



第4章 ファイバーの履歴モデル

4.1 はじめに

SPACE に組み込まれている履歴モデルは、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。本章では、部材モデル階層構造の最下層に位置するファイバーの履歴モデルについて解説する。

現在、SPACE は、Ver. 3.00 であり、ファイバーの履歴モデルには、以下の履歴が用意されており、鉄骨、RC、SRC あるいは木造部材の弾塑性履歴に対応するように設計されている。

- 4.2.1 対称バイリニア型
- 4.2.2 対称トリリニア型
- 4.2.3 直線コンクリート型
- 4.2.4 曲線コンクリート型（現在開発中）
- 4.2.5 バイリニア型（移動＋等方硬化用）
- 4.2.6 対称トリリニア型（移動＋等方硬化用）
- 4.2.7 非対称バイリニア型
- 4.2.8 非対称トリリニア型
- 4.2.9 降伏棚を有する対称バイリニア型（移動＋等方硬化用）
- 4.2.10 降伏棚を有する対称トリリニア型（移動＋等方硬化用）
- 4.2.11 鉄筋用履歴モデル
- 4.2.12 木質構造材用履歴モデル

さらに、新規部材モデルで使用可能な接合部履歴モデルとして以下の履歴モデルが用意されている。

- 4.3.1 S字型バイリニアスリップモデル
- 4.3.2 S字型トリリニアスリップモデル
- 4.3.3 バイリニア型スリップモデル
- 4.3.4 トリリニア型スリップモデル
- 4.3.5 ボックス型スリップモデル

これらの履歴モデルは、多くの研究者によって開発された。したがって、異なった手法で履歴が構築されている。どのようなアルゴリズムで、SPACE に組み込まれているかを理解し、新たな履歴モデルを開発する場合、読者が理解し易いアルゴリズムを参考にされると良い。

本節では、履歴特性としては、図 4-1 に示す最も単純なファイバーの履歴ルールであるバイリニア型について説明する。図中の記号は、

σ_1 : 降伏軸方向応力
ε_1 : 降伏ひずみ
E_1 : 初期剛性
E_2 : 第2剛性

である。

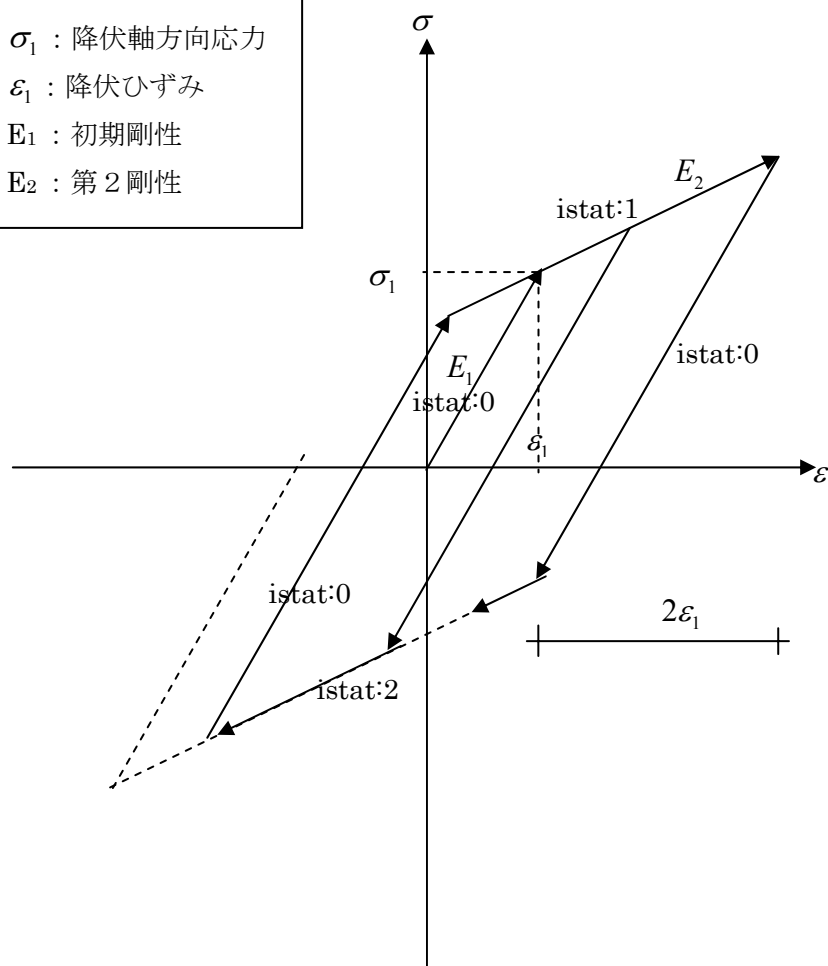


図 4-1 ファイバーモデルのバイリニア型履歴特性

履歴特性は骨格曲線と履歴ルールによって決定される。ここでは、最も単純なバイリニア Mesing 型の履歴特性について説明する。この履歴特性はスチールの履歴特性として用いられる。骨格曲線は図 4-1 に示すようなバイリニアであり、また、ファイバーの弾塑性状態は、istat:0 などと表されている。このバイリニア型では、状態 istat:0 は弾性状態を表し、状態 istat:1 は引張降伏以後の状態を、状態 istat:2 は圧縮降伏以後の状態を表す。また、除荷後の状態は、弾性状態と同じであるとする。

履歴特性をプログラム化する場合、全ての弾塑性状態を把握した後、各状態の特性を正確に理解することが大切である。特に、ある状態から他の状態に移り変わるルールと、その時点におけるパラメータなどの設

4.2 ファイバーの履歴

4.2.1 対称バイリニア型

定的確に行うことである。骨格曲線が折れ線で与えられている場合、状態が変移するとき、図 4-2 に示されるように必ず飛びこしが生じる。これを避けるために反復計算を導入して誤差を小さくすることも考えられるが、SPACE では、計算効率と解析誤差を考慮して、次の 3 方法のいずれかを用いている。

図 4-2 の左のタイプ（タイプ 1）は、剛性が強から弱に変移する場合で使用し、元の剛性で増分変位 du を求め、その後、弱の剛性で増分応力を求める。同図の中のタイプ（タイプ 2）は、除荷の場合で、剛性が弱から強に変移する場合に使用する。弱の剛性を用いて増分応力を求め、その後、強の剛性を用いて増分変位を計算する。図 4-2 の右のタイプ（タイプ 3）は、変移する前の剛性と後の剛性を用いてエネルギーを計算し、このエネルギーが同一となるように増分変位と増分応力を求める。

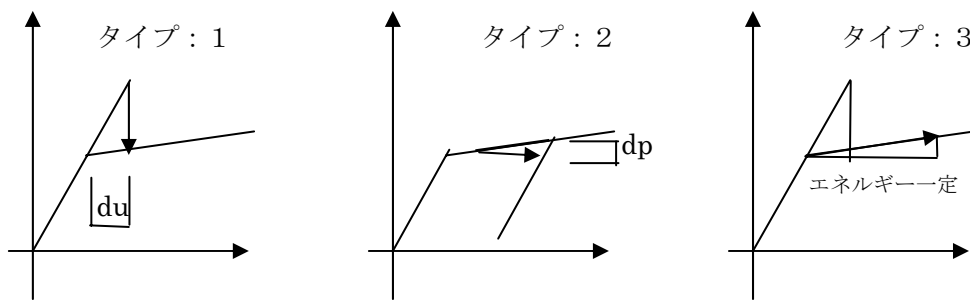


図 4-2 飛び越しの修正法

バイリニア Mesing 型では、弾塑性状態が 3 つに分類されており、その状態は以下のようなものである。

- 1) $istat:0$ 引張側と圧縮側の両方に境界を持つ。最初は、弾性状態であるため、その境界は降伏軸方向応力となるが、塑性状態に入った場合は新たに設定する必要がある。他の状態への移動は、引張側は $istat:1$ へ、圧縮側は $istat:2$ へとなる。
- 2) $istat:1$ 増分ひずみが引張ひずみである場合は境界がなく、除荷が生じてひずみが圧縮になると、その点が反曲点となり、弾性状態 $istat:0$ となる。ここで、 $istat:0$ 状態における両境界値を設定する必要がある。
- 3) $istat:2$ 増分ひずみが圧縮ひずみである場合は境界がなく、荷重が反転してひずみが引張となると、その点が反曲点となり、弾性状態 $istat:0$ となる。ここで、 $istat:0$ 状態における両境界値を設定

する必要がある。

次に、上記の履歴ルールにしたがって記述されたサブルーチン BiLinear() を具体的に見てみよう。

```

C
C  ● SUBROUTINE / BiLinear
C
C  ● BiLiner_Mesing 履歴モデル
C
subroutine BiLinear (AK, istat, AK_1, AK_2, Q_1, du, P, P1)
implicit real*8 (A-H, O-Z)
real*8 AK, AK_1, AK_2, Q_1, P, P1

c
c  AK          : 接線剛性
c  istat       : 現在の状態 (Work)
c  AK_1       : 第一勾配
c  AK_2       : 第二勾配
c  Q_1        : 第一折れ点のせん断力
c  du         : 増分変位
c  P          : 現在のせん断力 (Work)
c  P1         : istat=0 における反曲点上端 (Work)
c
c
100 continue
  if(istat.eq.0) then                                ! 1
    p = AK*du + p                                    ! 2
    if(du.gt.0.) then                                ! 3
      if(p.lt.p1) return                             ! 4
c-----★istat = 0 から istat=1 へ
    istat=1                                           ! 5
    du2= (p - p1)/AK
    AK=AK_2
    p = p1 + AK*du2
    return
  else                                                ! 6
    p2= p1 - 2.* Q_1                                  ! 7
    if(p.gt.p2) return                                ! 8
c-----★istat = 0 から istat=2 へ
    istat=2                                           ! 9
    du2= (p - p2)/AK                                  ! 10
    AK=AK_2                                           ! 11
    p = p2 + AK*du2                                   ! 12
    return
  endif
  elseif(istat.eq.1) then                            ! 13
    p = AK*du + p                                    ! 14
    if(du.ge.0.) return
c-----★istat =1 から istat=0 へ
    istat=0                                           ! 15
    dp = AK*du                                         ! 16
c
c 誤差が大きくなるため、除荷の場合、

```

```

c                                     一回は元の剛性を使用して、歪を計算する。
    AK=AK_1                                     ! 17
    du=dp/AK
    p=p-dp
    p1 = p
    p = AK*du + p
    return
elseif(istat.eq.2) then                                     ! 18
    p = AK*du + p
    if(du.le.0.) return                                     ! 19
c-----★istat =2 から istat=0 へ
    istat=0                                               ! 20
    dp = AK*du
    p=p-dp
    p1 = p + 2. * Q_1
    AK=AK_1
    du=dp/AK
    p = AK*du + p
    return
else
c-----★初期設定
    istat = 0                                               ! 21
    AK=AK_1
    p1=Q_1
    p=0.
    goto 100
endif
return
end

```

1. 状態が弾性の場合 (istat=0)、以下の処理を行う。
2. 増分ひずみと剛性を用いて増分応力を計算し、その値を増分前の応力に足しこむ。
3. 増分ひずみが引張の場合 (du>0)、以下の処理を行う。
4. 増分後の応力が引張側の境界値 p1 より小さい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
5. 増分後の応力が引張側の境界値 p1 より大きい場合、状態が istat:1 となる。そこで、飛び越し修正法のタイプ 1 を用いて状態が変化したときの処理を行う。まず、応力が境界値 p1 より飛び出た部分のひずみを求め、du2 にセットする。次に、接線剛性を第 2 勾配の値とし、さらに、その接線剛性と先ほどの飛び出した部分のひずみ du2 とで、増分応力を計算し、増分後の応力をセットする。その後このサブルーチンから戻る。
6. 増分ひずみが圧縮の場合、以下の処理を行う。
7. 現在の引張側の境界値と引張側降伏応力を用いて、圧縮側の境界値

- を求め、 p_2 にセットする。
8. 増分後の応力が境界値より大きい場合、弾性範囲内であることから、このサブルーチンより戻る。
 9. 増分後の応力が境界値より小さい場合、状態が変化し、 $istat:2$ となる。
 10. 飛び出た部分のひずみを求め、 du_2 にセットする。
 11. ユーザー設定の剛性第2勾配を接線剛性の値としてセットする。
 12. さらに、その接線剛性と先ほどの飛び出した部分のひずみ du_2 とで、増分応力を計算し、増分後の応力をセットする。その後、このサブルーチンから戻る。ここで、状態 $istat:0$ の処理が終了する。
 13. ここからは、状態 $istat:1$ の処理が始まる。
 14. 増分応力を計算し、増分後の応力を計算する。
 15. 増分ひずみ du が正の場合、つまり、引張ひずみの場合は、このサブルーチンより戻る。
 16. 増分ひずみ du が負の場合、つまり、除荷が生じた場合、状態パラメータ $istat$ を0に戻す。次に、タイプ2の飛び越し修正処理を用いて、増分変位と増分応力を修正する。まず、増分変位から増分応力を求める。
 17. 接線剛性を第1勾配の剛性にセットする。その剛性と先に求めた増分応力を用いて増分変位を求め直す。また、状態 $istat:0$ における境界値 p_1 をこの除荷位置の応力とする。求めた増分変位と増分応力より増分後の応力をセットし、サブルーチンから戻る。
 18. ここからは、状態 $istat:2$ の処理が始まる。増分応力を計算し、増分後の応力を計算する。
 19. 増分ひずみが負である場合、つまり、圧縮ひずみが進むとき、処理は終了し、サブルーチンより戻ることになる。
 20. 増分ひずみが正である場合、応力が反転するため、状態パラメータを0に戻す。次に、タイプ2の飛び越し修正処理を用いて、増分変位と増分応力を修正する。まず、増分応力を求め、増分後の応力から増分応力を引くことによって増分前の応力を求める。この増分前の応力から状態 $istat:0$ における境界値 p_1 を計算する。このとき、反転応力 p は圧縮応力であり、境界値 p_1 は引張側の境界値である。次に、接線剛性として第1勾配の剛性をセットし、その剛性を用いて先に求めた増分応力から増分変位を求める。この増分変位と第1勾配の剛性によって増分応力を求め、その増分応力を増分前の応力に足しこみ、増分後の応力を求める。これで、状態 $istat:2$ の処理

が終了し、サブルーチンから戻ることになる。

21. ここでは、このファイバーに関する初期設定を行う。まず、状態パラメータを $istat:0$ とする。次に、接線剛性 AK に第1勾配の剛性をセットする。また、現在の応力 p をゼロにし、この状態の引張側の境界値 $p1$ を降伏応力とする。以上で初期設定処理を終了し、処理1に戻る。

本節では、ファイバーに関する履歴ルールで、図4-3に示すトリリニア型について説明する。図中の記号は以下のようなものである。

4.2.2 対称トリリニア型

第1折れ点の応力	: Q_1
第2折れ点の応力	: Q_2
第1勾配	: AK_1
第2勾配	: AK_2
第3勾配	: AK_3
第1折れ点におけるひずみ	: $du1$
第2折れ点におけるひずみ	: $du2$

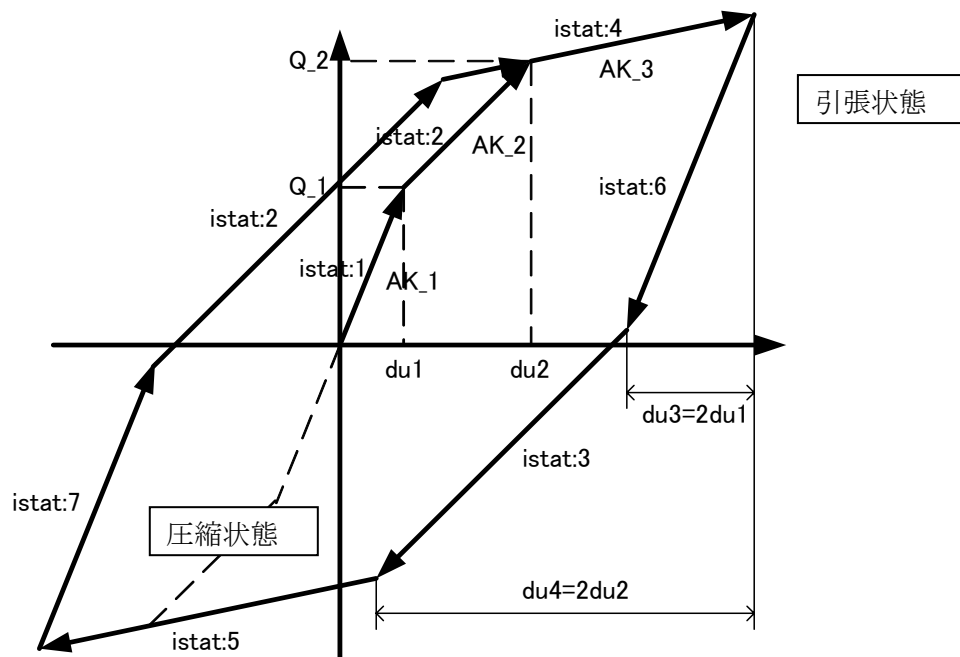


図4-3 ファイバーモデルのトリリニア型履歴特性

この履歴特性はスチールの履歴特性として用いられる。骨格曲線は図に示すようなトリリニアで、ファイバーの弾塑性状態は、istat:1は弾性状態を表し、istat:2とistat:4は引張降伏以後の状態を、状態istat:3とistat:5は圧縮降伏以後の状態を表す。また、除荷後の状態istat:6とistat:7は、弾性状態と同じであるとする。

※プレゼンターの断面表示では、第2折れ点以降を塑性状態としている。

この履歴ルールを2つに分けて検討する。

- 1) 第1折れ点を経験して除荷する履歴
- 2) 第2折れ点を経験して除荷する履歴

初めに、1)の第1折れ点を経験して除荷する履歴について、図4-4を用いて説明する。

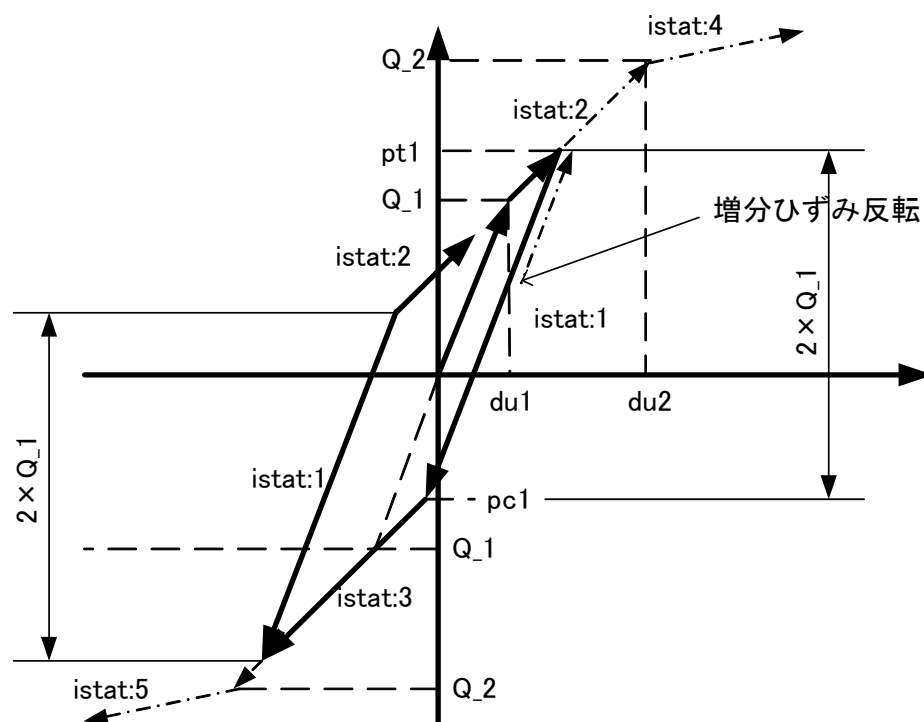


図4-4 第1折れ点を経験して除荷する履歴

応力が、 $Q_1 > \sigma > -Q_1$ の場合、通常の弾性範囲内と同様に履歴を描く。引張側で Q_1 を超えた場合、応力-ひずみの履歴は状態2の骨格線上に沿って動くが、その時点で増分ひずみが反転すると、その応力を $pt1$ として、 $pt1 > \sigma > pc1$ の間を直線的に動く事になる。ここでは状態1となり、 $pt1$ を越えると状態2へ、 $pc1$ を越えると状態3となる。

次に、2)の第2折れ点を経験して増分ひずみが反転する履歴について図4-5を用いて説明する。図4-5に示すように、ひずみが引張状態を進むと履歴は状態1から状態2へと骨格曲線上を移動し、 Q_2 を越えると状態4へと移動する事になる。状態4上で増分ひずみが反転すると、状態は6に入り、その時の応力を $pt2$ として、 $pt2$ から $p_length1$ を引いた値を $pc1$ とし、 $pc1$ から $p_length2$ を引いた値を $pc2$ として計算する。状態6においては $pt2 > \sigma > pc1$ の間を直線的に動く事になる。ひずみがそのまま進行し、 $pc1$ を越えると状態3に入り $\sigma > pc2$ の間で直線的に動く事になる。ここから増分ひずみが反転すると、状態は1に入り、反転せずに $pc2$ を越えると状態は5となる。また、ここで増分ひずみが反転すると状態は1となり、反転した時点での応力を $pc2$ とし、 $pc2$ から $p_length1$ を足した値を $pt1$ とし、 $pt1$ から $p_length2$ を足した値を $pc2$ として計算し履歴を描く事となる。

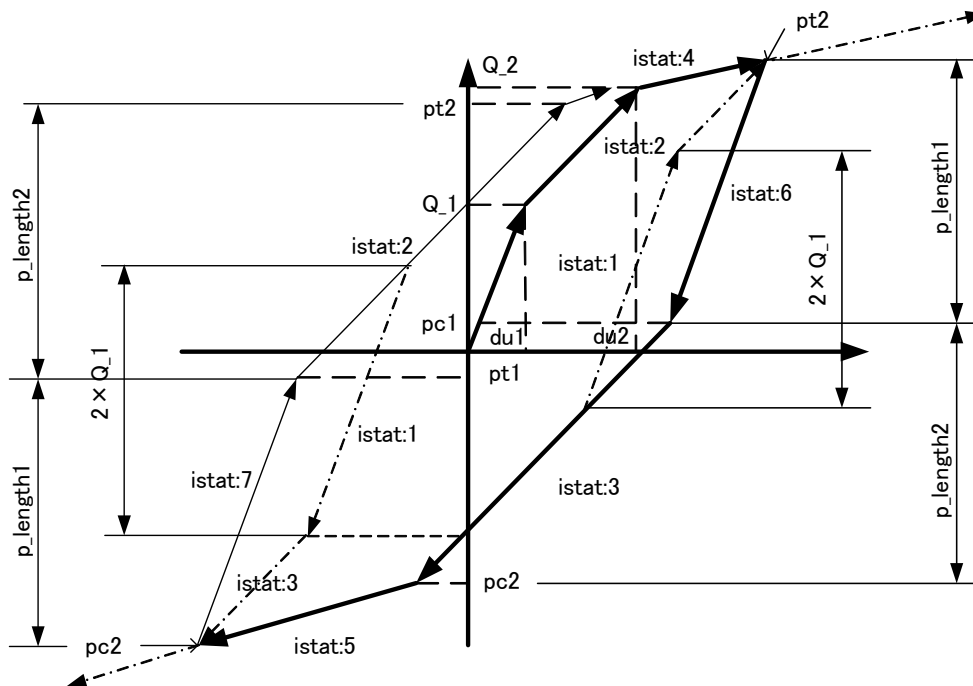


図4-5 第2折れ点を経験して除荷する履歴

以上の説明をまとめると、対称トリリニア型の履歴特性は、次の7つの状態で表現されることになる。

- istat : 1 弾性状態もしくは状態 2、3 から反転して戻るときの状態であり、初期の段階では $Q_1 > -Q_1$ の中で、釣合点は直線的に動く。 Q_1 を越えると状態 2 となり、 $-Q_1$ を越えると状態 3 となる。また、状態 2 もしくは状態 3 においてひずみが反転した時、その時の応力を状態 2 では $pt1$ とし、 $pt1$ から $2 \times Q_1$ を引いた値を $pc1$ 、状態 3 では反転した時の応力を $pc1$ とし、 $pc1$ から $2 \times Q_1$ を足した値を $pt1$ とする事によって、釣合点は、 $pc1 < -pc1$ の中で直線的に動く状態となる。
- istat : 2 引張側の第 2 勾配を移動する状態であり、境界値として増分ひずみが引張の場合 $Q_2 > 0$ もしくは、 $pt2 > 0$ で、 Q_2 もしくは $pt2$ を越えると状態は 4 となる。また、圧縮の場合はひずみが反転し、この時の応力を $pt1$ とし、状態は 1 となる。
- istat : 3 圧縮側の第 2 勾配を移動する状態であり、境界値として増分ひずみが圧縮の場合 $-Q_2 > 0$ もしくは、 $pc2 > 0$ で、 $-Q_2$ もしくは $pc2$ を越えると状態は 5 となる。また、引張の場合はひずみが反転し、この時の応力を $pc1$ とし、状態は 1 となる。
- istat : 4 引張側の第 3 勾配を移動する状態であり、増分ひずみが引張の時その境界はなく、増分ひずみが反転し圧縮となった時、その応力を $pt2$ として状態は 6 となる。この時状態 6 の境界である $pc1$ と状態 3 の境界である $pc2$ を計算する。
- istat : 5 圧縮側の第 3 勾配を移動する状態であり、増分ひずみが圧縮の時その境界はなく、増分ひずみが反転し引張となった時、その応力を $pc2$ として状態は 7 となる。この時状態 7 の境界である $pt1$ と状態 2 の境界である $pt2$ を計算する。
- istat : 6 ここでの接線勾配は状態 1 と同じ第 1 勾配であり、境界 $pt2 > pc1$ の間で直線的に動く事となる。引張の増分ひずみが進み $pt2$ を越えると状態は 4 に入り、圧縮の増分ひずみが進み $pc1$ を越えると状態は 3 となる。
- istat : 7 ここでの接線勾配は状態 1 と同じ第 1 勾配であり、境界 $pt1 > pc2$ の間で直線的に動く事となる。引張の増分ひずみが進み $pt1$ を越えると状態は 2 に入り、圧縮の増分ひずみが進み $pc2$ を越えると状態は 5 となる。

以上で、各状態の説明を終えるが、ここで、この履歴ルールを表現するサブルーチンを具体的に見ておこう。

```

c
C  ● SUBROUTINE /TriLinear (対称トリリニア型)
C
C  ● ファイバー要素の対称トリリニア履歴特性(ok)
C
      subroutine Trilinear (ak, istat, ak_1, ak_2, ak_3, Q_1, Q_2, du, p, pt1,
*                          pt2, pc1, pc2, p_length1, p_length2)
      implicit real*8 (A-H, O-Z)
c
c  AK          : 接線剛性
c  istat       : 現在の状態 (Work)
c  AK_1       : 第一勾配
c  AK_2       : 第二勾配
c  AK_3       : 第三勾配
c  Q_1        : 第一折れ点
c  Q_2        : 第二折れ点
c  du         : 増分変位
c  P          : 現在の応力
c  Pt1        : istat2における折れ点、及び、除荷した時の反曲点
c  Pt2        : istat4における折れ点、及び、除荷した時の反曲点
c  Pc1        : istat3における折れ点、及び、除荷した時の反曲点
c  Pc2        : istat5における折れ点、及び、除荷した時の反曲点
c
100  continue
      if (istat.eq.1) then! 1
C          ★ istat=1
          p=AK*du+p! 2
          if (du.gt.0) then! 3
              if (p.lt.pt1) return! 4
C          ★ istat=1 から istat=2 へ
              istat=2! 5
              du2=(p-pt1)/AK! 6
              AK=AK_2! 7
              p=AK*du2+pt1! 8
              return
              else! 9
                  if (p.gt.pc1) return! 10
C          ★ istat=1 から istat=3 へ
              istat=3! 11
              du2=(p-pc1)/AK
              AK=AK_2
              p=AK*du2+pc1
              return
              endif
C          ★ istat=2
              elseif (istat.eq.2) then! 12
                  p=AK*du+p
                  if (du.gt.0) then
                      if (p.lt.pt2) return
C          ★ istat=2 から istat=4 へ

