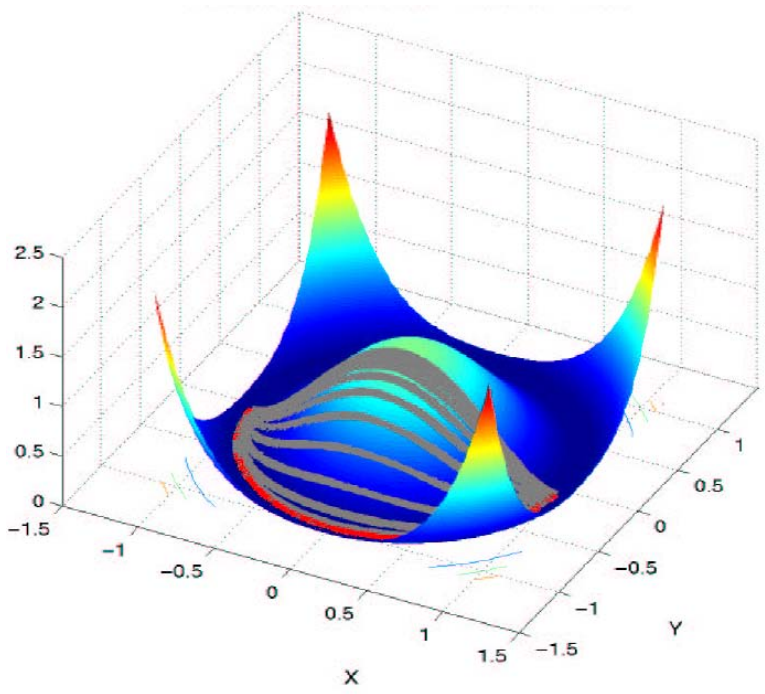
Introduction to Nonlinear Optimization

Paul J. Atzberger



Comments should be sent to:
atzberg@math.ucsb.edu

Introduction

We shall discuss in these notes a brief introduction to nonlinear optimization concepts, with wide applicability in finance and other applied fields. The basic problem is to solve

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1}$$

where typically $\mathbf{x} \in \mathbb{R}^n$, but this can also be subject to constraints.

To numerically approximate the solution (if it exists), we shall construct a sequence $\{\mathbf{x}_n\}_{n=1}^{\infty}$ such that $\mathbf{x}_n \to \mathbf{x}^*$, where $f(\mathbf{x}^*) = \min_{\mathbf{x}} f(\mathbf{x})$.

There are many algorithms which attempt to construct such a sequence, with different approaches appropriate depending on the problem. A good reference on nonlinear optimization methods is *Numerical Optimization* by Nocedal and Wright. In these notes, we shall discuss primarily line search methods and handle any constraints that arise using penalty terms.

In line search algorithms the sequence $\{\mathbf{x}_n\}_{n=1}^{\infty}$ is constructed iteratively at each step choosing a search direction $\mathbf{p}_k$ and attempting to minimize the objective function along the line (ray) in this direction. This reduces the problem essentially to a sequence of one dimensional problems, where $\mathbf{x}_k$ is given by the basic recurrence:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{2}$$

where the step-length $\alpha_k > 0$ must be estimated.

The key to having a sequence $\mathbf{x}_k$ which converges rapidly is to construct an algorithm which makes effective use of the information about $f(x)$ from the previous iterates to choose a good $\mathbf{p}_k$ and step-length α_k. To ensure that progress can be made each iteration, a natural condition to impose on $\mathbf{p}_k$ is that the function decrease, at least locally, in this direction. We call $\mathbf{p}_k$ a descent direction if $\nabla f \cdot \mathbf{p}_k < 0$. Typically, the descent direction has the form $\mathbf{p}_k = -B_k^{-1} \nabla f_k$. In the case that B_k is position definite this ensures that $\mathbf{p}_k$ is a descent direction

$$\nabla f_k \cdot \mathbf{p}_k \quad = \quad -\nabla f_k^T B_k^{-1} \nabla f_k < 0.$$

For example:

- Steepest descent has $B_k = \mathcal{I}$ set to the identity matrix so that $\mathbf{p}_k = -\nabla f_k$.

