# On the realization of the Wolfe conditions in reduced quasi-Newton methods for equality constrained optimization

Jean Charles Gilbert



INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *On the Realization of the Wolfe Conditions in Reduced Quasi-Newton Methods for Equality Constrained Optimization*

Jean Charles GILBERT

**N° 2127**
Décembre 1993

PROGRAMME 5

Traitement du signal,
automatique et
productique

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

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

# SUR LA REALISATION DES CONDITIONS DE WOLFE DANS LES METHODES DE QUASI-NEWTON REDUITES EN OPTIMISATION AVEC CONTRAINTES D'EGALITE

par

Jean Charles GILBERT

Rapport de Recherche n° 2127

6 Décembre 1993

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

# SUR LA REALISATION DES CONDITIONS DE WOLFE DANS LES METHODES DE QUASI-NEWTON REDUITES EN OPTIMISATION AVEC CONTRAINTES D'EGALITE

Jean Charles GILBERT[†]

## ABSTRACT

We propose a piecewise line-search technique for reduced quasi-Newton methods, which are designed for minimizing functions when nonlinear equality constraints are present. The search aims at realizing generalized Wolfe conditions. These conditions are suitable for the methods considered because they allow the algorithm to maintain naturally the positive definiteness of the matrices approximating the reduced Hessian of the Lagrangian.

## RÉSUMÉ

Nous proposons une technique de recherche linéaire par morceaux pour les méthodes de quasi-Newton réduites qui sont conçues pour minimiser des fonctions en présence de contraintes d'égalité non linéaires. Cette recherche vise à réaliser des conditions de Wolfe généralisées. Celles-ci conviennent aux algorithmes considérés car elles fournissent un moyen simple de maintenir la définie positivité des matrices approchant le hessien réduit du lagrangien.

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

**Abbreviated title:** Wolfe's conditions in RQN methods.

**AMS Subject Classification:** primary: 65K05; secondary: 49M37, 90C30.

[†] INRIA, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France).

# 1 Introduction

In unconstrained optimization, when a function $x \in \mathbb{R}^n \mapsto \varphi(x) \in \mathbb{R}$ is minimized using descent direction methods, there is a nice combination of a line-search technique attributed to Wolfe [35, 36] and some quasi-Newton methods. On the one hand, if $d_k$ is a descent direction of $\varphi$ at the current iterate $x_k$ (i.e., $\nabla\varphi(x_k)^\mathsf{T} d_k < 0$) 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

$$\varphi(x_{k+1}) \leq \varphi(x_k) + \omega_1\, \alpha_k\, \nabla\varphi(x_k)^\mathsf{T} d_k,$$

$$\nabla\varphi(x_{k+1})^\mathsf{T} d_k \geq \omega_2\, \nabla\varphi(x_k)^\mathsf{T} d_k,$$

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

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

where $\gamma_k = \nabla\varphi(x_{k+1}) - \nabla\varphi(x_k)$ is the change in the gradient 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

$$\gamma_k^\mathsf{T}\delta_k > 0.$$

Therefore, this curvature condition has to be satisfied if one expects $B_{k+1}$ to be positive definite. For some quasi-Newton formulæ (for instance the BFGS formula, see below), which update $B_k$ using $\gamma_k$ and $\delta_k$, 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 (clearly) provides this inequality. Hence, using the Wolfe line-search and the BFGS formula, say, ensures that all the search directions have the descent property.

For various reasons (see for example Powell [27]) it is not straightforward to extend the above scheme to constrained problems. This is unfortunate, because experiment has shown that the approach is very successful numerically in unconstrained minimization, even when the number of variables is large (see Liu and Nocedal [21], and Gilbert and Lemaréchal [16]). In this paper we study the matter more in detail and propose a "piecewise line-search" (PLS) technique that finds a point satisfying *generalized* Wolfe conditions. Our approach is given in the framework of reduced quasi-Newton methods for equality constrained optimization. Hence, the aim is to maintain positive definite the matrices approximating the *reduced* Hessian of the Lagrangian. Note that when the reduced Hessian is computed exactly, there is no

need to implement our search algorithm. In this case, a simple Armijo [1] backtracking along a search direction is preferable, since it is less expensive and easier to implement than the rather sophisticated search proposed here.

The work presented in this paper continues the previous study of the subject, made in [14]. The method proposed there succeeds in finding a Wolfe point, but it suffers from requiring at least two linearizations of the constraints per iteration. The next iterate is indeed obtained by doing a longitudinal search followed by a transversal search. It is by an adequate design of the longitudinal search path that the condition on the reduced gradient can be realized. This implies, however, that the reduced gradient must always be evaluated at some intermediate point. We believe that this extra evaluation is an unacceptable additional cost in most applications. A contribution of the present paper is to show that a technique similar to the one introduced in [14] can also be used, and *makes sense*, even when both searches are done simultaneously along a single path. As a result, only one linearization of the constraints per iteration is necessary *asymptotically* in the new algorithm.

More precisely, we consider the problem of minimizing a smooth function $f$ depending on $n$ variables $x$ lying in an *open* set $\Omega \subset \mathbb{R}^n$. These variables are also forced to satisfy the $m$ ($m < n$) constraints $c(x) = 0$, where $c : \Omega \to \mathbb{R}^m$ is a smooth function. For further reference, we write this problem as follows

$$\begin{cases} \min f(x) \\ c(x) = 0, \quad x \in \Omega. \end{cases} \tag{1.1}$$

When the function $c$ is a submersion on $\Omega$, i.e., when its $m \times n$ Jacobian matrix $A(x) = \nabla c(x)^\top$ has full row rank for all $x$ in $\Omega$, 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 $n - m$. Therefore, problem (1.1) can also be viewed as the problem of minimizing $f$ on the manifold

$$\mathcal{M}_* = \{x \in \Omega : c(x) = 0\}.$$

The algorithm we consider generates a sequence $\{x_k\}_{k \geq 1} \subset \Omega$ of approximations of a solution $x_*$ of (1.1), which, however, are not forced to belong to $\mathcal{M}_*$.

We recall that the *Lagrangian function* of problem (1.1) is the function defined for $(x, \lambda) \in \Omega \times \mathbb{R}^m$ by

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

Its Hessian with respect to $x$ is denoted by $L(x, \lambda)$.

Near a solution, the reduced quasi-Newton method we consider calculates the next iterate $x_{k+1}$ by

$$x_{k+1} = x_k + d_k, \tag{1.2}$$

where $x_k$ is the current iterate and $d_k$ is the solution of the following quadratic program

$$\begin{cases} \min \nabla f(x_k)^\top d + \frac{1}{2} d^\top M_k d \\ c_k + A_k d = 0. \end{cases} \tag{1.3}$$

We usually take the notation $\xi_k$ for the value of a function $\xi(\cdot)$ at $x_k$ and $\xi_*$ for $\xi(x_*)$. Hence $c_k = c(x_k)$ and $A_k = A(x_k)$ in (1.3). This method differs form the well known sequential quadratic programming (SQP) method (see Wilson [34], Han [17], and Pshenichnyi and Danilin [30]) in the sense that the $n \times n$ matrix $M_k$ ignores some part of the Hessian of the Lagrangian (see Murray and Wright [25], and Gabay [12]). We shall be more specific below. Note that our search algorithm can also be used for the reduced quasi-Newton method of Coleman and Conn [5].

To present the ideas without interference with computational issues, it is convenient, like Gabay [11], to make a decomposition of $\mathbb{R}^n$ that is adapted to the problem: see Figure 1. At a point $x \in \Omega$, we suppose given a basis $Z^-(x)$ of the

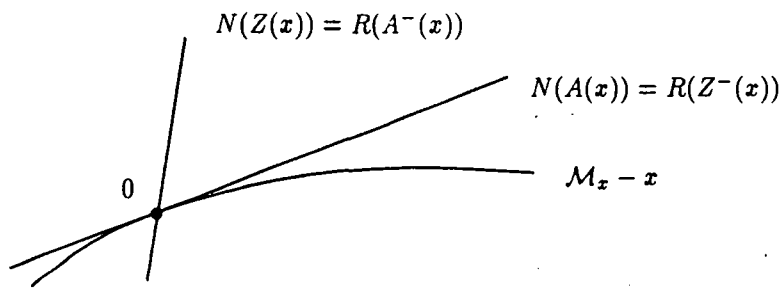


Figure 1: A suitable decomposition of $\mathbb{R}^n$.

null space $N(A(x))$ of $A(x)$, which is also the tangent space to $\mathcal{M}_x$ at $x$. In other words, $Z^-(x)$ is an injective (or full column rank) $n \times (n-m)$ matrix satisfying

$$A(x)Z^-(x) = 0,$$

in $\mathbb{R}^{m \times (n-m)}$. We use the notation $Z^-(x)$ instead of $Z(x)^-$ to indicate that the matrix is the evaluation at $x$ of a map $Z^- : x \mapsto Z^-(x)$. Next, a subspace complementary to $N(A(x))$ is supposed to be given through a right inverse $A^-(x)$ of $A(x)$:

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

in $\mathbb{R}^{m \times m}$. Hence, $A^-(x)$ is $n \times m$ and the range space $R(A^-(x))$ is complementary to $N(A(x))$. Then, there exists a unique $(n-m) \times n$ matrix $Z(x)$, such that

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

in $\mathbb{R}^{(n-m)\times(n-m)}$ and $\mathbb{R}^{(n-m)\times m}$, respectively. To summarize, the matrices $A^-(x)$ and $Z^-(x)$ are injective and right inverses of $A(x)$ and $Z(x)$, respectively. The superscript "$-$" indicates that their use generally involves the solution of some linear system. On the other hand, the matrices $A(x)$ and $Z(x)$ are surjective, i.e., they have full row rank. The following useful identity holds in $\mathbb{R}^{n\times n}$:

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

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

The advantage of the above setting is that it can be adapted to different situations by choosing appropriately the matrices $A^-(x)$ and $Z^-(x)$. For a recent use of this to optimal control, see Kupfer and Sachs [19], and Kunisch and Sachs [18].

