

SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)

信頼性課 藤井 08/6/4

SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)
(SalomeMeca 2008.1)

目次

1. 目的
2. 解析方法
 - 2-1. モデルの作成
 - 2-2. Code_Aster の作成
 - 2-3. Code_Aster の修正
 - 2-4. 実行と結果の確認
 - 2-5. アニメーション表示
 - 2-6. アニメーションファイルの作成保存
 - 2-7. 結果の再検討
 - 2-8. 指定した位置の変動
3. 負荷変動させる場合の解析方法
 - 3-1. Code_Aster の修正
 - 3-2. 実行と結果の確認

1. 目的

周波数応答を前章で行ったが、周波数応答では、実際にモデルがどのように運動しているのかがイメージとして捉え難い。ここではモデルの動解析を行って、モデルがどのように運動しているのかを解析する。

2. 解析方法

モデルは、L字モデルを使い、先端に1Nの荷重を掛けた瞬間のモデルの運動を解析する。

2-1. モデルの作成

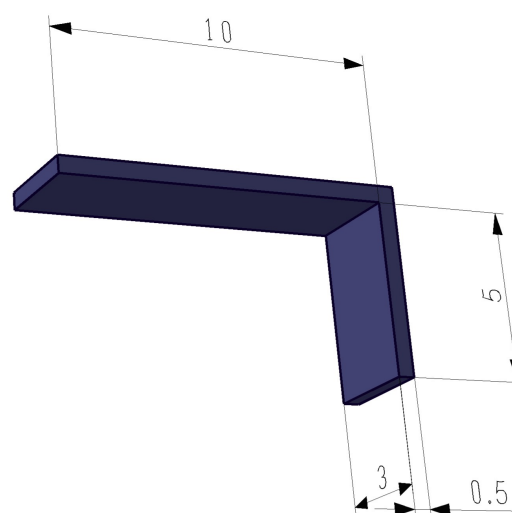
モデル「bar-2.stp」を読み込み、固定部分と荷重を付加する部分を fix、load でグループ化する。

