CompPhys Assignment 02

李尚坤 20307130215

1. 求解方程$x^3-5x+3=0$的两个正根

1.1. 题目描述

Sketch the function $x^3-5x+3=0$:

(1) Determine the two positive roots to 4 decimal places using the bisection method.

(2) Take the two roots that you found in the previous question (accurate to 4 decimals) and "polish them up" to 14 decimal places using the Newton-Raphson method.

(3) Determine the two positive roots to 14 decimal places using the hybrid method.

1.2. 解决方案描述

(1) Bisection method

• 首先以 0.5 为步长，找出使得函数符号相反的两个区间，即为两个正根所在的区间（这一步骤也可以通过观察函数图像代替，但 C/C++中绘图不便，因此用此方法）。

• 在这两个区间内分别用二分法求出函数的根。

• 二分法误差小于最后一次区间长度，因此当区间长度小于 0.0001 时，即精确到四位小数。

(2) Newton-Raphson method

• 将之前求得的两个根作为初始值，采用牛顿迭代公式进行求解；

• 当迭代中对根的修正值小于10^{-14}时，即精确到十四位小数。

(3) Hybrid method

• 将(1)中所求区间作为求根的初始区间[a,b]，以区间中点作为初始的根x

• 如果函数在x处切线与 x 轴不平行，则用牛顿迭代法进行下一次迭代。牛顿法所得结果如果在当前区间[a,b]内，则根据迭代后对应函数值的正负取新区间；若不在区间[a,b]内，舍弃牛顿迭代的结果，并采用二分法求解。

• 如果函数在x处切线与 x 轴平行，则用二分法进行下一次迭代。

• 当满足$|f(x)|<10^{-14}$时，即可认为x满足精度（注：此处不能使用区间长度作为精度判断标准）

1.3. 伪代码

Algorithm 1.1 Bisection method

Input: the range of the two positive roots [0,10], $a=0, b=10$

Output: two positive roots (4 decimal places) of the equation

Def $f(x)=x^3-5x+3$

End Def

Def FindBracket() //用于找到两个正根所在区间

  For $a \in [0,10]$ by step 0.5 **Do**

    If $f(a)*f(a+0.5)<0$

```
            Then bracket ← [a, a + 0.5]          //bracket为一个2 × 2的二维数组
            End If
        End for
    End Def
    Def FindRoot_bis(a, b)
        While abs(a − b) > 0.0001
            If f(a) ∗ f((a + b)/2) < 0
                Then b ← (a + b)/2
            Else
                a ← (a + b)/2
            End If
        End While
    End Def
    FindBracket()
    (a, b) ← bracket[0], (c, d) ← bracket[1]
    x1 ← FindRoot_bis(a, b), x2 ← FindRoot_bis(c, d)
    Return x1, x2
    End
```

Algorithm 1.2 Newton-Raphson method

Input: two roots $x1, x2$ obtained by the former method

Output: two positive roots (14 decimal places) of the equation

```
    Def f(x) = x^3 − 5x + 3, f′(x) = 3x^2 − 5
    End Def
    Def FindRoot_new(x)
        While abs(f(x)/f′(x)) > 10^14
            x ← x − f(x)/f′(x)
        End While
        Return x
    End Def
    x1 ← FindRoot_new(x1), x2 ← FindRoot_new(x2)
    Return x1, x2
    End
```

Algorithm 1.3 Hybrid method

Input: the intervals obtained in former method of the two roots: (a, b), (c, d)

Output: two positive roots (14 decimal places) of the equation

```
    Def f(x) = x^3 − 5x + 3, f′(x) = 3x^2 − 5
    End Def
    Def FindRoot_hyb(a, b)
         x ← (a + b)/2
         While abs(f(x)) > 10^−14
```

```
    If f'(x) ≠ 0
        Then x ← x − f(x)/f'(x)
        If a < x < b
            Then If f(a) ∗ f(x) < 0
                b ← x
            Else
                a ← x
            End If
        Else
            x ← (a + b)/2
            If f(a) ∗ f(x) < 0
                Then b ← x
            Else
                a ← x
            End If
            x ← (a + b)/2
        End If
    Else
        x ← (a + b)/2
        If f(a) ∗ f(x) < 0
            Then b ← x
        Else
            a ← x
        End If
        x ← (a + b)/2
    Return x
    End While
End Def
x1 ← FindRoot_hyb(a, b), x2 ← FindRoot_hyb(c, d)
Return x1, x2
End
```

1.4. 输入/输出示例

本题在程序中设定两个正根的存在范围为 0～10，在此范围内查找两个正根，结果如表 1 所示。终端的输出结果如图所示。

Method	Outputs		Note
	$x1$	$x2$	
Bisection	0.6566	1.8342	4 decimals places
Newton-Raphson	0.65662043104711	1.83424318431392	14 decimals places
Hybrid	0.65662043104711	1.83424318431392	14 decimals

