Sequential Quadratic Programming Methods

Klaus Schittkowski*
Ya-xiang Yuan†

June 30, 2010

Abstract

We present a brief review on one of the most powerful methods for solving smooth constrained nonlinear optimization problems, the so-called sequential quadratic programming (SQP) method. Starting during the late 70's, global and local convergence theorems were proved and efficient codes released. Today, SQP algorithms are the standard tool in academia and industry to solve highly complex application problems.

Keywords: nonlinear programming, nonlinear optimization, sequential quadratic programming, quasi-Newton, line search, trust region, filter, superlinear convergence, global convergence

1 Introduction

We consider the nonlinear, constrained optimization problem to minimize an objective function f under m nonlinear inequality constraints, i.e.,

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &x \in I\!R^n : \quad g(x) \geq 0 \ , \end{aligned} \tag{1}$$

where x is an n-dimensional parameter vector and where $g(x) := (g_1(x), \ldots, g_m(x))^T$, i.e., we have m nonlinear inequality constraints. (1) is now called a nonlinear program (NLP) in abbreviated form.

Equality constraints and simple bounds for the variables are omitted to simplify the subsequent notation. They are easily introduced again whenever needed, where equality constraints are linearized in the quadratic programming subproblem. It is assumed that objective function $f(x)$ and m constraint functions $g_1(x)$, ..., $g_m(x)$ are continuously differentiable on the whole $I\!R^n$.

*Dept. of Computer Science, University of Bayreuth, Germany
†AMSS, Chinese Academy of Sciences, Beijing, China

Optimization theory for smooth problems is based on the Lagrangian function that combines the objective function $f(x)$ and the constraints $g(x)$ in a proper way. In particular, the Lagrangian function allows us to state necessary and sufficient optimality conditions.

Let problem (1) be given. First we define the *feasible region* P as the set of all feasible solutions

$$P := \{x \in I\!R^n : g(x) \geq 0\} \quad . \tag{2}$$

and the set of *active constraints* with respect to $x \in P$ by

$$I(x) := \{j : g_j(x) = 0,\ 1 \leq j \leq m\} \quad . \tag{3}$$

The *Lagrangian function* of (1) is defined by

$$L(x, u) := f(x) - u^T g(x) \tag{4}$$

for all $x \in I\!R^n$ and $u = (u_1, \ldots, u_m)^T \in I\!R^m$, $u \geq 0$. The variables u_j are called the *Lagrangian multipliers* of the nonlinear programming problem. x is also called the primal and u the dual variable of the nonlinear program (1).

A first idea has been investigated in the Ph.D. thesis of Wilson [54]. Sequential quadratic programming methods became popular during the late 70's due to papers of Han [30, 31] and Powell [38, 39]. Their superiority over other optimization methods known at that time, was shown by Schittkowski [44]. Since then many modifications and extensions have been published on SQP methods. Nice review papers are given by Boggs and Tolle [2] and Gould and Toint [27]. All modern optimization textbooks have chapters on SQP methods, for example see Fletcher [19], Gill et al. [25] and Sun and Yuan [46]. As the presentation of even a selected overview is impossible due to the limited space here, we concentrate on a few important facts and highlights from a personal view without any attempt to be complete.

The basic ideas of sequential quadratic programming methods are found in Section 2. The role of quasi-Newton updates is outlined in Section 3. Several stabilization procedures are available by which convergence is guaranteed, e.g., line search, see Section 4, trust regions, see Section 5, or filter techniques, see Section 6. Some typical convergence results for global convergence, i.e., convergence towards a stationary point starting from an arbitrary point, and local convergence, fast final convergence speed close to a solution, are listed in Section 7.

2 Sequential Quadratic Programming Methods

Sequential quadratic programming or SQP methods belong to the most powerful optimization algorithms we know today for solving differentiable nonlinear programs of the form (1). The theoretical background is described for example in Stoer [47], and an excellent review is given by Boggs and Tolle [2]. From a more practical point of view, SQP methods are also introduced in the books of Papalambros and Wilde [37] and Edgar and Himmelblau [16], among many others. Their excellent numerical performance has been tested and compared with other methods, see Schittkowski [44], and for many years they

belong to the set of most frequently used algorithms for solving practical optimization problems.

The basic idea is to formulate and solve a quadratic programming subproblem in each iteration which is obtained by linearizing the constraints and approximating the Lagrangian function $L(x,u)$ (4) quadratically. Starting from any $x_0 \in I\!R^n$, suppose that $x_k \in I\!R^n$ is an actual approximation of the solution, $v_k \in I\!R^m$ an approximation of the multipliers, and $B_k \in I\!R^{n\times n}$ an approximation of the Hessian of the Lagrangian function, $k = 0, 1, 2, \ldots$. Then, a quadratic program (QP) of the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\, d^T B_k d + \nabla f(x_k)^T d \\ d \in I\!R^n : & \nabla g(x_k)^T d + g(x_k) \geq 0 \end{array} \tag{5}$$

is formulated and solved in each iteration. To simplify the subsequent presentation, we assume that (5) is always solvable. Let d_k be the optimal solution, u_k the corresponding multiplier of this subproblem. A new iterate is then obtained by

$$x_{k+1} := x_k + d_k \tag{6}$$

and, for the moment, $v_{k+1} := u_k$.

To motivate the above approach, let us assume that there are only equality constraints of the form $g(x) = 0$ instead of $g(x) \geq 0$. The Karush-Kuhn-Tucker optimality conditions are then written in the form

$$F(x,u) := \begin{pmatrix} \nabla_x L(x,u) \\ g(x) \end{pmatrix} = 0 \ .$$

In other words, the optimal solution and the corresponding multipliers are the solution of a system of $n+m$ nonlinear equations $F(x,u) = 0$ with $n+m$ unknowns $x \in I\!R^n$ and $u \in I\!R^m$.

