First-order methods in optimization - Evaluation

Olivier Leblanc, Guillaume Thiran.

1 Part 1 - Slide 45

1.1 6. (with code)

$$f(\boldsymbol{x}) = 2x_{[1]} + x_{[2]} = \max_{\boldsymbol{y}} \left\{ \sum_i y_i x_i ; \sum_i y_i = 3, 0 \leq y_i \leq 2 \right\} = \sigma_{\{\boldsymbol{y}|\boldsymbol{1}^\top \boldsymbol{y}=3, \boldsymbol{0} \leq \boldsymbol{y} \leq 2\boldsymbol{1}\}}.$$

Hence,

$$\mathrm{prox}_f(\boldsymbol{x}) = \boldsymbol{x} - \mathcal{P}_{\{\boldsymbol{y}|\boldsymbol{1}^\top \boldsymbol{y}=3, \boldsymbol{0} \leq \boldsymbol{y} \leq 2\boldsymbol{1}\}}(\boldsymbol{x}).$$

Writing $C = \{\boldsymbol{y}|\boldsymbol{1}^\top \boldsymbol{y} = 3, \boldsymbol{0} \leq \boldsymbol{y} \leq 2\boldsymbol{1}\}$, it can be compared with $H_{\boldsymbol{a},b} \bigcap \mathrm{Box}[\boldsymbol{l}, \boldsymbol{u}]$, with $\boldsymbol{a} = \boldsymbol{1}, b = 3, \boldsymbol{l} = \boldsymbol{0}, \boldsymbol{u} = \boldsymbol{2}$.

```python
import numpy as np
def error_fct(a,b,l,u,x,mu):
    y = projbox(x-mu*a,l,u)
    return error = a@y-b

def projbox(x, l, u):
    return np.minimum(np.maximum(x,l), u)

def proj_H_inter_box(a,b,l,u,x):
    mu_low = -1 #start with guesses for mu-levels
    mu_high = 1
    #check that the levels give respectively negative and positive values
    #for the function error = a@y-1 with y = proj_box(l,u,x-mu*a)
    #positive for lower bound
    j=0
    j_max =100
    error_l = -1
    while (error_l<0) & (j<j_max) :
        mu_low = mu_low*2 #more negative
        error_l=error_fct(a,b,l,u,x,mu_low)
        j=j+1
    #negative for upper bound
    k=0
    k_max =10
    error_h = 1
    while (error_h>0) & (k<k_max) :
        mu_high = mu_high*2 #more positive
        error_h=error_fct(a,b,l,u,x,mu_high)
        k=k+1

    i = 0
    i_max = 100
    tol = 1e-8
    error = 2*tol
    while (np.abs(error)>tol) & (i<i_max) :
        mu_mid = (mu_low+mu_high)/2
        error = error_fct(a,b,l,u,x,mu_mid)
        if error>0:
            mu_low = mu_mid
        else:
            mu_high = mu_mid
        i=i+1

    #Compute the solution with the good level
    return projbox(x-mu_mid*a,l,u)

def proxf(x):
    return x - proj_H_inter_box(np.ones(len(x)),3,np.zeros(len(x)),2*np.ones(len(x)),x)

x = np.array([2,1,4,1,2,1])
```

```
print(proxf(x))
```

The output is : $[1.5, 1., 2., 1., 1.5, 1.]$.

1.2 8.

$$f(t) = \begin{cases} 1/t, & t > 0, \\ \infty, & \text{else.} \end{cases}$$

$$\mathrm{prox}_{\lambda f}(t) = \arg\min_u \begin{cases} \lambda/u, & u > 0, \\ \infty, & \text{else.} \end{cases} + \tfrac{1}{2}\|u - t\|_2^2$$

Clearly, the minimum occurs when $u > 0$, i.e. on the differentiable part. Hence,

$$\frac{-\lambda}{u^2} + u - t = 0 \Leftrightarrow u^3 - tu^2 - \lambda = 0,$$

and it can be checked the second derivative is always positive on $u > 0$, implying it exists a unique solution of the above and it corresponds to a minimum. Finally,

$$\mathrm{prox}_{\lambda f}(t) = \{u > 0 | u^3 - tu^2 - \lambda = 0\}. \tag{1}$$

1.3 9.

$$f(\boldsymbol{X}) = \begin{cases} \mathrm{tr}\, \boldsymbol{X}^{-1}, & \boldsymbol{X} \succ 0, \\ \infty, & \text{else.} \end{cases} = \begin{cases} \sum_{i=1}^n \frac{1}{\lambda_i}, & \boldsymbol{X} \succ 0, \\ \infty, & \text{else.} \end{cases}$$

As one can write $f(\boldsymbol{X}) = g(\lambda_1(\boldsymbol{X}), \cdots, \lambda_n(\boldsymbol{X})) = \sum_{i=1}^n h(\lambda_i)$, f is a *symmetric spectral function.* With the EigenValue Decomposition (EVD) of $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}\, \mathrm{diag}\, \lambda(\boldsymbol{X})\boldsymbol{U}^T$, this gives

$$\begin{aligned} \mathrm{prox}_{\lambda f}(\boldsymbol{X}) &= \boldsymbol{U}\, \mathrm{diag}(\mathrm{prox}_{\lambda g}[\lambda_1, \cdots, \lambda_n])\boldsymbol{U}^\top \\ &= \boldsymbol{U}\, \mathrm{diag}(\mathrm{prox}_{\lambda h}(\lambda_1), \cdots, \mathrm{prox}_{\lambda h}(\lambda_n))\boldsymbol{U}^\top \\ &= \boldsymbol{U}\, \mathrm{diag}(\{\{u > 0 | u^3 - \lambda_i u^2 - \lambda = 0\}\}_{i=1}^n)\boldsymbol{U}^\top, \end{aligned}$$

with $h(t) = \frac{1}{t}$ for $t > 0$.

