

5.11 Maxwell モデル

5.11.1 Maxwell モデルの概要

本節では、Maxwell モデルについて解説する。この Maxwell モデルを含むと、前節までの振動方程式を少し変形して用いなければならない。Maxwell モデルを含む変形した振動方程式は第 3.4 節を参照すると以下のようになる。

$$\begin{aligned} [M]\{\ddot{y}\} + [\bar{C}]\{\dot{y}\} + [K]\{y\} = & -[M][I]\{\ddot{u}_g\} \\ & + \{P_s\} - \{\bar{f}_d\} - \{Q(\bar{y})\} - [K_T(\bar{y})]\{\Delta y\} + [K]\{y\} \end{aligned} \quad \dots\dots (5.24)$$

Maxwell モデルに関連する項は、上式中で次の 2 項である。

$$\left. \begin{aligned} [\bar{C}] &= [C] + [C_0] \\ \{\bar{f}_d\} &= \{(\alpha_1 - \alpha_0)\dot{u}_{n+1} + b\bar{\gamma}f_n + \bar{\alpha}\dot{u}_n + \bar{\alpha}'f_0 + b\bar{\alpha}''f_0'\} \end{aligned} \right\} \dots\dots (5.25)$$

ここで、 $[C_0]$ は Maxwell モデルの線形減衰項であり、 $[C]$ はそれ以外の線形減衰項である。この方程式に *Newmark* 法を適用すれば、振動方程式は式(5.26)として得られる。

$$\begin{aligned} [M]\{\ddot{y}_{n+1}\} + [\bar{C}]\{\dot{y}_{n+1}\} + [K]\{y_{n+1}\} = & -[M][I]\{\ddot{u}_g\} + \{P_s\} - \{Q(y_n)\} - \{\bar{f}_d\} \\ & - [K_T(y_n)]\{b\} + \mu_2\{\ddot{y}_{n+1}\} + [K]\{\bar{b}\} + \mu_2\{\ddot{y}_{n+1}\} \end{aligned} \quad \dots\dots (5.26)$$

さらに、上式を整理すると、

$$\begin{aligned} [M] + \mu_1[\bar{C}] + \mu_2[K] \{\ddot{y}_{n+1}\} = & -[M][I]\{\ddot{u}_g\} + \{P_s\} - \{Q(y_n)\} - [\bar{C}]\{a\} - [K_T(y_n)]\{b\} \\ & - \{\bar{f}_d\} + \mu_2([K] - [K_T(y_n)])\{\ddot{y}_{n+1}\} \end{aligned} \quad \dots\dots (5.27)$$

となる。ここで、係数を

$$\left. \begin{aligned} [F] &= [M] + \mu_1[\bar{C}] + \mu_2[K] \\ \{G(\ddot{y}_{n+1})\} &= \mu_2([K] - [K_T(y_n)])\{\ddot{y}_{n+1}\} - \{\bar{f}_d\} \\ \{g\} &= -[M][I]\{\ddot{u}_g\} + \{P_s\} - \{Q(y_n)\} \\ &\quad - [\bar{C}]\{a\} - [K_T(y_n)]\{b\} \end{aligned} \right\} \dots\dots (5.28)$$

のように整理すると、方程式(5.27)は次式となる。

$$[F]\{\ddot{y}_{n+1}\} = \{G(\ddot{y}_{n+1})\} + \{g\} \quad \dots\dots (5.29)$$

式(5.29)から分かるように両辺に未知ベクトルである増分後の加速度ベクトル $\{\ddot{y}_{n+1}\}$ を含んでおり、そのため、この方程式は反復法を用いて解かれることになる。

以上を整理すると、Maxwell モデルに関連する項は、線形の減衰項に付加される項 $[C_0]$ と右辺反復項 $\{\bar{f}_d\}$ である。節点力と節点変位の関係は、第3.4節中の式(3.26)、(3.43)より次のように得られる。

$$\begin{Bmatrix} p_i \\ p_j \end{Bmatrix} = \begin{bmatrix} \alpha & -\alpha \\ -\alpha & \alpha \end{bmatrix} \begin{Bmatrix} \dot{u}_i \\ \dot{u}_j \end{Bmatrix} + \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} (\alpha \dot{u}_n + b\gamma f_n + \alpha' f_0 + b\alpha' f_0') \quad \dots\dots (5.30)$$

ここで、各係数は、

$$\alpha = \frac{ck\Delta t}{2c+k\Delta t}; \alpha' = \frac{k\Delta t}{2c+k\Delta t}; \gamma = \frac{2c-k\Delta t}{2c+k\Delta t} \quad \dots\dots (5.31)$$

である。また、非線形性を考慮すると下に示すように各種の係数が必要となる。この係数に関する詳細は、理論マニュアルを参照されたい。

$$\left. \begin{aligned} \alpha_0 &= \frac{c_0 k \Delta t}{2c_0 + k \Delta t} & \gamma_0 &= \frac{2c_0 - k \Delta t}{2c_0 + k \Delta t} \\ \alpha_1 &= \frac{c_1 k \Delta t}{2c_1 + k \Delta t} & \gamma_1 &= \frac{2c_1 - k \Delta t}{2c_1 + k \Delta t} \\ \alpha'_1 &= \frac{k \Delta t}{2c_1 + k \Delta t} & f_0 &= f_1 (1 - c_1 / c_0) \end{aligned} \right\} \dots\dots (5.32)$$

SPACE では、パッシブ型とセミアクティブ型の制震装置に対応する Maxwell モデルが組み込まれている。しかしながら、セミアクティブ型は複雑なコードを含むため、ここでは説明を省く。これについては、他のマニュアルに譲ることとし、ここではパッシブ型のみ説明する。このパッシブ型においても限界速度があり、この速度を超えると速度に比例するダッシュポットの減衰定数は変化する。このような現象に対応するため、SPACE ではダッシュポットの減衰定数として、バイリニア型が組み込まれている。

Maxwell モデルに関連するサブルーチンは以下に示す 10 である。カッコ内のサブルーチンはこの Maxwell モデルに関連するサブルーチンをコールするサブルーチンである。

