Piecewise line-search techniques for constrained minimization by quasi-Newton algorithms

Jean Charles Gilbert[†]

September 1996

Abstract

Defining a consistent technique for maintaining the positive definiteness of the matrices generated by quasi-Newton methods for equality constrained minimization remains a difficult open problem.

In this paper, we review and discuss the results obtained with a new technique based on an extension of the Wolfe line-search used in unconstrained optimization. The idea is to follow a piecewise linear path approximating a smooth curve, along which some reduced curvature conditions can be satisfied. This technique can be used for the quasi-Newton versions of the reduced Hessian method and for the SQP algorithm.

A new argument based on geometrical considerations is also presented. It shows that in reduced quasi-Newton methods the appropriateness of the vectors used to update the matrices may depend on the type of bases representing the tangent space to the constraint manifold. In particular, it is argued that for orthonormal bases, it is better using the projection of the change in the gradient of the Lagrangian rather than the change in the reduced gradient.

Finally, a strong q-superlinear convergence theorem for the reduced quasi-Newton algorithm is discussed. It shows that if the sequence of iterates converges to a strong solution, it converges one step q-superlinearly. Conditions to achieve this speed of convergence are mild; in particular, no assumptions are made on the generated matrices. This result is, to our knowledge, the first extension to constrained optimization of a similar result proved by Powell in 1976 for unconstrained problems.

Key words: constrained optimization, nonlinear programming, piecewise line-search, quasi-Newton BFGS updates, reduced curvature condition, reduced Hessian method, SQP algorithm, Wolfe's line-search.

The paper is organized as follows. In Section 1, we give a succinct presentation of quasi-Newton algorithms for solving unconstrained and equality constrained minimization problems. We recall what is the process to maintain the positive definiteness of the updated matrices in unconstrained optimization and the difficulties to extend this approach to constrained problems.

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

In section 2, we discuss the different choices of pairs (γ_k, δ_k) that have been proposed to update the matrices in the quasi-Newton versions of the reduced Hessian and SQP algorithms. We also sketch the principles of the piecewise line-search (PLS), further developed in Sections 4 and 5.

In Section 3, we present a new argument, based on geometrical considerations, that can be helpful in determining the form of the pairs (γ_k, δ_k) to use for updating the matrices in reduced quasi-Newton algorithms. It is argued that, when γ_k is the change in the reduced gradient, the bases tangent to the constraint manifolds should satisfy a so-called "zero Lie bracket bases" property, which is equivalent to the existence of local parametrizations of these manifolds, whose derivatives give the bases. If this property holds for the tangent bases computed by partitioning the Jacobian matrix of the constraints, it is usually not satisfied when the tangent bases are orthonormal.

Section 4 deals with the PLS technique for minimizing functions subject to equality constraints by reduced quasi-Newton algorithms. The technique is viewed as an explicit Euler discretization of a smooth curve, along which a so called "reduced curvature condition" can be satisfied, while decreasing a merit function. A new update criterion is introduced, with which a strong q-superlinear convergence result can be proved.

In Section 5, it is shown how the PLS technique can be used for the quasi-Newton version of the SQP algorithm. A strategy to maintain positive definite the matrices M_k approximating of the Hessian of the augmented Lagrangian is presented. It is based on two observations. The first one is that the penalty parameter can be set to ensure $d^\top M_k d > 0$ when d is not in the tangent space to the constraint manifold. The second one is that the PLS can ensure the positivity of $d^\top M_k d$ when d is in this tangent space. The update scheme combines these two ideas without having to switch between them.

We end the paper with a conclusion section listing a few open problems.

1 Quasi-Newton algorithms

In this section, we present quasi-Newton algorithms for solving unconstrained (Section 1.1) and equality constrained (Section 1.2) minimization problems. We restrict our presentation to line-search methods and to algorithms having the potential to converge one step q-superlinearly (or one step q-quadratically, when second derivatives are computed). We emphasize the similarities between the approaches.

1.1 The Wolfe line-search in unconstrained optimization

Consider the problem of minimizing a smooth function $f : \mathbb{R}^n \to \mathbb{R}$, which we write:

$$\min f(x). \tag{1.1}$$

Descent direction algorithms solve problem (1.1) by generating a sequence of approximations $x_k \in \mathbb{R}^n$ of a solution x_* of (1.1). These iterates are obtained by the recurrence

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

where d_k is a descent direction of f at x_k, which means that $f'(x_k) \cdot d_k < 0$, and $\alpha_k > 0$ is a step-size found by a technique called *line-search*. The aim of this technique is, in particular, to force the decrease of f at each iteration. Frequently, this is controlled by requiring α_k to satisfy the inequality

$$f(x_k + \alpha_k d_k) \le f(x_k) + \omega_1 \alpha_k \nabla f(x_k)^\top d_k, \tag{1.2}$$

where $\omega_1 \in (0,1)$ is a constant and $\nabla f(x_k)$ denotes the gradient of f at x_k associated with the Euclidean (or ℓ_2) inner product. Since the last term in (1.2) is negative, f decreases at each iteration.

The type of direction d_k characterizes the minimization algorithm. In quasi-Newton methods, this direction takes the form

$$d_k = -M_k^{-1} \nabla f(x_k),$$

where M_k is a nonsingular symmetric matrix approximating the Hessian of f at x_k. In practice, it is suitable to maintain M_k positive definite, for two reasons that we will meet again when we will consider algorithms for constrained problems. First, this is natural since the Hessian of f is positive definite at a strong minimum, i.e., a solution of (1.1) satisfying the second order sufficient conditions of optimality. Secondly, this makes d_k a descent direction of f at x_k: $f'(x_k) \cdot d_k = -\nabla f(x_k)^\top M_k^{-1} \nabla f(x_k) < 0$, when x_k is not stationary.

The way of maintaining the positive definiteness of M_k is the central theme of this paper. In unconstrained optimization, it is obtained by a nice combination of ideas. First M_k is updated by a formula that allows this property to be sustained from one iteration to the next one. An example, which is very efficient in practice, is the BFGS formula:

$$M_{k+1} = M_k - \frac{M_k \delta_k \delta_k^\top M_k}{\delta_k^\top M_k \delta_k} + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k}. \tag{1.3}$$

In (1.3), γ_k and δ_k are two vectors of $\mathbb{R}^n$; γ_k gives the change in the gradient of f and δ_k is the step between two successive iterates:

$$\gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k) \quad \text{and} \quad \delta_k = x_{k+1} - x_k = \alpha_k d_k.$$

Since M_{k+1} defined by (1.3) satisfies the so-called quasi-Newton equation

$$\gamma_k = M_{k+1} \delta_k, \tag{1.4}$$

it incorporates information from the Hessian of f. On the other hand, taking the inner product of the two sides of (1.4) with δ_k gives $\gamma_k^\top \delta_k = \delta_k^\top M_{k+1} \delta_k$. Hence, if M_{k+1} is positive definite, one has

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

An interesting property of formula (1.3), which will play a key role in this paper, is that the converse is true: when M_k is positive definite and (1.5) holds, then M_{k+1} is also positive definite. This implies that satisfying (1.5) is crucial for the success of the algorithm.

In unconstrained optimization, (1.5) is realized by the line-search. The so-called Wolfe line-search [50, 51] is the appropriate concept. It finds a step-size $\alpha_k > 0$ satisfying (1.2) and

$$\nabla f(x_k + \alpha_k d_k)^\top d_k \geq \omega_2 \nabla f(x_k)^\top d_k, \tag{1.6}$$

where $\omega_1 < \omega_2 < 1$. These two conditions (1.2) and (1.6) can be satisfied simultaneously under mild assumptions and there are specific algorithms that can realize this; see [21, 36, 37, 2]. By subtracting $\nabla f(x_k)^\top d_k$ from both sides of (1.6), we see that (1.6) implies (1.5), so that we finally have the series of implications:

$$\begin{array}{lll} (1.6) & \Longrightarrow & \gamma_k^\top \delta_k > 0 \\ & \Longrightarrow & M_{k+1} \text{ is positive definite} \\ & \Longrightarrow & d_{k+1} \text{ is a descent direction.} \end{array}$$

All of this is described in detail in textbooks or review articles such as [31, 18, 22, 45, 19, 38, 23].

If satisfying inequality (1.5) is mandatory to maintain the positive definiteness of the updated matrices, the way of satisfying it in constrained optimization is still a subject of debate and research. In this paper, we describe a step-size determination process that implies (1.5), so generalizing the Wolfe line-search. Note that, for constrained problems, γ_k is no longer the change in the gradient of f. Also, the meaning of γ_k and its size change according to the algorithm considered.

1.2 Two classes of quasi-Newton algorithms for constrained problems

Consider now the problem of minimizing a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ subject to equality constraints given by a smooth function $c : \mathbb{R}^n \to \mathbb{R}^m$ ($m < n$):

$$\left\{ \begin{array}{l} \min\ f(x) \\ c(x) = 0. \end{array} \right. \tag{1.7}$$

We consider problems with equality constraints only. Algorithms dealing with inequalities are often of two types: active set methods (see [31, 22]) or interior point methods (see [7, 14, 17]). These algorithms may have to solve subproblems like (1.7), so that algorithms for solving (1.7) are also interesting in that case.

The optimality conditions at a solution x_* of (1.7) can be written: there exists $\lambda_* \in \mathbb{R}^m$ such that

$$\begin{cases} \nabla \ell(x_*, \lambda_*) = 0 \\ c(x_*) = 0, \end{cases} \tag{1.8}$$

where $\ell(x, \lambda) = f(x) + \lambda^\top c(x)$ is the *Lagrangian* of the problem and ∇ denotes the gradient with respect to x. The existence of λ_* is ensured when the $m \times n$ Jacobian matrix of the constraint at the solution $A(x_*) = c'(x_*)$ is surjective. We assume that $A(x) = c'(x)$ is surjective for all $x \in \mathbb{R}^n$.