- Newton's method has $B_k = \nabla^2 f_k$ set to the Hessian so that $\mathbf{p}_k = -\left(\nabla^2 f_k\right)^{-1} \nabla f_k$. It is important to note that $\mathbf{p}_k$ is only ensured to be descent direction if the Hessian is position definite. The Hessian however is often expensive to compute numerically.

- Quasi-Newton methods construct a B_k which approximates the Hessian by making use of the previous function evaluations of f and ∇f, see Nocedal and Wright for a further discussion.

In designing a good numerical optimization method some care must be taken in choosing the not only the direction $\mathbf{p}_k$ but the step-length α_k, and this will usually depend on the function $f(\mathbf{x})$ being optimized. We shall assume that $\mathbf{p}_k$ is some chosen descent direction and now discuss how to choose the step-length α_k.

Wolfe Conditions: Choosing a Step-Length α

Let $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k)$ then one choice of α_k which might seem ideal would be to minimize $\phi(\alpha)$ globally for all α. However, it is typically computationally expensive to find the global optimum even to the one dimensional problem. Since the ray $\mathbf{p}_k$ will not typically contain the minimizer to the full problem, multiple

iterations will almost always be required. This suggests that in designing an algorithm there should be a balance between making progress in minimizing the objective function f in each search direction $\mathbf{p}_k$ while iterating over a number of different search directions.

To ensure our algorithm converges we must be more mathematically precise about what we mean by making progress over each search direction. The simplest criteria we might consider to determine if progress has been made over a particular search direction is that the function be reduced $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < f(\mathbf{x}_k)$, however, this is not sufficient to ensure convergence to the minimizer of $\mathbf{x}^*$.

For example, consider $f(x, y) = x^2 + y^2$, and suppose the algorithm has the output $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ where $\alpha_k = \frac{1}{2^k}$. Then if $\mathbf{p}_k = [-1, 0]^T$ we have $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$, but $\mathbf{x}_{k+1} = \mathbf{x}_0 - \sum_{k=0}^{\infty} \frac{1}{2^k} \mathbf{p}_k$. If the first component is $[\mathbf{x}_0]_1 > 2$ then $\mathbf{x}_k \not\to 0$. In other words, to summarize, if the progress made each iteration is not sufficiently large, the sequence may converge prematurely to a non-optimal value.

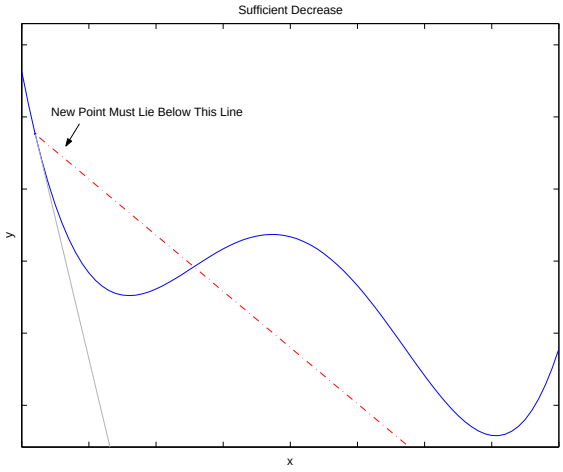


Figure 1: Sufficient Decrease Condition

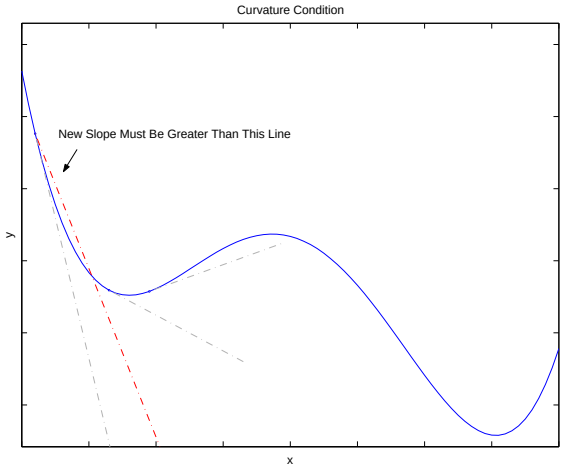


Figure 2: Curvature Condition

To ensure that "sufficient progress" is made each iteration in the line search we shall require the following two conditions to hold:

Wolfe Conditions:

(i) (sufficient decrease) $f(\mathbf{x}_k + \alpha \mathbf{p}_k) < f(\mathbf{x}_k) + c_1 \alpha \nabla f_k^T \mathbf{p}_k$, where $c_1 \in (0, 1)$.

