

牛頓法找根 F14081046 周呈陽

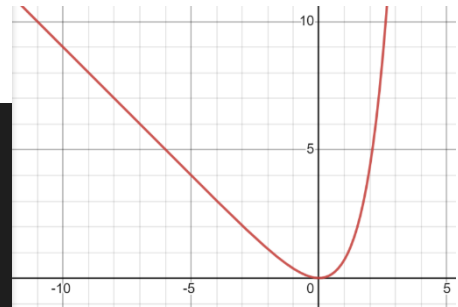
- 第一題 hw1

使用 Desmos 繪製方程式，在 $x=0$ 時最小值為 0，可以看出此方程式開口朝上恆正，因此無法用二分法找根。

演算法是老師課堂介紹的，此方程式在進入下方程式中的 while 產生無限迴圈，找不到根。

程式碼：

```
1  import math
2  x1  = -1
3  x2  = 1.5
4  tol = 1e-4
5  def f(x):
6      a = math.exp(x)-x-1.0
7      return a
8
9  while x1 <= x2:
10     x = 0.5 * (x1 + x2)
11     s = f(x1) * f(x)
12     print(x1,f(x1),x2,f(x2),x,f(x));
13     if abs(f(x)) <= tol:
14         break
15     elif s>0 :
16         x1 = x
17     elif s<0:
18         x2 = x
```



- 第二題 hw2_1

Hw2_1, 2_2, 3a_1, 3b_1 的整個演算法都是老師課堂介紹的，先判斷 $f(x_a)$ 和 $f(x_b)$ 相乘，若相乘小於 0 進到 newton function 去找解。

程式碼：

```
1  import math
2  def f(x):
3      return x**3 + 4.0*x**2 - 14.6      #eq2_1
4  def newton(xa,tol = 1e-5,N0=200):
5      x0=xa
6      for i in range(1,N0):              #老師的演算法
7          f = x0**3 + 4.0*x0**2 - 14.6  #eq2_1
8          fd = 3*x0**2 + 8*x0           #微分
9          x = x0-f/fd
10         if (abs(x-x0) < tol):
11             break;
12         i = i+1
13         x0 = x
14     print("solution:",x0,'tol:',abs(x-x0))
15
16     x1=-5.5 #起點
17     x2=5.5  #終點
18     n=11    #分段
19     dx=(x2-x1)/(n-1) #算delta x
20
21     for i in range(1,n):
22         xa=x1+(i-1)*dx
23         xb=x1+i*dx
24         if f(xa)*f(xb)>0:
25             print('no root between',xa,'and',xb)
26         if f(xa)==0:
27             print('root=',xa,'error=',f(xa))
28         if f(xa)*f(xb)<0:
29             print('have root between',xa,'and',xb)
30             newton(xa)
```

Output:

```
C:\Users\Windows>C:/Users/Windows/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/Windows/Desktop/hw2_1.py
no root between -5.5 and -4.4
no root between -4.4 and -3.3
no root between -3.3 and -2.1999999999999997
no root between -2.1999999999999997 and -1.0999999999999996
no root between -1.0999999999999996 and 0.0
no root between 0.0 and 1.1000000000000005
have root between 1.1000000000000005 and 2.2000000000000001
solution: 1.612819903974884 tol: 9.567118208764214e-10
no root between 2.2000000000000001 and 3.3000000000000007
no root between 3.3000000000000007 and 4.4
no root between 4.4 and 5.5
```

- 第二題 hw2_2
(想法同 hw2_1)

程式碼:

```
1  import math
2  def f(x):
3      return math.exp(x)-x-1      #eq2_2
4  def newton(xa,tol = 1e-5,N0=200):
5      x0=xa
6      for i in range(1,N0):      #老師的演算法
7          f = math.exp(x0)-x0-1  #eq2_2
8          fd = math.exp(x0)-1    #微分
9          x = x0-f/fd
10         if (abs(x-x0) < tol):
11             break;
12         i = i+1
13         x0 = x
14     print("solution:",x0,'tol:',abs(x-x0))
15
16     x1=-5.5 #起點
17     x2=5.5  #終點
18     n=11    #分段
19     dx=(x2-x1)/(n-1) #算delta x
20
21     for i in range(1,n):
22         xa=x1+(i-1)*dx
23         xb=x1+i*dx
24         if f(xa)*f(xb)>0:
25             print('no root between',xa,'and',xb)
26         if f(xa)==0:
27             print('root=',xa,'error=',f(xa))
28         if f(xa)*f(xb)<0:
29             print('have root between',xa,'and',xb)
30             newton(xa)
```