モデルの大きさは、荷重を付加する面の寸法が $3 \times 0.5\text{mm}$ になっている。

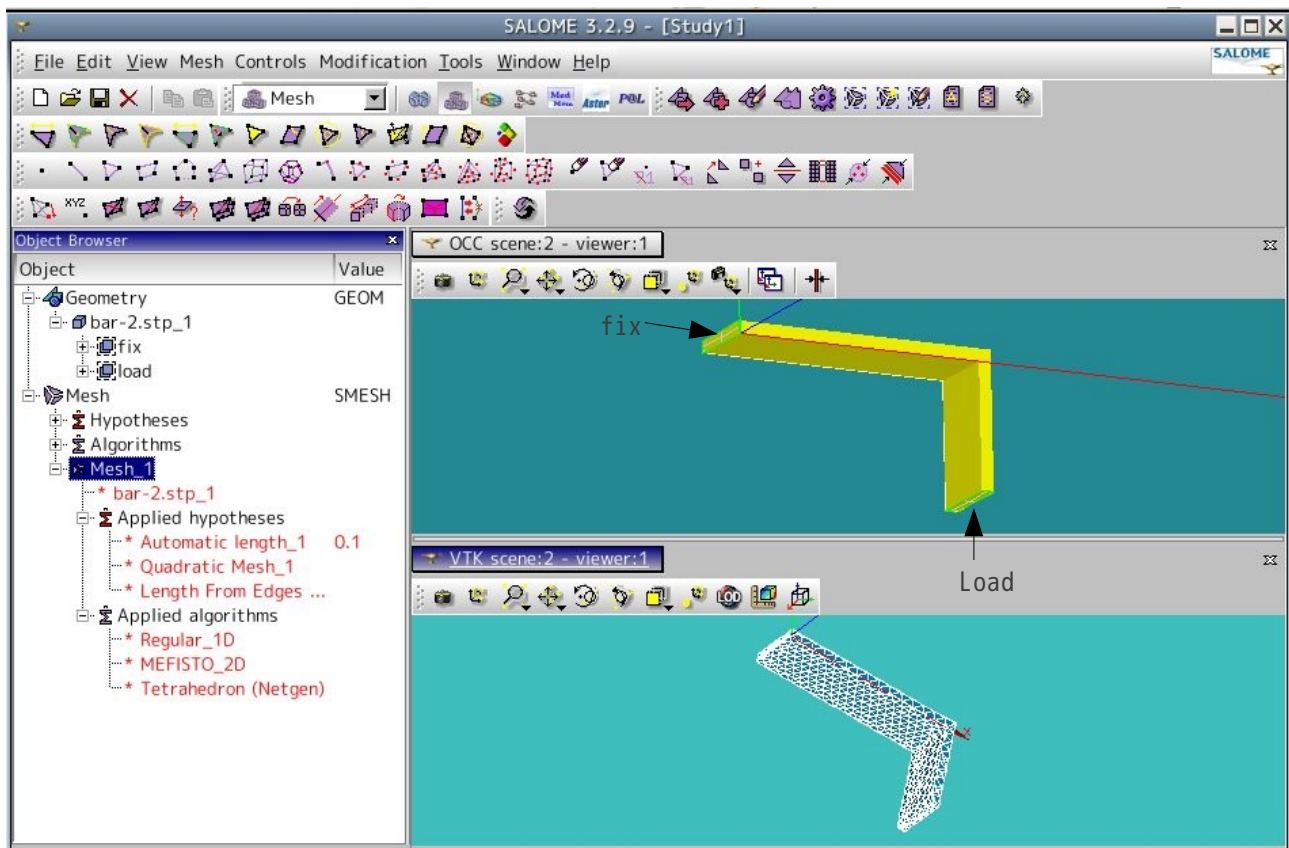
この面に垂直方向に1Nの荷重を掛ける事にする。

メッシュは、三角形の2次メッシュとし、大きさは、Automatic Length 0.1とした。

詳細は、下記。



SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)



2-2. Code_Aster の作成

Salome の Aster 画面からウィザードを使って、Code_Aster を作成する。

材料定数は、下記で設定した。 適当なので注意。

ヤング率	130300 MPa
ポアソン比	0.343
密度	7.8e-9 ton/mm ³

この状態で固有振動数を確認すると下記になる。

1次	1703Hz
2次	4625Hz
3次	6961Hz
4次	14360Hz
5次	20640Hz

2-3. Code_Aster の修正

作成した Code_Aster を修正する。粘性減衰は、いくつかが良いか判らないが、0.0001 に設定した。材料の定義は下記で設定。

DEFI_MATERIAU	ACIER
ELAS	
E	130300
NU	0.343

SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)

```
RHO          7.8e-9
AMOR_ALPHA   0.0001          #α粘性減衰を定義
```

境界条件を設定する。

荷重の設定は、付加する load 面の面積が $3 \times 0.5 = 1.5\text{mm}^2$ なので、1N を掛ける為の単位面積当たりの荷重は、 $1\text{N}/1.5\text{mm}^2 = 0.667\text{N}/\text{mm}^2$ になるので、0.667 をセットする。

```
AFFE_CHAR_MECA      BLOCAGE
MODELE              MODELE
DDL_IMPO
  GROUP_MA          fix
  DX                0.0
  DY                0.0
  DZ                0.0
FORCE_FACE          #この部分を追加
  GROUP_MA          load
  FY                0.667      #load面に計1Nに設定
```

次に減衰マトリックスを定義する

```
MACRO_MATR_ASSE
MODELE              MODELE
CHAM_MATER          CHMAT
CHARGE              BLOCAGE
NUME_DDL            NUMDDL
MATR_ASSE
  MATR_ASSE_1
  MATR_ASSE_2
  MATR_ASSE_3      #これ以下を追加
  MATRICE           amore
  OPTION            AMOR_MECA
```

次に動解析する時の計算時間間隔を定義する。

固有振動数が約 1700Hz の為、100 μs 以下 (固有振動数の 1/4 以下) での計算が必要。従って、計算間隔は、0.0001 とする。(計算してみて修正する。)

```
DEFI_LIST_REEL      stepTR
DEBUT                0.0001      #計算開始時間
INTERVALLE
  JUSQU_A            0.001      #計算終了時間
  PAS                 0.0001     #計算間隔 0.0001~0.001 まで0.0001 毎に計算
```

