9.3 ファイバーの履歴の組み込み方法

9.3.1 履歴の組み込み

本節では、新規のファイバー履歴を SPACE に組み込む方法について解説する。基本的には、せん断モデルの履歴組み込み方法と同じであるが、多少異なるので、その違いを中心に説明する。現在、SPACE に登録されている履歴モデルは、以下のようであり、履歴番号 1-8 までが使用されている。個人用としては、せん断型と同じく 101 を用いることにする。ここでは、新規のモデルを New_model_fiber とする。

１．対称バイリニア型：BiLinear()
２．対称トリリニア型：TriLinear()
３．直線コンクリート履歴モデル：Concrete()
４．曲線コンクリート履歴モデル：Concrete_e()
５．バイリニア型（移動＋等方硬化用）：BiLinear_h()
６．対称トリリニア型（移動＋等方硬化用）：TriLinear_h()
７．非対称バイリニア型：TriLinear_AS()
８．非対称トリリニア型：BiLinear_AS()

9.3.2 モデルの入力仕様

最初に、このファイバーモデルの入力仕様について考えてみよう。ファイバーの履歴モデルは全て特別仕様でデータ入力を行っており、データを保存する構造体が設計されている。

まず、データ入力を行うサブルーチン Fiber_input()の中で、要素データを入力する部分のコードを取り出す。新規履歴モデルに関するコードが太文字で追加されている。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Fiber_input
C     ―――――――――――――――――――――――――――――――――
C     ●   ファイバー要素の入力(ok)
C     ―――――――――――――――――――――――――――――――――
      subroutine Fiber_input(  )
      ierr=0
      if(it.eq.0) then
c――――――――――――――――――――――――――――――――★ファイバーデータの予備入力
      read(5,*,err=999) nm          ! 最大断面数
      write(76,'(a,i4)') ' ファイバー断面総数：',nm
      ii=0
      do i=1,nm
      read(5,*,err=999) n_m,nmm,(ddm(j),j=1,20)     ! 断面番号、エレメント数
c――――――――――――――――――――――――――――――★断面ファイバー数のセット
      do i1=1,n_element
      itype_m = Model_type.no_e_model(Element(i1).element_type)
      if(itype_m.eq.11) then
c――――――――――――――――――――――――――――――★モデル１１                    ! x1
       kk1 = kk1 + 1
        do k=1,2
```

```
          if(Element(i1).n_section(k).eq.n_m) then
           Element(i1).nm_section(k)     = nmm
           if(k.eq.1) then
             E_model11(kk1).n_section_1  = nmm
             E_model11(kk1).nm_section_1 = ii + 1
           endif
           if(k.eq.2) then
             E_model11(kk1).n_section_2  =nmm
             E_model11(kk1).nm_section_2 = ii+1
           endif
           endif
         enddo
       endif
c――――――――――――――――――――――――★モデル１２
       if(itype_m.eq.12) then
       ・
       endif
c――――――――――――――――――――――――★モデル１３
       if(itype_m.eq.13) then
       ・
       endif
c――――――――――――――――――――――――★モデル１５
       if(itype_m.eq.15) then
       ・
       endif
c――――――――――――――――――――――――★モデル２１
       if(itype_m.eq.21) then
       ・
       endif
c――――――――――――――――――――――――★モデル２２
       if(itype_m.eq.22) then
       ・
       endif
c――――――――――――――――――――――――★モデル３１
       if(itype_m.eq.31) then
       ・
       endif
c――――――――――――――――――――――――★モデル３２
       if(itype_m.eq.32) then
       ・
       endif
c――――――――――――――――――――――――★モデル３３
       if(itype_m.eq.33) then
       ・
       endif
       enddo
c――――――――――――――――――――――――★
       do j=1,nmm
       ii = ii + 1
       read(5,*,err=999) n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az                    ! 1
       if(nm_type.le.10) then
       goto(901,902,903,904,905,906,907,908,909,910),nm_type                      ! 2
  901  continue
       goto 900
```

```
902  continue
     read(5,*,err=999) E_3,Q_2                              ! 対称トリリニア型 （スチール用）
     goto 900
903  continue
     read(5,*,err=999) AK_3,AK_4,Q_2,Q_3,Q_4                ! コンクリート型
     goto 900
904  continue
     read(5,*,err=999) AK_4,Q_3,STR_3,STR_7                 ! 直線曲線コンクリート型
     goto 900
905  continue
     read(5,*,err=999) beta                                 ! 等方硬化＋移動硬化バイリニア型
     goto 900
906  continue
     read(5,*,err=999) E_3,Q_2,beta,beta_2                  ! 等方硬化＋移動硬化トリリニア型
     goto 900
907  continue
     read(5,*,err=999) Ec_1,Ec_2,Qc_1,beta                  ! 非対称バイリニア型
     goto 900
908  continue
     read(5,*,err=999) E_3,Q_2,Ec_1,Ec_2,Ec_3,Qc_1,Qc_2,beta,beta_2  ! 非対称トリリニア型
     goto 900
909  continue
     goto 900
910  continue
     goto 900
     elseif(nm_type.le.20) then
     goto(911,912,913,914,915,916,917,918,919,920),nm_type-10
911  continue
     goto 900
912  continue
     goto 900
913  continue
     goto 900
914  continue
     goto 900
915  continue
     goto 900
916  continue
     goto 900
917  continue
     goto 900
918  continue
     goto 900
919  continue
     goto 900
920  continue
     goto 900
     else
     goto(921,922,923,924,925,926,927,928,929,930),nm_type-100          ! 3
921  continue
     read(5,*,err=999) E_3,Q_2                              ! 個人用新規ファイバー履歴
     goto 900
922  continue
     read(5,*,err=999) E_3,Q_2                              ! 個人用新規ファイバー履歴
```

```
      goto 900
 923  continue
      goto 900
 924  continue
      goto 900
 925  continue
      goto 900
 926  continue
      goto 900
 927  continue
      goto 900
 928  continue
      goto 900
 929  continue
      goto 900
 930  continue
      goto 900
      endif
 900  continue
      enddo
      enddo
c―――――――――――――――――――――――★要素ファイバー数セット
      Model_type.nm_div_felement= ii
      write(76,'(a,i5)') ' ファイバー数：',ii
c―――――――――――――――――――――――――――★部材断面ファイバー数のセット
      jj = 0
      do i=1,n_member
      i1   = Member(i).nm_element
      immm = Member(i).n_model_type                ! モデルタイプ別番号
      itype_m = Model_type.no_e_model(Element(i1).element_type)
c―――――――――――――――――――――――★モデル１１                                  ! x2
      if(itype_m.eq.11) then
         k = 1
          M_model11(immm).n_section_1   = Element(i1).nm_section(k)
          M_model11(immm).nm_section_1  = jj + 1
          jj = jj + Element(i1).nm_section(k)
         k  = 2
          M_model11(immm).n_section_2   = Element(i1).nm_section(k)
          M_model11(immm).nm_section_2  = jj + 1
          jj = jj + Element(i1).nm_section(k)
          endif
c―――――――――――――――――――――――★モデル１２
      if(itype_m.eq.12) then
      ・
      endif
c―――――――――――――――――――――――★モデル１３
      if(itype_m.eq.13) then
      ・
      endif
c―――――――――――――――――――――――★モデル１５
      if(itype_m.eq.15) then
      ・
      endif
c―――――――――――――――――――――――★モデル２１
```

