# Math 4650/MSSC 5650 - Homework 1

Henri Medeiros Dos Reis

January 27, 2023

**Problem 1** (MATLAB, 5 pts). Starting from the provided script `sd.m`, use the steepest descent algorithm to find a minimizer of $f(x) = \frac{1}{2}x^2 - \sin(x)$ with the initial guess $x_0 = 0.5$ and a step-size of $\tau = 0.1$. Add an exit condition to the for-loop such that the algorithm terminates when the absolute difference between iterates $|x_{k+1} - x_k|$ is less than $10^{-6}$. How many iterations $k$ does it take to reach the exit condition? Did the algorithm converge to a local minimizer? How do you know? In your write-up, include a plot of $f(x_k)$ versus $k$, plus a screenshot or print-out of your code.

**Solution 1.** .

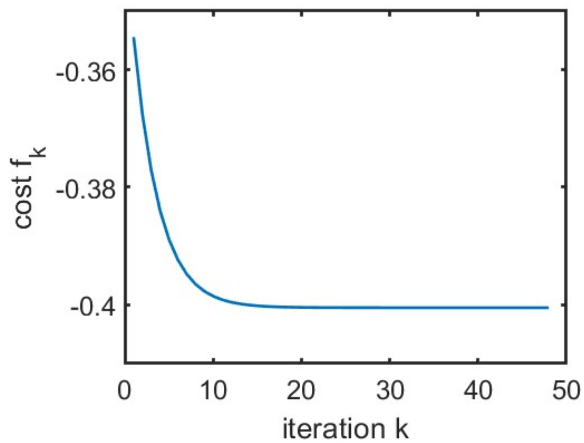
```
f = @(x) 0.5*x.^2 - sin(x);
df = @(x) x - cos(x);

x = 0.5;
tau = 0.1;

iter = 500;
f_k = [f(x)];
for k=1:iter
    xk = x;
    x = x - tau*df(x);
    f_k(end+1) = f(x);
    if abs(x - xk) < 10e-6
        break;
    end
end
disp(x);
disp( iter: + k)

figure(1);
plot(f_k,'linewidth',2);
ylabel('cost f_k');
xlabel('iteration k');
```

```
set(gca,'fontsize',18);
set(gca,'linewidth',2);
```




It took 47 iterations to reach the exit condition. And yes, it did converge to a local minimizer, as we can see on the graph, the line is a horizontal line, and if we did not have the exit condition and let it run for many more iterations, the result would be almost the same, maybe some really small difference, when I let it run for 500 iterations the result changed only from 0.739038 to 0.739085.

**Problem 2** (MATLAB, 5 pts)**.** Implement Newton's method to find an approximate minimizer of $f(x) = \frac{1}{2}x^2 - \sin(x)$ with the initial guess $x_0 = 0.5$. How does Newton's method compare with the steepest descent algorithm in terms of how many iterations it takes to reach the same exit condition in Problem 1? Did the algorithm converge to a local minimizer? How do you know? In your write-up, include a plot of $f(x_k)$ versus $k$, plus a screenshot or print-out of your code.

**Solution 2.** .

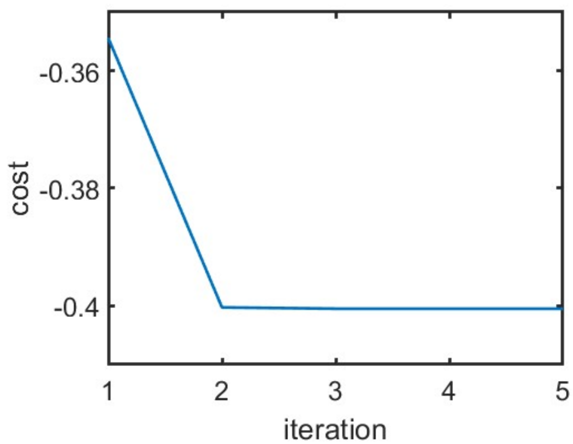
```
f = @(x) 0.5*x.^2 - sin(x);
df = @(x) x - cos(x);
d2f = @(x) 1 + sin(x);

x = 0.5;

iter = 500;
f_k = [f(x)];
for k=1:iter
    xk = x;
    tau = 1/d2f(x);
    x = x - tau*df(x);
```

```
    f_k(end+1) = f(x);
    if abs(x - xk) < 10e-6
        break;
    end
end
disp(x);
disp( iter: + k)

figure(1);
plot(f_k,'linewidth',2);
ylabel('cost');
xlabel('iteration');
set(gca,'fontsize',18);
set(gca,'linewidth',2);
```




