Homework 2

杨中义 物理学系 20307110314

2023.9.16

1. Problem 1 Finding roots

1.1. Problem Description

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

(i) Determine the two positive roots to 4 decimal places using the bisection method. Note: You first need to bracket each of the roots.

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

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

1.2. Code Description

本程序用三种方法确定了$x^3 - 5x + 3 = 0$.的两个非负根

(1) 二分法

注意到 $f(x) = x^3 - 5x + 3$ 两个正根的范围很好确定: $f(0) > 0, f(1) < 0, f(2) > 0$，因此两个正根分别位于$(0, 1), (1, 2)$两个区间之中，因此初始值分别设置为0,1,2。我们在每一次查找过程中考察区间中间，即$x = (a + b)\ /\ 2$的函数值$f(x)$，根据$f(x)$的符号确定方程的根的位置是处于左半区间还是右半区间，从而将搜索区域减小为原来的一半，不断缩小区间，直到找到符合精准要求的解。

(2) 牛顿法

将$f(x)$在x_0处展开：$f(x) = f(x_0) + f'(x)(x - x_0) + o\left(x^2\right)$, 对方程的根$x$，显然有$f(x) = 0$, 我们得到递推式：

$$x \to x' = x - \frac{f(x)}{f'(x)}$$

用该方法我们可以在平滑的函数零点附近慢慢迭代直至靠近零点，在上一个问题中我们已经将根精确到准确值的四位小数内，故认为附近的函数是单调且平滑的，可以使用牛顿法进一

步提高精确度。

(3)混合法

考虑到前两种方法遇到某些情况，无法有效处理，这里使用混合方法，逻辑如下图所示

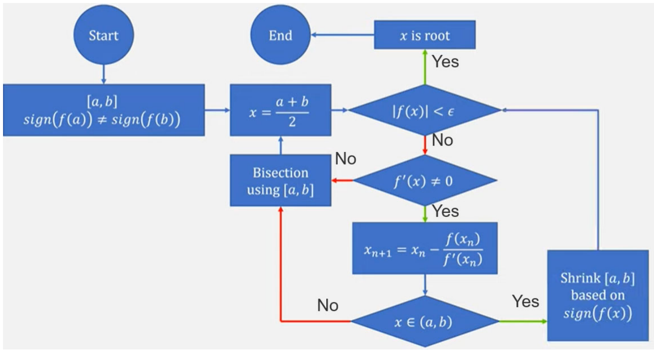


Figure 1: flow chart of hybrid method

程序所用到的源文件为/finding_roots/bisection_method.py , /finding_roots/newton-raphson_method.py , /finding_roots/hybrid_method.py

1.3. Pseudo Code

Algorithm 1.1 Finding 2 positive roots of the given equation using bisection method

Input: /

Output: 2 positive roots of the given equation,root1,root2

def bisection(a,b,c):

  while $b - a > c$:

    $x \leftarrow \frac{a+b}{2}$

    if $f(x) * f(a) < 0$:

      $b \leftarrow x$ $\Rightarrow [a, b] \rightarrow \left[a, \frac{a+b}{2}\right]$

    if $f(x) * f(b) < 0$:

      $a \leftarrow x$ $\Rightarrow [a, b] \rightarrow \left[\frac{a+b}{2}, a\right]$

    return x

root1 $\leftarrow$ bisection(0,1,1e-5) $\Rightarrow root\ bracket\ by\ 0, 1$

root2 $\leftarrow$ bisection(1,2,1e-5) $\Rightarrow root\ bracket\ by\ 0, 1$

Algorithm 1.2 Finding 2 positive roots of the given equation using Newton-Raphson method

Input: root1, root2 in Algorithm1.1

Output: 2 positive roots of the given equation,root1,root2 in 14 decimal places

1: def newrap(a,c):

2: $a* \leftarrow a$
$a \leftarrow a - f(a) \,/\, f'(a)$ $\Rightarrow a*\ represent\ a\ in\ last\ iteration$

3: while $abs(a - a*) > c$:

