

EITRay 声学软件

Version 1.0

用户手册

EIT 流体与声学实验室

目录

一、 总体功能介绍	1
二、 快速入门	2
2.1 安装与运行	2
2.2 输入文件详解	7
2.2.1 atmosphere.dat	7
2.2.2 GB_input.dat	8
2.2.3 observer_location-x.vtk/.dat	9
2.2.4 RT_input.dat	11
2.2.5 source_location.dat	12
2.2.6 source_directivity-x.dat/source_directivity-n_m.dat	12
2.2.7 Building.stl、Terrain.stl、Road.stl、Water.stl、Tree.stl	13
2.3 输出文件详解	13
2.4 术语描述	14
二、 声学射线追踪理论（忽略）	14
三、 基本算例	14
3.1 平面场	14
3.1.1 简介	14
3.1.2 输入文件设置	14
3.1.3 输出详情	16
3.2 室内声场计算	17
3.2.1 简介	17
3.2.2 输入文件设置	18
3.2.3 输出详情	18
3.3 室外声场计算	18
3.3.1 简介	18
3.3.2 输入文件设置	18
3.3.3 输出详情	18

一、总体功能介绍

该软件使用的 Gaussian beam tracing (GBT) 算法是一种常用的声场计算方法，它可以用来模拟声波在复杂环境中的传播和反射。这种算法的基本思想是将声波看作是由许多射线组成的，然后利用射线的传播规律来计算声波在空气中的传播和反射。GBT 算法的优点在于它可以快速计算声波在复杂环境中的传播和反射，而且计算结果精度较高。

该软件的输入包括四个部分：计算类型、源数及其各自的位置、大气条件和环境几何文件。其中，计算类型指的是声场计算的类型，可以是室内或室外、单频或多频等等。源数及其各自的位置指的是声源的数量和位置信息，这些信息对声场计算的结果有很大的影响。

大气条件包括空气温度、相对湿度和压力，这些参数会影响声波在大气中的传播速度和衰减。环境几何文件包括建筑物、地形、道路、水、树等物体的几何信息，格式为 STL 格式。这些信息可以用来模拟声波在不同环境中的传播和反射，从而得到声场的分布情况。

除了以上的输入信息外，该软件还需要一些射线追踪相关的参数，例如维数、方位角、射线总数、射线是否压缩等等。这些参数会影响声场计算的精度和计算时间。维数指的是声场计算的维度，可以是二维或三维，方位角指的是声源的方向，射线总数指的是计算时使用的射线数量，射线是否压缩指的是在计算过程中是否对射线进行压缩，以提高计算效率。

最终，该软件可以输出三维可渲染可视化的 VTK 文件，这个文件可以用来对声场进行可视化展示。VTK 文件是一种常用的三维可视化文件格式，可以在许多软件中进行打开和编辑。通过对声场进行可视化展示，用户可以更加直观地了解声场的分布情况和声波在不同环境中的传播和反射规律，从而更好地进行声学设计和优化。

二、快速入门

2.1 安装与运行

本程序尚未创建桌面应用的功能，所有的操作全部在 **Ubuntu** 系统下执行，这里推荐选择 **WSL**（**windows** 的 **linux** 子系统,且要 **2.0** 及以上的内核）对本程序进行操作，如还没有安装，请自行搜索安装。

本程序需要的依赖有：**gcc**（版本 **8** 及以上）、**Cmake**（**3.10** 版本以上）、**MPI**，请提前安装后再对本程序进行运行。

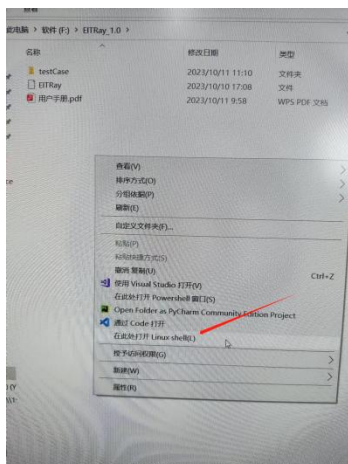
在收到本软件时，应该是一个命名为 **EITRay_1.0** 的压缩包，将其解压在自己想要的路径下。

此电脑 > 软件 (F:) > EITRay_1.0 >

所看到的，就是以下文件：

名称	修改日期	类型	大小
testCase	2023/10/11 11:24	文件夹	
City_case.sh	2023/10/11 11:31	SH 源文件	1 KB
Dockerfile	2023/10/11 16:24	文件	1 KB
EITRay	2023/10/10 17:08	文件	11,473 KB
Freeplane_case.sh	2023/10/11 11:31	SH 源文件	1 KB
room_SoundField_case.sh	2023/10/11 11:31	SH 源文件	1 KB
用户手册.pdf	2023/10/11 14:46	WPS PDF 文档	1,533 KB

