计算物理作业 5

杨远青 22300190015 CompPhys 24

2024 年 10 月 30 日

喜闻徐夫子体恤民情!

1 题目 1：五点公式

1.1 题目描述

Derive the five-point formula for the second-order derivative.

1.2 解答

利用函数 f 在点 $i \pm k$ 处的泰勒展开，得到以下差分表达式：

$$
\begin{aligned}
f_{i+2} &= f_i + 2hf_i' + 2h^2 f_i'' + \frac{4h^3}{3} f_i''' + \frac{2h^4}{3} f_i^{(4)} + \frac{8h^5}{15} f_i^{(5)} + \mathcal{O}(h^6), \\
f_{i+1} &= f_i + hf_i' + \frac{h^2}{2} f_i'' + \frac{h^3}{6} f_i''' + \frac{h^4}{24} f_i^{(4)} + \frac{h^5}{120} f_i^{(5)} + \mathcal{O}(h^6), \\
f_{i-1} &= f_i - hf_i' + \frac{h^2}{2} f_i'' - \frac{h^3}{6} f_i''' + \frac{h^4}{24} f_i^{(4)} - \frac{h^5}{120} f_i^{(5)} + \mathcal{O}(h^6), \\
f_{i-2} &= f_i - 2hf_i' + 2h^2 f_i'' - \frac{4h^3}{3} f_i''' + \frac{2h^4}{3} f_i^{(4)} - \frac{8h^5}{15} f_i^{(5)} + \mathcal{O}(h^6).
\end{aligned}
$$

目标是构造一个线性组合以消除一阶导数 f_i'、三阶导数 f_i''' 及五阶导数 $f_i^{(5)}$，不妨设：

$$
Af_{i+2} + Bf_{i+1} + Cf_i + Df_{i-1} + Ef_{i-2} = Kf_i'' + \mathcal{O}(h^6),
$$

通过匹配各阶 h 的系数，可以构建方程组，观察各系数，不妨设 $K = 12$ 并约分，改写为增广矩阵形式，并使用我们在 `Assignment_3/Problem_2` 中实现的高斯消元法解得（不出意外是行满秩的，有重复约束条件）

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & | & 0 \\
2 & 1 & 0 & -1 & -2 & | & 0 \\
4 & 1 & 0 & 1 & 4 & | & 24 \\
8 & 1 & 0 & -1 & -8 & | & 0 \\
16 & 1 & 0 & 1 & 16 & | & 0 \\
64 & 1 & 0 & -1 & -64 & | & 0
\end{pmatrix}
\rightarrow
\begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix}
=
\begin{pmatrix} -1 \\ 16 \\ -30 \\ 16 \\ -1 \end{pmatrix}
$$

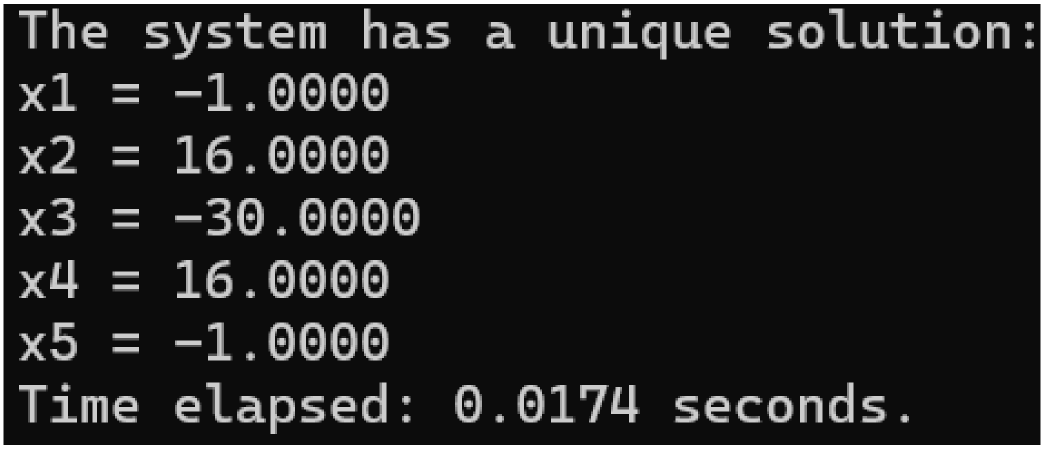
```
The system has a unique solution:
x1 = -1.0000
x2 = 16.0000
x3 = -30.0000
x4 = 16.0000
x5 = -1.0000
Time elapsed: 0.0174 seconds.
```

图 1: 运行结果

因此，求二阶差分的五点公式为：

$$-f_{i+2}+16f_{i+1}-30f_i+16f_{i-1}-f_{i-2}=12h^2f_i''+\mathcal{O}(h^6),$$

即，

$$\boxed{f_i''=\frac{-f_{i+2}+16f_{i+1}-30f_i+16f_{i-1}-f_{i-2}}{12h^2}+\mathcal{O}(h^4)}$$

~~2 题目 2：Romberg 积分~~

2.1 题目描述

Consider the function $f(x)=e^{-x^2}$ on the interval $[0,1]$. Use at least four layers of Romberg integration to compute the integral of $f(x)$ over $[0,1]$ and estimate the result's precision.

