Numerical Methods I: Nonlinear equations and nonlinear least squares

Georg Stadler
Courant Institute, NYU
stadler@cims.nyu.edu

Oct 15 & 22, 2015
Nonlinear continuous optimization
Main reference:
Section 7.2/7.3 in Quarteroni/Sacci/Saleri, or Section 2 (and parts of 3) in: Nocedal/Wright: *Numerical Optimization*, Springer 2006.
Optimization problems

Different optimization problems:

\[
\min_{x \in \mathbb{R}^n} f(x)
\]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \). Often, one additionally encounters constraints of the form

\[
\begin{align*}
  g(x) &= 0 & \text{(equality constraints)} \\
  h(x) &\geq 0 & \text{(inequality constraints)}
\end{align*}
\]

- Often used: “programming” \( \equiv \) optimization
- continuous optimization \((x \in \mathbb{R}^n)\) versus discrete optimization (e.g., \(x \in \mathbb{Z}^n\))
- nonsmooth (e.g., \(f\) is not differentiable) versus smooth optimization (we assume \(f \in C^2\))
- convex optimization vs. nonconvex optimization (convexity of \(f\))
Continuous unconstrained optimization

Assumptions

We assume that $f(\cdot) \in C^2$, and assume unconstrained minimization problems, i.e.:

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

A point $x^*$ is a **global solution** if

$$
 f(x) \geq f(x^*) \quad (1)
$$

for all $x \in \mathbb{R}^n$, and a **local solution** if (1) for all $x$ in a neighborhood of $x^*$. 
Continuous unconstrained optimization

Necessary conditions

At a local minimum $x^*$ holds the first-order necessary condition

$$\mathbb{R}^n \ni \nabla f(x^*) = 0$$

and the second-order (necessary) sufficient condition

$$\mathbb{R}^{n \times n} \ni \nabla^2 f(x^*) \text{ is positive (semi-) definite.}$$

To find a candidate for a minimum, we can thus solve the nonlinear equation for a stationary point:

$$G(x) := \nabla f(x) = 0,$$

for instance with Newton’s method. Note that the Jacobian of $G(x)$ is $\nabla^2(x)$. 
Descent algorithm

In optimization, one often prefers iterative descent algorithms that take into account the optimization structure.

**Basic (crude) descent algorithm:**

1. Initialize starting point \( x^0 \), set \( k = 1 \).
2. For \( k = 0, 1, 2, \ldots \), find a descent direction \( d^k \).
3. Find a step length \( \alpha_k > 0 \) for the update

\[
x^{k+1} := x^k + \alpha_k d^k
\]

such that \( f(x^{k+1}) < f(x^k) \). Set \( k := k + 1 \) and repeat.
Descent algorithm

Idea: Instead of solving an $n$-dim. minimization problem, (approximately) solve a sequence of 1-dim. problems:

- **Initialization**: As close as possible to $x^\ast$.
- **Descent direction**: Direction in which function decreases locally.
- **Step length**: Want to make large, but not too large steps.
- **Check for descent**: Make sure you make progress towards a (local) minimum.
Descent algorithm

**Initialization:** Ideally close to the minimizer. Solution depends, in general, on linearization (in the presence of multiple local minima).
Descent algorithm

Directions, in which the function decreases (locally) are called descent directions.

- **Steepest descent direction:**
  \[ d^k = -\nabla f(x^k) \]

- When \( B_k \in \mathbb{R}^{n \times n} \) is positive definite, then
  \[ d^k = -B_k^{-1}\nabla f(x^k) \]
  is the quasi-Newton descent direction.

