On the Local and Global Convergence of a Reduced Quasi-NEWTON Method

J. Ch. GILBERT

International Institute for Applied Systems Analysis (IIASA), Laxenburg

Summary: In optimization in \( \mathbb{R}^n \) with \( m \) nonlinear equality constraints, we study the local convergence of reduced quasi-NEWTON methods, in which the updated matrix is of order \( n-m \). Furthermore, we give necessary and sufficient conditions for superlinear convergence (in one step) and we introduce a device to globalize the local algorithm. It consists in determining a step along an arc in order to decrease an exact penalty function and we give conditions so that asymptotically the step-size will be equal to one.

AMS 1980 Subject Classification:
Primary: 49 D 05; Secondary: 49 D 30, 65 R 05

Key words: Constrained optimization, successive quadratic programming, reduced quasi-NEWTON method, superlinear convergence, exact penalty function, arc search, step-size selection procedure, global convergence.

1. Introduction

Let \( \omega \) be an open convex set in \( \mathbb{R}^n \), \( f \) be a real-valued function on \( \omega \) and \( c \) map \( \omega \) to \( \mathbb{R}^m \), where \( m < n \). We shall suppose that \( f \) and \( c \) are functions of class \( C^3 \) with \( n \geq 3 \), i.e., \( f \) and \( c \) are supposed three times continuously differentiable with bounded derivatives on \( \omega \). We shall endow \( \mathbb{R}^n \) with its canonical basis and with the Euclidean scalar product. We are interested in algorithms for solving the following minimization problem with equality constraints:

\[
\min \{ f(x) \mid x \in \omega, \ c(x) = 0 \}.
\]  

(1.1)

In addition to the smoothness of \( f \) and \( c \), we shall assume that \( c \) is a submersion on \( \omega \), that is to say that the \( m \times n \) Jacobian matrix

\[
A_c := A(x)
\]  

(1.2)

of partial derivatives of \( c \) at \( x \) is supposed surjective for all \( x \) in \( \omega \). If \( \omega \) is "large" this is a very strong hypothesis, but it is usual to suppose that the gradients of the constraints are linearly independent at a solution of (1.1) and, therefore, this hypothesis is satisfied in a neighbourhood of a solution. Then, if \( x^* \) is a local minimizer for problem (1.1), there exists a unique LAGRANGE multiplier \( \lambda^* \) so that the

\[^1\] Work supported in part by INRIA (Institut National de Recherche en Informatique et en Automatique), France and in part by the FNRS (Fonds National de la Recherche Scientifique), Belgium.

"Tous droits de propriété intellectuelle réservés. Reproduction, représentation intégrale ou partielle sans autorisation (code de la propriété intellectuelle)"
first order optimality conditions are satisfied at \( x=x^*_a \) and \( \lambda = \lambda^*_a \):

\[
\begin{align*}
\nabla f(x) &= 0, \\
\nabla^T c(x) + A^*_a \lambda &= 0,
\end{align*}
\]

(1.3)

where \( \nabla f(x) \) is the vector of partial derivatives of \( f \) at \( x \). The quantity on the left hand side of the second equation is the first derivative with respect to \( z \) of the Lagrangian \( l(z, \lambda) := f(x) + c(x)^T \lambda \). The second order sufficient condition will also be assumed: the \( n \times n \) Hessian matrix \( L_a \) of second derivatives with respect to \( x \) of \( l \) at \( (x^*_a, \lambda^*_a) \) is supposed positive definite in the null space \( N(A^*_a) \) of \( A^*_a := A(x^*_a) \).

For further references, we gather these hypotheses under the name of

**Assumption A:**

- \( f, c \) are in \( C^2_b(\Omega) \) with \( \nu \geq 3 \),
- \( c \) is a submersion,
- \( (x^*_a, \lambda^*_a) \) satisfies (1.3) at \( x = x^*_a \) and \( \lambda = \lambda^*_a \),
- \( h^T L_a h > 0 \) for all \( h \) in \( \mathbb{R}^n \) with \( h \neq 0 \) and \( A^*_a h = 0 \).

Quasi-Newton methods, also called variable metric or secant methods, are methods for solving a system of nonlinear equations on \( \mathbb{R}^n \), say \( F(x^*_a) = 0 \). They generate a sequence of points \((x_k)\) and a sequence of nonsingular matrices \((J_k)\) of order \( N \) from the data of a point \( x_0 \) and a nonsingular matrix \( J_1 \) by the formula:

\[
x_{k+1} = x_k - J_k^{-1} F(x_k),
\]

where \( J_k \) is updated at each iteration according to the following scheme:

\[
J_{k+1} = U(J_k, \gamma_k, \sigma_k),
\]

\[
\gamma_k := F(x_{k+1}) - F(x_k),
\]

\[
\sigma_k := x_{k+1} - x_k.
\]

The rule \( U \) is designed in order that \( J_{k+1} \) will satisfy the secant equation \( J_{k+1} F(x_k) = -\gamma_k \) and then will improve the approximation by \( J_k \) of the Jacobian matrix \( \nabla F(x_k) \) at the solution \( x^*_a \). These methods are particularly attractive because second order derivatives need not be calculated and because a superlinear rate of convergence for \( (x_k) \) can be obtained (see [16]), i.e.

\[
\frac{\|x_{k+1} - x^*_a\|}{\|x_k - x^*_a\|} \rightarrow 0 \quad \text{as} \quad k \rightarrow \infty.
\]

Such a method could be used for solving system (1.3), but matrices of order \( n + m \) should be updated. The aim of this paper is to introduce and study quasi-Newton methods that require the update of matrices of order \( n + m \) only.

The *Successive Quadratic Programming* (SQP) method proposed by Wilson [38] and Han [28] improves the method described above with regard to the order of the updated matrices. In this method \( x_{k+1} \) is obtained from \( x_k \) by solving a quadratic programme with linear constraints:

\[
\begin{align*}
\min f'(x_k) \cdot (x - x_k) + \frac{1}{2} (x - x_k)^T L_k (x - x_k), \\
x \in \mathbb{R}^n \quad \text{and} \quad c(x_k) + c'(x_k) \cdot (x - x_k) = 0,
\end{align*}
\]

(1.5)
where we have used a dot to separate the linear operators \( f'(x_k) \) and \( c'(x_k) \) from their argument \((x-x_k)\). The matrix \( L_k \) of order \( n \) is updated in order to approximate \( L_n \), the Hessian of the Lagrangian. Under assumption \( A \) and the non singularity of \( L_k \) on \( N(A_k) \), the solution of \((1.8)\) can be written in the form (see \([20]\)):

\[
x_{k+1} = x_k - A_k^{-1} c(x_k) - Z_k(Z_k^T L_k Z_k)^{-1} g(x_k),
\]

where \( A_k^{-1} \) is any right inverse of \( A_k := A(x_k) \), \( Z_k \) is any \( n \times (n-m) \) matrix whose columns form a basis of the tangent space \( N(A_k) \) to the manifold \( M_k := c^{-1}(c(x_k)) \) at \( x_k \) and \( g(x_k) \) is the reduced gradient defined at \( x \in \omega \) by

\[
g(x) := Z_k^T f(x) \in \mathbb{R}^{n-m}.
\]

The first part \((-A_k^{-1}c(x_k))\) of the displacement in \((1.8)\) is a restoration step, i.e., a Newton-like step for solving \( c(x_k) = 0 \). It belongs to \( R(A_k) \) which is a complementary space to \( N(A_k) \) in \( \mathbb{R}^n \). The second part of the displacement in \((1.6)\) is a minimization step belonging to the tangent space \( R(Z_k) = N(A_k) \) to \( M_k \) at \( x_k \).

Let us consider for a while the case where the constraints are linear:

\[
c(x) := Ax + b = 0,
\]

where \( A \) is an \( m \times n \) surjective matrix and \( b \) is a vector in \( \mathbb{R}^m \). As in the nonlinear case, let us introduce \( Z^* \), an \( n \times (n-m) \) injective matrix whose columns generate \( N(A) : AZ^* = 0 \). Suppose that the first iterate belongs to the plane of the constraints: \( c(x_1) = 0 \). Then, any point \( x \) satisfying the constraints \((1.8)\) can be expressed by using a reduced variable \( u \) in \( \mathbb{R}^{n-m} \) \((x = x_1 + Z^* u)\) and the problem of minimizing \( f \) subject to the constraints \((1.8)\) is equivalent to that of minimizing \( \Phi(u) := f(x_1 + Z^* u) \) on \( \{u \in \mathbb{R}^{n-m} \mid x_1 + Z^* u \in \omega \} \):

\[
\min \{ \Phi(u) \mid u \in \mathbb{R}^{n-m}, x_1 + Z^* u \in \omega \}.
\]

By considering the optimality equation \( \nabla \Phi(u_k) = Z^T \nabla f(x_1 + Z^* u_k) = 0 \), a quasi-Newton method for solving problem \((1.9)\) generates a sequence \( \{u_k\} \) in \( \mathbb{R}^{n-m} \) and a sequence \( \{G_k\} \) of nonsingular matrices of order \( n-m \). On the one hand, we have

\[
\begin{align*}
  u_{k+1} &= u_k - G_k^{-1} \nabla \Phi(u_k), \\
  x_{k+1} &= x_1 + Z^* u_{k+1}.
\end{align*}
\]

By setting \( \gamma_k := x_{k+1} - x_k = Z^* u_{k+1} \), we obtain

\[
\begin{align*}
  x_{k+1} &= x_k - Z^* \gamma_k,
\end{align*}
\]

where \( g(x) := Z^T \nabla f(x) \) is the reduced gradient of \( f \) at \( x \). On the other hand, \( G_k \) is updated as follows:

\[
\begin{align*}
  G_{k+1} &= \frac{1}{\lambda_k} U(G_k, \gamma_k, \alpha_k), \\
  \gamma_k &= g(x_{k+1}) - g(x_k), \\
  Z_{\gamma_k} &= x_{k+1} - x_k,
\end{align*}
\]

in order to approximate \( \nabla^2 \Phi(u_k) \), the Hessian of \( \Phi \) at \( u_k \), which is also the reduced Hessian of \( f \) at \( x_k \):

\[
Z^T \nabla^2 f(x_k) Z = \nabla^2 \Phi(u_k).
\]
The conditions so that the sequence \((u_k)\) will converge superlinearly can be satisfied and because of the injectivity of \(Z^{-}\), the same conditions will assure the superlinear convergence of the sequence \((x_k)\).

Algorithm (1.10)–(1.11) is a reduced quasi-Newton method because the order of the updated matrices is \(n - m\) rather than \(n\) the SQP method. Our aim is to study such methods in the case of nonlinear constraints. They are particularly well adapted to problems with \(n\) large and \(n - m\) small. That situation appears for example in the parametric identification of nonlinear sources in elliptic partial differential equations. If a finite element method is used to discretize the state equations (the constraints), \(m\) is large, say of the order of 1000, whereas the number \(n - m\) of identifiable parameters is usually small: 2 or 3 in the example given by Blum, Grünert and Tyron (3). In that case, a reduced quasi-Newton method is usable while the SQP method is not because of the order of the matrices that should be updated. Another advantage in developing reduced quasi-Newton methods comes from the fact that, under Assumption A, the projected Hessian of the Lagrangian is positive definite at the optimum. Therefore, positive definite quasi-Newton approximations of that operator can be generated, in particular by the BFGS formula, which is at present widely believed to be the best update formula. Thus, we see that reduced quasi-Newton methods appear rather natural. So, it is important to generalize algorithm (1.10)–(1.11) in case the constraints are nonlinear.