4: $a* \leftarrow a$
$a \leftarrow a - f(a) \,/\, f'(a)$

5: return x

6: root1 $\leftarrow$ newrap(x1,1e-5)

7: root2 $\leftarrow$ newrap(x2,1e-5) $\Rightarrow x1, x2\ are\ two\ roots\ from\ Alogorithm\ 1.1$

Algorithm 1.3 Finding 2 positive roots of the given equation using hybrid method

Input: /

Output: 2 positive roots of the given equation,root1,root2

def hybrid(a,b):

  $x \leftarrow \dfrac{a+b}{2}$

  while $b - a > c$:

    if $f'(x) \neq 0$:

      $temx \leftarrow x - f(x)\ /\ f'(x)$

      if $|x - temx| < c$:

        return $temx$

      if $temx \in [a,b]$:

        if $f(temx) * f(a) < 0$:

          $b \leftarrow temx$ $\Rightarrow [a,b] \rightarrow [a,temx]$

        elif $f(temx) * f(b) < 0$:

          $a \leftarrow temx$ $\Rightarrow [a,b] \rightarrow [temx,a]$

          $x \leftarrow temx$

      elif: $\Rightarrow temx \notin [a,b]$

        bisection(a,b)

    elif: $\Rightarrow f'(x) = 0$

      bisection(a,b)

  return

root1 $\leftarrow$ hybrid(0,1) $\Rightarrow root\ bracket\ by\ 0,1$

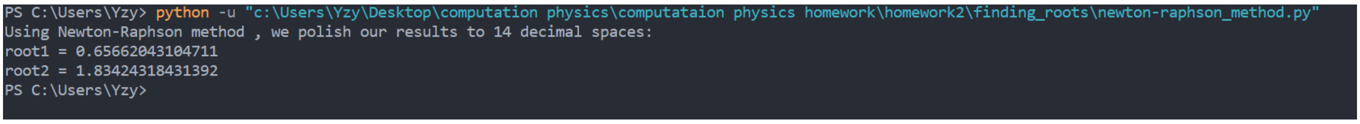
root2 $\leftarrow$ hybrid(1,2) $\Rightarrow root\ bracket\ by\ 0,1$

1.4. Testing case

三个程序的输出如下所示：

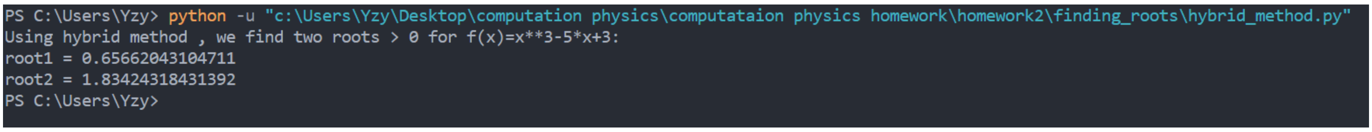
```
PS C:\Users\Yzy> python -u "c:\Users\Yzy\Desktop\computation physics\computataion physics homework\homework2\finding_roots\bisection_method.py"
Using bisecection method , we find 2 roots >0 for f(x)=x**3-5*x+3 :
root1 = 0.6566
root1 = 1.8342
PS C:\Users\Yzy>
```

Figure 2: bisection method



```
PS C:\Users\Yzy> python -u "c:\Users\Yzy\Desktop\computation physics\computataion physics homework\homework2\finding_roots\newton-raphson_method.py"
Using Newton-Raphson method , we polish our results to 14 decimal spaces:
root1 = 0.65662043104711
root2 = 1.83424318431392
PS C:\Users\Yzy>
```

Figure 3: Newton-Raphson method



```
PS C:\Users\Yzy> python -u "c:\Users\Yzy\Desktop\computation physics\computataion physics homework\homework2\finding_roots\hybrid_method.py"
Using hybrid method , we find two roots > 0 for f(x)=x**3-5*x+3:
root1 = 0.65662043104711
root2 = 1.83424318431392
PS C:\Users\Yzy>
```