The *reduced gradient* is an important tool in equality constrained optimization. It is the vector of $\mathbb{R}^{n-m}$ defined by

$$g(x) = Z^-(x)^\top \nabla f(x).$$

The reduced Hessian of the Lagrangian with respect to the basis $Z^-(x)$ is the matrix of order $n-m$ defined by

$$B(x,\lambda) = Z^-(x)^\top L(x,\lambda) Z^-(x).$$

We can now be more specific on the direction $d_k$ computed by the quadratic program (1.3). First, it is a feature of reduced quasi-Newton methods to force $M_k$ to have the form $M_k = Z_k^\top B_k Z_k$, where $B_k$ is an updated matrix. This makes $B_k$ an approximation of the reduced Hessian of the Lagrangian. Then, decomposing $d_k = A_k^- u_k + Z_k^- v_k$ and using the relations above show that $d_k$ can be written in the form

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

where $t_k = -Z_k^- B_k^{-1} g_k$ is the *tangent* or *longitudinal* part of the step, tangent to the manifold $\mathcal{M}_k$ at $x_k$, and $r_k = -A_k^- c_k$ is the *restoration* or *transversal* part of the step, aiming at reducing the norm of the constraint function $c$.

One of the main concern 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 (1.2)–(1.3). It is also natural since this matrix approximates $B(x,\lambda)$, which is positive semi-definite at the solution. A way of maintaining the positive definiteness of the matrices $B_k$ is to use an update formula allowing to have this property. A typical example is the BFGS formula (see [10, 8]):

$$B_{k+1} = B_k - \frac{B_k \delta_k \delta_k^\top B_k}{\delta_k^\top B_k \delta_k} + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k}. \tag{1.4}$$

This formula requires the use of two vectors $\gamma_k$ and $\delta_k$ 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 $\gamma_k$ and $\delta_k$ satisfy the following condition:

$$\gamma_k^T \delta_k > 0. \tag{1.5}$$

Our piecewise line-search technique can realize this inequality whenever desired.

The local analysis of algorithm (1.2)-(1.3) shows that it is convenient to take for $\gamma_k$ the change in the reduced gradient and for $\delta_k$ the reduced displacement:

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

Other choices are sometimes proposed: see for instance Coleman and Conn [6], and Nocedal and Overton [26]. All of them are asymptotically equivalent to the above choice, which is preferred for its geometrical interpretation (see Section 3) and its appealing simplicity. In these formulæ appears a step-size $\alpha_k > 0$, introduced in Section 3, because the matrices $B_k$ are also updated far from the solution where the algorithm differs from (1.2)-(1.3). 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 (1.5) (see [14]).

Condition (1.5) holds if the search algorithm determines $x_{k+1}$ so that

$$g_{k+1}^T Z_k t_k \geq \omega_2 \, g_k^T Z_k t_k, \tag{1.6}$$

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

$$\Theta_\sigma(x) = f(x) + \sigma \|c(x)\|, \tag{1.7}$$

where $\sigma$ is positive and sufficiently large, and $\|\cdot\|$ denotes a norm in $\mathbb{R}^m$. The decrease in $\Theta_\sigma$ is typically forced by requiring

$$\Theta_\sigma(x_{k+1}) \leq \Theta_\sigma(x_k) + \omega_1 \, \nu_k(\alpha_k), \tag{1.8}$$

where $0 < \omega_1 < 1$ and $\nu_k(\alpha)$ is negative for positive $\alpha$. Note that we shall not need $\omega_1 < \omega_2$ in the PLS algorithm. The difficulty of realizing both (1.6) and (1.8) simultaneously comes from the fact that, unlike what happens for unconstrained problems, the left hand side in (1.6) is not the directional derivative of $\Theta_\sigma$ at $x_{k+1}$ along commonly used search directions such as $t_k$ or $d_k$. It is in fact the directional derivative $\Theta'_\sigma(x_{k+1}; Z_{k+1}^- Z_k t_k)$. This suggests to make a reorientation of the search direction when (1.6) does not hold, using the new basis $Z_{k+1}^-$ although keeping the same reduced tangent direction $Z_k t_k$. This is the idea underlying the search algorithm

proposed in [14], in which the search path has only longitudinal components, i.e., components in the range space of the matrices $Z^-(x_k^i)$, where $x_k^i$ $(i = 0, \ldots, i_k - 1)$ are intermediate points. Here we show how to implement the same idea for paths having also transversal components, i.e., components in the range space of $A^-(x_k^i)$. The analysis is long but it results in a quite simple algorithm, which can be described as follows: at each iteration of the search algorithm, condition (1.8) is realized and next condition (1.6) is tested. If the latter holds, the algorithm terminates with a suitable point. Otherwise, a new search direction is defined with the same matrix *and the same reduced gradient* as the previous direction. A new iteration is then started.

Other papers 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 SQP method, in which approximations of the full Hessian of the Lagrangian are generated. In this case $\gamma_k$ is usually $\gamma_k^0$, the change in the gradient of the Lagrangian, and $\delta_k$ is the step. The simplest idea, due to Powell [27], is to take for $\gamma_k$ a convex combination of $\gamma_k^0$ and $B_k\delta_k$ such that (1.5) holds. In our opinion, this method lacks a strong geometrical interpretation and, according to Powell [29], it may lead to ill-conditioning when the problem is difficult to solve. Due to its undeniable simplicity, however, it is the most widely implemented technique. Another promising idea, proposed by Han [17] and Tapia [32], and subsequently explored by Tapia [33] and Byrd, Tapia, and Zhang [4], is to generate approximations of the Hessian of the augmented Lagrangian, which is positive definite at the solution when the augmented 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, Fenyes [9], and Coleman and Fenyes [7] update separately approximations of $Z_*^{-\mathsf{T}} L(x_*, \lambda_*) Z_*^-$ and $A_*^{-\mathsf{T}} L(x_*, \lambda_*) Z_*^-$, maintaining positive definite the former submatrix. Let us also mention the work of Biegler, Nocedal, and Schmid [2], which is somehow related to this subject.

The paper is organized as follows. In Section 2, we give some more notation. In Section 3, the search path is introduced and its meaning is discussed. Also, conditions to have finite termination of the search algorithm are given. Section 4 contains a global convergence result and finally some numerical experiment is reported in Section 5.

## 2 Some notation

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.,

$$\begin{cases} c(x_*) = 0 \\ \nabla f(x_*) + A_*^{\mathsf{T}} \lambda_* = 0 \end{cases}$$

and $h^T L(x_*, \lambda_*)h > 0$, for all nonzero $h \in N(A_*)$. We note $B_* = B(x_*, \lambda_*) = Z_*^{-T} L(x_*, \lambda_*) Z_*^-$, which is then positive definite, and

$$\lambda(x) = -A^-(x)^T \nabla f(x).$$

Hence $\lambda(x_*) = \lambda_*$, the Lagrange multiplier at the solution.

We recall that we use the penalty function $\Theta_{\sigma_k}$ defined in (1.7) to globalize the local method (1.2)–(1.3). The *penalty parameter* $\sigma_k$ is updated to satisfy at each iteration

$$\sigma_k \geq \|\lambda_k\|_D + \bar{\sigma}, \tag{2.1}$$

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 (1.7). It is defined by

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

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

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

Finally, $\xi'(x; d)$ denotes the directional derivative of a function $\xi$ along the direction $d$.

