



# EasyISTR 操作マニュアル

Ver 2.22.160910

EasyISTRは、Salome（前処理）、FrontISTR、paraView（後処理）を有機的に結びつけるGUI。

Salomeでメッシュを作成し、EasyISTRで条件設定してFrontISTRを起動して計算。計算結果をparaViewで確認する方法をとっている。

16/09/10 藤井

## 変更経歴

ver-2.14.151030	新規作成
ver-2.15.160205	CLOAD (集中荷重) の境界条件に、等分布荷重の設定を追加。 主応力、主ひずみの大きさとその方向を出力項目に追加。 abaqus メッシュ変換時、一部境界条件の変換を追加
ver-2.20.160405	シェル (731、741) を追加。 paraView 起動方法修正。
ver-2.21.160620	シェル (761、781) を追加して、シェルとソリッドの混在解析が可能。
ver-2.22.160910	FrontISTR_v4.5 (linux、windows) 対応。 FrontISTR、paraView への path の設定を usingApp ファイルで設定。

## 目次

1. EasyISTRとは.....	5
1-1. 概要.....	5
1-2. 起動画面.....	5
2. インストール.....	6
2-1. linux へのインストール.....	6
2-2. windows へのインストール.....	6
2-3. インストール後の設定.....	7
3. 操作方法 (片持ち梁の線形弾性静解析).....	8
3-1. workFolder の設定.....	8
3-2. メッシュ変換 (unv 形式 → fist 形式).....	10
3-3. メッシュ形状の確認.....	12
3-3-1. nodeGroup の表示.....	13
3-3-2. faceGroup の表示.....	14
3-3-3. elementGroup の表示.....	15
3-3-4. nodeNo、elementNo の表示.....	15
3-4. 解析の種類を設定.....	16
3-5. 材料物性値の設定.....	17
3-6. 境界条件の設定.....	19
3-6-1. 固定部の設定.....	19
3-6-2. 節点荷重の設定.....	21
3-6-3. 面圧力の設定.....	22
3-6-4. 境界条件の再確認.....	22
3-6-5. 等分布節点荷重の設定.....	23
3-7. solver の設定.....	23
3-7-1. 線形 solver の設定.....	24
3-7-2. 出力項目の設定.....	24
3-7-3. 計算開始.....	25
3-8. 結果の可視化.....	26
3-8-1. FrontISTR の出力結果を可視化.....	26
3-8-2. 主応力、主ひずみ、mises 応力、mises ひずみを追加する場合.....	30
3-9. 並列処理方法.....	34
3-9-1. 並列処理設定、計算開始.....	34
3-9-2. 並列計算結果の可視化.....	36
3-10. EasyISTR 下部ボタンの動作.....	37
3-10-1. 「folde 開く」ボタンの動作.....	38
3-10-2. 「制御 file 編集」ボタンの動作.....	38
3-10-3. 「meshFile 編集」ボタンの動作.....	39
3-10-4. 「端端起動」ボタンの動作.....	39
3-10-5. 「folder 内クリア」ボタンの動作.....	40
3-11. ファイル変換について.....	40
3-11-1. unv2fistr 変換 (unv 形式のメッシュを FrontISTR 用に変換).....	40
3-11-2. abaqus2fistr 変換 (abaqus 用 inp ファイルを FrontISTR 用に変換).....	41
3-11-3. fistr2vtk 変換 (FrontISTR 用のファイルを paraView 用に変換).....	44
3-11-3-1. メッシュファイルの変換.....	44
3-11-3-2. 結果ファイルの変換.....	50
3-12. salome によるメッシュの作成方法.....	50
3-12-1. ジオメトリの作成.....	50
3-12-2. メッシュの作成.....	53
3-13. 等分布節点荷重の算出方法.....	56
3-13-1. 面の等分布節点荷重の算出.....	56
3-13-2. 線分の等分布節点荷重の算出.....	59
3-13-3. 点の等分布節点荷重の算出.....	59

3-13-4. 等分布トータル荷重の設定結果の確認.....	60
4. 応用事例.....	62
4-1. 接触解析.....	62
4-1-1. モデル形状.....	62
4-1-2. 変位拘束の接触解析.....	63
4-1-3. 荷重（圧力）拘束の接触解析：弱いばね追加.....	71
4-2. 弾塑性解析.....	74
4-2-1. モデル形状（円柱）.....	74
4-2-2. 圧縮 20%の計算.....	74
4-2-3. 引き続き引張 20%の計算.....	79
4-3. 固有値解析.....	81
4-3-1. モデル形状.....	81
4-3-2. 固有値解析の開始.....	82
4-4. 周波数応答解析.....	85
4-4-1. モデル形状.....	85
4-4-2. 周波数応答解析の開始.....	85
4-5. 時刻歴応答解析.....	90
4-5-1. モデル形状.....	90
4-5-2. 境界条件一定値（線形・陽解法）.....	90
4-5-3. 境界条件に時間変化を与える（線形・陽解法）.....	95
4-6. 熱応力解析.....	99
4-6-1. モデル形状.....	99
4-6-2. 熱応力解析の開始.....	100
4-7. 熱伝導解析（静解析）.....	103
4-7-1. モデル形状.....	103
4-7-2. 熱伝導解析（静解析）の開始.....	103
4-7-3. 熱伝導解析結果から熱応力を求める.....	106
4-8. 熱伝導解析（動解析）.....	109
4-8-1. モデル形状.....	109
4-8-2. 熱伝導解析（動解析）の開始.....	109
4-8-3. 熱伝導解析結果から熱応力を求める.....	111
4-9. シェルの解析.....	114
4-9-1. モデルの作成、メッシュ読み込み.....	114
4-9-2. 材料の設定.....	115
4-9-3. 境界条件の設定.....	116
4-9-4. 計算開始、結果の確認.....	116
4-9-5. 結果に主応力等を追加する場合.....	117
4-10. ソリッドとシェルの混在モデル解析.....	119
4-10-1. モデル形状.....	119
4-10-2. メッシュ変換.....	119
4-10-3. 材料、境界条件の設定.....	121
4-10-4. 計算開始、結果の確認.....	122

## 1. EasyISTR とは

### 1-1. 概要

FrontISTR を使って計算する都合上、その前処理（メッシュ作成、境界条件設定等）と後処理（計算結果の可視化）は、必須になる。

前処理、後処理とも、オープンソースを使って処理することを前提に、FrontISTR と結びつける事を考え、EasyISTR を作成している。各々の役割として、以下を考えている。

Salome 等 (オープンソース)	メッシュを作成し、unv 形式で保存する。
EasyISTR	メッシュ変換 (unv 形式 → fist 形式) 境界条件設定 FrontISTR 実行 結果ファイルを変換 (fistr 形式 → vtk 形式)
paraView (オープンソース)	vtk 形式の結果ファイルを読み込み、可視化

上記が実現できるように、EasyISTR を作成している。

### 1-2. 起動画面

EasyISTR を起動すると以下の画面が現れる。この画面上で、操作することになる。



## 2. インストール

EasyISTRは、pythonとpygtkで作成している為、環境さえ整えば、linux、windowsで動作する。linux、windowsとも同じファイル内容で、違いは実行ファイルのみ異なっている。

Linuxの実行ファイル: easyistr  
windowsの実行ファイル: easyistr.bat

### 2-1. linuxへのインストール

インストール用のパッケージを準備しているので、これを使ってインストールできる。パッケージは、Debian用のdebパッケージとRedHat系のrpmパッケージを準備しているので、各々以下の様に端末を起動してコマンドを入力する事でインストールできる。

```
<Debian系>
$ sudo dpkg -i easyistr_2.22.160910_all.deb
```

```
<RedHat系>
$ su
パスワード:
# rpm -i easyistr_2.22.160910-2.noarch.rpm
```

パッケージでインストールすると、EasyISTRは、「/opt/easyIstr」フォルダにインストールされる。実行は、「/opt/easyIstr/easyistr」が実行ファイルになっているので、これを実行する事で、EasyISTRが起動する。また、同時に「/usr/share/applications」フォルダ内に、「easyIstr.desktop」ファイルがインストールされる。ここにアイコンと実行ファイルが登録されているので、ここからでも起動できる。

### 2-2. windowsへのインストール

EasyISTRの実行には、pythonとpygtkが必要なので、これらがインストールされている事が前提になる。EasyISTRは、python2.7とpygtk-2.24.0で作成しているので、以下のバージョンがインストールされている必要がある。

Python: python-2.7.10.msi  
pygtk: pygtk-all-in-one-2.24.0.win32-py2.7.msi

インストールに当たっては、EasyISTRの圧縮ファイル「eastistr-2.22-160910.tar.gz」をインストールしたいフォルダで展開する。

展開後、batファイル「easyistr.bat」の内容を修正する。

修正箇所は、環境変数の「%HOME%」と「%easyIstrPath%」を各々の環境に合わせる。(下記残照。)

```
----- easyistr.bat の内容 -----
:
set HOME=C:%DEXCS%easyIstr           #環境変数「%HOME%」の設定
set easyIstrPath=C:%dexcs2%easyIstr  #インストール先
set easyIstrUserPath=%HOME%easyIstrUser #ユーザ毎の設定データ保存先
:
```

この2ヶの環境変数の設定を修正して、このbatファイルを実行する事で、EasyISTRが起動する。

%HOME% このフォルダ内に%HOME%\easyIstrUser フォルダが作成され、この中に EasyISTR の default の設定ファイルが保存される。  
EasyISTR 起動時に、%HOME%\easyIstrUser フォルダを確認し、無ければフォルダを作成するが、存在していれば何もせずそのまま。

%easyIstrPath% インストール先。ここに PATH を通している。

### 2-3. インストール後の設定

EasyISTR 上から、FontISTR やエディタ、ファイルマネージャ等を起動して、操作を行っている。これらのアプリケーションは、

```
linux      $HOME/easyIstrUser/data/usingApp
windows    %HOME%\easyIstrUser\data\usingApp
```

ファイル内で定義している。

この為、これらの内容を変更する事で、これらアプリケーションを設定、変更する事ができる。

以下が、現在の設定内容になる。

frontIstrFolder は、FrontISTR がインストールされている folder を指定する。EasyISTR は、起動時にその folder に path を通して起動する。

尚、作動がおかしい場合は、エラー内容が easyUser/data/error.log に保存されているので、この内容を確認する。

```
----- usingApp の内容 -----
#
# アプリケーションの起動コマンドを設定
#
# linux 用
linux
  office      loffice --calc
  terminal     gnome-terminal
  fileManager nautilus --new-window
  editor       gedit --standalone
  frontIstrFolder ~/FrontISTR/bin          #FrontISTR への path
  paraView    /opt/paraviewopenfoam50/bin/paraview

# windows 用
# path が通っていない場合は、フルパスで記述
# 空白を含む directory の場合は、「"」で囲む
windows
  office      "C:\Program Files (x86)\LibreOffice 5\program\soffice.exe" -calc
  terminal     start cmd
  fileManager explorer
  editor       "C:\DEXCS\TeraPad\TeraPad"
  frontIstrFolder "C:\DEXCS\FrontISTRv45\FrontISTR_win64"  #FrontISTR への path
  paraView    paraview
-----
```

### 3. 操作方法（片持ち梁の線形弾性静解析）

EasyISTRは、SalomeとparaViewを使って操作する事を前提にしているが、これらは、一般のオープンソースであり、その操作方法は、多数ネット上に存在しているので、ここでは、主にEasyISTRの操作方法について、説明する。

EasyISTRの操作方法に関しては、片持ち梁の線形弾性静解析を例にとって説明する。基本的には、画面上で入力して、最後に「設定」ボタンをクリックして確定させる方法をとっている。この「設定」ボタンのクリックは、その設定内容が制御ファイル等のファイルに書き込まれる事を意味している。

#### 3-1. workFolderの設定

まず、最初にする事は、これから解析するworkFolderを指定するところから始まる。ここでは、「~/CAE/CAE-Fistr/cantilever」フォルダを新たに作成し、ここで解析する事にする。この「cantilever」は、folderのみで、この中にファイルが無い状態から始める。

workFolderの設定は、EasyISTR画面上の「参照...」ボタンで「~/CAE/CAE-Fistr/cantilever」を設定する。今回は、workFolderを設定した段階で、そのfolderが空の為、FrontISTRが必要としている以下の3ヶのファイルをコピーしてくる。

FistrModel.cnt	制御ファイル
FistrModel.msh	メッシュファイル（要素1ヶのメッシュファイル）
hecmw_ctrl.dat	全体制御ファイル





コピーされた3つのファイルは、「~/easyIstrUser」フォルダ内にある3つのファイルをコピーしてくる。  
(「~/easyIstrUser」フォルダは、EasyISTRの初回起動時に作成される。)  
もし、設定したworkFolder内に3つのファイルが存在していれば、ファイルの置き換えはしない。

解析の内容によって、制御ファイルの内容は大きく異なってくるが、制御ファイルのひな形が「FistrModel\_master.cnt」ファイルで、このファイルが「~/easyIstrUser/data」フォルダ内に保存されている。この「FistrModel\_master.cnt」ファイルは、必要最小限の内容の制御ファイルであり、動解析や弾塑性解析の様な複雑な解析では、設定追加の項目が多くなる。  
予め、これら解析専用の制御ファイルを「~/easyIstrUser/data」フォルダ内に別の名前で作成しておくことで、そのファイルを制御ファイルとして設定する事ができ、設定項目を減らす事ができる。

準備した専用の制御ファイルをデフォルトの制御ファイルとして設定する方法は、以下による。(予め、FistrModel\_dynamic.cnt (動解析用) と FistrModel\_plastic.cnt (弾塑性解析用) を準備している事を想定して説明する。)

EasyISTRを起動した段階で、以下の画面のように、予め準備した制御ファイルがリストボックス内に表示されているので、設定したい制御ファイルを選択した上で、「選択>>」ボタンをクリックして、「使用するcntファイル」テキストボックスに移動し、「cntファイル置き換え」ボタンをクリックする。この操作で、デフォルトの制御ファイルが「FistrModel\_plastic.cnt」に設定され、かつ、workFolder内の制御ファイルも置き換わった事になる。

この修正した設定 (デフォルトのcntファイル) は、記憶されるので、次回のeasyISTR起動時にも、その設定が保持される。



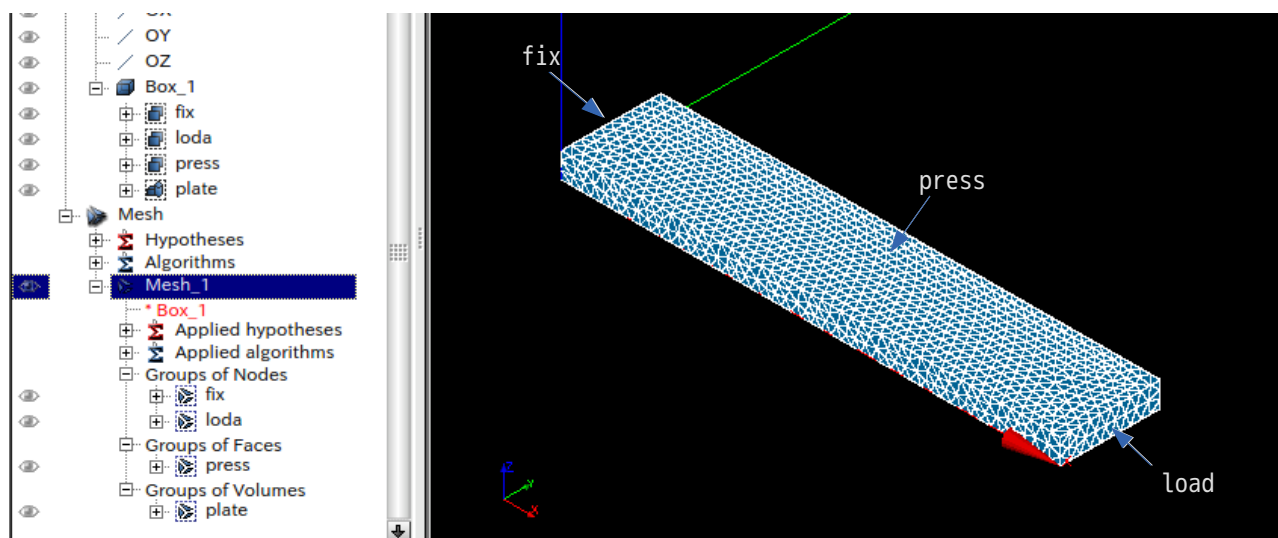
### 3-2. メッシュ変換 (unv 形式 → fist 形式)

salome で作成した unv 形式のメッシュを FrontISTR が取り扱う形式 (fistr 形式) に変換する。

salome で以下のメッシュを作成している。このメッシュの作成方法は、3-12 項を参照。  
 モデルサイズは、100 x 20 x 5 mm で作成している。  
 メッシュサイズは、板厚方向が 2 分割以上になるように、2mm でテトラメッシュを作成している。  
 メッシュのグループは、下図のように、

Node グループ :	fix、load	固定部と荷重印加部
Face グループ :	press	圧力印加部
volume グループ :	plate	モデル全体

を作成している。



このメッシュを「./plate.unv」として workFolder 内に保存しておき、この保存した unv メッシュを FrontISTR が取り扱える fistr 形式に変換する。

メッシュ変換の為に、EasyISTR の設定項目 Tree 上の「FistrModel.msh」を選択する。(下図参照。) メッシュ変換枠内の「unv2fistr」が選択されていることを確認の上、「参照...」ボタンで plate.unv ファイルを指定する。

「ファイル変換」ボタンをクリックして、unv 形式から fistr 形式に変換する。

以上の操作で、ファイル変換ができたことになる。

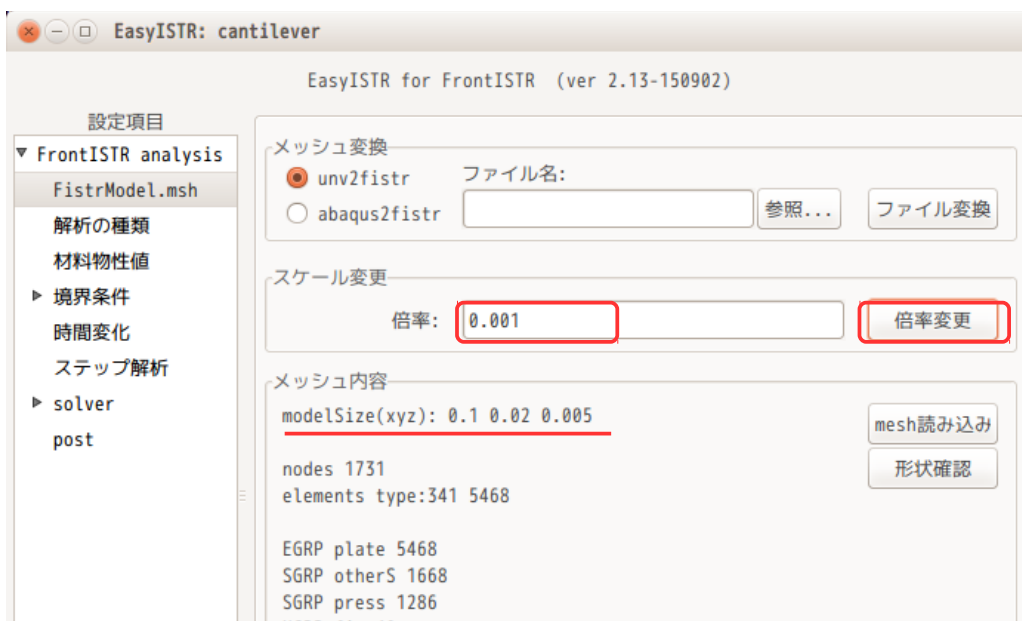
メッシュ変換後、EasyISTR のメッシュ内容枠内に、メッシュの概要が以下の様に表示されている。

modelSize(xyz): 100.0 20.0 5.0	モデルの大きさ
nodes 1731	節点数
elements type:341 5648	要素タイプ (341:四面体) と要素数
EGRP plate 5468	要素グループ plate の要素数
SGRP otherS 1668	面グループ otherS の面数
SGRP press 1286	面グループ press の面数
NGRP fix 40	節点グループ fix の節点数
NGRP load	節点グループ load の節点数



メッシュの概要から、モデルサイズが 100 x 20 x 5 であり、mm 単位で作成されている事が判る。この為、1/1000 に縮小する必要がある。

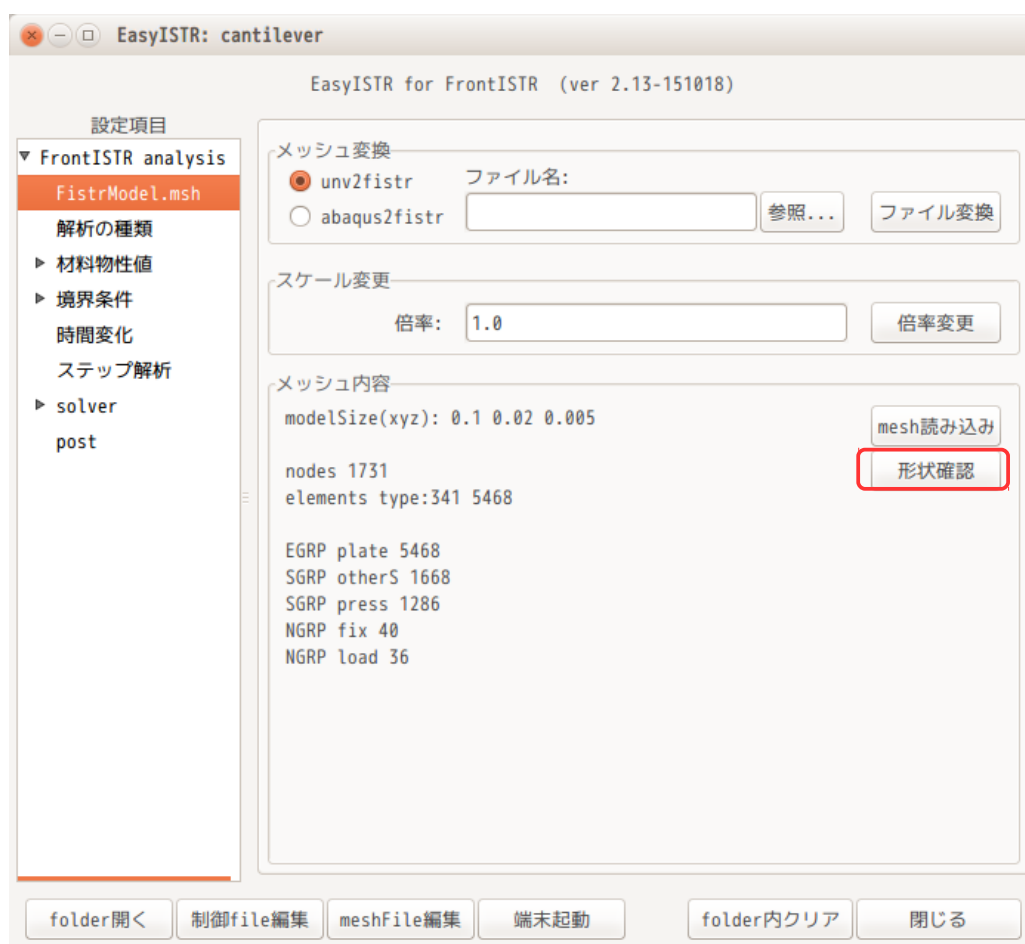
モデルのスケールを変更するためには、下図のようにスケール変更枠内の倍率を「0.001」に設定し、「倍率変更」ボタンをクリックする事で実現できる。この操作により、modelSize(xyz)の値が 1/1000 の値に修正されている。



### 3-3. メッシュ形状の確認

EasyISTR が認識したメッシュの情報（要素種類、要素数、節点数、グループ名等）は、EasyISTR 画面上で確認できる。また、その形状を paraView を起動して確認することができる。

paraView による形状の確認方法は、EasyISTR 画面上の「形状確認」ボタンをクリックする事で、paraView が起動し、メッシュ形状や face、node のグループの形状が確認できる。



paraView が起動すると、既に読み込むべきファイル「convFistrModel.msh.vtk」を読み込んだ状態で起動しているので、paraView 上の「Apply」ボタンをクリックする事で、画面上にその形状が現れる。

FrontISTR のメッシュファイル「FistrModel.msh」は、そのままでは paraView で読み込む事ができない為、「形状確認」ボタンをクリックした段階で、メッシュファイルを vtk 変換し「convFistrModel.msh.vtk」ファイルを作り出している。

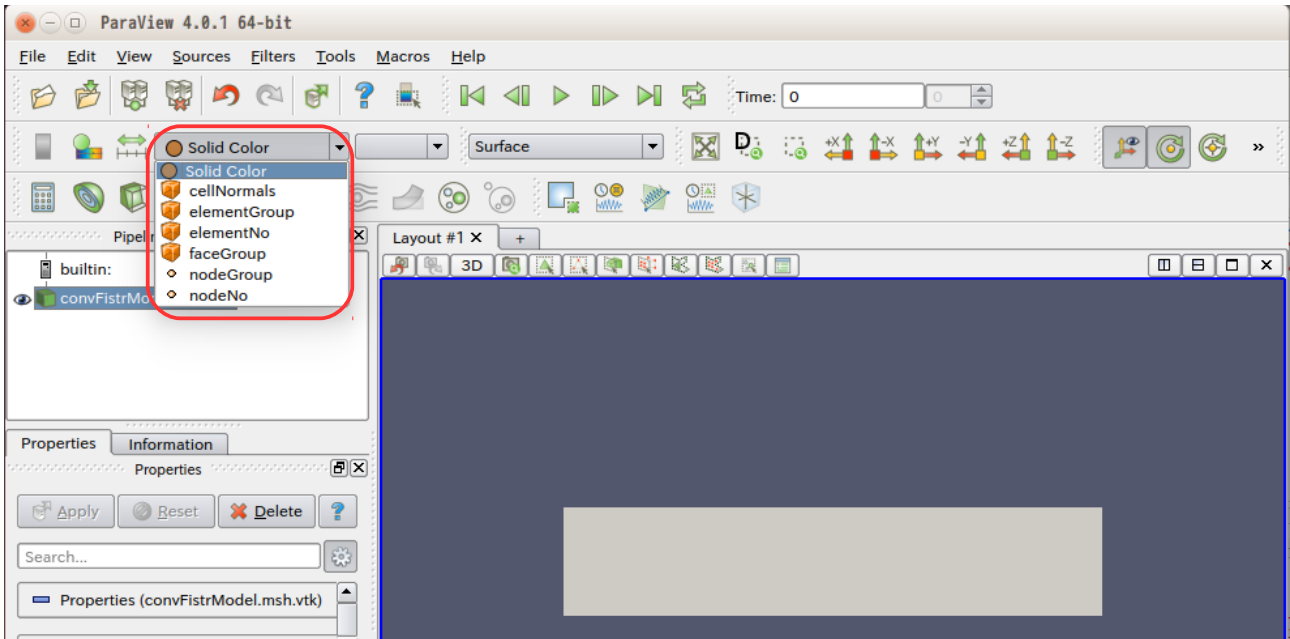
メッシュデータの vtk 変換は、メッシュ側で定義している要素、面、節点グループが表示できるように、volume 要素、モデル表面の face 要素、節点を vtk 形式で作成して、これを paraView が表示している。

下図は、paraView の「Apply」ボタンをクリックして、メッシュを読み込んだ状態。paraView で表示できる項目は、下図の様に表示項目内から選択できる。

これらの表示項目は、

elementGroup	要素グループ
elementNo	要素 No
faceGroup	面グループ

nodeGroup                    節点グループ  
 nodeNo                      節点 No  
 が確認でき、これが表示できる。



### 3-3-1. nodeGroup の表示

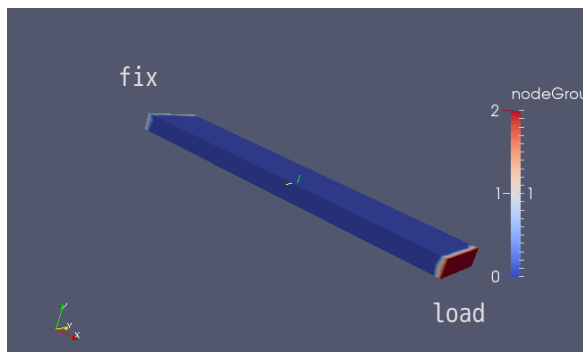
nodeGroup は、EasyISTR の画面上から、定義順に以下の nodeGroup が確認できる。

NGRP    fix  
 NGRP    load

メッシュデータの vtk 変換時に、各節点に以下の groupNo を付加している。

グループ名	nodeNo
fix	1
load	2
以外	0

paraView で表示項目として nodeGroup を選択すると、この値を表示する事ができ、各 nodeGroup の位置と形状が確認できる。（下図参照）



### 3-3-2. faceGroup の表示

faceGroup は、EasyISTR 画面上から、以下が確認できる。

SGRP	otherS	未定義の face (press 以外の外表面)
SGRP	press	

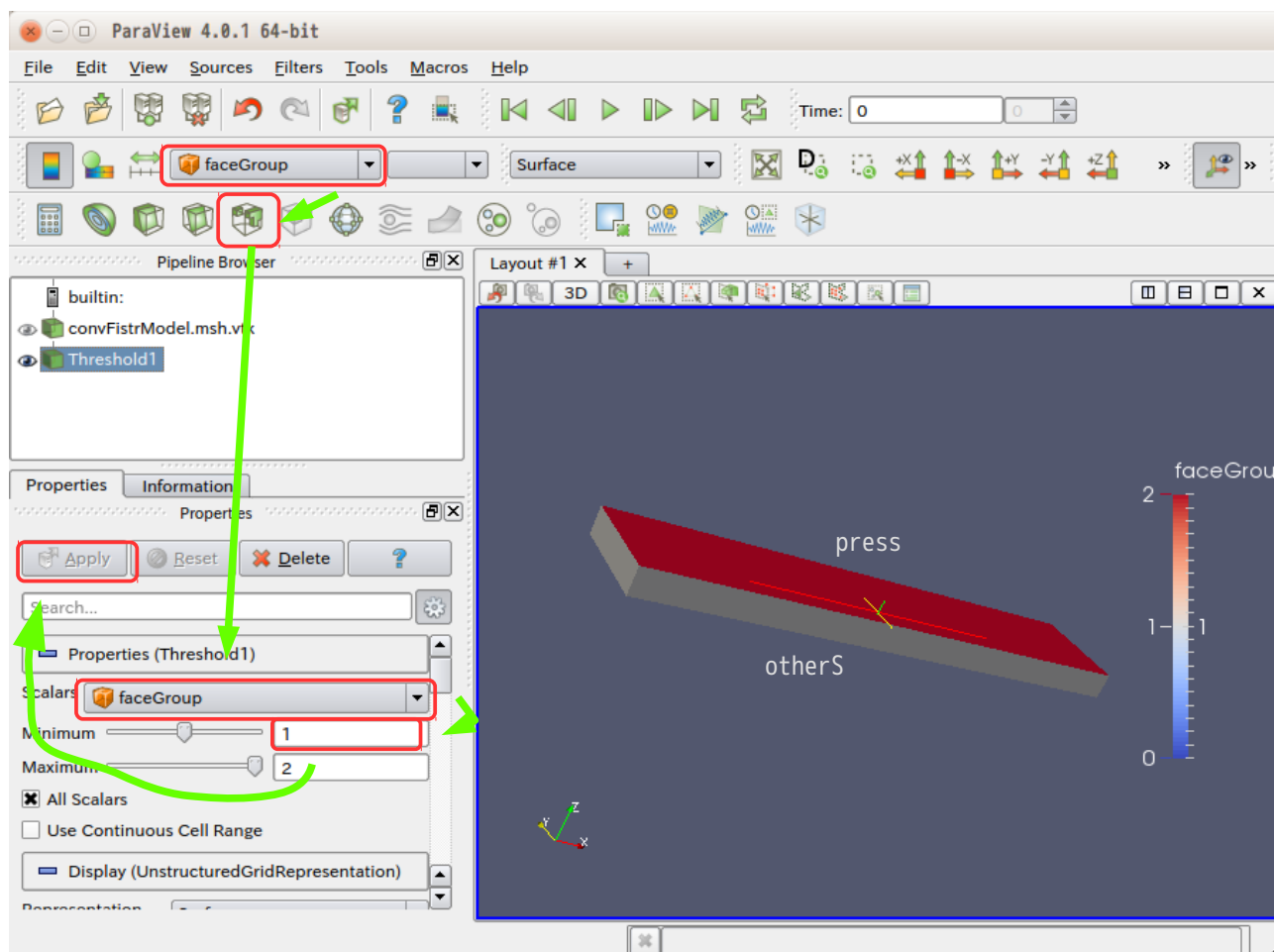
メッシュデータを vtk 変換した後は、要素 (volume 要素と face 要素) と節点の情報がある。faceGroup を表示するためには、この要素を表示する必要がある。face (外表面) を表示させる為には、volume 要素が邪魔になるので、volume 要素には、faceNo を「0」に設定している。

また、存在している faceGroup の「otherS」と「press」面については、定義順に faceNo を付加している。従って、faceNo は、以下の様に付加している事になる。

グループ名	faceNo
volume 要素	0
otherS	1
press	2

paraView で表示項目として「faceGroup」を選択する。

この後、volume 要素 (faceNo:0) が邪魔になるので、faceNo:0 を隠す操作が必要になる。これを隠す為に、引き続き、「Threshold」ボタンをクリックして、Scalars から「faceGroup」を選択し、minimum を「1」に設定し (これで、faceNo の 1~2 が表示される事になる。)、 「Apply」ボタンをクリックする事で、faceNo が「1~2」が表示される。下図が表示した状態になる。



### 3-3-3. elementGroup の表示

elementGroup は、easyISTR 画面上から、「plate」が確認できる。

今回は、全要素が plate として設定されている。もし、一部のみ定義してメッシュ変換 (unv2fistr を実行) すると、「other」と言う elementGroup が作成される。今回は、全てが plate で定義されている為、other と言う elementGroup は、存在していない。

また、elementGroup を未定義でメッシュ変換すると、変換するファイル名が elementGroup 名として定義される。

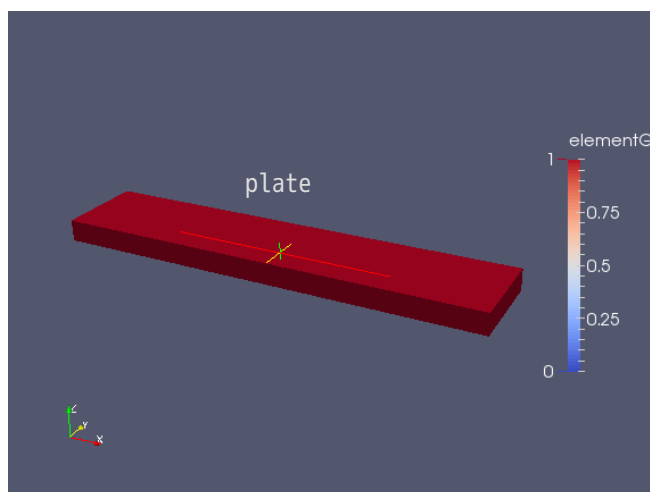
この elementGroup を paraView で表示させる為には、要素 (volume 要素と face 要素) を表示させれば良いが、前項とは反対に face 要素が邪魔になる為、face 要素には elementNo に「0」を与えている。この為、今回のモデルの elementNo は、以下の状態になる。

グループ名	elementNo
face 要素	0
plate	1

vtk 変換時の要素は、volume 要素を作成した後に face 要素を作成している。(volume 要素の方が外側にできあがっている。) この為、paraView で通常通り、elementGroup を選択して表示させても、volume 要素しか見えないので、正常に elementGroup が確認できる。

ただし、断面を取るなど複雑な操作を行うと、face 要素が見える事があるので、この場合は、前項と同様な操作を行って、elementNo の「0」を隠す操作を行う。

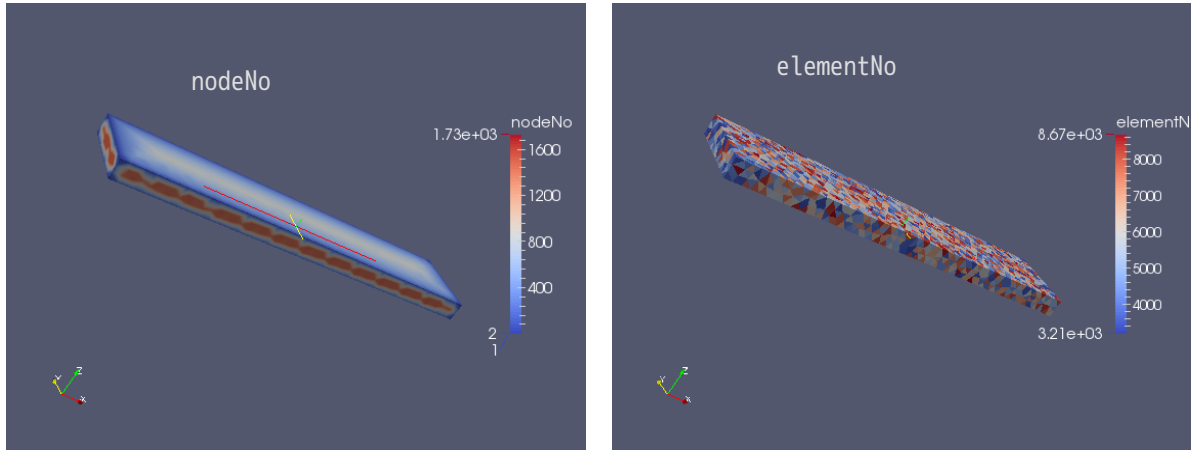
下図が今回の elementGroup 形状になる。



### 3-3-4. nodeNo、elementNo の表示

メッシュに振られた節点番号、要素番号の値がどのように振られているかが確認できる。

paraView 上の表示項目から nodeNo、elementNo を選択することで、確認できる。今回の場合、下図の様に番号が振られている。節点番号は、規則性があるが、要素番号は、規則性無くランダムに振られている様子。また、特定の節点、要素を選択することで、選択した節点 No や要素 No を paraView 上で知ることができる。

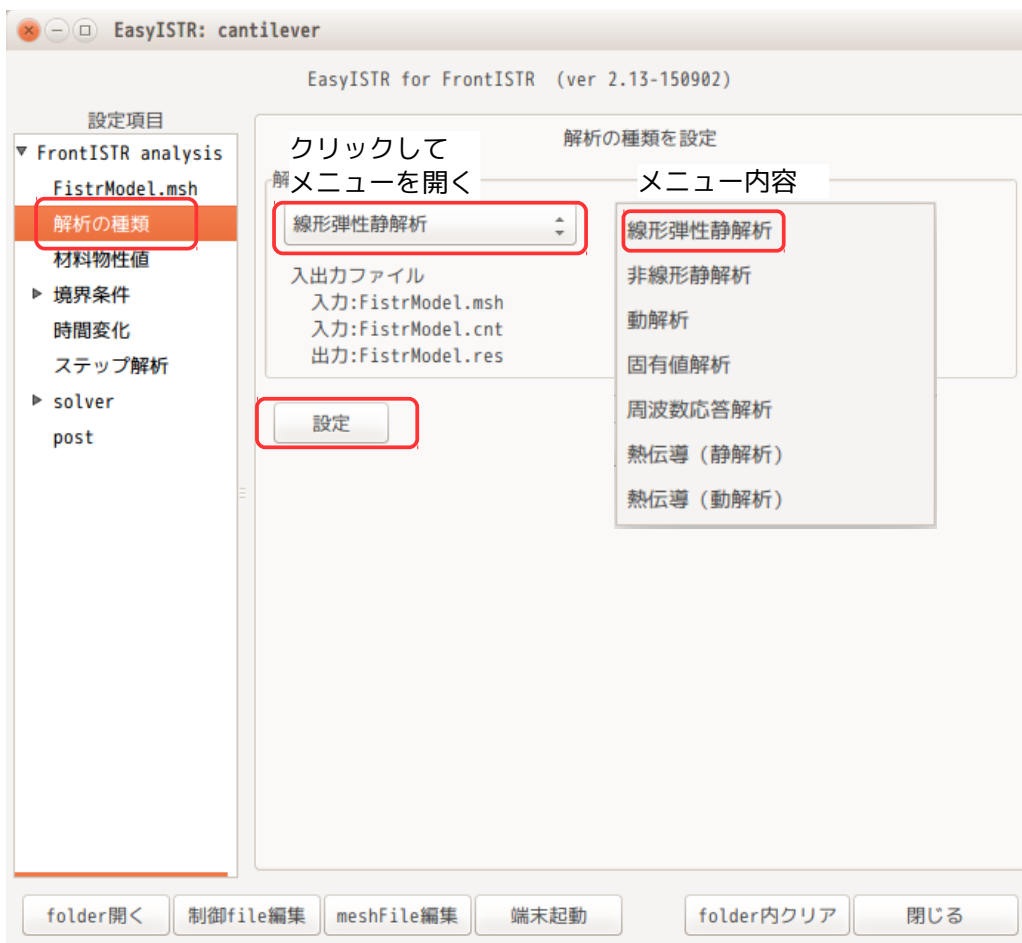


### 3-4. 解析の種類を設定

ここで解析の種類を設定する。今回は、線形弾性静解析を行う為、これに設定する。

設定方法は、EasyISTR 画面上の設定項目 Tree 内から「解析の種類」を選択する。

この後、ドロップダウンメニューから「線形弾性静解析」を選択し、「設定」ボタンをクリックして、解析の種類を設定する。





この操作（「設定」ボタンのクリック）により、cnt ファイルが修正され解析の種類が確定したことになる。

### 3-5. 材料物性値の設定

ここで、モデルの材料物性値を設定する。

EasyISTR 画面の設定項目 Tree 内から「材料物性値」を選択する。これにより、材料物性値の設定画面が変わる。

まず、材料 DB を確認する。デフォルトの設定は、「~/easyIstr/data/mat.csv」で設定されている。これは、EasyISTR インストール後の初回起動時に、このフォルダ内に材料 DB が作成され、これがデフォルトの設定になっている。