It took 4 iterations to reach the exit condition, which is less than 10 of what the steepest descent method needed reach the exit condition. And yes, it did converge to a local minimizer, as we can see on the graph, the line is a horizontal line, and if we did not have the exit condition and let it run for many more iterations, the result would be the same, when I let it run for 500 iterations the result did not change at all.

**Problem 3** (5 pts)**.** Suppose $f$ is a quadratic with positive curvature, i.e., $f(x) = ax^2+bx+c$ for some constants $a, b, c$ with $a > 0$. Prove that Newton's method reaches the global minimizer of $f$ in *one iteration*, regardless of the initial point $x_0$.

**Solution 3.** Since $f(x) = ax^2 + bx + c$ then $f'(x) = 2ax + b$ and $f''(x) = 2a$.

Then we can apply Newton's method:

$$x = x_0 - \frac{f'(x_0)}{f''(x_0)} = x_0 - \frac{2ax_0 + b}{2a} = \frac{2ax_0 - (2ax_0 + b)}{2a} = \frac{-b}{2a}$$

Then $x = \frac{-b}{2a}$is the global minimizer, and $x_0$ ends up not playing any role, since it cancels.

**Problem 4** (5 pts)**.** Let $f(x) = ax^2$ with $a > 0$. Given any initial point $x_0 \neq 0$, prove that for any step-size $\tau$ in the range $0 < \tau < 1/a$, we have $f(x_1) < f(x_0)$ where $x_1 = x_0 - \tau f'(x_0)$ is the first iterate of steepest descent.

**Solution 4.**

$$f(x_1) < f(x_0) \to f(x_0 - \tau f'(x_0)) < f(x_0) \Rightarrow$$
$$a(x_0 - \tau f'(x_0))^2 < ax_0^2 \text{ and } f'(x_0) = 2ax_0 \Rightarrow$$
$$a(x_0^2 - 2\tau ax_0)^2 < ax_0^2 \to x_0^2 - 4\tau x_0^2 + 4\tau^2 a^2 x_0^2 < x_0^2$$
$$-4\tau ax_0^2 + 4\tau^2 a^2 x_0^2 < 0 \Rightarrow$$
$$-4\tau ax_0^2(1 - \tau a) < 0$$

Since $a \neq 0$ and $0 < \tau < \frac{1}{a}$, then $1 - \tau a < 0 \to 1 < \tau a$. Which is true, since the largest possible value for $\tau$ is still smaller than $\frac{1}{a}$

**Problem 5** (5 pts)**.** Steepest descent and Newton's method are designed to minimize a function $f$. How should these algorithms be modified if the goal is to *maximize* a function $f$ instead?