次に solver の設定。

```
DYNA_LINE_TRAN      dynaTR
MODELE              MODELE      #モデルを設定
CHAM_MATER          CHMAT       #材料を設定
MATR_MASS           MASSE       #質量マトリックスを設定
MATR_RIGI           RIGIDITE    #剛性マトリックスを設定
MATR_AMOR           amore       #減衰マトリックスを設定
```

SalomeMeca の使いかた -- 12.0 動解析 (過渡解析)

```

NEWMARK                #NEWMARK 法で計算
  DELTA                0.5
EXCIT
  CHARGE                BLOCAGE          #境界条件を設定
INCREMENT
  LIST_INST            stepTR           #計算間隔を設定
  b_list_fonc

```

次に出力項目を設定する。

```

IMPR_RESU
  MODELE                MODELE
  FORMAT                MED
  b_format_med
  RESU
    RESULTAT            dynaTR
    b_info_med
    b_sensibilite
    b_partie
    b_extrac
    NOM_CHAM            DEPL
    b_cmp
    NOM_CHAM            (DX,DY,DZ)
    b_topologie

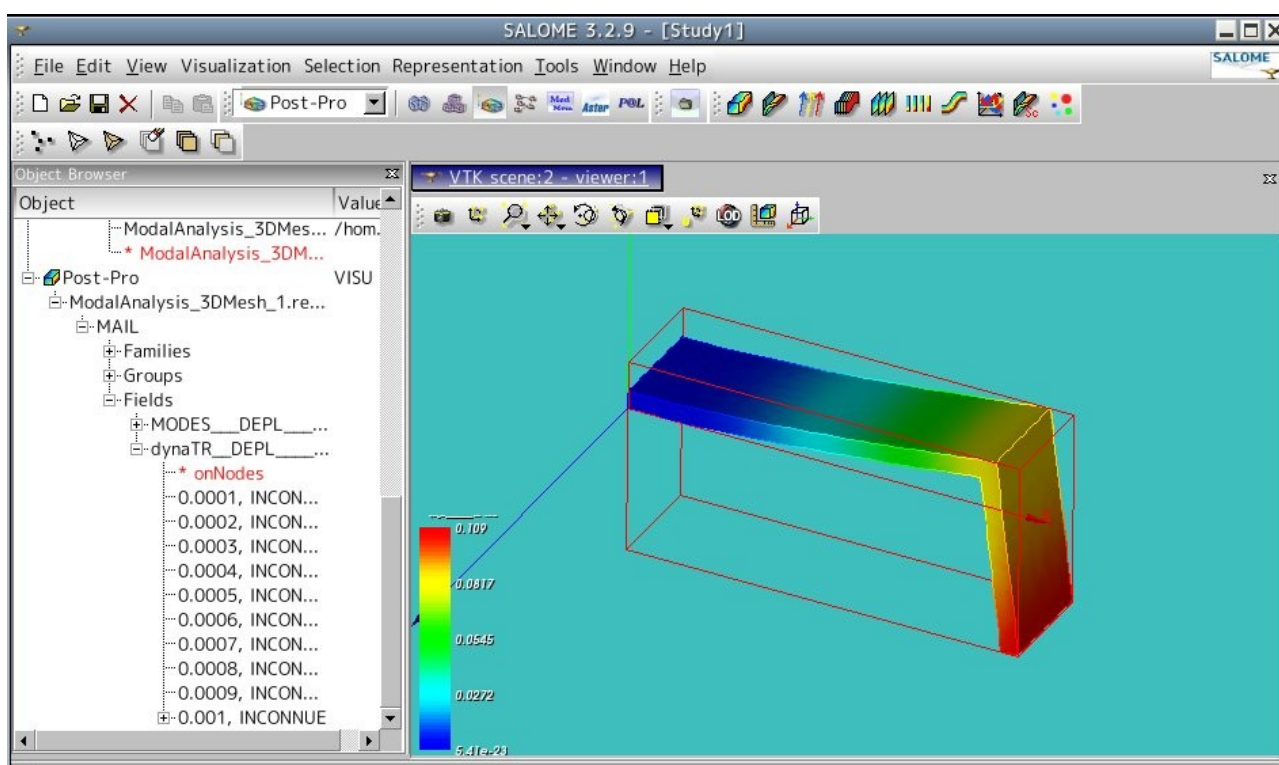
```

