



计算机与软件工程学院

上机报告

**（ 2021/2021 学年 第 1 学期 ）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **数值计算（C++）** | | | | | |
| 课程代码 | **190901319** | | | | | |
| 上机时间 | 2021 | 年 | 11 | 月 | 11 | 日 |
| 指导单位 | 物联网工程系 | | | | | |
| 任课教师 | 李显勇 | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学生姓名 | | 李子涵 | | | | | | |
| 学 号 | | 3120190971401 | | | | | | |
| 成 绩 | |  | | 年级专业 | | | 计算机科学与技术 | |
| **实验名称** | **【实验3】Newton迭代法，弦截法** | | | | | **实验地点** | | 8-322 | |
| **实验类型** | **验证** | | **实验学时** | | **2** | **实验日期** | | 2021-11-11 | |
| **实验目的和要求**   1. 了解**Newton迭代法**基本理论，编程实现该算法； 2. 了解**弦截法**基本理论，编程实现该算法； 3. 提交以下文档（用**学号+姓名**作为文件夹名，将所有内容放于该文件夹中）：   1）实验报告  2）程序代码 | | | | | | | | | |
| **实验环境(实验设备)**  windows XP及以上版本；PC；Matlab7.0及以上版本；其它高级语言 | | | | | | | | | |
| **实验原理及内容**  **一、实验原理**  根据**Newton迭代法，弦截法**相关知识和算法编程完成本实验  **二、实验内容**    求方程*x3-7.7x2+19.2x-15.3*=0在*x0*=*1*附近的根。   1. 用Newton迭代法求解（精度10-9）； 2. 用弦截法求解（精度10-9）。   **三、实验过程（可以文字说明+运行结果截图）**  1.实验代码  '''  文件名：西华大学数值计算C++实验报告——实验三  当前版本：1.0  完成作者：李子涵  学号：3120190971401  完成日期：2021.11.11  '''  def f(x):      """      函数表达式      """      y=0      for i in coef.keys():          y+=pow(x,i)\*coef[i]      return y  def f\_d(x):      """      求导数      """      y=0      for i in coef.keys():          if(i!=0):              y+=pow(x,i-1)\*coef[i]\*i      return y  import math  MAXREPT=1000000  def Newton\_iter(x\_k0,epsilon):      def g(x):          y=x-f(x)/f\_d(x)          return y      for i in range(1,MAXREPT):#最大迭代数          x\_k1=g(x\_k0)          if(abs(x\_k1-x\_k0)<epsilon):              print("牛顿迭代法近似解为:",end="")              print(x\_k1)              return          x\_k0=x\_k1      print("牛顿迭代法在x0附近无根")  def String(x\_k0,x\_k1,epsilon):      """      弦截法求根      """      #初始值      x\_k=x\_k0#x\_k      x\_k\_min1=x\_k1#x\_k-1      for i in range(1,MAXREPT):#最大迭代数          x\_k\_add1=x\_k-f(x\_k)\*(x\_k-x\_k\_min1)/(f(x\_k)-f(x\_k\_min1))          if(abs(x\_k\_add1-x\_k)<epsilon or abs(f(x\_k\_add1)<epsilon)):#当精度达到时，输出解              print("弦截法近似解为:",end="")              print(x\_k\_add1)              return          x\_k\_min1=x\_k#否则继续循环，更新当前xk和xk-1的值          x\_k=x\_k\_add1      print("弦截法在x0附近无根")  if \_\_name\_\_=='\_\_main\_\_':      #多项式格式为系数在x前，次数在x后,且已合并S所有同类项      coef=dict(zip([3,2,1,0],[1,-7.7,19.2,-15.3]))      Newton\_iter(1,pow(10,-9))      String(1.5,4.0,pow(10,-9))  2.实验结果： | | | | | | | | | |

|  |
| --- |
| **四、实验小结**（包括问题和解决方法、心得体会、意见与建议等）  此章节的实现过程较为简单，仅需要用到循环迭代设计代码即可。唯一的问题式在设计存储多项式方程的数据结构时我首先采用了较为复杂的方法，设计十分繁琐，后来改为以Python中的字典来进行存储，多项式的每一项次数与各项系数一一对应，如此可以快速且方便地索引到每一项，当然此种实现方式只适合于较短不复杂，好化简的多项式。当多项式过长，且需要化简式，此种设计将耗费过多人力检查多项式系数的正确性。 |