(ii) (curvature condition) $\nabla f(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}_k \geq c_2 \nabla f_k^T \mathbf{p}_k$, where $c_2 \in (c_1, 1)$.

(See the figures 1 – 2 for a geometric interpretation).

We now prove that such step lengths exist satisfying the Wolfe Conditions.

Convergence of Line Search Optimization Methods

Lemma: Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and let $\mathbf{p}_k$ be a descent direction for each $\mathbf{x}_k$. If we assume that along each ray $\{\mathbf{x}_k + \alpha \mathbf{p}_k | \alpha > 0\}$ f is bounded below then there exist intervals of step lengths $[\alpha^{(1)}, \alpha^{(2)}]$ such that the Wolfe Conditions are satisfied.

Proof: Since $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k)$ is bounded below for all $\alpha > 0$ and $0 < c_1 < 1$, the line $\ell(\alpha) = f(\mathbf{x}_k) + \alpha c_1 \nabla f_k^T \mathbf{p}_k$ must intersect the graph of $\phi(\alpha)$ at least once. Let $\alpha' > 0$ be the smallest such α, that is $f(\mathbf{x}_k + \alpha' \mathbf{p}_k) = f(\mathbf{x}_k) + \alpha' c_1 \nabla f_k^T \mathbf{p}_k$. The sufficient decrease condition (i) then clearly holds for $\alpha < \alpha'$.

By the differentiability of the function we have from the mean-value theorem that there exists $\alpha'' \in (0, \alpha')$ such that $f(\mathbf{x}_k + \alpha' \mathbf{p}_k) - f(\mathbf{x}_k) = \alpha' \nabla f(\mathbf{x}_k + \alpha'' \mathbf{p}_k)^T \mathbf{p}_k$. Now by substituting for $f(\mathbf{x}_k + \alpha' \mathbf{p}_k)$ the expression above (sufficient-decrease), we obtain $\nabla f(\mathbf{x}_k + \alpha'' \mathbf{p}_k)^T \mathbf{p}_k = c_1 \nabla f_k^T \mathbf{p}_k > c_2 \nabla f_k^T \mathbf{p}_k$, since $c_1 < c_2$ and $\nabla f_k \mathbf{p}_k < 0$. Thus α'' satisfies the curvature condition (ii).

Therefore, since $\alpha'' < \alpha'$, we have that in a neighborhood of α'' both Wolfe Conditions (i) and (ii) hold simultaneously ■.

We now discuss the convergence of line search algorithms when the Wolfe Conditions are satisfied.

Theorem: (Zoutendijk's Condition) Assume $\{\mathbf{x}_k\}$ is generated by a line search algorithm satisfying the Wolfe Conditions. Suppose that $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is bounded below with Lipschitz continuous ∇f,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$$

then

$$\sum_{k \geq 0} \cos^2(\theta_k) \|\nabla f_k\|^2 < \infty$$

where

$$\cos(\theta_k) = \frac{-\nabla f_k^T \mathbf{p}_k}{\|\nabla f_k\| \|\mathbf{p}_k\|}.$$

We remark that θ_k is the angle between the $-\nabla f_k^T$ steepest descent direction and the search direction $\mathbf{p}_k$.
Proof: From the Wolfe Conditions (i) and (ii) we have

$$\left(\nabla f_{k+1} - \nabla f_k\right)^T \mathbf{p}_k \geq \left(c_2 - 1\right) \nabla f_k^T \mathbf{p}_k.$$

From Lipschitz continuity we have

$$\left(\nabla f_{k+1} - \nabla f_k\right)^T \mathbf{p}_k \leq \alpha_k L \|\mathbf{p}_k\|^2.$$

By combining both relations we have

$$\alpha_k \geq \frac{c_2 - 1}{L} \frac{\nabla f_k^T \mathbf{p}_k}{\|\mathbf{p}_k\|^2}.$$

By substituting this in (i) we have

$$f_{k+1} \leq f_k - c_1 \frac{1 - c_2}{L} \frac{\left(\nabla f_k^T \mathbf{p}_k\right)^2}{\|\mathbf{p}_k\|^2}.$$

From the definition of $\cos(\theta_k)$ we have

$$f_{k+1} \leq f_k - c \cos(\theta_k) \|\nabla f_k\|^2$$

where $c = \frac{c_1(1-c_2)}{L}$. By summing all indices less than k we have $f_{k+1} \leq f_0 - c \sum_{j=0}^{k} \cos^2(\theta_j) \|\nabla f_j\|^2$. Since $f(x)$ is bounded below we must have

$$\sum_{j=0}^{\infty} \cos^2(\theta_j) \|\nabla f_j\|^2 < \infty$$

■.

