

音響解析ソフトウェア Advance/FrontNoise FOCUSスパコンを使ったハンズオンセミナー

2016年5月19日(木)開催

プログラム

9:30~10:00	(受付)
10:00~10:15 (15分)	音響解析ソフトウェア Advance/FrontNoise 機能紹介 ・機能概要の説明 ・Advance/REVOCAP インストール ・FOCUS への接続
10:15~11:30 (75分)	音響解析ソフトウェア Advance/FrontNoise 演習 I (消音器の性能解析) ・メッシュ生成 ・解析条件作成 ・計算実行 ・計算結果表示
11:30~12:30 (60分)	(昼食)
12:30~14:00 (90分)	音響解析ソフトウェア Advance/FrontNoise 演習 II (防音壁表面の吸音) ・メッシュ生成 ・解析条件作成 ・計算実行 ・計算結果表示
14:00~14:15 (15分)	(休憩)
14:15~15:30 (75分)	音響解析ソフトウェア Advance/FrontNoise 演習 III ・メッシュの細分化 ・並列性能の測定
15:30~16:45 (75分)	音響解析ソフトウェア Advance/FrontNoise 演習 IV ・可視化とその他の機能
16:45~17:00 (15分)	まとめ
17:00	(閉会)

memo

AdvanceSoft



音響解析ソフトウェアAdvance/FrontNoise FOCUSスパコンを使ったハンズオンセミナー

第1事業部
徳永健一、尾川 慎介
松原 聖

2016年5月19日（木）
アドバンスソフト株式会社

本日のセミナーの流れ

演習Ⅰ；消音器の解析 基本機能体験 （内部問題）	演習Ⅰ－1；試計算 演習Ⅰ－2；基本計算 演習Ⅰ－3；応用計算	小規模でのデータ作成方法 理論解と比較 境界条件を変更
演習Ⅱ；防音壁の解析 基本機能体験 （外部問題）	演習Ⅱ－1；基本計算 演習Ⅱ－2；可視化 演習Ⅱ－3；応用計算	データ作成手順確認 音の広がりを確認する 吸音率を設定
演習Ⅲ；並列計算等	演習Ⅲ－1；メッシュ細分化 演習Ⅲ－2；並列計算実行	大規模計算の準備 FOCUS計算機の「良さ」体験
演習Ⅳ；その他	・ご要望事項・質疑応答 ・残り課題 ・追加演習A、追加演習B	残課題あれば対応

本日よりご利用いただくソフトウェア

- プリポストAdvance/REVOCAP for Advance/FrontNoise
 - Windows 7
 - 各自インストールを行っていただく
- 音響解析ソフトウェアAdvance/FrontNoise Ver.4.3
 - FOCUS計算機サーバーのAシステムを利用して計算を実行します。
 - 計算サーバーはLinux環境です。
 - PC上でTeratermを利用して、PCから計算サーバーに接続します。計算の実行はバッチ処理で行います。
 - PC上のWinSCPを利用して、PCから計算サーバーにファイル転送します。一部の方にはWinSCPで計算サーバーのファイル編集を行っていただきます。
- 注意
 - セミナー中に作成いただいたファイルや実行結果は、本日よりで消去されます。

PC準備 ; Advance/REVOCAP等

- AdvanceREVOCAP_InstallManual.pdf の手順の通りにインストールします。
 - setup.exeを実行します。
 - ライセンスファイルはこの講習会用のものが用意されています。

FOCUS計算機準備

- TeraTermでFOCUS計算機に接続後に以下の操作を実施。
- 計算機環境(つぎの文字列を入力してください。)
 - \$ source ../share/fn-start
- データ等の解凍と展開
 - \$ fn-copy.sh
 - (上書きを防ぐため「Really?」確認されますので、「y」で答えてください。
現在の場所に「FrontNoise」というディレクトリが作成されます。)
- 確認
 - \$ ls FrontNoise
 - このコマンドで「ex1 ex2 exA exB result」と表示されます。
- Linux上のファイルの編集について
 - Linuxに慣れていらっしゃる方は、viでもemacsでも。
 - Linuxに慣れていない方は、WinSCPのエディタをお勧めします。

FOCUS計算機での内容

- データ内容

ディレクトリ		内容
FrontNoise	ex1	演習1および演習3のデータ
	ex2	演習2のデータ
	exA	追加演習Aのデータ
	exB	追加演習Bのデータ
	result	本日作業の結果で作成されるファイルのすべて

- すべての計算はバッチジョブで実行します。バッチファイルはデータディレクトリにgo*.shの名称でサンプルがあります。
- FOCUS計算機のAシステムのみを利用します。利用するジョブクラスはa024hです。

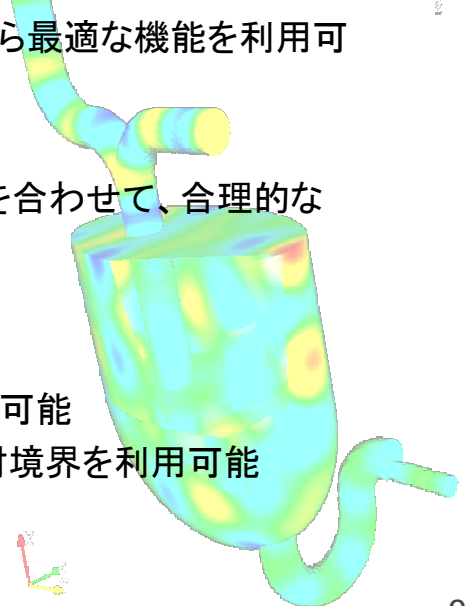
Advance/FrontNoiseとは

ソフトウェアの機能

項目	内容	
基礎方程式	(1) 基礎方程式は、音響ポテンシャルに関する波の方程式を周波数空間に変換した方程式 (2) 空間的に分布する場の流れおよび空間的に分布する音響伝播媒体を考慮可能	
解析領域	内部領域、および、外部領域(外部領域での外部境界はpc境界で与える)	
物性値等	速度	場の速度を指定可能(デフォルト;速度0)
	温度	場の温度(音の伝播媒体)を要素毎に指定することが可能(デフォルトは均一媒体)
境界条件	面での音源	面(壁境界)に対して、周波数毎に音圧または粒子速度を設定可能
	点音源	節点に対して、単極子、双極子、または、四重極子のパラメータを設定可能
	音響インピーダンス	面(壁境界)に対して、周波数毎に音響インピーダンスを設定可能
数値解法	外部境界	面(外部境界)に対して、pc境界を設定可能(音響インピーダンスを与える機能の一部)
	離散化手法	有限要素法
	利用可能な要素	四面体一次要素
	並列計算	自動領域分割によりMPIで並列化
	行列解法	GMRES系列の反復法
解析結果	大規模計算実績	4億要素・8000万節点(四面体一次要素)
解析結果	周波数毎の音響ポテンシャル、音圧、音圧レベルを、バイナリ形式でファイル出力	
プリポスト	Advance/REVOCAP for FrontNoise	

Advance/FrontNoiseの特長

- 大規模高速計算のさらなる強化
 - テトラ4億要素(7500万節点、15000万自由度)の解析実績
 - 392GBメモリ48CPUの計算機環境で処理時間は12時間
 - 並列手法は、領域分割・周波数分割から最適な機能を利用可能
- 合理的な価格
 - プリポストとソルバ並列計算無制限版を合わせて、合理的な年間ライセンスでご提供
 - 必要に応じて、ソルバのみもご提供
- その他の特長
 - 温度場、流れ場を考慮した音響解析が可能
 - 周波数依存性のある境界条件、無反射境界を利用可能
 - 形状適合性の高いテトラ要素を採用



専用プリポスト Advance/REVOCAP for FrontNoise

- Advance/FrontNoiseと一体化したプリポスト
- 音響解析ではほぼ均一なメッシュを作成することが多い。Advance/REVOCAPの自動メッシュの機能は、音響解析には非常に適合性が高い。
- 形状適合性の高い四面体ソルバ用のプリポストである。
- 境界条件設定作業が効率的になった。

音響解析に必要な情報

- メッシュ
 - 節点
 - 要素
- 境界条件
 - 場所の情報
 - 値
- 音源
 - 場所の情報
 - 値
- 解析条件
 - 開始・終了周波数
 - 周波数間隔
 - 音速・密度
 - 収束のための最大反復回数

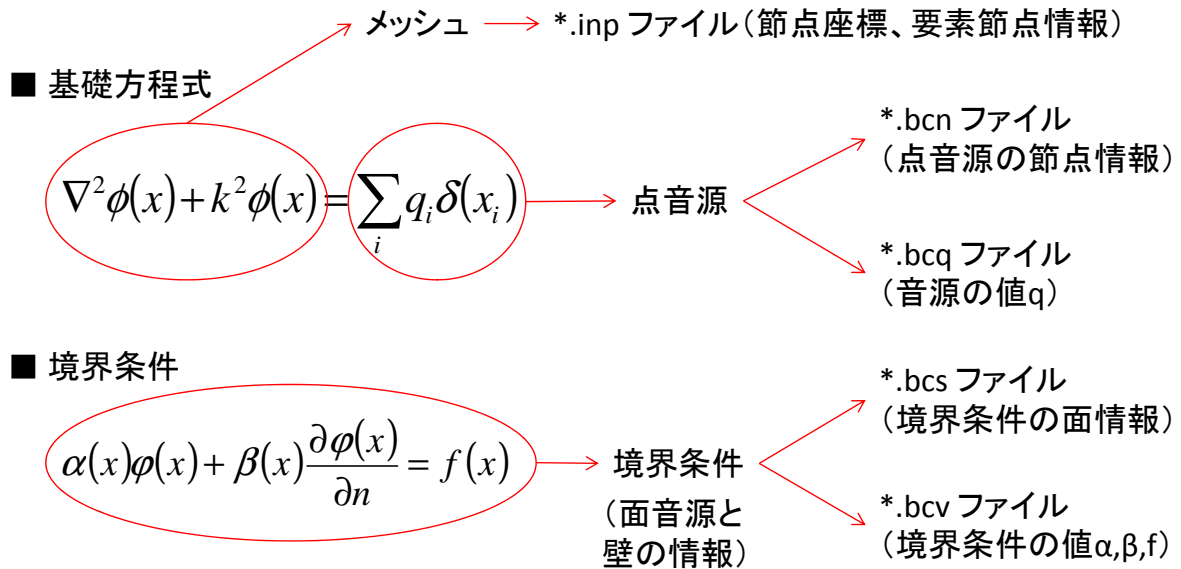
音響解析のために準備するファイル

項目	ファイル名	代表的なファイル作成方法
メッシュ	*.inp	メッシュ作成ソフト(Advance/REVOCAP)で作成する。または、メッシュ作成ソフトで作成したファイルをコンバートして作成する。
境界条件(面音源を含む)を面に与える場合	*.bcs	メッシュ作成ソフト(Advance/REVOCAP)で作成する。または、メッシュ作成ソフトで作成したファイルをコンバートして作成する。
	.bcv	実験値・理論値・文献値からexcel等で条件を作成して、csvでexportしたファイルを.bcvファイルとする。
点音源を節点に与える場合	*.bcn	メッシュ作成ソフト(Advance/REVOCAP)で作成する。または、メッシュ作成ソフトで作成したファイルをコンバートして作成する。または、節点探索ツールnfindを利用する。
	.bcq	手入力等で音源ファイルを作成して.bcqファイルとする。

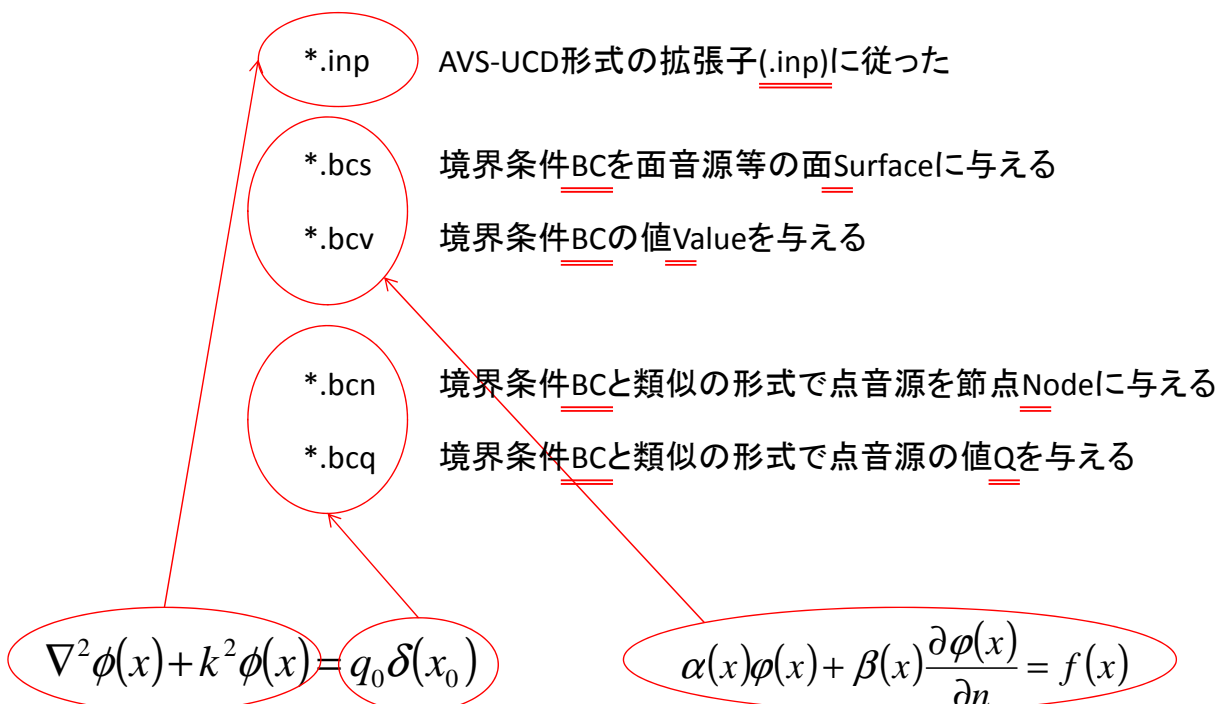
$$\nabla^2 \phi(x) + k^2 \phi(x) = q_0 \delta(x_0)$$

$$\alpha(x)\phi(x) + \beta(x)\frac{\partial \phi(x)}{\partial n} = f(x)$$

理論と入力データの関係



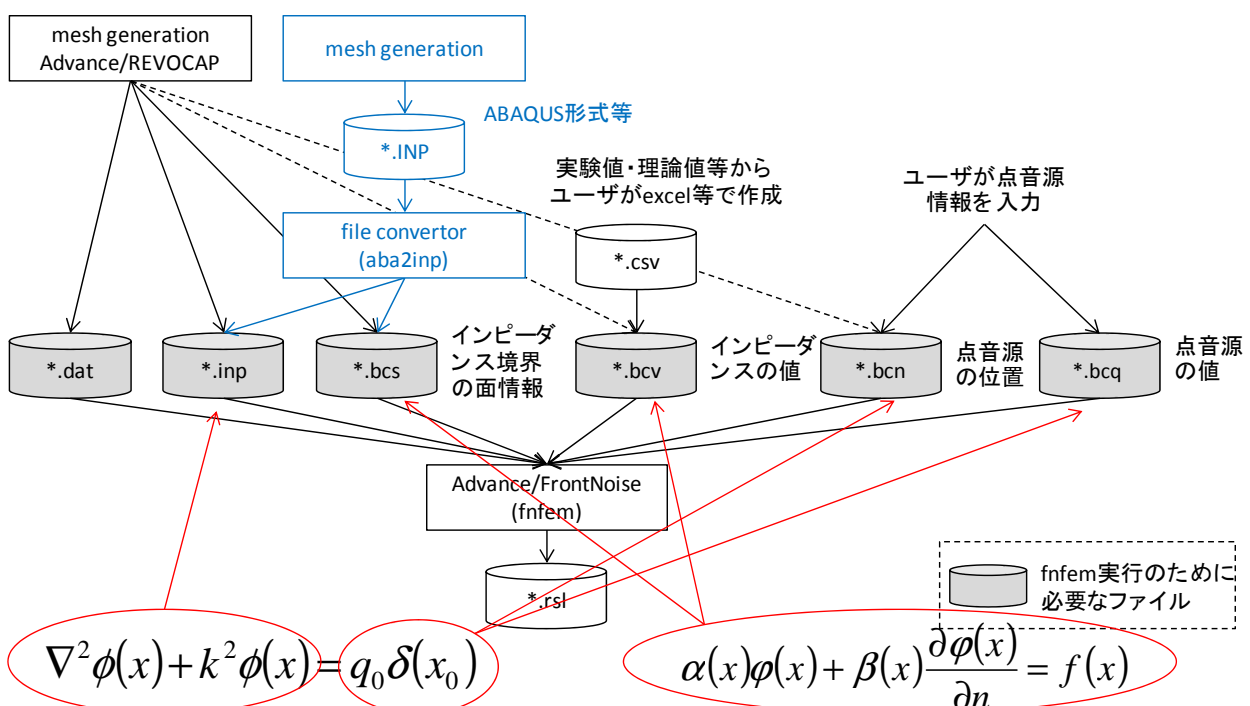
ファイルの拡張子名の由来



解析手順

- 格子ファイル(*.inp)を作成する。
 - 四面体1次要素
- 2つ(位置*.bcsと値*.bcv)の境界条件ファイルを作成する。
 - 境界条件の節点・面の情報
 - 境界条件の値を指定する
- (必要に応じて、2つの点音源ファイル(節点*.bcnと値*.bcq)を作成する。)
- 制御データ(*.dat)を作成する。
 - 解析対象とする周波数の指定
 - 音速と密度の指定
 - 収束最大回数等の計算パラメータ