- | | | | |
|---------------|-------------------------------|---|--------------------|
| 1 . 初期設定 : | Initset_maxwelldamp | : | (Set_model_data) |
| 2 . 線形剛性 : | Cal_link_maxwelldamp | : | (Cal_stiff_linear) |
| 3 . 線形減衰 : | Cal_lin_maxwelldamp | : | (Cal_damp_linear) |
| 4 . 非線形減衰 : | Cal_nonlin_maxwelldamp (陰解法用) | : | (Add_damp3_ld_ex) |
| 5 . 部材端力の計算 : | Cal_force_maxwelldamp (反復解法用) | : | (Add_fdd_ld) |

| | | |
|--------------|-------------------------------|--------------------------|
| | Cal_forcef_maxwelldamp (陰解法用) | : (Add_fdd_ld_ex) |
| 6 . 応力計算 : | Stress_maxwelldamp | : (Cal_stress) |
| 7 . 履歴チェック : | Check_Maxwell_stress | : (submain_dynamic_a) |
| | Check_maxwelldamp | : (Check_Maxwell_stress) |
| | cal_maxwelldamp | : (Check_maxwelldamp) |

5.11.2 初期設定

Maxwell モデルでは、先に示したように入力したデータから多くの係数を計算する必要がある。これを初期設定プログラムで行う。初期設定プログラム Set_initial_data() は、submain_dynamic_a() でサブルーチンコールされる。

```

c                                     モデルの初期設定
      call Set_initial_data(Element,Member,Parameter_C,Newmark_P,
*          E_model6_real,Model_type)

```

この初期設定サブプログラムは、多くの部材モデルに対応するように、階層構造となっているが、現在では、Maxwell モデルの初期設定のみ行い、他のモデルの初期設定は行っていない。他のモデルは線形の剛性を計算するサブルーチンで行っている。

```

c
c      SUBROUTINE /Set_initial_data
c
c      各モデルの初期設定(各部材モデルで初期設定が必要な場合はここを使用)
c
      subroutine Set_initial_data()
      do i=1,n_member
      iet = Member(i).element_type
      goto(11,12,13,14,15,16,17,18,19,20), iet
      .
      .
16 continue
c                                     Model_No.6 3次元制震 Maxwell モデル
      call Initset_maxwelldamp(Element(iet),Newmark_P,
*          E_model6_real(iet),Model_type.n_m_filter)
      .
      .
      end do
      return
      end

```

上記の Maxwell モデルに関する初期設定用サブルーチンを以下に示す。ここでは、Maxwell モデルの基本的な係数を計算し、その値を構造体にセットする。また、計算過程で必要となるワーク領域もゼロクリアする。

```

C
C      Maxwell model に関するサブルーチン群
C
C      制震用 3 次元モデル (パッシブ型、セミアクティブ型)
C
C      要素数 (モデル No.6 Maxwell モデルに対する構造体)
C      structure / element_s6/
C      integer element_type      ! 要素タイプ(6)
C      integer n_element        ! 非線形要素番号
C      real*8 damp_type         ! 0.:パッシブ型、1.:セミアクティブ型
C      real*8 AK                ! 剛性
C      real*8 C0                ! 第一減衰定数
C      real*8 C1                ! 第二減衰定数
C      real*8 C2                ! セミアクティブ用減衰定数
C      real*8 F0                ! リリース応力
C      real*8 Udmax             ! リリース速度
C      real*8 F0x               ! セミアクティブ型で、減衰定数変更応力
C      real*8 Dm1               ! ダミー
C      real*8 Dm2               ! ダミー
C      integer nm_damp          ! 部材減衰の有無(1)
C      integer nm_type          ! (maxwell モデルでは、1 : x 方向 2 : y 方向 3 : z 方向)
C      end structure
C      record /element_s6/ Element
C      end structure
C
C
C      制震用 3 次元モデル (モデル No.6 Maxwell モデルに対する構造体)
C
C
C      部材非線形データ
C
C      structure / e_model6_real_s/
C      real*8 alph_0            ! 0
C      real*8 gumma_0           ! 0
C      real*8 alph_d0           ! _d0
C      real*8 alph_1            ! 1
C      real*8 gumma_1           ! 1
C      real*8 alph_d1           ! _d1
C      real*8 alph_2            ! 2
C      real*8 gumma_2           ! 2
C      real*8 alph_d2           ! _d2
C      real*8 alph_3            ! 3
C      real*8 gumma_3           ! 3
C      real*8 alph_d3           ! _d3
C      real*8 f00               ! f0
C      real*8 alph              !
C      real*8 gumma             !
C      real*8 f01               ! f0'
C      real*8 f0                ! f0
C      real*8 alfd              ! '
C      real*8 cx                ! 現在の減衰定数 c
C      real*8 fn                ! 現在の応力
C      real*8 u1                ! ダッシュポットの変位
C      real*8 u2                ! ばねの変位

