Optimization Algorithms
Weekly Exercise 1

Marc Toussaint
Learning & Intelligent Systems Lab, TU Berlin
Marchstr. 23, 10587 Berlin, Germany

Winter 2024/25

1 Toy Problems and Plotting

Consider the following functions over $x \in \mathbb{R}^n$:

$$f_{\text{sq}}(x) = x^\top C x \ , \tag{1}$$

$$f_{\text{hole}}(x) = \frac{x^\top C x}{a^2 + x^\top C x} \ , \tag{2}$$

$$f_{\text{exp}}(x) = -\exp\Big(-\frac{1}{2} x^\top C x\Big) \ . \tag{3}$$

For $C = \mathbf{I}$ (identity matrix) the first would be fairly simple to optimize. The C matrix changes the *conditioning* (ratio of largest and smalles Hessian eigenvalues) of these functions and makes them more interesting. We assume that C is a diagonal matrix with entries $C_{ii} = c^{\frac{i-1}{n-1}}$.

a) What are the gradients $\nabla f(x)$ of these three functions?

b) What are the Hessians $\nabla^2 f(x)$ of these three functions?

c) Implement these functions in python and plot the above functions for $c = 10$ and $a = .1$ over $x = [-1, 1]^2$. Tip: First evaluate the function over a `np.meshgrid`, then `matplotlib` and `plot\_surface` are useful.

Outlook: We will soon have the first coding assignments, where you have to implement both, the problems (functions and their gradients), and the solvers. If you want to try already, implement the above functions to also return the gradient. Then try vanilla gradient descent starting at $x_0 = (1, 1)$, outputting basic information (#iteration, current-cost) in each iteration, and storing the trace $x_{0,..}$ in a matrix for plotting.

a) The following assumes that C is symmetric (not necessarily diagonal)
As column vectors:

$$\nabla f_{\text{sq}}(x) = 2Cx \tag{4}$$

$$\nabla f_{\text{hole}}(x) = \frac{2}{a^2 + x^T C x} C x - 2\frac{x^T C x}{(a^2 + x^T C x)^2} C x = \frac{2a^2}{(a^2 + x^T C x)^2} C x \tag{5}$$

$$\nabla f_{\text{exp}}(x) = \exp\Big(-\frac{1}{2} x^\top C x\Big)(Cx) \tag{6}$$

b)

$$\nabla^2 f_{\text{sq}}(x) = 2C \tag{7}$$

$$\nabla^2 f_{\text{hole}}(x) = \frac{2a^2}{(a^2 + x^T C x)^2} C - \frac{8a^2}{(a^2 + x^T C x)^3} C x x^T C \tag{8}$$

$$\nabla^2 f_{\text{exp}}(x) = -\exp\Big(-\frac{1}{2} x^\top C x\Big) C x (x^T C) + \exp\Big(-\frac{1}{2} x^\top C x\Big) C \tag{9}$$

Plotting solution at the end.

2 Convergence proof

The following aims to guide you through a proof of the convergence theorem. This course is not focussing on theory – so this is an exception. But the steps below are also a good exercise to train basic maths that is relevant throughout the course.

We are given a function $f : \mathbb{R}^n \to \mathbb{R}$ with $f_{\text{Min}} = \min_x f(x)$. Assume that the eigenvalues of its Hessian $\nabla^2 f$ are lower bounded by $m > 0$ and upper bounded by $M > m$, with $m, M \in \mathbb{R}$. Recall that the 2nd-order Taylor approximation of $f(y)$ around x is

$$f(y) \approx f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top \nabla^2 f(x)(y - x)$$

a) Analogous to the 2nd Taylor, provide an upper and lower bound of $f(y)$, using the upper and lower curvatures M and m, respectively. This tells us that the function $f(y)$ is "squeezed" between a lower bound paraboloid with minimal curvature, and an upper bound paraboloid with maximal curvature, which "touch" each other at location x with value $f(x)$ and gradient $\nabla f(x)$.

