

5.10 塑性論アナロジーモデル

5.10.1 アナロジーモデルの概要

本節では、塑性論アナロジーモデルについて解説する。なお、以後アナロジーモデルと呼ぶ。このアナロジーモデルの理論に関しては理論マニュアルを参照されたい。

アナロジーモデルにおける接線剛性は前節のファイバーモデルと同様に求められ、以下の式で与えられる。

$$[K_T] = [K_L] + [K_G] + [K_N] \quad \dots\dots(5.16)$$

最初に、増分弾塑性剛性 $[K_L]$ は、塑性論を応用した手法を用いて求めている。その結果、剛性行列 $[K_L]$ は、以下のよう求められる。

$$[K_L] = \begin{bmatrix} k_x - \frac{\alpha^2}{\omega} & 0 & 0 & 0 & -\frac{\alpha\beta}{\omega} & -\frac{\alpha\gamma}{\omega} & -\left(k_x - \frac{\alpha^2}{\omega}\right) & 0 & 0 & 0 & \frac{\alpha\beta}{\omega} & \frac{\alpha\gamma}{\omega} \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & k_{\theta y} - \frac{\beta^2}{\omega} & -\frac{\beta\gamma}{\omega} & \frac{\alpha\beta}{\omega} & 0 & 0 & 0 & -\left(k_{\theta y} - \frac{\beta^2}{\omega}\right) & \frac{\beta\gamma}{\omega} \\ & & & & & k_{\theta z} - \frac{\gamma^2}{\omega} & \frac{\alpha\gamma}{\omega} & 0 & 0 & 0 & \frac{\beta\gamma}{\omega} & -\left(k_{\theta z} - \frac{\gamma^2}{\omega}\right) \\ & & & & & & k_x - \frac{\alpha^2}{\omega} & 0 & 0 & 0 & -\frac{\alpha\beta}{\omega} & -\frac{\alpha\gamma}{\omega} \\ & & & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & 0 & 0 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & k_{\theta y} - \frac{\beta^2}{\omega} & -\frac{\beta\gamma}{\omega} \\ & & & & & & & & & & & k_{\theta z} - \frac{\gamma^2}{\omega} \end{bmatrix}$$

.....(5.17)

SPACE では、現在降伏条件として3つ用意しているが、ここでは、次の降伏条件について示し、プログラムについてもこの関数について説明する。他の降伏条件に関するプログラムは付録を参照されたい。

$$f = \left(\frac{N}{N_p}\right)^2 + \sqrt{\left(\frac{M_y}{M_{yp}}\right)^2 + \left(\frac{M_z}{M_{zp}}\right)^2} - 1 \quad \dots\dots(5.18)$$

この降伏条件に関する関数の微分関数は、応力を独立変数として、以下のように得られる。以後、降伏条件を表す式を降伏関数と呼ぶ。

$$\left. \begin{aligned} f_n &= \frac{2N}{N_p^2} \\ f_{my} &= -\frac{1}{\sqrt{\left(\frac{M_y}{M_{yp}}\right)^2 + \left(\frac{M_z}{M_{zp}}\right)^2}} \frac{M_y}{M_{yp}^2} \\ f_{mz} &= -\frac{1}{\sqrt{\left(\frac{M_y}{M_{yp}}\right)^2 + \left(\frac{M_z}{M_{zp}}\right)^2}} \frac{M_z}{M_{zp}^2} \end{aligned} \right\} \dots\dots(5.19)$$

なお、式(5.17)の剛性行列の中で使用しているパラメータは以下のようである。

$$\left. \begin{aligned} \alpha &= k_x f_n; \quad \beta = k_{\theta y} f_{my}; \quad \gamma = k_{\theta z} f_{mz} \\ \omega &= k_x f_n^2 + k_{\theta y} f_{my}^2 + k_{\theta z} f_{mz}^2 \end{aligned} \right\} \dots\dots(5.20)$$

上記の降伏関数には、曲げモーメントが0の点、つまり正と負の降伏軸力を示す位置の2箇所で、接線勾配に不連続点が存在する。この不連続性は収束計算を行うとき、反復時に解が振動して収束しない原因となる。そこで、この2箇所の近傍で、降伏関数を式(5.21)、(5.22)に示す関数に変更し、その不連続性を取り除くことにする。以下に示す式の誘導や内容の詳細は、参考文献4を参照されたい。

修正した2箇所での降伏関数として、次式のようにA点を中心とする球で表す。また、応力を独立変数とした降伏関数の微分関数も次式のように与えられる。

$$\left. \begin{aligned} \frac{N}{N_p} &< N_B \\ f &= \frac{1}{2M_B} \left\{ \left(\frac{N}{N_p} - a \right)^2 + \left(\frac{M_y}{M_{yp}} \right)^2 + \left(\frac{M_z}{M_{zp}} \right)^2 \right\} - \frac{r_0^2}{2M_B} = 0 \quad \dots\dots(5.21) \\ f_n &= \frac{1}{M_B} \left(\frac{N}{N_p} - a \right) \frac{1}{N_p} \\ f_{my} &= \frac{1}{M_B} \left(\frac{M_y}{M_{yp}} \right) \frac{1}{M_{yp}} \\ f_{mz} &= \frac{1}{M_B} \left(\frac{M_z}{M_{zp}} \right) \frac{1}{M_{zp}} \end{aligned} \right\} \dots\dots(5.21a)$$

$$\frac{N}{N_P} < -N_B$$

$$f = \frac{1}{2M_B} \left\{ \left(\frac{N}{N_P} + a \right)^2 + \left(\frac{M_y}{M_{yP}} \right)^2 + \left(\frac{M_z}{M_{zP}} \right)^2 \right\} - \frac{r_0^2}{2M_B} = 0 \quad \dots\dots(5.22)$$

$$\left. \begin{aligned} f_n &= \frac{1}{M_B} \left(\frac{N}{N_P} + a \right) \frac{1}{N_P} \\ f_{my} &= \frac{1}{M_B} \left(\frac{M_y}{M_{yP}} \right) \frac{1}{M_{yP}} \\ f_{mz} &= \frac{1}{M_B} \left(\frac{M_z}{M_{zP}} \right) \frac{1}{M_{zP}} \end{aligned} \right\} \dots\dots(5.22b)$$

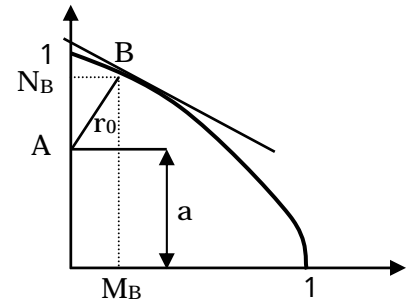


図 5-4 降伏関数の修正図

降伏関数に含まれているパラメータ N_B, M_B, r_0, a は、図 5-4 を参照されたい。同図で、B 点は 2 つの降伏関数の接点であり、A 点は、B 点での接線に垂直な線分が N 軸に交差する点である。この点が新たな降伏関数である球の中心点となる。ここで、無次元軸力 N_B を独立変数とすると、後のパラメータは次のように与えられる。

$$\left. \begin{aligned} M_B &= 1 - N_B^2 \\ a &= N_B(1 - 2M_B) \\ r_0 &= M_B \sqrt{1 + 4N_B^2} \end{aligned} \right\} \dots\dots(5.23)$$

SPACE では、現在 N_B を 0.95 としており、式(5.23)を用いると、他のパラメータは $M_B = 0.0975$, $a = 0.76475$, $r_0 = 0.209341$ となり、この値はプログラム内に Data 文として組み込まれている。

次に、アナロジーモデルにおける幾何学的非線形性を考えなければならないが、実はアナロジーモデルでは、そのエレメントに長さという概念がないので、ここでは幾何学的非線形性を考慮する必要はない。