```

```

    istat=4                                     ! 13
    du2=(p-pt2)/AK
    AK=AK_3
    p=AK*du2+pt2
    return
C-----★istat=2 から istat=1 へ
    else                                     ! 14
    istat=1                                   ! 15
    dp=AK*du                                 ! 16
    AK=AK_1                                 ! 17
    du=dp/AK                                ! 18
    p=p-dp                                  ! 19
    pt1=p                                   ! 20
    p=AK*du+p
    pc1=pt1-2*Q_1                             ! 21
    return
endif
C-----★istat=3
    elseif(istat.eq.3) then                 ! 22
    p=AK*du+p
    if(du.lt.0) then
    if(p.gt.pc2) return
C-----★istat=3 から istat=5 へ
    istat=5
    du2=(p-pc2)/AK
    AK=AK_3
    p=AK*du2+pc2
    pc1=p
    return
    else
C-----★istat=3 から istat=1 へ
    istat=1
    dp=AK*du
    AK=AK_1
    du=dp/AK
    p=p-dp
    pc1=p
    p=AK*du+p
    pt1=pc1+2*Q_1
    endif
    return
C-----★istat=4
    elseif(istat.eq.4) then                 ! 23
    p=AK*du+p
    if(du.gt.0) return                      ! 24
C-----★istat=4 から istat=1 へ
    istat=6                                  ! 25
    dp=AK*du
    AK=AK_1
    du=dp/AK
    p=p-dp
    pt2=p                                    ! 26
    pc1=-p_length1+pt2                     ! 27
    pc2=-p_length2+pc1                     ! 28

```

```

    p=AK*du+p
    return
C-----★ istat=5
    elseif(istat. eq. 5) then
    p=AK*du+p
    if(du. lt. 0) return
C-----★ istat=5 から istat=7 へ
    istat=7
    dp=AK*du
    AK=AK_1
    du=dp/AK
    p=p-dp
    pc2=p
    pt1=p_length1+pc2
    pt2=p_length2+pt1
    p=AK*du+p
    return
C-----★ istat=6
    elseif(istat. eq. 6) then
    p=AK*du+p
    if(du. gt. 0) then
    if(p. lt. pt2) return
C-----★ istat=6 から istat=2 へ
    istat=4
    du2=(p-pt2)/AK
    AK=AK_3
    p=AK*du2+pt2
    return
    else
    if(p. gt. pc1) return
C-----★ istat=6 から istat=3 へ
    istat=3
    du2=(p-pc1)/AK
    AK=AK_2
    p=AK*du2+pc1
    return
    endif
C-----★ istat=7
    elseif(istat. eq. 7) then
    p=AK*du+p
    if(du. gt. 0) then
    if(p. lt. pt1) return
C-----★ istat=7 から istat=2 へ
    istat=2
    du2=(p-pt1)/AK
    AK=AK_2
    p=AK*du2+pt1
    return
    else
    if(p. gt. pc2) return
C-----★ istat=7 から istat=5 へ
    istat=5
    du2=(p-pc2)/AK
    AK=AK_3

```

```

    p=AK*du2+pc2
    return
endif
C————★初期設定
    else ! 45
    istat=1
    AK=AK_1
    p=0
    pt1=Q_1
    pt2=Q_2
    pc1=-Q_1
    pc2=-Q_2
    du1=Q_1/AK_1 ! 46
    du2=(Q_2- Q_1)/AK_2+du1
    du3=2*du1
    du4=2*du2
    p_length1=du3*AK_1 ! 47
    p_length2=(du4-du3)*AK_2
    goto 100
    endif
end

```

1. 状態が $istat : 1$ の場合、以下の処理を行う。
2. 増分ひずみと剛性を用いて増分応力を計算し、その値を増分前の応力に足しこむ。
3. 増分ひずみが引張の場合 ($du > 0$)、以下の処理を行う
4. 増分後の応力が引張側の境界値 $pt1$ より小さい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
5. 増分後の応力が引張側の境界値 $pt1$ より大きい場合は、状態が $istat : 2$ となる。
6. 飛び越しの修正を行う為に、応力が境界地 $pt1$ から飛び出た分のひずみを求め、 $du2$ にセットする。
7. 状態が $istat : 2$ の時の接線剛性である第2勾配の値とする。
8. その接線剛性と先程の飛び出した部分のひずみ $du2$ とで、増分応力を計算し、増分後の応力をセットする。その後サブルーチンから戻る。
9. 増分ひずみが圧縮の場合、以下の処理を行う。
10. 増分後の応力が圧縮側の境界値 $pc1$ より大きい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
11. 増分後の応力が圧縮側の境界値 $pc1$ より小さい場合は、状態が $istat : 3$ となる。
12. 状態が $istat : 2$ の場合以下の処理を行う。

13. 増分後の応力が引張側の境界値 $pt2$ より大きい場合は、状態が $istat : 4$ となる。
14. 増分ひずみが負である場合、応力が反転するため、以下の処理を行う。
15. 状態を $istat : 1$ に戻す。
16. 増分応力を元の剛性を利用して修正する。
17. 接線剛性を第1勾配とする。
18. 修正した増分応力を用いて増分変位を修正する。
19. 増分後の応力から増分応力を引く事によって、増分前の応力を求める。
20. 求められた増分前の応力を $pt1$ にセットする。
21. $pt1$ から $2 \times Q_1$ を引く事によって、状態が $istat : 1$ における境界値 $pc1$ を計算する。
22. 状態が $istat : 3$ の場合以下の処理を行う。
23. 状態が $istat : 4$ の場合以下の処理を行う。
24. 増分ひずみが引張の時、その境界はなく、サブルーチンから戻る。
25. 増分ひずみが反転し圧縮となった時、状態が $istat : 6$ となる。
26. 反転した時の応力を $pt2$ にセットする。
27. 状態が $istat : 6$ における境界値を求めるため、 $pt2$ から $p_length1$ を引いた値を $pc1$ にセットする。
28. 反転後ひずみが圧縮で進行し、状態が $istat : 3$ となった時の境界値を求めるため、 $pc1$ から $p_length2$ を引いた値を $pc2$ にセットする。
29. 状態が $istat : 5$ の場合以下の処理を行う。
30. 増分ひずみが圧縮の時その境界はなく、サブルーチンから戻る。
31. 増分ひずみが反転し引張となった時、状態が $istat : 7$ となる。
32. 反転した時の応力を $pc2$ にセットする。
33. 状態が $istat : 7$ における境界値を求めるため、 $pc2$ から $p_length1$ を足した値を $pt1$ にセットする。
34. 反転後ひずみが引張で進行し、状態が $istat : 2$ となった時の境界値を求めるため、 $pt1$ から $p_length2$ を足した値を $pt2$ にセットする。
35. 状態が $istat : 6$ の場合以下の処理を行う。
36. 増分後の応力が引張側の境界値 $pt2$ より小さい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
37. 増分後の応力が引張側の境界値 $pt2$ より大きい場合は、状態が

- istat : 4 となり以下の処理を行う。
38. 増分ひずみが圧縮の時、増分後の応力が圧縮側の境界値 $pc1$ より大きい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
 39. 増分後の応力が圧縮側の境界値 $pc1$ より小さい場合は、状態が istat : 3 となり以下の処理を行う。
 40. 状態が istat : 7 の場合、以下の処理を行う。
 41. 増分応力が引張の時、増分後の応力が引張側の境界値 $pt1$ より小さい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
 42. 増分後の応力が引張側の境界値 $pt1$ より大きい場合は、状態が istat : 2 となり以下の処理を行う。
 43. 増分ひずみが圧縮の時、増分後の応力が圧縮側の境界値 $pc2$ より大きい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
 44. 増分後の応力が圧縮側の境界値 $pc2$ より小さい場合は、状態が istat : 5 となり以下の処理を行う。
 45. ここでは、ファイバーに関する初期設定を行う。まず状態パラメーターを istat : 1 とする。次に接線剛性 AK に第 1 勾配の剛性をセットする。また、現在の応力 p をゼロにし、初期状態での境界値をセットする。
 46. 除荷時の境界値の計算に用いる p_length を計算するために $du1 \cdot du2 \cdot du3 \cdot du4$ を計算する。
 47. $du3$ と $du4$ を用いて $p_length1$ と $p_length2$ を計算する。

4.2.3 直線コンクリート型

本節では、ファイバーに関する履歴ルールで、直線コンクリート型の履歴モデルについて説明する。図中の記号は以下のである。

圧縮と引張の第 1 勾配	AK_1
圧縮第 2 勾配	AK_2
圧縮第 3 勾配	AK_3
引張第 2 勾配	AK_4
引張強度	Q_1
圧縮側第 1 折れ点の応力	Q_2
圧縮強度	Q_3
圧縮流れ点	Q_4

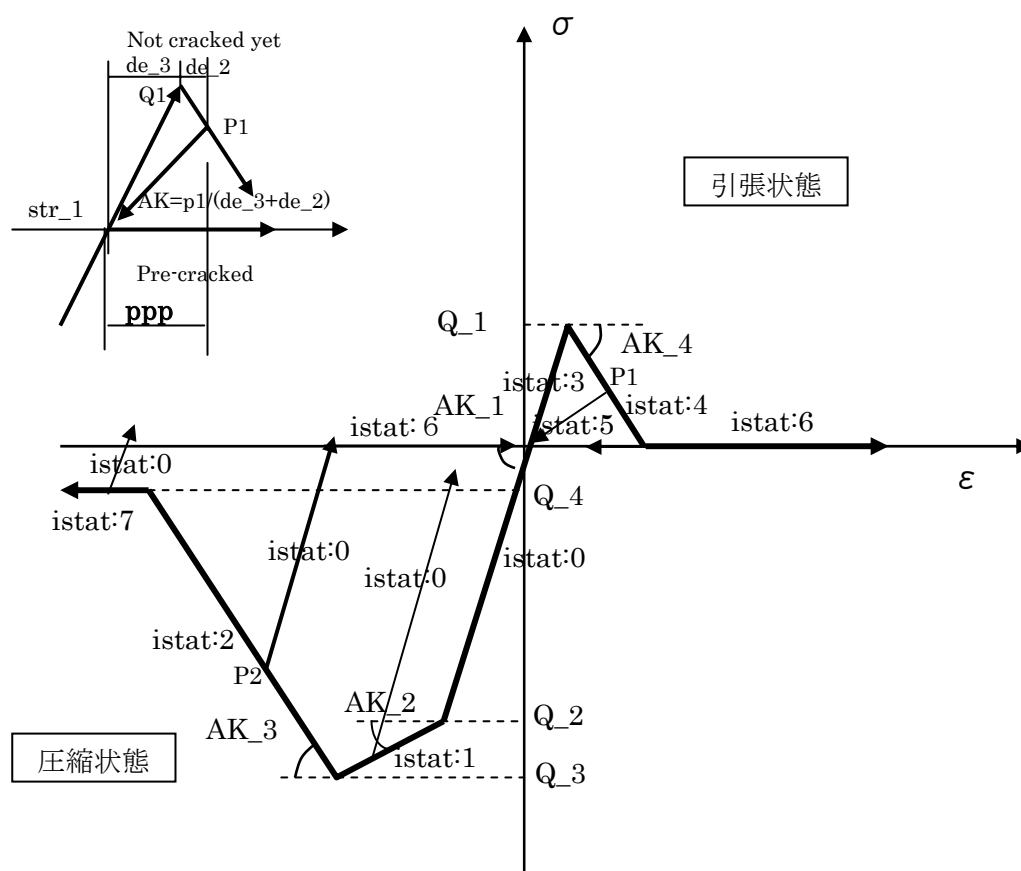


図 4-6 直線コンクリート型の履歴特性

コンクリートの履歴特性は、非常に複雑で、正確に模擬することはかなり難しい。例えば、骨格曲線はほとんど直線部分がなく、全てが非線形処理を行う必要がある。また、引張亀裂や圧縮破壊を経験すると、その後の履歴はそれ以前と大きく異なった挙動を示す。

SPACE の直線コンクリート型モデルは、このようなコンクリートの複雑な挙動を、近似的にしかも効率よく模擬できることを目的として開発されたものである。このモデルでは、コンクリートの履歴は全て直線で表す。また、履歴のルールは圧縮側と引張側の最大耐力を経験したかどうかで変化するため、複雑で理解し難い。そこで、ここでは履歴ルールを大きく 4 つに分けて検討する。

1. 圧縮側と引張側、共に最大耐力を経験していない履歴
2. 先に圧縮側の最大耐力を経験する履歴
3. 上記と逆で、先に引張側の最大耐力を経験する履歴
4. 両者共に、最大耐力を経験した後の履歴

履歴ルールを表すために、状態を表すパラメータ $istat$ の他に6つのパラメータが用いられており、それらが重要な役割を果たしている。その6つのパラメータとは、 $(str_1, P1)$ 、 $(str_2, P2)$ 、 $(ipret, ipre_c)$ である。括弧でくくられているパラメータは対になっており、一つ目は、引張側のひずみと応力、二つ目は圧縮側のひずみと応力、最後は引張側及び圧縮側の崩壊経験を表すパラメータである。ここで説明する履歴ルールの中で、どのようにこれらのパラメータが利用されているのか、また、実際のプログラムの中でどのように使われているかを理解されたい。

最初に、1) の圧縮側と引張側、共に最大耐力を経験していない場合について、図 4-7 を用いて説明する。

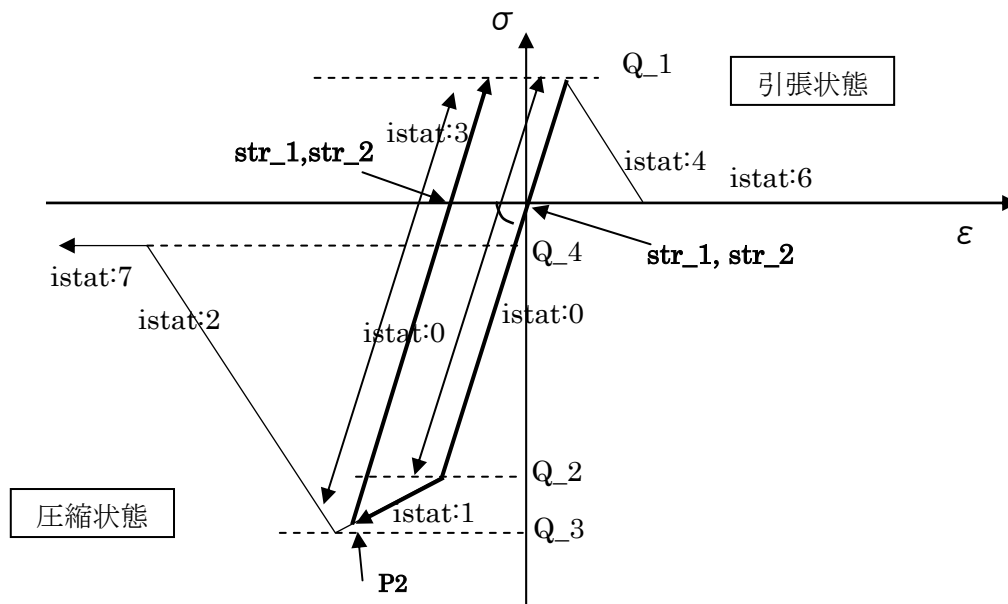


図 4-7 圧縮側と引張側、共に最大耐力を経験していない履歴

応力が、 $Q_1 > \sigma > Q_2$ の場合（引張側、圧縮側の最大耐力を表す $P1$ と $P2$ は、最初 Q_1 と Q_2 に設定されている）は、通常の弾性範囲と同様で、応力とひずみの関係を表す履歴は非常に単純である。圧縮側で、応力が Q_2 を超えた場合、履歴は状態 1 の骨格曲線上に沿って動くが、その時点で増分ひずみが反転すると、その応力を $P2$ として、 $Q_1 > \sigma > P2$ の間を直線的に動くことになる。コンクリートの崩壊経験を表すパラメータ $ipret$ （引張）と $ipre_c$ （圧縮）は、いずれも 0 である。また、状態 : 0 における直線のパラメータとして、その直線の引張側最大値は Q_1 、応力 0 におけるひずみは str_2 である。

次に、2) の先に圧縮側の最大耐力を経験する場合について説明する。

図 4-8 に示すように、ひずみが圧縮状態を進むと履歴は状態 1 から状態 2 へと骨格曲線上を移動し、圧縮側の最大耐力を経験する。状態 2 上で増分ひずみが反転すると、その応力を P_2 として、その後の履歴は $Q_1 > \sigma > P_2$ の間を直線で移動することになる。これは、前に説明した状態 1 からの反転した場合と同様である。ここでは、コンクリートの崩壊経験を表すパラメータ $ipre_c$ (圧縮) は 1 となる。また、状態 : 0 における直線のパラメータとして、未だ引張最大耐力を経験していないので、その直線の引張側最大値は P_1 、応力 0 におけるひずみは str_2 である。

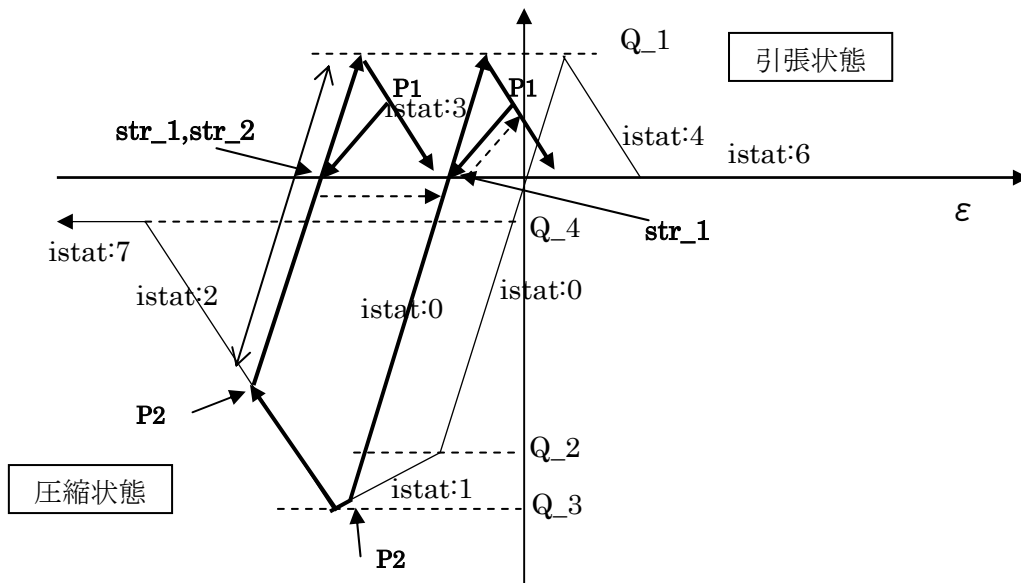


図 4-8 先に圧縮側の最大耐力を経験する履歴

ここで、先に圧縮側の最大耐力を超えてから、引張側の最大耐力を経験する場合について考える。無論、これ以前には引張側最大耐力を超える経験をしておらず、最初の引張側最大耐力経験である。もし、図 4-8 で示すように状態 1 から反転した履歴が、状態 0、状態 3 を通過した後、引張側の最大耐力を経験し、状態 4 において増分ひずみが再度圧縮になったとする。この時、最初に圧縮状態から引張状態に移ったときのひずみを str_1 とする。状態 4 において応力 P_1 で増分ひずみが圧縮となり、ひずみが反転することで状態 5 となる。この状態 5 では、ファイバーの接線剛性は、ひずみ str_1 と応力 P_1 を結ぶ直線の勾配となる。さらに同方向にひずみが進むと、ひずみ str_1 から状態 0 に戻るようになる。履歴が状態 5 から状態 0 に進む理由は、最初の引張側最大耐力経験であるため、ひずみ str_1 と str_2 が同じ値であることによる。この値が異なる場合については、次に説明する。

さらに、ひずみが進み、図 4-8 のように状態 0、1、2 へと進んだ後、

最後に、4)の図 4-10 で示す圧縮側も引張側も最大耐力を経験した後の履歴特性について説明する。まず、圧縮の状態 2 において応力 P_2 で増分ひずみが引張になり、反転して状態 0 となる場合を考えよう。さらに、ひずみが同方向に進行し、応力が 0 となって状態が 6 に変化する。この折れ曲がった点のひずみを str_2 として設定する。そのままひずみが進行すると、ひずみ str_1 に到達する。ここで、引張破壊パラメータ $ipret$ (引張) が 1 の場合は状態が 5 となり、応力 P_1 に向かってひずみが進行する。さらに、 P_1 を過ぎると状態 4 を経て、状態 6 へと移っていく。この時、引張破壊パラメータ $ipret$ (引張) は 2 となる。一方、既に引張破壊パラメータ $ipret$ (引張) が 2 の場合は、そのまま状態が 6 で変化せず、ひずみは同方向に進むことになる。

次に、逆進する場合について考える。引張破壊パラメータ $ipret$ (引張) が 1 の場合は、状態 4 で反転し、状態 5 となってひずみが str_1 に到達すると状態 6 となる。一方、引張破壊パラメータ $ipret$ (引張) が 2 の場合は、状態 6 に既に入っており、ひずみ str_1 を通過して、 str_2 まで進行する。次に、ひずみ str_2 で、圧縮側の剛性が発生し、状態 0 に変化する。さらに、ひずみが同方向に進行すると、応力 P_2 で状態 2 へと移り、最後にコンクリートの圧壊状態である状態 7 へと進行する。ここで、ひずみが反転すると状態 0 を経由して、状態 6 へと移っていく。この状態を経験すると引張側の剛性は期待できなくなり、 $ipre_c$ (圧縮) を 2 に設定する。

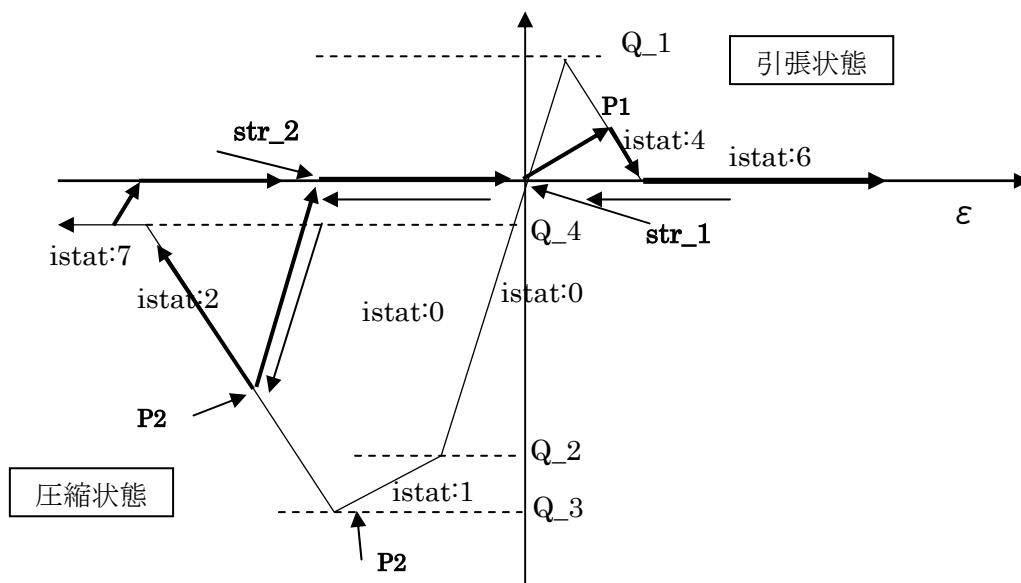


図 4-10 両者共に、最大耐力を経験した後の履歴

以上の説明をまとめると、直線コンクリート型の履歴特性は、次の8つの状態で表現される。

- istat : 0 圧縮側の弾性状態もしくは状態 1、2 から反転して戻るとき
の状態であり、境界値としては、 $P_2 < \sigma < 0$ である。圧縮側は、
骨格曲線との交点 P_2 の位置で異なり、状態 : 1 か 2 または 7
へと移動する。これは崩壊経験を表すパラメータ ipre_c (圧
縮) の値によって区別する。パラメータ ipre_c が 0 の場合、
状態 1 へ、ipre_c が 1 では状態 2 へ、ipre_c が 3 では状態 7
へ変化する。一方、引張側は以前に引張崩壊を経験している
か否かによって移動する状態が異なる。まず、経験していな
い場合は状態 : 3 に移動する。この時、応力 0 のひずみが str_1
および str_2 となる。状態 : 4 を経験し、しかも str_1 と str_2
が同じ場合は最大引張耐力 P_1 に向かって移動する。この状態
は 5 である。状態 : 6 を経験している場合は状態 : 6 へ移動す
る。これも、崩壊経験を表すパラメータ ipret (引張) の値に
よって区別する。パラメータ ipret が 0 の場合、状態 3 へ、
ipret が 1 でしかも str_1 と str_2 が同じ場合は状態 5 へ、
str_1 と str_2 が異なる場合もしくは ipret が 2 では状態 6 へ
変化する。
- istat : 1 圧縮側の第 2 勾配であり、境界値として増分ひずみが圧縮の
場合は $Q_3 < \sigma$ で、引張の場合はひずみが反転する。この場合、
状態 : 0 となる。
- istat : 2 圧縮側の第 3 勾配であり、特徴として接線剛性が負勾配を有
する。境界値として増分ひずみが圧縮の場合は $Q_4 > \sigma$ で、引
張の場合は、ひずみが反転する。この場合、状態 : 0 となる。
崩壊経験を表すパラメータ ipre_c は 1 となり、圧縮耐力を示
す応力 P_2 は、反転する応力となる。
- istat : 3 引張側の第 1 勾配であり、境界は $0 < \sigma < Q_1$ である。増分
ひずみが引張で応力が Q_1 を超える場合は、状態は 4 に移動
する。逆に、増分ひずみが圧縮の場合で、応力が 0 を超える
とき、str_1 と str_2 が同じ場合は状態 0 へ、str_1 と str_2
が異なる場合は状態 6 へ変化する。
- istat : 4 引張側の第 2 勾配であり、特徴として接線剛性が負勾配を有
する。境界値としては、 $0 < \sigma$ であり、増分ひずみが境界を越
えて大きくなると状態は 6 へと変化する。逆に、反転して圧

縮増分ひずみが生じるばあい、状態：5となる。状態：5における接線剛性は、状態：3で引張側に入ったひずみ str_1 に向かう直線より決定する。

istat:5 ここでの接線勾配は計算によって求める。境界値としては、圧縮側は、最大耐力 P_1 であり、引張側は、応力 0 である。圧縮側は状態：4になり、引張側は、ひずみが str_1 と str_2 等しいと状態：0となり、等しくない状態：6となる。

istat:6 引張側で勾配は 0 となる。境界値としては、引張側はなしで、圧縮側はひずみが str_2 になるまでである。ひずみが str_2 を超えると圧縮側では、弾性状態：0に変化する。

istat:7 圧縮側で勾配は 0 で応力値は Q_4 となる。圧縮側の境界値はなく、増分ひずみが反転して引張増分ひずみが生じると、弾性状態：0となる。この状態を経験すると引張側の剛性はないことになる。

以上で、各状態の説明を終えるが、ここで、この履歴ルールを表現するサブルーチンを具体的に見ておこう。

```

C
C  ● SUBROUTINE / Concrete (Ver. 3.01)
C
C  ● コンクリート履歴モデル
C
subroutine Concrete(AK, istat, AK_1, AK_2, AK_3, AK_4, Q_1, Q_2, Q_3, Q_4,
+                  de, P, P1, P2, IPRET, STR, STR_1, STR_2, IPRE_C)
implicit real*8 (A-H, O-Z)
C
C  AK          : 接線剛性
C  istat       : 現在の状態 (Work)
C  AK_1        : 圧縮 & 引張 第一勾配
C  AK_2        : 圧縮第二勾配
C  AK_3        : 圧縮第三勾配
C  AK_4        : 引張第二勾配
C  Q_1         : 引張強度
C  Q_2         : 圧縮第一折れ点の応力
C  Q_3         : 圧縮強度
C  Q_4         : 圧縮流れ点
C  de          : 増分ひずみ
C  P           : 現在の応力 (Work)
C  P1          : istat=4 における圧縮に切り替わったときの反曲点 (Work)
C  P2          : istat=1, 2 における引張に切り替わったときの反曲点 (Work)
C  IPRET       : 過去の引張履歴 (破壊したら 1, 流れたら 2) (Work)
C  IPRE_C      : 過去の圧縮履歴 (流れたら 1) (Work)
C  STR         : 現在のひずみ量 (Work)
C  STR_1       : 応力が最初に圧縮状態から引張状態になったときのひずみ (Work)

