CompPhys Assignment 05

李尚坤　物理学系　20307130215

1 Richardson Extrapolation

1.1 题目描述

Compute the derivative of $f(x) = \sin x$ at $x = \pi/3$ using the Richardson extrapolation algorithm. Start with $h = 1$ and find the number of rows in the Richardson table required to estimate the derivative with six significant decimal digits. Output the Richardson table.

1.2 解决方案描述

对于 Richardson 方法而言，最终要的是计算各项 $D(m,n)$，它们之间满足的关系为：

$$D(n,0) = \varphi\left(\frac{h}{2^n}\right) = \frac{f\left(x+\frac{h}{2^n}\right) - f\left(x+\frac{h}{2^n}\right)}{2\times\frac{h}{2^n}} \tag{1.1}$$

$$D(n,m) = D(n,m-1) + \frac{1}{4^m-1}\left[D(n,m-1) - D(n-1,m-1)\right] \tag{1.2}$$

在本题中，我们采用按行来构建 Richardson table，其具体步骤如下：

1. 首先由公式：$D(0,0) = \frac{f(x+h)-f(x-h)}{2h}$ 计算出 $D(0,0)$, 这相当于一个 1×1 的矩阵；

2. 先在 $D(0,0)$ 下方添加一个元素 $D(1,0)$，再由 (2) 式计算出 $D(1,1)$，于是我们就得到了如下左侧的 2×2 矩阵：

$$\begin{bmatrix} D(0,0) & 0 \\ D(1,0) & D(1,1) \end{bmatrix} \to \begin{bmatrix} D(0,0) & 0 & 0 \\ D(1,0) & D(1,1) & 0 \\ D(2,0) & D(2,1) & D(2,2) \end{bmatrix}$$

3. 对于上面左侧矩阵，我们再在 $D(1,0)$ 下方添加一个元素 $D(2,0)$，这样我们可以由 $D(1,0)$ 与 $D(2,0)$ 计算得到 $D(2,1)$，再由 $D(1,1)$ 与 $D(2,1)$ 计算得到 $D(2,2)$，这样我们就得到了一个 3×3 矩阵，如上方右侧矩阵所示；

4. 按照这上述步骤可以不断将矩阵扩充下去，其中对于已求出的 $n+1$ 阶方阵，元素 $D(n,n)$ 是精度最高的结果；

5. 我们每扩充完一行之后，就将元素 $D(n+1,n+1)$ 与元素 $D(n,n)$ 进行比较，如果二者之差满足精度要求，就停止构造矩阵，然后输出结果。

1.3 伪代码

Algorithm 1: Richardson Extrapolation

Input: h and the precision eps

Output: The derivative of $f(x)$ at $x = x_0$, the number of rows and Richardson table

$D(0,0) \leftarrow \frac{f(x_0+h)-f(x_0-h)}{2h}$
$i \leftarrow 0$
do
  $i \leftarrow i+1$
  $D(i,0) \leftarrow \frac{f(x_0+(\frac{h}{2^i}))-f(x_0-(\frac{h}{2^i}))}{2\times\frac{h}{2^i}}$
  for $j \leftarrow 1$ **to** i **do**
    $D(i,j) \leftarrow D(i,j-1) + \frac{1}{4^j-1}\left[D(i,j-1) - D(i-1,j-1)\right]$
  end
while $ABS(D(i,i) - D(i-1,i-1)) > eps$;
return $D(i,i), i, D$

1.4 输入输出示例

本题中我们要求函数 $f(x) = \sin x$ 在点 $x_0 = \frac{\pi}{3}$ 处的导数值。我们给定的 h 的初始值为 $h = 1$，精度定为 $eps = 10^{-6}$，输出结果如下：

1. **The number of rows** in the Richardson table required to estimate the derivative with six significant decimal digits is **5**.
2. **The derivative** of $f(x)$ at $x = \pi/3$ is 0.500000.
3. **The Richardson table** is

$$\begin{bmatrix} 0.4207354924 & & & & \\ 0.4794255386 & 0.4989888873 & & & \\ 0.4948079185 & 0.4999353785 & 0.4999984779 & & \\ 0.4986989335 & 0.4999959386 & 0.4999999759 & 0.4999999997 & \\ 0.4996745427 & 0.4999997458 & 0.4999999996 & 0.5000000000 & 0.5000000000 \end{bmatrix}$$

