

ON THE REALIZATION OF THE WOLFE CONDITIONS IN REDUCED QUASI-NEWTON METHODS FOR EQUALITY CONSTRAINED OPTIMIZATION *

JEAN CHARLES GILBERT[†]

Abstract. This paper describes a reduced quasi-Newton method for solving equality constrained optimization problems. A major difficulty encountered by this type of algorithm is the design of a consistent technique for maintaining the positive definiteness of the matrices approximating the reduced Hessian of the Lagrangian. A new approach is proposed in this paper. The idea is to search for the next iterate along a piecewise linear path. The path is designed so that some generalized Wolfe conditions can be satisfied. These conditions allow the algorithm to sustain the positive definiteness of the matrices from iteration to iteration by a mechanism that has turned out to be efficient in unconstrained optimization.

Key words. constrained optimization, exact penalty function, global convergence, piecewise line-search, reduced quasi-Newton, successive quadratic programming, Wolfe's conditions

AMS subject classifications. Primary, 49M37; Secondary, 65K05, 90C30

PII. S1052623493259604

1. Introduction. In unconstrained optimization, when a function $x \in \mathbb{R}^n \mapsto \xi(x) \in \mathbb{R}$ is minimized using descent direction methods, there is a nice combination of a line-search technique attributed to Wolfe [43, 44] and some quasi-Newton methods. On the one hand, if d_k is a descent direction of ξ at the current iterate x_k (i.e., $\nabla \xi(x_k)^\top d_k < 0$), the Wolfe line-search consists in determining a step-size $\alpha_k > 0$ along d_k such that the next iterate $x_{k+1} = x_k + \alpha_k d_k$ satisfies

$$(1.1) \quad \xi(x_{k+1}) \leq \xi(x_k) + \omega_1 \alpha_k \nabla \xi(x_k)^\top d_k,$$

$$(1.2) \quad \nabla \xi(x_{k+1})^\top d_k \geq \omega_2 \nabla \xi(x_k)^\top d_k,$$

where $0 < \omega_1 < \omega_2 < 1$ are constants (independent of k). These conditions contribute to the convergence of descent direction methods. On the other hand, in quasi-Newton methods the descent direction has the form $d_k = -B_k^{-1} \nabla \xi(x_k)$, where B_k is an updated symmetric matrix approximating the Hessian of ξ . It is interesting to maintain this matrix positive definite, in particular because d_k is then a descent direction. With most update formulas, the new matrix B_{k+1} satisfies the so-called quasi-Newton equation

$$\gamma_k = B_{k+1} \delta_k,$$

where $\gamma_k = \nabla \xi(x_{k+1}) - \nabla \xi(x_k)$ is the change in the gradient of ξ and $\delta_k = \alpha_k d_k$ is the step. Of course, if B_{k+1} is positive definite, the quasi-Newton equation implies that

$$(1.3) \quad \gamma_k^\top \delta_k > 0.$$

*Received by the editors December 9, 1993; accepted for publication (in revised form) February 12, 1996.

<http://www.siam.org/journals/siopt/7-3/25960.html>

[†]INRIA, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France (Jean-Charles.Gilbert@inria.fr).

Therefore, this curvature condition (1.3) has to be satisfied if one expects B_{k+1} to be positive definite. For some quasi-Newton formulas (for instance the BFGS formula, see below), which update B_k using γ_k and δ_k , this inequality is also sufficient to have B_{k+1} positive definite (provided B_k is already positive definite). The remarkable fact is that the second Wolfe condition above guarantees this inequality. Hence, using the Wolfe line-search and the BFGS formula, e.g., ensures that all the search directions have the descent property.

For various reasons (see, for example, Powell [33]), it is not straightforward to extend the above scheme to a minimization problem with constraints on the variables. Such an extension is desirable, however, because numerical experience has shown that the approach is very successful in unconstrained minimization, even when the number of variables is large (see Liu and Nocedal [26] and Gilbert and Lemaréchal [20]).

In this paper, we study in more detail the matter for the equality constrained minimization problem

$$(1.4) \quad \begin{cases} \min f(x) \\ c(x) = 0, \quad x \in \Omega, \end{cases}$$

where $\Omega \subset \mathbb{R}^n$ is an open set and $f : \Omega \rightarrow \mathbb{R}$ and $c : \Omega \rightarrow \mathbb{R}^m$ ($m < n$) are smooth functions.

Since the set Ω is supposed to be open, it cannot be used to define general constraints. It is the set where f and c have nice properties. For example, we always suppose that the $m \times n$ Jacobian matrix of the constraints

$$A(x) = \nabla c(x)^\top$$

is surjective (i.e., has full row rank) for any $x \in \Omega$. We also suppose that this matrix has a right inverse $A^-(x)$ depending smoothly on x :

$$A(x)A^-(x) = I \quad \forall x \in \Omega.$$

Besides, we assume that for all $x \in \Omega$, there is a basis $Z^-(x)$ of the null space $N(A(x))$ of $A(x)$, which means that $Z^-(x)$ is an injective (or full column rank) $n \times (n - m)$ matrix satisfying

$$A(x)Z^-(x) = 0 \quad \forall x \in \Omega.$$

We also suppose that the map $x \mapsto Z^-(x)$ is smooth. These assumptions on Z^- are not restrictive if Ω may differ from \mathbb{R}^n but can rarely be satisfied when $\Omega = \mathbb{R}^n$ (for example, the assumptions on Z^- cannot be satisfied on even-dimensional spheres). Observe that for $A^-(x)$ and $Z^-(x)$ defined as above, there exists a unique $(n - m) \times n$ matrix $Z(x)$ such that

$$(1.5) \quad Z(x)Z^-(x) = I \quad \text{and} \quad Z(x)A^-(x) = 0$$

in $\mathbb{R}^{(n-m) \times (n-m)}$ and $\mathbb{R}^{(n-m) \times m}$, respectively (see Gabay [14], for example).

The Lagrangian function of problem (1.4) is the function $\ell : (x, \lambda) \in \Omega \times \mathbb{R}^m \rightarrow \mathbb{R}$, defined by

$$\ell(x, \lambda) = f(x) + \lambda^\top c(x).$$

Its Hessian with respect to x is denoted by $L(x, \lambda) = \nabla_{xx}^2 \ell(x, \lambda)$. The reduced Hessian of the Lagrangian is the order $n - m$ matrix $Z^-(x)^\top L(x, \lambda) Z^-(x)$. We denote by x_*

a solution of (1.4) and by λ_* its associated multiplier and denote $L_* = L(x_*, \lambda_*)$ and $B_* = Z^-(x_*)^\top L_* Z^-(x_*)$.

Our study is done in the framework of those reduced quasi-Newton methods that, near a solution x_* , generate the sequence of iterates $\{x_k\} \subset \Omega$ approximating x_* by

$$(1.6) \quad x_{k+1} = x_k + d_k,$$

where d_k is the solution of the quadratic program

$$(1.7) \quad \begin{cases} \min \nabla f(x_k)^\top d + \frac{1}{2} d^\top Z_k^\top B_k Z_k d \\ c_k + A_k d = 0. \end{cases}$$

In (1.7), $c_k = c(x_k)$, $A_k = A(x_k)$, $Z_k = Z(x_k)$, and the order $n - m$ matrix B_k is an approximation of the reduced Hessian of the Lagrangian (see Murray and Wright [30] and Gabay [15]). Since $Z_k^\top B_k Z_k$ approximates only a part of the Hessian of the Lagrangian, the method differs from the well-known sequential quadratic programming (SQP) algorithm (see Wilson [42], Han [22], and Pshenichnyi and Danilin [10]) in which an approximation of the full Hessian of the Lagrangian is updated. These reduced quasi-Newton algorithms have a lower speed of convergence than SQP methods, but they may be used for larger problems because they need to update smaller matrices.

Any direction d satisfying the linear constraints in (1.7) has the form $d = Z_k^- h - A_k^- c_k$, where $Z_k^- = Z^-(x_k)$, $A_k^- = A^-(x_k)$, and h is some vector in \mathbb{R}^{n-m} . Substituting this in the objective function of (1.7), assuming that B_k is positive definite, and minimizing in h , we obtain as a solution of (1.7)

$$d_k = t_k + r_k = -Z_k^- B_k^{-1} g_k - A_k^- c_k,$$

where $g_k = g(x_k) = Z_k^{-\top} \nabla f(x_k) \in \mathbb{R}^{n-m}$ is called the *reduced gradient* of f at x_k , $t_k = -Z_k^- B_k^{-1} g_k$ is called the *tangential* or *longitudinal* component of the step, and $r_k = -A_k^- c_k$ is called the *restoration* or *transversal* component of the step.

One of the main concerns of this paper is to develop a technique that maintains the positive definiteness of the matrices B_k . This property is interesting because it makes the direction t_k a descent direction of most merit functions used to globalize the local method (1.6)–(1.7). It is also natural since this matrix approximates the reduced Hessian of the Lagrangian, which is positive semidefinite at the solution. To obtain this property, our approach mimics what is done in unconstrained optimization, as was recalled in the beginning of this introduction. First, we use an update formula allowing the positive definiteness to be transmitted from one matrix to the next one. A typical example is the BFGS formula (see [13, 21, 11])

$$(1.8) \quad B_{k+1} = B_k - \frac{B_k \delta_k \delta_k^\top B_k}{\delta_k^\top B_k \delta_k} + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k}.$$

This formula requires the use of two vectors γ_k and δ_k in \mathbb{R}^{n-m} , which will be specified in a moment. The important point is that the positive definiteness is sustained from B_k to B_{k+1} if the vectors γ_k and δ_k satisfy the following condition:

$$(1.9) \quad \gamma_k^\top \delta_k > 0.$$

Next, we propose a “piecewise line-search” (PLS) technique that finds a point satisfying *generalized* Wolfe conditions, which reduce to conditions (1.1)–(1.2) when there

are no constraints. These conditions imply inequality (1.9) for appropriate vectors γ_k and δ_k and, therefore, also the positive definiteness of the matrices updated by using these vectors.

The local analysis of algorithm (1.6)–(1.7) shows that it is convenient to take for γ_k the change in the reduced gradient and for δ_k the reduced displacement

$$(1.10) \quad \gamma_k = g_{k+1} - g_k \quad \text{and} \quad \delta_k = \alpha_k Z_k t_k.$$

Other choices are sometimes proposed: see, for instance, Coleman and Conn [7] and Nocedal and Overton [31]. All of them are asymptotically equivalent to the above choice, which is preferred for its geometrical interpretation (see section 3) and its simplicity. In these formulas appears a step-size $\alpha_k > 0$ (see section 3) because the matrices B_k are also updated far from the solution where the algorithm differs from (1.6)–(1.7). Note, however, that x_{k+1} is obtained in a more sophisticated way than a simple move along the tangent direction t_k . This is necessary because such a move does not usually yield (1.9) (see [17]).

Condition (1.9) holds if the search algorithm determines x_{k+1} such that

$$(1.11) \quad g_{k+1}^\top Z_k t_k \geq \omega_2 g_k^\top Z_k t_k,$$

where $0 < \omega_2 < 1$. This is actually what the search algorithm realizes. Now, this algorithm has another role to play, which is to contribute to the global convergence of the method. This is achieved by sufficiently decreasing some merit function, which we choose to be

$$(1.12) \quad \Theta_\sigma(x) = f(x) + \sigma \|c(x)\|,$$

where σ is positive number and $\|\cdot\|$ denotes a norm in \mathbb{R}^m . This penalty function is exact when σ is sufficiently large (see, for example, Han and Mangasarian [23]). The decrease in Θ_σ is typically forced by requiring that

$$(1.13) \quad \Theta_\sigma(x_{k+1}) \leq \Theta_\sigma(x_k) + \omega_1 \nu_k(\alpha_k),$$

where $0 < \omega_1 < 1$ and $\nu_k(\alpha)$ is negative for positive α . Note that we do not need $\omega_1 < \omega_2$ in the PLS algorithm.

The difficulty of realizing both (1.11) and (1.13) simultaneously comes from the fact that, unlike what happens for unconstrained problems, the left-hand side in (1.11) is not the directional derivative of Θ_σ at x_{k+1} along commonly used search directions such as t_k or d_k . We shall see that it is the directional derivative $\Theta'_\sigma(x_{k+1}; Z_{k+1}^- Z_k t_k)$. This suggests making a reorientation of the search direction when (1.11) does not hold by using the new basis Z_{k+1}^- , while keeping the same reduced tangent direction $Z_k t_k$. This is the idea underlying the search algorithm proposed in [17], where the search path has only longitudinal components, i.e., components in the range space of the matrices $Z^-(x_k^i)$, where x_k^i ($i = 0, \dots, i_k - 1$) are intermediate points. Here we show how to implement this idea for paths also having transversal components, i.e., components in the range space of $A^-(x_k^i)$. This improves the algorithm, since asymptotically the constraints need no longer be linearized twice per iteration of the overall algorithm.

The analysis results in a quite simple search algorithm, which can be described as follows. At each inner iteration i of the PLS algorithm, condition (1.13) is first realized and, next, condition (1.11) is tested. If the latter holds, the PLS algorithm

terminates with a suitable point. Otherwise, a new inner search direction is defined, using the *same* matrix B_k and the *same* reduced gradient g_k as for the previous inner direction. A new inner iteration is then started.

Other authors have proposed techniques for maintaining the positive definiteness of the generated matrices for constrained minimization problems, but none uses the search algorithm to achieve this goal. These papers also deal with the SQP method, in which approximations of the full Hessian of the Lagrangian are generated. In this case γ_k is usually γ_k^ℓ , the change in the gradient of the Lagrangian, and δ_k is the step. The first proposal, due to Powell [33], was to take for γ_k a convex combination of γ_k^ℓ and $B_k\delta_k$ such that (1.9) holds. According to Powell [35], the method may lead to ill conditioning when the problem is difficult to solve. We have also observed the failure of this technique on some academic problems (see Armand and Gilbert [1]). Due to its great simplicity, however, it is the most widely implemented technique. Another promising idea, proposed by Han [22] and Tapia [40] and subsequently explored by Tapia [41] and Byrd, Tapia, and Zhang [5], is to generate approximations of the Hessian of the augmented Lagrangian, which is positive definite at the solution when the penalty parameter is sufficiently large. The difficulty in choosing the penalty parameter has always been the stumbling block of this approach, and we believe that more research is needed to improve the method satisfactorily. Finally, Fenyves [12] and Coleman and Fenyves [8] separately update approximations of $Z^-(x_*)^\top L_* Z^-(x_*)$ and $A^-(x_*)^\top L_* Z^-(x_*)$, maintaining positive definite the approximations of the former matrix.

We conclude this introduction with a few remarks. First, our PLS algorithm also can be used for the reduced quasi-Newton method of Coleman and Conn [6] with minor modifications (see [18]), while its use for the SQP method has been investigated by Armand and Gilbert [1]. An important point to mention is that when the reduced Hessian of the Lagrangian is computed exactly and used in place of B_k in (1.7), there is no need to use the PLS algorithm. In this case, a simple Armijo [2] backtracking along d_k is preferable, since it is less expensive and easier to implement than the PLS algorithm.

The paper is organized as follows. In section 2, we make the hypotheses and notation more precise. In section 3, the search path is introduced and its meaning is discussed. Also, conditions for obtaining finite termination of the search algorithm are given. Section 4 contains a global convergence result and, finally, some numerical experiments are reported in section 5.

2. Hypotheses and notation. We suppose that the function c defining the constraints in (1.4) is a submersion on Ω , which means that its Jacobian matrix $A(x)$ is surjective for all x in Ω . Then, for any $x \in \Omega$, the set

$$\mathcal{M}_x = \{y \in \Omega : c(y) = c(x)\}$$

forms a smooth submanifold of \mathbb{R}^n , having dimension $n - m$ (for the geometrical tools, we refer the reader to Spivak [38], Boothby [3], or Conlon [9], for example).