按 **shift** 键+右击，你能看到如下选项：



点击进入，即可看到：



接下来需要安装已有的 docker 环境了。

安装 docker 环境

1 简介

Docker 是一个开源的应用容器化平台，它允许开发者将应用程序及其环境依赖打包到一个容器中，这个容器可以在任何安装了 docker 的机器上运行，无论这个机器的环境如何。这极大地简化了应用程序的部署、扩展和管理。

2 安装教程

<https://zhuanlan.zhihu.com/p/651148141>

3 导入已有的 docker 包环境

Docker 包为：

名称	修改日期	类型	大小
bin	2023/10/13 10:38	文件夹	
include	2023/10/13 10:38	文件夹	
out	2023/10/13 10:38	文件夹	
src	2023/10/13 10:38	文件夹	
testCase	2023/10/13 10:38	文件夹	
City_case.sh	2023/10/11 11:31	SH 文件	1 KB
CMakeLists.txt	2023/10/11 15:53	文本文档	2 KB
Dockerfile	2023/10/11 16:24	文件	1 KB
eeitray.tar	2023/10/11 17:09	360压缩	907,569 KB
EITRay	2023/10/10 17:08	文件	11,473 KB
Freeplane_case.sh	2023/10/11 11:31	SH 文件	1 KB
room_SoundField_case.sh	2023/10/11 11:31	SH 文件	1 KB
用户手册.pdf	2023/10/11 14:46	WPS PDF 文档	1,533 KB

然后在当前文件夹位置下执行命令：

```
docker load -i eeitray.tar
```

会有如下结果：

这个命令将读取 tar 文件并加载镜像到你的 Docker 实例。

```
dcszhang@WIN-1BGMU00KOIG:/mnt/c/Users/Administrator/Desktop/EITRay_1.0$ docker load -i eeitray.tar
01d4e4b4f381: Loading layer [=====>] 80.35MB/80.35MB
266af7f20055: Loading layer [=====>] 520.8MB/520.8MB
d2d9ba0918ba: Loading layer [=====>] 1.536kB/1.536kB
780babb38b8e: Loading layer [=====>] 225.2MB/225.2MB
11a71497d483: Loading layer [=====>] 8.546MB/8.546MB
83a18d2335d8: Loading layer [=====>] 94.36MB/94.36MB
Loaded image: eeitray:latest
```

4 运行

一旦镜像加载完成，你可以使用 `docker images` 命令来查看本地可用的镜像列表，确认你的镜像是否已经成功加载。

```
dcszhang@WIN-1BGMU00KOIG:/mnt/c/Users/Administrator/Desktop/EITRay_1.0$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
eeitray latest de4816ab8eb4 42 hours ago 913MB
hello-world latest 9c7a54a9a43c 5 months ago 13.3kB
```

然后再进入自己的镜像即可。

执行命令：

```
docker run -v /mnt/f/EITRay_1.0:/app -it eeitray
```

这个命令的中间那部分：

```
(base) dcszhang@dcszhang:/mnt/f/EITRay_1.0$ docker run -v /mnt/f/EITRay_1.0:/app -it eeitray
root@f53685a8blac:/app# ./Freeplane_case.sh
“-----”
Welcome to try EITRay Acoustic Software
“-----”
Enter the number of processors: 10
=====
minis has detected an attempt to run on root
```

而那个 app 是这个：

```
dcszhang@WIN-1BGMU00KOIG:/mnt/c/Users/Administrator/Desktop/EITRa
root@cd106a162ec1:/app# ./Freeplane_case.sh
“-----”
Welcome to try EITRay Acoustic Software
“-----”
```

这个部分是为了将容器的文件系统映射到宿主句目录的。

即可进入已配置好环境的镜像：

```
dcszhang@WIN-1BGMU00KOIG:/mnt/c/Users/Administrator/Desktop/EITRay_1.0$ docker run -it eeitray
root@cd106a162ec1:/app# ./Freeplane_case.sh
“-----”
Welcome to try EITRay Acoustic Software
“-----”
Enter the number of processors: 10
=====
EITRay
=====
total proc count is :10
Buildings: OFF
Terrain: OFF
Road: OFF
Water: OFF
Tree: OFF
Reading file: Control/RT_input.dat
=====
```

此时，执行此命令：（程序的启动脚本）3 个例子

./Freeplane_case.sh 或者

./City_case.sh 或者

./room_SoundField_case.sh

即可运行该程序，如果顺利的话，会看到以下界面：

```
root@cd106a162ec1:/app# ./Freeplane_case.sh
"-----"
      Welcome to try EITRay Acoustic Software
"-----"
Enter the number of processors: 10
```

