University of Mannheim
Scientific Computing, B6 26, C306, 68131 Mannheim
Dr. Andreas Sommer (ansommer@mail.uni-mannheim.de)

Nonlinear Optimization (FSS 2023)

Exercise Sheet #10

Due on 14.05.2023 (before 13:00).

---

1. **LICQ and linear constraints** [4 Points]

   Consider a constrained minimization problem with a feasible set X and a point $x \in X$.

   (a) Construct an example of a feasible set and a feasible point $x \in X$ at which the LICQ are satisfied, but the constraints are nonlinear.

   (b) Construct an example where the active constraints in a feasible point $x \in X$ are linear, but the LICQ is not satisfied in that point x.

2. **Penalty methods** [4 Points]

   Consider the equality constrained problem

   $$\min x_1^2 + x_2^2 \qquad \text{s.t. } x_2 = 1$$

   Determine the minimizer of the quadratic penalty function (formula 12.1) $P_\alpha(x)$ and calculate the condition number of its Hessian $\nabla^2 P_\alpha(x)$ for $\alpha \to \infty$.

   Describe the numerical problems that arise when using Newton's method to minimize the penalty function.

please turn over ...

3. Programming: Local SQP method I [10 Points]

Implement algorithm (14), the local SQP method I for equality constrained problems of the form

$$\min f(x) \quad \text{s.t. } h(x) = 0$$

Use a function header

```
function [X, MU, info] = localSQP(gradL, hessL, h, Jh, x0, mu0, tol, maxit)
```

where `gradL` and `hessL` are functions that return for given x and μ the gradient $\nabla_x \mathcal{L}(x, \mu)$ and the Hessian $\nabla^2_{xx}\mathcal{L}(x, \mu)$ of the Lagrangian function. Further, `h` and `Jh` are functions which return for given x the constraint function value $h(x)$ and its Jacobian $J_h(x) = \nabla h(x)^\top$. The input variables `x0` and `mu0` denote the initial guesses for the variables and Lagrangian multipliers.

The iteration should be terminated if the pair (x^k, μ^k) is identified as KKT-tuple (up to a given tolerance `tol`), or if a maximum number of iterations `maxit` has been reached.

The output variable `X` is a matrix containing all iterates x^k as columns; `MU` is a matrix containing all Lagrangian multipliers μ^k as columns. The flag `info` shall return 0 if a KKT point is found and -1 otherwise.

Test your implementation on the following equality constrained minimization problem:

$$\begin{aligned} \min \quad & e^{x_1 x_2 x_3 x_4 x_5} - \tfrac{1}{2}\left(x_1^3 + x_2^3 + 1\right)^2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ & x_2 x_3 - 5 x_4 x_5 = 0 \\ & x_1^3 + x_2^3 + 1 = 0 \end{aligned}$$

Use as initial guess $x^0 = (-1.8, 1.4, 1.9, -0.8, -0.8)^\top$ and $\mu^0 = 0 \in \mathbb{R}^3$.
The solution is approximately $\tilde{x} = (-1.7171, 1.5957, 1.8272, -0.76364, -0.76364)^\top$.

Use the Matlab function `fmincon` to solve the above problem and compare the solution and the multipliers.

Hint: A suitable call to `fmincon` *might look as follows:*

```
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(f, x0, [], [], [], [], [], [], c)
```

where `c` *is a suitably built constraint function (see the documentation of* `fmincon`*).*

Put all files into a single zip-archive, named by the lexicographically ordered family names of all group members separated by a hyphen, and send the zip-file to ansommer@mail.uni-mannheim.de. Please add printouts (PDF) from code to your submissions.
Avoid obvious inefficiencies like repeated evaluation of identical/unchanged expressions.
Comment your code intensely. Use a complete header that describes input and output arguments and also comment the implementation where appropriate *(see the examples at the course web site).*