TMA947 / MMG621 — Nonlinear optimization

Lecture 5 — Uncontrained optimization algorithms

Emil Gustavsson, Zuzana Nedělková

November 6, 2017

[Minor revision: Axel Ringh - August, 2023]

Consider the unconstrained optimization problem to

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\ f(\boldsymbol{x}), \tag{1}$$

where $f \in C^0$ on $\mathbb{R}^n$ (f is continuous). Mostly, we assume that $f \in C^1$ holds (f is continuously differentiable), sometimes even C^2. The choice of the algorithm depends on the size of the problem, availability of $\nabla f(\boldsymbol{x})$ and $\nabla^2 f(\boldsymbol{x})$, convexity of f and if the goal is to find a local or the global minimum.

Most algorithms for unconstrained optimization problems are what we call *line search type algorithms*.

Definition. *Line search type algorithm*

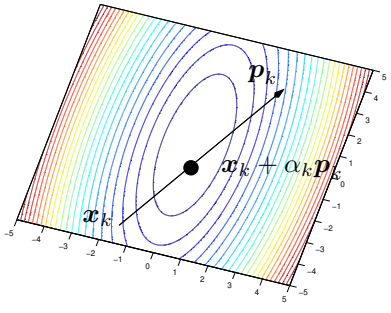
Step 0: **Starting point** $\boldsymbol{x}_0 \in \mathbb{R}^n$. *Let* $k := 0$.

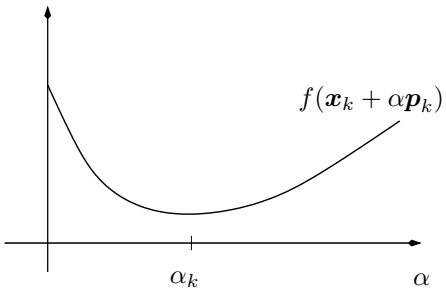
Step 1: Find **search direction** $\boldsymbol{p}_k \in \mathbb{R}^n$

Step 2: Perform **line search**, *i.e., find* $\alpha_k > 0$ *such that* $f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) < f(\boldsymbol{x}_k)$

Step 3: Let $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$.

Step 4: If **termination criteria** *is fulfilled then stop! Otherwise, let* $k := k + 1$ *and go to Step 1.*





Most algorithms we consider are inherently *local*, meaning that the search direction $\boldsymbol{p}_k$ is only based on the information at the current point $\boldsymbol{x}_k$, that is, $f(\boldsymbol{x}_k)$, $\nabla f(\boldsymbol{x}_k)$, and $\nabla^2 f(\boldsymbol{x}_k)$.

Think of a near-sighted mountain climber. The climber is in a deep fog and can only check his or her barometer for the height and feel the steepness of the slope under her feet.

Step 1: Search directions

Vector $\boldsymbol{p}_k$ is a descent direction at $\boldsymbol{x}_k$ if $f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) < f(\boldsymbol{x}_k)$ for all $\alpha \in (0, \delta]$ for some $\delta > 0$.

Let $f \in C^1$ in some neighborhood of $\mathbf{x}_k \in \mathbb{R}^n$, if $\nabla f(\boldsymbol{x}_k) \neq \mathbf{0}$, then $\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)$ is a descent direction for f at $\boldsymbol{x}_k$ (follows from optimality conditions). This search step is called *steepest descent direction* because it solves the problem to

$$\underset{\boldsymbol{p} \in \mathbb{R}^n : \|\boldsymbol{p}\| = 1}{\text{minimize}} \ \nabla f(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{p}.$$

Let $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ be an arbitrary symmetric, positive definite matrix. Then $\boldsymbol{p}_k = -\boldsymbol{Q} \nabla f(\boldsymbol{x}_k)$ is a descent direction for f at $\boldsymbol{x}_k$, because

$$\nabla f(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{Q} \nabla f(\boldsymbol{x}_k) < 0,$$

due to the positive definiteness of $\boldsymbol{Q}$.

Examples:

- *Steepest descent:* $\boldsymbol{Q} = \boldsymbol{I}$,
- *Newton's method:* $\boldsymbol{Q} = [\nabla^2 f(\boldsymbol{x}_k)]^{-1}$.

