**封面**

**目录**

[**1 绪论** 1](#_Toc152253748)

[1.1 研究背景及意义 2](#_Toc152253750)

[1.2 国内外研究现状 3](#_Toc152253751)

1.2.1 国外研究现状 3

1.2.2 国内研究现状 3

[1.3 研究内容与技术路线 3](#_Toc152253751)

1.3.1 研究内容 3

1.3.2 技术路线 3

[**2 三维模型构建** 2](#_Toc152253749)

[2.1 Delaunay 三角剖分方法 2](#_Toc152253750)

2.1.1 遵循Delaunay准则的约束三角剖分 3

2.1.2 三维地质模型数据结构 3

[2.2 断层模型构建 3](#_Toc152253751)

2.2.1 相交断层模型的构建方法 3

2.2.2 断层模型网格化 3

[2.3 地层模型构建 3](#_Toc152253751)

2.3.1 CDT剖分数据源提取 3

2.3.2 地层数据网格化 3

[2.4 地质模型集成 3](#_Toc152253751)

2.4.1 钻孔 3

2.4.2 剖面模型（勘探线、剖分的剖面） 3

2.4.2 地层模型 3

2.4.2 断层模型 3

[2.5 空间几何信息的计算 3](#_Toc152253751)

2.5.1 地层层间距计算 3

2.5.2（） 3

[**3 三维模型可视化方法** 5](#_Toc152253752)

[3.1 Web可视化技术 5](#_Toc152253753)

3.1.1 WebGL技术 3

3.1.2 Threejs技术 3

[3.2 构建三维可视化场景 5](#_Toc152253754)

[3.3 模型渲染方法 5](#_Toc152253754)

3.3.1 **巷道纹理uv计算方法** 3

3.3.2 巷道模型可视化 3

[3.4 三维地质模型交互功能 5](#_Toc152253754)

3.4.1 射线追踪 3

3.4.2 坐标转换 3

3.4.3 巷道漫游 3

3.4.4 点选功能 3

[**4 三维可视化系统开发** 6](#_Toc152253756)

[4.1 系统架构设计 6](#_Toc152253757)

4.1.1 体系结构 3

4.1.2 技术流程 3

[4.2 系统功能实现 7](#_Toc152253758)

4.2.1 射线追踪 3

4.2.2 系统功能测试 3

[**5 应用实例** 8](#_Toc152253759)

[5.1 平台总体框架搭建 2](#_Toc152253750)

[5.2 构建三维场景以及三维地质模型集成 3](#_Toc152253751)

[5.3 三维地质模型交互功能实现 3](#_Toc152253751)

[5.4 三维地质模型信息集成与展示 2](#_Toc152253750)

[5.5 (本章小结) 3](#_Toc152253751)

[**6 结论与展望** 8](#_Toc152253760)

[6.1 结论 2](#_Toc152253750)

[6.2 展望 3](#_Toc152253751)

[**参考文献** 9](#_Toc152253762)

# 绪论

# 三维模型构建

## 2.1Delaunay 三角剖分方法

### 2.1.1 遵循Delaunay准则的约束三角剖分（公式）

### 2.1.2 三维地质模型数据结构\*\*（包括地层、断层、钻孔模型等等）

## 2.2断层模型构建（公式）

### 2.2.1 相交断层模型的构建方法（公式）创新点

1. 两点之间的距离公式

用于计算两点之间的直线距离。代码中多个地方使用了公式，例如 `comditance` 函数：

double distance = Math.sqrt((X-X1)\*(X-X1) + (Y-Y1)\*(Y-Y1));

解释：两点 (X, Y) 和 (X1, Y1) 之间的距离公式：

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>d</mi><mo>=</mo><msqrt><msup><mfenced><mrow><mi>X</mi><mo>-</mo><mi>X</mi><mn>1</mn></mrow></mfenced><mn>2</mn></msup><mo>+</mo><msup><mfenced><mrow><mi>Y</mi><mo>-</mo><mi>Y</mi><mn>1</mn></mrow></mfenced><mn>2</mn></msup></msqrt></mstyle></math>","origin":"MathType for Microsoft Add-in"}

2.点到直线的距离公式

用于计算某个点到一条直线的距离。公式如下：

double A = Y2 - Y1;