# 3 The search algorithm

In unconstrained optimization, the path $0 \leq \alpha \mapsto p_k(\alpha)$ 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:

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

$$g(p_k(\alpha))^T Z_k t_k \geq \omega_2 g_k^T Z_k t_k, \tag{3.2}$$

for some $\alpha > 0$. In (3.1) and (3.2), the constants $\omega_1$ and $\omega_2$ are chosen in $(0, 1)$, and $\alpha \mapsto \nu_k(\alpha)$ is a function forcing the decrease of $\Theta_{\sigma_k}$ by the properties

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

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

In our proposal, the description of the search path is difficult, because it depends on some intermediate step-sizes. This is because, in the point of view taken here,

a reorientation of the search path is necessary at some unsuccessful step-sizes $\alpha_k^i$, $i = 1, \ldots, i_k - 1$. Furthermore, condition (3.1) also depends on the intermediate step-sizes $\alpha_k^i$ through the function $\nu_k$, which cannot be given before the search is completed. For these reasons, we have to specify simultaneously the way the search path is designed and the function $\nu_k$.

The algorithm we discuss has some similarities with the one given in [14], 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 [10] and Lemaréchal [20] 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 [23] 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 it is realistic to try to realize conditions (3.1) and (3.2) simultaneously. We also argue to what extent this attempt makes sense. This discussion aims at giving general ideas without being too precise on the hypotheses.

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

$$\begin{cases} \bar{p}_k'(\alpha) = Z^-(\bar{p}_k(\alpha)) Z_k t_k \\ \bar{p}_k(0) = x_k. \end{cases} \tag{3.3}$$

This trajectory belongs to the manifold $\mathcal{M}_k$, because, multiplying the first equation in (3.3) by $A(\bar{p}_k(\alpha))$ gives $(c \circ \bar{p}_k)'(\alpha) = 0$, meaning that $c$ remains constant along the path. As quoted in [14], if this path is defined for sufficiently large $\alpha$ 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):

$$\Theta_{\sigma_k}(\bar{p}_k(\bar{\alpha}_k)) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \, \bar{\alpha}_k \, g_k^\top Z_k t_k, \tag{3.4}$$

$$g(\bar{p}_k(\bar{\alpha}_k))^\top Z_k t_k \geq \omega_2 \, g_k^\top Z_k t_k. \tag{3.5}$$

This can be seen by considering the standard Wolfe [35, 36] 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, 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 (3.5). Note that condition (3.4) has the form (3.1) with a linear function $\nu_k$.

Locally, the search along $\bar{p}_k$ has also the following geometrical interpretation, illustrated in Figure 2. If $\psi : U \subset \mathbb{R}^{n-m} \to \mathcal{M}_k \subset \mathbb{R}^n$ is a parametric representation of $\mathcal{M}_k$ around $x_k$ such that $0 \in U$, $\psi(0) = x_k$, and $\psi'(u) = Z^-(\psi(u))$ for all $u \in U$, it is easy to see that $\bar{p}_k(\alpha) = \psi(\alpha Z_k t_k)$. Then, $(f \circ \psi)(\alpha Z_k t_k) = (\Theta_{\sigma_k} \circ \bar{p}_k)(\alpha) - \sigma_k \|c_k\|$ and $\nabla(f \circ \psi)(\alpha Z_k t_k) = g(\bar{p}_k(\alpha))$, so that the search to realize (3.4)–(3.5) can now

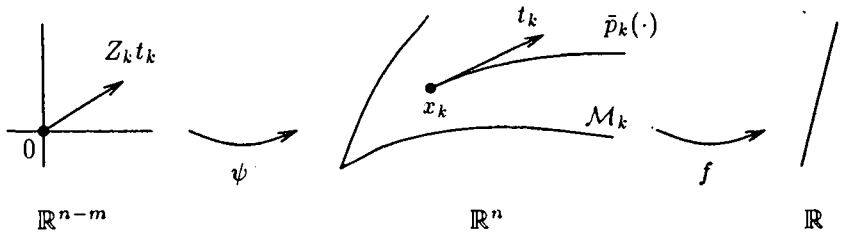


Figure 2: An interpretation of the longitudinal guiding path.

be seen as a standard Wolfe search on the function $f \circ \psi$ 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 $\bar{\delta}_k = \bar{\alpha}_k Z_k t_k$.

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*, 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 (3.4)–(3.5), but easier to compute than $\bar{p}_k(\cdot)$; see [14].

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. We proceed step by step and begin by showing why such a search can succeed.

First, let us consider the smooth function

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

Recall that $g'(x_*) = Z_*^{-\top} L_*$ (see Stoer [31] or Nocedal and Overton [26]). Then, when $B_*$ is nonsingular, $F'(x_*)$ is also nonsingular and there exists an open subset $\Omega_0$ of $\Omega$ such that $F : \Omega_0 \to F(\Omega_0)$ is a diffeomorphism. There is no restriction in supposing that $F(\Omega_0)$ 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. For a fixed point $x$ in $\Omega_0$, consider the map

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

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 $y$). To see that this set 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 $\Omega_0$, $\mathcal{M}_x \cap \mathcal{N}_y \cap \Omega_0 = F^{-1}((c(x), g(y))) \cap \Omega_0$ is a singleton. As we see, $\mu_x$ 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 $\mathcal{N}_y$.

Now, suppose that we have built a search path $\alpha \mapsto \tilde{p}_k(\alpha)$, with longitudinal and transversal components, such that its image by $\mu_{x_k}$ lies in $\bar{\mathcal{P}}_k$:

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

If this path exists for sufficiently large $\alpha$, 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 supposes that the path $\tilde{p}_k$ does not blow up for finite longitudinal displacement. Using (3.5), we obtain

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

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.* Secondly, 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 $\bar{\delta}_k = \bar{\alpha}_k Z_k t_k$.

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

$$g(\tilde{p}_k(\alpha)) = g(\bar{p}_k(\alpha)), \quad \text{for } \alpha \geq 0, \tag{3.6}$$

as long as both curves $\tilde{p}_k(\cdot)$ and $\bar{p}_k(\cdot)$ exist. This can be achieved when the maps $Z^-(\cdot)$ and $A^-(\cdot)$ are chosen such that

$$Z = g', \quad \text{on } \Omega, \tag{3.7}$$

and when $\tilde{p}_k(\cdot)$ is defined as a solution of the following differential equation

$$\begin{cases} \tilde{p}_k'(\alpha) = Z^-(\tilde{p}_k(\alpha)) Z_k t_k - A^-(\tilde{p}_k(\alpha)) c(\tilde{p}_k(\alpha)) \\ \tilde{p}_k(0) = x_k. \end{cases} \tag{3.8}$$

Indeed, multiplying the first equation in (3.3) by $Z(\bar{p}_k(\alpha))$ and the first equation in (3.8) by $Z(\tilde{p}_k(\alpha))$, and using (3.7), we get

$$(g \circ \bar{p}_k)'(\alpha) = Z_k t_k = (g \circ \tilde{p}_k)'(\alpha).$$

Hence $g \circ \bar{p}_k$ and $g \circ \tilde{p}_k$ satisfy the same differential equation with the same initial condition $g_k$ at $\alpha = 0$. The identity (3.6) follows. We call the path defined by $\tilde{p}_k$, solution of (3.8), the *bicomponent guiding path.*

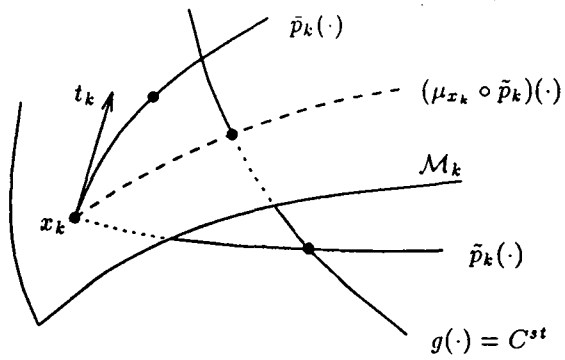


Figure 3: The bicomponent guiding path.

Clearly, it is unrealistic to ask the user to realize (3.7), because $g'$ is a second order information that is out of reach in the quasi-Newton framework. Also, we shall not assume that (3.7) holds and therefore (3.6) may not hold either, even at the first order. Figure 3 represents a still rather optimistic situation without (3.6). As we shall see below, no matter the realization of (3.6), one can satisfy the reduced Wolfe conditions (3.1)–(3.2) along the path $\tilde{p}_k$ by choosing the function $\nu_k$ appropriately. Of course, an update of the matrix without (3.6) may not be safe. Normally, it should be the role of the update criterion to detect the situations where (3.7) is not too much violated. The update criteria introduced by Nocedal and Overton [26], and Gilbert [13] are based, however, on a condition different from (3.7).