独自の材料 DB を作成する場合は、適当なフォルダに、同じフォーマットで csv ファイルを作成して、これを使うこともできる。この場合は、「参照...」ボタンで使用する DB を指定する。一度 DB の場所を変更すると、変更内容が記憶されるので、次回 EasyISTR 起動時にも変更した DB が表示される。

材料 DB を設定した後は、モデルに材料物性値を設定する。

材料の設定は、メッシュの EGRP に設定する。今回のメッシュ内には、EGRP として、plate を定義しているので、この plate に材料を定義することになる。

EasyISTR 画面上には、現在の EGRP 名のリストが表示されている。今回は、「plate」に材料を設定する為、plate を選択して、「選択>>」ボタンをクリックして、材料を定義する group 側に移動する。



「選択>>」ボタンをクリックした状態が以下になる。この状態で、「設定」ボタンをクリックして、確定す

ることになる。



以上の操作で、EasyISTR 画面の設定項目 Tree の材料物性値の下に「plate」が追加される。



設定項目 Tree 内の「plate」を選択して、これに材料物性値（材料 DB 内の材料名）を設定する。plate を選択後、材料名のドロップダウンメニューから設定する材料名を選択する。ここでは、「Steel」を選択した。このドロップダウンメニュー内の材料名は、設定されている材料 DB を読み込んで、その材料名を表示している。



下図が、Steel を選択後の画面になる。この後、「設定」ボタンをクリックして、材料名が確定する。



設定した「Steel」の物性値がどのような値になっているかは、材料DBの設定内容による。この値を確認するためには、「材料DBを開く」ボタンをクリックする事で、libreOfficeが開き、その内容が確認できる。

### 3-6. 境界条件の設定

境界条件は、plate 端面 (fix) を固定し、plate 先端 (load) に節点荷重、plate 上面 (press) に圧力をかけてみる。

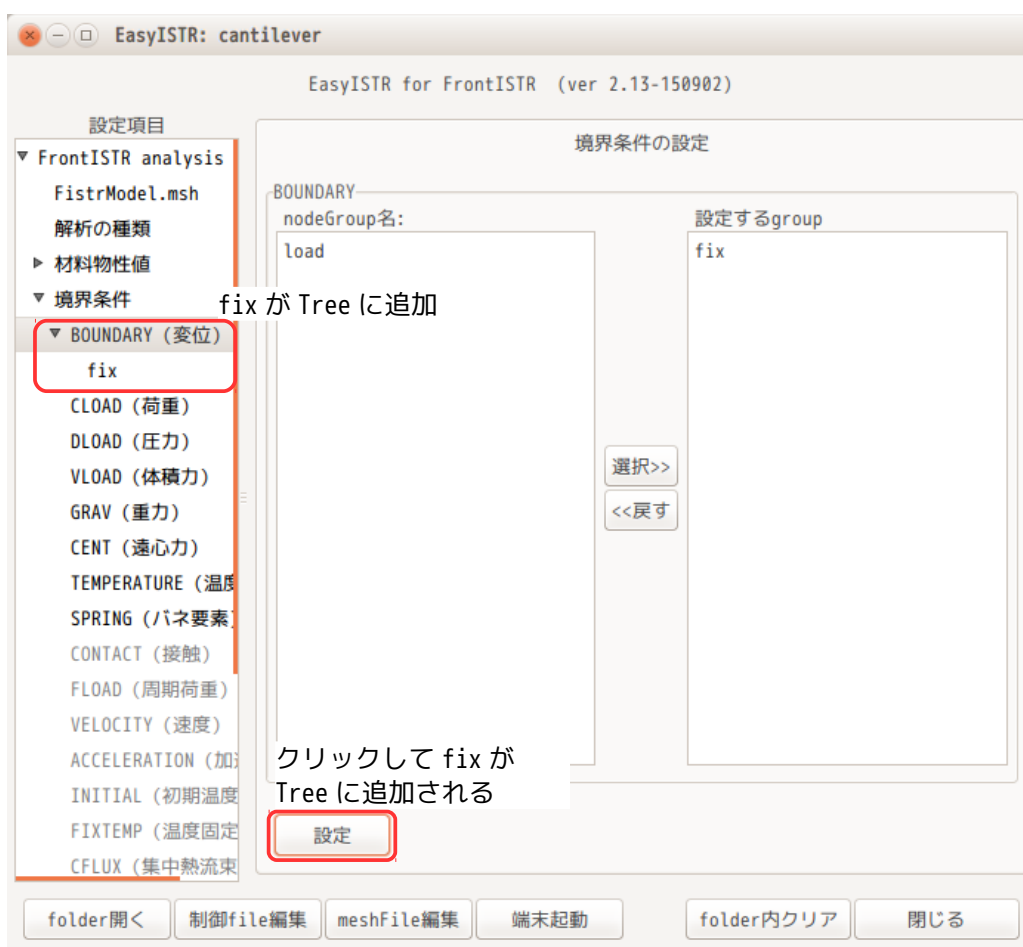
#### 3-6-1. 固定部の設定

固定の境界条件は、各方向の変位を「0」設定すれば、固定できる。この為、設定項目 Tree の「境界条件」>「BOUNDARY (変位)」を選択する。

この後、設定する NGRP fix を選択し、「選択>>」ボタンをクリックして、設定する group 側に移動する。



fixを設定するgroup側に移動させた後、「設定」ボタンをクリックして、設定するgroup名を確定させる。確定させると、下図のように設定項目Treeの「BOUNDARY (変位)」の下に「fix」が追加される。尚、追加した「fix」を削除する場合は、「fix」を選択して「<<戻す」ボタンをクリックし、「設定」ボタンをクリックする事で削除する事ができる。



この後、fixに境界条件を設定する事になる。

設定項目 Tree 内の「fix」を選択し、各方向の変位を「0」に設定し、「設定」ボタンをクリックして、値を確定させる。（下図参照）



### 3-6-2. 節点荷重の設定

節点荷重は、NGRP loadに設定する。

設定項目 Tree の「境界条件」>「CLOAD (荷重)」を選択する。

この画面上で、前項と同様に、NGRP loadを選択し、「選択>>」ボタンをクリックして、設定する group 側に移動する。

移動後、「設定」ボタンをクリックして、loadを確定させる。

確定させると、設定項目 Tree 内の「CLOAD (荷重)」の下に「load」が追加される。

この後、設定項目 Tree 内の「load」を選択して、「セットする集中荷重の種類」を選択後、各方向に印加する荷重を設定する。今回は、「節点当たりの荷重」を選択し、Z方向に「-100N」を設定した。

設定後、「設定」ボタンをクリックして、設定値を確定させる。

尚、「セットする集中荷重の種類」の内容については、3-6-5項を参照。



### 3-6-3. 面圧力の設定

面圧力は、SGRP press に設定する。

設定項目 Tree の「DLOAD (圧力)」を選択する。

設定する group 名の「press」を選択し、「選択>>」ボタンをクリックして、設定する group 側に移動する。

移動後、「設定」ボタンをクリックして、設定を確定させる。

これにより、設定項目 Tree の「DLOAD (圧力)」の下に「press」が追加される。

設定項目 Tree 内の「press」を選択して、圧力の値を入力する。今回は、1000000Pa を入力した。値を入力後、「設定」ボタンをクリックして、値を確定させる。(下図参照)



### 3-6-4. 境界条件の再確認

設定した境界条件を確認するためには、設定項目 Tree 内の group 名 (fix, load, press) を選択することで、選択した group に設定されている内容を確認する事ができる。

下図は、load の境界条件を確認した結果になる。



### 3-6-5. 等分布節点荷重の設定

各節点に集中荷重を設定する方法は、3-6-2項の方法で設定するが、この方法では、各節点に同じ値（今回の例では、z方向に-1000N）が設定されてしまう。実際は、各要素の face 面の面積や節点の密度が不揃いの場合が多く、境界条件を設定する面に対して、均等に荷重が設定されていない事になる。また、面に対して合計でどの程度の荷重が掛かっているのかは、この設定した荷重に節点数を乗じる必要があり、判り難い状態になっている。

この為、この集中荷重が、境界条件を設定する面に対して均等に分布（等分布荷重）する様な方法を準備している。その入力方法は、合計の荷重を入力して、その合計荷重を面に対して均等に分布させる方法をとっている。

今回の例では、下図の様に「等分布トータル荷重」を選択し、z方向に「-3600」を入力する事で「load」面に均等に分布する集中荷重（合計が-3600）を設定する事ができる。

尚、荷重の値を「-3600」とした理由は、load面の節点数が36ヶの為、3-6-2項と同じトータル荷重とする為に、「-3600」を入力している。この節点数は、3-3項の「NGRP load 36」で確認できる。

この等分布荷重は、四面体、六面体、五面体の1次2次要素に対して設定できる。これらの詳細は、3-13項を参照。



尚、セットする集中荷重の中で、「トータル荷重」は、トータル荷重を入力する事で、「トータル荷重/節点数」の値を全節点に設定するので、等分布荷重にはなっていない。（荷重の分布は、「節点当たりの荷重」と同じ分布になる。）

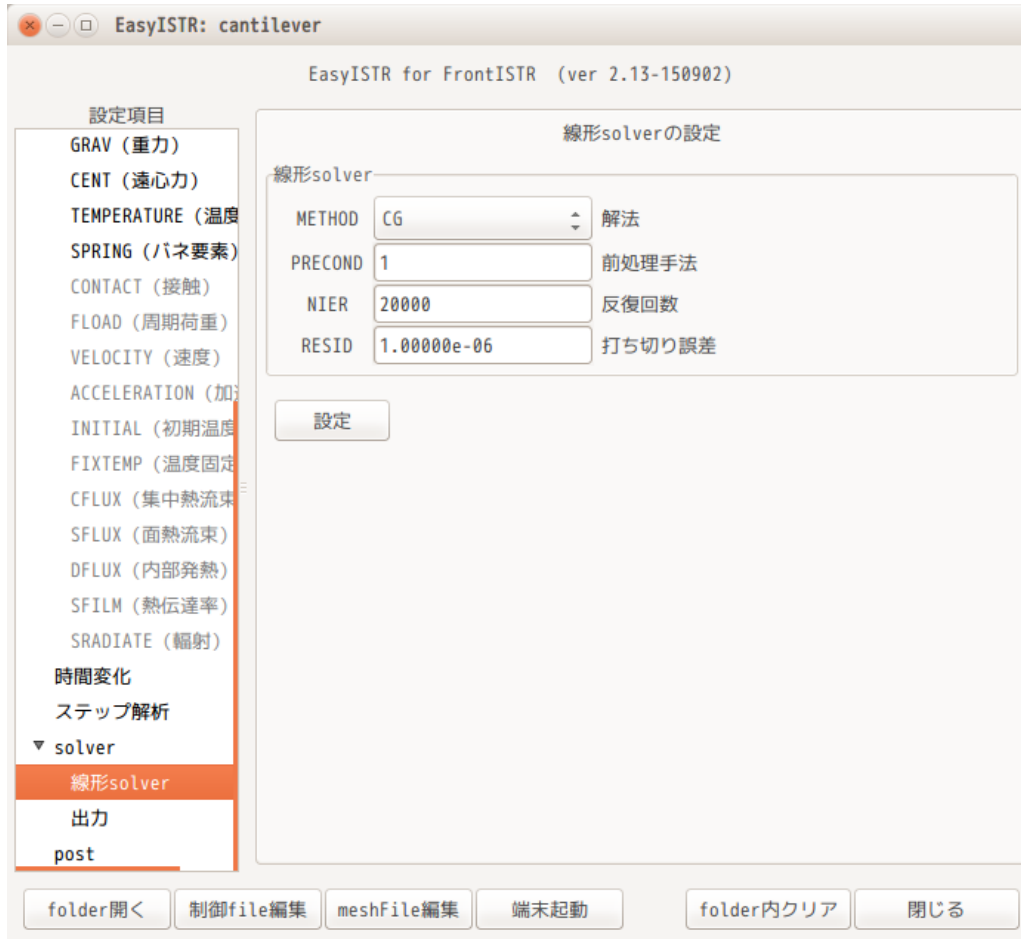
### 3-7. solver の設定

ここで、計算方法の設定を行う。

### 3-7-1. 線形 solver の設定

設定項目 Tree 内の「線形 solver」を選択する。下図が選択した画面になるが、デフォルトの設定のままで問題ない為、そのまま。

もし、収束しないようであれば、解法や反復回数を変更する。

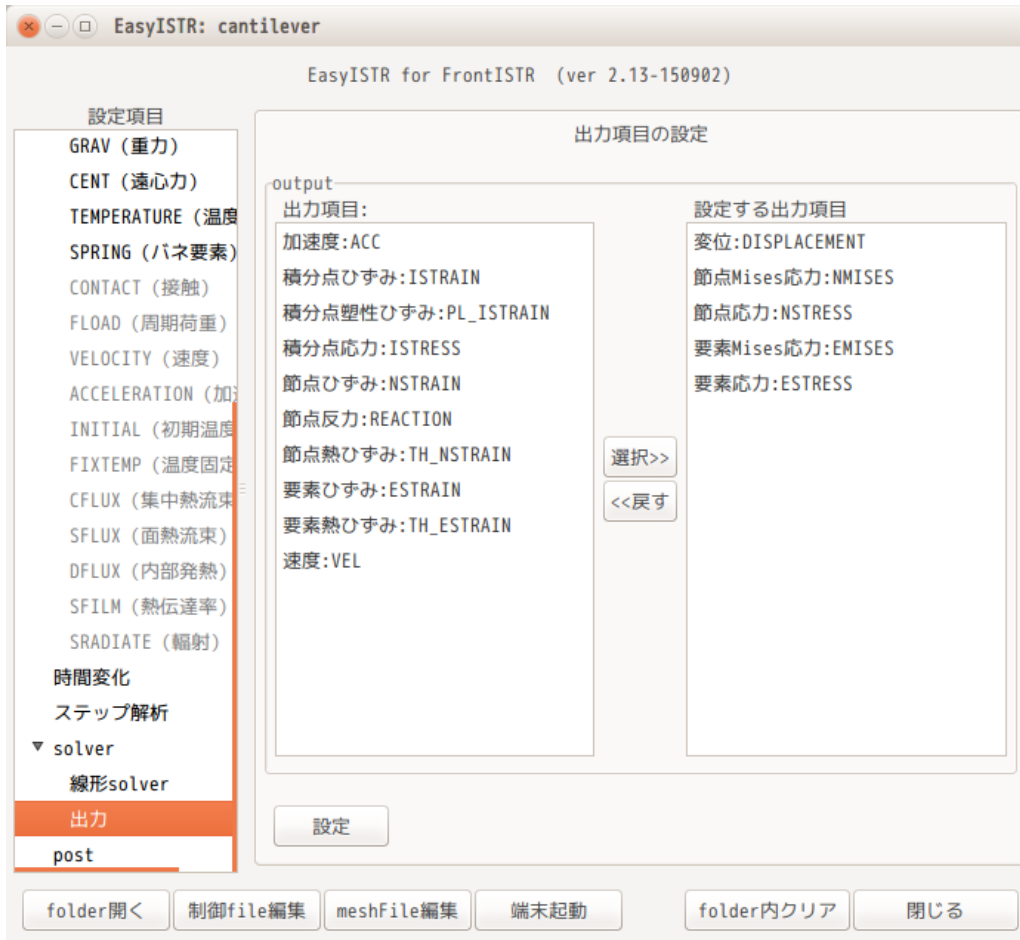


### 3-7-2. 出力項目の設定

計算結果の出力項目を確認する。

設定項目 Tree 内の「出力」を選択すると、現在設定されている出力項目が表示される。出力項目を追加、削除する場合は、「選択>>」ボタンで追加、「<<戻す」ボタンで削除ができる。

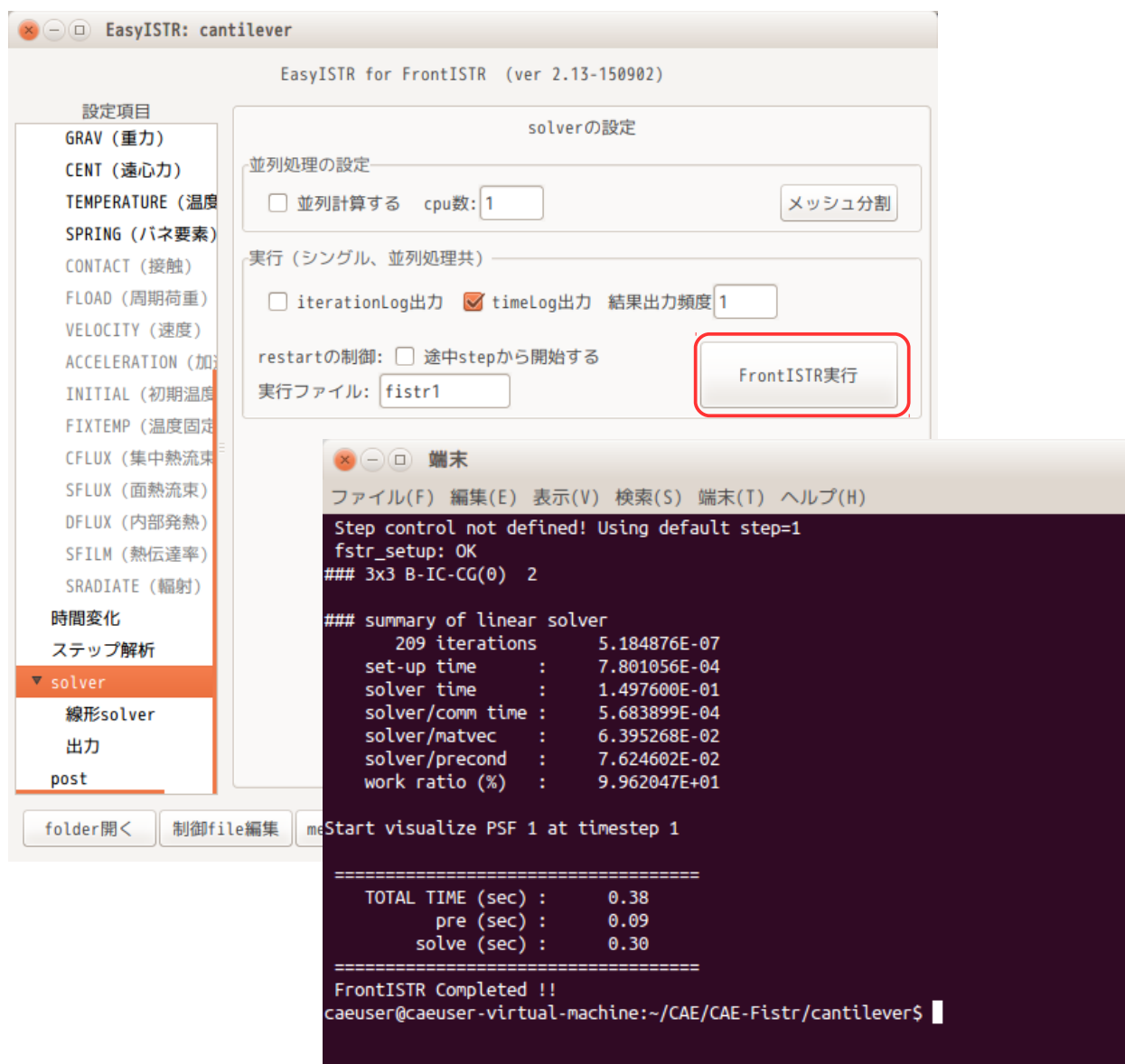




### 3-7-3. 計算開始

設定項目 Tree 内の「solver」を選択する。log の出力頻度等の設定を確認して、「FrontISTR 実行」ボタンをクリックする。

この操作で、fistr1 が起動し、計算が始まる。計算状況は、端末への出力で確認でき、この出力内容は、FistrModel.log ファイルへも出力されている。



### 3-8. 結果の可視化

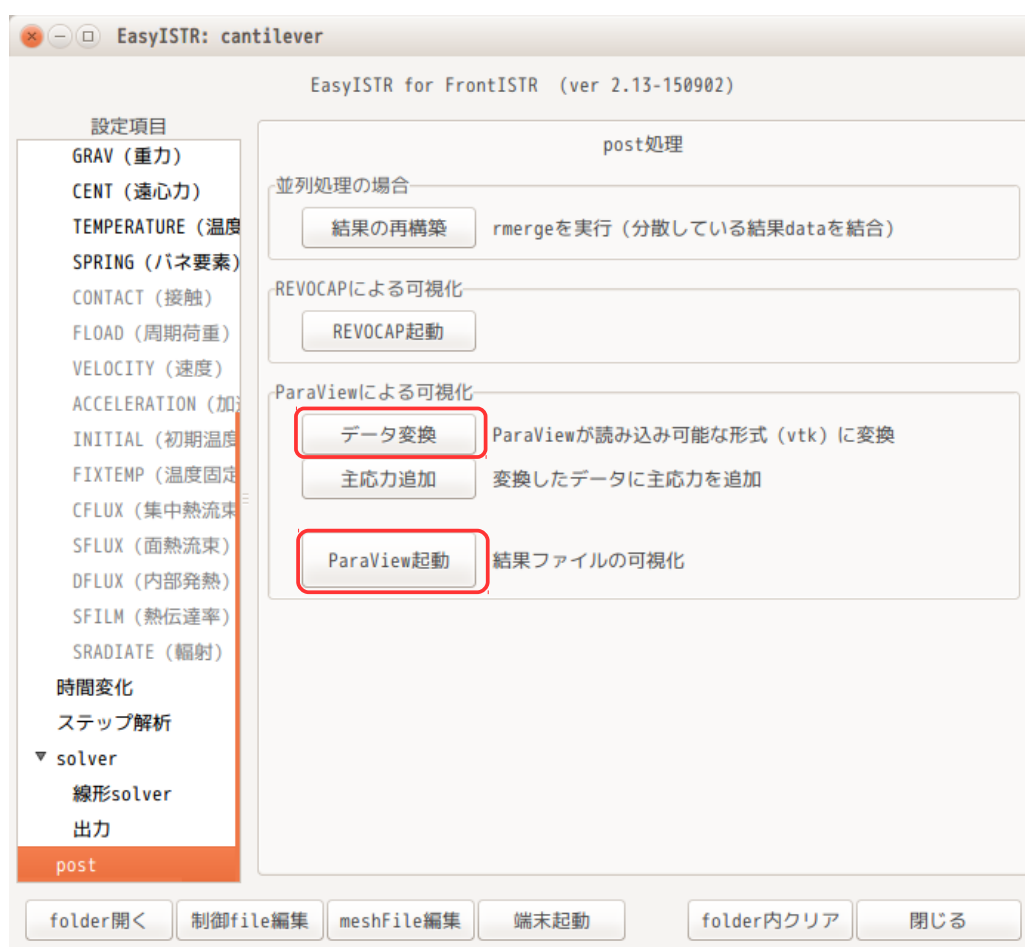
設定項目 Tree 内の「post」を選択して、計算結果を可視化する。

#### 3-8-1. FrontISTR の出力結果を可視化

ここで、FrontISTR が出力した項目 (3-7-2 項で出力項目を設定した項目) のみを可視化してみる。EasyISTR 画面上の「データ変換」ボタンをクリックして、FrontISTR が出力した結果ファイルを paraView が扱える vtk 形式に変換する。

FrontISTR が出力した項目に、主応力、主ひずみを追加するのであれば、「主応力追加」ボタンをクリックして、出力する項目を選択する。これら項目の追加に関しては、3-8-2 項を参照。

この後、「ParaView 起動」ボタンをクリックして、paraView を起動する。



paraView が起動すると、既にファイルを読み込んだ状態で起動するので、「Apply」ボタンをクリックする事で、結果が確認できる。

もし、paraView が起動した時、結果ファイルを読み込む事ができなかった場合や、望んだ結果ファイルでなかった場合は、paraView 上で、メニューバー「File」>「Open」を選択してファイル名を指定し、望みの結果ファイルを読み込むことになる。

結果のファイル名は、解析の種類や並列処理の有無で以下の様になっているので、これを考慮しながら、ファイル名を指定する。

#### 固有値解析

```
convFistrModel_eigen.res.<xxxx>.<n>.vtk          <xxxx>
                                                    並列処理:merge
                                                    シングル:0
```

#### 熱伝導(静・動)解析

```
convFistrModel_temp.res.<xxxx>.<n>.vtk          <n>
                                                    計算 step 数(1 ...)
```

#### 上記以外

```
convFistrModel.res.<xxxx>.<n>.vtk
```

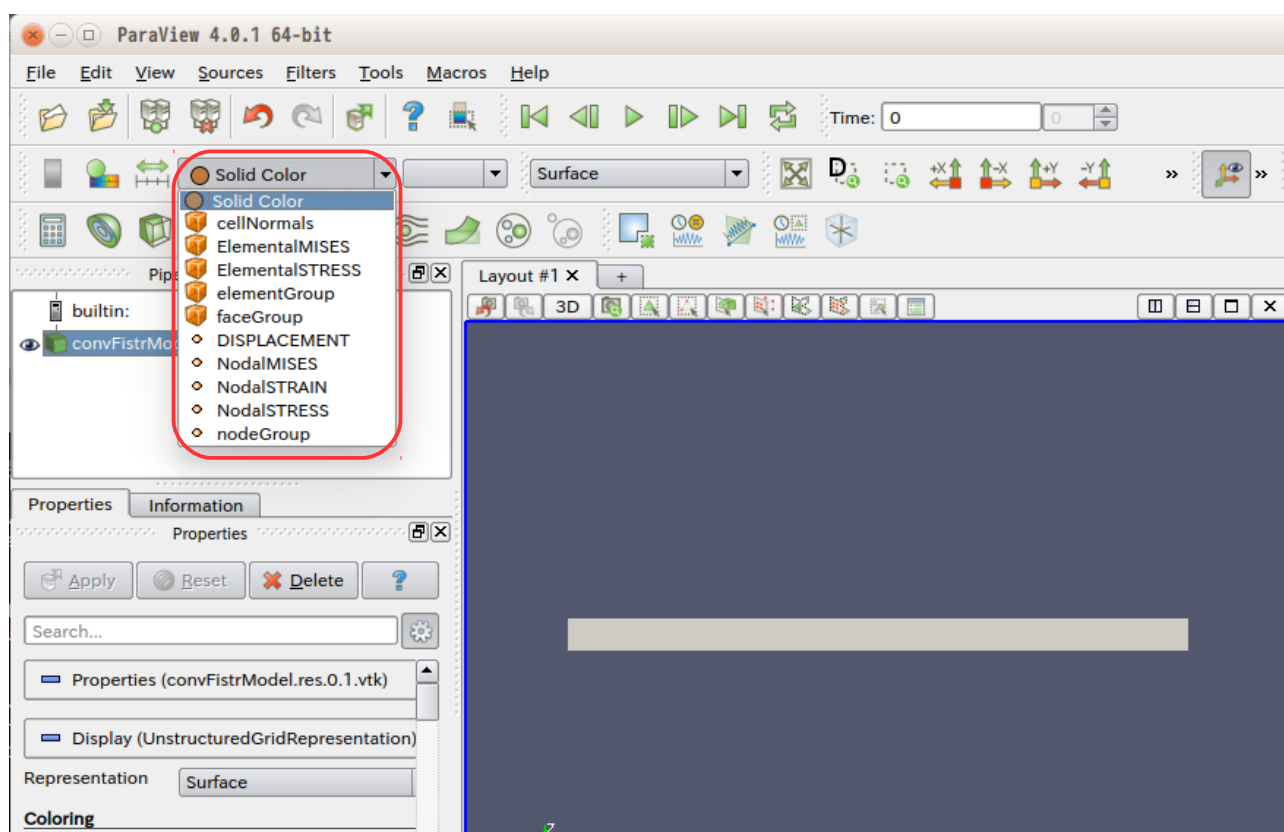
今回の計算結果として、3-7-2 項で以下の項目を設定している。

```
変位
節点 Mises 応力
節点応力
要素 Mises 応力
要素応力
```

paraView からは、下図の様に、

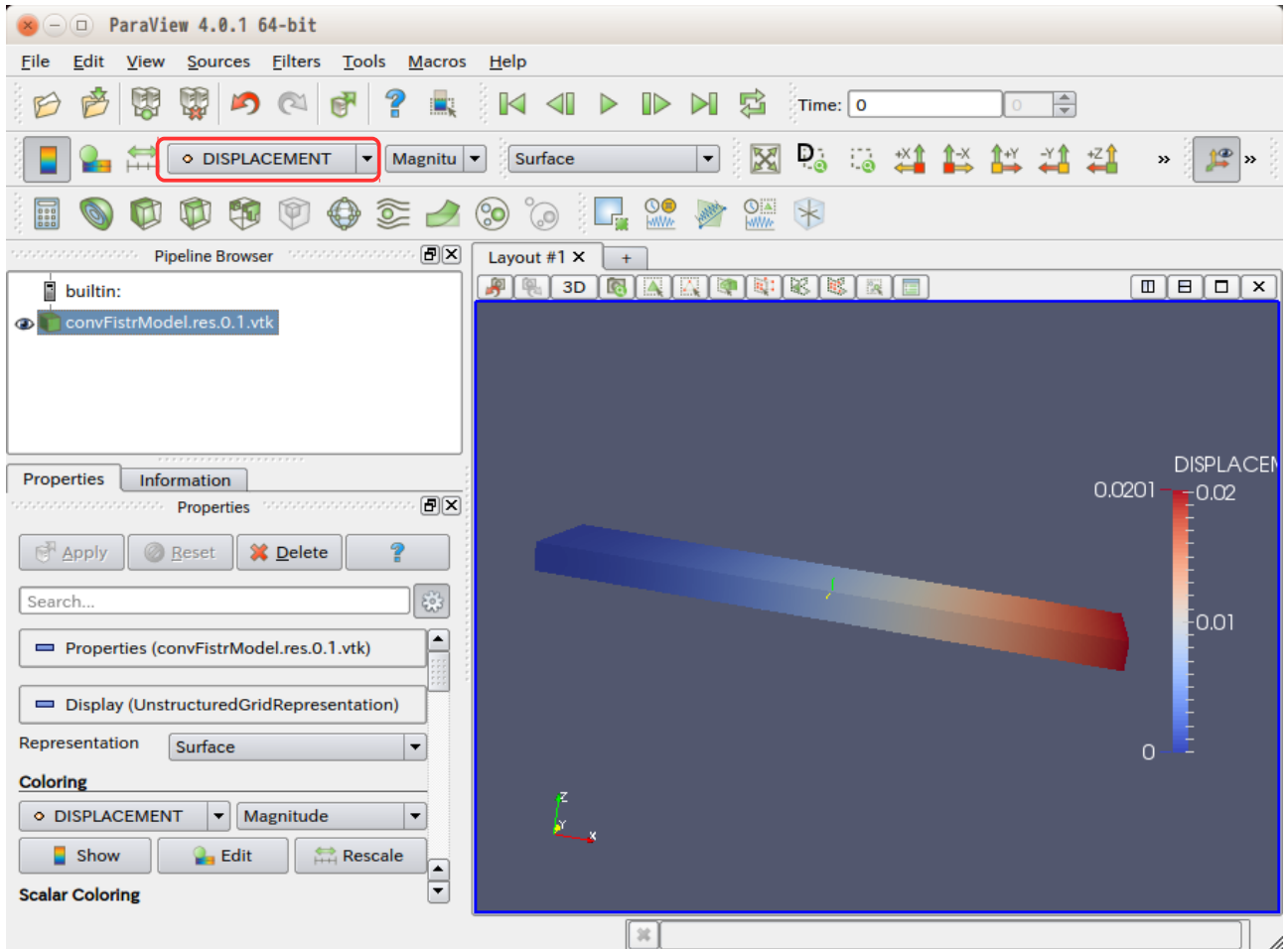
ElementalMISES	要素 Mises 応力
ElementalSTRESS	要素応力
DISPLACEMENT	変位
NodalMISES	節点 Mises
NodalSTRESS	節点応力

が選択できる。



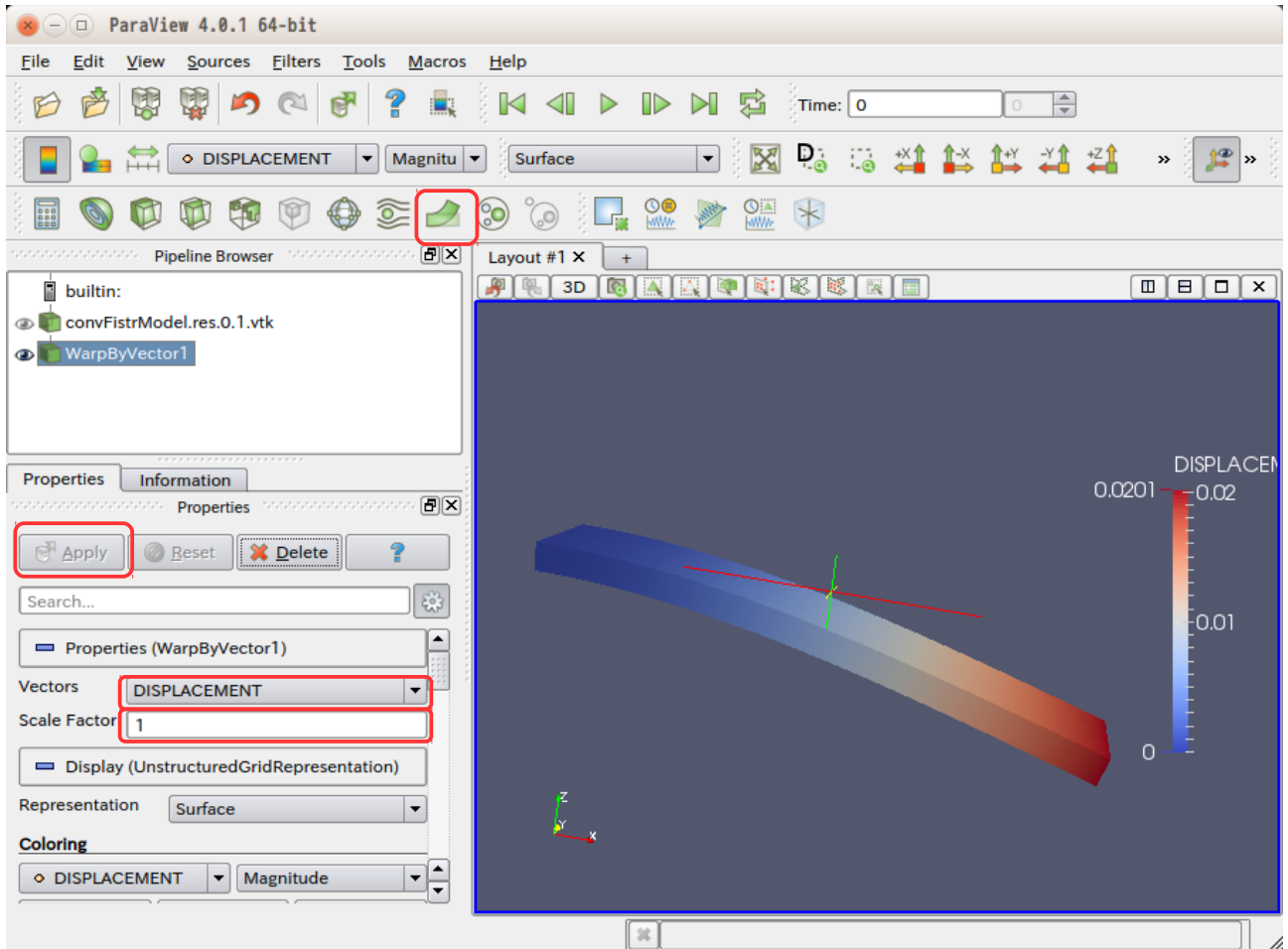
ここで、変位 (DISPLACEMENT) を読み込んで表示させてみる。

paraView 上で、表示項目内から「DISPLACEMENT」を選択する。下図が DISPLACEMENT を表示させた状態。



この状態では、変位量が色分けされているだけだが、モデル形状を変位に応じて変形させる事もできる。モデルを変位量分変形させる為には、「変形 (Wrap by vector) 」ボタンをクリックし、Vectros から「DISPLACEMENT」を選択し、「Apply」ボタンをクリックする。下図が、変形させた状態になる。

尚、変形の倍率は、自動で設定されないので、変形が大き過ぎたり、少な過ぎたりした場合は、「倍率 (Scale Factor) 」の数字を修正し、「Apply」ボタンをクリックする事で表示倍率を変更できる。



### 3-8-2. 主応力、主ひずみ、mises 応力、mises ひずみを追加する場合

現在のFrontISTRでは、出力項目として、主応力、主ひずみ、mises ひずみが出力できない。(mises 応力は、FrontISTRが出力できる。)

この為、これらを出力項目として追加して、その結果を可視化する事ができる様にしている。

その方法は、下図の画面上から、「データ変換」ボタンをクリックして、vtk ファイルを作成する。この後、算出の元データとなるテンソル名を確認後、「主応力追加」ボタンをクリックして、出力したい項目にチェックを入れ、「追加」ボタンをクリックする。今回の例では、全ての出力項目にチェックを入れている。「追加」ボタンのクリックによって、vtk ファイルに項目が追加される。

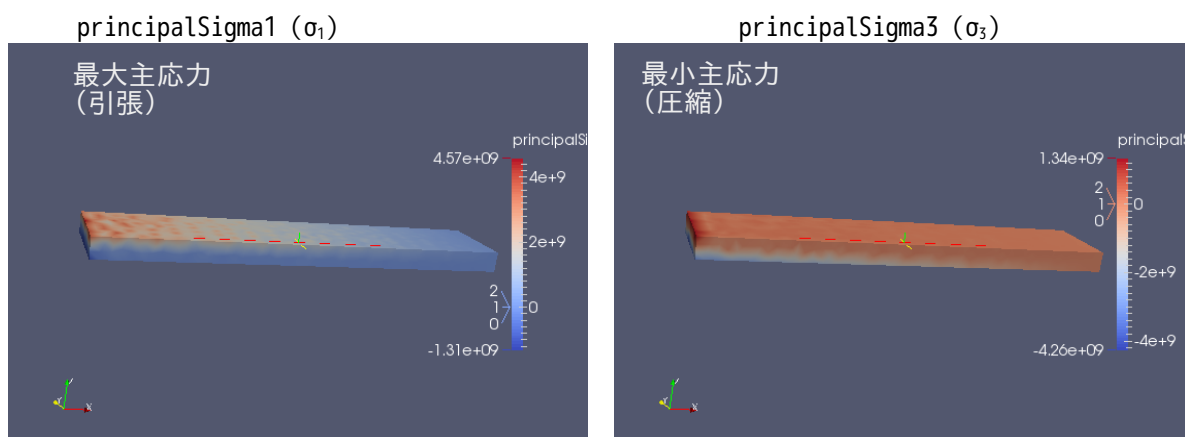
尚、これら出力項目は、

応力関係： 節点応力 (NSTRESS)  
ひずみ関係： 節点ひずみ (NSTRAIN)

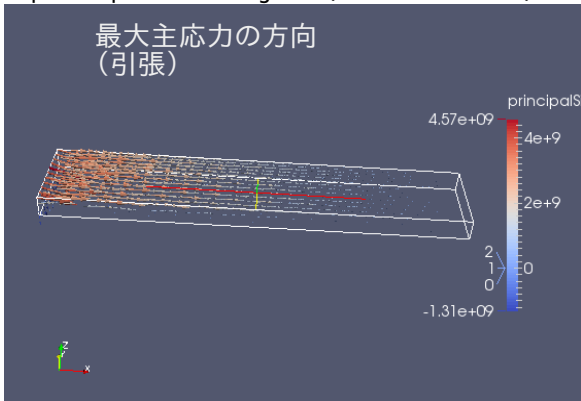
から計算しているので、これらの項目がFrontISTRから出力されている事が前提になる。これら項目が出力されていない場合は、3-7-2項の方法で出力項目を追加して、再計算し、データをvtk変換しておく必要がある。



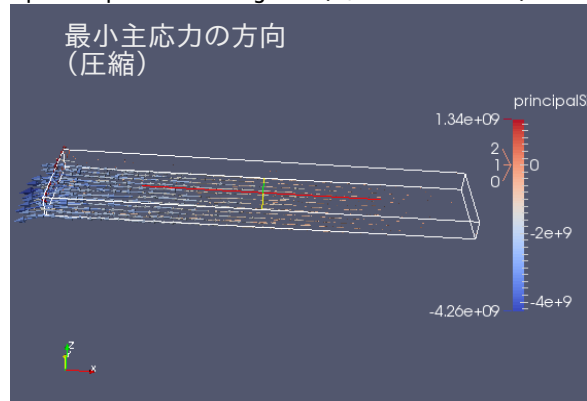
主応力 (主ひずみ) については、その主値と各方向の成分が出力できる様になっている。この為、その値と方向を知る事ができる。今回のモデルで、主応力 (圧縮と引張) の値とその方向を確認してみる。



principalVectorSigma1 ( $\sigma_1$ の各方向成分)



