Comput Optim Appl (2011) 50:555–590
DOI 10.1007/s10589-010-9318-6

Inexact restoration method for minimization problems arising in electronic structure calculations

Juliano B. Francisco · J.M. Martínez · Leandro Martínez · Feodor Pisnitchenko

Received: 20 July 2009 / Published online: 28 January 2010

Abstract An inexact restoration (IR) approach is presented to solve a matricial optimization problem arising in electronic structure calculations. The solution of the problem is the closed-shell density matrix and the constraints are represented by a Grassmann manifold. One of the mathematical and computational challenges in this area is to develop methods for solving the problem not using eigenvalue calculations and having the possibility of preserving sparsity of iterates and gradients. The inexact restoration approach enjoys local quadratic convergence and global convergence to stationary points and does not use spectral matrix decompositions, so that, in principle, large-scale implementations may preserve sparsity. Numerical experiments show that IR algorithms are competitive with current algorithms for solving closed-shell Hartree-Fock equations and similar mathematical problems, thus being a promising alternative for problems where eigenvalue calculations are a limiting factor.

This work was supported by PRONEX-Optimization (PRONEX-CNPq/FAPERJ E-26/171.510/2006-APQ1), FAPESP (Grants 2006/53768-0 and 2005/57684-2) and CNPq.

J.B. Francisco
Department of Mathematics, Federal University of Santa Catarina, Santa Catarina, Brazil
e-mail: juliano@mtm.ufsc.br

J.M. Martínez (✉)
Department of Applied Mathematics, University of Campinas, Campinas, Brazil
e-mail: martinez@ime.unicamp.br

L. Martínez
Institute of Chemistry, University of Campinas, Campinas, Brazil
e-mail: leandromartinez98@gmail.com

F. Pisnitchenko
Department of Applied Mathematics, University of Campinas, Campinas, Brazil
e-mail: feodor@ime.unicamp.br

Keywords Constrained optimization · Inexact restoration · Hartree-Fock · Self-consistent field · Quadratic convergence · Numerical experiments

1 Introduction

Assume that the 3D coordinates of the nuclei of atoms in an atomic arrangement are known. An electronic structure calculation consists of finding the wave functions from which the spatial electronic distribution of the system can be derived [1–4]. These wave functions are the solutions of the time-independent Schrödinger equation [4].

The practical solution of the Schrödinger equation from scratch is not possible, except in very simple situations. The solutions are functions of $\mathbb{R}^{3N} \to \mathbb{R}$ which must satisfy the Pauli exclusion principle and be anti-symmetric relative to interchange of electrons, which introduce additional complications [4]. Therefore, simplifications are made leading to more tractable mathematical problems.

The best-known approach consists of approximating the solution by a determinant composed by a combination of N functions, known as the Slater-determinant. In the most common case, in which the number of electrons of the system is even and every electron is paired, one uses one function for each pair of electrons to compose the Slater-determinant. One refer to systems satisfying these properties as "Restricted Closed-Shell" systems, and only systems of this type are of concern here. The approximation of the solution by the Slater-determinant allows for a significant simplification of the Schrödinger equation, which results in a "one-electron" eigenvalue equation, known as the Hartree-Fock equation [4]. The solutions of this new one-electron eigenvalue problem are used to reconstitute the Slater-determinant and, therefore, the electronic density of the system.

Writing each of the N functions composing the Slater-determinant as linear combinations of a given basis with K elements (here and in the sequel N denotes the number of electron pairs), the unknowns of the problem become the coefficients of each function with respect to the chosen basis, giving rise to a nonlinear eigenvalue problem known as the Hartree-Fock-Roothaan problem. The discretization technique uses plane wave basis or localized basis functions, with compact support [5] 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:

$$\text{Minimize } E(Z) \tag{1}$$

subject to

$$Z = Z^T, \qquad Z\mathcal{M}Z = Z, \qquad \text{Trace}\,(Z\mathcal{M}) = N \tag{2}$$

where $\mathcal{M}$ is a symmetric positive definite overlapping matrix that depends on the basis and $Z = CC^T$ is called the density matrix.

In the Restricted Closed-Shell Hartree-Fock-Roothaan problem, the form of $E(Z)$ in (1) is:

$$E_{SCF}(Z) = \text{Trace}\big[2HZ + G(Z)Z\big],$$

where Z is the one-electron density matrix in the atomic-orbital (AO) basis, H is the one-electron Hamiltonian matrix, $G(Z)$ is given by

$$G_{ij}(Z) = \sum_{k=1}^{K}\sum_{\ell=1}^{K}(2g_{ijk\ell} - g_{i\ell kj})Z_{\ell k},$$

$g_{ijk\ell}$ 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. Generally, $2N \leq K \leq 10N$. 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 matrix $F(Z)$ given by $F(Z) = H + G(Z)$ is known as Fock matrix and we have:

$$\nabla E_{SCF}(Z) = 2F(Z).$$

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

Defining $X = \mathcal{M}^{1/2}Z\mathcal{M}^{1/2}$, and $f(X) = E(\mathcal{M}^{-1/2}X\mathcal{M}^{-1/2})$ problem (1)–(2) becomes:

$$\text{Minimize } f(X) \tag{3}$$

subject to

$$X = X^T, \qquad XX = X, \qquad \text{Trace } (X) = N. \tag{4}$$

It is easy to see that:

- The feasible points (matrices) of (3)–(4) are orthogonal projection $K \times K$ matrices on subspaces of dimension N.
- Every feasible X may be written $X = CC^T$, where C has K rows and N orthonormal columns which form a basis of $\mathcal{R}(X)$, the column-space of X.
- Every feasible X satisfies $\|X\|_F^2 = N$.

We may always consider that $\nabla f(X)$ is a symmetric $K \times K$ matrix. In fact, for all X in the feasible set, we have that $f(X) = f((X + X^T)/2)$. Therefore, if $\nabla f(X)$ is not symmetric, we may replace $f(X)$ by $f((X + X^T)/2)$ without changing the optimization problem.

A fixed point iteration is frequently used to solve (3)–(4). The iteration function Ψ is defined as follows: Given a symmetric $X \in \mathbb{R}^{K\times K}$, one defines

$$\Psi(X) = CC^T, \tag{5}$$

where the columns of $C \in \mathbb{R}^{K\times N}$ are orthonormal eigenvectors corresponding to the N smallest eigenvalues of $\nabla f(X)$. Note that, strictly speaking, Ψ is a multivalued function due to possible coincidence between the N-th and the $(N + 1)$-th smallest eigenvalues.

It can be proved that, for all feasible X_k, $\Psi(X_k)$ is a solution of:

$$\text{Minimize } \langle \nabla f(X_k), X - X_k\rangle \quad \text{subject to} \quad (4). \tag{6}$$

(In (6) $\langle A, B\rangle$ denotes the standard scalar product $\langle A, B\rangle = \text{Trace}(A^T B)$.) The classical fixed-point SCF iteration for solving (3)–(4) is given by $X_{k+1} = \Psi(X_k)$. If X_k is a solution of (6), the point X_k is said to be "Aufbau". Since the objective function of problem (6) has the same gradient as the one of problem (3) and the constraints of both problems are the same, it turns out that any Aufbau point is a first-order stationary point of (3)–(4).

However, global minimizers of (3)–(4) are not necessarily Aufbau, as the following counter-example shows:

Let $K = 2$, $N = 1$. Define

$$f(X) = 2(x_{11} - 1/2)^2 + [(x_{12} + x_{21})/2]^2.$$

A global minimizer of this problem is:

$$X_* = \begin{pmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{pmatrix}.$$

Now:

$$\nabla f(X_*) = \begin{pmatrix} 0 & -1/2 \\ -1/2 & 0 \end{pmatrix}.$$

The eigenvalues of $\nabla f(X_*)$ are $\lambda_{min} = -1/2$ and $\lambda_{max} = 1/2$, corresponding to the eigenvectors $v_{min} = (1/\sqrt{2}, 1/\sqrt{2})^T$ and $v_{max} = (1/\sqrt{2}, -1/\sqrt{2})^T$ respectively.

However,

$$v_{min} v_{min}^T = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \neq X_*.$$

Therefore, X_* is not Aufbau. (In fact, $X_* = v_{max} v_{max}^T$, where v_{max} is the eigenvector corresponding to the biggest eigenvalue.)

The fact that all the global minimizers are not necessarily Aufbau is known in several problems of computational chemistry. In several cases, depending of the function f, it has been proved that global minimizers are Aufbau, but the problem seems to be open in other important cases. See, for example, [1], pp. 89–90.

The best-known algorithms for solving electronic structure problems are based on the fixed-point SCF iteration [2]. A popular variation consists of accelerating the sequence defined by $X_{k+1} = \Psi(X_k)$ by means of the DIIS (Direct Inversion Iterative Subspace) extrapolation procedure [6]. At each iteration, DIIS computes the point $\bar{X}_k$ in the affine subspace determined by $\{X_k, \ldots, X_{k-q}\}$ that minimizes an error in the least-squares sense and defines $X_{k+1} = \Psi(\bar{X}_k)$. Sequences generated by the SCF iteration or by its accelerated DIIS counterpart do not converge necessarily to minimizers of the problem. However, the SCF-DIIS algorithm is very successful in many problems and its employment is widely extended in practical calculations. The Optimal Damping Algorithm (ODA) [7] and the trust-region methods given in [8, 9] can be proved to be convergent, under different assumptions, to stationary points of (3)–(4). Other methods that incorporate trust-region concepts were given in [10, 11].

The methods mentioned above employ eigenvalue calculations for computing the SCF iteration or for solving the trust-region subproblems. When K, N are very large,

solving eigenvalue problems may be very expensive and, thus, efficient "eigen-free" methods are demanded. Moreover, possible sparse structure of the problem should be preserved.

In this paper we develop an Inexact Restoration algorithm [12–18] for solving (3)–(4) which, potentially, satisfies the requirements above.

The idea of the IR algorithm for solving (3)–(4) is the following. Given a feasible iterate Y_k we consider the tangent affine subspace to the constraints and we minimize (approximately) a Lagrangian approximation on that subspace. We will show that this can be done without matrix transformations using an adaptation of the Conjugate Gradient method. In this optimality phase we obtain a (not feasible) point X_{k+1}. A new feasible point Y_{k+1} is then obtained using a Newtonian iteration that guaranteedly converges to the closest feasible point to X_{k+1}. This is the basic "local" form of the method, which essentially corresponds to the IR description of Birgin and Martínez [13] that provides local quadratic convergence. Here we consider the modification introduced by Fischer and Friedlander [14] by means of which the method converges to KKT stationary points.

This paper is organized as follows:

- Section 2: We describe the IR method, in the versions [13] and [14].
- Section 3: We prove essential properties of the matricial optimization problem (3)–(4).
- Section 4: We describe the application of IR to (3)–(4).
- Section 5: We show how the method described in Sect. 4 may be implemented without eigenvalue calculations.
- Section 6: We show how to compute suitable approximations to the Lagrange multipliers at each iteration.
- Section 7: We describe a computer implementation of the method.
- Section 8: We show numerical experiments using a large set of mathematical test problems, with a comparison against classical methods.
- Section 9: We show numerical experiments with actual electronic structure calculations.
- Section 10: Conclusions and lines for future research.

Notation

- $\mathbb{N} = \{0, 1, 2, \ldots\}$.
- The symbol $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^n$. (Many times, it may be replaced by an arbitrary norm.)
- $\|A\|_F$ is the Frobenius norm of a matrix A.
- $P_D(x)$ is the Euclidean projection of x onto the convex set D. When we talk about projections we always mean Euclidean projections. In the case of matrices, this means projections with respect to the Frobenius norm.
- For all $X \in \mathbb{R}^{K\times K}$, $x = \text{vec}(X)$ denotes the vector of $\mathbb{R}^{K^2}$ that results from displaying the entries of X columnwise.
- If $h : \mathbb{R}^n \to \mathbb{R}^m$ we denote $\nabla h(x) = (\nabla h_1(x), \ldots, \nabla h_m(x)) \in \mathbb{R}^{n\times m}$.
- The column-space of a matrix X will be denoted $\mathcal{R}(X)$.
- $\text{Diag}(\sigma_1, \ldots, \sigma_n)$ will be the diagonal matrix with entries $\sigma_1, \ldots, \sigma_n$.
- If A and B are square matrices, we denote $\langle A, B\rangle = \text{Trace}\,(AB^T)$.

2 Inexact restoration method

In this section we describe the Inexact Restoration method (IR) that will be used in our main application. We will use the approach of Fischer and Friedlander [14], which consists of a simplified variation of the method introduced by Martínez [17], with improved global convergence properties.

Let Ω be a convex and closed polytope. Let $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$ be differentiable and assume that ∇f and ∇h are Lipschitz-continuous on Ω.