1.4 10.

$$\lambda f(\boldsymbol{x}) = \lambda(\|\boldsymbol{x}\|_2 - 1)^2 = \lambda\|\boldsymbol{x}\|_2^2 - 2\lambda\|\boldsymbol{x}\|_2 + \lambda.$$

Using the provided tables, one identifies it with $g(\boldsymbol{x}) + \frac{c}{2}\|\boldsymbol{x}\|_2^2 + \langle \boldsymbol{a}, \boldsymbol{x}\rangle + \gamma$, with $g(\boldsymbol{x}) = -2\lambda\|\boldsymbol{x}\|_2, c = 2\lambda, \boldsymbol{a} = \boldsymbol{0}, \gamma = 0$. Hence,

$$\mathrm{prox}_{\lambda f}(\boldsymbol{x}) = \mathrm{prox}_{\frac{-2\lambda\|\cdot\|_2}{1+2\lambda}}\left(\frac{\boldsymbol{x}}{1+2\lambda}\right) = \begin{cases} \left(1 + \frac{2\lambda}{\|\boldsymbol{x}\|_2}\right)\frac{\boldsymbol{x}}{1+2\lambda}, & \boldsymbol{x} \neq \boldsymbol{0}, \\ \{\boldsymbol{u} : \|\boldsymbol{u}\| = \frac{2\lambda}{1+2\lambda}\}, & \boldsymbol{x} = \boldsymbol{0}. \end{cases}$$

2 Part 2 - Exercise 0 - slide 40

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \underbrace{\frac{\lambda_1}{2}\|\boldsymbol{x}\|_2^2 + \lambda_2\|\boldsymbol{x}\|_1}_{\text{elastic net}},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ and $\lambda_1, \lambda_2 > 0$. Choosing $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\lambda_1}{2}\|\boldsymbol{x}\|_2^2$ as the σ-strongly convex and differentiable part and $g(\boldsymbol{x}) = \lambda_2\|\boldsymbol{x}\|_1$ as the closed convex but non differentiable part, one has $\nabla f(\boldsymbol{x}) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) + \lambda_1\boldsymbol{x}$ and $\mathrm{prox}_{\alpha g}(\boldsymbol{x}) = \mathcal{T}_{\alpha\lambda_2}(\boldsymbol{x})$.

- (*Proximal Gradient*)

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \mathrm{prox}_{\frac{1}{L}g}(\boldsymbol{x}^k - \tfrac{1}{L}\nabla f(\boldsymbol{x}^k)) \\ &= \mathcal{T}_{\frac{\lambda_2}{L}}\left(\boldsymbol{x}^k - \tfrac{1}{L}(\boldsymbol{A}^\top\boldsymbol{A} + \lambda_1\boldsymbol{I})\boldsymbol{x}^k + \tfrac{1}{L}\boldsymbol{A}^\top\boldsymbol{b}\right), \end{aligned}$$

  with $L = \left\|\boldsymbol{A}^\top\boldsymbol{A} + \lambda_1\boldsymbol{I}\right\| = \lambda_{\max}(\boldsymbol{A}^\top\boldsymbol{A}) + \lambda_1 = \|\boldsymbol{A}\|_2^2 + \lambda_1$.

- (*FISTA*)

$$\begin{cases} \boldsymbol{x}^{k+1} & = \mathcal{T}_{\frac{\lambda_2}{L}} \left(\boldsymbol{y}^k - \frac{1}{L}(\boldsymbol{A}^\top \boldsymbol{A} + \lambda_1 \boldsymbol{I}) \boldsymbol{y}^k + \frac{1}{L} \boldsymbol{A}^\top \boldsymbol{b} \right) \\ t_{k+1} & = \frac{1+\sqrt{1+4t_k^2}}{2} \\ \boldsymbol{y}^{k+1} & = \boldsymbol{x}^{k+1} + \left(\frac{t_k - 1}{t_{k+1}} \right) (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \end{cases}$$

- (*V-FISTA*)

$$\begin{cases} \boldsymbol{x}^{k+1} & = \mathcal{T}_{\frac{\lambda_2}{L}} \left(\boldsymbol{y}^k - \frac{1}{L}(\boldsymbol{A}^\top \boldsymbol{A} + \lambda_1 \boldsymbol{I}) \boldsymbol{y}^k + \frac{1}{L} \boldsymbol{A}^\top \boldsymbol{b} \right) \\ \boldsymbol{y}^{k+1} & = \boldsymbol{x}^{k+1} + \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa}+1} \right) (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k), \end{cases}$$

  with $\kappa = L/\sigma$, and $\sigma = \lambda_{\min}(\boldsymbol{A}^\top \boldsymbol{A}) + \lambda_1 = \lambda_1$ if $\boldsymbol{A}$ is not full rank.

Now, if one chooses $f(\boldsymbol{x}) = \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ as the differentiable (but not strongly convex) part and $\alpha g(\boldsymbol{x}) = \frac{\alpha\lambda_1}{2} \|\boldsymbol{x}\|_2^2 + \alpha\lambda_2 \|\boldsymbol{x}\|_1$ as the closed convex but non differentiable part, one has $\nabla f(\boldsymbol{x}) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$ and, by identification with $g'(\boldsymbol{x}) + \frac{c}{2}\|\boldsymbol{x}\|_2^2 + \langle \boldsymbol{a}, \boldsymbol{x} \rangle + \gamma$ with $g'(\boldsymbol{x}) = \alpha\lambda_2 \|\boldsymbol{x}\|_1$, $c = \alpha\lambda_1, \boldsymbol{a} = \boldsymbol{0}, \gamma = 0$, $\text{prox}_{\alpha g}(\boldsymbol{x}) = \text{prox}_{\frac{\alpha}{\alpha\lambda_1+1}\lambda_2\|\cdot\|_1}(\frac{\boldsymbol{x}}{\alpha\lambda_1+1}) = \mathcal{T}_{\frac{\alpha\lambda_2}{\alpha\lambda_1+1}}(\frac{\boldsymbol{x}}{\alpha\lambda_1+1})$.

- (*V-FISTA2*)

$$\begin{cases} \boldsymbol{x}^{k+1} & = \mathcal{T}_{\frac{\lambda_2/L_2}{\lambda_1/L_2+1}} \left(\frac{\boldsymbol{y}^k - \frac{1}{L_2}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{y}^k - \boldsymbol{b})}{\frac{\lambda_1}{L_2}+1} \right) = \mathcal{T}_{\frac{\lambda_2}{\lambda_1+L_2}} \left(\frac{L_2\boldsymbol{y}^k - \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{y}^k - \boldsymbol{b})}{\lambda_1 + L_2} \right) \\ \boldsymbol{y}^{k+1} & = \boldsymbol{x}^{k+1} + \left(\frac{\sqrt{\kappa_2} - 1}{\sqrt{\kappa_2}+1} \right) (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k), \end{cases}$$

  with $L_2 = \|\boldsymbol{A}\|_2^2$, $\kappa_2 = L_2/\sigma$, and $\sigma = \lambda_1$ (given by the statement).