Let (x_k, v_k) be an approximation of the solution. We apply Newton's method and get an estimate for the next iterate by

$$\nabla F(x_k, v_k) \begin{pmatrix} d_k \\ y_k \end{pmatrix} + F(x_k, v_k) = 0 \ .$$

After insertion, we obtain the equation

$$\begin{pmatrix} B_k & : & -\nabla g(x_k) \\ \nabla g(x_k)^T & : & 0 \end{pmatrix} \begin{pmatrix} d_k \\ y_k \end{pmatrix} + \begin{pmatrix} \nabla f(x_k) - \nabla g(x_k) v_k \\ g(x_k) \end{pmatrix} = 0 \tag{7}$$

with $B_k := \nabla_{xx} L(x_k, v_k)$. Defining now $u_k := y_k + v_k$, we get

$$B_k d_k - \nabla g(x_k) u_k + \nabla f(x_k) = 0$$

and

$$\nabla g(x_k)^T d_k + g(x_k) = 0 \ .$$

But these equations are exactly the optimality conditions for the quadratic program formulated for equality constraints only. To sum up, we get the following conclusion.

A sequential quadratic programming method is identical to Newton's method for solving the KKT optimality conditions, if B_k is the Hessian of the Lagrangian function and if we start sufficiently close to a solution.

Now we assume again that we have inequality constraints instead of the equality ones. A straightforward analysis based on the optimality conditions shows that if $d_k = 0$ is an optimal solution of (5) and u_k the corresponding multiplier vector, then x_k and u_k satisfy the necessary optimality conditions of (1).

Note that the above analysis also serves to point out the crucial role of the dual variable u_k. To get a fast convergence rate as expected for Newton's method, we have to adapt the multiplier values accordingly together with the primal ones.

To avoid the computation of second derivatives, B_k is set to a positive-definite approximation of the Hessian of the Lagrangian function. For the global convergence analysis, any choice of B_k is appropriate as long as the eigenvalues are bounded away from zero. However, to guarantee a superlinear convergence rate, we update B_k by a quasi-Newton method, e.g., the BFGS formula

$$B_{k+1} := B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} \tag{8}$$

with

$$\begin{aligned} y_k &:= \nabla_x L(x_{k+1}, u_k) - \nabla_x L(x_k, u_k) \ , \\ s_k &:= x_{k+1} - x_k \ . \end{aligned} \tag{9}$$

Usually, we start with the unit matrix for B_0 and stabilize the formula by requiring that $a_k^T b_k \geq 0.2\ b_k^T B_k b_k$. Positive definite matrices B_k guarantee unique solutions and in addition a sufficient descent property of a merit function needed for the global convergence analysis.

Moreover, it is possible to proceed from a Cholesky factorization of B_k in the form

$$B_k = L_k L_k^T$$

with a lower triangular matrix L_k and to update L_k directly to get a factorization of B_{k+1}, see Gill et al. [23],

$$B_{k+1} = L_{k+1} L_{k+1}^T = L_k L_k^T + a_k a_k^T - b_k b_k^T \ .$$

A particular advantage is that an initial Cholesky decomposition is avoided, if a primal-dual method is applied to solve the quadratic program (5).

Note that the above motivation proceeds from a local approximation of a solution. If starting from an arbitrary $x_0 \in I\!R^n$, $v_0 \in I\!R^m$, one has to stabilize the algorithm by introducing additional safeguards, e.g., a line search, a trust region, or a filter. These three possibilities are discussed in the subsequent sections.

3 Reduced Hessian SQP Steps

The linear system (7) derived from the quadratic program subproblem can be rewritten as

$$\begin{pmatrix} Y_k^T B_k Y_k & Y_k^T B_k Z_k & -R_k \\ Z_k^T B_k Y_k & Z_k^T B_k Z_k & 0 \\ -R_k^T & 0 & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \\ y_k \end{pmatrix} = \begin{pmatrix} -Y_k^T \nabla f(x_k) \\ -Z_k^T \nabla f(x_k) \\ g(x_k) \end{pmatrix} , \tag{10}$$

if we have the QR factorization

$$\nabla g(x_k) = (Y_k, Z_k) \begin{pmatrix} R_k \\ 0 \end{pmatrix},$$

and if we use the notation $p_k = Y_k^T d_k$ and $q_k = Z_k^T d_k$. Thus, it is easy to see that p_k and q_k can be obtained by

$$\begin{pmatrix} Z_k^T B_k Y_k & Z_k^T B_k Z_k \\ -R_k^T & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = \begin{pmatrix} -Z_k^T \nabla f(x_k) \\ g(x_k) \end{pmatrix}, \tag{11}$$

which indicates that we only need the reduced matrix $Z_k^T B_k$ instead of the full matrix B_k to compute the QP step d_k. Hence, Nocedal and Overton [35] suggest to replace $Z_k^T B_k$ by a quasi-Newton matrix $\bar{B}_k$, and to obtain the QP step d_k by solving

$$\begin{pmatrix} \bar{B}_k \\ -\nabla g(x_k)^T \end{pmatrix} d = \begin{pmatrix} -Z_k^T \nabla f(x_k) \\ g(x_k) \end{pmatrix}. \tag{12}$$

The matrix $\bar{B}_k \in I\!R^{(n-m)\times n}$ is a quasi-Newton matrix that approximates $Z_k^T \nabla_{xx}^2 L(x_k, u_k)$, which can be updated by the Broyden's unsymmetric Rank-1 formula

$$\bar{B}_{k+1} := \bar{B}_k + \frac{(\bar{y}_k - \bar{B}_k s_k) s_k^T}{||s_k||_2^2},$$

where $s_k := x_{k+1} - x_k$ and $\bar{y}_k := Z_{k+1}^T \nabla f(x_{k+1}) - Z_k^T \nabla f(x_k)$. This approach can be more efficient than the standard SQP because it reduces storage and calculation due to the fact that it uses a reduced matrix $B_k \in I\!R^{(n-m)\times n}$ instead of the full matrix B_k, particularly when m is close to n, namely for problems there are lots of variables and lots of constraints but less freedom in the variables. Moreover, this approach preserve the q-superlinearly convergent property of the standard SQP step, for example see Nocedal and Overton [35].

