第一章 数值计算导论 编程实验

> 孔瑞阳 计科91 2019010175

第一章上机题1：

编程实现例 1. 4 , 绘出图 1 - 2 , 体会两种误差对结果的不同影响。

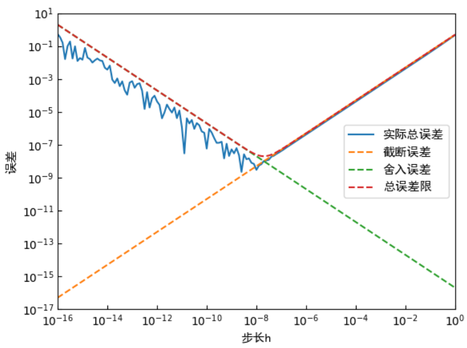
思路:

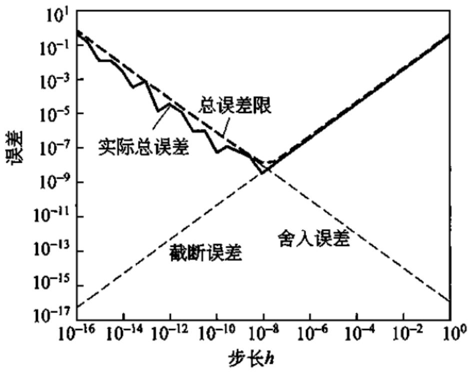
运用 python 中的 matplotlib.pyplot 包进行绘图。

利用 rcParams，gca，x(y)label，loglog，legend，x(y)ticks 指令控制图像的格式，用 plot 指令画图。

绘制结果:

左图为绘制生成的图，右图为课本中的图 1 - 2。





结果分析:

步长 h 越小，差商越接近导数的定义，所以截断误差越小；同时因为浮点数舍入误差在计算差商的过程中会 $/h$ ，所以舍入误差越大。实际误差不超过理论误差限。当步长 h 约为 $10^{-8.6}$ 时，实际总误差最小约为 $10^{-8.8}$ 。

第一章上机题3：

编程观察无穷级数 $\sum_{n=1}^{\infty}\frac{1}{n}$ 的求和计算。

(1)

问题:

采用 IEEE 单精度浮点数, 观察当 n 为何值时, 求和结果不再变化, 将它与理论分析的结论进行比较。

解题思路:

使用 C++ 进行编程。

使用以下循环代码求解，如果加上 $\frac{1}{n}$ 后的 harmonic_float 和之前的结果一样则退出循环。

```
float harmonic_float = 0.0, tmp;
int n = 0;
while (true)
{
    tmp = harmonic_float;
    ++n;
    harmonic_float += one_float / n;
    if (harmonic_float == tmp)
        break;
}
```

实验结果:

运行结果为:

```
When n = 2097152, the harmonic series recorded in float does not change.
Now the harmonic series is equal to 15.40368271.
```

即当 $n = 2097152$ 时，运算结果不再改变。

结果分析:

根据定理1.6，

当 $\left|\frac{x_1}{x_2}\right| \le \frac{1}{2}\varepsilon_{mach}$ 时，$x_1 + x_2$ 的结果一定等于 x_1；

当 $\left|\frac{x_1}{x_2}\right| > \varepsilon_{mach}$ 时，$x_1 + x_2$ 的结果一定不等于 x_1 。

在此问题中，有 $x_1 = \frac{1}{n}$ ，$x_2 = H_{n-1}$。

带入得，当 $nH_{n-1} \ge \frac{2}{\varepsilon_{mach}}$ 时，结果一定不变化；当 $nH_{n-1} < \frac{1}{\varepsilon_{mach}}$ 时，结果一定变化。

运用不等式 $\ln n \le H_n \le \ln n + 1$，当

$n\ln(n-1) \ge \frac{2}{\varepsilon_{mach}}$ 时，结果一定不变化；

当 $n[\ln(n-1)+1] < \frac{1}{\varepsilon_{mach}}$ 时，结果一定变化。

对于单精度浮点数，有 $\varepsilon_{mach} = 2^{-24}$。

运用以下 MATLAB 代码求解单精度浮点数的范围：

```
syms n_float1 n_float2
eqn_float1 = n_float1 * log(n_float1 - 1) == 2^25
eqn_float2 = n_float2 * (log(n_float2 - 1) + 1) == 2^24
n_float1 = solve(eqn_float1, n_float1)
n_float2 = solve(eqn_float2, n_float2)
```

解得 $1123573 \le n \le 2291248$，之前求得的 $n = 2097152$ 在这个范围之内。

(2)

问题:

用 IEEE 双精度浮点数计算 (1) 中前 n 项的和, 评估 IEEE 单精度浮点数计算结果的误差。

解答:

运行结果：

```
When n = 2097152, the harmonic series recorded in double is equal to
15.13330669507819.
The absolute error when recorded in float is 0.2703760137.
The relative error when recorded in float is 0.0178662879.
```

即绝对误差约为 0.27，相对误差约为 1.8%。

(3)

问题:

如果采用 IEEE 双精度浮点数，估计当 n 为何值时求和结果不再变化，这在当前做实验的计算机上大概需要多长的计算时间?

解答:

对于双精度浮点数，有 $\varepsilon_{mach} = 2^{-53}$。

运用以下 MATLAB 代码求解双精度浮点数的范围：

```
syms n_double1 n_double2
eqn_double1 = n_double1 * log(n_double1 - 1) == 2^54
eqn_double2 = n_double2 * (log(n_double2 - 1) + 1) == 2^53
n_double1 = solve(eqn_double1, n_double1)
n_double2 = solve(eqn_double2, n_double2)
```

解得 $263334173793273 \le n \le 531298735014385$。

将部分代码改为以下内容后，计算得 10^8 次循环需要 0.355 秒。

```
double harmonic_double = 0.0, tmp;
int n = 0;
while (n <= 100000000)
{
    tmp = harmonic_double;
    ++n;
    harmonic_double += one_double / n;
    if (harmonic_float == tmp)
        break;
}
```

最少需要 $263334173793273/100000000.0*0.355/60/60/24 \approx 10.82$ 天。

最多需要 $531298735014385/100000000.0*0.355/60/60/24 \approx 21.83$ 天。

实验计算机/编译配置:

CPU	GPU	内存	编译指令	编译器版本
i7-9570H 6核12线程	RTX2070	32G	g++	MinGW 8.1.0