- When \( H_k = H(x^k) = \nabla^2 f(x^k) \) is positive definite, then
  \[ d^k = -H_k^{-1}\nabla f(x^k) \]
  is the Newton descent direction. At a local minimum, \( H(x^*) \) is positive (semi)definite.
Idea behind Newton’s method in optimization: Instead of finding minimum of $f$, find **minimum of quadratic approximation of $f$** around current point:

$$q_k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d$$

Minimum is (provided $\nabla^2 f(x^k)$ is spd):

$$d = -\nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

is the Newton search direction. Since this is the minimum of the quadratic approximation, $\alpha_k = 1$ is the “optimal” step length.
Descent algorithm

**Step length:** Need to choose step length $\alpha_k > 0$ in

$$x^{k+1} := x^k + \alpha_k d^k$$

Ideally: Find minimum $\alpha$ of 1-dim. problem

$$\min_{\alpha > 0} f(x^k + \alpha d^k).$$

It is not necessary to find the exact minimum.
Descent algorithm

Step length (continued): Find $\alpha_k$ that satisfies the Armijo condition:

$$f(x^k + \alpha_k d^k) \leq f(x^k) + c_1 \alpha_k \nabla f(x^k)^T d^k,$$  

(2)

where $c_1 \in (0, 1)$ (usually chosen rather small, e.g., $c_1 = 10^{-4}$).

Additionally, one often uses the gradient condition

$$\nabla f(x^k + \alpha_k d^k)^T d^k \geq c_2 \nabla f(x^k)^T d^k$$

(3)

with $c_2 \in (c_1, 1)$.

The two conditions (2) and (3) are called Wolfe conditions.
Descent algorithm
Convergence of line search methods

Denote the angle between \( d_k \) and \( -\nabla f(x^k) \) by \( \Theta_k \):

\[
\cos(\Theta_k) = \frac{-\nabla f(x^k)^T d_k}{\|\nabla f(x^k)\| \|d_k\|}.
\]

Assumptions on \( f : \mathbb{R}^n \to \mathbb{R} \): continuously differentiable, derivative is Lipschitz-continuous, \( f \) is bounded from below.

Method: descent algorithm with Wolfe-conditions.
Then:

\[
\sum_{k \geq 0} \cos^2(\Theta_k) \|\nabla f(x^k)\|^2 < \infty.
\]
Descent algorithm
Convergence of line search methods

Denote the angle between $d^k$ and $-\nabla f(x^k)$ by $\Theta_k$:

$$\cos(\Theta_k) = \frac{-\nabla f(x^k)^T d^k}{\|\nabla f(x^k)\| \|d^k\|}.$$ 

Assumptions on $f : \mathbb{R}^n \to \mathbb{R}$: continuously differentiable, derivative is Lipschitz-continuous, $f$ is bounded from below.

Method: descent algorithm with Wolfe-conditions.

Then:

$$\sum_{k \geq 0} \cos^2(\Theta_k) \|\nabla f(x^k)\|^2 < \infty.$$ 

In particular: If $\cos(\Theta_k) \geq \delta > 0$, then $\lim_{k \to \infty} \|\nabla f(x^k)\| = 0$.,
Descent algorithm

Alternative to Wolfe step length: Find $\alpha_k$ that satisfies the Armijo condition:

$$f(x^k + \alpha_k d^k) \leq f(x^k) + c_1 \alpha_k \nabla f(x^k)^T d^k,$$

where $c_1 \in (0, 1)$. 
Descent algorithm

Alternative to Wolfe step length: Find $\alpha_k$ that satisfies the Armijo condition:

$$f(x^k + \alpha_k d^k) \leq f(x^k) + c_1 \alpha_k \nabla f(x^k)^T d^k,$$

where $c_1 \in (0, 1)$.

Use backtracking linesearch to find a step length that is large enough:

- Start with (large) step length $\alpha_k^0 > 0$.
- If it satisfies (4), accept the step length.
- Else, compute $\alpha_k^{i+1} := \rho \alpha_k^i$ with $\rho < 1$ (usually, $\rho = 0.5$) and go back to previous step.