因为程序采用 MPI 并行计算的模式，所以需要输入参与计算的 CPU 核心数，以 core i7-12700

CPU 为例，此时可以输入 10 个核心数参与计算和运行。

输入之后，进入正式的程序计算页面：

```
total proc count is :1
Buildings: ON
Terrain: ON
Road: ON
Water: OFF
Tree: OFF
Reading file: Control/RT_input.dat
=====
                        3D Run with Ray Compression
=====
MODE: RT + GB
Aprx. ray distance: 2720
Output STL zones: 0
Output Ray Paths: 0
Particle Tracker: 0, 10
Reading file: Control/source_location.dat
Reading GB_input file...
Beam parameter is -45874i.
Reading file: Control/observer_location-XX.dat
=====
Memory usage report
=====
MByte for single ray:(MB) 1.28
MByte for total case:(MB) 327.68
=====
Reading dir. file (No.1 source, No.1 frequency)...
Reading dir. file (No.1 source, No.2 frequency)...
setup completed in      : 0.00
MPI-ray distribution complete!
```

以下为对该运行页面的说明：

(1) **total proc count is :1**

这说明了总共参与计算的核心数设置为了 1；

(2)

```
Buildings: ON
Terrain: ON
Road: ON
Water: OFF
Tree: OFF
```

该程序共设置了 5 个 STL 元素种类，分别是建筑（buildings）、地形（Terrain）、路面（Road）、水面（Water）、树面（Tree），后面跟着的 ON/OFF，是程序是否检测到了对应的 STL 元素的结果显示，OFF 为未检测到相关的 STL 元素，ON 为已检测到相关的 STL 元素。

(3)

```
Reading file: Control/RT_input.dat
```

这个说明了输入的文件在当前文件夹下的 Control 文件夹。

(4)

```
=====
                        3D Run with Ray Compression
=====
MODE: RT + GB
Aprx. ray distance: 2720
Output STL zones: 0
Output Ray Paths: 0
Particle Tracker: 0, 10
Reading file: Control/source_location.dat
Reading GB_input file...
Beam parameter is -45874i.
Reading file: Control/observer_location-XX.dat
```

这个说明了一些设置的声学参数，MODE 是 RT（求得射线追踪路径）+GB（求得观测点声压），2720 是每根射线的长度上限。以下的 0 意味着不对 STL、射线路径进行一个单独的输出。射线参数被设置为了-45874i。

(5)

```
=====
Memory usage report
=====
MByte for single ray: (MB) 1.28
MByte for total case: (MB) 327.68
=====
```


这一点是对程序内存占用的说明，一根射线和所有射线的一个占比情况。

之后等待程序运行结束就 ok 啦。

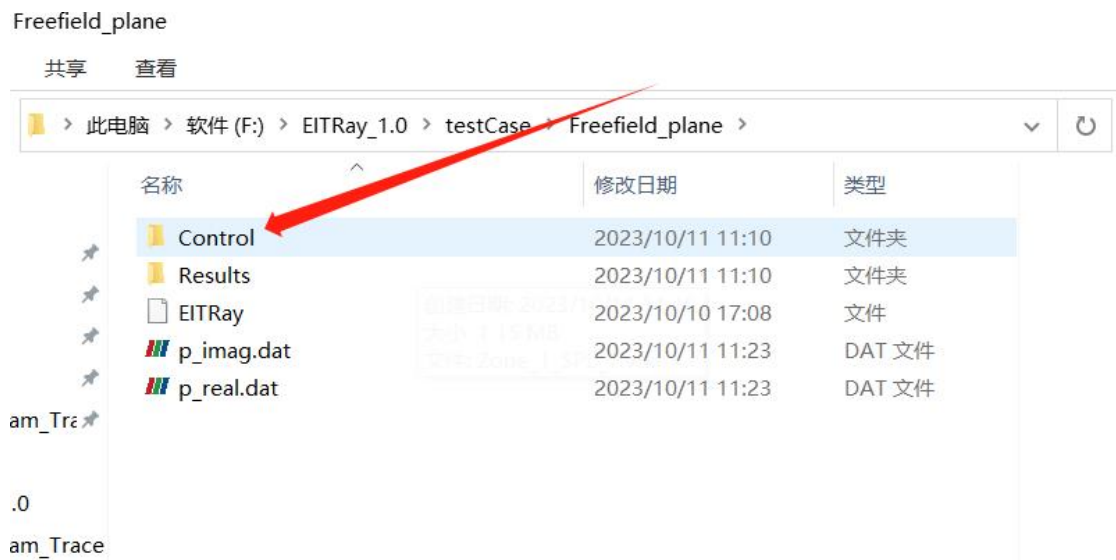
```

Estimated time : 0.00m
ray tracing completed in: 1.09
0.00
GB completed in: 1.09
=====
Program END. Aborting code ...
=====









```

2.2 输入文件详解

本程序的输入文件被保存在 **Control** 这个文件夹：



打开之后，会看到以下几种文件：

 atmosphere.dat	2021/8/10 16:35	DAT 文件	1 KB
 GB_input.dat	2021/8/10 16:35	DAT 文件	1 KB
 observer_location-1.vtk	2021/8/10 16:35	VTK 文件	646 KB
 RT_input.dat	2021/8/10 16:35	DAT 文件	1 KB
 source_directivity-1.dat	2021/8/10 16:35	DAT 文件	24 KB
 source_location.dat	2021/8/10 16:35	DAT 文件	1 KB
 Terrain-0.stl	2021/8/10 16:35	STL 文件	1 KB
 Terrain-1.stl	2021/8/10 16:35	STL 文件	245 KB

