



第2章 せん断型履歴モデル

2.1 はじめに

SPACE に組み込まれている履歴モデルには、せん断型モデルの履歴やファイバーの履歴、あるいは個材座屈を考慮したトラスモデルのように部材特有の履歴モデルもある。本節では、部材モデル階層構造の最下層に位置する履歴モデルについて解説する。履歴モデルは最下層にあるため、他のプログラムと強く関連することがないことから、複雑なバグを発生させる可能性は少ない。しかし、履歴モデルには、それ特有の難しさやエラーを生じさせる要因を含んでおり、あらゆる場合を想定して徹底的にチェックする必要がある。

特に、本章では、次に示すせん断型モデルの履歴について解説する。

2.2.2 Mesing 型トリリニア

2.2.3 最大点指向型トリリニア

2.2.4 武田モデル

2.2.5 木質パネル

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 Cal_lin_stiff_M2() と Cal_check_stiff_M2() 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。

2.2 せん断型履歴モデル

2.2.1 履歴モデルの構造

```

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
      do j=1,12
      ak_linear(j,i) = 0.0

```

! 1

```

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 DirectMax_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                                     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                                     バイリニア : S字スリップ型
  call S_slip_bilinear_s(Member,Element,vv,vpp)
  goto 999
60 continue
c                                     木質パネル : S字スリップ型
  call S_slip_woodpanel(Member,Element,vv,vpp)
  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 R0 モデル
  mro=Element.n_section(1)
  call Modify_R01(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 R0 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、変数 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、武田モデルである。
11. 履歴モデル番号 : 1 番は、トリリニア型の履歴特性であり。
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である。
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である。
15. 履歴モデル番号 : 12 番は、修正 R0 モデルの履歴特性である。

本節では、図 2-1 に示すせん断型モデルの履歴ルールである Mesing 型トリリニア履歴モデルについて説明する。このモデルでは、スケルトンカーブがトリリニアで Mesing 則に従う履歴ループを有する。

履歴ループをモデル化する際の入力パラメータを以下に示す。

2.2.2 Mesing 型 トリリニア

Q_1 : 第1降伏せん断力

Q_2 : 第2降伏せん断力

K_1 : 初期剛性

K_2 : 第2剛性

K_3 : 第3剛性

上記の入力パラメータより 初期設定で次のように求められる。

$$u_1 = Q_1 / K_1$$

$$u_2 = u_1 + (Q_2 - Q_1) / K_2$$

Masing 則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. $-u_2 \leq u \leq u_2$ の範囲は、バイリニア型として移動する。
3. 変位 u が u_2 を超えて折り返した場合は、剛性は初期剛性を用い、この状態では、変位は u_r から $2u_1$ の範囲である。続いて、変位が進むと接線剛性は第2勾配となり、その状態の変移は u_r から $2u_2$ までとなる。その後は第3勾配へ移行する。

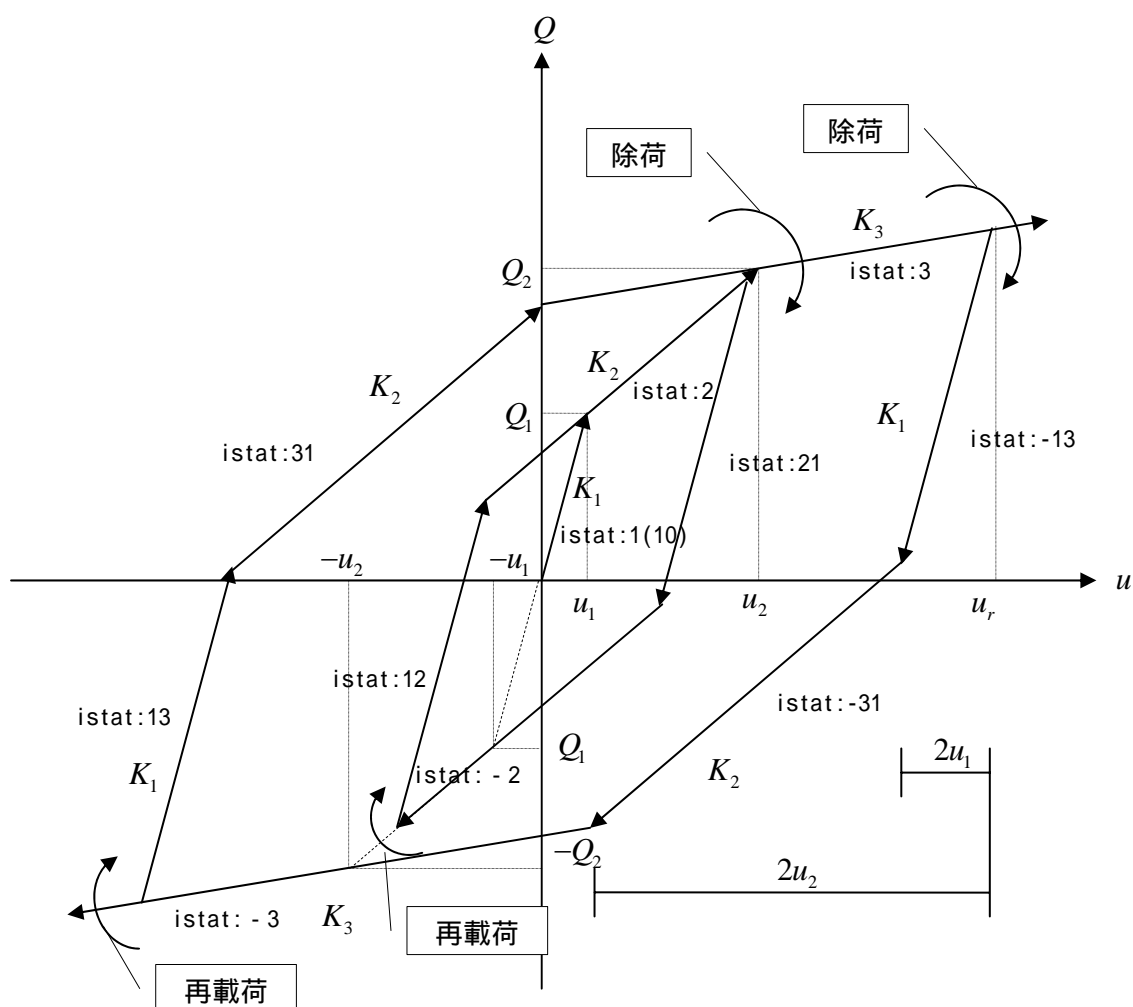


図 2-1 Masing 型トリリニアモデルの履歴特性

この履歴ルールにしたがって次のサブルーチンを設計する。このサブルーチン `Mesing_TriLiner()` を具体的に示そう。このサブルーチンは、後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、状態パラメータの値は、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、これより状態パラメータについて説明していく。

第1勾配の直線を状態:1に、同じく第2勾配を2に、第3勾配を3にすると、状態1、2、3は骨格曲線上であることを意味する。ここで U_0, Q_0 と U_r, Q_r は、目標点と反転時の点である。

次に第2勾配(状態2)と第3勾配(状態3)から除荷、又は再載荷した場合について述べる。まず、第2勾配より変位が反転した場合、初期勾配で変位が進むことになり、状態は12と21である。第3勾配より変位が反転した場合、初期勾配で U_r より $2u_1$ へと進む。第2勾配上の釣合点は、 $2(u_2 - u_1)$ へ進むと、第3勾配へと変位が移る。そのときの初期勾配での状態は13、第2勾配での状態は31となる。

以下に関連するサブルーチンを示す。

```

C
C      SUBROUTINE /Mesing_TriLiner
C
C      Mesing TriLiner 履歴モデル
C
      subroutine Mesing_TriLiner(Member,Element,vv,vpp)
      implicit real*8(V)
      include "submain.h"
      record / member_s2_Mesing / 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      第一ステップ処理
C
      if(Member.istat(1).eq.0.and.Member.istat(2).eq.0) then          ! 1
      call Initial_Mesing_TriLiner(Element,Member)                    ! 2
      return
      endif
C
C      第nステップ処理
C
      do i=1,2
      dU=vv(7+i)-vv(1+i)          ! 現在の相対増分変位の計算          ! 3

```

```

Up=vpp(7+i)-vpp(1+i)          ! 直前の相対変位の計算
call Cal_Mesing_TriLiner(i, Element, Member, dU, Up)          ! 4
end do
return
end

```

```

C
C      SUBROUTINE Cal_Mesing_TriLiner
C
C      n ステップ
C
      subroutine Cal_Mesing_TriLiner(id,Element,Member,dU,Up)
      include "submain.h"
      record / member_s2_Mesing / 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

! 現在の変位の計算
      U=Up+dU          ! 5
! 載荷方向の反転
      if( dU*real(Member.istat(id)).lt.0.0 ) then          ! 6
      if( iabs(Member.istat(id)).eq.10.or.          ! 7
*      iabs(Member.istat(id)).eq.12.or.
*      iabs(Member.istat(id)).eq.21.or.
*      iabs(Member.istat(id)).eq.13.or.
*      iabs(Member.istat(id)).eq.31) then
      tmp(1)          = Member.Uo(id)          ! 8
      tmp(2)          = Member.Qo(id)
      Member.Uo(id) = Member.Ur(id)
      Member.Qo(id) = Member.Qr(id)
      Member.Ur(id) = tmp(1)
      Member.Qr(id) = tmp(2)
      if(iabs(Member.istat(id)).eq.10) then          ! 9
          Member.istat(id) = -1 * Member.istat(id)
      else if(iabs(Member.istat(id)).eq.12) then          ! 10
          Member.istat(id) = -21 * IFlag_MT(Member.istat(id))
      else if(iabs(Member.istat(id)).eq.21) then          ! 11
          Member.istat(id) = -12 * IFlag_MT(Member.istat(id))
      else if(iabs(Member.istat(id)).eq.13) then          ! 12
          Member.istat(id) = -31 * IFlag_MT(Member.istat(id))
      else if(iabs(Member.istat(id)).eq.31) then          ! 13
          Member.istat(id) = -13 * IFlag_MT(Member.istat(id))
      endif
      else          ! 14
      !直前の剪断力の計算
      Qp=CalQ_Mesing_TriLiner(id,Element,Member,Up)          ! 15

```

```

        Member.Ur(id)=Up
        Member.Qr(id)=Qp
        if(iabs(Member.istat(id)).eq.2) then                                ! 16
            Member.istat(id)=-12*IFlag_MT(Member.istat(id))

            Member.UoN(id) = Member.Uo(id)                                ! 17
            Member.QoN(id) = Member.Qo(id)
            Member.UoP(id) = Member.Uo(id)                                ! 18
            *
            +2.0*Element.U_2*IFlag_MT(Member.istat(id))
            Member.QoP(id) = Member.Qo(id)
            *
            +2.0*Element.Q_2*IFlag_MT(Member.istat(id))
        else if(iabs(Member.istat(id)).eq.3) then                            ! 19
            Member.istat(id)=-13*IFlag_MT(Member.istat(id))
        endif
        Member.Uo(id)                                                        ! 20
        *
        =Up+2.0*Element.U_1*IFlag_MT(Member.istat(id))
        Member.Qo(id)
        *
        =Qp+2.0*Element.Q_1*IFlag_MT(Member.istat(id))
        Member.AK_tan(id) = Element.AK_1
    endif
    return
endif
!   接線剛性の変化のチェック
    if( iabs(Member.istat(id)).ne.3 ) then                                ! 21
    if( Member.istat(id).gt.0 ) then                                        ! 22
        if( Member.Uo(id).lt.U ) then
            iflag=1
        else
            iflag=0
        endif
    else                                                                    ! 23
        if( Member.Uo(id).gt.U ) then
            iflag=1
        else
            iflag=0
        endif
    endif
    if( iflag.eq.1 ) then                                                  ! 24
        if( iabs(Member.istat(id)).eq.10 ) then                          ! 25
            Member.AK_tan(id)=Element.AK_2
            Member.istat(id) = 2 * IFlag_MT(Member.istat(id))
            Member.Ur(id)=Element.U_1 * IFlag_MT(Member.istat(id))
            Member.Qr(id)=Element.Q_1 * IFlag_MT(Member.istat(id))
            Member.Uo(id)=Element.U_2 * IFlag_MT(Member.istat(id))
            Member.Qo(id)=Element.Q_2 * IFlag_MT(Member.istat(id))
        else if( iabs(Member.istat(id)).eq.2 ) then                        ! 26
            Member.AK_tan(id)=Element.AK_3
            Member.Ur(id)=Member.Uo(id)
            Member.Qr(id)=Member.Qo(id)

            Member.istat(id)=3 * IFlag_MT(Member.istat(id))
        else if( iabs(Member.istat(id)).eq.13 ) then                       ! 27
            tmp(1) = Member.Uo(id)
            tmp(2) = Member.Qo(id)

```

```

        Member.Uo(id)=Member.Ur(id)                                ! 28
*          +2.0*Element.U_2*IFlag_MT(Member.istat(id))
        Member.Qo(id)=Member.Qr(id)
*          +2.0*Element.Q_2*IFlag_MT(Member.istat(id))
        Member.Ur(id)=tmp(1)
        Member.Qr(id)=tmp(2)
        Member.AK_tan(id) = Element.AK_2
        Member.istat(id) = 2*IFlag_MT(Member.istat(id))
    else if( iabs(Member.istat(id)).eq.31 ) then                    ! 29
        Member.Ur(id)=Member.Uo(id)
        Member.Qr(id)=Member.Qo(id)
        Member.AK_tan(id) = Element.AK_3
        Member.istat(id) = 3*IFlag_MT(Member.istat(id))
    else if( iabs(Member.istat(id)).eq.12 ) then                    ! 30
        Member.Ur(id)=Member.Uo(id)
        Member.Qr(id)=Member.Qo(id)
        Member.Uo(id)=Member.UoP(id)
        Member.Qo(id)=Member.QoP(id)
        Member.AK_tan(id) = Element.AK_2
        Member.istat(id) = 2*IFlag_MT(Member.istat(id))
    else if( iabs(Member.istat(id)).eq.21 ) then                    ! 31
        Member.Ur(id)=Member.Uo(id)
        Member.Qr(id)=Member.Qo(id)
        Member.Uo(id)=Member.UoN(id)
        Member.Qo(id)=Member.QoN(id)
        Member.AK_tan(id) = Element.AK_2
        Member.istat(id) = 2*IFlag_MT(Member.istat(id))
    endif
    Member.stress(1+id)
*    =CalQ_Mesing_TriLiner(id,Element,Member,U)                    ! 32
endif
endif
return
end

```

```

C
C      real*8 function CalQ_Mesing_TriLiner
C
C      変位 U に対応する剪断力 Q の計算
C
    real*8 function CalQ_Mesing_TriLiner(id,Element,Member,U)
    include "submain.h"
    record / member_s2_Mesing / 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
    Q=Member.AK_tan(id)*(U-Member.Ur(id)) +Member.Qr(id)          ! 33
    CalQ_Mesing_TriLiner=Q                                         ! 34
    return
end

```

```

C
C      integer function IFlag_Mesing_TriLiner
C
C      整数の符号判定
C
      integer function IFlag_MT(n)
      integer n
      if( n.eq.0 ) then                                ! 35
      IFlag_MT=0
      else if( n.gt.0 ) then                            ! 36
      IFlag_MT=1
      Else                                              ! 37
      IFlag_MT=-1
      endif
      return
      end

```

```

C
C      SUBROUTINE Initial_Mesing_TriLiner
C
C      1ステップの初期化
C
      subroutine Initial_Mesing_TriLiner(Element,Member)
      include "submain.h"
      record / member_s2_Mesing / Member
      record / element_s2 / Element
      integer i                                ! カウンター
      !write(75,*) ' Mesing_TriLiner initial set on '
!エラー入力処理
      if(Element.AK_1.le.0.0) then                ! 38
      Element.AK_1=1.0
      endif
      if(Element.AK_2.le.0.0) then                ! 39
      Element.AK_2=Element.AK_1
      endif
      if(Element.Q_2.le.Element.Q_1) then        ! 40
      Element.Q_2=Element.Q_1
      endif
      Element.U_1=Element.Q_1/Element.AK_1      ! 41
      Element.U_2=(Element.Q_2-Element.Q_1)/Element.AK_2
      *
      +Element.U_1

!フラグの初期化
      do i=1,2                                ! 42
      Member.istat(i) = 10                    ! 43
      Member.AK_tan(i) = Element.AK_1        ! 44
      Member.Uo(i) = Element.U_1             ! 45
      Member.Qo(i) = Element.Q_1             ! 46
      Member.Ur(i) = -Element.U_1            ! 47
      Member.Qr(i) = -Element.Q_1            ! 48

      Member.UoN(i) = -Element.U_2           ! 49
      Member.QoN(i) = -Element.Q_2           ! 50

```