5.10.2 アナロジーモデルをコールするサブルーチン

これから解説する両端アナロジーモデルが、以下の図で示されている。部材は 6 分割され、内部には 5 節点を有する。

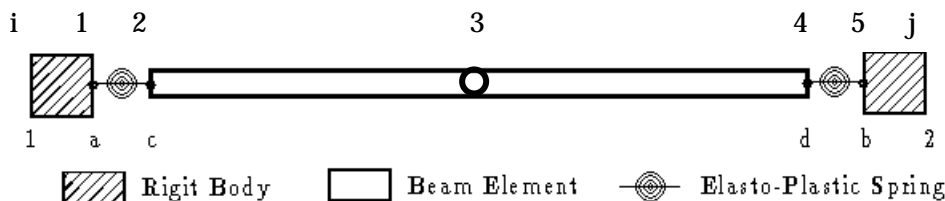


図 5-5 両端アナロジーモデルの部材解析モデル

静的縮合モデルである両端アナロジー部材に関するサブルーチンコールを以下に示す。これらのサブルーチンの中で、上記の接線剛性行列が作成されることになる。剛性を計算する上で、このアナロジーエレメントでは、長さがゼロであることに注意されたい。

```

c                                     Model_No.16 両端アナロジーモデル
      call Cal_lin_stiff_M31(Model_type,Member(i),Element(ie),
*      ak_linear(1,1,i) ,E_model131, E_model_fiber,
*      M_model131, M_model_fiber)

c                                     Model_No.16 両端アナロジーモデル
      call Cal_check_stiff_M31(N_analysis,
*      mem,Model_type,Member(i),Element(ie),
*      E_model131, E_model_fiber,
*      M_model131, M_model_fiber,
*      vv,vpp,f)

c                                     Model_No.16 両端、中央アナロジーモデル
      call Cal_nonlin_stiff_M31(N_analysis,
*      Model_type,Member(i),Element(ie),
*      ak ,ier,E_model131, E_model_fiber,
*      M_model131 , M_model_fiber)

```

上記の3つのサブルーチンの内容を以下に示す。これらは、前節で既に説明したファイバーモデルと内容はほぼ同じであり、理解することは難しくない。

```

C
C      SUBROUTINE /Cal_lin_stiff_M31
C
C      代表的な部材モデルの剛性(ok)
C
      subroutine Cal_lin_stiff_M31(Model_type,Member,Element,ak,
*      E_model131, E_model_fiber,
*      M_model131, M_model_fiber)
C
      implicit real*8(A-H,O-Z)
      include "submain.h"
      include "submainx.h"
      record / member_s      / Member
      record / element_s     / Element
      record / n_model_s     / Model_type
      record / E_model131_s  / E_model131
      record / E_model_analogy_s / E_model_fiber
      record / M_model131_s   / M_model131
      record / M_model_fiber_s / M_model_fiber
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension E_model131(*),M_model131(*)
      dimension ak(12,12),akk(12,12)
      real*8, ALLOCATABLE :: c(:,,:),ab(:,,:),ba(:,,:),alength(:)

```

```

integer, ALLOCATABLE :: irest_Point(:,:), n_type(:)

C
  iet = Member.n_model                ! モデルタイプ番号
  n_div = Model_type.n_div_model(iet) ! 部材分割数
  imm = Element.n_element             ! 要素番号
  immm = Member.n_model_type          ! モデルタイプ別番号
  n_if = 6*(n_div-1)                  ! 内部自由度
C   write(76, '(a,4i6)') ' model:', iet, n_div, imm, immm
C                                     動的記憶領域の確保
  ALLOCATE (
*   irest_Point(6,n_div+1), n_type(n_div), alength(n_div)
*   )
C                                     節点拘束表の作成
C                                     未知数等をセット
  call set_model31_dat(irest_Point, n_if, n_div, iubw,
*   Element, Member,
*   n_type, alength,
*   Member.i_rigid_length, ! i 端剛域
*   Member.j_rigid_length, ! j 端剛域
*   Member.i_shear_G,      ! i 端せん断剛性
*   Member.j_shear_G)      ! j 端せん断剛性
C                                     動的記憶領域の確保
  ALLOCATE (
*   c(0:iubw, n_if), ab(n_if, 12), ba(n_if)
*   )
C                                     剛性行列のゼロクリア
  do i=1, 12
  do j=1, 12
  ak(j, i)=0.
  enddo
  enddo
  do i=1, n_if
  do j=0, iubw
  c(j, i)=0.
  enddo
  enddo
  do i=1, 12
  do j=1, n_if
  ab(j, i)=0.
  enddo
  enddo
  do i=1, n_div
  Member.an_stress(i)=0.
  enddo
  do i=1, n_div
  Member.an_vv(i)=0.
  Member.an_wv(i)=0.
  enddo
C                                     部材剛性行列の作成
  it = 0
  do i=1, n_div
  call Stiff_M31_I(i, it, n_type(i), akk, Member, alength,
*   Model_type, Element,
*   E_model31(imm), E_model_fiber,

```

```

*      M_model31(imm), M_model_fiber)
call Bnd_FEM(i,akk,irest_Point,ak,c,ab,iubw,n_if)
enddo

C      部材剛性行列の縮合
call Typical_member_model(c,ab,ak, n_if,n_if,iubw,iubw,ba,ier)

C      両端の剛域処理
call Deal_Rigid_element(ak,Member.i_rigid_length,
*      Member.j_rigid_length)

C      動的記憶領域の解放
DEALLOCATE (
*      c ,ab ,ba,irest_point,n_type,alength
*      )
return
end

C
C      SUBROUTINE /Cal_nonlin_stiff_M31
C
C      代表的な部材モデルの接線剛性(ok)
C
subroutine Cal_nonlin_stiff_M31(N_analysis,
*      Model_type,Member,Element, ak,ier,
*      E_model31, E_model_fiber, M_model31, M_model_fiber)
C
implicit real*8(A-H,O-Z)
include "submain.h"
include "submainx.h"
record / member_s      / Member
record / element_s     / Element
record / n_model_s     / Model_type
record / E_model31_s    / E_model31
record / E_model_fiber_s / E_model_fiber
record / M_model31_s    / M_model31
record / M_model_fiber_s / M_model_fiber
dimension E_model_fiber(*),M_model_fiber(*)
dimension E_model31(*),M_model31(*)
dimension ak(12,12),akk(12,12)
real*8, ALLOCATABLE :: c(:,,:),ab(:,,:),ba(:),alength(:)
integer, ALLOCATABLE :: irest_Point(:,,:),n_type(:)

C
iet = Member.n_model      ! モデルタイプ番号
n_div = Model_type.n_div_model(iet) ! 部材分割数
imm= Element.n_element    ! 要素番号
immm= Member.n_model_type ! モデルタイプ別番号
n_if = 6*(n_div-1)        ! 内部自由度

C      動的記憶領域の確保
ALLOCATE (
*      irest_Point(6,n_div+1),n_type(n_div),alength(n_div)
*      )

C      節点拘束表の作成
C      未知数等をセット
call set_model31_dat(irest_Point,n_if,n_div,iubw,
*      Element,Member,
*      n_type,alength,

```

```

* Member.i_rigid_length,      ! i 端剛域
* Member.j_rigid_length,      ! j 端剛域
* Member.i_shear_G,           ! i 端せん断剛性
* Member.j_shear_G            ! j 端せん断剛性

c                                     動的記憶領域の確保
  ALLOCATE (
*   c(0:iubw,n_if),ab(n_if,12),ba(n_if )
*   )

c                                     剛性行列のゼロクリア
  do i=1,12
  do j=1,12
  ak(j,i)=0.
  enddo
  enddo
  do i=1,n_if
  do j=0,iubw
  c(j,i)=0.
  enddo
  enddo
  do i=1,12
  do j=1,n_if
  ab(j,i)=0.
  enddo
  enddo

c                                     部材剛性行列の作成
  it = 0
  EA=Element.A*Element.E
  do i=1,n_div
  call Stiff_M31(i,it,n_type(i),akk,Member,alength,
*   Model_type,Element,
*   E_model31(imm), E_model_fiber,
*   M_model31(imm), M_model_fiber)
  if(N_analysis.ne.9.and.N_analysis.ne.7.and.n_type(i).eq.1)then      ! 幾何学的非線形剛性
  call Cal_geomet_stiffx(Member.an_stress(i),Member,akk,alength(i))
  call Create_Kn(akk,Member.an_vv(i),Member.an_ww(i),
*   EA,alength(i))
  endif
  call Bnd_FEM(i,akk,irest_Point,ak,c,ab,iubw,n_if)
  enddo

c                                     部材剛性行列の縮合
  call Typical_member_model(c,ab,ak, n_if,n_if,iubw,iubw,ba,ier)

c                                     両端の剛域処理
  call Deal_Rigid_element(ak,Member.i_rigid_length,
*   Member.j_rigid_length)

c                                     動的記憶領域の解放
  DEALLOCATE (
*   c ,ab ,ba,irest_point,n_type,alength
*   )
  return
end

c
c   SUBROUTINE /Cal_check_stiff_M31
c
c   代表的な部材モデルの塑性チェック(ok)

