6.3 せん断型モデルの履歴

本節では、せん断型モデルの履歴特性について説明する。せん断型モデルの履歴に関連するサブルーチンは、初期設定と材料非線形性のチェックに関連する２つのサブルーチンである。ここでも、この履歴特性を管理するために階層構造を用いている。まず、この２つのサブルーチンを示すが、ここでは履歴に関連する部分についてのみ示し、他の部分は省く。

```
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     ―――――――――――――――――――――――――――――――――
      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            ! 4
        ak=Element.AK_1/Member.alength
      else                                                         ! 5
        ak=Element.AK_1
      endif
      ak_linear(2,2)= ak                                           ! 6
      ak_linear(2,8)=-ak
      ak_linear(8,8)= ak
      ak_linear(8,2)=-ak
      ak_linear(3,3)= ak
      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     ! 7
      Member.AKw_tan=ak          ! Member.AKw_tan=Element.AK_1
      Member.istat_v=0
      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（モデル No.2）
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                        ! 9
    5 continue
c―――――――――――――――――――――――――――――――★規定モデル：武田モデル
      call Takeda_TriLiner(Member,Element,vv,vpp)                  ! 10
      goto 999
   10 continue
c―――――――――――――――――――――――――――――――★トリリニア：Nomal
      call Mesing_TriLiner(Member,Element,vv,vpp)                  ! 11
      goto 999
   20 continue
c―――――――――――――――――――――――――――――――★トリリニア：最大点指向型
      call DirecMax_TriLiner(Member,Element,vv,vpp)                ! 12
      goto 999
   30 continue
c―――――――――――――――――――――――――――――――★トリリニア：武田モデル
      call Takeda_TriLiner(Member,Element,vv,vpp)                  ! 13
      goto 999
   40 continue
c―――――――――――――――――――――――――――――――★
      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
      mro=Element.n_section(1)
      call Modify_Bi_Liner1(Member,Element,RO_work(mro),vv,vpp)    ! 14
      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)                    ! 15
      goto 999
      endif
  999 continue
      return
      end
```

1. このサブルーチンは、せん断型モデル部材の線形剛性行列を求めるもので、ここで、このモデルに含まれる全履歴モデル共通の初期設定を行う。各履歴モデルで必要となる初期設定は他の部分で行う。これについては、直ぐ後で説明する。まず、線形剛性行列を計算するために、剛性行列をゼロクリアする。
2. 軸方向剛性を Element 構造体から取得する。せん断型モデルは一般には軸方向の変形場を含まないが、ここでは、平面骨組みに組み込まれることを考慮して軸方向剛性をセットする。この軸方向剛性を剛性行列の適切な位置にセットする。
3. せん断剛性を構造体より取得する。ここでは、修正バイリニアモデルの線形剛性を取り出す。
4. ここでは、修正 RO モデルの線形剛性を取り出す。
5. 一般のせん断型の線形剛性を取得し、ak にセットする。
6. 線形のせん断剛性を y 方向と z 方向のバネ定数として剛性行列に配置する。現在は、入力仕様によって、y 方向と z 方向の線形剛性並びに履歴特性が同じとしているが、せん断型の立体振動を行うためには、異なったデータをセットする必要がある。いずれ、入力仕様を変更して対処することになる。
7. ここで、履歴チェックのための初期設定を行う。接線剛性に線形のせん断剛性をセットする。また、y 方向と z 方向の状態パラメータを 0（弾性）とする。
8. このサブルーチンでは、せん断型の履歴モデルの弾塑性チェックを行い、その接線剛性を求める。まず、最初に部材の履歴モデル番号を取得する。
9. その番号にしたがって履歴モデルに関係するサブルーチンをコールする。
10. ここは、履歴モデル番号：0 番であり、既定モデルとなっている。現在の仕様では、武田モデルとなっている。
11. 履歴モデル番号：1 番は、トリリニア型の履歴特性となっている。

12. 履歴モデル番号：2 番は、最大点指向型トリリニア型の履歴特性となっている。
13. 履歴モデル番号：3 番は、武田モデルの履歴特性となっている。
14. 履歴モデル番号：11 番は、修正バイリニア型の履歴特性となっている。
15. 履歴モデル番号：12 番は、修正 RO モデルの履歴特性となっている。

