



EasyISTR 操作マニュアル

Ver 3.03.190505

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。

19/05/05 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。
ver-2.23.161127	beam (611、641) を追加して、solid、shell、beam の混在解析が可能。
ver-2.24.170109	beam、shell にモーメントの設定 (!CLOAD) を追加。 等分布荷重 (!CLOAD) の設定を高速化。
ver-2.25.171008	非線形解析、動解析時の各 step の状況をモニタする plotStepMonitor を追加。 条件設定時に settingDialog を表示。 大規模解析用に、処理を高速化。
ver-2.30.171218	非線形解析時の解析における step 解析が複数 step 設定できる様に修正。
ver-2.40.180406	複数 step 設定のバグ修正。vtk 変換の高速化。NGRP の重複定義削除追加。 複数接触の事例追加。 熱応力解析のバグ修正。(初期温度の読み取りが間違っていた。) 「初期温度」を「境界条件」の外に移動。
ver-2.40.180608	ubuntu-18.04 に対応。接触解析の並列処理方法を追加。
ver-2.41.180716	workFolder 移動先に必要な file が存在しない時に初期化を追加。(バグ修正) abaqus2fistr 変換時、小文字の keyword に対応。
ver-2.41.180901	物性値読み込み時、指数表記のチェック追加。反力算出を追加。 noutilus を裏で起動する様に修正 (ubuntu18.04 対応)。
ver-2.42.181117	vtk 変換時、要素 No の取得方法を修正。(バグ修正) 応用事例として、接触を含む弾塑性解析を追加。
ver-2.50.181211	大規模メッシュ対応として、tempFolder 内に必要なファイルを縮めて保存し、 この内容を実行する事に様に修正。これにより、大規模メッシュの操作性が 大幅に向上。
ver-2.51.181230	plotStepMonitor を非定常熱伝導解析にも対応させる。
ver-3.01.190328	GUI を GTK+2 から GTK+3 に変更し、python も python2 から python3 に変更。 これに伴い、windows 版においても 64bit の python3 が使用できたので Linux 版と同 様に大規模モデルが対応可能。

ver-3.03.190505

unv メッシュ変換時のバグ修正、高速化。
windows 版の場合、unv 変換が極端に遅かったので、これを高速化。

目次

1. EasyISTRとは.....	7
1-1. 概要.....	7
1-2. 起動画面.....	7
2. インストール.....	8
2-1. linux へのインストール.....	8
2-2. windows へのインストール.....	9
2-3. 起動方法.....	10
2-4. インストール後の設定.....	10
3. 操作方法 (片持ち梁の線形弾性静解析)	12
3-1. workFolder の設定.....	12
3-2. メッシュ変換 (unv 形式 → fist 形式)	14
3-3. メッシュ形状の確認.....	16
3-3-1. nodeGroup の表示.....	17
3-3-2. faceGroup の表示.....	18
3-3-3. elementGroup の表示.....	19
3-3-4. nodeNo、elementNo の表示.....	19
3-4. 解析の種類を設定.....	20
3-5. 材料物性値の設定.....	21
3-6. 境界条件の設定.....	23
3-6-1. 固定部の設定.....	23
3-6-2. 節点荷重の設定.....	25
3-6-3. 面圧力の設定.....	26
3-6-4. 境界条件の再確認.....	26
3-6-5. 等分布節点荷重を設定する場合.....	27
3-7. solver の設定.....	27
3-7-1. 線形 solver の設定.....	28
3-7-2. 出力項目の設定.....	28
3-7-3. 計算開始.....	29
3-8. 結果の可視化.....	30
3-8-1. FrontISTR の出力結果を可視化.....	30
3-8-2. 主応力、主ひずみ、mises 応力、mises ひずみを追加する場合.....	34
3-9. 並列処理方法.....	38
3-9-1. 並列処理設定、計算開始.....	38
3-9-2. 並列計算結果の可視化.....	40
3-10. EasyISTR 下部ボタンの動作.....	41
3-10-1. 「folde 開く」ボタンの動作.....	41
3-10-2. 「temp 開く」ボタンの動作.....	42
3-10-3. 「cnt,msh 編集」ボタンの動作.....	42
3-10-4. 「端末起動」ボタンの動作.....	42
3-10-5. 「temp 設定保存」ボタンの動作.....	43
3-10-6. 「folder 内クリア」ボタンの動作.....	43
3-11. ファイル変換について.....	44
3-11-1. unv2fistr 変換 (unv 形式のメッシュを FrontISTR 用に変換)	44
3-11-2. abaqus2fistr 変換 (abaqus 用 inp ファイルを FrontISTR 用に変換)	45
3-11-3. fistr2vtk 変換 (FrontISTR 用のファイルを paraView 用に変換)	48
3-11-3-1. メッシュファイルの変換.....	48
3-11-3-2. 結果ファイルの変換.....	54
3-12. salome によるメッシュの作成方法.....	55
3-12-1. ジオメトリの作成.....	55
3-12-2. メッシュの作成.....	57
3-13. 等分布節点荷重の算出方法.....	61
3-13-1. 面の等分布節点荷重の算出.....	61

3-13-2.	線分の等分布節点荷重の算出.....	63
3-13-3.	点の等分布節点荷重の算出.....	64
3-13-4.	等分布トータル荷重の設定結果の確認.....	64
3-14.	その他.....	66
3-14-1.	plotStepMonitor について.....	66
3-14-2.	複数 step 解析について.....	68
3-14-3.	反力の算出について.....	71
3-14-4.	大規模メッシュ作成 (既存メッシュの細分化)	72
4.	応用事例.....	74
4-1.	接触解析.....	74
4-1-1.	モデル形状.....	74
4-1-2.	変位拘束の接触解析.....	75
4-1-3.	荷重拘束の接触解析: 弱いばね追加.....	83
4-1-4.	すべりのあるモデル (スナップフィット) の接触解析.....	86
4-1-5.	多数の接触部が存在する場合の接触解析.....	91
4-1-6.	接触解析における並列処理方法.....	102
4-2.	弾塑性解析.....	103
4-2-1.	モデル形状 (円柱)	103
4-2-2.	圧縮 20% の計算.....	103
4-2-3.	引き続き引張 20% の計算.....	110
4-2-4.	複数 step による一括解析.....	113
4-3.	固有値解析.....	119
4-3-1.	モデル形状.....	119
4-3-2.	固有値解析の開始.....	120
4-4.	周波数応答解析.....	123
4-4-1.	モデル形状.....	123
4-4-2.	周波数応答解析の開始.....	123
4-5.	時刻歴応答解析 (陽解法)	128
4-5-1.	モデル形状.....	128
4-5-2.	境界条件一定値 (線形・陽解法)	128
4-5-3.	境界条件に時間変化を与える (線形・陽解法)	134
4-6.	時刻歴応答解析 (陰解法)	139
4-6-1.	モデルと材料.....	139
4-6-2.	境界条件一定値 (線形・陰解法)	139
4-6-3.	境界条件に時間変化を与える (線形・陰解法)	141
4-7.	接触を含む時刻歴応答解析.....	146
4-7-1.	モデル形状.....	146
4-7-2.	設定内容.....	147
4-7-3.	計算開始、結果の確認.....	149
4-8.	接触を含む弾塑性解析 (板のプレス曲げ加工)	151
4-8-1.	モデル形状.....	151
4-8-2.	設定内容.....	152
4-8-3.	計算開始、結果の確認.....	158
4-9.	熱応力解析.....	160
4-9-1.	モデル形状.....	160
4-9-2.	熱応力解析の開始.....	161
4-10.	熱伝導解析 (静解析)	164
4-10-1.	モデル形状.....	164
4-10-2.	熱伝導解析 (静解析) の開始.....	164
4-10-3.	熱伝導解析結果から熱応力を求める.....	167
4-11.	熱伝導解析 (動解析)	170
4-11-1.	モデル形状.....	170
4-11-2.	熱伝導解析 (動解析) の開始.....	170
4-11-3.	熱伝導解析結果から熱応力を求める.....	172

4-12. シェルの解析.....	175
4-12-1. モデルの作成、メッシュ読み込み.....	175
4-12-2. 材料の設定.....	176
4-12-3. 境界条件の設定.....	177
4-12-4. 計算開始、結果の確認.....	177
4-12-5. 結果に主応力等を追加する場合.....	178
4-13. ソリッドとシェルの混在モデル解析.....	180
4-13-1. モデル形状.....	180
4-13-2. メッシュ変換.....	180
4-13-3. 材料、境界条件の設定.....	182
4-13-4. 計算開始、結果の確認.....	184
4-14. 梁 (beam) 要素の解析.....	185
4-14-1. 梁のたわみ解析.....	185
4-14-2. 丸棒のねじり解析.....	188
4-15. ビーム、シェル、ソリッド混在モデルの解析.....	193
4-15-1. 梁のたわみ解析.....	193
4-15-2. シェルとビームのモデルの場合.....	199

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 (オープンソース)	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
gnuplot (オープンソース)	非線形解析や動解析実行時に、その実行状況をグラフ表示させる ver 5.0 以降

上記が実現できるように、EasyISTR を GTK+3 と python3 を使って作成している。
(ver 3.00 以降より。それ以前は、GTK+2 と python2 で作成。)

また、大規模メッシュに対応するため、仮の folder 内にファイルサイズを縮小した制御ファイルやメッシュファイルを作り、この folder を対象に操作する様にしている。(ver 2.50 以降)
これにより、サイズの大きいメッシュファイルを直接アクセスする必要がなくなり、操作性を大幅に向上させている。

仮の folder は、「easyIstrUser/data/temp」に設定しており、この folder 内に操作対象 folder 内のファイルイメージを縮小して、そのままコピーされている。

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、python3とGTK+3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

また、EasyISTRをフルに使用するのであれば、他にSalomeやparaView、gnuplot(いずれもオープンソース)のインストールが必要になる。

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_3.03.190505_all.deb
```

```
<RedHat系>
$ su
パスワード:
# rpm -i easyistr_3.03.190505-2.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は、python3.6とPyGObject (GTK+3) で作成しているため、これらがインストールされている必要がある。

これらのpython3とPyGObjectをインストールする方法は、

「https://pygobject.readthedocs.io/en/latest/getting_started.html」に記述がある。

<python3とPyGObjectのインストール方法>

まず、msys2をインストールする。「<http://www.msys2.org>」からwindows用のインストーラをダウンロードして、インストーラを実行する。実行すると端末が開くので以下を入力する。

```
$ pacman -Syu
```

この後、C:\msys64\mingw64.exeを実行して以下を入力して最新状態にする。

```
$ pacman -Su
```

以上で最新状態のmsys2がインストールできた事になる。これらの方法は、「<http://www.msys2.org>」に記述がある。

msys2のインストール後、再度端末「C:\msys64\mingw64.exe」を起動して以下を入力し、pythonとPyGObjectをインストールする。

```
$ pacman -Suy
```

```
$ pacman -S mingw-w64-x86_64-gtk3 mingw-w64-x86_64-python3-gobject
```

この操作により、64bit版のpython3とPyGObjectがインストールできた事になる。

EasyISTRのインストールに当たっては、圧縮ファイル「eastistr-3.03-190505.zip」をインストールしたいフォルダで展開する。

展開後、batファイルを修正する。batファイルは、ver 3.02以前は、easyistr.batのみだったが、ver3.03以降は、起動用「easyistr.bat」と環境設定用「easyistrEnv.bat」に分かれている。

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

----- easyistr.bat の内容 -----

```

:
cd C:\DEXCS\easyistr          #インストールフォルダ (このフォルダが%easyIstrPath%)

```

```

rem #環境変数の設定
call easyistrEnv.bat          #環境変数を設定

```

```

rem #以下の環境変数を設定
rem %easyIstrPath%          #currrdir に設定
rem %HOME%                  #C:\DEXCS に設定
rem %easyIstrUserPath%     #ユーザ path
rem %PATH%
rem %PYTHONPATH%
:

```

「easyistrEnv.bat」は、起動ファイル「easyistr.bat」から呼び出され、ここでPATHなどの各環境変数を設定している。

----- easyistrEnv.bat の内容 -----

```

:
set easyIstrPath=%CD%        #カレントディレクトリに設定

```

```
set HOME=C:\DEXCS          #C:\DEXCS に設定
set easyIstrUserPath=%HOME%\easyIstrUser
set PATH=C:\DEXCS\FrontISTR;C:\DEXCS\ParaView\bin;%PATH%
set PATH=C:\msys64\mingw64\bin;%PATH%          #python3 への PATH を追加
set PYTHONPATH=%easyIstrPath%\bin;%easyIstrPath%\python;%PYTHONPATH%
:
```

この2つの環境変数の設定を修正して、このeasyistr.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 と PYTHONPATH を通している。

python3 への path は、bat ファイル中で設定している。また EasyISTR 内では、python スクリプトをフルパスで指定しているので、「.py」を python3 に関連付けしなくても作動する。

2-3. 起動方法

起動は、以下の実行ファイルを実行すると EasyISTR が起動する。

```
Linux:          easyistr
windows:       easyistr.bat
```

もし、エラーが発生して起動しない状態に陥った場合は、

```
Linux:          $easyIstrUserPath/data      (~\easyIstrUser\data)
Windows:       %easyIstrUserPath%\data     (C:\DEXCS\easyIstrUser\data)
```

をフォルダ毎削除して、再度起動してみる。それでも起動しない場合は、easyIstrUser フォルダをフォルダ毎削除して起動してみる。

これらフォルダ内には、起動のための設定ファイルが保存されており、この内容が壊れた場合は、起動しない恐れがある。

EasyISTR は、起動時にこれらフォルダの存在を確認し、存在しない場合は、デフォルトの値に設定し直して、起動する様にしている。

2-4. 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。

```
----- 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
  REVOCAP     revocap
  gnuplot     gnuplot

# 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
  REVOCAP     revocap
  gnuplot     "C:\Program Files (x86)\bConverged\gnuplot\bin\gnuplot.exe"
```

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。

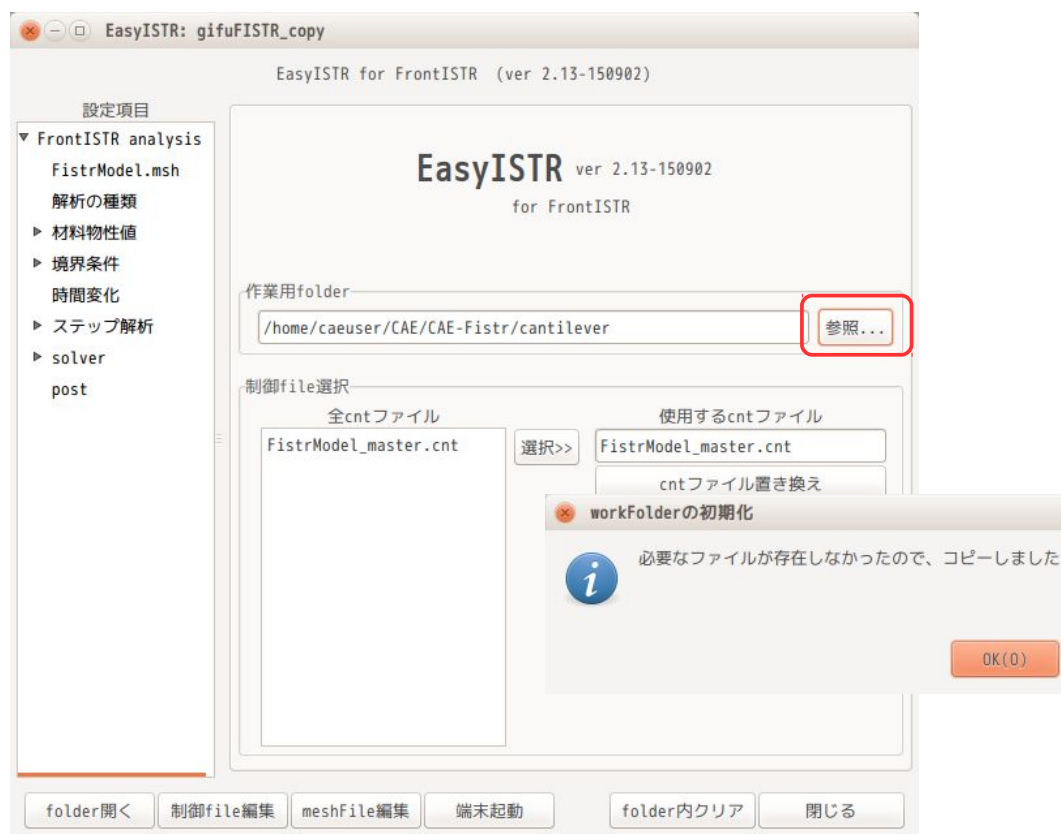
EasyISTR の操作方法に関しては、片持ち梁の線形弾性静解析を例にとって説明する。基本的には、画面上で入力して、最後に「設定」ボタンをクリックして確定させる方法をとっている。この「設定」ボタンのクリックは、その設定内容が制御ファイル等のファイルに書き込まれる事を意味している。

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ヶのファイルは、「~/easyIstrUser」フォルダ内にある3ヶ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ヶ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。

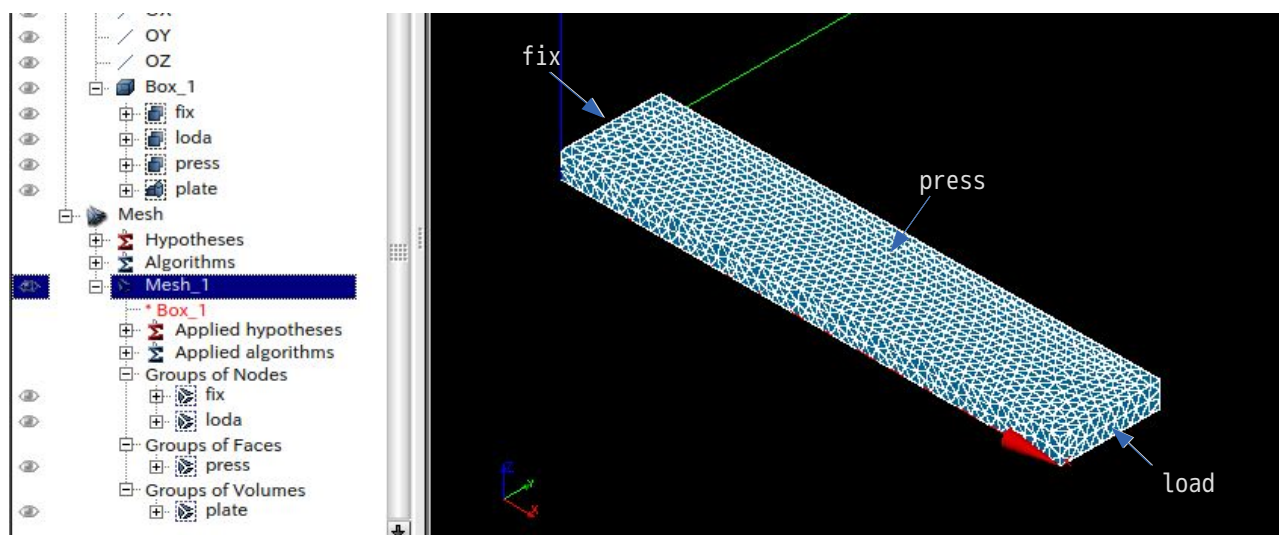
モデルサイズは、100 x 20 x 5 mm で作成している。

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。

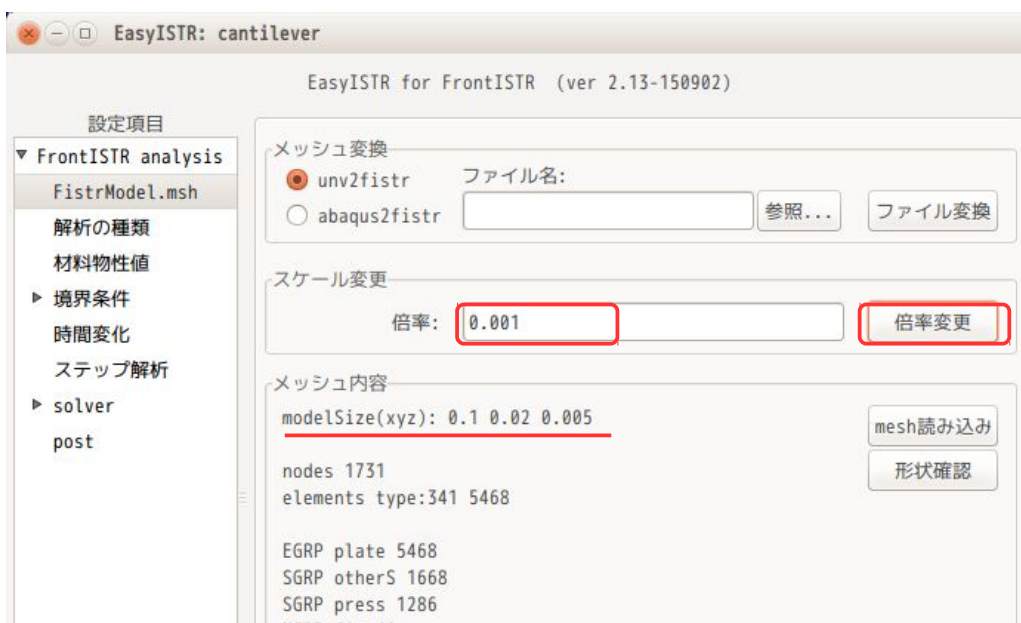
メッシュ変換後、EasyISTR のメッシュ内容枠内に、メッシュの概要が以下の様に表示されている。

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 36	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。

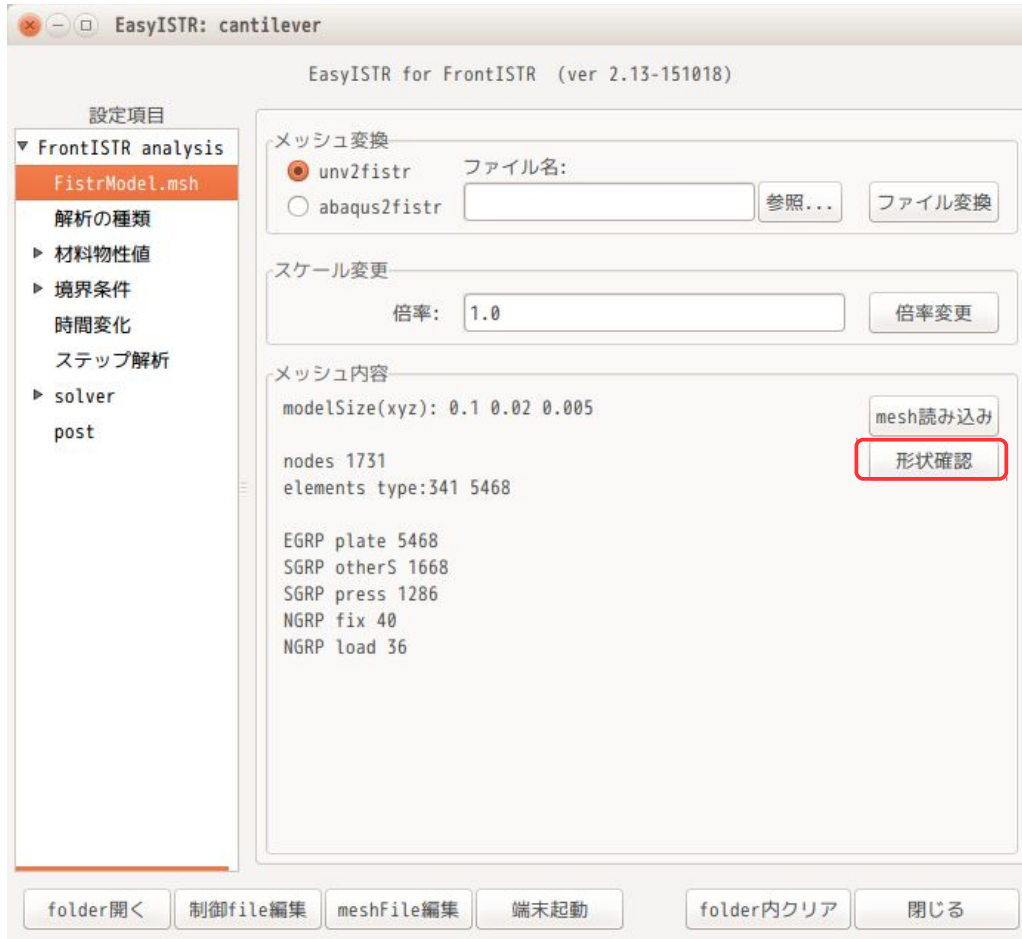
モデルのスケールを変更するためには、下図のようにスケール変更枠内の倍率を「0.001」に設定し、「倍率変更」ボタンをクリックする事で実現できる。この操作により、modelSize(xyz)の値が 1/1000 の値に修正されている。



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。

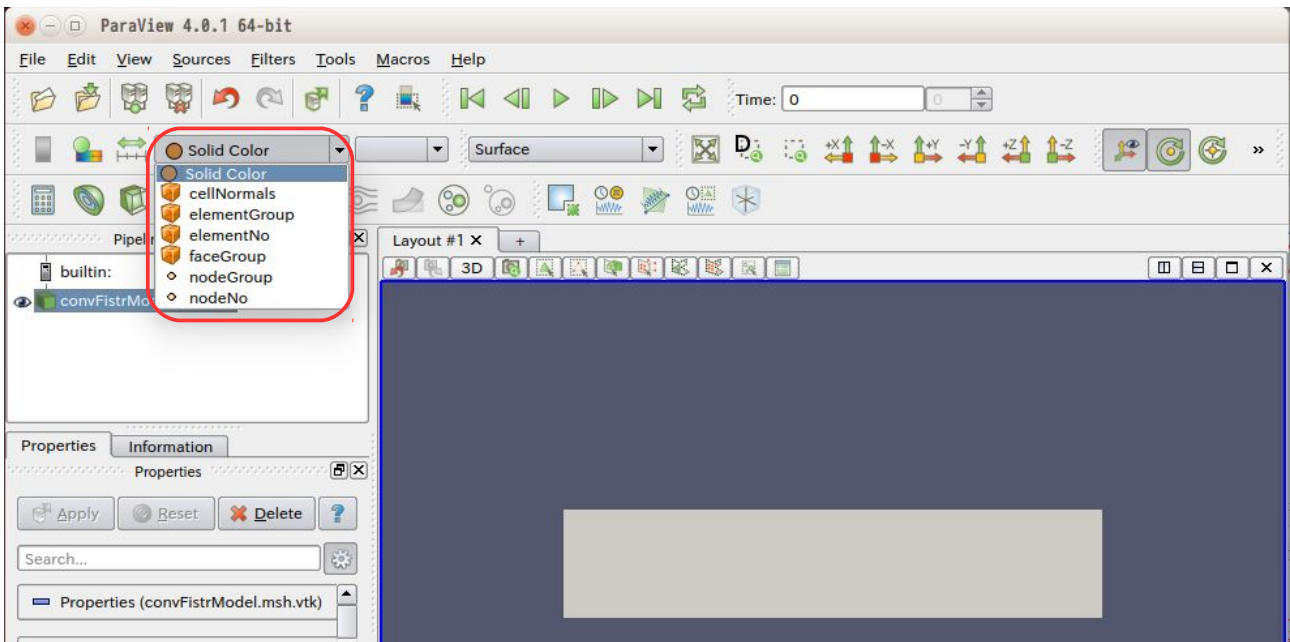
メッシュデータの vtk 変換は、メッシュ側で定義している要素、面、節点グループが表示できるように、volume 要素、モデル表面の face 要素、節点を vtk 形式で作成して、これを paraView が表示している。

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

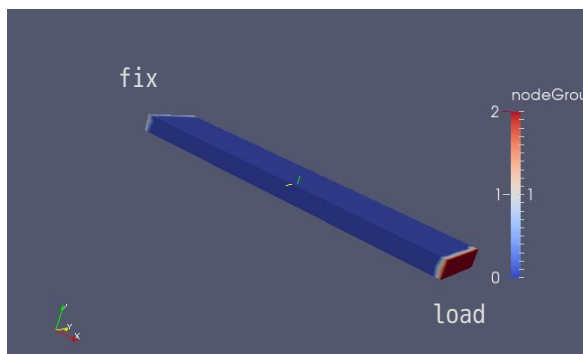
nodeGroup は、EasyISTR の画面上から、定義順に以下の nodeGroup が確認できる。

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	

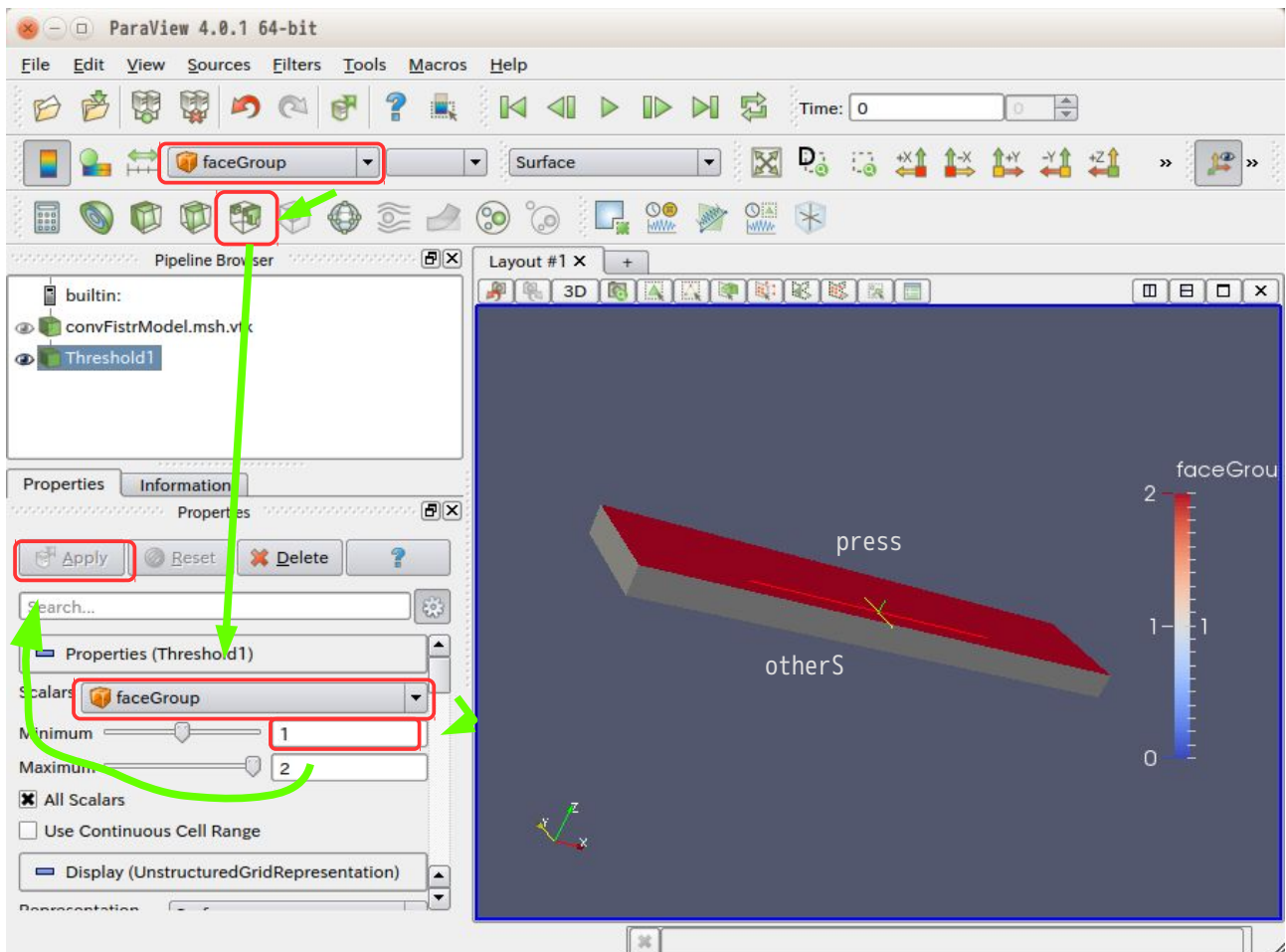
メッシュデータを vtk 変換した後は、要素 (volume 要素と face 要素) と節点の情報がある。faceGroup を表示するためには、この要素を表示する必要がある。face (外表面) を表示させる為には、volume 要素が邪魔になるので、volume 要素には、faceNo を「0」に設定している。

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。

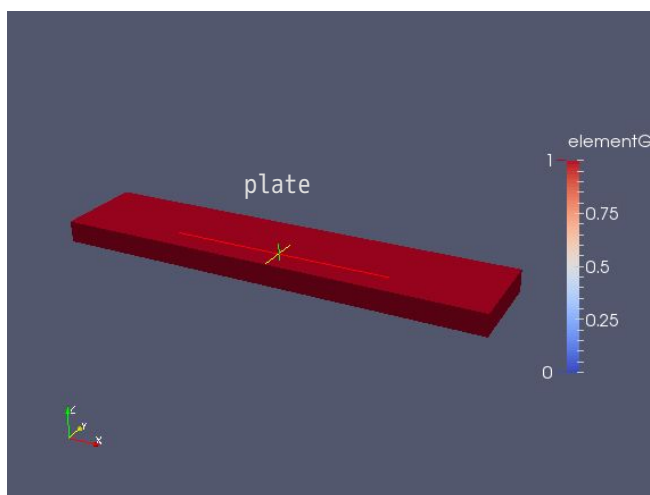
この elementGroup を paraView で表示させる為には、要素 (volume 要素と face 要素) を表示させれば良いが、前項とは反対に face 要素が邪魔になる為、face 要素には elementNo に「0」を与えている。この為、今回のモデルの elementNo は、以下の状態になる。

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。

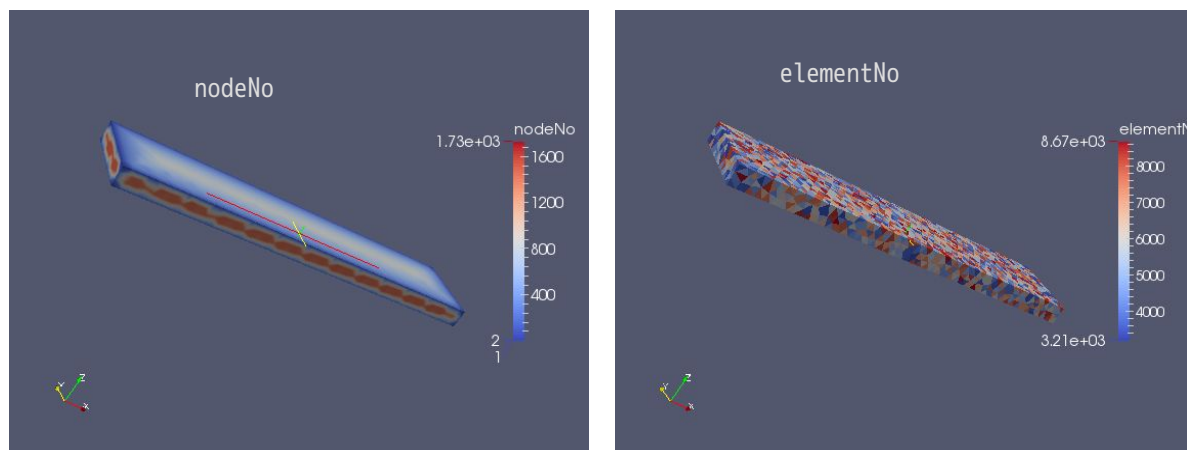
下図が今回の elementGroup 形状になる。



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。

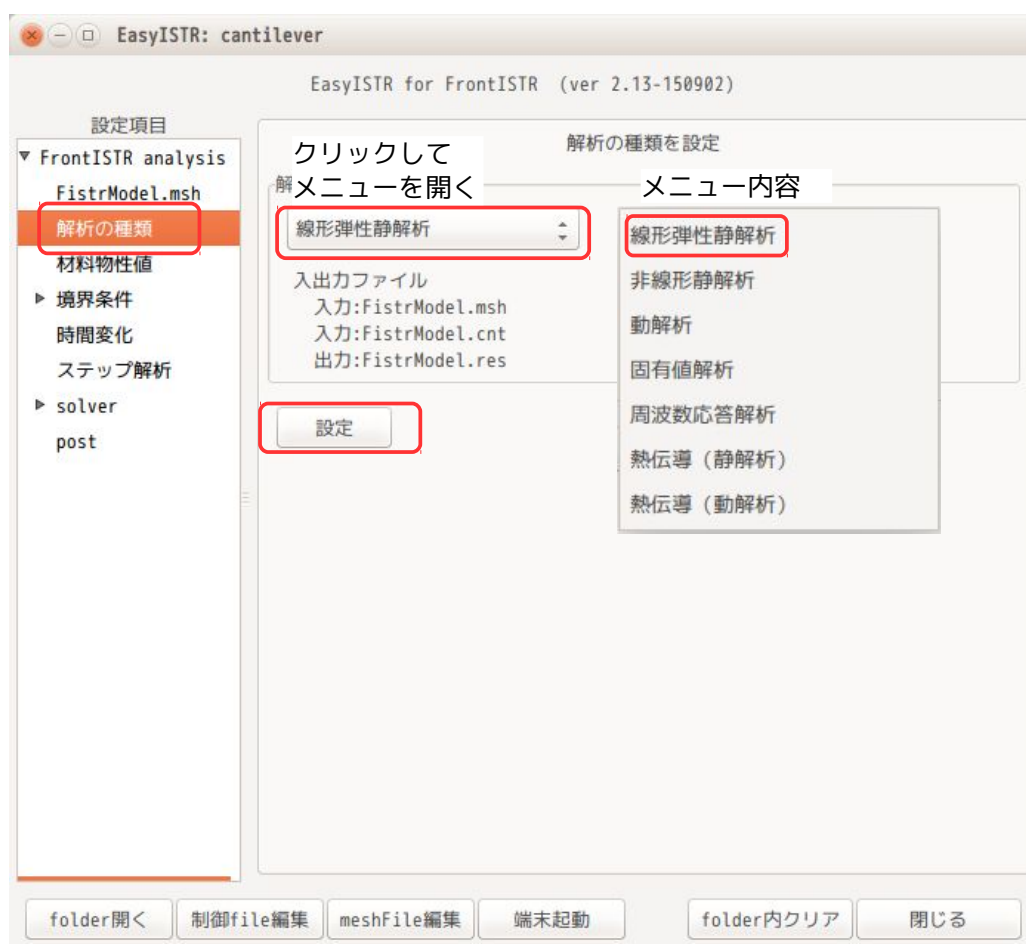


3-4. 解析の種類を設定

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に変わる。

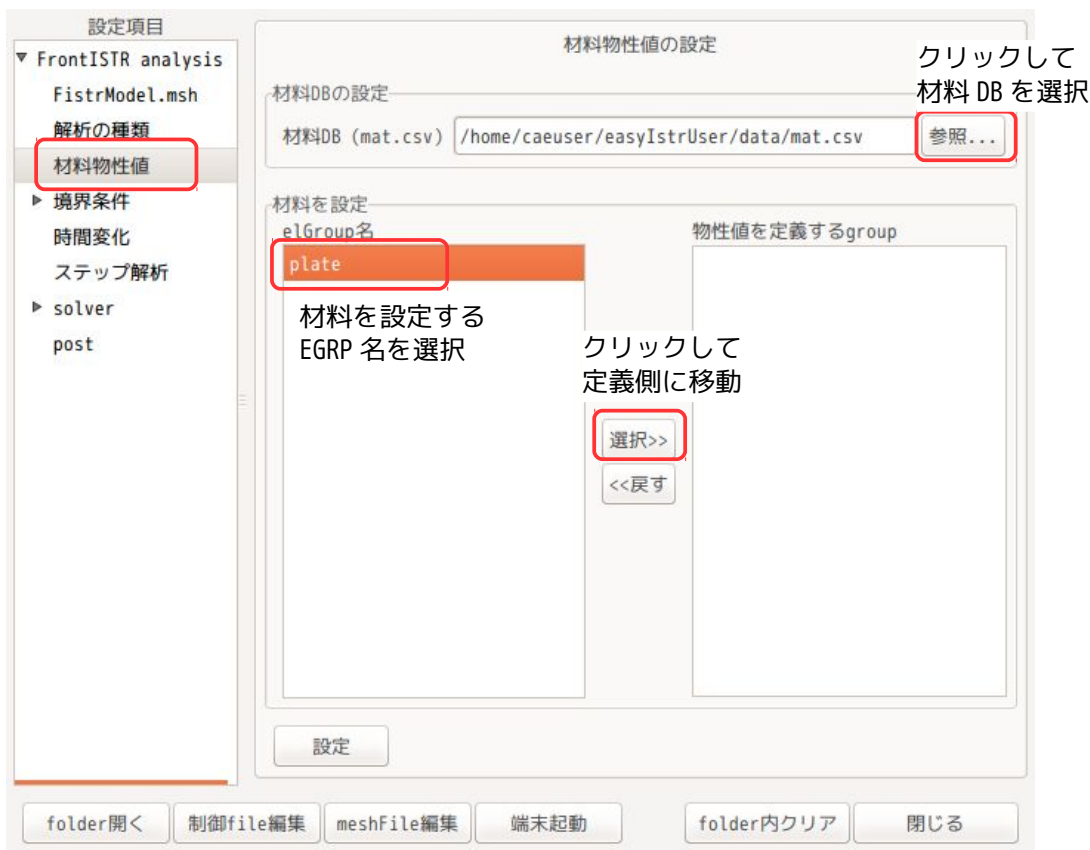
まず、材料 DB を確認する。デフォルトの設定は、「~/easyIstr/data/mat.csv」で設定されている。これは、EasyISTR インストール後の初回起動時に、このフォルダ内に材料 DB が作成され、これがデフォルトの設定になっている。

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 根本端面 (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。

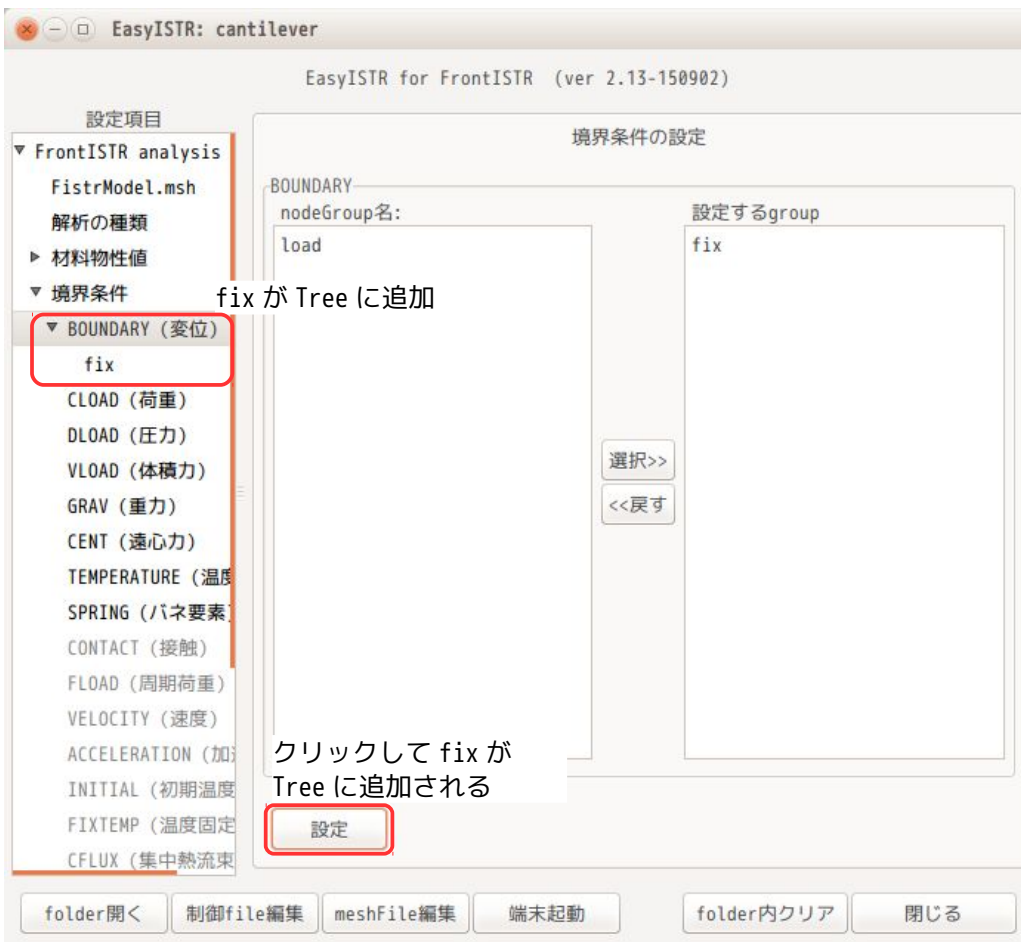
3-6-1. 固定部の設定

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 (1GPa) 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. 等分布節点荷重を設定する場合

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、シェル、ビームの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。）

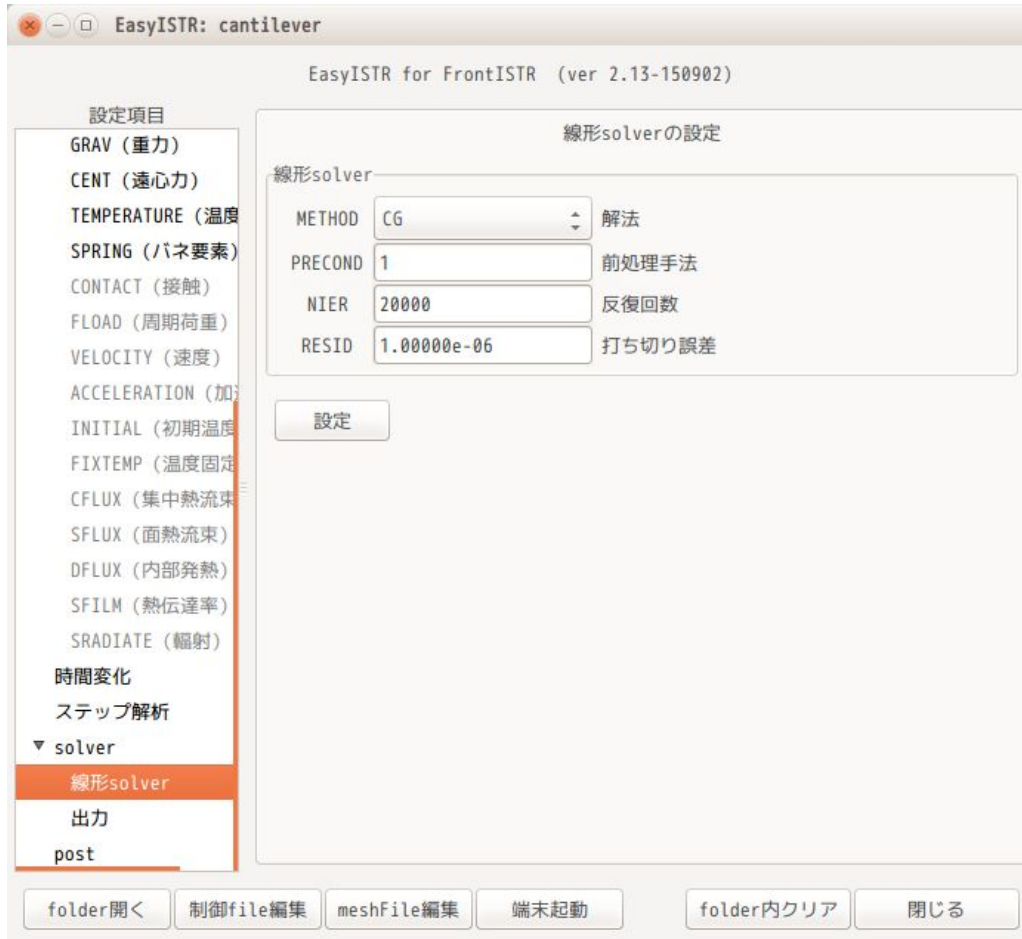
3-7. solver の設定

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。

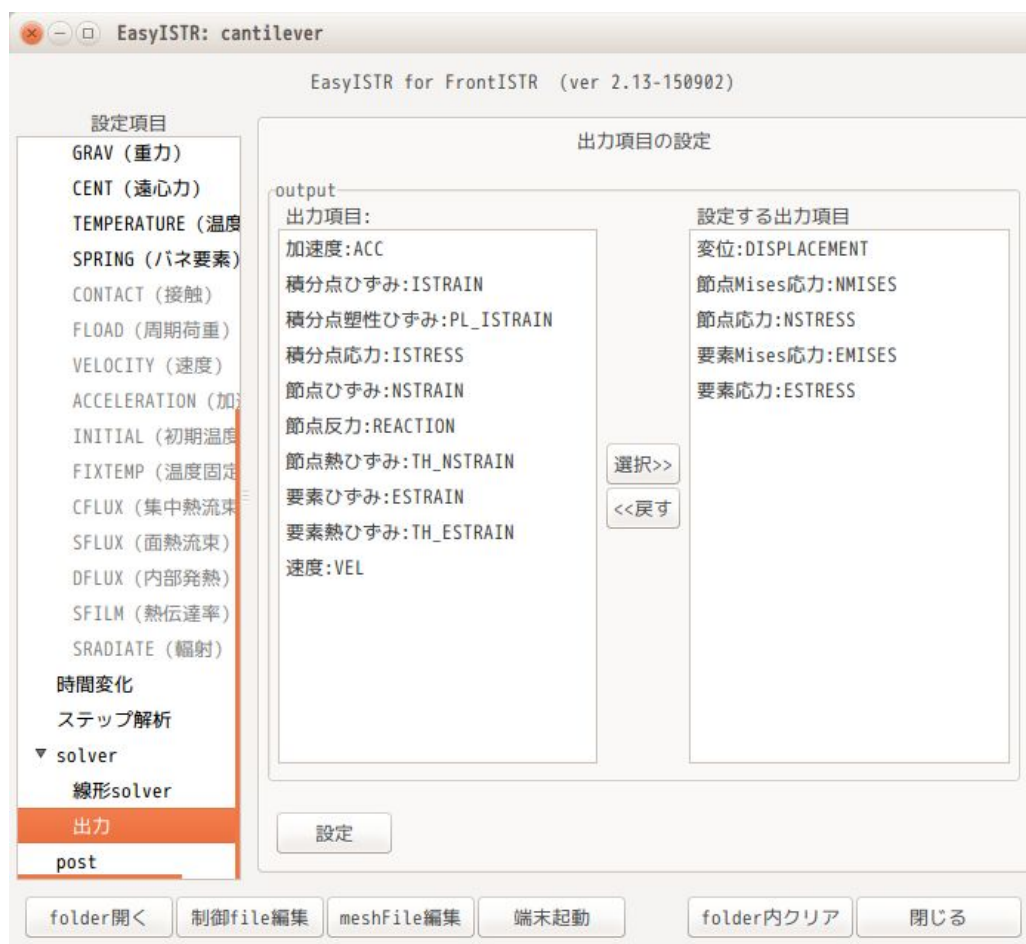
もし、収束しないようであれば、解法や反復回数を変更する。



3-7-2. 出力項目の設定

計算結果の出力項目を確認する。

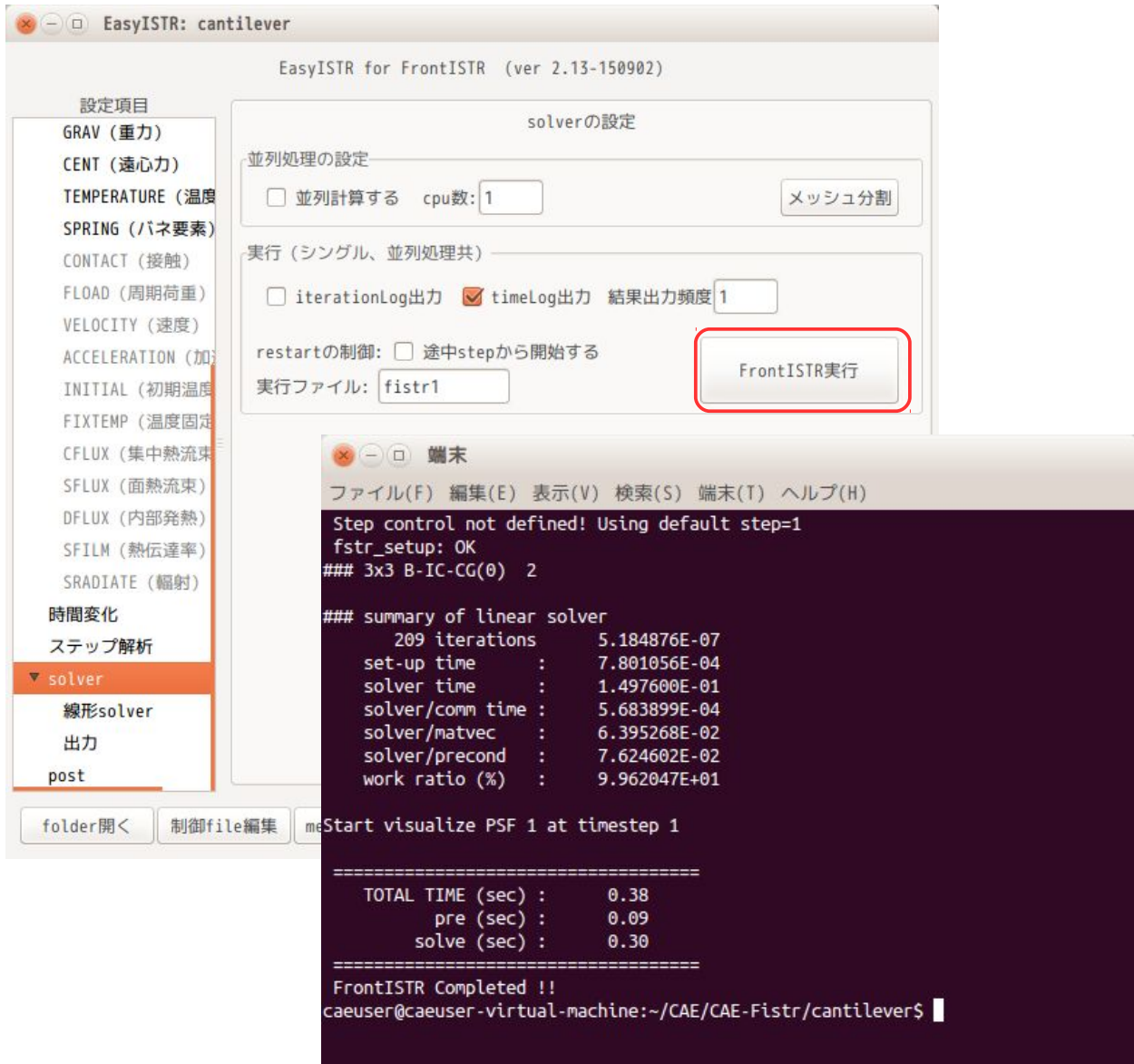
設定項目 Tree 内の「出力」を選択すると、現在設定されている出力項目が表示される。出力項目を追加、削除する場合は、「選択>>」ボタンで追加、「<<戻す」ボタンで削除ができる。



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

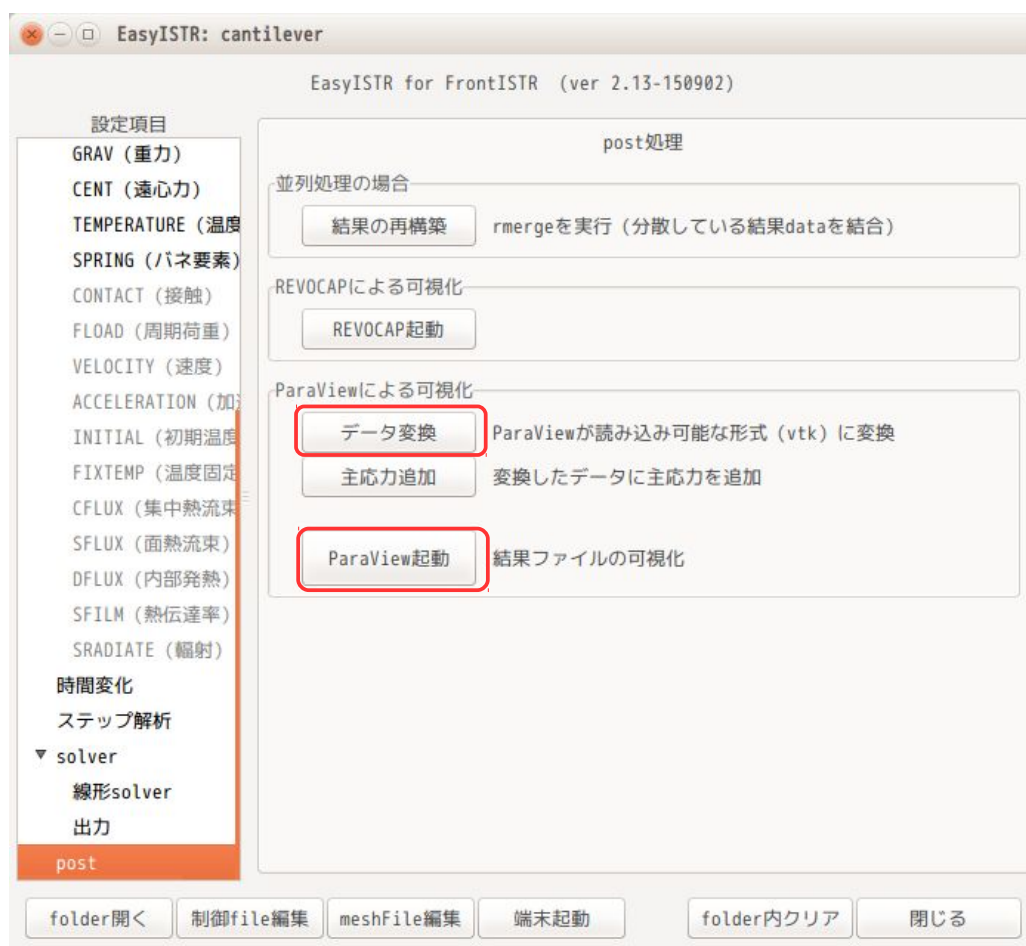
設定項目 Tree 内の「post」を選択して、計算結果を可視化する。

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

熱伝導（静・動）解析

convFistrModel_temp.res.<xxxx>.<n>.vtk

シングル:0

上記以外

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

<n>

計算 step 数(1 ...)

