第５章　部材モデル

5.1 はじめに

前章では、動的解析の数値計算フローについて説明した。どのように振動方程式を解いているかについて、おおよそ理解できたであろう。本章では、さらに詳しく計算手法について説明する。特に、部材モデルを中心に解説する。SPACE では、部材モデルを多数用意しており、これらを混乱なくコーディングするためには、部材モデルを上手に設計しなければならない。SPACE では部材モデルに階層構造を設定し、バグが発生しないように、また容易にメンテナンスが可能となるように備えている。次節では、この階層構造について説明し、その後は各モデルについて解説する。

5.2 各モデルの階層構造

SPACE システムの中には、多くの部材モデルが用意されており、しかも今後、より複雑なモデルが組み込まれることになる。これらを混乱なく、またバグを発生させないために、SPACE では部材モデルに階層構造を設定し、より安定して使用できるようにする。

図 5-1 には、部材の階層構造を表している。最上位層には、部材モデルがあり、その部材モデルには管理番号が付されている。１番から 10 番までが、図の左側にある幾何学的非線形性を考慮した弾性モデルやせん断型モデルなどで、部材ひとつで構成されるモデルである。また 11 番以上は図の右側に並んでいる静的縮合モデルに割り当てられており、現在は、図のようなエレメントが組み込まれている。静的縮合モデルには、表 5.1 に示されている両端ファイバーモデル以降で示されており、

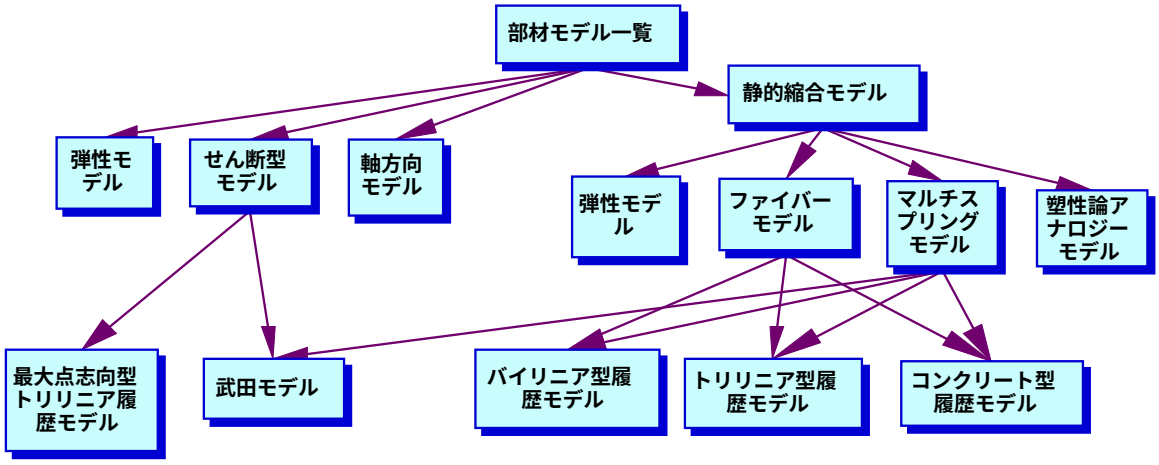

図 5-1 部材階層モデル

このモデルでは第２層として弾性モデル、ファイバーモデル、塑性論アナロジーモデルなどがある。例えば、管理番号 11 の両端ファイバーモデルは、図 5-2 に示すように剛域と両端のファイバー要素、中央部分弾性梁要素である。このように静的縮合モデルは下位レベルの部材モデル（エレメント）を組み込むことができる。上記したように、現在、SPACE では、表 5-1 の管理番号 11 以降が組み込まれている。なお、表中の部材管理番号と内部管理番号の違いは、部材管理番号はユーザーが使用する番号であり、内部管理番号はプログラム内で使用する番号である。

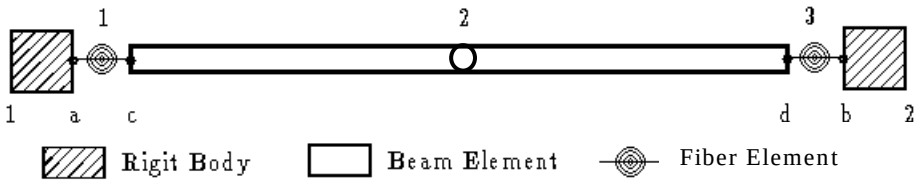

図 5-2 両端ファイバーモデル

表 5-1 部材管理番号（内部管理番号）　部材モデル

1	(1)	幾何学非線形型有限要素弾性モデル
2	(2)	3 次元せん断弾塑性モデル
3	(3)	3 次元軸力弾塑性モデル
4	(4)	3 次元ケーブル弾塑性モデル
5	(5)	3 次元免振モデル（MSS モデル）
6	(6)	3 次元制震 Maxwell モデル
7	(7)	3 次元弾塑性バネモデル
1 1	(11)	両端ファイバーモデル
1 2	(12)	両端、中央ファイバーモデル
1 3	(18)	両端ピンで中央ファイバーモデル
2 1	(13)	両端 MS モデル
2 2	(14)	両端、中央 MS モデル
3 1	(16)	両端アナロジーモデル
3 2	(17)	両端、中央アナロジーモデル
3 3	(19)	両端ピン、中央アナロジーモデル

次に、図 5-1 の部材モデルにおける階層構造では、様々なデータ領域やワーク領域が構造体として定義されている。多数のモデルが混乱することなく使用でき、それに対応して設計された構造体を効率良く、しか

もすばやくアクセスすることができなければならない。

ここでは、部材モデルの階層構造に合せて、その構造体と情報の流れ、また、他の構造体へのリンク方法を考えてみよう。例として、両端ファイバーモデルとして設計された部材モデル 11 について、構造体の階層構造並びにそれらのリンク方法を以下の表 5-2 にまとめた。構造体は、一般に対になっており、それらは解析中に構造体成分の値が変化しない構造体と、値が変化する構造体である。部材に割り付けられた標準の構造体として、前者が要素構造体の Element_s であり、後者は部材に関連する構造体 Member_s である。この２つの構造体は、全要素と全部材に各一つずつ存在することになる。表中の構造体は階層構造となっており、各部材モデルに対応する。したがって、表の下側の構造体が階層構造の下位層となっている。この下位層の構造体は効率よく記憶領域を確保するために、実際の解析モデルに合せて必要数分詰めて領域を確保している。したがって、下位の構造体から情報を取り出すためには、上位の構造体からのリンク情報が必要となる。この表ではリンク情報を構造体から抜き出して表示しており、また、その使用法の例を載せている。

表 5-2 部材モデル：両端ファイバーモデル

解析時に値が変化しない構造体　　解析時に値が変化する構造体

標準部材構造体

```
部材              Element_s                    Member_s
リンク情報     Element_s 構造体
      integer  n_element          !  非線形要素番号
リンク情報     Member_s 構造体
      integer  n_model            !  モデルの入れ物番号
      integer  n_model_type       !  モデル別の通し番号
例
        imm= Element.n_element      →    E_model11(imm)
     ie    = Member.nm_element     →    Element(ie)
     immm= Member.n_model_type     →    M_model11(immm).d_ra_1
```

静的縮合型部材モデル１１

```
                E_model11_s                  M_model11_s
リンク情報     E_model11_s 構造体
      integer  nm_section_1       !  i 端断面のファイバー開始番号
      integer  n_section_1        !  i 端断面のファイバー数
      integer  nm_section_2       !  j 端断面のファイバー開始番号
      integer  n_section_2        !  j 端断面のファイバー数
リンク情報     M_model11_s 構造体
```

```
      integer  nm_section_1        !  i 端断面のファイバー開始番号
      integer  n_section_1         !  i 端断面のファイバー数
      integer  nm_section_2        !  j 端断面のファイバー開始番号
      integer  n_section_2         !  j 端断面のファイバー数
例
    nm_div   = E_model.n_section_1  →  do i=1,nm_div
    nn       = E_model.nm_section_1 – 1 →  nm_type=E_model_fiber(nn).nm_type
    nnm      = M_model.nm_section_1 - 1  →  nmx=M_model_fiber(nnm).n_type
```

断面モデル

```
ファイバーデータ  E_model_fiber_s                M_model_fiber_s
リンク情報    E_model_fiber_s 構造体
      integer  nm_type              !  履歴モデルの番号
リンク情報    M_model_ fiber_s 構造体
      integer  n_type               !  履歴モデルの通し番号
例
       nm_type=E_model_fiber(nn).nm_type →  goto ( 10,20,30,40,50,60,70,80),nm_type
       nmx=M_model_fiber(nnm).n_type  →    Bilinear_work(nmx).i_stat
                                          Concrete_work(nmx).P1(2)
```

```
ファイバー要素のワーク領域
ワーク領域                                  Bilinear_work_s
                                            Trilinear_work_s
                                            Concrete_work_s
```

最初に部材モデルの管理番号について説明する。この部材モデルの管理番号は、サブルーチン Set_Model_type()で管理されている。このサブルーチンでは、管理番号と共にモデルに関する基礎的な情報が構造体の実態である Model_type に設定される。このサブルーチンを以下に示す。

```
C     ――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Set_Model_type
C     ――――――――――――――――――――――――――――――――
C     ●   Parameter 構造体の値セット
C     ――――――――――――――――――――――――――――――――
      subroutine Set_Model_type(Model_type)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / n_model_s / Model_type
c――――――――――――――――――――――――――――――――――――――――
c  モデルパラメータ
c      structure  / n_model_s/
c      integer    n_e_models          ! 要素モデルの最大数
c      integer    no_e_model(100)     ! 要素モデルの番号
c      integer    n_div_model(100)    ! 要素モデルの分割数
c      integer    nm_div_model(100)   ! ファイバー要素の最大数
c      integer    nm_div_element(100) ! ファイバー要素のエレメント最大数
```

```
c      integer    n_e_model(100)       ! 要素モデルの数
c      integer    n_m_model(100)       ! 部材モデルの数
c      integer    n_damp(100)          ! 部材減衰
c      integer    n_m_damp             ! 全部材減衰数
c      integer    n_m_bilinear         ! ファイバー要素バイリニア用の最大要素数
c      integer    n_m_trilinear        ! ファイバー要素トリリニア用の最大要素数
c      end structure
c      record / n_model_s / Model_type
c
c-----------------------------------------------------------------
      Model_type.n_e_models =      100
       Model_type.n_m_damp =        0
       Model_type.n_m_bilinear =    0
       Model_type.n_m_trilinear =   0
       Model_type.nm_div_fmodel =   0
       Model_type.nm_div_felement = 0
       Model_type.nm_div_msmodel =  0
       Model_type.nm_div_mselement= 0
       Model_type.n_m_ro_model    = 0
       do i=1,Model_type.n_e_models
        Model_type.no_e_model(i) = 0
       end do
c-----------------------------------------------------------------
c    ★        モデル No.１の定義 幾何学非線形型有限要素弾性モデル
c-----------------------------------------------------------------
      number = 1
      Model_type.no_e_model(number)   = 1  !要素モデルの番号
      Model_type.n_div_model(number)  = 1  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c-----------------------------------------------------------------
c    ★        モデル No.２の定義 3 次元せん断弾塑性モデル
c-----------------------------------------------------------------
      number = 2
      Model_type.no_e_model(number)   = 2  !要素モデルの番号
      Model_type.n_div_model(number)  = 1  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c-----------------------------------------------------------------
c    ★        モデル No.３の定義 3 次元軸力弾塑性モデル
c-----------------------------------------------------------------
      number = 3
      Model_type.no_e_model(number)   = 3  !要素モデルの番号
      Model_type.n_div_model(number)  = 1  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c-----------------------------------------------------------------
c    ★        モデル No.４の定義 3 次元ケーブル弾塑性モデル
c-----------------------------------------------------------------
      number = 4
      Model_type.no_e_model(number)   = 4  !要素モデルの番号
```

```
      Model_type.n_div_model(number)  = 1  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.５の定義 3 次元免振モデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 5
      Model_type.no_e_model(number)   = 5  !要素モデルの番号
      Model_type.n_div_model(number)  = 1  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.６の定義 3 次元制震 Maxwell モデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 6
      Model_type.no_e_model(number)   = 6  !要素モデルの番号
      Model_type.n_div_model(number)  = 1  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 1  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.７の定義 3 次元弾塑性バネモデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 7
      Model_type.no_e_model(number)   = 7  !要素モデルの番号
      Model_type.n_div_model(number)  = 1  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.１１の定義 両端ファイバーモデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number =11
      Model_type.no_e_model(number)   = 11 !要素モデルの番号
      Model_type.n_div_model(number)  = 6  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.１２の定義 両端、中央ファイバーモデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 12
      Model_type.no_e_model(number)   = 12  !要素モデルの番号
      Model_type.n_div_model(number)  = 7  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
      number =18
      Model_type.no_e_model(number)   = 13 !要素モデルの番号
c――――――――――――――――――――――――――――――★弾性要素４モデル
      Model_type.n_div_model(number)  = 5  !要素モデルの分割数
```

```
      Model_type.n_div_model(number)  = 3  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.１３の定義 両端 MS モデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 13
      Model_type.no_e_model(number)   = 21  !要素モデルの番号
      Model_type.n_div_model(number)  = 6  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.１４の定義 両端、中央 MS モデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 14
      Model_type.no_e_model(number)   = 22  !要素モデルの番号
      Model_type.n_div_model(number)  = 7  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.１６の定義 両端アナロジーモデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 16
      Model_type.no_e_model(number)   = 31  !要素モデルの番号
      Model_type.n_div_model(number)  = 6  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.１７の定義 両端、中央アナロジーモデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 17
      Model_type.no_e_model(number)   = 32  !要素モデルの番号
      Model_type.n_div_model(number)  = 7  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
c―――――――――――――――――――――――――――――――――――――――――――――
c    ★        モデル No.１９の定義 両端ぴん、中央アナロジーモデル
c―――――――――――――――――――――――――――――――――――――――――――――
      number = 19
      Model_type.no_e_model(number)   = 33  !要素モデルの番号
      Model_type.n_div_model(number)  = 3  !要素モデルの分割数
      Model_type.n_e_model(number)    = 0  !要素モデルの数
      Model_type.n_m_model(number)    = 0  !部材モデルの数
      Model_type.n_damp(number)       = 0  !部材減衰ありか
       return
      end
```

部材モデルの最下層は、履歴モデルである。したがって、解析モデル

の中で、履歴モデルの番号を変更することで異なった履歴モデルを使用することができる。現在、この履歴モデルには以下のモデル番号が付けられている。ただし、＊印は現在使用できない。また、せん断型モデル、ファイバーモデル、アナロジーモデル以外の履歴モデルは、そのモデル固有の履歴モデルとなっており、その部材モデルに組み込まれている。

１）せん断型モデル
　　1：トリリニア
　　2：最大点指向型
　　3：武田モデル
　　4：バイリニア
２）各ファイバーの履歴タイプ
　　１：対称バイリニア
　　２：対称トリリニア
　　３：直線コンクリート型
　　４：曲線コンクリート型＊
　　５：対称バイリニア型(移動＋等方硬化用)＊
　　６：対称トリリニア型 (移動＋等方硬化用)＊
　　７：非対称バイリニア型
　　８：非対称トリリニア型＊
３）アナロジーモデル履歴タイプ
　　11：完全弾塑性型
　　12：等方硬化弾塑性型＊
　　13：移動硬化弾塑性型＊
　　14：等方硬化＋移動硬化弾塑性型＊
４）マルチスプリング履歴タイプ
　　21：武田モデル＊
　　22：等方硬化＋移動硬化トリリニア型＊

5.3 部材モデルの作成手法

部材モデルに設定した階層構造をどのようにプログラムコードとして実現するのか、実際のコードを用いて説明しよう。数値解析を実行する上で、次の 6 つのサブルーチンで最上位の階層構造を使用している。この部材モデルの階層構造については、前章で説明した。ここでは、この階層構造を復習して、確実に理解しよう。

```
c――――――――――――――――――――――――――――――――★部材の線形剛性計算(ok)
      call Cal_stiff_linear( )
c――――――――――――――――――――――――――――――――★モデルの初期設定
      call Set_initial_data( )
c――――――――――――――――――――――――――――――――★部材の減衰行列計算(ok)
      call Cal_damp_linear( )
c――――――――――――――――――――――――――――――――★部材の整合質量行列計算(ok*)
      call Cal_mass_linear( )
c――――――――――――――――――――――――――――――――★ファイバー応力セット
      call Check_stress( ),
c――――――――――――――――――――――――――――――――★接線剛性の計算(ok)
      call Get_nonlinear_stiff( )
```

ここで示した６つのサブプログラムの構造はほとんど同じであり、代表として線形の剛性行列を求めるサブプログラム Cal_stiff_linear() の概要を以下に示す。このプログラムの全コードは、あるいは他の５つのサブプログラムは前章もしくは付録で示されている。

このプログラムの中で、コード右側にあるコメント番号１は、全部材について、ループして線形剛性を計算せよという命令である。同じく２は、部材の要素タイプをMember構造体の中から情報を取り出している。また３は、内部モデル管理番号によって飛び先を変える文であり、そのコードによって、モデルのひとつであるせん断型の弾塑性モデルの線形剛性行列を求めるサブルーチン(4)に飛ぶ。

```
C     ――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_stiff_linear
C       ――――――――――――――――――――――――――――――――
C     ●   線形剛性行列の計算(ok)
C     ――――――――――――――――――――――――――――――――
      subroutine Cal_stiff_linear(    )
C     ――――――――――――――――――――――――――――――――
c――――――――――――――――――――――――――――――――――――――――――――――
      n_member = Parameter_C.N_member
      do i=1,n_member                                              ! 1
      mem = i
      iet  = Member(i).element_type                                ! 2
      iett=(iet-1)/10
      if(iett.eq.0)then
      goto(11,12,13,14,15,16,17,18,19,20),iet                      ! 3
   11 continue
c――――――――――――――――――――――――――――――――★Model_No.1 通常の有限要素弾塑性モデル
      call Cal_lin_stiff_M1(  )
      goto 100
   12 continue
c――――――――――――――――――――――――――――――――★Model_No.2 3 次元せん断弾塑性モデル
      call Cal_lin_stiff_M2(  )                                    ! 4
      goto 100
   13 continue
```

```
c—————————————————————————————————★Model_No.3 3次元軸力弾塑性モデル
      call Cal_lin_stiff_M3(  )
      goto 100
   14 continue
c—————————————————————————————————★Model_No.4 3次元ケーブル弾塑性モデル
       call Cal_lin_stiff_M4(  )
      goto 100
   15 continue
c—————————————————————————————————★Model_No.5 3次元免振モデル
      call Cal_lin_stiff_M5(  )
      goto 100
   16 continue
c—————————————————————————————————★Model_No.6 3次元制震 Maxwell モデル
      call Cal_link_maxwelldamp(  )
      goto 100
   17 continue
c—————————————————————————————————★Model_No.7 3次元プレテンション動作モデル
c      call Cal_lin_stiff_M7(  )
      goto 100
   18 continue
c—————————————————————————————————★Model_No.8
      goto 100
   19 continue
c—————————————————————————————————★Model_No.9
      goto 100
   20 continue
c—————————————————————————————————★Model_No.10
      goto 100
      elseif(iett.eq.1)then
      goto(111,112,113,114,115,116,117,118,119,120),iet-10
  111 continue
c—————————————————————————————————★Model_No.11 両端ファイバーモデル
      call Cal_lin_stiff_M11(  )
      goto 100
  112 continue
c—————————————————————————————————★Model_No.12 両端、中央ファイバーモデル
      call Cal_lin_stiff_M12(  )
      goto 100
  113 continue
c—————————————————————————————————★Model_No.13 両端 MS モデル
      call Cal_lin_stiff_M21(  )
      goto 100
  114 continue
c—————————————————————————————————★Model_No.14 両端、中央 MS モデル
      call Cal_lin_stiff_M22(  )
      goto 100
  115 continue
c—————————————————————————————————★Model_No.15 幾何学非線形+弾塑性型有限要素モデル
      call Cal_lin_stiff_M15(  )
      goto 100
  116 continue
c—————————————————————————————————★Model_No.16 両端アナロジーモデル
      call Cal_lin_stiff_M31(  )
      goto 100
```

```
  117 continue
c――――――――――――――――――――――――――――――――――――――――――――――――――――――――★Model_No.17 両端、中央アナロジーモデル
      call Cal_lin_stiff_M32(  )
      goto 100
  118 continue
c――――――――――――――――――――――――――――――――――――――――――――――――――――――――★Model_No.18 両端ピン、中央ファイバーモデル
      call Cal_lin_stiff_M13(  )
      goto 100
  119 continue
c――――――――――――――――――――――――――――――――――――――――――――――――――――――――★Model_No.19 両端ピン、中央アナロジーモデル
      call Cal_lin_stiff_M33(  )
      goto 100
  120 continue
c――――――――――――――――――――――――――――――――――――――――――――――――――――――――★Model_No.20
      goto 100
      endif
 9999 continue
  100 continue
      end do
      return
      end
```

プログラムを見れば理解できるように、非常に単純な構成となっている。この単純さがバグを発生させない要因となる。

さらに、階層構造を深く見るために、Cal_check_stiff_M2()を調べてみよう。このサブルーチンは、先に示したCheck_stress()の中で同様な使い方がされている。このサブルーチンもその構造のみ載せる。全文は第5.8節で示されているので、そこを参照されたい。

プログラム内のコメント番号１は、せん断型モデルの履歴番号にしたがって、該当する履歴モデルを弾塑性チェックするプログラムに、制御を移動させる命令である。同じく２は、武田モデルの弾塑性履歴モデルを数値計算するサブルーチンである。武田モデルは、履歴番号が３であるが、規定履歴モデルとしても使用されており、履歴番号が０のときもこのサブルーチンを使用するようになっている。この履歴モデルの作成法については次章で述べる。

このように階層構造を表すプログラムコードは非常に単純に設計されているため、新たな履歴モデルをSPACEに組み込む作業はさほど難しくない。この件については、第９章で詳細に解説する。

```
C     ――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_check_stiff_M2
C       ――――――――――――――――――――――――――――――――――――
C     ●   Model_No.2 3 次元せん断弾塑性モデル
C     ――――――――――――――――――――――――――――――――――――
      subroutine Cal_check_stiff_M2(  )
c――――――――――――――――――――――――――――――――――――――――――――――――――――――――
```

```
      No_rireki=Element.nm_type
      if(No_rireki/10.eq.0) then
      goto(5,10,20,30,40,50,60),No_rireki+1                    ! 1
    5 continue
c――――――――――――――――――――――――――――――――★規定モデル：武田モデル
      call Takeda_TriLiner(Member,Element,vv,vpp)              ! 2
      goto 999
   10 continue
c――――――――――――――――――――――――――――――――★トリリニア：Nomal
      call Mesing_TriLiner(Member,Element,vv,vpp)              ! 3
      goto 999
   20 continue
c――――――――――――――――――――――――――――――――★トリリニア：最大点指向型
      call DirecMax_TriLiner(Member,Element,vv,vpp)
      goto 999
   30 continue
c――――――――――――――――――――――――――――――――★トリリニア：武田モデル
      call Takeda_TriLiner(Member,Element,vv,vpp)              ! 4
      goto 999
   40 continue
c――――――――――――――――――――――――――――――――★欠番
      goto 999
   50 continue
c――――――――――――――――――――――――――――――――★
      goto 999
   60 continue
c――――――――――――――――――――――――――――――――★
      goto 999
      elseif(No_rireki/10.eq.1) then
      goto(101,102),No_rireki - 10
  101 continue
c――――――――――――――――――――――――――――――――★修正バイリニアモデル
      mro=Element.n_section(1)
      call Modify_Bi_Liner1(Member,Element,RO_work(mro),vv,vpp)
      goto 999
  102 continue
c――――――――――――――――――――――――――――――――★修正 RO モデル
      mro=Element.n_section(1)
      call Modify_RO1(Member,Element,RO_work(mro),vv,vpp)
      goto 999
      endif
  999 continue
      return
      end
```

5.4 静的縮合モデルの作成手法

本節では、静的縮合モデルの一般的な取り扱い方法について述べる。内部節点を有する部材モデルでは、弾塑性挙動を効果的に実現できるように、両端、及び中央部にファイバーエレメントやアナロジーエレメントが配置される。この部材内部に異なった内部エレメントを持ち、しかも、静的縮合を行って部材両端において各種の物理量を評価するモデル

を静的縮合モデルという。ここでは、この静的縮合モデルの一般的な解析手法について述べる。

静的縮合モデル内の節点変位を、内部節点の増分変位ベクトル$\{\Delta u_i\}$と部材両端の変位ベクトル$\{\Delta u_o\}$とに分けて表し、また接線剛性行列も分割して表現する。その結果、この部材モデルの釣合式は以下のように表される。

$$\begin{bmatrix} [K_{1,1}] & [K_{1,2}] \\ [K_{2,1}] & [K_{2,2}] \end{bmatrix} \begin{Bmatrix} \Delta u_o \\ \Delta u_i \end{Bmatrix} = \begin{Bmatrix} f_o \\ f_i \end{Bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots (5.1)$$

ここで、$\{f_o\}$は部材両端の不釣合力であり、$\{f_i\}$は内部節点における不釣合力である。上式を用いて静的縮合を行い、部材両端における接線剛性行列を得る。上式を変更すると内部節点の増分変位は、

$$\{\Delta u_i\} = [K_{2,2}]^{-1} \left(\{f_i\} - [K_{2,1}] \{\Delta u_o\} \right) \quad \cdots\cdots\cdots\cdots\cdots\cdots (5.2)$$