6.3.1 最大点指向型トリリニアの履歴

本節では、図 6-8 に示すせん断型モデルの履歴ルールである最大点指向型トリリニア履歴モデルについて説明する。

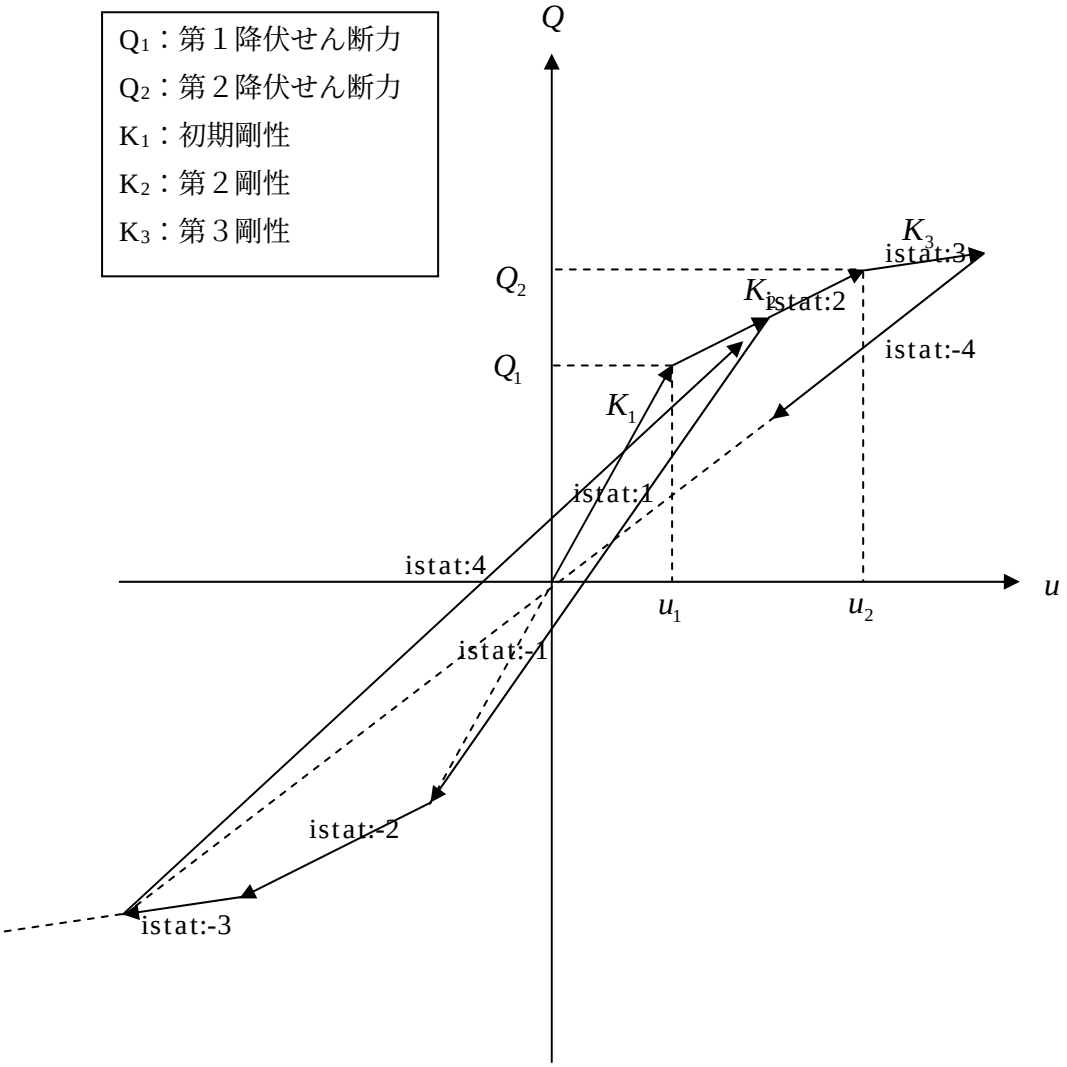

図 6-8 最大点指向型モデルの履歴特性

最大点指向型の履歴ルールを次のようにまとめる。

1. 骨格曲線はトリリニア型である。
2. 初めて一方の変位が最大変位 u_1 を超えて折り返した場合は、反対側の第１降伏点を目指す。
3. 変位が共に最大変位 u_1 を超えて折り返した場合は、反対側の最大変位点を目指す。

