EasyISTR5　操作マニュアル

Ver 3.32.220812

EasyISTR5 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 を実行して計算。　計算結果を paraView で確認する方法をとっている。

ver-3.30 以降から、国際化に対応させ、日本語 <-> 英語表記の選択を可能にさせている。

22/08/12　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、 この内容を操作する事に様に修正。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。
ver-3.10.200413	FrontISTR ver 5.0 専用に書き換え。 作業 folder の移動、確認が容易にできる様に、directoryTree を追加。 vtk を使ったモデル形状確認用の viewer を追加。
ver-3.20.200702	vtk の viewer 上で境界条件の設定場所（NGRP, SGRP, EGRP）が確認できる様に修正。 従来の FrontISTR ver 4.6 も使用できる様に修正。
ver-3.21.200804	vtk の viewer 上に 2 次要素が正しく表示されなかった事を修正。 設定ファイル usingApp が EasyISTR 上で編集可能。 usingApp の設定項目に使用する CAD と mesher を追加。
ver-3.21.200816	部品の結合を追加

ver-3.23.201113	節点・面 group の作成、節点・面・要素 group 名変更、削除を追加。 2 次要素の時、SGRP の vtk 表示（edge 表示）が正しく表示されていない事を修正。 mesh 変換時、変換前の設定を残したままの変換ができる様に修正し、全境界条件の一括クリアを追加。
ver-3.24.210117	要素 group の作成を追加。 部品の結合について nodeToFace を追加。 vtk 表示画面中に表示している要素 group 名を表示。
ver-3.25.210218	beam 要素の方向、断面定数の取得方法変更。 梁の接続部を pin 接続に変更して、トラス構造の解析を可能にした。
ver-3.26.210413	node, surface, element の各 group の作成を充実。 各 group の編集（reneme、削除）画面を追加。
ver-3.27.210510	beam 要素の断面形状に C 型、U 型断面を追加。ねじり定数算出方法を修正 画面構成修正、HDD 残り容量表示、結果 file 数表示を追加。
ver-3.28.210608	FrontISTR-5.2 に対応。plotStepMonitor の起動方法を修正。
ver-3.29.210719	dirTree の key 操作対応、shell の SGRP 作成追加。
ver-3.30.220324	国際化対応（日本語、英語表記の選択が可能）
ver-3.31.220420	存在する NGRP,SGRP,EGRP からマウスで pick して取得した grp を加算、減算を追加。 EasyISTR の GUI に log 表示を追加。 非線形材料データ（PLASTIC,BILNEAR や HYPERELASTIC 等）の扱いを追加。
ver-3.32.220501	thread 並列を追加。
ver-3.33.220812	widows 用に新しく easyistrPython_2208.zip（EasyISTR, python3, pyGObject, vtk 含む）を作成。 自動増分の初期値を修正。beam の断面 2 次モーメント算出時の YZ 軸入れ替え。

目次

1.　EasyISTR とは.....6
1-1.　概要.....6
1-2.　起動画面.....6
2.　インストール.....7
2-1.　linux へのインストール.....7
2-2.　windows へのインストール.....8
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
3-4.　解析の種類を設定.....17
3-5.　材料物性値の設定.....18
3-6.　境界条件の設定.....21
3-6-1.　固定部の設定.....21
3-6-2.　節点荷重の設定.....23
3-6-3.　面圧力の設定.....23
3-6-4.　境界条件の再確認.....24
3-6-5.　等分布節点荷重を設定する場合.....24
3-7.　solver の設定.....25
3-7-1.　線形 solver の設定.....25
3-7-2.　出力項目の設定.....26
3-7-3.　計算開始.....27
3-7-4.　ver4.6 を使って計算開始.....28
3-8.　結果の可視化.....29
3-8-1.　FrontISTR の出力結果を可視化.....29
3-8-2.　主応力、主ひずみ、mises 応力、mises ひずみを追加する場合.....32
3-9.　並列処理方法.....36
3-9-1.　並列処理設定、計算開始.....36
3-9-2.　並列計算結果の可視化.....37
3-10.　EasyISTR のメニューバー、ツールバーの動作.....37
3-10-1.　「folde 開く」ボタンの動作.....38
3-10-2.　「端末起動」ボタンの動作.....38
3-10-3.　「閉じる」ボタンの動作.....39
3-10-4.　「temp 開く」ボタンの動作.....39
3-10-5.　「保存（tmp → dir）」ボタンの動作.....39
3-10-6.　「読み込み（dir → tmp）」ボタンの動作.....39
3-10-7.　「cnt,msh ファイル編集」ボタンの動作.....39
3-10-8.　「folder クリア」ボタンの動作.....40
3-10-9.　「cnt, dat ファイル初期化」ボタンの動作.....41
3-10-10.　「CAD 起動」ボタンの動作.....41
3-10-11.　「mesher 起動」ボタンの動作.....41
3-11.　ファイル変換について.....41
3-11-1.　unv2fistr 変換（unv 形式のメッシュを FrontISTR 用に変換）.....41
3-11-2.　abaqus2fistr 変換（abaqus 用 inp ファイルを FrontISTR 用に変換）.....42
3-11-3.　fistr2vtk 変換（FrontISTR 用のファイルを paraView 用に変換）.....44
3-11-3-1.　メッシュファイルの変換.....44
3-11-3-2.　結果ファイルの変換.....46
3-12.　salome によるメッシュの作成方法.....46
3-12-1.　ジオメトリの作成.....46
3-12-2.　メッシュの作成.....48
3-13.　等分布節点荷重の算出方法.....52
3-13-1.　面の等分布節点荷重の算出.....52
3-13-2.　線分の等分布節点荷重の算出.....54
3-13-3.　点の等分布節点荷重の算出.....55
3-13-4.　等分布トータル荷重の設定結果の確認.....55
3-14.　部品の結合.....56
3-14-1.　節点を相手 face 上に結合.....57
3-14-2.　節点同士を結合.....62
3-15.　group の作成、名称変更、削除.....65
3-15-1.　節点(NGRP)、面(SGRP)、要素(EGRP) group の作成.....66
3-15-2.　節点(NGRP)、面(SGRP)、要素(EGRP) group の名称変更、削除.....73
3-16.　その他.....74
3-16-1.　plotStepMonitor について.....74
3-16-2.　複数 step 解析について.....76
3-16-3.　反力の算出について.....79

3-16-4.　大規模メッシュ作成（既存メッシュの細分化）.....80
3-16-5.　複数の要素タイプが混在するモデルについて.....81
4.　応用事例.....83
4-1.　接触解析.....83
4-1-1.　モデル形状.....83
4-1-2.　変位拘束の接触解析.....84
4-1-3.　荷重拘束の接触解析：弱いばね追加.....92
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：自動時間増分で解析.....95
4-1-5.　多数の接触部が存在する場合の接触解析.....102
4-2.　弾塑性解析.....113
4-2-1.　モデル形状（円柱）.....113
4-2-2.　圧縮 20%の計算.....113
4-2-3.　引き続き引張 20%の計算.....119
4-2-4.　複数 step による一括解析.....121
4-3.　固有値解析.....128
4-3-1.　モデル形状.....128
4-3-2.　固有値解析の開始.....129
4-4.　周波数応答解析.....132
4-4-1.　モデル形状.....132
4-4-2.　周波数応答解析の開始.....132
4-5.　時刻歴応答解析（陽解法）.....136
4-5-1.　モデル形状.....136
4-5-2.　境界条件一定値（線形・陽解法）.....136
4-5-3.　境界条件に時間変化を与える（線形・陽解法）.....141
4-6.　時刻歴応答解析（陰解法）.....146
4-6-1.　モデルと材料.....146
4-6-2.　境界条件一定値（線形・陰解法）.....146
4-6-3.　境界条件に時間変化を与える（線形・陰解法）.....148
4-7.　接触を含む時刻歴応答解析.....153
4-7-1.　モデル形状.....153
4-7-2.　設定内容.....154
4-7-3.　計算開始、結果の確認.....156
4-8.　接触を含む弾塑性解析（板のプレス曲げ加工）.....158
4-8-1.　モデル形状.....158
4-8-2.　設定内容.....159
4-8-3.　計算開始、結果の確認.....165
4-9.　熱応力解析.....167
4-9-1.　モデル形状.....167
4-9-2.　熱応力解析の開始.....168
4-10.　熱伝導解析（静解析）.....171
4-10-1.　モデル形状.....171
4-10-2.　熱伝導解析（静解析）の開始.....171
4-10-3.　熱伝導解析結果から熱応力を求める.....174
4-11.　熱伝導解析（動解析）.....176
4-11-1.　モデル形状.....176
4-11-2.　熱伝導解析（動解析）の開始.....176
4-11-3.　熱伝導解析結果から熱応力を求める.....177
4-12.　シェルの解析.....181
4-12-1.　モデルの作成、メッシュ読み込み.....181
4-12-2.　材料の設定.....182
4-12-3.　境界条件の設定.....182
4-12-4.　計算開始、結果の確認.....183
4-13.　ソリッドとシェルの混在モデル解析.....185
4-13-1.　モデル形状.....185
4-13-2.　メッシュ変換.....185
4-13-3.　材料、境界条件の設定.....187
4-13-4.　計算開始、結果の確認.....188
4-14.　梁（beam）要素の解析.....189
4-14-1.　梁のたわみ解析.....189
4-14-2.　三角すい構造の解析（beam の参考軸方向の取得方法）.....193
4-14-3.　梁構造（ラーメン構造）解析.....196
4-14-4.　梁構造（トラス構造）解析.....198
4-14-5.　梁の応力解析.....202
4-14-6.　梁の非線形解析（大変形）.....205
4-15.　ビーム、シェル、ソリッド混在モデルの解析.....206
4-15-1.　梁のたわみ解析（ビーム、シェル、ソリッド）.....207
4-15-2.　つり橋の解析（ビーム、ソリッド）.....210

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、EasyISTR5 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
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
VTK （オープンソース）	モデル形状を 3D 表示させる。 インストールする事により、EasyISTR 画面左上部に 3D モデルが表示。 （インストールしなくても、3D 表示されないのみで EasyISTR は使用可。） インストールパッケージが無い場合は、pip でインストールできる。
FrontISTR	solver 部

上記が実現できるように、EasyISTR5 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。）

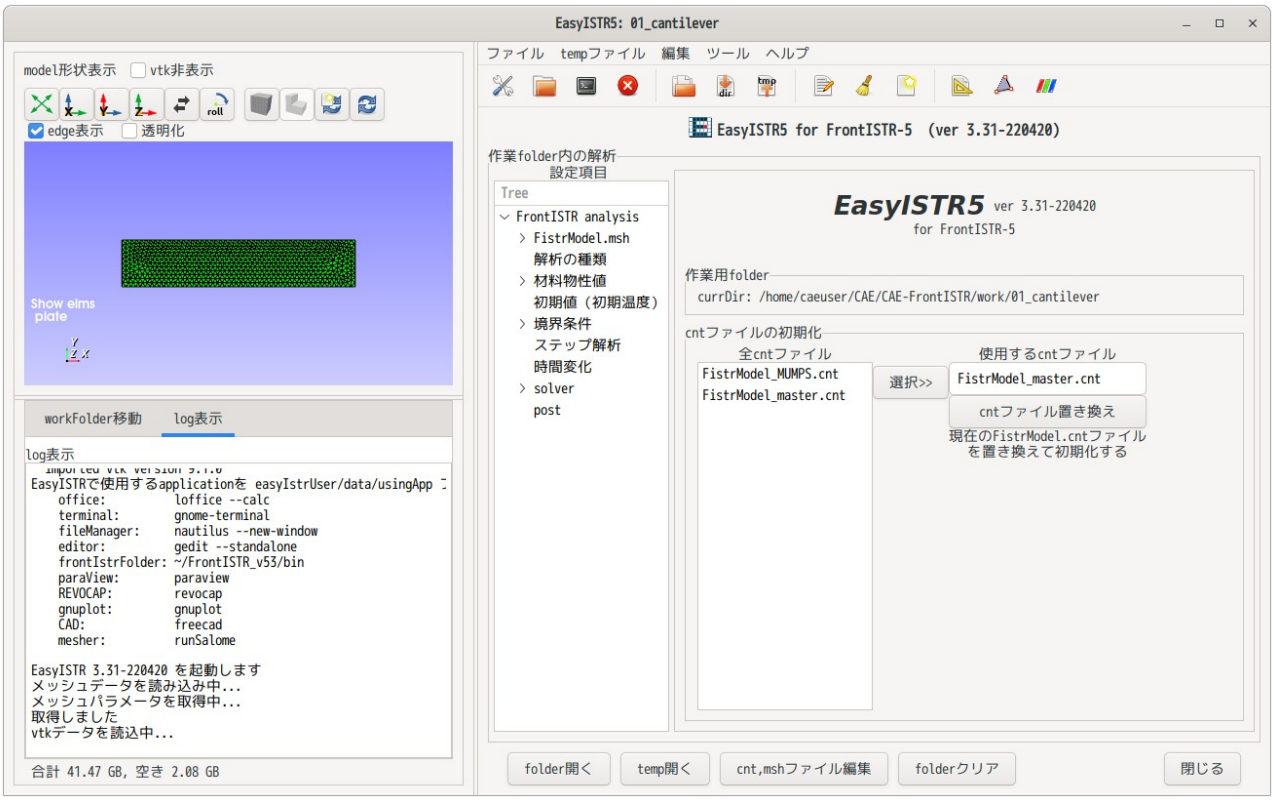
また、大規模メッシュに対応するため、仮の folder 内にファイルサイズを縮小した制御ファイルやメッシュファイルを作り、この folder を対象に操作する様にしている。（ver 2.50 以降）
これにより、サイズの大きいメッシュファイルを直接アクセスする必要がなくなり、操作性を大幅に向上させている。仮の folder は、「easyIstrUser/data/temp」に設定してあり、この folder 内に操作対象 folder 内のファイルイメージを縮小して、そのままコピーされている。

また、国際化にも対応しており、日本語⇔英語への言語の変更が可能になっている。（ver 3.30 以降）

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。
画面左側に sideWindow を設けており、画面の左エッジをドラッグすると、sideWindow 全体が確認できる。
sideWindow 中には、解析中のモデル形状が 3D 表示（VTK により実現）され、directory 又は log が表示される。

3D 表示については、
　　回転：ドラッグ
　　移動：センターホイールでドラッグ
　　拡縮：センターホイール回転
で表示を変える事ができる。



3D viewer には、境界条件を設定する時に選択した group（node, surface, element の group）が赤色で表示されるので、理解しやすい。

また、viewer への表示について、モデルの要素数が増えるに応じて、データ読み込み・表示に時間がかかってしまう。この為、起動時には、制限時間を設定して viewer に表示させているので、制限時間オーバで viewer に表示できずに起動する場合がある。
このような場合に、制限時間を設定せずに（制限時間を 1h に設定している）データを再読み込みして viewer に表示する再読込ボタンを準備しているので、これを使えば表示できる。

また、viewer には、要素グループ毎に表示しており、これらは、要素グループ毎に非表示できる。
大規模メッシュなどでは、vtk を非表示にする事ができる為、これによって操作性が向上する。

上図の sideWindow 下部は、log が表示されている。（ver-3.31 から log 表示を追加。以前は、directoryTree が表示されていた）
従来の directoryTree を表示させるには、「workFolder 移動」タグをクリックする事で directoryTree を表示させる事ができる。
directoryTree 画面上で、作業 folder の場所（解析 case）を移動する事ができる。

2.　インストール

EasyISTR5 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、EasyISTR5 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 系の deb パッケージと RedHat 系の rpm パ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、RedHat 系とも、Gnome デスクトップ環境を想定している。Gnome デスクトップ環境であれば、GTK+3 のライブラリが使える環境なので、インストールは容易。
（Gnome デスクトップ環境以外では、まず GTK+3 が使える環境を整える必要がある。）

＜Debian 系（Ubuntu）＞
```
$ sudo dpkg -i easyistr_3.33.220812_all.deb
```

＜RedHat 系＞
```
$ su
パスワード：
# rpm -i easyistr_3.33.220812-2.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。

ver3.10 以降、vtk が必要になっている。
ubuntu については、vtk の deb パッケージが準備されており、ubuntu-21.04 からは、python3-vtk9 が追加されている。（それ以前は、python3-vtk7 のみ）
python3-vtk7 は、バージョンが ver 7 であり古く、edge が透けて見えるので、裏表が認識しにくい。ver 8.2 以降では、edge が正しく表示されるので、vtk のインストールは、python-vtk9 を推奨する。

vtk のホームページでは、最新版が ver 9.2.0（22/08/12 時点）になっており、pip を使って最新版をインストールすることもできる。（最新の ver 9.2.0 は未確認。ver 9.1.0 は作動確認済。）
pip を使って最新版をインストールする場合は、端末を起動して以下の様に入力する。
（pip は、apt-get 又は synaptic で python3-pip を予めインストールしておく。）
```
$ pip install vtk
```

vtk のバージョンを指定して、vtk-9.0.3 をインストールするには、以下でインストールできる。
```
$ pip install vtk==9.0.3
```

vtk がインストールできたかどうかの確認は、以下で確認できる。
```
$ python3
>>> import vtk
```

2-2.　windows へのインストール

EasyISTR5 は、python3 と PyGObject（GTK+3）で作成しているので、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 windows に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。
この内容に従って、python3 と PyGObject をインストールする。

EasyISTR5 では、さらに vtk も必要なので、引き続き以下を入力して vtk をインストールする。
```
$ pacman -Suy
$ pacman -S mingw-w64-x86_64-vtk
```

以上の操作により、必要な環境が整ったので、以下を入力しインストール状況を確認する。
```
$ python3
>>> import gi
>>> import vtk
```

エラーが発生しなければインストールは正常だが、今回のインストーラ「msys2-x86_64-20220603.exe」では、「import vtk」で以下のエラーが発生する。
```
>>> import vtk
  :
  File "C:/msys64/mingw64/lib/python3.10/site-packages/vtkmodules/vtkIOGDAL.py", line 9, in <module>
    from vtkIOGDALPython import *
ModuleNotFoundError: No module named 'vtkIOGDALPython'
```

本来、エラーは発生しないはずだが、エラーが発生するので、
該当ファイル「C:/msys64/mingw64/lib/python3.10/site-packages/vtkmodules/vtkIOGDAL.py」の 9 行目をコメントアウト、10 行目に「pass」を追加した。

---------- vtkIOGDAL.py（修正前）----------

```
from __future__ import absolute_import

try:
    # use relative import for installed modules
    from .vtkIOGDALPython import *
except ImportError:
    # during build and testing, the modules will be elsewhere,
    # e.g. in lib directory or Release/Debug config directories
    from vtkIOGDALPython import *
-----------------------------------------------------------------
               ↓
------------- vtkIOGDAL.py (修正後)----------------------------------------
from __future__ import absolute_import

try:
    # use relative import for installed modules
    from .vtkIOGDALPython import *
except ImportError:
    # during build and testing, the modules will be elsewhere,
    # e.g. in lib directory or Release/Debug config directories
    #from vtkIOGDALPython import *
    pass
-----------------------------------------------------------------
```

同様なエラーが他でも発生したので、同様な修正を計 4 個のファイルに対して行った。

```
       vtkIOGDAL.py
       vtkIOMySQL.py
       vtkIOPostgreSQL.py
       vtkRenderingQt.py
```

以上の修正により、vtk が import できるようになる。
これで EasyISTR5 が実行できるシステム側の環境が整ったことになる。

この修正により、GDAL、MySQL、PostgreSQL、Qt が vtk 上でデータが読み込めない状態になるが、単独では python 上で使う事ができる。

EasyISTR5 のインストールは、「C:\DEXCS\easyIstrPython」フォルダを作成し、この中に「easyIstr-3.33-220812.zip」を展開してその内容（easyIstr フォルダ）を保存する。

```
       フォルダ構成
       C:\DEXCS
         easyIstrPython
           easyIstr
             easyistr.bat          #EasyISTR の実行ファイル
             easyistrEnv.bat       #EasyISTR の環境設定ファイル
               :
```

EasyISTR を実行する為には、easyistrEnv.bat ファイルを編集して、EasyISTR の環境を設定する必要がある。
修正内容は、windows にインストールした python3 の場所を設定する。

```
---------------- easyistrEnv.bat -------------------------------------
       :
rem #環境変数の設定
set easyIstrPath=%CD%
set HOME=C:\DEXCS
set easyIstrUserPath=%HOME%\easyIstrUser
set binApp=%easyIstrPath%\bin
set pythonApp=%easyIstrPath%\python

rem python3.exe への path を定義
rem   GTK3, pyGObject, VTK がインストールされている事が必要
set pythonGtkVtk=C:\msys64\mingw64\bin
       :
-------------------------------------------------------------------
```

以上の修正を加え、実行ファイル「easyistr.bat」を実行する事で EasyISTR が起動する。

windows へのインストールは、以上の様に、python3、pyGObject、vtk のインストールが必要な為、手続きが複雑になる。この為、これらをインストールした「easyIstrPython_2208.zip」を準備しているので、こ

れを使うとダウンロードして展開するだけで、EasyISTR5 を起動させる事ができる。

その方法は、easyIstrPython_2208.zip を展開した結果（easyIstrPython フォルダ）を D:\DEXCS フォルダ内に保存する。

```
フォルダ構成
C:\DEXCS
  easyIstrPython
    easyIstr
      easyistr.bat            #EasyISTR の実行ファイル
      easyistrEnv.bat         #EasyISTR の環境設定ファイル
    python-3.10.5_withGiVtk   #python3, pyGObject, vtk を含め（msys64 から抜粋）
        :
```

保存後は、実行ファイル「easyistr.bat」を実行すると EasyISTR5 が起動する。

今後の EasyISTR5 のバージョンアップは、前記したフォルダ構成の内の「easyistr」フォルダを新しいバージョンの easyIstr フォルダに入れ替える事で b-ジョンアップに対応できる。

2-3.　起動方法

起動は、easyIstr フォルダ内の以下の実行ファイルを実行すると EasyISTR5 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。EasyISTR5 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。

それでも起動しない場合は、端末上から、実行コマンドを直接入力して、EasyISTR5 を起動してみる。（下記参照）

```
Linux:
        $ easyistr

windows:
        > easyistr.bat
```

端末上で起動すると、端末上にエラーメッセージが出力されるので、エラー内容が確認できる。

2-4.　インストール後の設定

EasyISTR5 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。EasyISTR5 は、起動時にその folder に path を通して起動する。

また、この設定項目には、「language」の項目があり、この内容を「English」に変更する事で、英語表記に変更できる。（linux、windows とも）
言語の設定を適用するには、EasyISTR を閉じて、再起動させた時に適用される。

---------- 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
    CAD         runFreeCAD                        #CAD の起動用スクリプト
    mesher      runSalome                         #mesher 起動用のスクリプト
    language    Japanese                          #使用言語（日本語 or 英語）

#  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\FrontISTRv50\bin          #FrontISTR への path
    paraView    paraview
    REVOCAP     revocap
    gnuplot     "C:\Program Files (x86)\bConverged\gnuplot\bin\gnuplot.exe"
    CAD         C:\DEXCS\FreeCAD\bin\FreeCAD.exe
    mesher      "C:\DEXCS\SALOME-9.9.0\run_salome.bat"
    language    Japanese
-----------------------------------------------------------------------------------------
```

尚、これらの設定は、EasyISTR 起動後、メニューバーで「ファイル」>「設定内容の変更」、または、ツールバーの🛠アイコンをクリックして内容を確認修正できる。（以下の画面上で修正できる。）

3.　操作方法（片持ち梁の線形弾性静解析）

EasyISTR5 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 EasyISTR5 の操作方法について、説明する。

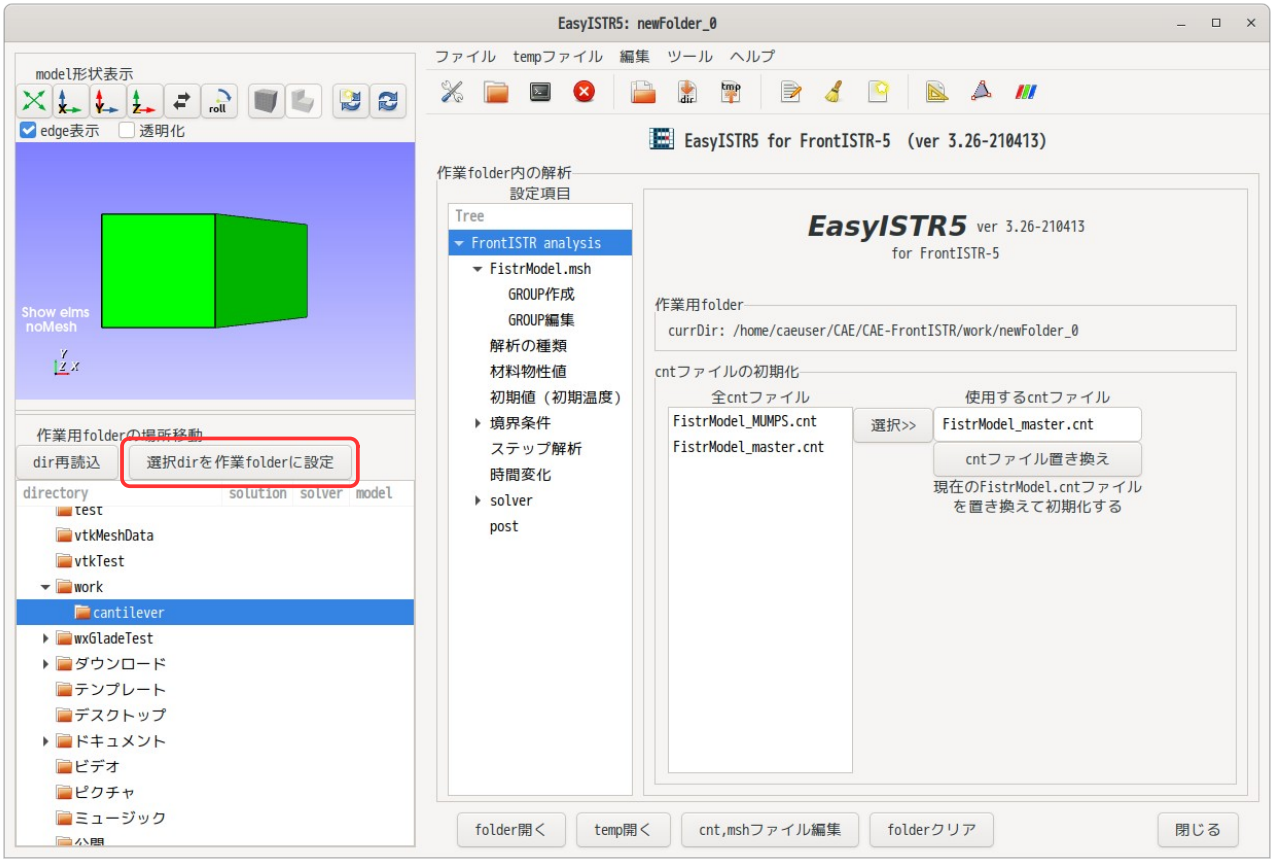
EasyISTR5 の操作方法に関しては、片持ち梁の線形弾性静解析を例にとって説明する。
基本的には、画面上で入力して、最後に「設定」ボタンをクリックして確定させる方法をとっている。この「設定」ボタンのクリックは、その設定内容が制御ファイル等のファイルに書き込まれる事を意味している。

3-1.　workFolder の設定

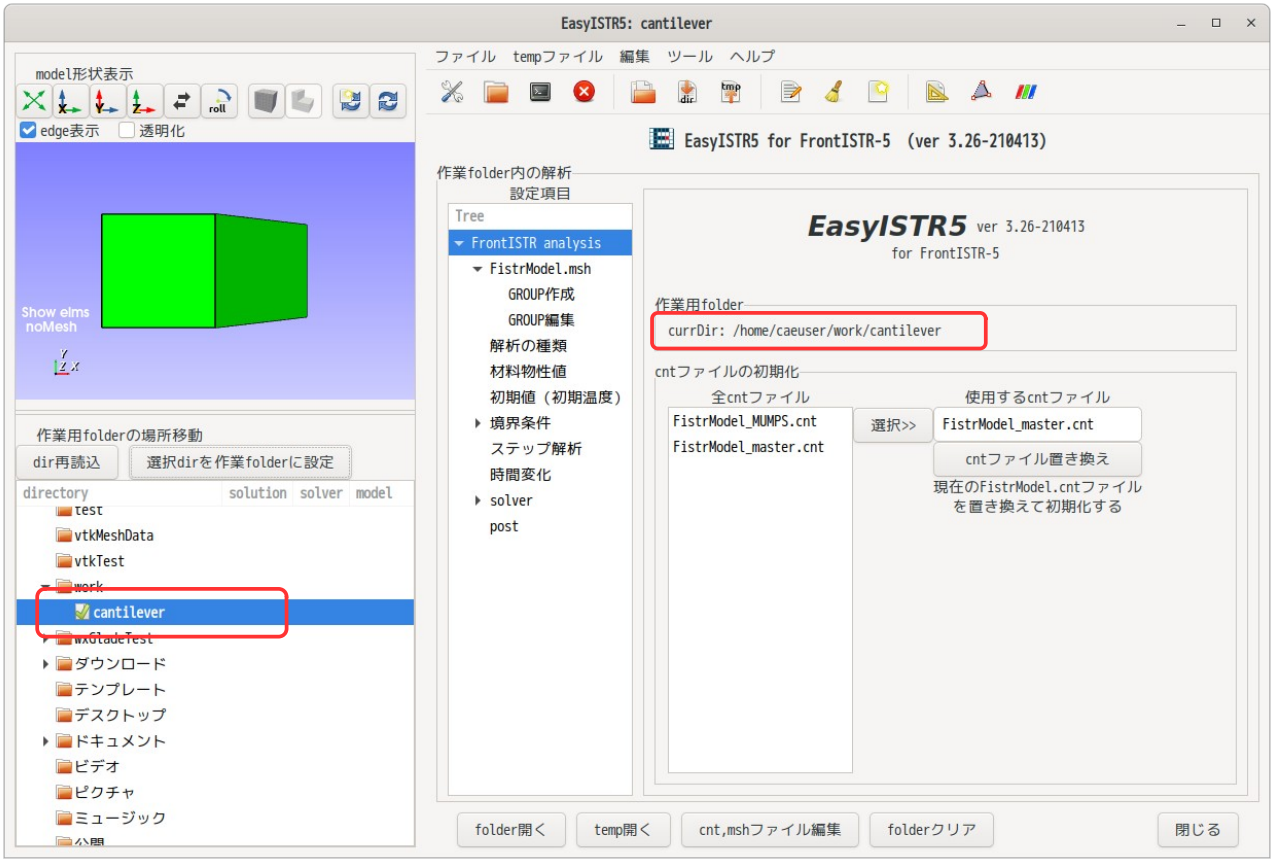
まず、最初にする事は、これから解析する workFolder を指定するところから始まる。
ここでは、「~/work/cantilever」フォルダを新たに作成し、ここで解析する事にする。
この「cantilever」は、folder のみで、この中にファイルが無い状態から始める。

EasyISTR5 の左画面上の directoryTree の該当 folder を選択して「選択 dir を作業 folder に設定」ボタンをクリックして、workFolder「~/work/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、FrontISTR5 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。
（directoryTree の folder 名をダブルクリックするとその folder が開くので、容易に folder 追加が可能）

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



この操作により、以下の画面に変わる。



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。
（「~/easyIsterUser」フォルダは、EasyISTR5 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。

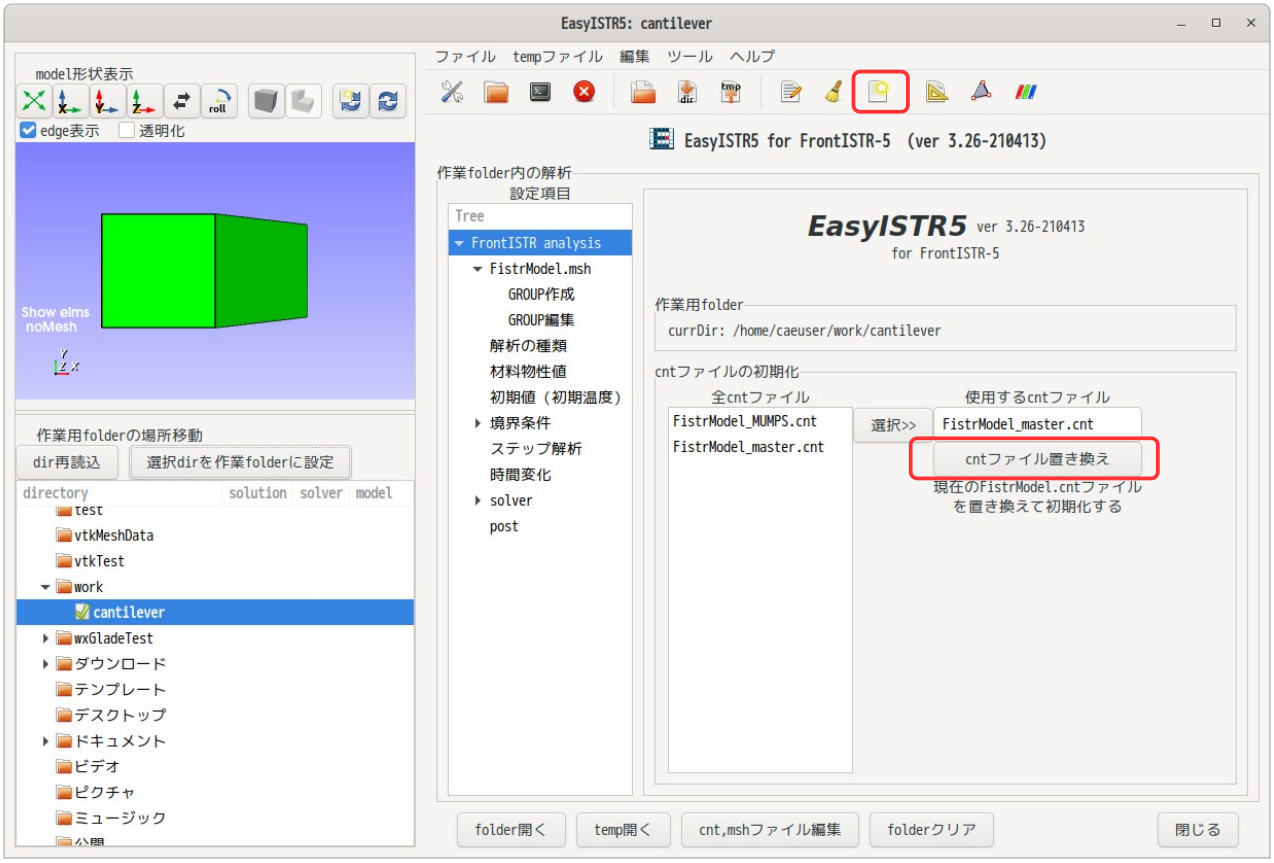
解析の内容によって、制御ファイルの内容は大きく異なってくるが、制御ファイルのひな形が「FistrModel_master.cnt」ファイルで、このファイルが「~/easyIstrUser/data」フォルダ内に保存されている。この「FistrModel_master.cnt」ファイルは、必要最小限の内容の制御ファイルであり、動解析や弾塑性解析の様な複雑な解析では、設定追加の項目が多くなる。
予め、これら解析専用の制御ファイルを「「~/easyIstrUser/data」フォルダ内に別の名前で作成しておくことで、そのファイルを制御ファイルとして設定する事ができ、設定項目を減らす事ができる。

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。

EasyISTR5 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と、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 easyISTR5 起動時にも、その設定が保持される。

尚、「cnt ファイル置き換え」ボタンと同じ操作は、toolBar の「cnt, dat ファイル初期化」アイコンでも行える。



3-2.　メッシュ変換（unv 形式　→　fist 形式）

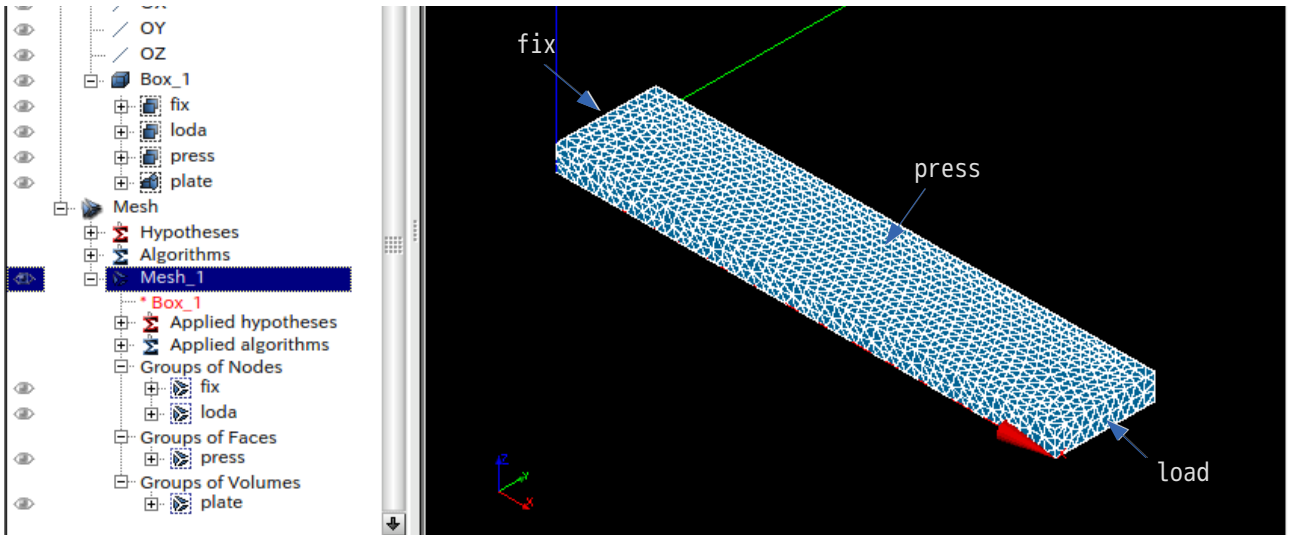
salome で作成した unv 形式のメッシュを FrontISTR が取り扱う形式（fistr 形式）に変換する。

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

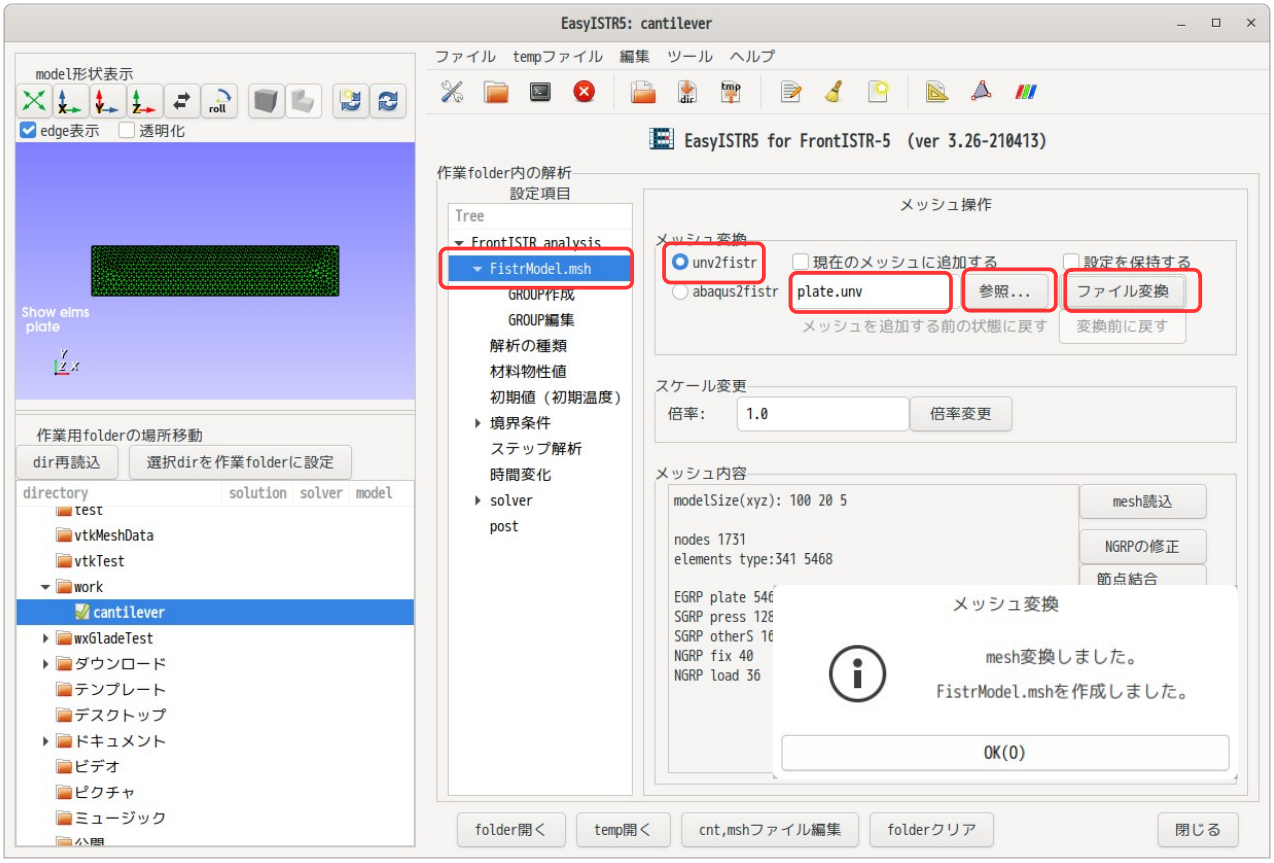
を作成している。

これらグループは、EasyISTR 上で名称変更、削除、作成する事ができる（3-15 項を参照）為、メッシュのみ作成して、必要なグループを EasyISTR 上で作成する事もできる。



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。「plate.unv」ファイルは、EasyISTR5 のインストール folder 内の unvFiles フォルダ内にあるので、これをコピーして使う事もできる。

メッシュ変換の為に、EasyISTR5 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。



尚、「ファイル変換」ボタン上部の「設定を保持する」をチェックすると、変換前の設定を保持した状態で、メッシュ変換する。この為、メッシュ形状を変更して再解析する様な解析では、非常に便利。

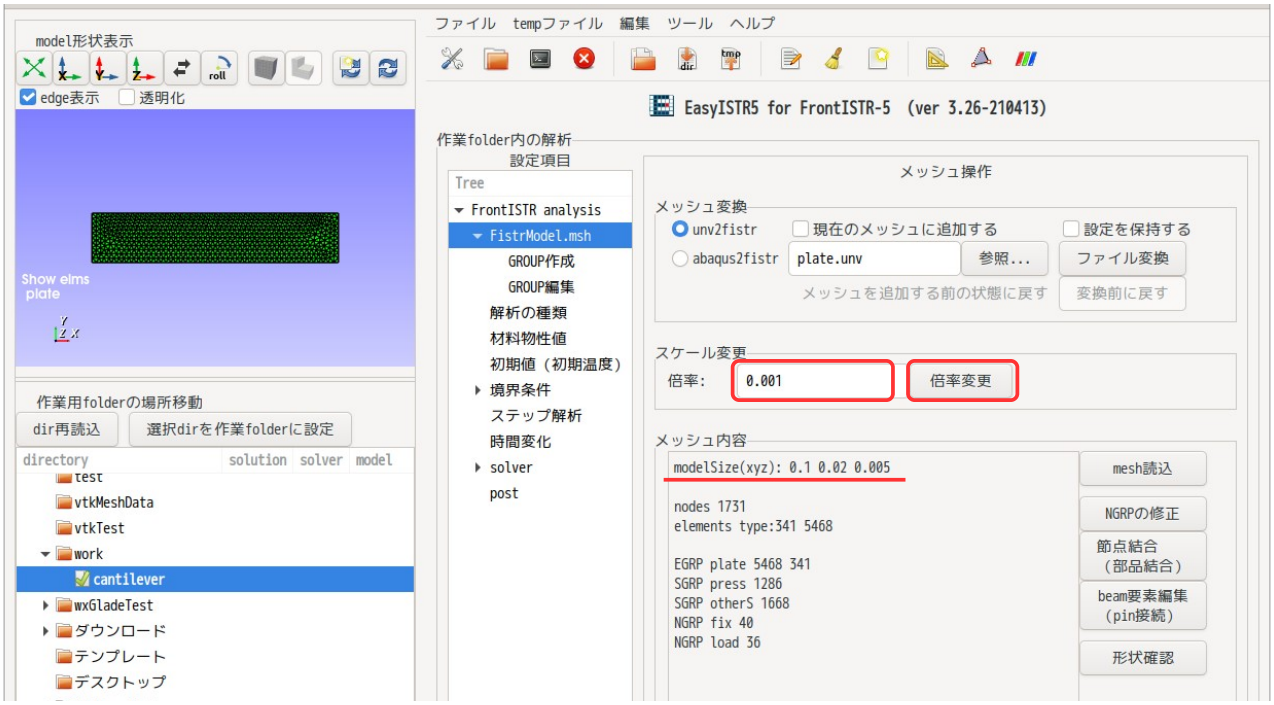
また、ファイル名入力欄上部の「現在のメッシュに追加する」をチェックすると、ファイル変換後、その変

換したメッシュを現在のメッシュに追加する事ができる。この為、複数の部品を扱う接触解析や、部品を結合して解析する場合に利用できる。

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

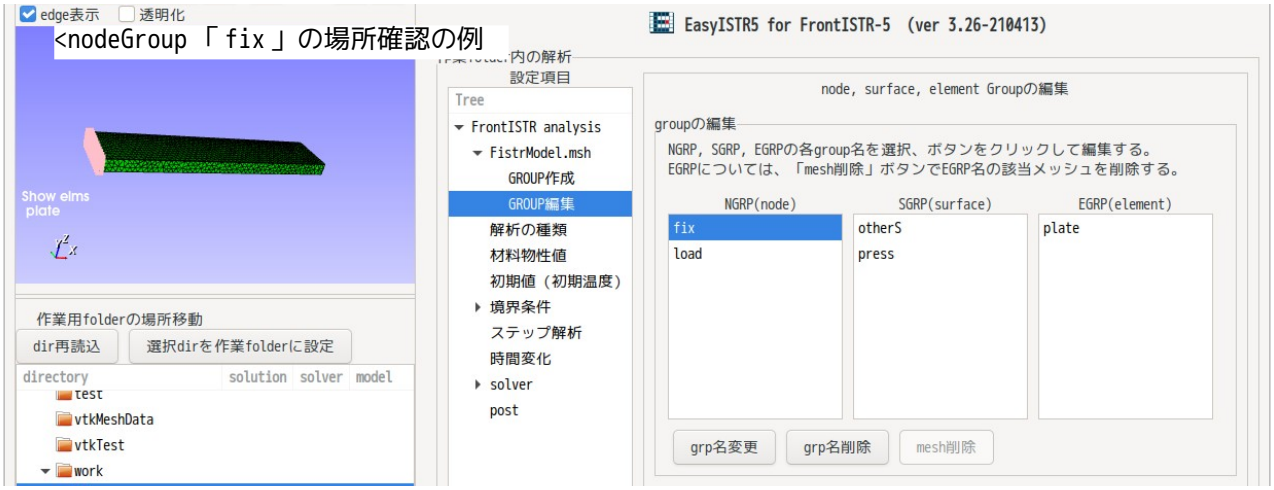
メッシュの概要から、モデルサイズが 100 x 20 x 5 であり、mm 単位で作成されている事が判る。この為、1/1000 に縮小する必要がある。
モデルのスケールを変更するためには、下図の様にスケール変更枠内の倍率を「0.001」に設定し、「倍率変更」ボタンをクリックする事で実現できる。この操作により、modelSize(xyz)の値が 1/1000 の値に修正されている。

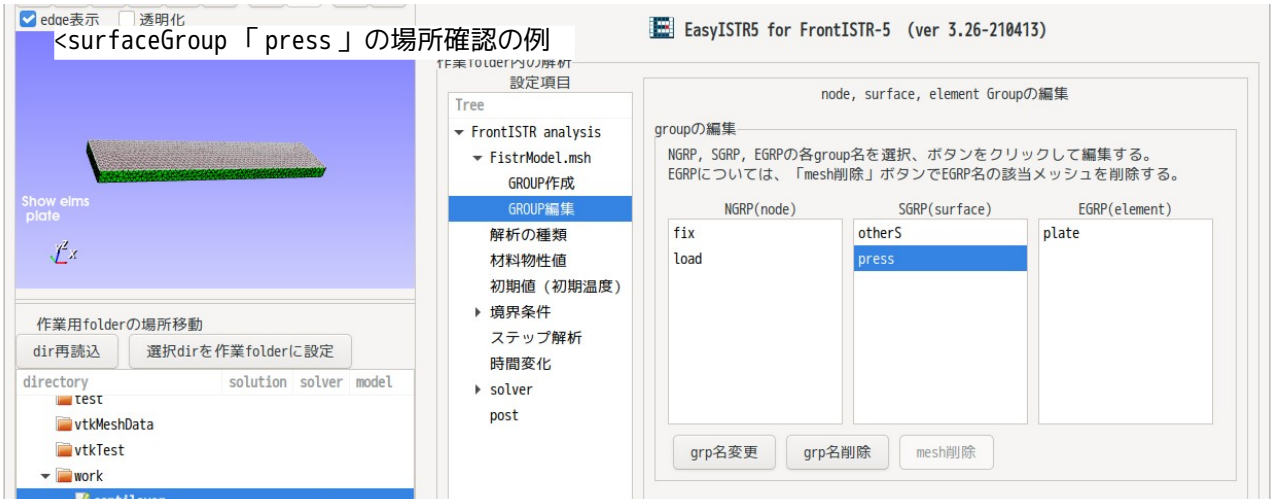


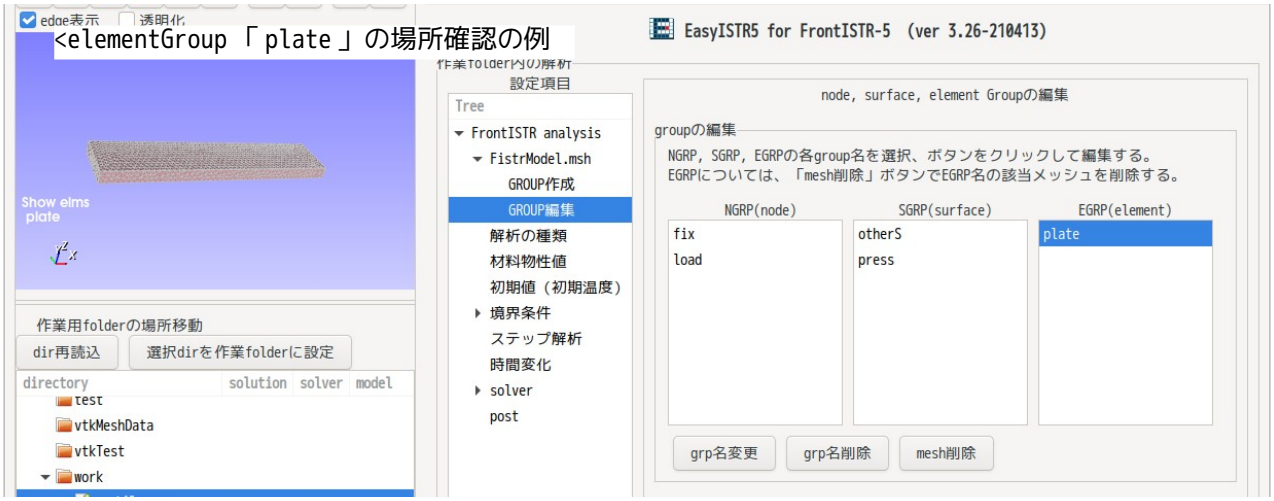
3-3.　メッシュ形状の確認

読み込んだメッシュ形状は、3D viewer に表示されている。この viewer を使って、plate.unv メッシュに定義されている nodeGroup、faceGroup、elementGroup が確認できる。

group 名と場所を確認する方法は、設定項目 Tree の「FistrModel.msh」＞「GROUP 編集」画面を表示する。この画面上には、ここで定義されている NGRP(nodeGroup) 、SGRP(surfaceGroup)、EGRP(elementGroup)名がリストアップされており、場所を確認したい group 名を選択すると、viewer 上にその場所が表示される。以下に、各々の group の場所を確認した例を示している。

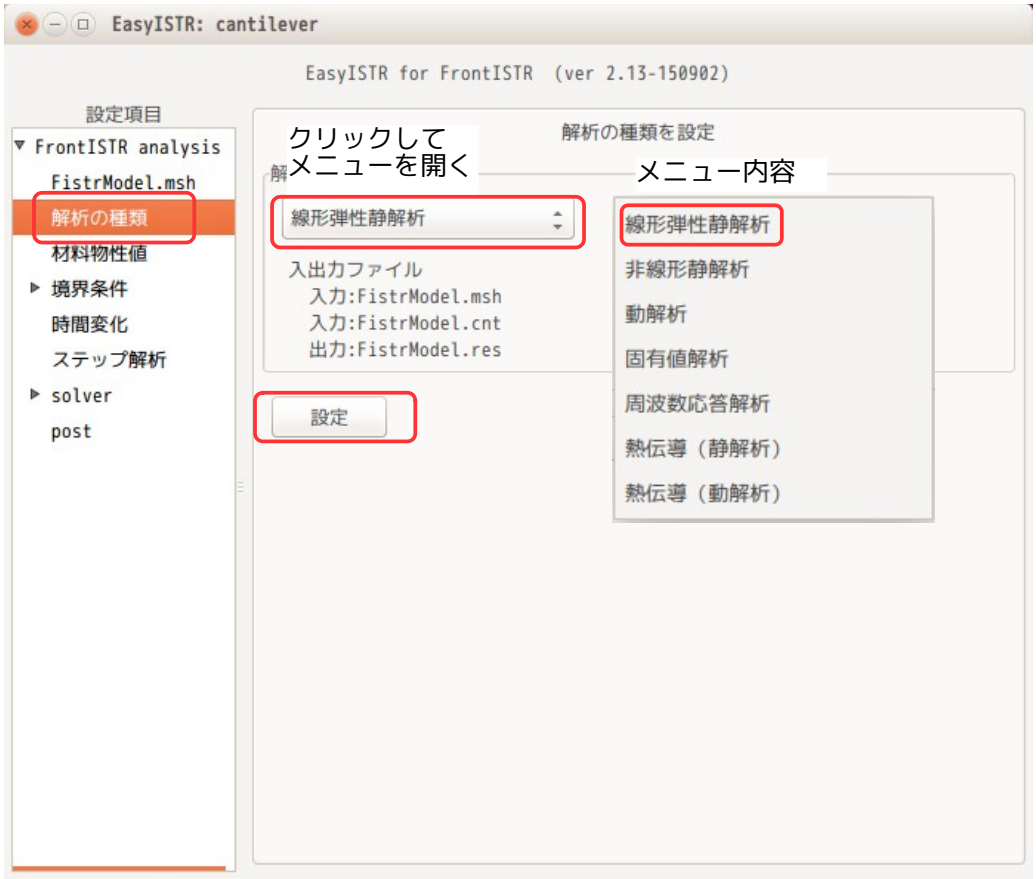






3-4.　解析の種類を設定

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、EasyISR 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、「~/easyIstrUser/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。

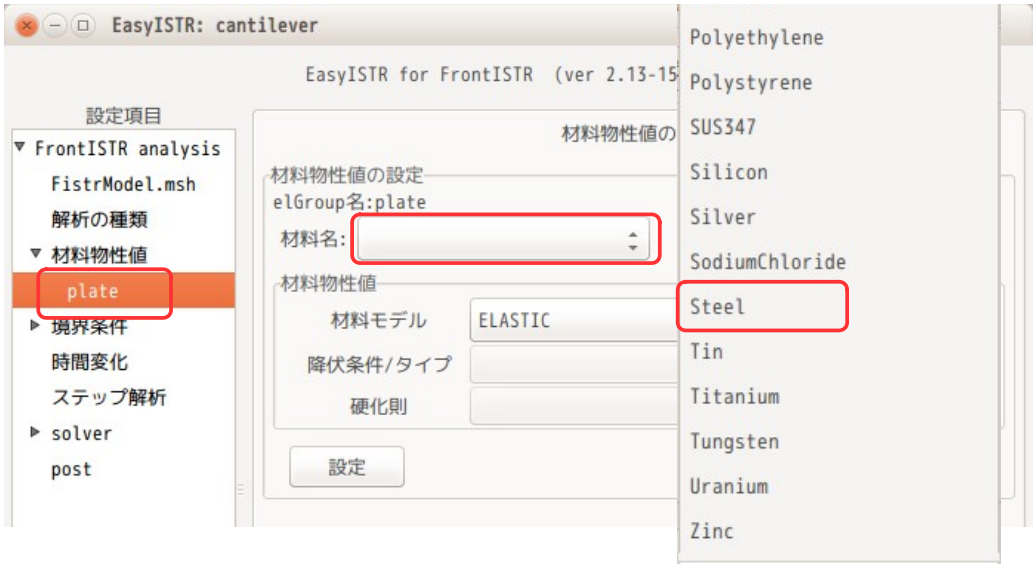
材料 DB を設定した後は、モデルに材料物性値を設定する。
材料の設定は、メッシュの EGRP に設定する。今回のメッシュ内には、EGRP として、plate を定義しているので、この plate に材料を定義することになる。

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、「物性値の確認」ボタンをクリックする事で、設定している材料の物性値が表示されるので、確認できる。「材料 DB 開く」ボタンは、libreOffice で材料 DB が開くので、他の材料を追加、修正する場合に利用できる。

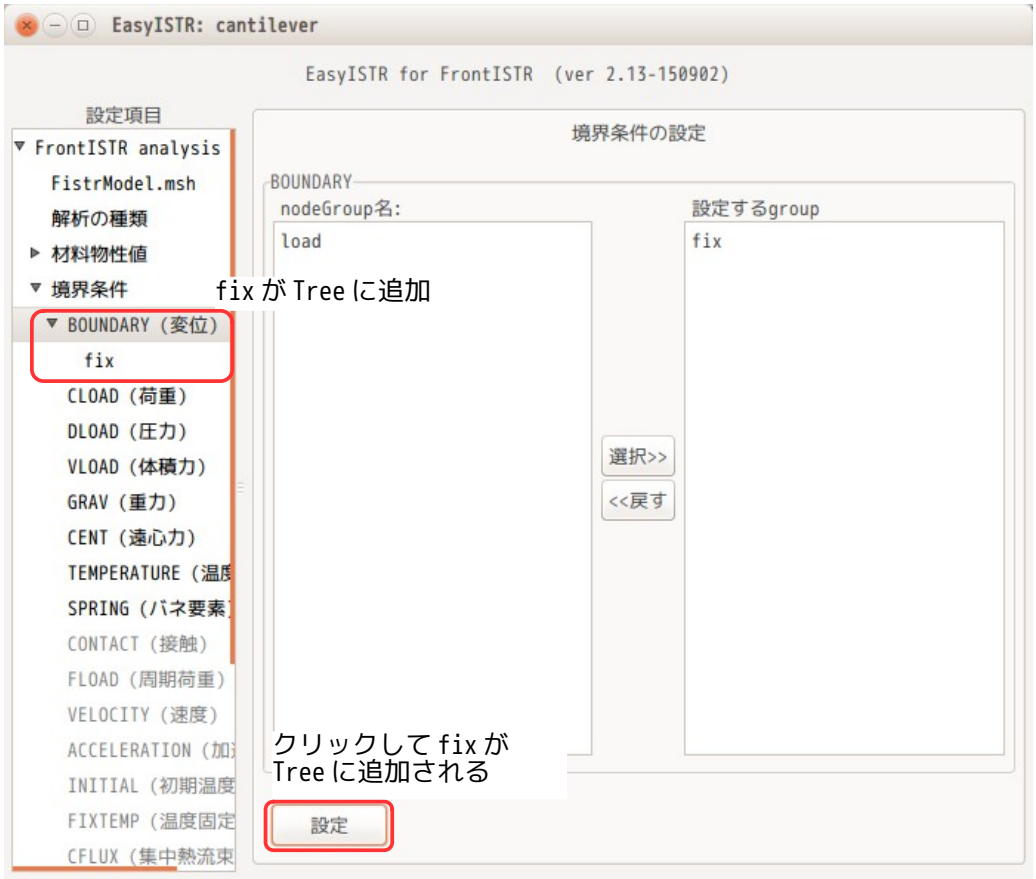
3-6.　境界条件の設定

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、下図の様に設定項目 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（1MPa）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 -100 N）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。

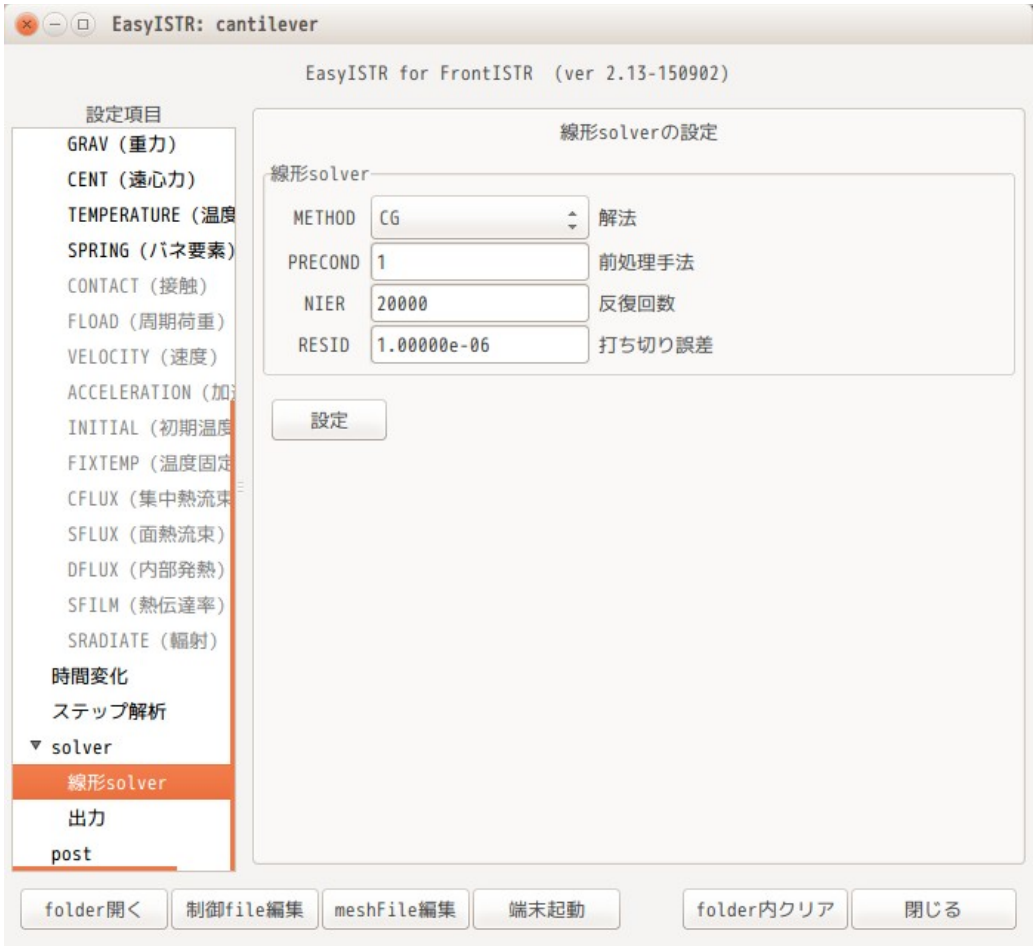
尚、セットする集中荷重の中で、「トータル荷重」は、トータル荷重を入力する事で、「トータル荷重／節点数」の値を全節点に設定するので、等分布荷重にはなっていない。（荷重の分布は、「節点当たりの荷重」と同じ分布になる。）

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。



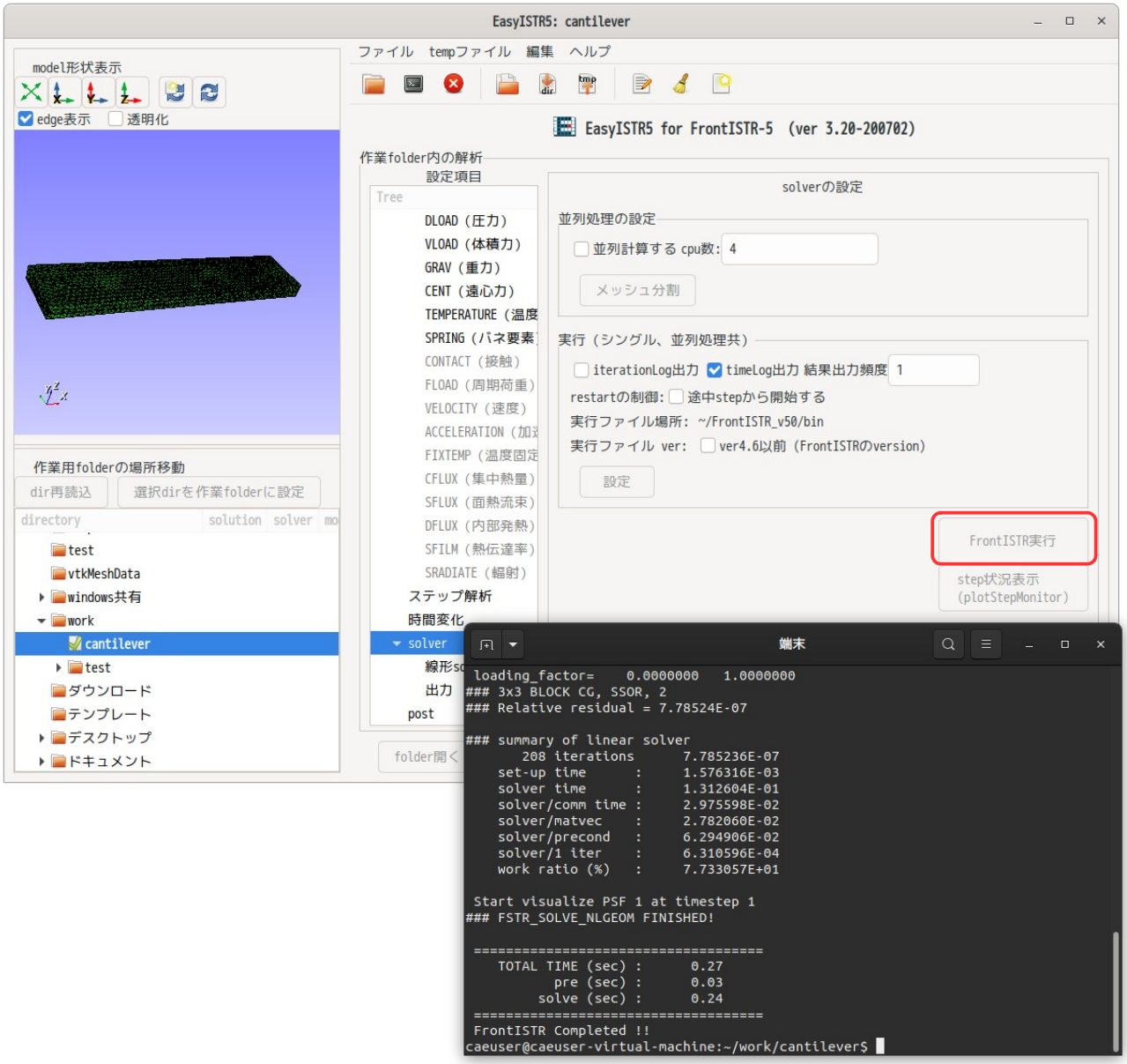
3-7-2.　出力項目の設定

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。

出力項目を確定させる為に、「設定」ボタンをクリックする。

3-7-3.　計算開始

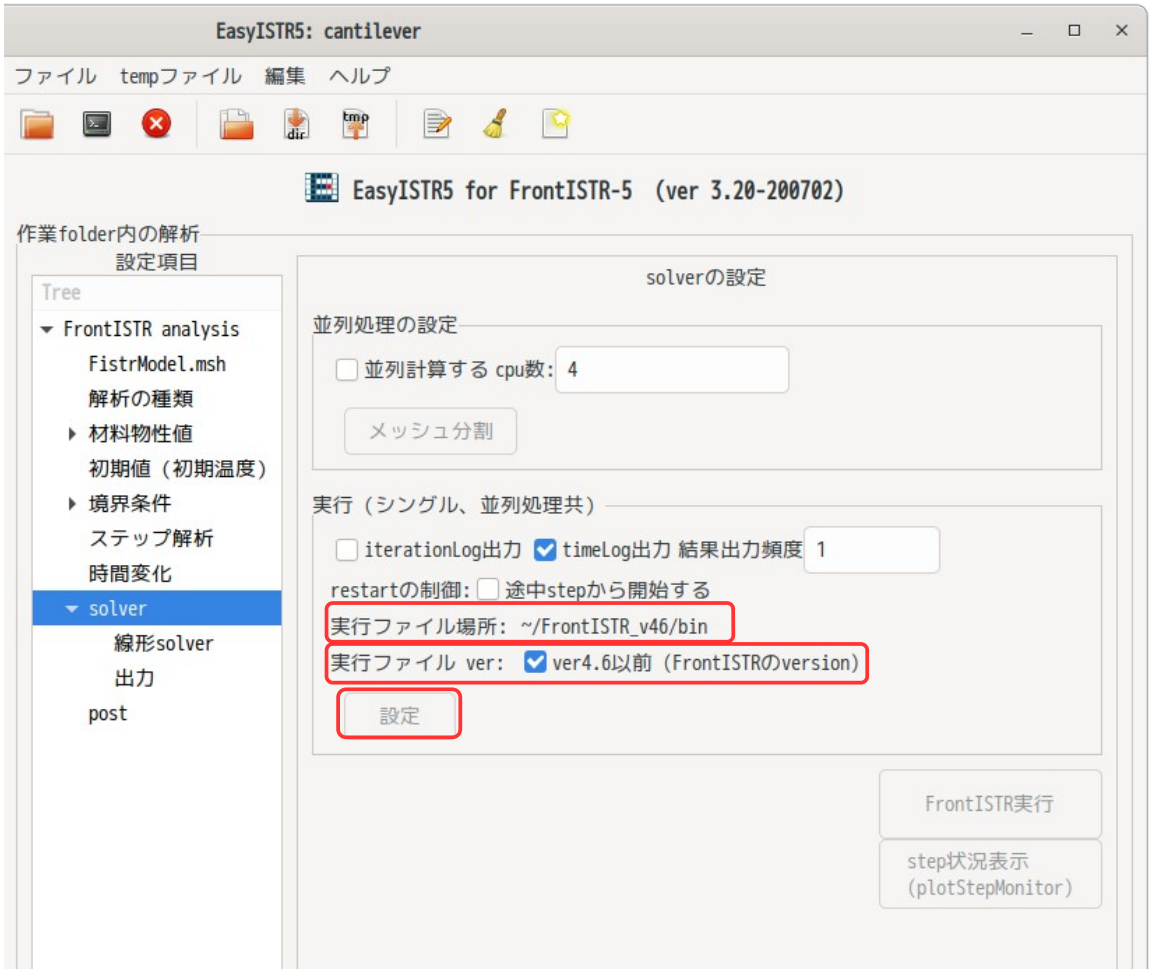
設定項目 Tree 内の「solver」を選択する。log の出力頻度等の設定を確認して、「FrontISTR 実行」ボタンをクリックする。
この操作で、fistr1 が起動し、計算が始まる。計算状況は、端末への出力で確認でき、この出力内容は、FistrModel.log ファイルへも出力されている。



