



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

《计算方法》课程工程应用设计

极限挑战

挑战，不是为着征服自然，而是为着突破自我，超越自我
生命有极限，思想无极限，高度有极限，境界无极限

作品名称: 三次样条插值模拟实际页岩气管道设计

学 院: 航空学院

团队队长: 冯铮浩 2017300281 01011704 班

联系方式: 13815233646

团队成员: 卢佐 2017300354 01011708 班

姜丽升 2017301123 01011704 班

单湘淋 2017300346 01011708 班

(注: 刘军老师班级学生)

秦嘉琛 2017300357 01081701 班

戴世城 2017300353 01011708 班

指导老师: 陶 亮

西北工业大学

2019 年 5 月 4 日

目录

1 问题提出与实际工程背景	3
2 开发工具	5
3 程序主要功能	5
4 程序设计思路	5
5 程序文件与工程名称	6
6 函数模块声明及功能	6
7 程序小型说明书	8
8 样例演示	12
9 程序完整源代码	15
10 团队分工及程序开发总结	27
11 参考文献	28

1 问题提出与实际工程背景

页岩气是指赋存于富有机质泥页岩及其夹层中，以吸附或游离状态为主要存在方式的非常规天然气，成分以甲烷为主，是一种清洁、高效的能源资源。我国富有机质页岩分布广泛，南方地区、华北地区 and 新疆塔里木盆地等发育海相页岩，华北地区、准噶尔盆地、吐哈盆地、鄂尔多斯盆地、渤海湾盆地和松辽盆地等广泛发育陆相页岩，具备页岩气成藏条件，资源潜力大。据专家预测，页岩气可采资源量为 25 万亿方，超过常规天然气资源。一种页岩气管道与油气藏分布示意图如下图 1、2 所示。[1]



图 1 一种常见页岩气管道实景图

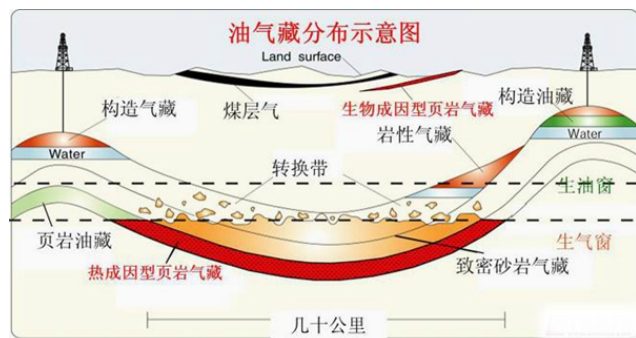


图 2 典型油气藏分布示意图

近几年，美国页岩气勘探开发技术突破，产量快速增长，对国际天然气市场及世界能源格局产生重大影响，世界主要资源国都加大了对页岩气的勘探开发力度。因此，我国也将页岩气开发提上重要发展议程。但页岩气管道常由于岩层结构复杂与地下探测受限，因而其设计与施工难度较大。

三次样条插值 (Cubic Spline Interpolation) 简称 **Spline 插值**，是通过一系列控制点的一条光滑曲线，数学上通过求解三弯矩方程组得出曲线函数数组的过程。早期工程师制图时，把富有弹性的细长木条（所谓样条）用压铁固定在样点上，在其他地方让它自由弯曲，然后沿木条画下曲线。成为样条曲线。[2] 三次样条曲线因具有分段光滑度高，修改简便等优点，

故在实际工程中应用广泛（一种示意图如图 3 所示）。对于一些数据点较多的插值设计问题，常常采用分段插值方法。采用分段线性插值和分段二次插值，可以构造一个整体连续的函数，而采用分段三次样条插值，则可以构造一个整体上具有一阶连续导数的插值函数。

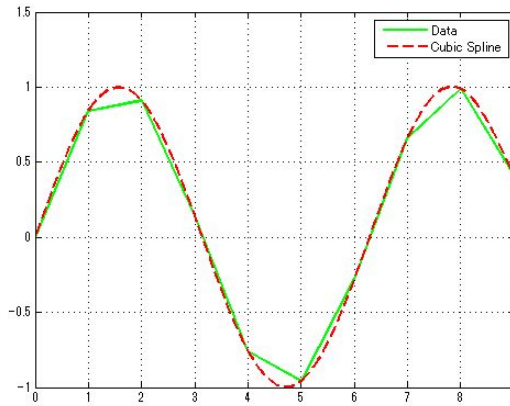


图 3 三次样条插值曲线示意图

在设计实际的页岩气管道形态时，通常只有一系列探测已知的离散数据点，我们构想可以采用三次样条插值方法，根据数据点，构造连续光滑的曲线，进而模拟连续的管道函数，以满足施工需要。同时，由管道弯曲与所受弯矩的近似关系式，即

$$K(x) = \frac{1}{\rho(x)} = \frac{M(x)}{EI_z} \quad (1)$$

可以得出弯矩 M 与曲线曲率 K 近似成正比关系，因此在设计管道过程中，需要特别注意每段三次样条曲线上的曲率最大位置，以保证管道满足刚度条件，防止发生结构破裂现象。

2 开发工具

Code::Blocks

Windows 10 64 位操作系统

MATLAB 2017a

3 程序主要功能

本程序可以利用三次样条插值方法模拟工程实际的页岩气管道设计。
具体功能如下：

- (1) 可根据手动输入或文件导入（已有或随机）的平面岩层数据点坐标，利用三次样条插值方法构造连续光滑的曲线，模拟并设计优良的页岩气管道形态，并输出详细数据文件结果；
- (2) 扫描计算得到各段区间上样条曲线的最大曲率及其坐标，为实际工程中需考虑的管道所受力与弯矩情况提供数据支持；
- (3) 最后利用 MATLAB 软件作出含原始插值节点控制点、三次样条管道曲线及分段区间曲率最大位置点的图像，以供程序验证参考。

4 程序设计思路

本程序遵循模板化编程思想，采用自顶向下的总体设计思路，分层次进行编写，并采用交互式界面设计，方便用户使用查看。

程序总体分为数据导入、数据处理、数据导出、绘制图像四部分。

数据导入部分，采用手动或自动（文件）输入方式，增加了程序的通用性，能够适应不同规模的数据资源。

数据处理部分，利用 **switch** 分支语句，实现函数构造过程中不同边界条件的设置。将方程计算、矩阵运算、曲率计算等模块进行子函数封装，方便查阅错误与主程序调用。

