Technical University of Crete
School of Electrical and Computer Engineering

Course: **Optimization**

Exercise Bonus(1) (50/1000)

Angelopoulos Dimitris, 2020030038

In this exercise, we will use the the projections onto convex sets to develop the (accelerated) projected gradient algorithm for constrained quadratic problems.

Let $f : \mathbb{R}^n \to \mathbb{R}$, defined as :

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x}$$

With $\mathbf{P} \in \mathbb{S}^n_{++}$ and $\mathbf{q} \in \mathbb{R}^n$. Let $L = \lambda_{max}(\mathbf{P})$ and $l = \lambda_{min}(\mathbf{P})$.

We consider the following quadratic problem :

$$(P) \qquad \underset{\mathbf{x}\in\mathbb{S}}{\text{minimize}}\, f(\mathbf{x})$$

for

(a) $\mathbb{S} = \mathbb{R}^n_+$

(b) $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}\}$

(c) $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b}\}$, with $\mathbf{A} \in \mathbb{R}^{p\times n}$, $\mathbf{b} \in \mathbb{R}^p$

(d) $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_2 \leq c\}$, with $c \in \mathbb{R}_{++}$

The algorithms that are to be used are the following.

Algorithm 1 Projected Gradient Algorithm

$\mathbf{x}_0 \in \mathbb{R}^n$, $k = 0$, $t = \frac{1}{L}$

While $(\frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2}{\|\mathbf{x}_k\|_2} > \epsilon)$

1. $\Delta \mathbf{x}_k := -\nabla f(\mathbf{x}_k)$
2. $\mathbf{x}_{k+1} = \mathcal{P}_{\mathbb{S}} \left(\mathbf{x}_k + t\Delta \mathbf{x}_k\right)$
3. $k := k + 1$

Algorithm 2 Accelerated Projected Gradient Algorithm

$\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{y}_0 = \mathbf{x}_0$, $k = 0$, $t = \frac{1}{L}$, $\beta = \frac{\sqrt{L} - \sqrt{l}}{\sqrt{L} + \sqrt{l}}$

While $(\frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2}{\|\mathbf{x}_k\|_2} > \epsilon)$

1. $\Delta \mathbf{y}_k := -\nabla f(\mathbf{y}_k)$
2. $\mathbf{x}_{k+1} = \mathcal{P}_{\mathbb{S}} \left(\mathbf{y}_k + t\Delta \mathbf{y}_k\right)$
3. $\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \beta(\mathbf{x}_{k+1} - \mathbf{x}_k)$
4. $k := k + 1$

Now we will compute every projection operator $\mathcal{P}_{\mathbb{S}}$ for every case and we will compare the algorithms above for each one.

(a) The projection operator for this case is given by the following expression :

$$\mathcal{P}_{\mathbb{S}}(\mathbf{x}) = max\{\mathbf{0}, \mathbf{x}\}$$

We set $L = 100$ and $l = 1$. After constructing the matrix $\mathbf{P}$ and choosing $\mathbf{q}$ randomly we use CVX to find solution $p_* = -0.7897$.

Now we will solve the problem using Algorithms 1 and 2 respectively. First of all we print our results in a contour plot as shown in Fig. 1.

