Caitlyn Jones

Prof. Thomas Fai

MATH 122A: Numerical Methods and Big Data

5 November 2024

Homework 6

1. Consider the function $f(x, y) = \frac{x^4}{16} - 2xy + y^4$

   a. Gradient of a function is $\Delta f(x, y) = \langle \frac{df}{dx}, \frac{df}{dy} \rangle$

   $\frac{df}{dx} = \frac{x^3}{4} - 2y$ and $\frac{df}{dy} = -2x + 4y^3$

   So, $\Delta f(x, y) = \langle \frac{x^3}{4} - 2y, -2x + 4y^3 \rangle$

   To find the critical points, we must first solve the following system of equations:

   $\frac{x^3}{4} - 2y = 0$

   $-2x + 4y^3 = 0$

   First, solving for y in terms of x:

   $\frac{x^3}{4} = 2y$

   $\frac{x^3}{8} = y$

   Plugging this back into to solve for the numerical values of x:

   $-2x + 4\left(\frac{x^3}{8}\right)^3 = 0$

   $-2x + \frac{4x^9}{512} = 0$

   $-2x + \frac{x^9}{128} = 0$

   $\frac{x^9}{128} - 2x = 0$

$\mathrm{x}\left(\frac{\mathrm{x}^8}{128}-2\right)=0$

$\mathrm{x}=0$

$\frac{\mathrm{x}^8}{128}=2$

$\mathrm{x}^8=256$

$\mathrm{x}=\pm 2$

The x-values obtained are $\mathrm{x}=\pm 2, \mathrm{x}=0$

Now, we plug these values back in to solve for y, which will give the critical points.

When $\mathrm{x}=0$,

$\mathrm{y}=\frac{(0)^3}{8}$

$\mathrm{y}=0$

When $\mathrm{x}=-2$,

$\mathrm{y}=\frac{(-2)^3}{8}$

$\mathrm{y}=\frac{-8}{8}$

$\mathrm{y}=-1$

When $\mathrm{x}=2$,

$\mathrm{y}=\frac{(2)^3}{8}$

$\mathrm{y}=\frac{8}{8}$

$\mathrm{y}=1$

Therefore, the critical points of this function are

$(0,0), (-2,-1)$, and $(2,1)$.

b. The Hessian of a function is the following matrix:

$$\mathrm{Hf}(x,y) = \begin{bmatrix} f_{xx} & f_{yx} \\ f_{xy} & f_{yy} \end{bmatrix}$$

To compute the Hessian, we first need to compute each second derivative. Given that $\frac{df}{dx} = \frac{x^3}{4} - 2y$ and $\frac{df}{dy} = -2x + 4y^3$, we can compute that

$f_{xx} = \frac{3x^2}{4}$, $f_{xy} = -2$, $f_{yx} = -2$, and $f_{yy} = 12y^2$

Therefore,

$$\mathrm{Hf}(x,y) = \begin{bmatrix} \frac{3x^2}{4} & -2 \\ -2 & 12y^2 \end{bmatrix}$$

In order to classify critical points, we must (1) find the determinant D of $\mathrm{Hf}(x,y)$ (2) evaluate D at critical point (x,y), and (3) evaluate f_{xx} at critical point (x,y). From here, we will be able to classify each critical point based on the following:

If D < 0, (x,y) is a saddle point

If D > 0 and $f_{xx}(x,y) > 0$, (x,y) is a local minimum

If D > 0 and $f_{xx}(x,y) < 0$, (x,y) is a local maximum

With the above rules, we can now classify each critical point.

For determinant D, we will use the formula

$D = f_{xx} * f_{yy} - {f_{xy}}^2$ or $D = \frac{3x^2}{4} * 12y^2 - (-2)^2$

$D(0,0) = \frac{3(0)^2}{4} * 12(0)^2 - (-2)^2$

$D(0,0) = 0 - 4$

$D(0,0) = -4$

Since D < 0, $(0,0)$ is a saddle point. ■

$\mathrm{D}(-2,-1) = \frac{3(-2)^2}{4} * 12(-1)^2 - (-2)^2$

$\mathrm{D}(-2,-1) = 3 * 12 - 4$

$\mathrm{D}(-2,-1) = 32$

$f_{xx}(-2,-1) = \frac{3(-2)^2}{4}$

$f_{xx}(-2,-1) = 36$

Since D > 0 and $f_{xx}(-2,-1) > 0, (-2,-1)$ is a local minimum. ■

$\mathrm{D}(2,1) = \frac{3(2)^2}{4} * 12(1)^2 - (-2)^2$

$\mathrm{D}(2,1) = 3 * 12 - 4$

$\mathrm{D}(2,1) = 32$

$f_{xx}(2,1) = \frac{3(2)^2}{4}$

$f_{xx}(2,1) = 36$

Since D > 0 and $f_{xx}(2,1) > 0, (2,1)$ is a local minimum. ■

In summary, $(0,0)$ is a saddle point and $(-2,-1), (2,1)$ are the local minima.

c. Since the function has two local minima, we can use these to determine if the function $f(x,y) = \frac{x^4}{16} - 2xy + y^4$ has a global minimum.