Let us now show how to realize (3.1)–(3.2) without (3.6). The case when $g_k = 0$ is easy, since (3.2) readily holds ($t_k = 0$). So, suppose that $g_k \neq 0$ and choose as function $\nu_k$ in (3.1) the solution of the following differential equation:

$$\begin{cases} \nu_k'(\alpha) = (\Theta_{\sigma_k} \circ \tilde{p}_k)'(\alpha) \\ \nu_k(0) = 0. \end{cases} \tag{3.9}$$

Clearly $\nu_k(\alpha) = \Theta_{\sigma_k}(\tilde{p}_k(\alpha)) - \Theta_{\sigma_k}(x_k)$ and (3.1) is equivalent to requiring that $\Theta_{\sigma_k}(\tilde{p}_k(\alpha)) \leq \Theta_{\sigma_k}(x_k)$, which is not very demanding. Let us show that, with this choice of $\nu_k$, if $\Theta_{\sigma_k}$ is bounded from below along the path $0 < \alpha \mapsto \tilde{p}_k(\alpha)$ and if $\sigma_k$ is sufficiently large, then (3.1) and (3.2) must be satisfied for some finite $\alpha > 0$. Indeed, observe that, as long as (3.2) is not verified (which is true for small positive $\alpha$) and $\sigma_k \geq \|\lambda(\tilde{p}_k(\alpha))\|_D$, one has

$$\begin{aligned} &\Theta_{\sigma_k}(\tilde{p}_k(\alpha)) - \Theta_{\sigma_k}(x_k) \\ &\quad = \int_0^\alpha (\Theta_{\sigma_k} \circ \tilde{p}_k)'(\theta)\, \mathrm{d}\theta \end{aligned}$$

$$
\begin{aligned}
&= \int_0^\alpha \Theta'_{\sigma_k}\Big(\tilde{p}_k(\theta); Z^-(\tilde{p}_k(\theta))Z_k t_k - A^-(\tilde{p}_k(\theta))c(\tilde{p}_k(\theta))\Big)\,\mathrm{d}\theta \\
&= \int_0^\alpha \Big(g(\tilde{p}_k(\theta))^\top Z_k t_k + \lambda(\tilde{p}_k(\theta))^\top c(\tilde{p}_k(\theta)) - \sigma_k \|c(\tilde{p}_k(\theta))\|\Big)\,\mathrm{d}\theta \\
&\leq \omega_2\, \alpha\, g_k^\top Z_k t_k.
\end{aligned}
$$

Therefore, (3.1) is trivially satisfied for these $\alpha$'s. 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, this situation cannot continue indefinitely. At the first step-size $\alpha$ at which (3.2) is verified, by continuity, (3.1) is still verified. Note that the inequality $\omega_1 < \omega_2$ was not needed in this reasoning.

We are now ready to describe the actual search path, which may be seen as an explicit Euler approximation of the solution of (3.8) with well chosen discretization points. Similarly, the actual function $\nu_k$ is a piecewise linear approximation of the function defined in (3.9) with the same discretization points. A successful idea is to introduce a discretization point $\alpha_k^i$ 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 $x_k \in \Omega$ and is not stationary: $\|g_k\| + \|c_k\| \neq 0$. Let the constants $\omega_1$ and $\omega_2$ be given in $(0,1)$.

The search algorithm is iterative and generates, for $i = 0, \ldots, i_k - 1$, intermediate step-size candidates $\alpha_k^i$, points $x_k^i$, descent directions $d_k^i$ of $\Theta_{\sigma_k}$ at $x_k^i$, and piecewise linear search paths $p_k^i$ and forcing functions $\nu_k^i$. This latter function, playing the role of $\nu_k$ in (3.1), is not necessarily continuous. The following conditions will hold for $i = 1, \ldots, i_k - 1$:

$$x_k^i \in \Omega \tag{3.10a}$$

$$\Theta_{\sigma_k}(x_k^i) \leq \Theta_{\sigma_k}(x_k) + \omega_1\, \nu_k^{i-1}(\alpha_k^i) \tag{3.10b}$$

$$g(x_k^i)^\top Z_k t_k < \omega_2\, g_k^\top Z_k t_k \tag{3.10c}$$

$$\sigma_k \geq \|\lambda(x_k^i)\|_D + \bar{\sigma}. \tag{3.10d}$$

Inequality (3.10b) means that descent is forced at each iteration, while (3.10c) means that the curvature condition does not hold. Note that (3.10c) implies that $x_k^i$ is not stationary.

At the beginning, the iteration index $i = 0$, $\alpha_k^0 = 0$, and $x_k^0 = x_k$. To initialize the recurrence, we define $p_k^{-1}(\alpha) = x_k$ and $\nu_k^{-1}(\alpha) = 0$ for all $\alpha$. It is also assumed that $\sigma_k \geq \|\lambda_k\|_D + \bar{\sigma}$. Then, (3.10abd) clearly holds for $i = 0$. Stage $i$ ($i \geq 0$) of the search is divided in several steps.

**Step 1.** The following direction is introduced:

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

where $\tau_k^i > 0$ scales the tangent component of the direction. This scaling factor is introduced for reasons of efficiency, discussed in Section 3.6. When $i = 0$ (initially) and $\tau_k^0 = 1$, $d_k^0$ is the reduced SQP direction $d_k$, which is tangent to $\bar{p}_k$ at 0. For $i \geq 1$, the direction comes from the discretization of (3.8): 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 iteration to the other. Observe that $d_k^i$ is a descent direction of $\Theta_{\sigma_k}$ at $x_k^i$, since

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

is negative, when (3.10c) and (3.10d) hold.

The search path is then modified (when $i \geq 1$) such that, after $x_k^i$, the search continues in the direction $d_k^i$:

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

The modification of $\nu_k^{i-1}$ is less easy to describe since we would like to include two natural possibilities. They correspond to the setting of a parameter $\rho_k^i$ to 0 or 1 below. So let $\rho_k^i$ be chosen in $[0, 1]$. We also introduce the notion of ***total decrease*** of $\Theta_{\sigma_k}$ at $x_k^i$, which is the positive quantity

$$T_k^i = \Theta_{\sigma_k}(x_k) - \Theta_{\sigma_k}(x_k^i). \tag{3.11}$$

Then, $\nu_k^i$ is defined by

$$\nu_k^i(\alpha) = \begin{cases} \nu_k^{i-1}(\alpha) & \text{for } 0 \leq \alpha < \alpha_k^i \\ (1 - \rho_k^i)\nu_k^{i-1}(\alpha_k^i) + \rho_k^i(-T_k^i/\omega_1) + (\alpha - \alpha_k^i)\Theta'_{\sigma_k}(x_k^i; d_k^i) & \text{for } \alpha \geq \alpha_k^i. \end{cases} \tag{3.12}$$

When $\rho_k^i = 0$, $\nu_k^i$ 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 below. It is also more demanding, since $\nu_k^i$ is more negative (use (3.10b)).

**Step 2.** A step-size candidate $\alpha_k^{i+1} > \alpha_k^i$ is determined such that

$$x_k^{i+1} = p_k^i(\alpha_k^{i+1}) \tag{3.13}$$

is in $\Omega$ and the descent condition

$$\Theta_{\sigma_k}(p_k^i(\alpha))) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \nu_k^i(\alpha) \tag{3.14}$$

holds for $\alpha = \alpha_k^{i+1}$. It is standard to verify that, with conditions (3.10a), (3.10b), and $\omega_1 \in (0, 1)$, such a step-size always exists.

**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 updating the reduced matrix correctly. This implies that if an update criterion judges that an update is not suitable, the search is interrupted here and takes

$$i_k = 1, \quad \alpha_k = \alpha_k^1, \quad x_{k+1} = x_k^1, \quad p_k = p_k^0, \quad \text{and} \quad \nu_k = \nu_k^0.$$

**Step 4.** Next, the following curvature condition is tested:

$$g(x_k^{i+1})^\top Z_k t_k \geq \omega_2 \, g_k^\top Z_k t_k. \tag{3.15}$$

If the inequality holds, the search terminates and takes

$$i_k = i + 1, \quad \alpha_k = \alpha_k^{i_k}, \quad x_{k+1} = x_k^{i_k}, \quad p_k = p_k^i, \quad \text{and} \quad \nu_k = \nu_k^i.$$

From (3.13), (3.14) with $\alpha = \alpha_k^{i+1}$, and (3.15), the point $x_{k+1}$ is in $\Omega$ and satisfies the reduced Wolfe conditions (3.1)–(3.2).

**Step 5.** If (3.15) 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 the following inequality holds:

$$\sigma_k \geq \|\lambda(x_k^{i+1})\|_D + \bar{\sigma}. \tag{3.16}$$

If such is the case, all the conditions in (3.10) hold and an new iteration can start after having increased $i$ by one. Otherwise, the search is interrupted.

