Sparse projected-gradient method as a linear-scaling low-memory alternative to diagonalization in self-consistent field electronic structure calculations*

Ernesto G. Birgin [†] J. M. Martínez [‡] Leandro Martínez [§] Gerd B. Rocha [¶]

December 17, 2012.

Abstract

Large-scale electronic structure calculations usually involve huge nonlinear eigenvalue problems. A method for solving these problems without employing expensive eigenvalue decompositions of the Fock matrix is presented in this work. The sparsity of the input and output matrices is preserved at every iteration and the memory required by the algorithm scales linearly with the number of atoms of the system. The algorithm is based on a projected gradient iteration applied to the constraint fulfillment problem. The computer time required by the algorithm also scales approximately linearly with the number of atoms (or non-null elements of the matrices), and the algorithm is faster than standard implementations of modern eigenvalue decomposition methods for sparse matrices containing more than 50,000 non-null elements. The new method reproduces the sequence of semiempirical SCF

*This work was supported by PRONEX-CNPq/FAPERJ Grant E-26/171.164/2003 - APQ1, FAPESP Grants 2003/09169-6, 2006/53768-0, 2008/00468-4, 2010/16947-9, CEPID on Industrial Mathematics, INCT-INAMI, and CNPq.

[†] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, SP, Brazil, email: egbirgin@ime.usp.br

[‡] Department of Applied Mathematics, Institute of Mathematics, Statistics and Scientific Computing, State University of Campinas, Campinas, SP, Brazil. email: martinez@ime.unicamp.br

[§] Institute of Chemistry, State University of Campinas, Campinas, SP, Brazil. email: leandro@iqm.unicamp.br

[¶] Department of Chemistry, Federal University of Paraíba, João Pessoa, Paraíba, Brazil. email: gbr@quimica.ufpb.br

iterations obtained by standard eigenvalue decomposition algorithms to good precision.

Key words: Electronic Structure Calculations, Semiempirical methods, Projected Gradient, linear scaling, sparsity.

1 Introduction

For fixed nuclei coordinates, an electronic structure calculation consists of finding the wave functions from which the spatial electronic distribution of the system can be derived.[1] These wave functions are the solutions of the time-independent Schrödinger equation.[1]

The practical solution of the Schrödinger equation is computationally very demanding. Therefore, simplifications are made leading to more tractable mathematical problems. The best-known approach consists of approximating the solution by a (Slater) determinant. Such approximation allows for a significant simplification which results in a "one-electron" eigenvalue (Hartree-Fock) equation. The solutions of this eigenvalue problem are used to reconstitute the Slater-determinant and, therefore, the electronic density of the system.

For simplicity, our discussion will focus on the Restricted-Hartree-Fock (RHF) case, for which the number of electrons is $2N$, where N is the number of functions that compose the Slater-determinant. These functions are written as linear combinations of a basis with K elements, thus the unknowns of the problem turn out to be the coefficients of the unknown functions with respect to the basis, giving rise to the Hartree-Fock-Roothaan nonlinear eigenvalue problem.[1] The discretization technique uses plane wave basis or localized basis functions with compact support[2] or with a Gaussian fall-off.[3] In this way, the unknowns of the problem are represented by a coefficient matrix $C \in \mathbb{R}^{K \times N}$. The optimal choice of the coefficients comes from the solution of the optimization problem:

$$\begin{aligned} &\text{Minimize} && E(P) \\ &\text{subject to} && P = P^T,\ P\mathcal{M}P = P,\ \text{Trace}(P\mathcal{M}) = N, \end{aligned} \tag{1}$$

where $\mathcal{M}$ is a symmetric positive definite overlap matrix which is computed from the basis and $P = CC^T$ is known as the Density matrix.

Within RHF, the form of $E(P)$ in (1) is:

$$E_{RHF}(P) = \text{Trace}\Big[2HP + G(P)P\Big],$$

where P is the one-electron Density matrix in the atomic-orbital (AO) basis, H is the one-electron Hamiltonian matrix, the entries of $G(P)$ are given by

$$G_{ij}(P) = \sum_{k=1}^{K}\sum_{\ell=1}^{K}(2g_{ijk\ell} - g_{i\ell kj})P_{\ell k},\ 1 \leq i, j \leq K,$$

$g_{ijk\ell}$ is a two-electron integral in the AO basis, K is the number of functions in the basis, and $2N$ is the number of electrons. For all $i, j, k, \ell = 1, \ldots, K$ one has:

$$g_{ijk\ell} = g_{jik\ell} = g_{ij\ell k} = g_{k\ell ij}.$$

The Fock matrix is defined by $F(P) \equiv H + G(P)$ and direct calculation shows that:

$$\nabla E_{RHF}(P) = 2F(P).$$

Since $G(P)$ is linear, the objective function $E_{RHF}(P)$ is quadratic.

The best known algorithm for solving (1) is given by the SCF fixed-point iteration.[1] Given an iterate P_k that satisfies the constraints of (1), the next iterate is defined as the minimizer of the linear approximation of $E(P)$ at P_k on the true feasible region.[4] Therefore, since $\nabla E(P) = 2F(P)$, it turns out that P_{k+1} is a solution of

$$\begin{array}{ll} \text{Minimize} & \text{Trace}[F(P_k)P] \\ \text{subject to} & P = P^T,\ P^2 = P,\ \text{Trace}(P) = N. \end{array} \tag{2}$$

Note that $F(P_k)$ is *the Fock Matrix* associated with the *Density Matrix* P_k. The solution of (2) is a projection matrix onto the subspace generated by the eigenvectors associated with the N lowest eigenvalues of $F(P_k)$. In the case of multiplicity of the N-th eigenvalue, multiple solutions exist. As a consequence, the standard form of solving (2) relies on well-established solvers for eigenvalue calculations like Lapack[5] or Arpack.[6]

In the context of the SCF iteration, two basic improvements are usually employed: DIIS-extrapolation,[7] by means of which convergence to the solution of (1) is accelerated, and approximate solution (instead of exact) of (2), in order to abbreviate the computer work associated with the first iterations of the Fixed Point method.[8–10] The global convergence properties of the Fixed-Point SCF method can be improved by means of trust-region techniques.[4,11–13] Moreover, effective optimization algorithms that include the trust-region paradigm and exploit the case in which $N \ll K$ were studied.[14–16]

If the number of atoms is large, both the computation of the Fock Matrix and the solution of (2) may be very expensive. Modern techniques as Fast Multipole methods[17] have been able to reduce the computer time associated with Fock Matrix computations to a multiple of N. However, the eigenvalue calculations typically involve $O(N^3)$ floating point operations, which is unaffordable for large molecular systems. Therefore, most recent research involving linear scaling methods for electronic structure calculations aims to reduce the computer time dedicated to solve (2).[8–10,18–22]

The solution of the eigenvalue problem for very large systems may be possible because the electron density is naturally sparse. Except when long-range electron delocalization is present (as for periodic systems at low temperature), every wave function is to some degree localized and, therefore, for sufficiently large systems, wave functions corresponding to distant groups do not overlap.[23] Only algorithms that explore the sparsity of the electron density at every step make it possible the computation of the electronic density of very large systems. Different strategies handle the sparsity using physically sensible arguments even for medium-size systems. Electronic structure calculations that incorporate cutoffs for long-range integral overlaps are quite common.[24–26] The more aggressive strategy for incorporating sparsity relies on localization of the molecular orbitals.[25,27] With these assumptions one can accelerate the calculation of the Fock matrices and we may substitute the solution of the eigenvalue problem by suitable approximations.[25] Localized orbital methods lead to rapid responses and are useful for many systems for which one seeks mostly structural and energetic parameters.[25,27] Nevertheless, as larger and more complex systems are studied, methods substituting the eigenvalue decomposition that are not restricted by any specific sparsity representation will be required.

