OpenCAE勉強会@岐阜
2016/04/09(土曜日)

FrontISTRとCalculixのユーザ定義関数機能を使った簡易カスタマイズ方法

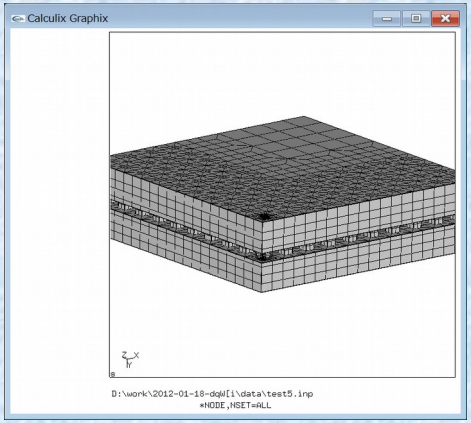
OpenCAE勉強会@岐阜
SH

本日の発表内容

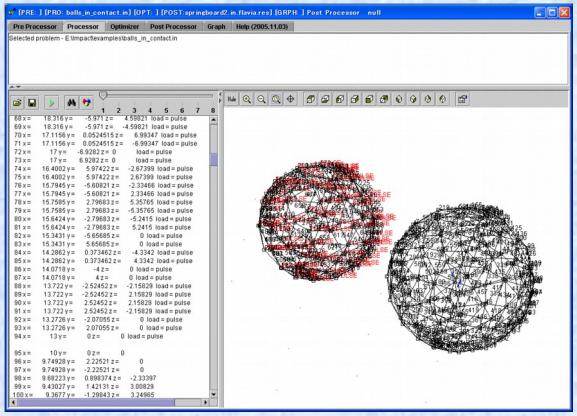
①CalculixとFrontISTR
②ユーザ定義関数について
③FrontISTRのユーザ定義関数
④Calculixのユーザ定義関数

代表的なオープンソース構造解析ソルバ

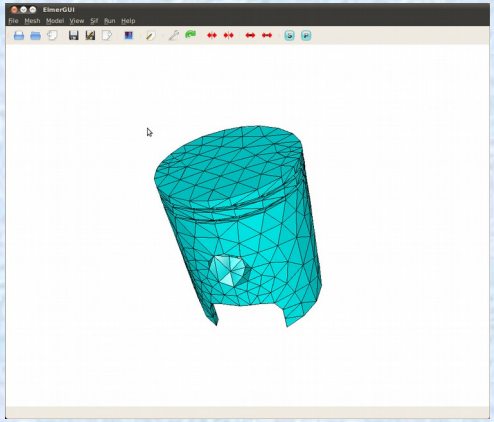
名前	URL	特徴など
Calculix	www.calculix.de	Abaqusライクな非線形構造解析、材料非線形、接触解析、動解析(ドイツ)
CodeAster (Salome-meca)	www.code-aster.org	大規模な非線形構造解析、日本では最近活用がさかん(フランス)
Impact	impact.sourceforge.net	陽解法非線形解析ソルバ(ロシア)
TOCHNOG	tochnog.sourceforge.net/	構造解析(非線形, 接触動解析etc.)
WARP3D	cern49.cee.uiuc.edu/cfm/warp3d.html	構造解析(き裂解析向けの非線形, 接触解析等) のソルバ(米国)
Elmer	www.csc.fi/english/pages/elmer	連成解析ソルバ (構造解析) (ﾌｨﾝﾗﾝﾄﾞ)
Adventure	adventure.sys.t.u-tokyo.ac.jp/jp/	大規模構造解析ソルバ(日本)
FrontISTR	www.ciss.iis.u-tokyo.ac.jp/riss/dl/	大規模構造解析ソルバ(日本)