```
      if(itype_m.eq.21) then
      ・
      endif
c―――――――――――――――――――――★モデル２２
      if(itype_m.eq.22) then
      ・
      endif
c―――――――――――――――――――――★モデル３１
      if(itype_m.eq.31) then
      ・
      endif
c―――――――――――――――――――――★モデル３２
      if(itype_m.eq.32) then
      ・
      endif
c―――――――――――――――――――――★モデル３３
      if(itype_m.eq.33) then
      ・
      endif
      enddo
      Model_type.nm_div_fmodel =  jj       ! ファイバー要素の最大数
C     ―――――――――――――――――――――――――――――――――
C       ★     ファイバーデータの入力その２
C     ―――――――――――――――――――――――――――――――――
      else
      read(5,*,err=999) nm
      write(76,'(a,i4)') ' Number of sections:',nm
      ii = 0
      do i=1,nm
      read(5,*,err=999) n_m,nmm,(ddm(j),j=1,20)
      do j=1,nmm
      read(5,*,err=999) n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az      ! 標準データ     ! 4
c―――――――――――――――――――――★部材断面ファイバー数セット
      ii = ii + 1
      E_model_fiber(ii).nm_type = nm_type             !  履歴特性番号
      E_model_fiber(ii).E_1     = E_1                 !  ファイバーの第一剛性 E1
      E_model_fiber(ii).E_2     = E_2                 !  ファイバーの第二剛性 E2
      E_model_fiber(ii).Q_1     = Q_1                 !  ファイバーの第一折れ点
      E_model_fiber(ii).G       = G                   !  ファイバー断面積 G
      E_model_fiber(ii).A       = A                   !  ファイバー断面積
      E_model_fiber(ii).Ay      = Ay                  !  ファイバーｙ軸せん断用断面積
      E_model_fiber(ii).Az      = Az                  !  ファイバーｚ軸せん断用断面積
      E_model_fiber(ii).ry      = ry                  !  中立軸から断面中心までのｙ方向距離
      E_model_fiber(ii).rz      = rz                  !  中立軸から断面中心までのｚ方向距離
      if(nm_type.le.10) then
      goto(801,802,803,804,805,806,807,808,809,810),nm_type                        ! 5
 801  continue
c―――――――――――――――――――――★バイリニア型
      write(76,'(2i4,9e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az
      E_model_fiber(ii).E_3     = 0.                  !  ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2     = 0.                  !  ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1    = E_1                 !  ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2    = 0.                  !  ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3    = 0.                  !  ファイバーの圧縮側第三剛性 E3
```

```
      E_model_fiber(ii).Qc_1    = 0.                  !  ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2    = 0.                  !  ファイバーの圧縮側第二折れ点
      goto 800
 802  continue
c―――――――――――――――――――――――★対称トリリニア型
      read(5,*,err=999) E_3,Q_2          ! 対称トリリニア型 等方 硬化＋移動硬化トリリニア型
      write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
     *     E_3,Q_2,beta,beta_2
      E_model_fiber(ii).E_3     = E_3                 !  ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2     = Q_2                 !  ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1    = 0.                  !  ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2    = 0.                  !  ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3    = 0.                  !  ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1    = 0.                  !  ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2    = 0.                  !  ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta    = 0.                  !  移動硬化用パラメータ
      E_model_fiber(ii).Beta_2  = 0.                  !  移動硬化用パラメータその２
      goto 800
 803  continue
c―――――――――――――――――――――――★直線コンクリート型
      read(5,*,err=999) AK_3,AK_4,Q_2,Q_3,Q_4                ! 直線コンクリート型
      write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
     *     AK_3,AK_4,Q_2,Q_3,Q_4
      E_model_fiber(ii).E_3     = AK_3                !  圧縮第三勾配
      E_model_fiber(ii).Q_2     = Q_2                 !  圧縮側第一折れ点の応力
      E_model_fiber(ii).Ec_1    = Q_3                 !  圧縮強度
      E_model_fiber(ii).Ec_2    = Q_4                 !  圧縮流れ点
      E_model_fiber(ii).Ec_3    = AK_4                !  引張第二勾配
      E_model_fiber(ii).Qc_1    = 0.                  !  ダミー
      E_model_fiber(ii).Qc_2    = 0.                  !  ダミー
      goto 800
 804  continue
c―――――――――――――――――――――――★曲線コンクリート型
      read(5,*,err=999) AK_4,Q_3,STR_3,STR_7                 ! 曲線コンクリート型
      write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
     *     AK_4,Q_3,STR_3,STR_7
      E_model_fiber(ii).E_3     = AK_4                !  引張第二勾配
      E_model_fiber(ii).Q_2     = Q_3                 !  圧縮強度
      E_model_fiber(ii).Ec_1    = STR_3               !  最大圧縮応力点におけるひずみ量
      E_model_fiber(ii).Ec_2    = STR_7               !  弾性限界ひずみ量
      E_model_fiber(ii).Ec_3    = 0.                  !  ダミー
      E_model_fiber(ii).Qc_1    = 0.                  !  ダミー
      E_model_fiber(ii).Qc_2    = 0.                  !  ダミー
      goto 800
 805  continue
c―――――――――――――――――――――――★等方硬化＋移動硬化バイリニア型
      read(5,*,err=999) beta
       write(76,'(2i4,10e12.4)') n,nm_type,A,ry,rz,E_1,E_2,
     *                     Q_1,G,Ay,Az,beta   !等方硬化＋移動硬化バイリニア型
      E_model_fiber(ii).E_3     = 0.                  !  ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2     = 0.                  !  ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1    = E_1                 !  ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2    = 0.                  !  ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3    = 0.                  !  ファイバーの圧縮側第三剛性 E3
```

```
      E_model_fiber(ii).Qc_1    = 0.                  !  ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2    = 0.                  !  ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta    = beta                !  移動硬化用パラメータ
      goto 800
  806 continue
c-----------------------★等方硬化＋移動硬化トリリニア型
      read(5,*,err=999) E_3,Q_2,beta,beta_2      !
      write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
     *     E_3,Q_2,beta,beta_2
      E_model_fiber(ii).E_3     = E_3                 !  ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2     = Q_2                 !  ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1    = 0                   !  ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2    = 0                   !  ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3    = 0                   !  ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1    = 0                   !  ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2    = 0                   !  ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta    = beta                !  移動硬化用パラメータ
      E_model_fiber(ii).Beta_2  = beta_2              !  移動硬化用パラメータ
      goto 800
  807 continue
c-----------------------★非対称バイリニア型
      read(5,*,err=999) Ec_1,Ec_2,Qc_1,beta
       write(76,'(2i4,15e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
     *              Ec_1,Ec_2,Qc_1,beta              !等方硬化＋移動硬化バイリニア型
      E_model_fiber(ii).E_3     = 0.                  !  ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2     = 0.                  !  ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1    = Ec_1                !  ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2    = Ec_2                !  ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3    = 0.                  !  ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1    = Qc_1                !  ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2    = 0.                  !  ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta    = beta                !  移動硬化用パラメータ
      goto 800
  808 continue
c-----------------------★非対称トリリニア型
      read(5,*,err=999) E_3,Q_2,Ec_1,Ec_2,Ec_3,Qc_1,Qc_2,beta,beta_2
      E_model_fiber(ii).E_3     = E_3                 !  ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2     = Q_2                 !  ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1    = Ec_1                !  ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2    = Ec_2                !  ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3    = Ec_3                !  ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1    = Qc_1                !  ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2    = Qc_2                !  ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta    = beta                !  移動硬化用パラメータ
      E_model_fiber(ii).Beta_2  = beta_2              !  移動硬化用パラメータ
      goto 800
  809 continue
      goto 800
  810 continue
      goto 800
      elseif(nm_type.le.20) then
      goto(811,812,813,814,815,816,817,818,819,820),nm_type-20
  811 continue
      goto 800
```