The best known methods for solving (2) in the large-scale case can be classified in two groups: the ones focusing on the explicit minimization of the functional, and the ones involving Density matrix iterations without explicitly evoking optimization arguments. The best known minimization-based methods were given by Millam et al.[8] and Li et al.[19] In both cases problem (2) is reduced to the unconstrained minimization of a cubic function, which is processed using conjugate gradients. The cubic nature of the objective function eliminates the possibility of having more than one local minimizer. By the same reason global minimizers do not exist and the iterated functional values could go to $-\infty$, although experiments suggest that this failure is not very common in real-life calculations.[8]

The second group of methods began with the purification scheme of McWeeny,[28] which was adapted by Palser and Manolopoulos[29] to the solution of problem (2). The idea is that, starting from a suitable initial Density matrix, the McWeeny iteration, which merely aimed to achieve idempotency ($P^2 = P$), in fact converges to solutions of the more structured problem (2). Alternative purification schemes have been suggested in many papers[20,21,30–33] and careful error analyses were given by Rubensson and Zahedi[20] and Rubensson and Rudberg.[31] In the Grand-Canonical Purification method[29] a point in the HOMO-LUMO gap is supposed to be known, the initial iterate is conveniently chosen as a transformation of the Fock matrix and global quadratic convergence follows as a consequence of elementary properties of the one-dimensional iteration $x_{k+1} = 3x_k^2 - 2x_k^3$. In the Canonical Purification method the iteration is considerably more complicated than in the Grand-Canonical scheme. On the other hand, the chemical potential is not assumed to be known. Instead, it is updated at each iteration exploiting flexibility of the unstable fixed point, maintaining the number of occupied states and guaranteeing monotone energy decrease. In these methods, the sparsity of the Density is obtained by the application of cutoffs.

The strategy presented in this paper shares characteristics of both groups of methods. On the one hand we iterate truncated Density matrices as done by Palser and Manolopoulus[29] and later improvements[20] but, on the other hand, we rely on a well established damped projected gradient optimization strategy with a global convergence theory for getting suitable solutions. This strategy is consistent with the imposition of a fixed sparsity pattern at each iteration. Although damping is not necessary in the case that sparsity is not imposed (because, in that case, convergence relies only on the properties of McWeeny-like purification), it is essential when one restricts the solution to some given (sparse-like) subspace. As a consequence, our method can benefit both from the development of purification based strategies and from the stability advantages of consolidated optimization approaches. Our outer iteration consists of obtaining an estimate of the position of the gap using a root-finding process. Numerical experiments demonstrate that the algorithm scales linearly with the number of non-null elements of the Fock matrix and, thus, with the number of atoms of the systems, and that the solutions obtained coincide with the solutions of standard eigenvalue decompositions methods to good precision. The method proposed here is faster than standard eigenvalue decomposition strategies for sparse systems with 50 thousand non-null matrix elements, and may incorporate any desired sparsity pattern for the Fock and Density matrices. We will illustrate the reliability of the proposed algorithm in full SCF semiempirical calculations of up to 6 thousand atoms.

Notation Given the symmetric $K \times K$ real matrices A and B, we denote $\langle A, B\rangle = Trace(AB)$. We also denote $\|A\| = \sqrt{\langle A, A\rangle}$.

2 Algorithms

The problem considered in this section is

$$\begin{aligned} &\text{Minimize} && \text{Trace}[AP] \\ &\text{subject to} && P = P^T,\ P^2 = P,\ \text{Trace}(P) = N. \end{aligned} \tag{3}$$

In the large-scale case, the reduction of (3) to an unconstrained optimization problem is very attractive because effective large-scale unconstrained minimization solvers are nowadays available. Problem (3) has been reduced to unconstrained optimization and handled using conjugate gradients.[8,19] However, in recent years projected gradient techniques proved to be very effective for large-scale optimization problems.[34,35] They use even less memory than conjugate gradient methods, they can handle simple constraints (i.e. constraints onto which one knows how to project) and guarantee descent directions that are not affected by the potential accumulation of roundoff errors that is inherent to conjugate gradients.[35] This is due to the fact that the search direction in a projected gradient method does not depend at all on the search directions at previous iterations.

Since the matrix A is symmetric, we have that

$$A = \sum_{i=1}^{K} \sigma_i v_i v_i^T, \tag{4}$$

where $\sigma_1 \leq \ldots \leq \sigma_K$ are its eigenvalues and $v_1, \ldots, v_K$ are the corresponding orthonormal eigenvectors. A solution of (3) is

$$P = \sum_{i=1}^{N} v_i v_i^T. \tag{5}$$

This solution is unique if $\sigma_N < \sigma_{N+1}$. The obvious way for computing the solution of (3) requires to compute the spectral decomposition of A, but this procedure may be unaffordable if N and K are very large.

In order to solve (3) without computing eigenvectors, we will consider the "associated feasibility problem" (AFP) given by

$$\text{Minimize } \Phi(P) \text{ subject to } P = P^T, \tag{6}$$

where $\Phi(P) = \frac{1}{2}\|P^2 - P\|^2$, and the "associated feasibility sparse problem" (AFSP) given by

$$\text{Minimize } \Phi(P) \text{ subject to } P = P^T \text{ and } P \in \mathcal{S}, \tag{7}$$

where $\mathcal{S}$ is a closed convex set of symmetric matrices such that $A \in \mathcal{S}$. In general we define $\mathcal{S}$ as an affine subspace of symmetric matrices with a given sparsity pattern.

We solve (7) using a particular case of the projected gradient method.[36–38] Note that problem (7) has in fact the constraints $P = P^T$ and $P \in \mathcal{S}$. However, since these constraints are simple enough and do not involve inequalities, the method derives its properties directly from its unconstrained counterpart.

Algorithm 2.1

Let the symmetric $K \times K$ matrix $P_0 \in \mathcal{S}$ be a given initial approximation. Initialize $k \leftarrow 0$.

Step 1. Compute $\Gamma_k \equiv \Gamma(P_k)$, the projection of $\nabla\Phi(P_k)$ onto $\mathcal{S}$.

Step 2. Set $t \leftarrow 1$.

Step 3. Test the descent condition

$$\Phi(P_k - t\Gamma_k) \leq \Phi(P_k) - 10^{-6}\, t\, \langle \Gamma_k, \nabla\Phi(P_k)\rangle. \tag{8}$$

If (8) holds, define $P_{k+1} = P_k - t\Gamma_k$, update $k \leftarrow k + 1$ and go to Step 1.

Step 4. Compute a new value of t in the interval $[0.1t, 0.5t]$ (usually by quadratic or quartic interpolation[37]) and go to Step 3.

It can be proved[36–38] that every limit point P_* of a sequence generated by Algorithm 2.1

is a critical point of (7) ($\Gamma(P_*) = 0$). Roughly speaking, as we will see below, this implies that the algorithm converges to a minimum of $\frac{1}{2}\|P^2 - P\|$ subject to $P \in \mathcal{S}$. We always choose $\mathcal{S}$ in such a way that it contains the Identity matrix, therefore global minimizers of $\frac{1}{2}\|P^2 - P\|$ are, in fact, solutions of $P^2 = P$. Our implementation of Algorithm 2.1 does not involve the explicit computation of the projection of $\nabla\Phi(P_k)$ onto $\mathcal{S}$. Instead, the problem (3) is formulated from the beginning in terms of the free variables that describe $\mathcal{S}$ and is handled as an unconstrained problem on this reduced set of variables. Accordingly, the gradient (with respect to the free variables) is computed employing an appropriate data structure with standard reverse differentiation[39] which trivially gives rise to the projected gradient.

