

《计算物理》第 3 次作业

1900017812 高乐耘*

2022 年 11 月 1 日

I. 高斯-拉盖尔积分

定义内积:

$$(p, q) := \int_0^{+\infty} p(x)q(x)e^{-x}dx \quad (1)$$

其中 p, q 均为区间 $[0, +\infty)$ 上平方可积的实函数。在该内积空间中, $1, x, x^2$ 线性无关, 我们将在这组基底张成的子空间中进行格拉姆-施密特正交化。首先我们取:

$$p_0 = 1 \quad (2)$$

并算得:

$$(p_0, p_0) = \int_0^{+\infty} p_0(x)p_0(x)e^{-x}dx = \int_0^{+\infty} e^{-x}dx = 1 \quad (3)$$

$$(x, p_0) = \int_0^{+\infty} xp_0(x)e^{-x}dx = \int_0^{+\infty} xe^{-x}dx = 1 \quad (4)$$

$$(x^2, p_0) = \int_0^{+\infty} x^2p_0(x)e^{-x}dx = \int_0^{+\infty} x^2e^{-x}dx = 2 \quad (5)$$

由格拉姆-施密特正交化算法得:

$$p_1 = x - \frac{(x, p_0)}{(p_0, p_0)}p_0 = x - 1 \quad (6)$$

并算得:

$$(p_1, p_1) = \int_0^{+\infty} p_1(x)p_1(x)e^{-x}dx = \int_0^{+\infty} (x-1)^2e^{-x}dx = 1 \quad (7)$$

$$(x^2, p_1) = \int_0^{+\infty} x^2p_1(x)e^{-x}dx = \int_0^{+\infty} x^2(x-1)e^{-x}dx = 4 \quad (8)$$

由格拉姆-施密特正交化算法得:

$$p_2 = x^2 - \frac{(x^2, p_0)}{(p_0, p_0)}p_0 - \frac{(x^2, p_1)}{(p_1, p_1)}p_1 = x^2 - 4x + 2 \quad (9)$$

*电子邮件地址: seeson@pku.edu.cn 手机号: 13759115414

至此我们得到了区间 $[0, +\infty)$ 上不超过 2 阶多项式组成的线性空间的一个标准正交基:

$$\begin{cases} p_0 = 1, \\ p_1 = x - 1, \\ p_2 = x^2 - 4x + 2. \end{cases} \quad (10)$$

令 $p_2(x) = 0$, 解得 $x_1 = 2 - \sqrt{2}$, $x_2 = 2 + \sqrt{2}$ 。假设对于任意不超过 3 阶多项式 p 均有:

$$\int_0^{+\infty} p(x)e^{-x}dx = A_1p(x_1) + A_2p(x_2) \quad (11)$$

此为高斯积分之定义。特别地, 对于 $p = p_0$ 和 $p = p_1$, 我们有:

$$\int_0^{+\infty} p_0(x)e^{-x}dx = A_1p_0(x_1) + A_2p_0(x_2) = 1, \quad (12)$$

$$\int_1^{+\infty} p_1(x)e^{-x}dx = A_1p_1(x_1) + A_2p_1(x_2) = 0. \quad (13)$$

由克莱姆法则立即得到:

$$A_1 = \frac{\begin{vmatrix} 1 & p_0(x_2) \\ 0 & p_1(x_2) \end{vmatrix}}{\begin{vmatrix} p_0(x_1) & p_0(x_2) \\ p_1(x_1) & p_1(x_2) \end{vmatrix}} = \frac{\begin{vmatrix} 1 & 1 \\ 0 & 1 + \sqrt{2} \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 1 - \sqrt{2} & 1 + \sqrt{2} \end{vmatrix}} = \frac{1 + \sqrt{2}}{2\sqrt{2}} = \frac{2 + \sqrt{2}}{4}, \quad (14)$$

$$A_2 = \frac{\begin{vmatrix} p_0(x_1) & 1 \\ p_1(x_1) & 0 \end{vmatrix}}{\begin{vmatrix} p_0(x_1) & p_0(x_2) \\ p_1(x_1) & p_1(x_2) \end{vmatrix}} = \frac{\begin{vmatrix} 1 & 1 \\ 1 - \sqrt{2} & 0 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 1 - \sqrt{2} & 1 + \sqrt{2} \end{vmatrix}} = \frac{-1 + \sqrt{2}}{2\sqrt{2}} = \frac{2 - \sqrt{2}}{4}. \quad (15)$$