この履歴ルールにしたがって次のサブルーチンが設計されている。このサブルーチン DirecMax_TriLiner()を具体的に示そう。このサブルーチンは、前節のファイバーの弾塑性チェックプログラムと、かなり異なった方法で作られている。特に、せん断力を増分型で求めていないのが特徴である。また、状態パラメータは、正方向に向かうときは正に、負方向に向かうときは負としている。骨格曲線を形成する 3 つの直線で、状態パラメータは、第１勾配を１に、第 2 勾配を 2 に、第 3 勾配を 3 に、また、最大点に向かう直線を 4 とする。良く読んで理解されたい。

```
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE /DirecMax_TriLiner
C       ―――――――――――――――――――――――――――――――――
C     ●   Direc. Max TriLiner 履歴モデル
C     ―――――――――――――――――――――――――――――――――
      subroutine DirecMax_TriLiner(Member,Element,vv,vpp)
      implicit real*8(V)
      include "submain.h"
      record / member_s2_DirecMax / Member
      record / element_s2         / Element
      integer           i          ! カウンター
      dimension vv(12)   ! 部材座標系増分変位
      dimension vpp(12)  ! 部材座標系直前変位
      real*8            dU               ! 現在の相対増分変位
      real*8            Up               ! 直前の相対変位
C     ―――――――――――――――――――――――――――――――――
c
c     部材
      structure / member_s2_DirecMax /
      integer  nm_element         !  要素番号
      integer  element_type       !  要素タイプ
      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     !  要素タイプ別番号
      integer  analysis_3D        !  解析型 0:3D 1:2D(x-z) 2:2D(y-z)
      integer  nm_so              !  部材の層番号
      integer  nm_dll_element     !  DLL を用いた要素か（０；システム内要素、１：DLL 要素）
      integer  nm_point(2)        !  節点番号
```

```
      integer  irest(12)          !  部材両端の自由度番号表
                                  !istat は履歴特性で使用（インデント 1:y 方向，2:z 方向）
      integer  istat(2)           !  履歴特性の状態(*)
      integer  nm_analysis        !  部材解析種別
      integer  nm_group           !  部材グループ
      integer  nm_local_coord(2)  !  局所座標系の有無とその回転行列の番号
      integer  nm_damp            !  部材減衰の有無とその減衰行列の番号
      real*8   alength            !  長さ
      real*8   i_rigid_length     !  i 端剛域長さ
      real*8   j_rigid_length     !  j 端剛域長さ
      real*8   i_shear_G          !  i 端せん断剛性
      real*8   j_shear_G          !  j 端せん断剛性
      real*8   rot_x              !  部材主軸の回転角度（度）
      real*8   force(12)          !  部材両端の部材端力（釣合座標系）
      real*8   stress(6)          !  部材中央の応力（部材座標系）
                                  !  AK_tan は履歴特性で使用（インデント 1:y 方向，2:z 方向）
      real*8   AK_tan(2)          !  接線剛性(*)
                                  !  以下，履歴特性で使用（インデント 1:y 方向，2:z 方向）
      real*8   Uo(2)              !  目標点の変位
      real*8   Qo(2)              !  目標点の剪断力
      real*8   Ur(2)              !  荷重反転時の目標点の変位(istat=1,4)
      real*8   Qr(2)              !  荷重反転時の目標点の剪断力(istat=1,4)
      real*8   UmaxP(2)           !  正載荷側の最大変位
      real*8   QmaxP(2)           !  正載荷側の最大剪断力
      real*8   UmaxM(2)           !  負載荷側の最大変位
      real*8   QmaxM(2)           !  負載荷側の最大剪断力
      real*8   dmw(4)             !  ダミー
      integer   d_stat(3)         !  断面の弾塑性状態　(1) i 端 (2) j 端　(3) 中央
      real*8   an_vv(10)          !  部材軸力（計算用内部変位ｖ）
      real*8   an_ww(10)          !  部材軸力（計算用内部変位w）
      end structure
C
C     ―――――――――――――――――――――――――――――――――
C       第一ステップ処理
      if(Member.istat(1).eq.0.and.Member.istat(2).eq.0) then                    ! 1
      call Initial_DirecMax_TriLiner(Element,Member)                           ! 2
      return
      endif
C     ―――――――――――――――――――――――――――――――――
C       第 n ステップ処理
      do i=1,2                                                                 ! 3
      dU=vv(7+i)-vv(1+i)                ! 現在の相対増分変位の計算
      Up=vpp(7+i)-vpp(1+i)              ! 直前の相対変位の計算
      call Cal_DirecMax_TriLiner(i, Element, Member, dU, Up)                   ! 4
      end do
      return
      end
C     ―――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE Initial_DirecMax_TriLiner
C       ―――――――――――――――――――――――――――――――――
C     ●   1 ステップの初期化
C     ―――――――――――――――――――――――――――――――――
      subroutine Initial_DirecMax_TriLiner(Element,Member)
      include "submain.h"
```

```
      record / member_s2_DirecMax / Member
      record / element_s2         / Element
      integer    i       ! カウンター
C     ―――――――――――――――――――――――――――――――――――
                                                        !エラー入力処理
      if(Element.AK_1.le.0.0) then                                    ! 5
      Element.AK_1=1.0
      endif
      if(Element.AK_2.le.0.0) then
      Element.AK_2=Element.AK_1
      endif
      if(Element.Q_2.le.Element.Q_1) then                             ! 6