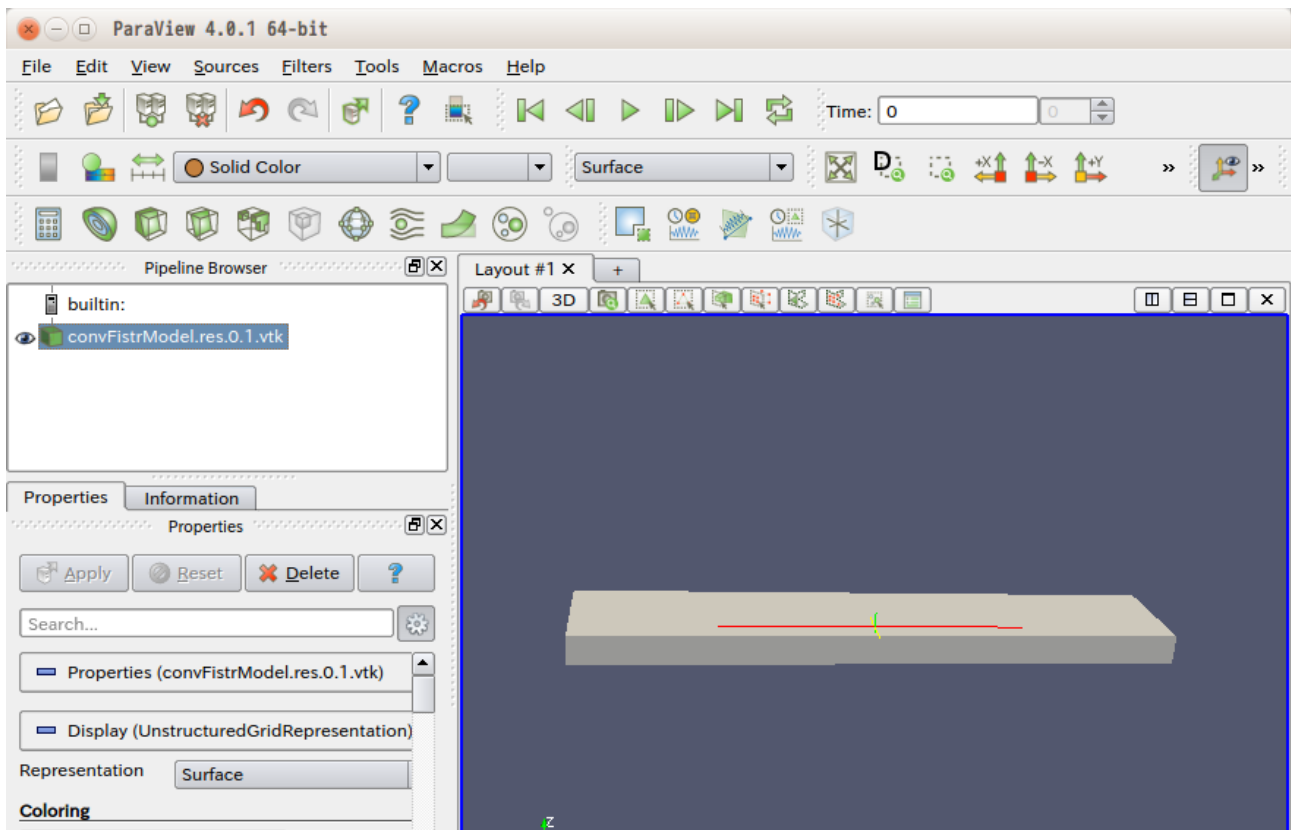
principalVectorSigma3 ( $\sigma_3$ の各方向成分)



上記の様に、片持ち梁の上面に引張が梁の長手方向に働き、下面に圧縮が梁の長手方向に働いている事が確認できる。

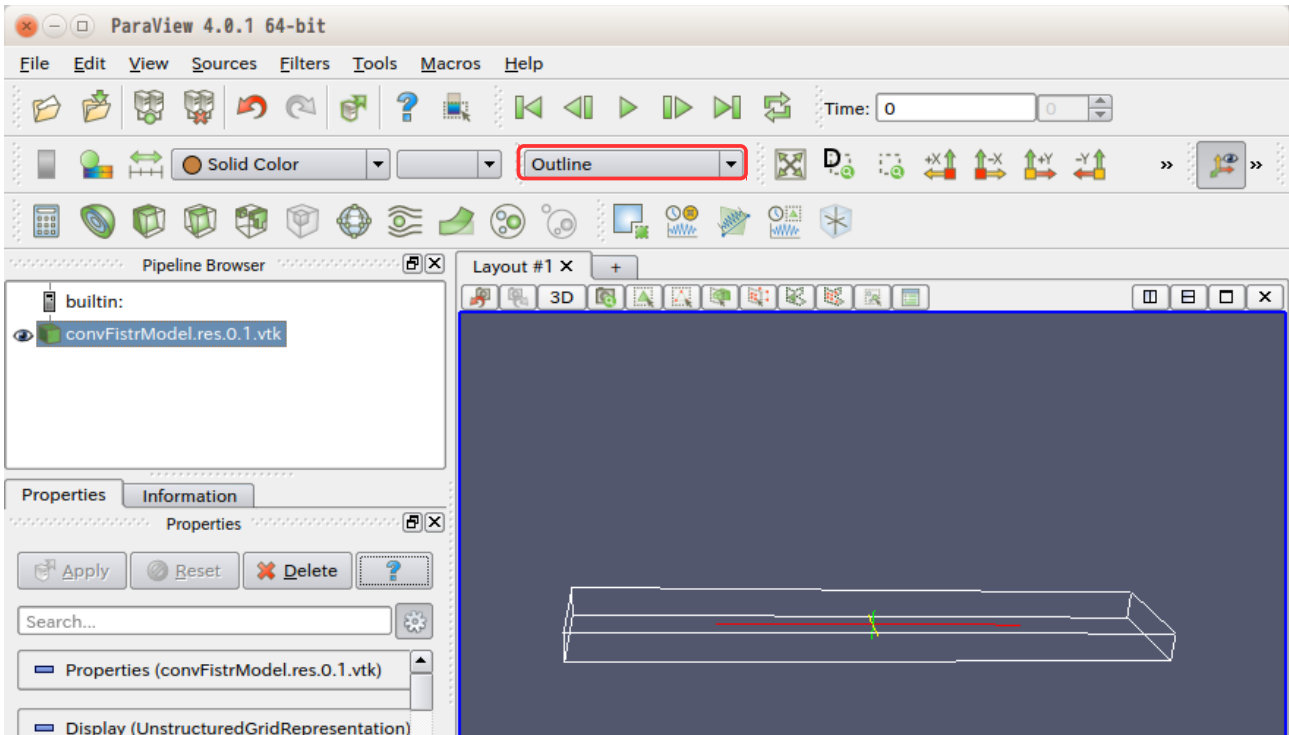
また、ParaViewによる矢印（ベクトル）の表示は、以下の方法で表示させる事ができる。以下の例では、最大主応力の表示を行っている。


その方法は、まずデータ変換した vtk ファイル (convFistrModel.res.0.1.vtk) を読み込む。下図が読み込んだ状態。

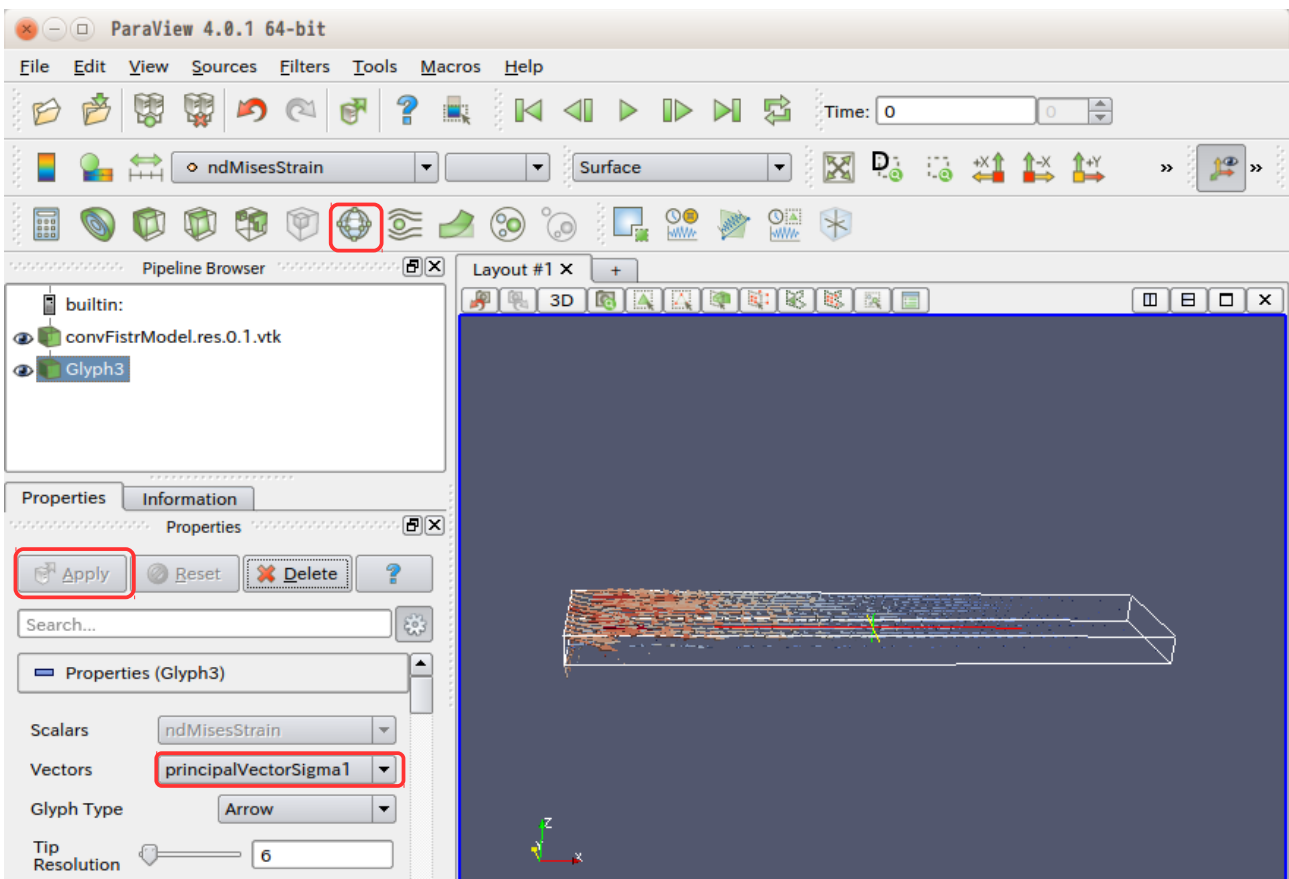



この後、下図の様に、現在の表示を「outline」に変更して、外形線を出力させておく。



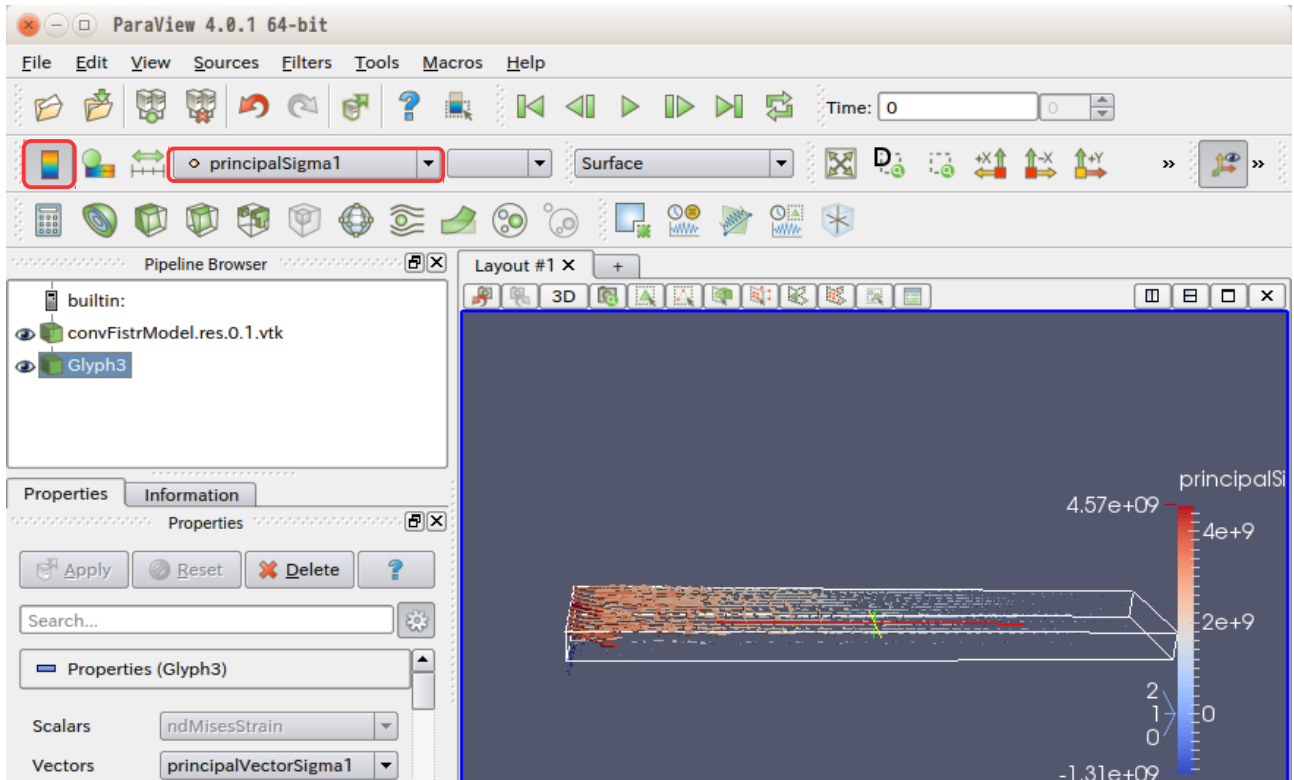


この後、ベクトルを表示させる為に、 ボタンをクリックし、vectors を「principalVectorSigma1」に設定し、「Apply」ボタンをクリックする。下図がこの状態になる。



次に、ベクトルの色の表示を「principalSigma1」に変更しておく。principalSigma1は、scalar の為、正負の値で、色分けされる。また、その色のスケールを表示させる為に、 ボタンをクリックする。下図がその状態になる。

以上で、最大主応力の方向が表示できたことになる。



### 3-9. 並列処理方法

材料物性値や境界条件の設定など、solver を起動させる前段階までは、並列、シングル処理とも同じ方法で、それ以降が並列処理特有の処理になる。

EasyISTR 上では、設定項目 Tree 内の「solver」以降で並列処理特有の処理をすることになる。ここでは、上記で解析したシングル処理方法を並列処理で行ってみる。

#### 3-9-1. 並列処理設定、計算開始

EasyISTR の設定項目 Tree 内の「solver」を選択し、この画面上で並列処理の設定を行う。

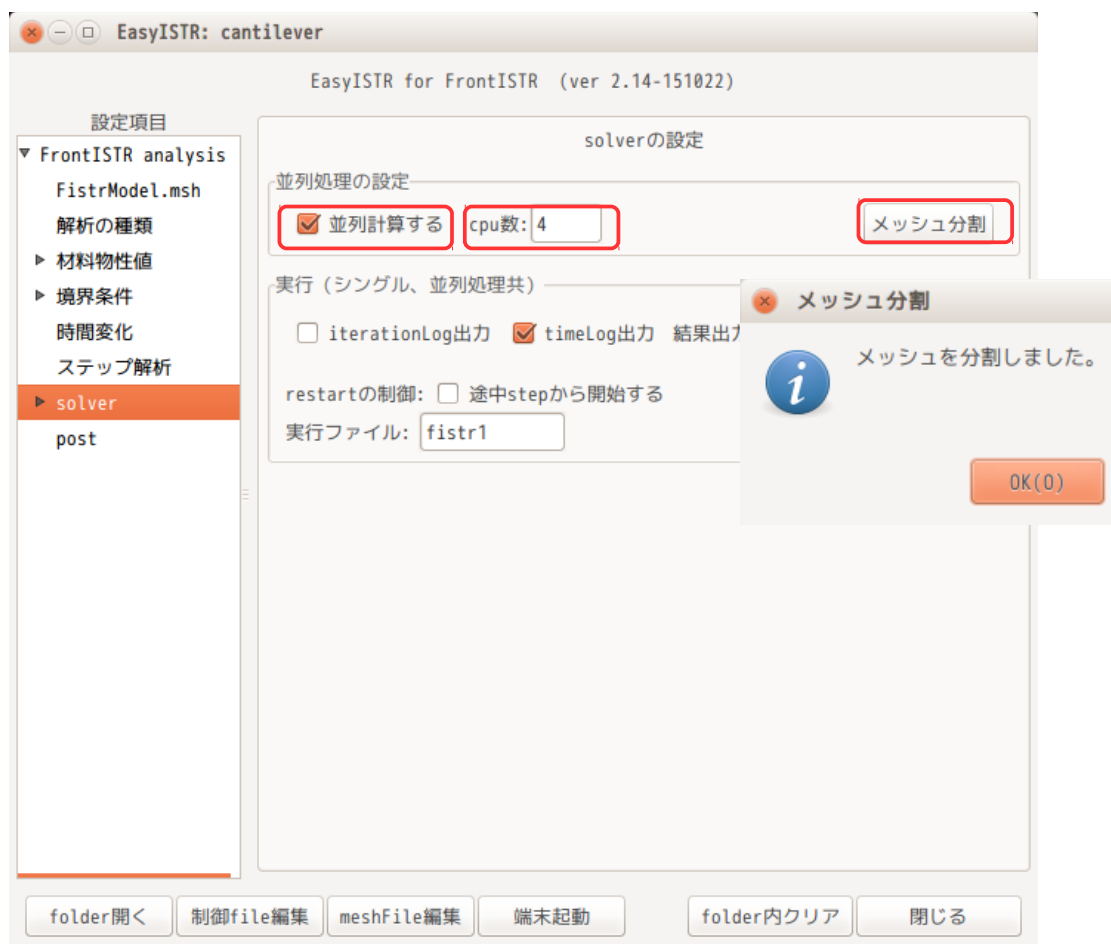
並列処理を行うために、チェックボックス「並列計算する」にチェックする。この時点で、並列処理するために必要な制御ファイル「hecmw\_part\_ctrl.dat」を確認し、workFolder 内に存在しない場合は、  
「hecmw\_part\_ctrl.dat」のバックアップファイルがあれば、そのファイルを元に戻す  
バックアップファイルが無ければ、デフォルトの「hecmw\_part\_ctrl.dat」をコピーする  
その後、hecmw\_part\_ctrl.dat ファイルに記述されている cpu 数を読み込んでテキストボックスに表示する。

この後、表示されている cpu 数を確認、修正する。

最後に、「メッシュ分割」ボタンをクリックして、メッシュを cpu 毎に分割する。「メッシュ分割」ボタン

をクリックした時点で、cpu 数を読み込み、「hecmw\_part\_ctrl.dat」ファイルに cpu 数を書き込み、そのファイル内容に従って、メッシュを分割していく。メッシュ分割が終了すると、その旨を表示する画面が現れる。

尚、チェックボックス「並列計算する」のチェックを外すと、並列制御ファイル「hecmw\_part\_ctrl.dat」をバックアップファイル「hecmw\_part\_ctrl.dat.bak」に置き換え、再びチェックした時に備えている。



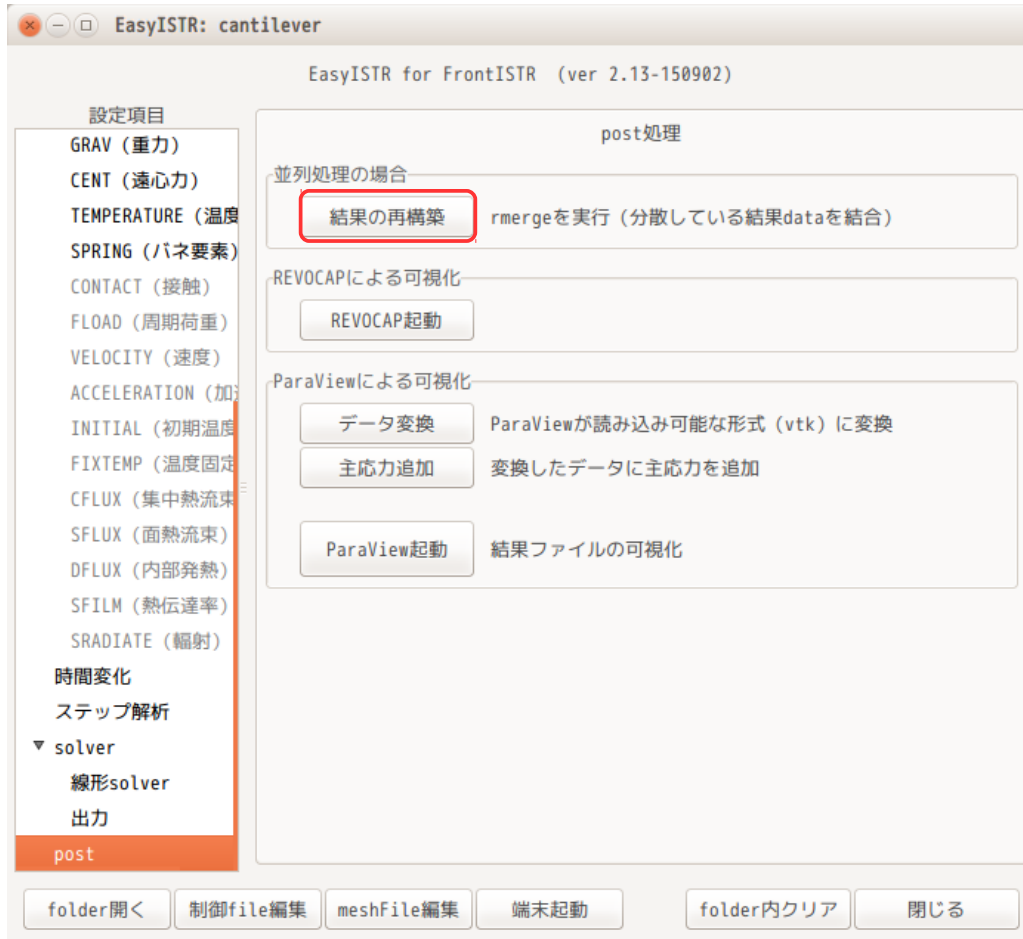
以上で、並列計算する為の設定が終了したので、「FrontISTR 実行（並列）」ボタンをクリックする事で、並列処理が開始する。



### 3-9-2. 並列計算結果の可視化

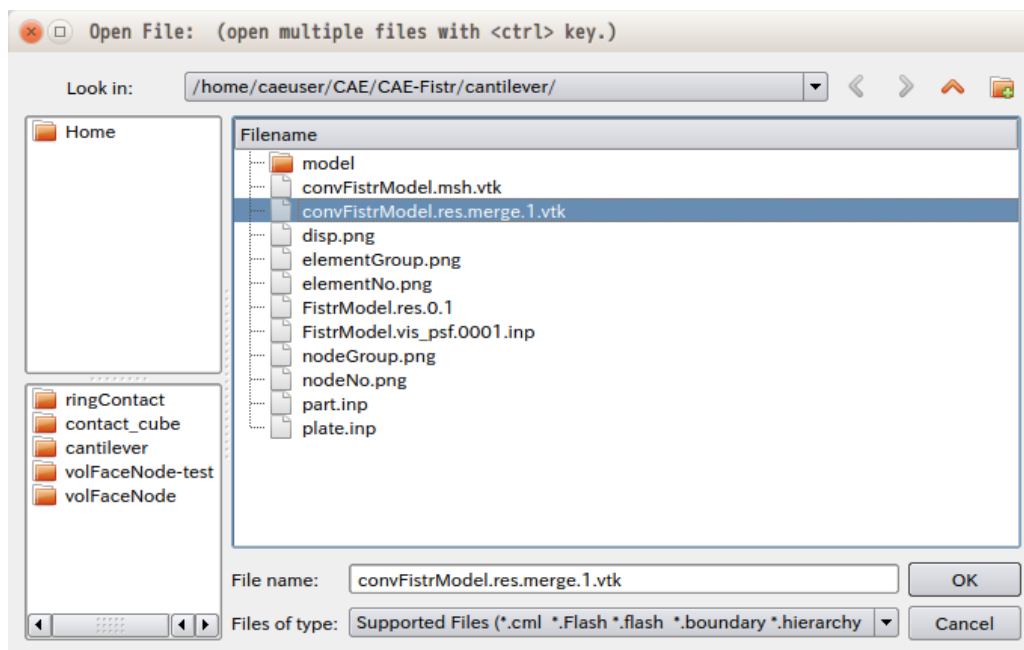
並列計算結果は、cpu 毎に分散して保存されているので、その分散結果を再構築して、ひと塊にする必要がある。

その方法は、下図の「結果の再構築」ボタンをクリックする事で、結果を再構築することができる。

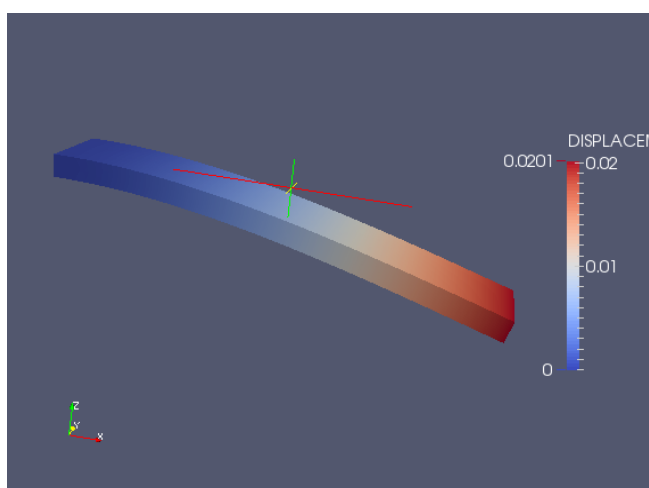


この後は、シングル処理と同様な操作で「データ変換」、必要に応じ「主応力追加」、「paraView 起動」ボタンをクリックして、計算結果が確認できる。

並列処理の場合は、以下の「\*\*\*\*.res.merge.\*\*\*.vtk」ファイルを選択する。(このファイル名が分散結果を merge して再構築した結果ファイルになる。)



「convFistrModel.res.merge.1.vtk」ファイルを選択して、変位を表示させた結果が以下になる。



### 3-10. EasyISTR 下部ボタンの動作

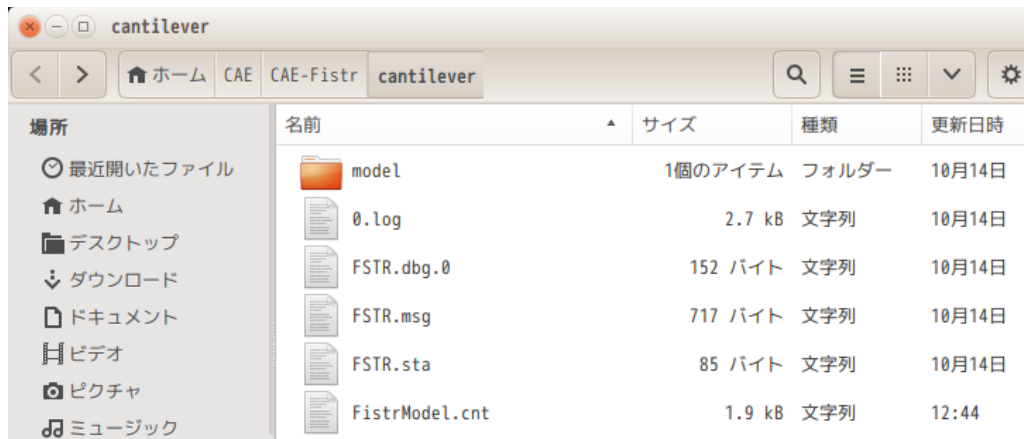
EasyISTR 下部には、以下のボタンを配置している。



次項以降に、これらボタンの動作について説明する。

### 3-10-1. 「folde 開く」ボタンの動作

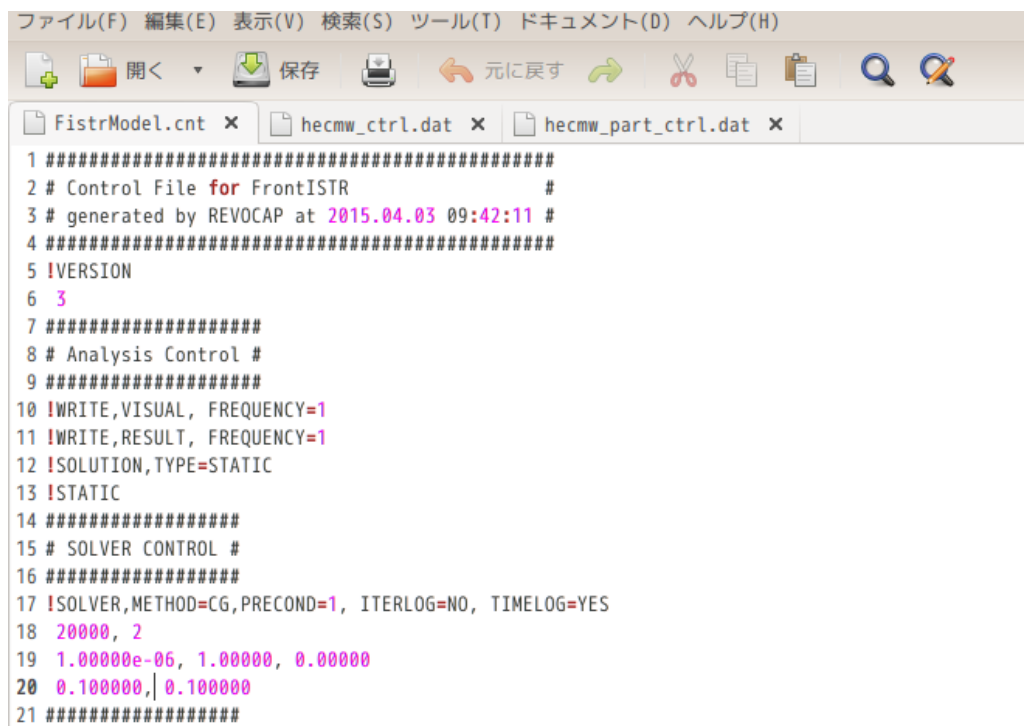
このボタンをクリックする事で、workFolder を開く事ができる。下図参照。



### 3-10-2. 「制御 file 編集」ボタンの動作

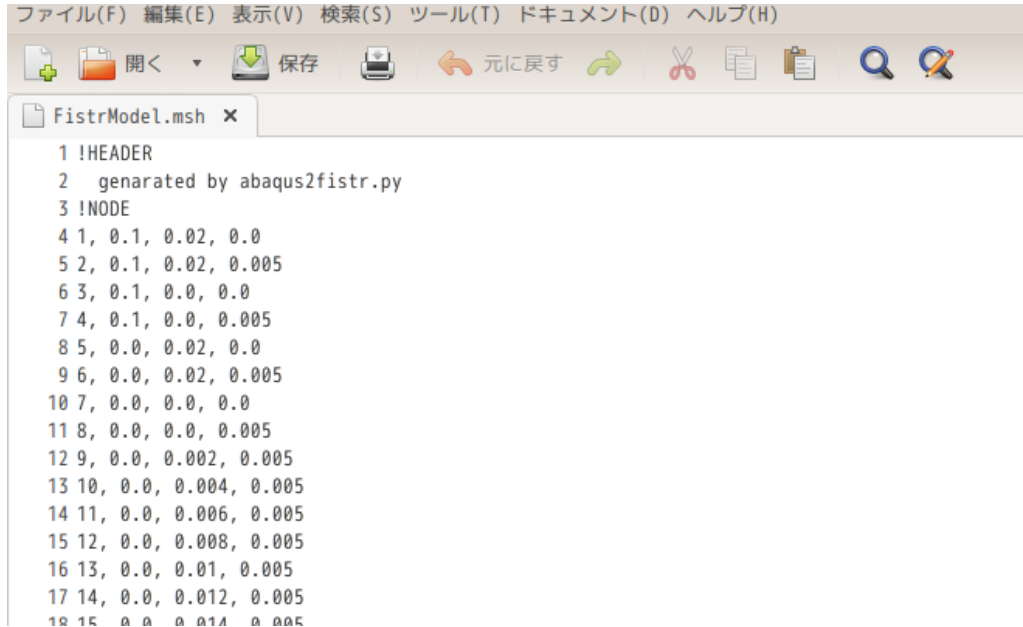
このボタンをクリックする事で、以下のファイルを editor で開き、編集する事ができる。

FistrModel.cnt	制御ファイル
hecmw_ctrl.dat	全体制御ファイル
hecmw_part_ctrl.dat	メッシュ分割ファイル (並列処理時のみ)



### 3-10-3. 「meshFile 編集」ボタンの動作

このボタンをクリックする事で、メッシュファイル「Fistrmodel.msh」を editor で開き編集する事ができる。



```
1 !HEADER
2 generated by abaqus2fistr.py
3 !NODE
4 1, 0.1, 0.02, 0.0
5 2, 0.1, 0.02, 0.005
6 3, 0.1, 0.0, 0.0
7 4, 0.1, 0.0, 0.005
8 5, 0.0, 0.02, 0.0
9 6, 0.0, 0.02, 0.005
10 7, 0.0, 0.0, 0.0
11 8, 0.0, 0.0, 0.005
12 9, 0.0, 0.002, 0.005
13 10, 0.0, 0.004, 0.005
14 11, 0.0, 0.006, 0.005
15 12, 0.0, 0.008, 0.005
16 13, 0.0, 0.01, 0.005
17 14, 0.0, 0.012, 0.005
18 15 0 0 0 0.014 0 0.005
```

### 3-10-4. 「端末起動」ボタンの動作

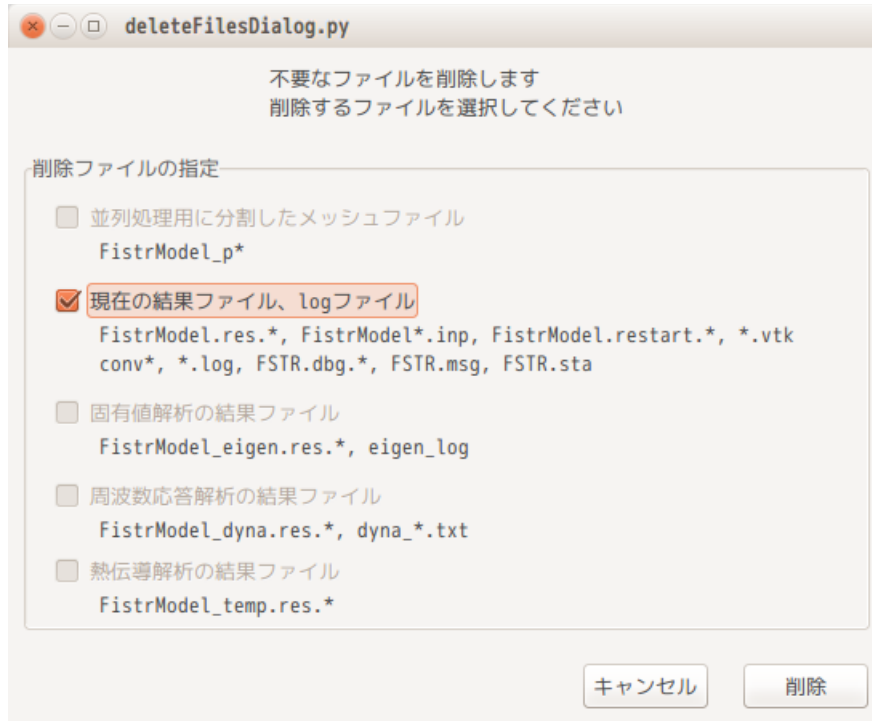
このボタンをクリックする事で、workFolder をカレントディレクトリとして、端末を起動する事ができる。この為、ここで、直接コマンドを実行して、確認する事もできる。



```
caeuser@caeuser-virtual-machine: ~/CAE/CAE-Fistr/cantilever
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
caeuser@caeuser-virtual-machine:~/CAE/CAE-Fistr/cantilever$
```

### 3-10-5. 「folder 内クリア」ボタンの動作

このボタンをクリックすると、以下の画面が現れるので、workFolder 内の file を目的に応じて削除できる。



操作方法は、削除したい内容をチェックし、「削除」ボタンをクリックする事で削除できる。

尚、削除されるファイルは、各項目に表示されているファイルになる。

また、起動時にチェックボタンが選択できない項目があるが、これは、その項目に関連するファイルが存在していない事を表している。

## 3-11. ファイル変換について

EasyISTR では、Salome、FrontISTR、paraView などのアプリケーションが扱える様にファイル変換を行っている。

### 3-11-1. unv2fistr 変換 (unv 形式のメッシュを FrontISTR 用に変換)

Salome が吐き出した unv 形式のメッシュファイルを FrontISTR が扱える fistr 形式に変換する。

この操作は、EasyISTR 上で、以下のコマンドを実行して、実現している。

```
$ unv2fistr.py plate.unv
-----
unv 形式のメッシュファイル
```

このコマンドを実行すると、fistr 形式の「plate.msh」ファイルができあがり、これをコピーして最終的に「FistrModel.msh」ファイルを作っている。



unv2fistr.py コマンドの動作は、以下の順番でメッシュ変換を行っている。

unv 形式 → abaqus 形式 → fistr 形式  
(unv2abaqus.py) (abaqus2fistr.py)

unv 形式から abaqus 形式への変換は、CAELinux に同梱されている「unv2abaqus.py」をそのまま利用している。abaqus 形式のメッシュから fistr 形式への変換は、新たに作成した「abaqus2fistr.py」で行っている。

abaqus2fistr.py の変換については、現在のところ以下の制限がある。

扱えるメッシュタイプ

四面体 (テトラ)	1次2次
六面体 (ヘキサ)	1次2次
五面体 (プリズム)	1次2次
三角形 (シェル)	1次
四角形 (シェル)	1次

elementGroup 単位で1種類のメッシュタイプ

同一の elementGroup 内では、1種類のメッシュタイプしか使えない

elementGroup が異なれば、異なるメッシュタイプが使える

シェルとソリッドの混在は、許されない。(シェルのみ、ソリッドのみのモデル)

この為、unv2fistr.py も上記制限が伴う。

### 3-11-2. abaqus2fistr 変換 (abaqus 用 inp ファイルを FrontISTR 用に変換)

この変換は、メッシュ部の変換 (abaqus2fistr.py) と境界条件部の変換 (abaqusinp2fistrCnt.py) に分かれている。

メッシュ部の変換は、以下のコマンドを実行して、メッシュ変換している。変換後は、FistrModel.msh ファイルができあがり、このファイルが FrontISTR 用のメッシュファイルになる。

```
$ abaqus2fistr.py plate.inp
```

-----  
変換対象の inp ファイル

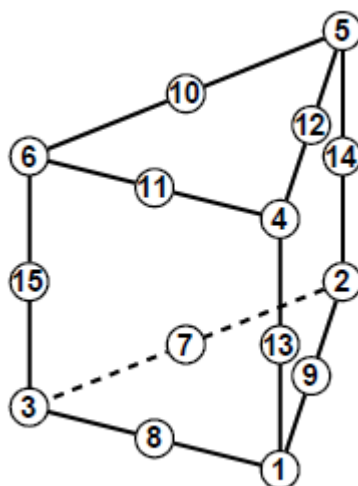
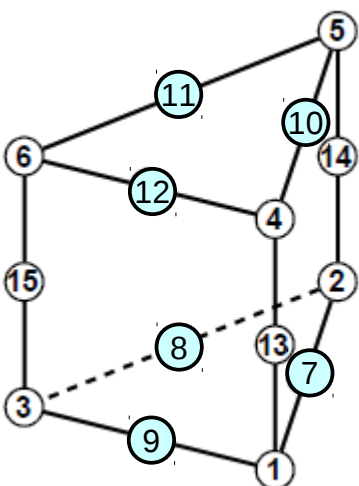
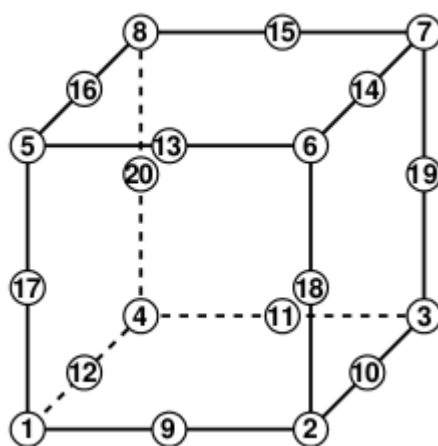
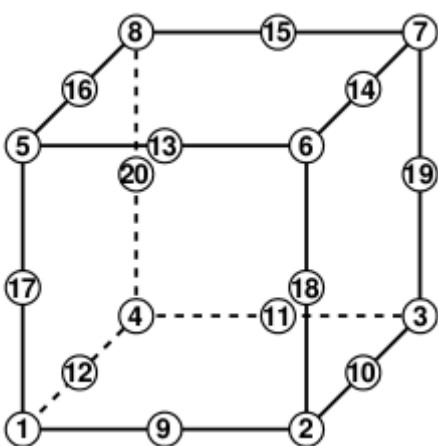
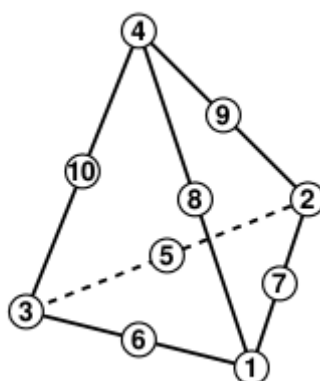
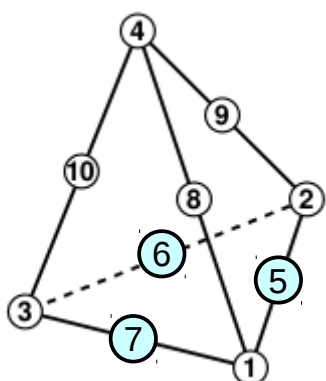
この変換は、現在のところ、前項で示したように、扱えるメッシュタイプは、以下に限られる。また、解析モデル中に solid と shell の混在は許されていない。