We quote the fact that the columns of the basis $Z^-(x)$ introduced in section 1 span the space tangent to \mathcal{M}_x at x and that the columns of the right inverse $A^-(x)$ span a space complementary to this tangent space. The matrix $Z(x)$ defined by (1.5) is also characterized by the useful identity

$$A^-(x)A(x) + Z^-(x)Z(x) = I,$$

which allows us to decompose a direction d of \mathbb{R}^n in its *transversal* component $A^-(x)A(x)d$ and its *longitudinal* component $Z^-(x)Z(x)d$. Also,

$$Z(x) = \begin{pmatrix} I & 0 \\ Z^-(x) & A^-(x) \end{pmatrix}^{-1},$$

so that the map $x \mapsto Z(x)$ inherits the smoothness of Z^- and A^- .

We assume that there is a pair $(x_*, \lambda_*) \in \Omega \times \mathbb{R}^m$ satisfying the sufficient second order conditions of optimality, i.e.,

$$\begin{cases} c(x_*) = 0, \\ \nabla f(x_*) + A(x_*)^\top \lambda_* = 0, \end{cases}$$

and $h^\top L_* h > 0$ for all nonzero $h \in N(A(x_*))$. By these assumptions, the reduced Hessian of the Lagrangian at the solution $B_* = Z^-(x_*)^\top L_* Z^-(x_*)$ is positive definite. We also introduce

$$(2.1) \quad \lambda(x) = -A^-(x)^\top \nabla f(x),$$

which estimates the Lagrange multiplier at the solution: $\lambda(x_*) = \lambda_*$.

We recall that we use the penalty function Θ_{σ_k} defined in (1.12) to globalize the local method (1.6)–(1.7). The *penalty parameter* σ_k depends on the iteration index k and is updated to satisfy at each iteration

$$(2.2) \quad \sigma_k \geq \|\lambda_k\|_D + \bar{\sigma},$$

where $\lambda_k = \lambda(x_k)$ and $\bar{\sigma}$ is a fixed positive number. We have denoted by $\|\cdot\|_D$ the dual norm of the norm $\|\cdot\|$ used in (1.12). It is defined by

$$\|v\|_D = \sup_{\|u\|=1} u^\top v.$$

The manifolds on which the reduced gradient g is constant are denoted by

$$\mathcal{N}_x = \{y \in \Omega : g(y) = g(x)\}.$$

These sets are indeed manifolds if Ω is sufficiently “small,” because g is a submersion in a neighborhood of x_* . To see this, observe that $g'(x_*) = Z_*^{-\top} L_*$ (see Stoer [39] or Nocedal and Overton [31]) and that $Z_*^{-\top} L_*$ is surjective.

We denote by $\xi'(u; v)$ the directional derivative of a function ξ at u along the direction v . In particular, if ξ is a function of a real variable α , $\xi'(\alpha; 1)$ denotes its right derivative. We quote the fact that if \mathcal{C} is a convex continuous function and if ξ has directional derivatives, then $\mathcal{C} \circ \xi$ also has directional derivatives (“ \circ ” denotes composition). This can be seen by using the local Lipschitz continuity of \mathcal{C} , implied by its continuity (see Theorem 10.4 in [36] or Theorem IV.3.1.2 in [24]). As a result, when the constraint function c is smooth, Θ_σ defined in (1.12) has directional derivatives.

The following identity will be used several times. If f and c are smooth and $h \in \mathbb{R}^{n-m}$, we have for Θ_σ defined by (1.12)

$$(2.3) \quad \Theta'_\sigma \left(x; Z^-(x)h - A^-(x)c(x) \right) = g(x)^\top h + \lambda(x)^\top c(x) - \sigma \|c(x)\|.$$

Indeed, function f in Θ_σ gives the first two terms in the right-hand side of (2.3) (use the definition of $g(x)$ and (2.1)). Next, taking the notation $\eta(x) = \|x\|$ and knowing

that $(\eta \circ c)'(x; v) = \eta'(c(x); A(x)v)$, the directional derivative of the second term in Θ_σ is given by

$$\begin{aligned} \sigma(\eta \circ c)'(x; Z^-(x)h - A^-(x)c(x)) &= \sigma\eta'(c(x); -c(x)) \\ &= \sigma \lim_{t \rightarrow 0^+} \frac{1}{t} (\|c(x) - tc(x)\| - \|c(x)\|) \\ &= -\sigma\|c(x)\|. \end{aligned}$$

3. The search algorithm. In unconstrained optimization, the path $p_k : \alpha \in \mathbb{R}_+ \mapsto p_k(\alpha)$ starting at the current iterate $p_k(0) = x_k \in \Omega$ and along which a step-size is taken is most commonly a straight line, which can be determined before the search begins. When constraints are present, a search along a line is no longer possible if one aims at satisfying the *reduced Wolfe conditions*

$$(3.1) \quad \Theta_{\sigma_k}(p_k(\alpha)) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \nu_k(\alpha),$$

$$(3.2) \quad g(p_k(\alpha))^\top Z_k t_k \geq \omega_2 g_k^\top Z_k t_k$$

for some $\alpha > 0$. In (3.1) and (3.2), the constants ω_1 and ω_2 are chosen in $(0, 1)$, and $\alpha \mapsto \nu_k(\alpha)$ is a function forcing the decrease of Θ_{σ_k} by the properties

$$(3.3) \quad \begin{cases} \nu_k(0) = 0 \\ \Theta'_{\sigma_k}(x_k; p'_k(0; 1)) \leq \nu'_k(0; 1) < 0. \end{cases}$$

These properties and $\omega_1 < 1$ make it possible to realize (3.1) for small positive α . We have assumed that p_k is a descent path for Θ_{σ_k} , i.e., $\Theta'_{\sigma_k}(x_k; p'_k(0; 1)) < 0$.

In our proposal, the description of the search path is not as easy as in unconstrained optimization, because it depends on some intermediate step-sizes. From the point of view taken here, a reorientation of the search path is indeed necessary at some intermediate step-sizes α_k^i , $i = 1, \dots, i_k - 1$. Furthermore, condition (3.1) also depends on the step-sizes α_k^i through the function ν_k , which cannot be given before the search is completed. For these reasons, we have to specify simultaneously the function ν_k and the way the search path is designed.

The algorithm we discuss has some similarities with the one given in [17], but here the path has at once a longitudinal and a transversal component. More basically, one can see it as an extension of the method proposed by Fletcher [13] and Lemaréchal [25] for finding a Wolfe point in unconstrained optimization. With the option $\rho_k^i = 1$ below, the algorithm is related to the search technique of Moré and Sorensen [28] for realizing the strong Wolfe conditions for unconstrained problems.

3.1. Guiding paths. Before giving a precise description of the search algorithm, we would like to show by some observations why trying to realize conditions (3.1) and (3.2) simultaneously can succeed. On the way, we exhibit conditions under which our search technique should be numerically efficient.

First, let us introduce a path $\alpha \mapsto \bar{p}_k(\alpha)$ as a solution of the following differential equation:

$$(3.4) \quad \begin{cases} \bar{p}'_k(\alpha) = Z^-(\bar{p}_k(\alpha))Z_k t_k, \\ \bar{p}_k(0) = x_k. \end{cases}$$

This trajectory belongs to the manifold $\mathcal{M}_k = \mathcal{M}_{x_k}$ because multiplying the first equation in (3.4) by $A(\bar{p}_k(\alpha))$ gives $(c \circ \bar{p}_k)'(\alpha) = 0$, which means that c remains

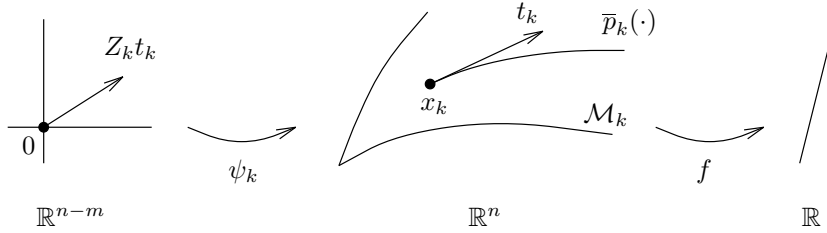


FIG. 3.1. An interpretation of the longitudinal guiding path.

constant along the path. As quoted in [17], if this path is defined for sufficiently large α and if f is bounded from below on \mathcal{M}_k , there exists a step-size $\bar{\alpha}_k$ such that (here $\omega_1 < \omega_2$ is necessary)

$$(3.5) \quad \Theta_{\sigma_k}(\bar{p}_k(\bar{\alpha}_k)) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \bar{\alpha}_k g_k^\top Z_k t_k,$$

$$(3.6) \quad g(\bar{p}_k(\bar{\alpha}_k))^\top Z_k t_k \geq \omega_2 g_k^\top Z_k t_k.$$

This can be seen by considering the standard Wolfe [43, 44] conditions (recalled in the introduction) on the function

$$\alpha \mapsto (\Theta_{\sigma_k} \circ \bar{p}_k)(\alpha) = (f \circ \bar{p}_k)(\alpha) + \sigma_k \|c_k\|.$$

Indeed, using (2.3), the derivative of this map at $\bar{\alpha}_k$ is $g(\bar{p}_k(\bar{\alpha}_k))^\top Z_k t_k$, the left-hand side of (3.6). Note that condition (3.5) has the form (3.1) with a linear function ν_k .

Locally, the search along \bar{p}_k also has the following geometrical interpretation, illustrated in Figure 3.1. Suppose that there exists a parametrization $\psi_k : U \subset \mathbb{R}^{n-m} \rightarrow \mathcal{M}_k \subset \mathbb{R}^n$ of \mathcal{M}_k around x_k such that $0 \in U$, $\psi_k(0) = x_k$, and

$$(3.7) \quad \psi'_k(u) = Z^-(\psi_k(u)) \quad \forall u \in U.$$

The existence of such parametrization depends on the choice of the tangent basis map Z^- (see [19]). Then, it is easy to see that

$$\bar{p}_k(\alpha) = \psi_k(\alpha Z_k t_k).$$

Indeed, denoting $q_k(\alpha) = \psi_k(\alpha Z_k t_k)$, we have $q'_k(\alpha) = Z^-(q_k(\alpha)) Z_k t_k$, by (3.7), and $q_k(0) = x_k$; hence, q_k satisfies the differential equation (3.4), which implies that $q_k = \bar{p}_k$. As a result, $(f \circ \psi_k)(\alpha Z_k t_k) = (\Theta_{\sigma_k} \circ \bar{p}_k)(\alpha) - \sigma_k \|c_k\|$ and $\nabla(f \circ \psi_k)(\alpha Z_k t_k) = g(\bar{p}_k(\alpha))$, so that the search to realize (3.5)–(3.6) can now be seen as a standard Wolfe search on the function $f \circ \psi_k$ starting at $0 \in \mathbb{R}^{n-m}$ along the reduced direction $Z_k t_k$. From this interpretation, we define the *reduced (longitudinal) displacement* from x_k to $\bar{p}_k(\bar{\alpha}_k)$ as the vector $\bar{\delta}_k = \bar{\alpha}_k Z_k t_k$.

The path $\alpha \mapsto \bar{p}_k(\alpha)$ shows that there is at least one way of generalizing the Wolfe conditions to equality constrained problems. We call this path the *longitudinal guiding path*; we say longitudinal because its image

$$\bar{\mathcal{P}}_k = \{\bar{p}_k(\alpha) : \alpha \geq 0 \text{ and } \bar{p}_k(\alpha) \text{ exists}\}$$

lies in \mathcal{M}_k . This trajectory can be used as a guide for designing a search path having points satisfying (3.5)–(3.6) but easier to compute than $\bar{p}_k(\cdot)$; see [17].

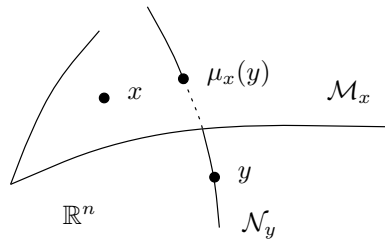


FIG. 3.2. The map μ_x .

In this paper, we follow the same strategy and introduce a smooth guiding path having longitudinal and transversal components, i.e., neither c nor g is constant along the path. Later, a discretization will be introduced. We proceed step by step and begin by some definitions.

DEFINITION 3.1. Let F be the function

$$F : \Omega \rightarrow \mathbb{R}^n : x \mapsto \begin{pmatrix} c(x) \\ g(x) \end{pmatrix}.$$

DEFINITION 3.2. Let us also introduce an open subset Ω_0 of Ω such that: (i) $x_* \in \Omega_0$; (ii) $F'(x)$ is nonsingular when $x \in \Omega_0$; (iii) $F(\Omega_0)$ has the form $U_0 \times V_0$, where U_0 and V_0 are open sets in \mathbb{R}^m and \mathbb{R}^{n-m} , respectively; (iv) $F : \Omega_0 \rightarrow U_0 \times V_0$ is a diffeomorphism.

Note that such open subset Ω_0 always exists when $B_* = Z^-(x_*)^\top L_* Z^-(x_*)$ is nonsingular, which we assume. Indeed, in this case, recalling that $g'(x_*) = Z_*^{-\top} L_*$, we see that $F'(x_*)$ is nonsingular, so conditions (i)–(iv) are satisfied for some (possibly large) neighborhood Ω_0 of x_* .

DEFINITION 3.3. For a point x fixed in Ω_0 , we introduce the map

$$\mu_x : \Omega_0 \rightarrow \mathcal{M}_x \cap \Omega_0,$$

defined in the following way. For $y \in \Omega_0$, $\mu_x(y) \in \mathcal{M}_x \cap \Omega_0$ is defined as the unique point in $\mathcal{M}_x \cap \mathcal{N}_y \cap \Omega_0$ (\mathcal{N}_y is the reduced gradient manifold containing y); see Figure 3.2.

To see that the set $\mathcal{M}_x \cap \mathcal{N}_y \cap \Omega_0$ is formed of just one point, note that $x \in \Omega_0$ and $y \in \Omega_0$ imply that $(c(x), g(y)) \in U_0 \times V_0 = F(\Omega_0)$. As F is a diffeomorphism on Ω_0 , $\mathcal{M}_x \cap \mathcal{N}_y \cap \Omega_0 = F^{-1}((c(x), g(y))) \cap \Omega_0$ is a singleton. As we see, μ_x maps a point $y \in \Omega_0$ to a point in $\mathcal{M}_x \cap \Omega_0$ by following the manifold of constant reduced gradient \mathcal{N}_y . The following result will be useful.

PROPOSITION 3.4. Suppose that c and g are of class C^l ($l \geq 1$) on Ω_0 and let $x \in \Omega_0$. Then, $\mu_x : \Omega_0 \rightarrow \mathcal{M}_x \cap \Omega_0$ is of class C^l and, as a function with values in \mathbb{R}^n , its Jacobian matrix at $y \in \Omega_0$ is given by

$$(3.8) \quad \mu'_x(y) = \tilde{Z}^-(z) \left(g'(z) \tilde{Z}^-(z) \right)^{-1} g'(y),$$

where $z = \mu_x(y)$ and $\tilde{Z}^-(z)$ is an arbitrary basis of the space tangent to \mathcal{M}_x at z (one can take $\tilde{Z}^-(z) = Z^-(z)$).

Proof. To show that μ_x is of class C^l , we “read” this map with appropriate C^l -compatible coordinate charts. Let us take $(U, \varphi) = (\Omega_0, F)$ as a chart of Ω_0 at y and

$(V, \psi) = (\mathcal{M}_x \cap \Omega_0, g|_{\mathcal{M}_x \cap \Omega_0})$ as a chart of $\mathcal{M}_x \cap \Omega_0$ at $z = \mu_x(y)$. These coordinate charts are C^l because c and g are C^l . Then, μ_x is read with φ and ψ as

$$\psi \circ \mu_x \circ \varphi^{-1},$$

which is the C^∞ map $\mathbb{R}^n \rightarrow \mathbb{R}^{n-m} : (u_1, \dots, u_n) \mapsto (u_{m+1}, \dots, u_n)$. This shows that μ_x is of class C^l .