```

Member.UoP(i)    = Element.U_2                ! 51
Member.Qop(i)    = Element.Q_2                ! 52
end do
return
end

```

1. 両方向(v 方向、w 方向)の初期設定を行うかどうかチェックする。状態パラメータが 0 の場合は、初期設定を行う。
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_MesingTriLiner() をコールする。
3. 両方向(v 方向、w 方向)の処理を行う。せん断に関する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_MesingTriLiner() をコールする。
5. 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
6. 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
7. 状態パラメータが絶対値 10、12、21、13、31 で、変位が反転した場合、次の処理を行う
8. 正側の第 1 折れ点の座標をワーク配列 tmp にコピーし、履歴が反転した場合、負側の第 1 折れ点は正側の座標を入れ替えた座標とする。
9. 状態パラメータが 10、-10 の場合、状態パラメータを 1 とし、符号は、以前と反対にする。
10. 状態パラメータが 12、-12 の場合、状態パラメータを 21 とし、符号は、以前と反対にする。
11. 状態パラメータが 21、-21 の場合、状態パラメータを 12 とし、符号は、以前と反対にする。
12. 状態パラメータが 13、-13 の場合、状態パラメータを 31 とし、符号は、以前と反対にする。
13. 状態パラメータが 31、-31 の場合、状態パラメータを 13 とし、符号は、以前と反対にする。
14. 状態パラメータが絶対値 1、2、3 で変位が反転した場合、次の処理を行う
15. サブルーチン Cal_Mesing_TriLiner() で増分前のせん断力を計算する。状態パラメータ 1 で変位が反転した場合、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. 状態パラメータが 2, -2 で変位が反転した場合、状態パラメータを 12 とする。ただし、以前の状態パラメータと逆の符号とする。
17. 目標点の変位とせん断力を構造体成分 `Member.UoN(id)`、`Member.QoN(id)` にセットする。
18. 計算された目標点の変位とせん断力を構造体成分 `Member.UoP(id)`、`Member.QoP(id)` にセットする。
19. 状態パラメータが 3, -3 で変位が反転した場合、状態パラメータを 13 とする。ただし、以前の状態パラメータと逆の符号とする。
20. 反転時の状態パラメータ 1, -1 の目標点の変位とせん断力を計算する。変位とせん断力共に増分前の変位とせん断力に第 1 折れ点の変位とせん断力の 2 倍の値とし、接線剛性を初期剛性の値にセットする。
21. 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 2、10(1)、12、21、13、31 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。
22. 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。
23. 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。
24. 状態変更フラグが 1(境界を超えた)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。
25. 状態パラメータが 10, -10 の場合、まず、接線剛性に第 2 勾配の剛性をセットし、状態パラメータ 2 へ履歴が移る。フラグが 1 の場合、荷重反転時の変位とせん断力の座標に第 1 折れ点の変位とせん断力に変更フラグを、目標点の変位とせん断力に第 2 折れ点の変位とせん断力を計算してセットする。
26. 状態パラメータが 2, -2 の場合、まず、接線剛性に第 3 勾配の剛性をセットし、負側と正側の基本座標を入れ替える。状態パラメータ 3 へ履歴が移る。

27. 状態パラメータが 13, -13 の場合、正側の第 1 折れ点の座標をワーク配列 tmp にコピーする。
28. 目標点の座標は、荷重反転時の座標に第 2 折れ点の 2 倍を加えた値とする。履歴が反転した場合は、座標を入れ替える。接線剛性に第 2 勾配の剛性をセットし、状態パラメータ 2 へ履歴が移る。
29. 状態パラメータが 31, -31 の場合、負側と正側の基本座標を入れ替え、接線剛性に第 3 勾配の剛性をセットし、状態パラメータ 3 へ履歴が移る。
30. 状態パラメータが 12, -12 の場合、負側と正側の基本座標を入れ替え、目標点の変位とせん断力を構造体成分 Member.UoP(id)、Member.QoP(id) にセットする。接線剛性に第 2 勾配の剛性をセットし、状態パラメータ 2 へ履歴が移る。
31. 状態パラメータが 21, -21 の場合、負側と正側の基本座標を入れ替え、目標点の変位とせん断力を構造体成分 Member.UoN(id)、Member.QoN(id) にセットする。接線剛性に第 2 勾配の剛性をセットし、状態パラメータ 2 へ履歴が移る。
32. サブルーチン CalQ_Mesing_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。
33. 状態の直線を決定する式からせん断力 Q を求める。
式は、 $Q = \tan(U - U_r) + Q_r$ となっている。
34. 求めたせん断力 Q を関数名 CalQ_Mesing_TriLiner にセットする。
35. ここでは、 n ステップ時の処理を行う。まず、 $n=0$ であれば変更フラグを IFlag_MT=0 とする。
36. n が正であれば変更フラグを IFlag_MT=1 とする。
37. n が負であれば変更フラグを IFlag_MT=-1 とする。
38. 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は、剛性の値として 1.0 を設定する。
39. 第 2 勾配の剛性が 0 より小さい場合、第 2 勾配の剛性を初期勾配の剛性と同じとする。
40. 第 1 折れ点のせん断力の値が、第 2 折れ点のせん断力の値より小さい場合は、第 2 折れ点のせん断力を第 1 折れ点の値とする。
41. 図 2-1 に示す第 1 折れ点の変位 u_1 と第 2 折れ点の変位 u_2 を計算し、構造体 Element の成分にセットする。
42. 両方向(1=v 方向, 2=w 方向)についてワーク用構造体 Member の成分をセットする。

43. 状態パラメータは初期で 10 に設定
44. 接線剛性は、初期剛性を設定する。
45. $\text{Member.Uo}(i)$ に正側の第 1 折れ点の変位を設定する。
46. $\text{Member.Qo}(i)$ に正側の第 1 折れ点のせん断力を設定する。
47. $\text{Member.Ur}(i)$ に負側の第 1 折れ点の変位を設定する。
48. $\text{Member.Qr}(i)$ に負側の第 1 折れ点のせん断力を設定する。
49. $\text{Member.UoN}(i)$ に負側の第 2 折れ点の変位を設定する。
50. $\text{Member.QoN}(i)$ に負側の第 2 折れ点のせん断力を設定する。
51. $\text{Member.UoP}(i)$ に正側の第 2 折れ点の変位を設定する。
52. $\text{Member.Qop}(i)$ に正側の第 2 折れ点のせん断力を設定する。

本節では、図 2-2 に示すせん断型モデルの履歴ルールの一つ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。この履歴ルールの設定パラメータは、以下のようなものである。

2.2.3 最大点指向型トリリニア

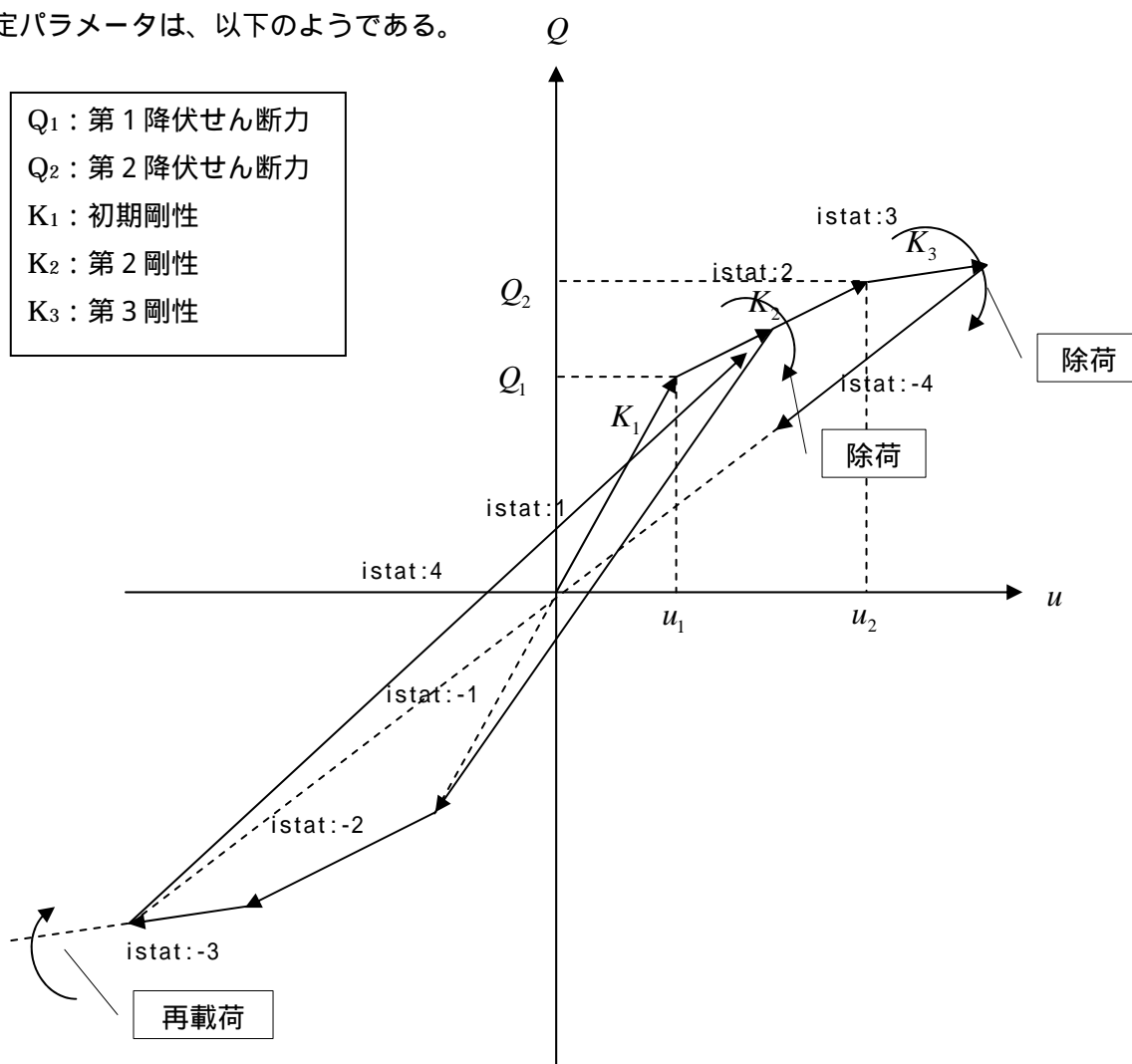


図 2-2 最大点指向型モデルの履歴特性

最大点指向型の履歴ルールを以下のようにまとめる。

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 `DiracMax_TriLiner()` を具体的に示そう。このサブルーチンは、後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るのが特徴である。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 /DiracMax_TriLiner
C
C      Dirac. Max TriLiner 履歴モデル
C
      subroutine DiracMax_TriLiner(Member,Element,vv,vpp)
      implicit real*8(V)
      include "submain.h"
      record / member_s2_DiracMax / 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_DiracMax /
      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_wv(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

```

```

      if( dU*real(Member.istat(id)).lt.0.0 ) then          ! 載荷方向の反転          ! 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_DirectMax_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
      if( Member.Uo(id).gt.U ) then                      ! 21
        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
      endif
    endif
  endif