Output:

```
C:\Users\Windows>C:/Users/Windows/AppData/Local/Microsoft/Windows/
:/Users/Windows/Desktop/數值hw1/hw2/hw2_2.py
no root between -5.5 and -4.4
no root between -4.4 and -3.3
no root between -3.3 and -2.1999999999999997
no root between -2.1999999999999997 and -1.0999999999999996
root= 0.0 error= 0.0
no root between 1.1000000000000005 and 2.2000000000000001
no root between 2.2000000000000001 and 3.3000000000000007
no root between 3.3000000000000007 and 4.4
no root between 4.4 and 5.5
```

- 第三題 hw3a_1
(想法同 hw2_1)

程式碼:

```
1 import math
2 def f(x):
3     return x**2 + 4*x - 14.6 #eq3a_1
4 def newton(xa,tol = 1e-5,N0=200):
5     x0=xa
6     for i in range(1,N0):
7         f = x0**2 + 4*x0 - 14.6
8         fd = 2*x0 + 4
9         x = x0-f/fd
10        if (abs(x-x0) < tol):
11            break;
12        i = i+1
13        x0 = x
14    print("solution:",x0,'tol:',abs(x-x0))
15
16 x1=-10.5 #起點
17 x2=10.5 #終點
18 n=11 #分段
19 dx=(x2-x1)/(n-1) #算delta x
20
21 for i in range(1,n):
22     xa=x1+(i-1)*dx
23     xb=x1+i*dx
24     if f(xa)*f(xb)>0:
25         print('no root between',xa,'and',xb)
26     if f(xa)==0:
27         print('root=',xa,'error=',f(xa))
28     if f(xa)*f(xb)<0:
29         print('have root between',xa,'and',xb)
30         newton(xa)
```

Output:

```
C:\Users\Windows>C:/Users/Windows/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/Windows/Desktop/數值hw1/hw3a/hw3a_1.py
no root between -10.5 and -8.4
have root between -8.4 and -6.3
solution: -6.312771730606759 tol: 3.7194247681782144e-11
no root between -6.3 and -4.199999999999999
no root between -4.199999999999999 and -2.0999999999999996
no root between -2.0999999999999996 and 0.0
no root between 0.0 and 2.1000000000000014
have root between 2.1000000000000014 and 4.200000000000001
solution: 2.3127752598480518 tol: 3.529277042613188e-06
no root between 4.200000000000001 and 6.300000000000001
no root between 6.300000000000001 and 8.400000000000002
no root between 8.400000000000002 and 10.5
```

- 第三題 hw3a_2
(想法同 hw2_1)
程式碼:

```
1  import math
2  def f(x):
3      return (x-1)*(x-4)*(x+2.46) #eq3a_2
4  def newton(xa,tol = 1e-5,N0=200):
5      x0=xa
6      for i in range(1,N0):
7          f = (x0-1)*(x0-4)*(x0+2.46)
8          fd = 3*x0**2 - 5.08*x0 - 8.3
9          x = x0-f/fd
10         if (abs(x-x0) < tol):
11             break;
12         i = i+1
13         x0 = x
14     print("solution:",x0,'tol:',abs(x-x0))
15
16     x1=-5.5 #起點
17     x2=5.5  #終點
18     n=11    #分段
19     dx=(x2-x1)/(n-1) #算delta x
20
21     for i in range(1,n):
22         xa=x1+(i-1)*dx
23         xb=x1+i*dx
24         if f(xa)*f(xb)>0:
25             print('no root between',xa,'and',xb)
26         if f(xa)==0:
27             print('root=',xa,'error=',f(xa))
28         if f(xa)*f(xb)<0:
29             print('have root between',xa,'and',xb)
30             newton(xa)
31
```

Output:

```
C:\Users\Windows>C:/Users/Windows/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/Windows/Desktop/數值hw1/hw3a/hw3a_2.py
no root between -5.5 and -4.4
no root between -4.4 and -3.3
have root between -3.3 and -2.1999999999999997
solution: -2.460000004894417 tol: 4.894416871792373e-09
no root between -2.1999999999999997 and -1.0999999999999996
no root between -1.0999999999999996 and 0.0
have root between 0.0 and 1.1000000000000005
solution: 0.9999996492693447 tol: 3.507306498562457e-07
no root between 1.1000000000000005 and 2.2000000000000001
no root between 2.2000000000000001 and 3.3000000000000007
have root between 3.3000000000000007 and 4.4
solution: 4.000007981543869 tol: 7.981512773369559e-06
no root between 4.4 and 5.5
```