We move along this direction further. If we replace $Z_k^T B_k Z_k$ by a quasi-Newton matrix $\hat{B}_k$ and replace $Z_k^T B_k Y_k$ by zero in the linear system (11), we obtain

$$\begin{pmatrix} 0 & \hat{B}_k \\ -R_k^T & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = \begin{pmatrix} -Z_k^T \nabla f(x_k) \\ g(x_k) \end{pmatrix}. \tag{13}$$

The reason for doing so is that Powell [38] discovered that the SQP method converges 2-step Q-superlinearly,

$$\lim_{k\to\infty} \frac{||x_{k+1} - x^*||}{||x_{k-1} - x^*||} = 0 \tag{14}$$

provided that $Y_k^T B_k Z_k$ is bounded. An advantage of this approach which is called two-side reduced Hessian method, compared to Nocedal and Overton's one-side reduced Hessian method, is that the matrix $\hat{B}_k$ is a square matrix and can be kept positive definite by updates such as the BFGS formula. Furthermore, the computation cost is also reduced. But, the price we have to pay is that we can only prove 2-step Q-superlinear convergence (14) instead of the one-step Q-superlinear convergence. Indeed, examples exist to show that the 2-step Q-superlinear convergence of the two-side reduced Hessian method can not be improved to one-step Q-superlinear convergence, for example, see Byrd [4] and Yuan [56].

4 Stabilization by Line Search

We have seen in the previous sections that one should better approximate the actual primal and dual variables simultaneously to satisfy the optimality conditions of (1) based on the Newton method. Thus, we define $z_k := (x_k, v_k)$, where $x_k \in I\!R^n$ and $v_k \in I\!R^m$ are approximations of a KKT point in the k-th step. A new iterate is then computed from

$$z_{k+1} := z_k + \alpha_k p_k \quad , \tag{15}$$

where $p_k := (d_k, y_k) = (d_k, u_k - v_k)$ is the search direction obtained from solving the quadratic program (5), see also (7), and where α_k is a scalar parameter called the steplength. The goal is to compute a sufficiently accurate stepsize within as few function evaluations as possible, so that the underlying algorithm converges. α_k should satisfy at least a sufficient decrease condition of a merit function

$$\phi_r(\alpha) := \psi_r \left(\begin{pmatrix} x \\ v \end{pmatrix} + \alpha \begin{pmatrix} d \\ u - v \end{pmatrix} \right) \tag{16}$$

with a suitable penalty function $\psi_r(x, v)$ and with $r = (r_1, \ldots, r_m)^T \in I\!R^m$, a vector of m positive penalty parameters.

A merit function is supposed to contain information about the objective function to be minimized, and the violation of constraints. The first one studied in the context of SQP methods was the L_1 penalty function

$$\psi_r(x, v) := f(x) + \sum_{j=1}^{m} r_j g_j(x)^- \quad , \tag{17}$$

where $g_j(x)^- := -\min(0, g_j(x))$ denotes violated constraints, see Han [31]. $\psi_r(x, v)$ is a so-called exact penalty function, i.e., a minimizer of $\psi_r(x, v)$ subject to x for sufficiently large r is also a solution of NLP. But there are two severe drawbacks: $\psi_r(x, v)$ is non-differentiable preventing efficient algorithms and, since $\psi_r(x, v)$ does not depend on multiplier information, it enables the Maratos-effect [32], i.e., the possibility of very slow convergence close to a solution.

Instead of (17), augmented Lagrangian merit functions

$$\psi_r(x, v) := f(x) - v^T g(x) + \frac{1}{2} \sum_{j=1}^{m} r_j g_j(x)^2 \tag{18}$$

are an appropriate alternative which differ by the choice of the multiplier approximation, see Boggs and Tolle [2] or Gill et al. [24]. A slightly different augmented Lagrangian function is given by

$$\psi_r(x, v) := f(x) - \sum_{j \in J} (v_j g_j(x) - \frac{1}{2} r_j g_j(x)^2) - \frac{1}{2} \sum_{j \in K} v_j^2 / r_j \tag{19}$$

with $J := \{j : 1 < j \leq m, g_j(x) \leq v_j / r_j\}$ and $K := \{1, \ldots, m\} \; - \; J$, cf. Schittkowski [45] and Rockafellar [43], but very many other merit functions have been investigated in the past.

In all cases, the objective function is *penalized* as soon as an iterate leaves the feasible domain. The corresponding penalty parameters r_j, $j = 1, \ldots, m$, which control the degree of constraint violation, must be carefully chosen to guarantee a descent direction of the merit function, see Schittkowski [45] or Wolfe [55] in a more general setting, i.e., to achieve at least

$$\phi'_{r_k}(0) = \bigtriangledown \psi_{r_k}(x_k, v_k)^T \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix} < 0 \tag{20}$$

when applied in the k-iteration step.

The implementation of a line search algorithm is a crucial issue when implementing a nonlinear programming algorithm, and has significant effect on the overall efficiency of the resulting code. On the one hand, we need a line search to stabilize the algorithm. On the other hand, it is not desirable to waste too many function calls. Moreover, the behavior of the merit function becomes irregular in case of constrained optimization because of very steep slopes at the border caused by large penalty terms. The implementation is more complex than shown above, if linear constraints and bounds of the variables are to be satisfied during the line search.

Usually, the steplength parameter α_k is chosen to satisfy a certain Armijo [1] condition, i.e., a sufficient descent condition of a merit function, which guarantees convergence to a stationary point. However, to take the curvature of the merit function into account, we need some kind of compromise between a polynomial interpolation, typically a quadratic one, and a reduction of the stepsize by a given factor, until a stopping criterion is reached. Since $\phi_r(0)$, $\phi'_r(0)$, and $\phi_r(\alpha_i)$ are given, α_i the actual iterate of the line search procedure, we easily get the minimizer of the quadratic interpolation. We accept then the maximum of this value and the Armijo parameter as a new iterate, as shown by the subsequent code fragment.

Algorithm 4.1. *Let β, μ with $0 < \beta < 1$, $0 < \mu < 0.5$ be given.*

Start: $\alpha_0 = 1$

For $i = 0, 1, 2, \ldots$ *do:*

1) *If* $\phi_r(\alpha_i) < \phi_r(0) + \mu\ \alpha_i\ \phi'_r(0)$, *then stop.*

2) *Compute* $\bar{\alpha}_i := \dfrac{0.5\ \alpha_i^2\ \phi'_r(0)}{\alpha_i \phi'_r(0) - \phi_r(\alpha_i) + \phi_r(0)}$.

3) *Let* $\alpha_{i+1} := \max(\beta\ \alpha_i, \bar{\alpha}_i)$.

