

解析ツールとしてのサブルーチン提供

近年、テーブル機能によりサブルーチンを使わずとも目的の解析ができるようになった。

お約束したスケール (scale.f、剛体の大きさ変更など) は、テーブルで簡単に設定できるので記載していません。

[今回提供できるユーザーサブルーチン/MARC解析]

同様の考え方でABAQUS等でも補えると考えます。

- 1) FORCDT : 変位条件を自在にコントロール
- 2) FORCEM : 荷重条件を自在に・・
- 3) PLOTV : MARCでのひずみは工学ひずみの出力がありません。出力を演算でコントロール、工学ひずみも・・
- 4) UFRIC : 摩擦をコントロール、節点圧力から面圧依存性も・・・
- 5) MOTION : 剛体の動きを関数などでコントロール、若しくは節点引数などで・・・

その他、UFORMなどでリンクの変更 (Tying_Change) など可能になり、
MENTATでできない機能も・・・

1) FORCDT：変位条件を自在にコントロール

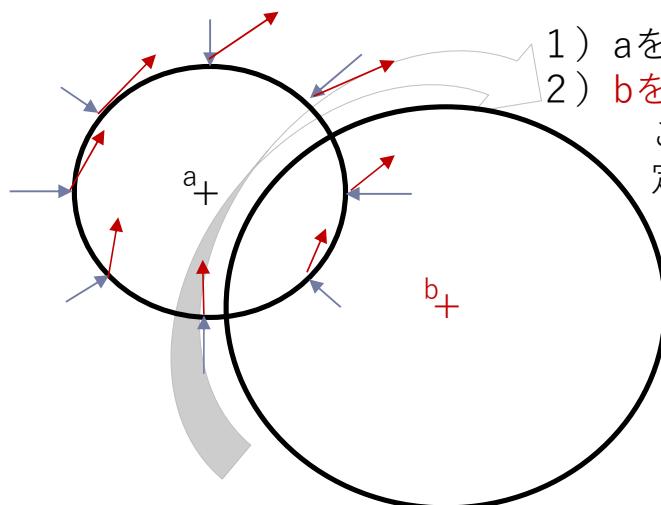
概要

2D-軸対称モデルでの外筒絞りや内筒矢通しであれば、Y方向変位で十分に絞りを表現できる。

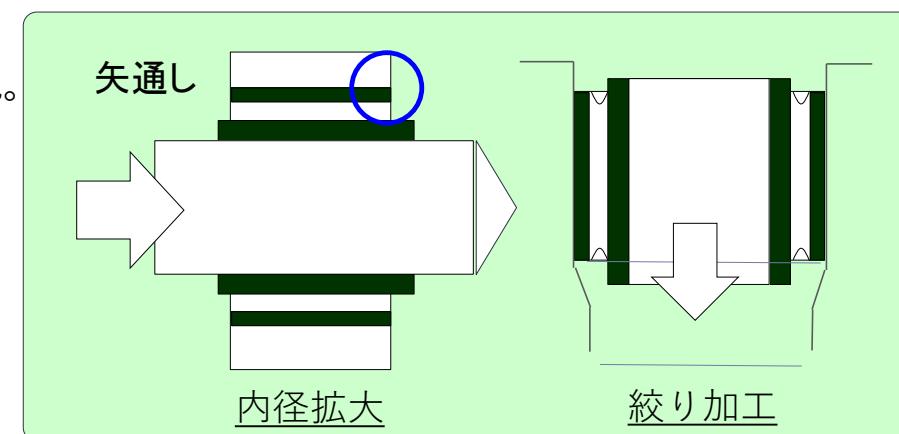
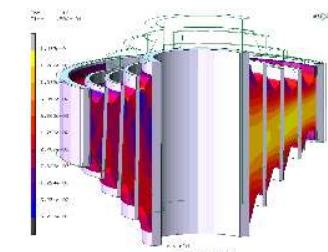
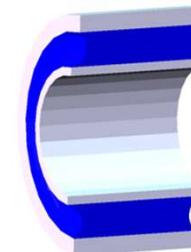
しかし、3次元で法線上に変形を与えようすると円筒座標系を定義して、法線方向への変位で解析が可能となる。