```
 812  continue
      goto 800
 813  continue
      goto 800
 814  continue
      goto 800
 815  continue
      goto 800
 816  continue
      goto 800
 817  continue
      goto 800
 818  continue
      goto 800
 819  continue
      goto 800
 820  continue
      goto 800
      else
      goto(821,822,823,824,825,826,827,828,829,830),nm_type-100                    ! 6
 821  continue
      read(5,*,err=999) E_3,Q_2                              ! 個人用新規ファイバー履歴
      goto 800
 822  continue
      read(5,*,err=999) E_3,Q_2                              ! 個人用新規ファイバー履歴
      goto 800
 823  continue
      goto 800
 824  continue
      goto 800
 825  continue
      goto 800
 826  continue
      goto 800
 827  continue
      goto 800
 828  continue
      goto 800
 829  continue
      goto 800
 830  continue
      goto 800
      endif
 800  continue
      enddo
c――――――――――――――――――――――――――――★ファイバーモデル剛性出力
      call Fiber_output(E_model_fiber(ii-nmm+1),nmm)
      enddo
c――――――――――――――――――――――――――――★ファイバー履歴特性数セット
      n_m_bilinear    = 0
      n_m_trilinear   = 0
      n_m_concrete    = 0
      n_m_analogy     = 0
      do i=1,n_member
```

```
      ie  = Member(i).nm_element
      imm = Element(ie).n_element
      im  = Member(i).n_model_type
c-----------------------★モデル１１                                    ! x3
      itype_m = Model_type.no_e_model(Element(ie).element_type)
      if(itype_m.eq.11) then
          ii  = E_model11(imm).n_section_1
          nmm = E_model11(imm).nm_section_1 - 1
          nnmm=M_model11(im).nm_section_1 - 1
          do j=1,ii
            nmm = nmm  + 1
           nnmm= nnmm + 1
          if(E_model_fiber(nmm).nm_type.eq.1.or.
     *       E_model_fiber(nmm).nm_type.eq.5.or.
     *       E_model_fiber(nmm).nm_type.eq.7) then
             n_m_bilinear  = n_m_bilinear + 1
             M_model_fiber(nnmm).n_type = n_m_bilinear
          elseif(E_model_fiber(nmm).nm_type.eq.2.or.
     *       E_model_fiber(nmm).nm_type.eq.6.or.
     *       E_model_fiber(nmm).nm_type.eq.8
     *       .or._model_fiber(nmm).nm_type.eq.101) then  ! 新規履歴モデルのための追加コード
             n_m_trilinear = n_m_trilinear + 1
             M_model_fiber(nnmm).n_type = n_m_trilinear
          elseif(E_model_fiber(nmm).nm_type.eq.3.or.
     *       E_model_fiber(nmm).nm_type.eq.4) then
             n_m_concrete  = n_m_concrete + 1
             M_model_fiber(nnmm).n_type = n_m_concrete
          endif
         enddo
         ii  = E_model11(imm).n_section_2
         nmm = E_model11(imm).nm_section_2 - 1
         nnmm= M_model11(im).nm_section_2 - 1
         do j=1,ii
          nmm  = nmm  + 1
          nnmm = nnmm + 1
           if(E_model_fiber(nmm).nm_type.eq.1.or.
     *       E_model_fiber(nmm).nm_type.eq.5.or.
     *       E_model_fiber(nmm).nm_type.eq.7) then
            n_m_bilinear = n_m_bilinear + 1
            M_model_fiber(nnmm).n_type = n_m_bilinear
           elseif(E_model_fiber(nmm).nm_type.eq.2.or.
     *       E_model_fiber(nmm).nm_type.eq.6.or.
     *       E_model_fiber(nmm).nm_type.eq.8
     *       .or._model_fiber(nmm).nm_type.eq.101) then  ! 新規履歴モデルのための追加コード
            n_m_trilinear = n_m_trilinear + 1
            M_model_fiber(nnmm).n_type = n_m_trilinear
          elseif(E_model_fiber(nmm).nm_type.eq.3.or.
     *       E_model_fiber(nmm).nm_type.eq.4) then
            n_m_concrete  = n_m_concrete + 1
            M_model_fiber(nnmm).n_type = n_m_concrete
           endif
         enddo
       endif
c-----------------------★モデル１２
```