```

```

c      real*8  u1d                ! ダッシュポットの速度
c      real*8  uudd               ! モデルの現在の速度
c      real*8  uud                ! フィルター通過後のダッシュポット速度
c      real*8  uudx               ! フィルター通過後の t 前のダッシュポット速度
c      integer jact               ! 0:パッシブ型、1:セミアクティブ型
c      integer istat              ! 現在の状態
c      real*8  cx_lag             ! t の減衰定数の変化量
c      integer max_istat_lag      ! 減衰定数の変化時の最大個数
c      integer istat_lag         ! 減衰定数の変化時の個数
c      real*8  work(41)          ! フィルター用ワークエリア
c      end structure
c      record / e_model6_real_s / E_model6_real
c      ALLOCATABLE :: E_model1_real (:)
c      ALLOCATE (E_model1_real (n_e_model1))
c
C
C      SUBROUTINE /Initset_maxwelldamp
C
C      Maxwell model の線形減衰
C
      subroutine Initset_maxwelldamp(Element,Newmark_P,
*      E_model6_real,m_filter)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / newmark_s      / Newmark_P
      record / element_s6     / Element
      record / e_model6_real_s / E_model6_real
      dt=Newmark_P.dt
      b = 2.*Element.C0 + Element.AK*dt
      E_model6_real.alph_0 = Element.C0*Element.AK*dt/b      ! 0
      E_model6_real.gumma_0 = (2.*Element.C0-Element.AK*dt)/b ! 0
      E_model6_real.alph_d0 = 0.0
      b = 2.*Element.C1 + Element.AK*dt
      E_model6_real.alph_1 = Element.C1*Element.AK*dt/b      ! 1
      E_model6_real.gumma_1 = (2.*Element.C1-Element.AK*dt)/b ! 1
      E_model6_real.alph_d1 = Element.AK*dt/b                 ! '1
      b = 2.*Element.C2 + Element.AK*dt
      E_model6_real.alph_2 = Element.C2*Element.AK*dt/b      ! 2
      E_model6_real.gumma_2 = (2.*Element.C2-Element.AK*dt)/b ! 2
      E_model6_real.alph_d2 = 0.0
      E_model6_real.f00 = Element.F0*(1.-Element.C1/Element.C0) !f0
      if(Element.Time_lag.ne.0.) then                          ! 2
        E_model6_real.max_istat_lag=Element.Time_lag/dt
        if(E_model6_real.max_istat_lag.eq.0) then              ! 3
          E_model6_real.cx_lag=0.
        else                                                    ! 4
          E_model6_real.cx_lag =
*      (Element.C0-Element.C2)/E_model6_real.max_istat_lag
        endif
      else                                                      ! 5
        E_model6_real.max_istat_lag=0
        E_model6_real.cx_lag=0.
      endif
      E_model6_real.jact = Element.damp_type +0.5              ! 6

```

```

E_model6_real.istat = 0
E_model6_real.gumma = E_model6_real.gumma_0
E_model6_real.alph = E_model6_real.alph_0
E_model6_real.f01 = 0.
E_model6_real.f0 = 0.
E_model6_real.alfd = 0.
E_model6_real.cx = Element.C0
write(76,'(a/a)') ' Maxwell member ', ! 7
* ' c alph gumma alph_d f0 f00'
write(76,'(a,10f10.3)') ' c0',Element.C0,E_model6_real.alph_0,
* E_model6_real.gumma_0,E_model6_real.alph_d0
write(76,'(a,10f10.3)') ' c1',Element.C1,E_model6_real.alph_1,
* E_model6_real.gumma_1,E_model6_real.alph_d1,Element.F0,
* E_model6_real.f00
write(76,'(a,10f10.3)') ' c2',Element.C2,E_model6_real.alph_2,
* E_model6_real.gumma_2,E_model6_real.alph_d2
write(76,'(a,i4,10e12.4)') ' lag',E_model6_real.max_istat_lag,
* Element.Time_lag,E_model6_real.cx_lag,dt
E_model6_real.fn=0. ! 8
E_model6_real.u2=0. ! ばねの変位
E_model6_real.u1=0. ! ダンパーの変位
E_model6_real.u1d=0. ! ダンパーの速度
E_model6_real.uud=0. ! フィルター通過後のダッシュポット速度
E_model6_real.uudx=0.
if(m_filter .eq.1 .and. Element.Filter .ne. 0. ) then ! 9
    icon=1
    call IIRDC(ICON,uudd,E_model6_real.uud,E_model6_real.WORK(1))
endif
E_model6_real.uudd =0.
return
end

```

1. Maxwell モデルで必要となる多くの係数を計算する。計算式横のコメント及び構造体のコメントと計算式(5.32)を比較して確かめられたい。
2. セミアクティブ型を使用する場合、ダンパーのオリフィスを開放することで、減衰定数が大きく変えることができるが、ユーザーの設定で、その時間に少しの遅れを生じさせるかどうかチェックする。もし遅れを生じさせる場合は、その増分ステップ数を構造体成分 E_model6_real.max_istat_lag にセットする。生じさせない場合は、その構造体にゼロをセットする(5)。さらに、遅れを生じさせる場合で、しかも、計算した増分ステップ数がゼロとなる場合は、 t 間の減衰定数の変化量をゼロとし(3)、ゼロでない場合は、その値を計算し、構造体 E_model6_real.cx_lag にセットする(4)。
6. さらに、各種の係数、パラメータの初期設定を行う。

7. Maxwell モデルの係数やパラメータなど、設定した値をワークファイルに出力する。
8. ワーク用のデータ領域をゼロクリアする。
9. モデルがセミアクティブ型で、しかもフィルターを使用する場合、フィルターに関する初期設定をサブルーチン IIRDC()を用いて行う。

5.11.3 線形剛性と線形減衰

このモデルの線形剛性行列はゼロ行列である。部材モデルの線形剛性を他の部材モデルと同様に配列にセットする必要があるため、次に示す線形の剛性行列を作成するサブルーチンの中で、このゼロ行列を作成するサブルーチンがコールされる。

```

C
C      SUBROUTINE /Cal_stiff_linear
C
C      線形剛性行列の計算(ok)
C
      subroutine Cal_stiff_linear()
      do i=1,n_member
      iet = Member(i).element_type
      goto(11,12,13,14,15,16,17,18,19,20),iet
      .
      .
16 continue
C
C      Model_No.6 3次元制震 Maxwell モデル
      call Cal_link_maxwelldamp(ak_linear(1,1,i))
      goto 100
      .
      .
      enddo

```

線形の剛性行列は、サブルーチン Cal_link_maxwelldamp()で行われ、その内容を以下に示す。プログラムは非常に単純であり、線形減衰行列である av をゼロクリアして戻す。

```

C
C      SUBROUTINE /Cal_link_maxwelldamp
C
C      Maxwell model の線形剛性
C
      subroutine Cal_link_maxwelldamp(Av)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      dimension av(12,12)