数据导出部分，采用控制台或文件输出方式，将结果分类保存，使格式清晰，方便用户阅读与分析。

绘制图像部分，采用转接 **MATLAB** 的跨软件设计方法，将需要共同使用的文件命名一致，便于调取与调试。

5 程序文件与工程名称

工程实践挑战题——三次样条插值模拟实际页岩气管道设计：

main.cpp: C++主函数程序文件

三次样条插值模拟实际页岩气管道设计.**cbp**: C++工程文件

Curve_Painting.m: **MATLAB** .m 文件

6 函数模块声明及功能

第一部分（界面初始化、数据文件导入）

(1) **void menu ();**

函数实现功能“菜单界面的设置”。

(2) void data_intro ();

函数实现功能“坐标点数据输入”，算法如下：

输出主菜单界面，通过switch语句中的 case 分支选择数据输入方式。

第二部分（三次样条曲线核心功能实现）

(1) void spline_construction ();

各段三次样条差值曲线构造，根据不同的输入格式、边界条件构造三次样条插值函数的线性方程组，并调用 cal_m (int n)函数求解方程组得到插值函数。

(2) double f (int x1,int x2,int x3);

求差分函数(含三个参数)。

(3) void cal_m (int n);

用追赶法求解出弯矩向量 M，采用追赶法求解三对角方程组，先进行三角 LU 分解，再带入求解。

(4) void cal_curvature (double a,double b,double c,double d,double x1,double x2,int m);

输入三次多项式函数系数，区间端点，区间编号。采用数值方法求已知三次函数在某段区间内的最大曲率。取步长为 h，逐步扫描各点处的曲率，进而求出最大曲率，并记录最大曲率所在坐标。

第三部分（数据文件导出）

(1) void data_output_console (int n);

函数实现功能“在控制台中输出各段三次样条插值结果”，算法如

下：调用 C++ 中的库函数，使用输出流函数将结果写入控制台中。

(2) void data_output_file (int n);

函数实现功能“在文件中输出各段三次样条插值结果”，算法如下：

调用 C++ 中的库函数，使用输出流函数将结果写入文件中。

7 程序小型说明书

首先打开 Code::Blocks 构建编译程序并运行程序，控制台界面将出现下图所示界面框（如图 4 所示）；

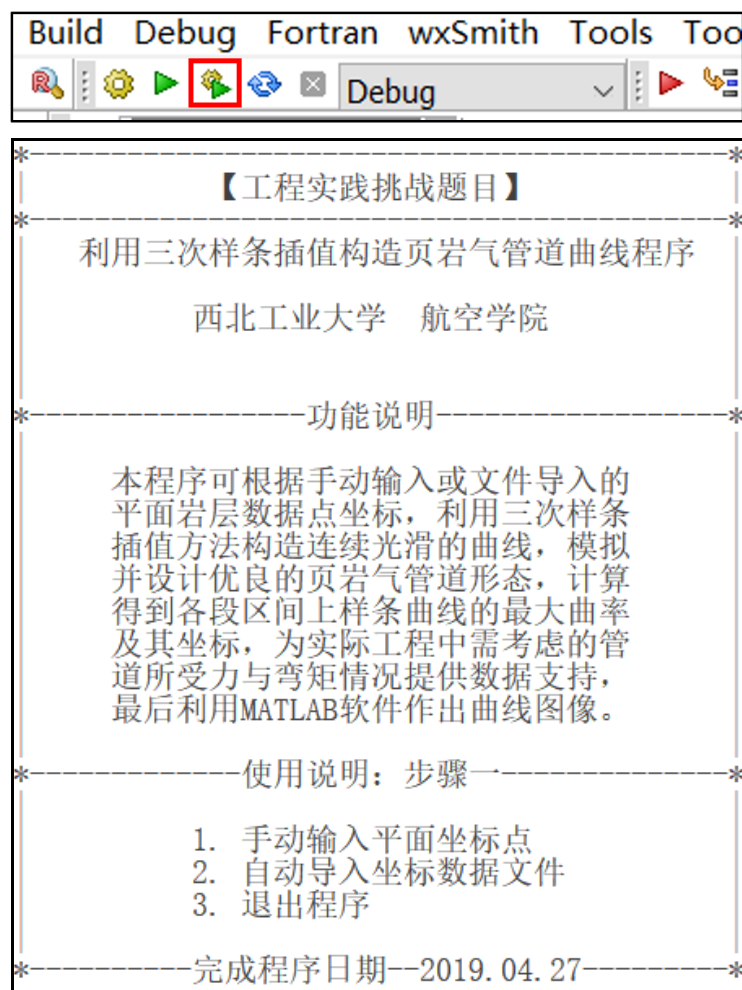


图 4 主菜单演示图

步骤一：

键盘输入 Enter 后输入 1（或 2），选择导入数据方式；

若输入 1，继续输入平面插值节点个数($N \leq 5000$ ，可修改 **const MAX** 调整设定)，以及相应的 x 、 y 坐标（单位： m ），如下图 5 所示；

```
请输入你的选则编号，然后按enter:1
请输入平面差值节点个数:
4
请输入各坐标点x坐标:
1 2 3 4
请输入各坐标点y坐标:
5 3 9 7
```

图 5 步骤一演示图（1）

若输入 2，请输入数据导入方式，继续输入 1（或 2）选择数据导入方式，如下图 6 所示（输入 1 则需将文件命名为"**data_intro.DAT**"，并保存在项目目录文件夹下， $N \leq 5000$ ，可修改 **const MAX** 调整设定）；

```
请输入你的选则编号，然后按enter:2
请输入数据导入方式
1: 导入已有数据文件（请将文件命名为"data_self.DAT"，并保存在项目目录文件夹下）
2: 随机生成一组测试数据（供验证程序参考使用）
请输入您的选择编号: 1
```

图 6 步骤一演示图（2）

步骤二：

选择边界条件，输入 1（或 2、3），如图 7 所示；

```
步骤二:
请输入边界条件
1: 已知两端的一阶导数
2: 两端的二阶导数已知
3: 默认:自然边界条件
请输入您的选择编号: 1
```