We will now derive *Newton's method*. First assume that $\nabla^2 f(\boldsymbol{x})$ is positive definite. A second-order Taylor approximation is then:

$$f(\boldsymbol{x}_k + \boldsymbol{p}) - f(\boldsymbol{x}_k) \approx \nabla f(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{p} + \frac{1}{2} \boldsymbol{p}^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}_k) \boldsymbol{p} =: \varphi_{\boldsymbol{x}_k}(\boldsymbol{p})$$

We now try to minimize this approximation by setting the gradient of $\varphi_{\boldsymbol{x}_k}(\boldsymbol{p})$ to zero:

$$\nabla_{\boldsymbol{p}} \varphi_{\boldsymbol{x}_k}(\boldsymbol{p}) = \nabla f(\boldsymbol{x}_k) + \nabla^2 f(\boldsymbol{x}_k) \boldsymbol{p} = \mathbf{0} \qquad \Leftrightarrow \qquad \nabla^2 f(\boldsymbol{x}_k) \boldsymbol{p} = -\nabla f(\boldsymbol{x}_k)$$

Now by choosing the vector fulfilling this we obtain $\boldsymbol{p}_k = -[\nabla^2 f(\boldsymbol{x}_k)]^{-1} \nabla f(\boldsymbol{x}_k)$ as the search direction. When $n = 1$, we get that $p_k = -f'(x_k)/f''(x_k)$.

When the Hessian $\nabla^2 f(\boldsymbol{x}_k)$ is positive definite this search direction is a descent direction. But when $\nabla^2 f(\boldsymbol{x}_k)$ is negative definite (may be also non invertible), the search direction is an ascent direction, meaning that Newton's method does differentiate between minimization and maximization problem. The solution to this problem is to modify $\nabla^2 f(\boldsymbol{x}_k)$ by adding a diagonal matrix $\gamma \boldsymbol{I}$ such that $(\nabla^2 f(\boldsymbol{x}_k) + \gamma \boldsymbol{I})$ is positive definite (this can always be done, why?). This method is called the *Levenberg-Marquardt modification*. We thus take as search direction

$$\boldsymbol{p}_k = -\left[\nabla^2 f(\boldsymbol{x}_k) + \gamma \boldsymbol{I}\right]^{-1} \nabla f(\boldsymbol{x}_k).$$

Note that

- *Steepest descent:* $\gamma = \infty$,
- *Newton's method:* $\gamma = 0$.

What happens when we can not compute $\nabla^2 f(\boldsymbol{x}_k)$? Try to approximate the Hessian in some way choosing approximate matrix $\boldsymbol{B}_k$. From Taylor expansion for $\nabla f(\boldsymbol{x}_k)$ we have that

$$\nabla^2 f(\boldsymbol{x}_k)(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \approx \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1})$$

so the approximate matrix $\boldsymbol{B}_k$ has to fulfill

$$\boldsymbol{B}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) = \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1}).$$

Many different choices of $\boldsymbol{B}_k$ exist, and they lead to what is called *quasi-Newton* methods.

To summarize:

$$\begin{aligned}
\text{Steepest descent:} \quad & \boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k) \\
\text{Netwon's method:} \quad & \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k) \\
\text{Levenberg-Marquardt:} \quad & (\nabla^2 f(\boldsymbol{x}_k) + \gamma \boldsymbol{I})\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k) \\
\text{Quasi-Newton:} \quad & \boldsymbol{B}_k\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k).
\end{aligned}$$

Step 2: Line search

In each iteration one would like to solve

$$\underset{\alpha \geq 0}{\text{minimize}}\, \varphi(\alpha) := f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k).$$

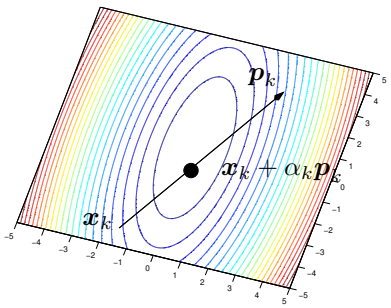
The optimality conditions for the problem are

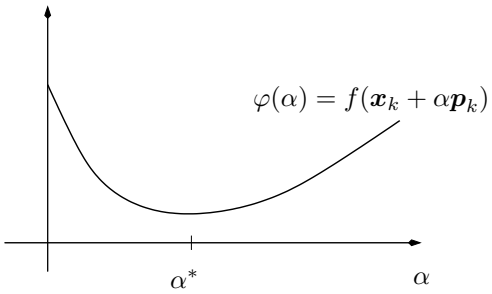
$$\begin{aligned}
\varphi'(\alpha^*) &\geq 0, \\
\alpha^* \varphi'(\alpha^*) &= 0, \\
\alpha^* &\geq 0.
\end{aligned}$$