要素タイプ	abaqus	FrontISTR	備考
四面体 1次2次	C3D4、C3D19	341、342	solid
六面体 1次2次	C3D8、C3D8R、C3D20、C3D20R	361、362	↑
五面体 1次2次	C3D6、C3D15	351、352	↑
三角形 1次	STRI35、S3	731、761	shell
四角形 1次	S4R5	741、781	↑

上記の abaqus メッシュを変換するに当たり、各要素の節点 No 順が abaqus と FrontISTR 間で、以下の様に異なっている部分が存在しているので、これを修正した上で変換している。

<Abaqus>

<FrontISTR>



また、このメッシュ変換に関しては、abaqus 用 inp ファイルから以下の keyword を解釈して、FrontISTR 用の msh ファイルを作成している。

Abaqus	FrontISTR	
*NODE	!NODE	
*ELEMENT		
TYPE=(solid 要素)	!ELEMENT	#solid 要素の場合、!ELEMENT として取得
TYPE=(表面要素)	!SGROUP	#表面要素の場合、!SGROUP として取得
ELSET	EGRP	#ELSET は、そのまま EGRP (要素グループ) で取得
*NSET	!NGROUP	
*ESET	!EGROUP	
*SURFACE		
TYPE=NODE	!NGROUP	
TYPE=ELEMENT	!SGROUP	

メッシュ部を変換した後、inp ファイル内の境界条件を確認し、一部境界条件について FrontISTR 用に変換して、結果を FistModel.cnt ファイルに追記している。

境界条件の変換は、以下のコマンドを実行している。

```
$ abaqusinp2fistrcnt.py plate.inp
-----
      変換対象のファイル
```

この変換は、現在のところ、以下の keyword を対象にしている。(解析方法や材料物性値は、変換の対象に含めていない。)

また、keyword の操作対象をグループ名ではなく、節点 No や要素 No で直接指定している場合は、これらの各々 No に対応した節点グループや要素グループを作成した上で、変換している。

Abaqus	FrontISTR	操作対象
*BOUNDARY	!BOUNDARY、!FIXTEMP	節点
*CLOAD	!CLOAD	節点
*DLOAD	!DLOAD	要素
P1~P6	P1~P6	
GRAV	GRAV	
CENTRIF	CENT	
*SPRING	!SPRING	節点
*INITIAL CONDITIONS	!INITIAL CONDITION	節点
*TEMPERATURE	!TEMPERATURE	節点
*CFLUX	!CFLUX	節点

この境界条件の変換は、全てを確認した訳ではないが、calculix の「CalculiX CrunchiX USER'S MANUAL」中の例題「5. Simple example problem」に現れている inp ファイル (beamf.inp, beamf2.inp, beamp.inp, rotor.inp) は、メッシュ、境界条件とも変換はできている。

以下に変換例を示してあるが、この inp ファイルは、calculix/test フォルダ内に収められている inp ファイルを用いている。

----- inp ファイルの境界条件を変換した例 -----

<pre>&lt; inp ファイル :beam10p.inp &gt; : *NSET,NSET=FIX 1,4,2,6,5,8,3,9,7 *NSET,NSET=LOAD 10,15,12,13,14,18,11,17,16 *BOUNDARY 1,1,2 3,1,1 FIX,3,3 *MATERIAL,NAME=EL *ELASTIC 210000.,.3 *SOLID SECTION,ELSET=EALL,MATERIAL=EL *STEP *STATIC *CLOAD LOAD,2,1. *NODE PRINT,NSET=NALL U,RF *EL PRINT,ELSET=EALL S *END STEP</pre>		<pre>&lt; msh ファイル &gt; 取得した境界条件 : ##### # Boundary Condition # ##### !BOUNDARY node_1, 1, 2, 0.0 !BOUNDARY node_3, 1, 1, 0.0 !BOUNDARY FIX, 3, 3, 0.0 !CLOAD LOAD, 2, 1. ##### :</pre>
---	--	---

### 3-11-3. fistr2vtk 変換 (FrontISTR 用のファイルを paraView 用に変換)

FrontISTR 用のメッシュファイルや結果ファイルは、そのままでは paraView で確認できないので、EasyISTR 側で vtk 変換している。

#### 3-11-3-1. メッシュファイルの変換

メッシュファイル「FistrModel.msh」を vtk 変換して「convFistrModel.msh.vtl」ファイルを作り出し、paraView で読むことができる形式に変換している。  
この vtk 変換は、以下のコマンドを実行して作成している。

```
$ fistr2vtk.py -mesh FistrModel
-----
メッシュファイルの fileHeader 名
```

この変換は、節点と要素 (volume 要素と外表面の face 要素) を作成し、節点に対応する値、要素に対応する値を設定している。  
以下は、一例として、小さいモデルを vtk 変換したファイル内容になる。(今回のメッシュデータではなく、簡易的に作成したモデルになる。)

```
----- convFistrModel.msh.vtk -----
# vtk DataFile Version 2.0
surface
```

ASCII

DATASET UNSTRUCTURED\_GRID

POINTS 14 float

座標データ (14 ポイント)

```

100.0 100.0 0.0
100.0 100.0 100.0
100.0 0.0 100.0
100.0 0.0 0.0
0.0 100.0 0.0
0.0 100.0 100.0
0.0 0.0 0.0
0.0 0.0 100.0
200.0 100.0 100.0
200.0 100.0 0.0
200.0 0.0 100.0
200.0 0.0 0.0
50.0 50.0 50.0
150.0 50.0 50.0

```

CELLS 46 208

要素データ (46 要素)

4 0 3 2 12

4 0 3 2 12 → 4 点の節点 No (4 面体)

4 0 2 1 12

volume 要素

4 7 12 6 5

4 2 6 7 12

4 3 6 2 12

4 2 7 1 12

4 1 12 7 5

4 1 0 12 5

4 5 4 0 12

4 4 3 0 12

4 6 3 4 12

4 4 5 6 12

4 13 11 9 10

4 13 3 2 0

4 13 2 3 10

4 2 1 8 13

4 1 2 0 13

4 13 9 11 0

4 10 8 9 13

4 8 10 2 13

4 13 11 3 0

4 1 0 9 13

4 1 9 8 13

4 13 3 11 10

3 11 9 10

3 11 9 10 → 3 点の節点 No (3 三角形の面)

3 0 3 2

face 要素

3 0 2 1

3 2 6 7

3 2 3 10

3 2 1 8

3 3 6 2

3 9 11 0

3 2 7 1

3 7 1 5

3 10 8 9

3 1 0 5

3 5 4 0  
3 4 3 0  
3 8 10 2  
3 6 3 4  
3 11 3 0  
3 1 0 9  
3 1 9 8  
3 3 11 10  
3 6 7 5  
3 4 5 6

CELL\_TYPES 46

要素タイプ設定 (46 要素)

10

10 → 1次四面体

10

volume 要素

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

10

7

7 → 多角形 ( polygon )

7

face 要素

7

7

7

7

7

7

7

7

7

7

7

7

7

7

7

7

7

7

7

7  
7  
7  
POINT\_DATA 14  
FIELD attributes 2  
nodeNo 1 14 int                    節点に値を設定 (節点数分設定)  
1                                    nodeNo を設定  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
nodeGroup 1 14 int                nodeGroupNo を設定 (節点数分を設定)  
0  
0  
0  
0  
0  
0  
0  
0  
1  
1  
1  
1  
0  
0  
CELL\_DATA 46  
FIELD attributes 3  
elementNo 1 46 int                要素に値を設定 (要素数分を設定)  
44                                  elementNo を設定  
45  
46  
47  
51  
54  
55  
57  
58  
59  
61  
62  
43  
48  
49  
50

52  
53  
56  
60  
63  
64  
65  
66  
43  
44  
45  
47  
49  
50  
51  
53  
54  
55  
56  
57  
58  
59  
60  
61  
63  
64  
65  
66  
46  
62

elementGroup 1 46 int

elementGroupNo を設定 (要素数分を設定)

1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
2  
2  
2  
2  
2  
2  
2  
2  
2  
2  
2  
2





1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
2  
2

---

### 3-11-3-2. 結果ファイルの変換

FrontISTR が吐き出した結果ファイル「FistrModel.res0.1」は、paraView で読み込む事ができないので、このファイルを vtk 変換して、「convFistrModel.res.0.1.vtk」を作り出し、paraView で読み込んでいる。

結果ファイルの vtk 変換は、以下のコマンドを実行して作成している。変換するファイルは、いずれも fileHeader 名を指定する。

```
$ fistr2vtk.py -mesh FistrModel -res FistrModel
                -----
                fileHeader 名   fileHeader 名
```

このコマンドの実行で、メッシュと結果ファイルをから、vtk 変換されたファイル「convFistrModel.res.0.vtk」ができあがる。

vtk 形式のファイルの内容は、前項で示しているが、この内容に節点結果データと要素結果データを追加している。

## 3-12. salome によるメッシュの作成方法

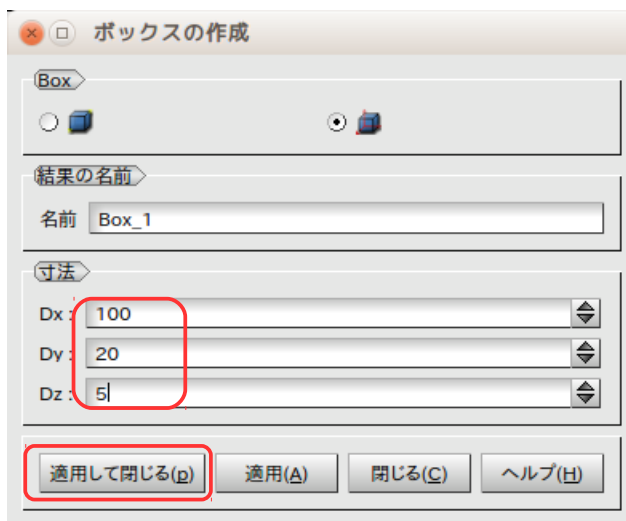
この項で例に取り上げているメッシュの作成方法について説明する。

### 3-12-1. ジオメトリの作成

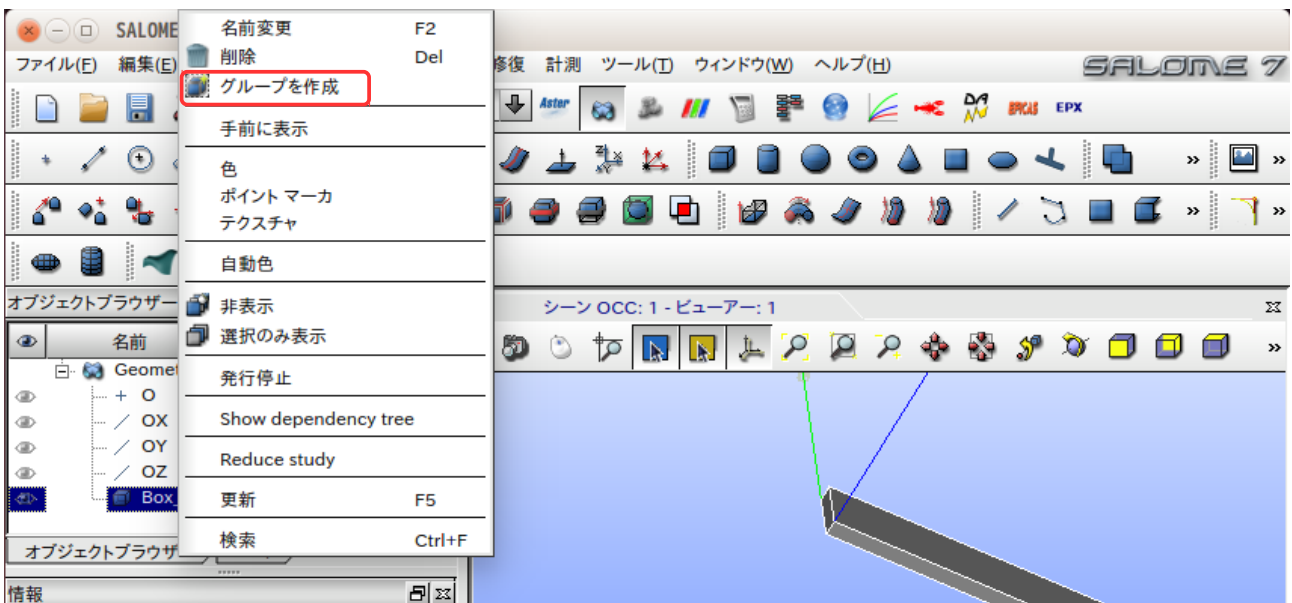
作成するモデル形状は、100 x 20 x 5 mm の直方体になる。Salome を起動後、Geometry モジュールに設定する。この後、「ボックスを作成」ボタンをクリックし、XYZ の寸法を入力する。



以下の画面が XYZ の寸法を入力した状態。この後、「適用して閉じる」ボタンをクリックして、直方体ができあがる。



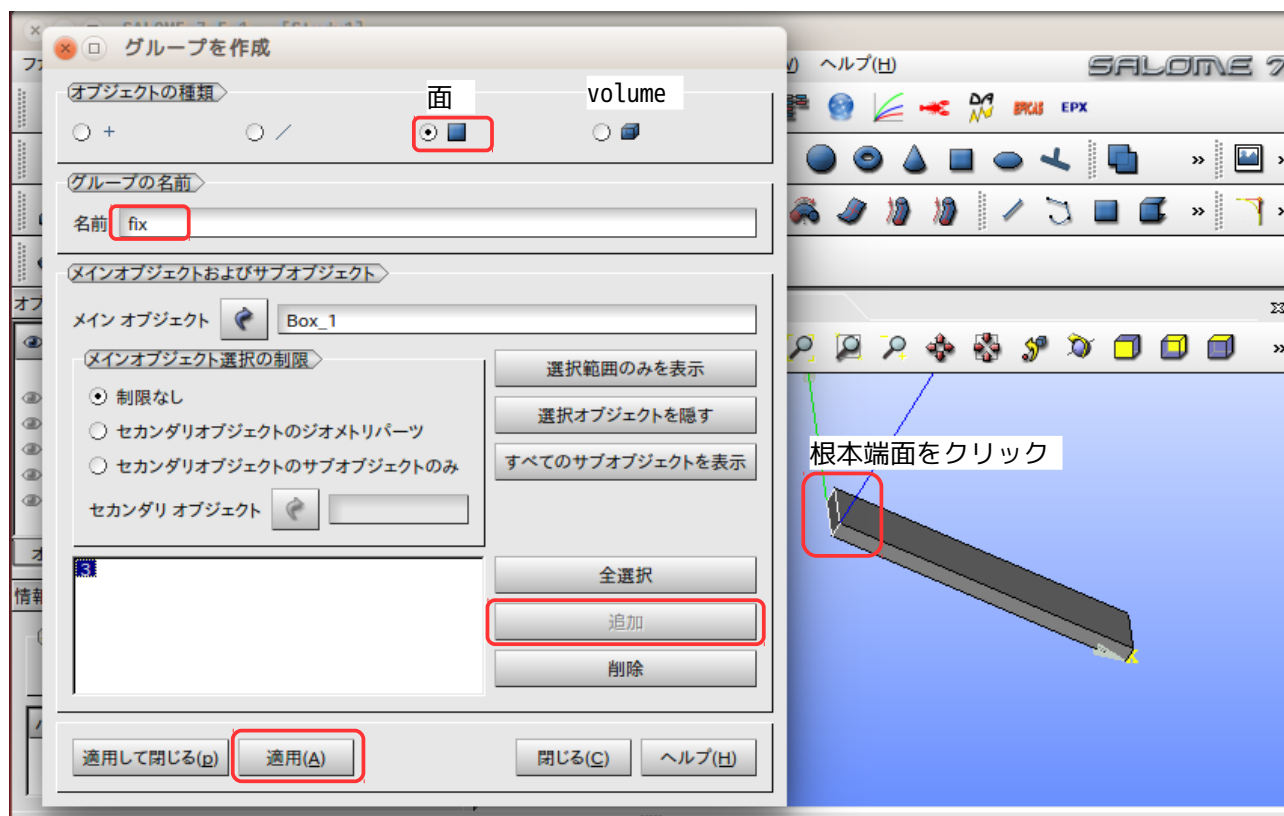
この後は、メッシュを作成した後の為に、作成した直方体に境界条件を設定する箇所に対してグループを作成する。直方体の根本、先端の端面と上面のグループを作成する。オブジェクトブラウザーの「Box\_1」を選択して、右クリックして、メニューを表示させ、「グループを作成」を選択する。



「グループ作成」を選択した後、下図の様に、「面」を選択し、名前「fix」を入力し、直方体の根本端面を、クリックして、「追加」ボタンをクリックする。下図が、その状態になる。この後、「適用」ボタンをクリックして、グループ「fix」を作成する。

同様にして、先端の端面「load」、上面「press」を作成する。

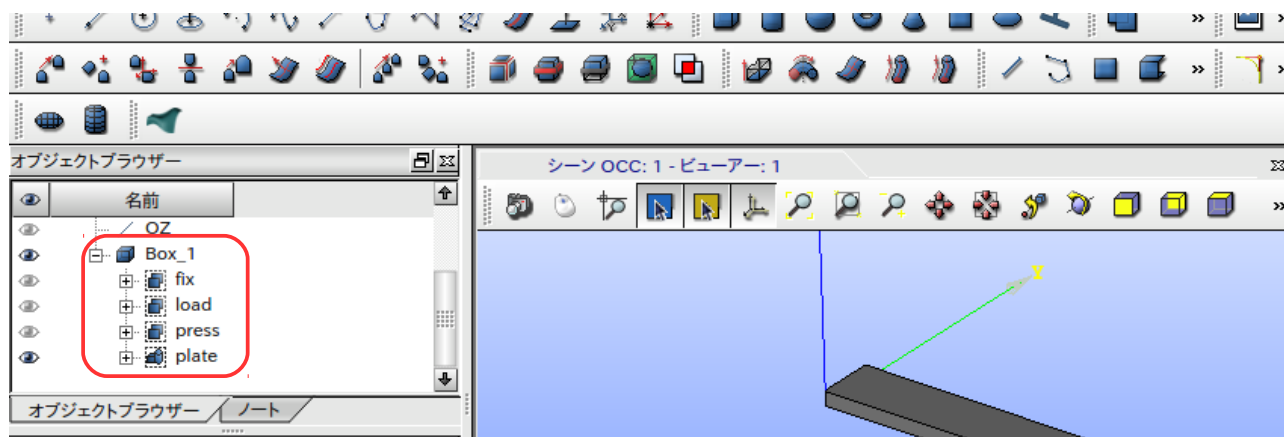
さらに、Volumeを選択して、モデル全体を選択して、volume「plate」を作成する。



以上の操作で、Geometry内で以下のグループが作成できた事になる。

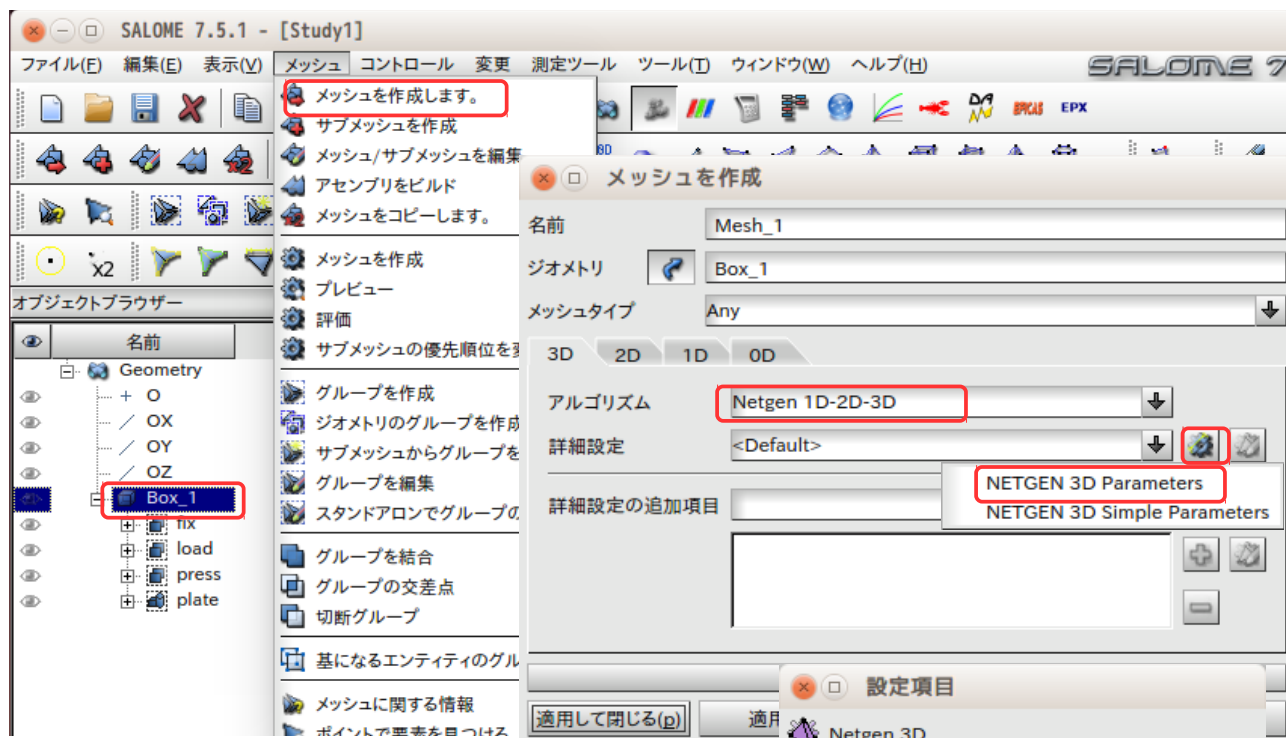
根本端面	fix	面グループ
先端端面	load	面グループ
上面	press	面グループ
全体	plate	volumeグループ

この段階で、オブジェクトブラウザーは、Box\_1の下に作成したグループが追加された状態になっている。下図参照。

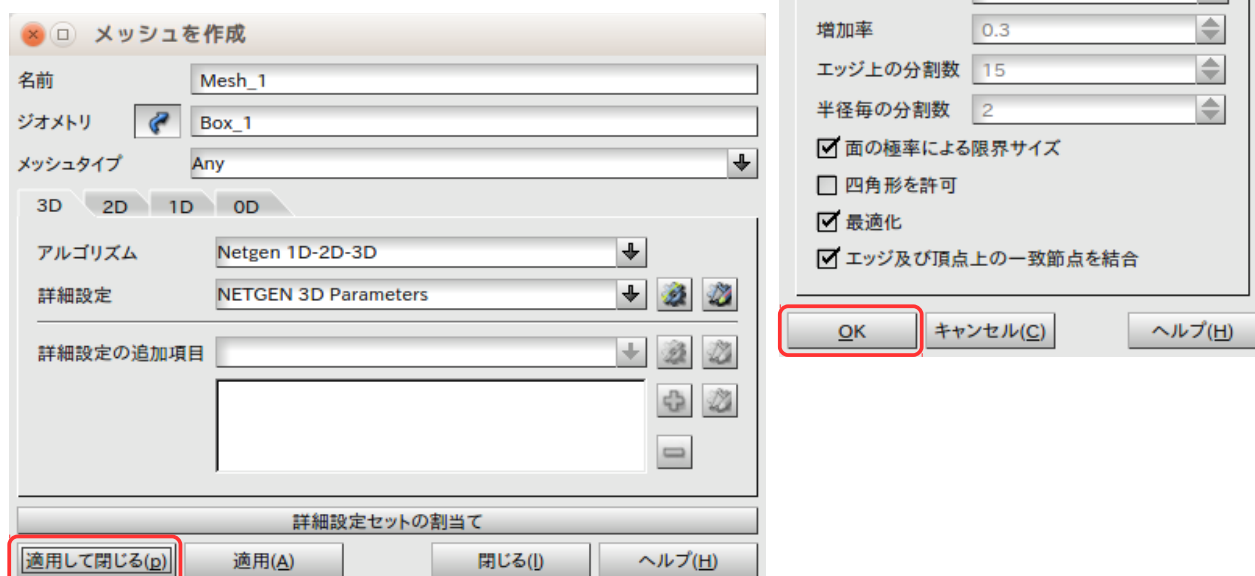


## 3-12-2. メッシュの作成

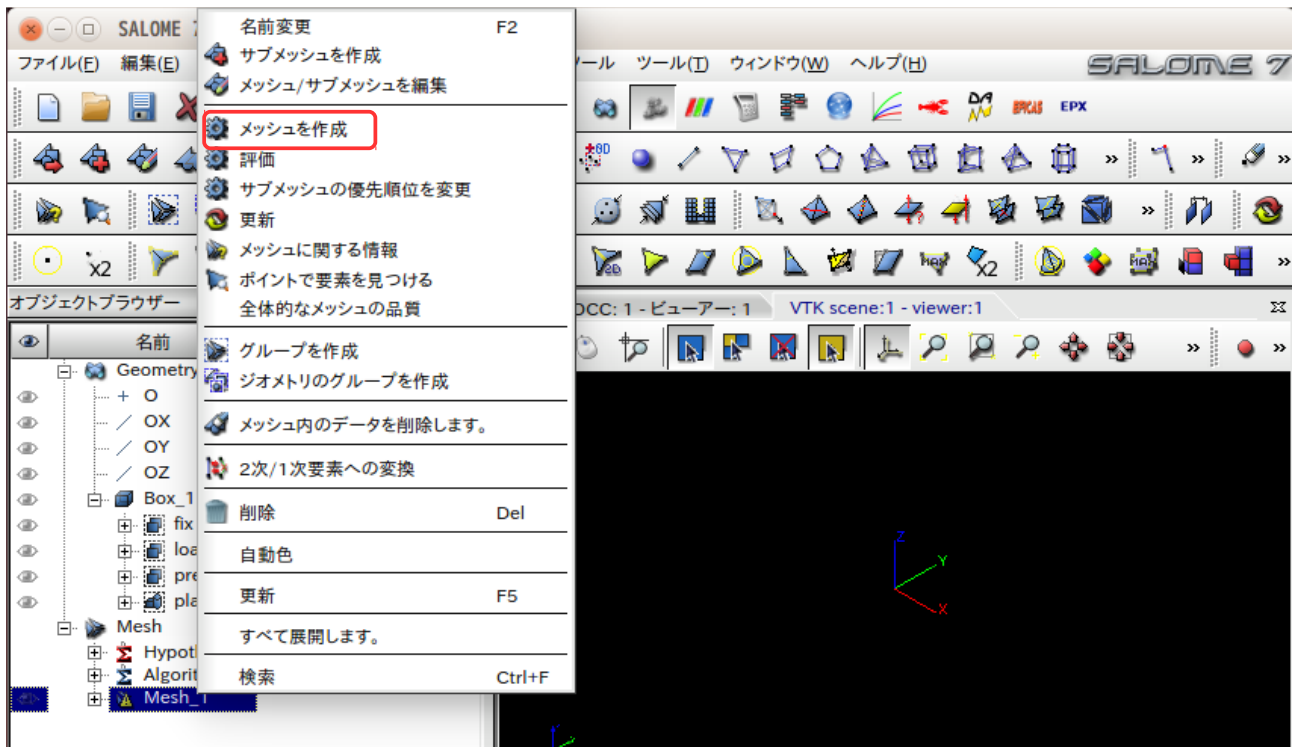
メッシュを作成する。まず、Mesh モジュールに変更する。  
オブジェクトブラウザ上の「Box\_1」を選択した状態で、メニューバー「メッシュ」>「メッシュを作成します。」を選択する。この後、アルゴリズムを「Netgen 1D-2D-3D」に設定し、ギアのマークをクリックして「NETGEN 3D Parameters」を選択する。



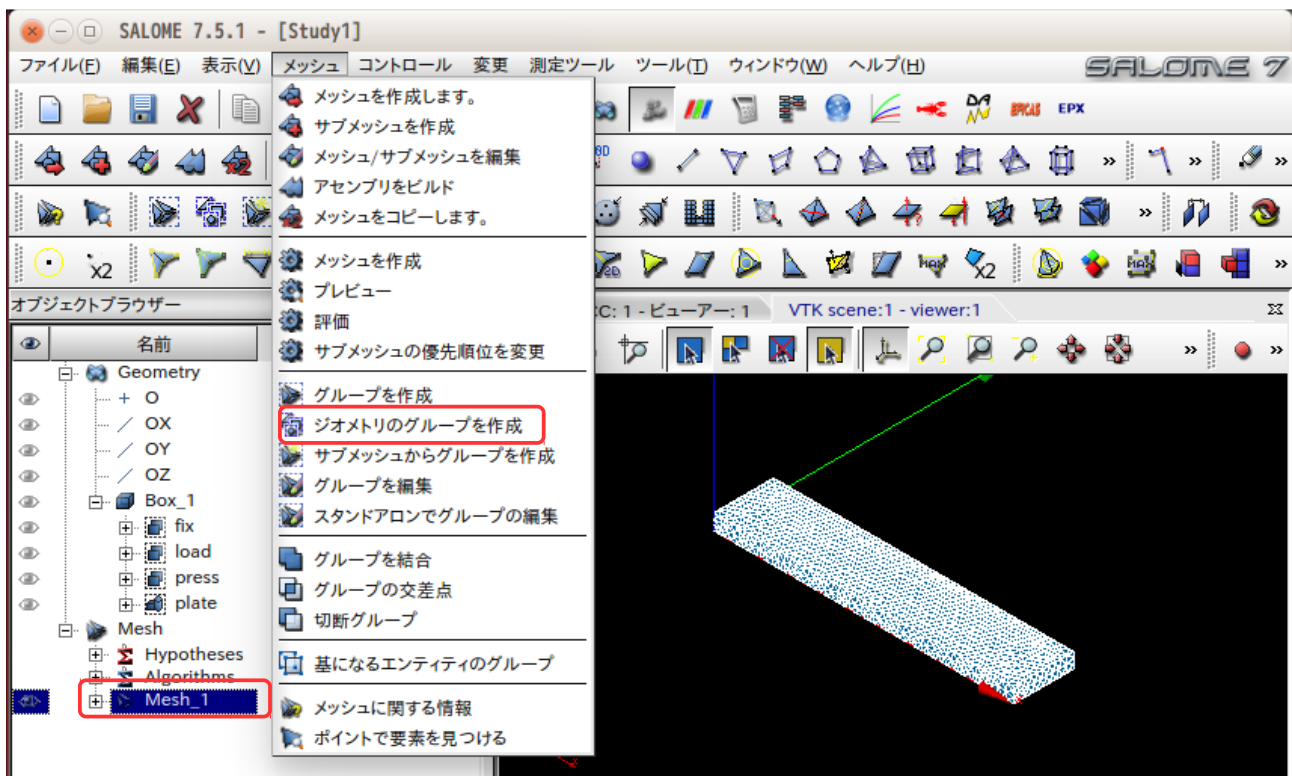
最大サイズを「2」、最小サイズ「1」に設定する。ここでメッシュ作成時のメッシュサイズを指定したことになる。設定後「OK」ボタンをクリックして閉じる。下図の画面に戻るので、「適用して閉じる」ボタンをクリックして、閉じる。



Salome のオブジェクトブラウザーが以下の画面に変わるので、ブラウザー上の「Mesh\_1」を選択後、右クリックして、「メッシュを作成」を選択する。これで、設定した条件でメッシュが作成される。

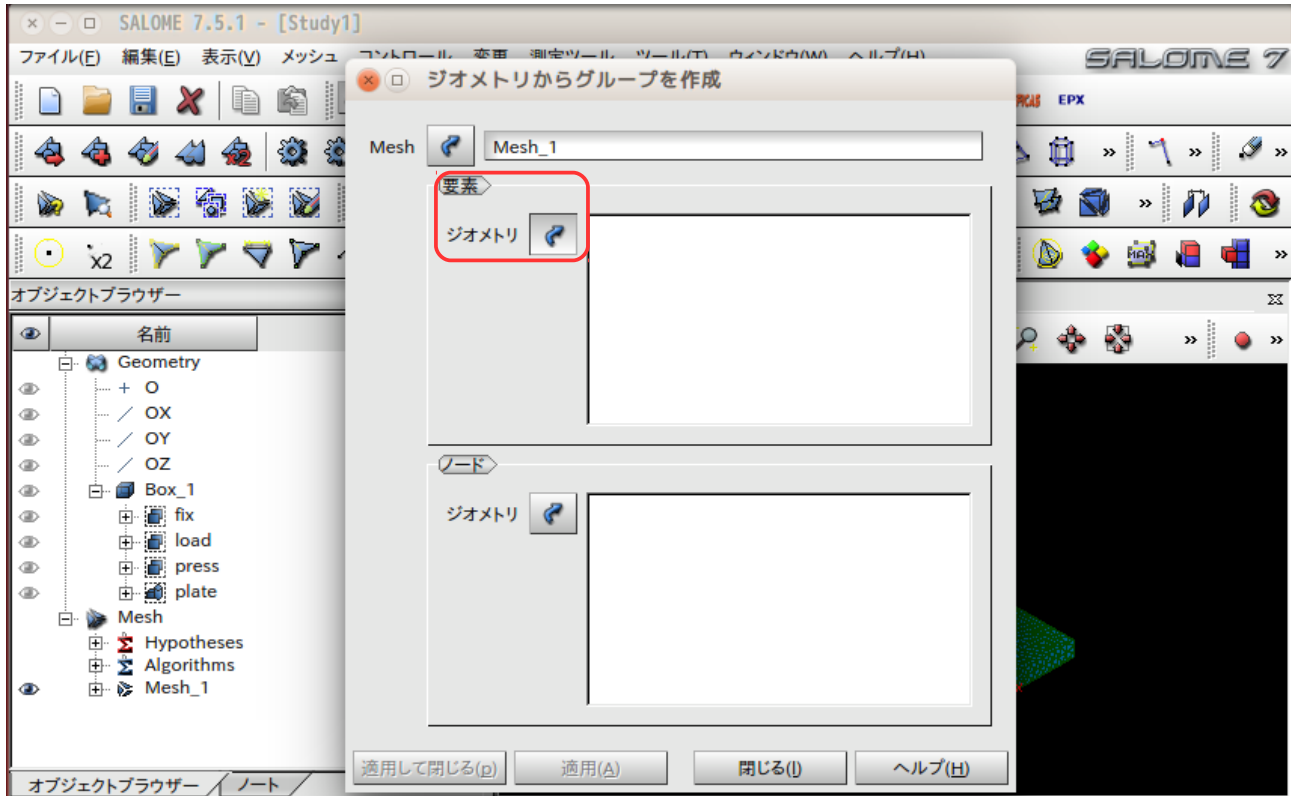


でき上がったメッシュに elementGroup、faceGroup、nodeGroup を作成する。「Mesh\_1」を選択後、メニュー「メッシュ」>「ジオメトリのグループ作成」を選択する。

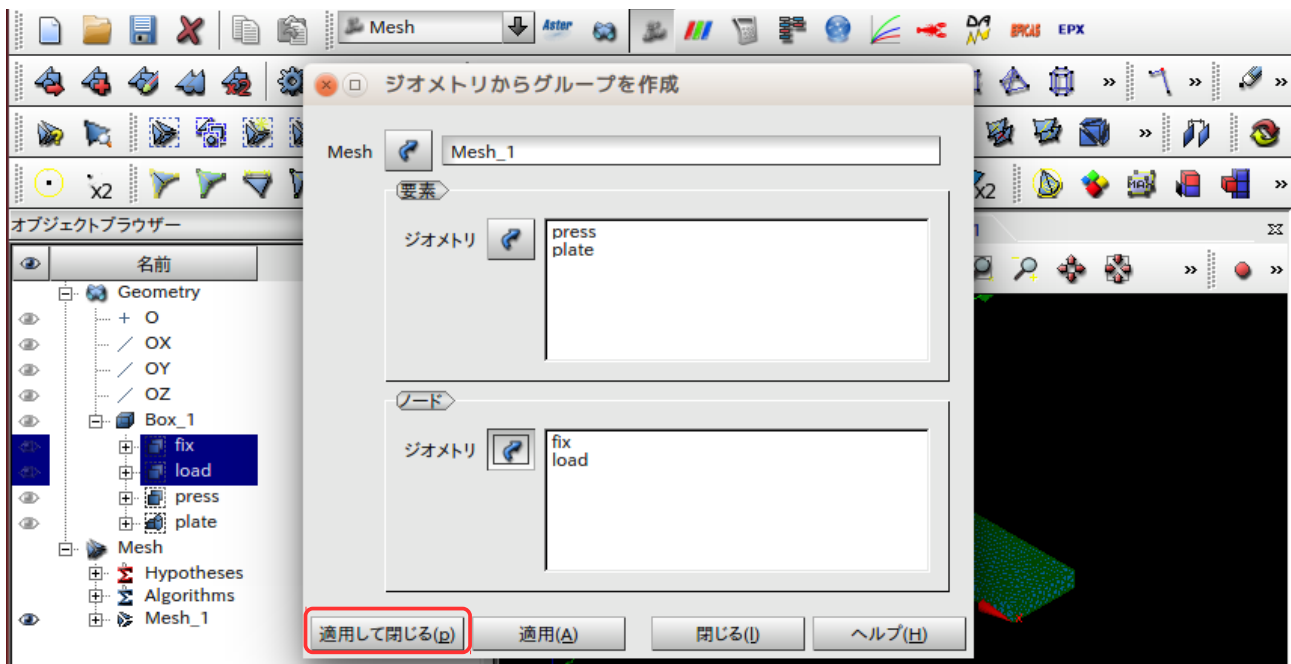


以下の画面上で各 Group を作成する。

まず、要素の「ジオメトリボタン」が ON 状態で、オブジェクトブラウザー上で、elementGroup, faceGroup にしたい Group 名 (plate、press) を Shift キーを押しながら選択する。同様にノードの「ジオメトリボタン」をクリックして、ON 状態に設定し、nodeGroup に設定したいグループ名を選択する。

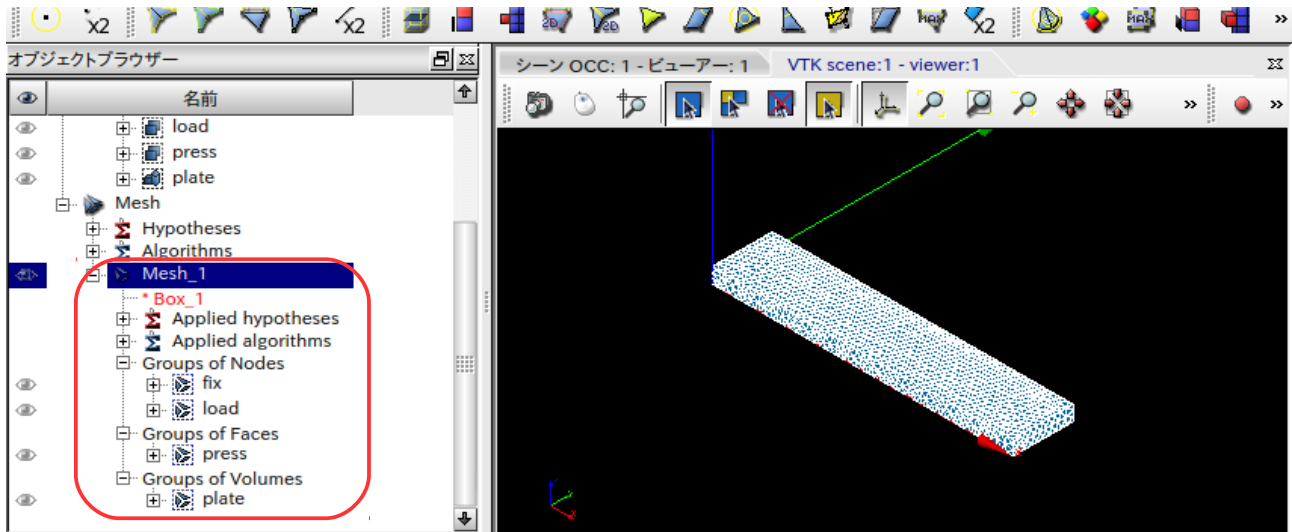


下図が選択した状態になる。この状態で、「適用して閉じる」ボタンをクリックして閉じる。



この状態で、オブジェクトブラウザーの内容は、以下になる。

「Mesh\_1」の下に、nodeGroup (fix、load)、faceGroup (press)、volumeGroup (plate) ができ上がっている。(volumeGroup が、elementGroup になる。)



グループ化された「Mesh\_1」を選択して、メニューバー「ファイル」>「エクスポート」>「UNV ファイル」を選択して、ファイル名を「plate.unv」として、保存する。

以上で今回のモデルの unv ファイルが完成した事になる。

### 3-13. 等分布節点荷重の算出方法

等分布節点荷重を算出する方法として、面、線分、点についてそれぞれ異なった方法で算出している。その算出方法を次項以降にて示す。

これらの面、線分、点については、選択した nodeGroup の状態を確認して、条件に合致したもの（面 or 線分 or 点）に等分布荷重を設定する。その合致する条件は、以下による。