图 7 步骤二演示图

步骤三：

若步骤 2 中输入 1，则直接继续输入断点一阶导数，如图 8 所示；

```
步骤三：
请输入断点的一阶导数值：Y0' 与Y3'
Y0'=1.3
Y3'=1.9
```

图 8 步骤三演示图（1）

若输入 2，则直接继续输入断点二阶导数，如图 9 所示；

```
步骤三：
请输入断点的二阶导数：Y0''与Y3''
Y0''=1.0
Y3''=2.0
```

图 9 步骤三演示图（2）

若输入 3，则采用自然边界条件，不需手动输入，如图 10 所示；

```
步骤三：
采用自然边界条件
```

图 10 步骤三演示图（3）

步骤四：

选择是否将结果写入文件中查看，输入 y 则结果保存至文件中；输入 n 则结果直接显示到屏幕上，如图 11 所示；

```
步骤四：
是否需要将结果写入文件中查看？ y/n

请输入您的选择： y

曲线构造结果已保存在文件Interpolation_Result.DAT中：
```

图 11 步骤四演示图（1）

若程序试验结束，继续输入 n ；若需再次进行实验，输入 y ；如图 12

所示；

```

您是否需要再进行一次试验？ y/n:
请输入您的选择： n
感谢您使用本程序！
Process returned 0 (0x0)    execution time : 130.125 s
  
```

图 12 步骤四演示图（2）

步骤五：

关闭或最小化 Code::Blocks 窗口，切换至项目目录，如下图 13 所示；

名称	修改日期	类型	大小
bin	2019/5/3 星期五 ...	文件夹	
obj	2019/5/3 星期五 ...	文件夹	
main	2019/5/3 星期五 ...	C++ source file	13 KB
Coefficient_Data	2019/5/3 星期五 ...	DAT 文件	0 KB
Curvature_Max_Coordinates	2019/5/3 星期五 ...	DAT 文件	0 KB
data_intro	2019/5/3 星期五 ...	DAT 文件	2 KB
Interpolation_Result	2019/5/3 星期五 ...	DAT 文件	0 KB
Original_Coefficient_Data	2019/5/3 星期五 ...	DAT 文件	0 KB
Range	2019/5/3 星期五 ...	DAT 文件	0 KB
三次样条插值模拟实际页岩气管道设计.d...	2019/5/3 星期五 ...	DEPEND 文件	1 KB
Curve_Painting.m	2019/5/3 星期五 ...	M 文件	2 KB
三次样条插值模拟实际页岩气管道设计	2019/5/3 星期五 ...	project file	2 KB

图 13 步骤五演示图（1）

目录树下生成各输出文件说明如下：

- (1) "Interpolation_Result.DAT" 中保存最终插值曲线构造结果
- (2) "Coefficient_Data.DAT" 中保存规范化后的各段三次函数系数
- (3) "Range.DAT" 中保存各区间端点值
- (4) "Original_Coefficient_Data.DAT" 中保存原始各段三次函数系数
- (5) "Curvature_Max_Coordinates.DAT" 中保存各区间曲线最大曲率的坐标值

启动 MATLAB 软件，并将工作区目录设置在上述 Code::Blocks 工程的目录树下，如图 14 所示；

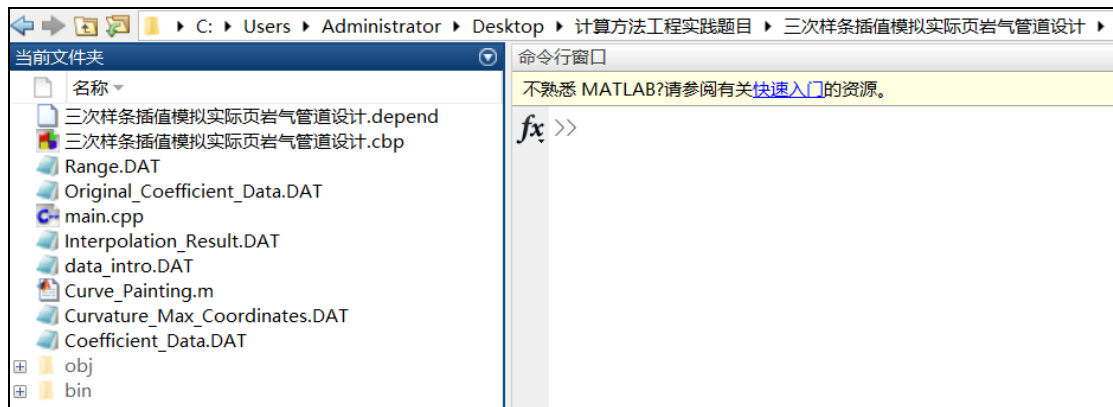




图 14 步骤五演示图（2）

打开 Curve_Painting.m 脚本文件，点击运行，绘图区即显示构造得到的三次样条曲线以及控制点、曲率最大点等图像信息。

8 样例演示

采用随机输入数据样例进行程序演示。

步骤一：打开 Code::Blocks 软件，依次点击编译  (Build)、运行  (Run) 按钮，启动控制台，进入交互式界面主菜单；

步骤二：依次选择“2-自动导入坐标数据文件”、“2-随机生成一组测试数据（供验证程序参考使用）”、“3-默认：自然边界条件”、“y-将结果写入文件查看”“n-不再进行试验”，退出程序；