Remark 1.1. Below, more hypotheses will be made on the problem data. Some of them are very restrictive when they are assumed to hold for all $x \in \mathbb{R}^n$ (for example, the existence of a continuous field of bases tangent to the constraint manifold). It would be cleaner to specify the open set Ω containing x_* on which these hypotheses hold. However, the case when $\Omega \neq \mathbb{R}^n$ is more complicated to describe, since the line-search has to control the step-size such that the next iterate is also in Ω. To avoid unnecessary complications, which would hide the essence of the algorithms, we will not do so, knowing that simple modifications can take into account the case when $\Omega \neq \mathbb{R}^n$.

The SQP algorithm

The *Sequential Quadratic Programming* (SQP) method determines the search direction d_k at an iterate (x_k, λ_k) by solving the quadratic program in the variable $d \in \mathbb{R}^n$:

$$\begin{cases} \min \nabla f(x_k)^\top d + \frac{1}{2} d^\top M_k d \\ c(x_k) + A(x_k) d = 0, \end{cases} \tag{1.9}$$

where M_k is the Hessian of the Lagrangian evaluated at (x_k, λ_k) or an approximation to it (see [22, 30, 5]).

It also makes sense to force M_k to be an approximation of the Hessian of the augmented Lagrangian

$$\ell_\tau(x, \lambda) = \ell(x, \lambda) + \frac{\tau}{2} \|c(x)\|_2^2,$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm (see [34, 48, 49]). This is the point of view that we take in this paper since, for τ sufficiently large, $\nabla^2 \ell_\tau(x_*, \lambda_*)$ is positive definite at a strong solution of (1.7) (a solution satisfying the strong sufficient conditions of optimality), so that it is reasonable to update positive definite approximations M_k.

As in unconstrained problems, another reason to maintain M_k positive definite is that the solution d_k of (1.9) can be a descent direction of the merit function

$$\Theta_\sigma(x) = f(x) + \sigma\|c(x)\|_{_P}, \tag{1.10}$$

where $\sigma > 0$ is a penalty parameter and $\|\cdot\|_{_P}$ denotes an arbitrary norm on $\mathbb{R}^m$. Indeed

$$\Theta'_\sigma(x_k; d_k) = -d_k^\top M_k d_k + (\lambda_k^{\mathrm{QP}})^\top c(x_k) - \sigma\|c(x_k)\|_{_P},$$

where λ_k^{QP} is the multiplier associated with the constraints in (1.9). Hence $\Theta'_\sigma(x_k; d_k)$ is negative if M_k is positive definite,

$$\sigma > \|\lambda_k^{\mathrm{QP}}\|_{_D}, \tag{1.11}$$

and x_k is not stationary for (1.7). We have used the dual norm

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

For the globalization of the SQP algorithm, i.e., for forcing its convergence from a remote starting point x_1, the following property is important. When $\sigma > \|\lambda_*\|_{_D}$, a strong solution of (1.7) is a strict local minimum of Θ_σ: Θ_σ is said to be an *exact penalty function*. In that case, Θ_σ can be used to measure the progress to optimality and to determine the step-size α_k along the descent direction d_k. The natural extension of inequality (1.2) is then

$$\Theta_{\sigma_k}(x_k + \alpha_k d_k) \le \Theta_{\sigma_k}(x_k) + \omega_1 \alpha_k \Theta'_{\sigma_k}(x_k; d_k). \tag{1.12}$$

In this inequality we have allowed the penalty parameter σ_k to change at each iteration to satisfy (1.11). To ensure convergence, the step-size cannot be too small and, for example, can be determined by backtracking [3]. Then, the next iterate is set to

$$x_{k+1} = x_k + \alpha_k d_k.$$

The reduced Hessian algorithm

The reduced Hessian algorithm is particularly useful when $n-m \ll n$, because, as we will see, its quasi-Newton version, also called *reduced quasi-Newton algorithm*, needs to update matrices of order $n-m$ only, while the SQP method requires updating matrices of order n.

An iteration of the *reduced Hessian algorithm* consists of two stages; see [11, 35, 26, 8]. Starting an iteration at $x_k \in \mathbb{R}^n$, a tangent step t_k is first computed by solving the quadratic program in the variable $t \in \mathbb{R}^n$:

$$\left\{\begin{array}{l} \min\ \nabla f(x_k)^\top t + \frac{1}{2} t^\top M_k t \\ A(x_k) t = 0, \end{array}\right. \tag{1.13}$$

where M_k is again the Hessian of the Lagrangian or an approximation to it. This problem is similar to (1.9), except that t_k is now in the null space of $A(x_k)$ which is also the tangent space at x_k to the manifold

$$\mathcal{M}_{x_k} = \{x \in \mathbb{R}^n : c(x) = c(x_k)\}.$$

Therefore, only the part of M_k acting on this tangent space is visible by the algorithm (M_k is not used in the second stage of the algorithm). It is that part that will have to be updated in the quasi-Newton version of the method.

Next, a Newton-like direction r_k for solving the second equation in (1.8) is determined. For this, one uses a right inverse $A^-(x_k)$ of the Jacobian $A(x_k)$ (hence $A(x_k)A^-(x_k) = I_m \in \mathbb{R}^{m\times m}$) and one computes

$$r_k = -A^-(x_k)c(x_k + t_k). \tag{1.14}$$

Finally, in the local algorithm (without line-search), x_{k+1} is set to $x_k + t_k + r_k$.

Now, let $Z^-(x_k)$ be a *basis* of the null space $N(A(x_k))$ of $A(x_k)$, i.e., an $n\times(n-m)$ injective matrix such that $A(x_k)Z^-(x_k) = 0$. Then, the solution of (1.13) can be written

$$t_k = -Z^-(x_k)G_k^{-1}g(x_k), \tag{1.15}$$

where $G_k = Z^-(x_k)^\top M_k Z^-(x_k)$ is or approximates the *reduced Hessian of the Lagrangian*

$$G(x, \lambda) = Z^-(x)^\top \nabla^2 \ell(x, \lambda) Z^-(x)$$

and

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

is the *reduced gradient* of f.

Note that for any choice of right inverse $A^-(x)$ of $A(x)$ and basis $Z^-(x)$ of $N(A(x))$, there exists an $(n-m)\times n$ matrix $Z(x)$, uniquely determined by

$$A^-(x)A(x) + Z^-(x)Z(x) = I_n. \tag{1.16}$$

We deduce from this that

$$Z(x)Z^-(x) = I_{n-m} \quad \text{and} \quad Z(x)A^-(x) = O_{(n-m)\times m},$$

so that $Z^-(x)$ is a right inverse of $Z(x)$.

When G_k is a quasi-Newton approximation of the reduced Hessian of the Lagrangian, it is interesting to maintain this matrix positive definite for two reasons similar to those met in Section 1.1 for unconstrained problems and in this section for the SQP algorithm. First, the reduced Hessian of the Lagrangian is positive definite at a strong solution of (1.7). Secondly, when G_k is positive definite and $g(x_k) \neq 0$, the directional derivative

$$\Theta'_{\sigma_k}(x_k; t_k) = -g(x_k)^\top G_k^{-1} g(x_k),$$

is negative, making t_k a descent direction of Θ_{σ_k} at x_k.

Then, the globalization of the reduced quasi-Newton algorithm can be done along the following lines [26, 28], similar to those followed above. Since the direction r_k is computed by evaluating functions at two different points, x_k and $x_k + t_k$, there is no guarantee that $t_k + r_k$ will be a descent direction of Θ_{σ_k} at x_k. On the other hand, introducing

$$r_k^0 = -A^-(x_k)c(x_k), \tag{1.17}$$

we observe that

$$\Theta'_{\sigma_k}(x_k; t_k + r_k^0) = -g(x_k)^\top G_k^{-1} g(x_k) + \lambda(x_k)^\top c(x_k) - \sigma_k \|c(x_k)\|_P,$$

where $\lambda(x)$ is the multiplier estimate

$$\lambda(x) = -A^-(x)^\top \nabla f(x). \tag{1.18}$$

We deduce from this that $t_k + r_k^0$ is a descent direction of Θ_{σ_k} provided

$$\sigma_k > \|\lambda(x_k)\|_D$$

and x_k is not stationary for (1.7). Therefore, if σ_k is sufficiently large, one can force the decrease of Θ_{σ_k} by determining a step-size $\alpha_k > 0$ along the quadratic curve joining x_k and $x_k + t_k + r_k$, and tangent to $t_k + r_k^0$ at x_k,

$$\alpha \mapsto p_k^0(\alpha) = x_k + \alpha(t_k + r_k^0) + \alpha^2(r_k - r_k^0).$$

The step-size will be determined, for example by backtracking, so that

$$\Theta_{\sigma_k}(p_k^0(\alpha_k)) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \alpha_k \Theta'_{\sigma_k}(x_k; t_k + r_k^0). \tag{1.19}$$

The next iterate is then set to $x_{k+1} = p_k^0(\alpha_k)$.

1.3 Realizing $\gamma_k^\top \delta_k > 0$

If we compare the two algorithms of the previous section, with the unconstrained algorithm of Section 1.1, we see that what is missing in the algorithms for constrained problems is the counterpart of condition (1.6), which ensures the inequality $\gamma_k^\top \delta_k > 0$ and by that the positive definiteness of the updated matrices.