The Inexact Restoration method described in this section aims the solution of the nonlinear programming problem:

$$\text{Minimize } f(x) \quad \text{subject to} \quad h(x) = 0, x \in \Omega. \tag{7}$$

For all $x \in \Omega, \lambda \in \mathbb{R}^m$ we define the Lagrangian:

$$L(x, \lambda) = f(x) + h(x)^T \lambda. \tag{8}$$

We say that $(\bar{x}, \bar{\lambda}) \in \Omega \times \mathbb{R}^m$ is a *critical pair* of the optimization problem (7) if

$$h(\bar{x}) = 0 \tag{9}$$

and

$$P_\Omega(\bar{x} - \nabla L(\bar{x}, \bar{\lambda})) - \bar{x} = 0.$$

If $(\bar{x}, \bar{\lambda})$ is a critical pair, we say that $\bar{x}$ is a Karush-Kuhn-Tucker (KKT) point. Local minimizers of (7) that satisfy some "constraint qualification" necessarily satisfy the KKT conditions [19].

We define, for all $x \in \Omega, \theta \in [0, 1]$, the following merit function:

$$\Phi(x, \theta) = \theta f(x) + (1 - \theta)\|h(x)\|. \tag{10}$$

For all $y \in \mathbb{R}^n$ we define the *tangent set*:

$$T(y) = \{z \in \Omega | \nabla h(y)^T (z - y) = 0\}. \tag{11}$$

Algorithm 2.1 Inexact Restoration

The algorithmic parameters are $r \in [0, 1), \beta > 0, \gamma > 0, \tau > 0$. We assume that $r_k \in [0, r]$ for all $k \in \mathbb{N}$.

Step 0. Initialization

As initial approximation we choose, arbitrarily, $x^0 \in \Omega$. We initialize $\theta_{-1} \in (0, 1)$ and $k \leftarrow 0$.

Step 1. Restoration step

Compute $y^k \in \Omega$ such that:

$$\|h(y^k)\| \leq r_k \|h(x^k)\| \tag{12}$$

and

$$\|y^k - x^k\| \leq \beta \|h(x^k)\|. \tag{13}$$

Step 2. Penalty parameter
Compute θ_k as the first element θ of the sequence $\{\theta_{k-1}/2^j\}_{j\in\mathbb{N}}$ such that

$$\Phi(y^k, \theta) \leq \Phi(x^k, \theta) + \frac{1}{2}(\|h(y^k)\| - \|h(x^k)\|). \tag{14}$$

Step 3. Tangent descent direction
Compute $d^k \in \mathbb{R}^n$ such that $y^k + d^k \in \Omega$,

$$f(y^k + td^k) \leq f(y^k) - \gamma t \|d^k\|^2 \tag{15}$$

for all $t \in [0, \tau]$ and

$$\nabla h(y^k)^T d^k = 0. \tag{16}$$

Step 4. Acceptance of the step
Compute t_k as the first element t of the sequence $\{1, 1/2, 1/4, \ldots\}$ such that

$$\Phi(y^k + td^k, \theta_k) \leq \Phi(x^k, \theta_k) + \frac{1-r}{2}(\|h(y^k)\| - \|h(x^k)\|) \tag{17}$$

and

$$f(y^k + t_k d^k) \leq f(y^k) - \gamma t_k \|d^k\|^2.$$

Set $x^{k+1} = x^k + t_k d^k$, update $k \leftarrow k+1$ and go to Step 1.

The "Restoration Phase" of an Inexact Restoration method is the process that leads to find y^k fulfilling the requirements of Step 1. Step 3 represents the "Optimality Phase" of the IR iteration. The point $z^k \equiv y^k + d^k$ may be considered as an approximate minimizer on the tangent set of a suitable merit function associated to f, and d^k should be a direction along which f decreases. At Step 4 one accepts or rejects trial points, according to a merit function whose penalty parameter is computed at Step 2.

In [14] the following theorem has been proved.

Theorem 2.1 *Suppose that, for all $k \in \mathbb{N}$, Steps 1 and 3 of Algorithm 2.1 are well defined. Then*:

1. *For all $k \in \mathbb{N}$, the next iterate x^{k+1} is well defined.*
2. *There exists $k_0 \in \mathbb{N}$ such that $\theta_k = \theta_{k_0}$ for all $k \geq k_0$.*
3. $\lim_{k\to\infty} \|h(x^k)\| = \lim_{k\to\infty} \|h(y^k)\| = 0$.
4. $\lim_{k\to\infty} d^k = 0$.

Observe that, by (13) and the fact that $\|h(x^k)\| \to 0$, we have that $\|y^k - x^k\| \to 0$ and, so, the sequences $\{x^k\}$ and $\{y^k\}$ have the same limit points. Clearly, if x^* is a limit point, as $\|h(x^k)\| \to 0$, we have that $h(x^*) = 0$. Since Ω is closed this implies that x^* is feasible. The optimality of the limit points of sequences computed by Algorithm 2.1 is related to the fact that $d^k \to 0$. If the directions d^k are properly chosen, the fact that d^k tends to zero implies the Approximate Gradient Projection (AGP)

first-order necessary condition defined by Martínez and Svaiter [20]. If a constraint qualification holds at the limit point, the KKT conditions are also fulfilled.

Theorem 2.2 below is a direct consequence of Theorems 2.3, 2.4 and 2.5 of [13]. It says that, under certain conditions, the Inexact Restoration algorithm produces superlinearly or even quadratically convergent sequences.

Theorem 2.2 *Suppose, as in Theorem 2.1, that Steps 1 and 3 of Algorithm 2.1 are well defined for all $k \in \mathbb{N}$. Assume that there exist $c > 0$, $\eta \in [0, 1)$ such that for all $k \in \mathbb{N}$ there exist $\lambda^k \in \mathbb{R}^m$, $\eta_k \in [0, \eta]$ such that*

$$\|P_\Omega(y^k + d^k - \nabla L(y^k + d^k, \lambda^{k+1})) - (y^k + d^k)\| \leq \eta_k \|P_\Omega(y^k - \nabla L(y^k, \lambda^k)) - y^k\| \tag{18}$$

and

$$\|d^k\| + \|\lambda^{k+1} - \lambda^k\| \leq c\|P_\Omega(y^k - \nabla L(y^k, \lambda^k)) - y^k\|. \tag{19}$$

Assume, finally, that $(\bar{x}, \bar{\lambda})$ is a stationary pair and that $t_k = 1$ for k large enough.

Then, there exist $\delta, \varepsilon > 0$ such that:

- *If, for some $k_0 \in \mathbb{N}$, $\|x^{k_0} - \bar{x}\| \leq \varepsilon$ and $\|\lambda^{k_0} - \bar{\lambda}\| \leq \delta$, the sequence (x^k, λ^k) converges to some stationary pair.*
- *If r_k and η_k tend to zero, the convergence is R-superlinear.*
- *If $r = \eta = 0$ the convergence is R-quadratic.*

The assumptions of Theorem 2.2 deserve some comments. Assumption (18) says that $y^k + d^k$ is computed as an approximate minimizer of the Lagrangian $L(y^k + d, \lambda^k)$ on the tangent set. This is a reasonable requirement if one wants fast local convergence. Assumption (19) is a stability assumption that says that the solution of the optimality phase does not blow up. The more serious hypothesis of Theorem 2.2 is that, asymptotically, the condition (17) should be satisfied taking $t_k = 1$. We do not have a theoretical sufficient condition for this requirement, although we observed that it holds in many numerical tests.

3 Matricial optimization problem

Let $f : \mathbb{R}^{K\times K} \to \mathbb{R}$, $N \in \{1, \ldots, K\}$. Assume that f admits continuous first derivatives for all $X \in \mathbb{R}^{K\times K}$. The optimization problem addressed in this section is

$$\text{Minimize } f(X) \quad \text{subject to} \quad X \in \mathcal{G}, \tag{20}$$

where

$$\mathcal{G} = \{X \in \mathbb{R}^{K\times K} | X = X^T, X^2 = X, \text{ Trace } (X) = N\}. \tag{21}$$

We consider that (20) is a particular case of (7), with $n = K^2$ and the obvious identifications of the constraints defined by (21) with the constraints $h(x) = 0$. We note that the number of constraints of (20) is bigger than the number of variables of the

problem. Therefore, the number of gradients of constraints exceeds the dimension of the space. As a consequence, the gradients of constraints are linearly dependent and, so, the Linear Independence Constraint Qualification does not hold. Moreover, these constraints fail to satisfy even the Constant Rank Constraint Qualification (CRCQ) introduced in [21]. This condition says that, when some gradients of active constraints are linearly dependent at a feasible point, the same gradients must be linearly dependent in a neighborhood of that point. Taking $X_0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$, we observe that the gradient of the constraint $x_{11}x_{12} + x_{12}x_{22} - x_{12} = 0$ vanishes at X_0 but it is different from zero at $X_\varepsilon = \left(\begin{smallmatrix} 1 & 0 \\ 0 & \varepsilon \end{smallmatrix}\right)$ for all $\varepsilon > 0$. Therefore, CRCQ does not hold at X_0. Since (21) only involves equality constraints it turns out that the Constant Positive Linear Dependence constraint qualification [22, 23] is not fulfilled either. This state of facts makes it necessary to employ specific techniques to prove that local minimizers satisfy the KKT conditions.

Since the feasible set of (20) is the set of Euclidean projection matrices onto N-dimensional subspaces of $\mathbb{R}^K$, each $X \in \mathcal{G}$ may be written in the form

$$X = CC^T$$

where $C \in \mathbb{R}^{K \times N}$ has orthonormal columns that form a basis of the N-dimensional subspace $\mathcal{R}(X)$. Therefore, if $X \in \mathcal{G}$, it has N eigenvalues equal to 1 and $K - N$ eigenvalues equal to 0. The set of matrices with orthonormal columns is obviously compact, therefore, $\mathcal{G}$ is compact too. This implies that (20) admits a global minimizer. There exists a biunivoque correspondence between $\mathcal{G}$ and the Grassmann manifold formed by the subspaces of dimension N of $\mathbb{R}^K$. For each subspace $\mathcal{S}$ there exists one and only one orthogonal projection matrix X such that $\mathcal{R}(X) = \mathcal{S}$. Therefore, the feasible set of (20) may be identified with the Grassmann manifold defined by K and N.

Problem (20) has no inequality constraints. Therefore, for all $Y \in \mathbb{R}^{K \times K}$, the set $T(Y)$ defined by (11) is an affine subspace. The parallel subspace to $T(Y)$ will be called $\mathcal{S}(Y)$. Since there are no inequality constraints, the KKT conditions coincide with the classical Lagrange optimality conditions. These conditions are satisfied at a feasible point Y_* if $\nabla f(Y_*)$ is a linear combination of the gradients of the constraints evaluated at Y_*. In other words, the KKT conditions say that $\nabla f(Y_*)$ is orthogonal to $\mathcal{S}(Y_*)$.

When (20) comes from an electronic structure calculation the number of variables and constraints of (20) may be very large. In this case, N is of the order of the number of electrons of the system and the best known methods use variations of the SCF fixed point iteration, which involve computing N eigenvalues and eigenvectors of a symmetric $K \times K$ matrix [2, 4]. The challenge is to develop methods that scale linearly with respect to N, preserving possible sparsity of X and $\nabla f(X)$ and being efficient in the absence of gap between eigenvalues N and $N + 1$ of $\nabla f(X)$ [24–27].

Most algorithms for solving (20) are based on the SCF fixed point iteration. Given $X_k \in \mathcal{G}$, the point X_{k+1} is defined as a solution of

$$\text{Minimize Trace } [\nabla f(X_k)(X - X_k)] \quad \text{subject to} \quad X \in \mathcal{G}. \tag{22}$$

Therefore, in (22) one minimizes the linear approximation of $f(X)$ around X_k on the true feasible set. A solution of (22) is given by $X_{k+1} = CC^T$ where $C^T C =$

$I_{N\times N}$ and the columns of C form an orthonormal basis of the subspace associated to the N smallest eigenvalues of $\nabla f(X_k)$. Thus, in principle, solving (22) requires an eigenvalue calculation. Several efforts have been done in order to reduce the cost of the fixed point iteration.

Our proposal here is to solve (20) using Algorithm 2.1. The iterates will be denoted X_k (therefore, $x^k = \text{vec}(X_k)$). The intermediate points y^k will be defined by $y^k = \text{vec}(Y_k)$ and the Euclidean norm in $\mathbb{R}^n$ corresponds to the Frobenius norm of $\mathbb{R}^{K\times K}$.

The plausibility of using Algorithm 2.1 for solving (20) comes from a sequence of theoretical results given below.

The first result is a characterization of the sets $T(Y)$ and $\mathcal{S}(Y)$ for the case in which $Y \in \mathcal{G}$. We will prove that, if $\text{Trace}\,(Y) = N$, the linearization of the constraints $X^2 - X = 0$ automatically implies that the constraint $\text{Trace}\,(X) = N$ is preserved. Moreover, in spite of the number of variables and equations, the dimension of $\mathcal{S}(Y)$ is equal to $N(K - N)$ for all $Y \in \mathcal{G}$.