These conditions state that if $\alpha^* > 0$, then $\varphi'(\alpha^*) = 0$, which implies that

$$\nabla f(\boldsymbol{x}_k + \alpha^* \boldsymbol{p}_k)^{\mathrm{T}} \boldsymbol{p}_k = 0,$$

meaning that the search direction $\boldsymbol{p}_k$ is orthogonal to the gradient of f at $\boldsymbol{x}_k + \alpha^* \boldsymbol{p}_k$.





However, solving the line search problem to optimality is unnecessary. The optimal solution to the original problem lies elsewhere anyway. Examples of methods to choose step lengths α_k

- *Interpolation:* Use $f(\boldsymbol{x}_k)$, $\nabla f(\boldsymbol{x}_k)$, and $\nabla f(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k$ to approximate $\varphi = f(\boldsymbol{x}_k + \alpha\boldsymbol{p}_k)$ quadratically. Then minimize this approximation of φ analytically.
- *Newton's method:* Repeat improvements from a quadratic approximation: $\alpha = \alpha - \varphi'(\alpha)/\varphi''(\alpha)$
- *Golden section:* Derivative-free method which shrinks an interval wherein a solution to $\varphi'(\alpha) = 0$ lies.

We will often use what is denoted as the *Armijo rule*. The idea is to choose a step length α which provides sufficient decrease in f. We have that

$$f(\boldsymbol{x}_k + \alpha\boldsymbol{p}_k) \approx f(\boldsymbol{x}_k) + \alpha\nabla f(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k,$$

for very small values of $\alpha > 0$, meaning that we predict that the objective function will decrease with $\alpha\nabla f(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k$ if we move a step length α in the direction of $\boldsymbol{p}_k$. Now this might be too optimistic, and we will therefore accept the step length if the actual decrease is at least a fraction μ (μ is small, typically $\mu \in [0.001, 0.01]$) of the predicted decrease, i.e., we will accept α if

$$f(\boldsymbol{x}_k + \alpha\boldsymbol{p}_k) - f(\boldsymbol{x}_k) \leq \mu\alpha\nabla f(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k,$$

or equivalently, if

$$\varphi(\alpha) - \varphi(0) \leq \mu\alpha\varphi'(0).$$

We usually start with $\alpha = 1$. If this is not fulfilled, then choose $\alpha := \alpha/2$.

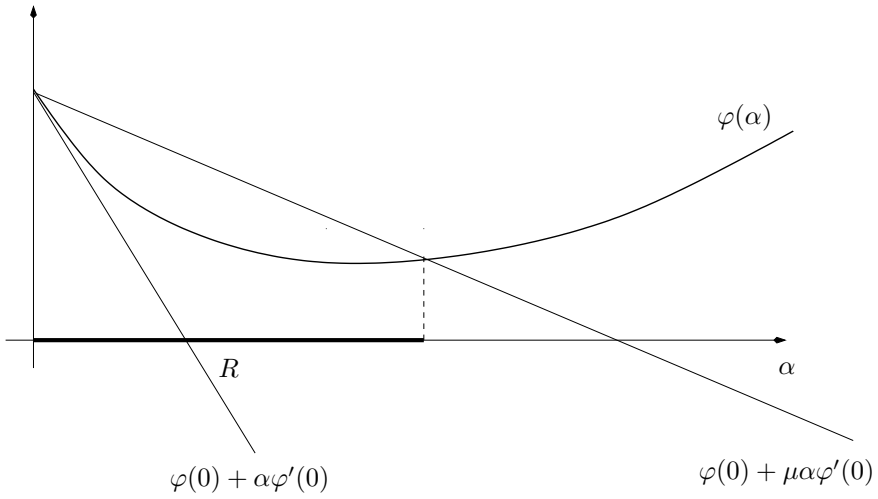


Figure 1: The interval (R) accepted by the Armijo step length rule

Convergence

In order to state a convergence result for the algorithm, we make an additional assumption for the search directions. We need the directions $\boldsymbol{p}_k$ to fulfill

