Optimization for Learning - FRTN50

Assignment 2

November 1, 2020

Task 1 Which of the following problem objectives are L-smooth? Which are σ-strongly convex?
Problem 1

$$\min_{x\in\mathbf{R}^n} f_1(x) = \min_{x\in\mathbf{R}^n} \frac{1}{4}||x||_2^2 + \frac{1}{2}||x||_2 \quad \textbf{(1)}$$

f_1 is not smooth:

$g_1 = \frac{L}{2}||.||_2^2 - f_1$ is not convex for any L:
Take an example of $x = \frac{1}{L}$, $y = \frac{-1}{L}$ and $\theta = \frac{1}{2}$
We have : $g_1(\theta x + (1-\theta)y) = 0 > \frac{-1}{4L^2} = \theta g_1(x) + (1-\theta)g_1(y)$

f_1 is strongly convex:

$g_1 = f_1 - \frac{L}{2}||.||_2^2 = \frac{\frac{1}{2}-L}{2}||.||_2^2 + \frac{1}{2}||.||_2$ is convex for $L \leq \frac{1}{2}$:
We know that $||.||_2$ is convex and positive and $\frac{\frac{1}{2}-L}{2}(.)^2$ is increasing for positive inputs. Then g_1 is a sum of positive convex functions. Hence g_1 is convex.

Problem 2

$$\min_{x\in\mathbf{R}^n} f_2(x) = \min_{x\in\mathbf{R}^n} \frac{1}{3}||x||_2^3 \quad \textbf{(2)}$$

f_2 is not smooth:

$g_2 = \frac{L}{2}||.||_2^2 - f_2$ is not convex for any L:
Take an example of $x = 2L$, $y = -2L$ and $\theta = \frac{1}{2}$
We have : $g_2(\theta x + (1-\theta)y) = 0 > \frac{-2L^3}{3} = \theta g_2(x) + (1-\theta)g_2(y)$

f_2 is not strongly convex:

$g_2 = f_2 - \frac{L}{2}||.||_2^2$ is not convex for any L:

Take an example of $x = L$, $y = -L$ and $\theta = \frac{1}{2}$

We have : $g_2(\theta x + (1-\theta)y) = 0 > \frac{-L^3}{6} = \theta g_2(x) + (1-\theta)g_2(y)$

Problem 3

$$\min_{x \in \mathbf{R}^n} f_3(x) = \min_{x \in \mathbf{R}^n} \frac{1}{2}||x||_2^2 \quad \textbf{(3)}$$

f_3 is smooth:

$g_3 = \frac{L}{2}||.||_2^2 - f_3 = \frac{L-1}{2}||.||_2^2$ is convex for $L \geq 1$:

f_2 is strongly convex:

$g_3 = f_3 - \frac{L}{2}||.||_2^2 = \frac{1-L}{2}||.||_2^2$ is convex for $L \leq 1$:

Problem 4

$$\min_{x \in \mathbf{R}^n} f_4(x) = \min_{x \in \mathbf{R}^n} \sum_{i=1}^{N} \frac{1}{2}\log(1 + e^{-l_i a_i^T x}) = \min_{x \in \mathbf{R}^n} \sum_{i=1}^{N} h(-l_i a_i^T x) \quad \textbf{(4)}$$

f_4 is smooth and convex:

We know that the function $-l_i a_i^T x$ is affine and $h(y) = \frac{1}{2}\log(1 + e^y)$ is convex, then f_4 is convex since it is a positive sum of convex functions.

We have h is $\frac{1}{8}$-smooth (and strictly convex):

$$\frac{d^2}{dx^2}h(x) = \frac{e^x}{2(1+e^x)^2} \in \quad \left]0, \frac{1}{8}\right]$$

We have:

$$\nabla^2 f_4(x) = \sum_{i=1}^{N} \nabla^2(h \circ -l_i a_i^T)(x) = \sum_{i=1}^{N} l_i^2 a_i \nabla^2 h(-l_i a_i^T x) a_i^T \leq \frac{1}{8}\sum_{i=1}^{N} a_i a_i^T = \frac{1}{8}A^T A$$

Therefore the problem is $\frac{1}{8}\lambda_{\max}(A^T A)$-smooth, with $A^T = (a_1, ..., a_N)$

f_4 is not strongly convex:

We see that $\frac{d^2}{dx^2}h(x) \to 0$ as $x \to +\infty$ and $x \to -\infty$. Therefore there is not positive lower bound i.e. h is not strongly convex. Hence the problem is not strongly convex.