至此我们得到了两点高斯-拉盖尔求积公式:

$$\int_0^{+\infty} p(x)e^{-x}dx = \frac{2 + \sqrt{2}}{4}p(2 - \sqrt{2}) + \frac{2 - \sqrt{2}}{4}p(2 + \sqrt{2}) \quad (16)$$

由此可以近似得到:

$$\begin{aligned} \int_0^{+\infty} \ln(1 - e^{-x})dx &= \int_0^{+\infty} e^x \ln(1 - e^{-x})e^{-x}dx \\ &\approx \frac{2 + \sqrt{2}}{4}e^x \ln(1 - e^{-x}) \Big|_{2 - \sqrt{2}} + \frac{2 - \sqrt{2}}{4}e^x \ln(1 - e^{-x}) \Big|_{2 + \sqrt{2}} \\ &\approx -1.396 \end{aligned} \quad (17)$$

II. 更多数值积分方法

查阅资料 [1][2] 知, 这个积分带有 22 位有效数字的结果为:

$$\int_1^{100} \frac{e^{-x}}{x}dx = \text{Ei}(-100) - \text{Ei}(-1) \approx -\text{Ei}(-1) \approx 0.2193839343955202736772 \quad (18)$$

源文件 `intg/intg.c` 实现了题中所述的三种积分方法，其中梯形法和辛普森方法合并了各相邻区间公共端点的贡献，以减少运算量和提高精度；高斯-切比雪夫积分方法考虑 n 个格点时各权重因子¹ 均为 $\pi \sin \theta_i/n$ ，此处 $\theta_i = (i+1/2)\pi/n$ ，格点坐标 $x_i = x_1 + (\cos \theta_i + 1)(x_2 - x_1)/2$ ，对于 $i = 0, 1, \dots, n-1$ 。源文件 `intg/intg-test.c` 实现了对三种积分方法的简单测试。源文件 `intg/intg-smp.c` 实现了本题的求解流程。各种积分方法的积分结果列于表 I 并绘于图 1 中。

表 I: 各种积分方法结果比较

方法类型	积分结果	偏差
10 格点梯形方法	2.023343	1.8×10^{-0}
100 格点梯形方法	0.274735	5.5×10^{-2}
1000 格点梯形方法	0.219985	6.0×10^{-4}
10 格点辛普森方法	0.676144	4.6×10^{-1}
100 格点辛普森方法	0.220757	1.4×10^{-3}
1000 格点辛普森方法	0.219384	2.0×10^{-7}
10 格点高斯-切比雪夫方法	0.304152	8.5×10^{-2}
100 格点高斯-切比雪夫方法	0.220139	7.6×10^{-4}

