chem. Soc. 107, 3902-3909 (1985).
AM1: A New General Purpose Quantum Mechanical Molecular Model.

26 J. J. P. Stewart, J. Comp. Chem. 10, 209-220 (1989).
Optimization of Parameters for Semiempirical Methods. I. Method.

27 J. J. P. Stewart, J. Comp. Chem. 10, 221-264 (1989).
Optimization of Parameters for Semiempirical Methods. II. Applications.

28 M. Kolb and W. Thiel, J. Comp. Chem. 14, 775-789 (1993).
Beyond the MNDO Model: Methodical Considerations and Numerical Results.

29 S. L. Dixon and K. M. Merz, J. Chem. Phys. 107, 879-893 (1997).
Fast accurate semiempirical molecular orbital calculations for macromolecules.

30 F. Jensen, *Introduction to Computational Chemistry*, Wiley, Chicester, 1999, pp. 71-75.

31 C. Kollmar, Int. J. Quantum Chem. 62, 617-637 (1997).
Convergence Optimization of Restricted Open-Shell Self-Consistent Field Calculations.

32 M. F. Guest and V. R. Saunders, Mol. Phys. 28, 819-828 (1974).
Methods for Converging Open-Shell Hartree-Fock Wavefunctions.

33 P. Pulay, Chem. Phys. Lett. 73, 393-398 (1980).
Convergence Acceleration of Iterative Sequences. The Case of SCF Iteration.

34 P. Pulay, J. Comp. Chem. 3, 556-560 (1982).
Improved SCF Convergence Acceleration.

35 T. P. Hamilton and P. Pulay, J. Chem. Phys. 84, 5728-5734 (1986).
Direct inversion in the iterative subspace (DIIS) optimization of open-shell, excited-state, and small multiconfigurational SCF wave function.

36 P. E. Maslen, C. Ochsenfeld, C. A. White, M. S. Lee and M. Head-Gordon, J. Phys. Chem. A 102, 2215-2222 (1998).
Locality and sparsity of ab initio one-particle density matrices and localized orbitals.

37 W. Yang, Phys. Rev. A 44, 7823-7826 (1991).
Direct Calculation of Electron Density in Density Functional Theory: Implementation for Benzene and a Tetrapeptide.

38 Q. Zhao and W. Yang, J. Chem. Phys. 102, 9598-9603 (1995).
Analytical energy gradients and geometry optimization in the divide-and-conquer method for large molecules.

39 W. Yang and T.-S. Lee, J. Chem. Phys. 103, 5674-5678 (1995).

A density-matrix divide-and-conquer approach for electronic structure calculations of large molecules.

40 A. van der Vaart, D. Suarez and K. M. Merz, J. Chem. Phys. 113, 10512-10523 (2000).
Critical assessment of the performance of the semiempirical divide and conquer method for single point calculations and geometry optimizations of large chemical systems.

41 M. S. Daw, Phys. Rev. B 47, 10895-10898 (1993).
Model for energetics of solids based on the density matrix.

42 R. W. Nunes and D. Vanderbilt, Phys. Rev. B 50, 17611-17614 (1994).
Generalization of the density matrix method to a nonorthogonal basis.

43 S.-Y. Qiu, C. Wang, K. M. Ho and C. Chan, J. Phys. Condens. Matter 6, 9153-9172 (1994).
Tight-binding molecular dynamics with linear system size scaling.

44 C. H. Xu and G. E. Scuseria, Chem. Phys. Lett. 262, 219-226 (1996).
An O(N) tight-binding study of carbon clusters up to C_{8640}.

45 E. Hernandez, M. J. Gillan and C. M. Goringe, Phys. Rev. B 53, 7147-7157 (1996).
Linear-scaling density-functional-theory technique: The density-matrix approach.

46 J. M. Millam and G. E. Scuseria, J. Chem. Phys. 106, 5569-5577 (1997).
Linear scaling conjugate gradient density matrix search as an alternative to diagonalization for first principles electronic structure calculations.

