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

Winter term 2024/2025
Sheet 04, 34 Points

Numerical Optimization
Solution to exercise sheet

review on 12.11.2024 during the exercise class

1. *(Step size too small or too big?)*
The following example shows that the step size has to be choosen very wisely. Consider the function $f : \mathbb{R} \to \mathbb{R}$ with $f(x) = x^2$ and the iteration

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)}, \qquad k \in \mathbb{N}_0,$$

where the initial value is given by $x^{(0)} = 1$.

a) Choose the search direction $p^{(k)} = -1$ and the step size $\alpha^{(k)} = 2^{-k-2}$ for $k \in \mathbb{N}_0$. Prove that $\lim_{k\to\infty} x^{(k)} = \frac{1}{2}$ and $\lim_{k\to\infty} f(x^{(k)}) = \frac{1}{4}$. Discuss this result.

b) Choose the search direction $p^{(k)} = (-1)^{k+1}$ and the step size $\alpha^{(k)} = 1 + \frac{3}{2^{k+2}}$ for $k \in \mathbb{N}_0$. Calculate $x^{(k)}$. Discuss your result.

(4 + 4 = 8 Points)

Solution:

a) For the $k+1$-th iterate there holds for the search direction $p^{(k)} = -1$ and arbitrary step size

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)} = x^{(k)} - \alpha^{(k)} = x^{(0)} - \sum_{i=0}^{k} \alpha^{(i)},$$

from which we deduce for $\alpha^{(k)} = 2^{-k-2}$

$$x^{(k+1)} = 1 - \frac{1}{4} \sum_{i=0}^{k} \left(\frac{1}{2}\right)^i = \frac{1}{2} + \left(\frac{1}{2}\right)^{k+2} \tag{1}$$

We notice that $0 < x^{(k+1)} < x^{(k)}$ and so $f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right)$ (because of the monoticity of f for $x \geq 0$), i.e. there actually is a decrease. But with (1) for the $k+1$-th iterate we have

$$\lim_{k\to\infty} x^{(k)} = \lim_{k\to\infty} \left(\frac{1}{2} + \left(\frac{1}{2}\right)^{k+2}\right) = \frac{1}{2}$$

and due to the continuity of f we get

$$\lim_{k\to\infty} f(x^{(k)}) = f(\lim_{k\to\infty} x^{(k)}) = f(1/2) = \frac{1}{4}.$$

The minimum of f is attained at $x^* = 0$ with the minimum value $f(x^*) = 0$. The step size has been chosen too small.

b) We deduce analogously to a) with the search direction $d^{(k)} := (-1)^{k+1}$ and the step size $\alpha^{(k)} := 1 + \frac{3}{2^{k+2}}$ for the k-th iterate

$$x^{(k)} = \frac{(-1)^k}{2}\left(1 + \frac{1}{2^k}\right). \tag{2}$$

With (2) we notice that, the sequence $(x^{(k)})_{k\in\mathbb{N}}$ has two accumulation points, namely

$$\lim_{k\to\infty} x^{(2k)} = \lim_{k\to\infty} \frac{1}{2}\left(1 + \frac{1}{2^{2k}}\right) = \frac{1}{2} \qquad \text{and} \qquad \lim_{k\to\infty} x^{(2k-1)} = \lim_{k\to\infty} \frac{-1}{2}\left(1 + \frac{1}{2^{2k-1}}\right) = -\frac{1}{2}.$$

Due to the continuity of f we have

$$\lim_{k\to\infty} f(x^{(2k)}) = \lim_{k\to\infty} f(x^{(2k-1)}) = \frac{1}{4}.$$

The step size has been chosen too large.

2. *(Wolfe conditions)*
Typical line search algorithms try out a sequence of candidate values for the step size α, stopping to accept one of these values when certain conditions are satisfied. We want to take a closer look at the Wolfe conditions which consist of

1.) a sufficient decrease condition of the objective function f (also known as *Armijo condition*)

$$f(x + \alpha p) \leq f(x) + c_1 \alpha \nabla f(x)^T p, \qquad c_1 \in (0, 1) \tag{3}$$

and

2.) the curvature condition

$$\nabla f(x + \alpha p)^T p \geq c_2 \nabla f(x)^T p, \qquad c_2 \in (c_1, 1) \tag{4}$$

where f is the objective function, p is a descent direction and α is the step size.

a) Show that if $0 < c_2 < c_1 < 1$, there may be no step lengths that satisfy the Wolfe conditions.

b) Prove Lemma 2.3.10.