```
       if(itype_m.eq.12) then
       ・
       endif
c――――――――――――――――――――★モデル１３
      if(itype_m.eq.13) then
       ・
      endif
c――――――――――――――――――――★モデル１５
      if(itype_m.eq.15) then
       ・
       endif
c――――――――――――――――――――★モデル２１
       if(itype_m.eq.21) then
       ・
       endif
c――――――――――――――――――――★モデル２２
       if(itype_m.eq.22) then
       ・
       endif
c――――――――――――――――――――★モデル３１
       if(itype_m.eq.31) then
       ・
       endif
c――――――――――――――――――――★モデル３２
       if(itype_m.eq.32) then
       ・
       endif
c――――――――――――――――――――★モデル３３
        if(itype_m.eq.33) then
       ・
        endif
c――――――――――――――――――――★
      enddo
      Model_type.n_m_bilinear   = n_m_bilinear                    ! x4
      Model_type.n_m_trilinear  = n_m_trilinear
      Model_type.n_m_concrete   = n_m_concrete
      Model_type.n_m_analogy    = n_m_analogy
      write(76,'(a,i8)') ' 履歴 NO.1:',n_m_bilinear
      write(76,'(a,i8)') ' 履歴 NO.2:',n_m_trilinear
      write(76,'(a,i8)') ' 履歴 NO.3:',n_m_concrete
      write(76,'(a,i8)') ' アナロジーモデル:',n_m_analogy
      endif
      return
 999  continue
      ierr=1
      return
       end
```

1. サブルーチン Fiber_input()は､対応するファイルを2度読みする。
   従って、2 箇所にデータを入力するコードが存在するので、同時に
   変更しなければならない。また、このファイバーデータの仕様を変
   更する場合は、静的解析と静的・動的プレゼンターの入力用プログ

ラムも変更しなければならない。まず、ここでは、ファイバーモデルの標準入力の第 1 レコードを入力する。変数 nm_type は、ファイバーの履歴特性番号を表す。ファイバーの履歴は階層構造を構成している。

2．ファイバーの履歴特性番号毎に入力仕様が異なるため、階層構造に従って処理を分類し、入力用コードを別個に記述する。

3．個人用新規ファイバー履歴の第 2 レコードを設計し、履歴特性番号 101 によりこの位置入力用コードを追加する。

4．2 度目のファイル読み込みを行う部分であり、標準第１レコードを読み込む。

5．ファイバーの履歴特性番号毎に入力仕様が異なるため、階層構造に従って処理を分類し、入力用コードを別個に記述する。

6．個人用新規ファイバー履歴の第 2 レコードを設計し、履歴特性番号 101 によりこの位置に入力用コードを記述する。つまり、新規ファイバー用履歴特性の入力データを設定する場合、標準仕様の第 1 レコードと残りのデータを第 2 レコードとして、ここと 3.の部分で入力するコードを追加する。

x1. ここでは、現在設計されている部材モデルのどの位置で、このファイバー断面が使用されているかをチェックする。もし使用している場合は、要素構造体にファイバー断面番号をセットする。以下の処理では､部材モデル 12､13 などファイバー断面に関連するモデルは、全てここでファイバー断面とリンクを取らなければならない。詳細は、次節の部材モデルの新規設計で説明する。

x2. 上記と同じく、このファイバー断面がどの部材モデルで使用されているかチェックする。ただし、上記は全要素についてであるが、ここでは全部材についてチェックする。

x3. ここでは、断面内で使用しているファイバーの履歴を解析するために必要となるワーク領域構造体の数を数えている。ここでは、両端ファイバーモデルについてであり、プログラムコードを見ると、i 端と j 端の 2 箇所でこの構造体の数を数えている。他の部材モデルでも同様の検索を行っていることは当然のことである。現在、このワーク領域構造体は、3 種類用意されており、その種類に対応する解析モデル中に存在するファイバー数分必要となる。その 3 種類とは、

　１．バイリニア型構造体：Bilinear_work(:)

　２．トリリニア型構造体：Trilinear_work(:)

３．コンクリート型構造体：Concrete_work(:)

である。ここで、チェックを行っているコードの順番は、上記のバイリニア型、トリリニア型、コンクリート型の順であり、その中の番号は履歴番号である。

新規の履歴モデルが追加される場合で、上記の３つのワーク領域を使用する場合は、コメントに示したコードを追加することになる。なお、追加する履歴特性の番号は 101 であり、使用するワーク領域はトリリニア型構造体とした。新規の履歴モデルが新しいワーク領域構造体を必要とする場合は、その構造体を新たに設計し、さらに、その数をここで数える必要があり、そのコードを追加することになる。もちろん、他の部材モデルでこの新規履歴モデルを組み込む場合は、該当する部分に同様の追加コードを加える必要がある。

x4. 最後に、各種の部材モデルで必要となる３つワーク領域構造体の数が、構造体成分 model_type.n_m_bilinear などにセットされる。当然、新しいワーク領域構造体を設定した場合は、ここでその個数をセットする必要があり、そのコードを追加する。

9.3.3 モデルの出力仕様

本節では、この新しい履歴モデルに関する出力仕様について説明する。出力データは他のモジュールで使用されており、変更する場合は、他のシステム、例えば、動的プレゼンターなどをチェックし、整合性を取る必要がある。

まずは、通常の部材モデルにおける標準出力仕様を見てみよう。解析結果を出力するサブルーチン Out_stress()は、submain_dynamic_a()の中でコールされている。ここでは、両端ファイバーモデルに関連する部分のみ示す。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Out_stress
C       ―――――――――――――――――――――――――――――――――
C     ●   部材両端と中央の応力を出力(ok)
C     ―――――――――――――――――――――――――――――――――
      subroutine Out_stress(Member,Element,E_model6_real,M_model11,
     *                      M_model12,M_model13,M_model15,M_model21,
     *                      M_model22,M_model31,M_model32,M_model33,
     *                      n_member,ifl,iflz,i_print,Out_section)
C     ―――――――――――――――――――――――――――――――――
      implicit real*8(A-H,O-Z)
      include "submain.h"
```

```
      include "submainx.h"
      record / Member_s         / Member
      record / Element_s        / Element
      record / Out_section_s    / Out_section
      record / E_model6_real_s  / E_model6_real
      ・
      ・
      do i=1,n_member
      if(Member(i).element_type.eq.6) then
c――――――――――――――――――――――――――――――★Maxwell モデル
      ・
      ・
      elseif(Member(i).element_type.eq.2) then
c――――――――――――――――――――――――――――――★3 次元せん断弾塑性モデル
      ・
      ・
      elseif(Member(i).element_type.eq.3) then
c――――――――――――――――――――――――――――――★3 次元軸力弾塑性モデル
      ・
      ・
      else
  c――――――――――――――――――――――――――――――★静的縮合モデル（ファイバーモデルなど）
      i_t=Member(i).element_type
      im=mxtype(i_t)
      ns=myytype(im)
      ie = Member(i).nm_element
      immm= Member(i).n_model_type              ! モデルタイプ別番号
      do j=1,2
      istat = Member(i).d_stat(j)
      jj=6*(j-1)
      v(1)=Member(i).stress(jj+1)          ! 軸力
      v(2)=Member(i).stress(jj+5)          ! ｙ軸モーメント
      v(3)=-Member(i).stress(jj+6)         ! ｚ軸モーメント
      rrxx=0.                              ! 現在ダミー　塑性関数値
      if(Element(ie).ANP.ne.0.)
     *      rrxx=(Member(i).stress(jj+1)/Element(ie).ANP)**2
      rrx=0.
      if(Element(ie).AMPY.ne.0.)
     *      rrx=(Member(i).stress(jj+5)/Element(ie).AMPY)**2
      if(Element(ie).AMPZ.ne.0.)
     *      rrx=rrx+(Member(i).stress(jj+6)/Element(ie).AMPZ)**2
      v(4)=rrxx+Dsqrt(rrx)
      write(iflz(5)) istat,(v(k),k=1,4)
      enddo
      if(ns.gt.2) then
      do j=3,ns
         istat = Member(i).d_stat(j)
      jj=6*(j-1)
      v(1)=Member(i).stress(jj+1)          ! 軸力
      v(2)=Member(i).stress(jj+5)          ! ｙ軸モーメント
      v(3)=-Member(i).stress(jj+6)         ! ｚ軸モーメント
      rrxx=0.                              ! 現在ダミー　塑性関数値
      if(Element(ie).ANP.ne.0.)
     *      rrxx=(Member(i).stress(jj+1)/Element(ie).ANP)**2
```

```
      rrx=0.
      if(Element(ie).AMPY.ne.0.)
     *     rrx=(Member(i).stress(jj+5)/Element(ie).AMPY)**2
      if(Element(ie).AMPZ.ne.0.)
     *     rrx=rrx+(Member(i).stress(jj+6)/Element(ie).AMPZ)**2
      v(4)=rrxx+Dsqrt(rrx)
      write(iflz(5)) istat,(v(k),k=1,4)
      endif
      enddo
      return
      end
```

このサブルーチンの該当する部分をみると、部材断面の応力が構造体成分 Member().stress()にセットされていれば、ファイバーデータに全く関係しない。したがって、ここでは、新規ファイバー履歴モデルを追加したとしても変更することはない。