3-7-4.　ver4.6 を使って計算開始

前項の計算は、FrontISTR ver 5.0 を使った計算になる。この計算に使う FrontISTR は、2.4 項で設定した usingApp ファイルで設定している FrontISTR になる。
設定した FrontISTR の version が ver4.6 の場合は、計算結果の出力フォーマットが異なっているので、計算はできるが、結果を見ることができなくなってしまう。この為、ver4.6 で計算する場合は、その処理方法を変更する必要がある。

ver4.6 で計算する時は、まず、usingApp の内容を ver4.6 の FrontISTR に変更する。
この後は、通常通り、材料設定、境界条件を設定し、計算開始する時に、ver4.6 を使う事を知らせる checkBox をチェックする。（下図参照）
このチェックは、usingApp で設定されている FrontISTR の version と合致させる事が必要。
（下図の「実行ファイル場所」と「実行ファイル ver」が合致している事が必要）
この後、「設定」ボタンをクリックして確定させる。



この設定で、ver4.6 での計算が可能になる。
ver4.6 では、FistrModel.cnt ファイル内の「!output_type=VTK」の設定でエラーがでるので、この行をコメントアウトしている。逆に、ver5.0 では、この設定がないと、vtk を出力せず、計算結果が見れないことになるので、この設定を ver に応じて修正している。

設定後、「FrontISTR 実行」ボタンをクリックして、ver4.6 の FrontISTR の計算を走らせる事ができる。

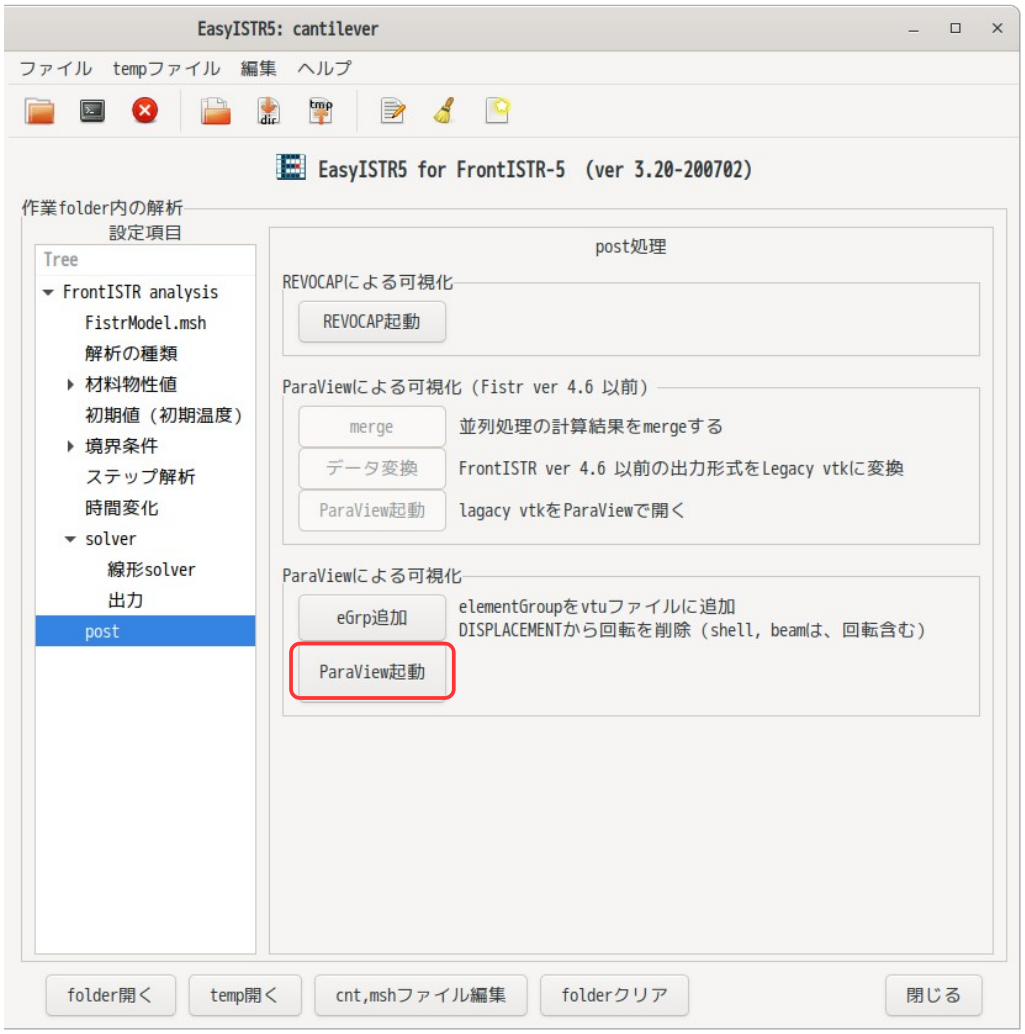
3-8.　結果の可視化

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）を可視化してみる。
FrontISTR からは、vtk の xml フォーマット（拡張子: vtu）が出力されているので、paraView で直接読み込む事ができる。（vtk 変換は不要。）

paraView を起動する為には、「ParaView 起動」ボタンをクリックする。



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。そのファイル名は、

FistrModel.vis_psf.0001.pvtu

なので、これを開く。

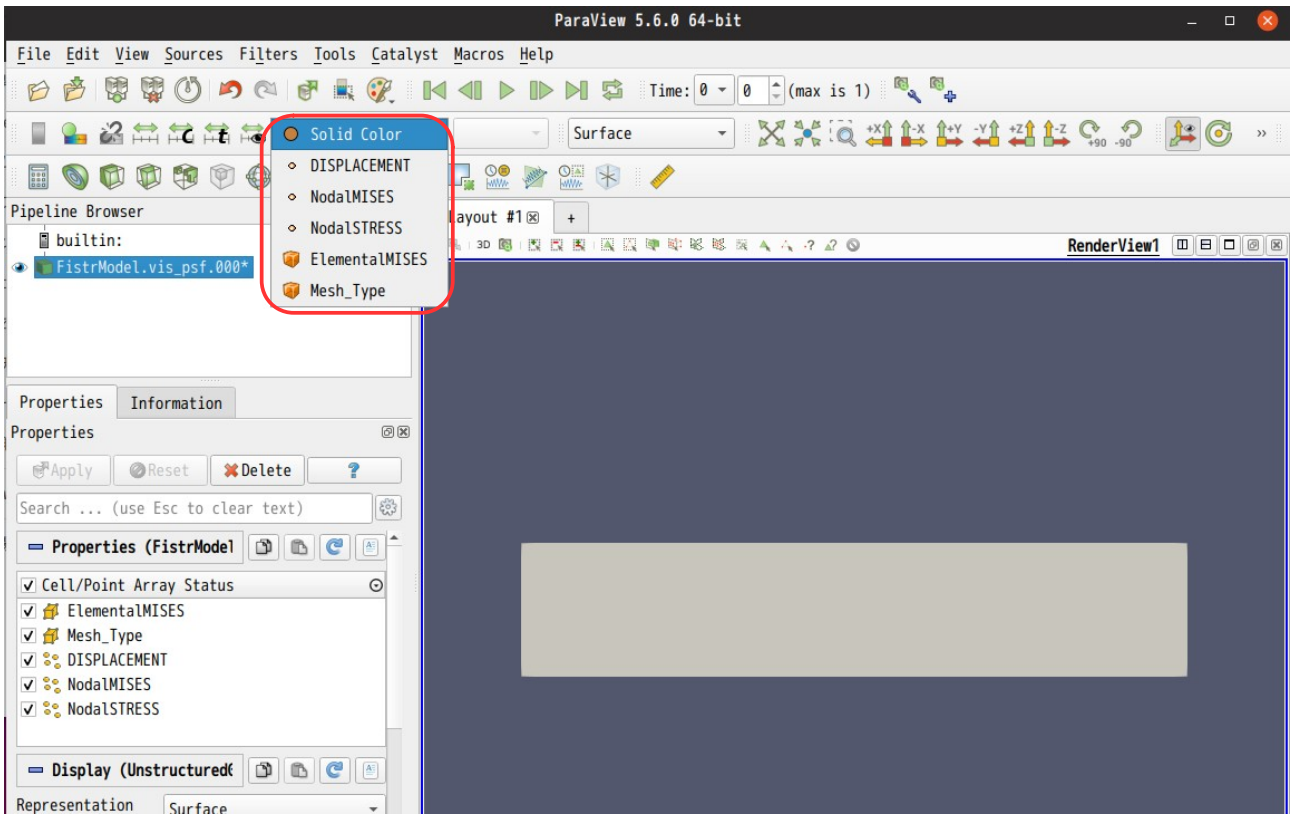
今回の計算結果として、3-7-2 項で以下の項目を設定している。

変位
節点 Mises 応力
節点応力
要素 Mises 応力

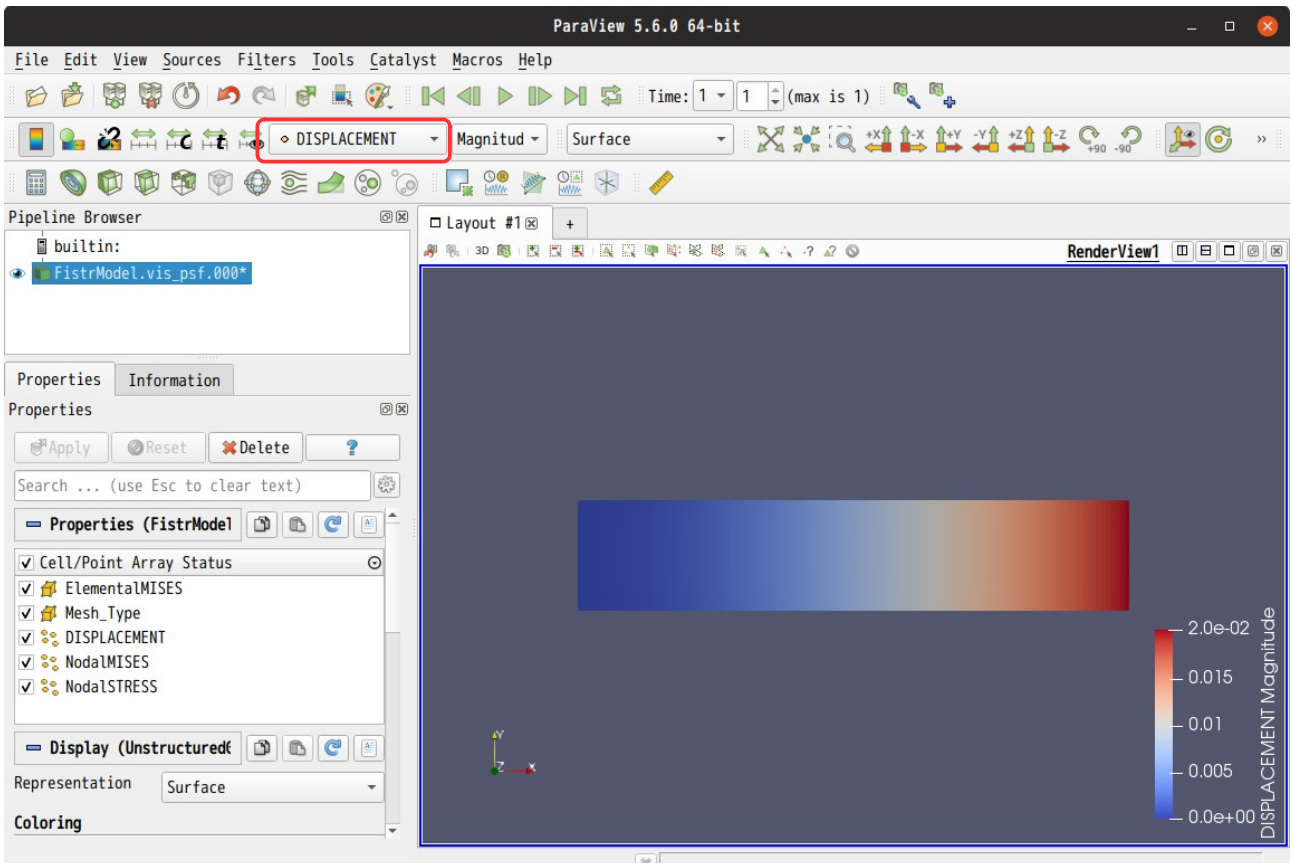
paraView からは、下図の様に、

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

が選択できる。
計算結果には、初期値が含まれている為、純粋な計算結果を知る為には、Time を進めて確認する。

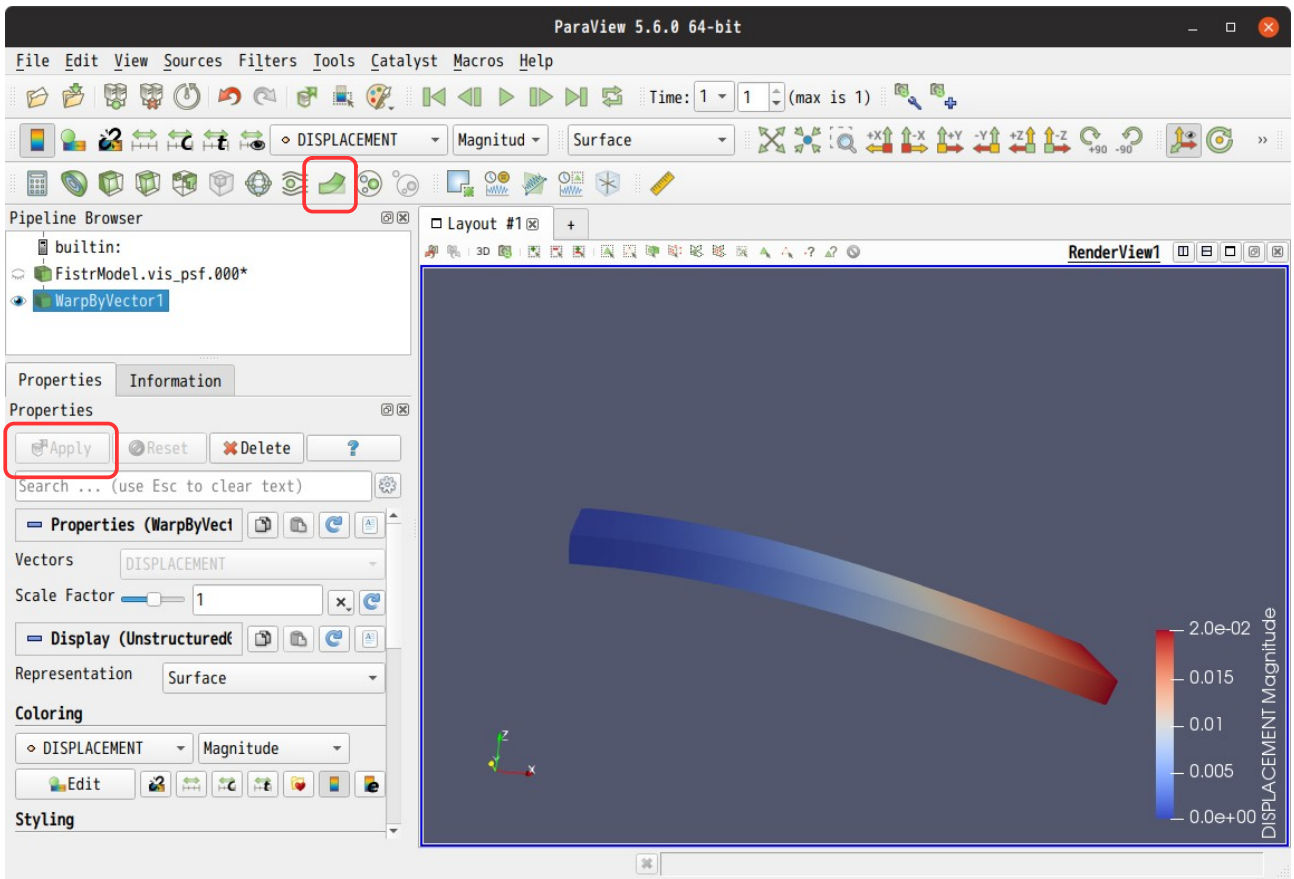


ここで、変位（DISPLACEMENT）を読み込んで表示させてみる。paraView 上で、表示項目内から「DISPLACEMENT」を選択し、Time を進めて表示。下図が DISPLCEMENT を表示させた状態。



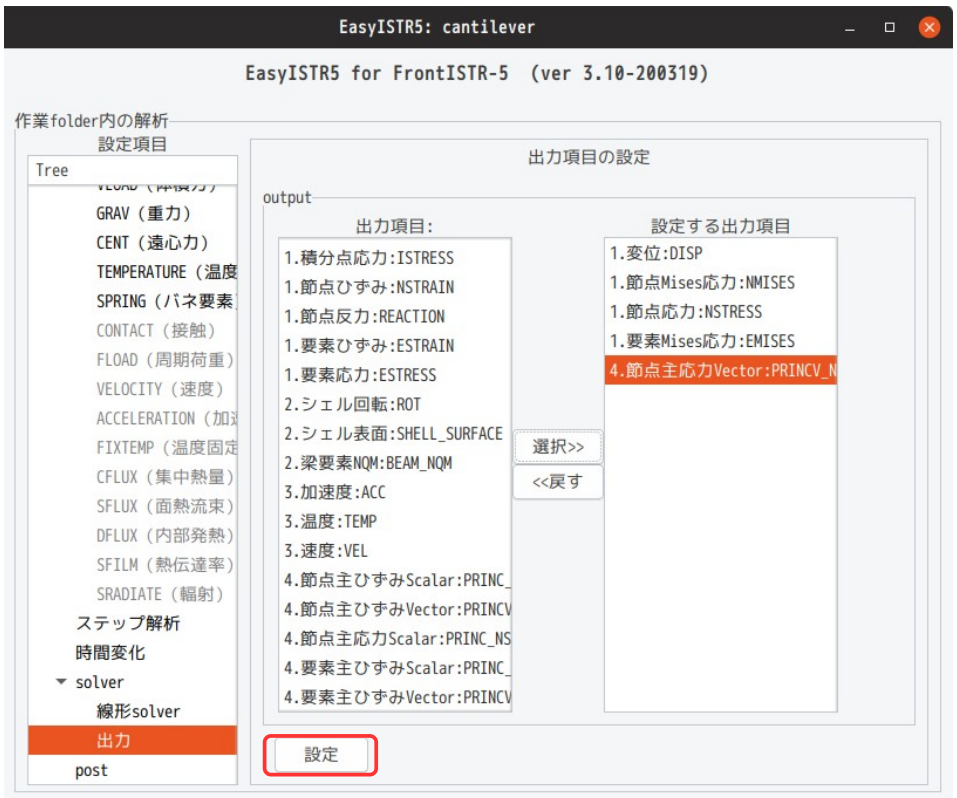
この状態では、変位量が色分けされているだけだが、モデル形状を変位に応じて変形させる事もできる。
モデルを変位量分変形させる為には、「変形（Wrap by vector）」ボタンをクリックし、Vectros から「DISPLACEMENT」を選択し、「Apply」ボタンをクリックする。
下図が、変形させた状態になる。

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が変更できる。



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

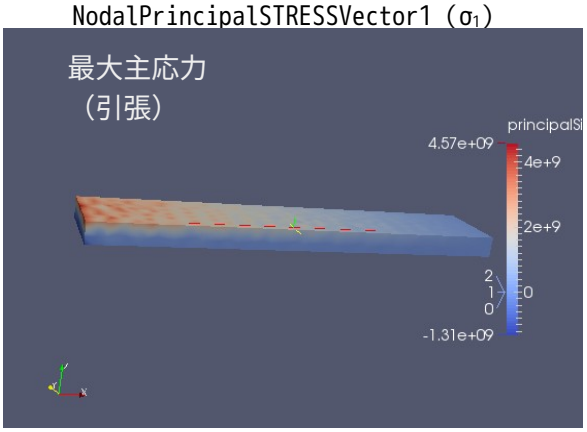
FrontISTR-5 では、出力項目として、主応力、主ひずみ、mises ひずみを出力させる事ができる。
これらを出力させる為には、設定項目 Tree 内の「solver」>「出力」から以下の様に選択する。
（「4.節点主応力 Vector:PRINCV_NSTRESS」を選択し、「設定」ボタンをクリックする。）



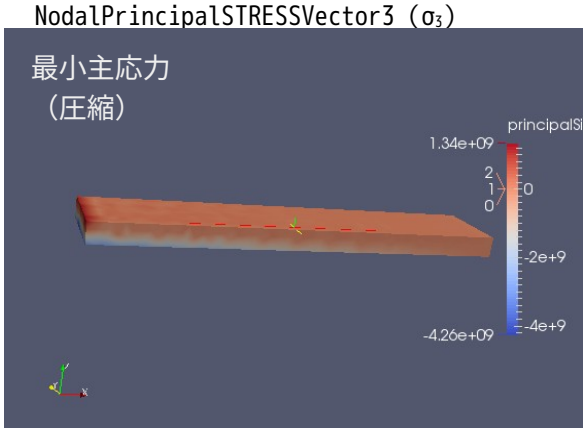
この後、「FrontISTR 実行」ボタンをクリックして、再度計算させると、FrontISTR が主応力を res ファイルに追加する。VTK ファイルには追加されないので、「eGrp 追加」ボタンをクリックして、VTK ファイルに追加する。

paraView を起動して、主応力を表示させることができる。
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している。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。

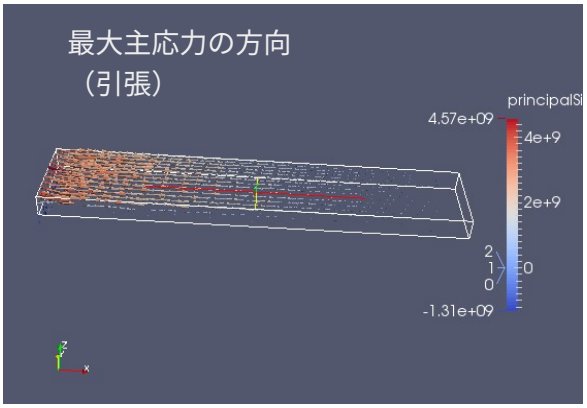
NodalPrincipalSTRESSVector1（σ_1）

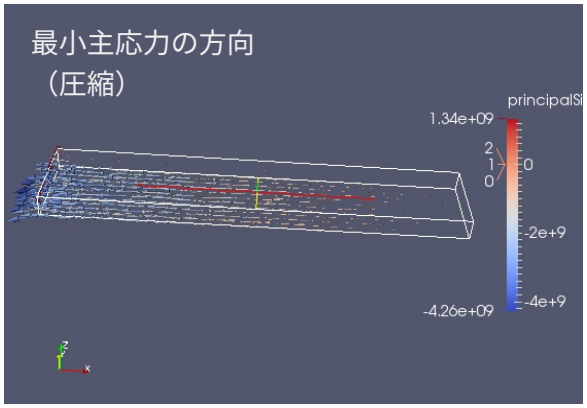


NodalPrincipalSTRESSVector3（σ_3）



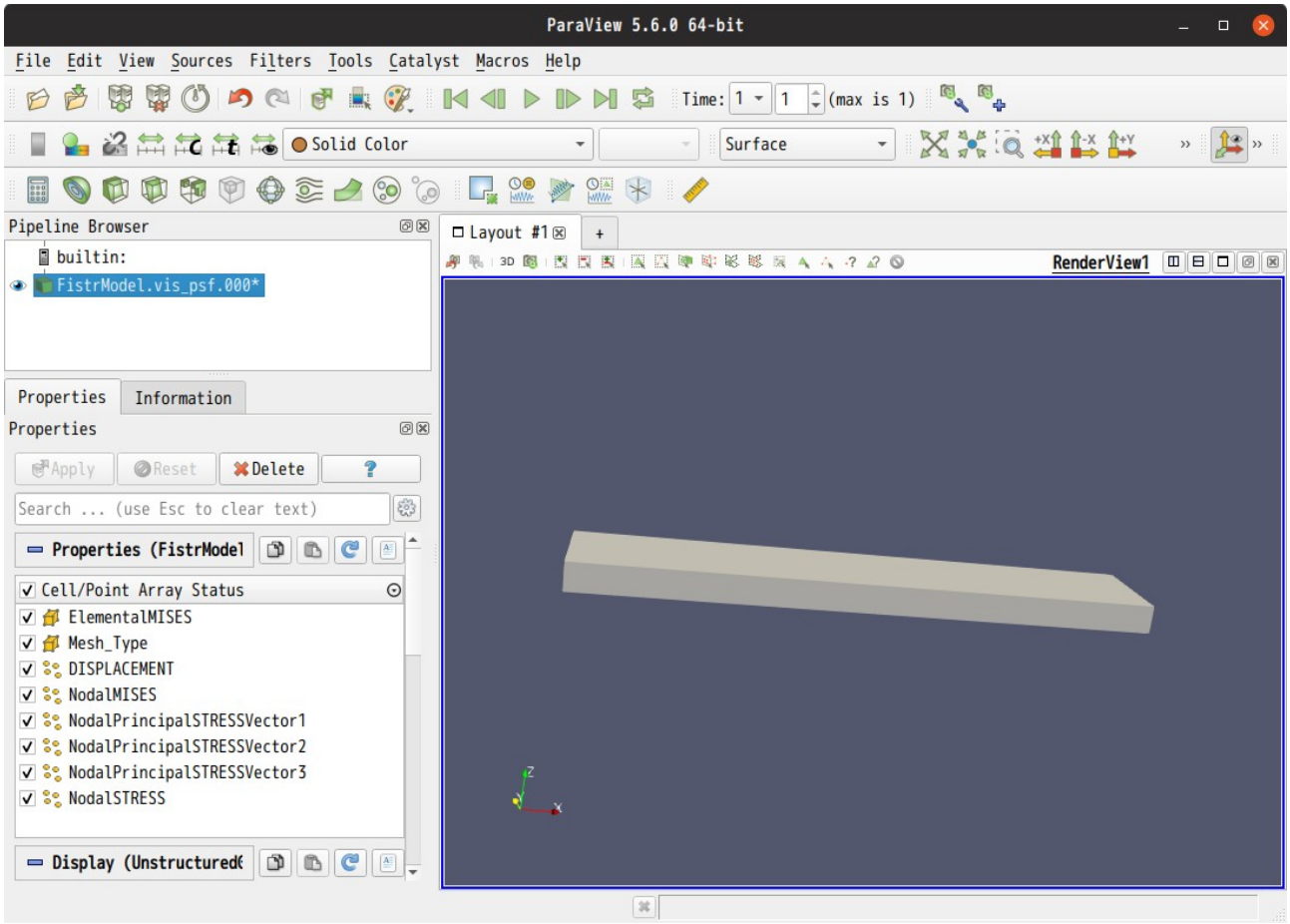
NodalPrincipalSTRESSVector1（σ_1の各方向成分）　NodalPrincipalSTRESSVector3（σ_3の各方向成分）



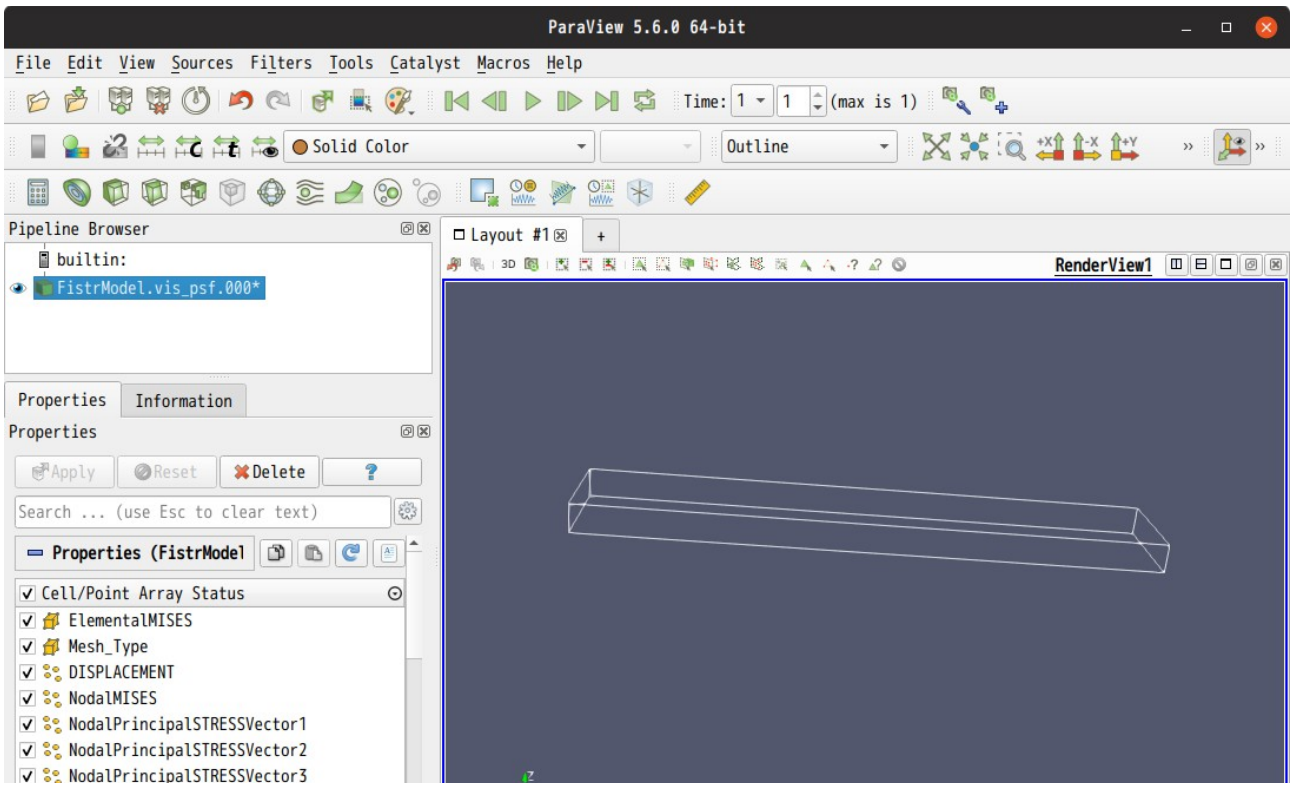


上記の様に、片持ち梁の上面に引張が梁の長手方向に働き、下面に圧縮が梁の長手方向に働いている事が確認できる。

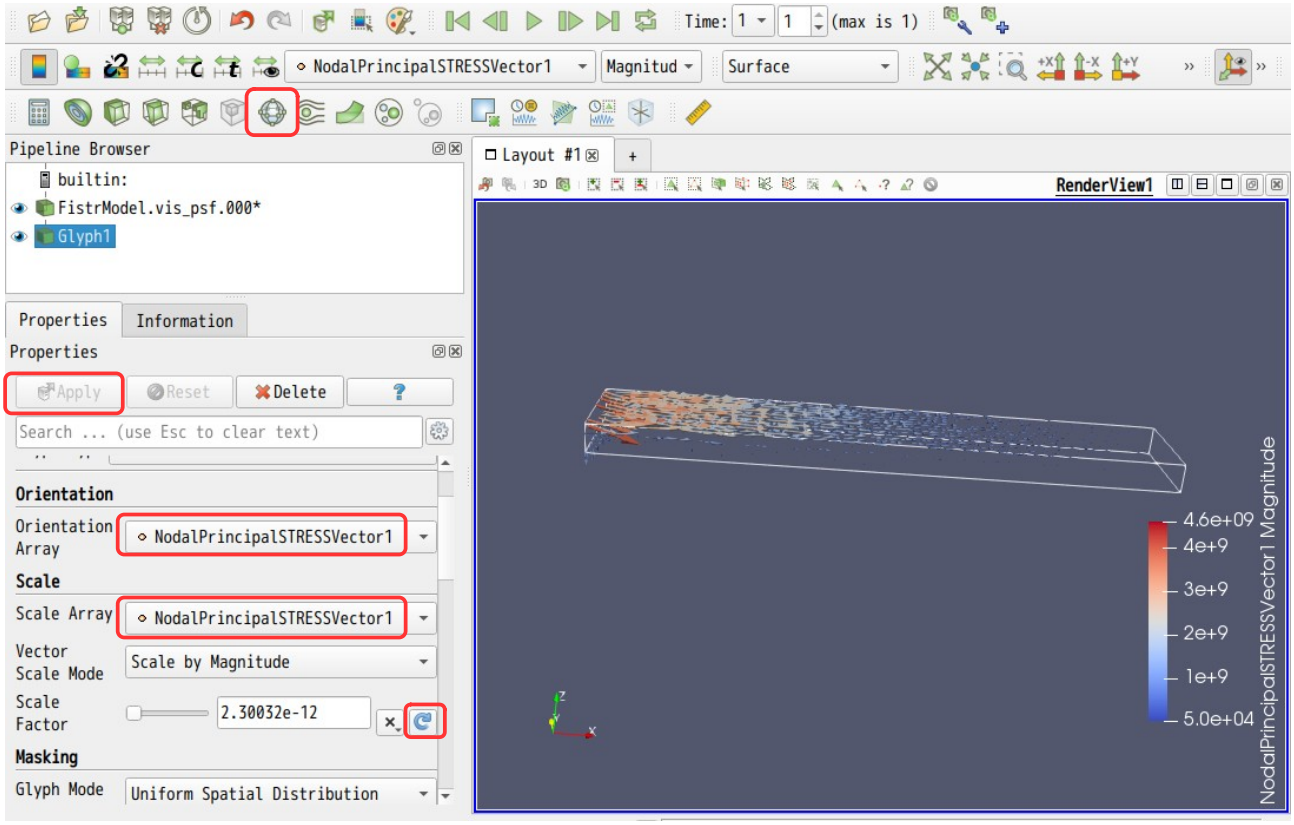
また、ParaView による矢印（ベクトル）の表示は、以下の方法で表示させる事ができる。以下の例では、最大主応力の表示を行っている。
その方法は、まず vtk ファイルを読み込む。下図が読み込んだ状態。



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、Orientation と Scalar Array を「NodalPrincipalSTRESSVector1」に設定し、ボタンをクリック、「Apply」ボタンをクリックする。
下図がこの状態になる。



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

EasyISTR5 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。

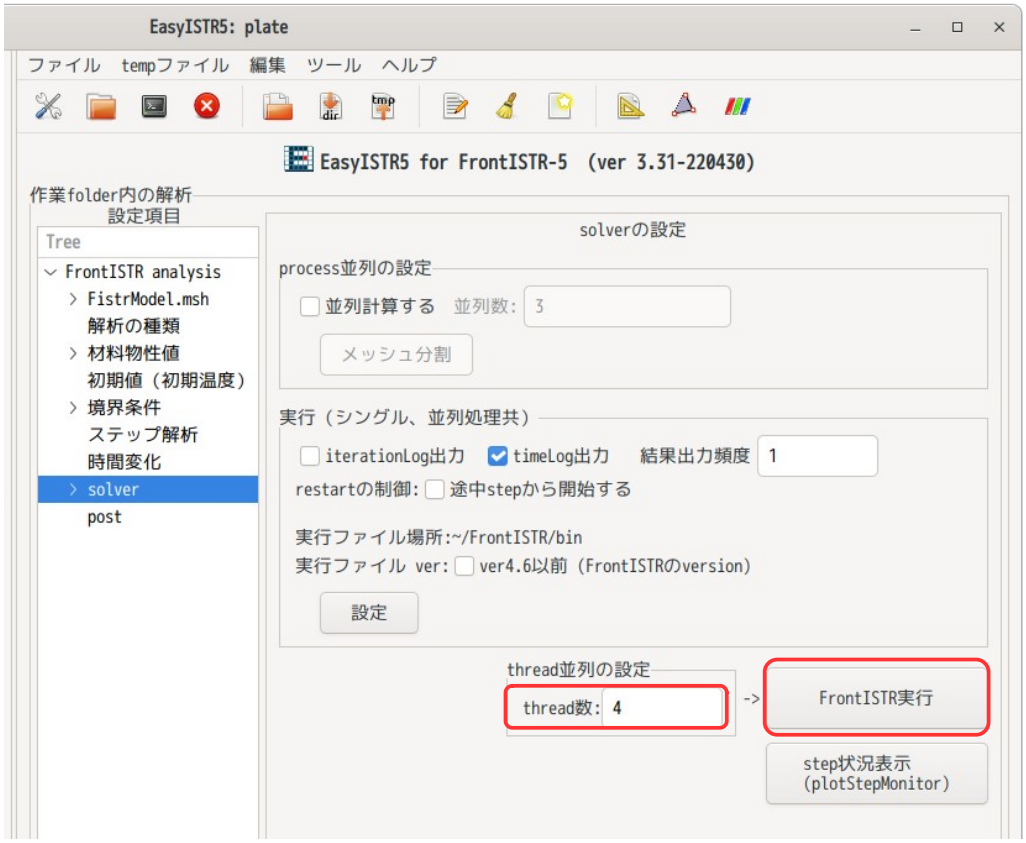
並列処理としては、「thread 並列」と「process 並列」ができる。

thread 並列：　1 個のプロセスを細かい処理に分解して、並列処理する。
実行例：　$ fistr1 -t 4
4 thread で実行（thread で 4 並列）

process 並列：　並列数分メッシュを分割し、その数分のプロセスを起動して処理する。
1 個のプロセスが、分割された 1 個のメッシュを担当して、並列処理する。
実行例：　$ mpirun -np 4 fistr1 -t 1
4 process を起動し、各 process は、1 thread で実行する。

3-9-1-1. thread 並列の実行方法

thread 並列の設定内の、thread 数に「4」を入力する。（4 thread で実行）この後、「FrontISTR 実行」ボタンをクリックして実行する事で、4 thread で実行できる。（メッシュ分割は不要）



尚、thread 数に「0」（0 以下）、「none」（数字に変換できない文字列）を入力すると、thread の設定を行わない。（「-t」オプションを付加しない。）
この状態で、実行した場合は、PC のコア数を調べ、そのコア数分の thread で実行する。

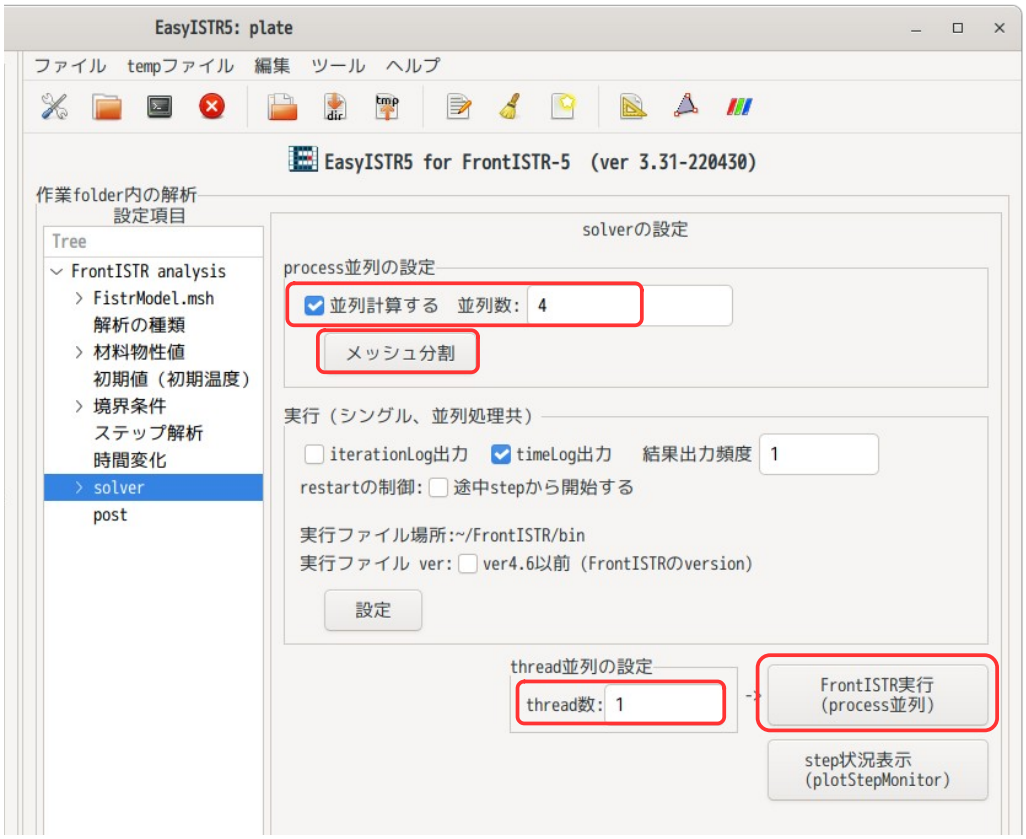
3-9-1-2.　process 並列の実行方法

これは、MPI による並列処理になる。

まず process 並列の設定内の「並列計算する」をチェックし、並列数に「4」を入力する。（4 並列の設定）
この後、「メッシュ分割」ボタンをクリックして、メッシュを分割する。（4 分割される）
さらに、thread 並列の thread 数を「1」に設定しておく。
最後に、「FrontISTR 起動」ボタンをクリックして、実行する。

以上の設定は、
　　4 process 起動し、各 process は、1 thread で実行する。
　　（4 process × 1 thread = 4 core　であり、4 CPU を必要とする。）
になる。

ここで、thread 並列の設定をしなかった場合は、PC が持っている core 数分の thread が設定されるので、
　　4 process × 4 thread = 16 core
となり、物理 core 数を超えてしまう為、計算が極端に遅くなってしまう。



3-9-2.　並列計算結果の可視化

process 並列計算結果は、各 process 毎に分散して結果が保存されているが、VTK の XML フォーマット（vtu ファイル）は、分散しているデータをそのまま paraView で読み込むことができ、表示させることができる。この為、シングル計算と同様に、「ParaView 起動」ボタンをクリックして、paraView を起動して、結果を確認することができる。

FrontISTR ver4.6 で process 並列計算させた時は、「merge」ボタンで分散結果を結合し、「データ変換」ボタンで結合した結果を legacyFormat で vtk 変換し、「ParaView 起動」で legacyFormat の vtk を読み込む必要がある。

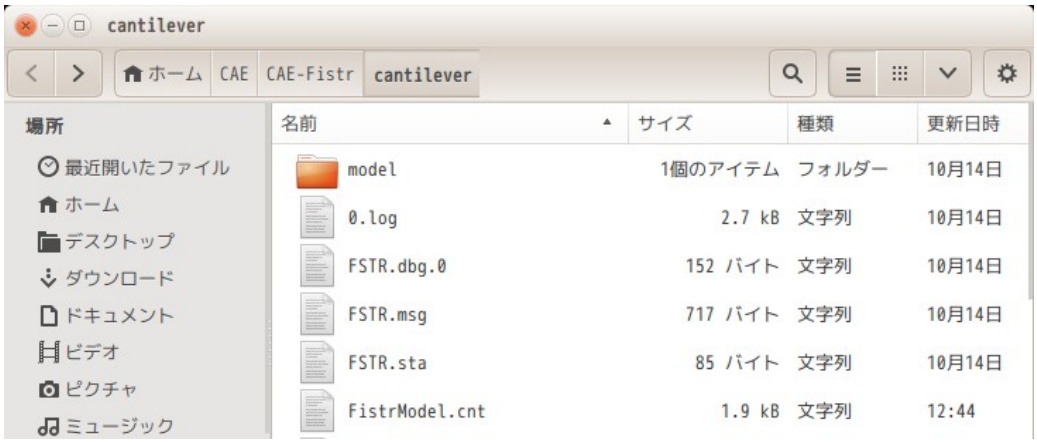
3-10.　EasyISTR のメニューバー、ツールバーの動作

EasyISTR 下部には、以下の toolBar を配置している。メニューバーも同様。

次項以降に、これらボタンの動作について説明する。

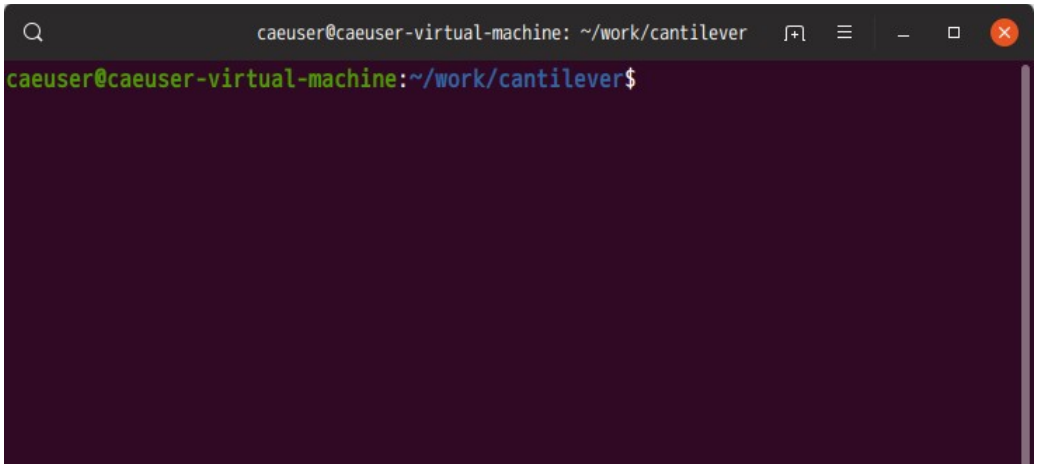
3-10-1.　「folde 開く」ボタンの動作

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.　「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。



3-10-3.　「閉じる」ボタンの動作

EasyISTR を閉じる。

3-10-4.　「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。

3-10-5.　「保存（tmp → dir）」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.　「読み込み（dir → tmp）」ボタンの動作

実 folder（workFolder）の内容を省略して、tempFolder に転送する。
前項と逆方向に転送する。

3-10-7.　「cnt,msh ファイル編集」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。

FistrModel.cnt　FistrModel.msh　hecmw_ctrl.dat　hecmw_part_ctrl.dat

```
 1 ##################################################
 2 # Control File for FrontISTR                     #
 3 # generated by REVOCAP at 2015.04.03 09:42:11 #
 4 ##################################################
 5 !VERSION
 6  3
 7 ####################
 8 # Analysis Control #
 9 ####################
10 !WRITE,VISUAL
11 !WRITE,RESULT
12 !SOLUTION,TYPE=STATIC
13 !STATIC
14 ##################
15 # SOLVER CONTROL #
16 ##################
17 !RESTART, FREQUENCY=1
18 !SOLVER, METHOD=CG, PRECOND=1,ITERLOG=NO,TIMELOG=YES
19 20000, 2
```

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。

3-10-8.　「folder 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。

3-10-9.　「cnt, dat ファイル初期化」ボタンの動作

同じ cnt ファイルを使って、線形、非線形解析を繰り返す、シングル、並列計算を繰り返す等を行うと、cnt ファイルが壊れ、実行時にエラーが発生することがある。
この様な場合に、cnt, dat ファイルをデフォルトの状態に戻す事で、エラーが解消する。
この操作は、EasyISTR トップページの「cnt ファイル置き換え」ボタンと同じ操作になる。

3-10-10.　「CAD 起動」ボタンの動作

作業 folder をカレントディレクトリに設定して、CAD を起動する。CAD の起動コマンドは、usingApp ファイルで設定された内容になる。

3-10-11.　「mesher 起動」ボタンの動作

作業 folder をカレントディレクトリに設定して、mesher を起動する。mesher の起動コマンドは、usingApp ファイルで設定された内容になる。
mesher については、linux、wondows とも Salome を推奨する。
Salome の方が部分的にメッシュサイズを変更できるので、思惑通りのメッシュが作成でき便利。
windows では、salome-platform の web 上で windows のインストーラ（exe ファイル）がダウンロードできるので、楽にインストールできる。

PrePoMax でメッシュを作成する場合は、CAD で作成したモデルを STL 形式で出力する。PrePoMax で STL を読み込み、メッシュを作成（netgen）し、境界条件を設定する要素 group、面 group、節点 group を作成して、Calculix の inp 形式で export する。これにより、EasyISTR が扱える状態になる。

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 コマンドの従来の動作は、unv 形式 → abaqus 形式 → fistr 形式 で変換していた。
理由は、unv → abaqus 形式に変換する「unv2abaqus.py」が CAELinux から提供されていた為、これをそのまま使っていた為。しかし、変換速度向上の為、unv → fistr 形式に直接変換する「unv2fistrEx.py」を作成し、これを使っている。

unv2fistrEx.py は、以下の要素タイプの変換に対応させている。

要素タイプ	unv	FrontISTR	備考
四面体　1 次 2 次	111、118	341、342	solid
六面体　1 次 2 次	115、116	361、362	↑
五面体　1 次 2 次	112、113	351、352	↑
三角形　1 次 2 次	41、42	731、761、732	shell
四角形　1 次	44	741、781	↑

beam	1 次	11	611、641	beam

unv 形式の 41, 42, 44 は、シェルとして変換される。
また、solid 要素中に 11, 41～44 の要素が含まれると、自動的に 3 自由度のシェル、beam に変換され、EQUATION を定義して、3 自由度シェル, 3 自由度 beam が solid に接続される。

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

この変換は、Abaqus や PrePoMax が吐き出した inp ファイルを FrontISTR 形式に変換する。
また、EasyISTR5-3.10 以前は、同じ要素に複数の要素 group を elset で定義した場合、その状態のままで変換できなかった。（要素 group を上書きしていく変換の為、inp ファイルの要素 group とは異なった要素 group が出来上がっていた。）
しかし、EasyISTR5-3.20 からは、inp ファイル内の*Element で定義した要素 group、*Elset で定義した要素 group をそのまま FrontISTR 形式変換できる様に修正している。

abaqus2fistr 変換は、メッシュ部の変換（abaqus2fistr.py）と境界条件部の変換（abaqusinp2fistrcnt.py）に分かれている。

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

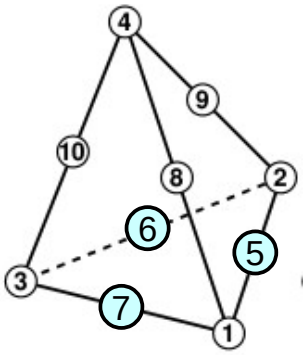
```
$ abaqus2fistr.py plate.inp
                  ---------
                  変換対象の inp ファイル
```

この変換で扱えるメッシュタイプは、以下に限られる。

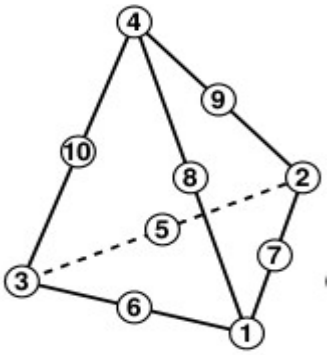
要素タイプ		abaqus	FrontISTR	備考
四面体	1 次 2 次	C3D4、C3D10	341、342	solid
六面体	1 次 2 次	C3D8、C3D8R、C3D20、C3D20R	361、362	↑
五面体	1 次 2 次	C3D6、C3D15	351、352	↑
三角形	1 次 2 次	STRI35、S3、S6	731、761、732	shell
四角形	1 次	S4R5	741、781	↑
beam	1 次	B31	611、641	beam

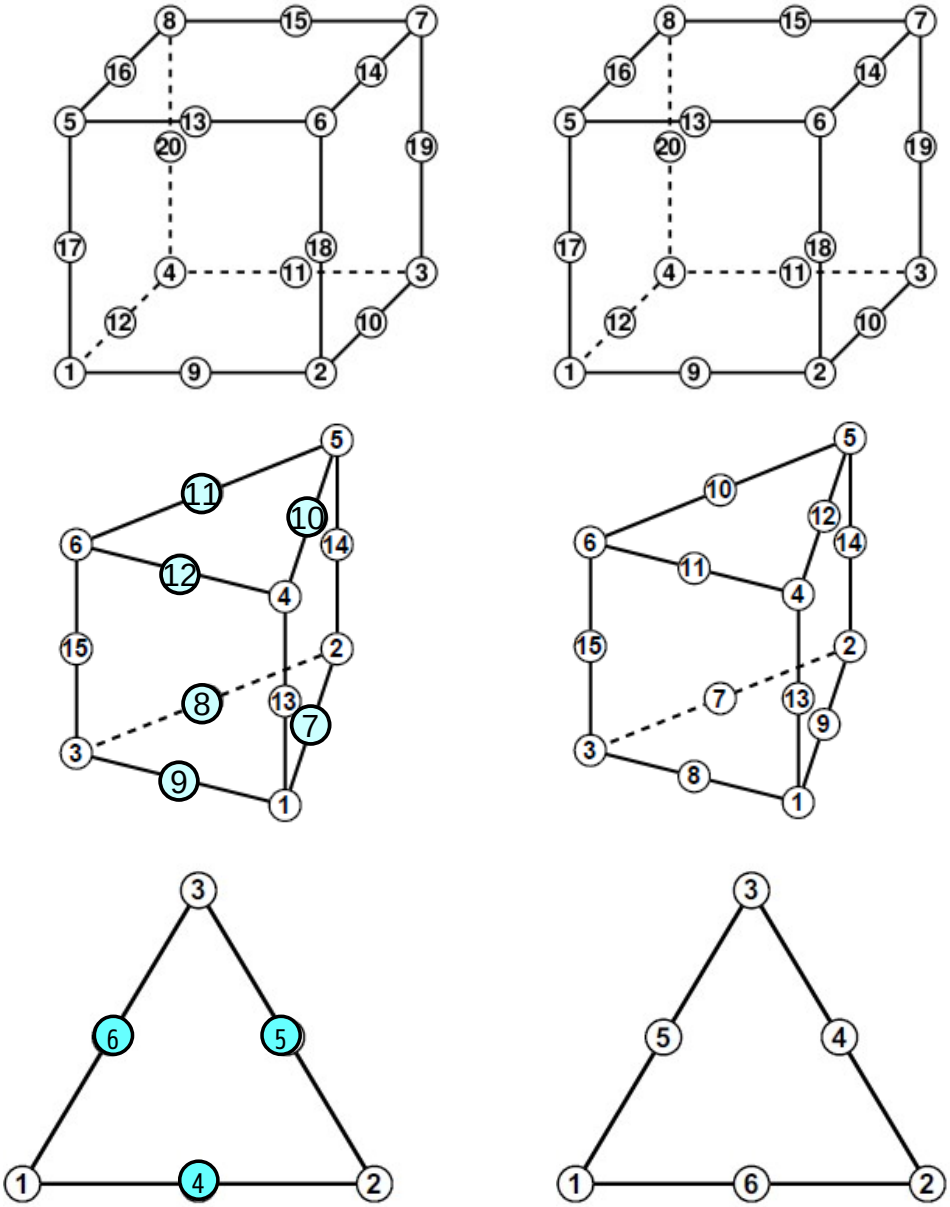
上記の abaqus メッシュを変換するに当たり、各要素の節点 No 順が abaqus と FrontISTR 間で、以下の様に異なっている部分が存在しているので、これを修正した上で変換している。

＜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。（大文字、小文字の区別を行っていない）

```
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 ----------

```
< 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
```

```
< 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.
#####################
     :
```

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

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、paraView で直接読み込むことができる形式に変換している。
vtk ファイルには、legacy フォーマットと xml フォーマットの 2 種類があり、xml フォーマットは、並列処理の様な分散しているデータを分散した状態で読み込むことができる利点がある。FrontISTR-5 は、xml フォーマットで結果を出力させる事ができる。

EasyISTR では、従来は legacy フォーマット（拡張子:vtk）に変換していたが、FrontISTR の出力形式に合わせ、xml フォーマット（拡張子:vtu）に変更した。

メッシュファイル「FistrModel.msh」は、vtk 変換により「convFistrModel.msh.vtu」ファイルを作り出している。この変換は、以下のコマンドを実行してファイルを作成している。

```
$ fistr2vtu.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(10x10x10 の直方体)になる。

```
---------------- convFistrModel.msh.vtu ------------------------
<?xml version="1.0"?>
<VTKFile type="UnstructuredGrid" version="1.0">
  <UnstructuredGrid>
    <Piece NumberOfPoints="8" NumberOfCells="17">
      <Points>
        <DataArray type="Float32" NumberOfComponents="3" format="ascii">
          10.0 10.0 0.0
          10.0 10.0 10.0
          10.0 0.0 0.0
          10.0 0.0 10.0
          0.0 10.0 0.0
          0.0 10.0 10.0
          0.0 0.0 0.0
          0.0 0.0 10.0
        </DataArray>
      </Points>
      <Cells>
        <DataArray type="Int32" Name="connectivity" format="ascii">
          6 0 4 5
          1 0 3 5
          6 5 7 3
          3 0 2 6
          6 0 5 3
          6 0 4
          0 4 5
          4 6 5
          1 0 3
          1 0 5
          3 1 5
          6 5 7
          5 7 3
          7 6 3
          3 0 2
          0 2 6
          2 3 6
        </DataArray>
        <DataArray type="Int32" Name="offsets" format="ascii">
          4 8 12 16 20 23 26 29 32 35
          38 41 44 47 50 53 56
        </DataArray>
        <DataArray type="UInt8" Name="types" format="ascii">
          10 10 10 10 10 5 5 5 5 5 5 5 5 5 5 5 5
        </DataArray>
      </Cells>
      <PointData>
        <DataArray type="Int16" Name="nodeGroup" NumberOfComponents="1" format="ascii">
          0 0 0 0 0 0 0 0
        </DataArray>
      </PointData>
      <CellData>
        <DataArray type="Int16" Name="elementGroup" NumberOfComponents="1" format="ascii">
```

```
          1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
        </DataArray>
        <DataArray type="Int16" Name="surfaceGroup" NumberOfComponents="1" format="ascii">
          0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
        </DataArray>
      </CellData>
    </Piece>
  </UnstructuredGrid>
</VTKFile>
-----------------------------------------------------------------------------------------
```

3-11-3-2.　結果ファイルの変換

FrontISTR は、計算結果を vtk 形式で出力させることができるので、計算結果をそのまま paraView で確認できる。この為、結果ファルの変換は、必要ない。
（主応力などの出力項目を追加した場合は、結果ファイルの変換が必要になる。）

しかし、FrontISTR の出力項目には、要素 Type の項目はあるが、要素 Group の項目がない。この為、要素 Group を現在の出力項目に追加したい場合には、これを追加する事ができる。

要素 Group を追加するには、FrontISTR 側の出力項目に要素の値（例：要素 Mises 応力:EMISES）が含まれている必要がある。これは、各要素の要素 No と要素 type を確認した上で要素 GroupNo を取得している為。
この為、熱解析した場合は、要素の値が出力できないので、この場合は要素 GroupNo を追加する事ができない。

計算結果に要素 Group を追加する方法は、以下のコマンド実行して要素 Group を追加している。

```
$ fistr2vtu.py -mesh FistrModel -res FistrModel
                     ----------      ----------
                     fileHeader 名    fileHeader 名
```

このコマンドの実行で、メッシュと結果ファイルから、FrontISTR が出力した vtk ファイルに要素 Group を追加する事ができる。この変換は、並列計算の様に計算結果が分散している状態でも、要素 Group を追加できる。

FrontISTR-4.6 以前の場合は、計算結果として vtk を出力しないので、FrontISTR が出力した計算結果を vtk 変換する必要がある。この vtk 変換は、legacyFormat の vtk に変換して、paraView で確認する方法をとっている。

3-12.　salome によるメッシュの作成方法

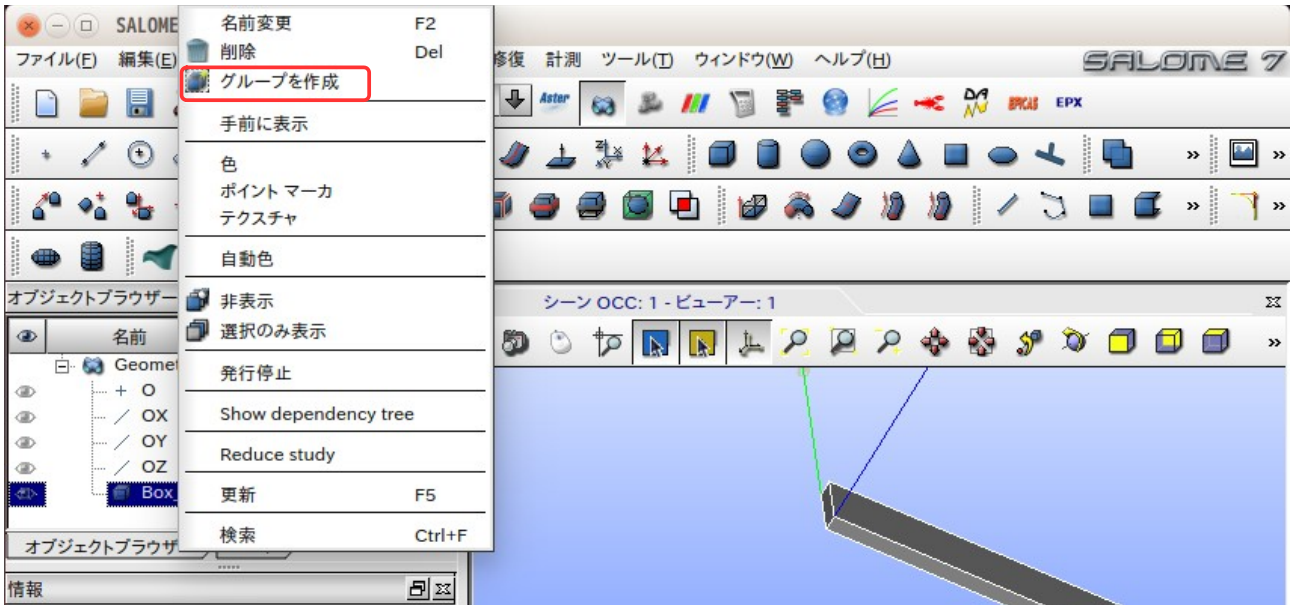
この項で例に取り上げているメッシュの作成方法について説明する。

3-12-1.　ジオメトリの作成

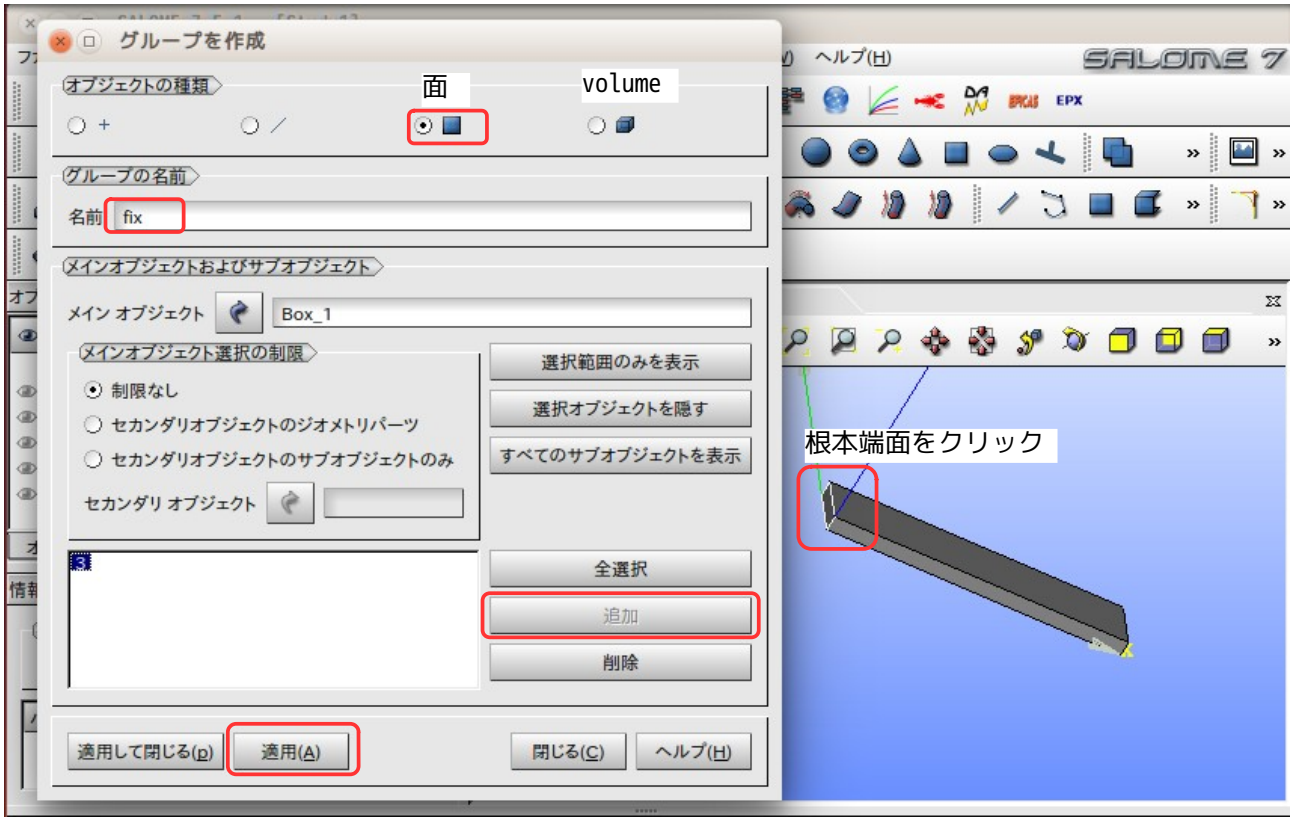
作成するモデル形状は、100 x 20 x 5 mm の直方体になる。Salome を起動後、Geometry モジュールに設定する。この後、「ボックスを作成」ボタンをクリックし、XYZ の寸法を入力する。

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

この後は、メッシュを作成した後の為に、作成した直方体に境界条件を設定する箇所に対してグループを作成する。直方体の根本、先端の端面と上面のグループを作成する。
オブジェクトブラウザーの「Box_1」を選択して、右クリックして、メニューを表示させ、「グループを作成」を選択する。



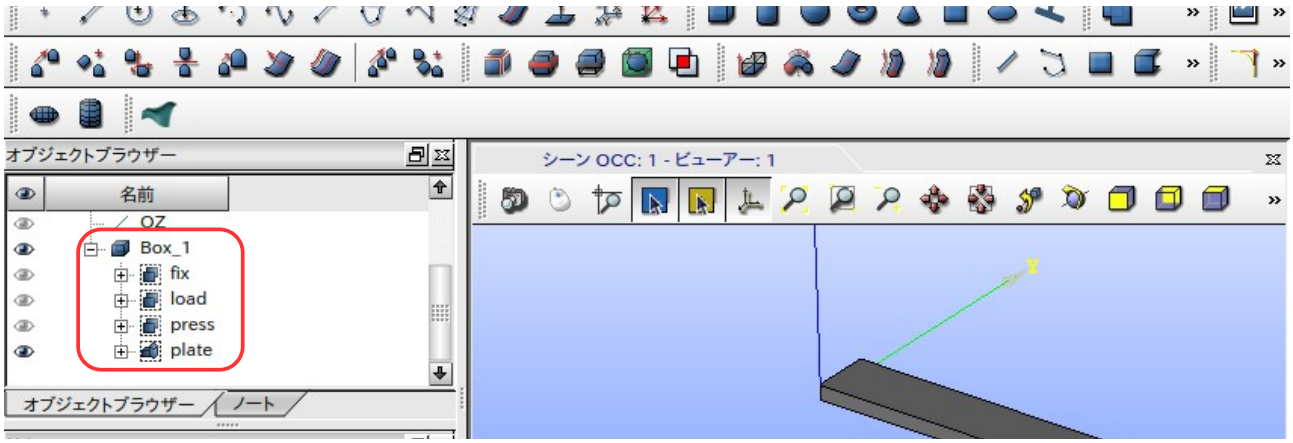
「グループ作成」を選択した後、下図の様に、「面」を選択し、名前「fix」を入力し、直方体の根本端面を、クリックして、「追加」ボタンをクリックする。下図が、その状態になる。この後、「適用」ボタンをクリックして、グループ「fix」を作成する。
同様にして、先端の端面「load」、上面「press」を作成する。
さらに、Volume を選択して、モデル全体を選択して、volume「plate」を作成する。



以上の操作で、Geometry 内で以下のグループが作成できた事になる。

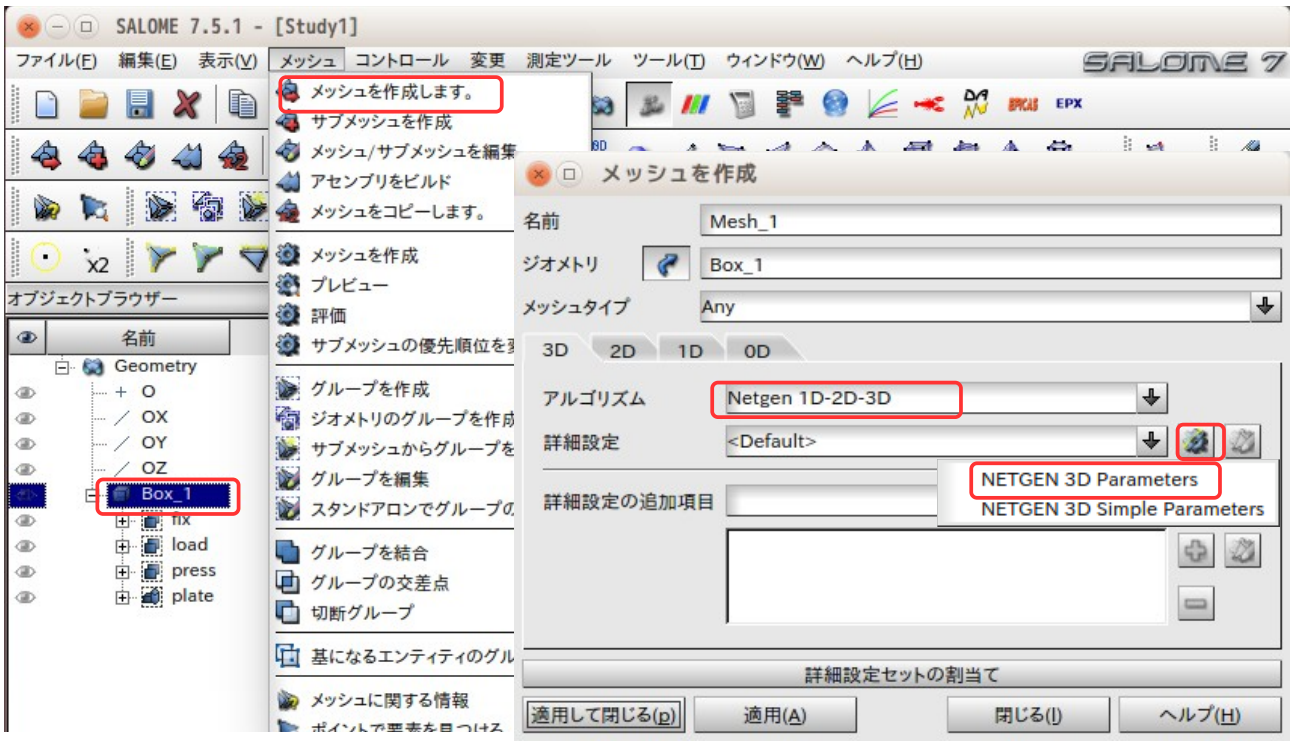
根本端面	fix	面グループ
先端端面	load	面グループ
上面	press	面グループ
全体	plate	volume グループ