Lemma 3.1 *Assume that $Y \in \mathcal{G}$. Then,*

$$\mathcal{S}(Y) = \{E \in \mathbb{R}^{K\times K} | E = E^T \text{ and } YE + EY - E = 0\} \tag{23}$$

and

$$T(Y) = \{Z \in \mathbb{R}^{K\times K} | Z = Z^T \text{ and } Y(Z - Y) + (Z - Y)Y - (Z - Y) = 0\}. \tag{24}$$

Moreover, the dimension of $\mathcal{S}(Y)$ is $N(K - N)$.

Proof See Lemma 4.1 of [28]. □

The following lemma gives a closed formula for the projection of an arbitrary symmetric matrix on the linearization of the constraints. By this formula, we see that when A and Y have an adequate sparsity pattern, the projected matrix on $\mathcal{S}(Y)$ tends to be sparse too.

Lemma 3.2 *Assume that $Y \in \mathcal{G}$. Let A be a symmetric $K \times K$ matrix. Then, the Euclidean (Frobenius) projection of A onto $\mathcal{S}(Y)$ is given by*:

$$P_{\mathcal{S}(Y)}(A) = YA + AY - 2YAY. \tag{25}$$

Consequently, the projection of a symmetric matrix $B \in \mathbb{R}^{K\times K}$ onto $T(Y)$ is given by:

$$P_{T(Y)}(B) = Y + Y(B - Y) + (B - Y)Y - 2Y(B - Y)Y. \tag{26}$$

Proof By Lemma 3.1, the projection $P_{\mathcal{S}(Y)}(A)$ is the solution of the following optimization problem:

$$\text{Minimize } \|A - E\|_F^2 \quad \text{subject to} \quad YE + EY - E = 0,\ E = E^T. \tag{27}$$

Writing, as in Lemma 3.1, $Y = QDQ^T$ ($QQ^T = I$, D diagonal) and $E' = Q^T E Q$, using elementary properties of the Frobenius norm and a simple manipulation of the constraints, we see that (27) is equivalent to

$$\text{Minimize } \|G - E'\|_F^2 \quad \text{subject to} \quad DE' + E'D - E' = 0,\ E' = (E')^T, \tag{28}$$

with $G = (g_{ij}) = Q^T A Q$. We assume again, without loss of generality, that D is the diagonal matrix with $d_{ii} = 1$ if $i \leq N$ and $d_{ii} = 0$ if $i > N$. As in Lemma 3.1, it turns out that the feasible matrices of (28) have the form:

$$E' = \begin{pmatrix} 0 & R^T \\ R & 0 \end{pmatrix}, \tag{29}$$

where $R \in \mathbb{R}^{(K-N)\times N}$. Therefore, the solution of (28) is $E' = (e'_{ij})$, where

$$e'_{ij} = 0 \quad \text{if } (i \leq N, j \leq N) \text{ or } (i > Nj > N)$$

and

$$e'_{ij} = (g_{ij})$$

otherwise.

Thus, by direct calculation it turns out that the solution E'_* of (28) is

$$E'_* = DG + GD - 2DGD$$

and the solution of (27) is $E_* = QE'_*Q^T$. Now,

$$\begin{aligned} QE'_*Q^T &= (QDQ^T)(QGQ^T) + (QGQ^T)(QDQ^T) \\ &\quad - 2(QDQ^T)(QGQ^T)(QDQ^T) \\ &= YA + AY - 2YAY. \end{aligned}$$

Therefore, (25) is proved. The statement (26) is a direct consequence of (25). □

In the following lemma we show that the eigenvalues of the matrices lying on $T(Y)$ are well separated. In fact, N of these eigenvalues are greater than or equal to 1 and the remaining $K - N$ eigenvalues are nonpositive. This property will make it easy the process of computing a feasible point in the context of the Inexact Restoration algorithm.

Lemma 3.3 *Let $Y \in \mathcal{G}$ and $B \in T(Y)$ (with $K \geq 2N$). Then, the eigenvalues of B are given by*

$$\{-\epsilon_N, -\epsilon_{N-1}, \ldots, -\epsilon_1, \underbrace{0, \ldots, 0}_{K-2N}, 1+\epsilon_1, 1+\epsilon_2, \ldots, 1+\epsilon_N\}, \tag{30}$$

where $\epsilon_i \geq 0$, for all $i = 1, \ldots, N$.

Proof As in (29), we have that the eigenvalues of $B \in T(Y)$ are the eigenvalues of the matrix $W = \begin{pmatrix} I & R^T \\ R & 0 \end{pmatrix}$, for some matrix $R \in \mathbb{R}^{(K-N)\times N}$. Let $\zeta_1, \zeta_2, \ldots, \zeta_N$ be the singular values of R (in nondecreasing order). So, $\zeta_1^2, \ldots, \zeta_N^2$ are the eigenvalues of $R^T R$. Let λ be an eigenvalue of W. Since $\zeta_i^2 \geq 0$, for all $i = 1, \ldots, N$ there exists $\epsilon_i \geq 0$ such that $\zeta_i^2 = \epsilon_i + \epsilon_i^2$. Then, if $v = [v_1^T, v_2^T]^T$ is the eigenvector associated to λ, we have that $R^T v_2 = (\lambda - 1)v_1$ and $Rv_1 = \lambda v_2$. Thus,

$$\lambda(\lambda - 1) = \zeta_i^2 = \epsilon_i + \epsilon_i^2,$$

for some $i \in \{1, \ldots, N\}$. So,

$$\lambda = 1 + \epsilon_i \quad \text{or} \quad \lambda = -\epsilon_i.$$

Therefore, for every singular value ζ_i we have two eigenvalues of B: $\lambda_{i_1} = 1 + \epsilon_i$ and $\lambda_{i_2} = -\epsilon_i$. Since the dimension of the null-space of B is the same as the dimension of the null-space of R^T, the desired result follows. □

Observe that if $N \leq K < 2N$ in Lemma 3.3, the result follows in the same way, changing (30) by

$$\{-\epsilon_{K-N}, \ldots, -\epsilon_1, \underbrace{1, \ldots, 1}_{2N-K}, 1 + \epsilon_1, \ldots, 1 + \epsilon_{K-N}\}, \tag{31}$$

where $\epsilon_i \geq 0$, for all $i \in \{1, \ldots, K - N\}$.

Theorem 3.1 *Let $Y_* \in \mathcal{G}$. Assume that $Y_* = C_* C_*^T$, where $C_* \in \mathbb{R}^{K\times N}$ has orthonormal columns. The following statements are equivalent*:

1. *Y_* satisfies the KKT conditions of problem (20).*
2. *$Y_* \nabla f(Y_*) + \nabla f(Y_*) Y_* - 2Y_* \nabla f(Y_*) Y_* = 0$.*
3. *$Y_* \nabla f(Y_*) = Y_* \nabla f(Y_*) Y_*$.*
4. *$\nabla f(Y_*) Y_* = Y_* \nabla f(Y_*) Y_*$.*
5. *$Y_* \nabla f(Y_*) - \nabla f(Y_*) Y_* = 0$.*
6. *$\nabla f(Y_*) C_* = C_* H$ for some $H \in \mathbb{R}^{N\times N}$ symmetric.*
7. *C_* satisfies the KKT conditions of the problem*

$$\text{Minimize } f(CC^T) \quad \text{subject to} \quad C^T C = I. \tag{32}$$

8. *$Y_* \nabla f(Y_*)$ is symmetric.*

Proof Statement 1 is equivalent to Statement 2 since the latter says that the projection of $\nabla f(Y_*)$ onto the tangent subspace $\mathcal{S}(Y)$ is null. Moreover, Statement 3 is obviously equivalent to Statement 4.

Now, assume that Statement 2 holds. Post-multiplying by Y_* and using that $Y_* Y_* = Y_*$, we obtain:

$$Y_* \nabla f(Y_*) Y_* + \nabla f(Y_*) Y_* - 2Y_* \nabla f(Y_*) Y_* = 0.$$

Therefore, Statement 3 holds. Statement 3 and Statement 4 obviously imply Statement 5.

Pre-multiplying Statement 5 by Y_* we obtain Statement 3.

Therefore, Statements 3, 4 and 5 are equivalent.

Now, adding Statements 3 and 4 we obtain Statement 2. So, the first 5 statements are equivalent.

Let us prove that Statement 5 implies Statement 6. By Statement 5,

$$\nabla f(Y_*)C_*C_*^T = C_*C_*^T\nabla f(Y_*).$$

Post-multiplying by C_* we get:

$$\nabla f(Y_*)C_* = C_*C_*^T\nabla f(Y_*)C_*.$$

Therefore, Statement 6 follows taking $H = C_*^T\nabla f(Y_*)C_*$, which is symmetric.

Reciprocally, assume that Statement 6 holds. Then,

$$\nabla f(Y_*)C_* = C_*H.$$

Pre-multiplying by C_*^T we get:

$$C_*^T\nabla f(Y_*)C_* = H.$$

Therefore, by Statement 6,

$$\nabla f(Y_*)C_* = C_*C_*^T\nabla f(Y_*)C_* = Y_*\nabla f(Y_*)C_*.$$

Post-multiplying by C_*^T we obtain:

$$\nabla f(Y_*)Y_* = Y_*\nabla f(Y_*)Y_*.$$

This is Statement 4.

Statement 6 is equivalent to say that the KKT conditions of (32) holds.

Finally, Statement 8 is obviously equivalent to Statement 5. This completes the proof. □

As we showed before, the number of constraints of (20) is larger than the number of variables and standard constraint qualifications do not hold. Therefore, the fact that local minimizers are KKT points is not guaranteed by classical optimization theory and, so, this fact needs a specific proof. The proof is given in the following theorem.

Theorem 3.2 *Assume that Y_* is a local minimizer of* (20). *Then, Y_* satisfies the KKT conditions of this problem.*

Proof Since Y_* is a local minimizer of (20), there exists $\varepsilon > 0$ such that $f(Y) \geq f(Y_*)$ for all feasible Y such that $\|Y - Y_*\| \leq \varepsilon$. Assume that $Y_* = C_*C_*^T$, where $C_* \in \mathbb{R}^{K\times N}$ and $C_*^TC_* = I$. By continuity, there exists $\delta > 0$ such that, for all $C \in \mathbb{R}^{K\times N}$ such that $\|C - C_*\| \leq \delta$, one has $\|CC^T - Y_*\| \leq \varepsilon$. Therefore, C_* is a local

minimizer of (32). Since the gradients of the constraints of (32) satisfy the constant-rank constraint qualification [21], C_* must be a KKT point of (32). By Theorem 3.1, Y_* is a KKT point of (20). □

In the context of the Inexact Restoration method we will need to project a matrix belonging to $T(Y_k)$ onto the feasible set $\mathcal{G}$. Below we give an explicit formula for this projection and we prove that, using this formula, the bound (13) holds with $\beta = 1$.

Proposition 3.1 *Let $Z \in \mathbb{R}^{K\times K}$, $Z = Z^T$. Assume that $Z = Q\Sigma Q^T$ where $Q^T Q = QQ^T = I$, and Σ is diagonal with elements in non-increasing order $\sigma_1, \ldots, \sigma_K$. Define*

$$\bar{S} = \mathrm{Diag}(\underbrace{1, \ldots, 1}_{N}, \underbrace{0, \ldots, 0}_{K-N}).$$

Then, $Q\bar{S}Q^T$ is a solution of

$$\text{Minimize } \|X - Z\|_F \quad \text{subject to} \quad XX - X = 0, X = X^T, \text{ Trace } (X) = N. \tag{33}$$

Conversely, every solution of (33) has the form $Q\bar{S}Q^T$ if Q and Σ are such that $Z = Q\Sigma Q^T$, $QQ^T = I$ and $\Sigma = \mathrm{Diag}(\sigma_1, \ldots, \sigma_K)$ with $\sigma_1 \geq \cdots \geq \sigma_K$.