Since c is of class C^l , the canonical injection $j : \mathcal{M}_x \cap \Omega_0 \rightarrow \mathbb{R}^n$ is of class C^l , and μ_x with values in \mathbb{R}^n (more precisely, $j \circ \mu_x$) is also of class C^l . Then, we can differentiate the identities

$$c(\mu_x(y)) = c(x) \quad \text{and} \quad g(\mu_x(y)) = g(y)$$

with respect to y . This gives, with $z = \mu_x(y)$,

$$A(z)\mu'_x(y) = 0 \quad \text{and} \quad g'(z)\mu'_x(y) = g'(y).$$

To solve this system in $\mu'_x(y)$, we introduce an arbitrary basis $\tilde{Z}^-(z)$ of the null space of $A(z)$. From the first identity, we see that $\mu'_x(y) = \tilde{Z}^-(z)M$ for some $(n - m) \times n$ matrix M . Then, by the nonsingularity of $F'(z)$ when $z \in \Omega_0$ (see Definition 3.2), $g'(z)\tilde{Z}^-(z)$ is nonsingular and the second identity above leads to (3.8). \square

Let us now go back to our problem of designing a suitable path $\alpha \mapsto \tilde{p}_k(\alpha)$, with longitudinal and transversal components. Suppose we ensure that its image by μ_{x_k} lies in $\overline{\mathcal{P}}_k$; i.e.,

$$\mu_{x_k}(\tilde{p}_k(\alpha)) \in \overline{\mathcal{P}}_k \quad \text{for } \alpha \geq 0.$$

We recall that $\bar{\alpha}_k$ is some step-size such that (3.5)–(3.6) hold. Then, if $\tilde{p}_k(\alpha)$ exists for sufficiently large α , it is reasonable to expect to find some positive $\tilde{\alpha}_k$ such that $g(\tilde{p}_k(\tilde{\alpha}_k)) = g(\bar{p}_k(\bar{\alpha}_k))$ —this assumes that the path \tilde{p}_k does not blow up for a finite longitudinal displacement. Using (3.6), we obtain

$$g(\tilde{p}_k(\tilde{\alpha}_k))^\top Z_k t_k \geq \omega_2 g_k^\top Z_k t_k.$$

This shows that condition (3.2) can be satisfied along a path not belonging to \mathcal{M}_k .

For two reasons, this is not enough, however, to have a satisfactory search. First, the two conditions (3.1) and (3.2) have to be satisfied simultaneously. Second, if we want to minimize approximation errors by updating the matrix with $\tilde{\gamma}_k = g(\tilde{p}_k(\tilde{\alpha}_k)) - g_k = g(\bar{p}_k(\bar{\alpha}_k)) - g_k$ and $\tilde{\delta}_k = \tilde{\alpha}_k Z_k t_k$, we also need to have $\tilde{\alpha}_k = \bar{\alpha}_k$ so that the changes in the reduced gradient along \bar{p}_k and \tilde{p}_k will correspond to the same reduced displacement $\tilde{\delta}_k = \bar{\alpha}_k Z_k t_k$.

This latter condition will be satisfied if we build a path $\alpha \mapsto \tilde{p}_k(\alpha)$ such that

$$(3.9) \quad g(\tilde{p}_k(\alpha)) = g(\bar{p}_k(\alpha))$$

for all α for which $\tilde{p}_k(\alpha)$ and $\bar{p}_k(\alpha)$ exist. In the next proposition, we show that this can be achieved when $\tilde{p}_k(\cdot)$ is defined as a solution of the following differential equation:

$$(3.10) \quad \begin{cases} \tilde{p}'_k(\alpha) = Z^-(\tilde{p}_k(\alpha))Z_k t_k - A^-(\tilde{p}_k(\alpha))c(\tilde{p}_k(\alpha)), \\ \tilde{p}_k(0) = x_k, \end{cases}$$

and the maps $Z^-(\cdot)$ and $A^-(\cdot)$ are chosen such that

$$(3.11) \quad \begin{cases} g'A^- = 0, \\ g'Z^- \text{ is constant on the reduced gradient manifolds.} \end{cases}$$

The first condition in (3.11) requires that the transversal displacements (in the range space of A^-) be in the space tangent to the reduced gradient manifold. The matrix $g'Z^-$ appearing in (3.11) is the matrix from which information is collected by the pair (γ_k, δ_k) in (1.10). At the solution, it is also the reduced Hessian of the Lagrangian. The second condition of (3.11) requires that this matrix be constant along the reduced gradient manifolds.

PROPOSITION 3.5. *Suppose that c and g are continuously differentiable on the set Ω_0 introduced in Definition 3.2, that $x_k \in \Omega_0$, and that the maps Z^- and A^- are such that (3.11) holds on Ω_0 . Consider the paths \bar{p}_k and \tilde{p}_k defined by (3.4) and (3.10), respectively. Then, (3.9) holds as long as both $\bar{p}_k(\alpha)$ and $\tilde{p}_k(\alpha)$ exist in Ω_0 .*

Proof. Let us define $q_k = \mu_{x_k} \circ \tilde{p}_k$, a path in \mathcal{M}_k . This path is well defined as long as \tilde{p}_k exists in Ω_0 . By the definition of μ_{x_k} , $g(q_k(\alpha)) = g(\tilde{p}_k(\alpha))$. Hence, we just have to prove that $q_k = \bar{p}_k$.

Note that since $\tilde{p}_k(\alpha)$ and $q_k(\alpha)$ belong to the same reduced gradient manifold, the second condition in (3.11) gives

$$(3.12) \quad g'(\tilde{p}_k(\alpha))Z^-(\tilde{p}_k(\alpha)) = g'(q_k(\alpha))Z^-(q_k(\alpha)).$$

Now, by Proposition 3.4, we see that q_k is differentiable; by using $\tilde{Z}^- = Z^-$ in (3.8), we have

$$\begin{aligned} q'_k(\alpha) &= \mu'_{x_k}(\tilde{p}_k(\alpha))\tilde{p}'_k(\alpha) \\ &= Z^-(q_k(\alpha))\left(g'(q_k(\alpha))Z^-(q_k(\alpha))\right)^{-1}g'(\tilde{p}_k(\alpha))Z^-(\tilde{p}_k(\alpha))Z_k t_k \\ &= Z^-(q_k(\alpha))Z_k t_k, \end{aligned}$$

where we also used (3.10), the first condition in (3.11), and (3.12). Therefore, q_k satisfies the same differential equation as \bar{p}_k , with the same initial condition x_k at $\alpha = 0$ (see (3.4)). Hence, $q_k = \bar{p}_k$ and the proposition is proved. \square

We call the path defined by \tilde{p}_k , the solution of (3.10), the *bicomponent guiding path*. The actual PLS path introduced in section 3.2 will be a discretization of this one. From Proposition 3.5 and the discussion that precedes it, one can say that *the PLS should be numerically efficient when Z^- and A^- are chosen such that (3.7) holds for some parametrization ψ_k and (3.11) holds.*

If (3.7) can always be realized by choosing suitable tangent bases (see [19]), it is unrealistic to ask the user to realize (3.11), because the computation of g' requires the evaluation of second derivatives, which are not available in the quasi-Newton framework. Also, we shall not assume that (3.11) holds and, therefore, (3.9) may not hold either, even at the first order. Figure 3.3 represents a still rather favorable situation without (3.9); it is favorable because the dashed curve in \mathcal{M}_k and in the reduced space is still rather close to the solid curve. On the other hand, we shall keep the path defined by (3.10). As we shall see below (Proposition 3.6), no matter the realization of (3.9), one can satisfy the reduced Wolfe conditions (3.1)–(3.2) along the path \tilde{p}_k for an appropriate function ν_k . Of course, an update of the matrix without (3.9) may not be safe. We believe, however, that it could be the role of the update

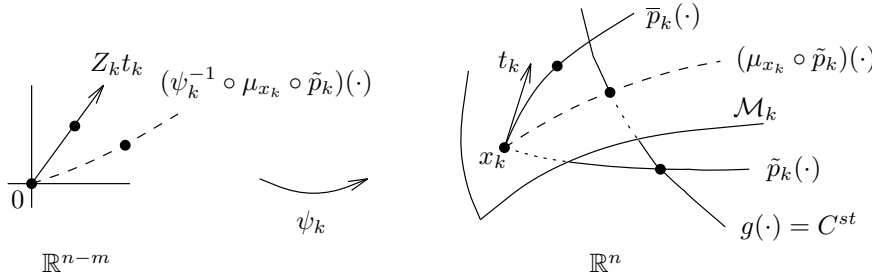


FIG. 3.3. The bicomponent guiding path.

criterion to detect situations where (3.11) is not violated by much. Nevertheless, the update criteria introduced by Nocedal and Overton [31] and Gilbert [16], as well as the one used in the numerical experiments below, are based on a condition different from (3.11).

Let us now show how to realize (3.1)–(3.2) along the path $p_k = \tilde{p}_k$ without condition (3.9). For this, we take for ν_k in (3.1) the function defined by

$$(3.13) \quad \nu_k(\alpha) = \Theta_{\sigma_k}(\tilde{p}_k(\alpha)) - \Theta_{\sigma_k}(x_k),$$

with \tilde{p}_k given by (3.10). Conditions (3.3) are satisfied for this choice, provided that σ_k is sufficiently large. Then (3.1) is equivalent to requiring that

$$(3.14) \quad \Theta_{\sigma_k}(\tilde{p}_k(\alpha)) \leq \Theta_{\sigma_k}(x_k).$$

At this point, function ν_k does not look very useful, since it no longer appears in the descent condition (3.14). But this is only true in the present smooth case. In the discretized version of the search algorithm, it is (3.1) with (3.13) that will be discretized, not (3.14), so that terms coming from the discretization will force the merit function to decrease.

Requiring (3.14) is not very demanding, but it gives the time for (3.2) to be realized before violating (3.1); this is shown in Proposition 3.6 below. Note that the result of this proposition can be obtained without the inequality $\omega_1 < \omega_2$ (ω_1 is not used in the statement of the proposition).

PROPOSITION 3.6. *Suppose that the path $\alpha \mapsto \tilde{p}_k(\alpha)$ defined by (3.10) exists for sufficiently large step-size $\alpha \geq 0$, that Θ_{σ_k} is bounded from below along this path, that $\sigma_k \geq \|\lambda(\tilde{p}_k(\alpha))\|_D$ whenever $\tilde{p}_k(\alpha)$ exists, and that $\omega_2 \in (0, 1)$. Then, the inequalities*

$$(3.15) \quad \Theta_{\sigma_k}(\tilde{p}_k(\alpha)) \leq \Theta_{\sigma_k}(x_k),$$

$$(3.16) \quad g(\tilde{p}_k(\alpha))^\top Z_k t_k \geq \omega_2 g_k^\top Z_k t_k$$

are satisfied for some $\alpha > 0$.

Proof. We recall that if ξ_1 and ξ_2 are continuous functions on an interval $[a, b]$ having right derivatives on (a, b) with $\xi_1'(\alpha; 1) \leq \xi_2'(\alpha; 1)$ for all $\alpha \in (a, b)$, then $\xi_1(b) - \xi_1(a) \leq \xi_2(b) - \xi_2(a)$ (see, for instance, Schwartz [37, Chapter III, section 5, Remark 3]). Also, as in the proof of (2.3), denoting the norm $\|\cdot\|$ by η , we have

from (3.10)

$$\begin{aligned} (\eta \circ c \circ \tilde{p}_k)'(\alpha; 1) &= \eta'(c(\tilde{p}_k(\alpha)); (c \circ \tilde{p}_k)'(\alpha)) \\ &= \eta'(c(\tilde{p}_k(\alpha)); -c(\tilde{p}_k(\alpha))) \\ &= -\|c(\tilde{p}_k(\alpha))\|. \end{aligned}$$

Hence, using (2.3) and $\sigma_k \geq \|\lambda(\tilde{p}_k(\alpha))\|_D$, we obtain

$$\begin{aligned} &(\Theta_{\sigma_k} \circ \tilde{p}_k)'(\alpha; 1) \\ &= g(\tilde{p}_k(\alpha))^\top Z_k t_k + \lambda(\tilde{p}_k(\alpha))^\top c(\tilde{p}_k(\alpha)) - \sigma_k \|c(\tilde{p}_k(\alpha))\| \\ (3.17) \quad &\leq g(\tilde{p}_k(\alpha))^\top Z_k t_k. \end{aligned}$$

Then, the result of the proposition is clear when $g_k = 0$, because (3.16) readily holds ($t_k = 0$) and (3.17) implies that $(\Theta_{\sigma_k} \circ \tilde{p}_k)'(\alpha; 1) \leq 0$ for small $\alpha \geq 0$ (those for which $\tilde{p}_k(\alpha)$ exists). Therefore, (3.15) is satisfied for small $\alpha \geq 0$.

Suppose now that $g_k \neq 0$. Since $g_k^\top Z_k t_k < 0$ and $\omega_2 < 1$, (3.16) is not verified for small positive α , so there is a nonempty interval of the form $(0, \bar{\alpha}]$ on which (3.16) is false. Now, when (3.16) is not verified, one has from (3.17) that

$$(\Theta_{\sigma_k} \circ \tilde{p}_k)'(\alpha; 1) \leq \omega_2 g_k^\top Z_k t_k.$$

Therefore, we obtain

$$\Theta_{\sigma_k}(\tilde{p}_k(\alpha)) - \Theta_{\sigma_k}(x_k) \leq \omega_2 \alpha g_k^\top Z_k t_k \quad \text{for } \alpha \in (0, \bar{\alpha}].$$

Hence, (3.15) is trivially satisfied on $(0, \bar{\alpha}]$. On the other hand, because of this last inequality and the fact that $\alpha \mapsto \Theta_{\sigma_k}(\tilde{p}_k(\alpha))$ is bounded below, the interval $(0, \bar{\alpha}]$ cannot be arbitrarily large. Therefore, (3.16) must eventually be satisfied. At the first step-size $\alpha > 0$ for which (3.16) holds, (3.15) is still verified by continuity. The proposition is proved. \square

We are now ready to describe the actual search path, which may be seen as an explicit Euler approximation of the solution of (3.10) with well-chosen discretization points. Similarly, the actual function ν_k is not given by (3.13) (with which global convergence could not be obtained) but is a piecewise linear approximation of this function with the same discretization points. A successful idea is to introduce a discretization point α_k^i only when the discretized form of (3.1) holds for $\alpha = \alpha_k^i$.

3.2. The search algorithm. We assume that the current iterate x_k is in Ω and that it is not stationary: $\|g_k\| + \|c_k\| \neq 0$. Let constants ω_1 and ω_2 be given in $(0, 1)$.

The search algorithm is iterative and generates, for $i = 0, \dots, i_k - 1$, intermediate step-size candidates α_k^i , points x_k^i , descent directions d_k^i of Θ_{σ_k} at x_k^i , piecewise linear search paths \tilde{p}_k^i , and piecewise linear forcing functions ν_k^i . These functions ν_k^i , playing the role of ν_k in (3.1), may be taken as discontinuous. The following conditions will hold for $i = 1, \dots, i_k - 1$:

$$\begin{aligned} (3.18a) \quad &x_k^i \in \Omega, \\ (3.18b) \quad &\Theta_{\sigma_k}(x_k^i) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \nu_k^{i-1}(\alpha_k^i), \\ (3.18c) \quad &g(x_k^i)^\top Z_k t_k < \omega_2 g_k^\top Z_k t_k, \\ (3.18d) \quad &\sigma_k \geq \|\lambda(x_k^i)\|_D + \bar{\sigma}. \end{aligned}$$

Inequality (3.18b) means that descent is forced at each iteration, while (3.18c) means that the curvature condition does not hold. Note that (3.18c) implies that x_k^i is not stationary.

At the beginning, the iteration index i is set to 0, $\alpha_k^0 = 0$, and $x_k^0 = x_k$. To initialize the recurrence, we define $\nu_k^{-1}(0) = 0$. It is also assumed that B_k is positive definite and that $\sigma_k \geq \|\lambda_k\|_D + \bar{\sigma}$. Then, (3.18a, b, d) clearly hold for $i = 0$. Stage i ($i \geq 0$) of the search comprises the following steps.

STAGE i OF THE PLS ALGORITHM.

1. Choose a tangent scaling factor $\tau_k^i > 0$ and compute the direction d_k^i defined by

$$(3.19) \quad d_k^i = \tau_k^i Z^-(x_k^i) Z_k t_k - A^-(x_k^i) c(x_k^i).$$

Update the search path p_k^i :

$$p_k^i(\alpha) = \begin{cases} p_k^{i-1}(\alpha) & \text{for } 0 \leq \alpha < \alpha_k^i \\ x_k^i + (\alpha - \alpha_k^i) d_k^i & \text{for } \alpha \geq \alpha_k^i. \end{cases}$$

Update the function ν_k^i (see below).