しかし、その後内外筒共に併進移動など行うためには、初期に定義した局座標を無効にしなければならないが、その方法はない。

また、2か所の中心に対して2段階定義、つまり途中で中心の変更など定義するのは、簡単ではない。



- 1) aを中心いて法線方向、絞り(↙) を表現。
 - 2) bを中心いて回転させる(↗)。
- これらをサブルーチンなしで定義できれば必要ありません。



複雑な動き、または節点ごとに設定など動きをコントロール可能

使用方法

[設定概要] 使用時は、
解析DATに右の書き込み + サブルーチン
《正弦波変位の例》

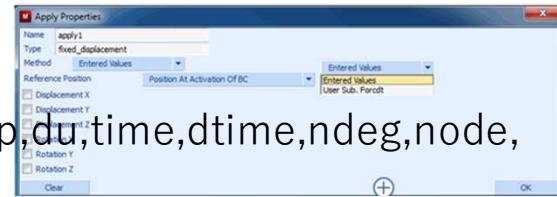
```
subroutine forcdt(u,v,a,dp,du,time,dtime,ndeg,node,  
    1ug,xord,ncrd,iacflg,inc)  
    implicit real*8 (a-h,o-z)  
C* * * * *  
c  
c      input of time dependent forcing functions and boundary  
c      conditions.  
c  
c* * * * *  
dimension u(3),v(3),a(3),dp(3),du(3)  
dimension ug(1),xord(1)  
dimension xfor(45)  
pi=3.141592  
f=.5  
hz=50  
omega=hz*2*pi  
du(1)=f*sin(omega*(dtime+time))-f*sin(omega*time)  
return  
end
```

節点番号指定、途中で節点変更も

モデル・ディイニション・カード

F O R C D T (変位又は荷重履歴の入力)

1. F O R C D T
2. 節点番号リスト,



subroutine forcdt(u,v,a,dp,du,time,dtime,ndeg,node,
転サブルーチン

```
1ug,xord,ncrd,iacflg,inc)  
implicit real*8 (a-h,o-z)
```

C* * * * *

dimension u(3),v(3),a(3),dp(3),du(3)

dimension ug(1),xord(3)

dimension xfor(45)

c rotetional speed omg(rad/inc)

omg=.0.17

rotational about o(a,b)

s=0

t=0 : 中心o(a,b) として設定

if(inc.eq.0) then

du(1)=0

du(2)=0

if(inc.eq.0) then

du(1)=0

du(2)=0

du(3)=0

end if

if(inc.ge.1) then

du(1)=cos(omg)*((

1 +sin(omg)*((

2) FORCEM：荷重条件を自在に・・

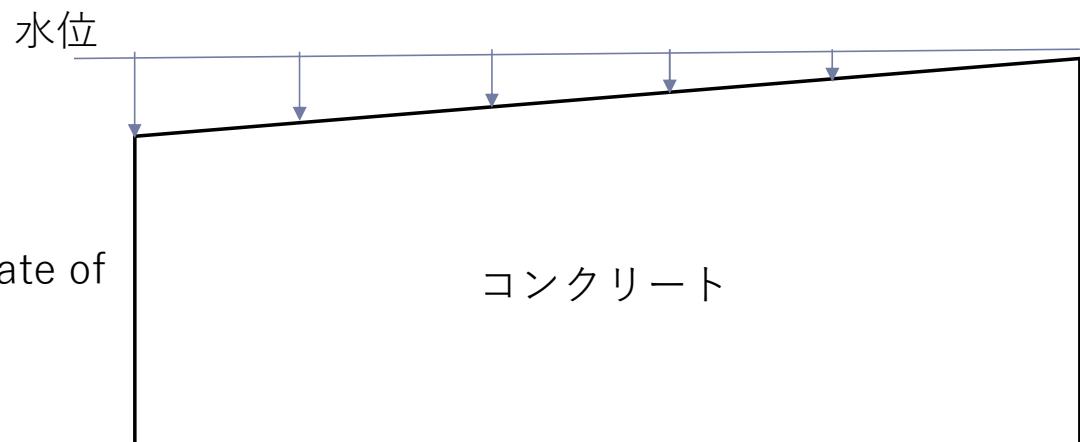
《サブルーチン》

```
subroutine forcem(press,x1,x2,nn,n)
c      called by: consb, consc, consd, conse,
c                  consf, consg, consh,
c* * * * *
c      nn      integration point number
c      n       element number
c* * * * *
dimension x1(3),x2(3),n(2)
cc
c      define pressure as proportional to second coordinate of
c      integration point
cc
ratio=x1(2)/2.0
value=20.0
press=value*ratio
write(6,9999) n(1),nn,x1(2),press
9999 format(2i5,2f10,4)
return
end
```

