2024 秋数值代数–实验报告 #1

姓名：李奕萱学号：PB22000161

2024 年 9 月 23 日

运行环境：[win11,vscode,py3]

实验内容与要求

- 问题 1：给定两个函数 $f(x) = \sqrt{x^2+25} - 5$ 和 $g(x) = \frac{x^2}{\sqrt{x^2+25}+5}$，采用单精度(即 float 型) 进行编程 (注意：开方 sqrt(x) 等内置函数的输出结果默认是双精度，需要强制转成单精度)，分别取 $x = 4^{-1}, 4^{-2}, 4^{-3}, \cdots, 4^{-11}$，输出相应的函数值 $f(x)$ 和 $g(x)$，计算结果**保留 12 位尾数** (用科学计数形式, 参见表 1 中的数据格式)，比较并分析两种方法得到的计算结果。你认为哪种方法得到的计算结果更可靠？请给出你的理由或分析。

- 问题 2：给定如下数据

  4042.045051380452, 0.000531415926535, -2759471.276702747, 0.000055705299674289 5, 2755463.874010974, -34.64291531256604, -0.000031415926535.

  分别采取以下 4 种方式求和：

  (a) 顺序求和；　(b) 逆序（从后往前）求和；

  (c) 按绝对值从大到小的顺序, 依次求和;

  (d) 按绝对值从小到大的顺序, 依次求和.

  采用双精度进行计算，计算结果中的尾数至少保留 9 位小数 (用科学计数形式，比如 1.234567899E−11). 比较 4 种方法得到的计算结果；你认为哪种方法得到的计算结果更精确 (即误差最小；提示：想办法算出精确值)？试给出你的理由或分析。

- 问题 3：(计算结果保留 10 位有效数字) Let $\pi \approx 3.14159265358979 32$.

  Write a routine to compute $\sin x$ for x in radians as follows. First, using properties of the sine function, reduce the range so that $-\pi/2 \leqq x \leqq \pi/2$. Then if $|x| < 10^{-8}$, set $\sin x \approx x$; if $|x| > \pi/6$, set $u = x/3$, compute $\sin u$ by the formula below, and then set $\sin x \approx [3 - 4\sin^2 u]\sin u$; if $|x| \leqq \pi/6$, set $u = x$ and compute $\sin u$ as follows:

$$\sin u \approx u \left[\frac{1 - \left(\frac{29593}{207636}\right)u^2 + \left(\frac{34911}{7613320}\right)u^4 - \left(\frac{479249}{11511339840}\right)u^6}{1 + \left(\frac{1671}{69212}\right)u^2 + \left(\frac{97}{351384}\right)u^4 + \left(\frac{2623}{1644477120}\right)u^6} \right]$$

  Use this routine to compute $\sin x$ for $x = \frac{\pi}{2024}, \frac{\pi}{10}, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}$, respectively.

  Compare those results with results obtained by the Taylor formula, i.e., $\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$.

1 数值结果（请尽量列表或作图）

- 问题 1

x	f(x)	g(x)
4^ -1 =2.500000000000e-01	6.246089935303e-03	6.246098317206e-03
4^ -2 =6.250000000000e-02	3.905296325684e-04	3.906097263098e-04
4^ -3 =1.562500000000e-02	2.431869506836e-05	2.441400283715e-05
4^ -4 =3.906250000000e-03	1.430511474609e-06	1.525878587927e-06
4^ -5 =9.765625000000e-04	0.000000000000e+00	9.536743306171e-08
4^ -6 =2.441406250000e-04	0.000000000000e+00	5.960464566357e-09
4^ -7 =6.103515625000e-05	0.000000000000e+00	3.725290353973e-10
4^ -8 =1.525878906250e-05	0.000000000000e+00	2.328306471233e-11
4^ -9 =3.814697265625e-06	0.000000000000e+00	1.455191544521e-12
4^ -10 =9.536743164062e-07	0.000000000000e+00	9.094947153255e-14
4^ -11 =2.384185791016e-07	0.000000000000e+00	5.684341970784e-15

图 1: 题 1 计算结果

- 问题 2 (用科学计数形式, 尾数至少保留 9 位小数)

```
Sequential Sum: 1.262001062e-11
Reverse Sum: -6.821210263e-11
Absolute Descending Sum: 4.684540313e-11
Absolute Ascending Sum: 0.000000000e+00
```

图 2: 题 2 计算结果（mathematica 精确计算结果同第三个）

- 问题 3 (保留 10 位有效数字, $\pi \approx 3.1415926535897932$)

PI/2024	1.552169660e-03	1.552169660e-03	-2.168404345e-19
PI/10	3.090169944e-01	3.090169943e-01	-8.207223789e-11
PI/6	5.000000000e-01	4.999999919e-01	-8.130976781e-09
PI/4	7.071067812e-01	7.071064696e-01	-3.116113539e-07
PI/3	8.660254038e-01	8.660212717e-01	-4.132126938e-06

