SOME RECENT DEVELOPMENTS IN NONLINEAR OPTIMIZATION ALGORITHMS

A. Sartenaer[1]

Abstract. This article provides a condensed overview of some of the major today's features (both classical or recently developed), used in the design and development of algorithms to solve nonlinear continuous optimization problems. We first consider the unconstrained optimization case to introduce the line-search and trust-region approaches as globalization techniques to force an algorithm to converge from any starting point. We then focus on constrained optimization and give the main ideas of two classes of methods, the Sequential Quadratic Programming (SQP) methods and the interior-point methods. We briefly discuss why interior-point methods are now so popular, in their primal-dual version, while they have been abandoned about twenty years ago. We also introduce a newly emerging alternative, called filter method, to the use of a merit function as a tool to measure progress from one iteration to the next in constrained optimization. We relate some of the most widely used nonlinear optimization solvers to the algorithmic features presented, and we finally give some useful tools for an easy and comprehensive access to recent developments in nonlinear optimization algorithms and to practical solvers and their performance.

Résumé. Nous présentons une sélection des principales stratégies (classiques ou récentes) utilisées à l'heure actuelle dans la conception et le développement d'algorithmes pour la résolution de problèmes d'optimisation continus non linéaires. Nous considérons tout d'abord le cas sans contraintes, afin d'introduire les techniques de recherche linéaire et de région de confiance, techniques de globalisation permettant de garantir la convergence d'un algorithme à partir de n'importe quel point de départ. Nous donnons ensuite les idées principales, dans le cadre de l'optimisation avec contraintes, de deux classes de méthodes : les méthodes de Programmation Quadratique Successive (PQS) et les méthodes de points intérieurs. Nous expliquons brièvement pourquoi les méthodes de points intérieurs sont actuellement si populaires, particulièrement dans leur version primale-duale, alors qu'elles avaient été laissées à l'abandon voici une vingtaine d'années. Nous introduisons d'autre part une récente alternative à l'utilisation d'une fonction de mérite comme outil de mesure, d'une itération à l'autre, de la progression d'un algorithme d'optimisation avec contraintes. Cette alternative porte le nom de méthode des filtres. Finalement, nous indiquons, pour certains logiciels d'optimisation non linéaire parmi ceux les plus utilisés à ce jour, les caractéristiques algorithmiques présentées dans ce travail qui les distinguent. Nous donnons quelques indications utiles pour un accès aisé et compréhensif aux développements récents en optimisation non linéaire, ainsi qu'à des logiciels récents et à leurs profils de performance.

INTRODUCTION

The development of numerical algorithms for the solution of (large and nonconvex) nonlinear optimization problems has been and still is a very important and active research field. As pointed out in [25] and [27],

[1] Department of Mathematics, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium,
e-mail: `annick.sartenaer@fundp.ac.be`

this field may now be considered as mature, offering the possibility to solve very large nonlinear optimization problems. Clearly the "interior-point revolution" has highly contributed to revitalizing this discipline and to encourage the development, these last years, of state-of-the-art algorithms and software. The recent advances in the field have also lead to an intense research activity in the solution of challenging or newly emerging types of problems, like optimization problems with equilibrium constraints (MPEC) or complementary constraints (MPCC), PDE and DEA-constrained optimization problems (i.e., problems whose constraints are partial differential equations or differential-algebraic equations), or problems involving both discrete and continuous variables (MINLP). We could also mention large-scale semidefinite programming problems, global optimization problems, nondifferentiable optimization problems, bilevel optimization problems, nonlinear stochastic problems, etc.

Our purpose in this paper is to propose a comprehensive overview of some of the main or most recent developments in the nonlinear optimization field, focussing on the principal algorithmic features on which are built today's nonlinear optimization solvers like SNOPT [22], LOQO [40], KNITRO [8], filterSQP [18] and IPOPT [41].

Throughout this paper, we will consider continuous nonlinear optimization problems of the form:

$$\text{NLP} \equiv \left\{ \begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) = 0 \\ & c(x) \geq 0, \end{array} \right. \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$ and $c : \mathbb{R}^n \to \mathbb{R}^p$ are assumed to be twice continuously differentiable. Calling a *feasible point* a vector x satisfying the constraints of (1), the set of all such point is called the *feasible set*. Nonlinear optimization algorithms generally aim at finding a *local solution* of problem (1), that is a feasible point x^* such that $f(x^*) \leq f(x)$ for all feasible x in a neighbourhood of x^*. For general nonlinear (nonconvex) problems indeed, the task of finding a *global solution* (a feasible point x^* such that $f(x^*) \leq f(x)$ for each feasible point x) may be very difficult to identify and to locate (see [19] for a discussion on this topic).

In order to compute or approach a local solution of problem (1), nonlinear optimization algorithms produce a sequence of iterates, $\{x_k\}$ say, that (hopefully) converges to such a solution, x^* say. As far as *unconstrained optimization* is concerned (i.e., when the feasible set is $\mathbb{R}^n$), important progress has been made since a while now in the solution of these problems, especially in the way to guarantee (both in theory and in practice) *global convergence* of the iterative process to a solution, i.e., convergence of the iterative process *from any starting point*. We will present in this paper the main ideas of the two major *globalization techniques*, called *line-search method* and *trust-region method*.

The solution of *constrained optimization* problems is more complex. When the problem involves inequality constraints, the difficulty of dealing with constraints is compounded by the combinatorial problem of identifying the set of *active inequality constraints* at the solution (i.e., the set of inequality constraints which are satisfied as equalities at the solution). Algorithmic advances to solve constrained optimization problems have been rather slow until the interior-point revolution, which started with the linear programming method proposed by Karmarkar in 1984 [31] to fasten the simplex method. The interior-point revolution has revitalized the area of nonlinear optimization, especially the competition between *active-set methods* and *interior-point methods* (see [25] for a discussion on the current state of active-set and interior-point methods for constrained optimization). We will draw here the main ideas of *Successive Quadratic Programming (SQP) methods* (which are of active-set type), and of *interior-point methods*, in particular *primal-dual interior-point methods*, which are by far the most popular interior-point methods today.

Globalization techniques such as those used in unconstrained optimization to ensure convergence of the iterative process from a general starting point are also widely used in constrained optimization. Classically, their adaptation to the constrained case requires the use of a *merit function*, designed to quantify the balance between the conflicting goals of reducing the objective function and satisfying the constraints in (1). We will discuss the use of a merit function in conjunction with line-search and trust-region techniques to force global convergence.

The balance between different goals in a merit function relies on the use of a *penalty parameter* whose initial choice and update are rather tricky. An alternative to the use of a merit function, recently proposed by Fletcher and Leyffer [18], is based on the concept of *filter* and dispenses with the need to handle penalty parameters. We will briefly present this alternative called *filter method*.

Along the way, we will (sometimes very shortly) outline performing state-of-the-art nonlinear optimization solvers (SNOPT [22], LOQO [40], KNITRO [8], filterSQP [18] and IPOPT [41]), in order to describe their main algorithmic features at the light of the algorithmic ideas covered in this paper. We will finally present some useful tools for the reader interested in an easy and comprehensive access to recent developments and packages related to nonlinear optimization.

1. Globalization techniques in unconstrained optimization

In order to introduce the line-search and trust-region techniques, we first introduce some background on *Newton's method*.

1.1. **Newton's method**

Newton's method is an iterative method for finding a solution to a system of n *nonlinear equations* of n variables. In many existing methods for solving nonlinear optimization problems, these nonlinear equations form the gradient of some twice continuously differentiable function f. That is, if we have to solve[1] the unconstrained minimization problem:

$$\min_{x \in \mathbb{R}^n} f(x),$$

a **first-order necessary optimality condition** for x^* to be a local minimizer of f is that $\nabla \mathbf{f}(\mathbf{x}^*) = \mathbf{0}$. In this context, Newton's method may be described as an iterative method for finding a point x^* such that $\nabla f(x^*) = 0$.

Using the Taylor's expansion of ∇f at a particular iterate x_k:

$$\nabla f(x_k + p) = \nabla f(x_k) + \nabla^2 f(x_k)p + r(x_k, p), \tag{2}$$

where $r(x_k, p)$ is the remainder term, the basic idea of Newton's method is to approximate ∇f by an affine function, that is, neglecting the remainder term. Given this linearization, the *Newton step* p_k^N from x_k is then determined so that the right-hand side of (2) is zero, that is, p_k^N satisfies the system of *linear* equations:

$$\nabla^2 f(x_k)p = -\nabla f(x_k), \tag{3}$$

known as the *Newton equations*.

If $\nabla^2 f(x_k)$ is nonsingular, p_k is the unique solution of (3). Consequently, if $\nabla^2 f(x)$ is nonsingular for all x, Newton's method is well defined and generates a sequence of iterates $\{x_k\}$ as given by the following algorithm.

Newton Algorithm.

    Step 0: The starting point $x_0 \in \mathbb{R}^n$ is given. Set $k = 0$.
    Step 1: Compute p_k^N solution of:

$$\nabla^2 f(x_k)p = -\nabla f(x_k),$$

    and set

$$x_{k+1} = x_k + p_k^N.$$

    Increment k by one and go to Step 1.

---

[1]Possibly as a subproblem of a nonlinear optimization method.

If $\nabla^2 f(x^*)$ is nonsingular and $\nabla^2 f$ is Lipschitz continuous in a neighbourhood of the solution x^*, then there exists a region surrounding x^* in which Newton's method converges, and the convergence is fast since the asymptotic rate of convergence is *quadratic*. For a complete discussion of Newton's method, see [14] or [37].

When Newton's method converges, it is generally agreed to be the most efficient method for solving a system of equations of the type $\nabla f(x) = 0$. However, the method may not converge from every starting point, and if at some iterate x_k the Hessian matrix $\nabla^2 f(x_k)$ is singular, (3) does not necessarily have a solution. Also, the need to solve a system of linear equations and to evaluate the second derivatives at each iteration may require a prohibitive amount of calculation if the dimension n is large[2]. Moreover, from an optimization viewpoint, what is desired is convergence to a point where not only the gradient vanishes, but where in addition the Hessian has some particular properties in order to ensure the minimizer character of the limit point, since **second-order sufficient optimality conditions** for x^* to be a local minimizer of f are that $\nabla \mathbf{f}(\mathbf{x}^*) = \mathbf{0}$ **and** $\nabla^2 \mathbf{f}(\mathbf{x}^*)$ **is positive definite**. Consequently, Newton's method without suitable modifications is generally not an appropriate method. We next introduce the line-search and trust-region techniques, whose goals are to enforce convergence of the iterative process to, at least, a *first-order critical point* of f (a point x^* that satisfies $\nabla f(x^*) = 0$) *from every starting point*, and to encourage convergence to a local minimizer by taking the *nonconvexity* of the objective function into account.