以上で修正は完了。

2-4. 実行と結果の確認

修正した Code_Aster を実行する。実行結果は、Post_Pro に結果が出力される。下図の様に 0.0001 から 0.001 まで 0.0001 間隔で 10 ヶの結果が出力されている。コンタ図は、最後の 0.001s 時点の変位を表している。

SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)

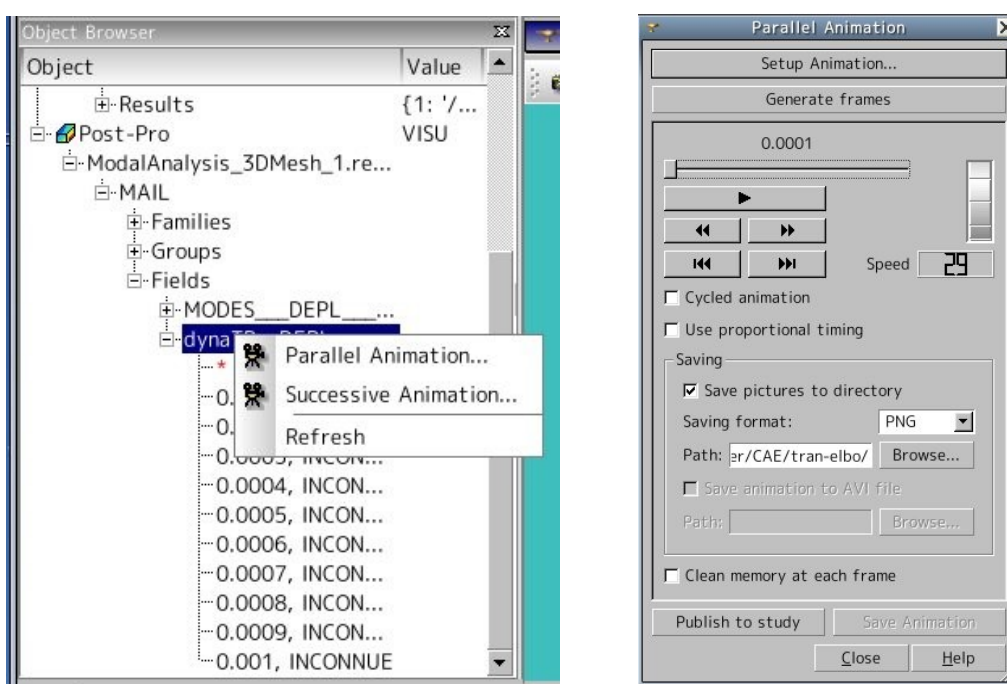


2-5. アニメーション表示

計算結果がでたのでこの結果をアニメーション表示してみる。

ウインド内のコンタ図を全て all hide でクリアした上で、Salome の Post-Pro 画面から、Object Browser ツリー上の「dynaTR_DEPL」上で右クリックして「Parallel Animation...」を選択する。(Successive Animation...を選択しても可。違いが良く判らず。)

選択後は、アニメーション作成画面が現れるのでこの上で作成する。



SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)

アニメーション作成画面上で、「Setup Animation...」をクリックして、何をアニメーション化するかを決定する。ここでは、「Scalar Map on Deformed Shape」を選択し、その「Properties...」を設定する。この後、「Generate Frames」をクリックして、フレームを作成する。

この後は、既にアニメーションができていたので、「>」をクリックして表示させる。表示速度は、つまみをドラッグして決定する。

できあがった、アニメーションの設定は、画面最後の「Publish to study」をクリックすると、Object Browser ツリー上にアニメーションの設定が保存される。

「Save Animation」をクリックした後、アニメーションを表示させると、Saving で設定した方法で設定した directory にアニメーションの各フレームが保存される。保存する場合は、Saving format を「png」に設定しておく。(directory は、~/anime に設定しておく。)

なお、現段階では、アニメーションを動画ファイルに変換する事はこの画面上ではできないので、別途手動で作成する。

2-6. アニメーションファイルの作成保存

前項の操作でアニメーションの画像データ (png フォーマット) が~/anime/フォルダ内に保存されている。この画像データを使って、動画ファイルを作成する。

方法は、~/anime/png2mpeg をダブルクリックして実行すると、結果が result.m1v に動画ファイルができあがる。

尚、動画を作る上で画像データの大きさが 16 の倍数のドット数しか受け付けなかったのが、画像の大きさをこの数字にあわせる必要がある。今回の場合、無理矢理 256x128 に変換している。この大きさを変えれば、自由な大きさを画像に変換できる。