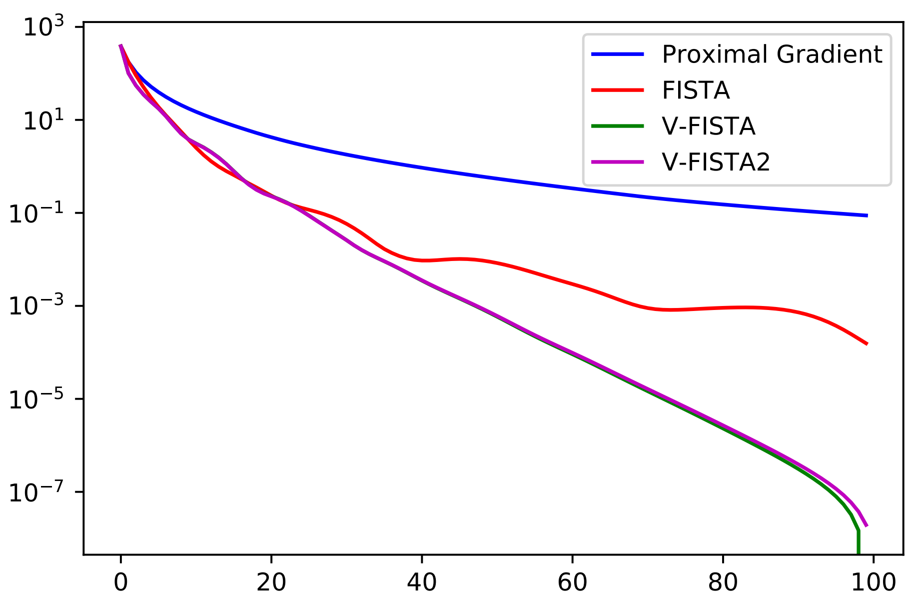

Figure 1: $F(\boldsymbol{x}^k) - F_{\text{opt}}$ in log-scale along the y-axis for the first 100 iterations of each of the methods with all-zeros vectors and a constant stepsize.

As can be observed in Fig. 1, the V-FISTA methods worked the best, with a slight improvement in V-FISTA which does not follow the theory. The four first elements are:

- (-0.43969331 0.01974521 1.42280231 -0.87819581)
- (-0.4319773 0.02881602 1.43373682 -0.9066518)

- (-0.43210834 0.0295975 1.43437285 -0.90583213)
- (-0.43210892 0.02959727 1.43437048 -0.9058291)

for the four methods, PG, FISTA, V-FISTA, V-FISTA2, respectively.

3 Part 2 - Exercise 1 - slide 41

$$\min_{\boldsymbol{x}\in\mathbb{R}^{30}} \sqrt{\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c}+0.2\left\|\boldsymbol{D}\boldsymbol{x}+\boldsymbol{1}\right\|_1,$$

where $\boldsymbol{Q}\in\mathbb{R}^{30\times 30}$, $\boldsymbol{b}\in\mathbb{R}^{30}$, $c\in\mathbb{R}$, $\boldsymbol{D}\in\mathbb{R}^{30\times 30}$. The matrix $\boldsymbol{Q}$ is positive definite.

(a) The first step of the problem is to show it is well-defined (i.e. $\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c\geq 0$ if $c>\boldsymbol{b}^\top\boldsymbol{Q}^{-1}\boldsymbol{b}$.). To that aim, letting the Cholesky factorisation of $\boldsymbol{Q}$ being denoted as $\boldsymbol{Q}=\boldsymbol{L}^\top\boldsymbol{L}$,

$$\begin{aligned}
\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c &= \boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{b}^\top\boldsymbol{L}^{-1}\boldsymbol{L}\boldsymbol{x}+\boldsymbol{x}^\top\boldsymbol{L}^\top\boldsymbol{L}^{-\top}\boldsymbol{b}+\boldsymbol{b}^\top\boldsymbol{L}^{-1}\boldsymbol{L}^{-\top}\boldsymbol{b}-\boldsymbol{b}^\top\boldsymbol{L}^\top\boldsymbol{L}^{-\top}\boldsymbol{b}+c,\\
&= \left\|\boldsymbol{L}\boldsymbol{x}+\boldsymbol{L}^{-\top}\boldsymbol{b}\right\|_2^2+c-\boldsymbol{b}^\top\boldsymbol{Q}^{-1}\boldsymbol{b}.
\end{aligned}$$

From the above, as a norm is always nonnegative, one can conclude the problem is well-defined if $c>\boldsymbol{b}^\top\boldsymbol{Q}^{-1}\boldsymbol{b}$.

(b) Starting from the norm expression, one can obtain

$$\sqrt{\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c}+0.2\left\|\boldsymbol{D}\boldsymbol{x}+\boldsymbol{1}\right\|_1=\left\|\begin{matrix}\boldsymbol{L}\boldsymbol{x}+\boldsymbol{L}^{-\top}\boldsymbol{b}\\ \sqrt{c-\boldsymbol{b}^\top\boldsymbol{Q}^{-1}\boldsymbol{b}}\end{matrix}\right\|_2+0.2\left\|\boldsymbol{D}\boldsymbol{x}+\boldsymbol{1}\right\|_1.$$

As a norm is convex, as a composition with a linear mapping preserves convexity and since a sum of convex functions is convex, the problem is convex.

(c) To fit the framework of FISTA, we denote by $f(\boldsymbol{x})=\sqrt{\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c}$ and $g(\boldsymbol{x})=0.2\left\|\boldsymbol{D}\boldsymbol{x}+\boldsymbol{1}\right\|_1$ where g is proper, closed and convex, and f is L_f-smooth and convex. More precisely, the Lipschitz constant of f can be identified by computing its Hessian:

$$\begin{aligned}
\boldsymbol{\nabla} f(\boldsymbol{x}) &= \frac{\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{b}}{\sqrt{\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c}},\\
\boldsymbol{\nabla}^2 f(\boldsymbol{x}) &= \frac{\boldsymbol{Q}}{\sqrt{\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c}}-\frac{(\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{b})(\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{b})^\top}{\left(\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c\right)^{\frac{3}{2}}}.
\end{aligned}$$

The latter can be bounded as

$$\boldsymbol{\nabla}^2 f(\boldsymbol{x})\preceq\frac{\boldsymbol{Q}}{\sqrt{\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x}+2\boldsymbol{b}^\top\boldsymbol{x}+c}}=\frac{\boldsymbol{Q}}{\sqrt{c-\boldsymbol{b}^\top\boldsymbol{Q}^{-1}\boldsymbol{b}}\sqrt{1+\frac{\left\|\boldsymbol{L}\boldsymbol{x}+\boldsymbol{L}^{-\top}\boldsymbol{b}\right\|_2^2}{c-\boldsymbol{b}^\top\boldsymbol{Q}^{-1}\boldsymbol{b}}}}\preceq\frac{\boldsymbol{Q}}{\sqrt{c-\boldsymbol{b}^\top\boldsymbol{Q}^{-1}\boldsymbol{b}}},$$

leading to $L_f=\frac{\lambda_{\max}(\boldsymbol{Q})}{\sqrt{c-\boldsymbol{b}^\top\boldsymbol{Q}^{-1}\boldsymbol{b}}}$. In the case of the exercise, we obtain $L_f=53.54$ and thus a step size of 0.019.
As $\boldsymbol{D}\boldsymbol{D}^\top=\boldsymbol{I}$, one can compute the proximal operator of αg as

$$\begin{aligned}
\operatorname{prox}_{\alpha g}(\boldsymbol{x}) &= \boldsymbol{x}+\boldsymbol{D}^\top\left(\mathcal{T}_{0.2\alpha}(\boldsymbol{D}\boldsymbol{x}+\boldsymbol{1})-\boldsymbol{D}\boldsymbol{x}-\boldsymbol{1}\right),\\
&= \boldsymbol{D}^\top\mathcal{T}_{0.2\alpha}(\boldsymbol{D}\boldsymbol{x}+\boldsymbol{1})-\boldsymbol{D}^\top\boldsymbol{1}.
\end{aligned}$$

