Lecture 4

Iterative algorithms for solving a problem of the form

$$\text{minimize} \quad f(\boldsymbol{x}) \tag{4.1}$$

on $\mathbb{R}^n$ generate a sequence of vectors $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots$ in the hope that this sequence converges to a (local or global) minimizer $\boldsymbol{x}^*$ of (4.1). In this lecture we first study step length selection procedures and then study what it means for a sequence to converge, and how to quantify the speed of convergence.

4.1 Step length selection

When moving in a descent direction $\boldsymbol{p}_k$ (not necessarily the steepest descent direction),

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$$

we can only guarantee that the function value will decrease if the step length α_k is small enough. If we move too far in a descent direction, we might even land at a point where f is larger than where we started! It is therefore important to choose a step length that

- is not too small (so that the algorithm does not take too long);
- is not too large (so that we don't end up at a point with larger function value);
- is easy to compute.

An optimal step α for a descent method would be the minimizer of the function

$$\alpha \mapsto \varphi(\alpha) := f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k).$$

In practice, minimizing this function is not always the most efficient thing to do (or even possible). One would rather choose a step length that satisfies some criteria that ensure that the sequence $\boldsymbol{x}_k$ converges to a minimizer $\boldsymbol{x}^*$ under suitable conditions on a function f. One such condition is a **sufficient decrease condition**,

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + c\alpha \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle =: \ell(\alpha),$$

with $c \in (0,1)$. Note that $\varphi'(0) = \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle < 0$, because $\boldsymbol{p}_k$ is a descent direction. The function $\ell(\alpha)$ is therefore a line through $f(\boldsymbol{x}_k)$ with a slope $c\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle = c\varphi'(0) > \varphi'(0)$. To see why this condition is necessary, consider the function $f(x) = x^2 - 1$ and the sequence $x_k = \sqrt{1 + 1/k}$ for $k \geq 1$. Clearly, the sequence $f(\boldsymbol{x}_k) = 1/k$ decreases, but fails to converge to the minimizer $f(0) = -1$.

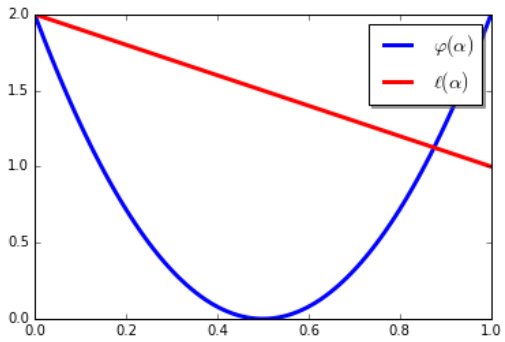

Figure 4.1: The sufficient decrease condition

The sufficient decrease condition (also called Armijo condition) can always be satisfied if α is chosen small enough, but the algorithm may become very slow. It is therefore common to supplement the sufficient descent condition with other criteria that guarantee that sufficient progress is made. Two of the commonly used criteria are:

- the Wolfe conditions, which add a *curvature condition*

  $$\varphi'(\alpha) \geq \tilde{c}\varphi'(0)$$

  for some $\tilde{c} \in (c, 1)$, which gives a lower bound on the slope of the new point;

- the Armijo-Goldstein conditions, which state that a step length α_k should additionally satisfy bound

  $$f(\boldsymbol{x}_k) + (1-c)\alpha_k \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle \leq f(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k), \tag{4.1}$$

  which gives a lower bound on the step size.

Another common approach is **backtracking**: in this method one uses a high initial value of α (for example, $\alpha = 1$), and then decreases it until the sufficient descent condition is satisfied.

Example 4.1. Consider the function $f\colon \mathbb{R}^2 \to \mathbb{R}$ defined by $f(\boldsymbol{x}) = x_1^2 + x_2^2$. The gradient is $\nabla f(\boldsymbol{x}) = 2\boldsymbol{x}$, and the φ function at $\boldsymbol{x}_k = (1,1)^\top$

$$\varphi(\alpha) = f(\boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)) = 2(1-2\alpha)^2, \quad \varphi'(\alpha) = -8(1-2\alpha).$$

The Armijo-Goldstein conditions (4.1) then state that we can choose α such that

$$2(1-4(1-c)\alpha) \leq 2(1-2\alpha)^2 \leq 2(1-4c\alpha).$$

The optimal step length in this case would be $\alpha = 0.5$.