47 A. D. Daniels and G. E. Scuseria, J. Chem. Phys. 110, 1321-1328 (1999).
What is the best alternative to diagonalization of the Hamiltonian in large-scale semiempirical calculations ?

48 C. Ochsenfeld and M. Head-Gordon, Chem. Phys. Lett. 270, 399-405 (1997).
A reformulation of the coupled perturbed self-consistent field equations entirely within a local atomic orbital density matrix-based scheme.

49 M. Challacombe, J. Chem. Phys. 110, 2332-2342 (1999).
A simplified density matrix minimization for linear scaling self-consistent field theory.

50 R. McWeeny, Rev. Mod. Phys. 32, 335-369 (1960).
Some recent advances in density matrix theory.

51 E. S. Krachko, Chem. Phys. Lett. 318, 210-213 (2000).
Generalized idempotency purification transform in linear scaling self-consistent field theory.

52 A. Holas, Chem. Phys. Lett. 340, 552-558 (2001).
Transforms for idempotency purification of density matrices in linear-scaling electronic-structure calculations.

53 S. Habershon and F. R. Manby, Chem. Phys. Lett. 354, 527-528 (2002).
Comment on: "Transforms for idempotency purification of density matrices in linear-scaling electronic-structure calculations".

54 R. Pino and G. E. Scuseria, Chem. Phys. Lett. 360, 117-122 (2002).
Purification of the first-order density matrix using steepest descent and Newton-Raphson methods.

55 P. D. Haynes and M. C. Payne, Phys. Rev. B 59, 12173-12176 (1999).
Corrected penalty-functional method for linear-scaling calculations within density-functional theory.

56 T. Helgaker, H. Larsen, J. Olsen and P. Jorgensen, Chem. Phys. Lett. 327, 397-403 (2000).
Direct optimization of the AO density matrix in Hartree-Fock and Kohn-Sham theories.

57 H. Larsen, J. Olsen, P. Jorgensen and T. Helgaker, J. Chem. Phys. 115, 9685-9697 (2001).
Direct optimization of the atomic-orbital density matrix using the conjugate-gradient method with a multilevel preconditioner.

58 H. Larsen, T. Helgaker, J. Olsen and P. Jorgensen, J. Chem. Phys. 115, 10344-10352 (2001).
Geometrical derivatives and magnetic properties in atomic-orbital density-based Hartree-Fock theory.

59 T.-S. Lee, J. P. Lewis and W. Yang, Comp. Mat. Sci. 12, 259-277 (1998).
Linear-scaling quantum mechanical calculations of biological molecules: The divide-and-conquer approach.

60 J. J. P. Stewart, A. Csaszar and P. Pulay, J. Comp. Chem. 3, 227-228 (1982).
Fast Semiempirical Calculations.

61 K. R. Bates, A. D. Daniels and G. E. Scuseria, J. Chem. Phys. 109, 3308-3312 (1998).
Comparison of conjugate gradient density matrix search and Chebyshev expansion methods for avoiding diagonalization in large-scale electronic structure calculations.

62 D. R. Bowler and M. J. Gillan, Comp. Phys. Comm. 120, 95-108 (1999).
Density matrices in O(N) electronic structure calculations: theory and applications.

63 U. Stephan, Phys. Rev. B 62, 16412-16424 (2000).
Comparison of the convergence properties of linear-scaling electronic-structure schemes for nonorthogonal bases.

64 J. P. Lewis, C. W. Carter, J. Hermans, W. Pan, T.-S. Lee and W. Yang, J. Am. Chem. Soc. 120, 5407-5410 (1998).
Active Species for the Ground-State Complex of Cytidine Deaminase: A Linear-Scaling Quantum Mechanical Investigation.

65 N. Diaz, D. Suarez, T. L. Sordo and K. M. Merz, J. Am. Chem. Soc. 123, 7574-7583 (2001).
A theoretical study of the aminolysis reaction of lysine 199 of human serum albumin with benzylpenicillin: Consequences of immunochemistry of penicillins.