c) Prove that the quadratic function $\Phi_1^{(k)}$ that interpolates

$$\Phi_1^{(k)}(0) = \Phi^{(k)}(0), \quad \left(\Phi_1^{(k)}\right)'(0) = \left(\Phi^{(k)}\right)'(0) \quad \text{and} \quad \Phi_1^{(k)}(\alpha_0^{(k)}) = \Phi^{(k)}(\alpha_0^{(k)})$$

is given by $\Phi_1^{(k)}$ in Algorithm 2.3.3 Line 5. Then, make use of the fact that the Armijo's condition is not satisfied at α_0 to show that the quadratic $\Phi_1^{(k)}$ has positive curvature and the minimizer satisfies

$$\alpha_1 < \frac{\alpha_0}{2(1-c_1)}.$$

Remark: Since $c_1 \approx 10^{-4}$, this inequality gives us an idea of the new step length, in case of acceptance.

$(4 + 5 + 5 = 14$ Points$)$

Solution:

a) We construct a quadratic function, $\Phi(x) = a\alpha^2 + b\alpha + c$ with $\Phi(0) = 1$ and $\Phi'(0) = 1$ which results in $c = 1$ and $b = -1$. $a > 0$ is rather arbitrary. We fix the function $\Phi(\alpha) = 2\alpha^2 - \alpha + 1$. The function for the Armijo condition is given by $\Phi_{\text{lin}}(\alpha) = \Phi(0) + c_1\Phi'(0)\alpha = 1 - c_1\alpha$. We search for the intersection point, hence

$$\begin{aligned} &\Phi(\alpha) = \Phi_{\text{lin}}(\alpha) \\ \Leftrightarrow\ &\alpha(2\alpha - 1 + c_1) = 0 \\ \Leftrightarrow\ &\alpha_1 = 0, \quad \alpha_2 = 1/2(1 - c_1) \end{aligned}$$

We choose $c_1 := \frac{18}{20} \in (0, 1)$ with which we get $\alpha \in (0, \frac{1}{20})$ from the Armijo condition and $c_2 \in (0, \frac{18}{20})$. The curvature condition requires

$$\begin{aligned} \Phi'(\alpha) &\geq c_2\Phi'(0) = -c_2 \\ 4\alpha - 1 &\geq -c_2, \end{aligned}$$

but for $\alpha \in (0, \frac{1}{20})$ we have $4\alpha - 1 \in (-1, -\frac{4}{5})$, so that for $c_2 \in (0, \frac{4}{5}] \subset (0, c_1)$ it follows $4\alpha - 1 < -c_2$. Therefore the curvature condition can never be satisfied.

b) Note that $\Phi(\alpha) = f(x^{(k)} + \alpha p^{(k)})$ is bounded from below for all $\alpha > 0$. Since $0 < c_1 < 1$, the line $l(\alpha) = f(x^{(k)}) + \alpha c_1 \nabla f(x^{(k)})^T p^{(k)}$ is unbounded below and must therefore intersect the graph of Φ at least once. Let $\alpha' > 0$ be the smallest intersecting value among all intersecting α's, that is,

$$f(x^{(k)} + \alpha' p^{(k)}) = f(x^{(k)}) + \alpha' c_1 \nabla f(x^{(k)})^T p^{(k)}.$$

Armijo's condition clearly holds for all step lengths less than α'. By the mean value theorem, there is $\alpha'' \in (0, \alpha')$ such that

$$f(x^{(k)} + \alpha' p^{(k)}) - f(x^{(k)}) = \alpha' \nabla f(x^{(k)} + \alpha'' p^{(k)})^T p^{(k)}$$

By combining the latter two equations, we obtain

$$\nabla f(x^{(k)} + \alpha'' p^{(k)})^T p^{(k)} = c_1 \nabla f(x^{(k)})^T p^{(k)} > c_2 \nabla f(x^{(k)})^T p^{(k)}$$

since $c_1 < c_2$ and $\nabla f(x^{(k)})^T p^{(k)} < 0$. From this inequality we deduce that α'' satisfies the Wolfe conditions strictly. By smoothness on f, there is an interval around α'' for which the Wolfe conditions hold. Moreover, since $c_1 \nabla f(x^{(k)})^T p^{(k)} < 0$ we also have

$$|\nabla f(x^{(k)} + \alpha'' p^{(k)})^T p^{(k)}| = |c_1 \nabla f(x^{(k)})^T p^{(k)}| < |c_2 \nabla f(x^{(k)})^T p^{(k)}|.$$

Therefore also the strong Wolfe conditions hold in the same interval.

