报告文档

‘一、程序优化说明

1、用户交互界面说明

主界面，包括打开文件，纵断面计算，横断面计算，打开报告，清空数据功能。

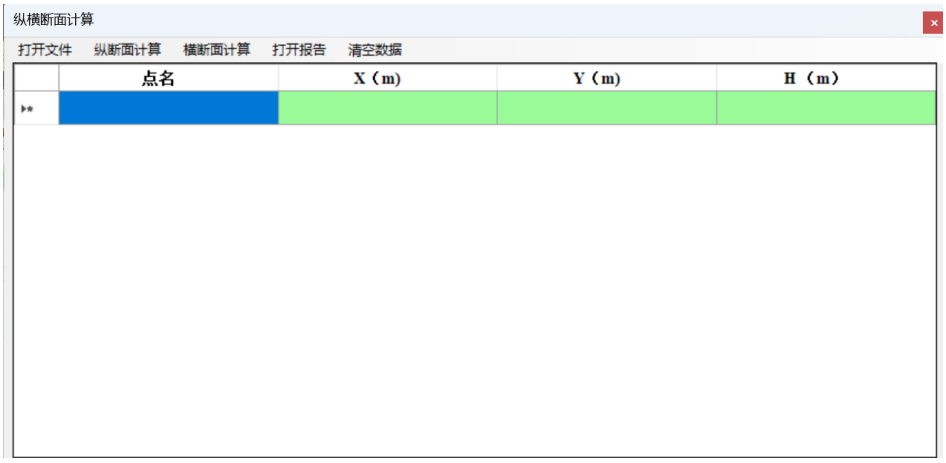


图 1 主界面

图 2 读取数据

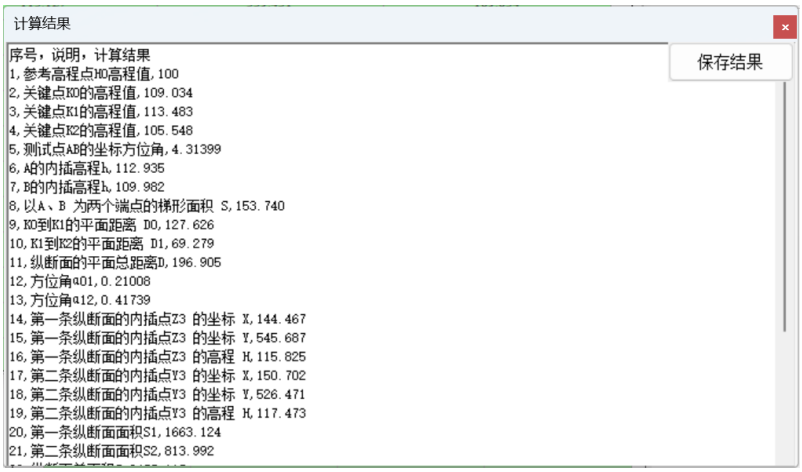


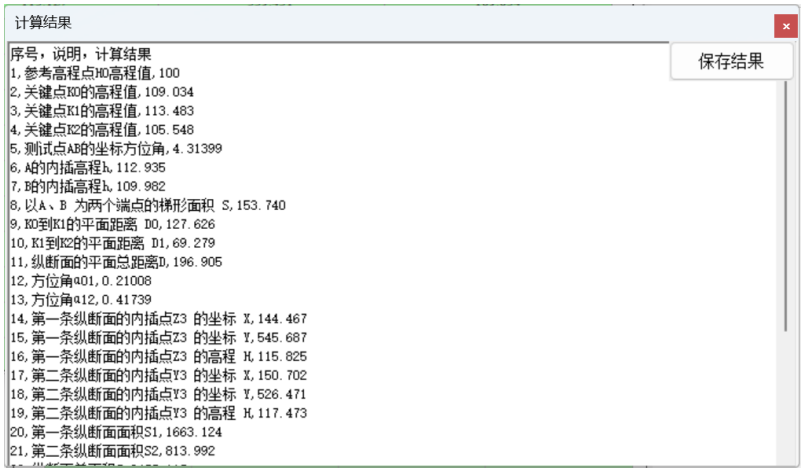
图 3 报告界面

点击打开报告出现报告界面。

2、程序运行过程说明

先点击打开文件，读入数据，然后点击纵断面计算，再点击横断面计算，完成计算。

3、程序运行结果



二、程序规范说明

1、程序功能与设计结构说明

```
class ZHPoints
{
    public string ID { get; set; }
    public double X { get; set; }
    public double Y { get; set; }
    public double H { get; set; }
    public double Dis { get; set; } //距离另一个点的距离
    public List<ZHPoints> LeastFivePoints;//最近的五个离散点

    public ZHPoints()
    {

    }
    public ZHPoints(string ID)
    {
        this.ID = ID;
    }
    public ZHPoints(string ID,double X,double Y,double H)
    {
        this.ID = ID;
        this.X = X;
        this.Y = Y;
        this.H = H;

    }
    public ZHPoints(string ID, double X, double Y)
    {
        this.ID = ID;
        this.X = X;
        this.Y = Y;
    }
```

该程序核心类为 ZHPoints，用来存放点数据，包括关键点，离散点等。通过定义 List<ZHPoints> LeastFivePoints 来存储最近的五个离散点。

```
    public static double H0 = 100.0;
    //计算坐标方位角
    public static double FangWei(double y,double x)
    {
        if(x>0 && y>0)
        {
            return Math.Atan(y / x);
        }
        else if(x< 0 && y>0)
        {
            return Math.PI - Math.Abs(Math.Atan(y / x));
        }
        else if(x <0 && y<0)
        {
            return Math.PI + Math.Atan(y / x);
        }
        else if (x > 0 && y < 0)
        {
            return Math.PI *2 - Math.Atan(y / x);