We still need to show that a limit point P_* of a sequence generated by Algorithm 2.1 satisfies, not only $P^2 = P$ but also $\text{Trace}(P) = N$ and that it minimizes $\text{Trace}(AP)$. For the sake of clarity, we will define Algorithm 2.1b as being identical to Algorithm 2.1 except for the definition of the set $\mathcal{S}$, which, in Algorithm 2.1b, will be the whole subspace of symmetric $K \times K$ real matrices. Therefore, in Algorithm 2.1b the matrix Γ_k is the projection of $\nabla\Phi(P_k)$ onto the set of symmetric matrices ($\Gamma_k = \frac{1}{2}(\nabla\Phi(P_k) + \nabla\Phi(P_k)^T)$. (Of course, no projection is necessary in this case since $\nabla\Phi(P_k)$ is already symmetric.) By direct calculations, if P and ΔP are symmetric matrices, we have:

$$\Phi(P + \Delta P) = \frac{1}{2}\|(P + \Delta P)^2 - (P + \Delta P)\|^2 = \Phi(P) + \langle 2P^3 - 3P^2 + P, \Delta P\rangle + O(\|\Delta P\|^2).$$

Therefore, the projection of $\nabla\Phi(P)$ onto the subspace of symmetric matrices is given by

$$\Gamma(P) = 2P^3 - 3P^2 + P.$$

It turns out that, if $\Gamma(P_*) = 0$, as guaranteed by the gradient projection theory, P_* is a fixed point of the McWeeny purification process $P_{k+1} = 3P_k^2 - 2P_k^3$. This means that the limit points of the projected gradient method are matrices with eigenvalues 0, 1, and $\frac{1}{2}$. Eigenvalue decomposition and second derivative computations show that eigenvalues 0 and 1 correspond to minimizers whereas the eigenvalue $\frac{1}{2}$ represents a direction along which the objective function is maximized. So, local minimizers correspond only to matrices with eigenvalues 0 and 1. Therefore, the limit points of the projected gradient process are global minimizers of Φ (solutions of $P^2 = P$) with probability 1.

Assume that the spectral decomposition (with increasing eigenvalues) of P_0 is:

$$P_0 = \sum_{i=1}^{K} \lambda_i w_i w_i^T. \tag{9}$$

Then, if $\lambda_1, \ldots, \lambda_{K-N_0}$ are in the interval $(\frac{1-\sqrt{3}}{2}, 1/2) \approx (-0.366, 0.5)$, and $\lambda_{K-N_0+1}, \ldots, \lambda_K$ are in the interval $(1/2, \frac{1+\sqrt{3}}{2}) \approx (0.5, 1.366)$, it is well known[40] that the purification process converges quadratically to the matrix P_* given by

$$P_* = \sum_{i=K-N_0+1}^{K} w_i w_i^T. \tag{10}$$

We aim to employ an initial point P_0 in such a way that the limit P_* will solve (3). According to (5), (9), and (10), the eigenvectors of P_0 should be those of A and the eigenvalues of P_0 should be in the adequate intervals to ensure convergence.

Since the matrix $\alpha I - A$ has the same eigenvectors as A and its eigenvalues are $\alpha - \sigma_K \leq \cdots \leq \alpha - \sigma_1$, for an appropriate computable value of $\beta > 0$ we have that all the eigenvalues of $\beta[\alpha I - A]$ lie between $-\frac{1}{2}$ and $\frac{1}{2}$. Therefore, choosing $P_0 = (1/2)I + \beta[\alpha I - A]$ we have that all the eigenvalues of P_0 are in $[0, 1]$ and the eigenvectors coincide with those of A. Applying Algorithm 2.1b, eigenvalues bigger than $\frac{1}{2}$ would converge to 1 and eigenvalues smaller than $\frac{1}{2}$ would converge to 0. Clearly, eigenvalues of P_0 bigger than $\frac{1}{2}$ correspond to eigenvalues of A smaller than α and eigenvalues of P_0 smaller than $\frac{1}{2}$ correspond to eigenvalues of A bigger than α. Thus, the limit matrix $P_*(\alpha)$ can be written as

$$P_*(\alpha) = \sum_{i=1}^{N_0(\alpha)} v_i v_i^T,$$

where $v_1, \ldots, v_{N_0(\alpha)}$ are the eigenvectors of A corresponding its $N_0(\alpha)$ smaller eigenvalues. Obviously, the eigenvalues of $P_*(\alpha)$ are 1, with multiplicity $N_0(\alpha)$, and 0 with multiplicity $K - N_0(\alpha)$, whereas the trace of $P_*(\alpha)$ is $N_0(\alpha)$. Therefore, $P_*(\alpha)$ will be the solution of (3) if and only if $N_0(\alpha) = N$. Note that $N_0(\alpha)$ is a non-decreasing function of α.

If the gap between the eigenvalues N and $N+1$ of A is not very small, it can be expected that the "affordable" Algorithm 2.1 and the "non-affordable" Algorithm 2.1b should exhibit similar qualitative behavior. In order to adress large-scale problems only Algorithm 2.1 is implemented. If a tentative α is smaller than σ_N, the trace of $P_*(\alpha)$ will be smaller than N whereas this trace will be bigger than N if the tentative α is bigger than σ_{N+1}. Therefore, the trace of P_k provides a suitable criterion for deciding whether α should be increased or decreased. The algorithm Bisalfa, described below, explains the way in which a satisfactory α is found, and, consequently, problem (3) is solved.

Algorithm 2.2: Bisalfa

Compute, using Gershgorin theorem,[41] lower and upper bounds $\sigma_{\min}$ and $\sigma_{\max}$ for the eigenvalues of A. Define $\alpha_{\min} = \sigma_{\min}$, $\alpha_{\max} = \sigma_{\max}$, and $\alpha = ((N + \frac{1}{2})\alpha_{\min} + (K - N - \frac{1}{2})\alpha_{\max})/K$.

Step 1. Compute β and P_0 as explained above. Compute $P_*(\alpha)$ and $N_0(\alpha)$ using Algorithm 2.1.

Step 2. If $N_0(\alpha) > N$ re-define $\alpha_{\max} \leftarrow \alpha$, choose α annihilating the linear interpolation between $(\alpha_{\min},\ \text{Trace}(\alpha_{\min}) - N)$ and $(\alpha_{\max},\ \text{Trace}(\alpha_{\max}) - N)$ (or taking $\alpha = (\alpha_{\min} + \alpha_{\max})/2$ if the value of α computed by interpolation is excessively close to $\alpha_{\min}$ or to $\alpha_{\max}$), and go to Step 1.

Both Algorithms 2.1 and 2.2 have been implemented using suitable stopping criteria which take into account compatibility with complete SCF calculations. Algorithm 2.1 stops when $\sum_{ij}[\Gamma(P)_{ij}]^2)/K \leq 10^{-12}$ and Algorithm 2.2 stops when $|N_0(\alpha) - N| \leq 0.45$. (In general, when this criterion is met we observe, in practice, that $|N_0(\alpha) - N| \leq 0.001$.) Stopping criteria revealing possible failures of the algorithms were also implemented.

Density matrix purification methods with rigorous error control, developed in[20,21,30–32] and other papers, resemble the Bisalfa technique in some aspects, although they are not based on optimization arguments. In these methods confidence intervals for the HOMO and LUMO eigenvalues are computed using specific Lanczos-like procedures and the purification technique is proved to be satisfactory due to careful monitoring of the error that results from truncating small elements of the approximate Density. These techniques can be used to estimate the Bisalfa error associated with the a priori sparsity constraint $P \in \mathcal{S}$. By formula (11) of Rubensson and Rudberg,[31] if $\tilde{A}$ is the projection of A onto $\mathcal{S}$, $E = \tilde{A} - A$, ξ is the HOMO-LUMO gap, $\|E\|_2 \leq \varepsilon\xi/(1+\varepsilon)$, and $\|\cdot\|_2$ is the spectral matricial norm, we have that the angle between the corresponding subspaces generated by the eigenvectors associated to the N smaller eigenvalues is smaller than ε. Thus, by knowing the HOMO-LUMO gap we are able to control, in principle, the error associated with the application of Algorithm 2.1 and the sparsity assumption. The cost associated with the computation of $\|E\|_2$ may be alleviated by the employment of a "mixed norm".[31] The error may be iteratively updated in the process of purification[20] with a dynamic choice of the constraint set $\mathcal{S}$. The possibility of changing $\mathcal{S}$ at each iteration of Algorithm 2.1 does not modify the projected gradient convergence result if the same $\mathcal{S}$ is eventually used at each iteration of 2.1.