c) We drop the superscript (k) to simplify the notation. Let $\Phi_1(\alpha) = a\alpha^2 + b\alpha + c$. We get a, b and c from the interpolation conditions

$$\begin{aligned} \Phi_1(0) = \Phi(0) \quad &\Rightarrow \quad c = \Phi(0), \\ \Phi_1'(0) = \Phi'(0) \quad &\Rightarrow \quad b = \Phi'(0) \quad \text{and} \\ \Phi_1(\alpha_0) = \Phi(\alpha_0) \quad &\Rightarrow \quad a = \left(\Phi(\alpha_0) - \Phi(0) - \Phi'(0)\alpha_0\right)/\alpha_0^2 \end{aligned}$$

This gives $\Phi_1^{(k)}$ in Algorithm 2.3.3 Line 5. The fact the α_0 does not satisfy Armijo's condition implies

$$\begin{aligned} \Phi(\alpha_0) &> \Phi(0) + c_1\alpha_0\Phi'(0) \\ 0 &< \Phi(\alpha_0) - \Phi(0) - c_1\alpha_0\Phi'(0) \\ 0 &< \Phi(\alpha_0) - \Phi(0) - \alpha_0\Phi'(0) \end{aligned}$$

where we have used $c_1 < 1$ and $\Phi'(0) < 0$. The curvature a is in fact positive. Hence, Φ_1 is convex with minimizer at

$$\alpha_1 = -\frac{\Phi'(0)\alpha_0^2}{2\left[\Phi(\alpha_0) - \Phi(0) - \Phi'(0)\alpha_0\right]}$$

Now, note that from the violation of Armijo's condition we get

$$\begin{aligned} & 0 < \Phi(\alpha_0) - \Phi(0) - c_1\alpha_0\Phi'(0) \\ \Leftrightarrow\ & 0 < \Phi(\alpha_0) - \Phi(0) - c_1\alpha_0\Phi'(0) + \alpha_0\Phi'(0) - \alpha_0\Phi'(0) \\ \Leftrightarrow\ & (c_1 - 1)\Phi'(0)\alpha_0 < \Phi(\alpha_0) - \Phi(0) - \Phi'(0)\alpha_0. \end{aligned}$$

Using these relations, we get

$$\alpha_1 < \frac{\alpha_0}{2(1 - c_1)}.$$

3. *(Newton with Armijo stepsize control,* MATLAB*)*
Let the Himmelblau function be given. We shall visualize the Armijo step size control Algorithm 2.3.3.

a) Implement the MATLAB-function

```
alpha = armijo_step_size(f, gradf, x, p, alpha0, c1,...
                         delta_min, delta_max, maxIt, plot_flag)
```

which realizes Algorithm 2.3.3. The input parameters should be clear, except for `maxIt` and the `plot_flag`. The `maxIt` is used to break the for-loop in Line 15. Ignore the `plot_flag` input parameter for the moment.

b) In the material you will find the MATLAB-function `newton_armijo`. Write a script which applies Newton's method with your step size control algorithm to the Himmelblau-function. Use as an initial value `x0 = [2.6, -3.9]'` and `x0 = [2, -3.5]'`. Plot the path of the iterates in the same plot as the contour lines of the Himmelblau function.

c) Adjust your MATLAB-function `armijo_step_size.m`, such that if the `plot_flag` is true, then the functions $\Phi(\alpha)$, $\Phi_1(\alpha)$, $\Phi_2(\alpha)$[1] and $\Phi_{\text{lin}}(\alpha) := \Phi(0) + c_1\Phi'(0)\alpha$ as well as the interpolation points and minimizer are plotted. Choose the same initial points as in b) and use `maxIt = 1` for Newton's method as well as for the step size control. What do you observe?

[1] This is the cubic interpolant. Plot it only within the loop of Line 15.

(4 + 4 + 4 = 12 Points)

Solution:

a) The MATLAB function could look like

```
function alpha = armijo_step_size(f, gradf, x, p, alpha0, c1, ...
                                  delta_min, delta_max, maxIt, plot_flag)
% Armijo step size determination after algorithm 2.3.3 NumOpt2024

% create Phi and its derivative as functions
Phi = @(alpha) f(x+alpha.*p);
Phi_prime = @(alpha) gradf(x+alpha.*p)'*p;

% Initial values
Phi_0       = Phi(0);
Phi_alpha0  = Phi(alpha0);
Phi_prime_0 = Phi_prime(0);

if Phi_alpha0 <= Phi_0 + c1*alpha0*Phi_prime_0
    % Armijo fulfilled for alpha0
    alpha = alpha0;
    return
else
    % quadratic interpolation
    % coefficients
    a = ((Phi_alpha0 - Phi_0 - alpha0*Phi_prime_0)/alpha0^2);
    b = Phi_prime_0;
    c = Phi_0;
    % quadratic polynomial
    Phi_q = @(alpha) a*alpha.^2 + b*alpha + c;

    % plot quadratic function
    if plot_flag
        alpha_plot = linspace(-alpha0,alpha0);
        figure(2)
        % line search function \Phi(\alpha)
        plot(alpha_plot, Phi(alpha_plot), 'b-', LineWidth=2)
        hold on
        % quadratic interpolant \Phi_1(\alpha)
        plot(alpha_plot, Phi_q(alpha_plot), 'r-', LineWidth=2)
        % Armijo condition
        plot(alpha_plot, Phi_0 + c1*Phi_prime_0*alpha_plot, 'k-', LineWidth=1)
        % interpolation points
        plot(0, Phi_0, 'ko', LineWidth=2, MarkerSize=10)
        plot(alpha0, Phi(alpha0), 'ko', LineWidth=2, MarkerSize=10)
        plot(0, Phi(0), 'ko', LineWidth=2, MarkerSize=10)
        % minimum of quadratic interpolant
        plot(-b/(2*a), Phi_q(-b/(2*a)), 'ro', LineWidth=2, MarkerSize=10)
        % \Phi at the minimum of quadratic interpolant
        plot(-b/(2*a), Phi(-b/(2*a)), 'bo', LineWidth=2, MarkerSize=10)
        grid on
        legend("$\Phi(\alpha) = f(x+\alpha p)$", ...
               "Quadratic Approximation",...
                   'Armijo condition', interpreter="latex")
        set(gca,"FontSize",18)
        hold off
    end

    % minimum of quadratic polynomial
    alpha1 = -b/(2*a);
```

```
end

% check Armijo again
if Phi(alpha1) <= Phi_0 + c1*alpha1*Phi_prime_0
    alpha = alpha1;
    return
end

% check if stepsize difference is too small
if alpha0 - alpha1 < delta_min || alpha0 - alpha1 > delta_max
    alpha1 = alpha0/2;
end

% check Armijo again
if Phi(alpha1) <= Phi_0 + c1*alpha1*Phi_prime_0
    alpha = alpha1;
    return
else
    % start cubic interpolation in for loop
    % define alpha values
    alpha_jm1 = alpha0;
    alpha_j   = alpha1;
    % count iterations
    iter_count = 1;
    while 1
        % factor
        gamma_jp1 = 1/((alpha_jm1*alpha_j)^2*(alpha_j-alpha_jm1));
        % determine cubic coefficients
        A = [[alpha_jm1^2, -alpha_j^2];[-alpha_jm1^3, alpha_j^3]];
        b = [Phi(alpha_j)   - Phi_0 - Phi_prime_0*alpha_j; ...
             Phi(alpha_jm1) - Phi_0 - Phi_prime_0*alpha_jm1];
        x = gamma_jp1*A*b;
        % get the coefficients
        a_jp1 = x(1);
        b_jp1 = x(2);

        % cubic polynomial
        Phi_c = @(alpha)   a_jp1*alpha.^3 + b_jp1*alpha.^2 ...
                         + Phi_prime_0*alpha + Phi_0;

        % determine root
        alpha_root = ((-b_jp1 + sqrt(b_jp1^2 - 3*a_jp1*Phi_prime_0))/(3*a_jp1));

        % plot quadratic function
        if plot_flag
            alpha_plot = linspace(-alpha0,alpha0);
            figure(3)
            subplot(maxIt,1,iter_count);
            % line search function \Phi(\alpha)
            plot(alpha_plot, Phi(alpha_plot), 'b-', LineWidth=2)
            hold on
            % quadratic interpolant \Phi_1(\alpha)
            plot(alpha_plot, Phi_q(alpha_plot), 'r-', LineWidth=2)
            % cubic interpolant \Phi_2(\alpha)
            plot(alpha_plot, Phi_c(alpha_plot), 'c-', LineWidth=2)
             % Armijo condition
            plot(alpha_plot, Phi_0 + c1*Phi_prime_0*alpha_plot, 'k-', LineWidth=1)
            % interpolation points
            plot(alpha_jm1, Phi(alpha_jm1), 'ko', LineWidth=2, MarkerSize=10)
            plot(alpha_j, Phi(alpha_j), 'ko', LineWidth=2, MarkerSize=10)
            plot(0, Phi(0), 'ko', LineWidth=2, MarkerSize=10)
```

