

计算物理作业 5

李明钰 22307110156

2024 年 10 月 27 日

1 题目 1：推导二阶导数的五点公式

1.1 题目描述

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

1.2 解答

我们要计算函数 $f(x)$ 在点 x_0 的二阶导数 $f''(x_0)$ 。为此，我们考虑点 $x_0 - 2h$, $x_0 - h$, x_0 , $x_0 + h$ 和 $x_0 + 2h$ 的泰勒展开：

$$f(x_0 - 2h) = f(x_0) - 2hf'(x_0) + \frac{(2h)^2}{2}f''(x_0) - \frac{(2h)^3}{6}f'''(x_0) + \frac{(2h)^4}{24}f^{(4)}(x_0) - \dots,$$

$$f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(4)}(x_0) - \dots,$$

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(4)}(x_0) + \dots,$$

$$f(x_0 + 2h) = f(x_0) + 2hf'(x_0) + \frac{(2h)^2}{2}f''(x_0) + \frac{(2h)^3}{6}f'''(x_0) + \frac{(2h)^4}{24}f^{(4)}(x_0) + \dots.$$

将这四个展开式加起来，注意到 $f'(x_0)$ 和 $f^{(3)}(x_0)$ 项会抵消：

$$f(x_0 - 2h) + 16f(x_0 - h) - 30f(x_0) + 16f(x_0 + h) + f(x_0 + 2h) = 12h^2f''(x_0) + O(h^4).$$

我们可以从中提取 $f''(x_0)$ 的逼近值：

$$f''(x_0) \approx \frac{f(x_0 - 2h) - 4f(x_0 - h) + 6f(x_0) - 4f(x_0 + h) + f(x_0 + 2h)}{h^2}.$$

这就是二阶导数的五点公式。

1.3 伪代码

Algorithm 1 五点法求二阶导数

```
1: procedure FIVEPOINTSECONDDERIVATIVE( $f, x_0, h$ )
2:    $f_{-2} \leftarrow f(x_0 - 2h)$ 
3:    $f_{-1} \leftarrow f(x_0 - h)$ 
4:    $f_0 \leftarrow f(x_0)$ 
5:    $f_1 \leftarrow f(x_0 + h)$ 
```

```

6:    $f_2 \leftarrow f(x_0 + 2h)$ 
7:    $f''(x_0) \leftarrow \frac{f_{-2} - 4f_{-1} + 6f_0 - 4f_1 + f_2}{h^2}$ 
8:   return  $f''(x_0)$ 
9: end procedure

```

2 题目 2：不同的求积分方法对比

2.1 题目描述

Radial wave function of the 3s orbital is: $R_{3s}(r) = \frac{1}{9\sqrt{3}} \times (6 - 6\rho + \rho^2) \times Z^{3/2} \times e^{-\rho/2}$, where:

- r = radius expressed in atomic units (1 Bohr radius = 52.9 pm)
- $e \approx 2.71828$
- Z = effective nuclear charge for that orbital in that atom.
- $\rho = \frac{2Zr}{n}$ 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: $r[i] = (i-1)h$; $i = 1, \dots, N$ (try different N)
2. A nonuniform integration grid, more finely spaced at small r than at large r : $r[i] = r_0(e^{t[i]} - 1)$; $t[i] = (i-1)h$; $i = 1, \dots, N$
(One typically choose $r_0 = 0.0005$ a.u., try different N). (1 a.u. = 1 Bohr radius)
3. Find out which one is more efficient, and discuss the reason.