This can be done by using the implicit function theorem in order to obtain a reduced objective function:
\[
\Phi(u) = f(\xi(u)),
\]
where \(\xi: \mathbb{V} \subset \mathbb{R}^{n-m} \rightarrow \mathbb{R}^n\) is a parametric representation of the regular manifold \(c^{-1}(0)\) around \(x_\alpha := \xi(u_\alpha)\). We have \(c(\xi(u)) = 0\) for all \(u\) in the neighborhood \(\mathbb{V}\) of \(u_\alpha\). This is the basic idea of methods like the Generalized Reduced Gradient (GRG) method (Abadie and Carpentier [1]). In fact, the parametric representation \(\xi(u)\) is usually not known and this leads to several difficulties. Because the method asks the generated sequence \((x_k)\) to be feasible \((c(x_k) = 0\) for all \(k)\), and because this cannot be achieved exactly in practice, some criterion has to be introduced to decide when to stop the restoration steps, i.e., how well the equality \(x_{k+1} = \xi(u_{k+1})\) has to be realized (Mukai and Polak [33]). Another difficulty appears when \(x_k\) is far from \(x_\alpha\) and a step-size has to be introduced in the \(u\)-space in order to globalize the method. Indeed, every time a step-size is tried, an infinite number of restoration steps have to be done: see [18], [21], [33], [19].

On the other hand, some non-feasible reduced quasi-Newton methods have been developed recently. Garay [20] has studied the following algorithm:
\[
x_{k+1} = x_k + r_k^1 + t_k^1,
\]  \(1.12\)
\[
r_k^1 := -A_k \xi(x_k),
\]  \(1.13\)
\[
t_k^1 := -Z_k G_k^{-1} g(x_k),
\]  \(1.14\)
where $A_k^c$ is any right inverse of $A_k$, $Z_k$ is any $n \times (n-m)$ matrix whose columns form a basis of the tangent space $N(A_k)$, $G_k$ is a nonsingular matrix of order $(n-m)$ and $g(x_k)$ is the reduced gradient of $f$ at $x_k$. The tangent step $t_k$ in (1.12), tangent to the manifold $M_k$, has the same structure as the displacement in (1.10) except for the basis $Z_k$, which changes here at each iteration. The restoration step $r_k$ in (1.12) is introduced to improve the feasibility of the sequence. The displacement in (1.12) can also be deduced from the displacement (1.6) of the SQP method by dropping the last part of the minimization step and by considering $G_k$ as an approximation of the projected Hessian $Z_k^T L(x_k, \lambda_k) Z_k$.

For their part, COLEMAN and CONN [11] have studied the following algorithm:

\begin{align*}
  x_{k+1} &= x_k + r_k + t_k, \quad \text{(1.15)} \\
  r_k^2 &= - A_k^c (x_k + t_k), \quad \text{(1.16)} \\
  t_k^2 &= Z_k G_k^{-1} g(x_k), \quad \text{(1.17)}
\end{align*}

where $Z_k$ is an $n \times (n-m)$ matrix whose columns form an orthogonal basis of $N(A_k)$ and $A_k^c$ is the Penrose pseudo-inverse of $A_k$: $A_k^c = A_k^T (A_k A_k^T)^{-1}$. The relevant difference to algorithm (1.12)–(1.14) lies in the restoration step. Here, the constraints are evaluated at $x_k + t_k^2$, after the tangent step, rather than at $x_k$ in algorithm (1.12)–(1.14).

The study of algorithms (1.12)–(1.14) and (1.15)–(1.17) showed that when the matrices $G_k$ are suitably updated and the initial point $x_0$ is close to $x_*$, the sequence $(x_k)$ generated by any of those algorithms converges to $x_*$ superlinearly in two steps, that is to say:

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \rightarrow 0 \quad \text{as} \quad k \rightarrow \infty.$$ 

This rate of convergence is not so good as the rate (1.4) obtained with reduced quasi-Newton methods when the constraints are linear or with the SQP method. On the other hand, counter-examples have been given by BYRD [8] and YUAN [39] for which both the methods of GABAY and COLEMAN and CONN do not converge better than with a two step superlinear rate of convergence. Therefore, the question of the rate of convergence of reduced quasi-Newton methods seemed closed.

However, BYRD [7] and GILBERT [22] have shown (independently) that the sequence $(x_k + t_k^2)$ of COLEMAN and CONN’s algorithm converges superlinearly in one step. A similar result has also been obtained by HOYER [30] who considers algorithm (1.15)–(1.17) when $G_k$ is the exact reduced Hessian of the Lagrangian: $G_k = Z_k^T L(x_k, \lambda_k) Z_k$ and $\lambda_k$ is a LAGRANGE multiplier estimate. This fact makes this method competitive. Indeed, the quasi-Newton version of COLEMAN and CONN’s algorithm (see [13]) needs two linearizations of the constraints per iteration. Consequently, as mentioned by BYRD [7], it was thought that a superlinear step was made in this algorithm for every four constraint linearizations. The convergence result of BYRD [7], HOYER [30] and GILBERT [22] shows however that one superlinear step is made for every two constraint

"Tous droits de propriété intellectuelle réservés. Reproduction, représentation interdites sans autorisation (cette de la propriété intellectuelle)."
linearizations. Furthermore, it can be shown (see [25]) that the use of an update criterion allows linearization of the constraints only once per iteration. Therefore, the method makes a superlinear step for each constraint linearization.

In this paper, we first show how algorithm (1.15)–(1.17) can be obtained from a very general principle. If unconstrained optimization problems are related to the solution of nonlinear equations (via the optimality condition \( f'(x^\circ) = 0 \)), we see from (1.3) that constrained optimization problems are closely related to the solution of two coupled nonlinear equations:

\[
c(x^\circ) = 0 \quad \text{in } \mathbb{R}^m, \tag{1.18}
\]

\[
g(x^\circ) = 0 \quad \text{in } \mathbb{R}^{n-m}. \tag{1.19}
\]

The equation (1.19) expresses the vanishment of the reduced gradient defined in (1.7) and is obtained by projecting the second equation of (1.3) on the tangent space \( N(A_x) \). A “decoupling” method for solving (1.18)–(1.19) is introduced in Section 2. At \( x_k \), the first part of the step of the method consists in doing a Newton-like displacement for solving (1.18). This leads to a point \( y_k \). Then, \( x_{k+1} \) is obtained by doing a Newton-like displacement for solving (1.19) from the point \( y_k \):

\[
y_k = x_k - A^{-\circ}_x c(x_k), \tag{1.20}
\]

\[
x_{k+1} = y_k - B^{-\circ}_x g(y_k). \tag{1.21}
\]

In (1.20), \( A^{-\circ}_x \) is a right inverse of \( \nabla c(x_k) \) and in (1.21), \( B^{-\circ}_x \) is a right inverse of \( \nabla g(x_k) \). We shall show that only conditions on \( B^{-\circ}_x \) have to be imposed in order to ensure the local quadratic convergence (in one step) of the process. In Section 4, we apply this algorithm to constrained optimization, when \( g \) in (1.19) has the special structure (1.7), and it takes the form of a reduced method. Its extension to reduced quasi-Newton method is then easily done. This presentation gives, in our opinion, some insight into method (1.15)–(1.17) and shows what the degrees of freedom are in the choice of the operators \( A^{-\circ}_x \) in (1.16) and \( Z^{-\circ}_x \) in (1.17). In fact, \( A^{-\circ}_x \) may be any right inverse of \( A_x \) (with a smoothness hypothesis of \( A^{-\circ}_x = A(x_k) \)) according to \( x_k \) and not necessarily the Penrose pseudo-inverse and the columns of \( Z^{-\circ}_x \) may form any basis of \( N(A_x) \) and not necessarily an orthogonal basis. This remark may be crucial in some applications like the one mentioned above where the “partitioned framework” (see Section 3) occurs naturally. In Section 5, we give a necessary and sufficient condition of superlinear convergence of the method that is weaker than the sufficient condition given by Byrd [7].

The globalization of the local method could then be done as in the paper by Coleman and Conn [12]. In Section 6, however, we examine another globalizing technique essentially based on the ideas of Han [29] for the SQP method (see also [14]). We introduce the following exact penalty function:

\[
\Theta_p(x) = f(x) + p \| c(x) \|_A, \tag{1.22}
\]
where \( p \) is a large enough penalty parameter and \(|\cdot|_1\) is the \( l_1 \) norm on \( \mathbb{R}^n \). We look for \( x_\omega \) by minimizing \( \Theta_\omega \) on \( \omega \). The idea is then to obtain a descent direction for \( \Theta_\omega \) at the current iterate from the displacements calculated by the local algorithm (1.20)–(1.21). Contrary to what happens with the SQP method, our total displacement is not necessarily a descent direction for \( \Theta_\omega \) any more. So, we shall introduce a descent arc, being inspired in this way by the work of Garay [20] for algorithm (1.12)–(1.14) and Mayne and Polak [32] for the SQP method, although in those algorithms, the arc was introduced for other reasons. A search on the arc is done in order to decrease the penalty function \( \Theta_\omega \) with the help of an Armijo-like criterion. This gives a theorem ensuring the global convergence of the method. Furthermore, under natural conditions, the "Maratos effect" is avoided: the step-size is equal to one after a finite number of iterations. Therefore there is a smooth transition from the global to the local method that does not prevent the superlinear convergence from occurring.

If \((v_k)\) is a sequence in a normed space \((E, \| \cdot \|_E)\) and \((\lambda_k)\) is a sequence of positive numbers, we shall say that \((v_k)\) is a big \( O \) of \((\lambda_k)\) (we shall note \( v_k = O(\lambda_k) \)) if the sequence \(|v_k|/\lambda_k\) is bounded and we shall say that \((v_k)\) is a small \( o \) of \((\lambda_k)\) (we shall note \( v_k = o(\lambda_k) \)) if the sequence \(|v_k|/\lambda_k\) converges to zero. We shall say that two positive real sequences \((v_k)\) and \((\beta_k)\) are equivalent (we shall note \( v_k \sim \beta_k \)) if \( v_k = O(\beta_k) \) and \( \beta_k = O(v_k) \). We shall note \( \nu^i \), the \( i \)-th component of a vector \( \nu \) in \( E \). If \( A \) is a linear operator from \((E, \| \cdot \|_E)\) to \((F, \| \cdot \|_F)\), we shall note \( \|A\| := \sup \{\|Av\|_F \mid \|v\|_E = 1\} \). If \( A \) and \( B \) are two square matrices of the same order, we shall note \( A \preceq B \) when \( B - A \) is positive semi definite.

This paper constitutes a revised version of a part of the INRIA report number RR-482 in which some techniques for updating the reduced matrix have also been investigated (see also [25]). A variant of the method is given in [23].