```

```

c   STR_2           : 最近の圧縮状態から引張状態になったときのひずみ(Work)
c
c
c   99 continue
c      go to (10, 100, 110, 120, 130, 140, 150, 160, 170), istat+2           ! 1
c
c   ----- ◎初期節点
c   10  str = 0.                                           ! 2
c      str_1 = 0.
c      str_2 = 0.
c      p=0.
c      AK = AK_1
c      ipret= 0
c      ipre_c=0
c      p1 = Q_1           ! 引張側の最大耐力 : 引張亀裂を経験すると低下する
c      p2 = Q_2           ! 圧縮側の最大耐力 : 圧壊を経験すると低下する
c      istat=0
c      if(de.gt.0.) istat=3                                     ! 3
c      goto 99
c
c   ----- ◎istat = 0
c   100 p = AK*de + p                                       ! 4
c      str = str + de
c      if(de.lt.0.) then                                     ! 圧縮側           ! 5
c      if(p.ge.p2) return                                     ! 圧縮側耐力より小       ! 6
c      if(ipre_c.eq.0) then                                   ! 7
c
c   ----- ★istat = 0 から istat=1 へ
c      istat=1
c      de_1= (p - p2)/AK                                     ! 8
c      AK=AK_2                                              ! 9
c      p = p2 + AK*de_1
c      elseif(ipre_c.eq.1) then                               ! 10
c
c   ----- ★istat = 0 から istat=2 へ
c      istat=2
c      de_1= (p - p2)/AK                                     ! 11
c      AK=AK_3                                              ! 12
c      p = p2 + AK*de_1
c      elseif(ipre_c.eq.2) then                               ! 13
c
c   ----- ★istat = 0 から istat=7 へ
c      istat=7
c      AK = AK_1*0.000000001
c      p = Q_4
c      endif                                               ! 15
c      else                                               ! 16
c      if(p.lt.0.) return                                     ! 17
c      if(ipret.eq.0) then                                   ! 18
c
c   ----- ★istat = 0 から istat=3 へ
c      istat=3
c      de_1 = p / AK           ! p は引張応力
c      str_1 = str - de_1
c      str_2 = str_1
c      elseif(ipret.eq.1) then                               ! 21
c      if(str.gt.str_1) then                                   ! 22
c
c   ----- ★istat = 0 から istat=5 へ
c      istat=5

```

```

    de_1 = p / AK ! 23
    de_2 =( p1 - Q_1 ) / AK_4 ! 24
    de_3 =Q_1 / AK_1 ! 25
    AK = p1 / ( de_2 + de_3 ) ! 26
    p = AK*de_1 ! 27
    else ! 28
c-----★istat = 0 から istat=6 へ
    istat=6
    de_1 = p / AK ! 29
    AK = AK_1*0.000000001
    p = 0.
    str_2 = str - de_1
    endif
    elseif(ipret.eq.2) then ! 30
c-----★istat = 0 から istat=6 へ
    istat=6
    de_1 = p / AK
    AK = AK_1*0.000000001
    p = 0.
    str_2 = str - de_1
    endif
    endif
    return
c-----◎istat = 1
110 p = AK*de + p ! 31
    str = str + de
    if(de.gt.0.) then ! 32
c-----★istat =1 から istat=0 へ
    istat=0
    pw=AK*de
    p = p - pw
    str = str - de
    p2 = p
    AK=AK_1
    de=pw/ak
    str = str + de
    p = AK*de + p
    return
    elseif(p.le.Q_3) then ! 33
c-----★istat =1 から istat=2 へ ! 34
    istat=2
    de_1 = (p - Q_3) / AK
    AK=AK_3
    p = Q_3 + AK*de_1
    ipre_c=1
    endif
    return
c-----◎istat = 2
120 p = AK*de + p ! 35
    str = str + de
    if(de.gt.0.) then ! 36
c-----★istat =2 から istat=0 へ
    istat=0
    p = p - AK*de ! 37

```

```

    pw=AK*de                                ! 38
    p2 = p                                  ! 39
    AK=AK_1
    str = str - de                          ! 40
    de=-pw/ak                               ! 41
    str = str + de
    p = AK*de + p
c-----★istat =0 におけるチェック
    if(ipret.le.1) then                      ! 42
    if(p.lt.p1) return                      ! 43
    str = str - de                          ! 44
    p = -AK*de + p                          ! 45
    pw=p1-p                                ! 46
    de=pw/AK
    str = str + de
    p=p1
    else                                    ! 47
    if(p.lt.0.) return                      ! 48
    str = str - de                          ! 49
    p = -AK*de + p
    pw= -p
    de=pw/ak
    str = str + de
    p=0.
    endif
    return                                  ! 50
    elseif(p.gt.Q_4)then                    ! 51
c-----★istat =2 から istat=7 へ
    istat=7                                ! 52
    AK = AK_1*0.000000001
    p = Q_4
    ipret = 2
    ipre_c = 2
    p2 = Q_4
    endif
    return                                  ! 53
c-----◎istat = 3
130 p = AK*de + p                          ! 54
    str = str + de
    if(p.gt.Q_1) then                       ! 55
c-----★istat =3 から istat=4 へ
    istat=4
    de_1 = (p - Q_1) / AK
    AK=AK_4
    p = AK*de_1 + Q_1
    ipret = 1
    elseif(p.lt.0.)then                    ! 56
c-----★istat =3 から istat=0 へ
    istat=0
    endif
    return
c-----◎istat = 4
140 p = AK*de + p                          ! 57
    str = str + de

```

```

    if(de.lt.0.) then                                     ! 58
c-----★istat =4 から istat=5 へ
    istat=5                                              ! 59
    p = p - AK*de
    p1 = p
    ppp=((str - str_1) - de)
    if(abs(ppp).gt.0.000000000001) then                  ! 60
    AK = p / ppp
    p = AK*de + p
    else                                              ! 61
    AK= AK_1
    p = AK*de + p
    endif
    if(p.lt.0. )then                                    ! 62
c-----★境界値の飛び越しチェック
    p = -AK*de + p
    str = str - de
    if(AK.ne.0.)then                                    ! 63
    de =-p/ak
    else                                              ! 64
    de=-(str-str_1)
    endif
    p=0.                                              ! 65
    str = str + de
    endif
    elseif(p.lt.0.)then                                ! 66
c-----★istat =4 から istat=6 へ
    istat=6                                              ! 67
    AK = AK_1*0.000000001
    p = 0.
    ipret = 2                                          ! 68
    endif
    return
c-----◎istat = 5
150 p = AK*de + p                                       ! 69
    str = str + de
    if(p.gt.p1) then                                   ! 70
c-----★istat =5 から istat=4 へ
    istat=4                                              ! 71
    de_1 = (p - p1) / AK
    AK=AK_4
    p = AK*de_1 + p1
    elseif(p.lt.0.)then                                ! 72
    if(str.lt.str_2)then
c-----★istat =5 から istat=0 へ
    istat=0                                              ! 73
    str = str_2
    AK=AK_1
    else                                              ! 74
c-----★istat =5 から istat=6 へ
    istat=6                                              ! 75
    AK = AK_1*0.000000001
    p = 0.
    endif

```

```

    endif
    return
c-----◎istat = 6
160 p = 0. ! 76
    str = str + de
    if(str.lt.str_2) then ! 77
c-----★istat =6 から istat=0 へ
    istat=0
    de=str-str_2
    str = str_2
    AK=AK_1
    endif
    if(ipret.eq.1) then ! 78
    if(str.lt.str_1) return ! 79
c-----★istat = 6 から istat=5 へ
    istat=5 ! 80
    de_1 = str - str_1
    de_2 =( p1 - Q_1 ) / AK_4
    de_3 =Q_1 / AK_1
    AK = p1 / ( de_2 + de_3 )
    p = AK*de_1
    endif
    return
c-----◎istat = 7
170 p = Q_4 ! 81
    str = str + de
    if(de.gt.0.) then ! 82
c-----★istat =7 から istat=0 へ
    istat=0
    AK=AK_1
    str = str - de ! 弾性復帰は一回無視する ! 83
    endif
    return
end

```

1. 状態パラメータにしたがって処理を分類する。状態パラメータが-1のときは初期設定を行う。
2. ここからは初期設定を行う。まず各種のワーク用データをゼロクリアする。接線剛性を第一剛性とする。状態パラメータを0にする。
3. 増分変位が正の場合は引張応力となり、状態パラメータを3にする。負の場合は状態パラメータを0とする。
4. ここからは状態パラメータが0の場合の処理を行う。圧縮側の弾性部分の処理を行う。まず、増分ひずみより増分後の応力を計算する。増分後のひずみもセットする。
5. まず、増分ひずみをチェックし、以降では圧縮の場合の処理を行う。増分ひずみが引張の場合、処理は16. に飛ぶ。
6. 増分後の応力が圧縮耐力 P2 より小さい場合は、この状態内であると

- して処理を終了し、このサブルーチンから戻る。
7. 増分後の応力が圧縮耐力より大きい場合は、これ以降では状態変化に関する処理を行う。まず、過去の圧縮履歴 $ipre_c$ をチェックして、コンクリートが圧壊していない場合は、状態パラメータを 1 にセットする。
 8. 飛び越した部分のひずみを計算する。
 9. 接線剛性に圧縮側第 2 勾配の剛性をセットする。また、飛び越した部分のひずみと圧縮側第 2 勾配の剛性を用いて、増分後の応力を再計算する。
 10. 過去の圧縮履歴 $ipre_c$ が 1 でコンクリートが圧壊した経験を有する場合は以下の処理を行う。
 11. 状態パラメータを 2 にセットする。
 12. 飛び越した部分のひずみを計算する。接線剛性に圧縮側第 3 勾配の剛性をセットする。また、飛び越した部分のひずみと圧縮側第 3 勾配の剛性を用いて、増分後の応力を再計算する。
 13. 過去の圧縮履歴 $ipre_c$ が 2 でコンクリートが完全圧壊した経験を有する場合は、以下の処理を行う。
 14. 状態パラメータを 7 にセットし、接線剛性をほとんどゼロにする。また、増分後の応力は第 3 折れ点の応力とする。
 15. ここで、状態パラメータが 0 で圧縮側の処理が終了する。
 16. ここからは、状態パラメータが 0 で引張側の処理を行う。
 17. 増分後の応力が圧縮の場合、この状態内であるとして処理を終了し、このサブルーチンより戻る。その他は以降の処理へ移る。
 18. 過去の引張履歴 $ipret=0$ で、引張亀裂が生じていない場合は、以下の処理を行う。
 19. 引張側の状態になったとして、状態パラメータを 3 にセットする。引張側のひずみ de_1 を計算する。
 20. 現在のひずみから引張側にひずみ de_1 を引いて、引張応力が働きの出したときのひずみを str_1 として保存する。また、初期設定として、圧縮状態で引張破壊して、応力が流れ出したときのひずみ str_2 として、 str_1 をコピーする。
 21. 過去の引張履歴 $ipret=1$ で、引張亀裂が生じている場合は、以下の処理を行う。
 22. 現在のひずみが str_1 より大きい場合は状態パラメータが 5 となる。
 23. 引張側のひずみ de_1 を計算する。
 24. 引張側最大耐力 Q_1 、過去の反転位置の応力 $P1$ 並びに状態 4 の剛

- 性勾配を用いて、最大耐力点から反転位置までのひずみ de_2 を計算する。
25. 原点から引張最大耐力までのひずみ de_3 を計算する。
 26. 引張応力へ入る位置から、反転する位置までの直線の傾き、つまり、状態5の接線剛性 AK を計算する（図4-6の左上の図参照）。
 27. 引張方向に飛び出したひずみと接線剛性を用いて、増分後の応力を求める。
 28. 現在のひずみが str_1 より小さい場合は、状態パラメータは6となる。
 29. 引張破壊して応力が流れ出したときのひずみ str_2 を、現在のひずみから飛びだしたひずみを引いて設定する。
 30. 過去の引張履歴 $ipret=2$ で、応力が流れた状態の場合は、以下の処理を行う。状態パラメータは6となる。引張破壊して流れ出したときのひずみ str_2 は、現在のひずみから飛び出したひずみを引いて設定する。ここまでで、状態：0に関する処理が終了する。
 31. ここからは、状態：1で圧縮側第2勾配に関する処理を行う。増分後の応力と増分後のひずみを計算する。
 32. 増分ひずみが正となると、ひずみが反転する場合であり、弾性状態である $istat=0$ とする。また、一旦増分後の応力とひずみを増分前の応力とひずみに戻す。反転位置の応力 $P2$ にこの増分前の応力をセットする。接線剛性を圧縮側の第1勾配の剛性をセットする。前に求めた増分応力を用いて増分ひずみを計算する。この値より増分後の応力とひずみを計算し直す。ここでサブルーチンより戻る。
 33. 応力が第2折れ点の応力より小さい場合、この第2折れ点を飛び越したことになる、以下の処理を行う。
 34. 状態パラメータを $istat=2$ とする。飛び出した部分のひずみ de_1 を計算し、接線剛性に圧縮側第3勾配の剛性をセットする。この飛び出した部分のひずみ de_1 と接線剛性を用いて増分後の応力を計算する。最後に圧縮崩壊を表すパラメータを $ipre_c=1$ とする。これで状態：1の処理が終了し、このサブルーチンから戻ることになる。
 35. ここからは、状態2の処理を行う。まず、増分ひずみを用いて増分後の応力を計算する。
 36. 増分ひずみが正のとき、ひずみは反転し、状態は0に戻る。
 37. 一旦、増分前の応力を計算する。
 38. 増分応力を計算する。
 39. 状態2で、反転する位置の応力を $P2$ として保存する。また、接線

剛性に圧縮第1勾配の剛性をセットする。

40. 増分前のひずみに戻す。
41. 新しい接線剛性を用いて、増分ひずみを計算する。ただし、前の接線剛性と符号が逆なので、増分ひずみに負符号を付ける。この増分ひずみを用いて、増分後のひずみと応力を計算する。
42. ここでは、反転して状態0となるが、飛び出したひずみがありにも大きいとき、状態0の境界を越えてしまう場合がある。このような場合に備えて以下の処理を行う。まず、引張側の崩壊パラメータが2でない場合の処理を行う。
43. このとき、増分後の応力が境界値 P_1 を超えない場合は、状態に変化なしとして、このサブルーチンより抜ける。
44. 増分後の応力が境界値 P_1 を超える場合は、応力とひずみを増分前に戻す。
45. 飛び出した応力 p_w を求める。
46. それに対応した増分ひずみを計算する。最後に、増分後の応力として P_1 にセットする。
47. 引張側の崩壊パラメータが2の場合、以降の処理を行う。
48. 増分後の応力が境界値である0より小さいとき、このサブルーチンより戻る。
49. 飛び出した応力 p_w を求め、それに対応した増分ひずみを計算する。最後に、増分後の応力として0をセットする。
50. ここまでで、状態2の反転処理を終了する。
51. 状態2で、増分ひずみが負の場合の処理を行う。まず、増分後の応力と境界応力 Q_4 を比較し、大きいときは状態2の範囲内であることよりサブルーチンから抜ける。
52. 小さいときは状態2の範囲から抜け、状態7に移る。状態パラメータを $istat=7$ とする。接線剛性をほとんど0とする。増分後の応力を Q_4 とする。また、崩壊パラメータを $ipret=2$ とする。また、圧縮側の崩壊パラメータも $ipre_c=2$ とする。最後に p_2 の値を Q_4 とする。
53. これで、状態2の処理を終了する。
54. ここからは、状態3についての処理を行う。まず、増分後の応力を計算し、また、増分後のひずみも計算する。ここでの境界値応力は、 Q_1 と0である。
55. 増分後の応力が引張側の最大耐力 Q_1 より大きい場合は、以下の処理を行う。まず、状態パラメータを $istat=4$ とし、飛び出したひず

- み de_1 を計算する。接線剛性を引張側の第2勾配の剛性をセットし、増分後の応力を再計算する。ここで、引張側の崩壊パラメータである $ipret$ を1にセットする。
56. 一方、増分後の応力が0より小さい場合は、状態を0にセットする。接線剛性は、状態3と同じであることから変更しない。
57. ここからは、状態4についての処理を行う。まず、増分後の応力と増分後のひずみを計算する。引張増分ひずみに対しては、境界値応力は0で状態が6に変化する。また、増分圧縮ひずみが生じると反転して状態5となる。
58. 増分ひずみが負の場合、反転処理を行う。
59. まず、状態パラメータを $istat=5$ とする。増分前の応力を計算し、その値を $P1$ とする。さらに、増分前のひずみからひずみ str_1 を引き、図4-6の左上の図に示す ppp の値を求める。
60. 求めた ppp がゼロでない場合は、接線剛性を計算し、その値を用いて増分後の応力を求める。
61. 求めた ppp がほとんどゼロの場合は（この場合はほとんどない）接線剛性として弾性剛性の K_1 をセットする。その値を用いて増分後の応力を求める。
62. ここで計算した応力と境界応力である0とを比較し、増分応力が大きくて境界応力を飛び越した場合は、次の処理を行う。まず、増分前の応力とひずみを求める。
63. 次に、増分ひずみを求めるが、接線剛性 AK がゼロであるかどうかチェックし、ゼロでない場合は割り算して増分ひずみを求める。
64. 接線剛性がゼロの場合は（ $P1$ がほとんど0の場合）、予備のためにこの式で増分ひずみをセットしておく。
65. 増分後の応力を0にセットし、同様に増分後のひずみをセットする。
66. 一方、増分ひずみが正の場合は、増分後の応力が境界応力である0より小さくなって飛び越した場合は、次の処理を行う。
67. 状態パラメータを $istat=6$ にセットする。接線剛性をほとんどゼロにする。増分後の応力を0にする。
68. 引張側の崩壊パラメータを $ipret=2$ にセットする。これで、状態4の処理は全て終了する。
69. ここからは、状態5についての処理を行う。まず、増分後の応力を計算し、また、増分後のひずみも計算する。
70. 応力 p が $P1$ より大きいとき、状態は4に変化する。
71. 状態パラメータを $istat=4$ にする。次に、飛び出したひずみ de_1

- を求め、接線剛性を引張側の第2勾配の剛性にセットする。この剛性と飛び出したひずみを用いて、応力を再計算する。
72. 一方、応力が0より小さくなる場合については、以下の処理を行う。
この状態は、増分ひずみが圧縮で、しかも、状態5の境界値である応力0を超えた場合に相当する。
73. 計算した増分後のひずみが str_2 を飛び越えて小さくなる場合は、以下の処理を行い、まだ、飛び越えない場合は、処理75へ制御が移る。ここでは状態6となる。
74. 状態パラメータを $istat=0$ とする。増分後のひずみを str_2 に変更する。また、接線剛性を圧縮の第1勾配の剛性にセットする。
75. 状態パラメータを $istat=6$ とする。増分後の応力を0に変更する。また、接線剛性をほとんどゼロにセットする。
76. ここからは、引張崩壊状態：6の処理を行う。まず、応力を0とする。次に、増分後のひずみを計算する。また、ここでの境界値は、増分ひずみが引張状態で、しかも、境界ひずみ str_2 を飛び越して小さくなったときである。
77. 増分後のひずみが境界値ひずみ str_2 より小さくなったときは、再度圧縮剛性が発生するため、状態を $istat=0$ に戻す。増分後のひずみを str_2 とする。これは、増分ひずみ de が非常に大きくなって、応力とひずみの関係を図化したとき、ユーザーに誤解を与えないためである。接線剛性を圧縮側の第1勾配の剛性を用いる。
78. ここからの処理は、増分ひずみが引張に対するものであり、まず、引張側の崩壊パラメータが $ipret=1$ の場合、つまり、引張剛性が多少残っている場合に以下の処理を行う。ただし、 $ipret=2$ の場合は、このサブルーチンより戻る。
79. 増分後のひずみが、引張開始点のひずみ str_1 より小さい場合は、サブルーチンから戻る。このひずみを飛び越すと引張側の剛性が発生して以下の処理を行うことになる。
80. 状態を5とし、飛び越したひずみ de_1 を計算する。次に、図4-6の左上の図を参考にして、ひずみ de_2 、 de_3 を求め、それらを利用して、状態5における剛性を計算する。さらに、増分後の応力も計算する。
81. ここからは、圧縮崩壊状態：7の処理を行う。まず、この状態の応力は一定であり、 Q_4 である。また、増分後のひずみを計算する。
82. 増分ひずみが負であれば、ひずみが進行したとして、サブルーチンを抜ける。また、正であれば、ひずみが反転したことより、状態：0

に変化する。接線剛性に圧縮側第1勾配の剛性をセットする。

83. 弾性復帰する場合、この一回の増分ひずみを足しこむことを取りやめる。これは、計算誤差によって状態0と状態:7が交互に現れるとき、図として表示する場合、ひずみの履歴が微細な振動状態を呈するように見えるのを防ぐためである。

コンクリートの履歴特性は非常に複雑であり、骨格曲線には直線部分がない。ここで、この骨格曲線を曲線で表す履歴を作成する予定である。しかしながら、この曲線型の履歴は様々な履歴が提案されており、今後SPACEに用いる履歴を検討していきたい。

4.2.4 曲線コンクリート型

4.2.5 対称バイリニア型(移動+等方硬化用)

本節では、ファイバーの履歴ルールで図 4-11 に示す移動+等方硬化用の対称バイリニア型について説明する。図中の記号は以下の通りである。

第1折れ点の応力	: Q_1
第1勾配	: AK_1
第2勾配	: AK_2
塑性応力増分量	: dpt dpc

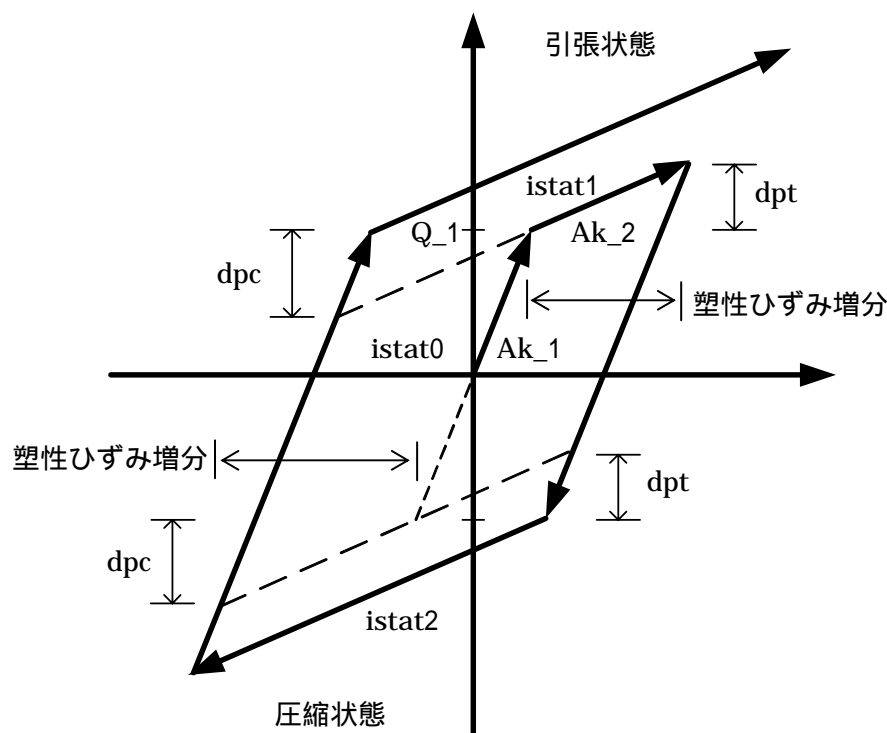


図 4-11 ファイバーモデルの対称バイリニア型 (移動+等方硬化用) 履歴モデル

この履歴特性は、移動硬化を考慮した 4.2.1 節の対称バイリニア型履歴特性に等方硬化を考慮したもので、スチールの履歴特性として用いられる。骨格曲線は図に示すようなバイリニアで、除荷前に塑性ひずみ増分があると塑性応力増分量 dpt 、 dpc だけ硬化する。ファイバーの弾塑性状態は、 $istat:0$ は弾性状態を表し、 $istat:1$ は引張降伏以後の状態を、 $istat:2$ は圧縮降伏以後の状態を表す。また、除荷後の状態は、弾性状態と同じであるとする。

この履歴ルールを大きく分けて下記の 2 つに分けて検討する。

- 1) 塑性ひずみ増分を有して除荷する履歴
- 2) 塑性ひずみ増分を有せず除荷する履歴

1)の塑性ひずみ増分を有して除荷する履歴について図4-12を用いて、
2)の塑性ひずみ増分を有せず除荷する履歴について図4-13を用いて説明する。図中の記号は以下の通りである。

現在の応力	: p	現在のひずみ	: u_g
最大引張側ひずみ	: sut	最大圧縮側ひずみ	: suc
istat:1 における折れ点、及び、除荷した時の反曲点 : $Pt1$			
istat:2 における折れ点、及び、除荷した時の反曲点 : $Pc1$			

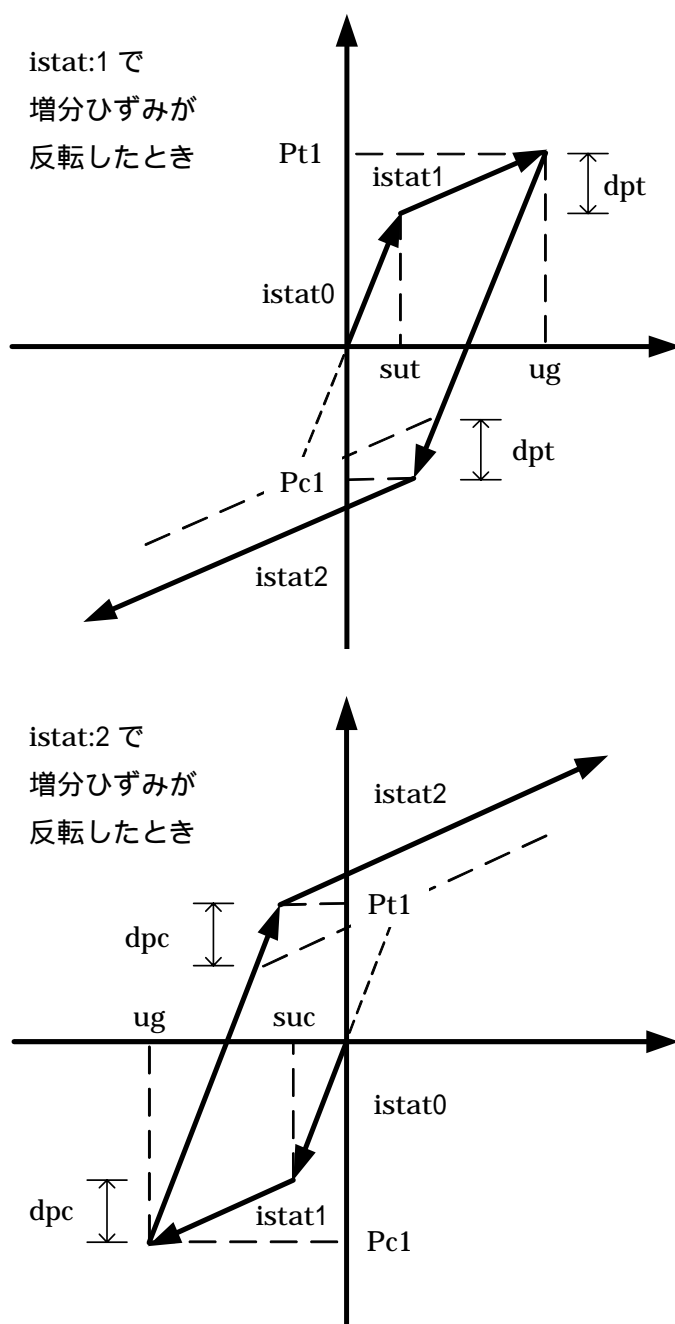


図 4-12 塑性ひずみ増分を有して除荷する履歴

まず、1) について説明する。istat:1 で増分ひずみが反転した場合、現在のひずみ u_g がそれまでの最大引張ひずみ s_{ut} を超えるとき、塑性応力増分量 dpt だけ反曲点を硬化させる（ただし s_{ut} の初期設定は、第1折れ点時のひずみに設定する）。反転時の応力を P_{t1} とし、除荷した時の反曲応力を P_{c1} とすると、 P_{c1} の計算方法は、 P_{t1} から $2 \times Q_1$ と今までの塑性応力増分量の総和 dpp を引いて求める。

状態 istat:2 の時も同様に、増分ひずみが反転した場合、現在のひずみ u_g がそれまでの最大圧縮ひずみ s_{uc} を超えるとき、塑性応力増分量 dpc だけ硬化させる（ただし s_{uc} の初期設定は、第1折れ点時のひずみに設定する）。反転時の応力を P_{c1} とし、除荷した時の反曲応力を P_{t1} とすると、 P_{t1} は、 P_{c1} から $2 \times Q_1$ と dpp を足して求める。