1.2. **Line-search methods**

The idea of the line-search method is simple: given a *descent direction* p_k (such that $\nabla f(x_k)^T p_k < 0$), a step length $\alpha_k \geq 0$ in this direction p_k is selected[3] that yields an "acceptable" next iterate $x_{k+1} = x_k + \alpha_k p_k$, that is, an iterate that *sufficiently* decreases the value of the objective function f. By "sufficiently", we mean that the decrease produced in f at each iteration should lead to the global convergence of the iterative process (under suitable assumptions, see Section 3.2 in [37]).

The ideal choice for α_k would be the solution of:

$$\min_{\alpha \geq 0} f(x_k + \alpha p_k), \tag{4}$$

but this value is in general too expensive to compute. On the other hand, the simple requirement "$f(x_k + \alpha_k p_k) < f(x_k)$" is not acceptable to guarantee the convergence of $\{x_k\}$ to a minimizer of f (see Section 3.1 in [37] for an illustration). Typical line-search algorithms generate a sequence of candidate values for α_k, stopping and accepting one of these values once certain conditions that enforce the desired sufficient decrease in f are satisfied. Note that a sufficient decrease condition alone is not enough to ensure reasonable progress by the algorithm and must be accompanied by a requirement that rules out unacceptably short steps. We list here some of the main conditions used in practice (see [4], [16] and [37], for instance, for a discussion of these rules).

(1) *Armijo condition:* Select fixed scalars $s > 0$, $\beta \in (0, 1)$ and $\sigma \in (0, \frac{1}{2})$ and set $\alpha_k = \beta^{n_k} s$ where n_k is the first nonegative integer n for which:

$$f(x_k) - f(x_k + \beta^n s p_k) \geq -\sigma \beta^n s \nabla f(x_k)^T p_k.$$

(2) *Wolfe conditions:* Select fixed scalars $\sigma \in (0, \frac{1}{2})$ and $\gamma \in (\sigma, 1)$, and choose α_k so that:

$$f(x_k) - f(x_k + \alpha_k p_k) \geq -\sigma \alpha_k \nabla f(x_k)^T p_k, \tag{5}$$

---

[2]We do not discuss these aspects here, see [37] for a discussion of the solution of the Newton equations and of alternatives for the evaluation (or approximation) of the Hessian matrix.

[3]The reason for the need to have a descent direction in line-search methods is easy to see. Consider the local Taylor's expansion of f around x_k:

$$f(x_k + \alpha p_k) = f(x_k) + \alpha \nabla f(x_k)^T p_k + \frac{1}{2}\alpha^2 p_k^T \nabla^2 f(x_k + t p_k) p_k,$$

for some $t \in (0, \alpha)$. At first order, a decrease in f from x_k in the direction p_k can be guaranteed for sufficiently short step lengths $\alpha > 0$ only if $\nabla f(x_k)^T p_k < 0$, i.e., if p_k is a descent direction.

and

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq \gamma \nabla f(x_k)^T p_k. \tag{6}$$

(3) *Strong Wolfe conditions:* Select fixed scalars $\sigma \in (0, \frac{1}{2})$ and $\gamma \in (\sigma, 1)$, and choose α_k so that:

$$f(x_k) - f(x_k + \alpha_k p_k) \geq -\sigma \alpha_k \nabla f(x_k)^T p_k, \tag{7}$$

and

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq \gamma |\nabla f(x_k)^T p_k|. \tag{8}$$

(4) *Goldstein conditions:* Select a fixed scalar $\sigma \in (0, \frac{1}{2})$ and choose α_k so that:

$$\sigma \leq \frac{f(x_k + \alpha_k p_k) - f(x_k)}{\alpha_k \nabla f(x_k)^T p_k} \leq 1 - \sigma.$$

Note that condition $\gamma > \sigma$ in the Wolfe (or strong Wolfe) conditions ensures that (5) and (6) (or (7) and (8)) can be satisfied simultaneously. Typical values of σ and γ are 10^{-4} and 0.9, respectively.

When a *backtracking* strategy is used, condition (6) needs not to be implemented, as this type of strategy avoids excessively small steps. The framework of such a strategy is the following.

Backtracking Algorithm.

Step 0: Choose $\sigma \in (0, \frac{1}{2})$ and l, u satisfying $0 < l < u < 1$. Set $\alpha = 1$.
Step 1: If:

$$f(x_k) - f(x_k + \alpha p_k) < -\sigma \alpha \nabla f(x_k)^T p_k,$$

then set:

$$\alpha = \rho \, \alpha \quad \text{for some } \rho \in [l, u], \tag{9}$$

and go to Step 1. Else set:

$$\alpha_k = \alpha,$$

and STOP.

The Armijo condition is a particular case of this last algorithm where $s = 1$ and $\rho = \beta$ at each iteration. Another strategy for reducing α (choosing ρ in (9)) is to minimize a quadratic (or even cubic) model of $\hat{f}(\alpha) = f(x_k + \alpha p_k)$, the one-dimensional restriction of f to the line through x_k in the direction p_k, using the current information about $\hat{f}$ to model it (see [14]).

Convergence results for the above list of conditions or strategy may be found in [4], [14] or [37]. Note that they not only depend on the choice of the step lengths but also on the choice of the search directions p_k. Restricting our attention to Newton-based methods, it is easy to show that if the Hessian matrix $\nabla^2 f(x_k)$ is positive definite for all k, then the Newton directions $p_k^N = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$ (see (3)) are descent directions. In this case, global convergence is guaranteed (see [37]). When the current Hessian is not positive definite, several approaches are used, among which the most popular are the so-called *modified line-search Newton methods*, in which the search direction p_k satisfies the linear system:

$$(\nabla^2 f(x_k) + E_k)p = -\nabla f(x_k),$$

where E_k is a positive semidefinite matrix (usually diagonal) chosen so that $\nabla^2 f(x_k) + E_k$ is sufficiently positive definite. Under suitable conditions, global convergence of such modified line-search Newton methods to a first-order critical point of f can be shown (see [37], Theorem 6.3). Similarly to the pure Newton method of Section 1.1, a quadratic rate of convergence is obtained asymptotically, under two conditions. Firstly, the modification $E_k = 0$ must be chosen for all k large enough, so that $p_k = p_k^N$ for all such k. This can be guaranteed if $\{x_k\}$ converges to a point x^* where $\nabla^2 f(x^*)$ is sufficiently positive definite. Secondly, a *full* step must be tried first, as suggested in the Backtracking Algorithm given above by setting $\alpha = 1$ in Step 0. Under

these conditions, modified line-search Newton methods reduce asymptotically to a pure Newton method and achieve the fast asymptotic rate of convergence of the latter. (See, e.g., [5, 6, 14, 16, 23, 29, 30, 37] for material on line-search methods and practical line-search algorithms.)

1.3. Trust-region methods

When the full Newton step is unsatisfactory, in order to achieve global convergence, the line-search approach will retain the same *step direction* and then search for an appropriate step along the line defined by this direction, selecting a *shorter step length*. The alternate approach to obtain global convergence is based on the observation that Newton's method models the objective function by a quadratic approximation, the Taylor's expansion of f around the current iterate x_k:

$$m_k(x_k + p) \stackrel{\text{def}}{=} f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p, \tag{10}$$

and that the full Newton step p_k^N satisfies the first-order necessary optimality condition for this model, i.e., $\nabla m_k(x_k + p_k^N) = 0$. If the model is strongly convex (that is, if $\nabla^2 f(x_k)$ is positive definite), p_k^N may then be interpreted as the global minimizer of $m_k(x_k + p)$. In this case, if the full step p_k^N does not produce a sufficient decrease in the objective function (and hence is not acceptable), it is so because this quadratic model does not adequately model f in a region containing this full step. The quadratic model being accurate only in a neighbourhood of the current iterate, the trust-region approach consists in choosing the next iterate to be an (approximate) minimizer of the quadratic model constrained to be in a *region* where we *trust* this quadratic model. Consequently, in trust-region methods, when we need to take a shorter step, we first choose a *shorter step length* and then use the full n-dimensional quadratic model to choose the *step direction*.

To obtain a step, we thus seek a solution p_k of the *trust-region subproblem*:

$$m_k(x_k + p_k) = \min\{m_k(x_k + p) \mid \|p\| \le \Delta_k\}, \tag{11}$$

for some given norm $\|\cdot\|$, and where $\Delta_k > 0$ is called the *trust-region radius*. Note that, in practice, p_k will be chosen as an *approximate* minimizer of problem (11) (see below). Once a step p_k is selected that guarantees a sufficient decrease in the model inside the trust region, the objective function is then evaluated at the new point $x_k + p_k$, called *candidate*. If f has decreased enough, the candidate is accepted as next iterate and the trust-region radius is possibly increased. Otherwise, the candidate is rejected and the trust-region radius is reduced. The updating of the iterate and of the trust-region radius are directly depending on a certain measure of agreement existing between the model and the objective function, namely, the quantity ρ_k below, defined as the *actual reduction*[4] in f divided by the *predicted reduction*, that is the reduction in m_k. The trust-region approach may be summarized by the following general scheme.

Trust-region Algorithm.

Step 0: An initial point x_0 and an initial trust-region radius Δ_0 are given, as well as the parameters η_1, η_2, α_1 and α_2 that satisfy:

$$0 \le \eta_1 < \eta_2 < 1 \text{ and } 0 < \alpha_1 < 1 < \alpha_2. \tag{12}$$

Set $k = 0$.

Step 1: Compute $m_k(x_k + p)$ and obtain p_k by (approximately) solving (11).

Step 2: Compute $f(x_k + p_k)$ and:

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}. \tag{13}$$

---

[4]Note that it could happen that f increases from x_k to $x_k + p_k$, so that a reduction in f is actually not guaranteed.

If $\rho_k \geq \eta_1$, then set $x_{k+1} = x_k + p_k$; otherwise, set $x_{k+1} = x_k$.

Step 3: Set:

$$\Delta_{k+1} = \begin{cases} \alpha_1 \|p_k\| & \text{if} \quad \rho_k < \eta_1, \\ \Delta_k & \text{if} \quad \eta_1 \leq \rho_k < \eta_2, \\ \max[\alpha_2 \|p_k\|, \Delta_k] & \text{if} \quad \rho_k \geq \eta_2. \end{cases}$$

Increment k by one, and go to to Step 1.