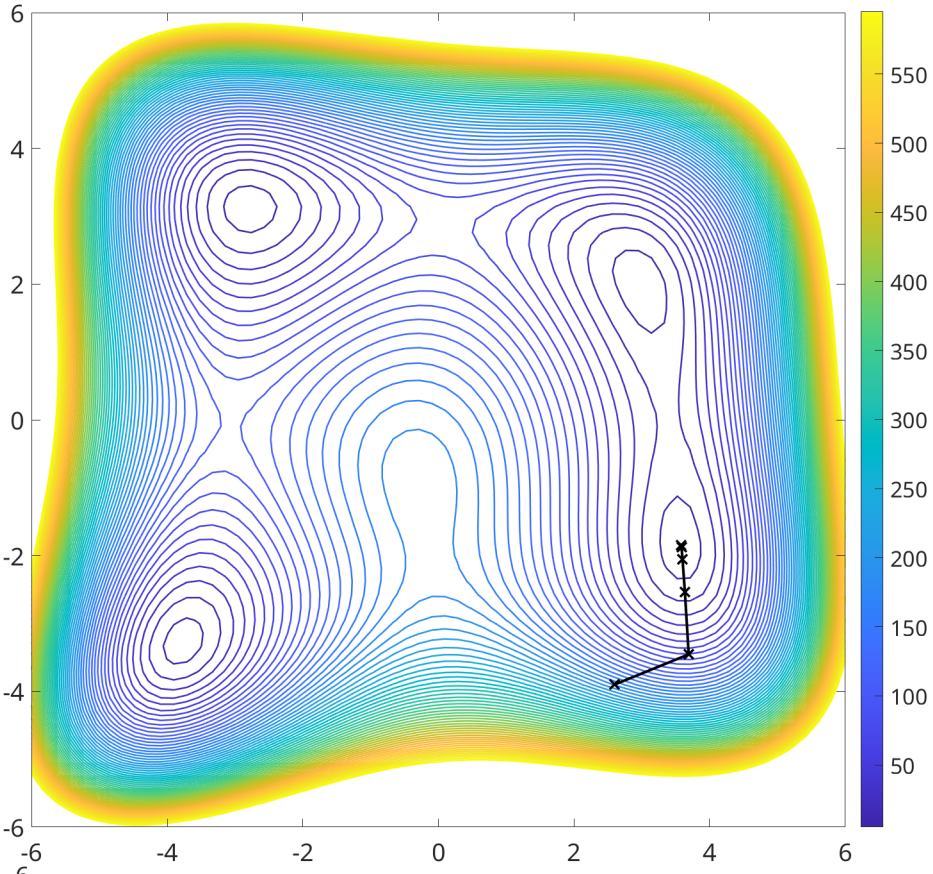
```
            % minimum of cubic interpolant
            plot(alpha_root, Phi_c(alpha_root), 'co', LineWidth=2, MarkerSize=10)
            % \Phi at the minimum of cubic interpolant
            plot(alpha_root, Phi(alpha_root), 'bo', LineWidth=2, MarkerSize=10)
            grid on
            legend("$\Phi(\alpha) = f(x+\alpha p)$", ...
                   "Quadratic Approximation", ...
                   "Cubic Approximation",...
                   'Armijo condition', interpreter="latex")
            set(gca,"FontSize",18)
            hold off
        end

        % assign the root
        alpha_jp1 = alpha_root;

        % compute the Phi value
        Phi_alpha_jp1 = Phi(alpha_jp1);

        % check if difference is too small
        if alpha_j - alpha_jp1 < delta_min || alpha_j - alpha_jp1 > delta_max
            alpha_jp1 = alpha_j/2;
        end

        % check Armijo condition
        if Phi_alpha_jp1 <= Phi_0 + c1*alpha_jp1*Phi_prime_0
            alpha = alpha_jp1;
            return
        end

        % check maxIt
        if iter_count == maxIt
            warning("The maximal number of iterations in step size " + ...
                "search has been reached. The algorithm is still " + ...
                "running and may end up with a very small step size.")
            alpha = alpha_jp1;
            return
        end

        % redefine old and new alpha values
        alpha_jm1 = alpha_j;
        alpha_j   = alpha_jp1;

        % update counter
        iter_count = iter_count + 1;
    end
end
end
```