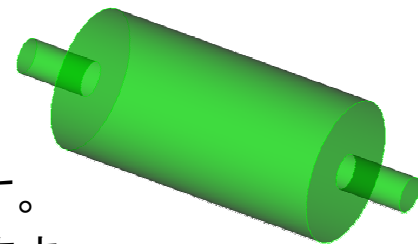
音響解析の流れと必要なデータ



演習 I - 1 ; 試計算	小規模でのデータ作成方法
演習 I - 2 ; 基本計算	理論解と比較
演習 I - 3 ; 応用計算	境界条件を変更

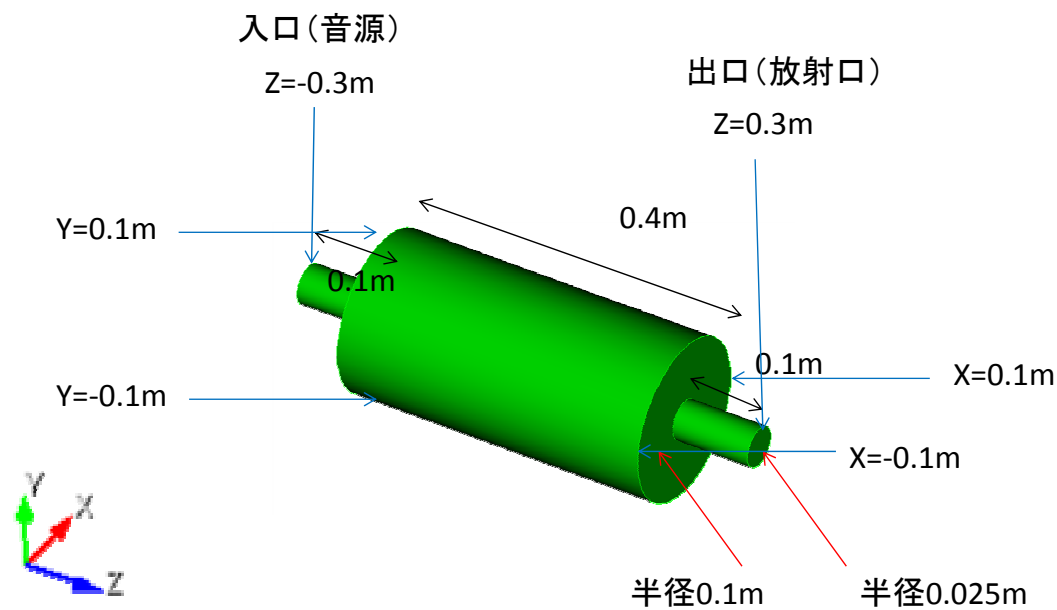
演習 I ; 消音器の解析

解析条件



- 単純形状のマフラーの解析を行います。
- マフラーの入口に1[Pa]の音圧を与えます。
- マフラー壁は完全反射とします。
- 演習 I - 1, 2ではマフラー出口は透過(完全吸収)条件とする。演習 I - 3ではより現実的な境界条件を設定します。
- 演習 I - 2, 3では、10Hzから2kHzまで10Hz毎に計算を行います(演習 I - 1では、100Hzから2kHzまで100Hz毎)。
- 出口の音圧を求め、入口の音圧との比を求めます。
- 1次元の理論解と本結果(3次元のシミュレーション)を比較します。

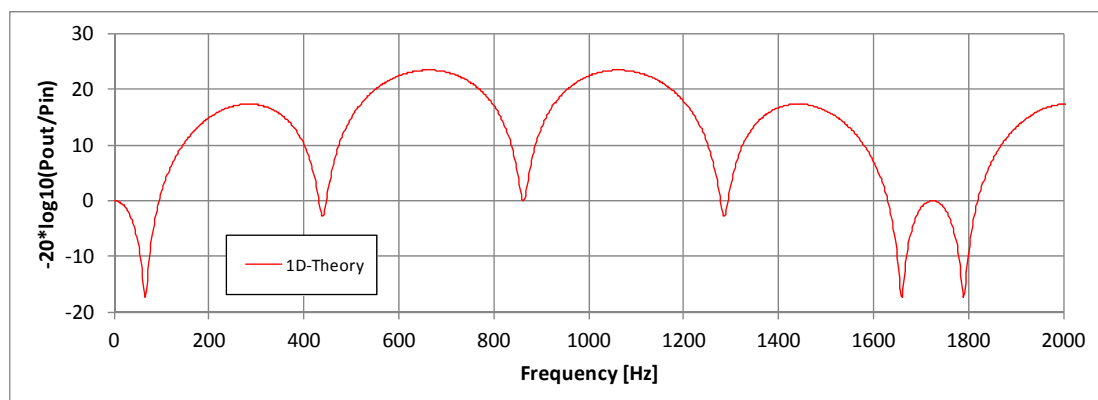
解析形状



形状の中心を座標の原点としました

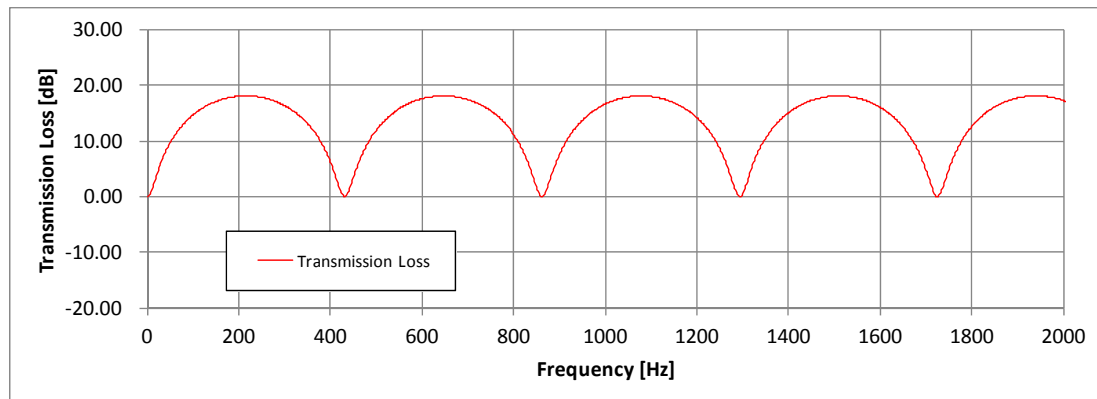
1次元理論解

- 入口の音圧 P_{in} と出口の位置の音圧 P_{out} の比 P_{out}/P_{in} を求めます。この値は1次元の伝達関数を利用して、周波数毎に求めることができます。本日の計算では、入口の面に1Paを与えた場合の出口の音圧を求めます。



1次元理論解の透過損失

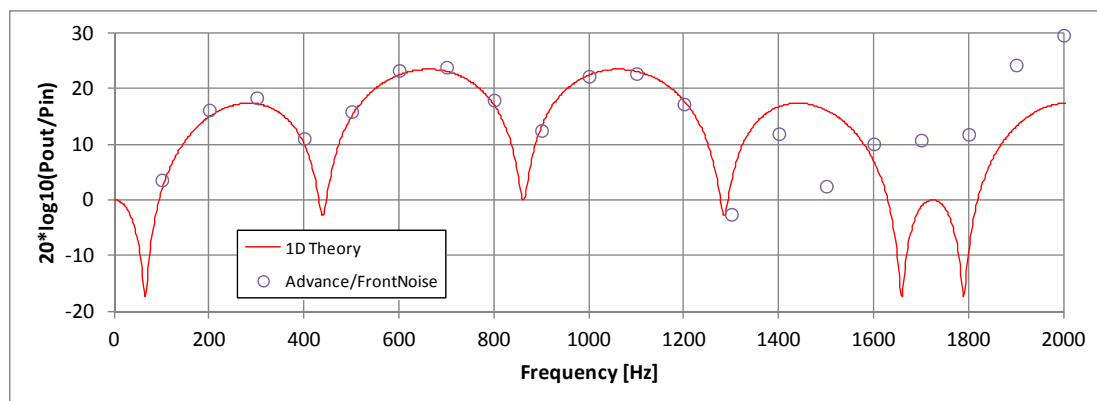
- 理論解から同様にして透過損失を求めることができる。また、解析結果を利用して透過損失を求めることができる(例えば、実験でも利用される2マイクロフォン法を利用する)。本日は計算結果から透過損失を求めることはしませんが、参考までこのケースの透過損失を示します。



演習 I - 1 ; 試計算

まずは、100Hz毎の試計算

- 理論解と一致するかどうかを確認することを目指して計算を行います。理論解のグラフを得るためには、多くの周波数で計算する必要があります。
- 最初に何点かの周波数でデータの確認をして、その次に多くの周波数を計算する手順とします。ここでは、まず、100Hzから2kHzまで100Hzごとに20点の周波数での計算を行います。下記のグラフの○を得ることを目的に計算を行います。



Copyright ©2016 AdvanceSoft Corporation. All rights reserved.

23

境界条件の種類

周波数領域での境界条件はどのように与えればいいのか？

$$\alpha(x)\varphi(x) + \beta(x)\frac{\partial\varphi(x)}{\partial n} = f(x) \quad \text{on } \partial\Omega$$

- 入口は音圧の境界としたい → つぎの一覧表から下記のように設定する。

$$i\omega\rho\varphi(x) = p_0(x) \longrightarrow i\omega\varphi(x) = -\frac{p_0(x)}{\rho}$$

数値を計算しやすくするために

- 出口は ρc 境界（インピーダンスを ρc にする。吸収境界=無反射）としたい。
→ つぎの一覧表から下記のように設定する。

$$i\omega\rho\varphi(x) - \rho c \frac{\partial\varphi(x)}{\partial n} = 0 \longrightarrow i\omega\varphi(x) - c \frac{\partial\varphi(x)}{\partial n} = 0$$

数値を計算しやすくするために

Copyright ©2016 AdvanceSoft Corporation. All rights reserved.

24

境界条件一覧

$$\alpha(x)\varphi(x) + \beta(x)\frac{\partial\varphi(x)}{\partial n} = f(x)$$

境界条件	$\alpha(x)$	$\beta(x)$	$f(x)$	備考
音圧	$i\omega\rho$	0	$-p_i(x)$	面音源
粒子速度	0	1	$v_i(x)$	面音源
粒子変位・加速度	0	1	$-i\omega u_i(x)$ 、 $-\frac{1}{i\omega}a_i(x)$	面音源
インピーダンス	$i\omega\rho$	$-Z$	0	吸音壁等
完全反射	0	1	0	デフォルト
完全吸収(無反射)	$i\omega\rho$	ρc	0	ρc 境界

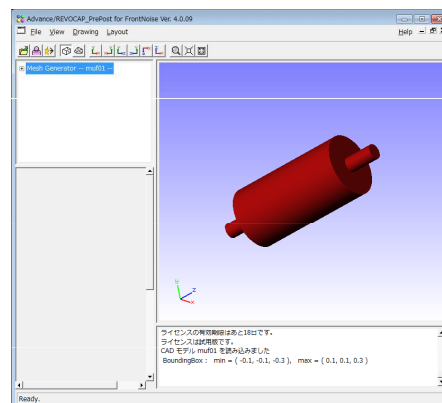
格子データの作成、格子幅の基準

- Advance/REVOCAPでメッシュを作成する。
- 四面体1次要素で格子データを作成する。
- FrontNoise形式でexportし、Advance/FrontNoiseの入力データとする。
- 使い方はやや面倒になるが、REVOCAP以外のメッシュも利用することは可能である。その場合には、特定の形式で出力し、FrontNoise形式に変換する。

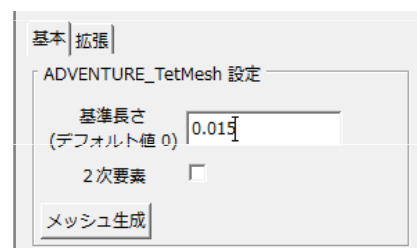
周波数 [Hz]	波長 [m]	推奨 ΔX [m]
200	1.650	0.206
400	0.825	0.103
600	0.550	0.069
800	0.413	0.052
1000	0.330	0.041
1200	0.275	0.034
1400	0.236	0.029
1600	0.206	0.026
1800	0.183	0.023
2000	0.165	0.021

Advance/REVOCAPでメッシュの作成

- ① FileメニューからImportCADを選択して、muf01.stepを読み込みます。

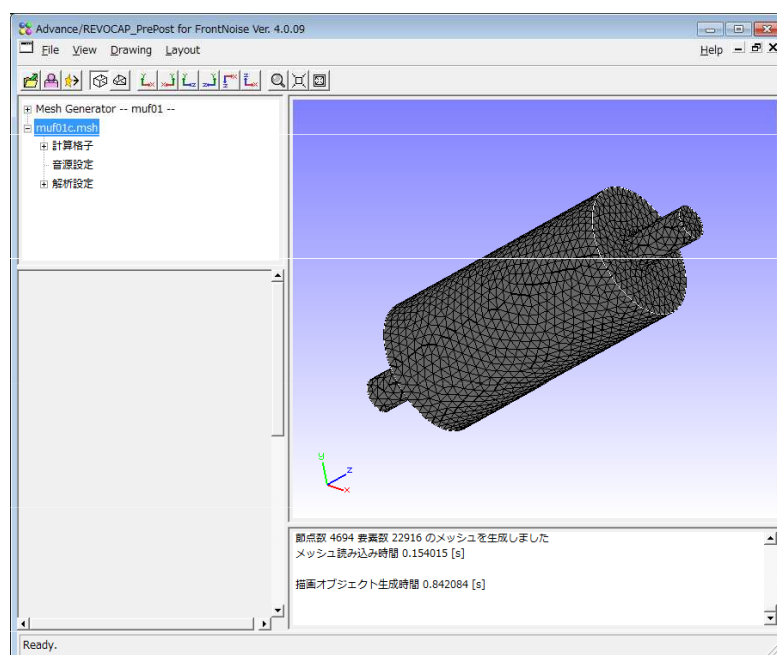


- ② TetMeshのメニューで基準長さを0.015と入力してメッシュ生成ボタンを押す



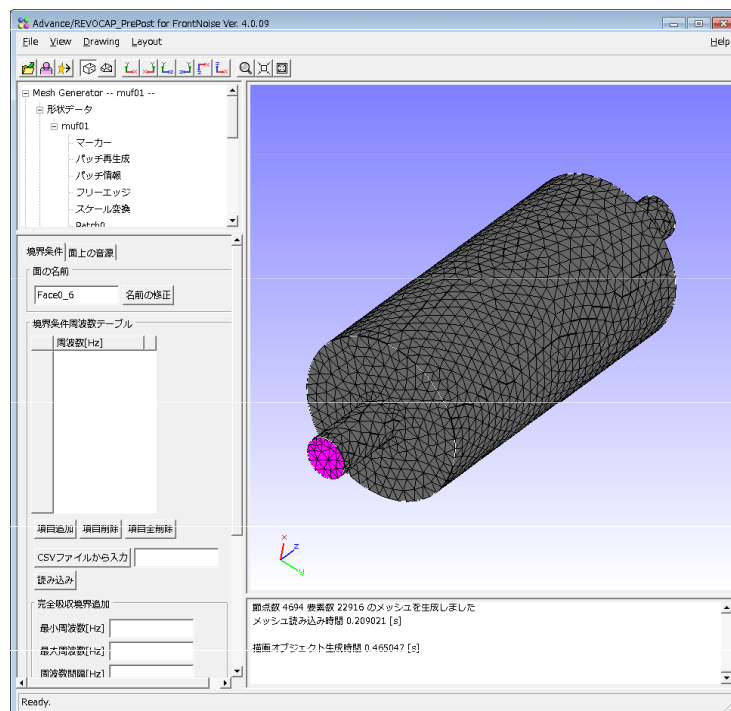
メッシュの作成結果

- ③ しばらく待つと作成されたメッシュが表示されます。



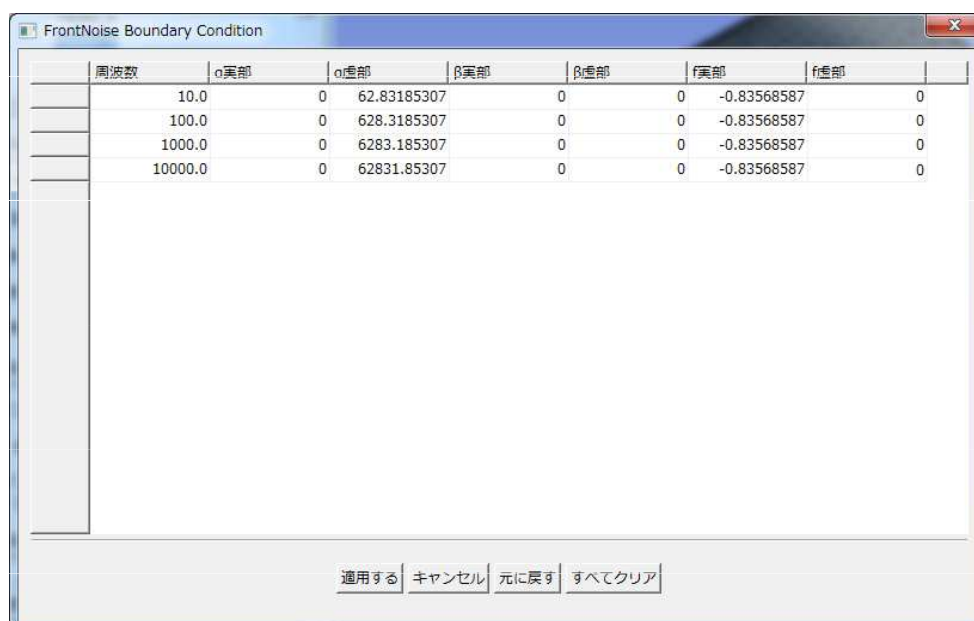
Advance/REVOCAPの設定方法(入口)

- ① 入口の面(z座標の小さい方)をマウスで選択します。
- ② 周波数テーブルに、設定対象の周波数10,100,1000,10000を追加します。



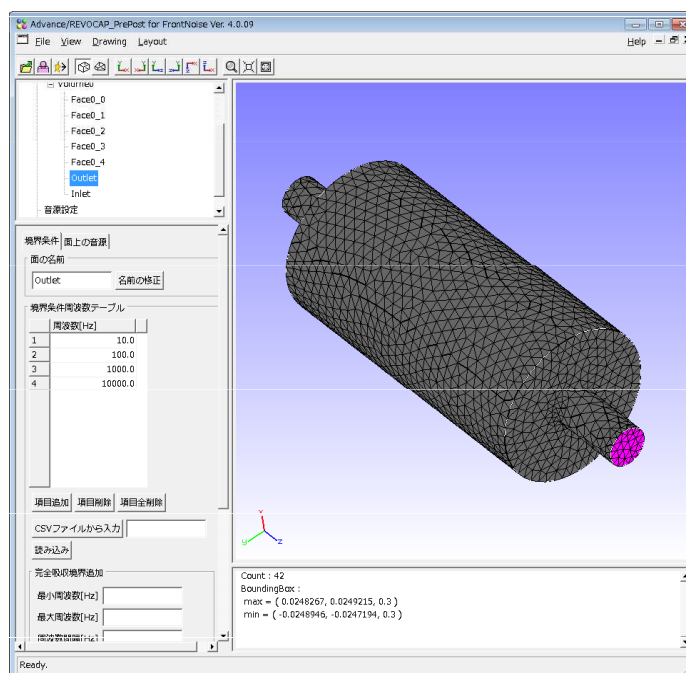
Advance/REVOCAPの設定方法(入口)

