Globally convergent trust-region methods for self-consistent field electronic structure calculations

Juliano B. Francisco[a)]
Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, Campinas SP, Brazil

José Mario Martínez[b)]
Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil

Leandro Martínez[c)]
Department of Physical Chemistry, IQ-UNICAMP, State University of Campinas, Campinas SP, Brazil

(Received 2 July 2004; accepted 21 September 2004)

As far as more complex systems are being accessible for quantum chemical calculations, the reliability of the algorithms used becomes increasingly important. Trust-region strategies comprise a large family of optimization algorithms that incorporates both robustness and applicability for a great variety of problems. The objective of this work is to provide a basic algorithm and an adequate theoretical framework for the application of globally convergent trust-region methods to electronic structure calculations. Closed shell restricted Hartree–Fock calculations are addressed as finite-dimensional nonlinear programming problems with weighted orthogonality constraints. A Levenberg–Marquardt-like modification of a trust-region algorithm for constrained optimization is developed for solving this problem. It is proved that this algorithm is globally convergent. The subproblems that ensure global convergence are easy-to-compute projections and are dependent only on the structure of the constraints, thus being extendable to other problems. Numerical experiments are presented, which confirm the theoretical predictions. The structure of the algorithm is such that accelerations can be easily associated without affecting the convergence properties.
 [DOI: 10.1063/1.1814935]

I. INTRODUCTION

Electronic structure calculations are being used in an increasingly large number of research fields. Several well-developed computer packages are available, which provide a large scope of algorithms and analytic tools in such a way that it is not required that the users fully understand the methods for obtaining valuable results. For this to happen, it has been necessary that the algorithms involved become faster, user-independent, and reliable. The basis of most electronic structure computations are the fast and inexpensive self-consistent field (SCF) algorithms. The ones based on Hartree–Fock (HF) and Kohn–Sham density functional theories are the most popular.[1,2] Since these problems are nonlinear, the algorithms require iterative updating of the variables (density matrix or eigenvectors of the Fock matrix) until self-consistency is achieved and hence a solution is found.

The first method designed to solve the HF problem was based on a naive fixed-point iteration that consists in the construction of the Fock matrix from the current guess followed by its diagonalization in order to obtain the new set of orbitals. This method has slow and unstable convergence properties and, thus, is no longer used for practical purposes. However, some of the methods currently used still rely on the fixed-point iteration in some way. Particularly, the direct inversion of the iterative subspace method of Pulay (DIIS) (Ref. 3) was designed to stabilize the fixed-point iterations by an extrapolation of the Fock matrix that aims to minimize residual vector norms. Although the DIIS method is also used in our days in different contexts,[4] it is in its original form that it is most frequently employed.[5,6]

Other techniques were proposed to improve convergence of SCF iterations.[3,4,7–11] In the level-shift method (the first algorithm claimed to have unconditional convergence properties[12]) the virtual orbitals are shifted to higher energies. This method depends on a user specified parameter which can be obtained only by trial and error.[4,13] The second-order SCF method (Ref. 14) relies on an exponential parametrization of the energy as a function of the density matrix. The energy minimization problem becomes unrestricted and Newton's method for unconstrained optimization is used. This method has robust convergence properties, but it relies on the availability of the Hessian, which is computationally very expensive. Other methods based on the exponential parametrization of the energy have been proposed, but global convergence is hard to obtain since matrix exponentials are not computed exactly.[4]

Let us recall here the meaning of *global convergence* in ordinary optimization literature. An algorithm is said to be

[a)]Electronic mail: juliano@ime.unicamp.br
[b)]Electronic mail: martinez@ime.unicamp.br
[c)]Electronic mail: lmartinez@iqm.unicamp.br

globally convergent if the sequences that it generates either stop at a point that satisfies first-order optimality conditions (usually called *stationary point*) or have the property that all its limit points are stationary. This property must hold independently of the initial starting point and, as much as possible, independently of any additional property of the sequence. In practical terms, this means that given a small tolerance $\varepsilon > 0$, any sequence generated by a globally convergent algorithm necessarily stops at a point that satisfies the convergence criterion defined by the tolerance. It must be warned that, since stationary points are not necessarily global minimizers, global convergence does not imply convergence to the global minimum of the optimization problems.

Aiming a fully reliable, user independent method, Cancès and Le Bris developed an optimal damping algorithm for which they proved global convergence whenever the iterates satisfy a *uniform well-posedness* (UWP) assumption.[15,16] Coupled with DIIS, this method provides more robust convergence than DIIS alone and preserves competitive convergence rates.[17] More recently, Thogersen *et al.*[18] introduced a trust-region method (TRSCF) for optimizing the total energy E_{SCF} of Hartree–Fock theory and Kohn–Sham density-functional theory. Trust-region methods (see Ref. 19) for unconstrained optimization were suggested by Powell[20] in 1970. The most complete global convergence proof of the Newton-based trust-region algorithm for unconstrained problems was given by Sorensen[21] in 1982. At each iteration of a trust-region algorithm one minimizes a quadratic approximation of the objective function on a ball centered in the current point and defined by some (not necessarily Euclidian) distance. If the reduction of the objective function obtained in this way is of the same order as the reduction of the quadratic approximation, the trial point is accepted as new iterate. Otherwise, the radius of the trust-region ball is reduced and the quadratic approximation is minimized on the new restricted region. If the gradient of the objective function at the current point is equal to the gradient of the quadratic model and the current point is not stationary, the iteration necessarily finishes successfully and, so, a better iterate is obtained. According to the degree of success of the iteration, the radius of the next trust region is increased or decreased. Many authors used trust region algorithms for solving minimization problems with simple constraints but, only in 1995, Martínez and Santos[22] gave a complete global convergence theory for feasible trust-region methods with arbitrary (possibly nonlinear and nonconvex) constraints. In Ref. 23 the same authors completed the theory with local and convergence-rate results.

The TRSCF algorithm of Thogersen *et al.* exhibits very nice practical behavior. In this algorithm the trust region at each iteration is not defined as the intersection of a compact *ball* with the feasible region (as in Ref. 22) but as the intersection of a linear half-space (defined in terms of the density variables) with the feasible set. When the reduction of the energy function E_{SCF} is not satisfactory, the frontier of the half-space (an hyperplane) is moved towards the current point so that, ultimately, energy decrease is obtained. The solution of the trust-region subproblem is always seeked on the hyperplane and it involves the diagonalization of a level-shift augmentation of the Fock matrix. Many implementation details are discussed in Ref. 18. The nonstandard trust-region approach of Ref. 18 makes it difficult to develop a rigorous convergence proof for this algorithm. The existence of such a proof remains a challenging open problem.

Although developed independently, the present research is related to, and, in some sense, complements the results of Ref. 18.

Here we introduce a new trust-region optimization algorithm with proved global convergence for closed shell restricted Hartree–Fock SCF iterations without UWP or related assumptions on the sequence itself. The classical fixed-point step is naturally incorporated to its structure. The first procedure at each iteration of the trust-region method will be to minimize a quadratic approximation of the energy function. We give a simple proof that a solution of this subproblem is given by the classical fixed-point iteration. The main ingredient for obtaining global convergence is not the choice of the first trial point at each iteration (which may lead to steps that are too long to be trusted, as observed in Ref. 18) but the choice of restricted steps after a possible failure of the first trial. (The first SCF trial step can even be skipped without violating global convergence.) The restricted steps after a possible failure of the first trial come from the minimization of a new quadratic model with a simplified block-diagonal Hessian motivated by the classical Barzilai–Borwein or spectral choice of the steplength in numerical optimization.[24–30] We will explain in which sense this is a Hessian approximation. The computation of this restricted step is cheap in the sense that it requires the diagonalization of a matrix whose dimension is of the order of the number of occupied molecular orbitals only. A key point of our algorithm is that, after each main iteration, an acceleration procedure is admissible with the sole requirement that it does not increase the value of E_{SCF}. In our experiments we use DIIS as accelerating algorithm but, of course, any other acceleration procedure is admissible such as, for example, the recently proposed successful EDIIS (Ref. 17) or TRSCF (Ref. 18) algorithms. In this sense, our algorithm may also be interpreted as a simple way to provide global convergence to methods that are known to be efficient in most cases. Summing up, the objective of this paper is to introduce a trust-region algorithmic framework for SCF electronic structure calculations with the following characteristics:

(1) Rigorous global convergence independently of the initial point.

(2) The structure of the iterations that ensure global convergence is independent of the structure of the objective function, so this type of iterations is applicable to other problems with similar constraints.

(3) The algorithm is such that the association with other efficient methods is straightforward.

(4) Experiments will show that, in practice, the algorithm behaves as predicted by theory.

This paper is organized as follows: In Sec. II we describe the general lines of the forthcoming trust-region algorithm and the main features of its implementation. In Sec. III we recall definitions and properties of the problem and we give a simple proof that the first subproblem solved at each itera-

tion coincides with the fixed-point iteration. In Sec. IV we briefly describe the resolution of simple quadratic trust-region problems and we state the fact that these solutions are easily computed nonlinear projections. The rigorous definition of the trust-region algorithm for our problem is given in Sec. V. In Sec. VI we describe the numerical experiments and in Sec. VII we state conclusions and lines for future results. The appendixes contain a rigorous convergence proof for the algorithm and the justification for the nonlinear projection procedure used in reduced trust regions.

II. ALGORITHMIC OVERVIEW

The algorithm presented here is a trust-region method.[19] The main iteration uses a quadratic approximation of the objective function around the current iterate and minimizes this quadratic model subject to the problem constraints (in this case, weighted orthonormality constraints). Once the quadratic model is minimized, a new *trial point* is obtained. Then, we test whether the decrease of the objective function at the trial point (*actual reduction*) is meaningful when compared to the reduction of the quadratic model (*predicted reduction*). Of course, the predicted reduction will be similar to the actual reduction whenever the quadratic model is a good approximation of the objective function. If the actual reduction is at least a given fraction of the predicted reduction, the trial point is accepted and the trust-region iteration finishes.

The energy at the trial point obtained by the minimization of the model may be higher (or, perhaps, not sufficiently lower) than the energy at the current iterate. In this case, the trial point is not accepted. Consequently, the algorithm proceeds minimizing a simple quadratic model of the energy in a smaller trust region around the current point. This process is repeated and, if the trust region is small enough, the decrease of the true energy becomes of the same order as the decrease of the quadratic model energy.

A trust-region method for arbitrary constraints[22] is computationally implementable when a meaningful quadratic model is easy to obtain, its minimum subject to the constraints of the problem is computable and minimizers of a suitable quadratic model subject to the problem constraints and smaller trust regions are also easy to compute. These conditions may not be easily fulfilled. For example, the quadratic model could be the complete second-order Taylor expansion, but this would require the computation of the Hessian, which may be very costly. Moreover, it is very difficult to compute a global minimum of the second-order Taylor model subject to orthonormality constraints. Finally, there do not exist practical methods for computing minimizers of arbitrary quadratic models subject to problem constraints *and* trust regions.

The algorithm presented in this paper provides suitable solutions for these difficulties. We show the following:

(1) The classical fixed-point iteration is the global minimization of a meaningful quadratic model of the energy subject to orthonormality constraints. Therefore the first step at each iteration of our trust-region method coincides with the classical fixed-point iteration.

(2) Global minimizers of a simplified quadratic model subject to orthonormality constraints *and* a smaller trust region are easy-to-compute projections. Therefore, the iterations that guarantee global convergence can be computed accurately in reasonable time.