動画データを作成するスクリプトは、最後に示した。

2-7. 結果の再検討

アニメーション結果から、モデルが振動しているのは、1周期と少ししか振動していない、また計算間隔は、1周期を 10 分割していることになるので、ほぼ妥当な間隔といえる。この為計算時間を伸ばし、かつ応力も出力させるてみる。

この為に、Code_Aster を以下のように修正する。

```
DEFI_LIST_REEL      stepTR
DEBUT               0                #0 に変える。
INTERVALLE
  JUSQU_A           0.005            #計算終了時間 この値を修正
  PAS               0.0001
```

JUSQU_A を 0.001 → 0.005 に変更。(5ms まで計算する)

アニメーションで確認すると、振動がすぐに終了している。粘性減衰が大きい様なので、減衰を 0.0001 → 0.00002 に変更して再計算する。

この結果、次第に振幅が小さくなって 5ms 後に振幅が 0 になる様な状態になった。(この状態が正しいのかはよくわからない。)

2-8. 指定した位置の変動

全体の変動は、アニメーションで理解できるが、ある位置を指定してその位置の振幅がどの様に変化していくかを確認してみる。

SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)

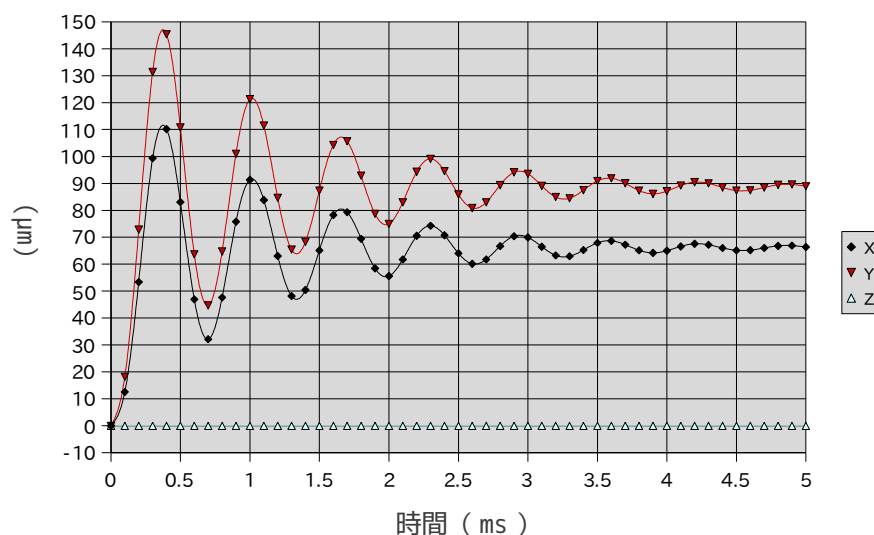
Code_Asterは、以下のように修正する

```
IMPR_RESU
MODELE                               MODELE           #FORMAT を削除する。
b_format_resultat
RESU
  RESULTAT                             dynaTR
  b_sensibilite
  b_extrac
  NOM_CHAM                             DEPL
  b_parametres
  b_cmp
  b_topologie
  NOEUD                                 N3              #この行を追加
  b_valeurs
```

以上の変更を加え、実行する。結果は、FORMATを削除したため、標準出力に出力されるので、周波数応答解析で行った方法で必要なデータのみ抽出し、OpenOfficeでグラフ化する。この結果が下図となる。

この結果は、節点N3 (load面の一部)がどの程度振れているかを示すもの。

Y方向とX方向が最も振れており、最大振幅が約60 μm であることが判る。また、周期は、約0.6msであり、これから、固有振動数は、約1700Hzとなり、2-2項で求めた固有振動数とほぼ一致する。(コーナの点に荷重を負荷させた場合には、ねじりが発生するので、ねじりの固有振動数成分も含むことになり、もっと複雑な動きになる。確認はしていないが・・・)