2.2 程序描述

首先，简单画出被积分函数的大致图像

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

然后根据算法2, 计算出对不同 N 两种不同积分网格划分方式下用辛普森积分法 (算法3和用自适应辛普森积分法 (算法4) 计算出的积分结果, 画出积分结果随 N 的变化趋势, 和两种积分方法下两种网格划分的区别, 最终选择用自适应辛普森积分和非均匀积分网格作为最终的计算方法。

2.3 运行结果

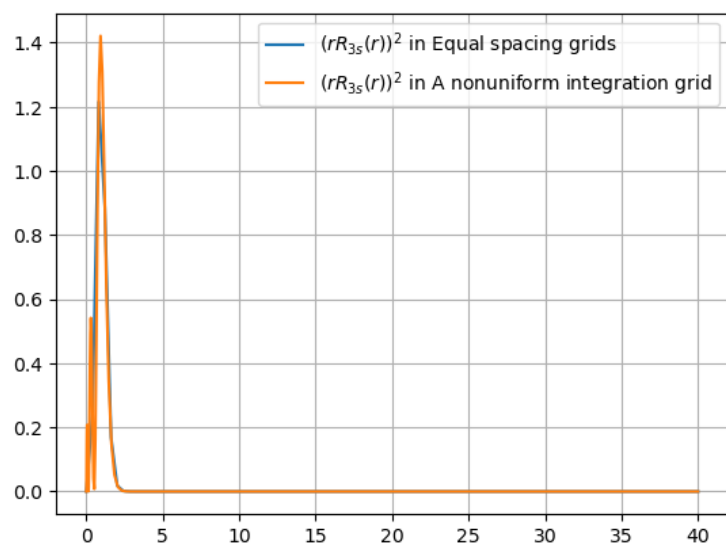


图 1: 被积函数图像

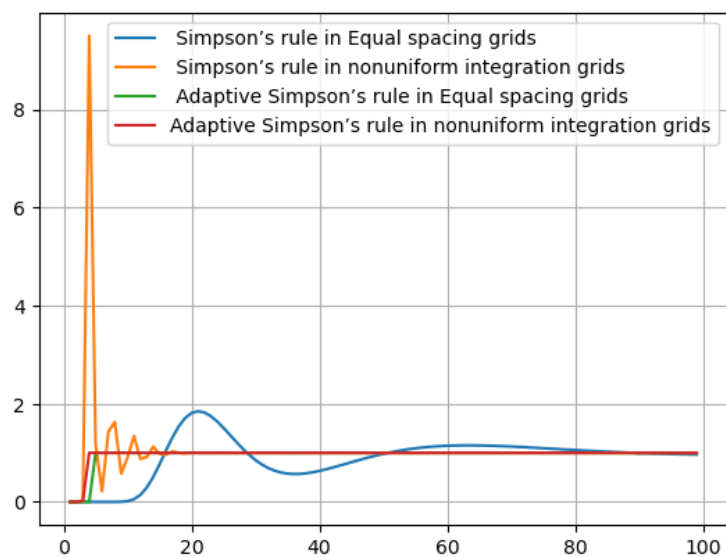


图 2: 不同积分方法结果

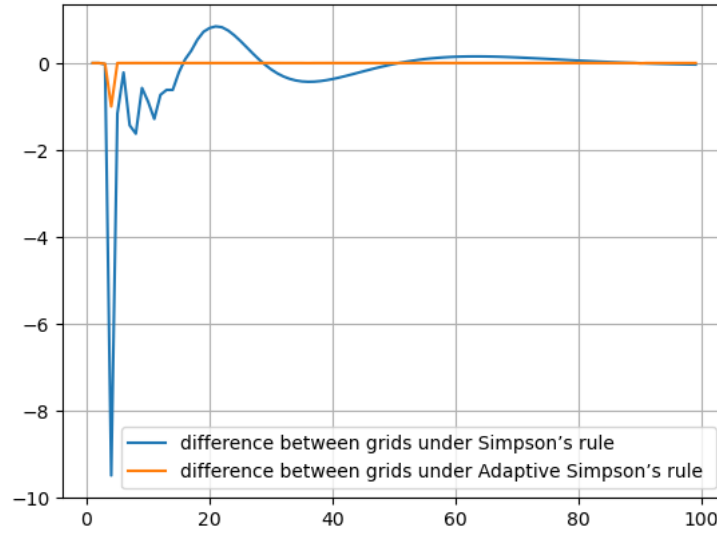


图 3: 两种积分规则下两种网格划分方式的区别

```

1 print("最终认为积分结果为: {}".format(Int_adapt(1000)[1]))
[56] ✓ 0.0s
... 最终认为积分结果为: 1.0000000000084965