III. THE FIXED-POINT ITERATION AS THE SOLUTION OF A QUADRATIC MODEL

A typical iteration of a trust-region method of the family introduced in (Ref. 22) begins by the minimization of a quadratic model of the objective function on the feasible region under consideration. In this section we will show that, in the case of restricted Hartree–Fock calculations, such minimization is accomplished by the classical fixed-point iteration.

The classical definition of the Hartree–Fock problem is as follows.[1,13] Let $2N$ and M be the number of electrons and nuclei in the system, respectively. We call H and S the core Hamiltonian and overlap matrices, respectively.[1]

Given K, the number of functions of the basis set X is the $K\times N$ matrix of coeficients for the expansion of the occupied molecular orbitals in terms of atomic orbitals. The closed-shell restricted Hartree–Fock energy is given by

$$E_{\mathrm{SCF}}(X)=\sum_{j=1}^{N} X_j^T[F(X)+H]X_j,$$

where $F(X)$ is the Fock matrix [see Appendix A].[1,13]

We consider the optimization problem

$$\text{Minimize } E_{\mathrm{SCF}}(X) \text{ subject to } X\in\Omega\subset\mathbb{R}^{K\times N}, \tag{1}$$

where Ω is the set of matrices of K rows and N columns whose columns satisfy the weighted orthonormality conditions $X_i^T S X_j=\delta_{ij}$.

Suppose that $\overline{X}\in\Omega$ is the current approximation to the solution of Eq. (1). In order to obtain an even better approximation, we are going to define a *quadratic model* of $E_{\mathrm{SCF}}(X)$. This quadratic model, denoted by $\mathcal{Q}(X)$, will be a good approximation of $E_{\mathrm{SCF}}(X)-E_{\mathrm{SCF}}(\overline{X})$ in a neighborhood of $\overline{X}$. We define

$$\mathcal{Q}(X)=4\sum_{j=1}^{N}(X_j-\overline{X}_j)^T F(\overline{X})\overline{X}_j + \frac{1}{2}\sum_{j=1}^{N}(X_j-\overline{X}_j)^T 4F(\overline{X})(X_j-\overline{X}_j). \tag{2}$$

The first derivatives of $E_{\mathrm{SCF}}(X)$ are

$$\frac{\partial E_{\mathrm{SCF}}(X)}{\partial X_j}=4F(X)X_j \tag{3}$$

and coincide with the derivatives of $\mathcal{Q}(X)$ when $X=\overline{X}$. The second derivatives of $E_{\mathrm{SCF}}(X)$ are hard to compute, so they are replaced in Eq. (2) by a simplification suggested by Eq. (3). The simplification comes from differentiation of both sides of Eq. (3) using the product rule $[(uv)'=u'v+uv']$ and neglecting, in the product formula, the term that involves derivatives of $F(X)$. Although, as pointed out in Ref. 18, this may represent a rather rough approximation of the true Hessian, very good Hessian approximations may not be necessary at all in trust-region calculations due to the necessity of

performing large steps when we are far from the solution.[19] In any case, as we will see later, global convergence properties do not depend of this specific choice of the model. Implementations of the main algorithm skipping this step are admissible.

The SCF problem consists on finding a set of vectors $X_1,\ldots,X_N$, which are generalized eigenvectors of the Fock matrix calculated from the same set of vectors. Thus, a solution of the SCF problem satisfies

$$F(X)X_j=\lambda_j SX_j \quad \forall j=1,\ldots,N. \tag{4}$$

The matrix X that fulfills this condition is called a *Fock fixed point*. If the generalized eigenvalues $\lambda_1,\ldots,\lambda_N$ corresponding to the eigenvectors $X_1,\ldots,X_N$ are the N smallest eigenvalues of $F(X)$, X is called an Aufbau Fock fixed point.

The classical fixed-point iteration is suggested by the definition of Aufbau fixed points: Given $\overline{X}_1,\ldots,\overline{X}_N$, one calculates $X_1,\ldots,X_N$, the generalized eigenvectors corresponding to the N smallest eigenvalues of $F(\overline{X})$. Therefore, we obtain X such that $F(\overline{X})X_j=\lambda_j SX_j$ for all $j=1,\ldots,N$.

Given a current iterate $\overline{X}\in\Omega$, the first step of our trust-region algorithm will consist on the minimization of the quadratic approximation (2) on the feasible set Ω. Now, we give a simple proof that this model minimization corresponds to a fixed-point iteration.

Theorem 3.1. Assume that $\overline{X}\in\Omega$, *and* X_{FP}, *the matrix of the eigenvectors corresponding to the N smallest generalized eigenvalues of* $F(\overline{X})$, *is the fixed-point iterate. Then* X_{FP} *is a global solution of*

$$\text{Minimize } \mathcal{Q}(X) \text{ subject to } X\in\Omega. \tag{5}$$

Proof. By Theorem 1.2 of Ref. 31, the fixed-point iterate X_{FP} solves the problem

$$\text{Minimize } \sum_{i=1}^{N} 2X_i^T F(\overline{X})X_i \text{ subject to } X\in\Omega. \tag{6}$$

The objective function of Eq. (6) is quadratic and direct calculation shows that it has the same first and second derivatives as $\mathcal{Q}(X)$ at the current point $\overline{X}$. Therefore, the difference between $\mathcal{Q}(X)$ and the objective function of Eq. (6) is a constant. This implies that Eqs. (5) and (6) are equivalent problems. So, the fixed-point iterate X_{FP} is a solution of Eq. (5), as we wanted to prove. □

IV. MINIMIZING A QUADRATIC MODEL ON A SMALLER TRUST REGION

If the trial point X_{FP} computed by the fixed-point iteration is such that $E_{\mathrm{SCF}}(X_{\mathrm{FP}})$ is sufficiently smaller than the energy at the current point $E_{\mathrm{SCF}}(\overline{X})$ then X_{FP} (or, perhaps, an accelerated step) will be accepted as the new iterate of the trust-region algorithm. If this is not the case, a quadratic model of $E_{\mathrm{SCF}}(X)$ must be minimized on the intersection of the feasible set Ω with a suitable trust region. If the energy at the solution of the new subproblem is sufficiently smaller than $E_{\mathrm{SCF}}(\overline{X})$ then this solution (or an accelerated point) is accepted. Otherwise, the trust region is reduced again, and so on.

Consider the case in which $E_{\mathrm{SCF}}(X_{\mathrm{FP}})$ *is not* sufficiently smaller than $E_{\mathrm{SCF}}(\overline{X})$. In principle, we could minimize the same quadratic model $\mathcal{Q}(X)$ on the intersection of the feasible region Ω and a suitably small trust region. However, we have two strong reasons for proceeding in a different way. On one hand, the fact that X_{FP} failed to produce a good decrease of the energy makes one think that the model is not good enough for approximating the energy at this point, perhaps because the Hessian approximation is not good or because higher order terms are dominant. On the other hand, although minimizing $\mathcal{Q}(X)$ on Ω is straightforward, minimizing this quadratic on the intersection of Ω with a trust region is not simple at all. Let us recall that, in the classical framework of trust-region methods, a trust region of radius r is defined as the set of points whose distance to the current iterate is less than or equal to r. In other words, a trust region is a *ball*, although not necessarily defined by the Euclidian distance.

Minimizing a quadratic model on the intersection of Ω with a trust region might be very difficult. However, we will define the new quadratic model and the new trust region radius in such a way that this solution is simple. The new quadratic model has the same first derivatives as the one defined by Eq. (2) but its second derivatives are different. Its definition is

$$\mathcal{Q}_{\mathrm{new}}(X)=4\sum_{j=1}^{N}(X_j-\overline{X}_j)^T F(\overline{X})\overline{X}_j + \frac{1}{2}\sum_{j=1}^{N}(X_j-\overline{X}_j)^T[\overline{\sigma}S](X_j-\overline{X}_j). \tag{7}$$

The first-order terms of $\mathcal{Q}_{\mathrm{new}}$ are the same as the ones of $\mathcal{Q}$ but the second-order terms are different. In the Hessian approximation used in Eq. (7) the matrix $4F(\overline{X})$ [used in Eq. (2)] is replaced by the $K\times K$ matrix $\overline{\sigma}S$.

In Eq. (7), the scalar $\overline{\sigma}$ is the so-called spectral coefficient,[24–30] the effect of which is that the matrix of second derivatives of the model is a simple approximation of the true Hessian of $E_{\mathrm{SCF}}(\overline{X})$. After the definition of Algorithm 5.1, with a more adequate notation, we will state the reasons why the Hessian of Eq. (7) is really a suitable Hessian approximation.[25,28,30] Roughly speaking, the Hessian approximation in Eq. (7) is the multiple of the matrix with diagonal blocks S which is closest to the true Hessian.

Even the minimization of Eq. (7) on the intersection of Ω and a given trust region might be difficult. For overcoming this difficulty, instead of minimizing explicitly the function $\mathcal{Q}_{\mathrm{new}}$ restricted to the trust region, we perform this task in an implicit way. Namely, we minimize the sum of $\mathcal{Q}_{\mathrm{new}}$ and a penalty term of the form $(t/2)\sum_{j=1}^{N}\overline{\sigma}(X_j-\overline{X}_j)^T S(X_j-\overline{X}_j)$ on the feasible set Ω *without* the explicit trust-region constraint. Fortunately, increasing t has the same effect as decreasing the trust-region radius. This process is illustrated in Fig. 1. In Fig. 1(a) we show, schematically, the rejected trial point X_{FP} and the result of solving Eq. (8) for $t=0$ [minimizer of Eq. (7) on Ω]. The center O of the level sets of $\mathcal{Q}_{\mathrm{new}}$ is the minimizer of $\mathcal{Q}_{\mathrm{new}}$ without the constraints Ω and can be obtained trivially solving a linear system. In Fig. 1(a)

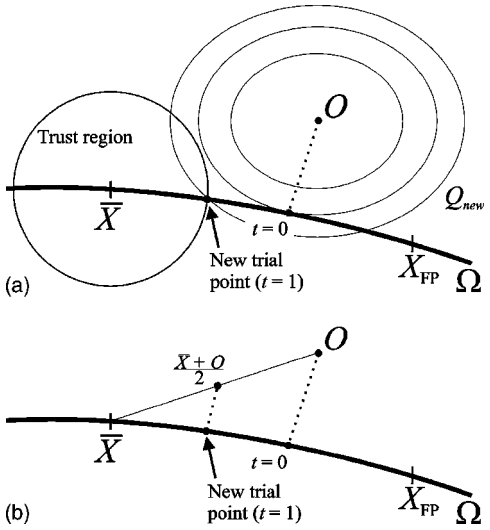

FIG. 1. The solution of the easy subproblems after the first rejected trial point. (a) The new trial point must be the minimizer $\mathcal{Q}_{\text{new}}$ on a smaller trust region. (b) This new trial point is obtained projecting $(\bar{X}+O)/2$ on the feasible set Ω.

we assume, further, that the trial point obtained with $t=0$ has been rejected [because its energy is not sufficiently smaller than $E_{\text{SCF}}(\bar{X})$] and, so, $\mathcal{Q}_{\text{new}}$ needs to be minimized on the intersection of Ω with a smaller trust region. The way in which this new trial point is obtained is shown in Fig. 1(b). The new trust-region radius is not considered explicitly. Instead, a point in the segment that joins the unconstrained minimizer of $\mathcal{Q}_{\text{new}}$ and $\bar{X}$ is computed and the new trial point is its projection on Ω.