步骤三：回到工作目录树，使用记事本打开“Interpolation_Result.DAT”文件，查看插值函数曲线构造结果及数据；

在该样例演示中，“Interpolation_Result.DAT”文件中的一部分内容如

图 15 所示，三次样条曲线函数的表达式、最大曲率及其发生位置均给出；

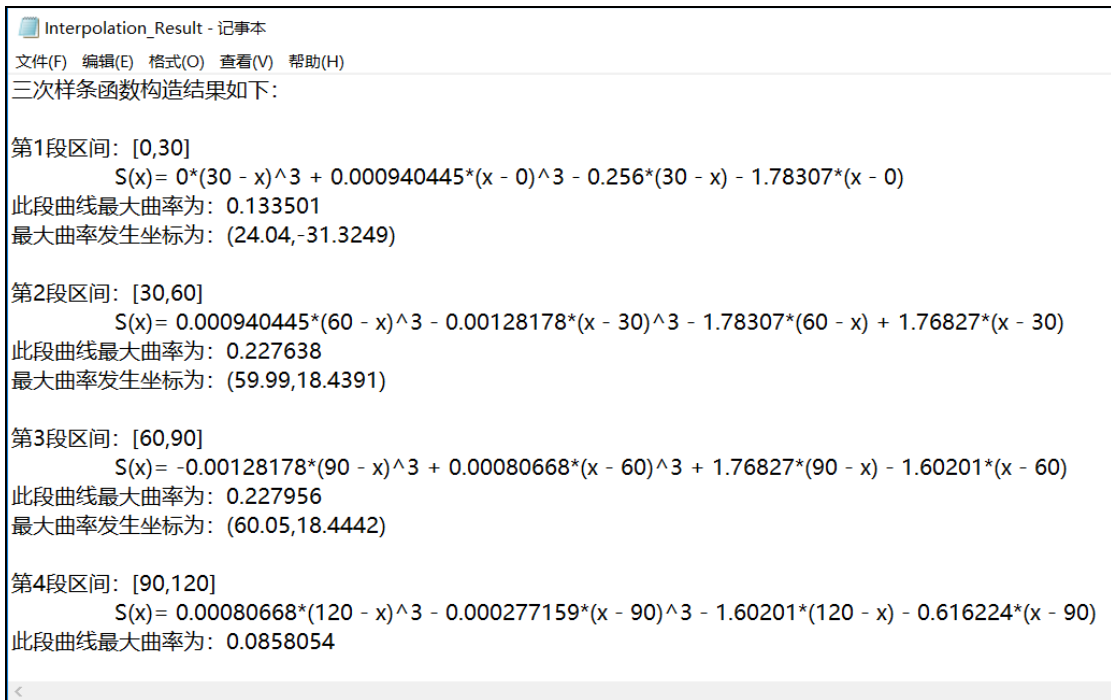



图 15 样例演示图-三次样条曲线插值结果

步骤四：启动 MATLAB R2017a，将工作目录设定与 C++工程文件目录一致，打开 Curve_Painting.m 文件，点击运行 ，显示如下三者组合图像（如图 16 所示），经简单修改即显示两两组合图像（如图 17、18 所示）；

