SQP and PDIP algorithms for Nonlinear Programming

November 2, 2007

Penalty and barrier methods are indirect ways of solving constrained optimization problems, using techniques developed in the unconstrained optimization realm. In what follows we shall give the foundation of two more direct ways of solving constrained optimization problems, namely Sequential Quadratic Programming (SQP) methods and Primal-Dual Interior Point (PDIP) methods.

1 Sequential Quadratic Programming

For the derivation of the Sequential Quadratic Programming method we shall use the equality constrained problem

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \end{array} \tag{ECP}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions. An understanding of this problem is essential in the design of SQP methods for general nonlinear programming problems.

The KKT conditions for this problem are given by

$$\boldsymbol{\nabla} f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \boldsymbol{\nabla} h_i(\mathbf{x}) = \mathbf{0} \tag{1a}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{1b}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ are the Lagrange multipliers associated with the equality constraints. If we use the Lagrangian

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) \tag{2}$$

we can write the KKT conditions (1) more compactly as

$$\begin{pmatrix} \boldsymbol{\nabla}_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \\ \boldsymbol{\nabla}_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \end{pmatrix} = \mathbf{0}. \tag{EQKKT}$$

As with Newton's method unconstrained optimization, the main idead behind SQP is to model problem (ECP) at a given point $\mathbf{x}^{(k)}$ by a quadratic programming subrpoblem and then use the solution to this problem to construct a more accurate approximation $\mathbf{x}^{(k+1)}$. If we perform a Taylor series expansion of system (EQKKT) about $(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})$ we obtain

$$\begin{pmatrix} \boldsymbol{\nabla}_{\mathbf{x}} \mathcal{L}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}) \\ \boldsymbol{\nabla}_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}) \end{pmatrix} + \begin{pmatrix} \boldsymbol{\nabla}^2_{\mathbf{x}} \mathcal{L}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}) & \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)}) \\ \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)})^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\boldsymbol{\lambda} \end{pmatrix} = \mathbf{0},$$

where $\delta\mathbf{x} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$, $\delta\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^{(k)}$, and

$$\boldsymbol{\nabla}^2_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \boldsymbol{\nabla}^2 f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \boldsymbol{\nabla}^2 h_i(\mathbf{x})$$

is the Hessian matrix of the Lagrangian function. The Taylor series expansion can be written equivalently as

$$\begin{pmatrix} \boldsymbol{\nabla}^2_{\mathbf{x}} \mathcal{L}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}) & \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)}) \\ \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)})^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\nabla} f(\mathbf{x}^{(k)}) - \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)})\boldsymbol{\lambda}^{(k)} \\ -\mathbf{h}(\mathbf{x}^{(k)}) \end{pmatrix},$$

or, setting $\mathbf{d} = \delta\mathbf{x}$ and bearing in mind that $\boldsymbol{\lambda}^{(k+1)} = \delta\boldsymbol{\lambda} + \boldsymbol{\lambda}^{(k)}$

$$\begin{pmatrix} \boldsymbol{\nabla}^2_{\mathbf{x}} \mathcal{L}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}) & \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)}) \\ \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)})^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{d} \\ \boldsymbol{\lambda}^{(k+1)} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\nabla} f(\mathbf{x}^{(k)}) \\ -\mathbf{h}(\mathbf{x}^{(k)}) \end{pmatrix}. \tag{LNS}$$

Algorithm (1) summarizes the Newton-Lagrange method for solving problem (ECP).

Algorithm 1 Lagrange-Newton method

1: Determine $(\mathbf{x}^{(0)}, \boldsymbol{\lambda}^{(0)})$
2: Set $k := 0$
3: **repeat**
4: Solve the Lagrange-Newton system (LNS) to determine $(\mathbf{d}^{(k)}, \boldsymbol{\lambda}^{(k+1)})$
5: Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{d}^{(k)}$
6: Set $k := k + 1$
7: **until** Convergence

In SQP methods, problem (ECP) is modeled by a quadratic programming subproblem (QPS for short), whose optimality conditions are the same as in the Lagrange-Newton system (LNS). The algorithm is the same as that of the Newton-Lagrange method, but instead of solving system (LNS) in Step 4 we solve the following QPS :

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & \nabla f(\mathbf{x}^{(k)})^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T \nabla^2_{\mathbf{x}} \mathcal{L}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})\mathbf{d} \\ \text{subject to} & \mathbf{h}(\mathbf{x}^{(k)}) + \nabla \mathbf{h}(\mathbf{x}^{(k)})^T \mathbf{d} = \mathbf{0}. \end{array} \qquad (\mathrm{ECQP}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}))$$

It is really straightforward to verify that the first order conditions for the previous problem at $(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})$ are given by the Lagrange-Newton system (LNS), and therefore $\mathbf{d}^{(k)}$ is a stationary point of $(\mathrm{ECQP}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}))$. If in addition $\mathbf{d}^{(k)}$ satisfies second order sufficient conditions , then $\mathbf{d}^{(k)}$ minimizes problem $(\mathrm{ECQP}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}))$.

