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

Winter term 2024/2025
Sheet 08, 30 Points

Numerical Optimization
Solution to exercise sheet

review on 11.12.2024 during the exercise class

1. *(Linear programming and KKT)*
We are going to show necessary and sufficient conditions for linear problems. From the Optimization lecture we know that every linear program can be written as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & \; c^T x \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned} \tag{1}$$

which has the associated dual problem

$$\begin{aligned} \max_{(\lambda,\mu) \in \mathbb{R}^m \times \mathbb{R}^n} & \; b^T \lambda \\ \text{s.t. } & A^T \lambda + \mu = c \\ & \mu \geq 0. \end{aligned} \tag{2}$$

Note that $\mu = c - A^T \lambda$ can be seen as a slack variable. For those linear problems, one can show

Theorem 1 (Necessary and sufficient optimality conditions)**.** *The following statements are equivalent:*

i) The primal problem (1) *has a solution* $\bar{x}$*.*

ii) The dual problem (2) *has a solution* $(\bar{\lambda}, \bar{\mu})$*.*

iii) The optimality system

$$A^T \lambda + \mu = c, \quad Ax = b, \quad x^T \mu = 0, \quad x \geq 0, \quad \mu \geq 0 \tag{3}$$

has a solution $(\bar{x}, \bar{\lambda}, \bar{\mu})$*.*

The tasks are:

a) Derive the dual problem (2) from the primal problem (1) with the definition of the dual problem (D) and the dual feasible region $\mathcal{F}_D$ of Section 3.2.1 in the lecture notes.

b) Proof Theorem 1.
Hint: Some hints are in order:
   - The saddle point property (Theorem 3.2.3) is important.
   - Show ACQ for such linear programs (you only have to prove it for the primal problem).
   - Linear programs are convex, so you may use Theorem 3.1.20 (KKT for convex problems) in some way or another.
   - How do the KKT conditions for problem (1) and problem (2) look like?

(14 Points)

Solution:

a) We first derive the dual problem from the primal problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} & c^T x \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$

By definition, the dual objective function is given by

$$\varphi(\lambda,\mu) = \inf_{x\in\mathbb{R}^n} \mathcal{L}(x,\lambda,\mu)$$

and the dual feasable region by

$$\mathcal{F}_D = \{(\lambda,\mu) \in \mathbb{R}^m \times \mathbb{R}^p_+ : \varphi(\lambda,\mu) > -\infty\}.$$

Then we have define the dual problem as

$$\begin{cases} \varphi(\lambda,\mu) \longrightarrow \max, & (\lambda,\mu) \in \mathbb{R}^m \times \mathbb{R}^p, \\ \text{s.t.}(\lambda,\mu) \in \mathcal{F}_D. \end{cases}$$

To get explicit we write down the Lagrange function

$$\begin{aligned} \mathcal{L}(x,\lambda,\mu) &= c^T x + \lambda^T(b - Ax) - \mu^T x \\ &= c^T x - (A^T\lambda)^T x + \lambda^T b - \mu^T x \\ &= x^T(c - A^T\lambda - \mu) + \lambda^T b \end{aligned}$$

Now we can look at the dual objective function

$$\varphi(\lambda,\mu) = \inf_{x\in\mathbb{R}^n}\{x^T(c - A^T\lambda - \mu)\} + b^T\lambda = \begin{cases} b^T\lambda, & (\lambda,\mu) \in \mathcal{F}_D \\ -\infty, & \text{else} \end{cases}.$$

We see that the dual feasible set is given by

$$\mathcal{F}_D := \{(\lambda,\mu) \in \mathbb{R}^m \times \mathbb{R}^p_+ : (c - A^T\lambda - \mu) = 0\}.$$

