Problem 1 [BV 9.3]

(a) $p^* = 1$ which is attained at $x^* \notin \mathbf{dom} f$, where $x^* = (1, 0)$.

(b) The sublevel set S is not a closed set, and f cannot be strongly convex on S.

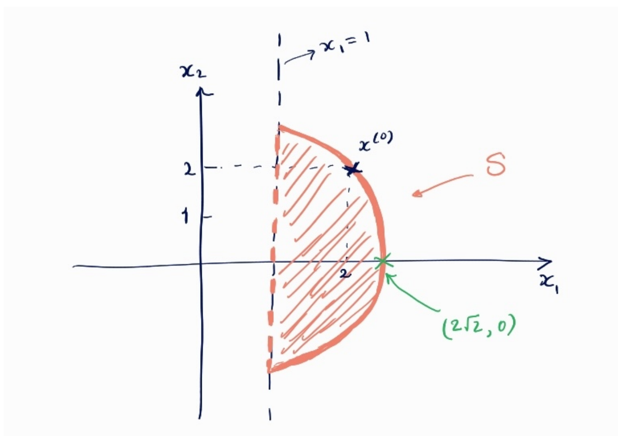


Figure 1: Sublevel set $S = \{x \in \mathbf{dom} f | f(x) \leq f(2, 2)\}$.

(c) No convergence is guaranteed in this case, since for a function f with open domain, we assume f is infinite on the boundary and outside the open domain, therefore the condition $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^T \Delta x$ is always true as the algorithms reaches the boundary and the step size t infinitely many times gets updated by βt. And since $\beta \in (0, 1)$, the step size will reach to zero and the algorithm stops.

Problem 2 [BV 9.5]

For this strong convex f we have:

$$f(x + t\Delta x) \leq f(x) + \nabla f(x)^T (t\Delta x) + \frac{M}{2} \|t\Delta x\|_2^2$$

By combining this inequality with the stopping condition of backtracking, i.e., $f(x+t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$, we must have this condition to stop:

$$
\begin{aligned}
f(x) + \nabla f(x)^T (t\Delta x) + \frac{M}{2}\|t\Delta x\|_2^2 &\leq f(x) + \alpha t \nabla f(x)^T \Delta x \\
t[(1-\alpha) f(x)^T \Delta x + t\frac{M}{2}\|\Delta x\|_2^2] &\leq 0 \\
t \leq \frac{-(1-\alpha) f(x)^T \Delta x}{\frac{M}{2}\|\Delta x\|_2^2} &\leq -\frac{f(x)^T \Delta x}{M\|\Delta x\|_2^2}
\end{aligned}
$$

where the last inequality holds since $\alpha \in (0, 0.5)$. To find an upperbound on the number of backtracking iterations, starting from $t = 1$ and reaching β^n after n iterations. We look for the n which makes the backtracking to stop, in other words, which makes the step size greater than the value we derived in previous part:

$$
\beta^n \geq -\frac{f(x)^T \Delta x}{M\|\Delta x\|_2^2} \rightarrow n \leq \frac{\log(-\frac{f(x)^T \Delta x}{M\|\Delta x\|_2^2})}{\log \beta}
$$

where the last inequality holds since $\beta \in (0, 1)$.

Problem 3 [BV 9.6]

First deriving the analytical solution to exact line search gradient descent:

$$
\begin{aligned}
t &= \arg\min_{s>0} f(x - s\nabla f(x)) \\
&= \arg\min_{s>0} f(x - s \begin{bmatrix} x_1 \\ \gamma x_2 \end{bmatrix}) \\
&= \arg\min_{s>0} f(\begin{bmatrix} x_1(1-s) \\ x_2(1-\gamma s) \end{bmatrix}) \\
&= \arg\min_{s>0} \frac{1}{2}(x_1^2(1-s)^2 + \gamma x_2^2(1-\gamma s)^2)
\end{aligned}
$$

Taking the derivative of this scalar function and setting it to zero, results in:

$$
t(x_1, x_2) = \frac{x_1^2 + \gamma^2 x_2^2}{x_1^2 + \gamma^3 x_2^2} = \frac{(\frac{x_1}{x_2})^2 + \gamma^2}{(\frac{x_1}{x_2})^2 + \gamma^3},
$$

where t is only a function of $\frac{x_1}{x_2}$.