2. A Decoupling Method for Solving Two Nonlinear Coupled Equations

Let us consider the following coupled system of nonlinear equations:

\[
\begin{align*}
F(x) &= 0, \\
G(x) &= 0, \\
\end{align*}
\]

(2.1)

where \( F \) and \( G \) are supposed smooth and map \( \mathbb{R}^n \) to \( \mathbb{R}^m \) and \( \mathbb{R}^{n-m} \) \((m < n)\) respectively. Let \( x_\omega \) be a solution of (2.1) and let us denote by \( A_\omega \) the \( m \times n \) Jacobian matrix of \( F \) at \( x_\omega \) and \( B_\omega \) the \( (n-m) \times n \) Jacobian matrix of \( G \) at \( x_\omega \). We shall say that \( x_\omega \) is a regular solution of (2.1) if the Jacobian matrix of the system (2.1),

\[
J_\omega := \begin{bmatrix}
A_\omega \\
B_\omega
\end{bmatrix}
\]

(2.2)

is nonsingular. This will be the case if and only if \( N(A_\omega) \cap N(B_\omega) = \{0\} \). We would like to define a Newton-like method for solving (2.1) without having to inverse...
the Jacobian $J(x_k)$ of (2.1) at $x_k$. For that, let us suppose that $x_k$ is a regular solution. Then $A^{-}$ and $B_+$ are surjective and we can introduce a right inverse $A^{-}_k$ of $A^{-}$ and a right inverse $B_+^-$ of $B_+$:

$$A^{-}_k A^{-}_k = I_m, \quad B_+^- B_+^- = I_{n-m}.$$  \hfill (2.3)

Two algorithms using $A^{-}_k$ and $B_+^-$ can be considered and we introduce them with the help of fixed point maps $\xi_1$ and $\xi_2$. The first one is

$$x_{k+1} = \xi_1(x_k), \quad \xi_1(x) := x - A^{-}_k F(x) - B_+^- O(x),$$  \hfill (2.4)

and the second one is

$$x_{k+1} = \xi_2(x_k) := (\Psi \circ \Phi)(x_k),$$  \hfill (2.5)

$$\Phi(x) := x - A^{-}_k F(x),$$  \hfill (2.6)

$$\Psi(y) := y - B_+^- O(y).$$  \hfill (2.7)

These algorithms are somewhat "ideal". Indeed, the matrices $A^{-}_k$ and $B_+^-$ are not known and so neither are $A^{-}_k$ and $B_+^-$. But they are simpler to study than their implementable versions that will be introduced later.

The relations (2.3) do not determine the right inverses $A^{-}_k$ and $B_+^-$ completely. Therefore, we may try to choose them so that the sequences generated by algorithms (2.4)–(2.5) and (2.6)–(2.8) will have a good local behaviour. The next two propositions show that this is possible: we can get conditions on $A^{-}_k$ and $B_+^-$ in order to have $\xi_1(x_k) = 0$ and $\xi_2(x_k) = 0$, which will ensure a quadratic rate of convergence for both algorithms. We shall say that an $n$-row matrix is a basis of a given subspace of $\mathbb{R}^n$ if it is injective and if its columns form a basis of that subspace.

**Proposition 2.1:** Suppose that $F$ and $G$ are differentiable at $x_0$, a regular solution of (2.1). Then, the following statements are equivalent:

(i) $\xi_1(x_k) = 0$,

(ii) $R(A^{-}_k) = N(B_+^k)$ and $R(B_+^-) = N(A^{-}_k)$,

(iii) for any right inverse $A^{-}_k$ of $A^{-}$ and any basis $Z_-$ of $N(A^{-}_k)$, we have

$$A^{-}_k = (I - Z_-^- B_+^- Z_-^-)^{-1} B_+^- A^{-}_k, \quad B_+^- = Z_-(B_+^k Z_+^k)^{-1}.$$  \hfill (2.9)

**Proof:** First, we prove (i)⇒(ii). Statement (i) is equivalent to

$$I = A_+^- A_+^- + B_+^- B_+^-.$$  \hfill (2.10)

The right hand side of (2.11) is equal to $[A_+^- B_+^-] J_n$. Then, (2.11) means that $[A_+^- B_+^-]$ is the inverse of $J_n$ and therefore is equivalent to $I = J_n [A_+^- B_+^-]$, i.e. $A_+^- B_+^- = 0$ and $B_+^- A_+^- = 0$, which is equivalent to statement (ii) because the matrices $A_+^-$, $B_+^-$, $A_+^-$ and $B_+^-$ have full rank. Next, we prove (ii)⇒(iii). Let $Z_-$ be any basis of $N(A^{-}_k)$; $R(B_+^-) = N(A^{-}_k)$ and $Z_-$ is injective. Because $J_n$ is nonsingular, $B_+^- Z_-^-$ is nonsingular. Indeed, if $u$ in $\mathbb{R}^{n-m}$ satisfies $B_+^- Z_-^u = 0$, we have $J_n Z_-^u = 0$ (because $A_+^- Z_-^u = 0$), then $Z_-^u = 0$ because $J_n$ is nonsingular and $u = 0$ because $Z_-^u$
is injective. Then, by multiplying (2.11) to the right by $Z_a$, we get $Z_a = B_g(B_a Z_a)$ and therefore (2.10). Formula (2.9) is obtained by multiplying (2.11) to the right by any right inverse $A_g^{-1}$ of $A_g$ and by using (2.10). It remains to prove (iii)⇒(ii).

If we take $A_g^{-1} = A_g^\top$ in (2.9), we obtain $B_g A_g^\top = 0$ and so $R(A_g^\top) = N(B_g)$ because these spaces have the same dimension $m$. Because $A_g Z_a = 0$, (2.10) gives $A_g B_g = 0$ and so $R(Z_a) = N(A_g)$ because these spaces have the same dimension $n - m$.

Proposition 2.2: Suppose that $F$ and $G$ are differentiable at $x_a$, a regular solution of (2.1). Then, the following statements are equivalent:

(i) $\xi_2(x_a) = 0$,

(ii) $R(B_g) = N(A_g)$,

(iii) for any basis $Z_a$ of $N(A_g)$, we have $B_g = Z_a (B_g Z_a)^{-1}$.

Proof: The equivalence (i)⇔(ii) comes from $\xi_2(x_a) = (I - B_g B_g) (I - A_g^{-1} A_g)$ and that the spaces $N(A_g) = R(I - A_g^{-1} A_g)$ and $R(B_g) = N(I - B_g B_g)$ have the same dimension $n - m$. To prove that (i) implies (iii), let $Z_a$ be any basis of $N(A_g)$.

By multiplying to the right both sides of

$$(I - B_g B_g) (I - A_g^{-1} A_g) = 0$$

by $Z_a$, we get $Z_a = B_g (B_g Z_a)$ and therefore (iii) because $B_g Z_a$ is nonsingular (see the proof of Proposition 2.1). From (iii), we get $A_g B_g = 0$ by multiplying to the left $B_g = Z_a (B_g Z_a)^{-1}$ by $A_g$ and $A_g B_g = 0$ is equivalent to (ii) because $R(B_g)$ and $N(A_g)$ have the same dimension $n - m$.

In statement (iii) of Proposition 2.1, we could equivalently have given to $B_g$ the role of $A_g$. We also see that the right inverses $A_g^{-1}$ and $B_g$ are completely determined by condition (i) of Proposition 2.1 and do not depend on the choice of $A_g^{-1}$ and $Z_a$ in (iii). Similarly, the right inverse $B_g$ is completely determined by condition (i) of Proposition 2.2 and does not depend on the choice of $Z_a$ in (iii).

From Proposition 2.1, we see that $\xi_2(x_a) = 0$ if and only if $[A_g^{-1} B_g]$ is the inverse of $J_{x_a}$. This means that algorithm (2.4)–(2.5) is in fact the “ideal” (with $J_{x_a}^{-1}$ rather than $J(x_a)^{-1}$) Newton method for solving (2.1) (see the displacement in (2.5)), the method we wanted to avoid. On the other hand, Proposition 2.2 shows that algorithm (2.6)–(2.8) needs fewer conditions to have a good local behaviour than algorithm (2.4)–(2.5). The fact that no conditions are required on the right inverse $A_g^{-1}$ means that any solver of the first equation in (2.1) can be used in (2.7), independently of the second equation of (2.1), whereas this is not true for the solver $B_g$ of the second equation of (2.1) that has to be adapted to the first equation.

The results of Propositions 2.1 and 2.2 have a geometrical interpretation. In the ideal Newton method, $(x_b)$ will converge quadratically if the displacements $(-A_g^{-1} F(x_a))$ and $(-B_g G(x_a))$ belong to the tangent space at $x_a$ to the manifolds defined respectively by the pre-image of 0 by $F$ and $G$. In method (2.6)–(2.8) only the second step $(-B_g G(y_b))$ has to belong to the tangent space $N(A_g)$; the first step is arbitrary (apart from the fact that $A_g^{-1}$ has to be a right inverse of $A_g$).
3. A Change of Coordinates

Before applying the results of the previous section to constrained optimization, let us give some examples of right inverses $A_x^-$ of $A_x$ and basis $Z_a$ of $N(A_x)$ that are frequently used in practice. The formalism adopted here has been introduced by Garay [19].

Once the injective matrices $A_x^-$ and $Z_a$ have been chosen, the columns of $[A_x^- Z_a]$ form a new basis of $\mathbb{R}^n$. Indeed, $R(A_x^-)$ is a complementary space of $N(A_x) = R(Z_a)$. To make a change of coordinates in that new basis, it is convenient to introduce the additional $(n - m) \times n$ matrix $Z_a$ given by the following proposition.

**Proposition 3.1:** Let $A_x$ be an $m \times n$ ($m < n$) surjective matrix, $A_x^-$ be any right inverse of $A_x$ and $Z_a$ be any basis of $N(A_x)$. Then, there exists a unique $(n - m) \times n$ matrix $Z_a$ such that

\[ Z_a A_x^- = 0, \]

\[ Z_a Z_a^T = I_{n-m}. \]

Furthermore, we have

\[ I_n = A_x^- A_x + Z_a Z_a^T. \]

**Proof:** Existence and unicity of the matrix $Z_a$ come from the nonsingularity of $[A_x^- Z_a]$ and (3.3) comes from the fact that $[A_x^- Z_a]^T$ is the inverse of $[A_x^- Z_a]$. ■

The relation (3.1) shows that $N(Z_a) = R(A_x^-)$ (the matrices $A_x^-$ and $Z_a$ have full rank) and (3.2) shows that $Z_a$ is a right inverse of $Z_a$. The equality (3.3) can be used to introduce a change of coordinates. Indeed, by applying it to a vector $\xi$ of $\mathbb{R}^n$, we see that $A_x \xi$ are the coordinates of $\xi$ in $R(A_x^-) = N(Z_a)$ and $Z_a \xi$ are the coordinates of $\xi$ in $R(Z_a) = N(A_x)$.

A first choice of matrices $A_x^-$ and $Z_a$, which is frequently made in constrained optimization, defines what could be called the orthogonal framework: $A_x^-$ is the Moore-Penrose pseudo-inverse of $A_x$ (see [2]) and $Z_a$ is an orthogonal basis of $N(A_x)$ for the Euclidean scalar product. We have:

\[ A_x^- = A_x^T \{ A_x A_x^T \}^{-1}, \]

\[ Z_a^T Z_a = I_{n-m}. \]

Then $Z_a = Z_a^T$ is the unique matrix satisfying (3.1) and (3.2). We see that $R(A_x^-)$ is orthogonal to $N(A_x)$. This framework has been adopted by Coleman and Conn [11-13] and by Byrne [7].

Another choice of matrices $A_x^-$ and $Z_a$ can be made when a separation of variables occurs naturally, as in optimal control problems or in parameter identification problems. This is also the natural framework to introduce the GRG method (Anaid and Carpenter [1]). It could be called the partitioned framework: $A_x$ is supposed to be partitioned in two submatrices

\[ A_x = [C_x D_x], \]
where the $m \times m$ matrix $C_x$ is nonsingular and $D_x$ has dimension $m \times (n-m)$. The right inverse $A_x^-$ is then taken as
\begin{equation}
A_x^- = \begin{bmatrix} C_x^{-1} \\ 0 \end{bmatrix}
\end{equation}
and the basis of $\mathcal{N}(A_x^-)$ is
\begin{equation}
Z_x^- = \begin{bmatrix} -C_x^{-1}D_x \\ I_{n-m} \end{bmatrix}.
\end{equation}
Then $Z_x = [0 \ I_{n-m}]$ is the unique matrix satisfying (3.1) and (3.2). This framework has been adopted by Hoyer [30].

In the following, we shall suppose that the choice of $(A_x^-, Z_x)$ is a smooth function of $x$:

**Assumption B:**
The function $x \rightarrow (A_x^-, Z_x)$ is bounded on $\omega$ and is in $C^{\eta-1} \omega$ with $\eta \geq 3$.