From that we deduce the dual problem

$$\begin{aligned} \max_{(\lambda,\mu)\in\mathbb{R}^m\times\mathbb{R}^n} & \quad b^T\lambda \\ \text{s.t. } & A^T\lambda + \mu = c \\ & \mu \geq 0. \end{aligned}$$

b) To show equivalence of the statements we start by showing iii) $\Rightarrow$ i), ii) with Theorem 3.2.3. Let $(\bar{x},\bar{\lambda},\bar{\mu})$ solve the optimality system (3). We show that this is a saddle point for $\mathcal{L}$. Indeed, we have

$$\begin{aligned} \mathcal{L}(\bar{x},\bar{\lambda},\bar{\mu}) &= (\bar{x})^T(c - A^T\bar{\lambda} - \bar{\mu}) - (\bar{\lambda})^T b \\ &\overset{c=A^T\bar{\lambda}+\bar{\mu}}{=} -b^T\bar{\lambda} \overset{\forall x\in\mathbb{R}^n}{\leq} -b^T\bar{\lambda} = \mathcal{L}(x,\bar{\lambda},\bar{\mu}) \end{aligned}$$

and on the other hand

$$\begin{aligned}
\mathcal{L}(\bar{x}, \bar{\lambda}, \bar{\mu}) &= c^T \bar{x} + \bar{\lambda}^T (\underbrace{b - A\bar{x}}_{=0}) - \bar{\mu}^T \bar{x} \\
&= c^T \bar{x} - \underbrace{\bar{\mu}^T \bar{x}}_{=0} \\
&\overset{\forall \mu \in \mathbb{R}^n_+}{\leq} c^T \bar{x} - \mu^T \bar{x} = \mathcal{L}(\bar{x}, \lambda, \mu).
\end{aligned}$$

We have the saddle point property and so $\bar{x}$ is the minimizer of the primal problem and $(\bar{\lambda}, \bar{\mu})$ is the maximizer of the dual problem. Now we show that ACQ is always fulfilled for linear programs and that the conditions in iii) are the KKT conditions for (1). Therefore with Theorem 3.1.20 b) we conclude i) $\Rightarrow$ iii).
For ACQ to hold we have to prove $\mathcal{T}_{\text{lin}}(\bar{x}, \mathcal{F}_P) \subseteq \mathcal{T}(\bar{x}, \mathcal{F})$, the other direction follows from Lemma 3.1.10. The linearized tangential cone is given by

$$\mathcal{T}_{\text{lin}}(\bar{x}, \mathcal{F}_P) = \{d \in \mathbb{R}^n : -(d_j)_{j \in \mathcal{A}(\bar{x})} \leq 0,\ Ad = 0\}.$$

Let $d \in \mathcal{T}_{\text{lin}}(\bar{x}, \mathcal{F}_P)$. We define the sequences $(\eta^{(\ell)})_{\ell \in \mathbb{N}}$ by $\eta^{(\ell)} := \ell$ and $(x^{(\ell)})_{\ell \in \mathbb{N}}$ by $x^{(\ell)} := \bar{x} + \frac{1}{\ell} d$. Therefore, we have $x^{(\ell)} \to \bar{x}$ as $\ell \to \infty$. So what is left to show is $(x^{(\ell)})_{\ell \in \mathbb{N}} \subset \mathcal{F}_P$. We get

$$Ax^{(\ell)} = A(\bar{x} + \frac{1}{\ell} d) = A\bar{x} = b$$

and for ℓ large enough we have

$$x^{(\ell)} = \bar{x} + \frac{1}{\ell} d \geq 0.$$

This follows from the fact that

$$\bar{x} \geq 0 \quad \text{and} \quad (d_j)_{j \in \mathcal{A}(\bar{x})} \geq 0$$

and for the inactive indices $i \in \mathcal{I}(\bar{x})$ we have $\bar{x}_i > 0$, so that for ℓ large enough we get $(\bar{x}_i)_{i \in \mathcal{I}(\bar{x})} + \frac{1}{\ell}(d_i)_{i \in \mathcal{I}(\bar{x})} > 0$. We conclude $(x^{(\ell)})_{\ell \in \mathbb{N}} \subset \mathcal{F}_P$ and $\mathcal{T}_{\text{lin}}(\bar{x}, \mathcal{F}_P) \subseteq \mathcal{T}(\bar{x}, \mathcal{F}_P)$. Note that we have not used that $\bar{x}$ is a local solution, therefore ACQ holds for all $x \in \mathcal{F}_P$.
To prove that (3) are the KKT conditions for (1) we just calculate them. They are given by

$$\begin{aligned}
\underbrace{c^T}_{\nabla f(x)} + \lambda^T \underbrace{A}_{\nabla h(x)} + \mu^T \underbrace{(-I)}_{\nabla g(x)} = 0 \quad &\Leftrightarrow \quad c + A^T \lambda = \mu \\
Ax - b = 0 \\
\mu \geq 0, \quad -x \leq 0, \quad \mu^T x = 0
\end{aligned}$$