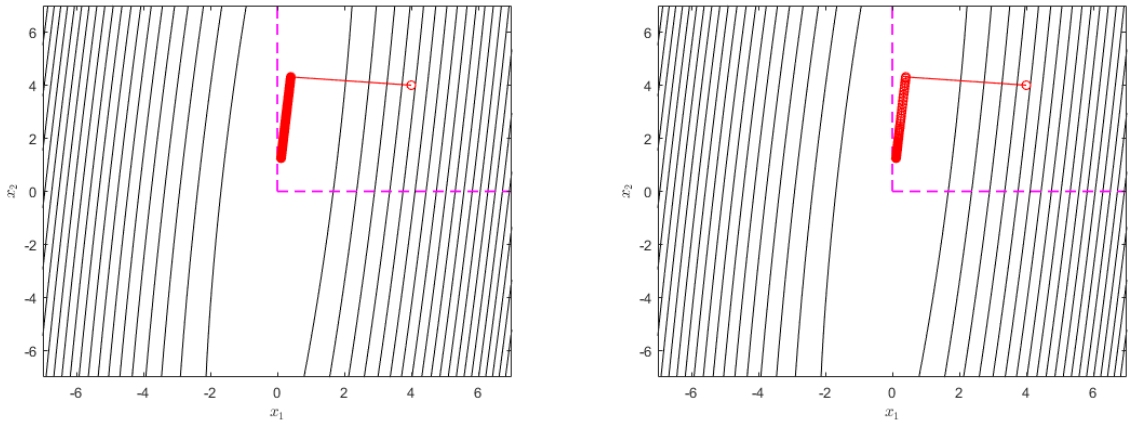

Fig. 1

On the left we can see the trajectory created by the projected gradient algorithm, while on the right we can see the trajectory of the accelerated gradient algorithm. The solution that both algorithms yield is $p_* = -0.7897$.

The common semilogy plot for each algorithm, for $\mathcal{K} = 100$ and $n = 2$ is

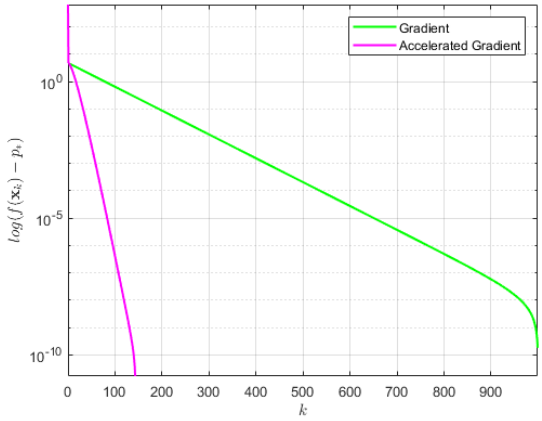

Fig. 2

We observe that the accelerated projected gradient converges almost 10 times faster than the common projected gradient algorithm, the majority of the times.

Increasing the dimensions from $n = 2$ to $n = 10$(left) and $n = 200$(right) with the same initial conditions but different realizations of f, we get the following figure.

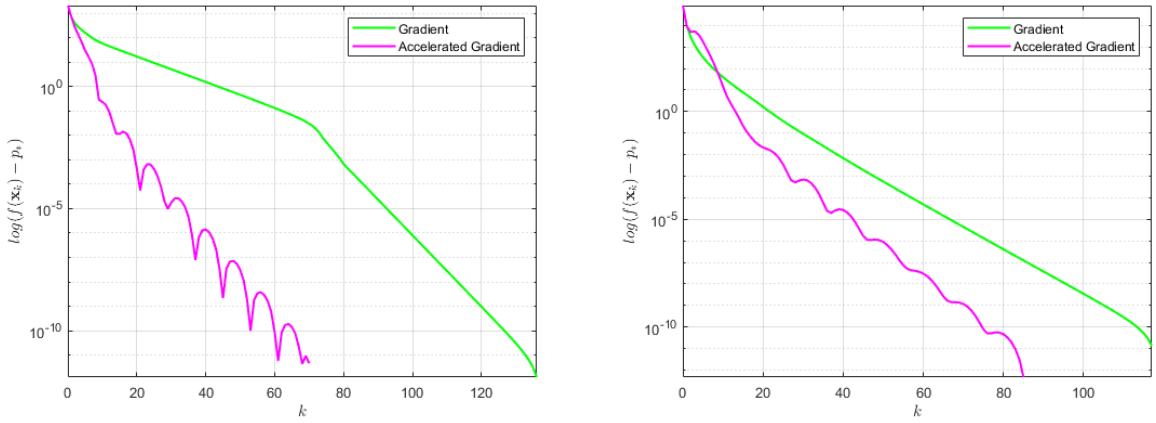

Fig. 3

For higher dimensions the accelerating gradient algorithm still guarantees us better convergence than the simple one.

(b) For this case we computed the projection to be the following :

$$\mathcal{P}_{\mathbb{S}}(\mathbf{x}) = \begin{cases} min\{\mathbf{b}, \mathbf{x}\}, & \mathbf{x} > \mathbf{a} \\ max\{\mathbf{a}, \mathbf{x}\}, & \mathbf{x} < \mathbf{b} \end{cases}$$

Solving the problem via CVX we get the solution $p_* = -0.0145$

Now as previously we will plot the sequence $\mathbf{x}_k$ on a contour for 2 dimensions.