Calculix

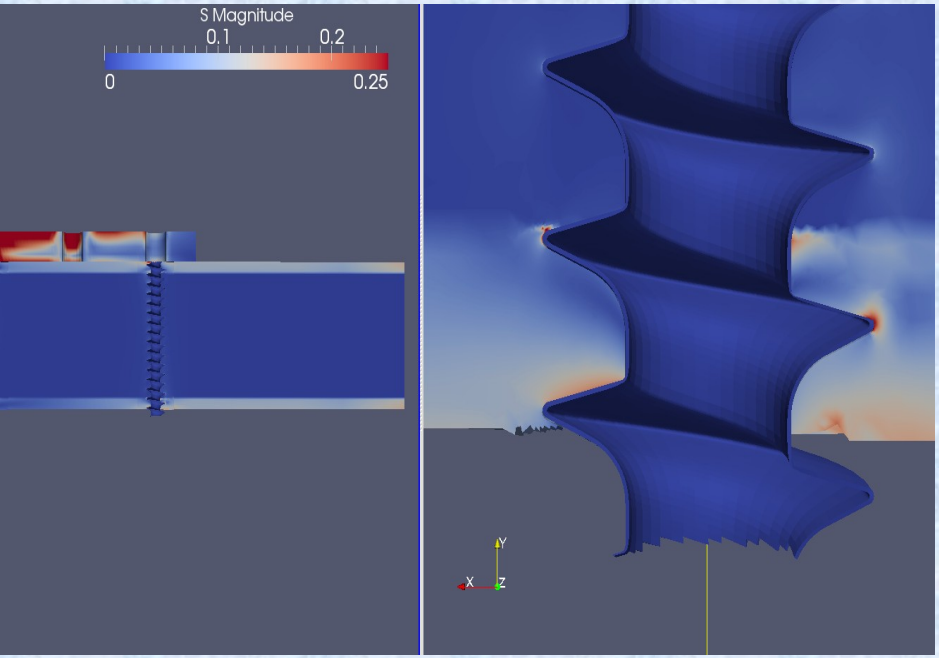


Impact

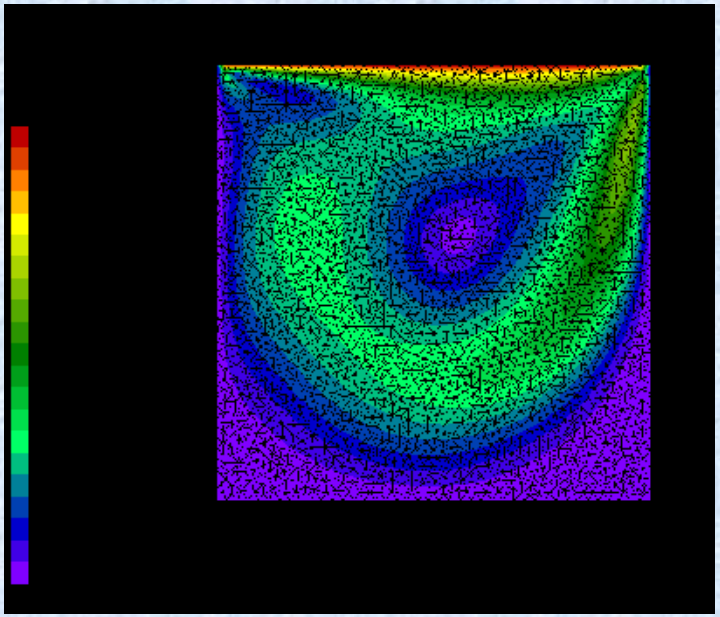


Elmer

Calculix



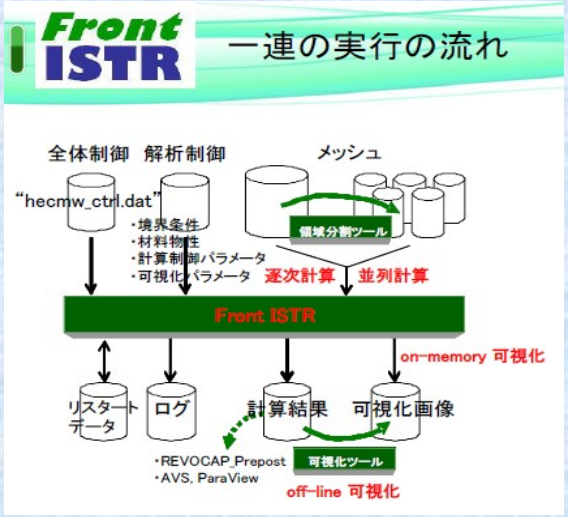
CalculiX Extras
project 解析事例



Cavity FLOW in Calculix

- 商用ソフトABAQUSと同様の入力書式をもつオープンソース ABAQUSを仕事で使っている人は文法を勉強しないでそのまま使える。知らない人もABAQUSのマニュアルを見れば大体使い方が分かる。
(ﾃｷｽﾄ入力ﾍﾞｰｽのﾓﾃﾞﾗｰ, ﾒｯｼｬｰ, ｿﾙﾊﾞ, POSTを包含した非線形構造解析ソフト、一部流体解析も可能)
- http://www.bconverged.com/calculix にてWindows実行バイナリも公開
- Linux で利用する場合は本家のHP からソースをダウンロードしてコンパイル→ http://www.dhondt.de/
するかCaelinux(DVD-iso)版を利用する。 ソースのコンパイルは結構大変。
- 非線形(大変形、接触解析、材料非線形(塑性、クリープ、温度依存etc)が可能
- 課題；使っている行列ソルバ(Spools)が古い→ 標準設定ではあまり大規模な計算(100万メッシュ以上?)には対応できない。Extras プロジェクトで別ソルバ(CUDAベース行列ソルバ等Cuda-CUSP, Cholmod) のインターフェースプログラムが公開されている→ http://homepages.wmich.edu/~pjm8969/research/ccx_extras-dl.html

Ｆｒｏｎｔ ＩＳＴＲ①



FrontISTRの機能一覧

注：赤字は平成24年度拡充機能

線形静解析	等方性／異方性（熱応力解析を含む）
非線形静解析	材料非線形： 超弾性／弾塑性／熱弾塑性／粘弾性／クリープ 等方／移動／複合硬化 幾何学的非線形： Total Lagrange法／Updated Lagrange法 境界非線形(接触)： Lagrange乗数法、有限すべり、摩擦
線形動解析	時刻歴応答（陽解法／陰解法）、周波数応答
非線形動解析	陽解法／陰解法、接触解析機能
固有値解析	ランチョス法 、変形後解析機能
熱伝導解析	定常／非定常（陰解法）
要素タイプ	四面体／六面体／五面体／シェル／トラス／梁 1次／2次、非適合モード、選択的次数低減積分
解析支援	境界条件ステップ制御、リスタート、ユーザーサブルーティン

