MAT3007 HM8 120090694

▼ Q1

First derive the derivative form:

$f(x)' - \frac{2log(x)}{(x-1)^3} + \frac{1}{x(x-1)^2} + \frac{4x}{((x-1)^2(x+1)^2}$

By bisection method, we can find the in 9 iteration (python code in q1.py)

```
x_mid =  3.0
x_mid =  2.25
x_mid =  1.875
x_mid =  2.0625
x_mid =  2.15625
x_mid =  2.203125
x_mid =  2.1796875
x_mid =  2.19140625
x_mid =  2.185546875
count =  9
2.1884765625
```

x = 2.1884765625 as a minimizer

$-\frac{1}{(x-1)^2}\left(\ln(x) - \frac{2(x-1)}{x+1}\right) = -0.02670718994$

▼ Q2

▼ (i)

To get the minimizer at a stopping tolerance tol= 10^{-5}

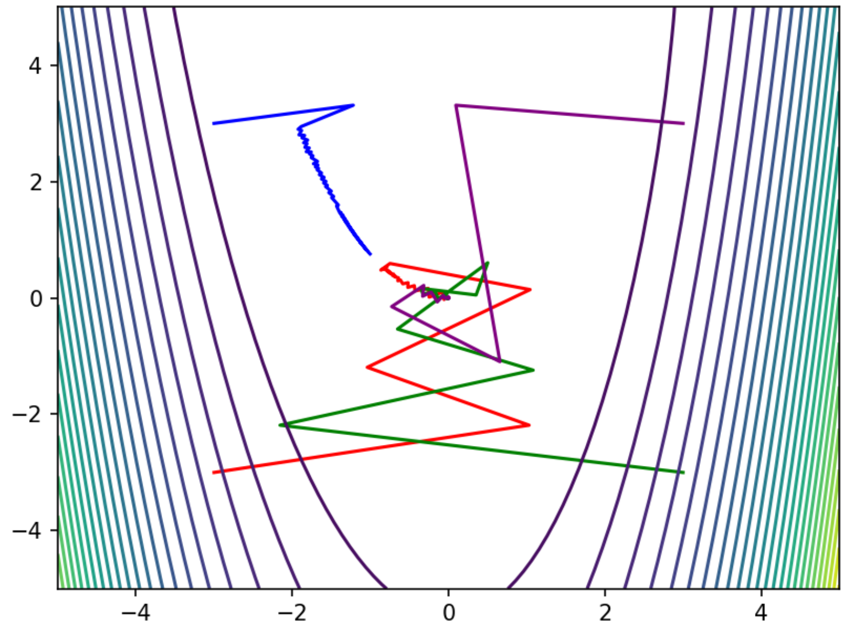
By backtracking line search, it takes 19 iterations and finds the minimizer $f(0,0) = 0$

By exact line search, it takes 3881 iterations, and finds the minimizer $f(-1.0037, 0.7555) = 0.0833$

▼ (ii)

Run q2_2.py

Trajectory by backtracking line search



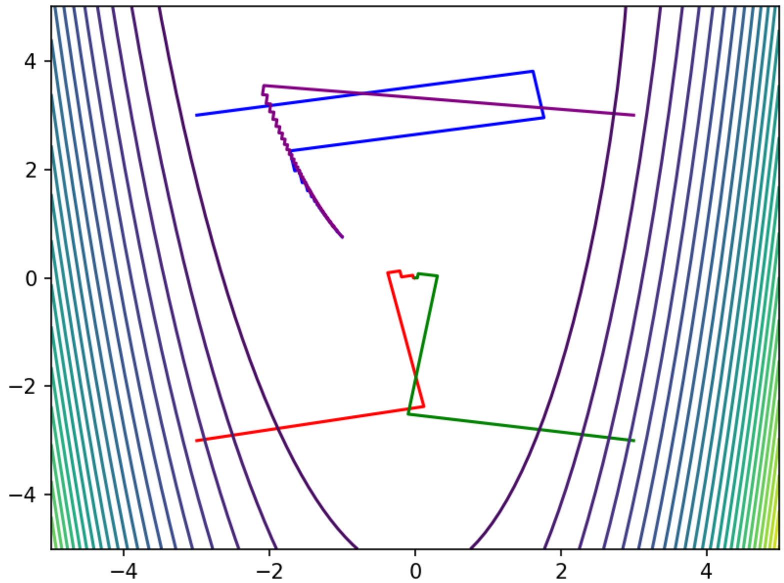

```
x =  -0.0
y =  -0.0
f(x,y) =  0.0
count =  47
x =  -0.0
y =  -0.0
f(x,y) =  0.0
count =  21
x =  -1.0042
y =  0.7563
f(x,y) =  0.0833
count =  1883
x =  -0.0
y =  -0.0
f(x,y) =  0.0
count =  19
```