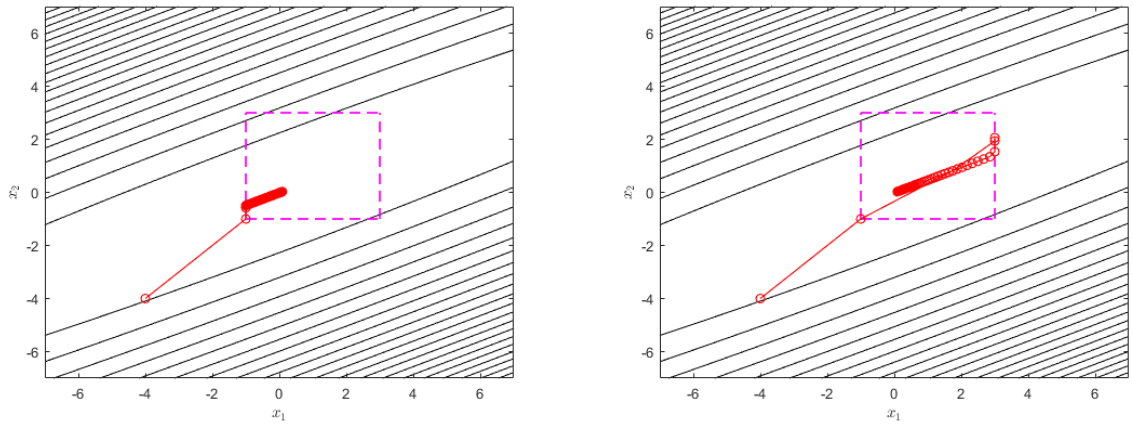

Fig. 4

We can indeed verify that both algorithms converge even for initial conditions outside of the feasible set. The solutions that both algorithms produce is $p_* = -0.0145$. Moreover in Fig. 4 we see the effect of the "acceleration". The right trajectory has large steps initially reaching the opposite "side" of the feasible set while still maintaining significantly less amount of steps as shown in the Fig. 5 below.

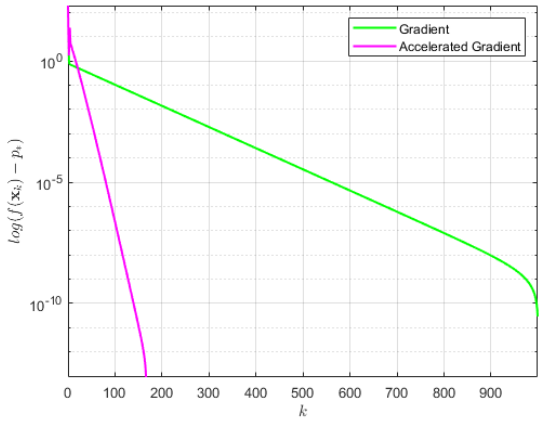

Fig. 5

As we saw in case (a) the gradient algorithm iterates 1000 times, while the accelerated one needs less than 200 iterations to converge.

As we increase dimensions the same properties hold. For the majority of the different realizations of f we get the following convergence results.

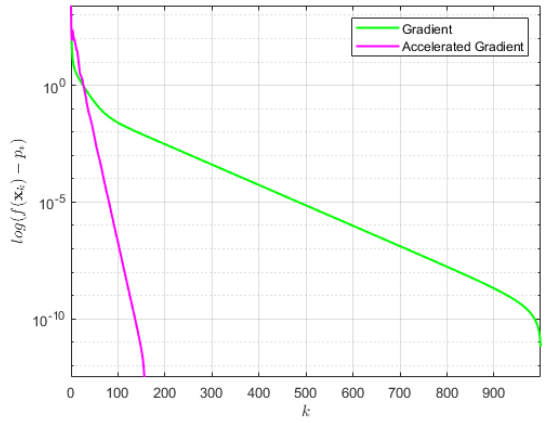

Fig. 6

(c) In this case we have affine equality constraints. The projection operator is the following :

$$\mathcal{P}_{\mathbb{S}}(\mathbf{x}) = \mathbf{F}^{-1}(1:n,:)\begin{bmatrix}\mathbf{x}\\ \mathbf{b}\end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix}\mathbb{I}_{n\times n} & \mathbf{A}^T\\ \mathbf{A} & \mathbf{O}\end{bmatrix}$$

The CVX solution is $p_* = 30.74$. For 2 dimensions and 1 equality constraint the algorithms have the following trajectories respectively.