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

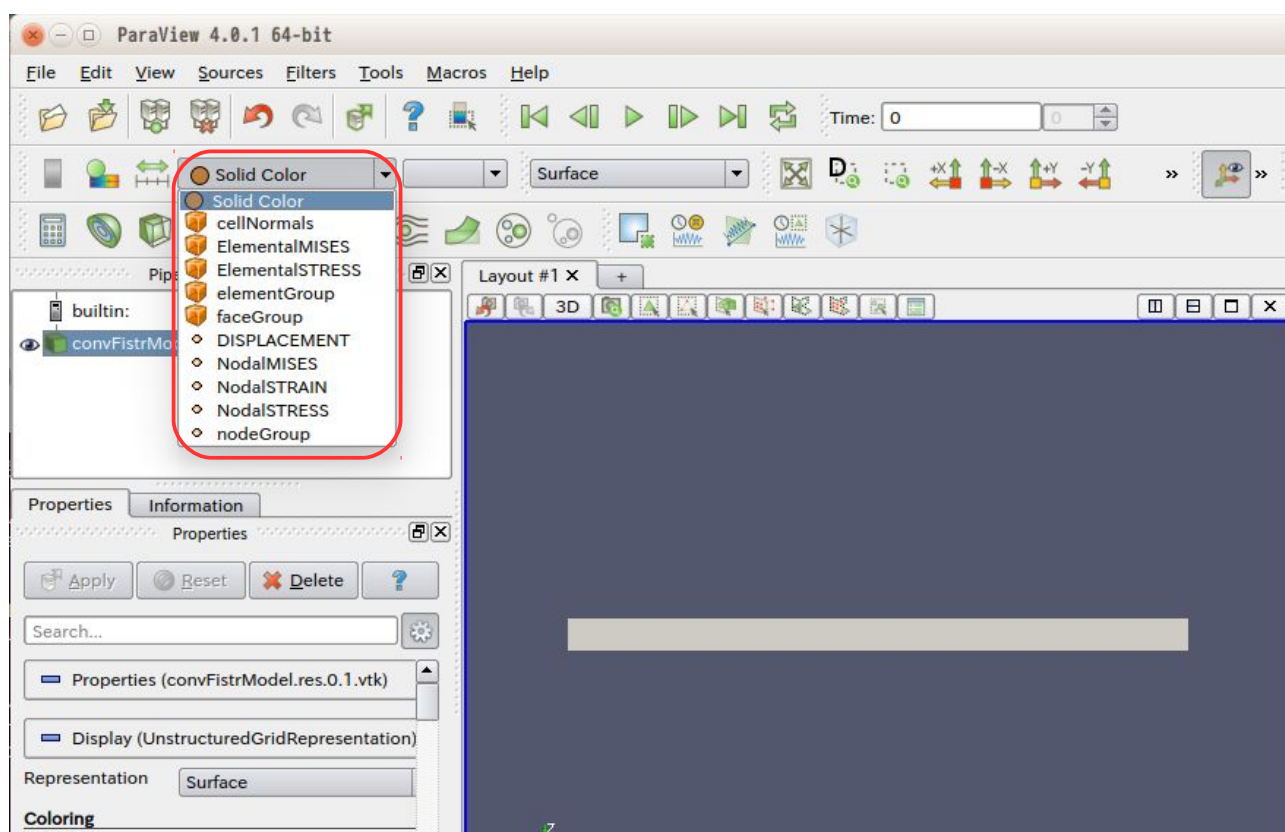
要素 Mises 応力

要素応力

paraView からは、下図の様に、

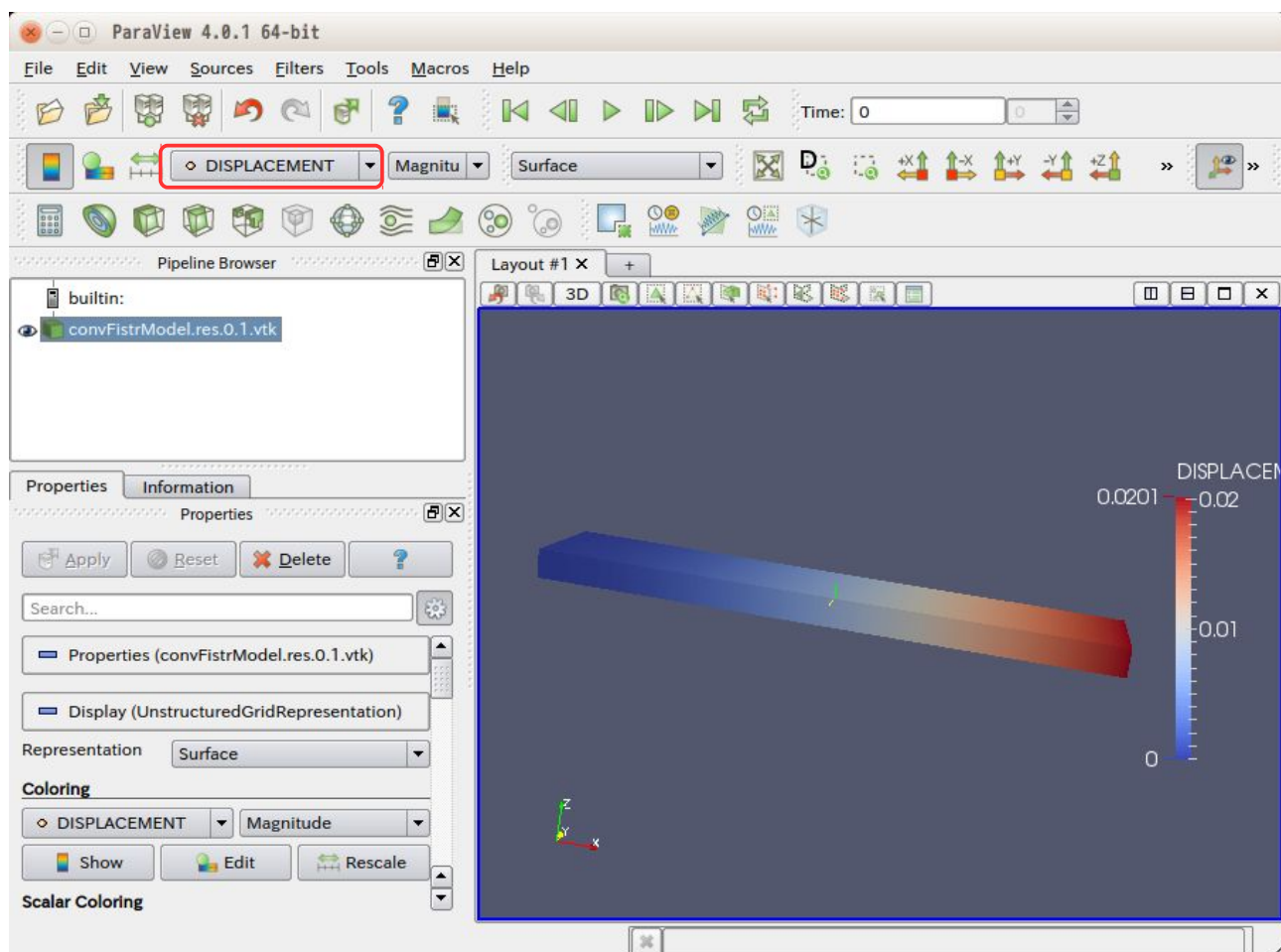
ElementalMISES	要素 Mises 応力
ElementalSTRESS	要素応力
DISPLACEMENT	変位
NodalMISES	節点 Mises
NodalSTRESS	節点応力

が選択できる。



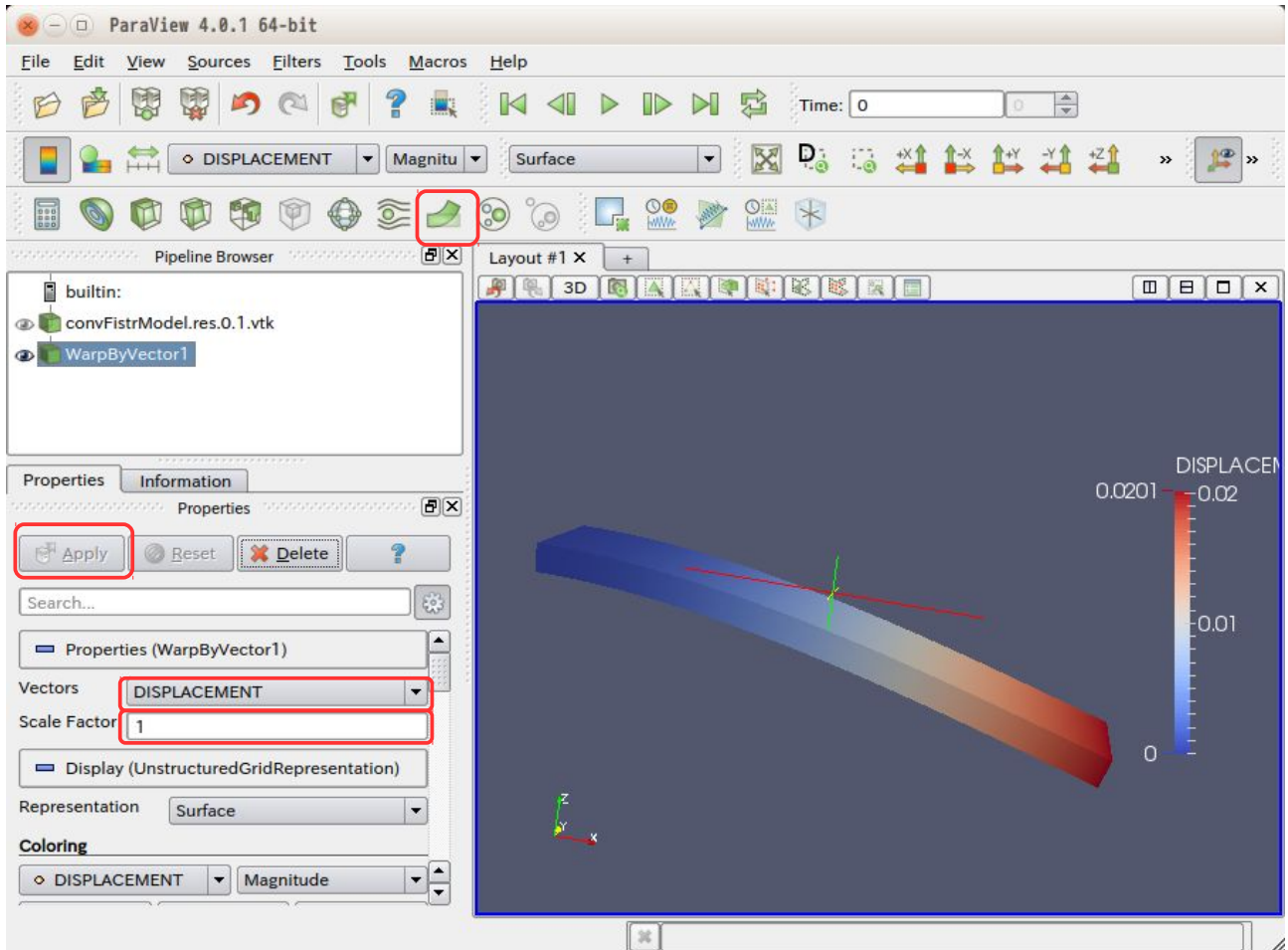
ここで、変位 (DISPLACEMENT) を読み込んで表示させてみる。

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 ひずみが出来ない。(mises 応力は、FrontISTRが出来ない。)

この為、これら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。

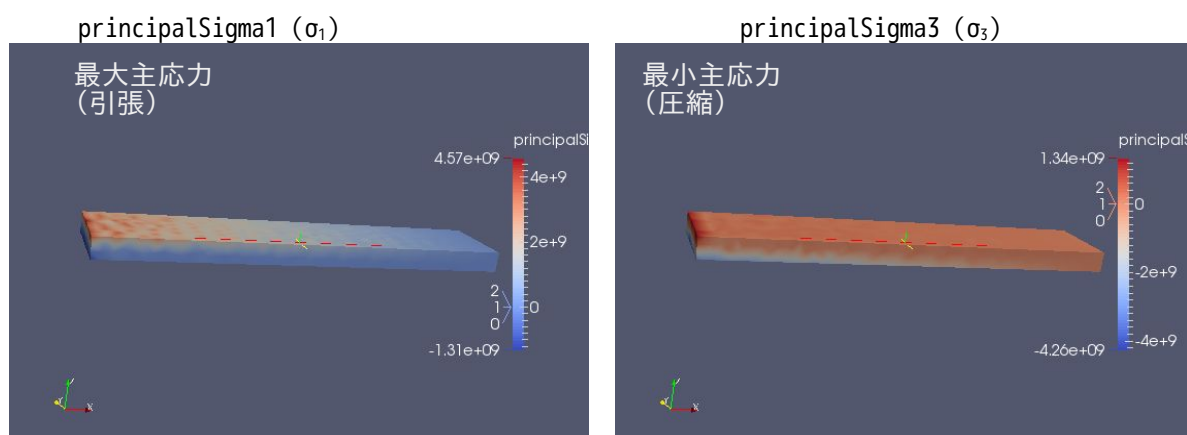
尚、これら出力項目は、

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

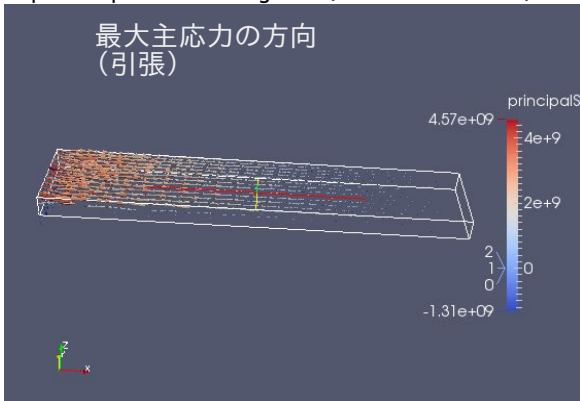
から計算しているの、これらの項目がFrontISTRから出力されている事が前提になる。これら項目が出力されていない場合は、3-7-2項の方法で出力項目を追加して、再計算し、データをvtk変換しておく必要がある。



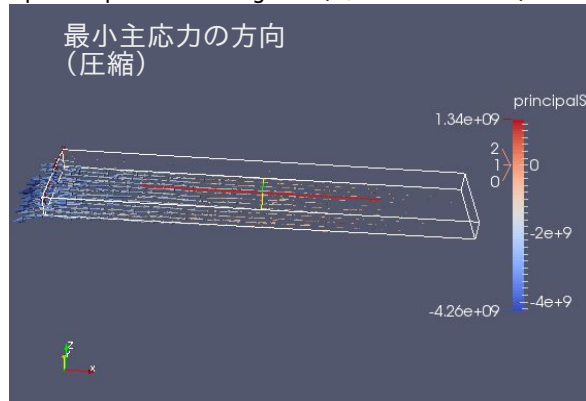
主応力（主ひずみ）については、その主値と各方向の成分が出力できる様になっている。この為、その値と方向を知る事ができる。今回のモデルで、主応力（圧縮と引張）の値とその方向を確認してみる。



principalVectorSigma1 (σ_1 の各方向成分)



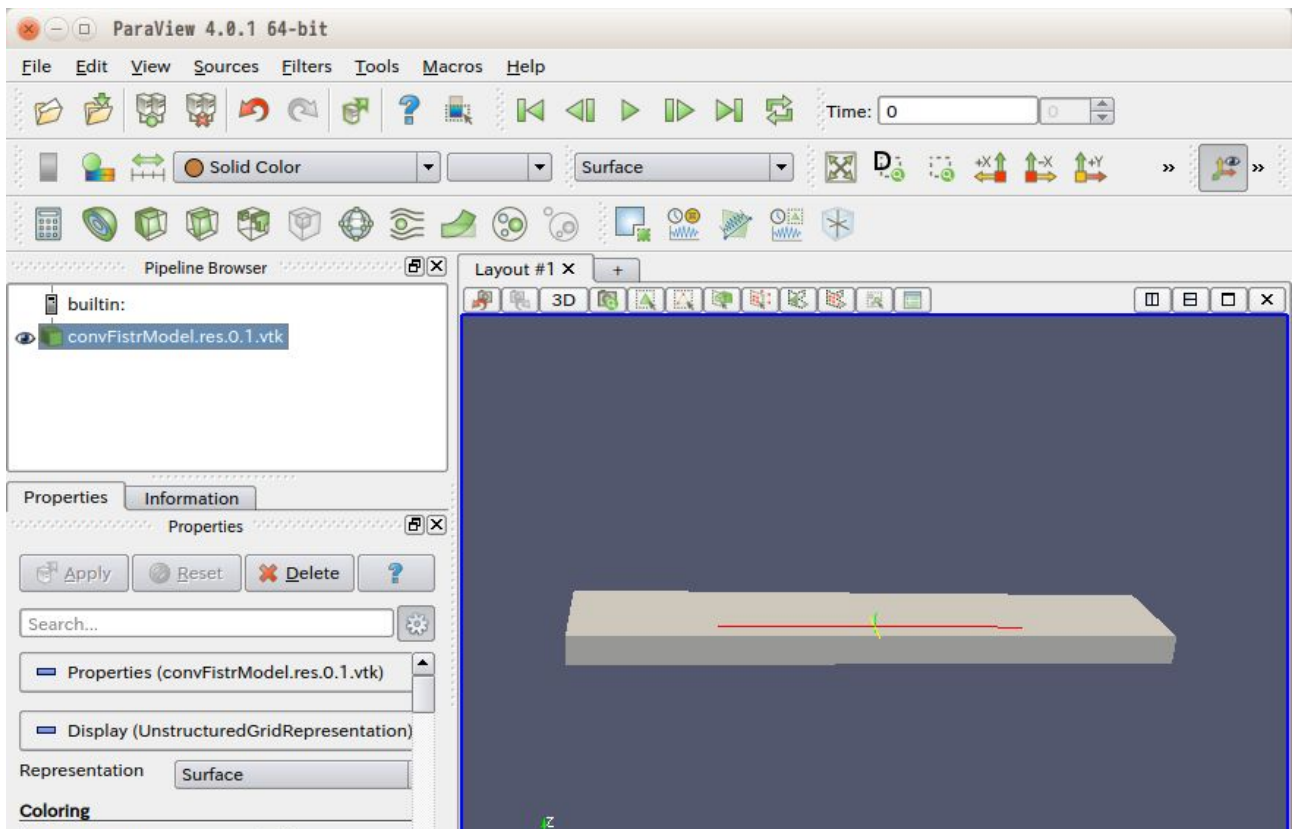
principalVectorSigma3 (σ_3 の各方向成分)



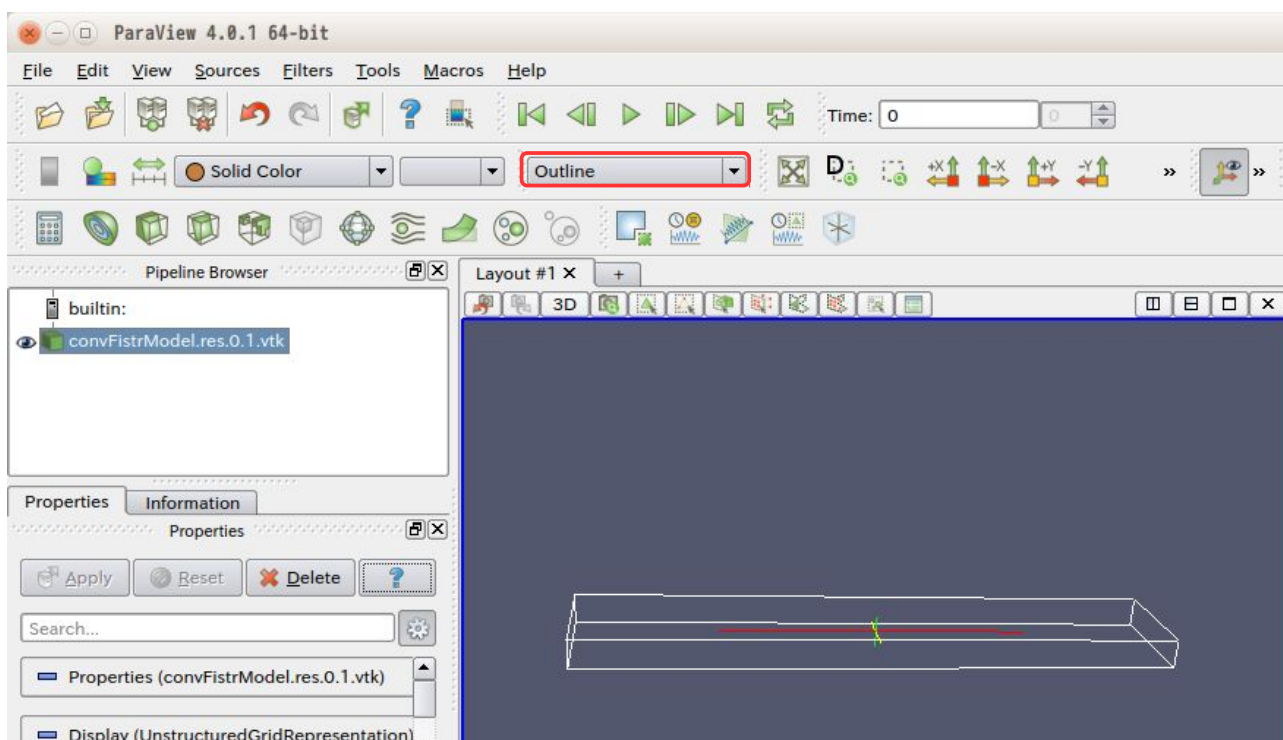
上記の様に、片持ち梁の上面に引張が梁の長手方向に働き、下面に圧縮が梁の長手方向に働いている事が確認できる。


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。

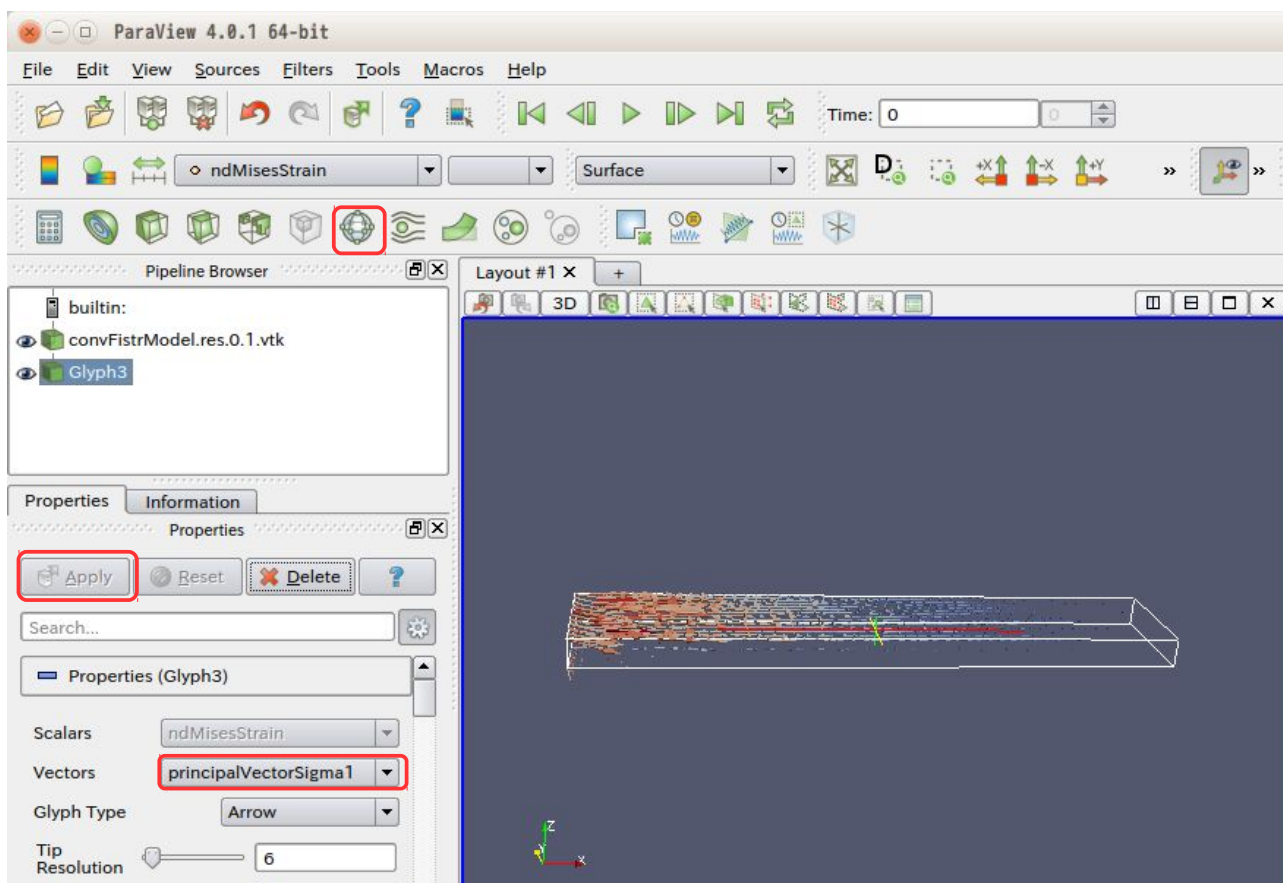
その方法は、まずデータ変換した vtk ファイル (convFistrModel.res.0.1.vtk) を読み込む。下図が読み込んだ状態。




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。

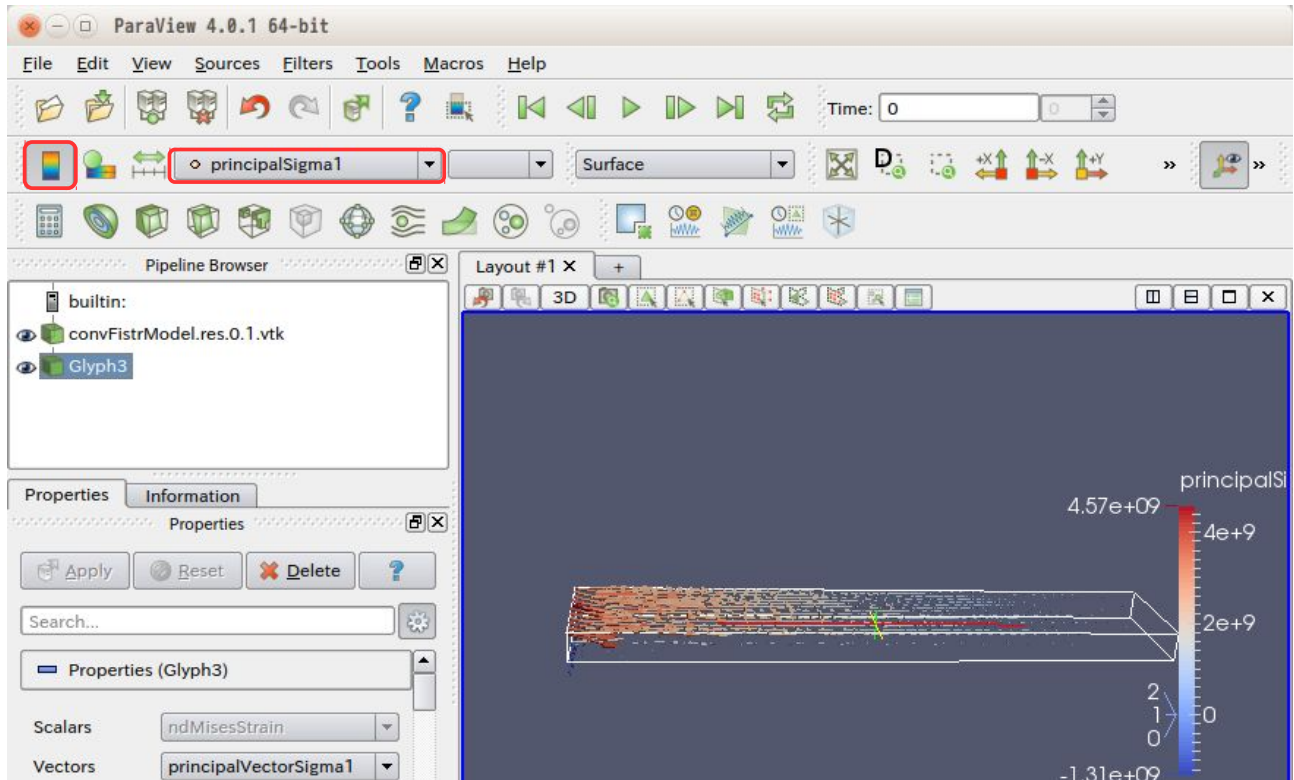


この後、ベクトルを表示させる為に、 ボタンをクリックし、vectors を「principalVectorSigma1」に設定し、「Apply」ボタンをクリックする。下図がこの状態になる。



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。

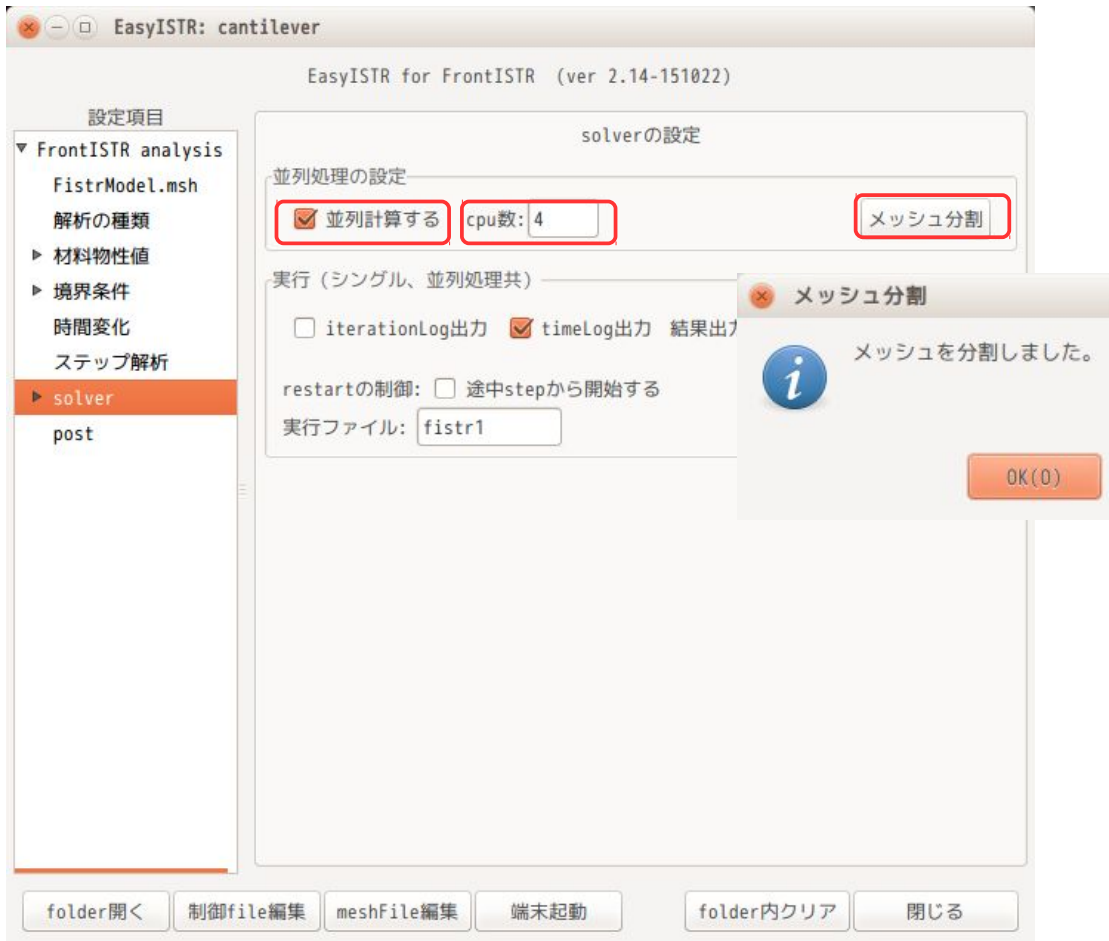
並列処理を行うために、チェックボックス「並列計算する」にチェックする。この時点で、並列処理するために必要な制御ファイル「hecmw_part_ctrl.dat」を確認し、workFolder 内に存在しない場合は、
「hecmw_part_ctrl.dat」のバックアップファイルがあれば、そのファイルを元に戻す
バックアップファイルが無ければ、デフォルトの「hecmw_part_ctrl.dat」をコピーする
その後、hecmw_part_ctrl.dat ファイルに記述されている cpu 数を読み込んでテキストボックスに表示する。

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。



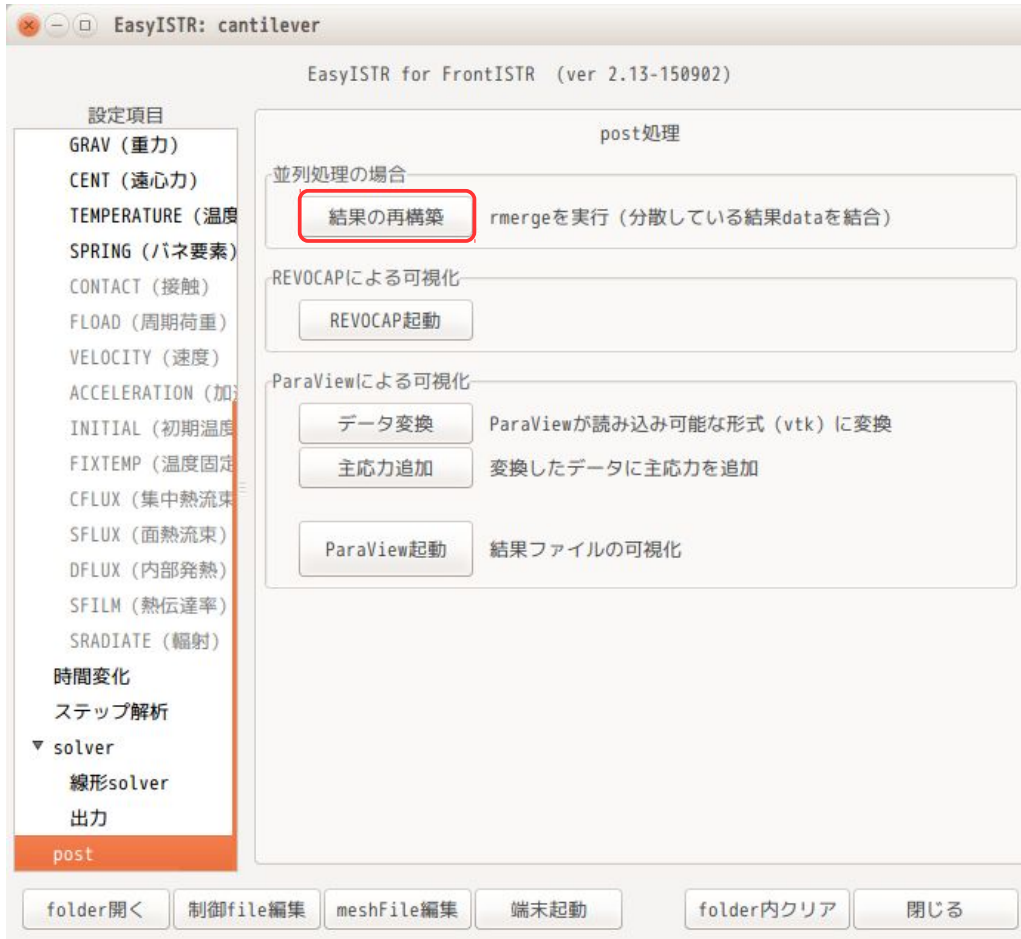
以上で、並列計算する為の設定が終了したので、「FrontISTR 実行 (並列)」ボタンをクリックする事で、並列処理が開始する。



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。

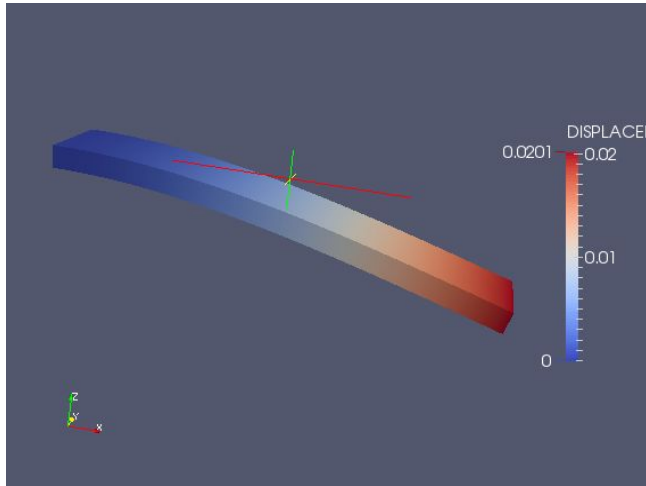
その方法は、下図の「結果の再構築」ボタンをクリックする事で、結果を再構築することができる。



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、「****.res.merge.***.vtk」形式のファイルを読み込む事になるが、paraView を起動すると、この形式のファイルを読み込んだ状態で起動するので、そのまま「apply」ボタンをクリックすれば、結果が表示される。

以下は、変位を表示させた結果になる。



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。

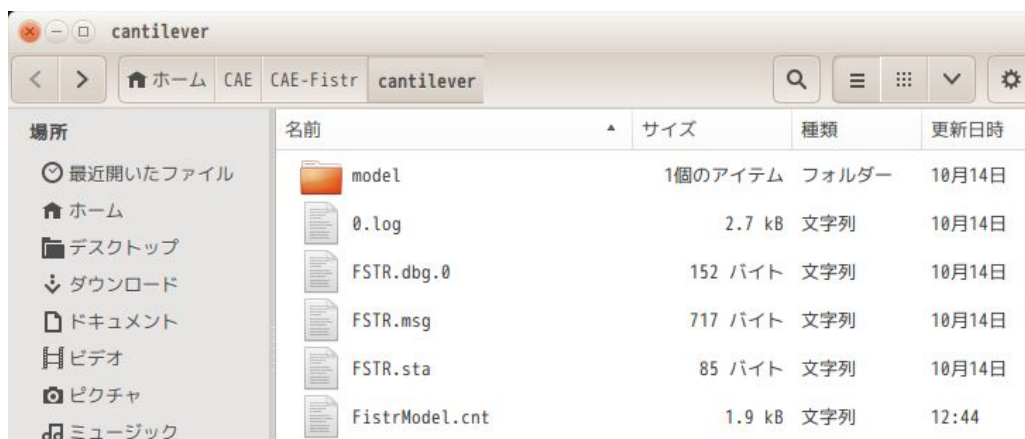


次項以降に、これらボタンの動作について説明する。

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。ただし、workFolder は、EasyISTR が直接編集している folder ではないので注意が必要。

EasyIstr は、workFolder 内の内容を tempFolder 内にコピーして、その tempFolder 内容が編集対象になっている。(実行は、workFolder 内で実行する。)



3-10-2. 「temp 開く」ボタンの動作

このボタンをクリックする事で、EasyISTRの編集対象の tempFolder (easyIstrUser/data/temp) を開く事ができる。

3-10-3. 「cnt,msh 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
FistrModel.msh	メッシュファイル
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)

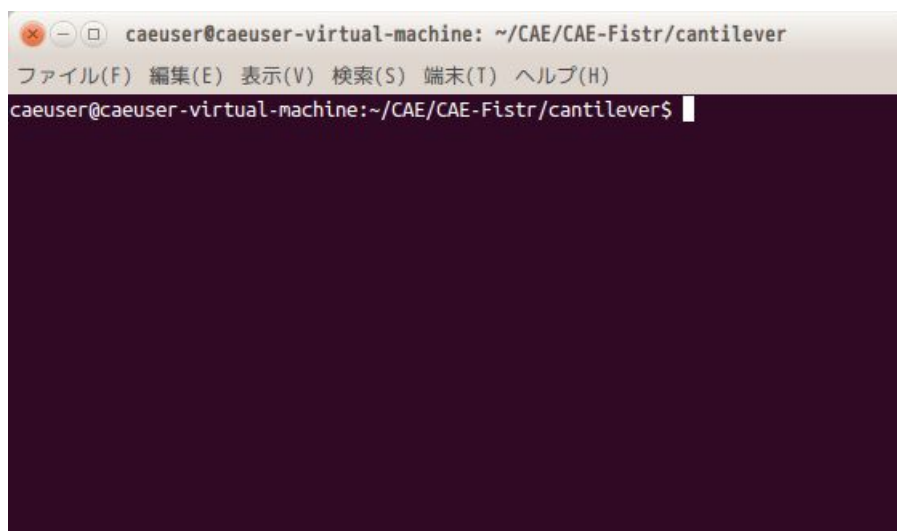
これらのファイルは、編集用の tempFolder (easyIstrUser/data/temp) 内のファイルを開いているので、実際の workFolder 内のファイルではない事に注意する。

また、制御ファイルとメッシュファイルについては、編集用の仮のファイルであることに注意する。ファイルサイズを抑える為に、制御ファイルについては、等分布荷重の設定が省略されており、メッシュファイルについては、keyword (!NODE, !ELEMENT, !EGROUP, !SGROUP, !NGROUP, !EQUATION) の行数が大幅に省略されてる。実際の制御ファイルとメッシュファイルは、実 folder (workFolder) 内にある。

大規模メッシュの場合は、メッシュファイルを editor で開くことができない程サイズが大きくなるが、この様な場合でも、メッシュファイルを開くことができるので、どのような keyword が定義されているかが容易に確認できる。

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。



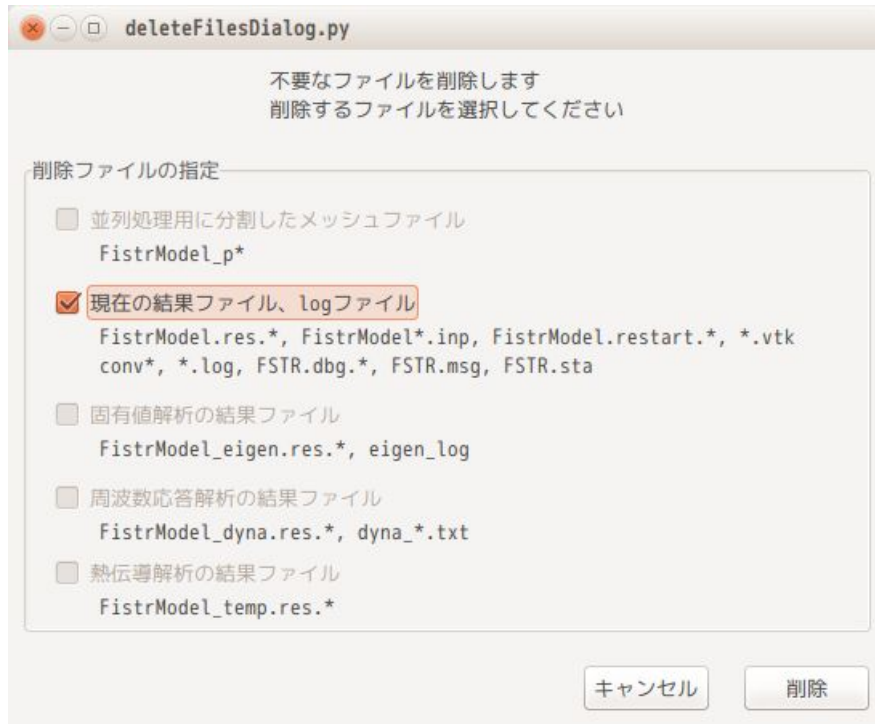
3-10-5. 「temp 設定保存」ボタンの動作

EasyISTR が編集しているのは、tempFolder 内のファイルを編集している。この folder 内の制御ファイルとメッシュファイルは、編集しやすい用にファイル内容が大幅に省略されているので、計算開始する場合は、編集した内容を翻訳して、実 folder (workFolder) に転送する必要がある。この tempFolder 内容を実 folder に翻訳転送する処理を行う。

EasyISTR 上で FrontISTR を実行する時（「FrontISTR 実行」ボタンをクリックした時）は、この操作を行った上で実行する様にしている。

3-10-6. 「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2次
四角形 (シェル)	1次
梁要素 (beam)	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

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 (abaqusinp2fistrcont.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
```

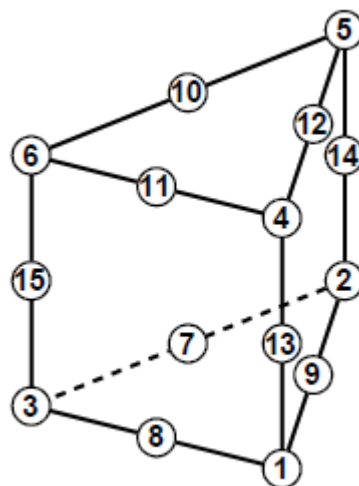
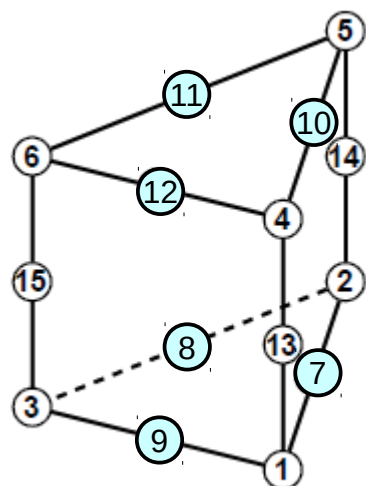
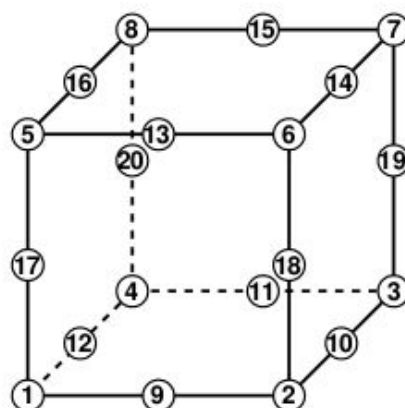
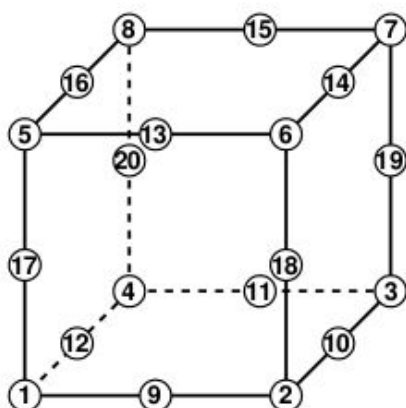
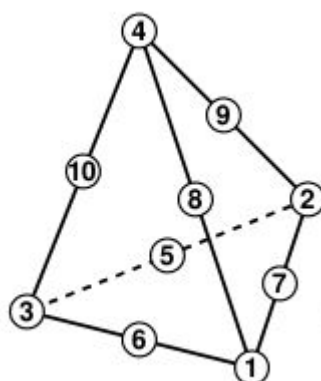
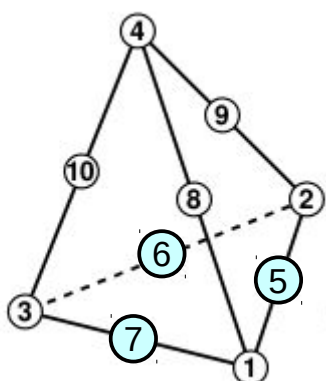
この変換は、現在のところ、前項で示したように、扱えるメッシュタイプは、以下に限られる。

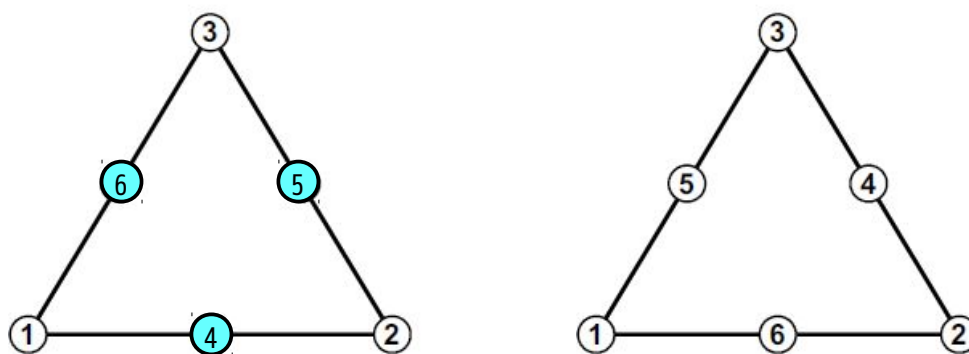
要素タイプ	abaqus	FrontISTR	備考
四面体 1次2次	C3D4、C3D10	341、342	solid
六面体 1次2次	C3D8、C3D8R、C3D20、C3D20R	361、362	↑
五面体 1次2次	C3D6、C3D15	351、352	↑
三角形 1次2次	STR135、S3、S6	731、761、732	shell
四角形 1次	S4R5	741、781	↑
beam 1次	B31	611、641	beam

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。inp ファイルの keyword は、大文字に変換後、解釈するので、小文字が混在しても構わない。

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>< inp ファイル : beam10p.inp > : *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>< msh ファイル > 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.vtk」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
```


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)
nodeNo を設定

1
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
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)
elementNo を設定

44
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
1
1
1
1
2

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。

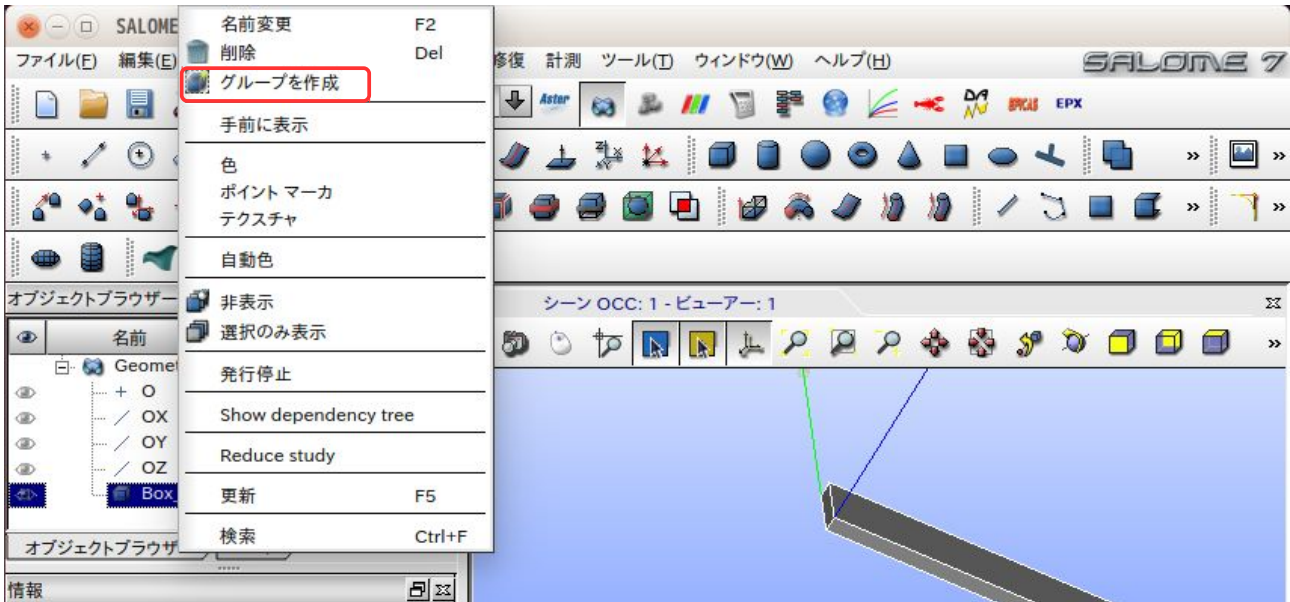


以下の画面が XYZ の寸法を入力した状態。この後、「適用して閉じる」ボタンをクリックして、直方体ができあがる。



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。

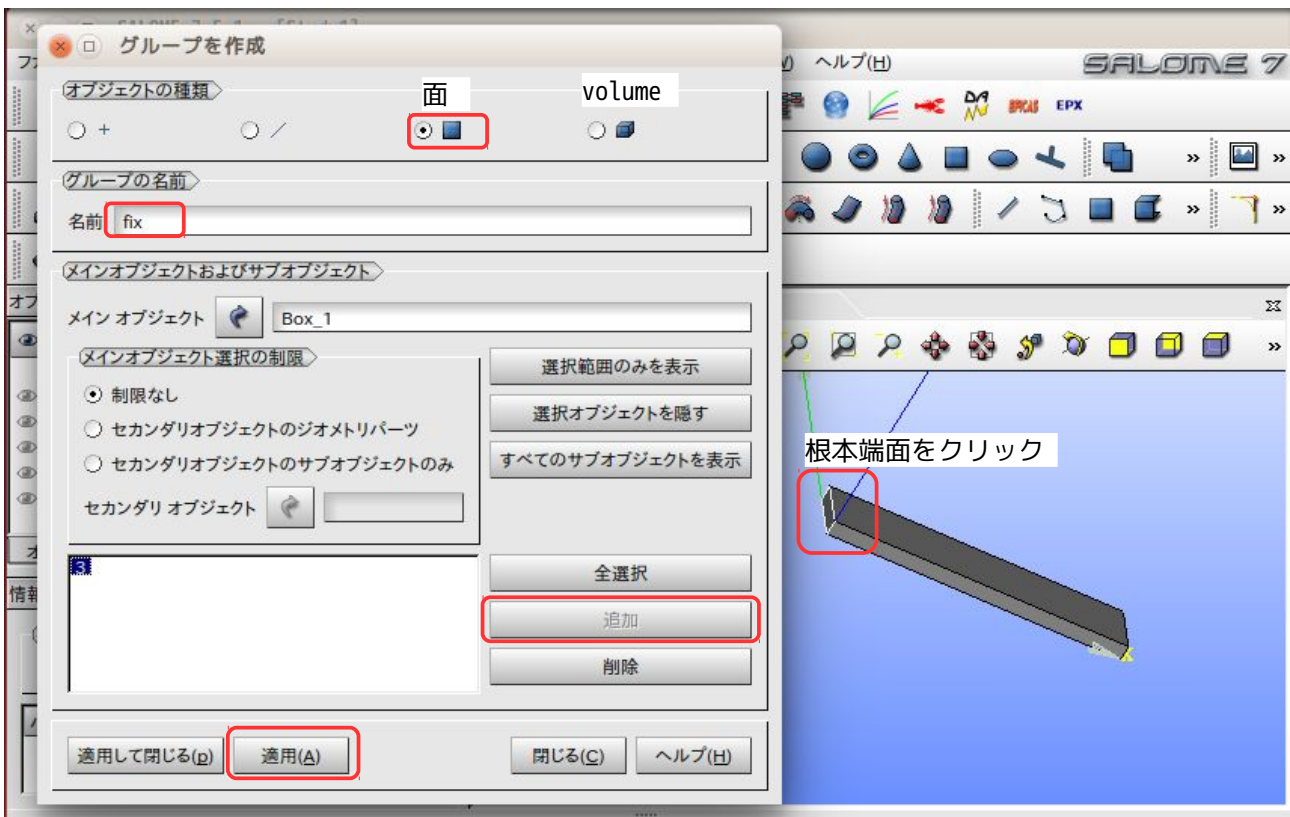
オブジェクトブラウザーの「Box_1」を選択して、右クリックして、メニューを表示させ、「グループを作成」を選択する。



「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、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。