We also observe that the constraints in $(\mathrm{ECQP}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}))$, are derived by a first order Taylor series approximation of the constraints of the original problem (ECP). The objective function of the QPS is a truncated second order Taylor series expansion of the Lagrangian function. This choice is justified in the next few lines.

1.1 The choice of the objective function in the QPS

The most natural option for the objective of the QPS, would be a second order Taylor series expansion of the objective function f instead of a second order Taylor series expansion of the Lagrangian function. This is a rather subtle but really important point for SQP algorithms. The choice of f in the objective might make the method to break down. This can be illustrated by problem

$$\begin{array}{ll} \text{minimize} & -x_1 - \frac{1}{2}x_2^2 \\ \text{subject to} & 1 - x_1^2 - x_2^2 = 0. \end{array}$$

The QPS taking f into account, would be

$$\begin{array}{ll} \text{minimize} & \boldsymbol{\nabla} f(\mathbf{x}^{(k)})^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T \boldsymbol{\nabla}^2 f(\mathbf{x}^{(k)})\mathbf{d} \\ \text{subject to} & \mathbf{h}(\mathbf{x}^{(k)}) + \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)})^T \mathbf{d} = \mathbf{0} \end{array}$$

or if we substitute for

$$\boldsymbol{\nabla} f(\mathbf{x}) = \begin{pmatrix} -1 \\ -x_2 \end{pmatrix}, \boldsymbol{\nabla}^2 f(\mathbf{x}) = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}, \boldsymbol{\nabla} g(\mathbf{x}) = \begin{pmatrix} -2x_1 \\ -2x_2 \end{pmatrix}$$

then the problem can be written as

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -d_1 - 2x_2 d_2 - d_2^2 \\ \text{subject to} & -2x_1 d_1 - 2x_2 d_2 + 1 - x_1^2 - x_2^2 = 0. \end{array}$$

Point $(1, 0)$ satisfies first order necessary and second order sufficient conditions. At point $(1 + \epsilon, 0)$ though, the quadratic programming problem becomes

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -d_1 - \frac{1}{2}d_2^2 \\ \text{subject to} & -2d_1 - 2\epsilon d_1 + 1 - (1 + \epsilon)^2 = 0, \end{array}$$

which is unbounded, no matter how small ϵ is[1]. If we add second order information from the constraints to the objective of the QPS, then the method is well defined and second order convergence can be attained.

The Sequential Quadratic Programming method is identical to the Lagrange-Newton method (Algorithm (1)), with the only difference being that in Step 4 we solve problem (ECQP($\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}$)) instead of (LNS). In fact SQP is preferred over the Lagrange-Newton iteration, because the latter can converge to a KKT point that does not satisfy second order conditions, *i.e.* is not a minimizer.

The convergence properties of the method can be summarized in the following theorem :

[1] d_1 is uniquely identified by the linearized constraint, but d_2 can take any arbitrary value.

Theorem 1 *If* $\mathbf{x}^{(0)}$ *is sufficiently close to* $\mathbf{x}^*$, *if the Lagrangian matrix*

$$\begin{pmatrix} \nabla^2_{\mathbf{x}}\mathcal{L}(\mathbf{x}^{(0)}, \boldsymbol{\lambda}^{(0)}) & \nabla\mathbf{h}(\mathbf{x}^{(0)}) \\ \nabla\mathbf{h}(\mathbf{x}^{(0)})^T & \mathbf{0} \end{pmatrix}$$

is non-singular, if second order sufficiency conditions hold at $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ *and the Jacobian of the constraints is of full rank, then Algorithm (1) converges and the rate is second order.*

The technicalities of the proof are not presented here, but can be found in Fletcher [1987] or Conn et al. [2000b]. It is interested to remark here that the initial estimate of the Lagrange multiplier $\boldsymbol{\lambda}^{(0)}$ plays a minor role in the overall convergence of the algorithm.