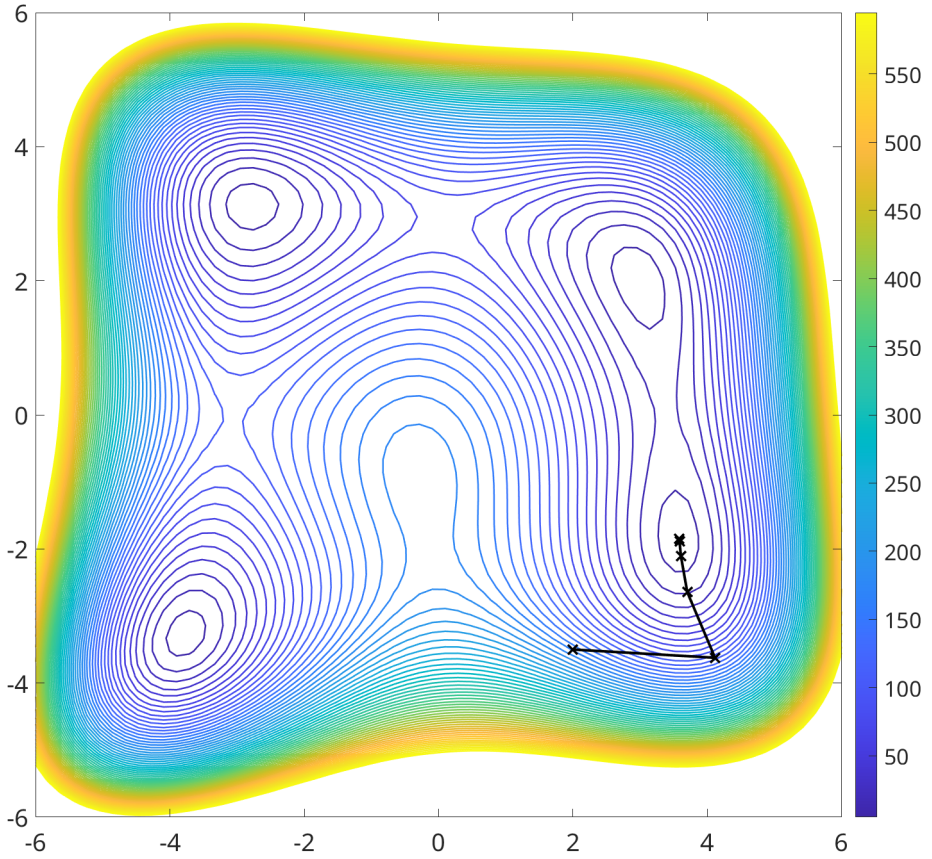
b) The MATLAB script could look like

```
% Newton method using Armijo stepsize rule.
% It terminates when the norm of the gradient is below 10^(-6).
clear, close all
clc

% Himmelblau function and derivative
Himmelblau = @(x,y) (x.^2+y-11).^2+(x+y.^2-7).^2;

f  = @(x) (x(1,:).^2+x(2,:)-11).^2+(x(1,:)+x(2,:).^2-7).^2;
gradf = @(x) [x(1).*(4*x(1).^2+4*x(2)-42)+2*(x(2).^2-7);
```