Problem 5

$$\min_{x\in\mathbf{R}^n} f_5(x) = \min_{x\in\mathbf{R}^n} \sum_{i=1}^{N} \frac{1}{2}(a_i^T x - l_i)^2 \quad \textbf{(5)}$$

The problem is seen as:

$$f_5(x) = \frac{1}{2}||Ax - l||_2^2$$

With:

$$A = \begin{pmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_N^T \end{pmatrix}, \quad l = \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_N \end{pmatrix}$$

We know that f_5 is twice differentiable, and its second derivative is always positive semi-definite:

$$\nabla^2 f_5(x) = A^T A$$

Therefore:

- f_5 is $\lambda_{\max}(A^T A)$-smooth.
- f_5 is $\lambda_{\min}(A^T A)$-strongly convex. **if and only if $A^T A$ is positive definite**

Task 2 Problems (1), (2) and (3) are particularly simple. Prove that all three have the unique solution 0.

Problems (1) and (3):

We know that problems (1) and (3) are strongly convex, then the solution exists and it is unique.

We also see that:

$$f_1(x) \geq 0 = f_1(0), \qquad f_3(x) \geq 0 = f_3(0) \quad \forall x \in \mathbf{R}^n$$

i.e. 0 is a solution of both problems, then it is the unique solution.

Problem (2):

In this case $f_2(x) = (h_2 \circ g_2)(x)$ the problem is strictly convex since:

- $g_2(.) = ||.||_2^2$ is strictly convex (since it is strongly convex)
- $h_2(.) = \frac{1}{3}(.)^{\frac{3}{2}}$ is strictly increasing function for strictly positive inputs.

We also see that:

$$f_2(x) \geq 0 = f_2(0) \quad \forall x \in \mathbf{R}^n$$

Then 0 is the unique minimizer (unique solution)

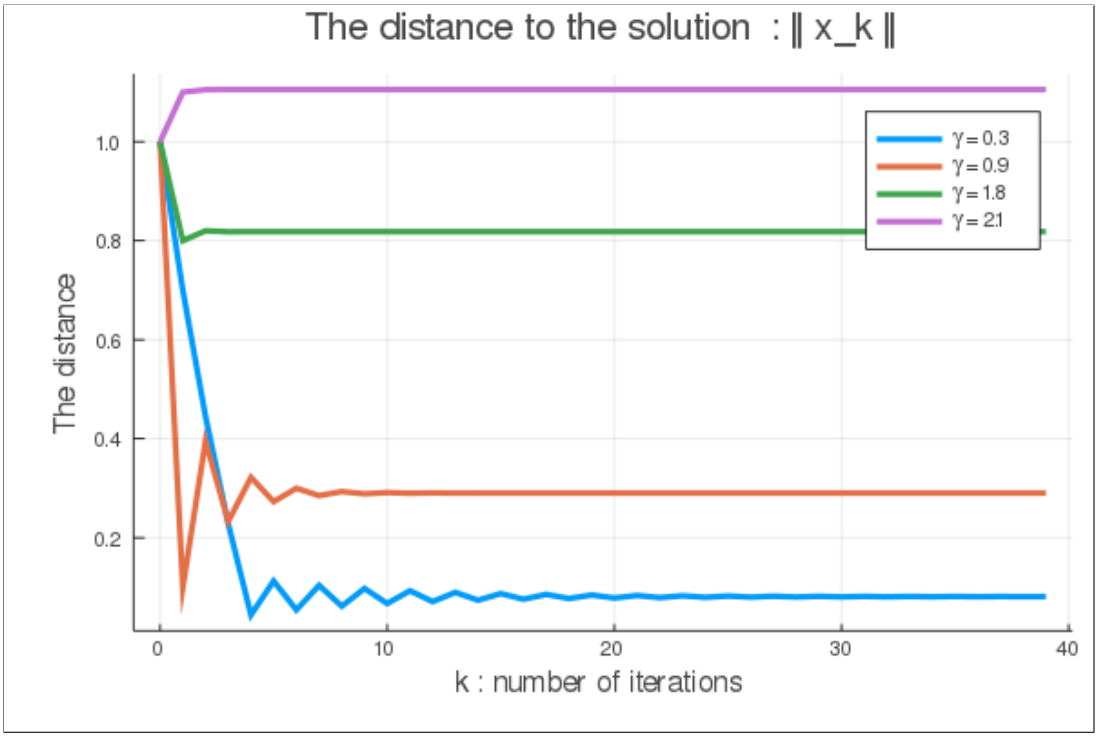
Task 3 Solve (1) with (sub-)gradient descent:

Does the distance to the solution converge?

We write the function **solve1(k, x)** that starts with x and plots the distance to the solution $||x^i - 0|| = ||x^i||$ at each iteration over k iterations, without forgetting that x must be an array

The case of the initial point $x = [1.0]$: In order to see clearly how fast we reach a given distance to the solution we use 40 iterations $k = 40$:

```
julia> solve1(40, [1.0])
```



Result
As you can see the distances to the solution do not converge, they are constant from a certain rank.

Does increasing or decreasing the step-size affect how close to the solution you can come?

It seems like decreasing the step-size makes the iterates close to the solution. Of course this not a rule, this what we notice from the figure above (we will see further that for $\gamma = 1.0$ the iterates converge).

Does increasing or decreasing the step-size affect how fast you can reach a given distance to the solution?