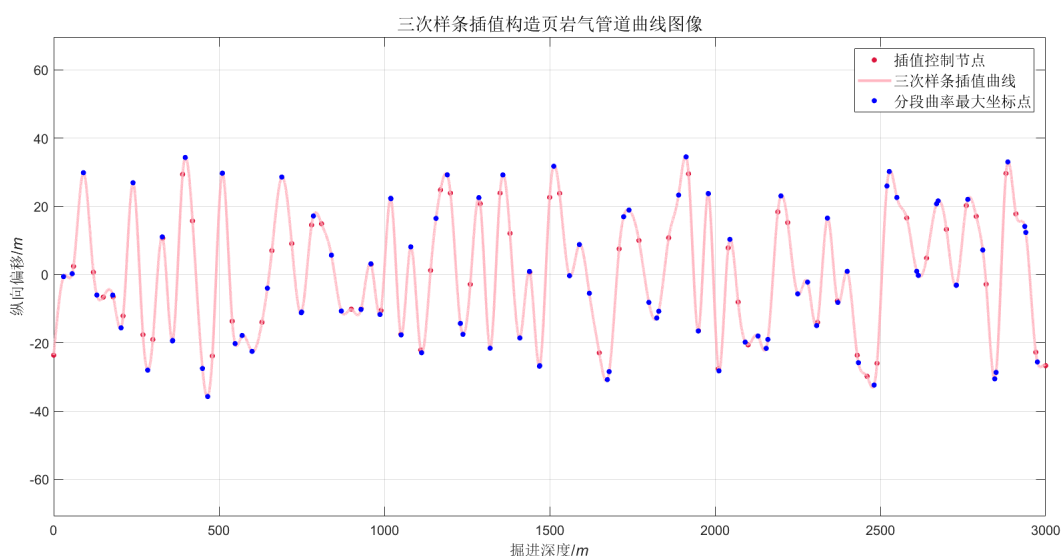


图 16 样例演示图-三次样条曲线插值结果（完整组合图像）

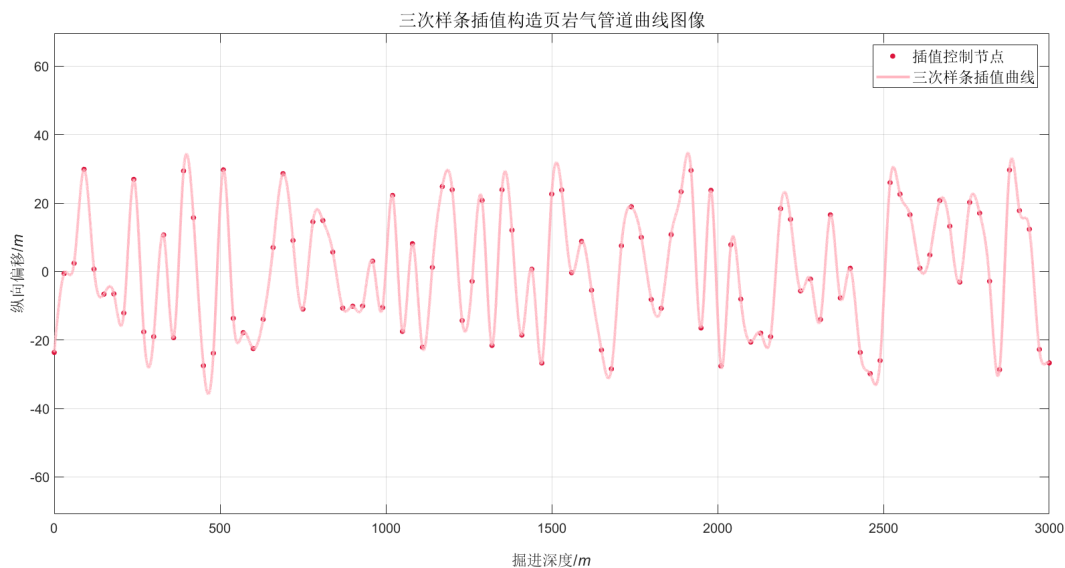


图 17 样例演示图-三次样条曲线插值结果（控制节点+插值曲线）

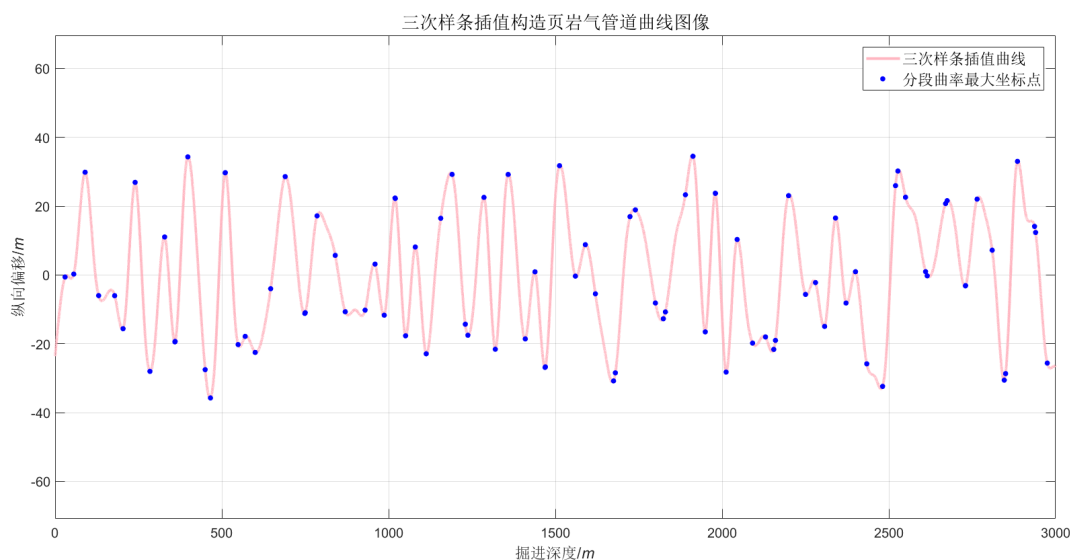


图 18 样例演示图-三次样条曲线插值结果（插值曲线+分段曲率最大点）

由图 16~18 的三幅组合图像，可以清晰看出，对于随机数据，该程序具有较好地实现了三次样条插值构造，并能够提供或显示出关键数据与图像信息，方便后续的工程进行设计应用，对于不用要求具有普适性，兼具便捷性、美观性，基本达到了预期的效果。

9 程序完整源代码

1. C++ Program

```
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include <ctime>
#include <fstream>
#include <sstream>

using namespace std;

// 全局变量定义：允许输入的最大节点数，可修改
const int MAX = 5000;
int N;

ofstream cout1("Interpolation_Result.DAT"); // "Interpolation_Result.DAT"中保存最终插值曲线构造结果
ofstream cout2("Coefficient_Data.DAT"); // "Coefficient_Data.DAT"中保存规范化后的各段三次函数系数
ofstream cout3("Range.DAT"); // "Range.DAT"中保存各区间端点值
ofstream cout4("Original_Coefficient_Data.DAT"); // "Original_Coefficient_Data.DAT"中保存原始各段三次函数系数
ofstream cout5("Curvature_Max_Coordinates.DAT"); // "Curvature_Max_Coordinates.DAT"中保存各区间曲线最大曲率的坐标值

double x[MAX]={0},y[MAX]={0}; // x为各点横坐标；y为各点纵坐标；
double h[MAX]={0}; // h为步长
double c[MAX]={0},a[MAX]={0},fxym[MAX]={0};
double q_max[MAX]={0},x_max[MAX]={0},y_max[MAX]={0}; // 每个分段区间的最大曲率及其坐标

// 函数提前声明

void menu();
void data_intro();
```



```

void data_random();
void spline_construction(); // 各段三次样条差值曲线构造
double f(int x1,int x2,int x3); // 求差分函数(含三个参数)
void cal_m(int n); // 用追赶法求解出弯矩向量M
void cal_curvature(double a,double b,double c,double d,double x1,double x2,int m); //输入三
次函数系数，区间端点，区间编号
void data_output_console(int n);
void data_output_file(int n);

int main() //主程序实现
{
    int choice,i;
    char ch;
    do
    {
        menu();
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: // 手动输入各个节点值
                printf("请输入平面差值节点个数:\n");
                scanf("%d",&N);
                printf("请输入各坐标点x坐标:\n");
                for (i=0;i<N;i++) scanf("%lf",&x[i]);
                printf("请输入各坐标点y坐标:\n");
                for (i=0;i<N;i++) scanf("%lf",&y[i]);
                break;
            case 2: // 自动导入文件数据
                data_intro(); // 文件数据导入与预处理
                break;
            case 3: // 退出程序
                exit(0);
        }
        spline_construction(); // 构造三次样条曲线
        cout<<"\n您是否需要再进行一次试验? y/n: \n";
        cout<<"\n请输入您的选择: ";
        cin>>ch;
    }
    while(ch=='y' || ch=='Y');
}

```



```

cout1.close(); // 关闭输出文件
cout2.close();
cout3.close();
cout4.close();
cout5.close();
cout<<"\n感谢您使用本程序！\n";
return 0;
}

void menu()
{
    //设置菜单界面
    printf("*-----*\n");
    printf("          【工程实践挑战题目】          |\n");
    printf("*-----*\n");
    printf("    利用三次样条插值构造页岩气管道曲线程序 |\n");
    printf("          |\n");
    printf("          西北工业大学 航空学院          |\n");
    printf("          冯铮浩 卢佐 姜丽升 秦嘉琛 戴世城 |\n");
    printf("          |\n");
    printf("*-----功能说明-----*\n");
    printf("          |\n");
    printf("    本程序可根据手动输入或文件导入的      |\n");
    printf("    平面岩层数据点坐标，利用三次样条      |\n");
    printf("    插值方法构造连续光滑的曲线，模拟      |\n");
    printf("    并设计优良的页岩气管道形态，计算      |\n");
    printf("    得到各段区间上样条曲线的最大曲率      |\n");
    printf("    及其坐标，为实际工程中需考虑的管      |\n");
    printf("    道所受力与弯矩情况提供数据支持，      |\n");
    printf("    最后利用MATLAB软件作出曲线图像。      |\n");
    printf("          |\n");
    printf("*-----使用说明：步骤一-----*\n");
    printf("          |\n");
    printf("          1. 手动输入平面坐标点          |\n");
    printf("          2. 自动导入坐标数据文件          |\n");
    printf("          3. 退出程序                      |\n");
    printf("          |\n");
    printf("*-----完成程序日期--2019.04.27-----*\n");
    printf("\n");
}

