

ADVENTURE_Sound

Steady / Non-steady Sound Analysis with HDDM

プログラム使用マニュアル

Version: 0.2b

September, 2024

1. はじめに

本書は ADVENTURE Project において開発中[1]の、波動音響解析のための有限要素解析ソルバ、ADVENTURE_Sound の使用マニュアルである。

1 章では ADVENTURE_Sound の概要及び実行までの操作手順を説明し、2 章以降では本プログラムの解析機能等を紹介する。

1.1. 本モジュールの特徴

ADVENTURE_Sound は以下のような特徴をもっている。

- 定常・非定常問題の解析が可能
- 4 面体 1 次要素と 4 面体 2 次要素が使用可能
- 階層型領域分割法 (HDDM) による負荷分散を行った並列処理が可能
- 線形ソルバにバランシング領域分割法 (BDD) [3]が使用可能

1.2. 稼働環境

本プログラムは以下の環境で動作確認を行っている。

対応プラットフォーム	Linux
並列通信ライブラリ	MPI

ADVENTURE_Sound では、複素数を用いた計算に c99 から用いられる `double _Complex` 型を用いており、そのため MPI の通信に用いる関数 “MPI_C_DOUBLE_COMPLEX” が必要である。MPI2 もしくは、MPI2.1 以降の MPI を用いる必要がある。

1.3. コンパイルとインストール

1.3.1. コンパイル

ADVENTURE_Sound モジュールをコンパイルするには、C コンパイラ、MPI のコンパイル環境、及び ADVENTURE_IO のインストールが必要である。

ADVENTURE_Sound モジュールのコンパイルは以下の手順で行う。

(1) アーカイブファイルの展開

```
% tar -zxvf Advsound-1.0.tar.gz
```

ただし、“%” はコマンドプロンプトを表しているため、実際の入力は必要ない。アーカイブを展開することで、ADVENTURE_Sound_1.0 のディレクトリが作成される。また、ADVENTURE_Sound は次のサブディレクトリを持つ。

サブディレクトリ名	内容
hddmsrc	ADVENTURE_Sound のソースファイル
doc	ドキュメント類
tools	境界条件設定ツール
libfem	有限要素ライブラリ

(2) 展開したディレクトリに移動し、コンパイルを行う。

```
% cd AdvSound-1.0
```

```
% ./configure
```

```
% make
```

configure は、環境に依存する部分を解決し、適切な **Makefile** を作成するためのシェルスクリプトである。configure に渡せる主なオプションを以下に示す。ただし、以下で用いるディレクトリの指定には、絶対パスを指定する必要がある。

`--with-advio=directory`

ADVENTURE_IO がインストールされたディレクトリを指定する。デフォルトは "\$HOME/ADVENTURE" である。

`--with-mpicc=command`

MPI のための C コンパイラのコマンド名を指定する。デフォルトは "mpicc" である。並列版の ADVENTURE_Sound をコンパイルするには、MPI 用 C コンパイラが必須である。

`--prefix=`

ADVENTURE_Sound のインストール先を指定する。デフォルトは "\$HOME/ADVENTURE" である。

1.3.2. インストール

コンパイルに成功したのち、以下のコマンドによりインストールが行われる。

```
% make install
```

デフォルトのインストール先は \$HOME/ADVENTURE/ である。インストール先の変更は以下のように行う。

```
% make install prefix = < install_dir >
```

ただし、インストール先のディレクトリへの書き込み権限をもったユーザによって行う必要がある。

- bin/advsound-s シングル版実行モジュール
- bin/advsound-p 静的負荷分散版実行モジュール
- bin/advsound-h 動的負荷分散版実行モジュール
- bin/makefem_sound 一体型 adv ファイル作成ツール
- bin/mkbc4snd 吸音境界変換ツール

1.4. プログラムの実行方法

ここでは、ADVENTURE_Sound の静的負荷分散実行モジュールの実行方法を示す。また、並列処理には MPICH における実行方法を示す。

```
% mpirun [option for mpirun] advsound-p [option] [data_dir]
```

ここで、[option for mpirun] は mpirun に対するオプションであり、主なものとして以下のようなものがある。詳細は MPICH のマニュアルを参照。

- -np n
起動する MPI プロセス数 n を指定する。
- -hostfile [machine_file]
並列計算に使用するマシン名のリストファイルを指定する。指定しない場合はシステムで設定されているデフォルトファイルが使用されている。また、[options] は ADVENTURE_Sound に対するオプションである。

2. 並列処理機能及び線形方程式ソルバについて

2.1. 並列解析手法

ADVENTURE_Sound では、階層型領域分割法(HDDM)を用いることで、並列処理を実現している。図 1 に階層型領域分割法の模式図を示す。1 階層目の大きな分割単位を“Part”とよび、2 階層目の細かい分割単位を“Subdomain”と呼ぶ。これらの領域分割は ADVENTURE_Metis によって行うことができる。

ADVENTURE_Sound は並列ライブラリとして MPI2 以降を用いており、起動時には指定に応じて複数のプロセスが起動される。1 ノードあたりに 1 プロセスを割り当てるのが普通であるが、1 ノードに複数のプロセスを割り当てることも可能である。

2.2. 前処理

ADVENTURE_Sound では 2 つの前処理手法を使用している。また、必要に応じて前処理を用いない場合の選択も可能である。また、ここで記載する前処理は階層型領域分割法の 領域間釣合問題に適応するものである。

- 対角スケーリング前処理 前処理を指定しない場合は通常この対角スケーリング前処理が用いられる。

BDD 前処理と比較し、メモリの使用量が少ない。また、非定常解析においては、対角スケーリング前処理のほうが効果的ある。

- BDD 前処理

BDD 前処理は、非常に強力な前処理法である BDD 法を使用している。BDD 法は Neumann-Neumann 前処理の改良あり、各 CG 反復において解析領域よりも少ない自由をもつ“コース問題”を解いている。BDD は対角スケーリングと比べ、大きく反復回数を減少することができ、計算時間の短縮が期待できる。ここで、前処理行列を作成するため、前処理の計算コストがかかり、使用メモリが対角スケーリングと比較し、増加する。しかし、解析時間全体では、高速化が期待できるため、十分なメモリを確保できる計算機環境であれば、BDD を薦める。