~~ダウンロードは下記から~~

~~http://www.ciss.iis.u-tokyo.ac.jp/riss/~~

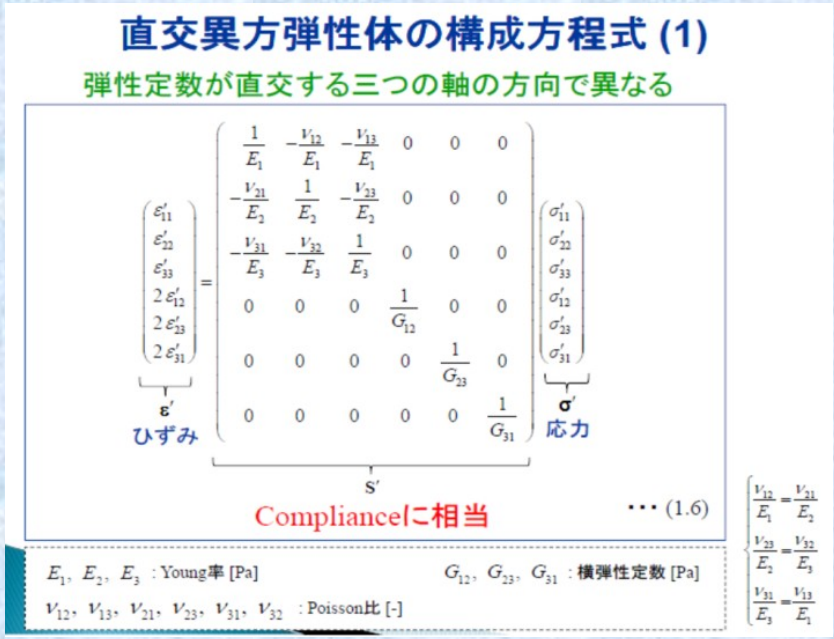
http://www.multi.k.u-tokyo.ac.jp/FrontISTR/index.html

- FrontISTRとは東大が国プロで開発しているオープンソースソフトウェア
- 有限要素法構造解析ソフトウェア各種非線形解析機能を有する
- 分散領域メッシュ＋反復法ソルバによるノード間並列解析機能を有する
- ライセンスフリー(商業利用時は独自契約が必要)
- プリは同じプロジェクトで開発されたRevocapを使用, MeshはABAQUSに似た独自書式
- 変形・応力解析機能
  -線形静解析, 非線形静解析, 大変形解析
  -材料非線形解析（弾塑性・超弾性・粘弾性・クリープ・ユーザ定義材料）
  -接触解析(拡張ラグランジュ、ラグランシュ法)
  -動的陽解法は非接触解析のみ可能
  -陰的時間積分法による接触を考慮した過渡解析(衝突解析)も2012年度に実装した

ＦｒｏｎｔＩＳＴＲ②

- ＦｒｏｎｔＩＳＴＲ研究会として東大奥田研究室が独自に開発は現在も継続。研究会は平日実施だがだれでも無料で参加できるので、興味のあるかたは参加を検討ください。
- 非線形有限要素法ソースコード実装方法についてかなり詳しく解説してくれるので貴重
  → ただしFrontISTRはバグがかなりあるので、マニュアルに書いてある内容で動かないことが多々あります。そんな時はこのような勉強会にて報告すると運が良いとすぐに開発者がコードを直してくれることもあります～。

http://www.multi.k.u-tokyo.ac.jp/FrontISTR/index.html



②ユーザ定義関数について

① 一般にユーザサブルーチンとかユーザ関数とか呼ばれており、簡単なプログラムを書くことで独自機能を定義できるもの。

② 元々はソースコード非公開の汎用ソフトでカスタマイズするため**に考えられた**

③ 某ABAQUSではユーザサブルーチンとなっている（無料で使えるABAQUS Student Editionではこのユーザサブルーチンは残念ながら使えない）

④ オープンソースCAEソフトでも同じような機能がある

⑤ 秋山さんがユーザ定義関数の一種のUMATについて調べた資料を公開してくれてます。

②ユーザ定義関数について

・秋山さんUMATに関する資料
（第35回オープンCAE勉強会＠富山）より

各サブルーチンの比較

ABAQUS

```
SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
    1 RPL,DDSDDT,DRPLDE,DRPLDT,
    2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
    3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
    4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
```

Code-Aster

```
SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
   1 RPL,DDSDDT,DRPLDE,DRPLDT,
   2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
   3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
   4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
```

CalculiX　umat.f

```
subroutine umat(stress,statev,ddsdde,sse,spd,scd,
    &  rpl,ddsddt,drplde,drpldt,
    &  stran,dstran,time,dtime,temp,dtemp,predef,dpred,cmname,
    &  ndi,nshr,ntens,nstatv,props,nprops,coords,drot,pnewdt,
    &  celent,dfgrd0,dfgrd1,noel,npt,layer,kspt,kstep,kinc)
```