2.2.1 atmosphere.dat

大气条件文件，在这里指定了大气条件，包括空气温度（C）、相对湿度（%）、压力（atm）。大气衰减根据大气条件计算。



```

atmosphere.dat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
20 ! air temperature(C)
70 ! Relative humidity(%)
1 ! pressure(atm)

```

2.2.2 GB_input.dat

为高斯光束输入文件，在其中要编写原频率的数量和要测试的频率列表。如果源频率的数量大于 1，则代码将运行宽带模拟。对于宽带情况，输出是具有不相关求和的总 SPL。观察者格式有两个选项：

VTK、TEC，即.vtk 文件和.dat 文件。而观察者文件和输出的 SPL 文件的格式应该保持相同。输出观察区域的数量等于观察者位置输入文件的数量。

VTK 格式如果有 3 个观察区域，则应在 Control 文件夹中包括输入文件：

Observer_location-1.vtk、observer_location-2.vtk、observer_location-3.vtk

其中的光束参数，是控制光束宽度和沿中心光线的曲率演变的常数。自适应光束参数被应用，不受此输入数字的影响。如果要使用手动定义的光束参数，应该修改 GBvars.cpp 中的源代码。只有一个负常数才能生成有限的宽度光束。一般来说，具有较大幅度的光束常数将在源区域附近给出相对较宽的光束，但沿传播方向缓慢增加。小幅度将在开始时具有较窄的光束，但增加会非常宽，这可能导致长距离传播的大偏差。

下图为不同反射表面的常用值：

Description of surface	Flow resistivity cgs rayls (1 cgs rayls = 1000 Pa s/m ²)
Dry snow, new fallen 0.1 m over about 0.4-m older snow	10–30
Sugar snow	25–50
In forest, pine, or hemlock (Ref. 22)	20–80
Grass: rough pasture, airport, public buildings, etc.	150–300
Roadside dirt, ill-defined, small rocks up to 0.1-m mesh	300–800
Sandy silt, hard packed by vehicles	800–2500
“Clean” limestone chips, thick layer (0.01- to 0.025-m mesh)	1500–4000
Old dirt roadway, fine stones (0.05-m mesh) interstices filled	2000–4000
Earth, exposed and rain-packed	4000–8000
Quarry dust, fine, very hard-packed by vehicles	5000–20 000
Asphalt, sealed by dust and light use	~ 30 000
Upper limit set by thermal-conduction and viscous boundary layer	2×10^5 to 1×10^6

具体的内部文件详情如下图所示：

```

GB_input.dat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
1                               !number of source freq.
100
VTK                             !observer file format
VTK                             !output SPL file format
1                               !number of observer zones for output
-45874                          !IMAG component of beam parameter
=====
Flow reactance of different zones (<0 == hard wall)
=====
Building  Terr   Road   Water   Tree
-10       -10    -10    -10     -10
1650      1650   1650   0        0
1500      1500   1500   0        0

```

头两行是声源数和源频率。

而在最后三行中，第一行是流阻抗、第二行是声速值、第三行是密度。

2.2.3 observer_location-x.vtk/.dat

观察者位置文件包含点坐标和三角形元素的信息。对于 VTK 格式，应该使用由 **Praview** 生成的 **Ascii** 类型。对于 **DAT** 格式，首先提供观察者数量及其坐标，然后是三角形数量和顶点索引。详情如下图所示：

observer_location-1.vtk - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

vtk DataFile Version 4.0

vtk output

ASCII

DATASET POLYDATA

POINTS 10201 float

0 0 3 0 0.5 3 0 1 3

0 1.5 3 0 2 3 0 2.5 3

0 3 3 0 3.5 3 0 4 3

0 4.5 3 0 5 3 0 5.5 3

0 6 3 0 6.5 3 0 7 3

0 7.5 3 0 8 3 0 8.5 3

50 44 3 50 44.5 3 50 45 3

50 45.5 3 50 46 3 50 46.5 3

50 47 3 50 47.5 3 50 48 3

50 48.5 3 50 49 3 50 49.5 3

50 50 3

POLYGONS 20000 80000

3 0 1 101

3 102 101 1

3 1 2 102

3 103 102 2

3 2 3 103

3 104 103 3

3 3 4 104

3 105 104 4

3 4 5 105

3 106 105 5

3 5 6 106

3 107 106 6

3 6 7 107

3 108 107 7

13 10200 10199 10099

POINT_DATA 10201
NORMALS Normals float[illegible]
$$\begin{bmatrix} 0 & 0 & -1 \end{bmatrix}$$

