ECE 490: Introduction to Optimization **Fall 2018**

Solutions for Homework 2

1.

(a)

Substituting $v_k = (1+\beta)x_k - \beta x_{k-1}$ and $x_{k+1} = (1+\beta)x_k - \beta x_{k-1} - \alpha\nabla f(v_k)$, we have

$$\begin{aligned}
&\nabla f(v_k)^\mathsf{T}(x_k - v_k) + \frac{m}{2}\|x_k - v_k\|^2 + \nabla f(v_k)^\mathsf{T}(v_k - x_{k+1}) - \frac{L}{2}\|v_k - x_{k+1}\|^2 \\
=&\beta\nabla f(v_k)^\mathsf{T}(x_{k-1} - x_k) + \frac{m\beta^2}{2}\|x_{k-1} - x_k\|^2 + \alpha\|\nabla f(v_k)\|^2 - \frac{L\alpha^2}{2}\|\nabla f(v_k)\|^2 \\
=&\begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \\ \nabla f(v_k) \end{bmatrix}^\mathsf{T} \left(\frac{1}{2}\begin{bmatrix} \beta^2 m & -\beta^2 m & -\beta \\ -\beta^2 m & \beta^2 m & \beta \\ -\beta & \beta & \alpha(2-L\alpha) \end{bmatrix} \otimes I_p \right) \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \\ \nabla f(v_k) \end{bmatrix}
\end{aligned}$$

Therefore, we have

$$X_1 = \frac{1}{2}\begin{bmatrix} \beta^2 m & -\beta^2 m & -\beta \\ -\beta^2 m & \beta^2 m & \beta \\ -\beta & \beta & \alpha(2-L\alpha) \end{bmatrix} \otimes I_p.$$

(b)

Substituting $v_k = (1+\beta)x_k - \beta x_{k-1}$ and $x_{k+1} = (1+\beta)x_k - \beta x_{k-1} - \alpha\nabla f(v_k)$, we have

$$\begin{aligned}
&\nabla f(v_k)^\mathsf{T}(x^* - v_k) + \frac{m}{2}\|x^* - v_k\|^2 + \nabla f(v_k)^\mathsf{T}(v_k - x_{k+1}) - \frac{L}{2}\|v_k - x_{k+1}\|^2 \\
=&-\nabla f(v_k)^\mathsf{T}((1+\beta)(x_k - x^*) - \beta(x_{k-1} - x^*)) + \frac{m}{2}\|(1+\beta)(x_k - x^*) - \beta(x_{k-1} - x^*)\|^2 \\
&+ \alpha\|\nabla f(v_k)\|^2 - \frac{L\alpha^2}{2}\|\nabla f(v_k)\|^2 \\
=&\begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \\ \nabla f(v_k) \end{bmatrix}^\mathsf{T} \left(\frac{1}{2}\begin{bmatrix} (1+\beta)^2 m & -\beta(1+\beta)m & -(1+\beta) \\ -\beta(1+\beta)m & \beta^2 m & \beta \\ -(1+\beta) & \beta & \alpha(2-L\alpha) \end{bmatrix} \otimes I_p \right) \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \\ \nabla f(v_k) \end{bmatrix}
\end{aligned}$$

Therefore, we have

$$X_2 = \frac{1}{2}\begin{bmatrix} (1+\beta)^2 m & -\beta(1+\beta)m & -(1+\beta) \\ -\beta(1+\beta)m & \beta^2 m & \beta \\ -(1+\beta) & \beta & \alpha(2-L\alpha) \end{bmatrix} \otimes I_p$$

(c)

Now it is straightforward to verify that the following holds

$$\begin{bmatrix} A^\mathsf{T}PA-\rho^2 P & A^\mathsf{T}PB \\ B^\mathsf{T}PA & B^\mathsf{T}PB \end{bmatrix} - X = \frac{\sqrt{m}(\sqrt{L}-\sqrt{m})^3}{2(L+\sqrt{Lm})} \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes I_p \leq 0$$

This above fact can be verified using Matlab symbolic toolbox.