②ユーザ定義関数について
-各解析ソフトのuser関数まとめ-

ソフト名称	ユーザ関数	言語他
Calculix	**User Creep, User Hardening** **User Initial condition(初期応力他)** **User load, User 境界条件** **, User Material**	**Fortran**
FrontISTR	**Uyield, Uelastic, Uload,Umat**	**Fortran90**
CodeAster	**User Material他**	**Frotran**
ABAQUS(商用ソフト)	Umat(ユーザ定義材料) User Creep(ユーザ定義グループ) User Field(ユーザ定義フィールド) etc.	Fortran or C

③FrontISTRのユーザ定義関数

- FrontISTRとCalculix のユーザ関数はソースコードをコンパイルする時に必要なFortranのソースプログラム部分を書きかえてコンパイルする。したがってABAQUSなどとは異なりユーザ関数とはいえ作業手順はソースコードの一般的なカスタマイズと全く変わらない。

- ただし、ユーザ関数は1つのソースプログラム内で必要最低限の部分の変更を行うだけで、他のソース部分を改変することなく利用可能なので比較的簡単に利用できる
（ハズなのだが、バグがあるのでそう簡単でもない）

- コンパイル作業が必須なのでビルド環境は必須である
(Fortran, C, makeなど)

③FrontISTRのユーザ定義関数

- FrontISTRのUser subroutineの場所

FrontISTR_V44/fistr1/src/lib/user

uelastic.f90　uhardening.f90　umat.f90
uload.f90　　uyield.f90

このuXXXX.f90
を自分のやりたいよう
に書きなおして利用す
る

```
!> This subroutine calculates constitutive relation
subroutine uElasticMatrix( matl, strain, D )
   use hecmw
   real(kind=kreal), intent(in)  :: matl(:)   !< material properties
   real(kind=kreal), intent(in)  :: strain(6) !< Green-Lagrangen strain
   real(kind=kreal), intent(out) :: D(6,6)    !< constitutive matrix

! following examples of linear elasticicty
   real(kind=kreal) :: EE, PP
   EE = matl(1)
   PP = matl(2)
   D(1,1)=EE*(1.0-PP)/(1.0-2.0*PP)/(1.0+PP)
   D(1,2)=EE*PP/(1.0-2.0*PP)/(1.0+PP)
   D(1,3)=D(1,2)
   D(2,1)=D(1,2)
   D(2,2)=D(1,1)
   D(2,3)=D(1,2)
   D(3,1)=D(1,3)
   D(3,2)=D(2,3)
   D(3,3)=D(1,1)
   D(4,4)=EE/(1.0+PP)*0.5
   D(5,5)=EE/(1.0+PP)*0.5
   D(6,6)=EE/(1.0+PP)*0.5
end subroutine
```

User Elastic の例

Sampleとして
線形等方弾性体の
定義が入力されている

コントロールファイルから

EE: ヤング率
PP: ポアソン比

を呼び込みDマトリックス
に代入する事例

例えばUser材料として
異方性材としてDマトリックス
にすべて任意の値を与えるこ
とができる

③FrontISTRのユーザ定義関数

- 利用方法：Controlファイルで User定義関数を呼び出す指定を行う

  (sample:examples/static/user)

Example Fileにある
uelas.cnt の記述例

```
!MATERIAL, NAME=M1
!ELASTIC, TYPE=USER
  10.0E+04,   0.0
```

TYPE=USER と指定することで
User定義線形弾性のルーチンが
呼ばれる

他のユーザ関数も同様に
！PLASTIC, ！Material, !CLOAD
Etc. の指定でTYPEをUSERに変更する
ことでユーザ関数を呼び出すことが
できる。