## 3.3 Comments

To summarize, there are three facts that can interrupt the search prematurely: either (1) the update criterion does not hold in Step 3 when $\alpha_k^1$ has been determined in Step 2, or (2) the penalty parameter $\sigma_k$ is not large enough to guarantee that the next search direction is a descent direction of $\Theta_{\sigma_k}$, or (3) the conditions (3.1)–(3.2) are satisfied in Step 4. We shall show in Section 4 that usually 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 $\tilde{p}_k$, obtained by an explicit Euler discretization of the differential equation (3.8) at the step-sizes $\alpha_k^i$. If furthermore $\rho_k^i = 0$ for all $i$, $\nu_k$ can also be viewed as a discretization of the function $\nu_k$ defined by (3.9) 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 the curvature condition does not hold. When $\rho_k^1 = 1$ in the definition of $\nu_k^1$ above, inequality (3.14) 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 $\tilde{d}_k^1$ 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 $Z(x_k^i)$ are good in the sense (3.7), the search continues to explore $(f \circ \psi)$ along the reduced direction $Z_k t_k$ (the meaning of $\psi$ is given in Figure 2), giving a sense in the update of $B_k$ with the vectors

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

On the other hand, keeping the previous value of the reduced gradient should not deteriorate the efficiency of the algorithm since the fact that (3.15) is not verified implies that $Z^-(x_k^{i+1})Z_k t_k$ is still a good descent direction of $\Theta_{\sigma_k}$ (provided $\omega_2$ is close to 1). We shall see in Section 3.6, however, that this can sometimes have unfortunate consequences.

When $\rho_k^i = 1$ for all $i$, our search algorithm applied to unconstrained problems ($c(x_k^i) = 0$ for all $i$) is related to the method of Moré and Sorensen [23] (see also Moré and Thuente [24, 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 $\nu_k^i$ 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 $\Theta_{\sigma_k}$ at $x_k^{i+1}$ as the positive quantity

$$F_k^{i+1} = -\omega_1 \sum_{l=0}^{i} (\alpha_k^{l+1} - \alpha_k^l)\Theta'_{\sigma_k}(x_k^l; d_k^l). \tag{3.17}$$

Using (3.10b) and the definition (3.12) of $\nu_k^i$, we get for $i \geq 1$:

$$F_k^i \leq -\omega_1\, \nu_k^{i-1}(\alpha_k^i) \leq T_k^i, \tag{3.18}$$

where $T_k^i$ is the total decrease of $\Theta_{\sigma_k}$, defined in (3.11).

## 3.4 Determination of the step-size candidates by "forth-backtracking"

When a step-size candidate $\alpha_k^i$ is not accepted, because inequality (3.15) does not hold, one has to determine the next tangent scaling factor $\tau_k^i > 0$ and the next step-size candidate $\alpha_k^{i+1}$ such that

$$\alpha_k^{i+1} > \alpha_k^i, \quad x_k^{i+1} = p_k^i(\alpha_k^{i+1}) \in \Omega, \quad \text{and} \quad (3.14) \text{ holds with } \alpha = \alpha_k^{i+1}.$$

This cannot be done in an uncontrolled manner. In particular, $\tau_k^i$ cannot be arbitrarily small or large, and $\alpha_k^{i+1}$ cannot be chosen too close to $\alpha_k^i$. In this section, we describe a method for determining $\alpha_k^{i+1}$ 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 $\Omega$ or (3.14) 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 $\alpha_k^i$ when $j \to \infty$, $p_k^i(\alpha_k^{i,j})$ will be in $\Omega$ and (3.14) with $\alpha = \alpha_k^{i,j}$ will hold for some finite index $j$. We note $j_i$ 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 $\alpha_k^i$: the closer $\alpha_k^{i+1}$ is to $\alpha_k^i$, the larger $j_i$ must be. The rigorous form of our assumptions follows.

**Assumptions 3.1.** 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 $\alpha_k^i$,

($iii$) 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 $\bar{\alpha}_k^j \neq \bar{\alpha}_k$,
- ($c$) the sequence $\{\bar{\alpha}_k^j\}_{j\geq 1}$ converges to $\bar{\alpha}_k$.

An easy way of satisfying Assumptions 3.1, while using its favorite extrapolation and interpolation formulae, is to require that some safeguard rules be satisfied. Here is an example of rules that guarantee Assumptions 3.1.

**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 3.1 ($ii$), ($iii$–$a$), and ($iii$–$c$) are satisfied if these rules are used. Observe that, for $i \geq 1$ and $j \geq 1$,

$$\alpha_k^i < \alpha_k^{i,j} \leq \alpha_k^i + (1-\varepsilon_I)^{j-1}(\alpha_k^{i,1} - \alpha_k^i). \tag{3.19}$$

Therefore Assumption 3.1 $(ii)$ 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

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

Now, because $\varepsilon_I < 1$, we obtain $j_i \geq j'$, for all $i \geq i'$, which is Assumption 3.1 $(iii{-}a)$. Finally, Assumption 3.1 $(iii{-}c)$ is also guaranteed by the above rules as this can be seen by taking the limit on $i$ and then on $j$ in (3.19).

We have not discussed the case of Assumption 3.1 $(iii{-}b)$, but it can also easily be satisfied, for example by taking for $i \geq 0$

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

which is compatible with the safeguard rules above. More appropriate choices would use the known values of $\Theta_{\sigma_k}$ and its directional derivatives.

### 3.5 Finite termination of the search algorithm

The next proposition gives conditions that ensure the finite termination of the piecewise line-search (PLS) algorithm described in Sections 3.2 and 3.4 at a point $x_{k+1}$ satisfying

$$\Theta_{\sigma_k}(x_{k+1}) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \, \nu_k(\alpha_k), \tag{3.20}$$

$$g_{k+1}^T Z_k t_k \geq \omega_2 \, g_k^T Z_k t_k, \tag{3.21}$$

where the function $\nu_k$ is defined recursively in Step 1 of the algorithm. Recall that the search path $p_k$ is also defined iteratively in Step 1 of the algorithm.

**Proposition 3.2.** *Suppose that $f$ and $c$ are differentiable on $\Omega$, $c$ is a submersion on $\Omega$, and the decomposition of $\mathbb{R}^n$ described in Section 1 is made with maps $Z^-$ and $A^-$ that are bounded on $\Omega$. Let $x_k$ be a point in $\Omega$ and $B_k$ be a symmetric positive definite matrix of order $n - m$. Suppose that the penalty factor $\sigma_k$ in (1.7) satisfies (2.1). Then, if the PLS algorithm described in Sections 3.2 and 3.4, with Assumptions 3.1, $\omega_1 \in (0,1)$, and $\omega_2 > 0$, is applied from $x_k$, one of the following situations occurs:*

*(i) the algorithm terminates after a finite number of stages with a step-size $\alpha_k$, a point $x_{k+1} \in \Omega$, and a function $\nu_k$ satisfying conditions (3.20) and (3.21);*

(*ii*) *the algorithm terminates prematurely with a step-size* $\alpha_k$, *a point* $x_{k+1} \in \Omega$, *and a function* $\nu_k$ *satisfying* (3.20) *only, because the update criterion does not hold at* $x_k^1$ *or because* (3.16) *fails at* $x_k^{i+1} = x_{k+1}$;

(*iii*) *the algorithm builds a sequence of points* $\{x_k^i\}_{i\geq 1}$ *in* $\Omega$ *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* $\Omega$.

**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 and that a sequence $\{x_k^i\}_{i\geq 1}$ is built in $\Omega$. This can only occur when $g_k \neq 0$, since (3.21) is always satisfied when $t_k = 0$ and (3.20) 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 $\Omega$. Of course, conditions (3.10) 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 (3.18), 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 bounded below.

On the other hand, using the definition of $F_k^{i+1}$, (2.1), (3.10d), $g_k^\top Z_k t_k \leq 0$, and (3.10c), 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

$$\sum_{i\geq 0} \tau_k^i \, (\alpha_k^{i+1} - \alpha_k^i) < +\infty. \tag{3.22}$$

As $\{\tau_k^i\}_{i\geq 0}$ is bounded away from 0 [Assumption 3.1.(*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 $F_k^{i+1}$, $g_k^\top Z_k t_k \leq 0$, (3.10c), and (3.10d), 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

$$\sum_{i\geq 0} (\alpha_k^{i+1} - \alpha_k^i) \, \|c(x_k^i)\| < +\infty. \tag{3.23}$$

Now, by definition of $x_k^i$

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

Using the boundedness of $Z^-(\cdot)$ and $A^-(\cdot)$ on $\Omega$, (3.22), and (3.23), we see that the series in the right hand side is absolutely convergent when $i \to \infty$. 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 $\Omega$, hence $\bar{x}_k \in \Omega$.