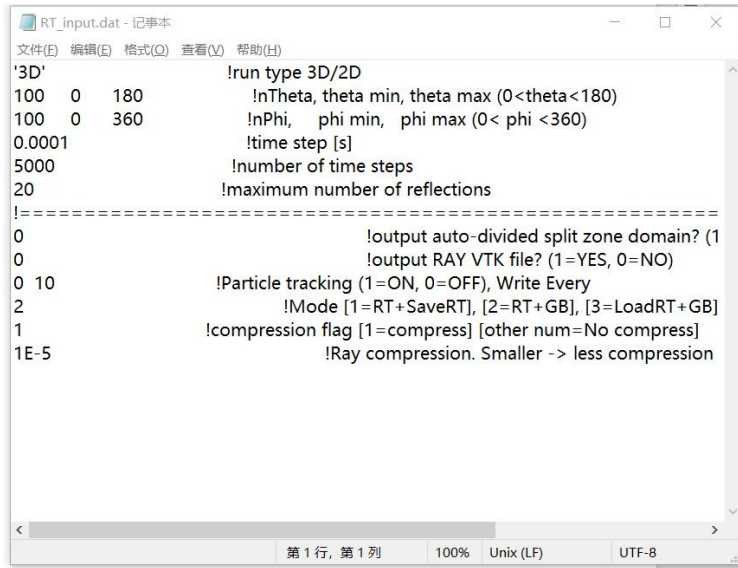
TEXTURE_COORDINATES TextureCoordinates 2 float

```

0.0 0.01 0.02 0.03 0.04
0.05 0.06 0.07 0.08 0
0.09 0.1 0.11 0.12 0.13
0.14 0.15 0.16 0.17 0
0.18 0.19 0.2 0.21 0.22
0.23 0.24 0.25 0.26 0
0.27 0.28 0.29 0.3 0.31
0.32 0.33 0.34 0.35 0
0.36 0.37 0.38 0.39 0.4
0.41 0.42 0.43 0.44 0
0.45 0.46 0.47 0.48 0.49
0.5 0.51 0.52 0.53 0
0.54 0.55 0.56 0.57 0.58
0.59 0.6 0.61 0.62 0
0.63 0.64 0.65 0.66 0.67
0.68 0.69 0.7 0.71 0
0.72 0.73 0.74 0.75 0.76
0.77 0.78 0.79 0.8 0
0.81 0.82 0.83 0.84 0.85
0.86 0.87 0.88 0.89 0
0.9 0.91 0.92 0.93 0.94
0.95 0.96 0.97 0.98 0
0.99 0 1 0 0.01 0.01 0.01 0.02
0.01 0.03 0.01 0.04 0.01 0.05 0.01 0.06 0.01
0.07 0.01 0.08 0.01 0.09 0.01 0.1 0.01 0.1
0.11 0.12 0.01 0.13 0.01 0.14 0.01 0.15 0.01
0.16 0.01 0.17 0.01 0.18 0.01 0.19 0.01 0.2
0.21 0.21 0.01 0.22 0.01 0.23 0.01 0.24 0.01
0.25 0.01 0.26 0.01 0.27 0.01 0.28 0.01 0.29
0.31 0.3 0.01 0.31 0.01 0.32 0.01 0.33 0.01
0.34 0.01 0.35 0.01 0.36 0.01 0.37 0.01 0.38
0.39 0.01 0.39 0.01 0.4 0.01 0.41 0.01 0.42 0.01
0.43 0.01 0.44 0.01 0.45 0.01 0.46 0.01 0.47
0.48 0.01 0.49 0.01 0.5 0.01 0.51 0.01

```

2.2.4 RT_input.dat



```

RT_input.dat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
'3D'      !run type 3D/2D
100  0   180      !nTheta, theta min, theta max (0<theta<180)
100  0   360      !nPhi,  phi min,  phi max (0< phi <360)
0.0001      !time step [s]
5000      !number of time steps
20      !maximum number of reflections
!=====
0      !output auto-divided split zone domain? (1
0      !output RAY VTK file? (1=YES, 0=NO)
0 10      !Particle tracking (1=ON, 0=OFF), Write Every
2      !Mode [1=RT+SaveRT], [2=RT+GB], [3=LoadRT+GB]
1      !compression flag [1=compress] [other num=No compress]
1E-5      !Ray compression. Smaller -> less compression

```

射线追踪输入文件，第一行是运行类型参数指定在垂直平面（2D）或者完整的 3D 空间中进行射线的发射。初始射线发射方向由仰角（**theta**）和方位角（**phi**）方向的细分数以及相应的角度范围定义。总射线数为 **ntheta** 与 **nphi** 的乘积。两个方向中的角度范围应该包含在源文件中定义的角度范围。每个射线追踪时间步长是均匀的。当射线总数或反射次数超出在此文件中定义的限制时，射线追踪过程将结束。对于射线文件的输出，标志应为 **1**。对于射线压缩选项，**1** 表示使用压缩技术。以下射线压缩参数设置了射线压缩的阈值，即相邻两个射线追踪步骤之间的方向变化。较小的射线压缩阈值会导致会导致较小的射线压缩。