Using the standard facts about convergence of series, we have $\cos^2(\theta_j)\|\nabla f_j\|^2 \to 0$, as $k \to \infty$. This result while appearing somewhat technical, is actually quite useful. For if we construct a line search method such that $\cos(\theta_k) > \delta > 0$, then the theorem above implies that

$$\lim_{k \to \infty} \|\nabla f_k\| = 0.$$

Thus if the line search sequence $\{\mathbf{x}_k\}$ remains bounded there will be a limit point $\mathbf{x}^*$ which is a critical point such that $\nabla f(\mathbf{x}^*) = 0$. Given the sufficient decrease condition this point will likely be a minimizer, although one must be careful to check if no special structure is assumed about f.

To further describe the convergence, we shall say that a method converges in f with rate p if

$$|f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)| \leq c |f(\mathbf{x}_k) - f(\mathbf{x}^*)|^p.$$

We say that a method converges in $\mathbf{x}$ with rate p if

$$|\mathbf{x}_{k+1} - \mathbf{x}^*| \leq c |\mathbf{x}_k - \mathbf{x}^*|^p.$$

We shall now discuss the convergence of two common optimization methods, namely, steepest descent and newton's method.

Convergence of Steepest Descent

Steepest Descent: In this case $\mathbf{p}_k = -\nabla f_k$ for each step. Let us consider a problem which we can readily analyze with an objective function of the form:

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T Q \mathbf{x} - \mathbf{b}^T \mathbf{x}$$

where $Q = Q^T$ is positive definite. The minimizer to this problem can be readily found as solves, $Q\mathbf{x}^* = \mathbf{b}$.

In this example, we can explicitly compute the step-length α which minimizes the function $\rho(\alpha)$ each interaction. This is given by

$$f(\mathbf{x}_k - \alpha \mathbf{g}_k) = \frac{1}{2} \left(\mathbf{x}_k - \alpha \mathbf{g}_k\right)^T Q \left(\mathbf{x}_k - \alpha \mathbf{g}_k\right) - \mathbf{b} \left(\mathbf{x}_k - \alpha \mathbf{g}_k\right)$$

so that $\nabla f(x) = Q\mathbf{x} - \mathbf{b}$ gives the minimizer in $\alpha > 0$. That is α should satisfy

$$\nabla f(\mathbf{x}_k - \alpha \mathbf{g}_k) = Q\left(\mathbf{x}_k - \alpha \mathbf{g}_k\right) - \mathbf{b} = 0.$$

This gives

$$\alpha = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T Q \mathbf{g}_k}$$

when $\mathbf{g}_k = -\nabla f_k$ we have

$$\alpha_k = \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k}$$

thus the next iterate is given by (when minimizing in the line search),

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k}\right) \nabla f_k.$$

From the expression above it can be shown that

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) = \left[1 - \frac{\left(\nabla f_k^T \nabla f_k\right)^2}{\left(\nabla f_k^T Q \nabla f_k\right)\left(\nabla f_k^T Q^{-1} \nabla f_k\right)}\right] \left(f(\mathbf{x}_k) - f(\mathbf{x}^*)\right).$$

This can be expressed in terms of the eigenvalues of Q as

$$\left(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)\right) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \left(f(\mathbf{x}_k) - f(\mathbf{x}^*)\right)$$

where $0 \leq \lambda_1 \leq \lambda_2 \leq \cdot \leq \lambda_n$ are eigenvalues of Q. (Proof is given elsewhere, see Luenberger 1984).

We remark that for nonlinear objective functions we can approximate the function near a minimizer using a truncation of the Taylor expansion at the quadratic term. In this case, the analysis above will be applicable up to this truncation error where we set Q to be the Hessian, $Q = \nabla^2 f$.

From the analysis we see find that steepest descent has a rate of convergence in f of first order, in the ideal case.