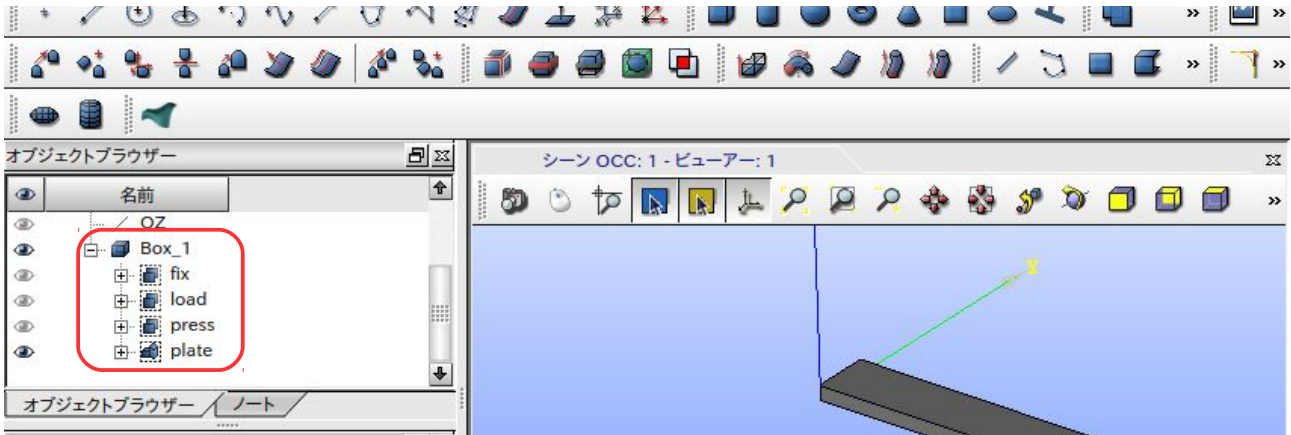


以上の操作で、Geometry内で以下のグループが作成できた事になる。

根本端面 fix 面グループ

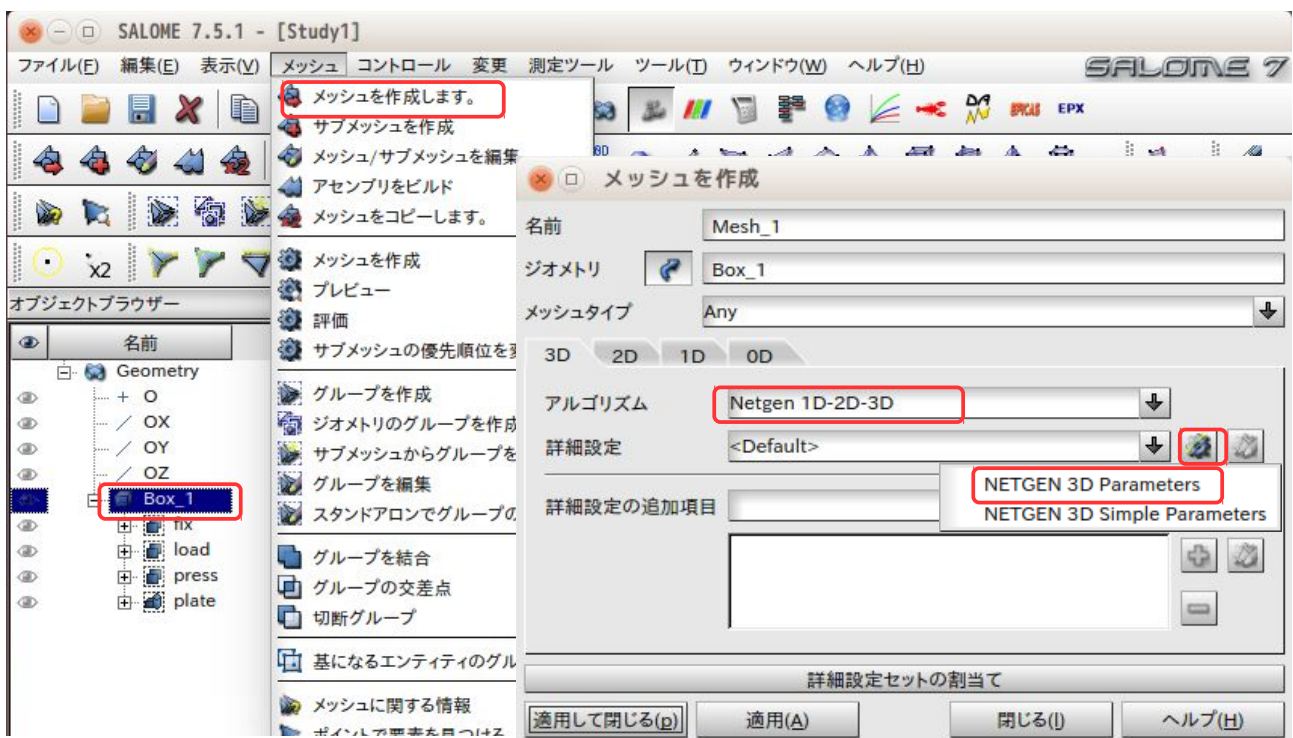
先端端面	load	面グループ
上面	press	面グループ
全体	plate	volume グループ

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。

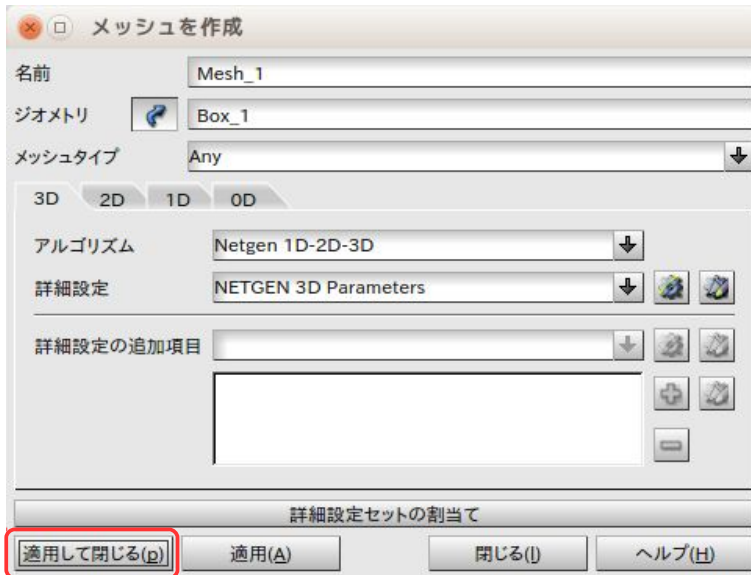


3-12-2. メッシュの作成

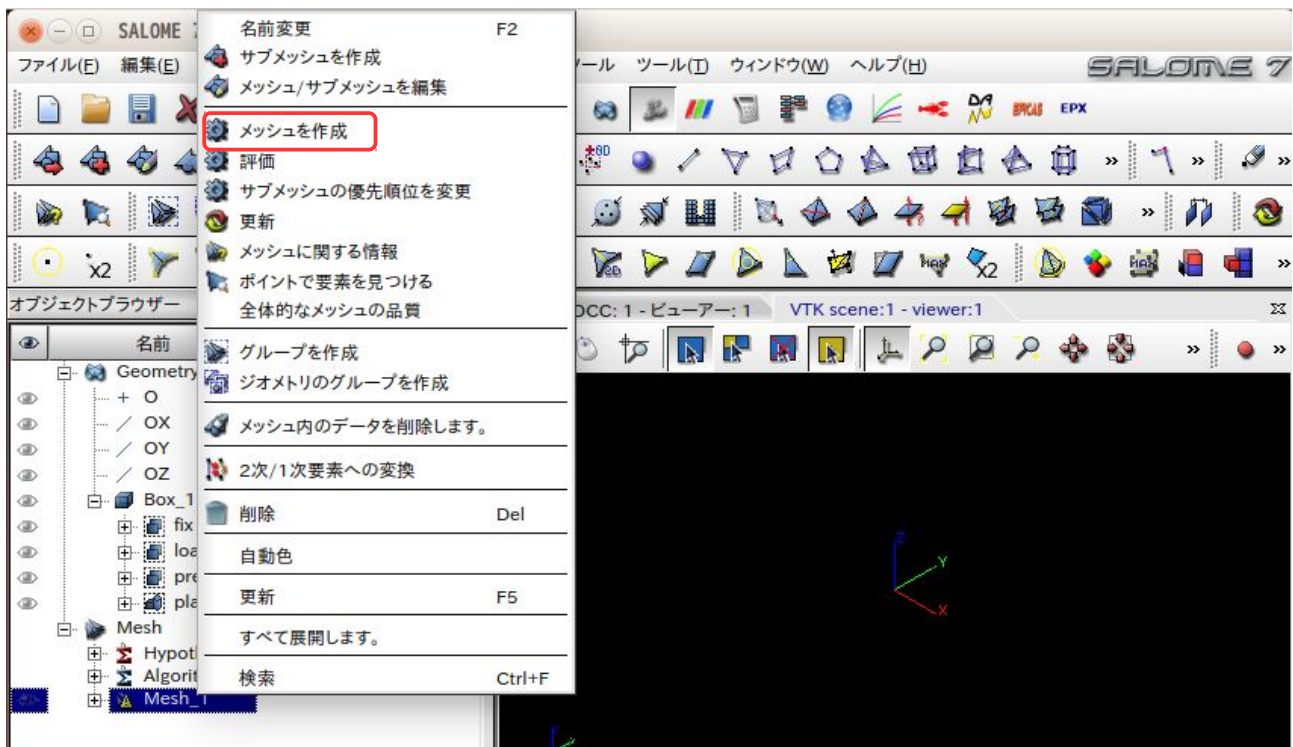
メッシュを作成する。まず、Mesh モジュールに変更する。オブジェクトブラウザー上の「Box_1」を選択した状態で、メニューバー「メッシュ」>「メッシュを作成します。」を選択する。この後、アルゴリズムを「Netgen 1D-2D-3D」に設定し、ギアのマークをクリックして「NETGEN 3D Parameters」を選択する。



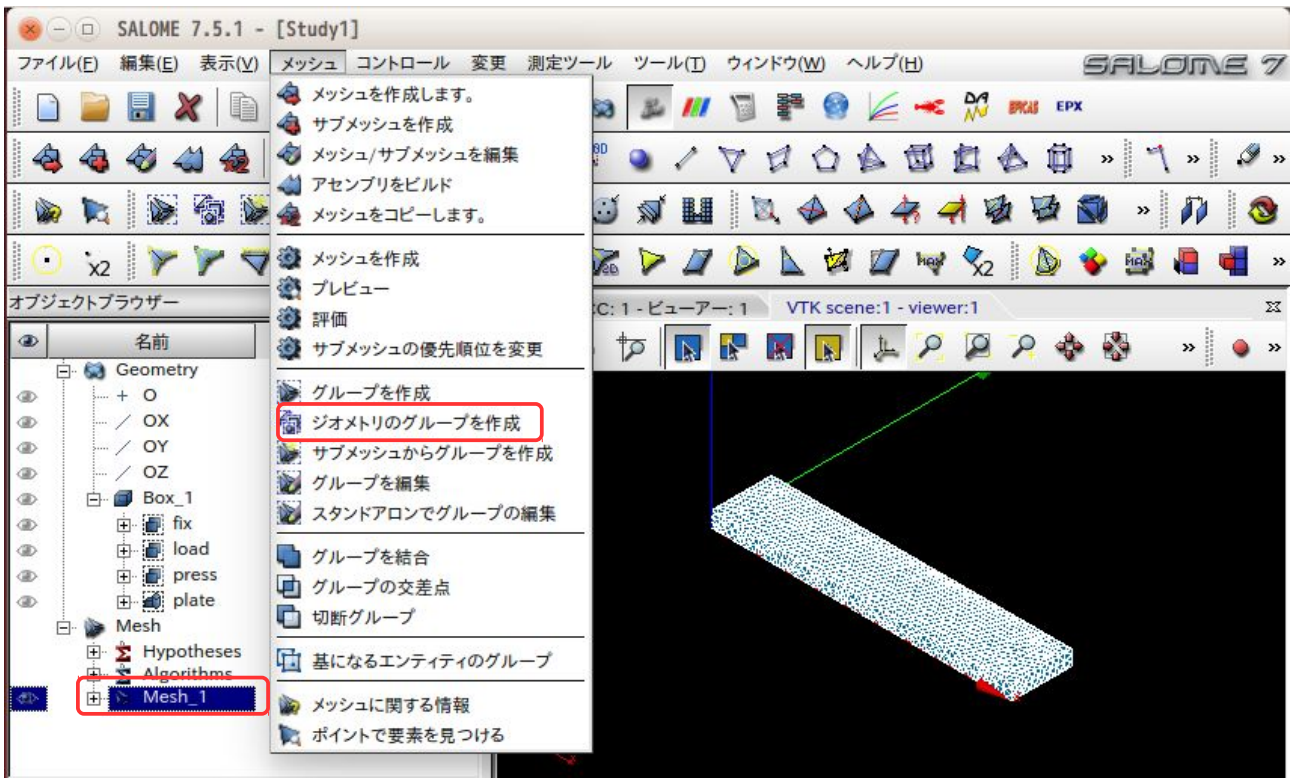
最大サイズを「2」、最小サイズ「1」に設定する。ここでメッシュ作成時のメッシュサイズを指定したことになる。設定後「OK」ボタンをクリックして閉じる。下図の画面に戻るので、「適用して閉じる」ボタンをクリックして、閉じる。



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。

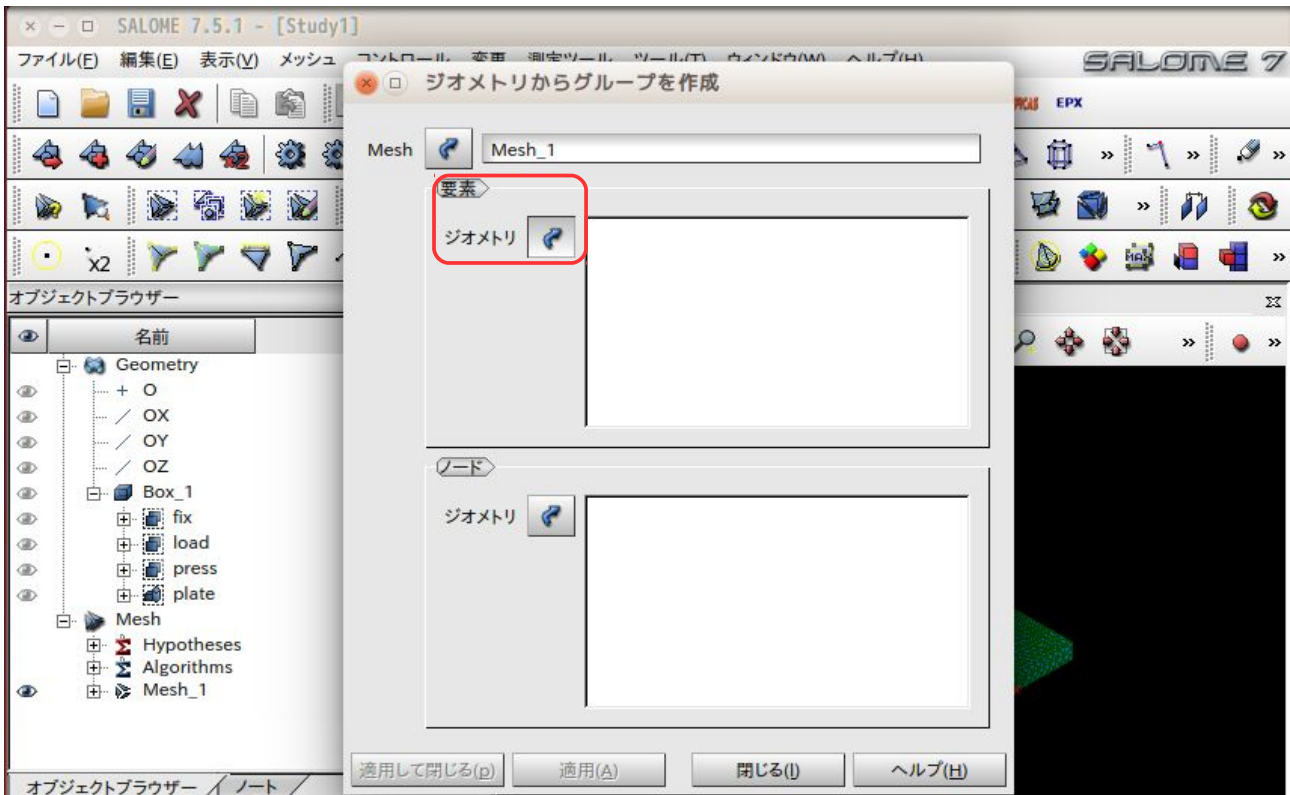


でき上がったメッシュに elementGroup、faceGroup、nodeGroup を作成する。「Mesh_1」を選択後、メニューバー「メッシュ」>「ジオメトリのグループ作成」を選択する。

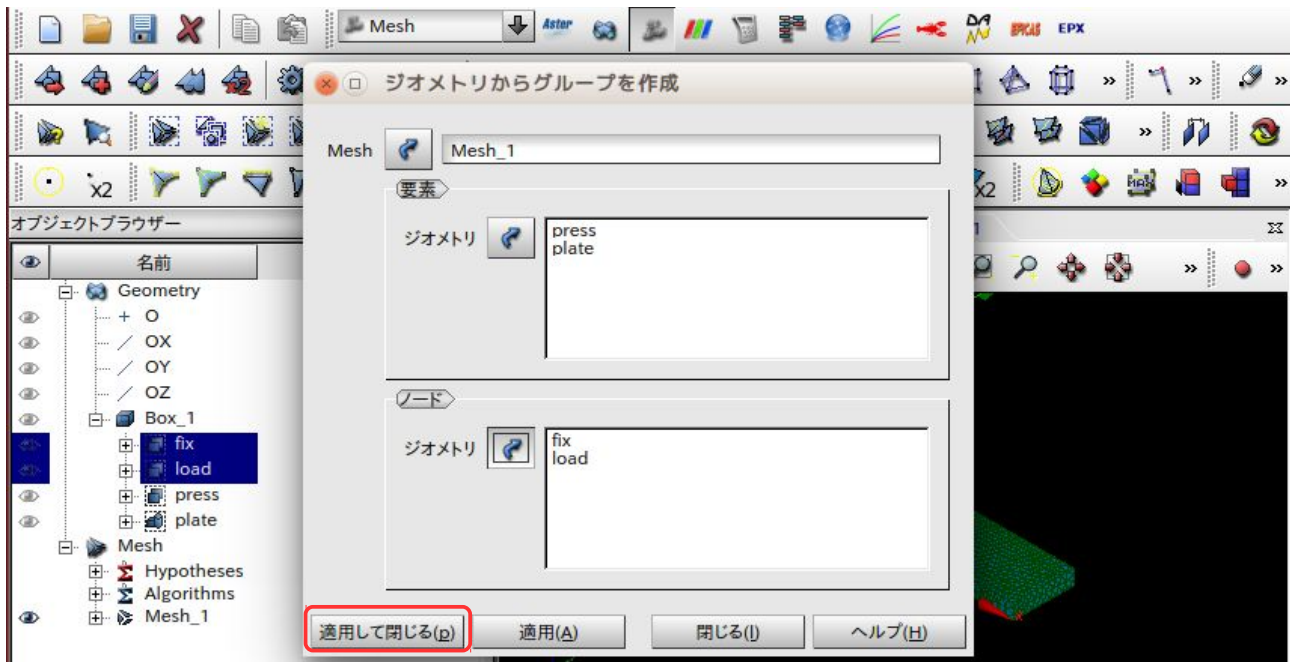


以下の画面上で各 Group を作成する。

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。

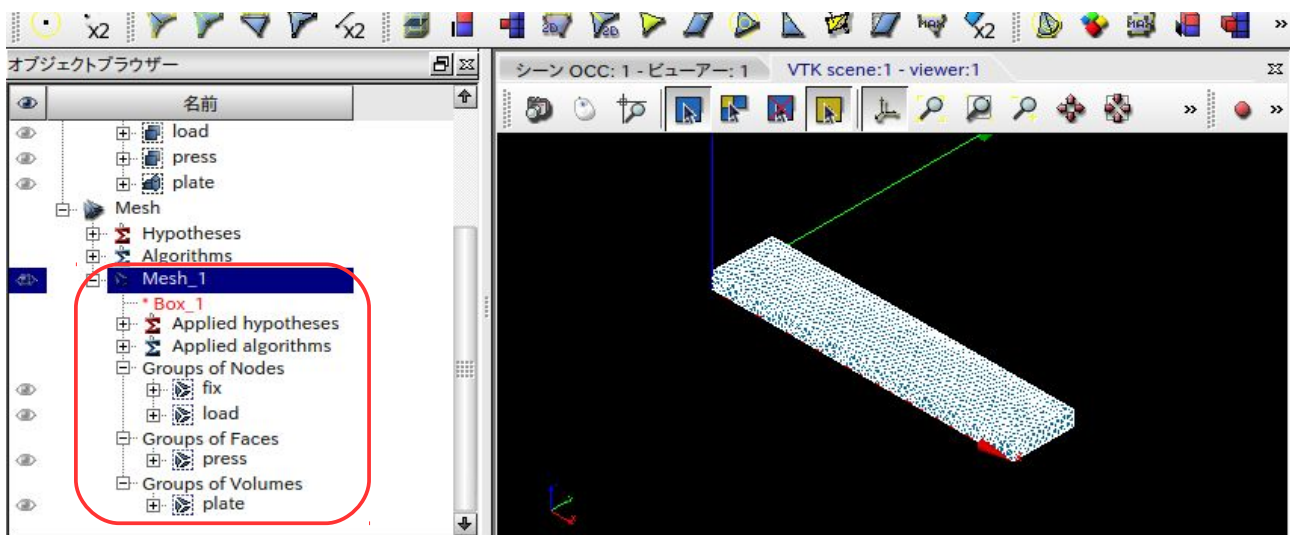


下図が選択した状態になる。この状態で、「適用して閉じる」ボタンをクリックして閉じる。



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
:
-----

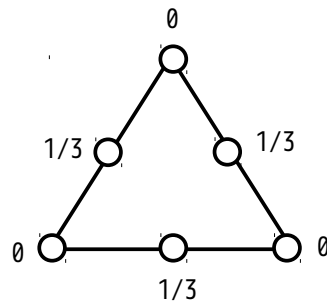
```

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。

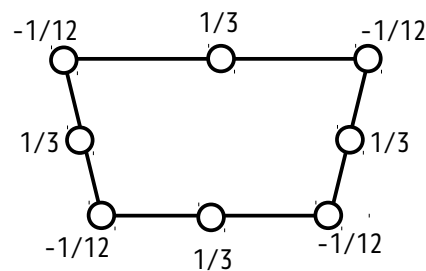
<三角形 2 次要素>

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。

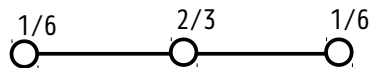
この線分の長さから、線分に印加される荷重が算出できるので、この荷重を節点に均等に配分する事で等分

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。

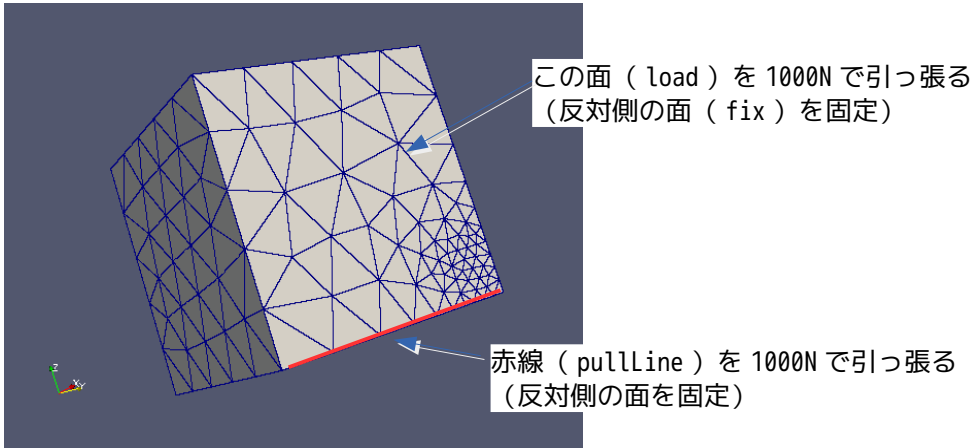
3-13-4. 等分布トータル荷重の設定結果の確認

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

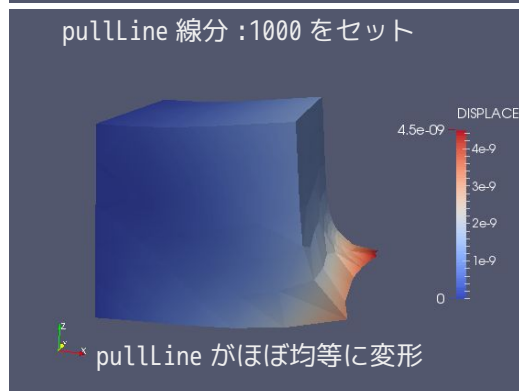
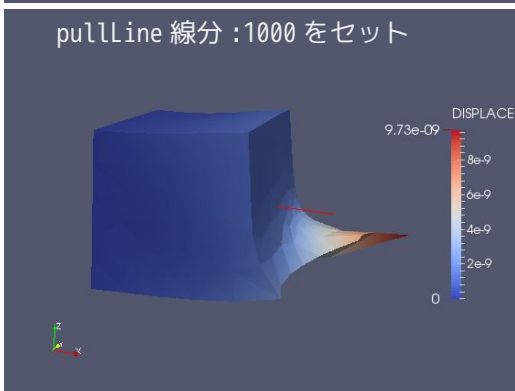
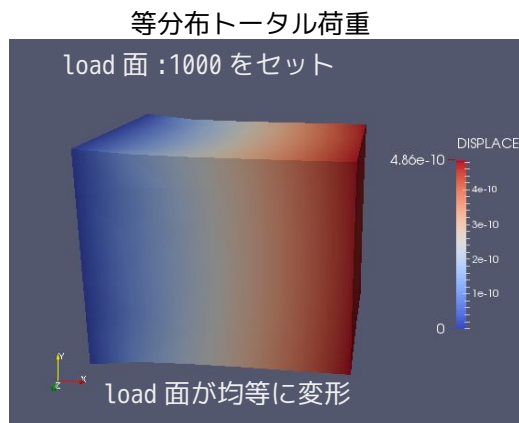
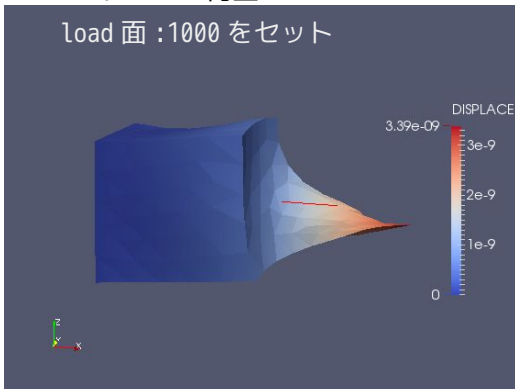


通常の設定では、要素サイズが異なっている部分がある為、細かい要素部分では、荷重密度が大きくなる。この為、変形形状が均一でなくなってしまう。これに対して、「等分布トータル荷重」の設定で引っ張った場合は、メッシュサイズの粗密に関係なくなる為、変形形状が均一になるはずなので、これを1次と2次要素で確認してみる。

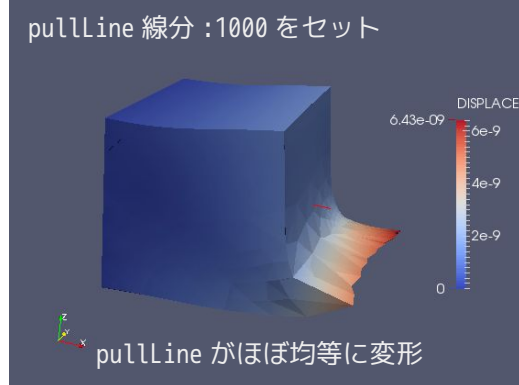
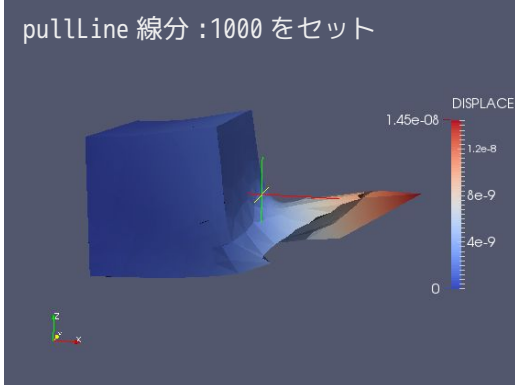
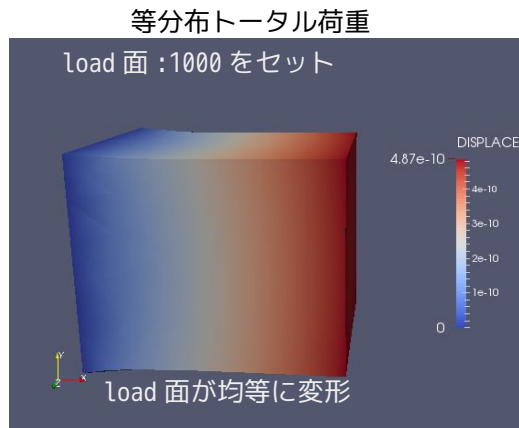
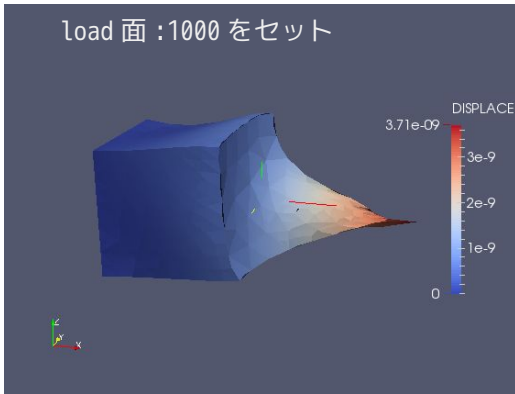
これを確認した結果、以下に示してある様に、「トータル荷重」で 1000N をセットすると、荷重分布がメッシュサイズに依存している為、細かいメッシュ部分 (荷重密度が大きい) は、荷重が大きくなるので、変形が大きくなってしまっている。

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、均等に变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



<四面体2次要素の場合>
トータル荷重



3-14. その他

3-14-1. plotStepMonitor について

非線形解析や動解析を行った場合、計算時間がかかる上、計算がうまく行われているかどうか、現在どこを計算しているのかが、計算途中では判断が難しい。計算が全て終了した後、計算結果を確認して、計算が正しく行われたかどうか、判断できる事になる。

この為、計算途中の各 step の状況を逐次グラフ表示させる様にして、計算途中においても、計算がうまく行われているかどうか判断できる様にした。このグラフ表示を gnuplot を使って実現したものが、plotStepMonitor になる。

plotStepMonitor は、現在のところ、非線形解析（接触解析、弾塑性解析）と動解析（過渡応答解析、非定常熱伝導解析）について対応している。

グラフ表示する項目は、以下の項目になる。

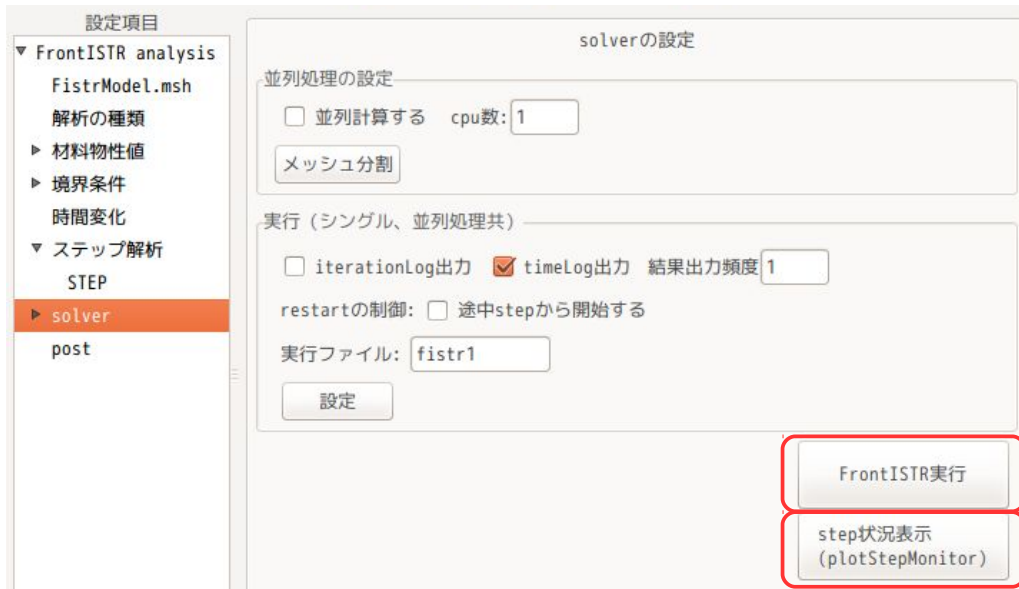
- X 方向の最大、最小変位 (Ux max、Ux min)
- Y 方向の最大、最小変位 (Uy max、Uy min)
- Z 方向の最大、最小変位 (Uz max、Uz min)
- 最大 mises 応力
- 残渣（非線形解析のみ）

これらの値は、計算途中で FrontISTR が書き込んでいる以下のファイルからデータを読み取っている。

0.log 各方向の最大、最小変位、最大 mises 応力
 FSTR.sta イタレーション回数と残渣

plotStepMonitor は、5s 間隔でこれらファイルの内容を読みにいき、値を取得して、gnuplot を使って逐次グラフ表示していく。

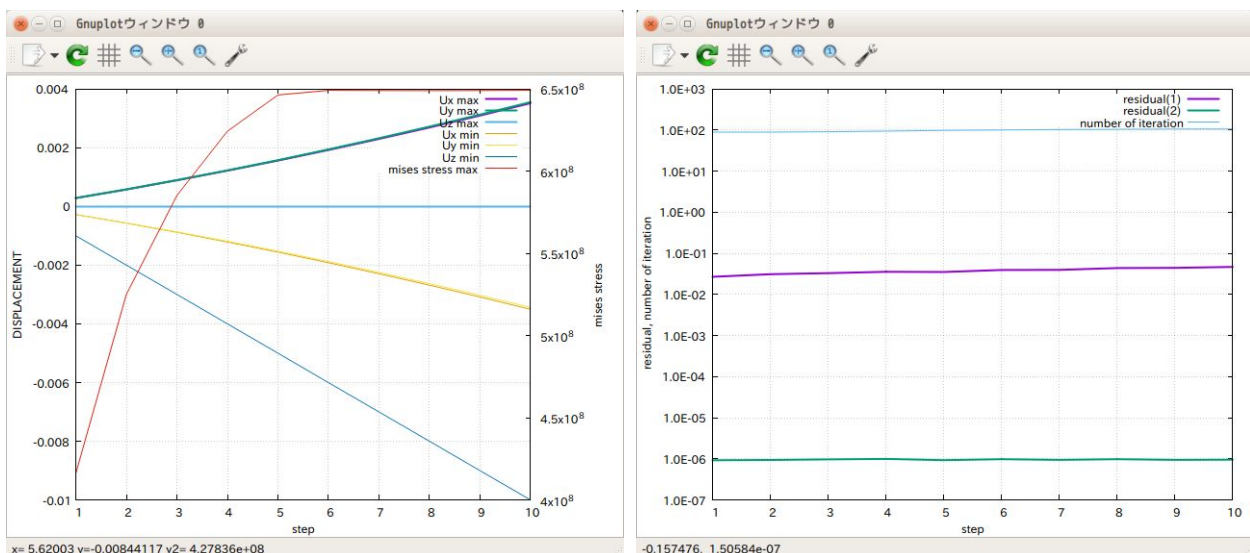
使い方は、下図の様に「FrontISTR 実行」ボタンをクリックして計算開始させた後、または計算が終了した後に「step 状況表示」ボタンをクリックする事で、plotStepMonitor が起動して、各 step の状況を表示する。



また、plotStepMonitor の起動は、端末から以下を入力しても起動できる。この場合は、カレントディレクトリにある「0.log」、「FSTR.sta」を読み込みグラフ表示する。

```
$ plotStepMonitor.py
```

plotStepMonitor の出力例として、4-2 項の弾塑性解析の例を以下に示す。



変位や、残渣、イタレーションの回数がグラフ表示され、逐次更新しているため、計算途中で解が収束していない場合、直ぐに判断できる事になる。

3-14-2. 複数 step 解析について

非線形解析や動解析において、計算途中で境界条件を変更して計算させる場合、計算を複数の step に分けて、計算させる事ができる。

複数 step 解析の方法は、設定項目 Tree 内の「ステップ解析」を選択し、「追加」ボタンを step 解析の回数分クリックして、「設定」ボタンで確定させる。下図の例では、3 step 分を追加した例になる。



これにより、設定項目 Tree 内に「境界条件 STEP1」と「境界条件 STEP2」が追加されるので、この中に各々の step 内で使用する境界条件を入力する。

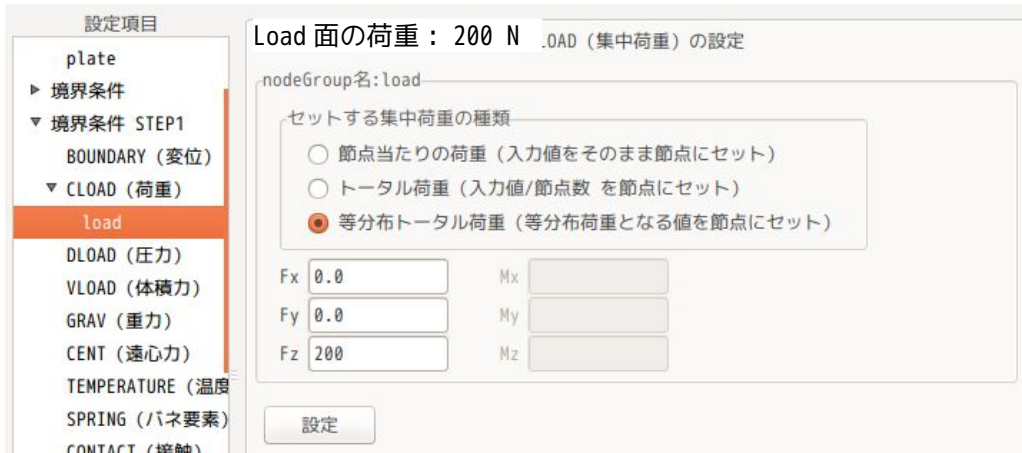
境界条件	#最初の step で使用する境界条件を設定
境界条件 STEP1	#2 番目の step で使用する境界条件を設定
境界条件 STEP2	#3 番目の step で使用する境界条件を設定

fix 面を固定したまま、load 面の荷重を順番に 100 → 200 → -100 N の様に変化させていく例を以下に示す。fix 面の固定は、全 step で共通の為、最初の step で定義しておけば、全ステップで使用できる。

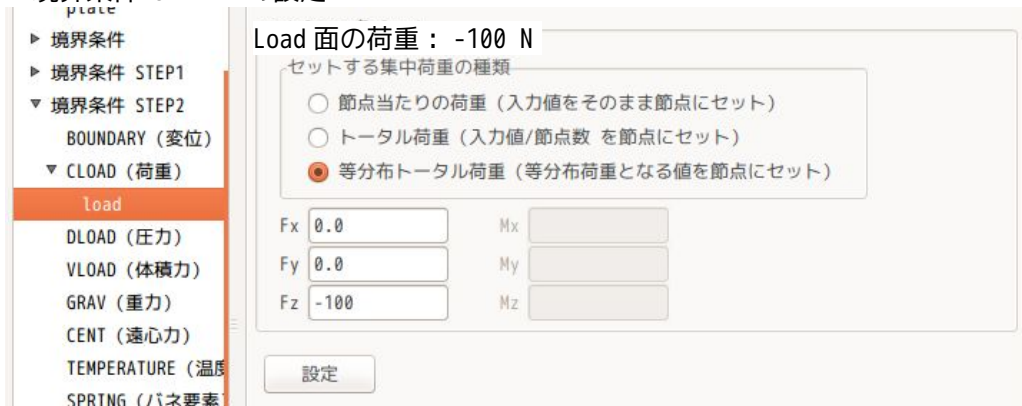
<境界条件 (最初の step) >



<境界条件 STEP1 の設定>



<境界条件 STEP2> の設定



境界条件を設定後、再び、「ステップ解析」内の各 step の設定とその step で使用する境界条件を選択する。

<STEP0>

FrontISTR analysis
FistrModel.msh
解析の種類
▼ 材料物性値
plate
▶ 境界条件
▶ 境界条件 STEP1
▶ 境界条件 STEP2
▼ ステップ解析
STEP0
STEP1
STEP2
時間変化
▶ solver
post

STEP0
TYPE: STATIC
CONVERG: 1e-6
SUBSTEPS: 5
MAXITER: 1000
DTIME, ETIME: [] []

step解析する境界条件
現在の境界条件
設定する境界条件
BOUNDARY, fix, STEP0
CLOAD, load, STEP0
選択>>
<<戻す
設定

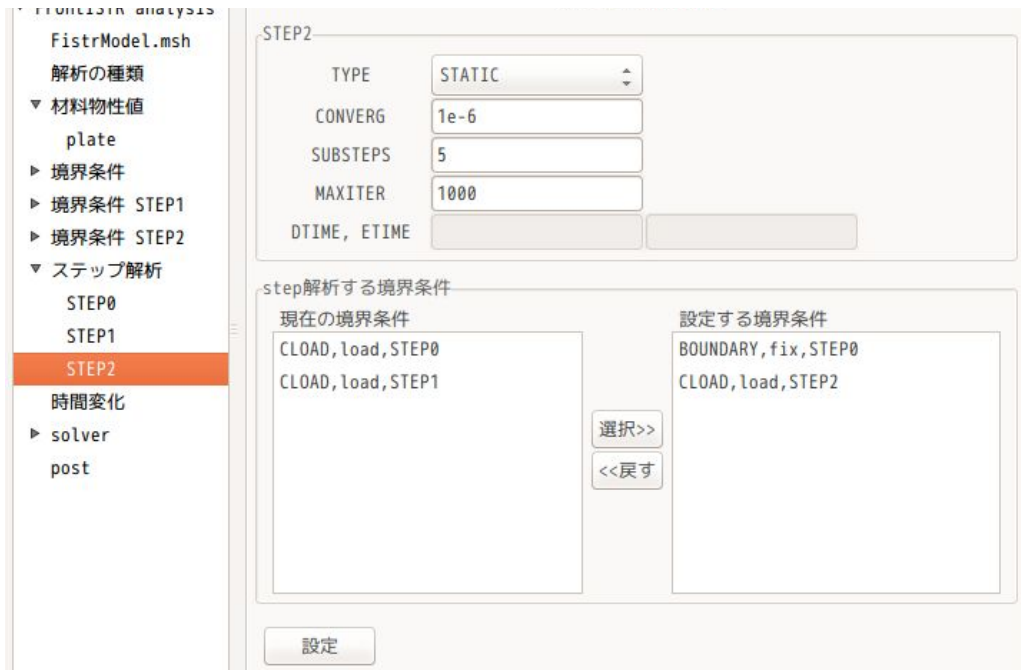
<STEP1>

FistrModel.msh
解析の種類
▼ 材料物性値
plate
▶ 境界条件
▶ 境界条件 STEP1
▶ 境界条件 STEP2
▼ ステップ解析
STEP0
STEP1
STEP2
時間変化
▶ solver
post

STEP1
TYPE: STATIC
CONVERG: 1e-6
SUBSTEPS: 5
MAXITER: 1000
DTIME, ETIME: [] []

step解析する境界条件
現在の境界条件
設定する境界条件
CLOAD, load, STEP0
BOUNDARY, fix, STEP0
CLOAD, load, STEP1
選択>>
<<戻す
設定

<STEP2>



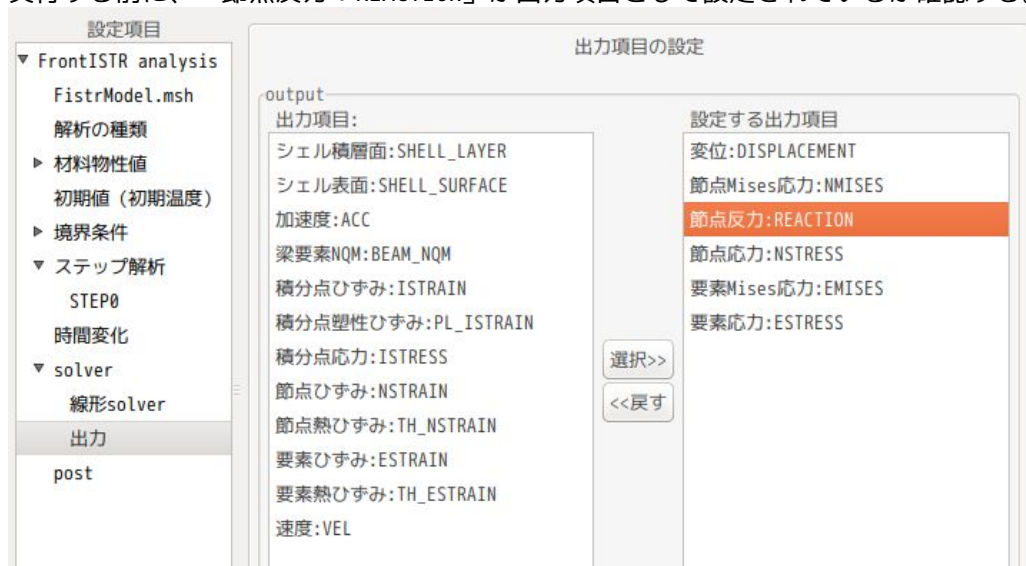
以上の設定で複数 step の解析が可能になり、計算を実行すれば、各 setp 毎に順番に計算が進んでいく。

3-14-3. 反力の算出について

変位拘束を設定している節点には、反力が生じている。節点グループを指定して、そのグループの反力（合力）を算出して、出力させる事ができる。これを実現するためのスクリプトを作成している。この使用方法は、以下による。

1) FrontISTR 実行

実行する前に、「節点反力：REACTION」が出力項目として設定されているか確認する。



また、解析の種類は「非線形解析」で実行する。FrontISTRのバージョンによって、線形弾性静解析では、反力が正しく計算されないため。線形弾性静解析で計算できるモデル、条件でも非線形として計算させる。（FrontISTR ver5以降は、線形解析でも正しく反力が計算でき

る。)

2) 端末起動

EasyISTR 画面下部の「端末起動」ボタンで端末を起動する。

3) コマンド入力 (反力算出)

端末上でコマンドを以下のように入力する。

この例は、節点グループ「fix」に発生する反力 (合力) を算出する例になる。

EasyISTR ver3.0以降から「getReactionForce.py」には、pathが通っていないので実行は、環境変数「\$binApp」を付加して実行する。

```
$ python3 $binApp/getReactionForce.py fix
FistrModel.res.0.1: [-8.938911832956364e-10, 5.184475071473571e-11,
300.000000000005195]
```

計算結果は、反力の合力がベクトルとして出力されており、この内容から、Z方向に反力が「300N」発生している事が確認できる。

以上の様な方法で、節点グループを指定して、反力を求めることができる。

3-14-4. 大規模メッシュ作成 (既存メッシュの細分化)

100万要素以上のメッシュを作成する場合、Salome等のメッシュャを使って、メッシュを作成した場合、メッシュ作成に多大な時間 (15分以上) が掛かってしまう。

この為、十数万程度の粗いメッシュから、メッシュを細かく作成し直す事で、大規模メッシュを短時間で作成できる。

メッシュを細かくする細分化の方法は、現在の要素に中間節点を追加し、この節点を使って新しい要素を作り出す方法を取っている。この為、この操作を1回行くと、要素数が8倍に増える。さらに行うと、64倍に要素数が増えていく。この方法では、100万要素のメッシュ作成時間は、1分以内で作成できる。

3-14-4-1. メッシュ細分化方法

メッシュ細分化のために、「fineSeparateMesh.py」スクリプトを作成しているため、これを実行するだけで実現できる。実行方法は、以下による。

EasyISTR ver3.0以降から「fineSeparateMesh.py」にpathが通っていないので、実行に当たっては、環境変数「\$binApp」を付加して実行する。

```
$ python3 $binApp/fineSeparateMesh.py FistrModel.msh 1
```

上記例では、メッシュファイル「FistrModel.msh」を numlevel「1」で細分化する。

numlevel:1 は、中間節点を1回作成して細分化する。numlevel:2 の場合は、2回中間節点を追加するので、要素数は64倍に増える。

このスクリプトは、四面体、六面体、五面体の1次2次要素に対応している。シェルやビーム要素には、対応していない。

3-14-4-2. メッシュ細分化の例

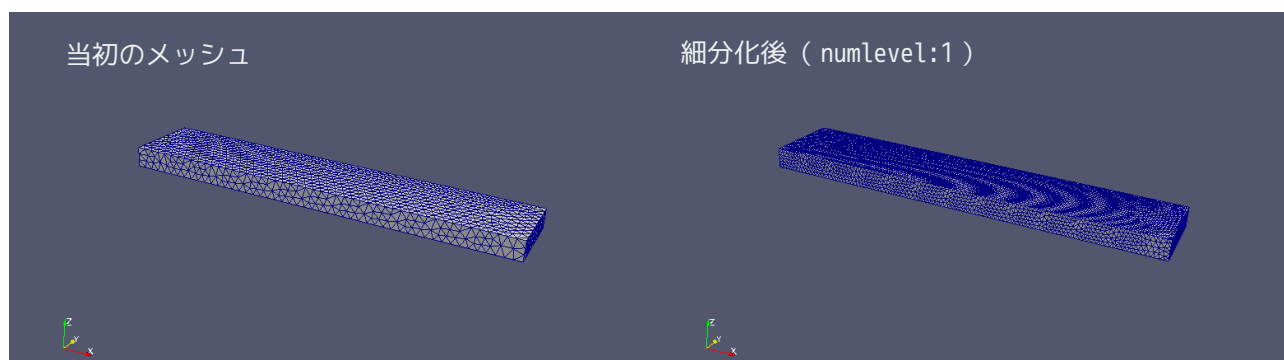
メッシュファイル「plate.unv」を使って、メッシュ細分化を行ってみる。

EasyISTR 上で、plate.unv ファイルをメッシュ変換して、FrontISTR 用のメッシュファイル「FistrModel.msh」を作成する。

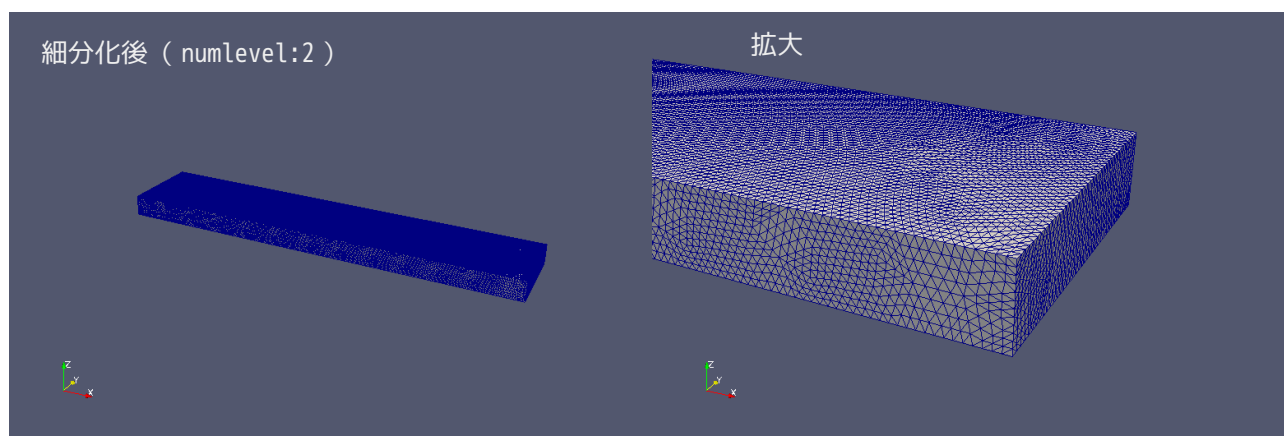
この後、EasyISTR 上の「端末起動」ボタンをクリックして、EasyISTR の端末を起動し、以下を入力して1回細分化する。

