中北大学大数据学院实验报告

**课程名称： 数值分析 实验类型： 综合型**

**实验题目：**

班级：17070142学号：1707004716姓名：王浩 指导教师：**韩慧妍** 成绩：

|  |  |  |  |
| --- | --- | --- | --- |
| 实验地点 | 虚拟仿真实验室 | 实验时间 | 2019.12.6 |
| 1．实验目的  理解数据插值和拟合应用场景，能够根据数据特点，正确选择算法，并且能够编程实验相关算法。 | | | |
| 2. 实验任务  战斗机机翼外形根据工艺要求由一组数据(x,y)给出，用程控铣床加工对每一刀只能沿x方向和y方向切割很小的一步，因此需要从已知数据得到满足加工所要求的步长很小的每一步坐标。  下表给出的x,y数据位于机翼断面的下轮廓上，假设需要得到x坐标每改变0.1时的y坐标，试完成加工所需要数据，并画出曲线。   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | x | 0 | 3 | 5 | 7 | 9 | 11 | 12 | 13 | 14 | 15 | | Y | 0 | 1.2 | 1.7 | 2.0 | 2.1 | 2.0 | 1.8 | 1.2 | 1.0 | 1.6 | | | | |
| 3. 相关知识  将所用到的数据使用matplotlib进行可视化，发现其数据离散且不均匀排列，使用多项式回归可能造成阶数很大，加重算法复杂度。而使用插值法可以避免该现象的发生。在使用牛顿插值时，从插值函数上看，会出现龙格现象，因此讨论采用三次样条插值来拟合该样本数据，最终在测试数据上得到了很好的效果。  **牛顿插值**  通过对牛顿插值的一次插值和二次插值分析可归纳可得：用n阶多项式对n+1个数据点进行拟合，n阶多项式为：，与线性和二次插值一样，可以用已知数据点计算系数：    一阶有差商通常表示为,二阶有限差商表示的是两个一阶差商的差分：,同理，n阶差商表示为：，将以上系数带入到前式当中，得到下面的插值多项式：  上式成为牛顿插值多项式。可以看出高阶差分是通过低阶差分实现来计算的。    **三次样条插值**  为了解决牛顿插值法带来的龙格现象问题，我们还可以使用多个低阶多项式分别对数据点的各个子集进行拟合。而三次样条的目标是在每两个相邻数据点之间的区间上推导一个如下三阶多项式：,对于n+1个数据点一共需要n个区间，所以要计算4n个未知数。需要4n个条件计算这些未知数：  （1）在内部结点处的函数值必须相等（xn-2个条件）；  （2）第一个和最后一个函数必须通过端点（2个条件）；  （3）内部结点处的二阶导数必须相等（n-1个条件）；  （4）内部结点处的一阶导数必须相等（n-1个条件）；  （5）端点处的二阶导数为0（2个条件）；  求出对应结果建立三次样条。 | | | |
| 4. 实验内容与实验结果（要求画出两种方法的流程图，并给出实验运行结果图，要求程序结果先输出个人姓名和学号，现输出实验结果，程序提交电子版即可）    图1拉格朗日插值流程图 图2牛顿插值流程图    图3、4 牛顿插值法拟合曲线-易出现龙格现象  牛顿插值斜对角线差商：[0.0, 0.39999999999999997, -0.029999999999999992, 0.000714285714285714, -7.936507936507932e-05, 7.215007215007158e-06, -7.215007215007185e-06, -3.3560189810189933e-06, 5.07387784173499e-06, -2.0806326609898058e-06]  +---+-----+--------+-------+---------+---------+---------+---------+---------+---------+---------+  | x | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |  +---+-----+--------+-------+---------+---------+---------+---------+---------+---------+---------+  | y | 0.0 | -4.944 | -8.82 | -11.773 | -13.931 | -15.412 | -16.319 | -16.746 | -16.775 | -16.479 |  +---+-----+--------+-------+---------+---------+---------+---------+---------+---------+---------+  +---+---------+---------+---------+---------+---------+--------+--------+--------+--------+--------+  | x | 1.0 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 |  +---+---------+---------+---------+---------+---------+--------+--------+--------+--------+--------+  | y | -15.924 | -15.165 | -14.252 | -13.229 | -12.131 | -10.99 | -9.833 | -8.681 | -7.554 | -6.465 |  +---+---------+---------+---------+---------+---------+--------+--------+--------+--------+--------+  +---+--------+--------+--------+--------+--------+--------+--------+-------+-------+-------+  | x | 2.0 | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | 2.6 | 2.7 | 2.8 | 2.9 |  +---+--------+--------+--------+--------+--------+--------+--------+-------+-------+-------+  | y | -5.427 | -4.448 | -3.535 | -2.692 | -1.922 | -1.225 | -0.602 | -0.05 | 0.432 | 0.847 |  +---+--------+--------+--------+--------+--------+--------+--------+-------+-------+-------+  +---+-----+-------+-------+-------+-------+-------+-------+-------+-------+-------+  | x | 3.0 | 3.1 | 3.2 | 3.3 | 3.4 | 3.5 | 3.6 | 3.7 | 3.8 | 3.9 |  +---+-----+-------+-------+-------+-------+-------+-------+-------+-------+-------+  | y | 1.2 | 1.494 | 1.734 | 1.925 | 2.071 | 2.176 | 2.247 | 2.287 | 2.301 | 2.293 |  +---+-----+-------+-------+-------+-------+-------+-------+-------+-------+-------+  +---+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+  | x | 4.0 | 4.1 | 4.2 | 4.3 | 4.4 | 4.5 | 4.6 | 4.7 | 4.8 | 4.9 |  +---+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+  | y | 2.267 | 2.226 | 2.176 | 2.117 | 2.054 | 1.989 | 1.924 | 1.862 | 1.802 | 1.748 |  +---+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+  +---+-----+-------+-------+-------+-------+------+-------+-------+-------+-------+  | x | 5.0 | 5.1 | 5.2 | 5.3 | 5.4 | 5.5 | 5.6 | 5.7 | 5.8 | 5.9 |  +---+-----+-------+-------+-------+-------+------+-------+-------+-------+-------+  | y | 1.7 | 1.659 | 1.625 | 1.599 | 1.581 | 1.57 | 1.568 | 1.572 | 1.584 | 1.602 |  +---+-----+-------+-------+-------+-------+------+-------+-------+-------+-------+  +---+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+  | x | 6.0 | 6.1 | 6.2 | 6.3 | 6.4 | 6.5 | 6.6 | 6.7 | 6.8 | 6.9 |  +---+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+  | y | 1.625 | 1.653 | 1.685 | 1.721 | 1.759 | 1.799 | 1.84 | 1.882 | 1.922 | 1.962 |  +---+-------+-------+-------+-------+-------+-------+------+-------+-------+-------+  +---+-----+-------+-------+-------+-------+-------+-------+-------+-------+-----+  | x | 7.0 | 7.1 | 7.2 | 7.3 | 7.4 | 7.5 | 7.6 | 7.7 | 7.8 | 7.9 |  +---+-----+-------+-------+-------+-------+-------+-------+-------+-------+-----+  | y | 2.0 | 2.036 | 2.069 | 2.098 | 2.125 | 2.148 | 2.167 | 2.182 | 2.193 | 2.2 |  +---+-----+-------+-------+-------+-------+-------+-------+-------+-------+-----+  +---+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+  | x | 8.0 | 8.1 | 8.2 | 8.3 | 8.4 | 8.5 | 8.6 | 8.7 | 8.8 | 8.9 |  +---+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+  | y | 2.204 | 2.204 | 2.201 | 2.195 | 2.186 | 2.175 | 2.162 | 2.148 | 2.132 | 2.116 |  +---+-------+-------+-------+-------+-------+-------+-------+-------+-------+-------+  +---+-----+-------+-------+-------+-------+-------+-------+-------+-----+-------+  | x | 9.0 | 9.1 | 9.2 | 9.3 | 9.4 | 9.5 | 9.6 | 9.7 | 9.8 | 9.9 |  +---+-----+-------+-------+-------+-------+-------+-------+-------+-----+-------+  | y | 2.1 | 2.084 | 2.068 | 2.053 | 2.039 | 2.027 | 2.016 | 2.007 | 2.0 | 1.994 |  +---+-----+-------+-------+-------+-------+-------+-------+-------+-----+-------+  +---+------+-------+-------+-------+------+-------+-------+-------+------+-------+  | x | 10.0 | 10.1 | 10.2 | 10.3 | 10.4 | 10.5 | 10.6 | 10.7 | 10.8 | 10.9 |  +---+------+-------+-------+-------+------+-------+-------+-------+------+-------+  | y | 1.99 | 1.988 | 1.988 | 1.989 | 1.99 | 1.993 | 1.995 | 1.998 | 2.0 | 2.001 |  +---+------+-------+-------+-------+------+-------+-------+-------+------+-------+  +---+------+-------+-------+-------+------+-------+-------+-------+-------+-------+  | x | 11.0 | 11.1 | 11.2 | 11.3 | 11.4 | 11.5 | 11.6 | 11.7 | 11.8 | 11.9 |  +---+------+-------+-------+-------+------+-------+-------+-------+-------+-------+  | y | 2.0 | 1.997 | 1.991 | 1.983 | 1.97 | 1.954 | 1.933 | 1.907 | 1.877 | 1.841 |  +---+------+-------+-------+-------+------+-------+-------+-------+-------+-------+  +---+------+-------+-------+-------+-------+-------+-------+-------+------+-------+  | x | 12.0 | 12.1 | 12.2 | 12.3 | 12.4 | 12.5 | 12.6 | 12.7 | 12.8 | 12.9 |  +---+------+-------+-------+-------+-------+-------+-------+-------+------+-------+  | y | 1.8 | 1.754 | 1.704 | 1.648 | 1.589 | 1.527 | 1.462 | 1.396 | 1.33 | 1.264 |  +---+------+-------+-------+-------+-------+-------+-------+-------+------+-------+  +---+------+------+-------+-------+-------+-------+-------+-------+-------+-------+  | x | 13.0 | 13.1 | 13.2 | 13.3 | 13.4 | 13.5 | 13.6 | 13.7 | 13.8 | 13.9 |  +---+------+------+-------+-------+-------+-------+-------+-------+-------+-------+  | y | 1.2 | 1.14 | 1.085 | 1.036 | 0.996 | 0.966 | 0.946 | 0.939 | 0.945 | 0.966 |  +---+------+------+-------+-------+-------+-------+-------+-------+-------+-------+  +---+------+-------+------+-------+-------+-------+-------+------+-------+-------+  | x | 14.0 | 14.1 | 14.2 | 14.3 | 14.4 | 14.5 | 14.6 | 14.7 | 14.8 | 14.9 |  +---+------+-------+------+-------+-------+-------+-------+------+-------+-------+  | y | 1.0 | 1.048 | 1.11 | 1.182 | 1.263 | 1.348 | 1.432 | 1.51 | 1.571 | 1.605 |  +---+------+-------+------+-------+-------+-------+-------+------+-------+-------+  表5 牛顿插值法拟合曲线-x每变化0.1时y的变化（x在0-3之间龙格现象严重）    图6、7 