In constrained optimization, realizing $\gamma_k^\top \delta_k > 0$ is a much harder task, at least if one tries to mimic the Wolfe line-search used in unconstrained optimization. Remark, however, that once this condition is realized and is compatible with the inequalities (1.12) or (1.19), the algorithms are quite simple to implement, since they amount to do what is sketched in Section 1.2. In Sections 4 and 5, we dissect the content of the black boxes, introduced in [29, 1], that realize the condition $\gamma_k^\top \delta_k > 0$, by means of a step-size determination process.

Regardless of the meaning of γ_k and δ_k, there is always the possibility to get $\gamma_k^\top \delta_k > 0$ by using the modification of the BFGS formula suggested by Powell [43]. In that approach, the vector γ_k is modified to $\tilde{\gamma}_k = \theta\gamma_k + (1-\theta)M_k\delta_k$, where M_k stands for the matrix to update and θ is the number in $(0,1]$, the closest to 1, such that the inequality $\tilde{\gamma}_k^\top \delta_k \geq \eta\, \delta_k^\top M_k \delta_k$ is satisfied (the constant $\eta \in (0,1)$ is usually set to 0.1 or 0.2 [41, 44]). Then $\tilde{\gamma}_k$ is used instead of γ_k in the BFGS formula (1.3). Powell's correction of γ_k is certainly the most widely used technique in practice. Its success is due to its appealing simplicity and its usually good numerical efficiency. We believe, however, that this technique is not satisfactory for at least three reasons: (*i*) it is hard to figure out what part of the problem data is taken into account by this modification, (*ii*) the asymptotic r-superlinear rate of convergence that can be proved in theory [42] is not as strong as one could expect; the stronger q-superlinear convergence that is obtained by the BFGS algorithm for unconstrained problems would be more satisfactory, and (*iii*) the technique can yield artificial ill-conditioning of the updated matrices, deteriorating the numerical efficiency of the algorithms (see [46, p. 125] and [44]). These facts have motivated further studies (see also [13, 33, 4, 15]).

2 Choices of γ_k and δ_k for constrained problems

Let us now consider how to choose the vectors γ_k and δ_k in the algorithms described in Section 1.2. If we denote by M_* the matrix at the optimal point that is approximated by M_k, what must guide us is the necessity to have

$$\frac{\gamma_k - M_*\delta_k}{\|\delta_k\|} \to 0, \quad \text{when } k \to \infty. \tag{2.1}$$

This estimate is required by the asymptotic analysis. Observe that this rule is satisfied in unconstrained optimization, since M_k approximates $\nabla^2 f(x_k)$ and, when the Hessian of f is Lipschitz continuous and $x_k \to x_*$, one has $\gamma_k = \nabla^2 f(x_k)\delta_k + O(\|\delta_k\|^2) = \nabla^2 f(x_*)\delta_k + o(\|\delta_k\|)$, when $k \to \infty$.

In this section, we limit the discussion on the choices of γ_k and δ_k for algorithms without line-search, since the line-search techniques described in Section 1.2 will be modified anyway. Our aim is to give insight into what γ_k and δ_k should be near the solution and to make explicit the difficulties that are encountered.

2.1 Vectors γ_k and δ_k in reduced quasi-Newton algorithms

Let us consider first the choice of γ_k and δ_k in reduced quasi-Newton algorithms. We observe that, since the reduced gradient is given by $g(x) = Z^-(x)^\top \nabla f(x) = Z^-(x)^\top \nabla \ell(x, \lambda_*)$, its derivative at a stationary point x_* is

$$g'(x_*) = Z^-(x_*)^\top L_*, \tag{2.2}$$

where $L_* = \nabla^2 \ell(x_*, \lambda_*)$ is the Hessian of the Lagrangian with respect to x at (x_*, λ_*).

Since the updated matrices G_k must approach the reduced Hessian of the Lagrangian at the optimal point, $G_* = Z^-(x_*)^\top L_* Z^-(x_*) = g'(x_*)Z^-(x_*)$, the rule (2.1) and formula (2.2) suggest to take

$$\left\{ \begin{array}{l} \gamma_k = g(x_k + t_k) - g(x_k) \\ \delta_k = Z(x_k)t_k. \end{array} \right. \tag{2.3}$$

Indeed, with this choice $t_k = Z^-(x_k)\delta_k$ and, when g' is Lipschitz continuous, we have

$$\gamma_k = Z^-(x_*)^\top L_* t_k + o(\|t_k\|) = G_* \delta_k + o(\|\delta_k\|), \quad \text{when } k \to \infty. \tag{2.4}$$

Another possibility is to take

$$\left\{ \begin{array}{l} \gamma_k = Z^-(x_k)^\top \Big(\nabla \ell(x_k + t_k, \lambda_k) - \nabla \ell(x_k, \lambda_k)\Big) \\ \delta_k = Z(x_k)t_k, \end{array} \right. \tag{2.5}$$

where λ_k is $\lambda(x_k)$ or the multiplier associated with the constraints in (1.13). If f'' and c'' are Lipschitz continuous, we obtain the same estimate (2.4) when (x_k, λ_k) converges to (x_*, λ_*).

Other formulas are proposed in [12, 39, 9]. Close to a solution, they are all equivalent, so that an asymptotic analysis cannot distinguish between them. Now, their global efficiency can be very different. In Section 3 below, we bring a new argument that can help in selecting one of these formulas according to the choice of tangent bases Z^-.

The choices of γ_k and δ_k mentioned above are safe, since they yield (2.1), but they can be very expensive in computing time. They require indeed linearizing functions (i.e., computing ∇f, A and sometimes Z^-) at the intermediate point $x_k + t_k$, while this linearization is not necessary in the local method of Section 1.2. Therefore, researchers have suggested to avoid this linearization by taking

$$\gamma_k = g(x_{k+1}) - g(x_k) \quad \text{or} \quad \gamma_k = Z^-(x_k)^\top (\nabla \ell(x_{k+1}, \lambda_k) - \nabla \ell(x_k, \lambda_k)) \tag{2.6}$$

and to update the matrix G_k only when the update with this new γ_k looks safe. The difficulty with these values of γ_k is that the estimate (2.1) no longer holds. To measure the appropriateness of an update, a possibility is to compare the length of the tangent step t_k to the length of the restoration step r_k. An update can occur if the *update criterion*

$$\|r_k\| \le \mu_k \|t_k\| \tag{2.7}$$

holds [39, 24]. The sequence $\{\mu_k\}$ in (2.7) has to converge to zero if G_k is updated infinitely often. The aim of (2.7) is to select the iterations where the cheap γ_k (given by (2.6)) and the safe γ_k (given by (2.3) or (2.5)) are similar because $x_k + t_k$ and x_{k+1} are closer and closer to each other relatively to the distance between x_k and

$x_k + t_k$. It is not difficult to see that when (2.7) holds with $\mu_k \to 0$, γ_k given by one of the formulas in (2.6) and $\delta_k = Z(x_k)t_k$ satisfy (2.1).

The update criterion (2.7) works well in theory [39], even in a global framework [25, 28], but in practice the numerical results are sometimes disappointing. Now, the criterion (2.7) is rather crude because there is no need to have a small transversal displacement r_k to update the matrix. This is particularly clear when formula (2.3) is used for γ_k. We see that what is necessary for a safe update is to have $g(x_{k+1})$ close to $g(x_k + t_k)$, which can occur even if x_{k+1} and $x_k + t_k$ are far from each other, provided these two points are close the same reduced gradient manifold. In other words, and this formulation of the remark is also valid when the step-size differs from one, the useful information to update G_k is the tangent first order approximation of the vector joining x_k and the intersection of the constraint manifold and the reduced gradient manifold. To our knowledge, there is no update criterion based on this observation. We believe that this topic deserves more attention.

Now that the asymptotic criterion (2.1) has led us to appropriate formulas for γ_k and δ_k, one can ask whether a line-search can help in getting $\gamma_k^\top \delta_k > 0$. The answer is negative in general, even if a search is made along the tangent direction t_k. An example with (2.3) is given in [27]. Therefore, more sophisticated step-size determination techniques have to be introduced. This is the matter of Section 4.

2.2 Vectors γ_k and δ_k in the SQP algorithm

Let us now consider the case of the SQP algorithm and let us use the same criterion (2.1) for determining γ_k and δ_k.

Since M_k may approximate the Hessian of the augmented Lagrangian, a first possibility is to use

$$\begin{cases} \gamma_k = \nabla \ell_\tau(x_{k+1}, \lambda_k^{\mathrm{QP}}) - \nabla \ell_\tau(x_k, \lambda_k^{\mathrm{QP}}) \\ \delta_k = x_{k+1} - x_k. \end{cases}$$

where λ_k^{QP} is the dual solution of (1.9) (see [34, 48, 32]). This approach has however serious practical difficulties: (i) a priori knowledge of the threshold value of τ making positive definite the Hessian of ℓ_τ at the solution is generally unavailable, (ii) large values of τ present severe numerical problems (examples are given in [48, 39]), and (iii) far from the solution, there may be no value of τ and no step-size along d_k that make the inner product $\gamma_k^\top \delta_k$ positive.