- ③ α 、 β 、 f の確認・編集ボタンを押して数値を入力します。



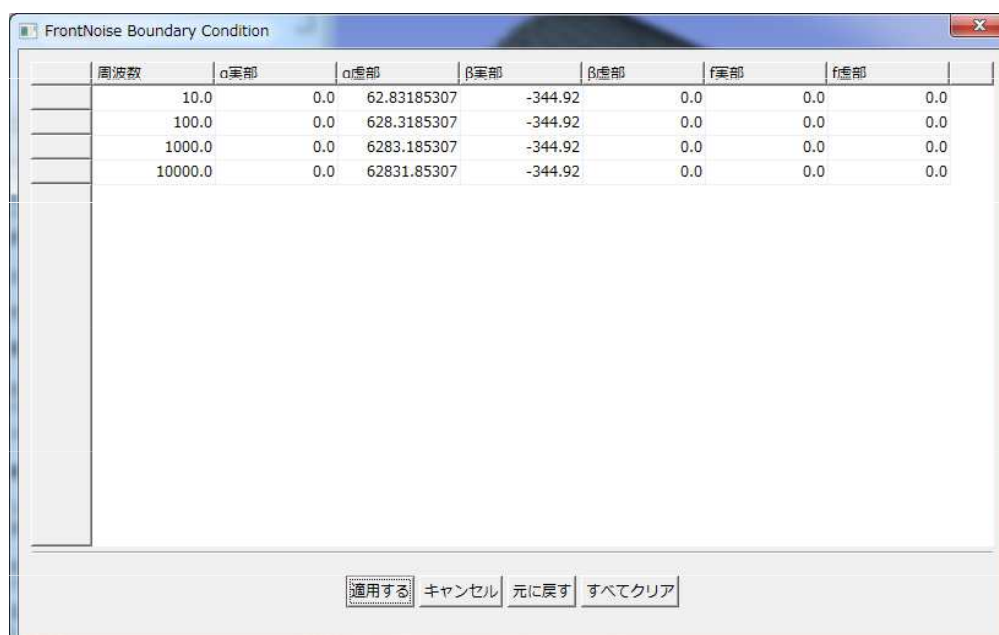
Advance/REVOCAPの設定方法(出口)

- ① 出口の面をマウスで選択します。
- ② 周波数テーブルに、設定対象の周波数10,100,1000,10000を追加します。



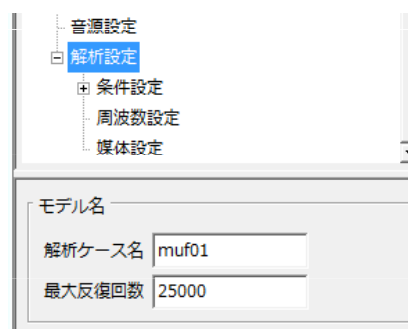
Advance/REVOCAPの設定方法(出口)

- ③ α 、 β 、 f の確認・編集ボタンを押して数値を入力します。

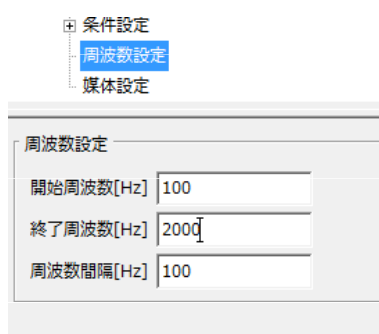


Advance/REVOCAPの設定方法(解析)

- ① 「解析設定」でケース名と最大反復回数を与える



- ② 「周波数設定」で開始、終了周波数と間隔を与える

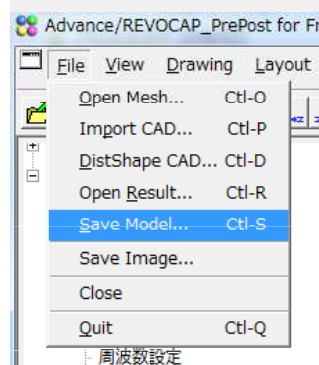


Advance/REVOCAPの設定方法(解析)

- ③ 「媒体設定」で音速と密度を与える



- ④ Fileメニューの「Save Model」からモデルを保存する



【確認】境界条件の値 *.bcv

$$\alpha(x)\varphi(x) + \beta(x)\frac{\partial\varphi(x)}{\partial n} = f(x)$$

周波数	α の実部	α の虚部	β の実部	β の虚部	fの実部	fの虚部
10	0	62.83185307	0	0	-0.83568587	0
10	0	62.83185307	-344.92	0	0	0
100	0	628.3185307	0	0	-0.83568587	0
100	0	628.3185307	-344.92	0	0	0
1000	0	6283.185307	0	0	-0.83568587	0
1000	0	6283.185307	-344.92	0	0	0
10000	0	62831.85307	0	0	-0.83568587	0
10000	0	62831.85307	-344.92	0	0	0

境界条件ファイル2(.bcv)

入口境界条件

出口境界条件

与えたい周波数テーブルの数だけ繰り返す

$$\alpha(x) = i\omega$$

$$\alpha(x) = i\omega$$

$$\beta(x) = 0$$

$$\beta(x) = -c$$

$$f(x) = -p_0/\rho$$

$$f(x) = 0$$

※ データを計算サーバーに移動した直後に、下記のコマンドを実施していただきたい。
`$ bcvconv muf01.bcv muf01.bcv` (本日だけの事情によるフォーマット変換)

【確認】制御データ *.dat

- 解析対象とする周波数、音速と密度、収束の最大反復回数等を指定する。
- 流体解析や構造解析と比較すると、音響解析における設定条件の数は、圧倒的に少ない。

```

Model Name      = muf01
Sound Velocity  = 344.92
Density         = 1.1966
Start Frequency = 100
Final Frequency = 2000
Frequency Step  = 100
Max Iteration   = 25000
END

```

```

ケース名
音速 [m/sec]
密度 [kg/m3]
開始周波数 [Hz]
終了周波数 [Hz]
周波数間隔 [Hz]
最大反復回数 [-]

```

さあ、ジョブを投入します。

- 利用するAシステムの計算機
 - 全体で224ノードであり、1ノードは12CPUです。ノード内は共有メモリであり、ノード間はInfiniband-QDR(40Gbit/s)で接続されています。また、1ノードは、Intel Xeon L5640(2.26GHz) × 2CPU(計12コア)で構成され、ノード当たりのメモリは48GBです。
 - 本日はAシステムのa024hのクラスを利用します。おおよそ100並列程度まで(可能であればもう少し多くの並列数)を体験していただくことを目標とします。
 - 演習 I と演習 II では、小規模なデータのため、シングルプロセス(1並列)の計算を行います。
- ジョブ投入等に関するコマンドはつぎの通りです。

\$ sbatch [ジョブスクリプト・ファイル名]	ジョブの投入
\$ squeue	ジョブの監視
\$ scancel [JOBID]	ジョブのキャンセル

ジョブスクリプト(ex1/go01.sh)

```
#!/bin/bash
#SBATCH -p a024h   ジョブクラスを指定(変更しないでください)
#SBATCH -n 1      並列数を指定
#SBATCH -N 1      利用ノード数を指定
#SBATCH -J ex1-1   自由に命名するジョブ名(squeueで表示)
#SBATCH -o %J.stdout  出力ファイル名(ジョブ番号+stdout)
#SBATCH -e %J.stderr  エラーファイル名(ジョブ番号+stderr)
module load PrgEnv-intel   おまじない
module load impi           同上
export PATH=/home1/glen/share:$PATH  同上
export FNFEM_SOLVER=80     同上(のちほど説明)
mpirun -np $SLURM_NTASKS fnfem < muf01.dat FrontNoise実行
```

実行するコマンド例

```
$ sbatch go01.sh
Submitted batch job 781859
$ squeue
.....
781858 a024h ex1-1 ulen0034 R 0:11 1 a210
($ scancel 781858 でジョブ中止できます)
このジョブは20秒以内で終了します。
$ ls
781858.stdout 781858.stderr
muf01-r0001....rs1(20個の結果ファイル→これを使います)
$ (ここで、781858.stdout を見てみます)
```

標準出力ファイル

Advance/FrontNoise/FEM/MPI Ver4.3
April 30, 2014

計算条件の確認

```
< file information >
- FEM grid file .....: muf01.inp
- boundary surface file .....: muf01.bcs
- boundary surface value file ....: muf01.bcv
- boundary node file .....: muf01.bcn
- boundary node value file .....: muf01.bcq
- header of result file .....: muf01-r***.rs1
```

```
< analysis conditions >
- sound velocity[m/sec] .....: 3.449E+02
- density[kg/m3] .....: 1.197E+00
- freq[Hz] (start) .....: 1.000E+02
- freq[Hz] (terminate) .....: 2.000E+03
- freq[Hz] (interval) .....: 1.000E+02
- number of nodes .....: 5337
- number of elements .....: 26785
- number of processors .....: 1
- iterative method .....: DIRECT
- pre condition method .....: DIRECT
- # restart for gmres .....: 30
```

```
< boundary conditions >
- number of surface group .....: 2
- number of surface .....: 36
- number of surf. freq. table ....: 6
- number of point sources .....: 0
- number of pt-src. freq. tab. ..< 6
```

* start of frequency domain *

(1) frequency 100.00[Hz]

```
* frequency[Hz] = 100.00
* number of iter. = 0
* rel. tol. (||Ax-b||/||Ax||) = 1.000E+20 ( 1.000E-08)
* abs. tol. (||Ax-b||) = 1.000E+20 ( 1.000E+20)
```

各周波数での計算

(2) frequency 200.00[Hz]

***** ACCOUNTING TABLE OF THE CPU-TIME *****

処理時間

PROCESS	TOTAL TIME
< 1> Total Time	: 7.43D+00 (100.0)
< 2> Input	: 7.89D-02 (1.1)
< 3> Time Loop (pre)	: 1.67D+00 (22.4)
< 4> Linear Solver (pre)	: 1.18D-02 (0.2)
< 5> Linear Solver	: 5.33D+00 (71.7)
< 6> Linear Solver (post)	: 1.82D-03 (0.0)
< 7> Time Loop (output)	: 3.44D-01 (4.6)
< 8> Post Process	: 1.00D-04 (0.0)

normally terminated

正常終了

解析結果編集の代表的な2つの手順

- muf01-r0006-00700.rsl は、「ケース名」+「ステップ番号」+「周波数」の命名規則となっています。この結果からつぎの2通りのパターンで結果を抽出します。
- 【①結果ファイル群から、指定座標結果の取り出し】ここでは、結果ファイル群から、出口1点(x,y,z)=(0.0, 0.0, 0.3)の結果を取り出します。その結果をファイルに出力します。

```
$ ls *.rsl | getnvalbin 0 0 0.3 muf01.inp > o1.csv
```

- 【②結果ファイルを可視化ファイルに変換】結果ファイルのひとつをREVOCAPやParaViewで可視化可能なファイルに変換します。例えば1000Hzの結果を「・・・」という名称のファイルに変換します。

```
$ rsl2inp muf01.inp muf01-r0009-01000.rsl out1000.inp
```

ひとことメモ

- Advance/FrontNoiseの「getnvalbin」や「rsl2inp」のようなツール群は、そのコマンド名のみを打ち込むと、その使い方を表示してくれます。

```
$ getnvalbin
```

```
Usage: getnval  % getnvalbin x0 y0 z0 grid-file
stdin  : file names      stdout : node values 0 0 0.3
muf01.inp > o1.csv
```

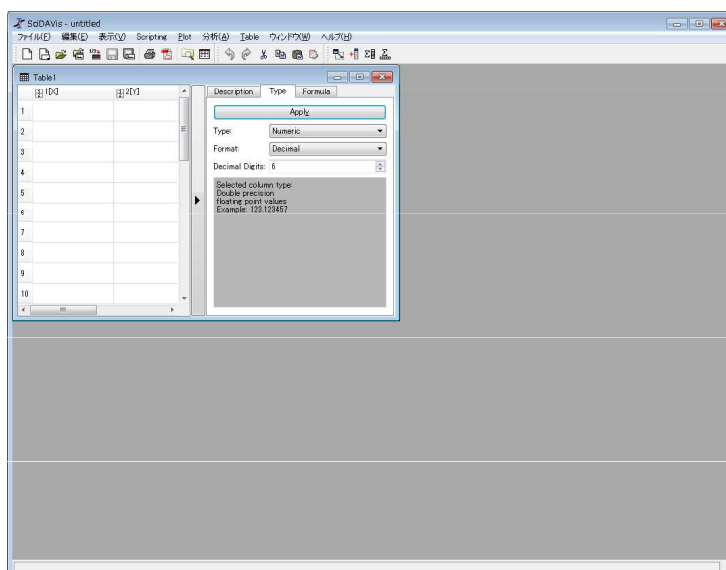
```
$ rsl2inp
```

```
rsl2inp inp-file binary-file vis-file
      inp-file      : grid file      (*.inp)
      binary-file   : result file    (*.rsl)
      ascii-file    : AVS UCD file   (*.inp)
```

- 前ページではLinuxの約束である「|」や「>」を使っています。前者は「出力を次のコマンドに送る」であり、後者は出力をファイル出力するのですが、不慣れな方はお近くのスタッフまで。

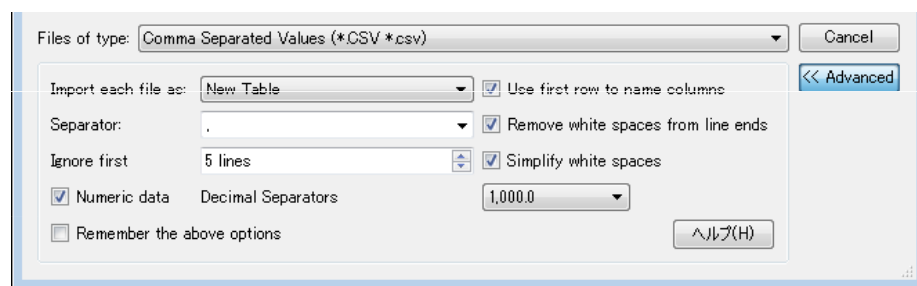
グラフの書き方 SciDAVis の準備

- 職場のPCではExcelなどの表計算ソフトを利用してください。
- ここではSciDAVisというグラフ作成ソフトを使って説明します。



音圧のグラフを書く手順(その1)

- ① ファイルメニューから Import Ascii を選択
- ② コマ区切りで最初の5行を無視してgetnvalbinで出力したファイルを読み込む(以下の設定)

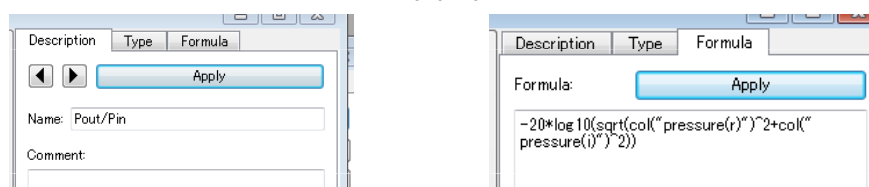


- ③ グラフを書くのに関係ない最後の2行は削除しておく
- ④ 音圧の実成分pressure(r)と虚成分pressure(i)の複素数の絶対値の対数値を計算する列を作成する。ここでは列の名前をPout/Pinとする。右クリックでInsertEmptyColumnsを選択して空の列を作る

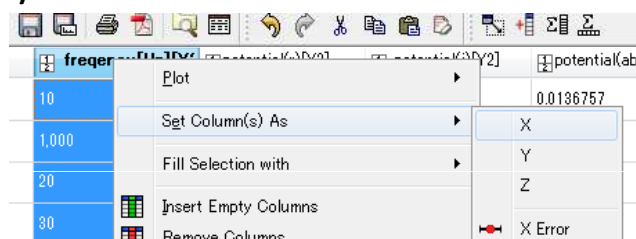
音圧のグラフを書く手順(その2)

- ⑤ 作成した列を選択して、右画面のDescriptionタブをPout/Pinとし、Formulaタブに

$$-20 * \log_{10}(\sqrt{\text{col}(\text{"pressure(r")})^2 + \text{col}(\text{"pressure(i")})^2})$$
 となるように入力して、Applyボタンを押す

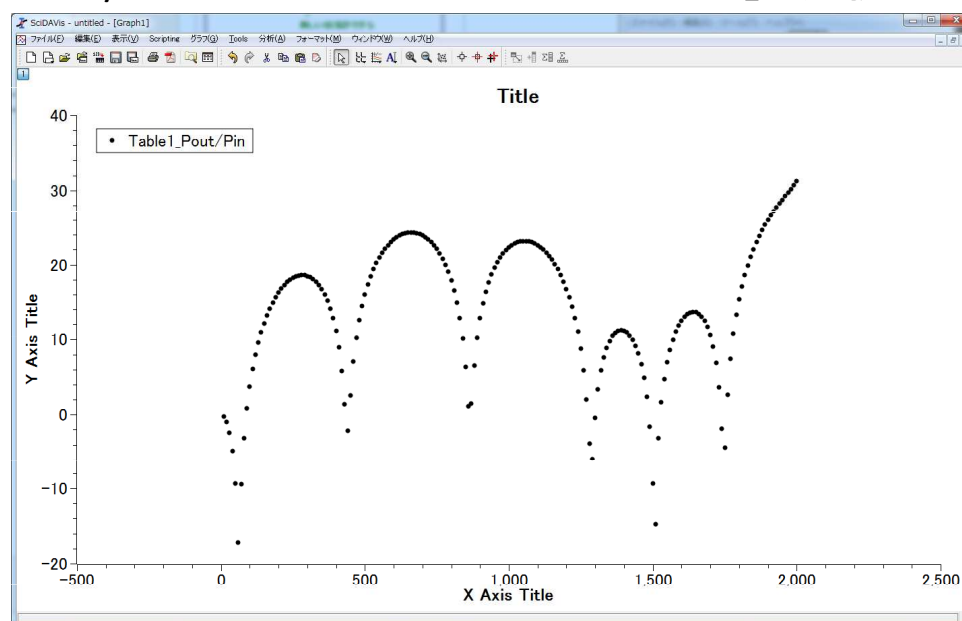


- ⑥ 列frequencyをXとする



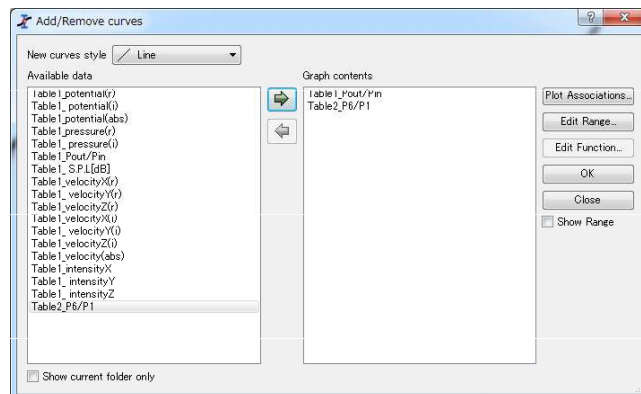
音圧のグラフを書く手順(その3)