この段階で、オブジェクトブラウザーは、Box_1 の下に作成したグループが追加された状態になっている。下図参照。

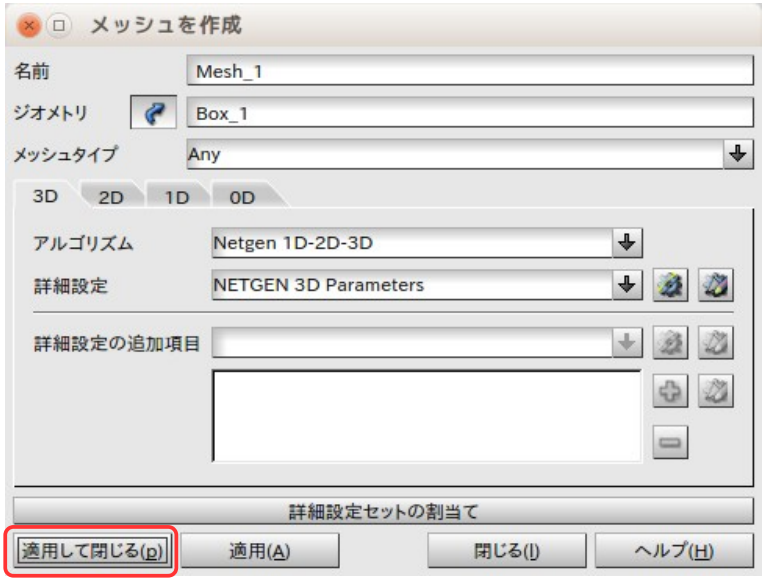


3-12-2.　メッシュの作成

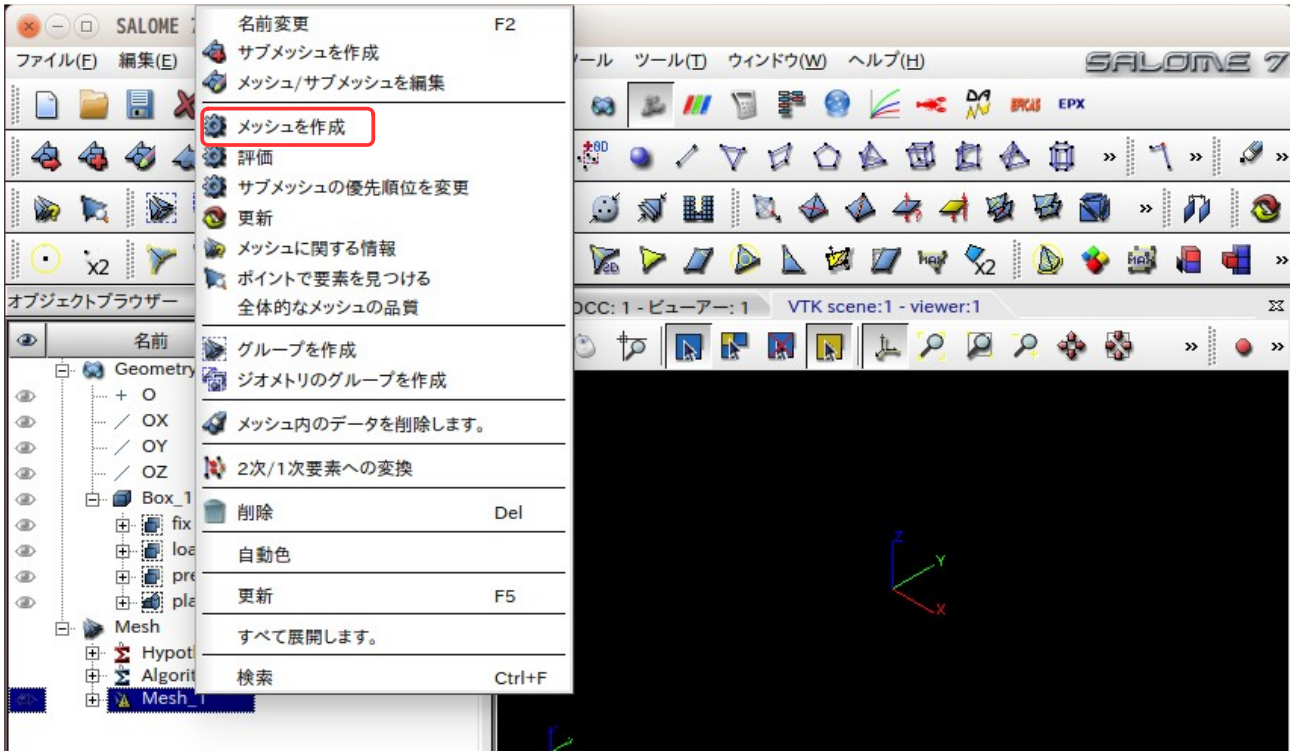
メッシュを作成する。まず、Mesh モジュールに変更する。
オブジェクトブラウザー上の「Box_1」を選択した状態で、メニューバー「メッシュ」>「メッシュを作成します。」を選択する。この後、アルゴリズムを「Netgen 1D-2D-3D」に設定し、ギアのマークをクリックして「NETGEN 3D Parameters」を選択する。



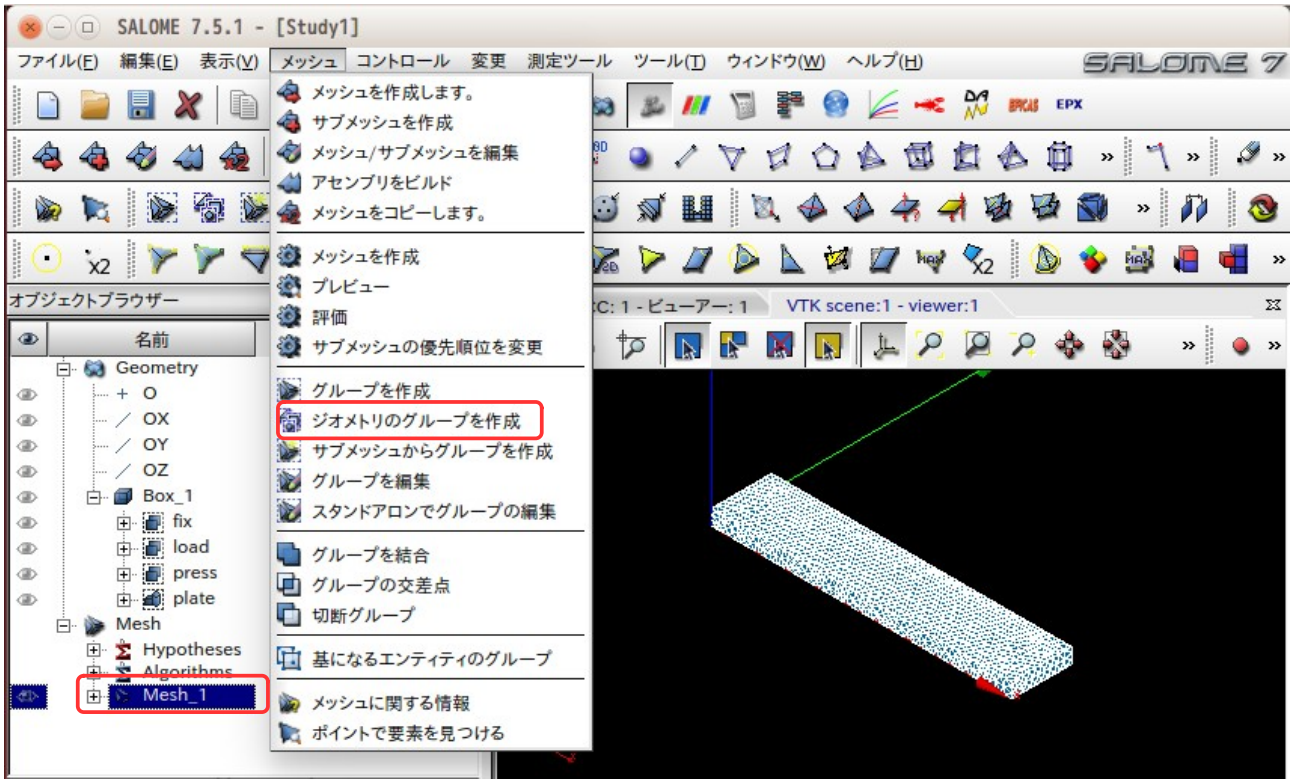
最大サイズを「2」、最小サイズ「1」に設定する。ここでメッシュ作成時のメッシュサイズを指定したことになる。設定後「OK」ボタンをクリックして閉じる。
下図の画面に戻るので、「適用して閉じる」ボタンをクリックして、閉じる。



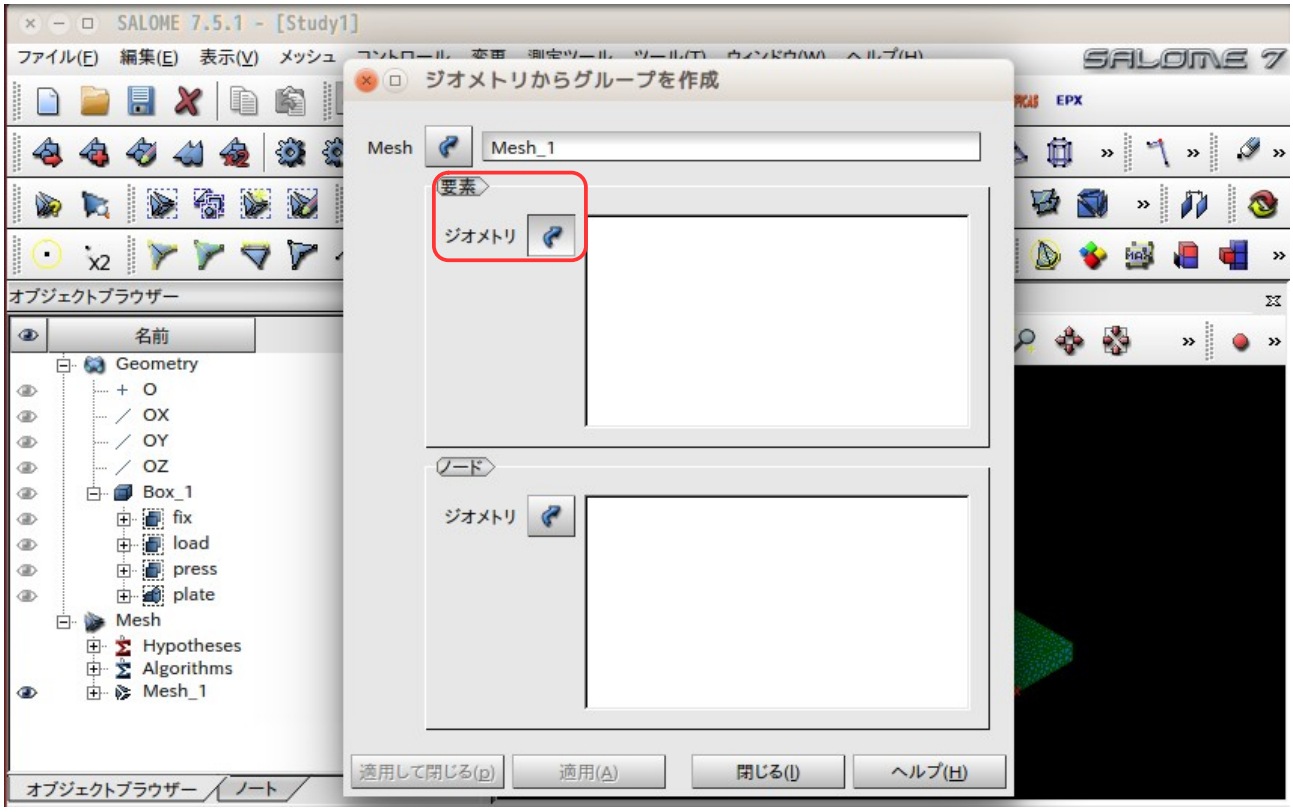
Salome のオブジェクトブラウザーが以下の画面に変わるので、ブラウザー上の「Mesh_1」を選択後、右クリックして、「メッシュを作成」を選択する。これで、設定した条件でメッシュが作成される。



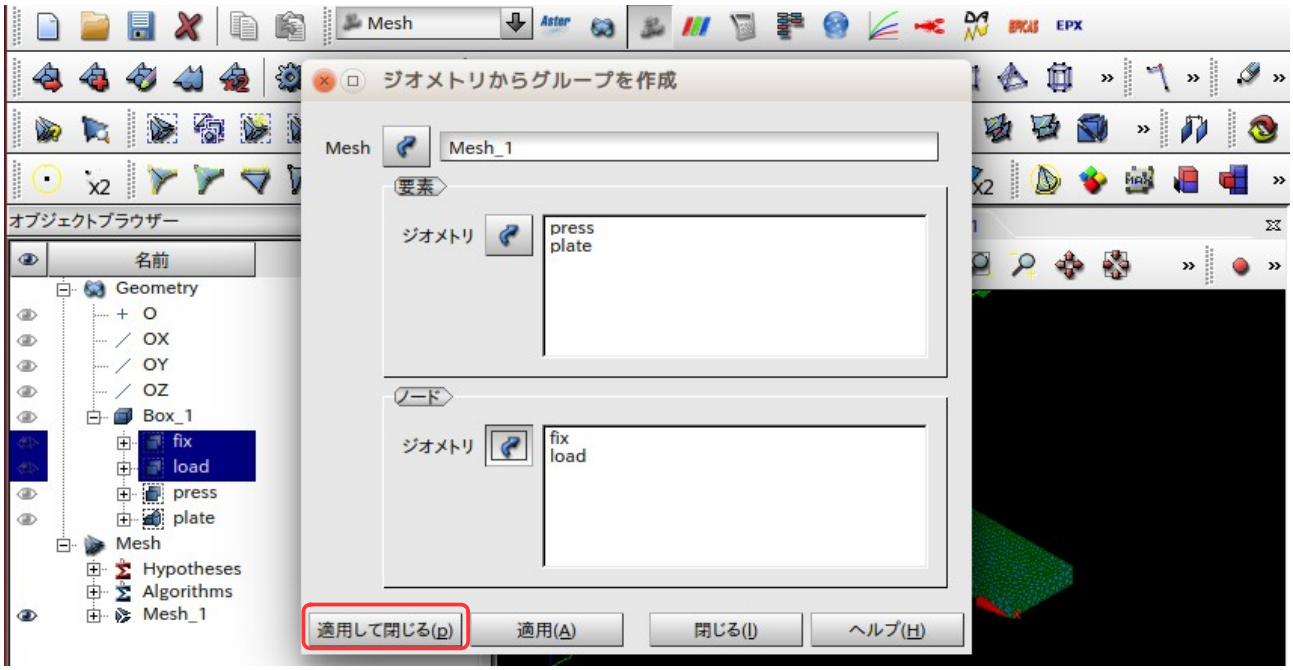
でき上がったメッシュに elementGroup、faceGroup、nodeGroup を作成する。「Mesh_1」を選択後、メニューバー「メッシュ」>「ジオメトリのグループ作成」を選択する。



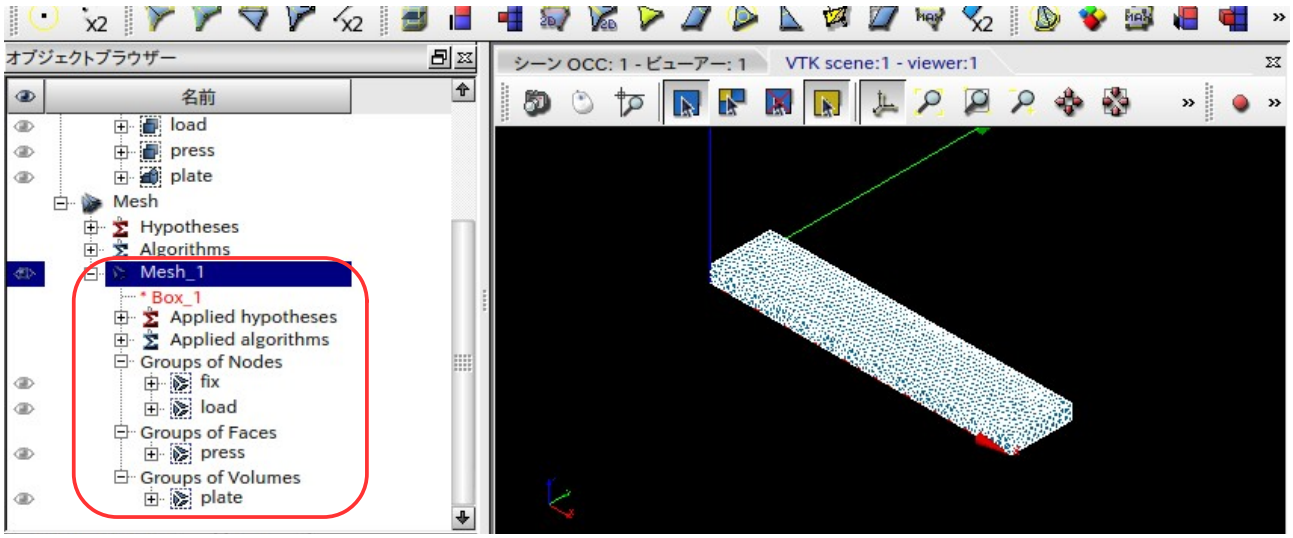
以下の画面上で各 Group を作成する。
まず、要素の「ジオメトリボタン」が ON 状態で、オブジェクトブラウザー上で、elementGroup, faceGroup にしたい Group 名（plate、press）を Shift キーを押しながら選択する。同様にノードの「ジオメトリボタン」をクリックして、ON 状態に設定し、nodeGroup に設定したいグループ名を選択する。



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。

$$A=\sqrt{s(s-a)(s-b)(s-c)}$$

a,b,c: 各辺の長さ
A: 三角形の面積

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

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
```

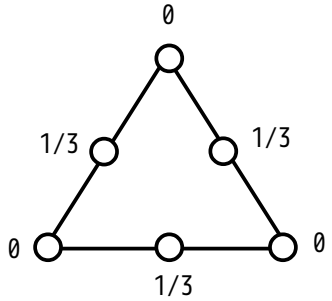
```
---------------- FistrModel.cnt ファイルの内容-------------------------------------
        :
######################
# Boundary Condition #
######################
!CLOAD, GRPID=1, forceType=trueForce, value=0.0, 0.0, 1000     追加したコマンド
load, 1, 0.0
load, 2, 0.0            index 用として残している
load, 3, 0.0
1, 1, 0.0
1, 2, 0.0               節点番号 1 に設定した 3 方向の節点力
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。

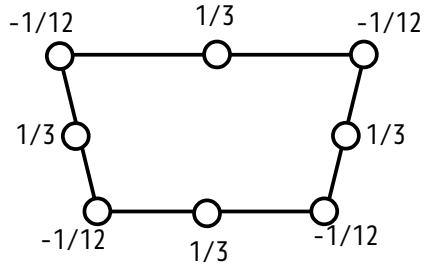
＜三角形 2 次要素＞

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事ができる。
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事ができる。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。

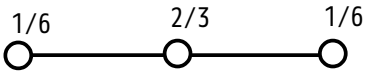
線分は、各要素（四面体、六面体、五面体）のエッジになっているので、各要素の節点番号から節点の座標が判るので、線分の長さが算出できる。
この線分の長さから、線分に印加される荷重が算出できるので、この荷重を節点に均等に配分する事で等分布荷重を設定する事ができる。

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

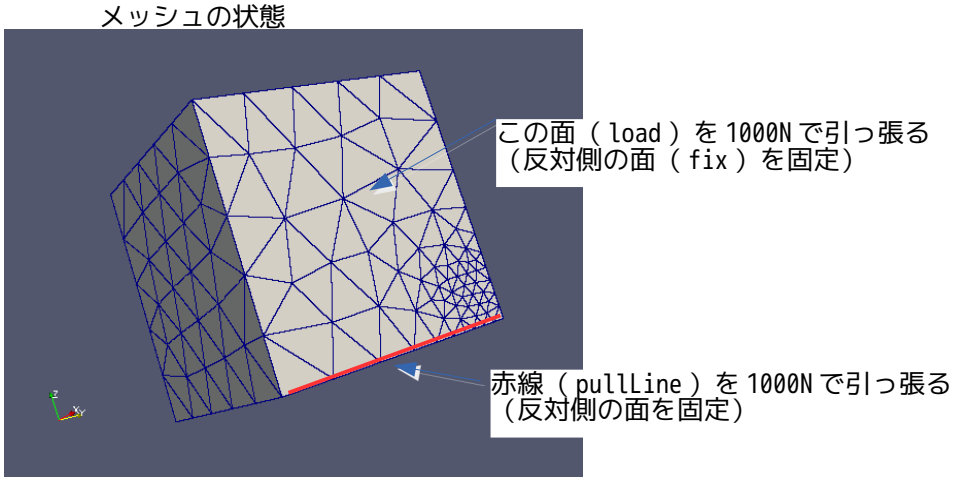
点に等分布荷重を設定する場合は、選択した nodeGroup 内に面（要素の face）と線分（要素の edge）が含まれていない事が必要になる。
もし、要素の face や edge が含まれていると、面や線分に等分布荷重が設定される。

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、□10m 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

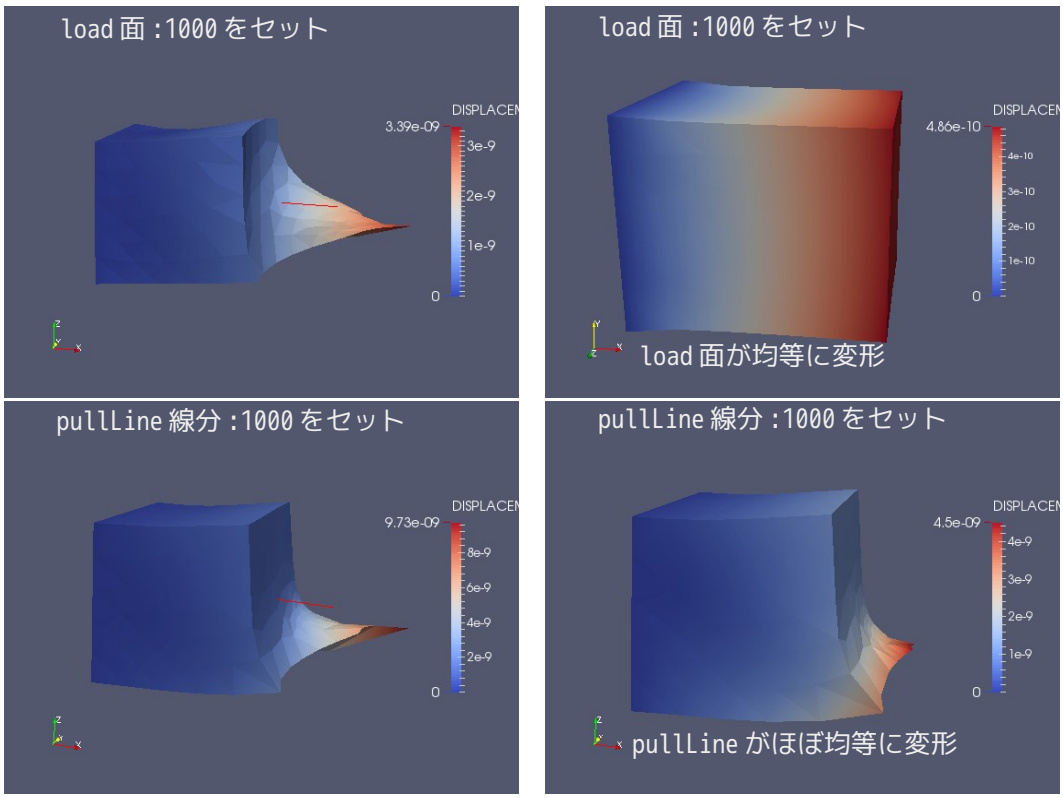


通常の設定では、要素サイズが異なっている部分がある為、細かい要素部分では、荷重密度が大きくなる。
この為、変形形状が均一でなくなってしまう。
これに対して、「等分布トータル荷重」の設定で引っ張った場合は、メッシュサイズの粗密に関係なくなる為、変形形状が均一になるはずなので、これを 1 次と 2 次要素で確認してみる。

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。

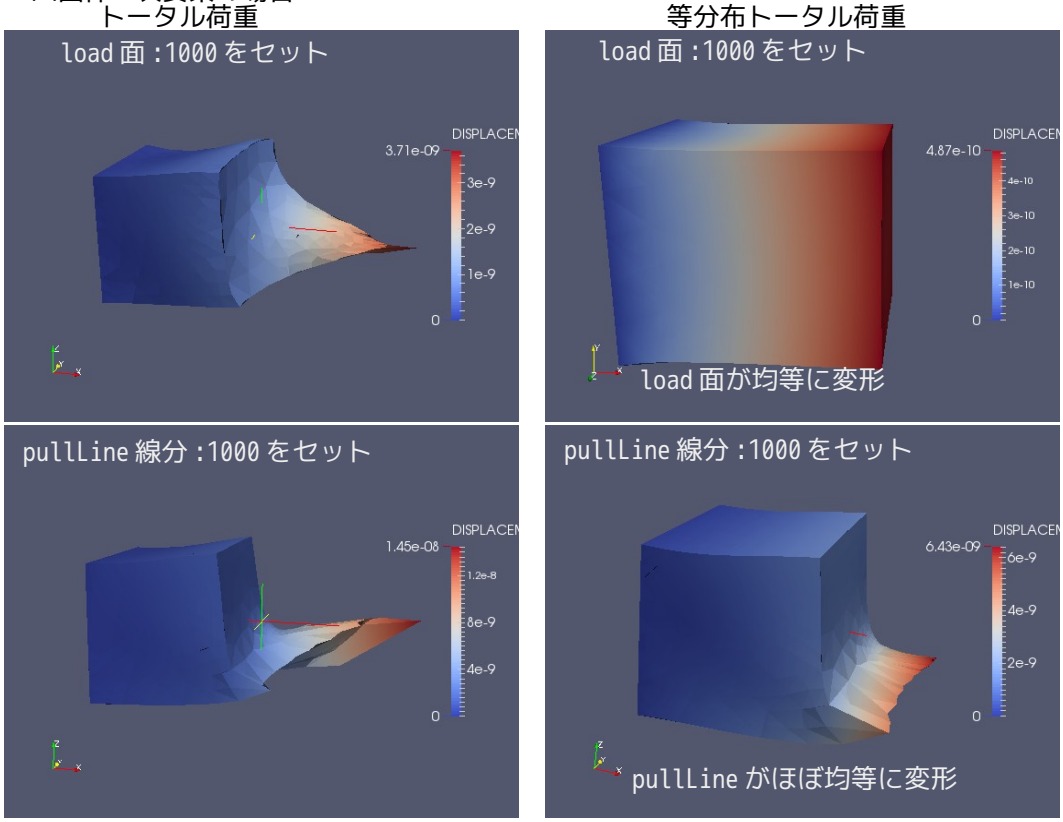
これに対し、「等分布トータル荷重」で同じ 1000N をセットすると、等分布荷重がセットされている為、均等に変形している様子が判る。

＜四面体 1 次要素の場合＞
トータル荷重　　等分布トータル荷重



＜四面体 2 次要素の場合＞

トータル荷重　　等分布トータル荷重



3-14.　部品の結合

複数の部品で構成される Assy モデルの場合、3D CAD 上で全部品を boolean 演算で結合して一体の solid を

作成し、メッシュを作成する事が一般的だが、部品の形状によっては、boolean 演算で結合できない事がある。この様な場合は、各々の部品を単独でメッシュ作成し、EasyISTR 上で単独のメッシュを結合する事ができる。
部品同士のメッシュを結合する方法として以下の 2 種類の方法を準備している。

1）節点を相手 face 上に結合
nodeGgrp の節点を surfaceGrp の face 上に結合するので、結合面の形状や節点数を意識する事無く結合する事ができる。

2）節点同士を結合
nodeGrp 同士の節点を結合する。この為、結合するペアの nodeGrp の形状、節点数は、極力合わせる必要があるので、使いにくい。

3-14-1.　節点を相手 face 上に結合

nodeGrp（slave 側）の節点を surfaceGrp（master 側）内の face 面上に、!EQUATION で結合する。

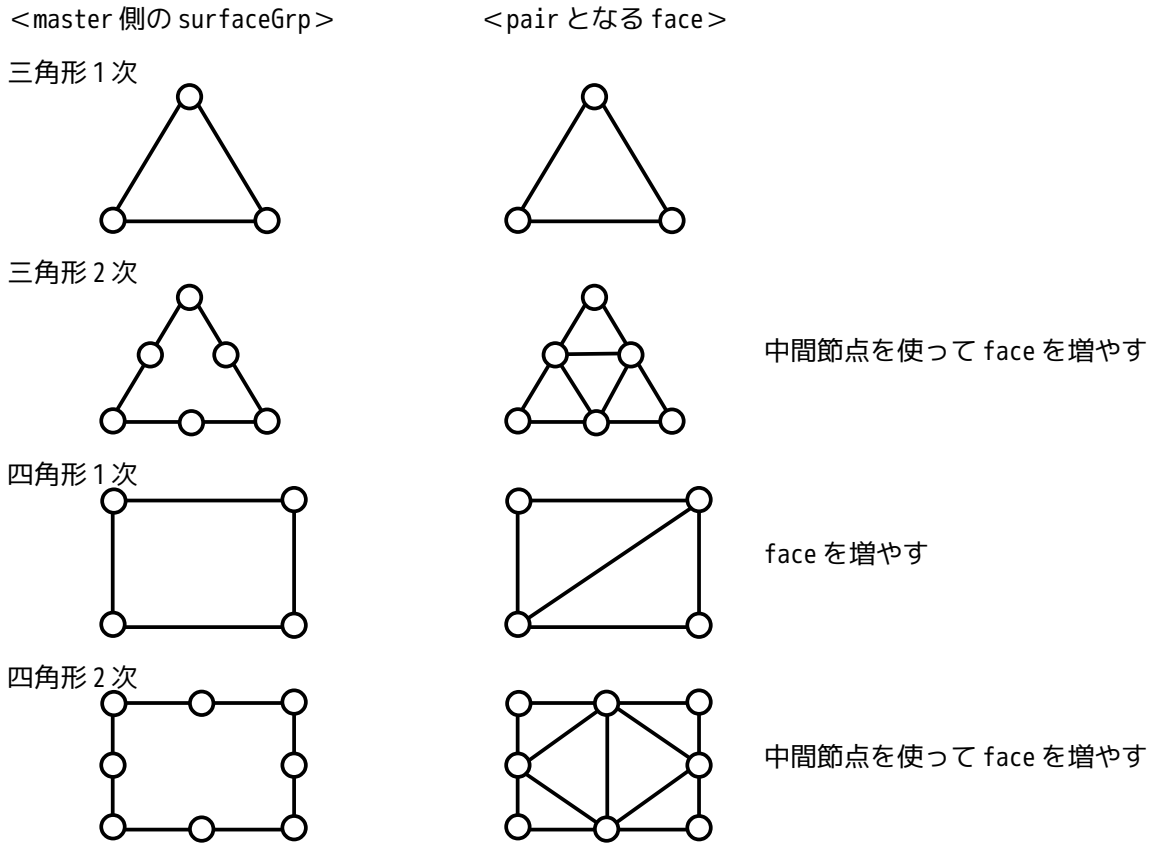
3-14-1-1.　結合の方法

slave 側の node と対応する master 側の face 面を pair として取得し、node の変位と face 面上の変位が一致する様に!EQUATION を作成する。

pair となる node と face は、以下に基づいて取得する。

node を face 面上に投影した点が face 面内に存在する事が pair の前提になる。この為、node を投影した点が master 側の全 face 面から外れる node は、結合の対象外となり、結合されない。
face 面から外れるかどうかは、face の辺の長さの 1/100 分の余裕を持って判断している。
従って、face の辺の長さの 1/100 以内の外れは、face 面内と判断して結合する。
また、face に投影した点と node の距離が一定値を超える場合も、結合の対象外となる。
投影した点と node の距離が face の辺の長さ以上離れた場合は、結合しない。
face の辺の長さは、1 次要素と 2 次要素では下図の様に変化する。

node と pair になる master 側の face は、三角形を基本としているので、六面体や 2 次要素の場合は、要素の面を以下の様に分割して pair となる三角形の face 面を取得し直している。



node の変位 δn は、face の頂点の変位 δ_1, δ_2, δ_3 の関数として以下の様に求めることができるので、これで!EQUATION が作成できる。

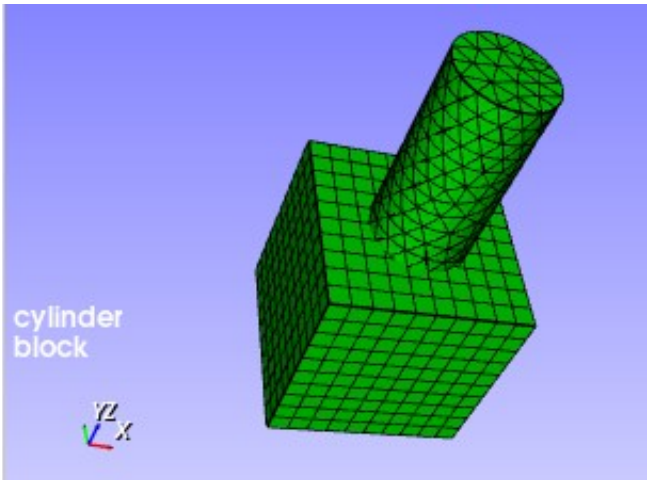
$$\delta_n = f(\delta_1, \delta_2, \delta_3)$$
$$\delta_n = C_1\delta_1 + C_2\delta_2 + C_3\delta_3$$

最終的に係数 C_1, C_2, C_3 を求めて!EQUATION を作成する。
作成した!EQUATION は、msh ファイルの最後に挿入される。

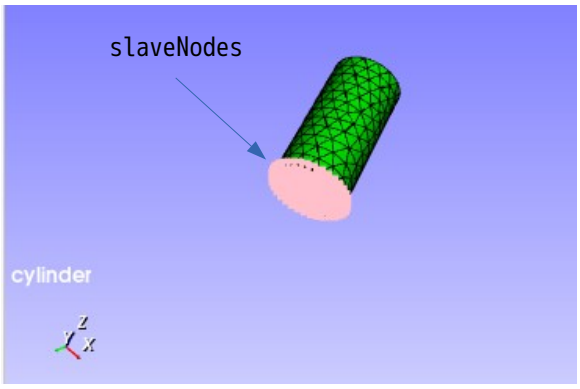
3-14-1-2.　結合の具体例

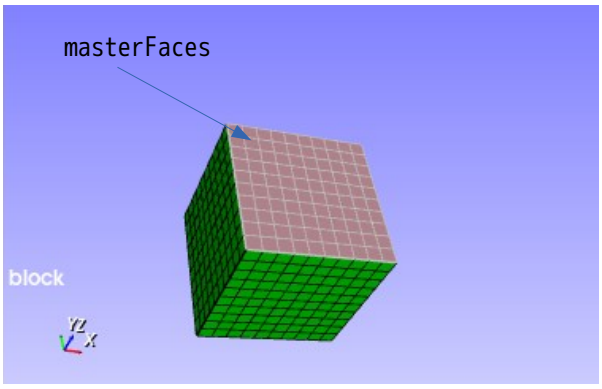
node を face 面上に結合する方法は、slave 側の nodeGrp と master 側の surfaceGrp を作成すれば、これらが結合できるので、自由度が高い。
今回、以下の様な六面体（block）と四面体 2 次要素（cylinder）を考えてみる。

メッシュは、easyIstr/unvFiles フォルダ内の「blockCylinderHexTet2nd.unv」を読み込み、スケールを 1/1000 に変更しておく。

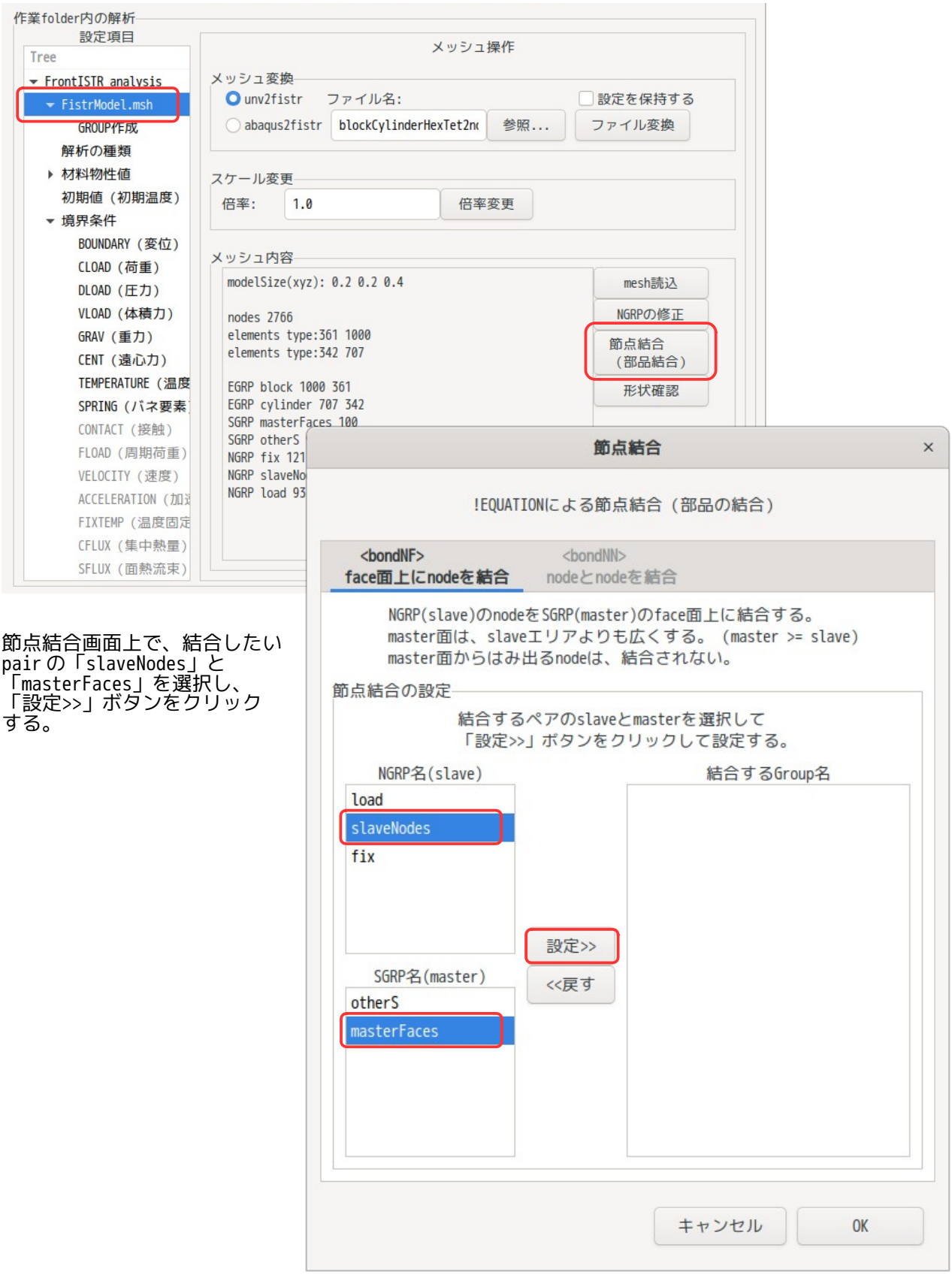


結合する pair は、
　　slave　slaveNodes　　(nodeGrp)
　　master masterFaces　　(surfaceGrp)
で設定している。



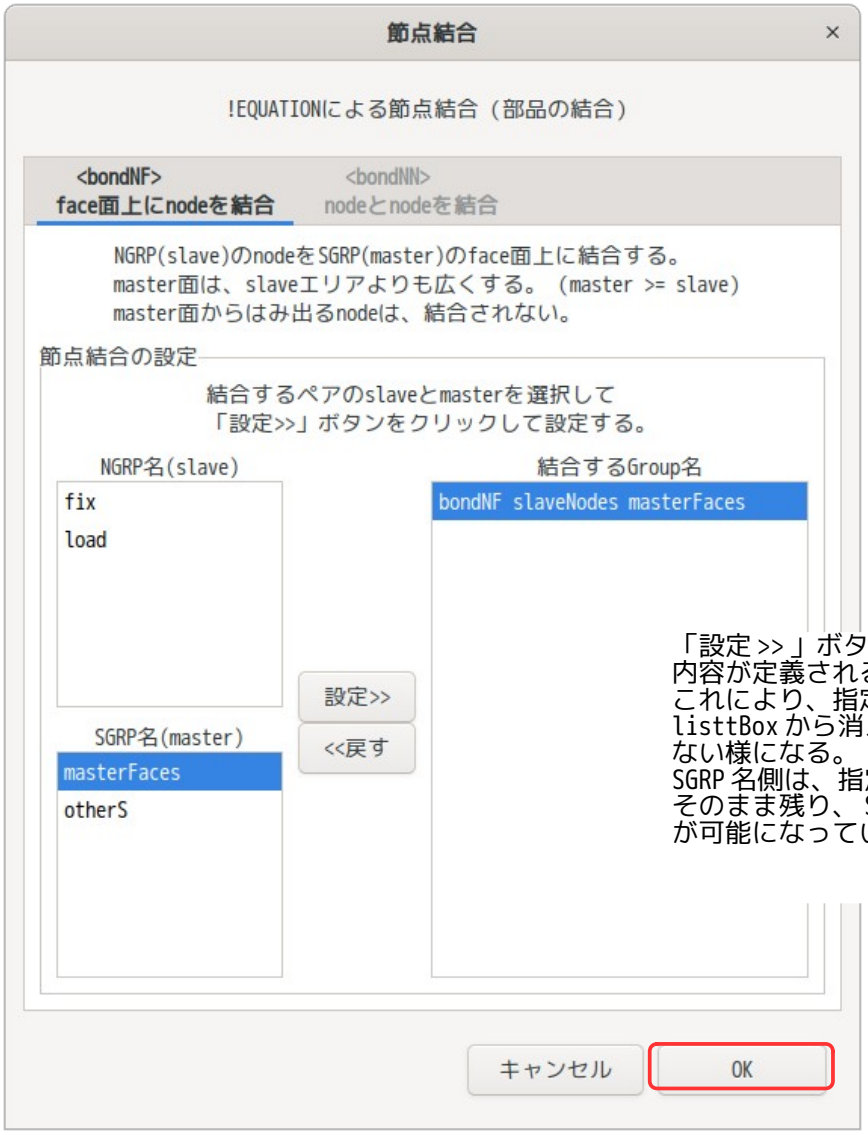


この slaveNodes と masterFaces を結合する。
その方法は、下図の様に、設定項目 Tree の「FistrModel.msh」を選択し、「節点結合（部品結合）」ボタンをクリックして「節点結合」画面を表示させる。
この画面内で、接合の設定を行う事になる。



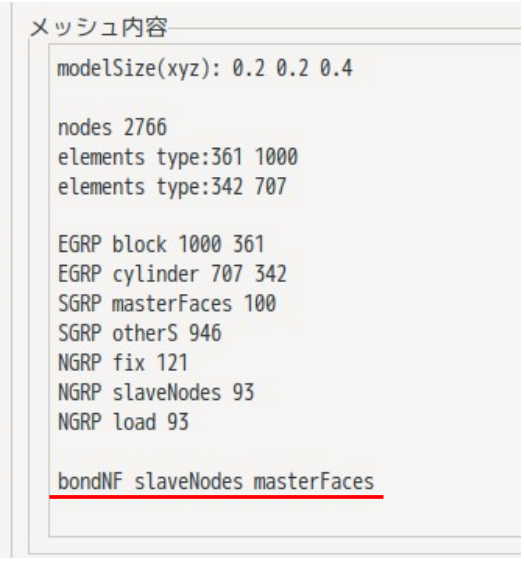
節点結合画面上で、結合したい pair の「slaveNodes」と「masterFaces」を選択し、「設定>>」ボタンをクリックする。

「設定>>」ボタンをクリックした結果が、以下の画面になる。
この後、「OK」ボタンをクリックする事で、「slaveNodes」と masterFaces」が結合される。
設定を解除したい場合は、「<<戻す」ボタンをクリックする事で、設定が解除できる。



「設定 >> 」ボタンのクリックで、設定内容が定義される。
これにより、指定した NGRP 名が listtBox から消えて、重複指定ができない様になる。
SGRP 名側は、指定した SGRP 名は消えず、そのまま残り、 SGRP 名側は、重複指定が可能になっている。

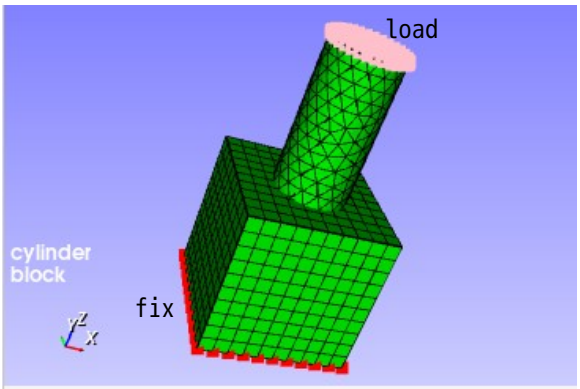
「slaveNodes」と「masterFaces」が結合されているかどうかは、右図の様に、メッシュ内容に「bondNF slaveNodes masterFaces」が表示される。
できあがった!EQUATION は、FistrModel.msh ファイルに追加されている。

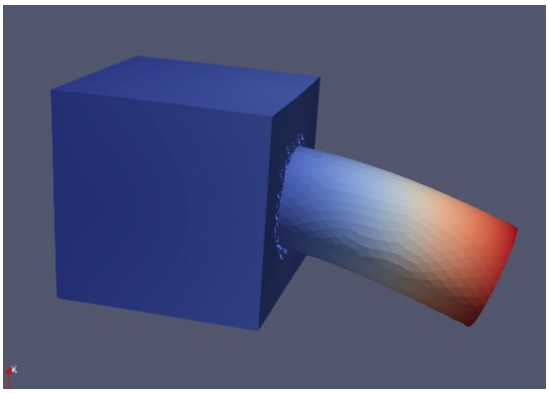


以下が、msh ファイルに追加された!EQUATION になる。

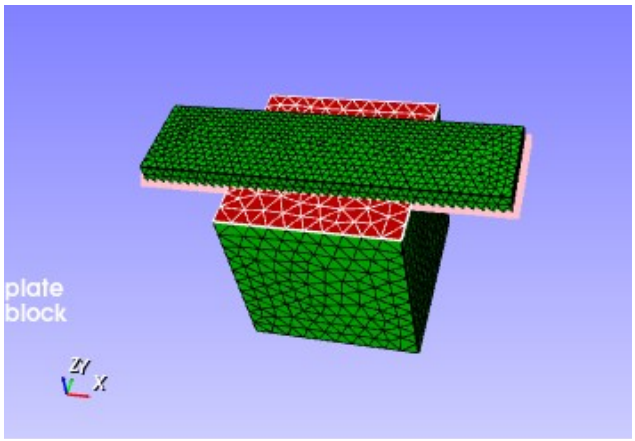
```
-------- FistrModel.msh の内容 ----------------------------------------------------------------
    :
#bondNF, slaveNodes, masterFaces
!EQUATION
3, 0.0
564, 1, 0.5, 565, 1, 0.5, 1332, 1, -1.0
3, 0.0
564, 2, 0.5, 565, 2, 0.5, 1332, 2, -1.0
3, 0.0
564, 3, 0.5, 565, 3, 0.5, 1332, 3, -1.0
4, 0.0
555, 1, 0.6470097496345083, 564, 1, 0.043291419087275095, 556, 1, 0.30969883127821657, 1334, 1, -1.0
4, 0.0
555, 2, 0.6470097496345083, 564, 2, 0.043291419087275095, 556, 2, 0.30969883127821657, 1334, 2, -1.0
    :
!END
-------------------------------------------------------------------------------------------------
```

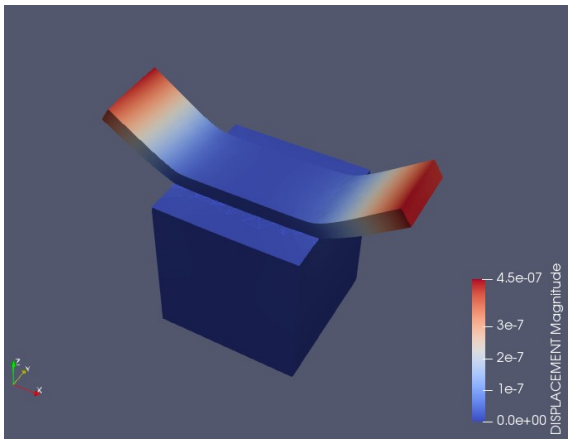
!EQUATION によって、slaveNodes と masterFaces が結合された後は、通常通り解析ができる。
以下は、材料を設定後、fix を固定して load に荷重をかけて、cylinder を曲げた結果になる。計算はうまくできている。
尚、solver を走らせる前に、線形 solver を「MUMPS」して、solver を走らせる事に注意する。
（「線形 solver が「CG」では、現状、計算できない。）





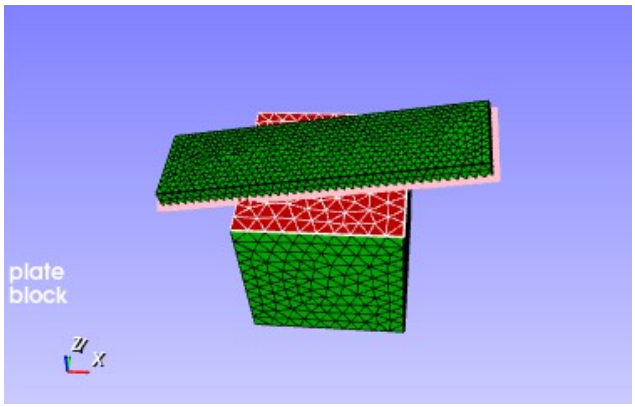
また、master の surfaceGrp からはみ出た slave の node は、結合されないが、これを確認する為に以下のモデルを作成して変位を確認してみる。
slaveNodes は、plate 下面の全 node、masterFaces は、block 上面の全 face を設定している。
計算は、plate 上面を引っ張る解析を行っている。この結果、masterFaces からはみ出た節点は、結合されていない事がわかる。

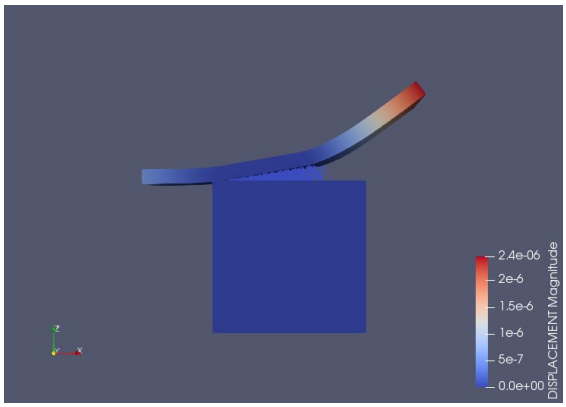




また、slaveNode が masterFace から離れすぎた場合、結合されないので、これを確認する為に、以下の様に plate を傾けたモデルを作成し、確認してみる。
計算は、plate 上面を引っ張る計算を行った結果、以下の様な結果であり、node が face 面から離れすぎると、結合されていない事がわかる。

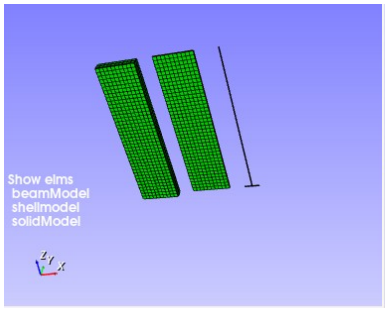
この離れ過ぎる時のしきい値は、master 側の face サイズによって変化する。

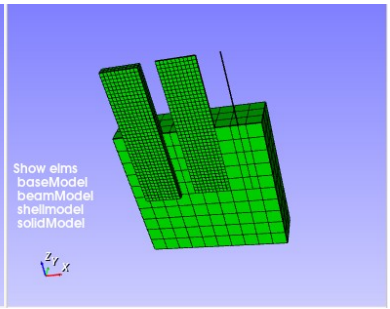


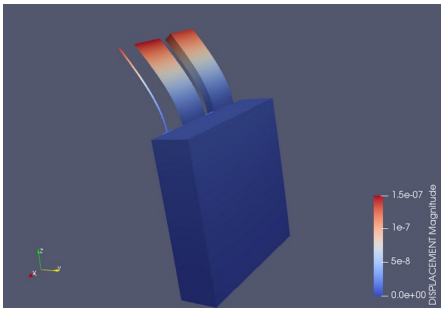


また、この結合は、節点グループ（slaveNode）の節点をソリッドの面グループ（masterFace）に結合する為、節点グループは、solid、shell、beam のいずれの節点グループでも構わない。（shell や beam もソリッドの面グループに貼り付ける事ができる。）

同じ形状の solid, shell, beam をベースの solid に結合してたわみを計算した結果が以下になる。
（beam は結合しても回転の自由度が残っているので、T 型の beam で結合している。）







たわみ量は solid, shell, beam とも同じ値になっている。

3-14-2.　節点同士を結合

結合したい nodeGrp の pair を指定して対応する node 同士を!EQUATION で結合する。
節点同士が同じ変位になる様に結合するので、pair となる nodeGrp 同士は、同じ形状で、同じ node 数が望ましい。

3-14-2-1.　結合の方法

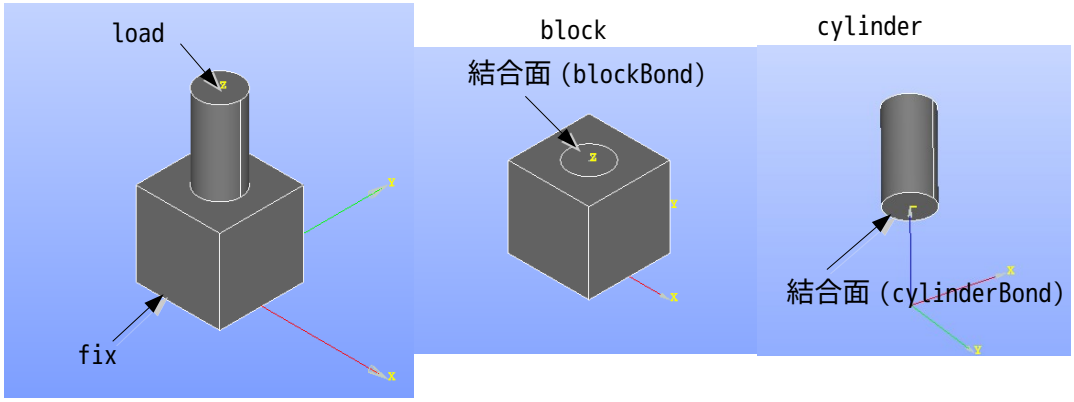
nodeGrp 同士の結合方法は、各々の結合面の節点を EQUATION で結合する。
結合面上で、最も近い節点同士（ペアの節点）を探し、これら 2 ヶの節点の各方向変位が同じ値になる様に EQUATION を作成する。（以下の例を参照。）

```
!EQUATION
2, 0.0
10, 1, 1, 100, 1, -1        #節点 10 の X 方向変位　ー　節点 100 の X 方向変位　＝　0.0
2, 0.0
10, 2, 1, 100, 2, -1        #節点 10 の Y 方向変位　ー　節点 100 の Y 方向変位　＝　0.0
2, 0.0
10, 3, 1, 100, 3, -1        #節点 10 の Z 方向変位　ー　節点 100 の Z 方向変位　＝　0.0
```

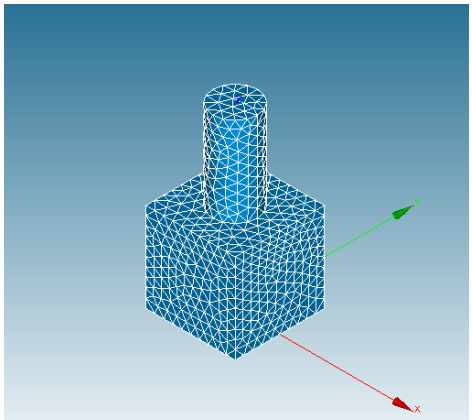
結合面上の全節点についてペアとなる節点を求め、EQUATION を作成する。（msh ファイル内に作成する）
この為、各々の結合面上の節点の数は、同数が望ましいが、異なっていても結合できる。

3-14-2-2.　結合の具体例

以下のモデルを考える。



block、cylinder のメッシュを結合面で結合させる。
上記モデルを salome でメッシュを作成する。以下がそのメッシュで、これを「blockCylinder.unv」として保存している。

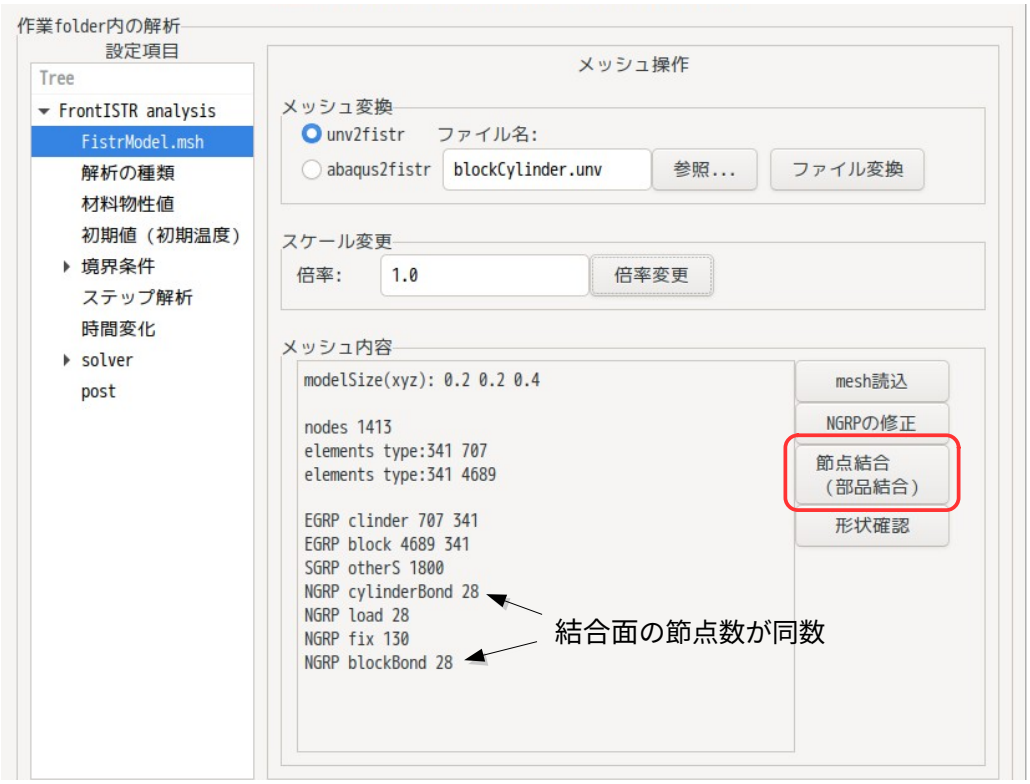


EasyISTR 上で、メッシュ変換し、スケールを 1/1000 にしておく。
この状態で、結合面の節点数を確認すると、

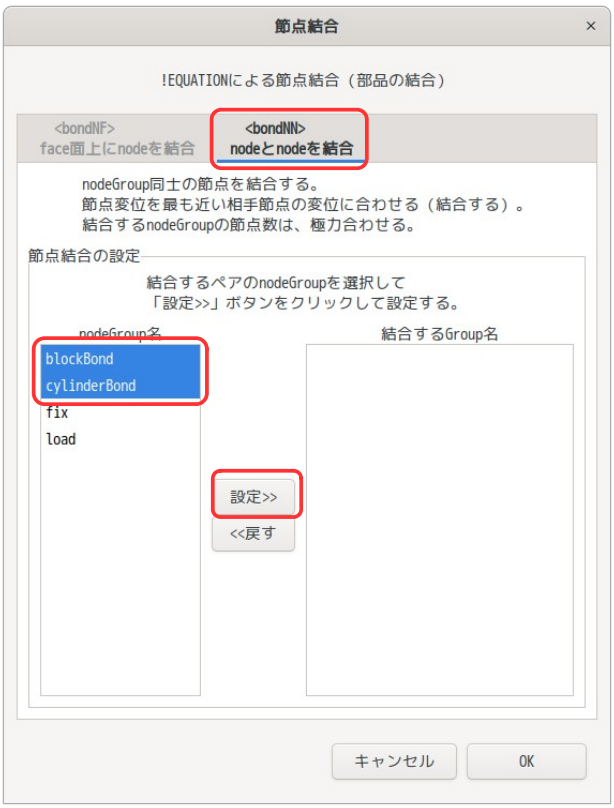
NGRP 名	節点数	備考
blockBond	28 ヶ	block 側の結合面
cylinderBond	28 ヶ	cylinder の結合面

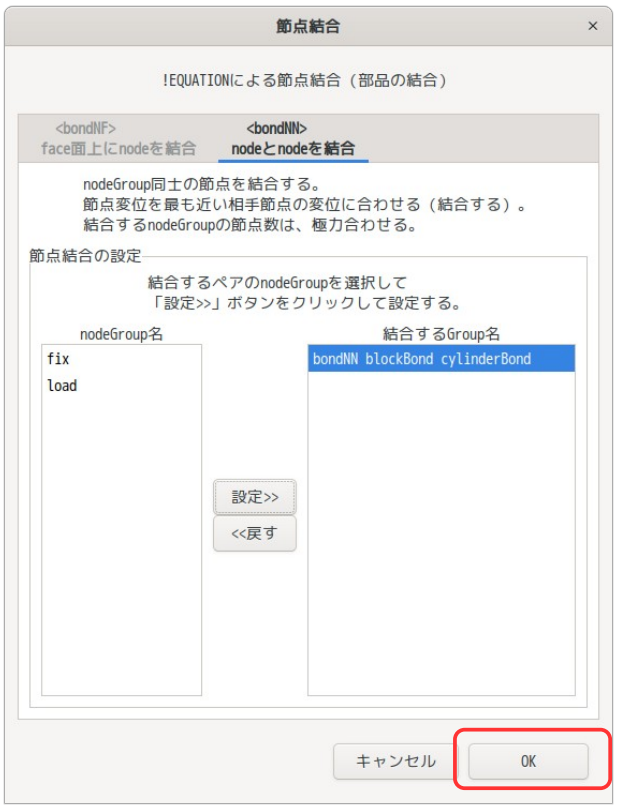
であり、結合面の節点数は、同数になっている。

この後、2 部品を EasyISTR 上で結合する。
まず、「節点結合（部品結合）」ボタンをクリックする。



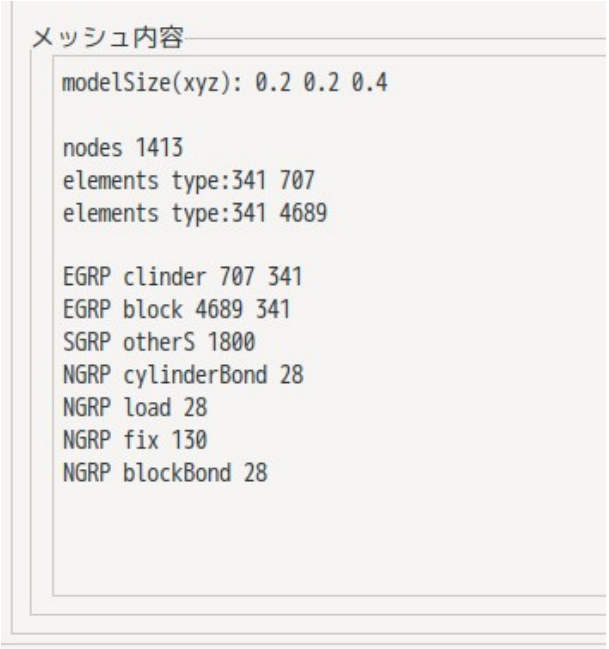
この後、「node と node を結合」タグを選択し、結合する 2 面を選択して、「設定>>」ボタンをクリックし、「OK」ボタンをすると 2 面が結合される。



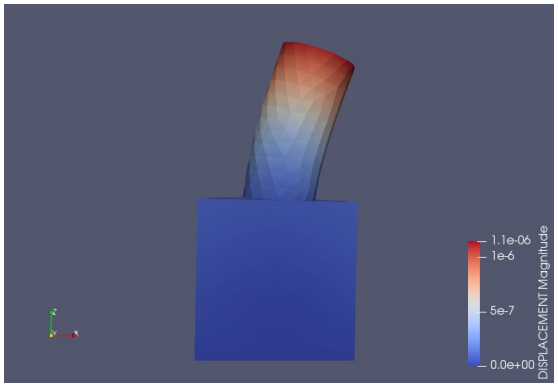


結合されると、以下のように「bondNN blockBond cylinderBond」がメッシュ内容に表示される。

結合前　　結合後



これで、通常通り、計算ができる。
材料は Al とし、境界条件は fix を固定し、load に X 方向計 100N の等分布荷重を印加してみる。
以下が変位の計算結果になる。（部品の結合ができているので、うまく計算できている。）

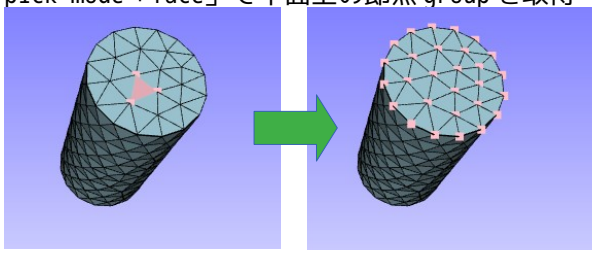


この節点を EQUATION で結合する方法は、節点を結合するので、要素 type は問わない。
この為、四面体 1 次要素と四面体 2 次要素でも接合できる。また、四面体と六面体も接合できる。
ただし、接合面の節点密度（節点数）は、合わせる必要がある為、使いづらい。この為、部品同士を結合する場合は、前項の「3-14-1. 節点と相手 face 上に結合」を使った方が楽。

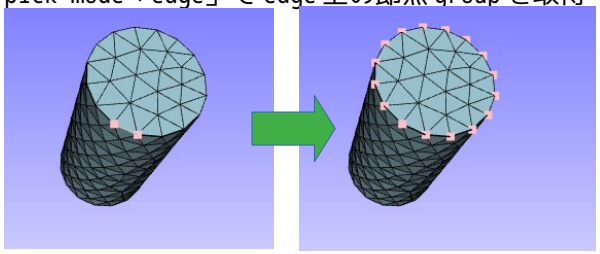
3-15. group の作成、名称変更、削除

EasyISTR 上で、メッシュに group（NGRP, SGRP, EGRP）を新たに作成する事ができる。
また、既にできあがっている group（NGRP, SGRP, EGRP）名を変更したり、削除する事もできる。
group 作成においては、pick mode を変更する事で、以下の様に group を取得することができる。

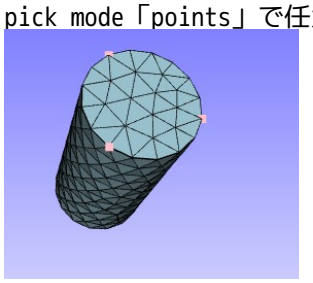
pick mode「face」で平面上の節点 group を取得



pick mode「edge」で edge 上の節点 group を取得



pick mode「points」で任意の節点 group を取得

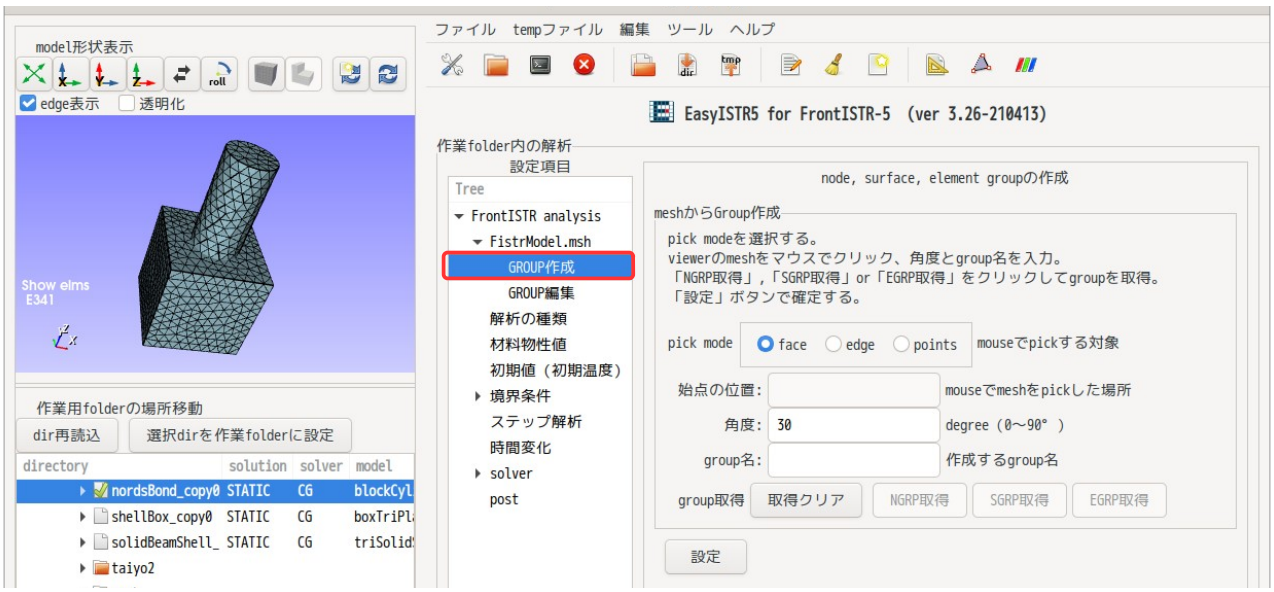


メッシュが solid、shell、beam で NGRP、SGRP、EGRP が取得できる。
また、存在する各グループに対し、pick で取得したグループを追加（加算）したり、削除（減算）する事もできる。
次項で、solid メッシュ、pick mode: face を使って、NGRP、SGRP、EGRP を取得する例を示している。

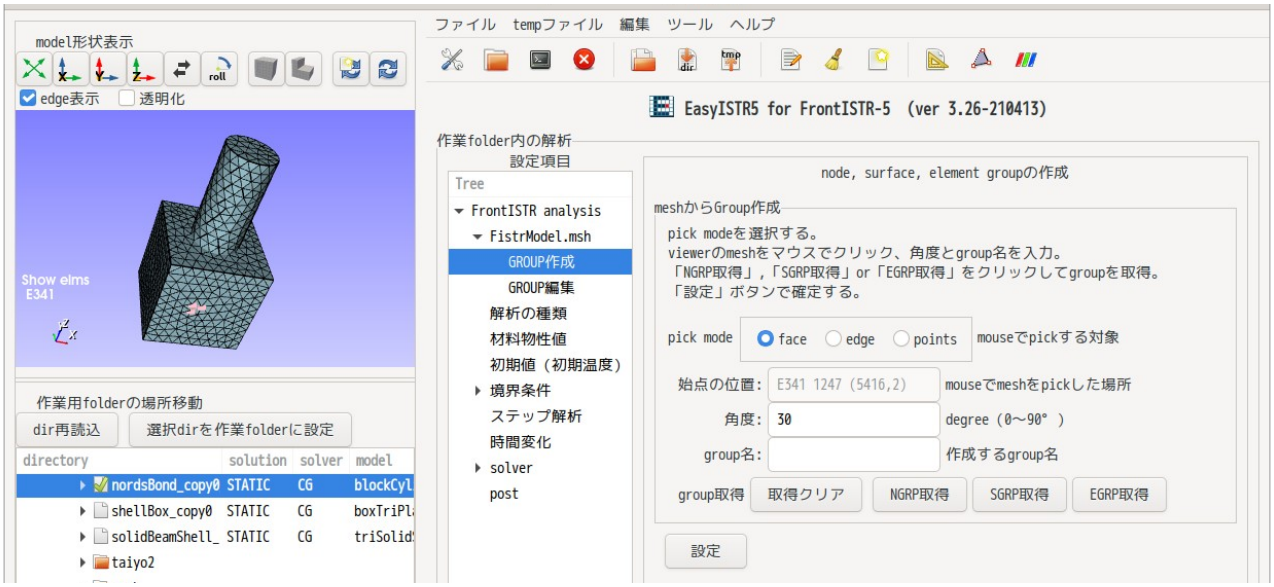
3-15-1.　節点(NGRP)、面(SGRP)、要素(EGRP) group の作成

easyIstr/unvFiles フォルダ内に、「blockCylinderEmp.unv」があるので、これを使って各 group を作成してみる。
この unv ファイルは、メッシュのみ作成してあり、group が作成されていない。
この unv ファイルを使って、要素 group、節点 group、面 group を作成してみる。
この為、「blockCylinderEmp.unv」ファイルを EasyISTR 上で読み込み、メッシュ変換して、1/1000 にスケール変更しておく。
（メッシュ変換時に全体の要素 group「E341」と全体の面 group「otherS」が定義される。）