Evaluating $f(-2,-1)$,

$$f(-2,-1) = \frac{(-2)^4}{16} - 2(-2)(-1) + (-1)^4$$

$f(-2,-1) = 1 - 2(2) + 1$

$f(-2,-1) = -2$

Evaluating $f(2,1)$,

$$f(2,1) = \frac{(2)^4}{16} - 2(2)(1) + (1)^4$$

$f(2,1) = 1 - 2(2) + 1$

$f(2,1) = -2$

After plugging in the local minima and evaluating the function, we see that the global minima are $f(-2,-1) = -2$ and $f(2,1) = -2$.

2. Before I fixed the broken code for each individual method, I first inputted the values for the arrays `Df(x,y) and DDf(x,y).`

```
##############################
#Your code for the gradient and Hessian goes here
def Df(x,y):
    return np.array([x**3/4 - 2*y, -2*x + 4*(y**3)])
def DDf(x,y):
    return np.array([[3*(x**2)/4, -2.],
                     [-2., 12*(y**2)]])
##############################
```

First, I calculated the gradient descent step.

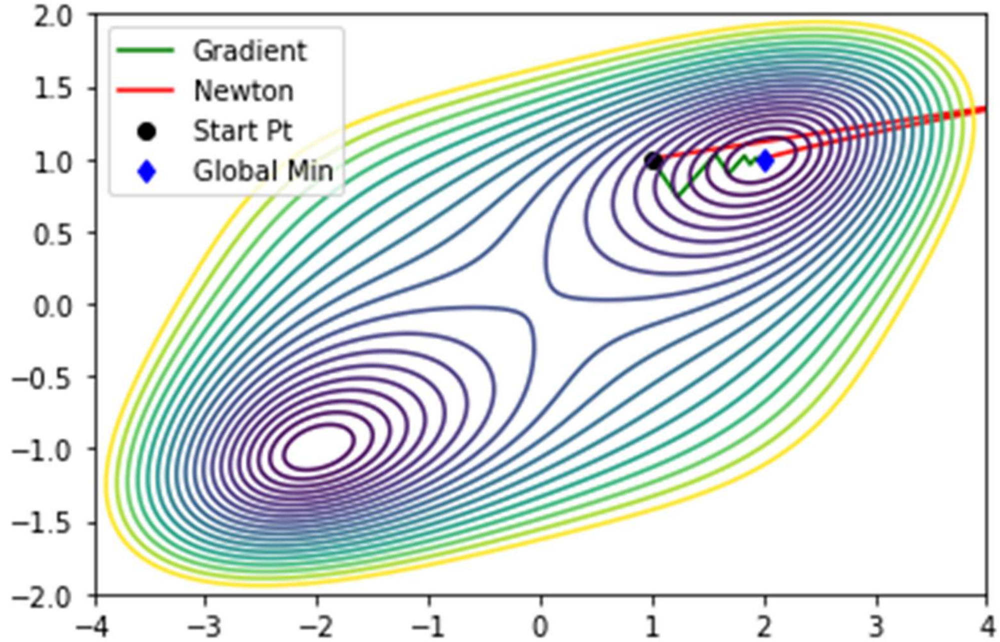
```
##############################
# Your code for the gradient descent step goes here
step = -tmin * DF(x)
##############################
```

Then, I calculated the step for Newton's method.