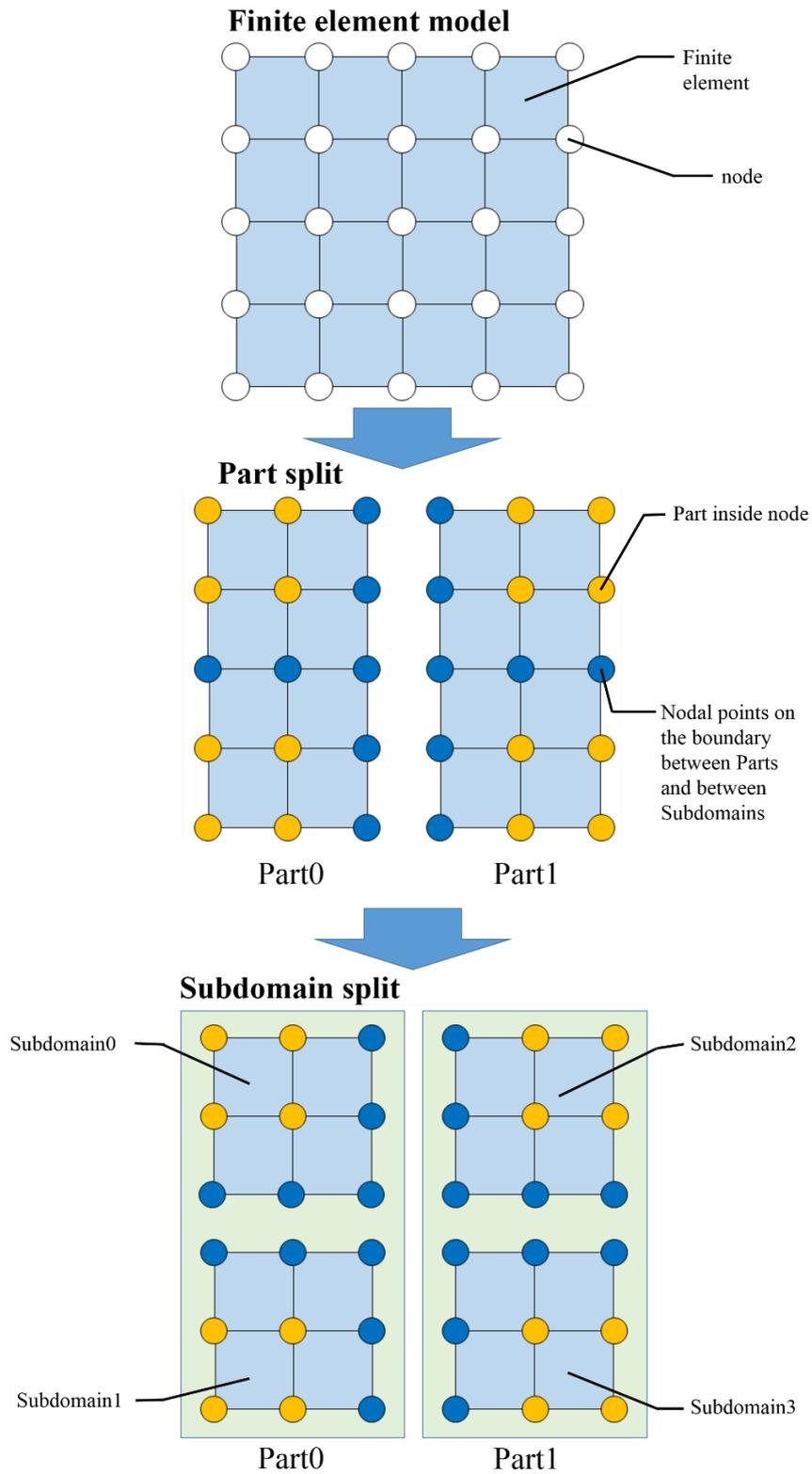


图 1 階層型領域分割法 6

2.3. 領域分割(ADVENTURE_Metis)について

ADVENTURE_Sound の計算性能は、ADVENTURE_Metis による領域分割数に依存している。ADVENTURE_Metis を実行する前に、部分(Part)数、領域(Domain)数をあらかじめ決めておく必要がある。基本的に、部分数は並列処理の手法、使用するノード数や計算機環境に基づいて決定される。領域数は、計算処理に必要とされるメモリに基づいて決定される。細かく分割すれば、メモリの使用量を少なくできるが、ADVENTURE_Metis では、非常に細かい分割を行った場合、要素を一つも含まない領域が作られてしまうことがある。ADVENTURE_Sound では実行時にそのような領域が見つかったら、警告を出して終了するようになっている。また、総要素数に対して非常に荒い分割を行った場合、計算量の増加、メモリ不足になることもあるが、この時も警告を出して終了する。静的負荷分散版 (advsound-p) の場合、1 領域あたりの要素数が 100-300 の時、BDD ソルバ、対角スケールングソルバともに最良の性能を得ることができる。この数は経験値であるため、必ずしもすべてのモデルにおいて、この範囲が最適な要素数になるとは限らない。

ADVENTURE_Metis では、部分数 N_{part} と、一部分あたりの領域数 $N_{subdomain}$ を指定して実行するため、総要素数を $N_{element}$ とした時、一領域あたりの要素数 n は字式によって与えられる。

$$n = \frac{N_{element}}{N_{part} \times N_{subdomain}}$$

$N_{element}$: 総要素数
$N_{subdomain}$: 領域分割数
N_{part}	: 部分数

2.4. 疎行列格納形式

疎行列をメモリ上に保持する場合、値がゼロである成分が多いという点を利用した、特別な形式を用いる必要がある。本モジュールでは、対称な疎行列を扱うため、下三角部分のみを保持する。メモリ使用量と計算時間を考慮しながら非零成分(**nnz**)のみを保持することを目的とした、いくつかの格納形式を以下に示す。

- Compressed Sparse Row (CSR)
- Coordinate Storage (COO)
- Diagonal Coordinate Storage (DCOO)
- Compressed Sparse Column (CSC)
- Modified Compressed Sparse Row (MSR)
- Incremental Compressed Sparse Row (ICSR)
- Variable Block Compressed Sparse Row (VBCSR)
- Diagonal Block Compressed Sparse Row (DBCSP)

3. 解析機能

ADVENTURE_Sound は、定常・非定常音響解析の解析が可能である。

これらの解析における機能を以下に示す。

3.1. 解析の流れ

ADVENTURE システムを用いた一連の解析の流れは以下のようなものである。

(1) メッシュデータの作成(ADVENTURE_TetMesh)

ADVENTURE_TetMesh モジュールを用いて、解析モデルのメッシュを作成する。

(2) 境界条件の付加(ADVENTURE_BCtool)

ADVENTURE_BCtool の msh2pch を用いてメッシュ表面情報の抽出を行い、BcGUI2 を用いて境界条件の設定を行う。GUI による条件設定の詳細は ADVENTURE_BCtool のマニュアル参照。

(3) 一体型解析モデルの作成(makefem_sound)

ADVENTURE_Sound に含まれる makefem_sound を持ちいて、メッシュデータ、物性値、境界条件から ADVENTURE_IO フォーマットの一体型解析モデルを作成する。ただし、吸音境界条件を設定する場合には mkbc4snd も用いる必要がある。詳細は、Appendix C.1 にて述べる。

(4) 領域分割(ADVENTURE_Metis)

ADVENTURE_Metis モジュールを用いて、一体型の解析モデルより階層型に領域分割されたモデルを作成する。なお、実行時にはオプション `-dofn 1` を必ず用いる必要がある。このオプションは内部境界上節点の自由度を 1 に指定するものである。音響解析は 1 節点に 1 自由度を持つためこのオプション設定となる。

(5) 音響解析(ADVENTURE_Sound)

ADVENTURE_Sound モジュールを用いて、分割された解析モデルを入力として、有限要素解析を行う。

(6) 結果の可視化(Paraview)

ADVENTURE_Sound_β では、paraview[3]を用いて可視化する。msh ファイルと dat ファイルから、vtk ファイルを作成し paraview で可視化する。詳細は、Appendix C.4 にて述べる。

3.2. 非定常解析機能

非定常解析においては，線形加速度法を用いて解析が可能である．大まかな処理の流れは図2のようになっている．全体として，2重ループ構造になっており，外側のループは時間積分による反復を，内側のループでは階層型領域分割法におけるCG法の反復を行っている．

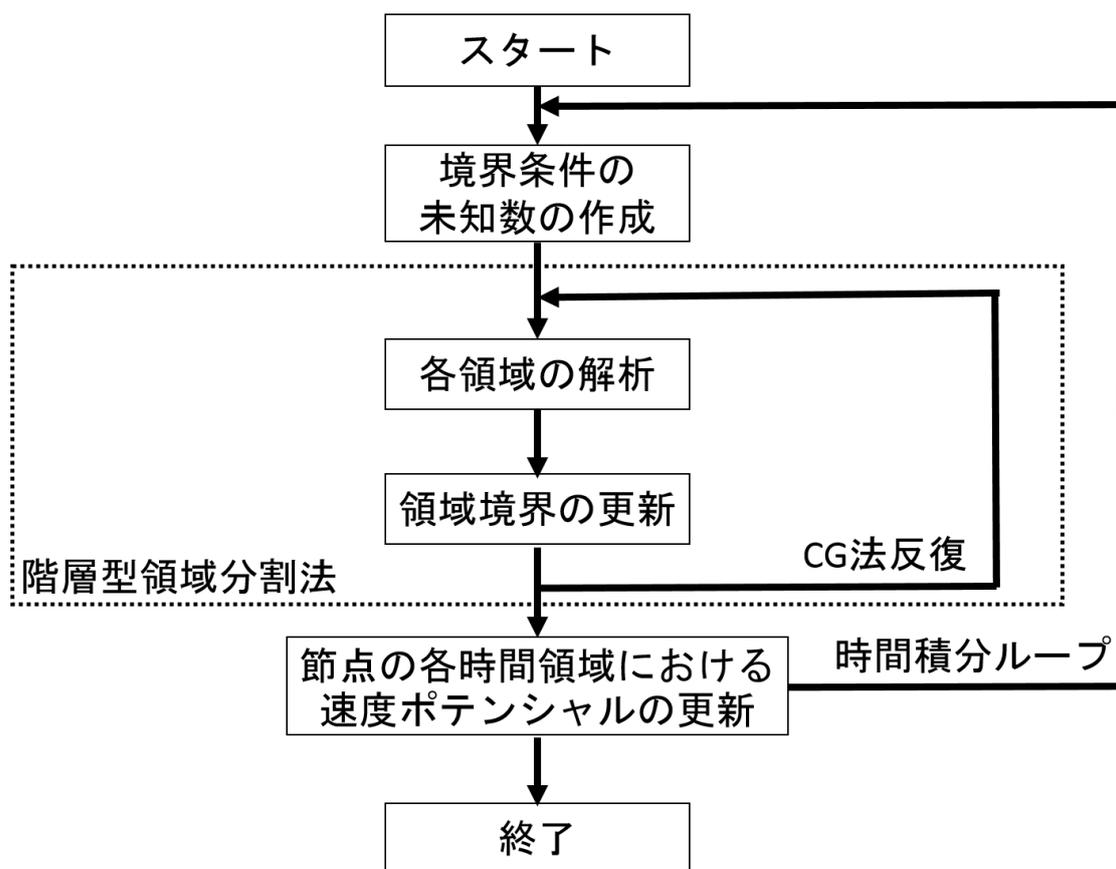


図 2 非定常解析の流れ

3.3. 入出力データについて

ADVENTURE_Sound における入出力ファイルは画面に出力される解析ログ以外は全て ADVENTURE フォーマットをファイルであり、1Part ごとに 1 ファイルとなっている。