Some of the inconvenients of the preceding formula can be overcome by taking advantage of the structure of the Hessian of the augmented Lagrangian at the solution:

$$\nabla^2 \ell_\tau(x_*, \lambda_*) = L_* + \tau A(x_*)^\top A(x_*). \tag{2.8}$$

This suggests to take [49]

$$\begin{cases} \gamma_k = \nabla\ell(x_{k+1}, \lambda_k^{\rm QP}) - \nabla\ell(x_k, \lambda_k^{\rm QP}) + \tau A(x_k)^\top A(x_k)\delta_k \\ \delta_k = x_{k+1} - x_k. \end{cases} \tag{2.9}$$

The inner product of γ_k and δ_k is then

$$\gamma_k^\top \delta_k = \Big(\nabla\ell(x_{k+1}, \lambda_k^{\rm QP}) - \nabla\ell(x_k, \lambda_k^{\rm QP})\Big)^\top \delta_k + \tau\|A(x_k)\delta_k\|^2,$$

so that, one can get $\gamma_k^\top\delta_k > 0$ by taking τ sufficiently large, as long as $A(x_k)\delta_k \neq 0$. It is clear that this strategy does not work when $A(x_k)\delta_k$ is numerically zero and $\gamma_k^\top\delta_k$ is negative. In [10], Byrd, Tapia, and Zhang propose the back-up strategy that consists in replacing $A(x_k)^\top A(x_k)\delta_k$ by δ_k in formula (2.9) when $A(x_k)\delta_k$ is small and $\gamma_k^\top\delta_k$ is not sufficiently positive. Then the positivity of $\gamma_k^\top\delta_k$ can be obtained as before by taking τ sufficient large. Numerical experiment in [10] has shown that this approach is numerically competitive with Powell's correction of the BFGS formula. They also proved that the convergence of the sequence $\{x_k\}$ implies its r-superlinear convergence and even its q-superlinear convergence if the penalty parameter τ is eventually maintained fixed and sufficiently large. This nice result is not completely satisfactory because the threshold value for τ giving the q-superlinear result is usually unknown.

An interesting aspect of the approach in [10] is to give an update rule for the penalty parameter τ. Clearly, the technique appropriately deals with the transversal component of the matrix M_k (its action in the range space of $A^\top$) by setting the parameter τ, but it needs a backup strategy for its longitudinal component (its action in the null space of A). In particular, when the constraints are linear and the iterates are feasible and away from the solution, the back-up strategy may often be active, which is not very desirable. It is clear that in this particular case Wolfe's line-search would overcome the difficulty. In Section 5, we will present a technique combining the use of τ for dealing with the transversal part of M_k and a piecewise line-search that takes care of the longitudinal part of M_k.

2.3 Principle of the new approach

The piecewise line-search (PLS) techniques presented in Sections 4 and 5 are based on the following principle. First, it is observed that a so-called *reduced curvature condition* can be satisfied along a curve that is defined as the solution of an ordinary differential equation. This condition implies the positivity of $\gamma_k^\top\delta_k$, in the same way as (1.6) implies (1.5). Computing a step-size by moving along this curve would be computationally too expensive in general. Also, a piecewise linear approximation of the curve is introduced, using an explicit Euler discretization. At each point of discretization, the reduced curvature condition is tested. If it holds the PLS is interrupted; otherwise, the search is pursued along a new direction.

3 Geometrical considerations

When the pair (γ_k, δ_k) used to update the matrix G_k in the reduced quasi-Newton algorithm is defined by (2.3), one has

$$\gamma_k = \left(\int_0^1 g'(x_k + \alpha t_k) Z^-(x_k)\, d\alpha\right) \delta_k.$$

Therefore, (γ_k, δ_k) collects information from a matrix close to $g'(x)Z^-(x)$. This section addresses the question of the symmetry of this matrix. This is a very desirable property. Indeed, the pair (γ_k, δ_k) is used to update the symmetric matrix G_k, so that it is better when (γ_k, δ_k) does not contain useless information from an unsymmetric matrix.

The matrix $g'(x)Z^-(x)$ is clearly symmetric at the solution, because by formula (2.2), $g'(x_*)Z^-(x_*) = Z^-(x_*)^\top L_* Z^-(x_*)$ is the reduced Hessian of the Lagrangian. But away from x_*, this matrix may not be symmetric. An equivalent condition to the symmetry of $g'(x)Z^-(x)$ is given in Proposition 3.1 below. It uses the following definitions. For further details on the geometrical concepts used in this section, we refer the reader to [47, 6, 16, 20].

A *vector field* on $\mathbb{R}^n$ is a smooth map $X : \mathbb{R}^n \to \mathbb{R}^n$. The *Lie bracket* of two vector fields X and Y on $\mathbb{R}^n$ is the vector field on $\mathbb{R}^n$, denoted by $[X, Y]$ and defined by

$$[X, Y](x) = Y'(x) \cdot X(x) - X'(x) \cdot Y(x).$$

In this formula, $Y'(x) \cdot X(x)$ denotes the usual directional derivative at x of the function Y in the direction $X(x)$. It is not difficult to show that if X and Y are tangent to a submanifold of $\mathbb{R}^n$ in an open neighborhood of a point x, then $[X, Y]$ is also tangent to this submanifold in this neighborhood.

Suppose that the function $c : \mathbb{R}^n \to \mathbb{R}^m$ defining the constraints in (1.7) is smooth and has surjective Jacobian matrices $A(x)$, for all $x \in \mathbb{R}^n$. Then, the set

$$\mathcal{M}_x := \{y \in \mathbb{R}^n : c(y) = c(x)\}$$

is a submanifold of $\mathbb{R}^n$. Let be given a smooth map $x \in \mathbb{R}^n \mapsto Z^-(x) \in \mathbb{R}^{n\times(n-m)}$ such that the columns of $Z^-(x)$ span the null space of $A(x)$ (see Remark 1.1). We denote by $Z^-_{,k}$ the vector field defined by the kth column of Z^-. These vector fields are tangent to the manifolds $\mathcal{M}_x$, so that their Lie brackets $[Z^-_{,k}, Z^-_{,l}]$ are also tangent to these manifolds.

Proposition 3.1. *The matrix $g'(x)Z^-(x)$ is symmetric if and only if*

$$\nabla f(x) \perp [Z^-_{,k}, Z^-_{,l}](x), \quad \textit{for all } k, l \in \{1, \dots, n-m\}. \tag{3.1}$$

In particular, $g'(x)Z^-(x)$ is symmetric when $g(x) = 0$.

Proof. Denoting by $\{e_k\}$ the canonical basis of $\mathbb{R}^{n-m}$, we have

$$e_k^\top g'(x) Z^-(x) e_l = g_k'(x) \cdot Z^-_{,l}(x).$$

Using $g_k(x) = Z^-_{,k}(x)^\top \nabla f(x)$,

$$e_k^\top g'(x) Z^-(x) e_l = ((Z^-_{,k})'(x) \cdot Z^-_{,l}(x))^\top \nabla f(x) + Z^-_{,k}(x)^\top \nabla^2 f(x) Z^-_{,l}(x).$$

Therefore $g'(x)Z^-(x)$ is symmetric if and only if (3.1) holds.

When $g(x) = 0$, $\nabla f(x)$ is perpendicular to the tangent space to $\mathcal{M}_x$ at x. Since $[Z^-_{,k}, Z^-_{,l}](x)$ belongs to that space, (3.1) holds and $g'(x)Z^-(x)$ is symmetric. $\square$

Since at a non stationary point, ∇f is arbitrary, the only way to be sure to have $\nabla f(x)$ perpendicular to $[Z^-_{,k}, Z^-_{,l}](x)$ is to choose tangent bases Z^- with zero Lie bracket columns. As shown in Proposition 3.2, this last property is equivalent to the existence of local parametrizations of the manifolds $\mathcal{M}_x$, whose derivatives give Z^-. Before making this statement precise, we recall the definition of a parametrization.

A *parametrization* of the manifold $\mathcal{M}_x$ around $y \in \mathcal{M}_x$ is a map $\psi : U \to \mathcal{M}_x$ defined on an open set U of $\mathbb{R}^{n-m}$ such that $y \in \psi(U)$, $\psi : U \to \psi(U)$ is a homeomorphism when $\psi(U)$ is endowed with the topology induced from that of $\mathbb{R}^n$, and $i \circ \psi$ (i denotes the canonical injection from $\mathcal{M}_x$ to $\mathbb{R}^n$) is smooth and has injective derivatives $(i \circ \psi)'(u)$ for all $u \in U$.

Proposition 3.2. *Consider the vector fields on $\mathcal{M}_x$ given by the columns of the tangent bases Z^-. The Lie brackets $[Z^-_{,k}, Z^-_{,l}](y) = 0$ for $k, l \in \{1, \ldots, n-m\}$ and all $y \in \mathcal{M}_x$ in a neighborhood of x if and only if there exists a parametrization $\psi : U \to \mathcal{M}_x$ of $\mathcal{M}_x$ around x such that for all $u \in U$, $(i \circ \psi)'(u) = Z^-(\psi(u))$ (i denotes the canonical injection from $\mathcal{M}_x$ to $\mathbb{R}^n$).*

It is easy to see that the *if-part* of the proposition holds. For this, assume the existence of a parametrization ψ with $(i \circ \psi)'(u) = Z^-(\psi(u))$ and let $k, l \in \{1, \ldots, n-m\}$. Then, for $u \in U$ and $y = \psi(u)$, one has

$$(i \circ \psi)'(u) \cdot e_k = Z^-_{,k}(\psi(u)),$$

$$(i \circ \psi)''(u) \cdot (e_k, e_l) = (Z^-_{,k})'(y) \cdot Z^-_{,l}(y).$$

Therefore $[Z^-_{,k}, Z^-_{,l}](y) = 0$ by the symmetry of $(v_1, v_2) \mapsto (i \circ \psi)''(u) \cdot (v_1, v_2)$. The *only-if-part* is more involved. For a proof, see [16, Theorem 4.3.1].

For the *interpretation* of the search algorithm described in Section 4, we will need one of the two equivalent statements of Proposition 3.2, which we quote in the form of an assumption.