forcemの場合

D I S T L O A D S

4 (荷重タイプ), 0 (荷重), ⇒ 5 (4 + 1), 0,
(必ずゼロを入力 ↗)
1 T O 5 (要素番号)



FOLLOW FOR (パラメータ・カド)による与える荷重値の違い
・ある場合 : 荷重は増分ステップ終了時の全荷重
・ない場合 : 荷重は増分値 (auto load で繰り返す)

残念ながらほとんど使ったことなし。

3) PLOTV : MARCでのひずみは工学ひずみの出力がありません。 出力を演算でコントロール、工学ひずみも・・

《サブルーチン》

```

subroutine plotv(v,s,sp,etot,eplas,ecreep,t,m,nn,layer,ndi
c 各方向成分のひずみと主ひずみの公称値を計算する
cc TOTAL LAGRANGE使用時：グリーンラグランジェひずみ-->公称ひずみ
include '../common/implicit'
dimension s(*),etot(*),eplas(*),ecreep(*),sp(*),m(2)
dimension pv(3),r(3,3)c
v_max7 = 0.0d0
v_min8 = 0.0d0
v_max9 = 0.0d0
v_min10= 0.0d0c
call PRINCV (pv,r,etot,ndi,nshear,1,0,0,0)c
if (JPLTCD.eq.1) thenc X軸方向公称ひずみ
(TOTALLAGRANGE使用時)
v=-1+sqrt(2*(etot(1)+0.5)) ****
*****
```

モデル・ディファイン・カート

c

TOTAL LAGRANGE使用時：グリーンラグランジェひずみ-->公称ひずみ

include '../common/implicit'

dimension

s(*),

etot(*),

eplas(*),

ecreep(*),

sp(*),

m(2)

dimension

pv(3),

r(3,3)c

v_max7 =

0.0d0

v_min8 =

0.0d0

v_max9 =

0.0d0

v_min10=

0.0d0c

call

PRINCV

(pv,r,

etot,

ndi,

nshear,

1,0,0,0)c

if

(JPLTCD.eq.1)

then

c

X軸方向公称ひずみ

(TOTALLAGRANGE使用時)

v=-1+

sqrt(2*(etot(1)+0.5))

POST (ポスト・テープ)