This is true, see the figure above, the lower the step-size is the slower the iterates reach a given distance to the solution.

Find a step-size in the interval (0,3) that makes the iterates converge to the solution?

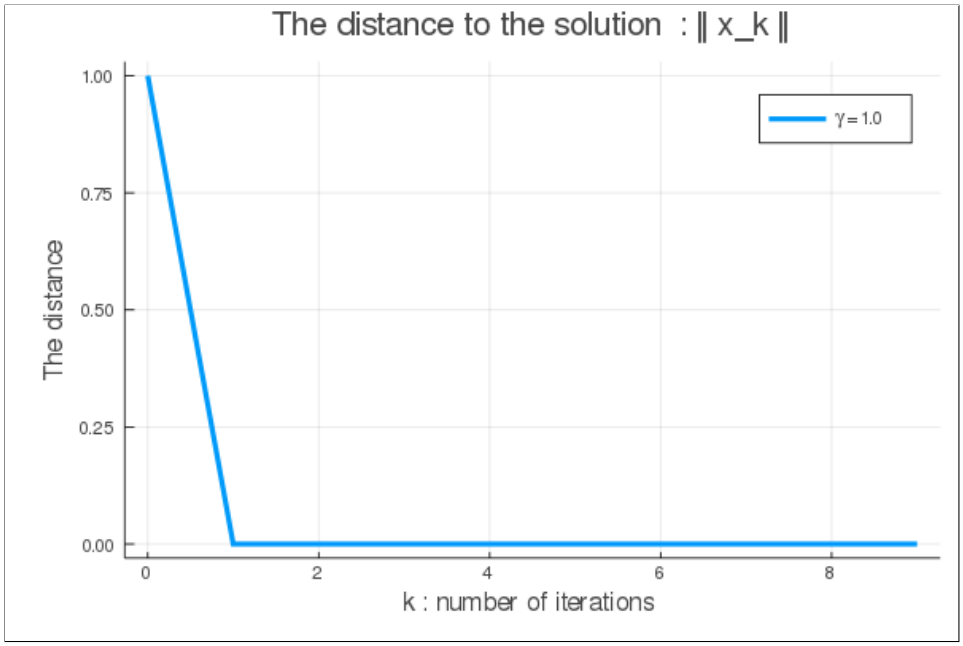
Using the sub-gradient method ($x^0 = 1$ dimension $n = 1$, we have $f_1(x) = \frac{1}{4}x^2 + \frac{1}{2}x$ for positive inputs)
Then:

$$x^1 = x^0 - \gamma \nabla f_1(x^0) = 1 - \gamma$$

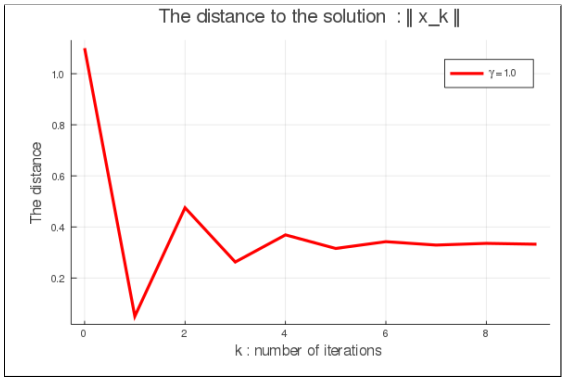
So if we choose $\gamma = 1$ we will get for each iterates $k \geq 1$:

$$x^k = x^{k-1} - \gamma \nabla f_1(x^{k-1}) = 0$$

Then the step-size $\gamma = 1$ makes the iterates converge to the solution (see figure below)



Try changing the initial point to $x = 1.1$, do you still converge?



If we use the same step-size $\gamma = 1$ as before, the iterates will not converge to the solution.

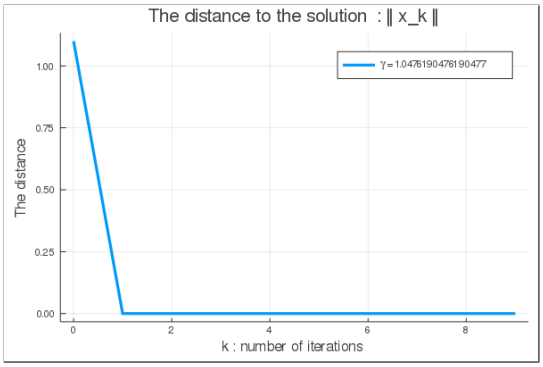
This solution is true only for positve inputs:

$$0 = x - \gamma \nabla f_1(x)$$

i.e.

$$\gamma = \frac{2x}{x+1}$$

Then we plot the distances according to the new step-size:



Task 4 Solve (2) with gradient descent:

Does the distance to the solution converge? Do the iterates appear to converge to the solution?

In this task we plot the norm of the sequence values for different initial points.

For the case of convergence, we initially see fast progress to medium accuracy (around 75 and 100 iterations) followed by a slower progress. Therefore, reaching the medium accuracy is enough for such problem, that is why we choose to stop at 100 iterations.