```

```

do i=1,12
do j=1,12
av(j,i)=0.0
enddo
enddo
return
end

```

次に、このモデルの部材型の線形減衰行列は、以下のように線形の減衰行列を作成するサブルーチンの中でコールされる。部材モデルで減衰を有している型は、現在の SPACE では、この Maxwell 型のみである。したがって、このサブルーチンでは、この Maxwell 型の部材モデルにのみサブルーチンコールが記述されている。

```

C
C      SUBROUTINE /Cal_damp_linear
C
C      線形減衰行列の計算(ok)
C
      subroutine Cal_damp_linear()
      do i=1,n_member
      iet = Member(i).element_type
C
C                                     部材減衰を持っているか
      if( Element(ie).nm_damp .ne. 0) then
      goto(11,12,13,14,15,16,17,18,19,20),iet
      .
      .
C
C                                     Model_No.6 3次元制震 Maxwell モデル
      ienn=Member(i).nm_damp
      ii=Element(ie).nm_type      ! (maxwell モデルでは、 1 : x方向 2 : y方向 3 : z方向)
      call Cal_lin_maxwelldamp(E_model6_real(ien),
*                               ac_linear(1,1,ienn),ii)
      goto 100
      .
      .
100 continue
      endif
      end do
      return
      end

```

上記の線形の減衰行列を計算するサブルーチンを以下に示す。

```

C
C      SUBROUTINE /Cal_lin_maxwelldamp
C
C      Maxwell model の線形減衰
C
      subroutine Cal_lin_maxwelldamp(E_model6_real,Av,ii)
      implicit real*8(A-H,O-Z)
      include "submain.h"

```



```

record / e_model6_real_s / E_model6_real
dimension av(12,12)
if(ii.le.0.and.ii.gt.3) return ! 1
do i=1,12 ! 2
do j=1,12
av(j,i)=0.0
enddo
enddo
av(ii,ii)= E_model6_real.alph_0 ! 3
av(ii,ii+6)= -E_model6_real.alph_0
av(ii+6,ii)= -E_model6_real.alph_0
av(ii+6,ii+6)= E_model6_real.alph_0
return
end

```

1. 制震ダンパーの配置方向を示す変数 ii が1から3以外はデータの設定エラーであり、このサブルーチンから戻る。
2. 減衰行列をゼロクリアする。
3. 構造体 E_model6_real.alph_0 より減衰定数 α_0 を取得し、制震ダンパーの配置方向を示す変数にしたがってこの値を減衰行列にセットする。

5.11.4 非線形減衰

Maxwell モデルの部材型非線形減衰行列を作成するサブルーチンは Cal_nonlin_maxwelldamp() であり、この非線形減衰行列は陰解法で使用する。また、このサブルーチンをコールする2つのサブルーチンは、Add_damp3_ld_ex() と Build_sky_c_ex() である。陰解法の部材型減衰行列をスカイライン行列に組み込むサブルーチン Build_sky_c_ex() について以下に示す。

```

C
C      SUBROUTINE /Build_sky_c_ex
C
C      部材非線形減衰行列のスカイライン行列への組み込み(ok)
C
      subroutine Build_sky_c_ex(gskym,Member,Element,n_member,rot_memb,
*                               ac_member ,Newmark_P, max_h_sky,E_model6_real)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      dimension ac_member(12,12,*),gskym(*)
      dimension max_h_sky(0:*),ac_nonlinear(12,12),bk(12,12)
      dimension rot_memb(3,3,2,*)
      record / newmark_s / Newmark_P
      record / member_s / Member

```

```

record / element_s / Element
record /E_model6_real_s / E_model6_real
dimension Member(*),Element(*)
dimension E_model6_real(*)

c
c  n_istep                :integer   第一ステップか第二ステップか
c  Member                 :structure
c  n_member               :integer   部材数
c  Ac_member(12,12,nc_member) :real*8 部材減衰行列 (釣合系)
c  Newmark_P              : structure
c  max_h_sky(n_unknown+1)  : integer   スカイライン行列の各列の高さ
c
  par=Newmark_P.ddt
  do i=1,n_member
    ij=Member(i).nm_damp
    if(ij.ne.0) then                                     ! 1
    if(Member(i).element_type.eq.6) then                 ! 2
c                                     Model_No.6 3次元制震 Maxwell モデル
      ien= Member(i).n_model_type
      ie = Member(i).nm_element
      ii=Element(ie).nm_type
      call Cal_nonlin_maxwelldamp(E_model6_real(ien),    ! 3
*                                     bk,ii)
      call Rotate_K(bk,rot_memb(1,1,1,i),                ! 4
*                                     rot_memb(1,1,2,i),ac_nonlinear)
      call Build_ak(gskym,                               ! 5
*       Member(i).irest(1), max_h_sky,
*       par,ac_nonlinear )
      else                                               ! 6
c                                     通常の線形減衰
      call Build_ak(gskym,
*       Member(i).irest(1), max_h_sky,
*       par,ac_member(1,1,ij) )
      endif
    end if
  end do
  return
end

```

1. この部材が減衰項を含むかどうかチェックし、減衰項を含む場合は以下の処理を行う。
2. 部材モデルが Maxwell 型である場合は以下の処理を行い、そうでない場合は、6. に進む。
3. 非線形減衰行列 bk を、サブルーチン `Cal_nonlin_maxwelldamp()` を用いて求める。
4. 求めた非線形減衰行列 bk を、部材座標系から釣合座標系に変換し、配列 `ac_nonlinear` にセットする。
5. 非線形減衰行列 `ac_nonlinear` を、サブルーチン `Build_ak()` を用いてスカイライン行列 `gskym` に組み込む。

6. 部材モデルが Maxwell 型でない場合は、線形の減衰行列 `ac_member` をサブルーチン `Build_ak()` を用いて、スカイライン行列 `gskym` に組み込む。