With that we can apply Theorem 3.1.20 and conclude i) $\Rightarrow$ iii) and so i) $\Rightarrow$ ii) follows also.
The only thing left to prove is ii) $\Rightarrow$ iii). This follows directly because the KKT system of the dual problem is the same as for the primal problem with exchanged roles of Langrange multipliers, i.e. for the dual problem (λ, μ) are the variables we seek for and x is the Lagrange multiplier. Due to the fact that ACQ also holds for the dual problem, as it is a linear program which can be written in form of (1), Theorem 3.1.20 again implies iii). This proves the equivalence of the statements.

2. *(Penalty Method)*
Consider for $f \in C(\mathbb{R}^n)$ and $h \in C(\mathbb{R}^n; \mathbb{R}^m)$ the constrained optimization problem

$$\begin{cases} \min & f(x) \\ \text{s.t.} & h(x) = 0 \end{cases} \tag{4}$$

with

$$\mathcal{F} := \{x \in \mathbb{R}^n : h(x) = 0\} \neq \emptyset.$$

Let $(\tau_k)_{k\in\mathbb{N}} \subset \mathbb{R}_{>0}$ be a strictly monotonically increasing sequence with $\lim_{k\to\infty} \tau_k = \infty$. Prove that for a sequence $(x^{(k)})_{k\in\mathbb{N}}$ generated by the penalty method, it holds:

a) The sequence $(\mathcal{L}_{\tau_k}(x^{(k)}))_{k\in\mathbb{N}}$ is monotonically increasing.
b) The sequence $(\|h(x^{(k)})\|)_{k\in\mathbb{N}}$ is monotonically decreasing.
c) The sequence $(f(x^{(k)}))_{k\in\mathbb{N}}$ is monotonically increasing.
d) It holds $\lim_{k\to\infty} h(x^{(k)}) = 0$.
e) Every accumulation point of $(x^{(k)})_{k\in\mathbb{N}}$ is a (global) solution of (4).

(8 Points)

Solution:

a) We recall the *penalty Lagrange function* $\mathcal{L}_\tau$ from the lecture notes

$$\mathcal{L}_\tau(x) = f(x) + \frac{1}{2}\tau||h(x)||^2,$$

and for τ_k, we pick $x^{(k)} \in \mathbb{R}^n$ such that

$$\mathcal{L}_{\tau_k}(x^{(k)}) = \inf_{x\in\mathbb{R}^n} \mathcal{L}_{\tau_k}(x),$$

which leads to the statement by

$$\mathcal{L}_{\tau_k}(x^{(k)}) \leq \mathcal{L}_{\tau_k}(x^{(k+1)}) \leq \mathcal{L}_{\tau_{k+1}}(x^{(k+1)}), \qquad k \in \mathbb{N}.$$

b) Furthermore, it can be expressed that

$$\mathcal{L}_{\tau_k}(x^{(k)}) \leq \mathcal{L}_{\tau_k}(x^{(k+1)}) \quad \text{and } \mathcal{L}_{\tau_{k+1}}(x^{(k+1)}) \leq \mathcal{L}_{\tau_{k+1}}(x^{(k)}),$$

and therefore

$$\begin{aligned} &\mathcal{L}_{\tau_k}(x^{(k)}) + \mathcal{L}_{\tau_{k+1}}(x^{(k+1)}) - \mathcal{L}_{\tau_k}(x^{(k+1)}) - \mathcal{L}_{\tau_{k+1}}(x^{(k)}) \leq 0 \\ \iff &\underbrace{(\tau_k - \tau_{k+1})}_{<0}\left(||h(x^{(k)})||^2 - ||h(x^{(k+1)})||^2\right) \leq 0. \end{aligned}$$