The trust-region subproblem is then

$$\text{Minimize } 4\sum_{j=1}^{N}(X_j-\bar{X}_j)^T F(\bar{X})\bar{X}_j+\frac{1}{2}\sum_{j=1}^{N}(X_j-\bar{X}_j)^T \times[(1+t)\bar{\sigma}S](X_j-\bar{X}_j) \text{ subject to } X\in\Omega \tag{8}$$

and the algorithm sketched in Fig. 1 for solving this problem is given below.

Recall that the symmetric matrix S admits a diagonalization,

$$S=U_S\Sigma_S U_S^T,$$

where U_S is unitary (square with orthonormal columns and rows), its columns are eigenvectors of S, and Σ_S is the diagonal matrix whose entries are the eigenvalues of S. Since the eigenvalues are positive, we may define the square root of S,

$$S^{1/2}=U_S\Sigma_S^{1/2}U_S^T.$$

Consequently, we define $S^{-1/2}=[S^{1/2}]^{-1}$. Moreover, any $K\times N$ matrix Z admits a (reduced) singular value decomposition of the form

$$Z=U_Z\Sigma_Z V_Z^T,$$

where U_Z and V_Z are $K\times N$ and $N\times N$ matrices, respectively, with orthonormal columns and Σ_Z is a diagonal $N\times N$ matrix. This decomposition can be easily computed starting from the diagonalization of the $N\times N$ matrix Z^TZ, although more efficient methods exist.[32]

The procedure for computing the solution of Eq. (8) is described below in such a way that a computer code can be easily written. The justification to this procedure is given in Appendix C.

(1) Compute

$$\bar{Z}=\left(S^{1/2}-\frac{4}{\bar{\sigma}(1+t)}S^{-1/2}F(\bar{X})\right)\bar{X}.$$

(2) Compute $\bar{U}$ and $\bar{V}$ from the reduced singular value decomposition of $\bar{Z}$,

$$\bar{Z}=\bar{U}\bar{\Sigma}\bar{V}^T. \tag{9}$$

(3) Compute the solution of (8) as

$$X=S^{-1/2}\bar{U}\bar{V}^T.$$

V. FULL ALGORITHMIC DESCRIPTION

The main ingredients of the new algorithm for solving Eq. (1) were given in the preceding sections. In this section we give a more precise description of our method and we state its theoretical convergence properties.

A. Nonlinear programming problem

Let us express Eq. (1) as an optimization problem with vectors (instead of matrices) as unknowns. Define $n=KN$. For all $X\in\mathbb{R}^{K\times N}$, $X=(X_1,\ldots,X_N)$, we define the vector $\text{vec}(X)\in\mathbb{R}^{KN}$ by

$$\text{vec}(X)=\begin{pmatrix}X_1\\ \cdot\\ \cdot\\ \cdot\\ X_N\end{pmatrix}.$$

Consequently, we define $f[\text{vec}(X)]=E_{\text{SCF}}(X)$. Moreover, $\mathcal{R}$ will be the set of points in $\mathbb{R}^n$ such that the corresponding $K\times N$ matrix is in Ω. Then, the problem (1) can be written as

$$\text{Minimize } f(x) \text{ subject to } x\in\mathcal{R}\subset\mathbb{R}^n. \tag{10}$$

In other words, the problem (10) is exactly the same as the problem (1), where the matricial variables X are replaced by vectorial variables x.

The set $\mathcal{R}$ is compact (closed and bounded). Closedness means that limit points of sequences completely contained in $\mathcal{R}$ necessarily belong to $\mathcal{R}$. This property is essential when one discusses convergence of iterative methods since one wants to guarantee that, when a sequence is completely contained in the feasible set (Ω or $\mathcal{R}$ in this case) its limit points also belong to this set. The closedness of $\mathcal{R}$ comes from the fact that the constraints that define Ω are equations and nonstrict (''less-than-or-equal-to'') inequalities. The feasible set is also bounded because the constraints $X_j^TSX_j=1$ are bounded ellipsoids in $\mathbb{R}^K$. Compactness (closedness plus boundedness) implies that every sequence completely contained in $\mathcal{R}$ admits at least one limit point that belongs to $\mathcal{R}$. In the convergence theory we prove that every limit point is

stationary. So, since limit points exist, we will be able to conclude that the algorithm finds, ultimately, stationary points. Geometric insight on the feasible set Ω and on Newton and conjugate-gradient algorithms for minimization with this type of constraints has been given in Ref. 33.

Every local minimizer of Eq. (1) must satisfy the Lagrange optimality conditions.[34] This is because all the points in $\mathcal{R}$ are *regular* in the sense that the gradients of the constraints are linearly independent. As usually, points that satisfy the optimality conditions are said to be *stationary*. Stationary points can be transformed into Fock fixed points by multiplication by an unitary $N\times N$ matrix.[13]

B. Nonlinear programming algorithm

Here we define our trust-region method for solving Eq. (1). The iterates of the algorithm will be called X^k and the corresponding points $\mathrm{vec}(X^k)\in\mathbb{R}^n$ will be denoted x^k. Moreover, given $X\in\mathbb{R}^{K\times N}$ we denote $x=\mathrm{vec}(X)$.

By Eq. (3), we have

$$\nabla f(x)=\mathrm{vec}[4F(X)X]$$

and

$$g^k=\mathrm{vec}[4F(X^k)X^k] \quad \text{for all } k.$$

We define $A\in\mathbb{R}^{n\times n}$ and $\mathcal{H}_k\in\mathbb{R}^{n\times n}$ by

$$A=\begin{bmatrix} S & & \\ & \ddots & \\ & & S \end{bmatrix}, \quad \mathcal{H}_k=\begin{bmatrix} 4F(X^k) & & \\ & \ddots & \\ & & 4F(X^k) \end{bmatrix}. \tag{11}$$

Therefore, A and $\mathcal{H}_k$ are N-block-diagonal matrices with $K\times K$ blocks.

At some iterations we will use $\mathcal{H}_k$ as Hessian approximation, which corresponds to use the quadratic model $\mathcal{Q}(X)$. At other iterations we use the Hessian approximation $\sigma_k A$, where σ_k is the spectral parameter mentioned before, and corresponds to use the "easy" quadratic model $\mathcal{Q}_{\mathrm{new}}(X)$. We already know how to solve the subproblems associated to each quadratic model. Iterations where $\mathcal{H}_k$ is the Hessian approximation will be said to be of *type 1*. Reciprocally, the iterations where the Hessian approximation is $\sigma_k A$ are said to be of *type 2*. Iterations of types 1 and 2 can be chosen in two basic ways:

(1) At its beginning, each iteration is always of type 1. This means that the model $\mathcal{Q}(X)$ is used. If X_{FP} is good enough, then this point (or an accelerated one) is accepted as a new iterate. On the contrary, if $E(X_{\mathrm{FP}})$ is not sufficiently smaller than the energy at the current point, the iteration is changed to be of type 2. This scheme corresponds essentially to the process sketched in previous sections when we defined $\mathcal{Q}$ and $\mathcal{Q}_{\mathrm{new}}$. It corresponds to choose $\mathrm{type}(0)=1$ and $\mathrm{type}(k)=1$ in Eqs. (12) and (19) below.

(2) In an alternative version of the algorithm all the iterations may be of type 2. This corresponds to choose $\mathrm{type}(0)=2$ and $\mathrm{type}(k)=2$ in Eqs. (12) and (19) below.

Algorithm 5.1 is stated in such a general way that other strategies are possible to choose the type of each iteration. For example, if iterations of type 1 systematically fail we may decide not to use them anymore, choosing always $\mathrm{type}(k)=2$. The numerical experiments will correspond to the first strategy described above.

1. Algorithm 5.1

Step 1. Choose $\alpha\in(0,1/2)$, $0<\sigma_{\min}<\sigma_{\max}<\infty$ and the initial approximation $X^0\in\Omega$ [so, $x^0=\mathrm{vec}(X^0)\in\mathcal{R}$]. Set $k\leftarrow 0$, $\sigma_0=1$. Choose

$$\mathrm{type}(0)\in\{1,2\}. \tag{12}$$

Step 2.

If $\mathrm{type}(k)=1$, define $B_k=\mathcal{H}_k$.

If $\mathrm{type}(k)=2$ and $k>0$, compute σ_k, the spectral scaling parameter,[26,28] by

$$\sigma_k=\max\left\{\sigma_{\min},\min\left\{\sigma_{\max},\ \frac{(x^k-x^{k-1})^T(g^k-g^{k-1})}{(x^k-x^{k-1})^TA(x^k-x^{k-1})}\right\}\right\} \tag{13}$$

and

$$B_k=\sigma_k A. \tag{14}$$

Step 3. Set $t\leftarrow 0$.

Step 4. Define

$$Q_{k,t}(x)=(g^k)^T(x-x^k)+\tfrac{1}{2}(x-x^k)^T[B_k+t\sigma_k A](x-x^k). \tag{15}$$

Compute x_{trial}, a global solution of

$$\text{Minimize } Q_{k,t}(x) \text{ subject to } x\in\mathcal{R}. \tag{16}$$

If $Q_{k,t}(x_{\mathrm{trial}})=0$, terminate the execution of the algorithm declaring that x^k (X^k) is stationary.

Step 5.

(1) If

$$f(x_{\mathrm{trial}})\leqslant f(x^k)+\alpha Q_{k,0}(x_{\mathrm{trial}}), \tag{17}$$

compute $x^{k+1}\in\mathcal{R}$ such that

$$f(x^{k+1})\leqslant f(x_{\mathrm{trial}}), \tag{18}$$

set $k\leftarrow k+1$, choose

$$\mathrm{type}(k)\in\{1,2\}, \tag{19}$$

and go to step 2.

If Eq. (17) does not hold, then

(1) If $type(k)=1$, redefine $type(k)=2$, and go to step 2.

(2) If $type(k)=2$, set $t\leftarrow\max\{1,2t\}$, and go to step 4.□

Algorithm 5.1 has been described in such a way that its implementation using the results on the solution of subproblems (Secs. III and IV) is not difficult. However, some additional explanation is necessary in order to make it more friendly.

(a) The requirement (17) states that the *actual reduction* $f(x^k)-f(x^{\mathrm{trial}})$ should be, at least, a fraction α of the *predicted reduction* $-Q_{k,0}(x_{\mathrm{trial}})$. According to Eq. (15), one is using the approximation

$$f(x) \approx f(x^k) + (g^k)^T(x-x^k) + \tfrac{1}{2}(x-x^k)^T B_k(x-x^k)$$
$$= f(x^k) + Q_{k,0}(x).$$

Observe that $Q_{k,0}(x^k)=0$. Since this approximation is correct up to the first-order terms, it is justified to require that the reduction of the true objective function should be of the same order as the reduction of the quadratic model. The parameter α quantifies the degree of agreement between both reductions. The first-order coincidence guarantees that, unless x^k is a stationary point, the condition (17) will hold if x_{trial} is close enough to x^k. The distance between the trial point x_{trial} and the current point x^k is controlled by the trust-region radius in classical trust-region methods and by the regularizing parameter t in Algorithm 5.1.