This section can be summarized in the following way:

1. In order to solve (3) we minimize the infeasibility function $\Phi(P)$ subject to sparsity constraints inherited from A. A convergent projected gradient method with sparse reverse differentiation is employed with that purpose.

2. If the initial approximation P_0 is correctly chosen and the HOMO-LUMO gap is not very small, the method plausibly converges to the solution of (3). (Quadratic convergence can also be expected in this case.)

3. The choice of the correct P_0 comes from a safeguarded interpolatory one-dimensional root-finding process with guaranteed convergence.

3 Numerical experiments

In order to compare the efficiency and scalability of Bisalfa relatively to standard diagonalization methods, we performed the eigenvalue decomposition of real Fock matrices obtained using the MOPAC2009 semimepirical quantum chemistry program[42] with RM1 parameters.[43] The matrices in these examples were obtained from the first SCF iteration for spherical clusters of water with 80, 120, 160, 300, 1000, and 5000 molecules. The structures were built with Packmol[44,45] with low density, 0.5 g×ml^{-1}, and a cutoff of 9.0Å was used for electron integrals, so that the matrices are sparse. We used MKL Lapack implementation of routine *dsyev* for eigenvalue decomposition. All examples were run in a single computing node of a SGI Altix XE cluster with Xeon X5670 CPUs and 24 Gb of RAM memory, running RedHat Linux 5.3. Our in-house code was compiled with the Intel Fortran Compiler version 12.0.0 with "`-O3 -mkl:sequential -mcmodel=medium -intel-shared`" options, unless if stated otherwise.

Table 1: Performance of Bisalfa and Lapack for the eigenvalue decomposition of single real Fock matrices obtained from low-density water clusters.

				Computer time / s	
N_{water}	K	N	$N_{non-null}$	Bisalfa	Lapack
80	480	320	25,354	0.51	0.17
120	720	480	39,367	0.91	0.53
160	960	640	52,142	1.21	1.17
300	1,800	1,200	108,907	3.06	8.52
1000	6,000	4,000	391,946	13.40	451.39
5000	30,000	20,000	2,112,571	79.37	43415.64

Bisalfa and Lapack calculations were completed for systems of up to 5000 water molecules. The results of these tests are summarized in Table 1 and Figure 1. The number of non-null elements on the Fock matrix is roughly proportional to the number of atoms of the system, as expected. The log-log plot of the required time as a function of the number of non-null elements indicates the scalability of each method. Lapack eigenvalue decomposition scales with the third power (slope 2.85) of the number of non-null elements, that is, with the third power of the

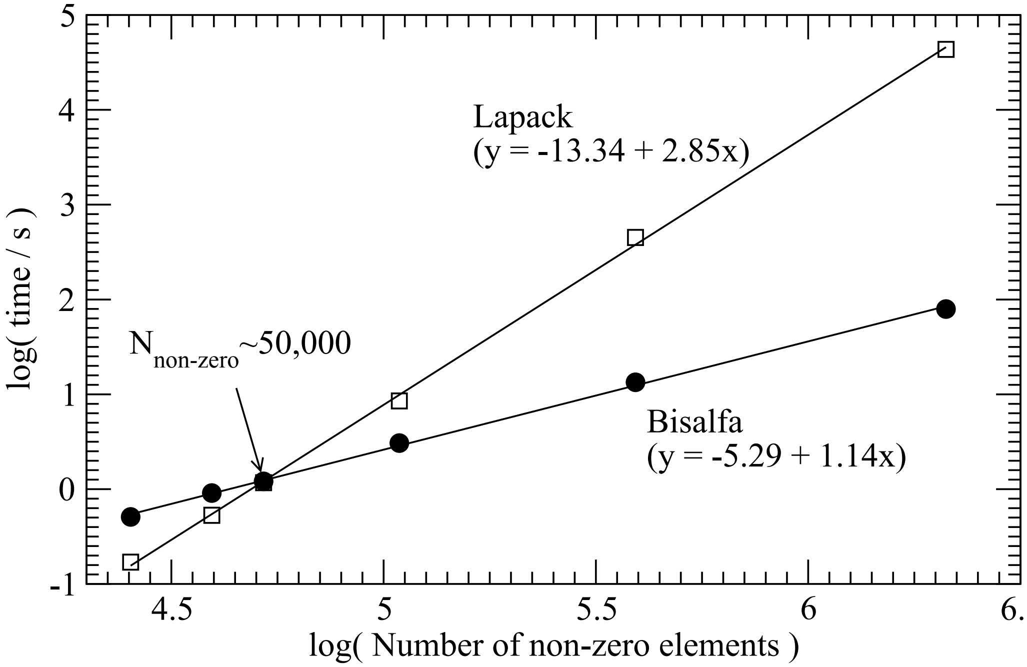


Figure 1: Scaling of Lapack and Bisalfa for systems of up to 2 million non-null elements. Bisalfa is nearly linear-scaling in practice, and it is faster than Lapack for sparse systems with more than 50 thousand non-null matrix elements and ≈ 99.5% sparsity.

number of atoms of the system. Bisalfa, on the other hand, scales almost linearly (slope 1.14). The result is that Bisalfa is faster than Lapack's eigenvalue decomposition for matrices with more than 50 thousand non-null elements and ≈ 99.5% sparse, which in this case corresponded to a system with 160 water molecules.

Now we will show numerically to which degree the substitution of a standard diagonalization method with Bisalfa affects successive iterations of a full SCF procedure. Bisalfa was implemented in Fortran and incorporated to MOPAC2009.[42] The UGTR algorithm[4,11] was also implemented in MOPAC2009 in order to stabilize the SCF calculation and to guarantee convergence. For such, we replaced the ITER_FOR_MOZYME subroutine with one containing our UGTR code. In addition, the modified MOPAC2009 code was compiled using both BLAS and LAPACK subroutines from Intel MKL. Similar strategies were previously used by Maia *et al.* to show that it is possible to obtain large speedups for single point energy calculations just by using CPU serial highly optimized linear algebra libraries in MOPAC2009 code.[46]

