作业3解答

郭凯一

2000011307

1.正交多项式与高斯点

由正交归一多项式的性质$(p_i, p_j) = \delta_{ij}$，我们可以推出所有多项式的值。

- 设$p_0 = a_0$

$$(p_0, p_0) = 1 = \int_0^\infty w(x) p_0(x)^2 dx = a_0^2 \Rightarrow a_0 = \pm 1$$

  取$a_0 = 1$，于是$p_0(x) = 1$
- 设$p_1 = a + bx$

$$\begin{aligned}(p_0, p_1) = \int_0^\infty e^{-x}(a+bx) \times 1 dx = a + b = 0 \\ (p_1, p_1) = \int_0^\infty e^{-x}(a+bx)^2 dx = a^2 + 2ab + 2b^2 = 1\end{aligned}$$

  于是$b = -a = \pm 1$，取$p_1(x) = x - 1$
- 设$p_2 = ax^2 + bx + c$，由

$$\begin{aligned}(p_0, p_2) = 0 \\ (p_1, p_2) = 0 \\ (p_2, p_2) = 1\end{aligned}$$

  可以解出所有系数：

$$\begin{cases} a = \frac{1}{2} \\ b = -2 \\ c = 1 \end{cases}$$

  于是$p_2(x) = \frac{1}{2}x^2 - 2x + 1$

$p_2(x)$对应的高斯点为其零点，得到：

$$
\begin{aligned}
x_1 &= 2-\sqrt{2} \\
x_2 &= 2+\sqrt{2}
\end{aligned}
$$

于是有高斯积分公式：

$$\int_0^\infty f(x)\omega_x dx = A_1 f(x_1) + A_2 f(x_2)$$

代入$f = p_k(k=0,1)$，计算各个A_k：

$$\sum_{i=1}^{2} A_i p_k(x_i) = \int_0^\infty p_k(x)\omega(x)dx = \begin{cases} 1, & k=0 \\ 0, & k\neq 0 \end{cases}$$

即

$$
\begin{gathered}
A_1 + A_2 = 1 \\
A_1(2-\sqrt{2}-1) + A_2(2+\sqrt{2}-1) = 0
\end{gathered}
$$

解出：

$$
\begin{aligned}
A_1 &= \frac{1}{2}+\frac{\sqrt{2}}{4} \\
A_2 &= \frac{1}{2}-\frac{\sqrt{2}}{4}
\end{aligned}
$$

于是所求积分的近似结果为：

$$
\begin{aligned}
&\int_0^\infty \ln(1-e^{-x})dx \\
&\approx (\frac{1}{2}+\frac{\sqrt{2}}{4})e^{2-\sqrt{2}}\ln(1-e^{-(2-\sqrt{2})}) + (\frac{1}{2}-\frac{\sqrt{2}}{4})e^{2+\sqrt{2}}\ln(1-e^{-(2+\sqrt{2})}) \\
&= -1.40
\end{aligned}
$$

2.三种方法求积分

1. n点的梯形法则公式为：

$$\int f(x)dx = h(\frac{1}{2}f_0 + f_1 + \cdots + f_{n-2} + \frac{1}{2}f_{n-1})$$

2. Simpson法对于三个点$x, x+h, x+2h$的积分近似公式为:

$$\int_x^{x+2h} f(x')dx' = \frac{1}{3}h(f_0 + 4f_1 + f_2)$$

   其中$f_k = f(x+kh)$
   推广到奇数个等间距点$f_0, \cdots, f_{2n}$:

$$\int_x^{x+2nh} = \frac{1}{3}h\left(2(f_2 + \cdots + f_{2n-2}) + f_0 + f_{2n} + 4(f_1 + \cdots + f_{2n-1})\right)$$

   可以看到，辛普森方法要求取点数量为奇数。
3. 利用Gauss-Chebyshev方法：根据切比雪夫多项式的基本性质:

$$T_n(x) = \cos(n \arccos x)$$
$$\int_{-1}^{1} dx \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}} = \begin{cases} 0, & i \neq j \\ \frac{\pi}{2}, & i = j \neq 0 \\ \pi, & i = j = 0 \end{cases}$$