```

```

C
    subroutine Cal_check_stiff_M31(N_analysis,
*      mem_x,Model_type,Member,Element,
*      E_model31, E_model_fiber,
*      M_model31, M_model_fiber,
*      vv,vp,f_p)
C
    implicit real*8(A-H,O-Z)
    include "submain.h"
    include "submainx.h"
    record / member_s      / Member
    record / element_s     / Element
    record / n_model_s     / Model_type
    record / E_model31_s   / E_model31
    record / E_model_analogy_s / E_model_fiber
    record / M_model31_s   / M_model31
    record / M_model_fiber_s / M_model_fiber
    dimension E_model_fiber(*),M_model_fiber(*)
    dimension E_model31(*),M_model31(*)
    dimension ak(12,12),f_p(12)
    dimension vv(12),vp(12),vvx(12)
    real*8, ALLOCATABLE :: c(:,,:),ab(:,,:),alength(:)
    real*8, ALLOCATABLE :: bav(:),f1(:),akk(:,,:,:)
    integer, ALLOCATABLE :: irest_Point(:,,:),n_type(:)
C
    iet = Member.n_model                ! モデルタイプ番号
    n_div = Model_type.n_div_model(iet) ! 部材分割数
    imm= Element.n_element              ! 要素番号
    immm= Member.n_model_type           ! モデルタイプ別番号
    n_if = 6*(n_div-1)                  ! 内部自由度
C
C
C      部材の剛性行列の設定
C
C
C
C      動的記憶領域の確保
    ALLOCATE (
*      irest_Point(6,n_div+1),n_type(n_div),alength(n_div),
*      akk(12,12,n_div)
*      )
C
C      節点拘束表の作成
C      未知数等をセット
    call set_model31_dat(irest_Point,n_if,n_div,iubw,
*      Element,Member,
*      n_type,alength,
*      Member.i_rigid_length,    ! i 端剛域
*      Member.j_rigid_length,    ! j 端剛域
*      Member.i_shear_G,        ! i 端せん断剛性
*      Member.j_shear_G)        ! j 端せん断剛性
C
C      動的記憶領域の確保
    ALLOCATE (
*      c(0:iubw,n_if),ab(n_if,12),
*      bav(n_if),f1(n_if)
*      )

```



```

c                                     剛性行列のゼロクリア
do i=1,12
do j=1,12
ak(j,i)=0.
enddo
enddo
do i=1,n_if
do j=0,iubw
c(j,i)=0.
enddo
enddo
do i=1,12
do j=1,n_if
ab(j,i)=0.
enddo
enddo
do i=1,12
f_p(i)=0.
enddo
do i=1,n_if
f1(i)=0.
enddo

c                                     部材剛性行列の作成

it = 0
EA=Element.A*Element.E
do i=1,n_div

c                                     内部部材の剛性行列の計算
call Stiff_M31(i,it,n_type(i),akk(1,1,i),Member,alength,
*      Model_type,Element,
*      E_model31(imm), E_model_fiber,
*      M_model31(imm), M_model_fiber)
if(N_analysis.ne.9.and.N_analysis.ne.7.and.n_type(i).eq.1)then    ! 幾何学的非線形剛性
call Cal_geomet_stiffx(Member.an_stress(i),Member,
*      akk(1,1,i),alength(i))
call Create_Kn(akk(1,1,i),Member.an_vv(i),Member.an_wv(i),
*      EA,alength(i))
endif

c                                     剛性行列の分配
call Bnd_FEM(i,akk(1,1,i),irest_Point,ak,c,ab,iubw,n_if)
enddo

c
c
c                                     部材内部の変位と不釣合力の計算
c
c
c
c                                     両端変位を剛域内部の変位に変換処理
call Deal_Rigid_element_v(vv,Member.i_rigid_length,
*      Member.j_rigid_length)
c                                     部材内部変位の計算(c 行列の分解計算)
call Typical_member_v(c,ab,f1,
*      n_if,n_if,iubw,iubw,ier,vv,bav)
c                                     部材内部、両端節点力の計算

it = 0
do i=1,n_div

```

```

        if(n_type(i).eq.2) then
            it=it+1
            if(it.eq.1) then
                nnm=M_model31(imm).nm_section_1
            else
                nnm=M_model31(imm).nm_section_2
            endif
        endif
        call Analogy_member_p_force(akk(1,1,i),irest_Point(1,i),
*                               vv,bav,f_p,f1,
*                               n_type(i),M_model31(imm),M_model_fiber,nnm)
        enddo
c                               部材両端節点力への縮合
        call Typical_member_f(c,ab,f1,f_p,n_if,n_if,iubw,iubw) ! f1 はデータが変更される
c                               部材節点力の両端剛域処理
        call Deal_Rigid_element_f(f_p,Member.i_rigid_length,
*                               Member.j_rigid_length)
c
c
c                               部材の弾塑性チェック
c
c
c                               部材内部応力のチェック
        it = 0
        do i=1,n_div
c                               変位の取りだし
            ii=0
            do j=1,2
            do k=1,6
                ii=ii+1
                irest=irest_Point(k,i+j-1)
                if(irest.lt.0) then
                    vx(ii)=vv(-irest)
                elseif(irest.gt.0) then
                    vx(ii)=bav(irest)
                else
                    vx(ii)=0.
                endif
            enddo
        enddo
c                               アナロジー要素チェック
        if(n_type(i).eq.2) then
            it=it+1
            call Analogy_check31(N_analysis,
*            mem_x,it,alength(i),Member,Element,
*            E_model31(imm),E_model_fiber,M_model31(imm),M_model_fiber,
*            vx,Member.an_stress(i))
        endif
c                               弾性部材の軸力計算（幾何剛性作成用）
        if(n_type(i).eq.1) then
            call nonlinear_stress_N(akk(1,1,i),vx,fnn)
            Member.an_stress(i)=Member.an_stress(i)+fnn
c                               部材内部変位を足しこむ
            Member.an_vv(i)=Member.an_vv(i)+(vx(8) - vx(2))

```

```

      Member.an_wv(i)=Member.an_wv(i)+(vvx(9) - vvx(3))
    endif
  enddo

C                                     動的記憶領域の解放
      DEALLOCATE (
*      c ,ab ,irest_point,n_type,alength,
*      bav,f1,akk      )
      return
    end

```

この3つのサブルーチンの解析フローは、一部のサブルーチンを除いて、前節で解説した両端ファイバーモデルと同じである。その中身は、同じサブルーチンを用いている場合と、名称は異なるがその内容はほとんど同じである場合とがある。例えば、解析モデルの未知数番号表を作成するサブルーチン `set_model31_dat()` は、`set_model11_dat()` とほとんど同じとなっている。プログラムの内容は付録を参照されたい。

5.10.3 アナロジー モデルの剛性

本節では、接線剛性行列のひとつである $[K_L]$ を求めるサブルーチン `Stiff_M31()` について説明する。まず、このサブルーチンを以下に示す。

```

C
C      SUBROUTINE /Stiff_M31
C
C      接線剛性行列の計算(ok)
C
      subroutine Stiff_M31(im,it,n_type,ak,Member,alength,
*      Model_type,Element,
*      E_model31, E_model_fiber,M_model31, M_model_fiber)
C
      implicit real*8(A-H,O-Z)
      include "submain.h"
      include "submainx.h"
      record / member_s      / Member
      record / element_s     / Element
      record / n_model_s     / Model_type
      record / E_model31_s   / E_model31
      record / M_model31_s   / M_model31
      record / M_model_fiber_s / M_model_fiber
      record / E_model_fiber_s / E_model_fiber
      dimension ak(12,12),alength(*)
      dimension vv(12)
      dimension E_model_fiber(*),M_model_fiber(*)

C                                     要素及びモデルのセット
      do i=1,12
      do j=1,12
      ak(j,i)=0.
      enddo
      enddo