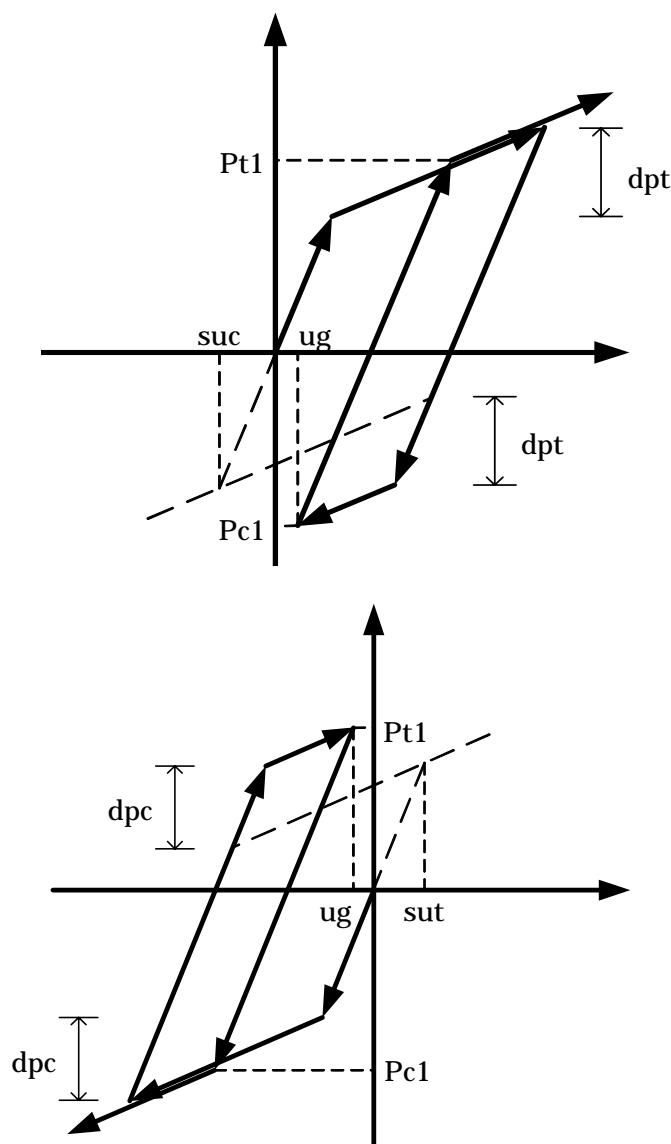


図 4-13 塑性ひずみ増分を有せず除荷する履歴

次に 2) について説明する。istat:1 で増分ひずみが反転したとき、現在のひずみ u_g がそれまでの最大引張ひずみ s_{ut} を超えないと塑性ひずみ増分はゼロと考える。塑性応力増分量 $d_p t$ もゼロとする(ただし s_{ut} の初期設定は、第 1 折れ点時のひずみに設定する)。反転時の応力を P_{t1} とし、除荷した時の反曲応力を P_{c1} とすると、 P_{c1} の計算方法は、 P_{t1} から $2 \times Q_1$ と今までの塑性応力増分量の総和 d_{pp} を引いて求める。

Istat:2 の時も同様に、増分ひずみが反転したとき、現在のひずみ u_g がそれまでの最大圧縮ひずみ s_{uc} を超えなければ塑性応力増分量 d_{pc} はゼロとする(ただし s_{uc} の初期設定は、第 1 折れ点時のひずみに設定する)。反転時の応力を P_{c1} とし、除荷した時の反曲応力を P_{t1} とすると、 P_{t1} は、 P_{c1} から $2 \times Q_1$ と d_{pp} を足して求める。

以上の説明をまとめると、対称バイリニア型(移動+等方硬化用)の履歴特性は、次の 3 つの状態で表現される。

- 1) istat:0 引張側と圧縮側の両方に境界を持つ。最初は、弾性状態であるため、その境界は降伏軸方向応力となるが、塑性状態に入った場合は新たに設定する必要がある。他の状態への移動は、引張側は istat:1 へ、圧縮側は istat:2 へとなる。
- 2) istat:1 増分ひずみが引張ひずみである場合は境界がなく、圧縮ひずみの場合は、その点が反曲点となり、弾性状態 istat:0 となる。ここで、istat:0 状態における両境界値を、塑性ひずみ増分を有する場合とそうでない場合に分けて設定する必要がある。
- 3) istat:2 増分ひずみが圧縮ひずみである場合は境界がなく、引張ひずみの場合は、その点が反曲点となり、弾性状態 istat:0 となる。ここで、istat:0 状態における両境界値を、塑性ひずみ増分を有する場合とそうでない場合に分けて設定する必要がある。

次に、上記の履歴ルールにしたがって記述されたサブルーチン BiLinear_CH() を具体的に見てみよう。

```

c
c      SUBROUTINE/BiLiner Combined Hardening ( 対称バイリニア型  移動+等方硬化用)
c
c      subroutine BiLinear_h(AK,istat,AK_1,AK_2,Q_1,du,P,Pt1,Pc1,
*
c      dpp,ug,sut,suc)
c      implicit real*8(A-H,O-Z)
c
c      AK      : 接線剛性
c      istat   : 現在の状態
c      AK_1    : 第一勾配

```

```

c      AK_2                      : 第二勾配
c      Q_1                      : 第一折れ点
c      du                      : 増分ひずみ
c      p                      : 現在の応力
c      pt1                    : istat:1 における折れ点、及び、除荷した時の反曲点
c      pc1                    : istat:2 における折れ点、及び、除荷した時の反曲点
c      ug                      : 現在のひずみ
c
100  continue
c                                istat=0
      if(istat.eq.0)then                                ! 1
      p=AK*du+p                                          ! 2
      if(du.gt.0.)then                                  ! 3
      if(p.lt.Pt1)return                                ! 4
c                                istat=0 から istat=1 へ
      istat=1                                           ! 5
      du2=(p-Pt1)/AK
      AK=AK_2
      p=Pt1+AK*du2
      return
      else                                              ! 6
      if(p.gt.Pc1)return                                ! 7
c                                istat=0 から istat=2 へ
      istat=2                                           ! 8
      du2=(p-Pc1)/AK
      AK=AK_2
      p=Pc1+AK*du2
      return
      endif
c                                istat=1
      elseif(istat.eq.1)then                            ! 9
      p=AK*du+p                                          ! 10
      if(du.gt.0.)return
c                                istat=1 から istat=0 へ
      istat=0                                           ! 11
      dp=AK*du                                          ! 12
      AK=AK_1
      du=dp/AK
      p=p-dp
      if(ug.gt.sut)then                                ! 13
      dpt=(ug-sut)*AK_2                                ! 14
      dpp=dpp+dpt                                       ! 15
      sut=ug                                            ! 16
      Pt1=p                                             ! 17
      Else                                             ! 18
      Pt1=p                                             ! 19
      endif
      Pc1=Pt1-2*Q_1-dpp                                ! 20
      p=AK*du+Pt1                                       ! 21
      return
c                                istat=2
      elseif(istat.eq.2)then                            ! 22
      p=AK*du+p
      if(du.lt.0)return                                ! 23

```

```

c                                istat=2 から istat=0 へ
    istat=0                                ! 24
    dp=AK*du
    AK=AK_1
    du=dp/AK
    p=p-dp
    if (ug.lt.suc) then                                ! 25
        dpc=(suc-ug)*AK_2                                ! 26
        dpp=dpp+dpc                                ! 27
        suc=ug                                ! 28
        Pc1=p                                ! 29
    else                                ! 30
        Pc1=p                                ! 31
    endif
    Pt1=Pc1+2*Q_1+dpp                                ! 32
    p=AK*du+Pc1                                ! 33
    return

c                                初期設定
    else                                ! 34
        istat=0
        AK=AK_1
        p=0
        Pt1=Q_1
        Pc1=-Q_1
        sut=Q_1/AK_1
        suc=-Q_1/AK_1
        dpp=0
        dpt=0
        dpc=0
        goto 100
    endif
    return
end

```

1. 状態が弾性的の場合 ($istat=0$) 以下の処理を行う。
2. 増分ひずみと剛性を用いて増分応力を計算し、その値を増分前の応力に足しこむ。
3. 増分ひずみが引張の場合 ($du>0$) 以下の処理を行う。
4. 増分後の応力が引張側の境界値 $Pt1$ より小さい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
5. 増分後の応力が引張側の境界値 $Pt1$ より大きい場合、状態が $istat:1$ となる。そこで、飛び越し修正法のタイプ 1 を用いて状態が変化したときの処理を行う。まず、応力が境界値 $Pt1$ より飛び出た部分のひずみを求め、 $du2$ にセットする。次に、接線剛性を第 2 勾配の値とし、さらに、その接線剛性と先ほどの飛び出した部分のひずみ $du2$ とで、増分応力を計算し、増分後の応力をセットする。その後このサブルーチンから戻る。

6. 増分ひずみが圧縮の場合、以下の処理を行う。
7. 増分後の応力が境界値より大きい場合、弾性範囲内であることから、このサブルーチンより戻る。
8. 増分後の応力が境界値より小さい場合、状態が変化し、 $istat:2$ となる。飛び出た部分のひずみを求め、 $du2$ にセットする。ユーザー設定の剛性第2勾配を接線剛性の値としてセットする。さらに、その接線剛性と先ほどの飛び出した部分のひずみ $du2$ とで、増分応力を計算し、増分後の応力をセットする。その後、このサブルーチンから戻る。
9. ここからは、状態 $istat:1$ の処理が始まる。増分応力を計算し、増分後の応力を設定する。
10. 増分ひずみ du が正の場合、つまり、引張ひずみの場合は、このサブルーチンより戻る。
11. 増分ひずみ du が負の場合、つまり、除荷が生じた場合、状態パラメータ $istat$ を0に戻す。
12. 次に、増分ひずみと増分応力を修正する。まず、増分応力を求める。接線剛性を第1勾配の剛性にセットする。その剛性と先に求めた増分応力を用いて増分ひずみを求め直す。増分応力より折れ点の応力を求め直す。
13. 現在のひずみが最大引張側ひずみより大きいとき ($ug>sut$)、以下の処理を行う。
14. まず、塑性応力増分量を求める。
15. 今までの塑性応力増分量に今回の塑性応力増分量を足し、塑性応力増分量の総和 dpp を求める。
16. 現在のひずみを sut とする。
17. 現在の応力を $Pt1$ とする。
18. 現在のひずみが最大引張側ひずみより小さいとき ($ug<sut$)、以下の処理をする。
19. 現在の応力を $Pt1$ とする。
20. 除荷した時の反曲応力 $Pc1$ を、 $Pt1$ から $2 \times Q_1$ と今までの塑性応力増分量の総和 dpp を引いて求める。
21. 12で設定した接線剛性と飛び出した部分のひずみ、増分応力を計算し、増分後の応力をセットする。
22. ここからは、状態 $istat:2$ の処理が始まる。増分応力を計算し、増分後の応力を計算する。
23. 増分ひずみが負である場合、つまり、圧縮ひずみが進むとき、処理

- は終了し、サブルーチンより戻ることになる。
24. 増分ひずみが正である場合、応力が反転するため、状態パラメータを 0 に戻す。次に、増分ひずみと増分応力を修正する。増分応力を求め、増分後の応力から増分応力を引くことによって増分前の応力を求める。
 25. 現在のひずみが最大圧縮側ひずみより大きいとき ($u_g < u_{suc}$) 以下の処理を行う。
 26. まず、塑性応力増分量を求める。
 27. 今までの塑性応力増分量に塑性応力増分量を足し、塑性応力増分量の総和 dpp を求める。
 28. 現在のひずみを u_{suc} とする。
 29. 現在の応力を P_{c1} とする。
 30. 現在のひずみが最大圧縮側ひずみより小さいとき ($u_g > u_{suc}$) 以下の処理をする。
 31. 現在の応力を P_{c1} とする。
 32. 載荷した時の反曲応力 P_{t1} を、 P_{c1} から $2 \times Q_1$ と今までの塑性応力増分量の総和 dpp を足して求める。
 33. 24 で設定した接線剛性と飛び出した部分のひずみ、増分応力を計算し、増分後の応力をセットする。
 34. ここでは、このファイバーに関する初期設定を行う。まず、状態パラメータを $istat:0$ とする。次に、接線剛性 AK に第 1 勾配の剛性をセットする。また、現在の応力 p をゼロにし、この状態の引張側の境界値 P_{t1} を圧縮側の境界地 P_{c1} を降伏応力とする。最大引張側ひずみ u_{ut} 、最大圧縮側ひずみ u_{suc} の初期設定は、第 1 折れ点時のひずみに設定する。塑性応力増分量 P_{t1} 、 P_{c1} 、全塑性応力増分量 dpp をゼロにセットする。以上で初期設定処理を終了し、処理 1 に戻る。

本節では、ファイバーに関する履歴ルールで、図 4-14 に示す移動 + 等方硬化用の対称トリリニア型について説明する。図中の記号は以下のようである。

第 1 折れ点の応力	: Q_1
第 2 折れ点の応力	: Q_2
第 1 勾配	: AK_1

4.2.6 対称トリリニア型(移動 + 等方硬化用)

第2勾配	: AK_2
第3勾配	: AK_3
第1折れ点におけるひずみ	: du1
第2折れ点におけるひずみ	: du2
塑性応力増分量	: dpt dpc

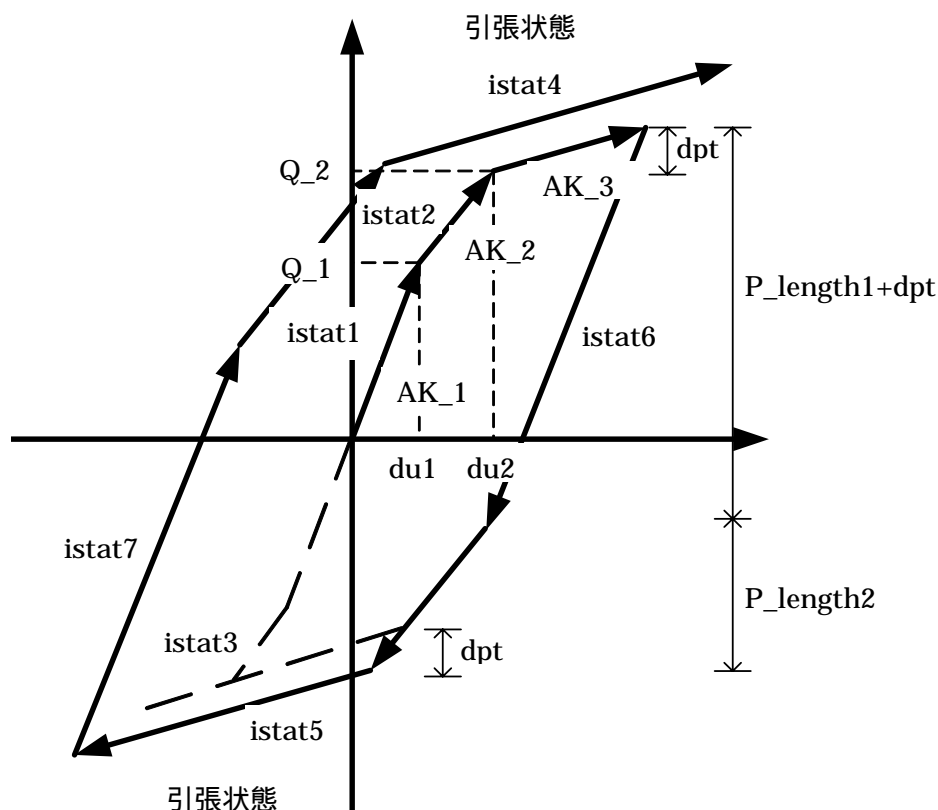


図 4-14 ファイバーモデルの対称トリリニア型（移動＋等方硬化用）履歴モデル

この履歴特性は、移動硬化を考慮した4.2.2節の対称トリリニア型履歴特性に等方硬化を考慮したもので、スチールの履歴特性とし用いられる。骨格曲線は図4-14に示すようなトリリニアで、除荷前に塑性ひずみ増分があると塑性応力増分量だけ硬化する。ファイバーの弾塑性状態は、istat:1は弾性状態を表し、istat:2とistat:4は引張降伏以後の状態を、istat:3とistat:5は圧縮降以後の状態を表す。また、除荷後のistat:6とistat:7は、弾性状態と同じであるとする。

プレゼンターの断面表示では、第2折れ点以降を塑性状態としている。この履歴ルールを大きく分けて下記の3つに分けて検討する。ここで

は、2)と3)について説明する。

- 1) 第2折れ点を経験せず除荷する履歴
- 2) 第2折れ点を経験して、塑性ひずみ増分を有して除荷する履歴
- 3) 第2折れ点を経験して、塑性ひずみ増分を有せず除荷する履歴

まず2)の第2折れ点を経験して、塑性ひずみ増分を有して除荷する履歴について図4-15を用いて説明する。

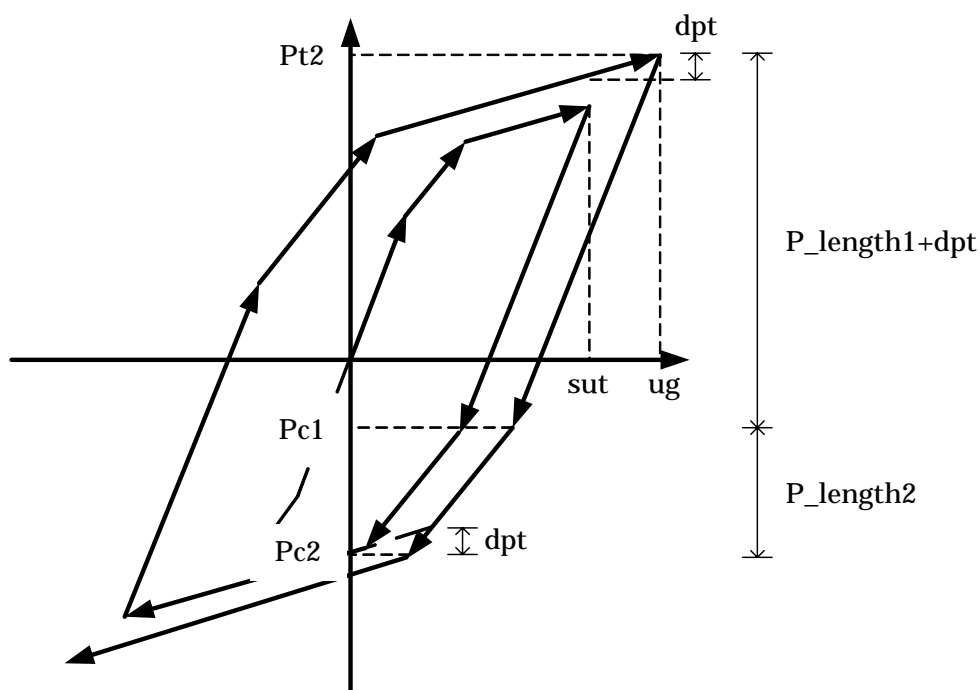


図4-15 第2折れ点を経験して、塑性ひずみ増分を有して除荷する履歴

図4-15に示すように、ひずみが引張状態を進むと履歴は $istat:1$ から $istat:2$ へと骨格曲線上を移動し、 Q_2 を越えると $istat:4$ へと移動する事になる。 $istat:4$ 上で増分ひずみが反転したとき、現在のひずみ ug がそれまでの最大引張ひずみ sut を超えたなら塑性応力増分量 dpt だけ反曲点 $pc1$ を硬化させる(ただし sut の初期設定は、第二折れ点時のひずみに設定する)。 $pc1$ の計算方法は、 $Pt2$ から $P_length1$ と今までの塑性応力増分量総和を引いて求める。 $Pc2$ は $Pc1$ から $P_length2$ を引いて求める。 $istat:6$ においては $Pt2 > Pc1$ の間を直線的に動く。ひずみがそのまま進行し、 $Pc1$ を超えると状態3に入り、 $Pc2$ の間で直線的に

動くことになる。ここから増分ひずみが反転すると、 $istat:1$ に入り、反転せずに $Pc2$ を超えると $istat:5$ となる。また、ここで増分ひずみが反転するとき、現在のひずみ ug がそれまでの最大圧縮ひずみ suc を超えたなら塑性応力増分量 dpc だけ反曲点 $pt1$ を硬化させる（ただし suc の初期設定は、第二折れ点時のひずみに設定する）。 $Pt1$ の計算方法は、 $Pc2$ から $P_length1$ と今までの塑性応力増分量総和 dpp を足して求める。 $Pt2$ は $Pt1$ に $P_length2$ を足して求める。

最後の 3) 第二折れ点を経験して、塑性ひずみ増分を有せず除荷する履歴について図 4-16 を用いて説明する。

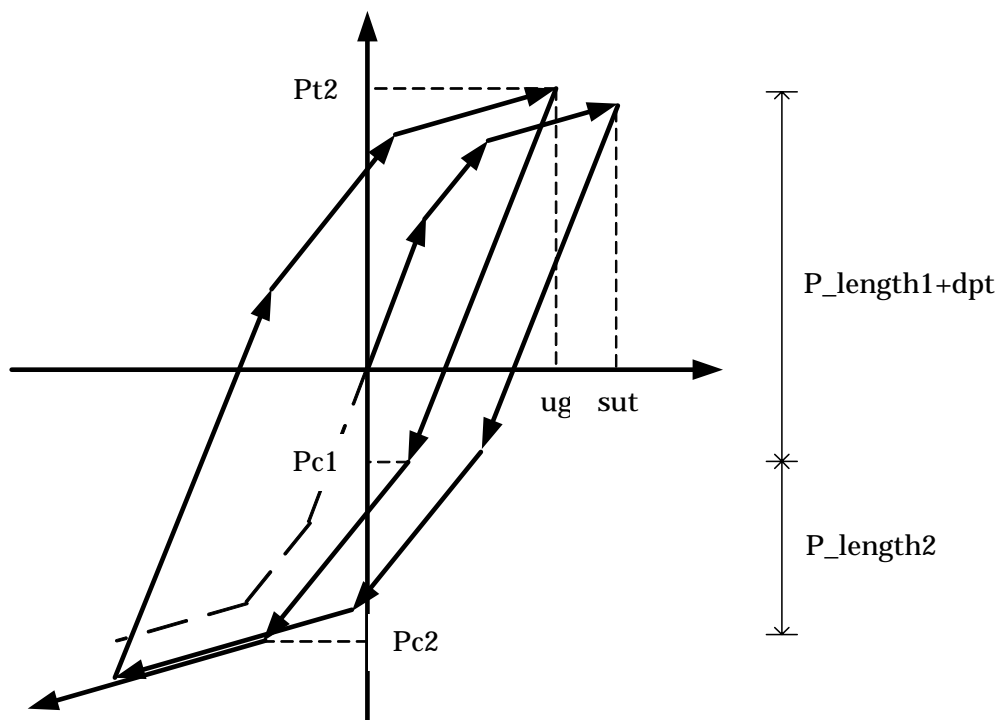


図 4-16 第 2 折れ点を経験して、塑性ひずみ増分を有せず除荷する履歴

$istat:4$ で増分ひずみが反転したとき、現在のひずみ ug がそれまでの最大引張りひずみ sut を超えないと塑性ひずみ増分はゼロと考える。塑性応力増分量 dpt もゼロとする。（ただし sut の初期設定は、第二折れ点時のひずみに設定する）。反転時の応力を $Pt2$ とし、除荷した時の反曲応力を $Pc1$ とする。 $Pc1$ の計算方法は、 $Pt2$ から $P_length1$ と今までの塑性応力増分量の総和 dpp を引いて求める。 $Pc2$ は $Pc1$ から $P_length2$ を引いて求める。

$istat:5$ の時も同様に、増分ひずみが反転したとき、現在のひずみ ug

がそれまでの最大圧縮ひずみ suc を超えてなければ塑性応力増分量 dpc はゼロとする (ただし suc の初期設定は、第一折れ点時のひずみに設定する)。反転時の応力を $Pc2$ とし、除荷した時の反曲応力を $Pt1$ とする。 $Pt1$ は、 $Pc2$ に $P_length1$ と dpp を足して求める。 $Pt2$ は $Pt1$ に $P_length2$ を足して求める。

以上の説明をまとめると、対称トリリニア型 (移動 + 等方硬化用) の履歴特性は、次の7つの状態で表現される。

- istat:1 弾性状態もしくは istat:2、istat:3 から反転して戻るときの状態であり、初期の段階では $Q_1 > -Q_1$ の中で直線的に動く。 Q_1 を越えると istat:2 となり、 $-Q_1$ を越えると istat:3 となる。また、istat:2 もしくは istat:3 においてひずみが反転した時、その時の応力を istat:2 では $Pt1$ とし、 $Pt1$ から $P_length1$ と dpp を引いた値を $Pc1$ 、istat:3 では反転した時の応力を $Pc1$ とし、 $Pc1$ から $P_length1$ と dpp を足した値を $Pt1$ とする事によって、 $Pc1 < Pt1$ の中で直線的に動く状態となる。
- istat:2 引張側の第2勾配であり、境界値として増分ひずみが引張の場合 $Q_2 >$ もしくは、 $Pt2 >$ で、 Q_2 もしくは $Pt2$ を越えると istat:4 となる。また、圧縮の場合はひずみが反転し、この時の応力を $Pt1$ とし、istat:1 となる。
- istat:3 圧縮側の第2勾配であり、境界値として増分ひずみが圧縮の場合 $-Q_2 >$ もしくは、 $Pc2 >$ で、 $-Q_2$ もしくは $Pc2$ を越えると istat:5 となる。また、引張の場合はひずみが反転し、この時の応力を $Pc1$ とし、istat:1 となる。
- istat:4 引張側の第3勾配であり、増分ひずみが引張の時その境界はなく、増分ひずみが反転し圧縮となった時、その応力を $Pt2$ とし、istat:6 となる。塑性ひずみ増分を有する場合とそうでない場合に分けて、istat:6 の境界である $Pc1$ と istat:3 の境界である $Pc2$ を計算する必要がある。
- istat:5 圧縮側の第3勾配であり、増分ひずみが圧縮の時その境界はなく、増分ひずみが反転し引張となった時、その応力を $Pc2$ とし、istat:7 となる。塑性ひずみ増分を有する場合とそうでない場合に分けて、istat:7 の境界である $Pt1$ と istat:2 の境界である $Pt2$ を計算する必要がある。
- istat:6 ここでの接線勾配は istat:1 と同じ第1勾配であり、境界 $Pt2 > Pc1$ の間で直線的に動く事となる。引張の増分ひずみが進み

Pt2 を越えると istat:4 に入り、圧縮の増分ひずみが進み Pc1 を越えると istat:3 となる。

istat:7 ここでの接線勾配は istat:1 と同じ第1勾配であり、境界 Pt1> >Pc2 の間で直線的に動く事となる。引張の増分ひずみが進み Pt1 を越えると istat:2 に入り、圧縮の増分ひずみが進み Pc2 を越えると istat:5 となる。

以上で、各状態の説明を終えるが、ここで、この履歴ルールを表現するサブルーチンを具体的に見ておこう。

```

c
c      SUBROUTINE/TriLinear Combined Hardening
c              ( 対称トリリニア型 移動 + 等方硬化用)
c
c      subroutine TriLinear_h(ak,istat,ak_1,ak_2,ak_3,Q_1,Q_2,du,p,pt1,
*          pt2,pc1,pc2,p_length1,p_length2,dpp,ug,sut,suc)
c      implicit real*8(A-H,O-Z)
c
c      ak          : 接線剛性
c      istat       : 現在の状態
c      ak_1        : 第一勾配
c      ak_2        : 第二勾配
c      ak_3        : 第三勾配
c      Q_1         : 第一折れ点
c      Q_2         : 第二折れ点
c      du          : 増分ひずみ
c      p           : 現在の応力
c      pt1         : istat2 における折れ点、及び、除荷した時の反曲点
c      pt2         : istat4 における折れ点、及び、除荷した時の反曲点
c      pc1         : istat3 における折れ点、及び、除荷した時の反曲点
c      pc2         : istat5 における折れ点、及び、除荷した時の反曲点
c      ug          : 現在のひずみ
c
c      100 continue
c
c              istat=1
c      if(istat.eq.1)then
c          p=ak*du+p
c          if(du.gt.0)then
c              if(p.lt.pt1) return
c          istat=1 から istat=2 へ
c          istat=2
c          du5=(p-pt1)/ak
c          ak=ak_2
c          p=ak*du5+pt1
c          return
c      else
c          if(p.gt.pc1)return
c          istat=1 から istat=3 へ
c      istat=3

```

```

    du5=(p-pc1)/ak
    ak=ak_2
    p=ak*du5+pc1
    return
endif

c                                istat=2                                ! 12
elseif(istat.eq.2)then
    p=ak*du+p
    if(du.gt.0)then
    if(p.lt.pt2)return
c                                istat=2 から istat=4 へ
    istat=4                                ! 13
    du5=(p-pt2)/ak
    ak=ak_3
    p=ak*du5+pt2
    return
c                                istat=2 から istat=1 へ
    else                                ! 14
    istat=1                                ! 15
    dp=ak*du                                ! 16
    ak=ak_1                                ! 17
    du=dp/ak                                ! 18
    p=p-dp                                ! 19
    pt1=p                                ! 20
    p=ak*du+p
    pc1=pt1-2*Q_1-dpp                                ! 21
    return
endif
c                                istat=3                                ! 22
elseif(istat.eq.3)then
    p=ak*du+p
    if(du.lt.0)then
    if(p.gt.pc2)return
c                                istat=3 から istat=5 へ
    istat=5
    du5=(p-pc2)/ak
    ak=ak_3
    p=ak*du5+pc2
    pc1=p
    return
    else
c                                istat=3 から istat=1 へ
    istat=1
    dp=ak*du
    ak=ak_1
    du=dp/ak
    p=p-dp
    pc1=p
    p=ak*du+p
    pt1=pc1+2*Q_1+dpp                                ! 23
    endif
    return
c                                istat=4                                ! 24
elseif(istat.eq.4)then