Proof See [1], pp. 90–91. □

Theorem 3.3 *Assume that $Y \in \mathcal{G}$, $Z \in T(Y)$ and X is a solution of (33). Then,*

$$\|X - Z\|_F \leq \|ZZ - Z\|_F. \tag{34}$$

Proof Since $\|Q^T AQ\|_F = \|A\|_F$ whenever $Q \in \mathbb{R}^{K\times K}$ is orthogonal, we may restrict ourselves to the case in which Z is diagonal. By Lemma 3.3, without loss of generality, we may write

$$Z = \mathrm{Diag}(\sigma_1, \ldots, \sigma_N, \sigma_{N+1}, \ldots, \sigma_K),$$

where $\sigma_i \geq 1$ for all $i = 1, \ldots, N$ and $\sigma_i \leq 0$ otherwise. Then, by Proposition 3.1, the solution X is given by:

$$X = \mathrm{Diag}(\underbrace{1, \ldots, 1}_{N}, \underbrace{0, \ldots, 0}_{K-N}).$$

Therefore:

$$\|X - Z\|_F^2 = (\sigma_1 - 1)^2 + \cdots + (\sigma_N - 1)^2 + \sigma_{N+1}^2 + \cdots + \sigma_K^2 \tag{35}$$

and

$$\|ZZ - Z\|_F^2 = (\sigma_1^2 - \sigma_1)^2 + \cdots + (\sigma_N^2 - \sigma_N)^2 + (\sigma_{N+1}^2 - \sigma_{N+1})^2 + \cdots + (\sigma_K^2 - \sigma_K)^2. \tag{36}$$

Now, if $i \leq N$, since $\sigma_i \geq 1$ we have:

$$|\sigma_i - 1| \leq \sigma_i |\sigma_i - 1|,$$

then,

$$(\sigma_i - 1)^2 \leq (\sigma_i^2 - \sigma_i)^2. \tag{37}$$

If $i > N$, since $\sigma_i \leq 0$, we have:

$$|\sigma_i| \leq |\sigma_i||\sigma_i - 1|.$$

Therefore,

$$\sigma_i^2 \leq (\sigma_i^2 - \sigma_i)^2. \tag{38}$$

By (35), (36), (37) and (38) we have that

$$\|X - Z\|_F \leq \|ZZ - Z\|_F.$$

This completes the proof. □

4 Inexact Restoration method for the matricial problem

The results of Sect. 3 allow us to define a suitable IR method for solving (20). The projection formula given by Lemma 3.2 makes it possible to minimize the Lagrangian in the tangent set using a reduced conjugate-gradient approach. The idea is the following: For computing the tangent direction d^k at Step 3 of Algorithm 2.1, we proceed minimizing the Lagrangian subject to the condition

$$\nabla h(y^k)^T d = 0.$$

Suppose, for a moment, that the columns of the matrix Z form an orthonormal basis of the null-space of $\nabla h(y^k)^T$. Then, the minimization of the Lagrangian reduces to the unconstrained minimization of a function whose variables are the coefficients of d with respect to the basis Z. The conjugate-gradient method may be applied to this problem, but the drawback is that the computation of Z involves an unaffordable factorization of the matrix $\nabla h(y^k)^T$. Fortunately, the availability of the projection formula (25) allows us to apply the conjugate gradient method for the tangent subproblem without explicit knowledge of the basis Z. In fact, it can be easily verified that, if one applies formally the CG method in the variables d, replacing gradients by projected gradients, one obtains exactly the iterates produced by the same method when the variables are the coefficients on the basis Z.

In the context of problem (20) we define:

$$\Omega = \{X \in \mathbb{R}^{K \times K} | X = X^T, \text{ Trace } (X) = N\}.$$

As in (8), we define the Lagrangian $L(X, \Lambda)$ by:

$$L(X, \Lambda) = f(X) + \langle X^2 - X, \Lambda \rangle$$

for all $X \in \Omega$, $\Lambda \in \mathbb{R}^{K\times K}$.

Therefore,

$$\nabla_X L(X, \Lambda) = \nabla f(X) + X\Lambda + \Lambda X - \Lambda. \tag{39}$$

A critical pair $(\bar{X}, \bar{\Lambda}) \in \Omega \times \mathbb{R}^{K\times K}$ for problem (20) is defined by

$$\bar{X}^2 - \bar{X} = 0$$

and

$$P_\Omega(\bar{X} - \nabla_X L(\bar{X}, \bar{\Lambda})) - \bar{X} = 0. \tag{40}$$

Since Ω is an affine subspace in this case, condition (40) is equivalent to:

$$P_{S_0}(\nabla_X L(\bar{X}, \bar{\Lambda})) = 0,$$

where

$$S_0 = \{X \in \mathbb{R}^{K\times K} | X = X^T, \text{ Trace } (X) = 0\}.$$

Algorithm 4.1 Inexact Restoration for solving (20)

Step 0. Initialization

The algorithmic parameters are $\gamma, \tau, \mu > 0$. As initial approximation we choose a symmetric matrix $X_0 \in \mathbb{R}^{K\times K}$ such that Trace $(X_0) = N$. We initialize $\theta_{-1} \in (0, 1)$ and $k \leftarrow 0$.

Step 1. Restoration step

Compute $Y_k \in \mathcal{G}$ as a solution of

$$\text{Minimize } \|X_k - Y\|_F \quad \text{subject to} \quad Y \in \mathcal{G}. \tag{41}$$

Step 2. Penalty parameter

Compute θ_k as the first element θ of the sequence $\{\theta_{k-1}/2^j\}_{j\in\mathbb{N}}$ such that

$$\theta f(Y_k) \leq \theta f(X_k) + \left(1 - \theta - \frac{1}{2}\right)\|X_k^2 - X_k\|_F. \tag{42}$$

Step 3. Tangent descent direction

Compute $E_k \in \mathcal{S}(Y_k)$ such that

$$f(Y_k + tE_k) \leq f(Y_k) - \gamma t\|E_k\|_F^2 \tag{43}$$

for all $t \in [0, \tau]$ and

$$\|E_k\|_F \geq \mu\|P_{\mathcal{S}(Y_k)}(\nabla f(Y_k))\|_F. \tag{44}$$

Step 4. Acceptance of the step

Compute t_k as the first element t of the sequence $\{1, 1/2, 1/4, \ldots\}$ such that

$$\begin{aligned}\theta_k f(Y_k + tE_k) + (1-\theta_k)\|(Y_k + tE_k)^2 - (Y_k + tE_k)\|_F \\ \leq \theta_k f(X_k) + (1-\theta_k)\|X_k^2 - X_k\|_F - \frac{1}{2}\|X_k^2 - X_k\|_F\end{aligned} \tag{45}$$

and

$$f(Y_k + t_k E_k) \leq f(Y_k) - \gamma t_k \|E_k\|_F^2.$$

Set $X_{k+1} = X_k + t_k E_k$, update $k \leftarrow k+1$ and go to Step 1.

Remarks Algorithm 4.1 is a particular case of Algorithm 2.1, with the substitutions $x^k = \text{vec}(X_k)$, $y^k = \text{vec}(Y_k)$, $d^k = \text{vec}(E_k)$. In fact, by (41) and Theorem 3.3, conditions (12) and (13) hold with $r_k = 0$ and $\beta = 1$ for all $k \in \mathbb{N}$. Condition (14) follows straightforwardly from (42). Moreover, (15) corresponds to (43) and $E_k \in \mathcal{S}(Y_k)$ implies that (16) also holds. Finally, the requirement (45) is equivalent to (17). These equivalences allow us to prove global convergence in Theorem 4.1.

The only assumption of Theorem 4.1 is that Step 3 of Algorithm 4.1 is well defined. This condition is satisfied, for example, if one chooses $E_k = -P_{S(Y_k)}(\nabla f(Y_k))$. We aim to use more sophisticated search directions, so, as we will see later, we compute an alternative matrix E_k first and we test the conditions (43) and (44). If at least one of these conditions is not satisfied we rely on the projected gradient choice.

Theorem 4.1 *Assume that, for all $k \in \mathbb{N}$, Step 3 of Algorithm 4.1 is well defined. Then*:

1. *For all $k \in \mathbb{N}$, the iterate X_{k+1} is well defined.*
2. *There exists $k_0 \in \mathbb{N}$ such that $\theta_k = \theta_{k_0}$ for all $k \geq k_0$.*
3. $\lim_{k\to\infty} \|X_k^2 - X_k\| = \lim_{k\to\infty} \|Y_k^2 - Y_k\| = 0$.
4. $\lim_{k\to\infty} \|E_k\| = 0$.
5. *Every limit point of $\{X_k\}$ is a KKT point of (20).*

Proof By Theorem 2.1, due to the equivalence between Algorithms 2.1 and 4.1, we only need to prove that every limit point of $\{X_k\}$ is a KKT point of (20). Recall that the sequences $\{X_k\}$ and $\{Y_k\}$ admit the same limit points. Let Y_* be a limit point. Since $\|X_k^2 - X_k\| \to 0$ we have that $Y_* \in \mathcal{G}$. By (44), since $\|E_k\| \to 0$ we have that

$$\lim_{k\to\infty} \|P_{\mathcal{S}(Y_k)}(\nabla f(Y_k))\|_F = 0.$$

Therefore, by Lemma 3.2,

$$\lim_{k\to\infty} Y_k \nabla f(Y_k) + \nabla f(Y_k) Y_k - 2Y_k \nabla f(Y_k) Y_k = 0.$$

Thus, by the continuity of ∇f,

$$Y_* \nabla f(Y_*) + \nabla f(Y_*) Y_* - 2Y_* \nabla f(Y_*) Y_* = 0.$$

Then, by Theorem 3.1, Y_* is a KKT point of (20). This completes the proof. □

The following theorem, which follows from Theorem 2.2, is a local convergence result that states that, ultimately, the convergence is superlinear or even quadratic. Quadratic convergence takes place if one solves exactly the tangent minimization subproblems. Note that in the restoration step of Algorithm 4.1 we used "exact restoration" ($r_k = 0$). The reason for this is that the eigenvalue characterization of the tangent subspace is crucial for the theoretical behavior of the algorithm and this characterization holds if Y_k is feasible. Due to the structure of the constraints, restoration is simple, so that we can stop the restoration procedure guaranteeing high precision, as we will see in the following section.

The hypotheses of Theorem 4.2 deserve some comments. Condition (46) essentially says that our solution of the subproblem at Step 3 should be an approximate minimizer of the Lagrangian, when this Lagrangian is a convex function on the tangent subspace. This generally occurs in the vicinity of a local minimizer. Condition (47) is a stability condition that generally holds close to local minimizers. Neither (46) nor (47) can be theoretically guaranteed but both are plausible properties that can be observed in practice.

As we mentioned after the statement of Theorem 2.2, the most serious hypothesis is that, eventually, $t_k = 1$. This means that inequality (45) should hold with $t = 1$ for k large enough. This property has been frequently observed in practice but its plausibility is not so clear as in the case of the remaining assumptions of the theorem. The possible lack of fulfillment of this assumption is related to a phenomenon known as "Maratos effect" in constrained optimization [19]. The Maratos effect takes place when the iterate that leads to high rates of convergence is not accepted by the merit function that guarantees global convergence. Possible consequences of the Maratos effect in our problem will be observed in Sect. 8.

Theorem 4.2 *Suppose that Step* 3 *of Algorithm* 4.1 *is well defined for all* $k \in \mathbb{N}$. *Assume that there exist* $c > 0, \eta \in [0, 1)$ *such that for all* $k \in \mathbb{N}$ *there exist* $\Lambda^k \in \mathbb{R}^{K\times K}$, $\eta_k \in [0, \eta]$ *such that*

$$\|P_{S_0}[\nabla L(Y_k + E_k, \Lambda_{k+1})]\|_F \leq \eta_k \|P_{S_0}[\nabla L(Y_k, \Lambda_k)]\|_F \tag{46}$$

and

$$\|E_k\|_F + \|\Lambda_{k+1} - \Lambda_k\|_F \leq c\|P_{S_0}[\nabla L(Y_k, \Lambda_k)]\|_F. \tag{47}$$

Assume, finally, that $(\bar{X}, \bar{\Lambda})$ *is a stationary pair and that* $t_k = 1$ *for* k *large enough.*

Then, there exist $\delta, \varepsilon > 0$ *such that*:

- *If, for some* $k_0 \in \mathbb{N}$, $\|X_{k_0} - \bar{X}\|_F \leq \varepsilon$ *and* $\|\Lambda_{k_0} - \bar{\Lambda}\|_F \leq \delta$, *the sequence* (X_k, Λ_k) *converges to some stationary pair.*
- *If* η_k *tends to zero, the convergence is R-superlinear.*
- *If* $\eta = 0$ *the convergence is R-quadratic.*

5 Restoration without diagonalization

Proposition 3.1 gives a characterization of the solution of the restoration problem (41). Using this characterization one may obtain Y_k at Step 1 of Algorithm 4.1 by

means of the diagonalization of X_k. Alternative restoration procedures that do not involve eigenvalue calculations are described in this section. Recall that, for $k > 1$ one has $X_k \in T(Y_{k-1})$. Therefore, by Lemma 3.3, X_k has N eigenvalues greater than or equal to 1 and $K - N$ eigenvalues less than or equal to 0.

Given $X_k \in T(Y_{k-1})$, an iterative procedure for computing Y_k may be defined by $Y_k^0 = X_k$, and

$$Y_k^{j+1} = Y_k^j - (2Y_k^j - I)^{-1}[(Y_k^j)^2 - Y_k^j] \tag{48}$$

for all $j \in \mathbb{N}$.