The algorithm depends on the constants η_1, η_2, α_1 and α_2. The values $\eta_1 = 0.25, \eta_2 = 0.75, \alpha_1 = 0.5$ and $\alpha_2 = 2$ are often used in practical implementations (see, for instance, [10] and [13]).

A nice feature of trust-region methods is that they allow to circumvent the difficulty caused by non positive definite Hessians in Newton's method without having to use modifications, like in line-search methods, since they do not need to use descent directions. Trust-region methods rather have the opportunity to naturally take advantage of directions of negative curvature when they exist, since directions of negative curvature can be followed safely up to the boundary of the trust region.

As for the determination of the step length α_k in the line-search approach, for which the exact solution of (4) is generally too expensive and replaced by some conditions of sufficient decrease in f, exact solution of the trust-region subproblem is also too expensive in general, and some condition of sufficient decrease *in the model* m_k is rather imposed on p_k. More precisely, if we define the *Cauchy point* x_k^{C} to be a model minimizer along the intersection of the steepest descent direction[5] and the trust region, we say that a step p_k produces a sufficient decrease in the model if and only if the reduction in the model satisfies:

$$m_k(x_k) - m_k(x_k + p_k) \geq \kappa \left[m_k(x_k) - m_k(x_k^{\text{C}}) \right], \tag{14}$$

for some $\kappa \in (0, 1)$. This condition is known to ensure global convergence of the algorithm, under suitable assumptions, to a first-order critical point of f (see [12], Section 6.4). Defining the *eigenpoint* x_k^{E} to be the point on the trust-region boundary along a direction of (approximately) minimal negative curvature, a stronger condition on the reduction in the model is:

$$m_k(x_k) - m_k(x_k + p_k) \geq \kappa \left[m_k(x_k) - \min \left[m_k(x_k^{\text{C}}), m_k(x_k^{\text{E}}) \right] \right], \tag{15}$$

with $\kappa \in (0, 1)$, which guarantees global convergence of the algorithm to a point at which **second-order necessary optimality conditions** hold, that is, to a point x^* for which $\nabla \mathbf{f}(\mathbf{x}^*) = \mathbf{0}$ and $\nabla^2 \mathbf{f}(\mathbf{x}^*)$ is positive semidefinite. Such a point is called a *second-order critical point* of f (see [12]).

When (11) is defined in term of the Euclidean norm, a necessary and sufficient condition for the step p_k to be a *global solution* of (11) is that it solves the system:

$$(\nabla^2 f(x_k) + \lambda I)p = -\nabla f(x_k), \tag{16}$$

for some scalar $\lambda \geq 0$ such that the matrix $\nabla^2 f(x_k) + \lambda I$ is at least positive semidefinite and:

$$\lambda(\Delta_k^2 - \|p_k\|^2) = 0.$$

(For a proof, see [12], Theorem 7.2.1 and Corollary 7.2.2, for instance.) Thus, if Δ_k is large enough and $\nabla^2 f(x_k)$ is positive definite, the solution of (11) is simply the Newton direction. Otherwise, the restriction on the norm will apply and $\|p_k\| = \Delta_k$. Methods for solving (16) where first suggested by Levenberg [32], Morrison [36] and Marquardt [33] in the context of nonlinear least-squares problems and then by Goldfeldt, Quandt and

---

[5]The steepest descent direction at x_k is the unit direction $d_k = -\nabla f(x_k)/\|\nabla f(x_k)\|_2$, solution to the problem:

$$\min_{d \in \mathbb{R}^n} \nabla f(x_k)^T d \quad \text{subject to} \quad \|d\|_2 = 1,$$

i.e., the direction along which the model (10) locally decreases at the fastest rate.

Trotter [24] for general nonlinear problems. All Levenberg–Morrison–Marquardt algorithms find a value $\lambda \geq 0$ such that $\nabla^2 f(x_k) + \lambda I$ is positive definite and solve (16) to determine p_k. There are many variations on this theme, and in some earlier methods, the precise restriction $\|p_k\| \leq \Delta_k$ on the length of p_k is not imposed. Instead, λ is used as the controlling parameter in the iteration and the length of p_k is determined by whatever value λ happens to take. Goldfeldt, Quandt and Trotter [24] then introduced an explicit updating procedure for the maximum stepsize Δ_k which is very similar to the procedure given in the Trust-region Algorithm above, controlling the iteration by using this radius Δ_k. In this case, equation (16) is regarded as defining a trajectory $p(\lambda)$, and the precise value of λ which makes $\|p(\lambda)\| = \Delta_k$ is sought to determine p_k. Note that practical methods usually approximate the curved trajectory $p(\lambda)$ by a piecewise linear path, such as the dogleg path or the conjugate gradient path for instance (see [12] and [37]).

The design of practical algorithms to compute an exact or approximate solution to the trust-region subproblem (11) is addressed in [12], Chapter 7. When an approximate solution is sought, the challenge is to find a point which is less expensive to compute than the exact model minimizer inside the trust region, but which is also "better" than the Cauchy point, in the sense that it not only ensures global convergence but also encourages a fast asymptotic rate of convergence. Note that if, for sufficiently large k, the model m_k is strongly convex and the exact solution of (11) is computed in Step 1 of the Trust-region Algorithm, then the trust-region constraint becomes asymptotically inactive in (11), allowing the algorithm to reach a quadratic rate of convergence. In this case indeed, the solution of (11) is the pure Newton step p_k^N, and the candidate $x_k + p_k^N$ is accepted for k large enough, similarly to the pure Newton method. For a comprehensive and detailed description of trust-region methods and their properties, see [12] and [35].

2. Constrained optimization

Consider now the constrained problem (1), that we restate here for clarity:

$$\text{NLP} \equiv \begin{cases} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) = 0 \\ & c(x) \geq 0, \end{cases} \tag{17}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$ and $c : \mathbb{R}^n \to \mathbb{R}^p$ are twice continuously differentiable. As for the unconstrained case, the design of algorithms to solve (17) relies on mathematical characterisations of a solution of (17). The topic of optimality conditions for nonlinearly constrained optimization is rather complicated and we present here only conditions that will be of interest in our algorithmic context. We refer the reader to [1, 12, 21, 28, 37], for instance, for a detailed discussion of optimality conditions.

We define the *Lagrangian function* for problem (17) as:

$$\mathcal{L}(x, y, z) = f(x) - y^T g(x) - z^T c(x),$$

where $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^p$ are vectors of *Lagrange multipliers* corresponding to the equality constraints and the inequality constraints, respectively. Consider now a local solution x^* of (17). Assuming that the constraints of (17) satisfy a *constraint qualification*[6] at x^*, first-order necessary optimality conditions for a local constrained minimizer are given by the *Karush–Kuhn–Tucker conditions* (*KKT conditions* for short): there exist $y^* \in \mathbb{R}^m$

---

[6]The most often used are the *linear independence constraint qualification* (LICQ), that assumes linear independency of the gradients of the active constraints at x^*, and the (weaker) *Mangasarian-Fromovitz constraint qualification* (MFCQ), see a description in [28] and [37], for instance.

and $z^* \in \mathbb{R}^p$ such that:

$$\nabla_x \mathcal{L}(x^*, y^*, z^*) = 0 \qquad \text{(stationarity)}, \tag{18}$$

$$g(x^*) = 0 \text{ and } c(x^*) \geq 0 \quad \text{(feasibility)}, \tag{19}$$

$$z^* \geq 0 \qquad \text{(nonnegativity of the multipliers)}, \tag{20}$$

$$c_i(x^*)z_i^* = 0, \;\; i = 1, \ldots, p \quad \text{(complementarity)}, \tag{21}$$

where $\nabla_x \mathcal{L}(x, y, z) = \nabla f(x) - \nabla g(x)^\top y - \nabla c(x)^\top z$, with $\nabla g(x)$ and $\nabla c(x)$ denoting the Jacobian matrices of g and c, respectively, at x.

Practical algorithms to solve (17) generate a sequence of guesses for a point x^*, and possibly for the Lagrange multipliers y^* and z^*, that satisfies the KKT conditions (18)–(21), with the hope to converge to a local constrained minimizer of (17). A given algorithm is characterized by the way these sequences are generated. In the next sections, we give the main algorithmic ideas for two well-known classes of methods, one based on Sequential Quadratic Programming (SQP) and the other one on interior-point strategies.

2.1. **Sequential Quadratic Programming methods**

The key idea of Sequential Quadratic Programming (SQP) is to model problem (17) at each iteration by an appropriate quadratic subproblem (i.e., a problem with a quadratic objective function and linear constraints), and to solve this subproblem to generate a next iterate. SQP methods are of the active-set variety, in the sense that, at each iteration, the set of active constraints at the solution of the quadratic subproblem supplies a guess of the correct active set of constraints at a solution of problem (17). SQP methods are the basis of some of the best software packages for solving constrained optimization problems, among which is SNOPT [22] (see Section 2.1.3).

2.1.1. *Motivation*

To motivate the choice of an appropriate quadratic subproblem at a given iteration k, consider first the equality-constrained problem:

$$\text{NLP}_\mathcal{E} \equiv \begin{cases} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) = 0, \end{cases} \tag{22}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions. The KKT conditions above, specialized to this problem, give the following $n + m$ nonlinear equations in the $n + m$ unknowns x and y:

$$\begin{bmatrix} \nabla_x \mathcal{L}(x, y) \\ g(x) \end{bmatrix} = \begin{bmatrix} \nabla f(x) - \nabla g(x)^\top y \\ g(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{23}$$

where $\mathcal{L}(x, y) = f(x) - y^T g(x)$. Let x_k and y_k being the current iterates, a natural way to solve the nonlinear equations (23) is to use *Newton's method*, yielding a Newton step, $(p_k^N, \triangle y_k^N)$, which satisfies the Newton equations (called *KKT system*):

$$\begin{pmatrix} \nabla_{xx}^2 \mathcal{L}(x_k, y_k) & -\nabla g(x_k)^\top \\ \nabla g(x_k) & 0 \end{pmatrix} \begin{pmatrix} p \\ \triangle y \end{pmatrix} = -\begin{pmatrix} \nabla_x \mathcal{L}(x_k, y_k) \\ g(x_k) \end{pmatrix}. \tag{24}$$

Under suitable assumptions (see [37], Section 18.1), the KKT matrix in (24) is nonsingular and the Newton iteration:

$$\begin{cases} x_{k+1} = x_k + p_k^N \\ y_{k+1} = y_k + \triangle y_k^N, \end{cases}$$

is well-defined and converges quadratically to a solution of problem (22), provided that the starting point is close enough to this solution.