```

```

        p=AK*du+p
        if (du.gt.0) return                                ! 25
c                                     istat=4 から istat=6 へ
        istat=6                                           ! 26
        dp=ak*du
        ak=ak_1
        du=dp/ak
        p=p-dp
        if (ug.gt.sut) then                                ! 27
            dpt=(ug-sut)*ak_3                              ! 28
            dpp=dpp+dpt                                    ! 29
            sut=ug                                         ! 30
            pt2=p                                          ! 31
        else                                              ! 32
            pt2=p                                          ! 33
        endif
        pc1=-p_length1+pt2-dpp                            ! 34
        pc2=-p_length2+pc1                                ! 35
        p=ak*du+p                                          ! 36
        return
c                                     istat=5
        elseif (istat.eq.5) then                          ! 37
            p=ak*du+p
            if (du.lt.0) return                            ! 38
c                                     istat=5 から istat=7 へ
            istat=7                                        ! 39
            dp=AK*du
            AK=ak_1
            du=dp/ak
            p=p-dp
            if (ug.lt.suc) then                            ! 40
                dpc=(suc-ug)*ak_3                          ! 41
                dpp=dpp+dpc                                ! 42
                suc=ug                                      ! 43
                pc2=p                                       ! 44
            else                                           ! 45
                pc2=p                                       ! 46
            endif
            pt1=p_length1+pc2+dpp                          ! 47
            pt2=p_length2+pt1                              ! 48
            p=ak*du+p                                       ! 49
            return
c                                     istat=6
        elseif (istat.eq.6) then                          ! 50
            p=ak*du+p
            if (du.gt.0) then
                if (p.lt.pt2) return                      ! 51
c                                     istat=6 から istat=2 へ
                istat=4                                    ! 52
                du5=(p-pt2)/ak
                ak=ak_3
                p=ak*du5+pt2
                return
            else

```

```

      if(p.gt.pc1)return                                ! 53
c      istat=6 から istat=3 へ                          ! 54
      istat=3
      du5=(p-pc1)/ak
      ak=ak_2
      p=ak*du5+pc1
      return
    endif
c      istat=7                                           ! 55
      elseif(istat.eq.7)then
      p=AK*du+p
      if(du.gt.0)then
      if(p.lt.pt1) return                                ! 56
c      istat=7 から istat=2 へ                          ! 57
      istat=2
      du5=(p-pt1)/ak
      ak=ak_2
      p=ak*du5+pt1
      return
      else
      if(p.gt.pc2)return                                ! 58
c      istat=7 から istat=5 へ                          ! 59
      istat=5
      du5=(p-pc2)/ak
      AK=AK_3
      p=ak*du5+pc2
      return
      endif
c      初期設定                                         ! 60
      else
      istat=1
      ak=ak_1
      p=0
      pt1=Q_1
      pt2=Q_2
      pc1=-Q_1
      pc2=-Q_2
      du1=Q_1/ak_1                                     ! 61
      du2=(Q_2-Q_1)/ak_2+du1
      du3=2*du1
      du4=2*du2
      p_length1=du3*ak_1                               ! 62
      p_length2=(du4-du3)*ak_2
      sut=du2                                           ! 63
      suc=-du2
      dpp=0
      goto 100
      endif
    end

```

1. 状態が istat:1 の場合、以下の処理を行う。
2. 増分ひずみと剛性を用いて増分応力を計算し、その値を増分前の応

- 力に足しこむ。
3. 増分ひずみが引張の場合 ($du > 0$) 以下の処理を行う。
 4. 増分後の応力が引張側の境界値 $Pt1$ より小さい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
 5. 増分後の応力が引張側の境界値 $Pt1$ より大きい場合は、 $istat:2$ となる。
 6. 飛び越しの修正を行うために、応力が境界地 $pt1$ から飛び出た分のひずみを求め、 $du2$ にセットする。
 7. 状態が $istat:2$ の時の接線剛性である第2勾配の値とする。
 8. その接線剛性と先程の飛び出した部分のひずみ $du2$ とで、増分応力を計算し、増分後の応力をセットする。その後サブルーチンから戻る。
 9. 増分ひずみが圧縮の場合、以下の処理を行う。
 10. 増分後の応力が圧縮側の境界値 $Pc1$ より大きい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
 11. 増分後の応力が圧縮側の境界値 $Pc1$ より小さい場合は、状態が $istat:3$ となる。
 12. 状態が $isata:2$ の場合以下の処理を行う。
 13. 増分後の応力が引張側の境界値 $Pt2$ より大きい場合は、状態が $istat:4$ となる。
 14. 増分ひずみが負である場合、応力が反転するため、以下の処理を行う。
 15. 状態を $istat:1$ に戻す。
 16. 増分応力を元の剛性を利用して修正する。
 17. 接線剛性を第1勾配とする。
 18. 修正した増分応力を用いて増分ひずみを修正する。
 19. 増分後の応力から増分応力を引く事によって、増分前の応力を求める。
 20. 求められた増分前の応力を $Pt1$ にセットする。
 21. $Pt1$ から $2 \times Q_1$ と今までの塑性応力増分量の総和 dpp を引く事によって、状態が $istat:1$ における境界値 $Pc1$ を計算する。
 22. 状態が $isata3$ の場合以下の処理を行う。
 23. $Pt1$ に $2 \times Q_1$ と今までの塑性応力増分量の総和 dpp を足す事によって、状態が $istat:1$ における境界値 $Pt1$ を計算する。
 24. 状態が $isata4$ の場合以下の処理を行う。
 25. 増分ひずみが引張の時その境界はなく、サブルーチンから戻る。

26. 増分ひずみが反転し圧縮となった時、状態が $istat:6$ となる。
27. 現在のひずみが最大引張側ひずみより大きいとき ($ug>sut$) 以下の処理を行う。
28. まず、塑性応力増分量を求める。
29. 今までの塑性応力増分量に塑性応力増分量を足し、塑性応力増分量の総和 dpp を求める。
30. 現在のひずみを sut とする。
31. 現在の応力を $Pt2$ とする。
32. 現在のひずみが最大引張側ひずみより小さいとき ($ug<sut$) 以下の処理を行う。
33. 現在の応力を $Pt2$ とする。
34. 状態が $istat:6$ における境界値を求めるため、 $Pt2$ から $P_length1$ と dpp を引いた値を $Pc1$ にセットする。
35. 反転後ひずみが圧縮で進行し、状態が $istat:3$ となった時の境界値を求めるため、 $Pc1$ から $P_length2$ を引いた値を $Pc2$ にセットする。
36. 設定した接線剛性と飛び出した部分のひずみと増分応力を計算し、増分後の応力を求める。
37. 状態が $isata5$ の場合以下の処理を行う。
38. 増分ひずみが圧縮の時その境界はなく、サブルーチンから戻る。
39. 増分ひずみが反転し引張となった時、状態が $istat:7$ となる。
40. 現在のひずみが最大圧縮側ひずみより大きいとき ($ug<suc$) 以下の処理を行う。
41. まず、塑性応力増分量を求める。
42. 今までの塑性応力ひずみ増分量に塑性応力増分量を足し、塑性応力増分量の総和 dpp を求める。
43. 現在のひずみを、 suc とする。
44. 現在の応力を、 $Pc2$ とする。
45. 現在のひずみが最大圧縮側ひずみより小さいとき ($ug>suc$) 以下の処理を行う。
46. 現在の応力を $Pc1$ とする。
47. 状態が $istat:7$ における境界値を求めるため、 $Pc2$ に $P_length1$ と dpp を足した値を $Pt1$ にセットする。
48. 反転後ひずみが引張で進行し、状態が $istat:2$ となった時の境界値を求めるため、 $Pt1$ から $P_length2$ を足した値を $Pt2$ にセットする。
49. 設定した接線剛性と飛び出した部分のひずみ、増分応力を計算し、増分後の応力を求める。

50. 状態が `isata6` の場合以下の処理を行う。
51. 増分後の応力が引張側の境界値 $Pt2$ より小さい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
52. 増分後の応力が引張側の境界値 $Pt2$ より大きい場合は、状態が `istat:4` となり以下の処理を行う。
53. 増分ひずみが圧縮の時、増分後の応力が圧縮側の境界値 $Pc1$ より大きい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
54. 増分後の応力が圧縮側の境界値 $Pc1$ より小さい場合は、状態が `istat:3` となり以下の処理を行う。
55. 状態が `istat:7` の場合以下の処理を行う。
56. 増分応力が引張の時、増分後の応力が引張側の境界値 $Pt1$ より小さい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
57. 増分後の応力が引張側の境界値 $Pt1$ より大きい場合は、状態が `istat:2` となり以下の処理を行う。
58. 増分ひずみが圧縮の時、増分後の応力が圧縮側の境界値 $Pc2$ より大きい場合は、未だこの状態内であるとして、このサブルーチンより戻る。
59. 増分後の応力が圧縮側の境界値 $Pc2$ より小さい場合は、状態が `istat:5` となり以下の処理を行う。
60. ここでは、ファイバーに関する初期設定を行う。まず状態パラメーターを `istat:1` とする。次に接線剛性 AK に第1勾配の剛性をセットする。また、現在の応力 p をゼロにし、初期状態での境界値をセットする。
61. 除荷時の境界値の計算に用いる P_length を計算するために $du1 \cdot du2 \cdot du3 \cdot du4$ を計算する。
62. $du3$ と $du4$ を用いて $P_length1$ と $P_length2$ を計算する。
63. 最大引張側ひずみ sut 、最大圧縮側ひずみ suc の初期設定を第2折れ点時のひずみに設定する。

4.2.7 非対称バイリニア型

本節では、ファイバーに関する履歴ルールで、図 4-17 に示す剛性低減型の非対称バイリニアについて説明する。図中の記号は以下のようである。

引張側第一折れ点の応力	: Q_t1	圧縮側最大ひずみ	: $u1$
圧縮側第一折れ点の応力	: Q_c1	引張側最大ひずみ	: $u2$
引張側第一勾配	: AK_t1	圧縮側第一勾配	: AK_c1
引張側第二勾配	: AK_t2	圧縮側第二勾配	: AK_c2
引張側除荷勾配	: AK_t3	圧縮側除荷勾配	: AK_c3

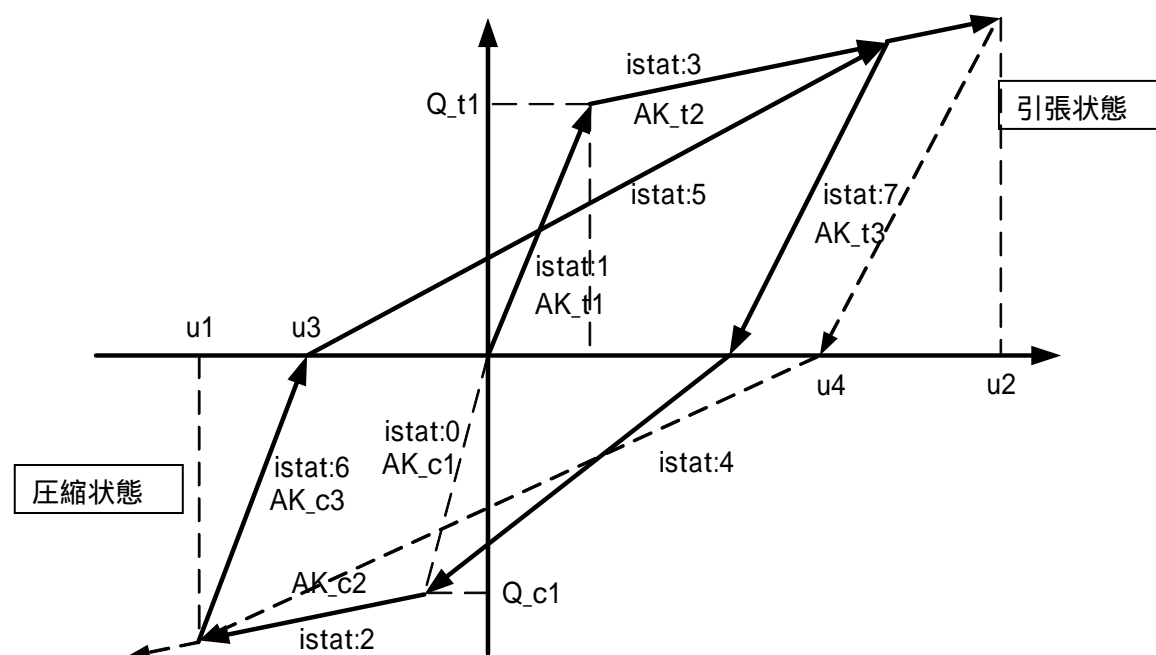


図 4-17 ファイバーモデルのバイリニア型履歴特性

この履歴特性は引張側と圧縮側で非対称のバイリニアとなる履歴に用いられる。ファイバーの弾塑性状態は、istat:0 と istat:1 では弾性状態を表している。また、istat:6 と istat:7 は除荷時の状態を表している。

また、istat:3 と istat:5 は引張降伏以降の状態を表しており、istat:2 と istat:4 は圧縮降伏以降の状態を表している。ここで、istat:4 では、ひずみが圧縮に進行する時、履歴はこの履歴特性の特徴である過去の圧縮側ひずみの最大点に向かって進む。一方、istat:5 では、ひずみが引張に進行する時、過去の引張側最大点に向かって進行する履歴を描く。

上記の履歴ルールを次の2つに分けて検討をする。

1. 第1折れ点を経験せず弾性体内で繰り返す履歴
2. 第1折れ点を経験して除荷する履歴

初めに、1)の第1折れ点を経験せず、弾性体内で繰り返す履歴について、図4-18を用いて説明する。

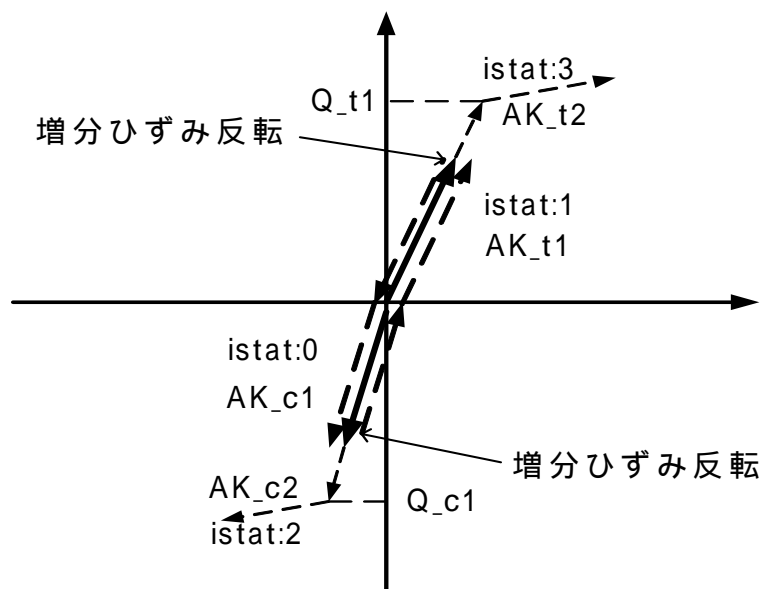


図4-18 第1折れ点を経験せず弾性体内で繰り返す履歴

図4-18に示すように、初めに得られる増分ひずみが引張ならば状態が1となり、逆に圧縮ならば状態が0となる。状態1の応力の境界値は $Q_{t1} > 0$ であり、 Q_{t1} を超えずに増分ひずみが反転すると、釣合点はそのまま状態1を直線的に変動する事になる。同様に、状態0の応力境界値は $0 > Q_{c1}$ であり、 Q_{c1} を超えない限り弾性状態として、釣合点は状態0を直線的に変化することになる。

また、状態0、1において、境界値0を越えると状態0は状態1に、状態1は状態0と各々変化し、それに伴って剛性も変わるが、この領域では、応力とひずみの関係は弾性体として振舞うことになる。

次に、2)の第1折れ点を経験して除荷する履歴について、図4-19を用いて説明する。ここでは、

2.1) 引張ひずみが進行し、第1折れ点を経験する場合

2.2) 圧縮ひずみが進行し、第1折れ点を経験する場合

に分けて述べることにする。

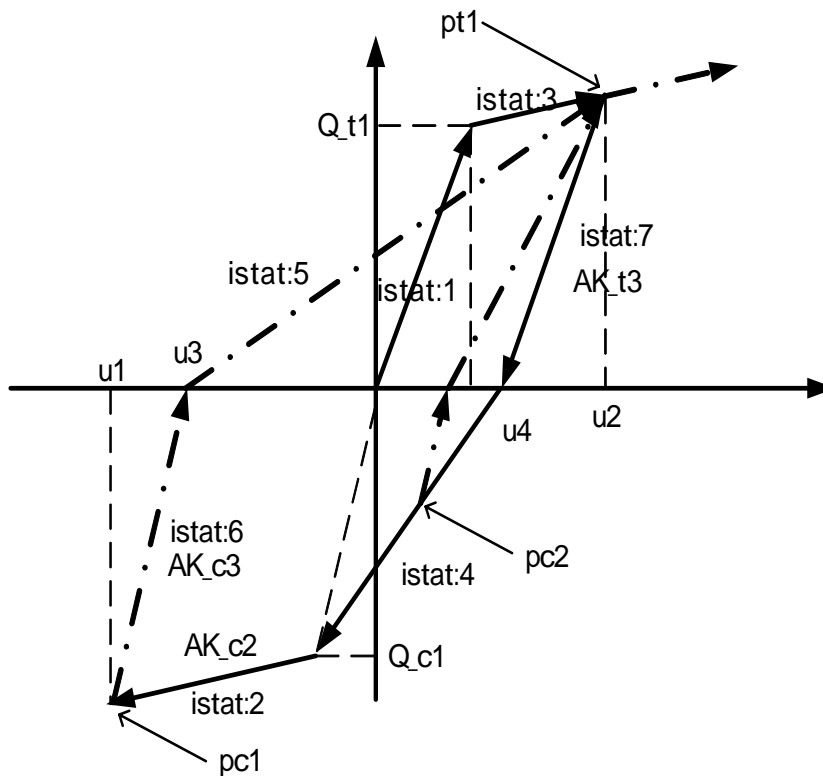


図 4-19 第 1 折れ点を経験して除荷する履歴

2.1) 引張側で第 1 折れ点を経験し、除荷する場合

図 4-19 に示すように、当初ひずみが状態 1 を引張方向に進行し、応力が Q_{t1} を超えると状態は 3 へと移る。そこで、状態 3 で増分ひずみが反転すると状態は 7 となる。ここでは、バイリニア型履歴モデルで説明した誤差修正タイプ 2 の方法で増分ひずみと増分応力を再計算し、増分後のひずみと応力を求める。その際、増分前のひずみと応力を最大引張ひずみ u_2 、最大引張応力 $pt1$ にセットする。

状態 7 における接線剛性は AK_{t3} であり、釣合点は、 $pt1 > 0$ の範囲内で直線的に移動することになる。この状態 7 において、ひずみが圧縮方向に進行して、応力が 0 を超えると状態は 4 に変更され、釣合点は座標 $(pc1, u1)$ を目指し進行する。このときの応力 0 のひずみを $u4$ とし、状態 4 における接線剛性を $u1, u4, pc1$ を用いて計算する。状態 4 では、増分ひずみが圧縮にある間、応力は $0 > pc1$ の間で直線的に変化するが、増分ひずみが引張に反転すると、状態は 0 となる。反転時の応力を $pc2$ とし、状態 0 での応力は $0 > pc2$ の範囲内で直線的に変化する。

2.2) 圧縮ひずみが進行し、第 1 折れ点を経験する場合

次に、圧縮ひずみが増加する場合について説明する。ただし、前項と

重複する内容については省略する。ひずみが状態0で圧縮方向に進行し、応力が Q_c1 を超えると、状態は2へと移る。状態2で、増分ひずみが反転して引張となると、状態は6となる。その際、タイプ2の方法を用いて誤差修正を行い、増分前のひずみと応力を求める。この時点のひずみが圧縮側の最大ひずみとなる時、この値を $u1$ にセットし、最大圧縮応力度として、この時点の応力を $pc1$ にセットする。状態6では、接線剛性は AK_c3 とする。ここでは、応力は $0 > > pc1$ の間で直線的に変化することとなる。この状態6で引張方向にひずみが進行し、応力が0を超える時、状態は5となる。応力が0の時のひずみを $u3$ にセットし、その後の釣合点は座標 $(pt1, u2)$ を目指し進行する。この時の接線剛性は $u2, u3, pt1$ を用いて計算されることになる。

状態5において、ひずみが反転する時、その時点の応力を $pt2$ にセットし、状態は7となる。この時の接線剛性は AK_t3 となり、応力は $pt2 > > 0$ の範囲内で直線的に変化することになる。

以上の説明をまとめると、剛性低減型の非対称バイリニア型履歴特性は、次の7つの状態で表現される。

- istat : 0 圧縮側の弾性状態であり、接線剛性は圧縮側の第1勾配の AK_c1 である。応力は $0 > > Q_c1$ の中で直線的に変化することになり、ここで、境界応力である Q_c1 を越えると状態2となり、0を越えると状態1となる。
- istat : 1 引張側の弾性状態であり、接線剛性は引張側の第1勾配 AK_t1 である。応力は $Q_t1 > > 0$ の中で直線的に変化することになり、ここで、境界応力 Q_t1 を越えると状態3となり、0を越えると状態0となる。
- istat : 2 骨格曲線である圧縮側の第2勾配上を示し、接線剛性は圧縮側の第2勾配の AK_c2 である。増分ひずみが負である圧縮となると、その上限の境界はなく、逆に、増分ひずみが反転して、正である引張となると、状態は6となる。この反転した時点での応力を $pc1$ にセットするが、この点が圧縮側の最大値となる。
- istat : 3 骨格曲線である引張側の第2勾配上を示し、接線剛性は引張側の第2勾配 AK_t2 である。増分ひずみが引張の時その境界はなく、増分ひずみが反転し圧縮となった時、状態は7となる。この反転した時点での応力を $pt1$ にセットするが、この点が引

張側の最大値となる。

istat : 4 圧縮側の最大点を指向する直線上の状態を示す。ここでは、増分ひずみが圧縮の時、その境界は $pc1 >$ であり、増分ひずみが反転して引張となった時、その応力を $pc2$ として状態は 0 となる。

istat : 5 引張側の最大点を指向する直線上の状態を示す。増分ひずみが引張の時、その境界は $>pt1$ であり、増分ひずみが反転し圧縮となった時、その応力を $pt2$ として状態は 1 となる。

istat : 6 状態 2 もしくは状態 4 において圧縮方向のひずみが、引張方向に反転した時の状態で、圧縮側の除荷時の状態を示す。接線剛性は圧縮側の除荷勾配 AK_c3 である。この状態に移る時の反転時の応力を状態 2 では $pc1$ とし、釣合点は、 $0 > >pc1$ の範囲内で直線的に動き、状態 4 では反転した時の応力を $pc2$ とし、同じく $0 > >pc2$ の範囲内で直線的に動く。

istat : 7 状態 3 もしくは状態 5 において引張方向のひずみが、圧縮方向に反転した時の状態で、引張側の除荷時の状態となる。接線剛性は引張側の除荷勾配の AK_t3 で反転時の応力を状態 3 では $pt1$ とし、釣合点は $pt1 > >0$ の範囲内で直線的に動き、状態 5 では反転した時の応力を $pt2$ とし、同じく $pt2 > >0$ の範囲内で直線的に動く。

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する。

```

C
C      SUBROUTINE /BiLiner_AS ( 最大点指向型非対称バイリニア )
C
C      BiLiner_Masing 履歴モデル
C
      subroutine BiLiner_AS(AK,istat,AK_t1,AK_t2,AK_t3,Q_t1,AK_c1,
*          AK_c2,AK_c3,Q_c1,du,P,Pt1,Pc1,st,ug,u2,u1,AK_t4,AK_c4)
      implicit real*8(A-H,O-Z)
C
C      AK              : 接線剛性
C      istat           : 現在の状態(Work)
C      AK_t1           : 引張り側第 1 勾配
C      AK_t2           : 引張り側第 2 勾配
C      AK_t3           : 引張り側除荷勾配
C      Q_t1            : 引張り側第 1 折れ点のせん断力
C      AK_c1           : 圧縮側第 1 勾配
C      AK_c2           : 圧縮側第 2 勾配

```

```

c   AK_c3           : 圧縮側除荷勾配
c   Q_c1            : 圧縮側第1折れ点のせん断力
c   du              : 増分変位
c   P               : 現在のせん断力(Work)
c   Pt1             : istat=3における反曲点上端(Work)
c   Pc1             : istat=2における反曲点上端(Work)
c   Pt2             : istat=5における反曲点上端(Work)
c   Pc1             : istat=4における反曲点上端(Work)
C
100  continue
C                                     istat0 (圧縮側)
      if (istat.eq.0) then                                     !1
        p=AK*du+p
        if (du.lt.0) then
          if (p.gt.-pc1) return                                !2
C                                     istat0 から istat=2 へ(圧縮)
          istat=2                                              !3
          du2=(p+pc1)/AK
          AK=AK_c2
          p=AK*du2-pc1
          return
        else
          if (p.lt.0) return                                    !4
C                                     istat0 から istat=1 へ(引張)
          istat=1                                              !5
          du2=p/AK
          u3=ug+du-du2                                         !6
          AK=pt1/(-u3+u2)                                       !7
          p=AK*du2
          return
        endif
C                                     istat1 (引張側)
      elseif (istat.eq. 1) then                                  !8
        p=AK*du+p
        if (du.gt.0) then
          if (p.lt.pt1) return                                  !9
C                                     istat1 から istat=3 へ(引張)
          istat=3                                              !10
          du2=(p-pt1)/AK
          AK=AK_t2
          p=AK*du2+pt1
          return
        else
          if (p.gt.0) return                                    !11
C                                     istat1 から istat=0 へ(圧縮)
          istat=0                                              !12
          du2=-p/AK
          u4=ug+du-du2                                         !13
          AK=pc1/(u1+u4)                                         !14
          p=-AK*du2
          return
        endif
C                                     istat2 (圧縮側)
      elseif (istat.eq.2) then                                  !15

```

```

      p=AK*du+p
      if (du.lt.0) return                                !16
C                                     istat2 から istat=6 へ(除荷)
      istat=6                                           !17
      dp=AK*du
      AK=AK_c3
      du=dp/AK
      p=p-dp
      pc1=-p                                           !18
      u1=-ug                                           !19
      p=AK*du+p
      return
C                                     istat=3 (引張側)
      elseif (istat.eq.3) then                          !20
      p=AK*du+p
      if (du.gt.0) return                                !21
C                                     istat3 から istat=7 へ(除荷)
      istat=7                                           !22
      dp=AK*du
      AK=AK_t3
      du=dp/AK
      p=p-dp
      pt1=p                                           !23
      u2=ug                                           !24
      p=AK*du+p
      return
C                                     istat4 (圧縮側)
      elseif (istat.eq.4) then                          !25
      p=AK*du+p
      if (du.lt.0) then
      if (p.gt.-pc1) return                                !26
C                                     istat4 から istat=2 へ(圧縮側)
      istat=2                                           !27
      du2=(p+pc1)/AK
      AK=AK_c2
      p=AK*du2-pc1
      return
      else
C                                     istat4 から istat=6 へ(除荷)
      istat=6                                           !28
      st=1                                           !29
      dp=AK*du
      AK_c4=AK                                           !30
      AK=AK_c3
      du=dp/AK
      p=p-dp
      pc2=-p                                           !31
      p=AK*du+p
      endif
      return
C                                     istat5 (引張側)
      elseif (istat.eq.5) then                          !32
      p=AK*du+p
      if (du.gt.0) then

