Numerical Methods Homework-2

B10602110 四電子三乙 呂和軒

1. Consider the following equation:

$$f(x) = 2x + \frac{3}{x}$$

(a) Iterate parabolic interpolation 10 times to locate the minimum.

ANS :

1.Code_function :

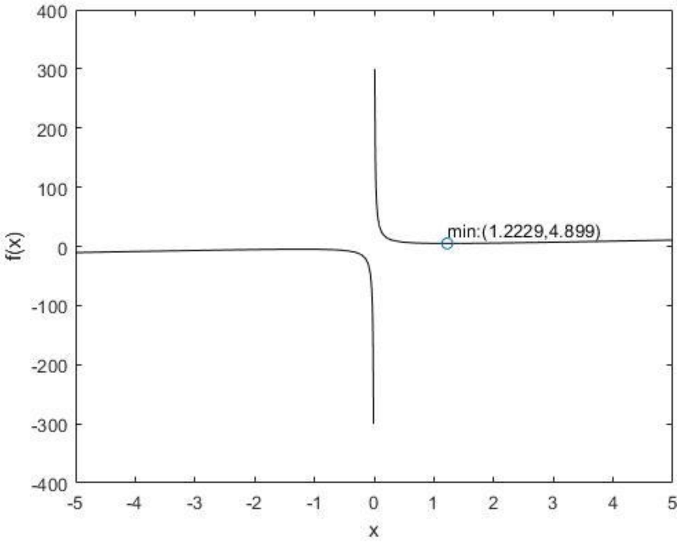
```
function [xe]= parabolic(f,x1,x2,x3,iter)
        x_m = [x1;x2;x3];
        for n = 1:iter
                n
                f_m = f(x_m);
                ai_2 =
f_m(1)*(x_m(2)^2-x_m(3)^2)+f_m(2)*(x_m(3)^2-x_m(1)^2)+f_m(3)*(x_m(1)^2
-x_m(2)^2)
                ai_3 =
f_m(1)*(x_m(2)-x_m(3))+f_m(2)*(x_m(3)-x_m(1))+f_m(3)*(x_m(1)-x_m(2))
                xe = ai_2/(2*ai_3)
                if(ismember(xe,x_m))
                      break;
                end
                [f_max,pos]= max(f_m,[],1);
                x_m(pos) = xe;
        end
end
```

2.Code_main :

```
close all
clear all
format long

x1 = 0.1
x2 = 1
x3 = 5
f = @(x) 2.*x+3./x;
iter = 10;
[x_min]= parabolic(f,x1,x2,x3,iter)
x=-5:0.01:5;
plot(x,f(x),'-k')
hold on
plot(x_min,f(x_min),'o')
text(x_min,f(x_min)+25,'min:('+string(x_min)+','+string(f(x_min))+')')
xlabel('x')
ylabel('f(x)')
```

3.Result :



使用 0.1、1、5 作為初始值，迭代 10-次的結果。

在 x=1.2229 時有區域最小值 4.899

```
x_min =

   1.222856768427421
```

圖 4-1-a local minimum

(b) Comment on the convergence of your results (x1 =0.1, x2 =0.5, x3 =5).
ANS :

1.Result :

在 0 到點最小值之間為單調下降曲線，而點最小值至 5 之間為單調上升曲線，符合二次曲線特徵，因此可以使用拋物線插值找出最小值，但經過插值得出的頂點必須也落在 0 至 5 之間，若落在負區間則結果會發散。

2. Find the local minima and maxima of the following function ranging from 0 to 2.

$$f(x) = 9e^{-x}\sin(2\pi x) - 3.5$$

(a) Use golden-section search.
ANS :