```
$ python3 $binApp/fineSeparateMesh.py FistrModel.msh 1
```

この結果が以下になる。1回の細分化で、要素数が8倍に増え、メッシュが細かくなっている。



numlevel:2 で細分化を行うと以下になる。



以上のように、既に出来上がったメッシュを細かくしたい場合には、この方法を使うことで、容易に細かくできるし、大規模メッシュが短時間で作成できる。

4. 応用事例

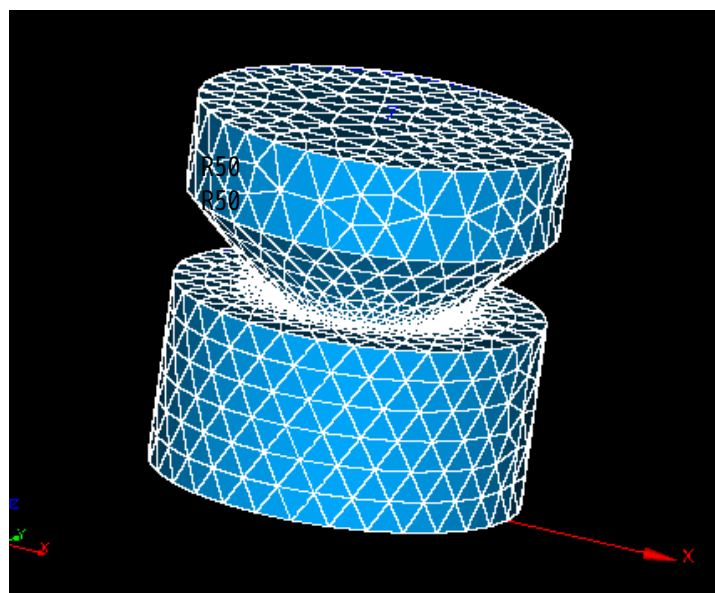
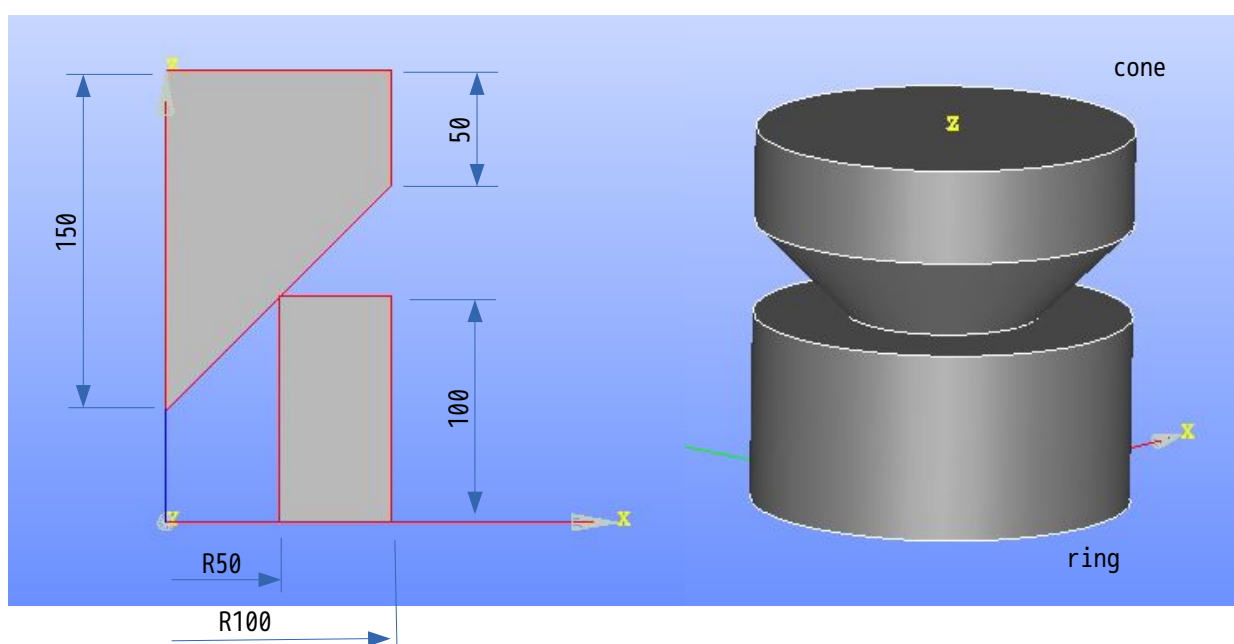
前項で弾性静解析の事例を取り扱った。ここでは、それ以外の事例について、その操作方法を説明する。

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。

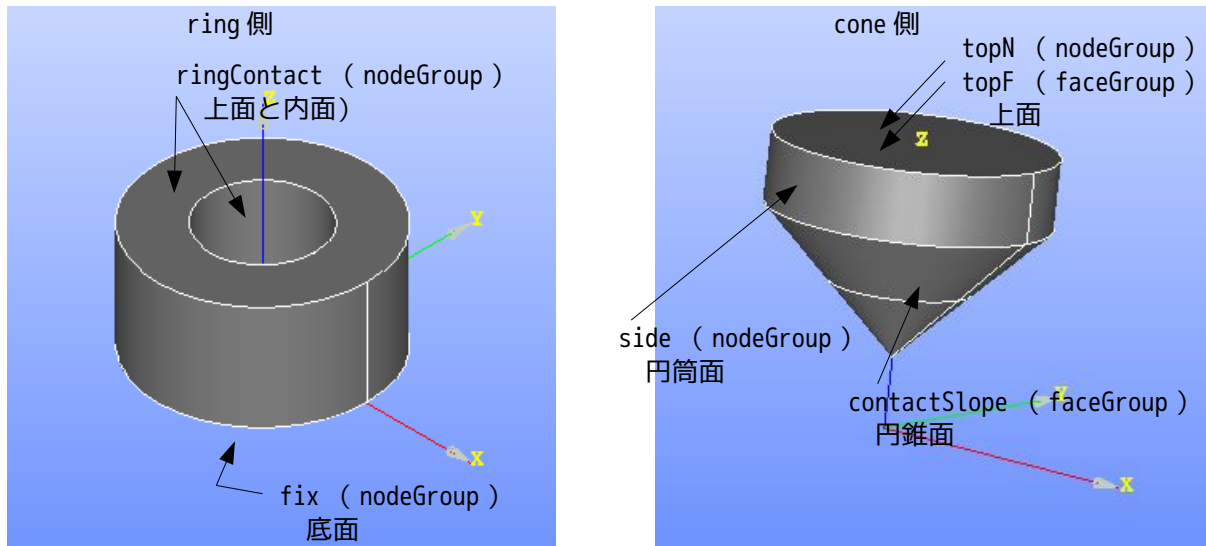
4-1-1. モデル形状

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

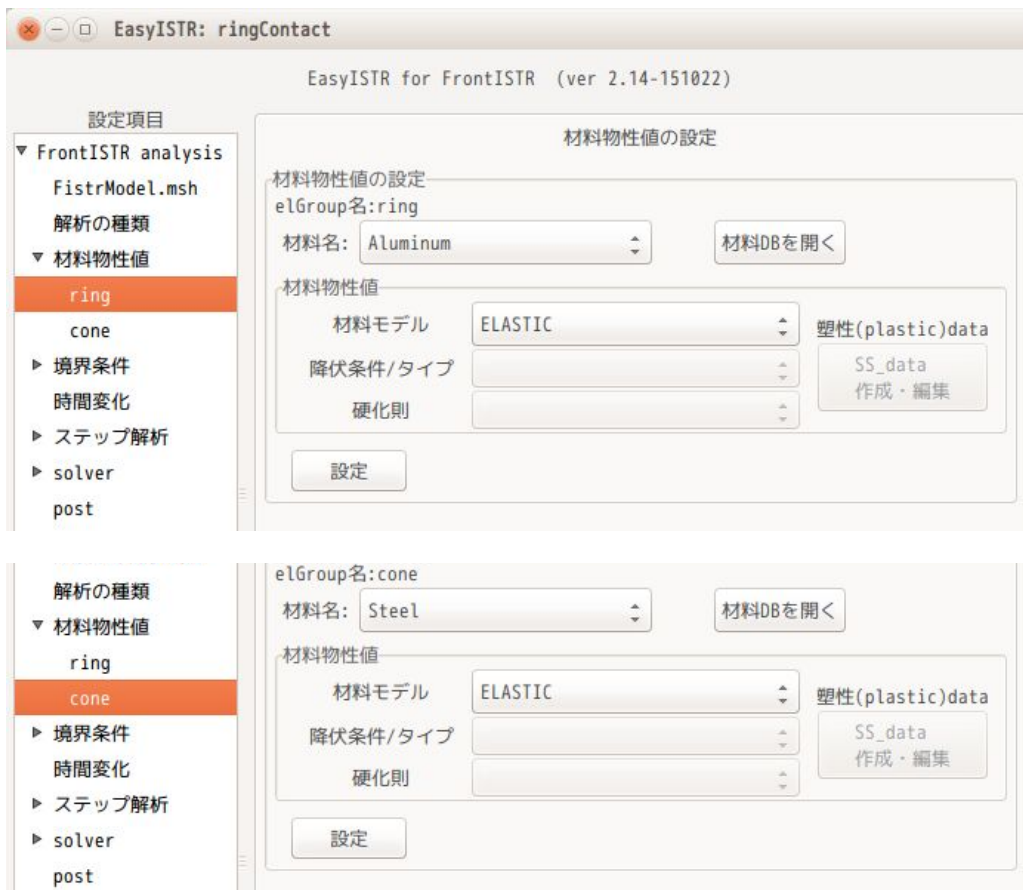
4-1-2-2. 解析の種類

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。



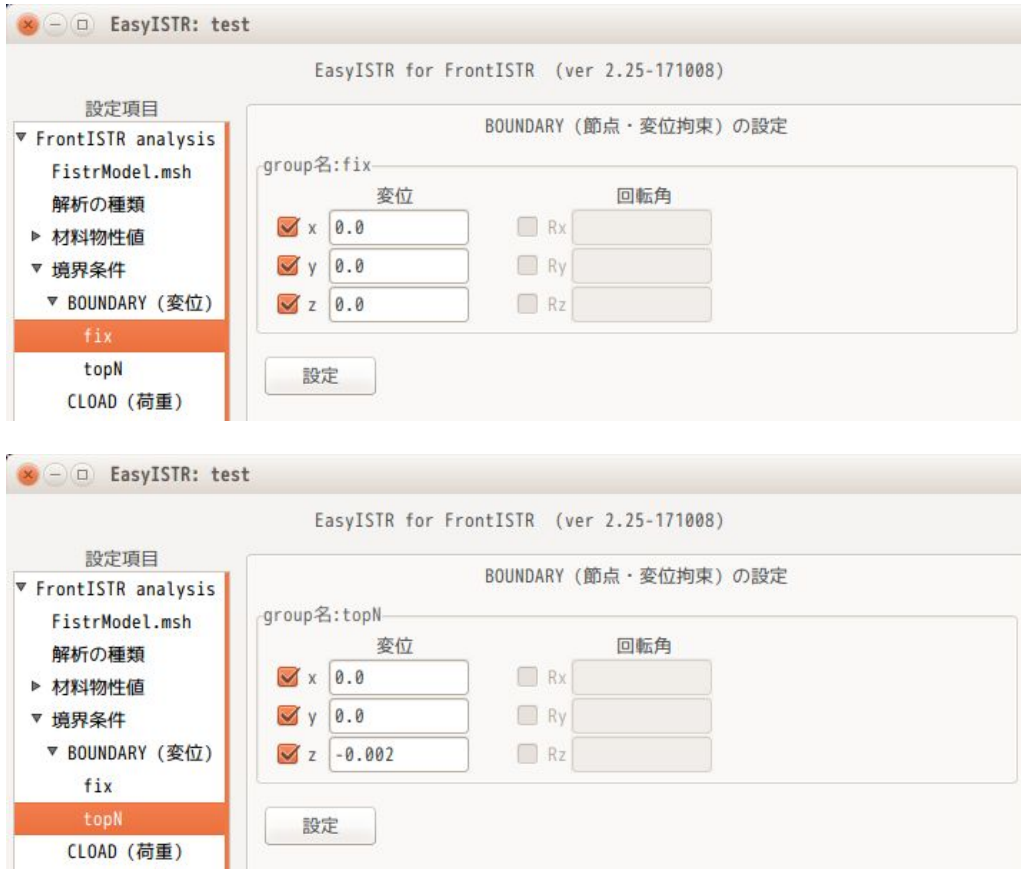
4-1-2-3. 材料物性値の設定

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。



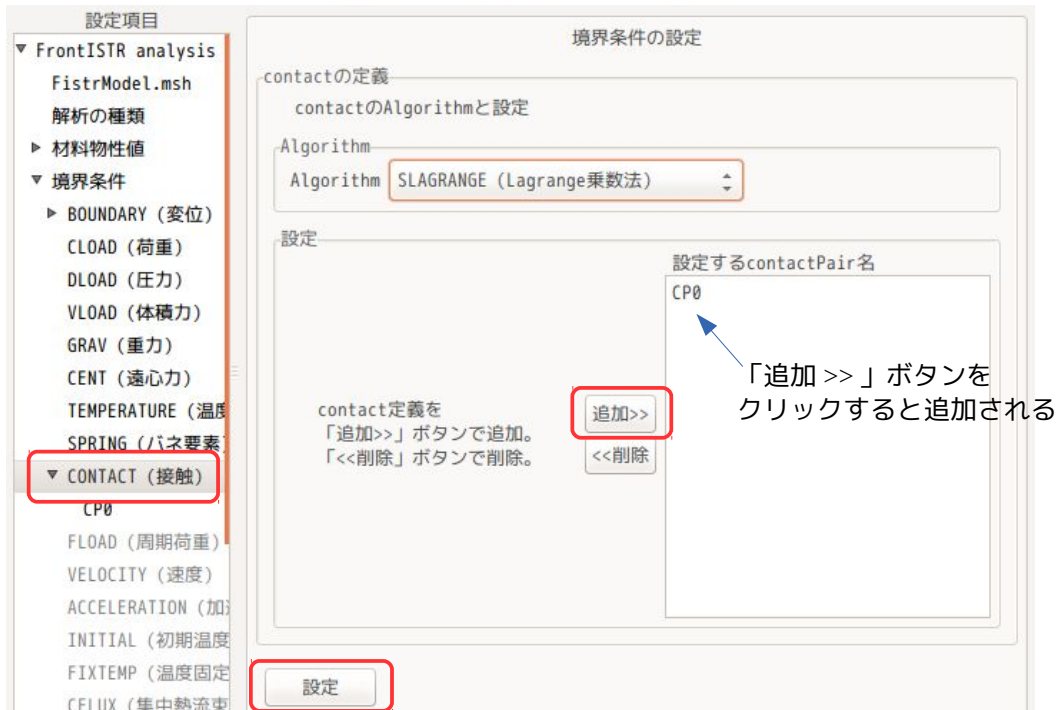
4-1-2-5. 接触の設定

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、「Lagrange 乗数法」が設定されている。
Algorithmは、「SLAGRANGE (Lagrange 乗数法)」と「ALAGRANGE (拡張 Lagrange 法)」が選択できるが、ここでは、「Lagrange 乗数法」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。

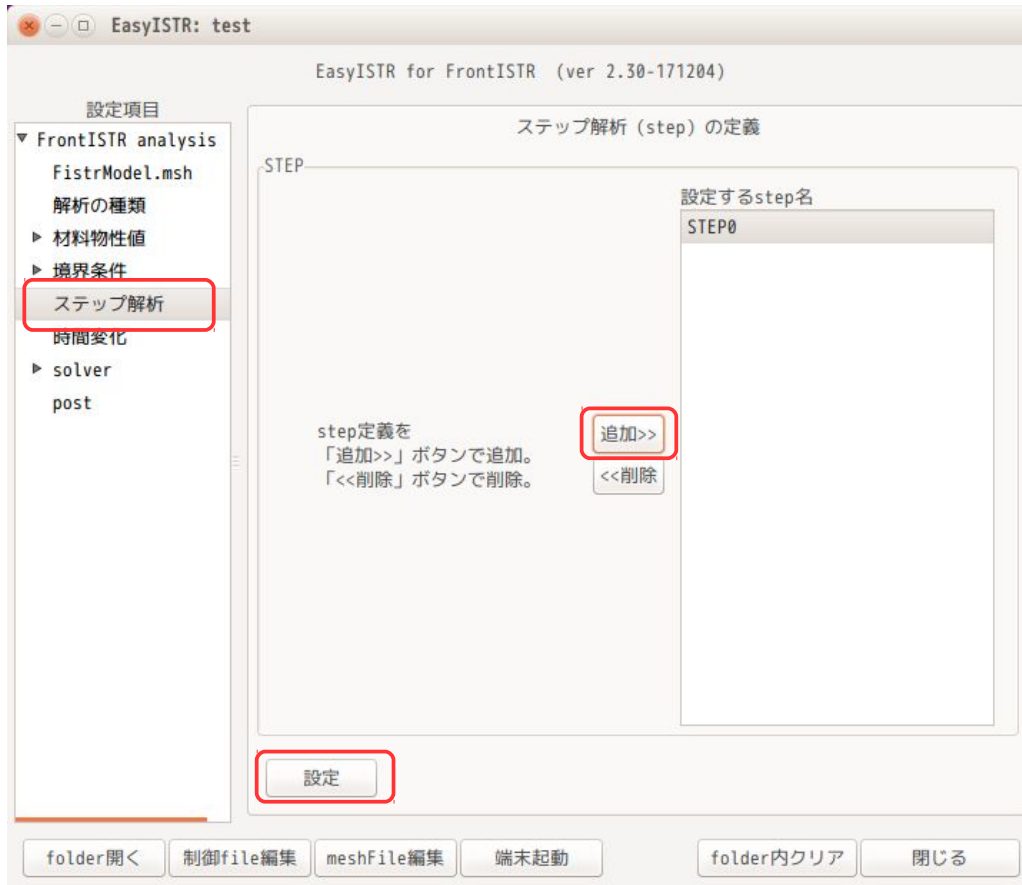


4-1-2-6. ステップ解析の設定

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「STEP0」を選択する。下図が「STEP0」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

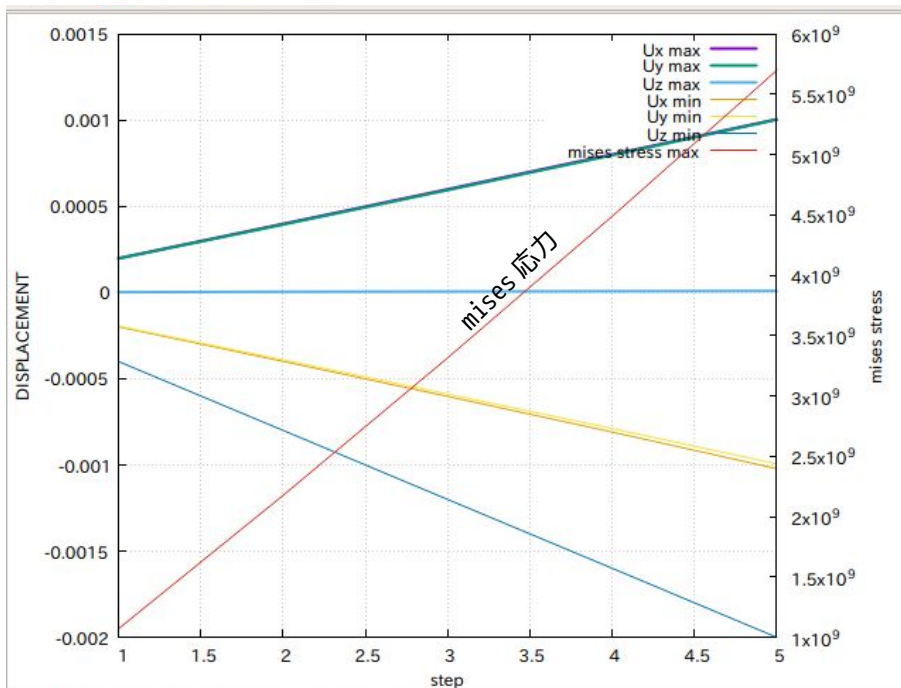
以上で設定が全て終了したので、計算を開始する。

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。



計算開始後、「step 状況表示」ボタンをクリックして、plotStepMonitor を起動すると、以下のグラフが表示され、各ステップの計算状況が逐次グラフ表示される。ここに、最大のmises 応力が表示されており、最終stepで約 5.7×10^9 Paと確認できる。今回の場合、計算時間が短いので、計算が終了した後、グラフが表示されるが、step 数が多く、計算時間がかかる場合は、計算の経過と共に、グラフ表示が更新され、逐次各stepの結果が確認できる。

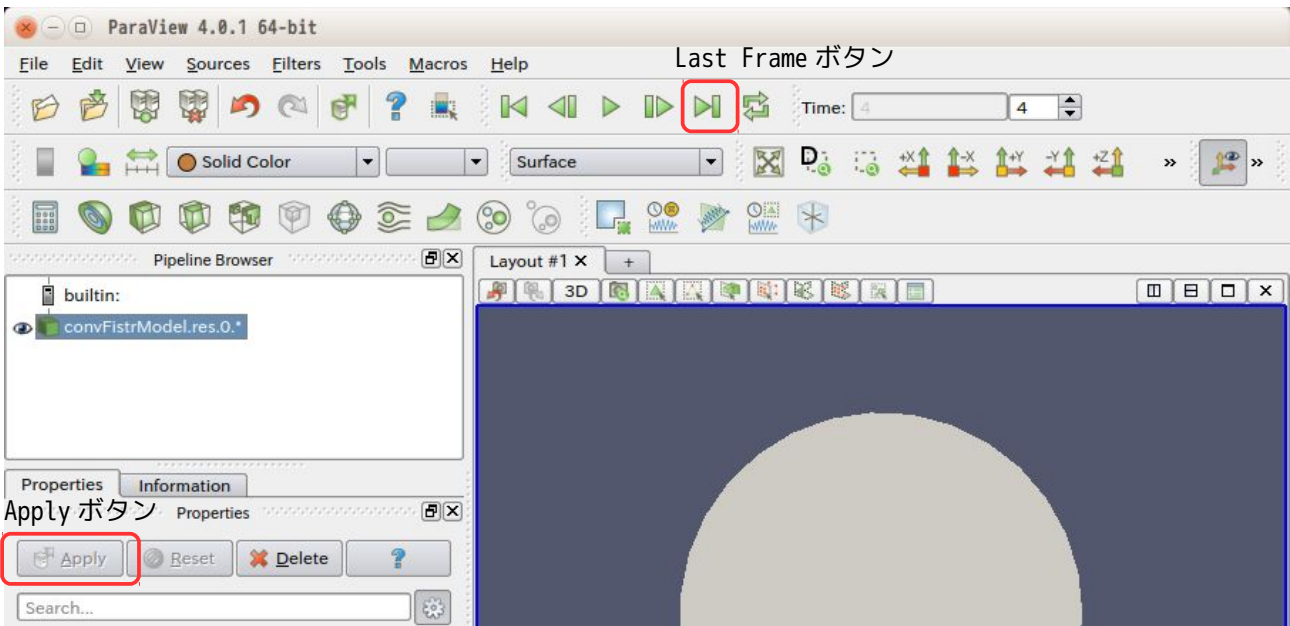


計算終了後、設定項目 Tree 内の「post」を選択し、「データ変換」、「主応力追加」、「ParaView 起動」ボタンをクリックして、結果を確認する。

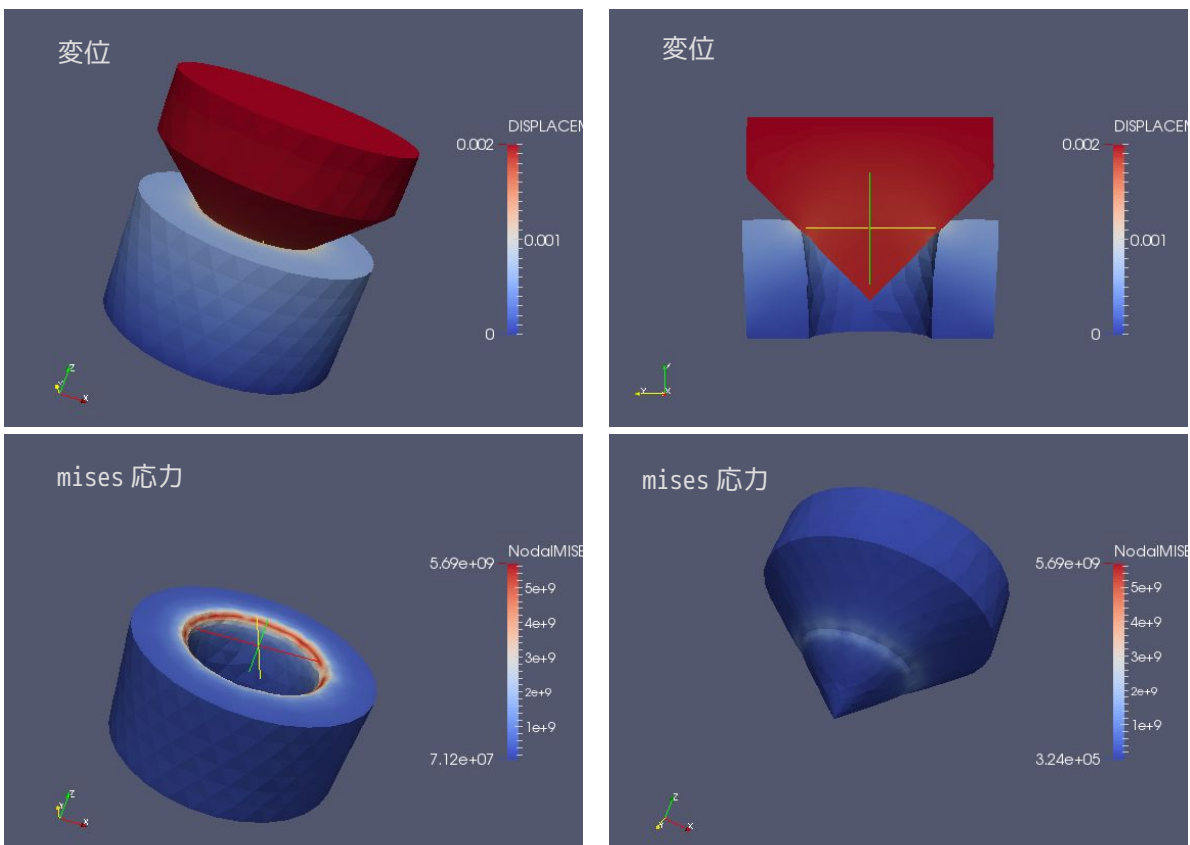
paraView 起動後、結果ファイルは、「convFistrModel.res.0...vtk」になる。

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。paraView を起動すると、これらファイルを読み込んだ状態で起動するので、起動後

「apply」ボタンをクリックしてモデルを表示させ、「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。



最大 mises 応力の値は、plotStepMonitor でグラフ表示された値とほぼ同じ値であることが確認できる。

4-1-3. 荷重拘束の接触解析：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、弱いばねを追加して、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) に弱いばね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	弱いばね追加
topN	cone 上面	等分布荷重を設定

ring 底面 (fix) に変位拘束 (固定) を設定する



cone 外側円筒面 (side) に弱いばねを追加する。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。)

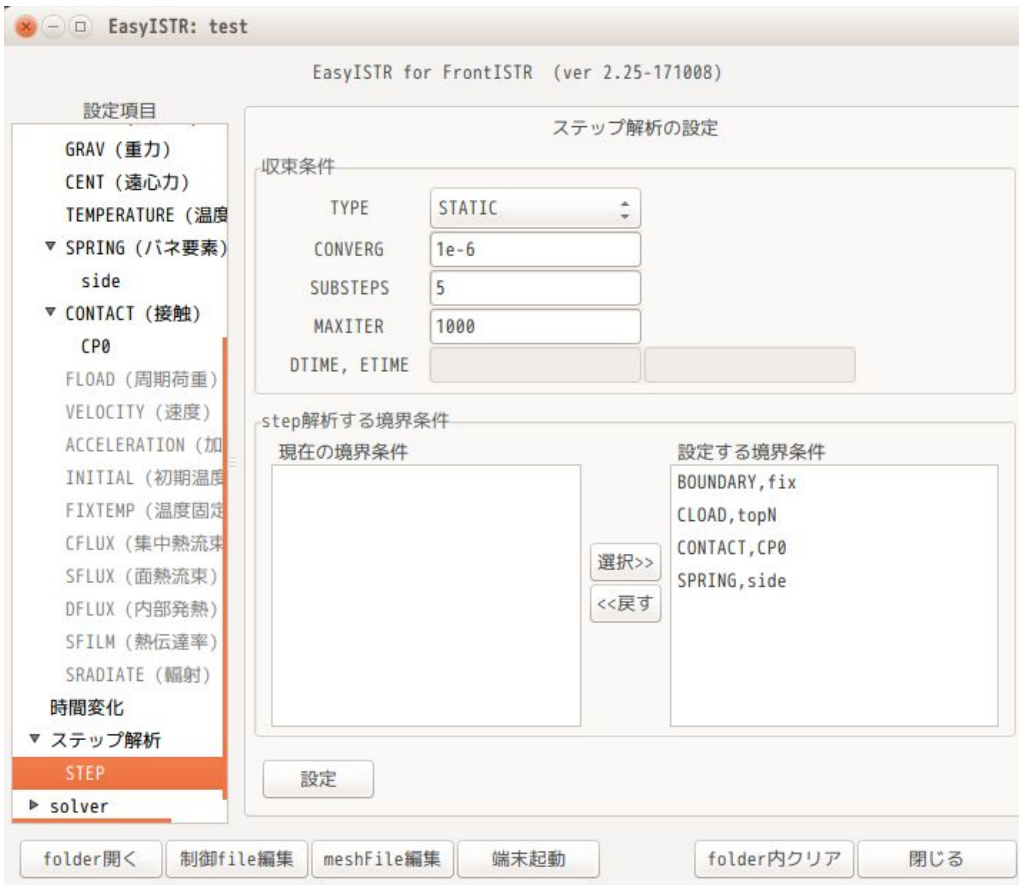


荷重は、topN に合計 $-3.0e6$ N の等分布荷重を掛ける事にする。



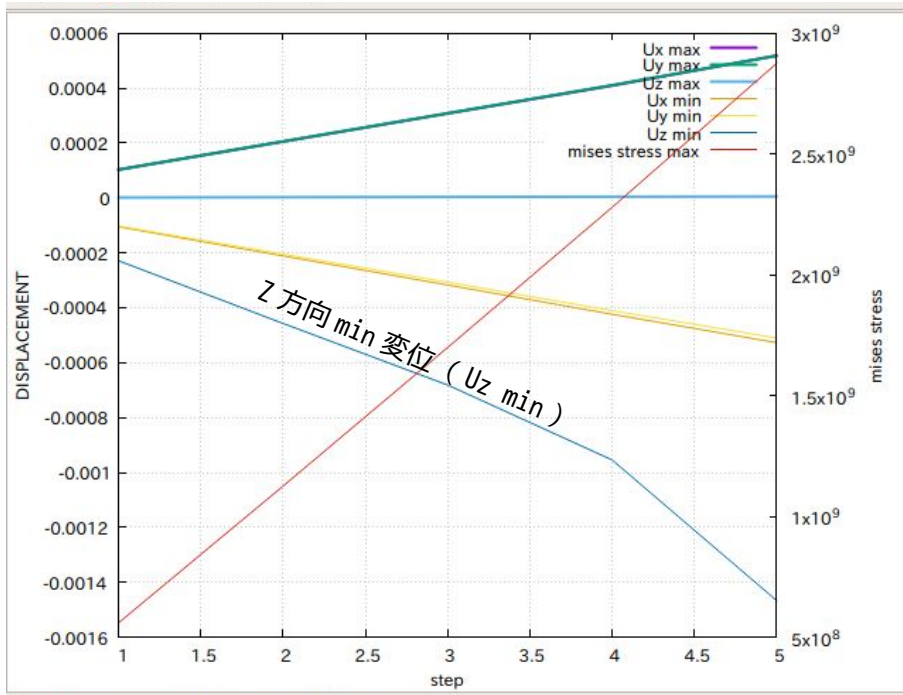
4-1-3-2. ステップ解析設定

以下のように設定している。

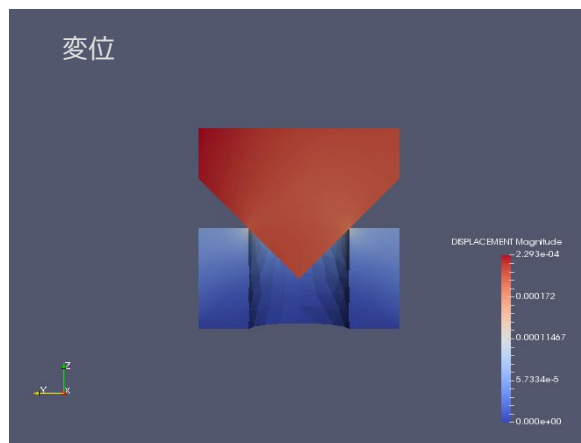
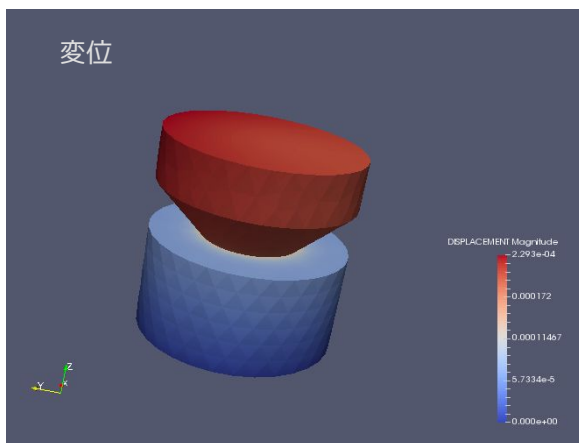


4-1-3-3. 計算開始、結果の確認

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。計算開始後、「setp 状況確認」ボタンをクリックして、plotStepMonitor を起動し、各 step 状況を確認した結果が以下になる。この結果から、Z 方向の min 変位が順調に下がっており、計算がうまく行っている事が確認できる。

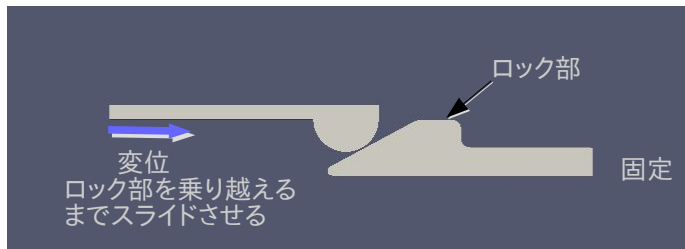


paraView で計算結果を確認した結果が以下になる。(剛体移動が発生せず、うまく計算できている。)



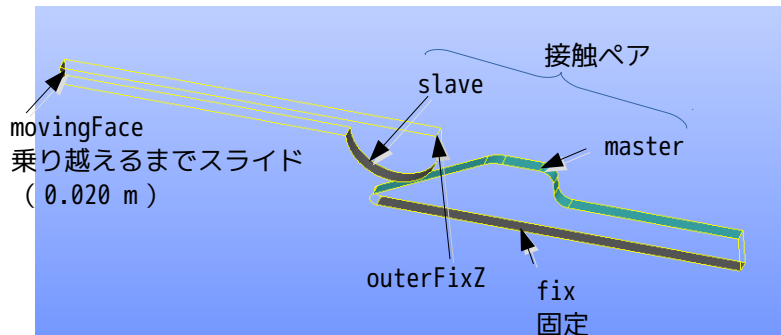
4-1-4. すべりのあるモデル (スナップフィット) の接触解析

前項までの接触解析は、殆ど滑りが無いモデルだが、ここでは滑りのあるモデルを使って解析してみる。モデル形状は、以下の様に、可動側がロック部をスライドして滑っていくスナップフィット構造を解析する。



4-1-4-1. モデル形状

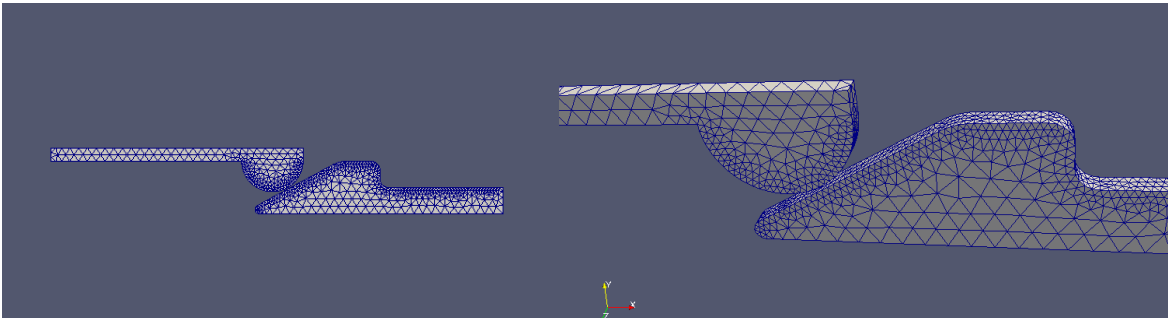
前項の様なモデル形状で、グループ化は、下図のように行っている。モデルは、厚さを持っている3次元モデルになっている。



group 名	区分	
outer	要素	可動側
inner	要素	固定側
MST_outer	面	接触面 (master 側)
movingFace	節点	この面を 20mm 移動させる。
fix	節点	固定部
SLV_inner	節点	接触面 (slave 側)
outerFixZ	節点	Z 方向の固定部 (可動側先端の edge 部)

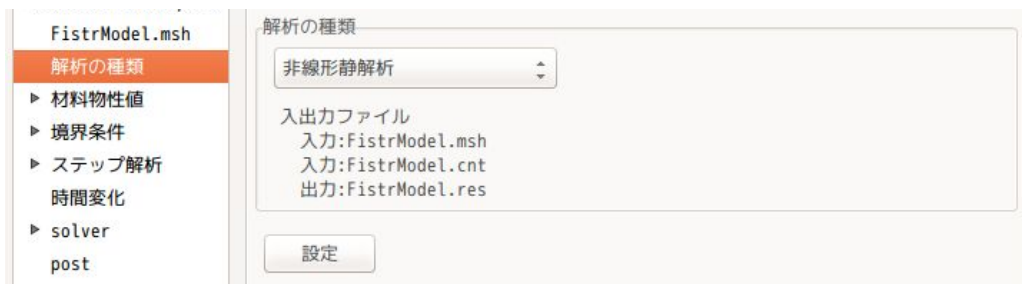
この形状のメッシュは、「snapFit2.unv」として保存されているので、このメッシュを Fistr 用の msh 形式に変換後、倍率を 1/1000 に変更しておく。

以下が、メッシュ変換後、メッシュを確認した結果になる。



4-1-4-2. 設定内容

解析の種類は、「非線形静解析」とし、材料は、両部品とも「Aluminum」としている。



境界条件は、「fix」を固定し、「movinfFace」をX方向に「0.020」（20mm）移動させ、「outerFixZ」のZ方向を固定する設定にしている。





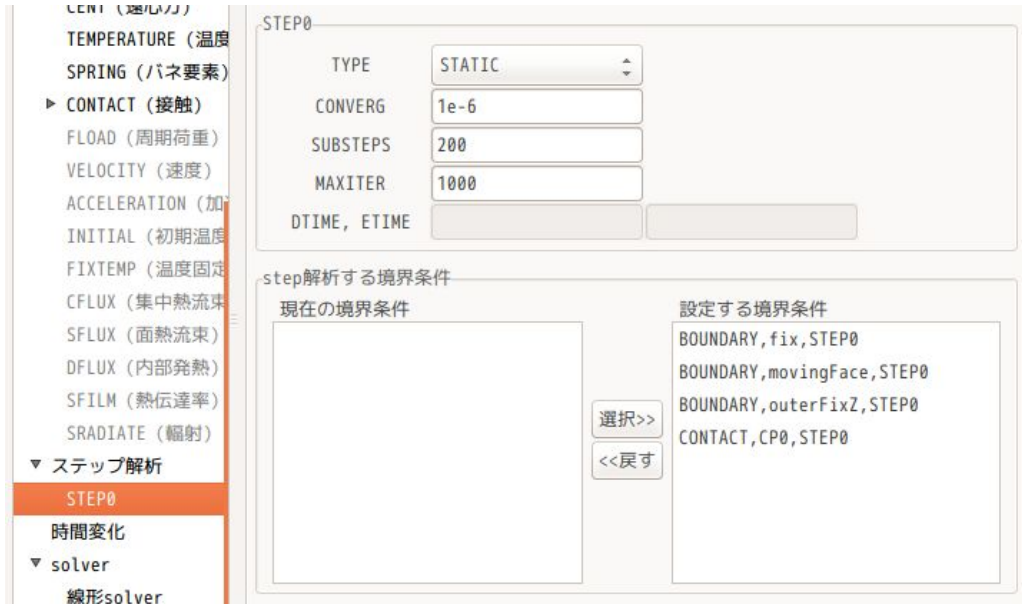
接触の設定は、「Lagrange 乗数法」としている。



今回の場合、接触面で滑っていくので、「FSLID (有限すべり)」とする。(「SSLID (微小すべり)」に設定すると、今回の場合解析できない。) また、摩擦係数は、「0.1」を設定した。

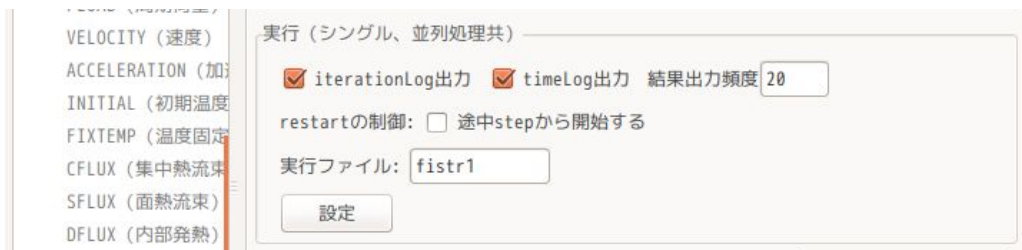


ステップ解析は下図のように、「SUBSTEPS」を200と設定した。20mm 移動させていくので、0.1mm/1step 間隔で計算を進めていく設定になる。今回の場合、接触面に R があるので、この部分で接触状態が変化して収束し難くなる為、0.1mm 間隔で計算していく。



4-1-4-3. 計算開始

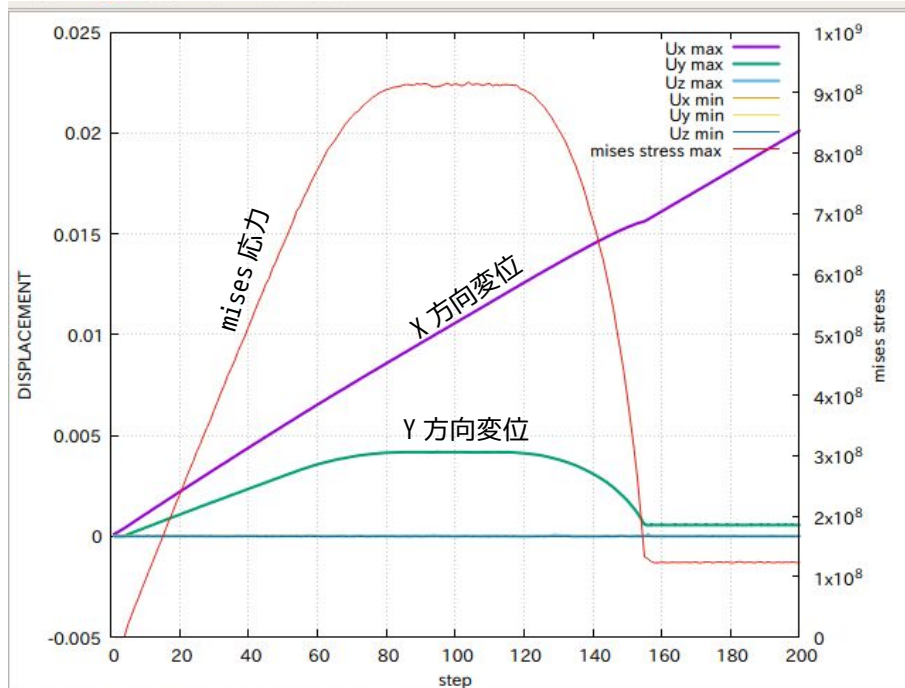
計算は、SUBSTES が 200 の為、結果の出力頻度を「20」に設定し、1 回 / 20step で出力する設定。



また、今回のモデルは、規模が小さいので、線形 solver を「MUMPS」に設定した。



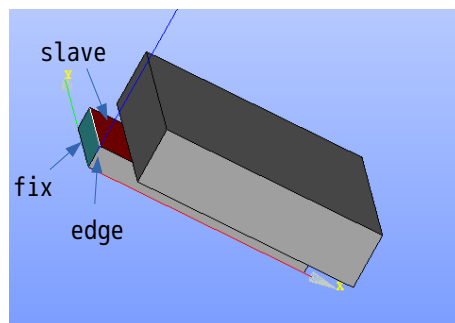
以上の設定で、計算を開始する。計算開始後、「step 状況表示」ボタンをクリックして、plotStepMonitor を起動して、計算状況を確認した結果が以下になる。計算は、うまく走っている。



計算結果は、データ変換後、paraviewで確認できるが、省略する。

4-1-5. 多数の接触部が存在する場合の接触解析

接触解析を行う場合、nodeGroupとして定義したslave側の接触部と、変位を拘束する為に定義したnodeGroupが、重複して定義されていた場合（下図参照）、接触解析中にエラーが発生する可能性がある。



fixとslaveが交差するedgeで、各々のgroupの節点が重複定義されている場合、接触解析中にエラーが発生する事がある。

slave側接触部は、計算過程で節点が移動して、荷重バランスを取りながら計算が進んでいくが、この節点が、boundaryで変位拘束されてしまうと、計算が進んで行かない事になってしまう。この事は、master面の節点と、他接触ペアのslaveの節点が重複定義されても同様にエラーが発生してしまう。特に多数の接触ペアが存在するようなモデルでは、エラーが発生しやすいので注意が必要になる。

また、前記した様な重複定義された箇所では、先に定義されたnodeGroupが有効で、後で定義された側の重複節点は、無視される。（FrontISTRのユーザマニュアル参照。）

nodeGroup間で節点が重複定義されてしまう事は、避けた方が望ましいが、nodeGroupの定義をCADを使って定義する場合は、通常edge部で節点が重複して定義されてしまう。これを避ける為には、edge部に小さな面取りを行って、各々の面がedgeを共有しない様な工夫をするか、

重複する節点を nodeGroup から削除する事が必要になる。

多数の nodeGroup や多数の接触ペアを定義する場合は、特に注意が必要になってくる。
ここで、多数の接触ペアを持つモデルの解析を行って、重複節点の扱い方について確認する。

また、接触は、master 面に slave の点が接触し、節点 (slave 点) の座標値を計算している。この為、接触面でどちらを master、slave とするかは、以下の考え方で決定する。

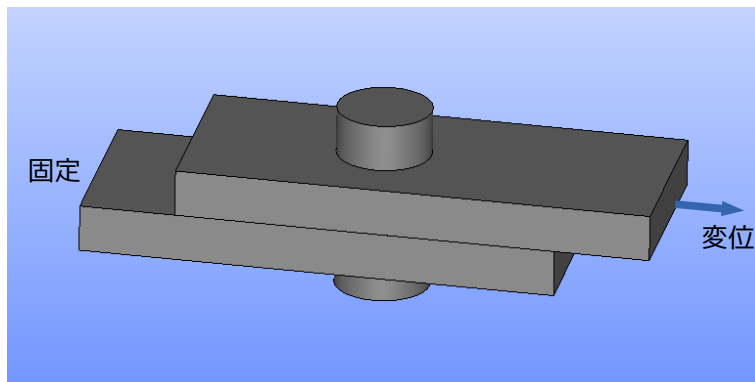
master 面: 面積が広く、接触による変位が少ない側。

slave 点: slave 側の node の変位を計算するので、接触による変位が大きい側。

master と slave を誤って設定した場合、接触判定できない事があるので注意。

4-1-5-1. モデル形状

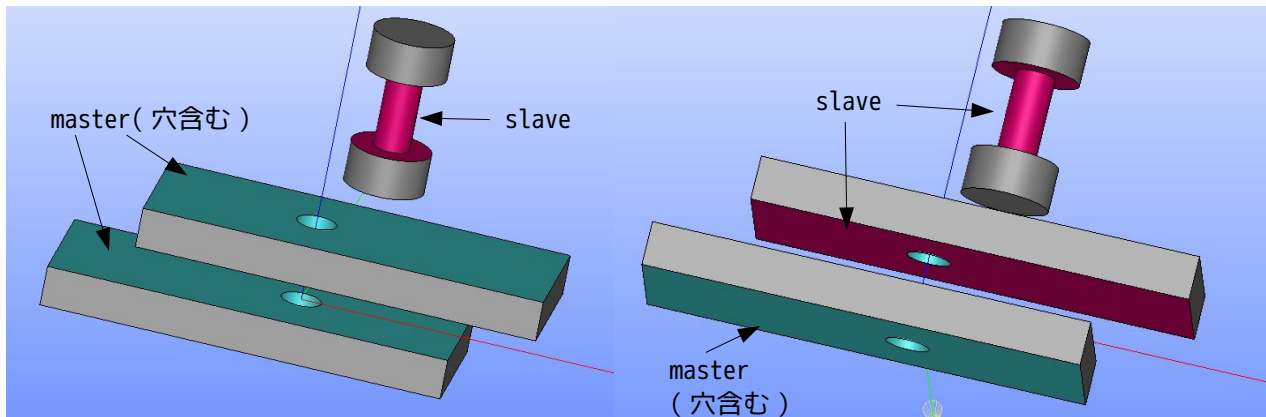
2枚の板をボルトで締結するモデルを考える。ボルト長は、被締結物の板厚よりも短めに設定し、ボルトに軸力が発生する状態にしている。(初期状態でボルトが板に食い込んでいる状態。)



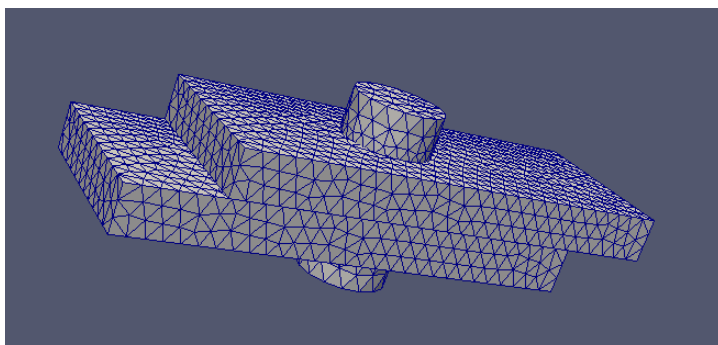
ボルト
頭径: $\Phi 20\text{mm}$
軸: $\Phi 8 \times 19.5\text{mm}$

上下板
 $100 \times 40 \times t10\text{mm}$
穴径: $\Phi 10\text{mm}$

接触ペアは、以下の様に定義する。ボルト上下座面、軸を slave として、ボルトと接触する板側を master としている。板同士も接触するので、各々 master、slave として定義する。



この solid モデルで以下の様にメッシュを作成し、「platesBolt_org.unv」として保存している。



メッシュの group 化は、以下の様に行っている。

group 名	区分	備考
bolt	volume	ボルト
plateD	volume	下側板
plateU	volume	上側板
MST_plateDbolt	face	下側板の下面 (ボルト頭部と接触)
SLV_boltPlateD	node	ボルト下部座面 (下側板と接触)
MST_plateUbolt	face	上側板の上面 (ボルト頭部と接触)
SLV_boltPlateU	node	ボルト上部座面 (上側板と接触)
MST_plateDhole	face	下側板の穴部 (ボルト軸と接触)
MST_plateUhole	face	上側板の穴部 (ボルト軸と接触)
SLV_boltSide	node	ボルト軸、頭座面
MST_plateDplateU	face	上側板の下面 (下側板との接触部)
SLV_plateUplateD	node	下側板の上面 (上側板との接触部)
boltTop	node	ボルト頭上面 (弱いバネを設定する面)
fixPlateD	node	下側板の端面 (固定面)
pullPlateU	node	上側板の端面 (変位させる面)

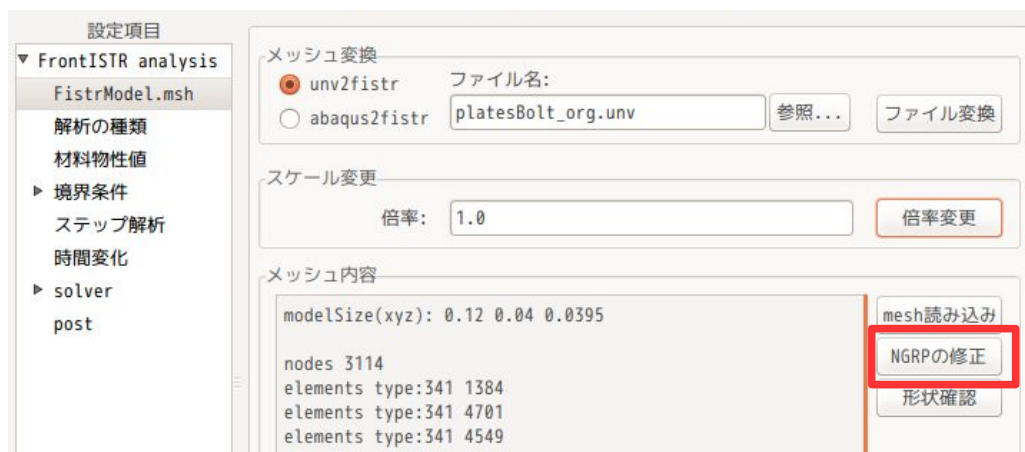
この解析を行うにあたって、ボルトの拘束が無いので、ボルトに弱いバネを設定する必要がある。この為にボルト上面 (boltTop) の group 化を行い、ここに弱いバネを設定する。

4-1-5-2. メッシュ変換と重複節点の削除

メッシュ変換は、通常通り、「platesBolt_org.unv」を指定して、メッシュ変換する。変換後、倍率を 1/1000 に設定して、単位を m に変換する。



メッシュ変換したメッシュには、nodeGroup 間や nodeGroup と master 面で節点が重複して定義されているので、重複節点を削除する必要がある。この削除のため、同じ画面上の「NGRP の修正」ボタンをクリックして重複節点を削除する。(下図参照)



以下の画面が現れるので、この画面上で、重複節点を削除する。今回の場合は、edge 部で節点が重複しているので、重複している nodeGroup 「SLV_boltPlateD」 「SLV_boltPlateU」 「SLV_boltSide」 「SLV_plateUpalteD」 「pullPlateU」 が表示されている。(重複していない nodeGroup は、表示されない)

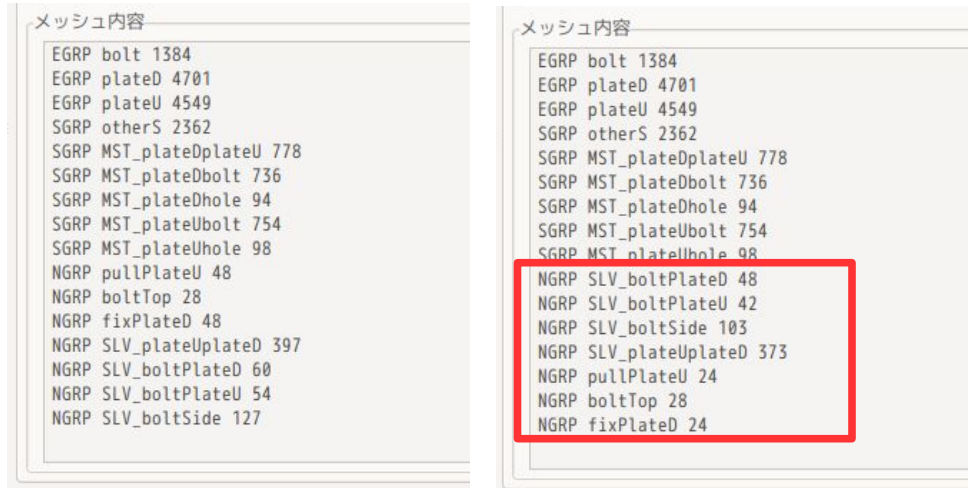


今回の場合、重複している節点 (edge 部) を両 group から削除し、且つ、nodeGroup と面 group 間で重複している節点も削除する。この為、この状態で、「OK」ボタンをクリックする。これにより、重複節点全てが削除される。(nodeGroup 間の edge の節点が削除されるので、edge 部に面取りをした事と等価になる。)

重複節点削除前後の node 数を確認すると、以下の様に node 数が変化している。

削除前

削除後



尚、nodeGroup に関して、上記設定では、重複している節点を両 group から削除する設定の為、重複している箇所（edge 部）の節点が両 group から削除されてしまうが、どちらかの group にその節点を残す事もできる。その方法は、まず、「重複箇所の節点を全 NGRP から削除する（Edge の節点を削除）」のチェックを外す。（下図は、チェックボックスのチェックを外した状態）



テキストボックス中に互いに重複している nodeGroup 名が明瞭に表示される。

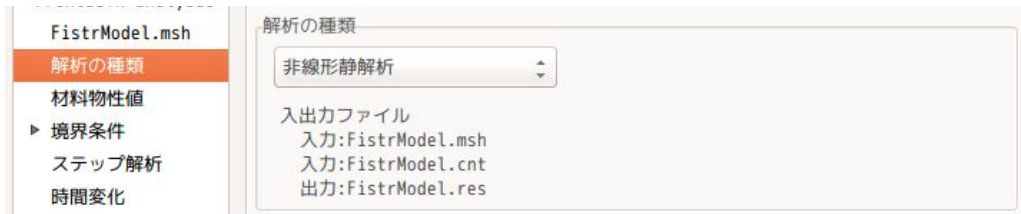
この状態は、表示されている nodeGroup を順番に確認し、相手側の節点を削除していく。この為、重複節点を残したい nodeGroup をリスト上部に移動する事で、その group は、節点が削除されない（相手側の nodeGroup が削除される）事になる。逆にリスト中の最後の nodeGroup は、重複節点が必ず削除される事に

なる。

以上の様に、重複節点を削除したくないnodeGroupをリスト上部に移動し、重複節点（edge部）を削除しても構わないnodeGroupをリスト下部に移動して、リストの順番を変更することで、望む状態にする事もできる。

4-1-5-3. 計算条件の設定

解析の種類は、接触解析の為、「非線形静解析」を選択する。



材料は、ボルト：Steel、上下板：Aluminumを設定する。



解析の方法として、

- 1) ボルトに軸力が発生する状態（ボルト長が板厚より0.5mm短い）の為、まず、この解析を行う。
下側板の端面（fixPlateD）、上側板の端面（pullPlateU）を固定する。
接触ペアをボルト座面（SLV_boltPlateD、SLV_boltPlateU）と上下板（MST_plateDbolt、MST_plateUbolt）の接触部と、上下板間（SLV_plateUplateD、MST_plateDplateU）のに設定。
ボルトが拘束されていないので、ボルト頭部（boltTop）に弱いバネを設定する。
- 2) 第2ステップとして、上側板（pullPlateU）を2mm引っ張る。
接触ペアとして、ボルト軸（SLV_boltSide）と上下板の穴（MST_plateDhole、MST_plateUhole）を追加

する。

の2ステップに分けて設定する。

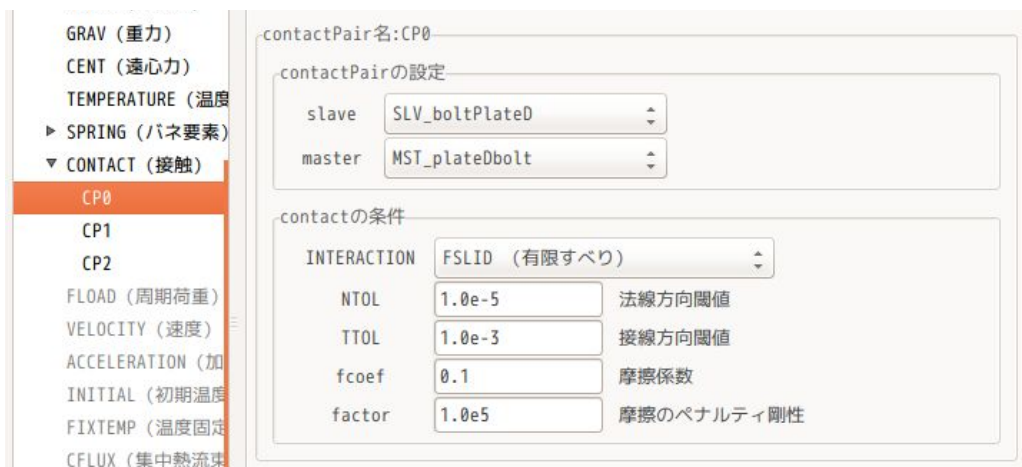
まず、第1ステップとして、1)の設定を行う。
boundaryとして、fixPlateDとpullPlateUを固定する。

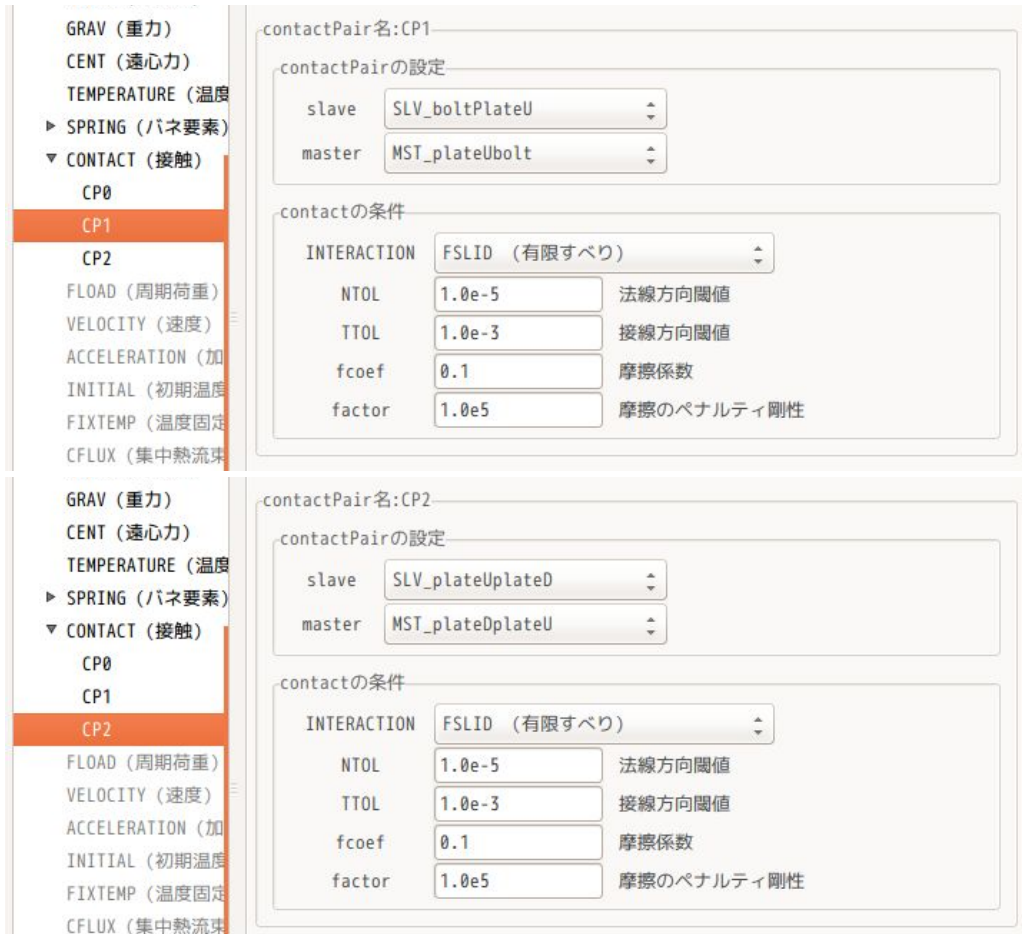


ボルト上部「boltTop」に弱いバネを設定する。XYZ各方向共「1e5」を設定。



接触ペアは、以下の3箇所を設定している。摩擦係数「0.1」を[設定している。





ステップ解析は、以下の様に、STEP0とSTEP1を設定する。(第2ステップまで計算するため)



第1ステップでは、以下で設定している。尚この時の subStep は、1回で設定している。ここでの境界条件は、「fixPlateD」「pullPlateU」を固定(変位「0.0」)の為、subStepを複数回設定しても、増分は「0.0」の為、subStepを複数回設定しても意味がないので、1回で設定した。



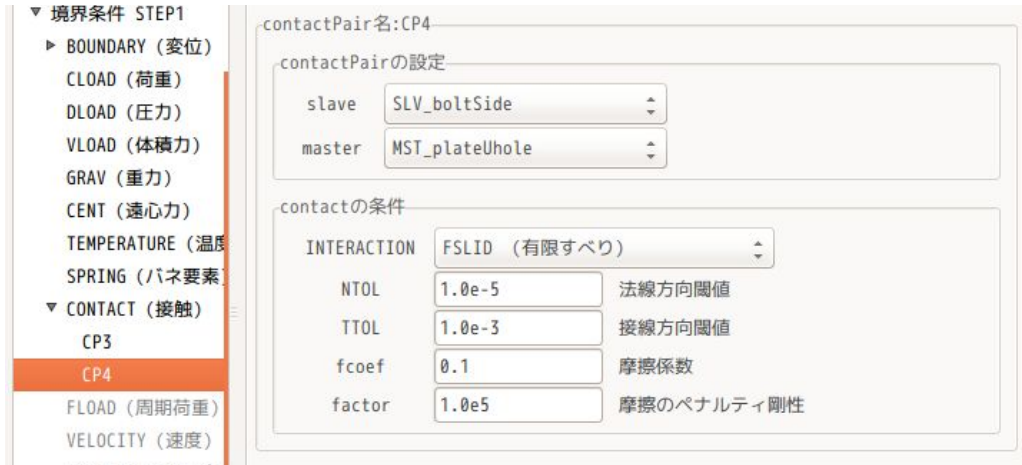
第2ステップ (STEP1) の境界条件を設定する。第1ステップで設定した境界条件から変更、追加する部分のみ設定する。

boundary は、上側板端面を X 方向に「2mm」引っ張る設定を「境界条件 STEP1」内に追加する。

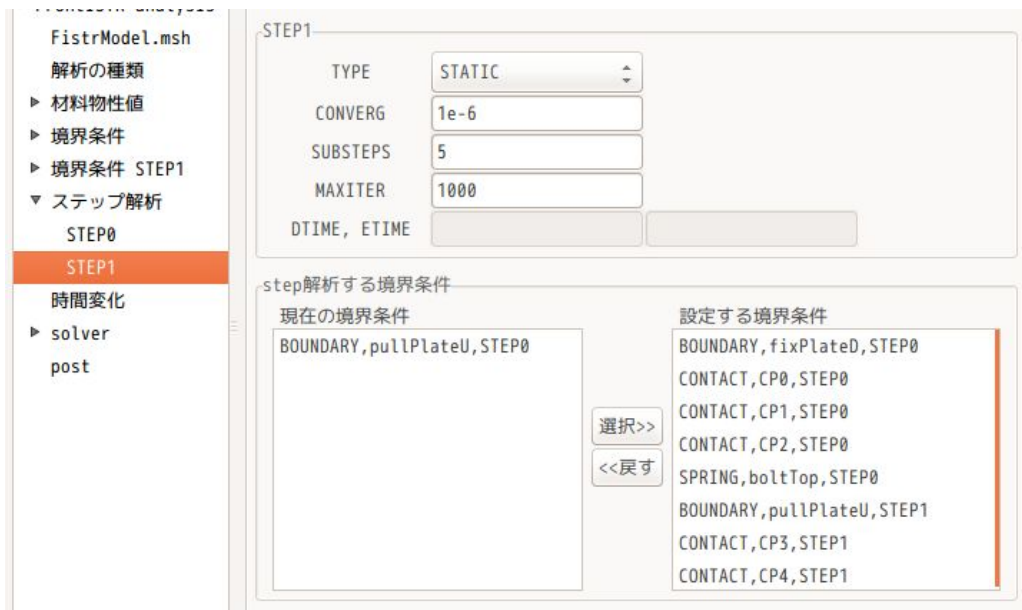


ボルト軸と上下板の穴部の接触ペア (2箇所) を追加設定する。摩擦係数「0.1」を設定している。



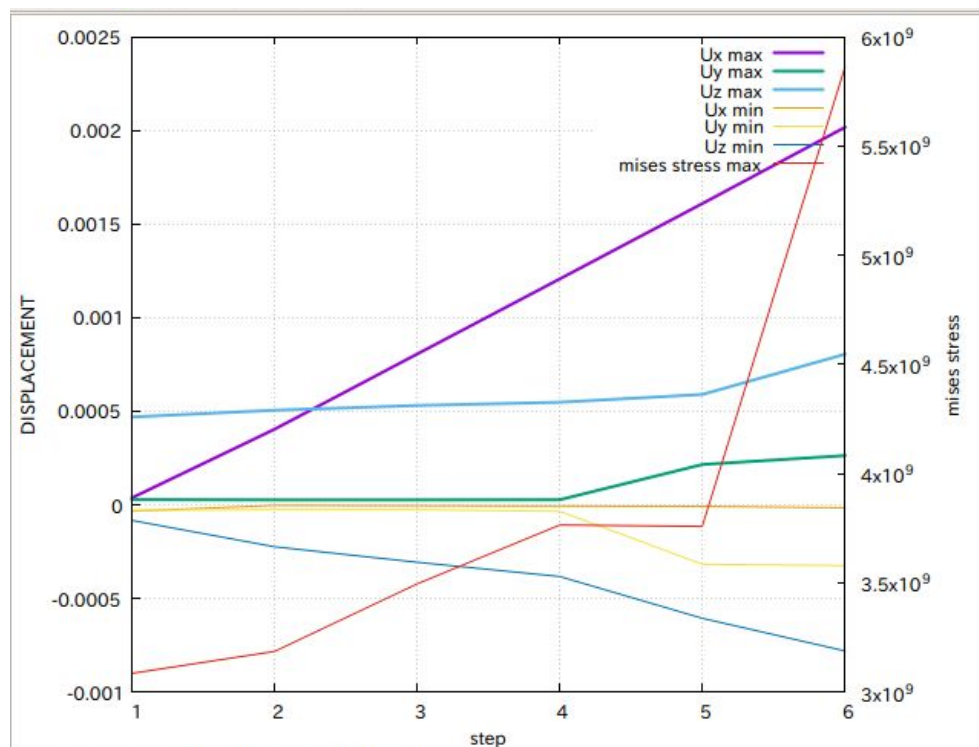


ステップ解析の内容は、以下で設定している。
 subStep 数は「5」で設定し、変位 2mm を 5 分割して計算させる。
 上側板の端面の境界条件が変わるので、STEP0 で設定した boundary の pullPlateU の固定の条件は外し、STEP1 で追加した接触ペアを追加している。

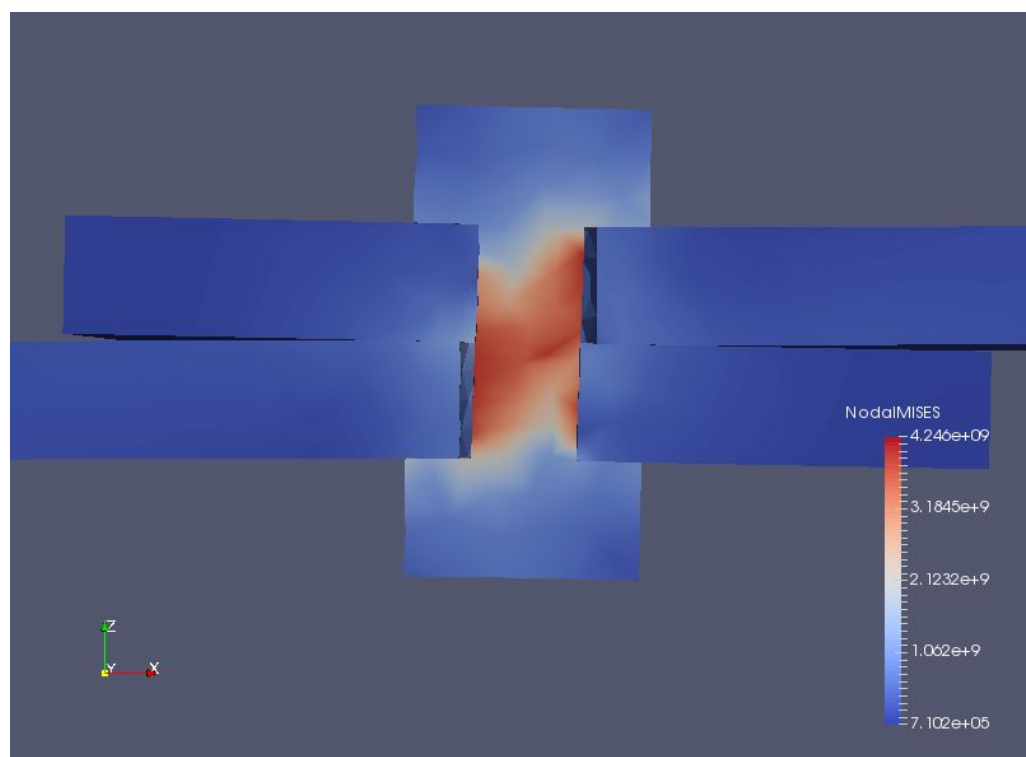


4-1-5-4. 計算開始、結果の確認

以上で全ての設定が終了したので、計算を開始させる。下図がその計算状況になる。



ボルト断面の状況は、以下になる。うまく接触解析ができています。



この解析中で、nodeGroup間の節点が重複節点を削除せずに計算を進めていくと、途中でエラーが発生するが、重複節点を削除する事で、上記の様にうまく計算が進み、結果が確認できる。

4-1-6. 接触解析における並列処理方法

接触を含む解析の並列処理は、通常通りの並列処理方法（3-9 項の方法）では、計算できない。これは、通常通りメッシュを並列処理用に分割した場合、接触ペアとなる 2 面を同一のプロセスに与える事ができていない為、接触判定ができなくなる為。接触判定する時は、メッシュ分割していない単一メッシュで判定するようにすれば、この問題は解決する。このような計算方法をさせる為には、FrontISTR をビルドする時に、以下の様な「--with-paracon」オプションを追加して、ビルドしておく必要がある。