入力ファイルは階層型に領域分割された解析モデルファイルである。

これは解析を行う前に、ADVENTURE_Metis によって作成する。

出力ファイルも階層型に領域分割された形式で作成される。節点ごとに速度ポテンシャルが出力される。非定常解析では時間積分ステップごとの出力も可能である。

3.4. 境界条件

付加できる境界条件は以下のようなものである。

- 速度ポテンシャル規定境界条件（節点指定）
- 吸音規定境界条件（節点指定）

また、ADVENTURE_Sound の未知数は速度ポテンシャルである。境界条件に音圧や粒子速度を用いる場合には、以下の式にて音圧や粒子速度を速度ポテンシャルに変換して導入する。

- 音圧 → 速度ポテンシャル

$$vp = -j \times \frac{sp}{\rho \times \omega}$$

- 粒子速度 → 速度ポテンシャル

$$vp = j \times \frac{pv \times c}{\omega}$$

- | | |
|-------------------|-------------------|
| ・ vp : 速度ポテンシャル | ・ ρ : 媒質の密度 |
| ・ sp : 音圧 | ・ ω : 角周波数 |
| ・ pv : 粒子速度 | ・ c : 音速 |
| | ・ j : 虚数単位 |

3.5. 非定常解析における境界条件の設定

非定常解析では速度ポテンシャルの規定境界条件を波の形で、速度ポテンシャル、時間の一階微分の速度ポテンシャル、時間の二階微分の速度ポテンシャルを与える。以下に正弦波として与えた場合の例を示す。

$$\{\Phi\}_t = \{\Phi\}\sin(\omega t)$$

$$\{\dot{\Phi}\}_t = \omega \times \{\Phi\}\cos(\omega t)$$

$$\{\ddot{\Phi}\}_t = -\omega^2 \times \{\Phi\}\sin(\omega t)$$

$\{\Phi\}$: 速度ポテンシャル

$\{\Phi\}_t$: 時間領域の速度ポテンシャル

$\{\dot{\Phi}\}_t$: 一階微分の速度ポテンシャル

$\{\ddot{\Phi}\}_t$: 二階微分の速度ポテンシャル

ω : 各周波数

t : 時間

ADVENTURE_Sound では、非定常解析における速度ポテンシャルの規定境界条件の波形を自由に設定することが可能である。以下にその設定手順を示す。

(1) set_bc.c の指定

ADVENTURE_Sound のホーム上に set_bc.c のシンボリックリンクが置かれている。
Emacs などのエディタを用いて、set_bc.c を開く。

(2) define 以下の書き換えによる波形の設定

以下に ADVENTURE_Sound の境界条件設定部分(set_bc.c)の中身を示す。ここでは例で示した正弦波を境界条件に与えた場合で設定した状態を示す。

```
#include "./set_bc.h"

/*-----*/
/*
境界条件設定部
__None_D          : 速度ポテンシャル
__First_D         : 一階微分の速度ポテンシャル
__Second_D        : 二階微分の速度ポテンシャル

-----

VAL      : 境界条件値  omega      : 角周波数
time     : 時間 (ループ数 × 時間刻み)
*/

#define __None_D      VAL*sin(omega*time)
#define __First_D     omega*VAL*cos(omega*time)
#define __Second_D   -1*VAL*(omega*omega)*sin(omega*time)

/*-----*/

~~~~~
```

ADVENTURE_Soundの境界条件設定部はC99以上のC言語によって開発されている。任意の波形を与えたい場合、上記の境界条件設定部の#define以下を書き換えることで設定が可能である。また、__None_Dは時間領域の速度ポテンシャル、__First_Dは時間領域の一階微分の速度ポテンシャル、__Second_Dは時間の二階微分の速度ポテンシャルを表し、VALは速度ポテンシャル、omegaは各周波数、timeは時間であり、timeに関しては設定した時間刻みに設定した時間ステップの計算時のループ回数をかけた値で求められる。

3.6. 物性値

ADVNTURE_Sound では以下の物性値が使用可能である。

- 媒質
- 音速
- 周波数
- スタート時間（非定常解析時）

非定常解析における，物性値：スタート時間は未知数として与える，波のどの時間帯から解析するかを決める値である。

3.7. 解析結果の出力

解析結果は階層型に領域分割された形式で与えられ，各節点での速度ポテンシャルが出力される。ADVENTURE_IO フォーマットで出力される。

4. コンパイルとインストール

4.1. コンパイル

ADVENTURE_Sound モジュールをコンパイルするには、C コンパイラと MPI のコンパイル環境、ADVENTURE_IO がインストールされている必要がある。ADVENTURE_Sound のモジュールをコンパイルするには、以下の手順に従えばよい。

```
% ./configure
% make
```

ただし、どちらのコマンドも ADVENTURE_Sound モジュールのトップディレクトリで実行する必要がある。各ディレクトリにおいて、make を実行する場合は、hddmsrc をコンパイルする前に、libfem をコンパイルしておく必要がある。

4.2. 実行モジュールのインストール

コンパイルに成功したら、以下によりモジュールのインストールが行われる。

```
% make install
```

インストール先は `configure` で指定した場所である。
以上の作業により、以下のファイルがインストールされる。

- ⇒ advsound-s シングル版実行モジュール
- ⇒ advsound-p 静的負荷分散実行モジュール
- ⇒ advsound-h 動的負荷分散実行モジュール
- ⇒ mkbc4snd 境界条件設定ツール
- ⇒ makefem_sound 一体型解析ファイル adv ファイルの作成

5. 実行方法

ADVENTURE_Sound は並列処理方法にて 2 つのモジュールがある。それぞれ、次のコマンドで実行を行う。

- シングル版の場合

```
% advsound-s [option] [data_dir]
```

- 静的負荷分散による並列版の場合(MPICH)

```
% mpirun [options for mpirun] advsound-p [options] [data_dir]
```

ここで、[options for mpirun]は mpirun コマンドに対するオプションであり、[options]は ADVENTURE_Sound に対するオプションである。このオプション指定により、解析種類の指定や、種々の設定を行う。[data_dir]は必須オプションであり、入出力データのトップディレクトリを指定する。この下のディレクトリおよびファイル名は 5.1. 節に示す。

5.1. 入出力ファイル名

各入出力ファイル名はデフォルト名であり、以下のようになる。[data_dir]は入出力ファイルのトップディレクトリ名である。なお、これらのディレクトリ名やファイル名は実行時のオプションで変更することが可能である。

- HDDM 型の解析モデル入力ファイル：