表 1 输出结果

			places

```
Using the bisection method:
x1=0.6566
x2=1.8342

Using the Newton-Rapson method:
x1=0.65662043104711
x2=1.83424318431392

Using the Hybrid method:
x1=0.65662043104711
x2=1.83424318431392
```

图 1 终端输出结果

1.5. 用户手册

- 本程序源文件为 SketchFuntion.cpp，可执行文件为 SketchFuntion.exe
- 为了防止直接运行.exe 文件时 cmd 在运行结束后直接关闭，在源文件的末尾增加了一行命令 system(“pause”)
- 本程序使用了三种方法求解方程$x^3-5x+3=0$，直接运行可执行文件，可以得到三种方法计算得到的结果
- 可通过修改 FindBracket 函数中 start 和 end 的值修改两个根存在的大致区间，本程序已设置为 0 和 10.

2. 寻找函数的全局最小值

2.1. 题目描述

Search for the minimum of the function $g(x,y)=\sin(x+y)+\cos(x+2y)$ in the whole place

2.2. 解决方案描述

本题目采用梯度下降法求得函数的全局最小值
(1)$g(x,y)$为一个周期函数，周期为2π
(2)因此我们只需要在一个$2\pi\times2\pi$的平面区域内找到最小值点，就可将其作为全局最小值点
(3)给定一个初始点$(x0,y0)$,我们在$[x0,x0+2\pi]\times[y0,y0+2\pi]$的平面内随机取 30 个点
(4)从这 30 个点出发，分别沿负梯度方向移动，分别到达对应的极小值点；
(5)从这 30 个局部极小值点中找到最小值，将其作为全局最小值点
(6)本算法中隐含有一个假设：就是从 30 个随机取点出发，一定可以找到函数所有的局部极小值
该假设的合理性在于：平面的大小仅为$2\pi\times2\pi$，并且函数是连续的，因此选取 30 个初始点就可以很好地代表这个局部平面内的点

2.3. 伪代码

Algorithm 2 寻找函数全局最小值

Input: initial point $(x0, y0)$, steps a
Output: minimum of the function and the point

Def Find_min($a, x0, y0$)
  For $i \leftarrow 1$ **to** 30 **by** 1 **Do**
    $(x, y) \leftarrow$ random point in plane $[x0, x0 + 2\pi] \times [y0, y0 + 2\pi]$
    While $|\nabla_x g(x, y)| > EPS$ **or** $|\nabla_x g(x, y)| > EPS$
      $x \leftarrow x - a \cdot \nabla_x g(x, y)$
      $y \leftarrow y - a \cdot \nabla_y g(x, y)$
    End While
    $min \leftarrow min + (x, y)$
  End For
  For each (x, y) in min **Do**
    If $g(x, y) < g(min\_x, min\_y)$
      Then $(min\_x, min\_y) \leftarrow (x, y)$
    End If
  End For
  Return $min\_x,\ min\_y,\ g(min\_x, min\_y)$
End Def
Find_min($a, x0, y0$)
End

2.4. 输入输出示例

本题的通过输入不同的初始点的位置和步长，所得的结果如表 2 所示。程序在终端执行的输入输出图如图 2 所示。

Inputs			**Outputs**		**Note**
a	$x0$	$y0$	(min_x, min_y)	$minima$	
0.1	1	5	(0.000007, 10.995570)	−2.000000	所得的全局最小值都相同
0.01	−5	−6	(−6.283178, −1.570801)	−2.000000	
0.001	100	−200	(100.530972, −196.349545)	−2.000000	

表 2 输入输出结果

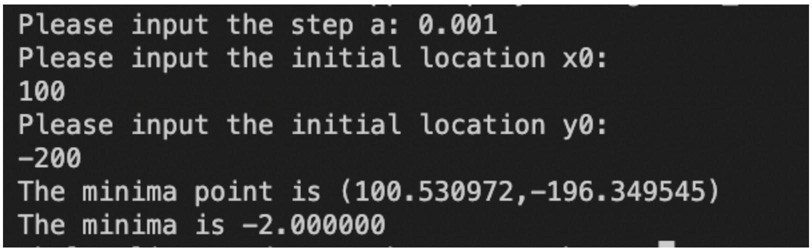

图 2 终端输入输出结果

2.5. 用户手册

- 本程序源文件为 FindMin.cpp，可执行文件为 FindMin.exe
- 为了防止直接运行.exe 文件时 cmd 在运行结束后直接关闭，在源文件的末尾增加了一行命令 system(“pause”)
- 本程序使用梯度下降法求得了周期函数$g(x,y)=\sin(x+y)+\cos(x+2y)$在全平面的最小值
- 运行程序后，首先会跳出提示语“Please input the step a:”，提示输入梯度下降的步长，用键盘键入一个数字之后按回车键结束
- 随后先后跳出提示语“Please input the initial location x0:”与“Please input the initial location y0:”，用键盘分别键入后按回车键结束，随后程序输出最小值以及取得最小值所在的点

3. 求解隐函数$\boldsymbol{m(t)=tanh(m(t)/t)}$

3.1. 题目描述

Determine $m(t)$ the reduced magnetization as a function of reduced temperature for simple materials. $m(t)=tanh(m(t)/t)$
For a given t, solve m, plot m as a function of t.