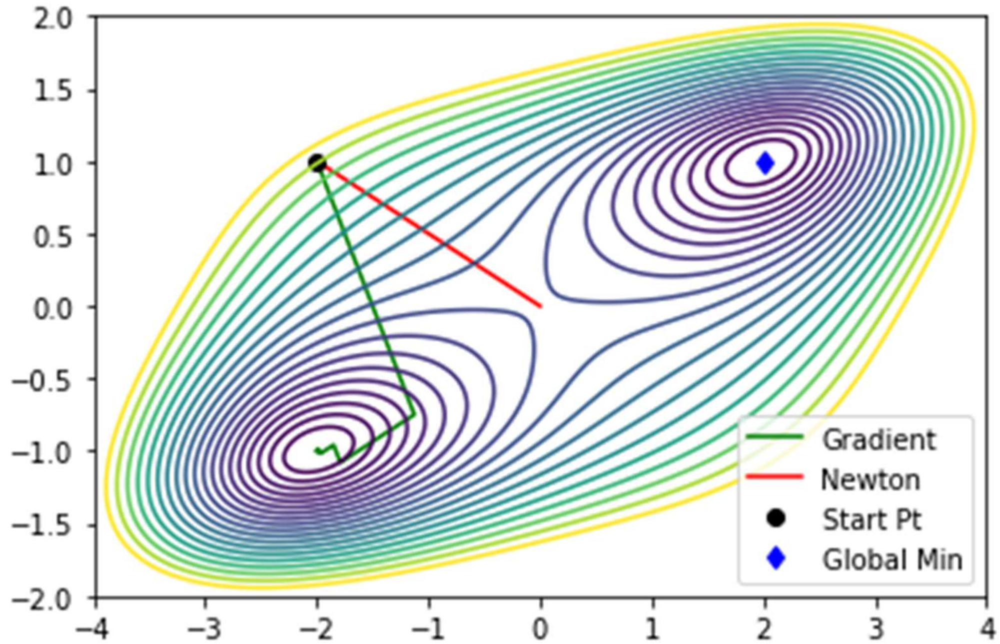
```
##############################
# Your code for the Newton's method step goes here
step = -np.linalg.inv(DDF(x)) @ DF(x)
##############################
```

Here are the following graphs for (1,1), (-2,1), and (-1,1).

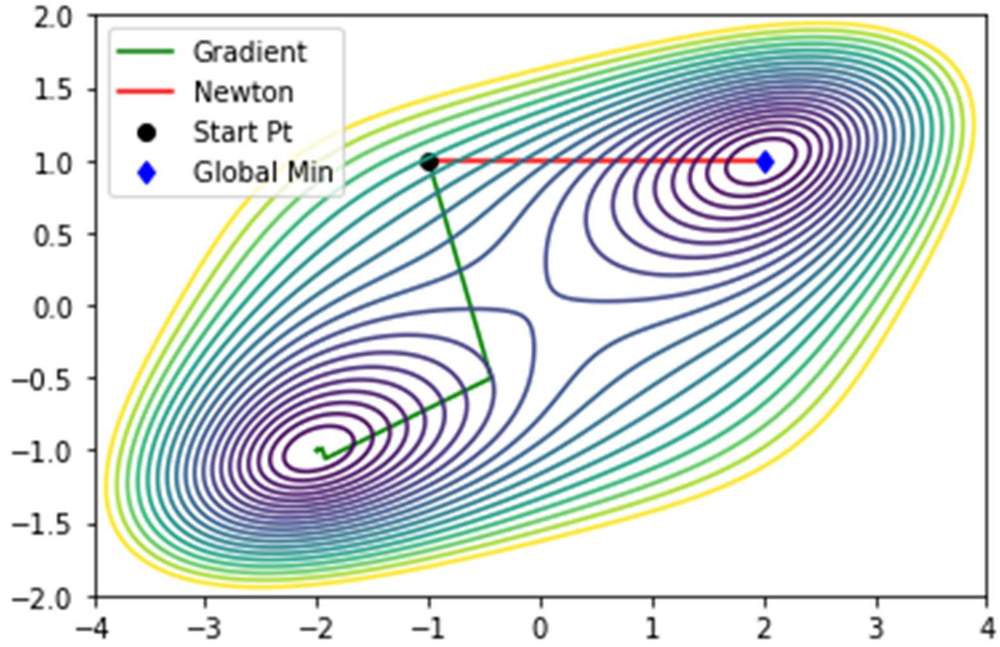
(1,1)


Gradient
Newton
Start Pt
Global Min
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−4
−3
−2
−1
0
1
2
3
4

(-2,1)


Gradient
Newton
Start Pt
Global Min
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−4
−3
−2
−1
0
1
2
3
4

(-1,1)



In using the Steepest/Gradient Descent method, we can see that the function converges to the global minimum for starting point (1,1). However, at starting points (-2,1) and (-1,1), the function does not converge to the global minimum, but instead converges to its negative counterpart ((-2, -1) instead of (2,1)). In terms of speed, this method converges slower than Newton's method due to the sharp peaks, plateaus, and other trends in the graph. As we can see in the figures above, Newton's method (red) is much smoother than Gradient Descent method (green). On the other hand, Newton's method converges to the global minimum whenever the starting point is close to such. In figure 2, when the starting point is (-2,1), the graph converges to (0,0) instead of the global minimum (2,1). Though this happens, Newton's method is more efficient and accurate than Steepest/Gradient Descent method, as the graph for Newton's method converges faster and closer to the global minimum for the various starting points.