Observe now that since $\nabla_x \mathcal{L}(x_k, y_k) = \nabla f(x_k) - \nabla g(x_k)^\top y_k$, the KKT system (24) is equivalent to the following system:

$$\begin{pmatrix} \nabla^2_{xx} \mathcal{L}(x_k, y_k) & -\nabla g(x_k)^\top \\ \nabla g(x_k) & 0 \end{pmatrix} \begin{pmatrix} p \\ y_k + \triangle y \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g(x_k) \end{pmatrix}. \tag{25}$$

The Newton step p_k^N can thus be identified with the solution of the quadratic program (under suitable assumptions again, see [37]):

$$\begin{array}{ll} \min_{p \in \mathbb{R}^n} & \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}(x_k, y_k) p + \nabla f(x_k)^T p \\ \text{s.t.} & \nabla g(x_k) p + g(x_k) = 0, \end{array} \tag{26}$$

with Lagrange multipliers $y_k + \triangle y_k^N$. Indeed, conditions (25) are exactly the KKT conditions for the quadratic program (26). Note that since the constraint of problem (26) is satisfied at p_k^N, we have that:

$$\nabla_x \mathcal{L}(x_k, y_k)^T p_k^N = \nabla f(x_k)^T p_k^N - y_k^T \nabla g(x_k) p_k^N = \nabla f(x_k)^T p_k^N + y_k^T g(x_k),$$

so that problem (26) can be transformed into the quadratic problem:

$$\begin{array}{ll} \min_{p \in \mathbb{R}^n} & \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}(x_k, y_k) p + \nabla_x \mathcal{L}(x_k, y_k)^T p \\ \text{s.t.} & \nabla g(x_k) p + g(x_k) = 0, \end{array} \tag{27}$$

where the linear term $\nabla f(x_k)^T p$ has been replaced by $\nabla_x \mathcal{L}(x_k, y_k)^T p$. In conclusion, Newton's method applied to (23) is *formally equivalent* to a Sequential Quadratic Programming approach and shows which choice is appropriate for the quadratic program in an SQP method: a quadratic approximation of the Lagrangian function and a linear approximation of the constraints around the current iterate. If p_k and λ_k denote the approximate solution and Lagrange multipliers, respectively, of this quadratic program, the next iterate in a local SQP method is then given by $x_{k+1} = x_k + p_k$ and $y_{k+1} = \lambda_k$.

We now turn to the general problem (17). Given an iterate (x_k, y_k, z_k), a natural extension of the above idea is to consider the following quadratic subproblem, where both the inequality and equality constraints are now linearized:

$$\text{SQP}_k \equiv \begin{cases} \min_{p \in \mathbb{R}^n} & \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}(x_k, y_k, z_k) p + \nabla f(x_k)^T p \\ \text{s.t.} & \nabla g(x_k) p + g(x_k) = 0 \\ & \nabla c(x_k) p + c(x_k) \geq 0. \end{cases} \tag{28}$$

Provided some assumptions are satisfied, quadratic convergence is obtained when starting the iterative process sufficiently close to a constrained minimizer of (17) (see [5] or [6], for instance). Moreover, an important property of the SQP method when inequality constraints are present in the problem is that the optimal set of active constraints at a solution of problem (17) is identified after a finite number of iterations (see [37]).

2.1.2. *Globalization strategies and merit functions*

As in the unconstrained case, convergence from a remote starting point cannot be guaranteed without some globalization technique and nonconvexity must be taken into account. In this section, we briefly describe how the ideas developed in Sections 1.2 and 1.3 can be adapted to globalize SQP methods.

Consider first a line-search approach. Similarly to the unconstrained case, line-search strategies rely on the capability of the method to compute a *descent direction* and on the satisfaction of certain rules of *sufficient decrease* along this direction. In the unconstrained case, both these conditions must be satisfied with respect to the objective function f, since the unique goal is to minimize this function. In the constrained case however, we need to make use of a so-called *merit function* to assess the quality of a candidate step length with respect to the two goals of reducing the objective function and satisfying the constraints. Such a merit function, denoted by ϕ, is thus designed to quantify the balance between these two goals. Before going further, we list here two instances of merit functions commonly used in practice, see [37] (for simplicity, we focus our attention on the equality-constrained problem (22)):

- The (nondifferentiable) ℓ_1 *merit function*:

$$\phi_1^\beta(x) = f(x) + \frac{1}{\beta} \sum_{i=1}^m |g_i(x)|, \tag{29}$$

where $\beta > 0$ is called the *penalty parameter*.
- The *Fletcher's augmented Lagrangian merit function*:

$$\phi_F^\beta(x) = f(x) - y(x)^T g(x) + \frac{1}{2\beta} \|g(x)\|_2^2, \tag{30}$$

where $\beta > 0$ is the penalty parameter and:

$$y(x) = [\nabla g(x) \nabla g(x)^\top]^{-1} \nabla g(x) \nabla f(x)$$

are *least-squares multiplier* estimates.

Consequently, the direction p_k generated by the solution of the SQP subproblem must be a descent direction *for ϕ*, and a set of line-search conditions such as those listed in Section 1.2 must be satisfied *by ϕ*. To get a descent direction for ϕ implies the need to impose conditions on the quadratic subproblem and on the merit function itself. Without going into detail, some convexity properties need first be satisfied by the quadratic model in (28), which implies to introduce appropriate modifications (or approximations) of the Hessian of the Lagrangian, $\nabla_{xx}^2 \mathcal{L}(x_k, y_k, z_k)$. Moreover, the penalty parameter β that appears in the definition of the merit function has to be sufficiently small. Depending on the merit function used, practical values for β at each iteration can be derived which, in conjunction with adequate convexity properties of the model's Hessian in (28), allow to establish global convergence results (see [37], Section 18.5). Note also that the merit functions (29) and (30) are *exact merit functions*, i.e., there is a positive scalar β^* such that for any $\beta \in (0, \beta^*]$, any local solution of the original problem is a local minimizer of $\phi_1^\beta(x)$ ($\phi_F^\beta(x)$, respectively) (see [37], Section 15.3).

We now turn to the globalization of SQP methods through a trust-region technique. In the unconstrained case, a *trust-region subproblem* is first solved (where a quadratic model of the objective function is approximately minimized inside a trust region) to produce a candidate. The acceptance of the candidate as the next iterate and the update of the trust region depends on the ratio ρ_k of the *actual reduction* in f to the *predicted reduction* by the model (see (13)). In order to adapt this approach to SQP methods, it is natural to consider the following trust-region subproblem:

$$\begin{aligned} \min_{p \in \mathbb{R}^n} \quad & \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}(x_k, y_k, z_k) p + \nabla f(x_k)^T p \\ \text{s.t.} \quad & \nabla g(x_k) p + g(x_k) = 0 \\ & \nabla c(x_k) p + c(x_k) \geq 0, \\ & \|p\| \leq \Delta_k, \end{aligned} \tag{31}$$

for some given norm $\| \cdot \|$, and where $\Delta_k > 0$. Since a trust-region bound is added on the step, there is no need to impose some convexity properties on the Hessian of the model. On the other hand, this restriction on the step size may cause the subproblem to become infeasible, and it may be necessary to introduce some relaxation of the constraints. We will come back to this feature in Section 2.2.4 (see also [12], Section 15.4, for a detailed description).

Similarly to the line-search approach, trust-region SQP methods have recourse to a merit function ϕ to assess the quality of the candidate step p_k (i.e., the approximate solution of (31)), with respect to the two goals of reducing the objective function and the infeasibility. More precisely, the numerator of ρ_k in (13) will be defined as the *actual reduction in ϕ* rather than in f. However, care must be taken in order for ρ_k to make sense, since ρ_k should reflect the level of adequacy between the merit function and some model closely connected to it, while the model:

$$m_k(x_k + p) = \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}(x_k, y_k, z_k) p + \nabla f(x_k)^T p$$

in (31) does not have this property. One way to proceed is, for instance, to augment, in the denominator of ρ_k in (13), the reduction predicted by the model by a weighted term (by means of a penalty parameter appropriately chosen), reflecting some predicted reduction in the constraint violation. Another possibility, once a merit function has been selected, is to adapt the quadratic subproblem (31) so as to be specifically related to the merit function (introducing a penalty term in the model $m_k(x_k + p)$, for instance). We refer the interested reader to [12], Section 15.3, and to [37], Section 18.8, for a discussion of the choice of merit functions in SQP methods and a study of global and asymptotic convergence properties.

A major drawback of line-search and trust-region methods based on the use of a merit function is the need to design a procedure to initialize and update the penalty parameter β, a tricky part whose role is crucial for the performance of a practical algorithm. An alternative to the use of a merit function to reach global convergence has been recently proposed by Fletcher and Leyffer [18] and is based on the concept of *filter*, which dispenses with the need to handle penalty parameters.

In *filter methods*, as they are called, a point is acceptable if it produces a sufficient decrease *either* in f *or* in the constraint violation (thus no more in a weighted combination of both, like in merit functions). Assume that a pair $(\vartheta(x), f(x))$ is associated to each point x, consisting of a measure of the constraint violation and the objective value at x. We could for instance define $\vartheta(\cdot)$ as the *maximum violation* at x:

$$\vartheta(x) = \max\left[\max_{i=1,\dots,m} |g_i(x)|, \max_{i=1,\dots,p} -c_i(x)\right].$$

We say that a point x *dominates* a point $\hat{x}$ if the constraint violation and the objective value at x are both less than or equal to the corresponding values at $\hat{x}$:

$$\vartheta(x) \leq \vartheta(\hat{x}) \text{ and } f(x) \leq f(\hat{x}).$$

Define now a *filter* as a list $\mathcal{F}$ of ordered pairs of the form (ϑ_i, f_i), each pair consisting of a measure of the constraint violation and the objective value at a particular point, with the property that no entry in the filter is *dominated* by any other entry:

$$\vartheta_i < \vartheta_j \text{ or } f_i < f_j \text{ for } i \neq j.$$

Doing so, the acceptability of a point generated by an algorithm can be handled by a filter: a new iterate is judged acceptable only if it is not dominated by any other point in the filter. Figure 1 illustrates the concept of a filter by showing four pairs (black dots) in the (ϑ, f) space. Any iterate whose associated (ϑ, f)-pair occurs above and to the right of the lines radiating from the four pairs in the filter is dominated by at least one entry in the filter and is thus not acceptable.

In practice, many details need to be specified and refined in order to produce an efficient and globally convergent filter algorithm. For instance, it would be inefficient to accept a point whose (ϑ, f)-pair is arbitrarily close to that of a point in the filter. A small "margin" is therefore considered below and to the left of the border of the dominated part of the (ϑ, f) space in Figure 1, in which points will also be rejected. We refer the reader to [12] and [17] for a detailed description.