```

	if(p.lt.pt1)return		!33
C		istat5(引張)	
	istat=3		!34
	du2=(p-pt1)/AK		
	AK=AK_t2		
	p=AK*du2+pt1		
	else		
C		istat5 から istat=7 へ(除荷)	
	istat=7		!35
	st=2		!36
	dp=AK*du		
	AK_t4=AK		!37
	AK=AK_t3		
	du=dp/AK		
	p=p-dp		
	pt2=p		!38
	p=AK*du+p		
	endif		
	return		
C		istat6 (圧縮側)	
	elseif (istat.eq.6)then		!39
	p=AK*du+p		
	if(du.lt.0)then		!40
	if(st.eq.0)then		
	if(p.gt.-pc1)return		!41
C		istat6 から istat=2 へ(圧縮)	
	istat=2		!42
	du2=(p+pc1)/AK		
	AK=AK_c2		
	p=AK*du2-pc1		
	return		
	else		
	if(p.gt.-pc2)return		!43
C		istat6 から istat=4 へ(圧縮)	
	istat=4		!44
	st=0		!45
	du2=(p+pc2)/AK		
	AK=AK_c4		
	p=AK*du2-pc2		
	return		
	endif		
	else		
	if(p.lt.0)return		!46
C		istat6 から istat=5 へ(引張)	
	istat=5		!47
	st=0		!48
	du2=p/AK		
	u3=ug+du-du2		!49
	AK=pt1/(-u3+u2)		!50
	p=AK*du2		
	return		
	endif		
C		istat7 (引張側)	
	elseif (istat.eq.7)then		!51

```

        p=AK*du+p
        if (du.gt.0) then
            if (st.eq.0) then
                if (p.lt.pt1) return
            C
            istat7 から istat=3 へ(引張)
            istat=3
            du2=(p-pt1)/AK
            AK=AK_t2
            p=AK*du2+pt1
            return
            else
            C
            istat7 から istat=5 へ(引張)
            istat=5
            st=0
            du2=(p-pt2)/AK
            AK=AK_t4
            p=AK*du2+pt2
            return
            endif
            else
            C
            istat7 から istat=4 へ(圧縮)
            istat=4
            st=0
            du2=-p/AK
            u4=ug+du-du2
            AK=pc1/(u1+u4)
            p=-AK*du2
            return
            endif
            C
            初期設定(引張)
            else
            if (du.gt.0) then
                istat=1
                AK=AK_t1
            C
            初期設定(圧縮)
            else
            istat=0
            AK=AK_c1
            endif
            p=0
            pt1=Q_t1
            pc1=Q_c1
            u1=Q_c1/AK_c1
            u2=Q_t1/AK_t1
            st=0
            goto 100
            end if
        end

```

このプログラム右のコメント番号にしたがって処理内容を説明しよう。

1. 状態が $istat : 0$ の場合、以下の処理を行う。
2. 増分ひずみが圧縮の場合 ($du < 0$) で、しかも増分後の応力が圧縮側の境界値 $-pc1$ より大きければ、このサブルーチンより戻る。
3. 増分後の応力が $-pc1$ より小さい場合は状態が 2 となり、以下の処理を行う。
4. 増分ひずみが引張の場合 ($du > 0$) 増分後の応力が 0 よりも小さければ、このサブルーチンより戻る。
5. 増分後の応力が 0 よりも大きい場合、状態を 5 とし、以下の処理を行う。
6. 応力が 0 となる時のひずみを計算し、 $u3$ にセットする。
7. ひずみ $u3$ と $u2$ を用いて、最大応力点の座標 ($u2, pt1$) を指向する接線剛性を計算し、その値を AK にセットする。
8. 状態が $istat : 1$ の場合、以下の処理を行う。
9. 増分ひずみが引張の場合 ($du > 0$) 増分後の応力が引張側の境界値 $pt1$ より小さければ、このサブルーチンより戻る。
10. 増分後の応力が $pt1$ より大きい場合は状態が 3 となり、以下の処理を行う。
11. 増分ひずみが圧縮の場合 ($du < 0$) 増分後の応力が 0 よりも大きければ、このサブルーチンより戻る。
12. 増分後の応力が 0 よりも小さい場合は状態が 4 となり、以下の処理を行う。
13. 応力が 0 となる時のひずみを計算し、 $u4$ に入力する。
14. ひずみ $u1$ と $u4$ を用いて、圧縮側最大応力点の座標 ($-u1, -pc1$) を指向する接線剛性を計算し、その値を AK にセットする。
15. 状態が $istat : 2$ の場合、以下の処理を行う。
16. 増分ひずみが圧縮の場合 ($du < 0$) このサブルーチンより戻る。
17. 増分ひずみが引張の場合 ($du > 0$) 状態は 6 となり、以下の処理を行う。
18. 増分前の応力を求め、 $pc1$ にセットする。
19. 前ステップまでのひずみを $u1$ にセットする。
20. 状態が $istat : 3$ の場合、以下の処理を行う。
21. 増分ひずみが引張の場合 ($du > 0$) このサブルーチンより戻る。
22. 増分ひずみが圧縮の場合 ($du < 0$) 状態は 7 となり、以下の処理を行う。
23. 増分前の応力を求め、 $pt1$ にセットする。
24. 前ステップまでのひずみを $u2$ にセットする。

25. 状態が $istat : 4$ の場合、以下の処理を行う。
26. 増分ひずみが圧縮の場合 ($du < 0$)、以下の処理を行う。増分後の応力が境界値 $-pc1$ より大きければ、このサブルーチンより戻る。
27. 増分後の応力が $-pc1$ より小さい時状態は 2 となり、以下の処理を行う。
28. 増分ひずみが引張の場合 ($du > 0$) 状態は 6 となり、以下の処理を行う。
29. 除荷状態を表すパラメーターである st を 1 にする。
30. 最大点を指向する現在の接線剛性を AK_c4 にセットする。
31. 境界条件となる $pc2$ に増分前の応力をセットする。
32. 状態が $istat : 5$ の場合、以下の処理を行う。
33. 増分ひずみが引張の場合 ($du > 0$)、以下の処理を行う。増分後の応力が $pt1$ より小さければ、このサブルーチンより戻る。
34. 増分後の応力が $pt1$ より大きい時、状態は 3 となり、以下の処理を行う。
35. 増分ひずみが圧縮の場合 ($du < 0$) 状態は 7 となり、以下の処理を行う。
36. 除荷状態を表すパラメーターである st を 2 にする。
37. 最大点を指向する現在の接線剛性を AK_t4 にセットする。
38. 境界条件となる $pt2$ に増分前の応力をセットする。
39. 状態が $istat : 6$ の場合、以下の処理を行う。
40. 増分ひずみが圧縮の場合 ($du < 0$)、以下の処理を行う。
41. 除荷状態を表すパラメーターの st が 0 の時、増分後の応力が圧縮側の境界値 $-pc1$ より大きければ、このサブルーチンより戻る。
42. 増分後の応力が $-pc1$ より小さい場合は状態が 2 となり、以下の処理を行う。
43. 除荷状態を表すパラメーターの st が 0 以外の場合、増分後の応力が圧縮側の境界値 $-pc2$ より大きければ、このサブルーチンより戻る。
44. 増分後の応力が $-pc2$ より小さい場合は状態が 4 となり、以下の処理を行う。
45. 除荷状態を表すパラメーターの st を 0 にする。
46. 増分ひずみが引張の場合 ($du > 0$) 増分後の応力が境界値 0 より小さければ、このサブルーチンより戻る。
47. 増分後の応力が境界値 0 より大きい場合は状態が 5 となり、以下の処理を行う。

48. 除荷状態を表すパラメーターの st を 0 にする。
49. 応力が 0 となる時のひずみを計算し、 $u3$ にセットする。
50. ひずみ $u3$ と $u2$ を用いて、引張側の最大応力点の座標 ($u2, pt1$) を指向する接線剛性を計算し、その値を AK にセットする。
51. 状態が $istat : 7$ の場合、以下の処理を行う。
52. 増分ひずみが引張の場合 ($du > 0$)、以下の処理を行う。
53. 除荷状態を表すパラメーターの st が 0 の時、増分後の応力が引張側の境界値 $pt1$ より小さければ、このサブルーチンより戻る。
54. 増分後の応力が $pt1$ より大きい場合は状態が 3 となり、以下の処理を行う。
55. 除荷状態を表すパラメーターの st が 0 以外の場合、増分後の応力が引張側の境界値 $pt2$ より小さければ、このサブルーチンより戻る。
56. 増分後の応力が $pt2$ より大きい場合、状態が 5 となり、以下の処理を行う。
57. 除荷状態を表すパラメーターの st を 0 にする。
58. 増分ひずみが圧縮の場合 ($du < 0$)、増分後の応力が境界値 0 より大きければ、このサブルーチンより戻る。
59. 増分後の応力が境界値 0 より小さい場合、状態が 4 となり、以下の処理を行う。
60. 除荷状態を表すパラメーターの st を 0 にする。
61. 応力が 0 となる時のひずみを計算し、 $u4$ にセットする。
62. ひずみ $u1$ と $u4$ を用いて、圧縮側の最大応力点の座標 ($-u1, -pc1$) を指向する接線剛性を計算し、その値を AK にセットする。
63. 状態が $istat : -1$ の場合、以下の処理を行う。
64. 得られた増分ひずみが引張方向 ($du > 0$) の場合、以下の処理を行う。
65. 得られた増分ひずみが圧縮方向 ($du < 0$) の場合、以下の処理を行う。
66. 初期設定として、引張と圧縮の最大点の座標として、両者の第 1 折れ点の座標をセットする。
67. 除荷状態を表すパラメーターである st に 0 をセットする。

4.2.8 非対称トリリニア型

本節では、ファイバーに関する履歴ルールで、図 4-20 に示す剛性低減型の非対称トリリニアについて説明する。図中の記号は以下のようである。

引張側第 1 折れ点の応力	: Q_t1	圧縮側最大ひずみ	: $u1$
引張側第 2 折れ点の応力	: Q_t2	圧縮側第 2 折れ点のひずみ	: $u5$
圧縮側第 1 折れ点の応力	: Q_c1	引張側最大ひずみ	: $u2$
圧縮側第 2 折れ点の応力	: Q_c2	引張側第 2 折れ点のひずみ	: $u6$
引張側第 1 勾配	: AK_t1	圧縮側第 1 勾配	: AK_c1
引張側第 2 勾配	: AK_t2	圧縮側第 2 勾配	: AK_c2
引張側第 3 勾配	: AK_t3	圧縮側第 3 勾配	: AK_c3
引張側除荷勾配	: AK_t4	圧縮側除荷勾配	: AK_c4
除荷時の状態	: st		

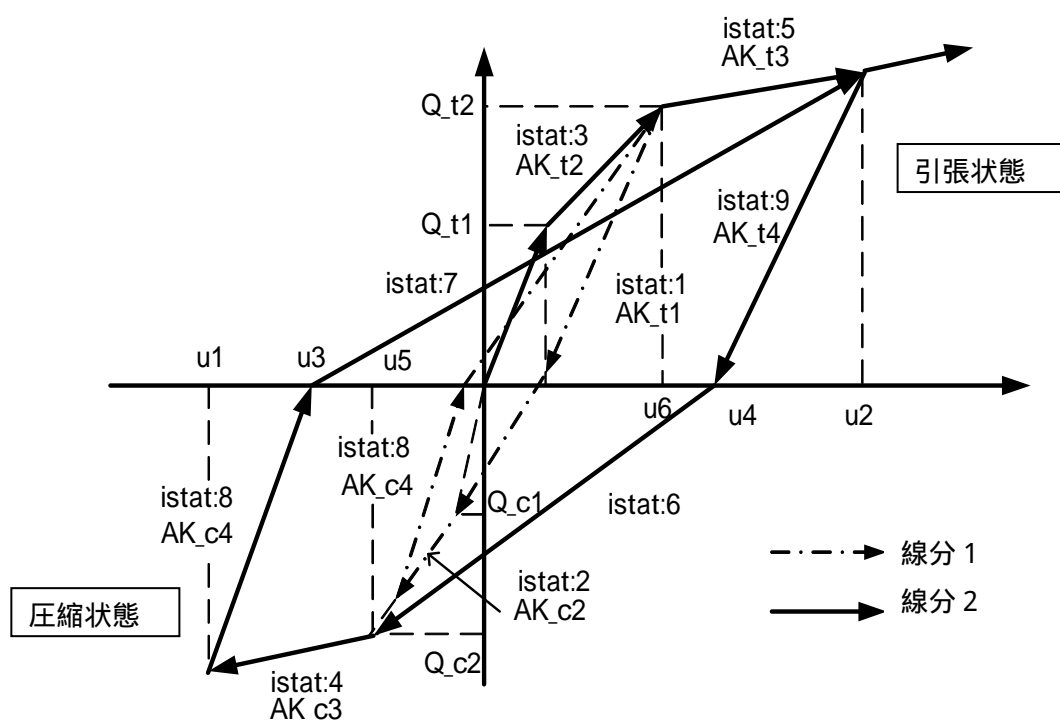


図 4-20 ファイバーモデルのトリリニア型履歴特性

非対称トリリニア型履歴は、引張側と圧縮側で非対称となる骨格曲線を有する。図 4-20 に示されるファイバーの弾塑性状態は、istat:0 と istat:1 が弾性状態を表し、istat:9 では引張側で除荷された時の状態を、同じく istat:8 では圧縮側で除荷された時の状態を示す。

降伏以後の状態として、istat:3、istat:5 と istat:7 では引張側の状

態を、同じく $istat:2$ 、 $istat:4$ と $istat:6$ では圧縮側の状態を示す。

プレゼンターの断面表示では、第2折れ点以降を塑性状態としている。

この履歴の特徴として、状態 $istat:6$ で圧縮にひずみが進行するとき、釣合点は、圧縮側の過去の最大応力を目指して進行し、一方、状態 $istat:7$ で引張にひずみが進行するとき、引張側の最大応力を目指して進行する。

以下に、これらの履歴についてさらに詳細に見ていこう。まず、第1折れ点を経過しない履歴、つまり、弾性状態を繰り返す履歴について説明する。

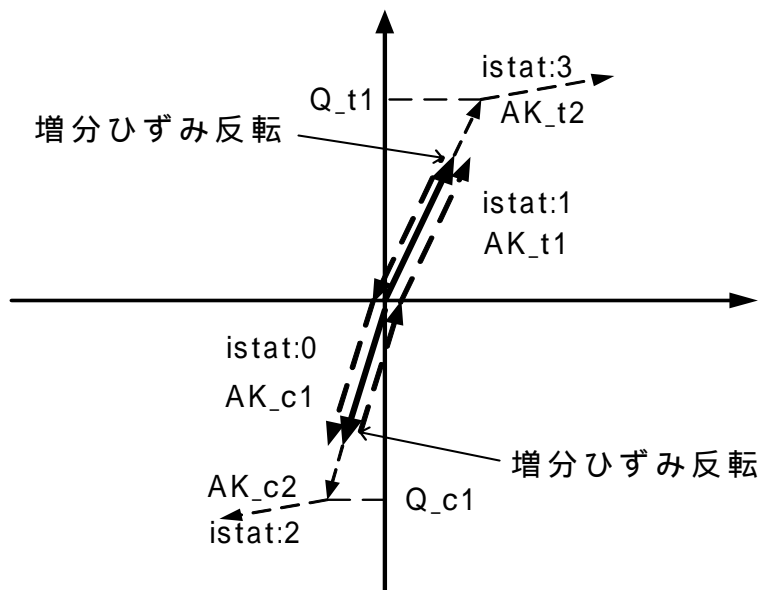


図 4-21 第1折れ点を経験せず弾性体内で繰り返す履歴

図 4-21 に示すように、初めに得られる増分ひずみが引張ならば、状態は 1 となり、圧縮ならば状態は 0 となる。状態 1 の境界値は $Q_{t1} > 0$ であり、応力が Q_{t1} を超えずに増分ひずみが反転すると、釣合点はそのまま状態 1 を直線的に逆行することになる。同様に、状態 0 の境界値は $0 > Q_{c1}$ であり、応力が Q_{c1} を超えない限り弾性状態として、釣合点は状態 0 を直線的に進行することになる。

弾性状態である 0、1 において、境界値 0 を越えると、状態 0 は状態 1 に、状態 1 は状態 0 となるが、引張と圧縮が逆転して弾性係数が変化するのはのみで、釣合点は弾性範囲内に位置する。

次に、第1折れ点や第2折れ点を経験して塑性域に入る履歴について図 4-20 に示す線分 1、2 に分けて説明する。

1) 線分1の第1折れ点を経験して除荷する履歴

図 4-22 を用いて第 1 折れ点を経験して除荷する履歴について説明する。初めに得られる増分ひずみが引張の場合、状態は 1 となり、境界値 Q_{t1} を越えると状態が 3 となる。この状態 3 において、ひずみが反転すると状態が 9 となり、後で必要となるため、この時点の増分前の応力を $pt1$ とする。状態 9 における応力境界は $pt1 > 0$ であり、釣合点はこの範囲内で直線的に移動する。

圧縮方向に増分ひずみが進行し、応力が 0 を越えると、状態は 6 となる。この状態では、釣合点は過去の最大圧縮応力点である座標 $(Q_{c1}, u1)$ を目指して進行する。当然、この状態では接線剛性は、計算して求めることになる。状態 6 では、圧縮方向にひずみが進むとき、応力境界値 $pc1$ まで直線的に変化するが、一端、増分ひずみが反転して引張側に進行すると状態は 8 に変わる。

次に、この状態 6 における 2 通りの履歴について説明する。

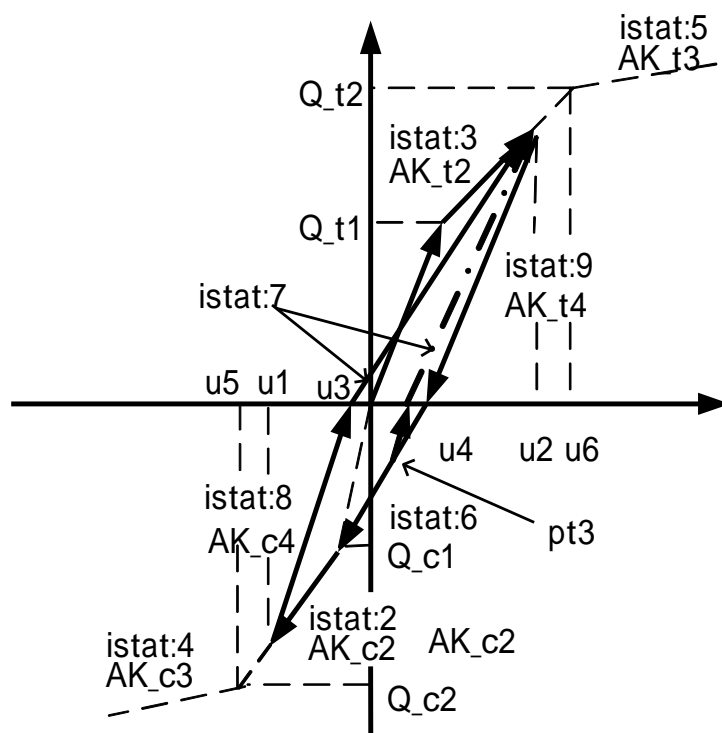


図 4-22 第 1 折れ点を経験して除荷する履歴

1-1) 状態 6 で圧縮方向に進行し、状態 2 となる場合

状態 6 でひずみが圧縮方向に進行し、第 1 折れ点である応力境界値 $pc1$ を越えると状態は 2 となる。状態 2 では圧縮方向にひずみが

そのまま進行する場合は、境界値 $pc2$ まで直線的に進むことになる。一方、この状態 2 で除荷され、増分ひずみが反転すると、状態は 8 となる。このとき反転する点である増分前の応力を $pc1$ にセットすると、状態 8 では応力が $0 > pc1$ の範囲内で直線的な履歴となる。

状態 8 で増分ひずみが引張方向に進行し、応力が 0 を超えると、状態は 7 となり、釣合点は、引張側の過去の最大応力点である座標 $(pt1, u2)$ を目指して進行する。状態 7 では、引張方向の境界値 $pt1$ を越えると状態 3 となるが、除荷され増分ひずみが反転して圧縮となる場合は状態が 9 となる。ここでは、状態 6 から状態 8 に進行する履歴と同様の方法で履歴を描くことになる。この履歴の特徴については、次の で説明する。

1-2) 状態 6 で除荷され、増分ひずみが引張方向に進行する場合

状態 6 において除荷され、増分ひずみが引張方向に反転するとき、増分前の応力を $pc3$ としてセットした後、状態を 8 とする。状態 8 では、応力が $0 > pc3$ の範囲で直線的に変化することになる。この状態で、増分ひずみが引張方向に進み、応力が 0 を超えると、状態は 7 となり、後は、 で示した履歴となる。

2) 第 2 折れ点を経験して除荷する履歴

図 4-23 を用いて第 2 折れ点を経験して除荷する部分の履歴について説明する。

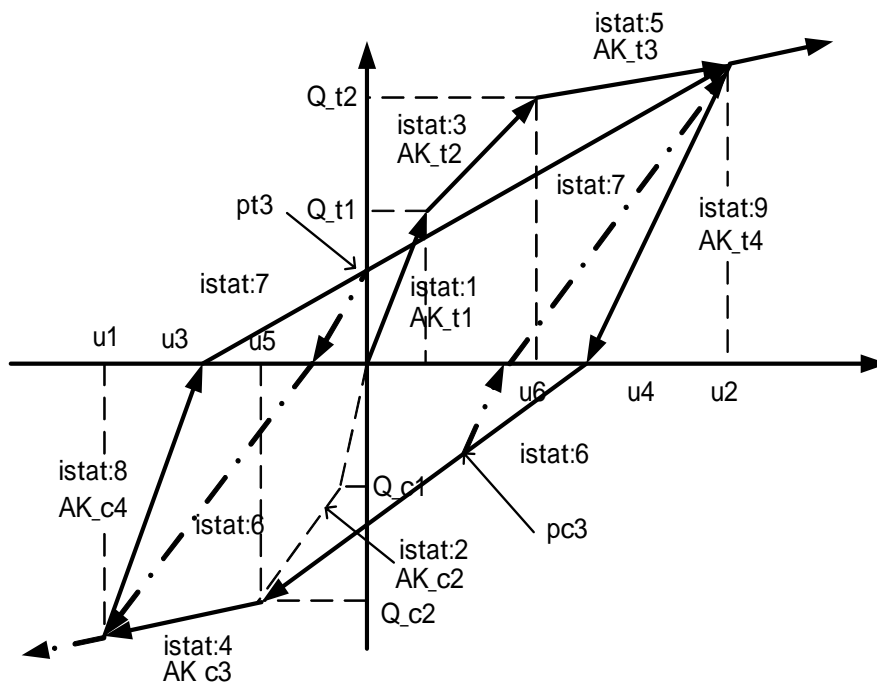


図 4-23 第 2 折れ点を経験して除荷する履歴

1)と同様に、ひずみが引張方向に進行し、応力が Q_{t2} を越えると状態は5となる。状態5において除荷が生じ、増分ひずみが反転して圧縮方向に進行すると、状態は9となる。その反転時の応力を $pt2$ にセットすると、状態9では、応力は、 $pt2 > 0$ の範囲内で直線的に履歴を描く。状態9において増分ひずみが圧縮方向に進行し、応力が0を越えると状態は6となり、圧縮側の第2折れ点の座標 ($u5, Q_{c2}$) を目指すことになる。当然、この状態の接線勾配は、計算によって求めることになる。

状態6において除荷されると、状態は8となるが、この履歴は1)で既に説明している。次に、ひずみが進行すると、状態7となるが、この状態7では、引張側のひずみ最大点は ($u2, pt2$) となることに注意されたい。また、状態6が圧縮方向に進行し、 Q_{c2} を越えると状態は4になる。状態4で除荷されて増分ひずみが引張方向に反転すると状態は8となり、増分前の応力を $pc2$ として $0 > pc2$ の範囲内で直線的に動くことになる。この先の履歴は1)と同様だが、圧縮側のひずみ最大点の座標が ($u1, pc2$) となることに注意が必要である。

以上の説明をまとめると、剛性低減型の非対称トリリニア型履歴特性は、次の9つの状態で表現することができる。

- | | |
|-----------|---|
| istat : 0 | 圧縮側の弾性状態であり、接線剛性は圧縮側の第1勾配 AK_{c1} である。応力は $0 > Q_{c1}$ の中で直線的に変化し、応力が Q_{c1} を越えると状態2となり、0を越えると状態1となる。 |
| istat : 1 | 引張側の弾性状態であり、接線剛性は引張側の第1勾配 AK_{t1} である。 $Q_{t1} > 0$ の中で直線的に動き、 Q_{t1} を越えると状態3となり、0を越えると状態0となる。 |
| istat : 2 | 圧縮側の第2勾配上の状態を示し、接線剛性は圧縮側の第2勾配 AK_{c2} である。増分ひずみが圧縮方向の場合、その応力の上限は Q_{c2} であり、 Q_{c2} を越えると状態は4となる。増分ひずみが反転し引張方向になると状態は8となり、その際の増分前の応力を $pc1$ にセットする。 |
| istat : 3 | 引張側の第2勾配上の状態を示し、接線剛性は引張側の第2勾配 AK_{t2} である。増分ひずみが引張方向の時、その応力境界は Q_{t2} であり、 Q_{t2} を越えると状態は5となる。増分ひずみが反転し、圧縮方向になると状態は9となる。その時の増分前の応力を $pt1$ にセットする。 |
| istat : 4 | 圧縮側の第3勾配上の状態を示し、接線剛性は圧縮側の第3 |

勾配 AK_c3 である。増分ひずみが圧縮方向の場合、その上限はなく、逆に増分ひずみが反転し引張方向になった場合、状態が 8 となる。その際、反転時における増分前の応力を $pc1$ にセットする。

istat : 5 引張側の第 3 勾配上の状態を示し、接線剛性は引張側の第 3 勾配 AK_t3 である。増分ひずみが引張方向の時、その境界はなく、逆に増分ひずみが反転して圧縮方向になった時、状態は 9 となる。その際、増分前の応力を $pt1$ とする。

istat : 6 圧縮側の最大点を指向する勾配であり、増分ひずみが圧縮の時その境界は $pc1 <$ 、もしくは $pc2 <$ であり、増分ひずみが反転し引張となった時、その応力を $pc3$ として状態は 8 となる。

istat : 7 引張側の最大点を指向する勾配であり、増分ひずみが引張の時その境界は $>pt1$ 、もしくは $>pt2$ であり、増分ひずみが反転し圧縮となった時、その応力を $pt3$ として状態は 9 となる。

istat : 8 圧縮側の除荷時の勾配であり、接線剛性は圧縮側の除荷勾配の AK_c4 である。履歴を描く条件によって $0 > >pc1$ 、 $0 > >pc2$ 、 $0 > >pc3$ の範囲内で直線的に履歴を描き、圧縮側の境界値が $pc1$ の時、その値を越えると状態は 2 に進み、圧縮側の境界値が $pc2$ の場合は、その境界値を越えると状態は 4 となる。また、圧縮側の境界値が $pc3$ の場合は、境界値を越えると状態は 6 となる。増分ひずみが引張方向に進行する時、境界値 0 を越えると状態は 7 となる。

istat : 9 引張側の除荷時の勾配であり、接線剛性は引張側の除荷勾配の AK_t4 である。履歴を描く条件によって $pt1 > >0$ 、 $pt2 > >0$ 、 $pt3 > >0$ の範囲内で直線に履歴を描き、引張側の境界値が $pt1$ の時、その値を越えると状態は 3 に進み、引張側の境界値が $pt2$ の場合は、その境界値を越えると状態は 5 となる。また、圧縮側の境界値が $pt3$ の場合は、境界値を越えると状態は 7 となる。増分ひずみが圧縮方向に進行する時、境界値 0 を越えると状態は 6 となる。

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 /TriLinear_AS(非対称トリリニア型)
C
subroutine TriLinear_AS(ak, istat, ak_t1, ak_t2, ak_t3, ak_t4, q_t1,
*                      q_t2, ak_c1, ak_c2, ak_c3, ak_c4, q_c1, q_c2, du, p, pt1, pt2,
*                      pc1, pc2, u5, u6, st, ug, u2, u1, ak_t5, ak_c5)
implicit real*8(A-H, O-Z)
dimension p1(5)
c
c      AK              : 接線剛性
c      istat           : 現在の状態(Work)
c      AK_t1           : 引張り側第1勾配
c      AK_t2           : 引張り側第2勾配
c      AK_t3           : 引張り側第3勾配
c      AK_t4           : 引張り側除荷勾配
c      Q_t1            : 引張り側第1折れ点のせん断力
c      Q_t2            : 引張り側第2折れ点のせん断力
c      AK_c1           : 圧縮側第1勾配
c      AK_c2           : 圧縮側第2勾配
c      AK_c3           : 圧縮側第3勾配
c      AK_c4           : 圧縮側除荷勾配
c      Q_c1            : 圧縮側第1折れ点のせん断力
c      Q_c2            : 圧縮側第2折れ点のせん断力
c      du              : 増分変位
c      P               : 現在のせん断力(Work)
c      Pt1             : istat=3における反曲点上端(Work)
c      Pt2             : istat=3における反曲点上端(Work)
c      Pc1             : istat=2における反曲点上端(Work)
c      Pc2             : istat=2における反曲点上端(Work)
C
100  continue
C
C                      istat0
      if (istat.eq.0) then
p=AK*du+p
      if (du.lt.0) then
      if (p.gt.-pc1) return
C                      istat0 から istat=2 へ(圧縮)
      istat=2
      du2=(p+pc1)/AK
      AK=AK_c2
      p=AK*du2-pc1
      return
      else
      if (p.lt.0) return
C                      istat0 から istat=1 へ(引張)
      istat=1
      st=0
      du2=p/AK
      u3=ug+du-du2
      if (pc2.eq.Q_c2) then
      if (pt2.eq.Q_t2) then
      AK=pt1/(-u3+u2)
      else
      AK=pt2/(-u3+u2)

```

