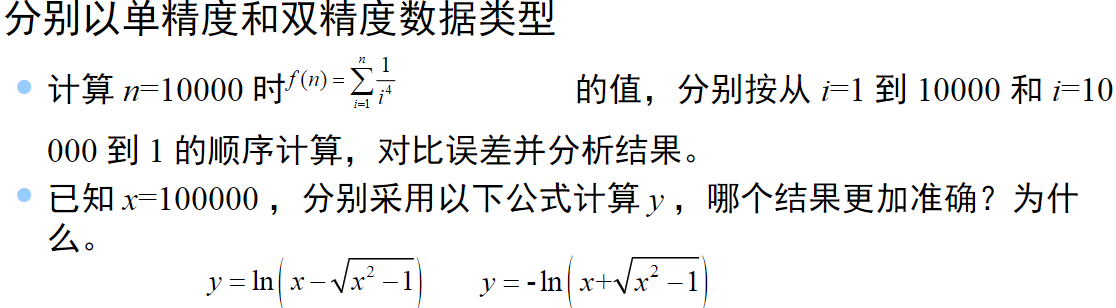
第一次上机作业

**问题描述：**



**问题分析：**

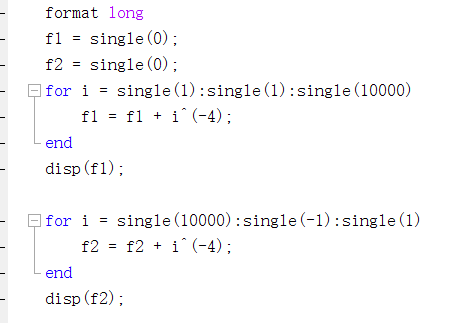
（问题1）：通过循环算法使得级数做和即可。误差分析方法，将有效位数为20位的更精确结果作为真值，通过公式百分误差 = |计算值-真值|/真值 \* 100%，可得误差，然后将上述获得的误差进行对比分析。

（问题2）：直接利用log函数和sqrt函数计算即可。

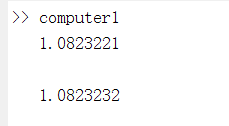
**Matlab程序：**

**问题一：**

单精度：



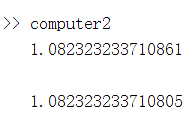
结果为：



双精度：

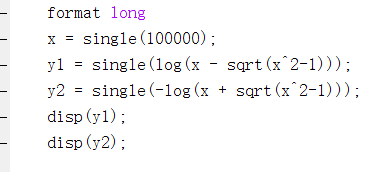
程序和单精度基本相同，将single类型换为double类型。

结果为：

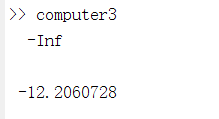


**问题二：**

单精度：



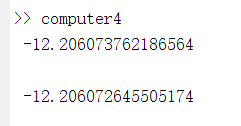
结果为：



双精度：

程序和单精度基本相同，将single类型换为double类型。

结果为：



**结果分析：**

**问题一：**

利用vpa（y，20）函数得到有效位数为20位更精确结果作为真值ans=1.0823232337108039669

|  |  |  |  |
| --- | --- | --- | --- |
| **精度** | **序号** | **当前计算值** | **相对误差** |
| **单精度** | **从1到10000** | 1.0823221 | 1.047478949532704e-04 |
| **从10000到1** | 1.0823232 | 3.114670634070485e-06 |
| **双精度** | **从1到10000** | 1.082323233710861 | 5.272497317652603e-12 |
| **从10000到1** | 1.082323233710805 | 1.025777688259261e-13 |

从i=1加到i=10000时，是大数加小数，会损耗掉比较多的有效位数。而从i=10000加到i=1，是小数加大数，误差较前一种方法更小。而双精度比单精度计算有效位数更多，误差更小。

**问题二：**

第二种方法更准确。因为100000相对于1来说非常大，第一种计算方法出现了两个相近数字的相减，会发生减性抵消，取入准确位数之后误差较大的部分位数，第二种方法中未出现减性抵消，所以比起第一种方法结果更加精确，

利用计算机直接计算得到-12.2060726455051737。和第二种方法比较接近。

**心得：**

这次的作业分别从大数加小数和减性抵消两个问题展示了舍入误差在利用MATLAB计算时会造成的影响。使我对计算机的有效位数和相对误差有了更深入的了解与直观的认识。在选择算法时要充分规避或减小舍入误差造成的影响，选择更好的算法。