```
data_dir/model/advhddm_in_P.adv
```

- 解析結果出力ファイル（定常解析時）：

```
data_dir/result/advhddm_out_P.adv
```

- 解析結果出力ファイル（非定常解析時）：

```
data_dir/result/advhddm_out_S_P.adv
```

ここで、 P は部分番号、 S は時間積分ステップ番号を示している。

5.2.1. 線形ソルバの指定

- `-solver hddm`
線形ソルバ HDDM を指定する。前処理は対角スケーリング前処理となる。
- `-solver bdd`
線形ソルバに BDD 前処理を指定する。
- `-precon none`
前処理を用いない場合に指定（基本的に隠しコマンド）

5.2.2. 非定常解析の指定

- `-ns`
非定常解析を行う。更に以下のようなサブオプションが可能であり、`-ns` に続けて指定することが可能である。
- `--lc`
線形加速度法による時間積分を行う。現在、本研究にて効果がわかっているのはこの手法である。
- `--step n`
時間積分の反復回数の上限を n 回に指定する。デフォルトは 10 回。
- `--out-interval n`
時間積分ステップの最終ステップ以外に、ステップ n 回ごとに解析結果ファイルを出力する。デフォルトでは出力しない。
- `--dt n`
時間刻みを n に指定する。

5.2.3. 疎行列格納形式に関するオプション

- -keep-kmat-csr
領域 FEM 計算のための行列を CSR 形式で格納する。デフォルトはこの形式である。
- -keep-kmat-coo 領域 FEM 計算のための行列を COO 形式で格納する。
- -keep-kmat-msr 領域 FEM 計算のための行列を MSR 形式で格納する。
- -keep-kmat-csc 領域 FEM 計算のための行列を CSC 形式で格納する。
- -keep-kmat-icsr 領域 FEM 計算のための行列を ICSR 形式で格納する。
- -keep-kmat-vbcsr 領域 FEM 計算のための行列を VBCSR 形式で格納する。
- -keep-kmat-dcoo 領域 FEM 計算のための行列を DCOO 形式で格納する。
- -keep-kmat-dbcsr
領域 FEM 計算のための行列を DBCSR 形式で格納する。このオプションは ADVENTURE_Solid のみ利用できる。
- -keep-kmat-dbcsr 領域 FEM 計算のための行列を SKY 形式で格納する。

5.2.4. その他オプション

- `-memlimit n`

各プロセスが使用するメモリの上限を `n[MByte]`とし、これを超えた場合、その時点で実行を停止する。デフォルトは `256[MByte]`である。また“0”を指定した場合はメモリ上限の設定を解除する。

Appendix

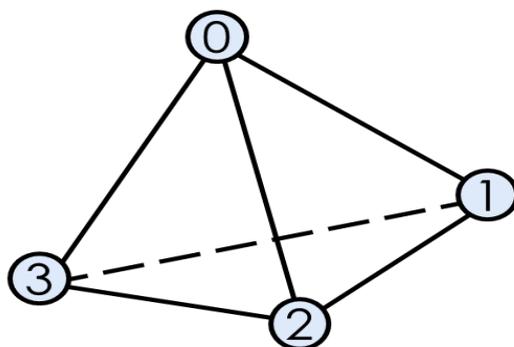
- ・ A 要素について
- ・ B 境界条件について
- ・ C ツール類について
- ・ D 解析例

A 要素について

ADVENTURE_SOUND は、四面体一次要素と四面体二次要素に対応している。ただし、境界条件として吸音条件を与える場合、吸音面ごとの剛性行列を作成する。この場合は、四面体一次要素の場合は、三角形一次要素を、四面体二次要素の場合は、三角形二次要素を用い、面積分を行う。以下に、各要素について示す。

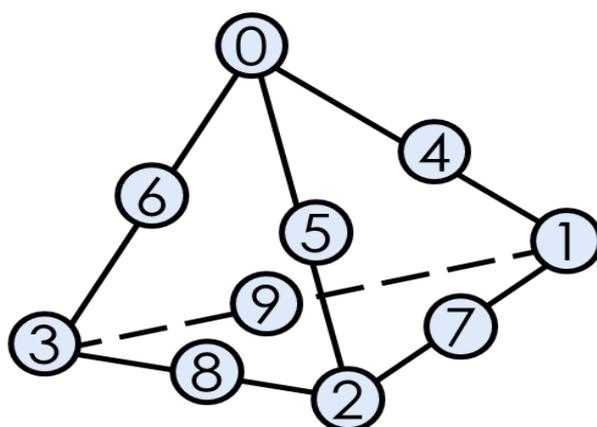
・四面体一次要素

節点数は 4 であり、要素コネクティビティの各節点の節点番号の並び順は図のようになっている。



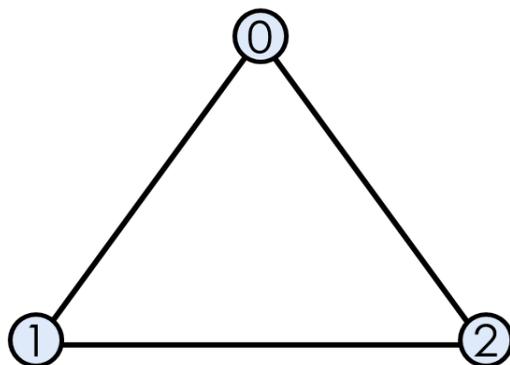
・四面体二次要素

節点数は 10 であり、要素コネクティビティの各節点の節点番号の並び順は図のようになっている。



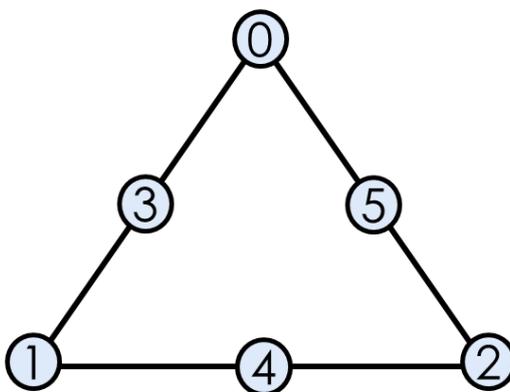
・ 三角形一次要素

節点数は 3 であり，要素コネクティビティの各節点の節点番号の並び順は図のようになっている。



・ 三角形二次要素

節点数は 6 であり，要素コネクティビティの各節点の節点番号の並び順は図のようになっている。



B 境界条件について

ADVENTURE_Sound で利用可能な境界条件のフォーマットの例を以下に示す.

なお, これらは, ADVENTURE_IO フォーマットを表示した結果であり, 実際には, 一体型解析ファイル作成ツール `makefem_sound`, 境界条件貼り付けツール `mk4snd` により作成を行う.

B.1 速度ポテンシャル規定条件

```
[Properties]
1: content_type=FEGenericAttribute
2:num_items=310
3: fega_type=NodeVariable
4: label=VelocityPotential
5: format=i4f8f8
6: index_byte=4

[Data]
0: 0 0 0.000000e+00 1.528524e-03
1: 3 0 0.000000e+00 1.528524e-03
2: 4 0 0.000000e+00 1.528524e-03
3: 7 0 0.000000e+00 1.528524e-03
4: 182 0 0.000000e+00 1.528524e-03
5: 183 0 0.000000e+00 1.528524e-03
...
...
```

[Data]フィールドは左から順に, 節点番号, 方向成分, 速度ポテンシャルの実部, 速度ポテンシャルの虚部である. ただし, 音響解析は 1 節点に 1 自由度であるため, 方向成分に関しては全境界条件で 0 と指定する.

B.2 吸音規定境界条件

```
[Properties]
1: content_type=FEGenericAttribute
2: feqa_type=NodeVariable
3: label=SoundAbsording
4: format=i4f8f8f8
5: index_byte=4
6: num_items=310
[Data]
0: 108701 0 1.300000e+00 4.459000e+02 0.000000e+00
1: 108714 3 1.300000e+00 4.459000e+02 0.000000e+00
2: 109083 2 1.300000e+00 4.459000e+02 0.000000e+00
3: 108725 0 1.300000e+00 4.459000e+02 0.000000e+00
4: 109086 2 1.300000e+00 4.459000e+02 0.000000e+00
5: 109085 1 1.300000e+00 4.459000e+02 0.000000e+00
...
...
```

[Data]フィールドは左から順に、要素番号、面番号、媒質、音響インピーダンスの実部、音響インピーダンスの実部である。また、面番号は各要素を形成される面にそれぞれ与えられるものであり、対する頂点の節点番号と等しくなるような番号付けを行っている。

B.3 物性値ファイル 1

音響解析用の物性値ファイルフォーマット例を示す。makefem_sound への入力ファイルとなる、

(1) 単一材料モデル例

```
SpeedOfSound 3.43000e+02
```

記述するのは音速である。

(2) 複数材料モデル例

```
#materialInfo  
materialN 2  
propertyN 1  
SpeedOfSound 3.43000e+02  
SpeedOfSound 1.50000e+03  
#volumeInfo  
volumeN 2  
0  
1
```

B.4 物性値ファイル 2

ADVENTURE_Sound では 6.3.節で示した adv ファイル用の物性値ファイルとは別に、もう一つ物性値ファイルを用意する必要がある。ファイル名は“sound_mate.dat”で固定する必要がある。以下に記述例を示す。

(1) 定常解析

```
1
FREQ
500.0
```

記述を行うのは**周波数**であり、定常解析、非定常解析共に必須項目である。また、定常解析では一行目には“1”を記述する。また、周波数を記述する場合は数値を書き込む前に“FREQ”と記載した次の行に記載する。

(1) 非定常解析