- ⑦ 列Pout/Pinで右クリックしてPlot⇒「点のみ」を選択



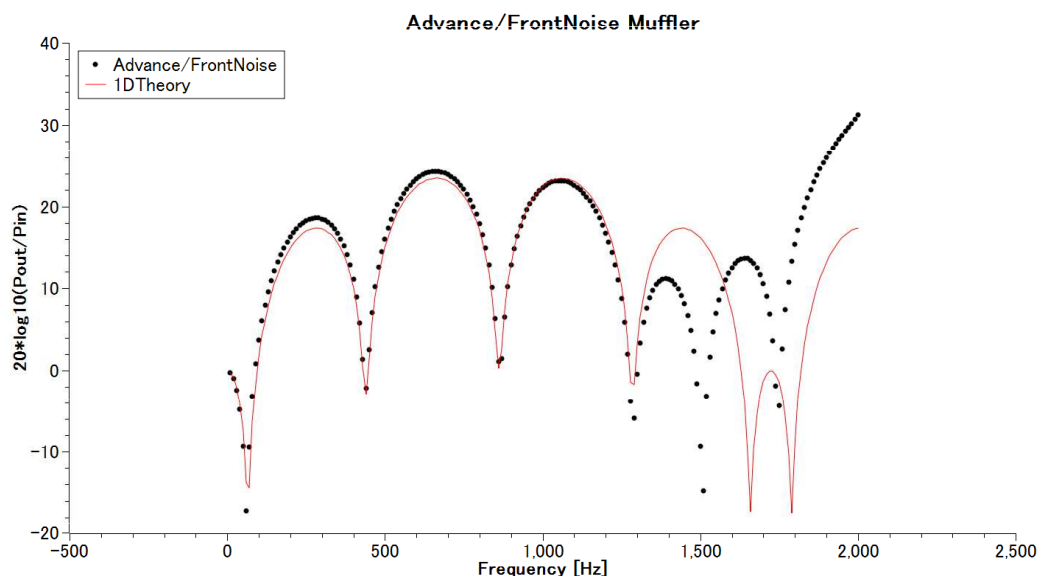
音圧のグラフを書く手順(その4)

- ⑧ 同様にして理論値 theory1.csv を読み込む(先頭行の無視は不要)。
- ⑨ 音圧のグラフに理論値を追加する。グラフメニューの「Add/Remove Curve」を選んで、Table2_P6/P1をNew curves style をLineにしてGraph contents に追加する



音圧のグラフを書く手順(その5)

- ⑩ 以下のようなグラフが得られます。タイトルなどは適宜修正しています。

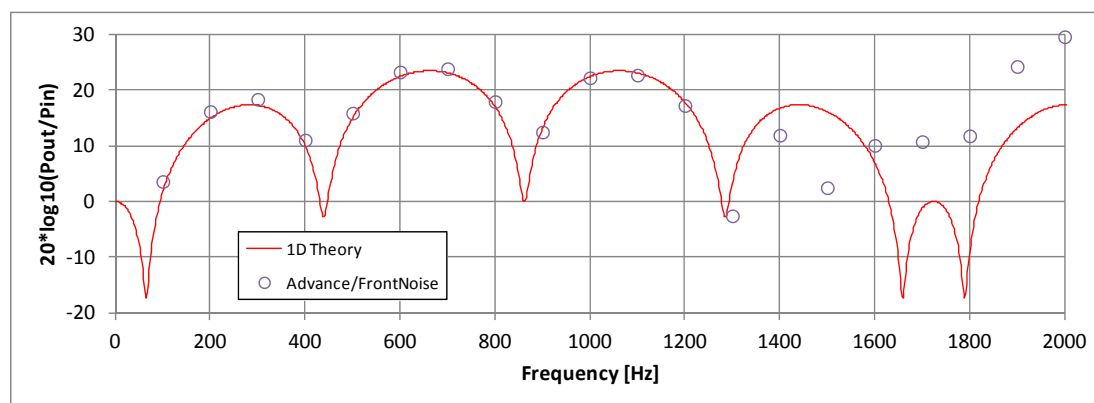


この結果が得られましたでしょうか？

- 結果確認後、結果ファイルをrs11に退避しておきます。

```
$ mkdir rs11
```

```
$ mv *.rsl rs11
```



演習 I - 2; 基本計算

次に10Hz毎、制御データ *.dat

- 解析周波数だけを変更して、解析を実施します。ここでは、10Hz毎に2kHzまで。200点の周波数の計算を行います。

Model Name	= muf01
Sound Velocity	= 344.92
Density	= 1.1966
Start Frequency	= 10
Final Frequency	= 2000
Frequency Step	= 10
Max Iteration	= 25000
END	

ケース名
音速[m/sec]
密度[kg/m ³]
開始周波数[Hz]
終了周波数[Hz]
周波数間隔[Hz]
最大反復回数[-]

この結果が得られましたでしょうか？

- muf01.datを変更後、つぎのコマンドを実行します。

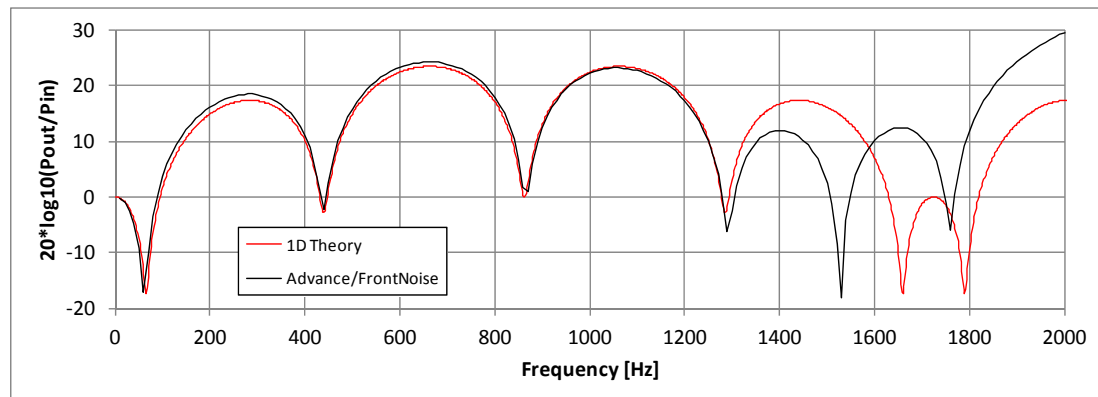
```
$ sbatch go01.sh
```

```
$ ls *.rsl | getnvalbin 0 0 0.3 muf01.inp > o2.csv
```

- 結果確認後、結果をrsl2に退避しておきます。

```
$ mkdir rsl2
```

```
$ mv *.rsl rsl2
```

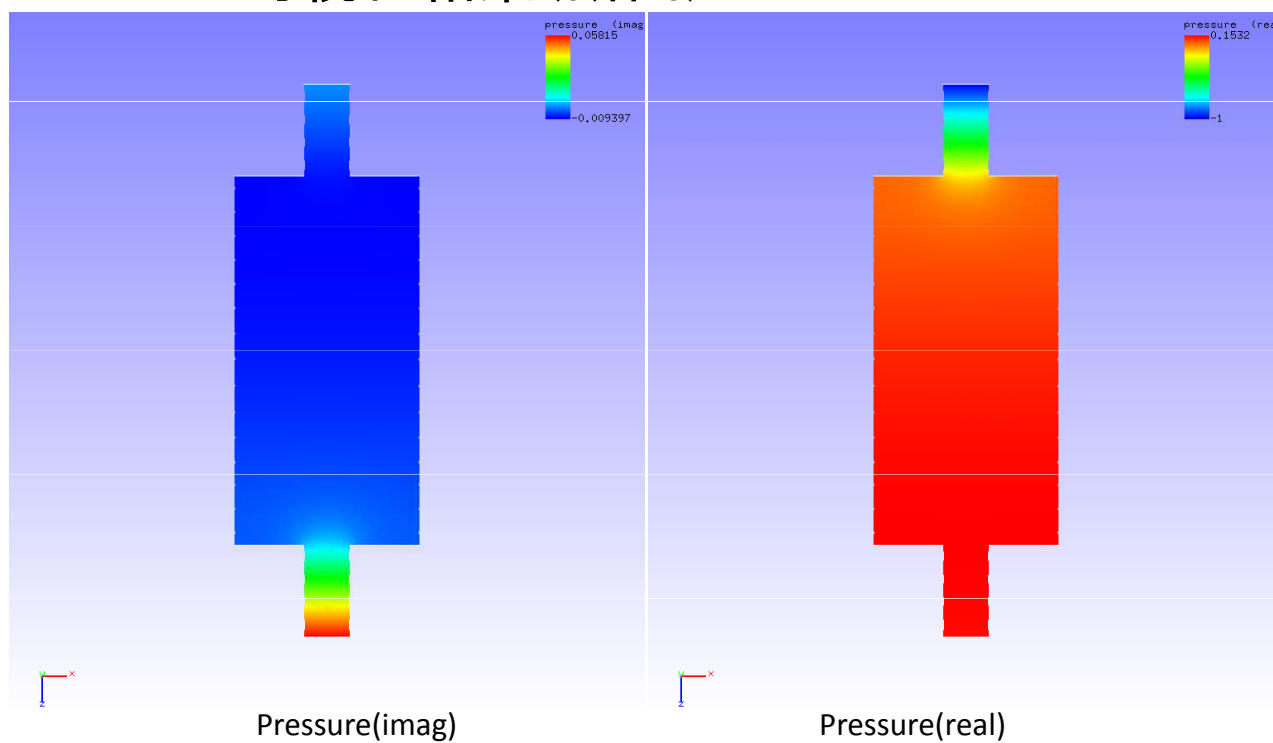


Advance/REVOCAPでの可視化

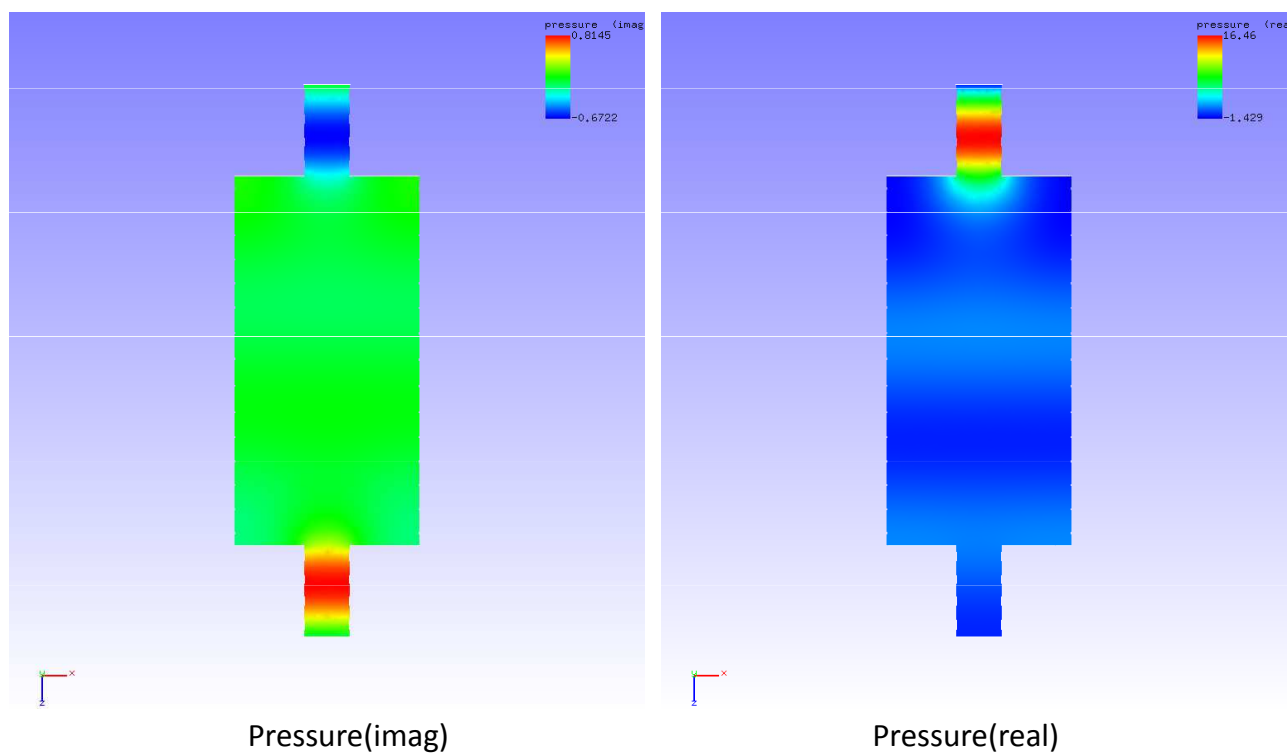
- ① Fileメニューから Open Result を選択して、rsl2inp で作成したUCD形式のファイルを開く
- ② ツリーメニューからContourを選択して、可視化したい項目をItemから選択して、設定ボタンを押し、カラーバー表示、コンター表示有効のチェックボックスをオンにする
- ③ ツリーメニューからSectionを選択して、法線ベクトルをy軸方向として、切断面表示を「断面表示」にする
- ④ 表示の座標系をZX平面にする



20Hzの可視化結果(断面)



1530Hzの可視化結果(断面)



高周波になると1次元の理論と合わない？

- 1次元の理論は、波長がダクトの直径よりも十分に長いという仮定です。
- 本日実施した解析は3次元の音響解析であるため、高周波では1次元の理論では表現できない部分が現われてきていると考えられます。

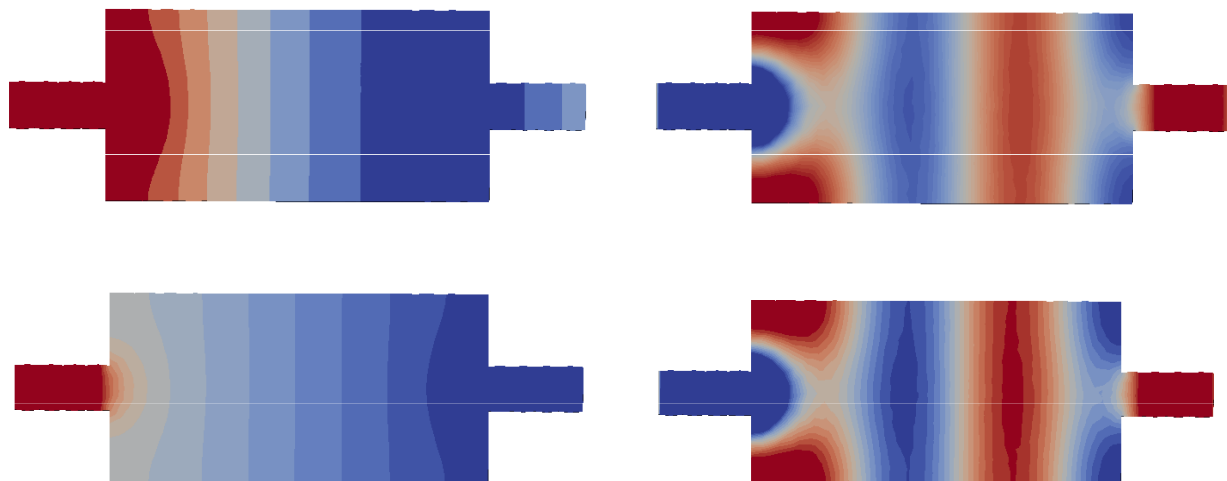
→

- 結果を可視化して、1次元の理論解とは異なる部分を見てみます。例えば、わかりやすい1530Hzの結果を見ます。1530Hzの波長は0.16m程度であり、マフラー胴体の直径0.2程度となっています。このあたりで1次元の分布とは異なってくる可能性があり、それを図で確認することができます。

可視化結果(断面)

低周波側

高周波側

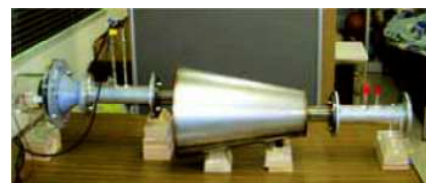


演習 I - 3; 境界条件変更

円形ダクトからの放射境界条件について

フランジあり円形ダクト

$$Z_r = \theta_r + i\chi_r = 1 - \frac{J_1(2ka)}{ka} + i \frac{H_1(2ka)}{ka}$$



鈴木昭次、西村正治等; "機械音響工学," コロナ社(2004)、4.2.3節

L.Rayleigh, "Theory of Sound," Macmillan and Company(1940)

※ $ka < 0.6$ での近似: $\theta_r \approx \frac{1}{2}(ka)^2, \chi_r \approx \frac{8}{3\pi}ka$ a はダクト半径

フランジなし円形ダクト

H.Levine, "On the Radiation of Sound from an Unflanged Circular Pipe," PHYSICAL REVIEW(1948)

※ これらの論文掲載の反射率 = Advance/FrontNoise で計算した反射率

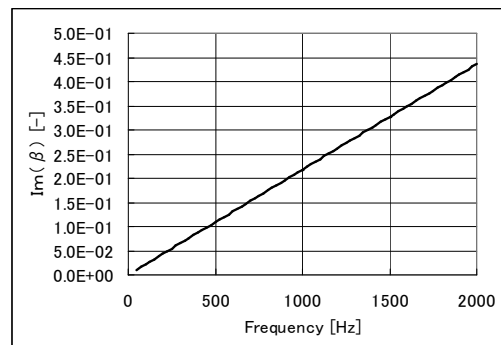
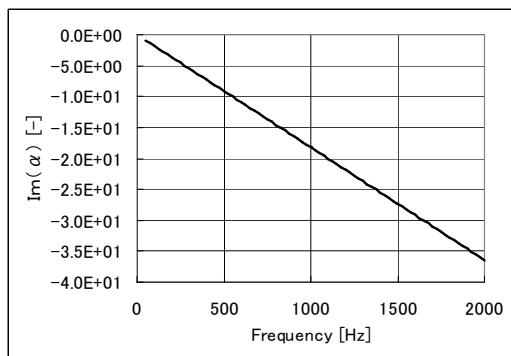
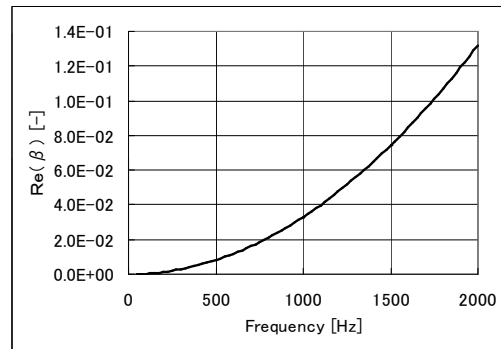
より実用的な出口境界条件 (円筒形ダクトの放射境界条件)

$$\alpha(x)\phi(x) + \beta(x)\frac{\partial\phi(x)}{\partial n} = f(x)$$

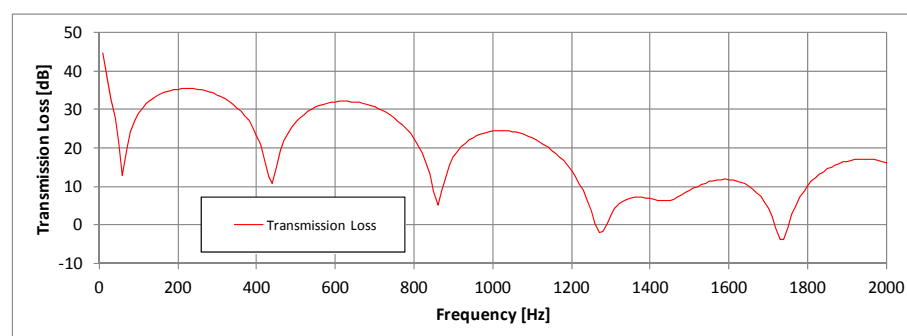
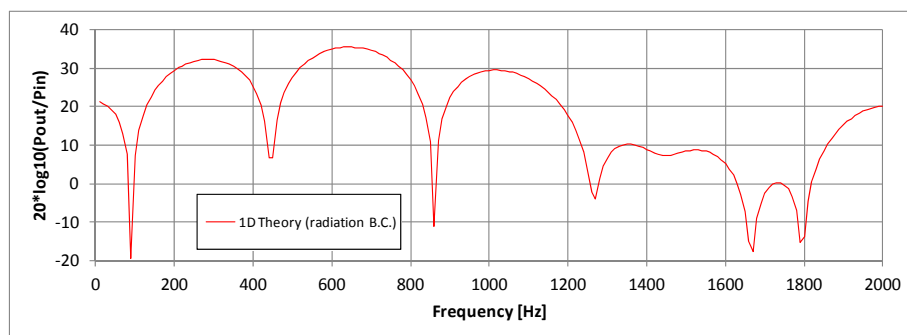
$$\alpha(x) = i\omega\rho$$

$$\beta(x) = -Z$$

$$f(x) = 0$$



出口音圧と等価損失(放射条件理論解)