Perfomance and behavior

around 20 iteration which is short, but iteration with high variance can goes to 1883,

the trajectory is not smooth and oscillates

Trajectory by exact line search




```
x =  -0.0
y =  -0.0
f(x,y) =  0.0
count =  12
x =  0.0
y =  0.0
f(x,y) =  0.0
count =  9
x =  -1.0036
y =  0.7554
f(x,y) =  0.0833
count =  2837
x =  -1.0037
y =  0.7555
f(x,y) =  0.0833
count =  3881
```

Perfomance and behavior

can be thousands iteration which is rather long,

the trajectory is not smooth and each time goes a vertical curve

▼ Q3

▼ (i)

to show d^k is a descent direction

is to show $f(x^k+\alpha d^k) - f(x^k) \le 0$, for $\alpha > 0$

By taylor expantion:

$$f(x^k+\alpha d^k) \simeq f(x^k) + \alpha \nabla f(x^k)^T d^k$$

$$f(x^k+\alpha d^k) - f(x)^k \simeq \alpha \nabla f(x^k)^T d^k$$

$$= -\alpha \nabla f(x^k)^T D_k^{-1} \nabla f(x^k)$$

$$= -\alpha \nabla f(x^k)^T \mathrm{diag}\left(\frac{1}{V_1^k}, \frac{1}{V_2^k}, \cdots, \frac{1}{V_n^k}\right)$$

Because $V_i^k = \sqrt{\varepsilon + \sum_{j=tm(k)}^{k} (\nabla f(x^j))_i^2} \geq 0$

$$\therefore \left[\nabla f(x^k)^T\right]_i^2 \cdot \frac{1}{V_i^k} \geq 0$$

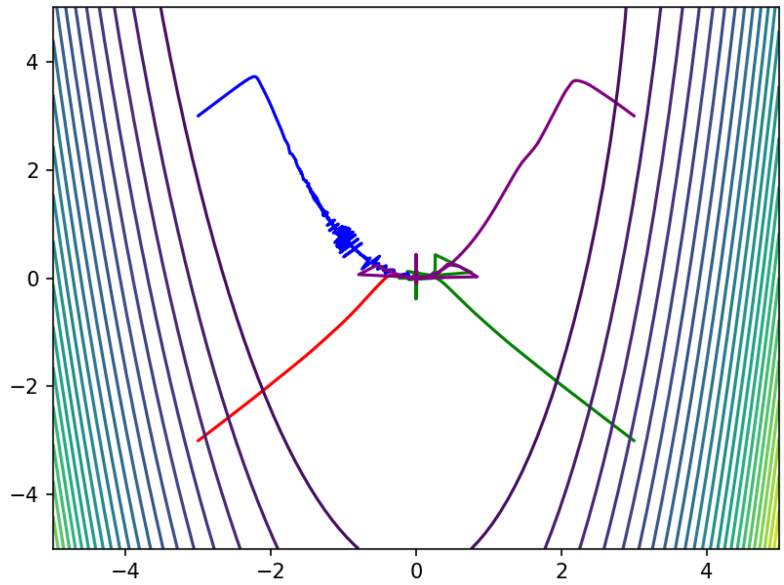
$$\therefore -\alpha \nabla f(x^k)^T \mathrm{diag}\left(\frac{1}{V_1^k}, \cdots, \frac{1}{V_n^k}\right) \le 0$$

$$\therefore f(x^k+\alpha d^k) - f(x^k) \le 0.$$

d^k is a descent direction.

▼ (ii)