1.Code_function :

```
function [x]= golden_section(f,x1,x2,iter)
    gr = (sqrt(5) + 1) / 2;
    x3 = x1+(x2-x1)/gr
    x4 = x2-(x2-x1)/gr
    for n=1:iter
        n
        if(f(x3)>f(x4))
            x2 = x3;
        else
            x1 = x4;
        end
        x3 = x1+(x2-x1)/gr
        x4 = x2-(x2-x1)/gr
    end
```

```
    x = (x3+x4)/2
end
```

2.Code_main :

```
close all
clear all
format long

x1 = 0
x2 = 2
f = @(x) (9*exp(-x).*sin(2*pi*x)-3.5);
inv_f = @(x) -(9*exp(-x).*sin(2*pi*x)-3.5);
iter = 50;
[x_min]= golden_section(f,x1,x2,iter)
[x_max]= golden_section(inv_f,x1,x2,iter)
figure(1)
x=-0.5:0.01:2.5;
plot(x,f(x),'-k')
hold on
text(x_min,f(x_min)-0.6,'min:('+string(x_min)+','+string(f(x_min))+')')
text(x_max+0.1,f(x_max),'max:('+string(x_max)+','+string(f(x_max))+')')
plot(x_min,f(x_min),'ro')
plot(x_max,f(x_max),'gx')
xlabel('x')
ylabel('f(x)')
```

3.Result :

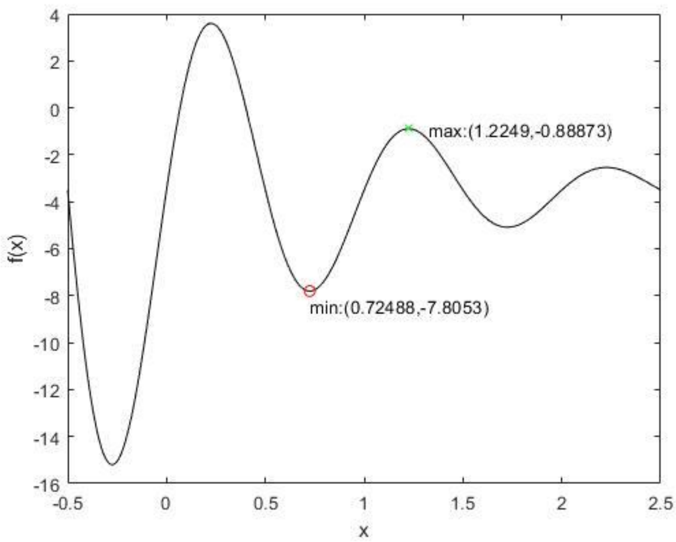


圖 4-2-a-1 local minimum and maxima with initial value 0、2

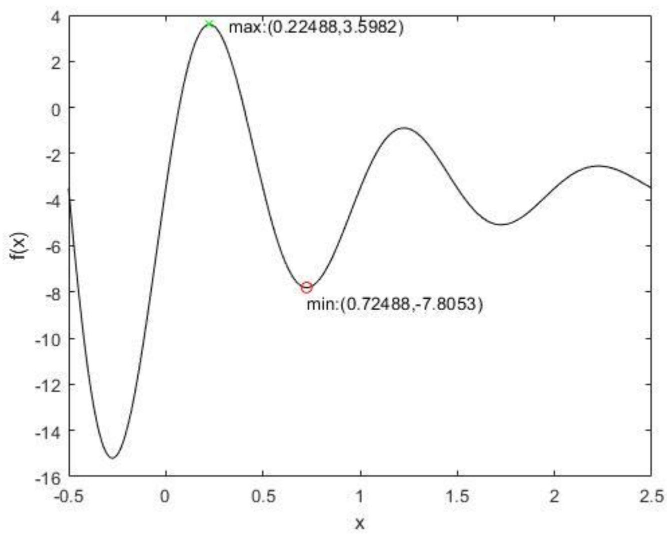


圖 4-2-a-2 local minimum and maxima with initial value 0、1

若初始值使用 0、2，則會因為計算得出的步幅過大而直接跨過最大值所在的位置，而找到錯誤的值。

(b) Use the brute force stepwise.
ANS :

