數值方法 Hw2 E94114023 鍾俐欣

1 $9^{\frac{1}{5}} = 1.5518$

method a

$P_0 = 1$

$P_1 = (1 + \frac{9-1^3}{1^2})^2 = 81$

$P_2 = (1 + \frac{9-81^3}{81^2})^2 = 6399.78$

$P_3 = (1 + \frac{9-6399.78^3}{6399.78^2})^2 = 4594435$

⇒ method a is disvergent

method b

$P_0 = 1$

$P_1 = 1 - \frac{1^5-9}{1^2} = 9$

$P_2 = 9 - \frac{9^5-9}{9^2} = -719$

$P_3 = -719 - \frac{(-719)^5-9}{(-719)^2} = 3716942400$

⇒ method b is disvergent

method c

$P_0 = 1$

$P_1 = 1 - \frac{1^5-9}{5\times1^4} = 2.6$

$P_2 = 2.6 - \frac{2.6^5-9}{5\times2.6^4} = 2.11$

$P_3 = 2.11 - \frac{2.11^5-9}{5\times2.11^4} = 1.7788$

$P_4 = P_3 - \frac{P_3^5-9}{5\times P_3^4} = 1.6028$

$P_5 = 1.55498$

$P_6 = 1.5518$

⇒ $9^{\frac{1}{5}} = 1.5518$

method c 做 6 次可以得到相近的答案

method d

$P_0 = 1$

$P_1 = 1 - \frac{1^5-9}{12} = 1.66667$

$P_2 = 1.66667 - \frac{1.66667^5-9}{12} = 1.34489$

$P_3 = 1.72823$

$P_4 = 1.193457$

$P_5 = 1.74169$

$P_6 = 1.1561$

$P_7 = 1.7340$

$P_8 = 1.1776$

$P_9 = 1.7388$

$P_{10} = 1.1639$

$P_{11} = 1.7359$

⇒ method d 收斂到 $P_{11} = 1.7359$、$P_{10} = 1.1639$，和 $9^{\frac{1}{5}} = 1.5518$ 有 $\varepsilon = 0.1841$ 的誤差

⇒ method c、method d、method b、method a

---

4. $f(x) = x^4 - 5x^3 + 18x^2 - 34x + 20 = 0$

$= (x-1)(x^3 - 4x^2 + 14x + 20)$

$= (x-1)(x-2)(x^2 - 2x + 10)$

$x = \frac{2 \pm \sqrt{4-40}}{2}$

$= \frac{2 \pm \sqrt{-36}}{2}$

$= \frac{2 \pm 6i}{2}$

$= 1 \pm 3i$

⇒ All roots : $1, 2, 1+3i, 1-3i$

$x-1$ 的綜合除法：

	1	−5	18	−34	20
1		1	−4	14	20
	1	−4	14	20	20

$x-2$ 的綜合除法：

	1	−4	14	20
2		1	−2	10
	1	−2	10	20

5 $f(x) = x^5 + 11x^4 - 21x^3 - 10x^2 - 21x - 5$

$f'(x) = 5x^4 + 44x^3 - 63x^2 - 20x - 21$

$x_0 = 0$

$f(x_0) = -5$

$f'(x_0) = -21$

$x_1 = 0 - \frac{-5}{-21} = -0.238095$

$f(x_1) = -0.248861$

$f'(x_1) = -20.38734$

$x_2 = -0.238095 - \frac{-0.248861}{-20.38734} = -0.25031$

$f(x_2) = 0.001338$

$f'(x_2) = -20.61134$

$x_3 = -0.25031 - \frac{0.001338}{-20.61134} = -0.25024$ #

$f(x_3) = 4.1117 \times 10^{-8}$

3、將每一種方法寫成程式並輸出

```
def f(x):
    y=-26+(85*x)-(91*x**2)+(44*x**3)-(8*x**4)+x**5
    return y

def df(x):
    y=85-(182*x)+(132*x**2)-(32*x**3)+(5*x**4)
    return y
def df2(x):
    y=-21-(20*x)-(63*x**2)+(44*x**3)+(5*x**4)
    return y
def bis(a,b):
    A=a
    B=b
    err=0.0001
    c=(A+B)/2
    while abs(f(c))>err:
        if f(c)<0 and f(A)<0:
            A=c
        else:
            B=c
        c=(A+B)/2
    return c
```

```
def falsepos(a,b):
    err=0.0001
    p0=a
    p1=b
    fp0=f(p0)
    fp1=f(p1)
    p2=p0-((p1-p0)*fp0)/(fp1-fp0)
    while abs(f(p2))>err:
        if f(p2)<0 and f(p1)<0:
            p1=p2
        else:
            p0=p2
        fp0=f(p0)
        fp1=f(p1)
        p2=p0-((p1-p0)*fp0)/(fp1-fp0)
    return p2
def newton(a):
    err=0.0001
    x0=a
    x1=x0-(f(x0)/df(x0))
    while abs(f(x1))>err:
        x0=x1
        x1=x0-(f(x0)/df(x0))
    return x1
def sec(a,b):
    err=0.0001
    x0=a
    x1=b
    x2=x1-(((x1-x0)*f(x1))/(f(x1)-f(x0)))
    while abs(f(x2))>err:
        x0=x1
        x1=x2
        x2=x1-(((x1-x0)*f(x1))/(f(x1)-f(x0)))
    return x2
```

```
def ste(a):
    err=0.0001
    p0=a
    p1=p0-(f(p0)/((f(p0+f(p0))/f(p0))-1))
    while abs(f(p1))>err:
      p0=p1
      p1=p0-(f(p0)/((f(p0+f(p0))/f(p0))-1))
    return p1

rootbis=bis(0,1)
print('Bisection method: '+str(rootbis))
rootfp=falsepos(0,1)
print('False-position method: '+str(rootfp))
rootnew=newton(0)
print('Newton's method: '+str(rootnew))
rootsec=sec(0, 1)
print('Secant method: '+str(rootsec))
rootste=ste(0)
print('Steffensen's method: '+str(rootste))
```

輸出結果:

```
In [50]: %runfile 'C:/Users/chung/OneDrive/Desktop/數值方法
hw2.py' --wdir
Bisection method: 0.5570220947265625
False-position method: 0.5570303113587685
Newton's method: 0.5570255051379259
Secant method: 0.5570257928969453
Steffensen's method: 0.5570251996000377
```