$$-\frac{\nabla f(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p}_k}{\|\nabla f(\boldsymbol{x}_k)\| \cdot \|\boldsymbol{p}_k\|} \geq s_1, \qquad \|\boldsymbol{p}_k\| \geq s_2 ||\nabla f(\boldsymbol{x}_k||, \qquad \text{and} \qquad \|\boldsymbol{p}_k\| \leq M \tag{2}$$

for some $s_1, s_2 > 0$, where the first inequality makes the angle between $\boldsymbol{p}_k$ and $\nabla f(\boldsymbol{x}_k)$ stay between 0 and $\pi/2$, but not too close to $\pi/2$. The second inequality makes sure that the only case when $\boldsymbol{p}_k$ can be zero is when the gradient is zero. These two conditions guarantee a certain descent quality.

Theorem (convergence of unconstrained algorithm)**.** *Suppose $f \in C^1$ and for the starting point $\boldsymbol{x}_0$ the level set $\{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$ is bounded. Consider the iterative algorithm described above. Suppose that for all k, $\boldsymbol{p}_k$ fulfills* (2) *and α_k is chosen according to the Armijo rule. Then*

a) the sequence $\{\boldsymbol{x}_k\}$ is bounded,

b) the sequence $\{f(\boldsymbol{x}_k)\}$ is descending and lower bounded, and

c) every limit point of $\{\boldsymbol{x}_k\}$ is a stationary point.

Proof. See Theorem 11.4 in the book. □

If we add the assumption that f is a convex function, then we can show that

$$\text{optimum exists} \qquad \Longleftrightarrow \qquad \{\boldsymbol{x}_k\} \text{ converges to an optimal solution.}$$

Step 4: Termination criteria

We can not terminate the algorithm when $\nabla f(\boldsymbol{x}_k) = 0$, since this rarely happens. We need to have some tolerance level. Three examples are

a) $\|\nabla f(\boldsymbol{x}_k)\| \leq \varepsilon_1(1 + |f(\boldsymbol{x}_k)|)$, where $\varepsilon_1 > 0$ is small.

b) $f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k) \leq \varepsilon_2(1 + |f(\boldsymbol{x}_k)|)$, where $\varepsilon_2 > 0$ is small.

c) $\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\| \leq \varepsilon_3(1 + \|\boldsymbol{x}_k\|)$, where $\varepsilon_3 > 0$ is small.

Can also use the max-norm $\|\cdot\|_\infty$ instead.

A note on trust region methods

Trust region methods use a quadratic approximation of the function around the current iterate $\boldsymbol{x}_k$, avoid a line search but instead bound the length of the search direction. Let

$$\varphi_{\boldsymbol{x}_k}(\boldsymbol{p}) := f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^{\mathrm{T}}\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p}.$$

Since this is a local approximation, we restrict our approximation to a *trust region* in the neighborhood of $\boldsymbol{x}_k$, i.e., we trust the model in the region where $\|\boldsymbol{p}\| \leq \Delta_k$. We then solve the problem to

$$\begin{aligned} &\text{minimize } \varphi_{\boldsymbol{x}_k}(\boldsymbol{p}), \\ &\text{subject to } \|\boldsymbol{p}\| \leq \Delta_k. \end{aligned}$$

and let the solution be $\boldsymbol{p}_k$. Then we update our iterate as $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$. We also update the trust region parameter Δ_k depending on the progress so far (actual reduction/predicted reduction). The method is robust and possess strong convergence. More detailed information about trust region methods can be found in the book on pages 301–302.

A note on black-box functions

In some cases the value of the objective function $f(\boldsymbol{x})$ is given through some unknown simulation procedure. This implies that we do not have a clear representation of the gradient of the objective function. In some cases, we can perform *numerical differentiation* and approximate the partial derivatives as, e.g.,

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} \approx \frac{f(\boldsymbol{x} + \alpha \boldsymbol{e}_i) - f(\boldsymbol{x})}{\alpha},$$

where $\boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}}$ is the unit vector in $\mathbb{R}^n$.

If the simulation is not accurate, we get a bas derivative information. We can use *derivative-free methods* instead. These try to build a model $\hat{f}$ of the objective function f from evaluating the objective function at some specific test points and optimize the model $\hat{f}$ instead of the function f.