```
2
FREQ
500.0 TIME
0.000001
```

周波数は必須項目である。非定常解析では一行目に“1”もしくは“2”を記載する。TIME に関しては、未知数として与える波形のいつの時間帯から解析を始めるかを指定する。指定が必要のない場合は、一行目を“2”とし TIME 記載の次の行に 0.0 を記載するか、記載をせず一行目を“1”と入力する。

C ツール類について

ADVENTURE_Sound には、本体モジュールの他に以下のツールが含まれている。

C.1 一体型解析モデルファイル作成ツール **makefem_sound**

makefem_sound は、音響解析の一体型解析モデルファイル作成ツールである。

このコマンドを用いることで、ADVENTURE_BCtool において作成したメッシュ表面情報ファイルと境界条件設定ファイルを用いて、音響解析用一体型解析モデルファイルを作成する。ただし、吸音境界条件に関しては mkbc4snd を用いる必要がある。

以下に実行方法を示す。

```
% makefem_sound [mshFILE] [fgrFILE] [cndFILE] [matFILE] [advmeshFILE]
```

[mshFILE]	: メッシュデータファイル (入力)
[fgrFILE]	: メッシュ表面データファイル (入力)
[cndFILE]	: 解析条件ファイル (入力)
[matFILE]	: 物性値ファイル 1 (入力)
[advmeshFILE]	: 一体型解析モデルファイル (出力)

ただし、[fgrFILE]は[meshFILE]から ADVENTURE_BCtool の msh2pch により作成される。
[cndFILE]は、面指定の場合 ADVENTURE_BCtool の bcGUI により作成されるものである。
節点指定の場合は、フォーマットを参考に [cndFILE]を作成する必要がある。以下に、解析条件ファイルのフォーマットを示す。

▪ 面単位 の付与

gravity 0.0 0.0 0.0	← とりあえずの値
boundary 1	← 境界条件データ数
dispOnFaceGroup 0 0 0 0.00000000 0.001528524	← 面グループ 0 の速度ポテンシャル

▪ 節点 単位 の付与

gravity 0.0 0.0 0.0	← とりあえずの値
boundary 2	← 境界条件データ数
node 0 0 0 0.00000000 0.001528524	← 節点(0, 0, 0)の速度ポテンシャル
node 1.0 1.5 2.0 0.00000000 0.001528524	← 節点(1.0, 1.5, 2.0)の速度ポテンシャル

C.2 吸音境界条件設定ツール `mkbc4snd`

`mkbc4snd` は、`ADVENTURE_BCtool` の `msk2pch` により得られるメッシュ表面データファイル（拡張子 `fgr`）、及び解析条件ファイル（拡張子 `cnd`）から、音響解析向けの吸音境界条件をの `ADVENTURE_IO` フォーマットファイルを作成するツールである。

以下に吸音境界条件ファイルを示す。

<code>boundary 1</code>
<code>dispOnFaceGroup 5 1.3000000 445.9 00000 0.00000</code>

ただし、左から順に面番号、媒質、音響インピーダンスの実部、音響インピーダンスの虚部である。また、拡張子 `cnd` である。
この境界条件ファイルを作成した後に次のコマンドを実行する。

```
% mkbc4snd [fgrFILE] [cndFILE] [advbcFILE]

[fgrFILE]      : メッシュ表面データファイル（入力）
[cndFILE]      : 吸音境界条件ファイル（入力）
[advbcFILE]    : 一体型解析モデルファイル（出力）
```

C.3 adv ファイルの合成

makefem_sound と mkbc4snd によって作成された[*advmeshFile*]と[*advbcFILE*]の 2 つの一体型ファイルを合成し、 ADVENTURE_Sound で扱う形へ変換する。

使用するツールは “ advcat ” であり、 ADVENTURE_Solid に含まれている。以下に使用方法を示す。

```
% advcat -o [ advFILE ] [ advmeshFILE ] [ advbcFILE ]
```

[*advmeshFILE*] : 一体型解析モデルファイル (出力)

[*advmeshFILE*] : 一体型解析モデルファイル (入力)

[*advmeshFILE*] : 一体型解析モデルファイル (入力)

C.4 可視化 dat2vtk

ADVENTURE_SOUND では、 paraview[3] を用いて可視化を行うツールを AdvSound- beta /tool/dat2vtk に用意しているが make ファイルは用意していないため、 make.sh を実行していただきたい。一行につき値は一つの読み取りのため、複素数の場合は実部の可視化となる。そのため、虚部の可視化は別ファイルで出力する必要がある。

解析に用いたメッシュファイルと結果ファイル(dat ファイル)を用いて、 vtk フォーマットファイルを作成する。実行方法は以下の通りである。

```
% dat2vtk [メッシュファイル名(.msh)] [結果ファイル名 (.dat)] [-m1 or -m2]
```

一次節点の場合は -m1、二次節点の場合は-m2 を指定する。これにより [結果ファイル.vtk] が出力される。そのファイルを paraview に読み込ませることにより可視化が可能である。詳しくは paraview[3]を参照していただきたい。

C.5 非定常の可視化 `nonsteady_dat2vtk`

非定常解析の可視化には `dat2vtk` ではなく、`nonsteadt_dat2vtk` を使用する必要がある。このコマンドを用いることで、複数存在する解析結果ファイルを一度に `vtk` ファイルに変換できる。さらに、プログラム内部では `vtk` ファイルだけでなく、`hddmrg` を使用することで `dat` ファイルへの変換も行っている。実行方法は以下の通りである。

```
% nonsteady_dat2vtk [メッシュファイル名(.msh)] [結果ファイル名(.dat)]  
                   [総 step 数] [出力ファイルの間隔] [-m1 or -m2]
```

[出力ファイルの間隔] は、`advsound-p` 実行時の `-out-interval` の値を指定する。これにより、変換された全ての `dat` ファイル・`vtk` ファイルが [メッシュファイル名]_dat と [メッシュファイル名]_vtk の内部に作成され、`vtk` ディレクトリごと `paraview` に読み込ませることで解析結果を可視化することができる。

C.6 音圧変換（定常・非定常）`vp2sp`, `nonsteady_vp2sp`

速度ポテンシャルで記載されている解析結果ファイルを音圧に変換し可視化を行うには、定常解析では `vp2sp` と `dat2vtk` を、非定常解析では `nonsteady_vp2sp` を行う必要がある。

以上のコマンドを用いることで、`paraview` で可視化可能な `vtk` フォーマットファイルを作成する。実行方法は以下の通りである。

[定常解析]

```
% vp2sp [境界条件値ファイル名(bc.cnd)] [物性値ファイル名(sound_mate.dat)]  
        [結果ファイル名(.dat)] [好きなファイル名(.dat)]
```

この後、C.4 を参考に、`dat2vtk` のツールを使用して `.vtk` ファイルを作成することで可視化が出来る。

[非定常解析]