It is usually agreed that filter methods are more permissive than methods using a merit function and allow to accept longer steps. Filter methods have been adapted to both line-search and trust-region techniques (see [12] and [18], for instance). As an example, the state-of-the-art software package filterSQP, developed by Fletcher and Leyffer [18], implements an SQP method globalized by a trust-region approach.

2.1.3. *SNOPT*

The state-of-the-art software package SNOPT has been developed by Gill, Murray and Saunders [22] for solving large-scale linear and nonlinear (non necessarily convex) optimization problems.

For nonlinear problems, SNOPT employs a (sparse) active-set SQP algorithm that uses convex quadratic programs, a line-search technique and a merit function of augmented Lagrangian type to encourage convergence
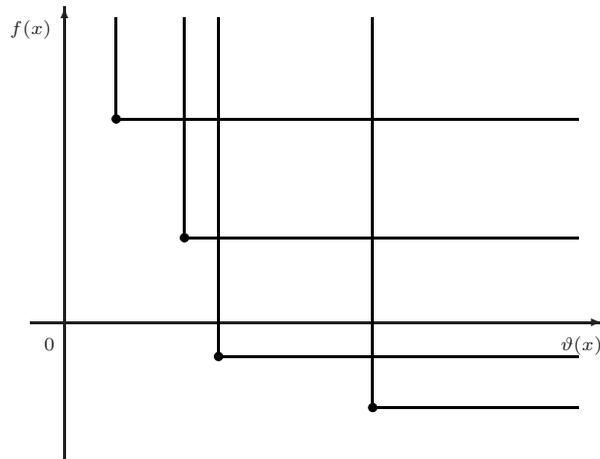
FIGURE 1. A filter with four pairs (source: Conn, Gould, Toint, [12]).

from an arbitrary starting point. It also uses limited-memory quasi-Newton approximations to the Hessian of the Lagrangian (see [37], Chapter 9).

SNOPT has of course a lot of others algorithmic features, among which the use of a modified Lagrangian function, of a *primal-dual*[7] augmented Lagrangian merit function and of elastic bounds to treat infeasible problems. We refer the interested reader to [22] for an in-depth description of SNOPT.

Note that SNOPT is especially effective for nonlinear problems whose functions and gradients are expensive to evaluate, for problems with a nonlinear objective function and large numbers of sparse linear constraints (as well as bounds on the variables), and for problems with few degrees of freedom at a solution (that is, if the number of active constraints at a solution is nearly as large as the number of variables).

2.2. Interior-point methods

Interior-point methods in nonlinear optimization are closely related, as we will see, to *barrier methods*. The latter solve an inequality-constrained problem by means of a sequential unconstrained minimization of a function combining f and a weighted barrier that prevents iterates from leaving the feasible set.

Classical barrier methods were first introduced during the 1960s, but turned out to be practically uninteresting. This lack of efficiency was attributed, at that time, to an increasing ill-conditioning of the Hessian matrix of the barrier function as the solution of the problem is approached. As we will point out in the next section, the culprit to this inefficiency is actually not the ill-conditioning of the Hessian matrix.

The announcement in 1984 of Karmarkar's polynomial-time method in linear programming [31] started the so-called "interior-point revolution." The close connection between this method and barrier methods, not only in linear programming but also in nonlinear programming, gave fresh impetus to the latter, resulting in a profusion of new algorithmic developments.

In this section, we will first motivate and illustrate the barrier technique. We then show the main drawback of the *classical Newton barrier approach* and present the main ideas of *primal-dual interior-point methods*. We finally describe the main characteristics of two software packages, KNITRO and LOQO.

---

[7]"Primal-dual" meaning that the variables of the problem and the Lagrange multipliers are treated as independent variables, see Section 2.2.3.

2.2.1. *Motivation*

Consider the inequality-constrained problem:

$$\text{NLP}_{\mathcal{I}} \equiv \begin{cases} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & c(x) \geq 0, \end{cases} \tag{32}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^p$ are smooth functions. Barrier methods generate a sequence of *feasible* iterates by successively minimizing composite functions designed to prevent feasible iterates from moving too close to the boundary of the feasible set. Today, the most widely used composite function is the *logarithmic barrier function*, which for the constraint set $c_i(x) \geq 0, i = 1, \ldots, p$, has the form:

$$B(x, \mu) = f(x) - \mu \sum_{i=1}^{p} \log c_i(x), \tag{33}$$

where $\mu > 0$ is the *barrier parameter*. Let $x(\mu)$ denote the unconstrained minimizer of $B(x, \mu)$. It can be shown that $c(x(\mu)) > 0$ (see [21]) and that, under suitable conditions, $x(\mu)$ *converges to* x^*, a local constrained minimizer of (32), when $\mu \searrow 0$. Moreover, $x(\mu)$ defines a *(local) unique and continuously differentiable path to x^** in the neighbourhood of $\mu = 0$ (see [21], Theorem 3.12). This path is called the *central path* or *barrier trajectory*, and is denoted by:

$$\mathcal{C} \stackrel{\text{def}}{=} \{x(\mu) \mid \mu > 0\}. \tag{34}$$

Figure 2 shows an example of central path for the inequality-constrained problem:

$$\min \quad \tfrac{10}{3} x_1 x_2 + \tfrac{1}{6} x_1$$

$$\text{s.t.} \quad \begin{cases} \tfrac{19}{16} - x_1^2 - \tfrac{5}{2} x_2^2 & \geq & 0, \\ x_1 - x_2 + \tfrac{3}{5} & \geq & 0. \end{cases} \tag{35}$$

We see from Figure 2 that there are two local minimizers of f in the feasible set for problem (35). The central path to one of them, $x^* = (\tfrac{3}{4}, -\tfrac{1}{2})$, is depicted in Figure 2. Figure 3 shows the contours of the logarithmic barrier function (33) for problem (35) for two different values of μ, $\mu = 1$ and $\mu = 0.1$. We see that, as μ becomes smaller, the barrier function becomes more and more like f, except very close to the boundary, where the logarithmic singularity has an effect. When $\mu = 0.1$, the barrier function has two local minimizers, like the original problem.

2.2.2. *The classical Newton barrier Method*

The *classical Newton barrier approach*, as proposed in the 1960s, amounts to apply a variant of *Newton's method* (globalized by a line-search or a trust-region technique) to compute, during an *inner iteration*, an unconstrained minimizer of the logarithmic barrier function $B(x, \mu)$ for a fixed value of μ. An *outer iteration* then tests convergence for the original problem and updates μ. This procedure generates a sequence of unconstrained minimizers of $B(x, \mu)$ for a sequence of decreasing barrier parameters μ, with the hope that this sequence converges to a local solution of (32).

The classical Newton barrier equations:

$$\nabla_{xx}^2 B(x, \mu) p = -\nabla_x B(x, \mu), \tag{36}$$

are explicitly given by:

$$\left(\nabla^2 f(x) - \sum_{i=1}^{p} \frac{\mu}{c_i(x)} \nabla^2 c_i(x) + \mu \nabla c(x)^T C(x)^{-2} \nabla c(x) \right) p$$

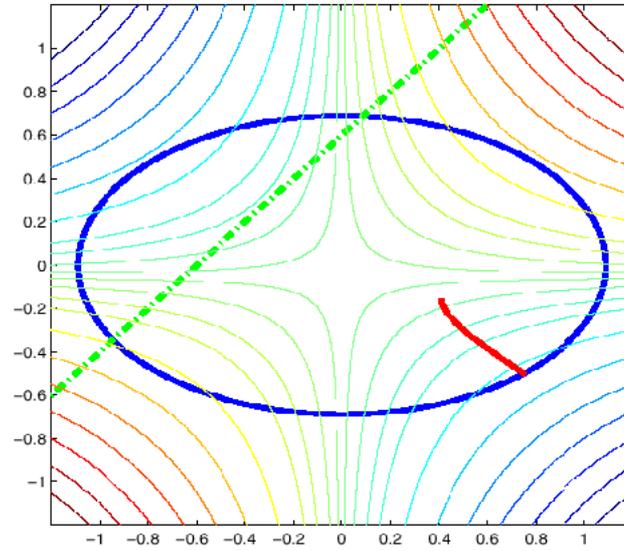$$= -\nabla f(x) + \mu \nabla c(x)^T C(x)^{-1} e, \tag{37}$$

FIGURE 2. A central path for problem (35), i.e., a trajectory of local unconstrained minimizers of the logarithmic barrier function that begins at the stricly feasible analytic center of the feasible set, corresponding to $\mu = \infty$, and converges to the boundary, as $\mu \to 0$ (source: Forsgren, Gill and Wright, [21]).
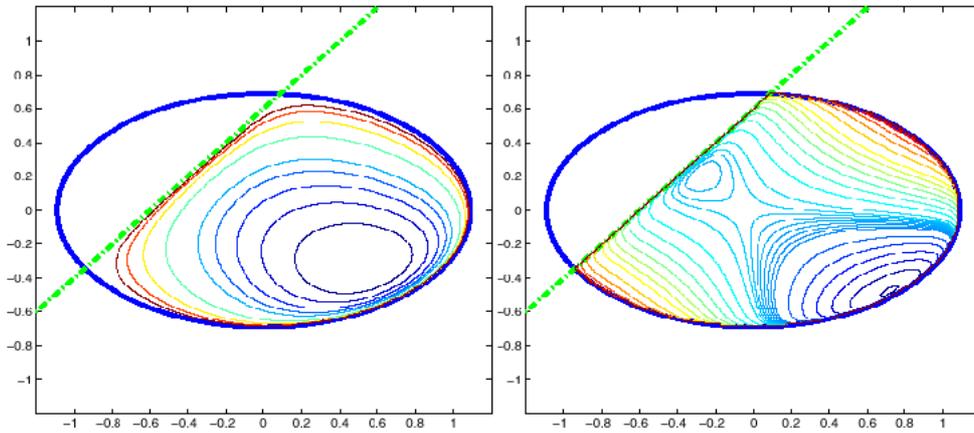

FIGURE 3. The contours of the logarithmic barrier function (33) for problem (35), for $\mu = 1$ and $\mu = 0.1$ (source: Forsgren, Gill and Wright, [21]).

where $C(x) = \mathrm{diag}(c_1(x), \ldots, c_p(x))$ and $e = (1, \ldots, 1)^T$. It is important to note that, even if the Hessian matrix, $\nabla^2_{xx} B(x, \mu)$, is increasingly ill-conditioned and asymptotically singular as $\mu \searrow 0$, the special structure of the Newton barrier equations (37) makes it possible to solve these equations with acceptable accuracy by a carefully implemented algorithm. Such practical issues of the classical Newton barrier approach are discussed in [21].