This assumption is satisfied for $A_x^-$ given by (3.4). With regard to $Z_x^-$, the question is more delicate, although the assumption can be satisfied locally by projection on $\mathcal{N}(A_x^-)$ of a basis $Z_x^-$ of $\mathcal{N}(A_x^-)$ (see [26], [9]).

### 4. A Reduced Quasi-Newton Method for Constrained Optimization

In this section, we apply the results of Section 2 to constrained optimization. The first step consists in reducing the size of the optimality system (1.3). This can be done because the second optimality condition can be expressed by $n-m$ equations rather than $n$, in fact, by the vanishing of the $n-m$ coordinates of the orthogonal projection of $\nabla f(x_\omega)$ on $\mathcal{N}(A_x^-)$. If $Z_x^-$ is any basis of $\mathcal{N}(A_x^-)$, the orthogonal projector on $\mathcal{N}(A_x^-)$ is $Z_x^- (Z_x^- Z_x^-)^{-1} Z_x^-T$. Then, the second equation of (1.3) is projected on $\mathcal{N}(A_x^-)$ by multiplying it by $Z_x^-T$. Using the definition (1.7) of the reduced gradient, the system (1.3) can be rewritten as follows:

\begin{align}
\left\{ \begin{array}{l}
  c(x_\omega) = 0, \\
  g(x_\omega) = 0.
\end{array} \right.
\end{align}

(4.1)

In order to apply the previous results, we need to calculate the first derivative of $g$ at $x_\omega$. This can be done as in [37]:
\begin{equation}
\nabla g(x_\omega) = \nabla (Z_x^-T (\nabla f(x) + A_x^T \lambda_a)) (x_\omega) = Z_x^T L_\omega.
\end{equation}

(4.2)

The Jacobian matrix of (4.1),
\[
\begin{bmatrix}
A_x^- \\
Z_x^- L_\omega
\end{bmatrix},
\]

is nonsingular because of the surjectivity of $A_x^-$ and the second order sufficient condition which is equivalent to the positive definiteness of
\begin{equation}
G_x := Z_x^- L_\omega Z_x^-.
\end{equation}

(4.3)

We shall note $H_x := G_x^{-1}$. 

*"Tous droits de propriété intellectuelle réservés, reproduction, représentation intégrale ou partielle sans autorisation (code de la propriété intellectuelle)."*
Now, let us apply algorithm (2.4)–(2.5) to the system (4.1). Using statement (iii) of Proposition 2.1 and (4.2), we get the following quadratically convergent algorithm:

\[ x_{k+1} = x_k - (I - Z_{e}^T H_{e} Z_{e}^{-1} I_b) A_e^T c(x_k) - Z_{e}^T H_{e} g(x_k), \]  

where \( A_e \) is any right inverse of \( A_e \) (playing the role of \( A_e^- \) in (2.9)) and \( Z_{e} \) is any basis of \( N(A_e) \). This is exactly the "ideal" SQP method: see (1.6). See also Goodman [27] for a related result. We shall not go further with this method.

If we apply algorithm (2.6)–(2.8) to the system (4.1), we get, using statement (iii) of Proposition 2.2 and (4.2):

\[ y_k = x_k - A_e c(x_k), \]  
\[ z_{k+1} = y_k - Z_{e}^T G_{e}^{-1} g(y_k), \]

where \( A_e^- \) is any right inverse of \( A_e \) and \( Z_{e} \) is any basis of \( N(A_e) \). The following lemma is a consequence of Proposition 2.2. This result has also been obtained by Hoeyer [30, Theorem 4.3] in the partitioned framework.

**Lemma 4.1:** Suppose that Assumptions A and B are satisfied and let \( x_0 \) be a solution of (1.1). There exists a positive constant \( C \) that depends only on \( f \) and \( c \) such that if \( x_k \) is sufficiently close to \( x_0 \), \( y_k \) is given by (4.5) and \( z_{k+1} \) is given by (4.6), we have

\[ ||z_{k+1} - x_0|| \leq C ||x_k - x_0||^2. \]  

From the quadratically convergent algorithm (4.5)–(4.6), a quasi-Newton method is easily introduced. In (4.6), \( G_{e} \) is replaced by an approximation \( G_{e} \) and \( Z_{e} \) is replaced by \( Z(y_k)^{-1} \), which intervenes in the calculation of the reduced gradient \( g(y_k) \). If \( A_e^- \) in (4.5) is replaced by \( A(x_k)^{-1} \), the constraints will have to be linearized twice per iteration: at \( x_k \) to calculate \( A(x_k) \) and at \( y_k \) to calculate the basis \( Z(y_k)^{-1} \). Since the constraints have to be linearized at \( y_k \) to calculate the reduced gradient in (4.6), we avoid one linearization of the constraints by replacing \( A_e^- \) in (4.5) by \( A(y_k^{-1}) \). So we obtain the following local algorithm:

\[ y_k := x_k - A(y_k^{-1}) c(x_k), \]  
\[ x_{k+1} := y_k - Z(y_k)^{-1} G_{e}^{-1} g(y_k). \]

We shall note \( H_k := G_{e}^{-1} \) and denote by \( r_k \) the restoration step and by \( t_k \) the tangent step:

\[ r_k := -A(y_k^{-1}) c(x_k), \]  
\[ t_k := -Z(y_k)^{-1} G_{e}^{-1} g(y_k). \]

We shall also use the total displacements

\[ d_k := r_k + t_k, \]  
\[ e_k := t_k + r_{k+1}. \]

In practice, the algorithm cannot start with (4.8) from a point \( x_0 \) without knowing a point \( y_0 \). So, we shall suppose in the following that the algorithm starts with (4.9) from a point \( y_0 \) in \( \omega \).
This is really the same type of algorithm as Coleman and Conn's method (1.15) -(1.17) if we exchange in (4.8)-(4.9) \( y_k \) with \( x_k \) and \( x_{k+1} \) with \( x_k + \xi_k \). However, our point of view shows that there is no reason to take a restoration step orthogonal to \( N(A(y_k-1)) \) in (4.10) or to calculate an orthogonal basis \( Z(y_k) \) of \( N(A(y_k)) \) in (4.11). In particular, this validates the use of the partitioned framework (3.5)-(3.7) that occurs often in practice, see also [30]. We shall see that contrary to the sequence \( (y_k) \) in (4.8)-(4.9), which does not usually converge superlinearly in one step (see the examples given by Byrd [8] and Yuan [39]), the sequence \( (x_k) \) converges superlinearly as expected from the behaviour of the ideal algorithm (4.5)-(4.6).

In fact, it is not essential to reduce the size of the optimality system before applying algorithm (2.6)-(2.8). The same method (4.8)-(4.9) can be obtained when the method is applied to the optimality conditions (1.3), see [25]. In this case, \( B_u = [L_u, A_u] \) and

\[
B_u = \begin{bmatrix} Z_a^{-1} \gamma \gamma^T \left( I - L_u Z_u G_a^T \gamma \gamma^T \right) 
\end{bmatrix},
\]

where \( A_u^{-1} \) is any right inverse of \( A_u \) and \( Z_u \) is any basis of \( N(A_u) \). Furthermore, this derivation of the algorithm gives an iteration scheme for the LAGRANGE multipliers \( \lambda_k \):

\[
\lambda_{k+1} = -A(y_k)^{-T} \nabla f(y_k) + A(y_k)^{-T} L_u Z(y_k) - G_u^{-1} g(y_k),
\]

(4.14)

where \( L_u \) is an approximation of \( L_u \). Therefore, if \( L_u \) in (4.14) is correctly updated, we shall also have \( \lambda_k = \lambda_{k+1} = 0(\|x_k - x_u\|) \); indeed, \( \lambda_k \) does not intervene in the iteration. Formula (4.14) simplifies the one obtained when quasi-Newton method is applied to (1.3), which can be written (see [20]):

\[
\lambda_{k+1} = -A(x_k)^{-T} \left( I - L_u Z(x_k) - G_u^{-1} Z(x_k) \right) (\nabla f(x_k) - L_u A(x_k) - c(x_k)).
\]

Algorithm (4.8)-(4.9) is a reduced quasi-Newton method because the only matrix to update is the approximation \( G_k \) of \( C_u \) and it is of order \( n = n \). Unfortunately, this is no longer the case when the sequence \( (\lambda_k) \) is generated by (4.14) since \( L_u \) intervenes in the formula and \( L_u \) is of order \( n \).

In the next two propositions, we study the local linear convergence of algorithm (4.8)-(4.9).

**Lemma 4.2:** Suppose that Assumptions A and B are satisfied. Then, there exist positive constants \( \delta, C_1, C_2 \) and \( C_3 \) that depend only on \( f, c \) and \( \omega \) such that on the one hand, \( \|G_k - C_u\| \leq \delta \) implies that \( G_k \) is nonsingular with \( \|G_k^{-1}\| \leq C_2 \) and on the other hand,

\[
\|y_{k+1} - x_u\| \leq (1 + C_1) \delta,
\]

(4.15)

\[
\|x_k - x_u\| \leq \delta,
\]

(4.16)

\[
\|G_k - C_u\| \leq \delta,
\]

(4.17)
with \( 0 < \delta \leq \delta \) imply that \( y_k \) and \( x_{k+1} \) are well defined by (4.8)–(4.9) and satisfy
\[
\|y_k - x_k\| \leq (1 + C_1) \delta , \tag{4.18}
\]
\[
x_{k+1} - x_k \| \leq C_2 \delta \|x_k - x_0\| . \tag{4.19}
\]

**Proof:** We shall denote by \( C_i (i = 1, 2, \ldots) \) any positive constant that depends only on \( f, c \) and \( w \). According to Assumptions A and B, there exists a positive constant \( C_1 \) such that
\[
\max \{||A(y)||, ||A(y')||, ||A(y')'||, ||y'(y)||\} \leq C_1 , \tag{4.20}
\]
for all \( y \) in \( \omega \) and
\[
\max \{||y_k - x_k||, ||y_k - x_0||\} \leq (1 + C_1) \|y_k - x_k\| , \tag{4.21}
\]
if \( y_{k-1} \) and \( x_k \) are in \( \omega \) and \( y_k \) and \( y_k \) are calculated from \( x_k \) by (4.5) and (4.8) respectively. Both inequalities in (4.21) are obtained from (4.5) and (4.8) by using TAYLOR’s expansion of \( c(x_k) \). Let \( \epsilon, \eta \) and \( \delta \) be three fixed positive constants such that
\[
B(x_k, \varepsilon) \subset \omega , \tag{4.22}
\]
\[
\|G\| \leq \eta , \tag{4.23}
\]
\[
\delta < \frac{1}{\eta} , \tag{4.24}
\]
\[
(1 + C_1) \delta < \varepsilon , \tag{4.25}
\]
where \( B(x_k, \varepsilon) \) denotes the ball of radius \( \epsilon \) centered at \( x_k \). These constants \( \epsilon, \eta \) and \( \delta \) depend only on \( f, c \) and \( \omega \). If \( G_k \) satisfies \( \|G_k - G\| \leq \delta \) then, by (4.24) and (4.23), \( G_k \) is nonsingular and satisfies (see for example [36, Chap. II, Theorem 62]):
\[
\|G_k\| \leq \frac{1}{1 - \delta} \tag{4.26}
\]
This proves the first part of the lemma.