Writing $Y_k^j = Q D_k^j Q^T$ ($QQ^T = Q^T Q = I$) it is easy to see that (48) corresponds to the application of Newton's method to each eigenvalue of the matrix, for solving the equation $\varphi(d_i) \equiv d_i^2 - d_i = 0$. Since $Y_k^0 \in T(Y_{k-1})$ we have that the iteration (48) converges to the solution of $Y^2 - Y = 0$ which is closest to X_k.

From now on, let us write, for the sake of simplicity, $x = d_i$. If $x \in \{0, 1\}$ we see that $\varphi'(x) = 1/\varphi'(x)$. This identity suggests the use of the iteration

$$x_{j+1} = x_j - \varphi'(x_j)\varphi(x_j) \tag{49}$$

instead of the usual Newton iteration, for solving $x^2 - x = 0$. Clearly, (49) corresponds to the matricial iteration

$$Y_k^{j+1} = Y_k^j - (2Y_k^j - I)[(Y_k^j)^2 - Y_k^j]. \tag{50}$$

Developing (49) and (50) we arrive to

$$x_{j+1} = 3x_j^2 - 2x_j^3 \tag{51}$$

and

$$Y_k^{j+1} = 3(Y_k^j)^2 - 2(Y_k^j)^3, \tag{52}$$

respectively.

The convergence properties of (52) can be directly deduced from the ones of (51), which are the following:

1. If $x_0 \in [-1/2, 3/2]$ the sequence $\{x_j\}$ is convergent. The possible limit points are $\{0, 1/2, 1\}$ and the convergence is quadratic [29]. If $x_0 \notin [-1/2, 3/2]$ the sequence diverges.
2. If $x_0 \in [-1/2, \frac{1-\sqrt{3}}{2})$ we have that x_j tends to 0. If $x_0 \in (\frac{1+\sqrt{3}}{2}, 3/2]$, we have that x_j tends to 1. If $x_0 = \frac{1-\sqrt{3}}{2}$ or $x_0 = \frac{1+\sqrt{3}}{2}$ the sequence converges to $1/2$ in just one iteration.
3. The sequence also converges to 0 starting from $x_0 \in (\frac{1-\sqrt{3}}{2}, 1/2) \approx (-0.366, 0.5)$ and converges to 1 starting from $x_0 \in (1/2, \frac{1+\sqrt{3}}{2}) \approx (0.5, 1.366)$.

A consequence of these properties for the iteration (52) is that Y_k^j converges quadratically to the projection matrix that is closest to Y_k^0 if N eigenvalues of Y_k^0 are in $(1/2, \frac{1+\sqrt{3}}{2}) \approx (0.5, 1.366)$ and $K - N$ eigenvalues of Y_0 are in $(\frac{1-\sqrt{3}}{2}, 1/2) \approx$

$(-0.366, 0.5)$. In any other case there are no chances that (52) converges to the correct projection.

Recall, from (30) and (31), that $X_k \in T(Y_{k-1})$ has N eigenvalues greater than or equal to 1 and $K - N$ eigenvalues non-positive, guaranteeing always the convergence of (48). However, since at every step of Newton iteration we have to solve a linear system, it would be interesting to apply the iterative process (52) whenever possible. The following result establishes a scheme which always allows us to apply (52) for solving (41), regardless of the matrix $X_k \in T(Y_{k-1})$.

Theorem 5.1 *Let $Y_{k-1} \in \mathcal{G}$, $X_k \in T(Y_{k-1})$ and c_{bound} be a strict upper bound of the eigenvalues of X_k. Define*

$$\widetilde{X}_k = \frac{1}{2c_{bound} - 1}\big(X_k + (c_{bound} - 1)I_{K\times K}\big). \tag{53}$$

Then, the iterative process (52), starting with $Y_k^0 = \widetilde{X}_k$, converges quadratically to the solution of (41).

Proof From (30) and (31), we have that $c_{lower} \equiv 1 - c_{bound}$ is a strict lower bound of the eigenvalues of X_k. Thus, defining

$$\widetilde{X}_k = \frac{1}{c_{bound} - c_{lower}}\big(X_k - (1 - c_{bound})I_{K\times K}\big),$$

it follows that $\widetilde{X}_k$ has N eigenvalues in $(1/2, 1]$ and $K - N$ eigenvalues in $[0, 1/2)$. Then, from the analysis above, related to (52), and taking into account that the eigenvectors of X_k and $\widetilde{X}_k$ are the same, the desired result follows. □

The iteration (52) was introduced by Mc Weeny [30] and is generally known as "purification" in the electronic structure literature (see, for example, [29, 31–33]). Our contribution here is to show that, due to the eigenvalue structure of the tangent subspace at Y_{k-1} (Lemma 3.3) this iterative process is not only locally quadratically convergent but also globally convergent starting from any point of $T(Y_{k-1})$, after the transformation (53). This implies that the purification process may be safely used, not only to refine an approximate solution when we know that we are close to an exact one, but also as a routine restoration procedure starting from the previous tangent subspace. Moreover, global convergence occurs to the closest point to X_k, which guarantees that convergence requirements of the Inexact Restoration methods (condition (13)) are fulfilled.

6 Estimation of the Lagrange multipliers

In the Inexact Restoration framework there are different implementation alternatives for both the feasibility and the optimality phases. For decreasing the overall number of iterations the best choice for the optimality phase is to (approximately) minimize the Lagrangian of the problem on the tangent subspace. This requires to compute suitable Lagrange multipliers estimates at every iteration.

Let us assume that a set of Lagrange multipliers $\Lambda \in \mathbb{R}^{K\times K}$, $\Lambda = \Lambda^T$, has been computed.

Therefore, in the optimality phase of the IR method we need to minimize the Lagrangian defined by

$$L(X,\lambda) = f(X) + \sum_{i,j=1}^{K} \lambda_{ij}(X^2 - X)_{ij}$$

on the affine subspace $T(Y)$. Observe that, in the case that f is quadratic (as in the classical Hartree-Fock), the Lagrangian L is quadratic too and, so, the optimality phase is a quadratic programming problem with equality constraints.

The gradient of $L(X,\Lambda)$ is given by the following formula:

$$\frac{\partial L}{\partial x_{ij}} = \frac{\partial f}{\partial x_{ij}} + \sum_{k=1}^{K}(\lambda_{ik}x_{kj} + x_{ik}\lambda_{kj}) - \lambda_{ij}.$$

Therefore,

$$\nabla_X L(X,\Lambda) = \nabla f(X) + X\Lambda + \Lambda X - \Lambda. \tag{54}$$

Now let us address the problem of computing Lagrange multipliers estimators. The reasoning is as follows. As it is well known, the purpose of using the Lagrangian in the tangent subspace is that this function approximates the true objective function on the true feasible set. Given $X \in T(Y)$, the value of the Lagrangian at X should approximate the value of f at a feasible point that is close to X. Now, the most straightforward approximation of a feasible point that is close to X is the Newtonian iterate:

$$\bar{X} = X - (2X - I)^{-1}(X^2 - X).$$

The value of f at $\bar{X}$ is, approximately, given by:

$$f(\bar{X}) \approx f(X) + \langle \nabla f(X), \bar{X} - X\rangle.$$

But, since $\mathrm{Trace}(AB) = \mathrm{Trace}(BA)$,

$$\begin{aligned}\langle \nabla f(X), \bar{X} - X\rangle &= \langle \nabla f(X), -(2X - I)^{-1}(X^2 - X)\rangle \\ &= \mathrm{Trace}[\nabla f(X)(-(2X - I)^{-1}(X^2 - X))^T] \\ &= -\mathrm{Trace}[\nabla f(X)(X^2 - X)(2X - I)^{-1}] \\ &= -\mathrm{Trace}[(2X - I)^{-1}\nabla f(X)(X^2 - X)]. \end{aligned}\tag{55}$$

On the other hand,

$$\begin{aligned}\mathrm{Trace}[(2X - I)^{-1}\nabla f(X)(X^2 - X)] &= \mathrm{Trace}[((2X - I)^{-1}\nabla f(X)(X^2 - X))^T] \\ &= \mathrm{Trace}[(X^2 - X)\nabla f(X)(2X - I)^{-1}] \\ &= \mathrm{Trace}[\nabla f(X)(2X - I)^{-1}(X^2 - X)]. \end{aligned}\tag{56}$$

By (55) and (56),

$$\begin{aligned}
&\langle \nabla f(X), \bar{X} - X \rangle \\
&\quad = -\text{Trace}\left[\frac{(\nabla f(X)(2X-I)^{-1} + (2X-I)^{-1}\nabla f(X))}{2} (X^2 - X) \right] \\
&\quad = -\text{Trace}\left[\frac{(2X-I)^{-1}\nabla f(X) + [(2X-I)^{-1}\nabla f(X)]^T}{2} (X^2 - X) \right] \\
&\quad = -\left\langle \frac{(2X-I)^{-1}\nabla f(X) + [(2X-I)^{-1}\nabla f(X)]^T}{2}, (X^2 - X) \right\rangle.
\end{aligned}$$

Since we wish to compute the Lagrange multipliers approximation only once along the optimality phase, we adopt the formula

$$\Lambda = -\frac{(2Y-I)^{-1}\nabla f(Y) + [(2Y-I)^{-1}\nabla f(Y)]^T}{2}, \tag{57}$$

where $Y = Y_k$ is the restored point at the k-th iteration of the IR process.

Moreover, it is trivial to see that, if $Y^2 = Y$ we have that $(2Y-I)^{-1} = 2Y - I$. This identity suggests that we may also use the approximation

$$\Lambda = -\frac{(2Y-I)\nabla f(Y) + [(2Y-I)\nabla f(Y)]^T}{2}. \tag{58}$$

7 Implementation

In this section we describe an implementation of Algorithm 4.1. For simplicity, we restrict ourselves to the case in which f is quadratic, as in the Hartree-Fock case.

We chose $\theta_{-1} = 0.999$. After the computation of $Y_k \in \mathcal{G}$ at Step 1 of the algorithm, we compute

$$G_k = Y_k \nabla f(Y_k) - \nabla f(Y_k) Y_k.$$

If the maximal absolute value of the entries of G_k is smaller than or equal to 10^{-8} we interrupt the execution declaring convergence. By Theorem 3.1, Y_k is an approximate KKT point.

7.1 Restoration

The restoration step (41) may be performed by means of (33), which involves an eigenvalue calculation, or by means of the globally convergent Newton-like methods described in Sect. 5. Algorithm 4.1 does not compute eigenvalues at all when the Newton restoration procedure is adopted.

7.2 Computation of the tangent descent direction

For computing E_k at Step 3 of the algorithm we consider the subproblem

$$\text{Minimize } Q_k(E) \quad \text{subject to} \quad E \in \mathcal{S}(Y_k), \tag{59}$$

where $Q_k(E) \equiv L(Y_k + E, \Lambda_k)$. ($\Lambda_k$ is computed using (58) with $Y = Y_k$.) Taking an orthonormal basis of $\mathcal{S}(Y_k)$ and expressing (59) in terms of the corresponding internal coordinates, the problem (59) becomes unrestricted and we can consider the Hestenes-Stiefel conjugate gradient method (CG) for its resolution. Fortunately, we do not need to compute a basis for applying CG. In fact, we may employ, formally, the CG method using gradient projections instead of the reduced-basis gradients of Q_k, obtaining exactly the same iterates. In this way, we get, in a finite number of iterations, a solution of (59) or a nonpositive-curvature direction. If a nonpositive-curvature direction E is computed at the first CG iteration we return $E_k = tE$ satisfying $t \geq 0$ and $\|Y_k + E_k\|_F^2 = 3N$. (Recall that, for all $X \in \mathcal{G}$, one has $\|X\|_F^2 = N$.) Otherwise, we stop the CG procedure at the last iterate obtained before finding nonpositive-curvature directions. In the vicinity of a local minimizer of (20), the Lagrangian restricted to the tangent subspace is convex. In this case, the solution of (59) is obtained in finitely many iterations.

Given the approximate solution of (59) so far obtained, we test the conditions (44) (with $\mu = 10^{-6}$) and (43). Condition (43) is tested indirectly, since it obviously holds, for some $\gamma > 0$, if (44) holds and E_k is a sufficient descent direction. Therefore, instead of (43) we test

$$\langle E_k, P_{\mathcal{S}(Y_k)}(\nabla f(Y_k))\rangle \leq -10^{-6}\|E_k\|_F \|P_{\mathcal{S}(Y_k)}(\nabla f(Y_k))\|_F. \tag{60}$$

If both (43) and (44) are fulfilled, we accept the direction E_k. Otherwise, we replace E_k by the projection of $-\nabla f(Y_k)$ on $\mathcal{S}(Y_k)$.

Note that the projection formula (25) is the main tool for the implementation of the procedure described here.