```

```

        goto(11,12,13,14,15,16,17,18,19,20),n_type
11 continue
c
    call Cal_nonlin_stiff_Mx(Member,Element,
    *      Member.stress(7),ak,alength(im) )
    goto 100
12 continue
c
    it=it+1
    if(it.eq.1)then
        nn=E_model31.nm_section_1
        nnm=M_model31.nm_section_1
        iep=M_model_fiber(nnm).i_elastic
    else
        nn=E_model31.nm_section_2
        nnm=M_model31.nm_section_2
        iep=M_model_fiber(nnm).i_elastic
    endif
    call Analogy_Model_G31(it,iep,ak,Member,Element,
    *      E_model31,E_model_fiber,
    *      M_model31,M_model_fiber,nn)
    goto 100
13 continue
c
    call Shear_G(it,ak,Member)
    goto 100
14 continue
c
    goto 100
15 continue
c
    goto 100
16 continue
c
    goto 100
17 continue
c
    goto 100
18 continue
c
    goto 100
19 continue
c
    goto 100
20 continue
c
100 continue
    return
end

```

弾性要素

アナロジー要素

せん断バネ要素

ダミー

ダミー

ダミー

ダミー

ダミー

ダミー

ダミー

このサブルーチン Stiff_M31() も、やはりサブルーチン Stiff_M11() と同様の階層構造を持っており、内容は Stiff_M11() とほとんど同じである。そこではエレメント型番号：2 のモデルがアナロジーモデルとな

っており、このモデルの剛性行列は、サブルーチン `Analogy_Model_G31()` を使用して求めている。

サブルーチン `Stiff_M31()` は、剛性行列を作成するプログラムであり、その構造は非常に単純である。エレメント型番号 `n_type` をパラメータにして、部材内のエレメントモデルが階層構造となっている。

プログラムは、最初、剛性行列 `ak` をゼロクリアする。次に、エレメント型番号によって処理が分類されており、現在は 1 が弾性エレメント、2 がアナロジーエレメント、3 がせん断ばねエレメントとなっている。各エレメント型番号にしたがってサブルーチンがコールされ、剛性行列 `ak` が計算される。弾性要素の剛性を計算するサブルーチン `Cal_nonlin_stiff_Mx()` は、表示は非線形となっているが、実際は、第 5.2 節で説明した線形の弾性剛性を求めている。非線形剛性 $[K_G]$ と $[K_N]$ は、このサブルーチンをコールした後に、次のサブルーチンコールで求めている。

```

if(N_analysis.ne.9.and.N_analysis.ne.7.and.n_type(i).eq.1)then      ! 幾何学的非線形剛性
call Cal_geomet_stiffx(Member.an_stress(i),Member,akk,alength(i))
call Create_Kn(akk,Member.an_vv(i),Member.an_ww(i),
*           EA,alength(i))
endif

```

上記の非線形剛性を求めるサブルーチンについては、第 5.3 節で説明した。ただし、アナロジーエレメントは長さが 0 であるため、幾何学的非線形剛性は存在しない。

次に、以下に示す弾塑性剛性行列に関連する 3 つのサブルーチン `Analogy_Model_G131()` と `Analogy_Model_G31()`、及び `Analogy_GK()` について説明する。最初のサブルーチンは塑性関数に関する初期設定であり、各種のアナロジーモデルのためのデータを構造体に設定する。次のサブルーチンは弾塑性行列を求めるための各種のデータを、構造体から取得するルーチンである。最後のサブルーチンは実際に剛性行列を計算するプログラムである。以下に、3 つのサブルーチンの内容を具体的に示すことにする。

```

C
C      SUBROUTINE /Analogy_Model_G1
C
C      線形剛性行列の計算(ok)
C
C      Model_No.31 両端 Analogy モデル
C
      subroutine Analogy_Model_G131(it,ak,Member,Element,

```

```

*      E_model,E_model_analogy,M_model,M_model_analogy)
implicit real*8(A-H,O-Z)
include "submain.h"
include "submainx.h"
record / member_s      / Member
record / element_s     / Element
record / E_model31_s   / E_model
record / E_model_analogy_s / E_model_analogy
record / M_model31_s   / M_model
record / M_model_fiber_s / M_model_analogy
dimension E_model_analogy(*),M_model_analogy(*)
dimension ak(12,12)

c
  if(it.eq.1) then                                ! 1
  do i=1,30                                       ! 2
    M_model.ff_ip(i)=0.
  enddo
  nm_div=E_model.n_section_1                      ! 3
  nn=E_model.nm_section_1 - 1
  nnm=M_model.nm_section_1 - 1
  elseif(it.eq.2) then
  nm_div=E_model.n_section_2                      ! 4
  nn=E_model.nm_section_2 - 1
  nnm=M_model.nm_section_2 - 1
  endif
  do i=1,nm_div                                  ! 5
  nn=nn+1                                         ! 6
  nnm=nnm+1

c
  M_model_analogy(nnm).i_elastic =0              ! 7
  M_model_analogy(nnm).d_eps_x =0.               ! 要素の状態（弾性の場合は0：塑性は1）
  M_model_analogy(nnm).d_stress_x =0.            ! 変位
  M_model_analogy(nnm).d_E = 0                   ! 応力
  M_model_analogy(nnm).d_E = 0                   ! 降伏関数の値をセット
c
  データセット
  nm_x=M_model_analogy(nnm).n_type
  nm_type=E_model_analogy(nm).nm_type-10
  goto ( 10,20,30),nm_type                       ! 8
10 continue
c
  goto 100
20 continue
c
  goto 100
30 continue
c
  goto 100
100 continue
  enddo
  if(it.eq.1) then                                ! 9
  M_model.fax_1 = 0.
  M_model.fay_1 = 0.
  M_model.faz_1 = 0.
  nnm=E_model.nm_section_1
  else

```

```

M_model.fax_2 = 0.
M_model.fay_2 = 0.
M_model.faz_2 = 0.
nnm=E_model.nm_section_2
endif
C                                     初期剛性行列の計算
C
C     iep=0
C     call Analogy_Model_G31(it,iep,ak,Member,Element,          ! 10
*       E_model,E_model_analogy,M_model,M_model_analogy,nnm)
C     return
C     end
C
C     SUBROUTINE /Analogy_Model_G31
C
C     接線剛性行列の計算(ok)
C
C     subroutine Analogy_Model_G31(it,iep,ak,Member,Element,
*       E_model,E_model_analogy,M_model,M_model_analogy,itt)
C     implicit real*8(A-H,O-Z)
C     include "submain.h"
C     include "submainx.h"
C     record / member_s      / Member
C     record / element_s     / Element
C     record / E_model31_s   / E_model
C     record / M_model31_s   / M_model
C     record / M_model_fiber_s / M_model_analogy
C     record / E_model_analogy_s / E_model_analogy
C     dimension E_model_analogy(*),M_model_analogy(*)
C     dimension ak(12,12)
C
C     if(it.eq.1) then          ! 11
C     ra = E_model_analogy(itt).AK_1      ! ばねの軸剛性
C     ray = E_model_analogy(itt+1).AK_1   ! z 軸のばねの曲げ剛性
C     raz = E_model_analogy(itt+2).AK_1   ! y 軸のばねの曲げ剛性
C     fax = M_model.fax_1                ! ポテンシャル関数の x 軸微分
C     fay = M_model.fay_1                ! ポテンシャル関数の y 軸微分
C     faz = M_model.faz_1                ! ポテンシャル関数の z 軸微分
C     else
C     ra = E_model_analogy(itt).AK_1      ! ばねの軸剛性
C     ray = E_model_analogy(itt+1).AK_1   ! z 軸のばねの曲げ剛性
C     raz = E_model_analogy(itt+2).AK_1   ! y 軸のばねの曲げ剛性
C     fax = M_model.fax_2                ! ポテンシャル関数の x 軸微分
C     fay = M_model.fay_2                ! ポテンシャル関数の y 軸微分
C     faz = M_model.faz_2                ! ポテンシャル関数の z 軸微分
C     endif
C     ra_x= E_model_analogy(itt).AK_2     ! ばねの最低軸剛性
C     ray_x=E_model_analogy(itt+1).AK_2   ! ばねの最低曲げ剛性
C     raz_x=E_model_analogy(itt+2).AK_2   ! ばねの最低曲げ剛性
C     call Analogy_GK(iep,ak,ra,ray,raz,fax,fay,faz,          ! 12
*       ra_x,ray_x,raz_x)
C     return
C     end
C
C     SUBROUTINE /Analogy_GK