1.2 SQP for general NLP problems

The Sequential Quadratic Programming framework can be extended to the general nonlinear constrained problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \begin{array}{ll} \mathbf{h}(\mathbf{x}) & = \mathbf{0} \\ \mathbf{g}(\mathbf{x}) & \geq \mathbf{0} \end{array} \end{array}.$$

The steps are the same as those of Algorithm (1), with the QPS defined as

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & \nabla f(\mathbf{x}^{(k)})^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T\nabla^2_{\mathbf{x}}\mathcal{L}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})\mathbf{d} \\ \text{subject to} & \begin{array}{ll} \mathbf{h}(\mathbf{x}^{(k)}) + \nabla\mathbf{h}(\mathbf{x}^{(k)})^T\mathbf{d} & = \mathbf{0} \\ \mathbf{g}(\mathbf{x}^{(k)}) + \nabla\mathbf{g}(\mathbf{x}^{(k)})^T\mathbf{d} & \geq \mathbf{0}. \end{array} \end{array}$$

Near the solution of generally constrained problems, only constraints that are satisfied as equalities affect the solution. Therefore in order to solve general problems, one has to define a strategy for identifying constraints that will be active in the solution (as is done in the case of generally constrained quadratic problems). In the design of SQP algorithms we exploit the fact that a QPS is solved, and identification of constraints that affect the solution is delegated to the QPS. This has been shown to work in practice and some assumptions that are implied by the majority of problems to be solved.

1.3 Notes and References

Sequential quadratic programming methods were mainly developed having in mind to ensure rapid convergence when close to the solution. The basic idea is to use Newton's method to find a stationary point of the Lagrangian function, hence the name Lagrange-Newton methods. As such though, SQP suffers from the same problems as Newton's method.

SQP was first introduced by Wilson [1963]. Fletcher [1970] proposed to solve constrained minimization problems via a sequence of quadratic programming subproblems. Bartholomew-Biggs [1972] also proposed algorithms that use a sequence of quadratic programming subproblems. Garcia-Palomares [1973] and Garcia-Palomares and Mangasarian [1976] studied the use of quasi-Newton methods in SQP frameworks. SQP algorithms received great attention after the work of Han [1976, 1977] and Powell [1977, 1978a,b]. Local convergence properties of SQP methods have been studied by Glad [1979], Bertsekas [1980], Schittkowski [1983], Fukushima [1986], Fletcher [1987], Fontecilla [1988], Coleman and Feynes [1992] and Panier and Tits [1993]. A survey of SQP methods is given by Boggs and Tolle [1995]. Gould and Toint [1999] survey SQP methods for large scale optimization.

2 Primal-Dual Interior Point Methods

Interior point methods were widely used in the form of barrier methods. In linear programming though, the simplex method dominated, mainly due to the inefficiencies of barrier methods. Interior point methods became quite popular again after 1984, when Karmarkar announced a fast polynomial-time interior point method for nonlinear programming [Karmarkar, 1984].

For convenience, we introduce the following form of a nonlinear programming problem, in which inequality constraints are taken to be variable bounds

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \begin{array}{lcl} \mathbf{h}(\mathbf{x}) & = & \mathbf{0} \\ \mathbf{x} & \geq & \mathbf{0}. \end{array} \end{array} \qquad \text{(NLPIP)}$$

The method introduced here can handle nonlinear inequality constraints. Variable bounds are chosen merely for ease of demonstration. The KKT conditions of problem (NLPIP) can be written in matrix form as

$$\mathbf{F}(\mathbf{w}) = \begin{pmatrix} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{w}) \\ \mathbf{h}(\mathbf{x}) \\ \mathbf{XZ} \end{pmatrix} = \mathbf{0}, \tag{KKTIP}$$

where, as is traditional in primal-dual interior point methods, we use $(\mathbf{y}, \mathbf{z})$ to denote the Lagrange multiplier vectors associated with equalities and inequalities, respectively and $\mathbf{w}$ to denote the triple $(\mathbf{x}, \mathbf{y}, \mathbf{z})$. If $\mathbf{x}, \mathbf{z}$ are vectors of the form

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad \mathbf{z} = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix},$$

then $\mathbf{X}, \mathbf{Z}$ denote diagonal matrices with diagonal $\mathbf{x}, \mathbf{z}$, i.e.

$$\mathbf{X} = \begin{pmatrix} x_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & x_n \end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix} z_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & z_n \end{pmatrix}$$

$\mathbf{e}$ is a vector of all ones whose dimension varies with the context. In this notation the Lagrangian function of problem (NLPIP) can be written as

$$\mathcal{L}(\mathbf{w}) = f(\mathbf{x}) + \mathbf{y}^T \mathbf{h}(\mathbf{x}) - \mathbf{z}^T \mathbf{x}.$$

If we use the logarithmic barrier function, problem (NLPIP) can be written equivalently as