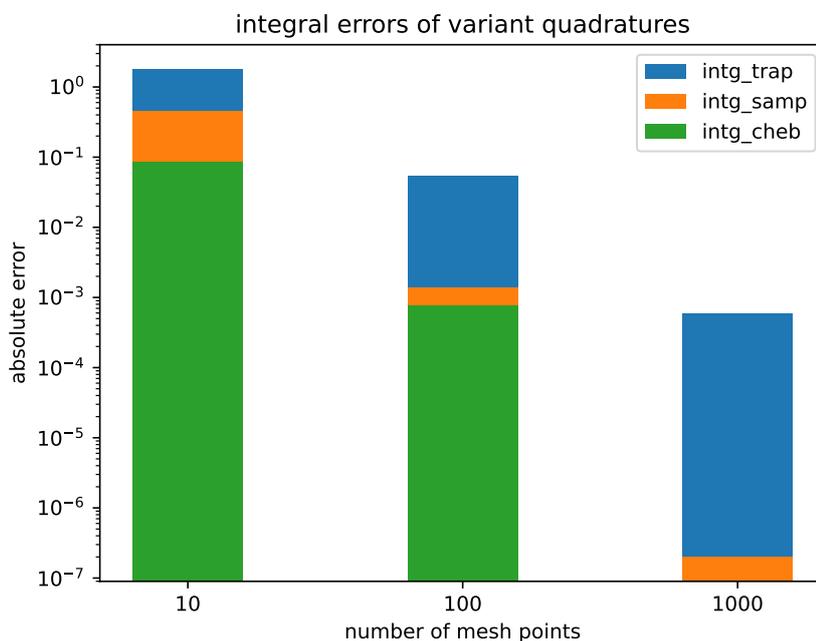


图 1: 不同求积方法的绝对值误差。intg_trap, intg_samp, intg_cheb 分别表示梯形法、辛普森方法和高斯-切比雪夫方法。按照题目要求，高斯-切比雪夫方法没有考虑 1000 个格点的情况。

从本实验中可以看到这样的规律：对于本题中的指数积分，梯形方法的表现是最差的。这

¹此处的权重因子是针对包括权函数部分在内的被积函数而言的。

种方法一般只能精确描述一次函数的积分，对各区间内任意一点邻域内函数泰勒展开高阶项的贡献一般无法精确描述，其精度正比于区间长度的二次方。与此形成对比的是，辛普森方法本身就是抛物线近似，由于还是偶数阶，还要考虑对称性带来的额外精度，它的精度正比于区间长度的四次方，比梯形方法表现得好得多。而高斯-切比雪夫方法作为高斯积分的一种，由 n 个格点可以精确描述任意不超过 $2n - 1$ 阶多项式的积分，在格点数目较小时就已经显示出了明显的精度优势。

III. 一元非线性方程求解

注意到：

$$0 = x^2 - 4x \sin x + (2 \sin x)^2 = (x - 2 \sin x)^2 \Leftrightarrow x - 2 \sin x = 0 \quad (19)$$

图 2 给出了函数 $y = x^2 - 4x \sin x + (2 \sin x)^2$ 和 $y = x - 2 \sin x$ 在 $[1.5, 2.5]$ 上的图像。从图中明显看出，函数 $y = x - 2 \sin x$ 在 $[1.5, 2.5]$ 上单调递增，在此区间内有且仅有一个一阶零点，我们将其记为 x_0 。根据 $\sin x$ 的有界性不难得知， x_0 是 $(0, +\infty)$ 区间内这一函数的唯一零点。由式 (19) 知， x_0 是函数 $y = x^2 - 4x \sin x + (2 \sin x)^2$ 的二阶零点，也是正半轴上唯一的零点，但直接求解这个二重根是一个病态问题。作为替代的是，我们将对函数 $y = x - 2 \sin x$ 在区间 $[1.5, 2.5]$ 上求零点。由于该函数在区间 $[1.5, 2.5]$ 上单调递增，且在此区间内导数均不太小，预计二分法、牛顿-拉尔夫森迭代法和割线法都将有较好的求根结果。