        }
        else if(x == 0 && y >0)
        {
            return Math.Atan(Alogrithm.Du2Rad(90));
        }
        else if (x == 0 && y < 0)
        {
            return Math.Atan(Alogrithm.Du2Rad(270));
        }
        else
        {
            return Math.Abs(Math.Atan(y / x));
        }
    }
```

方位角计算公式如上。

```
//计算一个点的五个最邻近点
public static void CalLeastFivePoints(List<ZHPoints> points)
{
    //定义一个点到其它点的距离集合
    for(int i = 0;i<points.Count;i++)
    {
        List<ZHPoints> OtherPoints = new List<ZHPoints>();
        for(int j =0;j<points.Count;j++)
        {
            if(i == j)
            {
                continue;
            }
            else
            {
                points[j].Dis = CalDistance(points[i],points[j]);
                OtherPoints.Add(points[j]);
            }
        }
        points[i].LeastFivePoints =  OtherPoints.OrderBy(p => p.Dis).Take(5).ToList();
        OtherPoints = null;
    }
}
```

通过 C# linq 中的 OrderBy，Take 来获取最近的五个点。

```
//内插高程计算，需先计算五个最近点
public static void CalPointsHeight(ZHPoints point)
{
    double Sum_up = 0, Sum_down = 0;
    for(int i =0; i<point.LeastFivePoints.Count;i++)
    {
        Sum_up += point.LeastFivePoints[i].H / CalDistance(point, point.LeastFivePoints[i]);
        Sum_down += 1 / CalDistance(point, point.LeastFivePoints[i]);
    }
    point.H = Sum_up / Sum_down;
}

//计算两点间断面面积
public static double CalDuanMianArea(ZHPoints p1,ZHPoints p2)
{
    double Si = (p1.H + p2.H - 2 * H0) / 2 * CalDistance(p2, p1);
    return Si;
}
```

高程内插计算和断面面积计算。