となる。内部節点の増分変位を釣合式(5.1)の上の式に代入することで、両端変位に関する接線剛性行列と不釣合力を用いた釣合式が得られる。

$$[K]\{\Delta u_o\} = \{f\} \quad \cdots\cdots\cdots\cdots\cdots\cdots (5.3)$$

ここで、接線剛性行列$[K]$と両端での不釣合力$\{f\}$は、

$$\left.\begin{aligned} \{f\} &= \{f_o\} - [K_{2,2}]^{-1} \{f_i\} \\ [K] &= \left[[K_{1,1}] - [K_{1,2}][K_{2,2}]^{-1}[K_{2,1}] \right] \end{aligned}\right\} \cdots\cdots\cdots\cdots (5.4)$$

として求められる。以上のように、上記の式によって静的縮合モデルが構成される。

前節では、部材モデルの下に履歴モデルという比較的単純な階層構造を持つせん断型モデルについて説明した。本節では、静的縮合を利用した複雑な階層構造を有するファイバーモデルについて解説する。この階層構造を実現するプログラムコードの骨組みを以下に示す。全文は、付録を参照されたい。例はモデル番号 11 の両端ファイバーモデルである。関連するサブルーチンは、以下の３つであり、このサブルーチンをコールするルーチンは、各々、Cal_stiff_linear()、Check_stress()、及び、Get_nonlinear_stff()である。先に示した階層構造を有するサブルーチ

ンの中で、上記以外の 3 つ Set_initial_data()と Cal_damp_linear()、及び Cal_mass_linear()において、両端ファイバーモデルでは、実際に何もしていないので、ここでは説明を省くことにする。以下に、３つのサブルーチンをコールする文を記す。

```
c―――――――――――――――――――――――――――――――★Model_No.11 両端ファイバーモデル
      call Cal_lin_stiff_M11(Model_type,Member(i),Element(ie),
     *      ak_linear(1,1,i) ,E_model11, E_model_fiber,
     *      M_model11, M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work)

c―――――――――――――――――――――――――――――――★Model_No.11 両端ファイバーモデル
      call Cal_check_stiff_M11(Control,N_analysis,
     *      mem,Model_type,Member(i),Element(ie),
     *      E_model11, E_model_fiber,
     *      M_model11, M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work,vv,vpp,f)

c―――――――――――――――――――――――――――――――★Model_No.11 両端ファイバーモデル
      call Cal_nonlin_stiff_M11(N_analysis,
     *      Model_type,Member(i),Element(ie),
     *      ak ,ier,E_model11, E_model_fiber,
     *      M_model11 , M_model_fiber)
```

上記 3 つのサブルーチンにおけるプログラム構造はほぼ同じである。ここでは、Cal_check_stff_M11()のコードの骨組みを以下に示す。全文は、付録を参照のこと。計算の流れを良く理解されたい。

このサブルーチンは、部材両端の増分変位から部材内部の節点増分変位を求め、その変位より内部応力を求める。さらに、部材の増分応力からファイバーの弾塑性状態をチェックし、ファイバーの応力―ひずみ状態が変化した場合は、ファイバーに関する構造体 M_model_fiber の情報を書き換える。それでは、このプログラムの流れを説明しよう。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_check_stiff_M11
C       ―――――――――――――――――――――――――――――――――
C     ●   代表的な部材モデルの塑性チェック(両端ファイバーモデル)
C     ―――――――――――――――――――――――――――――――――
      subroutine Cal_check_stiff_M11()
C     ―――――――――――――――――――――――――――――――――
      iet = Member.n_model                    ! モデルタイプ番号              ! 1
      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―――――――――――――――――――――――――――――★動的記憶領域の確保
       ALLOCATE (                                                       ! 2
     *   irest_Point(6,n_div+1),n_type(n_div),alength(n_div),EA(n_div),
     *    akk(12,12,n_div)
     *          )
c―――――――――――――――――――――――――――――★節点拘束表の作成
c                                                 未知数等をセット
      call set_model11_dat(  )                                         ! 3
c―――――――――――――――――――――――――――――★動的記憶領域の確保
       ALLOCATE (                                                       ! 4
     *   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                                                        ! 5
       do j=1,12
       ak(j,i)=0.
       enddo
       enddo
         ・
         ・
c―――――――――――――――――――――――――――――★部材剛性行列の作成
      it = 0
      do i=1,n_div                                                     ! 6
c――――――――――――――――――――――――★内部部材の剛性行列の計算
      call Stiff_M11(  )
      if(N_analysis.ne.9.and.N_analysis.ne.7.and.n_type(i).ne.3) then  ! 7
      call Cal_geomet_stiffx(  )                                       ! 8
      call Create_Kn(  )                                               ! 9
      endif
c――――――――――――――――――――――――★剛性行列の分配
      call Bnd_FEM(  )                                                 ! 10
      enddo
c―――――――――――――――――――――――――――――――――――――――
c
c     ★          部材内部の変位と不釣合力の計算
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(  )                                    ! 11
c――――――――――――――――――――――――★部材内部変位の計算(c 行列の分解計算)
      call Typical_member_v(  )                                        ! 12
c――――――――――――――――――――――――★部材内部、両端節点力の計算、両端節点力から不釣合力へ
      it = 0
      do i=1,n_div
      call Typical_member_p_force(  )                                  ! 13
      enddo
c――――――――――――――――――――――――★両端節点力を構造体へ保存する
      do i=1,n_if
       M_model11(immm).ff_ip(i)=f1(i)                                   ! 14
       enddo
c――――――――――――――――――――――――★部材両端節点力への縮合
```

```
      call Typical_member_f(  )                                          ! 15
c―――――――――――――――――――――――――★部材節点力の両端剛域処理
      call Deal_Rigid_element_f(  )                                      ! 16
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
      vvx(ii)=vv(-irest)                                                 ! 17
       elseif(irest.gt.0) then
      vvx(ii)=bav(irest)                                                 ! 18
      else
      vvx(ii)=0.                                                         ! 19
       endif
       enddo
       enddo
c――――――――――――――――――――――――――――★ファイバーチェック
      if(n_type(i).eq.2) then
       it=it+1
      call Fiber_check(  )                                               ! 20
      endif
c――――――――――――――――――――――――――――★弾性部材の軸力計算（幾何剛性作成用）
      call nonlinear_stress_N()
      Member.an_stress(i)=Member.an_stress(i)+fnn                        ! 21
c――――――――――――――――――――――――――――★部材内部変位を足しこむ
      Member.an_vv(i)=Member.an_vv(i)+(vvx(8) - vvx(2))                  ! 22
      Member.an_ww(i)=Member.an_ww(i)+(vvx(9) - vvx(3))                  ! 23
      enddo
c――――――――――――――――――――――――――――★動的記憶領域の解放
      DEALLOCATE (                                                       ! 24
     *   c ,ab ,irest_point,n_type,alength,EA,
     *   bav,akk,f1    )
       return
       end
```

プログラムの内容をコードに従って説明する。このプログラムは、両端にファイバーモデル有する部材モデルである。少し複雑なのでコードを良く理解し、数値計算の流れを確実に把握して頂きたい。

1. 構造体からモデルのタイプ番号や要素番号と取り出す。取得したパラメータの意味はコメントで示されている。

2. このサブルーチンの中でワーク領域として使用する動的領域を確保する。
3. 両端ファイバー部材モデルで必要となる未知数やテーブルを、サブルーチン set_model11_dat()で作成する。このサブルーチンの内容は第 5.8.4 節で説明する。
4. 未知数やバンド幅を用いて動的領域を確保する。
5. ワーク領域である剛性行列 ak などをゼロクリアする。
6. この部材モデルの分割数 n_div 分、7 から 10 までの処理を行い、式(5.1)に示す 4 つに分割した剛性行列を得る。ここで示す各サブルーチンの内容は後節で説明する。まず、階層構造を有するサブルーチン Stiff_M11()によって分割したエレメントの剛性行列を作成する。
7. 解析種別が幾何学的非線形性で、しかも、その部材が線形部材でない場合は以下の処理を行う。
8. サブルーチン Cal_geomet_stiffx()を用いて、そのエレメントの幾何学的非線形剛性を計算し、6. で得た剛性行列に足しこむ。
9. サブルーチン Create_Kn()を用いて、そのエレメントの大変位剛性行列を計算し、上記の剛性に足しこむ。
10. サブルーチン Bnd_FEM()を用いて、6, 7, 8 で得た剛性行列を４つに分割して、部材の全体剛性行列にセットする。
11. 部材両端に剛域を有している場合、Deal_Rigid_element_v()サブルーチンを用いて、外部節点変位をその剛域内部の節点変位に変換する。
12. サブルーチン Typical_member_v()を用いて、式(5.2)により外部節点変位から内部節点変位を求める。
13. サブルーチン Typical_member_p_force()を用いて、上で求めた内部節点変位より分割したエレメントの両端節点力を求め、さらに求めた両端節点力を各節点で和を取ることで不釣合力を求める。
14. 次ステップの内部節点における不釣合力として利用するために、構造体に保存する。
15. サブルーチン Typical_member_f()を用いて、式(5.4)の上式より部材両端の節点力を求める。
16. 部材に剛域がある場合、上で求めた部材両端の節点力をサブルーチン Deal_Rigid_element_f()を用いて、剛域外部の節点力に変換する。
17. 次からの処理は、部材内部のエレメントについて応力のチェックを行う。まず、部材内部エレメントの両端変位をセットする。部材内

部節点の未知数番号表である irest_Point が、負の場合、これは外部節点であるため、外部節点の変位 vv をセットする。

18. また、正の場合、これは内部節点であるため、内部節点の変位 bav をセットする。
19. ゼロの場合は、固定であるため、0 をセットする。
20. エレメントのタイプ番号 n_type が 2 で場合、ファイバー要素であることからサブルーチン Fiber_check()を用いて、ファイバーの弾塑性チェックを行う。
21. 部材内エレメントの非線形性を考慮した増分軸力 fnn を、サブルーチン non_linear_stress_N()により計算し、構造体成分である Member.an_stress に足しこむ。
22. 次回の幾何学的非線形剛性を求めるために必要となる内部エレメントの面外変位 v を、構造体成分 Member.an_vv に足しこむ。
23. 同じく、エレメントの面外変位 w を構造体成分 Member.an_ww に足しこむ。
24. ワーク用の動的領域を解放する。

次に、他の２つのサブルーチンを示す。プログラム構造はほとんど同じであるので、理解は容易であると思う。また、プログラムコード右側に付したコメントで、通常の番号は上で説明したものと同一である。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_lin_stiff_M11
C       ―――――――――――――――――――――――――――――――――
C     ●   代表的な部材モデルの剛性(両端ファイバーモデル)
C     ―――――――――――――――――――――――――――――――――
      subroutine Cal_lin_stiff_M11(  )
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_model11_s / E_model11
      record / E_model_fiber_s    / E_model_fiber
      record / M_model11_s        / M_model11
      record / M_model_fiber_s    / M_model_fiber
      record / Bilinear_work_s    / Bilinear_work
      record / Trilinear_work_s   / Trilinear_work
      record / Concrete_work_s    / Concrete_work
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension E_model11(*),M_model11(*)
      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              ! 1
      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 (                                                          ! 2
     *   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_model11_dat(  )                                            ! 3
c―――――――――――――――――――――――――――――――――★動的記憶領域の確保
      ALLOCATE (                                                          ! 4

     *   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                                                           ! 5
      do j=1,12
      ak(j,i)=0.
      enddo
      enddo
        •
        •
      do i=1,n_div
      Member.an_vv(i)=0.                                                  ! x1
      Member.an_ww(i)=0.
      enddo
c―――――――――――――――――――――――――――――――――★部材剛性行列の作成
      it = 0
      do i=1,n_div                                                        ! 6
      call Stiff_M11_I( )                                                 ! x2
      call Bnd_FEM( )                                                     ! 10
      enddo
c―――――――――――――――――――――――――――――――――★部材剛性行列の縮合
      call Typical_member_model( )                                        ! x3
c―――――――――――――――――――――――――――――――――★両端の剛域処理
      call Deal_Rigid_element( )                                          ! x4
c―――――――――――――――――――――――――――――――――★動的記憶領域の解放
      DEALLOCATE (                                                        ! 24
     *   c ,ab ,ba,irest_point,n_type,alength
     *         )
      return
      end
C     ―――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_nonlin_stiff_M11
C       ―――――――――――――――――――――――――――――――――――
C     ●   代表的な部材モデルの接線剛性(両端ファイバーモデル)
C     ―――――――――――――――――――――――――――――――――――――
      subroutine Cal_nonlin_stiff_M11( )
```

```
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_model11_s       / E_model11
      record / E_model_fiber_s   / E_model_fiber
      record / M_model11_s       / M_model11
      record / M_model_fiber_s   / M_model_fiber
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension E_model11(*),M_model11(*)
      dimension ak(12,12),akk(12,12)
      real*8, ALLOCATABLE :: c(:,:),ab(:,:),ba(:),alength(:),EA(:)
      integer, ALLOCATABLE :: irest_Point(:,:),n_type(:)
C     ―――――――――――――――――――――――――――――――――――――
      iet = Member.n_model                    ! モデルタイプ番号                  ! 1
      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 (                                                          ! 2
     *   irest_Point(6,n_div+1),n_type(n_div),alength(n_div),EA(n_div)
     *         )
c―――――――――――――――――――――――――――――――――――――――★節点拘束表の作成
c                                             未知数等をセット
      call set_model11_dat( )                                             ! 3
c―――――――――――――――――――――――――――――――――――――――★動的記憶領域の確保
      ALLOCATE (                                                          ! 4
     *   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                                                           ! 5
      do j=1,12
      ak(j,i)=0.
      enddo
      enddo
        •
        •
      EAx=Element.A*Element.E                                             ! x5
      it=0
      do i=1,n_div
      EA(i)=EAx
      if(n_type(i).eq.2) then
      it=it+1
      if(it.eq.1) then
      EA(i)=M_model11(immm).d_ra_1
      else
      EA(i)=M_model11(immm).d_ra_2
      endif
      endif
      enddo
```

```
c―――――――――――――――――――――――――――――――★部材剛性行列の作成
      it = 0
      do i=1,n_div                                                    ! 6
      call Stiff_M11( )
      if(N_analysis.ne.9.and.N_analysis.ne.7.and.n_type(i).ne.3) then ! 7
      call Cal_geomet_stiffx( )                                       ! 8
      call Create_Kn( )                                               ! 9
      endif
      call Bnd_FEM( )                                                 ! 10
      enddo
c―――――――――――――――――――――――――――――――★部材剛性行列の縮合
      call Typical_member_model( )                                    ! x3
c―――――――――――――――――――――――――――――――★両端の剛域処理
      call Deal_Rigid_element( )                                      ! x4
c―――――――――――――――――――――――――――――――★動的記憶領域の解放
      DEALLOCATE (                                                   ! 24
     *   c ,ab ,ba,irest_point,n_type,alength,EA
     *         )
      return
      end
```

プログラムの内容をコードに従って説明する。このプログラムは、両端ファイバーモデルの部材モデルに関する線形剛性と接線剛性を求めるものである。

x1. 初期設定として、幾何学的非線形剛性を求めるための部材内部エレメントの面外変位をゼロクリアする。

x2. サブルーチン Stif_M11_I()を用いて、部材内エレメントの線形剛性行列を求める。

x3. サブルーチン Typical_member_model()を用いて、式(5.4)の下式より、部材剛性行列の静的縮合を行う。

x4. 部材の両端もしくはどちらか一方に剛域を有している場合、上記で得た剛性行列を、サブルーチン Deal_Rigid_element()を用いて座標変換する。

x5. 部材内部エレメントの幾何学的非線形剛性を求めるとき、部材の軸方向剛性 EA が必要となる。ファイバーエレメントの場合、塑性域のファイバーを考慮した断面剛性 M_model11(immm).d_ra_1 を EA としてセットする。

次に、静的縮合を実行するサブルーチンについて解説する。この静的縮合に関連するサブルーチンは、以下の４つである。

```
c―――――――――――――――――――――――――――――――★部材剛性行列の縮合
      call Typical_member_model( )
```

```
c―――――――――――――――――――――――――★部材内部変位の計算(c 行列の分解計算)
      call Typical_member_v(  )
c―――――――――――――――――――――――――★部材内部、両端節点力の計算
      call Typical_member_p_force(  )
c―――――――――――――――――――――――――★部材両端節点力への縮合
      call Typical_member_f(  )
```

また、剛域に関係する座標変換を行うサブルーチンについては、第 5.6 節で説明するので、そちらを参照されたい。

以下に、上記の４つのサブルーチンの内容を示す。

```
C     ――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Typical_member_model
C     ――――――――――――――――――――――――――――――――――――
C     ●   代表的な部材モデルの剛性の縮合計算(ok)
C     ――――――――――――――――――――――――――――――――――――
      subroutine Typical_member_model(c,ab,ak,n,nx,nu,nux,ba,ier)
      implicit real*8(A-H,O-Z)
      dimension ak(12,12),f(12),f1(nx)
      dimension  c(0:nux,*),ab(nx,12),ba(nx)
c
c                n        :  部材の内部自由度
c                nx       :  部材の内部自由度の記憶域大きさ
c                nu       :  内部自由度の半バンド幅
c                nux      :  内部自由度の半バンド幅の記憶域大きさ
c
c                c(0:nux,nx):  内部自由度に関する剛性（半バンド行列)
c                ak(12,12):  in: 材端自由度に関連する剛性
c                            out:縮合された剛性
c                ab(nx,12):  材端と内部の相互の剛性
c                ba(nx)   :  ワークエリア
c――――――――――――――――――――――――――――――――★ｃ行列の分解:
      eps=0.000000000001
      ier=0
      call Decom_B( c, n, nux,nu,eps, ba , ier )
      if(ier.ne.0) return
c――――――――――――――――――――――――――――――――★剛性行列の縮合
      do 100 i=1,12
      do j=1,n
      ba(j)=ab(j,i)
      enddo
      call Solv_B( c, n, nux, nu, ba  )
      do j=i,12
      sum=0.0
      do k=1,n
      sum=sum+ab(k,j)*ba(k)
      enddo
      ak(j,i)=ak(j,i)-sum
      ak(i,j)=ak(j,i)
      enddo
  100 continue
      return
```

```
      end
C     ―――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Typical_member_v
C     ―――――――――――――――――――――――――――――――――――――
C     ●   代表的な部材モデルの内部節点変位の計算(ok)
C     ―――――――――――――――――――――――――――――――――――――
      subroutine Typical_member_v(c,ab,f1,n,nx,nu,nux,ier,vv,bav)
      implicit real*8(A-H,O-Z)
      dimension vv(12),f1(nx)
      dimension  c(0:nux,*),ab(nx,12)
      dimension  bav(nx)
c
c               n       :  部材の内部自由度
c               nx      :  部材の内部自由度の記憶域大きさ
c               nu      :  内部自由度の半バンド幅
c               nux     :  内部自由度の半バンド幅の記憶域大きさ
c
c               c(0:nux,nx):  内部自由度に関する剛性（半バンド行列）
c               ab(nx,12):  材端と内部の相互の剛性
c               vv(12)  :  両端の変位
c               bav(nx) :  内部の節点変位
c               f1(nx)  :  部材内部節点力
c――――――――――――――――――――――――――――――――★ｃ行列の分解:
      eps=0.000000000001
      ier=0
      call Decom_B( c, n, nux,nu,eps, bav , ier )    ! ここでの bav()はワークエリアとして使用
      if(ier.ne.0) return
c――――――――――――――――――――――――――――――――★f1 - ab*v
      do j=1,n
      sum=0.
      do k=1,12
      sum=sum+ab(j,k)*vv(k)
      enddo
      bav(j)=-sum + f1(j)
      enddo
c――――――――――――――――――――――――――――――――★c*bav = f1 - ab*v  を解く
      call Solv_B( c, n, nux, nu, bav  )          ! bav()は内部節点変位
      return
      end
C     ―――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Typical_member_f
C     ―――――――――――――――――――――――――――――――――――――
C     ●   代表的な部材モデルの節点力の縮合
C     ―――――――――――――――――――――――――――――――――――――
      subroutine Typical_member_f(c,ab,f1,f,n,nx,nu,nux)
      implicit real*8(A-H,O-Z)
      dimension f1(nx),f(12)
      dimension  c(0:nux,*),ab(nx,12)
c
c               n       :  部材の内部自由度
c               nx      :  部材の内部自由度の記憶域大きさ
c               nu      :  内部自由度の半バンド幅
c               nux     :  内部自由度の半バンド幅の記憶域大きさ
c
```

```
c                 c(0:nux,nx):  内部自由度に関する剛性（半バンド行列）
c                 ab(nx,12):  材端と内部の相互の剛性
c                 vv(12)    :  両端の変位
c                 bav(nx)   :  内部の節点変位
c                 f1(nx)    :  部材内部節点力   !  データの変更あり
c                 f(12)     :  両端の節点力
c————————————————————————————————★節点力の縮合
       call Solv_B( c, n, nux, nu, f1  )
       do j=1,12
      sum=0.0
      do k=1,n
      sum=sum+ab(k,j)*f1(k)
      enddo
      f(j)=f(j)-sum
      enddo
      return
      end
C     ———————————————————————————————————————
C     ●   SUBROUTINE /Typical_member_p_force
C     ———————————————————————————————————————
C     ●   代表的な部材モデルの内部節点合応力の計算(ok)
C     ———————————————————————————————————————
      subroutine Typical_member_p_force(akk,irest_Point,disp_p,
     *                                       disp_ip,f,f1)
      implicit real*8(A-H,O-Z)
       dimension  disp_p(12),disp_ip(*)
      dimension akk(12,12),f(12),f1(*),v(12)
      dimension irest_Point(*)
c————————————————————————————————★内部部材の両端変位の取り出し
      do i = 1,12
      irest=irest_Point(i)
      if(irest.gt.0)then
      v(i)=disp_ip(irest)             ! 内部節点変位
      elseif(irest.lt.0) then
      v(i)=disp_p(-irest)             ! 両端の変位
      else
      v(i)=0.
      endif
      enddo
c————————————————————————————————★内部部材両端の節点力を計算
      do 100 i=1,12
      irest=irest_Point(i)
      sum=0.
      if(irest.ne.0) then
      do j=1,12
      sum=sum+akk(i,j)*v(j)
      enddo
      if(irest.gt.0) then
      f1(irest)=f1(irest)+sum       ! 内部節点の節点力
      elseif(irest.lt.0) then
      f(-irest)=f(-irest)+sum       ! 両端節点の節点力
      endif
      endif
 100  continue
```

```
return
end
```

上記４つのサブルーチンは、静的縮合に関連する標準的なプログラムである。最初のサブルーチン Typical_member_model()は、式(5.5)を計算して静的縮合を行い、部材の両端変位に対する剛性行列を求める。ここで、行列$\left[K_{2,2}\right]$は、実対称バンド行列である。無論、ここでは、行列$\left[K_{2,2}\right]$の逆行列を計算しておらず、別の方法でこの計算を実行している。この手法を以下に示す。まず、式(5.5)の下式で示す方程式を作り、その解を求める。

$$\left.\begin{aligned}\left[K\right]&=\left[\left[K_{1,1}\right]-\left[K_{1,2}\right]\left[K_{2,2}\right]^{-1}\left[K_{2,1}\right]\right]\\ \left[K_{2,2}\right]\left[b\right]&=\left[K_{2,1}\right]\end{aligned}\right\}\cdots\cdots\cdots\cdots(5.5)$$

その解行列$\left[b\right]$に行列$\left[K_{1,2}\right]$を前から掛け、その結果を$\left[K_{1,1}\right]$から引き算するわけである。多少分かりにくいが、サブルーチンと式を比較すれば、理解できるであろう。なお、サブルーチン内の配列と式記号との対応は、$\left[K_{1,1}\right]$は配列 ak であり、$\left[K_{2,2}\right]$は配列 c、$\left[K_{2,1}\right]$は配列 ab である。

次のサブルーチン Typical_member_v()は、部材内部の節点変位を求める。内部節点変位を求める式は、以下のようである。

$$\left\{\Delta u_i\right\}=\left[K_{2,2}\right]^{-1}\left(\left\{f_i\right\}-\left[K_{2,1}\right]\left\{\Delta u_o\right\}\right)\qquad\cdots\cdots\cdots\cdots\cdots\cdots(5.6)$$

このサブルーチンでは、最初に、行列$\left[K_{2,2}\right]$の LDU 分解を行う。次に、右辺項 { } 内の計算を行い、配列 bav にセットする。最後にサブルーチン Solv_B()を用いて、内部節点変位を求める。

サブルーチン Typical_member_f()は、部材両端の節点力を求めるプログラムである。ここでは、既に、行列$\left[K_{2,2}\right]$は LDU 分解されているものとする。両端の節点力を求める式は以下のようである。

$$\left\{f\right\}=\left\{f_o\right\}-\left[K_{2,2}\right]^{-1}\left\{f_i\right\}\qquad\cdots\cdots\cdots\cdots\cdots\cdots(5.7)$$

プログラムは非常に単純などで容易に理解できると思う。

最後のサブルーチン Typical_member_force()は、内部節点と部材両端の応力を求めるプログラムである。このサブルーチンでは、ある部材について、両端の節点力を求め、その節点力を部材の各節点に足しこみ、部材全体の、内部節点と部材両端の応力を求めている。部材内の各エレ

メントの釣合式は、

$$\{f\} = [K]\{\Delta u\} \qquad \cdots\cdots\cdots\cdots\cdots\cdots (5.8)$$

である。まず、部材内エレメントの両端変位 v をセットする。次にこの変位と剛性係数 akk と掛け算を行い、その値を内部節点と部材両端の節点に分けて、その節点応力に足しこむ。