      Element.Q_2=Element.Q_1
      endif
                                                        !フラグの初期化
      do i=1,2                                                        ! 7
      Member.istat(i)  = 1                          ! 状態パラメータ
      Member.AK_tan(i) = Element.AK_1               ! 接線剛性
      Member.Uo(i)     = Element.Q_1/Element.AK_1    ! 第４状態における正側の直線式の変位
      Member.Qo(i)     = Element.Q_1                 ! 第４状態における正側の直線式のせん断力
      Member.Ur(i)     =-Element.Q_1/Element.AK_1    ! 第４状態における負側の直線式の変位
      Member.Qr(i)     =-Element.Q_1                 ! 第４状態における負側の直線式の応力
      Member.UmaxP(i)  = Element.Q_1/Element.AK_1    ! 正側の最大変位点の初期設定
      Member.QmaxP(i)  = Element.Q_1                 ! 正側の最大応力点の初期設定
      Member.UmaxM(i)  =-Element.Q_1/Element.AK_1    ! 負側の最大変位点の初期設定
      Member.QmaxM(i)  =-Element.Q_1                 ! 負側の最大応力点の初期設定
      end do
      Element.U_1=Element.Q_1/Element.AK_1                            ! 8
      Element.U_2=(Element.Q_2-Element.Q_1)/Element.AK_2
     *           +Element.U_1
      return
      end
C     ―――――――――――――――――――――――――――――――――――
C     ●   SUBROUTINE Cal_DirecMax_TriLiner
C       ―――――――――――――――――――――――――――――――――――
C     ●   n ステップ
C     ―――――――――――――――――――――――――――――――――――
      subroutine Cal_DirecMax_TriLiner(id,Element,Member,dU,Up)
      include "submain.h"
      record / member_s2_DirecMax / Member
      record / element_s2         / Element
      integer           id                    ! 方向 ID
      real*8            dU                    ! 増分相対変位
      real*8            Up                    ! 直前の相対変位
      real*8            Qp                    ! 直前の剪断力
      real*8            U                     ! 現在の相対変位
      real*8            Q                     ! 現在の剪断力
      real*8            tmp(10)               ! 作業領域
      integer           iflag                 ! 接線剛性変化フラグ
C     ―――――――――――――――――――――――――――――――――――
                                                        !現在の変位の計算
      U=Up+dU                                                         ! 9
                                                        !載荷方向の反転
      if( dU*real(Member.istat(id)).lt.0.0 ) then                     ! 10
```

```
        tmp(1)=Member.Uo(id)
        tmp(2)=Member.Qo(id)
        if( iabs(Member.istat(id)).eq.1.or.                              ! 11
     *      iabs(Member.istat(id)).eq.4 ) then
          Member.Uo(id)=Member.Ur(id)                                    ! 12
          Member.Qo(id)=Member.Qr(id)
          Member.Ur(id)=tmp(1)
          Member.Qr(id)=tmp(2)
          Member.istat(id)=-1*Member.istat(id)
        else                                                             ! 13
                                                          !直前の剪断力の計算
          Qp=CalQ_DirecMax_TriLiner(id,Element,Member,Up)                ! 14
          Member.Ur(id)=Up                                               ! 15
          Member.Qr(id)=Qp
          if( Member.istat(id).gt.0 ) then                               ! 16
            Member.Uo(id)=Member.UmaxM(id)
            Member.Qo(id)=Member.QmaxM(id)
            Member.istat(id)=-4
          else                                                           ! 17
            Member.Uo(id)=Member.UmaxP(id)
            Member.Qo(id)=Member.QmaxP(id)
            Member.istat(id)=4
          endif
        Member.AK_tan(id) = (Member.Qo(id) - Member.Qr(id))               ! 18
     *                    / (Member.Uo(id) - Member.Ur(id))
        endif
        return
      endif
C     ―――――――――――――――――――――――――――――――――――
                                                          !接線剛性の変化のチェック
      if( iabs(Member.istat(id)).ne.3 ) then                             ! 19
C     ―――――――――――――――――――――――――――――――――――
        if( Member.istat(id).gt.0 ) then                                 ! 20
          if( Member.Uo(id).lt.U ) then
            iflag=1
          else
            iflag=0
          endif
        else                                                             ! 21
          if( Member.Uo(id).gt.U ) then
            iflag=1
          else
            iflag=0
          endif
        endif
C     ―――――――――――――――――――――――――――――――――――
      if( iflag.eq.1 ) then                                              ! 22