- 面： nodeGroup 中で面積（要素の face）が取得できた場合、面に設定。
- 線分： nodeGroup 中で面積が取得できず、線分（要素の edge）が取得できた場合、線分に設定。
- 点： nodeGroup 中で、面積、線分とも取得できなかった場合、点に荷重を設定。

等分布荷重を面、線分、点のどれに設定したかは、モニタ画面に出力されるので、これで確認できる。

#### 3-13-1. 面の等分布節点荷重の算出

面に等分布荷重を設定する為には、選択した nodeGroup 中に面積（要素の face）が含まれている必要がある。もし、要素の face が含まれていない場合は、線分または点にその荷重が設定される。

面の等分布荷重としては、1次要素と2次要素では、異なっているので、各々について、説明する。



## 3-13-1-1. 1次要素の場合 (面)

1次要素の場合は、要素 face 面が三角形、四角形に関係なく、face 面の面積に相当する荷重を各節点に、均等に配分すれば済む。

face 面の面積は、各要素タイプ (四面体、六面体、五面体) の面の為、これら節点の座標を読み取ることができ、その座標から面積が算出できる。

要素の face 面が三角形の場合は、節点座標から3辺の長さを求め、これから面積を下式で算出している。face 面が四角形の場合は、対角線で2分割して三角形を2ヶ作り、これから算出している。

$$A = \sqrt{s(s-a)(s-b)(s-c)} \quad \begin{array}{l} a, b, c: \text{各辺の長さ} \\ A: \text{三角形の面積} \end{array}$$

$$s = \frac{a+b+c}{2}$$

各 face の面積を求めた後は、その面積に相当する荷重を各節点に均等に配分する事で、等分布節点荷重を定義する事ができる。その定義方法は、以下の様に、制御ファイル (FistrModel.cnt) 内の「!CLOAD」行に、節点番号毎に定義している。

「!CLOAD」行内に、新たに「forceType」と「value」コマンドを追加している。forceTypeは、CLOADの設定が3種類定義できるので、その定義している種類を表し、valueはその入力値を表している。

forceTypeが未定義の場合は、デフォルト値としてnodeForce (節点当たりの荷重) が設定される。

forceType:	nodeForce	節点当たりの荷重
	aveForce	トータル荷重
	trueForce	等分布トータル荷重
value:	x,y,z 方向の荷重	

----- FistrModel.cnt ファイルの内容 -----

```

:
#####
# Boundary Condition #
#####
!CLOAD, GRPID=1, forceType=trueForce, value=0.0, 0.0, 1000    追加したコマンド
load, 1, 0.0
load, 2, 0.0
load, 3, 0.0
1, 1, 0.0
1, 2, 0.0
1, 3, 0.535607916667
2, 1, 0.0
2, 2, 0.0
2, 3, 0.542549833333
3, 1, 0.0
3, 2, 0.0
3, 3, 2.06032719162
4, 1, 0.0
4, 2, 0.0
4, 3, 1.48456931751
5, 1, 0.0
5, 2, 0.0
5, 3, 1.0377386995
6, 1, 0.0

```

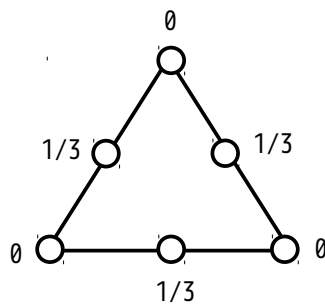
6, 2, 0.0  
 6, 3, 1.06654425747  
 7, 1, 0.0  
 7, 2, 0.0  
 7, 3, 1.11764805336  
 :

### 3-13-1-2. 2次要素の場合 (面)

2次要素の場合は、face 面が三角形と四角形とで、その主節点と中間節点に配分する係数が異なってくる。

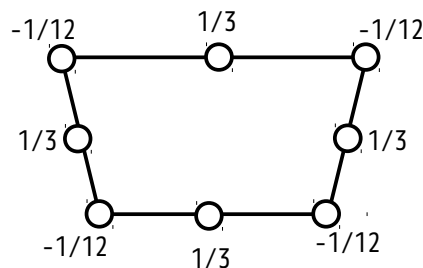
#### <三角形 2次要素>

三角形 2次要素 (四面体、五面体の 2 面) の場合、面積当たりの荷重 (面圧) を等分布荷重とする為に、下図の様に、主節点:0、中間節点:1/3 で配分している。



#### <四角形 2次要素>

四角形 2次要素 (六面体、五面体の 3 面) の場合、面積当たりの荷重 (面圧) を等分布荷重とする為に、下図の様に、主節点:-1/12、中間節点:1/3 で配分している。主節点が-1/12 の配分比率のため、中間節点の反対方向に荷重が働いている事になる。



これら主節点と中間節点への配分比率は、それぞれのモデルを作成し、変位拘束で変形させた時の反力を確認することでも、その配分比率を確認することができる。

2次要素の場合、主節点に反力がほとんど発生しない為、接触解析の場合には注意が必要になってくる。

主節点、中間節点の配分係数に基づいて、face 面の面積相当の荷重を各節点に配分する事で、等分布荷重を設定することができる。尚、face 面の面積の算出方法は、3-13-1-1 項と同様な方法で算出できる。

### 3-13-2. 線分の等分布節点荷重の算出

線分に等分布荷重を設定する為には、選択した nodeGroup 中に線分（要素の edge）を含んでおり、面積（要素の face）を含んでいない事が必要になる。

もし、面積（要素の face）が含まれていると、その面に等分布荷重が設定される。また線分（要素の edge）が含まれていない場合は、点に荷重が設定される。

線分に等分布荷重を設定する場合も、面と同様に 1 次と 2 次要素で設定が異なってくる。

#### 3-13-2-1. 1 次要素の場合（線分）

1 次要素の線分の場合、単位長さ当たりの荷重相当を節点に均等に配分することで、実現できる。

線分は、各要素（四面体、六面体、五面体）のエッジになっているので、各要素の節点番号から節点の座標が判るので、線分の長さが算出できる。

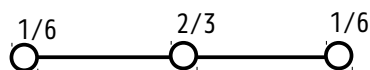
この線分の長さから、線分に印加される荷重が算出できるので、この荷重を節点に均等に配分する事で等分布荷重を設定する事ができる。

以上の方法で、求めた節点番号毎の荷重を制御ファイル (FistrModel.cnt) に定義する事になる。定義方法は、3-13-1-1 項と同じ方法で定義している。

#### 3-13-2-2. 2 次要素の場合（線分）

線分の場合、要素が四面体、六面体、五面体に関係なく、その線分の節点座標が判っていれば、以下の様に、主節点:1/6、中間節点:2/3として配分する事で、等分布荷重として設定できる。

求めた節点毎の荷重を制御ファイル (FistrModel.cnt) に定義する事になる。



### 3-13-3. 点の等分布節点荷重の算出

点に等分布荷重を設定する場合は、選択した nodeGroup 内に面（要素の face）と線分（要素の edge）が含まれていない事が必要になる。

もし、要素の face や edge が含まれていると、面や線分に等分布荷重が設定される。

nodeGroup 内の点は、要素の face や edge を含んでいない為、互いに独立している。これら独立した点に等分布荷重は設定できないので、等分布荷重として、入力した「等分布トータル荷重」の値を、節点数で除した値を各節点に設定している。

この設定方法は、「トータル荷重」と同じ設定方法になる。

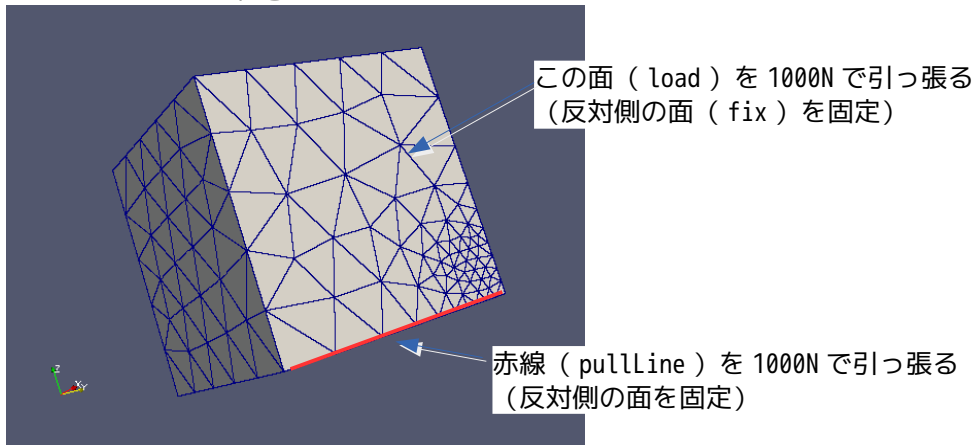
### 3-13-4. 等分布トータル荷重の設定結果の確認

ここで、等分布荷重で計算した結果を確認してみる。

モデル形状は、 $\square 10\text{m}$ の立方体の形状で、メッシュは、四面体で要素の粗密のあるメッシュ（1次と2次要素）で確認してみる。

確認は、片面（fix）を固定して、反対側の面（load）と線分（pullLine）を等分布荷重で引っ張る設定で確認してみる。材料は、Steelで確認する。

メッシュの状態



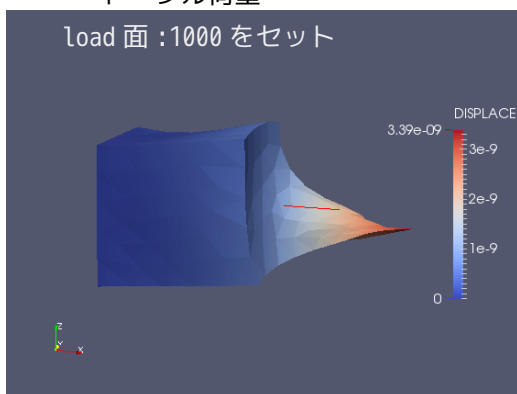
通常の設定では、要素サイズが異なっている部分がある為、細かい要素部分では、荷重密度が大きくなる。この為、変形状が均一でなくなってしまう。

これに対して、「等分布トータル荷重」の設定で引っ張った場合は、メッシュサイズの粗密に関係なくなる為、変形状が均一になるはずなので、これを1次と2次要素で確認してみる。

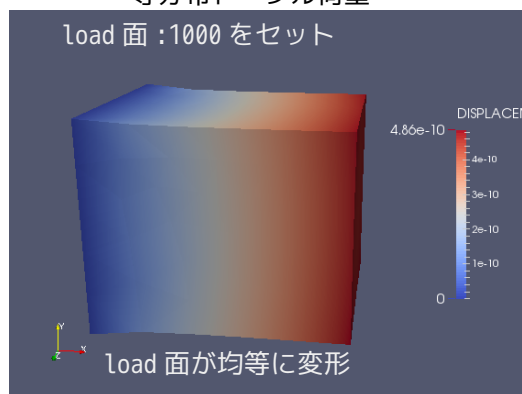
これを確認した結果、以下に示してある様に、「トータル荷重」で 1000N をセットすると、荷重分布がメッシュサイズに依存している為、細かいメッシュ部分（荷重密度が大きい）は、荷重が大きくなるので、変形が大きくなってしまっている。

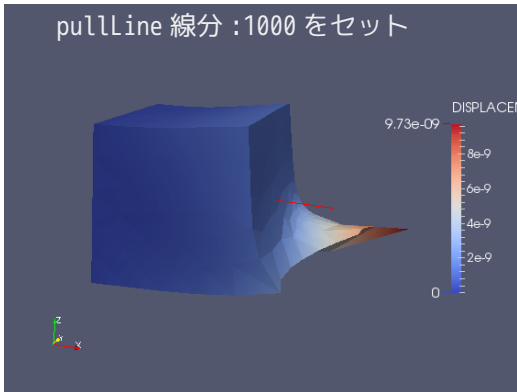
これに対し、「等分布トータル荷重」で同じ 1000N をセットすると、等分布荷重がセットされている為、均等に変形している様子が判る。

<四面体1次要素の場合>  
トータル荷重

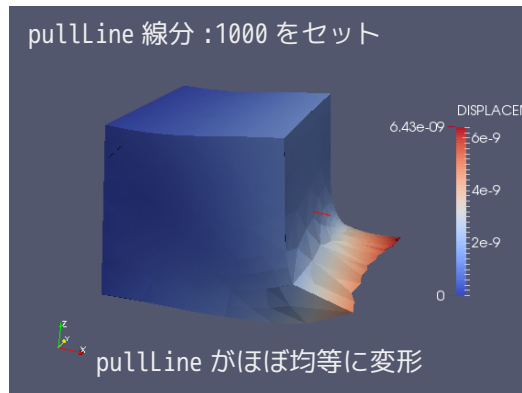
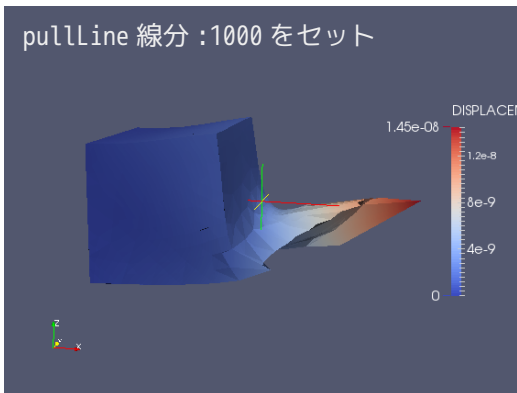
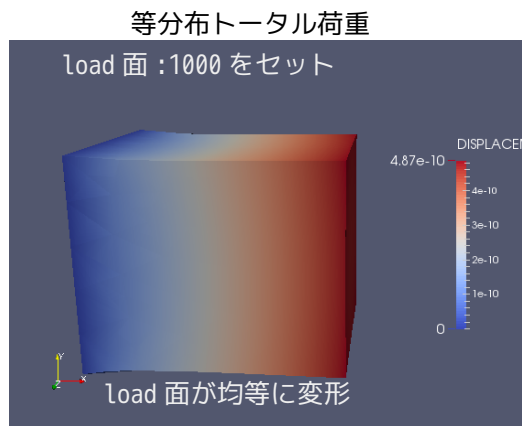
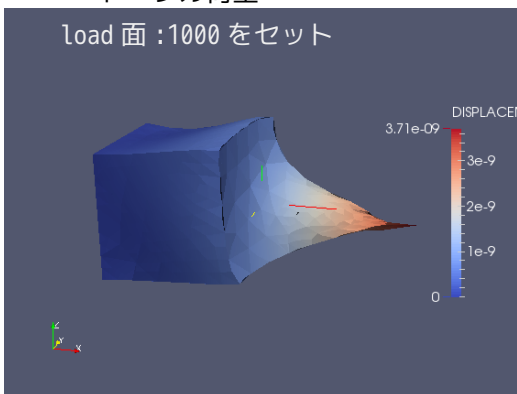


等分布トータル荷重





<四面体2次要素の場合>  
トータル荷重



## 4. 応用事例

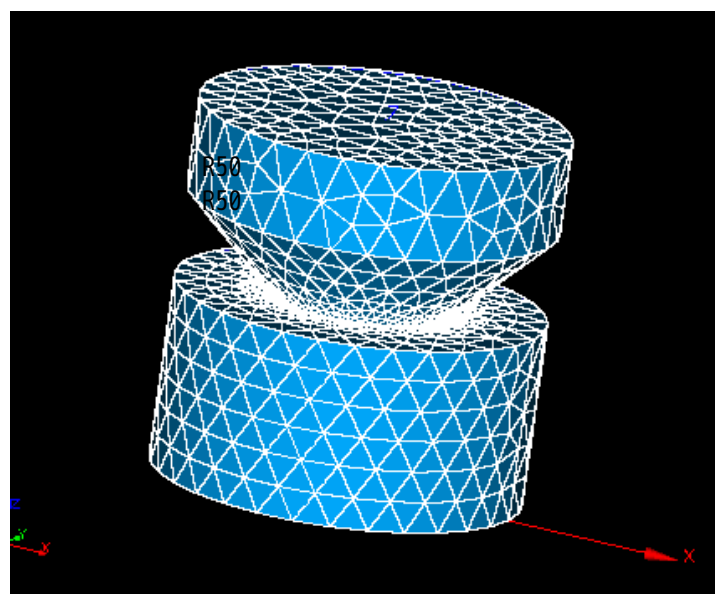
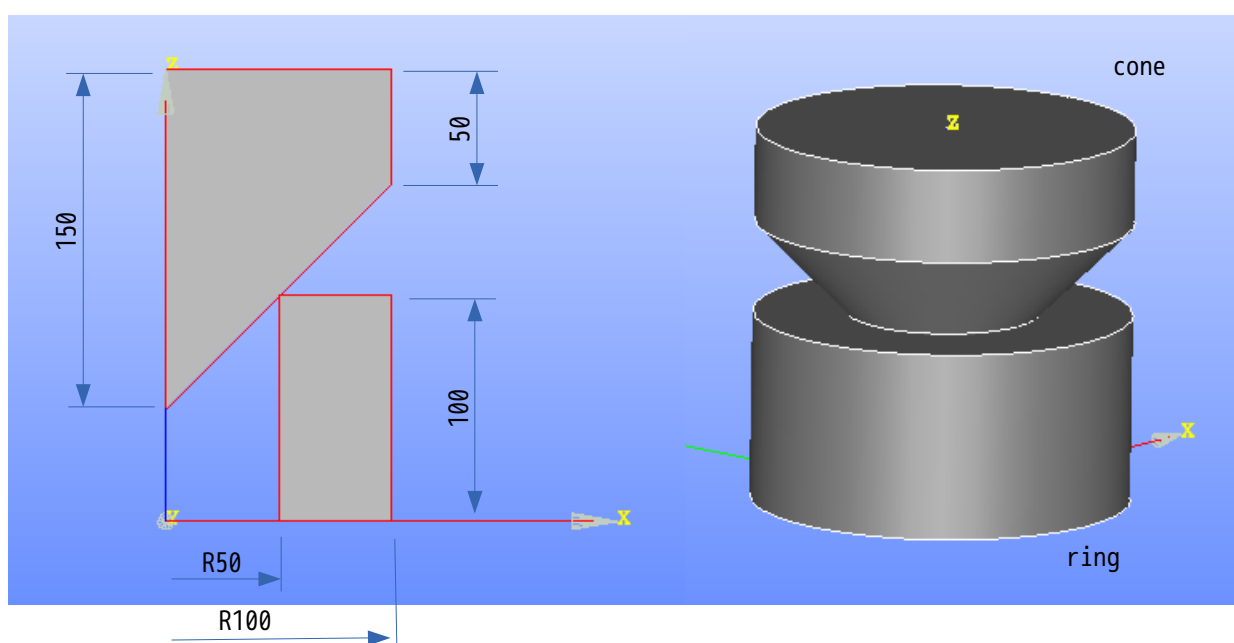
前項で弾性静解析の事例を取り扱った。ここでは、それ以外の事例について、その操作方法を説明する。

### 4-1. 接触解析

接触解析を EasyISTR で解析する事例について取り上げる。  
workFolder は、「CAE-fistr/Case/ringContact」を作成して解析する。

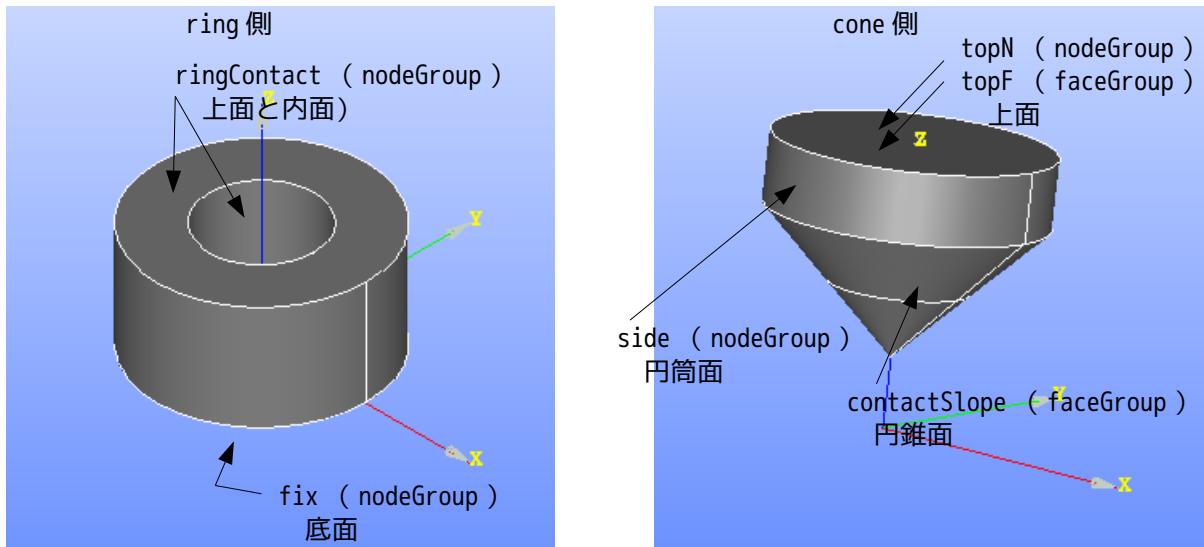
#### 4-1-1. モデル形状

モデルは、以下の形状を考える。salome で以下の形状でメッシュを作成している。



メッシュサイズ  
全体： netgen-1D-2D-3D  
20mm  
接触部： local length  
5mm

メッシュのグループ化は、以下で実施。



このメッシュを ringCone.unv として、workFolder 内に保存。  
 解析は、fix を固定して、cone の上面を変位させる変位拘束解析と cone 上面に圧力をかける圧力拘束の2種類の解析を行ってみる。

#### 4-1-2. 変位拘束の接触解析

##### 4-1-2-1. メッシュ変換

下図が、メッシュ変換して、1/1000 に scale 変更した結果になる。各々の group が下記のようにナンバリングされている為、paraView でその形状、位置が確認できる。

```
メッシュ内容
modelSize(xyz): 0.2 0.19982263 0.2

nodes 2032
elements type:341 3149
elements type:341 3896

EGRP ring 3149
EGRP cone 3896
SGRP otherS 1690
SGRP topF 175
SGRP contactSlope 1177
NGRP fix 94
NGRP ringContact 537
NGRP topN 104
NGRP side 107
```

- EGRP (elementGroup)
  1. ring
  2. cone
- SGRP (faceGroup)
  1. otherS
  2. topF
  3. contactSlope
- NGRP (nodeGroup)
  1. fix
  2. ringContact
  3. topN
  4. side

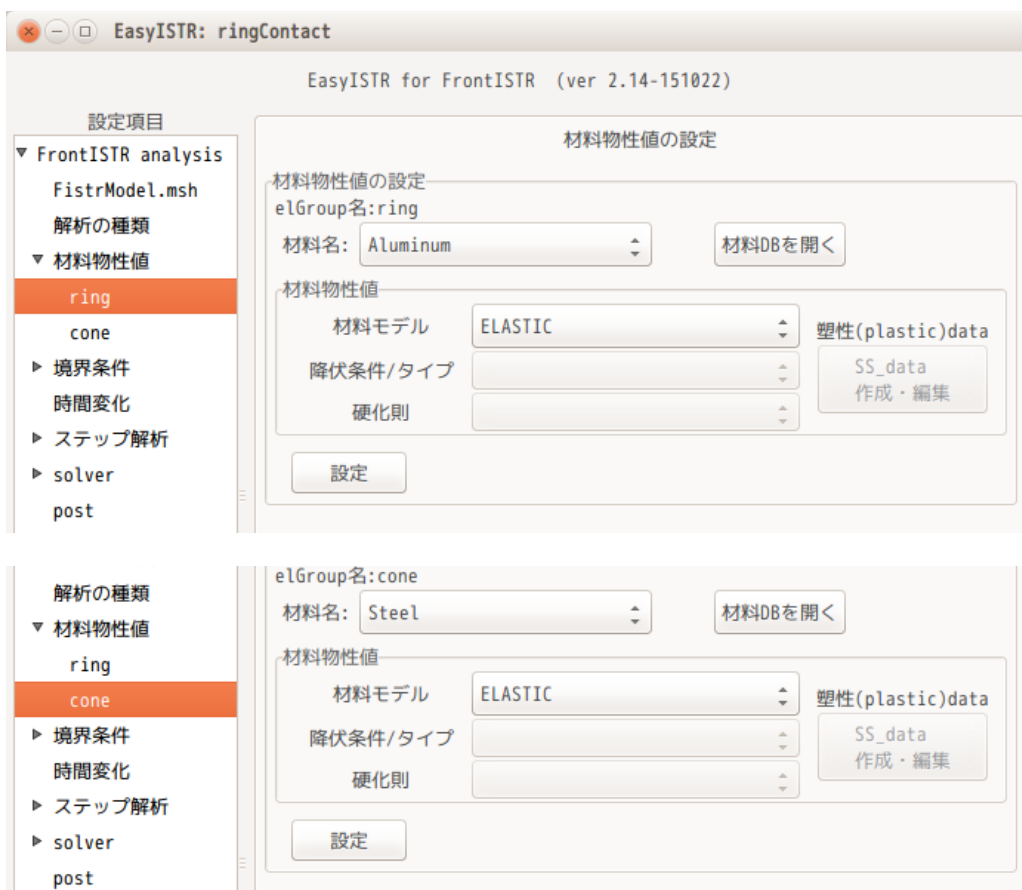
#### 4-1-2-2. 解析の種類

下図の様に、「非線形静解析」を選択し、「設定」ボタンをクリックして、設定する。



#### 4-1-2-3. 材料物性値の設定

ringとconeに各々AluminumとSteelを設定する。下図参照。





#### 4-1-2-4. 境界条件の設定

境界条件は、ring 底面の fix を固定し、cone 上面の topN を 2mm (-0.002) 下げる。  
下図の様に設定する。



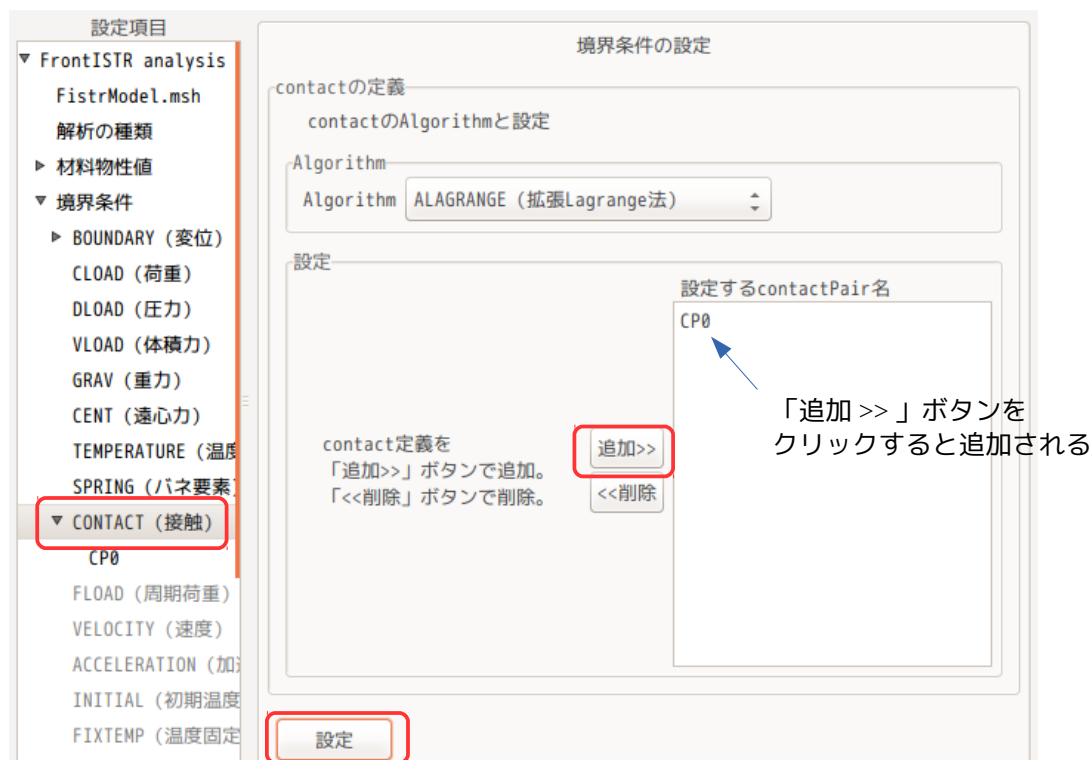
#### 4-1-2-5. 接触の設定

ここで、接触の設定を行う。この内容が接触解析特有の設定項目になる。

EasyISTR の設定項目 Tree 上の「境界条件」>「CONTACT (接触)」を選択する。

この画面上で、Algorithmを確認する。今の設定は、「拡張 Lagrange 法」が設定されている。  
Algorithmは、「ALAGRANGE (拡張 Lagrange 法)」と「SLAGRANGE (Lagrange 乗数法)」が選択できるが、ここでは、「拡張 Lagrange 法」を選択した。

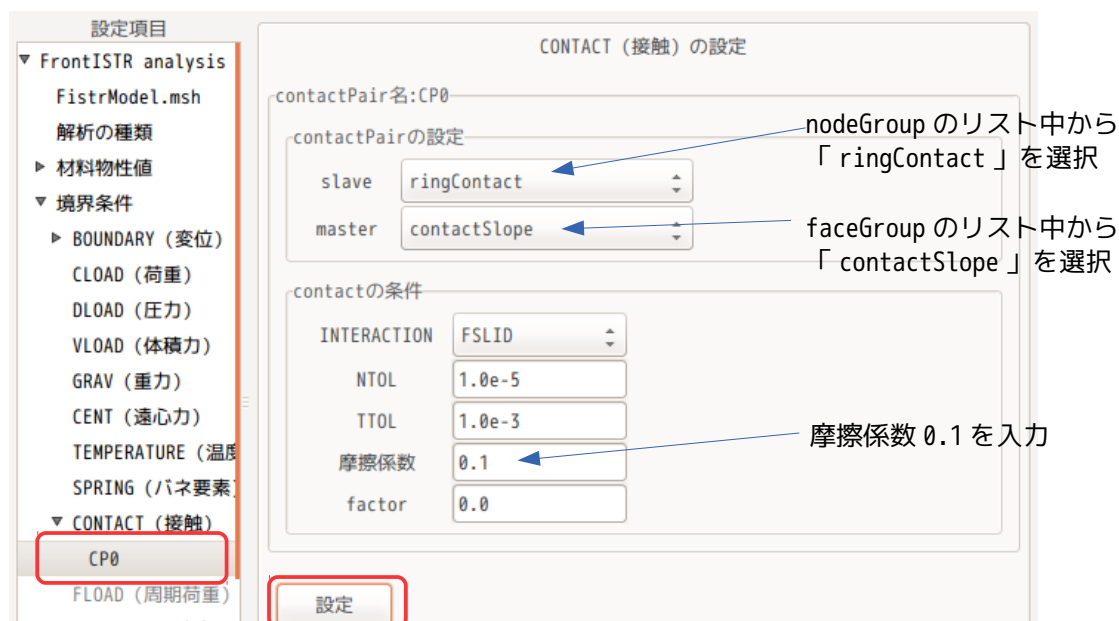
この後、「追加>>」ボタンをクリックし、「設定」ボタンをクリックして、「CP0」を Tree 上に追加する。  
(下図参照。)



この後、設定項目 Tree 上の「CP0」を選択して、接触の内容を設定する。（下図参照）  
contactPair の設定は、

slave : nodeGroup (点)  
master : faceGroup (面)

であり、点と面の接触を定義する。  
設定後「設定」ボタンをクリックして、内容を確定させる。

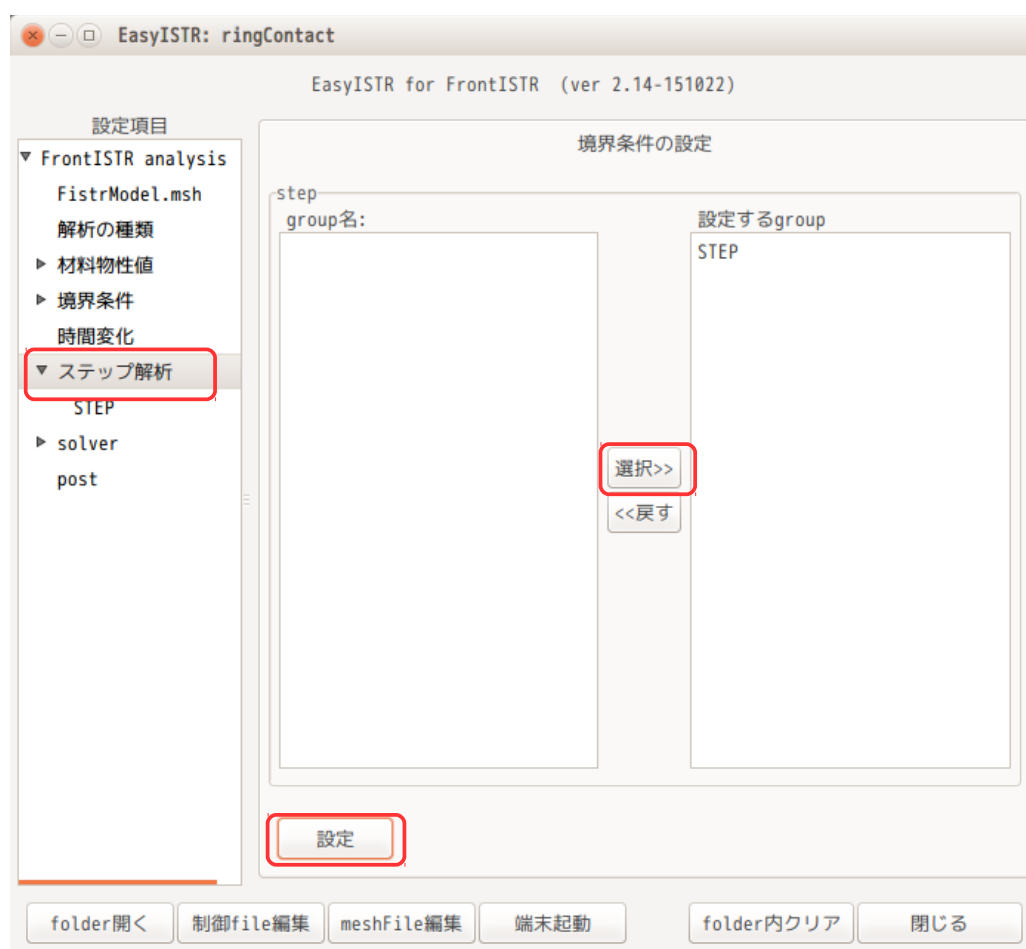


#### 4-1-2-6. ステップ解析の設定

非線形解析のため、ステップ解析が必要になる。(線形解析では、設定不要。)

設定方法は、設定項目 Tree 上の「ステップ解析」を選択し、「STEP」を選択、「選択>>」ボタンをクリックして、設定する group 側に「STEP」を移動する。

この後、「設定」ボタンをクリックして、設定を確定させる。確定すると、設定項目 Tree 内に「STEP」が追加される。



ここまでで、ステップ解析を設定する為の準備ができたことになる。

次に、ステップ解析の内容を設定する。

設定項目 Tree 内の「STEP」を選択する。下図が「STEP」選択直後の内容になる。



上図に対して、下図の様に修正している。修正後「設定」ボタンをクリックして、内容を確定する。

ステップ解析する境界条件は、ここでは、接触解析に関連する条件を選択する。

SubStep を「5」に設定している為、5分割して計算していく事になる。



## 4-1-2-7. 計算開始、結果の確認

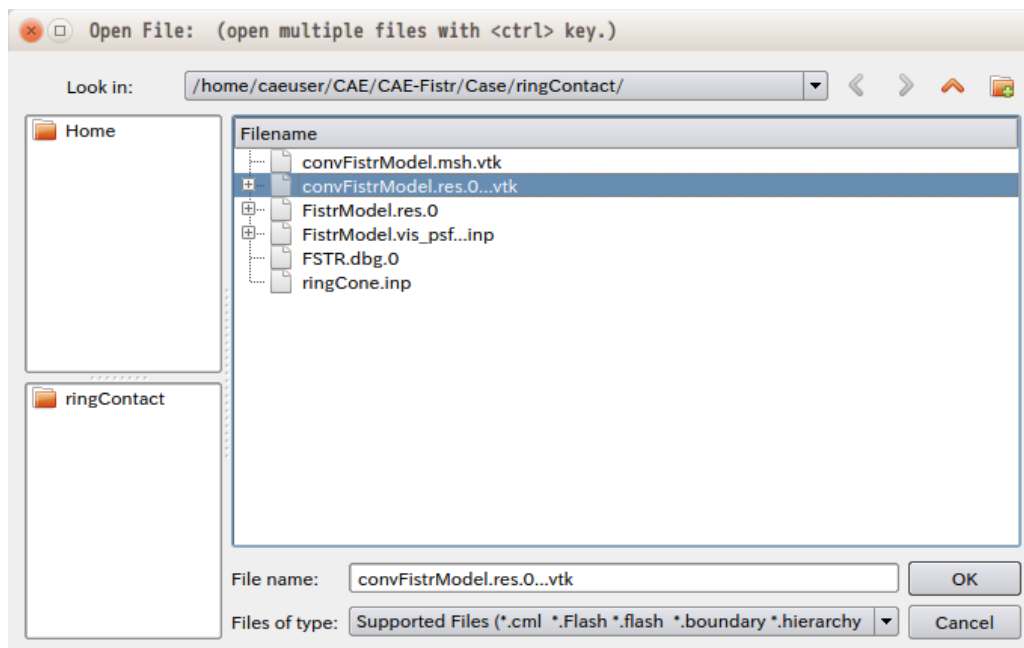
以上で設定が全て終了したので、計算を開始する。

計算開始は、設定項目 Tree 上の「solver」を選択し、「FrontISTR 実行」ボタンをクリックして、計算開始させる。

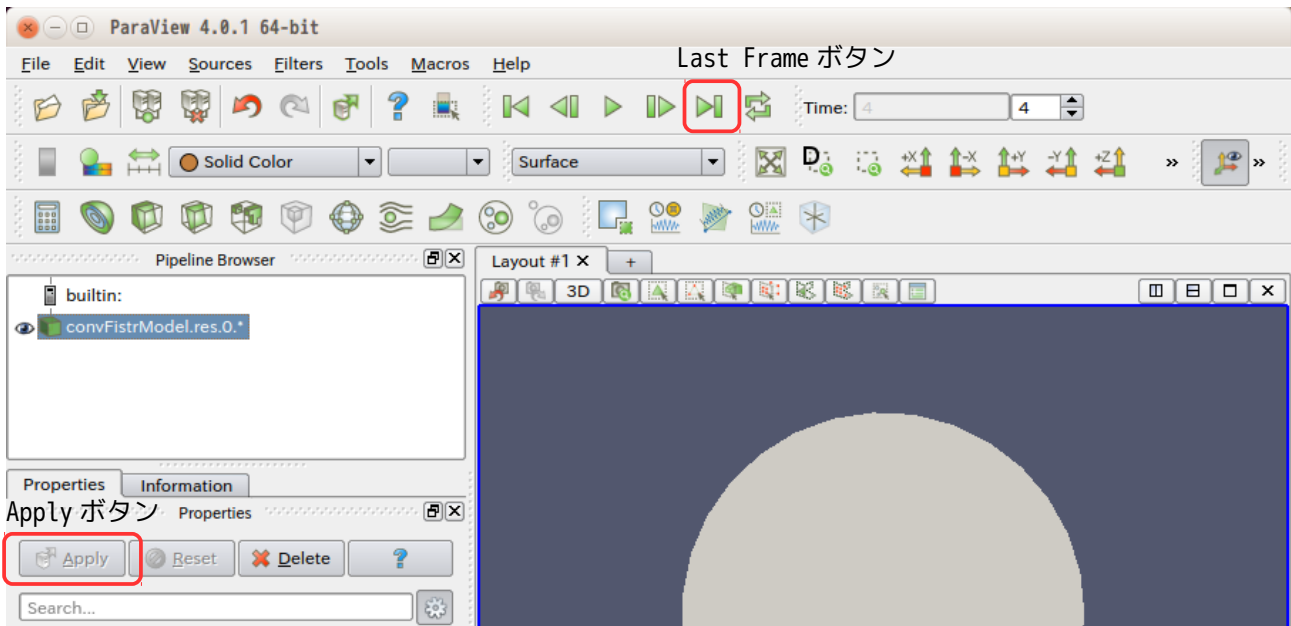


計算終了後、設定項目 Tree 内の「post」を選択し、「データ変換」、「主応力追加」、「ParaView 起動」ボタンをクリックして、結果を確認する。

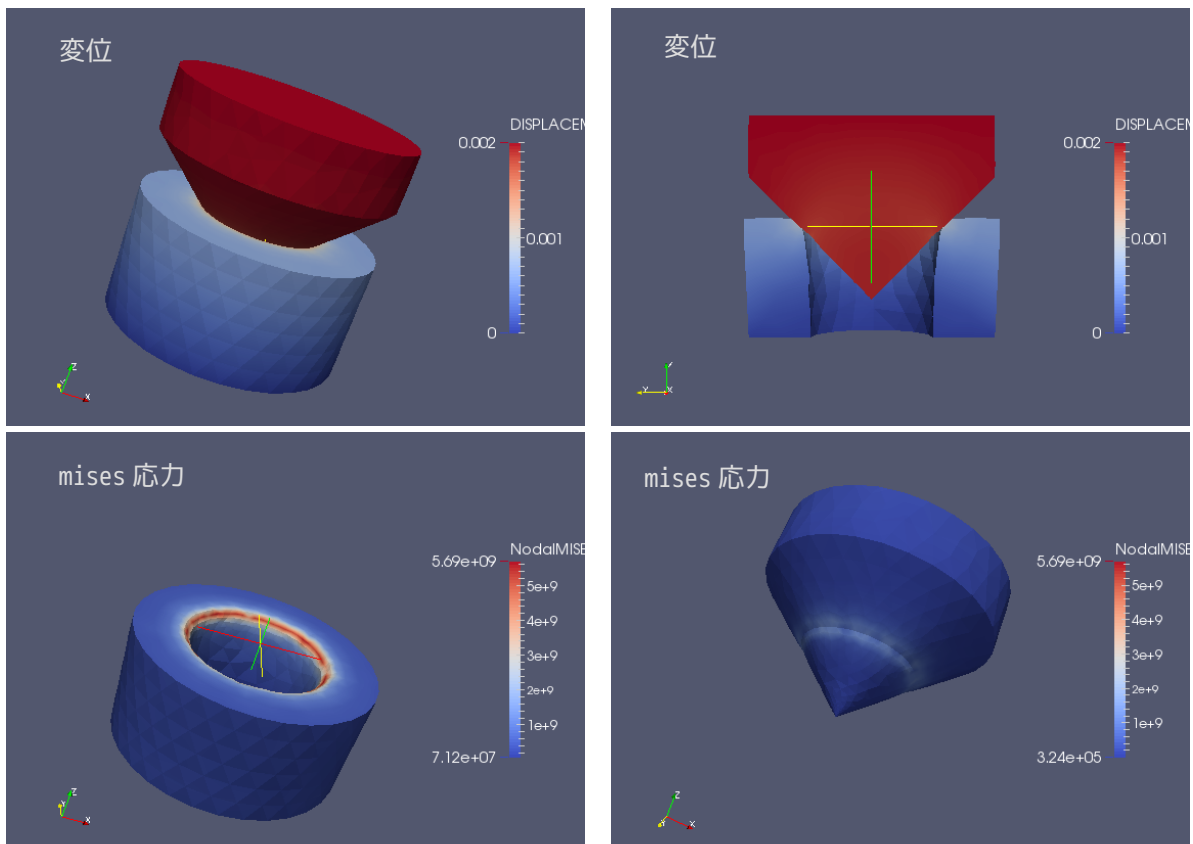
paraView 起動後、結果ファイルは、下図の様に、「convFistrModel.res.0...vtk」を選択する。ステップ解析で「SUBSTEP:5」と設定し、実行時に「結果出力頻度:1」と設定している為、結果ファイルは、5ヶ存在している。読み込むファイルを「convFistrModel.res.0...vtk」と選択したことで、これら5ヶのファイル全てを選択している。