Convergence of Newton's Method

Newton's Method:

Let us now consider the the case when the search direction is

$$\mathbf{p}_k^{(N)} = -\nabla^2 f_k^{-1} \nabla f_k.$$

As we mentioned above, we must be a little careful since in this case the Hessian $\nabla^2 f_k$ may not be positive definite and the search direction may not be a descent direction. For the example above Q positive definite the search direction is

$$\mathbf{p}_k^{(N)} = -Q^{-1} \nabla f_k.$$

Analysis similar to that done in the case of Steepest Descent can be carried out to show that the rate of convergence in f is quadratic. In particular, in the case of a quadratic objective function it can be shown that for each iteration the optimal $\alpha_k = 1$. This gives

$$\begin{aligned} \mathbf{x}_k + \mathbf{p}_k - \mathbf{x}^* &= \mathbf{x}_k - \mathbf{x}^* - \nabla^2 f_k^1 \nabla f_k \\ &= \nabla^2 f_k^{-1} \left[\nabla^2 f_k (\mathbf{x}_k - \mathbf{x}^*) - (\nabla f_k - \nabla f_*)\right] \end{aligned}$$

where $\nabla f_* = \nabla f(\mathbf{x}^*) = 0$.

Now using that

$$\nabla f_k - \nabla f_* = \int_0^1 \nabla^2 f(\mathbf{x}_k + t(\mathbf{x}^* - \mathbf{x}_k))(\mathbf{x}_k - \mathbf{x}^*)dt,$$

we have

$$\begin{aligned}\|\nabla^2 f(\mathbf{x}_k)(\mathbf{x}_k - \mathbf{x}^*) - (\nabla f_k - \nabla f(\mathbf{x}^*))\| &= \| \int_0^1 \left[\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k + t(\mathbf{x}^* - \mathbf{x}_k))\right](\mathbf{x}_k - \mathbf{x}^*)dt\| \\ &\leq \int_0^1 \|\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k + t(\mathbf{x}^* - \mathbf{x}_k))\|\|\mathbf{x}_k - \mathbf{x}^*\|dt \\ &\leq \|\mathbf{x}_k - \mathbf{x}^*\|^2 \int_0^1 Ltdt\end{aligned}$$

where L is the Lipschitz constant for $\nabla^2 f(\mathbf{x})$ for $\mathbf{x}$ near $\mathbf{x}^*$.

$$\|\mathbf{x}_k + \mathbf{p}_k^{(N)} - \mathbf{x}^*\| \leq L\|\nabla^2 f(\mathbf{x}^*)^{-1}\|\|\mathbf{x}_k - \mathbf{x}^*\|^2$$

Therefore, the method converges in $\mathbf{x}$ quadratically, $p = 2$.

A Line Search Algorithm

We now discuss an algorithm to find a step-length α which satisfies the Wolfe conditions. The basic strategy is to use interpolation and a bisection search by determining which half of an interval contains points satisfying the Wolfe conditions. We state a general algorithm here in pseudo code.

Algorithm: LineSearch (Input: $\mathbf{x}_k$, $\mathbf{p}_k$, $\alpha_{\max}$. Output α_*.)

$\alpha_0 \leftarrow 0$, and specify $\alpha_1 > 0$ and $\alpha_{\max}$;
$i \leftarrow 1$
repeat:
  Evaluate $\phi(\alpha_i)$;
  if $\phi(\alpha_i) > \phi(0) + c_1\alpha_i\phi'(0)$ or $\phi(\alpha_i) \geq \phi(\alpha_{i-1})$ and $i > 1$.
  $\alpha_* \leftarrow$ zoom (α_{i-1}, α_i) and stop.
  Evaluate $\phi'(\alpha_i)$;
  if $|\phi'(\alpha_i)| \leq -c_2\phi'(0)$
    set $\alpha_* \leftarrow \alpha_i$ and stop;
  if $\phi'(\alpha_i) \geq 0$
    set $\alpha_* \leftarrow$ zoom(α_i, α_{i-1}) and stop;
  Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{\max})$
  $i \leftarrow i + 1$;
end (repeat)

Algorithm: Zoom (Input: α_{lo}, α_{hi}. Output: α_*.)