```

```

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 2-2 に示す第 1 折れ点の変位 u_1 と第 2 折れ点の変位 u_2 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、どちらの方向に変位が進んでいるかチェックする。ここでは、正方向に進んでいたため、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、負方向に進んでいるので、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、ここより以降の処理で、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、接線剛性に第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、接線剛性に第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 $\text{CalQ_DirecMax_TriLiner}()$ 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 2-2 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。

本節では、図 2-3, 4, 5 に示すせん断型モデルの履歴ルールである武田履歴モデルについて説明する。

図中の記号は、以下のようである。

Q_1 : 第 1 降伏せん断力
 Q_2 : 第 2 降伏せん断力
 K_1 : 初期剛性
 K_2 : 第 2 剛性
 K_3 : 第 3 剛性
 : 戻り勾配を決めるためのパラメータ

武田モデル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. 変位 u が最大変位 u_1 より小さい場合には、原点を通る勾配 K_1 の直線上を動く
3. a) u が u_1 を超えた場合は、第 2 勾配 K_2 の直線上を動く。
 b) この直線から折り返した場合は、反対側の第 1 降伏点 Q_1 を目指す

2.2.4 武田モデル

す。

c) 反対側の最大変形が弾性域の場合は、この弾性域の範囲は、反対側の第1降伏点 Q_1 までとする。

d) 反対側の最大変形が u_1 を超えている場合には、弾性域の範囲は復元力が0となる点までとし、0を超えた場合は、反対側の最大変位点を目指す。

e) 最大点を超えた場合は、第2勾配 K_2 の直線上を動く。

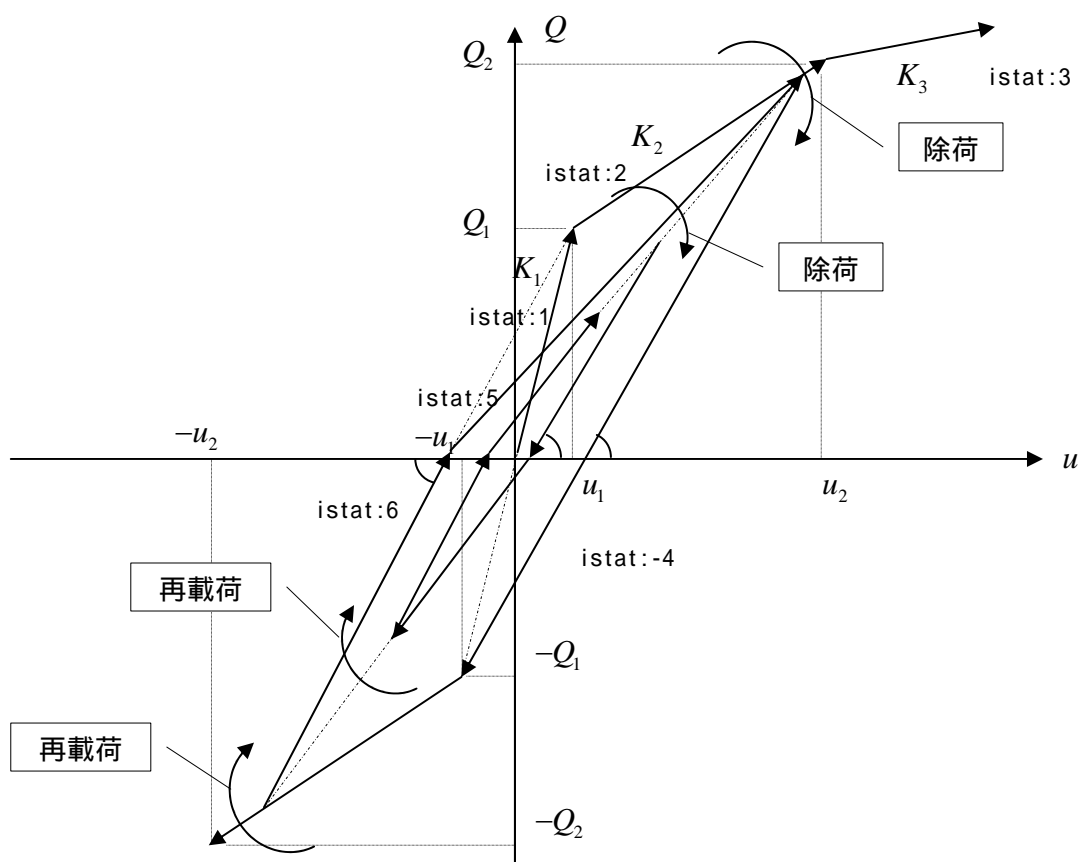


図 2-3 武田モデルの履歴特性(ともに変位が u_2 を超えていない場合)

4. a) u が u_2 を超えた場合は、第3勾配 K_3 の直線上を動く。

b) この直線から折り返した場合は、次式に示す勾配 K_d の直線上を動く。

$$K_d^{\pm} = K_0 \times \left| \frac{u_{\max}^{\pm}}{u_2} \right|^{\alpha}$$

K_d : 戻り勾配 (第2降伏点を超えた場合)
 K_0 : 戻り勾配 (第2降伏点を超えない場合)

c) 反対側の最大変形が u_1 を超えていない場合、勾配 K_d の範囲は、反対側の Q_1 までとし、 Q_1 を超えた場合は第2降伏点 Q_2 を目指す。

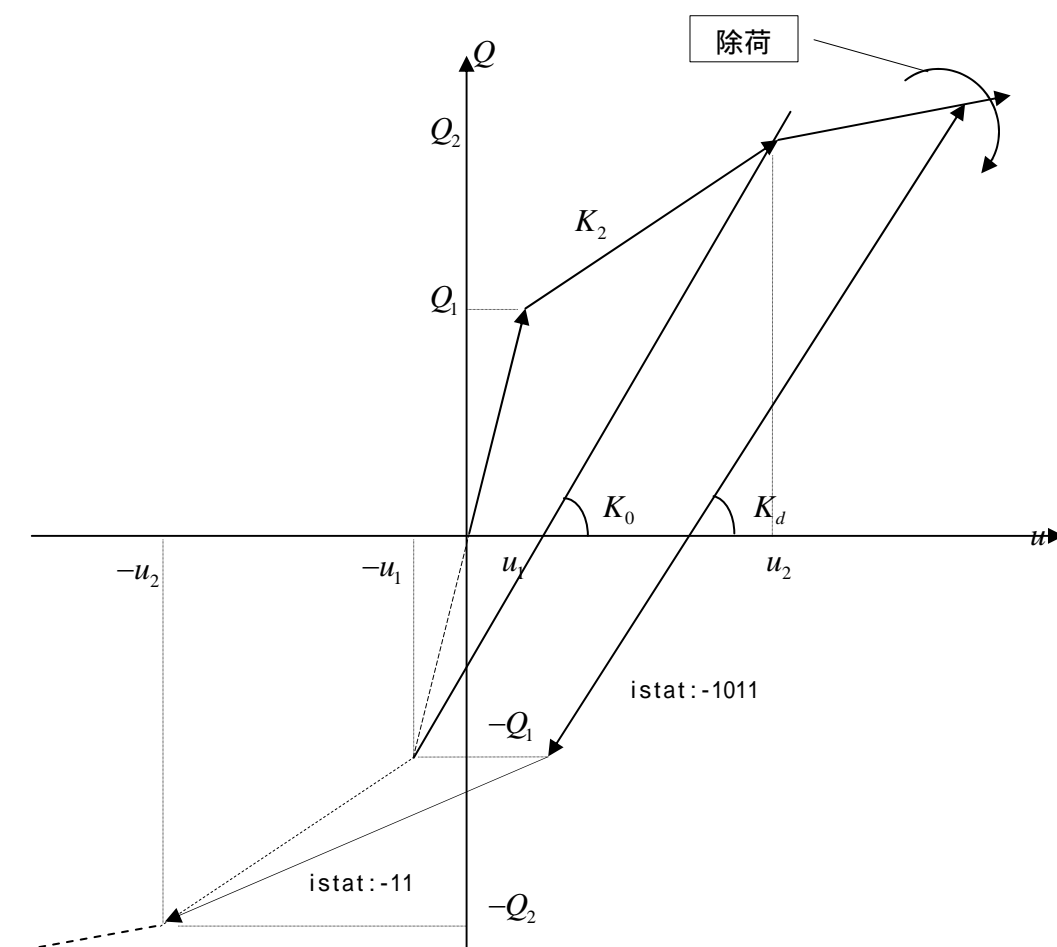


図 2-4 武田モデルの履歴特性(反対側の変位が u_1 を越えていない場合)

- d)c)において、第 降伏点を目指す直線上から折り返した場合は、勾配 K_0 の直線上を動く。さらに復元力が 0 点を超えた場合は、最大変形点を目指す。この最大変形点を目指す直線上から折り返した場合は、釣合点は上式で求められる勾配 K_d 上を動き、復元力が 0 点を超えると、反対側の最大変位点を目指す。
- e) 反対側の最大変位が u_1 を超えた場合は、剛性を勾配 K_0 とし、釣合点が移動可能な範囲は、復元力が 0 となる点までとする。応力が 0 を超えた場合は、過去に経験した最大変位点を目指す。
- f)e)において、最大変位点を目指す直線上から折り返した場合は、勾配 K_d の直線上を動く。さらに復元力が 0 を超える場合は、最大変位点を目指す。この最大変位点を目指す直線上から折り返えす場合も、勾配 K_d 上を動き、復元力が 0 点を超えると、内部ループの最大変位点を目指す。

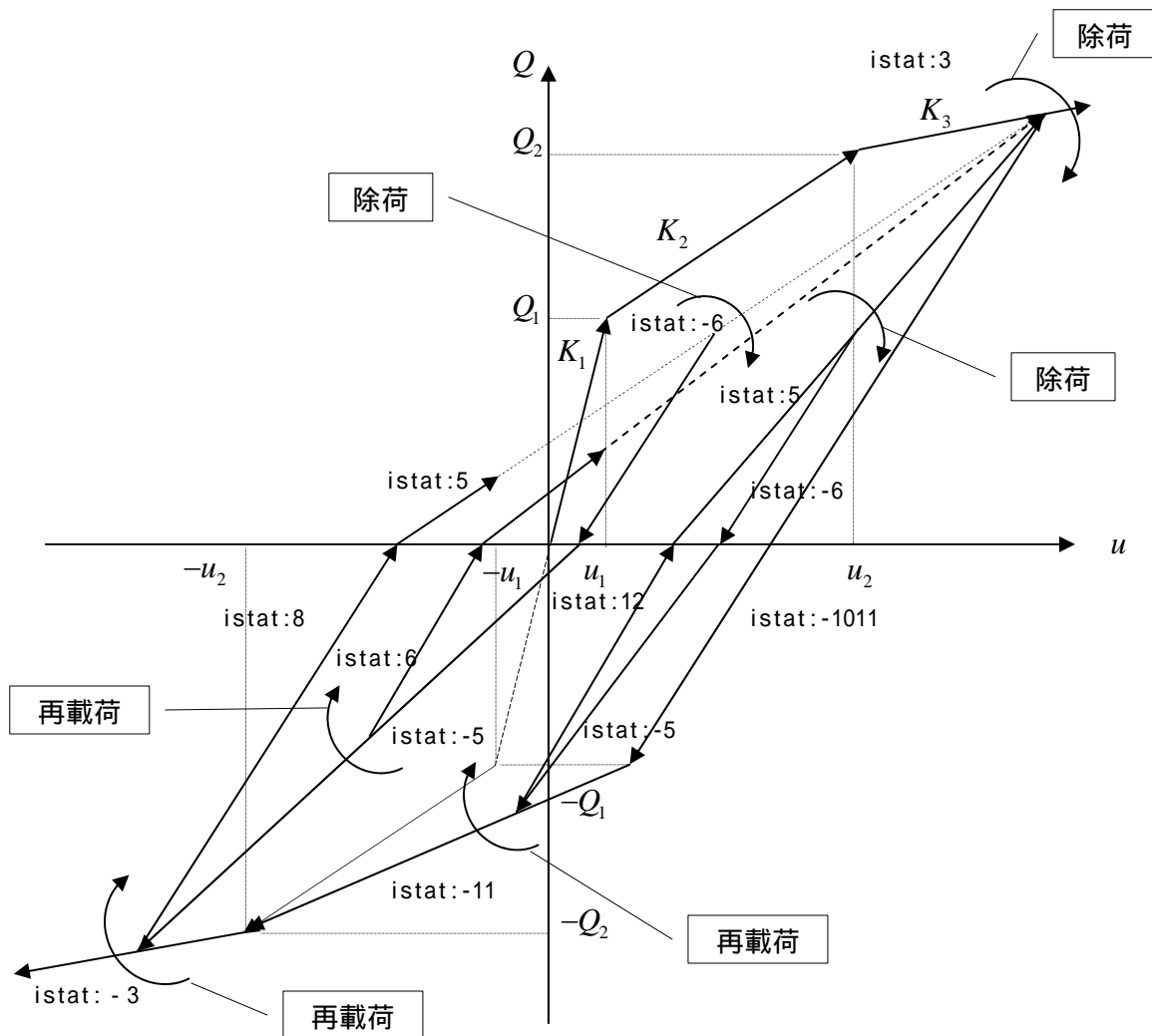


図 2-5 武田モデルの履歴特性(ともに変位が u_2 を越えた場合)

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 `Takeda DTriLiner()` を具体的に示そう。このサブルーチンは、後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るのが特徴である。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勾配で1に、第2勾配で2に、第3勾配で3に、また、諸条件によって変化する状態パラメータについて述べる。

状態パラメータは、接線勾配が K_0 である場合は、4、6とする。共に状態パラメータ2で荷重が除荷されたときに反対側の第1降伏点 Q_1 を目指す。状態6のとき、せん断力が0を超えると最大点を目指す状態パラメータ5に履歴が移る。

接線勾配が K_d である場合は、状態パラメータは 8、12、1003、1011 となる。状態 8、1011 は、状態パラメータ 3 で荷重が除荷されたときに反対側の Q_1 を目指す。一方、状態 8 ではせん断力が 0 を超えると状態 5 に移る。状態 12 では、状態パラメータ 11 で荷重が除荷されたときせん断力 0 を目指す。状態 1003 では 1011 で除荷されたときに履歴が移る。

```

C
C      SUBROUTINE /Takeda_TriLiner
C
C      Takeda DTriLiner 履歴モデル
C
      subroutine Takeda_TriLiner(Member,Element,vv,vpp)
      implicit real*8(V)
      include "submain.h"
      record / member_s2_Takeda / 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 第一ステップ処理
      if(Member.istat(1).eq.0.and.Member.istat(2).eq.0) then ! 1
      call Initial_Takeda_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_Takeda_TriLiner(i, Element, Member, dU, Up) ! 4
      end do
      return
      end

```

```

C
C      SUBROUTINE Cal_Takeda_TriLiner
C
C      n ステップ
C
      subroutine Cal_Takeda_TriLiner(id,Element,Member,dU,Up)
      include "submain.h"
      record / member_s2_Takeda / 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
C 現在の変位の計算
C
      U=Up+dU                                ! 5
C
C 載荷方向の反転
C
      if( dU*real(Member.istat(id)).lt.0.0 ) then          ! 6
        tmp(1)=Member.Uo(id)
        tmp(2)=Member.Qo(id)
        if( iabs(Member.istat(id)).eq.1.or.                ! 7
          * iabs(Member.istat(id)).eq.4.or.
          * iabs(Member.istat(id)).eq.6.or.
          * iabs(Member.istat(id)).eq.8.or.
          * iabs(Member.istat(id)).eq.12 ) then
          Member.Uo(id)=Member.Ur(id)                      ! 8
          Member.Qo(id)=Member.Qr(id)
          Member.Ur(id)=tmp(1)
          Member.Qr(id)=tmp(2)
          Member.istat(id)=-1*Member.istat(id)
        else if( iabs(Member.istat(id)).eq.1011 ) then      ! 9
          Member.Uo(id)=Member.Ur(id)                      ! 10
          Member.Qo(id)=Member.Qr(id)
          Member.Ur(id)=tmp(1)
          Member.Qr(id)=tmp(2)
          if(Member.istat(id).gt.0) then                    ! 11
            Member.istat(id)=-1003
          else                                              ! 12
            Member.istat(id)= 1003
          endif
        else if( iabs(Member.istat(id)).eq.1003 ) then      ! 13
          Member.Uo(id)=Member.Ur(id)                      ! 14
          Member.Qo(id)=Member.Qr(id)
          Member.Ur(id)=tmp(1)
          Member.Qr(id)=tmp(2)
          if(Member.istat(id).gt.0) then                    ! 15
            Member.istat(id)=-1011
          else                                              ! 16
            Member.istat(id)= 1011
          endif
        else if( iabs(Member.istat(id)).eq.2.or.            ! 17
          * iabs(Member.istat(id)).eq.5.or.
          * iabs(Member.istat(id)).eq.11 ) then
C
C 直前の剪断力の計算
C
          Qp=CalQ_Takeda_TriLiner(id,Element,Member,Up)    ! 18
          Member.Ur(id)=Up                                   ! 19
          Member.Qr(id)=Qp

          if( iabs(Member.istat(id)).eq.2 ) then            ! 20
            if( Member.istat(id).gt.0 ) then                ! 21

```

```

        tmp(3) = (-Element.Q_1 - Qp) / (-Element.U_1 - Up)           ! 22
        if( tmp(3).lt.Member.KoM(id).and.
*      Member.UmaxP(id).lt.Element.U_2 ) then                       ! 23
            Member.AK_tan(id) = tmp(3)
            Member.KoM(id)    = tmp(3)
        else                                                         ! 24
            Member.AK_tan(id) = Member.KoM(id)
        endif
        Member.istat(id)=-4
    else                                                             ! 25
        tmp(3) = (Element.Q_1 - Qp) / (Element.U_1 - Up)           ! 26
        if( tmp(3).lt.Member.KoP(id).and.
*      Member.UmaxM(id).gt.-Element.U_2 ) then                       ! 27
            Member.AK_tan(id) = tmp(3)
            Member.KoP(id)    = tmp(3)
        else                                                         ! 28
            Member.AK_tan(id) = Member.KoP(id)
        endif
        Member.istat(id)=4
    endif
else if( iabs(Member.istat(id)).eq.5 ) then                         ! 29
    if( Member.istat(id).gt.0 ) then                                 ! 30
        Member.AK_tan(id) = Member.KoM(id)
        Member.istat(id) = -6
    else                                                             ! 31
        Member.AK_tan(id) = Member.KoP(id)
        Member.istat(id) = 6
    endif
else if( iabs(Member.istat(id)).eq.11) then                         ! 32
    if( Member.istat(id).gt.0 ) then                                 ! 33
        Member.AK_tan(id)=(Element.Q_2+Element.Q_1)
*      /(Element.U_2+Element.U_1)
*      *dabs(Member.UmaxP(id)/Element.U_2)
*      **Element.arf
        Member.KoM(id)=Member.AK_tan(id)
        Member.istat(id) = -12
    else                                                             ! 34
        Member.AK_tan(id)=(Element.Q_2+Element.Q_1)
*      /(Element.U_2+Element.U_1)
*      *dabs(Member.UmaxM(id)/Element.U_2)
*      **Element.arf
        Member.KoP(id)=Member.AK_tan(id)
        Member.istat(id) = 12
    endif
endif
    Member.Uo(id)=Up - Qp/Member.AK_tan(id)                         ! 35
    Member.Qo(id)=0.0
else if(iabs(Member.istat(id)).eq.3) then                           ! 36
C
C 直前の剪断力の計算
C
    Qp=CalQ_Takeda_TriLiner(id,Element,Member,Up)                 ! 37
    Member.Ur(id)=Up                                                ! 38
    Member.Qr(id)=Qp

```

```

        if( Member.istat(id).gt.0 ) then                                ! 39
            Member.AK_tan(id)=(Element.Q_2+Element.Q_1)
            *
            /(Element.U_2+Element.U_1)
            *
            *dabs(Member.UmaxP(id)/Element.U_2)
            *
            **Element.arf
            Member.KoM(id)=Member.AK_tan(id)                            ! 40
            if(dabs(Member.UmaxM(id)).le.Element.U_1) then            ! 41
                Member.istat(id)=-1011
                Member.Qo(id)  =-Element.Q_1
                Member.Uo(id)  =Up-(Qp+Element.Q_1)
            *
                /Member.AK_tan(id)
            else                                                        ! 42
                Member.istat(id)=-8
                Member.Uo(id)=Up - Qp/Member.AK_tan(id)
                Member.Qo(id)=0.0
            endif
        else                                                            ! 43
            Member.AK_tan(id)=(Element.Q_2+Element.Q_1)
            *
            /(Element.U_2+Element.U_1)
            *
            *dabs(Member.UmaxM(id)/Element.U_2)
            *
            **Element.arf
            Member.KoP(id)=Member.AK_tan(id)                            ! 44
            if(Member.UmaxP(id).le.Element.U_1) then                  ! 45
                Member.istat(id)=1011
                Member.Qo(id)  =Element.Q_1
                Member.Uo(id)  =Up-(Qp-Element.Q_1)
            *
                /Member.AK_tan(id)
            else                                                        ! 46
                Member.istat(id)=8
                Member.Uo(id)=Up - Qp/Member.AK_tan(id)
                Member.Qo(id)=0.0
            endif
        endif
    endif
    return
endif
C
C  接線剛性の変化のチェック
C
    if( iabs(Member.istat(id)).ne.3 ) then                            ! 47
        if( Member.istat(id).gt.0 ) then                              ! 48
            if( Member.Uo(id).lt.U ) then
                iflag=1
            else
                iflag=0
            endif
        else                                                            ! 49
            if( Member.Uo(id).gt.U ) then
                iflag=1
            else
                iflag=0
            endif
        endif
    endif
    if( iflag.eq.1 ) then                                              ! 50

```