```

endif
else
  if(pt2.eq.Q_t2)then
    u2=u6 !9
  endif
  AK=pt2/(-u3+u2) !10
endif
p=AK*du2
return
endif

C                                     istat1
elseif (istat.eq. 1)then !11
  p=AK*du+p
  if(du.gt.0)then
    if(p.lt.pt1) return !12
C                                     istat1 から istat=3 へ(引張)
    istat=3 !13
    du2=(p-pt1)/AK
    AK=AK_t2
    p=AK*du2+pt1
    return
  else
    if(p.gt.0)return !14
C                                     istat1 から istat=0 へ(圧縮)
    istat=0 !15
    st=0
    du2=-p/AK
    u4=ug+du-du2 !16
    if(pt2.eq.Q_t2)then
      if(pc2.eq.Q_c2)then
        AK=pc1/(u4+u1) !17
      else
        AK=pc2/(u4+u1) !18
      endif
    else
      if(pc2.eq.Q_c2)then
        u1=u5 !19
      endif
      AK=pc2/(u4+u1) !20
    endif
    p=-AK*du2
    return
  endif

C                                     istat2
elseif(istat.eq.2)then !21
  p=AK*du+p
  if(du.lt.0)then
    if(p.gt.-pc2)return !22
C                                     istat2 から istat=4
    istat=4 !23
    du2=(p+pc2)/AK
    AK=AK_c3
    p=AK*du2-pc2

```

```

    pc1=Q_c2
    return
  else
C          istat2 から istat=8 へ(除荷)          !24
    istat=8
    dp=AK*du
    AK=AK_c4
    du=dp/AK
    p=p-dp
    u1=-ug          !25
    pc1=-p
    p=AK*du+p
    endif
    return
C          istat3          !26
    elseif(istat.eq.3) then
    p=AK*du+p
    if(du.gt.0) then
    if(p.lt.pt2) return          !27
C          istat3 から istat=5 へ(引張)          !28
    istat=5
    du2=(p-pt2)/AK
    AK=AK_t3
    p=AK*du2+pt2
    pt1=Q_t2
    return
    else
C          istat3 から istat=9 へ(除荷)          !29
    istat=9
    dp=AK*du
    AK=AK_t4
    du=dp/AK
    p=p-dp
    u2=ug          !30
    pt1=p
    p=AK*du+p
    endif
    return
C          istat4          !31
    elseif(istat.eq.4) then
    p=AK*du+p
    if(du.lt.0) return
C          istat4 から istat=8 へ(除荷)          !32
    istat=8
    dp=AK*du
    AK=AK_c4
    du=dp/AK
    p=p-dp
    u1=-ug          !33
    pc2=-p
    p=AK*du+p
    return
C          istat5          !34
    elseif(istat.eq.5) then

```

	p=AK*du+p		
	if (du.gt.0) return		!35
C		istat5 から istat=9 へ(除荷)	
	istat=9		!36
	dp=AK*du		
	AK=AK_t4		
	du=dp/AK		
	p=p-dp		
	u2=ug		!37
	pt2=p		
	p=AK*du+p		
	return		
C		istat6(圧縮)	
	elseif (istat.eq.6) then		!38
	p=AK*du+p		
	if (du.lt.0) then		
	if (pc2.eq.Q_c2) then		
	if (pt2.eq.Q_t2) then		
	if (p.gt.-pc1) return		!39
C		istat6 から istat=2 へ(圧縮)	
	istat=2		!40
	du2=(p+pc1)/AK		
	AK=AK_c2		
	p=AK*du2-pc1		
	return		
	else		
	if (p.gt.-pc2) return		!41
C		istat6 から istat=4 へ(圧縮)	
	istat=4		!42
	du2=(p+pc2)/AK		
	AK=AK_c3		
	p=AK*du2-pc2		
	return		
	endif		
	else		
	if (p.gt.-pc2) return		!43
C		istat6 から istat=4 へ(圧縮)	
	istat=4		!44
	du2=(p+pc2)/AK		
	AK=AK_c3		
	p=AK*du2-pc2		
	return		
	endif		
	else		
C		istat6 から istat=8 へ(除荷)	
	istat=8		!45
	st=1		!46
	dp=AK*du		
	AK_c5=AK		!47
	AK=AK_c4		
	du=dp/AK		
	p=p-dp		
	pc3=-p		!48
	p=AK*du+p		

	return		
	endif		
C		istat7(引張)	
	elseif(istat.eq.7)then		!49
	p=AK*du+p		
	if(du.gt.0)then		
	if(pt2.eq.Q_t2)then		
	if(pc2.eq.Q_c2)then		
	if(p.lt.pt1)return		!50
C		istat7 から istat3(引張)	
	istat=3		!51
	du2=(p-pt1)/AK		
	AK=AK_t2		
	p=AK*du2+pt1		
	return		
	else		
	if(p.lt.pt2)return		!52
C		istat7 から istat5(引張)	
	istat=5		!53
	du2=(p-pt2)/AK		
	AK=AK_t3		
	p=AK*du2+pt2		
	return		
	endif		
	else		
	if(p.lt.pt2)return		!54
C		istat7 から istat5(引張)	
	istat=5		!55
	du2=(p-pt2)/AK		
	AK=AK_t3		
	p=AK*du2+pt2		
	return		
	endif		
	else		
C		istat7 から istat=9 へ(除荷)	
	istat=9		!56
	st=1		!57
	dp=AK*du		
	AK_t5=AK		!58
	AK=AK_t4		
	du=dp/AK		
	p=p-dp		
	pt3=p		!59
	p=AK*du+p		
	endif		
	return		
C		istat8	
	elseif (istat.eq.8)then		!60
	p=AK*du+p		
	if(du.lt.0)then		
	if(st.eq.0)then		
	if(p.gt.-pc1)return		!61
C		istat8 から istat=2 へ(圧縮)	
	istat=2		!62

```

    du2=(p+pc1)/AK
    AK=AK_c2
    p=AK*du2-pc1
    return
    else
    if(p.gt.-pc3) return                                !63
C      istat8から istat=6へ(圧縮)
    istat=6                                             !64
    st=0                                              !65
    du2=(p+pc3)/AK
    AK=AK_c5
    p=AK*du2-pc3
    return
    endif
    else
    if(p.lt.0) return                                !66
C      istat8から istat=7へ(引張)
    istat=7                                             !67
    st=0                                              !68
    du2=p/AK
    u3=ug+du-du2                                     !69
    if(pc2.eq.Q_c2) then
    if(pt2.eq.Q_t2) then
    AK=pt1/(-u3+u2)                                    !70
    else
    AK=pt2/(-u3+u2)                                    !71
    endif
    else
    if(pt2.eq.Q_t2) then
    u2=u6                                              !72
    endif
    AK=pt2/(-u3+u2)                                    !73
    endif
    p=AK*du2
    return
    endif
C      istat9
    elseif (istat.eq.9) then                            !74
    p=AK*du+p
    if(du.gt.0) then
    if(st.eq.0) then
    if(p.lt.pt1) return                                !75
C      istat9から istat=3へ(引張)
    istat=3                                             !76
    du2=(p-pt1)/AK
    AK=AK_t2
    p=AK*du2+pt1
    return
    else
    if(p.lt.pt3) return                                !77
C      istat9から istat=7へ(引張)
    istat=7                                             !78
    du2=(p-pt3)/AK
    AK=AK_t5

```

	p=AK*du2+pt3		
	return		
	endif		
	else		
C	if(p.gt.0)return	istat9 から istat=6 へ(引張)	!79
	istat=6		!80
	st=0		!81
	du2=-p/AK		
	u4=ug+du-du2		!82
	if(pt2.eq.Q_t2)then		
	if(pc2.eq.Q_c2)then		
	AK=pc1/(u4+u1)		!83
	else		
	AK=pc2/(u4+u1)		!84
	endif		
	else		
	if(pc2.eq.Q_c2)then		
	u1=u5		!85
	endif		
	AK=pc2/(u4+u1)		!86
	endif		
	p=-AK*du2		
	return		
	endif		
C		初期設定(引張)	
	else		!87
	if(du.gt.0)then		!88
	istat=1		
	AK=AK_t1		
C		初期設定(圧縮)	
	else		!89
	istat=0		
	AK=AK_c1		
	endif		
	p=0		
	pt1=Q_t1		
	pt2=Q_t2		
	pc1=Q_c1		
	pc2=Q_c2		
	u1=Q_c1/AK_c1		!90
	u2=Q_t1/AK_t1		
	u5=(Q_c2-Q_c1)/AK_c2+u1		
	u6=(Q_t2-Q_t1)/AK_t2+u2		
	st=0		!91
	goto 100		
	end if		
	end		

1. 状態が istat : 0 の場合、以下の処理を行う。
2. 増分ひずみが圧縮の場合 (du<0) で、増分後の応力が圧縮側の境界

- 値 - $pc1$ より大きければ、このサブルーチンより戻る。
3. 増分後の応力が - $pc1$ より小さい場合は状態が 2 となり、以下の処理を行う。
 4. 増分ひずみが引張の場合で ($du > 0$) 増分後の応力が 0 よりも小さければ、このサブルーチンより戻る。
 5. この時、増分後の応力が 0 よりも大きい場合、状態は 7 となり、以下の処理を行う。
 6. 応力が 0 となる時のひずみを計算し、 $u3$ にセットする。
 7. $pc2$ が Q_c2 に等しく、 $pt2$ が Q_t2 に等しければ、 $u3$ と $u2$ を用いて座標 ($u2, pt1$) に指向する接線剛性を計算し AK にセットする。
 8. $pc2$ が Q_c2 に等しく、 $pt2$ が Q_t2 に等しくなければ、 $u3$ と $u2$ を用いて座標 ($u2, pt2$) に指向する接線剛性を計算し AK にセットする。
 9. $pc2$ が Q_c2 に等しくなく、 $pt2$ が Q_t2 に等しければ、 $u2$ に引張側の第 2 折れ点のひずみ $u6$ をセットする。
 10. $pc2$ が Q_c2 に等しくなく、 $pt2$ が Q_t2 に等しくなければ、 $u3$ と $u2$ を用いて座標 ($u2, pt2$) に指向する接線剛性を計算し AK にセットする。
 11. 状態が $istat : 1$ の場合、以下の処理を行う。
 12. 増分ひずみが引張の場合 ($du > 0$) 増分後の応力が引張側の境界値 $pt1$ より小さければ、このサブルーチンより戻る。
 13. 増分後の応力が $pt1$ より大きい場合は状態が 3 となり、以下の処理を行う。
 14. 増分ひずみが圧縮の場合 ($du < 0$) 増分後の応力が 0 よりも大きければ、このサブルーチンより戻る。
 15. 増分後の応力が 0 よりも小さい場合は状態が 6 となり、以下の処理を行う。
 16. 応力が 0 となる時のひずみを計算し、 $u4$ にセットする。
 17. $pt2$ が Q_t2 と等しく、 $pc2$ が Q_c2 と等しければ、 $u1$ と $u4$ を用いて座標 ($-u1, -pc1$) に指向する接線剛性を計算し AK にセットする。
 18. $pt2$ が Q_t2 と等しく、 $pc2$ が Q_c2 と等しくなければ、 $u1$ と $u4$ を用いて座標 ($-u1, -pc2$) に指向する接線剛性を計算し AK にセットする。
 19. $pt2$ が Q_t2 と等しくなく、 $pc2$ が Q_c2 と等しければ、 $u1$ に圧縮側の第 2 折れ点のひずみ $u5$ をセットする。
 20. $pt2$ が Q_t2 と等しくなく、 $pc2$ が Q_c2 と等しくなければ、 $u1$ と $u4$ を用いて座標 ($-u1, -pc2$) に指向する接線剛性を計算し AK にセ

ットする。

21. 状態が $istat : 2$ の場合、以下の処理を行う。
22. 増分ひずみが圧縮の場合 ($du < 0$) で、増分後の応力が圧縮側の境界値 $-pc2$ より大きければ、このサブルーチンより戻る。
23. 増分後の応力が $-pc2$ より小さい場合は状態が 4 となり、以下の処理を行う。
24. 増分ひずみが引張の場合 ($du > 0$) 状態は 8 となり、以下の処理を行う。
25. 前ステップまでの全ひずみを最大圧縮ひずみ $u1$ にセットする。
26. 状態が $istat : 3$ の場合、以下の処理を行う。
27. 増分ひずみが引張の場合 ($du > 0$) 増分後の応力が引張側の境界値 $pt2$ より小さければ、このサブルーチンより戻る。
28. 増分後の応力が $pt2$ より大きい場合は状態が 5 となり、以下の処理を行う。
29. 増分ひずみが圧縮の場合 ($du < 0$) 状態は 9 となり、以下の処理を行う。
30. 前ステップまでの全ひずみを最大引張ひずみ $u2$ にセットする。
31. 状態が $istat : 4$ の場合、以下の処理を行う。
32. 増分ひずみが圧縮の場合 ($du < 0$) このサブルーチンより戻るが、増分ひずみが引張の場合 ($du > 0$) 状態が 8 となり以下の処理を行う。
33. 前ステップまでの全ひずみを最大圧縮ひずみ $u1$ に入力する。
34. 状態が $istat : 5$ の場合、以下の処理を行う。
35. 増分ひずみが引張の場合 ($du > 0$) このサブルーチンより戻る。
36. 増分ひずみが圧縮の場合 ($du < 0$) 状態が 9 となり以下の処理を行う。
37. 前ステップまでの全ひずみを最大引張ひずみ $u2$ にセットする。
38. 状態が $istat : 6$ の場合、以下の処理を行う。
39. 増分ひずみが圧縮の場合 ($du < 0$) で、 $pc2$ が Q_c2 と等しく、 $pt2$ が Q_t2 に等しく、増分後の応力が圧縮側の境界値 $-pc1$ より大きければ、このサブルーチンより戻る。
40. 増分ひずみが圧縮の場合 ($du < 0$) で、 $pc2$ が Q_c2 と等しく、 $pt2$ が Q_t2 に等しく、増分後の応力が圧縮側の境界値 $-pc1$ より小さければ、状態は 2 となり以下の処理を行う。
41. 増分ひずみが圧縮の場合 ($du < 0$) で、 $pc2$ が Q_c2 と等しく、 $pt2$ が Q_t2 に等しくなく、増分後の応力が圧縮側の境界値 $-pc2$ より大き

- ければ、このサブルーチンより戻る。
42. 増分ひずみが圧縮の場合 ($du < 0$) で、 $pc2$ が Q_c2 と等しく、 $pt2$ が Q_t2 に等しくなく、増分後の応力が圧縮側の境界値 - $pc2$ より小さければ、状態は 4 となり以下の処理を行う。
 43. 増分ひずみが圧縮の場合 ($du < 0$) で、 $pc2$ が Q_c2 と等しくなく、増分後の応力が圧縮側の境界値 - $pc2$ より大きければ、このサブルーチンより戻る。
 44. 増分ひずみが圧縮の場合 ($du < 0$) で、 $pc2$ が Q_c2 と等しくなく、増分後の応力が圧縮側の境界値 - $pc2$ より小さければ、状態は 4 となり以下の処理を行う。
 45. 増分ひずみが引張の場合 ($du > 0$)、状態は 8 となり以下の処理を行う。
 46. 除荷時の状態を表すパラメーター st を 1 にする。
 47. 現在の接線剛性を AK_c5 にセットする。
 48. 増分前の応力を求め、 $pc3$ にセットする。
 49. 状態が $istat : 7$ の場合、以下の処理を行う。
 50. 増分ひずみが引張の場合 ($du > 0$) で、 $pt2$ が Q_t2 に等しく、 $pc2$ が Q_c2 に等しく、増分後の応力が引張側の境界値 $pt1$ より小さければ、このサブルーチンより戻る。
 51. 増分ひずみが引張の場合 ($du > 0$) で、 $pt2$ が Q_t2 に等しく、 $pc2$ が Q_c2 に等しく、増分後の応力が引張側の境界値 $pt1$ より大きければ、状態が 3 となり以下の処理を行う。
 52. 増分ひずみが引張の場合 ($du > 0$) で、 $pt2$ が Q_t2 に等しく、 $pc2$ が Q_c2 に等しくなく、増分後の応力が引張側の境界値 $pt2$ より小さければ、このサブルーチンより戻る。
 53. 増分ひずみが引張の場合 ($du > 0$) で、 $pt2$ が Q_t2 に等しく、 $pc2$ が Q_c2 に等しくなく、増分後の応力が引張側の境界値 $pt2$ より大きければ、状態は 5 となり以下の処理を行う。
 54. 増分ひずみが引張の場合 ($du > 0$) で、 $pt2$ が Q_t2 に等しくなく、増分後の応力が引張側の境界値 $pt2$ より小さければ、このサブルーチンより戻る。
 55. 増分ひずみが引張の場合 ($du > 0$) で、 $pt2$ が Q_t2 に等しくなく、増分後の応力が引張側の境界値 $pt2$ より大きければ、状態は 5 となり以下の処理を行う。
 56. 増分ひずみが圧縮の場合 ($du < 0$)、状態は 9 となり以下の処理を行う。

57. 除荷時の状態を表すパラメーター st を 1 にする。
58. 現在の接線剛性を AK_t5 にセットする。
59. 増分前の応力を求め、 $pt3$ にセットする。
60. 状態が $istat : 8$ の場合、以下の処理を行う。
61. 増分ひずみが圧縮の場合 ($du < 0$) で、除荷状態を表す st が 0 であり、増分後の応力が圧縮側の境界値 - $pc1$ より大きければ、このサブルーチンより戻る。
62. 増分ひずみが圧縮の場合 ($du < 0$) で、除荷状態を表す st が 0 であり、増分後の応力が圧縮側の境界値 - $pc1$ より小さければ、状態は 2 となり以下の処理を行う。
63. 増分ひずみが圧縮の場合 ($du < 0$) で、除荷状態を表す st が 0 でなく、増分後の応力が圧縮側の境界値 - $pc3$ より大きければ、このサブルーチンより戻る。
64. 増分ひずみが圧縮の場合 ($du < 0$) で、除荷状態を表す st が 0 でなく、増分後の応力が圧縮側の境界値 - $pc3$ より小さければ、状態は 6 となり以下の処理を行う。
65. 除荷状態を表すパラメーターである st を 0 にする。
66. 増分ひずみが引張の場合 ($du > 0$)、増分後の応力が 0 より小さければ、このサブルーチンより戻る。
67. 増分ひずみが引張の場合 ($du > 0$)、増分後の応力が 0 より大きければ、状態は 7 となり以下の処理を行う。
68. 除荷状態を表す st を 0 にする。
69. 応力が 0 となるひずみを計算し、 $u3$ にセットする。
70. $pc2$ が Q_c2 に等しく、 $pt2$ が Q_t2 に等しい場合、 $u2$ と $u3$ を用いて、座標 ($u2, pt1$) に指向する接線剛性を計算し AK にセットする。
71. $pc2$ が Q_c2 に等しく、 $pt2$ が Q_t2 に等しくない場合、 $u2$ と $u3$ を用いて、座標 ($u2, pt2$) に指向する接線剛性を計算し AK に入力する。
72. $pc2$ が Q_c2 に等しくなく、 $pt2$ が Q_t2 に等しい場合、引張側最大ひずみ $u2$ に第 2 折れ点のひずみ $u6$ をセットする。
73. $pc2$ が Q_c2 に等しくなく、 $pt2$ が Q_t2 に等しくない時、 $u2$ と $u3$ を用いて、座標 ($u2, pt2$) に指向する接線剛性を計算し AK にセットする。
74. 状態が $istat : 9$ の場合、以下の処理を行う。
75. 増分ひずみが引張の場合 ($du > 0$) で、除荷状態を表す st が 0 であり、増分後の応力が引張側の境界値 $pt1$ より小さければ、このサブルーチンより戻る。

76. 増分ひずみが引張の場合 ($du > 0$) で、除荷状態を表す st が 0 であり、増分後の応力が引張側の境界値 $pt1$ より大きければ、状態は 3 となり以下の処理を行う。
77. 増分ひずみが引張の場合 ($du > 0$) で、除荷状態を表す st が 0 でなく、増分後の応力が引張側の境界値 $pt3$ より小さければ、このサブルーチンより戻る。
78. 増分ひずみが引張の場合 ($du > 0$) で、除荷状態を表す st が 0 でなく、増分後の応力が引張側の境界値 $pt3$ より大きければ、状態は 7 となり以下の処理を行う。
79. 増分ひずみが圧縮の場合 ($du < 0$)、増分後の応力が 0 より大きければ、このサブルーチンより戻る。
80. 増分ひずみが圧縮の場合 ($du < 0$)、増分後の応力が 0 より小さければ、状態は 6 となり以下の処理を行う。
81. 除荷状態を表すパラメーターである st を 0 にする。
82. 応力が 0 となるひずみを計算し、 $u4$ にセットする。
83. $pt2$ が Q_t2 に等しく、 $pc2$ が Q_c2 に等しい場合、 $u1$ と $u4$ を用いて座標 ($-u1, -pc1$) に指向する接線剛性を計算し AK にセットする。
84. $pt2$ が Q_t2 に等しく、 $pc2$ が Q_c2 に等しくない時、 $u1$ と $u4$ を用いて座標 ($-u1, -pc2$) に指向する接線剛性を計算し AK にセットする。
85. $pt2$ が Q_t2 に等しくなく、 $pc2$ が Q_c2 に等しい場合、最大圧縮ひずみ $u1$ に圧縮側第 2 勾配のひずみ $u5$ をセットする。
86. $pt2$ が Q_t2 に等しくなく、 $pc2$ が Q_c2 に等しくない場合、 $u1$ と $u4$ を用いて座標 ($-u1, -pc2$) に指向する接線剛性を計算し AK にセットする。
87. 状態が -1 の時以下の処理を行う。
88. 初期の増分ひずみが引張の場合 ($du > 0$)、状態を 1 とし以下の処理を行う。
89. 初期の増分ひずみが圧縮の場合 ($du < 0$)、状態を 0 とし以下の処理を行う。
90. 圧縮・引張両方向の第 1 折れ点、第 2 折れ点のひずみを計算し、各変数にセットする。
91. 除荷状態を表すパラメーターを 0 とする。

この履歴モデルは、現在開発中である。

**4.2.9 降伏棚を有
する対称バイリ
ニア型（移動＋等
方硬化用）**

この履歴モデルは、現在開発中である。

**4.2.10 降伏棚を
有する対称トリ
ニア型（移動＋等
方硬化用）**

4.2.11 鉄筋用履歴モデル

本節では、ファイバーに関する履歴ルールで、図 4-24 に示す鉄筋の履歴ルー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。

第 1 折れ点の応力	: Q_1	圧縮側最大ひずみ : u_1
除荷時折れ点の応力 P_1	: $0.5Q_1$	引張側最大ひずみ : u_2
第 1 勾配	: AK_1	第 2 勾配 : AK_2

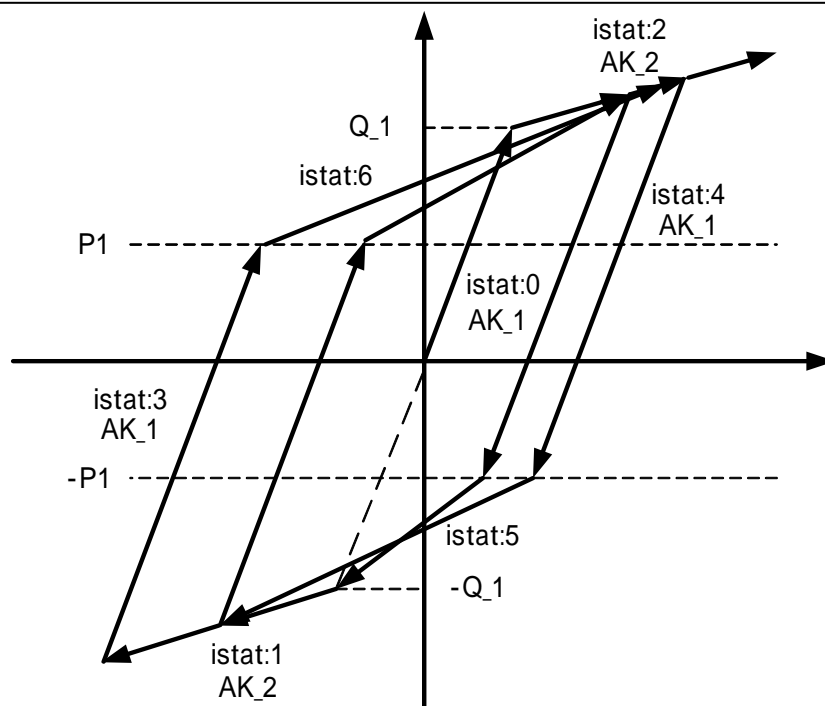


図 4-24 ファイバーモデルの鉄筋用履歴特性

この履歴特性は鉄筋の履歴として用いられる。ファイバーの弾塑性状態は、istat:0 では弾性状態を表している。また、istat:3 と istat:4 は除荷時の状態を表している。

istat:2 と istat:6 は引張降伏以降の状態を表しており、istat:1 と istat:5 は圧縮降伏以降の状態を表している。しかし、istat:5 は圧縮に進行する時は圧縮側の最大点を狙って進行し、istat:6 は引張に進行する時は引張側の最大点を狙って進行する履歴を描く。

この履歴ルールでは、初期の弾性体内では $Q_1 > -Q_1$ の範囲内で直線履歴を描く。

ここでは、図 4-25 の引張側で第 1 折れ点を経過し除荷する履歴を用いて、この履歴ルールの説明をする。

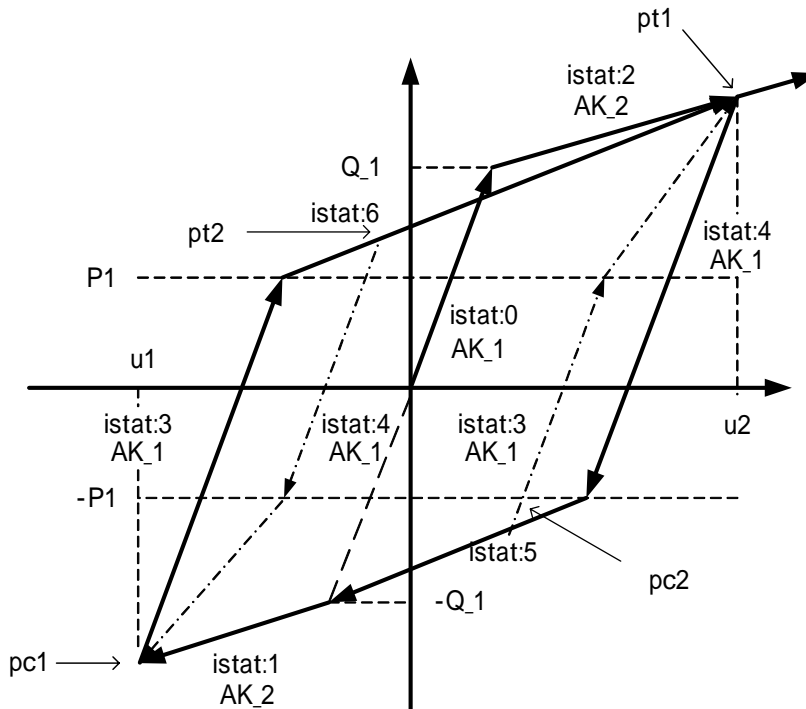


図 4-25 引張側で第1折れ点を経過する履歴

初期のひずみが状態0を引張方向に進行し、引張側の境界値 Q_1 を越えると状態は2へと移行し、接線剛性は AK_2 となる。状態2において増分ひずみが反転すると、図4-2のタイプ2を用いて増分応力と増分ひずみを修正し、増分前のひずみと応力を求め直す。この増分前のひずみを最大引張ひずみ u_2 とし、増分前の応力を pt_1 とする事で引張側の最大点座標 (pt_1, u_2) とする。

状態は4となり、接線剛性は AK_1 となる。状態4では、 $pt_1 > P_1$ の範囲内で直線履歴を描く。この状態4において、ひずみが圧縮方向に進行し圧縮側の境界値 $-P_1$ を越えると状態は5に進行し、座標 ($-Q_1, u_1$) を目指し進行する事になる。

この状態5は圧縮方向にひずみが進行する場合、 $> Q_{c1}$ の範囲内で進行するが、引張方向に増分ひずみが反転すると、状態が3となる。この2通りの履歴について以下にまとめる。

1) 圧縮方向にひずみが進行する場合

ひずみが圧縮方向に進行し、境界値 $-Q_1$ を越えると状態は1に進行する。状態が1に進行してからの履歴は後述する。

2) 引張方向に増分ひずみが反転する場合

増分ひずみが引張方向に反転すると、状態は3となり接線剛性は

AK_1 となる。この時、除荷時の修正を行い、増分前のひずみ、及び、増分前の応力を計算し、ひずみを $u5$ に応力を $pc2$ にセットする。状態 3 は $P1 > pc2$ の範囲内を進行し、 $pc2$ を越えると状態が 5 に戻る。逆に $P1$ を越えると状態は 6 になり、引張側の最大点座標 ($pt1, u2$) を目指し進行する履歴を描く。

状態 5 において圧縮方向にひずみが進行する時、境界値 Q_1 を越えると、状態は 2 となる。この時の接線剛性は AK_2 となり、圧縮方向にひずみが進行する限りその上限はない。状態 2 で増分ひずみが反転し引張方向に進行すると、状態は 6 となる。この時の履歴は 2) の履歴と同様だが、反転した時のひずみが $u1$ 、応力が $pc1$ となり、圧縮側の最大点座標とする。

2) と同様に状態が 6 に進行するとして、状態 6 で増分ひずみが反転し圧縮方向に進行すると状態は 4 となり、接線剛性は AK_1 となる。この時の履歴の描き方は 2) の履歴と同様だが、反転時のひずみを $u6$ 、応力を $pt2$ とし、状態 4 での境界値 $pt2 > P1$ とする。境界値 $P1$ を越えると状態は 5 となり、圧縮側の最大点を目指す履歴となる。

以上の説明より、鉄筋用履歴モデルは 6 つの状態で表される。この状態を以下にまとめる。

- istat : 0 弾性状態であり、接線剛性は第 1 勾配の AK_1 である。 $Q_1 > Q_1$ の中で直線的に動き、 Q_1 を越えると状態 2 となり、 Q_1 を越えると状態 1 となる。
- istat : 1 圧縮側の第 2 勾配上であり、接線剛性は第 2 勾配の AK_2 である。増分ひずみが負の場合、その上限の境界はなく、増分ひずみが反転し正になった場合、状態が 3 となり反転した時の応力を $pc1$ とし、ひずみを $u1$ とする。
- istat : 2 引張側の第 2 勾配上であり、接線剛性は第 2 勾配 AK_2 である。増分ひずみが正の時、その上限はなく、増分ひずみが反転し負となった時、その応力を $pt1$ としひずみを $u2$ とする。
- istat : 3 圧縮側の除荷勾配上であり、接線剛性は第 1 勾配 AK_1 である。 $P1 > pc1$ の範囲内で直線的に動き、 $P1$ を越えると状態は 6 となり、 $pc1$ を越えると状態は 1 に進行する。
- istat : 4 引張側の除荷勾配上であり、接線剛性は第 1 勾配 AK_1 である。 $pt1 > P1$ の範囲内で直線的に動き、 $pt1$ を越えると状態は 2 となり、 $P1$ を越えると状態は 5 に進行する。

istat:5 圧縮側の最大点を指向する勾配であり、増分ひずみが圧縮の時
その境界は $>pc1$ であり、増分ひずみが反転し圧縮となった時、
その応力を $pc2$ として状態は3となる。

istat:6 引張側の最大点を指向する勾配であり、増分ひずみが引張の時
その境界は $pt1 >$ であり、増分ひずみが反転し圧縮となった
時、その応力を $pt2$ として状態は4となる。

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 /reinforcement_b ( 鉄筋用履歴モデル )
C
      subroutine reinforcement_b(AK,istat,AK_1,AK_2,Q_1,du,P,
*          P1,Pt1,Pc1,Pt2,Pc2,st,u6,u5,ug,u2,u1,AK_t1,AK_c1)
      implicit real*8(A-H,O-Z)
C
C      AK              : 接線剛性
C      istat           : 現在の状態(Work)
C      AK_1            : 第1勾配
C      AK_2            : 第2勾配
C      AK_t1           : 引張側最高点指向勾配
C      AK_c1           : 圧縮側最高点指向勾配
C      Q_1             : 第1折れ点の応力度
C      du              : 増分変位
C      P               : 現在の応力
C      P1              : 除荷時の反曲点
C      Pt1             : istat=4,6における反曲点上端(Work)
C      Pt2             : istat=4における反曲点上端(Work)
C      Pc1             : istat=3,5における反曲点上端(Work)
C      Pc2             : istat=3における反曲点上端(Work)
C
100  continue
C
C                                     istat:0( 初期弾性域 )
      if (istat.eq.0)then
      p=AK*du+p
      if(du.lt.0)then
      if(p.gt.pc1)return
C                                     istat:0 から istat:1 へ(圧縮)
      istat=1
      du2=(p-pc1)/AK
      AK=AK_2
      p=AK*du2+pc1
      return
      else
      if(p.lt.pt1)return
C                                     istat:0 から istat:2 へ(引張)
      istat=2
      du2=p/AK
      AK=AK_2
      p=AK*du2+pt1