Case 1 : $x \in \{0.3, 0.9, 1.8\}$
As we can see all these initial points are less than 2, the sequence x^k seems **converges** to the solution $x^* = 0$, but very slowly.

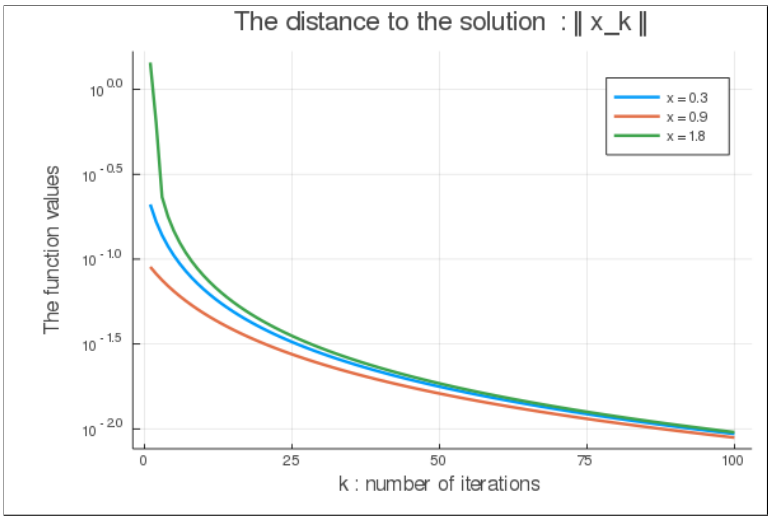
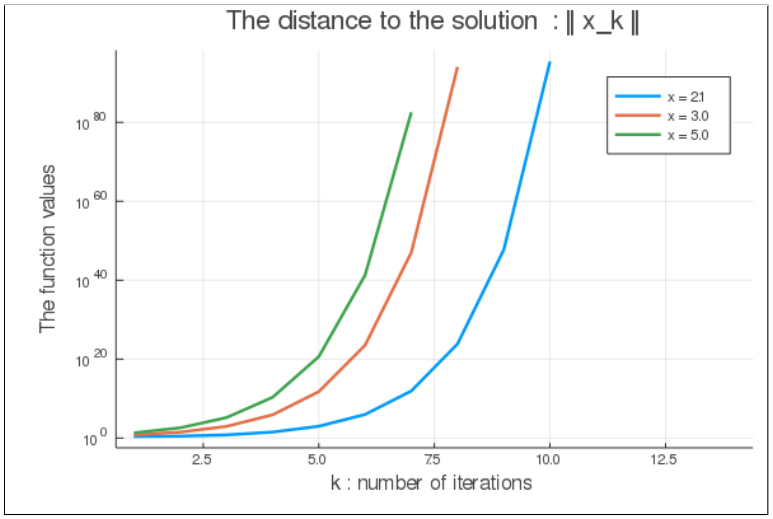


Figure 1: **step size** $\gamma = 1$



Case 2 : $x \in \{2.1, 3, 5\}$
According to this case, the iterates of all initial points that are strictly greater than 2 **diverge** just after few iterations $k = 10$.

Figure 2: **step sizes** $\gamma = 1$

Does initial point affect how close to the solution you can come?

As we can see, using intial points less than 2 makes the algorithm converges to 0, and the use of points strictly greater than 2 makes it diverge to infinity.

Using the definition of the gradient descent ($\gamma = 1$):

$$x^{k+1} = x^k - \nabla f_2(x^k) = x^k - x^k||x^k||_2$$

The function $g(x) = x - x||x||_2$ is differentiable and continuous, with:

- Decreasing on $\left]-\infty, \frac{-1}{2}\right]$ with $g\left(\left]-\infty, \frac{-1}{2}\right]\right) = \left[\frac{-1}{2}, +\infty\right[$ and $g(-2) = 2$.
- Increasing on $\left[\frac{-1}{2}, \frac{1}{2}\right]$ with $g\left(\left[\frac{-1}{2}, \frac{1}{2}\right]\right) = \left[\frac{-1}{2}, \frac{1}{2}\right]$
- Decreasing on $\left[\frac{1}{2}, +\infty\right[$ with $g\left(\left[\frac{1}{2}, +\infty\right[\right) = \left]-\infty, \frac{1}{2}\right]$ and $g(2) = -2$.

Case 1 : $x_0 \in [-2, 2]$ we have $x^1 = g(x_0) \in [-2, 2]$ and so on until we reach a point $x^k \in \left[\frac{-1}{2}, \frac{1}{2}\right]$
Then after a certain rank it converges to 0.

Case 2: $x_0 \notin [-2, 2]$ then x^1 won't belong to the interval and so on until infinity, which explains the divergence of this sequence.