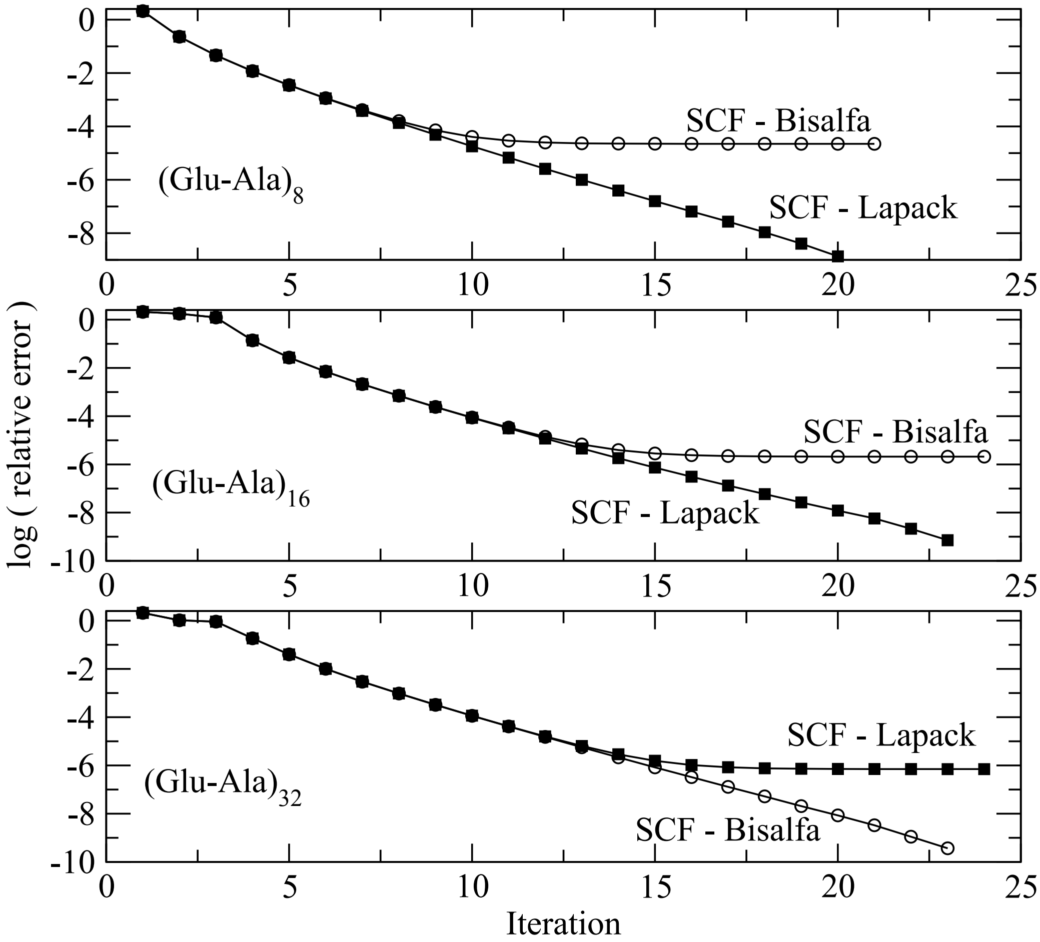


Figure 2: Iterative SCF calculations of poly-(Glu-Ala) peptides using standard diagonalization or Bisalfa. Both methods proceed identically for relative precisions of up to 10^{-4}, demonstrating that the Bisalfa algorithm reproduces the results of standard diagonalizations with good accuracy.

We performed SCF calculations on small Glutamic Acid-Alanine peptides (obtained at the ErgoSCF site: `http://ergoscf.org/xyz/gluala.php`[32]) to compare the sequence of iterates produced by using Bisalfa or Lapack's *dsyev* for small systems different from the water clusters of the previous examples. Figure 2 shows that the SCF iterations are essentially identical up to relative energy errors of about 10^{-4}. Only for precisions greater than those the SCF iterates do not coincide and, in these cases, the results have relative differences of about 10^{-5}. It is expected, in general, that the sparse computation converges to slightly higher energies than the denser computation, as the assumption of a sparsity pattern implies the introduction of additional constraints on the Density matrix. Fluctuations in termination resulting from the stopping criteria can also lead to a slightly smaller energy for Bisalfa relative to Lapack, as occurred for (Glu-Ala)$_{32}$.

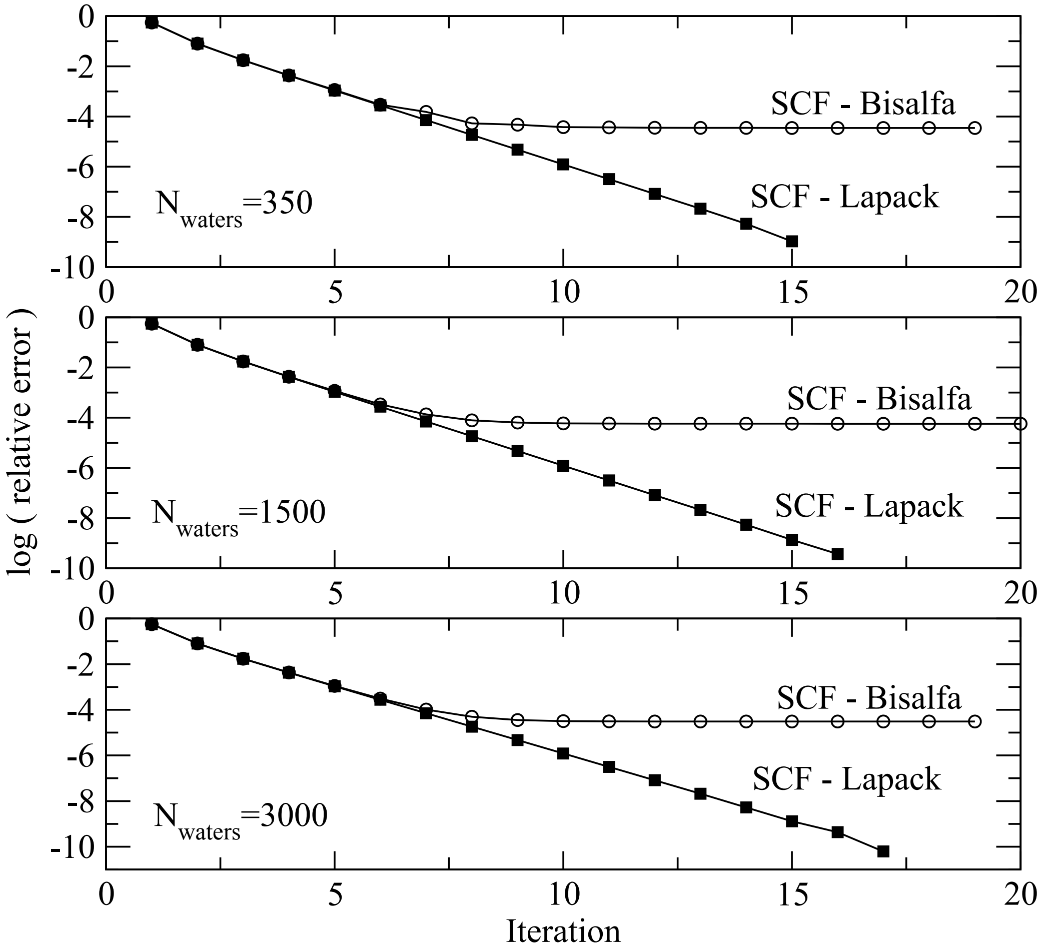


Figure 3: SCF calculations on water clusters of varying size. Detailed data on these problems are described in Table 2.

Similar tests for larger systems composed of spherical clusters containing up to 6 thousand water molecules with 1 g×ml^{-1} density, built with Packmol,[45] were performed. A cutoff for the computation of electron integrals of 9Å was used in these calculations. As only valence

Table 2: Details of the SCF calculations on water clusters of up to 6000 water molecules using Bisalfa or Lapack.

		Iterations[1]		Time		Final energy / kcal mol^{-1}	
N_{waters}	Sparsity (%)	Bisalfa	Lapack	Bisalfa	Lapack	Bisalfa	Lapack
350	89.9	19(19)	16	7.0 min	4.6 min	-18878.2084	-18879.1351
1500	97.2	18(22)	17	56.5 min	6.0 hours	-80772.7671	-80777.4152
3000	98.6	20(21)	18	2.2 hours	61 hours	-161353.3016	-161358.2292
3500	98.8	17(24)	16	4.3 hours	84 hours	-187935.4158	-187940.7421
3500 Parallel MKL using 12 cores					29 hours		-187940.7421
4000	98.9	16(20)	16	3.3 hours	122 hours	-214526.6706	-214532.6311
6000	99.3	22(25)	-	6.4 hours	*	-321889.2913	-

[1] Total number of calls to Algorithm 2.1 from Bisalfa in parentheses.