3. 負荷変動させる場合の解析方法

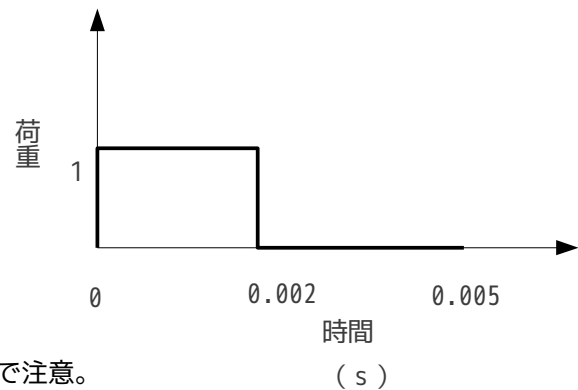
前項までは、負荷を変動させない状態で動解析を行ったが、ここでは負荷を変動させる場合を考えてみる。負荷を変動させる方法は、弾塑性解析で行っており、これと同じ方法で解析してみる。

3-1. Code_Asterの修正

SalomeMeca の使いかた -- 12.0 動解析 (過渡解析)

変動させる負荷のファンクションを定義する。

```
DEFI_FONCTION          loadV
NOM_PARA               INST
VALE                   (0.0,1.0
                       0.002,1.0
                       0.021,0.0
                       0.005,0.0)
```



指定したポイントまで linear に変化させていくので
(0.002, 1.0)、(0.0021, 0.0) を指定している。
ポイント (0.002, 1.0) の 1.0 は、AFFE_CHAR_MECA
で設定した負荷 (1N=0.667 N/m²) の 1 倍を示しているので注意。
次に solver の設定も変更する。

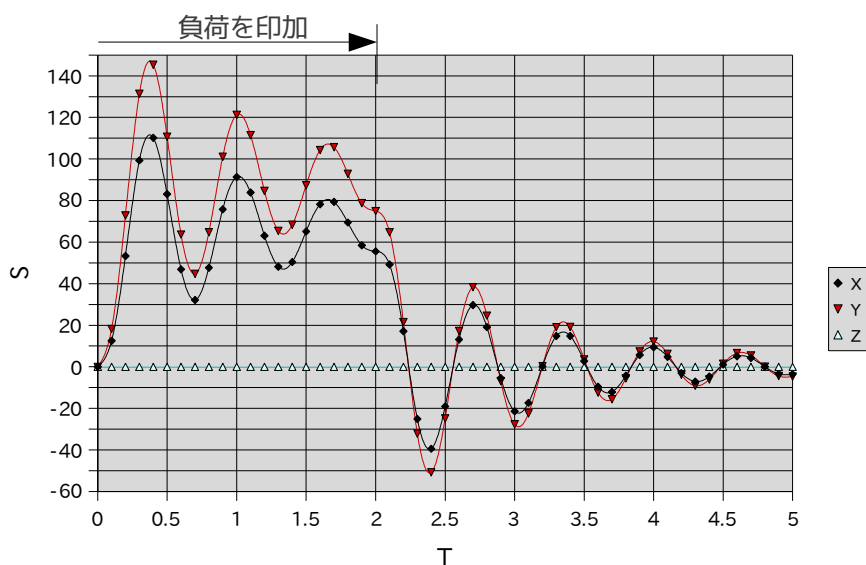
```
DYNA_LINE_TRAN        dynaTR
MODELE                 MODELE
CHAM_MATER             CHMAT
MATR_MASS              MASSE
MATR_RIGI              RIGIDITE
MATR_AMOR              amore
NEWMARK
  DELTA                 0.5
EXCIT
  CHARGE                BLOCAGE
  FONC_MULT             loadV
INCREMENT
  LIST_INST
  b_list_fonc
```

#この 1 行を追加。

3-2. 実行と結果の確認

修正した Code_Aster を実行する。結果をグラフかしたものが下図となる。荷重を負荷している時間が、2ms のみの為、2ms 間は振幅が大きいだがそれを過ぎた段階で、0 を中心に振動している。

SalomeMeca の使いかた -- 12.0 動解析 (過渡解析)