This implies the following convergence when $i \to \infty$:

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

Furthermore, since $\{\tau_k^i\}$ and $\{1/\tau_k^i\}$ are bounded, there is some $\bar{\tau}_k > 0$ and a subsequence $\mathcal{I} \in \mathbb{N}$ such that for $i \to \infty$, $i \in \mathcal{I}$, we have

$$d_k^i \to \bar{d}_k = \bar{\tau}_k Z^-(\bar{x}_k) Z_k t_k - A^-(\bar{x}_k) c(\bar{x}_k),$$

$$\Theta'_{\sigma_k}(x_k^i; d_k^i) \to \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k).$$

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 3.1 $(iii\text{-}b)$, the sequence $\{\alpha_k^{i,j}\}_{i \geq 1}$ converge to a step-size $\bar{\alpha}_k^{\cdot,j} \neq \bar{\alpha}_k$. Therefore, for any $j \geq 1$

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

Now, for fixed $j \geq 1$, Assumption 3.1 $(iii\text{-}a)$ 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 (3.14) 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 (3.18) and the results of Step 2, we obtain

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

Hence

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

Because $\bar{x}_k \in \Omega$, taking the limit in this implication when $j$ tends to infinity gives with Assumption 3.1 $(iii\text{-}c)$: $\Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) \geq \omega_1 \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k)$. 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 (3.10d), $g(\bar{x}_k)^\top Z_k t_k \leq \omega_2 \, g_k^\top Z_k t_k$ from the limit in (3.10c), $\bar{\tau}_k \omega_2 > 0$, 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. □

### 3.6 Escaping from the piecewise line-search more quickly

The use of the PLS described in Sections 3.2 and 3.4 is not always without inconvenient. In fact, it is easy to trap the algorithm in a situation where its behavior is very poor. Such a situation may happen when the path $\mu_{x_k} \circ p_k$ is not a descent path for $f$. Then, the search algorithm may take a large amount of inner iterations to find a generalized Wolfe point and the vectors $\gamma_k$ and $\delta_k$ may be erroneous.

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

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

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 $\mathbb{R}^2$

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

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 part 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 4.

Now, suppose that the current iterate $x_k$ has coordinates $(-1 - \epsilon, 1)$, with $\epsilon > 0$, and 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 (3.1). When $\epsilon = 0.5$ the search algorithm requires 5 inner iterations. The intermediates points $\{x_k^i\}_{i=1}^5$ are represented in Figure 4. By decreasing $\epsilon > 0$, one can obtain as many inner iterations as desired. For example, 21 inner iterations are necessary for $\epsilon = 0.1$ and 2001 for $\epsilon = 10^{-3}$! The reason is that when $\epsilon$ decreases, $x_k$ is closer to the manifold $g(\cdot) = 0$ and the reduced direction is smaller. Because the iterates go

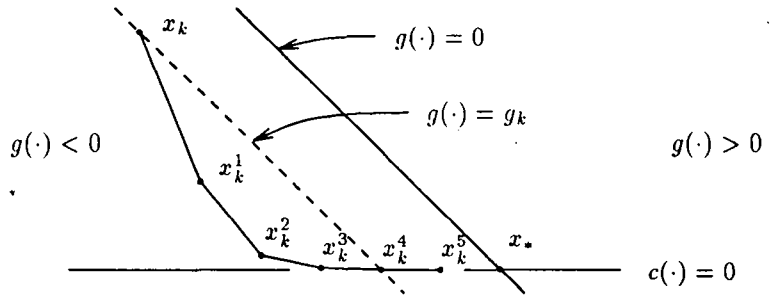


Figure 4: A difficult case for the PLS.

rapidly close to the constraint manifold where the reduced gradient is much more negative than at $x_k$ and because the reduced gradient is not re-evaluated at the intermediate points, the algorithm needs more and more inner iterations to cross the manifold $g(\cdot) = g_k$ (represented by a dashed line in Figure 4), beyond which the curvature condition (3.2) can be satisfied. If the search path is mapped by $\mu_{x_k}$ on the line $c(\cdot) = c_k$, it is clear from Figure 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 unfortunate behavior is, once again, that reduced methods have no information on the tangent space to the manifold of constant reduced gradient.

Normally, an update criterion should take care of the detection of a situation in which the reduced tangent direction is small with respect to the transversal direction and should suggest not updating the matrix and doing a simple backtracking search instead of a PLS in such a case. Now, it does not hurt if the PLS algorithm has a mechanism that faces the above situation with more success. For this we propose to replace the curvature condition (3.21) by

$$g_{k+1}^T Z_k t_k \geq \omega_2 \min_{0 \leq i < i_k} g(x_k^i)^T Z_k t_k. \tag{3.24}$$

The PLS algorithm with this new condition is said to be "with escape" because this inequality provides an escape way to terminate the search more quickly. It is denoted by "PLS-esc" below. Since inequality (3.24) is less restrictive than (3.21), it is clear that PLS-esc terminates more quickly than PLS. In particular, it still has the finite termination property of Proposition 3.2. Questions concerning the global convergence of the algorithm with PLS and PLS-esc 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. Finally, strategies adapting the tangent scaling factor $\tau_k^i$, which is always set to 1 in the runs above, by a quadratic extrapolation formula, using $g(x_k^{i-1})^T Z_k t_k$ and $g(x_k^i)^T Z_k t_k$, can also

improve the search algorithm. In the example above, the run with $\epsilon = 0.5$, 0.1 and $10^{-3}$ terminates in 3, 4, and 5 inner iterations, respectively.

With PLS-esc, the vectors $\gamma_k$ and $\delta_k$ used to update $B_k$ have to be modified. If $l_k$ denotes the largest index for which the minimum in (3.24) is reached, then it is appropriate to take

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

With this choice, $\gamma_k^T \delta_k > 0$. Note also that these vectors will usually put better information into the matrix $B_{k+1}$, because the new value of $\delta_k$ is generally closer to the reduced step from $x_k^{l_k}$ to $x_{k+1}$ than the previous value of $\delta_k$ is close to the reduced step from $x_k$ to $x_{k+1}$. This remark particularly applies to the example above. From Figure 4, we have $\gamma_k^{\text{PLS}} = g(x_k^5) - g_k \approx g(x_k^5) - g(x_k^4) \approx g(x_k^3) - g(x_k^2) = \gamma_k^{\text{PLS-esc}}$. But $\delta_k^{\text{PLS}} = 5\delta_k^{\text{PLS-esc}}$ and it is clear that $\delta_k^{\text{PLS-esc}}$ corresponds better to $\gamma_k^{\text{PLS}} \approx \gamma_k^{\text{PLS-esc}}$ than $\delta_k^{\text{PLS}}$.

# 4 Convergence result

In this section, we show that the PLS method of Section 3 provides the 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 [3] 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. A possibility is to use the BFGS update formula (1.4), which is always well defined when the PLS succeeds. There is still another facet of the algorithm that must be clarified, which is how the penalty parameter $\sigma_k$ is updated. We choose a rule such that ($\bar{\sigma} > 0$):

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

Many rules can satisfy these conditions. For example, Mayne and Polak [22] suggest to take ($\tilde{\sigma} > 1$):

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

We can now outline the algorithm, whose convergence is analyzed below. At the beginning, the iteration index $k$ is set to 1 and the constants $\omega_1$ and $\omega_2$ 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 (3.20) and possibly (3.21) hold. Then, the matrix $B_k$ is updated, provided the PLS algorithm has not been interrupted prematurely by an update criterion or the failure of (3.16). Finally, the penalty parameter is updated according to the rules (4.1).

In unconstrained optimization, the curvature condition corresponding to (3.21) prevents the step-size from being too small, which is important for the global convergence of the algorithm. In constrained problems, this is not necessary the case, because condition (3.21) ignores the transversal component of the search path. For example, when the objective function $f$ is constant the reduced gradient vanishes and (3.21) is satisfied for any step-sizes, independently of the form of the search path. Therefore, something has to be done so that the first step-size candidate $\alpha_k^1$ ($\leq \alpha_k$) will not be too small. For the same reason, the first tangent scaling factor $\tau_k^0$ 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 $\alpha_k^1$ 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 3.1 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 $\alpha_k^1$ 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 $\Omega$. This implicitly supposes that the PLS algorithm never cycles: situation ($i$) or ($ii$) of Proposition 3.2 occurs at each iteration. We denote by $\mathrm{dist}(x, \Omega^c)$ the Euclidean distance between a point $x$ and the complementary set of $\Omega$.

**Proposition 4.2.** *Suppose that $f$ and $c$ are differentiable on $\Omega$ with Lipschitz continuous derivatives, that $c$ is a submersion on $\Omega$, that the map $A^-$ is continuous and bounded on $\Omega$, and that $Z^-$ is bounded on $\Omega$. Suppose also that the algorithm for solving problem* (1.1) *outlined above generates a sequence $\{x_k\}$ in $\Omega$ by the PLS*

*method with Assumptions* 3.1 *and* 4.1, *and the constants* $\omega_1$ *and* $\omega_2$ *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* $\Omega$,

(*ii*) $\sigma_k$ *is modified finitely often and one of the following situations occurs:*

  (*a*) $g_k \to 0$ *and* $c_k \to 0$,

  (*b*) $\Theta_{\sigma_k}(x_k) \to -\infty$,

  (*c*) $\mathrm{dist}(x_k, \Omega^c) \to 0$, *for some subsequence of indices* $k \to \infty$.

