Prof. Dr. Karsten Urban
M. Sc. Lewin Ernst

Winter term 2024/2025
Sheet 09, 22 Points

Numerical Optimization
Solution to exercise sheet

review on 18.12.2024 during the exercise class

1. *(Null Space Method)*
Let $A \in \mathbb{R}^{n \times n}$ be a symmetric and positive semidefinite matrix, $B \in \mathbb{R}^{m \times n}$ and $g \in \mathbb{R}^m$ with $m \leq n$. Consider the following equality constrained optimization problem

$$\text{minimize } f(x) = \frac{1}{2} x^T A x + a^T x \quad \text{subject to} \quad Bx = b. \tag{1}$$

a) Let $Z \in \mathbb{R}^{n \times (n-m)}$ be a null space matrix of B, that means that the columns of Z are a basis for the null space of the matrix B. Prove: If $\text{rg}(B) = m$ and $Z^T A Z$ is positive definite, then the matrix

$$K = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}$$

is invertible. Especially there exists a unique pair $(x^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^m$, which is the solution of the saddlepoint problem

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} -a \\ b \end{pmatrix}. \tag{2}$$

b) Let the assumption of subtask a) be fullfilled. Prove that the unique solution of (2) is the unique solution of (1).

$(4 + 4 = 8$ Points)

Solution:

a) Let $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m$ be arbitrary such that:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3}$$

From the second equation we deduce $Bx = 0$, from which $x \in \text{kern}(B)$ follows. Therefore, there exists a vector $w \in \mathbb{R}^{n-m}$ with $x = Zw$, because the columns of Z are a basis of $\text{kern}(B)$. From $Bx = 0$ it follows also $x^T B^T = 0$. Therefore

$$0 = \begin{pmatrix} x \\ \lambda \end{pmatrix}^T \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = x^T A x = w^T Z^T A Z w.$$

Due to the assumption that $Z^T A Z$ is positive definite, $w = 0$ follows directly. From that we can deduce $x = Zw = 0$. It is left to prove that $\lambda = 0$. From the first equality of (3) it follows with $x = 0$ that $B^T \lambda = 0$. Due to the full rank of B and therefore its surjectivity, we have that B^T is injective and so $\lambda = 0$ must hold.

b) The function f is convex, because the Hessian $\nabla^2 f = A$ is positive semi-definite. The constraints are affine linear, so that the optimization problem is convex and every solution of the KKT-system is a global solution of the optimization problem. The claim follows immediately, because the KKT system is given in (2).

2. *(Null Space Method, (*MATLAB*))*
Consider the linear quadratic equality constrained optimization problem (1).

a) Apply the Matlab function **nullspace_method.m** (given in the material) to the equality constrained optimization problem (1) using

$$A := \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad a := \begin{pmatrix} 0 \\ -2 \\ -2 \\ -1 \\ -1 \end{pmatrix}, \quad B := \begin{pmatrix} 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & -2 \\ 0 & 1 & 0 & 0 & -1 \end{pmatrix}, \quad b := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

b) Adjust the method **nullspace_method.m** to **nonlinear_nullspace_method.m** such that general nonlinear functions $f : \mathbb{R}^n \to \mathbb{R}$ can be optimized.

c) Apply the method **nonlinear_nullspace_method.m** to the minimization problem given in a) by using the initial value $z^{(0)} = (1, 1)^T$.

(6 Points)

Solution:

a) The MATLAB-function could look like

```
clear, close all
clc

% Initializiation
B=[1 3 0 0 0; 0 0 1 1 -2; 0 1 0 0 -1];
b=[0;0;0];
a=[0;-2;-2;-1;-1];
A=[1 -1 0 0 0;-1 2 1 0 0; 0 1 1 0 0; 0 0 0 1 0; 0 0 0 0 1];
z0 = ones(2,1);

% Define the nonlinear function
f = @(x) 1/2*x'*A*x + a'*x;

% Anwendung des Nullraum-Verfahrens
[x, lambda] = nullspace_method(A, a, B, b);
[x_nonlin]  = nonlinear_nullspace_method(f, z0, B, b);

% Berechnung des Fehlers in der euklidischen Norm
x_exact = 1/43*[-33; 11; 27; -5; 11];
error = norm(x - x_exact);
display(['Fehler zwischen exaktem und numerischem Ergebnis: ' num2str(error)]);
error = norm(x_nonlin - x_exact);
display(['Fehler zwischen exaktem und numerischem Ergebnis: ' num2str(error)]);
```

b) The MATLAB-function could look like

```
function [x] = nonlinear_nullspace_method(f, z0, B, b)
```

```
    n=size(B,2);
    m=size(B,1);

    %-- (1) QR-decomposition and splitting
    [Q, R]=qr(B');
    R = R(1:m,1:m);
    Y = Q(1:n,1:m);
    Z = Q(1:n,m+1:n);

    %-- (2) determine x_Y
    x_Y = (R')\b;

    %-- (3) determine x_Z by minimizing f(Y x_y + Z z)
    w = Y*x_Y;
    z = fminunc(@(z)f(w + Z*z), z0, optimset('Display','off'));

    % determine the full solution
    x = w + Z*z;
end
```