It must hold

$$||h(x^{(k)})|| \geq ||h(x^{(k+1)})||, \qquad k \in \mathbb{N}.$$

c) By starting with the same argument

$$\mathcal{L}_{\tau_k}(x^{(k)}) \leq \mathcal{L}_{\tau_k}(x^{(k+1)}),$$

we get

$$\begin{aligned} &f(x^{(k)}) + \frac{1}{2}\tau_k||h(x^{(k)})||^2 \leq f(x^{(k+1)}) + \frac{1}{2}\tau_k||h(x^{(k+1)})||^2 \\ \iff &f(x^{(k+1)}) - f(x^{(k)}) \geq \frac{1}{2}\tau_k\left(||h(x)^{(k)}||^2 - ||h(x^{(k+1)})||^2\right). \end{aligned}$$

From part b) and from the properties of $(\tau_k)_{k\in\mathbb{N}}$, we can further state that

$$\begin{aligned} f(x^{(k+1)}) - f(x^{(k)}) &\geq \frac{1}{2}\tau_k\left(||h(x)^{(k)}||^2 - ||h(x^{(k+1)})||^2\right) \geq 0 \\ \Rightarrow f(x^{(k+1)}) &\geq f(x^{(k)}). \end{aligned}$$

d) Let $\tilde{x} \in \mathcal{F}$, then

$$\infty > f(\tilde{x}) =: S \geq \inf_{x \in \mathcal{F}} f(x)$$

and for the penalty Lagrange function follows

$$\mathcal{L}_{\tau_k}(x^{(k)}) = \inf_{x \in \mathbb{R}^n} \mathcal{L}_{\tau_k}(x) \leq \inf_{x \in \mathcal{F}} \mathcal{L}_{\tau_k}(x) = \inf_{x \in \mathcal{F}} f(x) \leq S. \tag{5}$$

Let's apply part c), so that

$$f(x^{(0)}) + \frac{1}{2}\tau_k ||h(x^{(k)})||^2 \leq f(x^{(k)}) + \frac{1}{2}\tau_k ||h(x^{(k)})||^2 = \mathcal{L}_{\tau_k}(x^{(k)}) \leq S,$$

which can be rewritten to

$$||h(x^{(k)})|| \leq \sqrt{\frac{2}{\tau_k}(\underbrace{S - f(x^{(0)})}_{\geq 0})}.$$

And since the right side converges to zero for $k \to \infty$, the left side also has to follow:

$$0 \leq \lim_{k \to \infty} ||h(x^{(k)})|| \leq \lim_{k \to \infty} \sqrt{\tfrac{2}{\tau_k}(S - f(x^{(0)}))} = 0 \quad \Rightarrow \lim_{k \to \infty} ||h(x^{(k)})|| = 0.$$

e) Let $\bar{x}$ be an accumulation point of $(x^{(k)})_{k \in \mathbb{N}}$. We have a subsequence $(x^{(k_l)})_{l \in \mathbb{N}}$ with

$$\lim_{l \to \infty} x^{(k_l)} = \bar{x}.$$

By part d) and continuity of the norm and of h it holds

$$||h(\bar{x})|| = ||h(\lim_{l \to \infty} x^{(k_l)})|| = ||\lim_{l \to \infty} h(x^{(k_l)})|| = \lim_{l \to \infty} ||h(x^{(k_l)})|| = 0,$$

and it follows that

$$\bar{x} \in \mathcal{F}.$$

Furthermore, it holds with (5)

$$\inf_{x \in \mathcal{F}} f(x) \leq f(\bar{x}) = f(\lim_{l \to \infty} x^{(k_l)}) = \lim_{l \to \infty} f(x^{(k_l)}) \leq \lim_{l \to \infty} \mathcal{L}_{\tau_{k_l}}(x^{(k_l)}) \leq \inf_{x \in \mathcal{F}} f(x),$$

and with the sandwich theorem

$$\lim_{l \to \infty} f(x^{(k_l)}) = \inf_{x \in \mathcal{F}} f(x).$$

3. *(Penalty Method and Lagrange Method)*

a) Write a MATLAB routine

```
x = penalty_method(f, h, x0, tau0, rho, tol, maxIt)
```

which performs the penalty method with a suitable termination criteria. The input parameter are given by:

- `f`: The objective function $f : \mathbb{R}^n \to \mathbb{R}$ (function handle).
- `h`: The constraint function $h : \mathbb{R}^n \to \mathbb{R}^m$ with $m \leq n$ (function handle).
- `x0`: Initial value $x^{(0)} \in \mathbb{R}^n$.
- `tau0`: Initial penalty parameter τ_0.