```
$ ./setup.sh -p --with-tools --with-refiner --with-metis --with-mumps -with-lapack \
  --with-ml --with-paracon
```

これにより、接触解析の並列処理ができる状態になる。実際に並列処理する場合は、以下の手順を踏む事になる。

- 1) 接触解析する為の境界条件を全て終えておく。
実際に計算を走らせて、うまく走ることを確認しておく。
- 2) メッシュを並列数分分割する。
ここまでは、通常の並列処理方法と同じ。
- 3) EasyISTR 上の「制御 file 編集」ボタンをクリックして、「hecmw_ctrl.dat」を編集する。
以下のように、2 行を修正する。

```
#
# for solver
#
!MESH, NAME=fstrMSH, TYPE=HECMW-DIST
FistrModel_p4
  ↓
#
# for solver
#
!MESH, NAME=fstrMSH, TYPE=HECMW-ENTIRE #修正（単一メッシュ）
FistrModel.msh #修正
```

修正後、保存しておく。

- 4) EasyISTR 上の「FrontISTR 実行（並列）」ボタンをクリックして、計算させる。
通常通り計算開始させる。

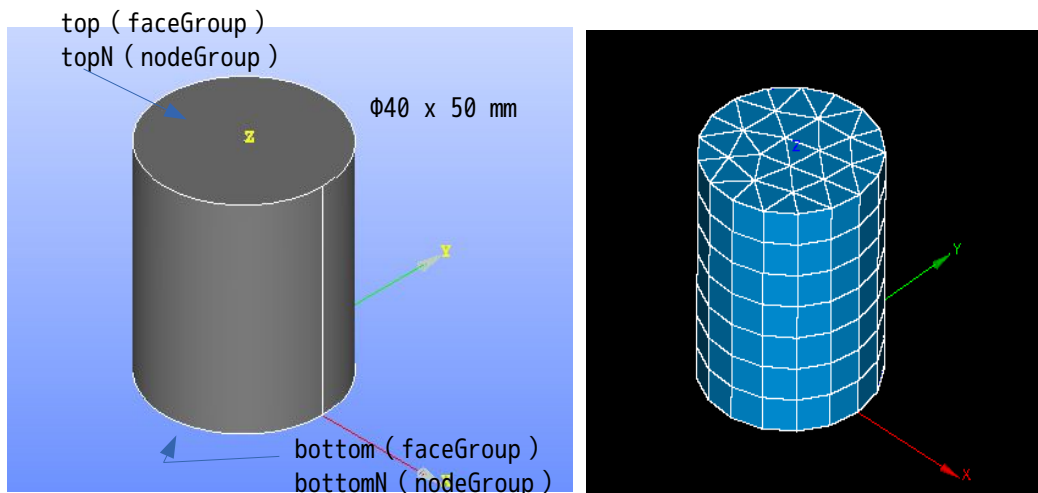
以上の方法で、接触解析を並列処理させることができる。ただし、今のビルド方法は、FrontISTR がマルチスレッドで実行できる状態のため、シングルコアで計算開始させてもマルチスレッドで計算しているため、実行速度は、mpi による並列処理よりも早くなっている。多くの cpu を持っているサーバ等を使って処理する場合は、この方法が有効かもしれない。

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>

3D: 3D Extrusion
2D: Quadrangle(Mapping)
1D: Wire Discretisation
Local Length
7.5

< SubMesh >

2D: Netgen 2D
max size 7.5
1D: Wire Discretisation
Local Length
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。

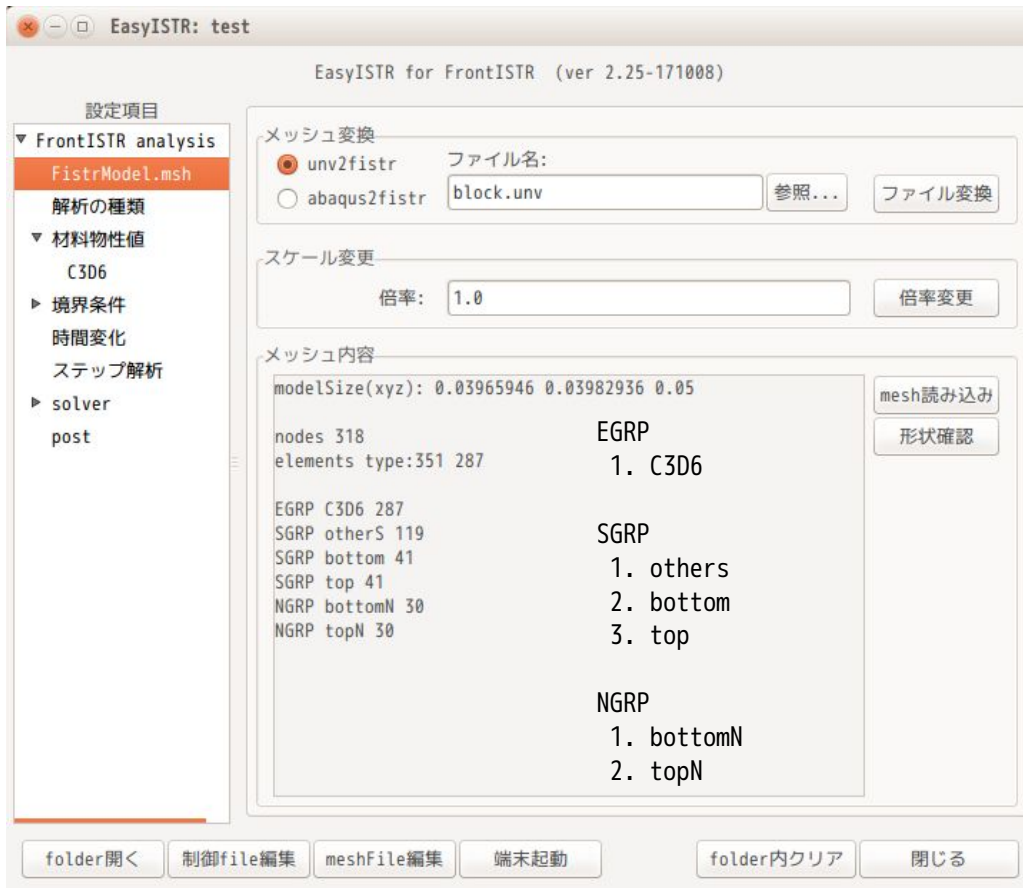
4-2-2. 圧縮 20%の計算

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

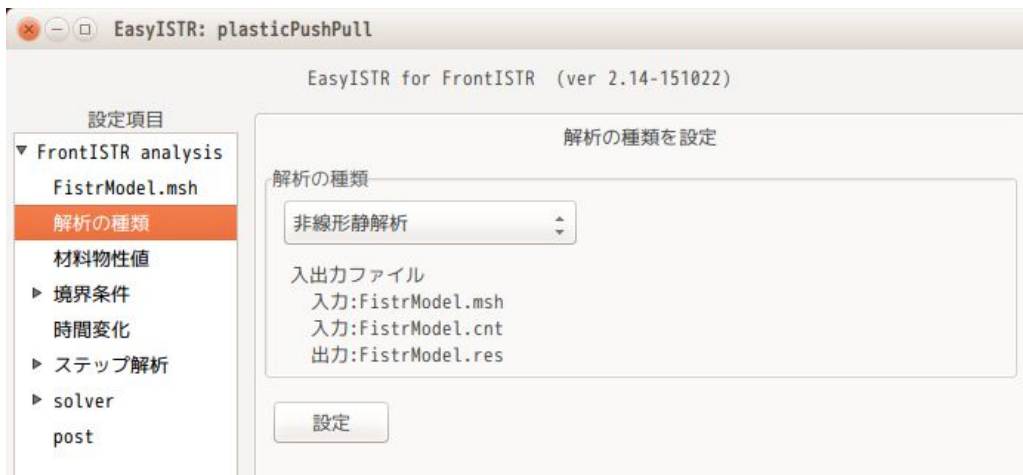
メッシュ変換の為に、workFolder内の「block.unv」を選択して、メッシュ変換し、モデルの倍率を 0.001 にしておく。下図が変換した状態になる。

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)

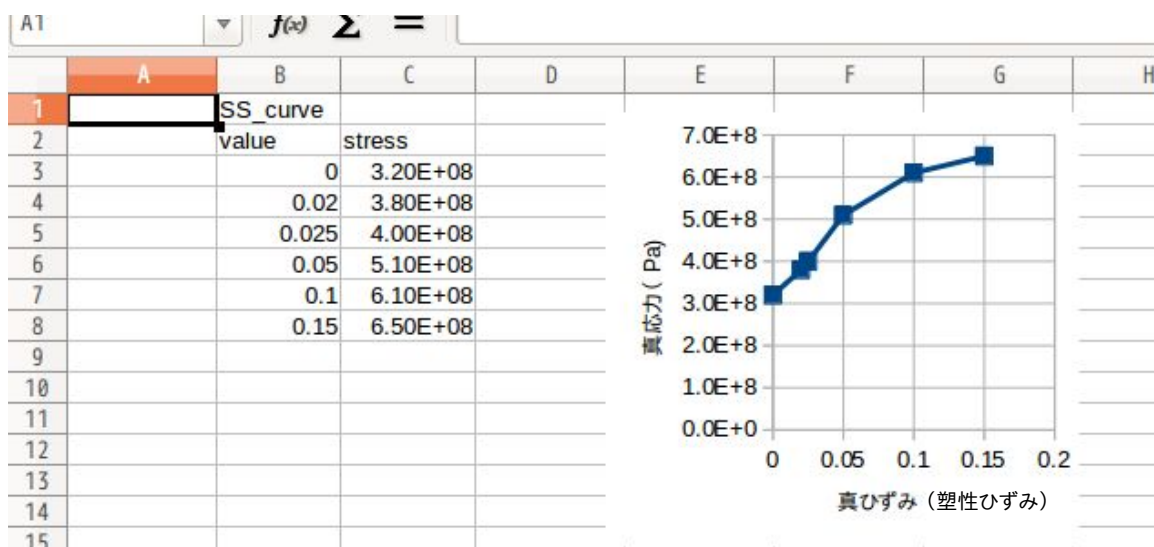


4-2-2-3. 材料物性値の設定

材料は、Steel で材料モデルは、以下で設定。

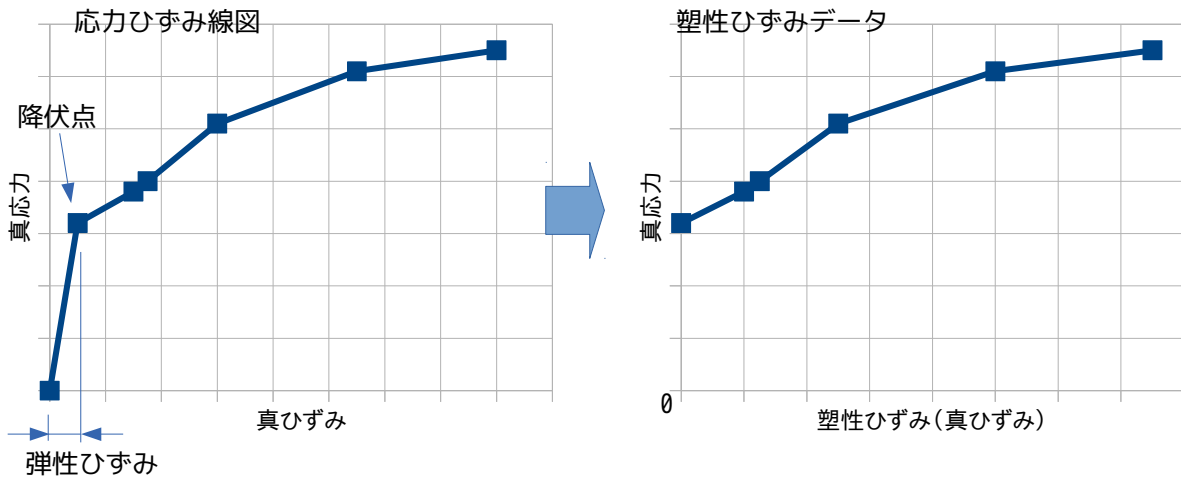
材料モデル： PLASTIC 塑性
 降伏条件/タイプ： MISES
 硬化則： MULTILINEAR 多曲線近似（等方硬化則）

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、弾性ひずみを含まない塑性ひずみデータを入力する。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、temp フォルダ内に保存される。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。（実行時には、workFolder に転送される。）



FrontISTR では、この入力データは、塑性ひずみとその時の降伏応力のデータを入力する。応力ひずみ線図

から以下の様にデータを取り出す。



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。



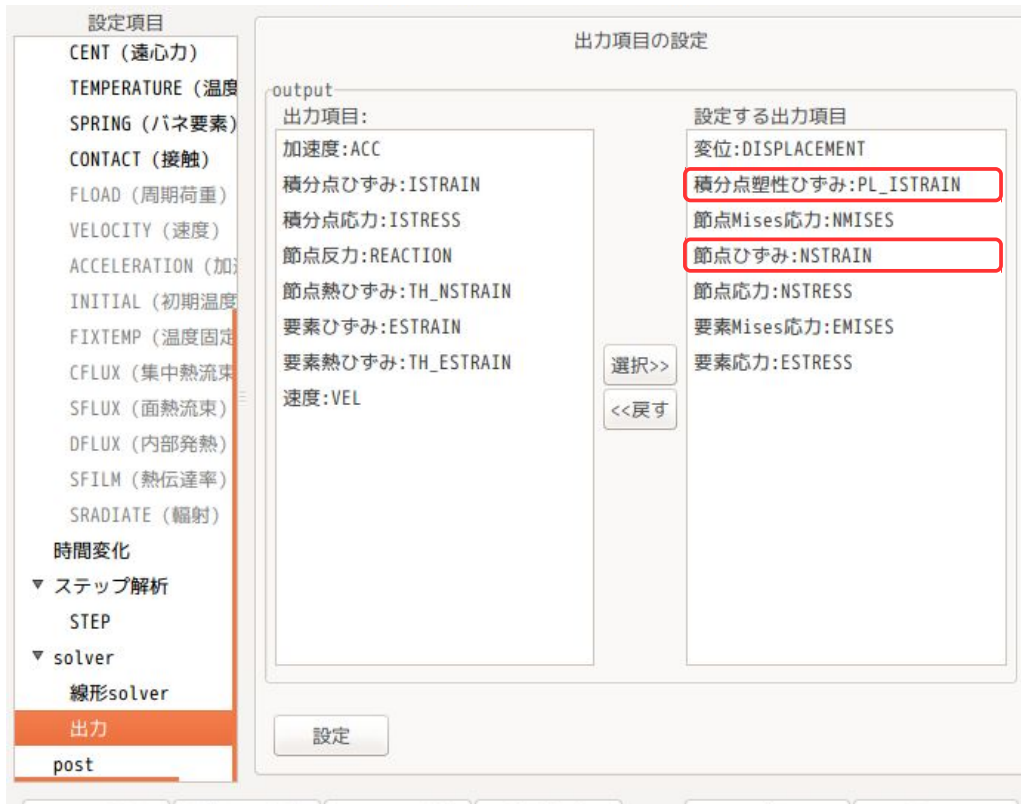
4-2-2-5. ステップ解析の設定

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。



設定が終了したので計算を開始させるが、次項で、計算させた後さらにその結果を使って計算を継続させる (restart) 為、その設定を行っておく。
 計算を restart できる様にする為には、下図の「途中 step から開始する」のチェック Box にチェックをし、「設定」ボタンをクリック。さらに「途中 step から開始する」のチェック Box のチェックを外し、「設定」ボタンをクリックして、元に戻しておく。



この操作により、制御ファイル内に「!RESUTART, FREQUENCY=1」の行が追加される。

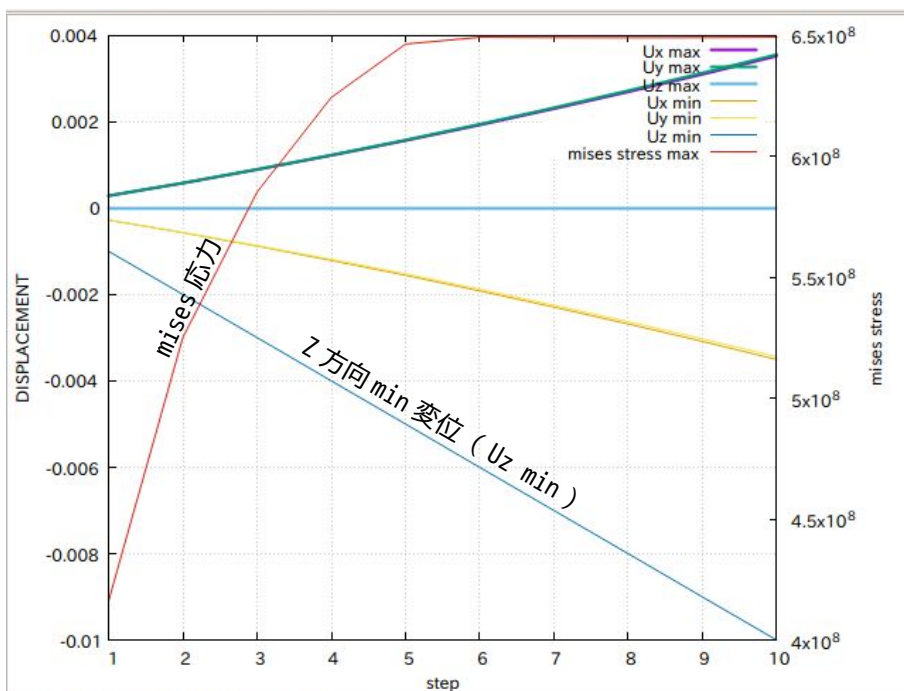
```
----- FistrModel.cnt -----
:
#####
# SOLVER CONTROL #
#####
!RESTART, FREQUENCY=1
!SOLVER,METHOD=CG,PRECOND=1, ITERLOG=NO, TIMELOG=YES
```

20000, 2
 1.00000e-06, 1.00000, 0.00000
 0.100000, 0.100000
 :

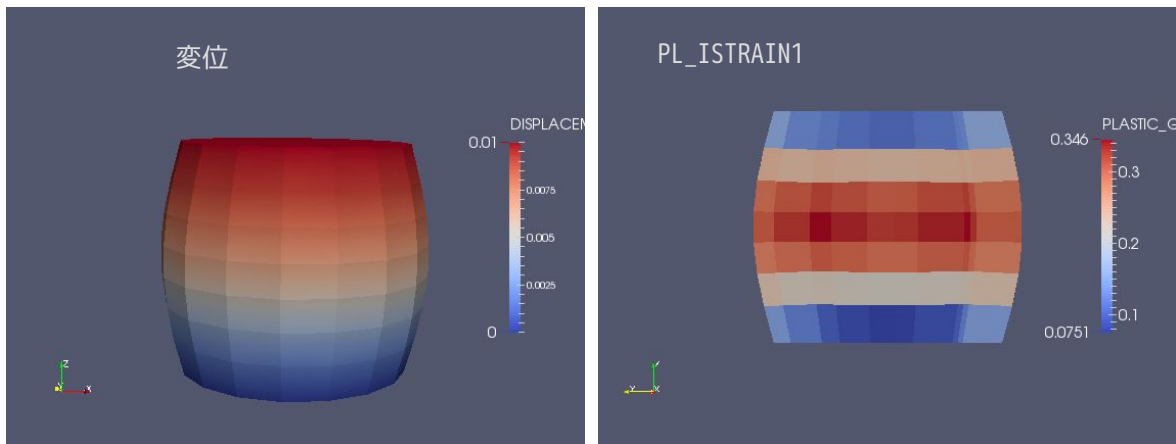
計算は、下図の様に、通常通り「FrontISTR 実行ボタン」をクリックして計算を開始させる。
 計算開始後、「step 状況表示」ボタンをクリックして、計算状況を確認する。



plotStepMonitor の表示は、以下になる。mises 応力が 5step 付近から上昇しておらず、この付近から塑性変形が大きくなっている事がわかる。また、Z 方向の min 変位が、最終的に「-0.01」であり、Z 方向に 0.01m 変異している事がわかる。



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 paraView で確認する。



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。

この計算は、前回の計算結果をそのまま使って、restart させる解析になる。この設定は、restart ファイルができあがっていない、restart 中にエラーが発生、等した場合には、restart 計算ができない。また設定が複雑になる事もあり、時間が許されるなら、次項の複数 step を使って計算させる方が設定が楽。

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。



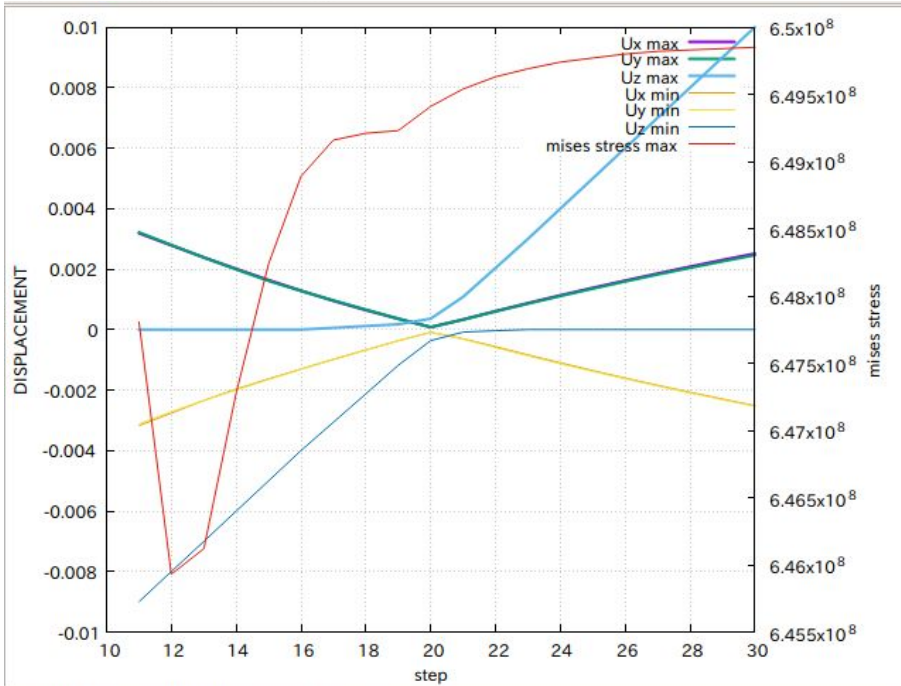
STEP の設定は、SUBSTEP の値を「30」に修正する。下図参照。



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。

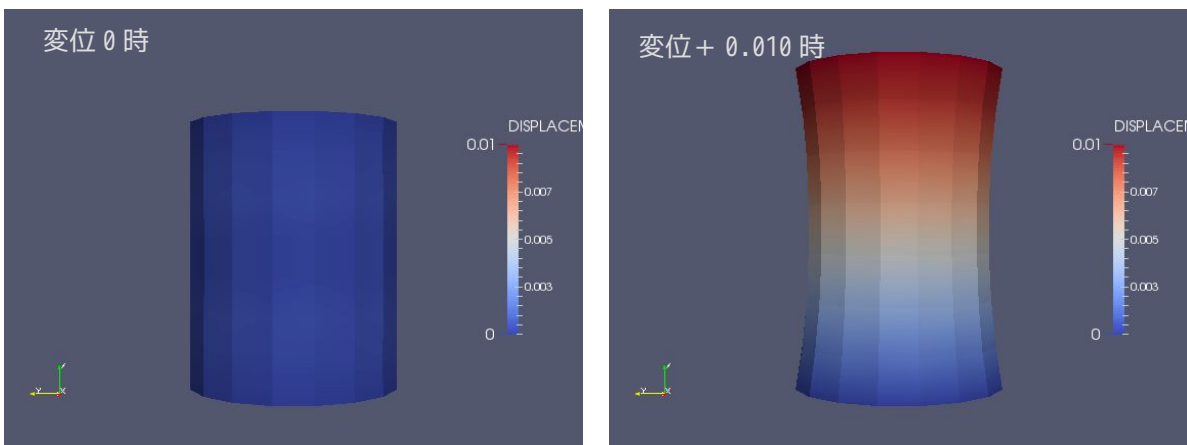


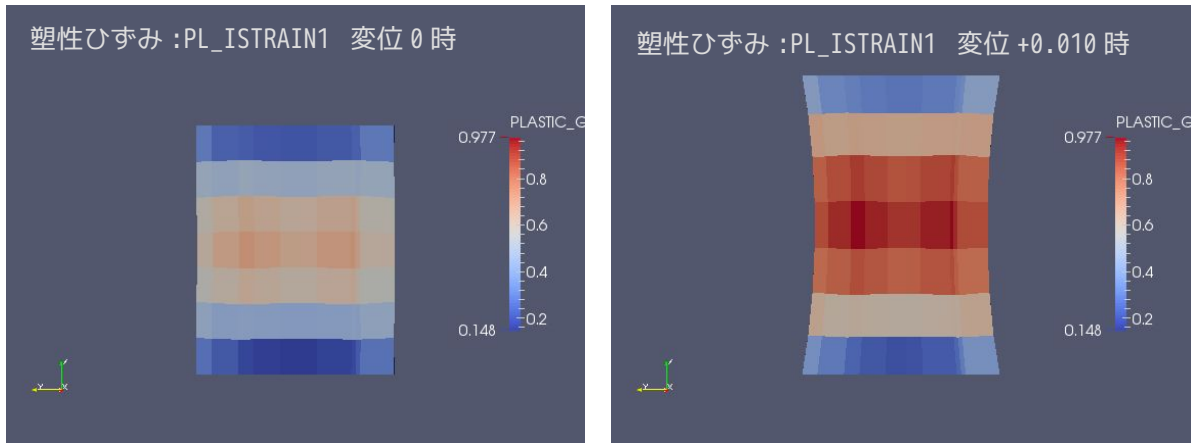
計算開始後、「step 状況表示」ボタンをクリックして、plotStepMonitor を起動すると、以下の step 状況が表示される。



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」のときと「 $+0.010$ 」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。





4-2-4. 複数 step による一括解析

前項では、非線形解析した後、その結果を使って restart させて、連続して非線形解析を行ったが、ここでは、各々の解析を各 step に分けて設定し、全 step を一括して解析してみる。

4-2-4-1. モデルとメッシュ作成

モデルは、前項と同じメッシュ「block.unv」を使う。このファイルを fistr 用のメッシュに変更し、スケールを 1/1000 にして mm 単位に変更しておく。

4-2-4-2. 解析の種類と材料の設定

解析は、前項と同じ「非線形静解析」で設定する。

材料も前項と同じ「steel」として設定し、

材料モデル： PLASTIC

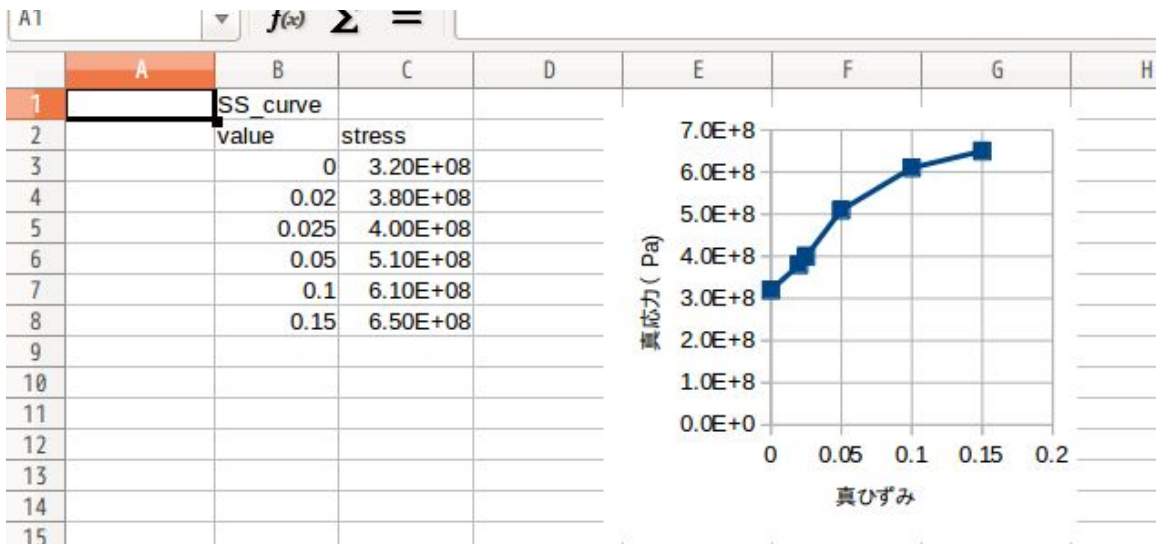
降伏条件： MISES

硬化則： MULTILINEAR

で設定する。



SS カーブ (SS_data) も同じデータを使う。下図参照。



4-2-4-3. 境界条件の設定

境界条件を設定する。解析は、block 上面を 10mm 下げ、その後、元の長さまで戻し、さらに 10mm 引き上げる解析を行う。

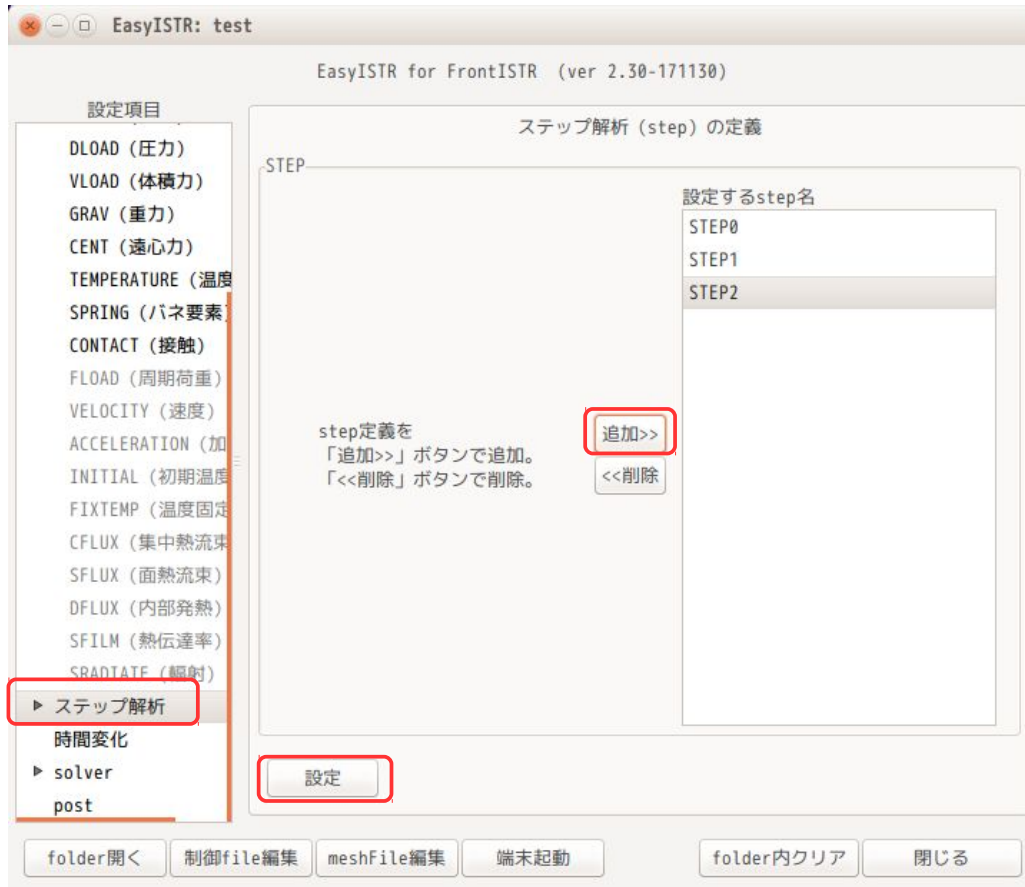
この為、まず、底面を固定し、上面を 10mm 縮める境界条件を設定する。下図参照。



4-2-4-4. step の設定

今回は、複数 step (3 step) 解析行うので、step を 3 ケ作成する必要がある。

この為、設定項目 Tree 内の「ステップ解析」を選択し、「追加」ボタンを 3 回クリックして、step を 3 ケ作成し、「設定」ボタンで確定する。



「設定」ボタンをクリックして、確定すると、設定項目 Tree 上に3ヶの step が表示される。
 この step の「STEP0」を選択して、SUBSTEPS 数を 10 回、前項で設定した BOUNDARY の内容を全て「設定する境界条件」側に移動、この後「設定」ボタンで確定させる。
 この設定で、最初の step (10mm 縮める) の設定ができた事になる。



4-2-4-5. step1 の設定

step1 (STEP1) の設定を行う。ここは、縮めた後、元の長さまで戻す設定になる。
step1 の境界条件は、設定項目 Tree 内の「境界条件 STEP1」内の項目を選択して、境界条件を設定する。
この step1 では、10mm 縮めた後元の寸法に戻す設定の為、以下の様に「0.010」を設定する。尚、底面の固定条件は、最初の step (STEP0) で設定した条件がそのまま使えるので、ここでは設定しない。



この後、ステップ解析の「STEP1」を選択し、「SUBSTEPS」数と「設定する境界条件」を設定する。
「設定する境界条件」は、
BOUNDARY, bottomN, STEP0 #最初の step0 で設定した境界条件 (底面固定)
BOUNDARY, topN, STEP1 #step1 で設定した境界条件 (上面を 10mm アップ)
を選択する。下図参照。



4-2-4-6. step2 の設定

step2 (STEP2) の設定を行う。step2 では、元の寸法に戻した後、さらに 10mm 伸ばす設定になる。

step2 の境界条件は、「境界条件 STEP2」内の BOUNDARY を設定する事になる。下図参照。



この後、「ステップ解析」を選択し、「SUBSTEPS」数と「設定する境界条件」を設定する。
「設定する境界条件」は、以下を選択する。

- BOUNDARY,bottomN,STEP0 #最初の step で設定した境界条件 (底面固定)
- BOUNDARY,topN, STEP2 #step2 で設定した境界条件 (上面をさらに 10mm アップ)

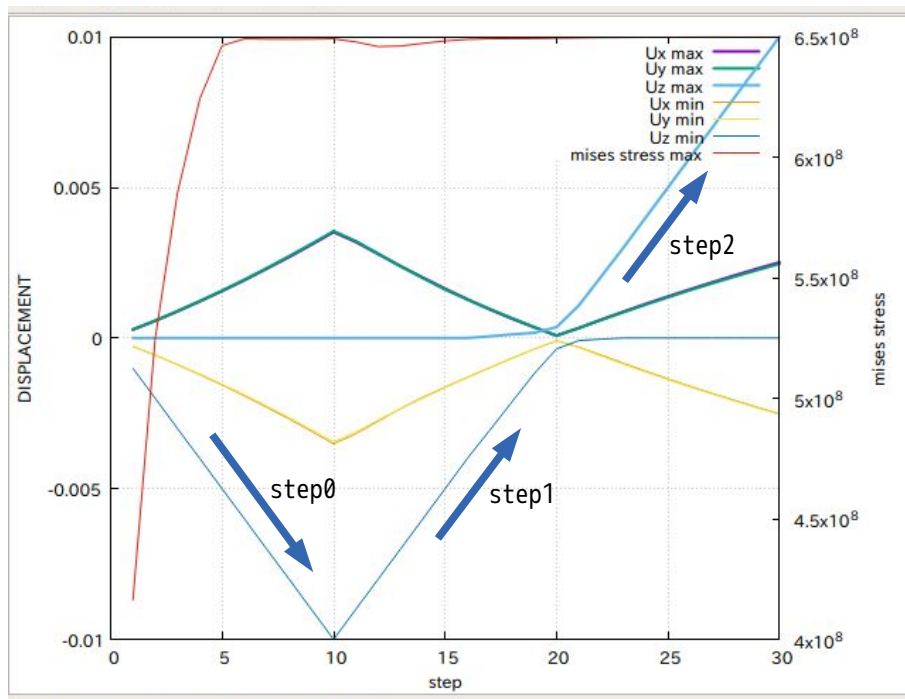


4-2-4-7. 計算開始、結果の確認

計算開始は、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、計算を開始させ、さらに「step 状況表示」ボタンをクリックして、plotStepMonitor を起動する。



plotStepMonitor に以下の状況が表示され、計算は、うまく走った事が確認できる。



今回の計算では、1回の計算で、上面を、-10mm (10mm縮める) → +10mm (元の寸法に戻す) → +10mm (さらに10mm伸ばす) に変化させた結果を取得することができる。変形形状や応力は、前項と同じ結果になるので、省略する。

4-3. 固有値解析

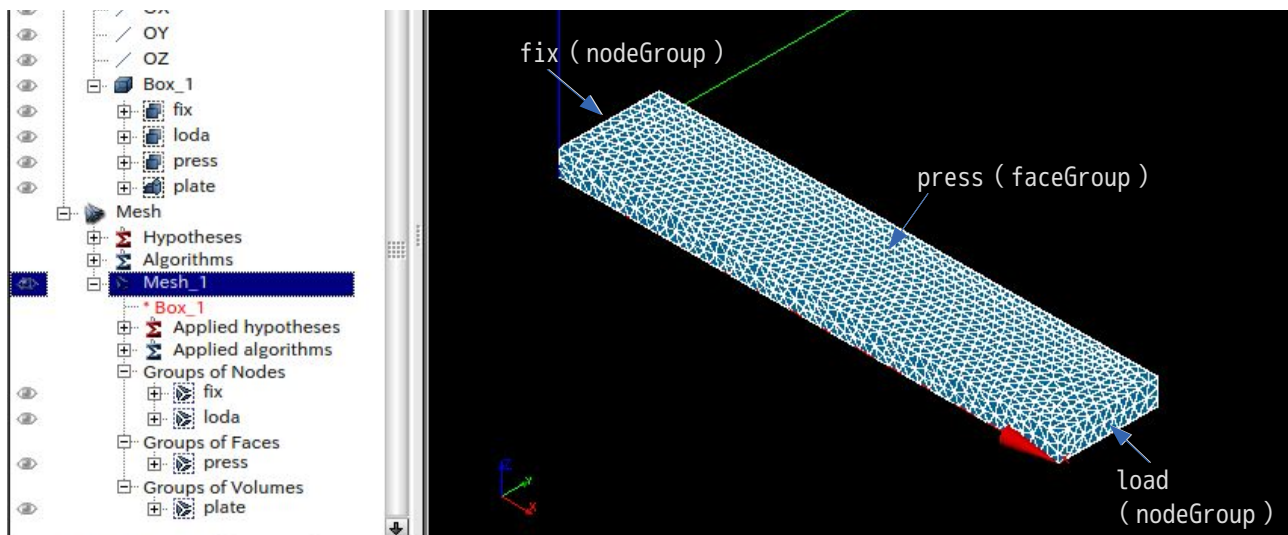
EasyISTR 上で固有値解析を行ってみる。

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)

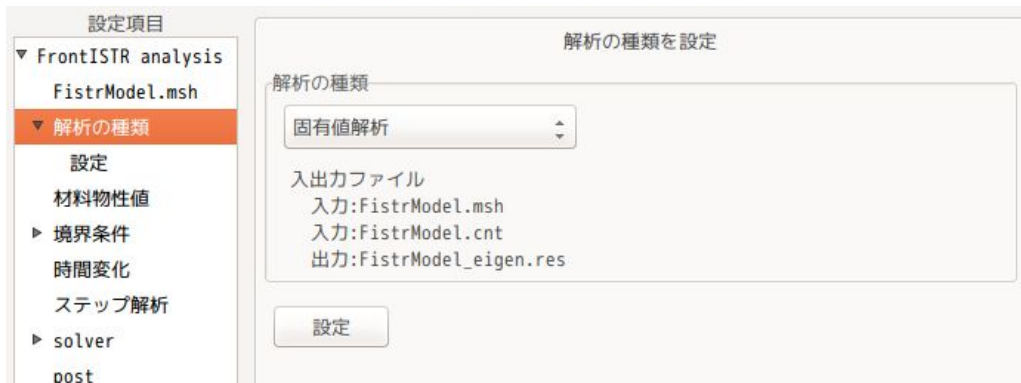
1. fix
2. load

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。

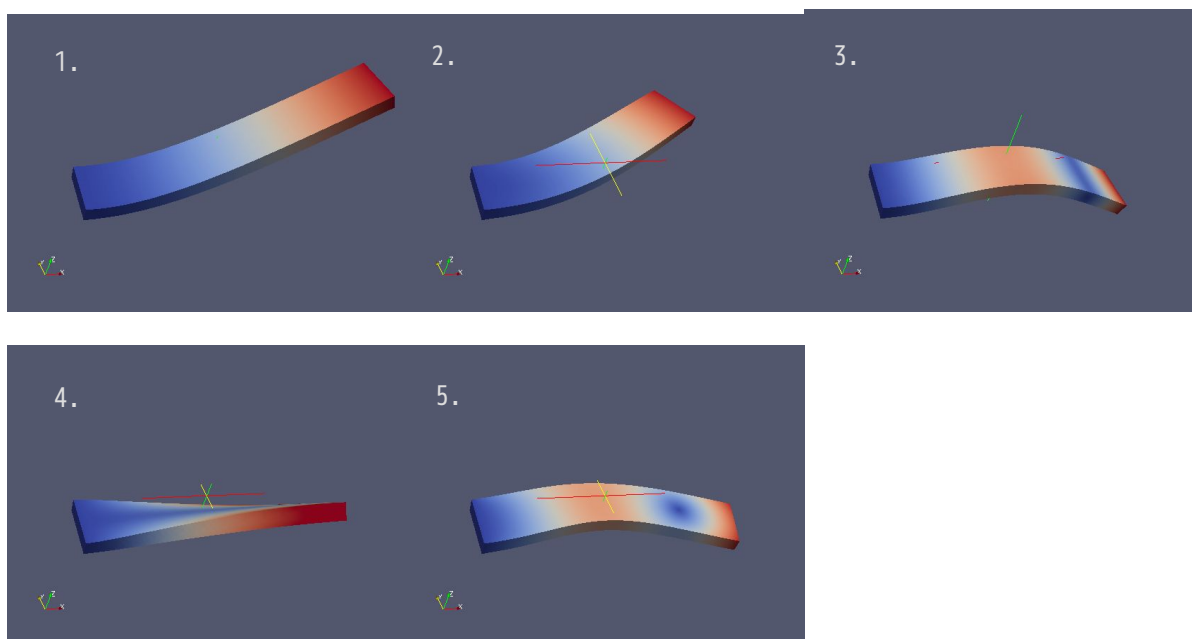


4-3-2-4. 計算開始、結果の確認

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

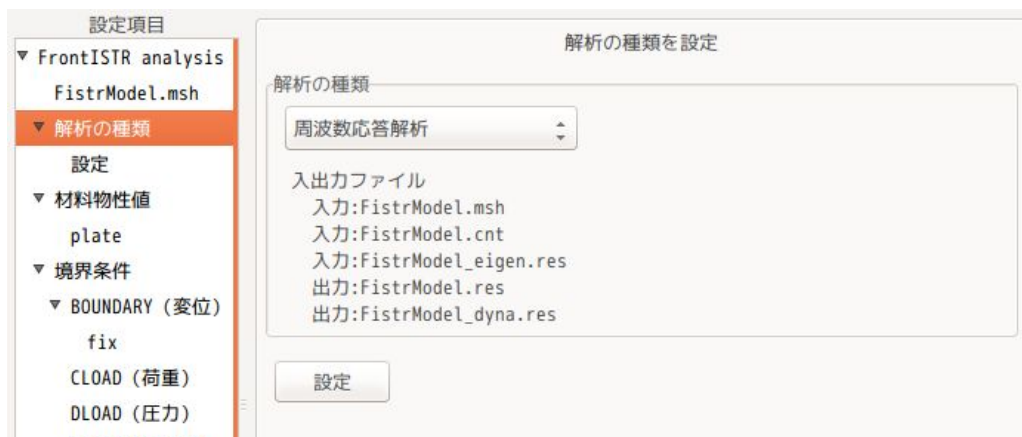
前項で使用した片持ち梁の固有値解析結果をコピーして workFolder としているので、モデル形状は、前項と同じ片持ち梁になる。

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。

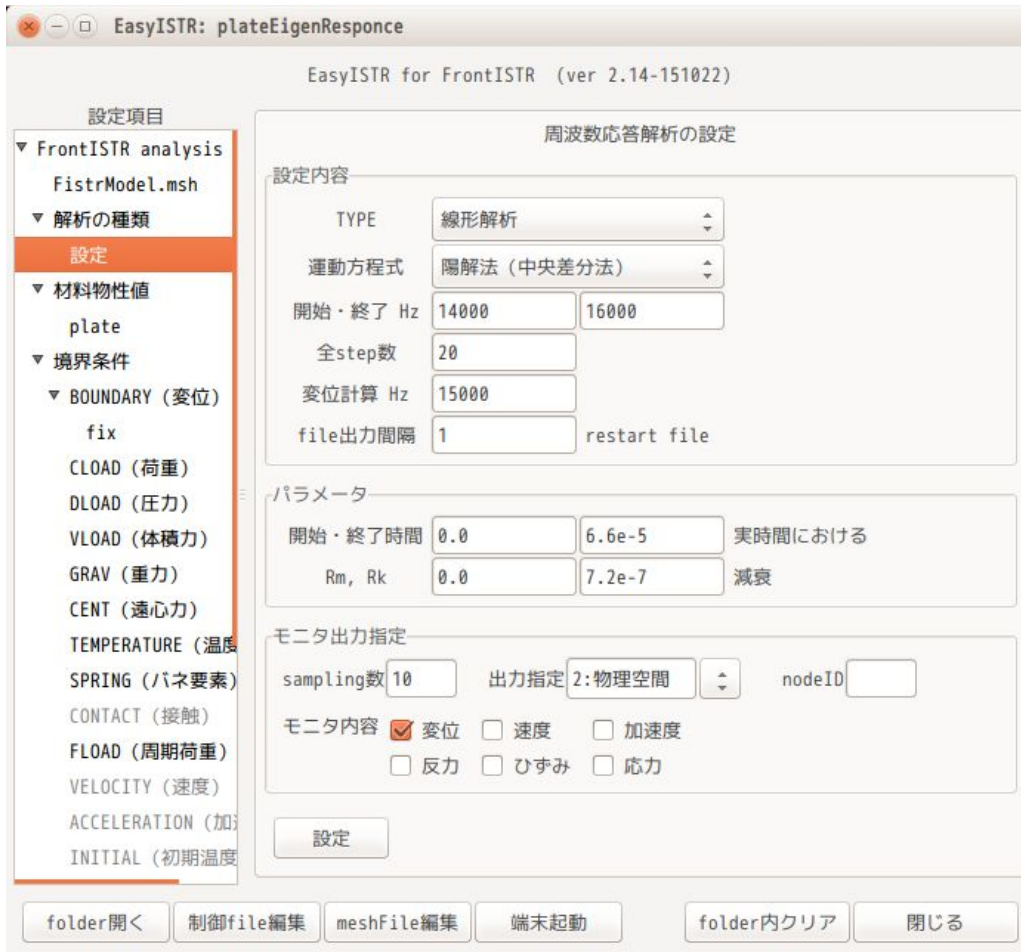
4-4-2-1. 解析の種類の設定

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。



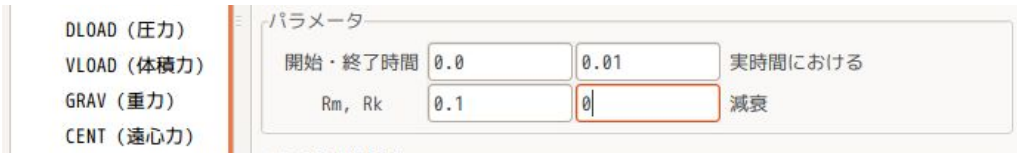
まず、TYPEと運動方程式は、そのまま「線形解析」、「陽解法（中央差分）」を選択。周波数は、前項の固有値解析の1次周波数が「536 Hz」の為、100～1000 Hzまで求めてみる。周波数の間隔は、10Hz間隔とする。（100～1000までを90分割する事になる為、「全step数:90」を設定する。）下図の様に設定。

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。

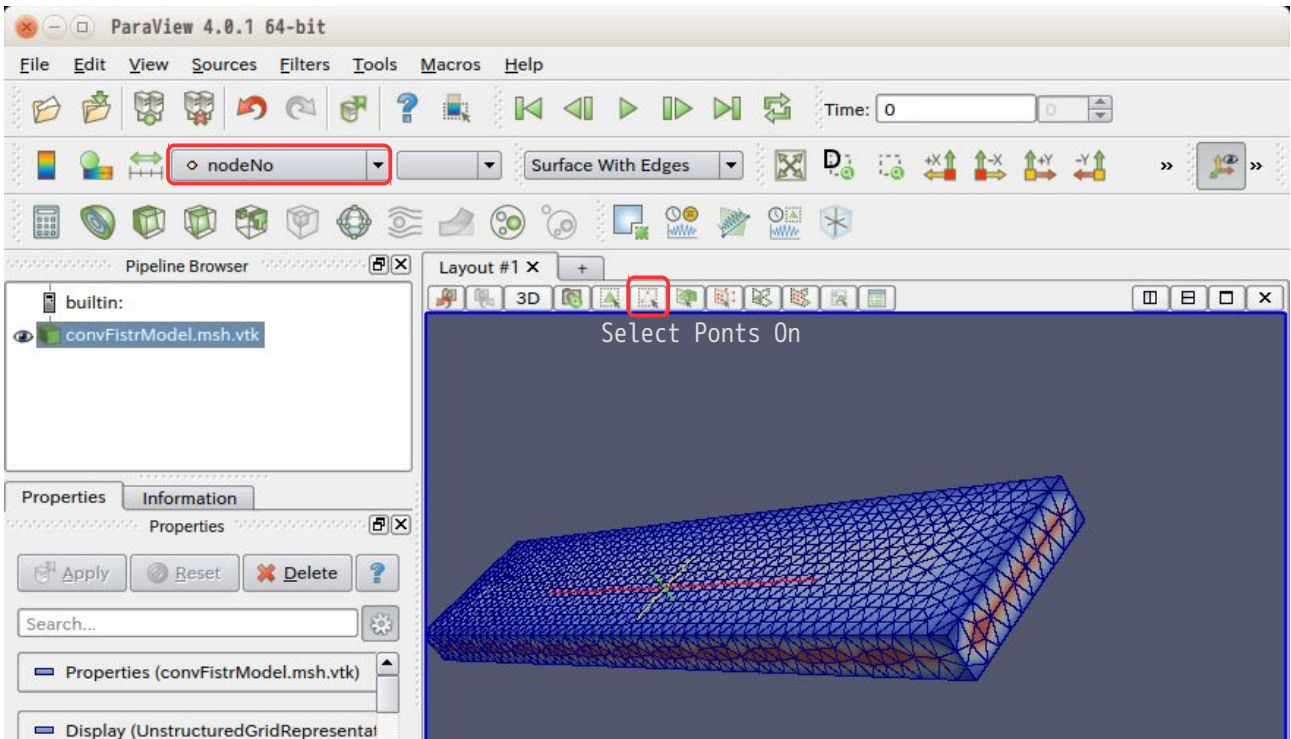


減衰は、質量マトリックスにかかる減衰（速度にかかる減衰）0.1を設定する。下図のように設定。

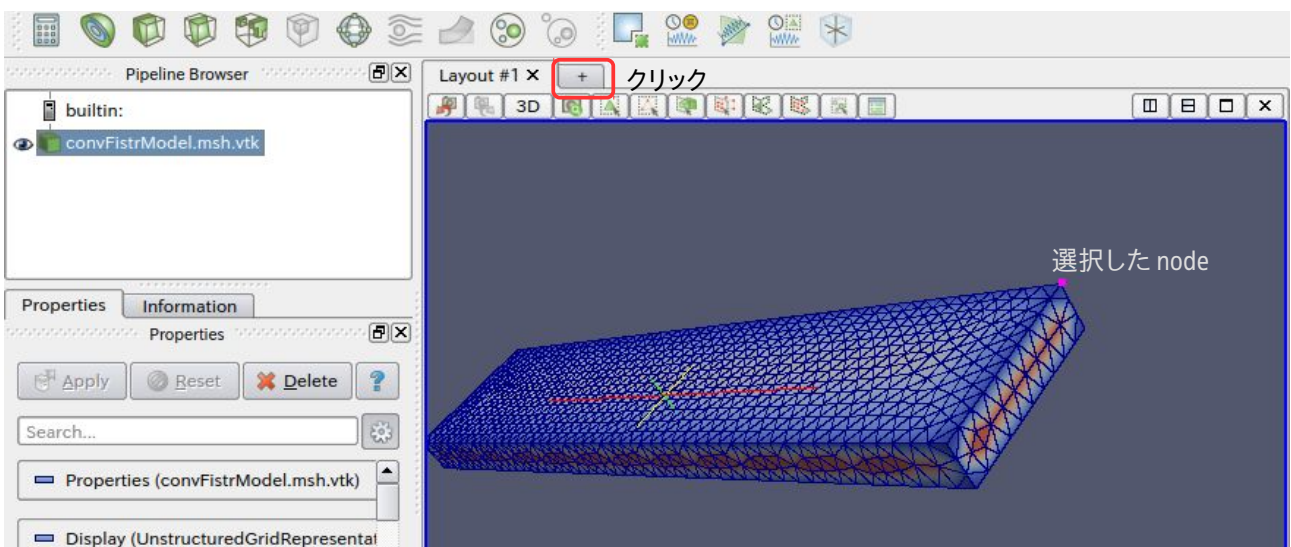
開始、終了時間は、周波数応答には関係ない。この時間は、上記の変位計算 Hz で指定した周波数で周期荷重を加えて過渡応答を求める時の開始、終了時間になる。



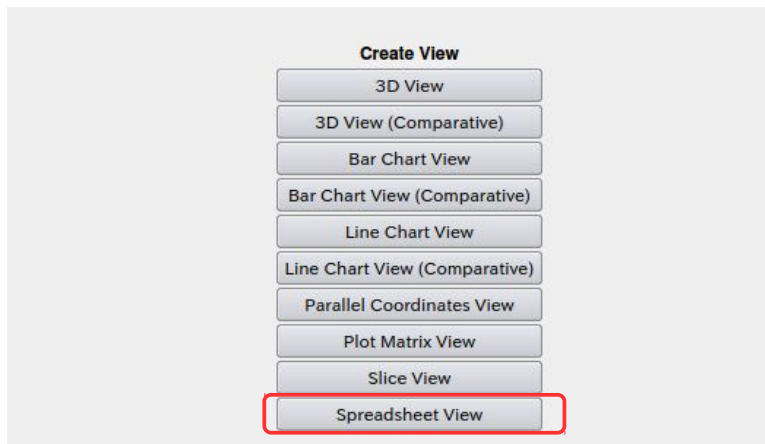
モニタする部位は、片持ち梁先端とする。この先端の nodeNo を調べる必要があるので、この nodeNo を paraView を使って調べる。下図は、nodeNo を表示させた状態。ここで「Select Points On」ボタンをクリックして、マウスをドラッグさせ、node を選択する。



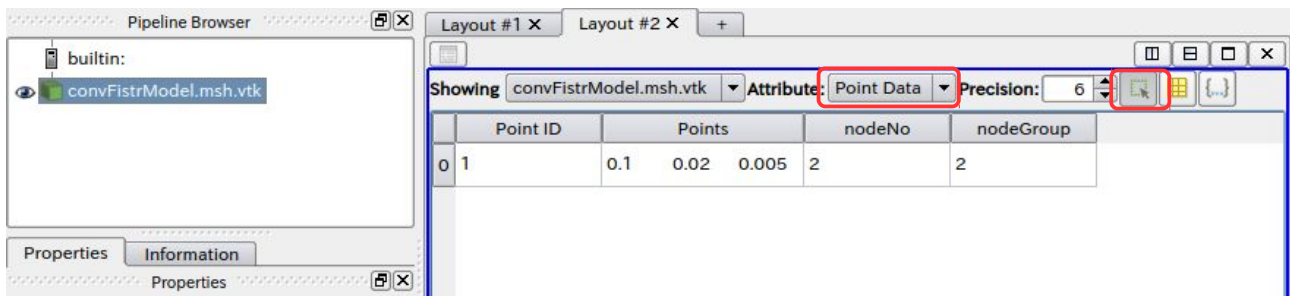
下図が node を選択した状態。ピンク色の点を選択した node になる。この状態で「+」タグをクリックして、新しい画面を表示させる。



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 数分計算する。



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

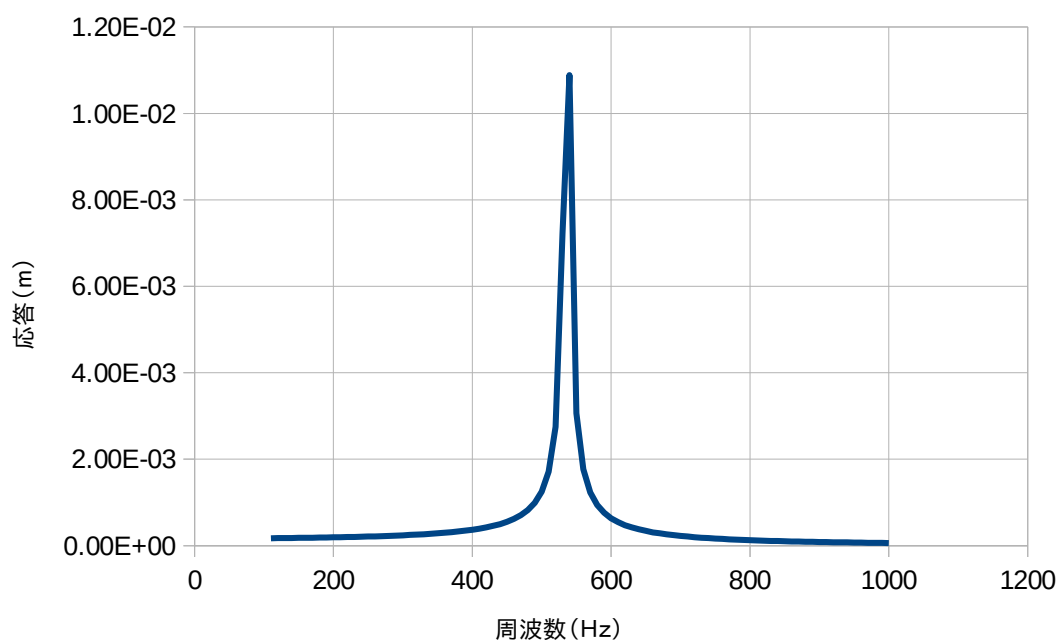
y

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 (陽解法)

EasyISTR で陽解法による時刻歴応答解析を行ってみる。陽解法は、収束計算を行わないので、計算は早い
が、時間増分 (Δt) を小さくしてやる必要がある。この為、衝撃等の過渡応答解析には適するが、ゆっく
り動かす様な解析では、次項の陰解法を使って解析する。

モデル形状は、固有値解析や周波数応答で使用し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用し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

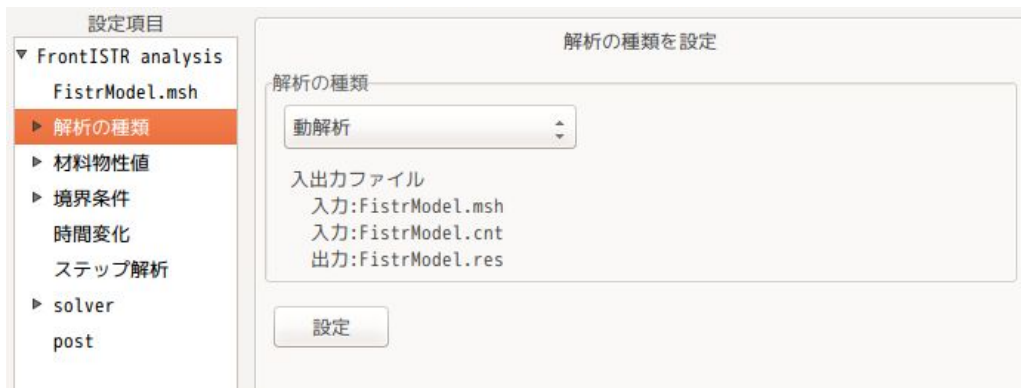
周波数応答解析の結果ファイル

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。



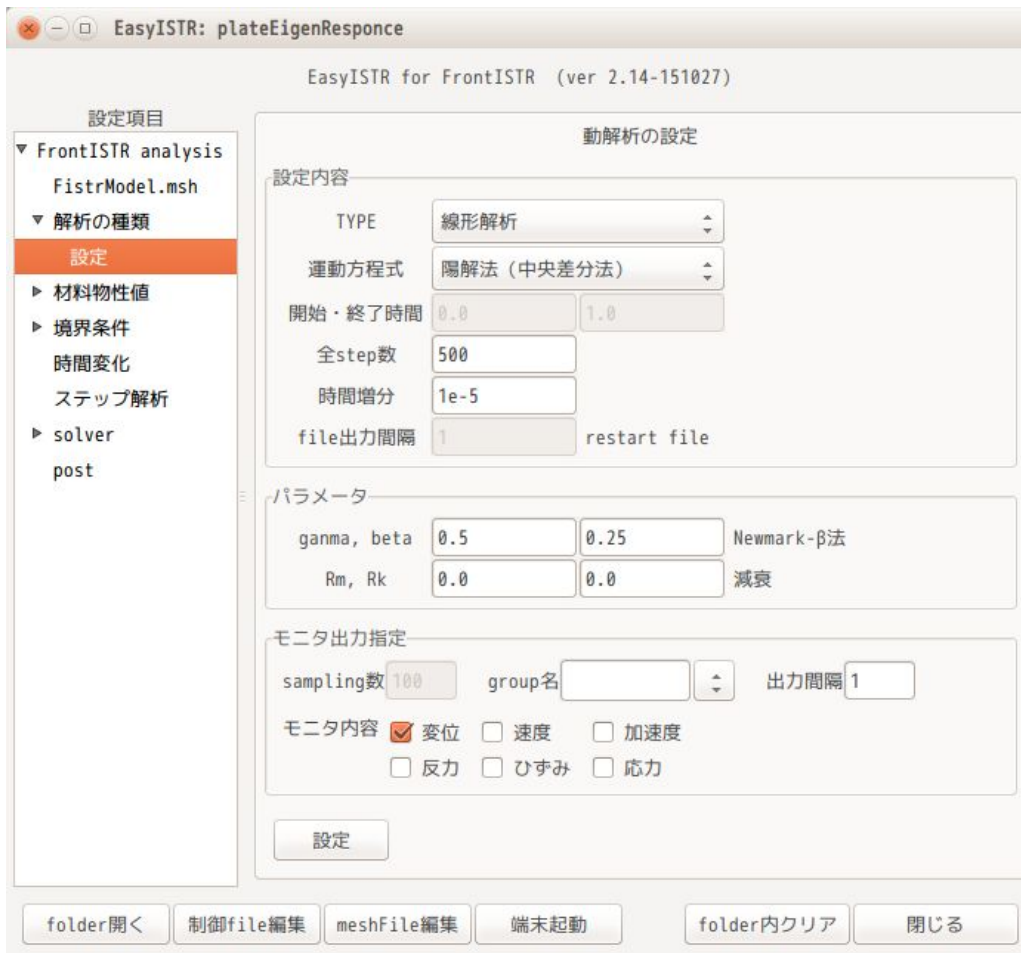
4-5-2-1. 解析の種類の設定

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 (Δt) 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。

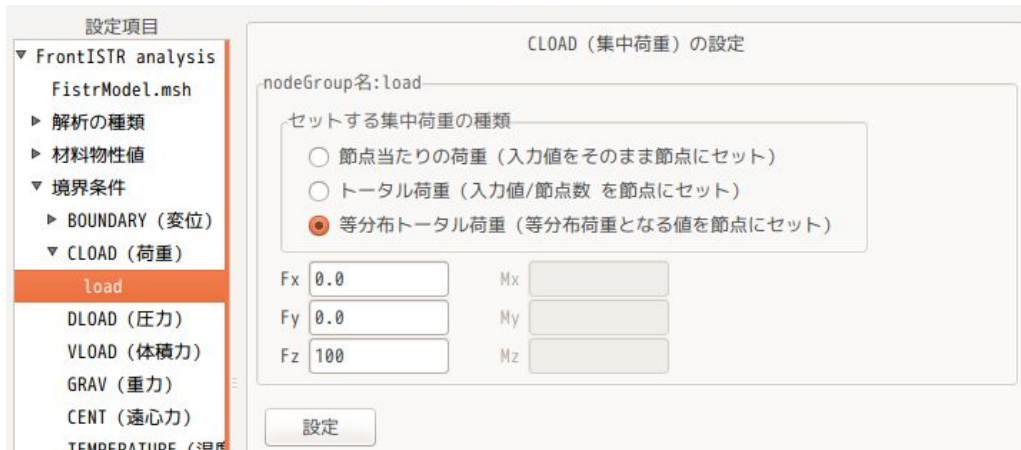
パラメータ内の gamma、beta は、運動方程式を陽解法に設定しているため、関係なくなり、このままの設定。
 減衰は、質量マトリックスにかかる減衰 Rm を「0.1」に設定した。

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 方向等分布荷重で合計「100.0」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6msまで計算してみる。1次の固有振動数が約540HZの為、6msで1次周波数の3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「60000」ステップになる。下図参照。



モニタの出力間隔は、「100」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。



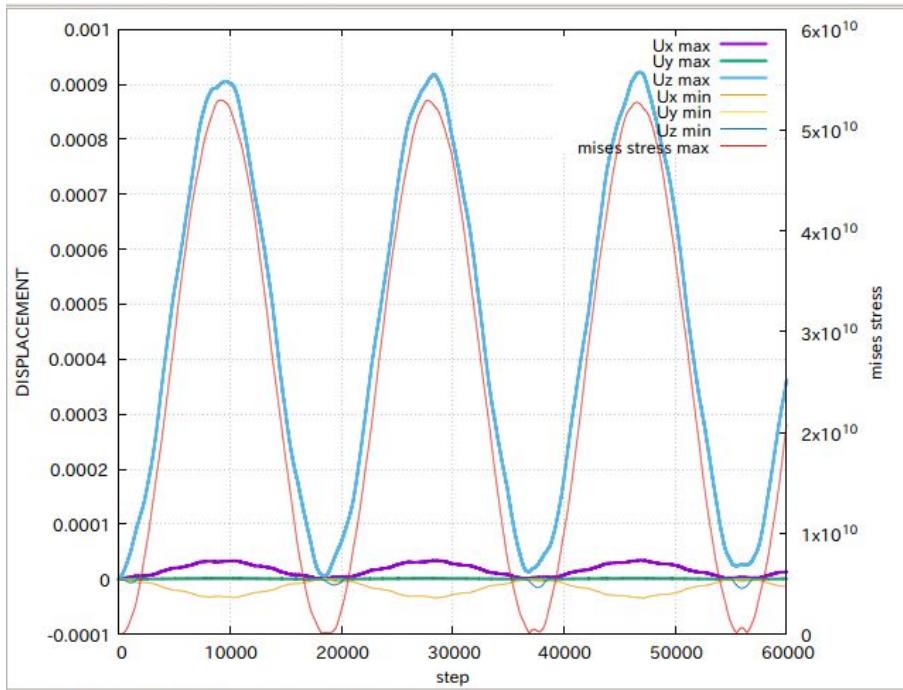
4-5-2-6. 計算開始、結果の確認

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、60000 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 2000 回に設定しておく。(結果ファイルが 30 ケでかきあがる設定。)



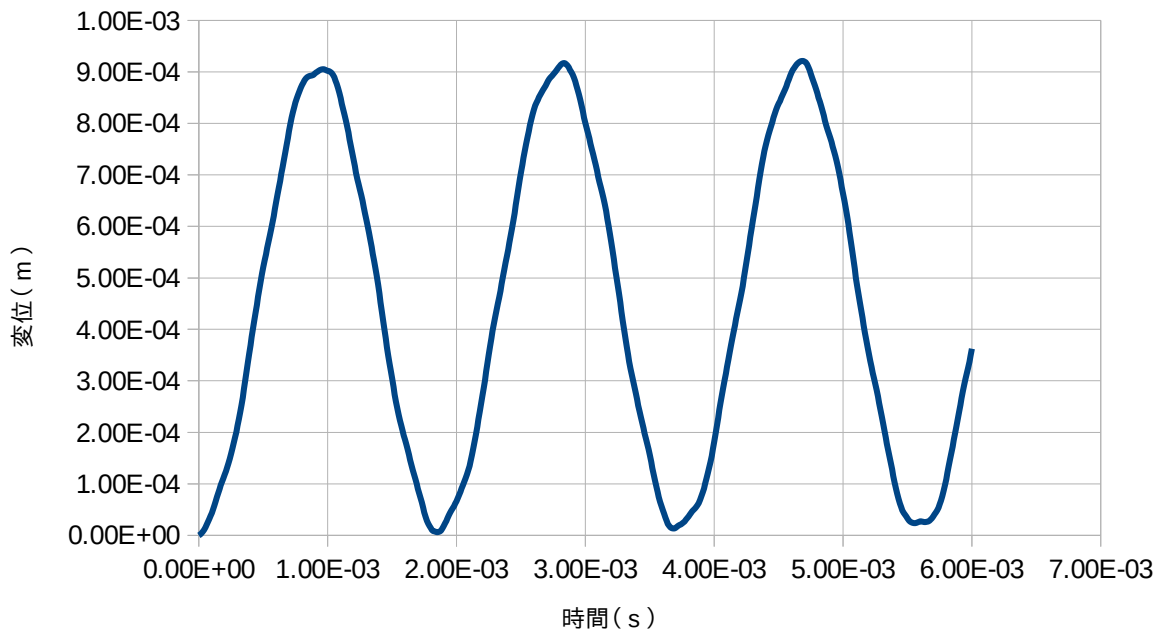
「FrontISTR 実行」ボタンをクリックして、計算を開始させる。その後、「step 状況表示」ボタンをクリックして、plotStepMonitor を起動し、step の計算状況を表示させる。



この状況から、最大変位が「0.0009」であることがわかる。

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。

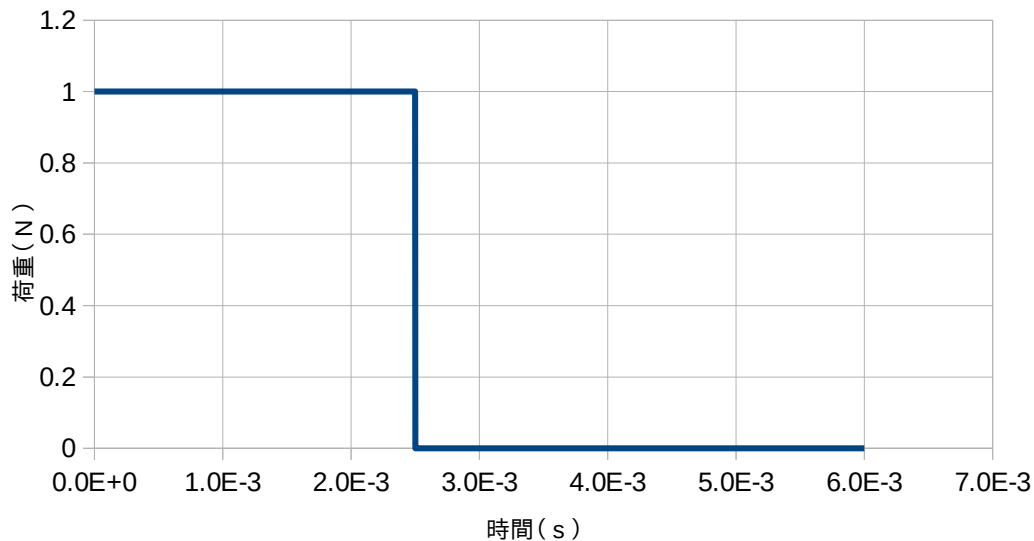


4-5-3. 境界条件に時間変化を与える（線形・陽解法）

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

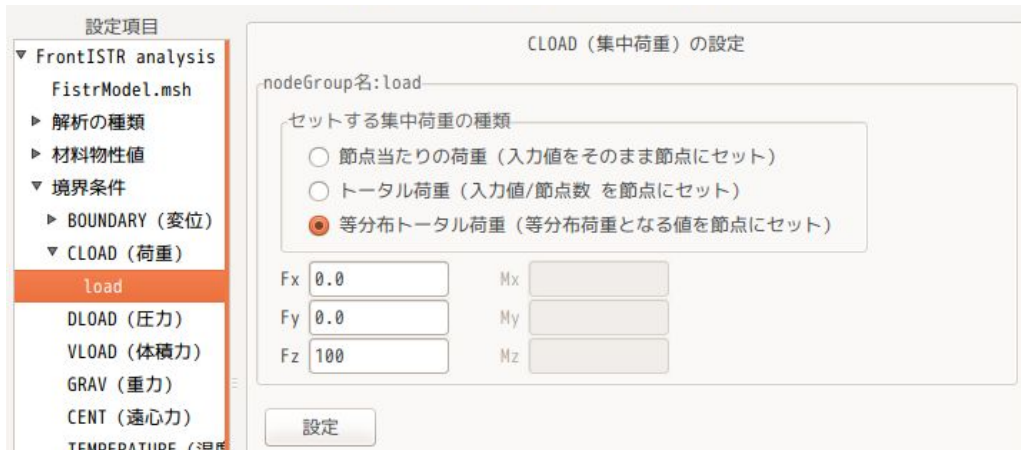
片持ち梁の先端（fix）のZ方向の値に時間変化を与える。以下の変化を与える解析を行う。



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方向に合計で100Nの等分布荷重をかけている。これは、変更せずこのまま。（下図参照。）

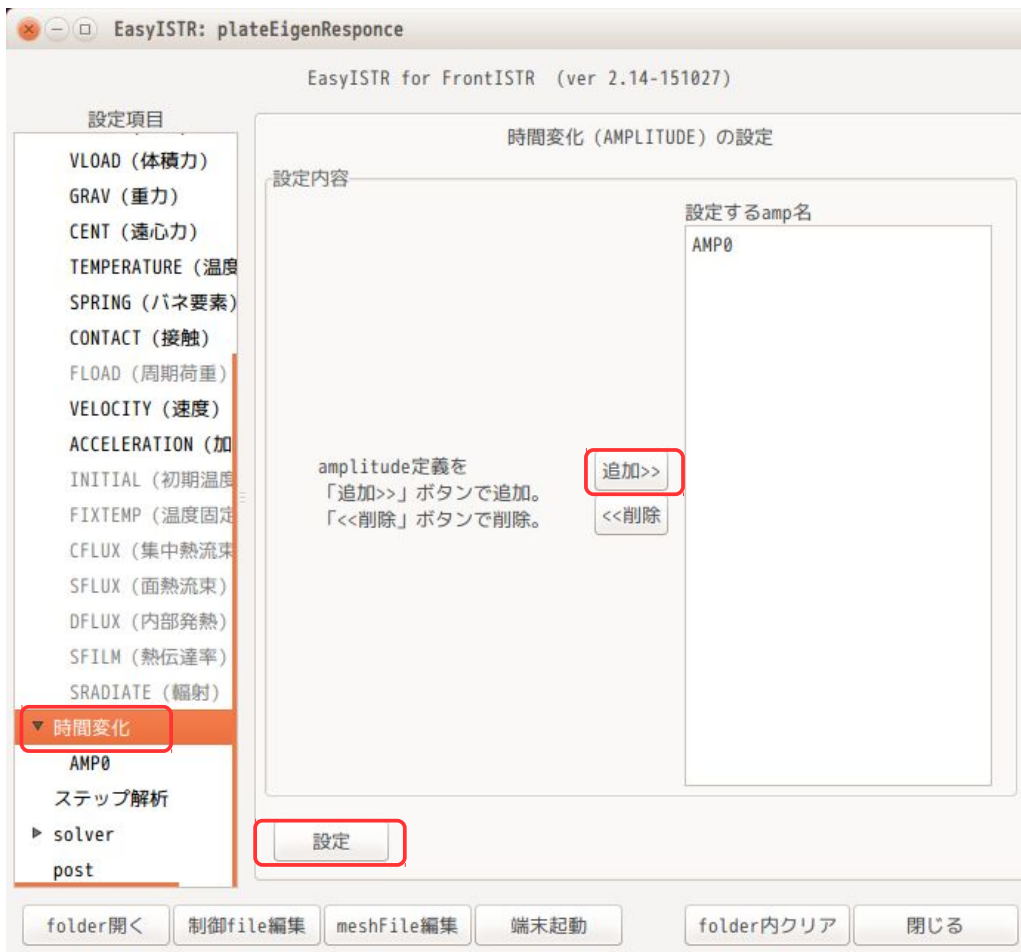




4-5-3-3. 境界条件に時間変化を与える

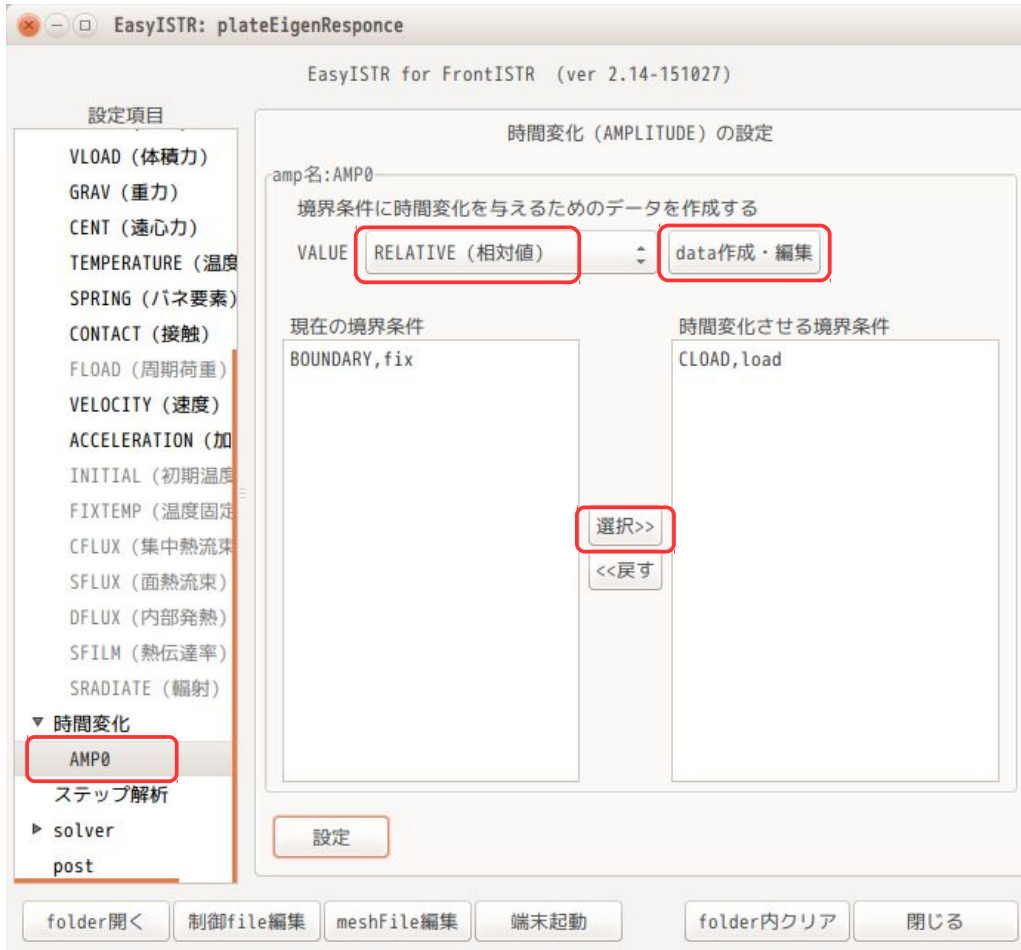
片持ち梁の先端 (load) に設定した境界条件 (Z方向に1.0Nの節点荷重) に、時間変化を与える。

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。

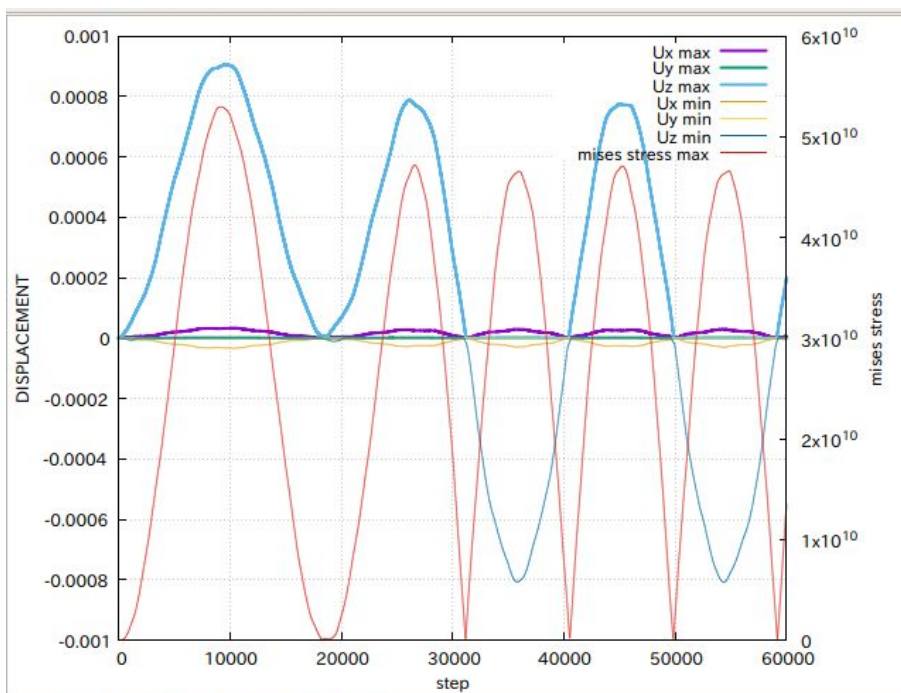
	A	B	C	D	E	F	G	H	I
1		ampData							
2		time	value						
3		0	1						
4		2.50E-03	1						
5		2.51E-03	0						
6		6.00E-03	0						
7									
8									

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書き込まれる。）

4-5-3-4. 計算開始、結果の確認

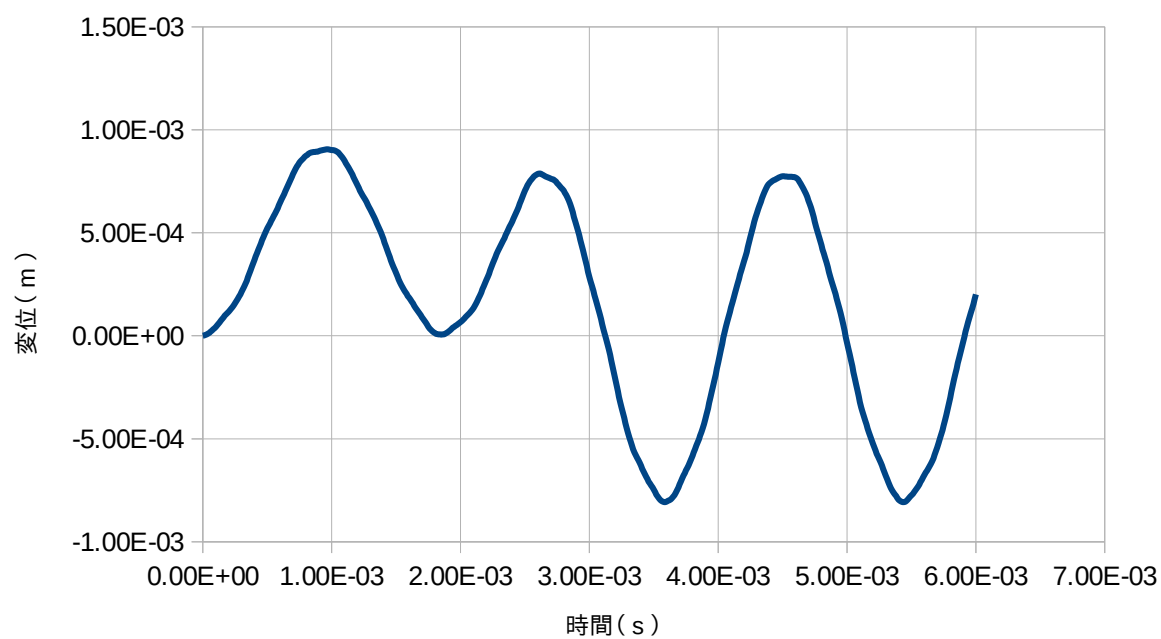
計算開始前に、folder 内の結果 file や並列用の分割メッシュを全て削除し、再度並列用にメッシュを分割し、4 並列で、計算を開始した。計算開始後、「step 状況表示」ボタンで plotStepMonitor を起動しておく事で、逐次現在の計算状況がグラフ表示される。



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、変位のグラフになるが、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. 時刻歴応答解析 (陰解法)