```

Member.Ur(id)=Member.Uo(id)
Member.Qr(id)=Member.Qo(id)
if( iabs(Member.istat(id)).eq.1 ) then                                ! 51
    Member.AK_tan(id)=Element.AK_2
    if ( Member.istat(id).gt.0 ) then                                ! 52
        Member.istat(id)= 2
        Member.Uo(id)  = Element.U_2
        Member.Qo(id)  = Element.Q_2
    Else                                                            ! 53
        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.or.                            ! 54
*   iabs(Member.istat(id)).eq.11 ) then
    Member.AK_tan(id)=Element.AK_3
    if ( Member.istat(id).gt.0 ) then                                ! 55
        Member.istat(id)=3
    else                                                            ! 56
        Member.istat(id)=-3
    endif
else if( iabs(Member.istat(id)).eq.4.or.                            ! 57
*   iabs(Member.istat(id)).eq.6 ) then
    if( iabs(Member.istat(id)).eq.4.and.
*   Member.Qo(id).ne.0.0 ) then                                ! 58
        Member.AK_tan(id)=Element.AK_2
    if ( Member.istat(id).gt.0 ) then                                ! 59
        Member.istat(id)= 2
        Member.Uo(id)  = Element.U_2
        Member.Qo(id)  = Element.Q_2
    else                                                            ! 60
        Member.istat(id)=-2
        Member.Uo(id)  =-Element.U_2
        Member.Qo(id)  =-Element.Q_2
    endif
else                                                                ! 61
    if ( Member.istat(id).gt.0 ) then                                ! 62
        Member.istat(id)= 5
        Member.Uo(id)  = Member.UmaxP(id)
        Member.Qo(id)  = Member.QmaxP(id)
    else                                                            ! 63
        Member.istat(id)=-5
        Member.Uo(id)  = Member.UmaxM(id)
        Member.Qo(id)  = Member.QmaxM(id)
    endif
    Member.AK_tan(id)=(Member.Qo(id) - Member.Qr(id))
*   /(Member.Uo(id) - Member.Ur(id))
endif
else if( iabs(Member.istat(id)).eq.8 ) then                        ! 64
    if( Member.Qo(id).ne.0.0 ) then                                ! 65
        Member.AK_tan(id)=Element.AK_3
    if ( Member.istat(id).gt.0 ) then                                ! 66
        Member.istat(id)= 3
    else                                                            ! 67

```

```

        Member.istat(id)=-3
    endif
    else ! 68
        if ( Member.istat(id).gt.0 ) then ! 69
            Member.istat(id)= 5
            Member.Uo(id) = Member.UmaxP(id)
            Member.Qo(id) = Member.QmaxP(id)
        else ! 70
            Member.istat(id)=-5
            Member.Uo(id) = Member.UmaxM(id)
            Member.Qo(id) = Member.QmaxM(id)
        endif
        Member.AK_tan(id)=(Member.Qo(id) - Member.Qr(id))
        * / (Member.Uo(id) - Member.Ur(id))
    endif
    else if( iabs(Member.istat(id)).eq.1011 ) then ! 71
        if(Member.istat(id).gt.0) then ! 72
            Member.istat(id)=11

            Member.Uo(id)=Element.U_2
            Member.Qo(id)=Element.Q_2
        else ! 73
            Member.istat(id)=-11

            Member.Uo(id)=-Element.U_2
            Member.Qo(id)=-Element.Q_2
        endif
        Member.AK_tan(id)=(Member.Qo(id) - Member.Qr(id))
        * / (Member.Uo(id) - Member.Ur(id))
    else if( iabs(Member.istat(id)).eq.1003 ) then ! 74
        Member.AK_tan(id)=Element.AK_3
        if(Member.istat(id).gt.0) then ! 75
            Member.istat(id)=3
        else ! 76
            Member.istat(id)=-3
        endif
    else if( iabs(Member.istat(id)).eq.5 ) then ! 77
        if(dabs(Member.Ur(id)).lt.Element.U_2) then ! 78
            Member.AK_tan(id)=Element.AK_2
            if ( Member.istat(id).gt.0 ) then ! 79
                Member.istat(id)= 2
                Member.Uo(id) = Element.U_2
                Member.Qo(id) = Element.Q_2
            else ! 80
                Member.istat(id)=-2
                Member.Uo(id) =-Element.U_2
                Member.Qo(id) =-Element.Q_2
            endif
        else ! 81
            Member.AK_tan(id)=Element.AK_3
            if ( Member.istat(id).gt.0 ) then ! 82
                Member.istat(id)= 3
            else ! 83
                Member.istat(id)=-3
            endif
        endif
    endif

```

```

        endif
    endif
    else if( iabs(Member.istat(id)).eq.12 ) then                ! 84
        if( Member.Qo(id).ne.0.0 ) then                        ! 85
            if( Member.istat(id).gt.0 ) then                    ! 86
                Member.istat(id)=11
                Member.Uo(id)=Element.U_2
                Member.Qo(id)=Element.Q_2
            else                                                ! 87
                Member.istat(id)=-11
                Member.Uo(id)=-Element.U_2
                Member.Qo(id)=-Element.Q_2
            endif
            Member.AK_tan(id)=(Member.Qo(id) - Member.Qr(id))
            *                                                    / (Member.Uo(id) - Member.Ur(id))
        else                                                    ! 88
            if ( Member.istat(id).gt.0 ) then                    ! 89
                Member.istat(id)= 5
                Member.Uo(id)  = Member.UmaxP(id)
                Member.Qo(id)  = Member.QmaxP(id)
            else                                                ! 90
                Member.istat(id)=-5
                Member.Uo(id)  = Member.UmaxM(id)
                Member.Qo(id)  = Member.QmaxM(id)
            endif
            Member.AK_tan(id)=(Member.Qo(id) - Member.Qr(id))
            *                                                    / (Member.Uo(id) - Member.Ur(id))
        endif
    endif
    Member.stress(1+id)
    *
    =CalQ_Takeda_TriLiner(id,Element,Member,U)
    endif
endif
C
C 現在の剪断力の計算
C
    Q=CalQ_Takeda_TriLiner(id,Element,Member,U)                ! 91
C
C 最大振幅値のチェック
C
    if( Member.istat(id).gt.0 ) then                            ! 92
        if( Member.UmaxP(id).lt.U ) then                        ! 93
            Member.UmaxP(id) = U
            Member.QmaxP(id) = Q
        endif
    else                                                        ! 94
        if( Member.UmaxM(id).gt.U ) then
            Member.UmaxM(id) = U
            Member.QmaxM(id) = Q
        endif
    endif
endif
return
end

```

```

C
C      real*8 function CalQ_Takeda_TriLiner
C
C      変位 U に対応する剪断力 Q の計算
C
      real*8 function CalQ_Takeda_TriLiner(id,Element,Member,U)
      include "submain.h"
      record / member_s2_Takeda / 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
      if( iabs(Member.istat(id)).eq.1) then                ! 95
        Q=Element.AK_1*U
      else if( iabs(Member.istat(id)).eq.2) then           ! 96
        if( Member.istat(id).gt.0 ) then                   ! 97
          Q=Element.AK_2*(U-Element.U_1) +Element.Q_1
        else                                               ! 98
          Q=Element.AK_2*(U+Element.U_1) -Element.Q_1
        endif
      else if( iabs(Member.istat(id)).eq.3) then           ! 99
        if( Member.istat(id).gt.0 ) then                   ! 100
          Q=Element.AK_3*(U-Element.U_2) +Element.Q_2
        else                                               ! 101
          Q=Element.AK_3*(U+Element.U_2) -Element.Q_2
        endif
      else                                               ! 102
        Q=Member.AK_tan(id)*(U-Member.Ur(id)) +Member.Qr(id)
      endif
      CalQ_Takeda_TriLiner=Q                                ! 103
      return
      end

```

```

C
C      SUBROUTINE Initial_Takeda_TriLiner
C
C      1 ステップの初期化
C
      subroutine Initial_Takeda_TriLiner(Element,Member)
      include "submain.h"
      record / member_s2_Takeda / Member
      record / element_s2      / Element
      integer      i              ! カウンター
      !write(75,*) ' Takeda_TriLiner initial set on '
C
C      エラー入力処理
C
      if(Element.AK_1.le.0.0) then
        Element.AK_1=1.0                ! 104
      endif
      if(Element.AK_2.le.0.0) then       ! 105
        Element.AK_2=Element.AK_1
      endif
      if(Element.Q_2.le.Element.Q_1) then ! 106

```