(b) At iterations of type 2 the Hessian approximation is $\sigma_k A$. This matrix is the multiple of A which, in some sense, is closest to the true Hessian $H_f(x^k)$. Roughly speaking, we have

$$g^k - g^{k-1} \approx H_f(x^k)(x^k - x^{k-1}),$$

so, premultiplying by $(x^k-x^{k-1})^T$, we obtain

$$(x^k-x^{k-1})^T(g^k-g^{k-1}) \approx (x^k-x^{k-1})^T H_f(x^k)(x^k-x^{k-1}).$$

Dividing by $(x^k-x^{k-1})^T A(x^k-x^{k-1})$, we get

$$\frac{(x^k-x^{k-1})^T(g^k-g^{k-1})}{(x^k-x^{k-1})^T A(x^k-x^{k-1})} \approx \frac{(x^k-x^{k-1})^T H_f(x^k)(x^k-x^{k-1})}{(x^k-x^{k-1})^T A(x^k-x^{k-1})}. \quad (20)$$

Making the (obviously wrong) simplification of the terms $(x^k-x^{k-1})^T$ and (x^k-x^{k-1}) on the right-hand side of Eq. (20) we obtain

$$H_f(x^k) \approx \frac{(x^k-x^{k-1})^T(g^k-g^{k-1})}{(x^k-x^{k-1})^T A(x^k-x^{k-1})} A.$$

This justifies the choice (13) except for the fact that, to prevent numerical instabilities, we require that the coefficient σ_k should belong to the closed interval $[\sigma_{\min}, \sigma_{\max}]$. In turn, $\sigma_{\min}$ and $\sigma_{\max}$ are parameters given by the user. If the quotient $(x^k-x^{k-1})^T(g^k-g^{k-1})/(x^k-x^{k-1})^T A(x^k-x^{k-1})$ lies outside the interval $[\sigma_{\min}, \sigma_{\max}]$, formula (13) forces σ_k to be one of the extremes of this interval. A more careful motivation of the spectral coefficient based on mean-value arguments (21) is given below.

(c) At iterations of type 1, the solution of the subproblem (16) is given by the fixed-point iteration. At iterations of type 2 the solution of the subproblem is given by the procedure described in Sec. IV. When the actual reduction defined by the trial point is not enough, the value of t is increased. (The new t is set to be equal to the maximum between 1 and $2t$ at the end of step 5.) The effect of increasing t is the same as the effect of reducing the trust-region radius.

(d) When the trial point (coming from any type of iteration) satisfies the sufficient descent condition (17), the new iterate may be any point satisfying Eq. (18). Clearly, the choice $x^{k+1}=x_{\text{trial}}$ is admissible, since $x=x_{\text{trial}}$ obviously satisfies Eq. (18). However, the weak requirement (18) allows one to choose x^{k+1} by means of acceleration procedures such as DIIS, or even TRSCF. By Eq. (18), the energy at the accelerated point only needs to be not greater than the energy at the trial point. The convergence proofs are not affected at all by the specific choice of x^{k+1}, provided that the condition (18) is satisfied.

(e) If $Q_{k,t}(x_{\text{trial}})=0$ after solving Eq. (16) then, since $Q_{k,t}(x^k)=0$, it turns out that x^k is a global solution of Eq. (16). Then, x^k satisfies the Lagrange optimality conditions of Eq. (16). But these conditions are exactly the Lagrange optimality conditions of the original problem (10) due to the fact that the first-order terms of f and $Q_{k,t}$ are the same. Therefore, x^k is a stationary point, which justifies terminating the execution of the algorithm in this case.

In subproblems of type 2 the Hessian approximation is defined by Eqs. (11), (13), and (14). Let us give here a more careful argument to show that Eq. (14) in fact defines a Hessian approximation. Let us recall[28] the mean-value formula

$$(x^k-x^{k-1})^T(g^k-g^{k-1}) = (x^k-x^{k-1})^T\left[\int_0^1 H_f[x^{k-1}+v(x^k-x^{k-1})]dv\right] \times (x^k-x^{k-1}), \quad (21)$$

where, as above, $H_f(x)$ denotes the Hessian of f. Therefore, the coefficient σ_k is the factor by which it is necessary to multiply the matrix A to become similar to the *average Hessian* $[\int_0^1 H_f[x^{k-1}+v(x^k-x^{k-1}]dv]$. This is exactly what we do in Eq. (14). Rigorous analysis of methods exclusively based on this approximation may be found in Refs. 25, 26, 28.

C. Convergence

In Appendix B we prove that Algorithm 5.1 is globally convergent without any additional assumption on the generated sequence $\{X^k\}$. This means the following:

(1) The algorithm terminates at an iteration k only if X^k is a stationary point. (Therefore, a set of N generalized eigenvectors can be immediately obtained from X^k.)

(2) The iterations of Algorithm 5.1 are well defined in the sense that each iteration necessarily finishes in finite time if X^k is not stationary.

(3) Any sequence generated by Algorithm 5.1 necessarily admits limit points and all the limit points are stationary. Therefore, approximate Fock fixed points can be obtained up to any desired precision.

Reading the steps of Algorithm 5.1, we observe that the algorithm terminates at an iteration k only when, at that iteration, $Q_{k,t}(x_{\text{trial}})=0$. In this case, as we mentioned before, $Q_{k,t}(x^k)=0$ and, so, x^k is a global minimizer of the subproblem (16) and satisfies the Lagrange optimality conditions. Of course, in computer implementations, a more tolerant stopping criterion is used.

Let us state now the main ingredients of the proof that the algorithm is well defined and globally convergent. Rigorous mathematical details are given in Appendix B. We say than an algorithm is well defined when each iteration necessarily finishes; that is, infinite loops within a particular iteration cannot occur. We only need to consider the case in

which the iterate x^k is not a stationary point. In this case, x^k is not a stationary point of the quadratic $Q_{k,t}$ either. Therefore, since x_{trial} is a global minimizer of $Q_{k,t}$ and $Q_{k,t}(x^k) = 0$, we have that $Q_{k,t}(x_{\text{trial}})$ is negative for all $t \geqslant 0$. Then, by the definition (15), $Q_{k,0}(x_{\text{trial}})$ is always negative. But the first-order terms of $Q_{k,0}(x)$ are the same as the first-order terms of $f(x)-f(x^k)$, therefore the fact that $Q_{k,0}(x_{\text{trial}})$ is negative forces that $f(x)-f(x_k)$ is also negative (with the same order of its linear and quadratic approximations) if x_{trial} is close enough to x^k. Since the effect of the penalty regularizing parameter t is to reduce the distance between x_{trial} and x^k, it turns out that, for t large enough, the sufficient descent condition (17) necessarily holds. This means that the iteration will finish after increasing t a finite number of times.

Now let us give the main ideas of the global convergence proof. Note that the existence of limit points is guaranteed by the compactness of $\mathcal{R}$, so it must only be justified the fact that every limit point is stationary.

In Proposition B.1 we will prove that the subproblem (16) is equivalent to the following trust-region subproblem

$$\text{Minimize } Q_{k,0}(x) \text{ subject to } x \in \mathcal{R},$$

and

$$(x-x^k)^T A(x-x^k) \leqslant \Delta$$

for an adequate trust-region radius Δ that depends on t. When t is increased, Δ decreases and Δ tends to zero when t tends to infinity. The restriction $(x-x^k)^T A(x-x^k) \leqslant \Delta$ is a typical trust-region constraint in the sense that it defines a *ball* with respect to the distance defined by the matrix A. (Balls with respect to this distance are ellipsoids in $\mathbb{R}^n$.) Due to this equivalence, the convergence theory of the algorithm is essentially reduced to the theory of convergence of trust-region methods on arbitrary domains given in Ref. 22. Several technical aspects of this equivalence are given in Appendix B.

The main ideas in the theory of convergence of trust-region methods in arbitrary domains[22] are the following:

(1) At each iteration a quadratic model of the objective function is minimized on the intersection of a (not necessarily Euclidian) ball and the feasible region. The initial trust-region radius at each iteration must be greater than a fixed radius $\Delta_{\min}$. If the trial point so far obtained is such that the decrease of the objective function is proportional to the decrease of the quadratic model, the trial point is accepted as new iterate. Otherwise, the trust-region radius is decreased. For global convergence it is essential that the amount of actual decrease must be proportional to the decrease of the quadratic model. The fact that $f(x^{k+1}) < f(x^k)$ is not enough to guarantee convergence to stationary points since the sequence might approach indefinitely to a nonstationary point in spite of monotone decrease of the objective function.

(2) For proving global convergence, an arbitrary sequence of iterates generated by the algorithm is considered. We deal with the theoretical case where there is no tolerance for the detection of a stationary point. In such a case, either the sequence stops abruptly when a stationary point is *exactly* found or the sequence has infinitely many terms. We want to prove that the limit point of any convergent subsequence, which we call x^*, must be a stationary point. Recall that for each accepted point x^k belonging to the sequence, there exists a final trust-region radius Δ_k and, therefore, there is also a sequence of accepted trust-region radii associated with the subsequence of iterates under consideration. Let us assume, by contradiction, that the limit point x^* is not stationary. There are two possibilities for the trust-region radii sequence:

(a) *The trust-region radius Δ_k tends to zero*: The initial trust-region radius of each iteration is greater than a fixed quantity $\Delta_{\min}$. Therefore, if the accepted trust-region radius Δ_k tends to zero, there exists also a sequence of nonaccepted trust-region radii that tends to zero. This means that even for arbitrarily small trust-regions, the actual reduction of the objective function would still be too small when compared to the predicted reduction of the quadratic model. This is impossible if x^* is not stationary, since the actual and predicted reductions must be similiar for very small trust region radii (the quotient between them must converge to one).

(b) *A subsequence of trust-region radius Δ_k is bounded away from zero*: Recall that the predicted reduction of the quadratic model around a nonstationary point is positive whenever the trust-region is greater than zero. Furthermore, the algorithm requires a constant proportionality between the actual reduction and the predicted reduction in order to accept a trial point. Therefore, the actual reduction of the objective function must also be positive for an iterate to be accepted. Then, since the subsequence of trust-region radii is bounded away from zero, the predicted and actual reductions are also positive and bounded away from zero infinitely many times. This implies that the objective function value tends to $-\infty$, which is impossible since we assumed that the sequence tends to x^* and, then, the function value must tend to $f(x^*)$.

Hence, we have that both alternatives (a) and (b) are false. This is a contradiction that raised from the assumption that x^* is not stationary. Therefore, any limit point of a sequence of iterate s generated by this algorithm must be stationary.

VI. NUMERICAL EXPERIMENTS

The (unaccelerated) GTR (global trust-region) method is given by Algorithm 5.1 with the choice $x^{k+1} = x_{\text{trial}}$ in Eq. (18). In the accelerated version of the algorithm we take advantage of the freedom implicit in Eq. (18) and we choose x^{k+1} as an accelerated step that uses the previous iterates to improve x_{trial} minimizing a residual approximation on an appropriate subspace. In the experiments, we incorporate the DIIS acceleration scheme to the basic structure of GTR. The resulting algorithm will be called GTR+DIIS. As stated in Sec. V and proved in Appendix B, the theoretical convergence properties of GTR and GTR+DIIS are the same. In both cases limit points are (not necessarily Aufbau) Fock fixed points and the tendency to converge to Aufbau points comes from the fact that the first step of each iteration is the classical fixed-point iteration.

In GTR+DIIS the acceleration is used from the second iteration on. Therefore, the first extrapolation uses two re-

TABLE I. Geometry parameters of the molecules used in the examples.

Molecule	Geometry		
	Bond length (Å)	Angle	Dihedral
CrC	2.00		
Cr_2	2.00		
CO	1.40		
CO(Dist)	2.80		
H_2O	0.95(OH)	109°(HOH)	
NH_3	1.008(NH)	109°(HNH)	120°(HNHH)

siduals. In the subsequent iterations the number of interpolating residuals is increased up to a maximum of 10. From then on, ten residuals are used. Moreover, residuals that correspond to points where energy increases are discarded for extrapolation purposes.