以上が静的縮合モデルで使用される汎用のサブルーチンである。このサブルーチンは、他の部材モデルにも使用されており、それらは付録を参照されたい。

5.5 データ管理法

解析モデルが大きくなると、ファイバーデータは、多量のデータ格納領域を必要とする。そのため、メモリーをよほど効率良く使用しなければ、直ぐに仮想メモリーを使用しなければならなくなる。ここでは、SPACE で用いているファイバーモデルのデータ領域管理技法を見てみよう。部材モデルは、モデル番号 11 の両端ファイバーモデルを例にとる。

ファイバーのデータ領域は、次の 2 つの構造体で定義されている。

```
record / E_model_fiber_s   / E_model_fiber
record / M_model_fiber_s   / M_model_fiber
```

この２つの構造体は、submainx.h というインクルードファイルで定義されおり、その内容は構造体中の成分右側にコメントとして記述されている。そのコメントを参照すれば、構造体成分の内容は概略理解できる。上の E_model_fiber_s は要素データを表し、入力したデータが格納され、解析途中で変更されることはない。また、下の M_model_fiber_s は、部材データに関連し、存在するファイバー全部に割り付けられる。したがって、解析途中でその値が変化する成分もある。

さらに、この部材モデル関連する構造体は、以下のように定義されている。

```
record / E_model11_s    / E_model11
record / M_model11_s    / M_model11
```

まず、関連する構造体を以下に示そう。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   ファイバーモデル E_model_fiber_s 構造体（バイリニアー、トリリニア用）
C     ―――――――――――――――――――――――――――――――――
c
```

```
c     部材
      structure / E_model_fiber_s/
      integer  nm_type              !  履歴モデルの番号
      real*8   E_1                  !  ファイバーの第一剛性 E1
      real*8   E_2                  !  ファイバーの第二剛性 E2
      real*8   E_3                  !  ファイバーの第三剛性 E3
      real*8   Q_1                  !  ファイバーの第一折れ点
      real*8   Q_2                  !  ファイバーの第二折れ点
      real*8   Ec_1                 !  ファイバーの圧縮側第一剛性 E1
      real*8   Ec_2                 !  ファイバーの圧縮側第二剛性 E2
      real*8   Ec_3                 !  ファイバーの圧縮側第三剛性 E3
      real*8   Qc_1                 !  ファイバーの圧縮側第一折れ点
      real*8   Qc_2                 !  ファイバーの圧縮側第二折れ点
      real*8   G                    !  ファイバーせん断剛性 G
      real*8   A                    !  ファイバー断面積
      real*8   Ay                   !  ファイバーｙ軸せん断用断面積
      real*8   Az                   !  ファイバーｚ軸せん断用断面積
      real*8   ry                   !  中立軸から断面中心までのｙ方向距離
      real*8   rz                   !  中立軸から断面中心までのｚ方向距離
      real*8   Ary                  !  断面積掛ける距離
      real*8   Arz                  !  断面積掛ける距離
      real*8   Ary2                 !  断面積掛ける距離の２乗
      real*8   Arz2                 !  断面積掛ける距離の２乗
      real*8   Aryz                 !  断面積掛ける距離の２乗
      real*8   Beta                 !  移動硬化用パラメータ
      real*8   Beta_2               !  移動硬化用パラメータその２
      end structure
c       record / E_model_fiber_s    / E_model_fiber
c
C     ―――――――――――――――――――――――――――――――――
C     ●　ファイバーモデル E_model_fiber_s 構造体
C     ―――――――――――――――――――――――――――――――――
c
c     部材
      structure / E_model_fiberc_s/
      integer  nm_type              !  履歴モデルの番号
      real*8   AK_1                 !  圧縮と引張　第一勾配
      real*8   AK_2                 !  圧縮第二勾配
      real*8   AK_3                 !  圧縮第三勾配
      real*8   Q_1                  !  引張強度
      real*8   Q_2                  !  圧縮側第一折れ点の応力
      real*8   Q_3                  !  圧縮強度
      real*8   Q_4                  !  圧縮流れ点
      real*8   AK_4                 !  引張第二勾配
      real*8   dm                   !  ダミー
      real*8   dm1                  !  ダミー
      real*8   G                    !  ファイバーせん断剛性 G
      real*8   A                    !  ファイバー断面積
      real*8   Ay                   !  ファイバーｙ軸せん断用断面積
      real*8   Az                   !  ファイバーｚ軸せん断用断面積
      real*8   ry                   !  中立軸から断面中心までのｙ方向距離
      real*8   rz                   !  中立軸から断面中心までのｚ方向距離
      real*8   Ary                  !  断面積掛ける距離
      real*8   Arz                  !  断面積掛ける距離
```

```
      real*8   Ary2                 !  断面積掛ける距離の２乗
      real*8   Arz2                 !  断面積掛ける距離の２乗
      real*8   Aryz                 !  断面積掛ける距離の２乗
      real*8   Beta                 !  移動硬化用パラメータ
      real*8   Beta_2               !  移動硬化用パラメータその２
      end structure
c       record / E_model_fiberc_s    / E_model_fiber
c
C     ―――――――――――――――――――――――――――――――――
C     ●   ファイバーモデル M_model_fiber_s 構造体
C     ―――――――――――――――――――――――――――――――――・
c
c     部材
      structure / M_model_fiber_s/
      integer  n_type               !  履歴モデルの通し番号
      integer  i_elastic            !  ファイバー要素の状態（弾性の場合は０：塑性は１）
      real*8   d_eps_x              !  軸方向歪
      real*8   d_stress_x           !  軸方向応力
      real*8   d_E                  !  接線剛性
      end structure
c       record / M_model_fiber_s    / M_model_fiber
c
C     ―――――――――――――――――――――――――――――――――
C     ●   Model_No.11 両端ファイバーモデル E_model11_s 構造体
C     ―――――――――――――――――――――――――――――――――
c
c     部材
      structure / E_model11_s/
      integer  nm_section_1         !  i 端断面のファイバー開始番号
      integer  n_section_1          !  i 端断面のファイバー数
      integer  nm_section_2         !  j 端断面のファイバー開始番号
      integer  n_section_2          !  j 端断面のファイバー数
      end structure
c       record / E_model11_s    / E_model11
c
C     ―――――――――――――――――――――――――――――――――
C     ●   Model_No.11 両端ファイバーモデル M_model11_s 構造体
C     ―――――――――――――――――――――――――――――――――
c
c     部材
      structure / M_model11_s/
      integer  nm_section_1         !  i 端断面のファイバー開始番号
      integer  n_section_1          !  i 端断面のファイバー数
      real*8  d_aa_1                !  断面積の和
      real*8  d_ra_1                !  E*断面積の和
      real*8  d_ray_1               !  E*A*z
      real*8  d_raz_1               !  E*A*y
      real*8  d_raz2_1              !  E*A*y*y
      real*8  d_ray2_1              !  E*A*z*z
      real*8  d_rayz_1              !  E*A*z*y
      real*8  d_gg_1                !  G*A
      real*8  d_epsi_x_1            !  軸方向歪
      real*8  d_epsi_y_1            !  y 軸に関する曲げ歪
      real*8  d_epsi_z_1            !  z 軸に関する曲げ歪
```

```
      integer  nm_section_2          !  j 端断面のファイバー開始番号
      integer  n_section_2           !  j 端断面のファイバー数
      real*8  d_aa_2                 !  断面積の和
      real*8  d_ra_2                 !  E*断面積の和
      real*8  d_ray_2                !  E*A*z
      real*8  d_raz_2                !  E*A*y
      real*8  d_raz2_2               !  E*A*y*y
      real*8  d_ray2_2               !  E*A*z*z
      real*8  d_rayz_2               !  E*A*z*y
      real*8  d_gg_2                 !  G*A
      real*8  d_epsi_x_2             !  軸方向歪
      real*8  d_epsi_y_2             !  y 軸に関する曲げ歪
      real*8  d_epsi_z_2             !  z 軸に関する曲げ歪
      real*8  disp_p(12)             !  両端の変位
      real*8  disp_ip(30)            !  内部節点の変位
      real*8  ff_ip(30)              !  内部節点の不釣合力
      end structure
c      record / M_model11_s    / M_model11
C     ―――――――――――――――――――――――――――――――
C     ●   element_s 構造体
C     ―――――――――――――――――――――――――――――――
c
c     要素
      structure / element_s/
      integer  element_type   !  要素タイプ
      integer  n_element      !  非線形要素番号
      .
      .
C     ―――――――――――――――――――――――――――――――
C     ●   member_s 構造体
C     ―――――――――――――――――――――――――――――――
c
c     部材
      structure / member_s/
      integer  nm_element          !  要素番号（入力した要素番号を示す）
      integer  element_type        !  要素タイプ番号
      integer  n_model             !  モデルの入れ物番号
      integer  n_model_type        !  モデル別の通し番号
      integer  n_element_type      !  要素タイプ別通し番号
      .
      .
```

構造体 E_model_fiber_s は、ファイバー要素の履歴について定義しており、バイリニアー、及びトリリニアーに関するデータ領域である。また、構造体 E_model_fiber_c は、直線型コンクリートの履歴データ領域であり、この２つの定義文は異なるが、実態として構造体 E_model_fiber は同じものを指す。

要素に関連するデータは、入力データが多く計算途中変化しない。この要素に関連する構造体のデータ領域は、element_s、E_model11_s、E_model_fiber_s である。部材に関連するデータは、各部材に関連し、

計算途中で変化する。この部材に関連する構造体のデータ領域は、member_s、M_model11_s、M_model_fiber_s である。これらの構造体領域は、次のように管理されている。

まず、当該の部材が mem であるとすると、その要素は、

```
ie=Member(mem).nm_element
```

となり、また、その部材モデルが両端ファイバー部材モデルであると、そのデータへのアクセスは、次のようにして配列の要素番号を取得し、

```
imm= Element(ie).n_element          ! 要素番号
immm= Member(mem).n_model_type      ! モデルタイプ別番号
```

例えば、次のように使用する。

```
      call Stiff_M11(mem,it,n_type(mem),akk,Member,alength,
     *       Model_type,Element,
     *       E_model11(imm), E_model_fiber,
     *       M_model11(immm), M_model_fiber)
```

次に、ファイバー構造体へのアクセスは、サブルーチン Stiff_M11()の中で、次のように使用する。

```
      if(it.eq.1) then                    ! i 節点側ファイバーデータ
      nm_div = E_model.n_section_1        ! 断面内ファイバー数
      nn     = E_model.nm_section_1 - 1   ! ファイバー要素番号初期値 - 1
      nnm    = M_model.nm_section_1 - 1   ! ファイバー部材番号初期値 - 1
      elseif(it.eq.2) then                ! j 節点側ファイバーデータ
      nm_div = E_model.n_section_2
      nn     = E_model.nm_section_2 - 1
      nnm    = M_model.nm_section_2 - 1
      endif
      do i=1,nm_div
      nn     =  nn  + 1
      nnm    =  nnm + 1
c――――――――――――――――――――――――――――――★データの初期設定
      M_model_fiber(nnm).d_E        = E_model_fiber(nn).E_1
       ・
       ・
```

以上のように、ファイバーデータやモデルデータは管理されており、解析モデルの中で、関連する部材モデルが使用されていないときは、実際の構造体の動的領域は確保されないし、また、動的領域を確保するときも隙間なく効率よく使用している。この管理技法を実際のプログラムの中で確実に理解していかなければ、SPACE 動的ソルバーのメンテナンスや変更、拡張は困難であろう。なお、上記に関して先に示した表 5-2 にまとめられている。

弾性部材における３次元部材の剛性行列$[K_L]$は、以下のように表される。これについては、理論マニュアルを参照されたい。

$$[K_L]=\begin{bmatrix}
\frac{EA}{\ell} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{\ell} & 0 & 0 & 0 & 0 & 0 \\
 & \frac{12EI_z}{\ell^3} & 0 & 0 & 0 & \frac{6EI_z}{\ell^2} & 0 & -\frac{12EI_z}{\ell^3} & 0 & 0 & 0 & \frac{6EI_z}{\ell^2} \\
 & & \frac{12EI_y}{\ell^3} & 0 & -\frac{6EI_y}{\ell^2} & 0 & 0 & 0 & -\frac{12EI_y}{\ell^3} & 0 & -\frac{6EI_y}{\ell^2} & 0 \\
 & & & \frac{GI_x}{\ell} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_x}{\ell} & 0 & 0 \\
 & & & & \frac{4EI_y}{\ell} & 0 & 0 & 0 & \frac{6EI_y}{\ell^2} & 0 & \frac{2EI_y}{\ell} & 0 \\
 & & & & & \frac{4EI_z}{\ell} & 0 & -\frac{6EI_z}{\ell^2} & 0 & 0 & 0 & \frac{2EI_z}{\ell} \\
 & & & & & & \frac{EA}{\ell} & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & \frac{12EI_z}{\ell^3} & 0 & 0 & 0 & -\frac{6EI_z}{\ell^2} \\
 & & & sym & & & & & \frac{12EI_y}{\ell^3} & 0 & \frac{6EI_y}{\ell^2} & 0 \\
 & & & & & & & & & \frac{GI_x}{\ell} & 0 & 0 \\
 & & & & & & & & & & \frac{4EI_y}{\ell} & 0 \\
 & & & & & & & & & & & \frac{4EI_z}{\ell}
\end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots (5.9)$$

この弾性剛性を計算するためのサブルーチンコールは、

```
c―――――――――――――――――――――――――――――――★Model_No.1 通常の有限要素弾性モデル
      call Cal_lin_stiff_M1(Member(i),Element(ie),ak_linear(1,1,i))
         if(Member(i).i_rigid_length.ne.0..or.Member(i).j_rigid_length.ne.0.)
     *     call Deal_Rigid_element(ak_linear(1,1,i),
     *               Member(i).i_rigid_length,Member(i).j_rigid_length)
```

である。また、後のサブルーチンコールは、この部材が両端もしくはどちらかに剛域を有する場合に対応し、そこでは剛性を座標変換する。上記２つに関連するサブルーチンを以下に示す。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_lin_stiff_M1
C       ―――――――――――――――――――――――――――――――――
```

```
C     ●　線形剛性行列の計算(ok)
C     ―――――――――――――――――――――――――――――――――――
      subroutine Cal_lin_stiff_M1(Member,Element,ak_linear)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s    / Member
      record / element_s   / Element
      dimension ak_linear(12,12)
c―――――――――――――――――――――――――――――――――――――――――――――――
c        Model_No.1 = 1              !  通常の有限要素弾塑性モデル
c
c    要素数
c      structure / element_s/
c      integer  element_type   !  要素タイプ
c      integer  n_element      !  部材モデルの数
c      real*8   E              !  ヤング係数
c      real*8   G              !  せん断係数
c      real*8   A              !  断面積
c      real*8   RIx            !  ねじり剛性
c      real*8   RIy            !  ｙ軸断面二次モーメント
c      real*8   RIz            !  ｚ軸断面二次モーメント
c      real*8   ASy            !  各部材のＹ軸回りのせん断変形用等価断面積
c      real*8   ASz            !  各部材のＺ軸回りのせん断変形用等価断面積
c      real*8   AM             !  単位長さ当たりの質量
c      integer  nm_damp        !  部材減衰の有無
c      end structure
c     record /element_s/ Element
c     ALLOCATABLE ::Element(:)
c     ALLOCATE (Element(n_element))
c
c     部材数
c      structure / member_s/
c      integer  nm_element         !  要素番号
c      integer  element_type       !  要素タイプ
c      integer  n_element_type     !  要素タイプ別番号
c      integer  nm_dll_element     !  DLL を用いた要素か（０；システム内要素、１：DLL 要素）
c      integer  nm_point(2)        !  節点番号
c      integer  irest(12)          !  部材両端の自由度番号表
c      integer  ijp(2)             !  両端節点への結合状況（0:剛結合 1:ピン結合）
c      integer  nm_analysis        !  部材解析種別
c      integer  nm_group           !  部材グループ
c      integer  nm_local_coord(2)  !  局所座標系の有無とその回転行列の番号
c      integer  nm_damp            !  部材減衰の有無とその減衰行列の番号
c      real*8   alength            !  長さ
c      real*8   rot_x              !  部材主軸の回転角度（度）
c      real*8   force(12)          !  部材両端の部材端力
c      end structure
c     record / member_s / Member
c     ak_linear            real*8   線形剛性行列
c―――――――――――――――――――――――――――――――――――――――――――――――
      do i=1,12
      do j=1,12
      ak_linear(j,i) = 0.0
      end do
```

```
      end do
      rl=Member.alength -
     *  Member.i_rigid_length - Member.j_rigid_length
      young = Element.E
      ra    = Element.A
      riz   = Element.RIz
      riy   = Element.RIy
      rix   = Element.RIx
      seng  = Element.G
      IP    = Member.ijp(1)      ! i端ピン接合か（現在使用不可）
      JP    = Member.ijp(2)      ! j端ピン接合か（現在使用不可）
      asy   = Element.ASy
      asz   = Element.ASz
C     ――――――――――――――――――――――――――――――――――
C     -   立体構造部材の部材座標系での部材剛性マトリックスを作成する。
C     -
C     -   RA     : 各部材の断面積。
C     -   RIX    : 各部材のＸ軸回りの極２次モーメント。
C     -   RIY    : 各部材のＹ軸回りの断面２次モーメント。
C     -   RIZ    : 各部材のＺ軸回りの断面２次モーメント。
C     -   YOUNG  : 各部材のヤング係数（ヤング率)。
C     -   SENG   : 各部材のせん断剛性係数。
C     -   ASY    : 各部材のＹ軸回りのせん断変形用等価断面積。
C     -   ASZ    : 各部材のＺ軸回りのせん断変形用等価断面積。
C     -   JCON   : 各部材の部材端拘束表。
C     -   RL     : 各部材の部材長。
C     -   ak_linear     : 部材剛性マトリックス。
C     ――――――――――――――――――――――――――――――――――
      ak_linear(1,1)= YOUNG*RA/RL
      ak_linear(1,7)=-ak_linear(1,1)
      ak_linear(7,7)=ak_linear(1,1)
      IF(IP.EQ.0.AND.JP.EQ.0) GOTO 40                ! 両端ピン接合処理へ
      RL2=RL*RL
      RL3=RL*RL*RL
      IF(ASY.NE.0.0.OR.ASZ.NE.0.0) GOTO 20           ! せん断変形を考慮する処理へ
      ZERL2= 6.*YOUNG*RIZ/RL2
      ZERL3= 2.*ZERL2/RL
      ak_linear(2,2) = ZERL3
      ak_linear(2,6) = ZERL2
      ak_linear(2,8) =-ZERL3
      ak_linear(2,12)= ZERL2
      YERL2= 6.*YOUNG*RIY/RL2
      YERL3= 2.*YERL2/RL
      ak_linear(3,3) = YERL3
      ak_linear(3,5) =-YERL2
      ak_linear(3,9) =-YERL3
      ak_linear(3,11)=-YERL2
      ak_linear(4,4) = SENG*RIX/RL
      ak_linear(4,10)=-ak_linear(4,4)
      ak_linear(5,5) = 4.0*YOUNG*RIY/RL
      ak_linear(5,9) = YERL2
      ak_linear(5,11)= ak_linear(5,5)*0.5
      ak_linear(6,6) = 4.0*YOUNG*RIZ/RL
      ak_linear(6,8) =-ZERL2
```

```
      ak_linear(6,12)= ak_linear(6,6)*0.5
      ak_linear(8,8)=  ZERL3
      ak_linear(8,12)=-ZERL2
      ak_linear(9,9)=  YERL3
      ak_linear(9,11)= YERL2
      ak_linear(10,10)= ak_linear(4,4)
      ak_linear(11,11)= ak_linear(5,5)
      ak_linear(12,12)= ak_linear(6,6)
      DO  I=1,11
      DO  J=i+1,12
      ak_linear(J,I) =ak_linear(I,J)
      enddo
      enddo
      GOTO 30
C    ――――――――――――――――――――――――――――――――――――
C     ---  せん断変形を考慮  ---------------------------------------
C     ――――――――――――――――――――――――――――――――――――
   20 CONTINUE
      FY=0.0
      FZ=0.0
      IF(ASY.NE.0.0) FY=12.*YOUNG*RIZ/(SENG*ASY*RL2)
      IF(ASZ.NE.0.0) FZ=12.*YOUNG*RIY/(SENG*ASZ*RL2)
      ZERL3=12.*YOUNG*RIZ/(RL3*(1.+FY))
      ZERL2= 6.*YOUNG*RIZ/(RL2*(1.+FY))
      ak_linear(2,2) = ZERL3
      ak_linear(2,6) = ZERL2
      ak_linear(2,8) =-ZERL3
      ak_linear(2,12)= ZERL2
      YERL3=12.*YOUNG*RIY/(RL3*(1.+FZ))
      YERL2= 6.*YOUNG*RIY/(RL2*(1.+FZ))
      ak_linear(3,3) = YERL3
      ak_linear(3,5) =-YERL2
      ak_linear(3,9) =-YERL3
      ak_linear(3,11)=-YERL2
      ak_linear(4,4) = SENG*RIX/RL
      ak_linear(4,10)=-ak_linear(4,4)
      ak_linear(5,5) = (4.0+FZ)*YOUNG*RIY/(RL*(1.+FZ))
      ak_linear(5,9) = YERL2
      ak_linear(5,11)= (2.0-FZ)*YOUNG*RIY/(RL*(1.+FZ))
      ak_linear(6,6) = (4.0+FY)*YOUNG*RIZ/(RL*(1.+FY))
      ak_linear(6,8) =-ZERL2
      ak_linear(6,12)= (2.0-FY)*YOUNG*RIZ/(RL*(1.+FY))
      ak_linear(8,8)=  ZERL3
      ak_linear(8,12)=-ZERL2
      ak_linear(9,9)=  YERL3
      ak_linear(9,11)= YERL2
      ak_linear(10,10)= ak_linear(4,4)
      ak_linear(11,11)= ak_linear(5,5)
      ak_linear(12,12)= ak_linear(6,6)
      DO  I=1,11
      DO  J=i+1,12
      ak_linear(J,I) =ak_linear(I,J)
      enddo
      enddo
```

```
C     ―――――――――――――――――――――――――――――――――――――――
C     ---  ピン結合処理  ----------------------------------------
C     ―――――――――――――――――――――――――――――――――――――――
   30 CONTINUE
      IF(IP.EQ.1.AND.JP.EQ.1) GOTO 31 ! 両端剛接合の場合処理終了
      KP=1
      IF(JP.EQ.0) KP=2
      CALL SPPIN(KP,ak_linear)        ! 部材の片方がピン接合の場合の処理
   31 CONTINUE
      RETURN
C     ―――――――――――――――――――――――――――――――――――――――
C     ---  トラス部材の剛性  -------------------------------------
C     ―――――――――――――――――――――――――――――――――――――――
   40 CONTINUE
      ak_linear(7,7) = ak_linear(1,1)
      ak_linear(7,1) =-ak_linear(1,1)
      RETURN
      END
C     ―――――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /SPPIN
C       ―――――――――――――――――――――――――――――――――――――
C     ●       一端ピン部材剛性行列変換
C     ―――――――――――――――――――――――――――――――――――――――
      SUBROUTINE SPPIN(KP,ak_linear)
      IMPLICIT  REAL*8(A-H,O-Z)
      DIMENSION ak_linear(12,12),MM(2,2),S(12,2),A(2)
      DATA MM/5,6,11,12/
C     ―――――――――――――――――――――――――――――――――――――――
C     -   ak_linear     : 部材剛性マトリックス。
C     ―――――――――――――――――――――――――――――――――――――――
      ak_linear(4,4)  =0.0
      ak_linear(4,10) =0.0
      ak_linear(10,4) =0.0
      ak_linear(10,10)=0.0
      DO 10 MX=1,2
      M=MM(MX,KP)
      A(MX)=0.0
      IF(ak_linear(M,M).EQ.0.0) GOTO 10
      A(MX)=1.0/ak_linear(M,M)
      DO 11 I=1,12
      S(I,MX)=ak_linear(I,M)
   11 CONTINUE
   10 CONTINUE
      DO 20 MX=1,2
      M=MM(MX,KP)
      DO 20 I=1,12
      ak_linear(I,M)=0.0
      ak_linear(M,I)=0.0
   20 CONTINUE
      DO 30 MX=1,2
      M=MM(MX,KP)
      IF(A(MX).EQ.0.0) GOTO 30
      AA=A(MX)
      S(M,MX)=0.0
```

```
      DO 31 I=1,12
      DO 31 J=1,12
      ak_linear(I,J)=ak_linear(I,J)-AA*S(I,MX)*S(J,MX)
   31 CONTINUE
   30 CONTINUE
      return
      end
C     ―――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Deal_Rigid_element
C     ―――――――――――――――――――――――――――――――――――――
C     ●   剛域のための剛性行列座標変換(ok)
C     ―――――――――――――――――――――――――――――――――――――
C     ―――――――――――――――――――――――――――――――――――――
C    部材に剛域を有する場合は、このルーチンを使用する。
C    剛域を含む部材の座標系は、
C
C        ================----------------------
C
C        |----- Lg ------|
C        |---------------------- L ---------------|
C
C    部材端部の変位(U,V,φ)及び節点力(Px,Py,Mz)と
C    剛域内部点(u,v,θ)の変位及び節点力(px,py,mz)の関係は、
C    幾何学的条件と力の釣合より
C
C    U = u           Px = px
C    V = v - Lgθ     Py = py                              (1)
C    φ= θ            Mz = mz + Lg Py
C
C   あるいは、
C
C    u = U           px = Px
C    v = V + Lgφ     py = Py                              (2)
C    θ= φ            mz = Mz - LgPy
C
C    ここで、Lg は剛域長さである。上式を行列で表すと、
C
C         | 1 0  0 |          | 1  0  0 |
C    Ru = | 0 1 Lg |     Rf = | 0  1  0 |                 (3)
C         | 0 0  1 |          | 0 -Lg 1 |
C
C    u = Ru U      p = Rf P                               (4)
C
C    u = (u,v,θ)      U = (U,V,φ)
C    p = (px,py,mz)   P = (Px,Py,Mz)                      (5)
C
C   で表される。次に、剛域内部端の変位と節点力を用いた力の釣合は、
C
C    p = K u                                              (6)
C
C  で表されるものとする。式(4)を用いて、式(6)を変換する。
C
C    RfP = K Ru U                                         (7)
C
```

```
C  さらに、行列 Rf の逆行列を両辺の左より掛けると、上式は、以下のようになる。
C
C    P = Rfb K Ru U                                          (8)
C
C  行列 Rf の逆行列 Rfb は、
C
C          | 1 0  0 |      T
C    Rfb = | 0 1  0 |  = Ru                                  (9)
C          | 0 Lg 1 |
C
C  であるため、部材両端で表す剛性行列は、以下のように表される。
C
C    ー      T
C    K  =  Ru K Ru
C
C     ――――――――――――――――――――――――――――――――――――
      subroutine Deal_Rigid_element(ak,rigid_i,rigid_j)
      implicit real*8(A-H,O-Z)
      dimension ak(12,12),akk(12,12),r1(6,6),r2(6,6)
      data r1/1.,0.,0.,0.,0.,0.,
     *        0.,1.,0.,0.,0.,0.,
     *        0.,0.,1.,0.,0.,0.,
     *        0.,0.,0.,1.,0.,0.,
     *        0.,0.,0.,0.,1.,0.,
     *        0.,0.,0.,0.,0.,1./
      data r2/1.,0.,0.,0.,0.,0.,
     *        0.,1.,0.,0.,0.,0.,
     *        0.,0.,1.,0.,0.,0.,
     *        0.,0.,0.,1.,0.,0.,
     *        0.,0.,0.,0.,1.,0.,
     *        0.,0.,0.,0.,0.,1./
      if(rigid_i.eq.0..and.rigid_j.eq.0.) return
      if(rigid_i.ne.0.) then
      r1(2,6)=rigid_i
      r1(3,5)=-rigid_i
      else
      r1(2,6)=0.
      r1(3,5)=0.
      endif
      if(rigid_j.ne.0.) then
      r2(2,6)=-rigid_j
      r2(3,5)=rigid_j
      else
      r2(2,6)=0.
      r2(3,5)=0.
      endif
      do i=1,12
      do j=1,12
      akk(i,j)=ak(i,j)
      enddo
      enddo
      call Rotate_K6(akk,r1,r2,aK)
      return
      end
```

```
C     ―――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Deal_Rigid_element_v
C     ―――――――――――――――――――――――――――――――――――
C     ●   剛域のための変位座標変換(部材端から剛域内部点)
C     ―――――――――――――――――――――――――――――――――――
C    u = U
C    v = V + Lgφz
C    w = W - Lgφy
C     ―――――――――――――――――――――――――――――――――――
      subroutine Deal_Rigid_element_v(vv,rigid_i,rigid_j)
      implicit real*8(A-H,O-Z)
      dimension vv(12)
      if(rigid_i.eq.0..and.rigid_j.eq.0.) return
      if(rigid_i.ne.0.) then
      vv(2)=vv(2)+rigid_i*vv(6)
      vv(3)=vv(3)-rigid_i*vv(5)
      endif
      if(rigid_j.ne.0.) then
      vv(8)=vv(8)-rigid_j*vv(12)
      vv(9)=vv(9)+rigid_j*vv(11)
      endif
      return
      end
C     ―――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Deal_Rigid_element_f
C     ―――――――――――――――――――――――――――――――――――
C     ●   剛域のための応力座標変換(剛域内部点から部材端)
C     ―――――――――――――――――――――――――――――――――――
C    Px = px
C    Py = py
C    My = my - Lg Pz
C    Mz = mz + Lg Py
C     ―――――――――――――――――――――――――――――――――――
      subroutine Deal_Rigid_element_f(ff,rigid_i,rigid_j)
      implicit real*8(A-H,O-Z)
      dimension ff(12)
      if(rigid_i.eq.0..and.rigid_j.eq.0.) return
      if(rigid_i.ne.0.) then
      ff(5)=ff(5)-rigid_i*ff(3)
      ff(6)=ff(6)+rigid_i*ff(2)
      endif
      if(rigid_j.ne.0.) then
      ff(11)=ff(11)+rigid_j*ff(9)
      ff(12)=ff(12)-rigid_j*ff(8)
      endif
      return
      end
```