次に、サブルーチン `Add_damp3_ld_ex()` では、部材非線形減衰行列を求め、速度と掛け算して、その結果を右辺項に足しこむ。このサブルーチンを見てみよう。

```

C
C      SUBROUTINE /Add_damp3_ld_ex
C
C      部材減衰による減衰項のセット(ok)
C
      subroutine Add_damp3_ld_ex(nx,ld_point,Member,n_member,ac_member,
*      a_vector,Element,rot_memb,E_model6_real,Newmark_P,n_damp)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / newmark_s      /Newmark_P
      record / member_s       /Member
      record / element_s      / Element
      record /E_model6_real_s / E_model6_real
      dimension Member(*),Element(*)
      real*8 ld_point(*),ac_nonlinear(12,12),bk(12,12)
      dimension rot_memb(3,3,2,*),ac_member(12,12,*)
      dimension u(12)
      dimension a_vector(*)
      dimension E_model6_real(*)

C
C      nx                      :integer  第一ステップか第二ステップか
C      ld_point(*)             :real*8   右辺荷重項
C      Point                   :structure
C      n_point                 :integer  節点数
C      a_vector                 :real*8   a ベクトル
C      ac_point(2,n_point)     :real*8   部材減衰行列(釣合系)
C      Newmark_P               :structure
C      Parameter_C             :structure
C      n_damp                  :integer  部材減衰の部材がありか
C
      if(n_damp.eq.0) return
C
      do i=1,n_member
      ij = Member(i).nm_damp
      if(ij .ne. 0) then
      do j=1,12
      irest = Member(i).irest(j)
      if(irest.gt.0) then
      u(j)= a_vector(irest)
      else
      u(j)=0.0
      endif
      ! 1
      ! 2

```

```

        end do
c
        Model_No.6 3次元制震 Maxwell モデル
        if(Member(i).element_type.eq.6) then
            ien= Member(i).n_model_type
            ie = Member(i).nm_element
            ii=Element(ie).nm_type
            call Cal_nonlin_maxwelldamp(E_model6_real(ien),
*                                     bk,ii)
            call Rotate_K(bk,rot_memb(1,1,1,i),
*                       rot_memb(1,1,2,i),ac_nonlinear)
c
        else
            do j=1,12
            do k=1,12
            ac_nonlinear(j,k)=ac_member(j,k,ij)
            enddo
            enddo
            endif
c
        do j=1,12
            irest=Member(i).irest(j)
            if(irest.gt.0) then
                sum=0.
                do k=1,12
                    sum=sum+ac_nonlinear(j,k)*u(k)
                enddo
                Id_point(irest)=Id_point(irest) + sum
            endif
        end do
        endif
        enddo
c
        return
        end

```

1. この部材が減衰項を含むかどうかチェックし、減衰項を含む場合は以下の処理を行う。
2. 式(3.7)の $\{a\}$ からこの部材両端に関連するベクトル u を取り出す。

$$\{a\} = \{\dot{y}_n\} + \Delta t(1-\delta)\{\ddot{y}_n\}$$
3. 部材モデルが Maxwell 型である場合は以下の処理を行い、そうでない場合は、6に進む。
4. 非線形減衰行列 bk を、サブルーチン Cal_nonlin_maxwelldamp()を用いて求める。
5. 求めた非線形減衰行列 bk を、部材座標系から釣合座標系に変換し、配列 ac_nonlinear にセットする。
6. 部材モデルが Maxwell 型でない場合は、線形の減衰行列 ac_member を配列 ac_nonlinear にコピーする。

7. エレメントの自由度を取り出し、拘束でない場合は以下の処理を行う。
8. 釣合座標系の減衰行列と、ベクトル $\{a\}$ から部材両端の速度ベクトル u を取り出し、そのベクトルと掛け算を行う。
9. 掛け算の結果をベクトル `ld_point` に足しこむ。

次に、非線形減衰行列を計算するサブルーチンを以下に示す。

```

C
C      SUBROUTINE /Cal_nonlin_maxwelldamp
C
C      Maxwell model の非線形減衰
C
      subroutine Cal_nonlin_maxwelldamp(E_model6_real,Av,ii)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / e_model6_real_s / E_model6_real
      dimension av(12,12)
      if(ii.le.0.and.ii.gt.3) return                ! 1
      do i=1,12                                     ! 2
      do j=1,12
      av(j,i)=0.0
      enddo
      enddo
      av(ii,ii)=   E_model6_real.alph                ! 3
      av(ii,ii+6)= -E_model6_real.alph
      av(ii+6,ii)= -E_model6_real.alph
      av(ii+6,ii+6)= E_model6_real.alph
      return
      end

```

1. 制震ダンパーの配置方向を示す変数 `ii` が1から3以外はデータの設定エラーであり、このサブルーチンから戻る。
2. 減衰行列をゼロクリアする。
3. 構造体 `E_model6_real.alph` より非線形減衰定数 `alph` を取得し、制震ダンパーの配置方向を示す変数にしたがって、この値を減衰行列にセットする。この非線形減衰定数 `alph` は、Maxwell モデルの弾塑性チェックでセットされる。

5.11.5 右辺項(fd) の計算

本節では、振動方程式の右辺項で、Maxwell モデルが関連する部材減衰 $\{\bar{f}_d\}$ について説明する。この $\{\bar{f}_d\}$ を計算するサブルーチンは、線形解析、反復解法と陰解法では異なり、両者について述べる。前者2つと後

者の解析で使用する2つのサブルーチンでは、submain_dynamic_a()でコールされる。

```

c                                     Maxwell 型モデルの計算(ok)
      call Add_fdd_ld(ld_point,E_model6_real,Element,
*         Member,n_member,est_vel_point,rot_memb)
c                                     Maxwell 型モデルの右辺項 fd の計算(ok)
      call Add_fdd_ld_ex(ld_point,E_model6_real,Element,
*         Member,n_member,est_vel_point,rot_memb)

```

まず、線形解析と反復解法で使用するサブルーチン Add_fdd_ld()を示す。

```

C
C      SUBROUTINE /Add_fdd_ld
C
C      Maxwell 減衰項に関するベクトルを加える。(ok)
C
      subroutine Add_fdd_ld()
      do i=1,n_member
      iet = Member(i).element_type
      ie = Member(i).nm_element
      if( Element(ie).nm_damp .ne. 0) then
      do j=1,12
      irest = Member(i).irest(j)
      if(irest.ne.0) then
      ud(j)=est_vel_point(irest)
      else
      ud(j)=0.
      endif
      enddo
      call RotateL_v(1,ud,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vpp)
      goto(11,12,13,14,15,16,17,18,19,20),iet
      .
      .
      16 continue
c                                     Model_No.6 3次元制震 Maxwell モデル
      ii=Element(ie).nm_type
      call Cal_force_maxwell_damp(vpp,E_model6_real(ien),av,ii,i)
      goto 100
      .
      .
      100 continue
c                                     右変更への追加
      call RotateL_v(2,av,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vpp)
      do j=1,12
      irest = Member(i).irest(j)
      if(irest.ne.0) then
      ld_point_repeat(irest)=ld_point_repeat(irest) - vpp(j)