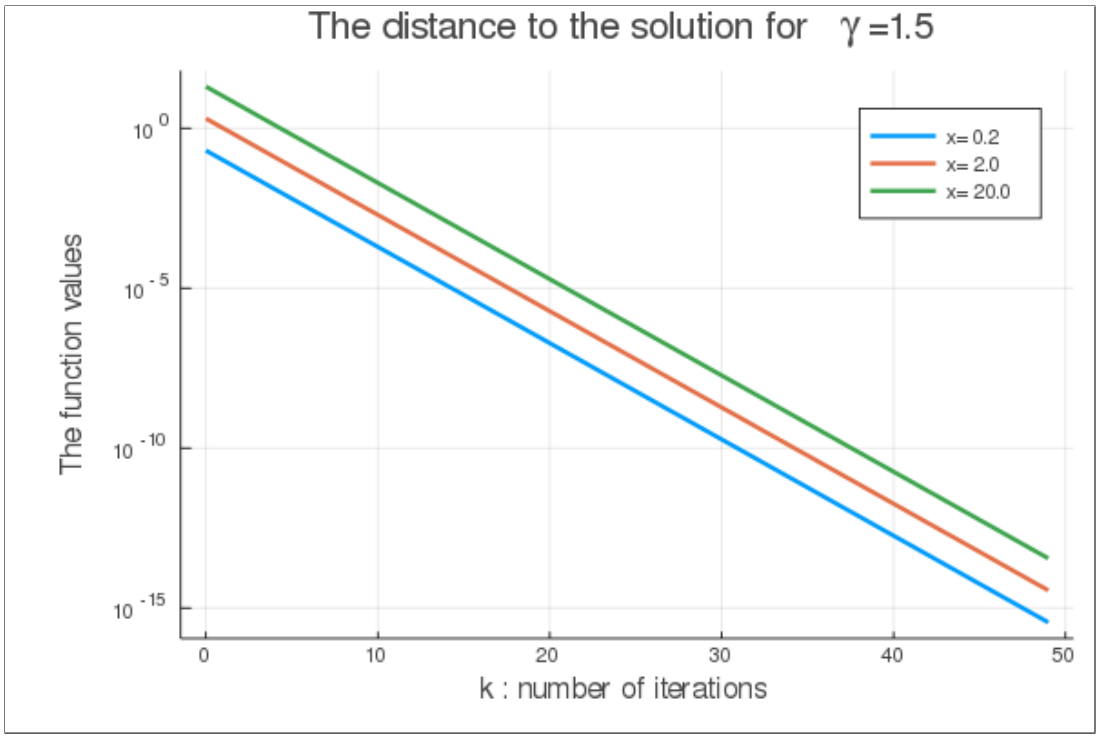
Task 5 Solve (3) with gradient descent:

Does the distance to the solution converge?

We write the function **solve3(k, x)** that uses the step-size γ and plots the distance to the solution $||x^i - 0|| = ||x^i||$ for the initial points $x \in \{0.2, 2, 20\}$ over k iterations.

For example:

```
julia> solve3(10, 1.5)
```



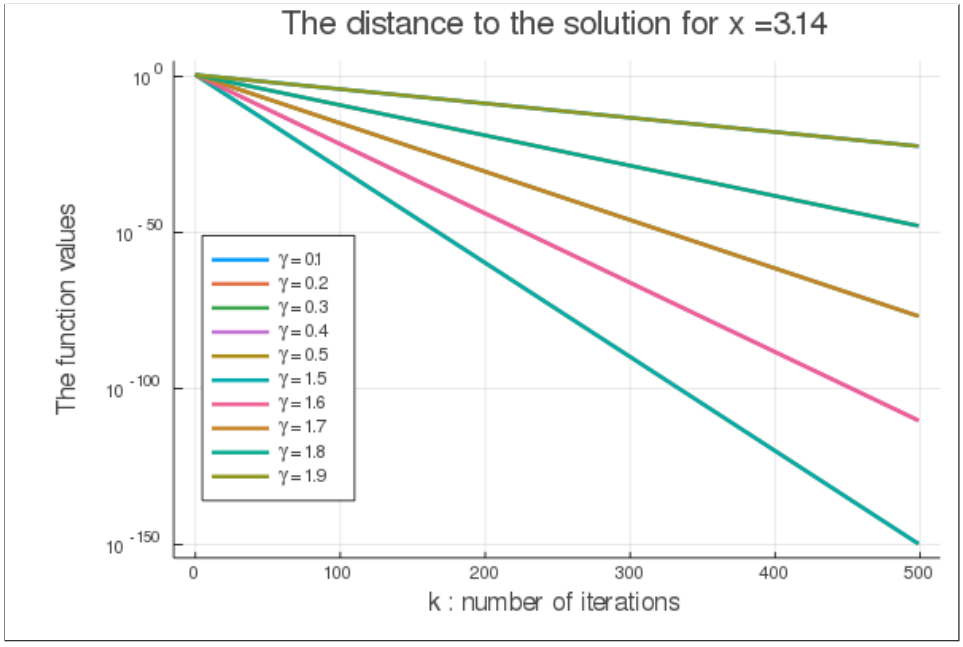
Result
As you can see the distances to the solution lineary converge.