66 J. Khandogin, A. Hu and D. M. York, J. Comp. Chem. 21, 1562-1571 (2002).
Electronic Structure Properties of Solvated Biomolecules: A Quantum Approach for Macromolecular Characterization.

67 J. Khandogin and D. M. York, J. Phys. Chem. B 106, 7693-7703 (2002).
Quantum Mechanical Characterization of Nucleic Acids in Solution: A Linear-Scaling Study of Charge Fluctuations in DNA and RNA.

68 A. van der Vaart, V. Gogonea, S. L. Dixon and K. M. Merz, J. Comp. Chem. 21, 1494-1504 (2000).
Linear Scaling Molecular Orbital Calculations of Biological Systems Using the Semiempirical Divide and Conquer Method.

69 T. Helgaker, P. Jorgensen and J. Olsen, *Molecular Electronic Structure Theory*, Wiley, Chicester, 2000, pp. 468-478.

70 W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in FORTRAN*, Second Edition, Cambridge University Press, Cambridge, 1992.

71 P. Ordejon, Phys. Stat. Sol. B 217, 335-356 (2000).
Linear scaling ab initio calculations in nanoscale materials with SIESTA.

72 A. D. Daniels, G. E. Scuseria, O. Farkas and H. B. Schlegel, Int. J. Quantum Chem. 77, 82-89 (2000).
Geometry Optimization of Kringle 1 of Plasminogen Using the PM3 Semiempirical Method.

73 J. J. P. Stewart, J. Mol. Struct. (Theochem) 401, 195-205 (1997).
Calculation of the geometry of a small protein using semiempirical methods.

74 S. J. Titmuss, P. L. Cummins, A. A. Bliznyuk, A. P. Rendell and J. E. Gready, Chem. Phys. Lett. 320, 169-176 (2000).
Comparison of linear-scaling semiempirical methods and combined quantum mechanical / molecular mechanical methods applied to enzyme reactions.

75 S. J. Titmuss, P. L. Cummins, A. P. Rendell, A. A. Bliznyuk and J. E. Gready, J. Comput. Chem. 23, 1314-1322 (2002).
Comparison of Linear-Scaling Semiempirical Methods and Combined Quantum Mechanical/Molecular Mechanical Methods for Enzymic Reactions. II. An Energy Decomposition Analysis.

76 A. A. Bliznyuk, A. P. Rendell, T. W. Allen and S.-H. Chung, J. Phys. Chem. B 105, 12674-12679 (2001).
The Potassium Ion Channel: Comparison of Linear-Scaling Semiempirical and Molecular Mechanics Representations of the Electrostatic Potential.

77 W. Thiel and A.A. Voityuk, Theoret. Chim. Acta 81, 391-404 (1992); 93, 315 (1996).
Extension of the MNDO Formalism to d Orbitals: Integral Approximations and Preliminary Numerical Results.

78 W. Thiel and A.A. Voityuk, J. Phys. Chem. 100, 616-626 (1996).
Extension of MNDO to d Orbitals: Parameters and Results for the Second-Row Elements and for the Zinc Group.

79 S. Pissanetzky, *Sparse Matrix Technology*, Academic Press, London, 1984.

80 I. S. Duff, A. M. Erisman and J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, 1986.

81 Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.

82 M. Challacombe, Comp. Phys. Comm. 128, 93-107 (2000).
A general parallel sparse-blocked matrix multiply for linear scaling SCF theory.

83 D. R. Bowler, T. Miyazaki and M. J. Gillan, Comp. Phys. Comm. 137, 255-273 (2001).
Parallel sparse matrix multiplication for linear scaling electronic structure calculations.

84 S. Itoh, P. Ordejon and R. M. Martin, Comp. Phys. Comm. 88, 173-185 (1995).
Order-N tight-binding molecular dynamics on parallel computers.

85 A. Canning, G. Galli, F. Mauri, A. de Vita and R. Car, Comp. Phys. Comm. 94, 89-102 (1996).
O(N) tight-binding molecular dynamics on massively parallel computers: an orbital decomposition approach.