```

```

end if
end do
endif
end do
return
end

```

1. この部材が減衰項を含むかどうかチェックし、減衰項を含む場合は以下の処理を行う。
2. 全体速度ベクトルから、部材両端の節点速度ベクトルを取り出す。
3. 部材両端の節点速度ベクトルを釣合座標系から部材座標系に変換する。
4. 部材モデル番号にしたがって処理を分類する。
5. Maxwell モデルに関連する $\{\bar{f}_d\}$ を、Cal_force_maxwelldamp() を用いて求める。
6. 求めた Maxwell モデルに関連する $\{\bar{f}_d\}$ を、部材座標系から釣合座標系に変換する。
7. 釣合式の右辺項 Id_point_repeat に足しこむ。

上記のサブルーチン Cal_force_maxwelldamp() を以下に示す。

```

C
C      SUBROUTINE /Cal_force_maxwelldamp
C
C      Maxwell model の反復解法用節点力
C
      subroutine Cal_force_maxwelldamp(ud,E_model6_real,Av,ii,im)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / e_model6_real_s / E_model6_real
      dimension av(12),ud(12)
      if(ii.le.0.and.ii.gt.3) return ! 1
      do j=1,12 ! 2
      av(j)=0.0
      enddo
      udn1=ud(ii+6)-ud(ii) !増分後の速度 ! 3
      udn =E_model6_real.uudd !増分前の速度 ! 4
      fd=(E_model6_real.alph - E_model6_real.alph_0)*udn1 + ! 5
      * E_model6_real.gumma*E_model6_real.fn +
      * E_model6_real.alph*udn +
      * 2.*E_model6_real.alfd*E_model6_real.f0
      av(ii)= -fd ! 6
      av(ii+6)= fd
      return
      end

```

1. 制震ダンパーの配置方向を示す変数 ii が1から3以外はデータの設定エラーであり、このサブルーチンから戻る。
2. 減衰に関連する応力ベクトル $\{\bar{f}_d\}$ をゼロクリアする。
3. 制震ダンパーの配置方向にしたがって、増分後の速度を計算する。
4. 増分前の速度を構造体より取得する。この値は弾塑性チェックを行うサブルーチンで設定する。
5. 応力を計算する。ここでは線形解析時と反復解法時の応力は、理論式では異なるが、線形解析における初期設定で $E_model6_real.alph$ に $E_model6_real.alph_0$ をセットしているため、線形解析においても正確に対応する。
6. 減衰に関連する応力ベクトル $\{\bar{f}_d\}$ に求めた応力をセットする。

次に、陰解法で使用されるサブルーチン `Add_fdd_ld_ex()` を示す。

```

C
C      SUBROUTINE /Add_fdd_ld_ex
C
C      Maxwell 減衰項に関するベクトルを加える。(ok)
C
      subroutine Add_fdd_ld_ex(ld_point_repeat,E_model6_real,Element,
*      Member,n_member,est_vel_point,rot_memb)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s / Member
      record / e_model6_real_s / E_model6_real
      record /element_s / Element
      dimension Member(*),E_model6_real(*),Element(*)
      real*8 ld_point_repeat(*),est_vel_point(*)
      dimension rot_memb(3,3,2,*)
      dimension av(12),ud(12),vpp(12)

C
C      ld_point_repeat(*) : real*8   線形右辺項ベクトル
C      Member              :structure
C      n_member            :integer   部材数
C      est_vel_point       :real*8    予測節点速度
C
      do i=1,n_member
      mem = i
      iet = Member(i).element_type
      ie = Member(i).nm_element

C
C      部材減衰を持っているか
      if( Element(ie).nm_damp .ne. 0) then
      ien= Member(i).n_model_type
      do j=1,12
      irest = Member(i).irest(j)
      if(irest.ne.0) then
      ud(j)=est_vel_point(irest)

```



```

        else
        ud(j)=0.
        endif
        enddo
        call RotateL_v(1,ud,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vpp)
        goto(11,12,13,14,15,16,17,18,19,20),iet
        .
        .
16 continue
c
Model_No.6 3次元制震 Maxwell モデル
        ii=Element(ie).nm_type
        call Cal_forcef_maxwelldamp(E_model6_real(ien),av,ii)
        goto 100
        .
        .
100 continue
c
右変更への追加
        call RotateL_v(2,av,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vpp)
        do j=1,12
        irest = Member(i).irest(j)
        if(irest.ne.0) then
        ld_point_repeat(irest)=ld_point_repeat(irest) - vpp(j)
        end if
        end do
        endif
        end do
        return
        end

```

上記サブルーチンは、Add_fdd_ld()と内容はほとんど同じであり、異なるのは応力を求めるサブルーチン Cal_forcef_maxwelldamp()のみである。次に、このサブルーチンを示す。

```

C
C      SUBROUTINE /Cal_forcef_maxwelldamp
C
C      Maxwell model の陰解法用節点力
C
      subroutine Cal_forcef_maxwelldamp(E_model6_real,Av,ii)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / e_model6_real_s / E_model6_real
      dimension av(12)
      if(ii.le.0.and.ii.gt.3) return ! 1
      do j=1,12 ! 2
      av(j)=0.0
      enddo
      udn =E_model6_real.uudd !増分前の速度 ! 3
      fd= ! 4
      * E_model6_real.gumma*E_model6_real.fn +
      * E_model6_real.alph*udn +
      * 2.*E_model6_real.alfd*E_model6_real.f0