境界条件*.bcv

- この境界条件は、通常は(お客様の良く慣れた)excelで作成して、貼り付ける手順が非常に便利です。ここでは、それを省略して、~/FrontNoise/ex1/muf01.bcv.radに該当するファイルを置いておきました。

- つぎのコマンドを実行します。

```
$ mv muf01.bcv muf01.bcv.rhoc
```

```
$ mv muf01.bcv.rad muf01.bcv
```

```
$ sbatch go01.sh
```

```
$ ls *.rsl | getnvalbin 0 0 0.3 muf01.inp > o3.csv
```

- 結果確認後、結果をrsl3に退避しておきます。

```
$ mkdir rsl3
```

```
$ mv *.rsl rsl3
```

【確認】境界条件の値 *.bcv

周波数	α の実部	α の虚部	β の実部	β の虚部	fの実部	fの虚部
10	0	62.83185307	0	0	-0.83568587	0
10	0	62.83185307	-0.00357677	-1.33333333	0	0
100	0	628.3185307	0	0	-0.83568587	0
100	0	628.3185307	-0.3576773	-13.3333333	0	0
500	0	3141.592654	0	0	-0.83568587	0
500	0	3141.592654	-8.94193255	-66.6666667	0	0
1000	0	6283.185307	0	0	-0.83568587	0
1000	0	6283.185307	-35.7677302	-133.333333	0	0
1500	0	9424.777961	0	0	-0.83568587	0
1500	0	9424.777961	-80.477393	-200	0	0
2000	0	12566.37061	0	0	-0.83568587	0
2000	0	12566.37061	-143.070921	-266.666667	0	0

入口境界条件

$$\alpha(x) = i\omega$$

$$\beta(x) = 0$$

$$f(x) = -p_0/\rho$$

出口境界条件

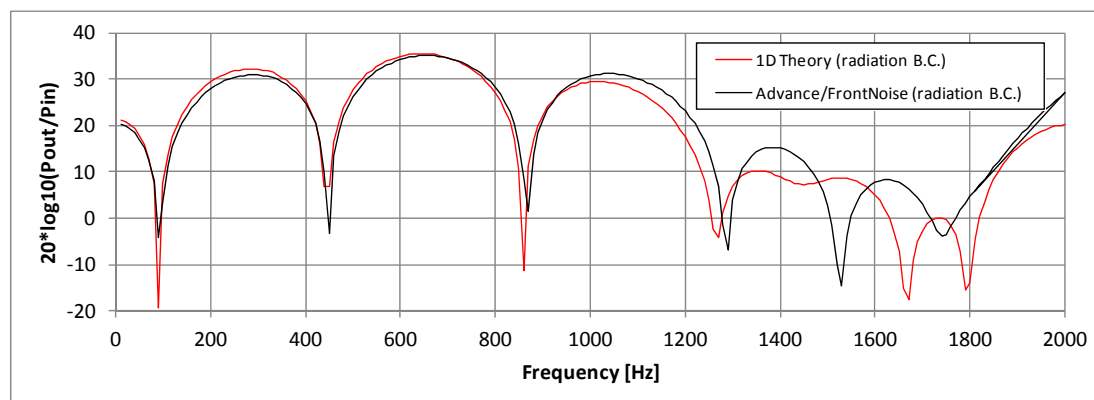
$$\alpha(x) = i\omega$$

$$\beta(x) = -Z/\rho = -(k^2 a^2/2 + i8ka/3\pi)/\rho$$

$$f(x) = 0$$

このような結果が得られましたでしょうか

- 理論解の結果ファイルが、~/FrontNoise/ex1/theory3.csvにあります。200点の計算結果と重ね描きしてみてください



演習 I のまとめ

■ 作成したメッシュ(演習 I 共通)

項目	内容
要素数	
節点数	

■ メモ

演習	内容
I - 1	muf01.bcv.rhoc利用
I - 2	同上、muf01.dat変更
I - 3	muf01.bcv.rad利用

■ 計算の結果

演習	出口境界	周波数		処理時間	ジョブ番号
I - 1	吸収境界	100~2000Hz	100Hz毎		
I - 2	吸収境界	10~2000Hz	10Hz毎		
I - 3	放射境界	10~2000Hz	10Hz毎		

演習 I のメモ

演習	メモ
I-1	
I-2	
I-3	

【休憩室A】マフラーミュージアムにて



演習Ⅱ－1;基本計算

データ作成手順確認

演習Ⅱ－2;可視化

音の広がりを確認する

演習Ⅱ－3;応用計算

吸音率を設定

演習Ⅱ ;防音壁の解析

解析条件(その1)

● 解析の概要

- － 高架橋上を列車が通過した場合に発生する高架橋付近の家屋における騒音レベル評価を模擬したデータです。
- － 電車の車輪付近の2か所の音源(同位相で同じエネルギー)を設定し、音源から空気中を伝播した音のみを対象としました。すなわち、車両から構造物を伝播して発生する音は考慮していません。

● 解析条件

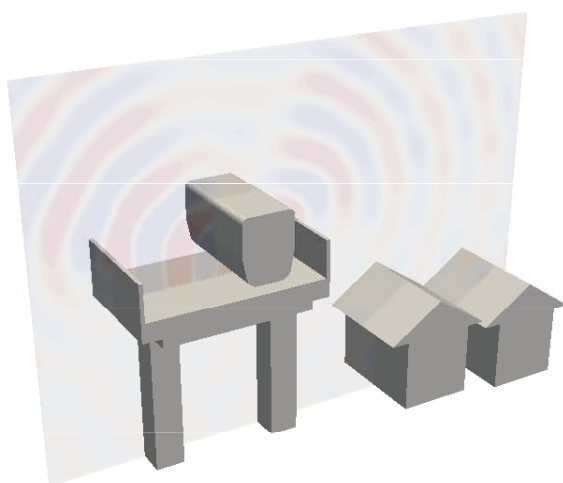
- ① 3次元形状を利用しますが、列車通過時のワーストケースを想定し、本質的に2次元の解析を行います。本解析は、車輪の位置に線音源を仮定したことと同等になります。
- ② 防音壁の吸音率を変化させて、家屋付近の騒音レベルをシミュレーションします。

解析条件(その2)

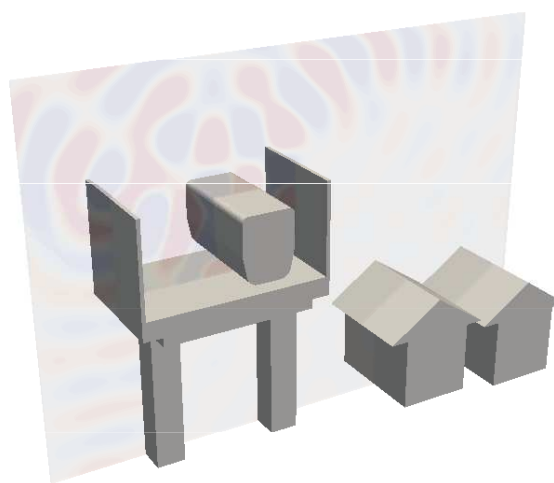
- 高架橋の防音壁の材質を変化させました。→演習Ⅱ-1、3
 - 壁や道床の吸音率(複素インピーダンス)を指定します。
 - 周囲の境界は無反射境界(吸収境界)としました。
 - その他の境界はすべて完全反射です。
- 防音壁の高さを変化させることが可能です。→演習Ⅱ-4
 - 防音壁高さ5mで基本解析を実施します。また、時間に余裕のある方向けに、防音壁高さ2mのCADデータも提供します。まったく同様の操作で手軽に低い防音壁の効果が確認できます。
 - 有限要素法を利用しているため、丸みを帯びた防音壁や斜めの防音壁等に対しても、制約条件なく形状を設定可能です。
- 備考;モデルは実在のものとは異なります。

解析形状

防音壁の低い場合

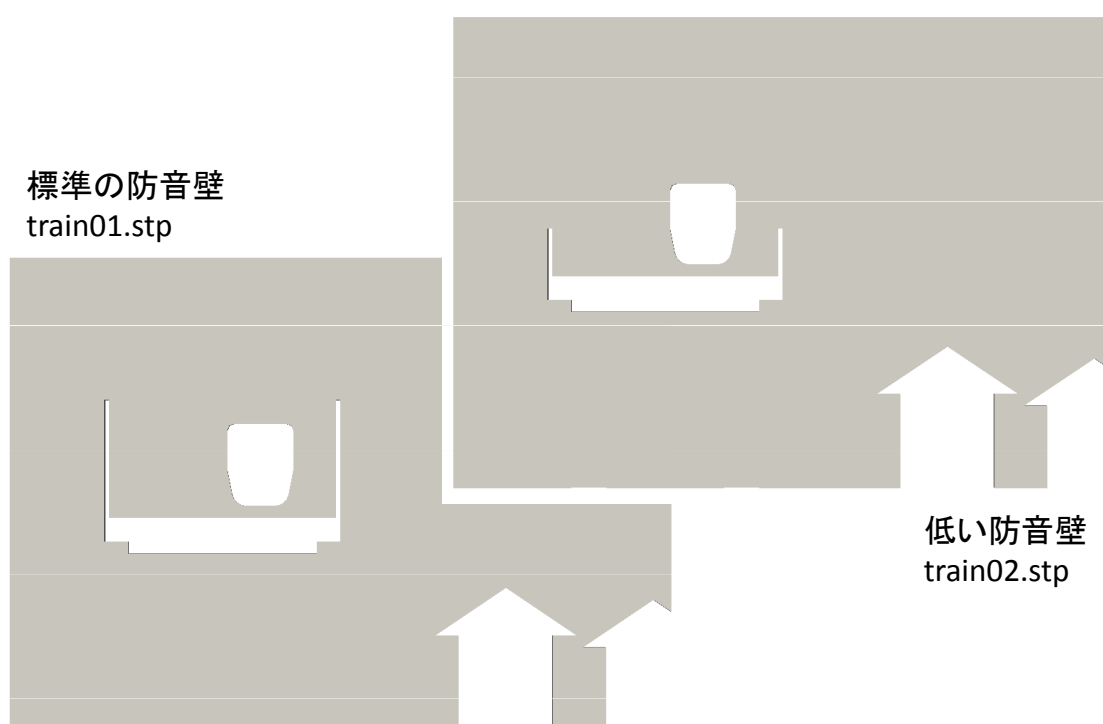


防音壁の高い場合

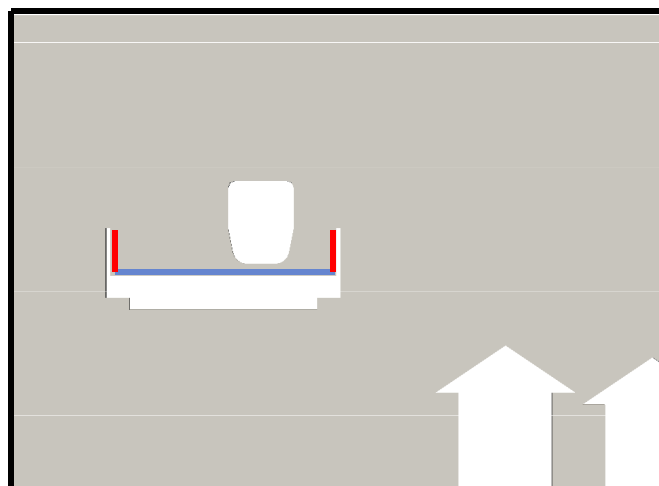





演習 II - 1 ; 基本計算

解析領域 (用意したCADデータ)



境界条件(3つの条件)

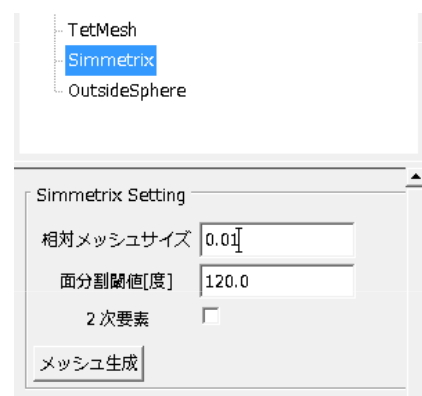
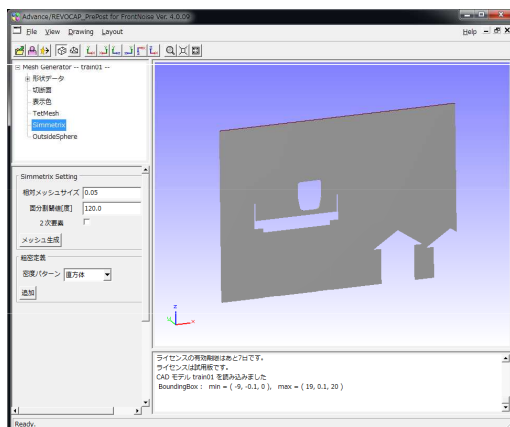


-  ①(音響的な)外部境界
-  ②防音壁側面
-  ③道床

※ 本演習では、その他の境界は完全反射としています

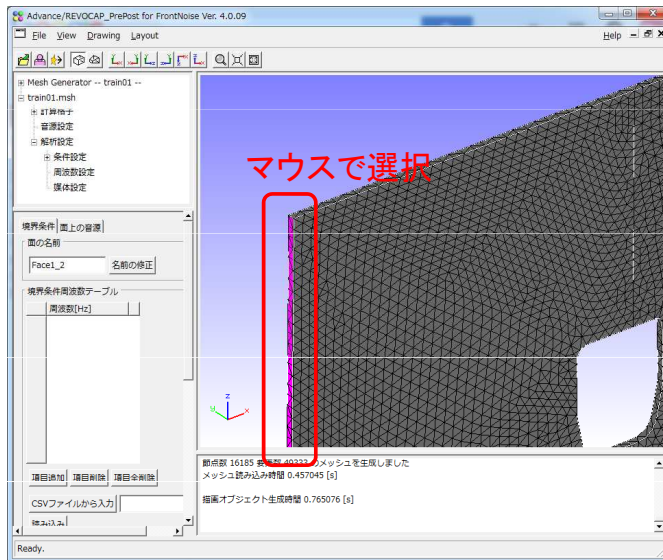
メッシュ作成

- ① ImportCADから wal01.stp を選択して読み込みます。
- ② ツリーメニューから Simmetrix を選んでメッシュを作成します。相対メッシュサイズは0.01とします。



Advance/REVOCAPでの境界条件設定

本解析では、境界条件の周波数特性は makebcv コマンドで作成するため、Advance/REVOCAP ではダミーの境界条件を作成します。メッシュの面分割の関係で複数の面に対して設定する必要があります。



境界条件	面の名前
外部境界	Outer1
	Outer2
	Outer3
	Outer4
防音壁側面	Wall1
	Wall2
道床	Floor1
	Floor2