```
% nonsteady_vp2sp [メッシュファイル名(.msh)] [境界条件値ファイル名(bc.cnd)]  
                [物性値ファイル名(sound_mate.dat)] [結果ファイル名(.dat)]  
                [好きなファイル名(.dat)] [総ステップ数] [出力ファイルの間隔]  
                [-m1 or -m2]
```

dat2vtk や nonsteady_dat2vtk のツールを使用するときと同様に
一次節点の場合は -m1, 二次節点の場合は -m2 を指定する. [出力ファイルの間隔] は,
advsound-p 実行時の -out-interval の値を指定する. これにより, 変換された全ての
dat ファイル・vtk ファイルが Changed_[メッシュファイル名]_dat と,
Changed_[メッシュファイル名]_vtk の内部に作成され, vtk ディレクトリごと paraview
に読み込ませることで解析結果を可視化することができる.

D 解析例

D.1 定常解析

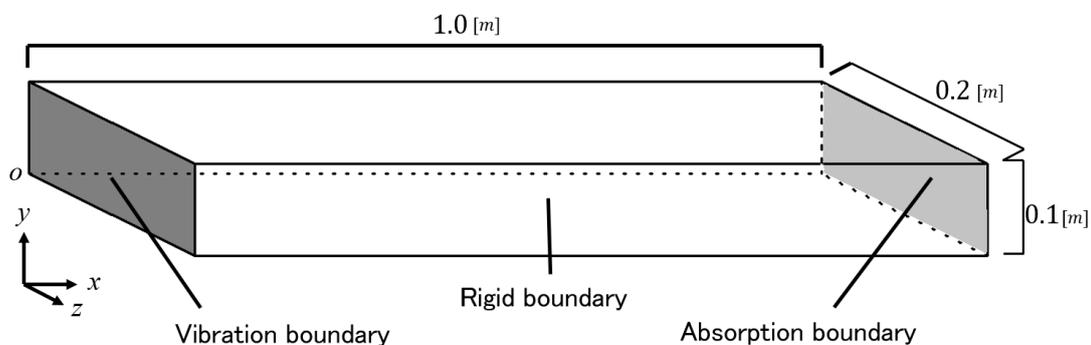


図 3 解析モデル

図 3 に示すような、0.1[m]、0.2[m]、1.0[m]の立体モデル[4]における、定常音響解析を考える。境界条件は以下の通りである。左端に振動境界条件(音源)、右端に吸音境界条件を与えた。音源と吸音端を除いた面はすべて全剛性(全反射)とする。境界条件は以下の通りである。また、本解析例では四面体一次要素でメッシュ分割した場合の解析例を示す。

振動境界条件 (粒子速度) :	0.014[m/s]
吸音境界 :	445.9[kg/m ² s]
解析領域の媒質 :	1.3[kg/m ³]
音速 :	343.0[m/s]
解析周波数 :	500.0[Hz]

D.2 入力データの作成

ADVENTURE_Sound_βによる解析を行うために入力ファイル作成例を示す.

- (1) メッシュデータ (msh 形式) の作成
- (2) msh2pch(ADVENTURE_BCtool)により
メッシュ表面情報データ (fgr, pch, pcg, trn 形式) の作成
- (3) BcGUI2(ADVENTURE_BCtool)により
速度ポテンシャル規定境界設定ファイル(cnd 形式) と吸音境界設定ファイル(cnd 形式)
の作成
- (4) 物性値ファイル 1 と物性値ファイル 2 の作成
- (5) makefem_sound と mkbc4snd, advcat により一体型解析ファイル (adv 形式) の作成
- (6) adventure_metis(ADVENTURE_Metis)により領域型分割解析モデルファイル (adv 形式)
の作成

以下に解析例に用いたモデルを作成したコマンドを示す.

■ メッシュデータの作成

Step.1: ADVENTURE_CAD を用いて AHLV100.gm3d を作成

AHLV100.gm3d

```
box 0.0 0.0 0.0 1.0 0.1 0.2
```

Step.2: エディタにより節点密度ファイル AHLV100.ptn を作成

作成しているメッシュデータの要素の最大辺長を記入する.

このとき, 要素の最大辺長の決定には, 定常解析では周波数から導くことのできる波長の長さの 1/40 (一次要素) 以下, もしくは, 1/20 (二次要素) 以下, 非定常解析では,

1/16 (一次要素) 以下, もしくは, 1/10 (二次要素) 以下であると ADVENTURE_Sound での精度が保証される.

AHLV100.ptn

BaseDistance

0.012

Step.3: ADVENTURE_CAD により表面 3 角形パッチファイルを作成

入力ファイル

AHLV100.gm3d (モデル作成ファイル)

実行コマンド

% advcad AHLV100.gm3d AHLV100.pch [0.012] ※[] は入力しない.

[] の数字は、節点密度ファイルと同じ値を用いる.

出力ファイル

AHLV100.pch (表面パッチデータファイル)

Step.4: ADVENTURE_TetMesh により 4 面体メッシュを作成

Step.4-1: 表面メッシュ作成プログラムにより表面メッシュデータ中間ファイルと
節点密度制御ファイルを作成

入力ファイル

AHLV100.pch (表面パッチデータファイル)

AHLV100.ptn (節点密度ファイル)

実行コマンド

% advtmesh9p AHLV100

出力ファイル

AHLV100c.pcc (表面メッシュデータ中間ファイル)

AHLV100c.ptn (節点密度制御ファイル)

Step.4-2: 4 面体メッシュ作成プログラムにより, 4 面体 **1 次要素**メッシュデータファイルを作成

入力ファイル

AHLV100c.pcc (表面メッシュデータ中間ファイル)

AHLV100c.ptn (節点密度制御ファイル)

実行コマンド

```
% advtmesh9m AHLV100c -p
```

出力ファイル

AHLV100c.msh (4 面体 **1 次要素**メッシュデータファイル)

Step.4-3: 2 次節点追加プログラムにより 4 面体 **2 次要素**メッシュデータファイルを作成

入力ファイル

AHLV100c.msh

実行コマンド

```
% advtmesh9s AHLV100c
```

出力ファイル

AHLV100cs.msh (4 面体 **2 次要素**メッシュデータファイル)

■ 境界条件の設定

Step5: ADVENTURE_BCtool により境界条件設定ファイルを作成

Step5-1: メッシュの表面抽出によりメッシュの表面を抽出及びグループ化

入力ファイル

AHLV100c.msh (四面体 1 次要素メッシュデータファイル)

実行コマンド

[PatchSize] には、今回は 8 を入力する。 [PatchSize] の決定に関して、詳しくは ADVENTURE_BCtool のマニュアルをご覧ください。

```
% msh2pch AHLV100c.msh [PatchSize] (四面体 1 次要素) ※[ ] は入力しない
```

```
% msh2pcm AHLV100c.msh (四面体 1 次要素)
```

出力ファイル

AHLV100c_[PatchSize].fgr (四面体 1 次要素)

AHLV100c_[PatchSize].pch (四面体 1 次要素)

AHLV100c_[PatchSize].pcg (四面体 1 次要素)

AHLV100c_[PatchSize].trn (四面体 1 次要素)

AHLV100c_[PatchSize].pcm (四面体 1 次要素)

Step5-2: GUI により境界条件の設定

入力ファイル

AHLV100c_[PatchSize].pch (表面メッシュ抽出データファイル)

AHLV100c_[PatchSize].pcg (表面パッチグループデータファイル)

実行コマンド

```
% BcGUI2 AHLV100c_[PatchSize].pch AHLV100c_[PatchSize].pcg
```

出力ファイル

AHLV100.cnd (速度ポテンシャル規定条件設定ファイル)

AHLV100bc.cnd (吸音境界設定ファイル)

・ここでの出力ファイル名はメッシュファイル名と揃える必要がある。

また、境界条件の設定ファイルは以下に示すようなものである。また、境界条件の設定ファイルの中身を以下に示す。

AHLV100.cnd

gravity 0.0 0.0 0.0	← 取り敢えずの値
boundary 1	← 振動境界条件の数
dispOnFaceGroup 0 0 0 0.00000000 0.001528524	← 面グループ 0 の速度ポテンシャルの実部, 虚部

AHLV100bc.cnd

boundary 1	← 吸音境界条件の数
dispOnFaceGroup 5 1.3000000 445.9 00000 0.00000	← 面グループ 5 の媒質の密度(1.300...) 音響インピーダンスの実部, 虚部

■ 物性値ファイルの作成

Step6: エディタにより、物性値ファイルを作成する

Step6-1: エディタにより、物性値ファイル 1 (dat ファイル) を作成する

AHLV100.dat

SpeedOfSound 3.43000e+02	←音速
--------------------------	-----

Step6-2: エディタにより、物性値ファイル 2 “sound_mate.dat” を作成する

sound_mate.dat

1
FREQ
500.0 ← 周波数

■ 一体型モデルファイルの作成

Step7: ADVENTURE_Sound により一体型モデルファイルを作成する

Step7-1: makefem_sound を用いて一体型モデルファイルを作成

入力ファイル

AHLV100c.msh	(メッシュデータファイル)
AHLV100c_[PatchSize].fgr	(メッシュ表面データファイル)
AHLV100.cnd	(解析条件ファイル)
AHLV100.dat	(物性値ファイル 1)

実行コマンド

```
% makefem_sound AHLV100c.msh AHLV100c_[PatchSize].fgr  
AHLV100.cnd AHLV100.dat AHLV100mesh.adv
```

出力ファイル

AHLV100mesh.adv

Step7-2: mkbc4snd を用いて一体型境界条件ファイルを作成

入力ファイル

AHLV100c_[PatchSize].fgr

AHLV100.cnd

実行コマンド

```
% mkbc4snd AHLV100c_[PatchSize].fgr AHLV100bc.cnd AHLV100bc.adv
```

出力ファイル

AHLV100bc.adv

Step7-3: advcat (ADVENTURE_Solid) によって一体型解析ファイルを作成

入力ファイル

AHLV100mesh.adv

AHLV100bc.adv

実行コマンド

