

# 第四章 3次样条插值

## 1.三次插值通用程序

A. 构建正规方程组左端矩阵

public double[,] CalculateA(double[] x,double[] y)

{

int len = x.Length;

double[,] arrray1 = new double[len,len+1];

for (int i = 0; i < len; i++)

{

for (int j = 0; j < len; j++)

{

arrray1[i,j] = 0;

}

}

arrray1[0, 1] = 1;

arrray1[len - 1, len - 2] = 1;

for (int i = 0; i < len;i++ )

{

arrray1[i, i] = 2;

}

for (int i = 1; i < len - 1;i++ )

{

double u = 1.0 \* (x[i] - x[i - 1]) / (x[i + 1] - x[i - 1]);

double w = 1 - u;

arrray1[i, i - 1] = u;

arrray1[i, i + 1] = w;

}

return arrray1;

}

B．计算右端向量d

public double[] CalculateD(double[] x,double[] y,double y\_0,double y\_n)

{

int len = x.Length;

double[] d = new double[len];

d[0] = 6 \* ((y[1] - y[0]) / (x[1] - x[0]) - y\_0) / (x[1] - x[0]);

d[len - 1] = 6 \* (y\_n - (y[len-1] - y[len-2]) / ((x[len - 1] - x[len - 2]))) / (x[len - 1] - x[len - 2]);

for (int i = 1; i < len - 1;i++ )

{

d[i] = 6 \* ((y[i] - y[i - 1]) / (x[i] - x[i - 1]) - (y[i+1] - y[i]) / (x[i+1] - x[i])) / (x[i+1] - x[i-1]);

}

return d;

}

C．追赶法计算M

public double[] CalculateM(double[,] A,double[] d)

{

// 追赶法

int len = d.Length;

double[] b = new double[len];

double[] y = new double[len];

double[] l = new double[len];

double[] M = new double[len];

b[0] = A[0,0];

y[0] = d[0];

for (int i = 1; i < len; i++) //消元过程

{

l[i] = A[i,i - 1] / b[i - 1];

b[i] = A[i,i] - l[i] \* A[i - 1,i];

y[i] = d[i] - l[i] \* y[i - 1];

}

M[len - 1] = y[len - 1] / b[len - 1];

for (int i = len - 2; i >= 0; i--)

{

M[i] = (y[i] - A[i,i + 1] \* M[i + 1]) / b[i];

}

return M;

}

D.输出S(X)

private void PrintFourthResult()

{

richTextBox4.Text = "";

double[] x = new double[] {0,1,2,3,4,5,6,7,8,9,10 };

double[] y = new double[] { 2.51,3.30,4.04,4.70,5.22,5.54,5.78,5.40,5.57,5.70,5.80};

double y\_0 = 0.8;

double y\_n = 0.2;

double[] M = fourth.CalculateSx(x, y, y\_0, y\_n);

double r;

int len = x.Length;

richTextBox4.Text += "S(x)的结果如下:";

richTextBox4.Text += "\r";

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

{

r=(y[i+1]-y[i])/(x[i+1]-x[i])-(M[i]\*(1.0/3)+M[i+1]\*(1.0/6))\*(x[i+1]-x[i]);

string text = "在区间（"+x[i]+","+x[i+1]+"内)：S(x)="+y[i]+"+("+r+")\*(x-"+x[i]+")+("+0.5\*M[i]+")\*(x-"+x[i]+")^2+("+1/(6\*(x[i+1]-x[i]))\*(M[i+1]-M[i])+")\*(x-"+x[i]+")^3";

richTextBox4.Text += text;

richTextBox4.Text += "\r";

}

richTextBox4.Text += "S(i+0.5)的结果如下:";

richTextBox4.Text += "\r";

double []S=new double [len];

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

{

r=(y[i+1]-y[i])/(x[i+1]-x[i])-(M[i]/3+M[i+1]/6)\*(x[i+1]-x[i]);

S[i]=y[i]+r\*0.5+0.5\*M[i]\*(0.5)\*(0.5)+1/(6\*(x[i+1]-x[i]))\*(M[i+1]-M[i])\*0.5\*0.5\*0.5;

string text = "S["+(i+0.5)+"]="+S[i];

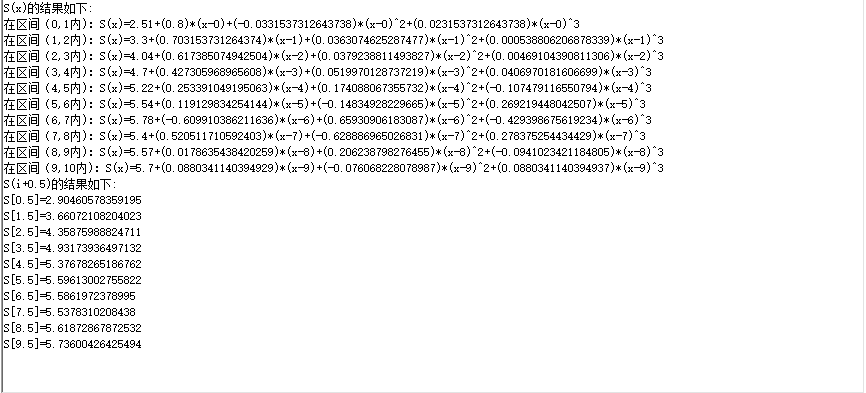
richTextBox4.Text += text;

richTextBox4.Text += "\r";