後の変換の都合上、上記の表の上から順に作成してください

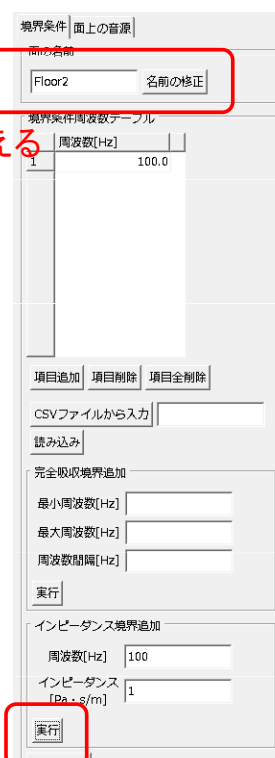
Advance/REVOCAPでの境界条件設定

ここではダミーの境界条件として、適当なインピーダンス境界をそれぞれ追加します。

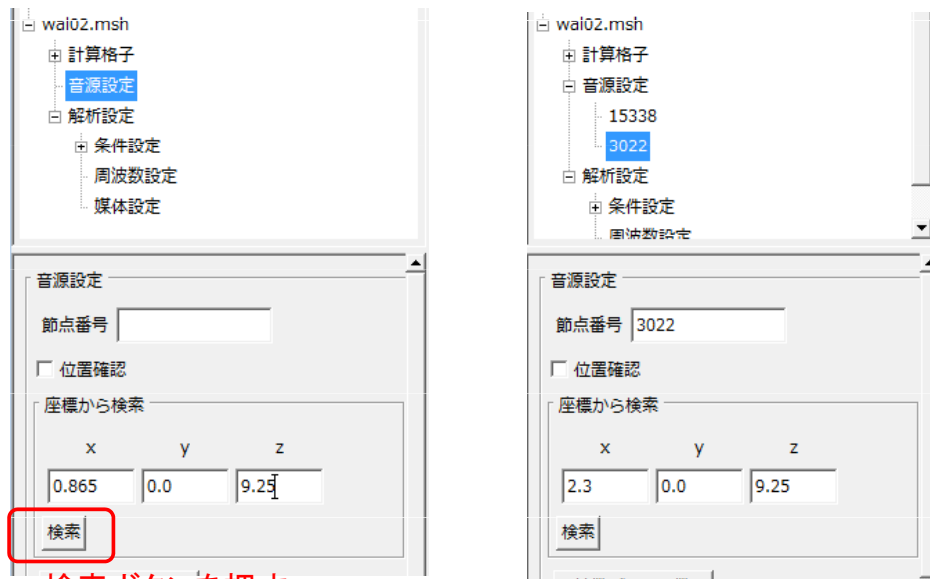
面の名前を変えてから、境界条件を追加します。

面の名前を変える

ダミー境界条件の追加



Advance/REVOCAPの音源設定

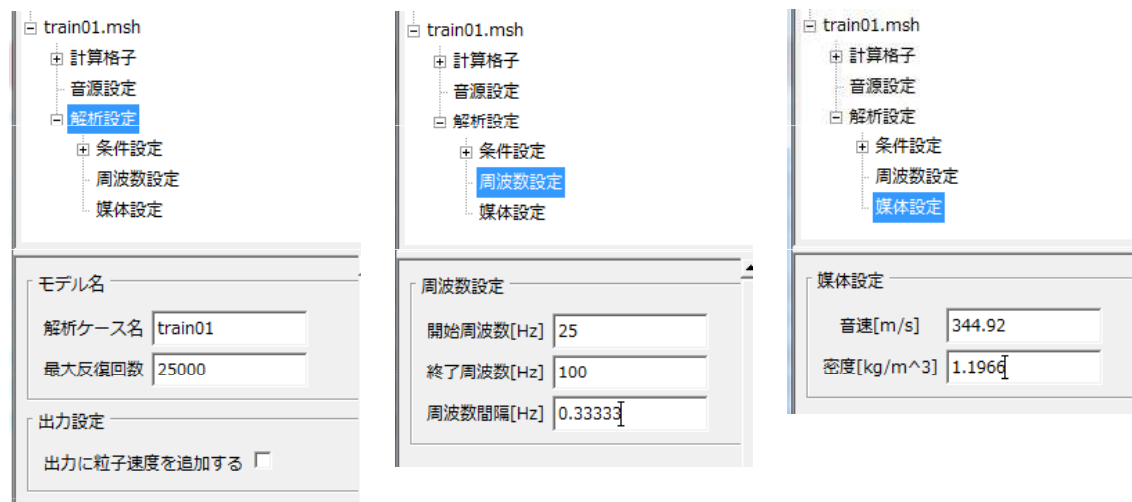


検索ボタンを押す

※体積速度については後で手で設定するのでここでは設定しません。

ツリーの「音源設定」を選択して、節点番号を座標から検索した後に、「追加」ボタンを押します。複数追加する場合は「音源設定」に戻ります。

Advance/REVOCAPの制御データ入力



出力に粒子速度を追加するオプションをオフにしておくことに注意。

境界条件 *.bcvの作成

- 個別に用意した周波数依存のインピーダンスから、境界条件ファイル*.bcvを作成することが可能です。

場所	境界条件	本日用意したデータ
外部境界	吸収境界	100.bcv
防音壁側面	均一の吸音率を指定	100.bcv～100.bcv
道床	均一の吸音率を指定	000.bcv～100.bcv

ファイル	吸音率	ファイル	吸音率	ファイル	吸音率
100.bcv	100%(完全吸収)	060.bcv	60%	020.bcv	20%
090.bcv	90%	050.bcv	50%	010.bcv	10%
080.bcv	80%	040.bcv	40%	000.bcv	0%(完全反射)
070.bcv	70%	030.bcv	30%		

【参考】吸音率 α と反射率 R とインピーダンス Z

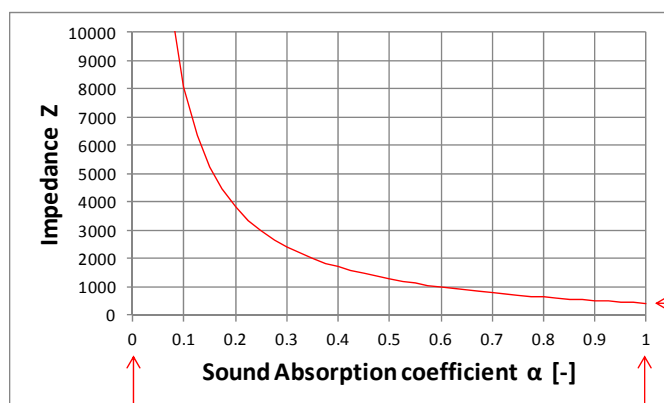
$$\alpha = 1 - R$$

$$R = \frac{Z - \rho c}{Z + \rho c}$$

$$Z = \frac{1 + R}{1 - R} \rho c$$

$$\rho = 1.24 [kg / m^3]$$

$$c = 3.43 \times 10^2 [m / sec]$$



ρc

完全反射

吸収(無反射)

境界条件*.bcvの作成

- REVOCAPで3つの境界(①外部、②壁、③道床)を指定しました。それぞれの境界について、吸音率100%、0%、0%に設定します。前に示した個別のインピーダンスから、bcvを作成する「makebcv」コマンドがあり、これを利用します。

```
$ makebcv 100.bcv 0.bcv 0.bcv train01.bcv
```

上記①②③の順番にインピーダンスファイルを指定し、最後の引数に作成した境界条件ファイル名を指定します。

- REVOCAPで8つの境界条件の面として指定しましたので、実際には下記のコマンドを実行します。

```
$ makebcv 100.bcv 100.bcv 100.bcv 100.bcv 0.bcv 0.bcv 0.bcv 0.bcv train01.bcv
```

- このコマンドは手で実行して、その後に演習 I と同じように実行可能である。また、バッチファイルに上記のコマンドを書くことができる。多くのケースを実行する場合には、後者の方法がより見通しのいい方法になる場合があり、ここではその方法を紹介する。

境界条件作成に関連したファイル

■ 吸音率0%データ 000.bcv

50	0.00	0.00	1.00	0.00	0.00	0.00
100	0.00	0.00	1.00	0.00	0.00	0.00
150	0.00	0.00	1.00	0.00	0.00	0.00
.....						

■ 吸音率100%データ 100.bcv

50	0.00	314.16	-344.92	0.00	0.00	0.00
100	0.00	628.32	-344.92	0.00	0.00	0.00
150	0.00	942.48	-344.92	0.00	0.00	0.00
.....						

■ 前ページのコマンドで作成された境界条件ファイル train01.bcv

5.000000D+01	0.000000D+00	3.141600D+02	-3.449200D+02	0.000000D+00	0.000000D+00	0.000000D+00
5.000000D+01	0.000000D+00	0.000000D+00	1.000000D+00	0.000000D+00	0.000000D+00	0.000000D+00
5.000000D+01	0.000000D+00	0.000000D+00	1.000000D+00	0.000000D+00	0.000000D+00	0.000000D+00
1.000000D+02	0.000000D+00	6.283200D+02	-3.449200D+02	0.000000D+00	0.000000D+00	0.000000D+00
1.000000D+02	0.000000D+00	0.000000D+00	1.000000D+00	0.000000D+00	0.000000D+00	0.000000D+00
1.000000D+02	0.000000D+00	0.000000D+00	1.000000D+00	0.000000D+00	0.000000D+00	0.000000D+00
1.500000D+02	0.000000D+00	9.424800D+02	-3.449200D+02	0.000000D+00	0.000000D+00	0.000000D+00
1.500000D+02	0.000000D+00	0.000000D+00	1.000000D+00	0.000000D+00	0.000000D+00	0.000000D+00
1.500000D+02	0.000000D+00	0.000000D+00	1.000000D+00	0.000000D+00	0.000000D+00	0.000000D+00
.....						

【参考】点音源の設定

- つぎの2点(電車の車輪の位置を想定)を音源とします。この音源を設定する手順を示します。
 - (x,y,z)=(0.865, 0.000, 9.250)
 - (x,y,z)=(2.300, 0.000, 9.250)
- 上記の2点を記述したファイルsource.pntを作成します。

0.865, 0.000, 9.250
2.300, 0.000, 9.250

左右車輪の位置を音源とした

- コマンドnfindの実行(音源の節点番号を求めます)


```
$ nfind train01.inp source.pnt
```
- 音源ファイルの作成(*.bcn, *.bcq)

【確認】音源ファイルの作成(*.bcn, *.bcq)

- 音源の節点番号(*.bcn)
- 音源の値(*.bcq)

22711
30

コマンドnfindの結果表示された節点を指定する

10	1e-1	0
10	1e-1	0
50	2e-2	0
50	2e-2	0
100	1e-2	0
100	1e-2	0
500	2e-3	0
500	2e-3	0
1000	1e-3	0
1000	1e-3	0

実際には、測定結果をFFTしたデータから設定する。

【確認】制御データ *.dat

- 解析周波数は、25Hzから100Hzまでの1/3オクターブバンドを計算するように指定します。
- 本データでは、周波数間隔の1Hz以下は特殊な意味を持っています。Advance/FrontNoiseでは、1/3オクターブバンドの場合には、ここで「0.33333」と指定します。

Model Name	= train01	ケース名
Sound Velocity	= 344.92	音速[m/sec]
Density	= 1.1966	密度[kg/m ³]
Start Frequency	= 25	開始周波数[Hz]
Final Frequency	= 100	終了周波数[Hz]
Frequency Step	= 0.33333	周波数間隔[Hz]
Max Iteration	= 25000	最大反復回数[-]

基本ケース実行のジョブスクリプト

```
#!/bin/bash
#SBATCH -p a024h
#SBATCH -n 1
#SBATCH -N 1
#SBATCH -J ex2
#SBATCH -o %J.stdout
#SBATCH -e %J.stderr

module load PrgEnv-intel
module load impi
export PATH=/home1/glen/share:$PATH
export FNFEM_SOLVER=80
NODEFILE=`generate_pbs_nodefile`

makebcv 100.bcv . . . 000.bcv 000.bcv train01.bcv
mpirun -hostfile $NODEFILE -np $SLURM_NTASKS fnfem < train01.dat
```

スクリプトの計算実行の前に境界条件の作成のコマンドを入れ、実行します。

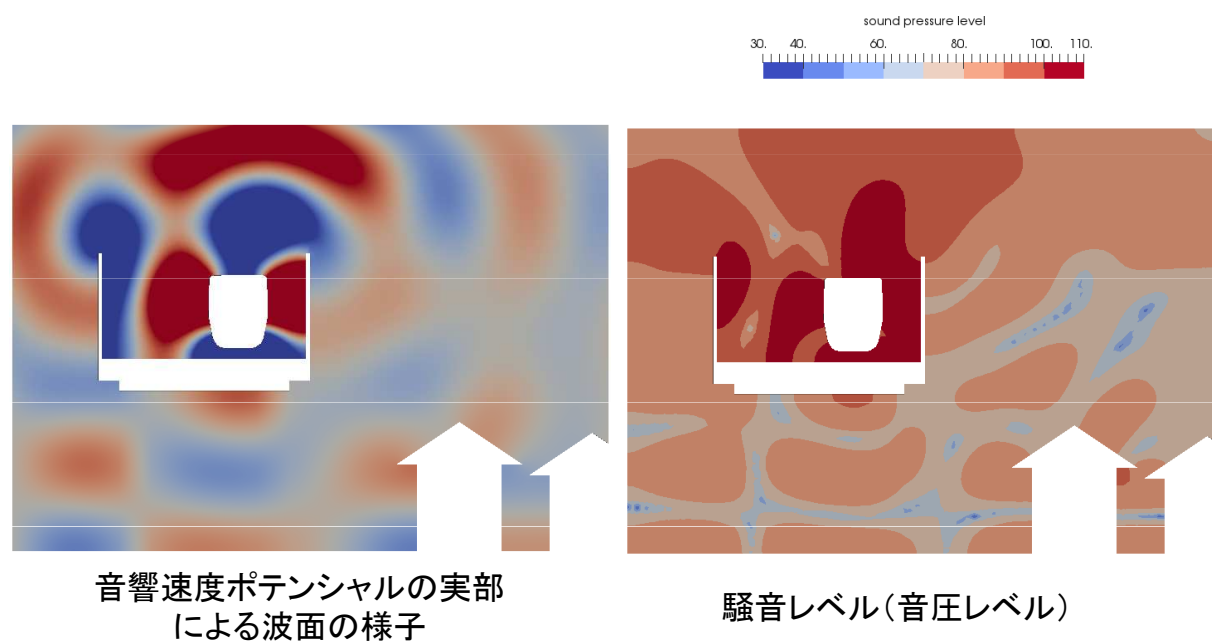
演習 II - 2 ; 可視化

可視化の準備

- 結果ファイルを可視化ファイルに変換します。まず、1/3オクターブバンドで計算されていることを確認し、rsl2inpコマンドで変換します。

```
$ ls *.rsl
train01-r0000-00025.rsl  train01-r0004-00063.rsl
train01-r0001-00031.rsl  train01-r0005-00079.rsl
train01-r0002-00040.rsl  train01-r0006-00100.rsl
train01-r0003-00050.rsl
$ rsl2inp train01.inp train01-r0003-00050.rsl 50.inp
```
- 作成できた*.inpファイルをPCにWinSCPで転送します。
- 結果ファイル(*.rsl)を一括して可視化ファイル(*.inp)に変換する方法はお近くのスタッフにお尋ねください。

標準防音壁、防音壁吸音率0%、50Hz



演習 II - 3 ; 吸音率を設定

吸音率の設定

- 防音壁の吸音率は演習 II - 1 で説明した「makebcv」コマンドを利用します。例えば、つぎのコマンドで防音壁②の吸音率40%の境界条件ファイルtrain01.bcvを作成できます。

```
$ makebcv 100.bcv 100.bcv 100.bcv 100.bcv 40.bcv 40.bcv 0.bcv 0.bcv train01.bcv
```

- コマンドプロンプトで実行することも可能ですが、このコマンドはスクリプトに書くことが便利です。

- 防音壁②の吸音率を40%、道床③の吸音率を60%と指定したい場合にはつぎの通りです。

```
$ makebcv 100.bcv 100.bcv 100.bcv 100.bcv 40.bcv 40.bcv 60.bcv 60.bcv train01.bcv
```

- つぎの頁に、防音壁②の吸音率を20%,40%,60%,80%とするスクリプトの一部を示します。ここでは道床③の吸音率は0%としていますが、これも変更することが可能です。実行は50Hzのみを行います。計算が終了するごとに計算結果を名称の異なる可視化ファイルに変換しています。

連続実行のジョブスクリプト(一部分)

不明な点はお近くのスタッフにお声かけください。

.....

```
makebcv 100.bcv 100.bcv 100.bcv 100.bcv 020.bcv 020.bcv 000.bcv 000.bcv train01.bcv
mpirun -hostfile $NODEFILE -np $SLURM_NTASKS fnfem < train01.dat
rsl2inp train01.inp train01-r0000-00050.rsl train01-020.inp
```

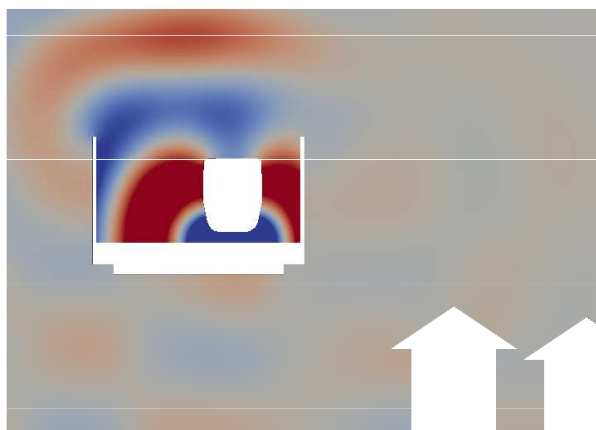
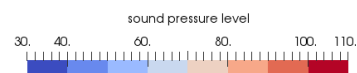
```
makebcv 100.bcv 100.bcv 100.bcv 100.bcv 040.bcv 040.bcv 000.bcv 000.bcv train01.bcv
mpirun -hostfile $NODEFILE -np $SLURM_NTASKS fnfem < train01.dat
rsl2inp train01.inp train01-r0000-00050.rsl train01-040.inp
```

```
makebcv 100.bcv 100.bcv 100.bcv 100.bcv 060.bcv 060.bcv 000.bcv 000.bcv train01.bcv
mpirun -hostfile $NODEFILE -np $SLURM_NTASKS fnfem < train01.dat
rsl2inp train01.inp train01-r0000-00050.rsl train01-060.inp
```

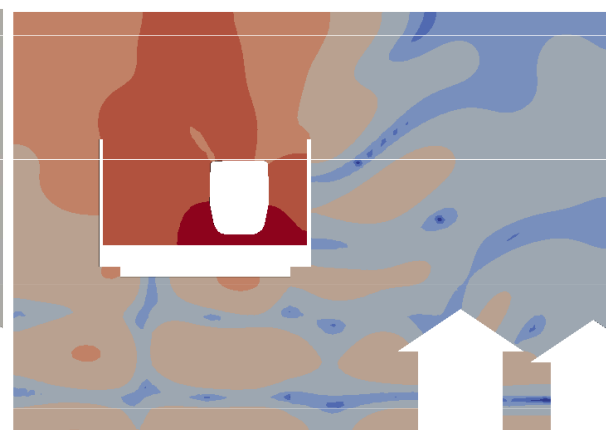
```
makebcv 100.bcv 100.bcv 100.bcv 100.bcv 080.bcv 080.bcv 000.bcv 000.bcv train01.bcv
mpirun -hostfile $NODEFILE -np $SLURM_NTASKS fnfem < train01.dat
rsl2inp train01.inp train01-r0000-00050.rsl train01-080.inp
```

.....