8 Numerical experiments

8.1 Mathematical test problems

In order to test the reliability of Algorithm 4.1 we defined a set of mathematical test problems. All the problems have the form (20) and different instances of them arise according to specific parameters. Considering objective functions, instances, variable dimensions and initial points we end up defining 1450 problems.

The constraints of all the test problems are the ones given in (20). Therefore, different problems differ only in the objective function and the dimensions K and N. For all the functions we assume that $f(0) = 0$.

Function 1 We define, for all $X \in \mathbb{R}^{K\times K}$, $x = \text{vec}(X) \in \mathbb{R}^n$. We denote by A the tridiagonal $(-1, 2, -1)$ $n \times n$ matrix and $b = (-1, 0, \ldots, 0, -1)^T$.

We define:

$$F_{vec}(x) = b^T x + \frac{1}{2} x^T A x, \tag{61}$$

$$F(X) = F_{vec}(\text{vec}(X)) \tag{62}$$

and

$$f(X) = F\left(\frac{X + X^T}{2}\right). \tag{63}$$

Function 2 This is a trivial problem with a linear objective function such that

$$\nabla f(X) = \text{Diag}(\underbrace{-1, \ldots, -1}_{N}, \underbrace{0, \ldots, 0}_{K-N}).$$

The solution of (20) is

$$X = \text{Diag}(\underbrace{1, \ldots, 1}_{N}, \underbrace{0, \ldots, 0}_{K-N}).$$

Function 3 The objective function is linear and $\nabla f(X)$ is the tridiagonal $(-1, 2, -1)$ $K \times K$ matrix T. The solution of this problem is the projection matrix on the subspace generated by the N smallest eigenvalues of the matrix T.

Function 4 The gradient of the linear objective function is the same as in Function 3, except that the first N diagonal entries of $\nabla f(X)$ are -8.

Function 5 The gradient of the objective function of this problem is the one of Function 1, subtracting 10 from the first N diagonal entries. Therefore, denoting by f_1 the objective function of Function 1,

$$f(X) = f_1(X) - 10\sum_{i=1}^{N}(X)_{ii}.$$

Function 6 Define T as in Function 3. We define:

$$(\nabla f(X))_{ij} = (T)_{ij}$$

if $i \neq j$ and

$$(\nabla f(X))_{ii} = 2(X)_{ii}$$

for all $i = 1, \ldots, K$.

Function 7 This problem is defined by

$$(\nabla f(X))_{ij} = \frac{(X)_{ij}}{i + j - 1}.$$

Function 8 The gradient of f is defined by:

$$a_{ij} = \sum_{r,s} \sin(i + j + r + s)(X)_{rs}$$

and

$$(\nabla f(X))_{ij} = \frac{a_{ij} + a_{ji}}{2}.$$

Function 9 In this problem we define A as the Hilbert $n \times n$ matrix $(A)_{ij} = 1/(i + j - 1)$. The definition of f follows as in Function 1.

Function 10 We define, for $i, j, r, s = 1, \ldots, K$,

$$a_{ijrs} = \sin(i + j) + \sin(r + s).$$

The objective function is given by:

$$f(X) = \frac{1}{2} \text{ Trace } [2HX + G(X)X], \tag{64}$$

where

$$(G(X))_{ij} = \sum_{r=1}^{K} \sum_{s=1}^{K} (2a_{ijrs} - a_{isrj})(X)_{sr}, \tag{65}$$

$$(H)_{ij} = H_{ji} = \text{ random between 0 and } 2w \tag{66}$$

and w is a parameter. Observe that $\nabla f(X) = H + G(X)$. Note that (64) and (65) define the optimization problem in closed-shell restricted Hartree-Fock equations, when X is the one-electron density matrix in the atomic-orbital basis, H is the one-electron Hamiltonian matrix and a_{ijrs} is a two-electron integral in the AO basis [4].

Function 11 The differences between this function and Function 10 are:

- The matrix H is such that $H_{ii} = 0$ for all i, and only the case $w = 1$ is considered.
- The function is evaluated in such a way that the computer time involved in this problem is considerably cheaper than the one involved in Function 10.

Function 12 We define, for $i, \ldots, 2K$,

$$z(i) = 0 \quad \text{with probability } p_1. \tag{67}$$

If $z(i) \neq 0$ we define:

$$z(i) = \text{ random between } a_1 \text{ and } b_1. \tag{68}$$

Moreover,

$$a_{ijrs} = z(i + j) + z(r + s). \tag{69}$$

The definition of the objective function is completed as in (64), (65), (66).

Function 13 In this problem, instead of (69) we define:

$$a_{ijrs} = z_{i+j} z_{r+s}. \tag{70}$$

Function 14 Define $z(i)$, for $i = 1, \ldots, 4K$, as in (67)–(68). Define

$$a_{ijrs} = z(i + j + r + s)$$

and complete as in Function 12.

Function 15 For all $i = 1, \ldots, 2K$, define

$$z(i) = \tan(i)$$

and complete as in Function 12.

Function 16 Define $z(i)$ as in Function 15 and complete as in Function 13.

Function 17 For all $i = 1, \ldots, 4K$ define $z(i) = \tan(i)$ and complete as in Function 14.

Function 18 Define $z(i) = \log_{10}(i)$ and complete as in Function 12.

Function 19 Define $z(i) = \log_{10}(i)$ and complete as in Function 13.

Function 20 Define $z(i) = \log_{10}(i)$ and complete as in Function 14.

8.2 Performance profiles

The code that produced the experiments in this section was written in Fortran and run in a computer with 2 processors Xeon E5462 Quad-Core 2.83 GHz, with 6 Mb cache; 4 Gb Ram DDR2 800 ECC.

The subroutines that compute functions and gradients were parallelized using OpenMP. The code was compiled using the ifort compiler with the keys "-O2 -openmp".

For comparing different methods we use performance profiles [34]. Each method M is represented by a curve $y = M(x)$. For each abscissa $x > 0$, $M(x)$ is the number of problems solved by the method in minimal time, where the "minimal" time definition involves a tolerance given by x. A useful interpretation of the performance profiles curves is that, roughly speaking, x represents (adimensional and problem-independent) computer time and $M(x)$ is the number of problems that would be solved by Method M if only "x time units" were available. Clearly, performance profile curves are always non-decreasing since the number of problems solved increase with time availability.

The graphics presented here differ in the way a problem is considered to be solved by a given method. One possibility is to establish that the method solved the problem when a feasible Y_k was found such that

$$\max_{i>j} |(Y_k \nabla f(Y_k))_{ij} - (Y_k \nabla f(Y_k))_{ji}| \leq 10^{-8}. \tag{71}$$

By Theorem 3.1, the criterion (71) corresponds to the approximate fulfillment of the KKT conditions of (20).

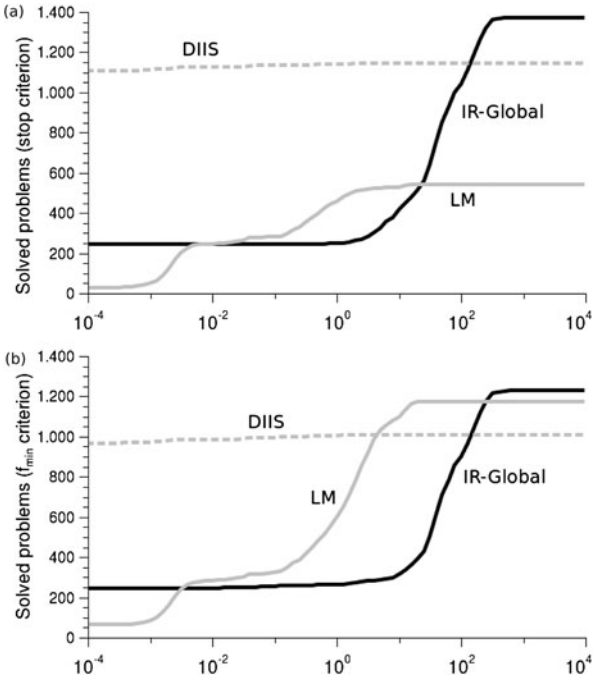

Fig. 1 Performance profiles: $K = 50$, $N = 5$ using (**a**) stopping criterion (71) and (**b**) minimum function value found (f_{min}) criterion (72)

The second possibility involves computing, for each problem, the minimum value f_{min} of the objective function obtained by any of the methods at feasible points and to declare that a particular method solved the problem when a feasible Y_k was found that

$$f(Y_k) \leq f_{min} + |f_{min}|10^{-6}. \tag{72}$$

For different values of K and N we ran 1450 instances of the mathematical problems defined above. The instances correspond to 10 different random initial points for 145 problems generated as follows:

1. The first 9 problems correspond to functions 1–9 described above.
2. Problems 10–145 were generated using (64)–(66).
3. For $j = 10, \ldots, 17$, the function j generated 8 problems, varying (66) and a_{ijrs}.
4. For $j = 18, \ldots, 20$, the function j generated 24 problems, according to the details given below.
5. In function 10, 4 matrices H were randomly defined with $w = 1$ and 4 matrices H were defined with $w = 5$.
6. In function 11, 8 matrices H were generated randomly with $w = 1$ and null diagonal.
7. For $j = 12, 13, 14$, function j generated 8 problems combining 4 choices of a_{ijrs} and 2 choices of H. The choices of a_{ijrs} came from combining $p_1 \in \{0, 0.99\}$ with $[a_1, b_1] \in \{[0, 5], [0, 500]\}$. The choices of H correspond to $w \in \{1, 5\}$ in (66).
8. For $j = 15, 16, 17$, function j generated 8 problems combining 2 choices of a_{ijrs} and 4 choices of H. The choices of a_{ijrs} correspond to $p_1 \in \{0, 0.99\}$. The choices

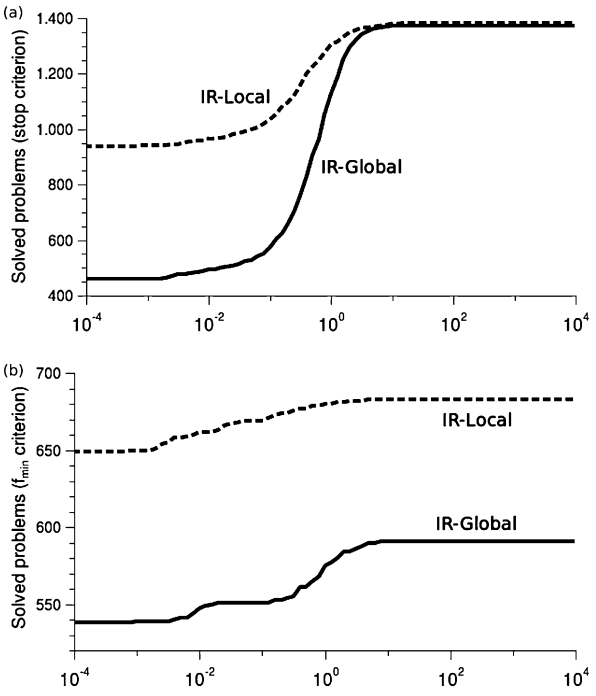

Fig. 2 Performance profiles, Local vs. Global strategies: $K = 50$, $N = 5$ using stopping criterion (71)

of H correspond to the generation of two matrices H (66) with $w = 1$ and two additional matrices H with $w = 5$.

9. For $j = 18, 19, 20$, function j generated 24 problems combining 2 choices of a_{ijrs} and 12 choices for H. The choices of a_{ijrs} correspond, as in 15–17, to $p_1 \in \{0, 0.99\}$. Finally, 6 matrices H were generated using (66) with $w = 1$ and 6 additional matrices came from taking $w = 5$.

In Fig. 1 we exhibit the comparison between the Inexact Restoration method described in this paper, the Levenberg-Marquardt (LM) (also called trust-region) method described in [9] and the classical SCF-DIIS method. In Fig. 1a we consider that a method solved the problem when it stopped at a point satisfying (71). In Fig. 1b we preserved (71) as stopping criterion, but we considered that a method solved a problem when (72) took place for some k.

For a better visualization of these figures, note that the curve value $M(x)$ at the middle point of the x-axis ($x = 10^0$) represents the number of problems solved by Method M if the maximum time allowed is twice the computer time used by the method that solves the problem fastest.

Figure 1a shows that, when minimal computer time is available, SCF-DIIS is the best method (left-wing of performance profile), followed by IR and LM. Doubling the available computer time for each problem, SCF-DIIS remains to be the method that solves the largest amount of problems, being LM now the second. However, if we allow the algorithms to run during enough computer time, IR turns out to be the algorithm that solves the largest amount of problems.

The main difference between Fig. 1a and b is related with the behavior of LM. When one considers, as in Fig. 1a, that a method "solves a problem" when it satisfies the convergence criterion (71), LM is the worst of the three methods. However, when we adopt, as solvability criterion, the minimal functional value reached by each method, LM outperforms SCF-DIIS for sufficiently large time tolerance. Giving enough time to the three methods, IR remains to be the best, being now LM the second. This reflects the fact that, many times, LM reaches in a reasonably small amount of time the smallest functional value, but it fails to achieve the KKT precision (71).