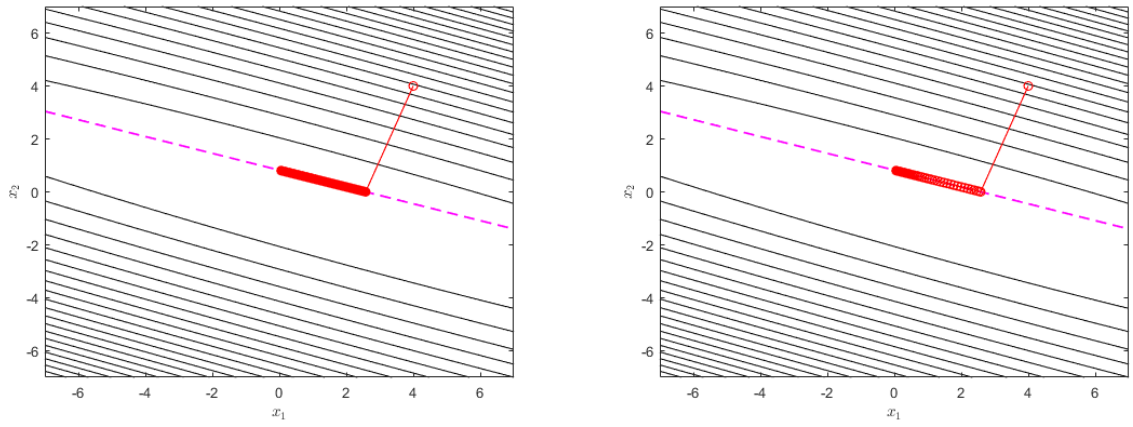

Fig. 7

The convergence results of these 2 algorithms are the following.

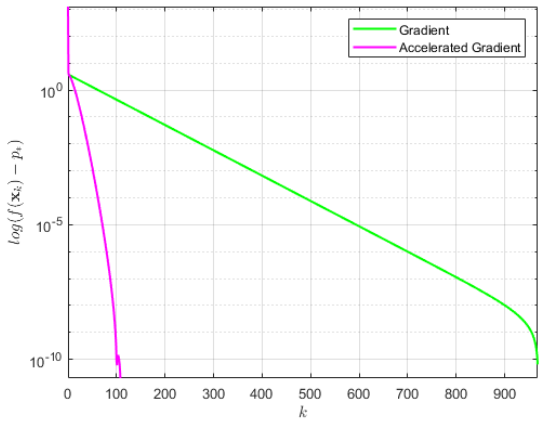

Fig. 8

The accelerated gradient method achieves much better performance, as expected. Although for different constraints there exist some possibilities where the simple projected gradient method can converge at the same rate with the accelerated one.

Now we increase the dimension while keeping the same number of constraints.

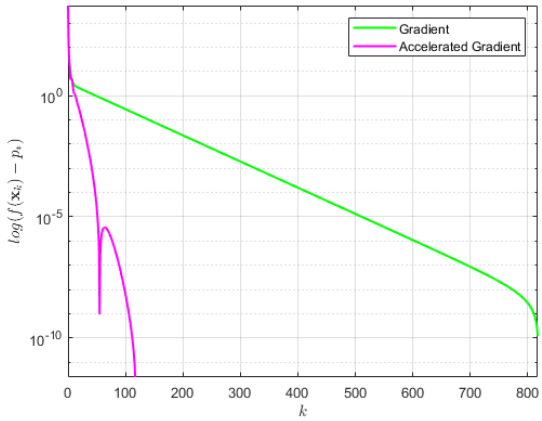

Fig. 9

We observe that the performance stays almost the same.

Last but not least we increase the dimensions even more and set 20 affine equality constraints to get the following results. The convergence results of these 2 algorithms are the following.

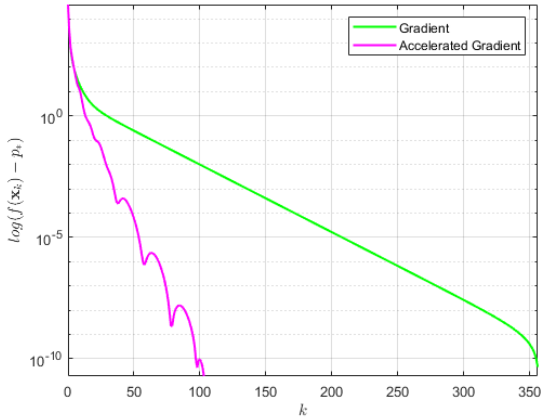

Fig. 10

In this case the accelerated gradient algorithm still has greater convergence rate. The difference we observe is that the convergence rates do not have such a significant difference as the we saw in the previous cases.