浪涌，即将被 ddl 的海洋淹没，此拓展题无暇细究

2 题目 2：波函数 Gauss 积分

2.1 题目描述

The radial wave function of the 3s orbital is given by:

$$R_{3s}(r)=\frac{1}{9\sqrt{3}}\times\left(6-6\rho+\rho^2\right)\times Z^{3/2}\times e^{-\rho/2},$$

where:

- r: radius expressed in atomic units (1 Bohr radius = 52.9 pm),
- $e\approx 2.71828$,
- Z: effective nuclear charge for that atom,
- $\rho=\frac{2Zr}{n}$, where n is the principal quantum number (3 for the 3s orbital).

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

(1) **Equal spacing grids**:

$$r[i] = (i-1)h, \quad i = 1, \ldots, N$$

Try different values of N.

(2) **Non-uniform integration grid**: more finely spaced at small r than at large r:

$$r[i] = r_0(e^{t[i]} - 1), \quad t[i] = (i-1)h, \quad i = 1, \ldots, N$$

Typically, choose $r_0 = 0.0005$ a.u. (1 a.u. = 1 Bohr radius).

Discuss the efficiency of each approach and explain the reasons.

2.2 程序描述

待积函数为电子径向分布函数，理论结果应该十分接近全空间积分值 1. 使用 Mathematica® 可以计算出不同积分终点 r_m 的定积分值

$$I(r_m) = 1 - \frac{1}{6561} e^{-\frac{28r_m}{3}} \left(6561 + 28r_m \left(2187 + 14r_m \left(729 + 392r_m^2 \left(81 + 28r_m \left(-9 + 14r_m\right)\right)\right)\right)\right)$$

对于本题终点 $r_m = 40$ 的积分值为

$$I(40) = 1 - \frac{1}{6561} e^{-\frac{1120}{3}} \cdot 242794431087524801 \approx 1 - 2.7 \times 10^{-149}$$

积分步骤与结果详见 `src/theoretical.nb`。原始的 Simpson 积分法源自等距节点的 Newton–Cotes 公式，因此在第 2 问使用非均匀节点计算时，需要给积分核 $|R_{3s}|^2 \, r^2$ 乘上换元因子 $dr = r_0 \cdot e^t$，本质是转换成了新的定积分 $g(t)$ 在等距节点 t 上的 Simpson 积分。

源代码在 `src/simpson.py` 中，其中的 `integrand` 函数即为原始积分核，`simpson_rule` 为基于给定点与函数值的 Simpson 积分法的实现。主程序中，在 $N = [3, 5, 7, 9, 11, 21, 51, 101, 201, 501, 1001, 10001]$ 分别计算了两种节点下的积分值。实际上在实现 `simpson_rule` 的时候，对传入的偶数节点数进行了舍尾处理，增强了程序的健壮性。在 `src` 中，运行`python -u simpson.py` 即可得到结果，需要安装 `numpy` 与 `matplotlib` 库。在随后的结果示例中，我们将看到，非均匀节点的积分值更快收敛于理论值。其根源在于坐标变换之后，采集点更密集地分布在积分核的峰值区域，避免了在积分核值较小的区域上的采样浪费。为了直观展现这一点，主程序选取了 $N = [11, 51, 201, 1001]$ 进行点分布绘制，详见结果示例。

2.3 伪代码

Algorithm 1: Simpson's Rule Integration

Input: r(array, shape $= N$) f(array, shape $= N$)

Output: integral(float)

$N \leftarrow \mathrm{len}(r)$; `// Number of grid points`

if $N < 3$ **then**

  return `Error: Simpson's rule requires at least 3 points`

end

if $N\%2 == 0$ **then**

  $r \leftarrow r[:-1]$; $f \leftarrow f[:-1]$; $N \leftarrow N-1$; `// Ensure an odd number of points`

end

$h \leftarrow \frac{r[N-1]-r[0]}{N-1}$;

$S \leftarrow f[0] + f[-1]$; `// Notice index from 0 in python`

$S \leftarrow S + 4 \times \sum f[1:-1:2]$; `// Notice range in python excludes the last element`