2. Determine a step-size $\alpha_k^{i+1} > \alpha_k^i$ from x_k^i along d_k^i such that

$$(3.20) \quad x_k^{i+1} = x_k^i + (\alpha_k^{i+1} - \alpha_k^i) d_k^i$$

is in Ω and the descent condition

$$(3.21) \quad \Theta_{\sigma_k}(p_k^i(\alpha)) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \nu_k^i(\alpha)$$

holds for $\alpha = \alpha_k^{i+1}$.

3. If $i = 0$ and some (unspecified) update criterion does not hold, set $i_k = 1$, $\alpha_k = \alpha_k^1$, $x_{k+1} = x_k^1$, $p_k = p_k^0$, $\nu_k = \nu_k^0$ and quit the PLS algorithm.
4. Linearize the constraints at x_k^{i+1} and test the curvature condition

$$(3.22) \quad g(x_k^{i+1})^\top Z_k t_k \geq \omega_2 g_k^\top Z_k t_k.$$

If the latter holds, set $i_k = i + 1$, $\alpha_k = \alpha_k^{i+1}$, $x_{k+1} = x_k^{i+1}$, $p_k = p_k^i$, $\nu_k = \nu_k^i$, and quit the PLS algorithm.

5. If the penalty parameter σ_k is not sufficiently large to have

$$(3.23) \quad \sigma_k \geq \|\lambda(x_k^{i+1})\|_D + \bar{\sigma},$$

set $i_k = i + 1$, $\alpha_k = \alpha_k^{i+1}$, $x_{k+1} = x_k^{i+1}$, $p_k = p_k^i$, $\nu_k = \nu_k^i$, and quit the PLS algorithm.

Let us give more details on the steps of the algorithm.

Step 1. The factor $\tau_k^i > 0$ scales the tangential component of the direction d_k^i . One reason for introducing this factor is that it may be convenient to use different step-sizes for the transversal and longitudinal part of the displacement. Indeed, second order information is used transversally, while a quasi-Newton model is used longitudinally. Another reason for using different transversal and longitudinal step-sizes will come from the discussion in section 3.6.

When $i = 0$ (initially) and $\tau_k^0 = 1$, d_k^0 has the form of the reduced SQP direction d_k and is tangent to \tilde{p}_k at 0. For $i \geq 1$, the direction comes from the discretization of (3.10): its longitudinal component $\tau_k^i Z^-(x_k^i) Z_k t_k$ is tangent to $\mathcal{M}_{x_k^i}$ at x_k^i , and the unscaled reduced direction $Z_k t_k$ is kept unchanged from one stage to the other.

The update formula of ν_k^{i-1} includes two natural possibilities. They correspond to the setting of the parameter ρ_k^i to 0 or 1 below. So, let ρ_k^i be any number in $[0, 1]$ and let us introduce the notion of *total decrease* of Θ_{σ_k} at x_k^i as the positive quantity

$$(3.24) \quad T_k^i = \Theta_{\sigma_k}(x_k) - \Theta_{\sigma_k}(x_k^i).$$

Then, ν_k^i is defined by

$$(3.25) \quad \nu_k^i(\alpha) = \begin{cases} \nu_k^{i-1}(\alpha) & \text{for } 0 \leq \alpha < \alpha_k^i, \\ (1 - \rho_k^i)\nu_k^{i-1}(\alpha_k^i) \\ \quad + \rho_k^i(-T_k^i/\omega_1) + (\alpha - \alpha_k^i)\Theta'_{\sigma_k}(x_k^i; d_k^i) & \text{for } \alpha \geq \alpha_k^i. \end{cases}$$

When $\rho_k^i = 0$, ν_k^i is continuous and the search can be viewed as a discretization of the smooth search described in section 3.1. This corresponds to a loose search. When $\rho_k^i = 1$, the search is closer to the “skipping rule” strategy discussed in section 3.3 below. It is also more demanding, since ν_k^i is more negative (use (3.18b)).

Step 2. Observe that d_k^i is a descent direction of Θ_{σ_k} at x_k^i , since by (2.3)

$$(3.26) \quad \Theta'_{\sigma_k}(x_k^i; d_k^i) = \tau_k^i g(x_k^i)^\top Z_k t_k + \lambda(x_k^i)^\top c(x_k^i) - \sigma_k \|c(x_k^i)\|,$$

which is negative when (3.18c) and (3.18d) hold. Then, it is standard to verify that, with conditions (3.18a), (3.18b), and $\omega_1 \in (0, 1)$, one can find a step-size $\alpha = \alpha_k^{i+1} > \alpha_k^i$ such that x_k^{i+1} is in Ω and the descent condition (3.21) holds.

Step 3. If $i = 0$, it is the right place to ask whether the pursuit of the search is useful. Indeed, unlike in unconstrained optimization, the curvature condition (3.2) is not strong enough to force global convergence (see section 4). It is only useful for guaranteeing the positive definiteness of the generated matrices. On the other hand, the role of the update criterion is to judge whether an update is appropriate by appreciating the quality of the information contained in the pair (γ_k, δ_k) . We believe that this appreciation has to be done when $i = 0$ so that a PLS is not launched without necessity. We shall not be more specific on this update criterion, because the results below do not need it. For these results, it can be any rule such as “never update” or “always update.” A better rule is used, however, in the numerical experiments of section 5. For more information on this subject, see Nocedal and Overton [31] or Gilbert [16].

Step 4. By linearization of the constraints at a point x , we mean the computation of the Jacobian matrix $A(x)$, the basis $Z^-(x)$, and the right inverse $A^-(x)$. At step 4, the curvature condition (3.22) is tested. If it holds, the search terminates. From step 2 and (3.22), the point x_{k+1} is in Ω and satisfies the reduced Wolfe conditions (3.1)–(3.2) with $\alpha = \alpha_k$.

Step 5. If (3.22) is not satisfied, one has to check whether the penalty parameter is sufficiently large to continue the search from x_k^{i+1} , i.e., whether (3.23) holds. If such is the case, all the conditions in (3.18) hold and a new iteration can start after having increased i by one. Otherwise, the search is interrupted (another possibility would have been to increase σ_k and to pursue the search).

3.3. Additional comments. To summarize, there are three facts that can interrupt the search algorithm: either (i) the update criterion does not hold in step 3 after α_k^1 is determined in step 2, or (ii) the conditions (3.1)–(3.2) are satisfied in step 4, or (iii) the penalty parameter σ_k is not large enough to guarantee that the next search direction is a descent direction of Θ_{σ_k} (step 5). We shall show in section 4

that under natural assumptions the algorithm does not cycle and terminates on one of these situations.

As announced above, when $\tau_k^i = 1$ for all i , the path p_k is a piecewise linear approximation of the bicomponent guiding path \tilde{p}_k , which was obtained by an explicit Euler discretization of the differential equation (3.10) at the step-sizes α_k^i . Furthermore, if $\rho_k^i = 0$ for all i , ν_k can also be viewed as a discretization of the function ν_k defined by (3.13) with p_k instead of \tilde{p}_k : $\nu_k'(\alpha_k^i; 1) = \Theta'_{\sigma_k}(x_k^i; d_k^i) = (\Theta_{\sigma_k} \circ p_k)'(\alpha_k^i; 1)$.

Remark that the search direction d_k^1 is close to

$$\check{d}_k^1 = -Z^-(x_k^1)B_k^{-1}g(x_k^1) - A^-(x_k^1)c(x_k^1),$$

which is the direction that would be taken in an algorithm skipping the update of B_k at x_k^1 when $\gamma_k^\top \delta_k$ is nonpositive or when the curvature condition does not hold (*skipping rule*). When $\rho_k^1 = 1$ in the definition of ν_k^1 above, inequality (3.21) becomes

$$\Theta_{\sigma_k}(x_k^1 + (\alpha - \alpha_k^1)d_k^1) \leq \Theta_{\sigma_k}(x_k^1) + \omega_1 (\alpha - \alpha_k^1) \Theta'_{\sigma_k}(x_k^1; d_k^1),$$

which is also the condition to realize in an algorithm with skipping rule.

The only difference between \check{d}_k^1 and d_k^1 is that in the latter the reduced gradient is also kept unchanged. The main motivation for this choice is explained in section 3.1: if the matrices $A^-(x_k^i)$ and $Z^-(x_k^i)$ are good in the sense of (3.11), the search consists of minimizing $(f \circ \psi_k)$ along the reduced direction $Z_k t_k$ (the meaning of ψ_k is given in Figure 3.1). With this in mind, it makes sense to update the matrix B_k using the vectors

$$(3.27) \quad \gamma_k = g_{k+1} - g_k \quad \text{and} \quad \delta_k = \left(\sum_{i=0}^{i_k-1} \tau_k^i (\alpha_k^{i+1} - \alpha_k^i) \right) Z_k t_k.$$

Note that when $\tau_k^i = 1$ for all i , $\delta_k = \alpha_k Z_k t_k$, simply.

When $\rho_k^i = 1$ for all i , the PLS algorithm applied to unconstrained problems ($c(x_k^i) = 0$ for all i) is related to the method of Moré and Sorensen [28] (see also Moré and Thuente [29, Section 2]). The differences are that Moré and Sorensen look for a point satisfying the *strong* Wolfe conditions (for this reason our method terminates more quickly), and the slope of the pieces of the forcing function ν_k^i is kept unchanged in their method (while we adapt it to the current point x_k^i).

For $i \geq 0$, we introduce the notion of *forced decreased* of Θ_{σ_k} at x_k^{i+1} as the positive quantity

$$(3.28) \quad F_k^{i+1} = -\omega_1 \sum_{l=0}^i (\alpha_k^{l+1} - \alpha_k^l) \Theta'_{\sigma_k}(x_k^l; d_k^l).$$

Using (3.18b) and the definition (3.25) of ν_k^i , we get, for $i \geq 1$,

$$(3.29) \quad F_k^i \leq -\omega_1 \nu_k^{i-1}(\alpha_k^i) \leq T_k^i,$$

where T_k^i is the total decrease of Θ_{σ_k} defined by (3.24).

We conclude this section by some comments on the cost of the PLS algorithm. The main requirement of this method is the linearization of the constraints (the computation of A , A^- , and Z^-) at the intermediate points x_k^i ($1 \leq i \leq i_k - 1$). This apparently damning cost must be reappraised in view of the following two facts. First, we have shown in [18] that it is possible to combine the PLS technique with

a suitable update criterion such that, asymptotically, each time the update criterion holds, the PLS algorithm succeeds without intermediate point ($i_k = 1$) and with unit step-size ($\alpha_k = 1$). Therefore, one can expect that in practice very few inner iterations will be necessary in the PLS algorithm. This is confirmed by the limited numerical experiments presented in section 5. Secondly, the work realized during the inner iterations of the PLS algorithm helps to find a better approximation of the solution: the search along the inner direction d_k^i makes the linearization at x_k^i useful. In fact, since d_k^i is close to a standard reduced SQP direction, one could consider all the intermediate iterates x_k^i as “true” iterates. It is a matter of presentation to group in a single iteration all the stages between two matrix updates.

3.4. Successive backtrackings. When a step-size candidate α_k^i is not accepted because inequality (3.22) does not hold, one has to determine the next tangent scaling factor $\tau_k^i > 0$ and the next step-size candidate α_k^{i+1} such that

$$\alpha_k^{i+1} > \alpha_k^i, \quad x_k^{i+1} = p_k^i(\alpha_k^{i+1}) \in \Omega, \quad \text{and} \quad (3.21) \text{ holds with } \alpha = \alpha_k^{i+1}.$$

This cannot be done in an uncontrolled manner. In particular, τ_k^i cannot be arbitrarily small or large, and α_k^{i+1} cannot be chosen too close to α_k^i . In this section, we describe a method for determining α_k^{i+1} that will ensure the finite termination of the search algorithm.

The determination can be divided into two stages. In the *forward* or *extrapolation* stage, a step-size $\alpha_k^{i,1} > \alpha_k^i$ is taken along d_k^i . The *backward* or *interpolation* stage is iterative: as long as (for the current trial with a step-size $\alpha_k^{i,j}$ ($j \geq 1$)) $p_k^i(\alpha_k^{i,j})$ is not in Ω or (3.21) does not hold for $\alpha = \alpha_k^{i,j}$, a new trial is made with a step-size $\alpha_k^{i,j+1} \in (\alpha_k^i, \alpha_k^{i,j})$. By requiring that $\alpha_k^{i,j}$ converges to α_k^i when $j \rightarrow \infty$, $p_k^i(\alpha_k^{i,j})$ will be in Ω and (3.21) with $\alpha = \alpha_k^{i,j}$ will hold for some finite index j . We denote by j_i the *first* index j for which this occurs and set

$$\alpha_k^{i+1} = \alpha_k^{i,j_i}.$$

We also suppose that $\{\alpha_k^{i,j}\}_{j \geq 1}$ does not tend too fast to α_k^i : the closer α_k^{i+1} is to α_k^i , the larger j_i must be. The rigorous form of our assumptions follows.

Assumptions 3.7. We suppose that the determination of the tangent scaling factor $\tau_k^i > 0$ and the step-sizes $\alpha_k^{i,j}$ is such that

- (i) the sequences $\{\tau_k^i\}_{i \geq 0}$ and $\{1/\tau_k^i\}_{i \geq 0}$ are bounded,
- (ii) the sequence $\{\alpha_k^{i,j}\}_{j \geq 1}$ converges to α_k^i ,
- (iii) if the increasing sequence $\{\alpha_k^i\}_{i \geq 1}$ converges to some step-size $\bar{\alpha}_k$, then
 - (a) for any index $j' \geq 1$, there is an index $i' \geq 1$ such that $j_i \geq j'$ for all $i \geq i'$,
 - (b) for any $j \geq 1$, the sequence $\{\alpha_k^{i,j}\}_{i \geq 1}$ converges to a step-size $\bar{\alpha}_k^{\infty,j} \neq \bar{\alpha}_k$,
 - (c) the sequence $\{\bar{\alpha}_k^{\infty,j}\}_{j \geq 1}$ converges to $\bar{\alpha}_k$.

Assumption 3.7 (iii-a) means that when $\{\alpha_k^i\}_{i \geq 1}$ converges, the number $(j_i - 1)$ of interpolations must go to infinity when $i \rightarrow \infty$.

Assumption 3.7 (i) is not difficult to satisfy. On the other hand, an easy way of satisfying Assumptions 3.7 (ii) and (iii), while using its favorite extrapolation and interpolation formulas, is to use some safeguard rules. Here is an example of rules that guarantee Assumptions 3.7 (ii) and (iii).

Example of safeguard rules for $\alpha_k^{i,j}$.

1. Choose $\varepsilon_E > 0$ and $\varepsilon_I \in (0, 1/2)$.
2. Extrapolation safeguard: for $i \geq 0$, choose $\alpha_k^{i,1} \geq \alpha_k^i + \varepsilon_E$.

3. Interpolation safeguard: for $i \geq 0$ and $j \geq 2$, choose

$$\alpha_k^{i,j} \in \left[(1 - \varepsilon_I)\alpha_k^i + \varepsilon_I\alpha_k^{i,j-1}, \varepsilon_I\alpha_k^i + (1 - \varepsilon_I)\alpha_k^{i,j-1} \right].$$

Let us show that Assumptions 3.7 (ii), (iii-a), and (iii-c) are satisfied if these rules are used. Observe that, for $i \geq 1$ and $j \geq 1$,

$$(3.30) \quad \alpha_k^i < \alpha_k^{i,j} \leq \alpha_k^i + (1 - \varepsilon_I)^{j-1}(\alpha_k^{i,1} - \alpha_k^i).$$

Therefore, Assumption 3.7 (ii) is verified. On the other hand, suppose that $\{\alpha_k^i\}_{i \geq 1}$ converges, and choose an index $j' \geq 1$. Then, one can find an index $i' \geq 1$ such that

$$\alpha_k^{i+1} - \alpha_k^i \leq \varepsilon_E \varepsilon_I^{j'-1} \quad \forall i \geq i'.$$

As

$$\alpha_k^{i+1} = \alpha_k^{i,j_i} \geq (1 - \varepsilon_I^{j_i-1})\alpha_k^i + \varepsilon_I^{j_i-1}\alpha_k^{i,1} \geq \alpha_k^i + \varepsilon_E \varepsilon_I^{j_i-1},$$

we have from the previous inequality that

$$\varepsilon_E \varepsilon_I^{j_i-1} \leq \varepsilon_E \varepsilon_I^{j'-1} \quad \forall i \geq i'.$$

Now, because $\varepsilon_I < 1$, we obtain $j_i \geq j'$ for all $i \geq i'$, which is Assumption 3.7 (iii-a). Finally, Assumption 3.7 (iii-c) is also guaranteed by the above rules as this can be seen by taking the limit on i and then on j in (3.30).

