《数值分析》课程实验报告

实验名称 插值与拟合

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **班级** | 信息1701 | **姓名** | 艾春辉 | **学号** | 201710000101 | **序号** |  |
| **教师** | 赵美玲 | **地点** | 数学实验中心 | | | **评分** |  |
| 1. 实验目的 2. 掌握多项式插值法的基本思路和步骤； 3. 了解整体插值的局限性及分段插值的基本思想。 4. 掌握最小二乘法拟合的基本原理和方法； 5. 培养运用计算机模拟解决问题的能力。 6. 实验原理   **三次样条插值**  **1、应用背景**  利用插值多项式计算被插函数在区间上任意一点的近似值，总希望截断误差的绝对值尽可能小，而影响截断误差的因素主要是插值多项式的次数和插值节点的选取，理论和实践都说明高次插值是不可取的，一个典型的例子就是Runge现象。因此通常在实际计算中采用分段低次（次数）插值，并且选择均匀分布在附近的节点作为插值节点，但是得到的分段低次插值多项式至多是连续的，光滑性较差，不能够满足工业造型设计对函数光滑性的要求。为了克服上述困难，可以使用分段三次样条插值来逼近被插函数。  **2、算法描述**  设插值节点为，插值函数，且在上是三次多项式，称之为三次样条插值函数。  记，  **1． 三转角法**  若已知，则在区间上的表达式为  （1）  因此，只要可以确定，则可求得。  由等式，可以得到关于的个方程  （2）    再结合边界条件  a）  即  （3）  或  b）  即  （4）  其中  得到关于的各方程，该方程称为**三转角方程**，利用追赶法解出，由表达式（1）可以计算上任意一点的函数值。  **2．三弯矩法**  同理也可以假定，则在区间上的表达式为  （5）  利用等式可得  （6）    再结合边界条件  a）  即  （7）  或  b）  即  （8）  得到关于的各方程，该方程称为**三弯矩方程**，利用追赶法解出，由表达式（5）可以计算上任意一点的函数值。 3.3.2 数据拟合 **最小二乘法基本原理**  已知数据对，求多项式    使得为最小，这就是多项式拟合的最小二乘法。  **最小二乘法的算法描述**  线性函数为例，拟合给定数据。  算法描述：  步骤1：输入值，及。  步骤2：建立法方程组其中  。  步骤3：解法方程组。  步骤4：输出   1. 实验过程和结果  3.4.1 多项式插值 1． 给定C:\Users\yt\AppData\Local\Temp\ksohtml\wps3803.tmp.png构造插值多项式计算C:\Users\yt\AppData\Local\Temp\ksohtml\wps3814.tmp.png。   1. 编程实现拉格朗日插值，并计算结果。 2. 将计算结果和查表结果进行比较。 3. 区间C:\Users\yt\AppData\Local\Temp\ksohtml\wps3815.tmp.png作等距划分： C:\Users\yt\AppData\Local\Temp\ksohtml\wps3816.tmp.png，以C:\Users\yt\AppData\Local\Temp\ksohtml\wps3817.tmp.png（C:\Users\yt\AppData\Local\Temp\ksohtml\wps3827.tmp.png）为节点对函数C:\Users\yt\AppData\Local\Temp\ksohtml\wps3828.tmp.png进行插值逼近。（**分别取**C:\Users\yt\AppData\Local\Temp\ksohtml\wps3829.tmp.png） 4. 用多项式插值对C:\Users\yt\AppData\Local\Temp\ksohtml\wps382A.tmp.png进行逼近，并在同一坐标系下作出函数的图形，进行比较。写出插值函数对C:\Users\yt\AppData\Local\Temp\ksohtml\wps383B.tmp.png的逼近程度与节点个数的关系，并分析原因。 5. 试用分段插值（任意选取）对C:\Users\yt\AppData\Local\Temp\ksohtml\wps383C.tmp.png进行逼近，在同一坐标下画出图形，观察分段插值函数对C:\Users\yt\AppData\Local\Temp\ksohtml\wps383D.tmp.png的逼近程度与节点个数的关系。  3.4.2 数据拟合 1．已知一组数据如下，求它的线性拟合曲线。   