The classical fixed-point method will be called FP and its acceleration using DIIS (with the same number of residuals as GTR+DIIS) will be called, simply, DIIS.

The algorithmic parameters used were $\sigma_{\min}=0.01$, $\sigma_{\max}=100$, $\sigma_0=0.5$, and $\alpha=10^{-4}$. The algorithms were stopped when the relative difference between two consecutive energies was smaller than 10^{-9}.

We used different types of initial points: diagonalized core Hamiltonians H, Huckel guesses provided by the GAMESS (Ref. 5) package for the same problem and (in some cases) the initial approximation induced by the Identity matrix is employed. These different initial approximations were chosen because they can be easily reproduced.

We used, for our tests, molecules with the geometries specified in Table I. The molecules CrC and Cr_2 are known as having unstable convergence properties.[4,15,17] Two CO geometries were chosen as examples since it is known that distorted geometries cause convergence difficulties.[13] Finally, water and ammonia examples were introduced to illustrate how the trust-region algorithm behaves in situations where the classical algorithms are successful.

A. Results

Table II shows that the number of iterations performed by FP and GTR on one side, and by DIIS and GTR+DIIS on the other side are the same for the water and ammonia examples. This is due to the fact that both the fixed-point iterations and the DIIS extrapolations are always successful in providing new trial points with a significantly lower energy. In that case, the reduction of the trust region is never needed and therefore the trust-region algorithms behave exactly as the supporting methods.

For the CO molecule with a STO-3G basis the classical FP method always fails to converge. The energy oscillates until the maximum number of iterations (5001) is achieved. For this example the DIIS method is very efficient, converging from any initial point in at most 11 iterations. The GTR method also converges in all cases, as expected, but it takes almost twice the number of iterations as DIIS and converges to a solution that does not satisfy the Aufbau principle when the initial point was derived from the Identity matrix. Finally, the accelerated GTR+DIIS method converges rapidly and with a few less iterations than DIIS, always to solutions that satisfy the Aufbau principle. In the distorted CO molecule the robustness of the trust-region algorithms becomes better illustrated. The FP method fails to converge in all cases. The DIIS method converges in 117 iterations to a point higher in energy than the solution found in 12 and 10 iterations by the

TABLE II. Number of iterations performed by each algorithm in some test problems. FP: classical fixed-point algorithm; DIIS: the DIIS acceleration of Pulay; GTR: the global trust-region algorithm without acceleration; GTR+DIIS: the new trust-region algorithm accelerated by DIIS.

Molecule	Basis	Initial point	Algorithm			
			FP	DIIS	GTR	GTR+DIIS
H_2O	STO-3G	H^{core}	7	5	7	5
	6-31G	H^{core}	18	8	18	8
NH_3	STO-3G	H^{core}	8	7	8	7
	6-31G	H^{core}	14	7	14	7
CO	STO-3G	H^{core}	X[a]	11	22	10
		Huckel	X[a]	7	16	7
		Identity	X[a]	11	17[b,c]	9
CO(Dist)	STO-3G	H^{core}	X[a]	117[c]	12	10
		Huckel	X[a]	85	13	15
	6-31G	H^{core}	X[a]	27[c]	158	115
		Huckel	X[a]	36[c]	384	59
Cr_2	STO-3G	H^{core}	52[c]	13	56	38
		Huckel	12[c]	33[c]	398	134
		Identity	7[c]	37	50[c]	26[c]
CrC	STO-3G	H^{core}	X[a]	X[a]	71[b,c]	29
		Huckel	X[a]	49	129	23
		Identity	X[a]	180	40[c]	36
	6-31G	H^{core}	X[a]	19	102[c]	29[c]
		Huckel	X[a]	52[c]	113[c]	37

[a]No convergence in 5001 iterations.
[b]Converged to a point with Aufbau principle violation.
[c]Converged to a higher energy than some of the other algorithms.

GTR and GTR+DIIS methods respectively when a STO-3G basis is used from a core Hamiltonian initial approximation. Using the Huckel approximation, DIIS converges to the lowest energy solution, but it takes 85 iterations against 13 and 15 iterations taken by GTR and GTR+DIIS, respectively. Finally, when using a larger 6-31G basis, DIIS converges fast but to points higher in energy than the ones obtained by the trust-region algorithms. We observe that the GTR method takes 384 iterations to converge from the Huckel initial approximation because its basic first-trial step is the classical fixed-point iteration which systematically fails for this problem.

For the Cr_2 molecule the DIIS method was more successful than the trust-region methods. We obtained convergence of all the instances, but the FP method converged to a point that lies 9.3 a.u. higher in energy than the solution found by the DIIS method. The differences in energy for the other solutions are of the order of 5×10^{-6} a.u. In these cases, the DIIS method converged in at most 37 iterations whereas 398 and 134 iterations were needed to achieve convergence for the GTR and GTR+DIIS methods respectively from the Huckel guess.

Finally, a very interesting test was provided by the CrC molecule. For the 6-31G basis, all but the FP methods converged. DIIS used fewer iterations when starting from the core Hamiltonian but more iterations than GTR+DIIS when starting from the Huckel guess. The pure GTR method employed significantly more iterations than both methods in all cases, and converged to a solution slightly higher in energy.

When using the STO-3G basis, the tests were more interesting and the results are highlighted in Fig. 2. Starting from the core Hamiltonian both FP and DIIS failed to converge, as can be seen in Fig. 2(a). The GTR method converged in 71 iterations to a higher-energy solution that does not satisfy the Aufbau principle and the GTR+DIIS method converged in 29 iterations to the lowest energy Aufbau solution. From the Huckel guess DIIS converged but not as fast as GTR+DIIS whereas GTR converged in significantly more iterations. See Fig. 2(b). Finally, from the Identity guess, DIIS oscillates at the beginning and stops oscillating probably thanks to numerical rounding errors. DIIS finally converges in 180 iterations, as shown in Fig. 2(c). GTR converges in 40 iterations to a solution higher in energy and GTR+DIIS method converges to the lowest energy solution in 36 iterations. This is an interesting example where the DIIS method fails to converge from one initial point while trust-region methods are successful.

It is worthwhile to highlight that the FP method fails to converge in 12 of 19 tests whereas the pure GTR method converged in all cases in spite of the fact that the first trial point computed at each iteration is identical to the fixed-point iteration. This fact illustrates the robustness of the trust-region strategy.

We note that for each iteration of the trust-region methods, more than one functional evaluation is needed when it is necessary to reduce the trust region. For this reason, in critical cases a small number of iterations of the trust-region method does not necessarily reflects a small computer time. However, since increasingly efficient linear-scaling procedures are continuously being developed for building Fock-matrices, reliability issues become more and more important.[17,35] We claim that trust-region methods as the ones introduced here could be used as an automatic alternative to provide convergence for difficult problems when divergence or oscillatory behaviors are detected in other algorithms.

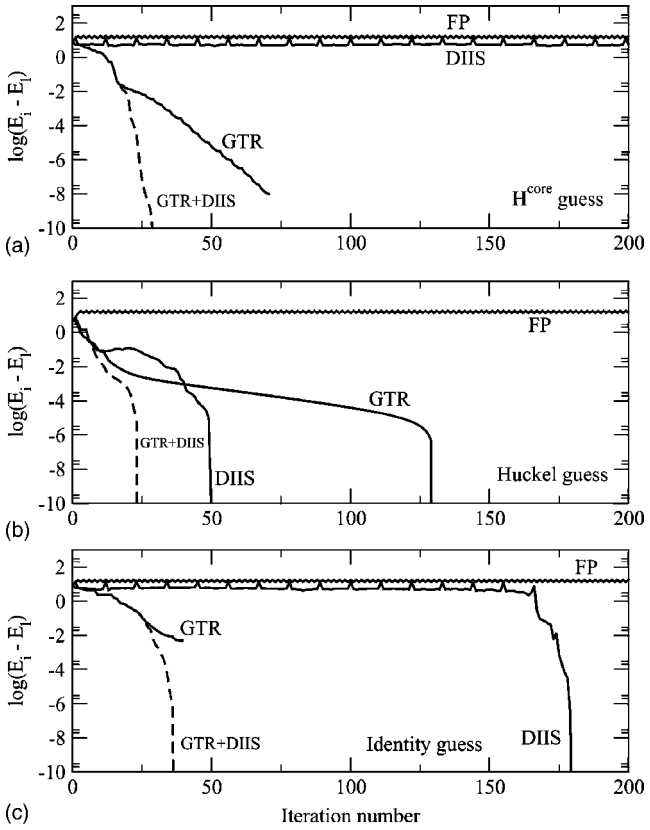

FIG. 2. Convergence behavior of the four methods for the CrC molecule using the STO-3G basis.

VII. CONCLUSIONS

We introduced a new trust-region algorithm for performing closed shell restricted Hartree–Fock electronic structure calculations. Global convergence was proved without any assumption on the sequence of iterates, thus showing that convergence must take place from any initial point. The method is also independent of any user-specified parameter. The trust-region method so far introduced uses the structure of the RHF problem to define the first trial point at each iteration, although this fact is not essential for global convergence properties. If the user chooses $type(k)=2$ for all k when running Algorithm 5.1, a globally convergent algorithm is also obtained. The resolution of the subproblems associated with the reduction of the trust region are easy-to-compute projections. Due to these algorithmic features, the trust-region method is implementable. Numerical experiments show that the method is robust and behaves as predicted by theory. Convergence is obtained in all the examples in spite of the fact that the naive fixed-point iteration is used as the first step of the trust-region iteration. The new method may be very useful when convergence failures of other algo-

rithms are detected thus providing reliability for routine RHF calculations. Any heuristic, case-oriented, nonconvergent or weakly convergent though efficient method may be used at the acceleration phase of our algorithmic framework without affecting the global convergence properties. In this sense, the GTR approach may also be interpreted, not as a competitor of other methods but a safeguarding procedure for guaranteeing global convergence. We showed that its association with DIIS is profitable and, certainly, we expect that its hybridization with other methods (especially those that produce feasible iterates, as TRSCF) may be efficient. This is a very important feature when one deals with systems with complex wave-functions.

A particular case of Algorithm 5.1 consists in taking $type(k)=2$ for all the iterations k. In this simplified version the Hessian approximations are always chosen as $B_k=\sigma_k A$. As a consequence, the subproblems (16) can always be solved using the (cheap) technique described in Sec. IV. The interesting fact about this version of the algorithm is that its implementation does not depend at all on the form of the objective function $f(x)$. In other words, it can be applied without modifications to any problem with weighted orthonormality constraints. Therefore, the algorithm may be applicable to unrestricted Hartree–Fock, configuration interaction, density functional theory and semiempirical methods without major modifications. Moreover, the dependence of Algorithm 5.1 with respect to the form of the constraints lies on the fact that we have a reliable method for solving the subproblems (16). The algorithm may also be applied to other types of constraints, provided that good methods for solving the subproblems are available.

We believe that the efficiency of the algorithm of Thogersen *et al.*[18] and the robustness and theoretical framework of our GTR approach provide a great support to the implementation and further development of trust-region strategies for electronic structure calculations.

ACKNOWLEDGMENTS

J.B.F. and J.M.M. were supported by PRONEX-optimization 76.79.1008-00, FAPESP (Grant No. 01-04597-4) and CNPq. L.M. was supported by FAPESP and by the Programa de Bolsas para Instrutores Graduados of the State University of Campinas. We also thank Luciano N. Vidal for providing the subroutine used for computing molecular integrals that is a key part of the package with which numerical tests were performed. The authors are deeply indebted to an anonymous referee whose comments helped us to improve the first version of the paper and who drew up our attention to Ref. 18.