We have not discussed the case of Assumption 3.7 (iii-b), but it also can easily be satisfied by taking for $i \geq 0$, for example,

$$\alpha_k^{i,j} = \begin{cases} \alpha_k^i + \varepsilon_E & \text{if } j = 1, \\ \frac{1}{2}(\alpha_k^i + \alpha_k^{i,j-1}) & \text{if } j \geq 2, \end{cases}$$

which is compatible with the safeguard rules given above. More appropriate interpolation rules would use the known values of Θ_{σ_k} and its directional derivatives.

3.5. Finite termination of the search algorithm. The next proposition gives conditions that ensure the finite termination of the PLS algorithm described in sections 3.2 and 3.4 at a point x_{k+1} satisfying

$$(3.31) \quad \Theta_{\sigma_k}(x_{k+1}) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \nu_k(\alpha_k),$$

$$(3.32) \quad g_{k+1}^\top Z_k t_k \geq \omega_2 g_k^\top Z_k t_k,$$

where the function ν_k is defined recursively in step 1 of the algorithm (see (3.25)). Recall that the search path p_k is also defined recursively in step 1 of the algorithm.

PROPOSITION 3.8. *Suppose that f and c are differentiable on Ω , c is a submersion on Ω , and the decomposition of \mathbb{R}^n described in section 1 is made with maps Z^- and A^- which are bounded on Ω . Let x_k be a point in Ω and B_k be a symmetric positive definite matrix of order $n - m$. Suppose that the penalty factor σ_k in (1.12) satisfies (2.2). Then, if the PLS algorithm described in sections 3.2 and 3.4, with Assumptions 3.7, $\omega_1 \in (0, 1)$, and $\omega_2 > 0$, is started from x_k , one of the following situations occurs:*

(i) *the algorithm terminates after a finite number of stages with a step-size $\alpha_k > 0$, a point $x_{k+1} \in \Omega$, and a function ν_k satisfying conditions (3.31) and (3.32);*

(ii) the algorithm terminates prematurely with a step-size $\alpha_k > 0$, a point $x_{k+1} \in \Omega$, and a function ν_k satisfying (3.31) only, because either the update criterion does not hold at x_k^1 or (3.23) fails at $x_k^{i+1} = x_{k+1}$;

(iii) the algorithm builds a sequence of points $\{x_k^i\}_{i \geq 1}$ in Ω and either $\Theta_{\sigma_k}(x_k^i)$ tends to $-\infty$ or $\{x_k^i\}_{i \geq 1}$ tends to a point on the boundary of Ω .

Proof. We have already observed in section 3.3 that if the algorithm terminates, then either situation (i) or (ii) occurs.

Suppose now that the algorithm cycles: a sequence $\{x_k^i\}_{i \geq 1}$ is built in Ω . This can only occur when $g_k \neq 0$, since (3.32) is always satisfied when $t_k = 0$ and (3.31) is satisfied at x_k^1 . We have to show that one of the events given in (iii) occurs. We proceed by contradiction, supposing that $\{\Theta_{\sigma_k}(x_k^i)\}_{i \geq 1}$ is bounded from below and that $\{x_k^i\}_{i \geq 1}$ does not converge to a point on the boundary of Ω . We recall that conditions (3.18) are satisfied for all $i \geq 1$.

Step 1. Let us prove that the sequences $\{F_k^i\}_{i \geq 1}$, $\{\nu_k^{i-1}(\alpha_k^i)\}_{i \geq 1}$, and $\{\alpha_k^i\}_{i \geq 1}$ converge, say to \bar{F}_k , \bar{N}_k , and $\bar{\alpha}_k$, respectively.

The first sequence is increasing and the second is decreasing; hence, from (3.29), they will converge if we prove that $\{T_k^i\}$ is bounded. But this is clear since $T_k^i = \Theta_{\sigma_k}(x_k) - \Theta_{\sigma_k}(x_k^i) \geq 0$ and $\{\Theta_{\sigma_k}(x_k^i)\}$ is supposed to be bounded below.

On the other hand, using the definition (3.28) of F_k^{i+1} , (3.26), (2.2), (3.18d), $g_k^\top Z_k t_k \leq 0$, and (3.18c), we obtain

$$\begin{aligned} F_k^{i+1} &\geq -\omega_1 \sum_{l=0}^i \tau_k^l (\alpha_k^{l+1} - \alpha_k^l) g(x_k^l)^\top Z_k t_k \\ &\geq -\omega_1 \omega_2 \left(\sum_{l=1}^i \tau_k^l (\alpha_k^{l+1} - \alpha_k^l) \right) g_k^\top Z_k t_k. \end{aligned}$$

Then, the boundedness of $\{F_k^i\}_{i \geq 1}$, $g_k^\top Z_k t_k < 0$, and $\omega_1 \omega_2 > 0$ imply that

$$(3.33) \quad \sum_{i \geq 0} \tau_k^i (\alpha_k^{i+1} - \alpha_k^i) < +\infty.$$

As $\{\tau_k^i\}_{i \geq 0}$ is bounded away from 0 by Assumption 3.7 (i), $\{\alpha_k^i\}_{i \geq 1}$ converges.

Step 2. Let us show that the sequence $\{x_k^i\}_{i \geq 1}$ converges to a point $\bar{x}_k \in \Omega$.

By definition of F_k^{i+1} , $g_k^\top Z_k t_k \leq 0$, (3.18c), (2.2), and (3.18d), we obtain

$$F_k^{i+1} \geq \omega_1 \bar{\sigma} \sum_{l=0}^i (\alpha_k^{l+1} - \alpha_k^l) \|c(x_k^l)\|.$$

Since $\{F_k^i\}_{i \geq 1}$ is bounded, we have the convergence of the series

$$(3.34) \quad \sum_{i \geq 0} (\alpha_k^{i+1} - \alpha_k^i) \|c(x_k^i)\| < +\infty.$$

Now, by definition of x_k^i ,

$$x_k^i = x_k + \sum_{l=0}^{i-1} (\alpha_k^{l+1} - \alpha_k^l) (\tau_k^l Z^-(x_k^l) Z_k t_k - A^-(x_k^l) c(x_k^l)).$$

Using the boundedness of $Z^-(\cdot)$ and $A^-(\cdot)$ on Ω , (3.33), and (3.34), we see that the series in the right-hand side is absolutely convergent when $i \rightarrow \infty$. Therefore, the

series is convergent and x_k^i converges to a limit point \bar{x}_k . By our assumptions, \bar{x}_k cannot be a point on the boundary of Ω ; hence, $\bar{x}_k \in \Omega$.

This implies the following convergence when $i \rightarrow \infty$ (see (3.24)):

$$T_k^i \rightarrow \bar{T}_k = \Theta_{\sigma_k}(x_k) - \Theta_{\sigma_k}(\bar{x}_k).$$

Furthermore, since $\{\tau_k^i\}$ and $\{1/\tau_k^i\}$ are bounded by Assumption 3.7 (i), there is some $\bar{\tau}_k > 0$ and a subsequence $\mathcal{I} \in \mathbb{N}$ such that for $i \rightarrow \infty$, $i \in \mathcal{I}$, we have $\tau_k^i \rightarrow \bar{\tau}_k$ and (using (3.19), (3.26), and (2.3))

$$\begin{aligned} d_k^i &\rightarrow \bar{d}_k = \bar{\tau}_k Z^-(\bar{x}_k) Z_k t_k - A^-(\bar{x}_k) c(\bar{x}_k), \\ \Theta'_{\sigma_k}(x_k^i; d_k^i) &\rightarrow \bar{\tau}_k g(\bar{x}_k)^\top Z_k t_k + \lambda(\bar{x}_k)^\top c(\bar{x}_k) - \sigma_k \|c(\bar{x}_k)\| = \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k). \end{aligned}$$

Step 3. Let us conclude with the expected contradiction.

Define

$$x_k^{i,j} = x_k^i + (\alpha_k^{i,j} - \alpha_k^i) d_k^i.$$

By Assumption 3.7 (iii-b), the sequence $\{\alpha_k^{i,j}\}_{i \geq 1}$ converges to a step-size $\bar{\alpha}_k^{\infty,j} \neq \bar{\alpha}_k$. Therefore, for any $j \geq 1$,

$$x_k^{i,j} \rightarrow \bar{x}_k^{\infty,j} = \bar{x}_k + (\bar{\alpha}_k^{\infty,j} - \bar{\alpha}_k) \bar{d}_k \quad \text{when } i \rightarrow \infty \text{ with } i \in \mathcal{I}.$$

Now, for fixed $j \geq 1$, Assumption 3.7 (iii-a) says that $j_i > j$ for sufficiently large i . This means that, for large i , $x_k^{i,j}$ is not accepted in step 2 of the PLS algorithm. Hence, either $x_k^{i,j} \notin \Omega$ or (3.21) is not verified with $\alpha = \alpha_k^{i,j}$. This can be written

$$\begin{aligned} x_k^{i,j} \in \Omega &\implies \Theta_{\sigma_k}(x_k^{i,j}) > \Theta_{\sigma_k}(x_k) + \omega_1 \nu_k^i(\alpha_k^{i,j}) \\ &= \Theta_{\sigma_k}(x_k) + \omega_1 \nu_k^i(\alpha_k^{i+1}) + \omega_1 (\alpha_k^{i,j} - \alpha_k^{i+1}) \Theta'_{\sigma_k}(x_k^i; d_k^i). \end{aligned}$$

Taking the limit on $i \in \mathcal{I}$ in this relation and using $\omega_1 \bar{N}_k \geq -\bar{T}_k$ from (3.29) and the results of step 2, we obtain

$$\begin{aligned} \bar{x}_k^{\infty,j} \in \Omega &\implies \Theta_{\sigma_k}(\bar{x}_k^{\infty,j}) \geq \Theta_{\sigma_k}(x_k) - \bar{T}_k + \omega_1 (\bar{\alpha}_k^{\infty,j} - \bar{\alpha}_k) \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) \\ &= \Theta_{\sigma_k}(\bar{x}_k) + \omega_1 (\bar{\alpha}_k^{\infty,j} - \bar{\alpha}_k) \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k). \end{aligned}$$

Hence,

$$\bar{x}_k^{\infty,j} \in \Omega \implies \frac{\Theta_{\sigma_k}(\bar{x}_k^{\infty,j}) - \Theta_{\sigma_k}(\bar{x}_k)}{\bar{\alpha}_k^{\infty,j} - \bar{\alpha}_k} \geq \omega_1 \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k).$$

Because $\bar{x}_k \in \Omega$, taking the limit in this implication when j tends to infinity gives (with Assumption 3.7 (iii-c)) $\Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) \geq \omega_1 \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k)$. Because $\omega_1 < 1$, we get

$$\Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) \geq 0.$$

On the other hand,

$$\begin{aligned} \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) &= \bar{\tau}_k g(\bar{x}_k)^\top Z_k t_k + \lambda(\bar{x}_k)^\top c(\bar{x}_k) - \sigma_k \|c(\bar{x}_k)\| \\ &\leq \bar{\tau}_k \omega_2 g_k^\top Z_k t_k \\ &< 0, \end{aligned}$$

because $\|\lambda(\bar{x}_k)\|_D \leq \sigma_k$ from the limit in (3.18d), $g(\bar{x}_k)^\top Z_k t_k \leq \omega_2 g_k^\top Z_k t_k$ from the limit in (3.18c), $\bar{\tau}_k \omega_2 > 0$, and $g_k \neq 0$. This inequality contradicts the nonnegativity of $\Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k)$ obtained above and concludes the proof. \square

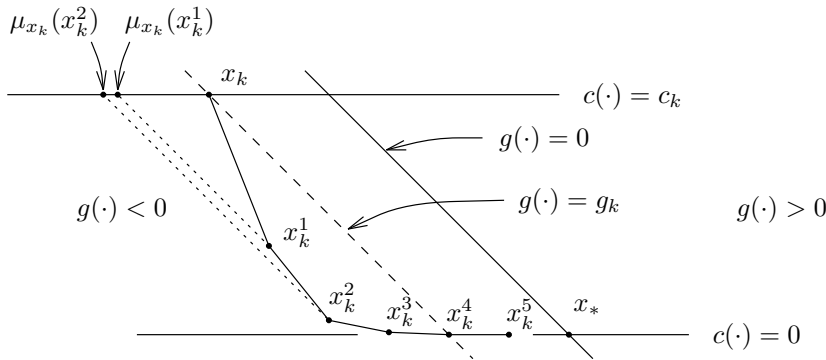


FIG. 3.4. A difficult case for the PLS.

3.6. Resetting the PLS. In some cases, the PLS described in sections 3.2 and 3.4 can be trapped in a situation where its behavior is poor. Such a situation may happen when conditions (3.11) do not hold, in particular, when the path $\mu_{x_k} \circ p_k$ is not a descent path for f . Then, the search algorithm may necessitate a large number of inner iterations to satisfy the reduced Wolfe conditions, and the vectors γ_k and δ_k may be erroneous. We show how to improve the PLS algorithm in this situation.

Here is an example of such a situation, in which $n = 2$ and $m = 1$. Take

$$f(x) = \frac{1}{4} (x_{(1)} + x_{(2)})^2 \quad \text{and} \quad c(x) = e^{x_{(2)}} - 1,$$

where $x_{(i)}$ denotes the i th component of x . The unique solution of this problem is clearly $x_* = 0$. With the following decomposition of \mathbb{R}^2 ,

$$Z^-(x) = e_1 \quad \text{and} \quad A^-(x) = e^{-x_{(2)}} e_2,$$

where (e_1, e_2) is the canonical basis of \mathbb{R}^2 , the reduced gradient is given by $g(x) = (x_{(1)} + x_{(2)})/2$, and the transversal component of the steps are orthogonal to the constraint manifold. The manifolds $c(\cdot) = 0$ and $g(\cdot) = 0$ are the lines $x_{(2)} = 0$ and $x_{(1)} + x_{(2)} = 0$ represented in Figure 3.4.

Now, suppose that the current iterate x_k has coordinates $(-1 - \epsilon, 1)$, with $\epsilon > 0$, and that $B_k = I$. Consider an implementation of the PLS in which $\omega_1 = 10^{-4}$, $\omega_2 = 0.9$, $\rho_k^i = 1$, $\tau_k^i = 1$, and the first step-size candidate is $\alpha_k^{i,1} = \alpha_k^i + 1$. If the penalty parameter $\sigma_k = 10$, the step-size $\alpha_k^{i,1}$ is always accepted by the Armijo condition (3.21). When $\epsilon = 0.5$ the search algorithm requires 5 inner iterations. The intermediate points $\{x_k^i\}_{i=1}^5$ are represented in Figure 3.4. By decreasing $\epsilon > 0$, one can obtain as many inner iterations as desired. For example, 21 inner iterations are necessary for $\epsilon = 0.1$, while 2001 are necessary for $\epsilon = 10^{-3}$! The reason is that when ϵ decreases, x_k is closer to the manifold $g(\cdot) = 0$ and the reduced tangent direction $Z_k t_k$ is smaller. Because the iterates go rapidly close to the constraint manifold where the reduced gradient is much more negative than at x_k and because the reduced gradients evaluated at the intermediate points are not used for defining the search directions, the algorithm needs more and more inner iterations to cross the manifold $g(\cdot) = g_k$ (represented by a dashed line in Figure 3.4), beyond which it has to go to satisfy the curvature condition (3.22). If the search path is mapped by μ_{x_k} (see Definition 3.3) on the line $c(\cdot) = c_k$, it is clear from Figure 3.4 that the mapped path starts in the

wrong left direction. The correct direction is followed from $\mu_{x_k}(x_k^2)$ only. The basic reason for this behavior is, once again, that reduced methods have no information on the space tangent to the manifold of constant reduced gradient.

