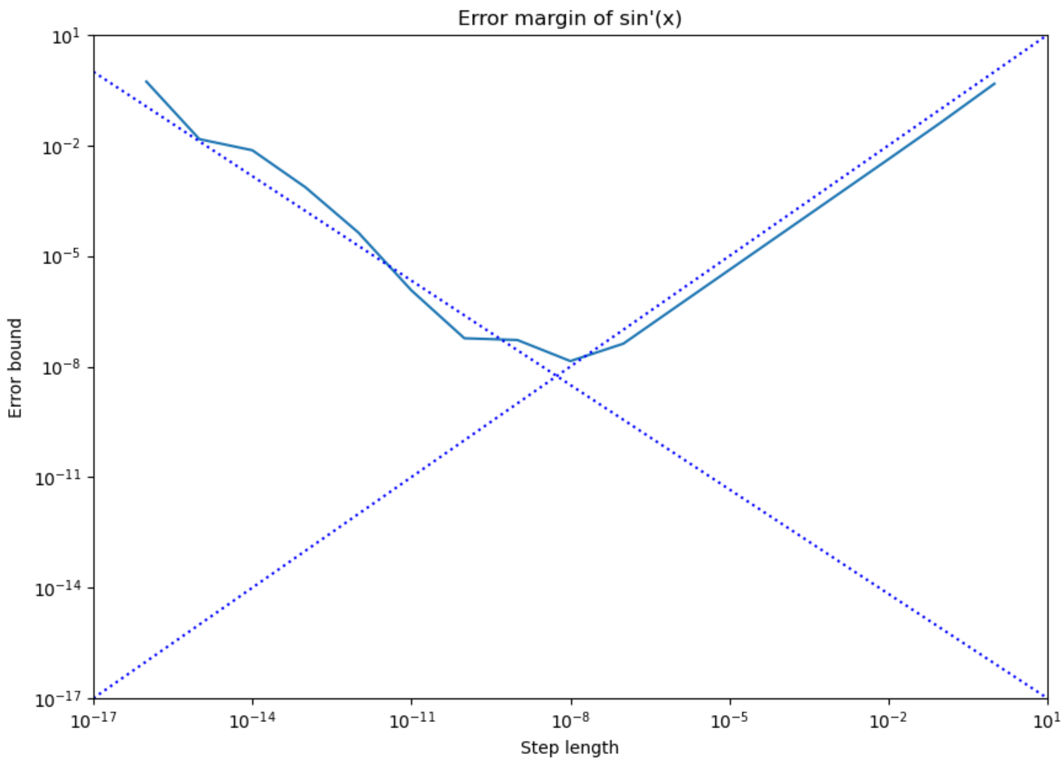
Experiment 1 Report

第一章上机题1

我们复现了课本例1.4，使用$\sin' 1 = \lim_{h\to 0} \frac{\sin(1+h)-\sin 1}{h}$计算了$\sin x$在$x = 1$处的导数及绝对误差，并绘制了图1-2：



其中所有的计算都使用双精度浮点数进行。可以看出，舍入误差（由浮点数精度导致）近似地与步长成反比，这是因为在计算式中，分子部分的舍入误差基本是个定值，因而误差与分母（步长）成反比；而截断误差基本与步长成正比，这是因为由泰勒展开，上式的截断误差约为$\frac{h^2}{2}\sin''(1)$，与步长二次方基本成正比。两种误差之和为总误差，在10^{-9}左右达到最小值。

第一章上机题3

我们在不同精度下计算了$\sum \frac{1}{n}$.

使用单精度浮点数计算

我们使用 `numpy.float32` 做单精度浮点数计算，逐个计算$\frac{1}{n}$的值并加到答案上，直到**当前答案与上一步的答案相等**为止。程序显示，计算到$n = 2\,097\,152$时结果不再发生变化，固定为15.403683，共花费973.95毫秒。

使用双精度浮点数计算

我们使用 `numpy.float_` 做双精度浮点数计算，对前$2\,097\,152$项进行求和，得到结果为15.133306，共花费739.86毫秒。假如认为双精度浮点数的计算结果为真实值，那么绝对误差为0.2704，相对误差约1.787%。可以看出，单精度浮点数在本任务中的误差是不可忽略的。

假如像上一部分那样在双精度下求和直到结果不发生变化，由于双精度浮点数的尾数部分位数大致是单精度浮点数的两倍（52+1 v.s. 23+1），因此需要的n大约为上述n的平方。程序运行时间因此估算为$739.86\ ms * 2097152 \approx 431$小时，可以认为是超过本问题计算限度的。

程序结果截图