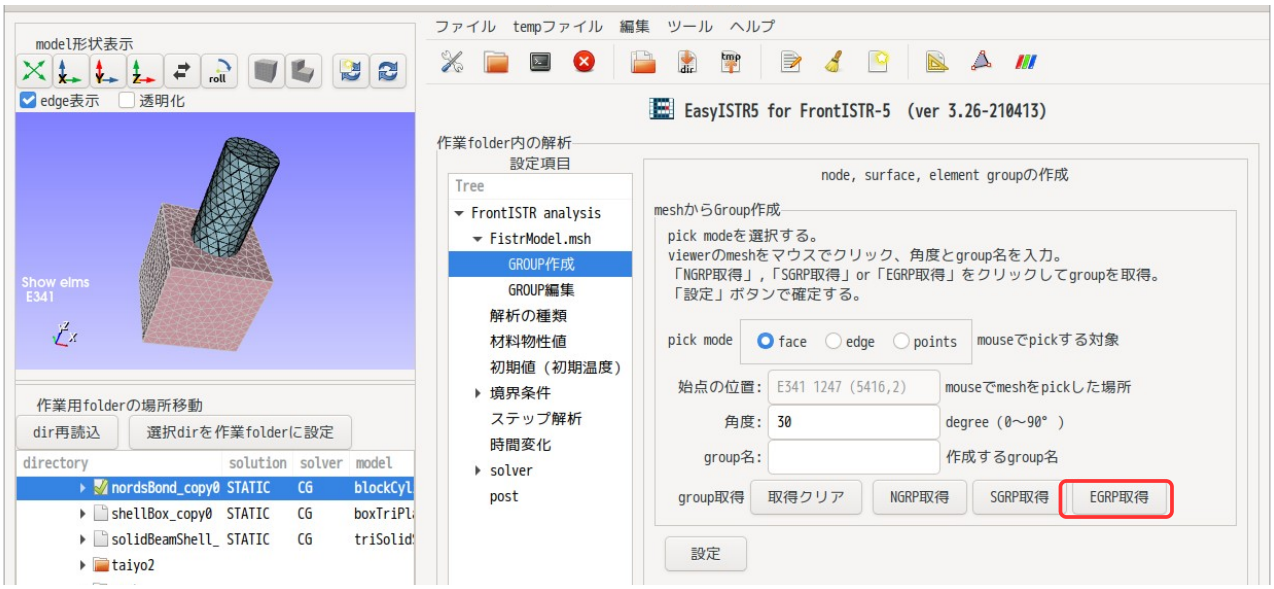
<要素 group の作成>
まず、要素 group の「block」と「cylinder」を作成してみる。
EasyISTR 上の設定項目 Tree 上で「FistrModel.msh」>「GROUP 作成」を選択する。この画面上で要素 group を作成する。下図参照
unv ファイルを変換した時点で、メッシュ全体の要素 group が「E341」で定義されており、この名称が vtk 画面に表示されている事が確認できる。



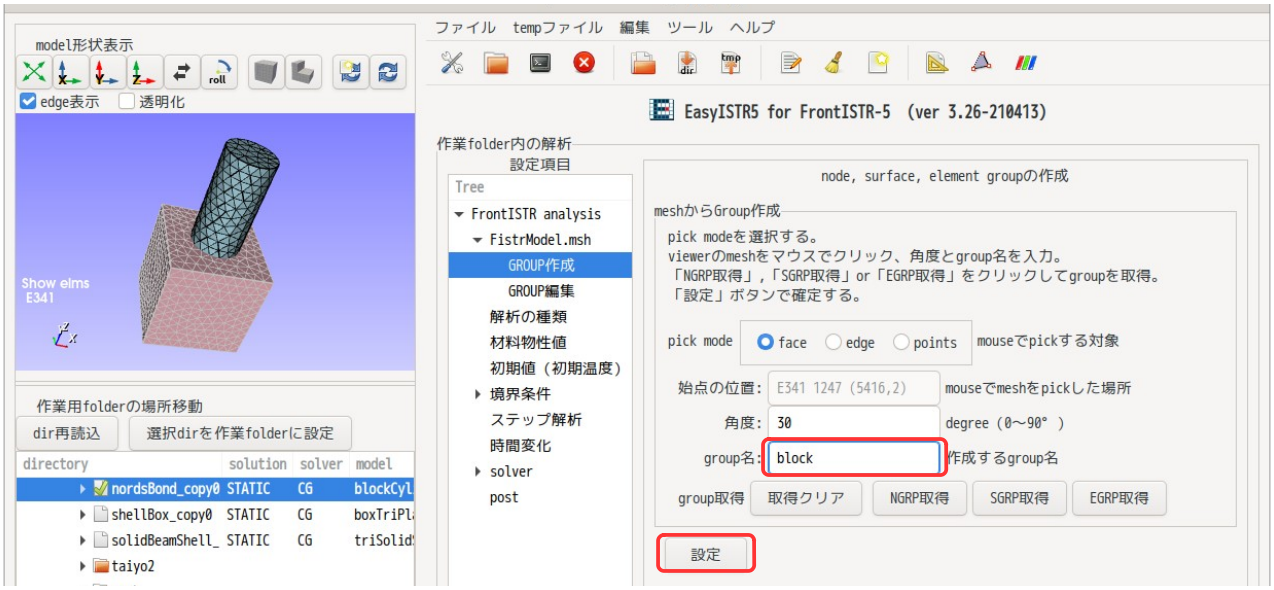
この後、pick mode が「face」である事を確認して、選択したい要素 group の面をマウスでクリックする。
クリックした部分の face の色が下図の様に変わる。



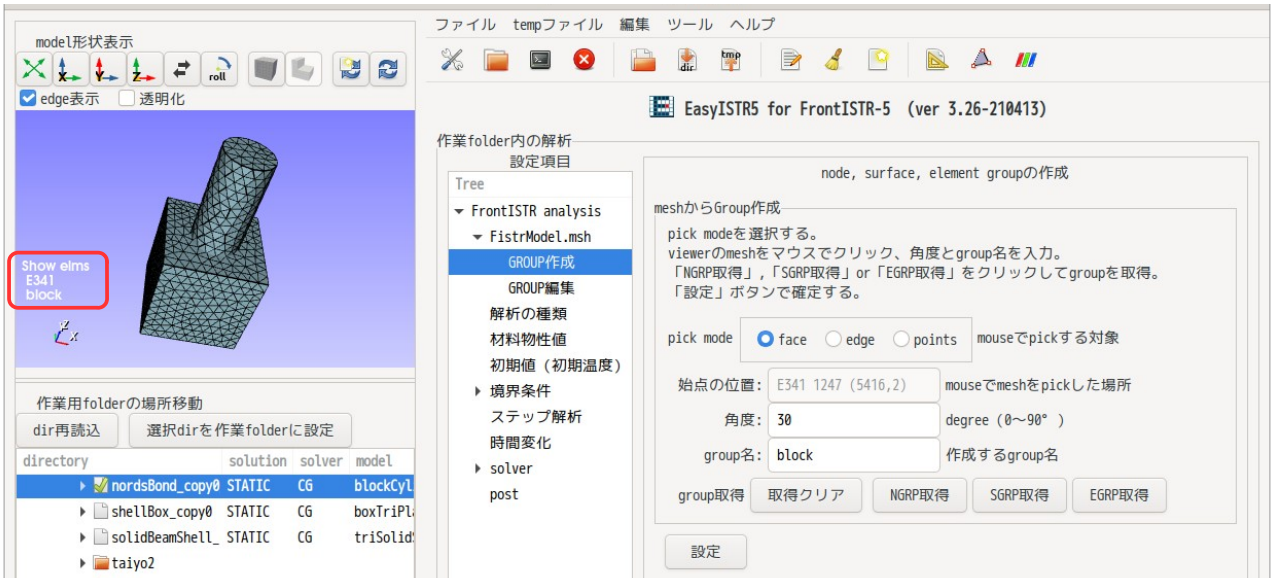
この後、「EGRP 取得」ボタンをクリックすると、選択した部分の face 面を基準にして、独立している要素 group が下図の様に取得できる。



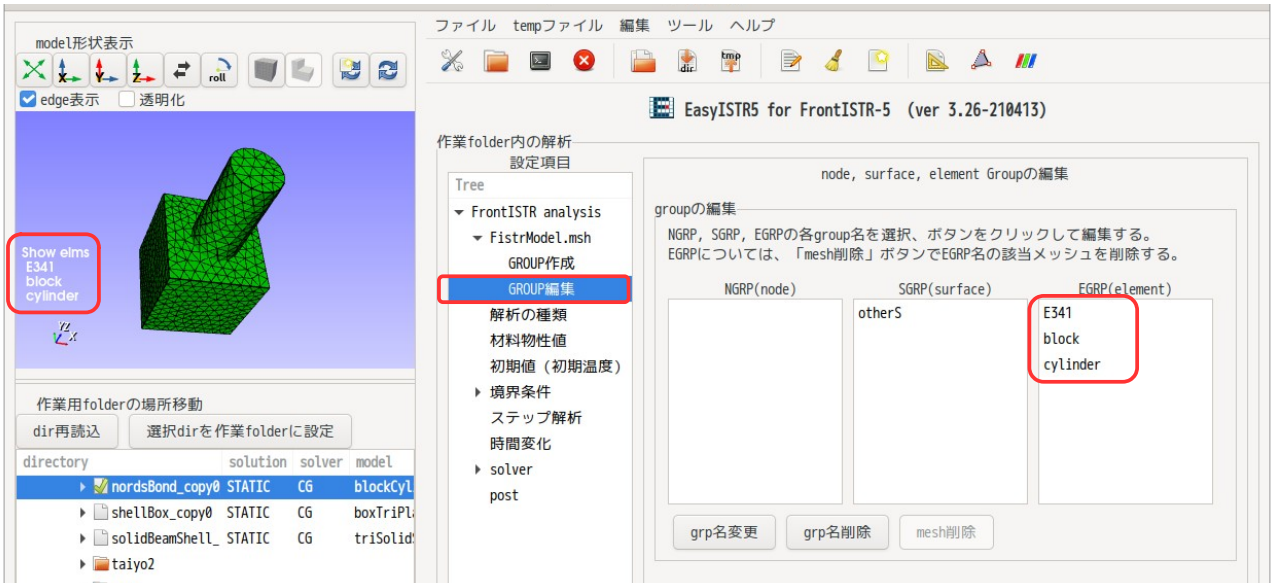
この後、作成する group 名「block」を入力し、「設定」ボタンをクリックして要素 group を作成する。



下図が要素 group「block」ができあがった画面になる。（vtk 画面上に要素名 block が表示されている。）

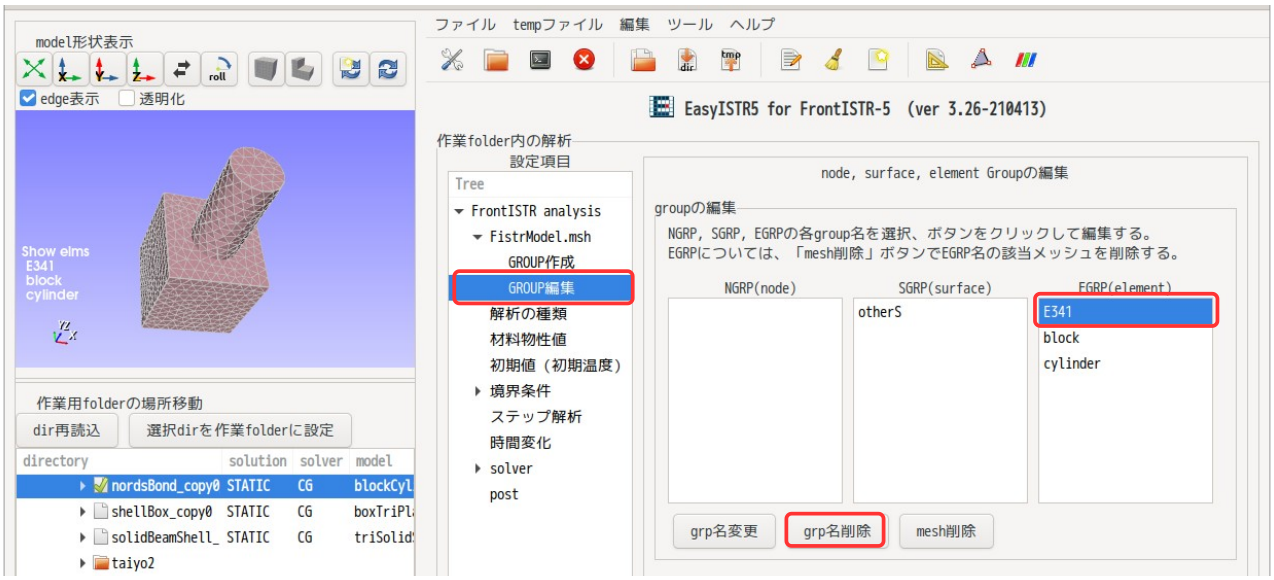


同様にして、円柱側を要素 group「cylinder」として定義した結果が以下になる。
下図は、設定項目 Tree 上で「GROUP 編集」画面を表示させた状態。vtk 画面上に E341, block, cylinder が確認できる。また、3 種類の要素 group が確認できる。

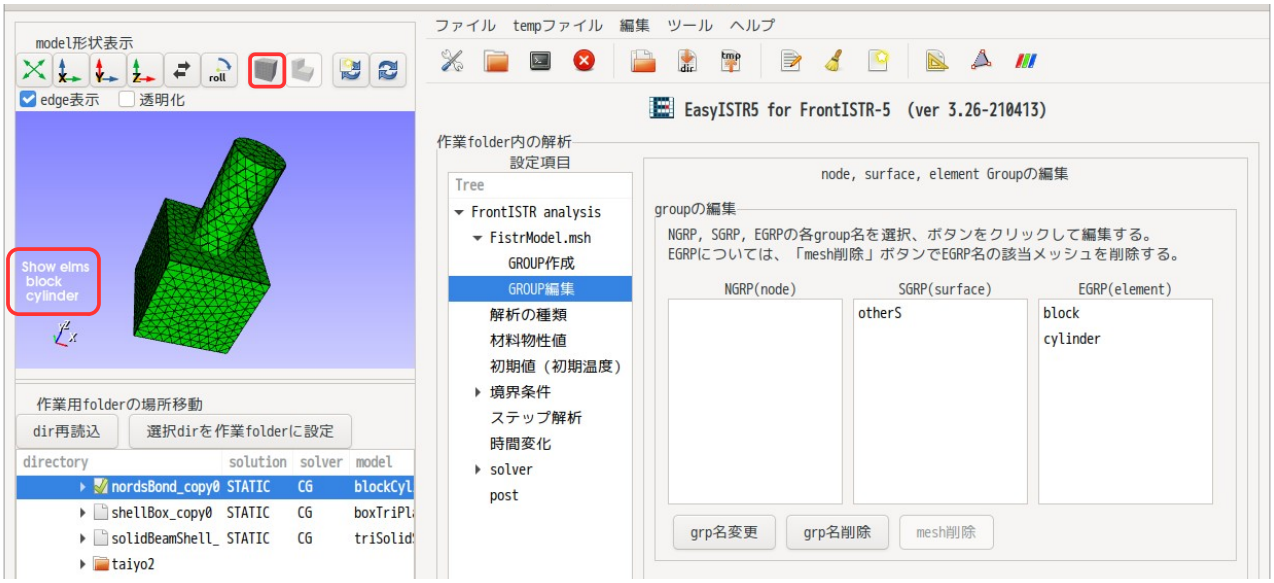


vtk 画面は、要素 group 毎に描画している。vtk 画面上に表示されている要素 group 名は、vtk 画面上に表示させている要素 group 名を表している。現状は、E341, block, cylinder の要素 group が表示されている事になる。つまり、全体の要素 group「E341」が描画されているので、block、cylinder の要素に重なって E341 も表示されている。
この状態は、vtk 画面上でモデル上をクリックしてピックした時、要素 group が重なっている為、どちらの要素 group をピックしたのかが判らなくなる。

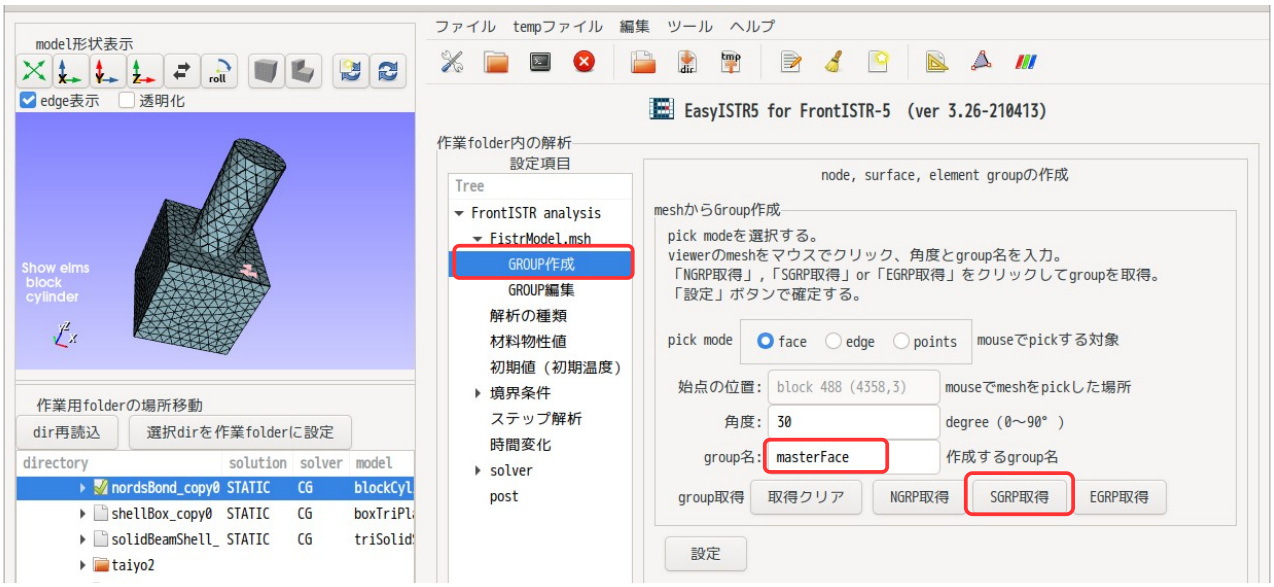
この為、全体の要素 group「E341」を削除するか、非表示設定にしておく必要がある。
ここでは、E341 の grp 名を削除しておく事にする。（「grp 名削除」は、EGRP 名の定義が削除されるのみで、要素メッシュが削除される訳ではない。）
その方法は、下図の様に GROUP 編集画面を表示させ、要素 group「E341」を選択後、「grp 名削除」ボタンをクリックして削除する。
これにより、要素 group「E341」が削除される。



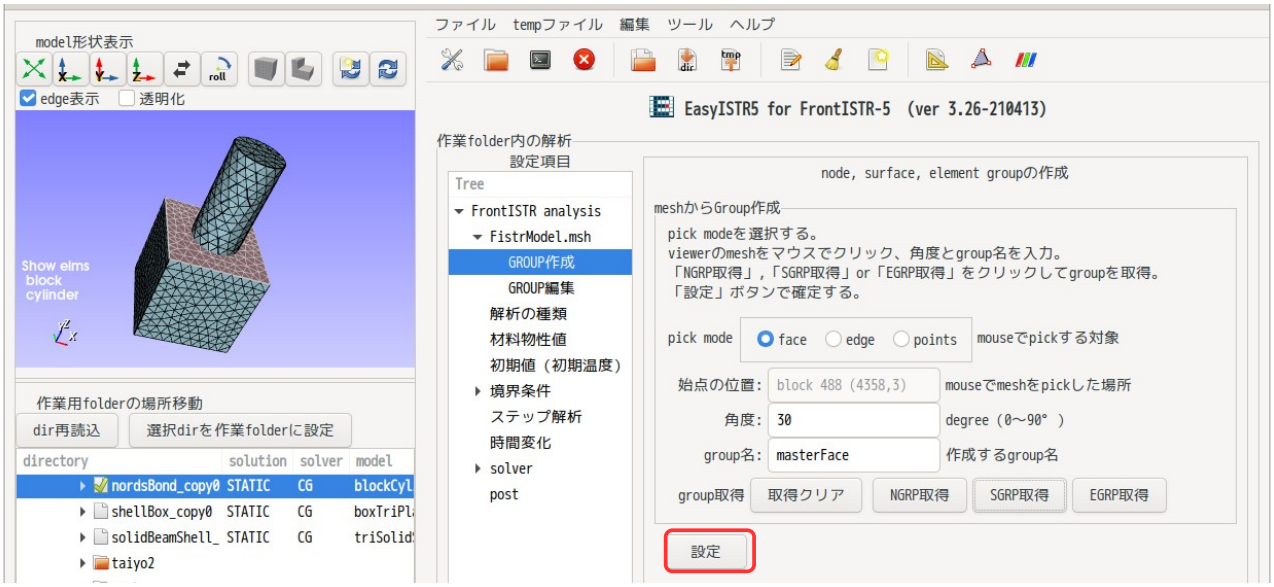
下図が、要素 group「E341」を削除した画面になる。vtk 画面に表示されている要素 group が block, cylinder である事が確認できる。
　要素 group「E341」は、!ELEMENT で定義されている要素 group の為、これを削除すると、同じ形状の「--undefine--」という名称の vtk データができあがる。この為、--undefine--を表示させると、E341 と同じ形状が表示させる事ができる。
　（!EGROUP で定義されている要素 group を削除しても「--undefine--」は、作成されない。）
　「--undefine--」は、下図の全表示アイコンをクリックする事で、再表示させる事ができる。



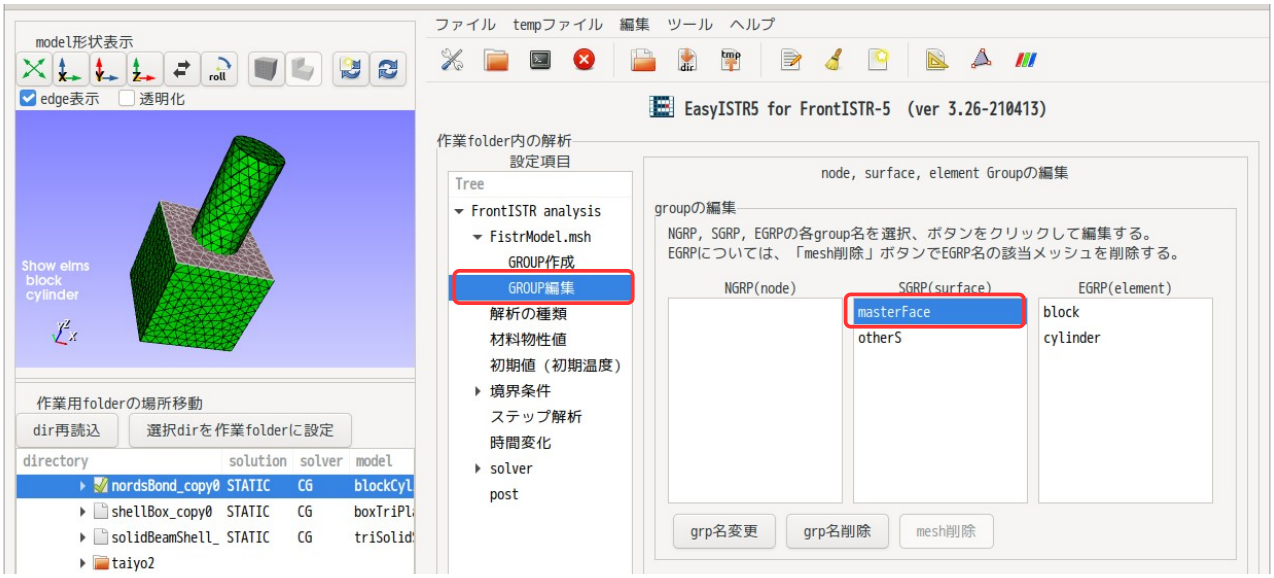
<面 group の作成>
要素 group「block」の上面を面 group「masterFaces」で定義してみる。
GROUP 作成画面上で、該当箇所をマウスでクリックして、face をピックする。（下図参照）
この後、作成する group 名「masterFaces」を入力後、「SGRP 取得」ボタンをクリックして面を取得する。



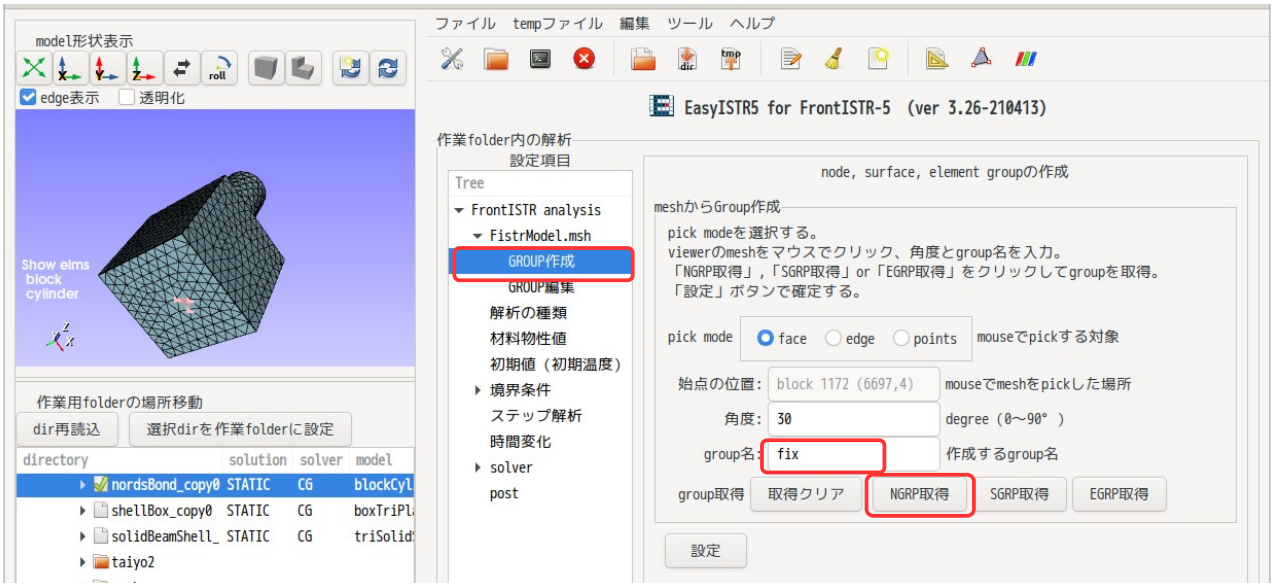
下図が、block の上面を取得した結果になる。面がうまく取得できている。
もし、思惑通りの面が取得できていない場合は、face 角度の値を変えて「SGRP 取得」ボタンをクリックして再取得してみる。
思惑通りの面が取得できた場合は、「設定」ボタンをクリックして、確定する。



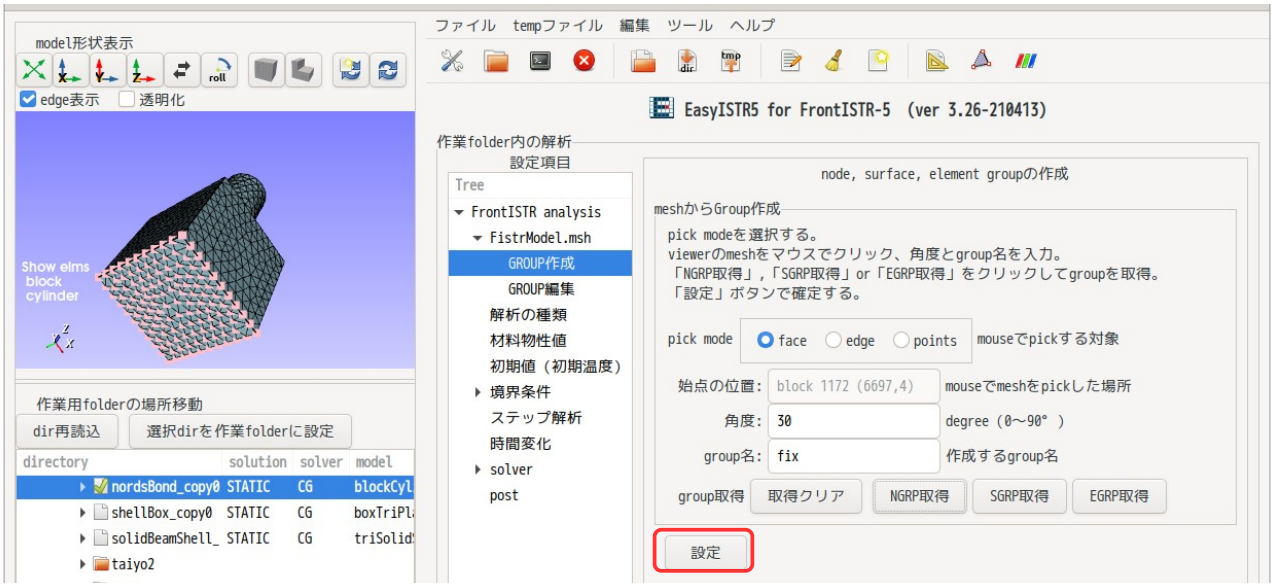
下図が取得した結果になる。GROUP 編集画面で、取得できたかどうかが確認できる。



<点 group の作成>
要素 group「block」の下面、「cylinder」の下面と上面をそれぞれ、「fix」、「slaveNodes」、「load」として定義してみる。
まず、block の下面を点 group「fix」で定義する。
下図の様に、block の下面の face をマウスでピックして、作成する group 名「fix」を入力して「NGRP 取得」ボタンをクリックする。

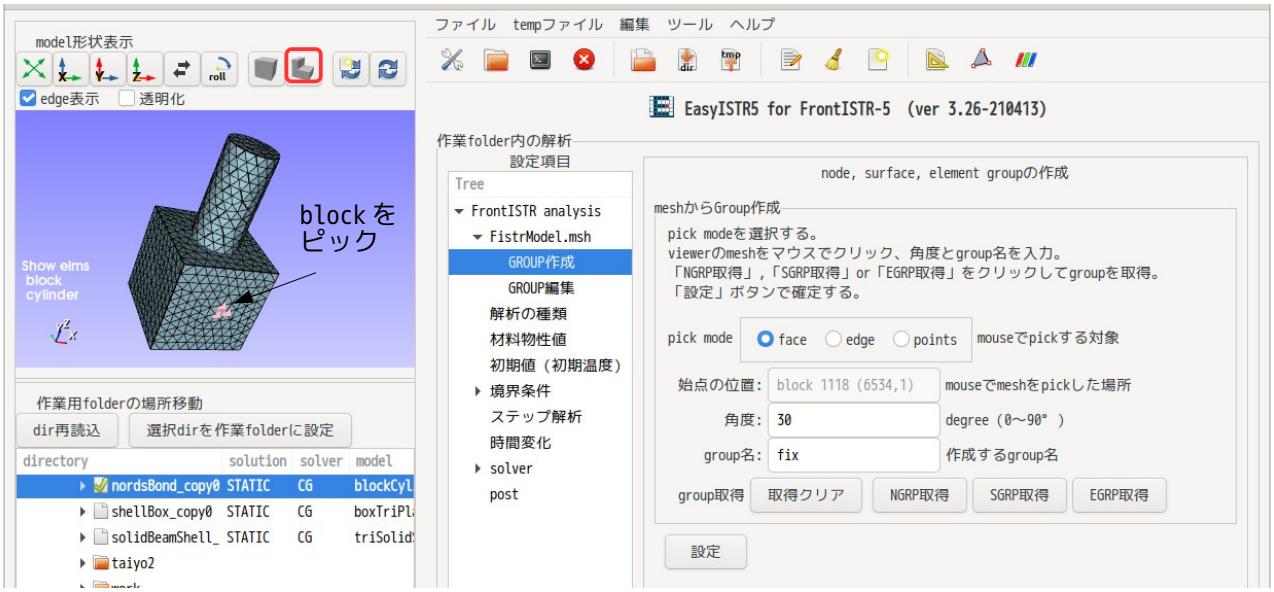


下図が、点 group「fix」を取得した結果になる。この後、「設定」ボタンをクリックして、確定させる。

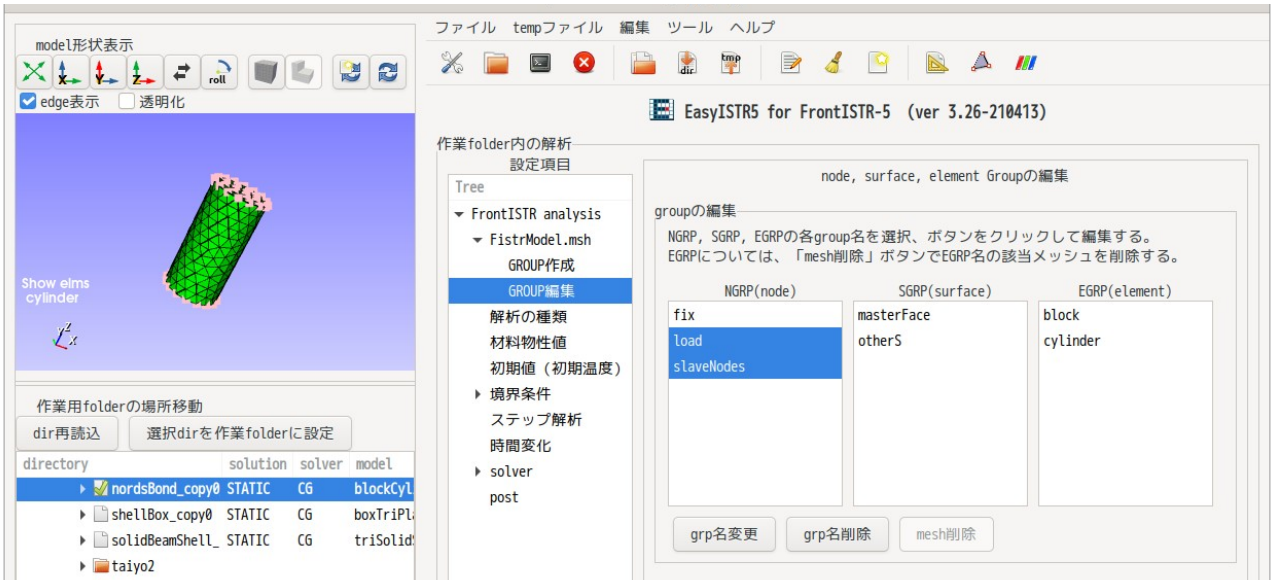


これにより、点 group「fix」ができあがる。
cylinder の上下面に点 group を作成してみる。この場合は、block が邪魔で cylinder の下面がピックできない為、block を非表示設定する。

非表示設定の方法は、対象の要素 group を選択（対象の要素の色がピンク、赤、薄茶色の状態）した状態で、非表示アイコンをクリックすると選択要素 group が非表示となるが、GROUP 作成画面においては、非表示したい要素要素 group の face をピックして、そのまま、非表示アイコンをクリックしても、非表示にできる。この為、下図の様に、block の face を選択し、非表示アイコンをクリックして、block が非表示にできる。



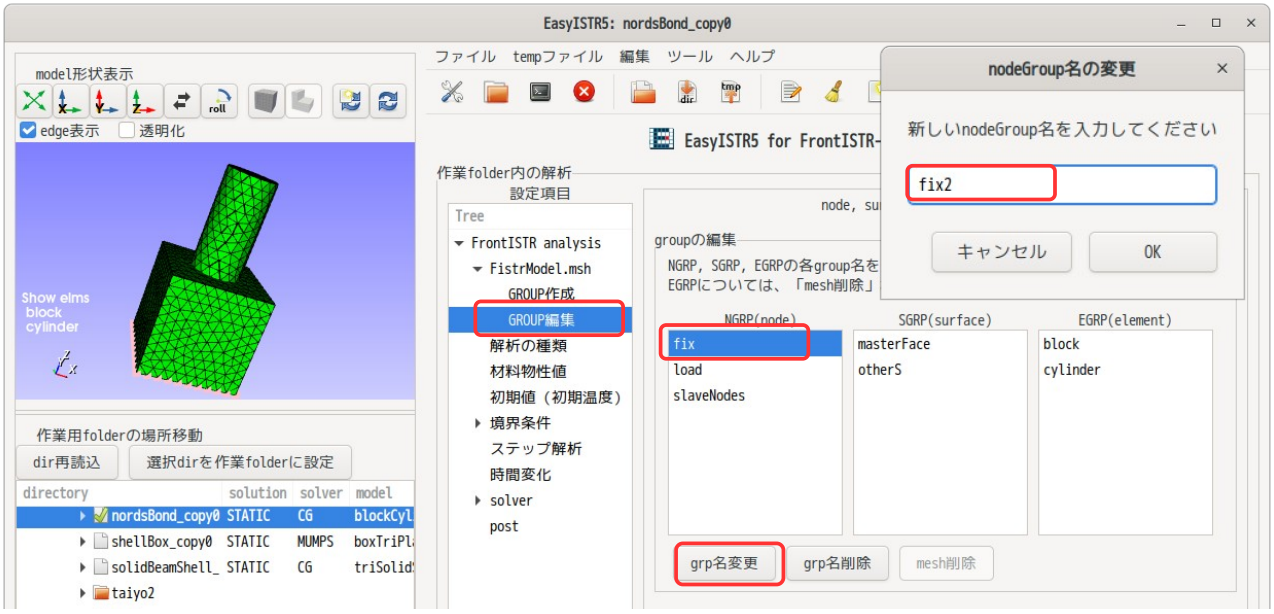
非表示設定後は、cylinder の上下面がピックできるので、各々「slaveNodes」、「load」で点 group が作成できる。
以下が slaveNodes と load の点 group を確認した画面になる。（load, slaveNodes が確認できる。）



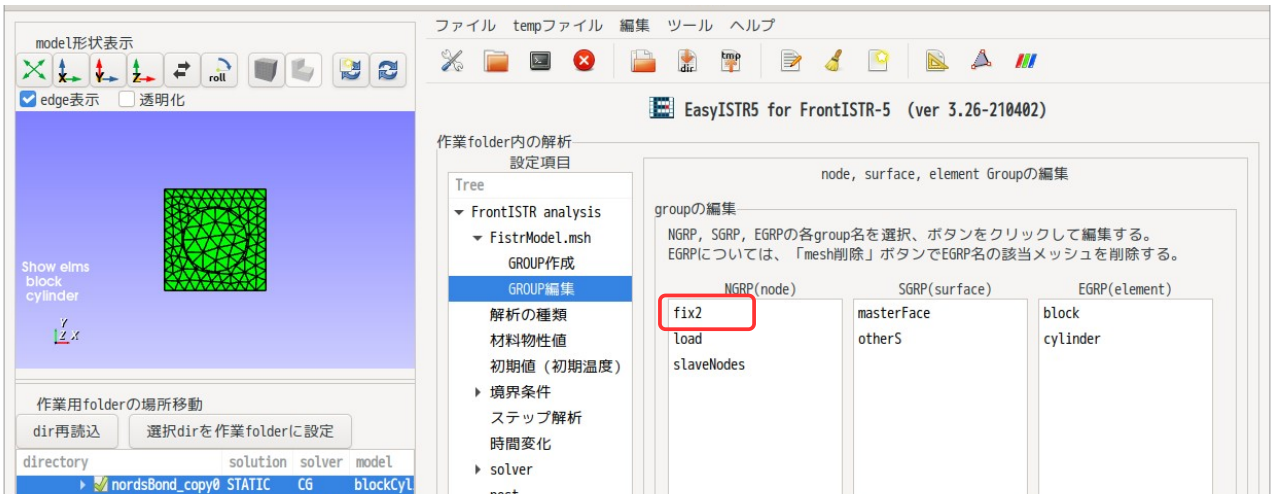
3-15-2.　節点(NGRP)、面(SGRP)、要素(EGRP) group の名称変更、削除

各 group の名称変更、削除は、GROUP 編集画面で設定する事になる。
節点 group を例として説明する。

節点 group「fix」を「fix2」に名称変更してみる。
まず、以下の様に対象の節点 group「fix」を選択し、「grp 名変更」ボタンをクリックして、新しい group 名「fix2」を入力する。（下図参照）



これにより、以下の様に名称が変更できる。



group の削除は、「grp 名削除」ボタンをクリックして、選択した group が削除できる。
この削除は、group 名が削除されるのみで、要素は削除されない。

尚、これらの名称変更、削除は、対象の group に境界条件等が設定されていない状態でこれらを実行する。もし、境界条件等が設定されている状態で、名称変更、削除を実行すると、その設定内容がクリアされてしまい、再設定の必要が生じる。

3-16.　その他

3-16-1.　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が、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が判断できる様した。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、大変形）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
iter 回数
最大、最小温度（max tenperature、min tenperature）

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	iter 回数

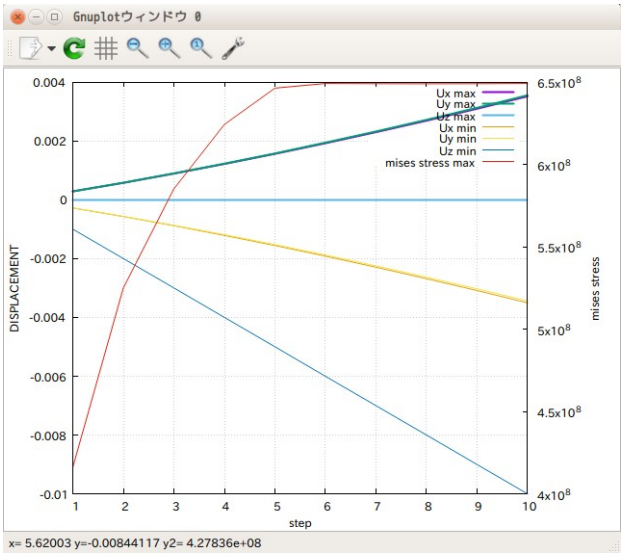
plotStepMonitor は、5s 間隔でこれらファイルの内容を読みにいき、値を取得して、gnuplot を使って逐次グラフ表示していく。

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。

```
$ plotStepMontor.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。



最大、最小変位がグラフ表示され、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。

3-16-2.　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。

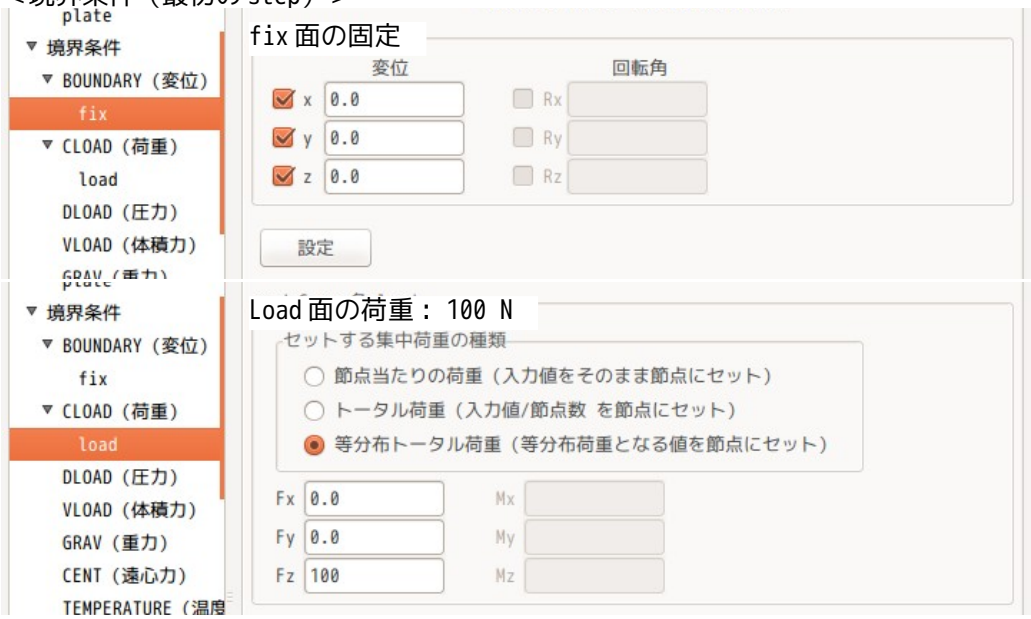
これにより、設定項目 Tree 内に「境界条件 STEP1」と「境界条件 STEP2」が追加されるので、この中に各々の step 内で使用する境界条件を入力する。

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

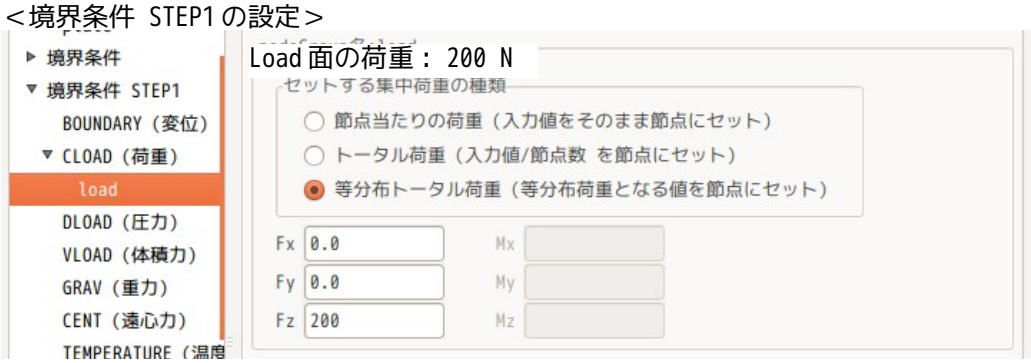
境界条件 STEP2　　#3 番目の step で使用する境界条件を設定

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。

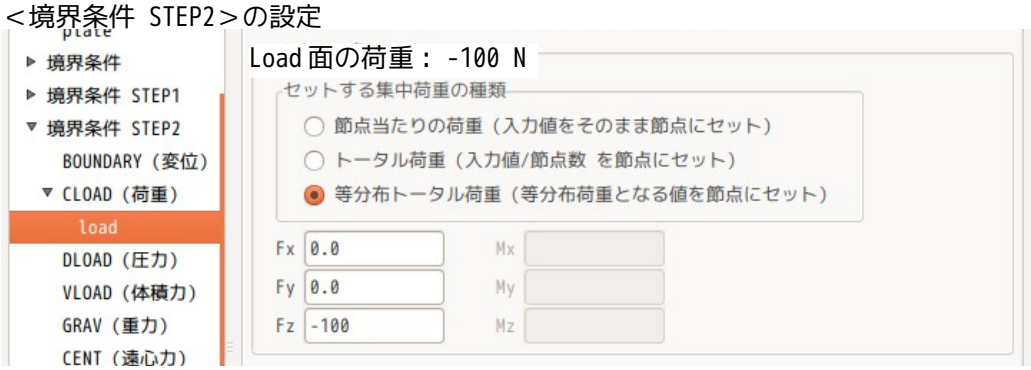
＜境界条件（最初の step）＞



＜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
初期値（初期温度）
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
自動時間増分を使う（動解析では使えません）
TYPE STATIC
CONVERG 1e-6
SUBSTEPS 5
MAXITER 1000
MAXCONTITER 10
DTIME ETIME minDT maxDT
1.0 1.0 0.1 1.0
時間幅減少条件: 0.50,10,50,10,1
時間幅増加条件: 1.41,10,50,10,1
cutBack条件: 0.50,10
編集...
step内の計算結果保存個数（等間隔） 1
step解析する境界条件
このstepまでの境界条件
設定する境界条件
BOUNDARY,fix,STEP0
CLOAD,load,STEP0
選択>>
<<戻す

＜STEP1＞

FrontISTR analysis
FistrModel.msh
解析の種類
材料物性値
初期値（初期温度）
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
自動時間増分を使う（動解析では使えません）
TYPE STATIC
CONVERG 1e-6
SUBSTEPS 5
MAXITER 1000
MAXCONTITER 10
DTIME ETIME minDT maxDT
1.0 1.0 0.1 1.0
時間幅減少条件: 0.50,10,50,10,1
時間幅増加条件: 1.41,10,50,10,1
cutBack条件: 0.50,10
編集...
step内の計算結果保存個数（等間隔） 1
step解析する境界条件
このstepまでの境界条件
CLOAD,load,STEP0
設定する境界条件
BOUNDARY,fix,STEP0
CLOAD,load,STEP1
選択>>
<<戻す

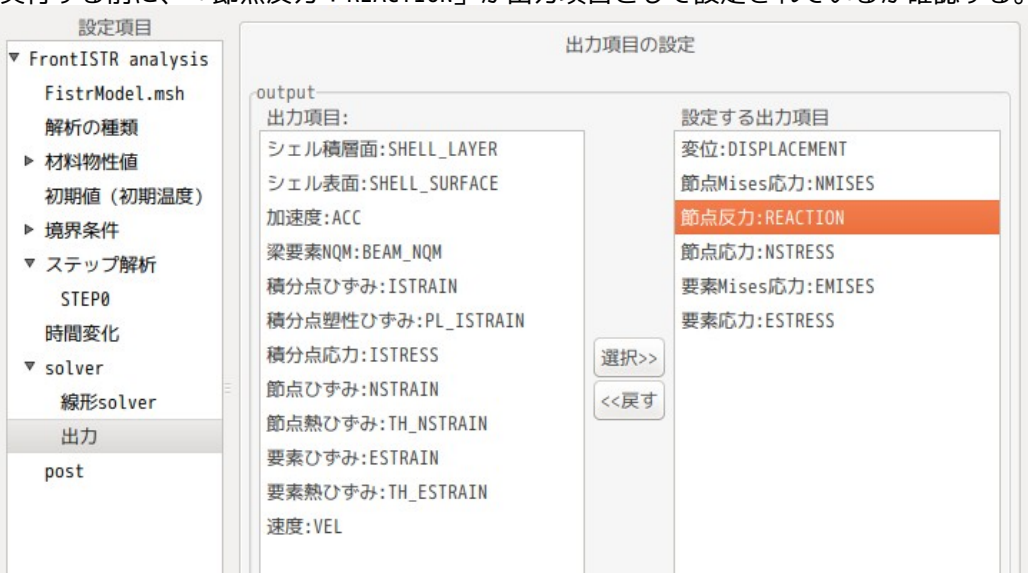
＜STEP2＞

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。

3-16-3.　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-4.6 以前のバージョンは、線形弾性静解析では、反力が正しく計算されない為、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。

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,
300.00000000005195]
```

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。

3-16-4.　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 回行うと、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。

3-16-4-1.　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ので、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
```

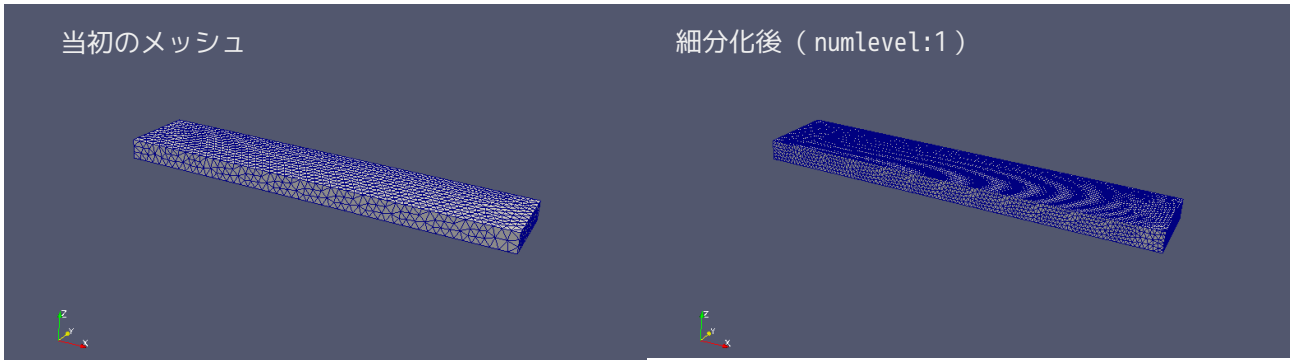
上記例では、メッシュファイル「FistrModel.msh」を numlevel「1」で細分化する。
numlevel:1 は、中間節点を 1 回作成して細分化する。numlevel:2 の場合は、2 回中間節点を追加するので、要素数は 64 倍に増える。
このスクリプトは、四面体、六面体、五面体の 1 次 2 次要素に対応している。シェルやビーム要素には、対応していない。

3-16-4-2.　メッシュ細分化の例

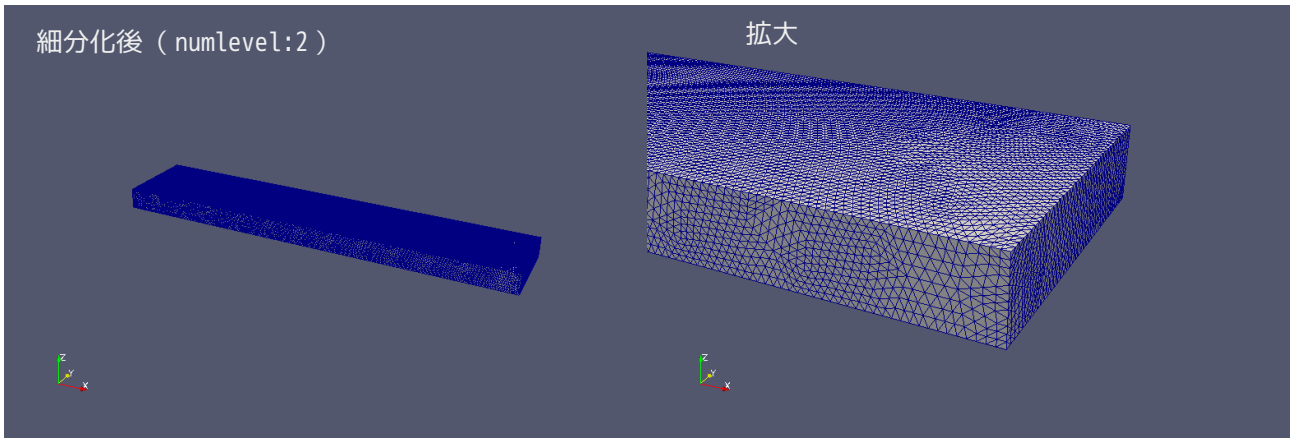
メッシュファイル「plate.unv」を使って、メッシュ細分化を行ってみる。
EasyISTR 上で、plate.unv ファイルをメッシュ変換して、FrontISTR 用のメッシュファイル「FistrModel.msh」を作成する。
この後、EasyISTR 上の「端末起動」ボタンをクリックして、EasyISTR の端末を起動し、以下を入力して 1 回細分化する。

```
$ 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。



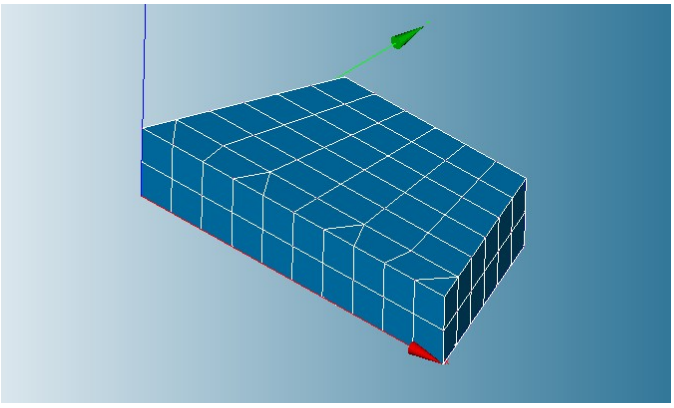
numlevel:2 で細分化を行うと以下になる。



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。

3-16-5.　複数の要素タイプが混在するモデルについて

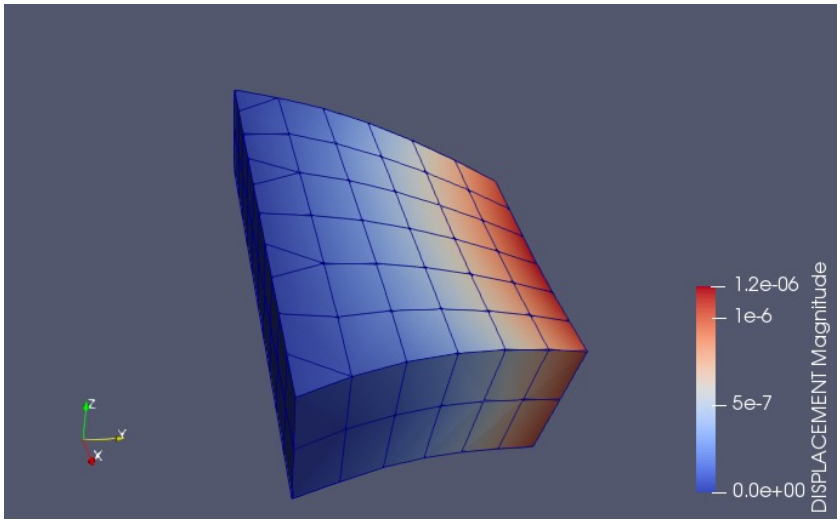
EasyISTR では、1 ヶの要素 group で 1 ヶの要素 type としてモデルを構成する事を基本としているが、モデル形状によっては、以下の様に、六面体要素中に部分的に五面体要素が含まれる場合がある。



この場合は、Salome を使って、六面体要素でメッシュを作成しようとしたが、部分的に五面体要素が出来上がった状態になる。
従来は、この状態の unv ファイルをメッシュ変換できなかったが、現在は、unv2fistr.py を使って、fistr 形式に変換できる。

現状、FrontISTR（EasyISTR）では、solid 要素として、四面体、五面体（prism）、六面体しか扱えないので、これらが混在しているモデルは、問題なくメッシュ変換でき、解析できる。しかし、ピラミッドの五面体は扱えないので、四面体と六面体の混在は扱えない。（要素 group を分けて、四面体のみの要素 group、六面体のみの要素 group とすれば、扱える。）

以下は、前記したモデルを使って、実際に計算した結果になる。


1.2e-06
1e-6
5e-7
0.0e+00
DISPLACEMENT Magnitude
Z
Y
X

4.　応用事例

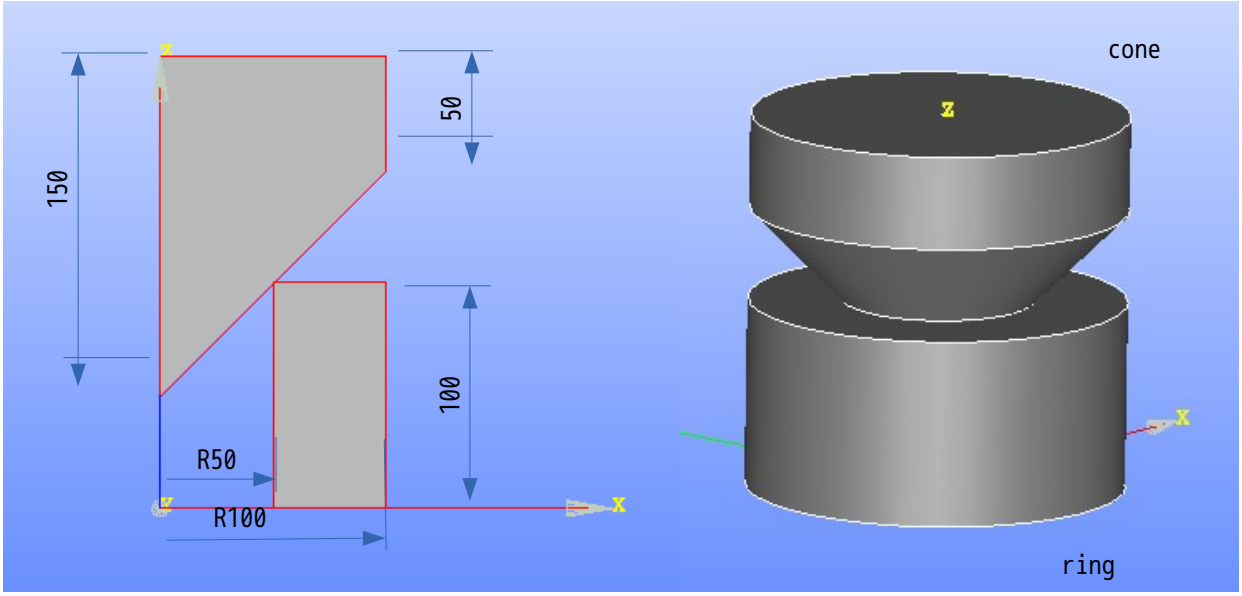
前項で弾性静解析の事例を取り扱った。ここでは、それ以外の事例について、その操作方法を説明する。
この応用事例で使用しているメッシュファイルは、<TreeFoam のインストール Dir>/unvFiles フォルダ内に保存されているので、ここから入手できる。

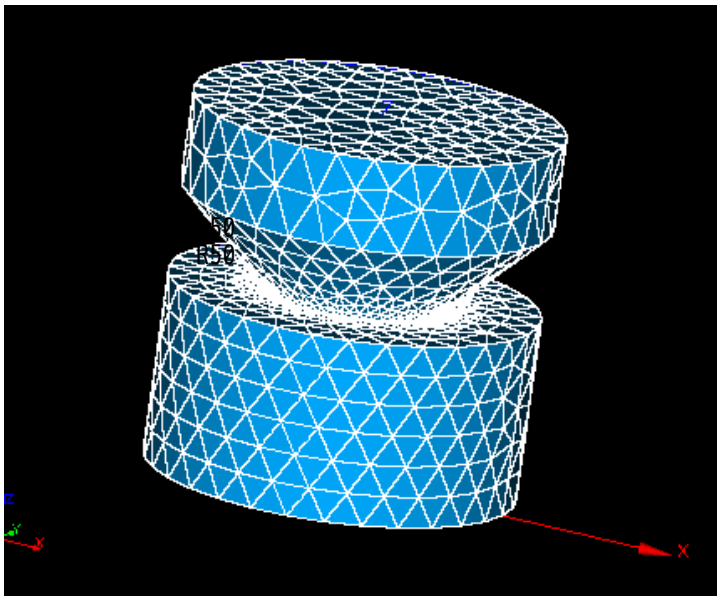
4-1.　接触解析

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

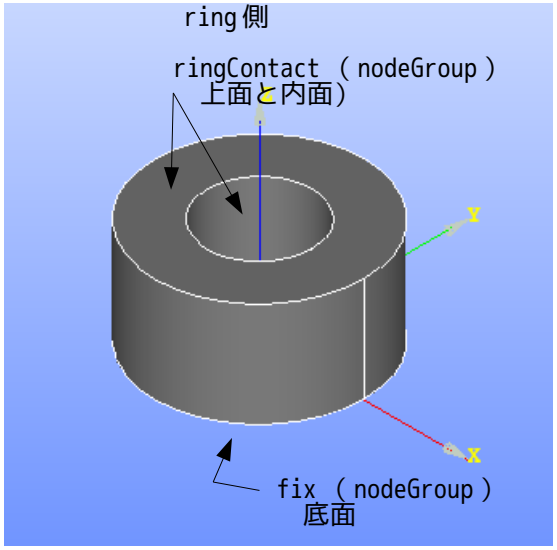
モデルは、以下の形状を考える。salome で以下の形状でメッシュを作成している。

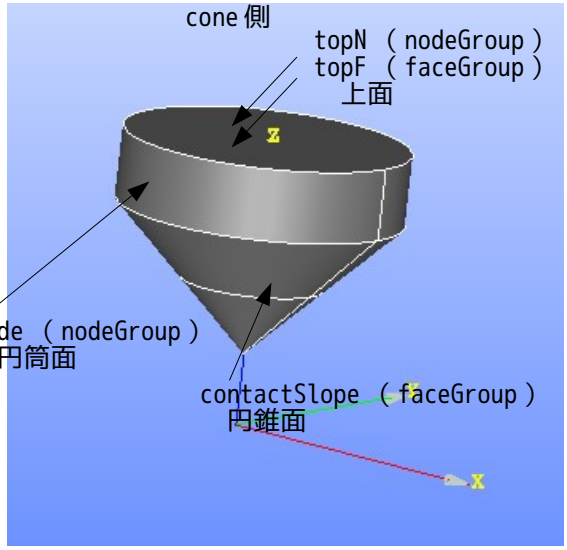




メッシュサイズ
　全体：　netgen-1D-2D-3D
　　20mm
　接触部：local length
　　5mm

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 上面に荷重をかける荷重拘束の 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 が下記の様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。