Figure 4: hybrid method

2. Problem 2 Searching Minimum

2.1. Problem Description

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

2.2. Code Description

程序中使用最速下降法来搜索二元函数$g(x, y)$的最小值。迭代方法如下所示：

$$x \to x - a * \frac{\partial}{\partial x} g(x, y)$$
$$y \to y - a * \frac{\partial}{\partial y} g(x, y)$$

其中$a > 0$，代表迭代使用的步长，程序中默认初始化为$a = 0.01$。程序通过此次迭代与上一次迭代的x, y的差值来判断是否达到理想的精度范围。

程序所用到的源文件为/searching_minimum/searching_minimum.py

2.3. Pseudo Code

Algorithm 2 Searching Minimum of the given function using steepest descent method

Input: starting point (x0,y0)

Output: the minimal of the function in the whole space

1: $\begin{aligned} x &\leftarrow x0 - a*\frac{\partial}{\partial x}g(x,y) \\ y &\leftarrow y0 - a*\frac{\partial}{\partial y}g(x,y) \end{aligned}$ $\Rightarrow$ movement toward the minimal point

2: while $|x - x0| < c$ and $|y - y0| < c$:

3: $\quad\begin{aligned} x &\leftarrow x0 - a*\frac{\partial}{\partial x}g(x,y) \\ y &\leftarrow y0 - a*\frac{\partial}{\partial y}g(x,y) \end{aligned}$

4: $g(x,y)$ $\Rightarrow$ the minimum of the $g(x,y)$

2.4. Testing case

从不同的起点开始搜索，得到同样的结果

```
PS C:\Users\Yzy> python -u "c:\Users\Yzy\Desktop\computation physics\computataion physics homework\homework2\searching_minimum\searching_minimum.py"
Input your starting point to find the minimum:
x=0
y=0
Minimum of the funciton f(x,y) is -2.0
PS C:\Users\Yzy>
```

Figure 5: searching minimum from(0,0)

```
PS C:\Users\Yzy> python -u "c:\Users\Yzy\Desktop\computation physics\computataion physics homework\homework2\searching_minimum\searching_minimum.py"
Input your starting point to find the minimum:
x=100
y=250
Minimum of the funciton f(x,y) is -2.0
```

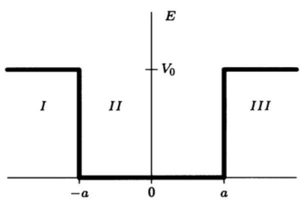
Figure 6: searching minimum from(100,200)

3. Problem 3 Finite Square Well Potential

3.1. Problem Description

Electron in the finite square-well potential is:

$$V(x) = \begin{cases} V_0 & x \le -a \quad \text{Region I} \\ 0 & -a < x < a \quad \text{Region II} \\ V_0 & x \ge a \quad \text{Region III} \end{cases}$$



Solve the energy E when$V_0 = 10eV, a = 0.2nm$

3.2. Code Description

根据量子力学的知识，从 Shrodinger 方程：$-\frac{\hbar^2}{2m}\frac{d^2\psi(x)}{dx} + V(x)\psi(x) = E\psi(x)$, 通过对$V(x)$的分段讨论,以及边界条件的应用。我们可以得到在波函数分别为奇数、偶数时能量满足的方程：

$$Even\ states:\ \alpha\ sin(\alpha a) = \beta\ cos(\alpha a) \tag{1}$$
$$Odd\ states:\ \alpha\ cos(\alpha a) = -\beta\ sin(\alpha a) \tag{2}$$

其中$\beta = \sqrt{2m(V_0 - E)}\ /\ \hbar$ ，$\alpha = \sqrt{2mE}\ /\ \hbar$。若我们令$x = \alpha a, x_0 = \sqrt{2mV_0}\ /\ \hbar$, 则(1)、(2)化为：

$$f(x) = xsin(x) - \sqrt{x_0^2 - x^2}cos(x) \tag{3}$$
$$g(x) = xcos(x) + \sqrt{x_0^2 - x^2}sin(x) \tag{4}$$

因此只需要求解两个函数的零点即可。

根据(Introduction to Quantum Mechanics (David J. Griffiths) pp80)上对这个问题的讨论以及拓展，我们可以得到如下结论：