The algorithm goes back to Powell [38] and corresponding convergence results are found in Schittkowski [45]. $\bar{\alpha}_i$ is the minimizer of the quadratic interpolation, and we use the Armijo descent property for checking termination. Step 3 is required to avoid irregular values, since the minimizer of the quadratic interpolation could be outside of the feasible domain $(0, 1]$. Additional safeguards are required, for example to prevent violation of bounds. Algorithm 4.1 assumes that $\phi_r(1)$ is known before calling the procedure, i.e., that the corresponding function values are given. We stop the algorithm, if sufficient descent is not observed after a certain number of iterations. If the tested stepsize falls below machine precision or the accuracy by which model function values are computed, the merit function cannot decrease further.

It is possible that $\phi_r'(0)$ becomes negative due to round-off errors in the partial derivatives or that Algorithm 4.1 breaks down because to too many iterations. In this case, we proceed from a descent direction of the merit function, but $\phi_r'(0)$ is extremely small. To avoid interruption of the whole iteration process, the idea is to repeat the line search with another stopping criterion. Instead of testing (20), we accept a stepsize α_k as soon as the inequality

$$\phi_{r_k}(\alpha_k) \leq \max_{k-p(k)<=j<=k} \phi_{r_j}(0) + \alpha_k \mu \phi'_{r_k}(0) \tag{21}$$

is satisfied, where $p(k)$ is a non-decreasing integer with $p(k) := \min\{k, p\}$, p a given tolerance. Thus, we allow an increase of the reference value $\phi_{r_{j_k}}(0)$ in a certain error situation, i.e., an increase of the merit function value.

To implement the non-monotone line search, we need a queue consisting of merit function values at previous iterates. In case of $k = 0$, the reference value is adapted by a factor greater than 1, i.e., $\phi_{r_{j_k}}(0)$ is replaced by $t\phi_{r_{j_k}}(0)$, $t > 1$. The basic idea to store reference function values and to replace the sufficient descent property by a sufficient 'ascent' property in max-form, is described in Dai [11] and Dai and Schittkowski [12], where convergence proofs are presented. The general idea goes back to Grippo, Lampariello, and Lucidi [29], and was extended to constrained optimization and trust region methods in a series of subsequent papers, see, e.g., Toint [49, 50].

5 Stabilization by Trust Regions

Another globalization technique for nonlinear optimization is trust region, where the next iterate requires to be in the neighborhood of the current point, namely $||x_{k+1} - x_k|| \leq \Delta_k$, where $\Delta_k > 0$ is the trust region radius updated from iteration to iteration. Thus, an essential constraint of a trust region subproblem is

$$\|d\|_2 \leq \Delta_k \quad . \tag{22}$$

But, simply adding the trust region constraint (22) to the quadratic program (5) may lead to an infeasible subproblem. There are mainly three approaches in constructing trust region subproblems. To simplify our discussions, first we consider only equality constraints.

The first approach is the null-space technique, which was studied by many authors, including Vardi [53], Byrd, Schnabel and Shultz [5], and Omojokun [36]. In this approach, the trial step s_k consists of a range space step v_k and the null space step h_k. v_k reduces the linearized constraint $||\nabla g(x_k)^T d + g(x_k)||$ and h_k reduces the approximate Lagrange function in the null space of $\nabla g(x_k)^T$. For example, v_k can be the least norm solution of

$$\begin{aligned} &\text{minimize} \quad ||\nabla g(x_k)^T v + g(x_k)||_2^2 \\ &v \in I\!R^n : \qquad ||v||_2 \leq \delta \Delta_k \quad , \end{aligned} \tag{23}$$

where $\delta \in (0, 1)$ is a given constant. The parameter $\delta < 1$ leaves freedom for defining h_k.

Once v_k is computed, h_k can be obtained by solving

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\,(v_k+h)^T B_k(v_k+h) + \nabla f(x_k)^T h \\ h \in I\!R^n: \quad & \nabla g(x_k)^T h = 0 \\ & ||h||_2 \le \sqrt{1-\delta^2}\Delta_k \ . \end{aligned} \tag{24}$$

The trust region step $s_k := v_k + h_k$ obtained by the null space technique can be regarded as the solution of the quadratic program

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\,d^T B_k d + \nabla f(x_k)^T d \\ d \in I\!R^n: \quad & \nabla g(x_k)^T d + \theta_k g(x_k) = 0 \\ & \|d\|_2 \le \Delta_k \end{aligned} \tag{25}$$

for some parameter $\theta_k \in (0,1]$.

The second approach is to replace all the linearized constraints in (5) by a single quadratic constraint, which was proposed by Celis, Dennis and Tapia [7],

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\,d^T B_k d + \nabla f(x_k)^T d \\ d \in I\!R^n: \quad & ||\nabla g(x_k)^T d + g(x_k)||_2 \le \xi_k \\ & ||d||_2 \le \Delta_k \ , \end{aligned} \tag{26}$$

where $\xi_k \ge \min_{||d||_2 \le \Delta_k} ||\nabla g(x_k)^T d + g(x_k)||_2$ to ensure the subproblem is feasible. One particular choice for ξ_k is given by Powell and Yuan [42]:

$$\min_{||d||_2 \le \delta_1 \Delta_k} ||\nabla g(x_k)^T d + g(x_k)||_2 \le \xi_k \le \min_{||d||_2 \le \delta_2 \Delta_k} ||\nabla g(x_k)^T d + g(x_k)||$$

where $0 < \delta_2 < \delta_1 < 1$ are two constants.

The third type of trust region subproblems is to minimize an approximation of some penalty function. Based on the L_∞ exact penalty, Yuan [58] suggests

$$\begin{aligned} \text{minimize} \quad & \Phi_k(d) := \tfrac{1}{2}\,d^T B_k d + \nabla f(x_k)^T d + \sigma_k ||\nabla g(x_k)^T d + g(x_k)||_\infty \\ d \in I\!R^n: \quad & ||d||_\infty \le \Delta_k \ , \end{aligned} \tag{27}$$

which can be converted into a standard QP. The penalty parameter $\sigma_k > 0$ may need to be updated to ensure the sufficient reduction in the norm of the residuals of linearized constraints. The L_∞ penalty function in (27) can be replaced by any other penalty functions, including the augmented Lagrange function studied by Niu and Yuan [34]. For example, if we use the L_1 penalty function, it leads to the famous Sl_1QP method given by Fletcher [17].

Once the trial step s_k is computed, a trust region algorithm for constrained optimization can be given as follows.

Algorithm 5.1. *(A General Trust Region SQP Method)*
Let $0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 < \tau_2 < 1$, and $x_0 \in I\!R^n$ be given.
For $k = 0, 1, 2, \ldots$ until convergence do

1) Compute the trial step s_k.