程序在终端运行时的输出结果如下：

```
The number of rows in the Richardson table required to estimate the derivative with six significant decimal digits is 5

The derivative of f(x) at x = π/3 is 0.500000

The Richardson table is

[0.4207354924039484]
[0.4794255386042031, 0.49898888733762137]
[0.4948079185090457, 0.49993537847732655, 0.49999847788664026]
[0.4986989335409113, 0.4999959385515332, 0.4999999758898136, 0.49999999966764175]
[0.4996745427390419, 0.4999997458050854, 0.4999999996219889, 0.49999999999869005, 0.4999999999999883]
```

图 1: Richardson Extrapolation

1.5 用户手册

1. 本程序的源程序为 Richardson.py

2. 本程序利用 Richardson Extrapolation 方法进行导数计算

3. 运行程序后，将在终端依次输出：为达到要求有效数字位数所需计算行数，符合有效数字要求导数结果以及 Richardson table

2 Simpson's Rule

2.1 题目描述

Radial wave function of the 3s orbital is:

$$R_{3s} = \frac{1}{9\sqrt{3}} \times (6 - 6\rho + \rho^2) \times Z^{\frac{3}{2}} \times e^{-\frac{\rho}{2}} \tag{2.3}$$

· r = radius expressed in atomic units (1 Bohr radius = 52.9 pm)

· e = 2.71828 approximately

· Z = effective nuclear charge for that orbital in that atom

· $\rho = 2Zr/n$ where n is the principal quantum number (3 for the 3s orbital)

Compute $\int_0^{40} |R_{3s}|^2 r^2 dr$ for Si atom ($Z = 14$) with Simpson's rule using two different radial grids:

(1) Equal spacing grids: $r[i] = (i-1)h$; $i = 1,\cdots, N$ (try different N)

(2) A nonuniform integration grid, more finely spaced at small r than at large r: $r[i] = r_0(e^{t[i]} - 1)$; $t[i] = (i-1)h$; $i = 1,\cdots, N$ (One typically choose $r_0 = 0.0005$ a.u., try different N)

(3) Find out which one is more efficient, and discuss the reason.

2.2 解决方案描述

由 Simpson's rule 可知，在区间 (a, b) 上，积分公式为：

$$\int_a^b f(x)dx = \frac{b-a}{6}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right] \tag{2.4}$$

当整个积分区间被分成偶数个区间时，整个区间的积分值可近似为：

$$\int_{x_1}^{x_n} f(x)dx \approx \sum_{i=1,3,5}^{n-2} \frac{x_{i+1} - x_i}{3}\left[f(x_i) + 4f(x_{i+1}) + f(x_{i+2})\right] \tag{2.5}$$

因此只要将区间分为偶数个，带入相应的积分函数进行计算即可。

题目中给出了两种不同的变量代换的方式，我们将 $n = 3, Z = 14$ 带入，对于 (1) 中的代换方式，其对应的积分公式为：

$$\int_0^{40} \frac{1}{81 \times 3}\left(6 - 56r + \left(\frac{28r}{3}\right)^2\right) \times 14^{\frac{3}{2}} \times e^{-\frac{14r}{3}} r^2 dr \tag{2.6}$$

对于 (2) 中的代换方式，其对应的积分公式为：

$$\int_0^{ln(1+\frac{40}{r_0})} \frac{14^{\frac{3}{2}} r_0}{81\times 3}\left(6-56r_0(e^t-1)+\left(\frac{28r}{3}r_0(e^t-1)\right)^2\right)\times e^{-\frac{14r_0(e^t-1)}{3}}(r_0(e^t-1))^2e^t dt \tag{2.7}$$