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}; \rho) \triangleq f(\mathbf{x}) - \rho \sum_{i=1}^{n} \log(x_i) \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{array} \tag{BNLP}$$

where $\mathbf{x} > 0$. The KKT conditions of this problem are

$$\nabla f(\mathbf{x}) - \nabla \mathbf{h}(\mathbf{x})\mathbf{y} - \rho \mathbf{X}^{-1}\mathbf{e} = \mathbf{0} \tag{3a}$$
$$\mathbf{h}(\mathbf{x}) = \mathbf{0}. \tag{3b}$$

Introducing $\mathbf{z} = \rho \mathbf{X}^{-1}\mathbf{e}$ (or equivalently $\mathbf{XZe} = \rho \mathbf{e}$), (3) are written as

$$\nabla f(\mathbf{x}) - \nabla \mathbf{h}(\mathbf{x})\mathbf{y} - \mathbf{z} = \mathbf{0} \tag{4a}$$
$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{4b}$$
$$\mathbf{z} = \rho \mathbf{X}^{-1}\mathbf{e} \tag{4c}$$

or using the Lagrangian function, we can write in matrix form

$$\mathbf{F}(\mathbf{w};\rho) = \begin{pmatrix} \boldsymbol{\nabla}_{\mathbf{x}}\,\mathcal{L}(\mathbf{w}) \\ \mathbf{h}(\mathbf{x}) \\ \mathbf{XZ} - \rho\mathbf{e} \end{pmatrix} = \mathbf{0}, \qquad \text{(PRTKKT)}$$

which are called the perturbed KKT conditions. It is obvious from Eqs. (PRTKKT), (KKTIP) that the perturbed KKT conditions differ from the KKT conditions of the original problem only in the complementarity conditions.

Definition 2.1 (Central path) *A point* $\mathbf{w}(\rho)$ *is said to belong to the central path* $\mathcal{C}$*, if it is the solution of* (PRTKKT) *for a fixed value of* $\rho > 0$.

As $\rho \to 0$, the perturbed KKT conditions approximate the original KKT conditions more and more accurately and $\mathbf{w}(\rho)$ converges to the solution of the KKT conditions along the central path (Definition (2.1)). The central path has been studied by Sonnevend [1986] and Megiddo [1988].

Primal-Dual Interior Point methods (PDIP for short) involve inner and outer iterations. Inner iterations use Newton's method to solve system (PRTKKT) for a fixed value of the barrier parameter. The first order change of the perturbed KKT conditions is

$$\begin{pmatrix} \boldsymbol{\nabla}^2_{\mathbf{x}}\,\mathcal{L}(\mathbf{w}^{(k)}) & -\boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)}) & -\mathbf{I} \\ \boldsymbol{\nabla}\mathbf{h}(\mathbf{x}^{(k)})^T & \mathbf{0} & \mathbf{0} \\ \mathbf{Z}^{(k)} & \mathbf{0} & \mathbf{X}^{(k)} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}^{(k)} \\ \delta\mathbf{y}^{(k)} \\ \delta\mathbf{z}^{(k)} \end{pmatrix} = -\mathbf{F}(\mathbf{w}^{(k)};\rho). \qquad \text{(PRTLNS)}$$

Outer iterations decrease the value of the barrier parameter so that in the limit $\delta\mathbf{w}$ is the Newton solution of (KKTIP). Algorithm (2) summarizes the framework of primal-dual interior point methods.

2.1 Notes and References

In the derivation of the complementarity condition of the perturbed KKT conditions we have introduced the nonlinear transformation $\mathbf{XZe} = \rho\mathbf{e}$, which is essential for the numerical success of the method and is discussed in [El-Bakry et al., 1996].

Convergence results for PDIP methods have been given by Wright [1992] and Conn et al. [2000a]. Local convergence results for PDIP methods have

Algorithm 2 Primal-Dual Interior Point method

Set $l := 0$
Determine $\mathbf{w}^{(0)}$
repeat
  Set $k := 0$
  repeat
    Solve system (PRTLNS) to determine $(\delta\mathbf{x}^{(k)}, \delta\mathbf{y}^{(k)}, \delta\mathbf{z}^{(k)})$
    Set

$$\hat{\alpha}_x := \min_{1\le i\le n}\left\{-\frac{x^{(k)}{}_i}{\delta x^{(k)}{}_i} : \delta x^{(k)}{}_i < 0\right\}, \quad \hat{\alpha}_z := \min_{1\le i\le n}\left\{-\frac{z^{(k)}{}_i}{\delta z^{(k)}{}_i} : \delta z^{(k)}{}_i < 0\right\}$$

    Choose $\tau^{(k)} \in (0,1]$ and set

$$\alpha = \min\left\{1, \tau^{(k)}\hat{\alpha}_x, \tau^{(k)}\hat{\alpha}_z\right\}$$