```

```

C
C      アナロジー要素の剛性行列
C
      subroutine Analogy_GK(id,ak,ra,riy,riz,fax,fay,faz,
*          ra_x,ray_x,raz_x)
      implicit real*8(A-H,O-Z)
      dimension ak(12,12)
      if(id.eq.0) then
C
C          線形剛性
C
          ak(1,1)  = ra
          ak(1,7)  = -ra
          ak(7,1)  = -ra
          ak(7,7)  = ra
          ak(5,5)  = riy
          ak(5,11) = -riy
          ak(11,5) = -riy
          ak(11,11) = riy
          ak(6,6)  = riz
          ak(6,12) = -riz
          ak(12,6) = -riz
          ak(12,12) = riz
          else
C
C          線形剛性
C
          a=ra*fax
          b=riy*fay
          c=riz*faz
          h=a*fax+b*fay+c*faz
          if(h.eq.0.) then
C
C          線形剛性
C
              ak(1,1)  = ra
              ak(1,7)  = -ra
              ak(7,1)  = -ra
              ak(7,7)  = ra
              ak(5,5)  = riy
              ak(5,11) = -riy
              ak(11,5) = -riy
              ak(11,11) = riy
              ak(6,6)  = riz
              ak(6,12) = -riz
              ak(12,6) = -riz
              ak(12,12) = riz
              else
C
C          非線形剛性
C
                  hh=1./h
                  ha=a*a*hh
                  hb=b*b*hh
                  hc=c*c*hh
                  hab=a*b*hh
                  hac=a*c*hh
                  hbc=b*c*hh
                  C          剛性が任意閾値より小さくなるのを防ぐ処理
                  raha = ra    ha
                  if( raha .lt.ra_x)  raha=ra_x
                  riyhb = riy - hb
                  if(riyhb .lt.ray_x) riyhb=ray_x

```



```

    rizhc = riz - hc
    if(rizhc .lt. raz_x) rizhc = raz_x
c      直線部材で3ヒンジ不安定になるのを避けるため
    if(hab.ne.0.) hab=hab*1.00001      ! 18
    if(hac.ne.0.) hac=hac*1.00001
    if(hbc.ne.0.) hbc=hbc*1.00001
c      直線部材で3ヒンジ不安定になるのを避けるため
    ak(1,1) =  raha      ! 19
    ak(1,7) = -raha
    ak(7,1) = -raha
    ak(7,7) =  raha
    ak(1,5) = -hab
    ak(5,1) = -hab
    ak(1,11)=  hab
    ak(11,1)=  hab
    ak(5,7) =  hab
    ak(7,5) =  hab
    ak(7,11)= -hab
    ak(11,7)= -hab
    ak(1,12)= hac
    ak(12,1)= hac
    ak(1,6) =-hac
    ak(6,1) =-hac
    ak(6,7) =  hac
    ak(7,6) =  hac
    ak(7,12)=-hac
    ak(12,7)=-hac
    ak(5,6) = -hbc
    ak(6,5) = -hbc
    ak(5,12) =  hbc
    ak(12,5) =  hbc
    ak(6,11) =  hbc
    ak(11,6) =  hbc
    ak(11,12)= -hbc
    ak(12,11)= -hbc

    ak(5,5) =  riymb
    ak(5,11) = -riymb
    ak(11,5) = -riymb
    ak(11,11) = riymb
    ak(6,6) =  rizhc
    ak(6,12) = -rizhc
    ak(12,6) = -rizhc
    ak(12,12) = rizhc
    endif
    endif
    return
end

```

上記の3つのサブルーチンについて、右側のコメント番号に従ってプログラムの内容を説明する。ここで使用しているアナロジーモデルのばねに関するデータ領域はファイバーモデルのそれと共用しており、その

ため同じ構造体を使用している。

1. 両端アナロジーモデルでは、i 端と j 端でアナロジーばねデータの管理位置が異なる。そのため、ここでどちらかをチェックする。パラメータ it が 1 の場合は i 端を意味し、2 は j 端を意味する。
2. 内部節点の不釣合力の保存場所である $M_model_ff_ip$ をゼロクリアする。
3. 構造体より i 端のアナロジーばねデータの管理用パラメータを取得し、その断面におけるアナロジーばね数を nm_div に、アナロジーのばねの要素に関連するデータの最初の配列番号 1 を nn に、同じく、部材に関連するデータの最初の配列番号 1 を nnm にセットする。
4. 上記と同様に、j 端に関するアナロジーばねデータの管理用パラメータを取得する。
5. 以下の処理を、その断面の全アナロジーばね数分行う。ただし、ばね数は軸方向ばね、y 軸回りの曲げばね、z 軸回りの曲げばねの 3 つである。
6. 構造体の配列番号に 1 を加え、次のアナロジーばねデータの配列番号をセットする。
7. 入力データ以外の各アナロジーばねデータの初期設定を行う。まず、部材に関連し、解析途中で変更するデータとして、アナロジーばねエレメントの弾塑性状態を表すパラメータ、軸方向ひずみ、軸方向応力、降伏関数の値にゼロをセットする。
8. アナロジーばねの降伏条件の型番号によって処理内容が分類されており、その番号にしたがって各プログラム位置に処理が移動する。ただし、現在は、完全弾塑性型を使用しているのみであるから、ここでは初期設定を行っていない。
9. 降伏関数の微分値を i 端と j 端に分けてゼロセットする。
10. この断面を要するエレメントの接線剛性行列のひとつである行列 $[K_L]$ を、サブルーチン $Analogy_Model_G31()$ で求める。
11. サブルーチン $Analogy_Model_G31()$ では、まず、この部材のどちらの端部かをチェックし、該当する端部断面に関する各パラメータを構造体より取得する。
12. この取得したパラメータを用いて、実際の剛性行列 ak をサブルーチン $Analogy_GK()$ を用いて計算する。
13. ここでは、アナロジーばねエレメントの剛性行列を計算する。塑性

状態のアナロジーばねエレメントの剛性行列を求めるコードと、式(5.17)の行列とを比較して、その内容を確認されたい。また、この式の誘導などについては理論マニュアルを参照されたい。最初に、エレメントが弾性状態であるか塑性状態であるか、パラメータ id を用いてチェックする。弾性状態である場合は線形の剛性行列を作成する。

14. 塑性状態のアナロジーばねエレメントの剛性行列を計算する。ここでは、降伏関数の微分値が必要となる。
15. 式(5.17)の中で、 σ の値がゼロである場合、ゼロ割り算となるので、この場合は、以降の処理で線形の剛性行列をセットする。
16. 上記の σ がゼロでない場合、式(5.20)で示す各係数を計算する。
17. エレメントの剛性行列の対角項がゼロとなるのを防ぐために、行列対角項がある閾値より小さくならないようにする。
18. 直線部材で3ヒンジの不安定部材となると、剛性が特異行列となり、計算不能状態となる。そこで、軸方向剛性と2つの曲げ剛性の関連部分の剛性に非常に小さな値をつけ加え、剛性行列が特異とならないようにする。
19. 最終的に設定した各係数を用いて剛性行列の該当する部分に係数をセットする。

最後に残った処理はアナロジーばねの弾塑性チェックである。この弾塑性チェックを行うサブルーチン `Analogy_check31()` をコールするコードは、サブルーチン `Cal_check_stiff_M31()` の中で、以下のように記述されている。

```

c                                     アナロジーばねチェック
      if(n_type(i).eq.2) then
        it=it+1
        call Analogy_check31(N_analysis,
*         mem_x,it,alength(i),Member,Element,
*         E_model31(imm),E_model_fiber,M_model31(imm),M_model_fiber,
*         vx,Member.an_stress(i))
      endif

```

サブルーチン `Analogy_check31()` の内容を以下に示す。

```

C
C      SUBROUTINE /Analogy_check31
C