```

        Element.Q_2=Element.Q_1
    endif
C
C   フラグの初期化
C
    do i=1,2                                ! 107
        Member.istat(i) = 1                  ! 108
        Member.AK_tan(i) = Element.AK_1      ! 109
        Member.Uo(i)    = Element.Q_1/Element.AK_1 ! 110
        Member.Qo(i)    = Element.Q_1        ! 111
        Member.Ur(i)    = -Element.Q_1/Element.AK_1 ! 112
        Member.Qr(i)    = -Element.Q_1        ! 113
        Member.UmaxP(i) = Element.Q_1/Element.AK_1 ! 114
        Member.QmaxP(i) = Element.Q_1        ! 115
        Member.UmaxM(i) = -Element.Q_1/Element.AK_1 ! 116
        Member.QmaxM(i) = -Element.Q_1        ! 117
        Member.KoP(i)   = Element.AK_1        ! 118
        Member.KoM(i)   = Element.AK_1        ! 119
    end do
    Element.U_1=Element.Q_1/Element.AK_1      ! 120
    Element.U_2=(Element.Q_2-Element.Q_1)/Element.AK_2
    *      +Element.U_1
    return
end

```

1. 両方向(v 方向、w 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_Takeda_TriLiner()をコールする。
3. 両方向(v 方向、w 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_Takeda_TriLiner()をコールする。
5. 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。
6. 状態パラメータの方向と増分変位の関係を調べ、変位が反転するとき、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。
7. 状態パラメータが絶対値 1,4,6,8,12 の場合は、次の処理を行う。
8. 正側と負側の基本座標を入れ替え、状態パラメータを-1 とする。
9. 状態パラメータが 1011, -1011 の場合は、次の処理を行う。
10. 正側と負側の基本座標を入れ替える。
11. 増分前の状態で、どちらの方向に変位が進んでいるかをチェックする。ここでは、正方向に進んでいたのので、状態パラメータを-1003

- にセットする。
12. ここでは、負方向に進んでいるので、状態パラメータを 1003 にセットする。
 13. 状態パラメータが 1003, -1003 の場合は、次の処理を行う。
 14. 正側と負側の基本座標を入れ替える。
 15. 増分前の状態で、どちらの方向に変位が進んでいるかをチェックする。ここでは、正方向に進んでいたなので、状態パラメータを -1011 にセットする。
 16. ここでは、負方向に進んでいるので、状態パラメータを 1011 にセットする。
 17. 状態パラメータが絶対値 2, 5, 11 の場合は、次の処理を行う。
 18. サブルーチン `Cal_DirectMax_TriLiner()` を用いて、増分前のせん断力を計算する。
 19. 反転位置の座標 (U_p, Q_p) を構造体成分 `Member.Ur(id)`、`Member.Qr(id)` にセットする。
 20. 状態パラメータが 2, -2 の場合は、次の処理を行う。
 21. 増分前の状態で、どちらの方向に変位が進んでいるかをチェックする。ここでは、正方向に進んでいるので、状態パラメータを -4 にセットする。
 22. 接線剛性を負側の第 1 折れ点と増分前の変位とせん断力から計算し、ワーク配列 `tmp(3)` に保存する。
 23. ワーク配列 `tmp(3)` が負の戻り勾配より小さく、正側の最大変位が、正側の第 2 折れ点の変位 u_2 を超えていなければ、接線剛性と負の戻り勾配は、ワーク配列 `tmp(3)` で計算された値と同じに設定する。
 24. ワーク配列 `tmp(3)` が負の戻り勾配より大きく、正側の最大変位が正側の第 2 折れ点の変位 u_2 を超えている場合、接線剛性を負側の戻り勾配として設定する。
 25. ここでは、負方向に進んでいるので、状態パラメータを 4 にセットする。
 26. 接線剛性を正側の第 1 折れ点と増分前の変位とせん断力から計算し、ワーク配列 `tmp(3)` に保存する。
 27. ワーク配列 `tmp(3)` が正の戻り勾配より小さく、負側の最大変位が、負側の第 2 折れ点の変位 u_2 を超えていなければ、接線剛性と正の戻り勾配は、ワーク配列 `tmp(3)` で計算される値と同じとして設定する。
 28. ワーク配列 `tmp(3)` が正の戻り勾配より大きく、負側の最大変位が負側の第 2 折れ点の変位 u_2 を超えている場合、接線剛性を正側の戻り

勾配に設定する。

29. 状態パラメータが 5, -5 の場合は、次の処理を行う。
30. 増分前の状態で、どちらの方向に変位が進んでいるかをチェックする。ここでは、正方向に進んでいるので、状態パラメータを -6 にセットする。接線剛性の値を $\text{Member.KoM}(\text{id})$ とする。
31. ここでは、負方向に進んでいるので、状態パラメータを 6 にセットする。接線剛性の値を $\text{Member.KoP}(\text{id})$ とする。
32. 状態パラメータが 11, -11 の場合は、次の処理を行う。
33. どちらに変位が進んでいるかをチェックする。ここでは、荷重が反転したときは、状態パラメータを -12 にセットする。接線剛性の値は、計算によって求める。
34. ここでは、再載荷なので状態パラメータを 12 にセットする。接線剛性の値は、計算によって求める。
35. 目標の変位は、増分前の変位とせん断力、接線剛性より求め、せん断力には、0.0 をセットする。
36. 状態パラメータが 3, -3 から履歴が反転するときは、次の処理を行う。
37. 最大変位と最大せん断力をセットするために、現時点でのせん断力を求める。
38. 反転位置の座標 (U_p, Q_p) を構造体 $\text{Member.Ur}(\text{id})$ 、 $\text{Member.Qr}(\text{id})$ にセットする。
39. 増分前の状態で、どちらの方向に変位が進んでいるかをチェックする。ここでは、正方向に進んでいるので、これ以降の処理で、接線剛性や状態パラメータの変更を行う。
40. 負側の戻り勾配を計算された接線剛性に設定する。
41. 負側の最大変位が変位 u_1 を超えていなければ、状態パラメータを -1011 にセットし、目標のせん断力を負側の第 1 折れ点に、変位を増分前の変位とせん断力、正側の第 1 折れ点、計算された接線剛性より求める。
42. 負側の最大変位が変位 u_1 を超えていれば、状態パラメータを -8 にセットし、目標の変位を増分前の変位とせん断力、計算された接線剛性より求める。
43. ここでは、負方向に進んでいるので、これ以降の処理で、接線剛性や状態パラメータの変更を行う。
44. 正側の戻り勾配を計算された接線剛性に設定する。
45. 正側の最大変位が変位 u_1 を超えていなければ、状態パラメータを 1011 にセットし、目標のせん断力を正側の第 1 折れ点とし、変位を

- 増分前の変位とせん断力、負側の第1折れ点、計算された接線剛性より求める。
46. 正側の最大変位が変位 u_1 を超える場合、状態パラメータを8にセットし、目標の変位を増分前の変位とせん断力、計算された接線剛性より求め、せん断力には0.0をセットする。
47. 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、つまり状態1、2、4、5、6、8、12、1003、1011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、境界を越えるかどうかのチェックを必要としない。
48. 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 $\text{Member.Uo}(id)$ 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 $\text{iflag}=1$ とする。変位が境界を超えない場合は、 $\text{iflag}=0$ とする。
49. 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 $\text{Member.Uo}(id)$ 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 $\text{iflag}=1$ とする。変位が境界を超えない場合は、 $\text{iflag}=0$ とする。
50. 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。
51. 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。
52. 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。
53. 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。
54. 状態パラメータが絶対値2か11の場合、まず、接線剛性に第3勾配の剛性をセットする。
55. 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。
56. 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。
57. 状態パラメータが4か6の場合、まず、接線剛性に第2勾配の剛性をセットする。
58. 状態パラメータが4でせん断力が負の場合、次の処理を行う。
59. 変位の進行方向が、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。
60. 変位の進行方向が、負方向の場合は状態パラメータを-2にセットし、境界条件として負側の第2折れ点の座標をセットする。
61. 状態パラメータが6の場合、次の処理を行う。
62. 変位の進行方向が、正方向の場合は、状態パラメータを5にセットし、境界条件として正側の最大変位の座標をセットする。
63. 変位の進行方向が、負方向の場合は、状態パラメータを-5にセットし、境界条件として負側の最大変位の座標をセットする。
64. 状態パラメータが8の場合は、まず、接線剛性に第3勾配をセットする。
65. せん断力が負側であれば、接線剛性に第3勾配を設定する。
66. 変位の進行方向が、正方向の場合は、状態パラメータを3にセットする。
67. 変位の進行方向が、負方向の場合は、状態パラメータを-3にセットする。
68. 増分前のせん断力が、正側の場合は、次の処理を行う。
69. 変位の進行方向が、正方向の場合は、状態パラメータを5にセットし、境界条件として正側の最大変位の座標をセットする。
70. 変位の進行方向が、負方向の場合は、状態パラメータを-5にセットし、境界条件として負側の最大変位の座標をセットする。
71. 状態パラメータが1011の場合、次の処理を行う。
72. 変位の進行方向が、正方向の場合は、状態パラメータを11にセットし、境界条件として正側の第2折れ点の座標をセットする。
73. 変位の進行方向が、負方向の場合は、状態パラメータを-11にセットし、境界条件として負側の第2折れ点の座標をセットする。
74. 状態パラメータが1003の場合は、まず接線剛性に第3勾配をセットする。
75. 変位の進行方向が、正方向の場合は、状態パラメータを3にセットする。
76. 変位の進行方向が、負方向の場合は、状態パラメータを-3にセットする。
77. 状態パラメータが5の場合、次の処理を行う。
78. 変位が第2折れ点の変位 u_2 を超えていなければ、接線剛性に第2勾配を設定する。
79. 変位の進行方向が、正方向の場合は、状態パラメータを2にセットし、境界条件として正側の第2折れ点の座標をセットする。

80. 変位の進行方向が、負方向の場合は、状態パラメータを-2 にセットし、境界条件として負側の第2折れ点の座標をセットする。
81. 変位が第2折れ点を超える場合、接線剛性に第3勾配をセットする。
82. 変位の進行方向が、正方向の場合は、状態パラメータを3 にセットする。
83. 変位の進行方向が、負方向の場合は、状態パラメータを-3 にセットする。
84. 状態パラメータが12の場合は、次の処理を行う。
85. 増分前のせん断力が、負側の場合は、次の処理を行う。
86. 変位の進行方向が、正方向の場合は、状態パラメータを11 にセットし、境界条件として正側の第2折れ点の座標をセットする。
87. 変位の進行方向が、負方向の場合は、状態パラメータを-11 にセットし、境界条件として負側の第2折れ点の座標をセットする。
88. 増分前のせん断力が、正側の場合は、次の処理を行う。
89. 変位の進行方向が、正方向の場合は、状態パラメータを5 にセットし、境界条件として正側の最大変位の座標をセットする。
90. 変位の進行方向が、負方向の場合は、状態パラメータを-5 にセットし、境界条件として負側の最大変位の座標をセットする。
91. 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。
92. 変位の進行方向をチェックする。
93. 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。
94. 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。
95. 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2-4 参照)。状態パラメータが1の場合は、線形剛性に変位を掛けて求める。
96. 状態パラメータが絶対値で2の場合の処理を行う。
97. 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。
98. 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。
99. 状態パラメータが絶対値で3の場合の処理を行う。
100. 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。
101. 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。

102. 状態パラメータが-4の場合、図2-4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。
103. 求めたせん断力 Q を関数名 `CalQ_DirecMax_TriLiner` にセットする。
104. 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。
105. 第2剛性が適切でない場合、第2剛性は初期剛性と同じとする。
106. 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。
107. $i=1$ の場合は、 v 方向の計算を行う。 $i=2$ の場合は w 方向の計算を行
う。
108. 状態パラメータ1の場合を設定する。
109. 接線剛性に初期剛性をセットする。
110. 正側の直線式の変位
111. 正側の直線式のせん断力
112. 負側の直線式の変位
113. 負側の直線式のせん断力
114. 正側の最大変位点の初期設定
115. 正側の最大応力点の初期設定
116. 負側の最大変位点の初期設定
117. 負側の最大応力点の初期設定
118. 正側の戻り勾配の初期設定
119. 負側の戻り勾配の初期設定
120. 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。

2.2.5 木質パネル

ここでは、せん断型履歴モデルとして木質パネルに用いる履歴モデルについて説明する。木質パネル用履歴モデルは、以下の2種類である。2.2.5.2の3点折れのS字型スリップモデルは、付3にその解析例を載せてあるので、ぜひ参考にされたい。

2.2.5.1 S字型スリップバイリニア

2.2.5.2 3点折れのS字型スリップモデル

本節では、せん断型履歴ルールで、図2-6に示すS字型スリップバイリニアについて説明する。図中の記号は以下のようなものである。

第1勾配	: AK_1	第2勾配	: AK_2
第3勾配	: AK_3	第1折れ点のせん断力	: Q_1

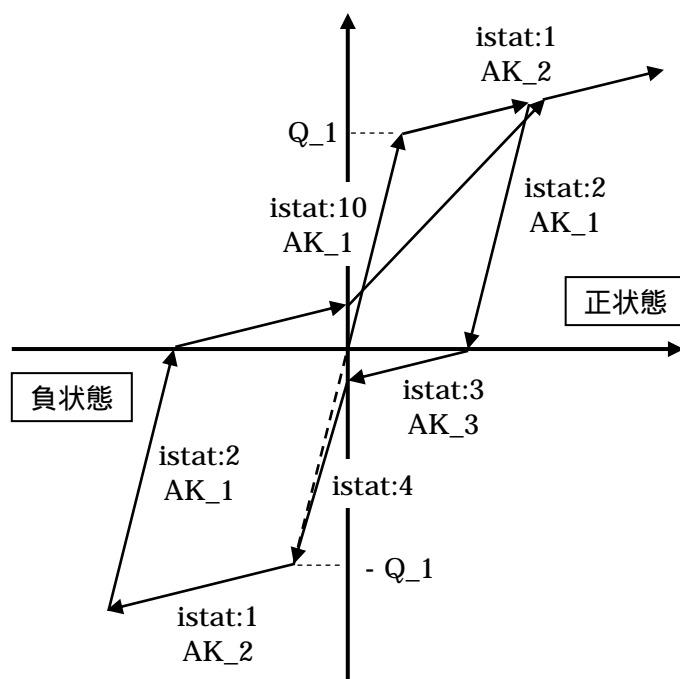
2.2.5.1 S字型
スリップ
バイリニア

図2-6 S字型スリップバイリニア履歴特性

この履歴特性は、木質パネルの抵抗機構をせん断要素でモデル化するために作成したものであるが、コンクリートのせん断型の履歴としても用いられることがある。

正負対称のバイリニアの骨格曲線であり、スリップ時の剛性がユーザー側で指定できるモデルである。弾性状態は、istat:10であり、istat:2

は、除荷された時の状態を表わしている。降伏以降の状態は、 $istat:1$ で表わしており、 $istat:3$ は、残留変位によるスリップ状態を表わす。 $istat:4$ では、スリップの状態の変位がゼロとなる座標から最高点を狙って進行する履歴を描く。

ここで、図 2-6 を用いてこの履歴を説明する。この履歴特性は正負対象の履歴であるため、正方向の境界値 Q_1 を越えて進行する履歴について説明する。初期では、状態 10 であり、はじめの像分変位が正方向に進行し、境界値 Q_1 を越えると状態 1 となる。状態 1 の正方向の境界値は無く、増分変位が反転し負方向に進むと状態 2 となる。除荷時のせん断力を $P1$ として $0 < Q < P1$ を境界値として進行する。増分変位が負方向に進み境界値 0 を越えると状態 3 になり、スリップを表わす履歴となる。状態 3 は、負方向に進行する場合は、変位がゼロを超えると状態 4 となり、一度も負方向の履歴を経験していなければ負方向の境界値を $Q > -Q_1$ 、経験していれば $Q > -P2$ の範囲で直線に進行する。状態 4 は負方向に増分変位が進行する場合、負側の最大点座標 ($-du4$ 、 $-Q_1$) もしくは ($-du4$ 、 $-P2$) を指向する勾配を持って進行する。状態 4 において不法区の境界値を越えると状態 1 に進行する。この場合の状態 1 は負方向の境界値は無く、正方向に増分変位が反転すると状態 2 となり、除荷時のせん断力を $-P2$ として、 $-P2 < Q < 0$ の範囲内で履歴を描く。正方向の境界値を越えると状態 3 となり、スリップを表わす履歴に進む。状態 3 では、先に述べた場合と同様な境界値を得て、正方向の境界値を越えると状態 4 となり、正側の最大点座標 ($du3$ 、 $P1$) を指向する履歴となる。

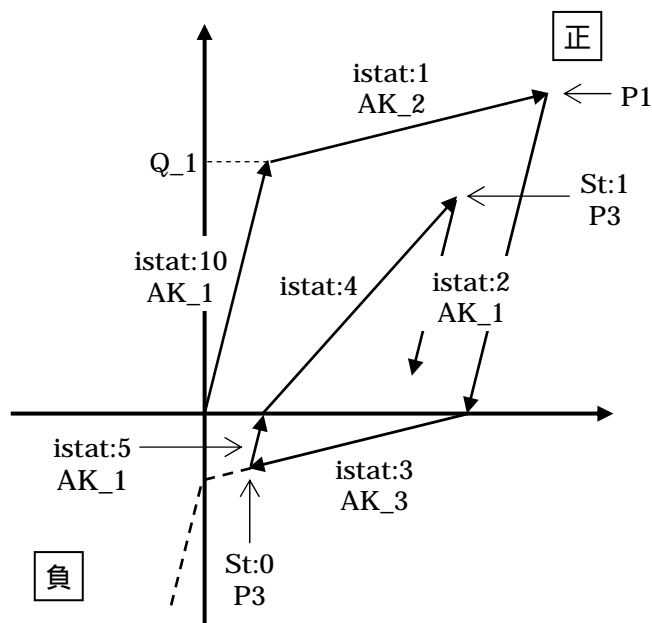


図 2-7 状態 3、状態 4 から除荷したときの履歴

次に、図 2-7 を参考に状態 3 より除荷され状態 5 に進行する履歴について説明する。状態 3 より除荷した場合、除荷したときのせん断力を P_3 として状態 5 の境界値とする。状態 5 の境界値は $0 > Q > P_3$ となり、除荷時の状態を $st:0$ として状態 5 を進む。状態 5 で境界値 0 を越えると状態 4 に進み正側の最大点を指向する履歴を描く。この時の状態 4 で除荷されると除荷したときのせん断力を P_3 として状態 2 に進行し、 $P_3 > Q > 0$ の範囲内で履歴を描く。状態 2 以降は先に述べた履歴と同様である。

状態 4 より除荷され状態 2 もしくは状態 5 に進行する履歴について説明する。状態 4 より除荷した場合、除荷したときのせん断力を P_3 として状態 2、5 の境界値とする。状態 4 から除荷した場合、2 通りの履歴を描く。1 つ目は除荷されせん断力が 0 に到達した時に残留変位が残りスリップが生じる場合であり、2 つ目は残留変位が残らずスリップが生じない場合である。したがって、ここではスリップが生じる場合は状態 4 で除荷されると状態 2 に進行し、スリップが生じない場合は状態 5 に進行する履歴となる。以下に 2 通りの履歴について説明する。

スリップが生じる場合 状態 4 で除荷される時、除荷勾配を用いて残留変位の有無を検討し、スリップが生じるとした時、状態 2 に進行する。除荷した時のせん断力を P_3 とし、除荷時の状態を $st:1$ とする。状態 2 の境界値は $P_3 > Q > 0$ とし、境界値 P_3 を越える場合は状態 4 に戻り、境界値 0 を越える場合はスリップを表す状態 3 に進行する。

スリップが生じない場合 状態 4 で除荷された時に除荷勾配を用いて残留変位の有無を検討し、スリップが生じないとした時、状態 5 に進行する。除荷した時のせん断力を P_3 とし、除荷時の状態を $st:1$ とする。状態 5 の境界値は $P_3 > Q > 0$ とし、境界値 P_3 を越える場合は状態 4 に戻り、境界値 0 を越える場合は負側の最大点を指向する状態 4 に進行する。

以上の説明をまとめると、S 字型の対称スリップバイリニア履歴特性は、次の 5 つの状態で表現される。

istat:10 弾性状態であり、剛性は AK_1 である。 $Q_1 > Q > -Q_1$ の中で動き、 Q_1 、 $-Q_1$ を越えると状態 1 となる。
istat:1 正負共に第 2 勾配であり、剛性は AK_2 である。せん断力が

正の場合、増分変位が正方向に進行する時その上限はなく、除荷されて増分変位が反転し負方向に進行すると、状態 2 となり、反転した時の増分前のせん断力を $P1$ にする。せん断力が負の場合は、増分変位が負方向に進行する時その上限はなく、除荷されて増分変位が反転し正方向に進行すると、状態 2 となって反転した時の増分前のせん断力を $P2$ とする。

istat:2 状態 1 で除荷された時の状態であり、剛性は AK_1 である。せん断力が正の場合は $P1 > Q > 0$ の中で動き、正方向に増分変位が進み、 $P1$ を越えると状態 1 に進む。負方向に増分変位が進み、0 を越えると状態 3 に進む。せん断力が負の場合は $0 > Q > -P2$ の中で動き、負方向に増分変位が進み、 $-P2$ を越えると状態 1 に進む、正方向に増分変位が進み、ゼロを越えると状態 3 に進む。

istat:3 スリップ時の勾配であり、剛性は AK_3 である。状態 3 における境界条件は変位 u によって得られており、せん断力が正の場合は増分変位が正方向の時、その境界は $0 > u$ であり、ゼロを越えると状態 4 となる。増分変位が反転し負方向になった時、状態 5 となり除荷時の状態を表す $St : 0$ にして増分前のせん断力を $P3$ とする。せん断力が負の場合は増分変位が負方向の時、その境界は $u > 0$ であり、ゼロを越えると状態 4 となる。増分変位が反転し正方向になった時、状態 5 となり除荷時の状態を表す $st:0$ にして増分前のせん断力を $P3$ とする。

istat:4 正負共に最大点を指向する勾配であり、せん断力が正の場合は増分変位が正方向の時その境界は $P1 > Q$ であり、増分変位が反転し負方向となった時、除荷時の状態を表す $St : 1$ にして増分前のせん断力を $P3$ とし状態 2、5 に進行する。状態 2 もしくは 5 に進行する条件としては、先に述べたとおりである。せん断力が負の場合は増分変位が負方向の時、その境界は $Q > -P2$ であり、増分変位が反転し正方向となった時、除荷時の状態を表す $St : 1$ にして増分前のせん断力を $P3$ とし状態 2、5 に進行する。状態 2 もしくは 5 に進行する条件としては、先に述べた。

istat:5 状態 3、4 から除荷する時の状態であり、剛性は AK_1 である。せん断力が正の場合は $P3 > Q > 0$ の中で動き、正方向に増分変位が進み、 $P3$ を越えると $St : 0$ の時は状態 3 に進行し、 $St : 1$ の時は状態 4 に進行する。また、負方向に増分変位が進み、

ゼロを越えると状態 4 に進む。せん断力が負の場合は $0 > Q > P3$ の中で動き、負方向に増分変位が進み、 $P3$ を越えると $St : 0$ の時は状態 3 に進行し、 $St : 1$ の時は状態 4 に進行する。正方向に増分変位が進みゼロを越えると状態 4 に進む。

以上で各状態の説明を終える。次に、この履歴ルールを表現するサブルーチンを具体的に説明しよう。まず、このサブルーチンを以下に示す。

```

c
c      SUBROUTINE /S_slip_bilinear_s ( 3次元せん断型S字型スリップバイリニア )
c
      subroutine S_slip_bilinear_s(Member,Element,vv,vpp)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s2_slip_bilinear / 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      第一ステップ処理
      if(Member.istat(1).eq.0.and.Member.istat(2).eq.0) then / 1
      call Initial_S_slip_bilinear_s(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_S_slip_bilinear_s(Member.AK_tan(i),Member.istat(i), / 4
      * Element.AK_1,Element.AK_2,Element.AK_3,Member.P3(i),
      * Element.Q_1,dU,Member.stress(i+1),
      * Member.QmaxP(i),Member.QmaxM(i),Member.P1(i),
      * Member.UmaxP(i),Member.UmaxM(i),
      * Up,Member.P2(i))
      end do
      return
      end

```

```

c
c      SUBROUTINE /Cal_S_slip_bilinear_s ( 3次元せん断型S字型スリップバイリニア )
c
      subroutine Cal_S_slip_bilinear_s(AK,istat,AK_1,AK_2,AK_3,AK_5,Q_1,du,
      * P,P1,P2,P3,du3,du4,ug,st)
      implicit real*8(A-H,O-Z)
c

```