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | C:\Users\yt\AppData\Local\Temp\ksohtml\wps383E.tmp.png | 1 | 2 | 3 | 4 | 5 | | C:\Users\yt\AppData\Local\Temp\ksohtml\wps384F.tmp.png | 4 | 4.5 | 6 | 8 | 8.5 |  1. 编程实现最小二乘算法，并画出其拟合曲线 2. 求出其平方误差    1. 已知一组数据如下，求其拟合曲线。  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | C:\Users\yt\AppData\Local\Temp\ksohtml\wps3850.tmp.png | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | C:\Users\yt\AppData\Local\Temp\ksohtml\wps3851.tmp.png | 2 | 3 | 4 | 7 | 8 | 10 | 11 | 14 | 16 | 18 | 19 | | C:\Users\yt\AppData\Local\Temp\ksohtml\wps3852.tmp.png | 106.42 | 108.2 | 109.5 | 110 | 109.93 | 110.49 | 110.59 | 110.6 | 110.76 | 111 | 111.2 |  1. 求以上数据形如C:\Users\yt\AppData\Local\Temp\ksohtml\wps3853.tmp.png的拟合曲线，及其平方误差。   （2） 求以上数据形如C:\Users\yt\AppData\Local\Temp\ksohtml\wps3863.tmp.png的拟合曲线，及其平方误差。  通过画出（1）（2）的图形，观察结果并结合其平方误差，写出你对数据拟合的认识热图润。  答：   1. （1）拉格朗日插值test1.py文件   运行结果：    由结果可知与真实值很接近   1. (1)多项式插值逼近程序:test2.py   运行结果：    分析：高次插值的稳定性较差，导致插值函数的两端剧烈震荡，但是低次对于较大区间的逼近精度又不够，所以拟合效果不好。   1. 采用分段线性插值test3.py   运行结果：    结点个数分别为5,10,15,20，由图可见对于分段线性插值来说，结点个数越多插值效果越好   1. （1）线性最小二乘法:test4.py   运行结果    拟合效果非常好    (2)第一小问代码：test5.py  运行结果：      件对于多项式的拟合曲线，得到的结果是，误差2.7796；第二小问代码见test6.py  运行结果：      对于指数函数形式的拟合，我们先将函数取对数，令Z=lny,t=1/x.得到函数z=lna+bt.进行拟合，得到的结果是,误差0.4719，可见，指数形式的拟合函数更加与原数据走势符合，但是不管哪种形式的拟合都不能保证点都在拟合曲线上。拟合可以表现出数据的真实走势，而且，通过拟合可以排除一些点的误差。   1. 思考题分析解答   1、整体插值有何局限性？如何避免？  2、基函数的选择对拟合的结果有何影响？  3、简述数据拟合与插值的异同。  4、试着编程实现三次样条插值（三弯矩法和三转角法任选其一），并对函数进行插值逼近（**分别取**C:\Users\yt\AppData\Local\Temp\ksohtml\wps26D7.tmp.png）。  1、随着节点的加密，采用高次的多项式插值，当n增大时，由于高次的多项式插值不稳定，插值函数在两端会发生激烈的震荡。而低次插值对于区间长度较大的逼近精度不够，解决的方法是采用分段低次代数插值。  2、选择不一样的基函数，曲线的拟合效果不同，根据经验选择的基函数拟合出来结果较好。  3、拟合是给定节点，找到一个已知未知参数形式的函数来最大限度地逼近这些点，可以依据最小二乘原则，即找到使整体的误差达到最小的参数，拟合不一定过点；  而插值是找到一个连续曲面来穿过这些点，插值一定要过点。  他们的共同点都是通过已知一些离散点集M上的约束，求取一个定义在连续集合S(M包含于S)的未知连续函数，从而达到获取整体规律的目的。   1. 代码test7.py     重点难点分析   1. 重点：掌握插值和拟合的不同，在什么情况下用什么插值方法效果较好，拟合要基于经验来选取拟合的基函数。 2. 难点：本实验难点在于如何通过C#来画函数图，通过查找资料，给代码增加了控件，得以解决。 | | | | | | | |