

长 沙 理 工 大 学

数学与统计学院

实 验 报 告

(第二版)

**课 程 名 称**

**任 课 教 师**

**实验项目名称**

**实 验 类 型**

**实 验 日 期**

**班 级**

**学 号**

**姓 名**

**实验报告填写说明**

**（实验项目名称、实验项目类型必须与实验教学大纲保持一致）**

**1．实验环境**：

详细书写实验使用的硬件和软件环境。

**2．实验目的**：

根据实验教学大纲，写出实验的要求和目的。

**3．实验原理：**

简要说明本实验项目所涉及的理论知识。

**4．实验内容**：

这是实验报告极其重要的容。对于验证性验，要写清楚操作方法，需要经过哪几个步骤来实现其操作。对于设计性和综合性实验，还应写出设计思路和设计方法。对于创新性实验，还应注明其创新点。

**5．实验结论：**

根据实验过程中得到的结果，做出结论。

**6．实验总结：**

本次实验的收获、体会和建议。

**7．指导教师评语及成绩：**

指导教师依据学生的实际报告内容，给出本次实验报告的评价和成绩。

实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1 实验环境**  硬件：Core i7-8750H  软件：Matlab 2020a  **2 实验目的**  掌握程序语言的基本知识，能够编写简单程序；熟练掌握用迭代法和牛顿法求非线性方程问题  **3实验原理**  迭代法和牛顿法基本原理  迭代法格式：  Xk+1 = φ(Xk); k = 0,1,...... 初值X0  牛顿法格式  设f(x)在[a,b]上有二阶导数，且满足：   1. f(a)f(b)<0; (2)f1(x)≠0，x∈[a,b];(3）f11(x)不变号，x∈[a,b]; 2. 初始值x0∈[a,b],使f11(x)f(x)>0.   产生的序列{xn}收敛于f(x)=0在[a,b]上的唯一实根。  **4实验内容**  方案一：  1、验证迭代法求解教材p145例题6.3 求方程在附近的根 ；  2、牛顿法求解教材P147例题6.4 求方程在取初值的根  方案二：  1、迭代函数对收敛性的影响  实验题目：用迭代法求方程的在初值的根。  实验方案：分别选取迭代函数为和  求解。迭代10次，20次分析比较迭代函数选取的不同对收敛性的影响。  2、初值的选取对迭代法的影响  实验题目：用牛顿法求的根。  实验方案：分别取初值和，用牛顿法迭代10次。  分析比较初值的选取对迭代法的影响，观察比较并分析原因。  **5实验结论**  方案一  1.迭代法求解结果如下：    结果取六位数字的情况下与例题所给结果完全吻合，与 完全相同，即所求根。  2.牛顿法求解结果如下：    在迭代到第三次时精度满足要求，得到所求根  方案二  1.迭代函数对收敛性的影响  初值，迭代函数为，迭代10次的结果如下;    右略，明显此迭代函数发散，并不适合用来求根。  初值，迭代函数为，迭代20次的结果同样发散，如下：    显然同样不适合求根  初值，迭代函数为，迭代10次的结果如下：    显然选次迭代函数的情况下为收敛的，在取六位数字的情况下在第8次即得到了所求的根。  初值，迭代函数为，迭代20次的结果如下：    同上，显然选次迭代函数的情况下为收敛的，在取六位数字的情况下在第8次即得到了所求的根。  2.初值的选取对迭代法的影响  初值，迭代10次的结果如下：    初值，迭代10次的结果如下：    可见初值时，方程可以在四次迭代求出所求的根，但在时，方程即便迭代10次也得不到我们所求的值，且大相径庭。可见初值的选取直接影响了求根时的运算次数与速度。  **6实验总结（收获、体会和建议）**  迭代法的迭代函数的选取直接影响了其收敛性，不同的选择在同样初值的情况下会有不同的结果，初值的选取则影响了计算的次数，合理的初值可以有效减少计算量。  牛顿法是先设一初值，这个初值完全可以任意给定。虽然，问题的最终结果与初值无关，但初值选择估计得当，则计算步骤会得到简化。  **7指导教师评语及成绩** | | | | | |
| **评 语** | **评语等级** | | | | |
| **优** | **良** | **中** | **及格** | **不及格** |
| **实验方案设计的合理程度** |  |  |  |  |  |
| **实验结论的记录情况** |  |  |  |  |  |
| **实验总结情况** |  |  |  |  |  |
| **实验报告是否按时完成,书写是否规范（文字叙述，层次结构）** |  |  |  |  |  |
| **成 绩：**    **指导教师签名：**    **批阅日期：** | | | | | |