最初のサブルーチン Cal_lin_stiff_M1()は、式(5.5)で表される線形の剛性行列 ak_linear を作成する。まず、剛性行列をゼロクリアする。部材長さ rl は、部材の実際の長さから両端の剛域長さを引いて設定する。後は、式(5.9)にしたがって剛性行列を計算する。このサブルーチ

ンは、両端ピン接合、どちらか片方ピン接合、部材のせん断変形を考慮することが可能であるが、現在では使用不可となっている。

次からのサブルーチンは、部材に剛域を有する場合に関連するもので、サブルーチン Deal_Rigid_element()は、剛性行列 ak を座標変換する。まず、両端に剛域がない場合は、そのままこのサブルーチンより戻る。次に、i 端と j 端の剛域長さを用いて座標変換行列を作成する。この座標変換行列を用いて、サブルーチン Rotate_K6()より、変換操作を行う。

サブルーチン Deal_Rigid_element_v()と Deal_Rigid_element_f()は、変位や応力ベクトルを、各々、部材端部から剛域内部点へ、逆に、剛域内部点から部材端部へと変換するプログラムである。変換式は理論マニュアルあるいはプログラム内のコメントを参照されたい。

5.7 幾何学的非線形モデル

5.7.1 幾何学的非線形剛性行列

本節では、部材モデルの幾何学的非線形剛性について解説する。幾何学的非線形性を考慮した接線剛性$[K_T]$は、次式で評価される。

$$[K_T]=[K_L]+[K_G]+[K_N] \qquad \cdots\cdots(5.10)$$

ここで、線形剛性行列$[K_L]$は前節で示した。次に幾何剛性行列$[K_G]$は、次式で示される。ここで、$\bar{N}$ は部材の軸力を示す。

$$[K_G]=\bar{N}\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & \frac{6}{5\ell} & 0 & 0 & 0 & \frac{1}{10} & 0 & -\frac{6}{5\ell} & 0 & 0 & 0 & \frac{1}{10} \\
 & & \frac{6}{5\ell} & 0 & -\frac{1}{10} & 0 & 0 & 0 & -\frac{6}{5\ell} & 0 & -\frac{1}{10} & 0 \\
 & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & \frac{2\ell}{15} & 0 & 0 & 0 & \frac{1}{10} & 0 & -\frac{\ell}{30} & 0 \\
 & & & & & \frac{2\ell}{15} & 0 & -\frac{1}{10} & 0 & 0 & 0 & -\frac{\ell}{30} \\
 & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & \frac{6}{5\ell} & 0 & 0 & 0 & -\frac{1}{10} \\
 & & & sym & & & & & \frac{6}{5\ell} & 0 & \frac{1}{10} & 0 \\
 & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & \frac{2\ell}{15} & 0 \\
 & & & & & & & & & & & \frac{2\ell}{15}
\end{bmatrix} \qquad \cdots\cdots(5.11)$$

また、大変位剛性行列$[K_N]$は、

$$[K_N]=\frac{EA}{\ell}\begin{bmatrix}
0 & -\frac{\bar{v}_1-\bar{v}_2}{\ell} & -\frac{\bar{w}_1-\bar{w}_2}{\ell} & 0 & 0 & 0 & 0 & \frac{\bar{v}_1-\bar{v}_2}{\ell} & \frac{\bar{w}_1-\bar{w}_2}{\ell} & 0 & 0 & 0 \\
 & \left(\frac{\bar{v}_1-\bar{v}_2}{\ell}\right)^2 & \frac{\bar{v}_1-\bar{v}_2}{\ell}\cdot\frac{\bar{w}_1-\bar{w}_2}{\ell} & 0 & 0 & 0 & \frac{\bar{v}_1-\bar{v}_2}{\ell} & -\left(\frac{\bar{v}_1-\bar{v}_2}{\ell}\right)^2 & -\frac{\bar{v}_1-\bar{v}_2}{\ell}\cdot\frac{\bar{w}_1-\bar{w}_2}{\ell} & 0 & 0 & 0 \\
 & & \left(\frac{\bar{w}_1-\bar{w}_2}{\ell}\right)^2 & 0 & 0 & 0 & \frac{\bar{w}_1-\bar{w}_2}{\ell} & -\frac{\bar{v}_1-\bar{v}_2}{\ell}\cdot\frac{\bar{w}_1-\bar{w}_2}{\ell} & -\left(\frac{\bar{w}_1-\bar{w}_2}{\ell}\right)^2 & 0 & 0 & 0 \\
 & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & 0 & -\frac{\bar{v}_1-\bar{v}_2}{\ell} & -\frac{\bar{w}_1-\bar{w}_2}{\ell} & 0 & 0 & 0 \\
 & & & & & & & \left(\frac{\bar{v}_1-\bar{v}_2}{\ell}\right)^2 & \frac{\bar{v}_1-\bar{v}_2}{\ell}\cdot\frac{\bar{w}_1-\bar{w}_2}{\ell} & 0 & 0 & 0 \\
 & & sym & & & & & & \left(\frac{\bar{w}_1-\bar{w}_2}{\ell}\right)^2 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & 0 & 0 \\
 & & & & & & & & & & & 0
\end{bmatrix} \quad \cdots\cdots(5.12)$$

で表される。

幾何剛性行列$[K_G]$と大変位剛性行列$[K_N]$を作成するサブルーチンをコールするコードは、例えば、Cal_nonlin_sitff_M1()で以下のように行われる。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_nonlin_stiff_M1
C       ―――――――――――――――――――――――――――――――――
C     ●   弾性立体フレーム部材の接線剛性行列の計算(ok)
C     ―――――――――――――――――――――――――――――――――
      subroutine Cal_nonlin_stiff_M1(Member ,Element ,
     *     an,ak )
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s    / Member
      record / element_s   / Element
      dimension ak(12,12)
       ・
       ・
C     ―――――――――――――――――――――――――――――――――
      call Cal_lin_stiff_M1(Member,Element,ak)  ! 線形剛性を計算する
      call Cal_geomet_stiff(an,Member,ak)       ! 幾何学的非線形剛性を作成
      EA=Element.A*Element.E
```

```
      call Create_Kn(ak,Member.an_vv(1),Member.an_ww(1),
     *               EA,Member.alength )        ! 大変位剛性剛列の作成
       .
       .
      return
      end
```

ここでは、幾何剛性行列を作成するサブルーチン Cal_geomet_stiff() について解説する。このサブルーチンは、軸力 an と部材長さ al より、幾何剛性行列$[K_G]$を作成する。このサブルーチンの内容を以下の示す。

```
C     ―――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_geomet_stiff
C       ―――――――――――――――――――――――――――――――
C     ●   幾何剛性の計算(ok)
C     ―――――――――――――――――――――――――――――――
      subroutine Cal_geomet_stiff(an,Member,ak)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s    / Member
      dimension ak(12,12)
c―――――――――――――――――――――――――――――――――――――――――――
c           an            real*8   部材軸力
c           Element       structure
c           Member        structure
c           ak            real*8   線形剛性行列
c―――――――――――――――――――――――――――――――――――――――――――
      al=Member.alength -
     *  Member.i_rigid_length - Member.j_rigid_length
      s1=an*6./(5.*al)
      s2=an/10.
      s3=an*al*2./15.
      s4=an*al/30.
      ak(2,2)=ak(2,2)  +s1
      ak(2,6)=ak(2,6)  +s2
      ak(2,8)=ak(2,8)  -s1
      ak(2,12)=ak(2,12)+s2
      ak(6,2)=ak(2,6)
      ak(8,2)=ak(2,8)
      ak(12,2)=ak(2,12)

      ak(3,3)=ak(3,3)  +s1
      ak(3,5)=ak(3,5)  -s2
      ak(3,9)=ak(3,9)  -s1
      ak(3,11)=ak(3,11)-s2
      ak(5,3)=ak(3,5)
      ak(9,3)=ak(3,9)
      ak(11,3)=ak(3,11)

      ak(5,5)=ak(5,5)  +s3
      ak(5,9)=ak(5,9)  +s2
      ak(5,11)=ak(5,11)-s4
```

```
      ak(9,5)=ak(5,9)
      ak(11,5)=ak(5,11)

      ak(6,6)=ak(6,6)  +s3
      ak(6,8)=ak(6,8)  -s2
      ak(6,12)=ak(6,12)-s4
      ak(8,6)=ak(6,8)
      ak(12,6)=ak(6,12)

      ak(8,8)=ak(8,8)   +s1
      ak(8,12)=ak(8,12) -s2
      ak(12,8)=ak(8,12)

      ak(9,9)=ak(9,9)   +s1
      ak(9,11)=ak(9,11) +s2
      ak(11,9)=ak(9,11)

      ak(11,11)=ak(11,11)+s3
      ak(12,12)=ak(12,12)+s3

      return
      end
```

5.7.2 大変位剛性行列

次に、大変位剛性行列を作成するサブルーチン Create_Kn()について解説する。このサブルーチンは、部材の面外変位 vv、ww と部材長さ rl より、式(5.8)にしたがって大変位剛性行列$[K_N]$を作成する。このサブルーチンの内容を以下に示す。

```
C     ------------------------------------------------
C     ●   SUBROUTINE /Create_Kn
C     ------------------------------------------------
C     ●   非線形剛性の計算(ok)
C     ------------------------------------------------
      subroutine Create_Kn(akn,vv,ww,ea,rl)
      implicit real*8(A-H,O-Z)
c-----------------------------------------------------------
      dimension akn(12,12),ak(3,3)
      do i=1,3
      do j=1,3
      ak(i,j)=0.
      enddo
      enddo
      rl2=rl*rl
      rl3=rl2*rl
      ak(1,2)=ea*vv/rl2
      ak(1,3)=ea*ww/rl2
      ak(2,2)=ea*vv*vv/rl3
      ak(2,3)=ea*vv*ww/rl3
      ak(3,3)=ea*ww*ww/rl3
      ak(2,1)=ak(1,2)
```

```
      ak(3,1)=ak(1,3)
      ak(3,2)=ak(2,3)
      do i=1,3
      do j=1,3
      akk=ak(i,j)
      akn(i,j)    =akn(i,j)+akk
      akn(i+6,j)  =akn(i+6,j)-akk
      akn(i,j+6)  =akn(i,j+6)-akk
      akn(i+6,j+6)=akn(i+6,j+6)+akk
      enddo
      enddo
      return
      end
```

5.8 せん断型モデル

本節では、せん断型弾塑性モデルの剛性行列を作成するサブプログラムについて解説する。ただし、この部材モデルに含まれている各種の履歴モデルについては次章で説明する。ここでは、３つのサブルーチンについて説明する。まず、このサブルーチンをコールするコードを以下に示す。

```
      call Cal_lin_stiff_M2(Member(i),Element(ie),ak_linear(1,1,i))
      call Cal_check_stiff_M2(Member(i),Element(ie),RO_work,vv,vpp )
      call Cal_nonlin_stiff_M2(Member(i),Element(ie),ak )
```

この３つのサブルーチンは、上から線形剛性を求め、次に部材の履歴特性をチェックし、せん断ばねの接線剛性をセットする。最後に、せん断ばねの接線剛性を用いて接線剛性行列を作成する。これらのサブルーチンの内容を具体的に以下に示す。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_lin_stiff_M2
C       ―――――――――――――――――――――――――――――――――
C     ●   Model_No.2 3 次元せん断弾塑性モデル
C     ―――――――――――――――――――――――――――――――――
      subroutine Cal_lin_stiff_M2(Member,Element,ak_linear)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s2    / Member
      record / element_s2   / Element
      dimension ak_linear(12,12)
      data  BI_MODEL_NUMBER /11/
      data  RO_MODEL_NUMBER /12/
C     ―――――――――――――――――――――――――――――――――
C     ●   3 次元せん断弾塑性モデル（モデル No.2）
C     ―――――――――――――――――――――――――――――――――
c    要素数（モデル No.2  3 次元せん断弾塑性モデル：トリリニア型）
c   element_s 構造体と同一
```

```
c
c       structure / element_s2/
c       integer  element_type   !  要素タイプ(6)
c       integer  n_element      !  非線形要素番号
c       real*8   AK_1           !  第一剛性
c       real*8   AK_2           !  第二剛性
c       real*8   AK_3           !  第三剛性
c       real*8   Q_1            !  第一折れ点のせん断力
c       real*8   Q_2            !  第二折れ点のせん断力
c       real*8   AKu            !  軸方向バネ
c       real*8   dm1            !  ダミー
c       real*8   dm2            !  ダミー
c       real*8   dm3            !  ダミー
c       real*8   dm4            !  ダミー
c       integer  nm_damp        !  部材減衰の有無(1)
c       integer  nm_type        !  履歴モデルのタイプ
c       real*8   ANP            !  軸方向耐力
c       real*8   AQPv           !  y 方向耐力
c       real*8   AQPw           !  z 方向耐力
c       real*8   dmm(3)         !  ダミー
c       end structure
c      record /element_s2/ Element
c     ALLOCATABLE ::Element(:)
c     ALLOCATE (Element(n_element))
c
C     ―――――――――――――――――――――――――――――――――――
C     ●　3 次元せん断弾塑性モデル用（モデル No.2）member_s 構造体
C     ―――――――――――――――――――――――――――――――――――
c     ALLOCATABLE :: Member (:)
c     ALLOCATE (Member (n_member))
c     Element              structure
c     Member               structure
c     ak_linear            real*8   線形剛性行列
c―――――――――――――――――――――――――――――――――――――――――――――――――
      do i=1,12                                                        ! 1
      do j=1,12
      ak_linear(j,i) = 0.0
      end do
      end do
      ak=Element.Aku                                                   ! 2
      ak_linear(1,1)= ak
      ak_linear(1,7)=-ak
      ak_linear(7,7)= ak
      ak_linear(7,1)=-ak
      if (Element.nm_type.eq.BI_MODEL_NUMBER) then                     ! 3
      ak=Element.AK_1*Member.alength
      else if(Element.nm_type.eq.RO_MODEL_NUMBER) then
      ak=Element.AK_1/Member.alength
      else
      ak=Element.AK_1
      endif
      ak_linear(2,2)= ak                                               ! 4
      ak_linear(2,8)=-ak
      ak_linear(8,8)= ak
```

```
      ak_linear(8,2)=-ak

      ak_linear(3,3)= ak                                                    ! 5
      ak_linear(3,9)=-ak
      ak_linear(9,9)= ak
      ak_linear(9,3)=-ak
c―――――――――――――――――――――――――――――――――★履歴特性の初期設定
      Member.AKv_tan=ak              ! Member.AKv_tan=Element.AK_1          ! 6
      Member.AKw_tan=ak              ! Member.AKw_tan=Element.AK_1
      Member.istat_v=0                                                      ! 7
      Member.istat_w=0
      return
      end
C     ―――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_check_stiff_M2
C       ―――――――――――――――――――――――――――――――――――
C     ●   Model_No.2 3 次元せん断弾塑性モデル
C     ―――――――――――――――――――――――――――――――――――――
      subroutine Cal_check_stiff_M2(Member,Element,RO_work,vv,vpp)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      include "submainx.h"
      record / member_s2    / Member
      record / element_s2   / Element
      record / RO_work_s    /  RO_work
      dimension vv(12),vpp(12),RO_work(*)
C     ―――――――――――――――――――――――――――――――――――――
C     ●   3 次元せん断弾塑性モデル用（モデル No.2）member_s 構造体
C     ―――――――――――――――――――――――――――――――――――――
c
c     部材
c      structure / member_s2/
c      integer  nm_element         !  要素番号
c      integer  element_type       !  要素タイプ
c      integer  n_element_type     !  要素タイプ別番号
c      integer  nm_so              !  部材の層番号
c      integer  nm_dll_element     !  DLL を用いた要素か（０；システム内要素、１：DLL 要素）
c      integer  nm_point(2)        !  節点番号
c      integer  irest(12)          !  部材両端の自由度番号表
c      integer  istat_v            !  y 方向:履歴特性の状態(*)
c      integer  istat_w            !  z 方向:履歴特性の状態(*)
c      integer  nm_analysis        !  部材解析種別
c      integer  nm_group           !  部材グループ
c      integer  nm_local_coord(2)  !  局所座標系の有無とその回転行列の番号
c      integer  nm_damp            !  部材減衰の有無とその減衰行列の番号
c      real*8   alength            !  長さ
c      real*8   rot_x              !  部材主軸の回転角度（度）
c      real*8   force(12)          !  部材両端の部材端力（釣合座標系）
c      real*8   stress(6)          !  部材中央の応力（部材座標系）
c      real*8   AKv_tan            !  接線剛性(*)
c      real*8   AKw_tan            !  接線剛性(*)
c      real*8   dmv(5)             !  y 方向耐力:履歴特性で使用(*)
c      real*8   dmw(5)             !  z 方向耐力:履歴特性で使用(*)
c      end structure
```

```
c
c     ALLOCATABLE :: Member (:)
c     ALLOCATE (Member (n_member))
c     Element             structure
c     Member              structure
c     ak_linear           real*8   線形剛性行列
c――――――――――――――――――――――――――――――――――――――――
      No_rireki=Element.nm_type                                          ! 8
      if(No_rireki/10.eq.0) then
      goto(5,10,20,30,40,50,60),No_rireki+1
    5 continue
c――――――――――――――――――――――――――――――――――――――――★規定モデル：武田モデル
      call Takeda_TriLiner(Member,Element,vv,vpp)
      goto 999
   10 continue
c――――――――――――――――――――――――――――――――――――――――★トリリニア：Nomal
      call Mesing_TriLiner(Member,Element,vv,vpp)
      goto 999
   20 continue
c――――――――――――――――――――――――――――――――――――――――★トリリニア：最大点指向型
      call DirecMax_TriLiner(Member,Element,vv,vpp)
      goto 999
   30 continue
c――――――――――――――――――――――――――――――――――――――――★トリリニア：武田モデル
      call Takeda_TriLiner(Member,Element,vv,vpp)
      goto 999
   40 continue
c――――――――――――――――――――――――――――――――――――――――★バイリニア：Nomal
      if(Member.istat_v.eq.0.and.Member.istat_w.eq.0) then
      Element.AK_2=Element.AK_1
      Element.Q_2 =Element.Q_1
      endif
      call Mesing_TriLiner(Member,Element,vv,vpp)
      goto 999
   50 continue
c――――――――――――――――――――――――――――――――――――――――★
      goto 999
   60 continue
c――――――――――――――――――――――――――――――――――――――――★履歴モデル
      goto 999
      elseif(No_rireki/10.eq.1) then
      goto(101,102),No_rireki - 10
  101 continue
c――――――――――――――――――――――――――――――――――――――――★修正バイリニアモデル（火力センター）
      mro=Element.n_section(1)
      write(222,'(A20,I5)') 'Modify Bi-Liner', mro
      call Modify_Bi_Liner1(Member,Element,RO_work(mro),vv,vpp)
      goto 999

  102 continue
c――――――――――――――――――――――――――――――――――――――――★修正 RO モデル（火力センター）
      mro=Element.n_section(1)
      write(222,'(A20,I5)') 'Modify R-O', mro
      call Modify_RO1(Member,Element,RO_work(mro),vv,vpp)
```

```
      goto 999
      endif
  999 continue
      return
      end
C     ――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_nonlin_stiff_M2
C       ――――――――――――――――――――――――――――――――
C     ●   Model_No.2 3 次元せん断弾塑性モデル
C     ――――――――――――――――――――――――――――――――――
      subroutine Cal_nonlin_stiff_M2(Member,Element,ak)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s2    / Member
      record / element_s2   / Element
      dimension ak(12,12)
C     ――――――――――――――――――――――――――――――――――
c     ALLOCATABLE :: Member (:)
c     ALLOCATE (Member (n_member))
c     Element              structure
c     Member               structure
c     ak                  real*8   非線形剛性行列
c――――――――――――――――――――――――――――――――――――――――
      do i=1,12                                                    ! 9
      do j=1,12
      ak(j,i) = 0.0
      end do
      end do
      akk=Element.Aku                                              ! 10
      ak(1,1)= akk
      ak(1,7)=-akk
      ak(7,7)= akk
      ak(7,1)=-akk
      akk=Member.AKv_tan                                           ! 11
      ak(2,2)= akk
      ak(2,8)=-akk
      ak(8,8)= akk
      ak(8,2)=-akk
      akk=Member.AKw_tan                                           ! 12
      ak(3,3)= akk
      ak(3,9)=-akk
      ak(9,9)= akk
      ak(9,3)=-akk
      return
      end
```