```

c      AK      :接線剛性          istat :現在の状態(Work)
c      AK_1    :第1 勾配          AK_2  :第2 勾配
c      AK_3    :スリップ時の剛性    AK_5  :istat=3、 istat=4 における剛性
c      Q_1     :第1 折れ点のせん断力
c      du      :増分変位          P      :現在のせん断力
c      P1      :istat=1 正側における反曲点上端
c      P2      :istat=1 負側における反曲点上端
c      P3      :st=1 時の istat=3、 istat=4 における反曲点上端
c      du3     :正方向最大変位      du4   :負方向最大変位
c      ug      :現在の変位          st    :除荷状態を表すパラメーター
C
100  continue
c                                     istat10
      if(istat.eq.10)then                                     ! 5
      p=AK*du+p
      if(du.gt.0)then
      if(p.lt.p1) return                                     ! 6
c                                     istat10 から istat=1 へ(正側)
      istat=1                                                 ! 7
      du2=(p-p1)/AK
      AK=AK_2
      p=AK*du2+p1
      return
      else
c                                     istat10
      if(p.gt.-p2)return                                     ! 8
c                                     istat10 から istat=1 へ(負側)
      istat=1                                                 ! 9
      du2=(p+p2)/AK
      AK=AK_2
      p=AK*du2-p2
      return
      endif
c                                     istat1
      elseif(istat.eq.1)then                                     !10
      p=AK*du+p
      if(p.gt.0)then
      if(du.gt.0)return                                     !11
      else
      if(du.lt.0)return                                     !12
      endif
c                                     istat1 から istat=2 へ(除荷)
      istat=2                                                 !13
      dp=AK*du
      AK=AK_1
      if(p.gt.p1)then
      du3=ug                                                 !14
      p1=p
      else
      du4=-ug                                                 !15
      p2=-p
      endif
      p=AK*du+p
      return

```

```

c                                     istat2
elseif(istat.eq.2)then                                     !16
  if(p.gt.0)then
    p=AK*du+p
    if(du.gt.0)then
      if(st.eq.0)then
        if(p.lt.p1) return                                     !17
c                                     istat2 から istat=1 へ(正側)
        istat=1                                             !18
        du2=(p-p1)/AK
        AK=AK_2
        p=AK*du2+p1
        return
      else
        if(p.lt.p3)return                                     !19
C                                     istat2 から istat=4 へ(正側)
        st=0                                                 !20
        istat=4                                             !21
        du2=(p-p3)/AK
        AK=AK_5
        p=AK*du2+p3
        return
      endif
    else
      if(p.gt.0)return                                     !22
C                                     istat2 から istat=3 へ(負側)
        istat=3                                             !23
        du2=p/AK
        AK=AK_3
        p=AK*du2
        return
      endif
C                                     istat2(負側)
        else                                                 !24
        p=AK*du+p
        if(du.lt.0)then
          if(st.eq.0)then
            if(p.gt.-p2)return                               !25
C                                     istat2 から istat=1 へ(負側)
            istat=1                                         !26
            du2=(p+p1)/AK
            AK=AK_2
            p=AK*du2-p2
            return
          else
            if(p.gt.p3)return                               !27
C                                     istat2 から istat=4 へ(負側)
            st=0                                             !28
            istat=4                                         !29
            du2=(p-p3)/AK
            AK=AK_5
            p=AK*du2+p3
            return
          endif

```

```

    else
    if(p.lt.0)return                                !30
C                                                    istat2 から istat=3 へ(正側)
    istat=3                                         !31
    du2=p/AK
    AK=AK_3
    p=AK*du2
    return
endif
endif

C                                                    istat3(正側)
elseif(istat.eq.3)then                             !32
p=AK*du+p
u=ug+du                                           !33
if(p.gt.0)then
if(du.gt.0)then
if(u.lt.0)return                                !34
C                                                    istat3 から istat4(正側)
    istat=4                                         !35
    p3=-AK*u+p                                     !36
    AK=(p1-p3)/du3                                !37
    p=AK*u+p3                                     !38
    return
else
C                                                    istat3 から istat5(除荷)
    st=0                                           !39
    istat=5                                       !40
    dp=AK*du
    AK=AK_1
    p=p-dp
    p3=p                                           !41
    p=AK*du+p
    return
endif
else
C                                                    istat3 ( 負側 )
if(du.lt.0)then
if(u.gt.0)return                                !42
C                                                    istat3 から istat4 ( 負側 )
    istat=4                                         !43
    p3=-AK*u+p                                     !44
    AK=(p2+p3)/du4                                !45
    p=AK*u+p3                                     !46
    return
else
C                                                    istat3 から istat5(除荷)
    st=0                                           !47
    istat=5
    dp=AK*du
    AK=AK_1
    p=p-dp
    p3=p                                           !48
    p=AK*du+p
    return

```

```

endif
endif
C                                istat4
elseif(istat.eq.4)then                                !49
p=AK*du+p
if(p.gt.0)then
if(du.gt.0)then
if(p.lt.p1)return                                !50
C                                istat4 から istat=1 へ(正側)
istat=1                                !51
du2=(p-p1)/AK
AK=AK_2
p=AK*du2+p1
return
else
C                                istat4 から istat=2,5 へ(除荷)
st=1                                !52
dp=AK*du
AK_5=AK                                !53
AK=AK_1
p=p-dp
p3=p                                !54
p=AK*du+p3
du5=-p3/AK_1+ug                                !55
if(du5.gt.0)then
istat=2                                !56
else
istat=5                                !57
endif
return
endif
C                                istat4
else
if(du.lt.0)then
if(p.gt.-p2)return                                !58
C                                istat4 から istat=1 へ(負側)
istat=1                                !59
du2=(p+p2)/AK
AK=AK_2
p=AK*du2-p2
return
else
C                                istat4 から istat=2,5 へ(除荷)
st=1                                !60
dp=AK*du
AK_5=AK                                !61
AK=AK_1
p=p-dp
p3=p                                !62
p=AK*du+p3
du5=-p3/AK_1+ug                                !63
if(du5.lt.0)then
istat=2                                !64
else

```

	istat=5	!65
	endif	
	return	
	endif	
C	istat5	
	elseif(istat.eq.5)then	!66
	u=ug+du	
	if(p.gt.0)then	
	p=AK*du+p	
	if(du.gt.0)then	
C	if(p.lt.p3)return	!67
		istat5 から istat=3/4 へ(正側)
	if(st.eq.0)then	
	istat=3	!68
	else	
	istat=4	!69
	endif	
	st=0	!70
	du2=(p-p3)/AK	
	AK=AK_5	!71
	p=AK*du2+p3	
	return	
C	istat5	
	else	
C	if(p.gt.0)return	!72
		istat5 から istat=4 へ(負側)
	st=0	!73
	istat=4	!74
	du2=p/AK	
	du5=du4+u+du2	!75
	AK=p2/du5	!76
	p=AK*du2	
	return	
C	endif	
	istat5	
	else	
	p=AK*du+p	
	if(du.lt.0)then	
C	if(p.gt.p3)return	!77
		istat5 から istat=3/4 へ(負側)
	if(st.eq.0)then	
	istat=3	!78
	else	
	istat=4	!79
	endif	
	st=0	!80
	du2=(p-p3)/AK	
	AK=AK_5	!81
	p=AK*du2+p3	
	return	
C	istat5	
	else	
	if(p.lt.0)return	!82

```

C                                     istat5 から istat=4 へ(正側)
      st=0                                     !83
      istat=4                                 !84
      du2=p/AK
      du5=du3-u+du2                                     !85
      AK=p1/du5                                     !86
      p=AK*du2
      return
    endif
  endif
end if
end

```