We observed that, in some problems, IR needed to use a very small value of t_k to satisfy the descent criterion (45). This motivated us to try a different (local) version of the Inexact Restoration method: Instead of performing the backtracking process (45) we set $X_{k+1} = X_k + E_k$ for all $k \in \mathbb{N}$. As a consequence, Step 2 of Algorithm 4.1 may also be skipped. The resulting algorithm is called IR-local, since its resembles the local algorithm defined in [13]. Figure 2 is the performance profile corresponding to the comparison of Algorithm 4.1 (IR-global from now on) with IR-local. We use the criterion (71) for declaring that a method solved a problem. This profile seems to show that IR-local is better than IR-global both from the point of view of efficiency and robustness in this set of problems. In fact, we also verified that IR-local tends to obtain lower functional values than IR-global. We are reluctant to recommend a merely local convergent algorithm instead of a global one based in a limited number of experiments. However, it is not unusual that a local quadratically convergent algorithm that produces bounded iterates reach the convergence basin of a solution faster than its global counterpart. Experiments comparing local and global forms of Newton's method for solving nonlinear systems frequently show this phenomenon [35].

8.3 Example with $K = 700$, $N = 70$

The experiments presented in Sects. 8.3 and 8.4 were run in a laptop with Intel Core 2 Duo 2.2 GHz processor bearing 2 Gb of RAM memory.

We consider Function 1 with $K = 700$, $N = 70$. Then, the number of variables n is 490,000 and the dimension of the tangent subspaces is $N(K - N) = 44{,}100$. Convergence of the global form of IR occurred in 142 iterations, using approximately 2 hours and 45 minutes of computer time. The final (and best) $f(X)$ obtained was 0.541707713190007.

At the first 131 IR-iterations the conjugate-gradient method in the optimality phase finished detecting "nonpositive curvature direction" using ≈ 100 CG-iterations.

At the last 11 IR-iterations CG converged using ≈ 240 CG-iterations (163 CG-iterations at the last one). The KKT values (71) at the final iterates were: 4.0×10^{-4}, 4.0×10^{-4}, 4.0×10^{-6}, 1.2×10^{-9}.

In this problem, the local IR method reached $f(X) = 0.55642673$ after 3 hours of computer time, but failed to improve this value during the next 9 hours (2297 iterations). The final KKT value was $\approx 10^{-2}$.

SCF-DIIS also failed to find an acceptable solution. The best functional value was obtained at iteration 5 ($f(X) = 28.6$) and did not improve in the next 5 hours. The

SCF method, without acceleration, got very small progress at the 858 first iterations (getting $f(X) = 35.94$) and, after that, could not improve the functional value any more.

The LM method, after 3090 iterations and 5 hours of CPU time, obtained a functional value of 0.541708936 with KKT $\approx 10^{-5}$. After 9800 iterations and 12 hours obtained $f(X) = 0.541707713272647$ with KKT $\approx 9. \times 10^{-8}$. It stopped due to impossibility of further decrease the function.

8.4 Electronic structure calculations

We studied the behavior of the Inexact Restoration algorithms in some typical electronic structure calculations arising in computational chemistry (from now on called "easy" problems) and in some designed problems known to display convergence instabilities [8, 9] (called "hard" problems). *Easy* problems are standard organic molecules Carbon dioxide, Ethylene, Ethanol and Benzene, and some common biologically relevant molecules, as Alanine, Alanine dipeptide, Histidine and Tyrosine. These *easy* examples were selected to illustrate the behavior of the algorithms in problems for which convergence is usually straightforward. On the other side, we also performed tests on some electronic structure calculations known to display multiple local minima and convergence problems: CrC, Cr_2, Rh_2 and an arrangement of atoms of formula Li_9F_9 [8, 9]. Two geometries were considered for *hard* cases: for diatomic molecules, one with atom-atom distance of 2 Å and, in addition, a distorted geometry, with atom-atom distance of 10 Å. For Li_9F_9, we used the standard and distorted geometries described in [9]. Distorted geometries are usually associated with increased convergence difficulties.

Standard 6-31G [36] and STO-3G [37] atomic orbital basis were used for *easy* and *hard* problems, respectively. The h-core (H), the overlap matrix and the two-electron integrals were computed with the GAMESS package [38], and loaded in an in-house implementation of the algorithms. The initial point in all cases is defined as $X_0 = \Psi(H)$, where Ψ is given by (5). The DIIS and LM algorithmic details are described in [9] and the implementation of the global and local IR methods was the same as described for previous examples. All algorithms were implemented in Fortran77. The gfortran version 4.2 was used for compilation with the "-O3" flag. The examples were run on a AMD Phenom 9850 Quad-Core machine with 4 Gb of RAM memory running Ubuntu Linux version 8.10.

We compared the algorithms IR-Global, IR-Local, LM and DIIS. To build Figs. 3 and 4, we define $E_{best} = E_{min} + 10^{-10}$ where E_{min} is the minimum energy solution found by all methods for each problem. In Table 1, the minimum energy found by each algorithm for each problem is compared by defining E_{sol} as the minimum energy obtained by each method and computing $10 + \log(E_{sol} - E_{best})$. This quantity is zero if the minimum energy is the minimum found for all methods, and is positive otherwise.

As can be seen in Table 1, all algorithms found the same minimum energy solution for all *easy* problems, with the exception of DIIS for Histidine, which oscillated. As expected, the LM algorithm requires much more iterations than other methods in most of these problems, but converges smoothly in all of them. IR-Local, IR-Global and DIIS are competitive when comparing the number of iterations required

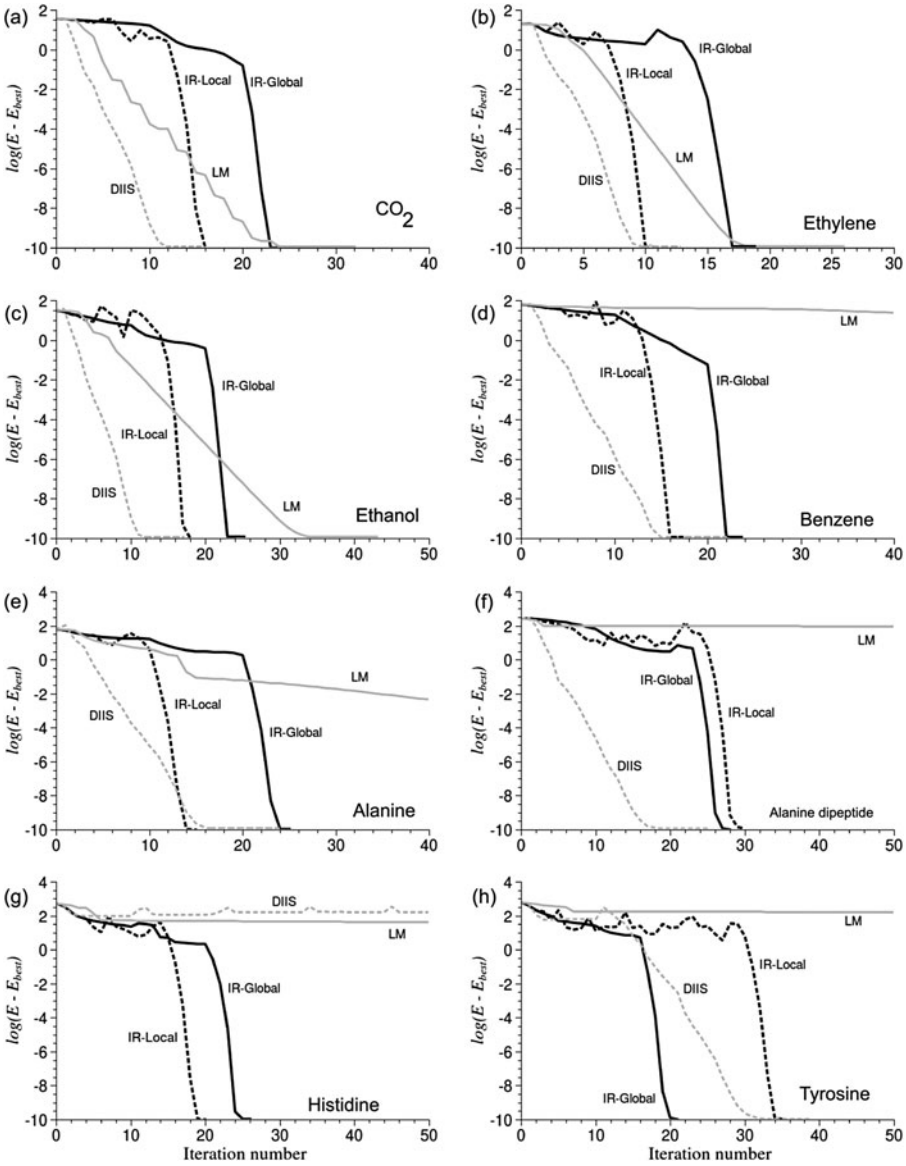

Fig. 3 *Easy* electronic structure problems

for convergence. IR-Local converged in the smallest number of iterations in 6 of the 8 problems, DIIS in 3 of 8 and IR-Global in one of them, the differences between IR-Local and DIIS being small. For the present problems and implementation, the cost of computing the gradient is larger than the cost of its spectral decomposition, in such a way that IR iterations are more costly. For larger molecules, however, the gradient becomes sparse and trivially parallelizable [39], thus using IR strategies may be advantageous.

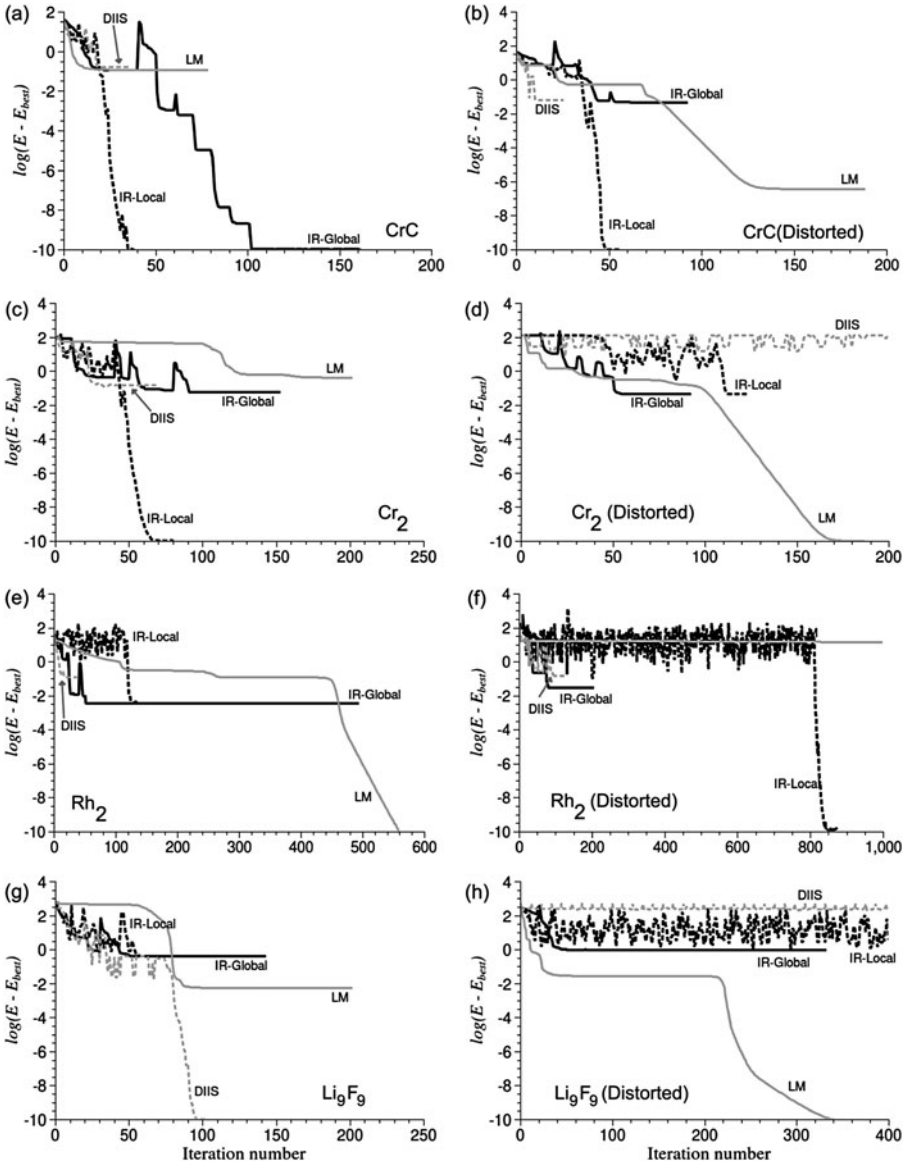

Fig. 4 *Hard* electronic structure problems

The fast local convergence of the IR algorithms is clearly visible in the sequence of iterates of Fig. 3. Far from the solution the decrease of the energy proceeds slowly, but a rapid drop in the error is obtained in final iterations. On the other side, DIIS displays an approximate linear convergence from the first iterations when it is successful, and is able to provide a more effective reduction of the energy far from the solution. LM decreases the energy monotonically but slowly. The combination of different