三次样条插值法拟合曲线-库函数实现    +------+-----------------------+  | x | y |  +------+-----------------------+  | 0.0 | 1.558063475203094e-17 |  | 0.1 | 0.049861178936089774 |  | 0.2 | 0.09899651581999246 |  | 0.3 | 0.14741091358880629 |  | 0.4 | 0.19510927517962942 |  | 0.5 | 0.24209650352956005 |  | 0.6 | 0.2883775015756965 |  | 0.7 | 0.3339571722551368 |  | 0.8 | 0.3788404185049792 |  | 0.9 | 0.42303214326232186 |  | 1.0 | 0.4665372494642631 |  | 1.1 | 0.5093606400479012 |  | 1.2 | 0.5515072179503341 |  | 1.3 | 0.59298188610866 |  | 1.4 | 0.6337895474599771 |  | 1.5 | 0.673935104941384 |  | 1.6 | 0.7134234614899786 |  | 1.7 | 0.7522595200428589 |  | 1.8 | 0.7904481835371232 |  | 1.9 | 0.8279943549098698 |  | 2.0 | 0.8649029370981973 |  | 2.1 | 0.9011788330392032 |  | 2.2 | 0.9368269456699858 |  | 2.3 | 0.9718521779276437 |  | 2.4 | 1.006259432749275 |  | 2.5 | 1.0400536130719773 |  | 2.6 | 1.07323962183285 |  | 2.7 | 1.10582236196899 |  | 2.8 | 1.1378067364174962 |  | 2.9 | 1.1691976481154667 |  | 3.0 | 1.1999999999999997 |  | 3.1 | 1.2302186950081935 |  | 3.2 | 1.259858636077146 |  | 3.3 | 1.2889247261439554 |  | 3.4 | 1.3174218681457202 |  | 3.5 | 1.3453549650195384 |  | 3.6 | 1.372728919702508 |  | 3.7 | 1.399548635131728 |  | 3.8 | 1.4258190142442957 |  | 3.9 | 1.4515449599773096 |  | 4.0 | 1.4767313752678684 |  | 4.1 | 1.5013831630530694 |  | 4.2 | 1.5255052262700113 |  | 4.3 | 1.5491024678557919 |  | 4.4 | 1.57217979074751 |  | 4.5 | 1.5947420978822637 |  | 4.6 | 1.616794292197151 |  | 4.7 | 1.6383412766292698 |  | 4.8 | 1.6593879541157193 |  | 4.9 | 1.6799392275935965 |  | 5.0 | 1.6999999999999997 |  | 5.1 | 1.719574454178747 |  | 5.2 | 1.7386638926005296 |  | 5.3 | 1.757268897642758 |  | 5.4 | 1.775390051682844 |  | 5.5 | 1.7930279370981974 |  | 5.6 | 1.8101831362662302 |  | 5.7 | 1.826856231564352 |  | 5.8 | 1.8430478053699741 |  | 5.9 | 1.8587584400605073 |  | 6.0 | 1.8739887180133623 |  | 6.1 | 1.8887392216059502 |  | 6.2 | 1.903010533215682 |  | 6.3 | 1.9168032352199682 |  | 6.4 | 1.9301179099962191 |  | 6.5 | 1.9429551399218463 |  | 6.6 | 1.9553155073742605 |  | 6.7 | 1.9671995947308718 |  | 6.8 | 1.978607984369092 |  | 6.9 | 1.9895412586663315 |  | 7.0 | 2.0000000000000004 |  | 7.1 | 2.0099834882768195 |  | 7.2 | 2.019485793520737 |  | 7.3 | 2.0284996832850126 |  | 7.4 | 2.0370179251229055 |  | 7.5 | 2.0450332865876724 |  | 7.6 | 2.052538535232573 |  | 7.7 | 2.0595264386108663 |  | 7.8 | 2.0659897642758107 |  | 7.9 | 2.071921279780664 |  | 8.0 | 2.0773137526786845 |  | 8.1 | 2.082159950523132 |  | 8.2 | 2.086452640867264 |  | 8.3 | 2.090184591264339 |  | 8.4 | 2.093348569267617 |  | 8.5 | 2.0959373424303545 |  | 8.6 | 2.0979436783058114 |  | 8.7 | 2.0993603444472457 |  | 8.8 | 2.1001801084079164 |  | 8.9 | 2.1003957377410813 |  | 9.0 | 2.0999999999999996 |  | 9.1 | 2.0989915927139795 |  | 9.2 | 2.0973929333165255 |  | 9.3 | 2.0952323692171944 |  | 9.4 | 2.092538247825539 |  | 9.5 | 2.089338916551115 |  | 9.6 | 2.085662722803479 |  | 9.7 | 2.0815380139921844 |  | 9.8 | 2.0769931375267867 |  | 9.9 | 2.0720564408168407 |  | 10.0 | 2.066756271271902 |  | 10.1 | 2.061120976301525 |  | 10.2 | 2.055178903315265 |  | 10.3 | 2.048958399722677 |  | 10.4 | 2.042487812933316 |  | 10.5 | 2.0357954903567372 |  | 10.6 | 2.0289097794024955 |  | 10.7 | 2.021859027480146 |  | 10.8 | 2.0146715819992433 |  | 10.9 | 2.0073757903693434 |  | 11.0 | 2.0 |  | 11.1 | 1.9924444797050296 |  | 11.2 | 1.984097183915291 |  | 11.3 | 1.9742179884659021 |  | 11.4 | 1.9620667691919833 |  | 11.5 | 1.9469034019286529 |  | 11.6 | 1.9279877625110302 |  | 11.7 | 1.9045797267742344 |  | 11.8 | 1.8759391705533848 |  | 11.9 | 1.8413259696836004 |  | 12.0 | 1.8000000000000005 |  | 12.1 | 1.7515664266355728 |  | 12.2 | 1.6970115719147865 |  | 12.3 | 1.637667047459977 |  | 12.4 | 1.574864464893483 |  | 12.5 | 1.5099354358376404 |  | 12.6 | 1.4442115719147857 |  | 12.7 | 1.379024484747258 |  | 12.8 | 1.3157057859573928 |  | 12.9 | 1.2555870871675279 |  | 13.0 | 1.2 |  | 13.1 | 1.150089813752678 |  | 13.2 | 1.1062565284255637 |    | 13.2 | 1.1062565284255637 |  | 13.3 | 1.0687138216941887 |  | 13.4 | 1.0376753712340854 |  | 13.5 | 1.0133548547207865 |  | 13.6 | 0.9959659498298246 |  | 13.7 | 0.9857223342367326 |  | 13.8 | 0.9828376856170427 |  | 13.9 | 0.9875256816462876 |  | 14.0 | 1.0 |  | 14.1 | 1.0204743183537128 |  | 14.2 | 1.0491623143829578 |  | 14.3 | 1.0862776657632678 |  | 14.4 | 1.1320340501701756 |  | 14.5 | 1.1866451452792135 |  | 14.6 | 1.2503246287659155 |  | 14.7 | 1.323286178305812 |  | 14.8 | 1.4057434715744366 |  | 14.9 | 1.4979101862473216 |  表8 三次样条插值法拟合曲线-x每变化0.1时y的变化（拟合效果较好）  **牛顿插值python代码**  **import** numpy **as** np **import** matplotlib.pyplot **as** plt *# 解决中文显示问题* plt.rcParams[**'font.sans-serif'**] = [**'KaiTi'**] *# 指定默认字体* plt.rcParams[**'axes.unicode\_minus'**] = **False** *# 解决保存图像是负号'-'显示为方块的问题* **from** prettytable **import** PrettyTable title = PrettyTable() *# 定义表头   # 给出的数据存放* plane = np.array([  [0, 3, 5, 7, 9, 11, 12, 13, 14, 15],  [0, 1.2, 1.7, 2.0, 2.1, 2.0, 1.8, 1.2, 1.0, 1.6] ])  *# 获取n阶差商* **def** get\_quotient(xList,yList):  count = 1  result = [yList[0]] *# 存放差商表对角线的第一个元素* **while** len(yList) > 