2) Compute the ratio between actual reduction and predicted reduction in a merit function Ψ_k,

$$r_k := \frac{\Psi_k(x_k) - \Psi_k(x_k + s_k)}{Pred_k}$$

3) Let

$$x_{k+1} := \begin{cases} x_k + s_k & \text{if } r_k > 0, \\ x_k & \text{otherwise,} \end{cases}$$

and

$$\Delta_{k+1} \in \begin{cases} [\tau_3 ||s_k||_2,\ \tau_4 \Delta_k] & \text{if } r_k < \tau_2, \\ [\Delta_k,\ \tau_1 \Delta_k] & \text{otherwise.} \end{cases}$$

4) Update B_{k+1} and define $\Psi_{k+1}(x)$.

For different trust region algorithms, $Pred_k$ and $\Psi_k(x)$ have to be chosen accordingly. For the L_∞ trust region algorithm of Yuan [58], we have that

$$\Psi_k(x) := f(x) + \sigma_k ||g(x)||_\infty$$

and

$$Pred_k := \Phi_k(0) - \Phi_k(s_k)\,,$$

where $\Phi_k(d)$ is defined in (27). The general requirements for $Pred_k$ are that it is always positive and it is a good approximation to the actual reduction in the merit function, particularly near the solution.

Unless the merit function $\Psi_k(x)$ is a differentiable penalty function, Maratos effect will occur. Thus, it is necessary to try a second order correction step $\hat{s}_k$ whenever the standard trust region trial step s_k is not acceptable. Normally, the second order correction step $\hat{s}_k$ is computed by solving another subproblem which is a slightly modification of the trust subproblem for obtaining s_k. For example, if s_k is obtained by (27), we can compute $\hat{s}_k$ by solving the following second order correction subproblem,

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2}\,(s_k + d)^T B_k (s_k + d) + \nabla f(x_k)^T d + \sigma_k ||\nabla g(x_k)^T d + g(x_k + s_k)||_\infty \qquad (28) \\ &d \in I\!R^n : \qquad\qquad ||s_k + d||_\infty \le \Delta_k \ . \end{aligned}$$

As the new subproblem differs from the original subproblem only by a term $g(x_k + s_k) - g(x_k) - \nabla g(x_k)^T s_k$ which is of magnitude of $||s_k||^2$, one can easily show that $\|\hat{s}_k\| = O(\|s_k\|^2)$, hence adding $\hat{s}_k$ will retain the nice superlinearly convergent property of the SQP step s_k. More important is that the second order correction step $\hat{s}_k$ will always make the merit function accept the new point $x_k + s_k + \hat{s}_k$ when x_k is close to the solution, which was first proved by Yuan [57].

Now, we give a brief discussion about inequality constraints. The inequality constraints in (1) can be transformed into equality constraints

$$g(x) - s = 0$$

by introducing slack variables $s = (s_1, ..., s_m)^T \in I\!R^m$, with the additional non-negative constraints:

$$s \geq 0. \tag{29}$$

The non-negative constraints (29) can be handled by the log-barrier function or by scaled trust region techniques, for example see Dennis, Heinkenschloss and Vincent [13], Coleman and Li [9], and Byrd, Gilbert and Nocedal [6].

For detailed discussions about trust region SQP methods, we recommend the related chapters in the monograph by Conn, Gould and Toint [10] and the review paper of Yuan [59].

6 Stabilization by a Filter

Filter method was first given by Fletcher and Leyffer [20]. This approach is based on viewing the constrained optimization problem (1) as two separate minimization problems,

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &x \in I\!R^n \end{aligned}$$

and

$$\begin{aligned} &\text{minimize} \quad h(g(x)) \\ &x \in I\!R^n \quad , \end{aligned}$$

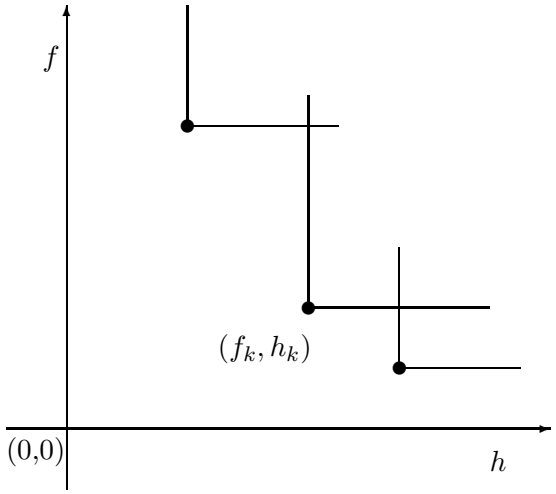
where

$$h(g(x)) := ||g^-(x)||_1 = \sum_{j=1}^{m} |\min(0, g_j(x))| \quad .$$

For each iterate point x_k, there is a corresponding pair

$$(f_k, h_k) = (f(x_k), ||g^-(x_k)||_1) \quad ,$$

see the subsequent sketch.



The concept of domination is crucial for the filter method.

Definition 6.1. *A pair (f_k, h_k) is said to dominate another pair (f_l, h_l) if and only if both $f_k < f_l$ and $h_k < h_l$.*

The idea of the filter approach is that a new point is acceptable as long as it is not dominated by any previous iterates. Such an acceptable point will be added to the filter which is used to control subsequent iterates.

Definition 6.2. *A filter is a list of pairs (f_l, h_l) such that no pair dominates any other. A point (f_k, h_k) is said to be acceptable for inclusion in the filter if it is not dominated by any point in the filter.*

At each iteration, a Filter-SQP algorithm needs to construct a QP subproblem. Fletcher and Leyffer [20] uses the subproblem

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\, d^T B_k d + \nabla f(x_k)^T d \\ d \in I\!R^n : \quad & \nabla g(x_k)^T d + g(x_k) \geq 0 \\ & ||d||_\infty \leq \Delta_k \quad . \end{aligned} \tag{30}$$

The outlines of a general filter-SQP algorithm can be stated as follows.