プログラムの内容をコードに従って説明する。これらのプログラムは、せん断型弾塑性モデルであり、単純なプログラムである。

1. 線形の剛性行列 ak_linear をゼロクリアする。
2. 軸方向剛性を構造体より取得し、剛性行列にセットする。

3. 要素の履歴特性番号にしたがって、横方向せん断ばね定数を取得する。修正バイリニアモデルの剛性は、ばね定数に部材長さを掛けて求め、また、修正 RO モデルでは、部材長さで割って求める。これについてはリファレンスマニュアルを参照されたい。通常のせん断型ではそのままのばね定数をせん断初期剛性としてセットする。
4. ばね定数を剛性行列のｙ方向位置にセットする。
5. ばね定数を剛性行列のｚ方向位置にセットする。
6. ばね定数を構造体に初期設定する。
7. ばね定数が弾性であるかどうかチェックするためのパラメータに、初期値として弾性を意味する 0 をセットする。
8. 要素の履歴特性番号（履歴モデルのタイプ）を表す構造体よりパラメータを取得し、その履歴特性番号にしたがって、処理を分割する。このサブルーチンについては前章で説明した。そちらを参照されたい。
9. 非線形の剛性行列 ak をゼロクリアする。
10. 軸方向剛性を構造体より取得し、剛性行列にセットする。
11. せん断ばねのｙ方向せん断剛性を取得し、剛性行列にセットする。
12. せん断ばねのｚ方向せん断剛性を取得し、剛性行列にセットする。

5.9 ファイバーモデル

5.9.1 ファイバーモデルの概要

ファイバーモデルとは、断面内を細分化して、その細分化した要素の応力とひずみの関係を一軸状態で表現する簡易なモデルである。ここでは、はりのひずみは平面保持の仮定を用いて、図芯位置での軸方向ひずみと面外の 2 方向曲げひずみによって決定する。詳しくは、理論マニュアルを見られたい。

ファイバーモデルにおける接線剛性は、次式で示すように、ファイバーの弾塑性状態を考慮した弾塑性剛性と、さらに幾何剛性と大変位剛性によって構成される。

$$[K_T]=[K_L]+[K_G]+[K_N] \qquad \cdots\cdots(5.13)$$

最初に、線形増分剛性$[K_L]$は、以下のように求められている。

$$[K_L]=\begin{bmatrix}
\frac{\overline{EA}}{\ell} & 0 & 0 & 0 & \frac{\overline{EAZ}}{\ell} & -\frac{\overline{EAY}}{\ell} & -\frac{\overline{EA}}{\ell} & 0 & 0 & 0 & -\frac{\overline{EAZ}}{\ell} & \frac{\overline{EAY}}{\ell} \\
 & \frac{12\overline{EAY^2}}{\ell^3} & 0 & 0 & 0 & \frac{6\overline{EAY^2}}{\ell^2} & 0 & -\frac{12\overline{EAY^2}}{\ell^3} & 0 & 0 & 0 & \frac{6\overline{EAY^2}}{\ell^2} \\
 & & \frac{12\overline{EAZ^2}}{\ell^3} & 0 & -\frac{6\overline{EAZ^2}}{\ell^2} & 0 & 0 & 0 & -\frac{12\overline{EAZ^2}}{\ell^3} & 0 & -\frac{6\overline{EAZ^2}}{\ell^2} & 0 \\
 & & & \frac{GJ_x}{\ell} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_x}{\ell} & 0 & 0 \\
 & & & & \frac{4\overline{EAZ^2}}{\ell} & -\frac{\overline{EAYZ}}{\ell} & -\frac{\overline{EAZ}}{\ell} & 0 & \frac{6\overline{EAZ^2}}{\ell^2} & 0 & \frac{2\overline{EAZ^2}}{\ell} & \frac{\overline{EAYZ}}{\ell} \\
 & & & & & \frac{4\overline{EAY^2}}{\ell} & \frac{\overline{EAY}}{\ell} & -\frac{6\overline{EAY^2}}{\ell^2} & 0 & 0 & \frac{\overline{EAYZ}}{\ell} & \frac{2\overline{EAY^2}}{\ell} \\
 & & & & & & \frac{\overline{EA}}{\ell} & 0 & 0 & 0 & \frac{\overline{EAZ}}{\ell} & -\frac{\overline{EAY}}{\ell} \\
 & & & & & & & \frac{12\overline{EAY^2}}{\ell^3} & 0 & 0 & 0 & -\frac{6\overline{EAY^2}}{\ell^2} \\
 & & & & & & & & \frac{12\overline{EAZ^2}}{\ell^3} & 0 & \frac{6\overline{EAZ^2}}{\ell^2} & 0 \\
 & & sym & & & & & & & \frac{GJ_x}{\ell} & 0 & 0 \\
 & & & & & & & & & & \frac{4\overline{EAZ^2}}{\ell} & -\frac{\overline{EAYZ}}{\ell} \\
 & & & & & & & & & & & \frac{4\overline{EAY^2}}{\ell}
\end{bmatrix} \qquad \cdots\cdots(5.14)$$

ここで、剛性行列内の各係数は

$$\overline{EA} = \sum E_i A_i$$

$$\left.\begin{aligned}
\overline{EAY_1} &= \int_A EydA\int_0^\ell \frac{1}{\ell}dx = \int_A EydA = \sum E_i Y_i A_i \\
\overline{EAY_2} &= \int_A EydA\int_0^\ell \frac{2x}{\ell^2}dx = \int_A EydA = \sum E_i Y_i A_i
\end{aligned}\right\} = \overline{EAY}$$

$$\left.\begin{aligned}
\overline{EAZ_1} &= \int_A EzdA\int_0^\ell \frac{1}{\ell}dx = \int_A EzdA = \sum E_i Z_i A_i \\
\overline{EAZ_2} &= \int_A EzdA\int_0^\ell \frac{2x}{\ell^2}dx = \int_A EzdA = \sum E_i Z_i A_i
\end{aligned}\right\} = \overline{EAZ}$$

$$\left.\begin{aligned}
\overline{EAY_1^2} &= \int_A Ey^2dA\int_0^\ell \frac{1}{\ell}dx = \int_A Ey^2dA = \sum E_i Y_i^2 A_i \\
\overline{EAY_2^2} &= \int_A Ey^2dA\int_0^\ell \frac{2x}{\ell^2}dx = \int_A Ey^2dA = \sum E_i Y_i^2 A_i \\
\overline{EAY_3^2} &= \int_A Ey^2dA\int_0^\ell \frac{3x^2}{\ell^3}dx = \int_A Ey^2dA = \sum E_i Y_i^2 A_i
\end{aligned}\right\} = \overline{EAY^2}$$

$$\left.\begin{aligned}
\overline{EAZ_1^2} &= \int_A Ez^2dA\int_0^\ell \frac{1}{\ell}dx = \int_A Ez^2dA = \sum E_i Z_i^2 A_i \\
\overline{EAZ_2^2} &= \int_A Ez^2dA\int_0^\ell \frac{2x}{\ell^2}dx = \int_A Ez^2dA = \sum E_i Z_i^2 A_i \\
\overline{EAZ_3^2} &= \int_A Ez^2dA\int_0^\ell \frac{3x^2}{\ell^3}dx = \int_A Ez^2dA = \sum E_i Z_i^2 A_i
\end{aligned}\right\} = \overline{EAZ^2}$$

$$\overline{EAYZ} = \int_A\int_0^\ell \frac{Eyz}{\ell}dxdA = \int_A EyzdA\int_0^\ell \frac{1}{\ell}dx = \int_A EyzdA = \sum E_i Y_i Z_i A_i$$

……(5.15)

となり、断面内の積分は各ファイバーにおける積の和となる。ただし、上の行列の中で、断面極２次モーメントについては弾性部材の剛性を用いることになる。

さらに、幾何剛性と大変位剛性は、弾性部材の幾何剛性行列$[K_G]$と同一となり、また、$[K_N]$は、断面に関する積分を除いて、弾性部材のそれと一致する。したがって、ファイバーモデルの接線剛性は、線形剛性を除いて、幾何剛性と大変位剛性は弾性の剛性を用いれば良いことになる。

5.9.2 各モデルの階層構造

本節では、剛性$[K_L]$を求めるサブルーチンについて解説する。例として用いる部材モデルは、両端ファイバーエレメントを組み込んだ部材モデルとする。剛性$[K_L]$に関連するサブルーチンは、以下のようにコールされる。

```
      call Cal_lin_stiff_M11(Model_type,Member(i),Element(ie),
     *      ak_linear(1,1,i) ,E_model11, E_model_fiber,
     *      M_model11, M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work)
      call Cal_check_stiff_M11(Control,N_analysis,
     *      mem,Model_type,Member(i),Element(ie),
     *      E_model11, E_model_fiber,
     *      M_model11, M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work,vv,vpp,f)
      call Cal_nonlin_stiff_M11(N_analysis,
     *      Model_type,Member(i),Element(ie),
     *      ak ,ier,E_model11, E_model_fiber,
     *      M_model11 , M_model_fiber)
```

上記の３つのサブルーチンの内容を以下の示す。これらは、前章で既に説明したので、内容はほぼ理解できていることと思う。本節では、このサブルーチンの中身を検証することにする。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_lin_stiff_M11
C       ―――――――――――――――――――――――――――――――――
C     ●   代表的な部材モデルの剛性(両端ファイバーモデル)
C     ―――――――――――――――――――――――――――――――――
      subroutine Cal_lin_stiff_M11(Model_type,Member,Element,ak,
     *      E_model11, E_model_fiber,
     *      M_model11, M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work)
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_model11_s / E_model11
      record / E_model_fiber_s    / E_model_fiber
      record / M_model11_s        / M_model11
      record / M_model_fiber_s    / M_model_fiber
      record / Bilinear_work_s    / Bilinear_work
      record / Trilinear_work_s   / Trilinear_work
      record / Concrete_work_s    / Concrete_work
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension E_model11(*),M_model11(*)
      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_model11_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_ww(i)=0.
      enddo
c―――――――――――――――――――――――――――――――――★部材剛性行列の作成
      it = 0
      do i=1,n_div
      call Stiff_M11_I(i,it,n_type(i),akk,Member,alength,
     *       Model_type,Element,
     *       E_model11(imm), E_model_fiber,
     *       M_model11(immm), M_model_fiber,
     *       Bilinear_work,Trilinear_work,Concrete_work)
      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_M11
C       ----------------------------------------
C     ●   代表的な部材モデルの塑性チェック(両端ファイバーモデル)
C     ------------------------------------------
      subroutine Cal_check_stiff_M11(Control,N_analysis,
     *      mem_x,Model_type,Member,Element,
     *      E_model11, E_model_fiber,
     *      M_model11, M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work,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_model11_s        / E_model11
      record / E_model_fiber_s    / E_model_fiber
      record / M_model11_s        / M_model11
      record / M_model_fiber_s    / M_model_fiber
      record / Bilinear_work_s    / Bilinear_work
      record / Trilinear_work_s   / Trilinear_work
      record / Concrete_work_s    / Concrete_work
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension E_model11(*),M_model11(*)
      dimension ak(12,12),f_p(12)
      dimension vv(12),vp(12),vvx(12)
      dimension Bilinear_work(*),Trilinear_work(*)
      dimension Concrete_work(*)
      real*8, ALLOCATABLE :: c(:,:),ab(:,:),alength(:),EA(:)
      real*8, ALLOCATABLE :: bav(:),akk(:,:,:),f1(:)
      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-------------------------------★動的記憶領域の確保
      ALLOCATE (
     *   irest_Point(6,n_div+1),n_type(n_div),alength(n_div),EA(n_div),
```

```
     *    akk(12,12,n_div)
     *         )
c----------★節点拘束表の作成
c                                                  未知数等をセット
      call set_model11_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),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
      EAx=Element.A*Element.E
      it=0
      do i=1,n_div
      EA(i)=EAx
      if(n_type(i).eq.2) then
      it=it+1
      if(it.eq.1) then
      EA(i)=M_model11(immm).d_ra_1
      else
      EA(i)=M_model11(immm).d_ra_2
      endif
      endif
      enddo
c----------★部材剛性行列の作成
      it = 0
      do i=1,n_div
c----------★内部部材の剛性行列の計算
```

```
      call Stiff_M11(i,it,n_type(i),akk(1,1,i),Member,alength,
     *       Model_type,Element,
     *       E_model11(imm), E_model_fiber,
     *       M_model11(immm), M_model_fiber)
      if(N_analysis.ne.9.and.N_analysis.ne.7.and.n_type(i).ne.3) 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_ww(i),
     *              EA(i),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-------------------------★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,M_model11(immm).ff_ip(1),
     *                    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
      call Typical_member_p_force(akk(1,1,i),irest_Point(1,i),
     *                           vv,bav ,f_p,f1)
      enddo
c-------------------------★部材内部、両端節点力から不釣合力へ
      do i=1,n_if
      M_model11(immm).ff_ip(i)=f1(i)
      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
```

```
      vvx(ii)=vv(-irest)
      elseif(irest.gt.0) then
      vvx(ii)=bav(irest)
      else
      vvx(ii)=0.
      endif
      enddo
      enddo
c-----------------------------★ファイバーチェック
      if(n_type(i).eq.2) then
      it=it+1
      call Fiber_check(Control,i,N_analysis,
     *      mem_x,it,alength(i),Member,Element,
     *      E_model11(imm),E_model_fiber,M_model11(immm),M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work,vvx)
      endif
c-------------------------------★弾性部材の軸力計算（幾何剛性作成用）
      call nonlinear_stress_N(akk(1,1,i),vvx,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)+(vvx(8) - vvx(2))
      Member.an_ww(i)=Member.an_ww(i)+(vvx(9) - vvx(3))
      enddo
c---------------------------------★動的記憶領域の解放
      DEALLOCATE (
     *   c ,ab ,irest_point,n_type,alength,EA,
     *   bav,akk,f1    )
      return
      end
C     ------------------------------------------
C     ●   SUBROUTINE /Cal_nonlin_stiff_M11
C       ----------------------------------------
C     ●   代表的な部材モデルの接線剛性(両端ファイバーモデル)
C     ------------------------------------------
      subroutine Cal_nonlin_stiff_M11(N_analysis,
     *            Model_type,Member,Element, ak,ier,
     *     E_model11, E_model_fiber, M_model11, 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_model11_s       / E_model11
      record / E_model_fiber_s   / E_model_fiber
      record / M_model11_s       / M_model11
      record / M_model_fiber_s   / M_model_fiber
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension E_model11(*),M_model11(*)
      dimension ak(12,12),akk(12,12)
      real*8, ALLOCATABLE :: c(:,:),ab(:,:),ba(:),alength(:),EA(:)
      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),EA(n_div)
     *         )
c-----------------------------------★節点拘束表の作成
c                                              未知数等をセット
      call set_model11_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
      EAx=Element.A*Element.E
      it=0
      do i=1,n_div
      EA(i)=EAx
      if(n_type(i).eq.2) then
      it=it+1
      if(it.eq.1) then
      EA(i)=M_model11(immm).d_ra_1
      else
      EA(i)=M_model11(immm).d_ra_2
      endif
      endif
      enddo
c-----------------------------------★部材剛性行列の作成
      it = 0
      do i=1,n_div
```

```
      call Stiff_M11(i,it,n_type(i),akk,Member,alength,
     *       Model_type,Element,
     *       E_model11(imm), E_model_fiber,
     *       M_model11(immm), M_model_fiber)
      if(N_analysis.ne.9.and.N_analysis.ne.7.and.n_type(i).ne.3) 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(i),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,EA
     *         )
      return
      end
```

5.9.3 静的縮合と剛域処理

前節で示した３つのサブルーチンについては、既に第 4.2 節で説明した。全体の計算流れはおおよそ理解できていると思う。ここでは、これら 3 つの中で同じようにコールされるサブルーチンについて解説する。まず、静的縮合を行う汎用の計算ルーチンを以下に示す。

```
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――――――――――――――――――――――――――――――――★部材内部変位の計算(c 行列の分解計算)
      call Typical_member_v(c,ab,M_model11(immm).ff_ip(1),
     *                      n_if,n_if,iubw,iubw,ier,vv,bav)
c――――――――――――――――――――――――――――――――★部材内部、両端節点力の計算
      call Typical_member_p_force(akk(1,1,i),irest_Point(1,i),
     *                            vv,bav ,f_p,f1)
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)
```

これらのサブルーチンは、第 5.3 節で説明した。内容については、その部分を参照されたい。ここで示したサブルーチンは汎用ルーチンになっているため、他のモデルでも使用している。それについては付録を参照されたい。

次は部材の両端もしくはどちらかに剛域がある場合で、剛域処理を行うサブルーチンコールについてである。

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

これらも既に第 5.6 節で説明した。そちらを参照されたい。次に、線形方程式の解析に関連するプログラムを以下にまとめる。