3. *(Hanging chain,* MATLAB*)*
We model a chain which is fixed on both ends. To get a mathematical model we consider the chain as point masses glued together by springs. This is depicted in the following:

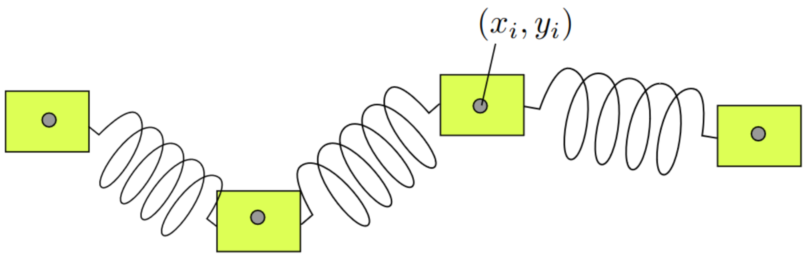

Figure 1: Chain modeled as point masses and springs.

The total energy of the system is given by $E_{\text{tot}} = E_{\text{pot}} + E_{\text{spring}}$, where $E_{\text{pot}} = mgh$ is the potential energy of the mass m with gravity $g = 9.81m/s^2$ at a height of h and $E_{\text{spring}} = \frac{1}{2}D\left((x_i - x_{i+1})^2 + (y_i - y_{i+1})^2\right)$ is the energy of a spring with spring constant D tensioned between the points (x_i, y_i) and (x_{i+1}, y_{i+1}). Now, nature tries to find the steady state of the system by minimizing the total energy. This means numerically, we face the problem

$$\min_{x,y} E_{\text{tot}}$$
$$\text{s.t. } h(x, y) = 0,$$

where h models where the chain is fixed. Implement and solve this problem with MATLAB. Play around with h, D and m. Plot the resulted chain.

(8 Points)

Solution: We can extend everything easily to the 3D case and decorate a christmas tree with a light chain. The MATLAB-function could look like

```
clear, close all
```

```
clc

 % Source:
 %    https://de.mathworks.com/matlabcentral/fileexchange/9337-xmas-tree
ov_xmasTree();

% points for the light chain
% coordinates on the tree
x_tree = [-0.1153, 2.986, 3.059, -4.7118, -5.0628, 5.01336, ...
           5.15578, -3.3961, -7.445, -4.9825, 5.1589];
y_tree = [-2.7033, -0.9761, 3.657, 2.729, -3.4359, -4.31467, ...
           5.4282, 7.1485, -2.6252, -7.38934, -6.8566];
z_tree = [22.657, 21.722, 18.024, 17.241, 15.1467, 13.739, ...
          11.9194, 10.994, 10.7183, 9.6119, 8.6214];

% size of equality constraints (actually a third of it)
M = size(x_tree,2);

% calculate how many lights between points
length_inbetween = zeros(1,M-1);
for i = 1:M-1
    length_inbetween(i) = sqrt(  (x_tree(i)-x_tree(i+1))^2 ...
                               + (y_tree(i)-y_tree(i+1))^2 ...
                               + (z_tree(i)-z_tree(i+1))^2);
end
total_length = sum(length_inbetween);



N = 91;
n = 3*(N+1);
mass_vec = 20*ones(N+1,1);
C_vec    = 10*N*ones(N,1);
g        = 9.81;
f        = @(x) total_energy(x, mass_vec, C_vec, g);

% constraints
% calculate the indices of constraints (this determines how many lights
% we want to have inbetween)
index_constr = floor([0, length_inbetween]/total_length * N);
for i = 2:size(index_constr,2)
    index_constr(i) = index_constr(i-1) + index_constr(i);
end
index_constr(1)      = 1;
index_constr(end)    = N;
index_constr(end-1) = index_constr(end-1) + 5;
index_constr         = cast(index_constr, "int32");
m        = 3*M;
B        = zeros(m,n);
b        = [x_tree, y_tree, z_tree]';
% fill in the constraint matrix
for i = 1:M
    % x coordinate
    B(i,index_constr(i)) = 1;
    % y coordinate
    B(M + i, N + 1 + index_constr(i)) = 1;
    % z coordinate
    B(2*M + i, 2*N + 2 + index_constr(i)) = 1;
end


% initial value (not needed with the other algorithm)
```

```
z0  = ones(n-m,1);
[x] = nonlinear_nullspace_method(f, z0, B, b);

hold on
figure(1)
plot3(x(1:N+1),x(N+2:2*N+2),x(2*N+3:3*N+3), 'r-o', ...
      LineWidth=1, MarkerSize=12, MarkerFaceColor='auto')
```

The results are given in the following figure

Figure 2: A christmas tree with decorated with a light chain, obtained by the quadratic optimization algorithm nullspace method. The white chain goes around the christmas tree.

Merry Christmas and a Happy New Year