1.Code_function :

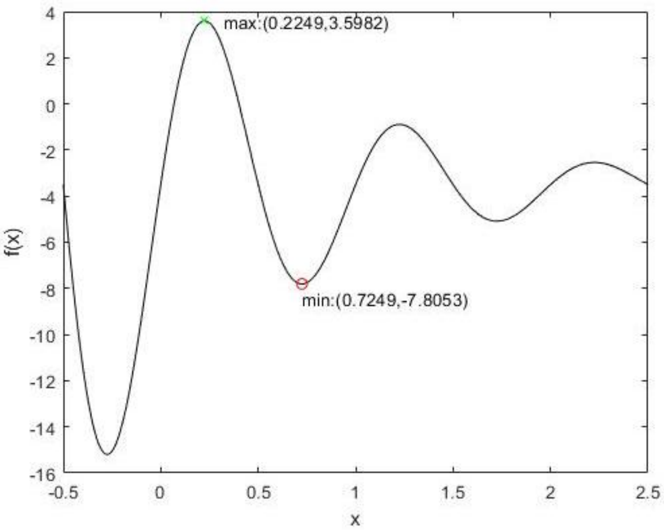
```
function [x_min,x_max]= brute(f,x1,x2,precise)
      x_range = x1:precise:x2;
      f_m = f(x_range);
      [f_min,pos_min] = min(f_m);
      [f_max,pos_max] = max(f_m);
      x_min = x_range(pos_min);
      x_max = x_range(pos_max);
end
```

2.Code_main :

```
close all
clear all
format long

x1 = 0
x2 = 2
f = @(x) (9*exp(-x).*sin(2*pi*x)-3.5);
precise = 0.0001;
x=-0.5:0.01:2.5;
[x_min,x_max]=brute(f,x1,x2,precise)
plot(x,f(x),'-k')
hold on
text(x_min,f(x_min)-0.6,'min:('+string(x_min)+','+string(f(x_min))+')')
text(x_max+0.1,f(x_max),'max:('+string(x_max)+','+string(f(x_max))+')')
plot(x_min,f(x_min),'ro')
plot(x_max,f(x_max),'gx')
xlabel('x')
ylabel('f(x)')
```

3.Result :



使用精度為 0.0001，範圍從 0 到 2。

圖 4-2-b local minimum and maxima

3. Given:

$$f(x) = -2x^6 - 1.5x^4 + 10x + 2$$

Use a root-location technique to determine the maximum of this function.

ANS :

1.Code_function :

```
function[root] = newton(f,x0,c,iter,error_std)
    eps = 1e-6;
    for n = 1:iter
        n;
        m = (f(x0+eps,c)-f(x0,c))/eps;
        x1 = x0-f(x0,c)/m;
        error = abs(x1-x0)/x0;
        x0 = x1;
    end
    root = x1;
end
```

2.Code_main :

```
close all
clear all
format long

f = @(x,c) -2.*x.^6-1.5.*x.^4+10.*x+2+c
%f_b = @(x) -2.*x.^6-1.5.*x.^4+10.*x+2
x0 = 5;
c = 0;
iter = 20
error_std = 0.00
c_iter = 0
x= -0.5:0.01:1;
plot(x,f(x,0))
hold on
root0 = 0;
root1 = 0;
root = [];
while(1)
    x0 = 5;
    root0 = newton(f,x0,c,iter,error_std);
    x0 = -5;
    root1 = newton(f,x0,c,iter,error_std);
    if(root1 > root0)
        max_range = root
        break
    end
    root = [];
    root = [root,root0,root1];
    f_root = f(root,0);
```

```
        plot(root(1),f_root(1),'ro')
        hold on
        plot(root(2),f_root(2),'bx')
        c = c - 0.005;
        c_iter = c_iter + 1
end
xlabel('x')
ylabel('f(x)')
figure(2)
x= max_range(2)-0.01:0.0001:max_range(1)+0.01;
plot(x,f(x,0))
hold on
plot(max_range(2),f(max_range(2),0),'bx')
hold on
plot(max_range(1),f(max_range(1),0),'ro')
xlabel('x')
ylabel('f(x)')

%[x_min,x_max]=brute(f_b,0.87,0.89,0.0000001)
```