Yet, and this is also important to notice, the exact solution of the Newton barrier equations is nevertheless defective. We know indeed, from Section 1, that it is important for a Newton-based method to take exact Newton steps asymptotically, in order to achieve Newton's fast rate of convergence. However, as is shown in [21], Section 4.3.3 (see also [11] and [44]), taking a full Newton barrier step, even asymptotically, is likely to produce substantial infeasibility. More precisely, assume that the current iterate x is equal to the exact barrier minimizer $x(\mu)$ (or is close to it), and that the barrier parameter is reduced to $\bar{\mu}$, with $\bar{\mu} < \mu$. Then the exact Newton step, p^N, computed at $x(\mu)$ with the barrier parameter taken as $\bar{\mu}$ (i.e., the solution of (36) with $x \equiv x(\mu)$ and $\mu \equiv \bar{\mu}$), is likely to produce an infeasible point. In other words, the Newton barrier step is poorly scaled immediately after a reduction in the barrier parameter. This implies that the full Newton step must be shorten to produce a feasible step, and this prevents the method from reaching a fast asymptotic rate of convergence. This phenomenon is illustrated in Figure 4, which shows a zoom on the region near the solution of problem (35). An exact Newton step corresponding to $\mu = 0.0125$ and computed from the barrier minimizer $x(0.1)$ on the barrier trajectory is drawn that largely violates the first constraint in (35). We can thus conclude
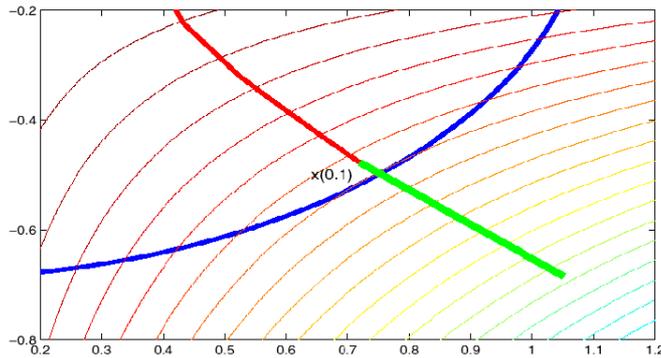

FIGURE 4. The exact Newton step for $\mu = 0.0125$ and starting from the barrier minimizer $x(0.1)$ (source: Forsgren, Gill and Wright, [21]).

that the classical Newton barrier approach is inefficient, even though the common belief that the culprit is the ill-conditioning of the Hessian of the barrier function is erroneous.

2.2.3. *Primal-dual interior-point methods*

The main goals of *primal-dual interior-point methods* are to exploit the properties of the central path – the existence of a continously differentiable "centered" path of approximate solutions that converges to the exact solution – while avoiding the drawback of the classical Newton barrier approach. For that, observe that the classical Newton barrier approach iterates on the variable x of the original problem *only* (this is why this method is called a *primal approach*). The key idea of *primal-dual methods* is to treat the Lagrange multipliers z, associated to the inequality constraints $c(x) \geq 0$, *independently* of the primal variables. These variables are then called *dual variables*.

There are different ways to motivate primal-dual methods for nonlinear programming, either from the approach used in linear programming, or in terms of the logarithmic barrier function (33). We will follow this last approach and make the link with the linear case thereafter.

Consider first problem (32), and remember the KKT conditions (18)–(21), which, specialized to this inequality-constrained problem, can be rewritten under the form: there exist $x^* \in \mathbb{R}^n$ and $z^* \in \mathbb{R}^p$ such that:

$$\text{KKT(NLP}_\mathcal{I}) \Leftrightarrow \begin{cases} \begin{bmatrix} \nabla f(x^*) - \nabla c(x^*)^\top z^* \\ C(x^*)z^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ c(x^*), z^* \geq 0, \end{cases} \tag{38}$$

where $C(x^*) = \text{diag}(c_1(x^*), \ldots, c_p(x^*))$. Consider now the barrier subproblem associated to (32):

$$\text{B}(\mu) \equiv \ \min_{x \in \mathbb{R}^n} B(x, \mu), \tag{39}$$

where $B(x, \mu)$ is the logarithmic barrier function (33), and its KKT conditions: there exists $x = x(\mu) \in \mathbb{R}^n$ such that:

$$\text{KKT(B}(\mu)) \Leftrightarrow \begin{cases} \nabla f(x) - \mu \nabla c(x)^\top C^{-1}(x)e = 0 \\ c(x) \geq 0. \end{cases} \tag{40}$$

Crucially, if we introduce the variables $z = \mu\, C^{-1}(x)e$ into (40) (which is equivalent to $C(x)z - \mu e = 0$), conditions KKT(B(μ)) are equivalent to the following ones: there exist $x = x(\mu) \in \mathbb{R}^n$ and $z = z(\mu) \in \mathbb{R}^p$ such that:

$$\begin{cases} \begin{bmatrix} \nabla f(x) - \nabla c(x)^\top z \\ C(x)z - \mu e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ c(x), z \geq 0, \end{cases} \tag{41}$$

where $z \geq 0$ comes from the fact that both $c(x)$ and μ are positive. Treating z as an independent variable when iteratively solving (41) yields a *primal-dual approach.*

Note that conditions (41) can also be interpreted as a perturbation of the KKT conditions for (32), see (38). That is, instead of the complementarity condition (21), the pairwise products $c_i(x)z_i$ are required to have the same positive values for all indices i. This condition forces each pairwise product $c_i(x)z_i$ to stay strictly positive and sufficiently away from zero for large μ, while converging to a solution of (38) when $\mu \searrow 0$. This is the way primal-dual interior-point methods are motivated in linear programming (see [45] for details).

The diagram below summarizes the motivation for primal-dual methods: find (x, z) satisfying the *primal-dual equations* (42), which may be seen as the KKT conditions for the barrier subproblem (39) when introducing independent dual variables $z = \mu\, C^{-1}(x)e$, or as a perturbed version of the KKT conditions for the original problem (32):

$$\text{KKT(NLP}_\mathcal{I}) \text{ perturbed}$$

$$\Leftrightarrow$$

$$\begin{cases} \begin{bmatrix} \nabla f(x) - \nabla c(x)^\top z \\ C(x)z \end{bmatrix} = \begin{bmatrix} 0 \\ \mu e \end{bmatrix} \\ c(x), z > 0 \end{cases} \tag{42}$$

$$\Leftrightarrow$$

$$\text{KKT(B}(\mu)), \text{ with } z = \mu\, C^{-1}(x)e \text{ treated independently.}$$

When solving the primal-dual equations (42), the set of solutions:

$$\mathcal{C} \stackrel{\text{def}}{=} \{(x(\mu), z(\mu)) \mid \mu > 0\}, \tag{43}$$

is said to define a (local) *primal-dual central path*. As $\mu \searrow 0$, $(x(\mu), z(\mu))$ converges to a solution (x^*, z^*) of (38) (under suitable assumptions, see [15] and [21]).

Introducing the notation:

$$F^\mu(x, z) = \left[\begin{array}{c} \nabla f(x) - \nabla c(x)^\top z \\ C(x)z - \mu e \end{array} \right],$$

a primal-dual method thus seeks to compute a feasible solution $(x(\mu), z(\mu))$ of the $n + p$ nonlinear equations $F^\mu(x, z) = 0$, such that $c(x(\mu)) > 0$ and $z(\mu) > 0$. To this aim, a *Newton's method* (again!) is applied. That is, if $v \stackrel{\text{def}}{=} (x, z)$, a primal-dual Newton direction, $\Delta v^N = (\Delta x^N, \Delta z^N)$, is computed as the solution of the *primal-dual Newton equations* $F^\mu(v)' \Delta v = -F^\mu(v)$, which we may write as:

$$\left(\begin{array}{cc} \nabla^2_{xx}\mathcal{L}(x, z) & -\nabla c(x)^\top \\ Z\nabla c(x) & C(x) \end{array} \right) \left(\begin{array}{c} \Delta x \\ \Delta z \end{array} \right) = -\left(\begin{array}{c} \nabla_x \mathcal{L}(x, z) \\ C(x)z - \mu e \end{array} \right), \tag{44}$$

where $Z = \text{diag}(z_1, \ldots, z_p)$. A strictly feasible step satisfying $c(x + \alpha\Delta x^N) > 0$ and $z + \alpha\Delta z^N > 0$ is then computed in this direction Δv^N.

Primal-dual methods based on the above frame do not suffer from the drawback of the primal approach, in that the primal-dual direction computed at a point on the central path, $v(\mu)$, with a new barrier parameter $\bar{\mu}$ ($\bar{\mu} < \mu$), is tangent to the barrier trajectory at $v(\mu)$. That is, the solution Δv^N to:

$$F^{\bar{\mu}}(v(\mu))' \Delta v = -F^{\bar{\mu}}(v(\mu))$$

will usually give a good approximation of the step to $v(\bar{\mu})$, the next point on the trajectory (see [21], Section 5).

Similarly to the primal Newton barrier approach, primal-dual methods are structured in inner and outer iterations. Inner iterations seek a point satisfying the primal-dual equations (42) for a given value of μ, while outer iterations test convergence for the original problem and update μ. Note that the cost of the solution of the primal-dual Newton equations (44) may be considerable. It is thus important to use efficient techniques for sparse linear systems (see [21]).

Under suitable assumptions, when close to a solution of problem (32), primal-dual methods converge rapidly to a solution of (32), provided μ is reduced at an appropriate rate (see [26] and [45]). When far from a solution, global convergence has to be forced using a line-search or a trust-region technique and making use of a merit function, as for the SQP approach. We refer the reader to [21], Section 5.2, for a discussion on this aspect for primal-dual methods. We simply point out that a *primal-dual merit function* has recently been proposed by Forsgren and Gill [20] for primal-dual interior-point methods. This merit function is an *augmented barrier function*, i.e., a classical barrier function augmented by a weighted proximity term that measures the distance of (x, z) to the trajectory $(x(\mu), z(\mu))$:

$$\phi_{FG}^{\beta,\mu}(x, z) = f(x) - \mu \sum_{i=1}^p \left(\log c_i(x) + \beta \left(\log \left(\frac{c_i(x)z_i}{\mu} \right) + 1 - \frac{c_i(x)z_i}{\mu} \right) \right), \tag{45}$$

where $\beta > 0$. As shown in [20], the function $\phi_{FG}^{\beta,\mu}(x, z)$ has several important properties, the most important one being that $\phi_{FG}^{\beta,\mu}(x, z)$ is minimized at a point $(x(\mu), z(\mu))$ on the barrier trajectory.