This leads to

- (*Proximal gradient*):

$$\begin{aligned}
\boldsymbol{x}^{k+1} &= \operatorname{prox}_{\frac{1}{L_f}g}(\boldsymbol{x}^k-\tfrac{1}{L_f}\nabla f(\boldsymbol{x}^k))\\
&= \boldsymbol{D}^\top\mathcal{T}_{\frac{0.2}{L_f}}\left[\boldsymbol{D}(\boldsymbol{x}^k-\tfrac{1}{L_f}\nabla f(\boldsymbol{x}^k))+\boldsymbol{1}\right]-\boldsymbol{D}^\top\boldsymbol{1}\\
&= \boldsymbol{D}^\top\mathcal{T}_{\frac{0.2}{L_f}}\left[\boldsymbol{D}(\boldsymbol{x}^k-\tfrac{1}{L_f}\frac{\boldsymbol{Q}\boldsymbol{x}^k+\boldsymbol{b}}{\sqrt{(\boldsymbol{x}^k)^\top\boldsymbol{Q}\boldsymbol{x}^k+2\boldsymbol{b}^\top\boldsymbol{x}^k+c}})+\boldsymbol{1}\right]-\boldsymbol{D}^\top\boldsymbol{1}.
\end{aligned}$$

- (*FISTA*):

$$\begin{cases} \boldsymbol{x}^{k+1} & = \boldsymbol{D}^\top \mathcal{T}_{\frac{0.2}{L_f}} \left[\boldsymbol{D}(\boldsymbol{y}^k - \frac{1}{L_f} \frac{\boldsymbol{Q}\boldsymbol{y}^k + \boldsymbol{b}}{\sqrt{(\boldsymbol{y}^k)^\top \boldsymbol{Q}\boldsymbol{y}^k + 2\boldsymbol{b}^\top \boldsymbol{y}^k + c}}) + \boldsymbol{1} \right] - \boldsymbol{D}^\top \boldsymbol{1} \\ t_{k+1} & = \frac{1+\sqrt{1+4t_k^2}}{2} \\ \boldsymbol{y}^{k+1} & = \boldsymbol{x}^{k+1} + \left(\frac{t_k - 1}{t_{k+1}} \right) (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \end{cases}$$

Implementing both methods, their objective values along the iterations are displayed in Figure 2. To get a more precise view, their objective values at iterations 1, 101, ..., 1001 are the following:

$$\begin{aligned} F\left(\boldsymbol{x}^{k,\mathrm{PG}}\right) &= [55.543, 46.768, 45.150, 44.398, 44.116, 44.007, 43.940, 43.894, 43.862, 43.840, 43.826], \\ F\left(\boldsymbol{x}^{k,\mathrm{FISTA}}\right) &= [55.543, 43.811, 43.772, 43.771, 43.770, 43.770, 43.770, 43.770, 43.770, 43.770, 43.770]. \end{aligned}$$

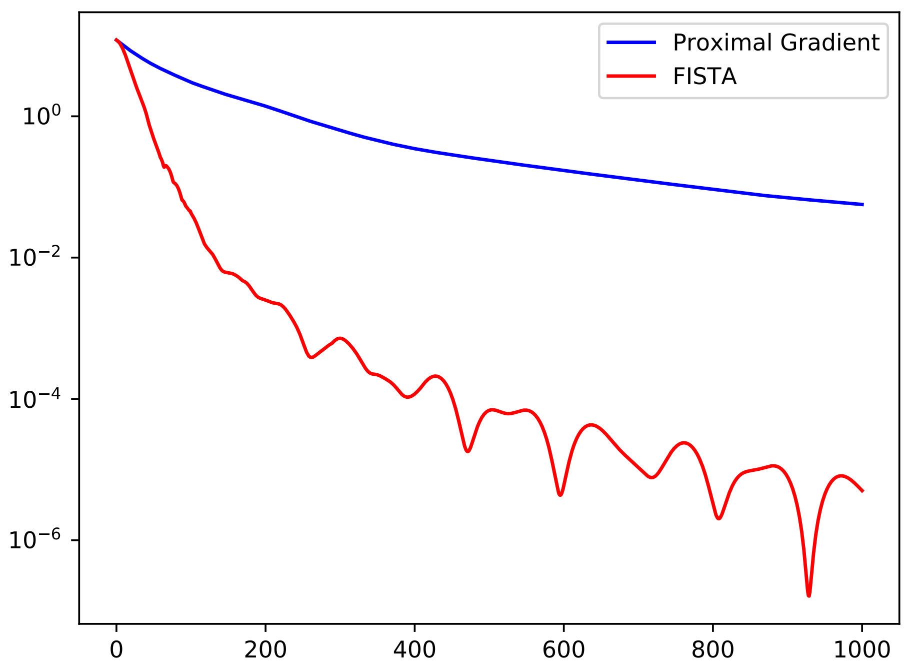

Figure 2: $F(\boldsymbol{x}^k) - F_{\mathrm{opt}}$ in log-scale along the y-axis for the first 1001 iterations of each of the methods with all-zeros vectors and a constant stepsize. The optimal solution has been computed as the best one across 10000 iterations.