    Set

$$\begin{aligned}\mathbf{x}^{(k+1)} &:= \mathbf{x}^{(k)} + \alpha\delta\mathbf{x}^{(k)}\\ \mathbf{y}^{(k+1)} &:= \mathbf{y}^{(k)} + \alpha\delta\mathbf{y}^{(k)}\\ \mathbf{z}^{(k+1)} &:= \mathbf{z}^{(k)} + \alpha\delta\mathbf{z}^{(k)}\end{aligned}$$

    Set $k := k + 1$
  until Inner-Convergence
  Generate $\rho^{(l+1)}$
  Set $l := l + 1$
until Outer-Convergence

been given by McCormick [1991], El-Bakry et al. [1996] and Yamashita and Yabe [1996]. The local convergence results of PDIP methods are inherited from the use of Newton's method to generate search directions. As with SQP methods (Section (1)), the drawbacks of Newton's method are also inherited. These drawbacks are analyzed in the next section. Global convergence results have been discussed by El-Bakry et al. [1996], Yamashita [1998] and Akrotirianakis and Rustem [2005].

We do not give further details on the methods and the aforementioned directions. Bertsekas [1995, sections 4.1 and 4.4] provides a detailed treatment of path-following interior point methods, and Wright [1997, chapters 4–5] also presents PDIP algorithms in an excellent manner. Wright [1992] has written an excellent monograph on interior point methods. Forsgren et al. [2002] have also written an excellent survey of interior point methods for nonlinear optimization.

3 Examples

Example 1 Solve the nonlinear programming problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = (x_2 - 2)^2 - x_1^2 \\ \text{subject to} & h(\mathbf{x}) = 4x_1^2 + x_2^2 - 1 = 0, \end{array}$$

starting from $\mathbf{x}^{(0)} = (2.0, 4.0)$ and $\lambda^{(0)} = 1/2$, using the Sequential Quadratic Programming method (Algorithm (1)).

Solution We have that

$$\begin{aligned} \boldsymbol{\nabla} f(\mathbf{x}) &= \begin{pmatrix} -2x_1 \\ 2(x_2 - 2) \end{pmatrix} \\ \boldsymbol{\nabla} h(\mathbf{x}) &= \begin{pmatrix} 8x_1 \\ 2x_2 \end{pmatrix} \\ \boldsymbol{\nabla}^2 \mathcal{L}(\mathbf{x}, \lambda) &= \begin{pmatrix} -2 - 8\lambda & 0 \\ 0 & 2 - 2\lambda \end{pmatrix} \end{aligned}$$

The quadratic programming subproblem at iteration k can be written as

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & \mathbf{d}^T \boldsymbol{\nabla} f(\mathbf{x}^{(k)}) + \frac{1}{2} \mathbf{d}^T \boldsymbol{\nabla}^2 \mathcal{L}(\mathbf{x}^{(k)}, \lambda^{(k)}) \mathbf{d} \\ \text{subject to} & \mathbf{d}^T \boldsymbol{\nabla} h(\mathbf{x}^{(k)}) + h(\mathbf{x}^{(k)}) = 0 \end{array}$$

or, in algebraic form

$$\begin{array}{ll}\underset{\mathbf{d}}{\text{minimize}} & -2 \cdot x_1^{(k)} \cdot d_1 + 2 \cdot (x_2^{(k)} - 2) \cdot d_2 - \frac{(2+8\cdot\lambda^{(k)})}{2} \cdot d_1^2 + \frac{(2-2\cdot\lambda^{(k)})}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot x_1^{(k)} \cdot d_1 + 2 \cdot x_2^{(k)} \cdot d_2 + 4 \cdot (x_1^{(k)})^2 + (x_2^{(k)})^2 - 1 = 0\end{array}$$

Step 0 The quadratic programming subproblem at $(\mathbf{x}^{(0)}, \lambda^{(0)})$ is

$$\begin{array}{ll}\underset{\mathbf{d}}{\text{minimize}} & -2 \cdot 2.0 \cdot d_1 + 2 \cdot (4.0 - 2) \cdot d_2 - \frac{(2+8\cdot(-0.5))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-0.5))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot 2.0 \cdot d_1 + 2 \cdot 4.0 \cdot d_2 + 4 \cdot (2.0)^2 + (4.0)^2 - 1 = 0\end{array}$$

the solution of which is $(d_1^{(0)}, d_2^{(0)}, \lambda^{(1)}) = (-0.804, -2.267, -0.350)$. The next iterate therefore is

$$\begin{aligned} x_1^{(1)} &= x_1^{(0)} + d_1^{(0)} = 2.0 - 0.804 = 1.196 \\ x_2^{(1)} &= x_2^{(0)} + d_2^{(0)} = 4.0 - 2.268 = 1.732 \end{aligned}$$