Assumption 3.3 (zero Lie bracket bases). The tangent bases Z^- are such that for all $x \in \mathbb{R}^n$ there exists a parametrization $\psi : U \to \mathcal{M}_x$ around x such that $(i \circ \psi)'(u) = Z^-(\psi(u))$ for all $u \in U$ (i denotes the canonical injection from $\mathcal{M}_x$ to $\mathbb{R}^n$). Equivalently, the Lie brackets $[Z^-_{,k}, Z^-_{,l}](x) = 0$ for $k, l \in \{1, \ldots, n-m\}$ and $x \in \mathbb{R}^n$.

Combining Propositions 3.1 and 3.2, we see that when Assumption 3.3 holds, the matrix $g'(x)Z^-(x)$ is symmetric for all $x \in \mathbb{R}^n$. In fact this statement is also a consequence of the following simple calculation.

Proposition 3.4. *Suppose that $\psi : U \subset \mathbb{R}^{n-m} \to \mathcal{M}_x$ is a parametrization of $\mathcal{M}_x$ around x such that $(i \circ \psi)'(u) = Z^-(\psi(u))$ for all $u \in U$. Then, for $u \in U$ and $y = \psi(u)$, one has*

$$g'(y)Z^-(y) = \nabla^2(f \circ \psi)(u).$$

Proof. By definition of g, we have for $u \in U$, $y = \psi(u)$, and $v_1 \in \mathbb{R}^{n-m}$

$$(f \circ \psi)'(u) \cdot v_1 = f'(\psi(u)) \cdot Z^-(\psi(u))v_1 = g(\psi(u))^\top v_1.$$

Differentiating again in u in the direction $v_2 \in \mathbb{R}^{n-m}$ gives

$$(f \circ \psi)''(u) \cdot (v_1, v_2) = v_1^\top g'(y)Z^-(y)v_2.$$

Hence the result. $\square$

The question that arises now is whether the usual ways of computing tangent bases give rise to matrices Z^- satisfying Assumption 3.3. The answer is positive for the tangent bases obtained by partitioning the Jacobian A and is usually negative for orthonormal bases, including those obtained by the QR factorization of $A^\top$. Let us consider these two cases successively.

Since $A(x)$ is surjective, it has m linearly independent columns, which, for simplicity, will be supposed to be the first ones. Then, the Jacobian of c can written

$$A(x) = \begin{pmatrix} B(x) & N(x) \end{pmatrix},$$

where $B(x)$ is an order m nonsingular matrix. Clearly, the columns of the matrix

$$Z^-(x) = \begin{pmatrix} -B(x)^{-1}N(x) \\ I_{n-m} \end{pmatrix} \tag{3.2}$$

form a basis of the null space of $A(x)$.

Proposition 3.5. *The tangent bases given by* (3.2) *satisfy Assumption* 3.3*. In particular, for these bases, $g'(x)Z^-(x)$ is a symmetric matrix.*

Proof. Let us partition the components of $y \in \mathbb{R}^n$ in (ξ, u), with $\xi \in \mathbb{R}^m$ and $u \in \mathbb{R}^{n-m}$. Then, by the implicit function theorem, the nonsingularity of $B(x)$ implies the existence an implicit function $u \in U \mapsto \xi(u) \in \mathbb{R}^m$, where U is an open set of $\mathbb{R}^{n-m}$, such that $c(\xi(u), u) = c(x)$ for all $u \in U$. The parametrization satisfying Assumption 3.3 is $\psi(u) = (\xi(u), u)$. Then, by proposition 3.4, $g'(x)Z^-(x)$ is symmetric. $\square$

The situation is completely different when the columns of $Z^-(x)$ are chosen to be orthonormal, i.e., $Z^-(x)^\top Z^-(x) = I_{n-m}$ for all x, no matter how this matrix is computed. To explain this, we use an argument from Riemannian geometry (see for example [20]). Let x be a point fixed in $\mathbb{R}^n$. With the bases Z^-, one can associate a Riemannian structure on the manifold $\mathcal{M}_x$, by defining the inner product of two tangent vectors $X = Z^-(y)u$ and $Y = Z^-(y)v$ at $y \in \mathcal{M}_x$ by $g(X, Y) = u^\top v$. If Assumption 3.3 holds, it gives parametrizations ψ, which are local isometries between $\mathbb{R}^{n-m}$ and $(\mathcal{M}_x, g)$. Indeed, $\psi^* g$ is the inner product of $\mathbb{R}^{n-m}$:

$$(\psi^* g)(u, v) := g(\psi_* u, \psi_* v) := g(Z^-(y)u, Z^-(y)v) = u^\top v,$$

where ψ_* and ψ^* denote the tangent and cotangent maps associated with ψ. A consequence of this is that the Riemannian curvature of $(\mathcal{M}_x, g)$ is zero, as the one of $\mathbb{R}^{n-m}$. Now, if the columns of the matrices $Z^-(y)$ are orthonormal, the Riemannian structure of $(\mathcal{M}_x, g)$ is also the one induced by $\mathbb{R}^n$ on $\mathcal{M}_x$ considered as a submanifold of $\mathbb{R}^n$, say $(\mathcal{M}_x, g_{_E})$. Indeed, for this induced Riemannian structure, the inner product of $X = Z^-(y)u$ and $Y = Z^-(y)v$ is the one in $\mathbb{R}^n$:

$$g_{_E}(X, Y) = X^\top Y = u^\top Z^-(y)^\top Z^-(y)v = u^\top v = g(X, Y),$$

by the orthonormality of the columns of $Z^-(y)$. Therefore, always under Assumption 3.3, we have shown that the Riemannian curvature of $(\mathcal{M}_x, g_{_E})$ is zero (it is the same as the one of $(\mathcal{M}_x, g)$). This property is satisfied by very particular submanifolds of $\mathbb{R}^n$. For example, it occurs when $\mathcal{M}_x$ is an affine subspace of $\mathbb{R}^n$ or a cylinder (the product $\mathcal{S}^1 \times E$ of the sphere of dimension one $\mathcal{S}^1$ and an affine subspace E). Therefore, when Z^- has orthonormal columns, Assumption 3.3 is rarely satisfied and the matrix $g'(x)Z^-(x)$ is not necessarily symmetric.

The conclusion of this discussion is that when the pairs (γ_k, δ_k) used to update the matrix G_k are given by (2.3), one should use tangent bases Z^- satisfying Assumption 3.3, in order to have $g'(x)Z^-(x)$ symmetric. In particular, formula (3.2) is suitable, but orthonormal bases are usually inadequate.

Finally, note that the question of the symmetry of $g'(x)Z^-(x)$ does not arise when one uses the pairs (γ_k, δ_k) given by formula (2.5), since these pairs collect information from a matrix close to $Z^-(y)^\top \nabla^2 \ell(y, \lambda) Z^-(y)$, which is always symmetric.

We believe that the comments given in this section could explain the much better efficiency of the reduced quasi-Newton algorithm tested in [29], which uses the pairs

(γ_k, δ_k) given by (2.3), when the bases satisfy the zero Lie bracket bases assumption 3.3. Further numerical tests should be necessary to confirm this impression.

4 PLS for reduced quasi-Newton algorithms

4.1 Longitudinal search

Suppose that Assumption 3.3 holds. Then, the parametrizations ψ can be used to interpret algorithms in $\mathbb{R}^n$ as piecewise linear approximations of the image by ψ of algorithms defined in the reduced space $\mathbb{R}^{n-m}$. For example, let $\psi_k : U \to \mathcal{M}_{x_k}$ be a local parametrization of the submanifold $\mathcal{M}_{x_k}$ around $x_k = \psi_k(u_k)$, $u_k \in U$, satisfying $(i \circ \psi_k)'(u) = Z^-(\psi_k(u))$, for all $u \in U$. A quasi-Newton algorithm in $\mathbb{R}^{n-m}$, whose search path has the form

$$\alpha \mapsto u_k + \alpha\delta_k \qquad (\delta_k = -G_k^{-1} g(x_k)),$$

in $\mathbb{R}^{n-m}$, is transformed by ψ_k, in the curve $\alpha \mapsto \bar{p}_k(\alpha)$, solution of the differential equation

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

The first order approximation of the path $\alpha \mapsto \bar{p}_k(\alpha)$ is $\alpha \mapsto x_k + \alpha t_k$, where t_k is the tangent step (1.15) of the reduced Hessian algorithm.

The interest of this interpretation is that we know that under mild assumptions the Wolfe conditions (1.2) and (1.6) can be satisfied in the linear space $\mathbb{R}^{n-m}$ on the function $(f \circ \psi_k)$ along the direction δ_k. This implies that one can find a step-size $\alpha_k > 0$ such that $(0 < \omega_1 < \omega_2 < 1)$:

$$(f \circ \psi_k)(u_k + \alpha_k\delta_k) \le (f \circ \psi_k)(u_k) + \omega_1\alpha_k(f \circ \psi_k)'(u_k) \cdot \delta_k,$$

$$(f \circ \psi_k)'(u_k + \alpha_k\delta_k) \cdot \delta_k \ge \omega_2(f \circ \psi_k)'(u_k) \cdot \delta_k. \tag{4.1}$$

Using the merit function Θ_{σ_k} defined by (1.10) and the properties of the parametrization ψ_k, these conditions can be rewritten (note that $c(\bar{p}_k(\alpha))$ remains constant):

$$\Theta_{\sigma_k}(\bar{p}_k(\alpha_k)) \le \Theta_{\sigma_k}(x_k) + \omega_1\alpha_k g(x_k)^\top\delta_k, \tag{4.2}$$

$$g(\bar{p}_k(\alpha_k))^\top\delta_k \ge \omega_2 g(x_k)^\top\delta_k. \tag{4.3}$$

The last inequality is called the *reduced curvature condition.* This inequality is very interesting because it implies that $(g(\bar{p}_k(\alpha_k)) - g(x_k))^\top\delta_k > 0$, which is the desired inequality $\gamma_k^\top\delta_k > 0$, provided γ_k is the change in the reduced gradient between the points x_k and $\bar{p}_k(\alpha_k)$.

