**《计算方法B》实验报告（一）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓 名 |  | 班 级 |  | **报告评分** |  |
| **学 号** |  | **地点/机号** | **数B316/No.63** | **指导教师** | **凌思涛** |
| **一、实验项目名称：算法的数值稳定性实验** | | | | | |
| **二、实验目的：实验并分析不同算法设计对数值稳定性的影响** | | | | | |
| **实验内容：P11 实验课题(一)**  **要求：（1）初值必须使用4位有效数字近似值；（2）对两种初值和两种迭代格式进行分析，哪种稳定？为什么稳定？** | | | | | |
| **四、程序设计**  **clear; clc;**  **% 定义 I\_n 列（1~10），转换为列向量**  **I\_n = (1:10)';**  **%% 针对 a = 0.05 的计算**  **a = 0.05;**  **method\_1\_min = zeros(10,1);**  **method\_2\_min = zeros(10,1);**  **for i = 1:length(I\_n)**  **n = I\_n(i);**  **method\_1\_min(i) = method1(a, n);**  **method\_2\_min(i) = method2(a, n);**  **end**  **% 将结果存放在表格中（各列均为 double 类型）**  **df\_min = table(I\_n, method\_1\_min, method\_2\_min, ...**  **'VariableNames', {'I\_n','method\_1','method\_2'});**  **%% 针对 a = 15 的计算**  **a = 15;**  **method\_1\_max = zeros(10,1);**  **method\_2\_max = zeros(10,1);**  **for i = 1:length(I\_n)**  **n = I\_n(i);**  **method\_1\_max(i) = method1(a, n);**  **method\_2\_max(i) = method2(a, n);**  **end**  **% 将结果存放在表格中（各列均为 double 类型）**  **df\_max = table(I\_n, method\_1\_max, method\_2\_max, ...**  **'VariableNames', {'I\_n','method\_1','method\_2'});**  **%% 显示结果**  **disp('a = 0.05')**  **disp(df\_min)**  **disp('--------------')**  **disp('a = 15')**  **disp(df\_max)**  **%% 提取变量并保存到 exp1.mat 文件**  **% 变量 A 为 df\_min 的 method\_1，B 为 df\_min 的 method\_2**  **% 变量 C 为 df\_max 的 method\_1，D 为 df\_max 的 method\_2**  **A = double(df\_min.method\_1);**  **B = double(df\_min.method\_2);**  **C = double(df\_max.method\_1);**  **D = double(df\_max.method\_2);**  **save('exp1.mat', 'A', 'B', 'C', 'D');**  **%% 辅助函数**  **function out = method1(a, n)**  **if n == 0**  **out = log((a + 1) / a);**  **else**  **out = -a \* method1(a, n - 1) + 1 / n;**  **end**  **end**  **function out = method2(a, n)**  **if a >= n / (n + 1)**  **out = (2 \* a + 1) / (2 \* a \* (a + 1) \* (n + 1));**  **else**  **out = 0.5 \* (1 / ((a + 1) \* (n + 1)) + 1 / n);**  **end**  **end** | | | | | |
| **五、实验结果（包含图表）**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 序号 | *a*=0.05 | | *a=*15 | | | 方案I | 方案II | 方案I | 方案II | | *I*1 | 0.847774 | 0.738095 | 0.031922 | 0.032292 | | *I*2 | 0.457611 | 0.40873 | 0.021167 | 0.021528 | | *I*3 | 0.310453 | 0.285714 | 0.015824 | 0.016146 | | *I*4 | 0.234477 | 0.220238 | 0.012633 | 0.012917 | | *I*5 | 0.188276 | 0.179365 | 0.010511 | 0.010764 | | *I*6 | 0.157253 | 0.151361 | 0.008999 | 0.009226 | | *I*7 | 0.134994 | 0.130952 | 0.007867 | 0.008073 | | *I*8 | 0.11825 | 0.11541 | 0.006988 | 0.007176 | | *I*9 | 0.105199 | 0.103175 | 0.006286 | 0.006458 | | *I*10 | 0.09474 | 0.09329 | 0.005706 | 0.005871 | | | | | | |
| **六、实验结果分析（实验总结、心得体会）**  对两种迭代方式使用评价函数  得到如下图像：  显然，方案一的迭代方式更加稳定可靠。这是由于因为它与积分的实际规模相符，且不会出现显著的舍入误差放大(只要 a 合理)。而方案二中，当很大时，对大部分 都成立，方案二就会采用的量级，可能比真实值偏小一个常数因子。   当很小时，就会频繁使用另一半公式，这对小可能还能对上，但随着增大也并不一定保持很好精度。  **心得体会：**  通过本次实验，，我认识到在数值计算中，直接递推格式往往更具稳定性与精度，能利用上一项结果准确迭代并避免误差放大。相反，另一种基于分段近似，只在部分参数下表现尚可，不能保证在所有区间都维持一致的精度和稳定性。同时，通过本次实验，提升了对matlab和python在计算方法相关方面的使用技能。 | | | | | |

**注：如果报告超过1页，需双面打印。**