# 上机题第七题实验报告

王伟任 计33 2013011333

1. **题目要求及分析**

**第六章上机题8**： 已知直升飞机旋转机翼外形曲线的采样点坐标，以及两端点的1阶导数值，利用第一种边界条件的三次样条插值函数计算翼型曲线在x=2, 30, 130, 350, 515各点上的函数值及1阶导数、2阶导数的近似值。

三次样条插值函数是一种分段多项式插值，计算各点函数值时需要首先确定x所属的区间，进而应用该区间上的函数公式计算函数值及导数值。

1. **实验结果及分析**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 函数值 | 1阶导数值 | 2阶导数值 |
| 2 | 7.8251553424 | 1.5568351436 | -0.2212596805 |
| 30 | 25.3862347614 | 0.3548743912 | -0.0078427101 |
| 130 | 37.2138405828 | -0.0103918878 | -0.0013821190 |
| 350 | 22.4751112814 | -0.1077842647 | -0.0002302644 |
| 515 | 0.5427133607 | -0.0899061742 | 0.0081197328 |

直接在命令行中输出的结果如下：

input an X value:2

ft = 7.8251553424; df = 1.5568351436; ddf = -0.2212596805.

input an X value:30

ft = 25.3862347614; df = 0.3548743912; ddf = -0.0078427101.

input an X value:130

ft = 37.2138405828; df = -0.0103918878; ddf = -0.0013821190.

input an X value:350

ft = 22.4751112814; df = -0.1077842647; ddf = -0.0002302644.

input an X value:515

ft = 0.5427133607; df = -0.0899061742; ddf = 0.0081197328.

1. **实验代码**

采用C++语言实现。

**#include <cstdio>**

**#include <cmath>**

**int n;**

**int main(){**

**FILE\* fp = fopen("test.txt", "r");**

**fscanf(fp, "%d", &n);**

**fgetc(fp);**

**double\* x = new double[n];**

**double\* f = new double[n];**

**for (int i = 0; i < n; i++)**

**fscanf(fp, "%lf", &x[i]), fgetc(fp);**

**for (int i = 0; i < n; i++)**

**fscanf(fp, "%lf", &f[i]), fgetc(fp);**

**double df0, dfn;**

**fscanf(fp, "%lf", &df0);**

**fgetc(fp);**

**fscanf(fp, "%lf", &dfn);**

**fclose(fp);**

**double\* h = new double[n - 1];**

**for (int i = 0; i < n - 1; i++)**

**h[i] = x[i + 1] - x[i];**

**double\* u = new double[n - 1];**

**for (int i = 0; i < n - 2; i++)**

**u[i] = h[i] / (h[i] + h[i + 1]);**

**u[n - 2] = 1;**

**double\* l = new double[n - 1];**

**for (int i = 1; i < n - 1; i++)**

**l[i] = h[i] / (h[i - 1] + h[i]);**

**l[0] = 1;**

**double\* d = new double[n];**

**for (int i = 1; i < n - 1; i++)**

**d[i] = 6 \* (f[i-1] / (h[i-1] \* (h[i-1] + h[i])) + f[i+1] / (h[i] \* (h[i-1] + h[i])) - f[i] / (h[i-1] \* h[i]));**

**d[0] = 6 \* ((f[1] - f[0]) / h[0] - df0) / h[0];**

**d[n-1] = 6 \* (dfn - (f[n-1] - f[n-2]) / h[n-2]) / h[n-2];**

**double\* array = new double[n];**

**for (int i = 0; i < n; i++) array[i] = 2;**

**double\* m = new double[n - 1];**

**for (int i = 2; i <= n ; i++){**

**m[i - 2] = u[i - 2] / array[i - 2];**

**array[i - 1] = array[i - 1] - m[i - 2] \* l[i - 2];**

**d[i - 1] = d[i - 1] - m[i - 2] \* d[i - 2];**

**}**

**double\* M = new double[n];**

**M[n - 1] = d[n - 1] / array[n - 1];**

**for (int i = n - 2; i >= 0; i--)**

**M[i] = (d[i] - l[i] \* M[i + 1]) / array[i];**

**double tx;**

**printf("input an X value:");**

**scanf("%lf", &tx);**

**while(tx != -1){**

**int kk = 0;**

**for (int i = 0; i < n - 1; i++)**

**if ((tx >= x[i]) && (tx <= x[i + 1])){**

**kk = i;**

**break;**

**}**

**double df, ddf, ft;**

**ft = M[kk]\*(x[kk+1]-tx)\*(x[kk+1]-tx)\*(x[kk+1]-tx)/(6\*h[kk])+M[kk+1]\*(tx-x[kk])\*(tx-x[kk])\*(tx-x[kk])/(6\*h[kk])+(f[kk]-M[kk]\*h[kk]\*h[kk]/6)\*(x[kk+1]-tx)/h[kk]+(f[kk+1]-M[kk+1]\*h[kk]\*h[kk]/6)\*(tx-x[kk])/h[kk];**

**df = -M[kk]\*(x[kk+1]-tx)\*(x[kk+1]-tx)/(2\*h[kk])+M[kk+1]\*(tx-x[kk])\*(tx-x[kk])/(2\*h[kk])+(f[kk+1]-f[kk])/h[kk]-(M[kk+1]-M[kk])\*h[kk]/6;**

**ddf = M[kk]\*(x[kk+1]-tx)/h[kk]+M[kk+1]\*(tx-x[kk])/h[kk];**

**printf("ft = %.10f; df = %.10f; ddf = %.10f.\n", ft, df, ddf);**

**printf("input an X value:");**

**scanf("%lf", &tx);**

**}**

**delete[] m;**

**delete[] M;**

**delete[] array;**

**delete[] d;**

**delete[] h;**

**delete[] l;**

**delete[] u;**

**delete[] x;**

**delete[] f;**

**return 0;**

**}**