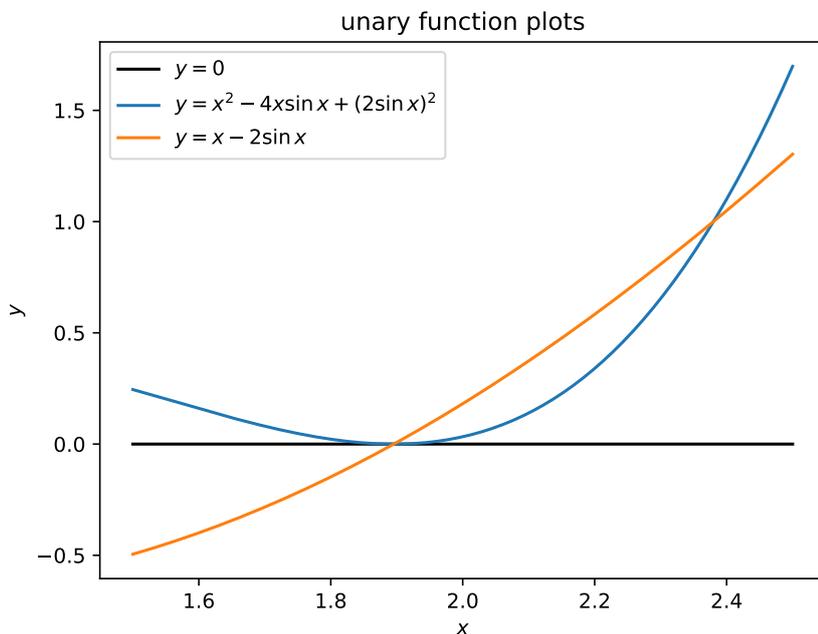


图 2: 待求零点函数及其约化函数图像

三种方程求根的方法实现于源文件 solve/solve.c 中。在这一实现中，除二分法还在函数值为零时停止迭代外，其余两种求根方法均以自变量精度无法在给定机器精度下进一步改善为判停标准。在实现中我们考虑了各种浮点代数可能产生的特殊值，例如 NAN 可能影响相

等性判断，正负 INF 可以作为特殊的“收敛”值结束迭代循环等，但我们忽略了可能因迭代发散（特别是周期性振荡）而造成的死循环问题。在此我们求解的问题没有奇异性，以上我们简单的判停标准和面对难以穷尽的极端情况的鸵鸟策略是没有问题的（当然，我们可以通过显式地限制迭代次数来规避死循环问题，但对于求解这个良态方程而言完全没有必要）。源程序 solve/solve-test.c 实现了对这三种求根方法的简单测试。

使用这三种求根方法求解方程 $x - 2 \sin x = 0$ 并输出中间过程和最终结果的代码逻辑位于 solve/solve-equ.c 中。为统一标准对收敛速度进行比较，对于牛顿-拉尔夫森方法，我们使用初值 $x_1 = 2.5$ ；对于二分法和割线法，我们都使用初值 $x_1 = 1.5, x_2 = 2.5$ 。运行这一程序，我们从标准错误流中明显看出，三种迭代方法最终收敛到了相同的数值结果：

```
solve_bin: 1.89549 (0x1.e53f1cbeaa9bcp+0)
solve_NR: 1.89549 (0x1.e53f1cbeaa9bcp+0)
solve_sec: 1.89549 (0x1.e53f1cbeaa9bcp+0)
```

在以上输出中我们还给出了圆括号包围的十六进制浮点值，以确定这三个浮点值的机器级表示完全相同。由于此处所用浮点数为 IEEE 754 标准下的 64 位浮点数，其尾数有 53 位，对应 14 位十六进制数码。此处最高位为 1，满足规范化条件，后 13 位十六进制数均为可以连续变化的有效数字，这一表示的绝对精度为 $16^{-13} \approx 2 \times 10^{-16}$ ，所以这三个数值求根结果可以共同地表示为：

$$x_0 = 1.8954942670339809 \pm 0.00000000000000002 \quad (20)$$

使用 Python 3 表达式 `lambda x: x**2 - 4*x*np.sin(x) + (2*np.sin(x))**2` 代入以上十六进制精确浮点值求值函数 $y = x^2 - 4x \sin x + (2 \sin x)^2$ ，我们在 64 位机器精度下得到 0.0，这并不稀奇，此处为该函数的极小值点，函数值对自变量的变化并不敏感，即使 x_0 还是有些误差，这一误差很难反映在它对应的函数值中。我们再用下面这个 Python 3 表达式 `lambda x: x - 2*np.sin(x)` 进一步检验函数 $y = x - 2 \sin x$ 在 x_0 处的函数值，发现还是为 0.0，而此处的一阶导数值约为 1.6，函数值对自变量的变化比较敏感，这样便有力地验证了我们求解结果的精确性。

根据程序标准输出中的统计结果，二分法、牛顿-拉尔夫森迭代法和割线法达到机器精度分别使用了 50、6 和 8 次迭代，从中可以深刻感受到后两者相比二分法收敛速度的加快。割线法的收敛速度稍慢于牛顿-拉尔夫森迭代法，这与收敛速度的理论估计结果（分别为渐进 $(\sqrt{5} + 1)/2$ 和 2 阶收敛）一致。图 3 向我们清晰地呈现了三者的不同收敛过程，图中下降越陡的曲线（组）表示该求根方法收敛越快。

最后，让我们在图 4 中描出求得的零点，来结束对本题的讨论。

参考文献

- [1] Integral from 1 to 100 of $(e^{-x})/x$ - Symbolab. view the page.
- [2] Exponential integral $Ei(x)$ Calculator - High accuracy calculation. view the page.