APPENDIX A: DEFINITIONS AND NOTATION

1. The Fock matrix

Given the definitions of N and M given in Sec. III the precise dependence of the Fock matrix with respect to X is as follows:[1,13] Let $\{g_1,\ldots,g_K\}\subset C^2(\mathbb{R}^3)$ be a set of linearly independent functions (the basis set), with $K\geqslant N$. We assume that, for all $\mu,\nu,\sigma,\lambda\in\{1,\ldots,K\}$, the following quantities are well defined:

$$(\mu\nu|\sigma\lambda)=\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} g_\mu(r_1)g_\nu(r_1)\frac{1}{\|r_1-r_2\|}\times g_\sigma(r_2)g_\lambda(r_2)dr_1dr_2.$$

We define, for all μ, ν, σ, λ,

$$B^{\sigma\lambda}_{\mu\nu}=2(\mu\nu|\sigma\lambda)-(\mu\lambda|\sigma\nu).$$

For all $X\in\mathbb{R}^{K\times N}$ $(X=(X_{ij}))$ we define $G(X)\in\mathbb{R}^{K\times K}$ by

$$G(X)_{\mu\nu}=\sum_{b=1}^{N}\sum_{\sigma,\lambda=1}^{K} B^{\sigma\lambda}_{\mu\nu}X_{\sigma b}X_{\lambda b} \quad \text{for all } \mu,\nu=1,\ldots,K.$$

The Fock matrix $F(X)\in\mathbb{R}^{K\times K}$ is given by

$$F(X)=H+G(X), \tag{A1}$$

where H is the core Hamiltonian matrix with elements

$$H_{\mu\nu}=\int_{\mathbb{R}^3} g_\mu(r)[h(g_\nu)(r)]dr,$$

and the core Hamiltonian operator h is given by

$$h(\varphi)(r)=-\frac{1}{2}\nabla^2\varphi(r)-\sum_{j=1}^{M}\frac{Z_j}{\|r-\overline{r_j}\|}\varphi(r).$$

2. Notation

(1) If f is a real-valued function of n variables, we denote $g(x)=\nabla f(x)$ and $g^k=g(x^k)$ for $x\in\mathbb{R}^n$, $x^k\in\mathbb{R}^n$.

(2) $C^2(\mathbb{R}^3)$: the set of twice continuously differentiable functions $\varphi:\mathbb{R}^3\to\mathbb{R}$.

(3) The transpose of a real matrix A will be denoted A^T. The identity matrix will be denoted I. The Frobenius norm of A is denoted $\|A\|_F$.

(4) A square matrix C will be said *unitary* if $C^TC=CC^T=I$.

(5) δ_{ij} denotes the Kroenecker symbol. ($\delta_{ij}=1$ if $i=j$, 0 otherwise.)

(6) If $X=(X_1,\ldots,X_N)\in\mathbb{R}^{K\times N}$ and $j\in\{1,\ldots,N\}$, we define

$$\frac{\partial E_{\mathrm{SCF}}(X)}{\partial X_j}=\begin{bmatrix}\dfrac{\partial E_{\mathrm{SCF}}(X)}{\partial X_{1j}}\\ \dfrac{\partial E_{\mathrm{SCF}}(X)}{\partial X_{2j}}\\ \vdots\\ \dfrac{\partial E_{\mathrm{SCF}}(X)}{\partial X_{Kj}}\end{bmatrix}.$$

APPENDIX B: GLOBAL CONVERGENCE RESULTS

In this appendix we give a rigorous proof that Algorithm 5.1 is globally convergent. We strongly rely on the theory developed in Ref. 22. As in Ref. 22, the optimization problem to which the trust-region algorithm applies will be quite general. We will define Algorithms B.1 and B.2. Algorithm B.1 is, essentially, the trust-region Algorithm 2.1 of Ref. 22 with slight differences that favor its application to our problem. Algorithm B.2 is a Levenberg-Marquardt[19] modification of Algorithm B.1. In the Levenberg-Marquardt regularization

approach the trust-region constraint is replaced by a penalty term added to the objective function of the subproblem. In this way, trust-region subproblems become easily solvable. Finally, we will see that Algorithm 5.1 is a particular case of Algorithm B.2.

1. General assumptions on the problem

Consider the problem

$$\text{Minimize } f(x) \text{ subject to } x \in \mathcal{R}, \tag{B1}$$

where $f:\mathbb{R}^n \to \mathbb{R}$. Assume that $\mathcal{R}$ is closed, f is differentiable and

$$\|\nabla f(y) - \nabla f(x)\|_2 \leqslant L\|y-x\|_2$$

for all x, y belonging to open and convex set that contains $\mathcal{R}$.

The feasible set $\mathcal{R}$ is defined by a finite set of smooth algebraic equations and inequations. We assume that all the points of $\mathcal{R}$ are *regular*, which means that the gradients of the active constraints are linearly independent at every feasible point. Under this condition (see Ref. 34, p. 314) every local minimizer of Eq. (B1) satisfies the Karush–Kuhn–Tucker (KKT) optimality conditions. Points in $\mathcal{R}$ that satisfy KKT are said to be *stationary*.

2. Trust-region algorithm

Let $\|\cdot\|_A$ denote an arbitrary norm on $\mathbb{R}^n$. Let $\alpha \in (0,1/2)$, $M>0$, B_k symmetric and

$$\|B_k\|_2 \leqslant M \quad \forall k \in \mathbb{N}.$$

For all $k \in \mathbb{N}$, let $\{\Delta_{k,\ell}\}_{\ell \in \mathbb{N}} \subset \{t \in \mathbb{R} | t>0\}$ be such that

$$\lim_{\ell \to \infty} \Delta_{k,\ell} = 0.$$

(Neither the matrices B_k nor the sequences of trust-region radius $\{\Delta_{k,\ell}\}_{\ell \in \mathbb{N}}$ need to be computed in advance, but only at the steps of the algorithm where they are used.)

The algorithm described below is, essentially, Algorithm 2.1 of Ref. 22 with a more liberal choice of the trust-region radius $\Delta_{k,\ell}$ and a stricter resolution of quadratic subproblems.

a. Algorithm B.1

Step 1. Choose $x^0 \in \mathcal{R}$ and set $k \leftarrow 0$.
Step 2. Set $\ell \leftarrow 0$.
Step 3. Compute a global solution $\overline{s_k}(\Delta_{k,\ell})$ of

$$\left.\begin{array}{lll} \text{Minimize} & \psi_k(s) \equiv & \frac{1}{2} s^T B_k s + g_k^T s \\ \text{subject to} & x^k + s \in \mathcal{R}, & \\ & \|s\|_A \leqslant \Delta_{k,\ell} & \end{array}\right\}, \tag{B2}$$

If $\psi_k[\overline{s_k}(\Delta_{k,\ell})] = 0$, terminate the execution of the algorithm.
Step 4. If

$$f[x^k + \overline{s_k}(\Delta_{k,\ell})] \leqslant f(x^k) + \alpha \psi_k[\overline{s_k}(\Delta_{k,\ell})], \tag{B3}$$

define

$$s_k = \overline{s_k}(\Delta_{k,\ell}), \Delta_k = \Delta_{k,\ell}, \text{acc}(k) = \ell,$$

compute $x^{k+1} \in \mathcal{R}$ such that

$$f(x^{k+1}) \leqslant f(x^k + s_k), \tag{B4}$$

set $k \leftarrow k+1$ and go to step 2.

If Eq. (B3) does not hold, set $\ell \leftarrow \ell + 1$ and go to step 3. □

Remarks. In Algorithm 2.1 of Ref. 22 the subproblems (B2) do not need to be solved accurately. Instead, each subproblem resolution is preceded by the minimization of a simple majorizing quadratic of the form $Q_k(s) = (1/2)M\|s\|_2^2 + g_k^T s$ and, after that, a trial point such that $\psi_k[\overline{s_k}(\Delta_{k,\ell})] \leqslant Q_k[s_k^Q(\Delta_{k,\ell})]$ is taken. Of course, if the trial increment is a global solution of Eq. (B2), the requirements of Algorithm 2.1 of Ref. 22 are also satisfied.

Observe that the condition (B4) has the same meaning and interpretation as the condition (18) in Algorithm 5.1.

The global convergence theory of a minimization algorithm usually involves two steps. First, one proves that the algorithm is *well defined*. This means that, unless the current point is stationary (generally, a solution) an iteration necessarily finishes in finite time obtaining a new iterate. The second step consists in showing that all the limit points of the sequence generated by the algorithm are stationary points and, of course, that such limit points exist. In this way, it can be guaranteed that stationary points are necessarily found up to any desired precision. Recall that, in our case, stationary points coincide with Fock fixed points.

Since Algorithm B.1 is based on Algorithm 2.1 of Ref. 22, the following results are true:

Theorem B.1. If Algorithm B.1 terminates at step 3, then x^k is stationary.
Proof. See Theorem 2.2 of Ref. 22. □

Theorem B.2. If x^k is not a stationary point of Eq. (B1), then Eq. (B3) holds for ℓ large enough, and, so, x^{k+1} is well defined.
Proof. See Theorem 2.3 of Ref. 22. □

For proving global convergence of Algorithm B.1 we need an additional assumption. Assumption A says that the sequences $\{\Delta_{k,\ell}\}_{\ell \in \mathbb{N}}$ should not converge to 0 too fast. As a consequence, a ''very small'' accepted trust-region radius is necessarily preceded by a small trust-region radius for which Eq. (B3) was not satisfied at the same iteration.

Assumption A. If $K \subset \mathbb{N}$ is an infinite sequence of indices such that

$$\lim_{k \in K} x^k = x_*.$$

and

$$\lim_{k \in K} \Delta_k = 0,$$

then, either x_* is a stationary point of Eq. (B1) or

$$\lim_{k \in K} \Delta_{k,\text{acc}(k)-1} = 0.$$

In Algorithm 2.1 of Ref. 22 Assumption A is guaranteed taking

$$\Delta_{k,0} \geqslant \Delta_{\min} > 0 \tag{B5}$$

and

$$\Delta_{k,\ell+1} \in [\underline{\tau}\Delta_{k,\ell}, \overline{\tau}\Delta_{k,\ell}] \quad \forall \ell \in \mathbb{N} \tag{B6}$$

for all $k \in \mathbb{N}$, where $\Delta_{\min} > 0$ and $0 < \underline{\tau} < \overline{\tau} < 1$.

Theorem B.3. Assume that Assumption A holds. Let $\{x^k\}$ *be a sequence generated by Algorithm B.1 and let* x_* *be an limit point. Then,* x_* *is stationary.*

Proof. All the arguments in the proof of Theorem 3.2 of Ref. 22 hold replacing the requirements (B5) and (B6) by Assumption A. □

3. Levenberg–Marquardt-like algorithm

The Levenberg–Marquardt (LM) or regularization approach is often used to enhance convergence properties of unconstrained (and some constrained) minimization algorithms based on sufficient decrease of the objective function. The connections of regularization approaches with trust-region ones are well known. See Ref. 19 and references therein. Briefly speaking, regularization parameters are the Lagrange multipliers of trust-region subproblems. In this section we define LM-like algorithms associated with the trust-region methods of Sec. B, we prove that they have similar global convergence properties and we introduce the LM version of the trust-region method.