Table 1 Number of iterations and relative energy (logarithmic scale) for electronic structure problems. N_{it} is the number of iterations required for convergence, E_{sol} is the energy of the solution found for each method and $E_{best} = E_{min} + 10^{-10}$ where E_{min} is the minimum energy found for all methods

	Example		Dimensions (N/K)	Results: $N_{it}(10 + \log(E_{sol} - E_{best}))$			
				IR-Global	IR-Local	LM	DIIS
Easy problems	(a)	Carbon dioxide	11/27	24(0)	15(0)	32(0)	15(0)
	(d)	Ethylene	9/30	18(0)	11(0)	26(0)	13(0)
	(b)	Ethanol	13/39	24(0)	18(0)	43(0)	17(0)
	(c)	Benzene	21/66	23(0)	17(0)	115(0)	22(0)
	(e)	Alanine	24/68	25(0)	15(0)	178(0)	24(0)
	(h)	Alanine dipeptide	43/123	28(0)	30(0)	412(0)	25(0)
	(f)	Histidine	41/117	26(0)	20(0)	341(0)	[a]([a])
	(g)	Tyrosine	48/139	21(0)	35(0)	535(0)	39(0)
Hard problems	(c)	CrC	15/24	161(0)	38(0)	78(9.1)	37(9.2)
	(d)	CrC(distorted)	15/24	92(8.6)	57(0)	199(3.6)	25(8.8)
	(a)	Cr_2	48/38	152(8.8)	80(0)	201(9.6)	68(9.2)
	(b)	Cr_2(distorted)	48/38	92(8.7)	122(8.7)	186(0)	[a]([a])
	(e)	Rh_2	45/58	492(7.6)	259(7.6)	559(0)	44(9.1)
	(f)	Rh_2(distorted)	45/58	202(8.5)	871(0)	[a]([a])	122(9.2)
	(g)	Li_9F_9	54/162	142(9.6)	75(9.6)	201(7.7)	103(0)
	(h)	Li_9F_9(distorted)	54/162	331(9.9)	[a]([a])	340(0)	[a]([a])

[a]Failed to convergence in 1000 iterations

methodologies in different stages of the optimization procedure, particularly by the use of IR strategies close to the solution, seems to be promising.

In challenging electronic structure problems, summarized on Fig. 4, the behavior of the algorithms is less predictable. The solution with minimum energy was found in 1 of 8 problems by IR-Global, 4 by IR-Local, 3 by LM, and in 1 of 8 problems by DIIS. Therefore, IR-Local and LM seem to be the most successful algorithms. On the other side, the only algorithm that was able to converge within 1000 iterations in all cases was IR-Global. IR-Local and LM failed in one case[1] and DIIS failed in two cases.

Overall, the behavior of the local implementation of IR is satisfactory. However, Li_9F_9 illustrates the importance of a globalization strategy. As can be seen in Fig. 4h, DIIS exhibited oscillatory behavior, and IR-Local was unable to reach the proximity of any solution, so that it could not take advantage of fast local convergence. This erratic behavior of the IR-Local algorithm was also observed in Rh_2 (Fig. 4e) and Rh_2 (distorted) (Fig. 4f), although it finally converged. The introduction of the globalization strategy clearly stabilized the iterations, and cannot be discarded.

[1]LM converged in 1353 iterations in Rh_2 (distorted), as predicted by theory, and obtained the lowest energy solution.

9 Final remarks

We showed that the Inexact Restoration approach provides a reliable globally and quadratically convergent method for solving the class of optimization problems that appear in Closed Shell electronic calculations. Its main attractiveness comes from the fact that eigenvalue computations are not necessary. IR takes proper advantage of the structure of the underlying optimization problem. In particular, suitable projection formulae make it possible to use the CG algorithm restricted to tangent spaces without large matrix manipulations. The CG algorithm finds tangent space solutions (or nonpositive curvature directions) in a small number of iterations, relatively to the dimension of the subspace. Moreover, due to the eigenvalue structure of points in the tangent subspace we are able to define globally and quadratically convergent Newton-based methods for restoration.

As a consequence of the theoretical facts mentioned above, the method has a good behavior in problems with moderate values of K and N, and is competitive with the popular DIIS algorithm, which depends on eigenvalue calculations, in typical electronic structure calculations. This encourages us to develop an implementation for large K, N, taking full advantage of the sparsity of iterates and gradients. The first steps were taken towards such implementation. We used a family of linear-objective problems, where the gradient was block-diagonal, with 17×17 blocks, $K \approx 2 \times 10^6$, $N = 4 \times 10^5$. The off-diagonal entries of the blocks were -1, N diagonal entries of the gradient matrix were equal to 2 and the remaining $K - N$ entries were equal to 20. We used a specific structure for the problem and we performed all the operations taking advantage of that structure, so that fill-in is not possible. The IR methods (both global and local) converged in about 4 iterations using less than 2 minutes of CPU time.

We guess that fast local convergence can also be exploited to improve convergence of other algorithms when close to the solution.

The application of IR is not restricted to the Hartree-Fock case, in which the objective function is quadratic. In fact, there is no restriction on the form of the objective function for using this approach. In general cases we cannot use the classical quadratic conjugate gradient algorithm in the tangent minimization phase, but we can use modern forms of the CG algorithm for non-quadratics, as the ADA algorithm of Hager and Zhang [40] or variations of the spectral projected gradient method [41–43]. Future research along these lines is expected.

Acknowledgements We are indebted to two anonymous referees whose comments and remarks helped a lot to improve the first version of the paper.

References

1. Cancès, E., Defranceschi, M., Kutzelnigg, W., Le Bris, C., Maday, Y.: Computational quantum chemistry: a primer. In: Le Bris, C., Ciarlet, P.G. (eds.) Handbook of Numerical Analysis, Special Volume, Computational Chemistry, vol. 10. North-Holland, Amsterdam (2003)
2. Cancès, E., Le Bris, C., Maday, Y.: Méthodes Mathematiques en chimie quantique. Une Introduction. Springer, Berlin (2006)

3. Hehre, W.J., Radom, L., Schleyer, P.V.R., Pople, J.A.: Ab Initio Molecular Orbital Theory. Wiley, New York (1986)
4. Helgaker, T., Jorgensen, P., Olsen, J.: Molecular Electronic-Structure Theory. Wiley, New York (2000)
5. Sánchez-Portal, D., Ordejón, P., Artacho, E., Soler, J.M.: Density-functional method for very large systems with LCAO basis sets. Int. J. Quantum Chem. **65**, 453–461 (1997)
6. Pulay, P.: Convergence acceleration of iterative sequences: the case of SCF iteration. Chem. Phys. Lett. **73**, 393–398 (1980)
7. Cancès, E., Le Bris, C.: Can we outperform the DIIS approach for electronic structure calculations? Int. J. Quantum Chem. **79**, 82–90 (2000)
8. Francisco, J.B., Martínez, J.M., Martínez, L.: Globally convergent trust-region methods for Self-Consistent Field electronic structure calculations. J. Chem. Phys. **121**, 10863–10878 (2004)
9. 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. **40**, 349–377 (2006)
10. 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. **121**, 16–27 (2004)
11. 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. **123**, 1–17 (2005)
12. Andreani, R., Castro, S.L.C., Chela, J., Friedlander, A., Santos, S.A.: An Inexact-Restoration method for nonlinear bilevel programming problems. Comput. Optim. Appl. **43**, 307–328 (2009)
13. Birgin, E.G., Martínez, J.M.: Local convergence of an Inexact-Restoration method and numerical experiments. J. Optim. Theory Appl. **127**, 229–247 (2005)
14. Fischer, A., Friedlander, A.: A new line search Inexact Restoration approach for nonlinear programming. Comput. Optim. Appl. (2009). doi:10.1007/s10589-009-9267-0
15. Gonzaga, C.C., Karas, E.W., Vanti, M.: A globally convergent filter method for nonlinear programming. SIAM J. Optim. **14**, 646–669 (2003)
16. Kaya, C.Y., Martínez, J.M.: Euler discretization and Inexact Restoration for Optimal Control. J. Optim. Theory Appl. **134**, 191–206 (2007)
17. Martínez, J.M.: Inexact Restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming. J. Optim. Theory Appl. **111**, 39–58 (2001)
18. Martínez, J.M., Pilotta, E.A.: Inexact Restoration algorithms for constrained optimization. J. Optim. Theory Appl. **104**, 135–163 (2000)
19. Fletcher, R.: Practical Methods of Optimization. Wiley, New York (1987)
20. Martínez, J.M., Svaiter, B.F.: A practical optimality condition without constraint qualifications for nonlinear programming. J. Optim. Theory Appl. **118**, 117–133 (2003)
21. Janin, R.: Direction derivative of the marginal function in nonlinear programming. Math. Program. Study **21**, 127–138 (1984)
22. Andreani, R., Martínez, J.M., Schuverdt, M.L.: On the relation between the Constant Positive Linear Dependence condition and quasinormality constraint qualification. J. Optim. Theory Appl. **125**, 473–485 (2005)
23. Qi, L., Wei, Z.: On the constant positive linear dependence condition and its application to SQP methods. SIAM J. Optim. **10**, 963–981 (2000)
24. Barrault, M., Cancès, E., Hager, W., Le Bris, C.: Multilevel domain decomposition for electronic structure calculations. J. Comput. Phys. **222**, 86–109 (2007)
25. Cancès, E., Le Bris, C., Lions, P.-L.: Molecular simulation and related topics: some open mathematical problems. Nonlinearity **21**, T165–T176 (2008)
26. Le Bris, C.: Computational chemistry from the perspective of numerical analysis. Acta Numer. **14**, 363–444 (2005)
27. Yang, C., Meza, J.C., Wang, L.-W.: A constrained optimization algorithm for total energy minimization in electronic structure calculations. J. Comput. Phys. **217**, 709–721 (2006)
28. Zhao, G.: Representing the space of linear programs as the Grassman manifold. Math. Program. **121**, 353–386 (2010)
29. Pino, R., Scuseria, G.E.: Purification of the first-order density matrix using steepest descent and Newton-Raphson methods. Chem. Phys. Lett. **360**, 117–122 (2002)
30. Mc Weeny, R.: Some recent advances in density matrix theory. Rev. Mod. Phys. **32**, 335–369 (1960)
31. Rubensson, E.H., Jensen, H.J.: Determination of the chemical potential and HOMO/LUMO orbitals in density purification methods. Chem. Phys. Lett. **432**, 591–594 (2006)

32. Rubensson, E.H., Rudberg, E., Salek, P.: Density matrix purification with rigorous error control. J. Comput. Chem. **26**, 1628–1637 (2008)
33. Palser, A., Manopoulos, D.: Canonical purification of the density matrix in electronic structure theory. Phys. Rev. B **58**, 12704–12711 (1998)
34. Dolan, E.E., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
35. Gomes-Ruggiero, M.A., Kozakevich, D.N., Martínez, J.M.: A numerical study on large-scale nonlinear solvers. Comput. Math. Appl. **32**, 1–13 (1996)
36. Dunning, T.H.: Gaussian basis sets for use in correlated molecular calculations. 1. The atoms Boron through Neon and Hydrogen. J. Chem. Phys. **90**, 1007–1023 (1989)
37. van Lenthe, E., Baerends, E.J.: Optimized Slater-type basis sets for the elements 1-118. J. Comput. Chem. **24**, 1142–1156 (2003)
38. Jensen, J.H., Koseki, S., Matsunaga, N., Nguyen, K.A., Su, S., Windus, T.L., Dupuis, M., Montgomery, J.A.: General atomic and molecular electronic-structure system. J. Comput. Chem. **14**, 1347–1363 (1993)
39. Goedecker, S.: Linear scaling electronic structure methods. Rev. Mod. Phys. **71**, 1085–1123 (1999)
40. Hager, W.W., Zhang, H.: Algorithm 851: CG-DESCENT, a conjugate gradient method with guaranteed descent. ACM Trans. Math. Softw. **32**, 113–137 (2006)
41. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10**, 1196–1211 (2000)
42. Birgin, E.G., Martínez, J.M., Raydan, M.: Algorithm 813: SPG-Software for convex-constrained optimization. ACM Trans. Math. Softw. **27**, 340–349 (2001)
43. Birgin, E.G., Martínez, J.M., Raydan, M.: Inexact Spectral Projected Gradient methods on convex sets. IMA J. Numer. Anal. **23**, 539–559 (2003)