OPTIMIZATION - OPTIMALISATIE

EXERCISE SESSION 6: GAUSS-NEWTON & NEWTON-LAGRANGE

Exercise 1 (Gauss-Newton method).

Let us consider the nonlinear least-squares (NLS) problem:

$$\underset{x \in \mathbb{R}^n}{\textbf{minimize}} \quad f(x) = \frac{1}{2}\|F(x)\|_2^2,$$

where $F(x) = (f_1(x), \ldots, f_m(x))$ and $f_i : \mathbb{R}^n \to \mathbb{R}$ are sufficiently smooth.

The gradient of f is given by

$$\nabla f(x) = JF(x)^\top F(x),$$

where $JF(x)$ is the Jacobian of F at x. If we linearize F around x we get

$$F(x + \Delta x) \approx F(x) + JF(x)\Delta x.$$

Therefore, we can approximate ∇f at the point $x + \Delta x$ by

$$\nabla f(x + \Delta x) \approx JF(x)^\top F(x) + JF(x)^\top JF(x)\Delta x. \tag{1.1}$$

When at point x_k, we can compute $x_{k+1} = x_k + \Delta x$ by setting (1.1) equal to zero and solving for Δx. Indicating $F_k = F(x_k)$ and $J_k = JF(x_k)$, that is

$$\Delta x = -(J_k^T J_k)^{-1} J_k^T F_k.$$

We already know how to do a linesearch, so let's use Δx as our search direction d_k. When you perform the linesearch, be sure you are evaluating the original function and gradient $f(x)$ and $\nabla f(x)$, and not $F(x)$ and $JF(x)$.

Tasks:

(1a) Implement a Gauss-Newton minimizer by completing the given function in `S6-GN.ipynb`. This minimizer takes in the function $F(x)$ and an initial guess x_0.

Warning 1. The provided `line_search` function has a slightly different prototype than last time: it is given $\nabla f_k^T d_k$ instead of ∇f_k as one of its arguments.

Warning 2. The provided `finite_difference_jacob` always returns the Jacobian and not the gradient, regardless whether the function is real- or vector-valued. Therefore, in order to retrieve the gradient f you need to transpose the Jacobian.

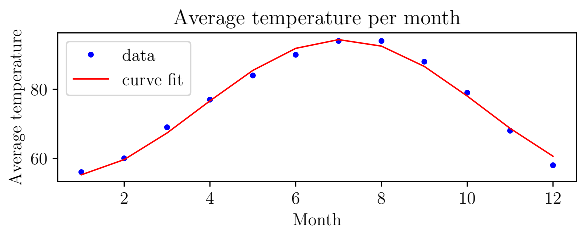
(1b) A typical problem solved by Gauss-Newton is nonlinear regression. We will fit monthly average temperatures using a sinusoidal model:

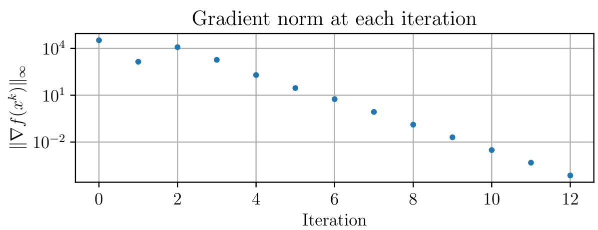
$$T(k) = A\,\mathbf{sin}(\omega k + \phi) + C. \tag{1.2}$$

Here T denotes the average temperature for month k. In particular, given a data set of $\{k_i, T_i\}$ pairs, the Gauss-Newton method minimizes the objective function:

$$\underset{A,\omega,\phi,C}{\textbf{minimize}} \quad \frac{1}{2}\sum_i (T(k(i)) - y_i)^2. \tag{1.3}$$

Fit these data by completing the given code. Try different initial points. What do you observe? A successful minimization should look like:





Exercise 2 **(Newton-Lagrange Method).**

Consider the problem:

$$\underset{x\in\mathbb{R}^n}{\textbf{minimize}} \quad f(x) \quad \text{subject to} \quad h(x) = 0, \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^p$. We handle the constraints by introducing the Lagrange multipliers $\lambda \in \mathbb{R}^p$ and the Lagrangian function:

$$L(x, \lambda) = f(x) + \lambda^\top h(x).$$

Our goal is to solve the nonlinear system of equations $\nabla L(x, \lambda) = 0$ defining the optimality conditions (KKT), namely

$$\begin{aligned} 0 &= \nabla_x L(x, \lambda) = \nabla f(x) + Jh(x)^\top \lambda \\ 0 &= \nabla_\lambda L(x, \lambda) = h(x). \end{aligned}$$

We linearize this system around (x_k, λ_k): denoting $f_k \coloneqq f(x_k)$, $h_k \coloneqq h(x_k)$, $x \coloneqq x_k + \Delta x$, $\lambda \coloneqq \lambda_k + \Delta\lambda$ and $L_k \coloneqq L(x_k, \lambda_k)$,

$$\begin{aligned} \nabla_x L(x, \lambda) \approx{}& \overbrace{\nabla_x L(x_k, \lambda_k)}^{\nabla f_k + Jh_k^\top \lambda_k} + \overbrace{\nabla_x \left(\nabla_x L(x_k, \lambda_k)\right)^\top}^{\nabla^2_{xx} L_k} \Delta x + \overbrace{\nabla_\lambda \left(\nabla_x L(x_k, \lambda_k)\right)^\top}^{Jh_k^\top} \Delta\lambda \\ &= \nabla f_k + Jh_k^\top \lambda_k + \nabla^2_{xx} L_k \Delta x + Jh_k^\top (\lambda - \lambda_k) \\ &= \nabla f_k + \nabla^2_{xx} L_k \Delta x + Jh_k^\top \lambda \end{aligned}$$

and

$$\nabla_\lambda L(x, \lambda) \approx h_k + Jh_k \Delta x.$$

We now have the linearized system:

$$\begin{bmatrix} \nabla^2_{xx} L_k & Jh_k^\top \\ Jh_k & 0 \end{bmatrix} \begin{pmatrix} \Delta x \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f_k \\ -h_k \end{pmatrix}. \tag{2.2}$$

By solving (2.2) we obtain Δx and λ. Therefore, we can set $\lambda_{k+1} = \lambda$ and find x_{k+1} with a line search in direction $d_k = \Delta x$ to obtain $x_{k+1} = x_k + \alpha_k\, d_k$. The step length α_k can be computed by employing the Armijo condition:

$$\varphi(x_k + \alpha_k d_k) \le \varphi(x_k) + \sigma \alpha_k \varphi'(x_k; d_k) \tag{2.3}$$

where $\varphi(x)$ is the *merit function* defined by

$$\varphi(x) \coloneqq f(x) + c\|h(x)\|_1, \tag{2.4}$$

where $c > 0$ is the *penalty parameter*. $\varphi(x)$ is nonsmooth due to the nonsmoothness of the $\|\cdot\|_1$ norm; however, the *directional derivative* $\varphi'(x_k; d_k)$ exists and equals

$$\varphi'(x_k; d_k) = \nabla f_k^\top d_k - c\|h_k\|_1. \tag{2.5}$$

Tasks:

(2a) Implement the Newton Lagrange method, using the BFGS method to approximate $\nabla^2_{xx} L_k$, by completing the function `minimize_nl`. The BFGS algorithm should follow Section 7.6.1 (page 135) of the Numerical Optimization script [1] or the Exercise 3 from Exercise Session 3. This minimizer takes in the functions $f(x)$, $h(x)$ and an initial guess x_0.

Use the the stopping criterion:

$$\left\| \begin{matrix} \nabla_x L(x_k, \lambda_k) \\ \nabla_\lambda L(x_k, \lambda_k) \end{matrix} \right\|_\infty = \left\| \begin{matrix} \nabla f_k + J h_k^\top \lambda_k \\ h_k \end{matrix} \right\|_\infty < 10^{-4}.$$

(2b) Solve the problem:

$$\underset{x,y}{\textbf{minimize}} \quad \tfrac{1}{2}\left(x^2 + (y/2)^2\right) \quad \text{s.t.} \quad y = (x-1)^2 - x + 3$$

by completing the given code.