Step 1 The quadratic programming subproblem at $(\mathbf{x}^{(1)}, \lambda^{(1)})$ is

$$\begin{array}{ll}\underset{\mathbf{d}}{\text{minimize}} & -2 \cdot 1.196 \cdot d_1 + 2 \cdot (1.732 - 2) \cdot d_2 - \frac{(2+8\cdot(-0.350))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-0.350))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot 1.196 \cdot d_1 + 2 \cdot 1.732 \cdot d_2 + 4 \cdot (1.196)^2 + (1.732)^2 - 1 = 0\end{array}$$

the solution of which is $(d_1^{(1)}, d_2^{(1)}, \lambda^{(2)}) = (-0.734, -0.201, -0.312)$. The next iterate therefore is

$$\begin{aligned} x_1^{(2)} &= x_1^{(1)} + d_1^{(1)} = 1.196 - 0.734 = 0.462 \\ x_2^{(2)} &= x_2^{(1)} + d_2^{(1)} = 1.732 - 0.201 = 1.531 \end{aligned}$$

Step 2 The quadratic programming subproblem at $(\mathbf{x}^{(2)}, \lambda^{(2)})$ is

$$\begin{array}{ll}\underset{\mathbf{d}}{\text{minimize}} & -2 \cdot 0.462 \cdot d_1 + 2 \cdot (1.531 - 2) \cdot d_2 - \frac{(2+8\cdot(-0.312))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-0.312))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot 0.462 \cdot d_1 + 2 \cdot 1.531 \cdot d_2 + 4 \cdot (0.462)^2 + (1.531)^2 - 1 = 0\end{array}$$

the solution of which is $(d_1^{(2)}, d_2^{(2)}, \lambda^{(3)}) = (-0.575, -0.024, -0.327)$. The next iterate therefore is

$$\begin{aligned} x_1^{(3)} &= x_1^{(2)} + d_1^{(2)} = 0.462 - 0.575 = -0.113 \\ x_2^{(3)} &= x_2^{(2)} + d_2^{(2)} = 1.531 - 0.024 = 1.507 \end{aligned}$$

Step 3 The quadratic programming subproblem at $(\mathbf{x}^{(3)}, \lambda^{(3)})$ is

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -2 \cdot (-0.113) \cdot d_1 + 2 \cdot (1.507 - 2) \cdot d_2 - \frac{(2+8\cdot(-0.327))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-0.327))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot (-0.113) \cdot d_1 + 2 \cdot 1.507 \cdot d_2 + 4 \cdot ((-0.113))^2 + (1.507)^2 - 1 = 0 \end{array}$$

the solution of which is $(d_1^{(3)}, d_2^{(3)}, \lambda^{(4)}) = (0.490, -0.292, -0.584)$. The next iterate therefore is

$$\begin{aligned} x_1^{(4)} &= x_1^{(3)} + d_1^{(3)} = -0.113 + 0.490 = 0.377 \\ x_2^{(4)} &= x_2^{(3)} + d_2^{(3)} = 1.507 - 0.292 = 1.215 \end{aligned}$$

Step 4 The quadratic programming subproblem at $(\mathbf{x}^{(4)}, \lambda^{(4)})$ is

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -2 \cdot 0.377 \cdot d_1 + 2 \cdot (1.215 - 2) \cdot d_2 - \frac{(2+8\cdot(-0.584))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-0.584))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot 0.377 \cdot d_1 + 2 \cdot 1.215 \cdot d_2 + 4 \cdot (0.377)^2 + (1.215)^2 - 1 = 0 \end{array}$$

the solution of which is $(d_1^{(4)}, d_2^{(4)}, \lambda^{(5)}) = (-0.382, 0.044, -0.588)$. The next iterate therefore is

$$\begin{aligned} x_1^{(5)} &= x_1^{(4)} + d_1^{(4)} = 0.377 - 0.382 = -0.005 \\ x_2^{(5)} &= x_2^{(4)} + d_2^{(4)} = 1.215 + 0.044 = 1.259 \end{aligned}$$

Step 5 The quadratic programming subproblem at $(\mathbf{x}^{(5)}, \lambda^{(5)})$ is

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -2 \cdot (-0.005) \cdot d_1 + 2 \cdot (1.259 - 2) \cdot d_2 - \frac{(2+8\cdot(-0.588))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-0.588))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot (-0.005) \cdot d_1 + 2 \cdot 1.259 \cdot d_2 + 4 \cdot ((-0.005))^2 + (1.259)^2 - 1 = 0 \end{array}$$

the solution of which is $(d_1^{(5)}, d_2^{(5)}, \lambda^{(6)}) = (0.009, -0.232, -0.881)$. The next iterate therefore is

$$\begin{aligned} x_1^{(6)} &= x_1^{(5)} + d_1^{(5)} = -0.005 + 0.009 = 0.004 \\ x_2^{(6)} &= x_2^{(5)} + d_2^{(5)} = 1.259 - 0.232 = 1.027 \end{aligned}$$

