HW1 report

1. 我利用 python 實現 bisection，選擇 a, b 並且 $m = \frac{a+b}{2}$，if $f(m) \cdot f(a) < 0$, $b = m$, else $a = m$.

我最後選擇了十九個區間來找根，從 [0.9, 0.905], [0.905, 0.91], ..., [0.99, 0.995]，我找到最接近 0.95 的四個根由小到大分別是

$x = 0.94397$, iteration 次數為 9, start interval [0.94, 0.945]

$x = 0.95235$, iteration 次數為 9, interval [0.95, 0.955]

$x = 0.95856$, iteration 次數為 9, interval [0.955, 0.96]

$x = 0.96333$, iteration 次數為 9, interval [0.96, 0.965]

我一開始只選九個區間從 [0.9, 0.91], [0.91, 0.92], ... 到 [0.98, 0.99]，但我發現會沒有 0.95856 這個根，原因是因為一個區間內不一定只有一個根或沒有根，所以後來我才把區間縮小並偵測沒有根的區間

```
No root or bisection fails 0
No root or bisection fails 0
0.913544921875 9
No root or bisection fails 0
No root or bisection fails 0
No root or bisection fails 0
0.9320019531250001 9
No root or bisection fails 0
0.943974609375001 9
No root or bisection fails 0
0.9523535156250003 9
0.9585644531249999 9
0.9633300781249998 9
0.9671191406250002 9
0.970205078125 9
0.9767285156250001 9
0.982958984375 9
0.988505859375 9
0.993720703124999 9
```

Result for Problem 1

2. 我利用 python 實現 secant method，

$x_2 = x_1 - f(x_1)\frac{x_0 - x_1}{f(x_0) - f(x_1)}$，$x_0 = x_1$，$x_1 = x_2$，if $|f(x_0)| < |f(x_1)|$，swap(x_0, x_1)

和上題一樣選擇了相同的十九個區間，找到最接近 0.95 的四個根由小到大分別是

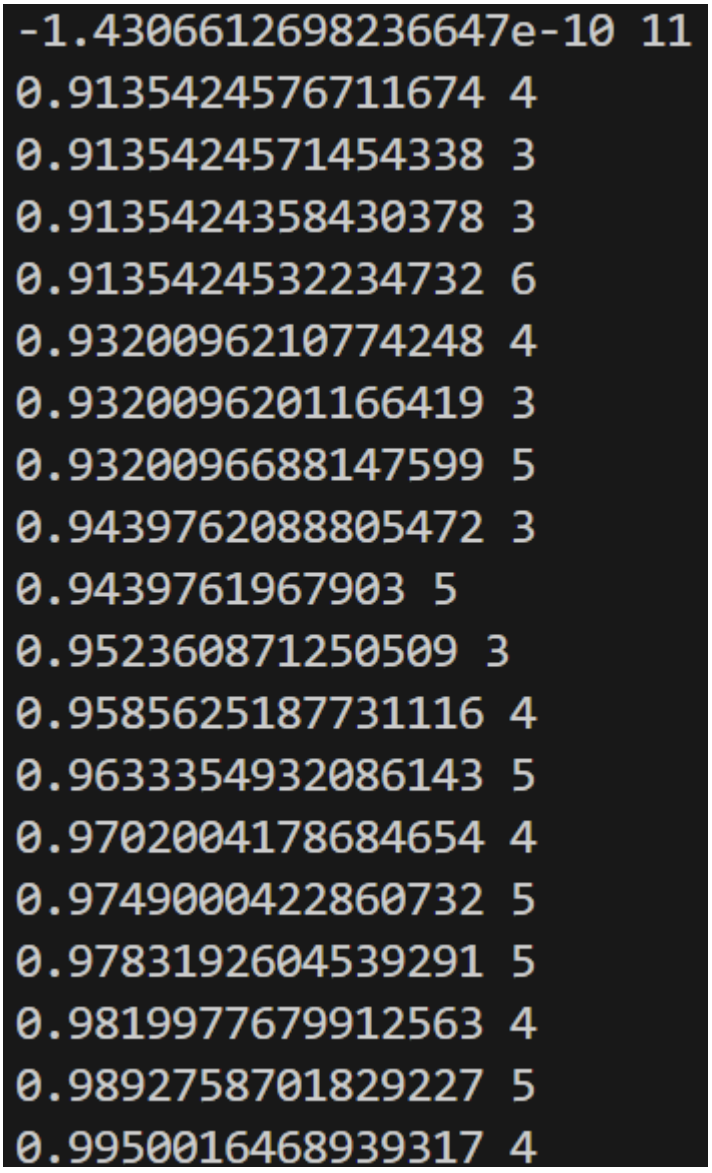
$x = 0.94398$，iteration 次數為 $\begin{cases} 3 \\ 5 \end{cases}$，start interval $\begin{cases} [0.94, 0.945] \\ [0.945, 0.95] \end{cases}$