● Hw3b_1

方法:

老師課堂上所講解的是用兩個點去做牛頓法，但我的想法是只用左邊的點去慢慢逼近，理想只要 dx 夠小就可以慢慢逼近答案出來，在 28 行以前跟 hw2_1 的做法差不多，但為了解決 dx 太小，導致跑出類似的解，因此 28~37 行就是透過判斷其整數是否相等，最後僅取出相同整數第一次出現的解當作最後的 solution，但這個問題會有一個盲點，若題目是重根就會被刪除掉，由於第三題的兩個多項式都沒有重根的問題，因此得出的解跟用 3(a) 方法是差不多的。

程式:

```
1  import math
2  def f(x):
3      return x**2 + 4*x - 14.6    #3_1
4  def newton(xa, tol=1e-5, N0=200):
5      x0 = xa
6      for i in range(1, N0):
7          f = x0**2 + 4*x0 - 14.6
8          fd = 2*x0 + 4
9          x = x0 - f/fd
10         if abs(x - x0) < tol:
11             break
12         x0 = x
13     return x0, abs(x - x0)
14
15     x1 = -10.5 #起點
16     x2 = 10.5  #終點
17     n = 11     #分段
18     dx = (x2 - x1)/(n - 1) #算delta x
19     ans = []    #全部解答的list
20
21     results = {}
22     for i in range(1, n):
23         xa = x1 + (i - 1)*dx #只用左邊的點
24         x0, tol = newton(xa)
25         results[x0] = tol    #key=sol,val=tol
26         ans.append(x0)
27
28     seen_ints = {}          #建立一個dict
29     final_result = []      #答案的list
30     for num in ans:
31         int_part = int(num) #取整數部分
32         if int_part not in seen_ints: #如果int_part是新的數字
33             final_result.append(num) #增加S進final_result
34             seen_ints[int_part] = True #看過的整數標示為T
35     for i in range(len(final_result)):
36         print('\n', 'solution=', final_result[i], 'tol=', results[final_result[i]])
37         #用key叫出dict的val
```

Output:

```
C:\Users\Windows>C:/Users/Windows/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/Windows/Desktop/hw3b_1.py
```

```
solution= -6.312771876559612 tol= 1.459900449418683e-07
```

```
solution= 2.3127726392045798 tol= 9.086349193054843e-07
```

- 第三題 hw3b_2
(想法同 hw3b_1)

```
1 import math #3_1
2
3 def f(x):
4     return (x-1)*(x-4)*(x+2.46)
5
6 def newton(xa, tol=1e-5, N0=200):
7     x0 = xa
8     for i in range(1, N0):
9         f = (x0-1)*(x0-4)*(x0+2.46)
10        fd = 3*x0**2 - 5.08*x0 - 8.3
11        x = x0 - f/fd
12        if abs(x - x0) < tol:
13            break
14        x0 = x
15    return x0, abs(x - x0)
16
17 x1 = -10.5 #起點
18 x2 = 10.5 #終點
19 n = 11 #分段
20 dx = (x2 - x1)/(n - 1) #算delta x
21 ans = [] #全部解答的list
22 results = {}
23 for i in range(1, n):
24     xa = x1 + (i - 1)*dx #只用左邊的點
25     x0, tol = newton(xa)
26     results[x0] = tol
27     ans.append(x0)
28 seen_ints = {} #建立一個dict
29 final_result = [] #答案的list
30 for num in ans:
31     int_part = int(num) #取整數部分
32     if int_part not in seen_ints: #如果int_part是新的數字
33         final_result.append(num) #增加進final_result
34         seen_ints[int_part] = True #看過的整數標示為T
35 for i in range(len(final_result)): #用key叫出dict的val
36     print('solution=', final_result[i], 'tol=', results[final_result[i]])
37
```

Output :

```
C:\Users\Windows>C:/Users/Windows/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/Windows/Desktop/hw3b_2.py
```

```
solution= -2.460000006881508 tol= 6.8815082521211934e-09
solution= 0.9999996492693447 tol= 3.507306498562457e-07
solution= 4.000000009330488 tol= 9.330488381920077e-09
```


● 第四題 hw4_1

方法:

若遇到函數無法微分，那就跟微積分學到的一樣，回到微分的定義去做，如下圖，只要找出 $f(a+h)$, $f(a)$ 和 h ，就可以解決無法被微分的情況，我將 h 設為 10^{-5} ，在這樣的情況下得出的解跟 hw3_1 直接微分的解差不了多少，所以凡遇到 $f(x)$ 不可微分的函數， h 取的夠小就可以算出 $f'(a)$ 的值。