repeat
  Interpolate (using quadratic, cubic, or bisection) to find a trial step length α_j between α_{lo} and α_{hi}
  Evaluate $\phi(\alpha_j)$;
  if $\phi(\alpha_j) > \phi(0) + c_1\alpha_j\phi'(0)$ or $\phi(\alpha_j) \geq \phi(\alpha_{\text{lo}})$
    $\alpha_{\text{hi}} \leftarrow \alpha_j$;

else
Evaluate $\phi'(\alpha_j)$;
if $|\phi(\alpha_j)| \leq -c_2\phi'(0)$
set $\alpha_* \leftarrow \alpha_j$ and stop;
if $\phi'(\alpha_j)(\alpha_{\text{hi}} - \alpha_{\text{lo}}) \geq 0$
$\alpha_{\text{hi}} \leftarrow \alpha_{\text{lo}}$;
$\alpha_{\text{lo}} \leftarrow \alpha_j$;
end (repeat)

For a more in depth discussion and motivation for these algorithms see (Nocedal and Wright, Numerical Optimization, pages 58 - 61).

Solving Nonlinear Unconstrained Optimization Problems

For unconstrained optimization problems of the form

$$\min_{\mathbf{x}} f(\mathbf{x})$$

the basic line search algorithm is as follows:

Unconstrained Line Search Optimization Algorithm: (Input: $\mathbf{x}_0$, $\alpha_{\max}, \epsilon$. Output: $\mathbf{x}^*$.)

$k \leftarrow 0$
$\mathbf{x}_k \leftarrow \mathbf{x}_0$
Evaluate $\nabla f_k \leftarrow \nabla f(\mathbf{x}_k)$;
repeat
$\mathbf{p}_k = -B_k^{-1}\nabla f_k$;
$\alpha_k \leftarrow \text{LineSearch}(\mathbf{x}_k, \mathbf{p}_k, \alpha_{\max})$;
$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$;
Evaluate $\nabla f_k \leftarrow \nabla f(\mathbf{x}_k)$;
If $\|\nabla f_k\| < \epsilon$ then
$\mathbf{x}^* \leftarrow \mathbf{x}_k$ and stop;
end (repeat)

Solving Nonlinear Constrained Optimization Problems

For constrained optimization problems of the form

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject} \quad & \\ & c_i(\mathbf{x}) = 0, \ i \in \mathcal{E} \\ & c_i(\mathbf{x}) \geq 0, \ i \in \mathcal{I} \end{aligned}$$

a line search algorithm can be constructed.

To handle the constraints we shall formulate an unconstrained optimization problem which incorporates the constraints through penalty terms. The strategy is to then solve each of the unconstrained problems with a penalty parameter μ_j up to a tolerance of ϵ_j. By repeatedly solving this problem using as the starting point the solution $\mathbf{x}_j^*$ of the previous iteration, and by successively reducing the values of μ_j and ϵ_j, a sequence of solutions $\mathbf{x}_j \rightarrow \mathbf{x}^*$ can be generated. One must take some care on how μ_j and ϵ_j are reduced each iteration to ensure convergence and efficient use of computational resources. These issues are further discussed in

Nocedal and Wright. The basic algorithm is:

A Constrained Line Search Optimization Algorithm: (Input: $\mathbf{x}_0$, $\alpha_{\max}, \epsilon_*, \delta_*$. Output: $\mathbf{x}^*$.)

$j \leftarrow 0$
$\mathbf{x}_j^* \leftarrow \mathbf{x}_0$
$\mu_j \leftarrow \mu_0$
repeat 1
    Let $F(\mathbf{x}, \mu_j) = f(\mathbf{x}) + \frac{1}{2\mu_j} \sum_{i \in \mathcal{E}} c_i^2(\mathbf{x}) - \mu_j \sum_{i \in \mathcal{I}} \log(c_i(\mathbf{x}))$
    $\mathbf{x}_j^* \leftarrow$ UnconstrainedOptimization$(\mathbf{x}_j^*, \alpha_{\max}, \epsilon_j)$
    Evaluate $\nabla F_j \leftarrow \nabla_{\mathbf{x}} F(\mathbf{x}_j^*, \mu_j)$;
    if $\|\nabla F_j\| < \epsilon_*$ and $\mu_j < \delta_*$ then $\mathbf{x}^* \leftarrow \mathbf{x}_j^*$ and stop;
    $j \leftarrow j + 1$
    reduce the value of μ_j
    reduce the value of ϵ_j
end (repeat 1)