

# PSMesher使用文档

## 1. PSMesher介绍

泊松网格引擎PSMesher导入STEP文件， 支持2D、3D网格的快速生成，能够设置最小最大单元尺寸，并输出BDF文件。

## 2. 头文件PSMesher.h

使用泊松网格引擎必须包含头文件PSMesher.h，此头文件包含所有数据结构和API定义。

## 3. 枚举定义

### 3.1 2D网格算法枚举MeshAlgorithm2D

```
enum class MeshAlgorithm2D
{
    MeshAdapt = 1,
    Automatic = 2,
    Delaunay = 5
};
```

| 枚举值       | 说明                                       |
|-----------|------------------------------------------|
| MeshAdapt | 此算法基于局部网格修改                              |
| Automatic | 自动算法对于平面使用Delaunay算法，对于其他曲面使用MeshAdapt算法 |
| Delaunay  | Delaunay三角分解算法                           |

### 3.2 3D网格算法枚举MeshAlgorithm3D

```
enum class MeshAlgorithm3D
{
    Delaunay = 1,
    Frontal = 4
};
```

| 枚举值      | 说明                       |
|----------|--------------------------|
| Delaunay | 3D Delaunay三角分解算法        |
| Frontal  | J. Schoeberl' s Netgen算法 |

### 3.3 网格维数控制枚举MeshDimension

```
enum class MeshDimension
{
    Mesh0D = 0,
    Mesh1D = 1,
    Mesh2D = 2,
    Mesh3D = 3
};
```

| 枚举值    | 说明             |
|--------|----------------|
| Mesh0D | 生成0D网格，只生成Node |
| Mesh1D | 生成1D线网格        |
| Mesh2D | 生成2D面网格        |
| Mesh3D | 生成3D体网格        |

## 4 全局函数PSGetMesher

```
PS_MESH_EXPORT PSMesher* PSGetMesher();
```

获取全局网格生成器，使用完后需要调用PSMesher::destroyMesher销毁。

## 5 网格生成器接口类PSMesher

```
class PS_MESHER_EXPORT PSMesher
{
public:
    virtual void importModel(const std::string &filePath) = 0;
    virtual void clearModel() = 0;
    virtual void setAlgorithm2D(const MeshAlgorithm2D alg2d) = 0;
    virtual void setAlgorithm3D(const MeshAlgorithm3D alg3d) = 0;
    virtual void setMeshSizeMin(const double val) = 0;
    virtual void setMeshSizeMax(const double val) = 0;
    virtual void setMeshDimension(const MeshDimension dim) = 0;
    virtual void generateMesh() = 0;
    virtual void exportMesh(const std::string& filePath) = 0;
    virtual void destroyMesher() = 0;
};
```

| API              | 说明                 |
|------------------|--------------------|
| importModel      | 导入模型文件，目前只支持STEP文件 |
| clearModel       | 清空模型               |
| setAlgorithm2D   | 设置2D网格算法           |
| setAlgorithm3D   | 设置3D网格算法           |
| setMeshSizeMin   | 设置最小单元尺寸           |
| setMeshSizeMax   | 设置最大单元尺寸           |
| setMeshDimension | 设置网格维数             |
| generateMesh     | 按要求生成网格            |
| exportMesh       | 输出网格到BDF文件         |
| destroyMesher    | 销毁网格生成器            |

## 6 一个完整例子

```

#include <PSMesher.h>
#include <exception>
#include <iostream>

int main(int argc, char **argv)
{
    try
    {
        PSMesher* mesher = PSGetMesher();
        mesher->importModel(std::string("D:\\mesh\\stp\\Car_Parts.step"));
        mesher->setMeshDimension(MeshDimension::Mesh2D);
        mesher->setAlgorithm2D(MeshAlgorithm2D::Automatic);
        mesher->setMeshSizeMin(500);
        mesher->setMeshSizeMax(500);
        mesher->generateMesh();
        mesher->exportMesh("T13.bdf");
        mesher->destroyMesher();

        mesher = PSGetMesher();
        mesher->importModel(std::string("D:\\mesh\\stp\\Car_Parts.step"));
        mesher->setMeshDimension(MeshDimension::Mesh2D);
        mesher->setAlgorithm2D(MeshAlgorithm2D::Automatic);
        mesher->setMeshSizeMin(50);
        mesher->setMeshSizeMax(50);
        mesher->generateMesh();
        mesher->exportMesh("T15.bdf");
        mesher->destroyMesher();
    }
    catch (std::exception& e)
    {
        std::cout << e.what();
    }
    return 0;
}

```