Algorithm 6.1. *(A General Filter-SQP Algorithm)*
Given $x_0 \in I\!R^n$, let $k := 1$. Let the initial filter be the pair $\{(f_0, h_0)\}$.
For $k = 0, 1, 2, \ldots$ until convergence do

1) *Solve the QP subproblem obtaining d_k.*
2) *Provisionally set $x_{k+1} := x_k + d_k$.*
3) *If (f_{k+1}, h_{k+1}) is dominated by some pair in the filter then $x_{k+1} := x_k$. Otherwise, accept x_{k+1}, add (f_{k+1}, h_{k+1}) to the filter and remove all points dominated by (f_{k+1}, h_{k+1}) from the filter.*
4) *Construct the QP subproblem for the next iteration.*

It is well known that when using a penalty function to ensure global convergence of nonlinear optimization algorithms, we normally require some sufficient decrease in the penalty function. Similarly, in the implementation of a filter method, the simple domination has to be replaced by some sufficient domination. Fletcher and Leyffer [20] suggest to accept the provisional point x_{k+1} to the filter if either

$$h_{k+1} \leq 0.99 h_l$$

or

$$f_{k+1} \leq f_k - \max(0.25\Delta q_l, 10^{-4} h_l \mu_l)$$

holds for all filter entries l, where $\Delta q_l := -\frac{1}{2} d_l^T B_l d_l - \nabla f(x_l)^T d_l$ is the predicted reduction of $f(x)$ based on the quadratic model, λ_l is the Lagrange multiplier of (30), and μ_l is the projection of the least power of ten larger than $||\lambda_l||_\infty$ in the interval $[10^{-6}, 10^6]$.

The trust region QP subproblem (30) may become infeasible, either because the trust region radius Δ_k is too small or due to the inconsistent of the linearized constraints. Thus, a restoration phase have to be used to move the iterate towards the feasible region. This is achieved by minimizing the norm of the residuals of the linearized constraints in the trust region.

7 Convergence

There remains the question whether the convergence of an SQP method can be proved in a mathematically rigorous way. In fact there exist numerous theoretical convergence results in the literature, see e.g. Boggs and Tolle [2]. We want to give here only an impression about the type of these statements, and repeat some results that have been stated in the early days of the SQP methods.

In the first case we consider the global convergence behavior, i.e., the question, whether the SQP methods converges when starting from an arbitrary initial point. Suppose that the augmented Lagrangian merit function (18) is implemented and that the primal and dual variables are updated in the form (15).

Theorem 7.1. *Let $\{(x_k, v_k)\}$ be a bounded iteration sequence of the SQP algorithm with a bounded sequence of quasi-Newton matrices $\{B_k\}$ and assume that*

$$d_k^T B_k d_k \geq \gamma d_k^T d_k \tag{31}$$

for all k and a $\gamma > 0$. Then there is an accumulation point of $\{(x_k, v_k)\}$ satisfying the Karush-Kuhn-Tucker conditions for (1).

The statement of the theorem is quite weak. But without any further information about second derivatives, we cannot guarantee that the approximated point is indeed a local minimizer.

To investigate now the local convergence speed, we start from an initial $x_0 \in \mathbb{R}^n$ sufficiently close to an optimal solution. Normally, we find the following assumptions for local convergence analysis.

Assumption 7.1. *Assume that*

1) $x_k \to x^*$,
2) $f(x)$, $g_1(x), \ldots, g_m(x)$ *are at least twice continuously differentiable,*
3) $g_i(x^*) = 0$ *for* $i = 1, \ldots, m$, *i.e., we know all active constraints,*
4) $\nabla g_i(x^*)$ *are linearly independent for* $i = 1, \ldots, m$, *i.e., the constraint qualification is satisfied,*
5) x^* *is a KKT point and* u^* *is the corresponding Lagrange multiplier,*
6) *The inequality*

$$d^T \nabla_{xx}^2 L(x^*, u^*) d > 0$$

holds for all non-zero vectors d satisfying $\nabla g(x^*)^T d = 0$, i.e., a second order sufficient condition is satisfied.

The fundamental result for local convergence analysis of SQP methods is the following theorem.

Theorem 7.2. *Assume that the conditions in Assumption 7.1 are satisfied. Let d_k be the solution of*

$$\begin{aligned} &\textit{minimize} \quad \tfrac{1}{2} d^T B_k d + \nabla f(x_k)^T d \\ &d \in I\!R^n : \quad \nabla g(x_k)^T d + g(x_k) = 0 \ . \end{aligned} \tag{32}$$

Then, the limit

$$\lim_{k\to\infty} \frac{||x_k + d_k - x^*||}{||x_k - x^*||} = 0 \tag{33}$$

holds if and only if

$$\lim_{k\to\infty} \frac{||P_k(B_k - \nabla^2_{xx} L(x^*, u^*) d_k||}{||d_k||} = 0 \tag{34}$$

where P_k is the projection from $I\!R^n$ onto the null space of $\nabla g(x_k)^T$, and $\nabla^2_{xx} L(x^*, u^*)$ is the Hessian of the Lagrangian at the solution x^*.

This important result was first given by Boggs, Tolle and Wang [3] and Powell [40]. A nice proof for this result can be found in Stoer and Tapia [48]

In order to apply Theorem 7.2 to establish suplinearly convergence results for various SQP methods, the general approach is to prove that $x_{k+1} = x_k + d_k$ holds eventually, where d_k is the solution for the QP subproblem (32). Thus, for line search SQP methods we need to prove that the unit step-length $\alpha_k = 1$ can be accepted by the correspoding line search conditions. While for trust region SQP algorithms, one has to show that the trust region constraint $||d|| \le \Delta_k$ will be inactive and the d_k can be accepted by the algorithm for all large k. And, for filter SQP algorithms, it is required to show that $x_k + d_k$ will be accepted by the filter for all large k.

However, no matter line search or trust region is used for globalization of the SQP method, due to the Maratos effect, $x_k + d_k$ may not be accepted though (33) is true, when the merit function is a non-smooth exact penalty function. Even for filter methods, Maratos efffect can also happen because a superlinearly convergent SQP step d_k can sometimes increase both the objective function f and the constraint violations $h(g)$. To overcome the Maratos effect, there are many techniques, which can be grouped into four categories. The first one is the watch-dog technique, in which standard line search conditions are relaxed at some iterations. The watchdog technique was proposed by Chamberlain et. al [8]. The second is the second order correction technique, where an additional subproblem is solved to obtain an accepted full step, for example see Fletcher [18], Mayne and Polak [33] and Fukushima [22]. The third is to use a smooth exact penalty function or the augmented Lagrange function as the merit function, for example see Schittkowski [45], Powell and Yuan [41, 42], and Ulbrich [51]. The fourth is the non-monotone technique, for example, see Ulbrich and Ulbrich [52], and Gould and Toint [28].