次に、断面のファイバー応力とひずみの出力に関連する２つのサブルーチンを検討しよう。まず出力部材の断面をチェックし、断面ファイバーデータのヘッダー部分を出力するサブルーチン out_section_check() を見てみよう。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /out_section_check
C     ―――――――――――――――――――――――――――――――――
C     ●   出力部材の断面応力をチェック
C     ―――――――――――――――――――――――――――――――――
      subroutine out_section_check(Member,Element,
     *               n_member,No_section,Out_section,
     *               ifl,iflz,i_print)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / Member_s  / Member
      record / Element_s  / Element
      record / Out_section_s  / Out_section
      dimension Member(*),Element(*),No_section(*)
      dimension ifl(16),iflz(16)
C     ―――――――――――――――――――――――――――――――――
C     ●   out_section_s 構造体
C     ―――――――――――――――――――――――――――――――――
c     解析パラメータ
c       structure  / out_section_s/
c       integer    n_member        !  出力部材数
c       integer    n_section       !  出力断面数
c       integer    no_member(10)   !  部材番号
c       integer    no_section(11)  !  断面数
c       integer    no_fiber(30)    !  ファイバー数
c       end structure
c       record /out_section_s/ Out_section
      if(ifl(6).eq.0) return
      nn=No_section(1)                          ! 出力断面数                    ! 1
```

```
      if(nn.gt.0) then
      iix=0
      iiy=0
      do i=1,nn                                                    ! 2
      n=No_section(i+1)
      if(n.ge.1.and.n.le.n_member)then                             ! 3
      im=Member(n).element_type
      ie= Member(n).nm_element
      if(im.eq.11.or.im.eq.13) then                                ! 4
      iix=iix+1
      iiy=iiy+2
      Out_section.no_member(iix) = n
      Out_section.no_section(iix)= 2
      Out_section.no_fiber(iiy-1)= Element(ie).nm_section(1)
      Out_section.no_fiber(iiy)  = Element(ie).nm_section(2)
      Out_section.nm_fiber(iiy-1)= Element(ie).n_section(1)
      Out_section.nm_fiber(iiy)  = Element(ie).n_section(2)
      elseif(im.eq.18) then                                        ! 5
      iix=iix+1
      iiy=iiy+1
      Out_section.no_member(iix) = n
      Out_section.no_section(iix)= 1
      Out_section.no_fiber(iiy)= Element(ie).nm_section(1)
      Out_section.nm_fiber(iiy)= Element(ie).n_section(1)
      elseif(im.eq.12.or.im.eq.14) then                            ! 6
      iix=iix+1
      iiy=iiy+3
      Out_section.no_member(iix) = n
      Out_section.no_section(iix)= 3
      Out_section.no_fiber(iiy-2)= Element(ie).nm_section(1)
      Out_section.no_fiber(iiy-1)= Element(ie).nm_section(2)
      Out_section.no_fiber(iiy)  = Element(ie).nm_section(3)
      Out_section.nm_fiber(iiy-2)= Element(ie).n_section(1)
      Out_section.nm_fiber(iiy-1)= Element(ie).n_section(2)
      Out_section.nm_fiber(iiy)  = Element(ie).n_section(3)
      endif
      endif
      enddo
      Out_section.n_member = iix
      Out_section.n_section = iiy
      else
      Out_section.n_member = 0
      endif
c-----------------------------★断面応力ヘッダー出力
      write(iflz(6)) Out_section.n_member                          ! 7
      if(Out_section.n_member.eq.0) return
      write(iflz(6))(Out_section.no_member(j),j=1,Out_section.n_member)
      write(iflz(6))(Out_section.no_section(j),j=1,Out_section.n_member)
      write(iflz(6)) Out_section.n_section
      write(iflz(6))(Out_section.nm_fiber(j),j=1,Out_section.n_section)
      write(iflz(6))(Out_section.no_fiber(j),j=1,Out_section.n_section)
c-----------------------------★断面応力ヘッダー出力終了
      write(76,'(a,i4)') ' No. member: ',Out_section.n_member
      write(76,'(a,i4)') ' No. section: ',Out_section.n_section
```

```
      if(Out_section.n_member.eq.0) return
      return
      end
```

1. ユーザーが出力を要求している部材数を取得する。
2. 出力部材数分、以下の処理を行う。
3. 出力要求を行っている部材番号が解析モデルの部材の範囲かどうかチェックする。範囲外の場合はその部材を無視する。
4. 部材モデルのコード番号が 11 と 13 の場合は、両端ファイバー断面もしくは両端 MS 断面であり、その管理番号をファイバー断面出力構造体 Out_section にセットする。
5. 部材モデルのコード番号が 18 の場合は、両端ピンで中央ファイバー断面を有するモデルであり、その管理番号をファイバー断面出力構造体 Out_section にセットする。
6. 部材モデルのコード番号が 12 と 14 の場合は、両端・中央ファイバー断面もしくは両端・中央 MS 断面であり、その管理番号をファイバー断面出力構造体 Out_section にセットする。
7. 断面ファイバー応力出力ファイルのヘッダー部分を出力する。

次に、実際の断面内のファイバー応力とひずみを出力するサブルーチン Out_Fiber()を見てみよう。

```
C     ―――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Out_Fiber
C       ―――――――――――――――――――――――――――――――
C     ●   部材の断面応力を出力(ok)
C     ―――――――――――――――――――――――――――――――
      subroutine Out_Fiber( )
      real*4    v(100),vf(3)
c     i_print         :integer 出力制御変数 0:ファイル出力あり
c――――――――――――――――――――――――――――★応力 5
      if(i_print.ne.0) return
      if(ifl(6).eq.0) return
c――――――――――――――――――――――――――――★断面応力出力
      if(Out_section.n_member.eq.0) return
      do j=1,Out_section.n_member                                    ! 1
      i=Out_section.no_member(j)                                     ! 2
c――――――――――――――――――――――――――――★断面応力
      ie = Member(i).nm_element
      imm= Element(ie).n_element                  ! 要素番号
      immm= Member(i).n_model_type                ! モデルタイプ別番号
      iet  = Member(i).element_type
      if(iet.eq.11) then                                             ! 3
c――――――――――――――――――――――――★モデル１１
      nm_div=E_model11(imm).n_section_1
      nnm=M_model11(immm).nm_section_1 - 1
```

```
      do k=1,nm_div
      v(k)=M_model_fiber(k+nnm).d_stress_x
      enddo
      write(iflz(6)) (v(k),k=1,nm_div)
      do k=1,nm_div
      v(k)=M_model_fiber(k+nnm).d_eps_x
      enddo
      vf(1) = M_model11(immm).d_epsi_x_1          !  軸方向歪
      vf(2) = M_model11(immm).d_epsi_y_1          !  y軸に関する曲げ歪
      vf(3) = M_model11(immm).d_epsi_z_1          !  z軸に関する曲げ歪
      write(iflz(6))(vf(k),k=1,3),(v(k),k=1,nm_div)
      nm_div=E_model11(imm).n_section_2
      nnm=M_model11(immm).nm_section_2 - 1
      do k=1,nm_div
      v(k)=M_model_fiber(k+nnm).d_stress_x
      enddo
      write(iflz(6)) (v(k),k=1,nm_div)
      do k=1,nm_div
      v(k)=M_model_fiber(k+nnm).d_eps_x
      enddo
      vf(1) = M_model11(immm).d_epsi_x_2          !  軸方向歪
      vf(2) = M_model11(immm).d_epsi_y_2          !  y軸に関する曲げ歪
      vf(3) = M_model11(immm).d_epsi_z_2          !  z軸に関する曲げ歪
      write(iflz(6))(vf(k),k=1,3),(v(k),k=1,nm_div)
      elseif(iet.eq.12) then
c―――――――――――――――――――――――★モデル１２
      .
      .
      elseif(iet.eq.15) then
c―――――――――――――――――――――――★モデル１５
      .
      .
      elseif(iet.eq.13) then
c―――――――――――――――――――――――★モデル２１
      .
      .
      elseif(iet.eq.14) then
c―――――――――――――――――――――――★モデル２２
      .
      .
      elseif(iet.eq.16) then
c―――――――――――――――――――――――★モデル３１
      .
      .
      elseif(iet.eq.17) then
c―――――――――――――――――――――――★モデル３２
      .
      .
      elseif(iet.eq.18.or.iet.eq.19) then
c―――――――――――――――――――――――★モデル１３
      .
      .
      elseif(iet.eq.101.or.iet.eq.102) then                    ! 4
c―――――――――――――――――――――――★個人用新規部材モデル
```