Let us stress the fact that the step-size α_k satisfying (4.2) and (4.3) exists without Assumption 3.3, as this can be observed by using the same argument as in unconstrained optimization (assuming the boundedness of Θ_{σ_k} from below on the manifold $\mathcal{M}_{x_k}$). This assumption was only used to view the path $\alpha \mapsto \bar{p}_k(\alpha)$ as the image by ψ_k of a straight line in the reduced space.

Two operations are necessary to derive an implementable algorithm from the path $\bar{p}_k$. First a discretization of $\bar{p}_k$ must be introduced. This one can be done such that a sufficient decrease of the merit function Θ_{σ_k} and the reduced curvature condition hold also along the discretized path. Next, a restoration step r_k (see (1.14)) has to be added to complete the iteration, in order to force the decrease of the norm of the constraints. This approach was developed in [27].

4.2 Longitudinal and transversal search

For some applications, the algorithm outlined in the last paragraph may be too expensive to use. Indeed, even if a single step is performed in the longitudinal part of the algorithm (formed by the discretization of the path $\bar{p}_k$), the functions have to be linearized twice per iteration: at x_k and $x_k + \alpha_k t_k$. In fact, one can view this algorithm as a globalization of the local reduced quasi-Newton algorithm, in which the matrix G_k is updated with the pair (γ_k, δ_k) defined by (2.3), requiring the additional linearization at the intermediate point $x_k + t_k$.

A way of avoiding this additional linearization is to add a transversal component (in the range space of A^-) to the derivative of $\bar{p}_k$ and to define a new search path $\tilde{p}_k$ by

$$\begin{cases} \tilde{p}_k'(\alpha) = Z^-(\tilde{p}_k(\alpha))\delta_k - A^-(\tilde{p}_k(\alpha))c(\tilde{p}_k(\alpha)) \\ \tilde{p}_k(0) = x_k, \end{cases} \tag{4.4}$$

where $\delta_k = -G_k^{-1}g(x_k)$, as before. The first order approximation of this path is $\alpha \mapsto x_k + \alpha(t_k + r_k^0)$, where r_k^0 is defined by (1.17). This is a simplified version of the reduced Hessian algorithm, r_k^0 being used instead of r_k.

Under the assumptions

$$g'A^- \equiv 0 \quad \text{and} \quad g'Z^- \text{ is constant on the reduced gradient manifolds,}$$

one has $g(\bar{p}_k(\alpha)) = g(\tilde{p}_k(\alpha))$, as long as both paths $\bar{p}_k$ and $\tilde{p}_k$ exist [29]. In this case, the same interpretation holds for a search along $\tilde{p}_k$ and $\bar{p}_k$: realizing $g(\tilde{p}_k(\alpha_k))^\top\delta_k \geq \omega_2 g(x_k)^\top\delta_k$ is equivalent to realizing (4.1) in the reduced space. These conditions are used to give an interpretation to the search along $\tilde{p}_k$ in terms of a search along a straight line. Now they are not necessary to be able to find an adequate step-size along $\tilde{p}_k$, as shown by the following proposition [29].

Proposition 4.1. *Suppose that the path $\alpha \mapsto \tilde{p}_k(\alpha)$ defined by (4.4) exists for a 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*

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

$$g(\tilde{p}_k(\alpha))^\top Z(x_k)t_k \geq \omega_2\, g(x_k)^\top Z(x_k)t_k \tag{4.5}$$

are satisfied for some $\alpha > 0$.

Again, the *reduced curvature condition* (4.5) is very attractive, since after discretization of the search path, it provides the desired inequality $\gamma_k^\top \delta_k > 0$, this time with $\gamma_k = g(x_{k+1}) - g(x_k)$, hence without having to linearize the functions at the intermediate point $x_k + t_k$.

Like for the longitudinal search, it remains to discretize the path $\tilde{p}_k$ to obtain an implementable algorithm. Let us denote by $\alpha_k^0 = 0 < \alpha_k^1 < \cdots < \alpha_k^{i_k} = \alpha_k$ the discretization step-sizes, which are not given a priori but computed as explained below, and by x_k^i the points approximating $\tilde{p}_k(\alpha_k^i)$ (with $x_k^0 = x_k$ and $x_k^{i_k} = x_{k+1}$). An explicit Euler approximation gives

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

where

$$d_k^i = Z^-(x_k^i)\delta_k - A^-(x_k^i)c(x_k^i).$$

For $i = 1, \ldots, i_k - 1$, the step-size α_k^{i+1} is determined such that Θ_{σ_k} decreases sufficiently:

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

The first intermediate point x_k^1 is determined in a slightly different way, so that, when the unit step-size is accepted, $x_k^1 = x_k + t_k + r_k$ and the local method is recovered, allowing the q-superlinear convergence of the algorithm; see [28] for the details. Once α_k^{i+1} has been computed, the reduced curvature condition is tested at x_k^{i+1}:

$$g(x_k^{i+1})^\top \delta_k \geq \omega_2 g(x_k)^\top \delta_k.$$

If it holds the search is completed, otherwise the search is pursued along the new intermediate direction d_k^{i+1}. It can be shown that this PLS algorithm terminates in a finite number of trials.

At the intermediate points x_k^i, the functions defining the problem have to be linearized. This may look expensive, but this impression have to be appreciated in view of the following observations.

1. Even when intermediate linearizations occur, the displacement along d_k^i helps in decreasing Θ_{σ_k}, so that a progress to the solution is done and this displacement is not useless.
2. The PLS is only used when the update criterion is verified (see Section 2.1), because there is no need to have $\gamma_k^\top \delta_k > 0$ when the matrix G_k is not updated. If an update criterion of the form (2.7) is used, the transversal part of the displacement is small, so that the search path $\tilde{p}_k$ is close to the manifold

$\mathcal{M}_{x_k}$. As for unconstrained optimization, we have observed that in practice the unit step-size is then accepted most of the time, so that no intermediate points are necessary. This fact is corroborated by the asymptotic result below (Theorem 4.2, outcome (ii)).

To conclude this section, we mention a superlinear convergence result obtained with the algorithm described above, equipped with an update criterion that we describe now. This one is of the form (2.7) with a sequence $\{\mu_k\}$ ruled by the algorithm itself (not a sequence given *a priori* as in [39]). The idea behind this criterion is the following. The sequence μ_k in (2.7) need not be decreased too rapidly, otherwise the matrix G_k would be rarely updated, which would prevent superlinear convergence. This suggests not changing μ_k, i.e., taking $\mu_{k+1} = \mu_k$, when the update criterion does not hold at iteration k. This strategy looks safe in particular in the case when the update criterion is not satisfied after a given iteration, say k_0. Then μ_k is constant for $k \geq k_0$ and $t_k = O(\|r_k\|)$. It can be shown that the latter estimate readily implies the superlinear convergence. More difficult is the case when G_k is both updated infinitely often and kept unchanged infinitely often. For dealing with this situation, the following update criterion is suitable

$$\Big(\|c(x_k)\| + \|c(x_k + t_k)\|\Big) \leq \mu \, \|e^1_{k\ominus 2}\| \, \|\delta_k\|. \tag{4.6}$$

In (4.6), $\mu > 0$ is an arbitrary constant and $e^1_k = x^1_k - x_k$ is the step from x_k to the first intermediate iterate x^1_k. The index $k \ominus 2$ is used to express that $\mu_k = \mu \|e^1_{k\ominus 2}\|$ will not change when there is no matrix update. More precisely, $k \ominus 1$ is the greatest index less than k at which an update occurred. Hence, $(k+1) \ominus 1$ (the greatest index less than $k+1$ at which an update occurred) is the same index as $k \ominus 1$ if there is no matrix update at iteration k. For technical reasons, it is not the index $k \ominus 1$ that must be used in (4.6) but $k \ominus 2 := (k \ominus 1) \ominus 1$ (the greatest index less than $k \ominus 1$ at which an update occurred).

Theorem 4.2. *Let (x_*, λ_*) be a primal-dual solution of problem* (1.7) *such that*

$$c(x_*) = 0, \quad g(x_*) = 0, \quad \textit{and} \quad Z^-(x_*)^\top L_* Z^-(x_*) \ \textit{is positive definite.}$$

Suppose that there is an open convex neighborhood Ω of x_*, such that:

(a) *f and c are twice continuously differentiable on Ω;*

(b) *the Jacobian matrix $A(x)$ of c is surjective on Ω and the map $x \mapsto (A^-(x), Z^-(x))$ is Lipschitz continuous on Ω;*

(c) *g is differentiable on Ω with Lipschitz continuous derivative.*

Suppose also that the reduced quasi-Newton algorithm with BFGS matrix updates, PLS, and the update criterion (4.6) *described above generates a sequence of points $\{x^i_k\}_{k\geq 1, 0\leq i\leq i_k-1}$ in Ω different from x_* and converging to x_* in the sense that*

$$\Big(\max_{0\leq i\leq i_k-1} \|x^i_k - x_*\|\Big) \to 0, \quad \textit{when } k \to \infty.$$

Then, the following properties hold:

(*i*) *the sequences of matrices* $\{G_k\}_{k\geq 1}$ *and* $\{G_k^{-1}\}_{k\geq 1}$ *are bounded;*

(*ii*) *the ideal step-size is accepted eventually:* $i_k = 1$ *and* $\alpha_k = 1$ *for* k *large;*

(*iii*) *the sequence* $\{x_k\}_{k\geq 1}$ *converges q-superlinearly in two steps to* x_**;*

(*iv*) *the sequence* $\{x_k + t_k\}_{k\geq 1}$ *converges q-superlinearly to* x_**.*

This result given in [28] is, to our knowledge, the first extension of a similar result proved by Powell [40] for the BFGS algorithm in unconstrained optimization, at least with hypotheses as weak as those listed above. This is not an obvious result. For example, it may occur that the matrices G_k are not updated infinitely often, although the sequence $\{x_k + t_k\}$ still converges superlinearly. This theorem shows that q-superlinear convergence with quasi-Newton methods for minimizing equality constrained problems is possible with a single linearization of the constraints per iteration (for a result with two linearizations per iteration, see [9]) and without knowing the threshold value of the penalty parameter τ making the Hessian of the augmented Lagrangian positive definite (a superlinear result with this hypothesis for the SQP method is proved in [10]).