86 Y. Saad, *SPARSKIT: a basic tool kit for sparse matrix computations*, version 2, Minneapolis, 1994.

87 W. Thiel and A.A. Voityuk, Int. J. Quant. Chem. 44, 807-829 (1992).
Extension of MNDO to d Orbitals: Parameters and Results for the Halogens.

88 See http://www.specbench.org/cpu95/results/cpu95.html
Data under CFP95: Compaq XP1000 = 65.5, IBM RS/6000 3CT = 10.2.

89 See http://www.t12.lanl.gov/~mchalla/

90 See http://www.rcsb.org/pdb.

91 H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, Nucl. Ac. Res. 28, 235-242 (2000).
The Protein Data Bank.

92 Insight II, Accelrys, 9685 Scranton Road, San Diego CA 92121-3752, USA.

93 B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, J. Comp. Chem. 1983, 4, 187-217.
CHARMM - a program for macromolecular energy, minimization, and dynamics calculations.

94 See http://yuri.harvard.edu.

95 W. R. P. Scott, P. H. Hünenberger, I. G. Tironi, A. E. Mark, S. R. Billeter, J. Fennen, A. E. Torda, T. Huber, P. Krüger, and W. F. van Gunsteren, J. Phys.Chem. A 1999, 103, 3596-3607.
The GROMOS biomolecular simulation program package.

96 See http://www.igc.ethz.ch/gromos-docs/index.html.

97 L. Streyer, *Biochemistry*, 4th edition, W. H. Freeman, New York, 1995.

98 S. R. Billeter, C. F. W. Hanser, T. Z. Mordasini, M. Scholten, W. Thiel, and W. F. van Gunsteren, Phys. Chem. Chem. Phys. 3, 688-695 (2001).
Molecular Dynamics Study of Oxygenation Reactions Catalysed by the Enzyme p-Hydroxybenzoate Hydroxylase.

99 C. Lennartz, A. Schäfer, F. Terstegen and W. Thiel, J. Phys. Chem. B 106, 1758-1767 (2002).
Enzymatic Reactions of Triosephosphate Isomerase: A Theoretical Calibration Study.

100 B. Entsch and W. J. H. van Berkel, FASEB J. 9, 476-483 (1995).
Structure and Mechanism of para-Hydrobenzoate Hydroxylase.

101 B. Entsch, D. P. Ballou, and V. Massey, J. Biol. Chem. 251, 2550-2563 (1976).
Flavin-Oxygen Derivatives Involved in Hydroxylation of p-Hydrobenzoate Hydroxylase.

102 B. Entsch and D. P. Ballou, Biochim. Biophys. Acta 999, 313-322 (1989).
Purification, properties and oxygen reactivity of p-hydrobenzoate hydroxylase from Pseudomonas aeruginosa.

103 M. Ortiz-Maldonado, D. P. Ballou, and V. Massey, Biochemistry 38, 8124-8137 (1999).
Use of Free Energy Relationships to Probe the Individual Steps of Hydroxylation by p-Hydrobenzoate Hydroxylase: Studies with a Series of 8-Substituted Flavins.

104 W. J. H. van Berkel and F. Müller, J. Biochem. 179, 307-314 (1989).
The temperature and pH dependence of some properties of p-hydrobenzoate hydroxylase from Pseudomonas fluorescens.

105 Q. Cui and M. Karplus, J. Am. Chem. Soc. 123, 2284-2290 (2001).
Triosephosphate Isomerase: A Theoretical Comparison of Alternative Pathways.

106 Q. Cui and M. Karplus, J. Phys. Chem. B 106, 1768-1798 (2002).
Quantum Mechanical / Molecular Mechanical Studies of the Triosephosphate Isomerase-Catalyzed Reaction: Verification of Methodology and Analysis of Reaction Mechanisms.

107 P. A. Bash, M. J. Field, R. C. Davenport, G. A. Petsko, D. Ringe, and M. Karplus, Biochemistry 30, 5826-5832 (1991).
Computer Simulation and Analysis of the Reaction Pathway of Triosephosphate Isomerase.