- `rho`: Parameter for increasing the penalty parameter τ.
- `tol`: Tolerance for termination criterion.
- `maxIt`: Maximal number of iterations for termination criterion.

The output parameter is given by:

- `x`: Vector containing the evolution of all solutions computed by the penalty method.

For minimizing the penalty Lagrange function $\mathcal{L}_\tau : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\mathcal{L}_\tau(x) = f(x) + \frac{1}{2}\tau\|h(x)\|^2$$

you can use `fminsearch` or `fminuc`.

b) Write a MATLAB routine

```
x = lagrange_method(f, h, x0, lambda0, tau0, rho, tol, maxIt)
```

which performs the method of lagrange multipliers (Section 3.5 in the lecture notes) with a suitable termination criteria. In addition to the notations from subtask a) the input parameter are given by

- `lambda0`: Initial lagrange parameter $\lambda^{(0)}$.

The output parameter is given by:

- `x`: Vector containing the evolution of all solutions computed by the penalty method.

For minimizing the function $\mathcal{G}_\tau : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\mathcal{G}_\tau(x, \lambda) = f(x) + \lambda^\top h(x) + \frac{1}{2}\tau\|h(x)\|^2$$

you can use `fminsearch` or `fminuc`.

c) Apply the MATLAB routine `penalty_method.m` and `lagrange_method.m` to the function $f : \mathbb{R}^2 \to \mathbb{R}$ with

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$$

subject to the constraint $h(x_1, x_2) = 0$, where

$$h(x_1, x_2) = (x_1 + 0.5)^2 + (x_2 + 0.5)^2 - 0.25,$$

by using the initial value $x^{(0)} = (1, 1)^\top$ and the parameter $\tau^{(0)} = 0.1$, $\rho = 6$ and $\lambda^{(0)} = 10$, respectively. Create a plot displaying the contour lines of f together with the solution of each iteration of the penalty method and the method of lagrange multipliers. Discuss your results.

(8 Points)

Solution:

a) The MATLAB-function for the penalty method could look like

```
function [x] = penalty_method(f, h, x0, tau0, rho, tol, maxIt)
% Solves the constrained optimization Problem
%
%   min f(x)   s.t.   h(x) = 0
%
```

```
% with the Penalty Method.
%
% INPUT:
% f:     [function handle R^n->R]     objective function
% h:     [function handle R^n->R^m]   constraint function
% x0:    [vector R^n]                 initial value
% tau0:  [scalar > 0]                 inital penalty parameter
% rho:   [scalar > 1]                 penalty parameter control,
%                                     tau(k) = rho*tau(k-1)
% tol:   [scalar >=0]                 error tolerance
% maxIt: [integer]                    max. number of iterations
%
% OUTPUT:
% x: [matrix R^n*k]     penalty method iterations

F =@(x,tau) f(x) + tau*(h(x)'*h(x));

x0 = x0(:);
x  = zeros(length(x0),maxIt);
x(:,1) = x0;

tau = tau0;

k = 1;

while k <= maxIt

    % x(:,k+1) = fminunc(@(x)F(x,tau), x(:,k), optimset('Display','off'));
    x(:,k+1) = fminsearch(@(x)F(x,tau), x(:,k), optimset('Display','off'));

    if norm(h(x(:,k+1))) < tol && norm(x(:,k)-x(:,k+1)) < tol
        x = x(:,1:k+1);
        return
    end

    k = k + 1;
    tau = rho*tau;

end

end
```

andMatlab-function for the penalty-lagrange method could look like

```
function [x] = lagrange_method(f, h, x0, lambda0, tau0, rho, tol, maxIt)
% Solves the constrained optimization Problem
%
%   min f(x)  s.t.  h(x) = 0
%
% with the Penalty-Lagrange Method.
%
% INPUT:
% f:     [function handle R^n->R]     objective function
% h:     [function handle R^n->R^m]   constraint function
% x0:       [vector R^n]                 initial value
% lambda0: [vector R^m]               initial value lagrange param
% tau0:  [scalar > 0]                 inital penalty parameter
% rho:   [scalar > 1]                 penalty parameter control,
%                                     tau(k) = rho*tau(k-1)
% tol:   [scalar >=0]                 error tolerance
% maxIt: [integer]                    max. number of iterations
%
```