3.Result :

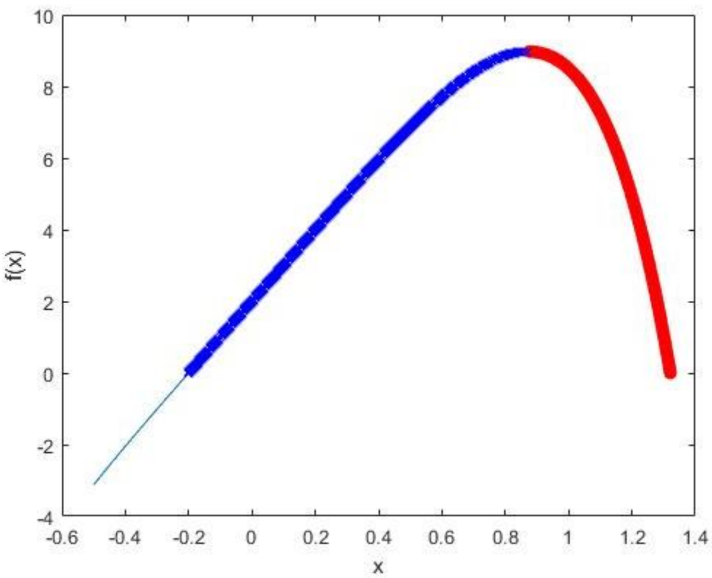


圖 4-3-a find roots per f(x)-c

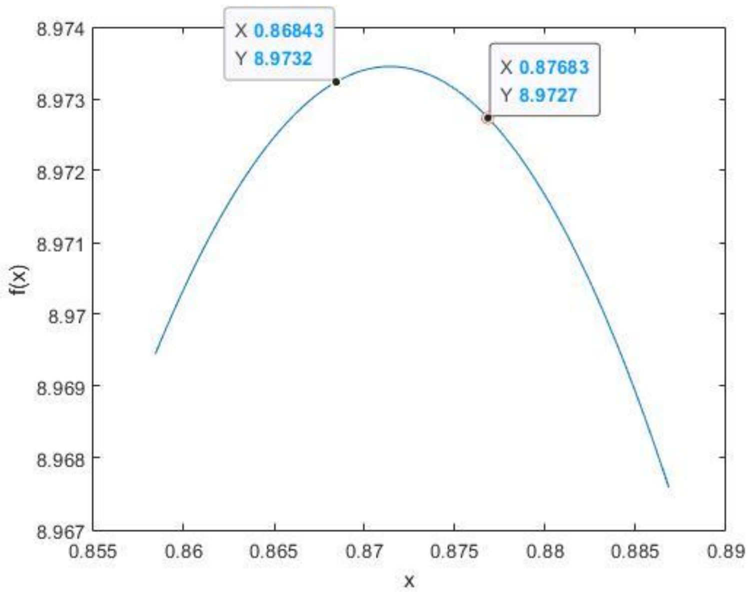


圖 4-2-b maxima range

使用牛頓法找根，先從左右兩邊各找出一個根，接下來將函數減去 0.005 的偏移量，使 f(x)圖形下降，再找出兩個根，漸漸逼近最大值，直到兩根交錯而停止，就可得到最大值的區間範圍。使用牛頓法的原因是因為函數為連續的，且在最大值附近的斜率變化較為平緩，沒有劇烈的變化，圖形凹口一致向下，使用牛頓法可以較穩定的找到根。