**Proof.** Consider first situation (*i*). Let $\mathcal{K}$ be the subsequence of indices $\{k : \sigma_k \neq \sigma_{k-1},\ k \geq k_1\}$. 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 $\{\|\lambda_k\|_D\}_{k\in\mathcal{K}}$ tends to $\infty$. Then, by continuity of $x \mapsto \lambda(x)$ on $\Omega$, $\{x_k : \sigma_k \neq \sigma_{k-1}\}$ has no accumulation point in $\Omega$.

Suppose now that $\{\sigma_k\}$ is bounded. By (4.1), $\sigma_k$ 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 $\Omega^c$. We have to prove that situation (*ii*–*a*) of the proposition occurs. We denote by $C > 0$ an "absorbing" positive constant independent of $k$.

From the definition (3.17) of $F_k^{i_k}$, the fact that (3.10c) holds for $i = 1, \ldots, i_k - 1$, $g_k^T Z_k t_k \leq 0$, (4.1), and the boundedness of $\{B_k\}$, we have

$$\begin{aligned} F_k^{i_k} &= -\omega_1 \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i)\Theta'_\sigma(x_k^i; d_k^i) \\ &\geq \omega_1 \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i) \left(\omega_2 \tau_k^i g_k^T B_k^{-1} g_k + \bar{\sigma}\|c(x_k^i)\|\right) \\ &\geq C \left( \sum_{i=0}^{i_k-1} \tau_k^i (\alpha_k^{i+1} - \alpha_k^i) \|g_k\|^2 + \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i) \|c(x_k^i)\| \right). \end{aligned} \tag{4.3}$$

As the sequence $\{\Theta_\sigma(x_k)\}_{k\geq k_2}$ decreases and is bounded below, it converges. Then, from (3.20) and $\omega_1 \nu_k(\alpha_k) \leq -F_k^{i_k}$ [from (3.18) with $i = i_k$], we see that $F_k^{i_k} \to 0$. Therefore, the terms in the right hand side of (4.3) converge to zero when $k \to \infty$:

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

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

$$\alpha_k^1 \to 0, \quad \text{when } k \to \infty \text{ in } \mathcal{K}. \tag{4.5}$$

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 $k \in \mathcal{K}$, $\alpha_k^{0,j_1-1}$ is not accepted by the search algorithm because the descent condition (3.14) 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

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

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 Assumption 4.1 $(i)$, (4.4), and the boundedness of $\{Z_k^-\}$, $\{B_k^{-1}\}$, and $\{A_k^-\}$, we have for $k \to \infty$ in $\mathcal{K}$

$$\|\alpha_k^{0,j_1-1}\, \tau_k^0\, t_k\|^2 \leq C\, \alpha_k^1\, \tau_k^0\, \|g_k\|^2 \to 0,$$

$$\|\alpha_k^{0,j_1-1}\, r_k\| \leq C\, \alpha_k^1\, \|c_k\| \to 0.$$

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

Therefore, we can suppose that (3.14) is not satisfied for $i = 0$, $\alpha = \alpha_k^{0,j_1-1}$, and $k \in \mathcal{K}$, i.e.,

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

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

First, using the Lipschitz continuity of $f'$ on $\Omega$:

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

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

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

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

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

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

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

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

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

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

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

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. □

When $g$ is Lipschitz continuous on $\Omega$, Assumptions 4.1 are no longer necessary to prove that $g_k \to 0$. This can be shown by a standard argument, using (3.21) and (4.4). But we were not able to prove that $c_k \to 0$ without them, for the reasons given above. On the other hand, once these assumptions hold, condition (3.21) 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-esc method described in Section 3.6, the conclusion of Proposition 4.2 still hold.

# 5 Numerical experiment

The behavior of the PLS technique introduced in Section 3 and the reduced quasi-Newton algorithm presented in Section 4 has been tested on a single model problem with a dimension ranging from $n = 2$ to 500 and a single constraint. The experiment has been done in double precision on a SUN SPARCstation 1, with a program written in Fortran-77.

The test-problem consists in minimizing a quadratic function with diagonal Hessian on the unit sphere. The function $f$ to minimize and the constraint function $c$ are defined on $\Omega = \{x \in \mathbb{R}^n : x_{(1)} > 0\}$ by

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

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

$$Z^-(x) = \begin{pmatrix} -\tilde{x}^T \\ x_{(1)} I_{n-1} \end{pmatrix}, \qquad A^-(x) = \frac{x}{\|x\|_2^2},$$

where $\tilde{x}$ is the vector of $\mathbb{R}^{n-1}$ formed of the last $n-1$ components of $x$ and $I_{n-1}$ 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 chosen orthogonal (for the Euclidean scalar product) to the tangent space of the constraint manifold.

Table 1 gives some information on the problems: $n$ is the number of variables

| $n$ | $x_{*(1)}$ | $\kappa_2(B_*)$ |
|---|---|---|
| 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. |

Table 1: Test-problems.

(hence $n-1$ is the dimension of the constraint manifold), $x_{*(1)}$ is the first component of the solution and $\kappa_2(B_*)$ is the $\ell_2$ condition number of the reduced Hessian of the Lagrangian at the solution (computed by the LAPACK program DSYEV).

To globalize the algorithm, the exact $\ell_1$ penalty function (with the $\ell_1$-norm in (1.7)) is used with $\sigma_1 = 2\|\lambda_1\|_2$ initially. Next, $\sigma_k$ 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 $x_k$ if

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

The decision to start a PLS, instead of a simple Armijo backtracking, is taken by an update criterion: if $\|r_k\|_2 \leq \mu\|e^1_{k\ominus 2}\|_2\,\|t_k\|_2$, an update is desirable and the PLS is started as explained in Section 3.2. In this criterion, $\mu$ is set to a positive constant such that equality occurs initially, $k \ominus 2$ is the index of the last but one iteration at which an update occurred before iteration $k$ (see [13] or [15]), and $e^1_k = \alpha^1_k d_k$.

As far as the PLS algorithm is concerned, we have always set $\rho_k^i = 1$ in (3.12), 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 $\tau_k^0$ 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 $\{\alpha_k^{i,j}\}_{j=2}^{j_i}$. When the PLS algorithm succeeds (situation $(i)$ of Proposition 3.2), the matrix $B_k^{-1}$ is updated by the inverse BFGS formula. The first time this occurs, say for $k = k_0$, the inverse matrix is first initialized to $\gamma_{k_0}^T \delta_{k_0} / \|\gamma_{k_0}\|^2 I$ before being updated.

The results of our experiment are given in Tables 2, 3, 4 and 5, in which some common symbols are used: '$n$' is the dimension of the problem, 'iter' is the number of iterations, 'func' is the number of function calls, 'lin' is the number of times the constraints are linearized, '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 the other symbols are 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 $\gamma_k^T \delta_k$ is positive, $B_k$ is updated, otherwise the update is skipped. The results are given in Table 2.

| $n$ | iter | func | skip | $\sigma \nearrow$ |
|---|---|---|---|---|
| 2 | 17 | 23 | 1 | 0 |
| 5 | 55 | 58 | 1 | 1 |
| 10 | 88 | 93 | 5 | 1 |
| 20 | 110 | 116 | 21 | 2 |
| 50 | 82 | 89 | 3 | 3 |
| 100 | 83 | 95 | 2 | 3 |
| 200 | 72 | 90 | 3 | 4 |
| 500 | 91 | 98 | 11 | 5 |

Table 2: AS-skip: Armijo's search and skipping rule.

In the next experiment, Armijo's backtracking is still used, but a correction is made to $\delta_k$ when $\gamma_k^T \delta_k$ is not sufficiently positive (see Powell [28]): $\tilde{\delta}_k = \theta \delta_k + (1-\theta)\gamma_k$, where

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

The update of $B_k^{-1}$ is then made with $\tilde{\delta}_k$ instead of $\delta_k$. Table 3 shows the results; 'P-cor' is the number of Powell's corrections. We see that this algorithm works usually better than the method with skipping rule (although the difference is not

| $n$ | 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 |

Table 3: AS-Powell: Armijo's search and Powell's correction.

important), except for the cases $n = 100$ and $n = 500$, for which AS-Powell is surprisingly much worse.