For the second part, let us suppose that inequalities (4.15)–(4.17) are satisfied with \( \delta \) in \([0, \delta]\). According to (4.15), (4.25) and (4.22), \( y_k - 1 \) belongs to \( \omega \) and according to (4.16), (4.25) and (4.22), \( x_k \) belongs to \( \omega \). Therefore \( y_k \) is well defined by (4.8) and we have (4.21). This inequality and (4.16) show (4.18). Now, according to (4.21), (4.16), (4.25) and (4.22), \( y_k \) and \( y_{k-1} \) belong to \( \omega \). So, \( x_{k+1} \) and \( x_{k+1} \) are well defined by (4.8) and (4.9) respectively. From (4.26), (4.23), (4.17) and \( G_k - G_k = G_k (G_k - G_k) G_k \), we deduce:
\[
\|G_k - G_k\| \leq \|G_k\| \|G_k - G_k\| \leq C_0 \|G_k - G_k\| = : C_0 \delta . \tag{4.27}
\]
Let \( C_0 \) be the constant given by Lemma 4.1. Then, TAYLOR expansions give easily the following inequalities:
\[
\|y_k - x_k\| = \|x_k - x_{k+1}\| + \|x_{k+1} - x_0\| ,
\]
\[
\|y_k - x_k\| = C_0 \|y_k - y_{k-1}\| + C_0 \|y_k - x_k\| + C_0 \|y_k - x_k\| + \|x_k - x_0\| ,
\]
\[
\|y_k - y_{k-1}\| = C_0 \|y_k - x_k\| + \|x_k - x_0\|. \tag{4.29}
\]

"Tous droits de propriété intellectuelle réservés. Reproduction reproduction interdite sauf autorisation écrite du propriétaire intellectuel."
By combining these inequalities with (4.7), (4.15), (4.16) and (4.21), we get (4.19) with 
\[ C_2 = C_5 + C_6 C_6 (1 + C_1) + C_7 (1 + C_1)^2 + C_8 (1 + C_1). \]

**Theorem 4.3:** Suppose that Assumptions A and B are satisfied. Then, there exists a positive constant \( C \) that depends only on \( f \), \( c \) and \( \omega \) such that if \( x \) is a real number in \([0, 1] \) and if

\[
\|y_0 - x_*\| \equiv C x, \\
\|G_0 - G_*\| \equiv C x \quad \text{for all subscripts } k,
\]

then algorithm (4.8)–(4.9) generates from \( y_0 \) a sequence \( (x_k) \) in \( \omega \) that converges linearly to \( x_* \) and

\[
\|x_{k+1} - x_*\| \equiv \kappa \|x_k - x_*\|,
\]

for all subscripts \( k \).

**Proof:** Let \( \delta, C_1, C_2 \) and \( C_3 \) be the positive constants given by Lemma 4.2. Then, if \( G_{k-1} \) satisfies \( \|G_{k-1} - G_*\| \leq \delta \), we have \( \|G_{k-1} - G_*\| \leq C_3 \). By expanding \( g(y_{k-1}) \) about \( x_* \) (with (4.2)), (4.9) shows that for \( k \geq 1 \), we have

\[
\|x_k - x_*\| \leq (1 + C_4) C_\kappa \|y_k - y_{k-1}\|,
\]

where \( C_4 \) is a positive constant that depends only on \( f, c \) and \( \omega \). Then, the theorem can be proved with \( C := \min (\delta, 1/C_3)/(1 + C_4) \). Indeed, if (4.27) and (4.28) are satisfied, we see, with the help of (4.30), that (4.15)–(4.17) are satisfied for \( k = 1 \) and \( \delta = (1 + C_4) C_\kappa \equiv \delta \):

\[
\|y_0 - x_*\| \equiv C x \equiv \delta \leq (1 + C_1) \delta, \\
\|x_1 - x_*\| \leq (1 + C_4) \|y_0 - x_*\| \equiv \delta, \\
\|G_0 - G_*\| \equiv C x \equiv \delta.
\]

We can then apply Lemma 4.2 and because \( C_2 \delta \equiv \kappa \), (4.19) shows that (4.20) is satisfied for \( k = 1 \). The fact that \( \kappa \) is less than 1, (4.28) and (4.18) for \( k = 1 \) show that (4.15)–(4.17) are still satisfied for \( k = 2 \). So we can conclude by induction.

The next proposition gives some useful estimates and equivalences.

**Proposition 4.4:** Suppose that Assumptions A and B are satisfied. Let \( (G_k) \) be a sequence of nonsingular matrices of order \( n \). Let \( (x_k) \) in \( \omega \) and \( (y_k) \) in \( \omega \) be the sequences generated by algorithm (4.8)–(4.9) starting from a point \( y_0 \) in \( \omega \). If \( (x_k) \) and \( (y_k) \) converge to a solution \( x_* \) of (1.1), we have

\[
r_k = -A_* x_* - x_* + \alpha (\|x_k - x_*\|), \tag{4.31}
\]

\[
y_k - x_* = Z_* Z_* (x_k - x_k) + \alpha (\|x_k - x_*\|). \tag{4.32}
\]

If moreover \( (G_k) \) and \( (G_k^{-1}) \) are bounded, we have

\[
\|A_k\| \sim \|x_k - x_*\|, \tag{4.33}
\]

\[
\|y_k - x_*\| \sim \|x_k - x_*\|. \tag{4.34}
\]

**Proof:** From the definition (4.10) of \( r_k \), the expansion of \( c(x_k) \) about \( x_* \) and the convergence of \( (y_k) \), we get (4.31). Then, using identities (3.3) and (4.31) on \( y_k = \ldots \)
Next, using the boundedness of \((G_k^{-1})^T\) and (4.2), we see that
\[
t_k = -Z_k G_k^{-1} Z_k^T I_k (y_k - x_k) + o(||y_k - x_k||)
\]
Finally, with (4.12), (4.31) and (4.32), this gives
\[
d_k = -A_k^{-1} A_k + Z_k G_k^{-1} G_k Z_k (x_k - x_k) + o(||x_k - x_k||)
\]
This estimate shows that \(d_k = O(||x_k - x_k||)\). To prove the \(x_k - x_k = O(||d_k||)\), we only have to show that the operator in square brackets is nonsingular with bounded inverse. If this were not the case, there would exist a subsequence \(K\) of subscripts and a sequence \((\bar{x}_k) \in K\) in \(R^n\) such that:
\[
||\bar{x}_k|| = 1 \text{ for } k \in K, \quad (4.35)
\]
\[
(A_k^{-1} A_k + Z_k G_k^{-1} Z_k) \bar{x}_k = 0 \text{ for } k \in K. \quad (4.36)
\]
By multiplying (4.36) by \(A_k\) (resp. \(Z_k\)), we would obtain \(A_k \bar{x}_k = 0\) (resp. \(G_k^{-1} G_k \bar{x}_k = 0\)), from which we would deduce \(Z_k \bar{x}_k = 0\) because of the boundedness of \((G_k)\) and the nonsingularity of \((G_k)^{-1}\). Finally, with (3.3), we would have \(\bar{x}_k = 0\), which would contradict (4.35). So (4.33) is proved. The proof of (4.34) is similar and based on the estimate
\[
e_k = -A_k^{-1} A_k + Z_k G_k^{-1} Z_k^T I_k (y_k - x_k) + o(||x_k - x_k||) \quad \Box
\]

5. Conditions for Superlinear Convergence

Theorem 4.3 has an immediate corollary, which states that if in addition to (4.28), the sequence \((G_k)\) converges to \(G_a\) then \((x_k)\) converges to \(x_a\) superlinearly (see for example the argument in the proof of corollary 3.5 in Han [28]). However, this assumption on \((G_k)\) is usually not satisfied when these matrices are generated by quasi-Newton formulas. Assuming that \((x_k)\) converges to \(x_a\), the next theorem gives necessary and sufficient conditions on \((G_k)\) to have the superlinear convergence of \((x_k)\). It is the analogue of Theorem 2.2 of Dennis and Moré [15], valid for quasi-Newton methods in optimization without constraints.

**Theorem 5.1:** Suppose that Assumptions A and B are satisfied and that \((y_k)\) and \((x_k)\) are generated in \(\omega\) from a point \(y_0\) by algorithm (4.8)-(4.9) with a sequence \((G_k)\) of nonsingular matrices. Suppose that \((x_k)\) and \((y_k)\) converge to \(x_a\). Then, the following statements are equivalent:

(i) \((x_k)\) converges superlinearly,

(ii) \(g(y_{k+1}) = o(||x_{k+1} - x_k||)\),

(iii) \((G_k - G_a) z_k = o(||x_k - x_a||)\).

**Proof:** The estimate (4.32) shows that
\[
y_k - x_k = O(||y_k - x_k||) \quad (5.1)
\]
Using \(A(y_k) - A_a = O(||y_k - x_k||)\), \(t_k = 0\) (because \(t_k = x_{k+1} - y_k\)), (4.32) and \(A_a x_a = 0\), we get:
\[
A_a (y_k - x_k) + A(y_k) t_k + o(||y_k - x_k||) = o(||x_k - x_a||) \quad (5.2)
\]
According to (3.3), it remains to estimate \( Z_e (x_{k+1} - x_a) \). This will depend on the quality of the tangent step \( t_k \).

Let us first prove the equivalence (i) \( \Rightarrow \) (ii). With (4.2) and (4.32), we have
\[
g(y_{k+1}) = Z_e^T L_e (y_{k+1} - x_a) + o (||y_{k+1} - x_a||) - G_e Z_e (x_{k+1} - x_a) + o (||x_{k+1} - x_a||).
\]

Then, (ii) is clear from (i). If (ii) is satisfied, this estimate and the nonsingularity of \( G_e \) give
\[
Z_e (x_{k+1} - x_a) = o (||x_{k+1} - x_a||) + o (||x_k - x_a||).
\]

This estimate, (5.2) and identity (3.3) show (i).

Now, let us show that in any of the situations (i), (ii) or (iii), we have
\[
t_k = O (||x_k - x_a||) \tag{5.3}
\]

This estimate is clear when \((G_e^{-1})\) is bounded, but we do not suppose this here. Writing \( t_k = (x_{k+1} - x_a) - (y_k - x_a) \) and using (5.1), we see that (5.3) is clearly satisfied when (i) is true and therefore when (ii) is true. When (iii) is satisfied, we have
\[
G_e Z(y_k) t_k = G_e Z(y_k) t_k + o (||x_k - x_a||) = -g(y_k) + o (||x_k - x_a||).
\]

Then expanding \( g(y_k) \) about \( x_a \) and using (5.1) and the nonsingularity of \( G_e \), we get
\[
Z(y_k) t_k = O (||x_k - x_a||) \tag{5.4}
\]

But \( t_k = Z(y_k) - Z(y_k) t_k \), therefore (5.3) is still satisfied.

Now, from (5.1) and (5.3), it follows that \( x_{k+1} - x_a = y_k - x_a + t_k = O (||x_k - x_a||) \), \( y_{k+1} - y_k = (y_{k+1} - x_a) - (y_k - x_a) = O (||x_k - x_a||) + O (||x_k - x_a||) = O (||x_k - x_a||) \), and, with (4.31) and (5.2),
\[
t_{k+1} = o (||x_k - x_a||) \tag{5.4}
\]

Let us now prove the equivalence (ii) \( \Rightarrow \) (iii). Expanding \( g(y_{k+1}) \) about \( y_k \) and using (4.2), (5.3) and (5.4), we have
\[
g(y_{k+1}) = g(y_k) + Z_e^T L_e t_k + o (||x_k - x_a||).
\]

But \( g(y_k) = -G_e Z(y_k) t_k \) and \( t_k = Z(y_k) - Z(y_k) t_k = Z_e Z(y_k) t_k + o (||x_k - x_a||) \). So, we obtain
\[
g(y_{k+1}) = - (G_e - G_a) Z(y_k) t_k + o (||x_k - x_a||).
\]

The equivalence (ii) \( \Rightarrow \) (iii) follows.