[1] https://p.cygnus.cc.kuleuven.be/bbcswebdav/pid-30052757-dt-content-rid-307696315_2/xid-307696315_2.

Exercise 3 **(Constrained Gauss-Newton).**

We now consider a constrained version of the problem in **Exercise 1**, and we solve it by integrating Gauss-Newton and Newton-Lagrange methods.

$$\underset{x\in\mathbb{R}^n}{\textbf{minimize}} \quad f(x) \quad \text{subject to} \quad h(x) = 0. \tag{3.1}$$

Again, we handle the constraints by introducing the Lagrange multipliers λ and the Lagrange function

$$L(x,\lambda) = \tfrac{1}{2}\|F(x)\|_2^2 + \lambda^\top h(x)$$

and use a linear approximation of F and h around x_k. Differently from what we did in **Exercise 2**, we only linearize the part in x:

$$L(x_k + \Delta x, \lambda) \approx \tfrac{1}{2}F_k^\top F_k + F_k^\top J_k \Delta x + \tfrac{1}{2}\Delta x^\top J_k^\top J_k \Delta x + \lambda^\top(h_k + Jh_k \Delta x)$$

where

$$h_k \coloneqq h(x_k) \quad \text{and} \quad J_k \coloneqq JF(x_k).$$

The *linearized* problem (linearized around x_k) therefore is

$$\underset{x}{\textbf{minimize}} \quad \tfrac{1}{2}\|F_k + J_k\Delta x\|_2^2 \quad \text{subject to} \quad h_k + Jh_k\Delta x = 0. \tag{3.2}$$

The optimality conditions of eq. (3.2) are

$$J_k^\top J_k \Delta x + J_k^\top F_k + Jh_k^\top \lambda = 0 \quad \text{and} \quad h_k + Jh_k \Delta x = 0,$$

which can be written in compact form as

$$\begin{bmatrix} J_k^\top J_k & Jh_k^\top \\ Jh_k & 0 \end{bmatrix} \begin{pmatrix} \Delta x \\ \lambda \end{pmatrix} = \begin{pmatrix} -J_k^\top F_k \\ -h_k \end{pmatrix}. \tag{3.3}$$

By solving eq. (3.3) we obtain Δx and λ. Once again we use $d_k = \Delta x$ as our search direction and perform a line search to obtain $x_{k+1} = x_k + \alpha_k\, d_k$. The step length α_k can be computed by employing the same Armijo condition as before, namely

$$\varphi(x_k + \alpha_k d_k) \le \varphi(x_k) + \sigma\alpha_k\varphi'(x_k; d_k) \tag{3.4}$$

where $\phi(x)$ is the merit function defined by

$$\varphi(x) \coloneqq f(x) + c\|h(x)\|_1 \tag{3.5}$$

whose directional derivative is

$$\varphi'(x_k; d_k) = \nabla F_k^\top d_k - c\|h_k\|_1. \tag{3.6}$$

Though methods have been developed to adapt c at each iteration, you might start by fixing the value to $c = 100$.

Tasks:

(3a) Implement a Constrained Gauss-Newton minimizer by completing the function `minimize_cgn`. This minimizer takes in the functions $F(x)$, $h(x)$ and an initial guess x_0. Remember that the provided `line_search` function has a slightly different prototype than last time (to account for the merit function).

Use the the stopping criterion:

$$\left\| \begin{matrix} \nabla_x L(x_k, \lambda_k) \\ \nabla_\lambda L(x_k, \lambda_k) \end{matrix} \right\|_\infty = \left\| \begin{matrix} \nabla f_k + Jh_k^\top \lambda_k \\ h_k \end{matrix} \right\|_\infty < 10^{-4}.$$

(3b) Again, solve the problem

$$\underset{x,y}{\textbf{minimize}} \quad \tfrac{1}{2}\left(x^2 + ({}^{y}\!/_{2})^2\right) \quad \text{s.t.} \quad y = (x-1)^2 - x + 3$$

by completing the given code.

Play with the initial guess x_0 and see how the convergence is affected.