ちなみにFrontISTRのuelastic.f90
はバグがあるみたいで
例題は動きますが数字を変更すると
正常動作しない！

この間違っているところのソース
修正と異方性材料への変更例を
今度の岐阜の夏合宿で実施予定

③FrontISTRのユーザ定義関数

実行手順

1）/fistr1/src/lib/user のuelastic.f90 のソースを変更

2) 既存のuelastic.o を消去

3) FrontISTR_V44/fistr1/src/lib/user で
make コマンドを実施

4) FrontISTR_V44/fistr1/lib の下のライブラリを一度消去

5）FrontISTR_V44/fistr1/bin の下のfistr1コマンドを一度消去

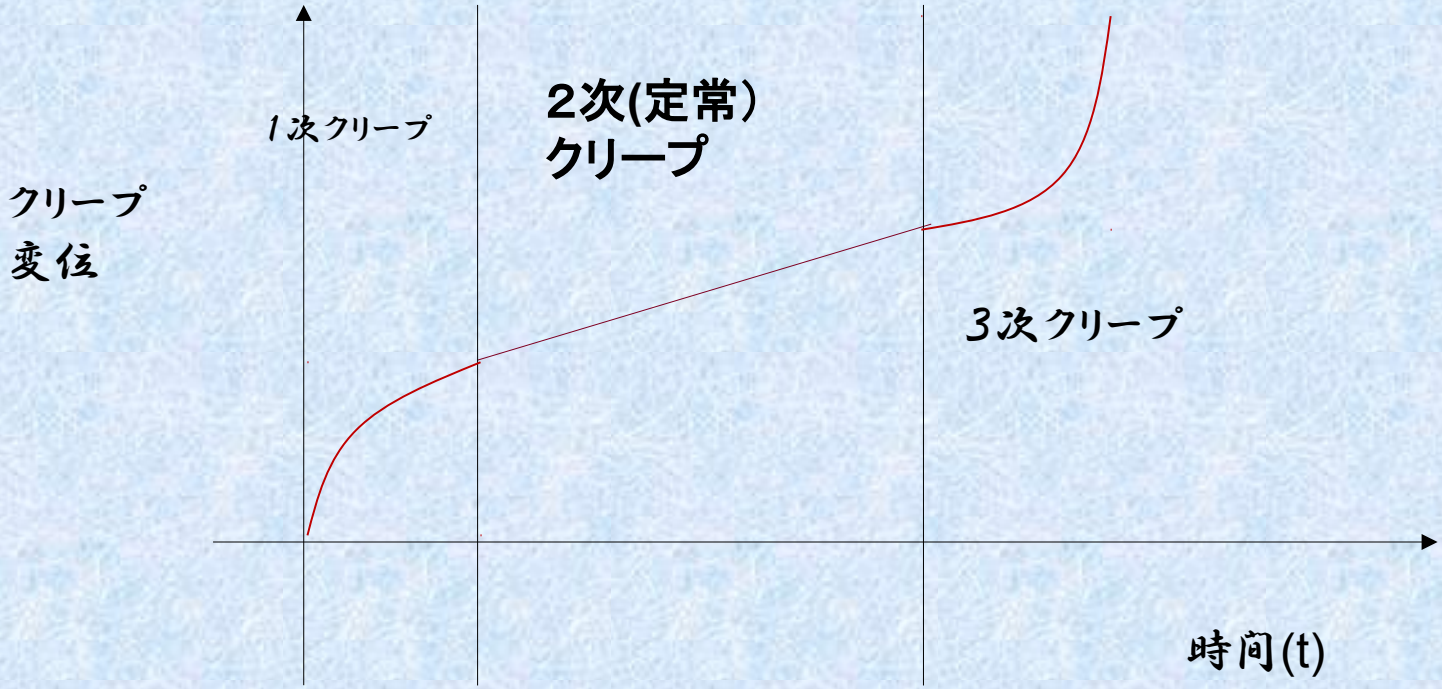
6）FrontISTR_V44/fistr1 の下でmakeコマンドを実施

④ Calculixのユーザ定義関数

- Calculixのユーザ関数もほぼFrontISTRと同様の方法・手順で利用できる。
- ちなみにCalculixのユーザサブルーチン記述はABAQUSとほぼ同じなのでABAQUSを使っていたユーザは違和感なく利用できるであろう？？
- 今回はユーザクリープ（Creep.f）の書式を例に挙げる

クリープについて その1

金属材料のクリープでは、1次クリープ(遷移クリープ)、2次クリープ(定常クリープ)、3次クリープの3段階モデルで表される。



クリープについて その2

クリープの計算モデルとしては定常クリープを表す代表的なモデルとしてノートン(Norton)則がある。

$$\frac{de}{dt} = A\sigma^n$$

e: クリープひずみ
S:応力
A: 定数(材料物性値)
n:定数(材料物性値)

これの応用モデルで遷移クリープも含めて表現できる時間硬化型モデルがある。

$$\frac{de}{dt} = A\sigma^n t^m$$

e: クリープひずみ
S:応力
A: 定数(材料物性値)
n:定数(材料物性値)
m:定数(材料物性値)

FrontISTRで使えるのはノートン型か時間硬化型のみである。同時に塑性は考慮できない(弾クリープモデルのみ)

クリープについて その3

FrontISTRでクリープを考慮するためには制御ファイルに以下の設定を行う

1) 材料にクリープ物性を定義する
2) 解析STEPで時間効果を含めた解析を実施する宣言を行う
3) 解析時間を定義する(何時間分計算をする)
  → 計算上は100年分とかのクリープ
    ひずみを計算することもできる

以下に制御ファイルの編集箇所を指定する

クリープについて　その4

- Tutorial例題は物性値設定の参考になるが、FrontISTRのCreep解析TutorialはSTEP設定がおかしいのであまり参考にならない。

```
!MATERIAL, NAME=XXXX
!ELASTIC, TYPE=ISOTROPIC
1000, 0.0
!DENSITY
7860
!EXPANSION_COEFF, DEPENDENCIES=0
0.000012
!CREEP, TYPE=NORTON
 1.e-4, 2.0, 0.0
```