Let $\alpha \in (0,1/2)$, $0 < \sigma_{\min} < \sigma_{\max} < \infty$, $1 < \tau_{\min} < \tau_{\max} < \infty$, and $A \in \mathbb{R}^{n \times n}$ be symmetric and positive definite. Define

$$\mathcal{B} = \{B \in \mathbb{R}^{n \times n} | B = B^T, \|B\|_2 \leqslant M\}.$$

a. Algorithm B.2

Step 1. Choose $x^0 \in \mathcal{R}$ and set $k \leftarrow 0$.

Step 2. Choose $B_k \in \mathcal{B}$, $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$. Set $\ell \leftarrow 0$, $t_{k,0} = 0$.

Step 3. Define, for all $s \in \mathbb{R}^n$,

$$Q_{k,\ell}(s) = (g^k)^T s + \tfrac{1}{2} s^T (B_k + t_{k,\ell} \sigma_k A) s. \tag{B7}$$

Step 4. Compute $\hat{s}(t_{k,\ell})$, a global solution of

$$\text{Minimize } Q_{k,\ell}(s) \text{ subject to } x^k + s \in \mathcal{R}. \tag{B8}$$

If $Q_{k,\ell}[\hat{s}(t_{k,\ell})] = 0$ terminate the execution of the algorithm.

Step 5.

If

$$f[x^k + \hat{s}(t_{k,\ell})] \leqslant f(x^k) + \alpha Q_{k,0}[\hat{s}(t_{k,\ell})], \tag{B9}$$

set $\text{acc}(k) = \ell$, $t_k = t_{k,\ell}$, compute $x^{k+1} \in \mathcal{R}$ such that

$$f(x^{k+1}) \leqslant f[x^k + \hat{s}(t_{k,\ell})], \tag{B10}$$

set $k \leftarrow k+1$ and go to step 2.

If Eq. (B9) does not hold, then, if $\ell = 0$ take $t_{k,\ell+1} > 0$. If $\ell > 0$, take $t_{k,\ell+1} \in [\tau_{\min} t_{k,\ell}, \tau_{\max} t_{k,\ell}]$. Set $\ell \leftarrow \ell + 1$ and go to step 3. □

From now on, we define

$$\|z\|_A = \sqrt{z^T A z} \quad \forall z \in \mathbb{R}^n.$$

The relation between the LM-like iteration defined by Algorithm B.2 and a trust-region iteration is given by the following proposition.

Proposition B.1. Assume that $\hat{s}(t_{k,\ell})$ *is a solution of Eq. (B8) and* s_{trust} *is a solution of*

$$\left.\begin{array}{ll} \text{Minimize} & Q_{k,0}(s) \\ \text{subject to} & x^k + s \in \mathcal{R}, \\ & \|s\|_A \leqslant \|\hat{s}(t_{k,\ell})\|_A \end{array}\right\}. \tag{B11}$$

Then, s_{trust} *is a global solution of Eq. (B8) and* $\hat{s}(t_{k,\ell})$ *is a global solution of Eq. (B11).*

Proof. For $\ell = 0$ the proof is trivial. Suppose that $\ell > 0$. Since s_{trust} is a minimizer of Eq. (B11) and $\hat{s}(t_{k,\ell})$ is a feasible point of Eq. (B11), we have

$$Q_{k,0}(s_{\text{trust}}) \leqslant Q_{k,0}[\hat{s}(t_{k,\ell})]. \tag{B12}$$

But, since $\|s_{\text{trust}}\|_A \leqslant \|\hat{s}(t_{k,\ell})\|_A$,

$$\frac{t_{k,\ell}}{2} s_{\text{trust}}^T \sigma_k A s_{\text{trust}} \leqslant \frac{t_{k,\ell}}{2} \hat{s}(t_{k,\ell})^T \sigma_k A \hat{s}(t_{k,\ell}). \tag{B13}$$

Adding Eqs. (B12) and (B13), we get

$$\begin{aligned} Q_{k,\ell}(s_{\text{trust}}) &= Q_{k,0}(s_{\text{trust}}) + \frac{t_{k,\ell}}{2} s_{\text{trust}}^T \sigma_k A s_{\text{trust}} \\ &\leqslant Q_{k,0}[\hat{s}(t_{k,\ell})] + \frac{t_{k,\ell}}{2} \hat{s}(t_{k,\ell})^T \sigma_k A \hat{s}(t_{k,\ell}) \\ &= Q_{k,\ell}[\hat{s}(t_{k,\ell})]. \end{aligned}$$

So, s_{trust} is a global solution of Eq. (B8). For the second part of the thesis, note that, if $\hat{s}(t_{k,\ell})$ is not a global solution of Eq. (B11) we have

$$Q_{k,0}(s_{\text{trust}}) < Q_{k,0}[\hat{s}(t_{k,\ell})]. \tag{B14}$$

So, adding Eqs. (B13) and (B14),

$$Q_{k,\ell}(s_{\text{trust}}) < Q_{k,\ell}[\hat{s}(t_{k,\ell})].$$

That is, $\hat{s}(t_{k,\ell})$ would not be a global solution of Eq. (B8). This completes the proof. □

By Proposition B.1, defining

$$\Delta_{k,\ell} = \|\hat{s}(t_{k,\ell})\|_A \tag{B15}$$

and

$$\psi_k(s) = Q_{k,0}(s) \quad \forall s \in \mathbb{R}^n,$$

Algorithm B.2 has exactly the same form as Algorithm B.1. For proving that it has the same global convergence properties it remains to prove that $\Delta_{k,\ell}$ defined by Eq. (B15) is such that, for fixed k, $\Delta_{k,\ell}$ tends to ∞ if ℓ tends to infinity and that Assumption A holds. This is done in the following two lemmas.

Lemma B.1. Assume that, at some iteration k of Algorithm B.2, ℓ *tends to infinity and* $\Delta_{k,\ell}$ *is defined by* Eq. (B15). *Then,*

$$\lim_{\ell \to \infty} \Delta_{k,\ell} = 0.$$

Proof. Since $\tau_{\min} > 1$, the fact that ℓ tends to infinity implies that $t_{k,\ell}$ tends to infinity too.

Since $Q_{k,\ell}(0) = 0$ and $\hat{s}(t_{k,\ell})$ is a global minimizer of $Q_{k,\ell}(s)$ we have that

$$Q_{k,\ell}[\hat{s}(t_{k,\ell})] \leqslant 0 \quad \forall \ell.$$

So,

$$(g^k)^T\hat{s}(t_{k,\ell})+\tfrac{1}{2}\hat{s}(t_{k,\ell})^T(B_k+t_{k,\ell}\sigma_k A)\hat{s}(t_{k,\ell})\leqslant 0.$$

Therefore,

$$\frac{t_{k,\ell}\sigma_k}{2}\hat{s}(t_{k,\ell})^T A\hat{s}(t_{k,\ell})$$

$$\leqslant -(g^k)^T\hat{s}(t_{k,\ell})-\frac{1}{2}\hat{s}(t_{k,\ell})^T B_k\hat{s}(t_{k,\ell})$$

$$\leqslant \|g(x^k)\|_2\|\hat{s}(t_{k,\ell})\|_2+\frac{M}{2}\|\hat{s}(t_{k,\ell})\|_2^2.$$

Since $\mathcal{R}$ is bounded, the right-hand side of this inequality is bounded independently of ℓ. But, since $t_{k,\ell}\to\infty$ and $\sigma_{\min}\leqslant\sigma_k\leqslant\sigma_{\max}$, we have that

$$\lim_{\ell\to\infty}\hat{s}(t_{k,\ell})^T A\hat{s}(t_{k,\ell})=0.$$

So, $\lim_{\ell\to\infty}\Delta_{k,\ell}=0$ as we wanted to prove. □

Lemma B.2. Assume that $\{x^k\}$ *is an infinite sequence generated by Algorithm B.2 and* $\Delta_{k,\ell}$ *is defined by Eq.* (B15). *Then, Assumption A holds*

Proof. Let K_1 be an infinite subset of $\mathbb{N}$ such that

$$\lim_{k\in K_1}x^k=x_*$$

and

$$\lim_{k\in K_1}\Delta_k=\lim_{k\in K_1}\Delta_{k,\mathrm{acc}(k)}=0.$$

We consider two possibilities:

(1) There exists an infinite subset $K_2\subset K_1$ such that $t_k\equiv t_{k,\mathrm{acc}(k)}$ is bounded.

(2) $\lim_{k\in K_1}t_k=\infty$.

Assume first that $\{t_k\}_{k\in K_1}$ is bounded. Then, there exists K_3, an infinite subsequence of K_2, such that

$$\lim_{k\in K_3}B_k+\sigma_k t_k A=B+\sigma tA=\overline{B}.$$

Let $s\in\mathbb{R}^n$ be such that $x_*+s\in\mathcal{R}$. Then, for all $k\in K_3$ we have that

$$(g^k)^T\hat{s}(t_k)+\tfrac{1}{2}\hat{s}(t_k)^T(B_k+t_k\sigma_k A)\hat{s}(t_k)$$
$$\leqslant(g^k)^T(x_*+s-x^k)$$
$$+\tfrac{1}{2}(x_*+s-x^k)^T(B_k+t_k\sigma_k A)(x_*+s-x^k).$$

So, taking limits for $k\in K_3$ and using that

$$\lim_{k\in K_3}\|\hat{s}(t_k)\|_A=\lim_{k\in K_3}\Delta_k=0,$$

we obtain

$$g(x_*)^T s+\tfrac{1}{2}s^T\overline{B}s\geqslant 0$$

for all $s\in\mathbb{R}^n$ such that $x_*+s\in\mathcal{R}$. Therefore, $0\in\mathbb{R}^n$ is a minimizer of $g(x_*)^T s+\frac{1}{2}s^T\overline{B}s$ subject to $x_*+s\in\mathcal{R}$. Since x_* is regular, the KKT conditions for this problem hold and, since these KKT conditions are the same as the KKT conditions of Eq. (B1), x_* is stationary.