```
      call Bnd_FEM(i,akk,irest_Point,ak,c,ab,iubw,n_if)
      call Decom_B( c, n, nux,nu,eps, ba , ier )
      call Solv_B( c, n, nux, nu, ba  )
```

これらのサブルーチンは、部材行列を分割全体行列に分配するルーチン、その中のバンド行列 c の LU 分解を行うルーチン、最後に方程式を解くルーチンである。これらについては数値計算に関する文献[3)]を参照されたい。

5.9.4 部材モデルの設定

本節では、部材モデルを作成するサブルーチンを以下に示す。このサブルーチン set_model11_dat()は、部材モデルが静的縮合を行うために必要となる情報を作り出す。

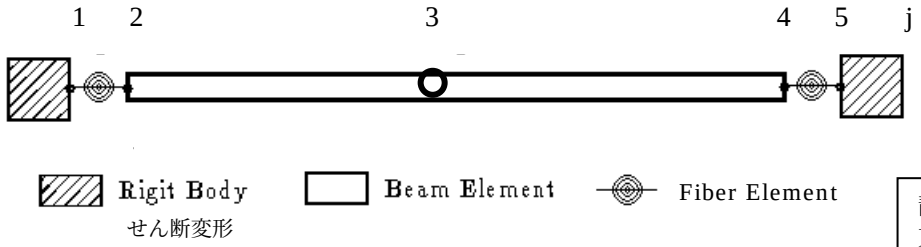

図 5-3 両端ファイバーモデルの部材解析モデル

このモデルは直線であるため、部材内部の釣合式を作成するとき、剛性行列の座標変換を行わない。このサブルーチンでは、節点における拘束表、あるいは未知数表を作成する。節点自由度に関する拘束のデータ仕様は、スペースフレームモデルの節点境界における拘束仕様と多少異なる。部材両端の節点自由度は負の値とし、ゼロは拘束とする。また、内部節点は自由は１で、拘束は 0 となる。

静的縮合部材の節点拘束仕様として、外部節点の自由度番号は負符号を付け、拘束は 0 とする。内部節点では、1 が自由で 0 が拘束となり、10 以上の値は、他の内部節点との変位の同一視を表す。この場合、第一位のけたは、自由度番号で、第二けた以上が節点番号を表す。

図 5.3 は両端ファイバーの部材モデルを表しているが、このモデルでは、両端剛域の有無、さらに両端接合部にせん断変形を許すモデルに変更することができる。この処理を行うのがこのサブルーチンであり、少し長いが以下にその内容を示す。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /set_model11_dat
C       ―――――――――――――――――――――――――――――――――
C     ●   部材モデルの拘束表作成(ok)  ４部材  ５節点
C     ―――――――――――――――――――――――――――――――――
      subroutine set_model11_dat(irest_Point,n_if,n_div,iubw,
     *         Element,Member,
     *         n_type,alength,gxi,gxj,asi,asj)
C     ―――――――――――――――――――――――――――――――――
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / element_s    / Element
      record / member_s     / Member
      dimension  irest(6,5,3),nn_type(4),aleng(4),irest_s(12,3)
      dimension irest_Point(6,*),n_type(*),alength(*)
      real*8  gxi,gxj                 ! 剛域
      real*8  asi,asj                 ! 材端せん断変形用面積
      data nn_type/2,1,1,2/           ! 部材型  2:ファイバー  1:梁柱            ! 1
      data irest/-1,-2,-3,-4,-5,-6,  1,1,1,1,1,1,                               ! 2
     *           1,1,1,1,1,1,
     *           1,1,1,1,1,1,      -7,-8,-9,-10,-11,-12,
     *          -1,0,-3,0,-5,0,    1,0,1,0,1,0,
     *           1,0,1,0,1,0,
     *           1,0,1,0,1,0,      -7,0,-9,-0,-11,0,
     *           -1,-2,0,0,0,-6,   1,1,0,0,0,1,
     *           1,1,0,0,0,1,
     *           1,1,0,0,0,1,      -7,-8,0,0,0,-12/
      data irest_s/0,-2,-3,0,0,0,   0,-8,-9,0,0,0,
     *            0, 0,-3,0,0,0,   0, 0,-9,0,0,0,
     *            0,-2, 0,0,0,0,   0,-8, 0,0,0,0/
      data aleng/0.03,0.47,0.47,0.03/                                           ! 3
c―――――――――――――――――――――――――――――――――――――★モデルチェック
      itype=1           !  両端せん断変形なし
      if(asi.gt.0.) then                                                        ! 4
      if(asj.gt.0.) then
      itype=4         !  両端せん断変形考慮
      else
      itype=2         !  i 端せん断変形考慮
      endif
      elseif(asj.gt.0.) then
      itype = 3       !  j 端せん断変形考慮
      endif
      goto(1001,1002,1003,1004),itype                                           ! 5
c―――――――――――――――――――――――――――――――――――――★両端せん断変形なし
 1001 continue
      n_div=n_div-2                                                             ! 6
      do i=1,n_div                                                              ! 7
```

```
      n_type(i)=nn_type(i)
      enddo
      al=Member.alength - gxi - gxj       ! 剛域を削除                    ! 8
      do i=1,n_div                                                        ! 9
      alength(i)=al*aleng(i)
      enddo
      nn=n_div+1                                                          ! 10
      do i=1,nn                                                           ! 11
      do j=1,6
      irest_Point(j,i)=irest(j,i,1)
      enddo
      enddo
c―――――――――――――――――――――――――★ねじり剛性ゼロの場合
      if(Element.RIx .eq.0.) then                                         ! 12
      do i=2,nn-1
      irest_Point(4,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――★ｙ軸断面二次モーメントゼロの場合
      if(Element.RIy .eq.0.) then                                         ! 13
      do i=2,nn-1
      irest_Point(3,i) = 0
      irest_Point(5,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――★ｚ軸断面二次モーメント
      if(Element.RIz .eq.0.) then                                         ! 14
      do i=2,nn-1
      irest_Point(2,i) = 0
      irest_Point(6,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――★ 内部変位の節点番号セット
      n_if =0                                                             ! 15
      do i=2,nn-1                                                         ! 16
      do j=1,6
      if(irest_Point(j,i).gt.0) then
      if(irest_Point(j,i).eq.1) then
      n_if = n_if +1                                                      ! 17
      irest_Point(j,i)= n_if
      else
      ip=irest_Point(j,i)/10                                              ! 18
      ipp= Mod(irest_Point(j,i),10)
      irest_Point(j,i)=irest_Point(ipp,ip)
      endif
      endif
      enddo
      enddo
      call set_iubw(irest_Point,n_div,iubw)                               ! 19
      return
c―――――――――――――――――――――――――――――――――★i 端せん断変形考慮
 1002 continue                                                            ! 20
      n_div=n_div-1                                                       ! 21
      n_type(1)=3                         ! せん断バネ型                  ! 22
```

```
      do i=2,n_div                                                    ! 23
      n_type(i)=nn_type(i-1)
      enddo
      al=Member.alength - gxi - gxj      ! 剛域を削除
      alength(1)=gxi
      do i=2,n_div
      alength(i)=al*aleng(i-1)
      enddo
      nn=n_div+1
      do j=1,6
      irest_Point(j,1)=irest_s(j,1)
      enddo
      do i=2,nn
      do j=1,6
      irest_Point(j,i)=irest(j,i-1, 1)
      enddo
      enddo
      irest_Point(2,2)=1               ! 内部節点に変更
      if(Member.analysis_3D.eq.1)     irest_Point(2,2)=0  ! 平面問題に対応(x-z)
      irest_Point(3,2)=1               ! 内部節点に変更
      if(Member.analysis_3D.eq.2)     irest_Point(3,2)=0  ! 平面問題に対応(y-z)
c――――――――――――――――――――――――――★ねじり剛性ゼロの場合
      if(Element.RIx .eq.0.) then
      do i=3,nn-1
      irest_Point(4,i) = 0
      enddo
      endif
c――――――――――――――――――――――――――★ｙ軸断面二次モーメントゼロの場合
      if(Element.RIy .eq.0.) then
      do i=3,nn-1
      irest_Point(3,i) = 0
      irest_Point(5,i) = 0
      enddo
      endif
c――――――――――――――――――――――――――★ｚ軸断面二次モーメント
      if(Element.RIz .eq.0.) then
      do i=3,nn-1
      irest_Point(2,i) = 0
      irest_Point(6,i) = 0
      enddo
      endif
c――――――――――――――――――――――――――★内部変位の節点番号セット
      n_if =0
      do i=2,nn-1
      do j=1,6
      if(irest_Point(j,i).gt.0) then
      if(irest_Point(j,i).eq.1) then
      n_if = n_if +1
      irest_Point(j,i)= n_if
      else
      ip=irest_Point(j,i)/10
      ipp= Mod(irest_Point(j,i),10)
      irest_Point(j,i)=irest_Point(ipp,ip)
      endif
```

```
      endif
      enddo
      enddo
      call set_iubw(irest_Point,n_div,iubw)
      return
c―――――――――――――――――――――――――――――――★ j 端せん断変形考慮
 1003 continue                                                          ! 24

      n_div=n_div-1
      do i=1,n_div-1
      n_type(i)=nn_type(i)
      enddo
      n_type(n_div)=3                  ! せん断バネ型                  ! 25
      al=Member.alength - gxi - gxj    ! 剛域を削除
      do i=1,n_div-1
      alength(i)=al*aleng(i)
      enddo
      alength(n_div)=gxj
      nn=n_div+1
      do i=1,nn-1
      do j=1,6
      irest_Point(j,i)=irest(j,i, 1)
      enddo
      enddo
      do j=1,6
      irest_Point(j,nn)=irest_s(j+6,Member.analysis_3D+1)
      irest_Point(j,nn)=irest_s(j+6, 1)
      enddo
      irest_Point(2,nn-1)=1             ! 内部節点に変更
      if(Member.analysis_3D.eq.1)    irest_Point(2,nn-1)=0   ! 平面問題に対応(x-z)
      irest_Point(3,nn-1)=1             ! 内部節点に変更
      if(Member.analysis_3D.eq.2)    irest_Point(3,nn-1)=0   ! 平面問題に対応(y-z)
c―――――――――――――――――――――――――――――――★ねじり剛性ゼロの場合
      if(Element.RIx .eq.0.) then
      do i=2,nn-2
      irest_Point(4,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――――――――★ｙ軸断面二次モーメントゼロの場合
      if(Element.RIy .eq.0.) then
      do i=2,nn-2
      irest_Point(3,i) = 0
      irest_Point(5,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――――――――★ｚ軸断面二次モーメント
      if(Element.RIz .eq.0.) then
      do i=2,nn-2
      irest_Point(2,i) = 0
      irest_Point(6,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――――――――★内部変位の節点番号セット
      n_if =0
```

```
      do i=2,nn-1
      do j=1,6
      if(irest_Point(j,i).gt.0) then
      if(irest_Point(j,i).eq.1) then
      n_if = n_if +1
      irest_Point(j,i)= n_if
      else
      ip=irest_Point(j,i)/10
      ipp= Mod(irest_Point(j,i),10)
      irest_Point(j,i)=irest_Point(ipp,ip)
      endif
      endif
      enddo
      enddo
      call set_iubw(irest_Point,n_div,iubw)
      return
c―――――――――――――――――――――――――――――――★両端せん断変形考慮
 1004 continue                                                          ! 26
      do i=2,n_div-1
      n_type(i)=nn_type(i-1)
      enddo
      n_type(1)    =3                  ! せん断バネ型                  ! 27
      n_type(n_div)=3                  ! せん断バネ型
      al=Member.alength - gxi - gxj    ! 剛域を削除
      do i=2,n_div-1
      alength(i)=al*aleng(i-1)
      enddo
      alength(1)    =gxi
      alength(n_div)=gxj
      nn=n_div+1
      do i=2,nn-1
      do j=1,6
      irest_Point(j,i)=irest(j,i-1, 1)
      enddo
      enddo
      do j=1,6
      irest_Point(j,1) =irest_s(j, 1)
      irest_Point(j,nn)=irest_s(j+6, 1)
      enddo
      irest_Point(2,2)   =1            ! 内部節点に変更
      if(Member.analysis_3D.eq.1)    irest_Point(2,2)=0  ! 平面問題に対応(x-z)
      irest_Point(3,2)   =1            ! 内部節点に変更
      if(Member.analysis_3D.eq.2)    irest_Point(3,2)=0  ! 平面問題に対応(y-z)
      irest_Point(2,nn-1)=1            ! 内部節点に変更
      if(Member.analysis_3D.eq.1)    irest_Point(2,nn-1)=0   ! 平面問題に対応(x-z)
      irest_Point(3,nn-1)=1            ! 内部節点に変更
      if(Member.analysis_3D.eq.2)    irest_Point(3,nn-1)=0   ! 平面問題に対応(y-z)
c―――――――――――――――――――――――――――――――★ねじり剛性ゼロの場合
      if(Element.RIx .eq.0.) then
      do i=2,nn-1
      irest_Point(4,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――――――――★ｙ軸断面二次モーメントゼロの場合
```

```
      if(Element.RIy .eq.0.) then
      do i=3,nn-2
      irest_Point(3,i) = 0
      irest_Point(5,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――★ｚ軸断面二次モーメント
      if(Element.RIz .eq.0.) then
      do i=3,nn-2
      irest_Point(2,i) = 0
      irest_Point(6,i) = 0
      enddo
      endif
c―――――――――――――――――――――――――★内部変位の節点番号セット
      n_if =0
      do i=2,nn-1
      do j=1,6
      if(irest_Point(j,i).gt.0) then
      if(irest_Point(j,i).eq.1) then
      n_if = n_if +1
      irest_Point(j,i)= n_if
      else
      ip=irest_Point(j,i)/10
      ipp= Mod(irest_Point(j,i),10)
      irest_Point(j,i)=irest_Point(ipp,ip)
      endif
      endif
      enddo
      enddo
      call set_iubw(irest_Point,n_div,iubw)
      return
      end
C     ―――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /set_iubw
C       ―――――――――――――――――――――――――――――――――
C     ●   半バンド幅の計算
C     ―――――――――――――――――――――――――――――――――――
      subroutine set_iubw(irest,n_div,iubw)
      implicit real*8(A-H,O-Z)
      dimension irest(6,*)
      iubw=0                                                    ! 28
      do 100 i=1,n_div                                          ! 29
      do j=1,6                                                  ! 30
      if(irest(j,i).gt.0) then                                  ! 31
      do k=1,6
      if(irest(k,i+1).gt.0) then
      iu=iabs(irest(j,i)-irest(k,i+1))                          ! 32
      if(iubw.lt.iu) iubw=iu
      endif
      enddo
      endif
      enddo
  100 continue
      do 101 i=1,n_div+1                                        ! 33
```

```
      do j=1,5
      if(irest(j,i).gt.0) then
      do k=j+1,6
      if(irest(k,i).gt.0) then
      iu=iabs(irest(j,i)-irest(k,i))                                    ! 34
      if(iubw.lt.iu) iubw=iu
      endif
      enddo
      endif
      enddo
  101 continue
      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. 両端せん断変形領域なしの場合の代表的なエレメント番号を、data 文を用いて配列 nn_type にセットする。
2. 代表的な節点拘束表を、data 文を用いて配列 irest にセットする。
3. 各エレメントの長さの比率を data 文を用いて配列 aleng に設定する。
4. 両端もしくは片方にせん断変形領域が存在する場合を考慮して、次の 4 タイプに分類する。
   １．両端せん断変形領域なし
   ２．i 端せん断変形領域あり
   ３．j 端せん断変形領域あり
   ４．両端せん断変形領域あり
5. 上で分類したタイプ別に処理を分類する。これ以降はタイプ１の両端せん断変形領域なしの場合に関する処理を行う。
6. ここにでの処理では、部材の分割数（エレメント数）n_div を 2 つ減らして４とする。
7. 部材の各エレメントにエレメント型番号をセットする。
8. 剛域長さを取り除いた実際の部材長さを計算する。
9. 各エレメントの比率 aleng(i)（データ文で比率をセット）を部材長さ al に掛けて、エレメントの長さをセットする。
10. エレメント数に１を足して、節点数 nn にセットする。
11. 節点数及び節点自由度数分、data 文でセットした拘束表 irest を節点拘束表 irest_Point にコピーする。
12. この部材のねじり剛性がゼロの場合、内部節点のねじり剛性項に関連する自由度を拘束する。もしこの拘束を行わないと、ねじり剛性がないため、該当する内部節点自由度の位置で、剛性行列の LDU 分

エレメント型番号は以下のようである。
１：弾性
２：ファイバー
３：せん断

解時に、エラーが発生する。

13. ｙ軸の断面二次モーメントがゼロの場合、上記と同じ理由で該当する内部節点自由度を拘束する。
14. 同様に、ｚ軸の断面二次モーメントがゼロの場合、該当する内部節点自由度を拘束する。
15. 以降では、内部節点自由度に関連する拘束表から未知番号表を作成する。ここでは、まず未知番号 n_if に 0 をセットする。
16. 内部節点及び 6 自由度について、以下の処理を行う。
17. 拘束番号 n_if が 0 以上で、しかも 1 に等しい場合、その自由度は拘束なしであるため、未知番号を 1 増やして、その値を拘束表にセットする。
18. 拘束番号が 0 以上で、しかも 1 でない場合、その自由度は他の節点自由度と同じである。そのため、仕様にしたがって一桁目の値を自由度番号 ip、二桁目以上を節点番号とし、その節点番号と自由度番号の拘束値を拘束表にセットする。この作業が終了すると内部節点の全自由度 n_if が決定される。
19. サブルーチン set_iubw()を用いて半バンド幅 iubw を求める。これで両端せん断変形領域なしの処理が終了する。
20. ここからが、i 端せん断変形領域ありの場合の処理である。ここからの説明は、前記の説明以外の部分について行う。
21. ここでの処理は、部材の分割数（エレメント数）n_div を 1 つ減らして 5 とする。
22. 最初のエレメントのエレメント型番号をせん断ばね型：3 とする。
23. エレメント型番号をひとつずらしてセットする。後の処理は両端せん断変形領域なしと同じである。
24. ここからが j 端せん断変形領域ありの場合の処理である。
25. 最後のエレメントのエレメント型番号をせん断ばね型：3 とする。
26. ここからが両端せん断変形領域ありの場合の処理である。
27. 最初と最後のエレメントのエレメント型番号をせん断ばね型：3 とする。
28. このサブルーチン set_iubw()では、未知番号表を用いて半バンド幅を求める。まず、半バンド幅 iubw をゼロセットする。
29. エレメント数分、以下の処理を行う。ここではエレメントの両節点について調査する。
30. エレメントの左節点の自由度 6 について以下の処理を行う。
31. その自由度 i が拘束されていないことを確かめる。

32. 左の節点と右の節点の間で、未知番号の差が最大のものを半バンド幅 iubw にセットする。
33. 全節点数について、以下の処理を行う。
34. 次は、当該の節点内で、未知番号の差が最大のものを半バンド幅 iubw にセットする。

以上のように、このサブルーチンで静的縮合モデルの部材解析モデルとして、未知数番号表と全未知数、半バンド幅が決定される。このようなサブルーチンは他の静的縮合モデルにも存在し、同様な処理を行っている。詳細は付録を参照されたい。

5.9.5 ファイバーモデルの剛性

次に、両端ファイバーモデルの弾塑性剛性$[K_L]$を作成するサブルーチンを見てみよう。このサブルーチンは第２層の階層構造を有している。サブルーチン Stiff_M11()を以下に示す。

```
C     ―――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Stiff_M11
C       ―――――――――――――――――――――――――――――――
C     ●   接線剛性行列の計算(ok)
C     ―――――――――――――――――――――――――――――――
      subroutine Stiff_M11(im,it,n_type,ak,Member,alength,
     *       Model_type,Element,
     *     E_model11, E_model_fiber,M_model11, 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_model11_s       / E_model11
      record / M_model11_s       / M_model11
      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―――――――――――――――――――――――――――――――★弾性要素
```

```
      call Cal_nonlin_stiff_Mx(Member,Element,
     *       Member.stress(7),ak,alength(im) )
      goto 100
   12 continue
c――――――――――――――――――――――――――――――――★ファイバー要素
      it=it+1
      call Fiber_Model_G11(it,ak,alength(im),Member,Element,
     *     E_model11,E_model_fiber,
     *     M_model11,M_model_fiber)
      goto 100
   13 continue
c――――――――――――――――――――――――――――――――★せん断バネ要素
      call Shear_G(it,ak,Member)
      goto 100
   14 continue
c――――――――――――――――――――――――――――――――★ダミー
      goto 100
   15 continue
c――――――――――――――――――――――――――――――――★ダミー
      goto 100
   16 continue
c――――――――――――――――――――――――――――――――★ダミー
      goto 100
   17 continue
c――――――――――――――――――――――――――――――――★ダミー
      goto 100
   18 continue
c――――――――――――――――――――――――――――――――★ダミー
      goto 100
   19 continue
c――――――――――――――――――――――――――――――――★ダミー
      goto 100
   20 continue
c――――――――――――――――――――――――――――――――★ダミー
  100 continue
      return
      end
```

このサブルーチンは、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、エレメント型番号 n_type によって、階層構造となっている。最初に、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 は弾性エレメント、2 はファイバーエレメント、3 は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。弾性エレメント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。

エレメント型番号 2 のファイバーにおけるサブルーチンは、ファイバー断面を有するエレメントの弾塑性剛性行列を求める処理を行う。この

弾塑性剛性行列に関連する３つのサブルーチン Fiber_Model_GI11()と Fiber_Model_G11()及び Fiber_GK()を以下に示し、説明する。最初のサブルーチンは初期設定であり、各種のフイバー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、各種のデータを構造体から取得するルーチンであり、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 /Fiber_Model_GI11
C     ―――――――――――――――――――――――――――――――――
C     ●   線形剛性行列の計算(ok)
C     ―――――――――――――――――――――――――――――――――
C     ●   Model_No.11 両端ファイバーモデル
C     ―――――――――――――――――――――――――――――――――
      subroutine Fiber_Model_GI11(it,ak,alength,Member,Element,
     *          E_model,E_model_fiber,M_model,M_model_fiber,
     *          Bilinear_work,Trilinear_work,Concrete_work)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      include "submainx.h"
      record / member_s           / Member
      record / element_s          / Element
      record / E_model11_s        / E_model
      record / E_model_fiber_s    / E_model_fiber
      record / M_model11_s        / M_model
      record / M_model_fiber_s    / M_model_fiber
      record / Bilinear_work_s    / Bilinear_work
      record / Trilinear_work_s   / Trilinear_work
      record / Concrete_work_s    / Concrete_work
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension Bilinear_work(*),Trilinear_work(*)
      dimension Concrete_work(*)
      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                                      ! 4
      nm_div = E_model.n_section_2
      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―――――――――――――――――――――――――――――――★データの初期設定
```

```
      M_model_fiber(nnm).d_E       = E_model_fiber(nn).E_1                    ! 7
      M_model_fiber(nnm).i_elastic  = 0        !  ファイバー要素の状態（弾性の場合は０：塑性は１）
      M_model_fiber(nnm).d_eps_x    = 0.       !  軸方向歪
      M_model_fiber(nnm).d_stress_x = 0.       !  軸方向応力
c―――――――――――――――――――――――――★ファイバーデータセット
      E_model_fiber(nn).Ary  = E_model_fiber(nn).A*E_model_fiber(nn).ry       ! 8
      E_model_fiber(nn).Arz  = E_model_fiber(nn).A*E_model_fiber(nn).rz
      E_model_fiber(nn).Ary2 =
     *                       E_model_fiber(nn).Ary*E_model_fiber(nn).ry
      E_model_fiber(nn).Arz2 =
     *                       E_model_fiber(nn).Arz*E_model_fiber(nn).rz
      E_model_fiber(nn).Aryz =
     *                       E_model_fiber(nn).Arz*E_model_fiber(nn).ry
      nmx                    = M_model_fiber(nnm).n_type
      nm_type                = E_model_fiber(nn).nm_type
      goto ( 10,20,30,30,10,20,10,20),nm_type                               ! 9
  10  continue
c―――――――――――――――――――――――――★バイリニア型（スチール用）
      Bilinear_work(nmx).i_stat  = -1             !  ファイバー要素の状態    ! 10
      goto 100
  20  continue
c―――――――――――――――――――――――――★対称トリリニア型                ! 11
      Trilinear_work(nmx).i_stat = -1             !  ファイバー要素の状態
      goto 100
  30  continue
c―――――――――――――――――――――――――★コンクリート型                ! 12
      Concrete_work(nmx).i_stat  = -1             !  ファイバー要素の状態
      goto 100
  100 continue
      enddo
c―――――――――――――――――――――――――――――――――――――――――★ファイバー要素のデータセット
      if(it.eq.1) then                                                      ! 13
      nm_div = E_model.n_section_1
      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
      nn     = E_model.nm_section_2 - 1
      nnm    = M_model.nm_section_2 - 1
      endif
      ra     = 0.                                                           ! 14
      ray    = 0.
      raz    = 0.
      raz2   = 0.
      ray2   = 0.
      rayz   = 0.
      gg     = 0.
      aa     = 0.
      do i=1,nm_div                                                         ! 15
      nn     = nn+1
      nnm    = nnm+1
      A      = E_model_fiber(nn).A
      E      = M_model_fiber(nnm).d_E
      ra     = ra   + E*A                                                   ! 16
```

```
      ray    = ray  + E*E_model_fiber(nn).Arz
      raz    = raz  + E*E_model_fiber(nn).Ary
      ray2   = ray2 + E*E_model_fiber(nn).Arz2
      raz2   = raz2 + E*E_model_fiber(nn).Ary2
      rayz   = rayz + E*E_model_fiber(nn).Aryz
      aa     = aa   + A
      gg     = gg   + A*E_model_fiber(nn).G
      enddo
      gg     = gg / aa
      if(it.eq.1) then                                                  ! 17
      M_model.d_aa_1     = aa             !  断面積の和
      M_model.d_ra_1     = ra             !  E*断面積の和
      M_model.d_ray_1    = ray            !  E*A*z
      M_model.d_raz_1    = raz            !  E*A*y
      M_model.d_raz2_1   = raz2           !  E*A*y*y
      M_model.d_ray2_1   = ray2           !  E*A*z*z
      M_model.d_rayz_1   = rayz           !  E*A*z*y
      M_model.d_gg_1     = gg             !  G*A
      M_model.d_epsi_x_1 =  0.            !  軸方向歪
      M_model.d_epsi_y_1 =  0.            !  y 軸に関する曲げ歪
      M_model.d_epsi_z_1 =  0.            !  z 軸に関する曲げ歪
      else
      M_model.d_aa_2     = aa             !  断面積の和
      M_model.d_ra_2     = ra             !  E*断面積の和
      M_model.d_ray_2    = ray            !  E*A*z
      M_model.d_raz_2    = raz            !  E*A*y
      M_model.d_raz2_2   = raz2           !  E*A*y*y
      M_model.d_ray2_2   = ray2           !  E*A*z*z
      M_model.d_rayz_2   = rayz           !  E*A*z*y
      M_model.d_gg_2     = gg             !  G*A
      M_model.d_epsi_x_2 =  0.            !  軸方向歪
      M_model.d_epsi_y_2 =  0.            !  y 軸に関する曲げ歪
      M_model.d_epsi_z_2 =  0.            !  z 軸に関する曲げ歪
      endif
c-----------------------------------★初期剛性行列の計算
      call Fiber_Model_G11(it,ak,alength,Member,Element,                ! 18
     *     E_model,E_model_fiber,M_model,M_model_fiber)
      return
      end
C     ------------------------------------------------
C     ●   SUBROUTINE /Fiber_Model_G11
C     ------------------------------------------------
C     ●   接線剛性行列の計算
C     ------------------------------------------------
      subroutine Fiber_Model_G11(it,ak,alength,Member,Element,
     *     E_model,E_model_fiber,M_model,M_model_fiber)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      include "submainx.h"
C     ------------------------------------------------
C     ●   Model_No.11 両端ファイバーモデル E_model11_s 構造体
C     ------------------------------------------------
c     部材
c      structure / E_model11_s/
```

```
c      integer  nm_section_1        !  i 端断面のファイバー開始番号
c      integer  n_section_1         !  i 端断面のファイバー数
c      integer  nm_section_2        !  j 端断面のファイバー開始番号
c      integer  n_section_2         !  j 端断面のファイバー数
c      real*8   i_rigid_length      !  i 端剛域長さ
c      real*8   j_rigid_length      !  j 端剛域長さ
c      real*8   i_shear_G           !  i 端せん断剛性
c      real*8   j_shear_G           !  j 端せん断剛性
c      end structure
c      record / E_model11_s    / E_model11
c
C     ―――――――――――――――――――――――――――――――――
C     ●　ファイバーモデル E_model_fiber_s 構造体
C     ―――――――――――――――――――――――――――――――――
c     部材
c      structure / E_model_fiber_s/
c      integer  nm_type              !  履歴モデルの番号
c      real*8   E_1                  !  ファイバーの第一剛性 E1
c      real*8   E_2                  !  ファイバーの第二剛性 E2
c      real*8   Q_1                  !  ファイバーの第一折れ点
c      real*8   G                    !  ファイバー断面積 G
c      real*8   A                    !  ファイバー断面積
c      real*8   Ay                   !  ファイバーｙ軸せん断用断面積
c      real*8   Az                   !  ファイバーｚ軸せん断用断面積
c      real*8   ry                   !  中立軸から断面中心までのｙ方向距離
c      real*8   rz                   !  中立軸から断面中心までのｚ方向距離
c      real*8   Ary                  !  断面積掛ける距離
c      real*8   Arz                  !  断面積掛ける距離
c      real*8   Ary2                 !  断面積掛ける距離の２乗
c      real*8   Arz2                 !  断面積掛ける距離の２乗
c      real*8   Aryz                 !
c      end structure
C     ―――――――――――――――――――――――――――――――――
C     ●　Model_No.11 両端ファイバーモデル M_model11_s 構造体
C     ―――――――――――――――――――――――――――――――――
c     部材
c      structure / M_model11_s/
c      integer  nm_section_1        !  i 端断面のファイバー開始番号
c      integer  n_section_1         !  i 端断面のファイバー数
c      real*8  d_state_1            !  ファイバー要素分割断面の弾塑性状態
c      real*8  d_aa_1               !  断面積の和
c      real*8  d_ra_1               !  E*断面積の和
c      real*8  d_ray_1              !  E*A*z
c      real*8  d_raz_1              !  E*A*y
c      real*8  d_raz2_1             !  E*A*y*y
c      real*8  d_ray2_1             !  E*A*z*z
c      real*8  d_rayz_1             !  E*A*z*y
c      real*8  d_gg_1               !  G*A
c      real*8  d_epsi_x_1           !  軸方向歪
c      real*8  d_epsi_y_1           !  ｙ軸に関する曲げ歪
c      real*8  d_epsi_z_1           !  ｚ軸に関する曲げ歪
c      integer  nm_section_2        !  j 端断面のファイバー開始番号
c      integer  n_section_2         !  j 端断面のファイバー数
c      real*8  d_state_2            !  ファイバー要素分割断面の弾塑性状態
```

```
c      real*8  d_aa_2              !  断面積の和
c      real*8  d_ra_2              !  E*断面積の和
c      real*8  d_ray_2             !  E*A*z
c      real*8  d_raz_2             !  E*A*y
c      real*8  d_raz2_2            !  E*A*y*y
c      real*8  d_ray2_2            !  E*A*z*z
c      real*8  d_rayz_2            !  E*A*z*y
c      real*8  d_gg_2              !  G*A
c      real*8  d_epsi_x_2          !  軸方向歪
c      real*8  d_epsi_y_2          !  y 軸に関する曲げ歪
c      real*8  d_epsi_z_2          !  z 軸に関する曲げ歪
c      real*8  disp_p(12)          !  両端の変位
c      real*8  disp_ip(18)         !  内部節点の変位
c      end structure
C     ―――――――――――――――――――――――――――――――――――
C     ●   Model_No.11 両端ファイバーモデル M_model_fiber_s 構造体
C     ―――――――――――――――――――――――――――――――――――
c     部材
c      structure / M_model_fiber_s/
c      integer  n_type             !  履歴モデルの通し番号
c      integer  i_submodel         !  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_sigma_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
      record / member_s          / Member
      record / element_s         / Element
      record / E_model11_s       / E_model
      record / M_model11_s       / M_model
      record / M_model_fiber_s   / M_model_fiber
      record / E_model_fiber_s   / E_model_fiber
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension ak(12,12)
c―――――――――――――――――――――――――――――――★
      if(it.eq.1) then                !  it:部材位置（１：i 端、 ２：j 端）      19
      aa   = M_model.d_aa_1           !  断面積の和
      ra   = M_model.d_ra_1           !  E*断面積の和
      ray  = M_model.d_ray_1          !  E*A*z
      raz  = M_model.d_raz_1          !  E*A*y
      raz2 = M_model.d_raz2_1         !  E*A*y*y
      ray2 = M_model.d_ray2_1         !  E*A*z*z
      rayz = M_model.d_rayz_1         !  E*A*z*y
      gg   = M_model.d_gg_1           !  G*A
      else
      aa   = M_model.d_aa_2           !  断面積の和
      ra   = M_model.d_ra_2           !  E*断面積の和
      ray  = M_model.d_ray_2          !  E*A*z
      raz  = M_model.d_raz_2          !  E*A*y
      raz2 = M_model.d_raz2_2         !  E*A*y*y
      ray2 = M_model.d_ray2_2         !  E*A*z*z
      rayz = M_model.d_rayz_2         !  E*A*z*y
      gg   = M_model.d_gg_2           !  G*A
      endif
```

```
      rix  = Element.RIx
      asy  = Element.ASy
      asz  = Element.ASz
c―――――――――――――――――――――――――――――――――★
      call Fiber_GK(ak,alength,ra,ray,raz,                          ! 20
     *              raz2,ray2,rayz,gg,aa,rix,asy,asz)
      return
      end
C     ―――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Fiber_GK
C     ―――――――――――――――――――――――――――――――――――――
C     ●   ファイバー要素の剛性行列
C     ―――――――――――――――――――――――――――――――――――――
      subroutine Fiber_GK(ak,RL,ra,ray,raz,riz2,riy2,riyz,
     *                    G,aa,rix,asy,asz)
      implicit real*8(A-H,O-Z)
      dimension ak(12,12)
      ak(1,1) = ra/RL                                               ! 21
      ak(1,7) =-ak(1,1)
      ak(7,7) = ak(1,1)
      ak(1,5) = ray/RL
      ak(1,11)=-ak(1,5)
      ak(5,7) =-ak(1,5)
      ak(7,11)= ak(1,5)
      ak(1,12)= raz/RL
      ak(1,6) =-ak(1,12)
      ak(6,7) = ak(1,12)
      ak(7,12)=-ak(1,12)
      ak(5,12)= riyz/RL
      ak(5,6) =-ak(5,12)
      ak(6,11)=  ak(5,12)
      ak(11,12)=-ak(5,12)
      RL2=RL*RL
      RL3=RL*RL*RL
      IF(asy.NE.0.0.OR.asz.NE.0.0) GOTO 20                          ! 22
      ZERL2   = 6.*riz2/RL2
      ZERL3   = 2.*ZERL2/RL
      ak(2,2) = ZERL3
      ak(2,6) = ZERL2
      ak(2,8) =-ZERL3
      ak(2,12)= ZERL2
      YERL2   = 6.*riy2/RL2
      YERL3   = 2.*YERL2/RL
      ak(3,3) = YERL3
      ak(3,5) =-YERL2
      ak(3,9) =-YERL3
      ak(3,11)=-YERL2
      ak(4,4) = G*RIX/RL
      ak(4,10)=-ak(4,4)
      ak(5,5) = 4.0*riy2/RL
      ak(5,9) = YERL2
      ak(5,11)= ak(5,5)*0.5
      ak(6,6) = 4.0*riz2/RL
      ak(6,8) =-ZERL2
```

```
      ak(6,12)= ak(6,6)*0.5
      ak(8,8)=  ZERL3
      ak(8,12)=-ZERL2
      ak(9,9)=  YERL3
      ak(9,11)= YERL2
      ak(10,10)= ak(4,4)
      ak(11,11)= ak(5,5)
      ak(12,12)= ak(6,6)
      do  i=1,11
      do  j=i+1,12
      ak(j,i) =ak(i,j)
      enddo
      enddo
      return
C     ----------------------------------------------------------------
C     ---  せん断変形を考慮  -----------------------------------------
C     ----------------------------------------------------------------
   20 CONTINUE                                                        ! 23
      FY = 0.0
      FZ = 0.0
      IF(ASY.NE.0.0) FY=12.*riz2/(G*ASY*RL2)
      IF(ASZ.NE.0.0) FZ=12.*riy2/(G*ASZ*RL2)
      ZERL3   = 12.*riz2/(RL3*(1.+FY))
      ZERL2   = 6.*riz2/(RL2*(1.+FY))
      ak(2,2) = ZERL3
      ak(2,6) = ZERL2
      ak(2,8) =-ZERL3
      ak(2,12)= ZERL2
      YERL3   = 12.*riy2/(RL3*(1.+FZ))
      YERL2   = 6.*riy2/(RL2*(1.+FZ))
      ak(3,3) = YERL3
      ak(3,5) =-YERL2
      ak(3,9) =-YERL3
      ak(3,11)=-YERL2
      ak(4,4) = G*RIX/RL
      ak(4,10)=-ak(4,4)
      ak(5,5) = (4.0+FZ)*riy2/(RL*(1.+FZ))
      ak(5,9) = YERL2
      ak(5,11)= (2.0-FZ)*riy2/(RL*(1.+FZ))
      ak(6,6) = (4.0+FY)*riz2/(RL*(1.+FY))
      ak(6,8) =-ZERL2
      ak(6,12)= (2.0-FY)*riz2/(RL*(1.+FY))
      ak(8,8) = ZERL3
      ak(8,12)=-ZERL2
      ak(9,9) = YERL3
      ak(9,11)= YERL2
      ak(10,10)= ak(4,4)
      ak(11,11)= ak(5,5)
      ak(12,12)= ak(6,6)
      DO  i=1,11                                                      ! 24
      DO  j=i+1,12
      ak(j,i) =ak(i,j)
      enddo
      enddo
```

```
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。

1. 両端ファイバーモデルでは、i 端と j 端でファイバー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 端であり、2 は j 端である。
2. 内部節点の不釣合力の保存場所である構造体内の成分である M_model.ff_ip()をゼロクリアする。
3. 構造体より i 端のファイバーデータの管理用パラメータを取得する。それらは、その断面におけるファイバー数 nm_div、ファイバー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 端に関するファイバーデータの管理用パラメータを取得する。
5. 以下の処理を、その断面の全ファイバー数分行う。
6. 上記の nn 及び nnm に１を加え、構造体への配列番号をセットする。
7. 入力データ以外の各ファイバー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、ファイバーのヤング係数、ファイバーの弾塑性状態を表すパラメータ、軸方向ひずみ、軸方向応力をゼロ、もしくは値をセットする。
8. 次に、ファイバーデータの断面図芯位置における断面特性を計算し、要素に関連する構造体 E_model にセットする。この値は、解析途中で変更しない。
9. ファイバーの履歴特性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、ここでは初期設定であるので、ファイバー履歴番号が 4 以上では、ワーク領域が同じであることより、１から３までの処理で行われる。
10. バイリニアデータの初期状態を-1 にセットする。
11. トリリニアデータの初期状態を-1 にセットする。
12. コンクリートデータの初期状態を-1 にセットする。
13. 次の処理を行うため、構造体より、i 端及び j 端のファイバーデータの管理用パラメータを取得する。
14. ファイバー断面の断面積や断面一次モーメント、断面二次モーメントを求めるために、ここでは各パラメータをゼロセットしている。

15. その断面における全ファイバーについて次の処理を行う。
16. 各ファイバーについて断面積や断面一次モーメントなどを計算し、和を取る処理を行う。
17. 断面全体で得られた断面積や断面一次モーメントなどを構造体にセットする。
18. ここで、この構造体でセットした値を用いて、この断面を要するエレメントの剛性行列をサブルーチン Fiber_Model_G11()で計算する。
19. サブルーチン Fiber_Model_G11()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。
20. この取得したパラメータとこのエレメントの長さを用いて、実際の剛性行列をサブルーチン Fiber_GK()を用いて計算する。
21. ここでは、ファイバーエレメントの剛性行列を計算する。ファイバー断面の剛性行列は、式(5.14)の行列と比較して確認されたい。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。
22. せん断変形用の断面積がセットされている場合は、せん断変形を考慮するとして文番号 20 へ制御が移る。せん断変形を考慮しない場合は以降の処理を行う。
23. せん断変形を考慮する場合は、以降の処理を行う。
24. 対称行列であるため、下三角行列部分へ値をコピーする。

5.9.6 ファイバーモデルの弾塑性チェック

最後に残ったのがファイバー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 Fiber_check()をコールするコードは、サブルーチン Cal_check_stiff_M11()の中で､以下のように記述されている。