時間硬化則のA, n,m の定数を指定する
$m=0$の場合はノートン則になる

クリープについて　その5

STEPの設定例

STEP1 では線形で通常のように荷重を負荷する

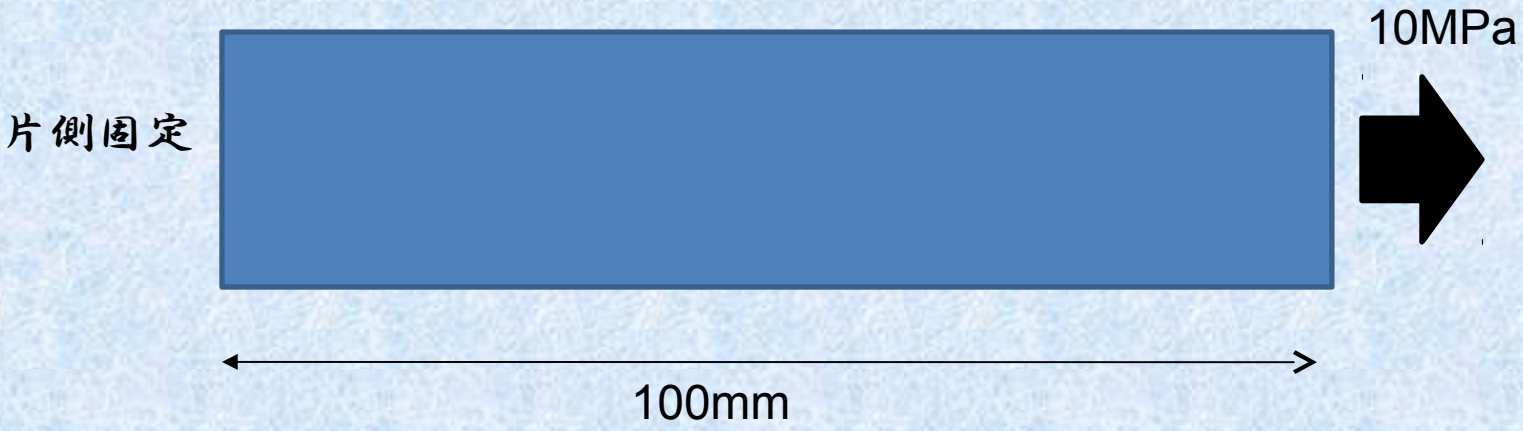
```
!STEP, TYPE=STATIC, CONVERG=1e-6, SUBSTEPS=1, MAXITER=50
BOUNDARY, 1
LOAD, 1
!STEP, TYPE=VISCO, CONVERG=1.0e-2, SUBSTEPS=10, MAXITER=50
1.0, 10.0
BOUNDARY, 1
LOAD, 1
```

STEP2でTYPE =VISCO (クリープまたは粘弾性材の解析)を指定、STEP時間には実際の時間を指定する(SEC,またはHOURなど)

ここの時間の単位はクリープ物性A,n, mと整合させる

クリープについて　その6

簡単なクリープ解析の例で以下の四角断面の棒に一定荷重（応力）を負荷したときのクリープ変位を計算する。



E=1000MPa

クリープ物性

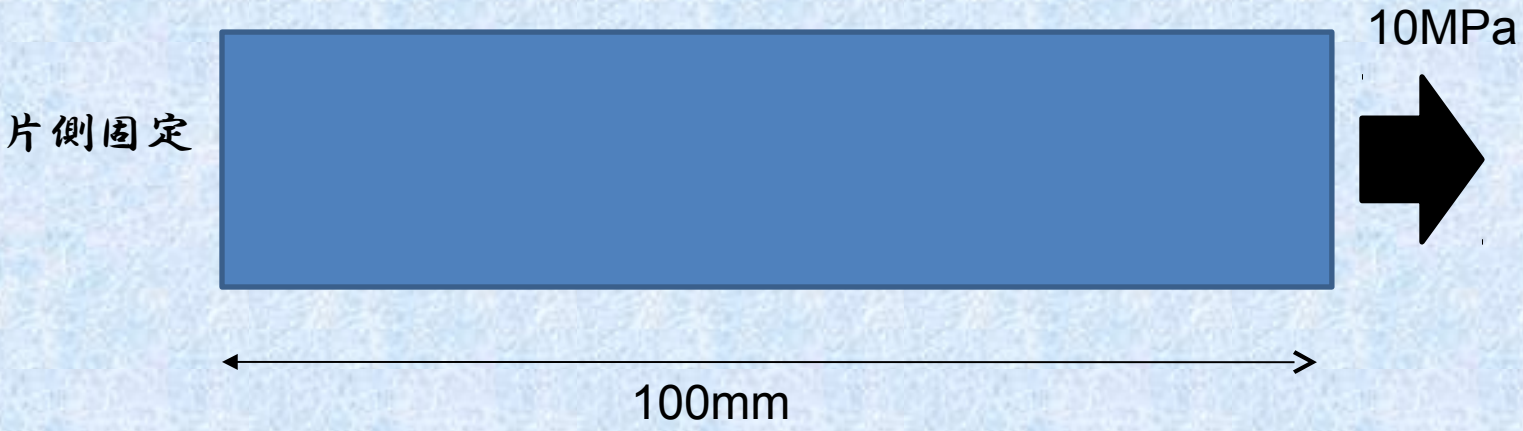
A=1e-3, n=2, m=0

Step1. 弾性解析

Step2. クリープ解析(10秒間)