*Unable to run problem due to lack of memory.

atomic orbitals are explicitly used by the semiempirical calculation, the dimension of the Fock and Density matrices is $6 \times N_{\text{waters}}$, and the number of non-null elements can be computed from the Sparsity by $(1 - \text{Sparsity}/100) \times [1/2 \times 6N_{\text{waters}} \times (6N_{\text{waters}} - 1)]$. As these systems are large, the computation of the Fock matrix with the Mozyme orbital localization method and cuttofs is sparse by itself. In these examples, null elements which are nevertheless allocated by Mozyme were disconsidered in the consecutive Bisalfa iteration.

The SCF profiles are very similar to the ones observed for the small peptide systems. SCF iterates based on Bisalfa and Lapack differ only for relative errors of about 10^{-4}, as shown in Figure 3. For the systems in which Lapack could be used, Bisalfa converged to slightly higher energies and performed two to four iterations more than SCF-Lapack. However, the diagonalization of the converged Bisalfa Fock matrix with Lapack leads in these examples to the same final energy than SCF-Lapack. Therefore, the error in these examples is not cumulative through SCF iterations, but limited to the restrictions imposed by the sparsity of the Density at each iteration.

We were not able to run examples with Lapack for systems with more than 4000 water molecules. The computational times required by SCF-Bisalfa and SCF-Lapack for each of these calculations are shown in Table 2. Computer times of SCF-Bisalfa are roughly proportional to the number of atoms of the system (or the number or water molecules), with oscillations that depend on the number of calls to Algorithm 2.1 and the total number of SCF iterations. We have also run the SCF calculation for 3500 water molecules using the parallel implementation of the MKL libraries (by compiling the codes with "`-mkl:parallel`"), using 12 cores of a single computing node. The wall time required decreased to 29 hours (instead of 84 hours for the serial MKL implementation), but for systems of this size the Bisalfa code was still much faster (4.3 hours).

Therefore, numerical experiments show that SCF calculations can be successfully undertaken by means of the replacement of a standard eigenvalue decomposition method with Bisalfa. In general, the final energy of the sparse computation is larger than the final energy of the dense computation, but this difference is, for the current setup of the methods, of the order of 10^{-4}, which is quite satisfactory for semiempirical calculations. This difference of course can be reduced by choosing larger cutoffs, or by preserving any subset of the Density matrix elements which one believes from physico-chemical arguments that must be non-null and contribute to the electron density of the system. The method presented here does not impose any restriction on the Fock sparsity pattern. The practical efficiency of Bisalfa depends on the size of the gap, but the interpolatory-bisection root-finding procedure can be trivially generalized and parallelized, so that we expect that large systems with small gaps will also benefit from the proposed algorithm.

4 Conclusions

We introduced an alternative algorithm to the standard diagonalization to be used in SCF calculations. The algorithm can be implemented to preserve sparsity of the Fock and Density matrices and scales linearly with the number of non-null elements. It can be used on any SCF calculation relying on standard diagonalizations. The new method is particularly fast for systems with not very small gaps, but its interpolatory root-finding procedure can be trivially generalized and parallelized in order to effectively deal with systems with small gaps as well. Bisalfa is faster than Lapack's eigenvalue decompositions for sparse systems with more than 50 thousands non-null elements and about 99.5% sparse, which in semiempirical calculations correspond to systems with a few hundred atoms. This suggests that the the new method can be useful, to date, for semiempirical calculations which cannot rely on approximate strategies as the localization of molecular orbitals. The method accepts any sparsity pattern for the Fock matrices and preserves the sparsity at every iteration, thus having only modest memory requirements.

The method introduced in this paper relies on projected gradient optimization techniques (Algorithm 2.1) and secant approximate root-finding (Algorithm 2.2). The root-finding process, since it is safeguarded by bijections and is applied to a non-decreasing function, necessarily converges to a solution. The projected gradient iterative method converges with probability 1 to a feasible solution by the projected gradient theory. Therefore, accuracy of the SCF iterate depends on the choice of the sparsity pattern $\mathcal{S}$. In our experiments we used $\mathcal{S}$ as the sparsity pattern of the Fock matrix but different adaptive choices are possible.

5 Appendix - Considerations on sparsity patterns

In our experiments we imposed that the sparsity pattern of the solution should be inherited from the one of the Fock matrix. Different strategies for the proposal of these patterns could be made, based on physico-chemical insights or molecular structure considerations.

First, let us provide a simple, perhaps new, argument to support the assumption that the sparsity structure of a $K \times K$ symmetric A is generally inherited by the solution of (3), which is denoted here by B. (Different arguments that lead to conclusions on the sparsity of P, are available.[47]) Without loss of generality, assume that all the eigenvalues of A are positive. By the Diagonalization theorem one has that $A = A_1 + A_2$, where $A_1 = \sum_{i=1}^{N} \sigma_i v_i v_i^T$ and $A_2 = \sum_{i=N+1}^{K} \sigma_i v_i v_i^T$, the eigenvalues of A are $\sigma_1 \leq \cdots \leq \sigma_K$ and $v_1, \ldots, v_K$ are the associated orthonormal eigenvectors. Thus, $B = \sum_{i=1}^{N} v_i v_i^T$. Assume now that $\sigma_N \ll \sigma_{N+1}$. Then, cancellation between entries of A_1 and A_2 turns out to be quite improbable and, so, a zero-entry of $A = A_1 + A_2$ very likely corresponds to a zero-entry both in A_1 and A_2. This means that, when the gap is sufficiently large, sparsity of A is probably inherited both by A_1 and A_2. Moreover, if the eigenvalues of A_1 are relatively clustered, there exists σ such that $A_1 \approx \sigma \sum_{i=1}^{N} v_i v_i^T$. Therefore, sparsity of A_1 implies sparsity of the solution matrix $\sum_{i=1}^{N} v_i v_i^T$ in this case.

Now we describe how the sparsity of both matrices evolve at each iteration of the calculation of the 350 water molecule cluster described in Table 2, showing that the assumption is valid to a good approximation in the examples presented here. In the calculations of this appendix, no element was desconsidered in the Density matrix, even if it was null (yet allocated) in the Fock matrix. This allows for potentially denser Fock and Density matrices at each iteration, permitting the study of evolution of the sparsity of both matrices. Note that this differs from the approach used in the experiments of Table 2, where null elements of the Fock matrix were not considered in the consecutive application of Algorithm 2.1, thus providing greater speed at the risk of eliminating significant Density elements (the final energies in Table 2 in comparison with Lapack calculations show that the error introduced is small).

Figures 4a&b display how the sparsity patterns of the Fock and Density matrices evolve from iteration to iteration. These plots show the number of elements of these matrices that were smaller than a threshold δ in an iteration and became greater than δ in the next iteration. Both sparsities coincide up to a precision of 10^{-3} after the 13th iteration. After the 15th iteration, only very few (less than 10) elements of these matrices become larger than 10^{-7}. Therefore, the sparsity patterns become constant while the SCF converges, indicating that the assumption of the Fock sparsity pattern does not introduce unexpected instabilities for the Density.

Figure 4c displays the comparison of the sparsity patterns of the Fock and Density matrices in this example. If we consider that an element of the Fock matrix is negligible if smaller than 10^{-7}, we would want to know how many corresponding elements of the Density matrix may be significant. As the Figure 4c shows, only a single negligible Fock element is greater than 10^{-4} at the solution Density, only about 20 elements are greater than 10^{-5}, a few hundred elements are greater than 10^{-6}, and about two thousand elements are greater than 10^{-7}. Therefore, the sparsity pattern of the converged Density matrix is contained to good precision in the sparsity pattern of the Fock matrix in this example, except on the very first iterations.

