NUMERICAL OPTIMISATION
COURSE WORK (50%)

Marta Betcke

Submit your solutions to `Turnitin`. Your report **must be typeset** and should not exceed 15 pages. Most questions can be answered with a few sentences in addition to equations and plots. Please only include code into your report when explicitly asked for (please follow the particular guidelines, most of the time we only want to see small snippets).

UCL rules and regulations on late submission and plagiarism apply.

EXERCISE 1 [27pt]

We are given a function $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x, y) = (y - \cos x)^2 + (y - x)^2.$$

(a) Calculate the gradient ∇f and the Hessian $\nabla^2 f$. **[2 pt]**

(b) Find the minimiser x^* of the function f and describe how you obtained it. ***Hint:*** *You may compute it numerically but verify analytically that it satisfies the 1st order necessary condition.* Show that x^* is unique. **[5 pt]**

(c) Plot the function and its contours. Show that the level sets of f are bounded, ***Hint:*** *To simplify the problem consider the level sets in the limit $|x|, |y|$ large.* **[5 pt]**

(d) Show that the gradient, ∇f, is locally Lipschitz continuous. ***Hint:*** *Recall integral form of Taylor's theorem and that Frobenious norm is an upper bound on the L_2 matrix norm.* **[5 pt]**

(e*) Show that the Hessian, $\nabla^2 f$, is locally Lipschitz continuous. ***Hint:*** *Young's inequality is a useful way of bounding products.* **[10 pt]**

EXERCISE 2 [18pt]

(a) Apply steepest descent with an appropriate line search to the function f in **Ex.1** starting from $x_0 = (1, -1)^T$ and $x_0 = (-1, 0)^T$. Plot the iterates over the function contours. State your choice of a line search and any important parameters. What do you observe? **[2pt]**

(b) Investigate convergence of the steepest descent **(a)** iterates a posteriori and include one relevant error plot. What are the empirical convergence rates and how did you obtain them? Do they agree with the theoretical predictions? Paraphrase the relevant theoretical results. **[4pt]**

(c) Apply Newton with an appropriate line search to the function f in **Ex.1** starting from $x_0 = (1, -1)^T$ and $x_0 = (-1, 0)^T$. Plot the iterates over the function contours. State your choice of a line search and any important parameters. What do you observe? **[2pt]**

(d) Investigate convergence of the Newton **(c)** iterates a posteriori and include one relevant error plot. What are the empirical convergence rates and how did you obtain them? Do they agree with the theoretical predictions? Paraphrase the relevant theoretical results. **[4pt]**

(e) Can global convergence of both methods be guaranteed or not and why? Paraphrase the relevant theoretical results. **[6pt]**

EXERCISE 3 [15pt]

(a) Implement the dogleg trust region method for strictly convex functions (with s.p.d. Hessian). Your implementation should return the *Cauchy point* whenever the gradient and Newton steps are collinear.

Include your implementation into the report. Highlight the part where you solve for the intersection point between the trust region and the dogleg path and provide a short narrative explanation. **[6pt]**

(b) Apply the dogleg trust region method to minimise the Rosenbrock function $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$$

with two different starting points $x_0 = (0.5, 1)^T$ and $x_0 = (-1.5, 1)^T$. Plot the trajectories traced by the iterates over the function contours. State your choice of the stopping condition and any relevant parameters. What do you observe? **[2pt]**

(c) Investigate convergence of the dogleg iterates in (b) a posteriori and include one relevant error plot. What are the empirical convergence rates and how did you obtain them? Do they agree with the theoretical predictions? Paraphrase the relevant theoretical results. **[4pt]**

(d) Can global convergence be expected or not, and why? Paraphrase the relevant theoretical results. **[3pt]**

EXERCISE 4 [15pt]

(a) Implement the Polar-Ribieṙe conjugate gradient method. *Hint: Modify the* `descentLineSearch.m` *template from tutorial 2.* Copy the relevant lines in your report. **[2pt]**