b) Find the minima of both, the upper and lower bound paraboloids. Then prove that for any $x \in \mathbb{R}^n$ it holds

$$f(x) - \frac{1}{2m}|\nabla f(x)|^2 \leq f_{\text{Min}} \leq f(x) - \frac{1}{2M}|\nabla f(x)|^2 \ .$$

as well as

$$|\nabla f(x)|^2 \geq 2m(f(x) - f_{\text{Min}}) \ .$$

c) Consider backtracking line search with Wolfe parameter $\varrho_{\text{ls}} \leq \frac{1}{2}$, and step decrease factor ϱ_α^-. Assume that $\alpha \leq \frac{1}{M}$. Prove that the step $x + \alpha\delta$ fulfills the Wolfe condition (is sufficiently decreasing the function) and therefore line search terminates.

d) Also argue that, if α is initially large but then repeatedly decreased with $\alpha \leftarrow \varrho_\alpha^- \alpha$, line search terminates for some α within $\frac{\varrho_\alpha^-}{M} \leq \alpha \leq \frac{1}{M}$.

e) Conclude the prove by showing that line search stops at a point y for which

$$f(y) \leq f(x) - \frac{\varrho_{\text{ls}}\varrho_\alpha^-}{M}|\nabla f(x)|^2 \ .$$

and

$$f(y) - f_{\text{Min}} \leq \left[1 - \frac{2m\varrho_{\text{ls}}\varrho_\alpha^-}{M}\right](f(x) - f_{\text{Min}}) \ .$$

a) From Taylor's theorem (with the mean-value forms of the remainder), we have

$$f(y) = f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top \nabla^2 f(z)(y - x), \tag{10}$$

with $z = tx + (1 - t)y$ for some $t \in [0, 1]$.

—

Note: The expression above is related to the 2nd-order Taylor approximation of $f(y)$ around x.

$$f(y) \approx f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top \nabla^2 f(x)(y - x), \tag{11}$$

In (11), we can repace the $\approx$ with $=$ when $||y - x|| \to 0$. On the other hand, (10) holds with equality for some t.

—

Let $\nabla^2 f(z) = Q\Lambda Q^\top$ be the Eigendecomposition of the Hessian matrix with $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n)$, $M \geq \lambda_1 > \cdots > \lambda_n \geq m > 0$. Because Q is an orthogonal matrix, we can represent any $v \in \mathbb{R}^n$ as $v = Qw$. This leads to

$$v^\top \nabla^2 f(z) v = w^\top Q^\top Q \Lambda Q^\top Q w = w^T \Lambda w = \sum_i \lambda_i w_i^2 \tag{12}$$

$$\therefore m|v|^2 \leq v^\top \nabla^2 f(z) v \leq M|v|^2. \tag{13}$$

Substituting this into (10) gives us an upper and lower bound of $f(y)$:

$$f(x) + \nabla f(x)^\top (y - x) + \frac{m}{2}||y - x||^2 \leq f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{M}{2}||y - x||^2. \tag{14}$$

b) The minimum of the lower bound, $f_{\text{lb}}(y) = f(x) + \nabla f(x)^\top (y - x) + \frac{m}{2}(y - x)^2$, can be obtained analytically as:

$$\frac{\partial f_{\text{lb}}(y)}{\partial y} = \nabla f(x)^\top + m(y - x)^\top = 0 \tag{15}$$

$$\therefore y_{\text{lb}}^* = x + \frac{\nabla f(x)}{m}, \ f_{\text{lb}}^* = f(x) - \frac{1}{2m}|\nabla f(x)|^2. \tag{16}$$

Similarly for the upper bound, $f_{\text{ub}}(y) = f(x) + \nabla f(x)^\top (y-x) + \frac{M}{2}||y-x||^2$, we have

$$y^*_{\text{ub}} = x + \frac{\nabla f(x)}{m},\ f^*_{\text{ub}} = f(x) - \frac{1}{2M}|\nabla f(x)|^2. \tag{17}$$