$x = 0.95236$，iteration 次數為 3，start interval $[0.95, 0.955]$

$x = 0.95856$，iteration 次數為 4，start interval $[0.955, 0.96]$

$x = 0.96334$，iteration 次數為 5，start interval $[0.96, 0.965]$

可以發現 secant method 明顯收斂的比 bisection 快，且就算 start interval 內沒有根也可以找到其它根，不過若只取九個區間 $[0.9, 0.91]$ ~ $[0.98, 0.99]$ 一樣會有根沒被找到，所以我選擇分成十九個區間



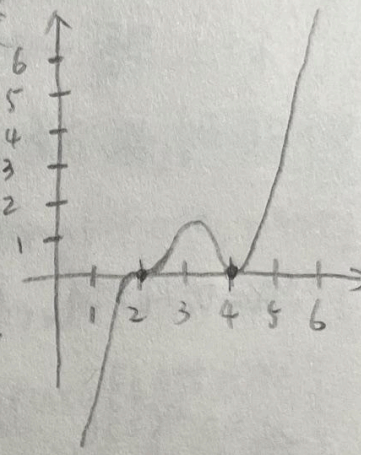
Result for Problem 2

3. 我用 python 來實現 bisection, secant, false position

False position $\Rightarrow X_2 = X_1 - f(X_1)\frac{X_0 - X_1}{f(X_0) - f(X_1)}$,

if $f(X_0)\cdot f(X_2) < 0$, $X_1 = X_2$, else, $X_0 = X_2$

我們可以先把大略圖畫出來觀察



(a) 我們可以考慮包含 root 的三種區間

(1) 只包含 $X=2$ 的區間 $\Rightarrow$ 可得到 root=2

(2) 只包含 $X=4$ 的區間 $\Rightarrow$ 無法得到 root, 因為此區間內 X 的函數值皆 ≥ 0

(3) 包含 2, 4 的區間 $\Rightarrow$ 結合上面兩種, 可得 root=2

$\Rightarrow$ bisection 只能找到 2, 不能找到 4

(b) 一樣考慮包含 root 的三種區間

(1) 只包含 $X=2$ 的區間, 一正一負, 會收斂至 root=2

(2) 只包含 $X=4$ 的區間, 兩者皆正, 不容易收斂到 4, 但我找到一個 start interval [2.5, 4.2] 可得 root=4

(3) 包含 2, 4 的區間 $\Rightarrow$ 結合上面兩種, 可得 root 應會收斂至 2

$\Rightarrow$ secant method 可以找到 2, 4

(c) 三種方法都會得到 root=2

```
4.000014583308609
```

Result for Problem 3 (b)(2)

```
2.0 2.0 2.0
```

Result for Problem 3 (c)

4、我利用 python 來實現 Muller's method.

$h_1 = x_1 - x_0$, $h_2 = x_0 - x_2$, $\gamma = \frac{h_2}{h_1}$, $c = f_0$.

$a = \frac{\gamma f_1 - f_0(1+\gamma) + f_2}{\gamma h_1^2 (1+\gamma)}$, $b = \frac{f_1 - f_0 - a h_1^2}{h_1}$

$root = x_0 - \frac{2c}{b \pm \sqrt{b^2 - 4ac}}$, rearrange x_0, x_1, x_2, root until converges

(a) 我選擇 $x_0 = 0.6$, $x_1 = 0.7$ $x_2 = 0.5$

得到的 root near 0.6 ⇒ 0.6057

(b) 我選擇 $x_0 = 1$, $x_1 = 1.1$, $x_2 = 0.9$ for the root near 1, 得到 1.2421

我選擇 $x_0 = -2$, $x_1 = -1.9$, $x_2 = -2.1$ for the root near -2, 得到 -2.2114

```
0.6057468441612972
1.2420971739763345 -2.2114013340350445
```