This also leads to a globally converging method to a stationary point.
Let us consider a simple case, where $f$ is quadratic:

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

where $Q$ is spd. The gradient is $\nabla f(x) = Q x - b$, and minimizer $x^*$ is solution to $Q x = b$. Using exact line search, the convergence is:

$$\|x^{k+1} - x^*\|_Q^2 \leq \frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{max}} + \lambda_{\text{min}}} \|x^k - x^*\|_Q^2$$

(linear convergence with rate depending on eigenvalues of $Q$)
Descent algorithms

Convergence of steepest descent
Descent algorithms

Convergence of steepest descent
Newton’s method: Assumptions on f: 2× differentiable with Lipschitz-continuous Hessian $\nabla^2 f(x^k)$. Hessian is positive definite in a neighborhood around solution $x^*$. Assumptions on starting point: $x^0$ sufficient close to $x^*$. Then: Quadratic convergence of Newton’s method with $\alpha_k = 1$, and $\|\nabla f(x^k)\| \to 0$ quadratically.
Newton’s method: Assumptions on \( f \): \( 2 \times \) differentiable with Lipschitz-continuous Hessian \( \nabla^2 f(x^k) \). Hessian is positive definite in a neighborhood around solution \( x^* \).

Assumptions on starting point: \( x^0 \) sufficient close to \( x^* \).

Then: Quadratic convergence of Newton’s method with \( \alpha_k = 1 \), and \( \| \nabla f(x^k) \| \to 0 \) quadratically.

Equivalent to Newton’s method for solving \( \nabla f(x) = 0 \), if Hessian is positive.

How many iterations does Newton need for quadratic problems?
Summary of Newton methods and variants

- Newton to solve nonlinear equation $F(x) = 0$.
- Newton to solve optimization problem is equivalent to solving for the stationary point $\nabla f(x) = 0$, provided Hessian is positive and full steps are used (compare also convergence result).
- Optimization perspective to solve $\nabla f(x)$ provided additional information.
- Gauss-Newton method for nonlinear least squares problem is a specific quasi-Newton method.
Constrained optimization

Equality constraints

\[
\min_{x \in \mathbb{R}^n} f(x)
\]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \).

\[
g(x) = 0 \quad \text{(equality constraints)}
\]

Lagrangian function:

\[
\mathcal{L}(x, \lambda) = f(x) - \lambda g(x),
\]

where \( \lambda \) is a Lagrange multiplier.
At the solution $x^*$ of the constrained minimization problem, there exists $\lambda^*$ such that

$$\mathcal{L}_x(x^*, \lambda^*) = 0, \quad \mathcal{L}_\lambda(x^*, \lambda^*) = 0$$

Example: $f(x_1, x_2) = x_1 + x_2$, $g(x_1, x_2) = x_1^2 + x_2^2 - 2 = 0$. Note that we are not minimizing $\mathcal{L}(x, \lambda)$—this is in general a saddle point.
Constrained optimization

Inequality constraints

\[ \min_{x \in \mathbb{R}^n} f(x) \]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \).

\[ h(x) \geq 0 \]  \hspace{2cm} \text{(inequality constraints)}

At the solution \( x^* \) of the constrained minimization problem, there exists \( \lambda^* \geq 0 \) such that

\[ \mathcal{L}_x(x^*, \lambda^*) = 0, \text{ and } \lambda^* h(x^*) = 0. \]

These conditions can be challenging (\( \lambda \) is unknown too!)—they are also called complementarity conditions.

Example: \( f(x_1, x_2) = x_1 + x_2, \ g(x_1, x_2) = x_1^2 + x_2^2 - 2 \geq 0. \)
Constrained optimization

- **Optimality conditions** for equality or inequality constrained problems can be solved, for instance using Newton’s method.
- Many alternative methods exist.
- Equality constrained problems are usually easier.
- For linear equality constraints, one can parametrize the constraint manifold.

For more, see e.g., Nocedal/Wright’s book on Numerical Optimization.