$S \leftarrow S + 2 \times \sum f[2:-2:2]$; `// Add 2 times sum of even-indexed terms`

integral $\leftarrow \frac{h}{3} \times S$

return integral

2.4 结果示例

```
Equal Spacing Grid Results:
N = 3: Integral = 0.000000
N = 5: Integral = 0.000000
N = 7: Integral = 0.000000
N = 9: Integral = 0.000000
N = 11: Integral = 0.000000
N = 21: Integral = 0.055361
N = 51: Integral = 1.366417
N = 101: Integral = 1.006313
N = 201: Integral = 0.961099
N = 501: Integral = 1.002767
N = 1001: Integral = 1.000698
N = 10001: Integral = 1.000000
```

```
Expotential Spacing Grid Results:
N = 3: Integral = 0.001675
N = 5: Integral = 0.018593
N = 7: Integral = 1.658037
N = 9: Integral = 0.160943
N = 11: Integral = 1.163772
N = 21: Integral = 1.015675
N = 51: Integral = 0.999801
N = 101: Integral = 1.000000
N = 201: Integral = 1.000000
N = 501: Integral = 1.000000
N = 1001: Integral = 1.000000
N = 10001: Integral = 1.000000
```

图 2: 两种节点比较

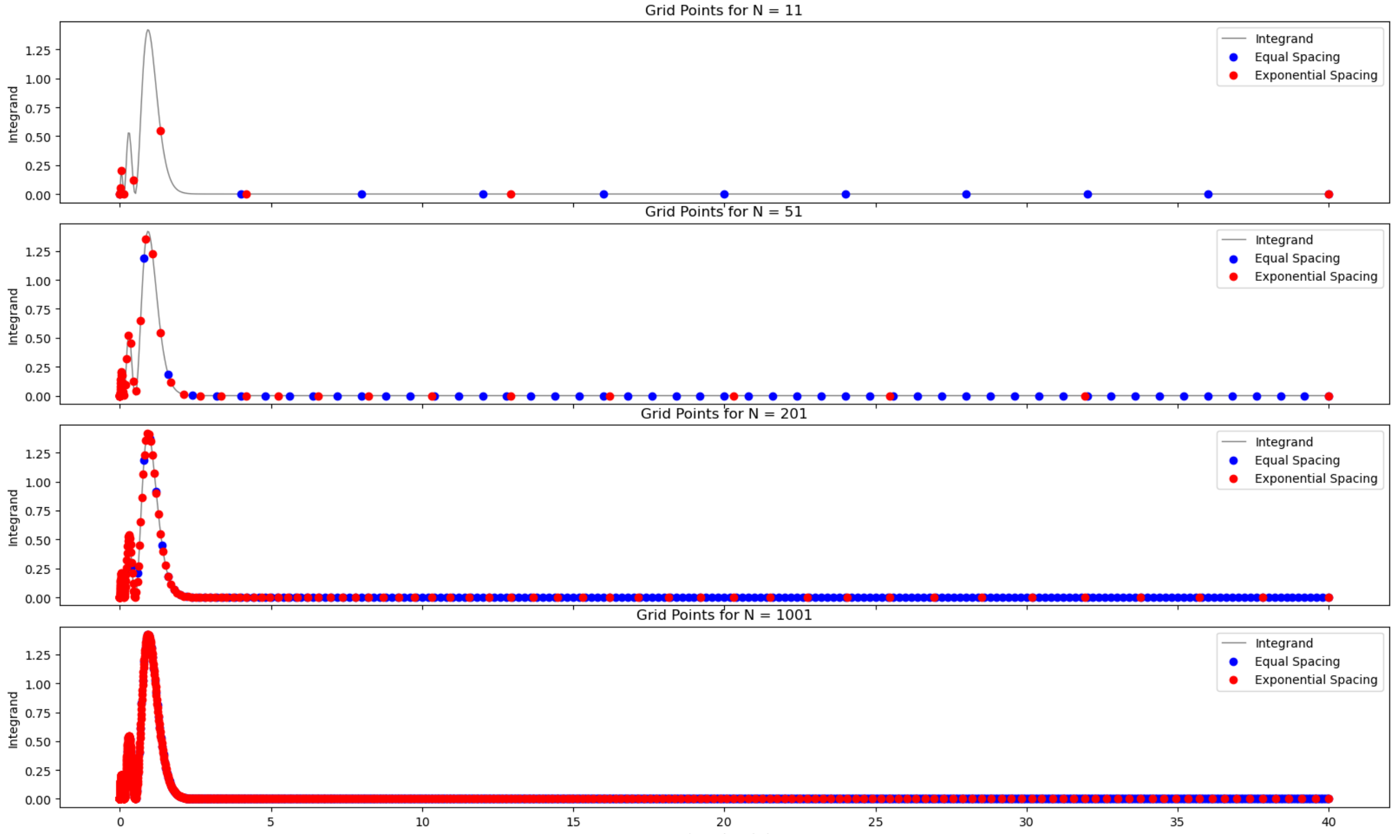

图 3: 不同节点取样分布对比