Finally, the vectors found by both methods after 1001 iterations are the following:

$$\begin{aligned} \boldsymbol{x}^{*,\mathrm{PG}} = [&0.406, -0.093, -0.093, -0.874, -1.025, -0.044, 0.521, -1.16, 0.877, 0.129, -0.242, 1.664, 1.32, -0.561, -0.079, -1.764, \\ &-1.351, -0.387, -1.158, 0.844, 0.43, -0.715, -0.349, -0.037, 1.408, -0.971, 1.206, 0.795, 0.568, 1.284], \\ \boldsymbol{x}^{*,\mathrm{FISTA}} = [&-0.455, -0.389, 0.291, -1.149, -1.309, -0.43, 0.667, -1.434, 1.082, -0.019, -0.416, 1.732, 1.293, -0.541, -0.143, \\ &-1.704, -1.396, -0.742, -1.539, 0.335, -0.218, -0.662, -0.151, -0.085, 1.801, -0.566, 1.321, 0.849, 0.701, 1.091]. \end{aligned}$$

4 Part 2 - Exercise 3 - slides 71-72

Given a set of data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \in \mathbb{R}^d$ and corresponding labels $y_1, y_2, \cdots, y_n$. The soft margin SVM problem is given by

$$\min \left\{ \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^{n} \max \left\{ 0, 1 - y_i \boldsymbol{w}^\top \boldsymbol{x}_i \right\} \right\}$$

(a) Letting $f(\boldsymbol{w}) = \frac{1}{2}\left\|\boldsymbol{w}\right\|_2^2$, $g(\boldsymbol{z}) = C\sum_{i=1}^n \max\left(0, 1 - z_i\right)$, and $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{x}_1 y_1 \\ \vdots \\ \boldsymbol{x}_n y_n \end{bmatrix}$, the problem matches the canonical form

$$\min f(\boldsymbol{w}) + g(\boldsymbol{A}\boldsymbol{w}),$$

with f proper closed and σ-strongly convex (here $\sigma = 1$), g proper closed and convex. Hence, the Dual Proximal Gradient (DPG) or the Fast DPG (FDPG) methods can be applied. To that aim, the following optimisation problem must be solved.

$$\operatorname*{argmax}_{\boldsymbol{w}} \left\{ \left\langle \boldsymbol{w}, \boldsymbol{A}^\top \boldsymbol{y} \right\rangle - f(\boldsymbol{w}) \right\} = \operatorname*{argmax}_{\boldsymbol{w}} \left\{ \left\langle \boldsymbol{w}, \boldsymbol{A}^\top \boldsymbol{y} \right\rangle - \frac{1}{2}\left\|\boldsymbol{w}\right\|_2^2 \right\} = \boldsymbol{A}^\top \boldsymbol{y}.$$

Also, we compute the proximal operator of ηg as

$$\begin{aligned} \operatorname{prox}_{\kappa \max(0, 1-x)}(x) &= 1 - \operatorname{prox}_{\kappa \max(0, x)}(1 - x), \\ &= 1 - \operatorname{prox}_{\kappa \sigma_{[0;1]}}(1 - x), \\ &= x + \min\left\{\max\left\{1 - x, 0\right\}, \kappa\right\}. \end{aligned}$$

This leads to the compact formulation

$$\operatorname{prox}_{\eta g}(\boldsymbol{w}) = \boldsymbol{w} + \min\left\{\max\left\{1 - \boldsymbol{w}, 0\right\}, \eta C\right\}.$$

Plugging these expressions into the DPG and FDPG iterations, one obtains

- (*DPG*):

$$\begin{cases} \boldsymbol{x}^k = \boldsymbol{A}^\top \boldsymbol{y}^k \\ \boldsymbol{y}^{k+1} = \min\left\{\max\left\{\boldsymbol{y}^k - \frac{1}{L}\boldsymbol{A}\boldsymbol{x}^k + \frac{1}{L}, 0\right\}, C\right\}. \end{cases}$$

- (*FDPG*):

$$\begin{cases} \boldsymbol{u}^k & = \boldsymbol{A}^\top \boldsymbol{w}^k \\ \boldsymbol{y}^{k+1} & = \min\left\{\max\left\{\boldsymbol{w}^k - \frac{1}{L}\boldsymbol{A}\boldsymbol{u}^k + \frac{1}{L}, 0\right\}, C\right\} \\ t_{k+1} & = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\ \boldsymbol{w}^{k+1} & = \boldsymbol{y}^{k+1} + \left(\frac{t_k - 1}{t_{k+1}}\right)\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\right) \end{cases}$$

  and with $\boldsymbol{x}^k = \boldsymbol{A}^\top \boldsymbol{y}^k$.

(b) Both methods have been implemented. As it was not given in the exercise, we have assumed $C = 1$. Also, the class information (originally equal to 1 or 2) has been transformed in to -1 and 1 as this is the setup of usual SVM. Finally, comparing the problem formulation with SVM, we have concluded the obtained hyperplane was assumed to go through the origin. This leads to the results displayed in Figure 3, were one can observe two points are actually classified incorrectly.

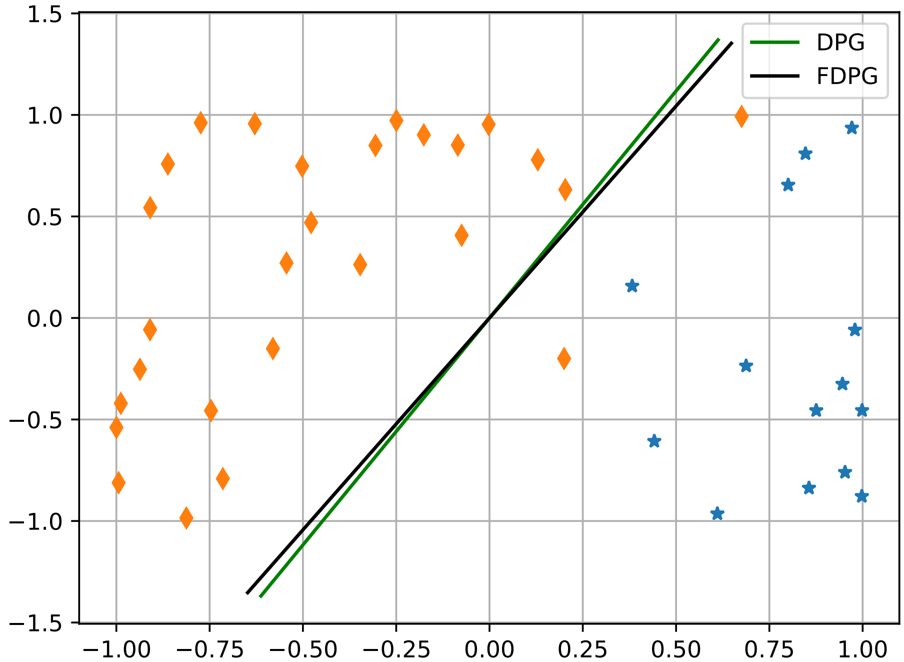

Figure 3: Separating hyperplanes obtained by both methods.

More precisely, the vectors obtained by both methods after 40 iterations are the following.

$$\boldsymbol{x}^{*,\text{DPG}} = [-2.040, 0.913],$$
$$\boldsymbol{x}^{*,\text{FDPG}} = [-2.124, 1.012].$$

5 Part 3 - Exercise 2 - slides 35-36

Consider the problem

$$\min\left\{-\sum_{i=1}^{m}\log\left(\boldsymbol{a}_i^\top\boldsymbol{x}-b_i\right)+\sum_{i=1}^{n-2}\sqrt{\left(x_i-x_{i+1}\right)^2+\left(x_{i+1}-x_{i+2}\right)^2}:\boldsymbol{a}_i^\top\boldsymbol{x}>b_i, i\in[m]\right\}$$

where $\boldsymbol{a}_i\in\mathbb{R}^n$ and $b_i\in\mathbb{R}$ for all $i\in[m]$.