ファイル読み込み後、「Apply」ボタンをクリックし、「Last Frame」ボタンをクリックして、最終の結果まで送っておく。



この後は、線形解析と同様にして結果を確認する。下図が確認した結果になる。



### 4-1-3. 荷重（圧力）拘束の接触解析：弱いばね追加

変位拘束の場合は、全ての自由度が拘束されている為、剛体移動が発生せず、素直に解析できる。しかし、荷重（圧力）拘束の場合は、変位が拘束されておらず、剛体移動が発生し、収束が難しくなってしまう。このような場合、弱いバネを追加して、剛体移動を防ぐ方法がある。

前項と同じモデルを使って、cone 上面の Z 方向に荷重（圧力）をかける解析を考えてみる。

ring の方は、ring 底面 (fix) を固定しているの、剛体移動は発生しないが、cone は、荷重（圧力）のみの設定のため、そのままでは、剛体移動が発生してしまう。このような場合、cone の外側円筒面 (side) に弱いバネを追加する事で剛体移動を防ぐ事ができる。

#### 4-1-3-1. 境界条件の設定

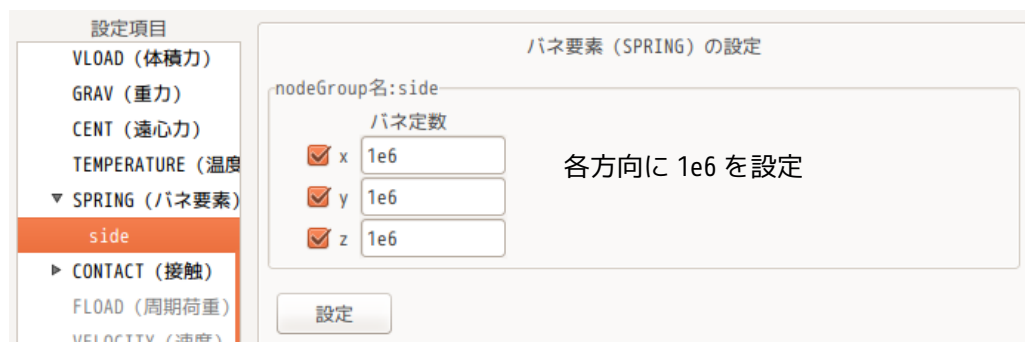
前記した様に、拘束すれば良い為、以下の様に拘束する。

nodeGroup 名	部位	拘束内容
fix	ring 底面	XYZ 方向拘束（固定）
side	cone 外側円筒面	弱いバネ追加
faceGroup	部位	拘束内容
topF	cone 上面	圧力を設定（Z 方向）

ring 底面 (fix) に変位拘束（固定）を設定する



cone 外側円筒面 (side) に弱いバネを追加する。ばね定数は、最も弱いAluminumのヤング率が70.0e9の為、1e6とし、これを各方向に設定した。（ヤング率の値とかけ離れすぎると、収束し難くなる。）

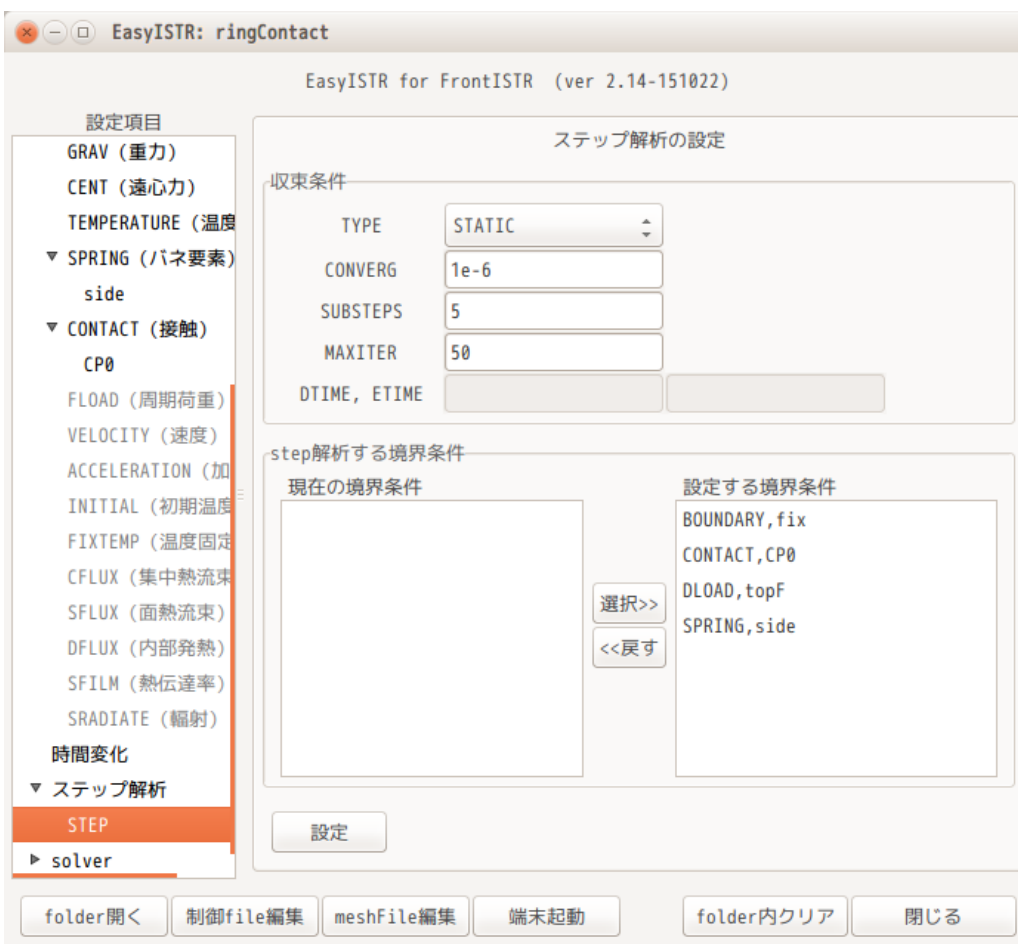


圧力は、topF に 100e6 Pa (3.14e6 N) をかける事にする。



#### 4-1-3-2. ステップ解析設定

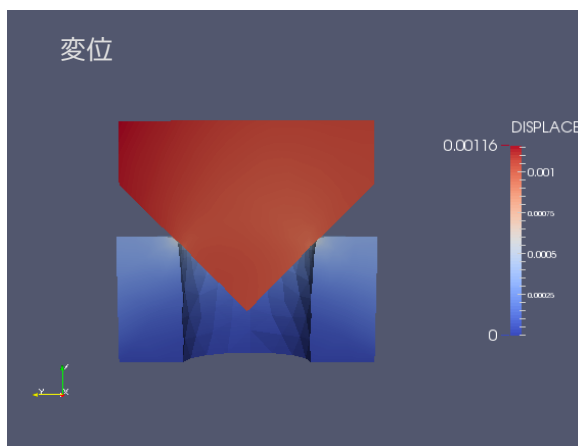
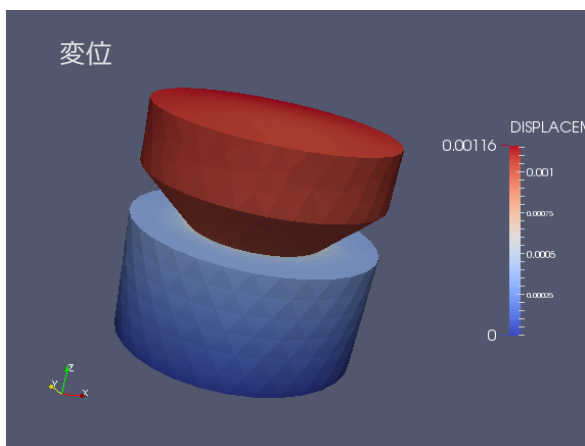
以下のように設定している。



#### 4-1-3-3. 計算開始、結果の確認

計算は、設定項目 Tree 内の「solver」を選択し、「FrontISTR 実行」ボタンをクリックして、計算させる。計算結果は、以下の様に確認できる。(剛体移動が発生せず、うまく計算できている。)



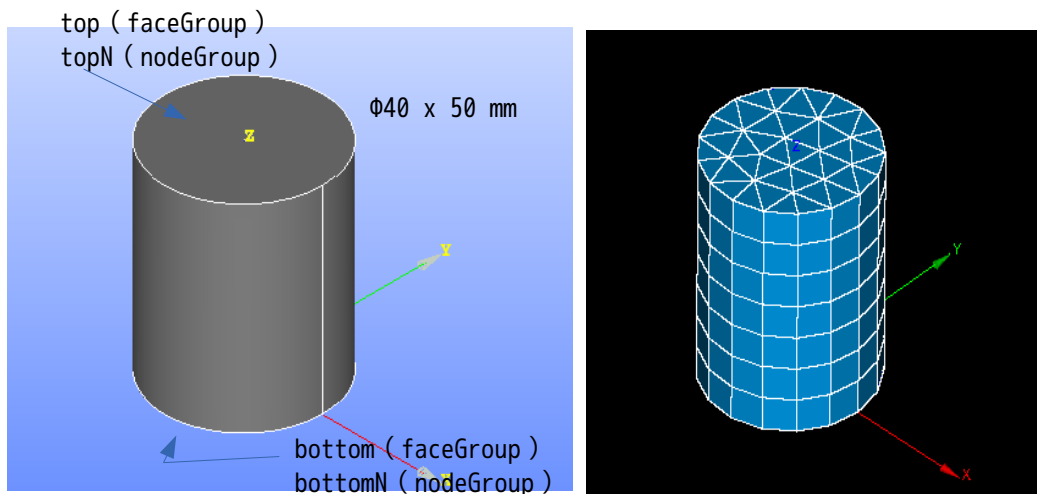


## 4-2. 弾塑性解析

EasyISTR 上で、弾塑性解析を行ってみる。  
 モデル形状は、円柱形状とし、圧縮した後、引張を行ってみる。  
 workFolder は、「CAE-fistr/Case/plasticPushPull」で解析する。

### 4-2-1. モデル形状 (円柱)

下記形状のモデル (φ40 x 50 mm) を Salome 上で作成した。



メッシュは、軸方向に圧縮、引張を行う為、prismで作成している。  
 Salomeでprismを作成する方法は、底面 (bottom面) のメッシュを押し出して、prismを作成している。以下の設定で、メッシュを作成した。

<全体>	< SubMesh >
3D: 3D Extrusion	2D: Netgen 2D
2D: Quadrangle(Mapping)	max size 7.5
1D: Wire Discretisation	1D: Wire Discretisation
Local Length	Local Length
7.5	7.5

でき上がったメッシュを「block.unv」として、workFolder内に保存する。

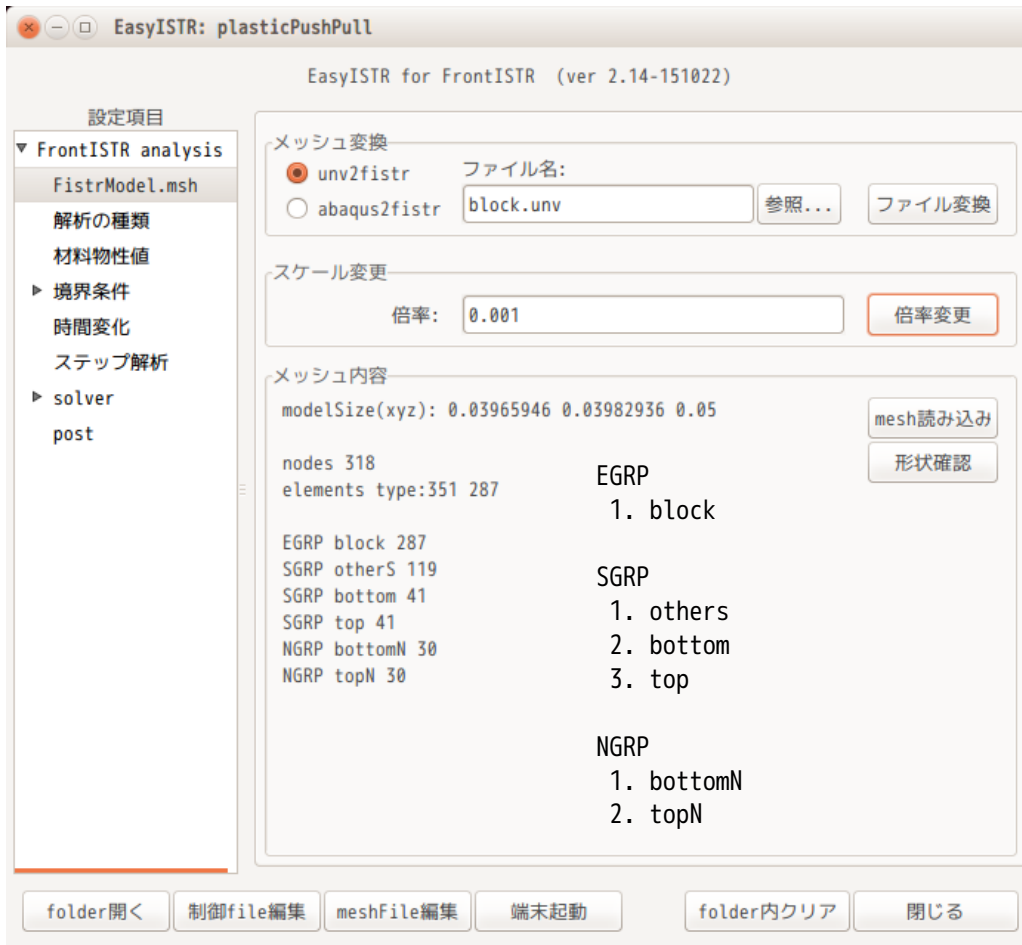
### 4-2-2. 圧縮20%の計算

モデルの底面 (bottomN) を固定し、上面 (topN) を20% (10mm) 圧縮する。

#### 4-2-2-1. メッシュ変換

メッシュ変換の為に、workFolder内の「block.unv」を選択して、メッシュ変換し、モデルの倍率を0.001にしておく。下図が変換した状態になる。

elementType が「351」と表示されており、prism1次メッシュと認識されている。



#### 4-2-2-2. 解析の種類

解析は、「非線形静解析」を選択する。(下図参照)

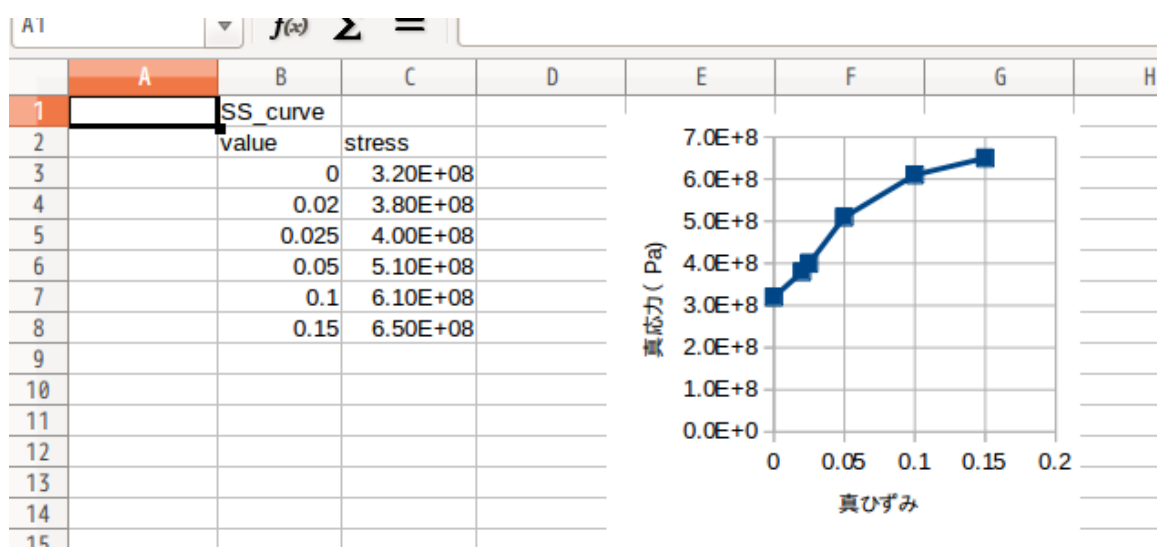


### 4-2-2-3. 材料物性値の設定

材料は、Steel で材料モデルは、以下で設定。

材料モデル： PLASTIC 塑性  
 降伏条件/タイプ： MISES  
 硬化則： MULTILINEAR 多曲線近似

多曲線近似の為、応力ひずみ線図が必要になる。「SS\_data 作成・編集」ボタンをクリックして、office を起動し、データを入力する。入力後は、csv 形式で保存しておく。  
 ファイル名は、「Steel\_PlasticSSdata.csv」として材料名を含むファイル名が自動で作成され、workFolder 内に保存される。この為、一度データを作成すると、同じ材料名 Steel であれば、そのファイルが開く。



### 4-2-2-4. 境界条件の設定

境界条件は、以下で設定する。

nodeGroup	部位	設定内容
-----------	----	------

bottomN	底面	固定	
topN	上面	-10mm (-0.010)	全長の20%圧縮する

この為、下図の様に設定した。



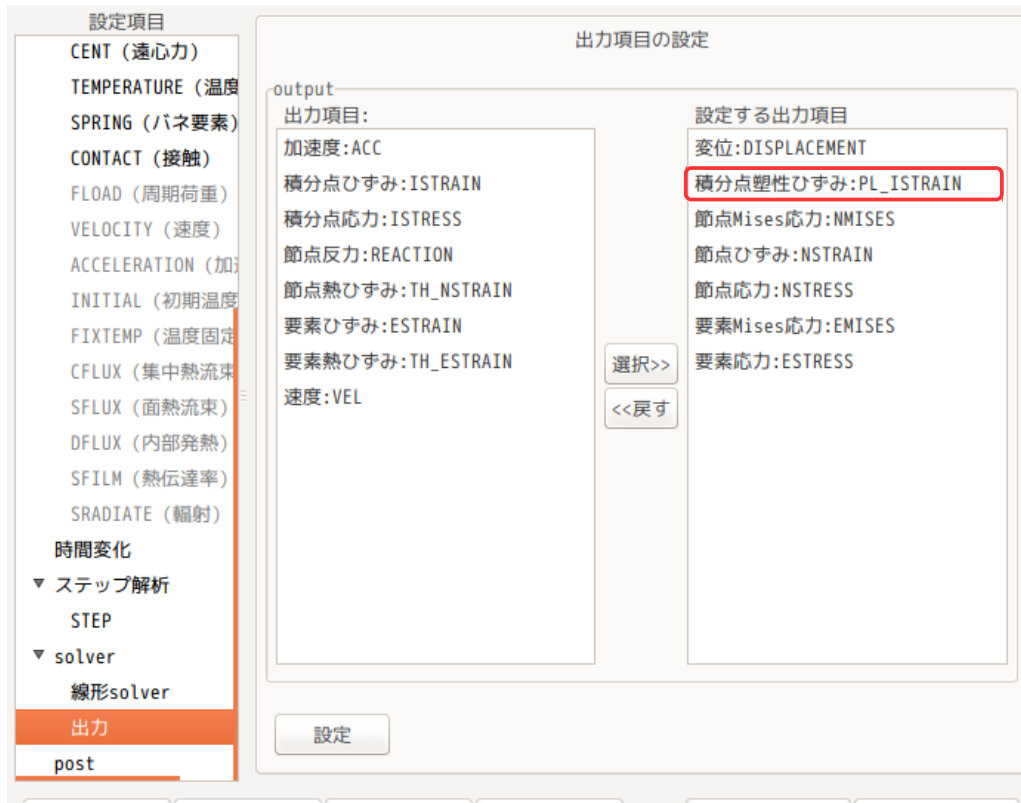
#### 4-2-2-5. ステップ解析の設定

SubStepを「10」、MAXITERを多めの「500」に設定した。(下図参照)  
SubStepが10の為、0.001/1stepの変化で計算していくことになる。



## 4-2-2-6. 計算開始、結果の確認

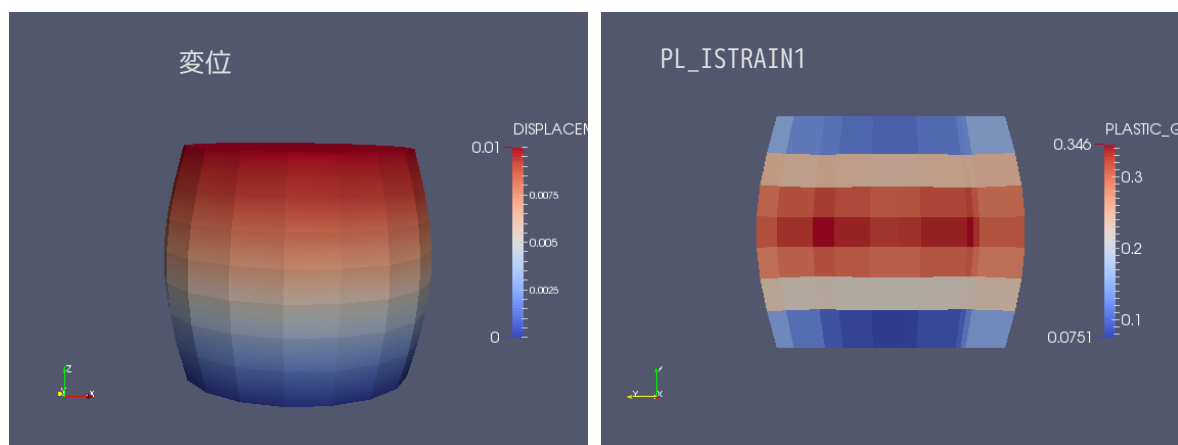
計算結果の出力に、「積分点塑性ひずみ:PL\_ISTRAIN」と「節点ひずみ:NSTRAIN」を追加した。



計算は、下図の様に、通常通り「FrontISTR 実行ボタン」をクリックして計算を開始させる。



計算終了後、設定項目 Tree の「post」を選択して、「データ変換」、「ParaView 起動」ボタンをクリックして、結果を確認する。



変位が、設定通り「0.01」であり、計算はうまくできている。

#### 4-2-3. 引き続き引張 20%の計算

ここまでの計算結果を使って、引き続き+20% (+0.010) まで引っ張ってみる。

前項で、0.001/1stepで0.010 (10step) まで圧縮した。ここから等間隔で、寸法を+0.010 まで引っ張る為には、+20step (計30step) 必要になる。

この為、上面 (topN) の境界条件は、0.001/1stepで30step変位させる値「0.030」になり、この値を境界条件として設定する。(既に10step進んでいるので、残り20stepで+0.020進むためには、境界条件を+0.030に設定する事で+0.020変化する事になる。)

この為、topNの境界条件は、以下で設定し直す。



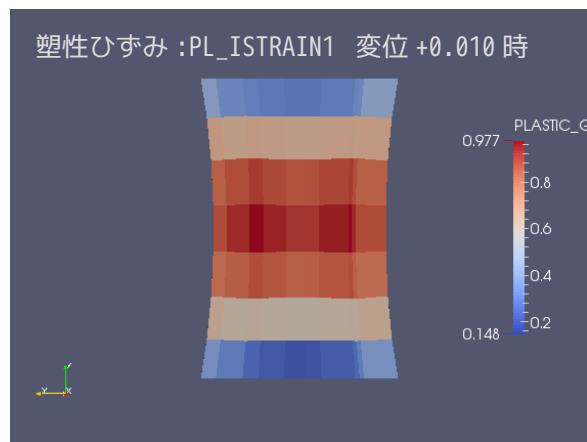
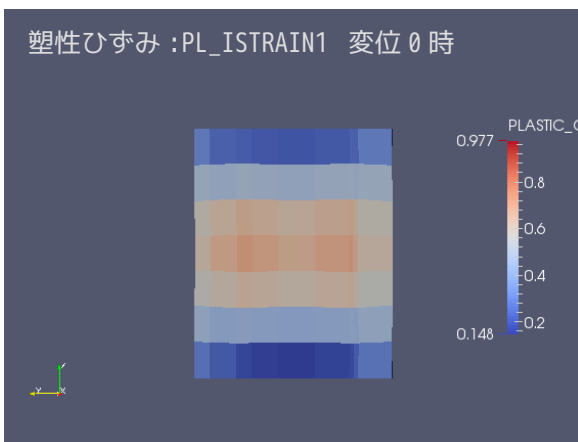
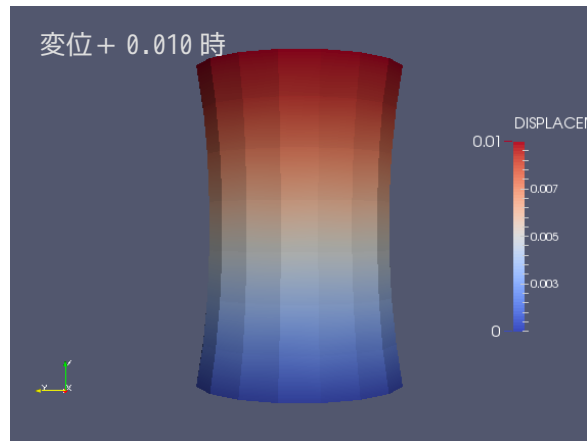
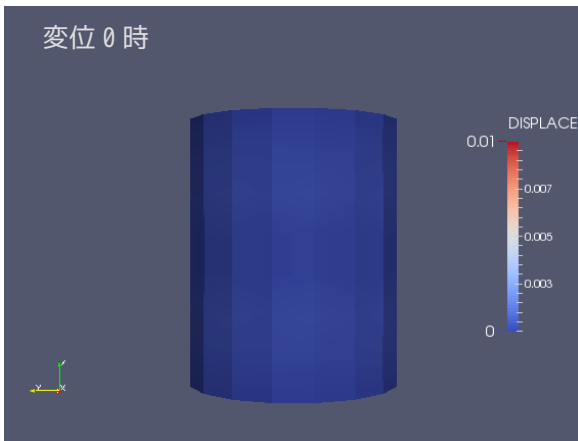
STEPの設定は、SUBSTEPの値を「30」に修正する。下図参照。



計算は、以下のように、restartの制御の「途中stepから開始する」チェックボックスにチェックを入れ、「FrontISTR実行」ボタンをクリックして、計算をスタートさせる。



計算結果は、30stepまで計算させたので、合計で30ファイル存在していることになる。このファイルをデータ変換し、結果を paraView で確認した結果が、以下になる。  
 計算は、変位が  $-0.010$  からスタートしているのので、途中「0」を通過して、「 $+0.010$ 」まで変化する。下図は、変位「0」の時と「 $+0.010$ 」の時の変位と塑性変形を表している。変位「0」でも塑性変形が残っているのが確認できる。





### 4-3. 固有値解析

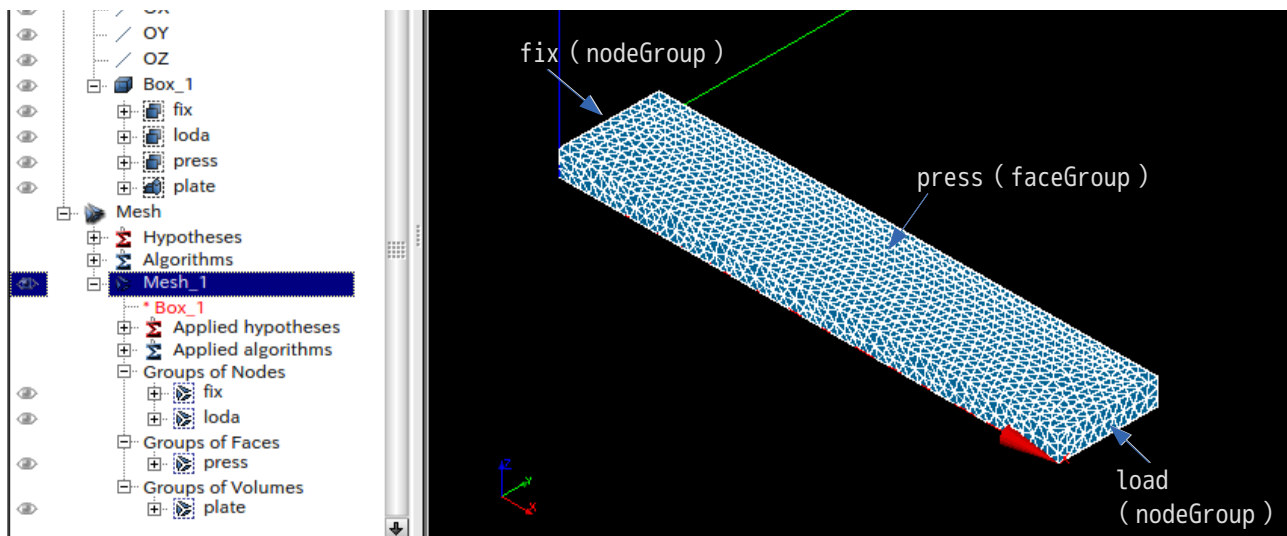
EasyISTR 上で固有値解析を行ってみる。

モデル形状は、線形弾性静解析で使った片持ち梁を使うことにする。

workFolder は、「CAE-fistr/Case/plateEigen」フォルダを新しく作成し、ここで解析する。

#### 4-3-1. モデル形状

モデルは、線形弾性静解析で使った片持ち梁を使う。その形状は、以下。メッシュファイル「plate.unv」を workFolder 内にコピーして、メッシュ変換する。



下図が、メッシュ変換した状態になる。

#### メッシュ内容

modelSize(xyz): 0.1 0.02 0.005

nodes 1731

elements type:341 5468

EGRP plate 5468

SGRP otherS 1668

SGRP press 1286

NGRP fix 40

NGRP load 36

EGRP (elementGroup)

1. plate

SGRP (faceGroup)

1. otherS
2. press

NGRP (nodeGroup)

1. fix
2. load

### 4-3-2. 固有値解析の開始

前項の形状の片持ち梁（材料：Steel）の固有値解析を行ってみる。

#### 4-3-2-1. 解析の種類の設定

設定項目 Tree 内の「解析の種類」を選択して、「固有値解析」を選択し、「設定」ボタンをクリックする。この操作により、設定項目 Tree 内の「解析の種類」の下に「設定」項目が追加される。



設定項目 Tree 内の「設定」を選択して、固有値解析の為の設定を行う。以下は、デフォルトの設定のまま。

固有値数	5	求める固有値の数
許容差	1e-8	誤差
最大反復数	100	収束しなかった場合は、ここを増やす。



#### 4-3-2-2. 材料物性値の設定

材料は、Steelとした。（下図参照。）



#### 4-3-2-3. 境界条件の設定

境界条件は、片持ち梁の端面 (fix) を固定する。

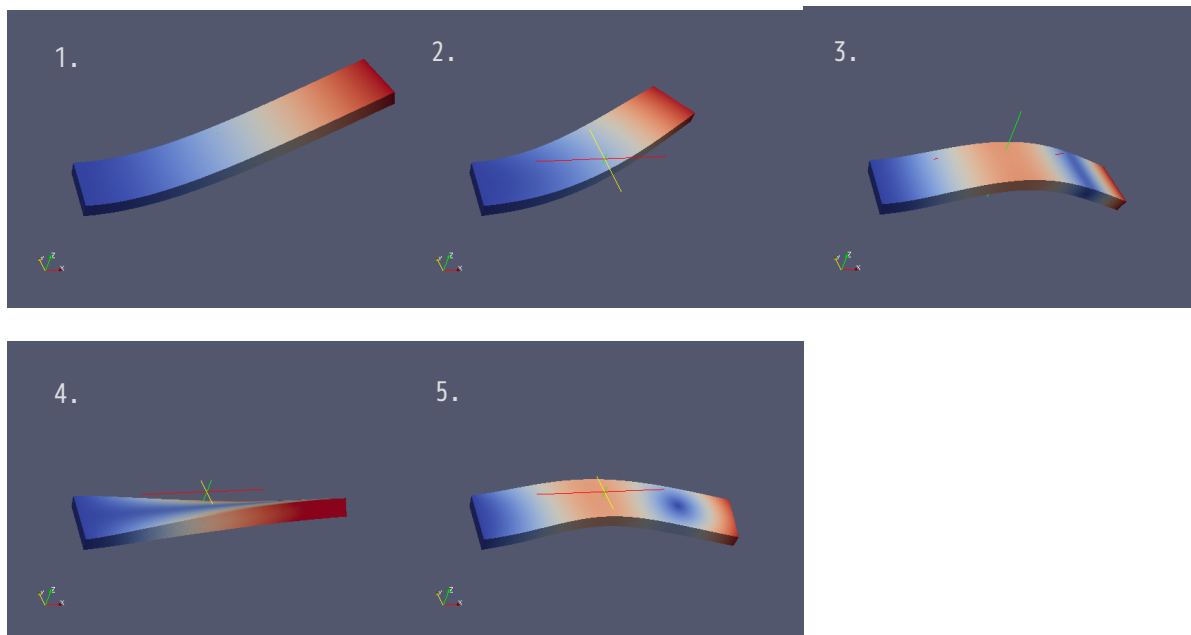


#### 4-3-2-4. 計算開始、結果の確認

設定項目 Tree 内の「solver」を選択し、「FrontISTR 実行」ボタンをクリックして、計算を開始させる。計算が終了すると、以下の画面が現れ、固有値が確認できる。

```
eigen_log x
1
2 *****
3 *RESULT OF EIGEN VALUE ANALYSIS*
4 *****
5
6 NUMBER OF ITERATIONS =      21
7
8 NO.   EIGENVALUE   ANGL.FREQUENCY  FREQUENCY(HZ)
9 ---   -
10  1    0.113434E+08  0.336800E+04    0.536034E+03
11  2    0.106112E+09  0.103011E+05    0.163947E+04
12  3    0.430635E+09  0.207517E+05    0.330274E+04
13  4    0.840066E+09  0.289839E+05    0.461293E+04
14  5    0.308137E+10  0.555101E+05    0.883471E+04
15 |
```

変形モードに関しては、「post」画面上で、データ変換 (vtk 変換) して、paraView で確認する。  
下図が確認した結果になる。2 と 5 は、板幅方向に変形している。



#### 4-4. 周波数応答解析

EasyISTR で周波数応答解析を行ってみる。

この周波数応答解析は、固有値解析の結果を使うので、前項で解析した「plateEigen」フォルダをコピーして、新しく「CAE-fistr/Case/plateEigenResponce」フォルダを作成し、これを workFolder として設定する。モデル形状は、前項と同じ片持ち梁の形状になる。

##### 4-4-1. モデル形状

前項で使用した片持ち梁の固有値解析結果をコピーして workFolder としているので、モデル形状は、前項と同じ片持ち梁になる。

##### 4-4-2. 周波数応答解析の開始

固有値解析の結果をコピーしているなので、メッシュや材料物性値は既に設定されている。

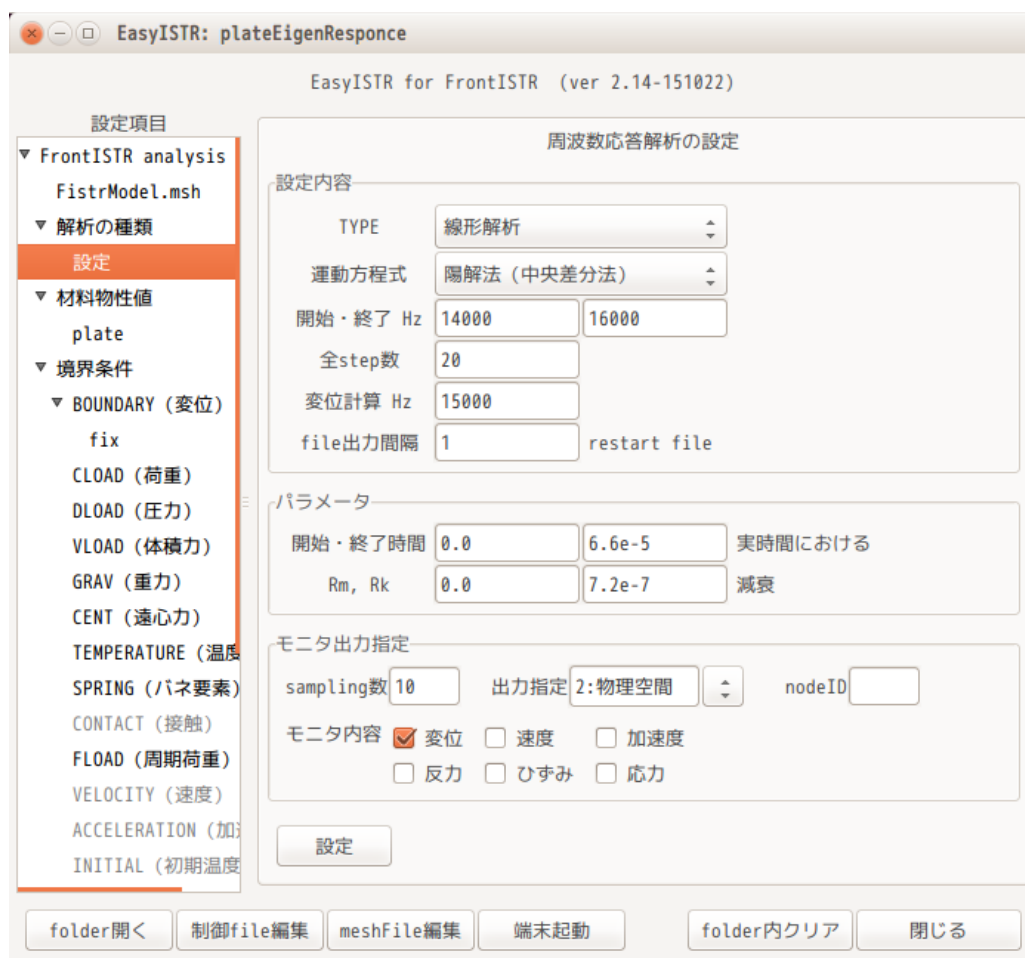
###### 4-4-2-1. 解析の種類の設定

設定項目 Tree 内の「解析の種類」を選択して、解析の種類を「周波数応答解析」に設定する。下図参照。



###### 4-4-2-2. 周波数応答解析の内容を設定

この後、設定項目 Tree 内の「解析の種類」>「設定」を選択して、周波数応答解析の為の設定を行う。下図がデフォルトの状態になる。



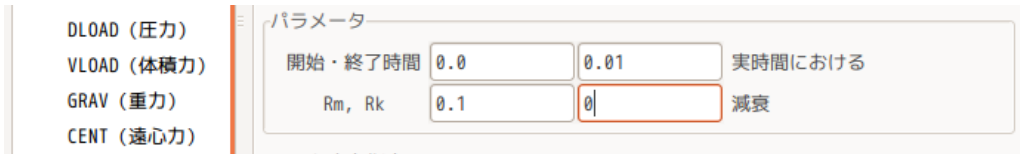
まず、TYPEと運動方程式は、そのまま「線形解析」、「陽解法（中央差分）」を選択。周波数は、前項の固有値解析の1次周波数が「536 Hz」の為、100～1000 Hzまで求めてみる。周波数の間隔は、10Hz間隔とする。（100～1000までを90分割する事になる為、「全step数:90」を設定する。）下図の様に設定。