```

C
C      SUBROUTINE /Initial_S_slip_bilinear_s
C                                     (3次元せん断型S字型スリップバイリニア初期設定)
C
      subroutine Initial_S_slip_bilinear_s(Element,Member)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_s2_slip_bilinear / Member
      record / element_s2                / Element
      integer i                                ! カウンター
      do i=1,2                                     !87
      Member.istat(i) = 10                         !88
      Member.AK_tan(i) = Element.AK_1             !89
      Member.QmaxP(i) = Element.Q_1               !90
      Member.QmaxM(i) = Element.Q_1               !91
      Member.stress(i+1) = 0                       !92
      Member.UmaxP(i) = Element.Q_1/Element.AK_1  !93
      Member.UmaxM(i) = Element.Q_1/Element.AK_1  !94
      Member.P2(i) = 0                             !95
      end do
      return
      end

```

1. 両方向 (v 方向、w 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_S_slip_bilinear_s() をコールする。
3. 両方向 (v 方向、w 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_S_slip_bilinear_s() をコールする。
5. 状態が istat : 10 の場合、以下の処理を行う。
6. 増分変位が正の場合 ($du > 0$) で、増分後のせん断力が正方向の境界値 $p1$ より小さければ、このサブルーチンより戻る。
7. 増分後のせん断力が $p1$ より大きい場合は状態が 1 となり、以下の処理を行う。
8. 増分変位が負の場合 ($du < 0$) で、増分後のせん断力が負方向の境界値 $-p3$ よりも大きければ、このサブルーチンより戻る。

9. 増分後のせん断力が $-p_3$ よりも小さい場合は状態が1となり、以下の処理を行う。
10. 状態が $istat : 1$ の場合、以下の処理を行う。
11. 現在のせん断力が0よりも大きく、増分変位が正 ($du > 0$) ならば、このサブルーチンより戻る。
12. 現在のせん断力が0よりも小さく、増分変位が負 ($du < 0$) ならば、このサブルーチンより戻る。
13. 現在のせん断力が0よりも大きく増分変位が負 ($du < 0$) であるか、現在のせん断力が0よりも小さく増分変位が正 ($du > 0$) ならば、状態は2となり、以下の処理を行う。
14. 状態1で除荷したため、現在のせん断力が p_1 よりも大きければ正側の最大変位として除荷したときの変位を du_3 に記憶する。また、増分前のせん断力を求めて正側の最大せん断力として p_1 に記憶する。
15. 状態1で除荷したため、現在のせん断力が $-p_2$ よりも小さければ負側の最大変位として除荷したときの変位を du_4 に記憶する。また、増分前のせん断力を求めて負側の最大せん断力として p_2 に記憶する。
16. 状態が $istat : 2$ の場合、以下の処理を行う。
17. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合で、除荷状態を表す st が0であり、増分後のせん断力が正方向の境界値 p_1 より小さければ、このサブルーチンより戻る。
18. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合で、除荷状態を表す st が0であり、増分後のせん断力が正方向の境界値 p_1 より大きければ、状態は1となり以下の処理を行う。
19. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合で、除荷状態を表す st が1であり、増分後のせん断力が正方向の境界値 p_3 より小さければ、このサブルーチンより戻る。
20. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合で、除荷状態を表す st が1であり、増分後のせん断力が正方向の境界値 p_3 より大きければ、除荷状態を表す st を0にして以下の処理を行う。
21. 状態を4にする。
22. 現在のせん断力が正で増分変位が負 ($du < 0$) の場合で、増分後のせん断力が負方向の境界値0よりも大きければ、このサブルーチンより戻る。
23. 現在のせん断力が正で増分変位が負 ($du < 0$) の場合で、増分後のせん断力が負方向の境界値0よりも小さければ、状態は3となり以下の処理を行う。
24. 現在のせん断力が負であるならば以下の処理を行う。
25. 増分変位が負 ($du < 0$) の場合で、除荷状態を表す st が0であり、増分後のせん断力が負側の境界値 $-p_2$ より大きければ、このサブルーチンより戻る。

26. 増分変位が負 ($du < 0$) の場合で、除荷状態を表す st が 0 であり、増分後のせん断力が負側の境界値 $-p_2$ より小さければ、状態を 1 として以下の処理を行う。
27. 増分変位が負 ($du < 0$) の場合で、除荷状態を表す st が 1 であり、増分後のせん断力が負側の境界値 $-p_3$ より大きければ、このサブルーチンより戻る。
28. 増分変位が負 ($du < 0$) の場合で、除荷状態を表す st が 1 であり、増分後のせん断力が負側の境界値 $-p_3$ より小さければ、 st を 0 にして以下の処理を行う。
29. 状態を 4 にする。
30. 増分変位が正 ($du > 0$) の場合で、増分後のせん断力が正方向の境界値 0 よりも小さければ、このサブルーチンより戻る。
31. 増分変位が正 ($du > 0$) の場合で、増分後のせん断力が正方向の境界値 0 よりも大きければ、状態を 3 にし以下の処理を行う。
32. 状態が $istat : 3$ の場合、以下の処理を行う。
33. 前ステップまでの全変位に、今回の増分変位を足して増分後の変位 u を求める。
34. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合、増分後の変位 u が 0 よりも小さければ、このサブルーチンより戻る。
35. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合、増分後の変位 u が 0 よりも大きければ、状態を 4 にして以下の処理を行う。
36. 変位 0 におけるせん断力を求め、 p_3 にセットする。
37. p_3 を用いて座標 $(0, p_3)$ から正方向の最大点座標 (du_3, p_1) を指向する接線剛性を計算し、 AK にセットする。
38. 求めた接線剛性に増分後の変位を掛けて、 p_3 に足し合わせ増分後のせん断力を計算する。
39. 現在のせん断力が正で増分変位が負 ($du < 0$) の場合、除荷状態を表す st を 0 にする。
40. 状態を 5 にする。
41. 増分前のせん断力を状態 5 の境界値 p_3 にセットする。
42. 現在のせん断力が負で増分変位が負 ($du < 0$) の場合、増分後の変位 u が 0 よりも大きければ、このサブルーチンより戻る。
43. 現在のせん断力が負で増分変位が、負 ($du < 0$) の場合、増分後の変位 u が 0 よりも小さければ、状態を 4 にして以下の処理を行う。
44. 変位 0 におけるせん断力を求め、 p_3 にセットする。
45. p_3 を用いて座標 $(0, p_3)$ から負方向最大点座標 $(-du_4, -p_2)$ を指向する接線剛性を計算し、 AK にセットする。
46. 求めた接線剛性に増分後の変位を掛けて、 p_3 に足し合わせ増分後のせん断力を計算する。
47. 現在のせん断力が負で増分変位が正 ($du > 0$) の場合、除荷状態を表す st を 0 に

- し、状態を 5 にして以下の処理を行う。
48. 増分前のせん断力を状態 5 の境界値 p_3 にセットする。
 49. 状態が $istat : 4$ の場合、以下の処理を行う。
 50. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合、増分後のせん断力が正方向の境界値 p_1 より小さければ以下の処理を行う。
 51. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合、増分後のせん断力が正方向の境界値 p_1 より大きければ状態を 1 にし、以下の処理を行う。
 52. 現在のせん断力が正で増分変位が負 ($du < 0$) の場合、除荷状態を表す st を 1 にして以下の処理を行う。
 53. 現在の接線剛性を AK_5 にセットする。
 54. 増分前のせん断力を境界値 p_3 にセットする。
 55. p_3 と除荷剛性 AK_4 を用いて、除荷してせん断力が 0 となる時の残留変位 du_5 を求める。
 56. du_5 が正 ($du_5 > 0$) の場合、残留変位が生じるため、スリップの履歴を持つ状態 2 とする。
 57. du_5 が負 ($du_5 < 0$) の場合、残留変位が生じないため、スリップの履歴を持たない状態 5 とする。
 58. 現在のせん断力が負で増分変位が負 ($du < 0$) の場合、増分後のせん断力が負方向の境界値 $-p_2$ より大きければ、このサブルーチンより戻る。
 59. 現在のせん断力が負で増分変位が負 ($du < 0$) の場合、増分後のせん断力が負方向の境界値 $-p_2$ より小さければ、状態を 1 とし以下の処理を行う。
 60. 現在のせん断力が負で増分変位が正 ($du > 0$) の場合、除荷状態を表す st を 1 にして以下の処理を行う。
 61. 現在の接線剛性を AK_5 にセットする。
 62. 増分前のせん断力を境界値 p_3 にセットする。
 63. p_3 と除荷剛性 AK_4 を用いて、除荷してせん断力が 0 となる時の残留変位 du_5 を求める。
 64. du_5 が負 ($du_5 < 0$) の場合、残留変位が生じるため、スリップの履歴を持つ状態 2 とする。
 65. du_5 が正 ($du_5 > 0$) の場合、残留変位が生じないため、スリップの履歴を持たない状態 5 とする。
 66. 状態が $istat : 5$ の場合、以下の処理を行う。
 67. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合、増分後のせん断力が正方向の境界値 p_3 よりも小さければ、このサブルーチンより戻る。
 68. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合、増分後のせん断力が正方向の境界値 p_3 よりも大きく、除荷状態を表す st が 0 の時、状態を 3 にする。

69. 現在のせん断力が正で増分変位が正 ($du > 0$) の場合、増分後のせん断力が正方向の境界値 $p3$ よりも大きく、除荷状態を表す st が 1 の時、状態を 4 にする。
70. 除荷状態を表す st を 0 にする。
71. 接線剛性を記憶しておいた AK_5 にする。
72. 現在のせん断力が正で増分変位が負 ($du < 0$) の場合、増分後のせん断力が負方向の境界値 0 よりも大きければ、このサブルーチンより戻る。
73. 現在のせん断力が正で増分変位が負 ($du < 0$) の場合、増分後のせん断力が負方向の境界値 0 よりも小さければ、除荷状態を表す st を 0 にする。
74. 状態を 4 にする。
75. 負方向の最大変位 $-du4$ とせん断力が 0 となる点の変位を求め、その間の変位量を計算し $du5$ にセットする。
76. $du5$ を用いてせん断力が 0 になる点から負側の最大点座標 ($-du4, -p2$) を指向する接線剛性を計算し AK にセットする。
77. 現在のせん断力が負で増分変位が負 ($du < 0$) の場合、増分後のせん断力が負方向の境界値 $p3$ よりも大きければ、このサブルーチンより戻る。
78. 現在のせん断力が負で増分変位が負 ($du < 0$) の場合、増分後のせん断力が負方向の境界値 $p3$ よりも小さく、除荷状態を表す st が 0 の時、状態を 3 にする。
79. 現在のせん断力が負で増分変位が負 ($du < 0$) の場合、増分後のせん断力が負方向の境界値 $p3$ よりも小さく、除荷状態を表す st が 1 の時、状態を 4 にする。
80. 除荷状態を表す st を 0 にする。
81. 接線剛性を記憶しておいた AK_5 にする。
82. 現在のせん断力が負で増分変位が正 ($du > 0$) の場合、増分後のせん断力が正方向の境界値 0 よりも小さければ、このサブルーチンより戻る。
83. 現在のせん断力が負で増分変位が正 ($du > 0$) の場合、増分後のせん断力が正方向の境界値 0 よりも大きければ、除荷状態を表す st を 0 にする。
84. 状態を 4 にする。
85. 正方向の最大変位 $du3$ とせん断力が 0 となる点の変位を求め、その間の変位量を計算し $du5$ にセットする。
86. $du5$ を用いてせん断力が 0 になる点から正側の最大点座標 ($du3, p1$) を指向する接線剛性を計算し AK にセットする。
87. 両方向 (1=v 方向、2=w 方向) についてワーク用構造体 Member の成分をセットする。
88. 状態パラメータは初期で 10 に設定する。
89. 接線剛性は、初期剛性を設定する。
90. Member.QmaxP(i) に正側の第 1 折れ点のせん断力を設定する。
91. Member.QmaxM(i) に負側の第 1 折れ点のせん断力を設定する。

92. せん断力は初期でゼロに設定する。
 93. Member.UmaxP(i)に正側の第1折れ点の変位を設定する。
 94. Member.UmaxM(i)に負側の第1折れ点の変位を設定する。
 95. st は初期でゼロに設定する。

本節では、接合部に関する履歴ルールで、図 2-8 に示す 3 点折れ S 字型スリップモデルについて説明する。図中の記号は以下のようなものである。

2.2.5.2

3 点折れ S 字型スリップモデル

- | | |
|--------------------------|--------------------------|
| 第 1 勾配 : AK_1 | 第 2 勾配 : AK_2 |
| 第 3 勾配 : AK_3 | 第 4 勾配 : AK_4 |
| 第 5 勾配 : AK_5 | |
| 第 1 折れ点のせん断力 : $\pm Q_1$ | 第 2 折れ点のせん断力 : $\pm Q_2$ |
| 第 3 折れ点のせん断力 : $\pm Q_3$ | |

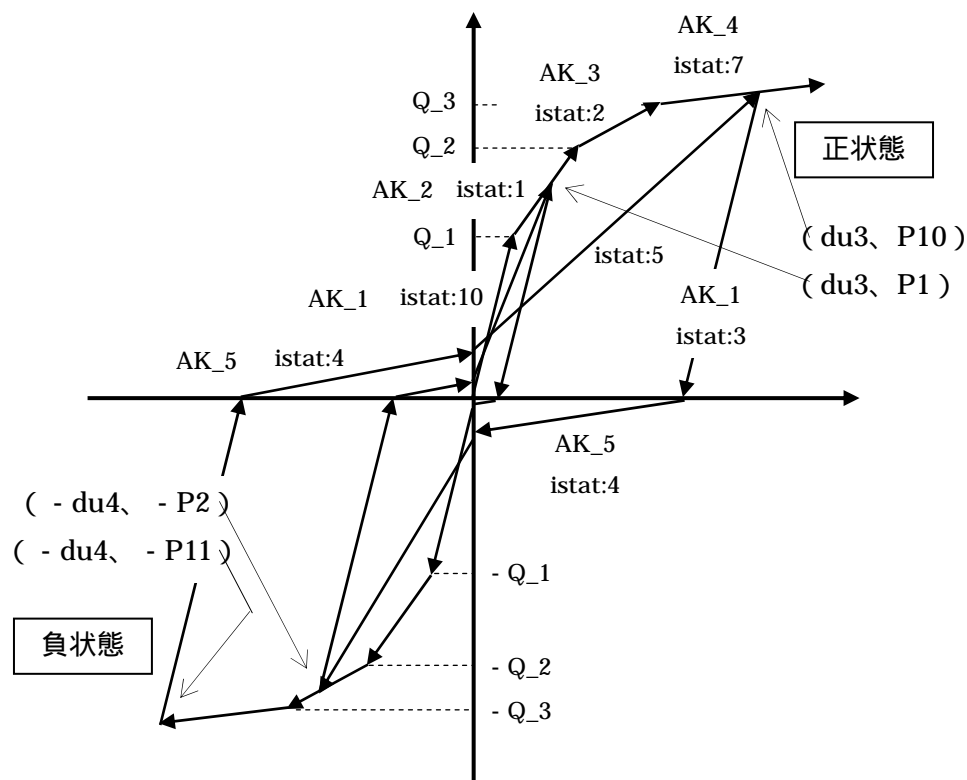


図 2-8 3 点折れ S 字型スリップモデル

この履歴特性は木質構造物の木質パネルの抵抗機構をせん断要素でモデル化するために作成した履歴特性である。正負対称の 3 点折れの骨

格曲線であり、スリップ時の剛性もユーザー側で指定できる履歴とした。弾性状態は、 $istat:10$ であり、 $istat:3$ は除荷された時の弾性状態を表している。

降伏以降の状態は、 $istat:1$ と $istat:2$ 、 $istat:7$ で表しており、 $istat:4$ は残留変位によるスリップ状態を表す。 $istat:5$ はスリップ状態の変位がゼロとなる座標から最高点を狙って進行する履歴を描く。

ここで、この履歴について説明する。まず初めに、図 2-8 を用いて履歴全体の流れについて説明する。

正負どちらの履歴においても初期の状態 10 の履歴は同じであり、状態 10 は境界値 $Q_1 > Q > -Q_1$ の範囲内で変位が反転したとしても直線履歴を描く。ここでは、境界値 Q_1 を越えて履歴を描く場合を用いて説明する。状態 10 において正方向境界値 Q_1 を越えると状態 1 となる。状態 1 の正方向の境界値は $Q_2 > Q$ であるが、 Q_2 を越えずに増分変位が反転し負方向に進むと状態 3 となり除荷時のせん断力を $P1$ として $P1 > Q > 0$ を境界値として直線的に進行する履歴となる。状態 3 において、増分変位が反転して正方向に進み、境界値 $P1$ を越えると状態 1 に戻る。また、状態 1 において増分変位が負方向に進み、境界値 0 を越えると状態 4 となり、スリップを表す状態に入る。状態 4 において増分変位が反転し正方向に進行する場合、状態 6 となるが、状態 6 以降の履歴については後述する。負方向に進行する場合は、変位がゼロを超えると状態 5 となり、1 度も負方向の履歴を経験していなければ負方向の境界値を $Q > -Q_1$ 、負方向の履歴を経験しており、第 1 折れ点を経験しておれば負方向の境界値を $Q > -P3$ 、第 2 折れ点を経験しておれば負方向の境界値を $Q > -P4$ 、第 3 折れ点を経験しておれば負方向の境界値を $Q > -P11$ としてこの範囲内で直線的に進行する。状態 5 は負方向に増分変位が進行する場合、負側の最高点座標 $(-du4, -Q_1)$ もしくは $(-du4, -P3)$ $(-du4, -P4)$ $(-du4, -P11)$ を指向する勾配をもって進行する。図 2-8 の場合は負方向の境界値 $(-Q_1)$ を越えると状態 1 に進行する。この時の状態 1 は負方向に増分変位が進行する場合、境界条件は $Q > -Q_2$ であり、境界値 $-Q_2$ を越えると状態 2 に進行する。また、状態 2 で増分変位が反転し正方向に進行すると状態は 3 となる。除荷時のせん断力を $-P4$ として、境界値 $0 > Q > -P4$ の範囲内で直線履歴を描く。状態 3 から境界値 $-P4$ を越えると状態 2 に戻り、境界値 0 を越えると正側のスリップを表す状態 4 に進行する。状態 4 は正方向に進む場合、変位がゼロを超えると状態 5 となり、正側の最大点座標を指向して進行する。また、増分変位が反転すると状態は 6 となる。

状態 6 以降の履歴については後述する。ここでは、正側の状態 1 からの除荷、負側の状態 2 からの除荷を説明したが、状態 1、2、7 の正負すべての場合において同様の履歴となる。

次に、状態 4、5 から除荷した時の履歴について、図 2-9 を用いて説明する。

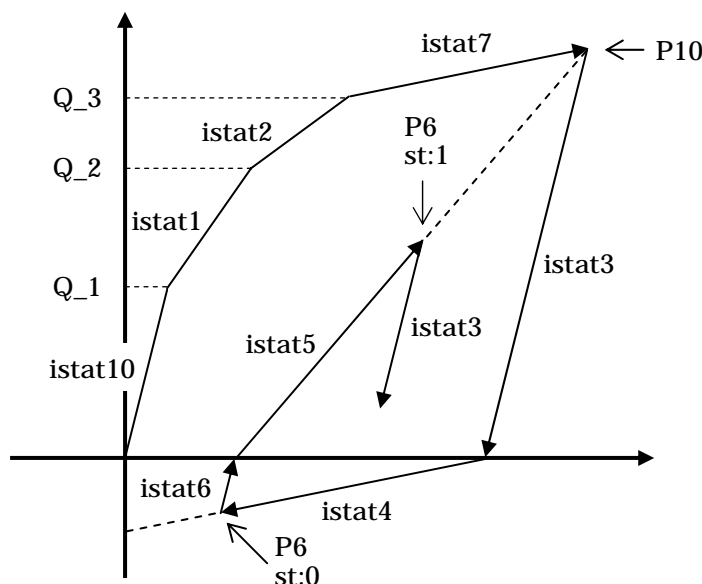


図 2-9 状態 4、5 から除荷したときの履歴

まず初めに、図 2-9 を用いて負側のスリップを表す状態 4 から除荷した時の履歴について説明する。状態 4 で正方向に増分変位が反転すると状態 6 となり、除荷時のせん断力を P_6 、除荷時の状態を $st:0$ とする。状態 6 の境界値は $0 > Q > P_6$ であり、境界値 0 を越えると正方向最大点座標を指向する状態 5 に進み、増分変位が反転し負方向に進むと状態 4 に戻る。状態 6 から状態 5 に進行し、増分変位が反転して負方向に進むと状態は 3 となり、除荷時のせん断力を P_6 、除荷時の状態を $st:1$ とする。この時の状態 3 の境界条件は $P_6 > Q > 0$ となり、 P_6 を越えると状態 5 に戻り、0 を越えると状態 4 に進行して、この後の流れは、上述の全体の流れと同様である。

次に、図 2-9 の正状態で状態 5 から除荷された履歴について説明する。この 2 通りの履歴を描かれる。1 つ目は除荷されてせん断力が 0 に到達した時に残留変位が残り、スリップが生じる場合であり、2 つ目は残留変位が残らずスリップが生じない場合である。したがって、ここではスリップが生じる場合は状態 5 から除荷されると状態 3 に進行し、スリップが生じない場合は状態 6 に進行する履歴とする。次に、2 通りの履歴について説明する。

1. スリップが生じる場合

状態 5 で除荷された時に除荷勾配を用いて残留変位の有無を検討し、スリップが生じている時、状態 3 に進行する。除荷した時のせん断力を P_6 とし、除荷時の状態を $st:1$ とする。状態 3 の境界値は $P_6 > Q > 0$ とし、境界値 P_6 を越える場合は状態 5 に戻り、境界値 0 を越える場合はスリップを表す状態 4 に進行する。

2. スリップが生じない場合

状態 5 で除荷された時に除荷勾配を用いて残留変位の有無を検討し、スリップが生じていない時、状態 6 に進行する。除荷した時のせん断力を P_6 とし、除荷時の状態を $st:1$ とする。状態 6 の境界値は $P_6 > Q > 0$ とし、境界値 P_6 を越える場合は状態 5 に戻り、境界値 0 を越える場合は負側の最高点を指向する状態 5 に進行する。

以上の説明をまとめると、3 点折れ S 字型スリップモデルの履歴特性は、次の 7 つの状態で表現される。

istat:10 弾性状態であり、剛性は AK_1 である。 $Q_1 > Q > -Q_1$ の中で動き、 Q_1 、 $-Q_1$ を越えると状態 1 となる。

istat:1 正負共に剛性は AK_2 である。せん断力が正の場合、増分変位が正方向に進行する時 $Q_2 > Q$ の範囲内で直線的に動き、除荷されて増分変位が反転し負方向に進行すると、状態 3 となって反転した時の増分前のせん断力を P_1 にする。

せん断力が負の場合は、増分変位が負方向に進行する時 $Q > -Q_2$ の範囲内で直線に動き、除荷されて増分変位が反転し正方向に進行すると、状態 3 となって反転した時の増分前のせん断力を P_3 とする。

istat:2 正負共に剛性は AK_3 である。せん断力が正の場合、増分変位が正方向に進行する時 $Q_3 > Q$ の範囲内で直線的に動き、除荷されて増分変位が反転し負方向に進行すると、状態 3 となって反転した時の増分前のせん断力を P_2 にする。

せん断力が負の場合は、増分変位が負方向に進行する時 $Q > -Q_3$ の範囲内で直線に動き、除荷されて増分変位が反転し正方向に進行すると、状態 3 となって反転した時の増分前のせん断力を P_4 とする。

istat:7 正負共に剛性は AK_4 である。せん断力が正の場合、増分変位

が正方向に進行する時その上限はなく、除荷されて増分変位が反転し負方向に進行すると、状態 3 となって反転した時の増分前のせん断力を P_{10} にする。

せん断力が負の場合は、増分変位が負方向に進行する時その上限はなく、除荷されて増分変位が反転し正方向に進行すると、状態 3 となって反転した時の増分前のせん断力を P_{11} とする。

istat:3 状態 1、2、7 で除荷された時の状態であり、剛性は AK_1 である。せん断力が正の場合は $P_1 > Q > 0$ もしくは $P_2 > Q > 0$ 、 $P_{10} > Q > 0$ の中で直線的に動き、正方向に増分変位が進み、 P_1 または P_2 、 P_{10} を越えると状態 1、2、7 に進む。また、負方向に増分変位が進み、0 を越えると状態は 4 に進む。

せん断力が負の場合は $0 > Q > -P_3$ もしくは $0 > Q > -P_4$ 、 $0 > Q > -P_{11}$ の中で直線的に動き、負方向に増分変位が進み、 $-P_3$ または $-P_4$ 、 $-P_{11}$ を越えると状態 1、2、7 に進む。また、正方向に増分変位が進み、0 を越えると状態は 4 に進む。

istat:4 スリップ時の勾配であり、剛性は AK_4 である。せん断力が正の場合は増分変位が正方向の時、変位がゼロを越えると状態 5 となる。増分変位が反転し負方向になった時、状態 6 となり除荷時の状態を表す st をゼロにして増分前のせん断力を P_6 とする。

せん断力が負の場合は増分変位が負方向の時、変位がゼロを越えると状態 5 となる。増分変位が反転し正方向になった時、状態 6 となり除荷時の状態を表す st をゼロにして増分前のせん断力を P_6 とする。

istat:5 正負共に最大点を指向する勾配であり、せん断力が正の場合は増分変位が正方向の時、その境界は $P_1 > Q$ もしくは $P_2 > Q$ 、 $P_{10} > Q$ であり、増分変位が反転し負方向となった時、除荷時の状態を表す st を 1 にして増分前のせん断力を P_6 とし状態 3、6 に進行する。

せん断力が負の場合は増分変位が負方向の時、その境界は $Q > -P_3$ もしくは $Q > -P_4$ 、 $Q > -P_{11}$ であり、増分変位が反転し正方向となった時、除荷時の状態を表す st を 1 にして増分前のせん断力を P_6 とし状態 3、6 に進行する。

istat:6 状態 4、5 から除荷する時の状態であり、剛性は AK_4 である。せん断力が正の場合は $P_6 > Q > 0$ の中で直線的に動き、正方向に増分変位が進み、 P_6 を越えると $st:0$ の時は状態 4 に進行し、

st:1 の時は状態 5 に進行する。また、負方向に増分変位が進み、ゼロを越えると状態は 5 にむ。

せん断力が負の場合は $0 > Q > P6$ の中で直線的に動き、負方向に増分変位が進み、P6 を越えると st:0 の時は状態 4 に進行し、st:1 の時は状態 5 に進行する。正方向に増分変位が進み、ゼロを越えると状態は 5 に進む。

以上で各状態の説明を終える。ここでは、この履歴ルールを表現するサブルーチンを具体的に説明しよう。