In statement (ii) of Theorem 5.1, \( g(y_{k+1}) \) could be replaced by \( g(x_{k+1}) \), but the reduced gradient is not evaluated at \( x_{k+1} \) in the algorithm. Statement (iii) is equivalent to
\[
(H_k - H_a) g(y_k) = o (||x_k - x_a||),
\]

which is based on the gap between the inverse of the reduced Hessians. Statement (iii) can also be replaced by many other equivalent estimates. For example, \( (G_e - G_a) Z_e (x_k - x_a) = o (||x_k - x_a||) \). The advantage of (iii) is that it does not re-
quire the boundedness of the sequences \((g_k)\) or \((G_k^{-1})\). If this boundedness is assumed, Proposition 4.4 shows that the estimates can be made in relation to \(\|d_k\|\) rather than \(\|x_k - x_0\|\).

Condition (4.28) and condition (iii) of Theorem 5.1 show the advantage of reduced quasi-Newton methods over the SQP method with regard to the approximation of the Hessian of the Lagrangian. Indeed, a necessary and sufficient condition for the SQP method to generate superlinearly convergent sequences is that

\[
Z_k^{-T} (L_k - L_\phi) (x_k - x_0) = o (\|x_k - x_0\|),
\]

where \(L_k\) is the updated approximation of \(L_\phi\). This famous result can be found in [4] and [34]. Therefore, in the SQP method, the \((n - m) \times n\) matrix \(Z_k^{-T}L_k\) has to be correctly approximated and not only the projected Hessian of the Lagrangian \(Z_k^{-T}L_kZ_k\) as in reduced methods.

As a final remark, let us mention that, if we suppose that the equivalence (4.34) holds, the sufficient condition given by Byrd [7] can be written

\[
(G_k - G_\phi) Z(y_k) t_k = o (\|y_k - x_0\|)
\]

and is therefore stronger than condition (iii) of Theorem 5.1. However, this estimate (5.5) is satisfied in practice with the update schemes currently proposed: Coleman and Conn [13, Theorem 3.6] proved it for their algorithm and it is proved in [22] that

\[
(G_k - G_\phi) Z(y_k) t_k = o (\|t_k\|)
\]

(which implies (5.5)) for both of the algorithms proposed. But those schemes are not completely satisfactory and condition (iii) of Theorem 5.1 may become useful in other circumstances. We shall see in the next section that, for the globalization technique proposed, the strongest condition (5.6) allows avoidance of the Maratos effect, i.e., allows the unit step-size to be accepted asymptotically.

6. Globalization of the Algorithm

In order to globalize the local algorithm (4.8)–(4.9), we introduce a step-size \(\alpha_k\). For that, we consider the following exact penalty function:

\[
\Theta_p(x) = f(x) + p\|c(x)\|_1,
\]

where \(p\) is a positive penalty parameter and \(\|\cdot\|_1\) is the \(l_1\)-norm on \(\mathbb{R}^m\). Other norms than the \(l_1\)-norm can be used in (6.1): see [6]. If \(p\) is taken to be greater than \(\|x_0\|_\infty\) (here it is the dual norm of the one used in (6.1) that is relevant), the feasible minimizers of (1.1) and (6.1) are the same (see [17], for example). It is therefore natural to look for \(x_\alpha\) by minimizing \(\Theta_p\). For this, we need to calculate descent directions of this non-differentiable function. On that point, a crucial observation has been made by Han [29]: the displacement \(d^2_k \Theta_p\) of the

("Tous droits de propriété intellectuelle réservés. Réproduction, représentation interdites sans autorisation (code de la propriété intellectuelle")
SQP method is a descent direction of \( \Theta_p \) at \( x_k \) (under some hypotheses). Therefore a better approximation \( x_{k+1} \) of the solution \( x_* \) will be obtained by taking

\[
x_{k+1} = x_k + \theta d_k^{\text{SQP}},
\]

where \( \theta_k \) gives the step-size and is obtained from some rule using \( \Theta_p \) as a “merit” function.

Let us try to use the same globalization technique for our algorithm. Is there any descent direction of \( \Theta_p \) among the displacements \( r_k, t_k, d_k, \) and \( e_k \) given by (4.10)–(4.13)? The inconvenience of \( r_k \) and therefore of \( d_k \) and \( e_k \) is that this displacement is calculated by using two different points \( y_{k-1} \) and \( x_k \) that can be far from each other when \( x_k \) is far from \( x_* \). So, it is difficult to see when these directions are descent directions for \( \Theta_p \). On the other hand, \( t_k \) uses only the point \( y_k \) in its definition and if \( \Theta_k \) is positive definite, it is certainly a descent direction of \( \Theta_p \) at \( y_k \). Indeed, this displacement is tangent to \( c^{-1}(c(y)) \) at \( y_k \) and \( f'(y_k) \cdot t_k \) is negative. Therefore at the first order, the first term of the right hand side of (6.1) will decrease while the second term will remain constant. These remarks lead us to define a descent arc of \( \Theta_p \) at \( y_k \), tangent to \( t_k \):

\[
y_k(a) = y_k + e_k + a^2 r_{k+1}, \quad a>1.
\]

Let us note that search arcs have already been proposed by Mayne and Polak [32] to cope with the Maratos effect in the SQP method (see further) and by Garay [20] also to avoid the Maratos effect in algorithm (1.12)–(1.14).

This globalizing technique based on the arc (6.2) gives priority to the minimization step \( t_k \); and this is due to the asymmetry of the local method (4.8)–(4.9). This priority can be harmful in certain circumstances but it can be suppressed by adding a restoration step to the local method (see [23]).

The point \( y_{k+1} \) is then obtained from \( y_k \) by selecting a particular value \( \theta_k \) of \( a \):

\[
y_{k+1} = y_k(\theta_k).
\]

The step-size \( \theta_k \) will be determined here so that the following Armijo-like criterion will be satisfied:

\[
\beta \in ]0, 1[ , \quad \theta_k = \beta l_k ,
\]

where \( l_k \) is the smallest non-negative integer such that

\[
\Theta_p(y_k(\beta l_k)) = \Theta_p(y_k) + \beta^2 k_f(y_k) \cdot t_k - \beta^2 a_k \lambda_k (p - \|\lambda(y_k)\|_\alpha) \|c(y_k)\|_1.
\]

In this inequality, \( \alpha \) is a real number chosen in \( ]0, 1/2[ \) for reasons that will be clear at the end of this section. The exponent \( \theta l_k \) of \( \beta \) in the last term of (6.6) takes into account the curvature of the search path (6.2). The vector \( \lambda(y_k) \) is an approximation at \( y_k \) of the LAGRANGE multiplier \( \lambda_* \). It is defined by

\[
\lambda(y) = - A(y)^-T \nabla f(y).
\]

It is just the first term of (4.14). So, usually, \( \lambda(y_k) \) will not converge superlinearly.
We shall define again

\[ x_{k+1} := y_k + t_k. \]  

(6.8)

Now, we have to examine in what conditions inequality (6.6) can be realized with a large enough \( I_k \). This is the subject of the following lemma.

**Lemma 6.1:** Suppose that Assumptions A and B are satisfied and that a point \( y_k \) is given in \( \omega \) such that \( x_{k+1} \) and \( x_{k+1} + r_{k+1} \) will also be in \( \omega \). Suppose that \( a \) is in \( [0, 1] \) and that there exist positive constants \( p, \bar{p}, h \) and \( \alpha, \beta \) and \( M \) such that

\[ p + \| \lambda (y_k) \| \leq \bar{p}, \]

\[ h I \equiv C_k^{-1} \equiv h^{-1} I. \]

Then the rule (6.4)–(6.6) allows determination of a positive step-size \( q \). If, moreover, \( M \) is a positive constant such that

\[ \| c(y_k) \| \leq M, \]

then there exists a positive real \( \bar{q} \) that depends only on \( f, c, \bar{p}, \bar{h}, \alpha, \beta \) and \( M \) such that

\[ q = \bar{q} > 0. \]

**Proof:** We shall denote by \( C_i \) \((i = 1, 2, ...)\) any positive constant. Using \( c'(y_k) \cdot t_k = 0 \), **Taylor's** theorem gives

\[ \| c(x_{k+1}) - c(y_k) \| \leq C_1 \| f_k \| \]

(6.11)

where \( C_1 \) depends only on \( c \). Using (6.11) and 0 < \( q \leq 1 \) and applying again **Taylor's** theorem, we get

\[ f(y_k + q \theta r_{k+1}) = f(y_k) + q \theta (y_k - y_k) + \frac{q^2}{2} \theta r_{k+1} + \frac{q^3}{6} \theta r_{k+1} \]

\[ + \frac{q^4}{24} \theta r_{k+1} + \frac{q^5}{120} \theta r_{k+1} \]

(6.12)

where \( C_2, C_3 \) and \( C_4 \) depend only on \( f \) and \( c \). Supposing \( q \) in \( [0, 1] \), we get from (6.12) and (6.13):

\[ \Theta_p [y_k + q \theta r_{k+1}] = \Theta_p (y_k) + q \theta (y_k - y_k) + \Theta_p (y_k) \cdot \theta r_{k+1} \]

(6.13)

where \( C_5 \) and \( C_6 \) depend only on \( f, c, \) and \( \bar{p} \). From the definition (4.10) of \( r_{k+1} \) and (6.11), we have

\[ \| r_{k+1} \| \leq C_7 \| c(y_k) \| + C_8 \| f_k \| \]

(6.14)

where \( C_7 \) and \( C_8 \) depend only on \( c \).

Now, let us suppose that (6.6) is not true for a given \( q = \beta \) in \( [0, 1] \). Then, with (6.9), (6.14) and (6.15), we get

\[ q \left( 1 - q \right) ( - f'(y_k) \cdot t_k + q \left( 1 - q \right) \bar{p} \| c(y_k) \| + C_6 \| f_k \| \]

\[ + C_8 q \| r_{k+1} \|^2. \]
But \(-f(y_k) \cdot t_k = g(y_k)^T C_k^{-1} g(y_k) \equiv h \|g(y_k)\|^2\) and \(\|t_k\| \equiv C_9 \|g(y_k)\|\) where \(C_9\) depends only on \(c\) and \(h\). Then, with (6.15), the last inequality becomes
\[
\varrho \|g(y_k)\|^2 + \varrho^a \|c(y_k)\| \equiv C_{10} (\varrho^a + \varrho^a) \|g(y_k)\|^2 + C_{11} \varrho \|c(y_k)\|^2,
\]
where \(C_{10}\) and \(C_{11}\) depend only on \(f\), \(c\), \(p\), \(\bar{p}\), \(\bar{h}\) and \(\alpha\). This inequality shows that \(\varrho\) cannot be arbitrarily small if \(\|g(y_k)\| + \|c(y_k)\| 
= 0\) (if \(\|g(y_k)\| + \|c(y_k)\| \equiv 0\), \(\varrho = \beta = 1\) clearly satisfies (6.6) because then \(t_k = 0\), \(r_{k+1} = 0\) and therefore \(y_k(\varrho) = y_k\). Indeed, otherwise letting \(\varrho\) converge to 0 in (6.16) previously divided by \(\varrho\), we would obtain \(g(y_k) = 0\) and then dividing (6.16) by \(\varrho^a\) and taking the limit on \(\varrho\) would give \(c(y_k) = 0\). This proves the first part of the lemma.