```

```
cout<<"请输入你的选则编号，然后按enter:";
}

void data_intro()
{
    int i,t;
    char c;
    FILE *fp;
    cout<<"\n请输入数据导入方式\n 1: 导入已有数据文件";
    cout<<"（请将文件命名为"<<"data_intro.DAT"<<"，并保存在项目目录文件夹下）\n";
    cout<<" 2: 随机生成一组测试数据（供验证程序参考使用） \n\n";
    cout<<"请输入您的选择编号： ";
    cin>>t;
    switch (t)
    {
        case 1:
            fp=fopen("data_intro.DAT","r");
            // 计算坐标点个数
            N=0;
            for (i=0;i<MAX;i++)
            {
                N++;
                fscanf(fp,"%lf%c",&x[i],&c);
                if (c=='\n') break;
            }
            for (i=0;i<N;i++) fscanf(fp,"%lf",&y[i]);
            break;
        case 2:
            data_random();
            fp=fopen("data_intro.DAT","r");
            // 计算坐标点个数
            N=0;
            for (i=0;i<MAX;i++)
            {
                N++;
                fscanf(fp,"%lf%c",&x[i],&c);
                if (c=='\n') break;
            }
            for (i=0;i<N;i++) fscanf(fp,"%lf",&y[i]);
    }
}
```

```

        break;
    }
}

void data_random()
{
    int i,dn;
    double x[MAX],b[MAX],c[MAX],y[MAX];
    FILE *fp;
    dn=100; // 随机生成区间段数
    if ((fp=fopen("data_intro.DAT","w"))==NULL)
    {
        printf("文件不能打开！ \n");
        exit(0);
    }
    x[0]=0;
    for(i=0;i<dn+1;i++) x[i+1]=x[i]+30;
    srand(((unsigned)time(NULL))); //用当前系统时间设置种子
    for(i=0;i<dn+1;i++)
    {
        b[i]=rand()%6001; //用rand函数生成0-6000的随机数，并赋值给数组b[i]
        c[i]=b[i]/100;
        y[i]=c[i]-30; //生成-30~+30的随机数，并赋值给数组y[i]
    }
    for(i=0;i<dn;i++) fprintf(fp,"%0.2f",x[i]);
    fprintf(fp,"%0.2f",x[dn]);
    fprintf(fp,"\n");
    for(i=0;i<dn+1;i++)
        fprintf(fp,"%0.2f",y[i]);
    fclose(fp);
    // getchar();
    printf("\n随机数据已存储在文件data_intro.dat中\n");
}

void spline_construction() // 各段三次样条差值曲线构造
{
    int n,i,t;
    char flag;
    double f0,f1;

```

```

n=N-1;
for(i=0;i<n;i++) h[i]=x[i+1]-x[i]; //求步长；其数组值较之坐标点个数少1
cout<<"\n步骤二： "<<endl;
cout<<"请输入边界条件\n 1: 已知两端的一阶导数\n 2: 两端的二阶导数已知\n 3: 默认:
自然边界条件\n\n";
cout<<"请输入您的选择编号： ";
cin>>t;
cout<<"\n步骤三： "<<endl;
switch(t)
{
    case 1:
        cout<<"请输入断点的一阶导数值： "<<"Y0\'与Y" <<n<<"\'\'n"; //显示数据为Y0'
        至Yn'，即断点的一阶导数
        cout<<"Y0\'="; cin>>f0;
        cout<<"Y" <<n<<"\'="; cin>>f1;
        c[0]=1;
        a[n]=1;
        fxym[0]=6*((y[1]-y[0])/(x[1]-x[0])-f0)/h[0];
        fxym[n]=6*(f1-(y[n]-y[n-1])/(x[n]-x[n-1]))/h[n-1];
        break;
    case 2:
        cout<<"请输入断点的二阶导数： "<<"Y0\'\'与Y" <<n<<"\'\'\'n"; //显示数据为Y0'
        至Yn'，即断点的一阶导数
        cout<<"Y0\'\'="; cin>>f0;
        cout<<"Y" <<n<<"\'\'="; cin>>f1;
        c[0]=a[n]=0;
        fxym[0]=2*f0;
        fxym[n]=2*f1;
        break;
    default:
        cout<<"采用自然边界条件\n"; //待定
};
for (i=1;i<n;i++)
    fxym[i]=6*f(i-1,i,i+1); //调用差分函数(only!)
for (i=1;i<n;i++)
{
    a[i]=h[i-1]/(h[i]+h[i-1]);
    c[i]=1-a[i];
}

```

```

a[n]=h[n-1]/(h[n-1]+h[n]);
cal_m(n); // 调用弯矩函数
cout<<"\n步骤四: "<<endl;
cout<<"是否需要将结果写入文件中查看? y/n\n";
cout<<"\n请输入您的选择: ";
cin>>flag;
if (flag=='y') data_output_file(n);
else data_output_console(n);
}

double f(int x1,int x2,int x3) // 求差分函数(含三个参数)
{
    double a=(y[x3]-y[x2])/(x[x3]-x[x2]);
    double b=(y[x2]-y[x1])/(x[x2]-x[x1]);
    return (a-b)/(x[x3]-x[x1]);
}

void cal_m(int n) // 用追赶法求解出弯矩向量M
{
    int i;
    double B[MAX];
    B[0]=c[0]/2;
    for (i=1;i<n;i++)
        B[i]=c[i]/(2-a[i]*B[i-1]);
    fxym[0]=fxym[0]/2;
    for (i=1;i<=n;i++)
        fxym[i]=(fxym[i]-a[i]*fxym[i-1])/(2-a[i]*B[i-1]);
    for (i=n-1;i>=0;i--)
        fxym[i]=fxym[i]-B[i]*fxym[i+1];
}

void cal_curvature(double a,double b,double c,double d,double x1,double x2,int m) //输入三
// 次函数系数, 区间端点, 区间编号
{
    int i,n;
    double h=0.01; //求最大曲率时的扫描步长
    double q1=0,x,t,y_1,y_2;
    if (x1>x2)
    {