```

5.10.4 アナロジー モデルの弾塑 性チェック

```

C      アナロジー要素の材料非線形性チェックし、応力を計算
C
      subroutine Analogy_check31(N_analysis,
*      mem_x,it,Alength,Member,Element,
*      E_model,E_model_analogy,M_model,M_model_analogy,vv,An_f)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      include "submainx.h"
      record / member_s      / Member
      record / element_s     / Element
      record / E_model31_s    / E_model
      record / E_model_analogy_s / E_model_analogy
      record / M_model31_s    / M_model
      record / M_model_fiber_s / M_model_analogy
      dimension E_model_analogy(*),M_model_analogy(*)
      dimension vv(12)

C                                     i 端部
C                                     変位のセット
      if(it.eq.1) then
d_epsilon_x_1 = (vv(7) - vv(1))  ! 増分軸方向変位
d_epsilon_y_1 = (vv(11) - vv(5)) ! 増分 y 軸に関する回転角
d_epsilon_z_1 = (vv(12) - vv(6)) ! 増分 z 軸に関する回転角
      if(Member.analysis_3D .eq. 1) d_epsilon_z_1 = 0. ! 平面問題における面外方向の曲げ変形を無視する
      if(Member.analysis_3D .eq. 2) d_epsilon_y_1 = 0. ! 平面問題における面外方向の曲げ変形を無視する
      M_model.d_epsilon_x_1 = d_epsilon_x_1 + M_model.d_epsilon_x_1 ! 軸方向変位
      M_model.d_epsilon_y_1 = d_epsilon_y_1 + M_model.d_epsilon_y_1 ! y 軸に関する回転角
      M_model.d_epsilon_z_1 = d_epsilon_z_1 + M_model.d_epsilon_z_1 ! z 軸に関する回転角

C                                     アナロジー要素のチェック
      nm_div=E_model.n_section_1
      nem=E_model.nm_section_1
      nnm=M_model.nm_section_1
      nm_x=M_model_analogy(nnm).n_type
      nm_type=E_model_analogy(nem).nm_type

C                                     各要素の増分歪セット
      M_model_analogy(nnm).d_eps_x = d_epsilon_x_1
      M_model_analogy(nnm+1).d_eps_x = d_epsilon_y_1
      M_model_analogy(nnm+2).d_eps_x = d_epsilon_z_1
      if(N_analysis.le.8.or.
*      Member.nm_analysis.eq.-1) then
C                                     弾性解析
      else
C                                     弾塑性解析
      goto ( 10,20,30),nm_type-10
10  continue

C                                     型
      call Check_Analogy_M1(E_model_analogy,nem,
*      M_model_analogy,nnm,Member.d_stat(1),
*      M_model.fax_1,M_model.fay_1,M_model.faz_1,i_hosei)
C                                     塑性ポテンシャルの微分計算
      call Cal_div_potential_M1(E_model_analogy,nem,
*      M_model_analogy,nnm,
*      M_model.fax_1,M_model.fay_1,M_model.faz_1,i_hosei)
      goto 100
20  continue

```

```

c                                     型                                     ! 9
    call Check_Analogy_M2(E_model_analogy,nem,
*       M_model_analogy,nnm,Member.d_stat(1),
*       M_model.fax_1,M_model.fay_1,M_model.faz_1)
c                                     塑性ポテンシャルの微分計算
    call Cal_div_potential_M2(E_model_analogy,nem,
*       M_model_analogy,nnm,
*       M_model.fax_1,M_model.fay_1,M_model.faz_1)
    goto 100
30 continue

c                                     型                                     ! 10
    call Check_Analogy_M3(E_model_analogy,nem,
*       M_model_analogy,nnm,Member.d_stat(1),
*       M_model.fax_1,M_model.fay_1,M_model.faz_1)
c                                     塑性ポテンシャルの微分計算
    call Cal_div_potential_M3(E_model_analogy,nem,
*       M_model_analogy,nnm,
*       M_model.fax_1,M_model.fay_1,M_model.faz_1)
    goto 100
endif
100 continue

c                                     要素応力の計算
c                                     j 端部
c                                     変位のセット

    elseif(it.eq.2) then                                     ! 11
        d_epsilon_x_2 = (vv(7) - vv(1))    ! 軸方向変位
        d_epsilon_y_2 = (vv(11) - vv(5))   ! y 軸に関する回転角
        d_epsilon_z_2 = (vv(12) - vv(6))   ! z 軸に関する回転角
        if(Member.analysis_3D .eq. 1) d_epsilon_z_2 = 0. ! 平面問題における面外方向の曲げ変形を無視する
        if(Member.analysis_3D .eq. 2) d_epsilon_y_2 = 0. ! 平面問題における面外方向の曲げ変形を無視する
        M_model.d_epsilon_x_2 = d_epsilon_x_2+M_model.d_epsilon_x_2    ! 軸方向変位
        M_model.d_epsilon_y_2 = d_epsilon_y_2+M_model.d_epsilon_y_2    ! y 軸に関する回転角
        M_model.d_epsilon_z_2 = d_epsilon_z_2+M_model.d_epsilon_z_2    ! z 軸に関する回転角
c                                     アナロジー要素のチェック
        nm_div=E_model.n_section_2
        nem=E_model.nm_section_2
        nnm=M_model.nm_section_2
        nm_x=M_model_analogy(nnm).n_type
        nm_type=E_model_analogy(nem).nm_type
c                                     各要素の増分歪セット
        M_model_analogy(nnm).d_eps_x = d_epsilon_x_2
        M_model_analogy(nnm+1).d_eps_x = d_epsilon_y_2
        M_model_analogy(nnm+2).d_eps_x = d_epsilon_z_2
        if(N_analysis.le.8.or.
*       Member.nm_analysis.eq.-1) then
c                                     弾性解析
        else
c                                     弾塑性解析
        goto ( 11,21,31),nm_type-10
11 continue

c                                     型
    call Check_Analogy_M1(E_model_analogy,nem,
*       M_model_analogy,nnm,Member.d_stat(2),
*       M_model.fax_2,M_model.fay_2,M_model.faz_2,i_hosei)

```

```

c                                     塑性ポテンシャルの微分計算
  call Cal_div_potential_M1(E_model_analogy,nem,
*                               M_model_analogy,nnm,
*                               M_model.fax_2,M_model.fay_2,M_model.faz_2,i_hosei)
  goto 101
21 continue

c                                     型
  call Check_Analogy_M2(E_model_analogy,nem,
*                               M_model_analogy,nnm,Member.d_stat(2),
*                               M_model.fax_2,M_model.fay_2,M_model.faz_2)
c                                     塑性ポテンシャルの微分計算
  call Cal_div_potential_M2(E_model_analogy,nem,
*                               M_model_analogy,nnm,
*                               M_model.fax_2,M_model.fay_2,M_model.faz_2)
  goto 101
31 continue

c                                     型
  call Check_Analogy_M3(E_model_analogy,nem,
*                               M_model_analogy,nnm,Member.d_stat(2),
*                               M_model.fax_2,M_model.fay_2,M_model.faz_2)
c                                     塑性ポテンシャルの微分計算
  call Cal_div_potential_M3(E_model_analogy,nem,
*                               M_model_analogy,nnm,
*                               M_model.fax_2,M_model.fay_2,M_model.faz_2)
  goto 101
  endif
101 continue

c                                     要素応力の計算
  endif
  return
end

```

上記のサブルーチンについて、右側のコメント番号に従ってプログラムの内容を説明する。

1. 両端アナロジーモデルでは、i 端と j 端でアナロジーデータの管理位置が異なる。そのため、ここでどちらであるかを調査する。パラメータ it が 1 の場合は i 端であり、2 は j 端である。
2. この断面の増分ひずみ、増分軸方向ひずみ、y 軸に関する曲げひずみ、z 軸に関する曲げひずみを求める。次に、解析が平面問題の場合は、面外方向の曲げひずみをゼロとする。最後に、各増分ひずみを増分前のひずみに足しこむ。
3. 断面内のアナロジーばねの数（アナロジーばねの数は 3 に固定されている）要素アナロジーばね用データの配列番号を nem に、部材アナロジーばね用データの配列番号を nnm にセットする。
4. アナロジーばねの増分ひずみをセットする。
5. 解析種別と部材の解析種別をチェックし、弾性である場合は何もせ

- ずにこのサブルーチンから戻る。
6. 降伏条件の型番号 nm_type によって処理を分類する。ただし、降伏条件の型番号は 11 番以降を用いている。
 7. 降伏条件の型番号 11 の型であり、降伏条件についてアナロジーばねの弾塑性チェックをサブルーチン Check_Analogy_M1()を用いて行う。
 8. 降伏関数の微分をサブルーチン Cal_div_potential_M1 を用いて、求める。
 9. ここでは降伏条件の型番号 12 の型であり、この降伏条件についてアナロジーばねの弾塑性チェックと降伏関数の微分を行う。
 10. ここでは降伏条件の型番号 13 の型であり、この降伏条件についてアナロジーばねの弾塑性チェックと降伏関数の微分を行う。
 11. これ以降は j 端における処理を行うが、ここでの処理は上記の i 端の処理と全く同一である。プログラムを一度検証されたい。