This leads to

$$f(x) - \frac{1}{2m}|\nabla f(x)|^2 \le f_{\text{Min}} \le f(x) - \frac{1}{2M}|\nabla f(x)|^2 \tag{18}$$

and

$$|\nabla f(x)|^2 \ge 2m(f(x) - f_{\text{Min}})\ . \tag{19}$$

c) Substitute $y = x - \alpha \nabla f(x)$ into f_{ub} and use $\alpha \le \frac{1}{M}$:

$$f(y) \le f(x) - \alpha|\nabla f(x)|^2 + \frac{M\alpha^2}{2}|\nabla f(x)|^2 \tag{20}$$

$$\le f(x) - \frac{\alpha}{2}|\nabla f(x)|^2 \tag{21}$$

$$\le f(x) - \varrho_{\text{ls}}\alpha|\nabla f(x)|^2 \tag{22}$$

(Here the step $\delta = -\nabla f$.) We just proved $\alpha \le \frac{1}{M} \Rightarrow$ (Wolfe condition holds) $\Rightarrow$ (line search stops)

d) Since α decays exponentially with the factor ϱ^-_α until it becomes smaller than $\frac{1}{\alpha}$ (as we proved in c)), the line search will terminate with the parameter $\frac{\varrho^-_\alpha}{M} < \alpha \le \frac{1}{M}$.

e) Applying $\frac{\varrho^-_\alpha}{M} < \alpha$ to (22) gives

$$f(y) < f(x) - \frac{\varrho_{\text{ls}}\varrho^-_\alpha}{M}|\nabla f(x)|^2 \tag{23}$$

Finally, we can prove the exponential convergence of backtracking line search:

$$f(y) - f_{\text{Min}} < f(x) - f_{\text{Min}} - \frac{\varrho_{\text{ls}}\varrho^-_\alpha}{M}|\nabla f(x)|^2 \tag{24}$$

$$< f(x) - f_{\text{Min}} - \frac{2m\varrho_{\text{ls}}\varrho^-_\alpha}{M}(f(x) - f_{\text{Min}}) \tag{25}$$

$$< \left[1 - \frac{2m\varrho_{\text{ls}}\varrho^-_\alpha}{M}\right](f(x) - f_{\text{Min}}) \tag{26}$$

Notes: This proof works for convex functions. During the proof, we have used that m and M are positive. In the non-convex case, gradient descent convergences only to a local optimum or saddle point.

```
#!/usr/bin/python3

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

# simple quadratic function
def myfunc(x):
    y = np.array(x)
    fx = y.T @ y
    return fx

def plotFunc(f, bounds_lo, bounds_up, trace_xy = None, trace_z = None):
    x = np.linspace(bounds_lo[0], bounds_up[0], 30)
    y = np.linspace(bounds_lo[1], bounds_up[1], 30)
    xMesh, yMesh = np.meshgrid(x, y, indexing='ij')
    zMesh = np.zeros_like(xMesh)
    for i in range(x.shape[0]):
        for j in range(y.shape[0]):
            zMesh[i,j] = f([xMesh[i,j], yMesh[i,j]])
```

```
    fig = plt.figure(figsize=(12,6))

    ax1 = fig.add_subplot(121, projection="3d")
    surf = ax1.plot_surface(xMesh, yMesh, zMesh, cmap=cm.coolwarm)
    if trace_xy is not None: ax1.plot(trace_xy[:,0], trace_xy[:,1], trace_z, 'ko-')
    fig.colorbar(surf)
    ax1.set_xlabel('x')
    ax1.set_ylabel('y')
    ax1.set_zlabel('f')

    ax2 = fig.add_subplot(122)
    surf2 = plt.contourf(xMesh, yMesh, zMesh, cmap=cm.coolwarm)
    if trace_xy is not None: ax2.plot(trace_xy[:,0], trace_xy[:,1], 'ko-')
    fig.colorbar(surf2)
    ax2.set_xlabel('x')
    ax2.set_ylabel('y')

    plt.show()


plotFunc(myfunc, [-2,-2], [2,2], np.array([[1,1],[.2,.2]]), [8,8])
```