因此，只需要将上面两个积分公式中的积分函数分别定义为 $f_1(r)$ 与 $f_2(t)$，再分别带入 (2.5) 式计算即可。

2.3 伪代码

Algorithm 2: Simpson's Rule

Input: The number of points N

Output: The integration of $f(x)$

for $i \leftarrow 0$ **to** $N-2$ ***by*** 2 **do**
  $integration \leftarrow integration + \frac{x_{i+1}-x_i}{3}\left[f(x_i)+4f(x_{i+1})+f(x_{i+2})\right]$
end
return *integration*

2.4 输入/输出示例

在本题中，对于积分函数 $f_1(r)$ 与 $f_2(t)$，我们将其分别划分为 N_1-1 与 N_2-1 个区间，输出结果如下：

N_1	N_2	$\int f_1(r)dr$	$\int f_2(t)dt$
131	131	0.075045	0.076360
101	101	0.071754	0.076360
71	71	0.069530	0.076360
51	51	0.090188	0.076360
31	31	0.119175	0.076788

表 1: 输入输出结果

程序在终端运行时的输出结果如下：

```
Plese input two numbers of points:(must be odd number)
101 101
intergra1= 0.071753873467746
intergra2= 0.0763603548321187
```

图 2: Simpson's Rule

由 Simpson's rule 的误差公式可知：$error = \frac{(x_{i+1}-x_i)^5}{90}\frac{d^4f(x_j)}{dx^4}$，当取 $N_1 = N_2 = 101$ 时，可大致估算得到 $error(\int f_1(r)dr) \sim 10^{-4}$，$error(\int f_2(t)dt) \sim 10^{-7}$。因此两个积分在取相同 N 的值时，积分 $\int f_2(t)dt$ 的精度更高。

由上表也可以看出，在我们所取的 N 的范围中，随着 N 的下降 $\int f_1(r)dr$ 的积分值误差急剧增加，而 $\int f_2(t)dt$ 的积分值基本稳定。也就是说，要达到相同的计算精度，采用计算 $\int f_2(t)dt$ 可以计算更少的次数。

下面我们简要分析一下原因。我们分别作出 $f_1(r)$ 在区间 $[0, 40]$，$f_2(t)$ 在区间 $[0, ln(1 + \frac{40}{r_0})]$ 内的函数图像：

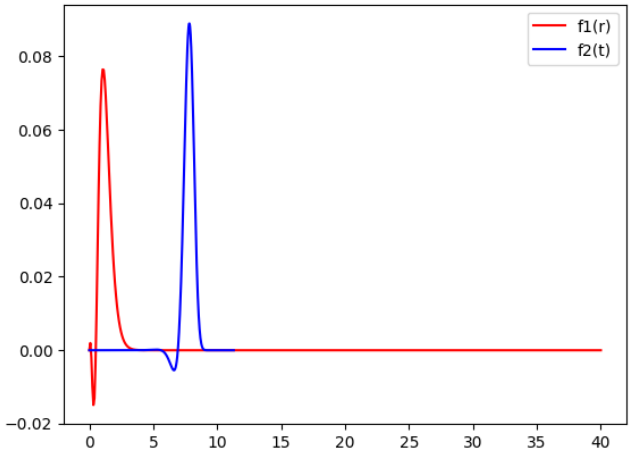

图 3: 积分函数图像

可以看出，积分函数 $f_1(r)$ 对应的积分区间 $[0, 40]$ 非常长，但主要对积分结果有贡献的函数区间只占很小一部分，大致为 $[0, 5]$。而积分函数 $f_2(t)$ 的积分区间较短，并且“有效”的积分区间占比较大。因此当我们取相同的 N 时，我们将区间分成了 $N - 1$ 个间隔，其中 r 的间隔较大，t 的间隔较小，因此 $\int f_2(t)dt$ 的精度更高。

其实，积分变量代换 $r = r_0(e^t - 1)$ 相当于将积分区间进行了一定的压缩，r_0 用于表征压缩的程度，如果 r_0 越大，就说明压缩的程度越大，取相同 N 时得到的积分结果也就更准确。

2.5 用户手册

1. 本程序的源程序为 Simpson.py
2. 在执行源程序之前，应当先安装 numpy 库
3. 本程序分别利用 Simpson's Rule 计算一个积分在两种变量代换下的结果
4. 运行程序后，在终端将输出积分值 $integra1$ 与 $integra2$