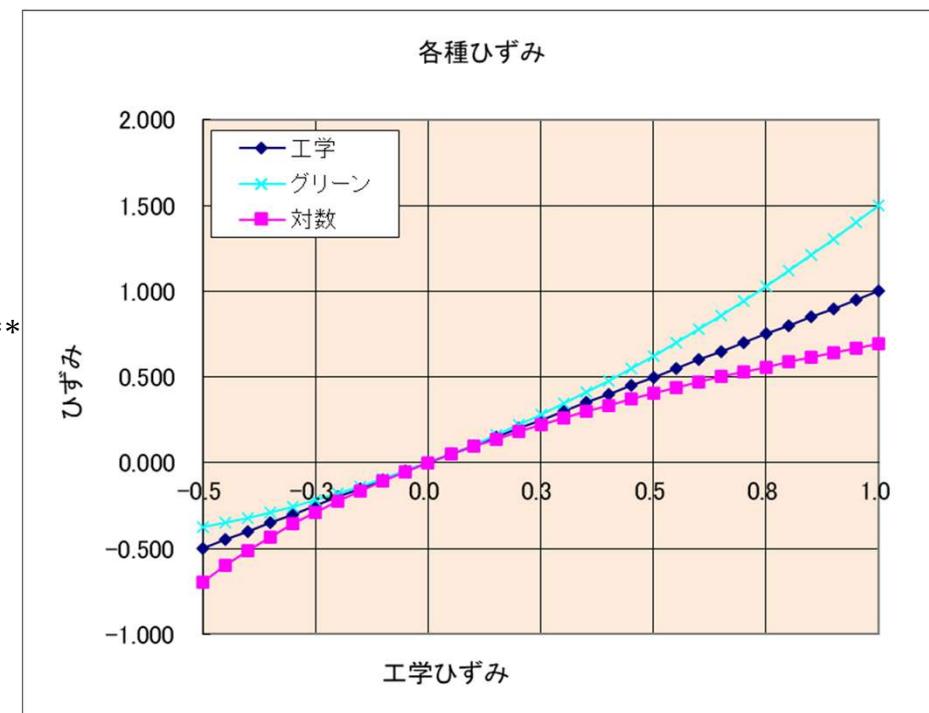
1. POST

2. ポスト・コードの数,

3. ポスト・コード

4.

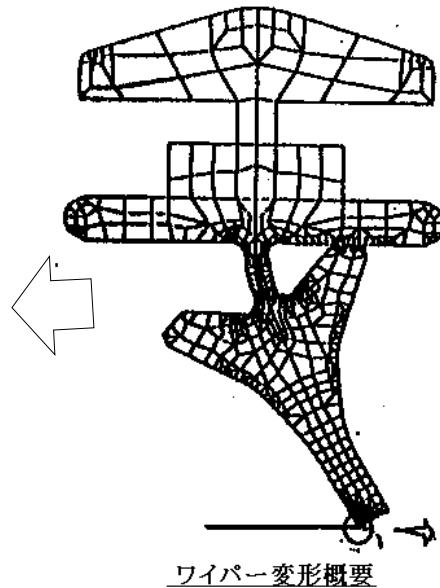
PLOTV 使用時のポスト・コードは、負のコードとなる



4) UFRIC：摩擦をコントロール、節点圧力から面圧依存性も・・・

《サブルーチン》

```
subroutine  
ufric(mibody,c,fn,vrel,temp,yiel,fric,time,inc,nsurf)  
    implicit real*8 (a-h,o-z)           dp  
    dimension mibody(1),c(1),vrel(1)  
    fric=.1  
    return  
end
```



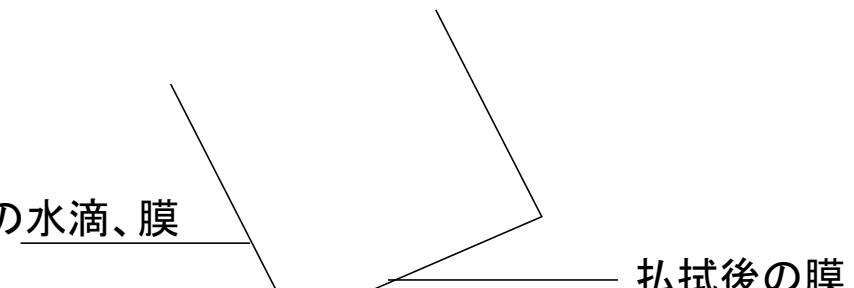
モデル・ディフィニション・カード

U F R I C T I O N

(サブルーチン U F R I C の呼び出し)

1. U F R I C

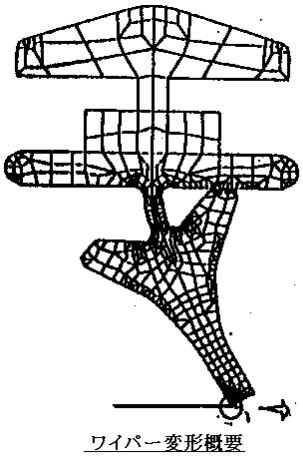
雨などの水滴、膜



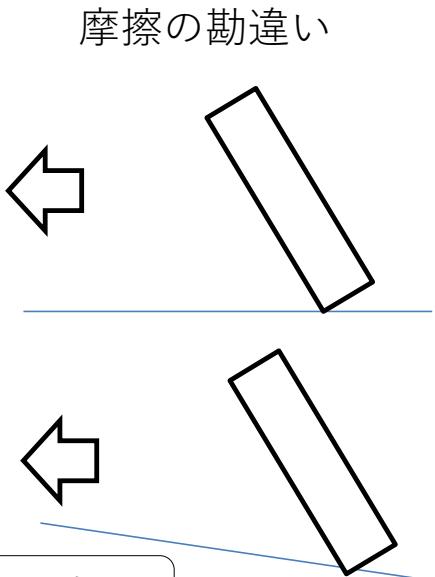
マイクロ単位の
メッシュ
(個体含めて)