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h} \approx \frac{f(a+h) - f(a)}{h}$$

程式碼:

```
1 import math
2 def f(x):
3     return x**2 + 4*x - 14.6 #eq3a_1
4 def newton(xa, tol = 1e-5, N0=200):
5     x0=xa
6     for i in range(1, N0):
7         f = x0**2 + 4*x0 - 14.6
8         h=1e-5 #令h=10^-5
9         fh = (x0+h)**2 + 4*(x0+h) - 14.6 #算f(a+h)
10        fd = (fh-f)/h #用微分定義算出f'(a)
11        x = x0 - f/fd #下面步驟沿用hw3a
12        if (abs(x-x0) < tol):
13            break;
14        i = i+1
15        x0 = x
16    print("solution:", x0, 'tol:', abs(x-x0))
17
18 x1=-10.5 #起點
19 x2=10.5 #終點
20 n=11 #分段
21 dx=(x2-x1)/(n-1) #算delta x
22
23 for i in range(1, n):
24     xa=x1+(i-1)*dx
25     xb=x1+i*dx
26     if f(xa)*f(xb)>0:
27         print('no root between', xa, 'and', xb)
28     if f(xa)==0:
29         print('root=', xa, 'error=', f(xa))
30     if f(xa)*f(xb)<0:
31         print('have root between', xa, 'and', xb)
32         newton(xa)
```

Output:

```
C:\Users\Windows>C:/Users/Windows/AppData/Local/Microsoft/WindowsApps/python3.9.exe c:/Users/Windows/Desktop/hw4_1.py
no root between -10.5 and -8.4
have root between -8.4 and -6.3
solution: -6.312771730585946 tol: 1.638156277294911e-11
no root between -6.3 and -4.199999999999999
no root between -4.199999999999999 and -2.0999999999999996
no root between -2.0999999999999996 and 0.0
no root between 0.0 and 2.10000000000000014
have root between 2.10000000000000014 and 4.3000000000000001
solution: 2.3127752658961565 tol: 3.5353210439126315e-06
no root between 4.3000000000000001 and 6.3000000000000001
no root between 6.3000000000000001 and 8.400000000000002
no root between 8.400000000000002 and 10.5
```

● 第四題 hw4_2

(方法同上)

程式碼：

```
1  import math
2  def f(x):
3      return (x-1)*(x-4)*(x+2.46) #eq3a_2
4  def newton(xa,tol = 1e-5,N0=200):
5      x0=xa
6      for i in range(1,N0):
7          f = (x0-1)*(x0-4)*(x0+2.46)
8          h=1e-5 #令h=10^-5
9          fh = (x0+h-1)*(x0+h-4)*(x0+h+2.46) #算f(a+h)
10         fd = (fh-f)/h #用微分定義算出f'(a)
11         x = x0-f/fd #下面步驟沿用hw3a
12         if (abs(x-x0) < tol):
13             break;
14         i = i+1
15         x0 = x
16     print("solution:",x0,'tol:',abs(x-x0))
17
18     x1=-5.5 #起點
19     x2=5.5 #終點
20     n=11 #分段
21     dx=(x2-x1)/(n-1) #算delta x
22
23     for i in range(1,n):
24         xa=x1+(i-1)*dx
25         xb=x1+i*dx
26         if f(xa)*f(xb)>0:
27             print('no root between',xa,'and',xb)
28         if f(xa)==0:
29             print('root=',xa,'error=',f(xa))
30         if f(xa)*f(xb)<0:
31             print('have root between',xa,'and',xb)
32             newton(xa)
```

Output :

```
C:\Users\Windows>C:/Users/Windows/AppData/Local/Microsoft/WindowsAp
s/Desktop/hw4_2.py
no root between -5.5 and -4.4
no root between -4.4 and -3.3
have root between -3.3 and -2.1999999999999997
solution: -2.460000004421169 tol: 4.4211887484379986e-09
no root between -2.1999999999999997 and -1.0999999999999996
no root between -1.0999999999999996 and 0.0
have root between 0.0 and 1.1000000000000005
solution: 0.9999996504875982 tol: 3.495125512342767e-07
no root between 1.1000000000000005 and 2.2000000000000001
no root between 2.2000000000000001 and 3.3000000000000007
have root between 3.3000000000000007 and 4.4
solution: 4.0000080021626925 tol: 8.002092375392067e-06
no root between 4.4 and 5.5
```