(b) Apply Polar-Ribieṙe conjugate gradient method to minimise the function $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x, y) = x^2 + 4y^4 + 2y^2.$$

Try two initial points $x_0 = (-1, 2)^T$ and $x_0 = (-1, -0.25)^T$ and set the tolerance `tol = 1e-4`. Plot the iterates over the function contours. State your choice of any relevant parameters. **[2pt]**

(c) What is the main limitation of Polar-Ribieṙe method, do you observe it or any other problems in optimisation in (b), can you explain? **[2pt]**

(d) Can global convergence be guaranteed for this problem or not and why? If not, can you modify your Polar-Ribieṙe solver to guarantee global convergence and how? ***Hint:*** *What are the conditions under which global convergence can be proven for Polar-Ribieṙe?* Implement the modification, rerun your solver and compare its behaviour to the one in (b). Paraphrase the relevant theoretical results. **[5pt]**

(e) Investigate convergence of the Polar-Ribieṙe method in (b) a posteriori and include one relevant error plot. What are the empirical convergence rates and how did you obtain them? Do they agree with the theoretical predictions? Paraphrase the relevant theoretical results. **[4pt]**

EXERCISE 5 [25pt]

(a) Implement the BFGS method by modifying the `descentLineSearch.m` function from tutorial 2. Make your implementation efficient i.e. avoid explicitly forming the inverse Hessian matrix H_k. Copy the code lines implementing the update of H_k into your report and briefly explain what makes your implementation efficient. **[6pt]**

(b) Minimise the function $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x, y) = (x - 3y)^2 + x^4$$

using BFGS implemented in (a) starting from $x_0 = (10, 10)^T$. Visualise the path traced by the iterates over the function contour. Which line search did you choose and why? State your choices of the relevant line search parameters. **[2pt]**

(c) Investigate convergence of the BFGS iterates in (b) a posteriori and include one relevant error plot. What is the empirical convergence rate and how did you obtain it? Does it agree with the theoretical prediction? Paraphrase the relevant theoretical result. **[4pt]**

(d) Can global convergence of BFGS and of what type be expected or not, and why? Paraphrase the relevant theoretical result. **[3pt]**

(e*) For comparison minimise the function f with the provided SR-1 method `trustRegion_SR1.m`, which itself calls `solverCM2dsubspaceExt.m` from tutorial 3. It is a simple small scale non-efficient implementation which explicitly constructs the inverse Hessian approximation. To call `trustRegion_SR1.m`, define your function structure `F` and make sure to remove the Hessian field `F.d2f`

```
Fsr1 = rmfield(F,'d2f');
[xSR1, fSR1, nIterSR1, infoSR1] = trustRegion_SR1(Fsr1, x0, @solverCM2dSubspaceExt,
Delta, eta, tol, maxIter,0);
```

The provided SR-1 implementation has an option (which is active) to return a sequence of Hessian approximations as a field of the `info` structure. Do the same with the inverse Hessian in your BFGS implementation aka extract:

(i) $\{H_k^{\mathrm{BFGS}}\}_{k\geq 0}$ when using BFGS,

(ii) $\{B_k^{\mathrm{SR1}}\}_{k\geq 0}$ when using SR-1.

Investigate the quality of the respective (inverse) Hessian approximations computed by BFGS and SR-1. In particular, plot

(i) $\{||I - H_k^{\mathrm{BFGS}}\nabla^2 f(x_k)||_2\}_{k\geq 0}$, $\{||I - (\nabla^2 f(x_k))^{-1}B_k^{\mathrm{SR1}}||_2\}_{k\geq 0}$.

(ii) $\{||B_k^{\mathrm{SR1}} - \nabla^2 f(x_k)||_2\}_{k\geq 0}$, $\{||(H_k^{\mathrm{BFGS}})^{-1} - \nabla^2 f(x_k)||_2\}_{k\geq 0}$,

What do you observe? Do your observations agree with theoretical predictions? Paraphrase the relevant theoretical results. **[10pt]**