弾塑性変形と同じようにして、負荷を変動させることができる。この為、どんな負荷でもデータさえ入力すれば解析できる。

4. ソースコード

以下にここで解析したソースコードを示す。

4-1. 動画作成用のスクリプト

このスクリプトは、shell スクリプトが Python スクリプトを呼び出して実行している。尚、これらスクリプトを実行するためには、mjpegtools をインストールする必要がある、これをインストールしている。インストール方法は、Salome のインストール方法参照。使用しているコマンドは、imageMagick と mjpegtools のコマンドを使用している。

shell スクリプト

```
-----png2mpeg (png 画像→mpeg1 動画への変換) -----
#!/bin/bash
#
# 画像を変換し MPEG1 動画を作成する
#

# png を jpg (256x256) の画像に変換
ls *.png > tmp.txt
python convj.py > cmd.txt
source cmd.txt
rm a.jpg

#png ファイルを抽出
#変換用のコマンド群を作成する
#コマンド群を実行
#temp ファイルを削除

# jpg 画像群を yuv 形式の動画に変換
ls *.jpg | jpeg2yuv -f 25 -I p > result.yuv

# yuv 形式を MPEG1 の画像に変換
mpeg2enc result.yuv -b 700 -o result.m1v
```

SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)

中間ファイルを削除

```
rm *.jpg
```

-----ここまで-----

Python スクリプト

-----convj.py (png 画像を jpg 画像に変換) -----

```
# coding: EUC-JP
```

```
#
```

```
# png 画像を縮小して jpg 画像に変換し
```

```
# サイズを 256x128 に変えるコマンド群を出力する。
```

```
#
```

```
import sys
```

```
f = open('tmp.txt')
```

```
fname = f.readline()
```

```
fname = fname[:-1]
```

```
while fname:
```

```
    line = 'convert -geometry 256x256 ' + fname + ' a.jpg'
```

#画像を縮小

```
    print line
```

```
    line = 'convert -crop 256x128+0+0 a.jpg ' + fname[:-4] + '.jpg'
```

#画像をトリミング

```
    print line
```

```
    fname = f.readline()
```

```
    fname = fname[:-1]
```

```
f.close()
```

-----ここまで-----

2)Code_Aster

以下に Code_Aster のソースコードを示す。

-----bar2.comm (負荷変動させる場合-アニメーション) -----

```
DEBUT();
```

```
MAIL=LIRE_MAILLAGE(UNITE=20,
                  FORMAT='MED',);
```

```
MODELE=AFFE_MODELE(MAILLAGE=MAIL,
                  AFFE=_F(TOUT='OUI',
                          PHENOMENE='MECANIQUE',
                          MODELISATION='3D',),);
```

```
ACIER=DEFI_MATERIAU(ELAS=_F(E=130300.0,
                          NU=0.343,
                          RHO=7.8e-9,
                          AMOR_ALPHA=0.00002,),);
```

```
CHMAT=AFFE_MATERIAU(MAILLAGE=MAIL,
                  AFFE=_F(TOUT='OUI',
                          MATER=ACIER,),);
```

```
BLOCAGE=AFFE_CHAR_MECA(MODELE=MODELE,
                      DDL_IMPO=_F(GROUP_MA='fix',
```

SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)

```

        DX=0.0,
        DY=0.0,
        DZ=0.0,),
    FORCE_FACE=_F(GROUP_MA='load',
        FY=0.667,)),);

```

```

MACRO_MATR_ASSE(MODELE=MODELE,
    CHAM_MATER=CHMAT,
    CHARGE=BLOCAGE,
    NUME_DDL=CO('NUMEDDL'),
    MATR_ASSE=(_F(MATRICE=CO('RIGIDITE'),
        OPTION='RIGI_MECA',),
        _F(MATRICE=CO('MASSE'),
        OPTION='MASS_MECA',),
        _F(MATRICE=CO('amore'),
        OPTION='AMOR_MECA',),),),);

```

```

loadV=DEFI_FONCTION(NOM_PARA='INST',VALE=(0.0,1.0,
    0.002,1.0,
    0.0021,0.0,
    0.005,0.0,
    ),);

```

```

stepTR=DEFI_LIST_REEL(DEBUT=0.0,
    INTERVALLE=_F(JUSQU_A=0.005,
        PAS=0.0001,)),);

```

```

MODES=MODE_ITER_SIMULT(MATR_A=RIGIDITE,
    MATR_B=MASSE,
    CALC_FREQ=_F(OPTION='PLUS_PETITE',
        NMAX_FREQ=5,)),);