(d) For the last case we compute the projection as shown below :

$$\mathcal{P}_{\mathbb{S}}(\mathbf{x}) = \begin{cases} \frac{c}{\|\mathbf{x}\|}\mathbf{x}, & \|\mathbf{x}\| > c \\ \mathbf{x}, & \|\mathbf{x}\| \leq c \end{cases}$$

Choosing $c = 2$, CVX yields the solution $p_* = -0.1822$. The 2 algorithms we developed converge to the same solution.

The trajectories for the last case are shown in Fig. 11.

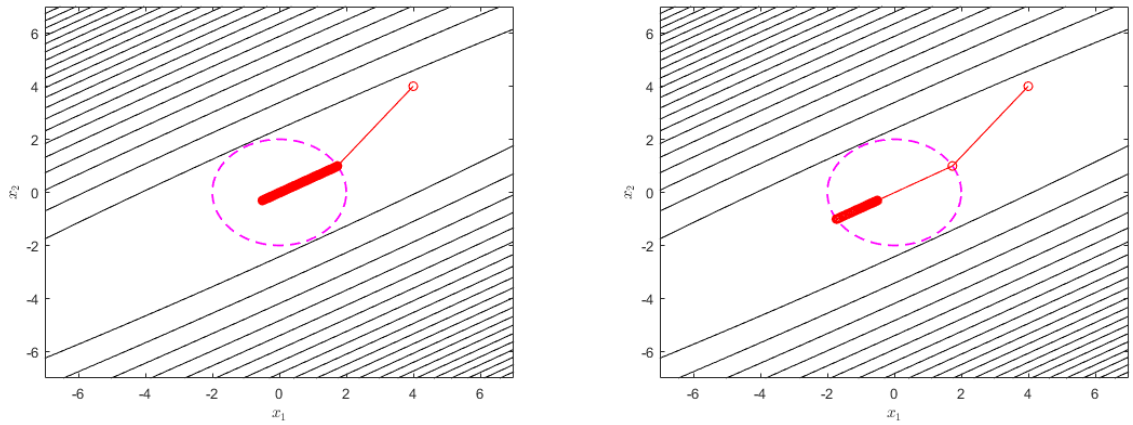

Fig. 11

We observe the same phenomenon as in case (b) where the accelerated trajectory reaches the opposite side of the feasible boundary.

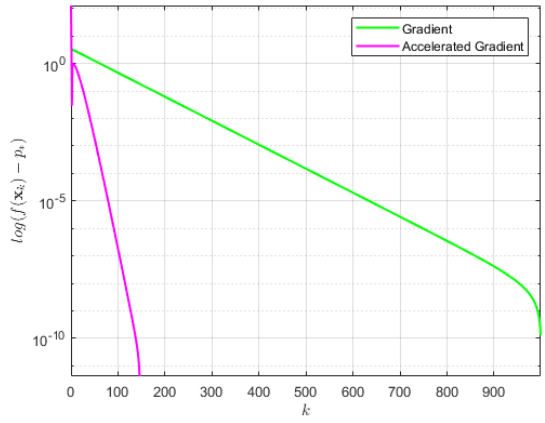

Fig. 12

The convergence results are the expected ones as we can verify in the figure above.

Increasing the dimensions the convergence properties hold. The majority of the implementations of the algorithms for higher dimensions has the the following semilogy plot.

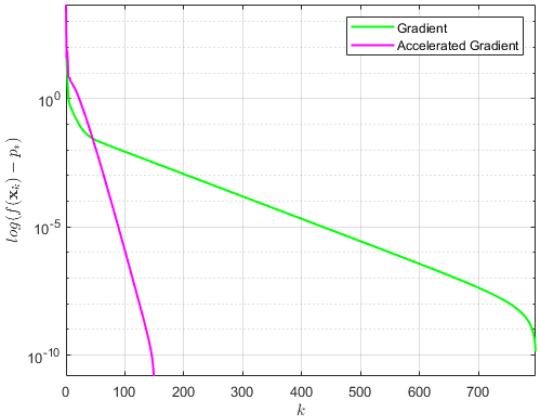

Fig. 13