**附录1：源 程 序**

|  |
| --- |
| **主函数**  clc;clear  format long  % 方案一  [precision\_I,x\_kI] = Iteration(@(x)(x+1)^(1/3),1.5,8);  for ii = 1:8  fprintf('方案一 迭代法第%d次 方程的根x=%.5f;误差=%.5f\n',ii,x\_kI(ii),precision\_I(ii));  end  [k\_NP,x\_kNP] = Newton\_pre(@(x)x\*exp(x)-1,0.5,1.0e-5);  for np = 1:max(k\_NP)  fprintf('方案一 牛顿法第%d次 方程的根x=%.5f;\n',k\_NP(np),x\_kNP(np));  end  % 方案二  [precision\_I,x\_kI] = Iteration(@(x)2\*x^3-1,0,10);  for ii = 1:10  fprintf('方案二 (1)迭代10次 迭代法第%d次 方程的根x=%.5f;误差=%.5f\n',ii,x\_kI(ii),precision\_I(ii));  end  [precision\_I,x\_kI] = Iteration(@(x)2\*x^3-1,0,20);  for ii = 1:20  fprintf('方案二 (1)迭代20次 迭代法第%d次 方程的根x=%.5f;误差=%.5f\n',ii,x\_kI(ii),precision\_I(ii));  end  [precision\_I,x\_kI] = Iteration(@(x)((x+1)/2)^(1/3),0,10);  for ii = 1:10  fprintf('方案二 (2)迭代10次 迭代法第%d次 方程的根x=%.5f;误差=%.5f\n',ii,x\_kI(ii),precision\_I(ii));  end  [precision\_I,x\_kI] = Iteration(@(x)((x+1)/2)^(1/3),0,20);  for ii = 1:20  fprintf('方案二 (2)迭代20次 迭代法第%d次 方程的根x=%.5f;误差=%.5f\n',ii,x\_kI(ii),precision\_I(ii));  end  [precision\_N,x\_kN] = Newton(@(x)x^3-x-1,1.5,10);  for nn = 1:10  fprintf('方案二 x0=1.5 牛顿法 第%d次 方程的根x=%.5f;误差=%.5f\n',nn,x\_kN(nn),precision\_N(nn));  end  [precision\_N,x\_kN] = Newton(@(x)x^3-x-1,0,10);  for nn = 1:10  fprintf('方案二 x0=0 牛顿法 第%d次 方程的根x=%.5f;误差=%.5f\n',nn,x\_kN(nn),precision\_N(nn));  end  **迭代法**  function[precision,x\_k] = Iteration(fun,x0,n)  % 迭代法 已知函数，初值，迭代次数  x(1) = x0;  for i = 1:n  k(i) = i; % 迭代次数  x\_k(i) = fun(x(i));  x(i+1) = x\_k(i);  precision(i) = abs(x\_k(i)-x(i));  while(precision(i)==0)  printf('当前迭代次数k=%d;方程根x=%.6f',k(i),x\_k(i));  return;  end  end  end  function [k\_all,x\_kall] = Iteration\_pre(fun,x0,precision)  % 迭代法，已知函数，初值以及精度  k = 0; % 迭代次数  x = x0;  x\_k = fun(x);  while abs(x\_k-x)>precision  k\_all(k) = k;  x\_kall(k) = x\_k;  k = k+1;  x = x\_k;  x\_k = fun(x);  if k>99  disp('没有找到根');  return  end  end  end  **牛顿法**  function[precision,x\_k] = Newton(fun,x0,n)  % 牛顿法 已知函数，初值，迭代次数  syms x  y = fun(x);  dif = matlabFunction(diff(y));  x(1) = x0;  for i = 1:n  k(i) = i; % 迭代次数  f = fun(x(i));  df = dif(x(i));  x\_k(i) = x(i)-f/df;  x(i+1) = x\_k(i);  precision(i) = abs(x\_k(i)-x(i));  while(precision(i)==0)  printf('当前迭代次数k=%d;方程根x=%.6f',k(i),x\_k(i));  return;  end  end  end  function [k\_all,x\_kall] = Newton\_pre(fun,x0,precision)  % 牛顿法 已知函数，初值，精度  syms x  y = fun(x);  dif = matlabFunction(diff(y));  k = 0;  x = x0;  f = fun(x);  df = dif(x);  x\_k = x-f/df;  while abs(x\_k-x)>precision  k = k+1;  k\_all(k) = k;  x\_kall(k) = x\_k;  x = x\_k;  f = fun(x);  df = dif(x);  x\_k = x-f/df;  if k>999  disp('没有找到根');  return  end  end  end |