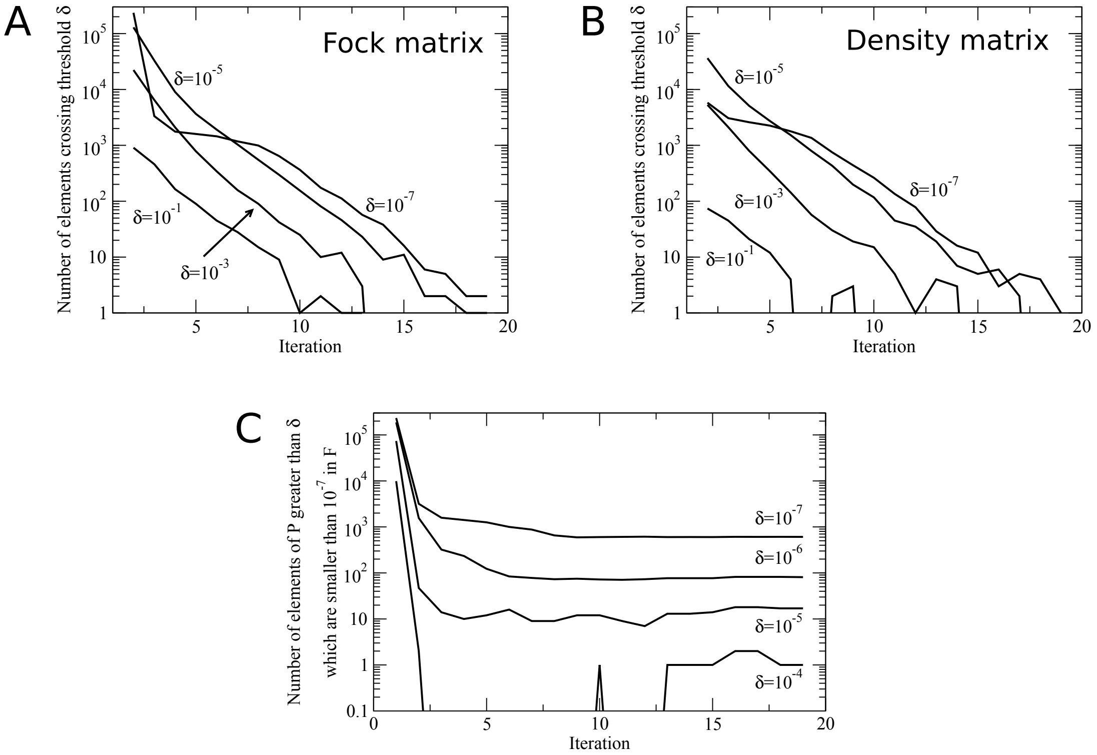


Figure 4: Evolution of the sparsity patterns of the Fock and Density matrices through SCF iterations of a 350 water-molecule cluster. (a) and (b) Stability of the sparsity patterns. The plots indicate the number of elements of each matrix which have increased and crossed a threshold δ from one iteration to the next, for each matrix. (c) Validity of the assumption that the sparsity pattern of the Density matrix is similar to that of the Fock matrix: number of elements of the Density matrix, P greater than a given threshold δ, which are smaller than 10^{-7} in in the Fock matrix, F. These matrices have 2100×2100 ($\sim 4.4 \times 10^6$) elements, of which $\sim 4.2 \times 10^5$ are allocated for calculations after use of orbital localization and structural cutoffs by Mozyme.

Acknowledgements We are indebted to Prof. Francisco Gomes for pointing us the most adequate eigenvalue subroutines to be considered for the problem at hand. LM thanks Fapesp (Proc. 2010/16947-9) for financial support. We also acknowledge two anonymous referees for their useful suggestions.

References

[1] Helgaker, T.; Jorgensen, P.; Olsen, J. *Molecular Electronic-Structure Theory*; John Wiley & Sons: New York, 2000, pp 433-502.

[2] Sánchez-Portal, D.; Ordejón, P.; Artacho, E.; Soler, J. M. Density-functional method for very large systems with LCAO basis sets. *Int. J. Quant. Chem.* **1997,** *65*(5), 453-461.

[3] Pople, J. A.; Headgordon, M.; Fox, D. J.; Raghavachari, K.; Curtiss, L. A. Gaussian-1 Theory - A general procedure for prediction of molecular energies. *J. Chem. Phys.* **1989**, *90*(10), 5622-5629.

[4] Francisco, J. B.; Martínez, J. M.; Martínez, L. Globally convergent trust-region methods for Self-Consistent Field electronic structure calculations. *J. Chem. Phys.* **2004,** *121*(22), 10863-10878

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

[6] Lehoucq, R. B.; Sorensen, D. C.; Yang, C. *ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998.

[7] Pulay, P. Convergence acceleration of iterative sequences: the case of SCF iteration. *Chem. Phys. Lett.* **1980,** *73*(2), 393-398.

[8] Millam J. M.; Scuseria, G. Linear scaling conjugate gradient density matrix search as an alternative to diagonalization for first principles electronic structure calculations. *J. Chem. Phys.* **1997,** *106*(13), 5569-5577.

[9] Daniels, A. D.; Scuseria, G. Semiempirical methods with conjugate gradient density

matrix search to replace diagonalization for molecular systems containing thousands of atoms. *J. Chem. Phys.* **1997,** *107*(2), 425-431.

[10] Daniels, A. D.; Scuseria, G. What is the best alternative to diagonalization of the Hamiltonian in large scale semiempirical calculations? *J. Chem. Phys.* **1999,** *110*(3), 1321-1328.

[11] Francisco, J. B.; Martínez, J. M.; Martínez, L. Density-based globally convergent trust-region method for Self-Consistent Field electronic structure calculations. *J. Math. Chem.* **2006,** *40*(4), 349-377.

[12] Thögersen, L.; Olsen, J.; Yeager, D.; Jörgensen, P.; Salek, P.; Helgaker, T. The trust-region self-consistent field method: Towards a black box optimization in Hartree-Fock and Kohn-Sham theories. *J. Chem. Phys.* **2004,** *121*(16), 16-27.

[13] Thögersen, L.; Olsen, J.; Köhn, A.; Jörgensen, P.; Salek, P.; Helgaker, T. The trust-region self-consistent field method in Kohn-Sham density-functional theory. *J. Chem. Phys.* **2005,** *123*(7), 1-17.

[14] Yang, C.; Meza, J. C.; Lee, B.; Wang, L.-W. KSSSOLV – a MATLAB toolbox for solving the Kohn–Sham equations. *ACM Trans. Math. Softw.* **2009,** *36*(2), 1-35.

[15] Yang, C.; Meza, J. C.; Wang, L.-W. A constrained optimization algorithm for total energy minimization in electronic structure calculations. *J. Comput. Phys.* **2006,** *217*(2), 709-721.

[16] Yang, C.; Meza, J. C.; Wang, L.-W. A trust region direct constrained optimization algorithm for the Kohn-Sham equation. *SIAM J. Sci. Comput.* **2007,** *29*(5), 1854-1875.

[17] Darve, E. The Fast Multipole Method: Numerical Implementation. *J. Comput. Phys.* **2000,** *160*(1), 195-240.

[18] Li, X.; Moss, C. L.; Liang, W.; Feng, Y. Carr-Parrinello density matrix search with a

first principles fictitious electron mass method for electronic wave function optimization. *J. Chem. Phys.* **2009,** *130*(23), 234115.

[19] Li, X.-P.; Nunes, R. W.; Vanderbilt, D. Density-matrix electronic-structure method with linear system-size scaling. *Phys. Rev. B* **1993,** *47*(16), 10891-10894.

[20] Rubensson, E. H.; Zahedi, S. Computation of interior eigenvalues in electronic structure calculations facilitated by density matrix purification. *J. Chem. Phys.* **2008**, *128*(17), 176101.

[21] Rubensson, E. H.; Rudberg, E.; Salek, P. Density matrix purification with rigorous error control. *J. Chem. Phys.* **2008,** *128*(7), 074106

[22] Barrault, M.; Cancès, E.; Hager, W.W.; Le Bris, C. Multilevel domain decomposition for electronic structure calculations, *J. Comput. Phys.* **2007,** *222* 86-109.

