Homework No.5

Ruqi Feng

November 2022

1 Richardson Extrapolation

1.1 Description

Compute the derivative of f (x) = sin x at x = /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 Solution

使用了 Richardson extrapolation algorithm。因为误差可以用一个多项式 $\sum_i a_i h^i$ 表示，所以我们可以取不同的步长 h 来确定各个系数 a_i 的大小，从而准确估计 $f'(x)$。对于某个函数的中心差分

$$\phi(h) = \frac{f(x+h) - f(x-h)}{2h} \tag{1}$$

可以取 $h \leftarrow h/2^n$，其中 $n = 0, 1, ..., n$。计算不同步长的 n 个导数，即 $D(1,0), D(2,0), ..., D(n,0)$，并利用公式

$$D(n,m) = D(n,m-1) + \frac{1}{4^m - 1}[D(n,m-1) - D(n-1,m-1)] \tag{2}$$

可以递推计算出 $D(n,n)$。这就是 Richadson extrapolation 给出的结果

Richadson 方法的意义之一在于，在同样的最小步长 $h/2^n$ 下，它比中心差分的精度更好。

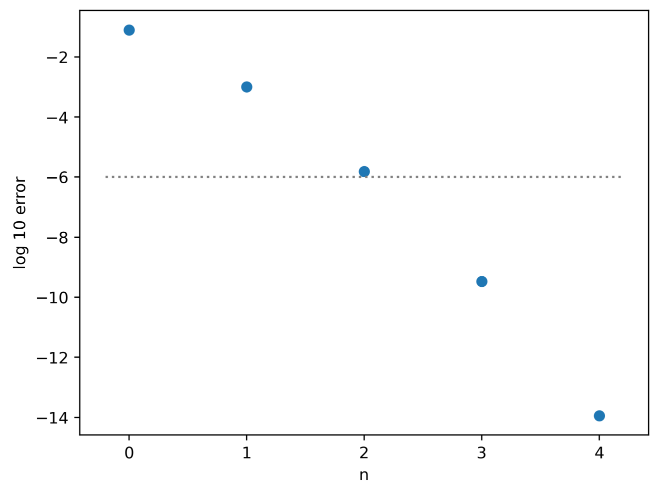


图 1: 将不同的 n 对应的误差绝对值 (log10) 在图中画出，灰色虚线表示精度达到 6 位有效数字的位置。

1.3 Input and Outputs

对 $\sin x$ 在 $x = 0.5$ 处进行 Richardson extrapolation，取 $h = 1$，在不同的 n 下进行计算并将结果和理论值进行比较，部分结果如下：

n	result
1	0.42073549240394825
2	0.49898888733762137
3	0.4999984778866406
4	0.499999999667641
5	0.49999999999998856

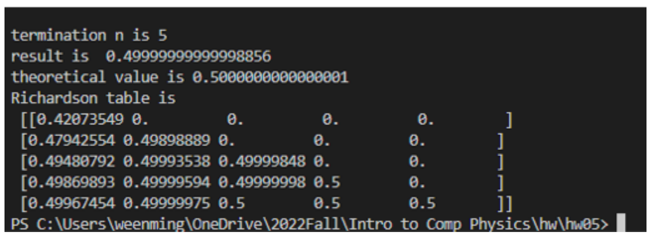
在图中画出结果，也可以观察到随着 n 增加 (算法的时间复杂度约为 $O(n^2)$)，结果的精度以超过指数的速度上升。这说明 Richardson extrapolation 十分适合于精度要求较高的差分计算。

在上表和图2中都可以看出，最小的达到 6 位有效数字精度的 n 是 4. 前 5×5 的 Richardson table 如下：

$$
\begin{pmatrix}
0.42073549 & 0. & 0. & 0. & 0. \\
0.47942554 & 0.49898889 & 0. & 0. & 0. \\
0.49480792 & 0.49993538 & 0.49999848 & 0. & 0. \\
0.49869893 & 0.49999594 & 0.49999998 & 0.5 & 0. \\
0.49967454 & 0.49999975 & 0.5 & 0.5 & 0.5
\end{pmatrix}
$$

出于排版考虑，这里 $D_{4,4}$ 标记为 0.5，但实际值是 0.499999999667641。其他显示 0.5 的值也以最前面的表为准。

程序使用了 numpy, matplotlib 和 scipy 库。在安装了依赖库的环境中运行 interpolation.py 即可。下面是程序运行的截图。



```
termination n is 5
result is  0.49999999999998856
theoretical value is 0.5000000000000001
Richardson table is
 [[0.42073549 0.         0.         0.         0.        ]
 [0.47942554 0.49898889 0.         0.         0.        ]
 [0.49480792 0.49993538 0.49999848 0.         0.        ]
 [0.49869893 0.49999594 0.49999998 0.5        0.        ]
 [0.49967454 0.49999975 0.5        0.5        0.5       ]]
PS C:\Users\weenming\OneDrive\2022Fall\Intro to Comp Physics\hw\hw05>
```

图 2: 程序运行截图

1.4 Pseudocode

Algorithm 1 Newton Interpolation

Input: n and n CD values $D(1,0), D(2,0), ..., D(n,0)$ where $D(i,0) = \phi(h/2^i)$, ϕ defined as in Eq.1

Output: Derivative $f'(x_0)$

$D[:,0] \leftarrow$ input $D(i,0)$s
for $i \leftarrow 1,2,3,...,n-1$ **do**
  $t \leftarrow 1$
  for $j \leftarrow 0,1,2,...,n-i$ **do**
    $D[j,i] \leftarrow D[j,i-1] + \frac{1}{t-1}(D[j,i-1] - D[j-1,i-1])$
    $t \leftarrow t \times 4$
  end for
end for
return $D[n,n]$

2 Simpson's Integration

2.1 Description

Compute $\int_0^{40} \|R_{3s}\|^2 r^2 dr$ for Si atom (Z=14) with Simpson' s rule using two different radial grids: Equal spacing grids: $r[i] = (i-1)h$; $i = 1, ..., N$(try different N) A nonuniform integration grid, more finely spaced at small r than at large r: $r[i] = r0\ (e^{t[i]} - 1)$; $t[i] = (i-1)h$; $i = 1, ..., N$(One typically choose $r_0 = 0.0005$a.u., try different N). Find out which one is more efficient, and discuss the reason.