        if( iabs(Member.istat(id)).eq.1 ) then                           ! 23
          Member.AK_tan(id)=Element.AK_2
          if ( Member.istat(id).gt.0 ) then                              ! 24
            Member.istat(id)= 2
            Member.Uo(id)   = Element.U_2
            Member.Qo(id)   = Element.Q_2
          Else                                                           ! 25
```

```
          Member.istat(id)=-2
          Member.Uo(id)   =-Element.U_2
          Member.Qo(id)   =-Element.Q_2
        endif
      else if( iabs(Member.istat(id)).eq.2 ) then                    ! 26
        Member.AK_tan(id)=Element.AK_3
        if ( Member.istat(id).gt.0 ) then                            ! 27
          Member.istat(id)=3
        else                                                         ! 28
          Member.istat(id)=-3
        endif
      else                                                           ! 29
        if( dabs(U).lt.dabs(Element.U_2) ) then                      ! 30
          Member.AK_tan(id)=Element.AK_2
          if ( Member.istat(id).gt.0 ) then                          ! 31
           Member.istat(id)= 2
            Member.Uo(id)   = Element.U_2
            Member.Qo(id)   = Element.Q_2
          else                                                       ! 32
            Member.istat(id)=-2
            Member.Uo(id)   =-Element.U_2
            Member.Qo(id)   =-Element.Q_2
          endif
        else                                                         ! 33
            Member.AK_tan(id)=Element.AK_3
          if ( Member.istat(id).gt.0 ) then                          ! 34
            Member.istat(id)=3
          else                                                       ! 35
            Member.istat(id)=-3
          endif
        endif
      endif
      Member.stress(1+id)                                            ! 36
     *      =CalQ_DirecMax_TriLiner(id,Element,Member,U)
      endif
      endif
C     ――――――――――――――――――――――――――――――――――――
                                                    !現在の剪断力の計算
      Q=CalQ_DirecMax_TriLiner(id,Element,Member,U)                  ! 37
                                                    !最大振幅値のチェック
      if( Member.istat(id).gt.0 ) then                               ! 38
        if( Member.UmaxP(id).lt.U ) then                             ! 39
          Member.UmaxP(id) = U
          Member.QmaxP(id) = Q
        endif
      else                                                           ! 40
        if( Member.UmaxM(id).gt.U ) then
          Member.UmaxM(id) = U
          Member.QmaxM(id) = Q
        endif
      endif
      return
      end
C     ――――――――――――――――――――――――――――――――――――
```

```
C     ●   real*8 function CalQ_DirecMax_TriLiner
C       ―――――――――――――――――――――――――――――――――
C     ●   変位 U に対応する剪断力 Q の計算
C     ―――――――――――――――――――――――――――――――――
      real*8 function CalQ_DirecMax_TriLiner(id,Element,Member,U)
      include "submain.h"
      record / member_s2_DirecMax / Member
      record / element_s2         / Element
      integer           id                    ! 方向 ID
      real*8            U                     ! 現在の相対変位
      real*8            Q                     ! 現在の剪断力
C     ―――――――――――――――――――――――――――――――――
      if( iabs(Member.istat(id)).eq.1) then                          ! 41
        Q=Element.AK_1*U
      else if( iabs(Member.istat(id)).eq.2) then                     ! 42
        if( Member.istat(id).gt.0 ) then                             ! 43
          Q=Element.AK_2*(U-Element.U_1) +Element.Q_1
        else                                                         ! 44
          Q=Element.AK_2*(U+Element.U_1) -Element.Q_1
        endif
      else if( iabs(Member.istat(id)).eq.3) then                     ! 45
        if( Member.istat(id).gt.0 ) then                             ! 46
          Q=Element.AK_3*(U-Element.U_2) +Element.Q_2
        else                                                         ! 47
          Q=Element.AK_3*(U+Element.U_2) -Element.Q_2
        endif
      else                                                           ! 48
          Q=Member.AK_tan(id)*(U-Member.Ur(id)) +Member.Qr(id)
      endif
          CalQ_DirecMax_TriLiner=Q                                   ! 49
      return
      end
```