(a) This problem matches the canonical problem

$$\min\left\{h_1(\boldsymbol{x})+h_2(\boldsymbol{v})\text{ s.t. }\boldsymbol{Ax}+\boldsymbol{Bv}=c\right\},$$

with

- $h_1(\boldsymbol{x})=0$;
- $h_2(\boldsymbol{v})=-\sum_{i=1}^m\log\left(v_i^{(1)}-b_i\right)+\sum_{i=1}^{n-2}\sqrt{\left(v_i^{(2)}\right)^2+\left(v_i^{(3)}\right)^2}+\delta_{\{\boldsymbol{x}|\boldsymbol{x}\geq\boldsymbol{b}\}}\left(v^{(4)}\right)$ with $\boldsymbol{v}=\begin{bmatrix}\boldsymbol{v}^{(1)}\\\boldsymbol{v}^{(2)}\\\boldsymbol{v}^{(3)}\\\boldsymbol{v}^{(4)}\end{bmatrix}$. As the case $\boldsymbol{a}_i^\top\boldsymbol{x}=b_i$ is assumed to be rejected by the log operation, a non strict inequality $\boldsymbol{x}\geq\boldsymbol{b}$ is is considered in the indicator function;
- $\boldsymbol{A}=\begin{bmatrix}\tilde{\boldsymbol{A}}\\\boldsymbol{D}\\\boldsymbol{E}\\\tilde{\boldsymbol{A}}\end{bmatrix}$, with $\tilde{\boldsymbol{A}}=\begin{bmatrix}\boldsymbol{a}_1^\top\\\vdots\\\boldsymbol{a}_m^\top\end{bmatrix}$, $\boldsymbol{D}=\begin{bmatrix}1&-1&&&\\0&1&-1&&\\&&\ddots&&\\&&1&-1&0\end{bmatrix}$, $\boldsymbol{E}=\begin{bmatrix}0&1&-1&&&\\&&1&-1&&\\&&\ddots&&&\\&&&&1&-1\end{bmatrix}$; $\boldsymbol{B}=-\boldsymbol{I}$; $\boldsymbol{c}=\boldsymbol{0}$.

In order to obtain the ADLPMM method, the proximal operators of h_1 and h_2 are needed. The first is obtained trivially as

$$\mathrm{prox}_{\eta h_1}(\boldsymbol{x}) = \mathrm{prox}_0(\boldsymbol{x}) = x.$$

For the second one, some preliminary computations are needed:

$$\begin{aligned}
\mathrm{prox}_{-\eta\sum_{i=1}^m \log(v_i-b_i)}(\boldsymbol{v}) &= \mathrm{prox}_{-\eta\sum_{i=1}^m \log(v_i)}(\boldsymbol{v}-\boldsymbol{b}) + \boldsymbol{b},\\
&= \frac{\boldsymbol{v}-\boldsymbol{b}+\sqrt{(\boldsymbol{v}-\boldsymbol{b})^2+4\eta}}{2} + \boldsymbol{b},\\
&= \frac{\boldsymbol{v}+\boldsymbol{b}+\sqrt{(\boldsymbol{v}-\boldsymbol{b})^2+4\eta}}{2},
\end{aligned}$$

where the computations should be understood componentwise.

$$\mathrm{prox}_{\eta\sqrt{x^2+y^2}}(x,y) = \left(1 - \frac{1}{\max\left\{\frac{1}{\eta}\sqrt{x^2+y^2}, 1\right\}}\right)\begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathrm{prox}_{\delta_{\{\boldsymbol{x}|\boldsymbol{x}\geq\boldsymbol{b}\}}}(\boldsymbol{v}) = \max\left\{\boldsymbol{v}, \boldsymbol{b}\right\}.$$

These enables to obtain the proximal operator of ηh_2 as

$$\mathrm{prox}_{\eta h_2}(\boldsymbol{v}) = \begin{bmatrix}
\frac{\boldsymbol{v}^{(1)}+\boldsymbol{b}+\sqrt{(\boldsymbol{v}^{(1)}-\boldsymbol{b})^2+4\eta}}{2} \\
\left(1 - \frac{1}{\max\left\{\frac{1}{\eta}\sqrt{\left(v_1^{(2)}\right)^2+\left(v_1^{(3)}\right)^2}, 1\right\}}\right) v_1^{(2)} \\
\vdots \\
\left(1 - \frac{1}{\max\left\{\frac{1}{\eta}\sqrt{\left(v_n^{(2)}\right)^2+\left(v_n^{(3)}\right)^2}, 1\right\}}\right) v_n^{(2)} \\
\left(1 - \frac{1}{\max\left\{\frac{1}{\eta}\sqrt{\left(v_1^{(2)}\right)^2+\left(v_1^{(3)}\right)^2}, 1\right\}}\right) v_1^{(3)} \\
\vdots \\
\left(1 - \frac{1}{\max\left\{\frac{1}{\eta}\sqrt{\left(v_n^{(2)}\right)^2+\left(v_n^{(3)}\right)^2}, 1\right\}}\right) v_n^{(3)} \\
\max\left\{\boldsymbol{v}^{(4)}, \boldsymbol{b}\right\}
\end{bmatrix}$$

This allows us to write the ADLPMM method with parameter $\rho > 0$ and $\alpha = \rho\lambda_{\max}\left(\boldsymbol{A}^\top\boldsymbol{A}\right)$ as

$$\begin{cases}
\boldsymbol{x}^{k+1} &= \boldsymbol{x}^k - \frac{\rho}{\alpha}\boldsymbol{A}^\top\left(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{z}^k + \frac{1}{\rho}\boldsymbol{y}^k\right) \\
\boldsymbol{z}^{k+1} &= \mathrm{prox}_{\frac{1}{\rho}h_2}\left[\boldsymbol{A}\boldsymbol{x}^{k+1} + \frac{1}{\rho}\boldsymbol{y}^k\right] \\
\boldsymbol{y}^{k+1} &= \boldsymbol{y}^k + \rho(\boldsymbol{A}\boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1})
\end{cases}$$

(b) The considered problem also matches the Chambolle-Pock canonical formulation

$$\min\left\{g(\boldsymbol{x}) + f(\boldsymbol{A}\boldsymbol{x})\right\},$$

with $f = h_2$ and $g = 0$. The corresponding proximal operators are obtained as

$$\begin{aligned}
\mathrm{prox}_{\tau g}(\boldsymbol{x}) &= \boldsymbol{x},\\
\mathrm{prox}_{\sigma f^*}(\boldsymbol{x}) &= \boldsymbol{x} - \sigma\,\mathrm{prox}_{\frac{1}{\sigma}f}\left(\frac{\boldsymbol{x}}{\sigma}\right) = \boldsymbol{x} - \sigma\,\mathrm{prox}_{\frac{1}{\sigma}h_2}\left(\frac{\boldsymbol{x}}{\sigma}\right)
\end{aligned}$$