```
      endif
c――――――――――――――――――――――――――――――★断面応力出力終わり
      enddo
      return
      end
```

1. ユーザーが出力を要求している部材数で以下の出力処理を行う。
2. 出力部材番号を取得する。
3. 両体ファイバーの部材モデル 11 について出力処理を行う。ここのコードを見れば理解できるように、履歴特性に関連するコードはない。そのため、新規ファイバー履歴特性を追加しても、このサブルーチンは変更する必要はない。
4. ここでは、次節以降で説明するファイバー断面を有する部材モデルを新規に作成する場合、この部分に出力コードを追加する必要がある。

9.3.4 構造体の定義

本節では、新規のファイバー用履歴モデルによって、新たな構造体を作成する必要がある場合について説明する。現在、ファイバーを含む部材モデルは全て静的縮合モデルであり、弾性はりエレメントとファイバーによって弾塑性状態を表すエレメント部分などによって構成される。弾性はりに関する情報の保存とワーク領域として、標準型の Element_s、Member_s 構造体が用いられている。ファイバーが集まった断面に関するワーク領域としては M_model_fiber_s が使用され、さらに、ファイバーに関する情報の確保とワーク領域として、構造体 E_model_fiber_s と Bilinear_work、Trilinear_work、Concrete_work が使用されている。なお、部材の弾塑性状態を表すエレメントである MS モデルとアナロジーモデルに関する構造体も、ファイバーモデルと同じ構造体を兼用している。

さて、新たにファイバーモデルの履歴特性を設計する場合であるが、標準入力仕様で、しかも解析中データが変化しないファイバー要素構造体 E_model_fiber_s（鉄骨及び鉄筋用）、及び E_model_fiberc_s（コンクリート用）と解析中データが変化するファイバー部材構造体 M_model_fiber_s の内容で十分な場合は、新たに構造体を設計する必要はない。しかし、入力データ数が多数で現在の仕様では不足する場合や、ワーク領域が不足する場合は、新たに構造体を設計する必要が発生する。

これに伴って、既存のコードを変更しなければならなくなる。これについては、後で解説する。しかし、標準仕様の構造体の内容を変更することで使用可能であれば、登録手続きは単純である。

まずは、標準仕様であるファイバー要素構造体 E_model_fiber_s と E_model_fiberc_s を示そう。両者の並びで対応する変数は同じ記憶領域を占めており、構造体配列の数はファイバー要素数分設定される。

```
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     ―――――――――――――――――――――――――――――――――
      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_fiberc_s 構造体（コンクリート用）
C     ―――――――――――――――――――――――――――――――――
      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
```

上の構造体 E_model_fiber_s をそのまま利用することも可能であるが、新規の履歴モデル用として、新たに設計することもできる。上に示した構造体 E_model_fiber_s を参照して作成すれば良い。この構造体は、ファイバー要素データを保存するためのものであり、部材毎に計算途中で変化するデータを入れるものではない。これについては、ファイバー部材構造体 M_model_fiber_s が用意されている。また、この構造体の全ての成分が使用可能となっておらず、上記の M_model_fiber_s の中で濃い文字で示したエレメントはシステムが使用する。

次に、ファイバー部材構造体 M_model_fiber_s を示そう。ここでも、新たな構造体を設計したい場合は、この構造体 M_model_fiber_s を参照して作成されたい。

```
C     ―――――――――――――――――――――――――――――――
C     ●   ファイバーモデル M_model_fiber_s 構造体
C     ―――――――――――――――――――――――――――――――・
      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
```

さらに、ファイバーモデルでは、ワーク用として履歴特性に合わせて３つの構造体が設計され、使用されている。この３つの構造体は、現在以下の８つの履歴特性で用いられている。

１．バイリニア型ワーク領域構造体：Bilinear_work_s
　　１．対称バイリニア型：**BiLinear()**
　　５．バイリニア型（移動＋等方硬化用）：**BiLinear_h()**
　　７．非対称バイリニア型：**TriLinear_AS()**

２．トリリニア型ワーク領域構造体：Trilinear_work_s
　　２．対称トリリニア型：**TriLinear()**
　　６．対称トリリニア型（移動＋等方硬化用）：**TriLinear_h()**
　　８．非対称トリリニア型：**BiLinear_AS()**

３．コンクリート型ワーク領域構造体： Concrete_work_s
　　３．直線コンクリート履歴モデル：**Concrete()**
　　４．曲線コンクリート履歴モデル：**Concrete_e()**

上記３つのワーク領域構造体を以下に示そう。

```
C     ―――――――――――――――――――――――――――――――――
C     ●　履歴モデル　Bilinear：ファイバー要素 用　Work エリア構造体
C     ―――――――――――――――――――――――――――――――――
c
c     部材
      structure / Bilinear_work_s/
      integer  i_stat              !  ファイバー要素の状態
      real*8   P1                  !  ワーク領域
      real*8   P2                  !  ワーク領域
      real*8   Sig_z               !  ワーク領域
      end structure
c       record / Bilinear_work_s    / Bilinear_work
c
C     ―――――――――――――――――――――――――――――――――
C     ●　履歴モデル　Trilinear：ファイバー要素 用　Work エリア構造体
C     ―――――――――――――――――――――――――――――――――
c
c     部材
      structure / Trilinear_work_s/
      integer  i_stat              !  ファイバー要素の状態
      real*8   P1(5)               !  ワーク領域
      end structure
c       record / Trilinear_work_s    / Trilinear_work
c
C     ―――――――――――――――――――――――――――――――――
C     ●　履歴モデル　Concrete：ファイバー要素 用　Work エリア構造体
C     ―――――――――――――――――――――――――――――――――
c
c     部材
      structure / Concrete_work_s/
      integer  i_stat              !  ファイバー要素の状態
      integer  ipret               !  過去の引張履歴
      integer  ipre_c              !  過去の圧縮履歴
      real*8   P1(6)               !  ワーク領域
      end structure
c       record / Concrete_work_s    / Concrete_work
```

標準仕様の構造体を使用する場合は、新たな構造体を必要としないが、その新規モデルに適合した構造体を設計する場合は、その構造体を新たなヘッダーファイルに作成する必要がある。このヘッダーファイルを New_submain.h としよう。読者はまずこの New_submain.h ファイルを作り、この中に新たな構造体を書き込むことになる。新たなヘッダーファイルは、この構造体を使用する場合に、そのサブルーチンの中にインクルードされる。

新規にファイバー要素の履歴特性を設計する際、入力データやワーク領域が先に示した構造体では対応できない場合、新たな構造体を設計し、使用することになる。その際、多くの手続きが必要となり、既存のプログラムに手続きのためのコードを追加する必要が生じる。その手続きを知るためにも、現在の構造体がどのような手続きで利用可能となっているか調査しよう。

現在、ファイバー履歴に関連する構造体として、ファイバー断面に関する構造体 2 種類、ファイバーの履歴に関する構造体 3 種類が用いられている。これらの構造体は、submain_dynamic_a()サブルーチンの中で、構造体の手続きにしたがって、宣言、動的確保、領域解放が行われている。その手続きの幾つかを以下に示す。