```

c
c      SUBROUTINE /S_slip_woodpanel
c
c      3次元せん断型3点折れS字型スリップモデル
c
c      subroutine S_slip_woodpanel(Member,Element,vv,vpp)
c      implicit real*8(A-H,O-Z)
c      include "submain.h"
c      record / member_woodpanel / Member
c      record / element_s2 / Element
c      integer i ! カウンター
c      dimension vv(12) ! 部材座標系増分変位
c      dimension vpp(12) ! 部材座標系直前変位
c      real*8 dU ! 現在の相対増分変位
c      real*8 Up ! 直前の相対変位
c
c      open(99,file='check_file.txt')
c      第一ステップ処理
c      if(Member.istat(1).eq.0.and.Member.istat(2).eq.0) then ! 1
c      call initial_S_slip_woodpanel(Element,Member) ! 2
c      return
c      endif
c
c      第nステップ処理
c      do i=1,2 ! 3
c      dU=vv(7+i)-vv(1+i) ! 現在の相対増分変位の計算
c      Up=vpp(7+i)-vpp(1+i) ! 直前の相対変位の計算
c      call Cal_S_slip_woodpanel(Member.AK_tan(i),Member.istat(i), ! 4
c      * Element.AK_1,Element.AK_2, Element.AK_3,
c      * Element.U_1, Element.dm4,
c      * Element.Q_1, Element.Q_2,Element.U_2,
c      * dU, Member.stress(i+1),
c      * Member.QmaxP(i),Member.QmaxM(i),
c      * Member.UmaxP(i),Member.UmaxM(i),
c      * Up,Member.P1(i),
c      * Member.P2(i),Member.P3(i),
c      * Member.P4(i),Member.P5(i),
c      * Member.P6(i),Member.P7(i))
c      end do

```

```

return
end

```

```

c
c      SUBROUTINE /Cal_S_slip_woodpanel
c
c      3次元せん断型3点折れS字型スリップモデル
c
c      subroutine Cal_S_slip_woodpanel (AK, istat, AK_1, AK_2, AK_3, AK_4, AK_5,
*                                     Q_1, Q_2, Q_3, du, P, p10, p11, du3, du4, ug, st,
*                                     AK_6, P1, P2, P3, P4, P6)
c      implicit real*8(A-H, O-Z)
c
c      AK          : 接線剛性
c      istat       : 現在の状態 (Work)
c      st         : 除荷状態を表すパラメーター
c
100  continue
c                                     istat10
c      if (istat.eq.10) then
c                                     ! 5
c      p=AK*du+p
c      if(du.gt.0) then
c      if(p.lt.p1) return
c                                     istat10 から istat=1 へ (正側)
c      istat=1
c      du2=(p-p1)/AK
c      AK=AK_2
c      p=AK*du2+p1
c      return
c      else
c      if(p.gt.-p3) return
c                                     istat10 から istat=1 へ (負側)
c      istat=1
c      du2=(p+p3)/AK
c      AK=AK_2
c      p=AK*du2-p3
c      return
c      endif
c                                     istat1
c      elseif(istat.eq.1) then
c                                     ! 6
c      p=AK*du+p
c      if(p.gt.0) then
c      if(du.gt.0) then
c      if(p.lt.p2) return
c                                     istat1 から istat=2 へ (正側)
c      istat=2
c      du2=(p-p2)/AK
c      AK=AK_3
c      p=AK*du2+p2
c      return
c      else
c                                     istat1 から istat=3 へ (除荷)
c      istat=3
c      dp=AK*du

```

```

    AK=AK_1
    p=p-dp
    du3=ug
    p1=p
    p=AK*du+p
    return
endif
else
    if(du.lt.0) then
    if(p.gt.-p4) return
c
    istat=2
    du2=(p+p4)/AK
    AK=AK_3
    p=AK*du2-p4
    return
    else
c
    istat=3
    dp=AK*du
    AK=AK_1
    p=p-dp
    du4=-ug
    p3=-p
    p=AK*du+p
    return
    endif
    endif
c
    elseif(istat.eq.2) then
    p=AK*du+p
    if(p.gt.0) then
    if(du.gt.0) then
    if(p.lt.p10) return
c
    istat=7
    du2=(p-p10)/AK
    AK=AK_4
    p=AK*du2+p10
    p2=Q_3
    return
    else
c
    istat=3
    dp=AK*du
    AK=AK_1
    p=p-dp
    du3=ug
    p2=p
    p=AK*du+p
    return
    endif
    else
    if(du.lt.0) then

```

istat1 から istat=2 へ (負側)

istat1 から istat=3 へ (除荷)

istat2

istat2 から istat=7 へ (正側)

istat2 から istat=3 へ (除荷)

! 7

```

    if(p.gt.-p11)return
c                                     istat2 から istat=7 へ (負側)
    istat=7
    du2=(p+p11)/AK
    AK=AK_4
    p=AK*du2-p11
    p4=Q_3
    return
    else
c                                     istat2 から istat=3 へ (除荷)
    istat=3
    dp=AK*du
    AK=AK_1
    p=p-dp
    du4=-ug
    p4=-p
    p=AK*du+p
    return
    endif
    endif
c                                     istat7
    elseif(istat.eq.7)then
    p=AK*du+p
    if(p.gt.0)then
    if(du.gt.0)return
c                                     istat7 から istat=3 へ (除荷)
    istat=3
    dp=AK*du
    AK=AK_1
    p=p-dp
    du3=ug
    p10=p
    p=AK*du+p
    return
    else
    if(du.lt.0)return
c                                     istat7 から istat=3 へ (除荷)
    istat=3
    dp=AK*du
    AK=AK_1
    p=p-dp
    du4=-ug
    p11=-p
    p=AK*du+p
    return
    endif
c                                     istat3
    elseif(istat.eq.3)then
    if(p.gt.0)then
    p=AK*du+p
    if(du.gt.0)then
    if(st.eq.0)then
    if(p2.eq.Q_2)then
    if(p.lt.p1)return

```

```

c                                istat3 から istat=1 へ (正側)
    istat=1
    du2=(p-p1)/AK
    AK=AK_2
    p=AK*du2+p1
    return
elseif (p2.eq.Q_3) then
if (p.lt.p10) return

c                                istat3 から istat=7 へ (正側)
    istat=7
    du2=(p-p10)/AK
    AK=AK_4
    p=AK*du2+p10
    return
else
if (p.lt.p2) return

c                                istat3 から istat=2 へ (正側)
    istat=2
    du2=(p-p2)/AK
    AK=AK_3
    p=AK*du2+p2
    return
endif
else
if (p.lt.p6) return

c                                istat3 から istat=5 へ (除荷)
    st=0
    istat=5
    du2=(p-p6)/AK
    AK=AK_6
    p=AK*du2+p6
    return
endif
else
if (p.gt.0) return

c                                istat3 から istat=4 へ (正側)
    istat=4
    du2=p/AK
    AK=AK_5
    p=AK*du2
    return
endif
else
p=AK*du+p
if (du.lt.0) then
if (st.eq.0) then
if (p4.eq.Q_2) then
if (p.gt.-p3) return

c                                istat3 から istat=1 へ (負側)
    istat=1
    du2=(p+p3)/AK
    AK=AK_2
    p=AK*du2-p3
    return

```

```

elseif(p4.eq.Q_3)then
  if(p.gt.-p11)return
c
  istat=7
  du2=(p+p11)/AK
  AK=AK_4
  p=AK*du2-p11
  return
  else
  if(p.gt.-p4)return
c
  istat=2
  du2=(p+p4)/AK
  AK=AK_3
  p=AK*du2-p4
  return
endif
else
  if(p.gt.p6)return
c
  st=0
  istat=5
  du2=(p-p6)/AK
  AK=AK_6
  p=AK*du2+p6
  return
endif
else
  if(p.lt.0)return
c
  istat=4
  du2=p/AK
  AK=AK_5
  p=AK*du2
  return
endif
endif
c
elseif(istat.eq.4)then
  p=AK*du+p
  u=ug+du
  if(p.gt.0)then
    if(du.gt.0)then
      if(u.lt.0)return
c
      istat=5
      du2=du+ug
      p5=-AK*u+p
      if(p2.eq.Q_2)then
        AK=(p1-p5)/du3
      elseif(p2.eq.Q_3)then
        AK=(p10-p5)/du3
      else
        AK=(p2-p5)/du3

```

istat3 から istat=7 へ (負側)
 istat3 から istat=2 へ (負側)
 istat3 から istat=5 へ (除荷)
 istat3 から istat=4 へ (負側)
 istat4
 ! 10
 istat4 から istat=5 へ (正側)

```

endif
p=AK*du2+p5
return
else
c                                     istat4 から istat6(除荷)
  St=0
  istat=6
  dp=AK*du
  AK=AK_1
  p=p-dp
  p6=p
  p=AK*du+p
  return
endif
else
  if(du.lt.0) then
  c                                     istat4 から istat=5 へ (負側)
    istat=5
    du2=du+ug
    p5=-AK*u+p
    if(p4.eq.Q_2) then
      AK=(p3+p5)/du4
    elseif(p4.eq.Q_3) then
      AK=(p11+p5)/du4
    else
      AK=(p4+p5)/du4
    endif
    p=AK*du2+p5
    return
  else
  c                                     istat4 から istat=6 へ (除荷)
    st=0
    istat=6
    dp=AK*du
    AK=AK_1
    p=p-dp
    p6=p
    p=AK*du+p
    return
  endif
endif
c                                     istat5 (正側)
elseif(istat.eq.5) then
  p=AK*du+p
  if(p.gt.0) then
    if(du.gt.0) then
      if(p2.eq.Q_2) then
        if(p.lt.p1) return
c                                     istat5 から istat=1 へ (正側)
        istat=1
        du2=(p-p1)/AK
        AK=AK_2
        p=AK*du2+p1

```

! 11

```

    return
elseif(p2.eq.Q_3)then
  if(p.lt.p10)return
c
    istat=7
    du2=(p-p10)/AK
    AK=AK_4
    p=AK*du2+p10
    return
  else
    if(p.lt.p2)return
c
    istat=2
    du2=(p-p2)/AK
    AK=AK_3
    p=AK*du2+p2
    return
  endif
  else
c
    st=1
    dp=AK*du
    AK_6=AK
    AK=AK_1
    p=p-dp
    p6=p
    p=AK*du+p6
    du5=-p6/AK_1+ug
    if(du5.gt.0)then
      istat=3
    else
      istat=6
    endif
    return
  endif
c
  else
    if(du.lt.0)then
      if(p4.eq.Q_2)then
        if(p.gt.-p3)return
c
          istat=1
          du2=(p+p3)/AK
          AK=AK_2
          p=AK*du2-p3
          return
        elseif(p4.eq.Q_3)then
          if(p.gt.-p11)return
c
            istat=7
            du2=(p+p11)/AK
            AK=AK_4
            p=AK*du2-p11
            return

```

istat3 から istat=7 へ (正側)

istat5 から istat=2 へ (正側)

istat5 から istat=3,6 へ (除荷)

istat5 (負側)

istat5 から istat=1 へ (負側)

istat5 から istat=7 へ (負側)

```

    else
    if(p.gt.-p4)return
c                                     istat5 から istat=2 へ (負側)
    istat=2
    du2=(p+p4)/AK
    AK=AK_3
    p=AK*du2-p4
    return
endif
c                                     istat5 から istat=3,6 へ (除荷)
    else
    st=1
    dp=AK*du
    AK_6=AK
    AK=AK_1
    p=p-dp
    p6=p
    p=AK*du+p6
    du5=-p6/AK_1+ug
    if(du5.lt.0)then
    istat=3
    else
    istat=6
    endif
    return
    endif
    endif
c                                     istat6 (正側)
    elseif(istat.eq.6)then
    if(p.gt.0)then
    p=AK*du+p
    u=ug+du
    if(du.gt.0)then
    if(p.lt.p6)return
c                                     istat6 から istat=4/5 へ (正側)
    if(st.eq.0)then
    istat=4
    du2=(p-p6)/AK
    AK=AK_4
    else
    istat=5
    du2=(p-p6)/AK
    AK=AK_6
    endif
    st=0
    p=AK*du2+p6
    return
    else
    if(p.gt.0)return
c                                     istat6 から istat=5 へ (除荷)
    st=0
    istat=5
    du2=p/AK
    du5=du4+u+du2

```

```

        if (p4.eq.Q_2) then
            AK=p3/du5
        elseif (p4.eq.Q_3) then
            AK=p11/du5
        else
            AK=p4/du5
        endif
        p=AK*du2
        return
    endif
c                                     istat6 ( 負側 )
    else
        p=AK*du+p
        u=ug+du
        if (du.lt.0) then
            if (p.gt.p6) return
c                                     istat6 から istat=4/5 へ ( 負側 )
            if (st.eq.0) then
                istat=4
                du2=(p-p6)/AK
                AK=AK_5
            else
                istat=5
                du2=(p-p6)/AK
                AK=AK_6
            endif
            st=0
            p=AK*du2+p6
            return
        else
            if (p.lt.0) return
c                                     istat6 から istat=5 へ ( 除荷 )
            st=0
            istat=5
            du2=p/AK
            du5=du3-u-du2
            if (p2.eq.Q_2) then
                AK=p1/du5
            elseif (p2.eq.Q_3) then
                AK=p10/du5
            else
                AK=p2/du5
            endif
            p=AK*du2
            return
        endif
    endif
c                                     初期設定
    endif
end

```

```

c
c      SUBROUTINE /Initial_S_slip_woodpanel
c

```

```

c      3次元せん断型S字型スリップ木質パネル初期設定
c
      subroutine Initial_S_slip_woodpanel(Element,Member)
      implicit real*8(A-H,O-Z)
      include "submain.h"
      record / member_woodpanel / Member
      record / element_s2 / Element
      integer i ! カウンター
      do i=1,2 ! 13
      Member.istat(i) = 10 ! 14
      Member.AK_tan(i) = Element.AK_1 ! 15
      Member.QmaxP(i) = Element.U_2 ! 16
      Member.QmaxM(i) = Element.U_2 ! 17
      Member.stress(i+1) = 0 ! 18
      Member.UmaxP(i) = Element.Q_1/Element.AK_1 ! 19
      Member.UmaxM(i) = Element.Q_1/Element.AK_1 ! 20
      Member.P1(i) = 0 ! 21
      Member.P3(i) = Element.Q_1 ! 22
      Member.P4(i) = Element.Q_2 ! 23
      Member.P5(i) = Element.Q_1 ! 24
      Member.P6(i) = Element.Q_2 ! 25
      end do
      return
      end

```

1. 両方向 (v 方向、w 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_S_slip_woodpanel()` をコールする。
3. 両方向 (v 方向、w 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_S_slip_woodpanel()` をコールする。
5. 状態が `istat : 10` の場合、以下の処理を行う。
6. 状態が `istat : 1` の場合、以下の処理を行う。
7. 状態が `istat : 2` の場合、以下の処理を行う。
8. 状態が `istat : 7` の場合、以下の処理を行う。
9. 状態が `istat : 3` の場合、以下の処理を行う。
10. 状態が `istat : 4` の場合、以下の処理を行う。
11. 状態が `istat : 5` の場合、以下の処理を行う。
12. 状態が `istat : 6` の場合、以下の処理を行う。
13. 両方向 (1=v 方向、2=w 方向) についてワーク用構造体 `Member` の成分をセットする。
14. 状態パラメータは初期で 10 に設定する。
15. 接線剛性は、初期剛性を設定する。
16. `Member.QmaxP(i)` に正側の第 3 折れ点のせん断力を設定する。
17. `Member.QmaxM(i)` に負側の第 3 折れ点のせん断力を設定する。

- 18 . せん断力は初期でゼロに設定する。
- 19 . Member.UmaxP(i)に正側の第 1 折れ点の変位を設定する。
- 20 . Member.UmaxM(i)に負側の第 1 折れ点の変位を設定する。
- 21 . st は初期でゼロに設定する。
- 22 . Member. P3(i)に正側の第 1 折れ点のせん断力を設定する。
- 23 . Member. P4(i)に正側の第 2 折れ点のせん断力を設定する。
- 24 . Member. P5(i)に負側の第 1 折れ点のせん断力を設定する。
- 25 . Member. P6(i)に負側の第 2 折れ点のせん断力を設定する。