標準防音壁、防音壁吸音率100%、50Hz



音響速度ポテンシャルの実部
による波面の様子



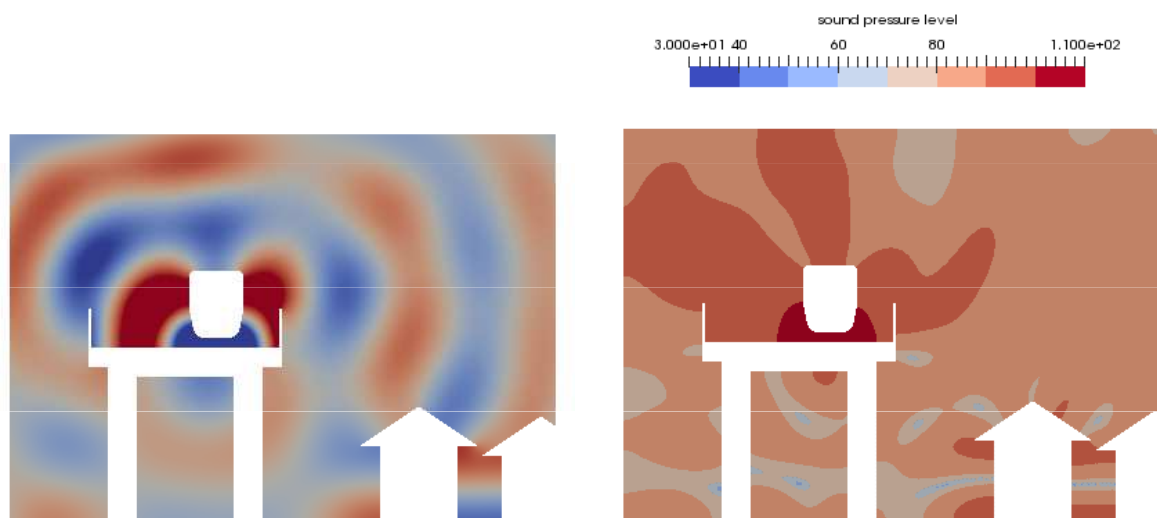
騒音レベル(音圧レベル)

演習Ⅱ-4; 低い防音壁 (時間が余った方はどうぞ)

低い防音壁を計算します

- train02.stpをもとにREVOCAPでメッシュを作成します。
- 境界条件は同じく3つの部分をそれぞれ指定します。
- nfindコマンドを利用して音源を見つけます。これは全く同じ手順です。
- makebcvコマンドで、境界条件ファイルを作成します。バッチファイルに書き込むことを推奨します。
- 例えば50Hzでいくつかの条件で実行します。
- 次ページに、防音壁および道床の吸音率を100%とした50Hzでの計算結果を示します。

低い防音壁、防音壁吸音率100%、50Hz



音響速度ポテンシャルの実部
による波面の様子

騒音レベル(音圧レベル)

演習 II のまとめ

■ 作成したメッシュ(標準防音壁)

項目	内容
要素数	
節点数	

■ 作成したメッシュ(低い防音壁)

項目	内容
要素数	
節点数	

■ メモ

演習	内容
I - 1	完全反射のケースの基本問題
I - 3	_____の吸音率を_____に変更

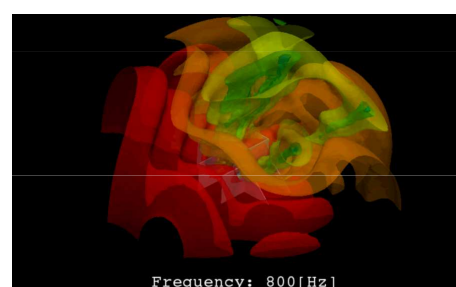
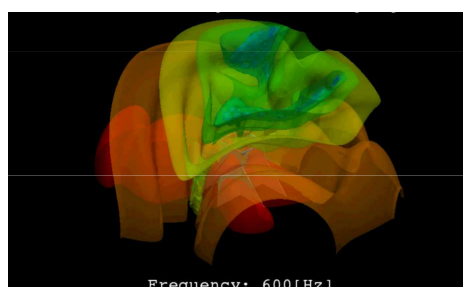
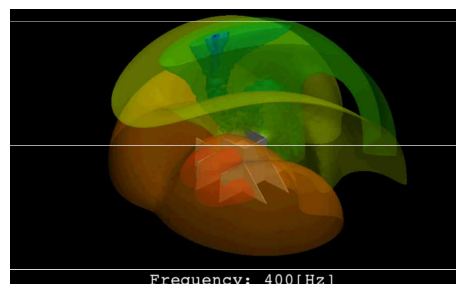
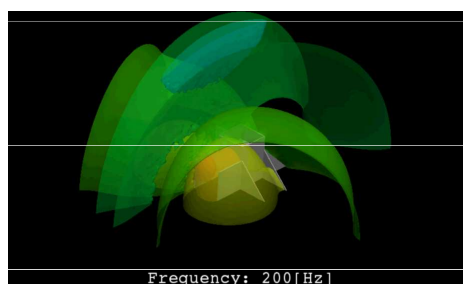
■ 計算の結果

演習	周波数		処理時間	可視化ファイル名・条件等
I - 1	25~100Hz	1/3バンド		
I - 3	50Hz	—		

演習Ⅱのメモ

演習	メモ
Ⅱ-1	
Ⅱ-2	
Ⅱ-3	

【休憩室B】機械部品からの放射音



演習Ⅲ-1;メッシュ細分化 大規模計算の準備体験

演習Ⅲ-2;並列計算実行 FOCUS計算機の「良さ」体験

演習Ⅲ;大規模問題への取り組み

大規模問題の困難さ

- 大規模メッシュ作成の困難さ
 - 数千万を超えるメッシュの作成は、計算機性能や処理時間の面で困難を伴います。→FrontNoiseは「refine:メッシュ細分化」のツールを提供しています。
- 高性能計算機の必要性
 - 高速な並列計算機が必要なこと→近年では、FOCUS等の社外の計算機資源を十分にセキュアな環境で利用可能となりました。
 - 高速な並列計算に対応したソフトウェアが必要なこと→FrontNoiseは並列計算に対応しています。
- 結果可視化可視化の困難さ
 - 結果評価のために大規模計算の可視化が必要なこと→REVOCAPで対応しているとともに、FrontNoiseでは「粗視化」も利用可能です。
- データ転送の困難さ
 - メッシュデータは粗いメッシュを転送した上で、計算サーバー上で「refine」にて大規模メッシュを作成可能です。
 - 大規模な計算結果は、計算サーバー上で「粗視化」等の出力処理をした上で転送可能です。

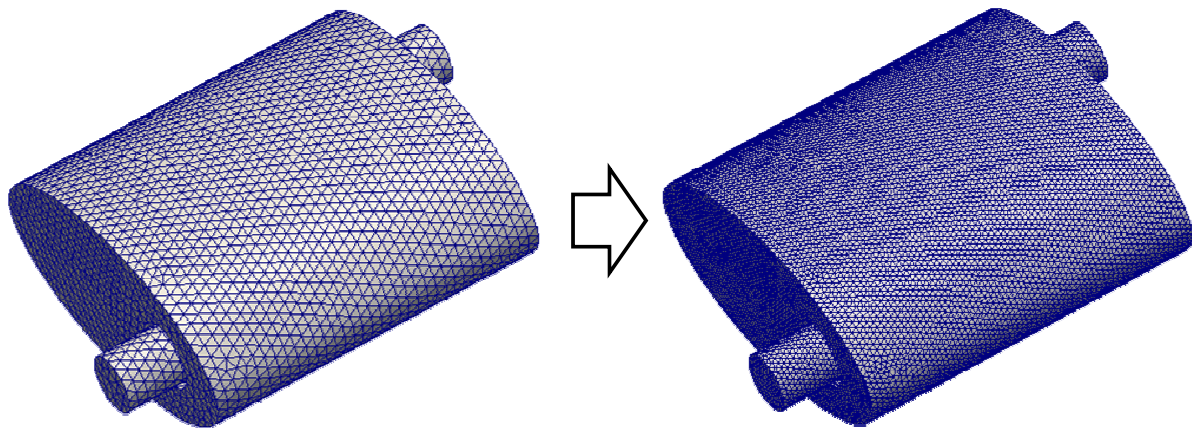
演習 III-1; メッシュの細分化

メッシュの細分化とは？

- **メッシュ細分化の意味**
 - 【高周波の解析を容易に実行する】音響解析では、周波数が高くなるに従いメッシュ幅を小さくしなければならない。メッシュを細分化して高周波での解析を容易に行うことができるツールを提供する。
 - 【結果の確認】解析結果を評価する場合にはメッシュの妥当性が常に問題となる。評価するケースの半分の大きさのメッシュ(メッシュ数は8倍)で同じ結果を得ることでメッシュの問題を解決できる。
- **メッシュ細分化の方法**
 - メッシュャでのメッシュ再作成; 工数がかかるが、自由度大。
 - サイズを半分にするメッシュ細分化; 手軽に実行できるが自由度小。
- **本日のセミナーでは、REVOCAPを利用しない計算サーバ上(FOCUS計算機上)でのメッシュ細分化の方法を示す。**

メッシュ細分化ツール(コマンド: refine)

- 境界条件を含めたメッシュ情報(*.inpと*.bcs)を細分化する。
- ひとつの四面体を8つの四面体にする。
- コマンドrefineに対する入力が元のメッシュの*.inpと*.bcsであり、出力が細分化されたメッシュの*.inpと*.bcsである。



細分化を使ってみる(0.015[m] ÷ 2)

- メッシュ細分化(ここではmuf02.inpとmuf02.bcsを作成)
`$ refine muf01 muf02`
- その他の入力ファイル等を作成する
`$ cp muf01.dat muf02.dat`
`$ cp muf01.bcv muf02.bcv`
- 入力ファイル等を編集する(下記の2つ、詳細は次ページ)
`$ muf02.dat の編集`
`$ go02.sh の編集`
- 実行し、結果からcsvファイルを作る。
`$ sbatch go02.sh`
`$ ls *.rsl | getnvalbin 0 0 0.3 muf02.inp > r1.csv`

入力ファイル等の編集(赤い字の部分)

そろそろ並列計算を始めます。
詳細はつぎの例題で説明します。

- go02.sh
- muf02.dat

```
#!/bin/bash
#SBATCH -p a024h
#SBATCH -n 12
#SBATCH -N 1
#SBATCH -J ex3-1
#SBATCH -o %J.stdout
#SBATCH -e %J.stderr
module load PrgEnv-intel
module load impi
NODEFILE=`generate_pbs_nodefile`
export PATH=/home1/glen/share:$PATH
export FNFEM_SOLVER=80

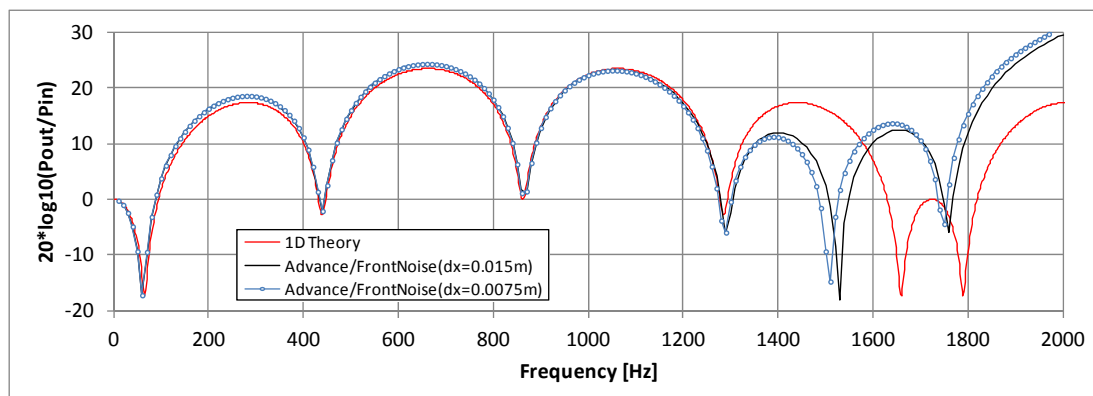
mpirun -hostfile $NODEFILE -np $SLURM_NTASKS fnfem < muf02.dat
```

12並列です

```
Model Name = muf02
Sound Velocity = 344.92
Density = 1.1966
Start Frequency = 10
Final Frequency = 2000
Frequency Step = 10
Output Level = 0
Max Iteration = 25000
Msglevel = 2
END
```

こんな結果が得られましたでしょうか

- 12並列で実行していますが、メッシュが8倍になったので、本データは10分弱の処理時間を要します。
- もとの結果とメッシュ細分化した結果を比較します。全般的に良好な一致をみっていますが、周波数の高い部分で若干のピークシフトがみられます。



演習 Ⅲ－２；並列性能測定

並列化速度測定用の大規模メッシュ作成

- 並列化の速度を計測するために大規模なメッシュを作成します。ここでは、muf01を3回refineしたモデルmuf04を利用します。1回refineのmuf02はさきほど作成しました。
- 細分化機能refineを利用して大規模メッシュを作成

```
$ refine muf02 muf03
$ refine muf03 muf04
```
- その他の入力ファイル等を作成

```
$ cp muf02.dat muf04.dat
$ cp muf02.bcv muf04.bcv
```
- muf04.datとgo04.shを編集
- これで、1000万要素程度の「muf04」のデータができました。

細分化したメッシュ規模をメモしておきましょう

データ名	要素数	節点数	備考(ファイルサイズ等)
muf01			
muf02			
muf03			
muf04			
muf05			

入力ファイル等の編集(赤い字の部分)

本日は時間が足りないため、
1周波数のみを計算します。

• go04.sh

```
#!/bin/bash
#SBATCH -p a024h
#SBATCH -n 12
#SBATCH -N 1
#SBATCH -J ex3-2
#SBATCH -o %J.stdout
#SBATCH -e %J.stderr
module load PrgEnv-intel
module load impi
NODEFILE=`generate_pbs_nodefile`
export PATH=/home1/glen/share:$PATH
export FNFEM_GMRES=300
```

「線形ソルバの選択について」を参照

```
mpirun -hostfile $NODEFILE -np $SLURM_NTASKS fnfem < muf04.dat
```

• muf04.dat

```
Model Name = muf04
Sound Velocity = 344.92
Density = 1.1966
Start Frequency = 1000
Final Frequency = 1000
Frequency Step = 1000
Output Level = 0
Max Iteration = 25000
Msglevel = 2
END
```

バッチファイルでの並列数の指定方法

並列数	ノード数	記述方法	備考
12	1	#SBATCH -n 12 #SBATCH -N 1	本日はこれを基本として計測
24	2	#SBATCH -n 24 #SBATCH -N 2	
48	4	#SBATCH -n 48 #SBATCH -N 4	
96	8	#SBATCH -n 96 #SBATCH -N 8	最低限、このジョブまで実行したい(ほぼ100並列まで実施)
192	16	#SBATCH -n 192 #SBATCH -N 16	
384	32	#SBATCH -n 384 #SBATCH -N 32	これより多いノード数はジョブ投入不可

線形ソルバの選択について

- Advance/FrontNoiseの処理時間は、線形ソルバの解法に要する処理時間が支配的です。解法には、直接法(LU分解)、反復法(GMRES法、MG法等の5種類)があります。
- 手法選択のガイドライン
 - 数十万要素以下;直接法が速い
 - 数十万～数千万要素;経験的に反復法のうちGMRES法が速い。収束が難しい問題は、GMRES法のリスタート数を300程度に指定します。
 - 1億要素以上8億要素以下;経験的に反復法のうちMG法が速い。
- 備考
 - 直接法はメモリを多く使用します。メモリおよび処理時間の制約条件から100万節点程度が限度です。
 - 先ほど実行した演習Ⅲ－1のmuf02ですと、反復法と直接法の処理時間が同じくらいです。

線形ソルバの選択肢

- バッチファイルの中で、環境変数を指定し、各種計算オプションを設定します (export AAA=BBB、使用説明書5.5節p.53)。

手法		指定方法	備考
直接法		FNFEM_SOLVER=80	時間をかければ必ず解ける。
反復法	GMRES法	FNFEM_SOLVER=31	FNREM_SOLVER=31がデフォルト値なので、指定しなくても可能である。
		FNFEM_GMRES=300 (大規模問題で推奨)	デフォルトは30。大きくすれば収束性が上がるが、速度が若干遅くなりメモリを必要とする。
	MG法	FNFEM_SOLVER=36	超大規模問題(数億要素)向けです。

並列実行の処理時間計測(1000Hzのみ)

データ名	並列数	ノード数	反復回数	全体時間	ソルバ時間	ジョブ番号等
muf04	12	1				
muf04	24	2				
muf04	48	4				
muf04	96	8				
muf04	192	16				
muf04	384	32				

その他にもいろいろやってみる

- 演習 I
 - 大規模結果を可視化してみる (rsl2inpを利用してファイルを作成し、PCに転送して可視化)。
 - その他の反復法を利用してみる→5.5節 p.53,54
 - GMRES法の反復法パラメータを変更してみる→5.5節 p.57
 - さらにrefineしてみる→muf05相当は、1周波数あたり8ノード48並列で30分程度の処理時間が必要。メモリが不足するため、1ノードに6プロセスを割り当てること(「-ppn 6」の指定)が必要です。FOCUS計算機の空き具合を見ながら実施可能性を検討します。→お近くのスタッフまで。
- 演習 II
 - その他の点に音源を設定してみる。
 - 道床も吸音率を変化させてみる。
 - 着目以外の壁に新たな境界条件を設定してみる (REVOCAPから変更)

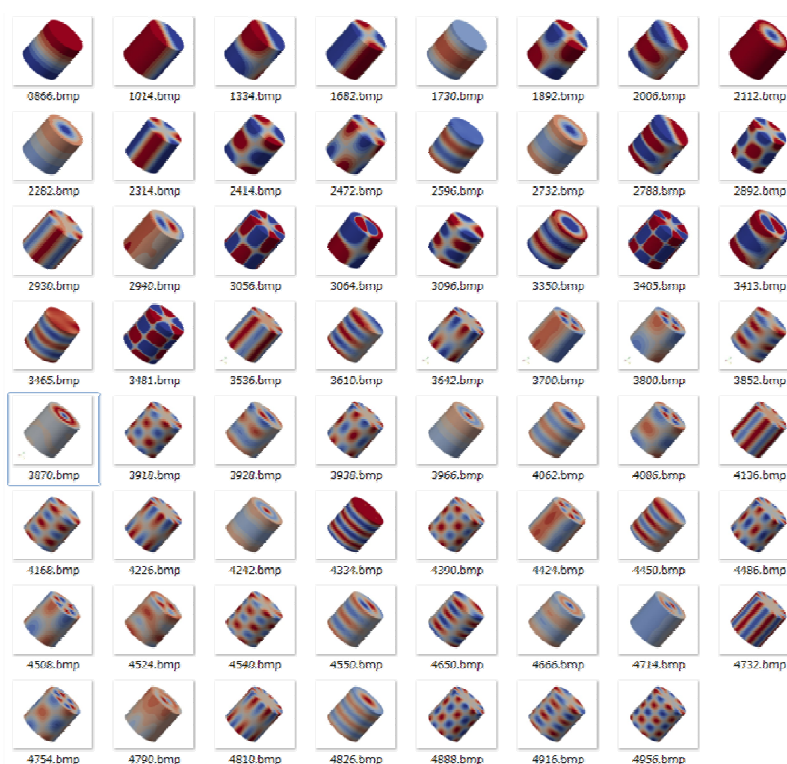
その他の計測パターン(時間余裕ある方)