```

```

        t=x1;    //若输入的x1小于x2，两数结果交换
        x1=x2;
        x2=t;
    }
    n=int((x2-x1)/h); //扫描次数
    q_max[m]=0;
    for (i=0;i<=n;i++)
    {
        x=x1+i*h;
        y_1=-3*a*(x2-x)*(x2-x)-b+3*c*(x-x1)*(x-x1)+d; //一阶导数
        y_2=6*a*(x2-x)+6*c*(x-x1); //二阶导数
        q1=y_2/pow(1+y_1*y_1,3.0/2); //计算曲率
        q1=fabs(q1);
        if (q1>q_max[m])
        {
            q_max[m]=q1;
            x_max[m]=x;
            y_max[m]=a*(x2-x)*(x2-x)*(x2-x)+b*(x2-x)+c*(x-x1)*(x-x1)*(x-x1)+d*(x-x1);
        }
    }
}

void data_output_file(int n)// 输出各段三次样条插值结果（因已知断点个数而异）
{
    int i;
    double a0,b0,c0,d0,a1,b1,c1,d1;
    cout<<"\n曲线构造结果已保存在文件Interpolation_Result.DAT中：\n";
    cout1<<"三次样条函数构造结果如下："<<endl<<endl;
    cout1<<setprecision(6);//通过操作器setprecision()设置有效位数；其为头文件
    <iomanip.h>所包含；括号内为参数。
    for(i=0;i<n;i++) //所输出函数个数由所设断点个数而定
    {
        // 计算原形式三次多项式系数
        a0=(fxym[i])/(6*h[i]); // a0
        b0=(y[i]-(fxym[i]*h[i]*h[i])/6)/h[i]; // b0
        c0=fxym[i+1]/(6*h[i]); // c0
        d0=(y[i+1]-(fxym[i+1]*h[i]*h[i]/6))/h[i]; // d0
        // 计算规范三次多项式系数
        a1=(a0+c0);
    }
}

```

```

b1=(-3*a0*x[i+1]-3*c0*x[i]);
c1=((3*a0*x[i+1]*x[i+1])+b0+(3*c0*x[i]*x[i])+d0);
d1=[(-1*a0*x[i+1]*x[i+1]*x[i+1])-(b0*x[i+1])-(c0*x[i]*x[i]*x[i])-(d0*x[i])];
// 计算该段曲线上的最大曲率
cal_curvature(a0,b0,c0,d0,x[i],x[i+1],i+1);
// 输出各段三次样条曲线方程
cout1<<"第"<<i+1<<"段区间"<<": ["<<x[i]<<","<<x[i+1]<<"]\n"<<"\t";
cout1<<"S(x)= ";

    cout1<<a0<<"*("<<x[i+1]<<" - x)^3";

if (c0>0)
    cout1<<" + "<<c0<<"*(x - "<<x[i]<<")^3";
else
    cout1<<" - "<<-c0<<"*(x - "<<x[i]<<")^3";

if (b0>0)
    cout1<<" + "<<b0<<"*("<<x[i+1]<<" - x)";
else
    cout1<<" - "<<-b0<<"*("<<x[i+1]<<" - x)";

if (d0>0)
    cout1<<" + "<<d0<<"*(x - "<<x[i]<<")";
else
    cout1<<" - "<<-d0<<"*(x - "<<x[i]<<")";
cout1<<endl;
//输出该段曲线最大曲率及其坐标
cout1<<"此段曲线最大曲率为: "<<q_max[i+1]<<endl;
cout1<<"最大曲率发生坐标为: "<<"("<<x_max[i+1]<<","<<y_max[i+1]<<")"<<endl;
cout1<<endl;
    // 输出系数数据
cout2<<a1<<" "<<b1<<" "<<c1<<" "<<d1<<"\n"<<endl;
cout3<<x[i]<<" "<<x[i+1]<<"\n"<<endl;
cout4<<a0<<" "<<b0<<" "<<c0<<" "<<d0<<"\n"<<endl;
cout5<<x_max[i+1]<<" "<<y_max[i+1]<<"\n"<<endl;
}
cout<<endl;
}

```

```

void data_output_console(int n)
{
    int i;
    double a0,b0,c0,d0,a1,b1,c1,d1;
    cout<<"\n结果已在下方屏幕显示：\n";
    cout<<"\n三次样条函数构造结果如下："<<endl<<endl;
    cout<<setprecision(6);//通过操作器setprecision()设置有效位数；其为头文件
    <iomanip.h>所包含；括号内为参数。
    for (i=0;i<n;i++) //所输出函数个数由所设断点个数而定
    {
        // 计算原形式三次多项式系数
        a0=(fxym[i])/(6*h[i]); // a0
        b0=(y[i]-(fxym[i]*h[i]*h[i])/6)/h[i]; // b0
        c0=fxym[i+1]/(6*h[i]); // c0
        d0=(y[i+1]-(fxym[i+1]*h[i]*h[i])/6)/h[i]; // d0
        // 计算规范三次多项式系数
        a1=(a0+c0);
        b1=(-3*a0*x[i+1]-3*c0*x[i]);
        c1=((3*a0*x[i+1]*x[i+1])+b0+(3*c0*x[i]*x[i])+d0);
        d1=[(-1*a0*x[i+1]*x[i+1]*x[i+1])-(b0*x[i+1])-(c0*x[i]*x[i]*x[i])-(d0*x[i])];
        // 计算该段曲线上的最大曲率
        cal_curvature(a0,b0,c0,d0,x[i],x[i+1],i+1);

        // 输出各段三次样条曲线方程
        cout<<"第"<<i+1<<"段区间"<<": ["<<x[i]<<","<<x[i+1]<<"]\n"<<"\t";
        cout<<"S(x)= ";

        if (a0>0)
            cout<<a0<<"*("<<x[i+1]<<" - x)^3";
        else
            cout<<-a0<<"*("<<x[i+1]<<" - x)^3";

        if (c0>0)
            cout<<" + "<<c0<<"*("<<x[i]<<")^3";
        else
            cout<<" - "<<-c0<<"*("<<x[i]<<")^3";

        if (b0>0)
            cout<<" + "<<b0<<"*("<<x[i+1]<<" - x)";