In reduced quasi-Newton methods, the update criterion is often a rule that suggests not updating the matrix when the tangential component of the direction is small with respect to its transversal component (see Nocedal and Overton [31] or Gilbert [16]). In the example above, this would result in skipping the update when ϵ is small. By step 3 of the PLS algorithm, the search would be interrupted at x_k^1 , avoiding the large number of inner iterations that we have observed. Unfortunately, the implementation of update criteria is often less efficient than expected. Therefore, we propose a modification of the PLS algorithm, such that the situation of the example above is faced with more success. In the modified version, the curvature condition (3.32) is replaced by

$$(3.35) \quad g_{k+1}^\top Z_k t_k \geq \omega_2 \min_{0 \leq i < i_k} g(x_k^i)^\top Z_k t_k.$$

The PLS algorithm with this new condition is said to be “with resetting” and it is denoted by “PLS-rst” below. Since inequality (3.35) is less restrictive than (3.32), it is clear that PLS-rst terminates more quickly than PLS. In particular, it still has the finite termination property of Proposition 3.8. Questions concerning the global convergence of the algorithm with PLS and PLS-rst are discussed in the next section. On the example above, this new version of the algorithm terminates in 3, 5, and 204 inner iterations when $\epsilon = 0.5, 0.1,$ and 10^{-3} , respectively.

When the PLS is reset at an intermediate point $x_k^{l_k}$, where l_k gives the current arg-minimum in (3.35), the reduced direction $Z_k t_k$ may be very small (this is the case in the example above), so that guessing the correct tangent step-size (or the tangent scaling factor τ_k^i) by using an extrapolation formula may be useful. For example, one can try to use $g(x_k^{i-1})^\top Z_k t_k$ and $g(x_k^i)^\top Z_k t_k$ to evaluate τ_k^{i+1} . The rationale behind this is that, when (3.7) and (3.11) hold, $g(x_k^{i-1})^\top Z_k t_k$ and $g(x_k^i)^\top Z_k t_k$ are derivatives of the function $\alpha \mapsto (f \circ \psi_k)(\alpha Z_k t_k)$. Hence, when $g(x_k^{i-1})^\top Z_k t_k < g(x_k^i)^\top Z_k t_k$, one can use quadratic interpolation to determine τ_k^{i+1} . Using this, the runs with $\epsilon = 0.5, 0.1,$ and 10^{-3} terminate now in 3, 4, and 5 inner iterations, respectively.

With PLS-rst, the vectors γ_k and δ_k used to update B_k have to be modified. If l_k denotes the largest index for which the minimum in (3.35) is reached, then it is appropriate to take

$$\gamma_k = g_{k+1} - g(x_k^{l_k}) \quad \text{and} \quad \delta_k = \left(\sum_{i=l_k}^{i_k-1} \tau_k^i (\alpha_k^{i+1} - \alpha_k^i) \right) Z_k t_k.$$

Note again that when $\tau_k^i = 1$ for all i , $\delta_k = (\alpha_k - \alpha_k^{l_k}) Z_k t_k$, simply. With this choice, $\gamma_k^\top \delta_k > 0$. Note also that these vectors usually put better information into the matrix B_{k+1} , because the new value of δ_k is generally closer to the reduced step from $x_k^{l_k}$ to x_{k+1} than the previous value of δ_k is close to the reduced step from x_k to x_{k+1} . This remark particularly applies to the example above. From Figure 3.4, we have $\gamma_k^{\text{PLS}} = g(x_k^5) - g_k \simeq g(x_k^5) - g(x_k^4) \simeq g(x_k^3) - g(x_k^2) = \gamma_k^{\text{PLS-rst}}$. But $\delta_k^{\text{PLS}} = 5\delta_k^{\text{PLS-rst}}$ and it is clear that $\delta_k^{\text{PLS-rst}}$ corresponds better to $\gamma_k^{\text{PLS}} \simeq \gamma_k^{\text{PLS-rst}}$ than δ_k^{PLS} .

4. Convergence result. In this section, we show that the PLS method of section 3 is able to force convergence of reduced secant algorithms from remote starting points. For this, we shall suppose that the calculation of the reduced matrices keeps

the sequences $\{B_k\}$ and $\{B_k^{-1}\}$ bounded. This is a rather strong assumption, but the present state of the convergence theory for constrained problems is not sufficiently developed to have significantly better results. For instance, Byrd and Nocedal [4] analyze the global convergence of reduced quasi-Newton algorithms under conditions that are not known to be guaranteed by the present step-size determination method.

The algorithm we consider is therefore not fully determined, since we shall not be very specific on the way the matrices are updated (in particular, the update criterion will remain unspecified). A possibility is to use the BFGS update formula (1.8), which is always well defined when the PLS succeeds. There is still another facet of the algorithm that must be clarified—this is how the penalty parameter σ_k is updated. We suppose that a rule is chosen such that the following three properties are satisfied ($\bar{\sigma} > 0$ is a constant):

$$(4.1) \quad \begin{cases} \sigma_k \geq \|\lambda_k\|_D + \bar{\sigma} \quad \forall k \geq 1, \\ \exists \text{ an index } k_1, \quad \forall k \geq k_1, \quad \sigma_{k-1} \geq \|\lambda_k\|_D + \bar{\sigma} \implies \sigma_k = \sigma_{k-1}, \\ \{\sigma_k\} \text{ is bounded} \implies \sigma_k \text{ is updated finitely often.} \end{cases}$$

Many rules can satisfy these conditions. For example, Mayne and Polak [27] suggest taking ($\tilde{\sigma} > 1$):

$$(4.2) \quad \text{if } \sigma_{k-1} \geq \|\lambda_k\|_D + \bar{\sigma}, \text{ then } \sigma_k = \sigma_{k-1}, \text{ else } \sigma_k = \max(\tilde{\sigma}\sigma_{k-1}, \|\lambda_k\|_D + \bar{\sigma}).$$

We can now outline the algorithm, whose convergence is analyzed in Proposition 4.2. At the beginning, the iteration index k is set to 1 and the constants ω_1 and ω_2 used in the PLS algorithm are chosen in $(0, 1)$. When the k th iteration starts, an iterate $x_k \in \Omega$ is known, as well as a positive definite matrix B_k . Then the PLS technique is used to determine the next iterate x_{k+1} such that $x_{k+1} \in \Omega$ and (3.31) (and possibly (3.32)) hold. Then, the matrix B_k is updated, provided that the PLS algorithm has not been interrupted prematurely by an update criterion or the failure of (3.23). Finally, the penalty parameter is updated according to the rules (4.1).

In unconstrained optimization, the curvature condition corresponding to (3.32) prevents the step-size from being too small, which is important for the global convergence of the algorithm. In constrained problems, this is not necessarily the case, because condition (3.32) ignores the transversal component of the search path. For example, when the objective function f is constant the reduced gradient vanishes and (3.32) is satisfied for any step-sizes, independently of the form of the search path. Therefore, something has to be done such that the first step-size candidate α_k^1 ($\leq \alpha_k$) will not be too small. For the same reason, the first tangent scaling factor τ_k^0 must be chosen bounded away from zero. We gather below additional conditions that the tuning of the PLS algorithm must take into account in order to get global convergence.

Assumptions 4.1. We suppose that the determination of the tangent scaling factors $\tau_k^0 > 0$ and the step-sizes α_k^1 is such that

- (i) the sequences $\{\tau_k^0\}_{k \geq 1}$ and $\{1/\tau_k^0\}_{k \geq 1}$ are bounded,
- (ii) the sequence $\{\alpha_k^{0,1}\}_{k \geq 1}$ is bounded away from zero,
- (iii) there exists a constant $\beta \in (0, 1)$ such that for all $k \geq 1$ and $j \geq 1$, $\alpha_k^{0,j+1} \geq \beta \alpha_k^{0,j}$.

Note that these assumptions are compatible with Assumptions 3.7 and the safeguard rules given afterwards. Assumptions 4.1 (ii) and (iii) can be satisfied, for instance, by using Armijo's backtracking to determine the first step-size candidate α_k^1 from a constant value for $\alpha_k^{0,1}$. In quasi-Newton methods, $\tau_k^0 = 1$ and $\alpha_k^{0,1} = 1$ are recommended.

In Proposition 4.2 below, we suppose that a sequence $\{x_k\}$ is generated in Ω . This implicitly supposes that the PLS algorithm never cycles: situation (i) or (ii) of Proposition 3.8 occurs at each iteration. We denote by $\text{dist}(x, \Omega^c)$ the Euclidean distance between a point x and the complementary set of Ω .

PROPOSITION 4.2. *Suppose that f and c are differentiable on Ω with Lipschitz continuous derivatives, that c is a submersion on Ω , that the map A^- is continuous and bounded on Ω , and that Z^- is bounded on Ω . Suppose also that the algorithm for solving problem (1.4) outlined above generates a sequence $\{x_k\}$ in Ω by the PLS method with Assumptions 3.7 and 4.1 and that the constants ω_1 and ω_2 are taken in $(0, 1)$. Suppose finally that the symmetric positive definite matrices B_k used in the algorithm are such that $\{B_k\}$ and $\{B_k^{-1}\}$ are bounded. Then, one of the following situations occurs:*

- (i) $\{\sigma_k\}_{k \geq 1}$ is unbounded and $\{x_k : \sigma_k \neq \sigma_{k-1}\}$ has no accumulation point in Ω ,
- (ii) σ_k is modified finitely often and one of the following situations occurs:
 - (a) $g_k \rightarrow 0$ and $c_k \rightarrow 0$,
 - (b) $\Theta_{\sigma_k}(x_k) \rightarrow -\infty$,
 - (c) $\text{dist}(x_k, \Omega^c) \rightarrow 0$ for some subsequence of indices $k \rightarrow \infty$.

Proof. First, consider situation (i): $\{\sigma_k\}$ is unbounded. Let \mathcal{K} be the subsequence of indices $\{k : \sigma_k \neq \sigma_{k-1}, k \geq k_1\}$ (k_1 given by (4.1)). From (4.1),

$$\sigma_{k-1} < \|\lambda_k\|_D + \bar{\sigma} \quad \text{for } k \in \mathcal{K}.$$

As $\{\sigma_k\}_{k \geq k_1}$ is increasing, if it is unbounded, the inequality above shows that the sequence $\{\|\lambda_k\|_D\}_{k \in \mathcal{K}}$ tends to ∞ . Then, by continuity of $x \mapsto \lambda(x)$ on Ω , $\{x_k : \sigma_k \neq \sigma_{k-1}\}$ has no accumulation point in Ω .

Suppose now that $\{\sigma_k\}$ is bounded. By (4.1), σ_k is modified finitely often: $\sigma_k = \sigma$ for $k \geq k_2$, say. Suppose also that $\Theta_{\sigma_k}(x_k)$ is bounded from below and that $\{x_k\}$ remains away from Ω^c . We have to prove that situation (ii-a) of the proposition occurs. We denote by C an ‘‘absorbing’’ positive constant independent of k .

From the definition (3.28) of $F_k^{i_k}$, (3.26), the fact that (3.18c) holds for $i = 1, \dots, i_k - 1$, $g_k^\top Z_k t_k \leq 0$, (4.1), (3.18d), and the boundedness of $\{B_k\}$, we have the following for $k \geq k_2$:

$$\begin{aligned} F_k^{i_k} &= -\omega_1 \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i) \Theta'_\sigma(x_k; d_k^i) \\ &\geq \omega_1 \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i) (\omega_2 \tau_k^i g_k^\top B_k^{-1} g_k + \bar{\sigma} \|c(x_k^i)\|) \\ (4.3) \quad &\geq C \left(\sum_{i=0}^{i_k-1} \tau_k^i (\alpha_k^{i+1} - \alpha_k^i) \|g_k\|^2 + \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i) \|c(x_k^i)\| \right). \end{aligned}$$

As the sequence $\{\Theta_\sigma(x_k)\}_{k \geq k_2}$ decreases and is bounded below, it converges. Then, from (3.31) and $\omega_1 \nu_k(\alpha_k) \leq -F_k^{i_k}$ (use (3.29) with $i = i_k$), we see that $F_k^{i_k} \rightarrow 0$. Therefore, the terms in the right-hand side of (4.3) converge to zero when $k \rightarrow \infty$:

$$(4.4) \quad \begin{cases} \sum_{i=0}^{i_k-1} \tau_k^i (\alpha_k^{i+1} - \alpha_k^i) \|g_k\|^2 \rightarrow 0, \\ \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i) \|c(x_k^i)\| \rightarrow 0. \end{cases}$$

The result (ii-a) will be proved if we show that the step-size candidates α_k^1 are bounded away from zero. Indeed, from (4.4) and Assumption 4.1 (i), this implies that $g_k \rightarrow 0$ and $c_k \rightarrow 0$. We proceed by contradiction, supposing that for some subsequence \mathcal{K} of indices $k \geq k_2$ we have

$$(4.5) \quad \alpha_k^1 \rightarrow 0, \quad \text{when } k \rightarrow \infty \text{ in } \mathcal{K}.$$

By Assumptions 4.1 (ii) and (iii), we can suppose that, for $k \in \mathcal{K}$, $\alpha_k^1 < \alpha_k^{0,1}$ (therefore $\alpha_k^1 = \alpha_k^{0,j_1}$ for some $j_1 \geq 2$) and $\alpha_k^{0,j_1-1} \leq 1$.

Observe first that we can also suppose that, for $k \in \mathcal{K}$, α_k^{0,j_1-1} is not accepted by the search algorithm because the descent condition (3.21) does not hold for $i = 0$ and $\alpha = \alpha_k^{0,j_1-1}$. Indeed, otherwise we would have a subsequence $\mathcal{K}' \subset \mathcal{K}$ such that

$$(4.6) \quad x_k^{0,j_1-1} \notin \Omega \quad \text{for } k \in \mathcal{K}'.$$

Recall that $r_k = -A_k^- c_k$. We have $x_k^{0,j_1-1} - x_k = \alpha_k^{0,j_1-1} (\tau_k^0 t_k + r_k)$ and, by Assumption 4.1 (iii), $\alpha_k^{0,j_1-1} \leq \alpha_k^1 / \beta \leq \alpha_k / \beta$. Then, using the boundedness of $\{\alpha_k^1\}_{k \in \mathcal{K}}$ (due to (4.5)); Assumption 4.1 (i); the boundedness of $\{Z_k^-\}$, $\{B_k^{-1}\}$, and $\{A_k^-\}$; and (4.4); we have for $k \rightarrow \infty$ in \mathcal{K}

$$\begin{aligned} \|\alpha_k^{0,j_1-1} \tau_k^0 t_k\|^2 &\leq C \alpha_k^1 \tau_k^0 \|g_k\|^2 \rightarrow 0, \\ \|\alpha_k^{0,j_1-1} r_k\| &\leq C \alpha_k^1 \|c_k\| \rightarrow 0. \end{aligned}$$

Therefore, $(x_k^{0,j_1-1} - x_k) \rightarrow 0$ for $k \rightarrow \infty$ in \mathcal{K} , and (4.6) would imply that $\text{dist}(x_k, \Omega^c)$ tends to 0 for $k \rightarrow \infty$ in \mathcal{K}' , in contradiction with our assumptions.

Therefore, we can suppose that (3.21) is not satisfied for $i = 0$, $\alpha = \alpha_k^{0,j_1-1}$, and $k \in \mathcal{K}$, i.e.,

$$(4.7) \quad \Theta_\sigma(x_k^{0,j_1-1}) > \Theta_\sigma(x_k) + \omega_1 \alpha_k^{0,j_1-1} (\tau_k^0 g_k^\top Z_k t_k + \lambda_k^\top c_k - \sigma \|c_k\|).$$

We obtain a contradiction with (4.5) by showing that this may not occur for too small α_k^{0,j_1-1} . For this, we expand the left-hand side of (4.7) about x_k .