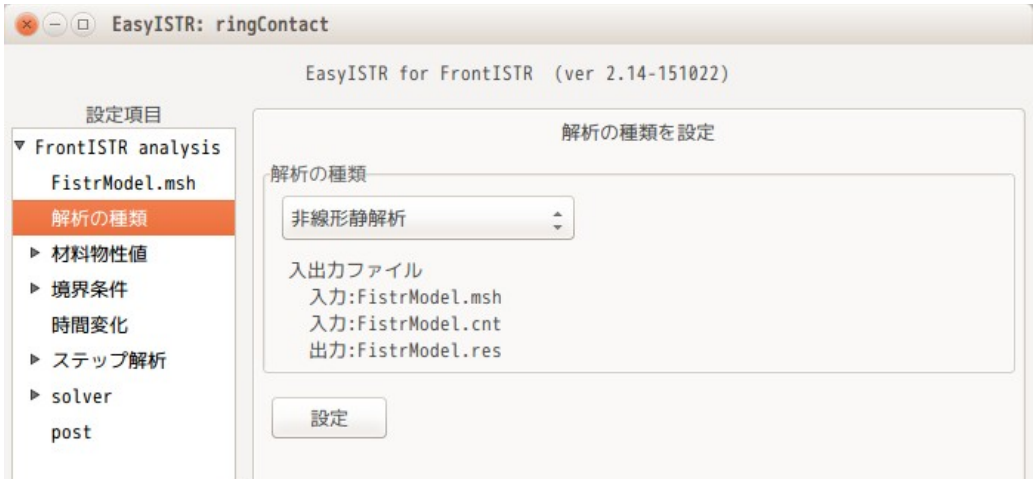
```
EGRP ( elementGroup )
 1. ring
 2. cone

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。

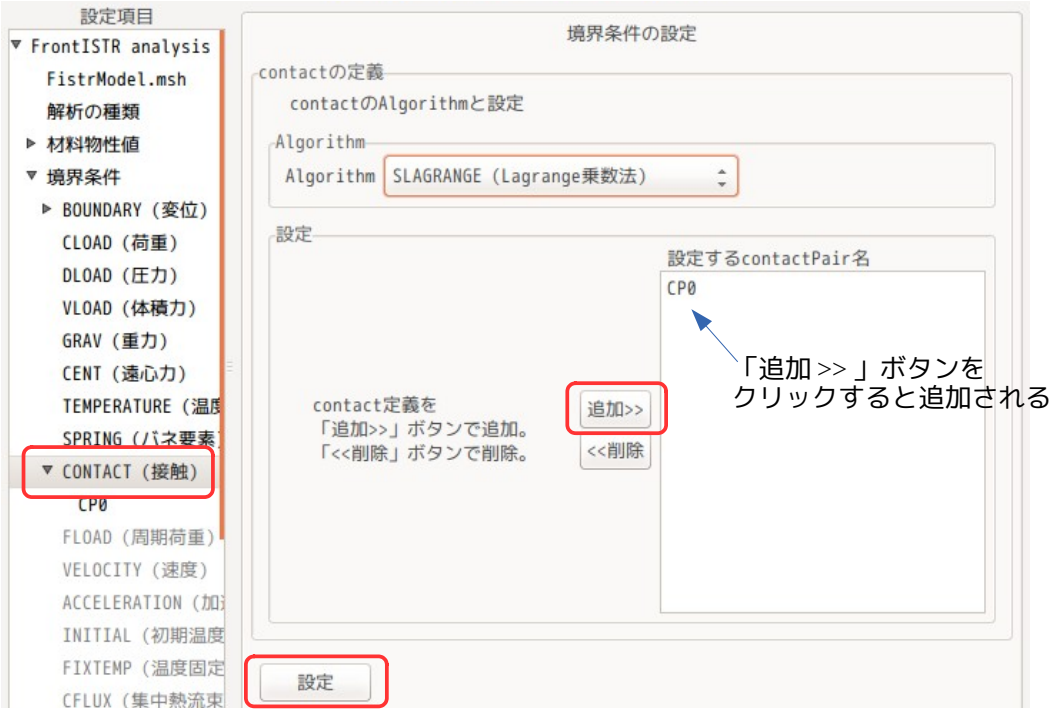
4-1-2-5.　接触の設定

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 Laglange 法）」が選択できるが、ここでは、「Lagrange 乗数法」を選択した。

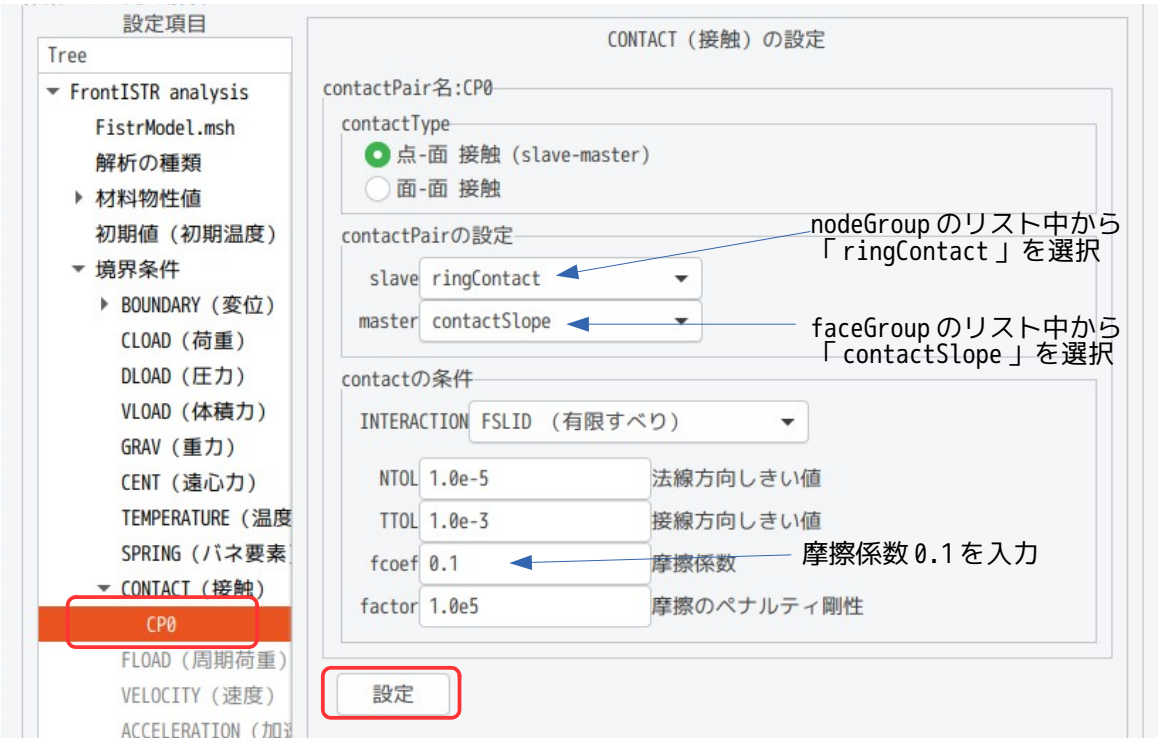
この後、「追加>>」ボタンをクリックし、「設定」ボタンをクリックして、「CP0」を Tree 上に追加する。（下図参照。）



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。

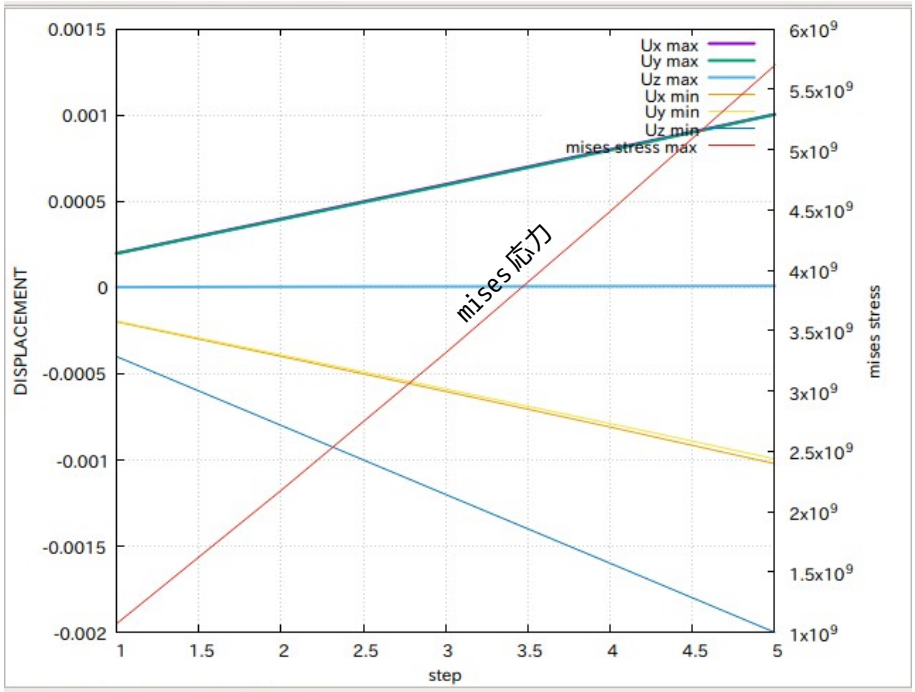
上図に対して、下図の様に修正している。修正後「設定」ボタンをクリックして、内容を確定する。
ステップ解析する境界条件は、ここでは、接触解析に関連する条件を選択する。
SubStep を「5」に設定している為、5 分割して計算していく事になる。

4-1-2-7.　計算開始、結果の確認

計算開始させる前に以下の様に線形 solver を「MUMPS」に変更する。（FrontISTR-4 では、「CG」でも収束していたが、FrontISTR-5 では、収束し難いので「MUMPS」に変更。）

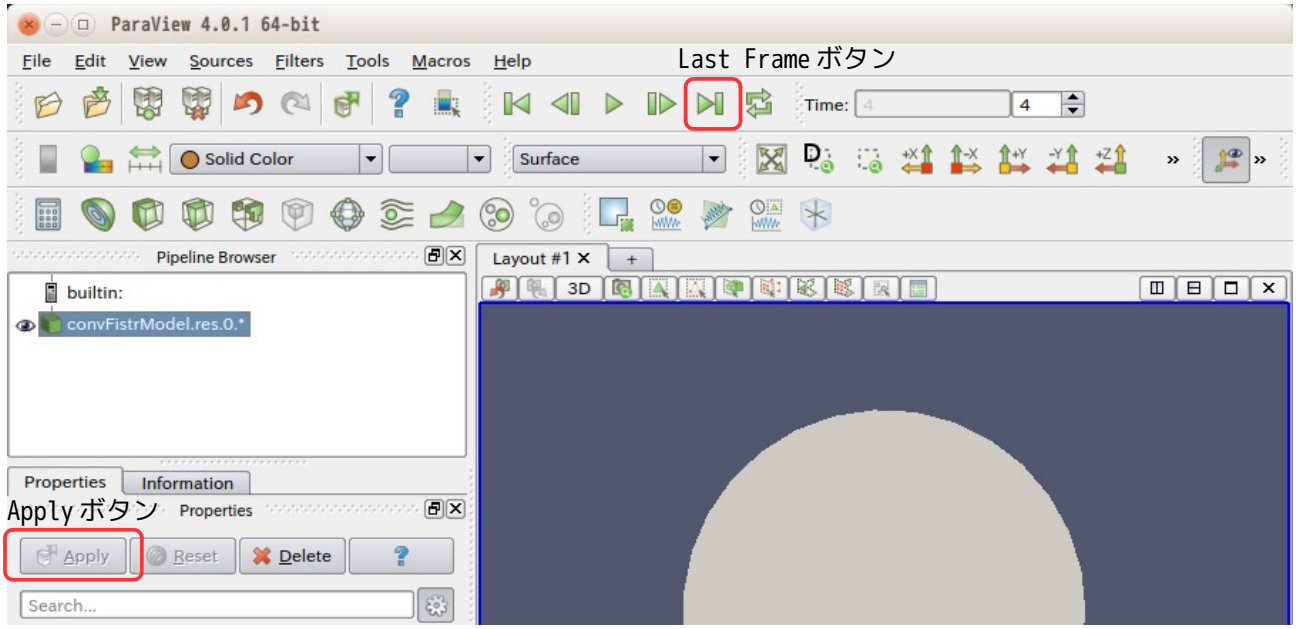
計算開始は、設定項目 Tree 上の「solver」を選択し、「FrontISTR 実行」ボタンをクリックして、計算開始させる。

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 5.7e9Pa 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。

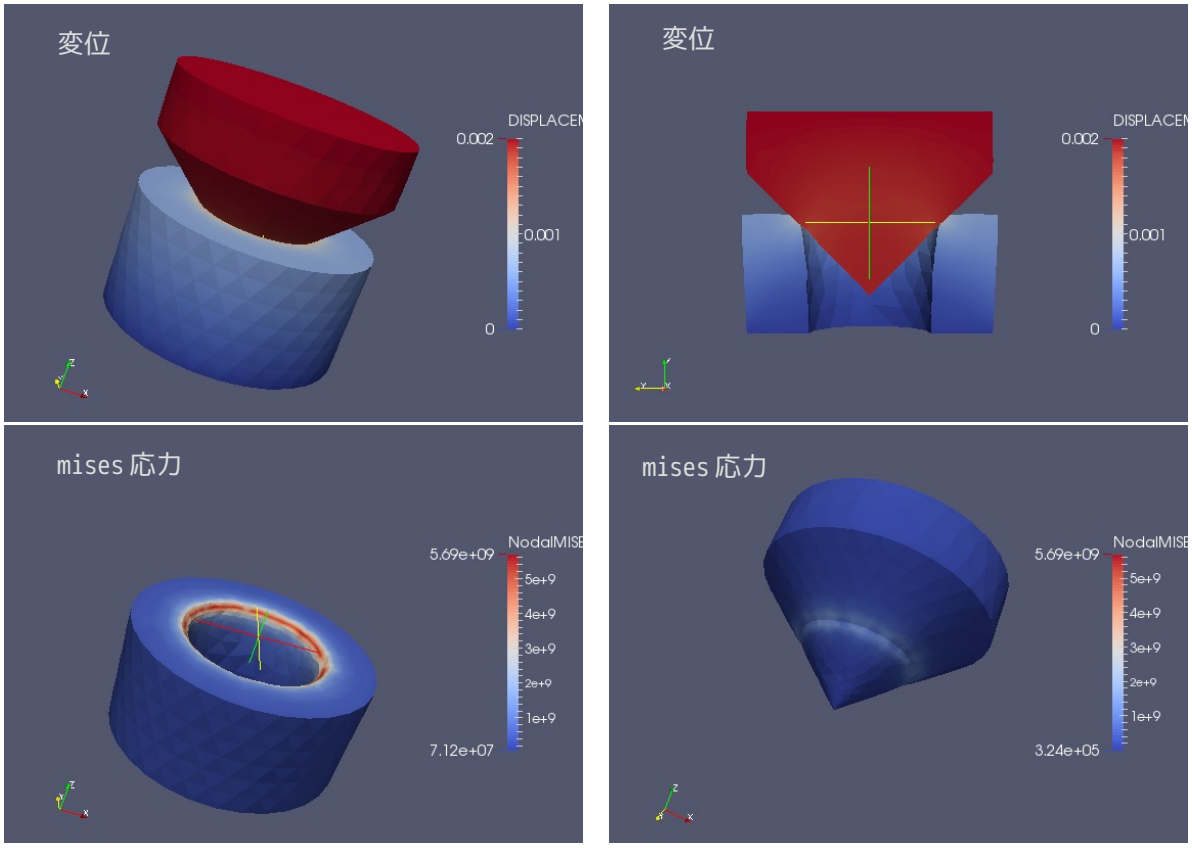


計算終了後、設定項目 Tree 内の「post」を選択し、「eGrp 追加」ボタンをクリックして、結果ファイルに要素 group（cone と ring）を追加しておく。その後、「ParaView 起動」ボタンをクリックして、結果を確認する。

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。



結果 file に要素 group（cone, ring）を追加することによって、上図の様に ring のみ、cone のみの表示をさせる事ができる。
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、弱いバネを追加して、剛体移動を防ぐ方法がある。

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

ring の方は、ring 底面（fix）を固定しているので、剛体移動は発生しないが、cone は、荷重のみの設定にすると、剛体移動が発生してしまう。
この様な場合、cone の上面（topN）の XY 方向を拘束し、cone の外側円筒面（side）に弱いバネを追加する事で剛体移動を防ぐ事ができる。

4-1-3-1.　境界条件の設定

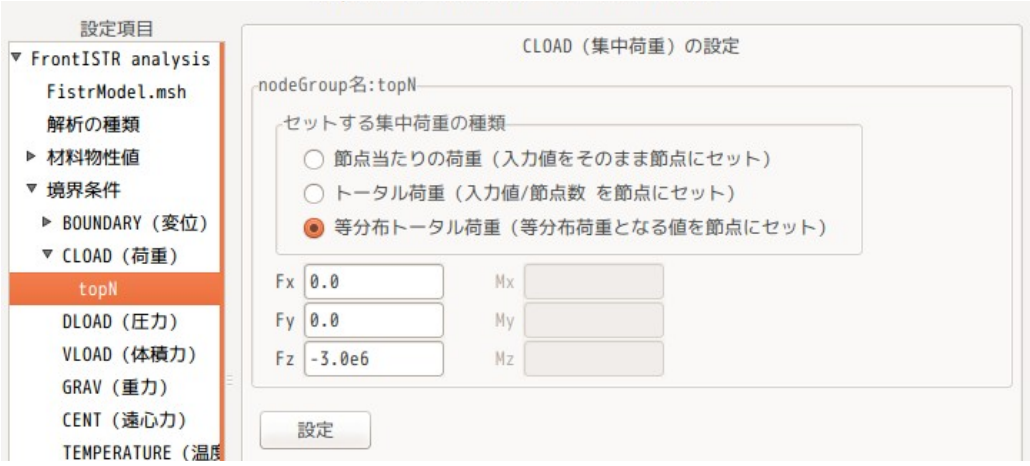
前記した様に、拘束すれば良い為、以下の様に拘束する。

nodeGroup 名	部位	拘束内容
fix	ring 底面	XYZ 方向拘束（固定）
side	cone 外側円筒面	弱いバネ追加
topN	cone 上面	等分布荷重を設定
topN	↑	XY 方向を拘束

ring 底面（fix）に変位拘束（固定）を設定する

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

荷重は、topN に合計 -3.0e6 N の等分布荷重を掛ける事にする。



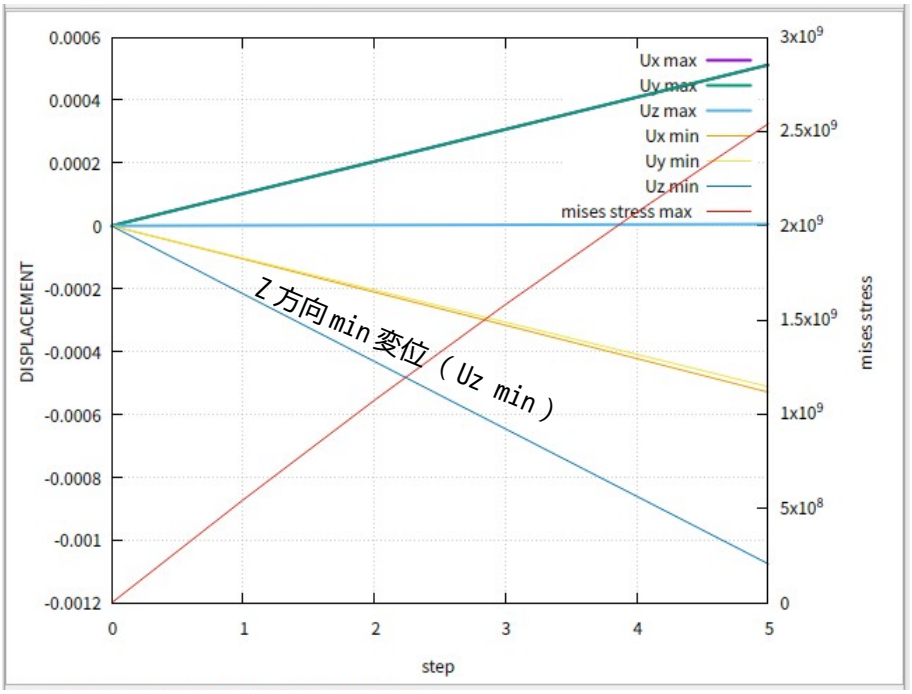
topN の XY 方向を拘束する。

4-1-3-2.　ステップ解析設定

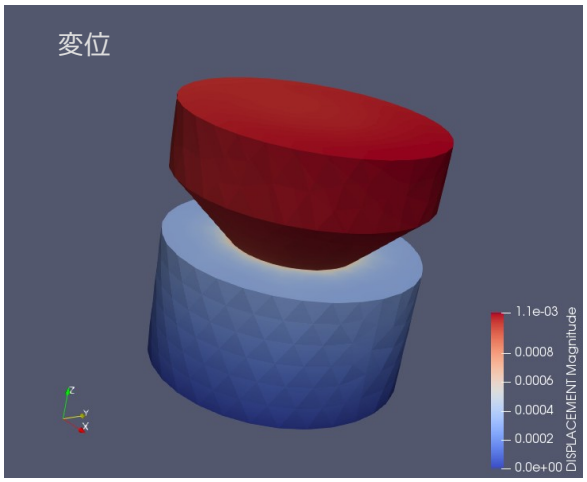
以下のように設定している。

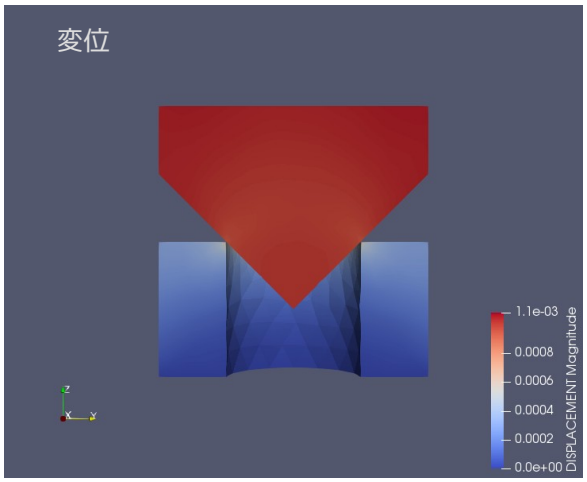
4-1-3-3.　計算開始、結果の確認

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。
もし、途中でエラー停止する場合は、前項の SUBSTEPS を「5」→「20」程度に増やして計算させてみる。



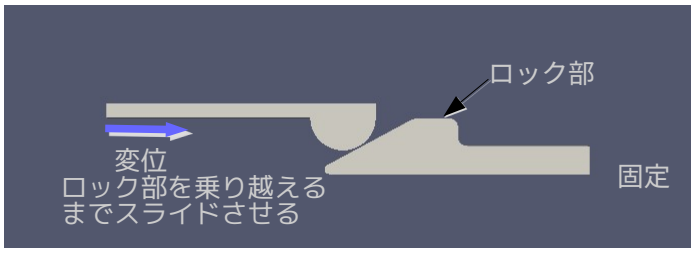
paraView で計算結果を確認した結果が以下になる。（剛体移動が発生せず、うまく計算できている。）





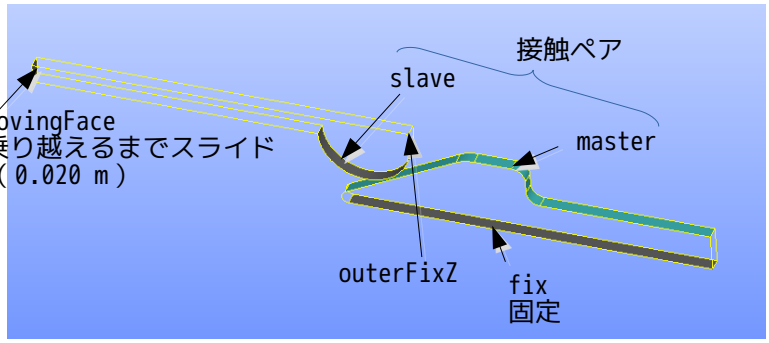
4-1-4.　すべりのあるモデル（スナップフィット）の接触解析：自動時間増分で解析

前項までの接触解析は、殆ど滑りがないモデルだが、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

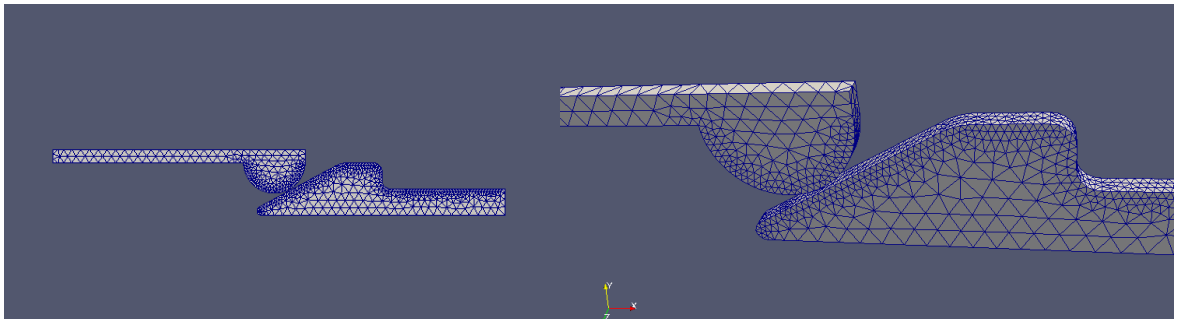
前項の様なモデル形状で、グループ化は、下図のように行っている。モデルは、厚さを持っている３次元モデルになっている。



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

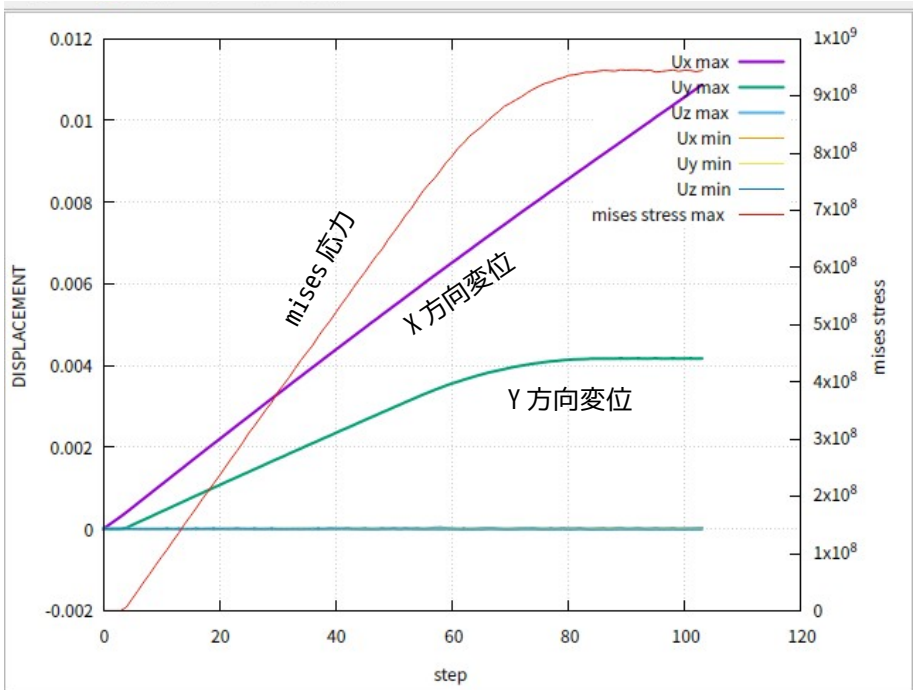
収束し難くなる為、0.1mm 間隔で計算していく。（この条件は、FrontISTR-4 で計算した時の条件）

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、線形 solver を「MUMPS」に設定する。

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、104step で停止した。収束状況は、FSTR.sta ファイルで確認できる。

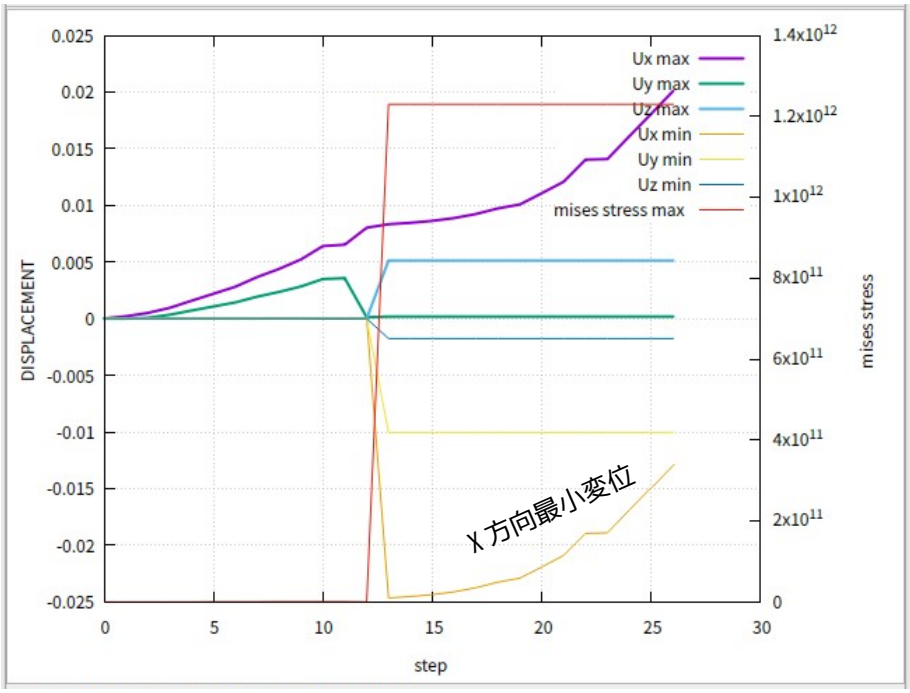


接触状態が位置によって変わってくるので、自動時間増分を使って再計算してみる。
この為、ステップ解析の「自動時間増分を使う」をチェックし、「保存個数」を「10」に設定する。（下図参照）

自動時間増分の時間は、仮想時間の為、時間を勝手に決めても構わない。
ここでは、デフォルトの状態で計算してみる。

初期増分 DTIME：　0.1　　0.2mm 相当
終了時間 ETIME：　10　　20mm に相当
min 増分 minDT：　0.0001　　0.2μm 相当
max 増分 maxDT：　2.0

min 増分は、0.2μm 毎に計算させる事になる。
計算結果は、この時間内で等間隔で 10 ヶ出力させる事になる。（2mm 間隔で結果を出力する。）
これで計算した結果、エラー停止した。



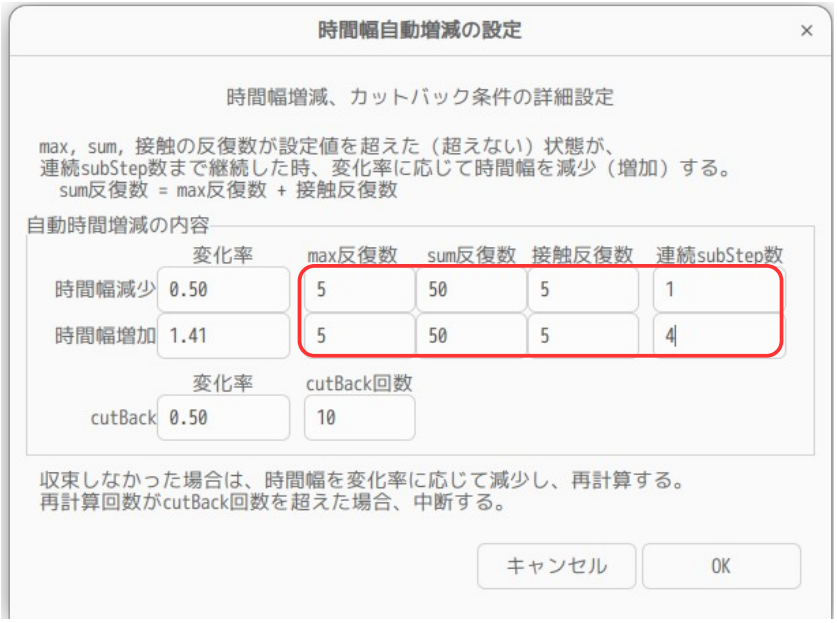
Uxmin（X 方向の最小変位）が「-0.025m」発生している。（本来 X 方向にはマイナス側には変位しない。）
接触状態が途中で変化していくので、時間減少、増加を調整する必要がある。
時間減少、増加の方法は、以下の考え方で行っている。

```
反復回数
 |
 |  収束しない領域:cutBack 域
 |   （増分を 0.5 倍して再計算）
c |-----------------------------  反復回数上限（MAXITER で設定）
 |  増分を減少（0.5 倍）
 |
b |-----------------------------  時間減少しきい値
 |  増分を維持                    (max 反復，sum 反復，接触反復のいずれかが設定値を超える)
 |
a |-----------------------------  時間増加しきい値
 |  増分を増加（1.41 倍）         (max 反復，sum 反復，接触反復の全てが設定値以下)
 |
o+-------------------------------------------------------------------------------
```

安定的に計算させる為、時間増分を減少させる場合は、敏感に反応して減少させる。
時間増分を増加させる場合は、反対に鈍感に反応して、増加させてみる。

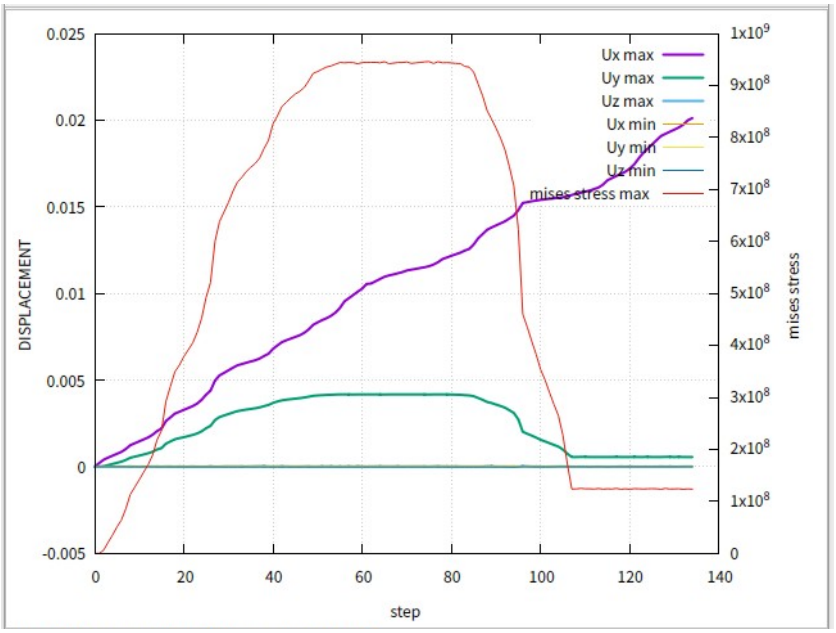
時間増分を減少：反復回数が 10 回を超えた時減少　→　5 回を超えた時減少させる
時間増分を増加：反復回数が 10 回以内で収束した時増加　→　5 回以内の収束が 4 回連続した時増加

この設定を変更する場合は、dialog 中の「編集...」ボタンをクリックして、以下の画面上で内容を修正する。

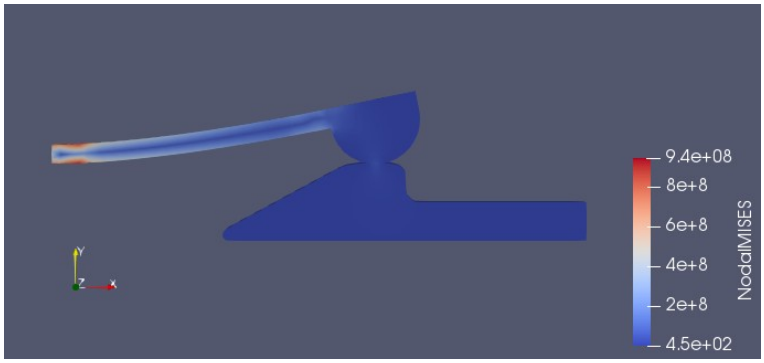


以上の設定で、再計算を試みる。この結果が以下になる。
今度は、うまく収束した。

収束性を向上させる方法として前記した方法の他に、時間増分の変化率を小さく（0.5→0.8、1.41→1.2）して、反復回数を増やす（10→20）方法もある。この為、前記した方法でも収束しなかった場合には、この方法に変更してみる。

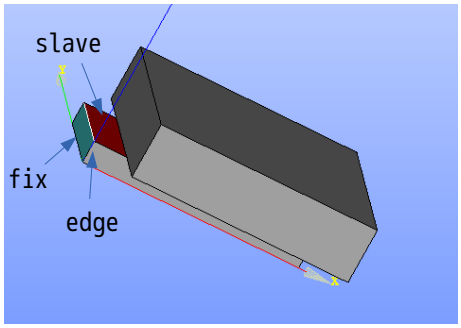


計算結果を確認すると、以下になる。



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場合がある。



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。）

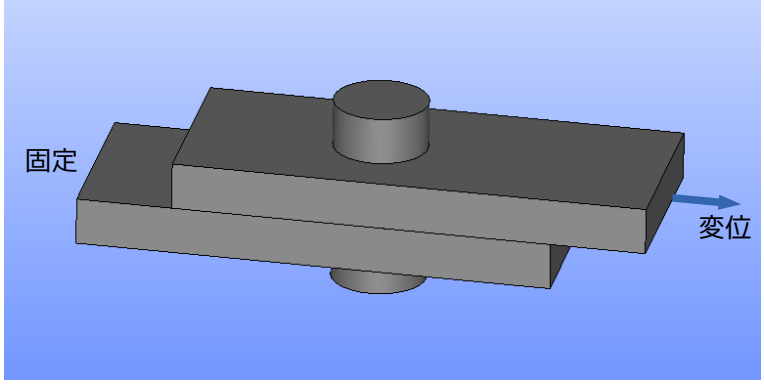
nodeGrpou 間で節点が重複定義されてしまう事は、避けた方が望ましいが、nodeGroup の定義を CAD を使って定義する場合は、通常 edge 部で節点が重複して定義されてしまう。
これを避ける為には、edge 部に小さな面取りを行って、各々の面が edge を共有しない様な工夫をするか、重複する節点を nodeGroup から削除する事が必要になる。

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

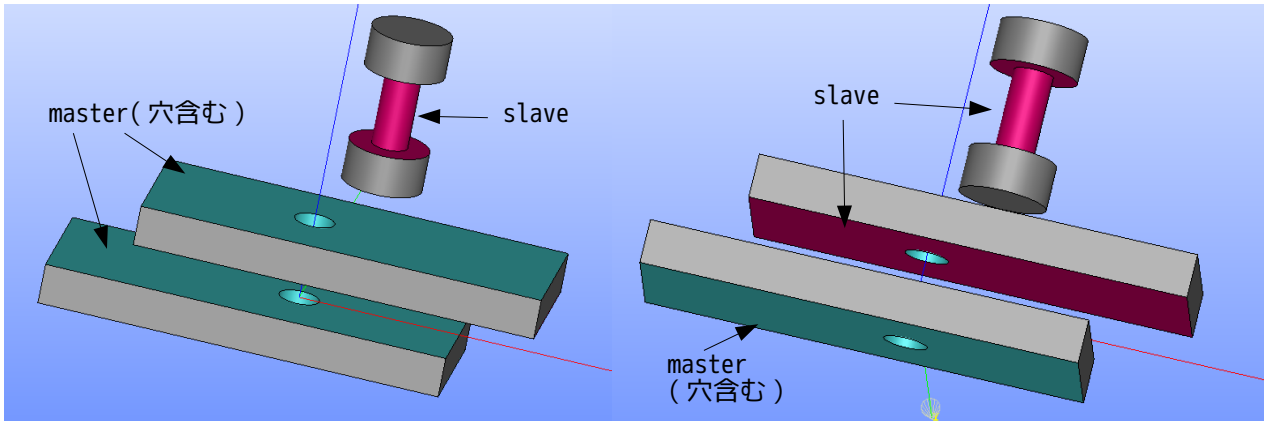
2 枚の板をボルトで締結するモデルを考える。ボルト長は、被締結物の板厚よりも短めに設定し、ボルトに軸力が発生する状態にしている。（初期状態でボルトが板に食い込んでいる状態。）



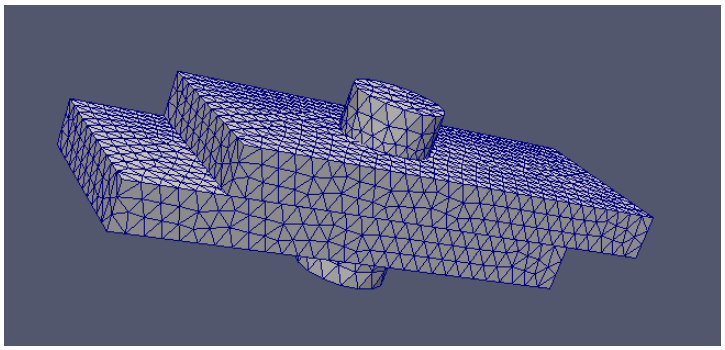
ボルト
　頭径：Φ20mm
　軸：Φ8×19.5mm

上下板
　100×40×t10mm
　穴径：Φ10mm

接触ペアは、以下の様にに定義する。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。このファイルは、EasyISTR のインストール directory 内の「unvFiles」folder 内に保存してある。



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）	接触ペア
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）	接触ペア
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）	接触ペア
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）	接触ペア
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。

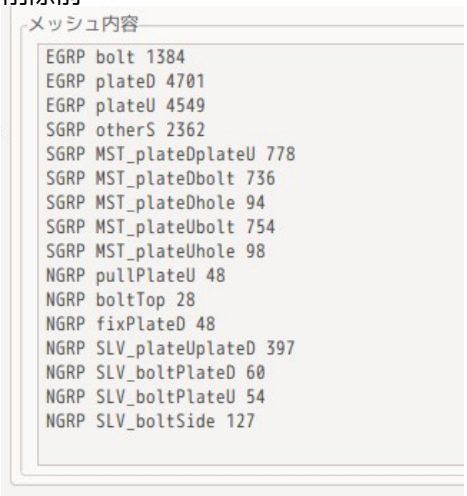
メッシュ変換したメッシュには、nodeGroup 間や nodeGroup と master 面で節点が重複して定義されているので、重複節点を削除する必要がある。この削除のため、同じ画面上の「NGRP の修正」ボタンをクリックして重複節点を削除する。（下図参照）

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）

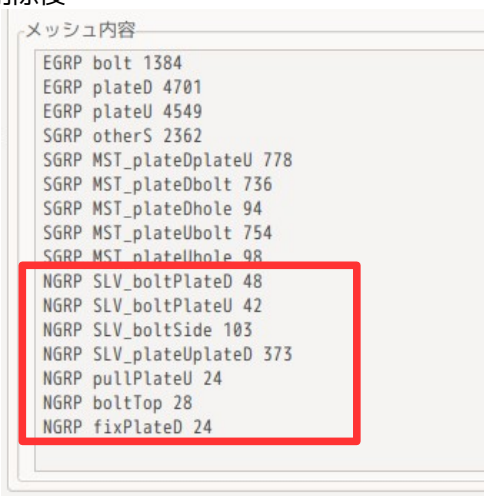
今回の場合、重複している節点（edge 部）を両 group から削除し、且つ、nodeGroup と面 group 間で重複している節点も削除する。この為、この状態で、「OK」ボタンをクリックする。これにより、重複節点全てが削除される。（nodeGroup 間の edge の節点が削除されるので、edge 部に面取りをした事と等価になる。）

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。

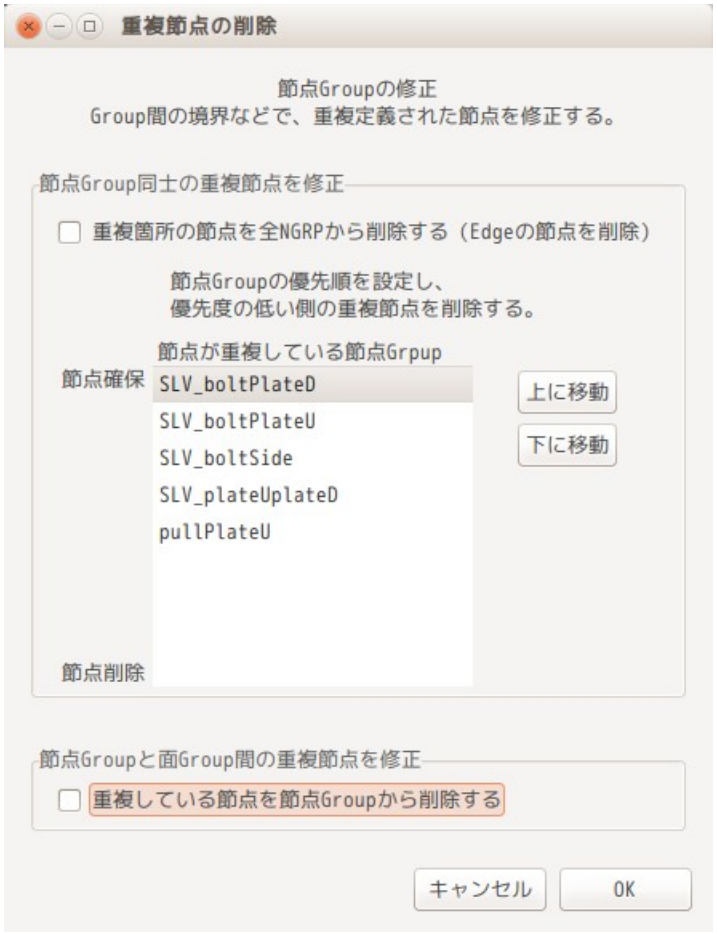
削除前



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）

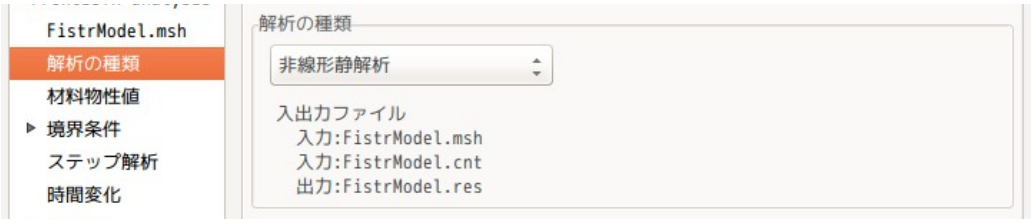


テキストボックス中に互いに重複している nodeGroup 名が表示される。

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、

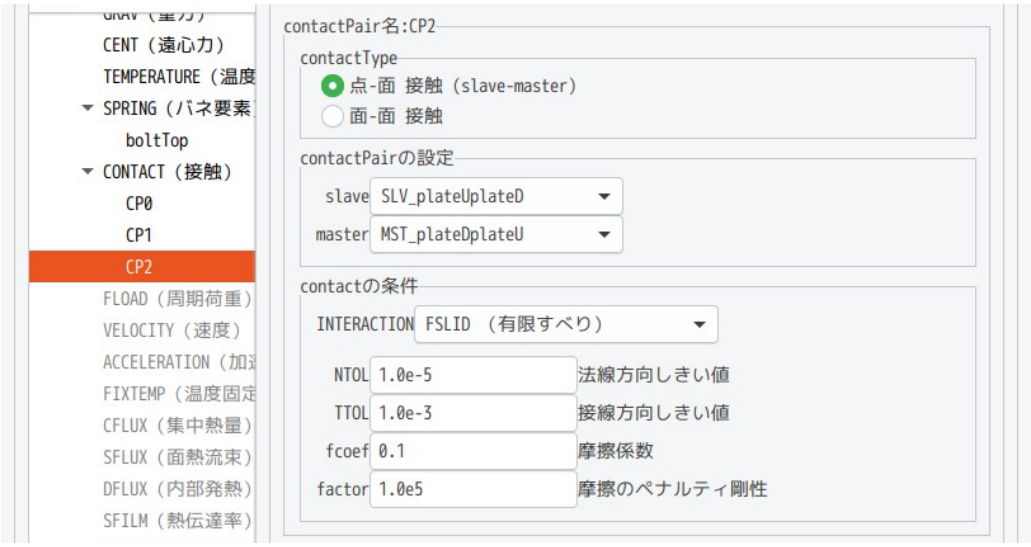
1)　ボルトに軸力が発生する状態（ボルト長が板厚より 0.5mm 短い）の為、まず、この解析を行う。
　　下側板の端面（fixPlateD）、上側板の端面（pullPlateU）を固定する。
　　接触ペアをボルト座面（SLV_boltPlateD、SLV_boltPlateU）と上下板（MST_plateDbolt、MST_plateUbolt）の接触部と、上下板間（SLV_plateUplateD、MST_plateDplateU）に設定。
　　ボルトが拘束されていないので、ボルト頭部（boltTop）に弱いバネを設定する。
　　また、ボルト頭部は、X、Z 方向には変位するが、Y 方向には変位しない様に拘束する。
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、boltTop の Y 方向を拘束し、fixPlateD と pullPlateU を XYZ 拘束する。

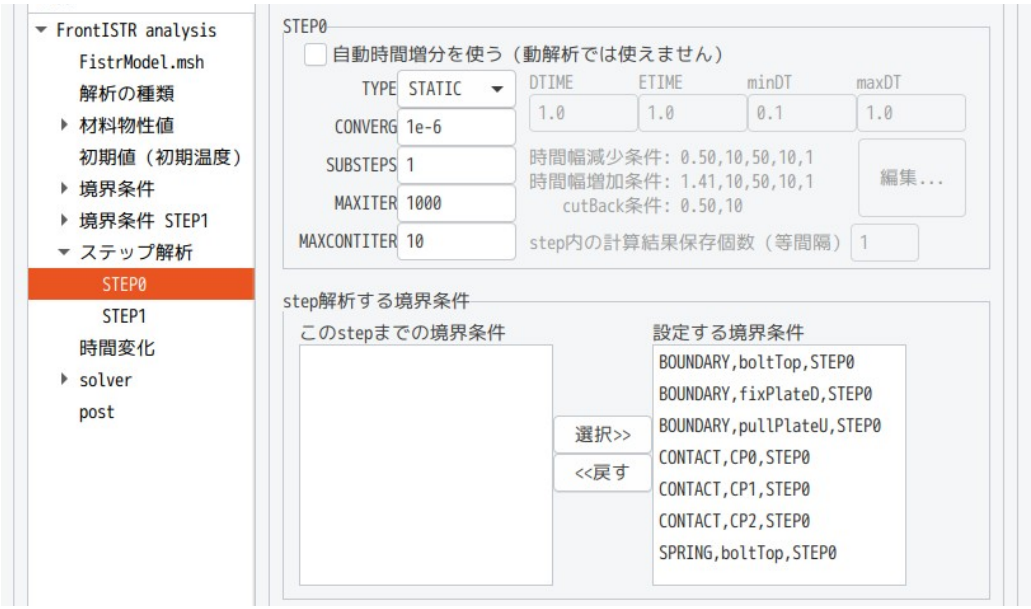
ボルト上部「boltTop」に弱いバネを設定する。XYZ 各方向共「1e5」を設定。

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。

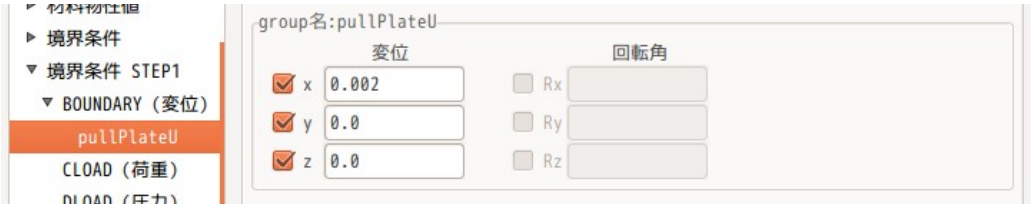


ステップ解析は、以下の様に、STEP0 と STEP1 を設定する。（第 2 ステップまで計算するため）

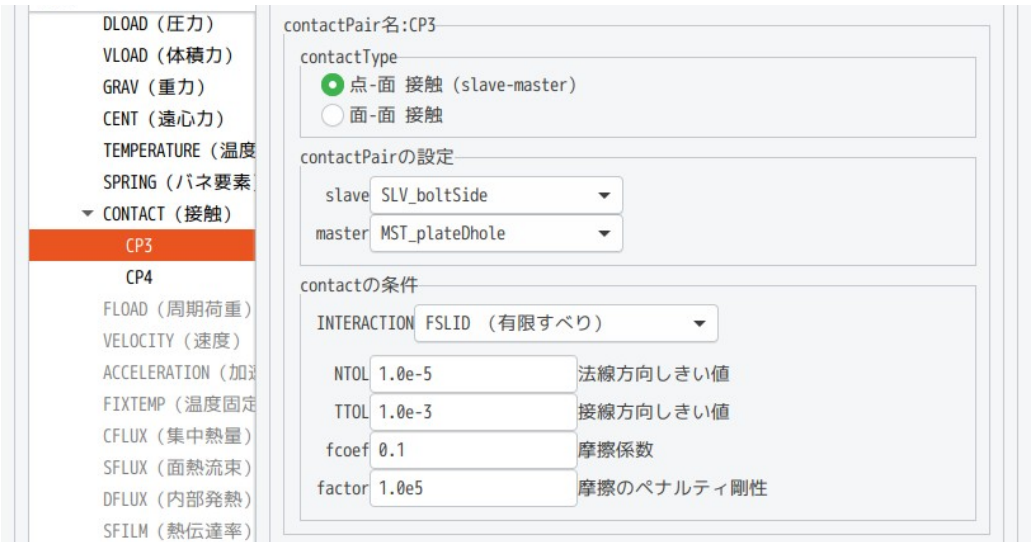
第 1 ステップでは、以下で設定している。尚この時の subStep は、1 回で設定している。
ここでの境界条件は、「fixPlateD」「pullPlateU」を固定（変位「0.0」）の為、subStep を複数回設定しても、増分は「0.0」の為、subStep を複数回設定しても意味がないので、1 回で設定した。

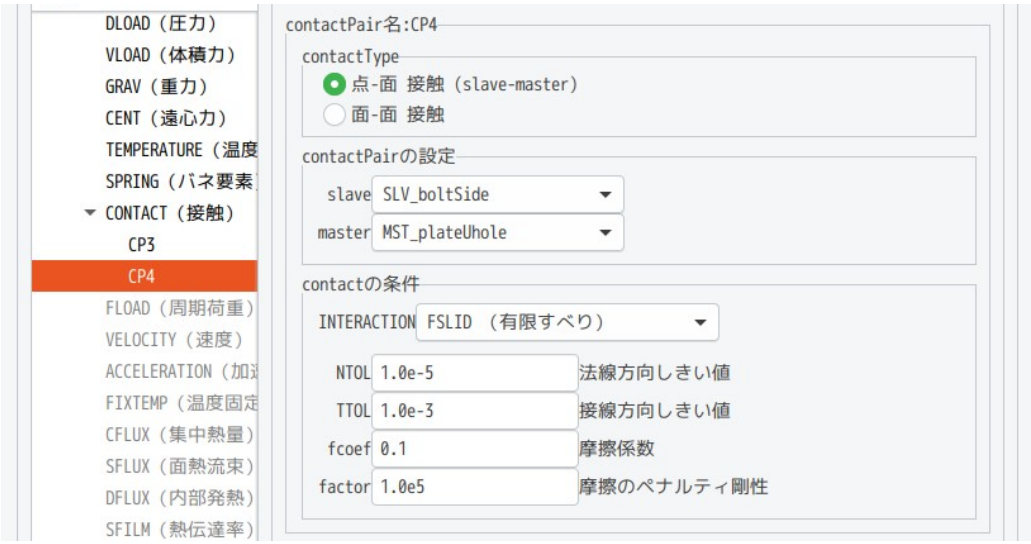


第 2 ステップ（STEP1）の境界条件を設定する。第 1 ステップで設定した境界条件から変更、追加する部分のみ設定する。
boundary は、上側板端面を X 方向に「2mm」引っ張る設定を「境界条件 STEP1」内に追加する。



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。

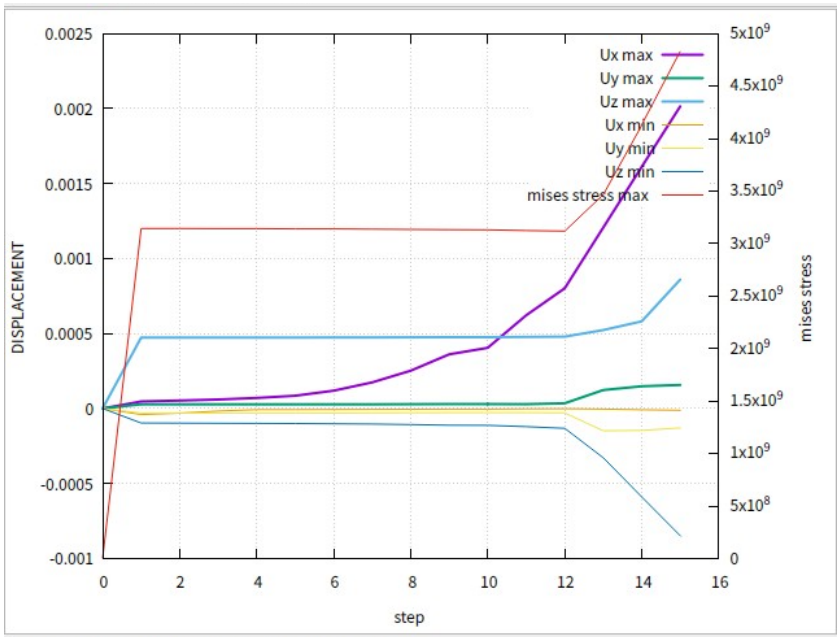




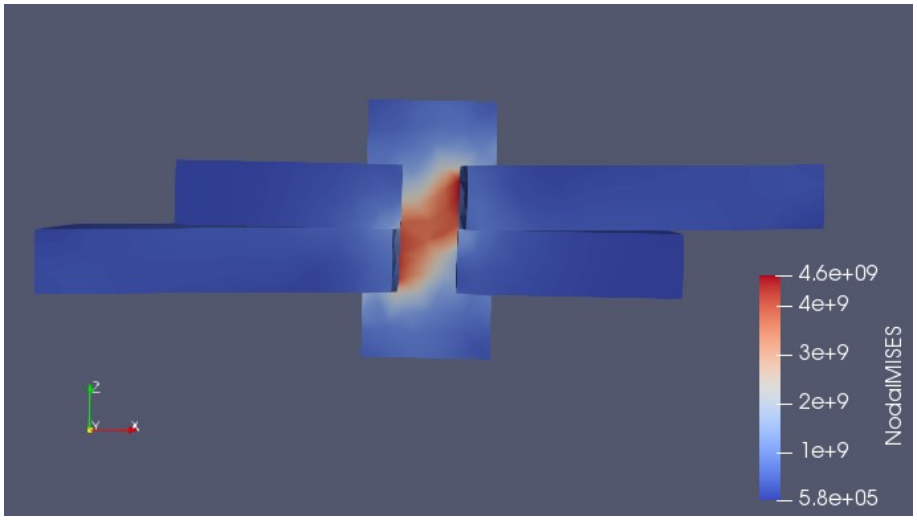
ステップ解析の内容は、以下で設定している。
この step は、自動時間増分として設定している。設定内容はデフォルトのまま。
step 内の計算結果保存は、等間隔で「5」保存する設定。
この step の計算は、上側板（plateU）の境界条件が変わるので、STEP0 で設定した boundary の pullPlateU の固定の条件は外し、STEP1 で追加した接触ペアを追加している。

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。
時間増分が次第に増加し、最終的に収束している。



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る。



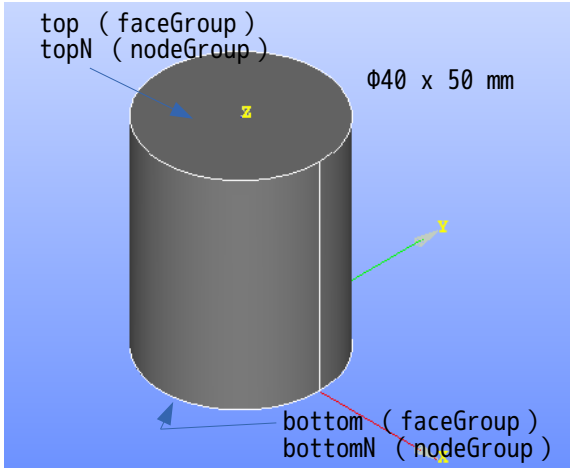
この解析中で、nodeGroup 間の節点が重複節点を削除せずに計算を進めていくと、途中でエラーが発生するが、重複節点を削除する事で、上記の様にうまく計算が進み、結果が確認できる。

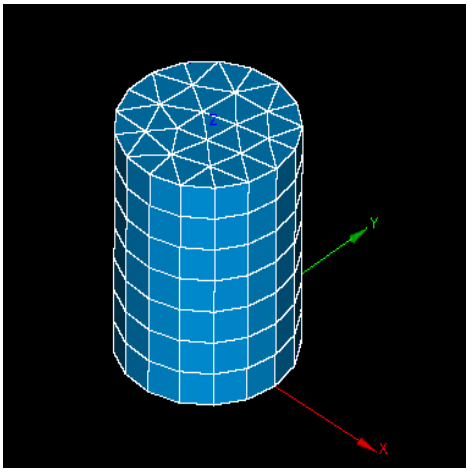
4-2.　弾塑性解析

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
```

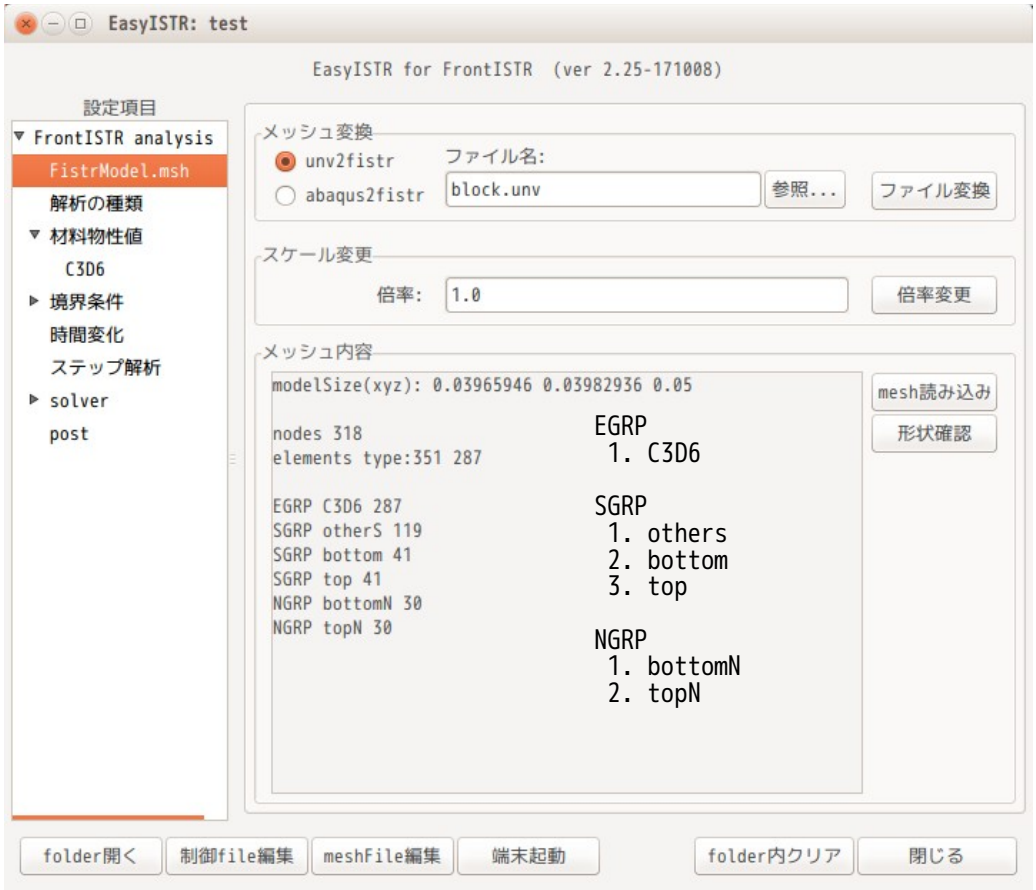
でき上がったメッシュを「block.unv」として、workFolder 内に保存する。

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。

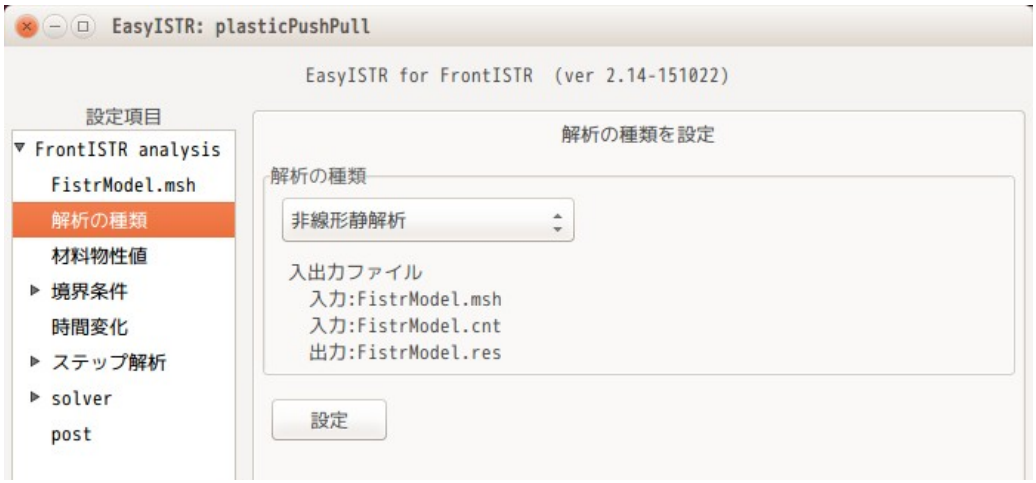
4-2-2-1.　メッシュ変換

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）



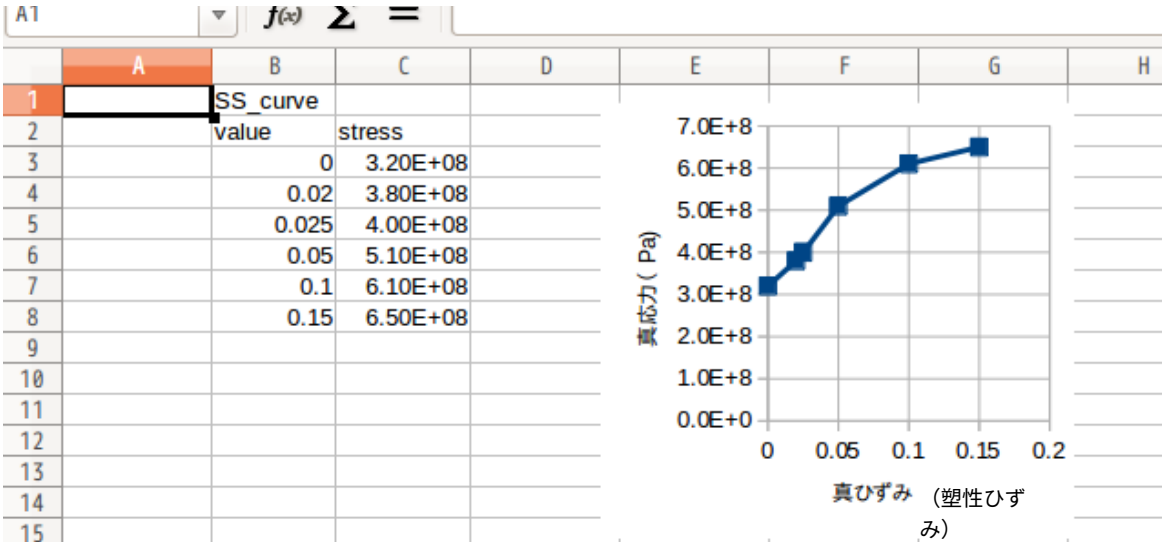
4-2-2-3.　材料物性値の設定

材料は、Steel で材料モデルは、以下で設定。

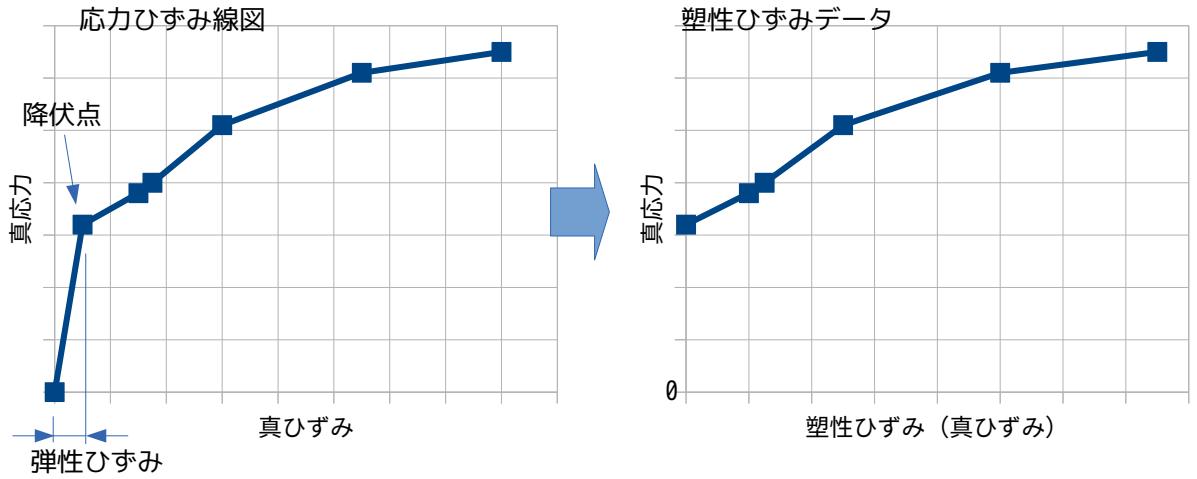
材料モデル：	PLASTIC	塑性
降伏条件/タイプ：	MISES	

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。



今回の塑性材料は、降伏条件:MISES、硬化則:MULTILINEAR で設定したが、硬化則:BILINEAR でも GUI 上で設定できる。設定データは、Steel_PlasticMisesBilinear ファイルに保存される。
他に非線形材料として、HYPERELASTIC や CREEP も GUI 上で設定できる。

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、「1.節点ひずみ:NSTRAIN」を追加した。

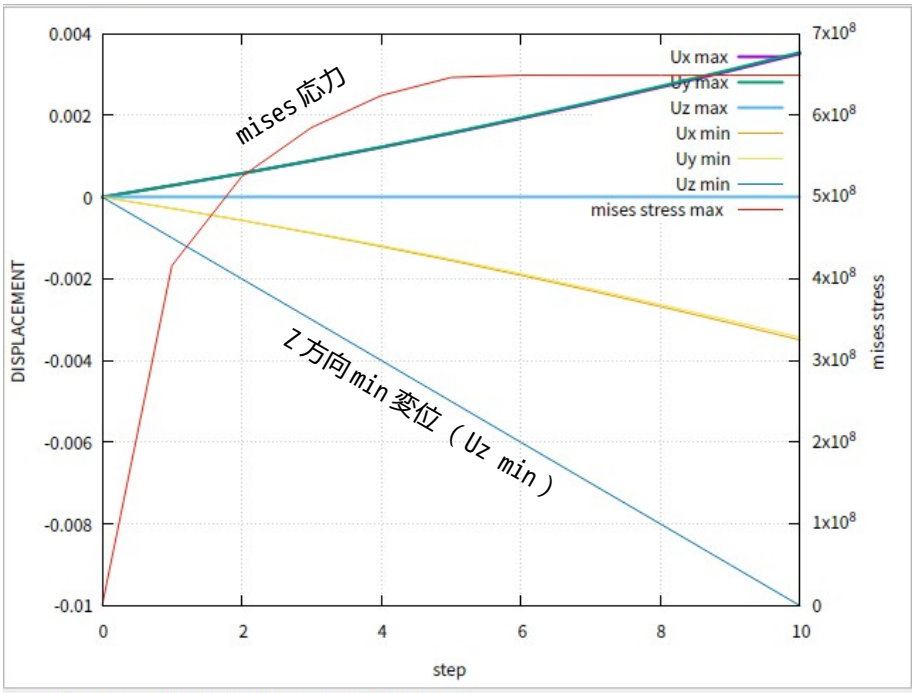
設定が終了したので計算を開始させるが、次項で、計算させた後さらにその結果を使って計算を継続させる（restart）為、その設定を行っておく。
計算を restart できる様にする為には、下図の「途中 step から開始する」のチェック Box にチェックをし、「設定」ボタンをクリック。さらに「途中 step から開始する」のチェック Box のチェックを外し、「設定」ボタンをクリックして、元に戻しておく。

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。

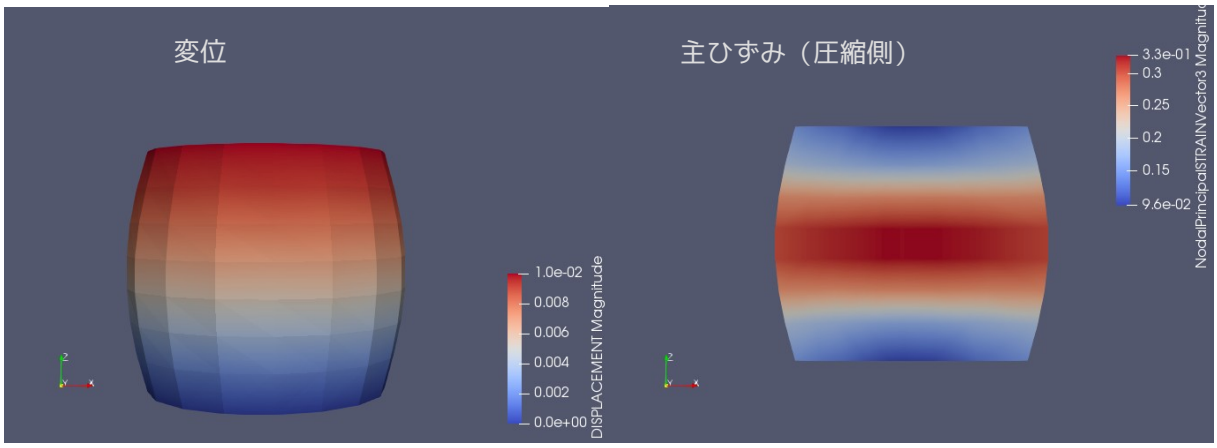
```
----------------- FistrModel.cnt -----------------------
        :
##################
# SOLVER CONTROL #
##################
!RESTART, FREQUENCY=1
!SOLVER,METHOD=CG,PRECOND=1, ITERLOG=NO, TIMELOG=YES
 20000, 2
 1.00000e-06, 1.00000, 0.00000
 0.100000, 0.100000
        :
-------------------------------------------------------
```

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。

plotStepMonitror 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。

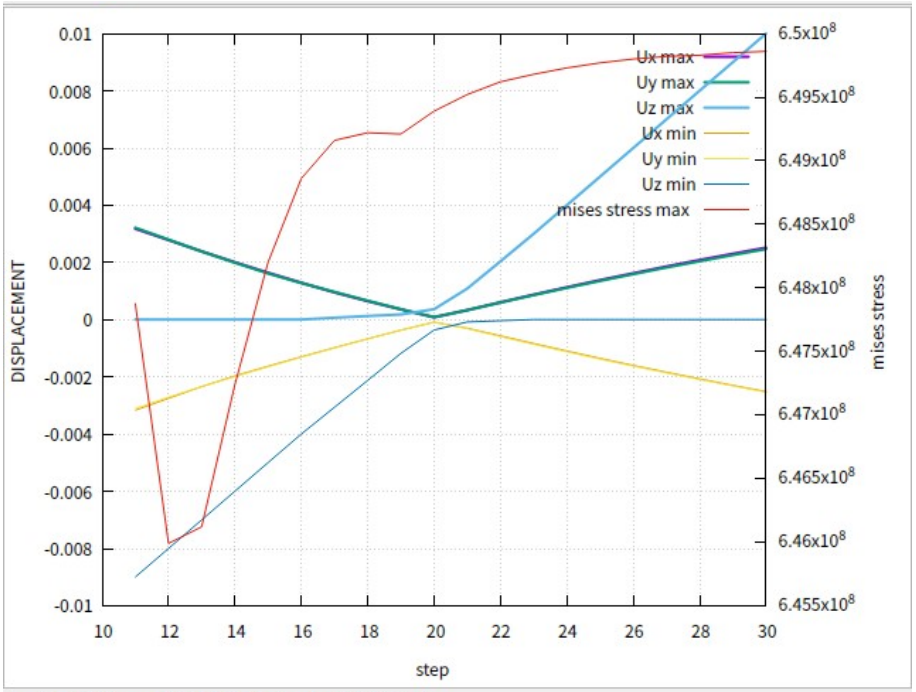
前項で、0.001/1step で 0.010（10step）まで圧縮した。ここから等間隔で、寸法を+0.010 まで引っ張る為には、+20step（計 30step）必要になる。
この為、上面（topN）の境界条件は、0.001/1step で+20step 分変位させる値「0.020」になり、この値を境界条件として設定する。

この為、topN の境界条件は、以下で設定し直す。

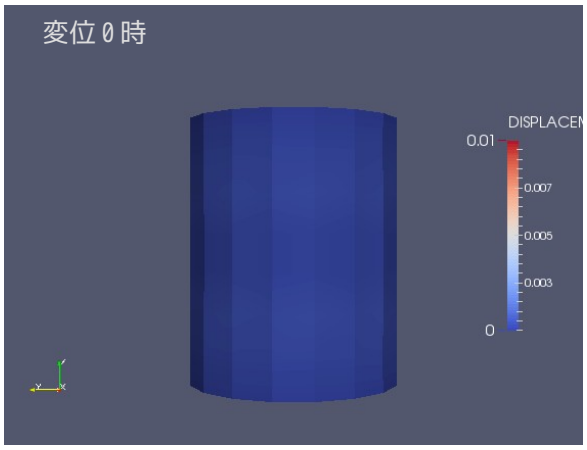
STEP の設定は、SUBSTEP の値を「30」に修正する。下図参照。

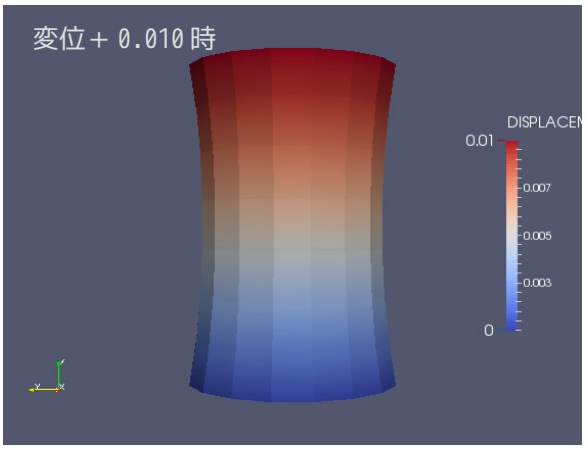
計算は、以下のように、restart の制御の「途中 step から開始する」チェックボックスにチェックを入れ、「FrontISTR 実行」ボタンをクリックして、計算をスタートさせる。

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 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で、途中「0」を通過して、「+0.010」 まで変化する。
下図は、変位「0」の時と「+0.010」の時の変位と塑性変形を表している。変位「0」でも塑性変形が残っているのが確認できる。