Does initial point affect how close to the solution you can come?

I seems like decreasing the initial point makes the iterates close to the solution.

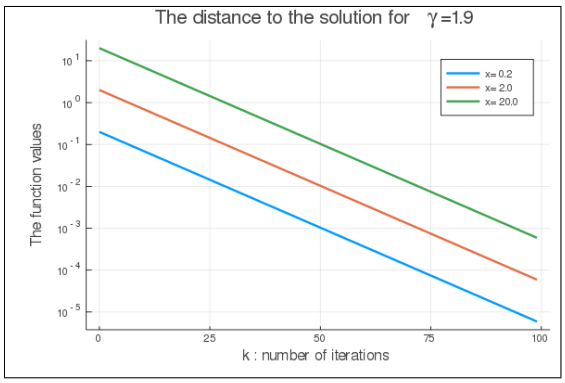
Startat $x = 3.14$ and search in (0,3) for the maximal step-size that still guarantee convergence

We try the following step-sizes $0.1 : 0.1 : 3$ for the initial point $x = 3.14$ and we plot only graphs that converge to lower numbers over iterations:



We know from the task 1 that the problem is L-smooth for any $L \geq 1$, then the step-size upper bound is $\gamma < \dfrac{2}{L} \leq 2$.
We choose $\gamma = 1.9$ as the maximal step from the labels that can guarantee the convergence
Remark you may see, there are more labels than graphs (since some graphs are mingled).

Try this step-size on the previous three initial points



If we use the same step-size $\gamma = 1.9$ as before, the iterates will converge to the solution, Then the step-size required for convergence does not depend on the initial point, but it takes more iterations than before since the linear rate is greater..

Task 6 Which property, apart from convexity, seems necessary?:

From the previous three tasks we notice that the gradient descent is applicable only for the third problem which is smooth.

Then it seems necessary to have smoothness apart from convexity in order to construct a globally convergent gradient descent algorithm.

Task 7 Solve (2) and (3) with the proximal point method:

Start at $x = 1$ and try several step-sizes in the interval $(0, \infty)$

It does not appear to exist an upper bound on the step-size in order to achieve convergence, see figures below:

Problem 2 :
$\gamma \in \{0.3, 3, 30, 300, 3000\}$
As we can see, the more we increase the step-size the fewer iterations needed to reach the solution, then the faster the sequence converges.
Notice that the convergence rate is the same (sublinear).

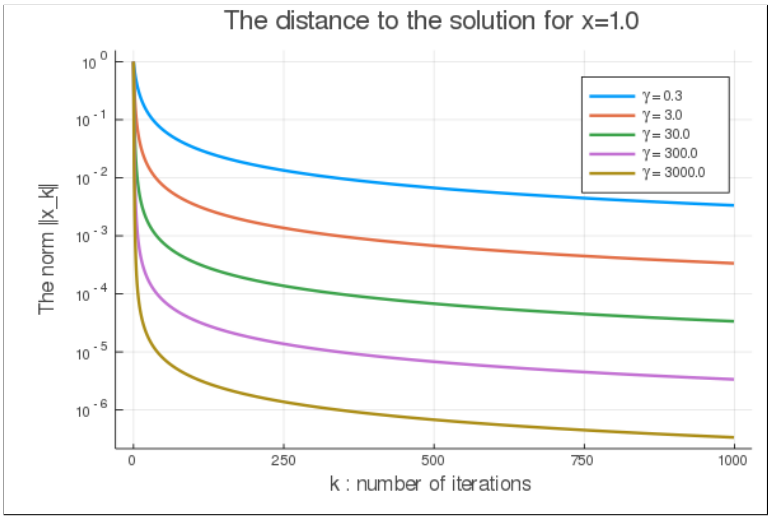
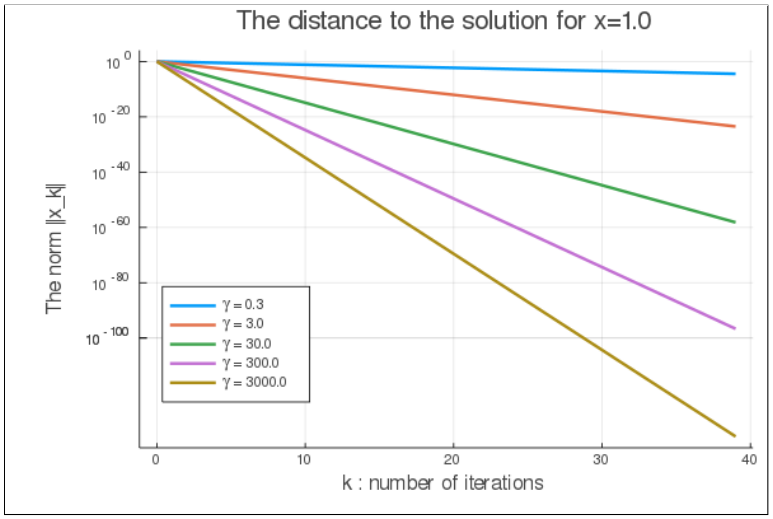


Figure 3: **Problem (2) starting with** $x = 1$



Problem 3 :
$\gamma \in \{0.3, 3, 30, 300, 3000\}$
The same thing for the third problem, increasing the step-size means better factor of the linear rate, then the faster we converge.