クリープについて　その6

簡単なクリープ解析の例で以下の四角断面の棒に一定荷重(応力)を負荷したときのクリープ変位を計算する。

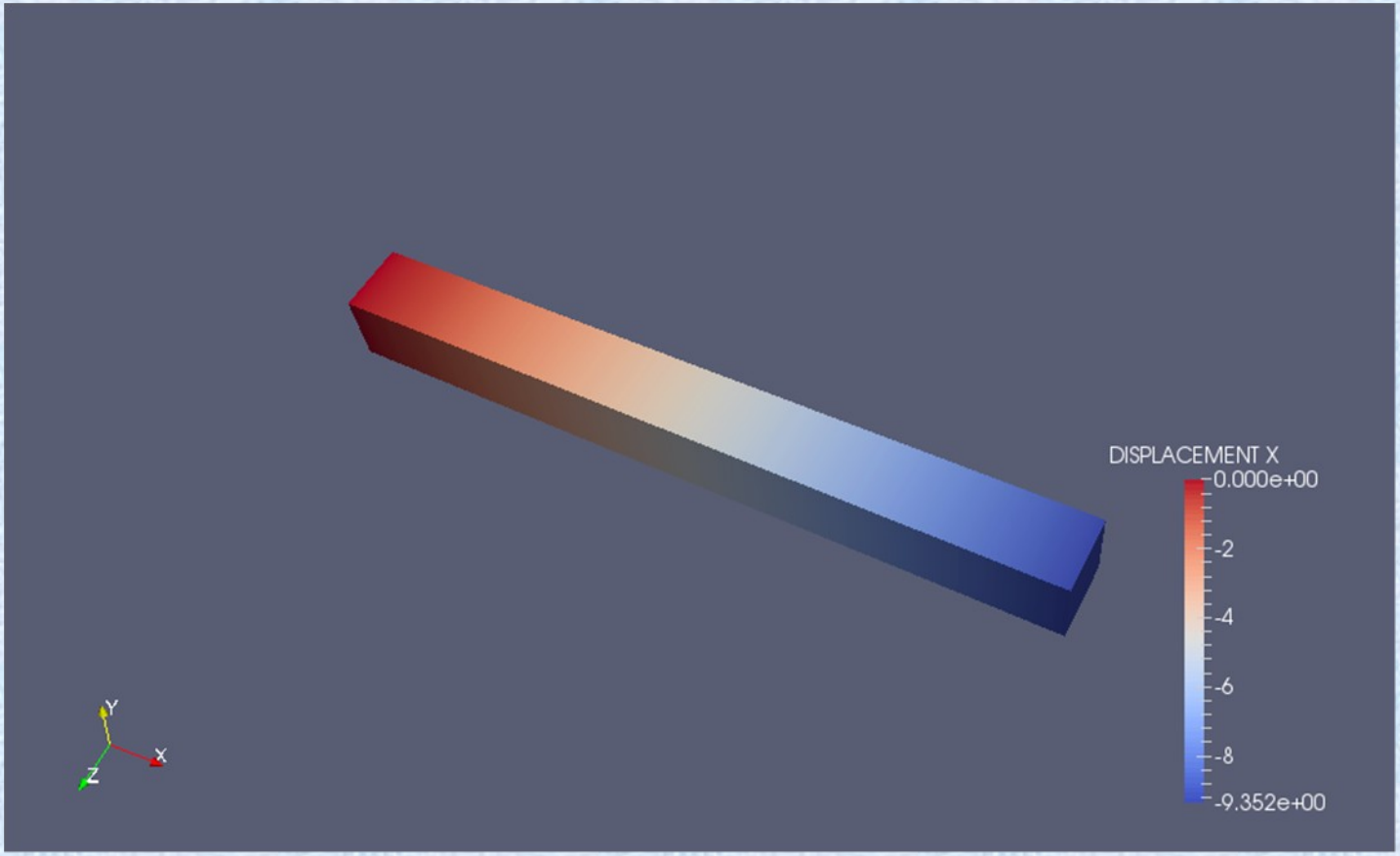


E=1000MPa
クリープ物性
A=1e-3, n=2, m=0

Step1. 弾性解析
Step2. クリープ解析(10秒間)

クリープについて　その7

Step1の変位、Step2の変位は手計算で計算できる。
(計算方法は省略する)
Step1の変位= 1mm
Step2のクリープ変位=10mm
FrontISTRの計算結果と比較する。



Step2の計算結果変位=9.352mm

STEP1と比べるとかなり手計算とずれが大きいが、まあ大体手計算結果とあっている

④ Calculixのユーザ定義関数

- Calculixの場合、ソースファイルはすべて同じ階層（src) にある。
- ユーザクリープは”creep.f” で定義される。

```
      subroutine creep(decra,deswa,statev,serd,ec,esw,p,qtild,
     &  temp,dtemp,predef,dpred,time,dtime,cmname,leximp,lend,
     &  coords,nstatv,noel,npt,layer,kspt,kstep,kinc)
!
!     user creep routine
!
!     INPUT (general):
!
!     statev(1..nstatv)  internal variables
!     serd               not used
!     ec(1)              equivalent creep at the start of the increment
!     ec(2)              not used
```

④ Calculixのユーザ定義関数

```
!     temp            temperature at the end of the increment
!     dtemp           not used
!     predef          not used
!     dpred           not used
!     time(1)         value of the step time at the end of the increment
!     time(2)         value of the total time at the end of the increment
!     dtime           time increment
!     cmname            material name
!     leximp          not used
!     lend           if = 2: isotropic creep
!                  if = 3: anisotropic creep
!     coords(1..3)      coordinates of the current integration point
!     nstatv          number of internal variables
!     noel            element number
!     npt            integration point number
!     layer           not used
!     kspt            not used
!     kstep           not used
!     kinc           not used
```