```

```

dynaTR=DYNA_LINE_TRAN(MODELE=MODELE,
    CHAM_MATER=CHMAT,
    MATR_MASS=MASSE,
    MATR_RIGI=RIGIDITE,
    MATR_AMOR=amore,
    NEWMARK=_F(DELTA=0.5,),
    EXCIT=_F(CHARGE=BLOCAGE,
        FONC_MULT=loadV,),
    INCREMENT=_F(LIST_INST=stepTR,)),);

```

```

IMPR_RESU(MODELE=MODELE,
    FORMAT='MED',
    RESU=_F(MAILLAGE=MAIL,
        RESULTAT=MODES,
        NOM_CHAM='DEPL',),),);

```

```

IMPR_RESU(MODELE=MODELE,
    FORMAT='MED',

```

SalomeMeca の使いかた -- 12.0 動解析 (過渡解析)

```
RESU=_F(RERESULTAT=dynaTR,
        NOM_CHAM='DEPL',,);
```

```
FIN();
```

```
-----ここまで-----
```

```
-----bar2.comm (負荷変動させる場合-データ抽出) -----
DEBUT();
```

```
MAIL=LIRE_MAILLAGE(UNITE=20,
                  FORMAT='MED',);
```

```
MODELE=AFPE_MODELE(MAILLAGE=MAIL,
                  AFPE=_F(TOUT='OUI',
                        PHENOMENE='MECANIQUE',
                        MODELISATION='3D',),);
```

```
ACIER=DEFI_MATERIAU(ELAS=_F(E=130300.0,
                          NU=0.343,
                          RHO=7.8e-9,
                          AMOR_ALPHA=0.00002,),);
```

```
CHMAT=AFPE_MATERIAU(MAILLAGE=MAIL,
                   AFPE=_F(TOUT='OUI',
                         MATER=ACIER,),);
```

```
BLOCAGE=AFPE_CHAR_MECA(MODELE=MODELE,
                      DDL_IMPO=_F(GROUP_MA='fix',
                                  DX=0.0,
                                  DY=0.0,
                                  DZ=0.0,),
                      FORCE_FACE=_F(GROUP_MA='load',
                                    FY=0.667,),);
```

```
MACRO_MATR_ASSE(MODELE=MODELE,
                CHAM_MATER=CHMAT,
                CHARGE=BLOCAGE,
                NUME_DDL=CO('NUMEDDL'),
                MATR_ASSE=( _F(MATRICE=CO('RIGIDITE'),
                              OPTION='RIGI_MECA',),
                          _F(MATRICE=CO('MASSE'),
                              OPTION='MASS_MECA',),
                          _F(MATRICE=CO('amore'),
                              OPTION='AMOR_MECA',),),);
```

```
loadV=DEFI_FONCTION(NOM_PARA='INST',VALE=(0,1,
      0.002,1,
      0.0021,0,
      0.005,0,
      ),);
```

SalomeMecaの使いかた -- 12.0 動解析 (過渡解析)

```
stepTR=DEFI_LIST_REEL(DEBUT=0.0,
                      INTERVALLE=_F(JUSQU_A=0.005,
                                      PAS=0.0001,)),);

MODES=MODE_ITER_SIMULT(MATR_A=RIGIDITE,
                       MATR_B=MASSE,
                       CALC_FREQ=_F(OPTION='PLUS_PETITE',
                                      NMAX_FREQ=5,)),);

dynaTR=DYNA_LINE_TRAN(MODELE=MODELE,
                      CHAM_MATER=CHMAT,
                      MATR_MASS=MASSE,
                      MATR_RIGI=RIGIDITE,
                      MATR_AMOR=amore,
                      NEWMARK=_F(DELTA=0.5,)),
                      EXCIT=_F(CHARGE=BLOCAGE,
                              FONC_MULT=loadV,)),
                      INCREMENT=_F(LIST_INST=stepTR,)),);

IMPR_RESU(MODELE=MODELE,
          FORMAT='MED',
          RESU=_F(MAILLAGE=MAIL,
                  RESULTAT=MODES,
                  NOM_CHAM='DEPL',)),);

IMPR_RESU(MODELE=MODELE,
          RESU=_F(RESULTAT=dynaTR,
                  NOM_CHAM='DEPL',
                  NOEUD='N3',)),);

FIN();
-----ここまで-----
```