**Solution 5.** We would need to change the signs in the steps, such that for steepest descent, the step would be $x_{k+1} = x_k + \tau f'(x_k)$. And for Newton's method, we do not need to change anything, since Newton's method just finds a local extrema, it is not guaranteed to find a maximum or minimum, so we can use the same method, and be carefull with our starting point. Also, the functions that we use as input need to be maximizable. follow one example in MATLAB.

```
%% declare functions
f = @(x) -5*x.^2+5;
df = @(x) -10*x;
d2f = @(x) -10;
%% steepest descent
x = 10;
tau = 0.01;
iter = 500;
cost = [f(x)];
for k=1:iter
    xk = x;
    x = x + tau*df(x);
    cost(end+1) = f(x);
    if abs(x - xk) < 10e-6
        break;
    end
end
```

```
disp(x);
disp( iter: + k)

%% newton's method
iter = 500;
f_k = [f(x)];
for k=1:iter
    xk = x;
    tau = 1/d2f(x);
    x = x - tau*df(x);
    f_k(end+1) = f(x);
    if abs(x - xk) < 10e-6
        break;
    end
end
disp(x);
disp( iter: + k)
```

---

**Problem 6** (MSSC, 5pts)**.** Construct (or draw) a function for which Newton's method fails to converge, but steepest descent does converge when starting from the same initial point $x_0$. Justify your answer.

**Solution 6.** Let's consider the following function:

$$f(x) = x^4 - x^3 + 2 \text{ then}$$

$$f'(x) = 4x^3 - 3x^2 \text{ and } f''(x) = 12x^2 - 6x$$

If we pick $x_0$ to be $\frac{1}{2}$, then let's apply Newton's method

$$x = \frac{1}{2} - \frac{4(\frac{1}{2})^3 - 3(\frac{1}{2})^2}{12(\frac{1}{2})^2 - 6(\frac{1}{2})} = \frac{1}{2} - \frac{4(\frac{1}{2})^3 - 3(\frac{1}{2})^2}{0}$$

Then it would fail to converge. While if we use the steepest descent method it will converge to 0.74997, by using the following MATLAB code.

```
f = @(x) x.^4-x.^3+2;
df = @(x) 4*x.^3-3*x.^2;

x = 0.5;
tau = 0.1;
```

```
iter = 500;
f_k = [f(x)];
for k=1:iter
    xk = x;
    x = x - tau*df(x);
    f_k(end+1) = f(x);
    if abs(x - xk) < 10e-6
        break;
    end
end
disp(x);
disp( iter: + k)

figure(1);
plot(f_k,'linewidth',2);
ylabel('cost');
xlabel('iteration');
set(gca,'fontsize',18);
set(gca,'linewidth',2);
```

The reason why this happens is the fact that the second derivative of $f(x)$ at this exact point is 0, and we cannot divide by 0, so the algorithm fails.

**Problem 7** (MSSC, 5pts)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be continuously differentiable (meaning $f'$ exists and is continuous everywhere), and let $\{x_k\}_{k=0}^{\infty}$ be the sequence of iterates obtained by applying steepest descent to $f$ with some fixed step-size $\tau > 0$. Suppose the iterates $\{x_k\}_{k=0}^{\infty}$ converge to a point $x^*$, i.e., $\lim_{k\to\infty} x_k = x^*$. Prove that $x^*$ is a critical point of $f$. [Hint: Recall that for any continuous function $g$, $\lim_{k\to\infty} x_k = x$ implies $\lim_{k\to\infty} f(x_k) = f(x^*)$.]

**Solution 7.** To show that $x^*$ is a critical point, we need to show that $f'(x) = 0$. Since $\lim_{k\to\infty} x_k = x^*$ implies $\lim_{k\to\infty} g(x_k) = g(x^*)$.

We can use the mean value theorem to check if there is a value $x_{k+i}$ between $x_k$ and $x^*$ such that $f'(x_{k+i}) = 0$ then

$$f'(x_{k+i}) = \frac{\lim_{k\to\infty} f(x_k) - f(x^*)}{\lim_{k\to\infty} x_k - x^*} = \frac{f(x^*) - f(x^*)}{x^* - x^*}$$

And both the numerator and the denominator will approach 0 at the same rate. Therefore $f'(x_{k+i}) = 0$ and if we apply the steepest descent:

$$x_{k+i} = x_{k+i} - \tau f'(x_{k+i}) = x_{k+i} - 0$$

Which shows that it converged. and that this specific point is a critical point.