First, using the Lipschitz continuity of f' on Ω ,

$$f(x_k + \alpha \tau t_k + \alpha r_k) \leq f_k + \alpha \tau g_k^\top Z_k t_k + \alpha \lambda_k^\top c_k + C \alpha^2 (\tau^2 \|t_k\|^2 + \|r_k\|^2).$$

Similarly, using the Lipschitz continuity of c' on Ω , we get the following for $\alpha \leq 1$:

$$\begin{aligned} \|c(x_k + \alpha \tau t_k + \alpha r_k)\| &\leq \|c_k - \alpha c_k\| + C \alpha^2 (\tau^2 \|t_k\|^2 + \|r_k\|^2) \\ &= \|c_k\| - \alpha \|c_k\| + C \alpha^2 (\tau^2 \|t_k\|^2 + \|r_k\|^2). \end{aligned}$$

Grouping these estimates, we obtain the following for $\alpha \leq 1$:

$$\begin{aligned} \Theta_\sigma(x_k + \alpha \tau t_k + \alpha r_k) &\leq \Theta_\sigma(x_k) + \alpha (\tau g_k^\top Z_k t_k + \lambda_k^\top c_k - \sigma \|c_k\|) \\ &\quad + C \alpha^2 (\tau^2 \|t_k\|^2 + \|r_k\|^2). \end{aligned}$$

Using this inequality in (4.7) gives (recall that $\alpha_k^{0,j_1-1} \leq 1$ for $k \in \mathcal{K}$)

$$\begin{aligned} (1 - \omega_1) \alpha_k^{0,j_1-1} (\tau_k^0 g_k^\top B_k^{-1} g_k - \lambda_k^\top c_k + \sigma \|c_k\|) \\ < C (\alpha_k^{0,j_1-1})^2 \left((\tau_k^0)^2 \|g_k\|^2 + \|c_k\|^2 \right) \quad \text{for } k \in \mathcal{K}. \end{aligned}$$

With the boundedness of $\{B_k\}$, $\{\tau_k^0\}$ and $\{1/\tau_k^0\}$ and the inequalities $\omega_1 < 1$ and $\sigma \geq \|\lambda_k\|_D + \bar{\sigma}$, we obtain

$$\alpha_k^{0,j_1-1} \|g_k\|^2 + \alpha_k^{0,j_1-1} \|c_k\| < C(\alpha_k^{0,j_1-1})^2 (\|g_k\|^2 + \|c_k\|^2) \quad \text{for } k \in \mathcal{K}.$$

By (4.4) and $\alpha_k^{0,j_1-1} \leq \alpha_k^1/\beta$, $\alpha_k^{0,j_1-1} \|c_k\| \rightarrow 0$. Hence, the inequality above gives

$$\alpha_k^{0,j_1-1} \|g_k\|^2 + \alpha_k^{0,j_1-1} \|c_k\| < C(\alpha_k^{0,j_1-1})^2 \|g_k\|^2 + C\epsilon_k \alpha_k^{0,j_1-1} \|c_k\| \quad \text{for } k \in \mathcal{K},$$

where $\epsilon_k \rightarrow 0$ for $k \in \mathcal{K}$. Finally,

$$\alpha_k^{0,j_1-1} \|g_k\|^2 < C(\alpha_k^{0,j_1-1})^2 \|g_k\|^2 \quad \text{for large } k \in \mathcal{K}.$$

Clearly, this strict inequality shows that $\{\alpha_k^{0,j_1-1}\}_{k \in \mathcal{K}}$ is bounded away from zero. As $\alpha_k^1 \geq \beta \alpha_k^{0,j_1-1}$, $\{\alpha_k^1\}_{k \in \mathcal{K}}$ cannot converge to zero, contradicting (4.5).

This contradiction concludes the proof. \square

When g is Lipschitz continuous on Ω , Assumptions 4.1 are no longer necessary to prove that $g_k \rightarrow 0$. This can be shown by a standard argument, using (3.32) and (4.4). But we were not able to prove that $c_k \rightarrow 0$ without these assumptions, for the reasons given above the statement of Assumptions 4.1. On the other hand, once Assumptions 4.1 hold, condition (3.32) is no longer useful for the global convergence (it is not used in the proof above). In this case, if the PLS algorithm is replaced by the PLS-rst method described in section 3.6, the conclusion of Proposition 4.2 still holds.

5. Numerical experiment. The behavior of the PLS technique introduced in section 3 and the reduced quasi-Newton algorithm presented in section 4 have been tested on two model problems with a dimension ranging from $n = 2$ to 500 and a single constraint. They consist in minimizing quadratic functions on the unit sphere. Since there is just one constraint, this problem does not favor reduced SQP methods. A full SQP method should be more efficient on this problem.

The numerical experiments have been done in double precision on a SUN SPARC-station 1, with a program written in Fortran-77.

Test problem I. In the first test problem, the function f to minimize and the constraint function c are defined on $\Omega = \{x \in \mathbb{R}^n : x_{(1)} > 0\}$ by

$$f(x) = \frac{1}{2} \sum_{i=1}^n (a_{(i)} x_{(i)} - 1)^2, \quad c(x) = \frac{1}{2} (\|x\|_2^2 - 1).$$

Here $v_{(i)}$ denotes the i th component of a vector v . The constants $a_{(i)}$ are set to $(n + 1 - i)/n$ for $1 \leq i \leq n$. The problem is more and more difficult to solve as n increases because the order of the updated matrices and the condition number of the reduced Hessian of the Lagrangian increase with n .

The Jacobian matrix of the constraints $A(x) = x^\top$ is surjective if $x \neq 0$. The matrix $Z^-(x)$, whose columns form a basis of the space tangent to the constraint manifold, and the restoration operator $A^-(x)$ are chosen as follows:

$$(5.1) \quad Z^-(x) = \begin{pmatrix} -x_{(2)} & -\cdots & -x_{(n)} \\ x_{(1)} I_{n-1} \end{pmatrix}, \quad A^-(x) = \frac{x}{\|x\|_2^2},$$

where I_{n-1} is the identity matrix of order $n - 1$. These matrices are well defined and injective for $x \in \Omega$. The form of $A^-(x)$ shows that the transversal steps are

TABLE 5.1
Test problems.

| n | $x_{*(1)}$ | $\kappa_2(B_*)$ |
|-----|------------|-----------------|
| 2 | 0.69 | 1. |
| 5 | 0.53 | 6. |
| 10 | 0.42 | 9. |
| 20 | 0.32 | 14. |
| 50 | 0.22 | 26. |
| 100 | 0.16 | 46. |
| 200 | 0.12 | 84. |
| 500 | 0.075 | 192. |

orthogonal (for the Euclidean scalar product) to the space tangent to the constraint manifold.

Table 5.1 gives some information on the problems: n is the number of variables (hence, $n - 1$ is the dimension of the constraint manifold and the order of the matrix to update), $x_{*(1)}$ is the first component of the solution, and $\kappa_2(B_*)$ is the ℓ_2 condition number of the reduced Hessian of the Lagrangian at the solution (computed by the LAPACK program DSYEV). Note that although the Hessian of the Lagrangian $L(x, \lambda)$ is a diagonal matrix and $Z^-(x)$ is sparse, the reduced Hessian $Z^-(x)^\top L(x, \lambda) Z^-(x)$ is dense: its (i, j) element is $(a_1^2 + \lambda)x_{i+1}x_{j+1} + (a_{i+1}^2 + \lambda)x_1^2\delta_{ij}$.

To globalize the algorithm, the exact ℓ_1 penalty function (with the ℓ_1 -norm in (1.12)) is used with $\sigma_1 = 2\|\lambda_1\|_\infty$ initially. Next, σ_k is updated by the rule (4.2) with $\bar{\sigma} = \sigma_1/100$ and $\tilde{\sigma} = 2$. The initial point x_1 has its i th component set to $(-1)^{i-1}10$, and the algorithm stops at the point x_k when

$$\|c_k\|_2 \leq 10^{-7}\|c_1\|_2 \quad \text{and} \quad \|g_k\|_2 \leq 10^{-7}\|g_1\|_2.$$

The update of the matrix B_k^{-1} is done with the inverse BFGS formula when it is appropriate (this depends on the algorithm and is specified below). The first time this occurs, for $k = k_0$ say, the inverse matrix is first initialized to $\gamma_{k_0}^\top \delta_{k_0} / \|\gamma_{k_0}\|^2 I$ before being updated.

The results of our experiments on test problem I are given in Tables 5.2 to 5.6 and summarized in Table 5.7. Here are some common symbols: “ n ” is the dimension of the problem, “iter” is the number of iterations, “lin” is the number of times the constraints are linearized, “func” is the number of function calls, “skip” is the number of times the matrix update is skipped, and “ $\sigma \nearrow$ ” is the number of increases of the penalty parameter. The meaning of some other symbols is given below.

To serve as a reference, the first runs have been made with Armijo’s backtracking along $d_k = t_k + r_k$ and the *skipping rule*: if at the point found by the search algorithm $\gamma_k^\top \delta_k$ is positive, B_k is updated; otherwise, the update is skipped. This algorithm is denoted by AS-skip. The results are given in Table 5.2.

We see that the number of skips is usually small, except for the cases $n = 20$ and $n = 500$.

In the next experiment, Armijo’s backtracking is still used as search technique, but a correction is made to δ_k when $\gamma_k^\top \delta_k$ is not sufficiently positive (the so-called Powell’s correction; see Powell [34]): $\tilde{\delta}_k = \theta \delta_k + (1 - \theta) B_k^{-1} \gamma_k$, where

$$\theta = \begin{cases} 1 & \text{if } \gamma_k^\top \delta_k \geq 0.2 \gamma_k^\top B_k^{-1} \gamma_k, \\ 0.8 \frac{\gamma_k^\top B_k^{-1} \gamma_k}{\gamma_k^\top B_k^{-1} \gamma_k - \gamma_k^\top \delta_k} & \text{otherwise.} \end{cases}$$

TABLE 5.2

AS-skip: Armijo's search and skipping rule (I).

| n | iter | func | skip | $\sigma \nearrow$ |
|-----|------|------|------|-------------------|
| 2 | 17 | 23 | 1 | 0 |
| 5 | 55 | 58 | 1 | 1 |
| 10 | 88 | 93 | 5 | 1 |
| 20 | 110 | 116 | 31 | 2 |
| 50 | 82 | 89 | 3 | 3 |
| 100 | 83 | 95 | 2 | 3 |
| 200 | 72 | 90 | 3 | 4 |
| 500 | 91 | 98 | 11 | 5 |

TABLE 5.3

AS-Powell: Armijo's search and Powell's correction (I).

| n | iter | func | P-cor | $\sigma \nearrow$ |
|-----|------|------|-------|-------------------|
| 2 | 17 | 23 | 1 | 0 |
| 5 | 55 | 58 | 2 | 1 |
| 10 | 86 | 89 | 3 | 1 |
| 20 | 98 | 110 | 14 | 2 |
| 50 | 74 | 82 | 6 | 3 |
| 100 | 95 | 118 | 18 | 3 |
| 200 | 78 | 89 | 4 | 4 |
| 500 | 104 | 132 | 11 | 5 |

The update of B_k^{-1} is then made with $(\gamma_k, \tilde{\delta}_k)$ instead of (γ_k, δ_k) . This algorithm is denoted by AS-Powell. Table 5.3 shows the results: "P-cor" is the number of Powell's corrections, i.e., the number of times $\theta \neq 1$ in the formula of $\tilde{\delta}_k$ above. We see that this algorithm works slightly better than the method with skipping rule for small n ($n \leq 50$) and slightly worse for larger n . We believe that this may not be fortuitous and may come from update pairs (γ_k, δ_k) of bad quality, in particular of the initial one, which is used to scale the matrix. Indeed, for small n , the effect of an initial pair with wrong information is rapidly compensated by updates with good pairs (this is clearly the case when $n = 2$, since then the matrix to update has order 1 and the update formula is memoryless). On the other hand, from our experience [20], if n is large and if the first pair used to scale the matrix is spoiled, it may take many updates to recover from this bad initial scaling.

For two reasons, we introduce an update criterion in the algorithm AS-Powell. First, we want to see whether an update criterion improves the algorithm by selecting good pairs (γ_k, δ_k) and, second, we want to offer a fairer comparison with algorithms using the PLS, which naturally require update criteria. We take the following inequality as update criterion:

$$(5.2) \quad \|r_k\|_2 \leq \mu \|e_{k \ominus 2}^1\|_2 \|t_k\|_2.$$

An update is desirable when the inequality holds. In this criterion, μ is a positive constant, $k \ominus 2$ is the index of the last but one iteration at which an update occurred before iteration k (see [16] or [18]), and $e_k^1 = \alpha_k^1 d_k$. The value used for μ is important for the efficiency of the criterion. In order to get a sufficiently good initial pair, we

TABLE 5.4

AS-Powell-UC: Armijo's search, Powell's correction, and update criterion (I).

| n | iter | func | skip | P-cor | σ ↗ |
|-----|------|------|------|-------|------------|
| 2 | 18 | 27 | 3 | 1 | 0 |
| 5 | 19 | 26 | 7 | 0 | 1 |
| 10 | 27 | 35 | 5 | 0 | 1 |
| 20 | 40 | 51 | 8 | 2 | 2 |
| 50 | 56 | 72 | 8 | 4 | 3 |
| 100 | 43 | 50 | 4 | 1 | 3 |
| 200 | 48 | 63 | 5 | 1 | 3 |
| 500 | 57 | 79 | 10 | 6 | 4 |

TABLE 5.5

PLS: Piecewise line-search and update criterion (I).

| n | iter | lin | func | skip | σ ↗ |
|-----|------|-----|------|------|------------|
| 2 | 16 | 21 | 30 | 3 | 0 |
| 5 | 19 | 20 | 26 | 7 | 1 |
| 10 | 27 | 28 | 35 | 5 | 1 |
| 20 | 37 | 38 | 46 | 7 | 2 |
| 50 | 51 | 53 | 63 | 7 | 3 |
| 100 | 43 | 44 | 50 | 7 | 3 |
| 200 | 47 | 48 | 61 | 5 | 3 |
| 500 | 43 | 56 | 66 | 10 | 4 |

take for μ the quotient

$$\mu = 0.1 \frac{\|r_1\|_2}{\|e_1^1\|_2 \|t_1\|_2},$$

so the update criterion cannot be satisfied before a few iterations have been done. This forces the algorithm to choose as its initial scaling pair $(\gamma_{k_0}, \delta_{k_0})$ a better pair than (γ_1, δ_1) .

The results of algorithm AS-Powell with this update criterion, denoted as AS-Powell-UC, are given in Table 5.4. They are remarkably better than those of algorithm AS-Powell: the number of iterations and function calls has decreased by 49% and 43%, respectively. This confirms our feeling on the importance of selecting good pairs (in particular the first one).

The last two experiments use the PLS technique, provided that the update criterion (5.2) holds. Hence, the update is skipped when the PLS is interrupted by the update criterion or by the test on the penalty parameter (step 5 of the search algorithm). As far as the PLS algorithm is concerned, we have always set $\rho_k^i = 1$ in (3.25), which corresponds to a demanding search. The results with $\rho_k^i = 0$ hardly differ, essentially because the unit step-size is usually accepted by the PLS. The first tangent scaling factor τ_k^0 and the step-size candidates $\alpha_k^{i,1}$ are always set to 1 and $\alpha_k^i + 1$, respectively. Safeguarded quadratic interpolation is used to determine the intermediate step-sizes $\{\alpha_k^{i,j}\}_{j=2}^{j_i}$.