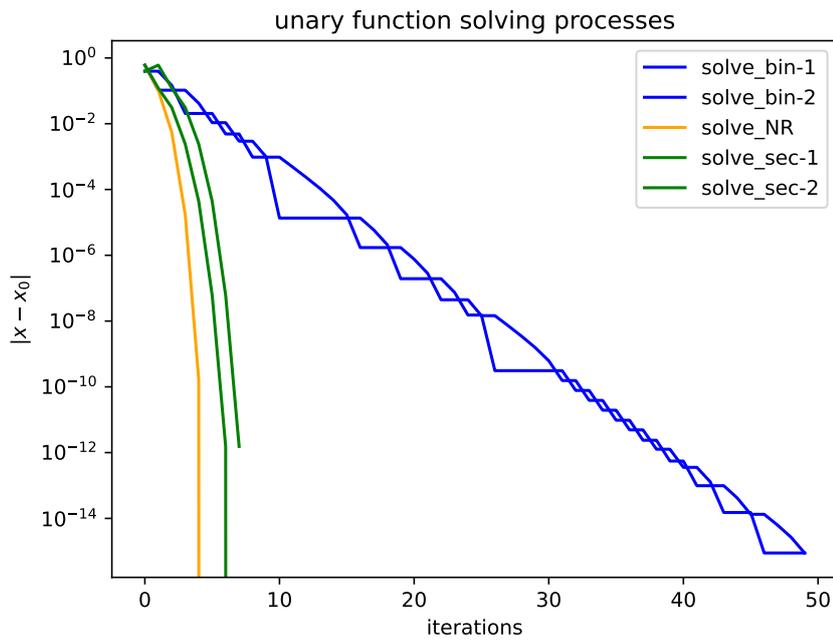


图 3: 三种求根方法的收敛过程。横坐标为迭代次数。纵坐标给出了零点区间端点 (二分法) 或零点估计值 (牛顿-拉尔夫森方法为一个, 割线法为两个) 与求得的零点之差的绝对值, 其比例尺为对数线性。平行于纵轴的直线表示该绝对值为零, 在图中无法表示。

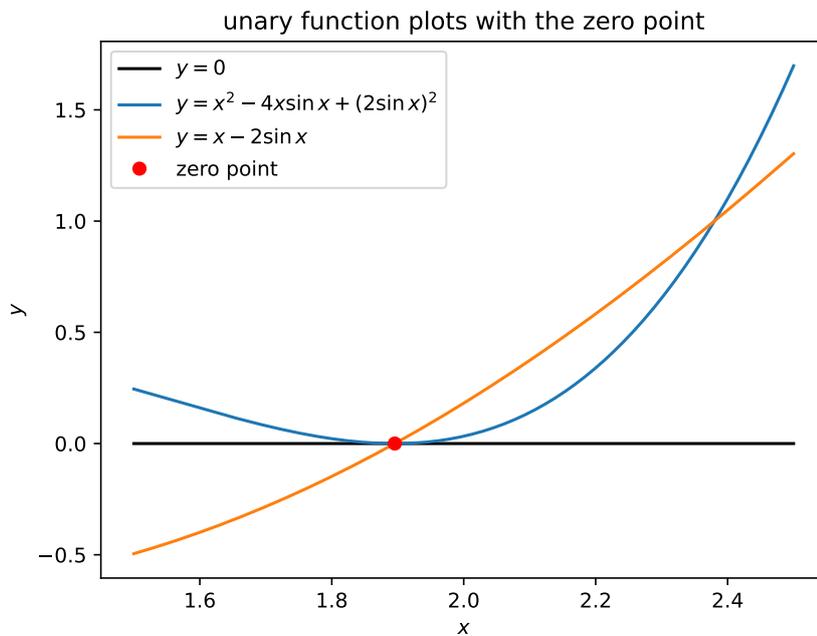


图 4: 求得的零点及相关函数图像

A 项目源代码

参见 <https://github.com/lyazj/numphy-03>。