Figure 4: **Problem (3) starting with** $x = 1$

In order to explain why the step-size affects how fast we converge we use the definition of the prox:

$$\underset{\gamma f}{\text{prox}}(x^k) = \underset{z}{\text{argmin}}(f(z) + \frac{1}{2\gamma}||z - x^k||^2)$$

If one could set $\gamma = \infty$, we will get an entire minimization:

$$\underset{\gamma f}{\text{prox}}(x^k) = \underset{z}{\text{argmin}}(f(z))$$

Which explains how fast we converge when using large step-size.

Task 8 Solve (4) and (5) with the gradient descent method:

Problem (4): The sequence of gradients converges slower than lineary (see figure below).
We plot the distance between the gradient and zero for each iteration using the log scale:

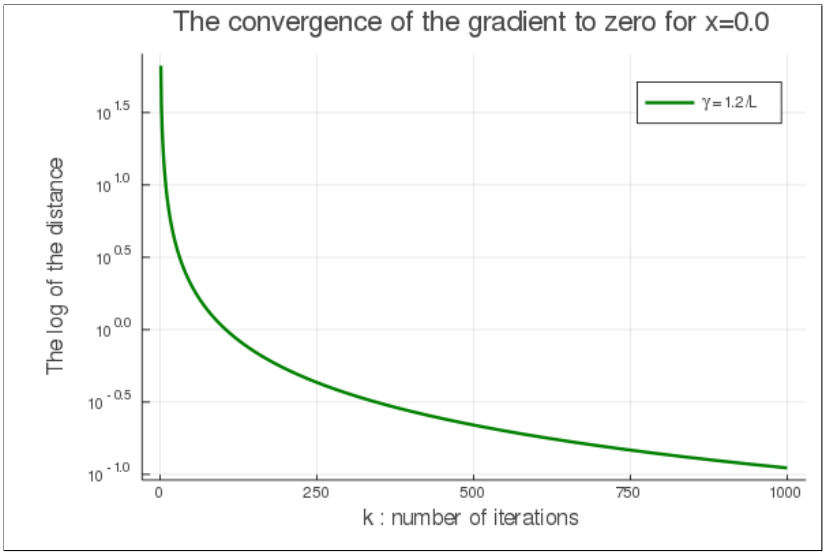


Figure 5: **The sub-linear convergence of the gradient**

Problem (5): The sequence of gradients lineary converges to zero (see figure below).
It begins slower and after a certain rank of iterations it becomes linear.

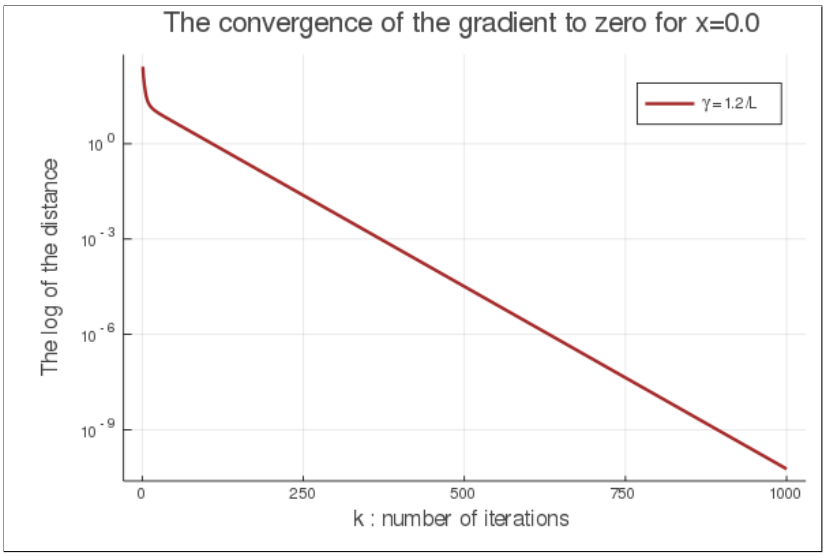


Figure 6: **The linear convergence of the gradient**

Which objective property seems responsible for achieving fast convergence?

We will exlain which property seems responsible for achieving fast convergence in the case of the problem (5) (since the convergence rate is linear).