4-2-4.　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。

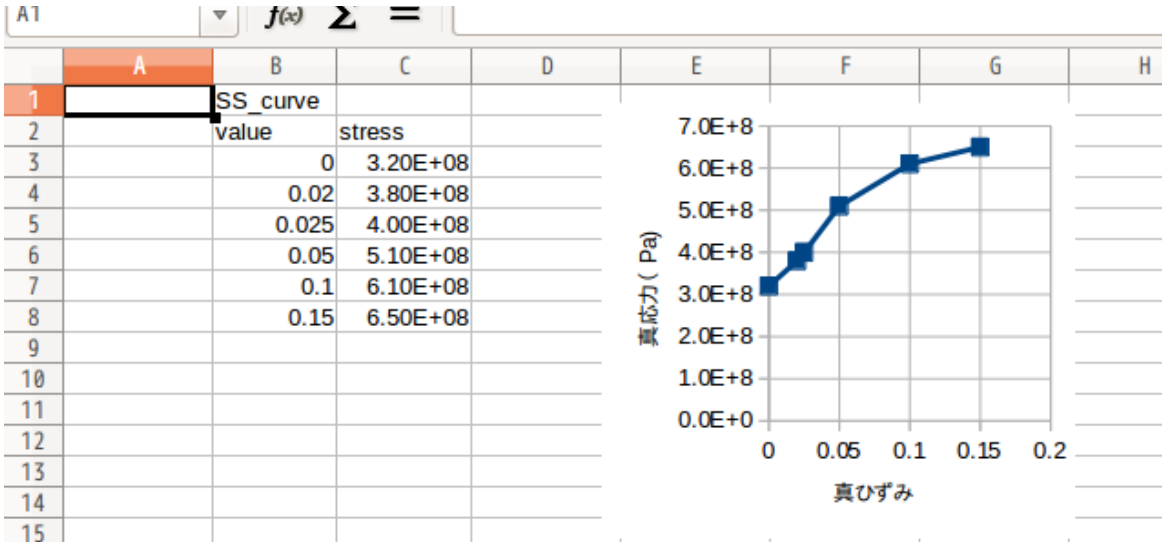
4-2-4-1.　モデルとメッシュ作成

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。

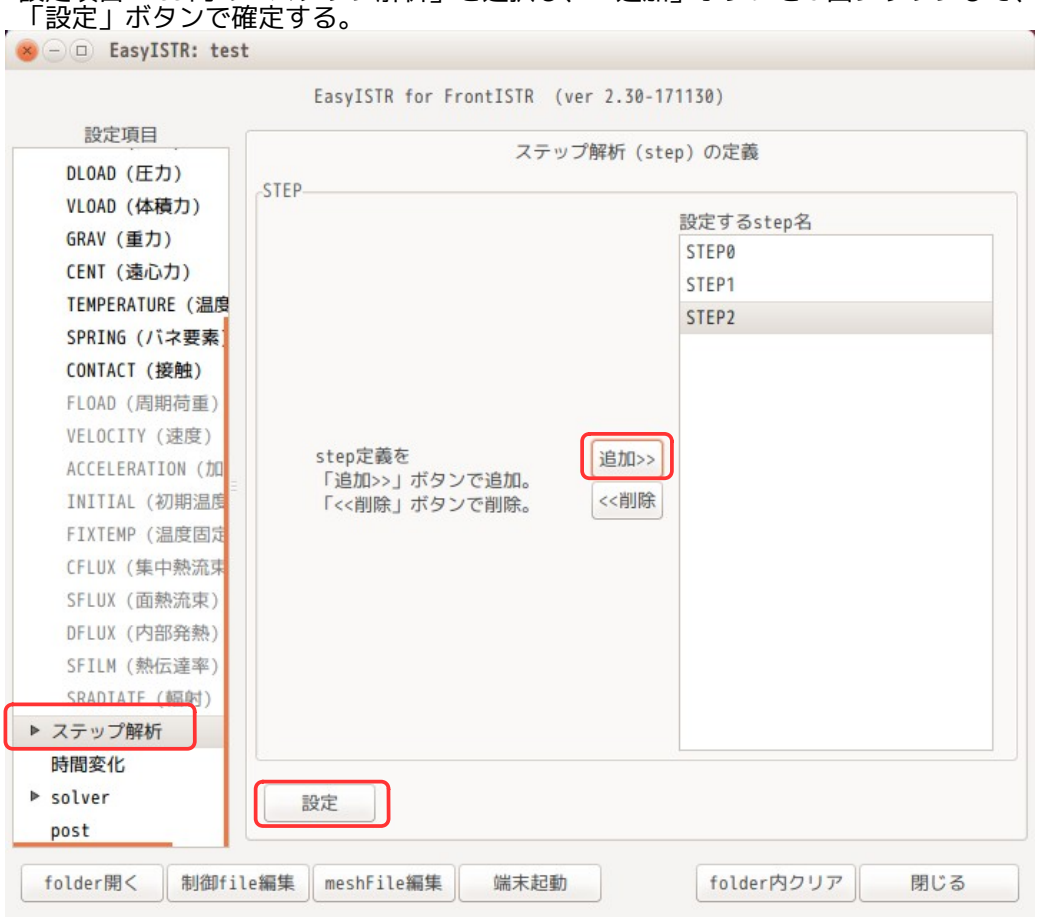


4-2-4-3.　境界条件の設定

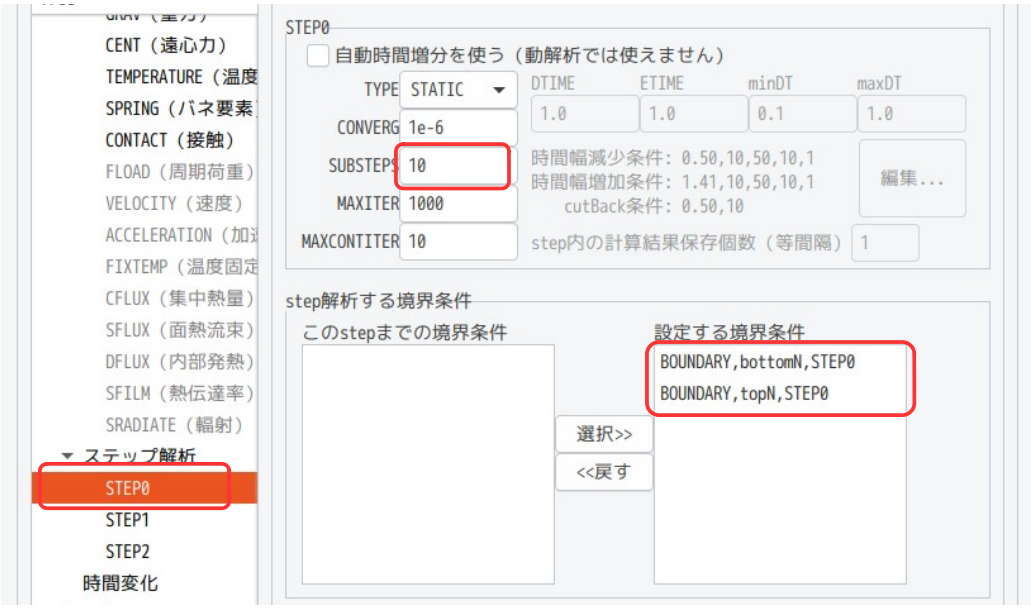
境界条件を設定する。解析は、block 上面を 10mm 下げ、その後、元の長さまで戻し、さらに 10mm 引き上げる解析を行う。
この為、まず、底面を固定し、上面を 10mm 縮める境界条件を設定する。下図参照。

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 ヶ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 ヶ作成し、「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,topnN,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。

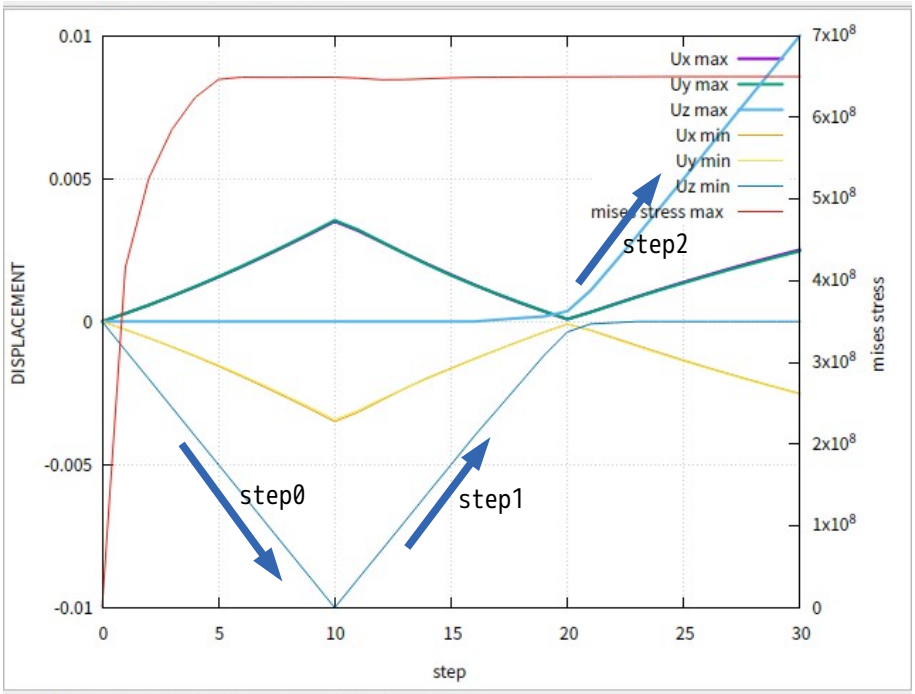
この後、「ステップ解析」を選択し、「SUBSTEPS」数と「設定する境界条件」を設定する。
「設定する境界条件」は、以下を選択する。

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。



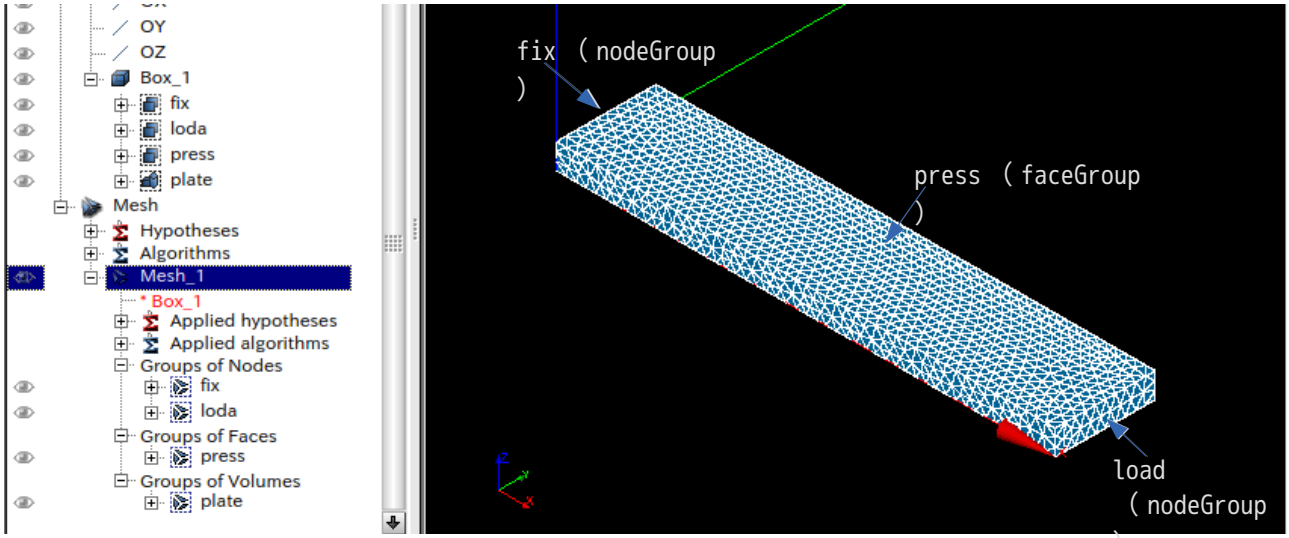
今回の計算では、1回の計算で、上面を、-10mm（10mm縮める）　→ +10mm（元の寸法に戻す）　→ +10mm（さらに10mm伸ばす）に変化させた結果を取得することができる。
変形形状や応力は、前項と同じ結果になるので、省略する。

4-3.　固有値解析

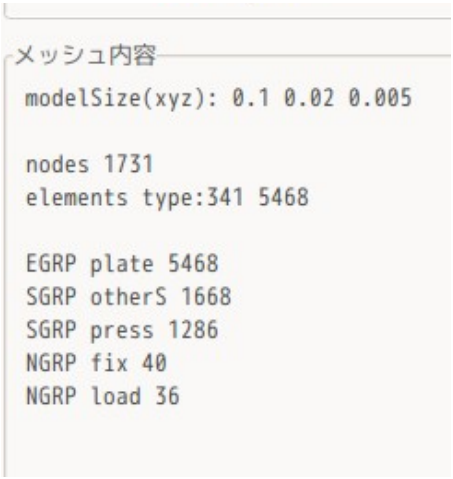
EasyISTR 上で固有値解析を行ってみる。
モデル形状は、線形弾性静解析で使った片持ち梁を使うことにする。
workFolder は、「CAE-fistr/Case/plateEigen」フォルダを新しく作成し、ここで解析する。

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。




```
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

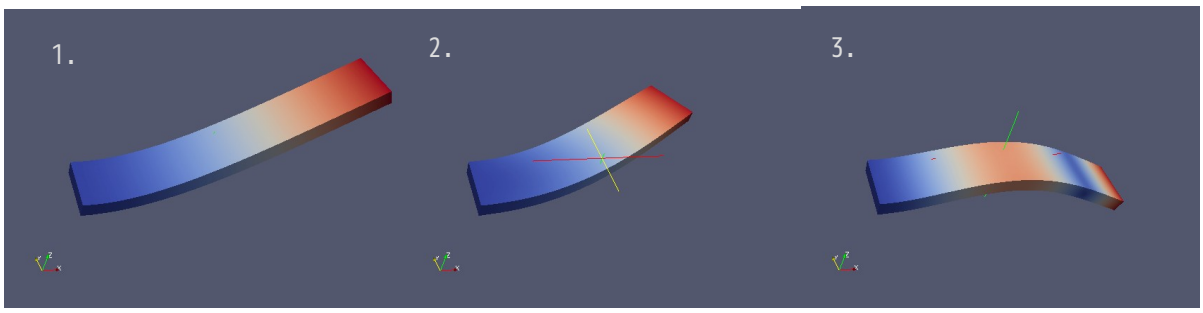
境界条件は、片持ち梁の端面（fix）を固定する。

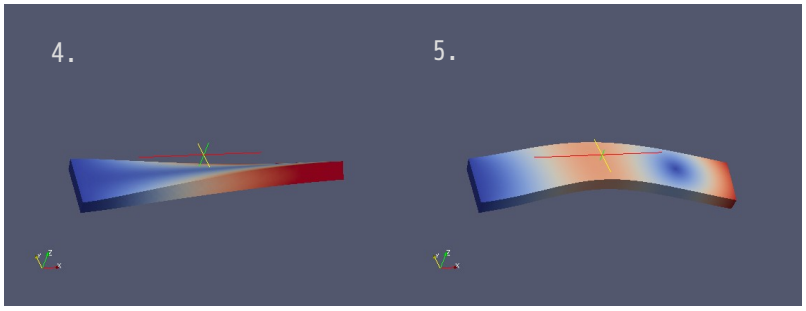
4-3-2-4.　計算開始、結果の確認

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。

```
 1 fstr_setup: OK
 2
 3 ********************************
 4 *RESULT OF EIGEN VALUE ANALYSIS*
 5 ********************************
 6
 7 NUMBER OF ITERATIONS =       25
 8 TOTAL MASS =   7.8600E-02
 9
10                    ANGLE       FREQUENCY   PARTICIPATION FACTOR                EFFECTIVE MASS
11   NO.  EIGENVALUE  FREQUENCY   (HZ)        X           Y           Z           X           Y           Z
12   ---  ----------  ----------  ----------  ----------  ----------  ----------  ----------  ----------  ----------
13     1  1.1344E+07  3.3681E+03  5.3605E+02  5.2223E-06  4.0958E-04  1.5631E+00  5.3543E-13  3.2935E-09  4.7969E-02
14     2  1.0612E+08  1.0301E+04  1.6395E+03 -2.2218E-05  1.5392E+00  7.1659E-04  1.0013E-11  4.8058E-02  1.0416E-08
15     3  4.3064E+08  2.0752E+04  3.3028E+03  6.0470E-05 -1.3211E-03  6.3506E-01  1.3481E-10  6.4346E-08  1.4869E-02
16     4  8.3983E+08  2.8980E+04  4.6123E+03  5.7715E-04 -5.2094E-04  6.3888E-04  4.6232E-09  3.7665E-09  5.6650E-09
17     5  3.0813E+09  5.5509E+04  8.8345E+03 -1.0166E-04  6.3202E-01  1.0437E-02  4.1309E-10  1.5966E-02  4.3544E-06
18
19     Iter.#   Eigenvalue    Abs. Residual
20     1  1.1344E+07  1.0987E-01
21     2  1.0612E+08  4.0755E-01
22     3  4.3064E+08  2.1211E+00
23     4  8.3983E+08  1.2953E+00
24     5  3.0813E+09  2.4910E+01
```

変形モードの関しては、「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。


1.
2.
3.


4.
5.

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/plateEigenResponse」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

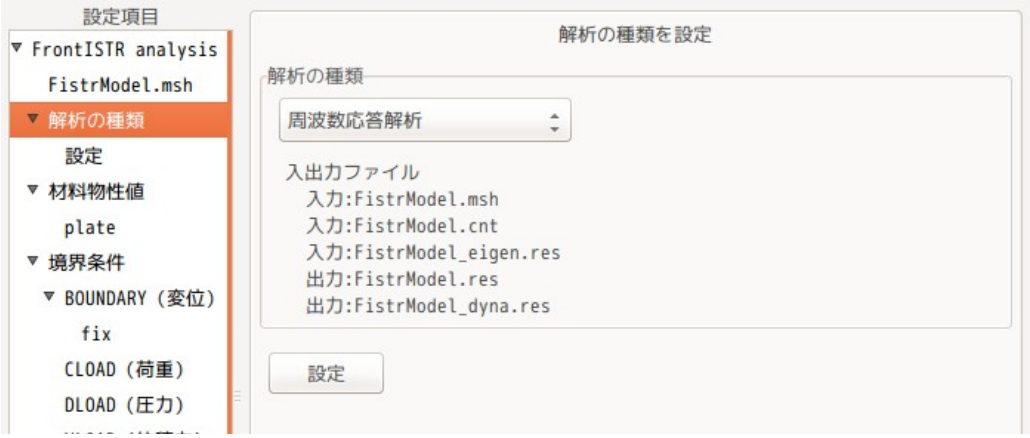
前項で使用した片持ち梁の固有値解析結果をコピーして workFolder としているので、モデル形状は、前項と同じ片持ち梁になる。

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。

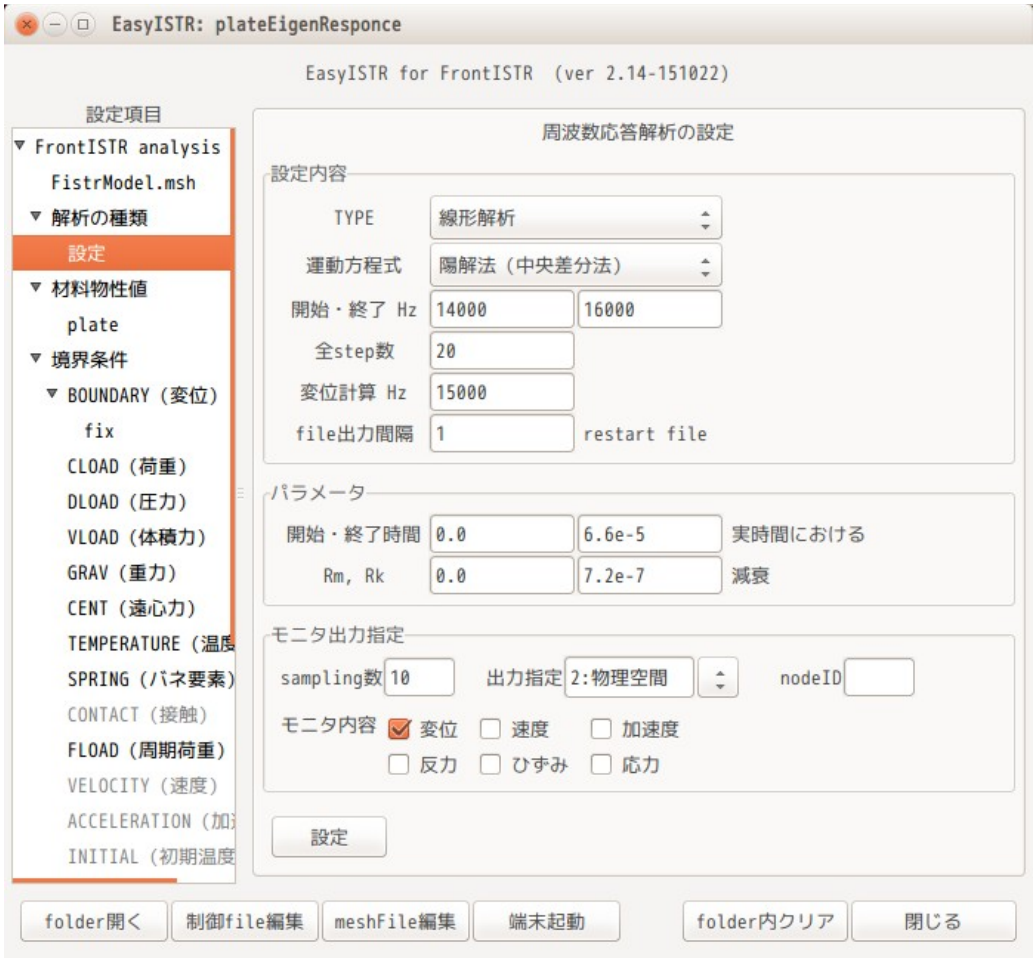
4-4-2-1.　解析の種類の設定

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。

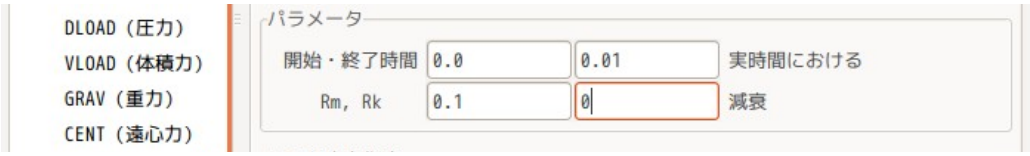


まず、TYPE と運動方程式は、そのまま「線形解析」、「陽解法（中央差分」を選択。
周波数は、前項の固有値解析の 1 次周波数が「536 Hz」の為、100～1000 Hz まで求めてみる。
周波数の間隔は、10Hz 間隔とする。（100～1000 までを 90 分割する事になる為、「全 step 数:90」を設定する。）下図の様に設定。

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。

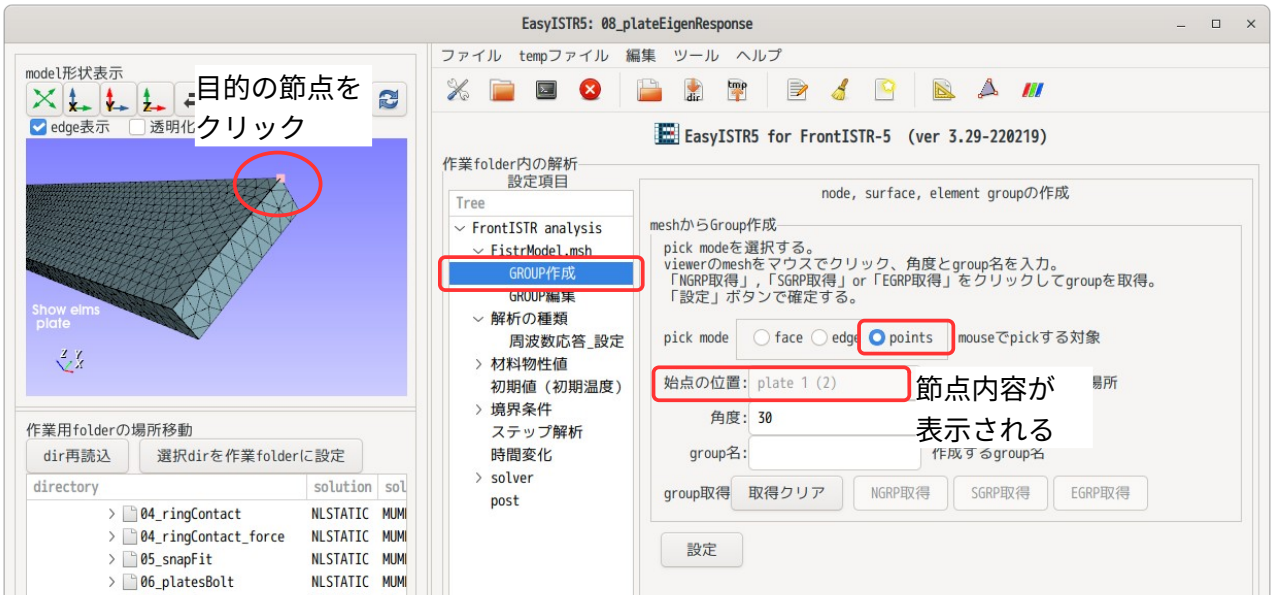
減衰は、質量マトリックスにかかる減衰（速度にかかる減衰）0.1 を設定する。下図のように設定。

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ので、先端の nodeNo を調べる必要がある。
この nodeNo は、EasyISTR 上で調べる事ができる。その方法は、下図の様に、「GROUP 作成」を選択し、pick mode を「points」に設定した後、VTK 画面上で目的の節点をクリックすると、クリックした節点の内容が「始点の位置」として表示されるが、この()内数値が nodeNo になっている。

```
始点の位置：plate 1 (2)
            -------- ---
                ↑       FrontISTR の nodeNo
            VTK の block 名と nodeo
```



選択した節点の nodeNo は「2」である事が確認できる。

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。

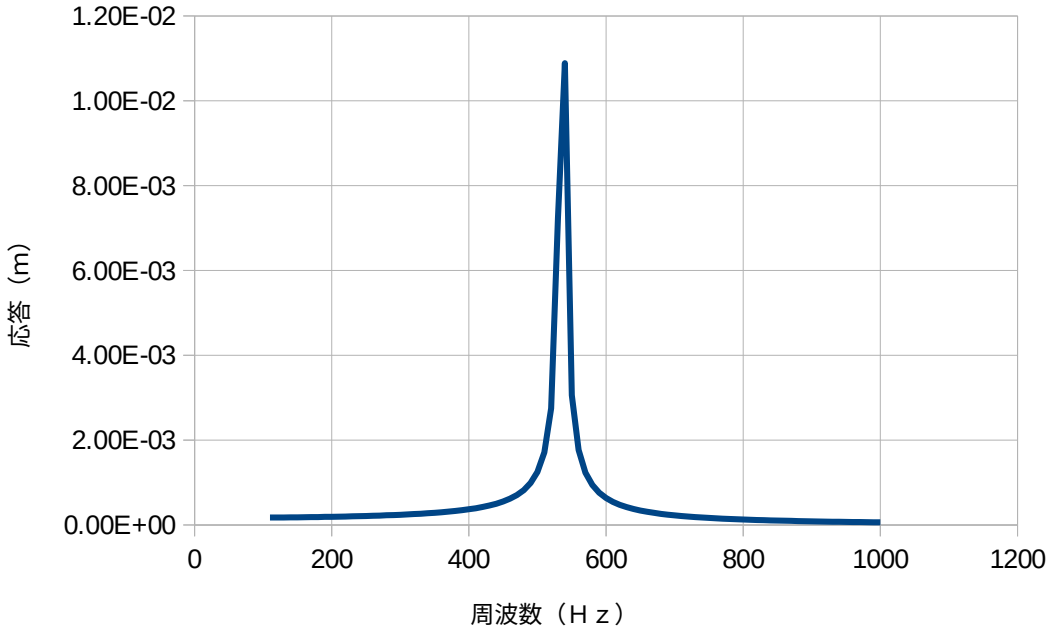
4-4-2-3.　境界条件の設定

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。

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 などで、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（⊿t）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

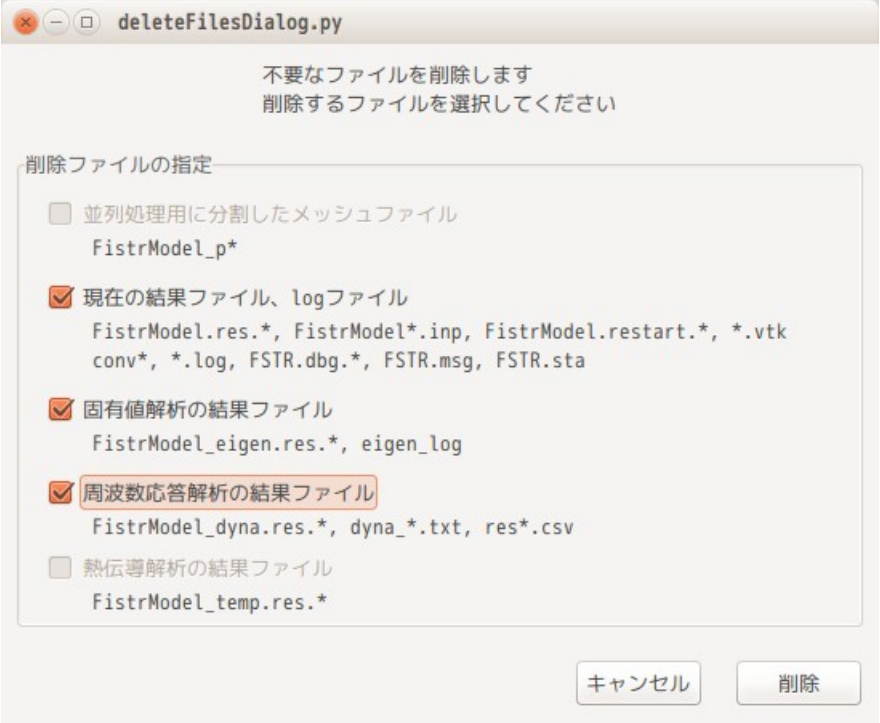
モデルは、固有値解析や周波数応答解析で使用したモデルと同じモデルを使う。
物性値もそのまま使用する。

4-5-2.　境界条件ー定値（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。

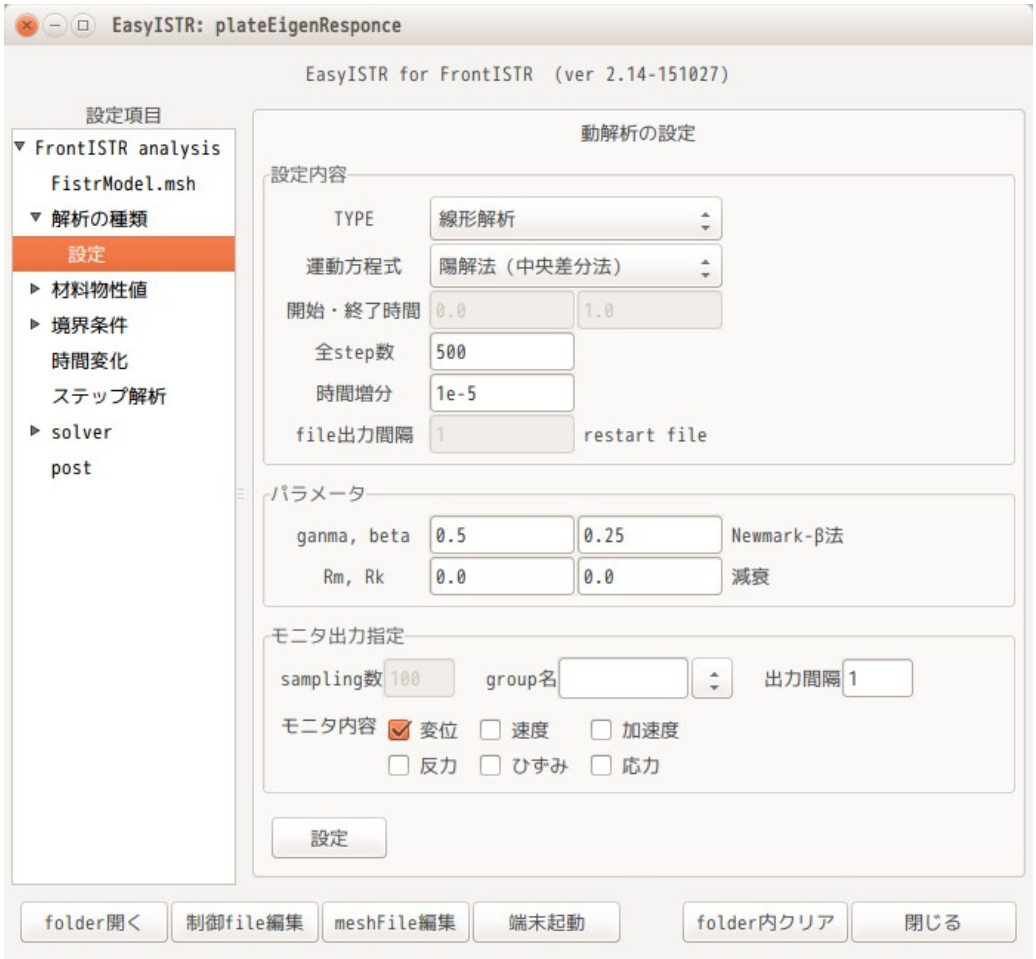


4-5-2-1.　解析の種類の設定

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

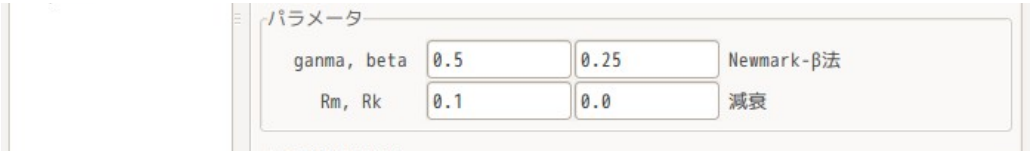
設定項目 Tree 内の「解析の種類」>「設定」を選択して、内容を設定する。
下図が、default の設定内容になる。



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（⊿t）は、全 step 数と時間増分で設定する。ここは、時間増分がいくつなら計算できるか確認できていないので、ここは default のままとしておく。下図参照。

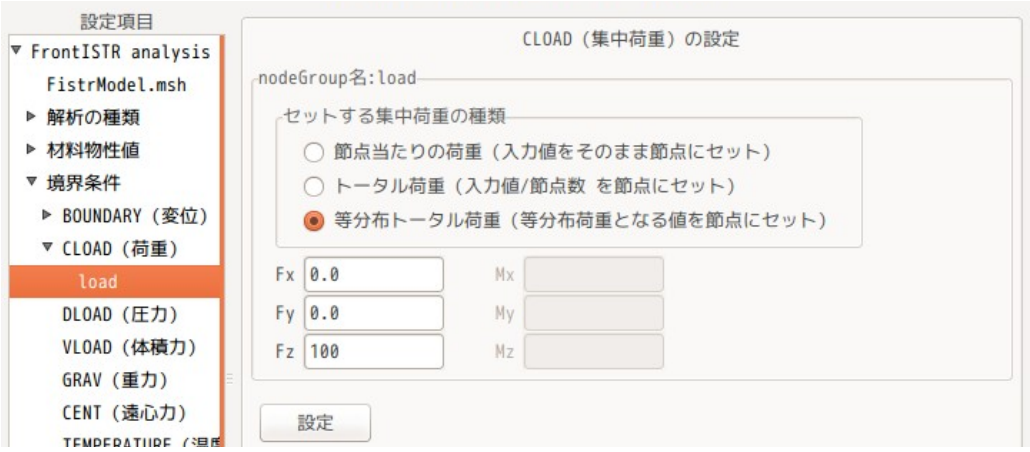
パラメータ内の gamma、beta は、運動方程式を陽解法に設定しているので、関係なくなり、このままの設定。
減衰は、質量マトリックスにかかる減衰 Rm を「0.1」に設定した。



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

4-5-2-3.　境界条件の設定

境界条件は、片持ち梁の端面（fix）を固定し、先端（load）に Z 方向等分布荷重で合計「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。

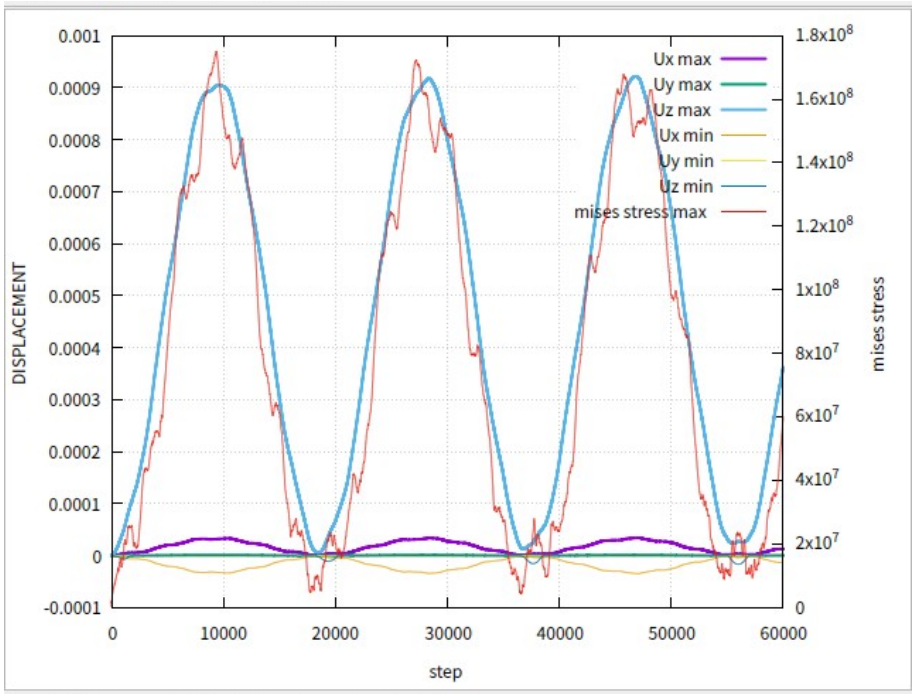
モニタの出力間隔は、「100」ステップ間隔にする。モニタ内容は、全てにチェックを入れた。下図の様に設定した。

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。

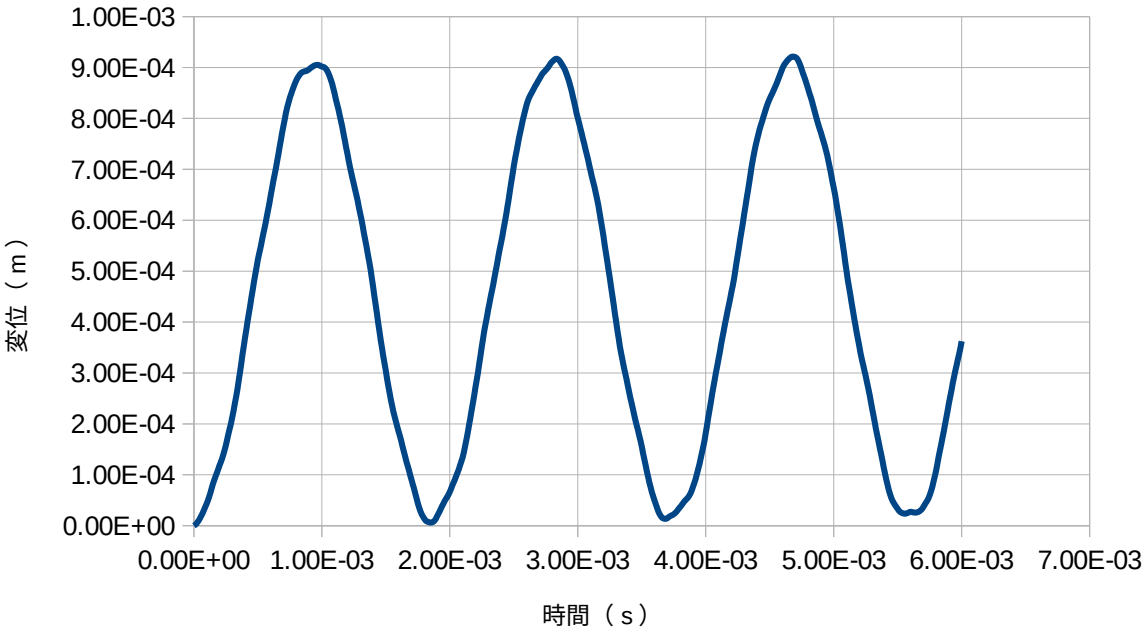
今回の時刻歴応答解析は、60000 ステップまで計算させるので、時間がかかる。この為、並列計算を行う。また、計算ステップが多いため、結果ファイルの出力頻度を 2000 回に設定しておく。（結果ファイルが 30 ヶできあがる設定。）

「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。

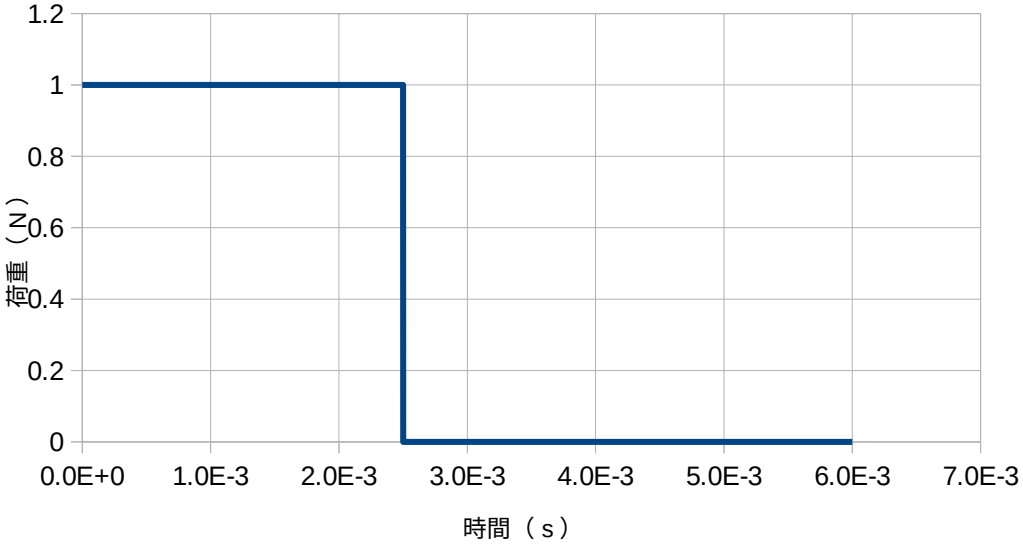


4-5-3.　境界条件に時間変化を与える（線形・陽解法）

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

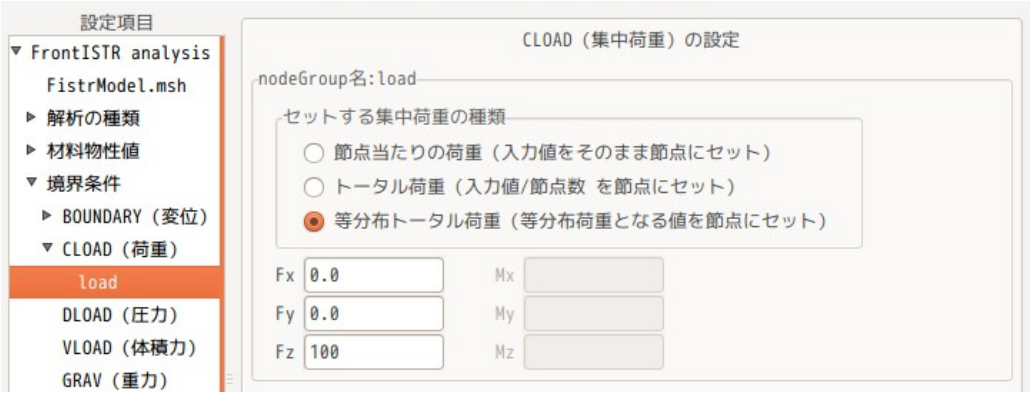
片持ち梁の先端（fix）の Z 方向の値に時間変化を与える。以下の変化を与える解析を行う。



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

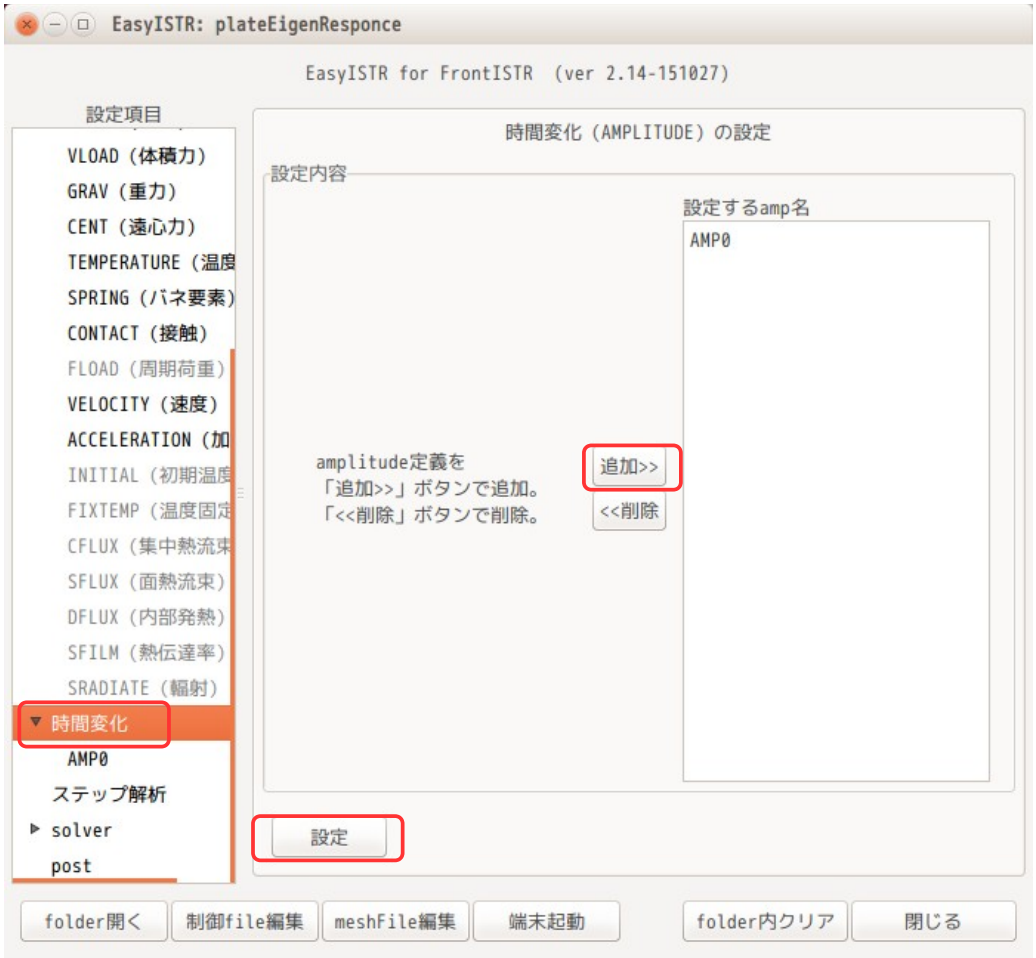
100N の等分布荷重をかけている。これは、変更せずこのまま。（下図参照。）



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 100 N の節点荷重）に、時間変化を与える。

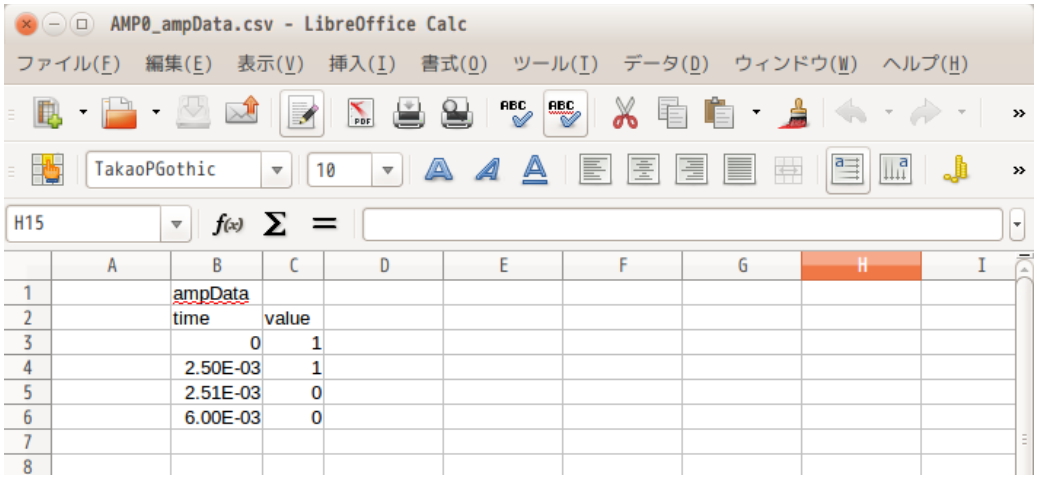
与える方法は、設定項目 Tree 内の「時間変化」を選択し、「追加」ボタンをクリックすると、設定する amp 名側に「AMP0」が追加される。この後。「設定」ボタンをクリックして、確定させる。
これにより、「AMP0」が設定項目 Tree 内に追加される。（下図参照。）



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。

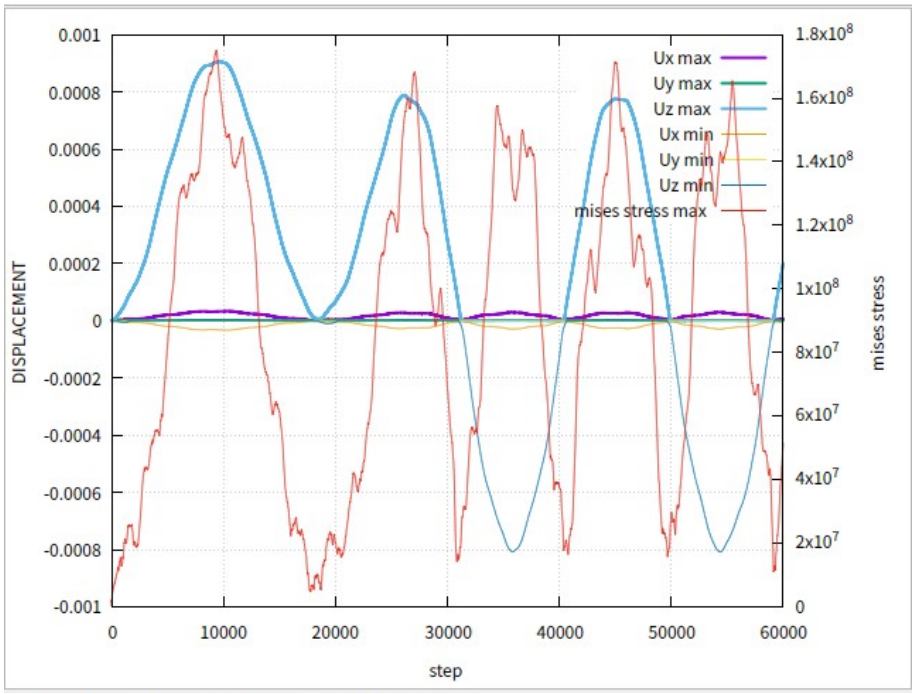


データ保存は、csv データとして保存する。作成したデータは、「AMP0_ampData.csv」と自動的にファイル名が振られ、保存されるので、後からでも内容を編集できる。

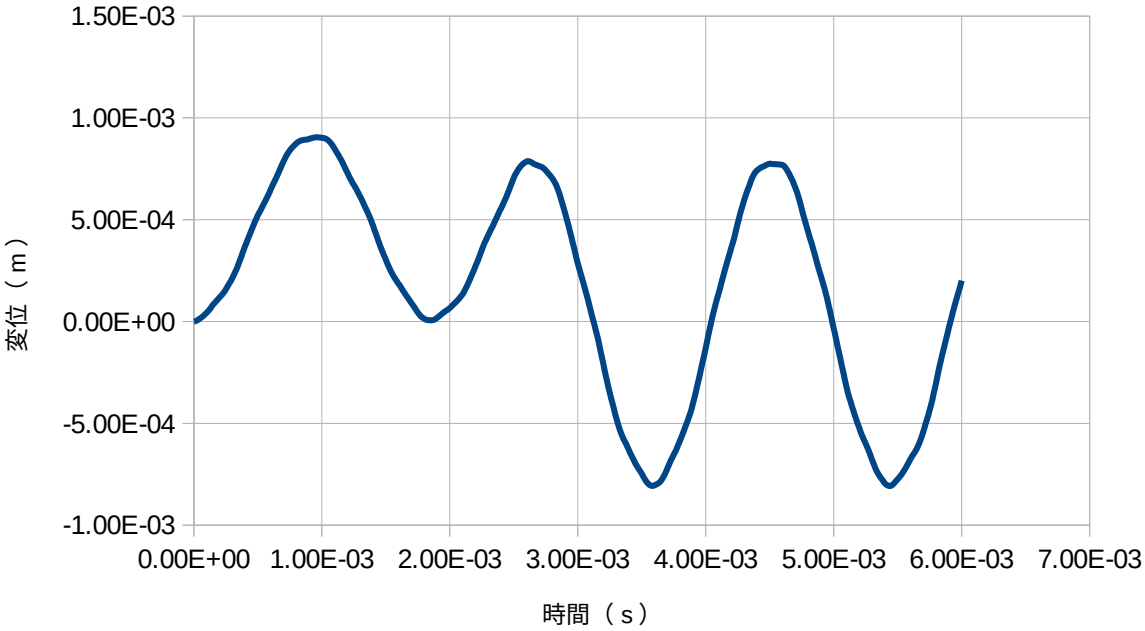
全ての設定（data 作成、時間変化を与える境界条件）の設定が終了した段階で「設定」ボタンをクリックして内容を確定させる。この操作により、FistrModel.cnt、FistrModel.msh ファイルへ書き込まれる。
（amp データは、メッシュファイルの最後に書き込まれる。）

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、毎回収束計算を行うので、陽解法に比べて時間増分（⊿t）を大きくできる利点がある。
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。
粘性系数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。

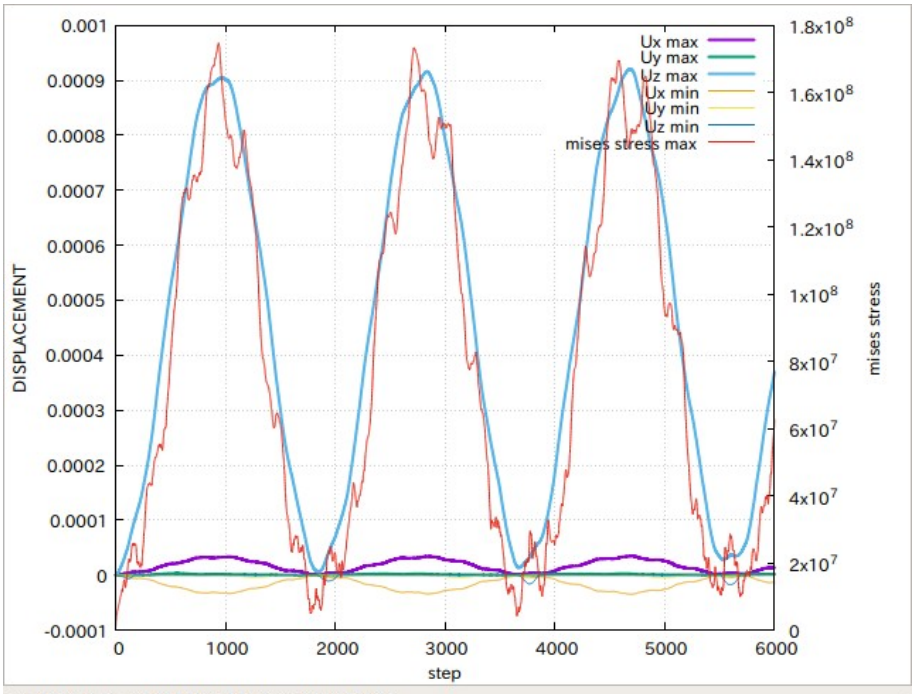
4-6-2-2. 境界条件の設定

境界条件は、前項と同じ設定とする。（下図参照）

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。



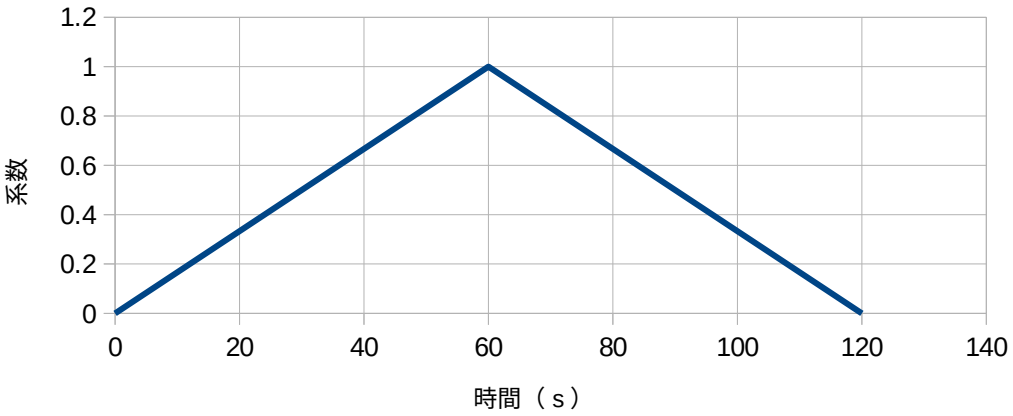
今回の陰解法は、前項の陽解法の条件に比べて、時間増分を「1e-7」→「1e-6」に変更して、陽解法では計算できない条件で計算させた。計算の step 数が減っているので、トータル的に計算時間は、早くなっている。

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。



この様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）

4-6-3-2.　境界条件の設定

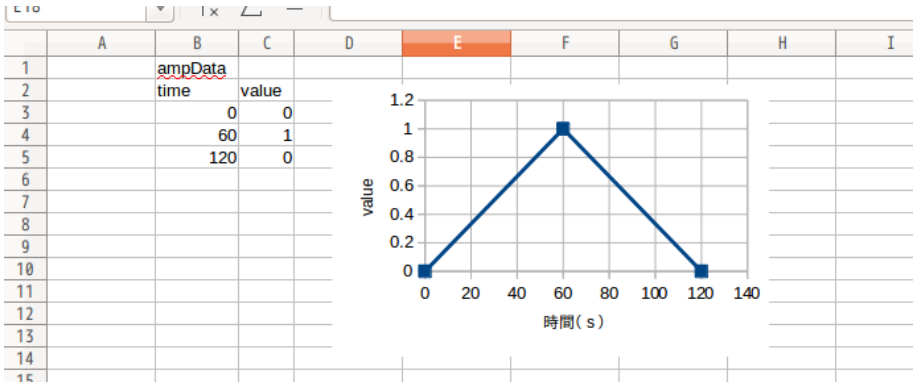
境界条件の設定は、前項と同じ条件とする。

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）

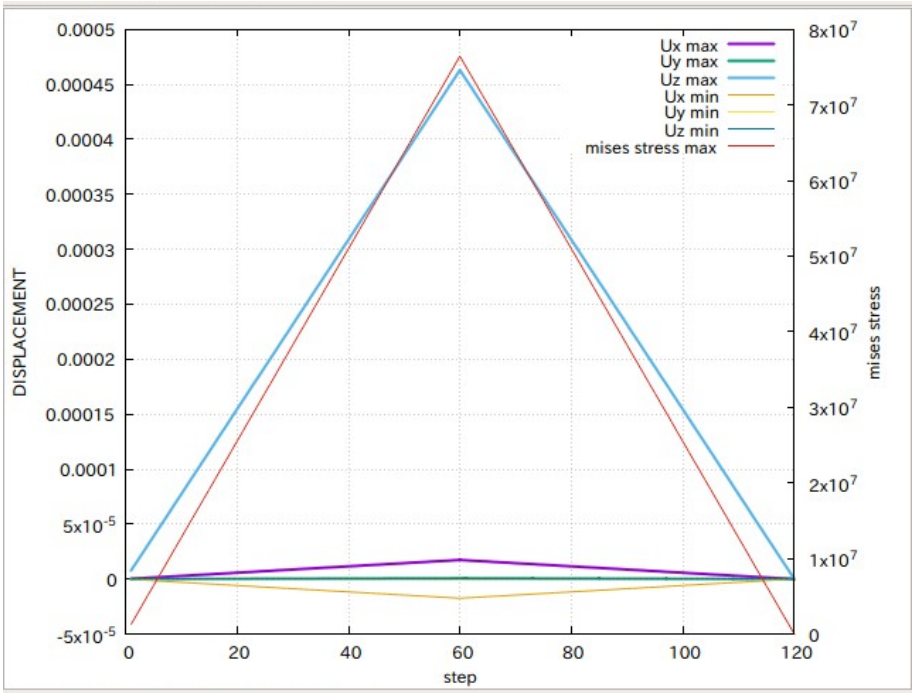


これで時間変化させる準備が整った事になる。

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。



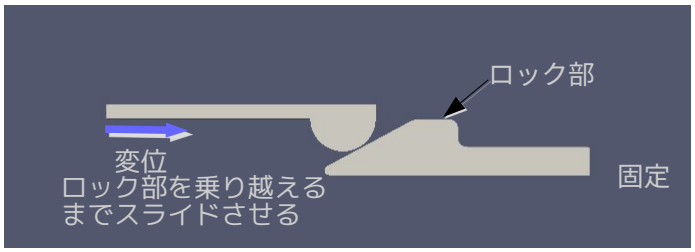
計算は、うまく走っている。ゆっくり荷重を変化させているので、細かな変動は存在していない。

4-7.　接触を含む時刻歴応答解析

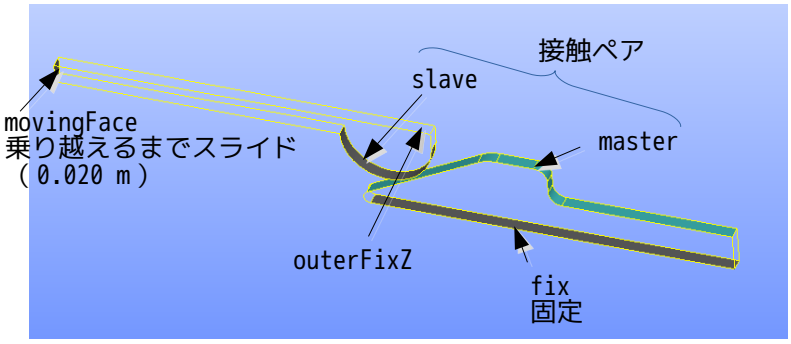
前項までの時刻歴応答解析は、線形の動解析であるが、ここでは、接触（非線形）の動解析を行ってみる。

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。



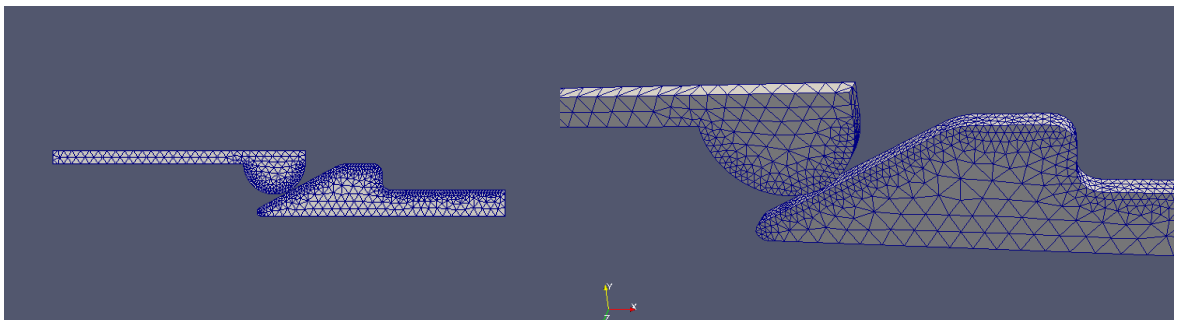
このモデルの group 化は、以下で実施している。



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）

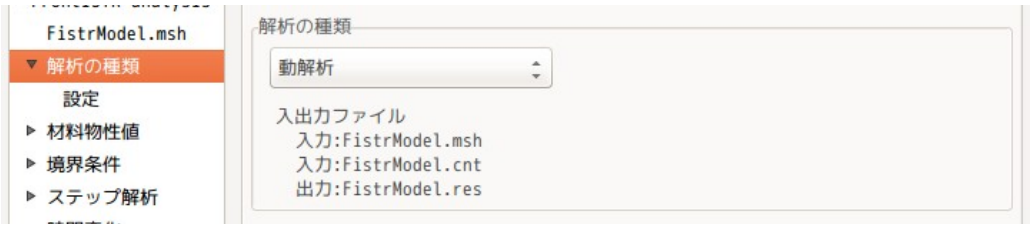
この形状のメッシュを下図の様に作成し、「snapFit2.unv」として保存している。接触面のメッシュは、細かくしている。

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

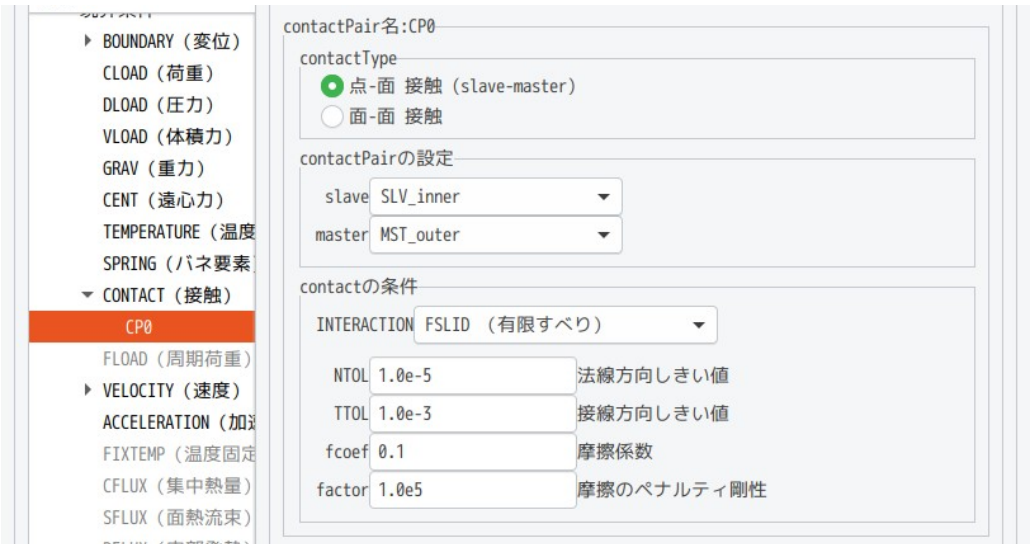
前項のメッシュを読み込み、倍率を 1/1000 としておく。
この後、解析は、「動解析」に設定する。



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「Alminum」とした。

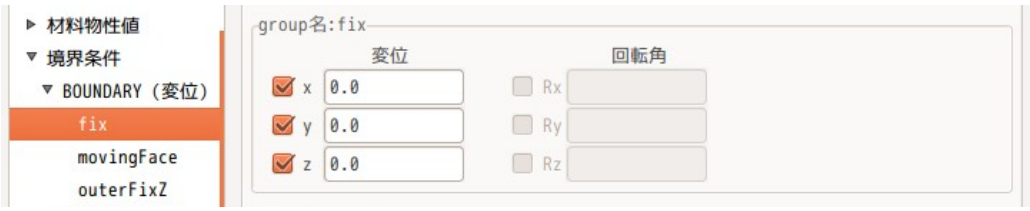
接触の設定は、以下で設定。（摩擦係数以外は、default のままの設定。）

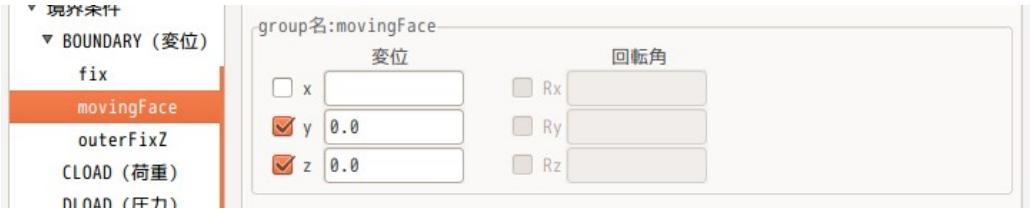


境界条件については、

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。

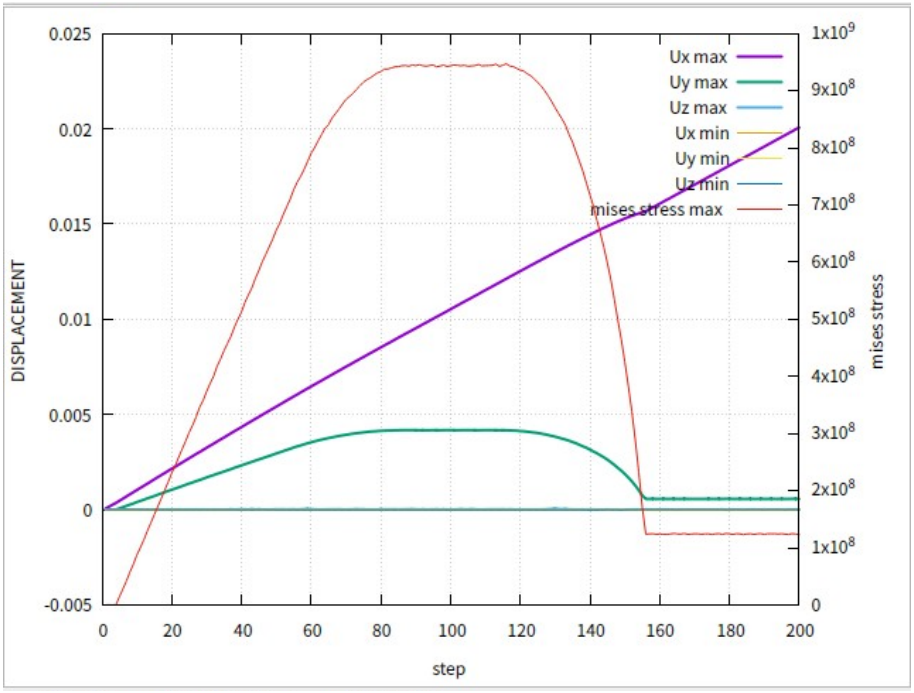




今回は、接触（非線形）を含む為、ステップ解析の設定が必要になる。
この設定は、DTIME（時間増分）、ETIME（終了時間）を設定したのみになる。（これ以外は、default の設定。）

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、「MUMPS」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。



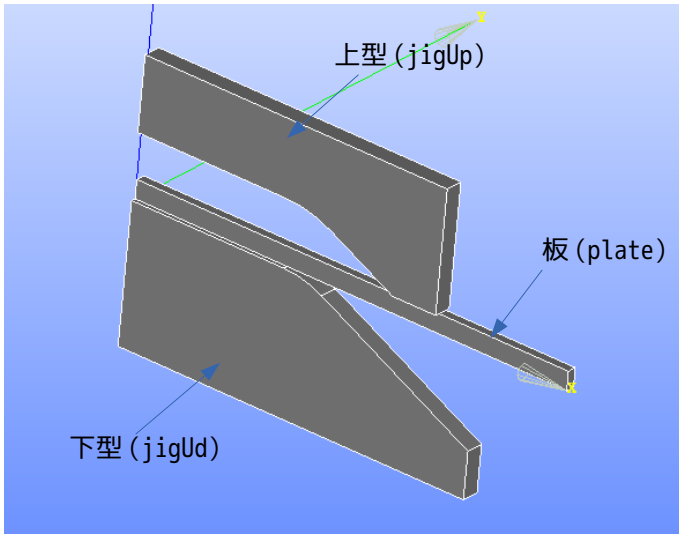
計算は、うまく走っている様子。動解析の為、自動時間増分を使っていない。この為、上図は、X 方向の変位がほぼリニアに変化している。

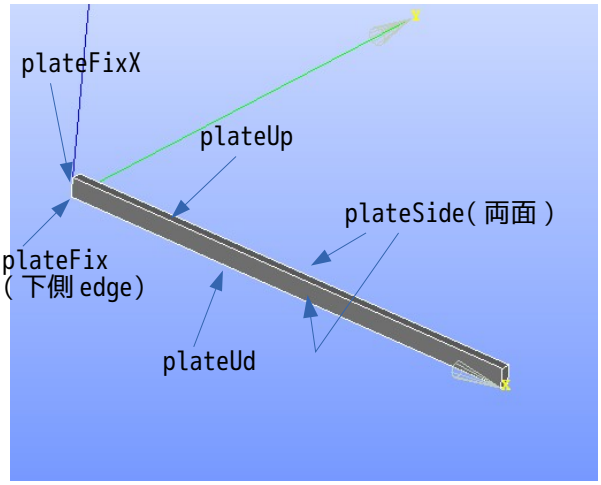
4-8.　接触を含む弾塑性解析（板のプレス曲げ加工）

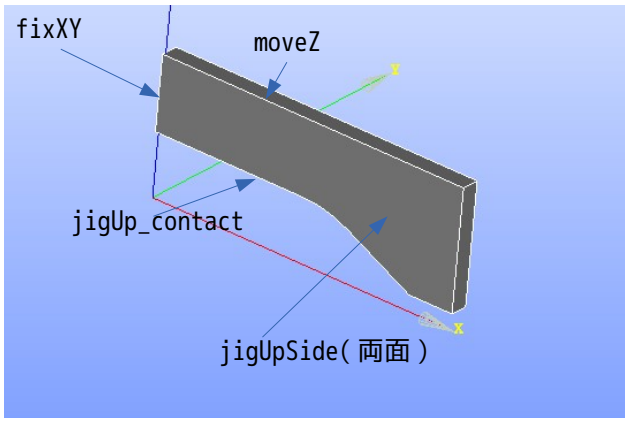
接触解析、弾塑性解析は、単独で解析できているので、ここで接触と弾塑性を含む解析（板のプレス曲げ加工）を行ってみる。
この解析は、接触、弾塑性、大変形の非線形解析を行うことになるので、計算負荷は大きくなる。

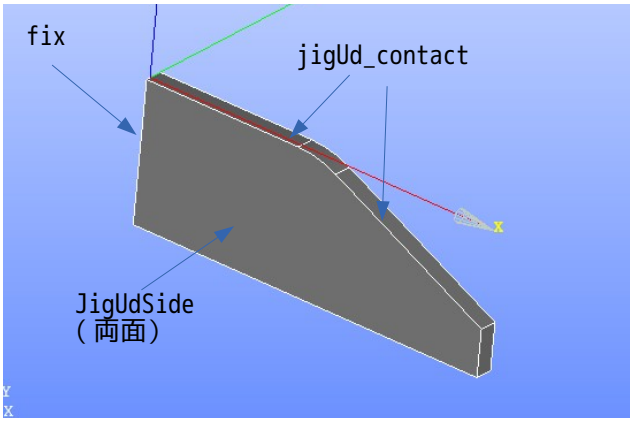
4-8-1.　モデル形状

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）





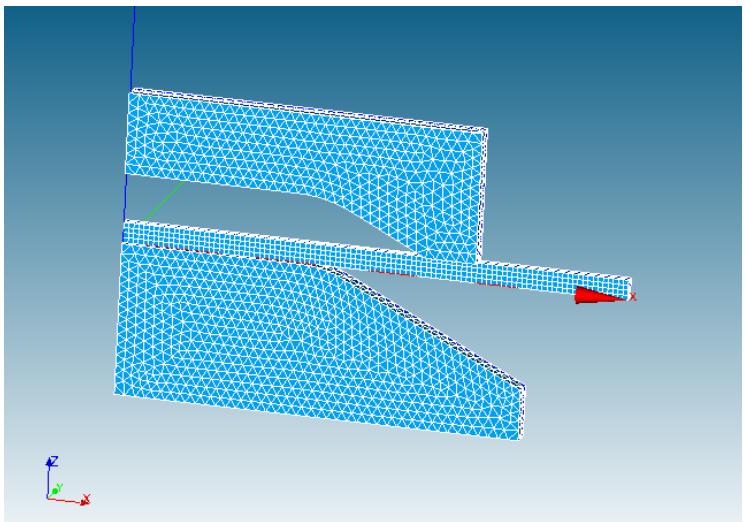




group 名	区分	meshGroup	備考
plate	volume	element	板
plateFixX	face	node	plate 根本側端面（X 方向固定）
plateUp	face	node	plate 上面（接触面：slave 側）
plateUd	face	node	plate 下面（接触面：slave 側）
plateSide	face	node	plate 両側面（Y 方向固定）
plateFix	edge	node	plate 根本端面下側 edge（XYZ 固定）
jigUp	volume	element	上型
fixXY	face	node	根本側端面（XY 固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



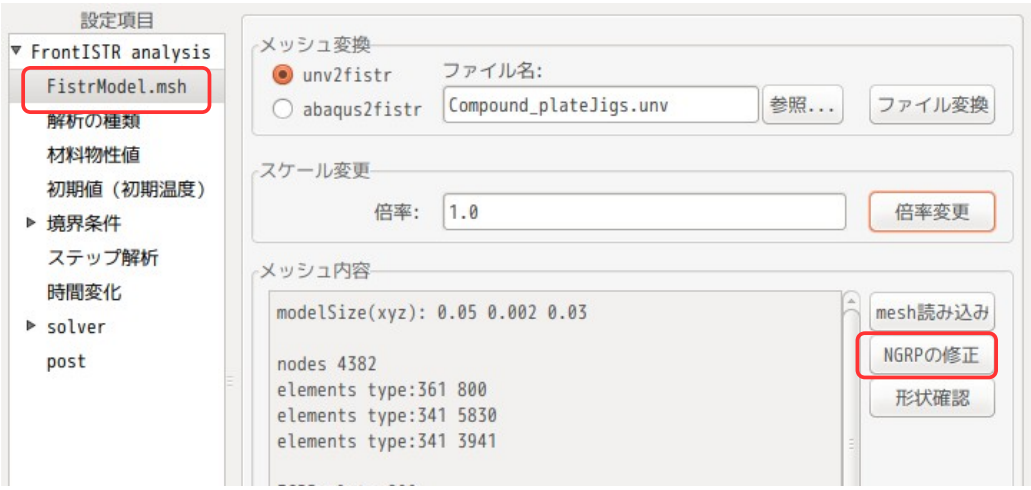
このメッシュを「Compound_plateJigs.unv」として保存する。（EasyISTR のインストール directory の unvFiles フォルダ内に保存されている。）

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

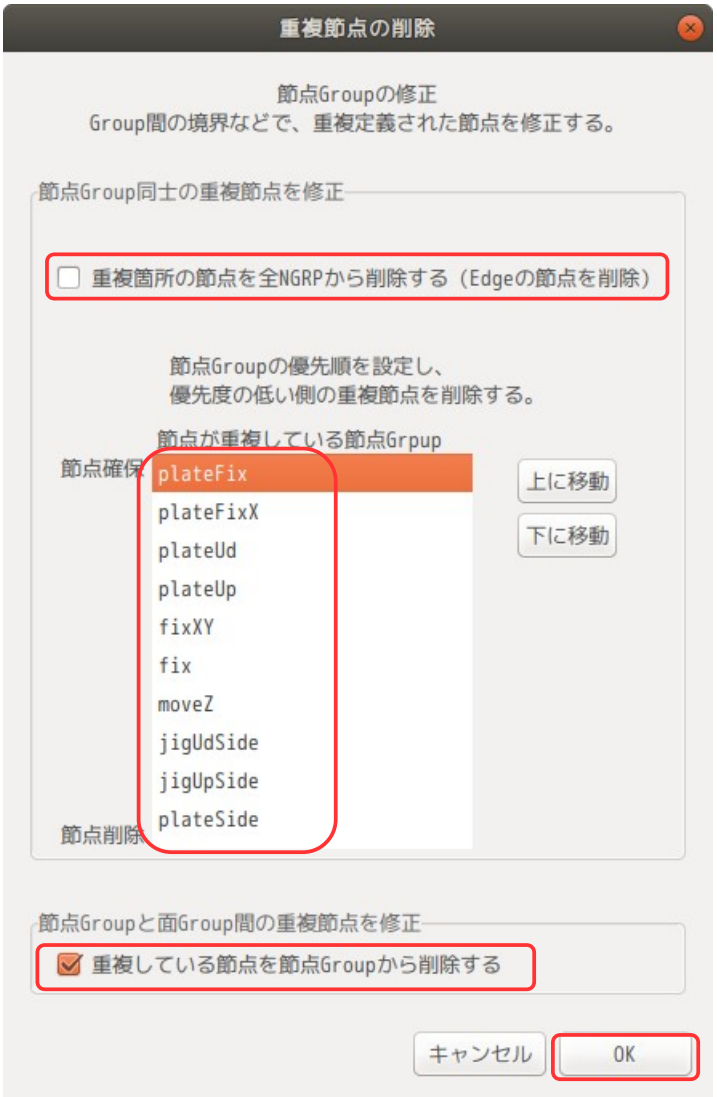
作成された unv メッシュを fistr 用のメッシュに変換し、scale を 1/1000 に設定しておく。
変換後、各節点 Group は、節点が重複して定義されている為、これを修正する。
修正の方法は、EasyISTR 上で、「NGRP 修正」ボタンをクリックして現れた画面「重複節点の削除」で修正する。



「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

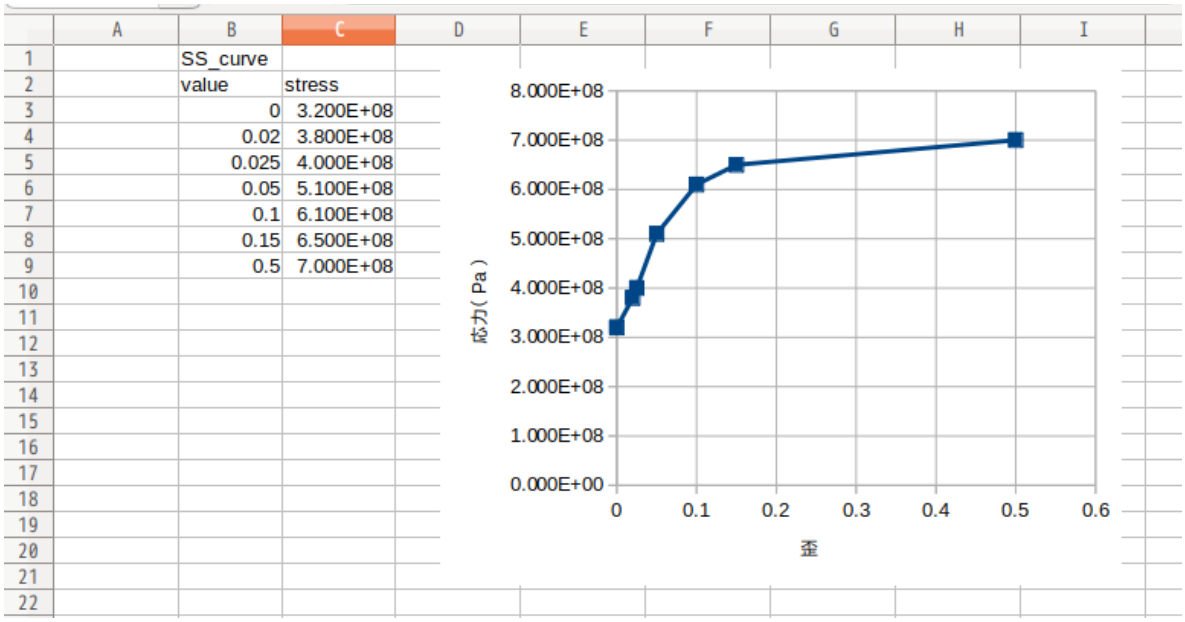
材料は、以下で設定する。

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

塑性材料（steelOther）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、steelOther 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）

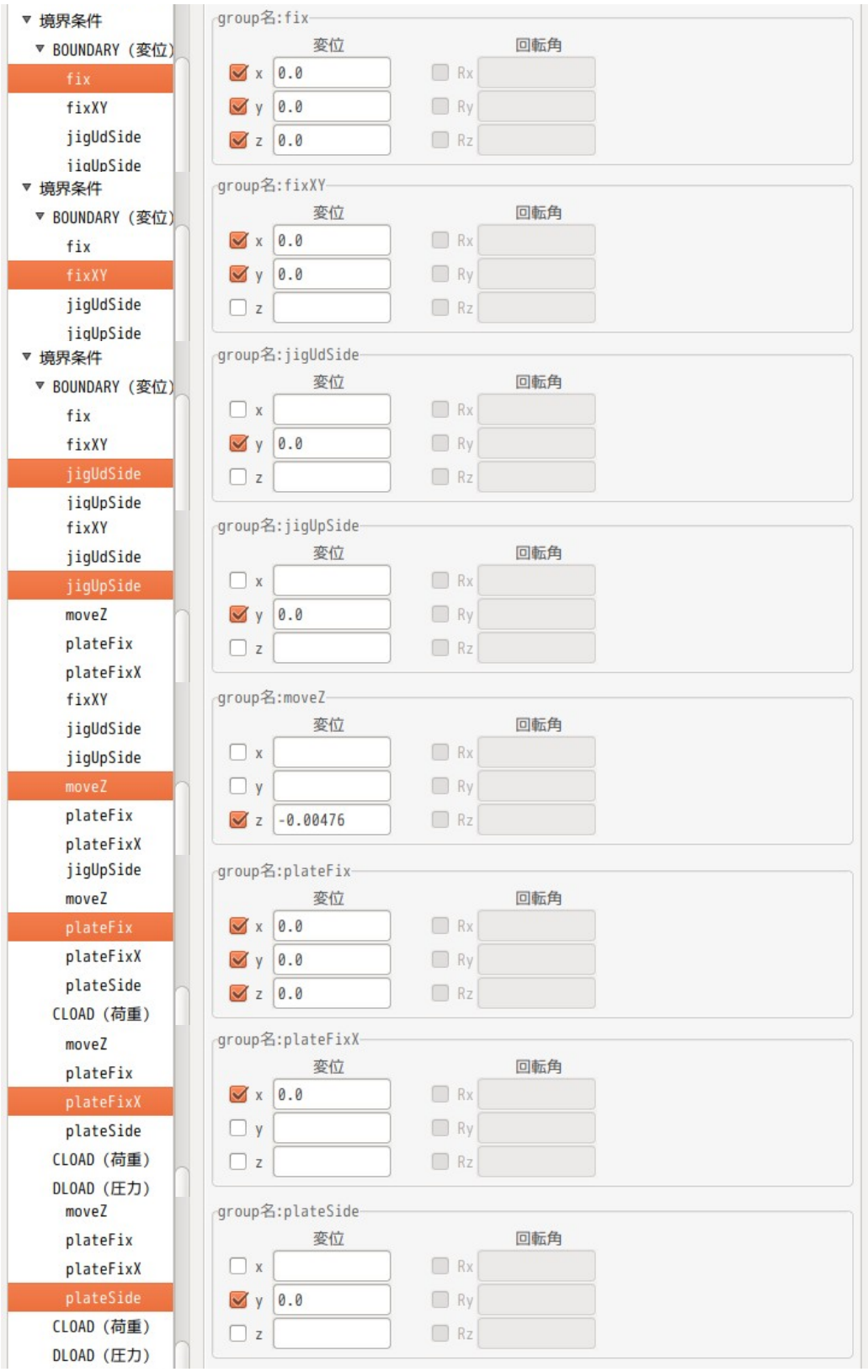


	A	B	C	D	E	F	G	H	I
1		SS_curve							
2		value	stress						
3		0	3.200E+08						
4		0.02	3.800E+08						
5		0.025	4.000E+08						
6		0.05	5.100E+08						
7		0.1	6.100E+08						
8		0.15	6.500E+08						
9		0.5	7.000E+08						
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									

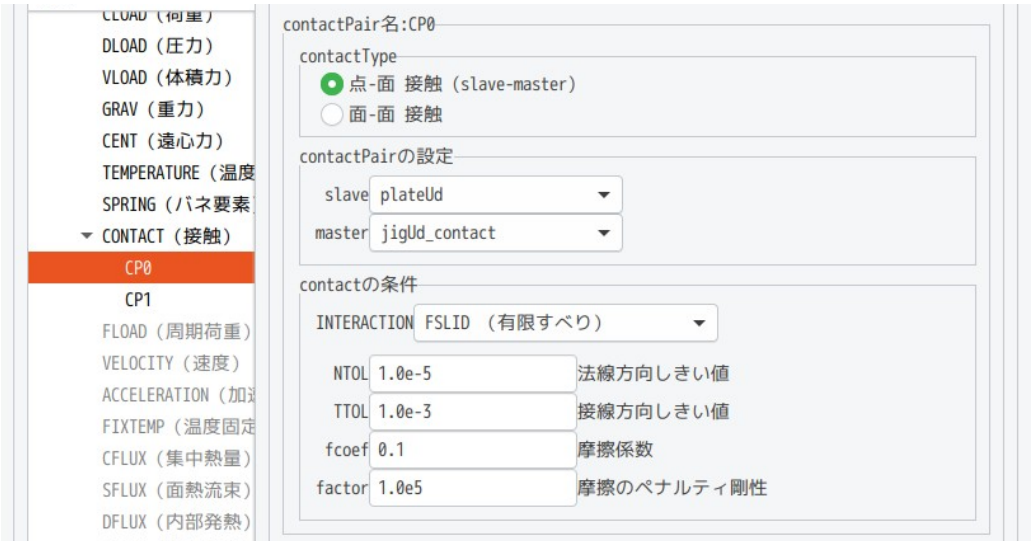
4-8-1-4.　境界条件の設定

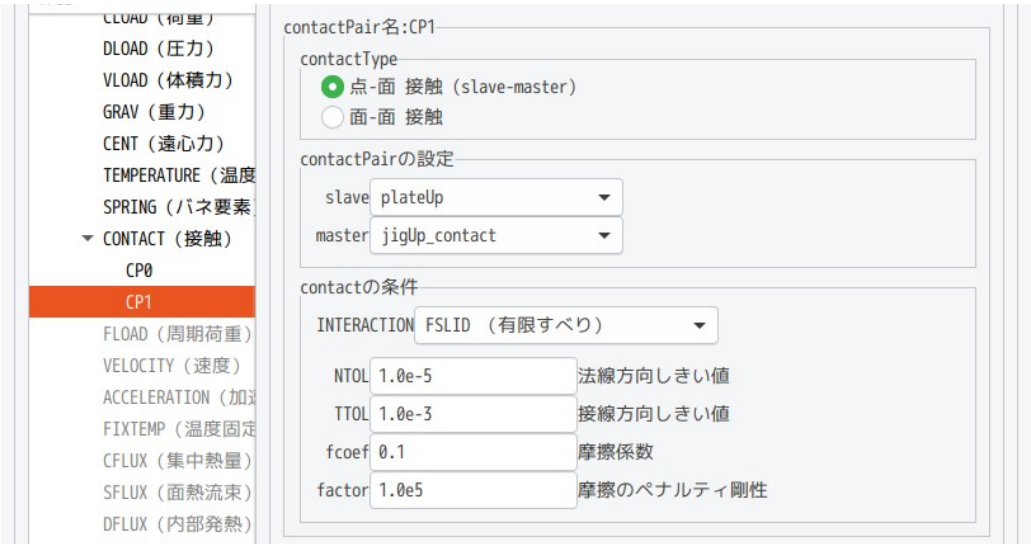
変位拘束は、以下で設定している。
上型は、Z 方向に移動させるので、「moveZ」については、Z 方向に「-0.00476」移動させる条件に設定して

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）を自動時間増分のデフォルト設定で、保存個数を各々「5」として設定する。

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）の設定も自動時間増分のデフォルト設定で、計算結果の保存は、等間隔で「3」個保存する設定。

自動時間増分の設定は、デフォルトの設定のままで計算してみる。

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。
　節点ひずみ:NSTRAIN
　節点反力:REACTION

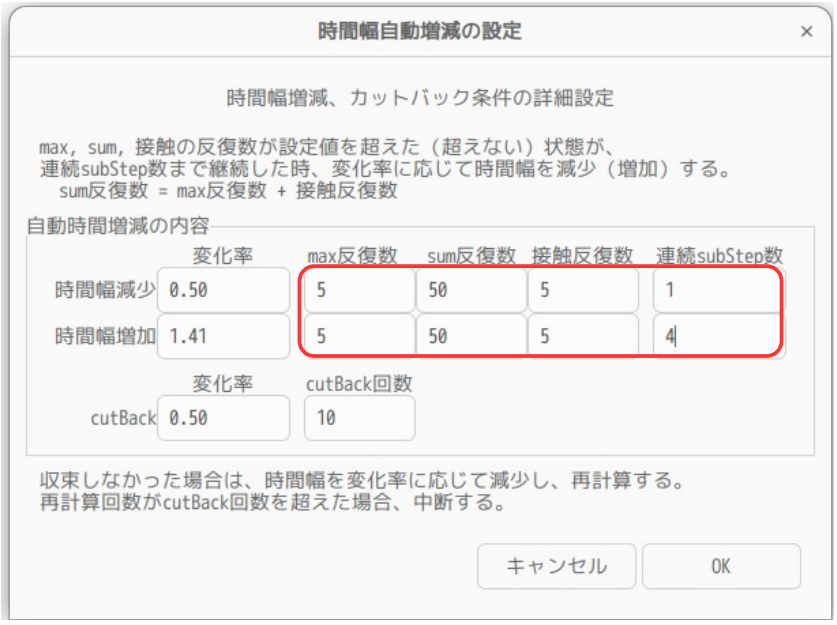
以上の設定で、計算開始させた結果、途中でエラー停止。

この為、自動時間増分の設定を調整する。

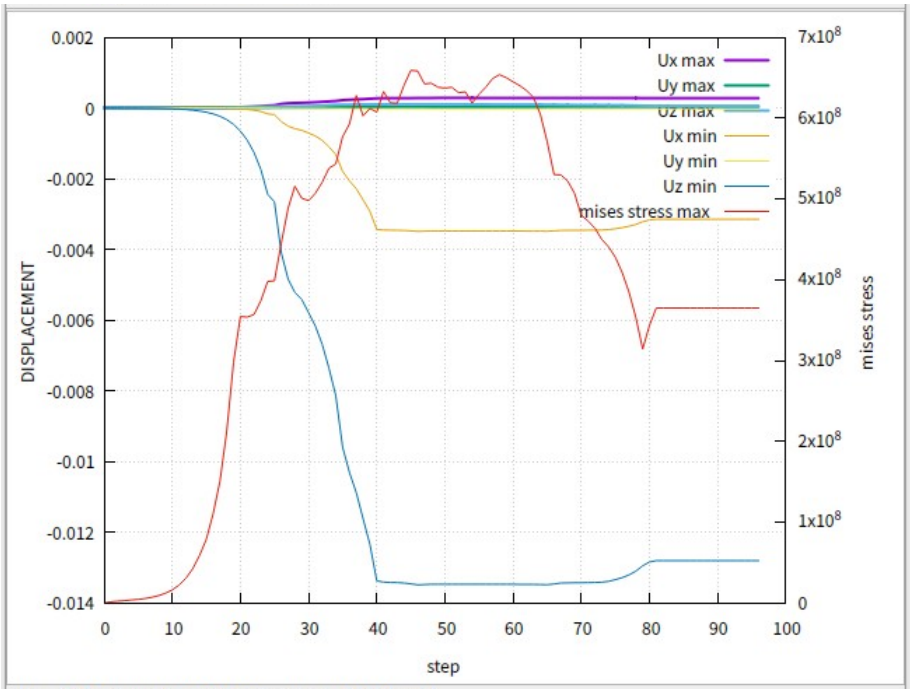
接触の状態が変化していく解析の為、収束し難くなっている。この為、時間増分の減少を敏感に、増加を鈍感に設定してみる。

　時間増分を減少：反復回数が 10 回を超えた時減少　→　5 回を超えた時減少させる
　時間増分を増加：反復回数が 10 回以内で収束した時増加　→　5 回以内の収束が 4 回連続した時増加

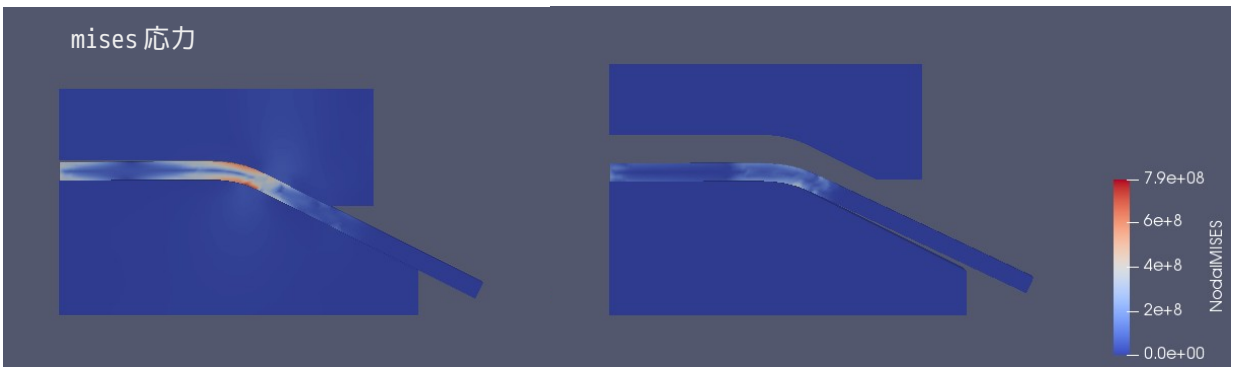
この設定を変更する場合は、dialog 中の「編集...」ボタンをクリックして、以下の画面上で内容を修正する。



この設定を STEP0、STEP1 ともこの設定で計算させてみる。
この結果が以下になる。今度は、うまく計算している。



スプリングバックや残留応力が確認できる。



この計算は、自動時間増分を使って計算させた結果、約 4 分で終了した。
FrontISTR-4 で subStep = 50 + 30 = 80 step で計算させた結果、3.5 時間かかっていた。この為、自動時間増分の計算が早いことが判る。

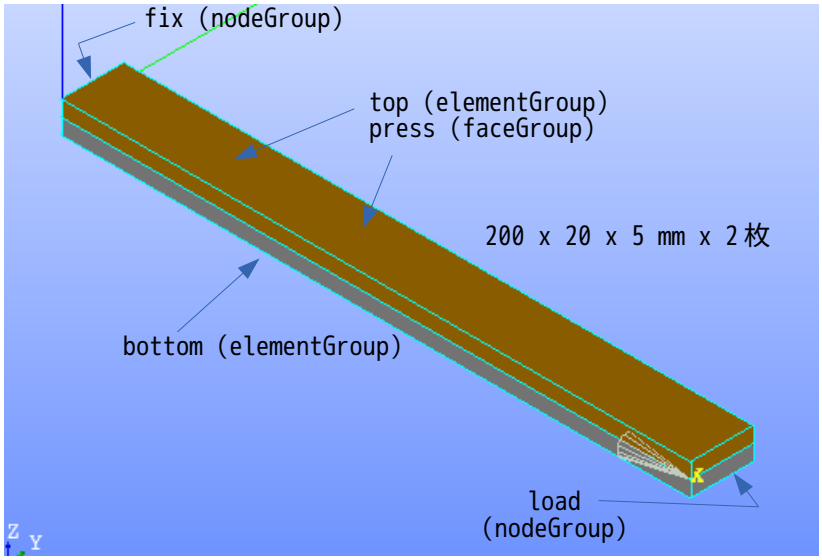
もし、計算が収束しなかった場合は、「FSTR.sta」ファイルの内容を確認する事で、どの時点でどんなエラーが発生したかが確認できる。これを確認した後、自動時間増分の設定を操作する。

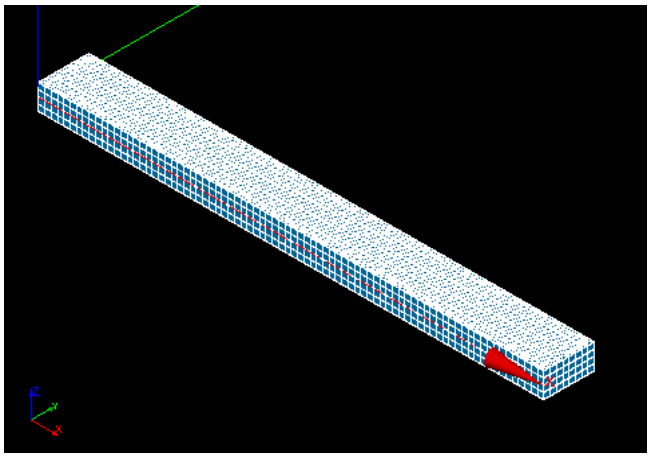
4-9.　熱応力解析

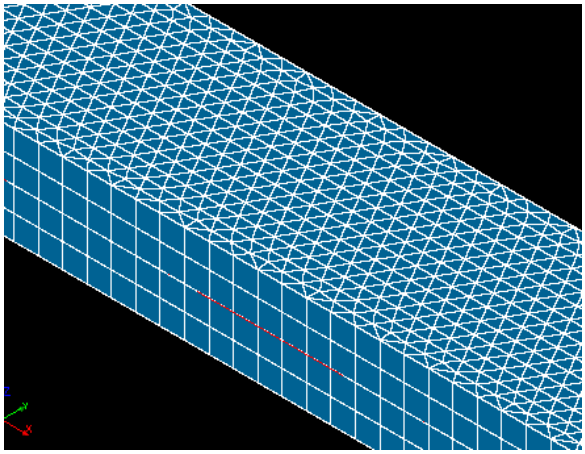
EasyISTR で熱応力解析を行ってみる。
モデル形状は、Aluminum と Steel を貼りあわせたバイメタルとして解析する。
workFolder は、「CAE-fistr/Case/biMetal」フォルダを作成して、この中で解析を行う。

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。







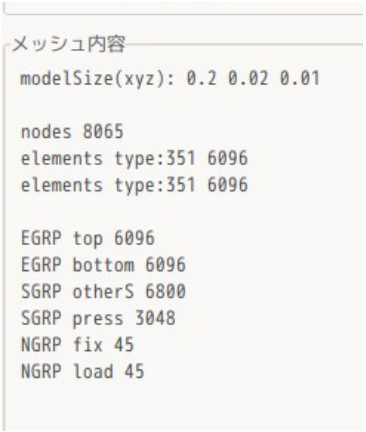
解析は、片側（fix）を固定して、温度上昇させる解析を行ってみる。

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。）