With this step size, if we let $x^{(0)} = (\gamma, 1)^T$ using the update rule of gradient descent we have:

$$x_1^{(k+1)} = x_1^{(k)} - t x_1^{(k)} = (1-t)x_1^{(k)} = \frac{\gamma^2(\gamma-1)}{\gamma^3 + (x_1^{(k)}/x_2^{(k)})^2} x_1^{(k)},$$
$$x_2^{(k+1)} = x_2^{(k)} - t\gamma x_2^{(k)} = (1-t\gamma)x_2^{(k)} = \frac{-(x_1^{(k)}/x_2^{(k)})^2(\gamma-1)}{\gamma^3 + (x_1^{(k)}/x_2^{(k)})^2} x_2^{(k)}$$

Therefore evaluating these sequences for a couple of k:

$$x_1^{(1)} = \frac{\gamma^2(\gamma-1)}{\gamma^3 + (\gamma)^2}\gamma = \frac{\gamma-1}{\gamma+1}\gamma,$$
$$x_2^{(1)} = \frac{-(\gamma)^2(\gamma-1)}{\gamma^3 + (\gamma)^2}1 = \frac{-(\gamma-1)}{\gamma+1}1,$$

and

$$x_1^{(2)} = \frac{\gamma^2(\gamma-1)}{\gamma^3 + (-\gamma)^2}x_1^{(1)} = \frac{\gamma-1}{\gamma+1}x_1^{(1)} = (\frac{\gamma-1}{\gamma+1})^2\gamma,$$
$$x_2^{(2)} = \frac{-(-\gamma)^2(\gamma-1)}{\gamma^3 + (-\gamma)^2}x_2^{(1)} = \frac{-(\gamma-1)}{\gamma+1}x_2^{(1)} = (\frac{-(\gamma-1)}{\gamma+1})^2,$$

This geometric sequence continues with scaling as k grows, and all the updates are just scaled version of previous values, by induction we have:

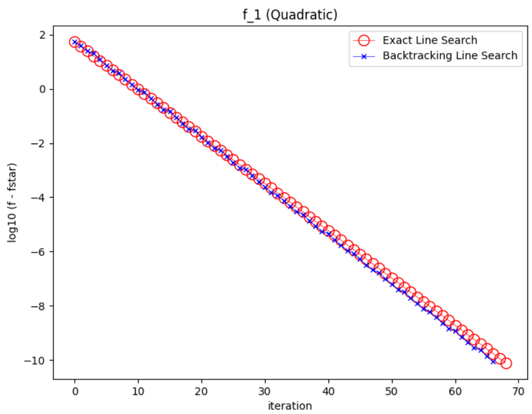
$$\begin{aligned} x_1^{(k)} &= (1-t)^k x_1^{(0)} \\ &= \left(\frac{\gamma-1}{\gamma+1}\right)^k \gamma \end{aligned}$$

Exactly same reasoning for the second coordinate will result in update terms of:

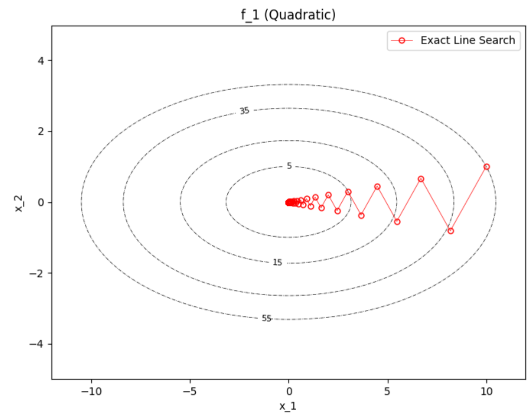
$$\begin{aligned} x_2^{(k)} &= (1-t\gamma)^k x_2^{(0)} \\ &= \left(-\frac{\gamma-1}{\gamma+1}\right)^k 1 \end{aligned}$$