```



```

else
    cout<<" - "<<-b0<<"*("<<x[i+1]<<" - x");

if (d0>0)
    cout<<" + "<<d0<<"*(x - "<<x[i]<<"");
else
    cout<<" - "<<-d0<<"*(x - "<<x[i]<<"");
cout<<endl;
//输出该段曲线最大曲率及其坐标
cout<<"此段曲线最大曲率为: "<<q_max[i+1]<<endl;
cout<<"最大曲率发生坐标为: "<<"("<<x_max[i+1]<<","<<y_max[i+1]<<")"<<endl;
cout<<endl;
cout2<<a1<<" "<<b1<<" "<<c1<<" "<<d1<<" "<<x_max[i+1]<<" "<<y_max[i+1]<<"
"<<"\n"<<endl;
cout3<<x[i]<<" "<<x[i+1]<<"\n"<<endl;
cout4<<a0<<" "<<b0<<" "<<c0<<" "<<d0<<"\n"<<endl;
cout5<<x_max[i+1]<<" "<<y_max[i+1]<<"\n"<<endl;
}
cout<<endl;
}

```

2. MATLAB Program

%% 三次样条插值构造页岩气管道曲线MATLAB作图

%% 数据初始化

clear all

clf

clc

A=load('data_intro.DAT');

C=load('Original_Coefficient_Data.DAT');

D=load('Range.DAT');

M=load('Curvature_Max_Coordinates.DAT');

[m,n]=size(C);

gap=0.01;

%% 绘制曲线

% 提取各个原始点坐标

A1=A(1,:);

A2=A(2,:);

```

% 提取各个原始系数
C1=C(:,1);
C2=C(:,2);
C3=C(:,3);
C4=C(:,4);
% 提取区间端点值
D1=D(:,1);
D2=D(:,2);
Max_x_min=min(D1);
Max_x_max=max(D2);
% 提取区间最大曲率坐标值
M1=M(:,1);
M2=M(:,2);
temp=(max(M2)-min(M2))/2;
Max_y_max=max(M2)+temp;
Max_y_min=min(M2)-temp;
% 作出每段插值曲线函数图像
plot(A1,A2,'.','MarkerSize',16,'Color',[220/255,20/255,60/255]);
% 第k段区间绘制
hold on
for k=1:m
    xx=D1(k):gap:D2(k);
    Ds=ones(1,length(xx))*D1(k);
    Dt=ones(1,length(xx))*D2(k);
    yy=C1(k)*((Dt-xx).^3)+C3(k)*((xx-Ds).^3)+C2(k)*(Dt-xx)+C4(k)*(xx-Ds);
    plot(xx,yy,'-','LineWidth',2.4,'Color',[1,182/255,193/255]);
    plot(M1,M2,'.','MarkerSize',16,'Color',[0,0,1]);
end
legend('插值控制节点','三次样条插值曲线','分段曲率最大坐标点');
xlabel('掘进深度/\text{itm}');ylabel('纵向偏移/\text{itm}');
title('三次样条插值构造页岩气管道曲线图像');
axis([Max_x_min,Max_x_max,Max_y_min,Max_y_max]);
grid on
box on

```

10 团队分工及程序开发总结

本团队的具体分工如下：

冯铮浩：三次样条核心程序编写；交互式页面设计；完成函数组合连接；报告主体构架排版与撰写。

姜丽升、秦嘉琛：数据导入导出部分程序的编写；生成随机数据的程序编写，以检验程序的正确性。

卢佐：程序调试与测试；利用 MATLAB 绘制三次样条曲线示意图。

单湘淋、戴世城：最大曲率位置计算函数模块及相应程序报告书撰写。

部分团队成员程序开发感悟：

在本学期《计算方法》的课程学习中，前人众多精妙而严谨的计算理论进一步提升了我探索算法的热情。虽然自己曾经接触过一些关于数学建模与常用算法的知识，但是过去我的状态基本是“会用但不明其理”。通过系统的课堂理论学习与自主实习环节，特别是在挑战完成老师布置的三次样条模拟构造页岩气管道的问题时，我充分感受到理论与实践相结合迸发出的强大力量。这些凝聚着无数智慧的计算方法，有坚实的理论基础，更有广泛的工程应用前景。通过用最基础的 C 语言编写程序进行问题求解，我认识问题，理性分析问题以及自主解决问题的综合能力都得到了显著的提升，也对实际工程问题有了更加深刻的理解。同时，我对算法的理解从停留在“会”层面上升到“懂”层面，这对于今后我在科学研究与工程设计等方面的发展都大有裨益。但是，实习内容细节还不够充分，方法还不够多元。在今后的学习生活中，我也会更加注重将课堂学习与亲身实验相

结合，培养能力与兴趣，为未来的科研之路打下更加坚实的基础！

——团队队长 冯铮浩

通过程序设计解决一个工程实际问题,首先要了解这个问题的基本要求，即输入、输出、完成从输入到输出的要求是什么；其次，从问题的关键入手，从前到后的解决问题的每个方面，着重考虑如何从输入导出输出，在这个过程中,可确定所需的变量、数组、函数，然后确定处理过程——算法。通过实际工程问题的求解，使我们对课堂中所学的内容有了更深刻的理解，是一次拓展能力、发散思维的机会，让我们体会到了程序设计的魅力，计算方法的美妙。

——团队成员 秦嘉琛

11 参考文献

- [1] 天工. 《页岩气发展规划(2016—2020 年)》发布[J]. 天然气工业,2016(10).
- [2] 百度百科词条 “三次样条差值”
<https://baike.baidu.com/item/%E4%B8%89%E6%AC%A1%E6%A0%B7%E6%9D%A1%E6%8F%92%E5%80%BC/3476729?fr=aladdin>