Step 6 The quadratic programming subproblem at $(\mathbf{x}^{(6)}, \lambda^{(6)})$ is

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -2 \cdot 0.004 \cdot d_1 + 2 \cdot (1.027 - 2) \cdot d_2 - \frac{(2+8\cdot(-0.881))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-0.881))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot 0.004 \cdot d_1 + 2 \cdot 1.027 \cdot d_2 + 4 \cdot (0.004)^2 + (1.027)^2 - 1 = 0 \end{array}$$

the solution of which is $(d_1^{(6)}, d_2^{(6)}, \lambda^{(7)}) = (-0.005, -0.027, -0.996)$. The next iterate therefore is

$$\begin{aligned} x_1^{(7)} &= x_1^{(6)} + d_1^{(6)} = 0.004 - 0.005 = -0.001 \\ x_2^{(7)} &= x_2^{(6)} + d_2^{(6)} = 1.027 - 0.027 = 1.000 \end{aligned}$$

Step 7 The quadratic programming subproblem at $(\mathbf{x}^{(7)}, \lambda^{(7)})$ is

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -2 \cdot (-0.001) \cdot d_1 + 2 \cdot (1.0 - 2) \cdot d_2 - \frac{(2+8\cdot(-0.996))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-0.996))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot (-0.001) \cdot d_1 + 2 \cdot 1.0 \cdot d_2 + 4 \cdot ((-0.001))^2 + (1.0)^2 - 1 = 0 \end{array}$$

the solution of which is $(d_1^{(7)}, d_2^{(7)}, \lambda^{(8)}) = (5.0 \cdot 10^{-4}, 1.0 \cdot 10^{-6}, -1.0)$. The next iterate therefore is

$$\begin{aligned} x_1^{(8)} &= x_1^{(7)} + d_1^{(7)} = -0.001 + 5.0 \cdot 10^{-4} = -5.0 \cdot 10^{-4} \\ x_2^{(8)} &= x_2^{(7)} + d_2^{(7)} = 1.000 - 1.0 \cdot 10^{-6} = 1.0 \end{aligned}$$

Step 8 The quadratic programming subproblem at $(\mathbf{x}^{(8)}, \lambda^{(8)})$ is

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -2 \cdot (-5.0 \cdot 10^{-4}) \cdot d_1 + 2 \cdot (1.0 - 2) \cdot d_2 - \frac{(2+8\cdot(-1.0))}{2} \cdot d_1^2 + \frac{(2-2\cdot(-1.0))}{2} \cdot d_2^2 \\ \text{subject to} & 8 \cdot (-5.0 \cdot 10^{-4}) \cdot d_1 + 2 \cdot 1.0 \cdot d_2 + 4 \cdot ((-5.0 \cdot 10^{-4}))^2 + (1.0)^2 - 1 = 0 \end{array}$$

the solution of which is $(d_1^{(8)}, d_2^{(8)}, \lambda^{(9)}) = (0.5 \cdot 10^{-4}, 2.5 \cdot 10^{-7}, -1.0)$. The next iterate therefore is

$$\begin{aligned} x_1^{(9)} &= x_1^{(8)} + d_1^{(8)} = -5.0 \cdot 10^{-4} + 0.5 \cdot 10^{-4} = 0.0 \\ x_2^{(9)} &= x_2^{(8)} + d_2^{(8)} = 1.0 + 2.5 \cdot 10^{-7} = 1.0 \end{aligned}$$

We conclude that, after 9 iterations, the optimal solution of the problem is $(x_1^*, x_2^*, \lambda^*) = (0.0, 1.0, -1.0)$.

References

Ioannis Akrotirianakis and Berç Rustem. A globally convergent interior point algorithm for non-linear problems. *Journal of Optimization Theory and Applications*, 125(3):497–521, 2005.

Michael C. Bartholomew-Biggs. Constrained minimization using recursive equality quadratic programming. In F. A. Lootsma, editor, *Numerical methods for nonlinear optimization*, pages 411–428. Academic Press, London, 1972.

Dimitri P. Bertsekas. Penalty and multiplier methods. In *Nonlinear optimization (Proc. Internat. Summer School, Univ. Bergamo, Bergamo, 1979)*, pages 253–278. Birkhäuser Boston, Mass., 1980.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995. 2nd edition 1999.

Paul T. Boggs and Jon W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4:1–51, 1995.