Thus, the CP method reads as

$$\begin{cases} \boldsymbol{x}^{k+1} & = \boldsymbol{x}^k - \tau \boldsymbol{A}^\top \boldsymbol{y}^k \\ \boldsymbol{y}^{k+1} & = \boldsymbol{y}^k + \sigma \boldsymbol{A}\left(2\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right) - \sigma \operatorname{prox}_{\frac{1}{\sigma}h_2}\left[\frac{\boldsymbol{y}^k + \sigma\boldsymbol{A}\left(2\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right)}{\sigma}\right]. \end{cases}$$

(c) Both methods have been implemented. Notice that we have understood $\|\boldsymbol{A}\|_2^2$ as the norm of the full matrix $\boldsymbol{A}$ and not of $\tilde{\boldsymbol{A}}$. The results are presented in Figure 4 with more iterations than what is written in the statement as otherwise most of the iterations are not feasible. We observe that the ADLPMM v1 ($\rho = 1$) and CP v2 ($\tau = \frac{1}{\|\boldsymbol{A}\|_2^2}$,$\sigma = 1$) are almost equivalent, and that they ADLPMM v2 ($\rho = \frac{1}{\|\boldsymbol{A}\|_2^2}$) is almost identical to CP v1 ($\tau = \sigma = \frac{1}{\|\boldsymbol{A}\|_2}$). More precisely, the first three components of the obtained solution (after 500 iterations) are given by

$$\begin{aligned} \boldsymbol{x}^{*,\mathrm{ADLPMM1}} &= [1.416, 1.430, -0.751, \ldots], \\ \boldsymbol{x}^{*,\mathrm{ADLPMM2}} &= [1.418, 1.404, -0.732, \ldots], \\ \boldsymbol{x}^{*,\mathrm{CP1}} &= [1.418, 1.404, -0.732 \ldots], \\ \boldsymbol{x}^{*,\mathrm{CP2}} &= [1.416, 1.430, -0.751, \ldots]. \end{aligned}$$

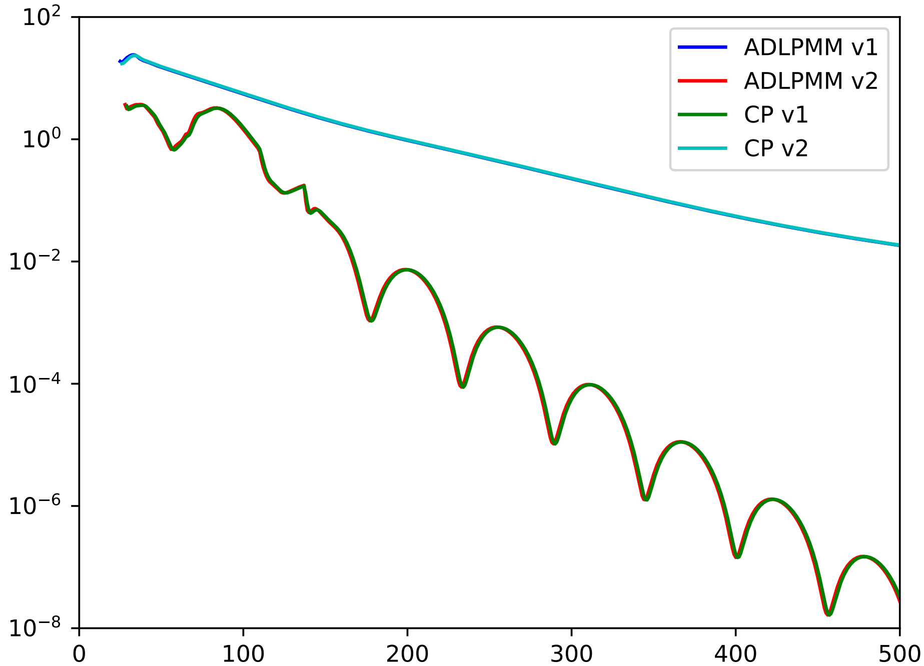

Figure 4: $F(\boldsymbol{x}^k) - F_{\mathrm{opt}}$ in log-scale along the y-axis for the first 500 iterations of each of the methods with all-zeros vectors and a constant stepsize. The optimal solution has been computed as the best one across 10000 iterations.

(d) We consider now the problem

$$\min \frac{\|\boldsymbol{x}\|_2^2}{2} + \left\{ -\sum_{i=1}^{m} \log\left(\boldsymbol{a}_i^\top \boldsymbol{x} - b_i\right) + \sum_{i=1}^{n-2} \sqrt{\left(x_i - x_{i+1}\right)^2 + \left(x_{i+1} - x_{i+2}\right)^2} : \boldsymbol{a}_i^\top \boldsymbol{x} > b_i, i \in [m] \right\}.$$

This fits the CP framework with $f = h_2$ and $g(\boldsymbol{x}) = \frac{\|\boldsymbol{x}\|_2^2}{2}$ which is 1-strongly convex, and has a proximal operator given by

$$\operatorname{prox}_{\frac{\lambda\|\boldsymbol{x}\|_2^2}{2}}(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\lambda + 1}.$$

This leads to the following ACP method:

$$\begin{cases} \boldsymbol{y}^{k+1} &= \boldsymbol{y}^k + \sigma_k \boldsymbol{A}\left(\boldsymbol{x}^k + \theta_k\left(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}\right)\right) - \sigma_k \operatorname{prox}_{\frac{1}{\sigma_k} h_2}\left[\frac{\boldsymbol{y}^k + \sigma_k \boldsymbol{A}\left(\boldsymbol{x}^k + \theta_k\left(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}\right)\right)}{\sigma_k}\right] \\ \boldsymbol{x}^{k+1} &= \frac{\boldsymbol{x}^k - \tau_k \boldsymbol{A}^\top \boldsymbol{y}^{k+1}}{\tau_k + 1} \\ \theta_{k+1} &= \frac{1}{\sqrt{1+\tau_k}}, \quad \tau_{k+1} = \theta_{k+1}\tau_k, \quad \sigma_{k+1} = \frac{\sigma_k}{\theta_{k+1}}. \end{cases}$$