変位計算 Hz については、周波数応答のみ求めるのであれば、いくつでも構わないので、「1000」を入力した。

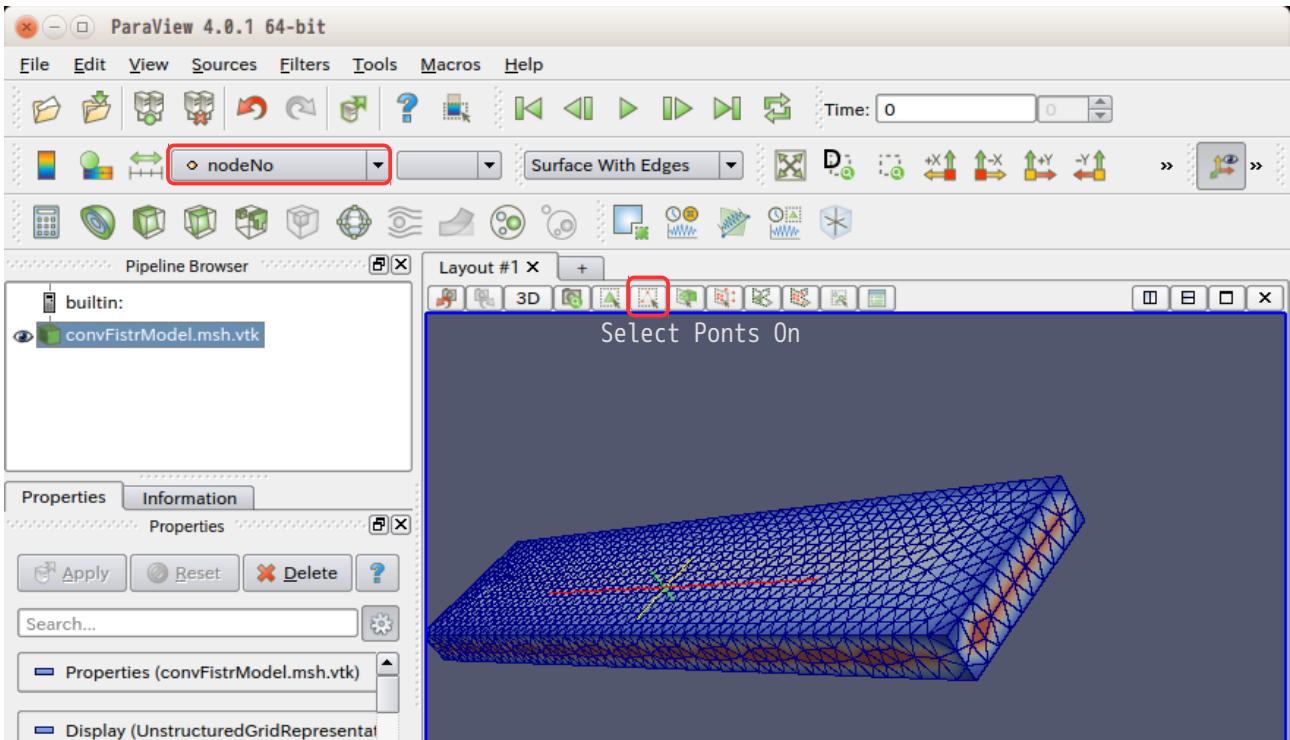


減衰は、質量マトリックスにかかる減衰（速度にかかる減衰）0.1を設定する。下図のように設定。

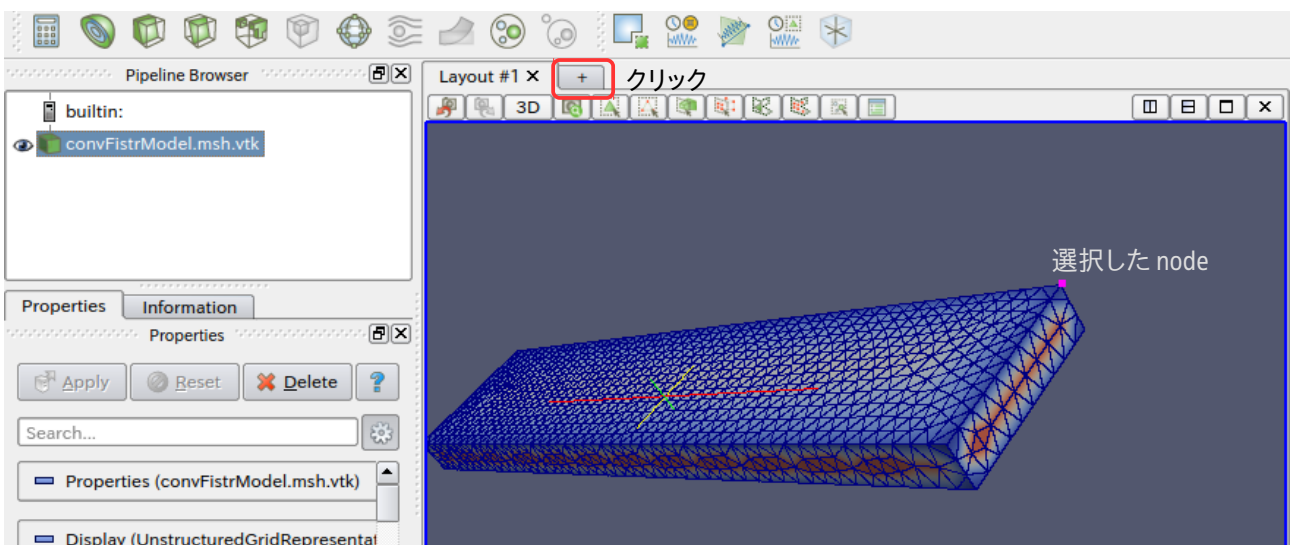
開始、終了時間は、周波数応答には関係ない。この時間は、上記の変位計算 Hz で指定した周波数で周期荷重を加えて過渡応答を求める時の開始、終了時間になる。



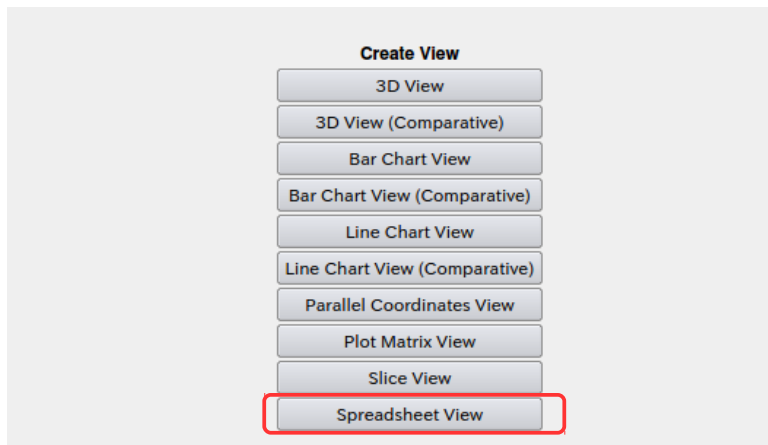
モニタする部位は、片持ち梁先端とする。この先端の nodeNo を調べる必要があるので、この nodeNo を paraView を使って調べる。下図は、nodeNo を表示させた状態。ここで「Select Points On」ボタンをクリックして、マウスをドラッグさせ、node を選択する。



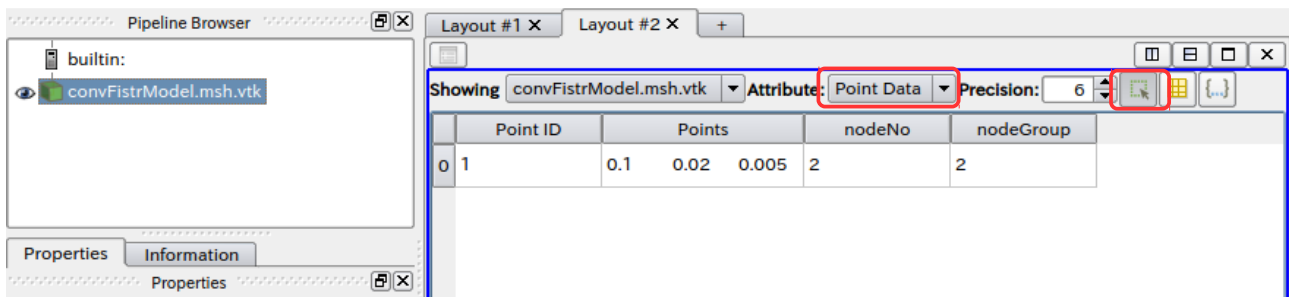
下図が node を選択した状態。ピンク色の点を選択した node になる。この状態で「+」タグをクリックして、新しい画面を表示させる。



この後、「Spreadsheet view」をクリックして、表を表示させる。



この後、Point Data が表示されていることを確認の上、「show only selected elements」ボタンをクリックして、選択した node のみを表示させる。下図がその状態。これから、選択した nodeNo が「2」であることが確認できる。



この nodeNo をモニタする nodeID として設定する。下図が設定した状態。

sampling 数は、周波数応答には関係ないので、いくつでも構わない。この値は、変位計算 Hz で周期荷重をかけた時の過渡応答を求める時の sampling 数になる。開始・終了時間で設定した時間をこの sampling 数で計算する。



#### 4-4-2-3. 境界条件の設定

境界条件は、周期荷重 (FLOAD) を設定する必要がある。この境界条件は、どの場所に、どの程度の周期荷重を設定するかを指定するもの。(この荷重で周波数応答を計算する。)

今回は、梁の先端 (load) 面の Z 方向に 1.0N の周期荷重を加える。下図の様に設定した。



設定項目

- VLOAD (体積力)
- GRAV (重力)
- CENT (遠心力)
- TEMPERATURE (温度)
- SPRING (バネ要素)
- CONTACT (接触)
- ▼ FLOAD (周期荷重)
  - Load
  - VELOCITY (速度)
  - ACCELERATION (加

FLOAD (周期荷重) の設定

nodeGroup名: Load

LOAD\_CASE  1:実部, 2:虚部

x

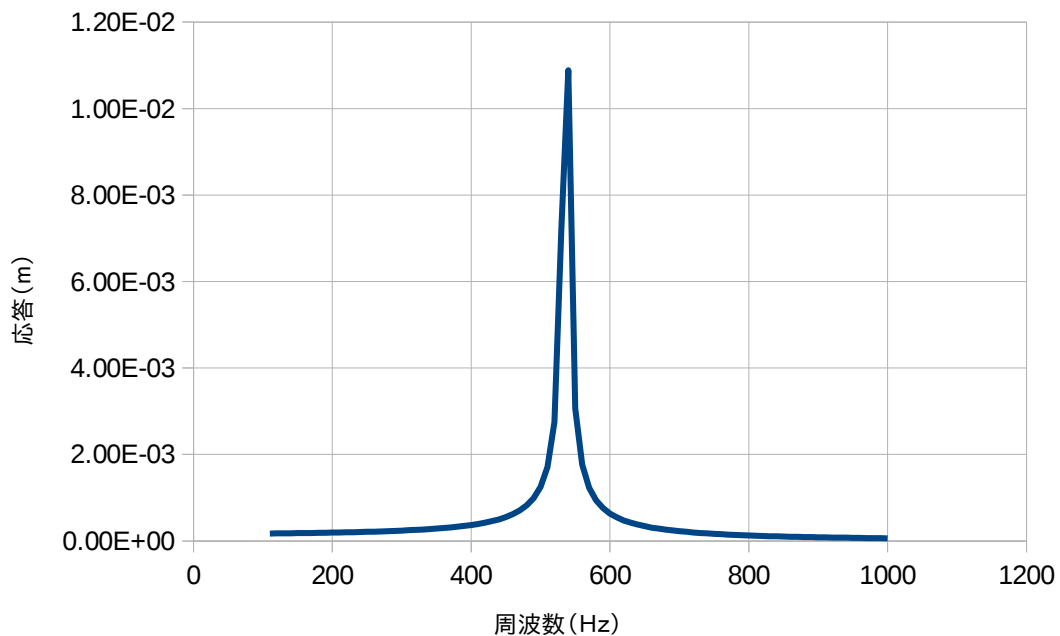
y

z

#### 4-4-2-4. 計算開始、結果の確認

設定項目 Tree 内の「solver」から「FrontISTR 実行」ボタンをクリックして計算を開始する。計算が終了すると、EasyISTR は、「0.log」ファイルから「resFrequencyResponse.csv」ファイルを作成する。このファイルは、周波数応答の結果が csv 形式で保存されているので、office など、結果を直ぐに確認できる。

下図が、「resFrequencyResponse.csv」を開いて、グラフを作成した結果になる。



## 4-5. 時刻歴応答解析

EasyISTR で時刻歴応答解析を行ってみる。

モデル形状は、固有値解析や周波数応答で使った片持ち梁で解析する。workFolder は、前項で作成した「plateEigenResponse」をコピーして、「CAE-fistr/Case/plateTimeResponse」フォルダを作成して、この中で時刻歴応答解析を行う。

### 4-5-1. モデル形状

モデルは、固有値解析や周波数応答解析で使ったモデルと同じモデルを使う。物性値もそのまま使用する。

### 4-5-2. 境界条件一定値（線形・陽解法）

まず、固有値解析や周波数応答解析の結果が邪魔になるので、これらを削除する。削除の方法は、EasyISTR 画面下の「folder 内のクリア」ボタンをクリックして、

現在の結果ファイル

固有値解析の結果ファイル

周波数応答解析の結果ファイル

にチェックをつけて、「削除」ボタンをクリックして、削除する。下図参照。



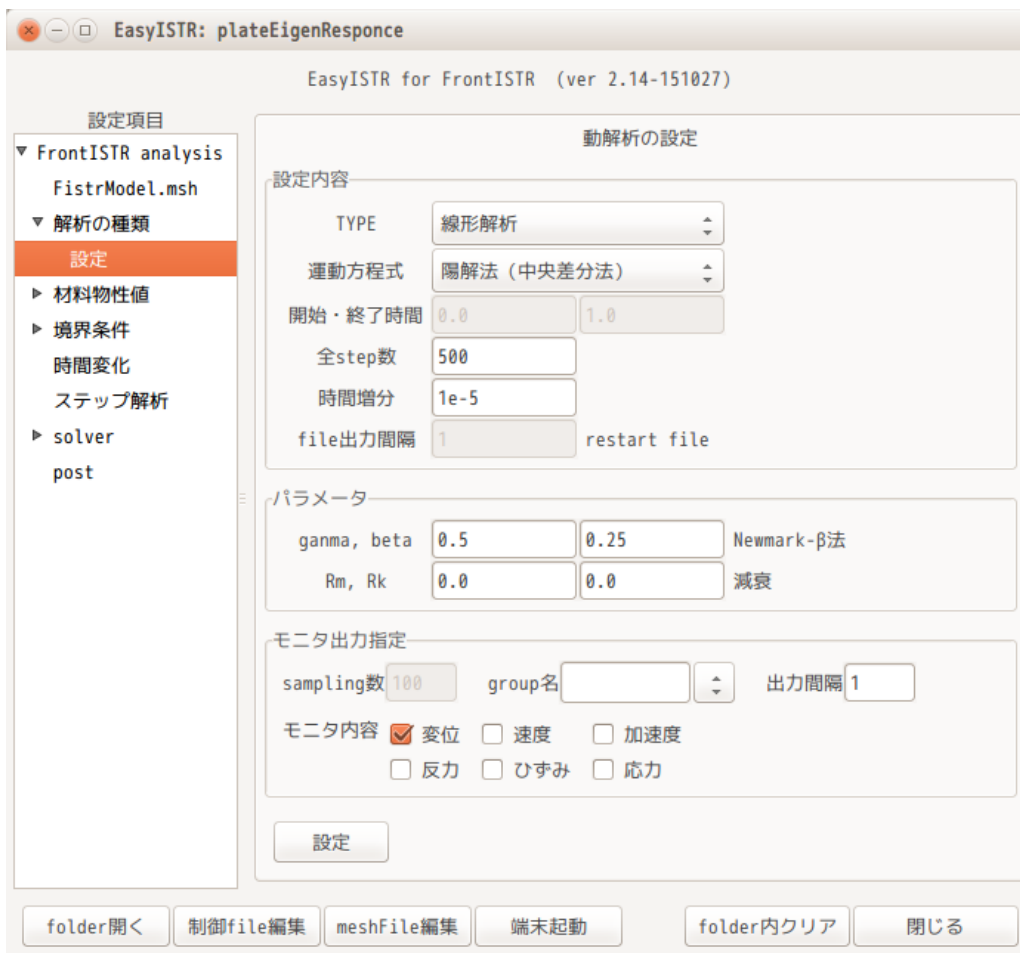
#### 4-5-2-1. 解析の種類の設定

設定項目 Tree 内の「解析の種類」を選択して「動解析」に設定する。（下図参照）



#### 4-5-2-2. 時刻歴応答解析の内容を設定

設定項目 Tree 内の「解析の種類」>「設定」を選択して、内容を設定する。  
下図が、default の設定内容になる。



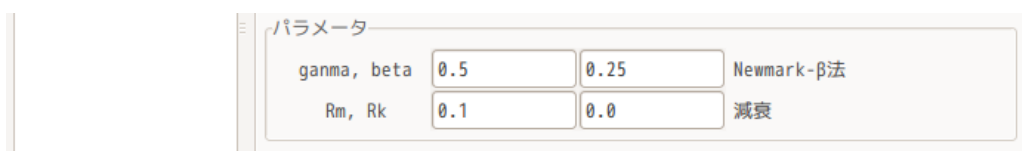
これに対して、現在設定できる項目について設定する。

TYPE と運動方程式は、「線形解析」、「陽解法 (中央差分法)」を選択する。  
計算する時間と、時間増分 ( $\Delta t$ ) は、全 step 数と時間増分で設定する。ここは、時間増分がいくつなら計

算できるか確認できていないので、ここは default のままとしておく。下図参照。



パラメータ内の gamma、beta は、運動方程式を陽解法に設定しているので、関係なくなり、このままの設定。減衰は、質量マトリックスにかかる減衰 Rm を「0.1」に設定した。



モニタ出力は、指定した場所の変位などの値をテキスト形式で出力させる指定になる。周波数応答解析で指定した nodeNo「2」を設定しておく。出力間隔は、後で設定する。(group名は、nodeGroup名または nodeNoを入力する。)



### 4-5-2-3. 境界条件の設定

境界条件は、片持ち梁の端面 (fix) を固定し、先端 (load) に Z 方向節点荷重「1.0」をかけることにする。荷重がかかっていない状態から、片持ち梁先端に荷重をかけて変形させる事を想定。以下の様に設定した。





#### 4-5-2-4. 試しに計算

時間増分がいくつなら計算ができるか確認する。

固有値解析の結果、5次の周波数が8800Hzだったので、この周波数の $1/10 = 1e-5$ で計算してみる。時間増分 $1e-5$ は、デフォルトの値なのでそのまま計算してみる。「FrontISTR実行」ボタンをクリックして計算させる。

この結果、「Displacement increment too large」のエラーが発生。この為、時間増分を減らして再確認。時間増分が「 $1e-7$ 」では、安定して計算できたので、この値を設定する。

この時間増分は、境界条件にも関係するので、固有値解析の結果だけでは、決まらない。

#### 4-5-2-5. 時刻歴応答の内容を再設定

時間増分が決まったので、時刻歴応答の再設定を行う。(全step数と時間増分を設定する。)

時間増分 $1e-7$ で5msまで計算してみる。1次の固有振動数が約540HZの為、5msで1次周波数の2波形分までは計算できることになる。

この設定にすると、全step数は「50000」ステップになる。下図参照。



モニタの出力間隔は、「10」ステップ間隔にする。モニタ内容は、全てにチェックを入れた。下図の様に設定した。



#### 4-5-2-6. 計算開始、結果の確認

計算を開始させる前に、結果 file を全て削除しておく。（試しに計算させているので、結果ファイルが残っている。） EasyISTR 画面下の「folder 内クリア」ボタンをクリックして、削除する。

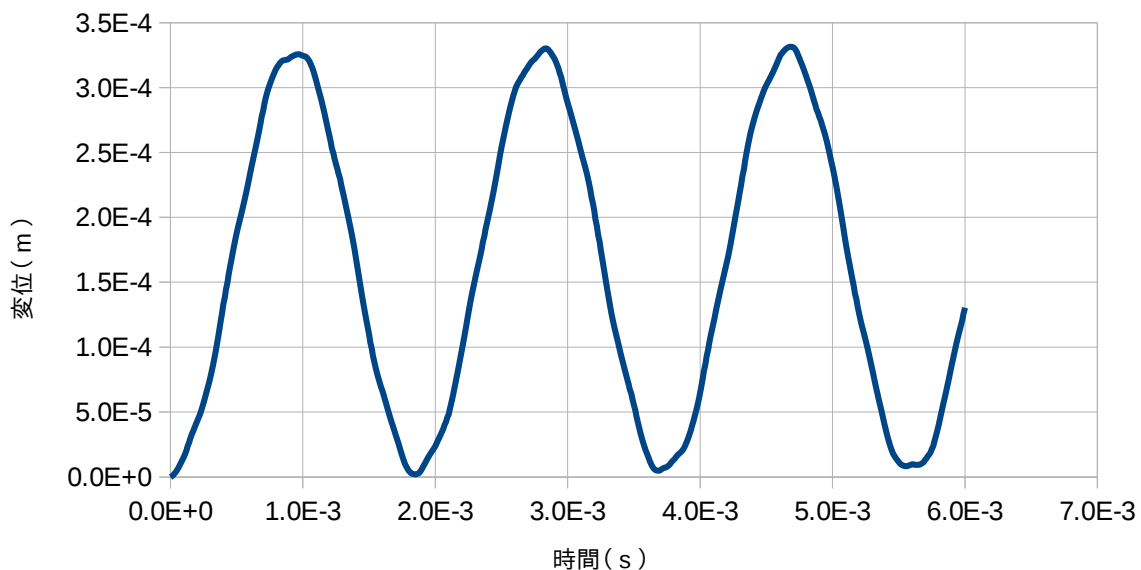
今回の時刻歴応答解析は、50000 ステップまで計算させるので、時間がかかる。この為、並列計算を行う。また、計算ステップが多いため、結果ファイルの出力頻度を 1000 回に設定しておく。（結果ファイルが 50 ケでかあがる設定。）



計算を途中で停止させた場合は、「途中 step から開始する」にチェックを入れて計算を開始させる事で途中からでも計算開始できる。

計算結果は、モニタしている nodeNo「2」の変位の計算結果が「dyna\_disp\_2.txt」ファイルとして保存されているので、この結果を確認する。下図がその結果になる。

下図は、60000 ステップまで計算されているが、これは、50000 ステップに引き続き、途中ステップから再計算させた結果になる。

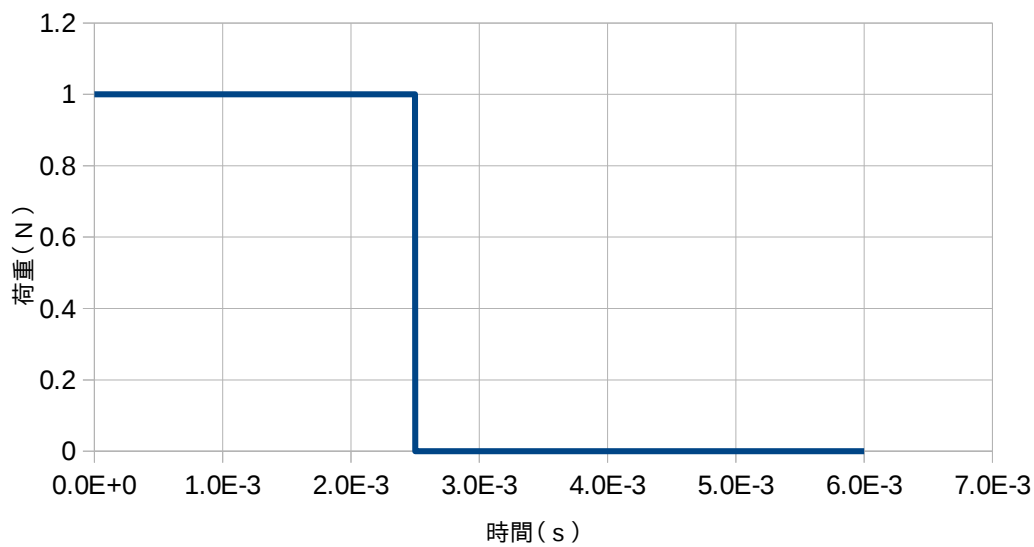


### 4-5-3. 境界条件に時間変化を与える（線形・陽解法）

前項の計算は、片持ち梁の先端（load）に一定荷重をかける解析だが、ここでは、この荷重に時間変化を与える解析を行う。

#### 4-5-3-1. 時間変化の計算開始

片持ち梁の先端（fix）のZ方向の値に時間変化を与える。以下の変化を与える解析を行う。



#### 4-5-3-2. 境界条件の設定

現在、境界条件を設定しているのは、片持ち梁の端面（fix）を固定し、先端（load）のZ方向に1Nの節点荷重をかけている。これは、変更せずこのまま。（下図参照。）

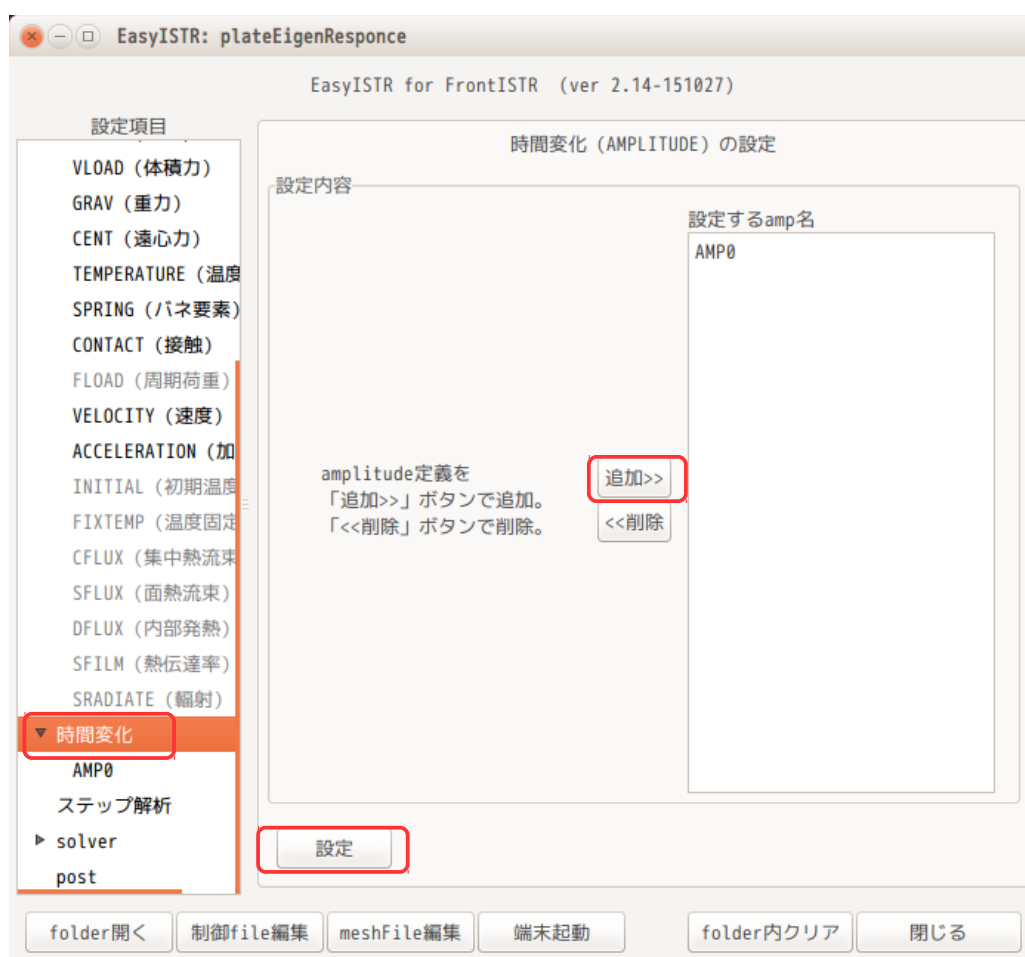




#### 4-5-3-3. 境界条件に時間変化を与える

片持ち梁の先端 (load) に設定した境界条件 (Z方向に1.0Nの節点荷重) に、時間変化を与える。

与える方法は、設定項目 Tree 内の「時間変化」を選択し、「追加」ボタンをクリックすると、設定する amp 名側に「AMP0」が追加される。この後、「設定」ボタンをクリックして、確定させる。これにより、「AMP0」が設定項目 Tree 内に追加される。(下図参照。)



この後、設定項目 Tree 内の「時間変化」>「AMP0」を選択して、時間変化の詳細を設定する。この画面上で、時間変化させる境界条件名と時間変化のデータを指定する。

まず、時間変化させる境界条件は、梁の先端 (load) に与えた条件になるので、「CLOAD,load」を選択して



「選択>>」ボタンをクリックして、「時間変化させる境界条件」側に移動しておく。下図が移動させた状態。この後、VALUEを「RELATIVE (相対値)」を選び、「data作成・編集」ボタンをクリックして、dataを入力する。



「data作成・編集」ボタンをクリックすると、office が起動するので、データを入力する。amp データは、RELATIVE (相対値) を選択している為、境界条件に対する倍率になる。下図の様に入力した。

	A	B	C	D	E	F	G	H	I
1		ampData							
2		time	value						
3		0	1						
4		2.50E-03	1						
5		2.51E-03	0						
6		6.00E-03	0						
7									
8									

データ保存は、csv データとして保存する。作成したデータは、「AMP0\_ampData.csv」と自動的にファイル

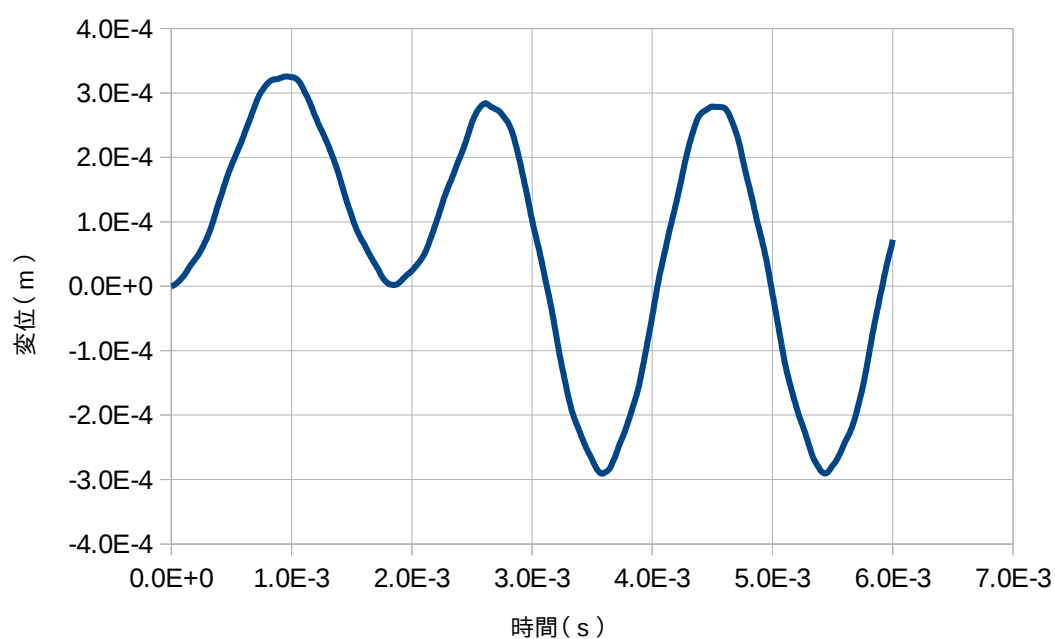
名が振られ、保存されるので、後からでも内容を編集できる。

全ての設定 (data 作成、時間変化を与える境界条件) の設定が終了した段階で「設定」ボタンをクリックして内容を確認させる。この操作により、FistrModel.cnt、FistrModel.msh ファイルへ書き込まれる。(amp データは、メッシュファイルの最後には書き込まれる。)

#### 4-5-3-4. 計算開始、結果の確認

計算は、4 並列で、計算を開始した。実行時間は、前項の時間変化を与えない設定とほぼ同じ計算時間。(時間変化の設定を追加しても、計算時間は増えない。)

以下が、変化のグラフになるが、荷重が変化している様子が確認できる。



この方法を用いると、振動データ等を取得した後、そのデータをそのまま境界条件として設定もできる事になり、応用範囲が広がる。

## 4-6. 熱応力解析

EasyISTR で熱応力解析を行ってみる。

モデル形状は、Aluminum と Steel を貼りあわせたバイメタルとして解析する。

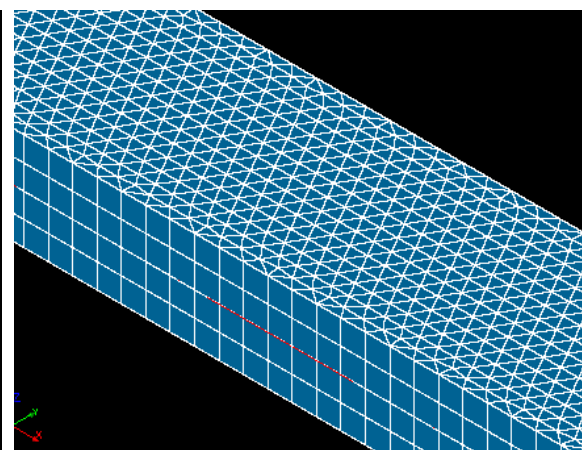
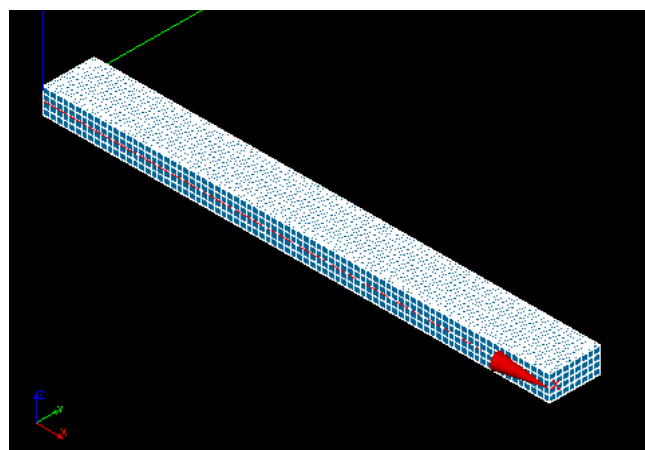
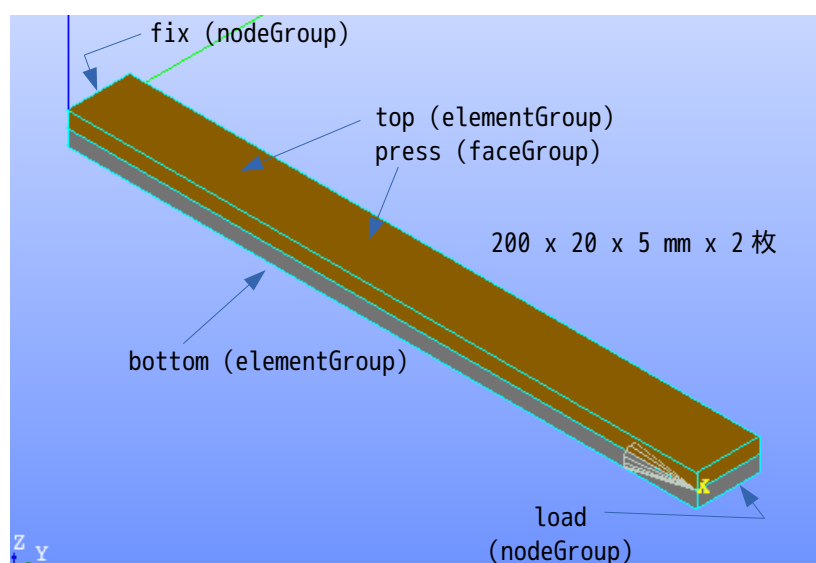
workFolder は、「CAE-fistr/Case/biMetal」フォルダを作成して、この中で解析を行う。

### 4-6-1. モデル形状

200 x 20 x 5 mm の 2 枚の板を貼りあわせた構造。

メッシュは、prism で作成しており、top と bottom は、節点を共有している。メッシュサイズは 2.5mm で作成している。これらは、salome で作成した。

このメッシュを「biMetal.unv」として、workFolder 内に保存しておく。



解析は、片側 (fix) を固定して、温度上昇させる解析を行ってみる。

## 4-6-2. 熱応力解析の開始

熱応力解析は、初期温度を 20℃として、100℃まで温度上昇させて、変位を確認してみる。

### 4-6-2-1. メッシュ変換

設定項目 Tree 内の「FistrModel.msh」を選択して、「参照…」ボタンをクリックして「biMetal.unv」を選択する。「ファイル変換」ボタンをクリックして、unv 形式→fistr 形式に変換する。この後、倍率を「0.001」に設定し、「倍率変更」ボタンをクリックして 1/1000 に縮小する。(salome 側で mm 単位でモデルを作成しているため。)

```
メッシュ内容
modelSize(xyz): 0.2 0.02 0.01

nodes 8065
elements type:351 6096
elements type:351 6096

EGRP top 6096
EGRP bottom 6096
SGRP otherS 6800
SGRP press 3048
NGRP fix 45
NGRP load 45
```

EGRP (elementGroup)

1. top
2. bottom

SGRP (faceGroup)

1. otherS
2. press

NGRP (nodeGroup)

1. fix
2. load

### 4-6-2-2. 解析の種類を設定

EasyISTR の設定項目 Tree 内の「解析の種類」を選択して、「線形弾性静解析」を選択し、「設定」ボタンをクリックして、これを確定させる。(下図参照)



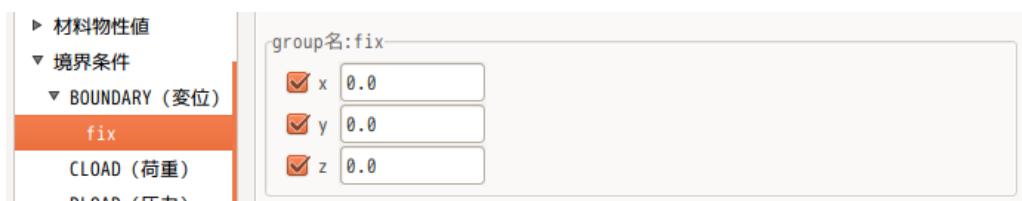
### 4-6-2-3. 材料物性値の設定

下部 (bottom) を Steel、上部 (top) を Aluminum に設定する。下図参照。



#### 4-6-2-4. 境界条件の設定

境界条件は、パイメタルの片側 (fix) を固定し、温度を 20°C から 100°C に上昇させる。下図は、fix の境界条件 (固定) の内容。



温度の設定は、参照温度を「20.0」に設定し、Temp\_Value を「100.0」に設定する。参照温度が、初期温度 (応力「0」の状態) になる。

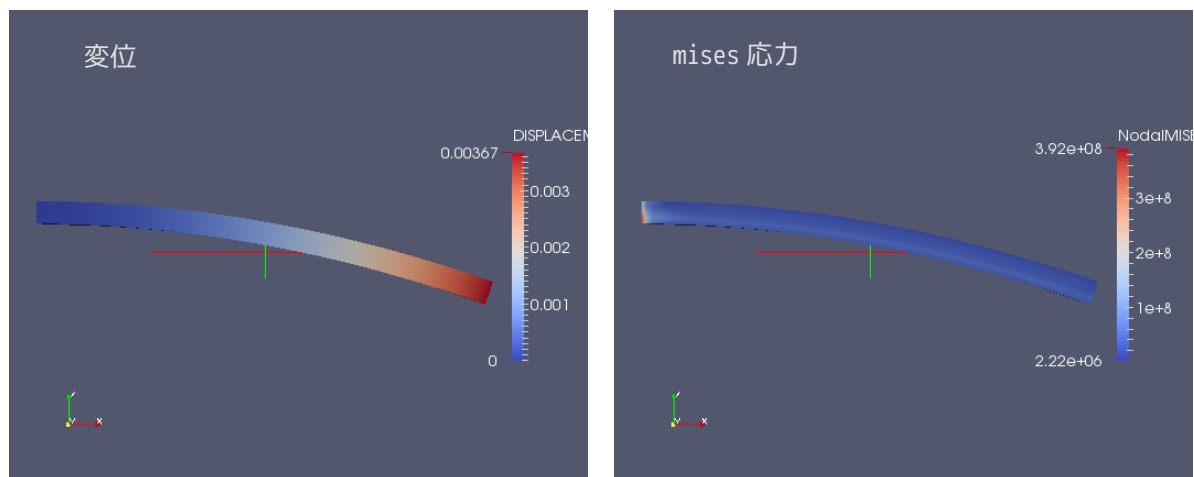
チェックボックス「熱伝導解析結果を使う」は、熱伝導の解析を行った結果 (温度分布が生じている状態) を使うかどうかであり、今回は使わないので、チェックしない。今回は、全てが均一な温度として解析する。



#### 4-6-2-5. 計算開始、結果の確認

境界条件の設定ができたので、計算を開始する。

設定項目 Tree 内の「solver」を選択し、「FrontISTR 実行」ボタンをクリックして、計算を開始させる。計算が終了したら、設定項目 Tree 内の「post」を選択し、「データ変換」、「paraView 起動」ボタンをクリックして、paraView で結果を確認する。下図が確認した状態。