陰解法を使った時刻歴応答解析を行ってみる。陰解法は、毎回收束計算を行うので、陽解法に比べて時間増分 (Δt) を大きくできる利点がある。この為、モデルをゆっくり動かす場合には、陰解法を使う。ここでは、前項と同じモデル、条件設定で、陰解法による解析を行ってみる。

4-6-1. モデルと材料

モデルは、前項と同じ「plate.unv」を使うので、このファイルを読み込み、メッシュ変換後、スケールを 1/1000 にしておく。材料も前項と同じ「steel」を設定する。

4-6-2. 境界条件一定値 (線形・陰解法)

まずは、境界条件を変化せずに、一定値のままで計算する。

4-6-2-1. 解析の種類の設定

選択項目 Tree 内の「解析の種類」を選択して「動解析」を選択し、この内容は、以下で設定した。

運動方程式は、「陰解法」を選択。

全 step 数、時間増分は、前項より一桁遅く設定。

粘性係数 Rm を「0.1」に設定。

group 名を「2」に設定。

この条件では、時間増分が大きく、前項の陽解法では計算できない設定になっている。

4-6-2-2. 境界条件の設定

境界条件は、前項と同じ設定とする。(下図参照)

The screenshot displays the software's configuration window. On the left, a sidebar shows a tree view with 'BOUNDARY (変位)' expanded to 'fix'. The main panel is divided into two sections:

- group名: fix**: This section is for fixed boundary conditions. It has two columns: '変位' (Displacement) and '回転角' (Rotation Angle). Under '変位', the x, y, and z axes are all checked with checkboxes, and each has a text input field containing '0.0'. Under '回転角', the Rx, Ry, and Rz axes are unchecked, and each has an empty text input field. A '設定' (Settings) button is located below these fields.
- nodeGroup名: load**: This section is for load conditions. It starts with a heading 'セットする集中荷重の種類' (Type of concentrated load to set) and three radio button options:
 - 節点当たりの荷重 (入力値をそのまま節点にセット) (Load per node (set input value as is to node))
 - トータル荷重 (入力値/節点数 を節点にセット) (Total load (set input value/node number to node))
 - 等分布トータル荷重 (等分布荷重となる値を節点にセット) (Uniform total load (set value that becomes uniform load to node))
 Below these options are two columns of force/moment input fields: Fx, Fy, Fz and Mx, My, Mz. Fx, Fy, and Mx are set to '0.0', while Fz is set to '100'. The other fields (Ry, Rz, My, Mz) are empty. A '設定' (Settings) button is at the bottom.

4-6-2-3. 計算開始、結果の確認

設定項目Tree内の「solver」を選択して、solverの設定画面を評させる。

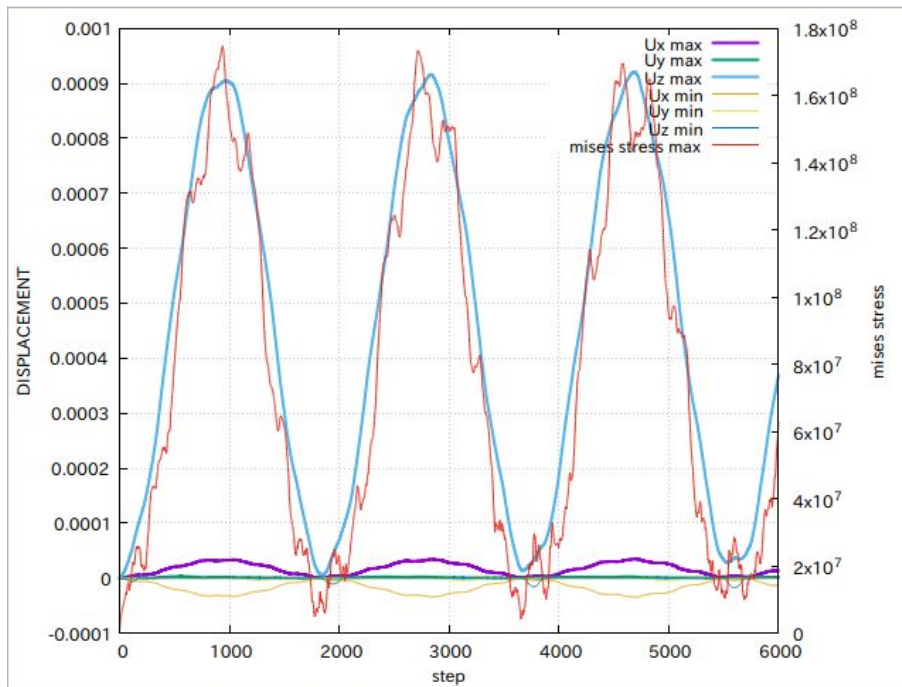
計算は、6000step計算させるので、結果の出力頻度は、「200」に設定する。(計算結果が30ヶ出来上がる設定)

この後、「FrontISTR実行」ボタンをクリックし、実行する。

計算開始後、「step状況確認」ボタンをクリックして、plotStepMonitorを起動させ、step状況を逐次グラフ表示させる。



下図が計算状況の結果になる。計算はうまく走っている。



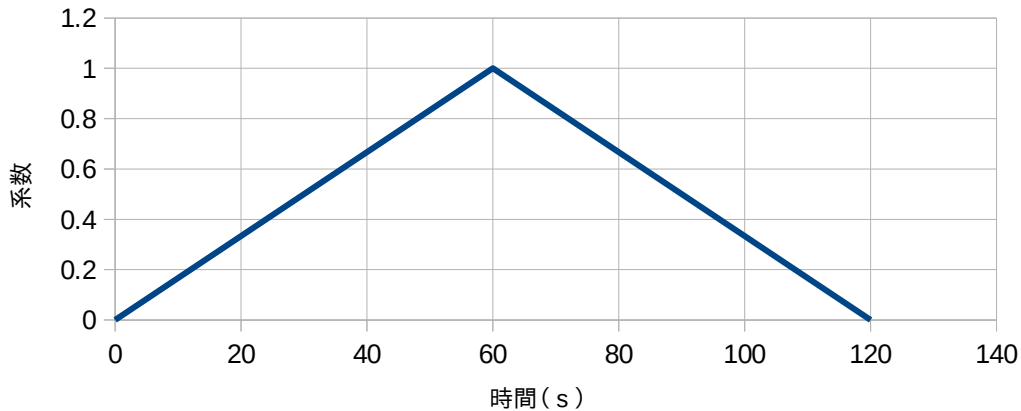
今回の陰解法は、前項の陽解法の条件に比べて、時間増分を「 $1e-7$ 」→「 $1e-6$ 」に変更して、陽解法では計算できない条件で計算させた。計算の step 数が減っているので、トータル的に計算時間は、早くなっている。

4-6-3. 境界条件に時間変化を与える（線形・陰解法）

ここでは、設定した境界条件の荷重に時間変化を与えて計算させてみる。この時間変化は、陰解法で計算させるので、ゆっくりとした時間変化を与えることにする。

4-6-3-1. 時間変化の内容

時間変化は、下図の様に、分単位のゆっくりとした時間変化を与えることにする。



このような時間変化を与える為、計算のステップを再設定する。以下の様に「全 step 数」と時間増分を再設定する。(1s 間隔で 120step まで計算する設定)

設定項目

- FrontISTR analysis
- FistrModel.msh
- 解析の種類
- 設定
- 材料物性値
- 境界条件
- ステップ解析
- 時間変化
- AMP0
- solver
- post

動解析の設定

設定内容

TYPE	線形解析
運動方程式	陰解法 (Newmark-β法)
開始・終了時間	0.0 1.0
全step数	120
時間増分	1.0
file出力間隔	1 restart file

パラメータ

gamma, beta	0.5 0.25	Newmark-β法
Rm, Rk	0.1 0.0	減衰

モニタ出力指定

sampling数	100	group名	2	出力間隔	1
-----------	-----	--------	---	------	---

モニタ内容

<input checked="" type="checkbox"/> 変位	<input type="checkbox"/> 速度	<input type="checkbox"/> 加速度
<input type="checkbox"/> 反力	<input type="checkbox"/> ひずみ	<input type="checkbox"/> 応力

4-6-3-2. 境界条件の設定

境界条件の設定は、前項と同じ条件とする。



4-6-3-3. 境界条件に時間変化を与える

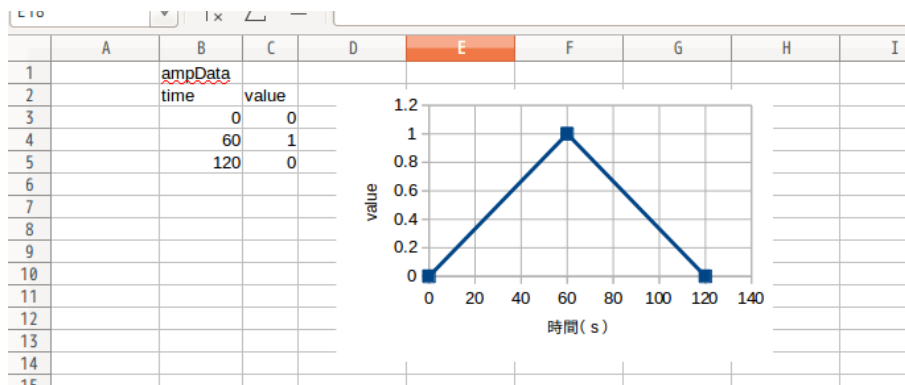
まず、時間変化を与えるデータファイルを作成する。
設定項目 Tree 内の「時間変化」を選択し、「追加」ボタンをクリック、「設定」ボタンをクリックして、「AMP0」を作成する。



この後、設定項目 Tree 内の「AMP0」をクリックして、時間変化させる境界条件「CLOAD,load,STEP0」を「時間変化させる境界条件」側に移動させる。
この後、「data 作成・編集」ボタンをクリックして、時間変化させるデータを作成する。



「data作成・編集」ボタンをクリックすると、office が起動するので、時間変化のデータを作成して、そのまま csv 形式で上書き保存する。（下図参照）



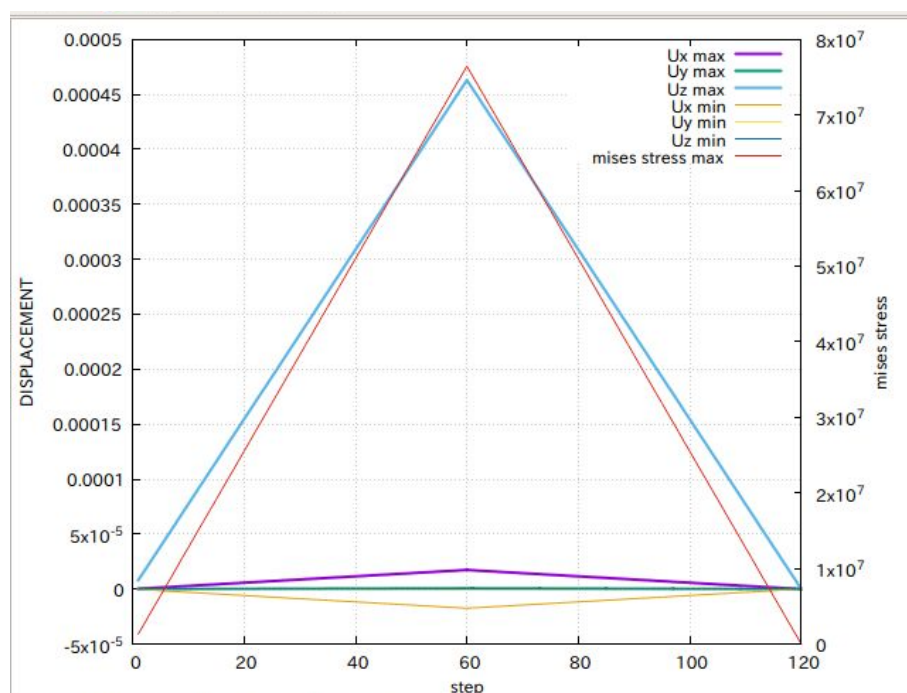
これで時間変化させる準備が整った事になる。

4-6-3-4. 計算開始、結果の確認

計算を開始させるが、全 120 ステップあるので、結果の出力頻度は「10」として、10s 間隔で保存する。（下図参照。）



この設定で、「FrontISTR 実行」ボタンをクリックして計算を開始する。計算開始後、「step 状況表示」ボタンをクリックして、計算状況を表示させる。以下がその結果になる。



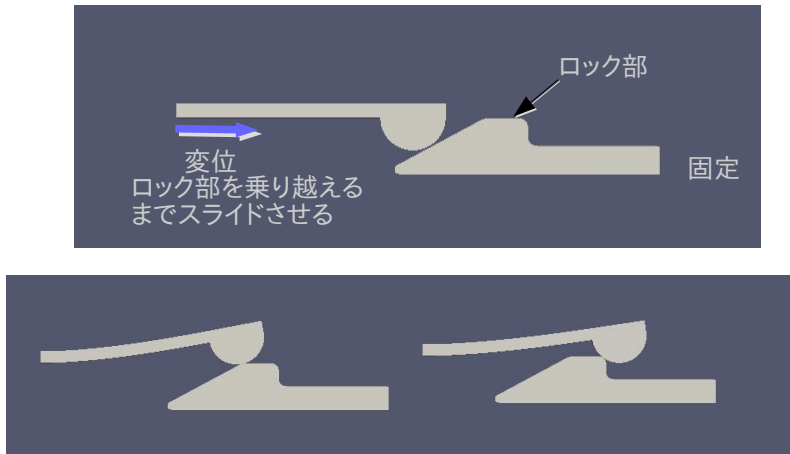
計算は、うまく走っている。ゆっくり荷重を変化させているので、細かな変動は存在していない。

4-7. 接触を含む時刻歴応答解析

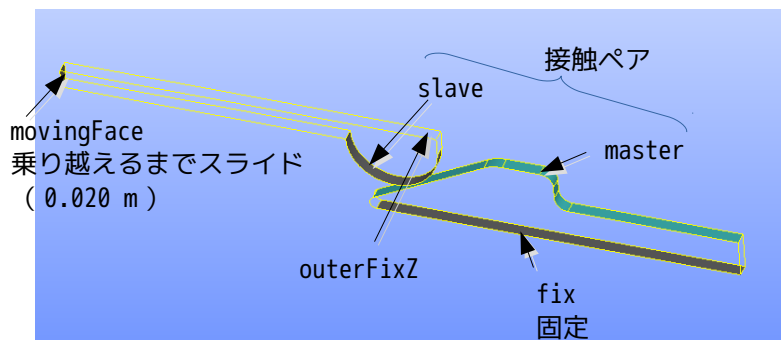
前項までの時刻歴応答解析は、線形の動解析であるが、ここでは、接触（非線形）の動解析を行ってみる。

4-7-1. モデル形状

モデルは、スナップフィット (snapFit2.unv) を使って、接触を含む時刻歴応答解析を行う。



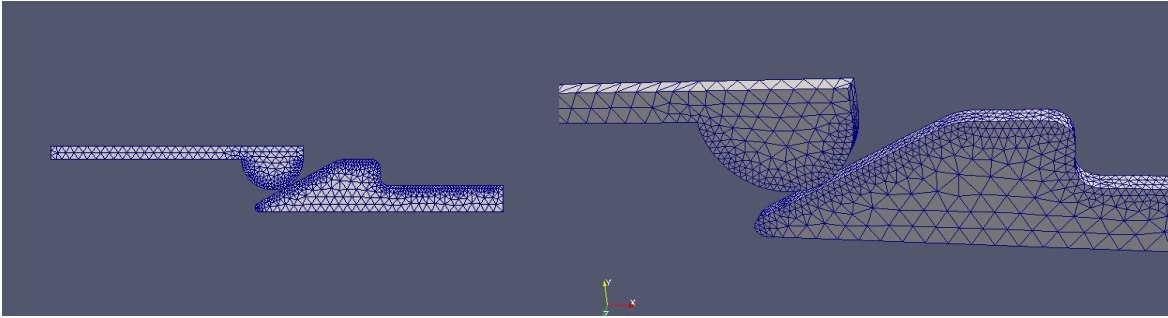
このモデルの group 化は、以下で実施している。



group 名	区分	
outer	要素	可動側
inner	要素	固定側
MST_outer	面	接触面 (master 側)
movingFace	節点	この面を 20mm 移動させる。
fix	節点	固定部
SLV_inner	節点	接触面 (slave 側)
outerFixZ	節点	Z 方向の固定部 (可動側先端の edge 部)

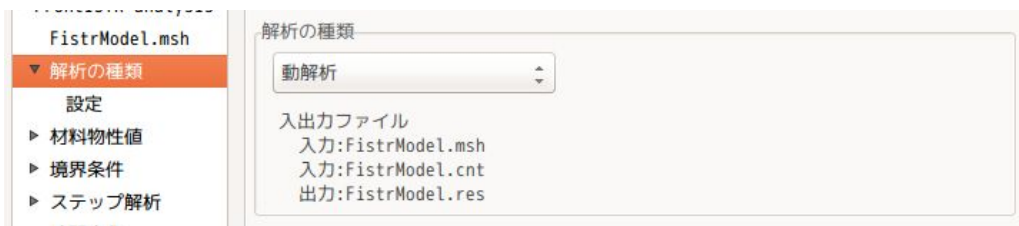
この形状のメッシュを下図の様に作成し、「snapFit2.unv」として保存している。接触面のメッシュは、細かくしている。

解析は、movingFace 面を 20mm/1s の速度で移動させ、計算は、1s まで計算させてみる。



4-7-2. 設定内容

前項のメッシュを読み込み、倍率を 1/1000 としておく。
この後、解析は、「動解析」に設定する。



動解析の設定は、「非線形」の「陰解法」とし、計算は、0.005s 間隔で 200step (1s) まで計算させる。
また、減衰は質量マトリックスに「0.1」を設定し、モニタ出力は今回設定しないので、group 名に「1」を設定している。



材料の設定は、両材料とも「Aluminum」とした。



接触の設定は、以下で設定。(摩擦係数以外は、default のままの設定。)



境界条件については、

- fix xyz 各方向固定
- movingFace yz 方向固定
- outerFixZ z 方向固定

に設定する。

movingFaceは、20mm/1s の速度で移動させるが、移動させる方法として、

速度を設定する (20mm/1s = 0.02 m/s)

20mm の変位を設定し、その値に時間変化を与える

方法がありどの方法でも計算できるが、速度を与えた方が手間が省け、収束性が若干良いので、速度を設定することにする。以下の様に設定した。



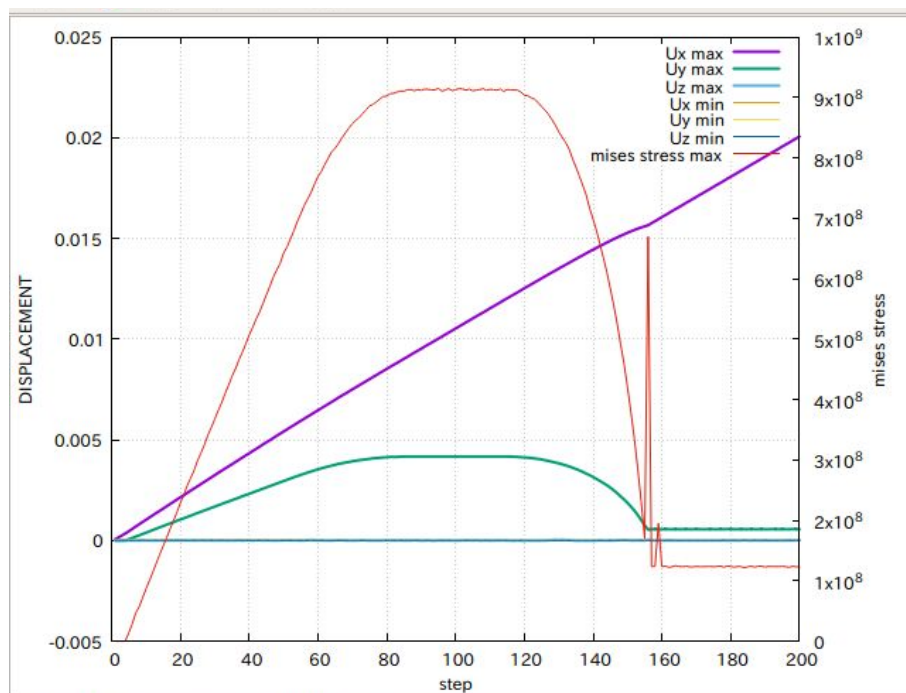


今回は、接触（非線形）を含む為、ステップ解析の設定が必要になる。
この設定は、DTIME（時間増分）、ETIME（終了時間）を設定したのみになる。（これ以外は、defaultの設定。）



4-7-3. 計算開始、結果の確認

全ての設定が完了したので計算を開始させる。
線形 solver は、「CG」を選択し、結果出力頻度は、「10」に設定し、10step 毎に 200step まで、計 20 ヶの結果ファイルが出来上がる設定で計算を開始した。
以下が、plotStepMonitor の出力結果になる。



計算は、うまく走っている。動解析を行っている為、ロック部を乗り越えた後に、衝突によると思われる mises 応力のピークが発生している。

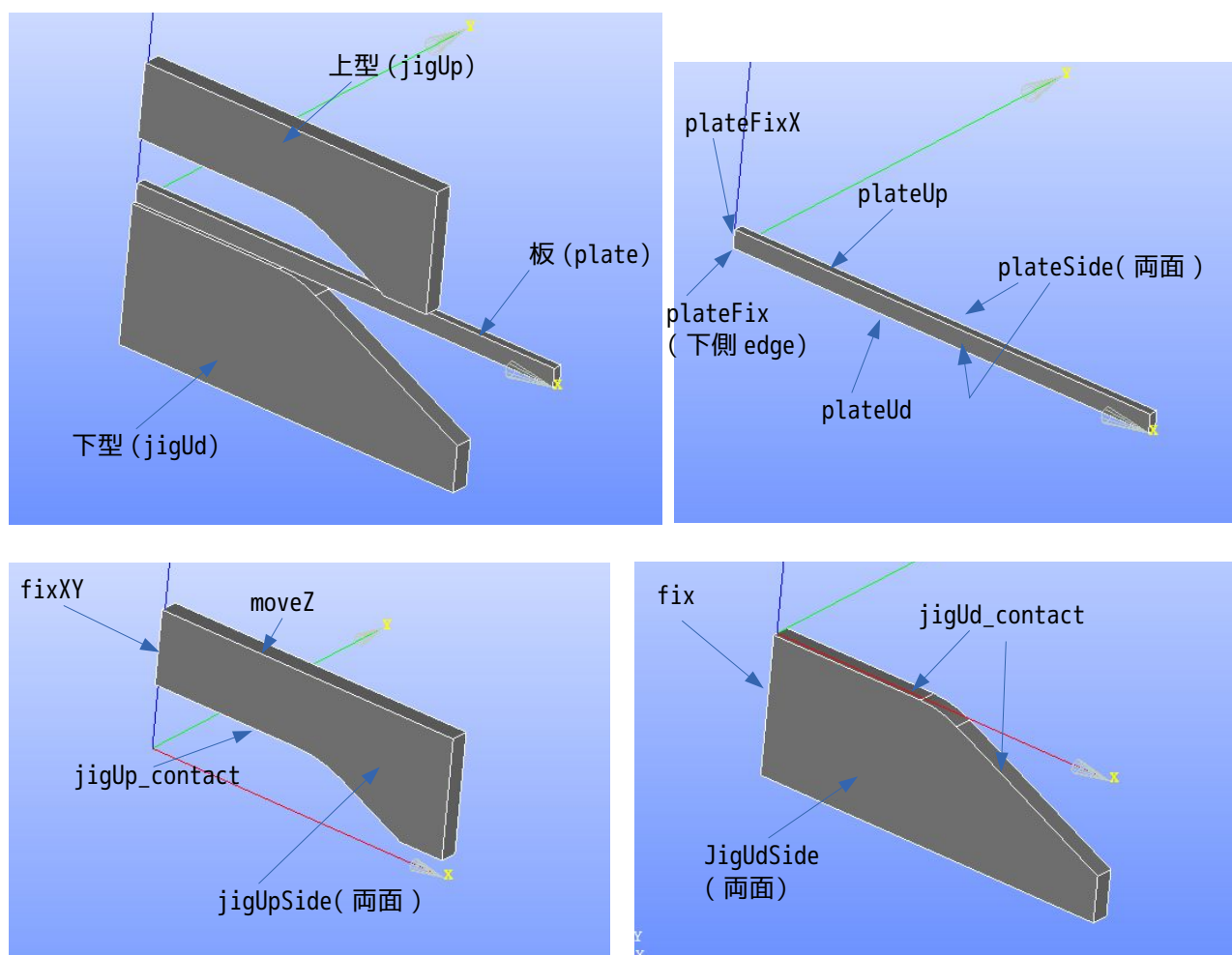
4-8. 接触を含む弾塑性解析 (板のプレス曲げ加工)

接触解析、弾塑性解析は、単独で解析できているので、ここで接触と弾塑性を含む解析 (板のプレス曲げ加工) を行ってみる。
 この解析は、接触、弾塑性、大変形の非線形解析を行うことになるので、計算負荷は大きくなる。

4-8-1. モデル形状

板のプレス加工を考えて、以下のモデル形状としている。この解析は計算負荷が掛かるので、幅方向は、薄い形状としている。

また、上型、下型と接触する板の幅は、上下型よりも少し小さくしている。これは、接触判定時に、確実に接触判定ができる様にする為。(板側を slave、上下型側を master とした時、slave の node が確実に master 側の face 内に収める為)



group 名	区分	meshGroup	備考
plate	volume	element	板
plateFixX	face	node	plate 根本側端面 (X 方向固定)
plateUp	face	node	plate 上面 (接触面: slave 側)
plateUd	face	node	plate 下面 (接触面: slave 側)
plateSide	face	node	plate 両側面 (Y 方向固定)
plateFix	edge	node	plate 根本端面下側 edge (XYZ 固定)

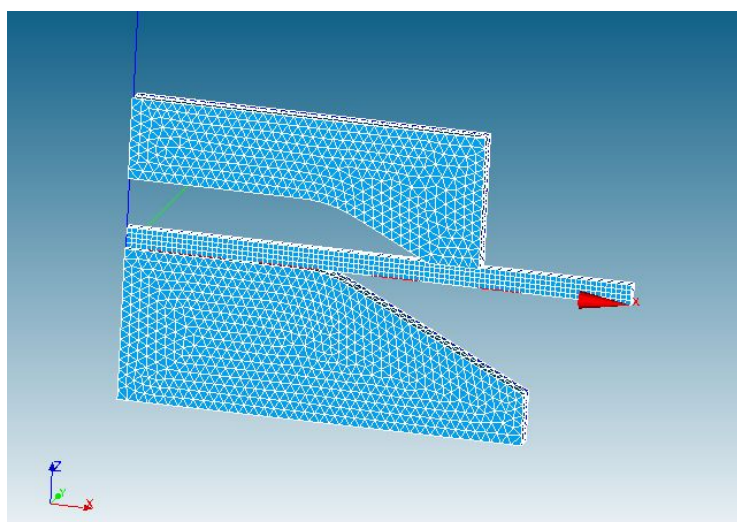
jigUp	volume element	上型
fixXY	face node	根本側端面 (XY 固定)
moveZ	face node	上面 (Z 方向移動面)
jigUpSide	face node	両側面 (Y 方向固定)
jigUp_contact	face surface	下面 (接触面: master 側)
jigUd	volume element	下型
fix	face node	根本端面 (XYZ 固定)
jigUdSide	face node	両側面 (Y 方向固定)
jigUd_contact	face surface	上面 (接触面: master 側)

メッシュは、以下の様に作成している。

解析は、上型を下げて板をプレスし、この後、上型を上げて型を離す解析を行う。

この解析は、接触を含む解析の為、1次要素とする。

上下型は形状が複雑な為 tetra、板は大きく変形する為、1次要素でも精度がある hexa でメッシュを作成している。



このメッシュを「Compound_plateJigs.unv」として保存する。

4-8-2. 設定内容

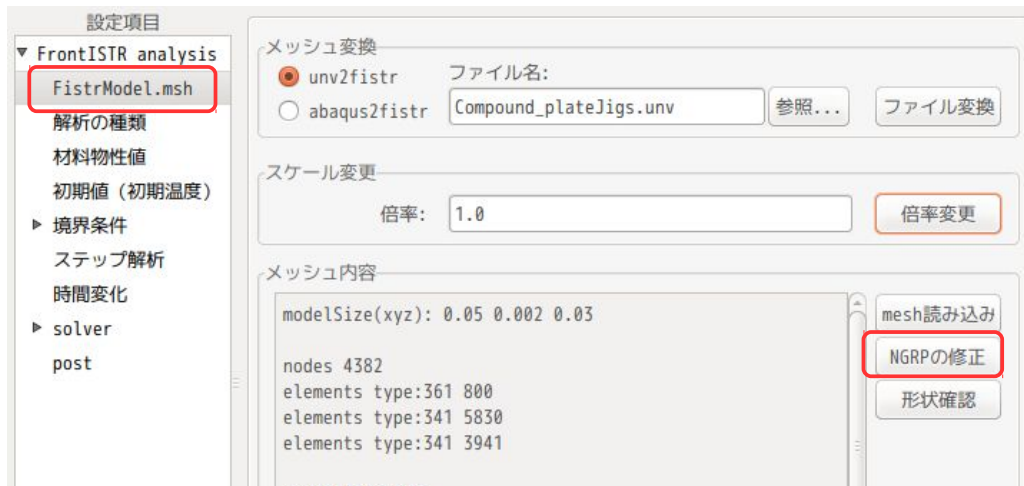
ここから EasyISTR 上で操作する。

4-8-1-1. メッシュ変換と節点グループの重複節点の削除

作成された unv メッシュを fistr 用のメッシュに変換し、scale を 1/1000 に設定しておく。

変換後、各節点 Group は、節点が重複して定義されている為、これを修正する。

修正の方法は、EasyISTR 上で、「NGRP 修正」ボタンをクリックして現れた画面「重複節点の削除」で修正する。



「重複節点の削除」画面上で、右図の様に節点グループの順番を設定する。

この設定は、plateの固定面、接触のslave面グループ、上下型の固定面、上型の移動面の優先順を高くして、これらに属する節点を残す。

削除する側の重複節点は、板の側面、上下型の側面に属する節点を削除する設定にしている。

この実行により、下図の画面が現れ、重複節点が削除された事が確認できる。



4-8-1-2. 解析の種類

解析の種類は、「非線形解析」に設定する。



4-8-1-3. 材料の設定

材料は、以下で設定する。

JigUd 下型 Steel
 jigUp 上型 Steel
 plate 板 steel0ther (塑性材料)

塑性材料 (steel0ther) は、材料 DB に存在しないので、Steel の物性値をそのままコピーして、steel0ther として DB に登録して塑性材料として使う。

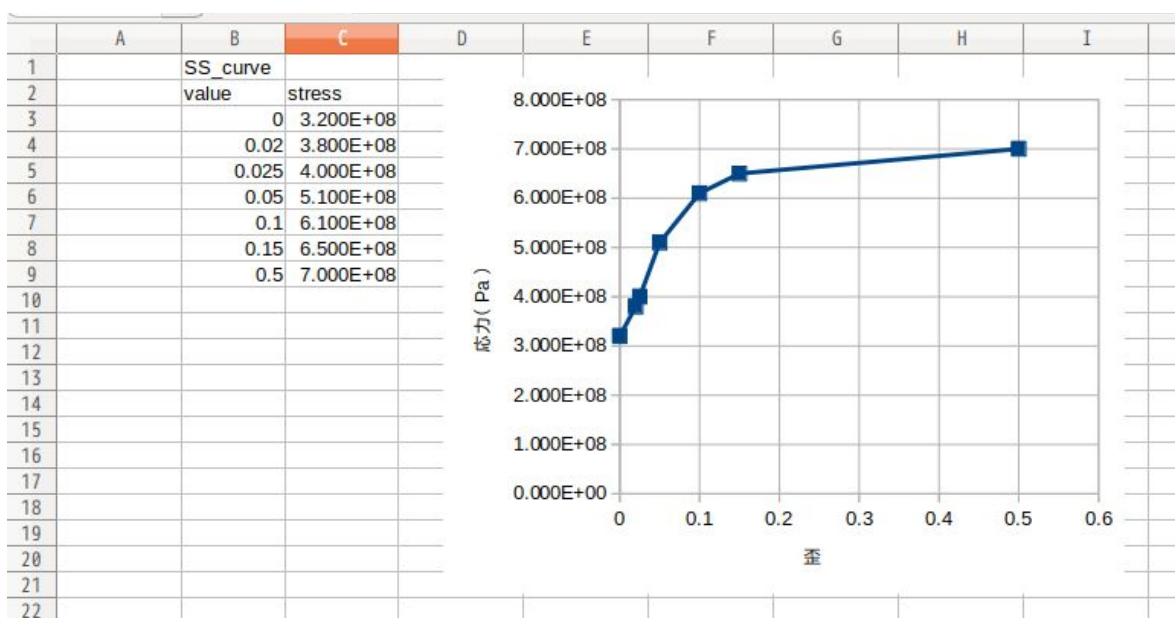
jigUd、jigUp については、弾性材料として以下の様に設定する。



plate については、塑性材料とするため、以下の様に設定する。



この後、塑性材料の応力歪曲線を定義する必要があるので、上図の「SS_data 作成・編集」ボタンをクリックして、これを作成する。以下の様に作成している。(値は、適当に作成している)

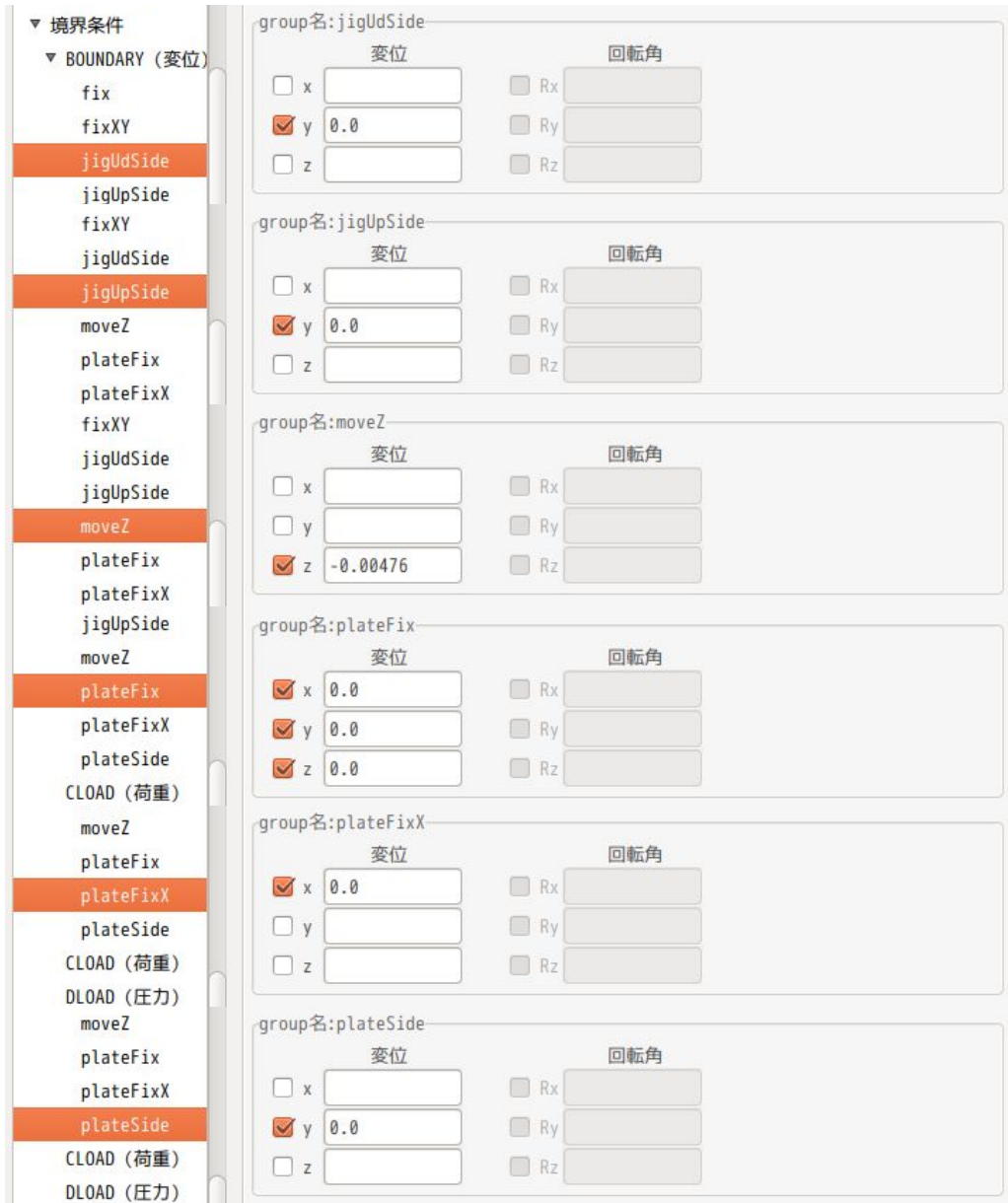


4-8-1-4. 境界条件の設定

変位拘束は、以下で設定している。

上型は、Z方向に移動させるので、「moveZ」については、Z方向に「-0.00476」移動させる条件に設定している。この値は、上型の下死点に相当する位置になる。





接触の設定は、以下で設定している。接触ペアは、2種類あるので各々設定する。



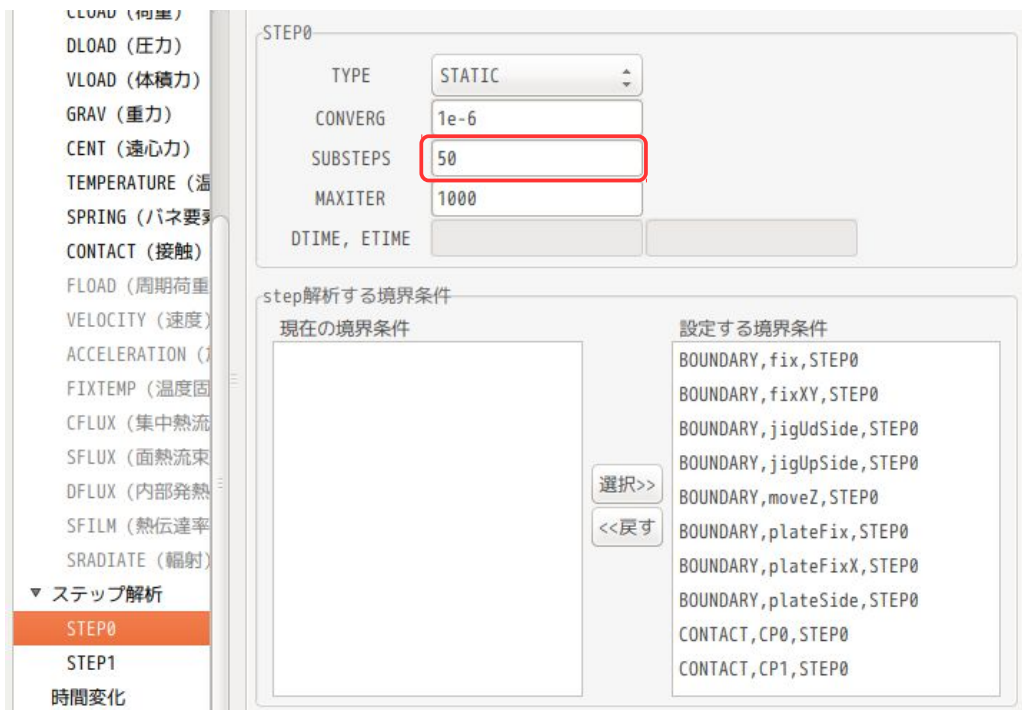


非線形解析のため、ステップ解析の設定が必要になる。

解析方法は、上型をZ方向に「-0.00476」移動させてプレスし、この後、上型をZ方向に「0.003」移動させて、型を離す設定にする。この為2ステップの解析になる。

まず、第1ステップ (STEP0) の設定は、以下で設定する。

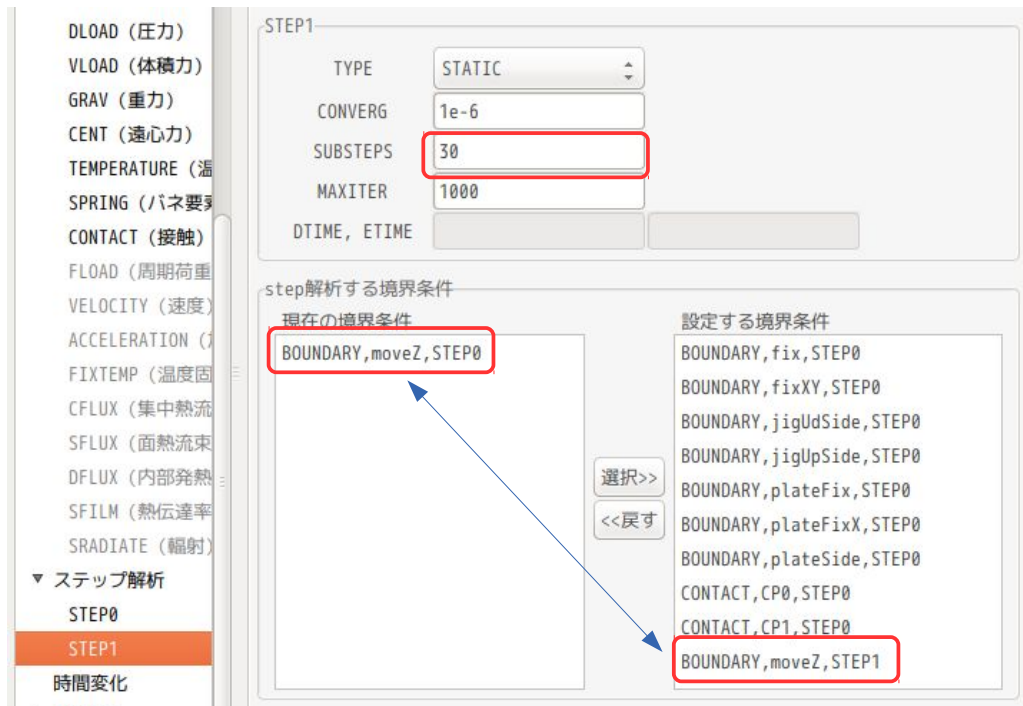
subStepは「50」としている。約0.1mm/1subStep で変化させていく設定。(ここを「5」程度の粗い設定では、変化が大きすぎ、収束しなかったため、0.1mm毎の変化に留めている。)



第2ステップ (STEP1) 側の境界条件は、「moveZ」をZ方向に「0.003」を設定して、上型を上方に移動させる設定にする。



第2ステップ (STEP1) の設定は、以下で設定した。
 subStepは、3mm 移動させるので、「30」としている。(0.1mm/1subStep の設定)
 関係する境界条件は、moveZの境界条件を STEP1 で設定した条件に入れ替えている。



4-8-3. 計算開始、結果の確認

計算開始させる前に、線形 solver を「MUMPS」に変更しておく。
 また、出力項目についても、以下を追加した。

積分点塑性歪: PL_ISTRAIN

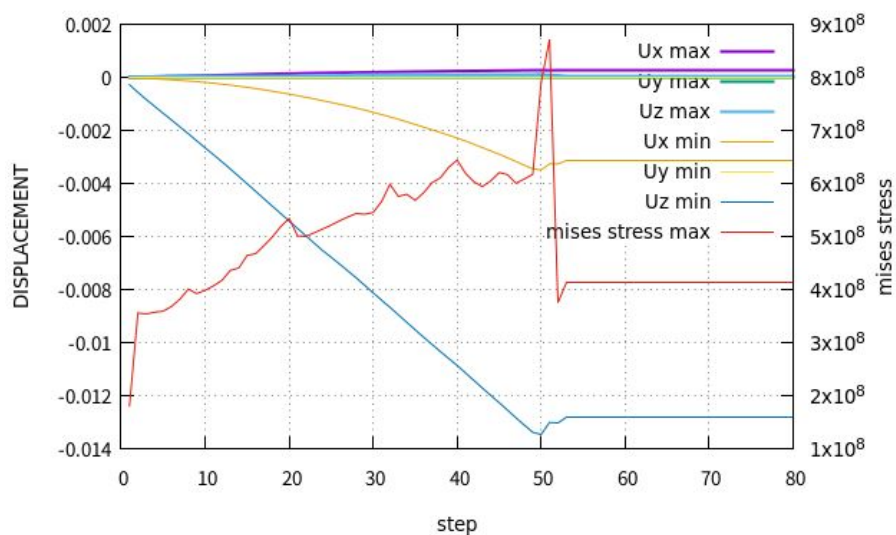
節点ひずみ: NSTRAIN

節点反力: REACTION

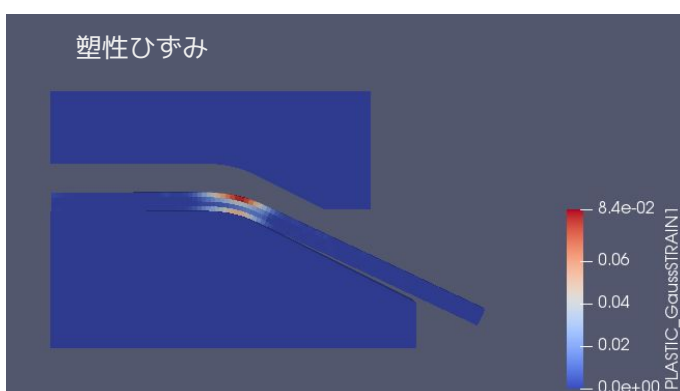
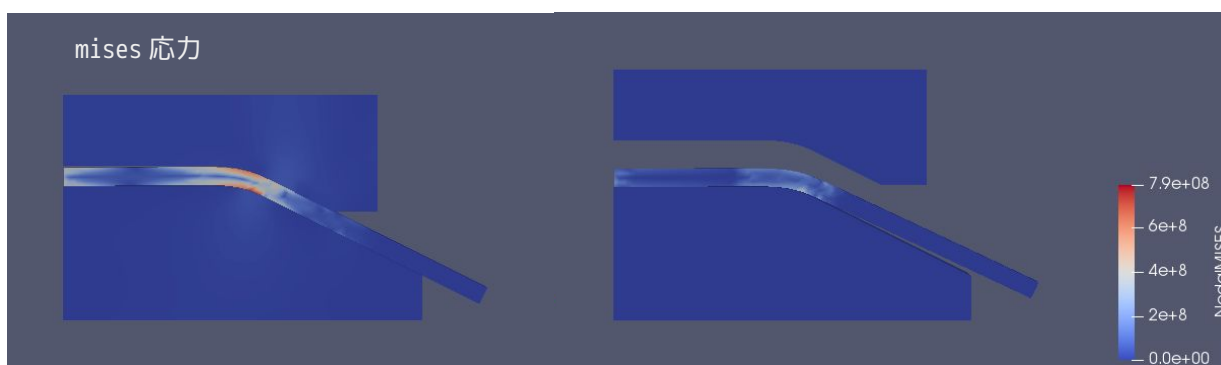
今回の計算のサブステップ数は、合計で 80 ステップになるので、結果出力頻度を「5」とした。

以上の設定で、計算開始させる。

計算負荷が大きく計算時間は、3.5h 掛かった。この間の計算状況は、以下になる。



計算結果は、以下になった。スプリングバックや塑性ひずみが確認できる。



4-9. 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。

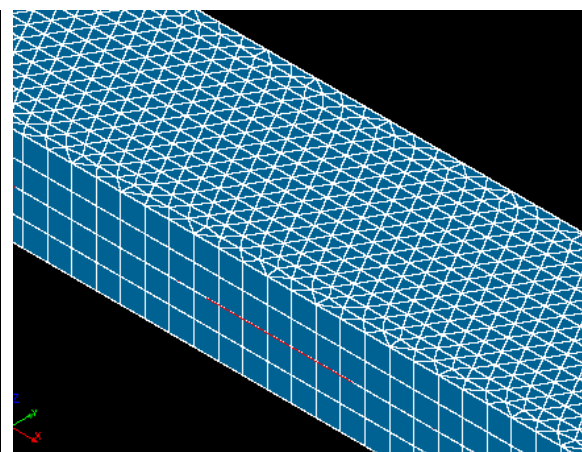
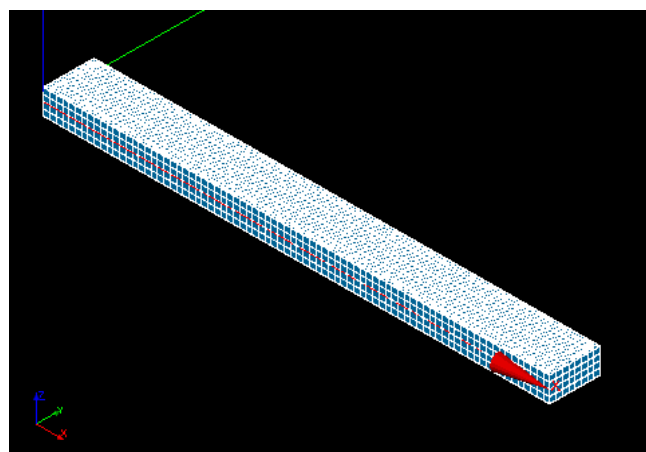
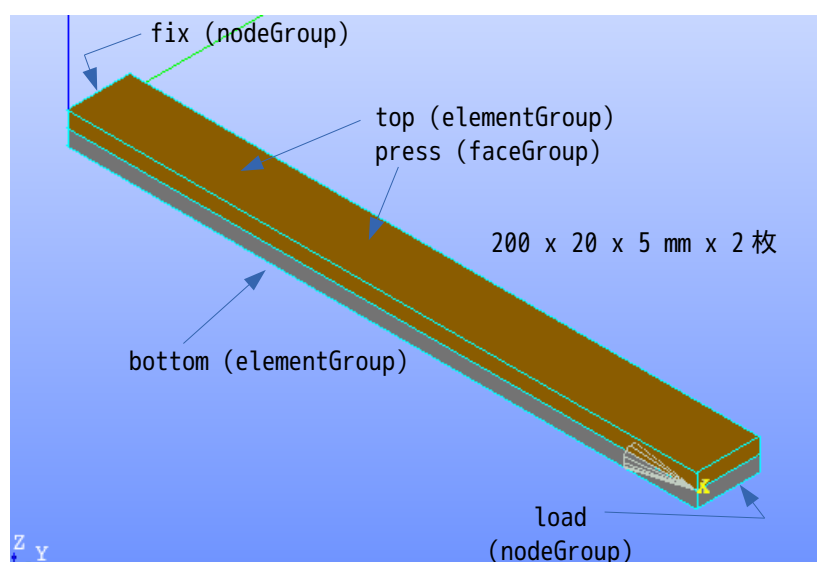
workFolder は、「CAE-fistr/Case/biMetal」フォルダを作成して、この中で解析を行う。

4-9-1. 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。

4-9-2. 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。

4-9-2-1. 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)

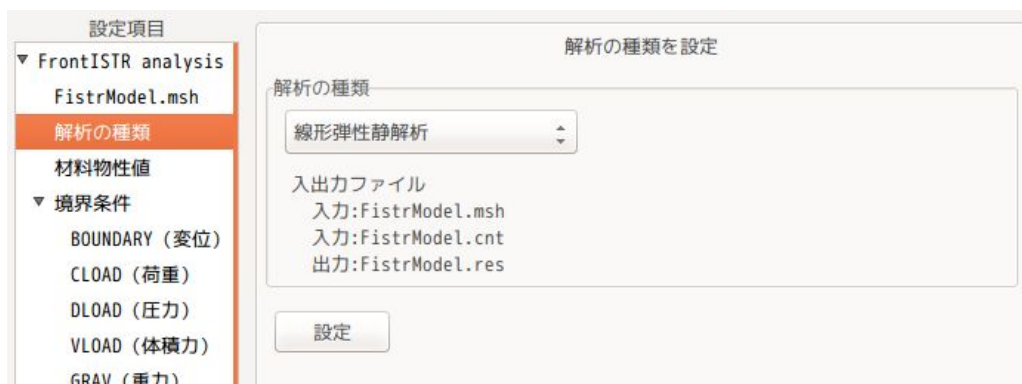
1. otherS
2. press

NGRP (nodeGroup)

1. fix
2. load

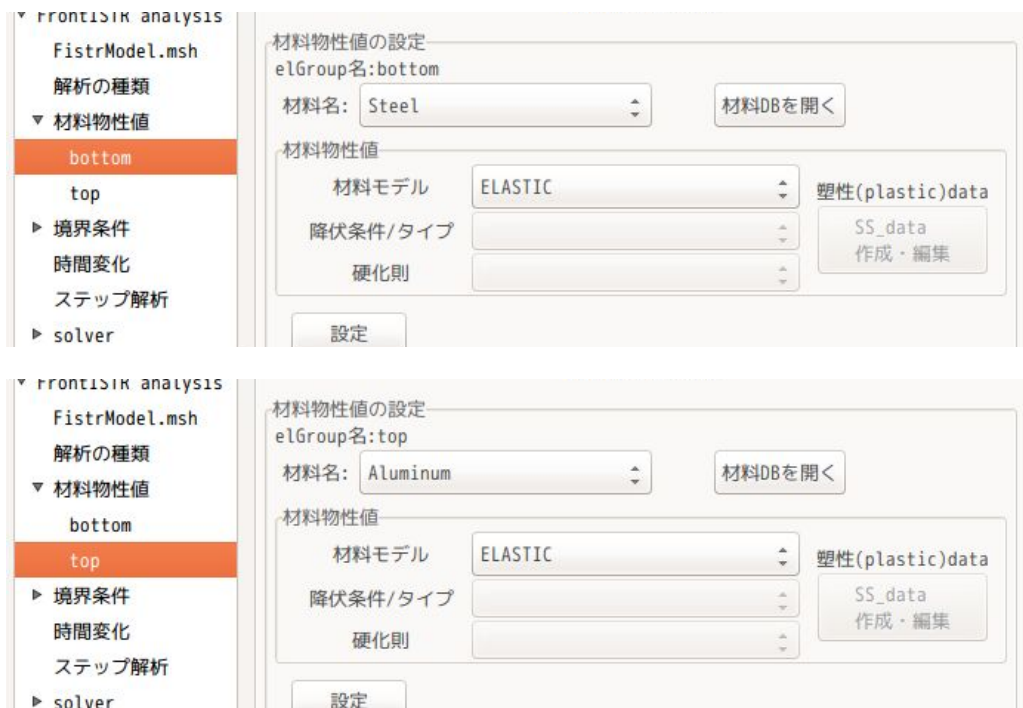
4-9-2-2. 解析の種類を設定

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)



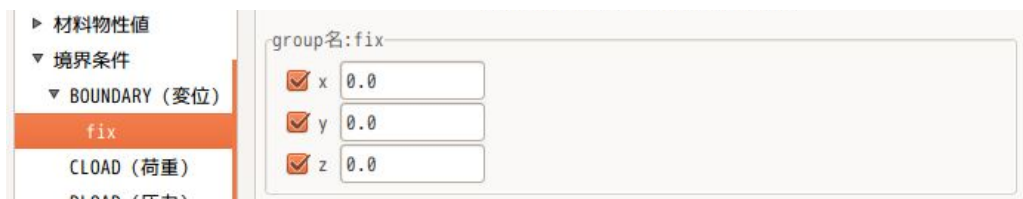
4-9-2-3. 材料物性値の設定

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



4-9-2-4. 境界条件の設定

境界条件は、ハイメタルの片側 (fix) を固定し、温度を 20℃から 100℃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、初期温度を「20.0」に設定し、Temp_Value を「100.0」に設定する。初期温度が、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。



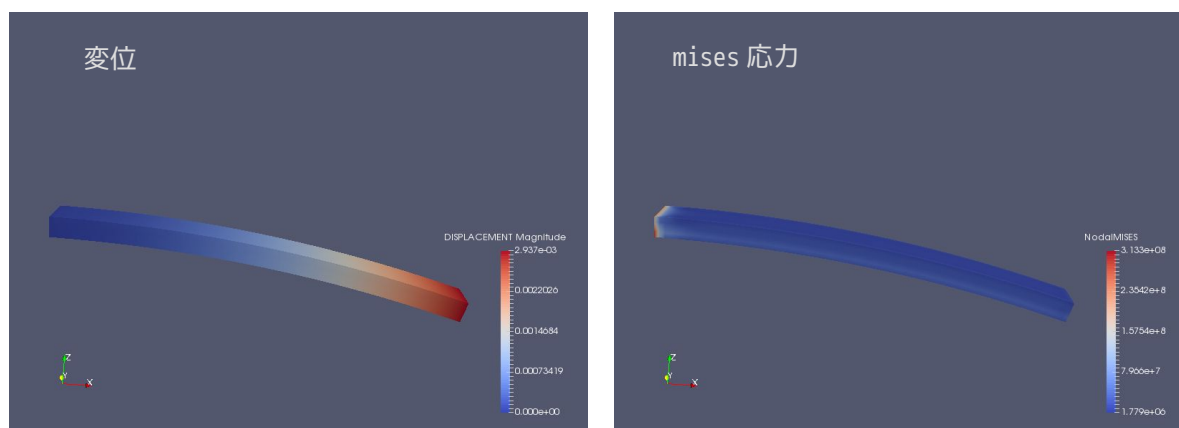
尚、ここで設定した初期温度については、mshファイル内の「!INITIAL CONDITION, TYPE=TEMPERATURE」で設定する内容になる。この設定は、設定項目 Tree 内の「初期値 (初期温度)」内で設定 (下図参照) しても同様な設定が行えるので、どちらで設定しても構わない。



4-9-2-5. 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、パイメタルが変形している。

4-10. 熱伝導解析（静解析）

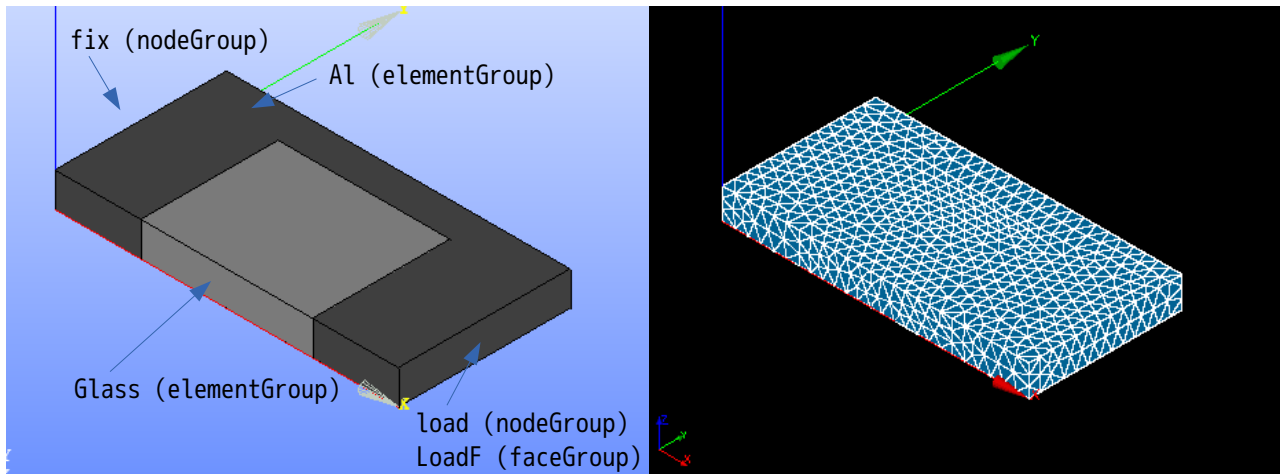
EasyISTR で熱伝導の静解析を行ってみる。

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。

4-10-1. 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。

4-10-2. 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。

4-10-2-1. 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
```