1:  list = []  **for** i **in** range(len(yList)-1):  *# 计算差商* commerce = (yList[i+1] - yList[i]) / (xList[i+count] - xList[i])  list.append(commerce) *# 添加到每列差商列表* result.append(list[0]) *# 只保存写斜对角线我们所用到的差商* count += 1 *# 距离值加1* yList = list *# 更新yList列表* **return** result  *# 获取第i阶差商的因子 # 根据n阶差商因子和差商求函数* **def** get\_w(xList,yList):  list = get\_quotient(xList, yList)  *# 定义关于x的Wi函数* **def** Wi(x):  result = 0.0  **for** j **in** range(len(xList)):  w\_i = list[j] *# 赋值计算后的差商* **for** k **in** range(0,j): *# 计算因子  # 与每上一项的因子相乘* w\_i = w\_i \* (x - xList[k])  result += w\_i *# 结果相加* **return** result *# 返回关于x的因子* **return** Wi *# 返回对应函数* **if** \_\_name\_\_ == **'\_\_main\_\_'**:  plt.scatter(plane[0], plane[1], color=**'red'**, label=**'原始数据标记点'**)  plt.title(**"战斗机机翼外形工艺数据-牛顿插值"**)  plt.xlabel(**"x"**)  plt.ylabel(**"y"**)  Wi = get\_w(plane[0],plane[1]) *# 获取计算得到的* line = np.arange(0,15,0.01)  plt.plot(line,Wi(line),label=**'插值拟合曲线'**)  plt.legend() *# 进行渲染* plt.grid() *# 绘制格线* plt.show()  print(**"牛顿插值斜对角差商："**,end=**""**)  print(get\_quotient(plane[0],plane[1]))  **for** j **in** range(0,15):  title.add\_column(**"x"**, [**"y"**])  xi = np.arange(j,j+1,0.1)  **for** i **in** xi:  title.add\_column(str(round(i,1)),[round(Wi(i),3)])  print(title)  title.clear()  **三次样条插值python代码**  **import** numpy **as** np **import** matplotlib.pyplot **as** plt *# 解决中文显示问题* plt.rcParams[**'font.sans-serif'**] = [**'KaiTi'**] *# 指定默认字体* plt.rcParams[**'axes.unicode\_minus'**] = **False** *# 解决保存图像是负号'-'显示为方块的问题* **from** scipy **import** interpolate **from** prettytable **import** PrettyTable title = PrettyTable([**"x"**,**"y"**]) *# 定义表头  # 存放给出的数据* plane = np.array([  [0, 3, 5, 7, 9, 11, 12, 13, 14, 15],  [0, 1.2, 1.7, 2.0, 2.1, 2.0, 1.8, 1.2, 1.0, 1.6] ])  plt.scatter(plane[0], plane[1], color=**'red'**, label=**'原始数据标记点'**) plt.title(**"战斗机机翼外形工艺数据-三次样条插值"**) plt.xlabel(**"x"**) plt.ylabel(**"y"**) *# 求一维曲线的样条插值* t = interpolate.splrep(plane[0],plane[1]) *# x数据点* xi = np.arange(0,15,0.01) yi = interpolate.splev(xi,t) plt.plot(xi,yi,color=**'blue'**,label=**'三次样条插值'**) plt.legend() *# 进行渲染* plt.grid() *# 画出表格* x\_test = np.arange(0,15,0.1) y\_test = interpolate.splev(xi,t) **for** i **in** range(0,len(x\_test)):  title.add\_row([round(x\_test[i],1),y\_test[i]]) print(title) plt.show() | | | |
| 5．总结  根据实验要求，编写了牛顿插值、三次样条插值程序，求出已知数据的插值函数S（x）并同时做出插值拟合图像，方便观察与分析实验结果。  在此次试验当中，尝试使用了拉格朗日、牛顿插值法、三次样条插值法对数据进行插值求拟合函数，从插值后的曲线上来看，插值的次数过高会在曲线两端造成龙格现象，导致最终数据不准确。接着使用了分段插值-三次样条插值法对数据进行拟合，达到了非常好的效果。通过本次实验，让我对三次样条插值有一定的认识，能对结果进行分析，从图像分析得到信息，让我认识到插值的现实运用意义。 | | | |