5.10.5 降伏関数の微分

最後に、このサブルーチンの中で使用されている降伏関数の微分を求めるサブルーチン Cal_div_potential_M1()と弾塑性チェックを行うサブルーチン Check_Analogy_M1()について示す。このサブルーチンは、降伏条件型番号に従ってその関数の微分を求める。以下にその内容を示す。

```

C
C      SUBROUTINE /Cal_div_potential_M1
C
C      アナロジー要素の塑性ポテンシャルの微分計算
C
C      subroutine Cal_div_potential_M1(E_model_analogy,nem,
*          M_model_analogy,nmm,ffx,ffz,i_hosei)
C
C      ファイバーモデル E_model_analogy_s 構造体
C
C
C      部材
C      structure / E_model_analogy_s/
C      integer nm_type          ! 履歴モデルの番号
C      real*8  AK_1              ! ばねの第一勾配
C      real*8  AK_2              ! ばねの第二勾配
C      real*8  AK_3              ! ばねの第三勾配
C      real*8  Q_1               ! 第一折れ点の応力
C      real*8  Q_2               ! 第二折れ点の応力
C      real*8  dm(16)           ! ダミー
C      end structure
C

```

```

C
C      ファイバーモデル M_model_analogy_s 構造体
C
C
C      部材
C      structure / M_model_analogy_s/
C      integer  n_type           ! 履歴モデルの通し番号
C      integer  i_elastic        ! ファイバー要素の状態（弾性の場合は0：塑性は1）
C      real*8   d_eps_x          ! 軸方向歪
C      real*8   d_stress_x       ! 軸方向応力
C      real*8   d_E              ! 接線剛性
C      end structure
C
      implicit real*8(A-H,O-Z)
      include "submainx.h"
      record / E_model_analogy_s / E_model_analogy
      record / M_model_fiber_s    / M_model_analogy
      dimension E_model_analogy(*),M_model_analogy(*)
      data x_Nb,x_Mb,x_a,x_r0/0.95,0.0975,0.76475,0.209341/

      if(M_model_analogy(nmm).i_elastic .eq.0 )then      ! (弾性の場合は0：塑性は1) 1
        ffx=0.
        ffy=0.
        ffz=0.
      else
        if(i_hosei.eq.1)then                                ! 2
C
C                                     f_N > x_Nb
          ffx=(M_model_analogy(nmm).d_stress_x/E_model_analogy(nem).Q_1-
*            x_a)/(E_model_analogy(nem).Q_1*x_Mb)
          ffy=M_model_analogy(nmm+1).d_stress_x/
*            (E_model_analogy(nem+1).Q_1**2*x_Mb)
          ffz=M_model_analogy(nmm+2).d_stress_x/
*            (E_model_analogy(nem+2).Q_1**2*x_Mb)
          elseif(i_hosei.eq.2)then                            ! 3
C
C                                     f_N < -x_Nb
          ffx=(M_model_analogy(nmm).d_stress_x/E_model_analogy(nem).Q_1+
*            x_a)/(E_model_analogy(nem).Q_1*x_Mb)
          ffy=M_model_analogy(nmm+1).d_stress_x/
*            (E_model_analogy(nem+1).Q_1**2*x_Mb)
          ffz=M_model_analogy(nmm+2).d_stress_x/
*            (E_model_analogy(nem+2).Q_1**2*x_Mb)
          else                                                ! 4
C
C                                     x_Nb > f_N > -x_Nb
          ffx=2.*M_model_analogy(nmm).d_stress_x/E_model_analogy(nem).Q_1**2
          ff=dsqrt(
*            (M_model_analogy(nmm+1).d_stress_x/E_model_analogy(nem+1).Q_1)**2+
*            (M_model_analogy(nmm+2).d_stress_x/E_model_analogy(nem+2).Q_1)**2
*            )
          if(ff.eq.0.) then                                    ! 5
            ffy=0.
            ffz=0.
          Else                                                ! 6
            ffy=M_model_analogy(nmm+1).d_stress_x/
*            (E_model_analogy(nem+1).Q_1**2*ff)

```



```

      ff=M_model_analogy(nmm+2).d_stress_x/
*      (E_model_analogy(nem+2).Q_1**2*ff)
    endif
  endif
  endif
  return
end

C
C      SUBROUTINE /Check_Analogy_M1
C
C      アナロジー要素の塑性チェック
C
      subroutine Check_Analogy_M1(E_model_analogy,nem,
*      M_model_analogy,nmm,iet,fax,fay,faz,i_hosei)
      implicit real*8(A-H,O-Z)
      include "submainx.h"
      record / E_model_analogy_s      / E_model_analogy
      record / M_model_fiber_s        / M_model_analogy

      dimension E_model_analogy(*),M_model_analogy(*)
      data x_Nb,x_Mb,x_a,x_r0/0.95,0.0975,0.76475,0.209341/

C      軸力チェック
      iett=M_model_analogy(nmm).i_elastic                      ! 7
      f_N=M_model_analogy(nmm).d_stress_x/E_model_analogy(nem).Q_1
      ff_past = M_model_analogy(nmm).d_E                      ! 降伏関数の値をセット
C      f_N > x_Nb 引張
      if(f_N.gt.x_Nb) then                                     ! 8
        i_hosei=1
C      降伏条件式
      ff=((f_N - x_a)**2 +                                     ! 9
*      (M_model_analogy(nmm+1).d_stress_x/E_model_analogy(nem+1).Q_1)**2+
*      (M_model_analogy(nmm+2).d_stress_x/E_model_analogy(nem+2).Q_1)**2-
*      x_r0**2)
      M_model_analogy(nmm).d_E = ff                          ! 降伏関数の値をセット      ! 10
      if(iett.eq.0) then                                     ! 11
C      弾性状態の場合
      if(ff.ge.0.) then                                       ! 12
        M_model_analogy(nmm).i_elastic=2
        write(76, '(a,3i4,8e12.4)') ' 塑性発生 ',
*      M_model_analogy(nmm).i_elastic,nem,nmm,ff,
*      M_model_analogy(nmm).d_stress_x,M_model_analogy(nmm+1).d_stress_x,
*      M_model_analogy(nmm+2).d_stress_x,E_model_analogy(nem).Q_1,
*      E_model_analogy(nem+1).Q_1,E_model_analogy(nem+2).Q_1
      endif
      else
C      部材が既に塑性の場合
      a=fax*E_model_analogy(nem).AK_1
      b=fay*E_model_analogy(nem+1).AK_1
      c=faz*E_model_analogy(nem+2).AK_1
      df=a*M_model_analogy(nmm).d_eps_x +                      ! 13
*      b*M_model_analogy(nmm+1).d_eps_x +
*      c*M_model_analogy(nmm+2).d_eps_x
      ddd=a*fax+b*fay+c*faz
      if(ddd.ne.0.) df=df/ddd