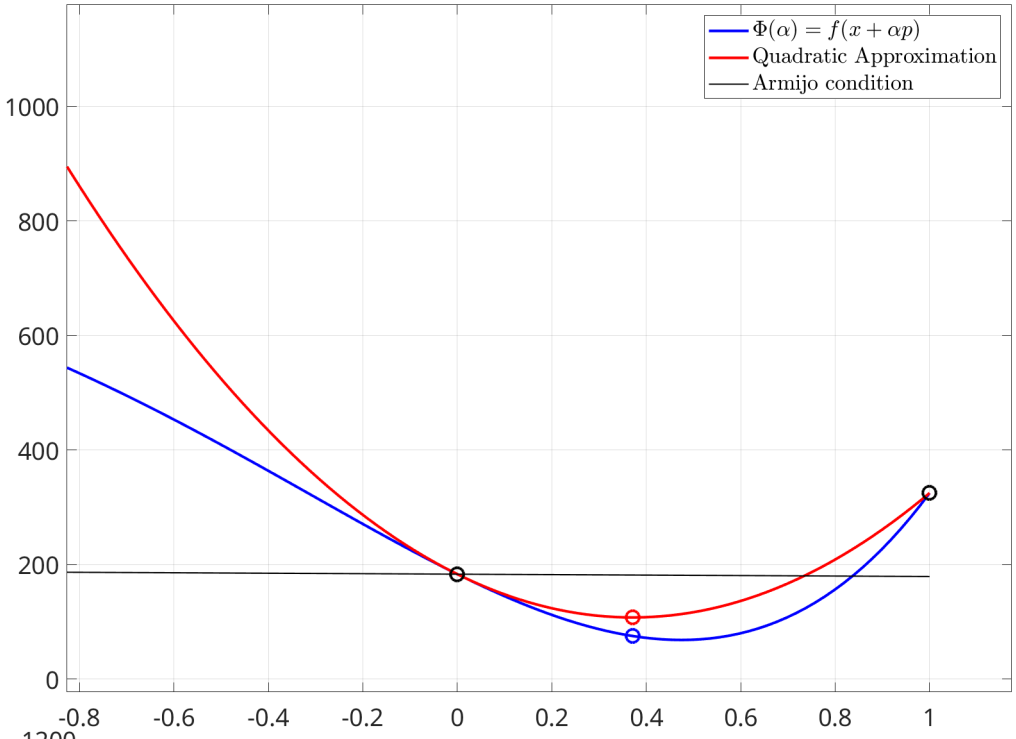
```
           x(2).*(4*x(2).^2+4*x(1)-26)+2*(x(1).^2-11)];
Hessf = @(x)  [12*x(1).^2+4*x(2)-42, 4*x(1)+4*x(2);...
                   4*x(1)+4*x(2), 12*x(2).^2+4*x(1)-26];


% Netwon settings
maxIter = 1;
tol     = 1e-6;
% initial value
% try:
x0 = [2.6, -3.9]'; % quadratic function not sufficient for c1 = 8e-1
% try also:
% x0 = [2, -3.5]';

% Armijo stepsize rule parameters
alpha0           = 1;
c1               = 1e-2;
delta_min        = 1e-2;
delta_max        = 1e+1;
maxIter_stepsize = 1;
% flag if interpolation should be plotted
plot_flag = 1;

[allx, iter] = newton_armijo(f, gradf, Hessf, ...
                             x0, ...
                             maxIter, tol, ...
                             alpha0, c1, delta_min, delta_max, ...
                             maxIter_stepsize, plot_flag);

x = linspace(-6,6,64);
y = linspace(-6,6,64);
[xx,yy] = meshgrid(x,y);
ff = Himmelblau(xx,yy);
levels = 5:10:600;
figure(1)
contour(x,y,ff,levels,LineWidth=1.2),
colorbar
axis([-6 6 -6 6]),
axis square,
hold on
plot(allx(1,:),allx(2,:),'kx-', MarkerSize=10, LineWidth=2);
set(gca, "FontSize", 18)
hold off
```

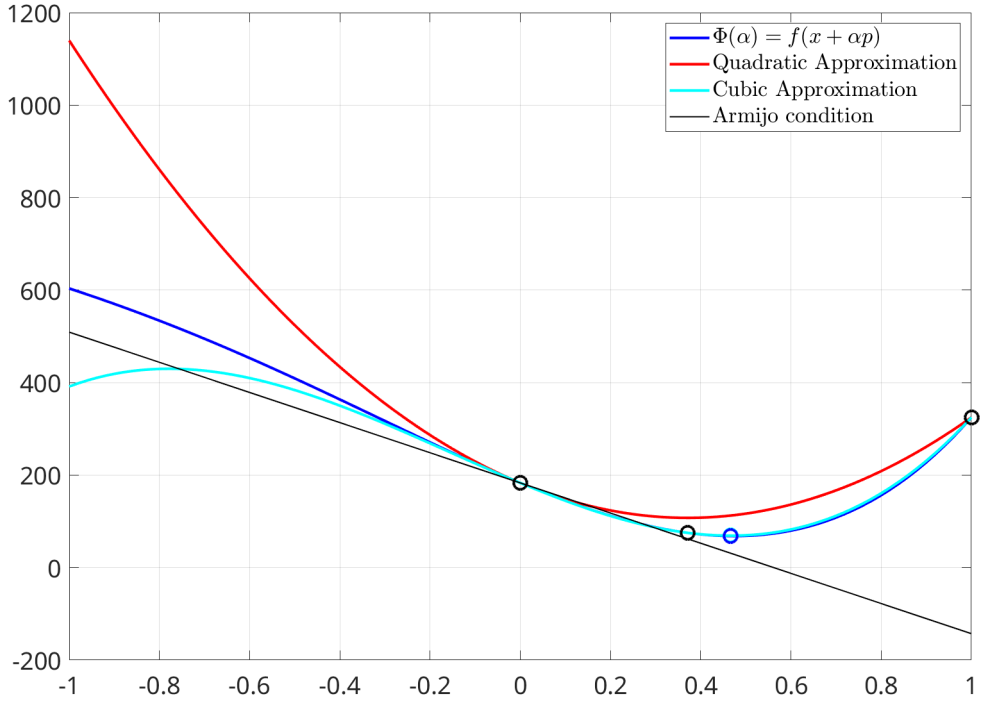
and the iterations for the two initial values could look like