The last two experiments use the PLS technique, provided the update criterion mentioned above holds. Hence, an update skip occurs when the PLS is interrupted by the update criterion or by the test on the penalty parameter. In the first experiment, whose results are in Table 4, the plain PLS method described in Sections

| $n$ | iter | lin | func | skip | $\sigma \nearrow$ |
|---|---|---|---|---|---|
| 2 | 17 | 18 | 23 | 3 | 0 |
| 5 | 53 | 62 | 65 | 3 | 1 |
| 10 | 70 | 82 | 87 | 3 | 1 |
| 20 | 75 | 173 | 178 | 4 | 2 |
| 50 | 82 | 107 | 121 | 4 | 3 |
| 100 | 82 | 86 | 98 | 5 | 3 |
| 200 | 80 | 142 | 165 | 5 | 4 |
| 500 | 97 | 164 | 195 | 6 | 5 |

Table 4: PLS: Piecewise line-search and update criterion.

3.2 and 3.4 is used with $\tau_k^i$ 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 3.2). Now, comparing the number of linearizations with those of the previous experiment, we see that the results are not very satisfactory: the PLS algorithm requires a great number of inner iterations. By looking more precisely at the results, however, we have observed that the deterioration is mainly due to a few iterations (never more than 5 for each run), during which the behavior of the PLS is very bad, in a way that may be viewed as the one described in Section 3.6.

The last experiment is done with the PLS-esc algorithm of Section 3.6. The results are given in Table 5. The PLS algorithm is interrupted as soons as condition

| $n$ | iter | lin | func | esc | skip | $\sigma\nearrow$ |
|---|---|---|---|---|---|---|
| 2 | 17 | 18 | 23 | 0 | 3 | 0 |
| 5 | 52 | 56 | 60 | 0 | 3 | 1 |
| 10 | 80 | 85 | 90 | 1 | 3 | 1 |
| 20 | 72 | 79 | 82 | 1 | 3 | 2 |
| 50 | 79 | 84 | 94 | 1 | 3 | 3 |
| 100 | 81 | 85 | 95 | 2 | 4 | 3 |
| 200 | 69 | 77 | 84 | 3 | 4 | 4 |
| 500 | 97 | 108 | 131 | 4 | 6 | 5 |

Table 5: PLS-esc: Piecewise line-search (with escape) and update criterion.

(3.24) holds. Furthermore, the tangent scaling factor $\tau_k^i$ may be different from 1: either a safeguarded extrapolation formula is used or (when this is useless, which is generally the case) $\tau_k^i$ is doubled at each inner iteration (provided the descent test (3.10b) has always been verified with $\alpha_k^i = \alpha_k^{i-1,1}$ during the current PLS). The number of "escapes" are given in a column labeled by 'esc' in Table 5: it is the number of times condition (3.24) differs from (3.21). We see that, despite the very few "escapes", the results improve dramatically. We also observe that the number of additional inner iterations used by the PLS-esc algorithm (= 'lin' – 'iter' – 1) is small. The results of PLS-esc compare favorably with those of the other techniques.

To summarize, we add up the number of iterations, linearizations and function calls for the considered algorithms: see Table 6 (for every run with AS-skip or AS-

| Algorithm | iter | lin | func |
|---|---|---|---|
| AS-skip | 598 | 606 | 662 |
| AS-Powell | 607 | 615 | 701 |
| PLS | 556 | 834 | 932 |
| PLS-esc | 547 | 592 | 659 |

Table 6: Compared performance of the algorithms.

Powell, 'lin' = 'iter' + 1). This allows us to make a comparison. If the PLS-esc technique appears to be the best one, the difference with the other search methods is not significant. Moreover, our experiment is limited to a single problem, which impedes to make firm conclusions. We believe, however, that this limited number of tests validates the PLS approach.

# References


[1] L. Armijo (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16, 1–3.

[2] L.T. Biegler, J. Nocedal, C. Schmid (1993). A reduced Hessian method for large-scale constrained optimization. Preprint MCS-P364-0693, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439.

[3] R.H. Byrd, J. Nocedal (1991). An analysis of reduced Hessian methods for constrained optimization. *Mathematical Programming*, 49, 285–323.

[4] R.H. Byrd, R.A. Tapia, Y. Zhang (1992). An SQP augmented Lagrangian BFGS algorithm for constrained optimization. *SIAM Journal on Optimization*, 2, 210–241.

[5] T.F. Coleman, A.R. Conn (1982). Nonlinear programming via an exact penalty function: asymptotic analysis. *Mathematical Programming*, 24, 123–136.

[6] T.F. Coleman, A.R. Conn (1984). On the local convergence of a quasi-Newton method for the nonlinear programming problem. *SIAM Journal on Numerical Analysis*, 21, 755–769.

[7] T.F. Coleman, P.A. Fenyes (1992). Partitioned quasi-Newton methods for nonlinear constrained optimization. *Mathematical Programming*, 53, 17–44.

[8] J.E. Dennis, R.B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs.

[9] P. Fenyes (1987). *Partitioned quasi-Newton methods for nonlinear equality constrained optimization*. PhD Thesis, Department of Computer Science, Cornell University.

[10] R. Fletcher (1980). *Practical Methods of Optimization. Volume 1: Unconstrained Optimization*. John Wiley & Sons, Chichester.

[11] D. Gabay (1982). Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37, 177–219.

[12] D. Gabay (1982). Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization. *Mathematical Programming Study*, 16, 18–44.

[13] J.Ch. Gilbert (1988). Mise à jour de la métrique dans les méthodes de quasi-Newton réduites en optimisation avec contraintes d'égalité. *Modélisation Mathématique et Analyse Numérique*, 22, 251–288.

[14] J.Ch. Gilbert (1991). Maintaining the positive definiteness of the matrices in reduced secant methods for equality constrained optimization. *Mathematical Programming*, 50, 1–28.

[15] J.Ch. Gilbert (1993). Superlinear convergence of a reduced BFGS method with piecewise line-search and update criterion. Rapport de recherche, INRIA, BP 105, 78153 Le Chesnay, France.

[16] J.Ch. Gilbert, C. Lemaréchal (1989). Some numerical experiments with variable-storage quasi-Newton algorithms. *Mathematical Programming*, 45, 407–435.

[17] S.-P. Han (1976). Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11, 263–282.

[18] K. Kunisch, E.W. Sachs (1993). Reduced SQP methods for parameter identification problems. *SIAM Journal on Numerical Analysis*, 29, 1793–1820.

[19] F.S. Kupfer, E.W. Sachs (1992). Numerical solution of a nonlinear parabolic control problem by a reduced SQP method. *Computational Optimization and Applications*, 1, 113–135.

[20] C. Lemaréchal (1981). A view of line-searches. In A. Auslender, W. Oettli, J. Stoer (editors), *Optimization and Optimal Control*, Lecture Notes in Control and Information Science 30, pages 59–78. Springer-Verlag, Heidelberg.

[21] D.C. Liu, J. Nocedal (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45, 503–520.

[22] D.Q. Mayne, E. Polak (1982). A superlinearly convergent algorithm for constrained optimization problems. *Mathematical Programming Study*, 16, 45–61.

[23] J.J. Moré, D.C. Sorensen (1984). Newton's method. In G.H. Golub (editor), *Studies in Numerical Analysis*, pages 29–82. The Mathematical Association of America.

[24] J.J. Moré, D.J. Thuente (1992). Line search algorithms with guaranteed sufficient decrease. Preprint MCS-P330-1092, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439.

[25] W. Murray, 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 94305.

[26] J. Nocedal, M.L. Overton (1985). Projected Hessian updating algorithms for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, 22, 821–850.

[27] M.J.D. Powell (1978). Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, 14, 224–248.

[28] M.J.D. Powell (1978). A fast algorithm for nonlinearly constrained optimization calculations. In W.A. Watson (editor), *Numerical Analysis*, pages 144–157. Springer.

[29] M.J.D. Powell (1985). The performance of two subroutines for constrained optimization on some difficult test problems. In P.T. Boggs, R.H. Byrd, R.B. Schnabel (editors), *Numerical Optimization 1984*. SIAM Publication, Philadelphia.

[30] B.N. Pshenichnyi, Yu.M. Danilin (1978). *Numerical Methods for Extremal Problems.* MIR, Moscow.

[31] J. Stoer (1984). Principles of sequential quadratic programming methods for solving nonlinear programs. In *Proceedings of the NATO ASI on Computational Mathematical Programming*, Bad Windsheim, Germany.

[32] R.A. Tapia (1977). Diagonalized multiplier methods and quasi-Newton methods for constrained optimization. *Journal of Optimization Theory and Applications*, 22, 135–194.

[33] R.A. Tapia (1988). On secant updates for use in general constrained optimization. *Mathematics of Computation*, 51, 181–202.

[34] R.B. Wilson (1963). *A simplicial algorithm for concave programming.* PhD Thesis, Graduate School of Business Administration, Harward University, Boston.

[35] P. Wolfe (1969). Convergence conditions for ascent methods. *SIAM Review*, 11, 226–235.

[36] P. Wolfe (1971). Convergence conditions for ascent methods II: some corrections. *SIAM Review*, 13, 185–188.

Unité de Recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399