}

}

## 2.实验结果



## **3.心得**

通过编写此次程序，进一步加深了对三次样条插值的理解。

# 第五章 重积分

## 1.重积分通用程序

public double TanSumofSquare(double x, double y)

{

double result = 0.0d;

result = Math.Tan(x\*x + y\*y);

return result;

}

public double GetIntergal(double a,double b,double c,double d,double h,double k)

{

double result = 0.0d;

result = h \* k / 4.0 \* (TanSumofSquare(a, c) + TanSumofSquare(a, d) + TanSumofSquare(b, c) + TanSumofSquare(b, d));

return result;

}

public String CalculateResult(double a, double b, double c, double d, double ToleranceError)

{

double h, k; // 步长

h = b - a;

k = d - c;

int m, n;

m = 1;

n = 1;

double Tf = 0;

double Tfnew = 0;

Tfnew = GetIntergal(a, b, c, d, h, k);

Tf = Tfnew + 1;

// 计算过程中不同阶数精度数值

double[] T0 = new double[100];

double[] T1 = new double[100];

double[] T2 = new double[100];

double[] T3 = new double[100];

int index = 0;

while ((Math.Abs(Tf-Tfnew)>ToleranceError) && m < 10000)

{

h = (b - a) / m;

k = (d - c) / n;

Tf = Tfnew;

double temp=0,temp1,temp2,temp3,temp4,temp5,temp6;

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n;j++ )

{

temp1 = a + 1.0 \* (i + 0.5) \* h; // X(i+1/2)

temp2 = c + 1.0 \* (j + 0.5) \* k; // Y(i+1/2)

temp3 = a + 1.0 \* i \* h; // Xi

temp4 = c + 1.0 \* j \* k; // Yi

temp5 = a + 1.0 \* (i + 1) \* h; // X(i+1)

temp6 = c + 1.0 \* (j + 1) \* k; // Y(i+1)

temp += TanSumofSquare(temp1, temp4) + TanSumofSquare(temp3, temp2) + TanSumofSquare(temp5, temp2)

+ TanSumofSquare(temp1, temp6) + 2\*TanSumofSquare(temp1, temp2);

}

}

temp = 1.0\* temp \* h \* k / 8;

Tfnew = Tf / 4 + temp;

T0[index] = Tfnew;

index++;

n = n \* 2;

m = m \* 2;

}

// 计算更高阶精度数值

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

{

T1[i] = 4.0 / 3 \* (T0[i + 1]) - 1.0 / 3 \* (T0[i]);

}

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

{

T2[i] = 16.0 / 15 \* (T1[i + 1]) - 1.0 / 15 \* (T1[i]);

}

for (int i = 0; i < (index - 3); i++)

{

T3[i] = 64.0 / 63 \* (T2[i + 1]) - 1.0 / 63 \* (T2[i]);

}

String result = "";

result = "T0:\r";

for(int i=0;i<index;i++)

{

result += T0[i] + "\r";

}

result += "T1:\r";

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

{

result += T1[i] + "\r";

}

result += "T2:\r";

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

{

result += T2[i] + "\r";

}

result += "T3:\r";

for (int i = 0; i < index-3; i++)

{

result += T3[i] + "\r";

}

return result;

}

## 2.实验结果

|  |  |  |  |
| --- | --- | --- | --- |
| T0 | T1 | T2 | T3 |
| 0.519796458502325 | 0.344032422801714 | 0.337393306711976 | 0.336579708876358 |
| 0.387973431726867 | 0.337808251467585 | 0.336592421342539 | 0.336522702775241 |
| 0.350349546532405 | 0.336668410725355 | 0.336523792127855 | 0.336520476014649 |
| 0.340088694677117 | 0.336532830790199 | 0.336520527828918 | 0.336520433446865 |
| 0.337421796761928 | 0.336521296763998 | 0.336520434921585 | 0.33652043302844 |
| 0.336746421763481 | 0.336520488786735 | 0.336520433058021 | 0.336520433025955 |
| 0.336576972030922 | 0.336520436541065 | 0.336520433026456 | 0.336520433025941 |
| 0.33653457041353 | 0.336520433246119 | 0.33652043302595 |  |
| 0.336523967537972 | 0.33652043303971 |  |  |
| 0.336521316664275 |  |  |  |

## **3.心得**

通过编写程序，自学了重积分相关知识，进一步理解了数值精度相关问题。