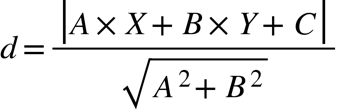
double B = X1 - X2;

double C = X2 \* Y1 - X1 \* Y2;

distance = Math.abs(A \* X + B \* Y + C) / Math.sqrt(A \* A + B \* B);

```

解释：给定两点 (X1, Y1) 和 (X2, Y2)，表示一条直线，和一个点 (X, Y)，该点到这条直线的距离公式为：



其中：

- A = Y2 - Y1

- B = X1 - X2

- C = X2 \* Y1 - X1 \* Y2

3. 两条直线的交点公式

用于计算两条直线的交点。该公式在 `intersection` 函数中被实现：

double x = (line2.b - line1.b) / (line1.k - line2.k);

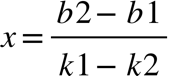
double y = x \* line1.k + line1.b;

- 解释：两条直线的方程分别为：

- y = k1 \* x + b1

- y = k2 \* x + b2

它们的交点坐标 (x, y) 的计算公式为：

，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>y</mi><mo>=</mo><mi>k</mi><mn>1</mn><mo>&#xD7;</mo><mi>x</mi><mo>+</mo><mi>b</mi><mn>1</mn></mstyle></math>","origin":"MathType for Microsoft Add-in"}

4. 判断点是否在线段范围内

用于判断一个点是否位于一条线段上，公式如下：

private static boolean isBetween(double start, double middle, double end) {

return start <= middle && middle <= end;

}

解释：这是用于判断一个点是否在两点之间的方法。例如判断middle 是否在start 和 end 之间。

### 5. \*\*判定两条线段是否相交\*\*

在 `intersection` 函数中，通过比较斜率和截距来判断两条线段是否相交，代码如下：

if (Math.abs(line1.k - line2.k) <= epslion) {

if (Math.abs(line1.b - line2.b) <= epslion) {

}

}

```

- \*\*解释\*\*：如果两条直线的斜率相同（表示平行），可以通过比较它们的截距来判断是否重合。重合则可能有交点，否则平行。

这些是代码中提取出的主要数学公式，涵盖了点到直线的距离、线段相交、几何计算等方面。

### 2.2.2 断层模型网格化

# 2.3地层模型构建

### 2.3.1 CDT剖分数据源提取（\*\*具体计算的内容、方法名）（公式）

**上下盘**

主要集中在计算平面方程和计算点的坐标位置上。

**1. 平面方程的计算：**

在 get\_panel 方法中，计算了平面方程的系数 , ，这是计算一个通过三点的平面方程：

• **计算平面方程系数**：

• {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>a</mi><mo>=</mo><mfenced><mrow><mfenced><mrow><mi>p</mi><mn>2</mn><mo>.</mo><mi>y</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>y</mi></mrow></mfenced><mo>&#xD7;</mo><mfenced><mrow><mi>p</mi><mn>3</mn><mo>.</mo><mi>z</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>z</mi></mrow></mfenced><mo>-</mo><mfenced><mrow><mi>p</mi><mn>2</mn><mo>.</mo><mi>z</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>z</mi></mrow></mfenced><mo>&#xD7;</mo><mfenced><mrow><mi>p</mi><mn>3</mn><mo>.</mo><mi>y</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>y</mi></mrow></mfenced></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

• {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>b</mi><mo>=</mo><mfenced><mrow><mfenced><mrow><mi>p</mi><mn>2</mn><mo>.</mo><mi>z</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>z</mi></mrow></mfenced><mo>&#xD7;</mo><mfenced><mrow><mi>p</mi><mn>3</mn><mo>.</mo><mi>x</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>x</mi></mrow></mfenced><mo>-</mo><mfenced><mrow><mi>p</mi><mn>2</mn><mo>.</mo><mi>x</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>x</mi></mrow></mfenced><mo>&#xD7;</mo><mfenced><mrow><mi>p</mi><mn>3</mn><mo>.</mo><mi>z</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>z</mi></mrow></mfenced></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

• {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>a</mi><mo>=</mo><mfenced><mrow><mfenced><mrow><mi>p</mi><mn>2</mn><mo>.</mo><mi>x</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>x</mi></mrow></mfenced><mo>&#xD7;</mo><mfenced><mrow><mi>p</mi><mn>3</mn><mo>.</mo><mi>y</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>y</mi></mrow></mfenced><mo>-</mo><mfenced><mrow><mi>p</mi><mn>2</mn><mo>.</mo><mi>y</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>y</mi></mrow></mfenced><mo>&#xD7;</mo><mfenced><mrow><mi>p</mi><mn>3</mn><mo>.</mo><mi>x</mi><mo>-</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>x</mi></mrow></mfenced></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