データ名	並列数	ノード数	反復回数	全体時間	ソルバ時間	ジョブ番号

演習Ⅲのメモ

演習	メモ
Ⅲ-1	
Ⅲ-2	

【休憩室C】共鳴点を求める



演習 IV ; その他

利用可能なツール

分類	ツールの機能	ツール名
入力支援	ABAQUS形式のメッシュからAVS-UCD形式への変換	aba2inp
	境界条件フォーマット変換ツール	bcvconv
	メッシュ細分化ツール	refine
	節点検索ツール	nfind
	要素検索ツール	efind
	リスタートファイル作成ツール	rsl2res
出力支援	計算結果粗視化ツール	coarse
	計算結果ファイルをアスキーUCD形式へ変換するツール	rsl2inp
	結果抽出ツール	getnval
	バイナリ結果編集ツール	getnvalbin
	リスタート機能結果の比較ツール	difrs1

ツールの紹介(その1)

- **【aba2inp】**他の形式のメッシュファイルをFrontNoiseのメッシュ形式に変換することが可能です。
 - ABAQUS形式のメッシュからFrontNoise形式への変換aba2inp
 - FEMAPからexportされたABAQUSファイル
 - PATRANからexportされたABAQUSファイル
 - hypermeshからexportされたABAQUSファイル
 - fv形式のメッシュからFrontNoise形式への変換fv2inp
- **【rsl2rst】**反復法を利用し収束しなかった場合に、その計算結果からリスタートが可能です。
 - 結果を見るとあと少しで収束しそうだった場合等に有効
 - リスタートで反復計算手法等の計算条件を変更することが可能

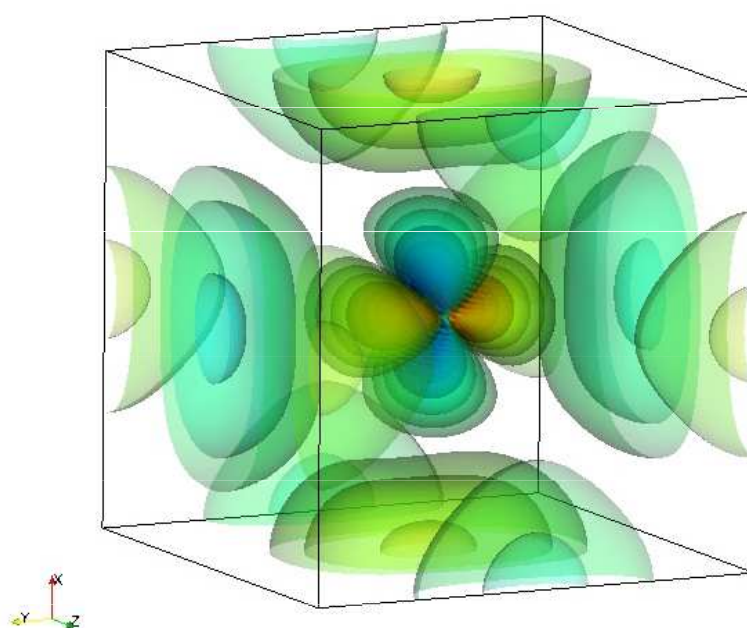
ツールの紹介(その2)

- **【difrsl】**; 同じメッシュを利用した異なる計算結果を比較可能です。相対誤差や絶対誤差等を出力します。
- **【nfind】**と**【efind】**; 座標からメッシュデータを検索します。本日の演習で行ったように音源を設定する場合にnfindが利用できます。また、efindで要素を見つけることも可能です。
- **【inpchk】**; 入力データをチェックすることが可能です。

演習Ⅳのメモ

演習	メモ
IV	

【休憩室D】閉じ込められた四重極音源



追加演習 A

円筒内の共鳴－解析解との比較－

- 円筒内の共鳴のシミュレーションを行い、共鳴周波数とその解析解とを比較する。
- 格子数と精度の関係を定量的に確認する。
- シミュレーション方法
 - 音源を円筒内の偏心した位置に配置する。
 - 節線に合わないよう観測点を設定する。
 - 周波数をスイープさせ、観測点の音圧レベルを求める。
 - 周波数は、10～5kHzで2Hz毎とし、約2500点の解析を行う。
 - メッシュサイズは、最大周波数の波長を1/10したサイズに対して、そのサイズの1/3で精度を確認する。その後、メッシュを粗くして、精度がどの程度得られるかを確認する。
- 理論的に得られる節直径および節半径の数および円筒方向の節線の数を、シミュレーション結果の音圧コンタで確認する。

解析解との比較 共鳴の問題

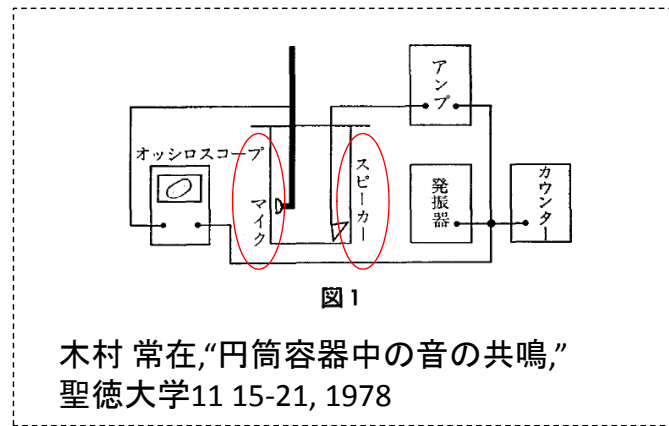
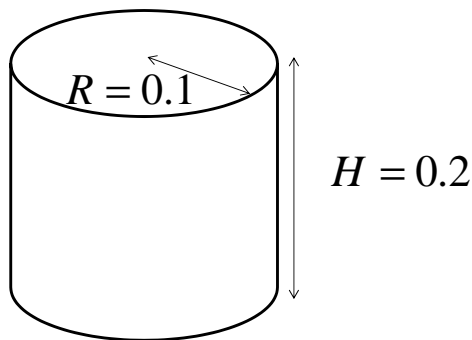
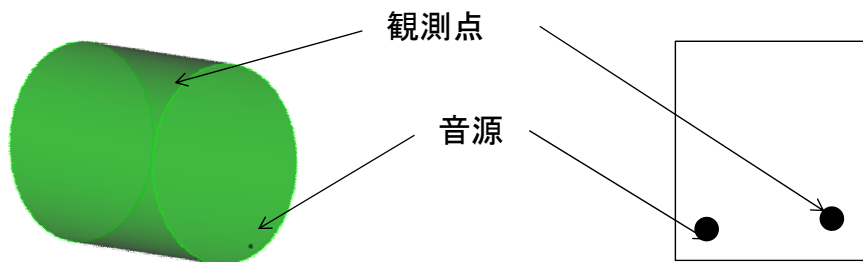


図1

木村 常在, "円筒容器中の音の共鳴," 聖徳大学11 15-21, 1978



方程式と境界条件および解析解

<p>方程式</p> $\Delta\varphi + k^2\varphi = 0 \quad k = \frac{2\pi f}{c}$	<p>境界条件(全面完全反射)</p> $\frac{\partial\varphi}{\partial n} = 0 \quad z = 0, H \quad r = R$
<p>円筒座標系</p>	<p>$J'_m(r)$ の零点</p> <p>$J_{m,1} \quad J_{m,2} \quad J_{m,3} \quad \dots$</p>
$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial\varphi}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2\varphi}{\partial\theta^2} + \frac{\partial^2\varphi}{\partial z^2} + k^2\varphi = 0$	$l^2 = k^2 - n^2 \quad l = \frac{J_{m,s}}{R}$
<p>変数分離後、 Bessel関数の標準型に変換</p>	$rl = r'$
$r'^2 \frac{\partial^2\varphi}{\partial r'^2} + r' \frac{\partial\varphi}{\partial r'} - (r'^2 + m^2)\varphi = 0$	

共鳴周波数の解析解

- q, m, s を任意の正の整数として、共鳴周波数は次の通り。

$$k = \frac{2\pi f}{c}$$

$$f = \frac{c}{2\pi} \sqrt{\left(\frac{J_{m,s}}{R}\right)^2 + \left(\frac{\pi q}{H}\right)^2}$$

$$k^2 = l^2 + n^2$$

$$n = \frac{2\pi}{2H} \times q$$

$$l = \frac{J_{m,s}}{R}$$

すべての共鳴周波数

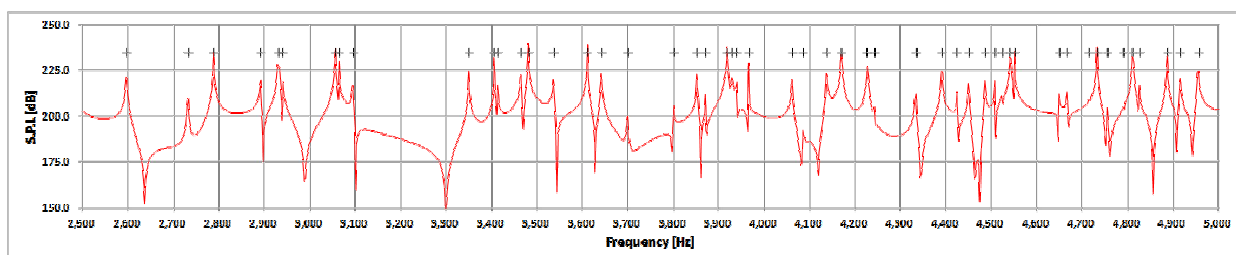
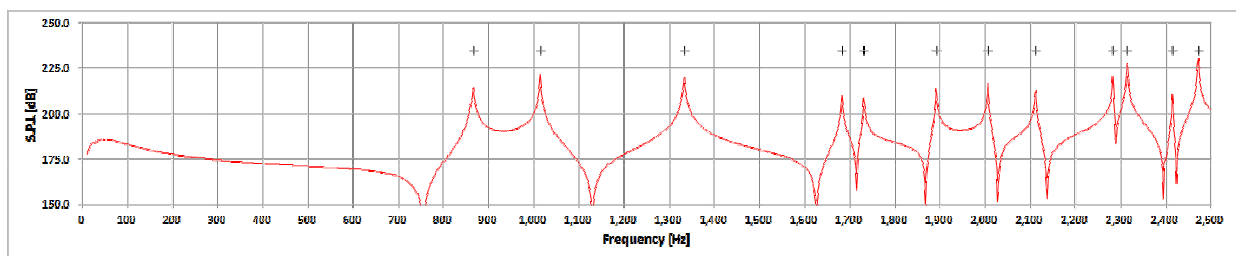
865.00Hz, 1013.90Hz, 1332.75Hz,
1681.87Hz, 1730.00Hz, 1891.28Hz,
2005.22Hz, 2110.03Hz, 2280.45Hz,
2313.50Hz, 2412.80Hz, 2469.92Hz,
2595.00Hz,

ベッセル関数微分の正の零点

	m=0	m=1	m=2	m=3	m=4
s=1	3.832	1.841	3.054	4.201	5.318
s=2	7.016	5.331	6.706	8.015	9.282
s=3	10.174	8.536	9.970	11.346	12.682
s=4	13.324	11.706	13.170	14.586	15.964
s=5	16.471	14.864	16.348	17.789	19.196

解析結果

- 周波数をスイープさせた結果を示す。+印は、理論解を示す。
- 高周波までよく一致していることがわかる。



準備したファイル

- 「円筒のCADデータ」のみ ; FrontNoise/exA/cylinder.stp

(以下の手順で実行して見ていただくことが可能です)

- 解析結果として示したグラフから、0～5kHzの内で本日の時間内に固有値を見つけることができそうな範囲に狙いを定める。
- 周波数に合わせたサイズでメッシュを作成する(REVOCAP利用)
- 境界面はすべて完全反射のため、境界条件は必要ない
- 音源を適当な位置に設定する(コマンドnfind利用)
- 実行する(コマンドnfm)
- コマンドrsl2inpでファイル変換し、REVOCAPで結果を見て、周方向・半径方向・高さ方向のモード数を数える。

追加演習Aのまとめ

■ 作成したメッシュ

項目	内容
要素数	
節点数	

■ 音源

演習	内容
音源位置	(、)
観測点位置	(、)

■ 計算の結果(並列数: 、ノード数:)

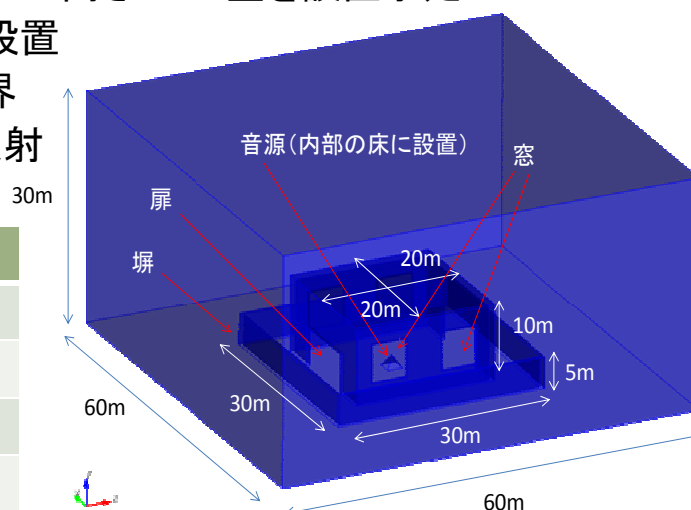
番号	周波数	共鳴周波数	ジョブ番号	計算時間
1	～ Hz Hz毎			
2				
3				

追加演習 B

環境騒音の解析

- 建屋の中の騒音源が敷地境界への影響評価を目的
- 建屋の大きさは、1辺の長さ20m、高さ10mのサイズ
- 4側面に、解放した窓、または、解放した扉を設置
- 建屋の回りに、1辺の長さ30mの高さ5mの壁を設置予定
- 音源は、建屋内部の中心に設置
- 解析範囲の外側には、pc境界
- 床と地面および壁は、完全反射

ケース名	塀	塀吸音材	内壁吸音材	窓
ケースA	NO	NO	NO	普通
ケースB	YES	NO	NO	普通
ケースC	YES	NO	YES	普通
ケースD	YES	NO	NO	小

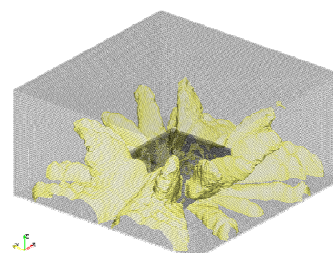
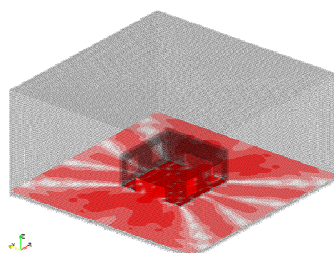


環境騒音の解析(上; 塀なし、下; 塀あり) 50Hz

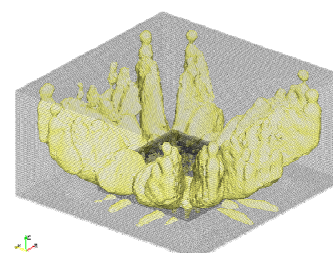
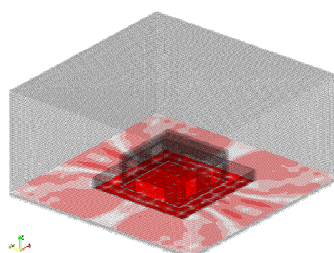
床上1mの騒音レベル

80dBの等値面

ケースA



ケースB

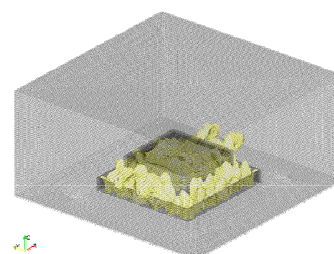
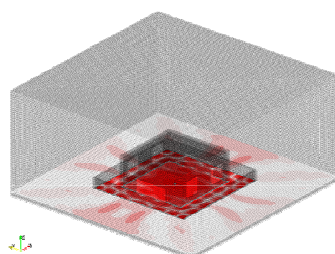


環境騒音の解析(上: 吸音材、下: 窓小) 50Hz

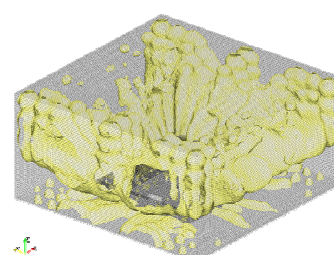
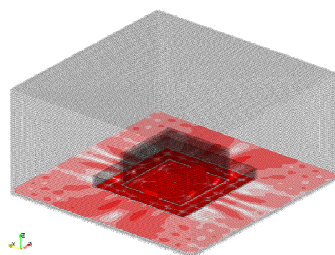
床上1mの騒音レベル

80dBの等値面

ケースC



ケースD



準備したファイル

- 「メッシュファイル」のみ FrontNoise/exB/box02.inp

(解析手順)

- エディタで50Hz用の制御データbox02.datを作成します。
- REVOCAPで境界条件ファイルをbox02.bcs作成します。
- コマンドmakebcvで境界値ファイルbox02.bcvを作成します。
- コマンドnfindで点音源節点データbox02.bcnを作成します
- エディタで点音源データbox02.bcqを作成します
- ジョブスクリプトgo02.shを作成します。
- sbatchで実行します。
- 結果の可視化を行います。

追加演習Bのまとめ

■ 作成したメッシュ

項目	内容
要素数	
節点数	

■ 音源

演習	内容
音源位置	(、)
音源大きさ	

■ 計算の結果(並列数: 、ノード数:)

番号	境界条件	ジョブ番号	計算時間
1			
2			
3			

まとめ

- 消音器の解析と防音壁の解析により、Advance/FrontNoiseを利用した音響解析(内部音響問題および外部音響問題)を体験していただきました。
- 上記演習では、CADデータから出発し、Advance/REVOCAPを利用したメッシュ作成と計算条件の設定、FOCUS計算機でのAdvance/FrontNoiseを利用した音響解析を実施し、その結果をAdvance/REVOCAPで可視化しました。
- FOCUS計算機を利用した大規模並列計算を体験していただきました。そのデータ作成には、Advance/FrontNoiseのメッシュ細分化機能を利用し、効率的に大規模計算を実施しました。

※本日のセミナー内容にご質問等ある場合には、下記営業担当まで連絡いただきたくお願い申し上げます。

お問い合わせ先: アドバンスソフト株式会社 営業部 東田
電話; 03-6826-3971、e-mail; higashida@advancesoft.jp



Advance/FrontNoiseの開発開始から、早10年が経過しました。今後とも、お客様のニーズをもとに開発を進めてまいります。これまでと同様、お客様から意見をいただきながら、本ソフトウェアを成長させ、実績を積み重ねていきたいと考えています(開発チーム一同)。



警告

このレポートに収録されている文章および内容については、ご自身のために役立つ用途に限定して無料配布しています。このレポートを、販売、オークション、その他の目的で利用するには、著作権者の許諾が必要になります。このレポートに含まれている内容を、その一部でも著作権者の許諾なしに、複製、改変、配布を行うことおよびインターネット上で提供する等により、一般へ送ることは法律によって固く禁止されています。