```
% advcat -o AHLV100.adv AHLV100mesh.adv AHLV100bc.adv
```

出力ファイル

AHLV100.adv

■ 領域分割型の解析モデルファイルの作成

Step8: ADVENTURE_Metis により領域分割解析モデルファイルを作成。ここでは、並列計算用として CPU を 2 台用いる。解析モデルの要素数が 140,604 であるため、

$$N_{element} = 140,604$$

また、静的負荷分散版における 1 領域あたりの要素数が 80 – 200 であるので、

$$n = 200$$

$$N_{part} = \text{並列数}$$

とすると、

$$\begin{aligned} N_{subdomain} &= \frac{N_{element}}{n \times N_{part}} \\ &= \frac{140,604}{2 \times 200} \\ &= 351.51 \\ &\cong 351 \end{aligned}$$

よって、ADVENTURE_Metis において次のコマンドを実行する。

入力ファイル

AHLV100.adv

実行コマンド

```
% mpirun -np 2 adventure_metis -difn 1 AHLV100.adv AHLV100 351
```

出力ファイル

AHLV100/model/advhddm_in_0.adv

AHLV100/model/advhddm_in_1.adv

■ 出力データ

Step9: ADVENTURE_Sound による解析を行う。解析するには AHLV100 ディレクトリと同様の場所に物性値ファイル 2 “sound_mate.dat” 置かれている必要がある。

入力ファイル

AHLV100/model/advhddm_in_0.adv

AHLV100/model/advhddm_in_1.adv

実行コマンド

```
% mpirun -np 2 advsound-p AHLV100
```

出力ファイル

AHLV100/result/advhddm_out_0.adv

AHLV100/result/advhddm_out_1.adv

Step10-1: 解析結果の統合

解析結果は領域ごとに出すようになっている。以下に結果の統合方法を示す。統合するために用いるのは、ADVENTURE_Solid のツールに含まれる “hddmmrg” である。

入力ファイル

AHLV100/result/advhddm_out_0.adv

AHLV100/result/advhddm_out_1.adv

実行コマンド

```
% hddmmrg VelocityPotential_adv AHLV100
```

出力ファイル

VelocityPotential_adv.dat

出力ファイルの結果を以下に示す.

VelocityPotential_adv.dat

```
label= VelocityPotential_adv
num_items=26927 ←節点の総数

0:      0.00000000e+00    1.52852400e-03 ←節点番号, 速度ポテンシャルの実部, 虚部
1:      4.06196815e-04   -1.47270790e-03
2:      4.06014433e-04   -1.47204968e-03
3:      0.00000000e+00    1.52852400e-03
4:      0.00000000e+00    1.52852400e-03
5:      4.06152074e-04   -1.47246711e-03
...
...
```

Step10-2: 解析結果ファイル(.dat)を vtk ファイル に変換

解析結果ファイル(.dat)は, このままでは paraview では可視化することができないため以下の方法により解析結果ファイルを可視化可能な vtk ファイルの作成を行う.

入力ファイル

AHLV100c.msh

VelocityPotential_adv.dat

実行コマンド

```
% dat2vtk AHLV100 VelocityPotential_adv -m1( or -m2 )※四面体二次要素の場合は --m2
```

出力ファイル

VelocityPotential_adv.vtk

結果ファイルは節点番号, 速度ポテンシャルの実部, 速度ポテンシャルの虚部の順で記載される. 得られた速度ポテンシャルを可視化したもの[5]を以下に示す.

Step10-3: 速度ポテンシャルの解析結果ファイル (.dat) を音圧 (.dat) に変換する

解析結果ファイルは、速度ポテンシャルで表記されている。そのため、音圧での結果ファイルが必要な場合、以下の変換ツールを用いて変換する。

入力ファイル

AHLV100bc.cnd
Sound_mate.dat
VelocityPotential_adv.dat

実行コマンド

```
% vp2sp AHLV100 sound_mate VelocityPotential_adv SoundPressure_adv
```

出力ファイル

SoundPressure_adv.dat

Step10-4: 音圧変換した解析結果ファイル(.dat)を vtk ファイル に変換

先ほどと同様に、解析結果ファイル(.dat)は、このままでは paraview では可視化することができないため以下の方法により解析結果ファイルを可視化可能な vtk ファイルの作成を行う。

入力ファイル

AHLV100c.msh
VelocityPotential_adv.dat

実行コマンド

```
% dat2vtk AHLV100 SoundPressure_adv -m1( or -m2) ※四面体二次要素の場合は --m2
```

出力ファイル

SoundPressure_adv.vtk

結果ファイルは節点番号，音圧の実部，音圧の虚部の順で記載される．得られた音圧[Pa]を可視化したもの[5]を以下に示す．

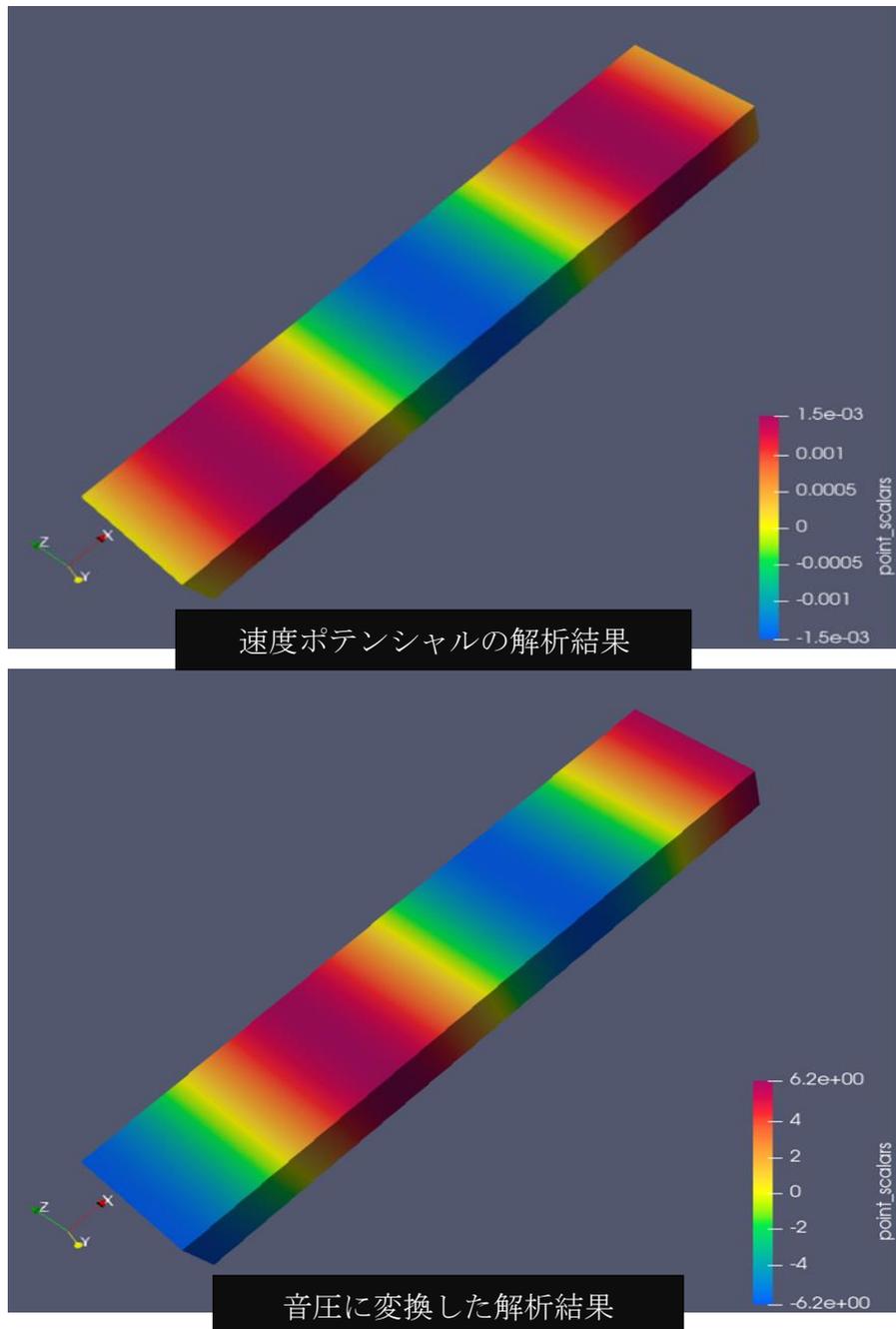


図 4 定常解析可視化結果

D.2 非定常解析

解析例を示すにあたり，図 3 のモデルを使用する．ただし，時間刻みを 0.000001[s]とし，時間ステップを 2915 回，未知数に与えた波形の開始時間を 0.001[s]する．

■ メッシュデータの作成

定常解析と同様である

■ 境界条件の設定

基本的に定常解析時と同様である．ただし，Step6-2: 物性値ファイル 2 の作成のみ別である．

以下に作成例を示す．

Step6-2-2: エディタにより，物性値ファイル 2 “sound_mate.dat” を作成する

sound_mate.dat