3.2. 解决方案描述

(1) 首先研究方程$f(m,t)=\tanh(m/t)-m$解的情况，即以 0.5 为步长取定t分别为$[-2,2]$中的数，作出$f(m)$的图像
(2) 由图可以知道，当$t\le 0$与$t\ge 1$的时候，函数都只有$m=0$这一个零点；当$0<t<1$时，函数有三个零点，除$m=0$这一平凡解之外，还有关于原点对称的两个零点，并且这两个零点的绝对值总在 0 到 1 之间
(3) 考虑真实的物理场景：约化温度$t>0$，$m(t)$的两个非零解大小相等，正负代表磁化方向，$m(t)$的平凡解则不予考虑
(3) 因此我们只需要着重分析当$t\in(0,1)$时$m\in(0,1)$的解, 这一部分的解可以由二分法求得

3.3. 伪代码

Algorithm 3 Temperature dependence of magnetization

Input: the interval m may exsit $(0.000001, 1)$

Output: the graph of $m(t)$

```
For t ← −2 to 2 by 0.2 Do
    For m ← −5 to 5 by 0.005 Do
        Plot f(m,t)
    End For
End For
Def FindRoot(a,b,t)
    If t > 0
        Then Return 0
    Else
        While abs(a − b)> 0.0001
            If f(a,t) ∗ f((a + b)/2,t) < 0
                Then b ← (a + b)/2
            Else
                a ← (a + b)/2
    End If
    Return a
End Def
For t ∈ (0,1) Do
    m ← FindRoot(0.000001,1,t)
End For
Plot m(t)
Return the graph of m(t)
End
```

3.4. 输入/输出示例

直接执行程序，我们将得到$m(t)$在$t > 0$时的函数图像，如图 3 所示。

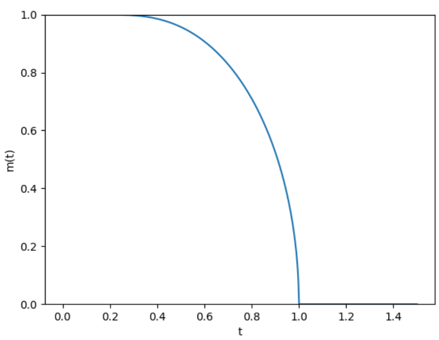

图 3 程序输出结果

我们同时给出在分析过程中绘制的，t的值取定时，$f(m)$的函数图像。如图

4、图 5、图 6、图 7 所示，分别为$t = -0.2, 0.4, 1.0, 1.4$时的图像。

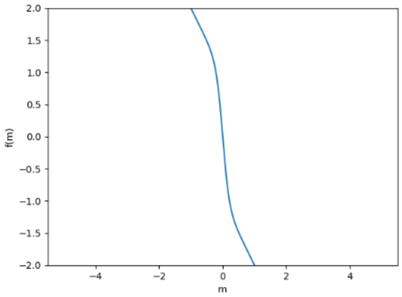

图 4 t=-0.2 一个平凡解

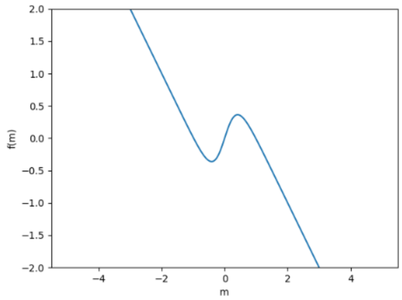

图 5 t=0.4 三个解

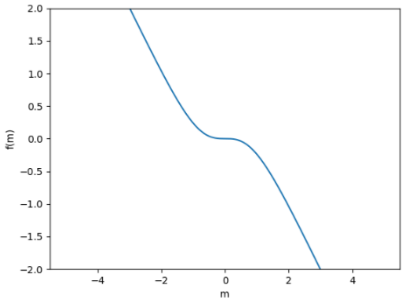

图 6 t=1 一个平凡解

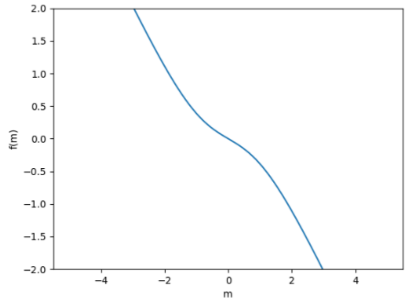

图 7 t=1.4 一个平凡解

3.5. 用户手册

- 本程序源文件为 Plotm_t.py，可执行程序为 Plotm_t.exe
- 在执行源程序之前，应当先安装 numpy 以及 matplotlib 库
- 本程序用于求解隐函数$m(t) = tanh(m(t)/t)$，绘制约化磁化强度与约化温度之间的关系（$t > 0$）
- 运行程序后，将直接输出$m(t) \sim t$函数图像