• **平面方程的常数项**：

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>d</mi><mo>=</mo><mo>-</mo><mfenced><mrow><mi>a</mi><mo>&#xD7;</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>x</mi><mo>+</mo><mi>b</mi><mo>&#xD7;</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>y</mi><mo>+</mo><mi>c</mi><mo>&#xD7;</mo><mi>p</mi><mn>1</mn><mo>.</mo><mi>z</mi></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

以上计算结果形成一个平面方程：

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>a</mi><mi>x</mi><mo>+</mo><mi>b</mi><mi>y</mi><mo>+</mo><mi>c</mi><mi>z</mi><mo>+</mo><mi>d</mi><mo>=</mo><mn>0</mn></mstyle></math>","origin":"MathType for Microsoft Add-in"}

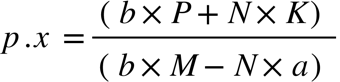
**2. Z 坐标的计算：**

在代码中直接更新了p.z的值：

p.z = p3.z + Q

**3. X 坐标的计算：**

在 get\_panel 方法中，根据其他参数计算x坐标：



其中：

• M = p3.x - p1.x

• N = p3.y - p1.y

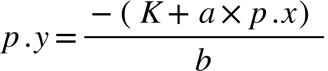
• K = c \* p.z + d

• G = (p3.z - p1.z) \* (p3.z - p.z)

• P = G + M\* p3.x + N\*p3.y

**4. Y 坐标的计算：**

最后，您根据 K 和 a 来计算 y 坐标：



### 2.3.2 地层数据网格化

### 2.3.3 模型构建

## 2.4地质模型集成

2.4.1 钻孔

2.4.2 剖面模型（勘探线、剖分的剖面）\*\*坐标转换等\*\*

2.4.3 地层模型

2.4.4 断层模型

2.4.5 .....

# 2.5 空间几何信息的计算

### 2.4.1 地层层间距计算 \*\*高程不同颜色显示\*\*（\*\*有额外的计算？三级：二级\*\*）（公式）

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>d</mi><mo>=</mo><msqrt><msup><mfenced><mrow><msub><mi>x</mi><mn>2</mn></msub><mo>-</mo><msub><mi>x</mi><mn>1</mn></msub></mrow></mfenced><mn>2</mn></msup><mo>+</mo><msup><mfenced><mrow><msub><mi>y</mi><mn>2</mn></msub><mo>-</mo><msub><mi>y</mi><mn>1</mn></msub></mrow></mfenced><mn>2</mn></msup><mo>+</mo><msup><mfenced><mrow><msub><mi>z</mi><mn>2</mn></msub><mo>-</mo><msub><mi>z</mi><mn>1</mn></msub></mrow></mfenced><mn>2</mn></msup></msqrt></mstyle></math>","origin":"MathType for Microsoft Add-in"}

### 2.4.2

# 3 三维模型可视化方法（查阅确定一下标题）10

3.1 Web可视化技术

- 3.1.1 WebGL技术

- 3.2.1 Threejs技术

## 3.2构建三维可视化场景（公式）

## 3.3模型渲染方法

- 3.3.1\*\*巷道纹理uv计算方法\*\*（公式）\*\* \*\*方法的创新点\*\*

- 3.4地质模型可视化（和3.3是否需要交换）

- 3.4.1 光照

- 3.4.2

### 3.5三维地质模型交互功能（公式）

- 4.2.1 射线追踪

- 4.2.2 坐标转换

- 4.3.1 巷道漫游

- 4.3.2 点选功能

# 4 三维可视化系统开发

（部分伪代码）

## 4.1系统架构设计

- 4.1.1 体系结构（原生H5开发\*\*系统升级优化\*\*为VUE3的框架开发）

- 4.1.2 技术流程

## 4.2系统功能实现（伪代码）

- 4.2.1

- 4.2.1

- 系统功能测试

\*\*创新点：系统更新换代 优化\*\* 提高开发效率、项目运行流畅等。

代码复用 （数据结构 可视化方式 api等等）

# 5、应用实例

# 6、结论与展望

- 6.1 结论

- 6.2 展望