**计算方法作业:求和差**

古宜民

PB17000002

**问题1**

使用C++编程计算，用float类型，结果如下表:

|  |  |  |
| --- | --- | --- |
| X | (x^2+4)^0.5-2 | (x^2)/((x^2+4)^0.5+2) |
| 1.250000000000e-01 | 3.902435302734e-03 | 3.902442753315e-03 |
| 1.562500000000e-02 | 6.103515625000e-05 | 6.103422492743e-05 |
| 1.953125000000e-03 | 9.536743164062e-07 | 9.536740890326e-07 |
| 2.441406250000e-04 | 0.000000000000e+00 | 1.490116119385e-08 |
| 3.051757812500e-05 | 0.000000000000e+00 | 2.328306436539e-10 |
| 3.814697265625e-06 | 0.000000000000e+00 | 3.637978807092e-12 |
| 4.768371582031e-07 | 0.000000000000e+00 | 5.684341886081e-14 |
| 5.960464477539e-08 | 0.000000000000e+00 | 8.881784197001e-16 |
| 7.450580596924e-09 | 0.000000000000e+00 | 1.387778780781e-17 |
| 9.313225746155e-10 | 0.000000000000e+00 | 2.168404344971e-19 |

而由mathematica计算得到的准确值如下：

结果分析：

可见由第一列两个相近数相减计算得到的结果远不如使用除法避开两相近数相减得到的结果。由于(x^2+4)^0.5在x很小时很接近2，直接相减造成了精度损失，应该避免这种情况。而利用相除避开了这一运算后，结果保持了应有的浮点数精度。

问题2

使用double类型对向量（数组）元素求和，结果如下：

|  |  |
| --- | --- |
| 顺序： | 1.0251881e-10 |
| 倒序： | -1.5643309e-10 |
| 正负分开计算： | 1.4068746e-12 |
| 准确结果 | 0 |

可见计算都出现了误差，正负分开计算得到的结果误差较小。但如果按数的绝对值大小从小到大相加，如按数组的4、2、0、3、1元素顺序相加，则得到了结果准确0.0。由于计算机中的浮点数按照有效数字部分和指数部分分开存储，大数加小数可能导致小数被“吞掉”而损失精度，对数组求和时按绝对值从小到大是较好的方法，这样能尽可能保证小数的相加准确性。