In the first experiment, whose results are given in Table 5.5, the plain PLS method described in sections 3.2 and 3.4 is used with τ_k^i always set to 1 (without tangential extrapolation). A first observation is that the PLS algorithm never cycles, as this is suggested by the theory (Proposition 3.8). Now, comparing the number of linearizations with those of the algorithms with Powell's correction, we observe an important

TABLE 5.6

PLS-rst: Piecewise line-search (with resetting) and update criterion (I).

| n | iter | lin | func | rst | skip | $\sigma \nearrow$ |
|-----|------|-----|------|-----|------|-------------------|
| 2 | 17 | 19 | 27 | 1 | 3 | 0 |
| 5 | 19 | 20 | 26 | 0 | 7 | 1 |
| 10 | 27 | 28 | 35 | 0 | 5 | 1 |
| 20 | 37 | 38 | 46 | 0 | 7 | 2 |
| 50 | 51 | 53 | 65 | 1 | 8 | 3 |
| 100 | 43 | 44 | 50 | 0 | 7 | 3 |
| 200 | 47 | 48 | 61 | 0 | 5 | 3 |
| 500 | 48 | 52 | 65 | 1 | 10 | 4 |

TABLE 5.7

Compared performance of the algorithms (I).

| Algorithm | iter | lin | func |
|--------------|------|-----|------|
| AS-skip | 598 | 606 | 662 |
| AS-Powell | 607 | 615 | 701 |
| AS-Powell-UC | 308 | 316 | 403 |
| PLS | 283 | 308 | 377 |
| PLS-rst | 289 | 302 | 375 |

improvement with respect to algorithm AS-Powell and a small one with respect to algorithm AS-Powell-UC. The results look quite satisfactory, particularly if we observe that the small improvement with respect to algorithm AS-Powell-UC is due to a very limited use of the PLS technique. Only the cases $n = 2$, $n = 50$, and $n = 500$ use this technique, as this can be seen by a positive number of inner iterations: “lin” – “iter” – 1 > 0. Now the results with $n = 500$ are not very good, since the PLS algorithm requires a great number of inner iterations. By looking more closely at these results, however, we have observed that the deterioration is due to a single iteration and that a phenomenon resembling the one described in the example of section 3.6 occurs.

The last experiment is done with the PLS-rst algorithm of section 3.6. The PLS algorithm is interrupted as soon as condition (3.35) holds. Furthermore, the tangent scaling factor τ_k^i may be different from 1: either τ_k^i is determined by a safeguarded quadratic or cubic extrapolation formula using the values $g(x_k^i)^\top Z_k t_k$ or (when this is unsuccessful, due to the inconsistency of the interpolating values) τ_k^i is doubled at each inner iteration (provided that the descent test (3.18b) has always been verified with $\alpha_k^i = \alpha_k^{i-1,1}$ during the current PLS). The results are given in Table 5.6. The number of iterations with “resettings” are given in a column labeled by ‘rst’: it is the number of times condition (3.35) differs from (3.32). Of course, only the results of the cases $n = 2$, $n = 50$, and $n = 500$ may change. We see that the very few “resettings” slightly improve the results. We also observe that the number of inner iterations used by the PLS-rst algorithm (= “lin” – “iter” – 1) is now very small.

To summarize, we add up the number of iterations, linearizations, and function calls used by the considered algorithms for all the runs: see Table 5.7 (for every run with AS-skip or AS-Powell, “lin” = “iter” + 1). The results of PLS-rst compare favorably with those of the other techniques.

Test problem II. The second test problem is obtained by changing the objective function in the first test problem. It is now the quadratic form $\frac{1}{2}x^\top Qx - q^\top x$, where

TABLE 5.8
Compared performance of the algorithms (II).

| Algorithm | iter | lin | func |
|--------------|-------|-------|-------|
| AS-skip | 1109* | 1117* | 1294* |
| AS-Powell | 495 | 503 | 583 |
| AS-Powell-UC | 381 | 389 | 490 |
| PLS | 375 | 462 | 533 |
| PLS-rst | 354 | 382 | 440 |

TABLE 5.9
Compared performance for Test problem I with tangent basis (5.3).

| Algorithm | iter | lin | func | saving in “lin” |
|--------------|------|-----|------|-----------------|
| AS-skip | 453 | 461 | 503 | 24 % |
| AS-Powell | 388 | 396 | 440 | 36 % |
| AS-Powell-UC | 279 | 287 | 370 | 9 % |
| PLS | 254 | 262 | 324 | 15 % |
| PLS-rst | 254 | 262 | 324 | 13 % |

TABLE 5.10
Compared performance for Test problem II with tangent basis (5.3).

| Algorithm | iter | lin | func | saving in “lin” |
|--------------|------|-----|------|-----------------|
| AS-skip | 235 | 243 | 298 | 78 %* |
| AS-Powell | 230 | 238 | 303 | 53 % |
| AS-Powell-UC | 252 | 260 | 357 | 33 % |
| PLS | 226 | 242 | 300 | 48 % |
| PLS-rst | 228 | 238 | 295 | 38 % |

$Q_{i,j} = (i + j - 1)^{-1}$ and $q_i = n$ for all $i, j \in \{1, \dots, n\}$. Table 5.8 summarizes the results. The “*” in this table indicates that algorithm AS-skip failed to satisfy the stopping test for $n = 500$. This is due to the fact that the matrix update is very often skipped in this run. The same type of comments as for Test problem I can be given:

- Algorithm AS-Powell works much better than AS-skip;
- The update criterion in AS-Powell-UC improves algorithm AS-Powell significantly;
- The less satisfactory results of the plain PLS algorithm are due to a large number of inner iterations in the PLS (no resettings, see section 3.6);
- The PLS-rst algorithm has the best results, with very few inner iterations.

Change of tangent basis. We would like to mention the results obtained by changing the field of tangent basis Z^- . We now take

$$(5.3) \quad Z^-(x) = \begin{pmatrix} -x_{(2)}/x_{(1)} & \cdots & -x_{(n)}/x_{(1)} \\ & & I_{n-1} \end{pmatrix}.$$

Hence, the elements of the previous matrix Z^- in (5.1) have been divided by $x_{(1)}$. This is motivated by the fact that this new basis satisfies property (3.7) for some parametrization ψ_k , while the basis (5.1) does not (see [19]). All the other parameters of the algorithms have been kept unchanged.

Tables 5.9 and 5.10 give the results corresponding to Test problems I and II. We observe an important improvement in the number of linearizations (last column, i.e., saving in “lin”). Note that this is not due to a change in the conditioning of the

problem: since the previous basis has just been divided by $x_{(1)}$, the condition number of the reduced Hessian of the Lagrangian at the solution has not changed. We still do not know whether the fact that the basis (5.3) satisfies (3.7) is a key to explain the improvement (see [19] for a possible explanation).

Comments on the numerical experiments. Among the techniques used to maintain the positive definiteness of the reduced matrices that have been tested (skipping rule, Powell's corrections, PLS technique), the PLS technique appears to be the best one, provided that the method is carefully implemented (PLS-rst version of the algorithm). For the test-problems we considered, two other tools are of great importance for the efficiency of reduced SQP methods: the use of update criteria and a proper choice of the tangent basis field Z^- . It is clear that this small amount of tests impedes from giving final conclusions. More experiments with more realistic problems are necessary before asserting the usefulness of the PLS technique. We have found, however, that these results are encouraging and we believe that this limited number of tests demonstrates the feasibility of the PLS approach.

6. Conclusions. This paper proposes a method for maintaining the positive definiteness of the matrices in reduced quasi-Newton algorithms for equality constrained optimization. By using a PLS (as opposed to a traditional line-search) technique, which conducts the search of the next iterate along a piecewise linear path, some reduced Wolfe conditions are satisfied whenever desired. One of these conditions is such that between two successive iterates, the function to minimize, reduced to the current manifold, seems to have positive curvature. This allows the algorithm to sustain the positive definiteness of the reduced Hessian approximations from one iteration to the other.

A few numerical experiments have shown that a careful implementation of the technique can do better than other methods, such as the skipping rule or Powell's correction of the BFGS update. This improvement is obtained with reduced methods, despite of their important defect, which is that they have no means to improve the orientation of the transversal component of the step. Because this defect is crucial in the present context (due to the first condition in (3.11)) and because it is not shared with the SQP method, it is expected that the PLS technique could be more clearly efficient when the updated matrices approximate the full Hessian of the augmented Lagrangian. This discussion also raises the question whether an update criterion based on (3.11) rather than on the comparison of the transversal and tangential components of the step can be conceived.

Another feature of the PLS technique is to offer the possibility to have cleaner algorithms. At least, this is an advantage for their analysis. For example, in [18], a strong superlinear convergence result has been proven for an algorithm with PLSs and the update criterion (5.2). It is shown, indeed, that if in the Coleman and Conn reduced algorithm the points $\{x_k^i\}_{k,i}$ converge to a solution satisfying sufficient second order conditions of optimality, then no intermediate point exists eventually ($i_k = 1$ for k large) and the sequence converges q -superlinearly. In this result, the matrices are supposed to be generated by the BFGS formula from any positive definite starting matrix. No other assumptions on the generated matrices are necessary. To our knowledge, this is the first extension of Powell's result (see [32]) to constrained problems.

REFERENCES

- [1] P. ARMAND AND J. CH. GILBERT (1995), *A Piecewise Line-Search Technique for Maintaining the Positive Definiteness of the Updated Matrices in the SQP method*, Rapport de Recherche 2615, INRIA, BP 105, 78153 Le Chesnay, France. Http server: <http://www.inria.fr/RRRT/RR-2615.html>; ftp server: <ftp://ftp.inria.fr/INRIA/publication/RR>, file RR-2615.ps.gz (submitted to *Comput. Optim. Appl.*).
- [2] L. ARMIJO (1966), *Minimization of functions having Lipschitz continuous first partial derivatives*, *Pacific J. Math.*, 16, pp. 1–3.
- [3] W. M. BOOTHBY (1986), *An Introduction to Differentiable Manifolds and Riemannian Geometry*, 2nd ed., Academic Press, Boston, MA.
- [4] R. H. BYRD AND J. NOCEDAL (1991), *An analysis of reduced Hessian methods for constrained optimization*, *Math. Programming*, 49, pp. 285–323.
- [5] R. H. BYRD, R. A. TAPIA, AND Y. ZHANG (1992), *An SQP augmented Lagrangian BFGS algorithm for constrained optimization*, *SIAM J. Optim.*, 2, pp. 210–241.
- [6] T. F. COLEMAN AND A. R. CONN (1982), *Nonlinear programming via an exact penalty function: Asymptotic analysis*, *Math. Programming*, 24, pp. 123–136.
- [7] T. F. COLEMAN AND A. R. CONN (1984), *On the local convergence of a quasi-Newton method for the nonlinear programming problem*, *SIAM J. Numer. Anal.*, 21, pp. 755–769.
- [8] T. F. COLEMAN AND P. A. FENYES (1992), *Partitioned quasi-Newton methods for nonlinear constrained optimization*, *Math. Programming*, 53, pp. 17–44.
- [9] L. CONLON (1993), *Differentiable Manifolds – A First Course*, Birkhäuser Boston, Cambridge, MA.
- [10] B. N. PSHENICHNYI AND YU. M. DANILIN (1978), *Numerical Methods for Extremal Problems*, MIR, Moscow.
- [11] J. E. DENNIS AND R. B. SCHNABEL (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.
- [12] P. FENYES (1987), *Partitioned Quasi-Newton Methods for Nonlinear Equality Constrained Optimization*, Ph.D. thesis, Department of Computer Science, Cornell University, Ithaca, NY.
- [13] R. FLETCHER (1980), *Practical Methods of Optimization. Volume 1 : Unconstrained Optimization*, John Wiley & Sons, Chichester, UK.
- [14] D. GABAY (1982), *Minimizing a differentiable function over a differential manifold*, *J. Optim. Theory Appl.*, 37, pp. 177–219.
- [15] D. GABAY (1982), *Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*, *Math. Programming Study*, 16, pp. 18–44.
- [16] J. CH. GILBERT (1988), *Mise à jour de la métrique dans les méthodes de quasi-Newton réduites en optimisation avec contraintes d'égalité*, *Modélisation Math. Anal. Numér.*, 22, pp. 251–288.
- [17] J. CH. GILBERT (1991), *Maintaining the positive definiteness of the matrices in reduced secant methods for equality constrained optimization*, *Math. Programming*, 50, pp. 1–28.
- [18] J. CH. GILBERT (1993), *Superlinear Convergence of a Reduced BFGS Method with Piecewise Line-Search and Update Criterion*, Rapport de Recherche 2140, INRIA, BP 105, 78153 Le Chesnay, France. Http server: <http://www.inria.fr/RRRT/RR-2140.html>; ftp server: <ftp://ftp.inria.fr/INRIA/publication/RR>, file RR-2140.ps.gz.
- [19] J. CH. GILBERT (1997), *Piecewise line-search techniques for constrained minimization by quasi-Newton algorithms*, in *Advances in Nonlinear Programming*, Proc. International Conference on Nonlinear Programming, Beijing, China, Kluwer Academic Publishers, Norwell, MA.
- [20] J. CH. GILBERT AND C. LEMARÉCHAL (1989), *Some numerical experiments with variable-storage quasi-Newton algorithms*, *Math. Programming*, 45, pp. 407–435.
- [21] P. E. GILL, W. MURRAY, AND M. H. WRIGHT (1981), *Practical Optimization*, Academic Press, New York.
- [22] S.-P. HAN (1976), *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, *Math. Programming*, 11, pp. 263–282.
- [23] S.-P. HAN AND O. L. MANGASARIAN (1979), *Exact penalty functions in nonlinear programming*, *Math. Programming*, 17, pp. 251–269.
- [24] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL (1993), *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, New York.
- [25] C. LEMARÉCHAL (1981), *A view of line-searches*, in *Optimization and Optimal Control*, A. Auslender, W. Oettli, and J. Stoer, eds., Lecture Notes in Control and Information Science 30, Springer-Verlag, Heidelberg, pp. 59–78.

- [26] D. C. LIU AND J. NOCEDAL (1989), *On the limited memory BFGS method for large scale optimization*, Math. Programming, 45, pp. 503–520.
- [27] D. Q. MAYNE AND E. POLAK (1982), *A superlinearly convergent algorithm for constrained optimization problems*, Math. Programming Study, 16, pp. 45–61.
- [28] J. J. MORÉ AND D. C. SORENSEN (1984), *Newton's method*, in Studies in Numerical Analysis, G. H. Golub, ed., The Mathematical Association of America, pp. 29–82.
- [29] J. J. MORÉ AND D. J. THUENTE (1992), *Line Search Algorithms with Guaranteed Sufficient Decrease*, Preprint MCS-P330-1092, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill.
- [30] W. MURRAY AND M. H. WRIGHT (1978), *Projected Lagrangian Methods Based on the Trajectories of Penalty and Barrier Functions*, Technical report SOL-78-23, Department of Operations Research, Stanford University, Stanford, CA.
- [31] J. NOCEDAL AND M. L. OVERTON (1985), *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22, pp. 821–850.
- [32] M. J. D. POWELL (1976), *Some global convergence properties of a variable metric algorithm for minimization without exact line searches*, Nonlinear Programming, SIAM-AMS Proceedings, R. W. Cottle and C. E. Lemke, eds., American Mathematical Society, Providence, RI.
- [33] M. J. D. POWELL (1978), *Algorithms for nonlinear constraints that use Lagrangian functions*, Math. Programming, 14, pp. 224–248.
- [34] M. J. D. POWELL (1978), *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis, G. A. Watson, ed., Springer, pp. 144–157.
- [35] M. J. D. POWELL (1985), *The performance of two subroutines for constrained optimization on some difficult test problems*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, PA.
- [36] R. T. ROCKAFELLAR (1970), *Convex Analysis*, Princeton University Press, Princeton, NJ.
- [37] L. SCHWARTZ (1992), *Analyse II: Calcul Différentiel et Équations Différentielles*. Hermann, Paris, France.
- [38] M. SPIVAK (1979), *Differentiable Geometry, Volume I*, 2nd ed., Publish or Perish, Inc., Houston, TX.
- [39] J. STOER (1984), *Principles of sequential quadratic programming methods for solving nonlinear programs*, in Proc. NATO ASI on Computational Mathematical Programming, Bad Windsheim, Germany.
- [40] R. A. TAPIA (1977), *Diagonalized multiplier methods and quasi-Newton methods for constrained optimization*, J. Optim. Theory Appl., 22, pp. 135–194.
- [41] R. A. TAPIA (1988), *On secant updates for use in general constrained optimization*, Math. Comput., 51, pp. 181–202.
- [42] R. B. WILSON (1963), *A Simplicial Algorithm for Concave Programming*, Ph.D. thesis, Graduate School of Business Administration, Harvard University, Boston, MA.
- [43] P. WOLFE (1969), *Convergence conditions for ascent methods*, SIAM Rev., 11, pp. 226–235.
- [44] P. WOLFE (1971), *Convergence conditions for ascent methods II: Some corrections*, SIAM Rev., 13, pp. 185–188.