Result for Problem 4

5. 我利用 python 來實現 fixed-point method,

$x_{n+1} = g(x_n)$

(a) $g_1(x) = \sqrt{\frac{e^x}{2}}$, $g_2(x) = -\sqrt{\frac{e^x}{2}}$

把 1.5 和 -0.5 用 $g_1(x)$ 做 fixed-point method
都會得到 1.4879 ⇒ converges to the root near 1.5

把 1.5 和 -0.5 用 $g_2(x)$ 做 fixed-point method
都會得到 -0.5398 ⇒ converges to the root near -0.5

(b) 把 2.5, 2.6, 2.7 用 $g_1(x)$ 做 fixed-point method

$X_0 = 2.5, 2.6$ ⇒ 1.4879, $X_0 = 2.7$ ⇒ infinite

把 2.5, 2.6, 2.7 用 $g_2(x)$ 做 fixed-point method
都會得到 -0.5398 ⇒ 以上說明 do not converge to the root near 2.6 even though values near to the root are used to begin the iterations

(c) $e^x = 2x^2 \Rightarrow x = \ln(2x^2) \Rightarrow g_3(x) = \ln(2x^2)$

把 2.5, 2.6, 2.7 用 $g_3(x)$ 做 fixed-point method
都會得到 2.6178 ⇒ converges to the root near 2.6

```
1.487986526435826 1.487934302885673
-0.539836902383924 -0.539834145749554
```

Result for Problem 5 (a)

```
1.4879892359152165 1.4879843381733782 inf
-0.539834545085182 -0.5398345155698323 -0.5398344863544899
```

Result for Problem 5 (b)

```
2.6178356620578405 2.6178382769093305 2.6178979498219626
```

Result for Problem 5 (c)

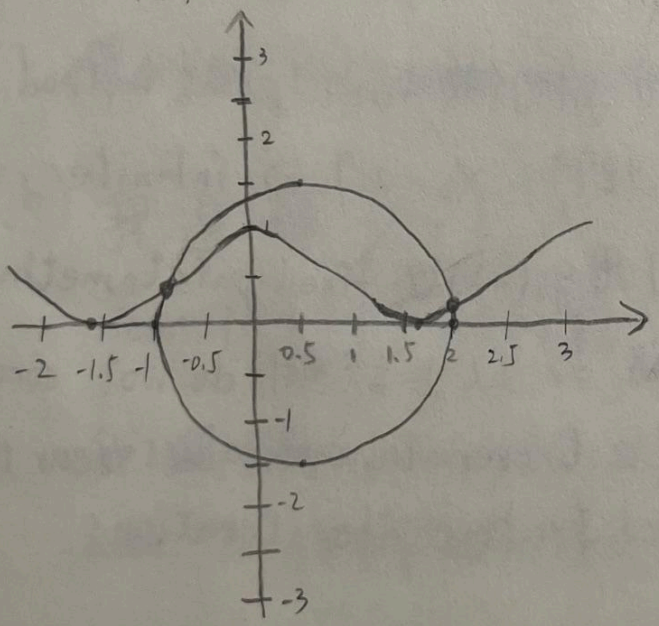
6. $f(x) = \begin{bmatrix} f_1(x,y) \\ f_2(x,y) \end{bmatrix} = \begin{bmatrix} \cos^2 x - y \\ x^2 + y^2 - x - 2 \end{bmatrix} = 0$

$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} -\sin 2x & -1 \\ 2x-1 & 2y \end{bmatrix}$,

然後利用 python 來實現 Newton's method

$x_{i+1} = x_i + s_i$, $J(x_i)s_i = -f(x_i)$ for $i \geq 0$,

但需要好的 initial guess X_0, 可以畫圖觀察 ⇒ 發現 $f_2(x,y) = (x-0.5)^2 + y^2 = 1.5^2$ 為一圓



可大略觀察到有二交點, 故選

$X_0 = [2, 0.2]^T$

$X_0 = [-0.9, 0.4]^T$

If $X_0 = [2, 0.2]$ ⇒ solution is $[1.9907595, 0.16624116]$

If $X_0 = [-0.9, 0.4]$ ⇒ solution is $[-0.96441693, 0.32478156]$

⇒ solution ⇒ $x = 1.99$, $y = 0.166$ and $x = -0.964$, $y = 0.325$

Result for Problem 6

```
[1.9907595   0.16624116]
[-0.96441693  0.32478156]
```