   在这组正交完备基矢下，将任意函数的展开写作:

$$f(x) = \sum_m c_m T_m(x),$$

   其中c_m可以由积分严格求出，或由求和来近似:

$$c_m = \frac{2}{\pi(1+\delta_{m0})} \int_{-1}^{1} f(x) T_m(x) \frac{1}{\sqrt{1-x^2}} dx \tag{1}$$
$$\approx c_{N,m} = \frac{2}{N(1+\delta_{m0})} \sum_{k=0}^{N-1} \cos\frac{m\pi(k+1/2)}{N} f\left(\cos\frac{\pi(k+1/2)}{N}\right) \tag{2}$$

   令$m=0$:

$$\int_{-1}^{1} f(x)\frac{1}{\sqrt{1-x^2}}dx=\pi c_0 \approx \frac{\pi}{N}\sum_{k=0}^{N-1} f(\cos\frac{\pi(k+1/2)}{N})$$

令$f(x)=\sqrt{1-x^2}g(x)$:

$$\int_{-1}^{1} g(x)dx \approx \frac{\pi}{N}\sum_{k=0}^{N-1} g\left(\cos\frac{\pi(k+1/2)}{N}\right)\sin\frac{\pi(k+1/2)}{N}$$

再对区间进行线性平移与缩放，得到:

$$\begin{aligned}\int_a^b h(x)dx &= \int_{-1}^{1} h\left(a+\frac{b-a}{2}(x'+1)\right)\frac{b-a}{2}dx' \quad (3)\\ &\approx \frac{(b-a)\pi}{2N}\sum_{k=0}^{N-1}\sin\frac{\pi(k+1/2)}{N}h\left(a+\frac{b-a}{2}(1+\cos\frac{\pi(k+1/2)}{N})\right) \quad (4)\end{aligned}$$

计算结果如下图所示:

```
D:\PKU\notes\Computational Physics\homework-3>python T2.py
11 points:Trapezoidal:1.821019999057492 simpson:1.21402450925036
101 points:Trapezoidal:0.27372390615766756      simpson:0.2312791601399645
1001 points:Trapezoidal:0.21998408386198728     simpson:0.21938700392497693
points:10,Gauss-Chebyshev:0.3041521287969225
points:100,Gauss-Chebyshev:0.22013932287381993
```

可以看到，高斯-切比雪夫方法在计算量接近的前提下相比前两种方法更加精确。

3.三种方法求根

由于原方程的零点为二重根，在任意区间的两端函数值均恒为正，二分法无法进行下去:

```
(base) d:\PKU\notes\Computational Physics\homework-3>python T3.py
Traceback (most recent call last):
  File "d:\PKU\notes\Computational Physics\homework-3\T3.py", line 52, in <module>
    x=binary_division(f,-10,10,1e-6)
  File "d:\PKU\notes\Computational Physics\homework-3\T3.py", line 8, in binary_division
    assert(a<b and f(a)*f(b)<0)
AssertionError
```

编写程序使用Newton-Ralphson与割线法计算方程的正根($x\neq 0$)，得到结果如下（其中函数参数中精度$prec=10^{-7}$）：

```
(base) d:\PKU\notes\Computational Physics> d: && cd "d:\PKU\notes\Computational Physics" && cmd /C "D:\anaconda\python.exe "c:\Users\life is physics\.vscode\extensions\ms-python.python-2022.18.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher" 3332 -- "d:\PKU\notes\Computational Physics\homework-3\T3.py" "
1.895494373469356
1.895616498713762
```

图中上面为N-R方法得到的根，下面为割线法得到的根。由于是二重零点，这两种迭代方法得到的精度均只有大约$\sim \sqrt{10^{-7}}$.如需更高精度可以继续减小prec的值。