2.2.5 source_location.dat



```

source_location.dat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
'A'
1
30 30 5

```

源位置文件定义了计算类型、源数量和它们各自的位置。

对于第一行计算类型，如果为‘**M**’，代码将输出每个源位置的结果，如果计算类型为‘**A**’，则输出总体的结果，其中包括所有源的贡献之和。

第二行是声源的数量，而第三行是声源所处的坐标。

2.2.6 source_directivity-x.dat/source_directivity-n_m.dat

对于单一声源来说，详细设置的文件名应该为 `source_directivity-x.dat`，而对于多声源来说，文件名应该命名为 `source_directivity-n_m.dat`，其中 `n` 是源的数量，`m` 是频率分量的数量。

*source_directivity-1.dat - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```

31 !nTheta
0.00000 0.10472 0.20944 0.31416 0.41888 0.52360 0.62832
61 !nPhi
0.00000 0.10472 0.20944 0.31416 0.41888 0.52360 0.62832
!directivity
1      1      1      1      1      1      1      '
1      1      1      1      1      1      1      '
1      1      1      1      1      1      1      '
1      1      1      1      1      1      1      '
!phase
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000

```

在前四行中定义了仰角 (`theta`) 的数量、方位角 (`phi`) 的数量以及它们各自的角度值（以弧度为单位）。在之后的内容中列出了辐射球上压力场对应表格的幅值和相位。

2.2.7 Building.stl、Terrain.stl、Road.stl、Water.stl、Tree.stl

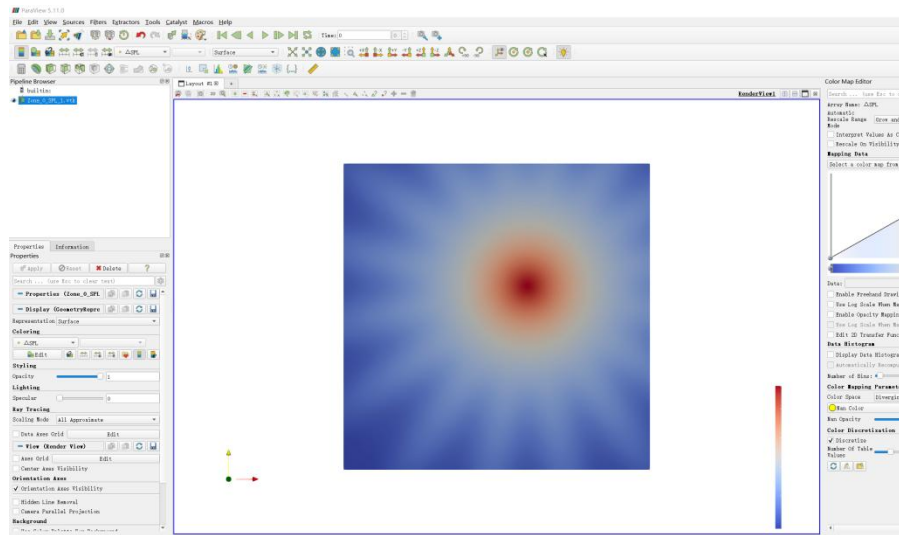
环境几何文件作为反射的障碍物读入到代码中。应该使用二进制 `STL` 文件。用户应该将几何文件重命名为上述列表中选择的文件名。如果找不到上述名称的文件，则将执行自由场的模拟。

2.3 输出文件详解

最终程序会输出一个包含 `SPL` 信息的 `VTK\TEC` 文件，其具体为：`Zone_m_SPL_n.vtk/.dat`，其格式对应于输入观察者文件，`m` 和 `n` 是两个整数，分别区分不同的观察者位置区域和不同的源位置。

 Zone_0_SPL_1.dat	2023/10/9 10:42	DAT 文件	832 KB
 Zone_0_SPL_1.vtk	2023/9/22 17:34	VTK 文件	553 KB

该文件推荐用 **Paraview** 软件进行打开，其打开后页面如下：



2.4 术语描述

该程序中的距离单位是米，时间的单位是秒。程序输出的是声压级。

二、声学射线追踪理论（忽略）

三、基本算例

在本节中，将使用平面场、室内声场、室外声场三个场景来展示如何运行该程序，以及如何使用 Paraview 处理获取的 SPL 结果。

3.1 平面场

3.1.1 简介

平面声场是指声波在某一特定方向上传播，而在垂直于该方向的平面内，声波的所有相关参数（例如声压、声速度）都保持不变的场景。

3.1.2 输入文件设置

（1）对于 RT_input.dat 这个文件来说，

'3D'	!run type 3D/2D
100 0 180	!NTheta, theta min, theta max (0<theta<180)
100 0 360	!NPhi, phi min, phi max (0< phi <360)
0.0001	!time step [s]
5000	!number of time steps
20	!maximum number of reflections
!=====!	
0	!output auto-divided split zone domain? (1=YES, 0=NO)
0	!output RAY VTK file? (1=YES, 0=NO)

0 10 !Particle tracking (1=ON, 0=OFF), Write Every
 2 !Mode [1=RT+SaveRT], [2=RT+GB], [3=LoadRT+GB]
 1 !compression flag [1=compress] [other num=No
 compress]
 1E-5 !Ray compression. Smaller -> less compression but more accurate.

(2) 对于 GB_input.dat 这个文件来说,

1 !number of source freq.
 100 ! source freq.
 VTK !observer file format
 VTK !output SPL file format
 1 !number of observer zones for output
 -45874 !IMAG component of beam parameter

=====

Flow reactance of different zones (<0 == hard wall)

=====

Building	Terr	Road	Water	Tree
-10	-10	-10	-10	-10
1650	1650	1650	0	0
1500	1500	1500	0	0

(4) 对于 source_location.dat 这个文件来说,

'A'
 1
 30 30 5

(5) 对于 source_directivity-1.dat 这个文件来说。

31 lnTheta

0.00000	0.10472	0.20944	0.31416	0.41888	0.52360	0.62832
0.73304	0.83776	0.94248	1.04720	1.15192	1.25664	1.36136
1.46608	1.57080	1.67552	1.78024	1.88496	1.98968	2.09440
2.19911	2.30383					