ミクロで見るかマクロでみるか

摩擦係数に関する考察



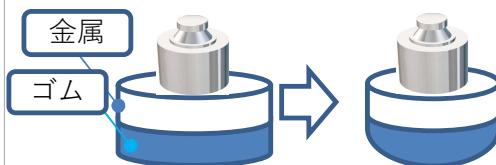
ミクロ的には落とし穴へ



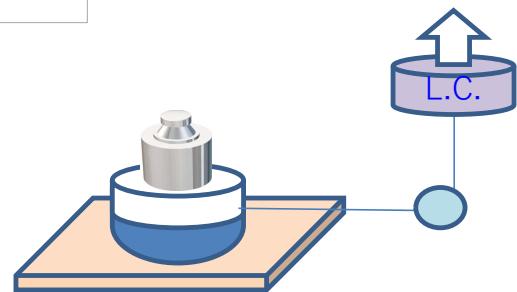
ゴムの場合、この傾きが微小でも追随してしまい、金属が滑るところを密着する。

ミクロ的な摩擦とマクロ的な摩擦は、解析をする際にしっかり区別、2以上の摩擦係数も現実的にある。

摩擦測定の例



正確に面圧依存性まで確認できないため、球状の試験片が良好



※解析により、もしくは理論解での面圧を確認可能

実験と解析により摩擦の面圧依存、関連してデータ蓄積可能

金属側

金属の場合
表面を滑る

ゴム側

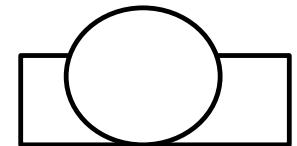
ゴムの場合
表面が密着し易い

摩擦がないと転がり続けると言うが・・・

ボール（球）をカーブした器、平面でも同様に
下図のような解析は、初期に安定しない。



曲面上のボール



平面上のボール

5) MOTION (2・3次元剛体壁の運動の定義)

```

title k4-1 contact analysis : with user sub
sizing,1000000,
elements 11
large disp
update
finite
print,5,
end
connectivity

coordinates
1 0.00000+0 0.00000+0 0.00000+0
property
1
100,0.3,,,10,
1,
fixed disp
2,
0,
1,
geometry
1,
1,1,
1,
umotion
contact
2,10,10,1,
,0.01,

```

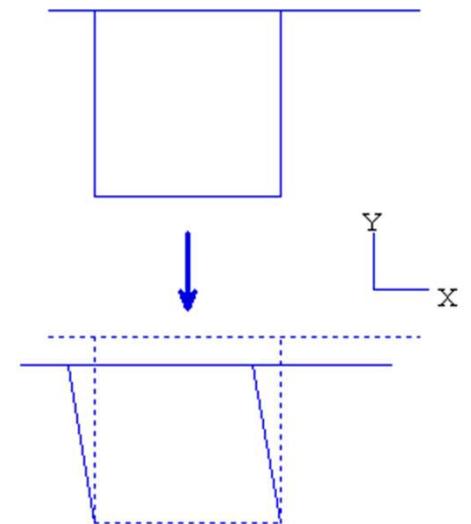
モデル・ディファイン・カード 1. UMOTION : 宣言

《サブルーチン》

```

subroutine motion(x,f,v,time,dtime,nsurf,inc)
dimension x(1),v(1),f(1),
c
10 v(1)=0
if(nsurf.ne.2)goto
c
if(inc.le.2) v(2)=-1
if(inc.gt.2) v(2)=1
c
write(6,200) nsurf v
200 format(' nsurf, v , ',15,se15.5)
99 return
End

```



これだけなら剛体制御のみで十分、関数での設定や、MENTATで不可能な中心変更など、どこかと関連付けなど。