Several algorithms have also been proposed recently that use a filter to force convergence in primal-dual interior-point methods (see [2, 39, 42]), making use of appropriate choices to define the filter pair, such as the norm of the gradient of the Lagrangian and a proximity measure to the central path. The IPOPT software

package, developed by Wächter [41], is based on an interior-point method globalized by a line-search technique and uses a filter approach.

Extension of the above primal-dual approach to the general problem (17) can be easily done by considering the barrier subproblem associated to (17):

$$B(\mu) \equiv \begin{cases} \min_{x \in \mathbb{R}^n} & B(x, \mu) \\ \text{s.t.} & g(x) = 0, \end{cases} \tag{46}$$

where $B(x, \mu)$ is the logarithmic barrier function (33), and whose KKT conditions are given by: there exist $x = x(\mu) \in \mathbb{R}^n$ and $y = y(\mu) \in \mathbb{R}^m$ such that:

$$\text{KKT}(B(\mu)) \Leftrightarrow \begin{cases} \begin{bmatrix} \nabla f(x) - \nabla g(x)^\top y - \mu \nabla c(x)^\top C^{-1}(x)e \\ g(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ c(x) \geq 0. \end{cases} \tag{47}$$

If we introduce the variables $z = \mu \, C^{-1}(x)e$ in these conditions we obtain: there exist $x = x(\mu) \in \mathbb{R}^n$, $y = y(\mu) \in \mathbb{R}^m$ and $z = z(\mu) \in \mathbb{R}^p$ such that:

$$\begin{cases} \begin{bmatrix} \nabla f(x) - \nabla g(x)^\top y - \nabla c(x)^\top z \\ g(x) \\ C(x)z - \mu e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ c(x), z \geq 0, \end{cases} \tag{48}$$

which can again be interpreted as a perturbation of the KKT conditions for (17). The primal-dual central path is now defined as the set of solutions of the primal-dual equations (48):

$$\mathcal{C} \stackrel{\text{def}}{=} \{(x(\mu), y(\mu), z(\mu)) \mid \mu > 0\}, \tag{49}$$

and the primal-dual Newton equations are given by:

$$\begin{pmatrix} \nabla_{xx}^2 \mathcal{L}(x, y, z) & -\nabla g(x)^\top & -\nabla c(x)^\top \\ \nabla g(x) & 0 & 0 \\ Z \nabla c(x) & 0 & C(x) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = - \begin{pmatrix} \nabla_x \mathcal{L}(x, y, z) \\ g(x) \\ C(x)z - \mu e \end{pmatrix}. \tag{50}$$

Finally note that the merit function (45) can be generalized (see [20]) as the following *augmented penalty-barrier function*:

$$\phi_{FG}^{\beta,\mu}(x, y, z) = f(x) + \frac{1}{2\mu} \left(\|g(x)\|_2^2 + \beta \|g(x) + \mu y\|_2^2 \right)$$

$$- \mu \sum_{i=1}^{p} \left(\log c_i(x) + \beta \left(\log \left(\frac{c_i(x)z_i}{\mu} \right) + 1 - \frac{c_i(x)z_i}{\mu} \right) \right),$$

with $\beta > 0$, which is a classical penalty-barrier function[8] augmented by a weighted proximity term that measures the distance of (x, y, z) to the trajectory $(x(\mu), y(\mu), z(\mu))$.

---

[8] A penalty-barrier function is a merit function that combines the barrier function (33) with a penalty term involving the squares of the equality-constraint violation, to measure infeasibility with respect to the equality constraints:

$$f(x) - \mu \sum_{i=1}^{p} \log c_i(x) + \frac{1}{2\mu} \|g(x)\|_2^2. \tag{51}$$

2.2.4. *LOQO and KNITRO*

LOQO and KNITRO are two state-of-the-art interior-point software packages for large-scale nonlinear optimization. In this section, we summarize the main features of these two solvers in connection with the algorithmic features given in the previous sections.

LOQO is a primal-dual interior-point method developed by Shanno and Vanderbei [40], first to solve convex problems, then extended to handle nonconvexities. It starts by adding slack variables to the inequality constraints in (17), which becomes:

$$\begin{aligned}
\min_{x,s\in\mathbb{R}^n} \quad & f(x) \\
\text{s.t.} \quad & g(x) = 0 \\
& c(x) - s = 0 \\
& s \geq 0.
\end{aligned} \tag{52}$$

It then solves the primal-dual Newton equations associated to this new problem by a *modified line-search Newton method*. The system of primal-dual Newton equations is first symmetrized and reduced, then solved using a modified Cholesky factorization[9].

Once a search direction has been computed, LOQO proceeds to a new point by computing a step length that ensures positivity of the slack and dual variables, together with a sufficient decrease (using a backtracking line search) in a penalty-barrier merit function of the form (51) adapted to problem (52)[10]:

$$\phi_{LOQO}^{\beta,\mu}(x,s) = f(x) - \mu \sum_{i=1}^{p} \log s_i + \frac{\beta}{2} \left\| \left[\begin{array}{c} g(x) \\ c(x) - s \end{array} \right] \right\|_2^2,$$

where $\beta > 0$. Note that LOQO is an *infeasible* primal-dual interior-point method, in the sense that it does not require feasibility of the iterates with respect to the inequality constraints in (17), but only forces the slack variables in (52) to remain positive. For a complete description of LOQO, see [40].

KNITRO has been developed by Byrd, Waltz, Nocedal and their collaborators (see [8] and [43]) and is also an infeasible primal-dual interior-point method. KNITRO is supported by a convergence theory that can be found in [7] and [9]. Unlike LOQO, KNITRO does not factor the primal-dual matrix in the system of primal-dual Newton equations, but rather uses a *barrier–SQP approach*. More precisely, KNITRO, as in LOQO, employs slack variables to formulate barrier subproblems of the form:

$$\begin{aligned}
\min_{x,s\in\mathbb{R}^n} \quad & f(x) - \mu \sum_{i=1}^{p} \log s_i \\
\text{s.t.} \quad & g(x) = 0 \\
& c(x) - s = 0,
\end{aligned} \tag{53}$$

and applies, in an inner iteration, an SQP-type method for equality-constrained optimization (see Section 2.1.1) to compute steps that approximately solve the barrier subproblem (53). Moreover, a *trust-region technique* is used to globalize the process and handle nonconvexities.

At a given iteration k and for given x_k, s_k, y_k and z_k, the quadratic subproblem in KNITRO is formulated as:

$$\begin{aligned}
\min_{p_x,p_s} \quad & \nabla f(x_k)^T p_x + \tfrac{1}{2} p_x^T \nabla_{xx}^2 \mathcal{L}(x_k,s_k,y_k,z_k)p_x - \mu e^T S_k^{-1} p_s + \tfrac{1}{2} p_s^T W_k p_s \\
\text{s.t.} \quad & \nabla g(x_k)p_x + g(x_k) = r_g \\
& \nabla c(x_k)p_x - p_s + c(x_k) - s_k = r_c \\
& (p_x, p_s) \in T_k,
\end{aligned} \tag{54}$$

---

[9]A modified Cholesky factorization is a Cholesky factorization applied to a non necessarily positive definite matrix in such a way that the diagonal elements encountered during the factorization are increased (where necessary) to ensure that they are sufficiently positive. This technique guarantees the existence of the modified Cholesky factors but avoids to modify the matrix to make it positive definite if this one is already sufficiently positive definite.

[10]A filter approach to LOQO has also been developed in [2].

where:

- $\mathcal{L}(x, s, y, z)$ is the Lagrangian function associated to (53):

$$\mathcal{L}(x, s, y, z) = f(x) - \mu \sum_{i=1}^{p} \log s_i - y^T g(x) - z^T (c(x) - s).$$

- S_k is the diagonal matrix with the components of s_k as diagonal elements.
- W_k is a p by p positive definite diagonal matrix chosen as:

$$W_k = S_k^{-1} Z_k, \tag{55}$$

where Z_k is the diagonal matrix with the components of z_k as diagonal elements. This choice for W_k is crucial for the iteration to have *primal-dual* characteristics. Indeed, with W_k defined by (55), it can be shown, under appropriate assumptions, that the subproblem (54) (where (r_g, r_c) is set to zero and the trust-region constraint $(p_x, p_s) \in T_k$ is removed) has a unique solution which satisfies the *primal-dual Newton equations* associated to (53). Note that this choice of W_k may be viewed as an approximation to $\nabla_{ss} \mathcal{L}(x_k, s_k, y_k, z_k) = \mu S_k^{-2}$, since the KKT conditions for problem (53) are:

$$\begin{aligned}
\nabla_x \mathcal{L}(x, s, y, z) &= \nabla f(x) - \nabla g(x)^\top y - \nabla c(x)^\top z = 0, & (56) \\
\nabla_s \mathcal{L}(x, s, y, z) &= -\mu S^{-1} e + z = 0, & (57) \\
g(x) &= 0, & (58) \\
c(x) - s &= 0, & (59)
\end{aligned}$$

and thus equation $\mu S^{-1} = Z$ is satisfied at a KKT point, by (57). Substituting this equation in $\nabla_{ss} \mathcal{L}(x_k, s_k, y_k, z_k) = \mu S_k^{-2}$ gives W_k.

- T_k defines a trust region where the quadratic model and the linearized constraints in (54) can be trusted to be good approximations to the problem, and where feasibility of the slack variables ($s \geq 0$) is also ensured.
- The residuals r_g and r_c are computed as small as possible such that the constraints of the subproblem (54) are consistent. Indeed, we would like ideally that a step (p_x, p_s) from (x_k, s_k) satisfies the linearized constraints:

$$\nabla g(x_k) p_x + g(x_k) = 0 \text{ and } \nabla c(x_k) p_x - p_s + c(x_k) - s_k = 0, \tag{60}$$

but this may be inconsistent with the trust-region constraint $(p_x, p_s) \in T_k$. The computation of r_g and r_c is done by solving a preliminary subproblem in which a so-called *normal step* is computed (that is, a step that lies within the trust region T_k and approximately satisfies (60) in the least-squares sense).

To approximately solve problem (54), KNITRO uses the decomposition proposed by Byrd and Omojokun [38], that is, it first computes a *normal step* that attempts to satisfy the linear constraints (60) as well as possible, determining r_g and r_c, and then a *tangential step* that remains in the tangent space of the constraints and tries to achieve optimality of (54) (see [8] for details). Finally, KNITRO determines if the candidate step is acceptable and updates the trust-region radius (see Section 1.3), according to the reduction obtained in the following (nondifferentiable) merit function:

$$\phi_{KNITRO}^{\beta, \mu}(x, s) = f(x) - \mu \sum_{i=1}^{p} \log s_i + \beta \left\| \begin{bmatrix} g(x) \\ c(x) - s \end{bmatrix} \right\|_2.$$

We refer the reader to [7] and [8] for an in-depth description of KNITRO.