```

2.40855  2.51327  2.61799  2.72271  2.82743  2.93215  3.03687  3.14159
    61      lnPhi
    0.00000  0.10472  0.20944  0.31416  0.41888  0.52360  0.62832
0.73304  0.83776  0.94248  1.04720  1.15192  1.25664  1.36136  1.46608
1.57080  1.67552  1.78024  1.88496  1.98968  2.09440  2.19911  2.30383
2.40855  2.51327  2.61799  2.72271  2.82743  2.93215  3.03687  3.14159
3.24631  3.35103  3.45575  3.56047  3.66519  3.76991  3.87463  3.97935
4.08407  4.18879  4.29351  4.39823  4.50295  4.60767  4.71239  4.81711
4.92183  5.02655  5.13127  5.23599  5.34071  5.44543  5.55015  5.65487
5.75959  5.86431  5.96903  6.07375  6.17847  6.28319

!directivity
1  1  1  ...

!phase
0.00000  0.00000  0.00000 ...

```

(6) 对于 `observer_location-1.vtk` 这个文件来说，


需要去在 STL 文件中通过 `paraview` 自行导出，平面场参考示例文件 `testcase/Freefield_plane/Control/observer_location-1.vtk`。

(7) 对于环境元素 STL 这个文件来说，

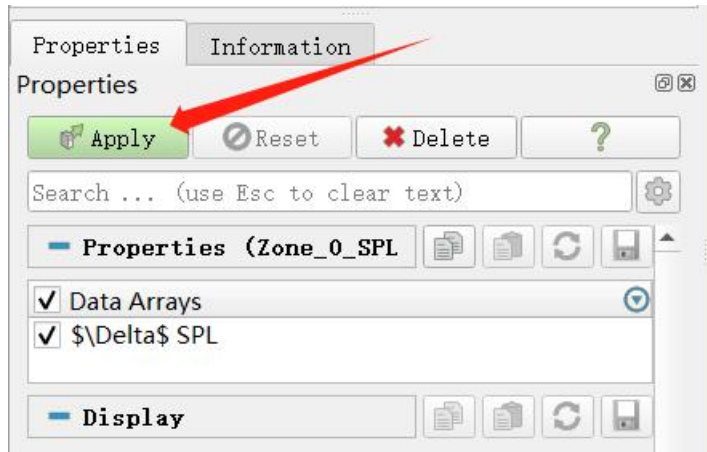
在平面场中不需要进行一个设置

3.1.3 输出详情

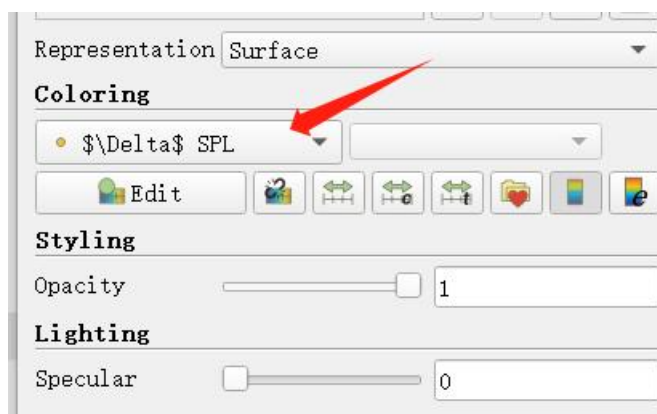
当程序计算完成之后，会看到一个文件：

 Zone_0_SPL_1.dat	2023/10/11 8:55	DAT 文件	813 KB
--	-----------------	--------	--------

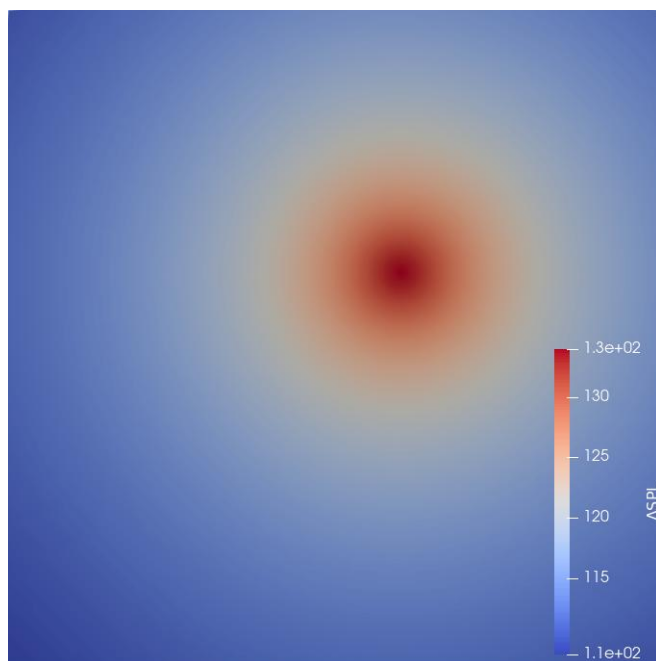
用 `Paraview` 打开后：



颜色显示选择:



会看到:



以上为这个程序所生成的平面声场。

3.2 室内声场计算

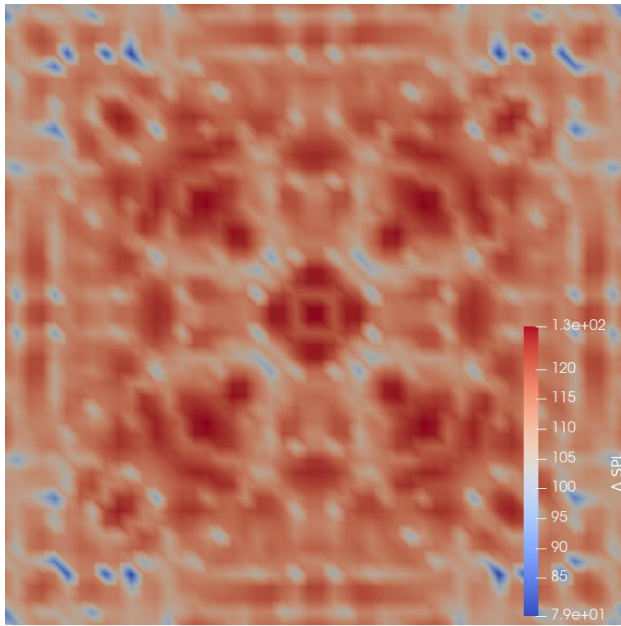
3.2.1 简介

室内声场是指在有限封闭空间（如房间、大厅或剧院）内的声学环境，其中声波与空间的边界（墙、天花板和地板）以及其中的物体相互作用。

3.2.2 输入文件设置

在 `source_location.dat` 文件中将平面场声源位置修改为 `(0, 0, 0)`，然后加入一个房间的元素 `building.stl`，并且通过 `paraview` 生成一个新的观测点文件即可，具体的示例文件在 `testcase/room_SoundField/Control`。

3.2.3 输出详情



以上为这个程序所生成的室内声场。

3.3 室外声场计算

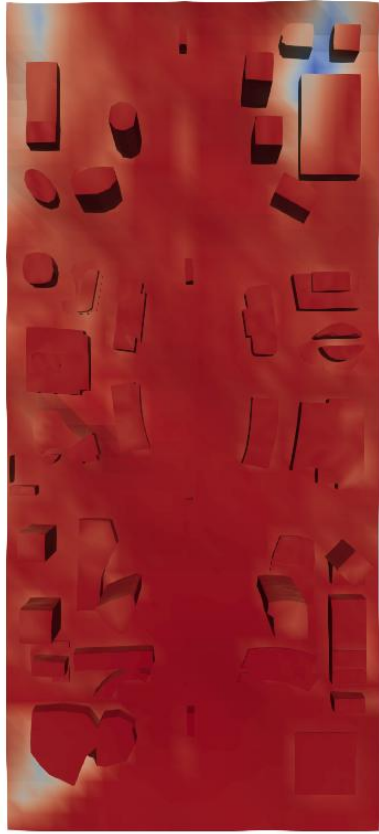
3.3.1 简介

室外声场是指在开放空间中的声学环境，不受封闭结构的限制。它设计声波与地形、建筑物、植被、湖泊等物体和介质的相互作用。

3.3.2 输入文件设置

具体的示例输入文件在 `testcase/City_case/Control` 中。

3.3.3 输出详情



以上为这个程序所生成的室外声场。