For the second part, let us suppose that rule (6.4)–(6.6) gives a step-size \(\varrho_1\) smaller than 1. Then (6.6) is not satisfied with \(\varrho = \varrho_1/\beta\) and we have inequality (6.16) for this \(\varrho\). Using (6.10), we obtain:
\[
\varrho \|g(y_k)\|^2 + \varrho^a \|c(y_k)\| \equiv C_{12} \varrho \left( \varrho \|g(y_k)\|^2 + \varrho^a \|c(y_k)\| \right),
\]
where \(b := \min (1, \alpha - 1)\) and \(C_{12}\) depends only on \(f\), \(c\), \(p\), \(\bar{p}\), \(\alpha\), \(h\) and \(M\). Because \(\varrho \|g(y_k)\|^2 + \varrho^a \|c(y_k)\| \equiv 0\) (otherwise \(\varrho_1 = 1\)), the last inequality proves the second part of the lemma with \(\varrho = \varrho_1\)\(^b\).

Inequality (6.9) shows that the penalty parameter \(p\) has to be large enough to ensure the decrease of \(\Theta_p\) along the arc (6.2) and that its lower bound depends on the current point \(y_k\). So, sometimes it will be necessary to update the penalty parameter, which we shall denote by \(p_k\). We shall suppose that the adapting rule of \(p_k\) will satisfy the following three conditions:
\[
\begin{align*}
\|\lambda(y_k)\| &= p_k, \quad \text{for every } k; \quad \text{(6.17)} \\
\text{there exists a subinterval } K \text{ such that for every } k \text{ greater than } K, \\
& \quad (p_{k+1} \equiv \|\lambda(y_k)\| \equiv p_k) \quad \text{implies that } p_k = p_{k-1}; \quad \text{(6.18)} \\
& \quad (p_k) \text{ is bounded if and only if } p_k \text{ is modified finitely often.} \quad \text{(6.19)}
\end{align*}
\]
In (6.17) and (6.18), \(p\) is a given positive constant. Condition (6.18) means that eventually (for \(k \equiv K\)), \(p_k\) is modified only if it is necessary to have (6.17). So \((p_k | k \equiv K)\) is an increasing sequence. An example of an adapting rule satisfying these conditions is given by MAYNE and POLAK [32]:
\[
\begin{align*}
& \quad \text{if } p_{k-1} = \|\lambda(y_k)\| \equiv p_k, \\
& \quad \text{then } p_k := p_{k-1}, \\
& \quad \text{else } p_k := \max (\delta p_{k-1}, \|\lambda(y_k)\| \equiv p_k),
\end{align*}
\]
where \(\delta\) is a given constant greater than 1.

We are now able to state the algorithm that globalizes the local method (4.8)–(4.9).

**Algorithm RQN:**
1. Choose a convergence tolerance \(\varepsilon > 0\), \(\beta \in [0, 1]\), \(\alpha \in [0, 1/2]\) and \(d > 1\).
2. Choose \(y_0\) to \(\omega\) and a symmetric positive definite matrix \(G_0\) of order \(n - m\).
3. Let \( k := 0 \).
4. Repeat:
   4.1. Linearize the constraints at \( y_k \): choose a right inverse \( A(y_k)^{-1} \) of \( \nabla c(y_k) \) and a basis \( Z(y_k) \) of \( N(\nabla c(y_k)) \) according to assumption B.
   4.2. Evaluate \( \lambda(y_k) := -A(y_k)^{-T} \nabla f(y_k) \) and \( g(y_k) := Z(y_k)^{-T} \nabla f(y_k) \).
   4.3. If \( k \geq 1 \) then evaluate the symmetric positive definite matrix \( G_k \) by updating \( G_{k-1} \).
   4.4. Tangent step: evaluate \( t_k := -Z(y_k)^{-1} G_k^{-1} g(y_k) \) and \( x_{k+1} := y_k + t_k \).
   4.5. Restoration step: evaluate \( c(x_{k+1}) \) and \( r_{k+1} := -A(y_k)^{-1} c(x_{k+1}) \).
   4.6. If \( ||g(y_k)|| + ||c(x_{k+1})|| \approx \varepsilon \) then stop.
   4.7. Adapt \( p_k \) according to (6.17)–(6.19).
   4.8. Search a point \( y_{k+1} \) from \( y_k \) along the arc (6.2) in order to decrease the penalty function (6.1) (with \( p = g_k \)) with the help of the rule (6.3)–(6.6).
   4.9. Next iteration: set \( k := k + 1 \).

In the partitioned framework (see Section 3), only one linear system has to be solved at the step 4.2. Indeed, if \( A(y_k) = [C(y_k) \ D(y_k)] \), \( \lambda(y_k) \) is obtained by solving

\[
C(y_k)^T \lambda(y_k) = -\nabla f(y_k),
\]

where \( \nabla f(y_k) \) is the vector formed by the first \( m \) components of \( \nabla f(y_k) \). Then \( g(y_k) = D(y_k)^T \lambda(y_k) + \nabla f(y_k) \), where \( \nabla f(y_k) \) is the vector formed by the last \( n - m \) components of \( \nabla f(y_k) \).

The important question of the update of the matrices \( (G_k) \), which is mentioned at step 4.3 of Algorithm RQN, has been investigated by Coleman and Conn [13] and by Gillbert [22], [25]. See Noedel and Overton [34] for algorithm (1.12)–(1.14). \( G_{k+1} \) is obtained from \( G_k \) by the BFGS formula using two vectors \( g_k \) and \( s_k \) in \( \mathbb{R}^{n-m} \):

\[
G_{k+1} = G_k - \frac{g_k g_k^T}{g_k^T s_k} + \frac{s_k s_k^T}{s_k^T s_k}.
\]

Therefore, \( G_{k+1} \) satisfies the secant equation:

\[
\gamma_k = G_{k+1} s_k.
\]

The point now is to choose adequately the vectors \( \gamma_k \) and \( s_k \) so that \( G_{k+1} \) will approximate \( G_e := Z_e^{-T} L_e Z_e^{-1} \). This form of \( G_e \) and formula (4.2) suggest taking \( \gamma_k \) as the difference of two reduced gradients. A first possible choice consists in taking:

\[
\gamma_k = g(x_{k+1}) - g(y_k),
\]

\[
s_k = -G_e^{-1} g(y_k).
\]

Then, if we suppose that \( (y_k) \) and \( (x_k) \) converge to \( x_e \), Taylor's theorem gives:

\[
\gamma_k = G_e s_k + o(||x_e||)
\]

This relation and (6.21) show that \( \gamma_k \) and \( s_k \) are correctly chosen. Unfortunately, this choice needs an additional linearization of the constraints at \( x_{k+1} \) in order to
calculate the reduced gradient at this point. This may be avoided by taking:

$$

gamma_k = g(y_{k+1}) - g(y_k),
$$

(6.26)

$$

sigma_k = -gamma_k \cdot G_k^{-1} g(y_k).
$$

(6.26)

But in this case, (6.24) will not be necessarily satisfied any more, which means that updating $G_k$ by formula (6.20) would deteriorate the matrix. Therefore, an update criterion of the form

$$
\|G_k^{(k+1)}\| \leq \mu_k \|E_k\|,
$$

(6.27)

where $(\mu_k)$ is an appropriate sequence converging to zero, has to be introduced. When (6.27) is satisfied, it is not too difficult to show that the estimate (6.24) is still valid with $\gamma_k$ and $\sigma_k$ given by (6.25) and (6.26). The crucial point is now to choose correctly the sequence $(\mu_k)$ so that when (6.27) is not satisfied the super-linear rate of convergence of $(x_k)$ can be preserved. A good choice for $\mu_k$ is

$$
\mu_k = \mu \|c_k\|,
$$

where $\mu$ is a small enough constant and $(k \to -)$ is the subscript of the last but one iteration at which (6.27) was satisfied, i.e. at which $G_{k-1}$ was updated by formula (6.20).

The update scheme at step 4.3 of Algorithm RQN is expected to generate a sequence of nonsingular matrices $G_k$ satisfying

$$
\|I - G_k^{-1}\| \leq h^{-1},
$$

(6.28)

for some positive constant $h$. This property is really not easy to obtain. However, using the same type of arguments that are used in unconstrained optimization, it can be proved either in a local framework (when $(x_0, G_0)$ is supposed to be close to $(x_*, G_*)$ and $g_k(1)$) or when it is assumed that $(x_k)$ and $(y_k)$ converge to $x_*$ with

$$
\sum_{k=0}^{\infty} \|x_k - x_*\| = +\infty \quad \text{and} \quad \sum_{k=0}^{\infty} \|y_k - x_*\| = +\infty.
$$

See [28].

The next theorem gives some global convergence result for Algorithm RQN under hypothesis (6.28).

**Theorem 6.2**: Suppose that Assumptions A and B are satisfied and that $f$ is bounded from below on $\omega$. Let $(x_k)$, $(y_k)$ and $(G_k)$ be the sequences generated by Algorithm RQN with $a$ in $[0, 1]$. Suppose that $(x_k)$ and $(y_k)$ are in $\omega$ and that the matrices $G_k$ are nonsingular and satisfy (6.18) with a positive constant $h$ independent of $k$. Then, either $(p_k)$ is unbounded and $(y_k | p_k + p_{k-1})$ has no accumulation point in $\omega$, or $(p_k)$ is bounded and

$$
\|g(y_k)\| + \|c(y_k)\| = 0.
$$

(6.29)

**Proof**: Suppose first that $(p_k)$ is unbounded and let $k$ be the subsequence of the subscripts $k \geq K$ $(K$ given in (6.18)) for which $p_k + p_{k-1}$. By (6.18),

$$
p_{k-1} < \|c(y_k)\| + p_k.
$$

("Tous droits de propriété intellectuelle réservés. Reproduction représentation interdites sans autorisation (code de la propriété intellectuelle)")
for \( k \in K \). Because \( (p_k : k \geq K) \) is an increasing sequence, we see from this inequality that \( \| \lambda(y_k) \| \rightarrow \infty \) for \( k \rightarrow \infty \) in \( K \). Therefore \( (y_k : p_k + p_{k-1}) \) has no accumulation point in \( \omega \) (here, we use the continuity of \( y \rightarrow \lambda(y) \) and therefore, the surjectivity of \( \nabla c(y) \) and assumption B are strongly invoked).

Now, let us suppose that \( (p_k) \) is bounded. From (6.19), \( y_k \) is constant when \( k \) is great enough. Let us say that \( p_k = p \) for \( k \geq K \). So, at each iteration the same penalty function \( \Theta_p \) decreases. The function \( f \) being bounded from below, we get
\[
\| c(y_k) \| \leq \Theta_p(y_k) - \inf f, \quad \text{for } k \geq K.
\]

Therefore, \( \| c(y_k) \| \) is bounded and we can apply Lemma 6.1, which states the existence of a positive lower bound \( q \) for the sequence \( (q_k) \). From (6.28), we get
\[
-\| c(y_k) \| + t_k \leq \| c(y_k) \| + q_k \leq \Theta_p(y_k) - \Theta_p(y_{k+1}), \quad \text{for } k \geq K.
\]

But \( (\Theta_p(y_k)) \) converges (a decreasing bounded from below sequence). Therefore, taking the limit on \( k \) in this inequality shows that \( \| c(y_k) \| \) and \( \| c(y_k) \| \) converge to zero.

The last problem we tackle concerns the admissibility of the unit step-size. When \( q_k = 1 \) is accepted by (6.6), Algorithm RQN proceeds like the local method (4.8)-(4.9) and superlinear convergence of \( (x_k) \) will occur when the reduced Hessian \( G_k \) is correctly approximated by \( G_k \) (see Theorem 5.1, statement (iii)). It is known that this admissibility property is not satisfied when the SQP method is globalized with the penalty function (5.1) and the technique described at the beginning of this section. This has been called the "Maratos effect" of the SQP method (see Maratos [31]) and several remedies have been proposed to overcome this drawback: see [20], [10], [32] and [5]. This inconvenience is not shared with our algorithm. In fact, when \( c(y_k) = 0 \), which is a favourable situation for the appearance of the Maratos effect, the total displacement \( s_k = t_k + r_{k+1} \) is exactly the same as that of the SQP method with the Mayne and Polak correction.