4.2 Convergence of iterative methods

A sequence of vectors $\{\boldsymbol{x}_k\}$ in $\mathbb{R}^n$, $k \geq 0$, *converges* to a vector $\boldsymbol{x}^*$ with respect to a norm $\|\cdot\|$ as $k \to \infty$, written $\boldsymbol{x}_k \to \boldsymbol{x}$, if the sequence of numbers $\|\boldsymbol{x}_k - \boldsymbol{x}^*\|$ converges to zero. More formally, if for every $\varepsilon > 0$ there exists an index N such that for all $n \geq N$,

$$\|\boldsymbol{x}_n - \boldsymbol{x}^*\| < \varepsilon.$$

Iterative algorithms will rarely find the exact solution to a problem like (4.1), so we will usually be happy to find a solution that differs from the true one by at most some specified accuracy.

Definition 4.2. A sequence of vectors $\{\boldsymbol{x}_k\}$, $k \geq 0$, is said to converge to $\boldsymbol{x}^*$

(a) linearly (or Q-linear, Q for quotient), if there exist an $r \in (0,1)$ such that for sufficiently large k,

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq r\|\boldsymbol{x}_k - \boldsymbol{x}^*\|.$$

(b) superlinearly, if

$$\lim_{k\to\infty} \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|} = 0,$$

(c) with order p, if there exists a constant $M > 0$, such that for sufficiently large k,

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

The case $p = 2$ is called *quadratic convergence*.

Of course, as mentioned earlier, these definitions depend on the choice of a norm. It can be shown that quadratic convergence implies superlinear convergence, and superlinear convergence implies linear convergence.

Example 4.3. Consider the sequence of numbers $x_k = 1/2^{r^k}$ for some $r > 1$. Clearly, $x_k \to x^* = 0$ as $k \to \infty$. Moreover,

$$x_{k+1} = \frac{1}{2^{r^{k+1}}} = \frac{1}{2^{r^k r}} = \left(\frac{1}{2^{r^k}}\right)^r = x_k^r,$$

which shows that the sequence has rate of convergence r.

4.3 Convergence of gradient descent

In this section, a norm $\|\cdot\|$ will refere to the 2-norm, unless otherwise stated. We now study the convergence of gradient descent for the least squares problem

$$\text{minimize} \quad f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2, \tag{4.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$ of full rank. As we have seen in Lecture 3, the gradient descent method is the procedure

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{r}_k,$$

where the step length and the *residual* are given by

$$\alpha_k = \frac{\|\boldsymbol{r}_k\|^2}{\|\boldsymbol{A}\boldsymbol{r}_k\|^2}, \quad \boldsymbol{r}_k = \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) = -\nabla f(\boldsymbol{x}_k).$$

At the minimizer, the residual is

$$\boldsymbol{r} = -\nabla f(\boldsymbol{x}^*) = \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^*) = 0, \tag{4.2}$$

and as the sequence $\boldsymbol{x}_k$ converges to $\boldsymbol{x}^*$, the norms of the residuals converge to 0. Conversely, the residual is related to the difference $\boldsymbol{x}_k - \boldsymbol{x}^*$ by

$$\boldsymbol{r}_k = \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) = \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k - (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^*)) = \boldsymbol{A}^\top\boldsymbol{A}(\boldsymbol{x_k} - \boldsymbol{x}^*), \tag{4.3}$$

where we used the "intelligent zero" (4.2). Therefore

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\| = \|(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{r}\| \leq \|(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\|\|\boldsymbol{r}_k\|,$$

where $\|\boldsymbol{B}\| = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \|\boldsymbol{B}\boldsymbol{x}\|/\|\boldsymbol{x}\|$ is the operator norm of a matrix $\boldsymbol{B}$ with respect to the 2-norm. Consequently, if the sequence $\|\boldsymbol{r}_k\|$ converges to zero, so does the sequence $\|\boldsymbol{x}_k - \boldsymbol{x}^*\|$. A reasonable criterium is to stop the algorithm is therefore when the residual norm $\|\boldsymbol{r}_k\|$ is below a predefined tolerance ε.

The following theorem (whose proof we omit) shows that the gradient descent method for linear least squares converges linearly with respect to the $\boldsymbol{A}$ norm. The statement involves the *condition number* of $\boldsymbol{A}$[1]. This quantity is defined as

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^\dagger\|.$$

Theorem 4.4. *The error in the $k+1$-th iterate is bounded by*

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| \leq \left(\frac{\kappa^2(\boldsymbol{A}) - 1}{\kappa^2(\boldsymbol{A}) + 1}\right)\|\boldsymbol{x}_k - \boldsymbol{x}^*\|.$$

In particular, the gradient descent algorithm converges linearly.

[1]The concept of condition number, introduced by Alan Turing while in Manchester, is one of the most important ideas in numerical analysis, as it is indispensable in studying the performance of numerical algorithms.