```

```

av(ii)=      -fd
av(ii+6)=     fd
return
end

```

1. 制震ダンパーの配置方向を示す変数 ii が1から3以外はデータの設定エラーであり、このサブルーチンから戻る。
2. 速度ベクトル av をゼロクリアする。
3. 増分前の速度を構造体成分 E_model6_real.uudd より取得する。この値は弾塑性チェックを行うサブルーチンで設定される。
4. 陰解法の応力 \bar{f}_d を計算する。
5. 減衰に関連する応力ベクトル $\{\bar{f}_d\}$ に求めた応力 \bar{f}_d をセットする。

5.11.6 Maxwell モデルの応力計算

Maxwell モデルのばねに生じる応力は、submain_dynamic_a()においてコールされるサブルーチン Cal_stress()によって計算される。その部分のコードを以下に示す。

```

C
C      SUBROUTINE /Cal_stress
C
C      部材内応力の計算(ok)
C
      subroutine Cal_stress()
C
      do i=1,n_member
C
C                                     部材両端の変位取得
        do j=1,12
          ires=Member(i).irest(j)
          if(ires.gt.0) then
            vp(j)=past_disp_point(ires)
            v(j)=est_ddisp_point(ires)
          else
            v(j)=0.
            vp(j)=0.
          endif
        enddo
C
C                                     変位を釣合系から部材座標系に変換
        call RotateL_v(1,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vv)
        call RotateL_v(1,vp,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vpp)
C
C                                     要素及びモデルのセット
        mem = i
        iet = Member(i).element_type
        iett=(iet-1)/10
        goto(11,12,13,14,15,16,17,18,19,20),iet
11 continue

```

```

      .
      .
16 continue
c
Model_No.6 3次元制震 Maxwell モデル(ok)
      do j=1,12
      ires=Member(i).irest(j)
      if(ires.gt.0) then
      v(j)=past_vel_point(ires)
      else
      v(j)=0.
      endif
      enddo
      call RotateL_v(1,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vv)
      ii=Element(ie).nm_type
      call Stress_maxwelldamp(vpp,vv,E_model6_real(ien),
*           Element(ie),ii,Member(i),Model_type.n_m_filter)
      goto 100
      .
      .
100 continue
      end do
      return
      end

```

上記の Maxwell モデルの応力計算を行うサブルーチンを具体的に示す。

```

C
C      SUBROUTINE /Stress_maxwelldamp
C
C      Maxwell model の応力計算
C
      subroutine Stress_maxwelldamp(u,ud,E_model6_real,
*           Element,ii,Member,m_filter)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / e_model6_real_s / E_model6_real
      record / element_s6      / Element
      record / Member_s        / Member
      dimension u(12),ud(12)
      if(ii.le.0.and.ii.gt.3) return ! 1
      udn1=ud(ii+6)-ud(ii) ! 速度 ! 2
      un =u(ii+6) - u(ii) ! 変位 ! 3
      fn = E_model6_real.fn ! 前回の応力 ! 4
      fn1 = E_model6_real.gumma*fn + E_model6_real.alph*udn1 + ! 5
*      E_model6_real.alph*E_model6_real.uudd +
*      2.*E_model6_real.alfd*E_model6_real.f0
      E_model6_real.fn=fn1 ! 今回の応力のセット ! 6
      E_model6_real.u2=fn1/Element.AK ! ばねの変位
      E_model6_real.u1=un - E_model6_real.u2 ! ダンパーの変位
      E_model6_real.u1d=(fn1-E_model6_real.f0)/E_model6_real.cx ! ダンパーの速度
C
c      非線形から線形へ戻るときのダンパー速度の飛び越し処理

```

```

C
  if(E_model6_real.istat.eq.1.or.E_model6_real.istat.eq.2) then          ! 7
    if(abs(fn1).lt.Element.F0)then
      E_model6_real.u1d=(fn1)/Element.C0      ! ダンパーの速度
    endif
  endif

C
  c          データを構造体にセット
C
  E_model6_real.uudd= udn1          ! 8
  E_model6_real.f01=E_model6_real.f0
  Member.stress(ii)=fn1            ! 応力のセット
  E_model6_real.uudx=E_model6_real.uud

C
  c          速度にフィルターをかける
C
  if(m_filter .eq.1 .and. Element.Filter .ne. 0. ) then          ! 9
    uudx=ud(ii+6)-ud(ii)
    icon=2
    call IIRDC(ICON,uudx,E_model6_real.uud,E_model6_real.WORK(1))
  else
    E_model6_real.uud=(ud(ii+6)-ud(ii) )      ! フィルター用速度
  endif
  return
end

```

1. 制震ダンパーの配置方向を示す変数 ii が1から3以外はデータの設定エラーであり、このサブルーチンから戻る。
2. Maxwell 部材内に生じる速度を計算する。
3. 部材両端に生じる相対変位を求める。
4. 前ステップの応力 fn を構造体より取得する。
5. 式(5.42)より今回の応力 fn1 を計算する。
6. 応力 fn1 を構造体にセットする。また、ばねの変位、ダンパーの変位、ダンパーの速度を計算し、該当する構造体にセットする。
7. ダンパーの状態を表す E_model6_real.istat が1か2で、非線形状態であるとき、しかも、今回の応力値が限界応力より小さくなったとき、線形状態に戻ったとしてダンパーの速度を再計算する。
8. 今回計算した速度などを構造体にセットする。
9. セミアクティブダンパー用で、速度にフィルターを掛ける場合は、サブルーチン IIRDC()をコールする。

5.11.7 Maxwell モデルの非線形チェック

SPACE では、Maxwell モデルの応力と速度の関係がバイリニア型とするパッシブ型の制震装置を組み込んでいる。ここでは、この非線形性をチェックするサブルーチン `Check_Maxwell_stress()` について説明する。このサブルーチンは、`submain_dynamic_a()` で以下のように呼ばれる。

```
c
                                Maxwell 型モデルの非線形性チェック(ok)
      call Check_Maxwell_stress(Member,n_member,
*      Element,E_model6_real,Newmark_P)
```