```
c―――――――――――――――――――――――――――――――★ファイバーモデル用エリア
      record / E_model_fiber_s   / E_model_fiber
      record / M_model_fiber_s   / M_model_fiber
c―――――――――――――――――――――――――――――――★履歴モデル用エリア
      record / Bilinear_work_s    / Bilinear_work
      record / Trilinear_work_s   / Trilinear_work
      record / Concrete_work_s    / Concrete_work
c―――――――――――――――――――――――――――――――★ファイバーモデル用エリア
      ALLOCATABLE :: E_model_fiber(:)
      ALLOCATABLE :: M_model_fiber(:)
c―――――――――――――――――――――――――――――――★履歴モデル用エリア
      ALLOCATABLE :: Bilinear_work(:)
      ALLOCATABLE :: Trilinear_work(:)
      ALLOCATABLE :: Concrete_work(:)
```

これらの構造体は、次に示すように解析モデルで使用する数に従って、動的に領域を確保されなければならない。当然、この構造体の数をどこかで数えておく必要がある。

```
c―――――――――――――――――――――――――――★ファイバーモデル領域セット
      n= Model_type.nm_div_fmodel          !部材内のファイバー要素の数
      if(n.ne.0) then
      ALLOCATE (M_model_fiber(n))
      endif
      n= Model_type.nm_div_felement        !ファイバー要素のエレメント最大数
```

```
      if(n.ne.0) then
      ALLOCATE (E_model_fiber(n))
      endif
c―――――――――――――――――――――――――――――――★ファイバー用履歴要素
      n= Model_type.n_m_bilinear          !  バイリニアの履歴要素の数
      if(n.ne.0) then
      ALLOCATE (Bilinear_work(n))
      endif
      n= Model_type.n_m_trilinear         !  トリリニアの履歴要素の数
      if(n.ne.0) then
      ALLOCATE (Trilinear_work(n))
      endif
      n= Model_type.n_m_Concrete          !  コンクリートの履歴要素の数
      if(n.ne.0) then
      ALLOCATE (Concrete_work(n))
      Endif
```

これらの構造体は動的に確保する配列であることから、サブルーチンsubmain_dynamic_a()のなかで確保する必要がある。手続きとしては、動的確保の他に、動的領域であることの宣言、使用後の領域解放を行うことになる。構造体を新たに設計、使用する場合は、これらの処理を必ず既存プログラムの中に書き込む必要が生じる。

9.3.5 必要となるサブルーチン

次に、この履歴特性New_model_fiberをSPACEに組み込むために、必要となるサブルーチンについて説明する。この必要となるサブルーチンは

１）線形剛性を求めるサブルーチン
２）非線形剛性を求めるサブルーチン
３）応力を求めるサブルーチン
４）応力をチェックし、接線剛性を求めるサブルーチン

である。ただし、この中で標準の静的縮合モデルに関連するサブルーチンを転用できる場合は、そのサブルーチンを使用すれば良い。まず、上記４つの標準的な静的縮合モデルに関するサブルーチンを検討しよう。例として、両端ファイバー部材モデルを取り上げる。

最初は、線形剛性行列を求めるサブルーチンFiber_Model_GI11()であり、次に示す。

```
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,New_fiber_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
      record / New_fiber_work_s   / New_fiber_work
      dimension E_model_fiber(*),M_model_fiber(*)
      dimension Bilinear_work(*),Trilinear_work(*)
      dimension Concrete_work(*),New_fiber_work(*)
      dimension ak(12,12)
c―――――――――――――――――――――――――――――――――――――――★
      if(it.eq.1) then
      do i=1,30
      M_model.ff_ip(i) = 0.
      enddo
      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
      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
      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
      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
```

```
      if(nm_type.le.10) then
      goto ( 10,20,30,30,10,20,10,20),nm_type
   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
      goto 100
   20 continue
c――――――――――――――――――――――――★非対称トリリニア型
      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
      Concrete_work(nmx).i_stat  = -1              !  ファイバー要素の状態
      goto 100
      else
      goto ( 110,120),nm_type-100
   110 continue
c――――――――――――――――――――――――★個人用ファイバーモデル
      New_fiber_work(nmx).i_stat  = -1             !  新規ファイバー要素の状態
      goto 100
   120 continue
c――――――――――――――――――――――――★
      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
      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.
      ray    = 0.
      raz    = 0.
      raz2   = 0.
      ray2   = 0.
      rayz   = 0.
      gg     = 0.
      aa     = 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
      enddo
      gg     = gg / aa
      if(it.eq.1) then
      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,
     *     E_model,E_model_fiber,M_model,M_model_fiber)
      return
      end
```

このサブルーチンを見れば分かるように、ファイバー要素に関する構造体の中で、Bilinear_work、Trilinear_work、Concrete_work 以外に、新たな構造体 New_fiber_work を使用する場合は、太文字で示した部分を変更し、初期設定を行うことになる。

次に、非線形剛性を求めるサブルーチン Stiff_M11()について調査する。このプログラムは、以下のように履歴モデルに関連するコードを含まない。そのため、新規履歴モデルを追加しても非線形剛性を求めるサブルーチンは影響しない。

```
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
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     ―――――――――――――――――――――――――――――――――――――
      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）
      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,
     *              raz2,ray2,rayz,gg,aa,rix,asy,asz)
      return
```

```
      end
```

次に、応力を求めるサブルーチンを調べよう。以下に示すサブルーチン Cal_stress_M11()の内容から、静的縮合型部材モデルでは、特殊な応力計算を行わない限り、現在のサブルーチンが転用可能となる。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /Cal_stress_M11（両端ファイバーモデル）
C       ―――――――――――――――――――――――――――――――
C     ●   代表的な部材モデルの応力計算：非線形剛性を用いて、材端応力を計算する
C     ―――――――――――――――――――――――――――――――――
      subroutine Cal_stress_M11(Model_type,Member,Element, ak,vv,r1,r2)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      include "submainx.h"
      record / member_s    / Member
      record / element_s   / Element
      dimension ak(12,12) ,vv(12),r1(3,3),r2(3,3),st(12),ss(12)
C     ―――――――――――――――――――――――――――――――――
      do i=1,12
      s=0.
      do j=1,12
      s=s+ak(i,j)*vv(j)
      enddo
      st(i)=s
      enddo
c―――――――――――――――――――――――――――――――★全体座標から部材座標へ変換
      call RotateL_v(1,st,r1,r2,ss)
      do i=1,6
      Member.stress(i)=-ss(i)+Member.stress(i)
      enddo
      do i=7,12
      Member.stress(i)=ss(i)+Member.stress(i)
      enddo
      return
      end
```