4-10-2-2. 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。



4-10-2-3. 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。



4-10-2-4. 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/m² 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。



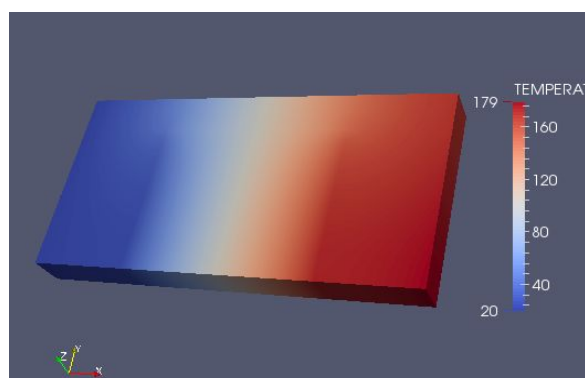
熱流束の設定方法は、nodeGroup に設定する方法と faceGroup に設定する方法があるが、ここでは faceGroup に熱流束を設定した。

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/m² を入力する。下図が入力した状態。



4-10-2-5. 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。



4-10-3. 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。

4-10-3-1. 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。



4-10-3-2. 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。





4-10-3-3. 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。

初期温度は、「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。



4-10-3-4. 計算開始、結果の確認

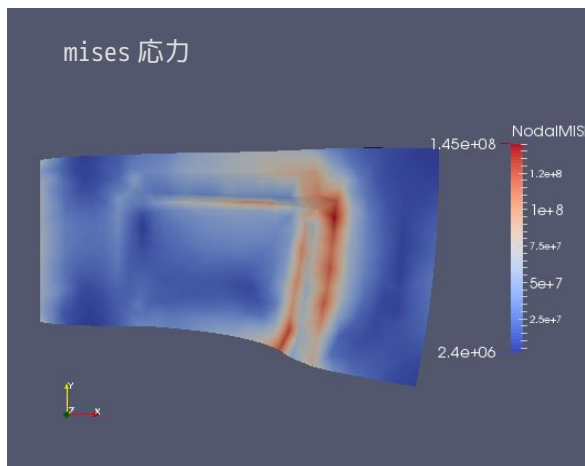
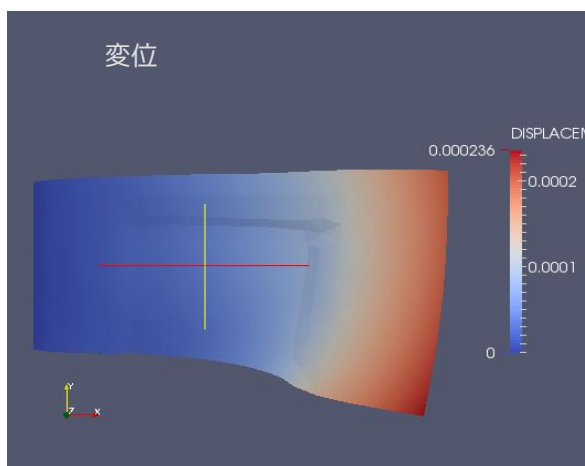
準備が全て整ったので、計算を開始する。

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。)



4-11. 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。

4-11-1. 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。

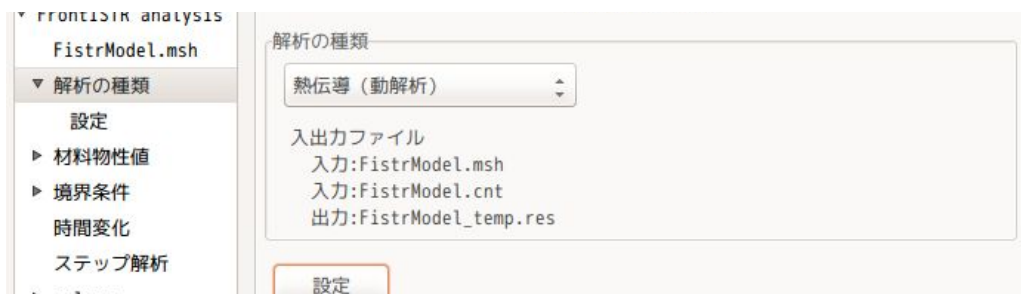
4-11-2. 熱伝導解析（動解析）の開始

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。

4-11-2-1. 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。



4-11-2-2. 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。



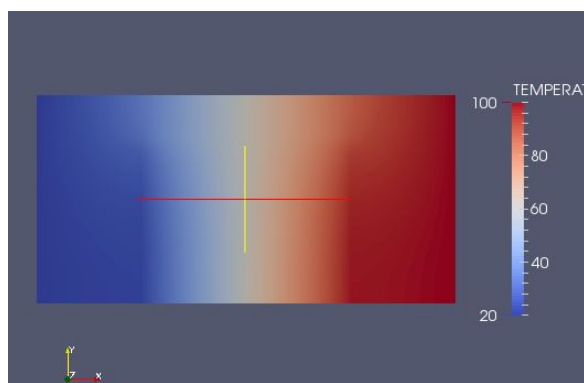


4-11-2-3. 計算開始、結果の確認

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。



4-11-3. 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。

4-11-3-1. 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。



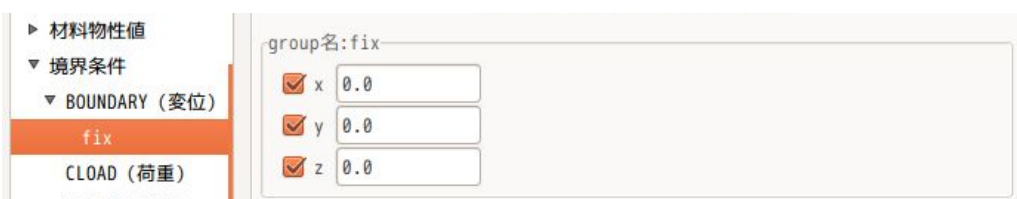
4-11-3-2. 材料物性値の設定

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。



4-11-3-3. 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。

初期温度は、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。

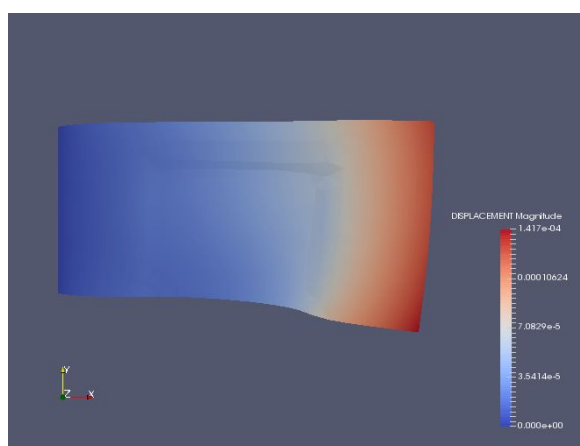
熱応力解析は、このデフォルトの設定を使って計算する。この計算は、最終の 60step の温度データを使って計算する設定になっている。



4-11-3-4. 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。



4-12. 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。

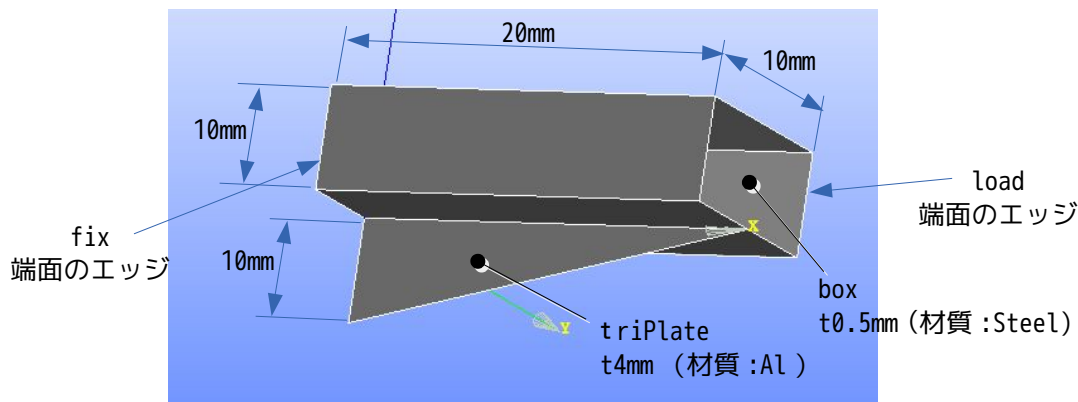
BOUNDARY の境界条件は、各軸の回転が追加されている。

応力、ひずみの計算結果は、シェル表裏の値が存在する。

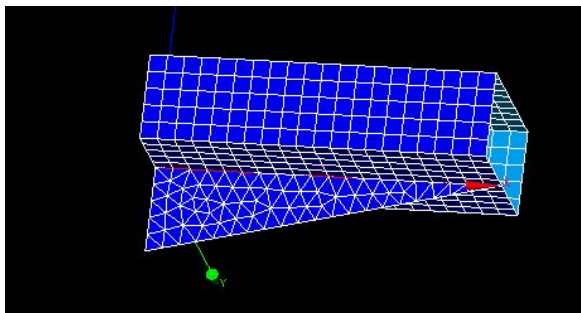
シェルの場合、線形 solver は DIRECT or MUMPUS のみ対応。

4-12-1. 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。



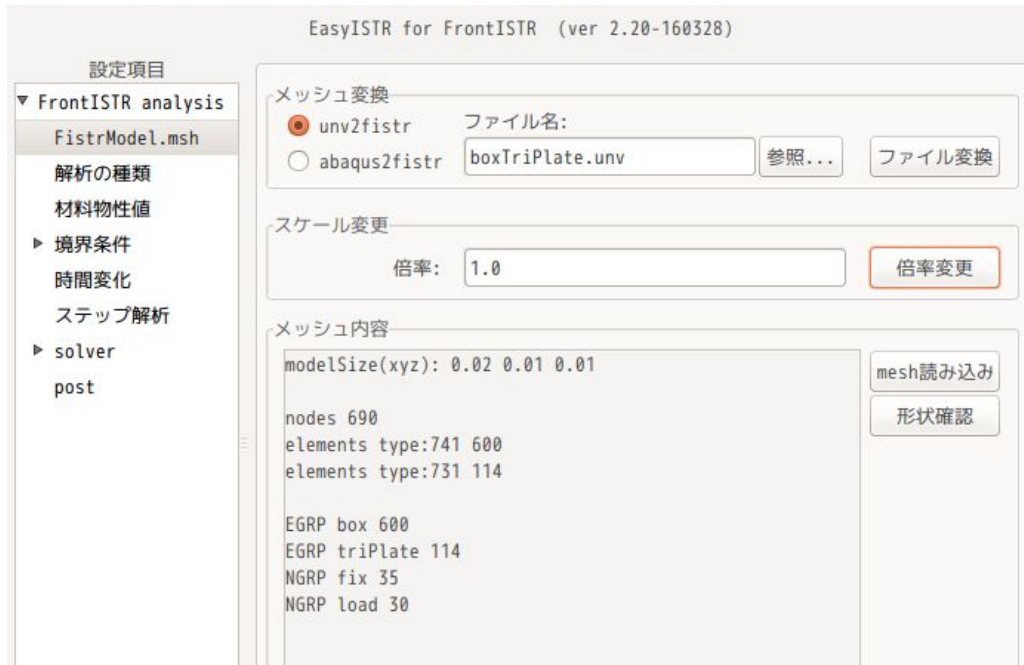
グループ化：
要素グループ : box, triPlate
節点グループ : fix, load

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。



4-12-2. 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。



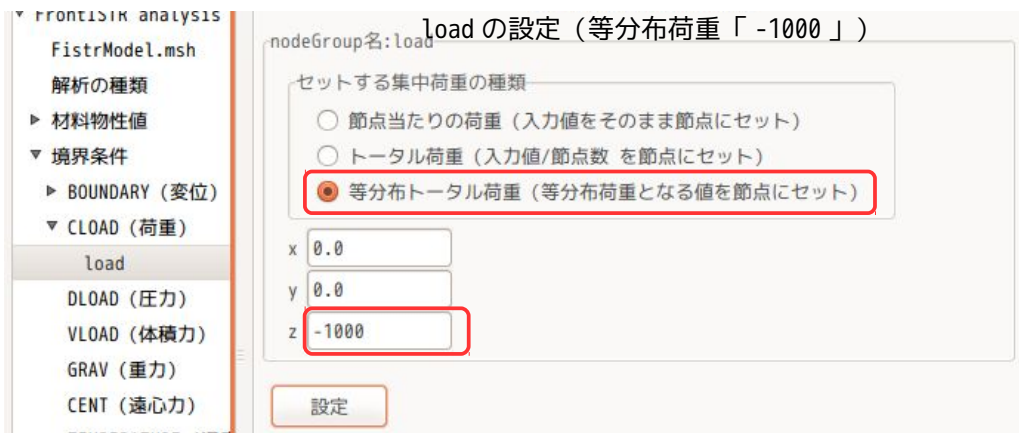
4-12-3. 境界条件の設定

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。

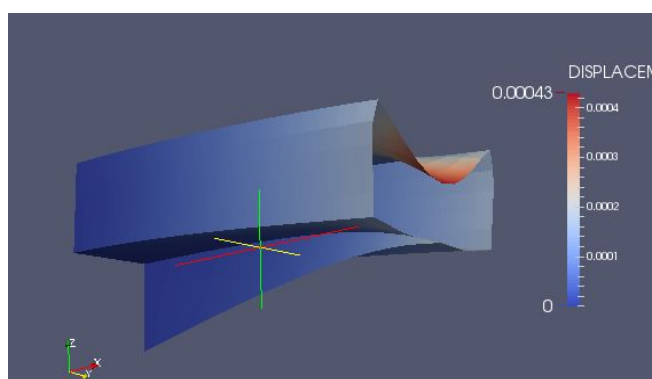


4-12-4. 計算開始、結果の確認

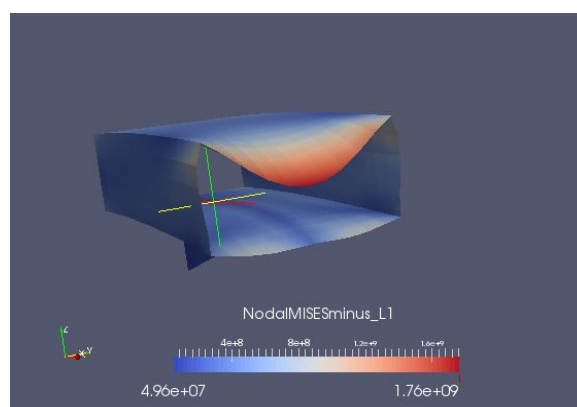
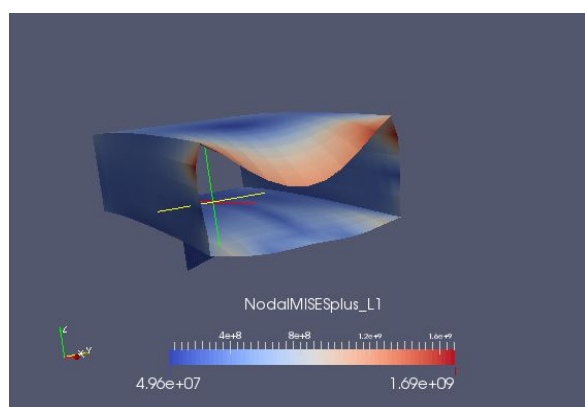
シェルの場合は、線形 solver が「DIRECT」or「MUMPS」しか計算できないので、どちらかを設定するが、ここでは、以下の様に「MUMPS」を設定している。



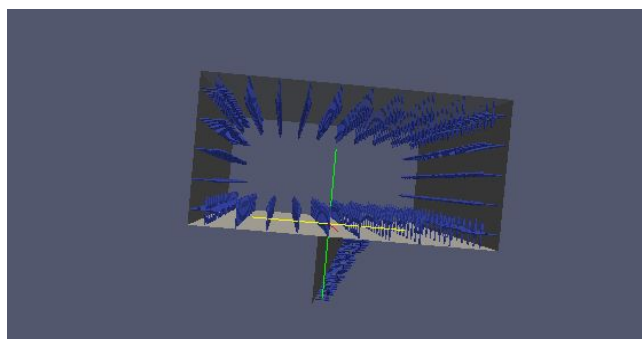
この後、通常通りに、「FrontISTR 実行」ボタンをクリックして計算させる。
計算終了後、結果を vtk 変換して、変位の計算結果を確認した結果が以下になる。



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。



4-12-5. 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。

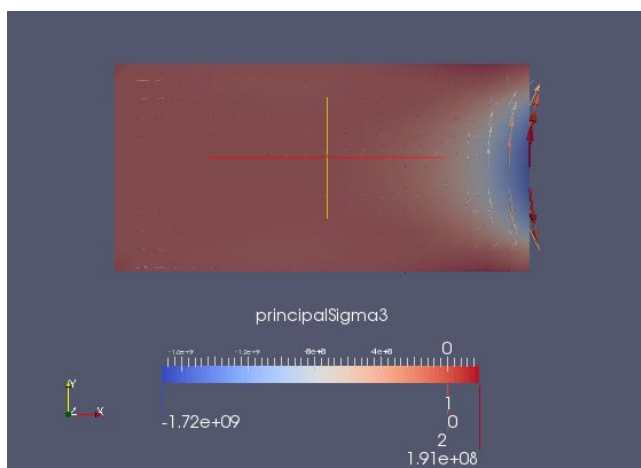
- NodalSTRESSplus_L1 応力：シェル要素の表側
- NodalSTRESSminus_L1 応力：シェル要素の裏側
- NodalSTRAINplus_L1 ひずみ：シェル要素の表側
- NodalSTRAINminus_L1 ひずみ：シェル要素の裏側

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 σ_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。



4-13. 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。）

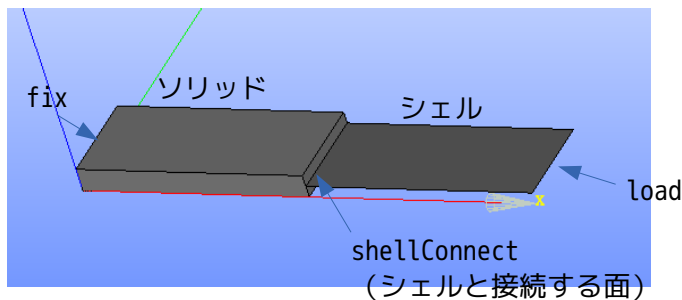
また、この解析は、微小変形に限られる。

4-13-1. 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は、特別な処理が必要な為、特別な名称を設定する様にしている。その名称として、「shellConnect***」又は「toShell***」にしている。（「***」は任意の文字）

ここでは、その接続面を「shellConnect」名でグループ化している。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。



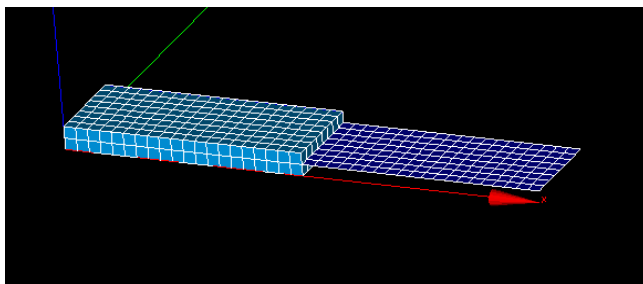
<グループ化>

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。



4-13-2. 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***、又は toShell***（***は任意の文字）と定義された面で変位と回転角が接続された状態になる。尚、接続面で前記した名称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。

今回は、接続面を「shellConnect」で定義しているのので、ここで回転角と変位の両方が接続された状態。

メッシュ変換により、FistrModel.msh ファイルができあがり、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。