This concise description of LOQO and KNITRO shows that even though these two software belong to the class of interior-point methods, they are conceptually quite different. Nevertheless it is at present impossible to

say the advantage of one over the other. The reality today is that, if a problem is solved with both software, nobody can predict if one will perform better than the other. As an example, both software use a different approach to handle the barrier parameter, but so little is understood about this issue for nonlinear optimization problems that it is hard to know which approach is more appropriate. However, one argument which can be invoked in favour of KNITRO is that it is supported by a convergence theory, unlike LOQO.

2.3. SQP or interior-point methods?

Still a lot of questions related to the behaviour, the efficiency and the robustness of interior-point and active-set methods require investigation (see [25]). It is also far from clear which among interior-point and SQP appoaches is the best to solve constrained optimization problems. No-one can answer this question, even though it seems at present that interior-point methods surpass active-set methods for large-scale problems. The fact is that interior-point and active-set methods are still competing and changing (and will probably coexist for the next few years).

Advantages of active-set SQP methods is that they are parameter-free, relatively scale invariant and can make use of a good starting point. On the other hand, the identification of the optimal set of active constraints at a solution is not a simple task for degenerate problems, and current algorithms to approximately solve the SQP subproblems (see (28)) are not fully efficient yet.

Interior-point methods can be very fast and powerful and are not too sensible to degeneracy. The disadvantages are that they are dependent on certain parameters, such as the barrier parameter, whose updates are to a large extent heuristic. Nonlinear interior-point methods are also quite scale dependent and their efficiency closely depends on the starting point.

We conclude by pointing out two references, [3] and [34], where the results of a comparative numerical study of several SQP and interior-point software (mainly SNOPT, filterSQP, LOQO and KNITRO) are reported.

3. Useful tools and references

To conclude this overview, we list several useful tools for the reader interested in various kinds of information related to nonlinear optimization methods, algorithms, software and testing environment:

- The **NEOS optimization guide** and the **NEOS optimization server** are quite useful tools that have been developed by the Optimization Technology Center (Argonne National Laboratory and Northwestern University). The NEOS guide gives access to a comprehensive guide to optimization problems, algorithms and applications. The NEOS server allows automatic solution of optimization problems, with minimal input from the user, remotely over the Internet, and proposes a large choice of state-of-the-art optimization solvers. These tools are very easy to use and accessible at the following address:

  http://www-neos.mcs.anl.gov/

- **CUTEr** is a versatile testing environment for optimization (and linear algebra) solvers. The package contains a collection of test problems, along with Fortran 77, Fortran 90/95 and Matlab tools intended to help developers design, compare and improve new and existing solvers. Ready-to-use interfaces to existing packages, such as SNOPT, LOQO, KNITRO, filterSQP, IPOPT (and more), are provided. See:

  http://hsl.rl.ac.uk/cuter-www/

- Hans Mittelmann proposes a site of **Benchmarks for Optimization Software** where benchmarks of a large choice of state-of-the-art solvers are presented for a range of topics related to optimization (nonlinear programming, but also linear programming, mixed integer linear programming, quadratic programming, mixed integer nonlinear programming, semidefinite programming, etc). See:

  http://plato.asu.edu/bench.html

- **Optimization Online** is a repository of eprints about optimization and related topics. Authors can announce their new report (no claim about quality or correctness is done) by posting it on Optimization Online, and visitors have an easy electronic access to these reports. One may also subscribe to the Optimization Online monthly digest to receive an email message at the end of each month, with titles of and links to the reports submitted during that month. Optimization Online also proposes a list of interesting links (Optimization Online Links). See:

  http://www.optimization-online.org/

We would also like to emphasize four references, [12], [18], [21] and [37], i.e., two books and two articles, which in our opinion may be quite useful to the reader interested in the various topics outlined in this paper.

Acknowledgments

References

[1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, J. Wiley and Sons, New-York, 2nd ed., 1993.

[2] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: filter methods and merit functions*, Tech. Rep. ORFE–00-06, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2000.

[3] ——, *A comparative study of large-scale nonlinear optimization algorithms*, Tech. Rep. ORFE–01-04, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2001.

[4] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1982.

[5] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Optimisation Numérique : aspects théoriques et pratiques*, Springer Verlag, Berlin, 1997.

[6] ——, *Numerical Optimization. Theoretical and Practical Aspects*, Springer Verlag, Berlin, 2002.

[7] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust-region method based on interior-point techniques for nonlinear programming*, Mathematical Programming, Series A, 89 (2000), pp. 149–185.

[8] R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior-point algorithm for large-scale nonlinear programming*, SIAM Journal on Optimization, 9 (1999), pp. 877–900.

[9] R. H. Byrd, G. Liu, and J. Nocedal, *On the local behavior of an interior point method for nonlinear programming*, in Numerical Analysis: Proceedings Dundee 1997, D. F. Griffiths and D. J. Higham, eds., Addison Wesley Longman, 1997, pp. 37–56.

[10] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Mathematics of Computation, 50 (1988), pp. 399–430.

[11] ——, *A note on using alternative second-order models for the subproblems arising in barrier function methods for minimization*, Numerische Mathematik, 68 (1994), pp. 17–33.

[12] ——, *Trust-Region Methods*, no. 01 in MPS-SIAM Series on Optimization, SIAM, Philadelphia, USA, 2000.

[13] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente, *Trust-region interior-point SQP algorithms for a class of nonlinear programming problems*, SIAM Journal on Control and Optimization, 36 (1998), pp. 1750–1794.

[14] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1983. Reprinted as *Classics in Applied Mathematics 16*, SIAM, Philadelphia, USA, 1996.

[15] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, J. Wiley and Sons, Chichester, England, 1968. Reprinted as *Classics in Applied Mathematics 4*, SIAM, 1990.

[16] R. Fletcher, *Practical Methods of Optimization*, J. Wiley and Sons, Chichester, England, second ed., 1987.

[17] R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint, and A. Wächter, *Global convergence of trust-region SQP-filter algorithms for nonlinear programming*, SIAM Journal on Optimization, 13 (2002), pp. 635–659.

[18] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Mathematical Programming, 91 (2002), pp. 239–269.

[19] C. A. Floudas and P. M. Pardalos, eds., *Recent Advances in Global Optimization*, Princeton University Press, Princeton, USA, 1992.

[20] A. Forsgren and P. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM Journal on Optimization, 8 (1998), pp. 1132–1152.

[21] A. Forsgren, P. E. Gill, and M. H. Wright, *Interior methods for nonlinear optimization*, SIAM Review, 44 (2002), pp. 525–597.

[22] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Journal on Optimization, 12 (2002), pp. 979–1006.

[23] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[24] S. M. Goldfeldt, R. E. Quandt, and H. F. Trotter, *Maximization by quadratic hill-climbing*, Econometrica, 34 (1966), pp. 541–551.

[25] N. I. M. Gould, *Some reflections on the current state of active-set and interior-point methods for constrained optimization*, SIAG/OPT Views-and-News, 14 (2003), pp. 2–7.

[26] N. I. M. Gould, D. Orban, A. Sartenaer, and P. L. Toint, *Superlinear convergence of primal-dual interior-point algorithms for nonlinear programming*, SIAM Journal on Optimization, 11 (2001), pp. 974–1002.

[27] N. I. M. Gould and P. L. Toint, *How mature is nonlinear optimization?*, in Proceedings of ICIAM 2003, Sydney, Australia, 2003 (to appear).

[28] J.-B. Hiriart-Urruty, *L'optimisation*, Que sais-je? 3184. Presses Universitaires de France, 1996.

[29] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms. Part 1: Fundamentals*, Springer Verlag, Heidelberg, Berlin, New York, 1993.

[30] ———, *Convex Analysis and Minimization Algorithms. Part 2: Advanced Theory and Bundle Methods*, Springer Verlag, Heidelberg, Berlin, New York, 1993.

[31] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[32] K. Levenberg, *A method for the solution of certain problems in least squares*, Quarterly Journal on Applied Mathematics, 2 (1944), pp. 164–168.

[33] D. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, SIAM Journal on Applied Mathematics, 11 (1963), pp. 431–441.

[34] J. L. Morales, J. Nocedal, R. A. Waltz, G. Liu, and J. P. Goux, *Assessing the potential of interior methods for nonlinear optimization*, in First Sandia Workshop on Large-Scale PDE-Constrained Optimization (Santa Fe, New Mexico, USA, April 2001), O. Ghattas, ed., Springer's Lecture Notes in Computational Science and Engineering, 2003 (to appear).

[35] J. J. Moré and D. C. Sorensen, *Computing a trust-region step*, SIAM Journal on Scientific and Statistical Computing, 4 (1983), pp. 553–572.

[36] D. D. Morrison, *Methods for nonlinear least squares problems and convergence proofs*, in Proceedings of the Seminar on Tracking Programs and Orbit Determination, J. Lorell and F. Yagi, eds., Pasadena, USA, 1960, Jet Propulsion Laboratory, pp. 1–9.

[37] J. Nocedal and S. J. Wright, *Numerical Optimization*, Series in Operations Research, Springer Verlag, Heidelberg, Berlin, New York, 1999.

[38] E. O. Omojokun, *Trust-region algorithms for optimization with nonlinear equality and inequality constraints*, PhD thesis, University of Colorado, Boulder, Colorado, USA, 1989.

[39] M. Ulbrich, S. Ulbrich, and L. Vicente, *A globally convergent primal-dual interior-point filter method for nonconvex nonlinear programming*, Tech. Rep. TR00-11, Department of Mathematics, University of Coimbra, Coimbra, Portugal, 2000.

[40] R. J. Vanderbei and D. F. Shanno, *An interior point algorithm for nonconvex nonlinear programming*, Computational Optimization and Applications, 13 (1999), pp. 231–252.

[41] A. Wächter, *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*, PhD thesis, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, USA, 2002.

[42] A. Wächter and L. T. Biegler, *Global and local convergence of line-search filter methods for nonlinear programming*, Tech. Rep. CAPD B-01-09, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, USA, 2001.

[43] R. A. Waltz and J. Nocedal, *KNITRO user's manual*, Tech. Rep. OTC 2003/05, Optimization Technology Center, Northwestern University, Evanston, IL, USA, April 2003.

[44] M. H. Wright, *Why a pure primal Newton barrier step may be infeasible*, SIAM Journal on Optimization, 5 (1995), pp. 1–12.

[45] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, USA, 1997.