```

```

    if(ff.lt.0.or.df.lt.0.)M_model_analogy(nmm).i_elastic = 0
    if(abs(df).lt.0.0001) M_model_analogy(nmm).i_elastic = 2          ! 14
    iet=M_model_analogy(nmm).i_elastic
    if(iet.eq.0) then                                                ! 15
        write(76,'(a,i4,9e12.4)') ' 弾性復活',iet,
* ff,df,fax,fay,faz,M_model_analogy(nmm).d_eps_x,
* M_model_analogy(nmm+1).d_eps_x,
* M_model_analogy(nmm+2).d_eps_x
    else
        write(76,'(a,i4,9e12.4)') ' 塑性継続',iet,
* ff,df,fax,fay,faz,M_model_analogy(nmm).d_eps_x,
* M_model_analogy(nmm+1).d_eps_x,
* M_model_analogy(nmm+2).d_eps_x
    endif
endif

c                                f_N < -x_Nb 圧縮
elseif(f_N.lt. -x_Nb ) then          ! 16
    i_hosei=2

c                                降伏関数
    ff=((f_N + x_a)**2 +
* (M_model_analogy(nmm+1).d_stress_x/E_model_analogy(nem+1).Q_1)**2+
* (M_model_analogy(nmm+2).d_stress_x/E_model_analogy(nem+2).Q_1)**2-
* x_r0**2)
    M_model_analogy(nmm).d_E = ff      ! 降伏関数の値をセット
    if(iett.eq.0) then

c                                弾性状態の場合
        if(ff.ge.0.) then
            M_model_analogy(nmm).i_elastic=2
            write(76,'(a,3i4,8e12.4)') ' 塑性発生',
* M_model_analogy(nmm).i_elastic,nem,nmm,ff,
* M_model_analogy(nmm).d_stress_x,M_model_analogy(nmm+1).d_stress_x,
* M_model_analogy(nmm+2).d_stress_x,E_model_analogy(nem).Q_1,
* E_model_analogy(nem+1).Q_1,E_model_analogy(nem+2).Q_1
        endif
    else

c                                部材が既に塑性の場合
        a=fax*E_model_analogy(nem).AK_1
        b=fay*E_model_analogy(nem+1).AK_1
        c=faz*E_model_analogy(nem+2).AK_1
        df=a*M_model_analogy(nmm).d_eps_x +
* b*M_model_analogy(nmm+1).d_eps_x +
* c*M_model_analogy(nmm+2).d_eps_x
        ddd=a*fax+b*fay+c*faz
        if(ddd.ne.0.) df=df/ddd
        if(ff.lt.0.or.df.lt.0.)M_model_analogy(nmm).i_elastic = 0
        if(abs(df).lt.0.0001) M_model_analogy(nmm).i_elastic = 2
        iet=M_model_analogy(nmm).i_elastic
        if(iet.eq.0) then
            write(76,'(a,i4,9e12.4)') ' 弾性復活',iet,
* ff,df,fax,fay,faz,M_model_analogy(nmm).d_eps_x,
* M_model_analogy(nmm+1).d_eps_x,
* M_model_analogy(nmm+2).d_eps_x
        else
            write(76,'(a,i4,9e12.4)') ' 塑性継続',iet,

```

```

* ff,df,fax,fay,faz,M_model_analogy(nmm).d_eps_x,
* M_model_analogy(nmm+1).d_eps_x,
* M_model_analogy(nmm+2).d_eps_x
endif
endif

c                                x_Nb > f_N > -x_Nb                                ! 18
else
i_hosei=0

c                                降伏関数
ff=(M_model_analogy(nmm).d_stress_x/E_model_analogy(nem).Q_1)**2
* + dsqrt(
*(M_model_analogy(nmm+1).d_stress_x/E_model_analogy(nem+1).Q_1)**2+
*(M_model_analogy(nmm+2).d_stress_x/E_model_analogy(nem+2).Q_1)**2)
M_model_analogy(nmm).d_E = ff      ! 降伏関数の値をセット
c                                弾性状態の場合
if(iett.eq.0) then
if(ff.ge.1.) then
M_model_analogy(nmm).i_elastic=2
write(76,'(a,3i4,8e12.4)') ' 塑性発生',
*M_model_analogy(nmm).i_elastic,nem,nmm,ff,
*M_model_analogy(nmm).d_stress_x,M_model_analogy(nmm+1).d_stress_x,
*M_model_analogy(nmm+2).d_stress_x,E_model_analogy(nem).Q_1,
*E_model_analogy(nem+1).Q_1,E_model_analogy(nem+2).Q_1
endif
else
c                                部材が既に塑性の場合
a=fax*E_model_analogy(nem).AK_1
b=fay*E_model_analogy(nem+1).AK_1
c=faz*E_model_analogy(nem+2).AK_1
df=a*M_model_analogy(nmm).d_eps_x +
* b*M_model_analogy(nmm+1).d_eps_x +
* c*M_model_analogy(nmm+2).d_eps_x
ddd=a*fax+b*fay+c*faz
if(ddd.ne.0.) df=df/ddd
if(ff.lt.1..or.df.lt.0.) M_model_analogy(nmm).i_elastic = 0
if(abs(df).lt.0.00001) M_model_analogy(nmm).i_elastic = 2
iet=M_model_analogy(nmm).i_elastic if(iet.eq.0) then
write(76,'(a,i4,9e12.4)') ' 弾性復活',iet,
* ff,df,fax,fay,faz,M_model_analogy(nmm).d_eps_x,
* M_model_analogy(nmm+1).d_eps_x,
* M_model_analogy(nmm+2).d_eps_x
else
write(76,'(a,i4,9e12.4)') ' 塑性継続',iet,
* ff,df,fax,fay,faz,M_model_analogy(nmm).d_eps_x,
* M_model_analogy(nmm+1).d_eps_x,
* M_model_analogy(nmm+2).d_eps_x
endif
endif
endif
return
end

```

上記のサブルーチンについて、右側のコメント番号に従ってプログラ

ムの内容を説明する。

1. 降伏関数の微分を求めるサブルーチン Cal_div_potential_M1() では、式(5.19)にしたがって微係数を求める。ただし、部材が軸力のみの場合は、降伏関数に尖った点が生じる。これを避けるために、 $M=0$ で、 $\bar{N}=+N_b$ 、 $\bar{N}=-N_b$ の2点近傍において降伏関数を球で近似する。従って、軸力の大きさによって降伏関数の状態が3つに分類され、パラメータ i_hosei (0: $N_b > \bar{N} > -N_b$ 、1: $\bar{N} > N_b$ 、2: $-N_b > \bar{N}$) で見分ける。ここで、 N_b は、その断面の降伏軸力の95%の値としている。

まず、ここでは、そのアナロジーばねが弾性状態であるかどうかチェックする。弾性の場合は、微係数をゼロとする。

2. 以降の処理は塑性状態に関する処理である。ここでは、 $\bar{N} > N_b$ の場合で、各微係数を計算する。この領域の降伏関数と微係数は式(5.21)と(5.21a)を用いて求める。
3. 次は、 $-N_b > \bar{N}$ の場合であり、同様に微係数を計算する。この領域の降伏関数と微係数は、式(5.22)と(5.22a)を用いて求める。
4. 最後に、 $N_b > \bar{N} > -N_b$ の場合で、式(5.19)で表される微係数が計算される。
5. 係数 ff がゼロの場合、ゼロ割り算を生じるため、微係数 ffy と ffz をゼロとする。
6. 係数がゼロでない場合は、通常の微係数を計算する。これで微係数が求まったことになるのでこのサブルーチンから戻る。
7. このサブルーチン Check_Analogy_M1() では、アナロジーばねの弾塑性チェックを行う。最初に、このアナロジーばねの弾塑性状態を表すパラメータを構造体より iett に取得する。次に、次式で表す無次元軸力 \bar{N} (コード内では f_N である) を求める。

$$\bar{N} = \frac{N}{N_p}$$

8. この無次元軸力が、 $\bar{N} > N_b$ であるとき、領域を表すパラメータ i_hosei に1をセットする。
9. 降伏関数(5.18)の値を求める。
10. その値を構造体成分 M_model_analogy(nmm).d_E にセットする。
11. このアナロジーばねの現在の弾塑性状態をチェックし、弾性の場合以下の処理を行う。
12. 弾性の状態で降伏関数の値が正より大きい場合、塑性状態

になったことを示すので、塑性状態：2を、弾塑性状態を表す構造体 `M_model_analogy(nmm).i_elastic` にセットする。部材に塑性が発生したことを出力する。

13. バネが既に塑性状態で、さらに降伏関数の値が弾性復活したかチェックし、復活した場合は `M_model_analogy(nmm).i_elastic` に弾性状態を示す0をセットする。
14. ただし、その値の変化量が非常に小さい場合は、弾性復活を認めず、塑性状態とする。
15. 以上の処理を終了した後、部材が塑性状態のままか、弾性復活したかを出力する。
16. これ以降の処理は、 $-Nb > \bar{N}$ の場合について処理を行う。処理の流れは、上記とほとんど同じである。領域を表すパラメータ `i_hosei=2` にセットする。
17. 降伏関数の値を計算し、構造体にセットする。後の処理はほとんど同じである。
18. これ以降の処理は、 $Nb > \bar{N} > -Nb$ の場合について処理を行う。処理の流れは上記とほとんど同じである。領域を表すパラメータ `i_hosei` に0をセットする。