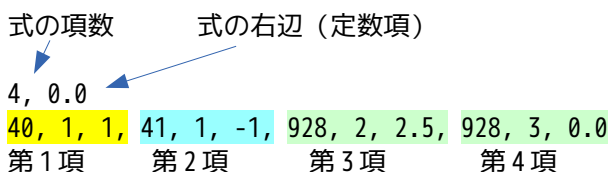
また、シェルとソリッドの回転角接続の為に、FistrModel.msh ファイル内に、「!EQUATION」が追加され、この方程式で接続される。

```
----- 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
!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、
shell の dummy 節点が追加される。
(solid、 shell 混在モデルの場合)

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***」又は
「toShell***」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。



- 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」

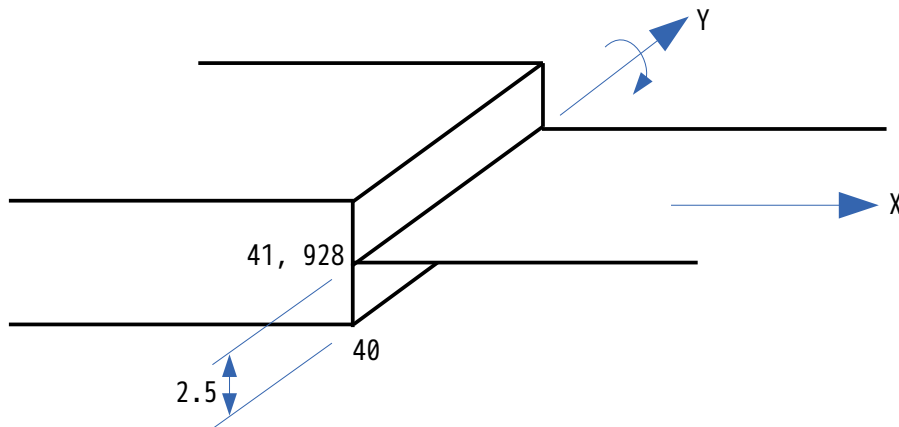
$$d_{40x} - d_{41x} + 2.5\theta_{928y} + 0.0\theta_{928z} = 0.0$$

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

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 × 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***」又は「toShell***」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

```

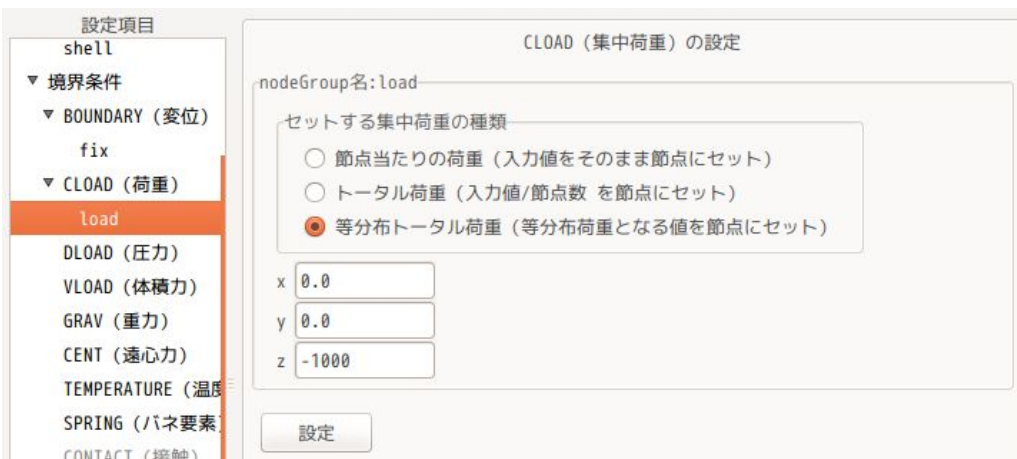
尚、上記のように、メッシュ変換時に!EQUATION を設定しているので、メッシュ変した後に、「スケール変更」をしない様に注意する事。この理由は、「スケール変更」すると、節点座標の値のみスケールが変わり、!EQUATION の寸法は変わらずそのままの為、方程式が成り立たなくなる為。

4-13-3. 材料、境界条件の設定

材料は、以下で設定している。

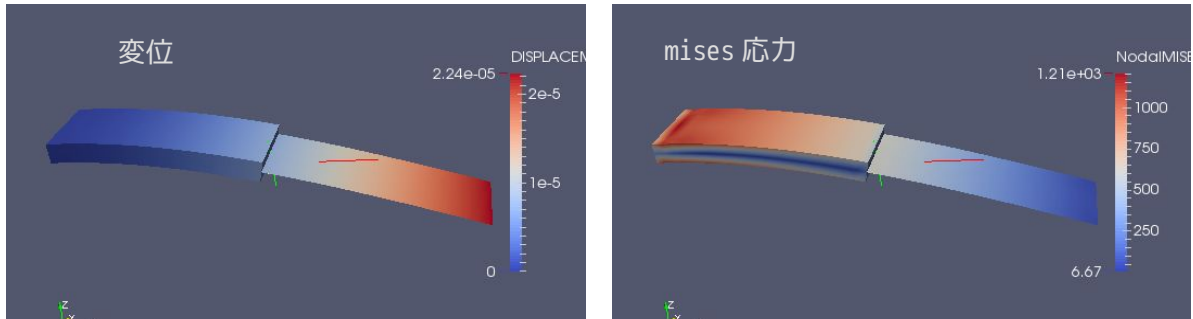


境界条件は、fix を固定し、load の等分布トータル荷重-1000 を入力して計算。

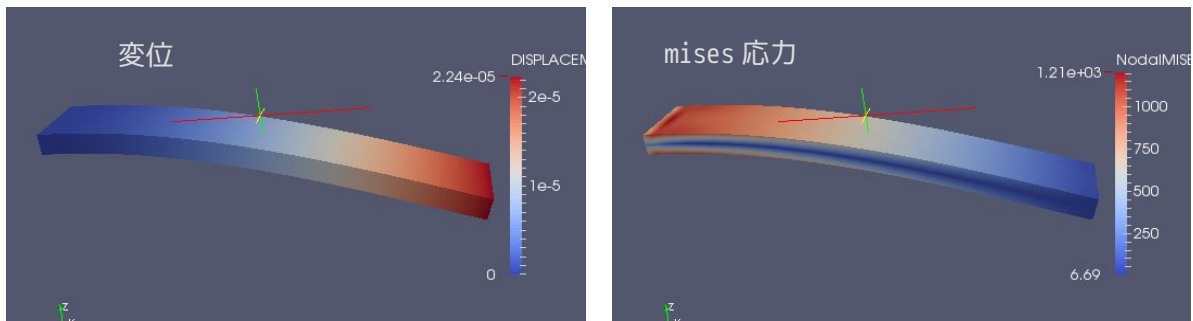


4-13-4. 計算開始、結果の確認

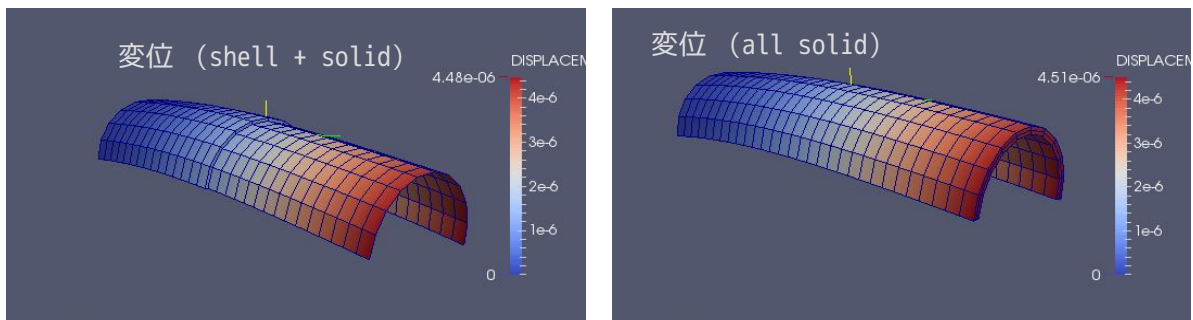
線形 solver を、「MUMPS」または「DIRECT」に設定して、計算を開始させる。
計算結果は、以下になる。



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。



4-14. 梁 (beam) 要素の解析

beam 要素 (線) 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。

beam 要素の定数 (断面積、各断面 2 次モーメント) 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。

beam の場合、線形 solver は DIRECT or MUMPUS のみ対応。

また、ここで、梁のたわみとねじりの 2 種類の解析を行い、理論解と比較してみる。

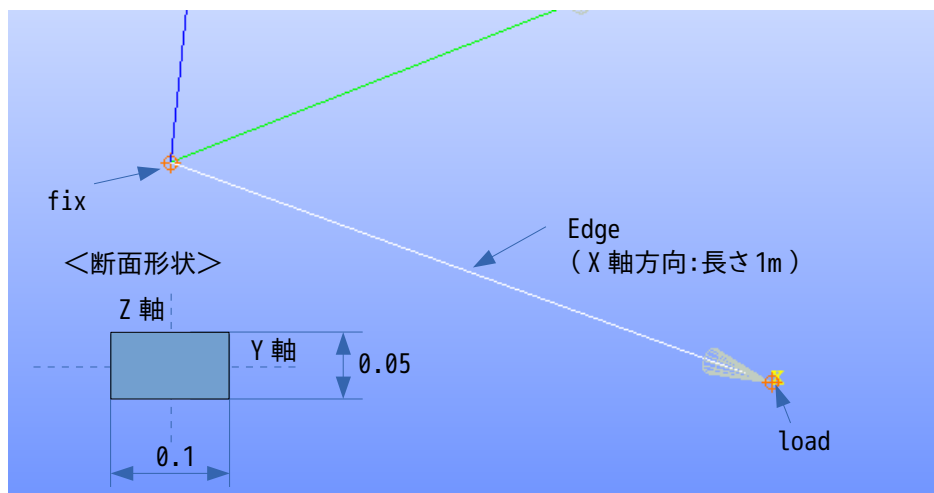
4-14-1. 梁のたわみ解析

角柱の片持ち梁の先端に荷重を掛け、先端のたわみを計算して、理論解と比較してみる。

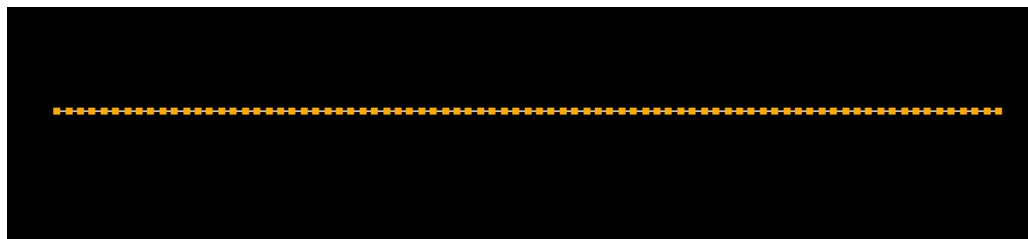
4-14-1-1. モデルの作成、メッシュの作成

モデルは、以下の様な片持ち梁を考える。

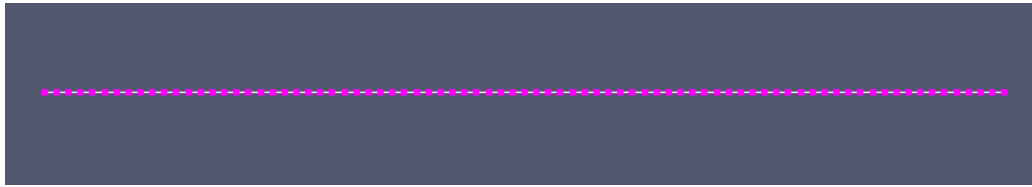
解析の条件は、fix を固定し、load の Z 軸方向に -100N を印加した場合と、Y 軸方向に -100N を印加した場合の 2 種類の解析を行ってみる。



上記の線分を salome 上でメッシュサイズ : 0.0125 で 1 次要素としてメッシュを切った。(要素数 : 80 ケ)
このメッシュを beamLine.unv として beamLine フォルダ内に保存する。

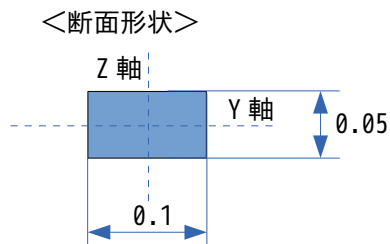


保存した beamLine.unv を EasyISTR 上でメッシュ変換する。
下図が、メッシュ変換後、paraView で確認した結果になる。



4-14-1-2. 材料の設定

beam 要素の場合、断面積と各軸方向の断面 2 次モーメントを求めておく必要がある。以下の様に求めた。



材料 : Al
 断面積 : $0.05 \times 0.1 = 0.005 \text{ m}^2$
 断面 2 次モーメント :
 $I_y = bh^3/12 = 1.041e-6 \text{ m}^4$
 $I_z = bh^3/12 = 4.166e-6 \text{ m}^4$
 ねじり定数 :
 $J_x = I_p = I_y + I_z = 5.207e-6 \text{ m}^4$

今回の beam 要素は、global 座標系の X 軸方向に作成している。
 これに対し、beam の座標系は、beam の方向が X 軸で YZ 軸は beam に直交する方向になる。この為、beam の YZ 軸方向を決定する必要がある。この方向を設定するのが「参考軸方向」になる。この「参考軸方向」に、beam の Z 軸方向を示す単位ベクトルを設定する。今回の場合は、この beam の Z 軸方向は、global 座標の Z 軸方向とし、その単位ベクトル $(0,0,1)$ を「参考軸方向」に設定する。
 これらの値を EasyISTR 上で以下の様に設定する。



4-14-1-3. 境界条件の設定

境界条件は、fix を固定し、load の Z 軸方向に -100N を設定する。以下の様に設定した。

The screenshot displays two configuration panels from the EasyISTR software. The top panel, titled 'BOUNDARY (節点・変位拘束) の設定', shows the 'fix' boundary condition. It includes a 'group名: fix' and a table of constraints:

変位		回転 (shellのみ)	
<input checked="" type="checkbox"/> x	0.0	<input checked="" type="checkbox"/> Rx	0.0
<input checked="" type="checkbox"/> y	0.0	<input checked="" type="checkbox"/> Ry	0.0
<input checked="" type="checkbox"/> z	0.0	<input checked="" type="checkbox"/> Rz	0.0

The bottom panel, titled 'CLOAD (集中荷重) の設定', shows the 'load' node group. It includes a 'nodeGroup名: load' and a section for 'セットする集中荷重の種類' with three radio button options:

- 節点当たりの荷重 (入力値をそのまま節点にセット)
- トータル荷重 (入力値/節点数 を節点にセット)
- 等分布トータル荷重 (等分布荷重となる値を節点にセット)

Below these options are input fields for the load values:

x	0.0
y	0.0
z	-100

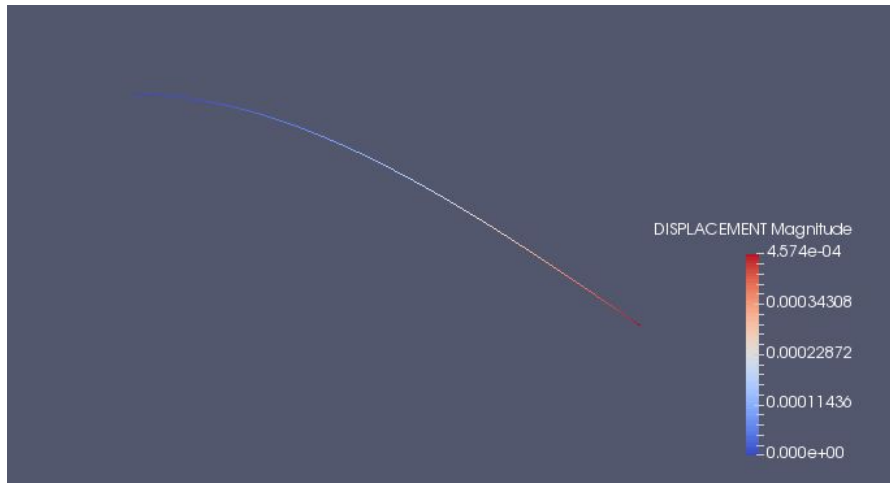
4-14-1-4. 計算開始、結果の確認

beam の場合は、シェルと同様に線形 solver が「DIRECT」or「MUNPS」しか計算できないので、どちらかを選択する。ここでは、「MUNPUS」を選択している。

The screenshot shows the '線形 solver' configuration panel. It includes a 'METHOD' dropdown set to 'MUMPS' and a '設定' button. The following table summarizes the solver parameters:

Parameter	Value	Description
METHOD	MUMPS	解法
PRECOND	1	前処理手法
NIER	20000	反復回数
RESID	1.00000e-06	打ち切り誤差

設定後「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。



最大変位が、「4.57e-4 m」と確認できる。

また、荷重の設定を Y 軸方向に -100N を印加した場合について、同様に計算した結果は、「1.14e-4 m」になる。

4-14-1-5. 結果の検証

前項の結果を理論解と比較してみる。

片持ち梁の先端のたわみ δ は、下式で求めることができるので、値を代入して先端のたわみを求めてみる。

$$Z \text{ 方向に } 100\text{N}: \delta = \frac{Pl^3}{3EI} = \frac{100 \times 1.0^3}{3 \times 70 \times 10^9 \times 1.041 \times 10^{-6}} = 4.574 \times 10^{-4} \text{ m}$$

$E = 70 \times 10^9$ (解析で使用した値)

$$Y \text{ 方向に } 100\text{N}: \delta = \frac{Pl^3}{3EI} = \frac{100 \times 1.0^3}{3 \times 70 \times 10^9 \times 4.166 \times 10^{-6}} = 1.143 \times 10^{-4} \text{ m}$$

となり、各方向とも、一致している。

4-14-2. 丸棒のねじり解析

片持ち梁（丸棒）の先端に、モーメントを印加して、丸棒をねじる解析を行い、理論解と比較してみる。

4-14-2-1. モデルの作成

モデルは、下図の様なモデル（beam が 3 本）を作成して解析する。

解析する条件は、

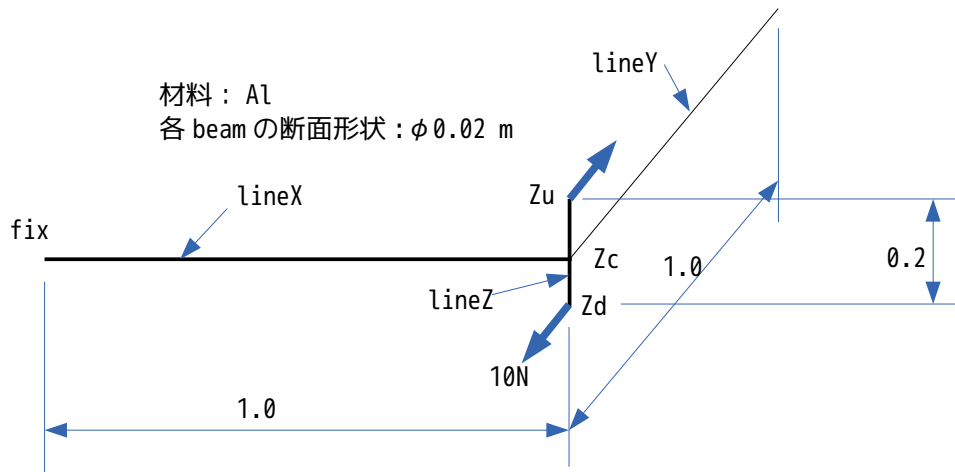
fix を固定する。

Zu 点と Zd 点に偶力を加え、lineX をねじる。

lineX の先端 Zc 点は、動かない様に、変位のみを固定する。

ねじれ角は、lineY の先端のたわみから角度を算出する。

として、解析する。



メッシュは、salome を使って、要素サイズ 0.025 でメッシュを作成している。
できあがったメッシュを crossBeamLine.unv として保存しておく。

4-14-2-2. 材料物性値の設定

今回の beam は、3 本あるが、いずれも $\phi 0.02$ の丸棒としているので、beam の定数は、以下になる。

断面積: 0.000314 m^2
 $I_y = I_z = \pi d^4 / 64 = 7.85e-9 \text{ m}^4$
 $J_x = J_p = \pi d^4 / 32 = 15.7e-9 \text{ m}^4$

また、各 beam の「参考軸方向」は、以下で定義した。

beam 名	参考軸方向	beam 座標系の Z 軸方向
lineX	(0, 0, 1)	global 座標系の Z 軸方向
lineY	(0, 0, 1)	global 座標系の Z 軸方向
lineZ	(0, 1, 0)	global 座標系の Y 軸方向

この為、以下で beam の定数を設定した。(材料は、Al で設定している。)



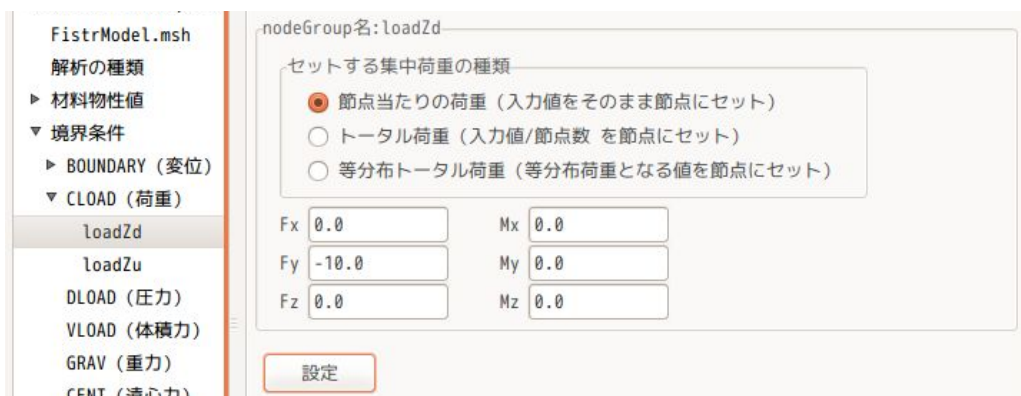


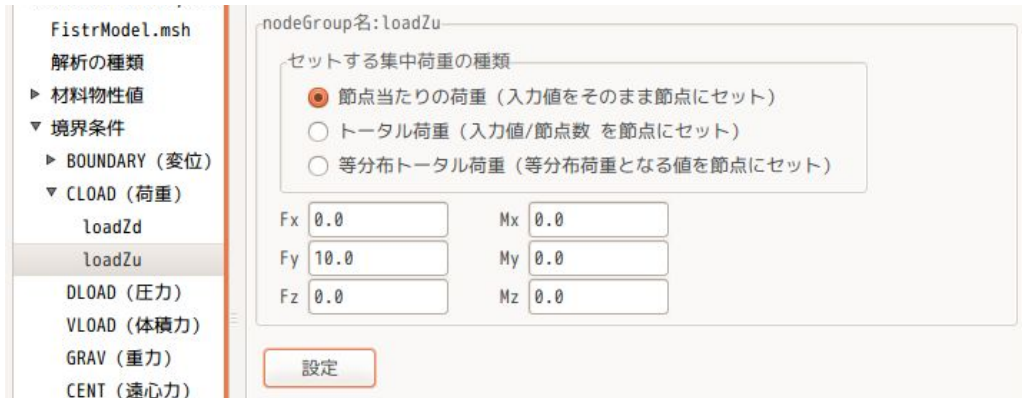
4-14-2-3. 境界条件の設定

境界条件は、変位拘束を「fix」「loadZc」に以下の様に設定する。

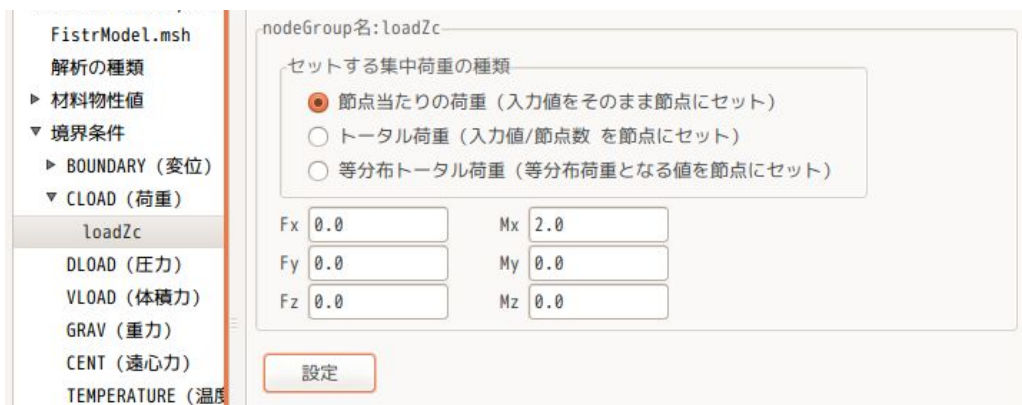


荷重は、「loadZd」と「loadZu」で偶力となる様に 10.0、-10.0 Nを設定する方法と、偶力相当のモーメント (2.0 Nm) を「loadZc」の x 軸回りに負荷する方法の 2 種類の方法で解析してみる。まず、偶力を設定して、計算してみる。下図の様に設定した。





「loadZc」の x 軸回りに 2.0 Nm を負荷する場合は、以下の様に設定する。



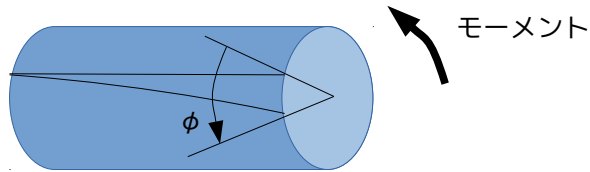
4-14-2-4. 計算開始、結果の確認

計算は、線形 solver を MUNPS に設定し計算を開始させる。下図がこの結果になる。最大変位は、lineY の先端部で、 $4.895e-3$ m と確認できる。（偶力、モーメントの設定共、同じ結果になる。）



4-14-2-5. 結果の検証

丸棒のねじれ角 ϕ の理論式は、以下で与えられる。



$$\phi = \frac{Tl}{GI_p}$$

$$l: \text{梁の全長} \quad l = 1.0 \text{ m}$$

$$T: \text{モーメント} \quad T = 0.2 \text{ m} \times 10 \text{ N} = 2 \text{ Nm}$$

$$G: \text{横弾性係数} \quad G = \frac{E}{2(1+\nu)} = \frac{70\text{e}9}{2 \times (1+0.345)} = 26.0\text{e}9 \text{ Pa}$$

$$I_p: \text{断面極 2 次モーメント} \quad I_p = \frac{\pi d^4}{32} = 15.7\text{e-}9 \text{ m}^4$$

$$\phi = \frac{2 \times 1}{26.0\text{e}9 \times 15.7\text{e-}9} = 0.00490 \text{ rad}$$

lineX 先端のねじれ角は、0.00490 rad になる。lineY の長さは 1m の為、lineY 先端のたわみは、0.00490m になる。この値は、beam の解析結果とほぼ一致している。

4-15. ビーム、シェル、ソリッド混在モデルの解析

ビーム、シェル、ソリッドが混在しているモデルを解析してみる。
混在するモデルの場合、使用できる要素は、以下の要素に限定される。

ビーム： 641 (beamの1次要素)
シェル： 741、781 (三角形、四角形の1次要素)
ソリッド： 341、361、351 (tetra, hexa, prismの1次要素)

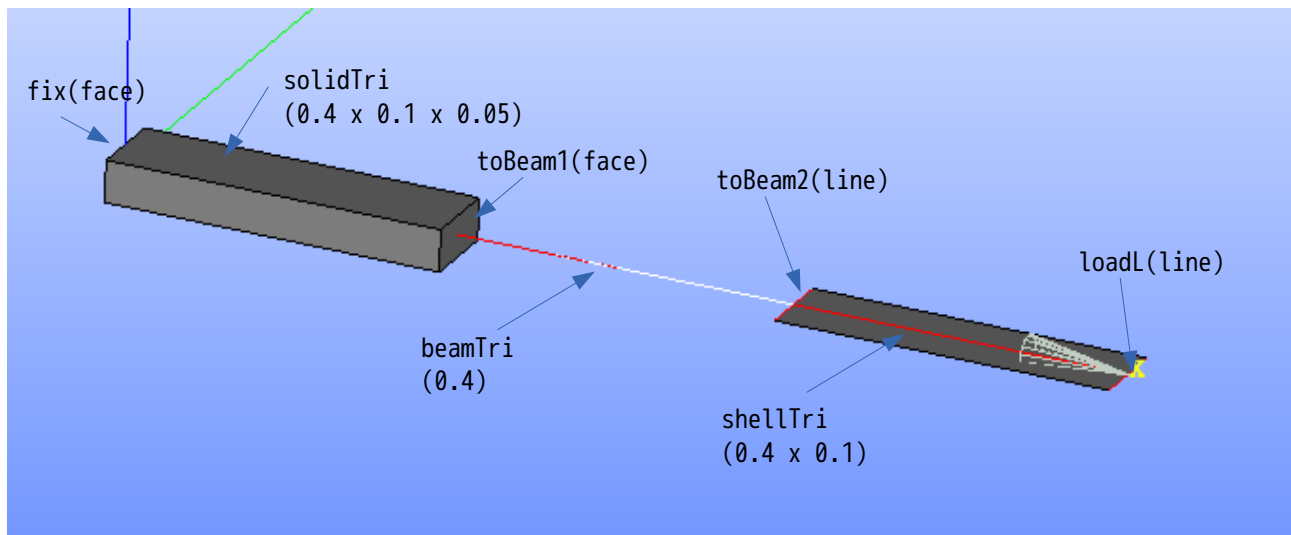
また、計算は、線形弾性静解析に限られ、線形 solver も MUMPU or DIRECT に限られる。

4-15-1. 梁のたわみ解析

ビーム、シェル、ソリッドで構成された片持ち梁のたわみを解析してみる。
梁の断面形状は、ビーム、シェル、ソリッドとも同じ形状 (0.1m x 0.05m) として計算し、理論計算と比較してみる。

4-15-1-1. モデルの作成

以下のようにビーム、シェル、ソリッドが混在するモデルを考えて解析してみる。ビーム、シェルの断面形状は、ソリッド (solidTri) と同じ形状 (0.1 x 0.05) で設定する。



beam や shell の接続部の名称は、以下のルールにより設定する。

solid 側の shell 接続部：	toShell***	(又は shellConnect***)
solid 側の beam 接続部：	toBeam***	(又は beamConnect***)
shell 側の beam 接続部：	toBeam***	(又は beamConnect***)

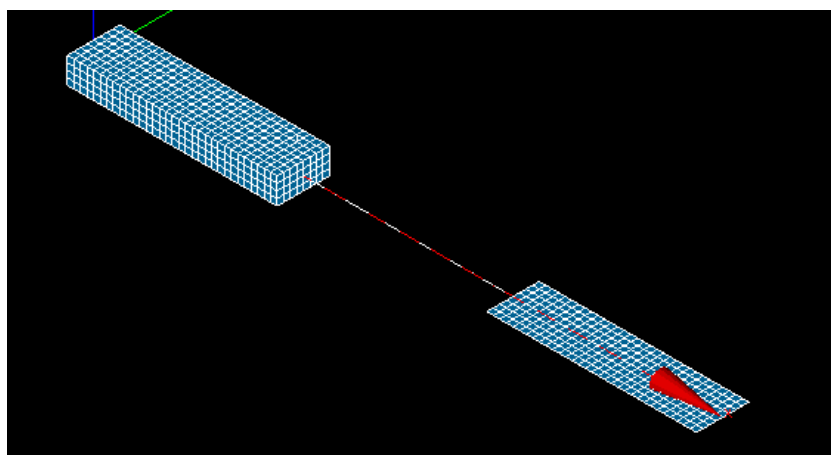
上記の「***」は、任意の文字で設定できる。この設定を行っておくと、メッシュ変換時に、各接続部で回転を伝達させる「!EQUATION」を作成してくれる。上記の名称を設定しなかった場合は、回転が拘束されない解析を行うことになる。(!EQUATION が作成されない。)

今回のモデルでは、beam との接続部は 2 箇所あるので、各々の接続部の名称を「toBeam1」、「toBeam2」として、別な名称としている。下表を参照。

区分	名称	内容
solid	solidTri	ソリッド
	fix	固定面 (face)
	toBeam1	beam への接続部 (face)
beam	beamTri	ビーム
shell	shellTri	シェル
	toBeam2	beam への接続部 (line)
	loadL	荷重印加部 (line)

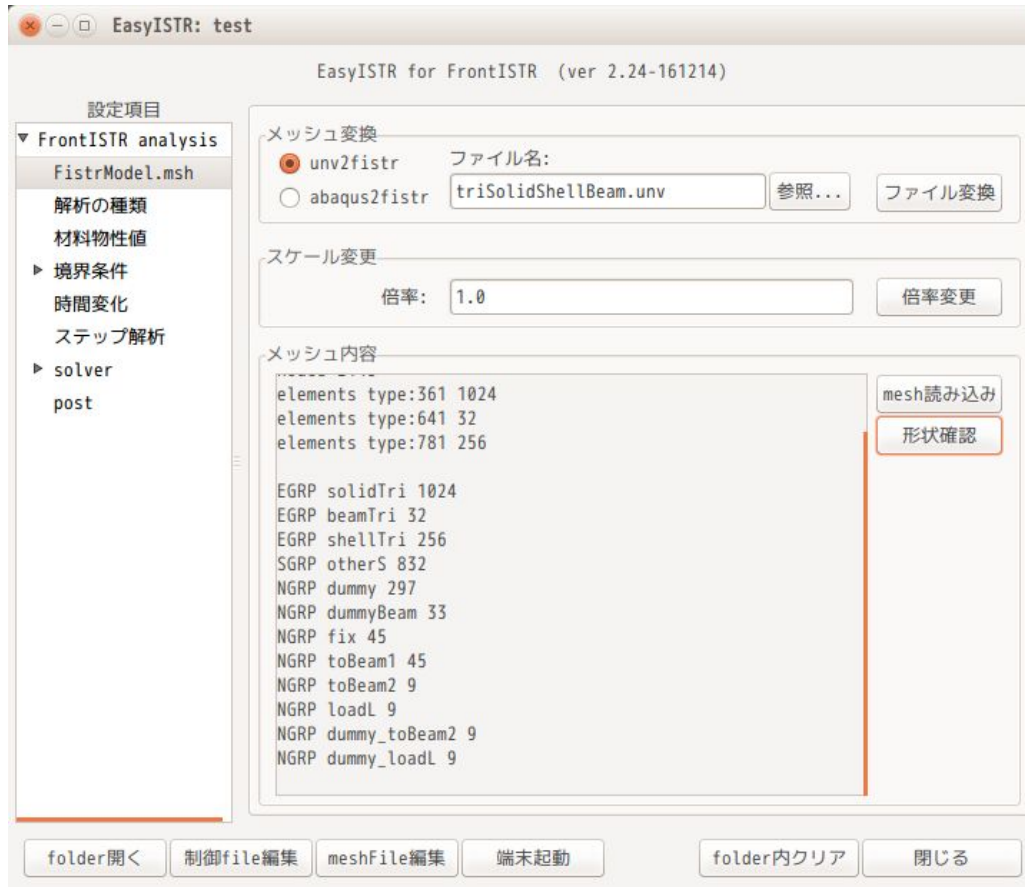
このモデルのメッシュを以下の様に作成。
できあがったメッシュを「triSolidShellBeam.unv」として保存する。

salome でメッシュを作成する場合は、solid、shell、beam 各々単独でメッシュをきり、できあがった3個のメッシュを「Build Compound」で一体のメッシュを作成する。compound する時、「Merge coincident nodes and element」にチェックを付けておく。これにより、3個のメッシュが節点を共有する状態で、一体のメッシュができあがる。



4-15-1-2. メッシュ変換

EasyISTR 上で、できあがった unv 形式のメッシュファイルを FrontISTR 形式に変換する。



要素が「361, 641, 781」の3種類存在していることが判る。また、このメッシュ変換により、dummyのnodeGroup (dummy, dummyBeam, dummy_toBeam2, dummy_loadL) ができあがっている。

また、この変換により、beamとの接続部に、beamの回転を拘束する「!EQUATIN」がメッシュファイルに追加される。このEQUATIONは、shellの回転拘束と同じ考え方で作成している。この!EQUATIONは、メッシュ変換時に作成しているので、メッシュ変換後、scaleの変更する事はできない。(solid、shell混在モデルの解析と同様の理由による。)

4-15-1-3. 解析の種類

solid、shell、beam混在モデルの場合は、線形弾性解析に限られるので、「線形弾性静解析」を選択する。



4-15-1-4. 材料の設定

全ての材料は「Aluminum」として、以下の様に設定した。beamの断面形状は、4-13-1項と同じ形状の為、断面2次モーメント等の計算は省略する。

The screenshot displays the 'beamの設定' (Beam Settings) dialog box. The left sidebar shows a tree view with 'solidTri' selected under '材料物性値' (Material Properties). The main panel is divided into three sections:

- beamの設定**: Reference axis directions (vx: 0.0, vy: 0.0, vz: 1.0), cross-sectional area (0.005), and second moments of area (Iyy: 1.041e-6, Izz: 4.166e-6, Jx: 5.207e-6). A '設定' (Settings) button is at the bottom.
- 板厚の設定 (shell)**: Plate thickness (0.05) and thickness direction integration points (5).
- 材料物性値の設定**: Material name set to 'Aluminum'. Below, the material model is 'ELASTIC', and the plasticity data is 'SS_data' (with '作成・編集' button).

4-15-1-5. 境界条件の設定

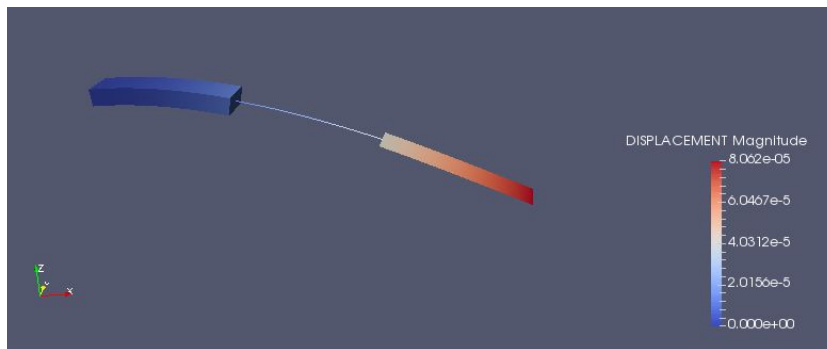
境界条件は、fixを固定し、loadLに等分布荷重 10Nを設定する。

The screenshot displays the 'group名: fix' and 'nodeGroup名: loadL' dialog boxes. The left sidebar shows 'fix' selected under '境界条件' (Boundary Conditions) and 'loadL' selected under 'CLOAD (荷重)' (Loads).

- group名: fix**: Displacement (変位) is fixed for x, y, and z (checked). Rotation (回転) is not fixed for Rx, Ry, or Rz. A '設定' (Settings) button is at the bottom.
- nodeGroup名: loadL**: Concentrated load type (集中荷重の種類) is set to '等分布トータル荷重' (Uniform total load). Force components are: Fx: 0.0, Fy: 0.0, Fz: -10, Mx: 0.0, My: 0.0, Mz: 0.0. A '設定' (Settings) button is at the bottom.

4-15-1-6. 計算開始、結果の確認

計算は、線形 solver を「MUMPS」に設定して、計算を開始させる。(MUMPS or DIRECT を選択する。) 結果をデータ変換して、内容を確認すると以下の結果になる。



最大変位は、「8.062e-5」となっている。

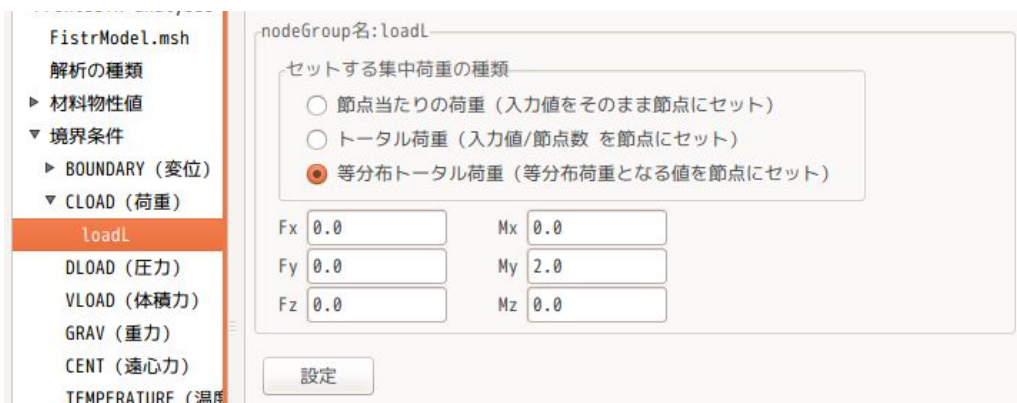
4-15-1-7. 結果の検証

片持ち梁のモーメントによるたわみは、以下で表され、計算すると「7.90e-5」となり、ほぼ一致する。

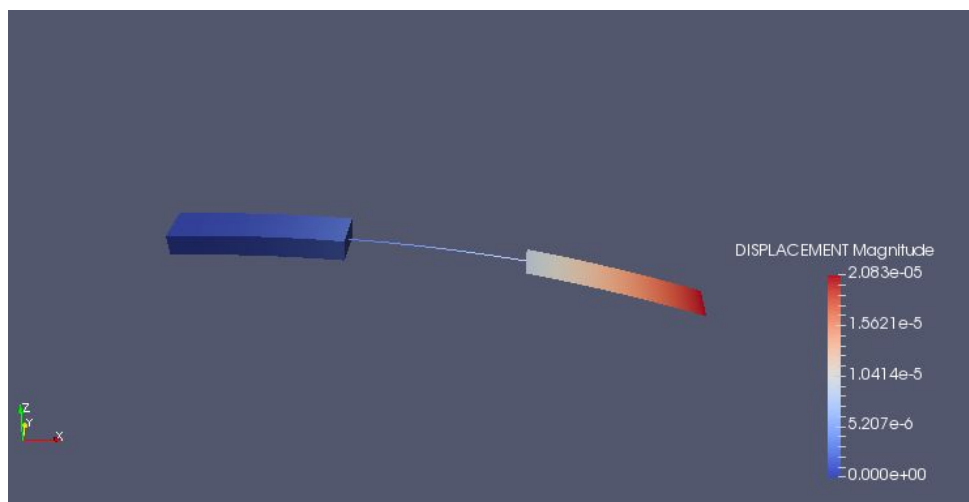
$$\delta = \frac{Pl^3}{3EI} = \frac{10 \times 1.2^3}{3 \times 70e9 \times 1.041e-6} = 7.90e-5$$

4-15-1-8. シェルにモーメントを負荷した場合

シェルの先端 (loadL) に y 軸回りにモーメントを負荷して、たわみの計算を行ってみる。修正点は、CLOAD (荷重) の部分のみであり、この部分を以下の様に修正する。



この修正を加えて、同様に計算した結果が、以下になる。モーメントを加えて曲げる計算ができており、最大変位は、「2.083e-5」となっている。



梁の先端に、曲げモーメントを加えた時の最大変位は、下式で与えられ、計算すると「1.976e-5」であり、ほぼ計算結果と合致している。

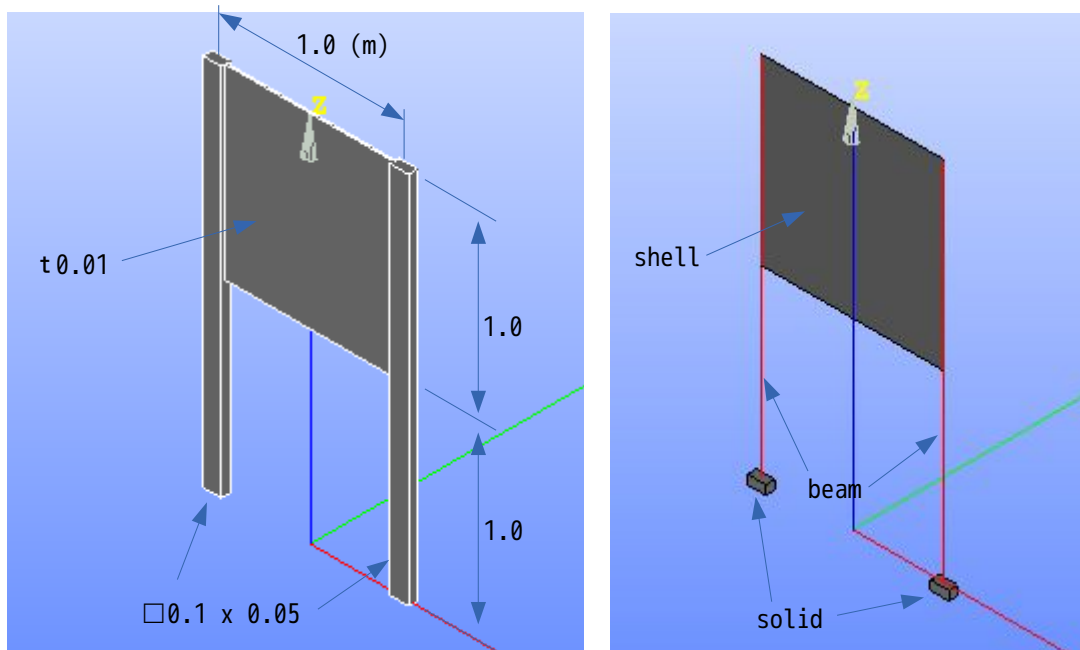
$$\delta = \frac{Ml^2}{2EI} = \frac{2.0 \times 1.2^2}{2 \times 70 \times 10^9 \times 1.041 \times 10^{-6}} = 1.976 \times 10^{-5}$$

4-15-2. シェルとビームのモデルの場合

現在のFrontISTR (ver4.5) の仕様では、シェルとビームのみの混在モデルの計算ができないので、この様な場合は、意識的にソリッド部を作成して、解析する事になる。

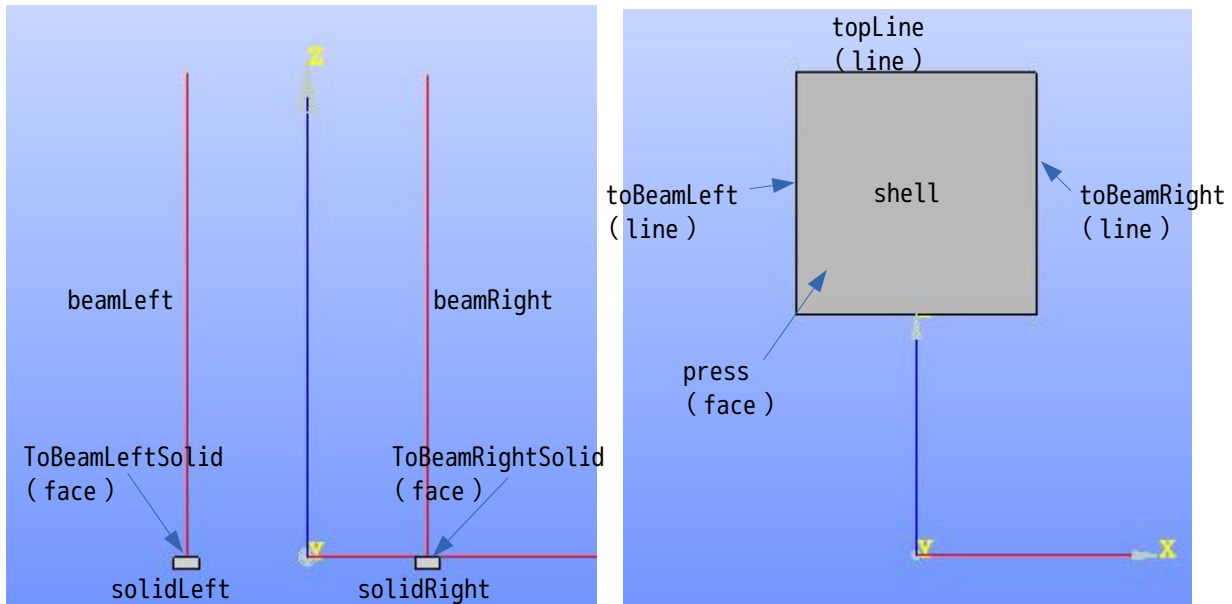
4-15-2-1. モデルの作成

下図の様な、ビームとシェルで表現できるモデルを考える。beamの下部に意識的に solid を付加している。



各部のグループ化は、下図で行っている。

group 名	geometry	mesh	備考
solidLeft	volume	solid 要素	
solidRight	volume	solid 要素	
fixLeftSolid	face	nodeGroup	
fixRightSolid	face	nodeGroup	
toBeamLeftSolid	face	nodeGroup	solid 側の beam 接続部
toBeamRightSolid	face	nodeGroup	↑
shell	face	shell 要素	
press	face	nodeGroup	荷重 (等分布) 印加部
beamLeft	line	beam 要素	
beamRight	line	beam 要素	
toBeamLeft	line	nodeGroup	shell 側の beam 接続部
toBeamRight	line	nodeGroup	↑
topLine	line	nodeGroup	

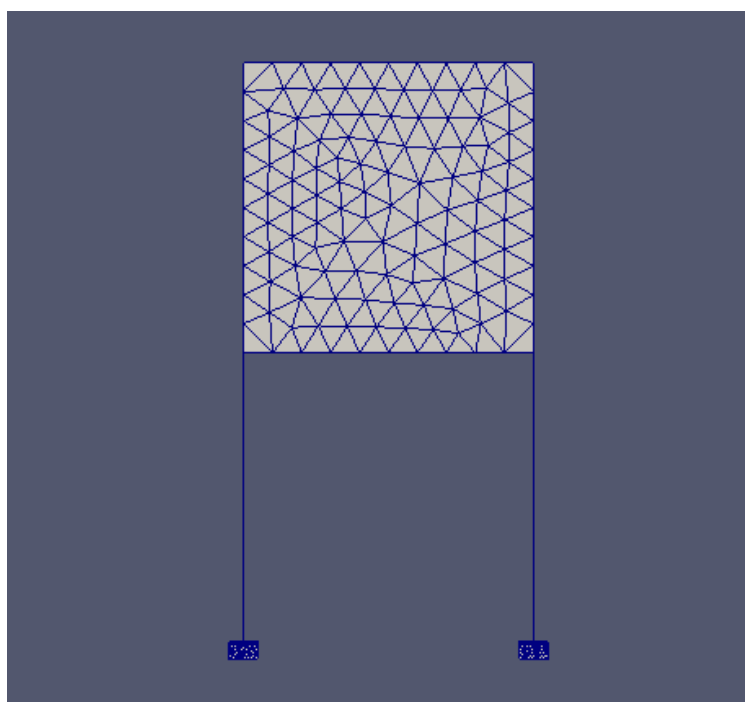
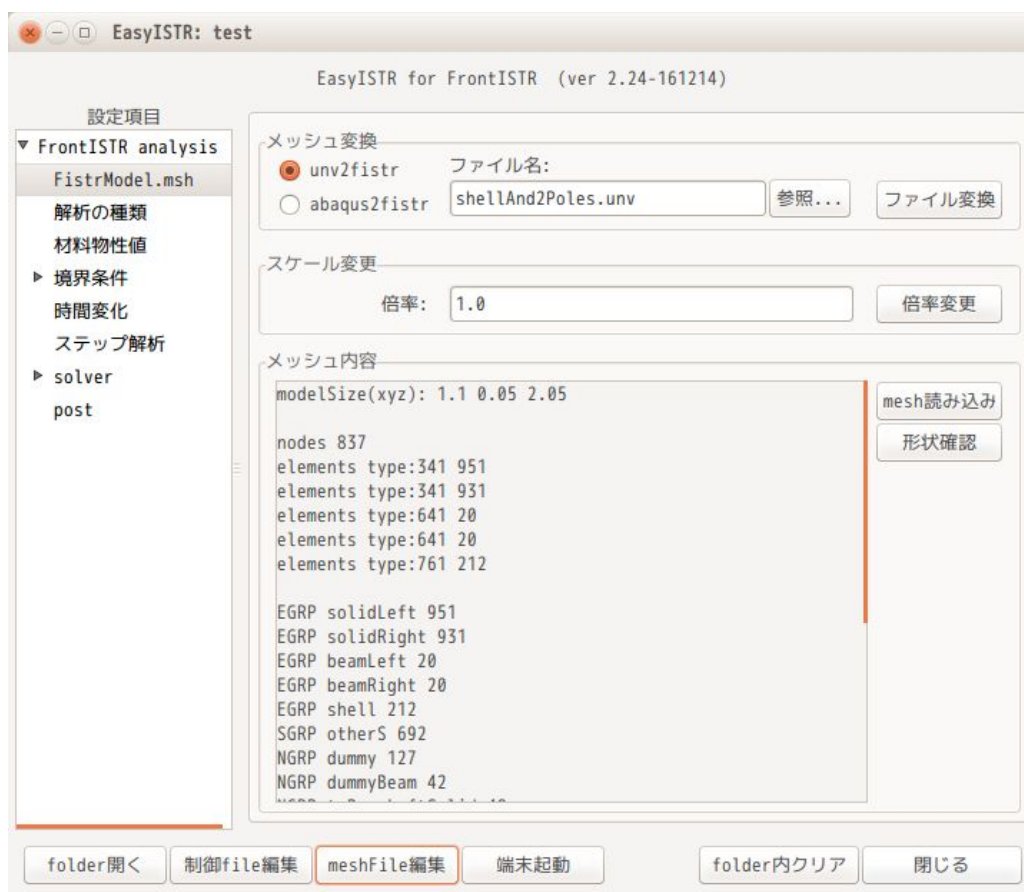


このモデルを salome で作成し、solidは0.0125サイズの tetra、shellは0.1サイズの tri、beamも0.1サイズでメッシュを切る。各々のメッシュを compound して1ヶのメッシュとし、「shellAnd2Poles.unv」名で保存する。尚、compound 時に、各メッシュが節点を共有する様に tolerance を設定して、merge している。

解析は、各 solid の底面を固定して、shell 表面 (press) に荷重をかけてみる。

4-15-2-2. メッシュ変換

保存されている「shellAnd2Poles.unv」を FrontISTR 用にメッシュ変換する。変換した結果が、以下になる。この変換により、beam と solid の接続部には、!EQUATION が作成される。beam と shell の接続部は、今回の場合、!EQUATION を作成する必要がなく、dummy 節点を共有させるだけで済むので、!EQUATION は作成されない。



4-15-2-3. 解析の種類

解析の種類は、「線形弾性静解析」を選択する。solid、shell、beamの混在モデルの場合は、これしか選

択肢がないので、「線形弾性静解析」を選択。

4-15-2-4. 材料物性値

beamの場合は、断面形状が 0.1 x 0.05 の為、前項と同じ形状。shellは、板厚 t0.01。
材料名は、いずれも Al として設定する。
以下の様に設定した。

< beamLeft >	beamの設定 参考軸方向 断面積 断面2次モーメント ねじり定数 vx: 0.0 area: 0.005 Iyy: 1.041e-6 Jx: 5.207e-6 vy: 1.0 Izz: 4.166e-6 vz: 0.0
< beamRight >	beamの設定 参考軸方向 断面積 断面2次モーメント ねじり定数 vx: 0.0 area: 0.005 Iyy: 1.041e-6 Jx: 5.207e-6 vy: 1.0 Izz: 4.166e-6 vz: 0.0
solution ▶ < shell > 時間変化 ステップ解析	板厚の設定 (shell) 板厚: 0.01 厚さ方向積分点数: 5
解析の種類 ▼ 材料物性値 beamLeft beamRight shell solidLeft solidRight	材料物性値の設定 elGroup名:solidLeft 材料名: Aluminum 物性値の確認 材料DBを開く 材料物性値 材料モデル: ELASTIC 塑性(plastic)data 降伏条件/タイプ: SS_data 硬化則: 作成・編集
解析の種類 ▼ 材料物性値 beamLeft beamRight shell solidLeft solidRight	材料物性値の設定 elGroup名:solidRight 材料名: Aluminum 物性値の確認 材料DBを開く 材料物性値 材料モデル: ELASTIC 塑性(plastic)data 降伏条件/タイプ: SS_data 硬化則: 作成・編集

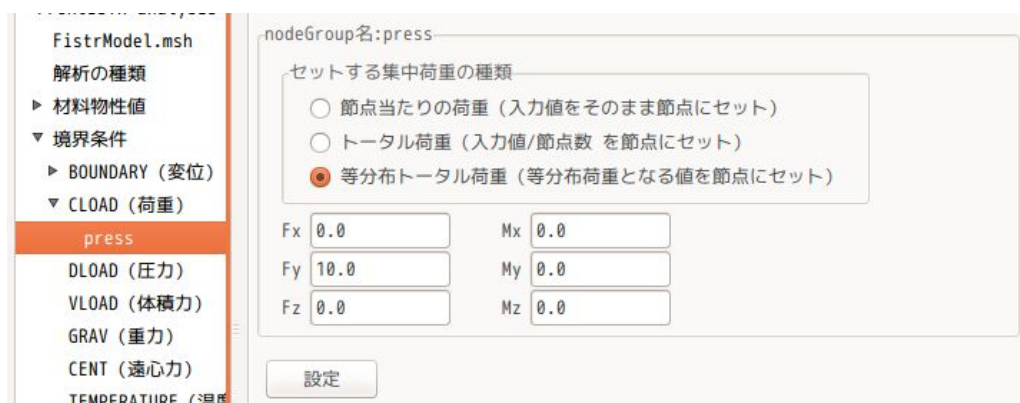
4-15-2-5. 境界条件

境界条件は、solidの底面 (fixLeftSolid, fixRightSolid) を固定し、shellの表面 (press) に等分布荷重 10N を負荷する。

まず、BOUNDARYは、以下の様に設定。



荷重は、以下の様に press 面の Y 軸方向に 10N の等分布荷重を設定した。



4-15-2-6. 計算開始、結果の確認

混在モデルの場合、計算時の線形 solver は、「MUNPUS」or「DIRECT」しか選択肢がないので、「MUMPUS」に設定し、計算させた。

計算終了後、データ返還後、paraView で結果を確認する。以下が、この結果になる。