5 PLS for quasi-Newton-SQP algorithms

We have seen in Section 1.3 that, when $(\gamma_k, \delta_k) \in \mathbb{R}^{2n}$ is given by (2.9), the penalty parameter τ can be used to ensure the positive definiteness of M_{k+1} (by means of $\gamma_k^\top \delta_k > 0$), the matrix approximating the Hessian of the augmented Lagrangian, provided $\delta_k = x_{k+1} - x_k$ is not in the tangent space to the constraint manifold. A back-up strategy is proposed in [10] to deal with the case when $A(x_k)\delta_k$ is relatively near zero. In this case, δ_k is almost tangent to the constraint manifold and we have seen, with the longitudinal search of Section 4, that the PLS is appropriate to find a longitudinal displacement that can ensure the positive definiteness of matrices approximating $Z^-(x_*)^\top L_* Z^-(x_*)$. Here, we follow [1] and show how to combine the two ideas: controlling the "transversal part" of M_{k+1} by τ and its "longitudinal part" by the PLS algorithm.

The description of this approach requires some preliminaries. First, the presentation is simplified if we introduce the right inverse $\widehat{A}_k^-(x)$ of $A(x)$ associated with the quadratic program (1.9). Assume that the matrix $M_k \in \mathbb{R}^{n\times n}$ is positive definite in the null space of $A(x)$. Then, the quadratic program

$$\begin{cases} \min \frac{1}{2} d^\top M_k d \\ c + A(x)d = 0. \end{cases} \tag{5.1}$$

has a unique solution, which is given by the optimality conditions of (5.1): for some $\lambda \in \mathbb{R}^m$,

$$\begin{cases} M_k d + A(x)^\top \lambda = 0 \\ A(x)d = -c. \end{cases} \tag{5.2}$$

This shows that the solution d is a linear function of the vector $c \in \mathbb{R}^m$. We denote by $-\widehat{A}_k^-(x)$ the $n \times m$ matrix representing this linear function (it depends on M_k, hence the index k in $\widehat{A}_k^-(x)$). Then, the solution of (5.1) can be written $-\widehat{A}_k^-(x)c$. Substituting this quantity in the second equation of (5.2) shows that $\widehat{A}_k^-(x)$ is a right inverse of $A(x)$. Suppose now that a tangent basis $Z^-(x)$ is given and that we denote by $\widehat{Z}_k(x)$ the unique $(n-m) \times n$ matrix satisfying

$$\widehat{A}_k^-(x)A(x) + Z^-(x)\widehat{Z}_k(x) = I_n.$$

Multiplying the first equation of (5.2) by $Z^-(x)^\top$ leads to

$$Z^-(x)^\top M_k \widehat{A}_k^-(x) = 0,$$

so that the solution of (1.9) can be written

$$d_k = Z^-(x_k)\widehat{Z}_k(x_k)d_k - \widehat{A}_k^-(x_k)c(x_k),$$

where $\widehat{Z}_k(x_k)d_k = -(Z^-(x_k)^\top M_k Z^-(x_k))^{-1}g(x_k)$.

It is also useful to compute the derivative of the multiplier estimate λ defined by (1.18). One has $\lambda(x) = -A^-(x)^\top \nabla f(x) = -A^-(x)^\top \nabla \ell(x, \lambda_*) + \lambda_*$, so that

$$\lambda'(x_*) = -A^-(x_*)^\top L_*. \tag{5.3}$$

We are now ready to present the approach of [1]. Since M_{k+1} has to approximate the Hessian of the augmented Lagrangian, (γ_k, δ_k) should verify

$$\gamma_k \simeq L_*\delta_k + \tau A(x_k)^\top A(x_k)\delta_k,$$

at the first order (see (2.1) and (2.8)). The first term of the sum in the right hand side is approximated, in (2.9), by the change in the gradient of the Lagrangian. Here, we split this term in two parts, for a reason that will be clear below. Using (1.16), (2.2), and (5.3):

$$\begin{array}{rcl} L_*\delta_k & = & Z(x_k)^\top Z^-(x_k)^\top L_*\delta_k + A(x_k)^\top A^-(x_k)^\top L_*\delta_k \\ & \simeq & Z(x_k)^\top (g'(x_*)\cdot\delta_k) - A(x_k)^\top(\lambda'(x_*)\cdot\delta_k) \\ & \simeq & Z(x_k)^\top(g(x_{k+1}) - g(x_k)) - A(x_k)^\top(\lambda(x_{k+1}) - \lambda(x_k)), \end{array}$$

provided $\delta_k \simeq x_{k+1} - x_k$. This approximate computation leads us to the following formula:

$$\gamma_k = Z(x_k)^\top(g(x_{k+1}) - g(x_k)) - A(x_k)^\top(\lambda(x_{k+1}) - \lambda(x_k)) + \tau A(x_k)^\top A(x_k)\delta_k,$$

where τ has to be adjusted.

The inner product of γ_k with δ_k is

$$\gamma_k^\top \delta_k = (g(x_{k+1}) - g(x_k))^\top Z(x_k)\delta_k - (\lambda(x_{k+1}) - \lambda(x_k))^\top A(x_k)\delta_k + \tau\|A(x_k)\delta_k\|_2^2.$$

Like for formula (2.9), one can make $\gamma_k^\top \delta_k$ positive by taking τ sufficiently large, provided $A(x_k)\delta_k \neq 0$. Recall that, when δ_k is tangent, the approach in [10] requires a back-up strategy. Here, when $A(x_k)\delta_k = 0$,

$$\gamma_k^\top \delta_k = (g(x_{k+1}) - g(x_k))^\top Z(x_k)\delta_k, \tag{5.4}$$

so that the positivity of $\gamma_k^\top \delta_k$ can be obtained by choosing the next iterate such that $g(x_{k+1})^\top Z(x_k)\delta_k > g(x_k)^\top Z(x_k)\delta_k$. This reminds us of the reduced curvature inequality (4.5) satisfied by the PLS in reduced quasi-Newton method. Here, this inequality may not be feasible, because $Z(x_k)\delta_k$ may not be a reduced descent direction, meaning that $g(x_k)^\top Z(x_k)\delta_k$ may not be negative when $A(x_k)\delta_k \neq 0$. Therefore, we ask instead to realize the following *reduced curvature condition*

$$g(x_{k+1})^\top \widehat{Z}_k(x_k)d_k \geq \omega_2 g(x_k)^\top \widehat{Z}_k(x_k)d_k,$$

where d_k is the SQP direction. Note that $Z(x_k)d_k = \widehat{Z}_k(x_k)d_k$ when $A(x_k)d_k = 0$, so that $\gamma_k^\top \delta_k > 0$ when $A(x_k)\delta_k = 0$ and δ_k is parallel to d_k. The following result is similar to proposition 4.1.

Proposition 5.1. *Suppose that the path $\alpha \mapsto p_k(\alpha)$ defined by*

$$\begin{cases} p_k'(\alpha) = Z^-(p_k(\alpha))\widehat{Z}_k(x_k)d_k - \widehat{A}_k^-(p_k(\alpha))c(p_k(\alpha)) \\ p_k(0) = x_k \end{cases}$$

exists for a sufficiently large step-size $\alpha \geq 0$. Suppose also that Θ_{σ_k} is bounded from below along this path, that $\sigma_k \geq \|\lambda_k^{\mathrm{QP}}(p_k(\alpha))\|_D$ whenever $p_k(\alpha)$ exists, that M_k is positive definite, and that $\omega_2 \in (0,1)$. Then, the inequalities

$$\Theta_{\sigma_k}(p_k(\alpha)) \leq \Theta_{\sigma_k}(x_k)$$

$$g(p_k(\alpha))^\top \widehat{Z}_k(x_k)d_k \geq \omega_2\, g(x_k)^\top \widehat{Z}_k(x_k)d_k$$

are satisfied for some $\alpha > 0$.

Like for the reduced quasi-Newton algorithm, an explicit Euler discretization of the path p_k is introduced: $\alpha_k^0 = 0 < \alpha_k^1 < \cdots < \alpha_k^{i_k} = \alpha_k$ are the discretization step-sizes determined as explained below and the points x_k^i are approximations of $p_k(\alpha_k^i)$ (with $x_k^0 = x_k$ and $x_k^{i_k} = x_{k+1}$) given by

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

where

$$d_k^i = Z^-(x_k^i)\widehat{Z}_k(x_k)d_k - \widehat{A}^-(x_k^i)c(x_k^i).$$

For $i = 0, \dots, i_k - 1$, the step-size α_k^{i+1} is determined such that Θ_{σ_k} decreases sufficiently (σ_k^i may need to be adapted before the determination of α_k^{i+1}, in order to make d_k^i a descent direction):

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

Once α_k^{i+1} has been computed, the reduced curvature condition is tested at x_k^{i+1}:

$$g(x_k^{i+1})^\top \widehat{Z}_k(x_k) d_k \geq \omega_2 g(x_k)^\top \widehat{Z}_k(x_k) d_k. \tag{5.5}$$

If it holds the search is completed, otherwise the search is pursued along the new intermediate direction d_k^{i+1}. It can be shown that this PLS algorithm terminates in a finite number of trials.