Run q3_2.py and it shows that the 4 initial point go to same minimzer



iteration count = 261

iteration count = 343

iteration count = 1099

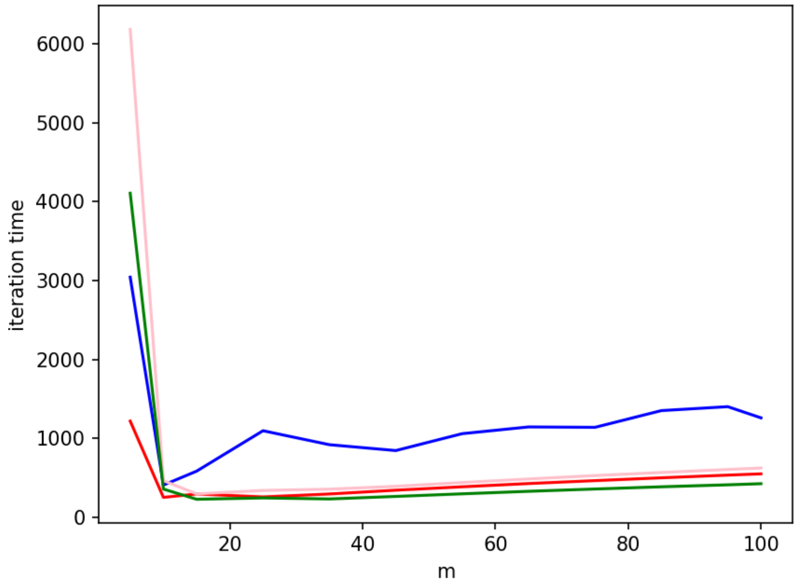
iteration count = 248

Perfomance and behavior

iteration around 200 which is small, the trajactory is first smooth later oscillate, finally goes to minimizer

▼ (iii)

run q3_3.py



Four initial points and Adagrad performance

Adagrad behavior first reduces iteration, then slowly increases.

Q4

(i)

$$s^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k)$$

with Newton method

x = 0.9999999276013231

y = 0.9999998547431523

iteration count = 9958

Newton direction number = 9958 used

alpha is 1_number time = 0, which is never

with Gradiant descent method and back track line search

x = 1.0000000786936023

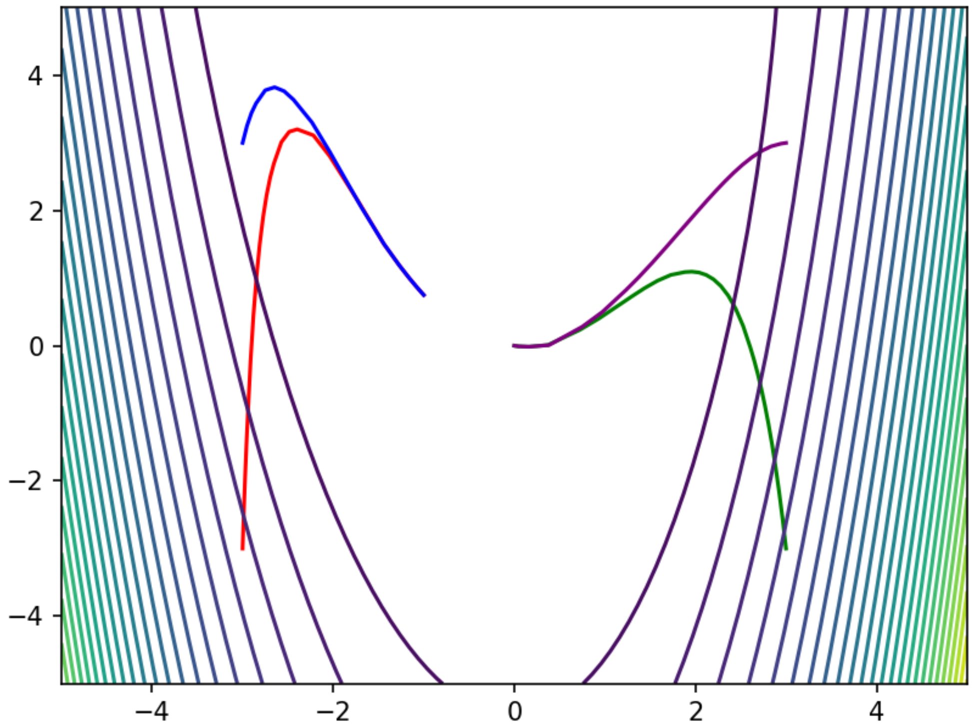
y = 1.0000001578605724

f(x,y) = 0

iteration count = 16896

▼ (ii)

run q4_2.py



the four results are

```
x =  -1.000117328126872
y =  0.7501759922095955
count =  32
Newton_direction_number =  32
alpha_is_1_number 15
x =  1.101694786594565e-08
y =  -2.751615930804218e-09
count =  32
Newton_direction_number =  32
alpha_is_1_number 6
x =  -1.000114886575062
y =  0.7501723298807026
count =  24
Newton_direction_number =  24
alpha_is_1_number 15
x =  1.9566776627232517e-08
y =  -4.886008080367979e-09
count =  23
Newton_direction_number =  23
alpha_is_1_number 6
```

Perfomance and behavior

iteration around 20 to 30 which is small, the trajactory is smooth