```
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.　初期温度の設定

熱応力の計算には、初期温度の設定が必要になる。この温度で応力が「0」状態。
EasyISTR では、ver 3.29 以前では、この初期温度は、温度の境界条件設定時に設定していたが、ver 3.30 以降から初期温度の設定を分離し独立させている。
この初期温度の設定データは、msh ファイル内に設定される。

設定は、以下の様に設定した。ALL（全節点）を 20℃に設定

4-9-2-5.　境界条件の設定

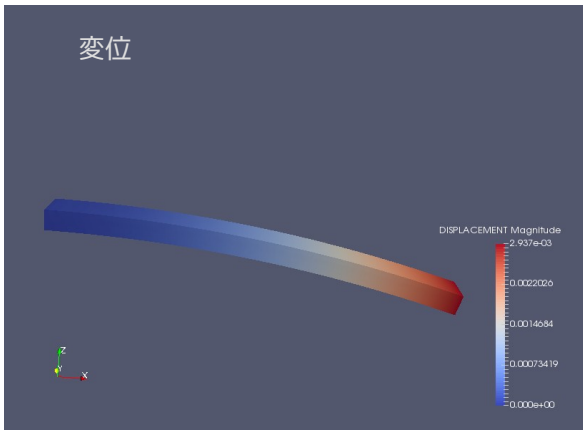
境界条件は、バイメタルの片側（fix）を固定し、温度を 20℃から 100℃に上昇させる。
下図は、fix の境界条件（固定）の内容。

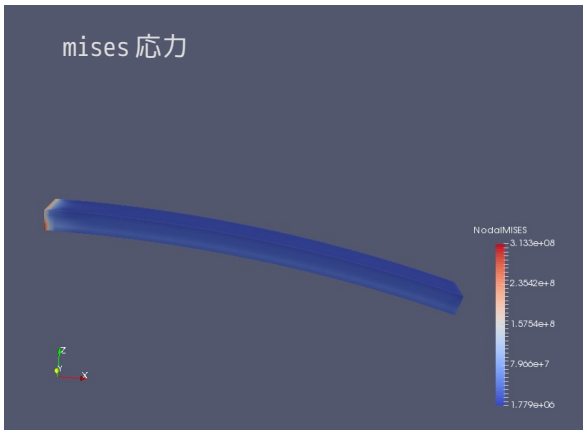
温度の設定は、Temp_Value を「100.0」に設定する。尚、この画面上で初期温度の設定値「20」が確認できる。初期温度が、応力「0」の状態になる。

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。

4-9-2-6.　計算開始、結果の確認

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





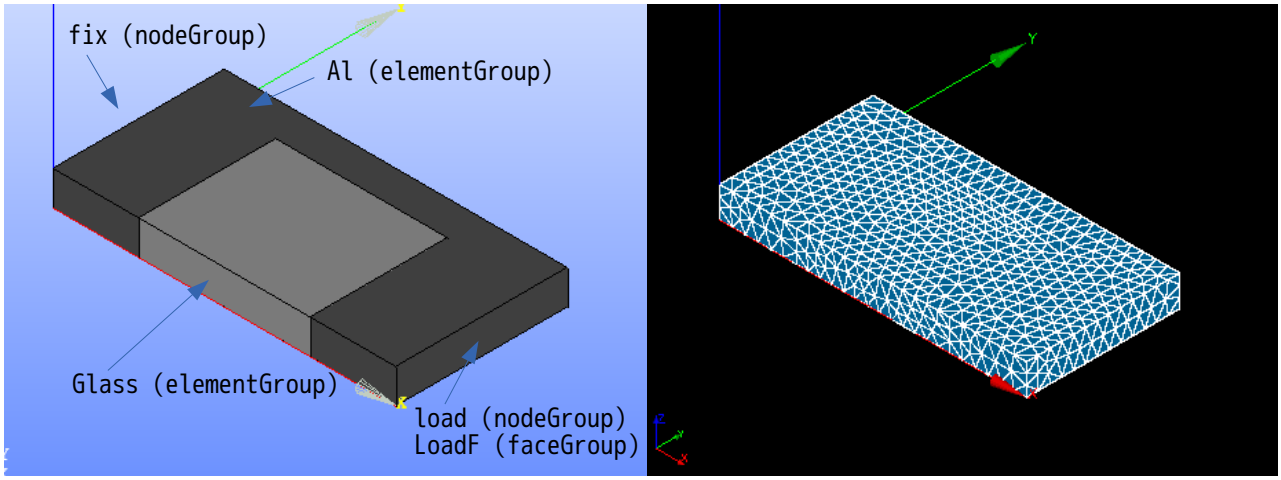
温度変化により、バイメタルが変形している。

4-10.　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。



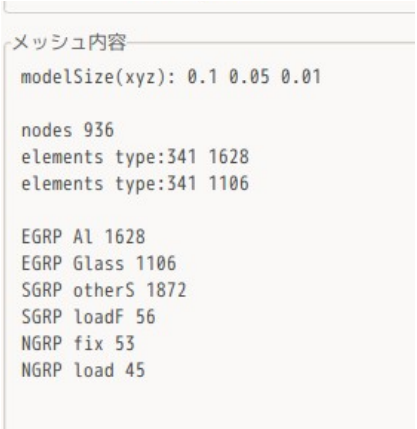
でき上がったメッシュを「AlGlassPlate.unv」として workFolder 内に保存しておく。