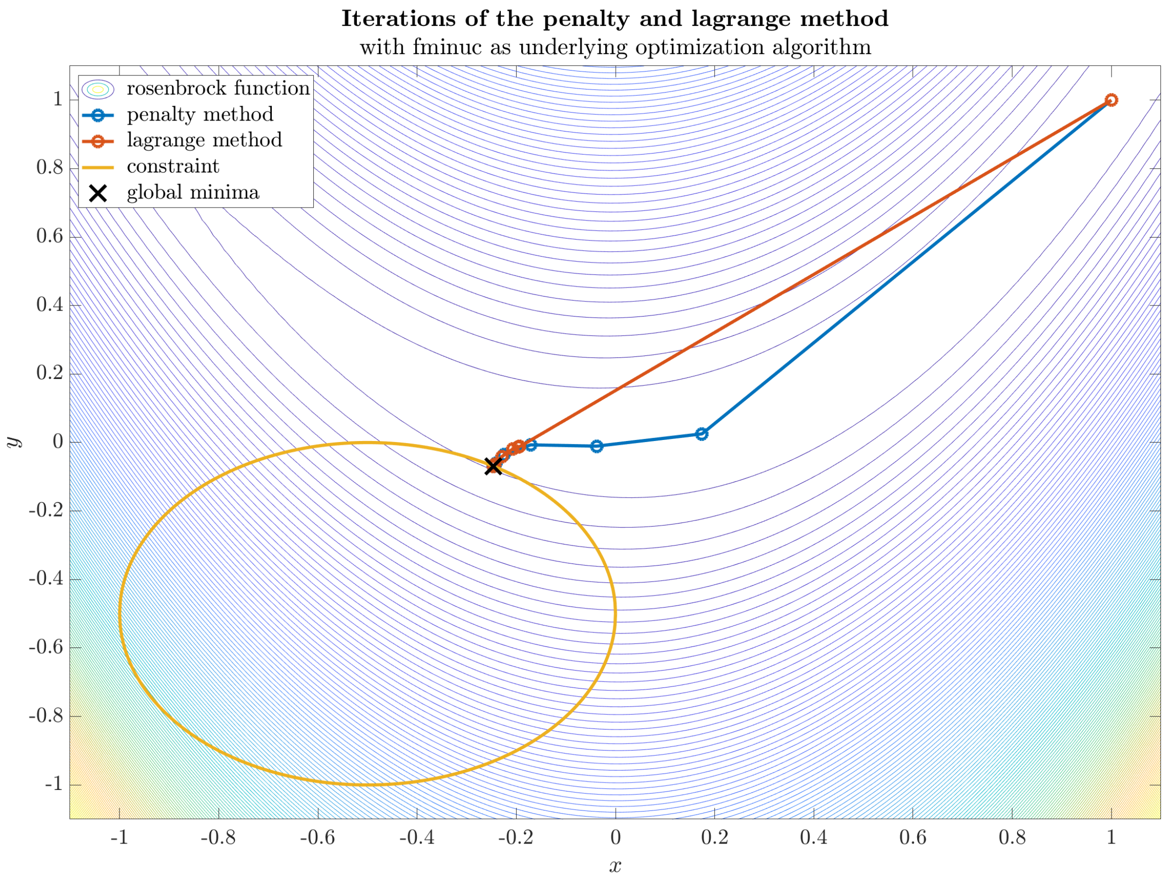
```
% OUTPUT:
% x: [matrix R^n*k]     penalty-lagrange method iterations

F =@(x,lambda,tau) f(x) + lambda'*h(x)+ 1/2*tau*(h(x)'*h(x));

x0       = x0(:);
x        = zeros(length(x0),maxIt);
x(:,1)   = x0;

tau    = tau0;
lambda = lambda0;
k      = 1;

while k <= maxIt
    % x(:,k+1) = fminunc(@(x)F(x,lambda,tau), x(:,k), optimset('Display','off'));
    x(:,k+1) = fminsearch(@(x)F(x,lambda,tau), x(:,k), optimset('Display','off'));

    if norm(h(x(:,k+1))) < tol && norm(x(:,k)-x(:,k+1)) < tol
        x = x(:,1:k+1);
        return
    end

    lambda = lambda + tau*h(x(:,k+1));
    tau    = rho*tau;
    k = k + 1;

end

end
```