内部変数として温度Step時間、時間増分、要素番号、積分点等利用できる

④ Calculixのユーザ定義関数

```
!     INPUT only for elastic isotropic materials:
!      qtild              von Mises stress
!
!     INPUT only for elastic anisotropic materials:
!      decra(1)           equivalent deviatoric creep strain increment
!
!
!     OUTPUT (general):
!
!     decra(5)           derivative of the equivalent deviatoric
!                        creep strain increment w.r.t. the von Mises
!                        stress
      qtild=(1.d10*decra(1)/dtime)**0.2d0
      decra(5)=5.d-10*dtime*qtild**4
!
```

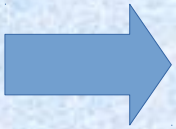
ここに相当応力と
クリープひずみの
関係式を記述する
Ec = Aσ＊＊ n*dt

入力変数として　qtild (Von Mises 相当応力）
クリープ歪み増分（計算インクリメント初期）が利用できる

④ Calculixのユーザ定義関数

- FrontISTR同様に利用する場合はcreep.f を書き直し、creep.o を消してからmake を再度実行する（うまくいかなかったら諦めて全部 再make する）
- 入力ファイルの中でクリープ構成則をUSERに変更する

```
*Material, name=MAT1
*Elastic
1000., 0.3
*CREEP,LAW=NORTON
0.1,2,0.0
```

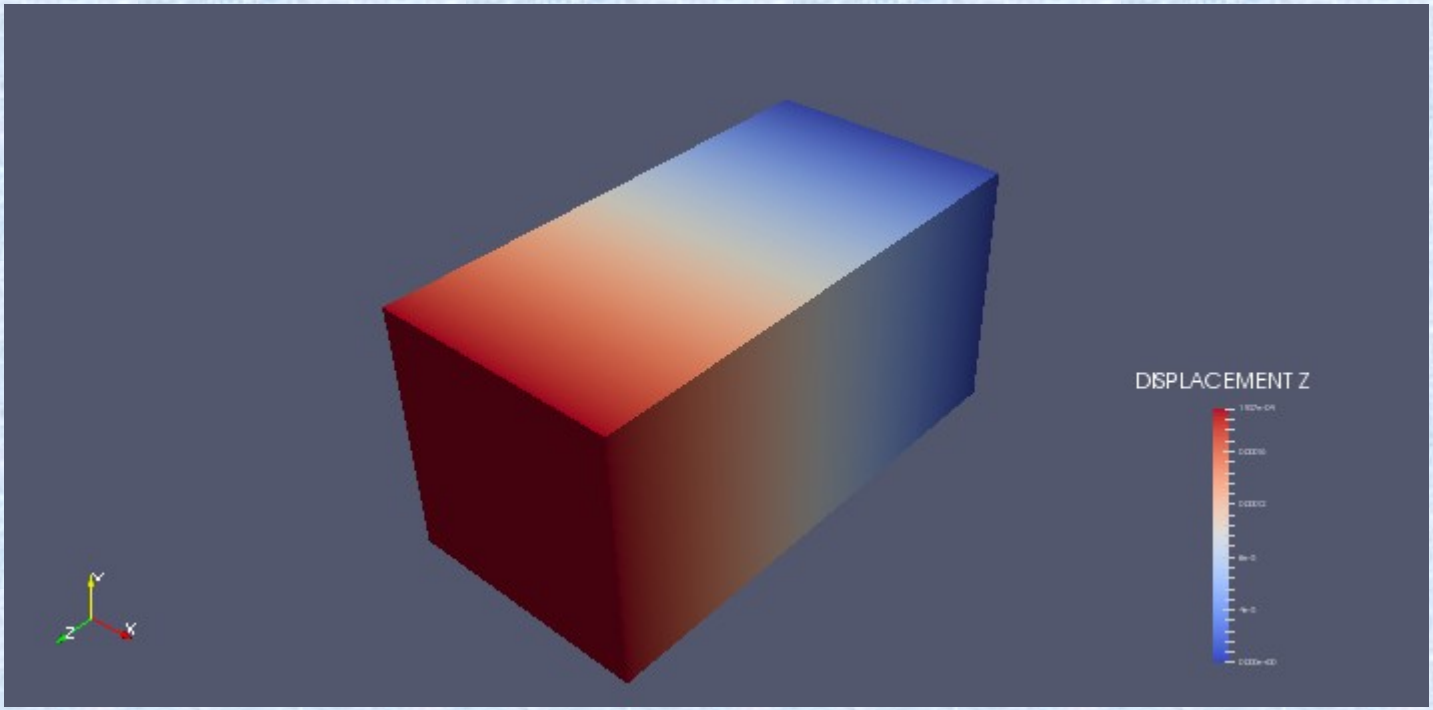


```
*Material, name=MAT1
*Elastic
1000., 0.3
*CREEP,LAW=USER
```

これでOK！

クリープ解析のテスト

- 簡単なモデルでクリープ解析のテストを行ってみる。
  ABAQUS(Norton則)、Calculix(Norton則)、FrontISTR(Norton則)の解析結果を手計算結果と比較する
  → 答えが合わないので調査中



まとめ

- ユーザ関数・ユーザサブルーチン
  調査を行った。
- CalculixとFrontISTRの簡易モデルで
  検証しようとしたが今のところ妥当な結果が得
  られていない。