1. 両方向の初期設定を行うかどうかチェックする。
2. せん断型モデルの履歴特性用ワーク領域を初期設定するために、サブルーチン Initial_DirecMax_TriLiner()をコールする。
3. 両方向の処理を行う。増分相対変位と増分前の相対変位をセットする。
4. 履歴チェックを行うサブルーチン Cal_DirecMax_TriLiner()をコールする。
5. このサブルーチンでは初期設定を行う。まず、初期剛性が適切でない場合、剛性の値として 1.0 を設定する。
6. 第 1 折れ点と第 2 折れ点のせん断力が逆転している場合は、第 2 折れ点のせん断力を第 1 折れ点の値とする。
7. 両方向についてワーク用構造体 Member の成分を初期設定する。
8. 図に示す第 1 折れ点の変位 u1 と第 2 折れ点の変位 u2 を計算し、構

造体 Element の成分にセットする。

9. このサブルーチンでせん断型モデルの履歴チェックを行う。まず、増分後の相対変位をセットする。
10. 状態パラメータの方向と増分変位の関係を調べ、変位が反転すると次の処理を行う。まず、正側の第 1 折れ点の座標をワーク配列 tmp にコピーする。
11. 状態パラメータが 1 もしくは 4 の場合は、次の処理を行う。
12. 正側と負側の基本座標を入れ替える。
13. 状態パラメータが 2 もしくは 3 の場合は、履歴が反転し、次の処理を行う。
14. サブルーチン Cal_DirecMax_TriLiner()を用いて、増分前のせん断力を計算する。
15. 反転位置の座標(Up,Qp)を構造体成分 Member.Ur(id)、Member.Qr(id)にセットする。
16. 増分前の状態で、どちら方向に変位が進んでいたかチェックする。ここでは、正方向に進んでいたので、状態パラメータを-4 にセットする。また、直線式の相手側の座標を、負側の最大点の座標としてセットする。
17. ここでは、負方向に進んでいたので、状態パラメータを 4 にセットする。また、直線式の相手側の座標を、正側の最大点の座標としてセットする。
18. 骨格曲線より反転したため、最大点を指向する直線式の傾きを計算し、その値を接線剛性とする。その接線剛性の値を構造体成分 Member.AK_tan(id)に保存する。これで、骨格曲線より反転した処理を終了し、このサブルーチンより戻る。
19. ここからは、骨格曲線からの反転処理以外の処理について行う。したがって、状態パラメータの方向と増分変位の方向は同じである。まず、状態パラメータが 3 以外の場合、つまり、1、2、4 の場合について次の処理を行う。状態パラメータが 3 の場合は、境界がないため、境界を越えたかどうかのチェックを必要としない。
20. まず、変位の進行方向が正の場合について処理を行う。増分後の変位がその状態の境界変位 Member.Uo(id)より大きい場合、状態を変更しなければならないので、変更フラグを iflag=1 とする。変位が境界を超えない場合は、iflag=0 とする。
21. ここからは、変位の進行方向が負の場合について処理を行う。増分後の変位がその状態の境界変位 Member.Uo(id)より小さい場合、状態を変更しなければならないので、変更フラグを iflag=1 とする。変