[23] Goedecker, S. Linear scaling electronic structure methods. *Rev. Mod. Phys.* **1999,** *71*(4), 1085-1123.

[24] Szekeres, Z.; Mezey, P. G. Fragmentation selection strategies in linear scaling methods. In *Linear-Scaling Techniques in Computational Chemistry and Physics,* first edition; Zaleny, R., Papadopoulus, M. G., Mezey, P. G., Leszczynski, J., Eds.; Springer, New York, 211; pp 147-156.

[25] Stewart, J. J. P. Application of Localized Molecular Orbitals to the Solution of Semiempirical Self-Consistent Field Equations. *Int. J. Quant. Chem.* **1996,** *58*(2), 139-146.

[26] Anikin, N. A.; Anisimov, V. M.; Bugaenko, V. L.; Bobrikov, V. V.; Andreyev, A. M. LocalSCF method for semiempirical quantum-chemical calculation of ultralarge biomolecules. *J. Chem. Phys.* **2004,** *121*(3), 1266-1270.

[27] Ordejón, P.; Artacho, E.; Soler, J. M. Self-consistent order-N density-functional calculations for very large systems. *Phys. Rev. B* **1996**, *53*(16), 10441-10444.

[28] McWeeny, R. Some Recent Advances in Density Matrix Theory. *Rev. Mod. Phys.* **1960,** *32*(2), 335-369.

[29] Palser, A. H. R.; Manolopoulos, D. E. Canonical purification of the density matrix in electronic structure theory. *Phys. Rev. B* **1998,** *58*(19), 12704-12711.

[30] Rudberg, E.; Rubensson, E. H.; Salek, P. Hartree-Fock calculations with linearly scaling memory usage. *J. Chem. Phys.* **2008,** *128*(18), 184106.

[31] Rubensson E. H.; Rudberg, E. Bringing about matrix sparsity in linear-scaling electronic structure calculations. *J. Comput. Chem.* **2011,** *32*(7), 1411-1423

[32] Rudberg, E.; Rubensson, E. H.; Salek, P. Kohn-Sham Density Functional Theory Electronic Structure Calculations with Linearly Scaling Computational Time and Memory Usage. *J. Chem. Theory Comput.* **2011,** *7*(2), 340-350.

[33] Rubensson, E. H. Nonmonotonic Recursive Polynomial Expansions for Linear Scaling Calculation of the Density Matrix. *J. Chem. Theory Comput.* **2011** *7*(5), 1233-1236.

[34] Figueiredo, M. A.; Nowak, R. D.; Wright, S. J. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems. *IEEE J. Sel. Top. Signal Proc.* **2007**, *1*(4), 586-597.

[35] E. G. Birgin, J. M. Martínez, M. Raydan. Spectral Projected Gradient Methods. In *Encyclopedia of Optimization*; 2nd ed. Floudas, C. A.; Pardalos, P. M., Eds.; Springer, 2009, pp 3652-3659.

[36] Birgin, E. G.; Martínez, J. M.; Raydan, M. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.* **2000,** *10*(4), 1196-1211.

[37] Birgin, E. G.; Martínez, J. M.; Raydan. M. Algorithm 813: SPG- Software for convex-constrained optimization. *ACM Trans. Math. Soft.* **2001,** *27*(3), 340-349.

[38] Birgin, E. G.; Martínez, J. M.; Raydan, M. Inexact Spectral Projected Gradient methods on convex sets. *IMA J. Num. Analys.* **2003,** *23*(4), 539-559.

[39] Griewank, A.; Walther, A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Society for Industrial and Applied Mathematics: Philadelphia, PA, 2008, pp 37-65.

[40] Francisco, J. B.; Martínez, J. M.; Martínez, L.; Pisnitchenko, F. I. Inexact Restoration method for minimization problems arising in electronic structure calculations. *Comput. Optim. Appl.* **2011**, *50*(3), 555-590.

[41] Golub, G. H.; Van Loan, C. *Matrix Computations*, Johns Hopkins: Baltimore, MA, 1996.

[42] Stewart, J. J. P. MOPAC2009, version 10.060W; Stewart Computational Chemistry: Colorado Springs, CO, **2009**.

[43] Rocha, G. B.; Freire, R. O.; Simas, A. M.; Stewart, J. J. P. A Reparameterization of AM1 for H, C, N, O, P, S, F, Cl, Br, and I. *J. Comp. Chem.*, **2006,** *27*(10), 1101-1111.

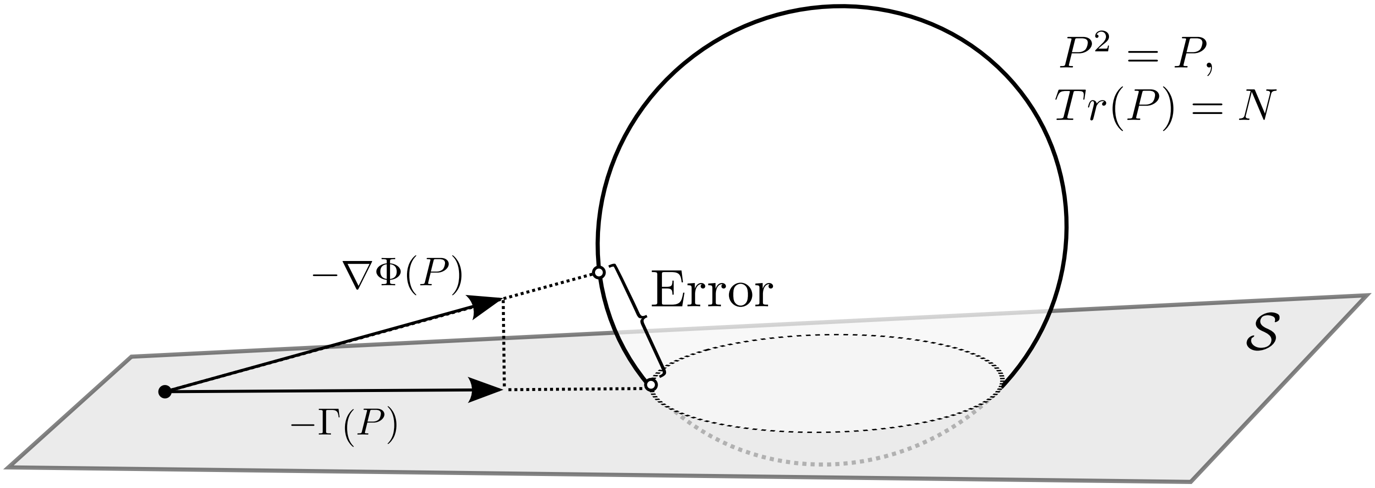
[44] Martínez, J. M.; Martínez, L. Packing optimization for the automated generation of complex system's initial configurations for molecular dynamics and docking. *J. Comp. Chem.* **2003,** *24*(7), 819-825.

[45] Martínez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M. Packmol: A package for building initial configurations for molecular dynamics simulations. *J. Comp. Chem.* **2009,** *30*(13), 2157-2164.

[46] Maia, J. D. C.; Carvalho, G. A. U.; Mangueira Jr., C. P.; Santana, S. R.; Cabral L. A. F.; Rocha, G. B. GPU Linear Algebra Libraries and GPGPU Programming for Accelerating MOPAC Semiempirical Quantum Chemistry Calculations, *J. Chem. Theory Comput.* **2012,** *8*(9), 3072-3081.

[47] Le Bris, C. Computational chemistry from the perspective of numerical analysis. *Acta Numer.* **2005**, 14, 363-444.

Table of Contents Graphic


$P^2 = P,$
$Tr(P) = N$
$-\nabla\Phi(P)$
Error
$\mathcal{S}$
$-\Gamma(P)$