We know that $x^{k+1} = x^k - \gamma \nabla f_5 x^k$ quad then:

$$||\nabla f_5(x^{k+1}|| = ||A^T(Ax^{k+1} - l)|| = ||A^T(Ax^k - l) - \gamma A^T A \nabla f_5(x^k)||$$

This is exactly:

$$||\nabla f_5(x^{k+1}|| = ||(I - \gamma A^T A)\nabla f_5(x^k|| \leq ||I - \gamma A^T A||.||\nabla f_5(x^k||$$

So in this case $\rho = ||I - \gamma A^T A|| = \left|1 - 1.2\frac{\lambda_{\min}(A^T A)}{\lambda_{\max}(A^T A)}\right|$ is factor of the linear rate, which is responsible for the fast convergence (Of course $A^T A$ must be positive definite).

Then **strong convexity and smoothness** keep linear convergence.

Task 9 Solve the least squares problem (5) with a prescaled gradient descent:

For each prescaling parameter we calculate the condition number $\kappa = \frac{\lambda_{\min}(\hat{A}^T \hat{A})}{\lambda_{\max}(\hat{A}^T \hat{A})}$ of the scaled problem:

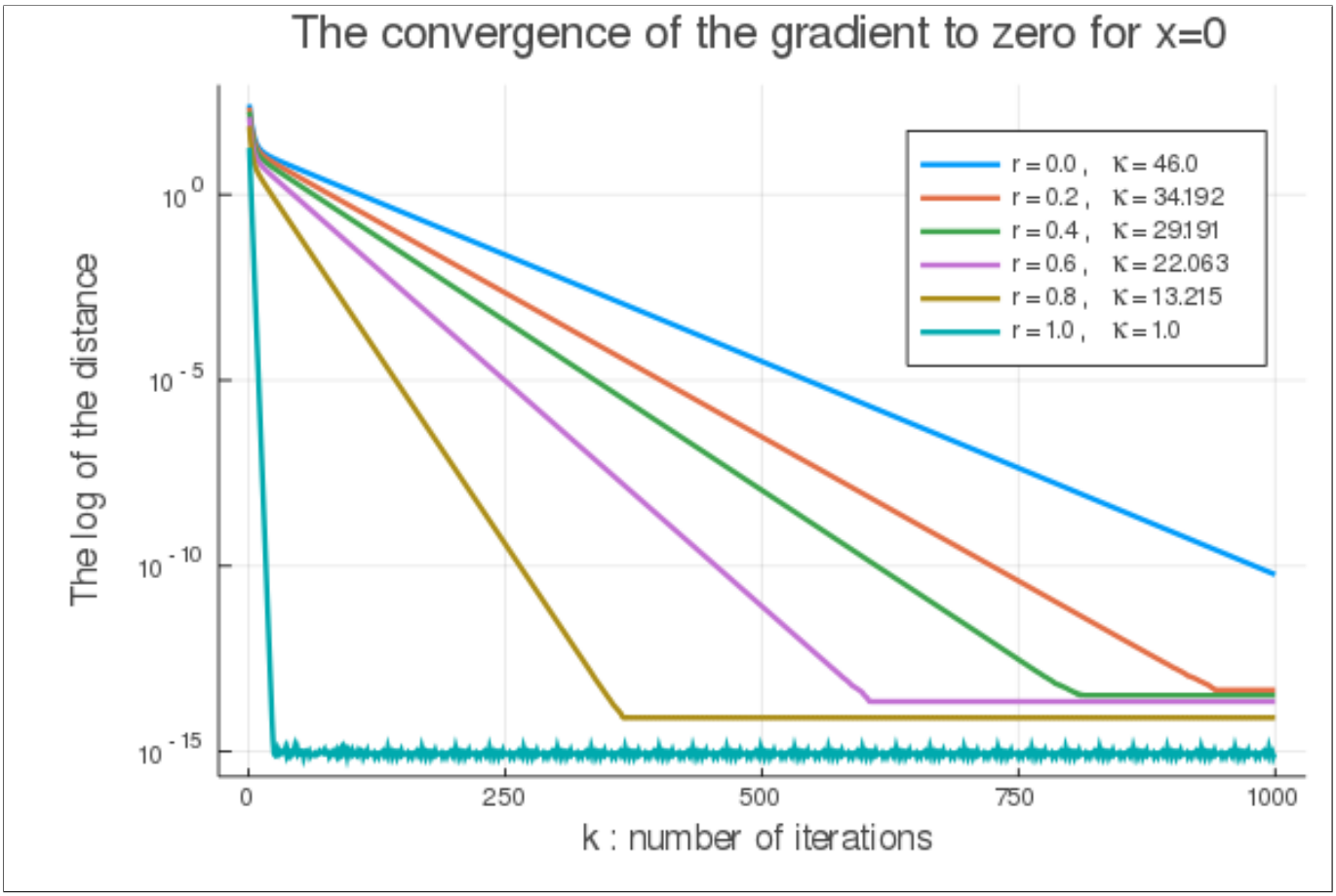


Figure 7: **The linear convergence of the gradient**

Result:

When increasing the prescaling parameter r, the condition number decreases and the convergence speed increases since the convergence becomes faster than before (The more we increase r the better performance we get).