The MATLAB script could be

```
clear, close all;
clc

xmax = [-1.1,1.1];
ymax = [-1.1,1.1];

% define problem
rosenbrock_sep =@(x,y) (1-x).^2 + 100*(y-x.^2).^2;
rosenbrock     =@(x) rosenbrock_sep(x(1,:),x(2,:));

h =@(x) (x(1,:) + 0.5).^2 + (x(2,:) + 0.5).^2 - 0.25;
h_aufgeloest =@(x) sqrt(0.25 - (x(1,:) + 0.5).^2) - 0.5;

x0 = [1;1];

% calculate reference solution
options = optimoptions('fmincon', 'Display','off');
x_exact = fmincon(rosenbrock,x0,[],[],[],[],[],[],@(x)h_fminsearch(h,x),options);

% solve problem
tau0 = 1;
rho  = 6;
lambda0 = 10;

maxIt = 100;
tol = 1e-9;

XP = penalty_method(rosenbrock, h, x0, tau0, rho, tol, maxIt);
XL = lagrange_method(rosenbrock, h, x0, lambda0, tau0, rho, tol, maxIt);
```

```
% print results
str_line  = ['+' repmat('=',[1 10]) '+' repmat('=',[1 18]) ...
             '+' repmat('=',[1 12]) '+' repmat('=',[1 12]) ...
             '+' repmat('=',[1 13]) '+\n'];
fprintf(str_line);
fprintf('|  method  |        x*         |    h(x*)    |    f(x*)    | #Iterations |\n')
fprintf(str_line);
fprintf('| penalty  | (%6.3f, %6.3f) | %10.3e | %10.3e |     %4u    |\n',...
    XP(1,end), XP(2,end), h(XP(:,end)), rosenbrock(XP(:,end)), size(XP,2));
fprintf(replace(str_line,"=","-"));
fprintf('| lagrange | (%6.3f, %6.3f) | %10.3e | %10.3e |     %4u    |\n',...
    XL(1,end), XL(2,end), h(XL(:,end)), rosenbrock(XL(:,end)), size(XL,2));
fprintf(replace(str_line,"=","-"));
fprintf('| fmincon  | (%6.3f, %6.3f) | %10.3e | %10.3e |      N/A     |\n',...
    x_exact(1), x_exact(2), h(x_exact), rosenbrock(x_exact));
fprintf(str_line);

% plot
[xx,yy] = meshgrid(linspace(xmax(1),xmax(2),100),linspace(ymax(1),ymax(2),250));
zz = rosenbrock_sep(xx,yy);
figure()

contour(xx,yy,zz,150,...
    'DisplayName','\, rosenbrock function');

hold on
plot(XP(1,:),XP(2,:),'-o','linewidth',3,'MarkerSize',10,...
    'DisplayName','\, penalty method');
plot(XL(1,:),XL(2,:),'-o','linewidth',3,'MarkerSize',10,...
    'DisplayName','\, lagrange method');
nb = plot(linspace(-1,0,1000),h_aufgeloest(linspace(-1,0,1000)),'linewidth',3,...
    'HandleVisibility','off');
plot(linspace(-1,0,1000),-h_aufgeloest(linspace(-1,0,1000))-1,'linewidth',3, 'color', nb.Col
    'DisplayName', '\, constraint');
plot(x_exact(1),x_exact(2),'xk','MarkerSize',20,'linewidth',5,...
    'DisplayName','\, global minima');

tit  = {'\textbf{Iterations of the penalty and lagrange method}',...
        'with fminuc as underlying optimization algorithm'};

title(tit,  'Interpreter','latex')
xlabel('$x$','Interpreter','latex')
ylabel('$y$','Interpreter','latex')
set(legend,    'Interpreter','latex','Location','northwest','FontSize',20);

set(gca,'TickLabelInterpreter', 'latex', 'FontSize',20)
set(gcf,'units','normalized','outerposition',[0 0.05 1 0.9]); % Maximize figure window

function [c, ceq] = h_fminsearch(h,x)
    c = 0;
    ceq = h(x);
end
```

The plot could look like


Iterations of the penalty and lagrange method
with fminuc as underlying optimization algorithm
rosenbrock function
penalty method
lagrange method
constraint
global minima
x
y
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1