Let us describe more precisely how δ_k is computed. On the one hand, we have said that it is necessary to take $\delta_k \simeq x_{k+1} - x_k$. On the other hand, one needs to control the positivity of $\gamma_k^\top \delta_k$, even when $A(x_k)d_k = 0$. In our algorithm, $\gamma_k^\top \delta_k$ is given by (5.4) and the positivity of this inner product is not guaranteed by the reduced curvature condition (5.5) when $\delta_k = x_{k+1} - x_k$, since this vector may not be parallel to d_k. Therefore, we prefer to take the following approximation of $x_{k+1} - x_k$:

$$\delta_k = \alpha_k Z^-(x_k)\widehat{Z}_k(x_k)d_k - \alpha_k^{\mathrm{A}} \widehat{A}_k^-(x_k)c(x_k), \tag{5.6}$$

where

$$\alpha_k^{\mathrm{A}} = \sum_{i=0}^{i_k - 1} (\alpha_k^{i+1} - \alpha_k^i) e^{-\alpha_k^i}.$$

This formula aims at taking into account the fact that the value of c at x_k^i is used in the search directions d_k^i, while $c(x_k)$ is used in δ_k. It is based on the observation that along the path p_k, we have $c(p_k(\alpha)) = e^{-\alpha} c(x_k)$, which gives after discretization: $c(x_k^i) \simeq e^{-\alpha_k^i} c(x_k)$.

Let us now check that the form (5.6) of δ_k is appropriate. Suppose that $A(x_k)\delta_k = 0$. Then, $c(x_k) = 0$, $\delta_k = \alpha_k Z^-(x_k)\widehat{Z}_k(x_k)d_k$, and

$$\begin{aligned}
\gamma_k^\top \delta_k &= (g(x_{k+1}) - g(x_k))^\top Z(x_k)\delta_k \\
&= \alpha_k (g(x_{k+1}) - g(x_k))^\top \widehat{Z}_k(x_k) d_k \\
&> 0,
\end{aligned}$$

by the reduced curvature condition (5.5), which is satisfied for $i = i_k - 1$ (in which case $x_k^{i+1} = x_{k+1}$).

The conclusion of this discussion is that for any $k \geq 1$, one can find a (finite) $\tau \geq 0$ such that $\gamma_k^\top \delta_k > 0$, either because $A(x_k)\delta_k \neq 0$ or because $A(x_k)\delta_k = 0$ and $\gamma_k^\top \delta_k > 0$ by the reduced curvature condition (5.5). More details, including a global convergence result and numerical experiment, are given in [1].

6 Conclusion

By way of conclusion, we list some open problems and questions raised in this paper.

First, it would be very useful to have an update criterion based on the difference between $g(x_{k+1})$ and $g(x_k+t_k)$ instead of the criterion (2.7) based on the comparison between the norm of the tangent step t_k and the norm of the restoration step r_k, as explained in Section 1.3.

Another problem that would deserve more attention is to give a geometrical interpretation to the piecewise line-search introduced in Section 4 and 5, in the case when Assumption 3.3 does not hold. The PLS can still find a step-size satisfying reduced curvature conditions but its interpretation as a usual search along a straight line in the reduced space no longer holds.

As shown by the discussion in Section 3, γ_k given by formula (2.5) or the second formula in (2.6) may be more appropriate, in particular when the tangent bases are chosen orthonormal. In this case, can we still use the PLS approach presented here for a γ_k given by (2.3) or the first formula in (2.6)?

The q-superlinear convergence of the BFGS version of the SQP algorithm is still an open problem, for hypotheses similar to those of Theorem 4.2. Can a technique ensuring $\gamma_k^\top \delta_k > 0$ at each iteration, such as the PLS technique, be helpful in getting this result?

References

[1] P. Armand, 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 *Computational Optimization and Applications*).

[2] P. Armand, J.Ch. Gilbert (1996). A line-search technique with sufficient decrease and curvature conditions. Rapport de recherche (to appear), INRIA, BP 105, 78153 Le Chesnay, France.

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

[4] L.T. Biegler, J. Nocedal, C. Schmid (1995). A reduced Hessian method for large-scale constrained optimization. *SIAM Journal on Optimization*, 5, 314–347.

[5] P.T. Boggs, J.W. Tolle (1995). Sequential quadratic programming. In *Acta Numerica 1995*, pages 1–51. Cambridge University Press.

[6] W. Boothby (1975). *An Introduction to Differentiable Manifolds and Differential Geometry.* Academic Press, New York.

[7] R. Byrd, J.Ch. Gilbert, J. Nocedal (1996). A trust region method based on interior point techniques for nonlinear programming. Rapport de Recherche 2896, INRIA, BP 105,

78153 Le Chesnay, France. Http server `http://www.inria.fr/RRRT/RR-2896.html`; ftp server `ftp://ftp.inria.fr/INRIA/publication/RR`, file `RR-2896.ps.gz` (submitted to *Mathematical Programming*).

[8] R.H. Byrd (1990). On the convergence of constrained optimization methods with accurate Hessian information on a subspace. *SIAM Journal on Numerical Analysis*, 27, 141–153.

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

[10] 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.

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

[12] 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.

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

[14] T.F. Coleman, Y. Li (1993). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization* (to appear).

[15] T.F. Coleman, W. Yuan (1995). A quasi-Newton L_2-penalty method for minimization subject to nonlinear equality constraints. Technical Report 95-1481, Department of Computer Science, Cornell University, Ithaca, New York 14853.

[16] L. Conlon (1993). *Differentiable Manifolds – A first Course*. Birkhauser, Boston.

[17] J.E. Dennis, M. Heinkenschloss, L.N. Vicente (1994). Trust-region interior-point SQP algorithms for a class of nonlinear programming problems. Technical Report 94-45, Department of Computational and Applied Mathematics, Rice University.

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

[19] J.E. Dennis, R.B. Schnabel (1989). A view of unconstrained optimization. In G.L. Nemhauser, A.H.G. Rinnooy Kan, M.J. Todd (editors), *Handbooks in Operations Research and Management Science*, volume 1: Optimization, chapter 1, pages 1–72. Elsevier Science Publishers B.V., North-Holland.

[20] M.P. do Carmo (1993). *Riemannian Geometry*. Birkhauser, Boston.

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

[22] R. Fletcher (1987). *Practical Methods of Optimization* (second edition). John Wiley & Sons, Chichester.

[23] R. Fletcher (1994). An overview of unconstrained optimization. In E. Spedicato (editor), *Algorithms for continuous Optimization – The State of the Art*, NATO ASI Series C: Mathematical and Physical Sciences, Vol. 434, pages 109–143. Kluwer Academic Publishers.

[24] J.Ch. Gilbert (1986). *Sur quelques problèmes d'identification et d'optimisation rencontrés en physique des plasmas.* PhD Thesis, University Pierre et Marie Curie (Paris VI), Paris.

[25] 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.

[26] J.Ch. Gilbert (1989). On the local and global convergence of a reduced quasi-Newton method. *Optimization*, 20, 421–450.

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

[28] 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`.

[29] J.Ch. Gilbert (1996). On the realization of the Wolfe conditions in reduced quasi-Newton methods for equality constrained optimization. *SIAM Journal on Optimization, 7* (to appear).

[30] P.E. Gill, W. Murray, M.A. Saunders, M.H. Wright (1989). Constrained nonlinear programming. In G.L. Nemhauser, A.H.G. Rinnooy Kan, M.J. Todd (editors), *Handbooks in Operations Research and Management Science*, volume 1: Optimization, chapter 3, pages 171–210. Elsevier Science Publishers B.V., North-Holland.

[31] P.E. Gill, W. Murray, M.H. Wright (1981). *Practical Optimization.* Academic Press, New York.

[32] S.T. Glad (1979). Properties of updating methods for the multipliers in augmented Lagrangians. *Journal of Optimization Theory and Applications*, 28, 135–156.

[33] C.B. Gurwitz (1994). Local convergence of a two-piece update of a projected Hessian matrix. *SIAM Journal on Optimization*, 4, 461–485.

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

[35] W. Hoyer (1986). Variants of the reduced Newton method for nonlinear equality constrained optimization problems. *Optimization*, 17, 757–774.

[36] 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.

[37] J.J. Moré, D.J. Thuente (1994). Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software*, 20, 286–307.

[38] J. Nocedal (1992). Theory of algorithms for unconstrained optimization. In *Acta Numerica 1992*, pages 199–242. Cambridge University Press.

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

[40] M.J.D. Powell (1976). Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In R.W. Cottle, C.E. Lemke (editors), *Nonlinear Programming*, SIAM-AMS Proceedings 9. American Mathematical Society, Providence, RI.

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

[42] M.J.D. Powell (1978). The convergence of variable metric methods for nonlinearly constrained optimization calculations. In O.L. Mangasarian, R.R. Meyer, S.M. Robinson (editors), *Nonlinear Programming 3*, pages 27–63.

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

[44] 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.

[45] M.J.D. Powell (1988). A review of algorithms for nonlinear equations and unconstrained optimization. In *Proceedings of the First International Conference on Industrial and Applied Mathematics*. SIAM, Philadelphia.

[46] M.J.D. Powell (1991). A view of nonlinear optimization. In J.K. Lenstra, A.H.G. Rinnooy Kan, A. Schrijver (editors), *History of Mathematical Programming, A Collection of Personal Reminiscences*. CWI North-Holland, Amsterdam.

[47] M. Spivak (1979). *A Comprehensive Introduction to Differential Geometry*. Publish or Perish.

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

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

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

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