さらに、このサブルーチンは、現在は、次のように Maxwell モデルの非線形性をチェックするサブルーチンのみコールする。

```
C
C      SUBROUTINE /Check_Maxwell_stress
C
C      Maxwell model の応力計算
C
      subroutine Check_Maxwell_stress(Member,n_member,
*      Element,E_model6_real,Newmark_P)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s      / Member
      record / element_s     / Element
      record / E_model6_real_s / E_model6_real
      record / newmark_s      / Newmark_P
      dimension Member(*),Element(*),E_model6_real(*)

c
      do i=1,n_member
c
c      要素及びモデルのセット
          iet = Member(i).element_type
          ie  = Member(i).nm_element
          ien = Member(i).n_model_type
          if(iet.eq.6)then
c
c      Model_No.6 3次元制震 Maxwell モデル(ok)
              call Check_maxwelldamp(E_model6_real(ien),Element(ie),Newmark_P)
          endif
      end do
      return
      end
```

サブルーチン `Check_maxwelldamp()` の内容を具体的に示す。

```
C
C      SUBROUTINE /Check_maxwelldamp
C
C      Maxwell model の応力計算
C
      subroutine Check_maxwelldamp(E_model6_real,Element,Newmark_P)
      implicit real*8(A-H,O-Z)
      include "submain.h"
```

```

      record / e_model6_real_s / E_model6_real
      record / element_s6      / Element
      record / newmark_s      / Newmark_P
      udmx = Element.udmx
      istat = E_model6_real.istat          ! 1
      ud1= E_model6_real.ud1
      ichange =0                          ! 2
      if(E_model6_real.jact .eq. 1) goto 9900 ! セミアクティブ型へ          ! 3
C
c      パッシブ型応力チェック
C
      goto (101,102,103),istat+1          ! 4
101 continue
      if(ud1 .ge. udmx ) then              ! 5
      ichange = 1
      istat = 1
      elseif(ud1 .le. -udmx) then          ! 6
      ichange = 1
      istat = 2
      endif
      goto 9800
102 continue
      if(ud1 .ge. udmx ) goto 9800          ! 7
      ichange = 1
      istat = 0
      goto 9800
103 continue
      if(ud1 .le. -udmx) goto 9800          ! 8
      ichange = 1
      istat = 0
      goto 9800
C
c      セミアクティブ型応力チェック
C
9900 continue                             ! 9
      uud=E_model6_real.uud*E_model6_real.uudx
      f0x=Element.f0x
      fn1=E_model6_real.fn
      goto(201,202,203,204,205,206,207,208,209),istat+1
201 continue
      .
      .
C
c      データの設定
C
9800 continue
      if(ichange .eq. 0) return              ! 10
      E_model6_real.istat= istat
      goto (10,11,12,13,14,15,16,17,18),istat+1
10 continue
      E_model6_real.alph=E_model6_real.alph_0          ! 11
      E_model6_real.gumma=E_model6_real.gumma_0
      E_model6_real.f01=E_model6_real.f0
      E_model6_real.f0=0.

```

```

      E_model6_real.alfd=0.
      E_model6_real.cx=Element.C0
      return
11  continue
      E_model6_real.alph=E_model6_real.alph_1
      E_model6_real.gumma=E_model6_real.gumma_1
      E_model6_real.f01=0.
      E_model6_real.f0=E_model6_real.f00
      E_model6_real.alfd=E_model6_real.alph_d1
      E_model6_real.cx=Element.C1
      return
12  continue
      E_model6_real.alph=E_model6_real.alph_1
      E_model6_real.gumma=E_model6_real.gumma_1
      E_model6_real.f01=0.
      E_model6_real.f0= -E_model6_real.f00
      E_model6_real.alfd=E_model6_real.alph_d1
      E_model6_real.cx=Element.C1
      return
13  continue
      .
      .
      .
      return
end

```

プログラムの内容をコードに従って説明する。このサブルーチンはパッシブ型とセミアクティブ型の応力チェックとデータの設定の2つに分かれている。ただし、ここではセミアクティブ型の応力チェックに関するコードは複雑であるため省略した。

1. モデルの状態を構造体 E_model6_real.istat より取得する。また、ダンパーの速度を取得する。
2. モデルの状態変化を知るパラメータ ichange をゼロクリアする。
3. 構造体 E_model6_real.jact が 1 で、セミアクティブ型である場合、処理を文番号 9900 に制御を移す。構造体 E_model6_real.jact が 2 でパッシブ型である場合、次の処理を行う。
4. モデルの状態 istat を 3 つに分け、その 3 区間で状態の変化を調査する。状態 istat が 0 の場合は線形状態であり、1 の場合は第 2 勾配区分状態であり、ダンパー速度が uamax より大きい場合に対応する。また、2 の場合も第 2 勾配区分状態であり、ダンパー速度が逆方向で uamax より大きい場合に対応する。
5. ここは線形状態であり、速度のチェックを行う。速度が uamax より大きい場合は、2 つのパラメータ ichange、istat を共に 1 にし、状

態変化があったことを示す。

6. 速度が逆方向に u_{\max} より大きい場合は、2つのパラメータを $i_{\text{change}}=1$ 、 $i_{\text{stat}}=2$ にセットし、状態変化があったことを示す。
7. ここは状態が1で、速度が u_{\max} より大きい場合は、状態変化なし。小さい場合は、 $i_{\text{change}}=1$ 、 $i_{\text{stat}}=0$ に設定する。
8. ここは状態が2で、速度が逆方向に u_{\max} より大きい場合は、状態変化なし。小さい場合は、 $i_{\text{change}}=1$ 、 $i_{\text{stat}}=0$ に設定する。
9. ここ以降は、セミアクティブ型の制震装置の状態チェックであり、ここでは省略する。
10. 状態変化がない場合は、このサブルーチンから戻る。
11. 線形状態のパラメータを設定する。
12. 状態1のパラメータを設定する。
13. 状態2のパラメータを設定する。
14. これ以降は、セミアクティブ型のパラメータを設定する。以降の説明は省略する。

これで、部材モデルの説明は終了するが、例として用いた両端ファイバーモデル以外に多数の静的縮合部材モデルが、SPACE には組み込まれている。それらについては付録のプログラムコードを参照されたい。また、新たに部材モデルをこの SPACE に組み込みたい場合は、第9章を参考にされると良い。多くの情報が得られるはずである。