(d)

To guarantee $f(x_T) - f(x^*) \leq \varepsilon$, we can use the bound $f(x_T) - f(x^*) \leq C\left(1-\sqrt{\frac{m}{L}}\right)^k$. If we choose T such that $C\left(1-\sqrt{\frac{m}{L}}\right)^T \leq \varepsilon$, then we guarantee $f(x_T) - f(x^*) \leq \varepsilon$. Notice $C\left(1-\sqrt{\frac{m}{L}}\right)^k \leq \varepsilon$ is equivalent to

$$\log C + k \log\left(1-\sqrt{\frac{m}{L}}\right) \leq \log(\varepsilon)$$

The above inequality is equivalent to

$$k \geq -\log\left(\frac{C}{\varepsilon}\right) / \log\left(1-\sqrt{\frac{m}{L}}\right) \tag{1}$$

Notice we have $\sqrt{\frac{L}{m}} \geq -1/\log\left(1-\sqrt{\frac{m}{L}}\right)$. Therefore, we can choose $T = O\left(\sqrt{\frac{L}{m}}\log(\frac{1}{\varepsilon})\right)$ to guarantee $f(x_T) - f(x^*) \leq \varepsilon$.

2.

(a)

A Matlab code is provided on the course website. From Figure 1, we can see Heavy-ball method performs best for the positive definite quadratic minimization problem. Nesterov's accelerated method performs also well, and is just worse than Heavy-ball method by a constant factor. When the condition number is large, the gradient method is very slow. But Nesterov's method and Heavy-ball method still work well.

Finally, another thing worth mentioning is that the iteration complexity is independent of the problem dimension p. We can also see this in the plots. When p is changed and the condition number is fixed, the required iteration number does not change.