图 3: 题 3 计算结果（其中第二列是题给方法，第三列是 Taylor 展开方法，最右边为二者之差）

2 算法分析

- 问题 1

  Step1: 分别定义 $f(x)$ 和 $g(x)$，注意控制精度。

  Step2: 构建循环输出结果。

- 问题 2

Step1: 输入数据

Step2: 分别定义四种求和（注意控制精度）

step3: 输出四种求和的结果

- 问题 3

  Step1: 定义题目中的函数

  Step2: 定义 Taylor 函数

  step3: 分别输出二者结果并进行比较

3 结果分析

- 问题一 $f(x)$ 和 $g(x)$ 为同一算式的不同表达方式

  二者计算误差如下：

```
4^  -1 =2.500000000000e-01   -8.381903171539e-09
4^  -2 =6.250000000000e-02   -8.009374141693e-08
4^  -3 =1.562500000000e-02   -9.530776878819e-08
4^  -4 =3.906250000000e-03   -9.536711331748e-08
4^  -5 =9.765625000000e-04   -9.536743306171e-08
4^  -6 =2.441406250000e-04   -5.960464566357e-09
4^  -7 =6.103515625000e-05   -3.725290353973e-10
4^  -8 =1.525878906250e-05   -2.328306471233e-11
4^  -9 =3.814697265625e-06   -1.455191544521e-12
4^ -10 =9.536743164062e-07   -9.094947153255e-14
4^ -11 =2.384185791016e-07   -5.684341970784e-15
```

图 4: f(x) 和 g(x) 误差如下

我认为 $f(x)$ 计算结果更可靠，math.sqrt 这一步结果为双精度，当强行切换为单精度时，会产生舍入误差，除法亦是同理，误差积累，所以我认为 $f(x)$ 结果更可靠。

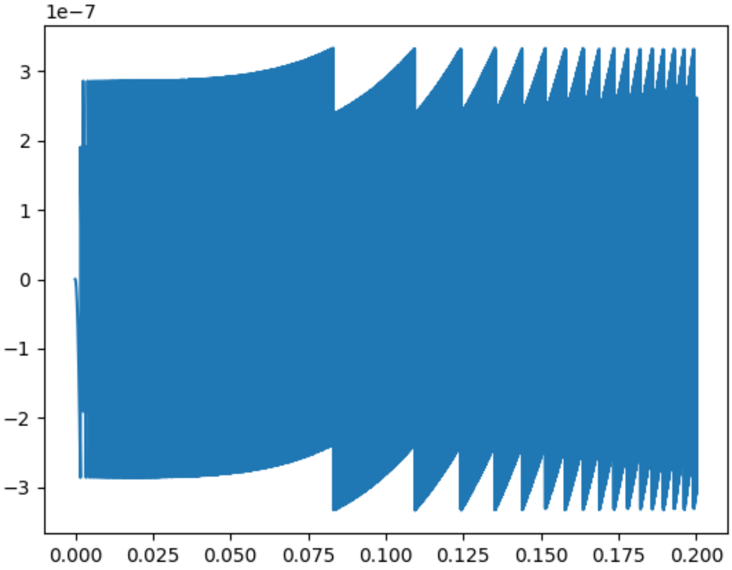

图 5: f(x) 和 g(x) 误差作图（步长为 1e-7）

同时可以发现，当数字较大的时候（对应步长较大），二者相对误差接近 0,，当逐渐减小时（对应步长减小），误差会突然增大，然后再逐渐降低。

- 问题二

  在与精确结果进行对比后得知，第三种方法得到的结果最精确，进分析，绝对值较大的数可存储的小数点后的位数最少，而绝对值较小的数则可储存较多，由此可见，按绝对值从大到小的顺序进行计算可以尽可能地保留小数点后面的精度。

- 问题三

PI/2024	0.000000000e+00	-2.168404345e-19
PI/10	5.551115123e-17	-8.207218238e-11
PI/6	5.551115123e-17	-8.130976725e-09
PI/4	-1.554312234e-14	-3.116113695e-07
PI/3	-1.128319660e-12	-4.132128066e-06

图 6: 两种算法分别和 sin 的误差值（左边是题目算法，右边是 Taylor 算法

从计算结果可以看出，数值越小，二者相对误差越小，同时与标准值进行比较后，可以得出，题目给的方法会更加精确。

4 实验小结

在本次数值实验中，我们分别做了三个小问题，从同一个函数不同表达形式、不同顺序求和方式、对同一个数值的不同近似表达分别进行了实验，可以得出：

1. 对于同一个式子的不同精确表达形式，计算次数少的会更加可靠准确，
2. 对于不同顺序求和，按绝对值从小到大的顺序更加准确
3. 对于一个值的不同近似表达式，精确结果也不一样，