Now, assume that $\lim_{k\in K_1}t_k=\infty$. Since $t_k\leqslant\tau_{max}t_{k,\mathrm{acc}(k)-1}$ we have that

$$\lim_{k\in K_1}t_{k,\mathrm{acc}(k)-1}=\infty.$$

Since $Q_{k,t_{k,\mathrm{acc}(k)-1}}(0)=0$ and $\hat{s}[t_{k,\mathrm{acc}(k)-1}]$ is a global minimizer of $Q_{k,t_{k,\mathrm{acc}(k)-1}}(s)$ we have

$$Q_{k,\mathrm{acc}(k)-1}[\hat{s}(t_{k,\mathrm{acc}(k)-1})]\leqslant 0\quad\forall k\in K_1.$$

So

$$(g^k)^T\hat{s}(t_{k,\mathrm{acc}(k)-1})+\tfrac{1}{2}\hat{s}(t_{k,\mathrm{acc}(k)-1})^T$$
$$\times(B_k+t_{k,\mathrm{acc}(k)-1}\sigma_k A)\hat{s}(t_{k,\mathrm{acc}(k)-1})\leqslant 0\quad\forall k\in K_1.$$

Therefore, for all $k\in K_1$,

$$\frac{t_{k,\mathrm{acc}(k)-1}\sigma_k}{2}\hat{s}(t_{k,\mathrm{acc}(k)-1})^T A\hat{s}(t_{k,\mathrm{acc}(k)-1})$$

$$\leqslant -(g^k)^T\hat{s}(t_{k,\mathrm{acc}(k)-1})-\tfrac{1}{2}\hat{s}(t_{k,\mathrm{acc}(k)-1})^T B_k\hat{s}(t_{k,\mathrm{acc}(k)-1})$$

$$\leqslant\|g(x^k)\|_2\|\hat{s}(t_{k,\mathrm{acc}(k)-1})\|_2+\frac{M}{2}\|\hat{s}(t_{k,\mathrm{acc}(k)-1})\|_2^2.$$

Since $\mathcal{R}$ is bounded, the right-hand side of this inequality is bounded too. But, since $t_{k,\mathrm{acc}(k)-1}\to\infty$ and $\sigma_{\min}\leqslant\sigma_k\leqslant\sigma_{\max}$, we have that

$$\lim_{k\in K_1}\hat{s}(t_{k,\mathrm{acc}(k)-1})^T A\hat{s}(t_{k,\mathrm{acc}(k)-1})=0.$$

So, $\lim_{k\in K_1}\Delta_{k,\mathrm{acc}(k)-1}=0$ as we wanted to prove. □

We proved that Algorithm B.2 is a particular case of Algorithm B.1 and that Assumption A is satisfied. Therefore, by Theorem B.2, the following global convergence theorem also holds.

b. Theorem B.3

(1) *If Algorithm B.2 terminates at step 4, then* x^k *is a stationary point of Eq. (B1).*

(2) *If* x^k *is not a stationary point of Eq. (B1), then Eq. (B9) holds for* ℓ *large enough, and, so,* x^{k+1} *is well defined.*

(3) *Let* $\{x^k\}$ *be a sequence generated by Algorithm B.2. Then,* $\{x^k\}$ *admits at least one limit point and every limit point is stationary.*

So far, we defined a globally convergent method for solving nonlinear programming problems [Eq. (B1)] such that all the iterates are feasible points ($x^k\in\mathcal{R}$) and $f(x^{k+1})<f(x^k)$ for all k. Algorithm B.2 tends to be more easily implementable than Algorithm 2.1 of Ref. 22 because in the latter the feasible set of the subproblems is the intersection of $\mathcal{R}$ with a trust-region ball whereas in Algorithm B.2 the feasible region of the subproblems is $\mathcal{R}$. However, in many general nonlinear programming problems, even subproblem (B8) can be very difficult (perhaps, as difficult as the original problem). In our case, with the appropriate definition of B_k, subproblems (B8) are easy and, so, the Levenberg–Marquardt (LM)-like algorithm becomes attractive.

Algorithm V.1 shares the same theoretical properties of Algorithm B.2, as stated in the following theorem.

c. Theorem B.4

(1) *If Algorithm 5.1 terminates at step 4, then* x^k *is a stationary point of Eq. (1).*

(2) *If* x^k *is not a stationary point of Eq. (1) and* $type(k)=2$, *then Eq. (17) holds for t large enough, and, so,* x^{k+1} *is well defined.*

(3) *Let* $\{x^k\}$ *be a sequence generated by Algorithm 5.1. Then,* $\{x^k\}$ *admits at least one limit point and every limit point is stationary.*

Proof. Algorithm 5.1 is a particular case of Algorithm B.2. Then, the thesis follows from Theorem B.3. □

APPENDIX C: RESOLUTION OF THE EASY SUBPROBLEMS

In this appendix we explain why the subproblems described in Sec. IV are computationally simple. We analyze the solution of Eq. (16) with $type(k)=2$. Let $\rho=1+t$. Then, the "easy" subproblem (16) is equivalent to

$$\text{Minimize}\ \frac{2}{\sigma_k\rho}(g^k)^T(x-x^k)+(x-x^k)^TA(x-x^k)$$

$$\text{subject to}\ \ x\in\mathcal{R}.$$

We perform the following change of variables in $\mathbb{R}^n$:

$$y=A^{1/2}x.$$

Consequently,

$$x=A^{-1/2}y,\ \ y_k=A^{1/2}x^k,\ \ x^k=A^{-1/2}y_k.$$

Moreover, writing

$$Y_i=S^{1/2}X_i\quad \forall i=1,\ldots,N,$$

we also have

$$Y_i^k=S^{1/2}X_i^k\quad \forall i=1,\ldots,N.$$

Let us write

$$y=\begin{pmatrix} Y_1\\ \cdot\\ \cdot\\ \cdot\\ Y_N\end{pmatrix}\in\mathbb{R}^{KN},\ \ y_k=\begin{pmatrix} Y_1^k\\ \cdot\\ \cdot\\ \cdot\\ Y_N^k\end{pmatrix}\in\mathbb{R}^{KN},$$

$$Y=(Y_1,\ldots,Y_N)\in\mathbb{R}^{K\times N},\ \ Y_k=(Y_1^k,\ldots,Y_N^k)\in\mathbb{R}^{K\times N}.$$

So, the easy subproblem becomes

$$\text{Minimize}\ \frac{2}{\sigma_k\rho}(g^k)^TA^{-1/2}(y-y_k)+\|y-y_k\|_2^2$$

$$\text{subject to}\ \ Y^TY=I_N.$$

Calling

$$\overline{g_k}=\frac{1}{\sigma_k\rho}A^{-1/2}g^k,$$

the easy subproblem is equivalent to

$$\text{Minimize}\ \ 2\overline{g_k}^T(y-y_k)+\|y-y_k\|_2^2\ \ \text{subject to}\ \ Y^TY=I_N.$$

This is equivalent to

$$\text{Minimize}\ \ \|y-(y_k-\overline{g_k})\|_2^2\ \ \text{subject to}\ \ Y^TY=I_N.$$

Let us write

$$z_k=y_k-\overline{g_k}=\begin{pmatrix} Z_1^k\\ \cdot\\ \cdot\\ \cdot\\ Z_N^k\end{pmatrix}\in\mathbb{R}^{KN}$$

and

$$\overline{Z}=(Z_1^k,\ldots,Z_N^k)\in\mathbb{R}^{K\times N}.$$

Then the easy subproblem is

$$\text{Minimize}\ \ \|Y-\overline{Z}\|_F^2\ \ \text{subject to}\ \ Y^TY=I_N,\tag{B16}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Assume that

$$\overline{Z}=U\Sigma V^T$$

is the SVD decomposition of $\overline{Z}$. Therefore, $U\in\mathbb{R}^{K\times K}$ and $V\in\mathbb{R}^{N\times N}$ are unitary and $\Sigma\in\mathbb{R}^{K\times N}$ is diagonal. Since $\|Q_1A\|_F=\|AQ_2\|_F=\|A\|_F$, whenever Q_1 and Q_2 are unitary, the easy problem is equivalent to

$$\text{Minimize}\ \ \|U^TYV-\Sigma\|_F^2\ \ \text{subject to}\ \ Y^TY=I_N.$$

Write $W=U^TYV$. The statements $Y^TY=I_N$ and $W^TW=I_N$ are clearly equivalent, therefore the solution of the problem above is $Y=UWV^T$, where W solves

$$\text{Minimize}\ \ \|W-\Sigma\|_F^2\ \ \text{subject to}\ \ W^TW=I_N.$$

A solution of this problem is the diagonal matrix $W\in\mathbb{R}^{K\times N}$ that has 1's on its diagonal. We will call $I_{K\times N}$ this matrix for now on. So, the solution Y of Eq. (B16) is

$$Y=UI_{K\times N}V^T.$$

Therefore, writing $U=(U_1,\ldots,U_K)$, $V=(V_1,\ldots,V_N)$ we have

$$Y=U_1V_1^T+\cdots+U_NV_N^T.$$

Finally, the solution of the easy subproblem is

$$X=S^{-1/2}Y.$$

[1] A. Szabo and S. N. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Eletronic Structure Theory* (Dover, New York, 1989).
[2] W. Kohn, A. D. Becke, and R. G. Parr, J. Phys. Chem. **100**, 12974 (1996).
[3] P. Pulay, J. Comput. Chem. **3**, 556 (1982).
[4] A. D. Daniels and G. E. Scuseria, Phys. Chem. Chem. Phys. **2**, 2173 (2000).
[5] M. W. Schmidt, K. K. Baldridge, J. A. Boatz *et al.*, J. Comput. Chem. **14**, 1347 (1993).
[6] M. J. Frisch, G. W. Trucks, H. B. Schlegel *et al.*, GAUSSIAN 03, Gaussian, Inc., Pittsburgh, PA, 2003.
[7] P. Pulay, Chem. Phys. Lett. **180**, 461 (1991).
[8] R. Seeger and J. Pople, J. Chem. Phys. **65**, 265 (1976).
[9] G. Vacek, J. K. Perry, and J.-M. Langlois, Chem. Phys. Lett. **310**, 189 (1999).
[10] A. D. Rabuck and G. E. Scuseria, J. Chem. Phys. **110**, 695 (1999).
[11] R. Fournier, J. Andzelm, A. Goursot, N. Russo, and D. R. Salahub, J. Chem. Phys. **93**, 2919 (1990).
[12] V. R. Saunders and I. H. Hillier, Int. J. Quantum Chem. **7**, 699 (1973).
[13] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (Wiley, New York, 2000).

[14] G. B. Bacskay, Chem. Phys. **61**, 385 (1981).
[15] E. Cancès and C. Le Bris, Int. J. Quantum Chem. **79**, 82 (2000).
[16] E. Cancès and C. Le Bris, Math. Modell. Numer. Anal. **34**, 749 (2000).
[17] K. N. Kudin, G. E. Scuseria, and E. Cancès, J. Chem. Phys. **116**, 8255 (2002).
[18] L. Thogersen, J. Olsen, D. Yeager, P. Jorgensen, P. Salek, and T. Helgaker, J. Chem. Phys. **121**, 16 (2004).
[19] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-region Methods*, MPS-SIAM Series on Optimization Vol. 1 (SIAM, Philadelphia, 2000).
[20] M. J. D. Powell, in *Nonlinear Programming*, edited by J. B. Rosen, O. L. Mangasarian, and K. Ritter (Academic, London, 1970).
[21] D. C. Sorensen, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **19**, 409 (1982).
[22] J. M. Martínez and S. A. Santos, Math. Program. **68**, 267 (1995).
[23] J. M. Martínez and S. A. Santos, RAIRO-Oper. Res. **31**, 269 (1997).
[24] J. Barzilai and J. M. Borwein, IMA J. Numer. Anal. **8**, 141 (1988).
[25] M. Raydan, IMA J. Numer. Anal. **13**, 321 (1993).
[26] M. Raydan, SIAM J. Optim. **7**, 26 (1997).
[27] F. Luengo, M. Raydan, W. Glunt, and T. L. Hayden, Numer. Algorithms **30**, 241 (2002).
[28] E. G. Birgin, J. M. Martínez, and M. Raydan, SIAM J. Optim. **10**, 1196 (2000).
[29] E. G. Birgin, J. M. Martínez, and M. Raydan, IMA J. Numer. Anal. **23**, 539 (2003).
[30] R. Fletcher, On the Barzilai-Borwein method, Dundee, Scotland, 2001 (http://www.maths.dundee.ac.uk/~ftp/na-reports/NA207_RF.ps.Z).
[31] A. H. Sameh and J. A. Wisniewski, IMA J. Numer. Anal. **19**, 1243 (1982).
[32] G. H. Golub and Ch. F. Van Loan, *Matrix Computations* (The Johns Hopkins University Press, Baltimore, 1996).
[33] A. Edelman, T. A. Arias, and S. T. Smith, SIAM J. Matrix Anal. Appl. **20**, 303 (1998).
[34] D. Luenberger, *Linear and Nonlinear Programming* (Addison-Wesley, Massachussets, 1984).
[35] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).