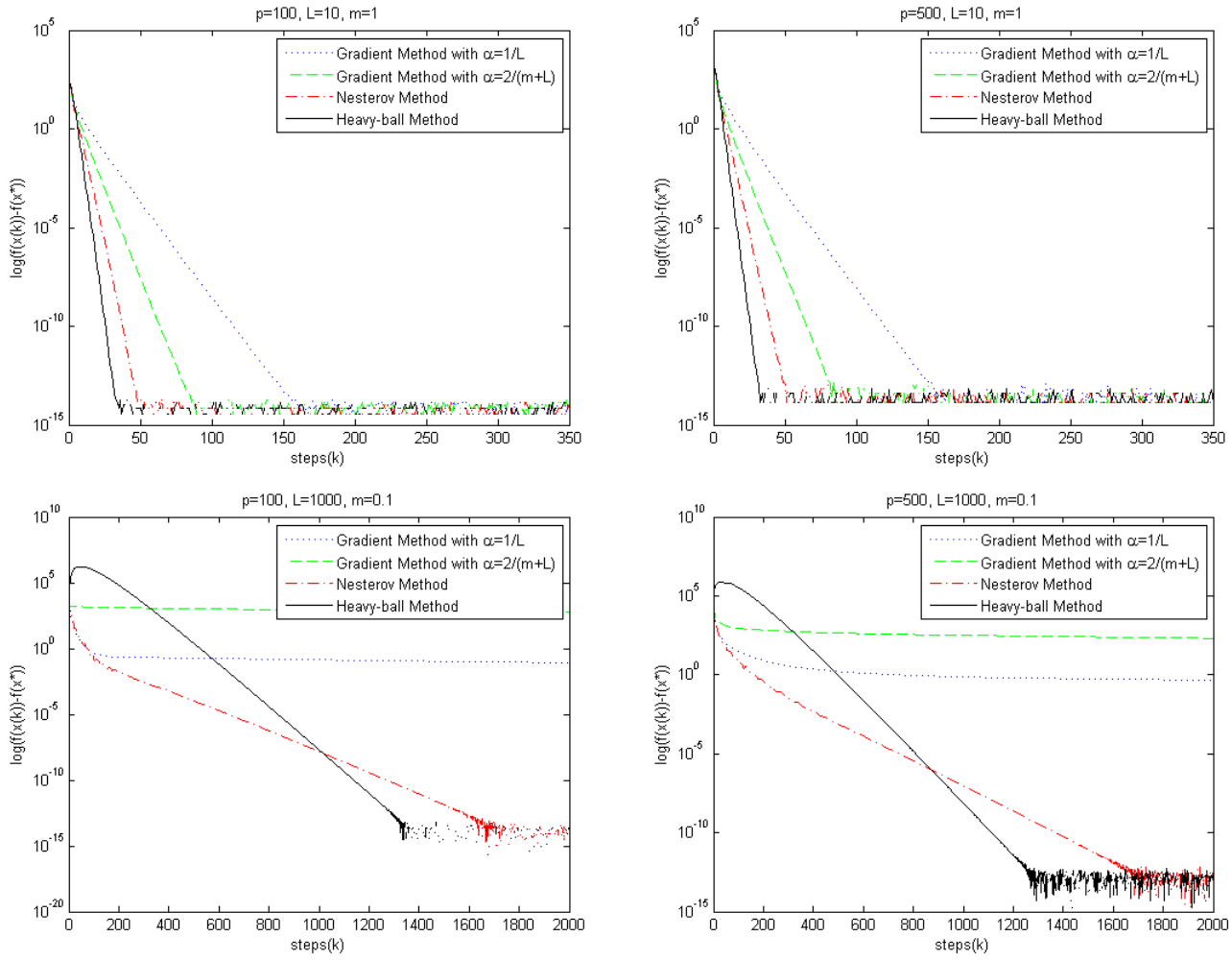

Figure 1. In the simulations, we vary p, m, and L. The condition number for the plots in the first row is small, i.e. $L/m = 10$. The condition number of the plots in the second row is large, i.e. $L/m = 10000$. Then the gradient method becomes extremely slow.

(b) A Matlab code for this problem is also posted on the course website. From the simulation, we can see that Heavy-ball method with the given parameters does not converge. Although Heavy-ball method with these parameter choices works well for the positive definite quadratic minimization problem, it is not guaranteed to work for optimization of all smooth strongly-convex functions. Both the gradient method and Nesterov's method still work well for this example, as guaranteed by the iteration complexity theory.

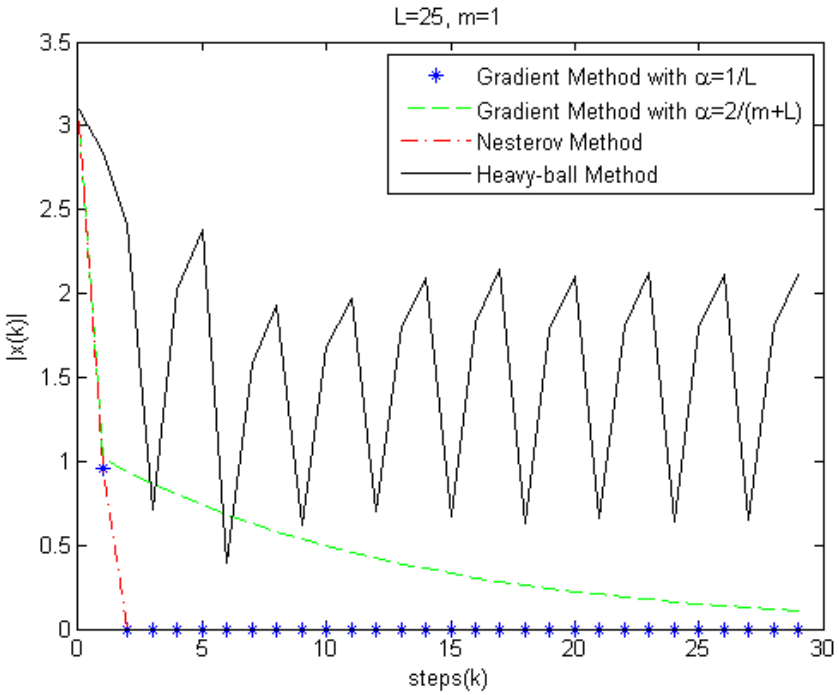

Figure 2. Heavy-ball method does not converge in this case.