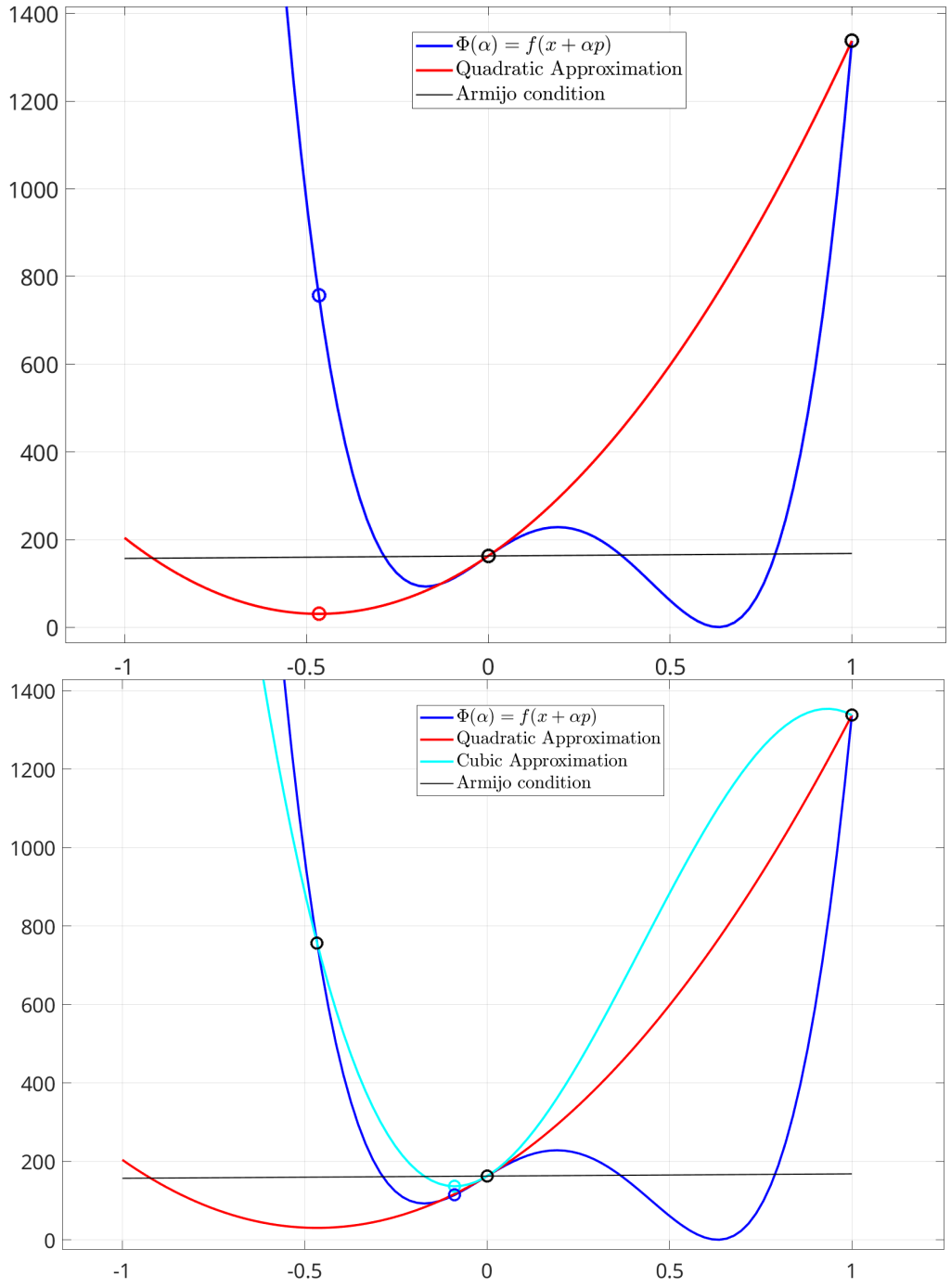


c) The plots could for the first initial value look like





The cubic interpolation is just for reference, the quadratic minimizer has been accepted in this case. For the second initial value we get



The first thing we notice is, that we are not looking in a decent direction for $\alpha > 0$. This is because the Hessian is not positiv definit. Besides that, the quadratic interpolant gives us a value for α for which the Armijo condition is not fulfilled. The cubic interpolant does.