```
2
FREQ
500.0          ←周波数
TIME
0.001          ←解析開始時間
```

■ 領域分割型の解析モデルファイルの作成

定常解析と同様である．

■ 出力データ

Step9: ADVENTURE_Sound による解析を行う。解析するには AHLV100 ディレクトリと同様の場所に物性値ファイル 2 “sound_mate.dat” 置かれている必要がある。

入力ファイル

```
AHLV 100 /model /advhddm_in_0.adv  
AHLV100/model/advhddm_in_1.adv
```

実行コマンド

```
% mpirun -np 2 advsound-p -memlimit 0 -ns --lc --out-interval 50 --dt 0.000001  
--step 4500 AHLV100
```

出力ファイル

```
AHLV100/result/advhddm_out_49_0.adv  
AHLV100/result/advhddm_out_49_1.adv  
↓  
AHLV100/result/advhddm_out_4499_0.adv  
AHLV100/result/advhddm_out_4499_1.adv
```

このとき、時間刻みはクーラン条件をもとに決定する。

[クーラン条件]

$$\Delta t^{crit} = \frac{l_{min}}{c}$$

$$\Delta t \leq \Delta t^{crit}$$

- Δt : 時間刻み幅
- Δt^{crit} : クリティカル時間刻み幅
- l_{min} : 要素の最小辺長
- c : 音速

step10-1: 非定常解析結果の統合・vtk ファイルへの変換

ADVENTURE_Sound 付属ツール nonsteady_dat2vtk を用いて ParaView で読み込む vtk ファイルを作成する。統合した dat ファイルは AHLV100_dat、vtk ファイルは AHLV100_vtk の中にある。

入力ファイル

AHLV100/result/advhddm_out_49_0.adv → AHLV100/result/advhddm_out_4499_0.adv
AHLV100/result/advhddm_out_49_1.adv → AHLV100/result/advhddm_out_4499_1.adv

実行コマンド

```
% nonsteady_dat2vtk AHLV100 VelocityPotential_adv 4500 50 -m1 ( or -m2 )
```

※四面体二次要素の場合は -m2

出力ファイル

AHLV100_dat/VelocityPotential_adv_49.dat → VelocityPotential_adv_4499.dat
AHLV100_vtk/VelocityPotential_adv_49.vtk → VelocityPotential_adv_4499.vtk

step10-2: 非定常解析結果の音圧変換と統合・vtk ファイルへの変換

ADVENTURE_Sound 付属ツール nonsteady_vp2sp を用いて非定常解析結果である速度ポテンシャル (.dat) を音圧 (.dat) に変換し, ParaView で読み込む vtk ファイルを作成する。統合した dat ファイルは Changed_AHLV100_dat, vtk ファイルは Changed_AHLV100_vtk の中にある。

入力ファイル

AHLV100_dat/VelocityPotential_adv_49.dat → VelocityPotential_adv_4499.dat

実行コマンド

```
% nonsteady_vp2sp AHLV100 AHLV100 sound_mate VelocityPotential_adv  
SoundPressure_adv 4500 50 -m1 ( or -m2 )
```

※四面体二次要素の場合は -m2

出力ファイル

Changed_AHLV100_dat/SoundPressure_adv_49.dat → SoundPressure_adv_4499.dat
Changed_AHLV100_vtk/SoundPressure_adv_49.vtk → SoundPressure_adv_4499.vtk

以下に、速度ポテンシャルの非定常可視化結果と、速度ポテンシャルの虚部を音圧に変換した非定常可視化結果を示す

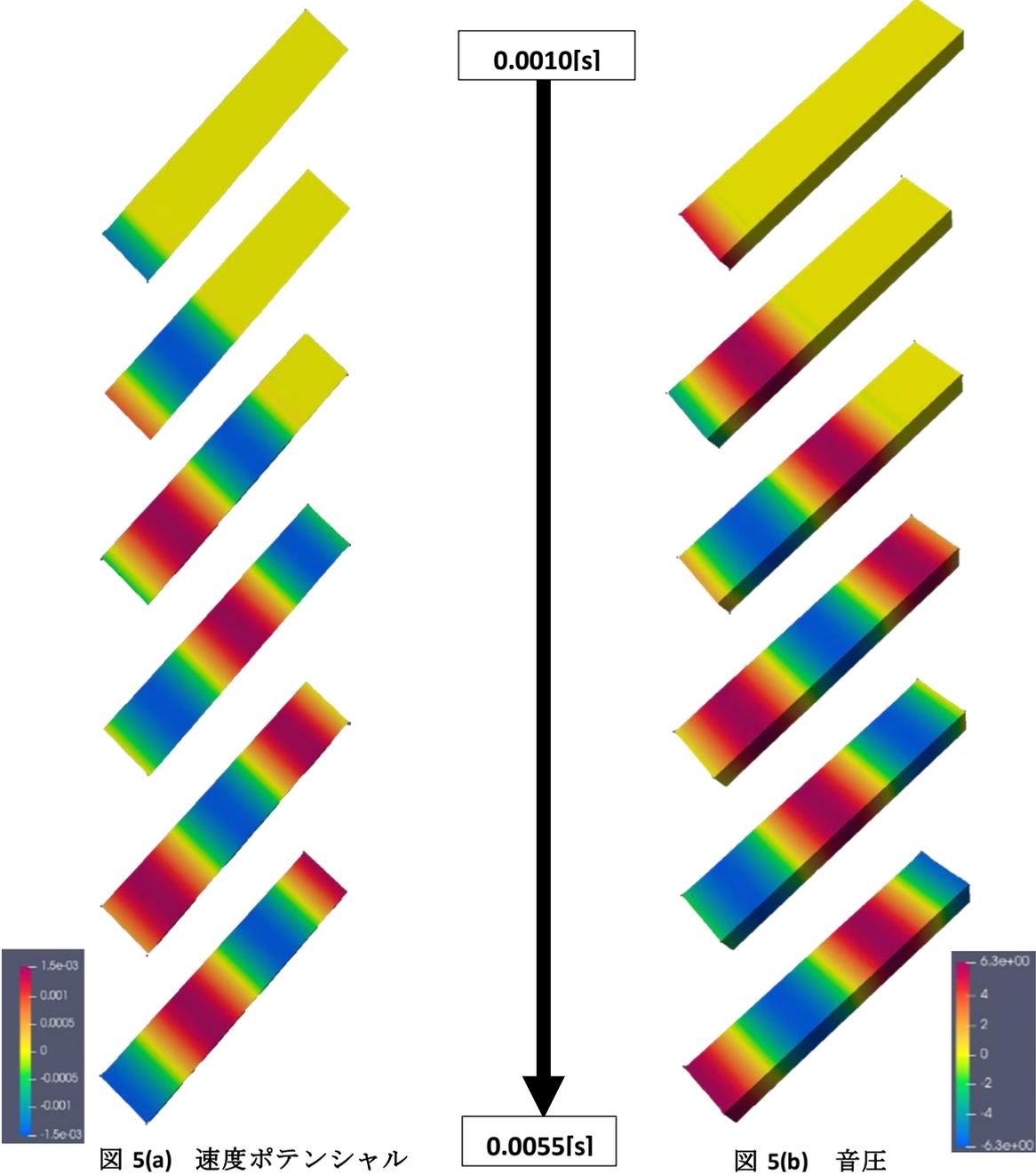


図 5 非定常可視化結果[6]

参考文献

- [1] ADVENTURE Project: <http://adventure.sys.t.u-tokyo.ac.jp>
- [2] Jan Mandel: Balancing Domain Decomposition, Communications on Numerical Methods in Engineering, 9(1993), 233-241
- [3] ParaView: <https://www.paraview.org>
- [4] Josselin DELMAS, “AHLV100 - Guide wave at anechoic exit , ” Code_Aster, Version 11, https://www.codeaster.org/V2/doc/v13/en/man_v/v8/v8.22.100.pdf , 2018
- [5] 山本広太, 工藤彰洋, 武居周, “並列有限要素法に基づく大規模音響解析,” 日本シミュレーション学会論文誌, Vol. 9, No2, pp. 27-37, 2017.
- [6] 村上 裕哉, 山本 広太, 工藤 彰洋, 武居 周, “時間領域並列有限要素法に基づく大規模音響解析,” 日本シミュレーション学会論文誌, Vol. 10, No4, pp. 89-98, 2018.