The above problem also fits the FDPG canonical problem

$$\min\left\{f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})\right\},$$

with $f(\boldsymbol{x}) = \frac{\|\boldsymbol{x}\|_2^2}{2}$ which is 1-strongly convex and $g(\boldsymbol{x}) = h_2(\boldsymbol{x})$. We also obtain that

$$\underset{\boldsymbol{u}}{\operatorname{argmax}}\left\{\left\langle \boldsymbol{u}, \boldsymbol{A}^\top \boldsymbol{y}\right\rangle - f(\boldsymbol{u})\right\} = \underset{\boldsymbol{u}}{\operatorname{argmax}}\left\{\left\langle \boldsymbol{u}, \boldsymbol{A}^\top \boldsymbol{y}\right\rangle - \frac{1}{2}\|\boldsymbol{u}\|_2^2\right\} = \boldsymbol{A}^\top \boldsymbol{y}.$$

This leads to the following FDPG method:

$$\begin{cases} \boldsymbol{u}^k &= \boldsymbol{A}^\top \boldsymbol{w}^k \\ \boldsymbol{y}^{k+1} &= \boldsymbol{w}^k - \frac{1}{L}\boldsymbol{A}\boldsymbol{u}^k + \frac{1}{L}\operatorname{prox}_{Lh_2}\left(\boldsymbol{A}\boldsymbol{u}^k - L\boldsymbol{w}^k\right) \\ t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2} \\ \boldsymbol{w}^{k+1} &= \boldsymbol{y}^{k+1} + \left(\frac{t_k - 1}{t_{k+1}}\right)\left(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k\right) \end{cases}$$

and with $\boldsymbol{x}^k = \boldsymbol{A}^\top \boldsymbol{y}^k$.

(e) Both methods have been implemented. The results are presented in Figure Figure 5 with more iterations than what is written in the statement as otherwise most of the iterations are not feasible. We observe that the ACP method works way better than the FDPG one. More precisely, the first three components of the obtained solution (after 100 iterations) are given by

$$\boldsymbol{x}^{*,\mathrm{ACP}} = [1.160, 1.986, -1.285, \ldots],$$
$$\boldsymbol{x}^{*,\mathrm{FDPG}} = [1.140, 1.945, -1.245, \ldots].$$

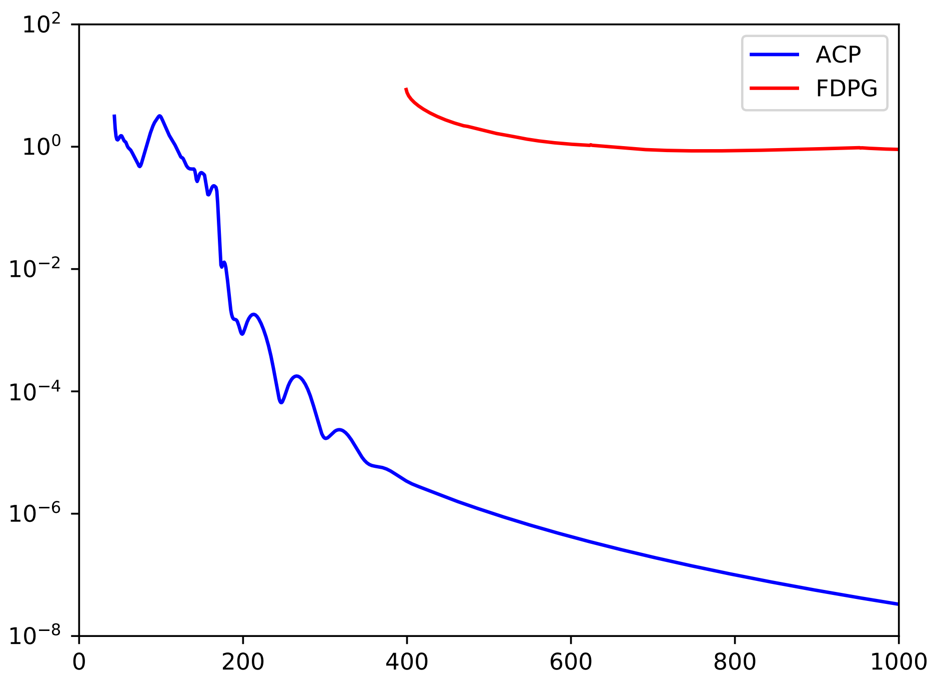

Figure 5: $F(\boldsymbol{x}^k) - F_{\mathrm{opt}}$ in log-scale along the y-axis for the first 1000 iterations of each of the methods with all-zeros vectors and a constant stepsize. The optimal solution has been computed as the best one across 10000 iterations.

In the above figure, FDPG does not seem to converge. Yet, looking at more iterations, we obtain Figure 6 that depicts FDPG finally converges.

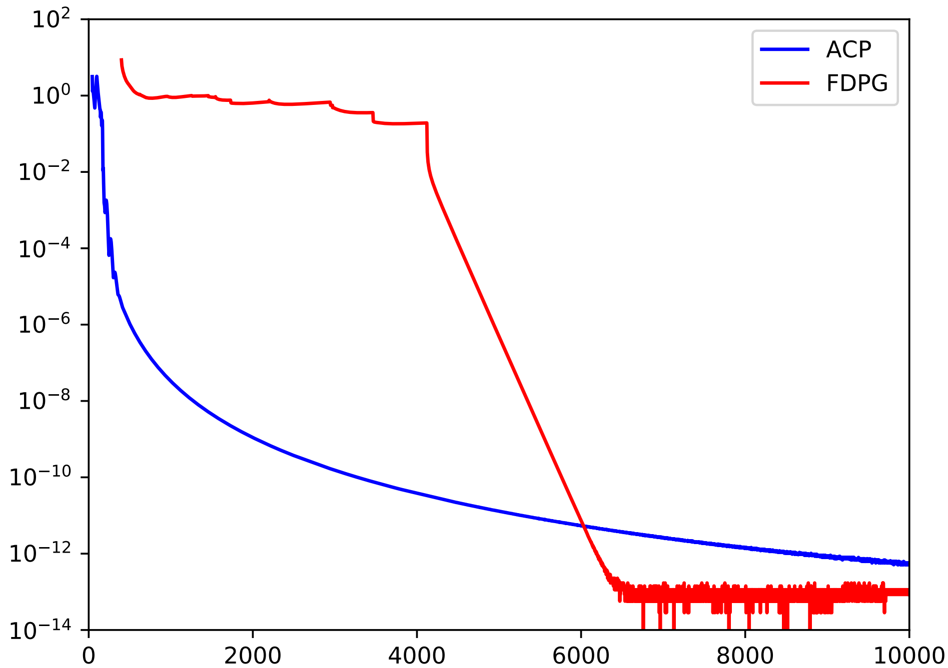

Figure 6: $F(\boldsymbol{x}^k) - F_{\text{opt}}$ in log-scale along the y-axis for the first 10000 iterations of each of the methods with all-zeros vectors and a constant stepsize. The optimal solution has been computed as the best one across 10000 iterations.