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、1/1000 にスケール変更した結果が以下になる。




```
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 ℃で温度固定し、先端（loadF）に熱流束 1e5 W/m2 を加える。

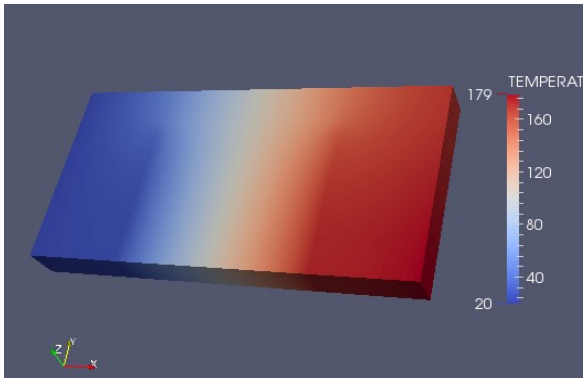
温度固定の方法は、設定項目 Tree 内の「FIXTEMP（温度固定）」を選択して、設定する nodeGroup「fix」を確定して、この fix に設定温度「20」を設定する。　下図が設定した状態。

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、「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。

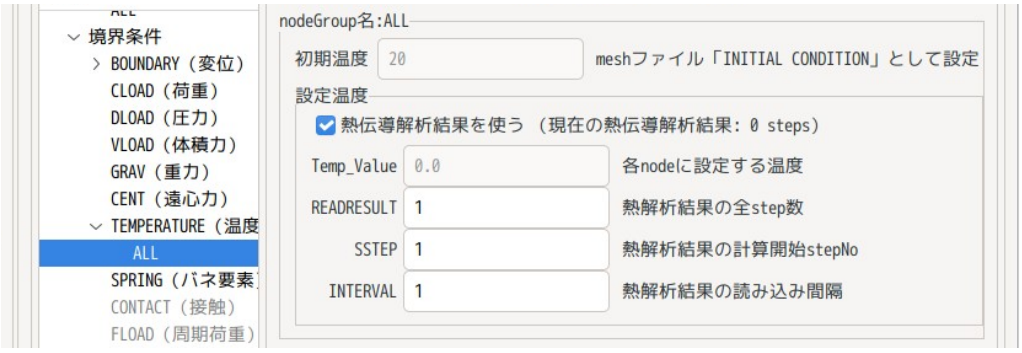
4-10-3-3.　初期温度の設定

初期温度は、20℃を設定する。この温度の状態が応力「0」の状態になる。

4-10-3-4.　境界条件の設定

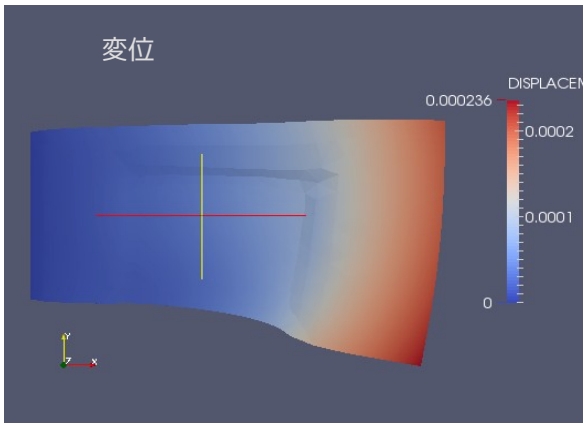
境界条件は、plate 端面（fix）を固定する。下図参照。

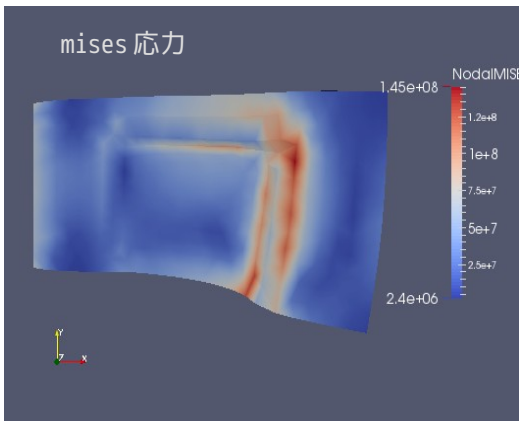
温度条件は、熱伝導解析結果を使用するので、以下のように設定する。
チェックボックス「熱伝導解析結果を使う」にチェックをつける。熱伝導解析結果は、静解析の結果のため、READRESULT、SSTEP、INTERVAL は、全て「1」になる。



4-10-3-5.　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、計算結果を 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

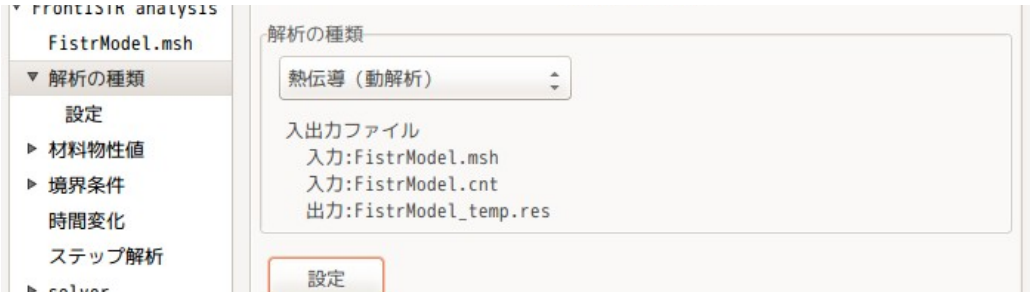
前項の熱伝導（静解析）と同じモデルを使用する。

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。



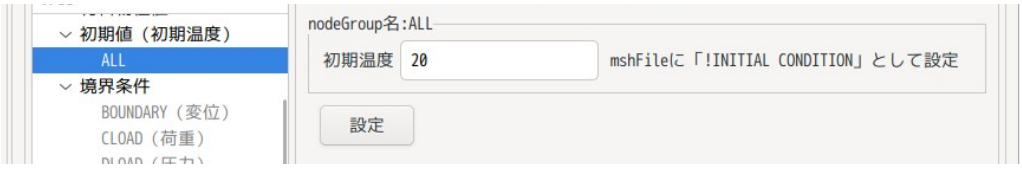
この後、設定項目 Tree 内の「解析の種類」>「設定」を選択して、熱伝導解析の内容を設定する。

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.　初期温度

まず、初期温度「初期値（初期温度）」を 20℃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。



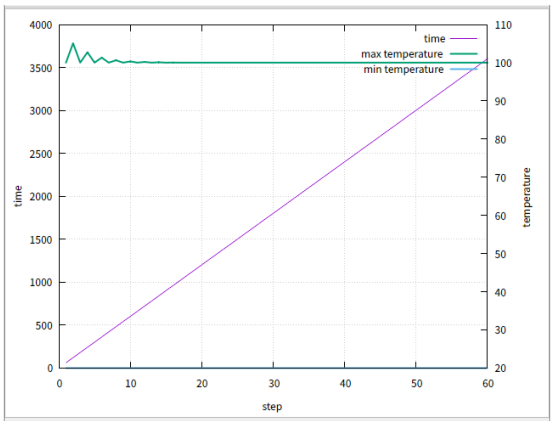
4-11-2-3.　境界条件

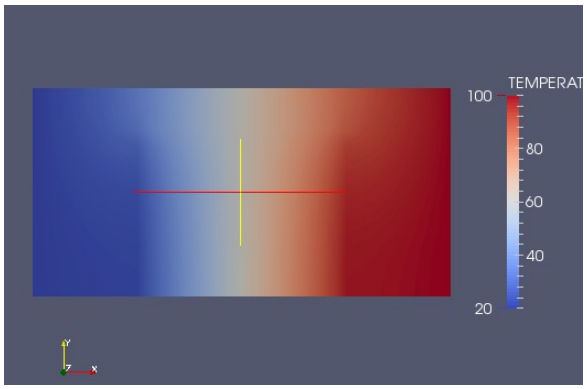
plate 端面の fix を 20℃に温度固定し、load を 100℃に温度固定する。以下の様に設定。

4-11-2-4.　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。

この解析は、動解析の為、plotStepMonitor が表示できる。下図のように、min,maxTemperature の推移が確認できる。
計算終了後、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。

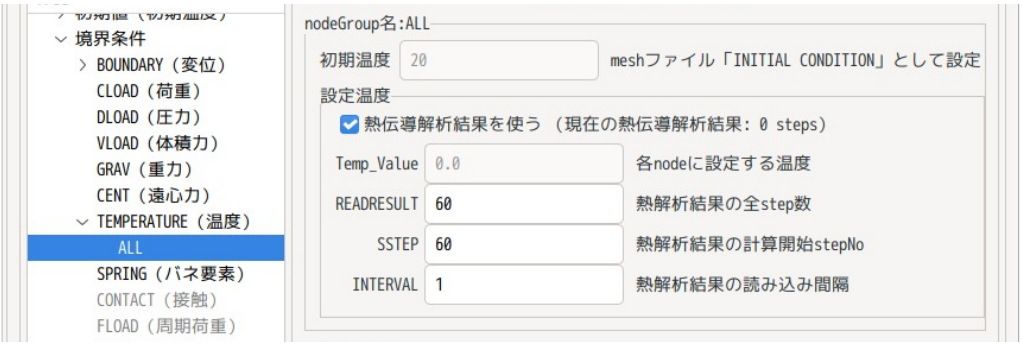
4-11-3-3.　境界条件の設定

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℃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。

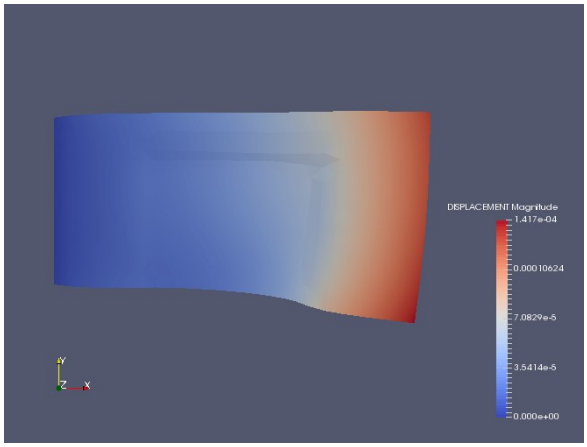
熱応力解析は、このデフォルトの設定を使って計算する。この計算は、最終の 60step の温度データを使って計算する設定になっている。



4-11-3-4.　計算開始、結果の確認

計算開始する前に、vtk ファイルを削除しておく。（削除しなくても問題ないが、熱解析結果の応力解析結果が上書きされる。）
以下の様に「現在の結果ファイル、log ファイル」にチェックして、vtk を削除する。
熱伝導解析の結果ファイルは、削除しない。（チェックしない。）これを削除してしまうと、熱応力解析ができなくなる。

削除後は、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、paraView で結果を確認する。
下図が確認した結果になる。

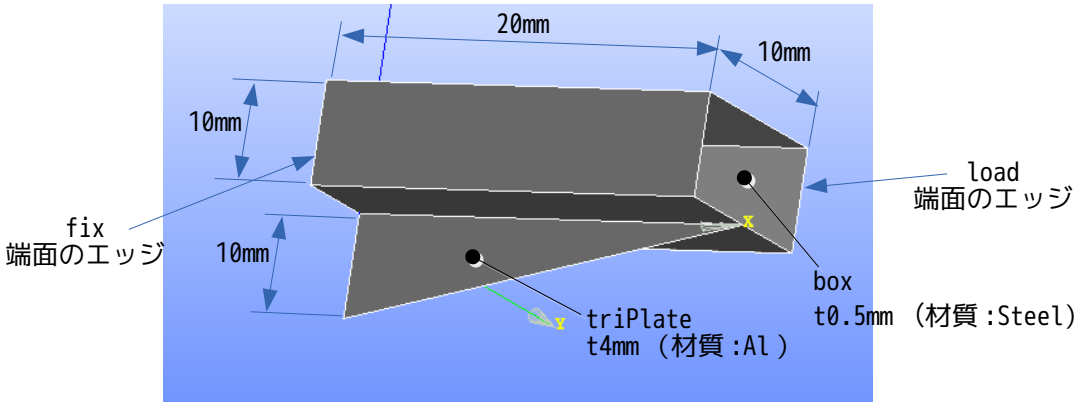


4-12.　シェルの解析

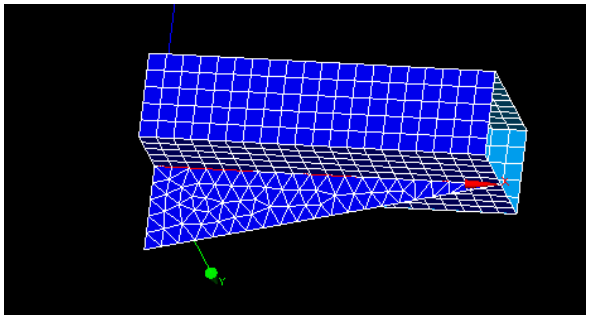
シェル（面）のみで構成されたモデルの線形弾性静解析を行ってみる。solid 解析との違いは、以下になる。
シェルの板厚を材料物性値と共に設定する。
BOUNDARY の境界条件は、各軸の回転が追加されている。
応力、ひずみの計算結果は、シェル表裏の値が存在する。
シェルの場合、線形 solver は DIRECT or MUMPUS のみ対応。

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。

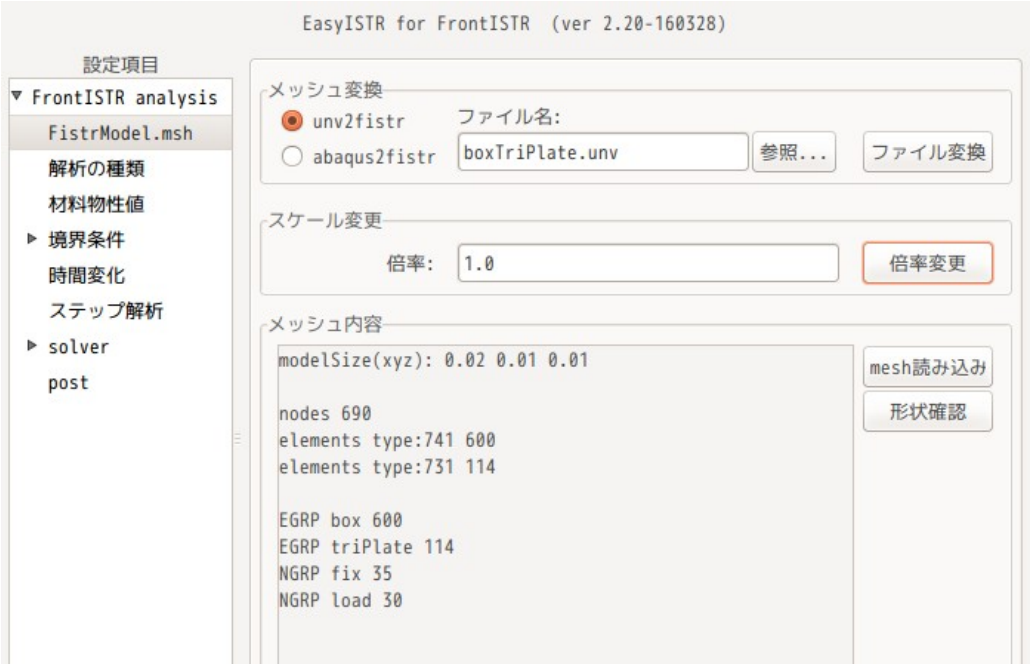


このモデルを、salome で以下の様にメッシュを作成する。
box を四角形 1 次要素、triPlate を三角形 1 次要素でメッシュを作成している。



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。



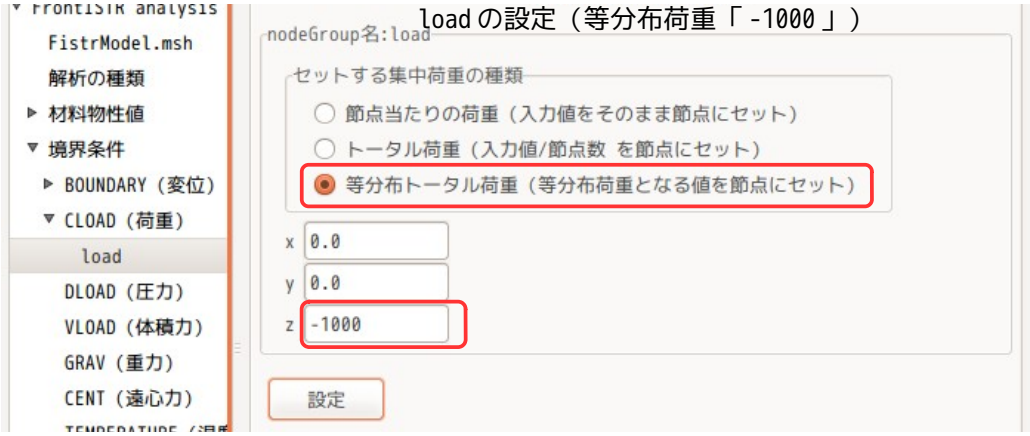
4-12-2.　材料の設定

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

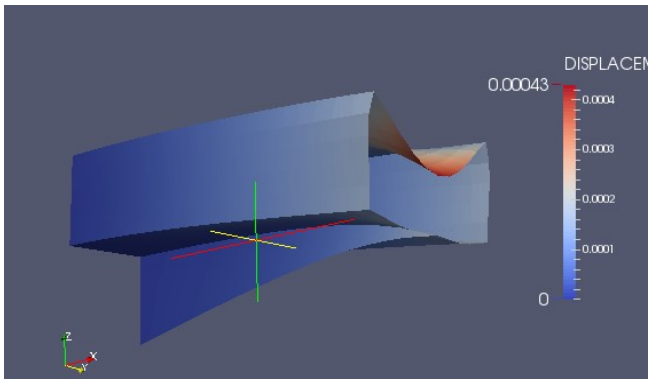
シェルの場合は、線形 solver が「DIRECT」or「MUMPS」しか計算できないので、どちらかを設定するが、ここでは、以下の様に「MUNPS」を設定している。

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。

計算終了後は、「post」画面上で、下図の様に、「eGrp 追加」ボタンをクリックして、DISPLACEMENT から回転を削除しておく。

shell のみ、beam のみモデルの場合、DISPLACEMENT 変数には、変位と回転の値が含まれているので、paraView で変位のベクトルを抽出することが難しいため。

この後、「ParaView 起動」ボタンをクリックして、paraView を起動し、変位を確認した結果が以下になる。

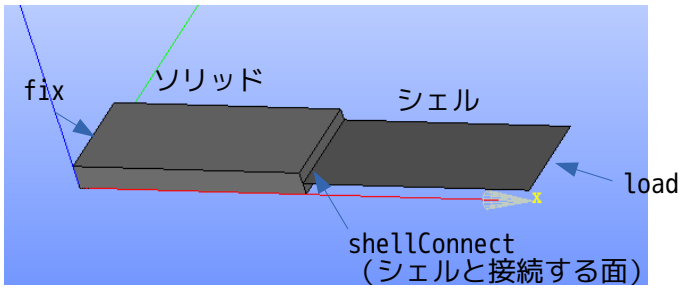


4-13.　ソリッドとシェルの混在モデル解析

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。

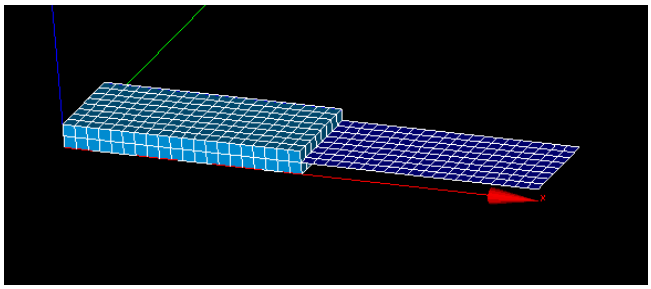
4-13-1.　モデル形状

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
  genarated 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。

式の項数　　式の右辺（定数項）

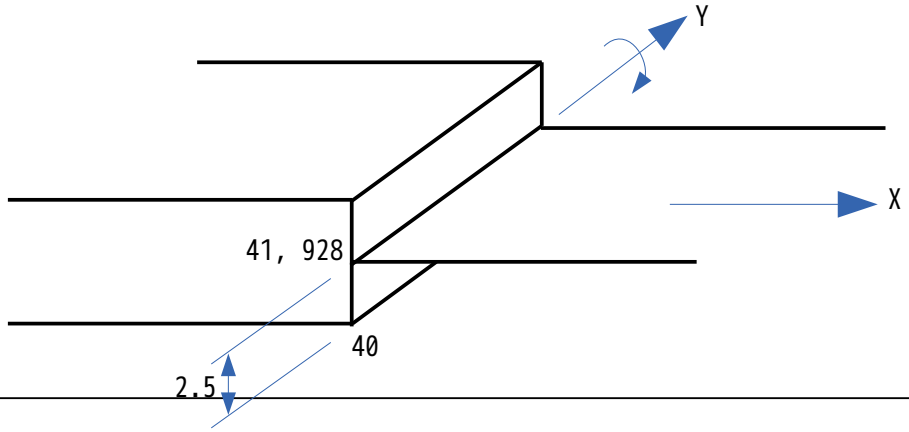
```
4, 0.0
40, 1, 1, 41, 1, -1, 928, 2, 2.5, 928, 3, 0.0
```

第 1 項　第 2 項　第 3 項　第 4 項

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

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

この意味は、以下の様に考えると理解できる。
節点 41 と 928 は同じ座標で 928 は dummy 節点になる。
節点 40 の x 方向変位は、41 の x 方向変位に 928 節点の回転角×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。

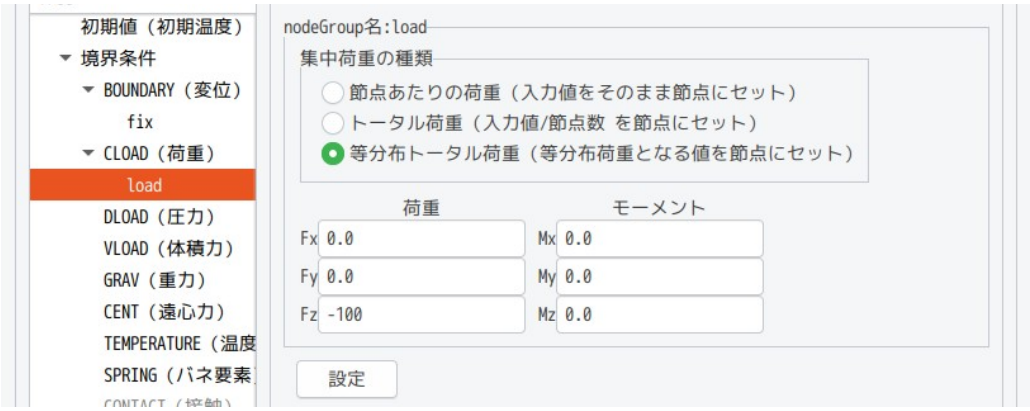
尚、上記のように、メッシュ変換時に!EQUATION を設定しているので、メッシュ変換後に、「スケール変更」をすると、節点座標だけでなく、!EQUATION の係数もスケール変更の対象となる。この為、「スケール変更」を実行すると、!EQUATION 内の係数もスケール変更するかどうかの確認の dialog が表示され、確認の上、スケール変更する様にしている。

今回のモデルは、mm 単位で作成しているので、以下の様に 0.001 倍でスケール変更する。

4-13-3.　材料、境界条件の設定

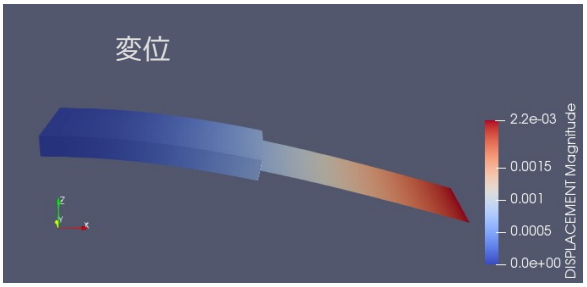
材料は、以下で設定している。

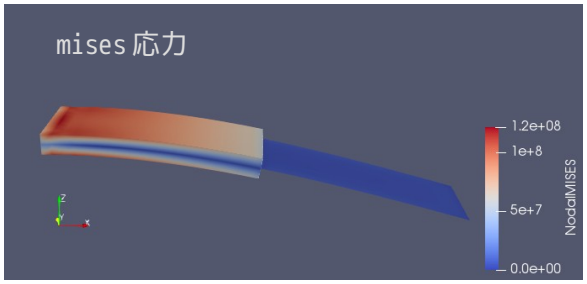
境界条件は、fix を固定し、load の等分布トータル荷重-1000 を入力して計算。



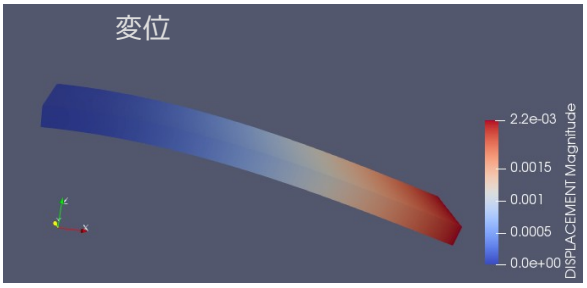
4-13-4.　計算開始、結果の確認

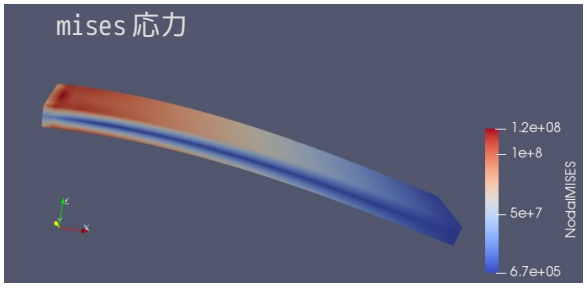
線形 solver を、「MUMPS」または「DIRECT」に設定して、計算を開始させる。
計算結果は、以下になる。





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。





今回、solid、shell の混在モデルを計算したが、シェルについては、前項にも期してあるが、シェルの応力が表示されていない。（22/02/28 現在、FrontISTR-v5.3 においては、シェルの応力が出力できない。）

4-14.　梁（beam）要素の解析

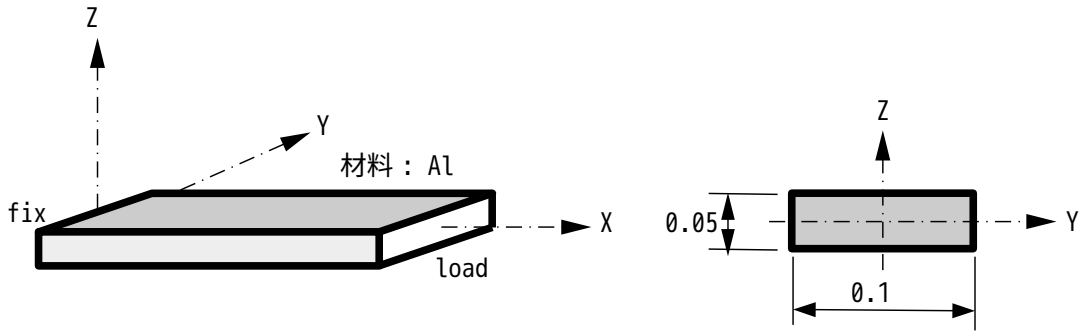
beam 要素（線）のみで構成されたモデルの線形弾性静解析を行ってみる。
solid 解析との違いは、以下になる。
　　1）beam 要素の定数（断面積、各断面 2 次モーメント）を材料物性値と共に設定する。
　　2）BOUNDARY の境界条件は、各軸の回転が追加されている。
　　3）beam の場合、線形 solver は DIRECT or MUMPUS のみ対応。
この内、1)の定数として、beam の参考軸方向と断面 2 次モーメント等の入力が必要であり、複雑な梁構造のモデルについては、かなり煩雑になる。しかし、EasyISTR を使用することで、これらが楽に設定できる。

4-14-1.　梁のたわみ解析

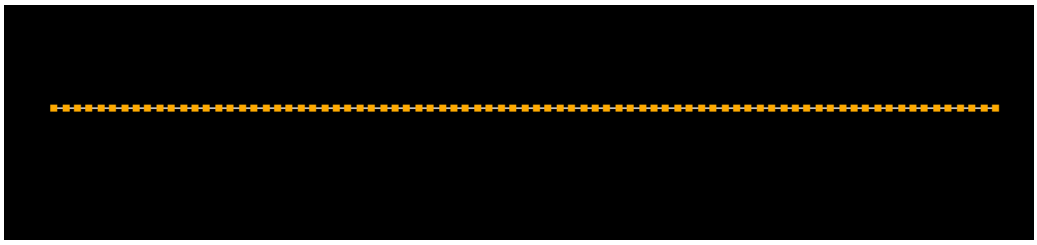
角柱の片持ち梁の先端に荷重を掛け、先端のたわみを計算して、理論解と比較してみる。

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。（原点から X 軸方向に長さ 1m の梁）
下図は、solid の梁だが、解析モデルは、edge で作成する。
解析の条件は、fix を固定し、load の Z 軸方向に-100N を印加した場合の解析を行ってみる。



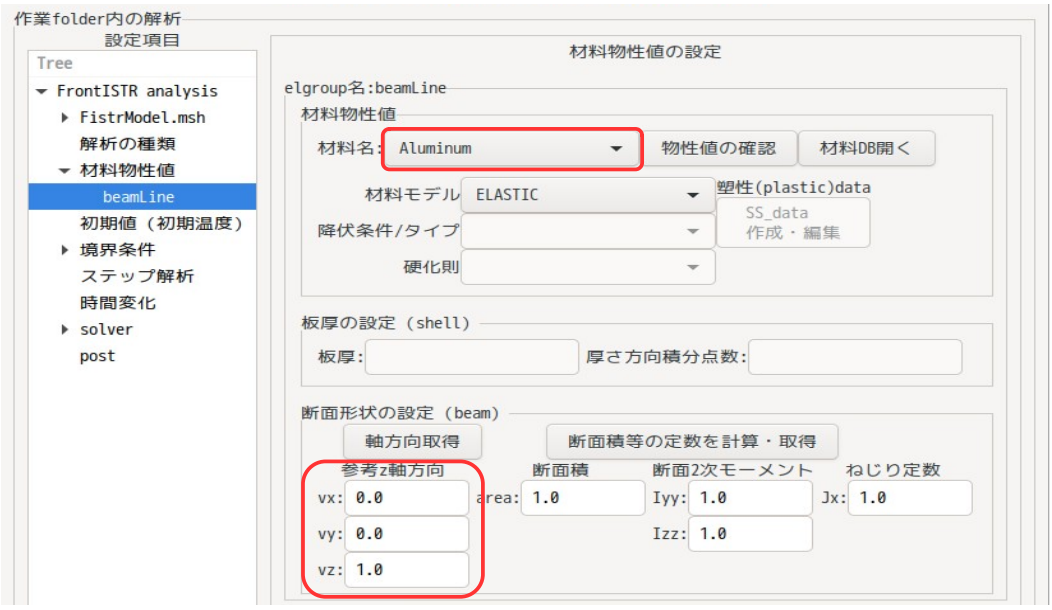
この線分（edge）を salome 上でメッシュサイズ：0.0125 で 1 次要素としてメッシュを切った。（要素数：80 ヶ）このメッシュが beamLine.unv として保存されている。



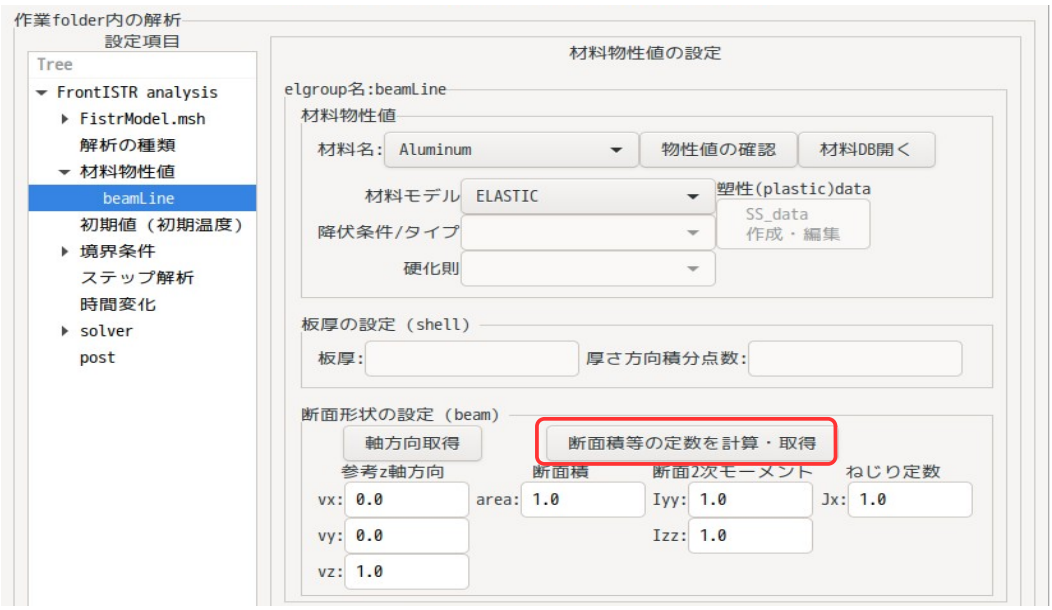
保存した beamLine.unv を EasyISTR 上でメッシュ変換する。

4-14-1-2.　材料の設定

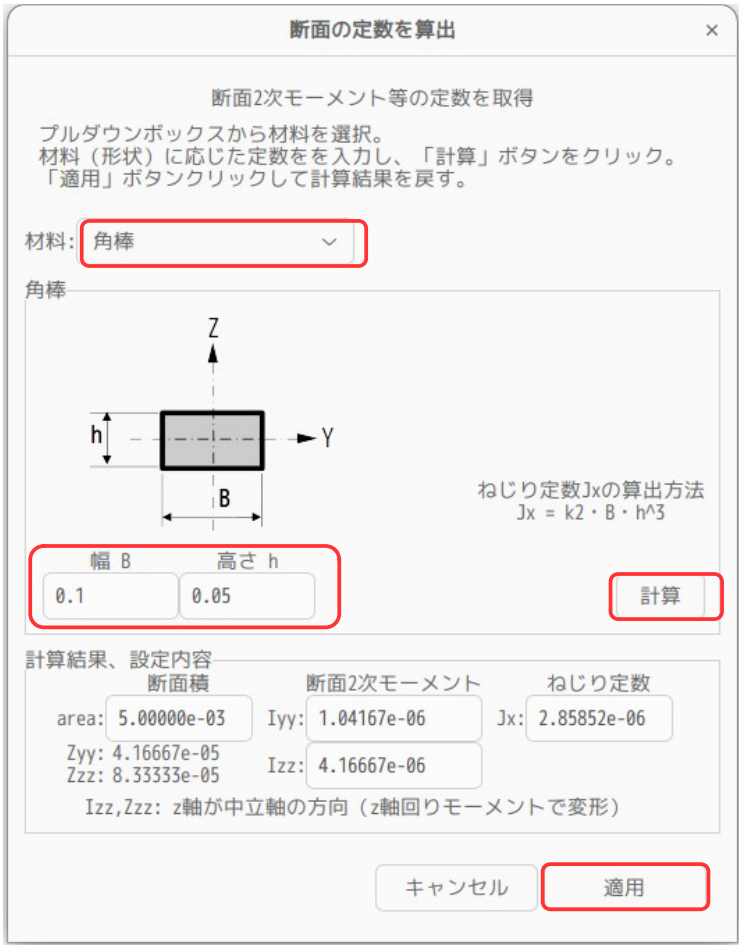
このモデルは、X 軸上で作成されている。この beam 要素の参考 Z 軸方向（local 座標の Z 軸）を求める必要がある。
beam 要素の方向が、local 座標の X 軸方向の為、これに直行する local 座標の YZ 軸を定義する。
今回の場合、beam の方向が global 座標の X 軸と一致しているので、local 座標の Z 軸は global 座標の Z 軸とする事ができる。この為、以下の様に参考 Z 軸方向を「0.0, 0.0, 1.0」を入力する。
（この参考 Z 軸方向は、local の Y 軸方向を決める為のベクトルになる。beam 方向とこのベクトルの外積から Y 軸方向が決まる為、必ずしも Z 軸方向でなくても良い。XZ 平面上のベクトルなら問題ない。）



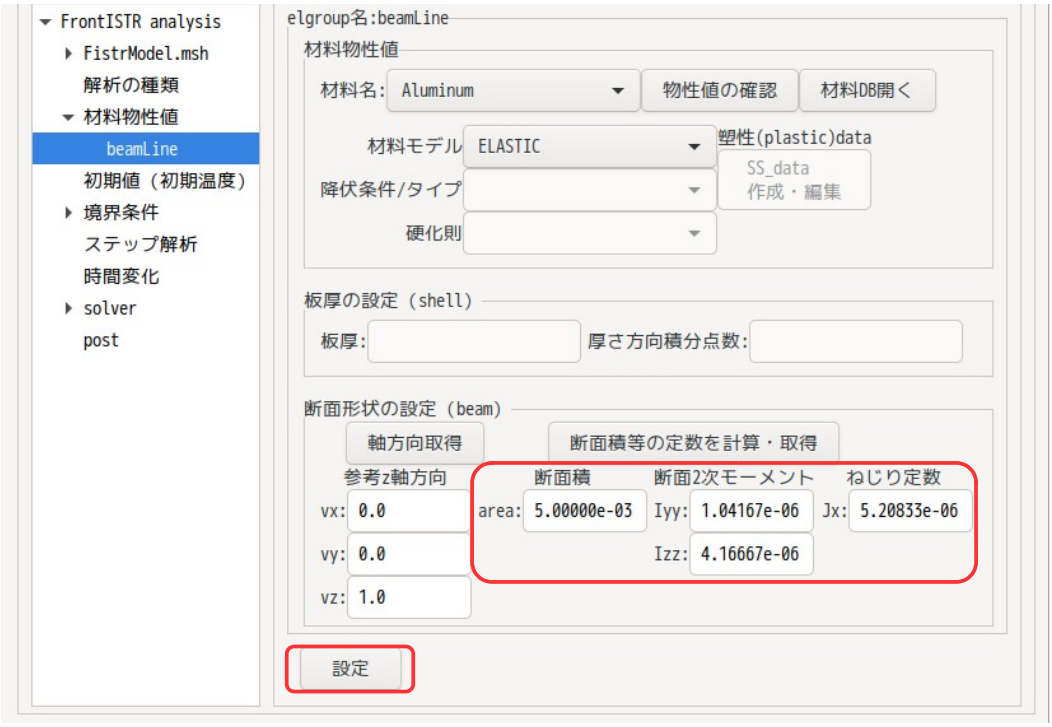
断面積、断面 2 次モーメントは、「断面積等の定数を計算・取得」ボタンをクリックして設定する。



現れた dialog 上で、まず、プルダウンボックスを「角棒」に設定する。
この後、幅、高さを以下の様に入力して、「計算」ボタンをクリックする。
これにより、計算結果が、表示されるので、確認後、「適用」ボタンをクリックして、dialog を閉じる。



これにより、以下の様に、計算結果が取得できるので、「設定」ボタンをクリックして、確定させる。



断面２次モーメントなどの取得は、断面形状が「丸、パイプ、角、中空角、H側、I型、L型、C型、U型」を準備しているので、これらの形状であれば、容易にデータを取得できる。

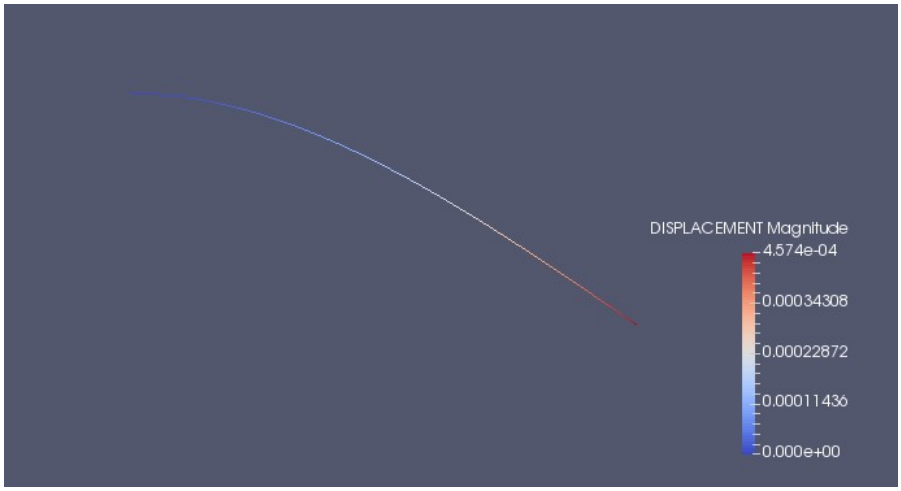
4-14-1-3.　境界条件の設定

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。

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。

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 方向に 100N:　$\delta = \frac{P l^3}{3EI} = \frac{100 \times 1.0^3}{3 \times 70e9 \times 1.041e\text{-}6} = 4.574e\text{-}4$　m

E = 70e9（解析で使用した値）

Y 方向に 100N:　$\delta = \frac{P l^3}{3EI} = \frac{100 \times 1.0^3}{3 \times 70e9 \times 4.166e\text{-}6} = 1.143e\text{-}4$　m

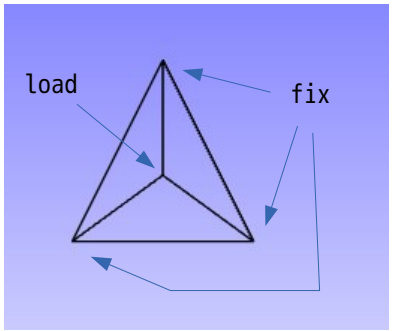
となり、各方向とも、一致している。

4-14-2.　三角すい構造の解析（beam の参考軸方向の取得方法）

beam 要素の解析は、参考 z 軸方向を取得する必要がある。モデル形状が単純に XYZ 軸に並行な状態であれば、容易に取得できるが、今回のように beam が三角すい構造の場合、斜辺の参考軸方向の取得が難しくなる。
しかし、今回の方法を使う事で、楽に取得できる。

4-14-2-1.　モデル形状

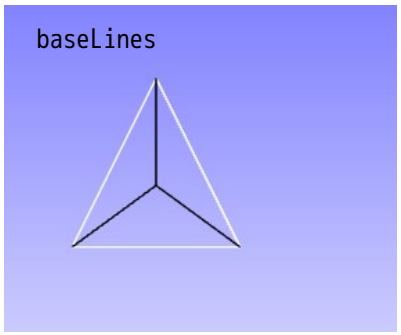
モデルは、以下の様な三角すい構造をしている。
解析は、底面の頂点（fix）を固定し、三角すい頂点（load）を下方に 1000N で押し下げる解析を行う。
beam 材料は、全て Al で外径が φ0.05m の丸棒として計算する。
このモデルのメッシュファイルは、「beamTetra.unv」として保存されている。

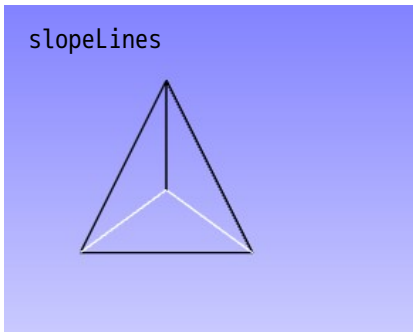


材料名： Aluminum
断面形状：外径 φ0.05m の丸棒

4-14-2-2.　参考 z 軸方向の取得方法

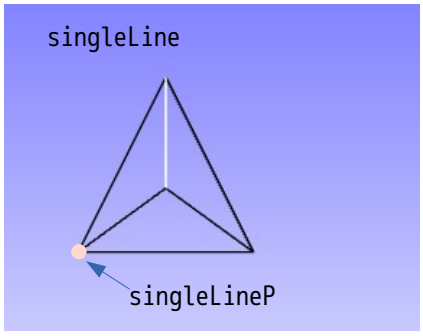
同じ平面上に配置される beam であれば、それらの beam の参考 z 軸（local 座標の z 軸）は、その平面に対する垂直方向として設定できる。
この為、今回の三角すい構造の底面の 3 本の beam は、まとめて 1 ヶの要素 group「baseLines」として設定している。また、斜辺の 2 本もまとめて「slopeLines」として設定している。





これらの要素 group は、group 単独で面が取得できるので、参考 z 軸方向（local 座標の z 軸）は、面に垂直方向として取得できる。

最後の 1 本の beam（singeLine）は、単独では参考 z 軸方向が取得できないが、その beam と他の 1 点（singleLineP）を指定する事で、面が取得できる。



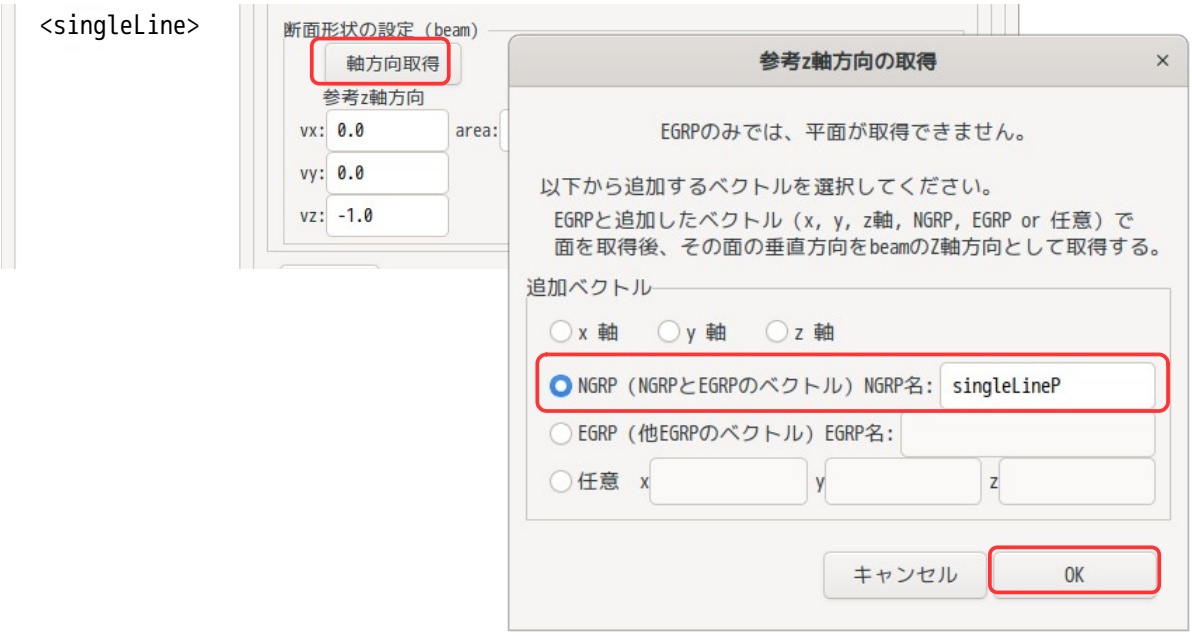
line（singleLine）と point（singleLineP）のセットで面が取得できるので、参考 z 軸方向がその面の垂直方向で取得できる。

local 座標の Y 軸、Z 軸を正しく求める為には、前記した方法になるが、今回の場合、材料が丸棒の軸対象の材料の為、local 座標の YZ 軸がどの方向でも問題ない。この為、参考 Z 軸方向は、全て global の Z 軸（0.0, 0.0, 1.0）に設定しても、問題ない。

4-14-2-3.　材料設定

前項の内容を踏まえて、材料を設定してみる。
baseLines と slopeLines は、これら単独で面が取得できるので、以下の様に「軸方向取得」ボタンをクリックする事で、参考軸方向が取得できる。

singleLine については、節点 Group「singleLineP」のセットで面を取得するので、まず、「軸方向取得」ボタンをクリックする。singleLine 単独では面が取得できないので、以下の dialog が表示される。
この画面上で、以下の様に NGRP 名を入力する事で、参考 z 軸方向が取得できる。



最終的に以下の様に取得できる。

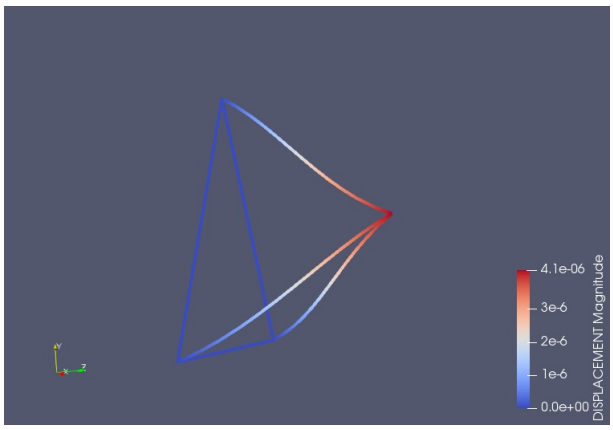
また、材料名 Aluminum の設定と、断面 2 次モーメントも「断面積等の定数を計算・取得」ボタンをクリックして設定している。

4-14-2-4.　境界条件

境界条件は、fix を固定して、load の Z 方向に「-1000N」を印加する設定。

4-14-2-5.　計算開始、結果の確認

全ての設定が終了したので、計算開始させる。以下がその結果になる。

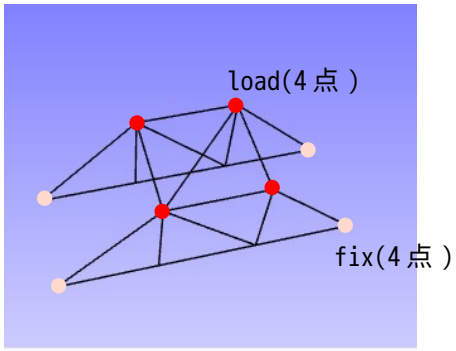


4-14-3.　梁構造（ラーメン構造）解析

梁の方向、梁の断面の定数の取得が簡単にできるので、梁の構造解析を行ってみる。

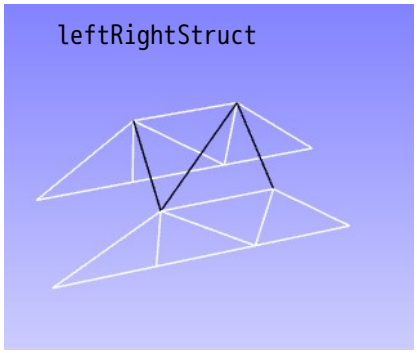
4-14-3-1.　モデル形状、メッシュ変換

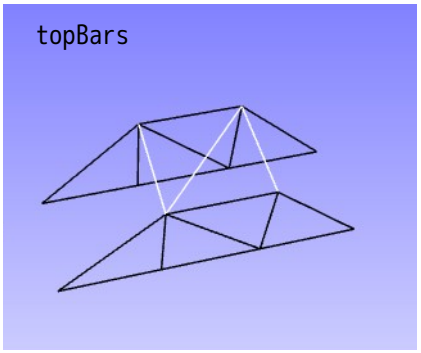
モデルは、以下様に橋を模した構造としている。
梁の材料は、steel とし、断面形状は、等辺 L 型アングル（30 x 30 x t3 mm）としている。



fix を固定し、 load に 1.0e6N/ 点の荷重を掛けて計算。

要素のグループ化は、参考 z 軸方向が取得できる様にグループ化している。





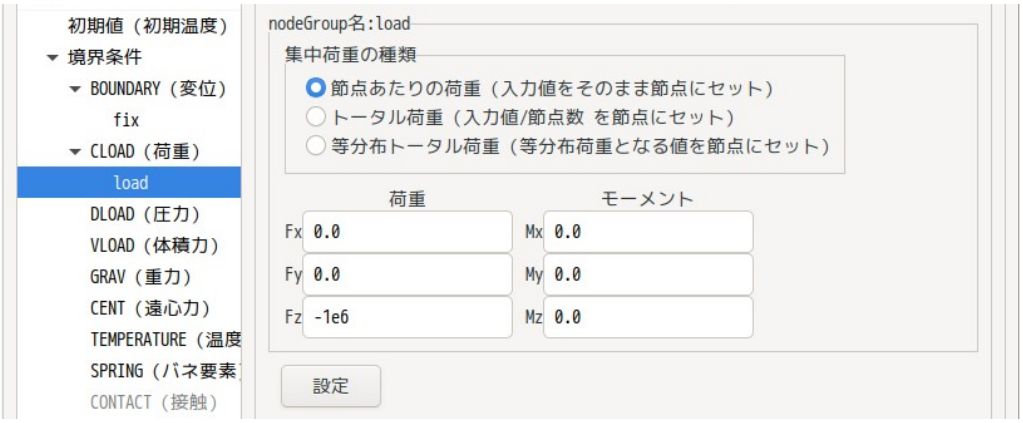
このメッシュが「beamStruct.unv」として保存されているので、このメッシュを変換して使用する。

4-14-3-2.　材料の設定

材料は、参考 z 軸方向が取得しやすい様に group 化しているので、容易に取得できる。
断面 2 次モーメント等も全ての梁の断面形状が等辺 L 型アングル（30 x 30 x t3 mm）なので、容易に取得できる。
以下が設定した結果になる。

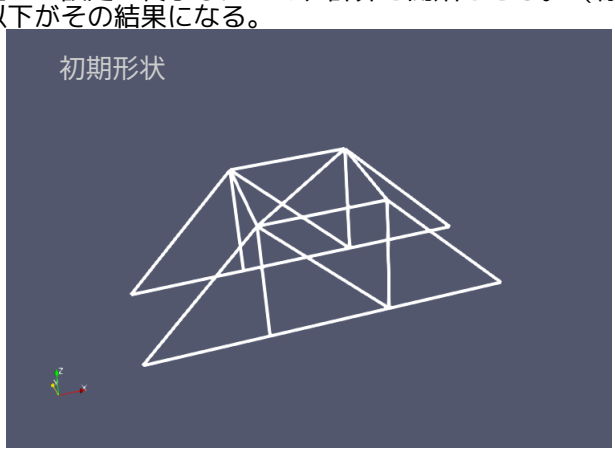
4-14-3-3.　境界条件の設定

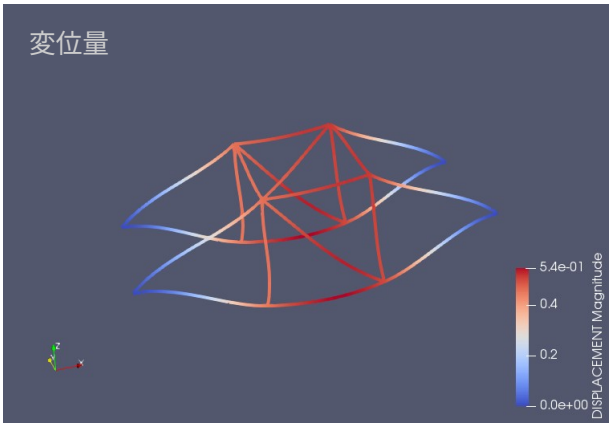
境界条件は、fix を固定し、load に荷重（-1.0e6N/点）を印加する。



4-14-3-4.　計算開始、結果の確認

全ての設定が終了したので、計算を開始させる。（線形 solver は、MUMPUS で計算させる）
以下がその結果になる。



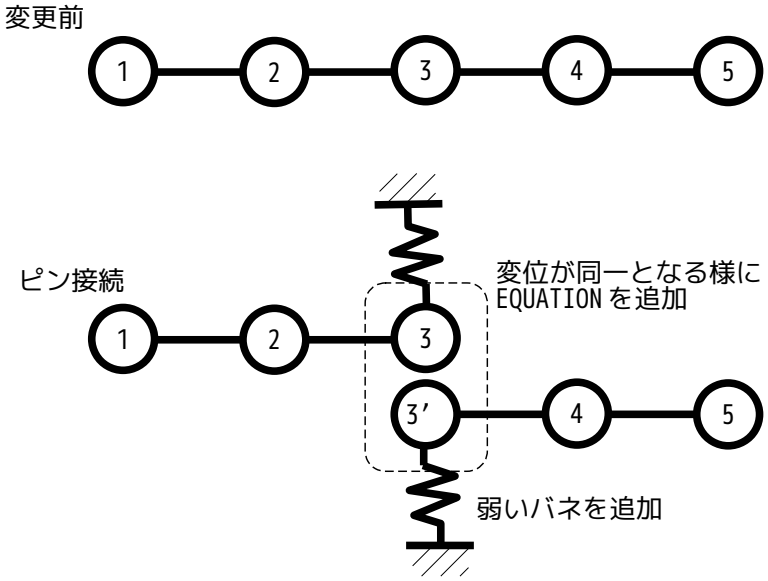


4-14-4.　梁構造（トラス構造）解析

通常通り CAD で edge のモデルを作成し、メッシュをきると、梁同士の接続部は、固定された状態（接続部にモーメントが発生する）になり、ラーメン構造となる。
これに対し、梁の接続部をピン接続（接続部でモーメントが発生しない）としたトラス構造の解析をおこなってみる。

4-14-4-1.　ラーメン構造からトラス構造への変換方法

梁同士の接続部をピン接続とする為には、下図の様に新たな節点 3’ を追加して、接続を断ち、3-3’ の変位が同一となる様に EQUATION を追加する。この状態では、回転の自由度が残っているので、弱いバネを追加して回転を拘束する。



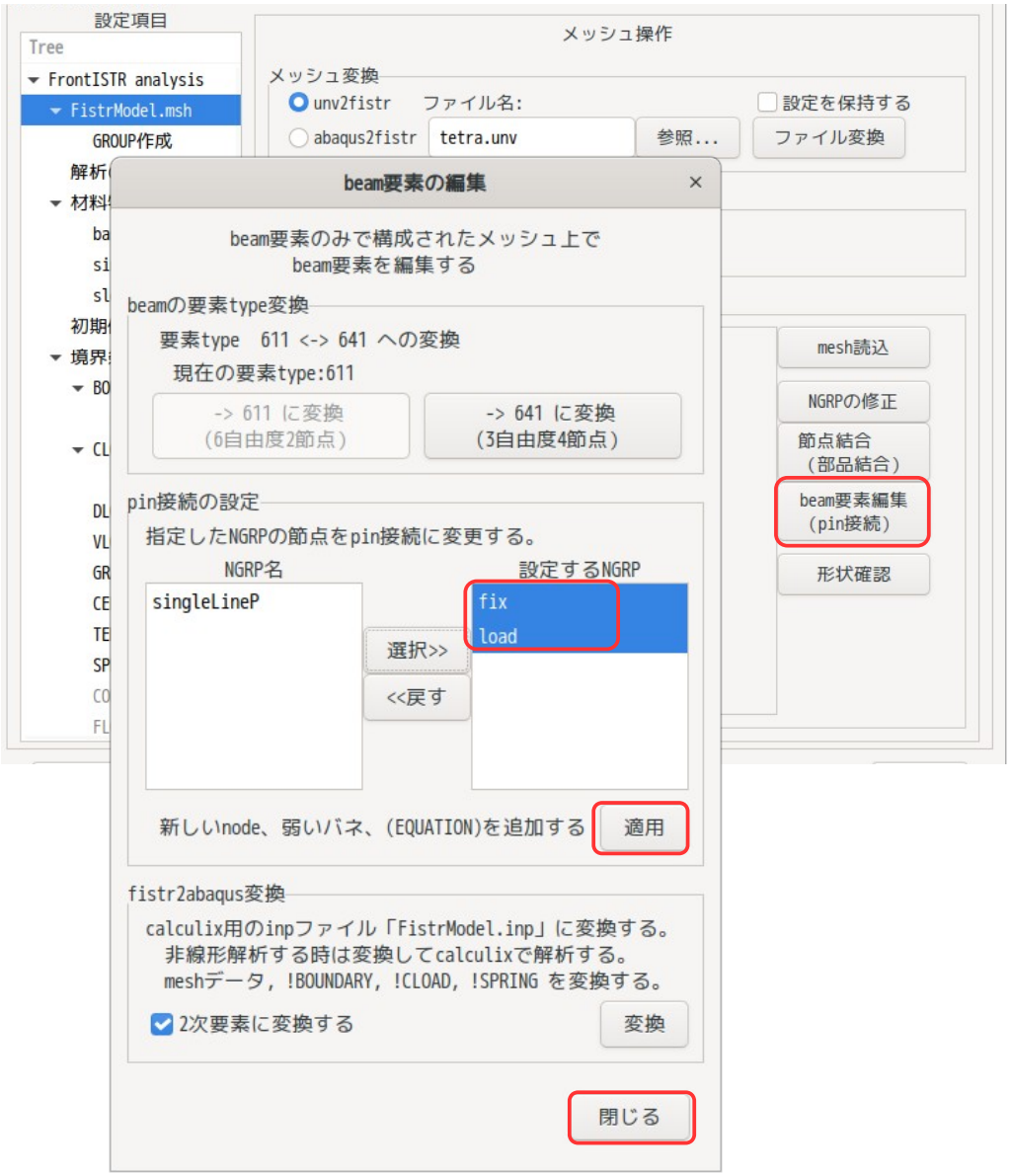
EasyISTR 上では、ピン接続としたい節点 group を指定して、ピン接続が実現できる様にしている。

4-14-4-2.　三角すいのトラス構造解析

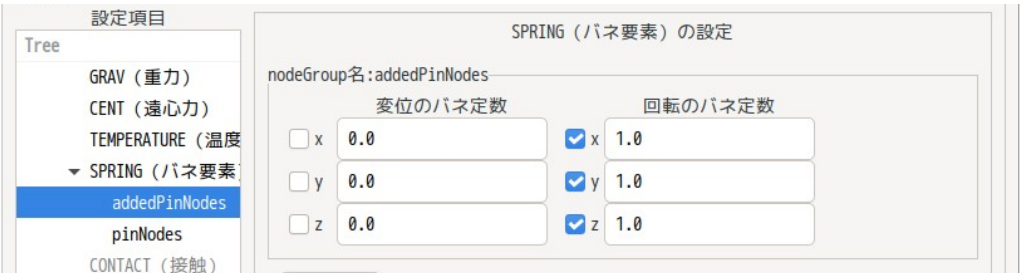
4-14-2 項で解析した三角すい構造の解析をピン接続に変更して解析してみる。
ピン接続とする場所は、三角すいの各頂点（4 箇所:fix, load）として解析する。

まず、4-14-3 項に従って、材料、境界条件を設定し、solver を走らせて、ラーメン構造がうまく走ることを確認しておく。

この後、メッシュ操作画面上で、「beam 要素編集(pin 接続)」ボタンをクリックする。
現れた dialog で、pin 接続とする fix、load を「設定する NGRP」側に移動して、「適用」ボタンをクリックし、「閉じる」ボタンで dialog を閉じる。



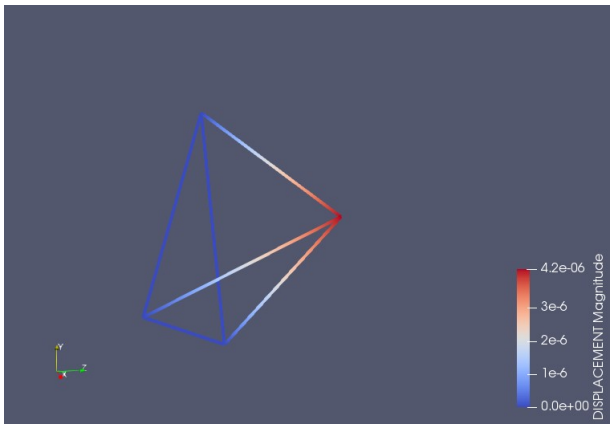
これにより、指定した fix と load の節点がピン接続に変換される。
（新しい節点、!EQUATION と弱いバネが追加されている。）
弱いバネのバネ定数は、デフォルトでは「1.0」で設定されている（下図参照）が、モデルによっては、弱すぎる場合もあるので、計算結果を見ながら調整が必要。
（異常に変位が大きい場合は、バネ定数を大きくする。）



尚、ピン接続を元の剛接続に戻す場合は、「設定する NGRP」側の NGRP 名を全て戻し、「適用」ボタンをクリックする事で、元に戻すことができる。

このままの設定で計算させた結果は、以下になる。ピン接続のため、各梁には、軸力しか働いておらず、梁

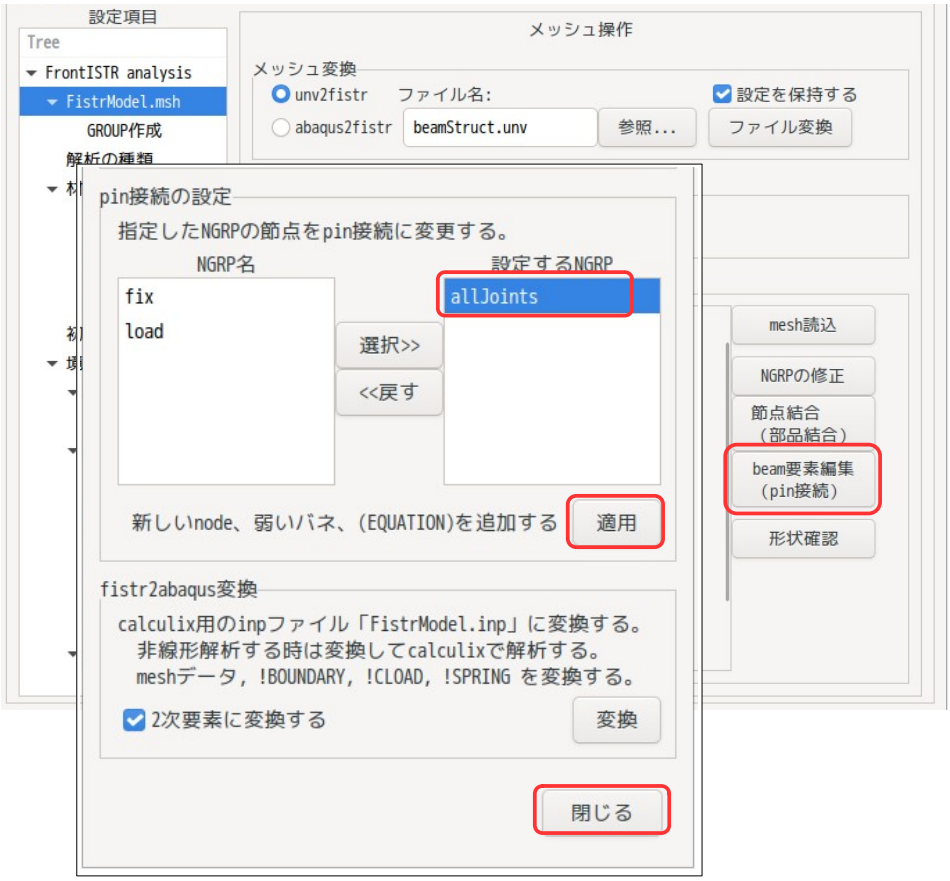
は直線のままで、曲がっていない。



4-14-4-3.　橋を模したモデルのトラス構造解析

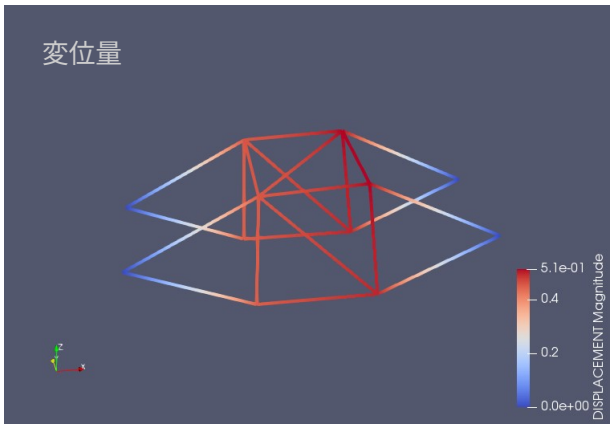
4-14-3 項で解析したモデルをトラス構造として解析してみる。
まず、ラーメン構造として計算できる状態にある事を前提に進める。

ピン接続とする場所は、「allJoints」として設定しているので、この節点 group をピン接続に変換する。
（下図参照）



fix の固定は、変位のみとし、回転の拘束を外しておく。（全ての接合をピン接続としている為）

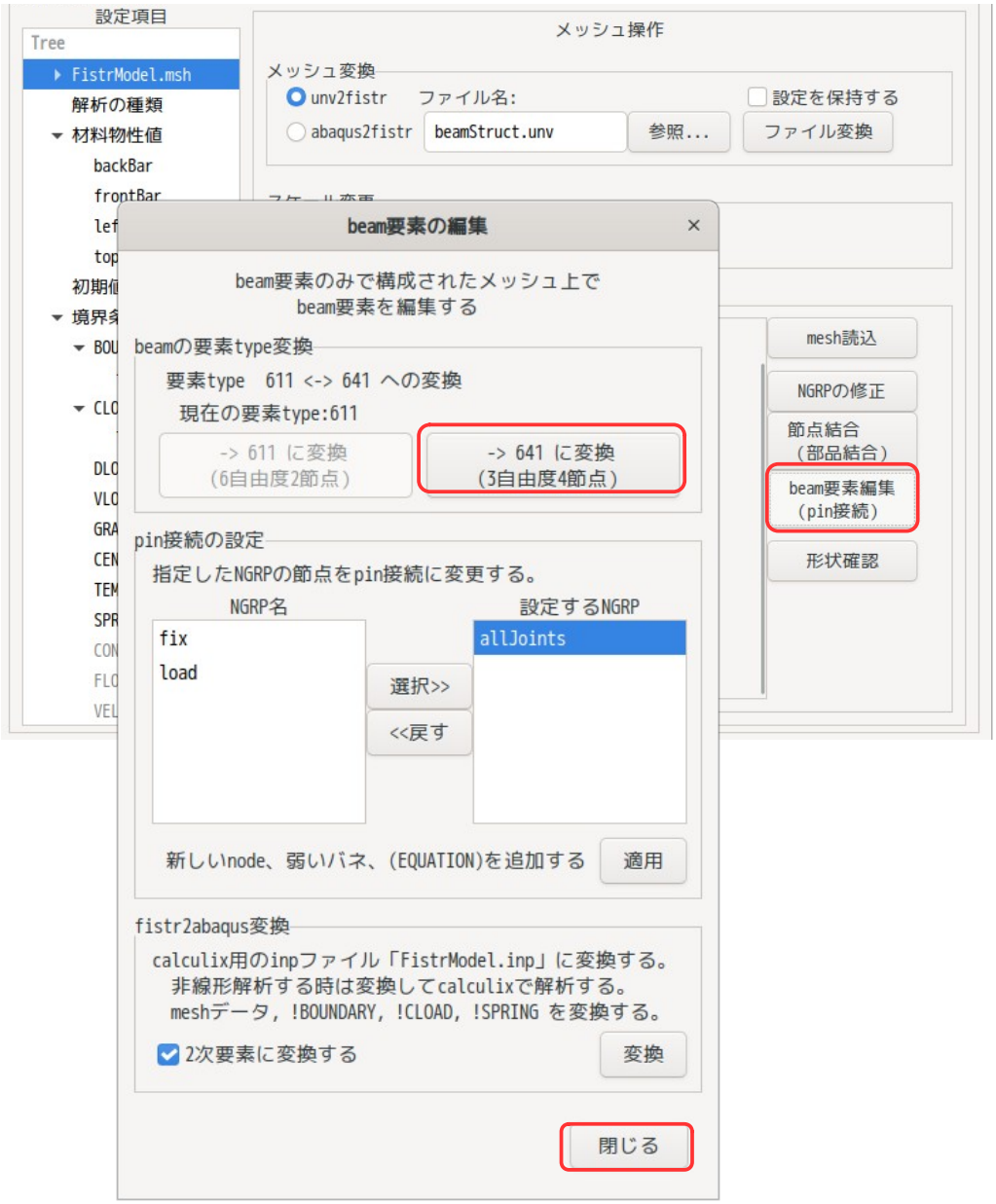
この後、計算して、結果を確認すると、以下になる。
各梁は、ピン接続の為、曲げの変形はしていない事が確認できる。



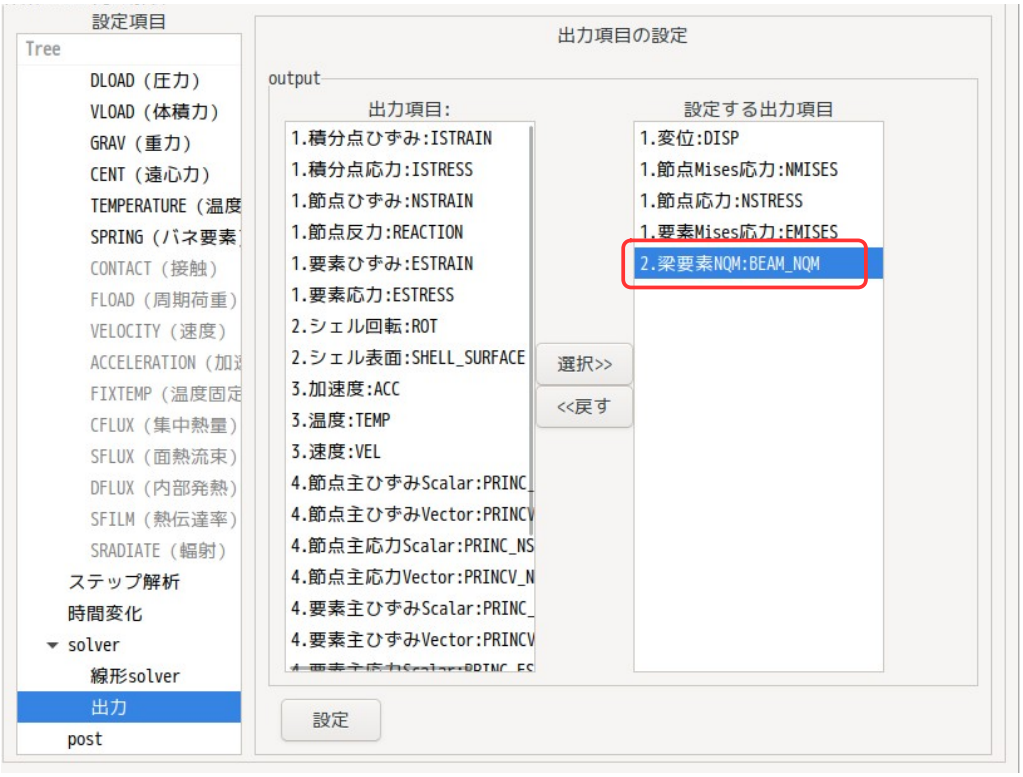
4-14-5.　梁の応力解析

梁には、軸力、せん断力、モーメントが働いている。これらの結果を確認するためには、
　　・beam の要素 type が「641」
　　・BEAM_NQM の出力を ON に設定
する必要がある。
（要素 type「611」では、BEAM_NQM を ON に設定しても、結果が正しく出力されない。）
この設定で、橋を模したモデルのラーメン構造の計算結果を確認してみる。

「beamStruct.unv」をメッシュ変換すると要素 type が「611」になっているが、この状態で、材料設定、境界条件を設定して、計算が走る事を確認する。
この後、要素 type を「611」から「641」に変更する。
その変更方法は、「メッシュ操作」画面上で「beam 要素編集」ボタンをクリックして、dialog を表示させる。この dialog 上で、「-> 641 に変換」ボタンをクリックし「641」に変換後「閉じる」で dialog を閉じておく。これにより要素 type が「641」に変換された事になる。



要素 type 変換後、下図の様に出力項目「2.梁要素:BEAM_NQM」を追加する。



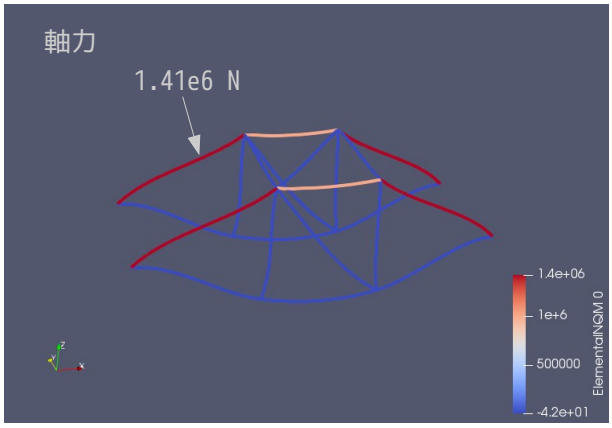
この後、solver を起動して、計算開始させる。
計算後、paraView で「ElementalNQM」を読み込む。
ElementalNQM は、全 12 項目の値が存在しており、各々項目は、以下で設定されている。

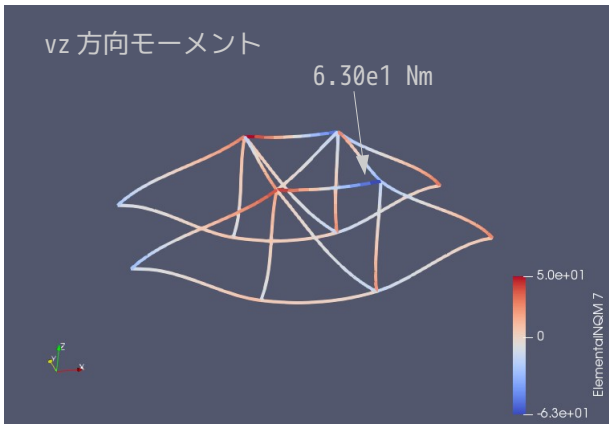
No	内容	備考
0	:軸力	beam 要素方向　1→2 方向　1 の値
1	:vy 方向せん断力	
2	:vz 方向せん断力	
3	:軸力	beam 要素の反対方向　2→1 方向　2 の値
4	:vy 方向せん断力	（0～2 の符号反転）
5	:vz 方向せん断力	
6	:軸ねじりモーメント	beam 要素方向　1→2 方向　1 の値
7	:vy 軸回り曲げモーメント	
8	:vz 軸回り曲げモーメント	
9	:軸ねじりモーメント	beam 要素の反対方向　2→1 方向　2 の値
10	:vy 軸回り曲げモーメント	（6～8 の符号反転）
11	:vz 軸回り曲げモーメント	

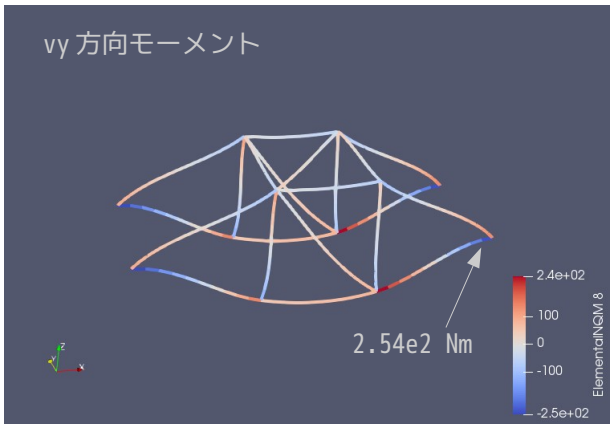
となる。この為、結果を確認するのであれば、

0　軸力
1　vy 方向せん断力
2　vz 方向せん断力
6　ねじりモーメント
7　vy 軸回り曲げモーメント
8　vz 軸回り曲げモーメント

を確認しても、差し支えない様に思う。vy, vz の方向は、beam の local 座標の方向になる。
せん断力、ねじりは小さいので、無視してある。また、モーメントの方向は、変形図からも確認できる。







最終的に以下の結果になる。

最大軸力：	1.41e6 N	1.41e6 / A = 8.25e9 Pa	(A = 1.71e-4)
最大曲げモーメント：	2.48e6 Nm	2.54e6 / Z = 3.85e12 Pa	(Z = 6.6e-7)

応力は、かなり大きい。

また、これら荷重（軸力、YZ せん断力）とモーメント（ねじり、YZ 軸回り曲げモーメント）については、post 画面上の「eGrp 追加」ボタンクリック時に、「NodalBeamAbsForce」「NodalBeamAbsMoment」項目を vtu ファイルに追加しているので、paraView でこれら項目が確認できる。尚、これらの値は絶対値であり、方向（正負）は、変形図で判断する必要がある。

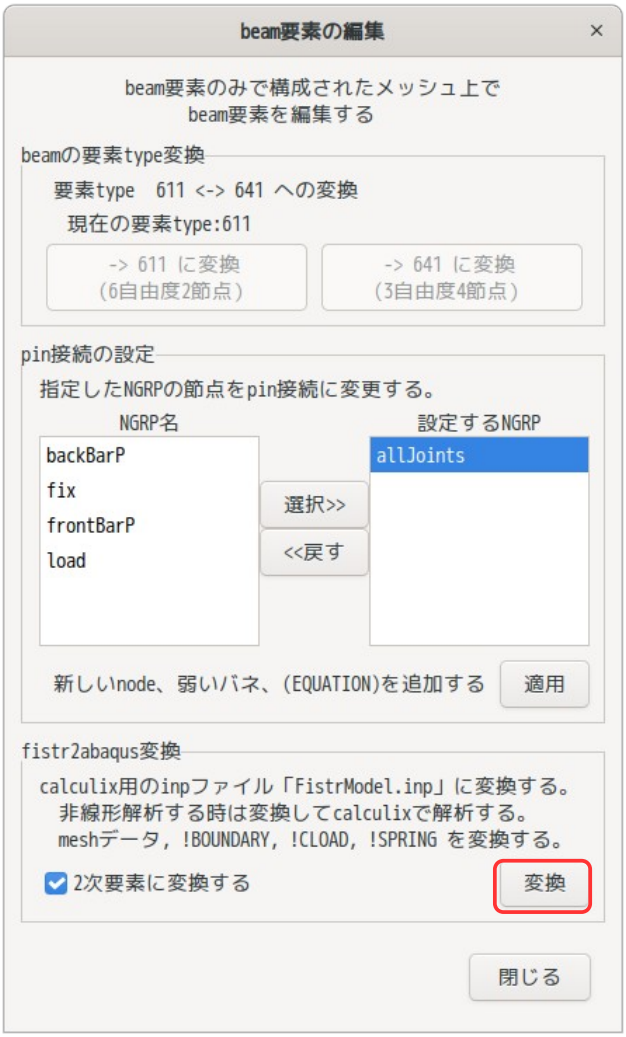
NodalBeamAbsForce
- X: 軸力（絶対値）
- Y: local 座標の Y 軸方向せん断力（絶対値）
- Z: local 座標の Z 軸方向せん断力（絶対値）

NodalBeamAbsMoment
- X: ねじりトルク（絶対値）
- Y: local 座標の Y 軸回り曲げモーメント（絶対値）
- Z: local 座標の Z 軸回り曲げモーメント（絶対値）

4-14-6.　梁の非線形解析（大変形）

FrontISTR では、beam の非線形（大変形）解析ができない。非線形解析を行う為には、calculix を使うとこれが実現できる。

この為、FrontISTR の msh、cnt ファイルを calculix の inp ファイルに変換する方法を準備している。
その方法は、下図の「変換」ボタンをクリックする事で変換できる。



この変換は、今回の梁接続部が剛接続、ピン接続の両方変換でき calculix で計算できる。

また、calculix 側の要因で、断面は、現状、丸棒、角棒、パイプしか対応していない。
また、丸棒、パイプの場合は、2 次要素（B32R）しか対応していないので、「2 次要素に変換する」にチェックを付けて変換する。

変換後は、断面形状を指定する「*BEAM SECTION」、「*BEAM GENERAL SECTION」カードを編集して、断面形状を指定する必要がある。

また、calculix で非線形計算を行い、その結果を vtu ファイルに変換する為に「createAbaBeamVtu.py」を準備しているので、これを使うと vtu に変換でき、paraView で結果を確認することができる。
その変換方法は、端末を起動して以下の様に入力して変換する。

```
$binApp/createAbaBeamVtu.py <headerFile 名>
```

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）

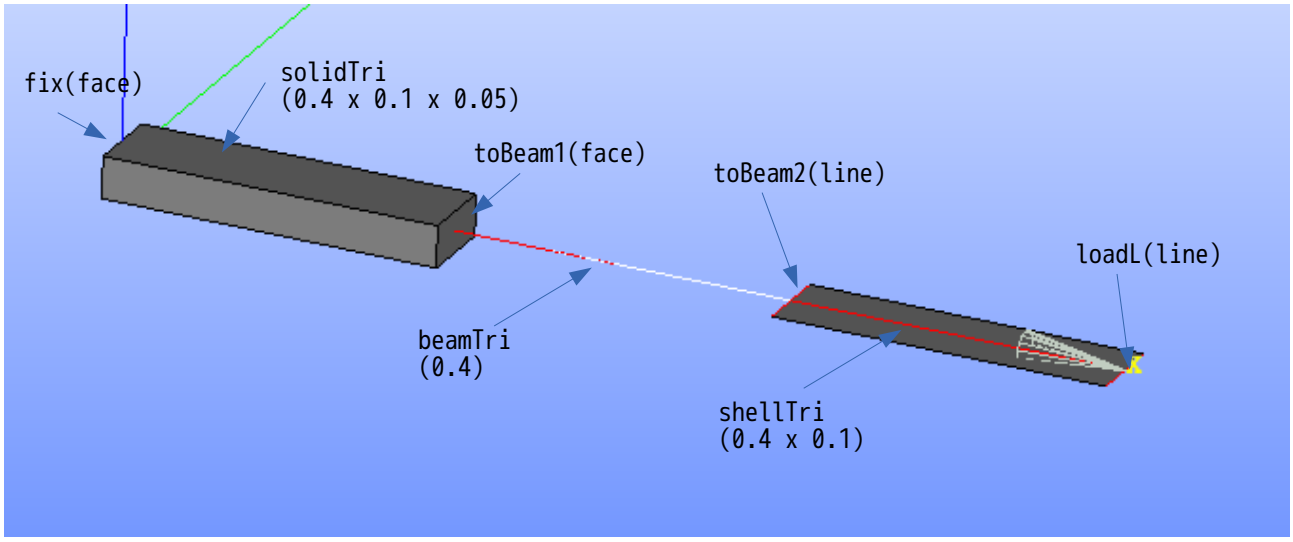
また、計算は、線形弾性静解析に限られ、線形 solver も MUMPU or DIRECT に限られる。

4-15-1.　梁のたわみ解析（ビーム、シェル、ソリッド）

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 x0.05m）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

以下の様にビーム、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***）
```

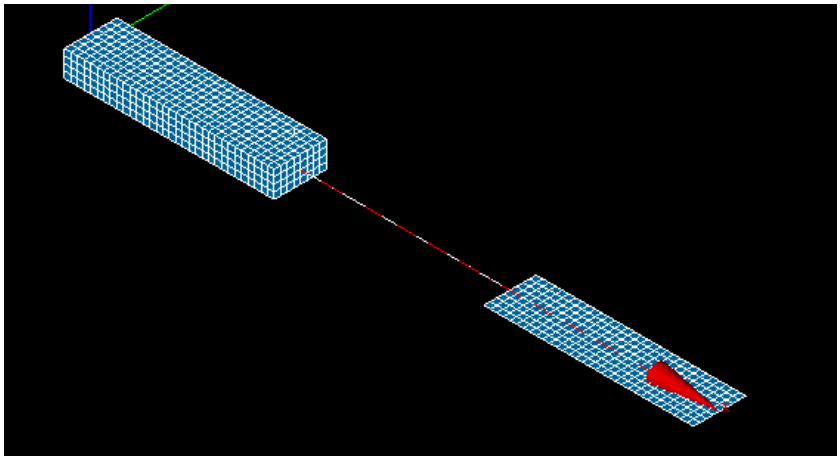
上記の「***」は、任意の文字で設定できる。この設定を行っておくと、メッシュ変換時に、各接続部で回転を伝達させる「!EQUATION」を作成してくれる。上記の名称を設定しなかった場合は、回転が拘束されない解析を行うことになる。（!EQUATION が作成されない。）

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）

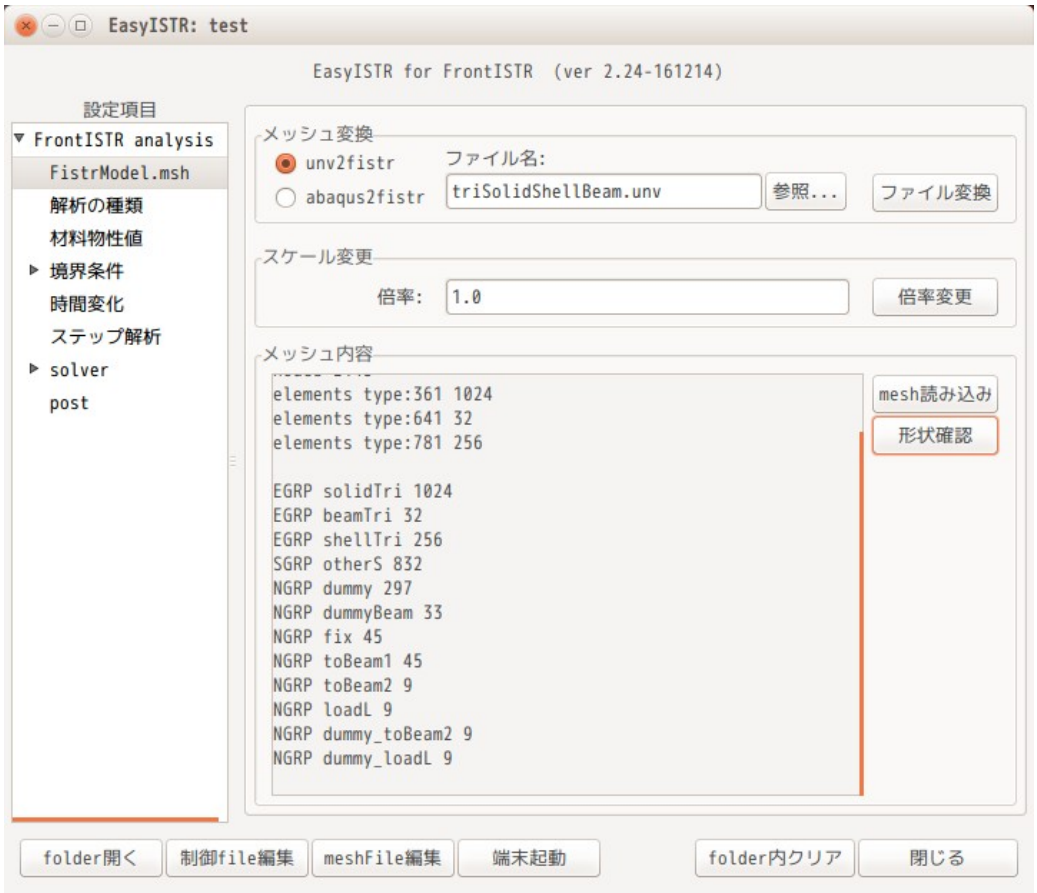
このモデルのメッシュを以下の様に作成。
できあがったメッシュを「triSolidShellBeam.unv」として保存する。

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。

4-15-1-3.　解析の種類

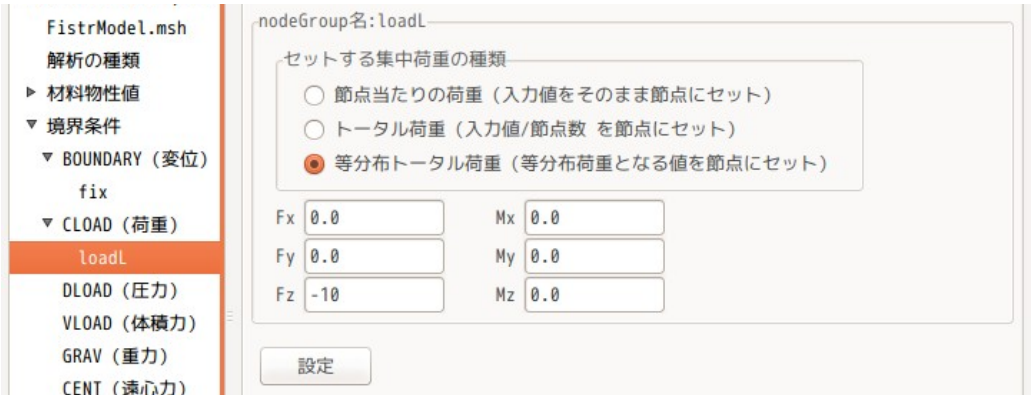
solid、shell、beam 混在モデルの場合は、線形弾性解析に限られるので、「線形弾性静解析」を選択する。

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、4-14-1 項の beam 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。

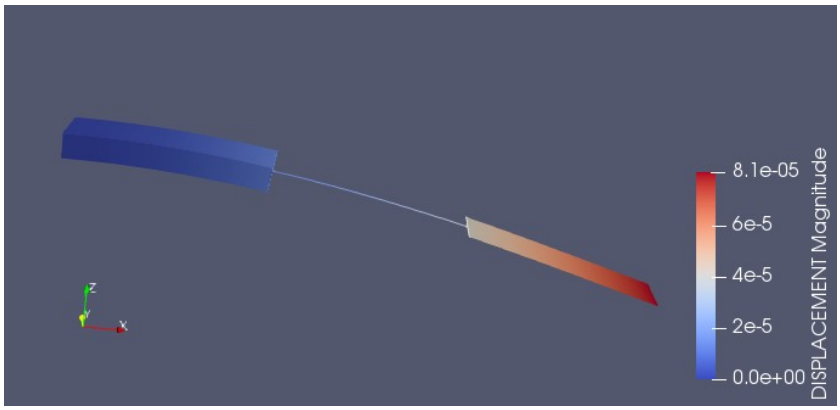
4-15-1-5.　境界条件の設定

境界条件は、fix を固定し、loadL に等分布荷重 10N を設定する。



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、「8.060e-5」となっている。

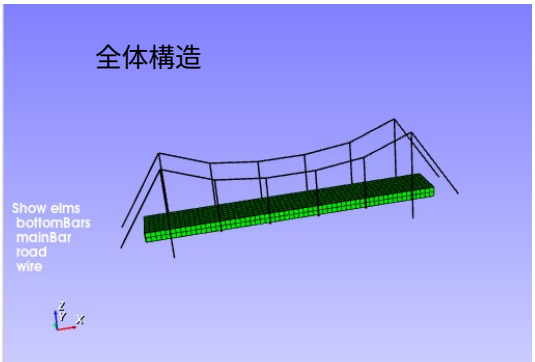
4-15-1-7.　結果の検証

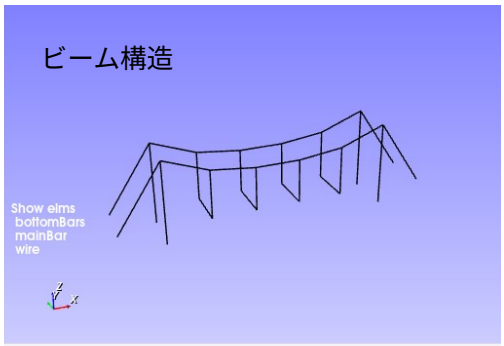
片持ち梁のモーメントによるたわみは、以下で表され、計算すると「7.90e-5」となり、ほぼ一致する。

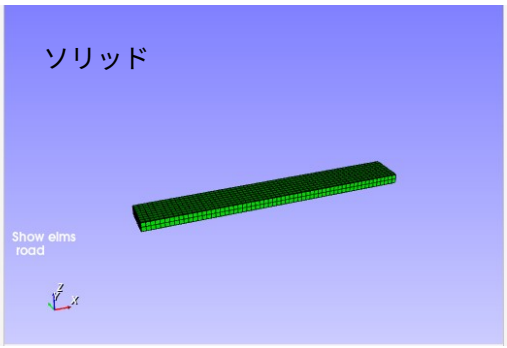
$$\delta=\frac{Pl^3}{3EI}=\frac{10\times1.2^3}{3\times70\mathrm{e}9\times1.041\mathrm{e}\text{-}6}=7.90\mathrm{e}\text{-}5$$

4-15-2.　つり橋の解析（ビーム、ソリッド）

以下の様にビーム構造上にソリッドが乗っている構造（つり橋構造）をしているモデルを解析する。





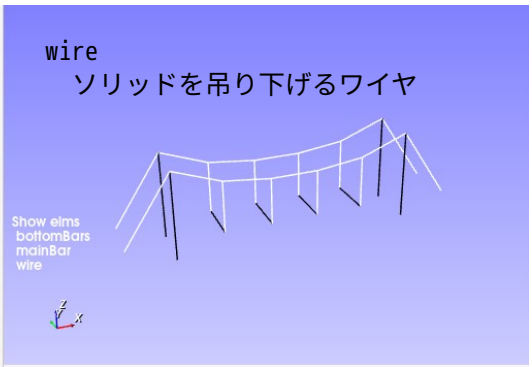


解析方法は、まず、ソリッドとビームを結合する。この後、ビームの固定部（支柱とワイヤ部）とソリッドの両端面の下側 edge を固定し、ソリッドの上面に荷重を掛けて、変位量を確認する。

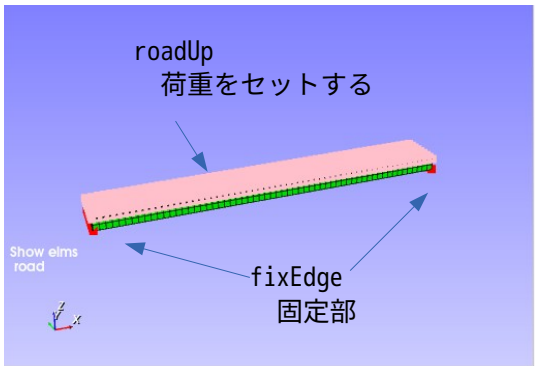
4-15-2-1.　メッシュ読み込み、変換

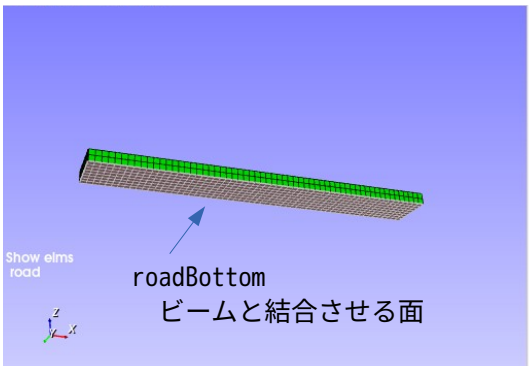
メッシュは、「hangRoadWideFix.unv」として、$easyIstrPath/unvFiles フォルダ内に保存されているので、この file を指定してメッシュ変換する。
このメッシュは、ビーム構造とソリッドに分かれている。（節点は共有していない。）

ビーム構造側の要素 group は、以下がある。（各々の group でビームの断面形状が設定できる）

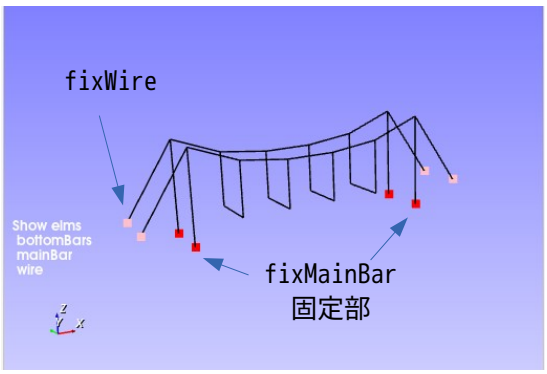


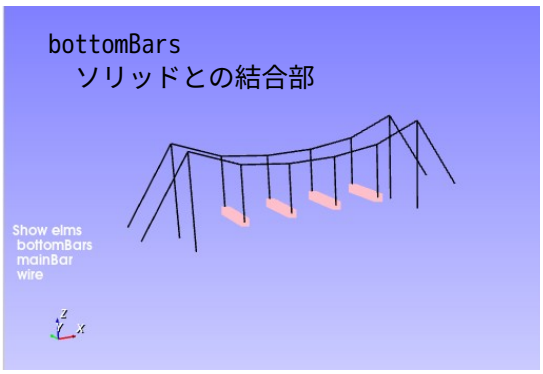
ソリッドは、以下の様に面 group、節点 group を作成している。





ビーム側は、以下で節点 group を作成している。





4-15-2-2.　部品の結合

ソリッドの面 group「roadBottom」とビーム構造の節点 group「bottomBars」を結合する。
この結合は、slave:bottomBars, master:roadBottom として結合する。
（結合の方法は、「3-14-1.　節点を相手 face 上に結合」を参照。）

今回の結合は、ビーム要素の節点 group（ダミー節点を除く）をソリッドの面に結合している為、変位のみが結合されている事になる。（回転は結合されていない。）

4-15-2-3.　材料物性値の設定

材料は、以下で設定する。（各材料や寸法は、適当に設定している。）

材料名	材料	断面	寸法	備考
bottomBars	Steel	I 型鋼	B:0.3, h:0.4, t1:0.04, t2:0.04	beam solid との結合部

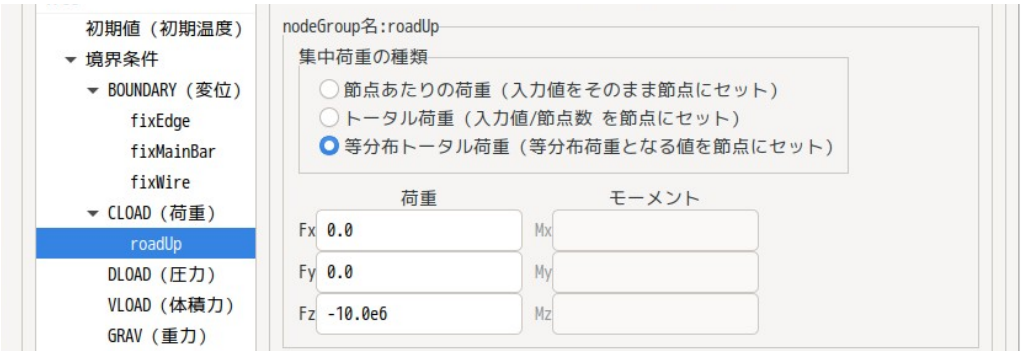
```
mainBar        Steel  パイプ d:0.6, t:0.06                    beam 支柱
wire           Steel  丸棒   d:0.1                           beam 吊り下げ用ワイヤ
road           Al     -      -                               solid
```

ソリッドの材料は、適当な材料が見つからなかったので、Al に設定している。

4-15-2-4.　境界条件の設定

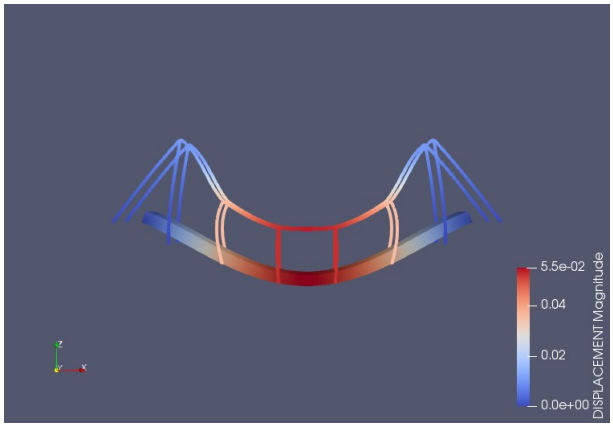
境界条件は、「fixEdge」、「fixMainBar」、「fixWire」を固定する。

ソリッドの上面「roadUp」に荷重「-10e6」を Z 軸方向に印加する。



4-15-2-5.　計算開始、結果の確認

全ての準備が整ったので、計算させる。
solver が「MUMPS」に設定されている事と、出力に「梁要素 NQM:BEAM_NQM」が追加されている事を確認後、計算をスタートさせる。以下が計算結果になる。

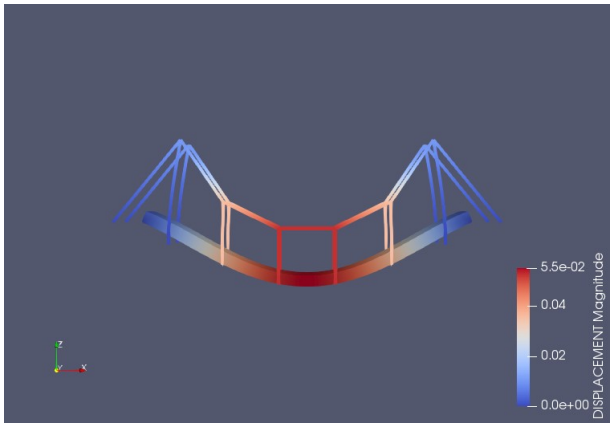


ワイヤが曲がっているので、ワイヤ部のジョイントにモーメントが掛かっている。ビーム同士が剛接続になっている為。

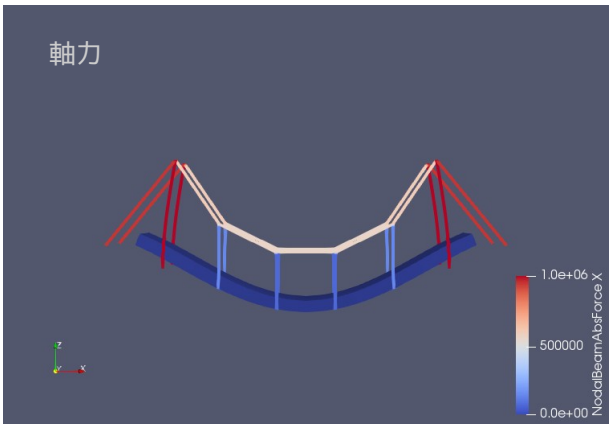
4-15-2-5.　ビーム同士をピン接続に変更して再計算

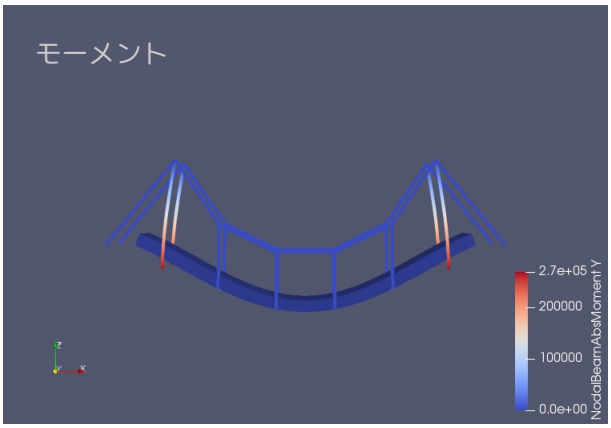
ビーム同士をピン接続に変更するには、「4-14-4. 梁構造（トラス構造）解析」の方法に従って、節点group「allJoints」の節点を pin 接続に変更する。

変更後再計算した結果が以下になる。



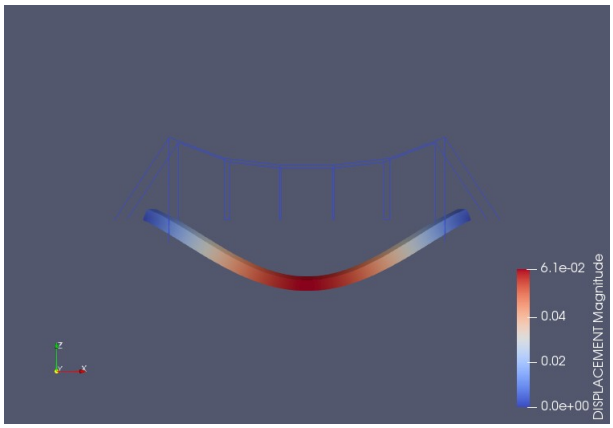
ワイヤには、モーメントが掛かっておらず、うまく計算できている。
この状態で、ビームの軸力とモーメントを確認してみる。
以下がこの結果になる。





軸力は、支柱とワイヤ固定部に大きな軸力が働いている事がわかる。
モーメントは、支柱の根本に働いている。

参考までに、ビームとソリッドの結合を外して計算した結果が以下になる。



ビームの影響が無くなると、変形が大きくなっているので、ビームがソリッドの変形を抑えていた事が確認できる。