Problem 4 [Numerical Problem]

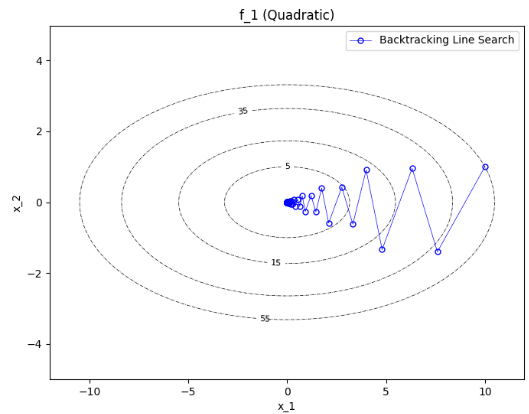
The figures below are showing the results of the implemented algorithms, further details and Python implementation can be found in the **attached Jupyter notebook**.



(a) Convergence rate of the gradient descent.

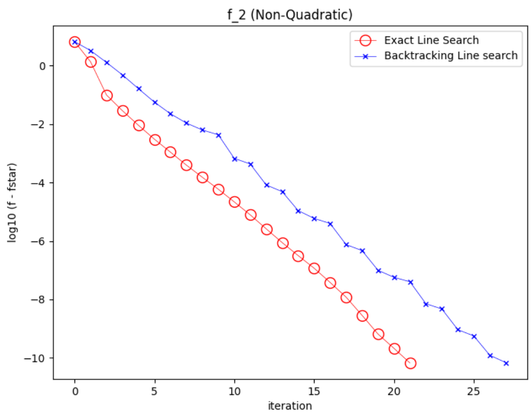


(b) Exact line search steps on contour plot.

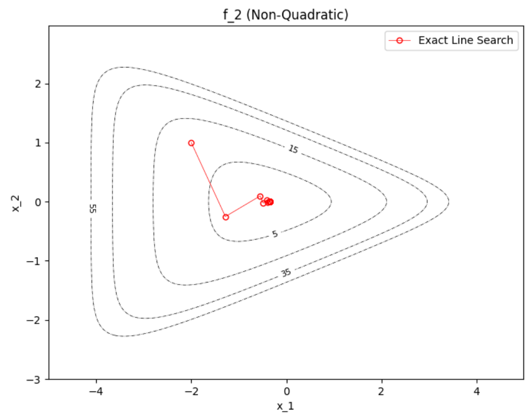


(c) Backtracking line search steps on contour plot.

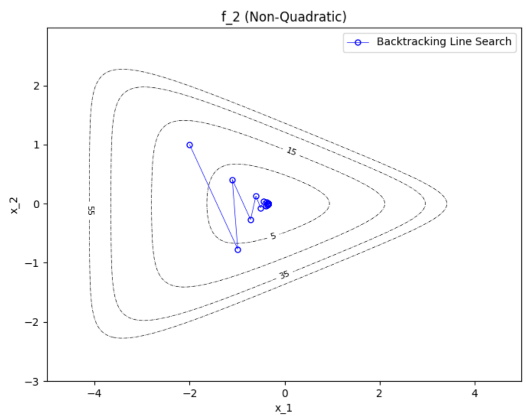
Figure 2: Gradient descent on $f_1(x_1, x_2) = \frac{1}{2}(x_1^2 + \gamma x_2^2)$ with $\gamma = 10$.



(a) Convergence rate of the gradient descent.



(b) Exact line search steps on contour plot.



(c) Backtracking line search steps on contour plot.

Figure 3: Gradient descent on $f_2(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$.