以上のように、静的縮合部材モデルのファイバー履歴特性では、新たな構造体を追加しなければ、標準仕様の３つのサブルーチンがそのまま使用できることが分かる。後は応力をチェックし、接線剛性を求めるサブルーチンであるが、このサブルーチンについては、次節で説明しよう。

9.3.6 モデルの階層構造とサブルーチンの組み込み

静定縮合部材モデルでは、サブルーチン submain_dynamic_a()から Check_stress()がコールされ、その中で、両端ファイバーモデルの応力の弾塑性チェックプログラム Cal_check_stiff_M11()が呼び出される。以下にその部分のコードを示す。

```
      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_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)
```

さらに、このサブルーチン Cal_check_stiff_M11()は、以下のような静的縮合に関する処理を行い、ファイバーの弾塑性チェックを行う。このサブルーチンから関連する部分を抜き出してみよう。

```
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     ――――――――――――――――――――――――――――――――――――――――
       .
       .
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
      enddo
       .
       .
c――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――
c    ★        部材の弾塑性チェック
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
```

```
      ・
      ・
      return
      end
```

このサブルーチンの中で、接線剛性を求める Stiff_M11()サブルーチンがコールされているが、この中でこの静的縮合モデルで使用されるエレメントモデルが階層構造となっている。このサブルーチンは前節で示した。

さらに、このサブルーチンの中で、静定縮合部材モデルである両端ファイバーモデルの応力チェック用サブルーチン Fiber_check()がコールされている。次は、この応力チェック用サブルーチンを見てみよう。

```
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(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"
      include "New_submain.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 vv(12)
c―――――――――――――――――――――――――――――――――――――★i 端部ファイバー
c―――――――――――――――――――――――――――――――――――――★歪のセット
      if(it.eq.1) then
      d_epsi_x_1 =  (vv(7)  - vv(1)) / Alength   !  軸方向歪
      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
      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
      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
      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_1 +                                    ! 軸方向歪
     *      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
      M_model_fiber(nnm).d_eps_x = M_model_fiber(nnm).d_eps_x + du !ファイバー要素の歪
      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
      if(nm_type/10.eq.0) then
c-----------------------------★弾塑性解析
      goto ( 10,20,30,40,50,60,70),nm_type
   10 continue
c-----------------------------★バイリニア型（スチール用）
      call BiLinear()
      goto 100
   20 continue
c-----------------------------★対称トリリニア型 （スチール用）
      call TriLinear()
      goto 100
   30 continue
c-----------------------------★コンクリート型
      call Concrete()
      goto 100
   40 continue
c-----------------------------★曲線コンクリート型
      call Concrete_e()
      goto 100
   50 continue
c-----------------------------★バイリニア型（移動＋等方効果用）
      call BiLinear_h()
      goto 100
   60 continue
c-----------------------------★対称トリリニア型 （移動＋等方効果用）
      call TriLinear_h()
      goto 100
   70 continue
c-----------------------------★非対称バイリニア型
      call BiLinear_AS()
```

```
      goto 100
      elseif(nm_type/10.eq.10) then
c―――――――――――――――――――――――――★弾塑性解析
      goto (110,120),nm_type - 100
  110 continue
c―――――――――――――――――――――――――★個人用ファイバー履歴特性
      call New_model_fiber()
      goto 100
  120 continue
c―――――――――――――――――――――――――★個人用ファイバー履歴特性
      goto 100
      endif
      endif
c―――――――――――――――――――――――――★弾塑性解析終了
  100 continue
      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(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
      nn      = E_model.nm_section_1 - 1
      nnm     = M_model.nm_section_1 - 1
      ra      = 0.
      ray     = 0.
      raz     = 0.
      raz2    = 0.
      ray2    = 0.
      rayz    = 0.
      gg      = 0.
      aa      = 0.
      AN      = 0.
      AMy     = 0.
      AMz     = 0.
      ra_before = M_model.d_ra_1
      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
      if(iistat.ge.nm_div) then
      ra=ra_before
      endif
      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
c―――――――――――――――――――――――――――――――★j 端部ファイバー
c―――――――――――――――――――――――――――――――★歪のセット
      elseif(it.eq.2) then
       •
       •
      do i=1,nm_div
       •
       •
      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
      if(nm_type/10.eq.0) then
c―――――――――――――――――――――――――★弾塑性解析
      goto ( 11,21,31,41,51,61,71),nm_type
  11  continue
c―――――――――――――――――――――――――★バイリニア型（スチール用）
      call BiLinear()
      goto 101
  21  continue
c―――――――――――――――――――――――――★対称トリリニア型 （スチール用）
      call TriLinear()
      goto 101
  31  continue
c―――――――――――――――――――――――――★コンクリート型
      call Concrete()
      goto 101
  41  continue
c―――――――――――――――――――――――――★曲線コンクリート型
      call Concrete_e(M_model_fiber()
      goto 101
  51  continue
c―――――――――――――――――――――――――★バイリニア型（移動＋等方効果用）
      call BiLinear_h()
      goto 101
  61  continue
c―――――――――――――――――――――――――★対称トリリニア型 （移動＋等方効果用）
```

```
      call TriLinear_h()
      goto 101
  71  continue
c――――――――――――――――――――――――★非対称バイリニア型
      call BiLinear_AS()
      goto 101
      elseif(nm_type/10.eq.10) then
c――――――――――――――――――――――――★弾塑性解析
      goto (111,121),nm_type - 100
  111 continue
c――――――――――――――――――――――――★個人用ファイバー履歴特性
      call New_model_fiber()
      goto 101
  121 continue
c――――――――――――――――――――――――★個人用ファイバー履歴特性
      goto 101
      endif
      endif
c――――――――――――――――――――――――★弾塑性解析終了
  101 continue
      enddo
c――――――――――――――――――――――――――――――――★ファイバー要素応力の計算
       ・
       ・
      return
      end
```

このサブルーチンの中に履歴モデルに関する第３層の階層構造が見られる。この中に履歴特性番号 101 として、太文字で示すコードがあり、新規モデルが追加されている。この中で、応力を弾塑性チェックする新規履歴モデルに関するサブルーチンが、両端ファイバーモデルであることから 2 箇所に加えられている。ここでは、サブルーチン名を New_model_fiber()としている。これで、既存サブルーチンに新規モデルが組み込まれたことになる。引数として受け渡すデータとして、構造体の Member と Element など多くの構造体や配列が、新規に登録されたサブルーチンには見られるはずである。新規サブルーチンでもデータの受け渡しには細心の注意が必要である。また、特殊な構造体を設計した場合、その構造体は引数として受け渡すことになるが、当然、上位のサブルーチンから受け渡されることになる。

この新規サブルーチンでは、実際にファイバーの履歴を追うことが主な処理であるが、少なくとも構造体 M_model_fiber と３つのワーク領域 Bilinear_work、Trilinear_work、Concrete_work かもしくは新たに設計した New_fiber_work 中で、次の値を設定しなければならない。

1. New_fiber_work (nmx).i_stat が-1 のとき、初期設定を行う。
2. ファイバーの接線剛性(M_model_fiber(nnm).d_E を設定する。
3. ファイバーの応力 M_model_fiber(nnm).d_stress_x を求める。