```
c――――――――――――――――――――――――――★ファイバーチェック
      if(n_type(i).eq.2) then
      it=it+1
      call Fiber_check(Control,i,N_analysis,
     *      mem_x,it,alength(i),Member,Element,
     *      E_model11(imm),E_model_fiber,M_model11(immm),M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work,vvx)
      endif
```

サブルーチン Fiber_check()の内容を以下の示す。

```
C     ――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Fiber_check11
```

```
C     ―――――――――――――――――――――――――――――――――――
C     ●　ファイバー要素の材料非線形性チェックし、応力を計算
C     ―――――――――――――――――――――――――――――――――――
      subroutine Fiber_check(Control,idiv,N_analysis,
     *      mem_x,it,Alength,Member,Element,
     *      E_model,E_model_fiber,M_model,M_model_fiber,
     *      Bilinear_work,Trilinear_work,Concrete_work,vv)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      include "submainx.h"
      record /control_s    / Control
      record / member_s           / Member
      record / element_s          / Element
      record / E_model11_s        / E_model
      record / E_model_fiber_s    / E_model_fiber
      record / M_model11_s        / M_model
      record / M_model_fiber_s    / M_model_fiber
      record / Bilinear_work_s    / Bilinear_work
      record / Trilinear_work_s   / Trilinear_work
      record / Concrete_work_s    / Concrete_work
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension Bilinear_work(*),Trilinear_work(*)
      dimension Concrete_work(*)
      dimension vv(12)
c―――――――――――――――――――――――――――――――★i 端部ファイバー
c―――――――――――――――――――――――――――――――★歪のセット
      if(it.eq.1) then                                                  ! 1
      d_epsi_x_1 =  (vv(7)  - vv(1)) / Alength   !  軸方向歪             2
      d_epsi_y_1 =  (vv(11) - vv(5)) / Alength   !  y 軸に関する曲げ歪
      d_epsi_z_1 =  (vv(12) - vv(6)) / Alength   !  z 軸に関する曲げ歪
      if(N_analysis.eq.10.or.N_analysis.eq.8) then                ! 非線形軸方向歪  3
      d_epsi_x_1= d_epsi_x_1+strain_nonlinear(Member.an_vv(idiv),
     *      Member.an_ww(idiv),vv,Alength)
      endif
      if(Member.analysis_3D .eq. 1) d_epsi_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_epsi_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_epsi_x_1 =  d_epsi_x_1 + M_model.d_epsi_x_1    !  軸方向歪
      M_model.d_epsi_y_1 =  d_epsi_y_1 + M_model.d_epsi_y_1    !  y 軸に関する曲げ歪
      M_model.d_epsi_z_1 =  d_epsi_z_1 + M_model.d_epsi_z_1    !  z 軸に関する曲げ歪
c―――――――――――――――――――――――――――――――★ファイバー要素のチェック
      nm_div   = E_model.n_section_1                                    ! 4
      nn       = E_model.nm_section_1 - 1
      nnm      = M_model.nm_section_1 - 1
      iistat   = 0                          ! 塑性率を計算するための指標  5
      do i=1,nm_div                                                     ! 6
      nn       = nn  + 1
      nnm      = nnm + 1
      nmx      = M_model_fiber(nnm).n_type
      nm_type  = E_model_fiber(nn).nm_type
      du =  d_epsi_x_1 +                                  ! 軸方向歪     7
     *      d_epsi_y_1 * E_model_fiber(nn).rz -           ! y 軸に関する曲げ歪
     *      d_epsi_z_1 * E_model_fiber(nn).ry             ! z 軸に関する曲げ歪
      if(N_analysis.le.8.or.Member.nm_analysis.eq.-1) then
c―――――――――――――――――――――――――――――――★弾性解析                ! 8
```

```
c                                        ファイバー軸力計算
      M_model_fiber(nnm).d_stress_x = M_model_fiber(nnm).d_E*du +
     *                               M_model_fiber(nnm).d_stress_x
      else
c―――――――――――――――――――――――――★弾塑性解析
      goto ( 10,20,30,40,50,60,70,80),nm_type                          ! 9
  10  continue
c―――――――――――――――――――――――――★バイリニア型（スチール用）
      call BiLinear(M_model_fiber(nnm).d_E,Bilinear_work(nmx).i_stat,   ! 10
     *              E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *              E_model_fiber(nn).Q_1,du,
     *              M_model_fiber(nnm).d_stress_x,Bilinear_work(nmx).P1)
      if(Bilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      if(Bilinear_work(nmx).i_stat.ne.0) then
      endif
      goto 100
  20  continue
c―――――――――――――――――――――――――★対称トリリニア型 （スチール用）
      call TriLinear(M_model_fiber(nnm).d_E,Trilinear_work(nmx).i_stat,
     *               E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *               E_model_fiber(nn).E_3,
     *               E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *               du,M_model_fiber(nnm).d_stress_x,
     *               Trilinear_work(nmx).P1(1))
      if(Trilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      goto 100
  30  continue
c―――――――――――――――――――――――――★コンクリート型
      call Concrete(M_model_fiber(nnm).d_E,Concrete_work(nmx).i_stat,   ! 11
     *              E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *              E_model_fiber(nn).E_3,E_model_fiber(nn).Ec_3,
     *              E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *              E_model_fiber(nn).Ec_1,E_model_fiber(nn).Ec_2,
     *              du,  M_model_fiber(nnm).d_stress_x,
     *              Concrete_work(nmx).p1(1),Concrete_work(nmx).P1(2),
     *              Concrete_work(nmx).ipret,Concrete_work(nmx).P1(3),
     *              Concrete_work(nmx).p1(4),Concrete_work(nmx).P1(5),
     *              Concrete_work(nmx).ipre_c)
      if(Concrete_work(nmx).i_stat.ne.0.and.Concrete_work(nmx).i_stat
     *            .ne.3) iistat = iistat + 1
      goto 100
  40  continue
c―――――――――――――――――――――――――★曲線コンクリート型
      call Concrete_e(M_model_fiber(nnm).d_E,Concrete_work(nmx).i_stat,
     *              E_model_fiber(nn).E_1,E_model_fiber(nn).E_3,
     *              E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *              du,  M_model_fiber(nnm).d_stress_x,
     *              Concrete_work(nmx).P1(1),Concrete_work(nmx).P1(2),
     *              Concrete_work(nmx).ipret,Concrete_work(nmx).ipre_c,
     *              Concrete_work(nmx).P1(3),Concrete_work(nmx).P1(4),
     *              Concrete_work(nmx).P1(5),E_model_fiber(nn).Ec_1,
     *              Concrete_work(nmx).P1(6),E_model_fiber(nn).Ec_2)
      if(Concrete_work(nmx).i_stat.eq.0)then
      if(du.lt.E_model_fiber(nn).Ec_1)then
```

```
      iistat = iistat + 1
      endif
      elseif(Concrete_work(nmx).i_stat.ne.3)then
      iistat = iistat + 1
      endif
      goto 100
   50 continue
c――――――――――――――――――――――――――★バイリニア型（移動＋等方硬化用）
      call BiLinear_h(M_model_fiber(nnm).d_E,Bilinear_work(nmx).i_stat,
     *              E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *              E_model_fiber(nn).Q_1,du,
     *              M_model_fiber(nnm).d_stress_x,Bilinear_work(nmx).P1,
     *              E_model_fiber(nn).Beta)
      if(Bilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      goto 100
   60 continue
c――――――――――――――――――――――――――★対称トリリニア型（移動＋等方硬化用）
      call TriLinear_h(M_model_fiber(nnm).d_E,
     *               Trilinear_work(nmx).i_stat,
     *               E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *               E_model_fiber(nn).E_3,
     *               E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *               du,M_model_fiber(nnm).d_stress_x,
     *              Trilinear_work(nmx).P1(1),Trilinear_work(nmx).P1(2),
     *              E_model_fiber(nn).Beta, E_model_fiber(nn).Beta_2)
      if(Trilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      goto 100
   70 continue
c――――――――――――――――――――――――――★非対称バイリニア型
      call BiLinear_AS(M_model_fiber(nnm).d_E,Bilinear_work(nmx).i_stat,
     *              E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *              E_model_fiber(nn).Q_1,E_model_fiber(nn).Ec_1,
     *              E_model_fiber(nn).Ec_2,E_model_fiber(nn).Qc_1,du,
     *              M_model_fiber(nnm).d_stress_x,Bilinear_work(nmx).P1,
     *              Bilinear_work(nmx).P2,Bilinear_work(nmx).Sig_z,
     *              E_model_fiber(nn).Beta)
      if(Bilinear_work(nmx).i_stat.eq.2.or.
     *   Bilinear_work(nmx).i_stat.eq.3) iistat = iistat + 1
      goto 100
   80 continue
c――――――――――――――――――――――――――★非対称トリリニア型
      call TriLinear_AS(M_model_fiber(nnm).d_E,
     *               Trilinear_work(nmx).i_stat,
     *               E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *               E_model_fiber(nn).E_3,E_model_fiber(nn).Ec_1,
     *               E_model_fiber(nn).Ec_2,E_model_fiber(nn).Ec_3,
     *               E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *               E_model_fiber(nn).Qc_1,E_model_fiber(nn).Qc_2,
     *               du,M_model_fiber(nnm).d_stress_x,
     *              Trilinear_work(nmx).P1(1),Trilinear_work(nmx).P1(2),
     *              Trilinear_work(nmx).P1(3),
     *               E_model_fiber(nn).Beta, E_model_fiber(nn).Beta_2)
      if(Trilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      goto 100
```

```
c――――――――――――――――――――――――――★弾塑性解析終了
      endif
  100 continue                                                          ! 12
      M_model_fiber(nnm).d_eps_x = M_model_fiber(nnm).d_eps_x + du !ファイバー要素の歪
      enddo
c――――――――――――――――――――――――――――――――――★ファイバー要素応力の計算
      d_state = float(iistat)/float(nm_div)   ! 塑性した面積の計算          ! 13
      if(d_state .eq.0) then
      Member.d_stat(1) = 0
      elseif(d_state .ge.0.8) then
      Member.d_stat(1) = 2
      else
      Member.d_stat(1) = 1
      endif
      nm_div  = E_model.n_section_1                                     ! 14
      nn      = E_model.nm_section_1 - 1
      nnm     = M_model.nm_section_1 - 1
      ra      = 0.                                                      ! 15
      ray     = 0.
      raz     = 0.
      raz2    = 0.
      ray2    = 0.
      rayz    = 0.
      gg      = 0.
      aa      = 0.
      AN      = 0.
      AMy     = 0.
      AMz     = 0.
      do i=1,nm_div                                                     ! 16
      nn      = nn  + 1
      nnm     = nnm + 1
      A       = E_model_fiber(nn).A
      E       = M_model_fiber(nnm).d_E
      ra      = ra   + E*A
      ray     = ray  + E*E_model_fiber(nn).Arz
      raz     = raz  + E*E_model_fiber(nn).Ary
      ray2    = ray2 + E*E_model_fiber(nn).Arz2
      raz2    = raz2 + E*E_model_fiber(nn).Ary2
      rayz    = rayz + E*E_model_fiber(nn).Aryz
      aa      = aa + A
      gg      = gg + A*E_model_fiber(nn).G
      ANN     = M_model_fiber(nnm).d_stress_x*E_model_fiber(nn).A
      AN      = AN + ANN
      AMy     = AMy + ANN * E_model_fiber(nn).rz
      AMz     = AMz + ANN * E_model_fiber(nn).ry
      enddo
      nn=E_model.nm_section_1
      call jikuzero_control(Control,ra,E_model_fiber(nn).E_1,           ! 17
     *                      E_model_fiber(nn).A)
      M_model.d_aa_1 = aa               !  断面積の和                   ! 18
      M_model.d_ra_1 = ra               !  E*断面積の和
      M_model.d_ray_1 = ray             !  E*A*z
      M_model.d_raz_1 = raz             !  E*A*y
      M_model.d_raz2_1 = raz2           !  E*A*y*y
```

```
      M_model.d_ray2_1 = ray2           !  E*A*z*z
      M_model.d_rayz_1 = rayz           !  E*A*z*y
      M_model.d_gg_1  = gg             !  G*A
c―――――――――――――――――――――――――――――――★j 端部ファイバー
c―――――――――――――――――――――――――――――――★歪のセット
      elseif(it.eq.2) then                                                  ! 19
      d_epsi_x_2 =  (vv(7)  - vv(1)) / Alength   !  軸方向歪
      d_epsi_y_2 =  (vv(11) - vv(5)) / Alength   !  y 軸に関する曲げ歪
      d_epsi_z_2 =  (vv(12) - vv(6)) / Alength   !  z 軸に関する曲げ歪
      if(N_analysis.eq.10.or.N_analysis.eq.8) then                ! 非線形軸方向歪
      d_epsi_x_2= d_epsi_x_2+strain_nonlinear(Member.an_vv(idiv),
     *      Member.an_ww(idiv),vv,Alength)
      endif
      if(Member.analysis_3D .eq. 1) d_epsi_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_epsi_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_epsi_x_2 = d_epsi_x_2 + M_model.d_epsi_x_2          !  軸方向歪
      M_model.d_epsi_y_2 = d_epsi_y_2 + M_model.d_epsi_y_2           !  y 軸に関する曲げ歪
      M_model.d_epsi_z_2 = d_epsi_z_2 + M_model.d_epsi_z_2          !  z 軸に関する曲げ歪
c―――――――――――――――――――――――――――――――★ファイバー要素のチェック
      nm_div      = E_model.n_section_2
      nn          = E_model.nm_section_2 - 1
      nnm         = M_model.nm_section_2 - 1
      iistat      = 0                              ! 塑性率を計算するための指標
      do i=1,nm_div
      nn      = nn  + 1
      nnm     = nnm + 1
      nmx     = M_model_fiber(nnm).n_type
      nm_type = E_model_fiber(nn).nm_type
      du =  d_epsi_x_2 +                           ! 軸方向歪
     *      d_epsi_y_2 * E_model_fiber(nn).rz -     ! y 軸に関する曲げ歪
     *      d_epsi_z_2 * E_model_fiber(nn).ry       ! z 軸に関する曲げ歪
      if(N_analysis.le.8.or.Member.nm_analysis.eq.-1) then
c―――――――――――――――――――――――★弾性解析
c                                    ファイバー軸力計算
      M_model_fiber(nnm).d_stress_x = M_model_fiber(nnm).d_E * du +
     *                               M_model_fiber(nnm).d_stress_x
      else
c―――――――――――――――――――――――★弾塑性解析
      goto ( 11,21,31,41,51,61,71,81),nm_type
  11  continue
c―――――――――――――――――――――――★バイリニア型（スチール用）
      call BiLinear(M_model_fiber(nnm).d_E,Bilinear_work(nmx).i_stat,
     *              E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *              E_model_fiber(nn).Q_1,du,
     *              M_model_fiber(nnm).d_stress_x,Bilinear_work(nmx).P1)
      if(Bilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      if(Bilinear_work(nmx).i_stat.ne.0) then
      endif
      goto 101
  21  continue
c―――――――――――――――――――――――★対称トリリニア型 （スチール用）
      call TriLinear(M_model_fiber(nnm).d_E,Trilinear_work(nmx).i_stat,
     *               E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *               E_model_fiber(nn).E_3,
```

```
     *              E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *              du,M_model_fiber(nnm).d_stress_x,
     *              Trilinear_work(nmx).P1(1))
      if(Trilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      goto 101
   31 continue
c-----------------------★コンクリート型
      call Concrete(M_model_fiber(nnm).d_E,Concrete_work(nmx).i_stat,
     *             E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *             E_model_fiber(nn).E_3,E_model_fiber(nn).Ec_3,
     *             E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *             E_model_fiber(nn).Ec_1,E_model_fiber(nn).Ec_2,
     *             du,  M_model_fiber(nnm).d_stress_x,
     *             Concrete_work(nmx).p1(1),Concrete_work(nmx).P1(2),
     *             Concrete_work(nmx).ipret,Concrete_work(nmx).P1(3),
     *             Concrete_work(nmx).p1(4),Concrete_work(nmx).P1(5),
     *             Concrete_work(nmx).ipre_c)
      if(Concrete_work(nmx).i_stat.ne.0.and.Concrete_work(nmx).i_stat
     *            .ne.3) iistat = iistat + 1
      goto 101
   41 continue
c-----------------------★曲線コンクリート型
      call Concrete_e(M_model_fiber(nnm).d_E,Concrete_work(nmx).i_stat,
     *             E_model_fiber(nn).E_1,E_model_fiber(nn).E_3,
     *             E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *             du,  M_model_fiber(nnm).d_stress_x,
     *             Concrete_work(nmx).P1(1),Concrete_work(nmx).P1(2),
     *             Concrete_work(nmx).ipret,Concrete_work(nmx).ipre_c,
     *             Concrete_work(nmx).P1(3),Concrete_work(nmx).P1(4),
     *             Concrete_work(nmx).P1(5),E_model_fiber(nn).Ec_1,
     *             Concrete_work(nmx).P1(6),E_model_fiber(nn).Ec_2)
      if(Concrete_work(nmx).i_stat.eq.0)then
      if(du.lt.E_model_fiber(nn).Ec_1)then
      iistat = iistat + 1
      endif
      elseif(Concrete_work(nmx).i_stat.ne.3)then
      iistat = iistat + 1
      endif
      goto 101
   51 continue
c-----------------------★バイリニア型移動＋等方硬化用）
      call BiLinear_h(M_model_fiber(nnm).d_E,Bilinear_work(nmx).i_stat,
     *             E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *             E_model_fiber(nn).Q_1,du,
     *             M_model_fiber(nnm).d_stress_x,Bilinear_work(nmx).P1,
     *             E_model_fiber(nn).Beta)
      if(Bilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      if(Bilinear_work(nmx).i_stat.ne.0) then
      endif
      goto 101
   61 continue
c-----------------------★対称トリリニア型（移動＋等方硬化用）
      call TriLinear_h(M_model_fiber(nnm).d_E,
     *              Trilinear_work(nmx).i_stat,
```

```
     *               E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *               E_model_fiber(nn).E_3,
     *               E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *               du,M_model_fiber(nnm).d_stress_x,
     *              Trilinear_work(nmx).P1(1),Trilinear_work(nmx).P1(2),
     *              E_model_fiber(nn).Beta, E_model_fiber(nn).Beta_2)
      if(Trilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      goto 101
  71  continue
c―――――――――――――――――――――――――★非対称バイリニア型
      call BiLinear_AS(M_model_fiber(nnm).d_E,Bilinear_work(nmx).i_stat,
     *              E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *              E_model_fiber(nn).Q_1,E_model_fiber(nn).Ec_1,
     *              E_model_fiber(nn).Ec_2,E_model_fiber(nn).Qc_1,du,
     *              M_model_fiber(nnm).d_stress_x,Bilinear_work(nmx).P1,
     *              Bilinear_work(nmx).P2,Bilinear_work(nmx).Sig_z,
     *              E_model_fiber(nn).Beta)
      if(Bilinear_work(nmx).i_stat.eq.2.or.
     *   Bilinear_work(nmx).i_stat.eq.3) iistat = iistat + 1
      goto 101
  81  continue
c―――――――――――――――――――――――――★非対称トリリニア型
      call TriLinear_AS(M_model_fiber(nnm).d_E,
     *               Trilinear_work(nmx).i_stat,
     *               E_model_fiber(nn).E_1,E_model_fiber(nn).E_2,
     *               E_model_fiber(nn).E_3,E_model_fiber(nn).Ec_1,
     *               E_model_fiber(nn).Ec_2,E_model_fiber(nn).Ec_3,
     *               E_model_fiber(nn).Q_1,E_model_fiber(nn).Q_2,
     *               E_model_fiber(nn).Qc_1,E_model_fiber(nn).Qc_2,
     *               du,M_model_fiber(nnm).d_stress_x,
     *              Trilinear_work(nmx).P1(1),Trilinear_work(nmx).P1(2),
     *              Trilinear_work(nmx).P1(3),
     *               E_model_fiber(nn).Beta, E_model_fiber(nn).Beta_2)
      if(Trilinear_work(nmx).i_stat.ne.0) iistat = iistat + 1
      goto 101
      endif
c―――――――――――――――――――――――――★弾塑性解析終了
  101 continue
      M_model_fiber(nnm).d_eps_x = M_model_fiber(nnm).d_eps_x + du !ファイバー要素の歪
      enddo
c―――――――――――――――――――――――――――――――――★ファイバー要素応力の計算
      d_state  = float(iistat)/float(nm_div)   ! 塑性した面積の計算
      if(d_state.eq.0) then
      Member.d_stat(2) = 0
      elseif(d_state .ge.0.8) then
      Member.d_stat(2) = 2
      else
      Member.d_stat(2) = 1
      endif
      nm_div           = E_model.n_section_2
      nn               = E_model.nm_section_2 - 1
      nnm              = M_model.nm_section_2 - 1
      ra    = 0.
      ray   = 0.
```