2.2 Solution

Simpson 算法是一种简单的算法，但是在局域却有 $O(h^3)$ 的精度。它在区间上取 3 个点，进行二次 Lagrange 插值，将插值函数进行积分，可得

$$I = \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) \tag{3}$$

由于 Lagrange 插值的误差为

$$\varepsilon = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^{n} (x - x_i) \tag{4}$$

在 Simpson 法中为三次函数。若采样点选择在区间中点左右对称分布，在区间中进行积分就能得到积分的三阶误差结果为 0。这也就是说，Simpson 法的局域误差可以达到 h^4(正比于 $f^{(4)}(x)$)，而花费的计算并不多。

在题目要求下，我们等间隔和不等间隔地进行采样，其中不等间距采样的方式是选取 N 后，确定 $h = \log(\frac{x_{max}}{r_0} + 1)/N$，为了在区间内获得较高精度，在每个区间的中心再取 1 个点，区间长度于是为 $h/2$。

被积函数是

$$\psi = \frac{2}{3\sqrt{3}}(1 - \frac{2x}{3} + \frac{2x^2}{27})e^{-\frac{x}{3}} \tag{5}$$

在 $[0, 40]$ 内进行积分。

2.3 Input and Outputs

在不同 N 下进行积分，结果和理论值 1 之差如图3所示。在 N 较小 (< 约 25) 时，均匀采样效果较好 (此时数值积分结果也较差，所以意义不太

大)；但是 N 较大时，不均匀采样效果更好。这和被积函数的性质有关。从图4中可以看出被积函数在 r 较小时变化较剧烈，而 r 约为 10 20 时函数值较大。

在指数采样中，如果采样点过少，在函数值较大的区域采样不多，容易导致较大的误差；但采样点足够多后，由于 Simpson 法的误差正比于四阶导数，主要误差由 r 较小时函数的波动引起，此时在 r 较小时采样较密的方法就能获得更大的精度。总之，在一般的模拟精度下，非均匀采样具有更高的效率

程序使用了 numpy 和 matplotlib 库。在安装了依赖库的环境下运行 Simpson.py 即可。

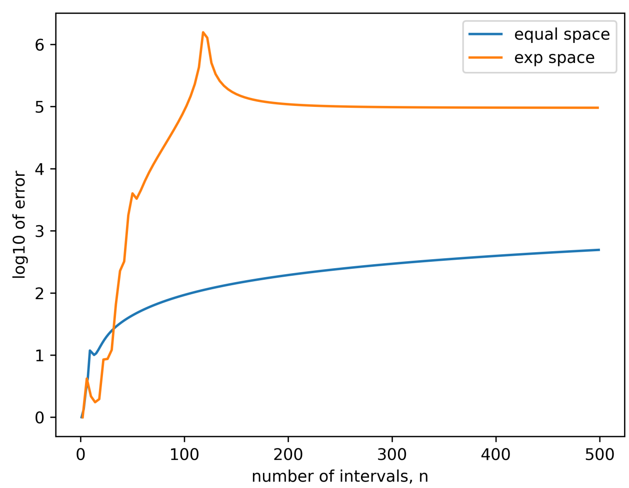


图 3: 两种采样方法的误差随 N 的变化

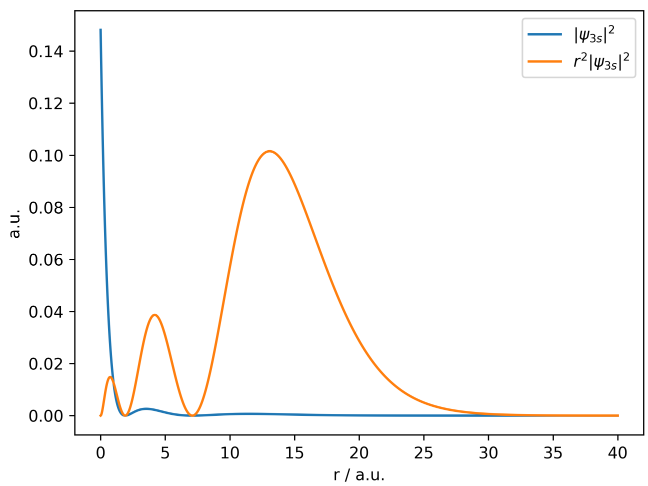


图 4: 图中，蓝色线是径向波函数模方，但橙色线才是被积函数 (相差 r^2)。

2.4 Pseudocode

Algorithm 2 linear fit

Input: sampling points: array pts, function , integration range x_1, x_2

Output: Integration value I

$I \leftarrow 0$

for i in range(len(pts)-2) **do**

$\quad I \leftarrow I + \frac{pts[i+1]-pts[i]}{6}((pts[i]) + 4f(pts[i+1]) + f(pts[i+2]))$

end for

Return: I