温度変化により、バイメタルが変形している。

## 4-7. 熱伝導解析（静解析）

EasyISTR で熱伝導の静解析を行ってみる。

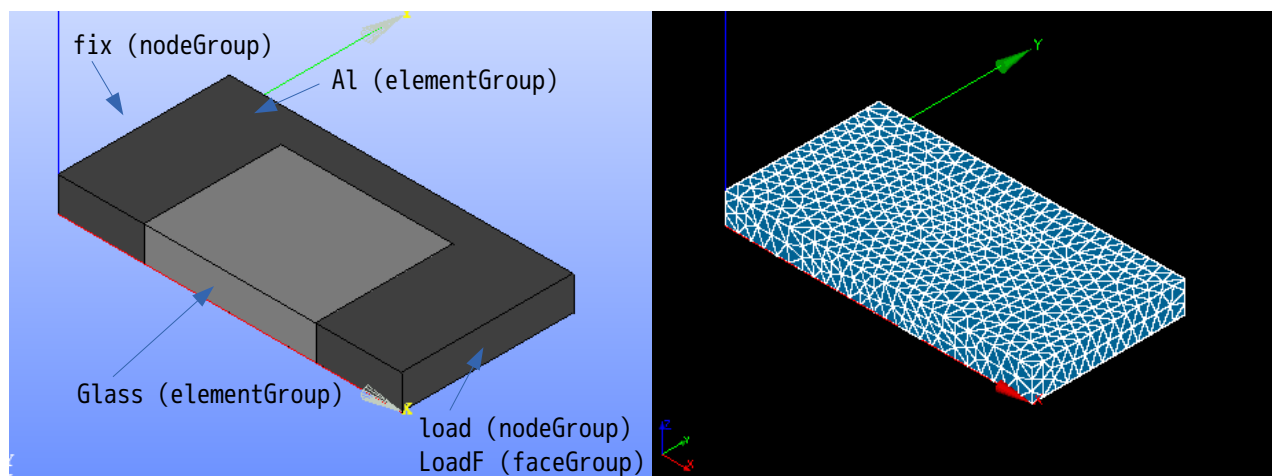
モデル形状は、Aluminum と熱伝導の悪い Glass を組み合わせたモデルとして解析する。

workFolder は、「CAE-fistr/Case/biMetalStaticHeat」フォルダを作成して、この中で解析を行う。

### 4-7-1. モデル形状

モデルは、全体が、100 x 50 x 10 mm でこの中に、50 x 40 x 10 mm の Glass が埋め込まれた形状。

メッシュは、Salome で Netgen-1D-2D-3D、4mm でメッシュを切っている。



でき上がったメッシュを「AlGlassPlate.unv」として workFolder 内に保存しておく。

### 4-7-2. 熱伝導解析（静解析）の開始

plate 端面（fix）を温度固定し、先端（loadF）に熱流束を加えて、各部温度を求めてみる。

#### 4-7-2-1. メッシュ変換

EasyISTR 側で「AlGlassPlate.unv」ファイルを指定して、メッシュ変換した結果が以下になる。

```
メッシュ内容
modelSize(xyz): 0.1 0.05 0.01

nodes 936
elements type:341 1628
elements type:341 1106

EGRP Al 1628
EGRP Glass 1106
SGRP otherS 1872
SGRP loadF 56
NGRP fix 53
NGRP load 45
```

EGRP (elementGroup)

1. Al
2. Glass

SGRP (faceGroup)

1. otherS
2. loadF

NGRP (nodeGroup)

1. fix
2. load

#### 4-7-2-2. 解析の種類の設定

設定項目 Tree 内の「解析の種類」を選択して、「熱伝導（静解析）」を選択する。下図参照。



#### 4-7-2-3. 材料物性値の設定

材料は、Al を「Aluminum」、Glass を「Glass」として設定する。下図参照。



#### 4-7-2-4. 境界条件

境界条件は、plate 端面 (fix) を 20 °C で温度固定し、先端 (loadF) に熱流束 1e5 W/m<sup>2</sup> を加える。

温度固定の方法は、設定項目 Tree 内の「FIXTEMP (温度固定)」を選択して、設定する nodeGroup 「fix」を確定して、この fix に設定温度「20」を設定する。下図が設定した状態。





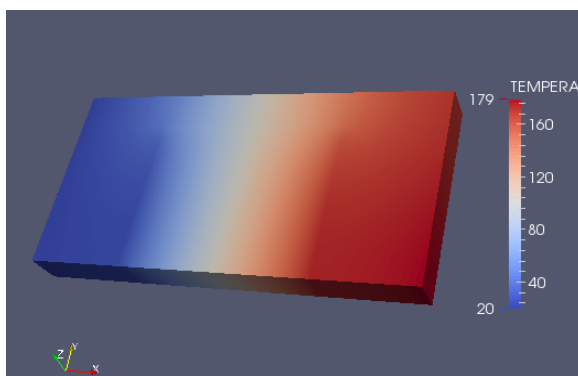
熱流束の設定方法は、nodeGroupに設定する方法と faceGroupに設定する方法があるが、ここでは faceGroupに熱流束を設定した。

設定項目 Tree 内の「SFLUX (面熱流束)」を選択して、設定する faceGroup「loadF」を確定して、この loadF に熱流束  $1e5$  W/m<sup>2</sup> を入力する。下図が入力した状態。



#### 4-7-2-5. 計算開始、結果の確認

設定項目 Tree 内の「solver」を選択して「FrontISTR 実行」ボタンをクリックして実行する。計算終了後、設定項目 Tree 内の「post」を選択して、「データ変換」ボタンをクリックして vtk 変換し、「ParaView 起動」ボタンで paraView を起動し確認する。下図が確認した結果になる。



### 4-7-3. 熱伝導解析結果から熱応力を求める

前項の熱伝導解析で求めた温度分布を使って、熱応力を求めてみる。  
熱伝導解析結果を保存する為に、熱応力解析は、「biMetalStaticHeat」をコピーして、「staticHeatStress」フォルダを作成し、これを workFolder として、熱応力解析を行う。

#### 4-7-3-1. 解析の種類の設定

解析の種類を「線形弾性静解析」に設定する。



#### 4-7-3-2. 材料物性値の設定

材料名は、既に設定されているが、熱伝導解析で使う物性値と応力解析で使う物性値が異なっているので、これを再設定する意味で、それぞれの材料名を選択して、「設定」ボタンをクリックして、設定し直す。「設定」ボタンをクリックする事で、応力解析用の物性値に入れ替わる。





#### 4-7-3-3. 境界条件の設定

境界条件は、plate 端面 (fix) を固定する。下図参照。



温度条件は、熱伝導解析結果を使用するので、以下のように設定する。

参照温度は、「20」を設定。

チェックボックス「熱伝導解析結果を使う」にチェックをつける。熱伝導解析結果は、静解析の結果のため、READRESULT、SSTEP、INTERVAL は、全て「1」になる。



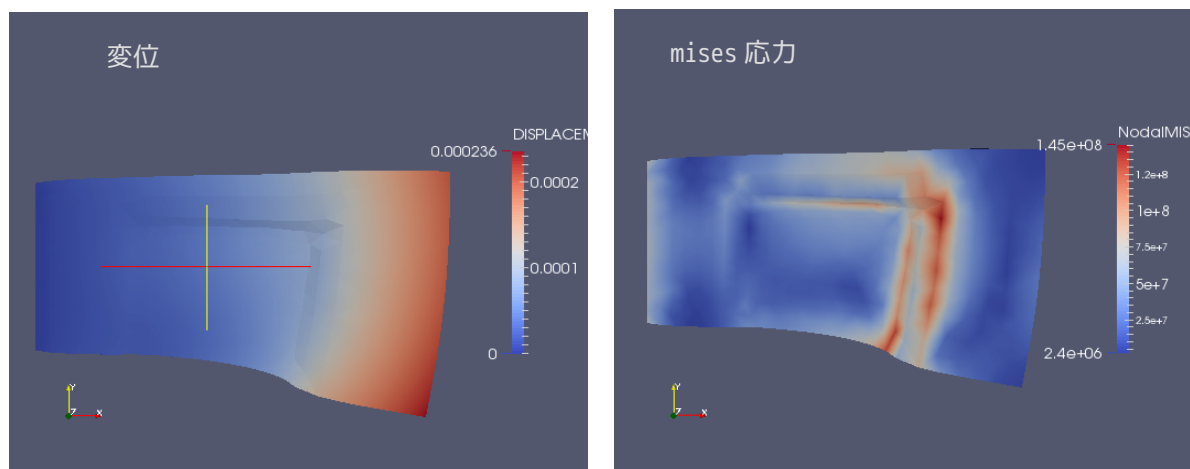
#### 4-7-3-4. 計算開始、結果の確認

準備が全て整ったので、計算を開始する。

計算が終了後、結果を vtk 変換し、結果を確認する。vtk 変換結果は、

熱伝導解析結果      convFistrModel\_temp.res.0.1.vtk  
 応力解析結果        convFistrModel.res.0.1.vtk

があるので、応力解析結果 (convFistrModel.res.0.1.vtk) を paraView で読み込み確認する。以下が確認した結果になる。(変位は、100 倍に拡大して表示している。)



## 4-8. 熱伝導解析（動解析）

EasyISTR 上で熱伝導解析（動解析）を行ってみる。

モデルは、前項と同じモデルを使用する。この為、前項で解析した workFolder 「biMetalStaticHeat」をコピーして、新たに「biMetalDynamicHeat」を作成し、これを workFolder として解析する。

### 4-8-1. モデル形状

前項の熱伝導（静解析）と同じモデルを使用する。

### 4-8-2. 熱伝導解析（動解析）の開始

解析は、plate 端面（fix）を 20℃に温度固定し、その反対側の端面（load）を 100℃で温度固定する。

前項では、熱流束を境界条件として定義したが、ここでは、温度固定で解析する。この理由は、熱伝導の悪い Glass を材料に設定している為、温度変化にかなりの時間がかかってしまう。この為、ここでは簡易的に温度固定で計算させる。

#### 4-8-2-1. 解析の種類の設定

設定項目 Tree 内の「解析の種類」を選択して、「熱伝導（動解析）」を選択し、「設定」ボタンをクリックして、これを確定させる。下図は、確定させた状態。



この後、設定項目 Tree 内の「解析の種類」>「設定」を選択して、熱伝導解析の内容を設定する。

まず、チェックボックス「時間増分を固定して計算する」にチェックを入れる。この理由は、境界条件として、片側を高温で温度固定する為、この面に接する要素は、温度上昇が激しい。（短時間で温度が上昇する。）この為、時間固定で計算させる。

温度固定ではなく、熱流束等を境界条件として設定するのであれば、温度上昇の程度が不明なので、設定が悪いと、細かく計算し過ぎてしまう事がある為、チェックボックスにチェックを外して、時間増分を自動的にまかせた方が、融通がきく。

今回の設定は、60s 間隔で 3600s まで計算する設定。



#### 4-8-2-2. 境界条件

境界条件は、まず、初期温度「INITIAL (初期温度)」を 20°C に設定する。静解析の場合は、この設定は不要だが、動解析の場合は、設定が必要になる。下図の様に設定した。



plate 端面の fix を 20°C に温度固定し、load を 100°C に温度固定する。以下の様に設定。



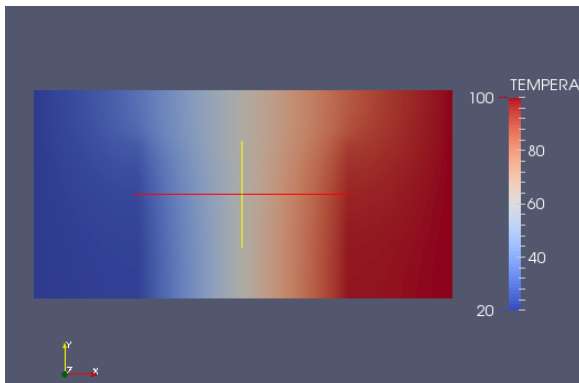


#### 4-8-2-3. 計算開始、結果の確認

計算を開始させる前に、EasyISTR 画面下の「folder 内クリア」ボタンをクリックして、前回の結果ファイル全てを削除しておく。

削除後、「FrontISTR 実行」ボタンをクリックして、計算を開始させる。

計算終了後、vtk 変換して、結果を paraView で確認する。下図が確認した結果になる。



#### 4-8-3. 熱伝導解析結果から熱応力を求める

前項で求めた温度分布を使って、熱応力を求めてみる。

workFolder は、そのまま変えずに「biMetalDynamicHeat」で解析する。

##### 4-8-3-1. 解析の種類の設定

設定項目 Tree 内の「解析の種類」を選択して、「線形弾性静解析」に設定する。  
下図参照。



#### 4-8-3-2. 材料物性値の設定

今の設定は、材料名が設定されているが、この設定は熱伝導解析用の材料物性値の為、これを線形弾性解析用の物性値に変更する必要がある。この変更方法は、各 elementGroup 名 (Al、Glass) の設定画面上で「設定」ボタンをクリックする事で材料物性値が入れ替わる。



#### 4-8-3-3. 境界条件の設定

境界条件は、plate 端面 (fix) を固定する。下図参照。





温度設定は、熱伝導解析結果を使う。設定項目 Tree 内の「TEMPERATURE (温度)」を選択して、「ALL」を確定し、その内容を設定する。

参照温度は、20°Cを設定。

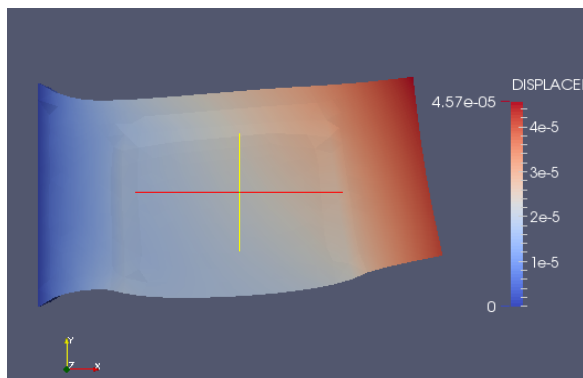
チェックボックス「熱伝導解析結果を使う」にチェックを入れる。チェックを入れると、デフォルトの設定が表示される。デフォルトの設定値は、熱伝導解析結果のステップ数を確認し、最後のステップの結果を使う設定になっている。

熱応力解析は、このデフォルトの設定を使って計算する。



#### 4-8-3-4. 計算開始、結果の確認

設定項目 Tree 内の「solver」を選択し、「FrontISTR 実行」ボタンをクリックして実行する。計算が終了した後は、設定項目 Tree 内の「post」を選択し、vtk 変換後、paraView で結果を確認する。下図が確認した結果になる。表示倍率を 400 倍に設定して、表示させている。



## 4-9. シェルの解析

シェル（面）のみで構成されたモデルの線形弾性静解析を行ってみる。solid 解析との違いは、以下になる。  
シェルの板厚を材料物性値と共に設定する。

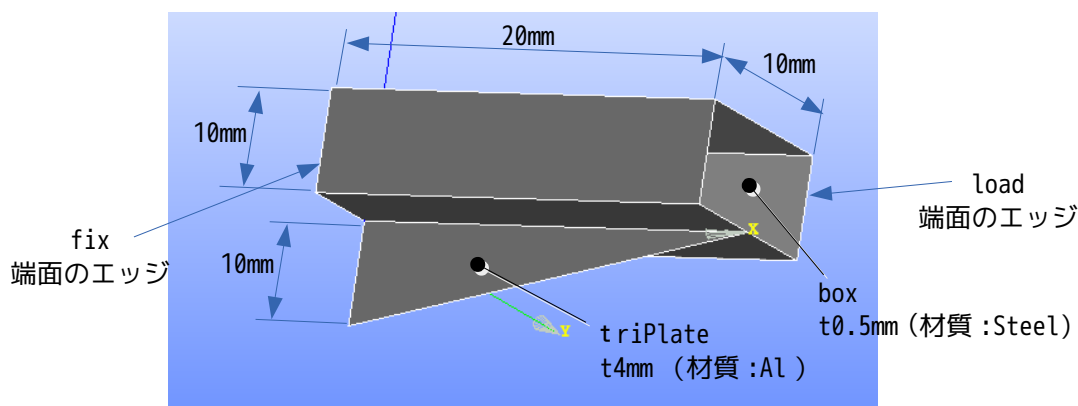
BOUNDARY の境界条件は、各軸の回転が追加されている。

応力、ひずみの計算結果は、シェル表裏の値が存在する。

シェルの場合、線形 solver は DIRECT or MUMPUS のみ対応。

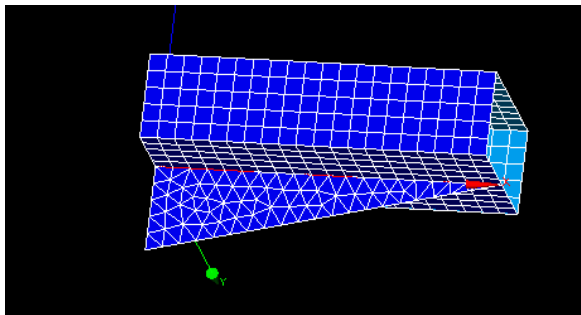
### 4-9-1. モデルの作成、メッシュ読み込み

モデルは、以下の様に面のみで構成されたモデルを考える。計算は、左側端面のエッジ全てを固定し、右側端面の全てのエッジに 1000N の等分布荷重を掛けた解析を行う。



このモデルを、salome で以下の様にメッシュを作成する。

box を四角形 1 次要素、triPlate を三角形 1 次要素でメッシュを作成している。



グループ化:

要素グループ: box, triPlate

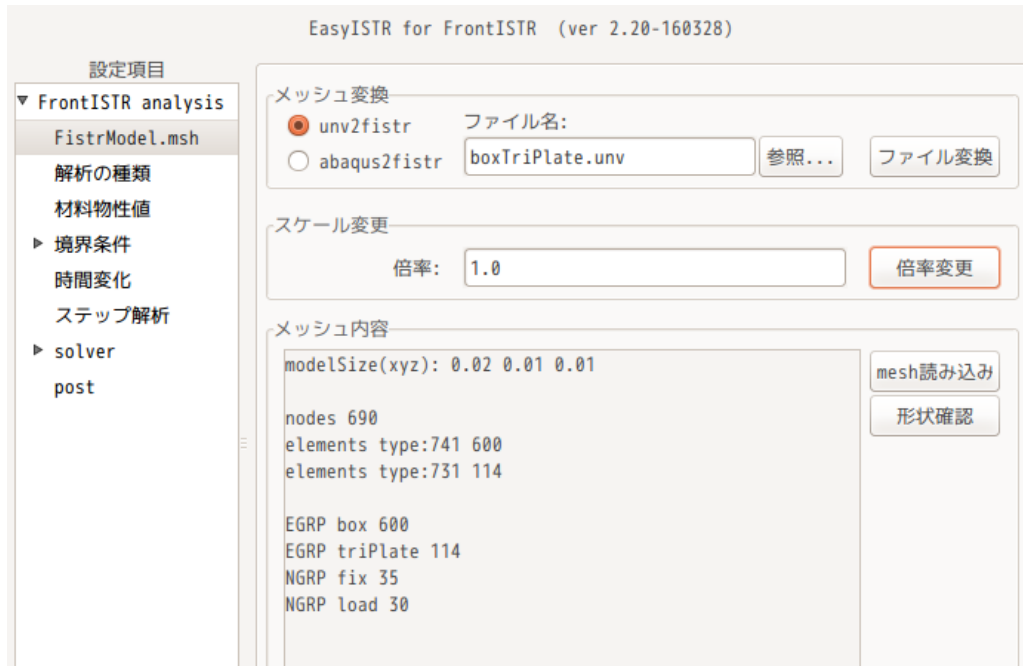
節点グループ: fix, load

EasyISTR の解析用の folder として、適当な場所に「shellBox」フォルダを作成し、ここに作成したメッシュファイル「boxTriPlate.unv」を保存しておく。

保存後、EasyISTR を起動し、作業用フォルダを「shellBox」に設定する。

この後、「boxTriPlate.unv」ファイルを fistr 用にメッシュ変換する。変換後、モデル倍率を 0.001 倍して、単位を m にしておく。

以下が、ファイル変換、倍率変更した後の状態。モデルサイズや要素グループ、節点グループが正常に読み込まれている。



#### 4-9-2. 材料の設定

ここで、box、triPlateの材料名と板厚（積分点数）を設定する。以下の様に設定する。



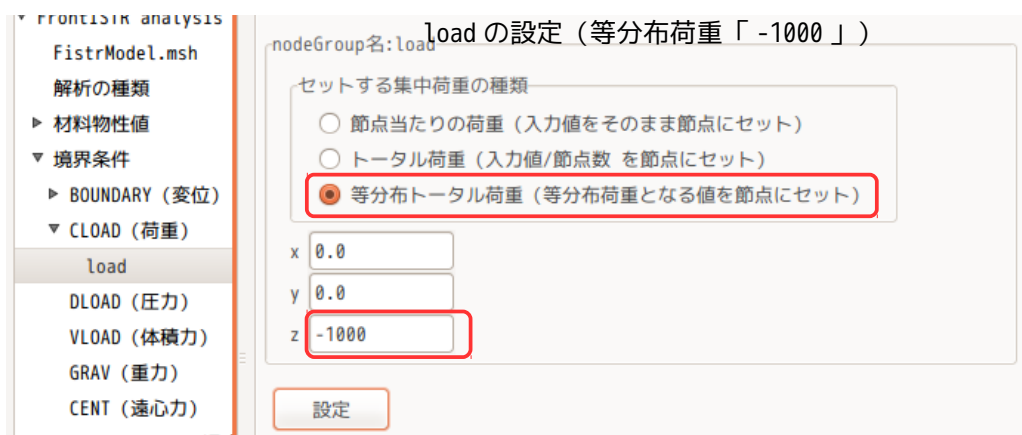
### 4-9-3. 境界条件の設定

境界条件は、左側端面エッジ fix を固定し、右側端面エッジ load に Z 方向の等分布荷重「-1000N」を設定する。

シェルを固定する場合は、変位と回転の両方を以下の様に固定する。  
(solid モデルの固定は、変位のみを設定する。)



load には、以下の様に等分布荷重で Z 方向に「-1000N」を設定する。

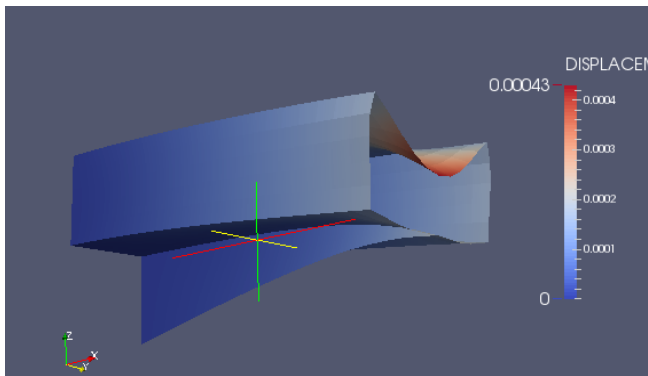


### 4-9-4. 計算開始、結果の確認

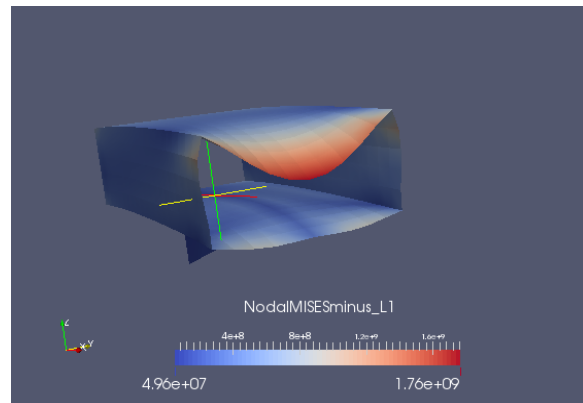
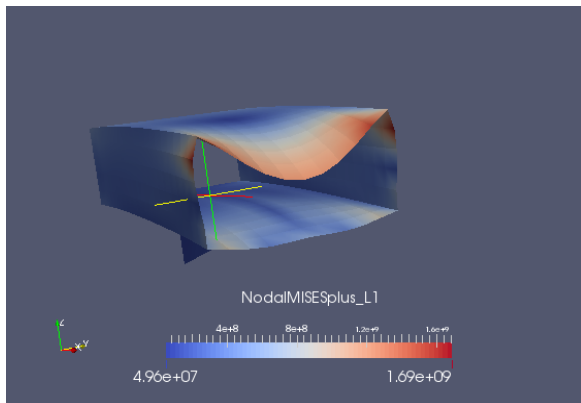
シェルの場合は、線形 solver が「DIRECT」or「MUMPS」しか計算できないので、どちらかを設定するが、ここでは、以下の様に「MUMPS」を設定している。



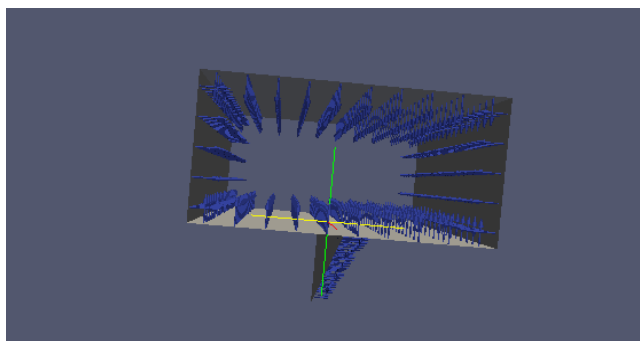
この後、通常通りに、「FrontISTR 実行」ボタンをクリックして計算させる。  
計算終了後、結果を vtk 変換して、変位の計算結果を確認した結果が以下になる。



mises 応力を確認する時は、計算結果が  
 NodalMISESplus\_L1 シェル要素の表側  
 NodalMISESminus\_L1 シェル要素の裏側  
 の2種類あるので注意する。以下が各々のmises 応力の値になる。



シェルの表裏は、メッシュ作成時に決まってしまうが、これを ParaView 上で確認するには、ParaView のメニューバー「Filters」>「Alphabetical」>「Normal Glyphs」を選択して「Apply」ボタンをクリックし、矢印の大きさを調整することで、下図のように確認できる。（矢印が裏から表に向かって表示される。）今回のモデルは、box の外表面が「裏側」になっている事がわかる。



#### 4-9-5. 結果に主応力等を追加する場合

シェルの場合、前項と同様に、元データとなる応力テンソルが以下の様に2種類あるので、必要に応じて使い分ける。

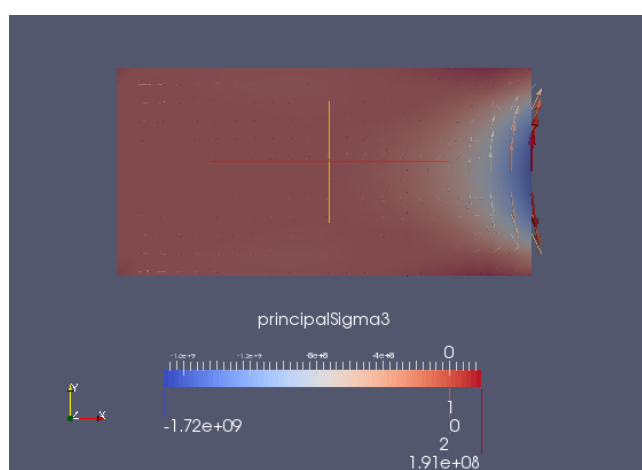
- NodalSTRESSplus\_L1 応力：シェル要素の表側
- NodalSTRESSminus\_L1 応力：シェル要素の裏側
- NodalSTRAINplus\_L1 ひずみ：シェル要素の表側
- NodalSTRAINminus\_L1 ひずみ：シェル要素の裏側

EasyISTR 上は、以下の出力項目設定画面上で、計算の元データとなる応力テンソル名やひずみテンソル名を設定する事ができるので、これを設定した上で、各項目を追加する。

以下は、各々「NodalSTRESSplus\_L1」「NodalSTRAINplus\_L1」を選択している状態になる。



前項で、最大の mises 応力は「NodalMISESminus\_L1」(box 外表面)で発生している為、最大応力は圧縮側になる。この為、NodalSTRESSminus\_L1の最小主応力  $\sigma_3$  を表示させてみる。以下がその結果になる。同時に主応力の方向も表示させている。



## 4-10. ソリッドとシェルの混在モデル解析

前項でシェル（面）で構成されたモデルの解析を行ったが、ここでは、ソリッドとシェルの混在モデルの解析を考えてみる。

現在のところ、EasyISTR で扱えるソリッドと混在可能なシェルは、1次の三角形（761）、1次の四角形要素（781）のみとしている。（2次要素は、扱えない。）

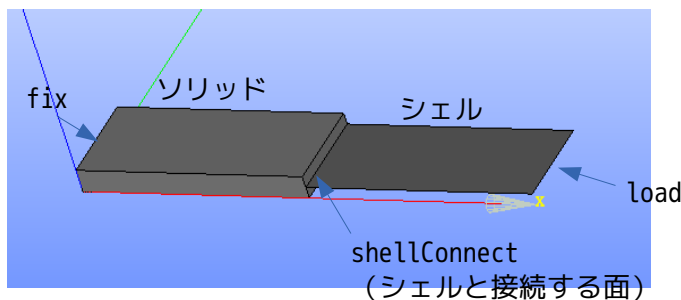
また、この解析は、微小変形に限られる。

### 4-10-1. モデル形状

以下のようなモデルを考える。100 x 20 x 5 の大きさの梁を考え、半分の長さまではソリッドで、それ以降は、シェルとして解析する。

シェルと接続するソリッド側の面を「shellConnect」名でグループ化しておく。シェルをソリッドに接続する為に、特別な処理が必要なため、シェルとの接続面を必ず「shellConnect」名で定義しておく。

また、シェルは、下図の様に shellConnect 面の中心に配置する様にする。



<グループ化>

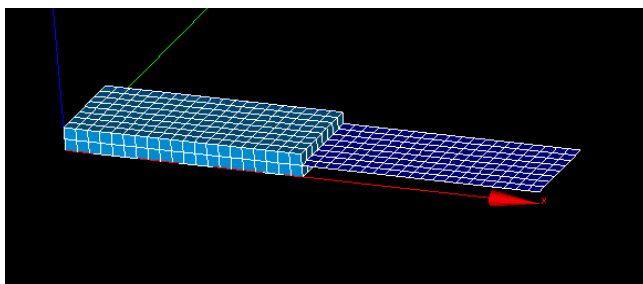
fix: 固定面

shellConnect: 接続面

Load: 荷重セット部

このモデルを以下の様に Salome でメッシュを作成し、「solidShell\_x.unv」として保存する。

尚、このメッシュは、solidとshellは、節点が共有する様にメッシュを作成する。



### 4-10-2. メッシュ変換

作成した unv メッシュを EasyISTR 上で、ファイル名を指定して unv2fistr 変換する。この変換により、シェルとソリッドは、shellConnect 面で変位と回転角が接続された状態になる。

尚、接続面「shellConnect」を定義しなかった場合は、シェルとソリッドは、変位のみ接続された状態であり、回転角は接続されない状態となる。

メッシュ変換により、FistrModel.msh ファイルができ上がり、今回の様にソリッドとシェルが混在しているモデルの場合は、シェルに dummy 節点が追加される。

また、シェルとソリッドの回転角接続の為に、FistrModel.msh ファイル内に、「!EQUATION」が追加され、この方程式で接続される。

----- FistrModel.msh -----

!HEADER

generated by abaqus2fistr.py

!NODE

1, 0.0, 0.0, 0.0  
 2, 0.0, 0.0, 2.5  
 3, 0.0, 0.0, 5.0  
 4, 0.0, 2.5, 5.0  
 5, 0.0, 5.0, 5.0  
 6, 0.0, 7.5, 5.0  
 7, 0.0, 10.0, 5.0

:

744, 52.5, 10.0, 2.5  
 745, 52.5, 12.5, 2.5  
 746, 52.5, 15.0, 2.5  
 747, 52.5, 17.5, 2.5

!NODE, NGRP=dummy

748, 100.0, 0.0, 2.5  
 749, 97.5, 0.0, 2.5  
 750, 97.5, 2.5, 2.5  
 751, 100.0, 2.5, 2.5  
 752, 97.5, 5.0, 2.5

:

932, 50.0, 10.0, 2.5  
 933, 50.0, 12.5, 2.5  
 934, 50.0, 15.0, 2.5  
 935, 50.0, 17.5, 2.5  
 936, 50.0, 20.0, 2.5

メッシュ変換により、  
 shell の dummy 節点が追加される。  
 ( solid、 shell 混在モデルの場合)

!EQUATION

4, 0.0  
 40, 1, 1, 41, 1, -1, 928, 2, 2.5, 928, 3, 0.0  
 4, 0.0  
 108, 1, 1, 109, 1, -1, 936, 2, 2.5, 936, 3, 0.0  
 4, 0.0  
 110, 1, 1, 123, 1, -1, 929, 2, 2.5, 929, 3, 0.0  
 4, 0.0  
 111, 1, 1, 122, 1, -1, 930, 2, 2.5, 930, 3, 0.0

メッシュ変換により、  
 「!EQUATION」が追加される。  
 ( solid、 shell 混在モデルで且つ、  
 nodeGroup 「shellConnect」が存在する場合)

:

EQUATION は、以下の意味がある。  
 最初の 2 行は、以下で記述されている。

式の項数	式の右辺 (定数項)
↓	↙
4, 0.0	
40, 1, 1,	41, 1, -1, 928, 2, 2.5, 928, 3, 0.0
第1項	第2項 第3項 第4項

- 第1項: 節点 40 の x 方向変位の係数「1」
- 第2項: 節点 41 の x 方向変位の係数「-1」
- 第3項: 節点 928 (dummy 節点) の y 軸回転角の係数「2.5」
- 第4項: 節点 928 (dummy 節点) の z 軸回転角の係数「0.0」



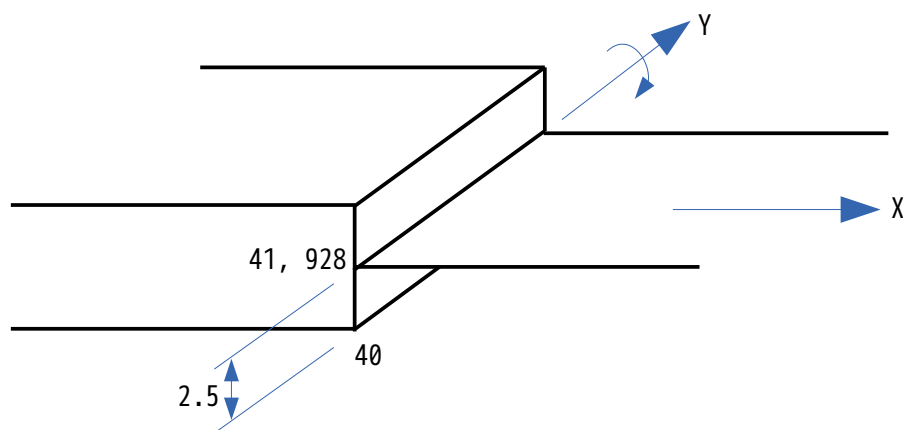
$$d_{40x} - d_{41x} + 2.5\theta_{928y} + 0.0\theta_{928z} = 0.0$$

$$d_{40x} = d_{41x} - 2.5\theta_{928y}$$

この意味は、以下の様に考えると理解できる。

節点 41 と 928 は同じ座標で 928 は dummy 節点になる。

節点 40 の x 方向変位は、41 の x 方向変位に 928 節点の回転角  $\times 2.5$  を減じた値になる。この内容を式で表すと、 $d_{40x} = d_{41x} - 2.5\theta_{928y}$  となる。



この関係式を接続面 (shellConnect) の全節点、shell の全 dummy 節点に対して各方向の関係式を記述した結果が「!EQUATION」の部分になる。この部分は、メッシュ変換 (unv2fistr) 時に設定してくれる。この方程式が成り立つのは、微小変形の場合になる為、solid と shell の混在モデルの解析は、微小変形に限られる事になる。

unv2fistr のメッシュ変換は、以下で変換しており、abaqus2fistr.py 中で、solid と shell の混在モデルの場合、shell に dummy 節点を追加し、さらに「shellConnect」の nodeGroup が存在していれば、EQUATION を追加している。

unv2fistr のメッシュ変換方法

unv 形式	→	abaqus 形式	→	fistr 形式
unv2abaqus.py		abaqus2fistr.py		ここで shell の dummy 節点や EQUATION を追加する

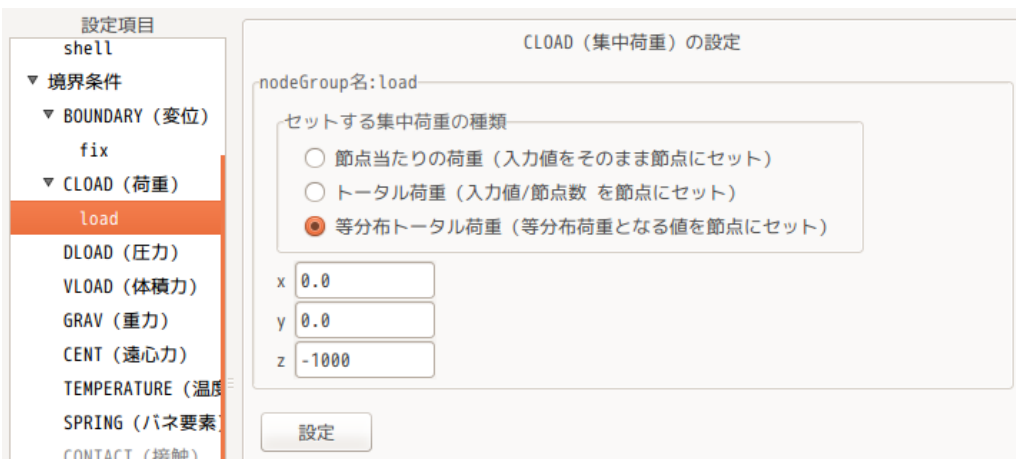
尚、上記のように、メッシュ変換時に!EQUATION を設定しているので、メッシュ変した後に、「スケール変更」をしない様に注意する事。この理由は、「スケール変更」すると、節点座標の値のみスケールが変わり、!EQUATION の寸法は変わらずそのままの為、方程式が成り立たなくなる為。

#### 4-10-3. 材料、境界条件の設定

材料は、以下で設定している。

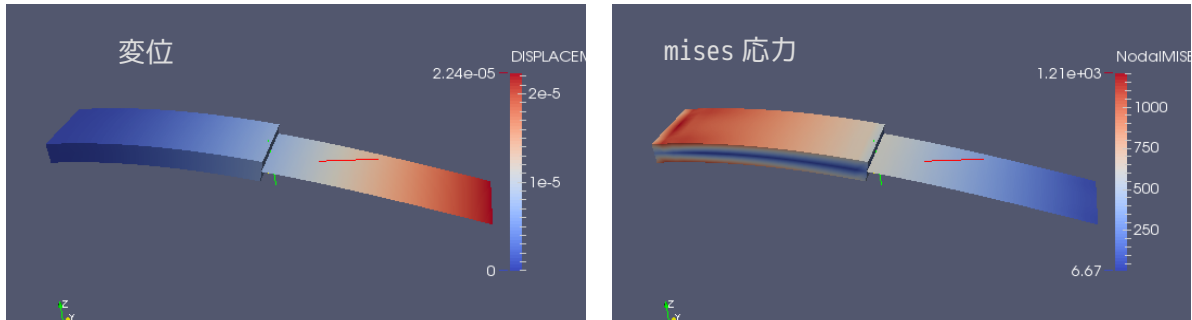


境界条件は、fix を固定し、load の等分布トータル荷重-1000 を入力して計算。

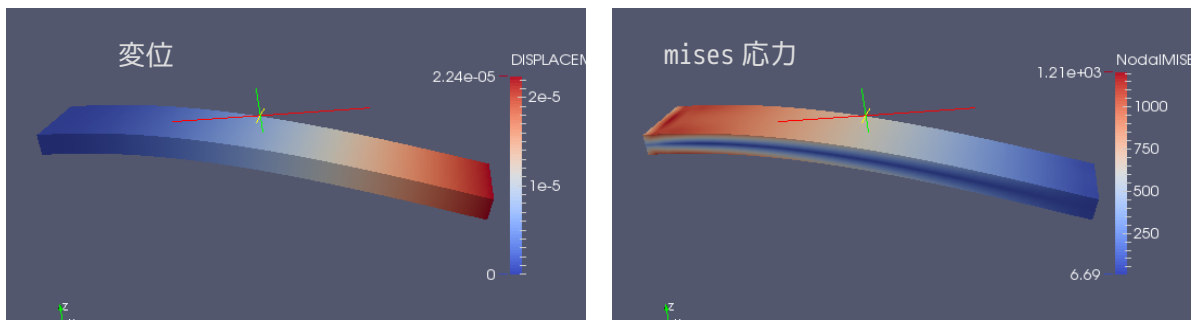


#### 4-10-4. 計算開始、結果の確認

線形 solver を、「MUMPS」または「DIRECT」に設定して、計算を開始させる。  
計算結果は、以下になる。



確認のため、全てソリッドで同じメッシュサイズ、同じ材料、同じ境界条件で計算させた結果が以下になる。  
最大変位と最大 mises 応力とも同じ値であることが判る。



更に、確認のため、半円環状のモデルで、solid+shell と all solid でも確認した。このモデルは、shell との接続部に角度が生じているので、EQUATION が複雑になってくる。このモデルの計算結果が以下になるが、上記同様に変位の値がほぼ合致している。  
この為、!EQUATION の拘束方法は、正しく設定されていると言える。