From the characterization of the local convergence property of the SQP method given in Theorem 7.2, the superlinearly convergence of a specific SQP method depends on

the relation (34), which is, generally, hard to verify directly if B_k is updated by quasi-Newton updates. The standard approach is more or less following the sophisticated and elegant technique for analyzing quasi-Newton methods for unconstrained optimization given by Dennis and Moré [14, 15]. Their technique is to prove the local Q-superlinearly convergence by two steps. The first step is to show that the sequence $\{x_k\}$ converges R-linearly. Then, the R-linear convergence of the sequence is used to obtain more local convergence properties, such as the following relations

$$\sum_{k=1}^{\infty} ||x_{k+1} - x_k|| < \infty\,, \qquad \sum_{k=1}^{\infty} ||x_k - x^*|| < \infty\,,$$

the uniformly boundedness of B_k and B_k^{-1}, and the limit (34), in order to establish the Q-superlinear convergence of the sequence $\{x_k\}$.

However, for a given specific quasi-Newton method, there is still a gap to apply the beautiful convergence results discussed above. For example, it is not easy to know whether condition (31) holds for a particular quasi-Newton updates. Moreover, to ensure the update formulae generate positive definite matrices, we always require $s_k^T y_k > 0$, a condition is normally satisfied for unconstrained optimization. However, for constrained optimization problems, as the solution (x^*, u^*) is only a saddle point of the Lagrange function, the vector y_k defined by (9) can not guarantee $s_k^T y_k > 0$. Powell [39] suggests replacing y_k by the vector

$$\bar{y}_k = \begin{cases} y_k, & \text{if } s_k^T y_k > 0.2 s_k^T B_k s_k\ , \\ t_k y_k + (1 - t_k) B_k s_k, & \text{otherwise}\ , \end{cases} \tag{35}$$

where $t_k = 0.8 s_k^T B_k s_k / (s_k^T B_k s_k - s_k^T y_k)$. With this modification, Powell [39] proved the following remarkable result for the SQP method with BFGS update.

Theorem 7.3. *Assume that the above assumptions are valid and $\alpha_k = 1$ for all k. Then the sequence $\{x_k\}$ converges R-superlinearly, i.e.,*

$$\lim_{k\to\infty} \|x_{k+1} - x^\star\|^{1/k} = 0\ .$$

The R-superlinear convergence speed of $\{x_k\}$ is somewhat weaker than the Q-superlinear convergence rate of $\{z_k := (x_k, v_k)\}$ defined below which was first proved in the form

$$\lim_{k\to\infty} \frac{\|z_{k+1} - z^\star\|}{\|z_k - z^\star\|} = 0$$

for the so-called DFP update formula, i.e. a slightly different quasi-Newton method. In this case, we get a sequence β_k tending to zero with

$$\|z_{k+1} - z^\star\| \le \beta_k \|z_k - z^\star\|\ .$$

But, it should be noted that the Q-superlinear convergence of z_k does not imply the Q-superlinear convergence of x_k.

8 Acknowledgement

This article was written during the second author's visit at Bayreuth supported by Alexander-von-Humboldt Foundation. The second author would like to thank Humboldt Foundation for the support and thank the first author for being his host.

References

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

[2] Boggs P.T., Tolle J.W. (1995): *Sequential quadratic programming*, Acta Numerica, Vol. 4, 1 - 51

[3] Boggs P.T., Tolle J.W., Wang P. (1982): *On the local convergence of quasi-Newton methods for constrained optimization,*, SIAM Journal on Control and Optimization, Vol. 20, 161–171

[4] Byrd R.H. (1985): *An example of irregular convergence in some constrained optimization methods that use prjected Hessian,* Mathematical Programming, Vol. 32, 232–237

[5] Byrd R.H., Schnabel R.B., Shultz G.A. (1987): *A trust region algorithm for nonlinearly constrained optimization,* SIAM Journal on Numererical Analysis, Vol. 24, 1152–1170

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

[7] Celis M.R., Dennis Jr.J.E., Tapia R.A. (1985): *A trust region algorithm for nonlinear equality constrained optimization,* in: Boggs P.T., Byrd R.H. Schnabel R.B. eds., *Numerical Optimization*, SIAM, Philadelphia, pp. 71–82

[8] Chamberlain R.M., Lemarechal C., Pedersen H.C., Powell M.J.D. (1982): *The watchdog technique for forcing convergence in algorithms for constrained optimization,* Mathematical Programming Study, Vol. 16, 1–17

[9] Coleman T.F., Li Y. (2000): *A trust region and affine scaling interior point method for nonconvex minimization with linear inequality constraints,* Mathematical Programming Vol. 88, 1–31

[10] Conn A.R., Gould N.I.M, Toint Ph.L. (2000): *Trust Region Methods,* MPS/SIAM Series on Optimization, SIAM, Philadelphia, PA.

[11] Dai Y.H. (2002): *On the nonmonotone line search,* Journal of Optimization Theory and Applications, Vol. 112, 315–330

[12] Dai Y.H., Schittkowski K. (2008): *A sequential quadratic programming algorithm with non-monotone line search*, Pacific Journal of Optimization, Vol. 4, 335-351

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

[14] Dennis, J.E., Moré J.J. (1974): *A charaterization of superlinear convergence and its application to quasi-Newton methods,* Mathematics of Computation, Vol. 28, 549–560

[15] Dennis, J.E., Moré, J.J. (1977): *Quasi-Newton methods, motivation and theory,* SIAM Review, Vol. 19, 46–89

[16] Edgar T.F., Himmelblau D.M. (1988): *Optimization of Chemical Processes,* McGraw Hill

[17] Fletcher R. (1982): *A model algorithm for composite non-differentiable optimization problems,* Mathenmatical Programming, Vol. 17, 67–76.