Thomas F. Coleman and Peter A. Feynes. Partitioned quasi-Newton methods for nonlinear equality constrained optimization. *Mathematical Programming*, 53(1, Ser. A):17–44, 1992. ISSN 0025-5610.

Andrew R. Conn, Nicholas I. M. Gould, Dominique Orbarn, and Philippe L. Toint. A primal-dual trust-region algorithm for non-convex nonlinear programming. *Mathematical Programming*, 87:215–249, 2000a. URL `ftp://ftp.numerical.rl.ac.uk/pub/reports/cgotRAL99054.ps.gz`.

Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-region methods*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000b. ISBN 0-89871-460-5.

A. S. El-Bakry, Richard A. Tapia, T. Tsuchiya, and Yin Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Applications*, 89(3): 507–541, 1996. ISSN 0022-3239.

Roger Fletcher. An efficient globally convergent algorithm for unconstrained and linearly constrained optimisation problems. In *The seventh international symposium on mathematical programming*, The Hague, 1970.

Roger Fletcher. *Practical Methods of Optimization,* 2nd ed. John Wiley, New York, 1987.

Rodrigo Fontecilla. Local convergence of secant methods for nonlinear constrained optimization. *SIAM Journal on Numerical Analysis*, 25(3):692–712, June 1988. ISSN 0036-1429 (print), 1095-7170 (electronic).

Anders Forsgren, Philip E. Gill, and Margaret H. Wright. Interior methods for nonlinear optimization. *SIAM Rev.*, 44(4):525–597 (electronic) (2003), 2002. ISSN 0036-1445.

Masao Fukushima. A successive quadratic programming algorithm with global and superlinear convergence properties. *Mathematical Programming*, 35:253–264, 1986.

Ubaldo M. Garcia-Palomares. *Superlinearly Convergence Quasi-Newton Methods for Nonlinear Programming.* PhD thesis, University of Wisconsin, Madison, 1973.

Ubaldo M. Garcia-Palomares and Olvi L. Mangasarian. Superlinearly convergent quasi-Newton algorithms for nonlinearly constraint optimization problems. *Mathematical Programming*, 11:1–13, 1976.

S. T. Glad. Properties of updating methods for the multipliers in augmented Lagrangians. *Journal of Optimization Theory and Applications*, 28(2): 135–156, 1979. ISSN 0022-3239.

Nicholas I. M. Gould and Philippe L. Toint. SQP methods for large-scale nonlinear programming. Technical Report RAL-TR-1999-055, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 1999. URL `http://www.numerical.rl.ac.uk/reports/reports.html`.

Shih Ping Han. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11: 263–282, 1976.

Shih Ping Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:297–309, 1977.

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

Garth P. McCormick. The superlinear convergence of a nonlinear primal-dual algorithm. Technical Report T–550/91, School of Engineering and Applied Science, George Washington University, Washington, D.C., 1991.

N. Megiddo. Pathways to the optimal set in linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming, Interior Point and Related Methods*. Springer-Verlag, 1988.

Eliane R. Panier and André L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Mathematical Programming*, 59:261–276, 1993.

Michael J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G.A. Watson, editor, *Lecture notes in Mathematics, Numerical Analysis, Dundee 1977*, Dundee, Scotland, 1977. Springer-Verlag.

Michael J. D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In Olvi L. Mangasarian, R. R. Meyer, and Stephen M. Robinson, editors, *Nonlinear Programming 3*, pages 27–63. Academic Press, New York, NY, 1978a.

Michael J. D. Powell. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, 14, 1978b.

Klaus Schittkowski. On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function. *Mathematishe Operations Forschung und Statistik, Ser. Optimization*, 14:193–216, 1983.

G. Sonnevend. *An 'Analytic Center' for Polyhedrons and New Classes of Global Algorithms for Linear, Smooth, Convex Programming*, volume 84 of *Lecture Notes in Control Inform. Sci.*, pages 866–876. Springer-Verlag, Budapest, 1986.

R.B. Wilson. *A simplical method for convex programming*. PhD thesis, Harvard University, 1963.

Margaret H. Wright. Interior methods for constrained optimization. *Acta Numerica*, pages 341–407, 1992. URL `ftp://netlib.att.com/netlib/att/cs/doc/91/4-10.ps.Z`.

Stephen J. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997. ISBN 0-89871-382-X.

Hiroshi Yamashita. A globally convergent primal–dual interior point method for constrained optimization. *Optimization Methods and Software*, 10:443–469, 1998.

Hiroshi Yamashita and Hiroshi Yabe. Superlinear and quadratic convergence of some primal–dual interior point methods for constrained optimization. *Mathematical Programming*, 75:377–397, 1996.