```

图 4: 最终积分结果

2.4 讨论

由被积函数图像 (图1) 可以看出, 被积分函数在 r 较小时函数变化速度较快, 在 r 较大时几乎不再变化, 第二种网格划分方式2, 在 r 较小的部分取点较密集在 r 较大的部分取点较稀疏, 与被积函数性质相适应, 因此其误差较小, 表现为图2中橙色线随 N 增大较快收敛。

而使用自适应辛普森积分法 (算法4), 则会自动在网格内找到一合适的积分区间, 因此不同网格划分几乎没有差别。

2.5 伪代码

Algorithm 2 不同 N 下两种网格划分方式的积分计算

```

1: procedure COMPUTEINTEGRALS( $r_{\min}, r_{\max}, N, r_0, f$ )
2:    $r\_array1 \leftarrow \text{linspace}(r_{\min}, r_{\max}, N)$                                 ▷ 创建等间距数组
3:    $h2 \leftarrow \frac{\ln(\frac{r_{\max}}{r_0} + 1)}{N-1}$                                           ▷ 计算步长因子
4:    $t\_array \leftarrow \text{arange}(N) \times h2$                                        ▷ 创建 t 数组
5:    $r\_array2 \leftarrow r_0 \times (\exp(t\_array) - 1)$                              ▷ 创建非等间距数组
6:    $int1 \leftarrow 0$ 
7:    $int2 \leftarrow 0$ 
8:   for  $i \leftarrow 0$  to  $N - 2$  do
9:      $int1 \leftarrow int1 + \text{Int}(f, r\_array1[i], r\_array1[i + 1])$              ▷ 在  $r\_array1$  上积分
10:     $int2 \leftarrow int2 + \text{Int}(f, r\_array2[i], r\_array2[i + 1])$              ▷ 在  $r\_array2$  上积分
11:  end for

```

12: **return** $int1, int2$

13: **end procedure**

Algorithm 3 辛普森法积分 (Simpson's Rule Integration)

1: **procedure** SIMPSONINTEGRATION(f, a, b, n)

2: $h \leftarrow \frac{b-a}{n}$ ▷ 计算步长

3: $S \leftarrow f(a) + f(b)$ ▷ 初始化积分和

4: **for** $i \leftarrow 1$ to $n - 1$ **do**

5: **if** i 是奇数 **then**

6: $S \leftarrow S + 4 \cdot f(a + i \cdot h)$ ▷ 奇数项系数为 4

7: **else**

8: $S \leftarrow S + 2 \cdot f(a + i \cdot h)$ ▷ 偶数项系数为 2

9: **end if**

10: **end for**

11: $S \leftarrow \frac{h}{3} \cdot S$ ▷ 乘上 $\frac{h}{3}$ 得到最终积分结果

12: **return** S

13: **end procedure**

Algorithm 4 自适应辛普森法积分 (Adaptive Simpson's Rule Integrations)

1: **procedure** ADAPTIVESIMPSON(f, a, b, ϵ)

2: $m \leftarrow \frac{a+b}{2}$

3: $S1 \leftarrow \frac{b-a}{6} (f(a) + 4f(m) + f(b))$ ▷ 整个区间的辛普森法估计

4: $S2 \leftarrow \frac{m-a}{6} (f(a) + 4f(\frac{a+m}{2}) + f(m))$ ▷ 左半区间的估计

5: $S3 \leftarrow \frac{b-m}{6} (f(m) + 4f(\frac{m+b}{2}) + f(b))$ ▷ 右半区间的估计

6: $S2 \leftarrow S2 + S3$ ▷ 组合左半区间和右半区间的结果

7: **if** $|S2 - S1| < 15 \cdot \epsilon$ **then**

8: **return** $S2 + \frac{S2-S1}{15}$ ▷ 返回积分值和误差估计

9: **else**

10: **return** ADAPTIVESIMPSON($f, a, m, \frac{\epsilon}{2}$) + ADAPTIVESIMPSON($f, m, b, \frac{\epsilon}{2}$) ▷ 递归计算

11: **end if**

12: **end procedure**