位が境界を超えない場合は、iflag=0 とする。
22. 状態変更フラグが 1 の場合では、ここ以降の処理で、接線剛性や状態パラメータの変更を行う。
23. 状態パラメータが 1 か -1 の場合、まず、接線剛性を第 2 勾配の剛性をセットする。
24. 次に、変位の進行方向を調査し、正方向の場合は状態パラメータを 2 にセットし、境界座標として第 2 折れ点の座標をセットする。
25. 変位の進行方向が負方向の場合、状態パラメータを-2 にセットし、境界座標として負の第 2 折れ点の座標をセットする。
26. 状態パラメータが 2 か -2 の場合、まず、接線剛性を第 3 勾配の剛性をセットする。
27. 次に、変位の進行方向を調査し、正方向の場合は状態パラメータを 3 にセットする。
28. 変位の進行方向が負方向の場合､状態パラメータを-3 にセットする。
29. ここでは、状態パラメータの絶対値が 1 か 4 の場合についてチェックする。
30. 変位 U が第 2 折れ点 Element.U_2 より小さい場合、状態変更フラグが 1 であるので、まず、接線剛性に第 2 勾配の剛性をセットする。
31. 変位の進行方向が正の場合、状態パラメータを 2 にセットし、境界座標として第 3 折れ点の座標をセットする。
32. 変位の進行方向が負方向の場合、状態パラメータを-2 にセットし、境界座標として、負の第 3 折れ点の座標をセットする。
33. 変位 U が第 2 折れ点 Element.U_2 より大きい場合、状態変更フラグが 1 であるので、まず、接線剛性に第 3 勾配の剛性をセットする。
34. 変位の進行方向が正の場合、状態パラメータを 3 にセットし、境界座標は必要としないので設定しない。
35. 変位の進行方向が負の場合、状態パラメータを-3 にセットし、境界座標は必要としないので設定しない。
36. サブルーチン CalQ_DirecMax_TriLiner()を用いて、増分後のせん断力を求め、構造体の応力にセットする。ここで、状態変更フラグが 1 の場合の処理が終了する。
37. 最大変位と最大せん断力をセットするために、現時点のせん断力を求める。
38. 変位の進行方向をチェックする。
39. 変位が正方向の変位より、大きい場合は最大値をセットする。
40. 変位が負方向の変位より、小さい場合は最小値をセットする。

41. このサブルーチン CalQ_DirecMax_TriLiner()では、状態パラメータにしたがって増分後のせん断力を計算する（図 6-8 参照)。状態パラメータが１の場合は、簡単で、線形剛性に変位を掛けて求める。
42. 状態パラメータが絶対値で 2 の場合の処理を行う。
43. 状態パラメータが 2 の場合、正側第 2 勾配上のせん断力 Q を計算する。
44. 状態パラメータが-2 の場合、負側第 2 勾配上のせん断力 Q を計算する。
45. 状態パラメータが絶対値で 3 の場合の処理を行う。
46. 状態パラメータが 3 の場合、正側第 3 勾配上のせん断力 Q を計算する。
47. 状態パラメータが-3 の場合、負側第 3 勾配上のせん断力 Q を計算する。
48. 状態パラメータが-4 の場合、図に示すような状態４の直線を決定する式からせん断力 Q を求める。
49. 求めたせん断力 Q を関数名 CalQ_DirecMax_TriLiner にセットする。

これで、履歴特性を表す代表的なサブルーチンの説明を終えるが、他の履歴については、付録に収録されているサブルーチンを見て、学習されたい。無論、ここで示した手法だけがこの履歴ルールを実現するわけではない。読者も他の手法で書いてみるとより理解が深まることと思う。是非、試されたい。また、他の履歴ルールを実現し、この SPACE に組み込みたい場合は、第 9 章を参考にして実行されると良い。