[18] Fletcher R. (1982): *Second order correction for nondifferentiable optimization,* in: Watson G.A., ed., Numerical Analysis, Springer, Berlin, pp. 85–115

[19] Fletcher R. (1987): *Practical Methods of Optimization*, John Wiley, Chichester

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

[21] Fletcher R., Leyffer S. Toint Ph. (2002): *On the global convergence of a filter–SQP algorithm,* SIAM Journal on Optimization, Vol. 13, 44–59

[22] Fukushima M. (1986): *A succesive quadratic programming algorithm with global and superlinear convergence properties,* Mathematical Programming, Vol. 35, 253–264

[23] Gill P.E., Murray W., Saunders M.A. (1975): *Methods for computing and modifying the LDL factors of a matrix*, Mathematics of Computation, Vol. 29, 1051-1077

[24] Gill P.E., Murray W., Saunders M.A., Wright M. (1992): *Some theoretical properties of an augmented Lagrangian merit function*, in: Advances in Optimization and Parallel Computing, P.M. Pardalos ed., 101-129, North Holland

[25] Gill P.E., Murray W., Wright M. (1982): *Practical Optimization*, Academic Press, London

[26] Goldfarb D., Idnani A. (1983): *A numerically stable method for solving strictly convex quadratic programs*, Mathematical Programming, Vol. 27, 1-33

[27] Gould N.I.M, Toint Ph.L. (1999): *SQP methods for large-scale nonlinear programming,*, Proceedings of the 19th IFIP TC7 Conference on System Modelling and Optimization: Methods, Theory and Applications, pp. 149–178.

[28] Gould N.I.M, Toint Ph.L. (2006): *Global convergence of a non-monotone trust-region filter algorithm for nonlinear programming,* in: Hager W.W., Huang S., Pardalos P.M., Prokopyev O.A. eds., Multiscale Optimization Methods and Applications, Springer, New York, pp. 125–150

[29] Grippo L., Lampariello F., Lucidi S. (1986): *A nonmonotone line search technique for Newtons's method,* SIAM Journal on Numerical Analysis, Vol. 23, 707–716

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

[31] Han S.-P. (1977): *A globally convergent method for nonlinear programming* Journal of Optimization Theory and Applications, Vol. 22, 297–309

[32] Maratos N. (1978): *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*, Ph.D. Thesis, Imperial College Sci. Tech., University of London

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

[34] Niu L.F., Yuan Y. (2010): *A new trust-region algorithm for nonlinear constrained optimization,* Journal on Computational Mathematics, Vol. 28, 72–86

[35] Nocedal J., Overton M.L. (1985): *Projected Hessian update algorithms for nonlinear constrained optimization,* SIAM Journal on Numerical Analysis, Vol. 22, 821–850

[36] Omojokun E.O. (1989): *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints,*, Ph.D. Thesis, University of Colorado at Boulder, USA

[37] Papalambros P.Y., Wilde D.J. (1988): *Principles of Optimal Design,* Cambridge University Press

[38] Powell M.J.D. (1978): *A fast algorithm for nonlinearly constraint optimization calculations,* in: Numerical Analysis, G.A. Watson ed., Lecture Notes in Mathematics, Vol. 630, Springer

[39] Powell M.J.D. (1978): *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in: Nonlinear Programming 3, O.L. Mangasarian, R.R. Meyer, S.M. Robinson eds., Academic Press

[40] Powell M.J.D. (1983): *Variable metric methods for constrained optimization,* in: Bachem A., Grötschel M., and Korte B. eds., Mathematical Programming: the State of the Art, Springer, Berlin. pp. 288–311

[41] Powell M.J.D. and Yuan Y. (1986) *A recursive quadratic programming algorithm that uses differentiable exact penalty function,* Mathematical Programming, Vol. 35, 265–278

[42] Powell M.J.D. and Yuan Y. (1991) *A trust region algorithm for equality constrained optimization,* Mathematical Programming, Vol. 49, 189–211

[43] Rockafellar T. (1974): *Augmented Lagrangian multiplier functions and duality in nonconvex programming*, SIAM Journal of Control, Vol. 12, 268-285

[44] Schittkowski K. (1980): *Nonlinear Programming Codes,* Lecture Notes in Economics and Mathematical Systems, Vol. 183 Springer

[45] Schittkowski K. (1983): *On the convergence of a sequential quadratic programming method with an augmented Lagrangian search direction,* Optimization, Vol. 14, 197-216

[46] Sun W.Y., Yuan Y. (2006) *Optimization Theory and Methods: Nonlinear Programming*, Springer, New York

[47] Stoer J. (1985): *Foundations of recursive quadratic programming methods for solving nonlinear programs,* in: Computational Mathematical Programming, K. Schittkowski, ed., NATO ASI Series, Series F: Computer and Systems Sciences, Vol. 15, Springer

[48] Stoer J., Tapia R.A. (1987): *On the characterization of q-superlinear convergence of quasi-Newton methods for constrained optimization,*, Mathematics of Computation, Vol. 49, 581–584

[49] Toint P.L. (1996): *An assessment of nonmontone line search techniques for unconstrained optimization,* SIAM Journal on Scientific Computing, Vol. 17, 725–739

[50] Toint P.L. (1997): *A nonmonotone trust-region algorithm for nonlinear optimization subject to convex constraints,* Mathematical Programming, Vol. 77, 69–94

[51] Ulbrich S. (2004): *On the superlinear local convergence of a filter-SQP method,* Mathematical Programming, Vol. 100, 217–245

[52] Ulbrich M., Ulbrich S. (2003): *Non-monotone trust region methods for nonliear equality constrained optimization without a penalty function,* Mathematical Programming Ser B, Vol. 95, 103–135

[53] Vardi A. (1985): *A trust region algorithm for equality constrained minimization: convergence properties and implementation,* SIAM Journal on Numerical Analysis, Vol. 22, 575–591

[54] Wilson R.B. (1963): *A simplicial algorithm for concave programming*, Ph.D. Thesis, Harvard University

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

[56] Yuan Y. (1985): *An only 2-step Q-superlinear convergence example for some algorithm that used reduced Hessian approximation,* Mathematical Programming, Vol. 32, 269–285

[57] Yuan Y. (1985): *On the superlinear convergence of a trust region algorithm fornonsmooth optimization,* Mathematical Programming, Vol. 31, 269–285

[58] Yuan Y. (1995): *On the convergence of a new trust region algorithm,* Numerische Mathematik, Vol. 70, 515–539

[59] Yuan Y. (2000) *A review of trust region algorithms for optimization,* in: J.M. Ball and J.C.R. Hunt, eds., ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics, Oxford University Press, 271–282