```
      raz   = 0.
      raz2  = 0.
      ray2  = 0.
      rayz  = 0.
      gg    = 0.
      aa    = 0.
      AN    = 0.
      AMy   = 0.
      AMz   = 0.
      do i=1,nm_div
      nn    = nn  + 1
      nnm   = nnm + 1
      A     = E_model_fiber(nn).A
      E     = M_model_fiber(nnm).d_E
      ra    = ra   + E * A
      ray   = ray  + E * E_model_fiber(nn).Arz
      raz   = raz  + E * E_model_fiber(nn).Ary
      ray2  = ray2 + E * E_model_fiber(nn).Arz2
      raz2  = raz2 + E * E_model_fiber(nn).Ary2
      rayz  = rayz + E * E_model_fiber(nn).Aryz
      aa    = aa   + A
      gg    = gg   + A * E_model_fiber(nn).G
      ANN   = M_model_fiber(nnm).d_stress_x * E_model_fiber(nn).A
      AN    = AN   + ANN
      AMy   = AMy  + ANN * E_model_fiber(nn).rz
      AMz   = AMz  + ANN * E_model_fiber(nn).ry
      enddo
      nn=E_model.nm_section_2
      call jikuzero_control(Control,ra,E_model_fiber(nn).E_1
     *                     ,E_model_fiber(nn).A)
      M_model.d_aa_2   = aa              !  断面積の和
      M_model.d_ra_2   = ra              !  E*断面積の和
      M_model.d_ray_2  = ray             !  E*A*z
      M_model.d_raz_2  = raz             !  E*A*y
      M_model.d_raz2_2 = raz2            !  E*A*y*y
      M_model.d_ray2_2 = ray2            !  E*A*z*z
      M_model.d_rayz_2 = rayz            !  E*A*z*y
      M_model.d_gg_2   = gg              !  G*A
      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. 両端ファイバーモデルでは、i 端と j 端でファイバー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 端であり、2 は j 端である。
2. この断面の増分ひずみである増分軸方向ひずみ、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。

3. 幾何学的非線形解析を行う場合は、非線形増分軸方向ひずみを足し込む。次に、解析が平面問題の場合、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。
4. 断面内のファイバーの数、要素に関連するファイバー用データの配列番号 -1 を nn に、部材に関連するファイバー用データの配列番号 -1 を nnm に設定する。
5. 断面内塑性率を計算するパラメータをゼロセットする。
6. 断面内の全ファイバーについて以下の処理を行う。まず、配列番号 nn と nnm に 1 を足し、次の配列番号をセットする。ファイバー履歴特性番号 nm_type とそのタイプのワーク用配列のエレメント番号を構造体より取得する。
7. 増分軸方向ひずみと各増分曲げひずみより、該当するファイバーの軸方向増分ひずみを求める。
8. 解析種別が弾性である場合、もしくはその部材が弾性部材として設定されている場合、弾性材であるとして軸力を計算する。
9. ファイバー履歴特性番号 nm_type にしたがって処理を分類する。ここで第 3 層の階層構造が見られる。現在、ファイバー履歴特性種類は８つであるが、現在使用許可になっているのは 1 と 3 の 2 つである。
10. ここではバイリニア型の履歴特性となっており、サブルーチン BiLinear()を用いて処理している。後は仕様にしたがって各サブルーチンがコールされることになる。
11. ここでは直線コンクリート型の履歴特性となっており、サブルーチン Concrete()を用いて処理している。
12. 履歴特性を処理する各サブルーチンの処理が終了すると、ここに処理が集結する。ここではファイバーの増分ひずみが増分前のひずみに足しこまれる。
13. 塑性化したファイバーの数を断面内のファイバーの総数で割った値を d_state にセットする。この値がゼロのとき断面は弾性であるとする。また、d_state が 0.8 以上の場合は塑性ヒンジが発生したとする。それ以外は弾塑性状態であるとする。
14. 断面全体の断面積や断面二次モーメントを求めるために、まずファイバーデータが保存されている配列の番号を取得する。
15. 各断面特性パラメータをゼロセットする。
16. 全ファイバーについて断面積などの和を取っていく。
17. 全てのファイバーについて上記の断面特性を計算した後、軸方向剛

性の調整をサブルーチン jikuzero_control()を用いて行う。ここでは、軸方向剛性が全断面塑性化してゼロとなるのを防ぐ処理を行っている。

18. 求めた断面特性を構造体にセットし、次の剛性行列を計算するときに使用する。
19. 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。

5.9.7 ファイバーモデルのせん断剛性

最後に、せん断変形を考慮する部材の剛性行列を作成するサブルーチンを以下に示す。プログラムは非常に単純なので容易に理解できよう。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Shear_G
C       ―――――――――――――――――――――――――――――――
C     ●   せん断バネ剛性行列の計算(ok)
C     ―――――――――――――――――――――――――――――――――
      subroutine Shear_G(it,ak,Member)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s    / Member
      dimension ak(12,12)
C     ―――――――――――――――――――――――――――――――――
      if(it.lt.1) then
      akk=Member.i_shear_G
      else
      akk=Member.j_shear_G
      endif
      ak(2,2)=akk
      ak(2,8)=-akk
      ak(8,2)=-akk
      ak(8,8)=akk
      ak(3,3)=akk
      ak(3,9)=-akk
      ak(9,3)=-akk
      ak(9,9)=akk
      return
      end
```

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。

$$[K_T]=[K_L]+[K_G]+[K_N] \qquad \cdots\cdots(5.16)$$

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、以下のように求められる。

$$[K_L]=\begin{bmatrix}
k_x-\frac{\alpha^2}{\omega} & 0 & 0 & 0 & -\frac{\alpha\beta}{\omega} & -\frac{\alpha\gamma}{\omega} & -\left(k_x-\frac{\alpha^2}{\omega}\right) & 0 & 0 & 0 & \frac{\alpha\beta}{\omega} & \frac{\alpha\gamma}{\omega} \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & k_{\theta y}-\frac{\beta^2}{\omega} & -\frac{\beta\gamma}{\omega} & \frac{\alpha\beta}{\omega} & 0 & 0 & 0 & -\left(k_{\theta y}-\frac{\beta^2}{\omega}\right) & \frac{\beta\gamma}{\omega} \\
 & & & & & k_{\theta z}-\frac{\gamma^2}{\omega} & \frac{\alpha\gamma}{\omega} & 0 & 0 & 0 & \frac{\beta\gamma}{\omega} & -\left(k_{\theta z}-\frac{\gamma^2}{\omega}\right) \\
 & & & & & & k_x-\frac{\alpha^2}{\omega} & 0 & 0 & 0 & -\frac{\alpha\beta}{\omega} & -\frac{\alpha\gamma}{\omega} \\
 & & & & & & & 0 & 0 & 0 & 0 & 0 \\
 & & & & sym & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & k_{\theta y}-\frac{\beta^2}{\omega} & -\frac{\beta\gamma}{\omega} \\
 & & & & & & & & & & & k_{\theta z}-\frac{\gamma^2}{\omega}
\end{bmatrix} \qquad \cdots\cdots(5.17)$$

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。

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

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。

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

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。

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

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。

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

$$f = \frac{1}{2M_B}\left\{\left(\frac{N}{N_P}-a\right)^2+\left(\frac{M_y}{M_{yP}}\right)^2+\left(\frac{M_z}{M_{zP}}\right)^2\right\}-\frac{r_0^2}{2M_B} = 0 \quad \cdots\cdots (5.21)$$

$$\left.\begin{aligned} f_n &= \frac{1}{M_B}\left(\frac{N}{N_P}-a\right)\frac{1}{N_P} \\ f_{my} &= \frac{1}{M_B}\left(\frac{M_y}{M_{yP}}\right)\frac{1}{M_{yP}} \\ f_{mz} &= \frac{1}{M_B}\left(\frac{M_z}{M_{zP}}\right)\frac{1}{M_{zP}} \end{aligned}\right\} \cdots\cdots (5.21a)$$

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

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

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

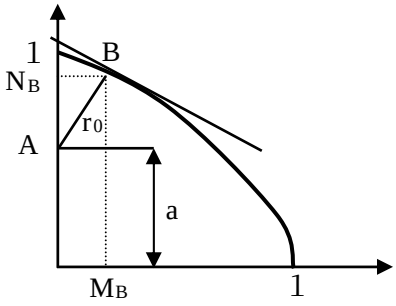

図 5-4 降伏関数の修正図

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つの降伏関数の接続点であり、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。

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

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。

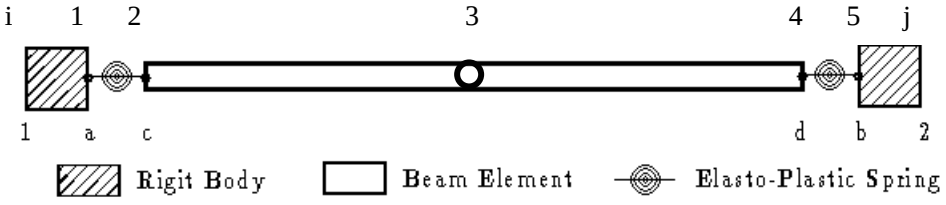

図 5-5 両端アナロジーモデルの部材解析モデル

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_model31, E_model_fiber,
     *      M_model31, 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_model31, E_model_fiber,
     *      M_model31, 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_model31, E_model_fiber,
     *      M_model31 , 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_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_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),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_ww(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(immm), 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(immm), 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――――――――――――――――――――――――――――――――★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――――――――――――――――――――――――――――――――★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),
     *   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(immm), 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_ww(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―――――――――――――――――――――――――★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(immm).nm_section_1
      else
      nnm=M_model31(immm).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(immm),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
      vvx(ii)=vv(-irest)
      elseif(irest.gt.0) then
      vvx(ii)=bav(irest)
      else
      vvx(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(immm),M_model_fiber,
     *      vvx,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),vvx,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)+(vvx(8) - vvx(2))
```

```
      Member.an_ww(i)=Member.an_ww(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――――――――――――――――――――――――――――――――――★弾性要素
      call Cal_nonlin_stiff_Mx(Member,Element,
     *        Member.stress(7),ak,alength(im) )
      goto 100
   12 continue
c――――――――――――――――――――――――――――――――――★アナロジー要素
      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――――――――――――――――――――――――――――――――――★せん断バネ要素
      call Shear_G(it,ak,Member)
      goto 100
   14 continue
c――――――――――――――――――――――――――――――――――★ダミー
      goto 100
   15 continue
c――――――――――――――――――――――――――――――――――★ダミー
      goto 100
   16 continue
c――――――――――――――――――――――――――――――――――★ダミー
      goto 100
   17 continue
c――――――――――――――――――――――――――――――――――★ダミー
      goto 100
   18 continue
c――――――――――――――――――――――――――――――――――★ダミー
      goto 100
   19 continue
c――――――――――――――――――――――――――――――――――★ダミー
      goto 100
   20 continue
c――――――――――――――――――――――――――――――――――★ダミー
  100 continue
      return
      end
```

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_GI31()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_GI
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_GI31(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———————————————————————————★データの初期設定          ! 7
      M_model_analogy(nnm).i_elastic  =0      !  要素の状態（弾性の場合は０：塑性は１）
      M_model_analogy(nnm).d_eps_x =0.        !  変位
      M_model_analogy(nnm).d_stress_x =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
      nmx=M_model_analogy(nnm).n_type
      nm_type=E_model_analogy(nn).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
      iep=0
      call Analogy_Model_G31(it,iep,ak,Member,Element,                    ! 10
     *      E_model,E_model_analogy,M_model,M_model_analogy,nnm)
      return
      end
C     ―――――――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Analogy_Model_G31
C     ―――――――――――――――――――――――――――――――――――――――
C     ●   接線剛性行列の計算(ok)
C     ―――――――――――――――――――――――――――――――――――――――
      subroutine Analogy_Model_G31(it,iep,ak,Member,Element,
     *      E_model,E_model_analogy,M_model,M_model_analogy,itt)
      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 / M_model31_s         / M_model
      record / M_model_fiber_s     / M_model_analogy
      record / E_model_analogy_s   / E_model_analogy
      dimension E_model_analogy(*),M_model_analogy(*)
      dimension ak(12,12)
c―――――――――――――――――――――――――――――――――★
      if(it.eq.1) then                                                    ! 11
      ra  = E_model_analogy(itt).AK_1             !  ばねの軸剛性
      ray = E_model_analogy(itt+1).AK_1           !  z 軸のばねの曲げ剛性
      raz = E_model_analogy(itt+2).AK_1           !  y 軸のばねの曲げ剛性
      fax = M_model.fax_1                         ! ポテンシャル関数の x 軸微分
      fay = M_model.fay_1                         ! ポテンシャル関数の y 軸微分
      faz = M_model.faz_1                         ! ポテンシャル関数の z 軸微分
      else
      ra  = E_model_analogy(itt).AK_1             !  ばねの軸剛性
      ray = E_model_analogy(itt+1).AK_1           !  z 軸のばねの曲げ剛性
      raz = E_model_analogy(itt+2).AK_1           !  y 軸のばねの曲げ剛性
      fax = M_model.fax_2                         ! ポテンシャル関数の x 軸微分
      fay = M_model.fay_2                         ! ポテンシャル関数の y 軸微分
      faz = M_model.faz_2                         ! ポテンシャル関数の z 軸微分
      endif
      ra_x= E_model_analogy(itt).AK_2             !  ばねの最低軸剛性
      ray_x=E_model_analogy(itt+1).AK_2           !  ばねの最低曲げ剛性
      raz_x=E_model_analogy(itt+2).AK_2           !  ばねの最低曲げ剛性
      call Analogy_GK(iep,ak,ra,ray,raz,fax,fay,faz,                      ! 12
     *               ra_x,ray_x,raz_x)
      return
      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                                          ! 13
c―――――――――――――――――――――――――――――――――――――★線形剛性
      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                                                      ! 14
      a=ra*fax
      b=riy*fay
      c=riz*faz
      h=a*fax+b*fay+c*faz
      if(h.eq.0.) then                                          ! 15
c―――――――――――――――――――――――――――――――――――――★線形剛性
      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                                                      ! 16
c―――――――――――――――――――――――――――――――――――――★非線形剛性
      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                                          ! 17
      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)   =  riyhb
      ak(5,11)  = -riyhb
      ak(11,5)  = -riyhb
      ak(11,11) =  riyhb
      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、ω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ω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。

5.10.4 アナロジーモデルの弾塑性チェック

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(immm),M_model_fiber,
     *      vvx,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     ――――――――――――――――――――――――――――――――
```

```
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                                                  ! 1
      d_epsi_x_1 =  (vv(7)  - vv(1))   !  増分軸方向変位                ! 2
      d_epsi_y_1 =  (vv(11) - vv(5))   !  増分ｙ軸に関する回転角
      d_epsi_z_1 =  (vv(12) - vv(6))   !  増分ｚ軸に関する回転角
      if(Member.analysis_3D .eq. 1)  d_epsi_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_epsi_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_epsi_x_1 =  d_epsi_x_1+M_model.d_epsi_x_1          !  軸方向変位
      M_model.d_epsi_y_1 =  d_epsi_y_1+M_model.d_epsi_y_1          !  ｙ軸に関する回転角
      M_model.d_epsi_z_1 =  d_epsi_z_1+M_model.d_epsi_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                                        ! 3
      nem=E_model.nm_section_1
      nnm=M_model.nm_section_1
      nmx=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_epsi_x_1                      ! 4
      M_model_analogy(nnm+1).d_eps_x =  d_epsi_y_1
      M_model_analogy(nnm+2).d_eps_x =  d_epsi_z_1
      if(N_analysis.le.8.or.
     *   Member.nm_analysis.eq.-1) then                                 ! 5
c―――――――――――――――――――――――――――――――★弾性解析
      else
c―――――――――――――――――――――――――――――――★弾塑性解析
      goto ( 10,20,30),nm_type-10                                       ! 6
  10  continue
c―――――――――――――――――――――――――――――――★　Ｉ型
      call Check_Analogy_M1(E_model_analogy,nem,                        ! 7
     *          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,                    ! 8
     *           M_model_analogy,nnm,
     *           M_model.fax_1,M_model.fay_1,M_model.faz_1,i_hosei)
      goto 100
  20  continue
```

```
c―――――――――――――――――――――――★ II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―――――――――――――――――――――――★ III型
      call Check_Analogy_M3(E_model_analogy,nem,                                 ! 10
     *           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_epsi_x_2 =  (vv(7) - vv(1))    !  軸方向変位
      d_epsi_y_2 =  (vv(11) - vv(5))   !  y 軸に関する回転角
      d_epsi_z_2 =  (vv(12) - vv(6))   !  z 軸に関する回転角
      if(Member.analysis_3D .eq. 1) d_epsi_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_epsi_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_epsi_x_2 =  d_epsi_x_2+M_model.d_epsi_x_2            !  軸方向変位
      M_model.d_epsi_y_2 =  d_epsi_y_2+M_model.d_epsi_y_2             !  y 軸に関する回転角
      M_model.d_epsi_z_2 =  d_epsi_z_2+M_model.d_epsi_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
      nmx=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_epsi_x_2
      M_model_analogy(nnm+1).d_eps_x =  d_epsi_y_2
      M_model_analogy(nnm+2).d_eps_x =  d_epsi_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――――――――――――――――――――――――★ I 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――――――――――――――――――――――――★ II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――――――――――――――――――――――――★ III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Ⅰ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_potentioal_M1 を用いて、求める。
9. ここでは降伏条件の型番号 12 のⅡ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Ⅲ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     ―――――――――――――――――――――――――――――――――
      subroutine Cal_div_potential_M1(E_model_analogy,nem,
     *           M_model_analogy,nmm,ffx,ffy,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――――――――――――――――――――――――★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――――――――――――――――――――――――★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――――――――――――――――――――――――★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)
```

```
      ffz=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 +                                                     ! 17
     *(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
      else                                                          ! 18
      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 +                           ! 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.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、$\bar{N}$=+ Nb、$\bar{N}$=- Nb 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：Nb ＞ $\bar{N}$ ＞ -Nb、1：$\bar{N}$ ＞ Nb、2：-Nb ＞ $\bar{N}$）で見分ける。ここで、Nb 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、$\bar{N}$ ＞ Nb 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、-Nb ＞ $\bar{N}$ 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、Nb ＞ $\bar{N}$ ＞ -Nb 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、$\bar{N}$ ＞ Nb 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か等しい場合、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. ばねが既に塑性状態で、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。

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。

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

Maxwell モデルに関連する項は、上式中で次の 2 項である。

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

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*β 法を適用すれば、振動方程式は式(5.26)として得られる。

$$\begin{aligned}&[M]\{\ddot{y}_{n+1}\}+[\bar{C}]\{\{a\}+\mu_1\{\ddot{y}_{n+1}\}\}+[K]\{\{\bar{b}\}+\mu_2\{\ddot{y}_{n+1}\}\}\\&\quad=-[M][I]\{\ddot{u}_g\}+\{P_S\}-\{Q(y_n)\}-\{\bar{f}_d\}\\&\quad-[K_T(y_n)]\{\{b\}+\mu_2\{\ddot{y}_{n+1}\}\}+[K]\{\{\bar{b}\}+\mu_2\{\ddot{y}_{n+1}\}\}\end{aligned}\qquad\cdots\cdots(5.26)$$

さらに、上式を整理すると、

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

となる。ここで、係数を

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

のように整理すると、方程式(5.27)は次式となる。

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

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。

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

ここで、各係数は、

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

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。

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

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(ie),Newmark_P,
     *           E_model6_real(ien),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                ! α0
c       real*8  gumma_0               ! γ0
c       real*8  alph_d0               ! α_d0
c       real*8  alph_1                ! α1
c       real*8  gumma_1               ! γ1
c       real*8  alph_d1               ! α_d1
c       real*8  alph_2                ! α2
c       real*8  gumma_2               ! γ2
c       real*8  alph_d2               ! α_d2
c       real*8  alph_3                ! α3
c       real*8  gumma_3               ! γ3
c       real*8  alph_d3               ! α_d3
c       real*8  f00                   ! f0
c       real*8  alph                  ! α
c       real*8  gumma                 ! γ
c       real*8  f01                   ! f0'
c       real*8  f0                    ! f0
c       real*8  alfd                  ! α'
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Δt 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              ! Δｔ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                                              ! 1
      b = 2.*Element.C0 + Element.AK*dt
      E_model6_real.alph_0 =  Element.C0*Element.AK*dt/b       !α0
      E_model6_real.gumma_0 = (2.*Element.C0-Element.AK*dt)/b  !γ0
      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        !α1
      E_model6_real.gumma_1 = (2.*Element.C1-Element.AK*dt)/b  !γ1
      E_model6_real.alph_d1 = Element.AK*dt/b                  !α'1
      b = 2.*Element.C2 + Element.AK*dt
      E_model6_real.alph_2 = Element.C2*Element.AK*dt/b        !α2
      E_model6_real.gumma_2 = (2.*Element.C2-Element.AK*dt)/b  !γ2
      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、Δｔ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―――――――――――――――――――――――――――――――★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――――――――――――――――――――――――――――――――★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――――――――――――――――――――――――――――――――★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される。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                                          ! 1
      do j=1,12
      irest = Member(i).irest(j)                                  ! 2
      if(irest.gt.0) then
      u(j)=  a_vector(irest)
      else
      u(j)=0.0
      endif
```

```
      end do
c―――――――――――――――――――――――――――――――★Model_No.6 3 次元制震 Maxwell モデル
      if(Member(i).element_type.eq.6) then                              ! 3
      ien= Member(i).n_model_type
      ie = Member(i).nm_element
      ii=Element(ie).nm_type
      call Cal_nonlin_maxwelldamp(E_model6_real(ien),                   ! 4
     *                            bk,ii)
      call Rotate_K(bk,rot_memb(1,1,1,i),                               ! 5
     *              rot_memb(1,1,2,i),ac_nonlinear)
c―――――――――――――――――――――――――――――――★
      else                                                              ! 6
      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)                                          ! 7
      if(irest.gt.0) then
      sum=0.
      do k=1,12
      sum=sum+ac_nonlinear(j,k)*u(k)                                    ! 8
      enddo
      ld_point(irest)=ld_point(irest) -  sum                            ! 9
      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α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α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されるサブルーチン 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                              ! 1
      do  j=1,12
      irest = Member(i).irest(j)
      if(irest.ne.0) then
      ud(j)=est_vel_point(irest)                                        ! 2
      else
      ud(j)=0.
      endif
      enddo
      call RotateL_v(1,ud,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vpp)      ! 3
c―――――――――――――――――――――――――――――――★部材減衰を持っているか

      goto(11,12,13,14,15,16,17,18,19,20),iet                           ! 4
        •
        •
  16 continue
c―――――――――――――――――――――――――――――――★Model_No.6 3 次元制震 Maxwell モデル
      ii=Element(ie).nm_type
      call Cal_force_maxwelldamp(vpp,E_model6_real(ien),av,ii,i)        ! 5
      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)      ! 6
      do  j=1,12
      irest = Member(i).irest(j)
      if(irest.ne.0) then
      ld_point_repeat(irest)=ld_point_repeat(irest) - vpp(j)            ! 7
```

```
      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 ld_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――――――――――――――――――――――――――――――――★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                                          ! 5
      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――――――――――――――――――――――――――――――★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――――――――――――――――――――――――――――――★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――――――――――――――――――――――――――――――★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 Ceck_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―――――――――――――――――――――――――――――――★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―――――――――――――――――――――――――――――――★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
      udmax = Element.udmax
      istat = E_model6_real.istat                                    ! 1
      ud1= E_model6_real.u1d
      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.udmax ) then                                        ! 5
      ichange = 1
      istat = 1
      elseif(ud1 .le. -udmax) then                                   ! 6
      ichange = 1
      istat = 2
      endif
       goto 9800
 102  continue
      if(ud1 .ge.udmax )goto 9800                                    ! 7
      ichange = 1
      istat = 0
      goto 9800
 103  continue
      if(ud1 .le. -udmax) 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                    ! 12
      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                    ! 13
      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                                                   ! 14
      •
      •
      •
      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 udmax 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 udmax 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 udmax 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 udmax より大きい場合は、2 つのパラメータを ichange=1、istat=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 udmax より大きい場合は、状態変化なし。小さい場合は、ichange=1、istat=0 に設定する。
8. ここは状態が 2 で、速度が逆方向に udmax より大きい場合は、状態変化なし。小さい場合は、ichange=1、istat=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。