1. 对于偶函数，(3)至少有一个解，并且解均位于$\left(k\pi, k\pi + \frac{1}{2}\pi\right) k \in N$, 每个区间最多只有一个解。

2. 对于奇函数，(4)有解的充要条件为$x_0 > \frac{\pi}{2}$,并且解均位于$\left(-\frac{\pi}{2} + k\pi, k\pi\right) k \in Z*$, 每个区间最多只有一个解。

因此利用这两个性质，我们不仅可以确定给定参数条件下解的个数：$\lfloor x_0\ /\ \pi \rfloor + 1$，还可以确定相应解的区间。有了解所在的区间我们就可以利用/finding_roots/hybrid_method.py文件中的混合法求出相应的根

本程序的源代码文件为/finite square well potential/finite_square_well_potential.py

3.3. Pseudo Code

Algorithm 3 Solving energy levels using hybrid method

Input: functiontype(odd/even) , the number of the level:k

Output: energy required

if functiontype = even:

$n \leftarrow \lfloor x_0 / \pi \rfloor + 1$ $\Rightarrow n: the\ total\ numbers\ of\ the\ energy\ level$

$a_0 \leftarrow (k-1)\pi,\ \ b_0 \leftarrow (k-1/2)\pi$ $\Rightarrow k: the\ number\ of\ the\ level\ wanted$; $a_0, b_0: the\ initial\ bracket$

$x \leftarrow hybrate(a_0, b_0), E \leftarrow \dfrac{x\hbar^2}{2ma^2}$ $\Rightarrow energy\ of\ the\ parity\ even\ function$

if functiontype = odd:

$n \leftarrow \lfloor x_0 / \pi \rfloor + 1$

$a_0 \leftarrow (k-1/2)\pi,\ \ b_0 \leftarrow k\pi$

$x \leftarrow hybrate(a_0, b_0), E \leftarrow \dfrac{x\hbar^2}{2ma^2}$ $\Rightarrow energy\ of\ the\ parity\ odd\ function$

3.4. Testing case

对于不同的质量的选择会导致不同的解的个数，对于函数奇偶性的选择、能级序号的选择也会有不同的结果。下面是几个测试样例：

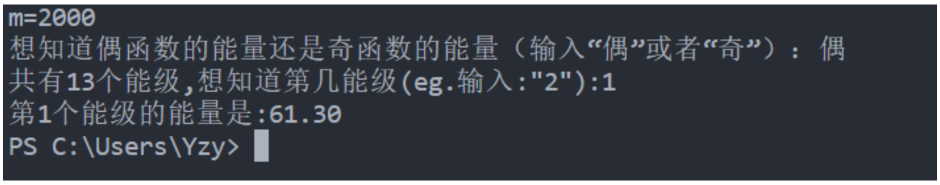


Figure 7: m=2000,偶函数,第1能级

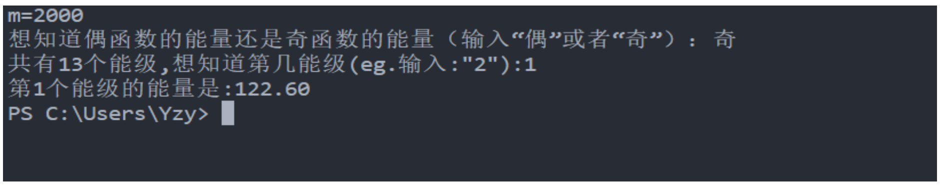


Figure 8: m=2000,奇函数,第一能级

```
m=100
想知道偶函数的能量还是奇函数的能量（输入“偶”或者“奇”）： 偶
共有3个能级,想知道第几能级(eg.输入:"2"):3
第3个能级的能量是:9.28
PS C:\Users\Yzy>
```

Figure 9: m=100,偶函数,第三能级

```
m=1
想知道偶函数的能量还是奇函数的能量（输入“偶”或者“奇”）： 奇
能量太低不足以形成定态！
PS C:\Users\Yzy>
```

Figure 10: m=1,奇函数,不能形成定态