Let \( (x_k) \) in \( \omega \), \( (y_k) \) in \( \omega \) and \( (G_k) \) be the sequences generated by Algorithm RQN and suppose that \( (y_k) \) converges to a solution \( x_\ast \) of (1.1). Let \( K \) be a subsequence of subscripts. We are interested in finding conditions under which \( q_k \) will be equal to 1 for all but finitely many subscripts \( k \) in the subsequence \( K \). The following four properties will be meaningful:
\[
\| G_k - G_\ast \| \leq M \quad \text{for } k \in K, \tag{6.30}
\]
\[
(G_k - G_\ast) Z_k t_k = o(\| t_k \|) \quad \text{for } k \in K, \tag{6.31}
\]
\[
t_k = O(\| t_k \|) \quad \text{for } k \in K, \tag{6.32}
\]
\[
q_k = 1 \quad \text{and } t_k = o(\| t_k \|) \quad \text{for } k \in K. \tag{6.33}
\]

Properties (6.30) and (6.31) concern the approximation of the reduced Hessian \( G_\ast \) by \( G_k \). Property (6.30) is very strong when \( M \) is small and is usually not satis-
fied when second order derivatives are not calculated. Property (6.31) recalls condition (iii) of Theorem 5.1, which is when \((G_k)\) and \((G^{-1}_k)\) are bounded:
\[
(G_k - G^*_k)Z^T d_k = o(||d_k||). \tag{6.34}
\]
Therefore, (6.31) is usually stronger than (6.34) and, in fact, is satisfied by some subsequences of subscripts when \((G_k)\) is updated by the BFGS formula (see the discussion following the proof of Theorem 5.1). Property (6.32) concerns the comparison of the tangent step and the restoration step.

The next theorem shows that for the subsequences \(K\) for which (6.30) with \(M\) small enough or (6.31) or (6.32) is satisfied, the rule (6.3)–(6.6) will give \(g_k = 1\) for all but finitely many \(k\) in \(K\). Therefore, the unit step-size will be admissible either when \(G_k\) is correctly approximated by \(G_k\) (properties (6.30) and (6.31)) or when \(t_k\) is of the same order of magnitude as \(r_{k+1}\) (property (6.32)).

Property (6.33) is more particular. In concrete algorithms using the update scheme (6.20), (6.25) and (6.26) with the update criterion (6.27), neither of the properties (6.31)–(6.33) is satisfied for the entire sequence. These properties are satisfied only for subsequences \(K\). The result obtained in Theorem 6.3 with property (6.33) is then used to prove that only property (6.31) or (6.32) may occur in the considered algorithm (see [28]).

**Theorem 6.3:** Suppose that Assumptions A and B are satisfied. Let \((x_k)\), \((y_k)\) and \((G_k)\) be the sequences generated by Algorithm RQN with \(a\) in \([0, 1/2]\). Suppose that \((x_k)\) and \((y_k)\) are in \(K\), that \((y_k)\) converges to \(x_\infty\) and that the matrices \(G_k\) are nonsingular and satisfy (6.28) with a positive constant \(h\) independent of \(k\). Let \(K\) be a subsequence of subscripts. Then,

(i) there exists a positive constant \(M\) that depends only on \(c\), \(a\) and \(h\) such that if (6.30) is satisfied with \(M < M\) then \(g_k = 1\) for all but finitely many \(k\) in \(K\),

(ii) if (6.31) or (6.32) is satisfied then \(g_k = 1\) for all but finitely many \(k\) in \(K\),

(iii) if (6.33) is satisfied, then \(r_{k+1} = o(||r_k||)\) for \(k\) in \(K\).

**Proof:** Since \((y_k)\) converges, Proposition 6.2 shows that \((y_k)\) is bounded and by (6.19), \(g_k\) is modified finitely often. So we can suppose that \(g_k = 1\) for all \(k\). By Taylor's theorem, we expand \(f(y_k + t_k + r_{k+1})\) at the second order in \(t_k\) and the first order in \(r_{k+1}\). First, note that because \((y_k)\) converges to \(x_\infty\) and \((G^{-1}_k)\) is bounded, \((x_k)\) converges to \(x_\infty\). Then, we have
\[
o(x_{k+1}) = o(y_k) + \frac{1}{2}c''(x_\infty) \cdot t_k^2 + o(||t_k||^2). \tag{6.35}
\]
We also have
\[
f(y_k + e_k) = f(y_k) + f'(y_k) \cdot t_k + c(x_k + e_k)^T \lambda(y_k) + \frac{1}{2} f''(x_\infty) \cdot t_k^2 + o(||t_k||^2) + o(||r_{k+1}||)
\]
and using the estimate (6.35), we get
\[
f(y_k + e_k) = f(y_k) + f'(y_k) \cdot t_k + c(y_k)^T \lambda(y_k)
+ \frac{1}{2} c''(y_k + e_k) \cdot t_k^2 + o(||t_k||^2) + o(||r_{k+1}||). \tag{6.36}
\]
On the other hand, expanding \( c(y_k + e_k) \) about \( y_k \) we obtain
\[
c(y_k + e_k) = c(y_k) - c(x_{k+1}) + \frac{1}{2} c''(y_k) \cdot e_k^2 + o(\|e_k\|^2) + o(\|r_{k+1}\|)
\]
and using again (6.35), we get
\[
c(y_k + e_k) = o(\|e_k\|^2) + o(\|r_{k+1}\|) \tag{6.37}
\]
Let us define
\[
\Delta_k := f'(y_k) \cdot t_k - (p - \|\lambda(y_k)\|_\infty) \|c(y_k)\|_\infty t_k,
\]
which is negative by (6.28) and (6.17). Finally, (6.36) and (6.37) give
\[
\Theta_p (y_k + e_k) \leq \Theta_p (y_k) + \Delta_k + \frac{1}{2} t_k^* L y_k t_k + o(\|e_k\|^2) + o(\|r_{k+1}\|).
\]
But \( t_k = Z_k y_k + o(\|e_k\|) \) and the boundedness of \( (G_k) \) allows us to write \( g(y_k) = -G_k y_k + o(\|e_k\|) \). Therefore, using \( f'(y_k) \cdot t_k = -g(y_k) \cdot t_k = -G_k Z_k y_k t_k + o(\|e_k\|^2) \), we obtain
\[
\Theta_p (y_k + e_k) - \Theta_p (y_k) = \Delta_k + \frac{1}{2} Z_k^* G_k Z_k y_k t_k + o(\|e_k\|^2) + o(\|r_{k+1}\|).
\]
Using this inequality we now prove the theorem. Suppose that the step-size \( \epsilon_k \) is different from 1 for infinitely many \( k \) in a subsequence \( K \), say for \( k \in K' \subset K \). Then, according to Armijo's rule (6.4)-(6.6), the left hand side of (6.38) is positive and we have for \( k \in K' \):
\[
-\Delta_k \equiv -\frac{1}{1 - 2x} Z_k^* (G_k - G_0) Z_k y_k + o(\|e_k\|^2) + o(\|r_{k+1}\|).
\]
Using the inequality \( C_1 \|e_k\| \equiv \|g(y_k)\| \) (where \( C_1 \) is a positive constant that depends only on \( c \) and \( h_1 \), property (6.28), inequality (6.17), the definition (4.10) of \( r_{k+1} \) and (6.35), we can obtain a lower bound for the left hand side:
\[
\lambda C_1^2 \|e_k\|^2 + p \|c(y_k)\| \leq \frac{1}{1 - 2x} Z_k^* (G_k - G_0) Z_k y_k + o(\|e_k\|^2) + o(\|r_{k+1}\|).
\]
Now, if one of the properties (6.30) with \( M = \tilde{M} := (1 - 2x) \lambda C_1^2 \|Z_e\|^2 \) or (6.31) or (6.32) is verified for \( k \in K' \), this inequality leads to
\[
\|e_k\|^2 + \|c(y_k)\| = o(\|e_k\|^2) + o(\|r_{k+1}\|),
\]
for \( k \in K' \), which shows that \( K' \) cannot be infinite. This proves statements (i) and (ii) of the theorem.

It remains to prove (iii). With (6.33), inequality (6.39) is valid for \( k \in K \). And as \( t_k = o(\|e_k\|) \), it implies for \( k \in K \):
\[
\|e_k\|^2 + \|c(y_k)\| = o(\|e_k\|^2) + o(\|r_{k+1}\|).
\]
from which we deduce

$$\|s_k\|^p + \|c(y_k)\|_2 = o(\|r_k\| \|s_k\|).$$

Then, this estimate, the definition of $r_{k+1}$ and (6.35) give

$$r_{k+1} = o(\|r_k\| \|s_k\|) + o(\|r_k\|^p).$$

Therefore, using $t_k = o(\|r_k\|)$, we get the estimate in (iii). 

7. Conclusion

In this paper, we have studied the local and global convergence of a variable metric algorithm for equality constrained optimization in which the order of the updated matrices is $n - m$. This reduced method can be seen as making a link between GRG-like methods which are feasible methods ($c(x_k) = 0$ for all $k$) with reduced matrices (of order $n - m$) and the SQP method, which is an unfeasible method with full matrices (of order $n$). The studied algorithm is indeed an unfeasible method with reduced matrices. The algorithm inherits also the good properties of both methods (reduced metrics, superlinear convergence and unfeasibility) and shows, in particular, that locally only one restoration step is necessary to obtain the superlinear convergence of GRG-like methods when the reduced matrices are correctly approximated.

The global convergence is obtained by Han's technique to globalize the SQP method. The $l_1$ penalty function is used as a merit function and is decreased along an arc-shaped search path. Conditions for the asymptotic admissibility of the unit step-size are given that turn out to be satisfied in practice.

An important facet of the method has not been tackled here and is reported elsewhere (Gilbert [25]). This concerns the update of the reduced matrices $G_k$. This one is based on a secant equation using the change in the reduced gradient $g$. The fact that the gradient of $g$ at $x_k$ (see (4.2)) is not equal to $G_k$ (and cannot be equal because $\nabla g(x_k)$ is an $(n - m) \times n$ matrix while $G_k$ is of order $n - m$) leads to an alternative. Either the reduced gradient is evaluated twice per iteration, at $y_k$ and $x_{k+1}$, or it is evaluated only once per iteration, at $y_k$. In the first case, the change $g(x_{k+1}) - g(y_k)$ is used in the secant equation and the matrices $G_k$ are updated at each iteration but with the inconvenience of having to linearize the constraints twice per iteration: see Coleman and Conn [13] and Gilbert [25]. In the second case, the change $g(y_{k+1}) - g(y_k)$ is used in the secant equation but usually the matrices $G_k$ can no longer be updated at each iteration. An update criterion has to be introduced in order to decide when an update is appropriate. Despite this, the superlinear convergence can be achieved either in a local framework (see [34] for algorithm (1.12)–(1.14)) or in a global framework (see [26]) for Algorithm RQN of Section 8).
Acknowledgements: This work was supported in part during 1984–1985 by the Centre d’Études Nucléaires, Fontenay-aux-Roses (France) under a grant from the Commission of the European Community and in part by the Institut National de Recherche en Informatique et en Automatique (INRIA), Le Chesnay (France). I would like to thank those organizations and to express my gratitude to J. F. Bonnans for helpful discussions and suggestions and for his friendly and constant advice.

References


Received November 1987, revised October 1988

J. C. Gilbert
INRIA
Domaine de Voluceau, Rocquencourt
F - 78153 - Le Chesnay Cedex
France

"Tous droits de propriété intellectuelle réservés. Reproduction, représentation interdite sans autorisation (droits de la propriété intellectuelle)."