```

```

        return
    endif
C                                     istat:1(圧縮側)
    elseif (istat.eq. 1)then
        p=AK*du+p
        if(du.lt.0)return
C                                     istat:1から istat:3へ(除荷)
        istat=3
        dp=AK*du
        AK=AK_1
        du=dp/AK
        p=p-dp
        pc1=p
        u1=ug
        p=AK*du+p
        return
C                                     istat:2(引張側)
    elseif(istat.eq.2)then
        p=AK*du+p
        if(du.gt.0)return
C                                     istat:2から istat:4へ(除荷)
        istat=4
        dp=AK*du
        AK=AK_1
        du=dp/AK
        p=p-dp
        pt1=p
        u2=ug
        p=AK*du+p
        return
C                                     istat:3(圧縮側)
    elseif(istat.eq.3)then
        p=AK*du+p
        if(du.lt.0)then
            if(st.eq.0)then
                if(p.gt.pc1)return
C                                     istat:3から istat:1へ(圧縮)
                istat=1
                du2=(p+pc1)/AK
                AK=AK_2
                p=AK*du2+pc1
                return
            else
                if(p.gt.pc2)return
C                                     istat:3から istat:5へ(圧縮)
                istat=5
                st=0
                du2=(p-pc2)/AK
                AK=AK_c1
                p=AK*du2+pc2
                return
            endif
        else
            if(p.lt.p1)return

```

```

C                                istat:3から istat:6へ(引張)
    istat=6                                !23
    du2=(p-p1)/AK
    if(st.eq.0)then
    u3=-(p1-pc1)/AK_1-u1+u2                !24
    AK=(pt1-p1)/u3                        !25
    else
    u3=-(p1-pc2)/AK_1-u5+u2                !26
    AK=(pt1-p1)/u3                        !27
    st=0                                !28
    endif
    p=AK*du2+p1
    return
endif

C                                istat:4(引張側)
    elseif(istat.eq.4)then                !29
    p=AK*du+p
    if(du.gt.0)then
    if(st.eq.0)then
    if(p.lt.pt1)return                    !30
C                                istat:4から istat:2へ(引張)
        istat=2                                !31
        du2=(p-pt1)/AK
        AK=AK_2
        p=AK*du2+pt1
        return
    else
    if(p.lt.pt2)return                    !32
C                                istat:4から istat:6へ(引張)
        istat=6                                !33
        st=0                                !34
        du2=(p-pt2)/AK
        AK=AK_t1
        p=AK*du2+pt2
        return
    endif
    else
    if(p.gt.-p1)return                    !35
C                                istat:4から istat:5へ(引張)
        istat=5                                !36
        du2=(p+p1)/AK
        if(st.eq.0)then
        u4=-(p1+pt1)/AK_1-u1+u2            !37
        AK=(-pc1-p1)/u4                    !38
        else
        u4=-(p1+pt2)/AK_1-u1+u6            !39
        AK=(-pc1-p1)/u4                    !40
        st=0                                !41
        endif
        p=AK*du2-p1
        return
    endif
C                                istat:5(圧縮側)
    elseif(istat.eq.5)then                !42

```

	p=AK*du+p		
	if (du.lt.0) then		
	if (p.gt.pc1) return	!43	
C		istat:5 から istat:1 へ(圧縮)	
	istat=1	!44	
	du2=(p-pc1)/AK		
	AK=AK_2		
	p=AK*du2+pc1		
	return		
	else		
C		istat:5 から istat:3 へ(除荷)	
	istat=3	!45	
	st=1	!46	
	dp=AK*du		
	AK_c1=AK		
	AK=AK_1		
	du=dp/AK		
	p=p-dp		
	pc2=p	!47	
	u5=ug	!48	
	p=AK*du+p		
	return		
	endif		
C		istat:6(引張側)	
	elseif (istat.eq.6) then	!49	
	p=AK*du+p		
	if (du.gt.0) then		
	if (p.lt.pt1) return	!50	
C		istat:6 から istat:2 へ(引張)	
	istat=2	!51	
	du2=(p-pt1)/AK		
	AK=AK_2		
	p=AK*du2+pt1		
	return		
	else		
C		istat:6 から istat:4 へ(除荷)	
	istat=4	!52	
	st=1	!53	
	dp=AK*du		
	AK_t1=AK		
	AK=AK_1		
	du=dp/AK		
	p=p-dp		
	pt2=p	!54	
	u6=ug	!55	
	p=AK*du+p		
	return		
	endif		
C		初期設定	
	else	!56	
	istat=0		
	AK=AK_1		
	p=0		
	pt1=Q_1	!57	

```

pc1=-Q_1
p1=Q_1/2
u1=-Q_1/AK_1                                     !58
u2=-u1
st=0                                              !59
goto 100
end if
end

```

1. 状態が $istat : 0$ の場合、以下の処理を行う。
2. 増分ひずみが圧縮の場合 ($du < 0$)、増分後の応力が圧縮側の境界値 $pc1$ より大きければ、このサブルーチンより戻る。
3. 増分後の応力が $pc1$ より小さい場合は、状態が 1 となり、以下の処理を行う。
4. 増分ひずみが引張の場合 ($du > 0$)、増分後の応力が $pt1$ よりも小さければ、このサブルーチンより戻る。
5. 増分後の応力が $pt1$ よりも大きい場合は、状態が 2 となり、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. 増分ひずみが圧縮の場合 ($du < 0$)、このサブルーチンより戻る。
8. 増分ひずみが引張の場合 ($du > 0$)、状態が 3 となり、以下の処理を行う。
9. 増分前の応力を求め、 $pc1$ にセットする。
10. 前ステップまでの全ひずみを $u1$ にセットする。
11. 状態が $istat : 2$ の場合、以下の処理を行う。
12. 増分ひずみが引張の場合 ($du > 0$)、このサブルーチンより戻る。
13. 増分ひずみが圧縮の場合 ($du < 0$)、状態が 4 となり、以下の処理を行う。
14. 増分前の応力を求め、 $pt1$ にセットする。
15. 前ステップまでの全ひずみを $u2$ にセットする。
16. 状態が $istat : 3$ の場合、以下の処理を行う。
17. 増分ひずみが圧縮の場合 ($du < 0$) で、除荷状態を表すパラメーター st が 0 の時、増分後の応力が圧縮側の境界値 $pc1$ より大きければこのサブルーチンより戻る。
18. 増分ひずみが圧縮の場合 ($du < 0$) で、除荷状態を表すパラメーター st が 0 の時、増分後の応力が圧縮側の境界値 $pc1$ より小さければ状態は 1 となり、以下の処理を行う。
19. 増分ひずみが圧縮の場合 ($du < 0$) で、除荷状態を表すパラメーター st が 1 の時、増分後の応力が圧縮側の境界値 $pc2$ より大きければこ

のサブルーチンより戻る。

20. 増分ひずみが圧縮の場合 ($du < 0$) で、除荷状態を表すパラメーター st が 1 の時、増分後の応力が圧縮側の境界値 $pc2$ より小さければ、状態は 5 となり、以下の処理を行う。
21. 除荷状態を表すパラメーター st を 0 にする。
22. 増分ひずみが引張の場合 ($du > 0$) 増分後の応力が $p1$ よりも小さければ、このサブルーチンより戻る。
23. 増分後の応力が $p1$ よりも大きい場合、状態が 6 となり、以下の処理を行う。
24. 圧縮側の最大ひずみと引張側の最大ひずみを用いて、境界値 $p1$ 時のひずみと、引張側最大ひずみ間のひずみを求め、 $u3$ にセットする。
25. 引張側の最大点座標 ($u2, pt1$) を指向する接線剛性 AK を $u3$ を用いて計算する。
26. 除荷状態を表すパラメーターである st が 1 の時、圧縮側から除荷した時のひずみ $u5$ を用いて、境界値 $p1$ 時のひずみと引張側最大ひずみ間のひずみを求め、 $u3$ にセットする。
27. 引張側の最大点座標 ($u2, pt2$) を指向する接線剛性 AK を $u3$ を用いて計算する。
28. 除荷状態を表すパラメーター st を 0 にする。
- 29. 状態が $istat : 4$ の場合、以下の処理を行う。**
30. 増分ひずみが引張の場合 ($du > 0$) で状態を表すパラメーター st が 0 の時、増分後の応力が $pt1$ よりも小さければ、このサブルーチンより戻る。
31. 増分後の応力が $pt1$ よりも大きい場合は、状態が 2 となり、以下の処理を行う。
32. 増分ひずみが引張の場合 ($du > 0$) で状態を表すパラメーター st が 1 の時、増分後の応力が $pt2$ よりも小さければ、このサブルーチンより戻る。
33. 増分後の応力が $pt2$ よりも大きい場合、状態が 6 となり、以下の処理を行う。
34. 除荷状態を表すパラメーター st を 0 にする。
35. 増分ひずみが圧縮の場合 ($du < 0$) で、増分後の応力が $-p1$ よりも大きければ、このサブルーチンより戻る。
36. 増分後の応力が $-p1$ よりも小さい場合は状態が 5 となり、以下の処理を行う。
37. 引張側の最大ひずみと圧縮側の最大ひずみを用いて、境界値 $-p1$ 時

- のひずみと、圧縮側最大ひずみ間のひずみを求め、 u_4 にセットする。
38. 圧縮側の最大点座標 (u_1, pc_1) を指向する接線剛性 AK を u_4 を用いて計算する。
39. 除荷状態を表すパラメーターである st が 1 の時、引張側から除荷した時のひずみ u_6 を用いて、境界値 - p_1 時のひずみと圧縮側最大ひずみ間のひずみを求め、 u_4 にセットする。
40. 圧縮側の最大点座標 (u_1, pc_2) を指向する接線剛性 AK を u_4 を用いて計算する。
41. 除荷状態を表すパラメーター st を 0 にする。
- 42. 状態が $istat : 5$ の場合、以下の処理を行う。**
43. 増分ひずみが圧縮の場合 ($du < 0$) で、増分後の応力が pc_1 よりも大きければ、このサブルーチンより戻る。
44. 増分後の応力が pc_1 よりも小さい場合、状態が 1 となり、以下の処理を行う。
45. 増分ひずみが引張の場合 ($du > 0$) 状態が 3 となり、以下の処理を行う。
46. 除荷状態を表すパラメーター st を 1 にする。
47. 増分前の応力を求めて pc_2 にセットする。
48. 増分前のひずみを圧縮側の最大ひずみ u_5 にセットする。
- 49. 状態が $istat : 6$ の場合、以下の処理を行う。**
50. 増分ひずみが引張の場合 ($du > 0$) で、増分後の応力が pt_1 より小さければ、このサブルーチンより戻る。
51. 増分後の応力が pt_1 よりも大きい場合、状態が 2 となり、以下の処理を行う。
52. 増分ひずみが圧縮の場合 ($du < 0$) 状態を 4 とし、以下の処理を行う。
53. 除荷状態を表すパラメーター st を 1 にする。
54. 増分前の応力を求めて pt_2 にセットする。
55. 増分前のひずみを引張側の最大ひずみ u_6 にセットする。
- 56. 状態が $istat : -1$ の場合、初期設定を行う。** まず、状態パラメーターを $istat : 0$ とし、初期の接線剛性として第 1 勾配の剛性 AK_1 をセットする。さらに、応力をゼロにする。
57. 初期の引張側の境界値 pt_1 に Q_1 をセットし、圧縮側の境界値 pc_1 に $-Q_1$ をセットする。
58. 初期の圧縮側最大ひずみ u_1 と引張側最大ひずみ u_2 を Q_1 と AK_1 を用いて計算する。
59. 除荷状態を表すパラメーター st をゼロにセットする。

4.2.12 木質構造材用
履歴モデル

本節では、ファイバーの履歴ルールで図 4-26 に示す木質構造材用履歴モデルについて説明する。図中の記号は以下の通りである。

引張側第 1 折れ点の応力 : Q_{t1}	圧縮側第 1 折れ点の応力 : $-Q_{c1}$
引張側第 2 折れ点の応力 : Q_{t2}	圧縮側第 2 折れ点の応力 : $-Q_{c2}$
引張側第 1 勾配 : AK_{t1}	圧縮側第 1 勾配 : AK_{c1}
引張側第 2 勾配 : AK_{t2}	圧縮側第 2 勾配 : AK_{c2}
引張側第 3 勾配 : AK_{t3}	圧縮側第 3 勾配 : AK_{c3}
istat:3 における勾配 : AK_{t4}	istat:7 における勾配 : AK_{c4}
istat:8 における勾配 : AK_{t5}	

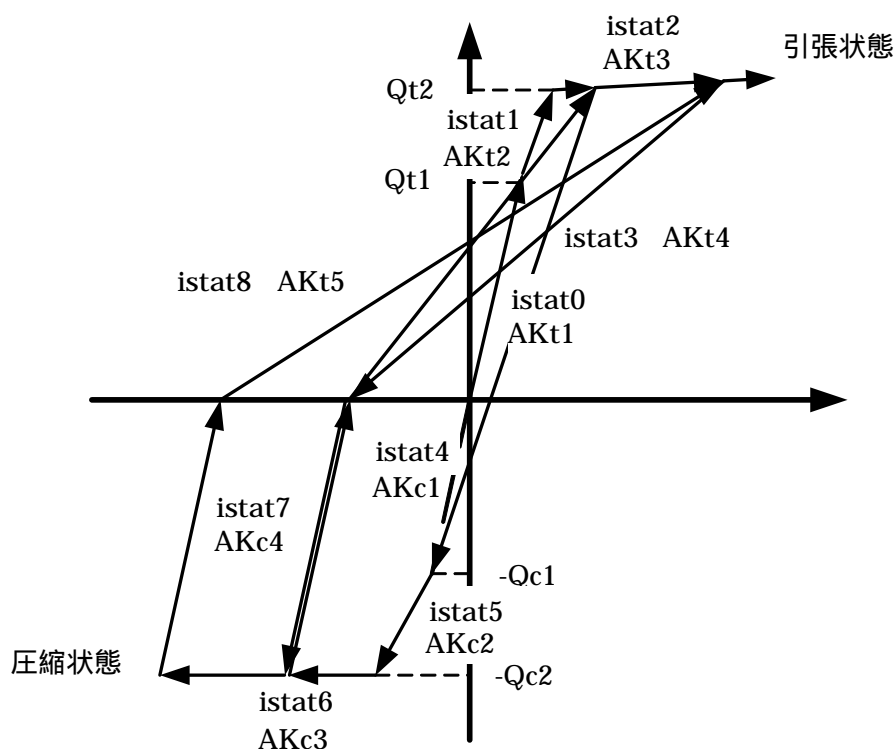


図 4-26 木質構造材用履歴モデル

木材は天然の生物材料であるため様々な特異性を示す。これは構造用材料としてみた場合でも同じことが言える。木材の化道管や木繊維の配列方向（軸方向）においても引張方向と圧縮方向で異なる特性をもち、そのため、実際の応力 ひずみ関係をよく模擬する履歴特性が必要となる。木材の履歴特性は、引張側は最大耐力までは弾性的な挙動を示すが、それ以降はほとんど耐力が期待できない極めて脆性的な破壊を示す性質、圧縮側は最大荷重を超えた後も耐力を保持したままで推移す

る延性的な性質を表している。ここでは、このような特性を比較的模擬している図 4-26 の履歴モデルを木質構造材用履歴モデルとして採用する。

図 4-26 の中で、ファイバーの弾塑性状態を表す $istat:0$ は引張側弾性状態を表し、 $istat:4$ は圧縮側弾性状態を表す。また、 $istat:7$ は圧縮側で除荷された時の状態を表している。降伏以降では、 $istat:1$ と $istat:2$ が引張降伏以後の状態を、 $istat:5$ と $istat:6$ が圧縮降伏以後の状態を表す。

プレゼンターの断面表示では、第2折れ点以降を塑性状態としている。

$istat:3$ は圧縮にひずみが進行する場合は圧縮側の最大点を狙って進行し、 $istat:8$ は引張にひずみが進行する場合は引張側の最大点を狙って進行する履歴を描く。

この履歴ルールを大きく分けて下記の3つに分けて検討する。

- 1) 弾性状態で第1折れ点を経過せず繰り返す履歴
- 2) 第1折れ点を経験して除荷する履歴
- 3) 第2折れ点を経験して除荷する履歴

まず、1) の弾性状態で第1折れ点を経過せず繰り返す履歴について図 4-27 を用いて説明する。

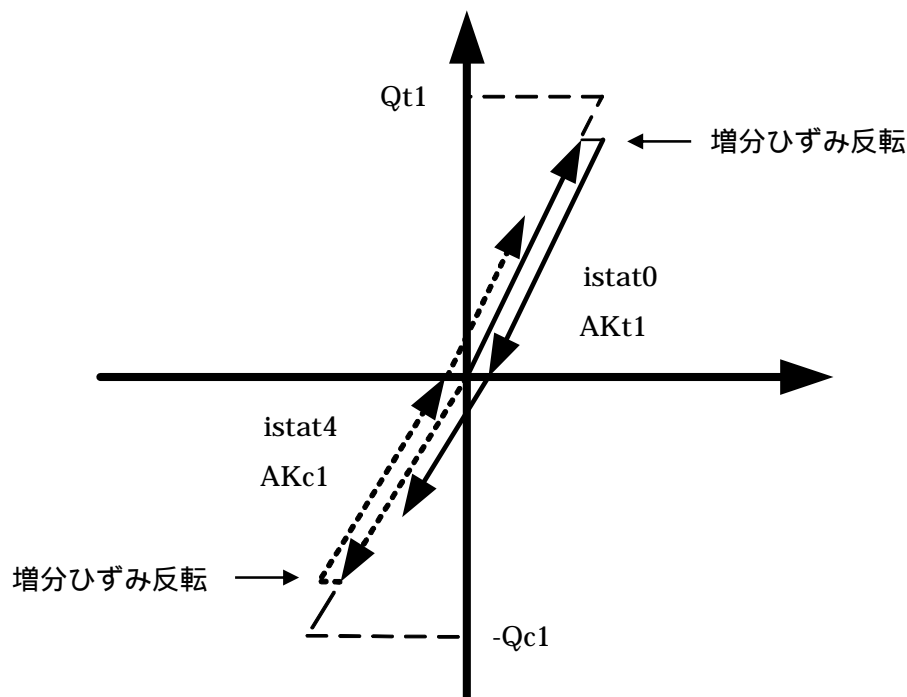


図 4-27 弾性状態で第1折れ点を経過せず繰り返す履歴

図 4-27 に示すように、初めに得られる増分ひずみが引張ならば $istat:0$ となり、圧縮ならば $istat:4$ となる。 $istat:0$ の境界値は $Qt1 > 0$ であり、 $Qt1$ を超えずに増分ひずみが反転すると、そのまま $istat:0$ を直線的に進行する。同様に、 $istat:4$ の境界値は $0 > -Qc1$ であり、 $-Qc1$ を超えない限り弾性状態として $istat:0$ を直線的に進行する。また、 $istat:0$ 、 $istat:4$ において、境界値 0 を越えると $istat:0$ は $istat:4$ に、 $istat:4$ は $istat:0$ となり、弾性範囲内を進行する。

次に、2) の第 1 折れ点を経験して除荷する履歴について図 4-28 に示す線 1、線 2 を用いて説明する。

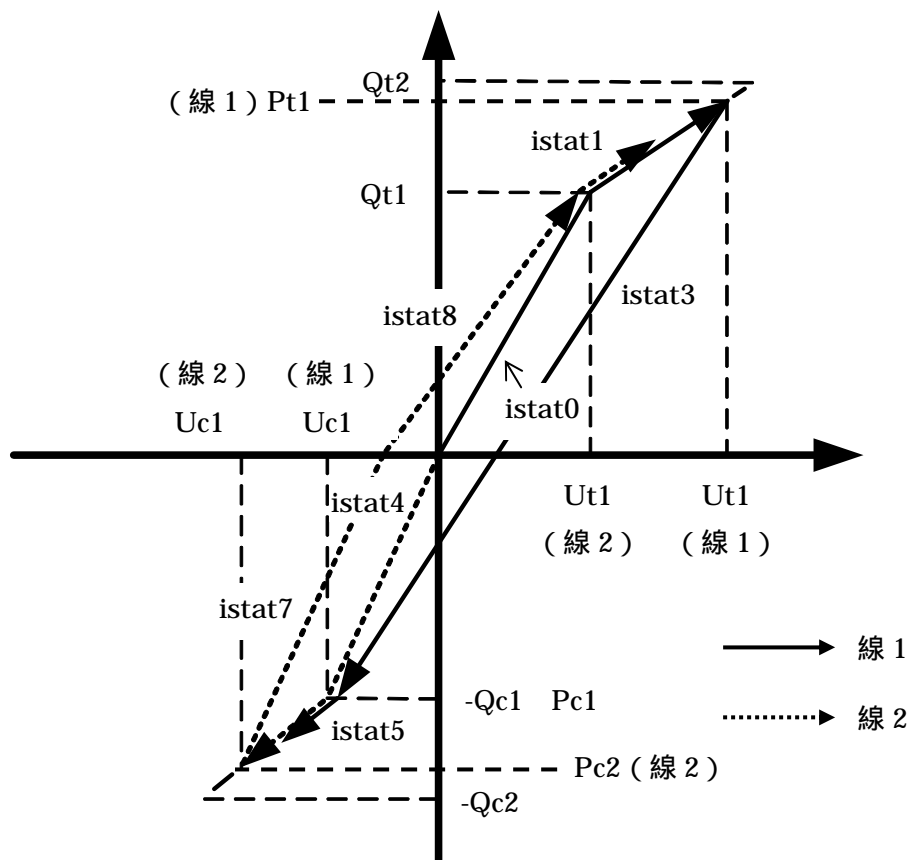


図 4-28 第 1 折れ点を経験して除荷する履歴

線 2 では、初めに得られる増分ひずみが圧縮の場合、状態は $istat:4$ となり、境界値 $-Qc1$ を越えると $istat:5$ に進行する。この $istat:5$ においてひずみが反転すると $istat:7$ となり、増分前の応力を $Pc2$ とする。この時の境界は $Pc2 < 0$ であり、釣合点はこの範囲内で直線的に進行する。引張方向に増分ひずみが進行し境界値 0 を越えると $istat:8$ となり（境界値 0 の点を $Pc1$ 、 $Uc1$ とする）、釣合点は引張側の最大点 $Pt1$ 、

Ut1 を指向する。istat:8 の勾配 AKt5 は、圧縮側の Pc1、Uc1 と引張側の Pt1、Ut1 より計算し、この勾配でひずみ増分時と除荷時共に $Pc1 < Pt1$ の境界内を進行する。増分ひずみが引張側に進行して応力が Pt1 になると istat:1 になる。

線 1 では、初めに得られる増分ひずみが引張の場合、状態は istat:0 となり、境界値 Qt1 を越えると istat:1 に進行する。この istat:1 においてひずみが反転すると istat:3 となり、釣合点は、圧縮側の Pc1 (Uc1) を指向する。勾配 AKt4 は、反転時の応力 Pt1、変位 Ut1 と圧縮側の Pc1、Uc1 より計算される。増分ひずみが圧縮側に進行し、応力が Pc1 になると状態は istat:5 または istat:7 になる。Pc1 の値は istat:5 を経験するまでは -Qc1 である。しかし、istat:5 を経験すると Pc1 は 0 となり図 4-28 のような履歴になる。

最後に、3) の第 2 折れ点を経験して除荷する履歴について図 4-29 を用いて説明する。

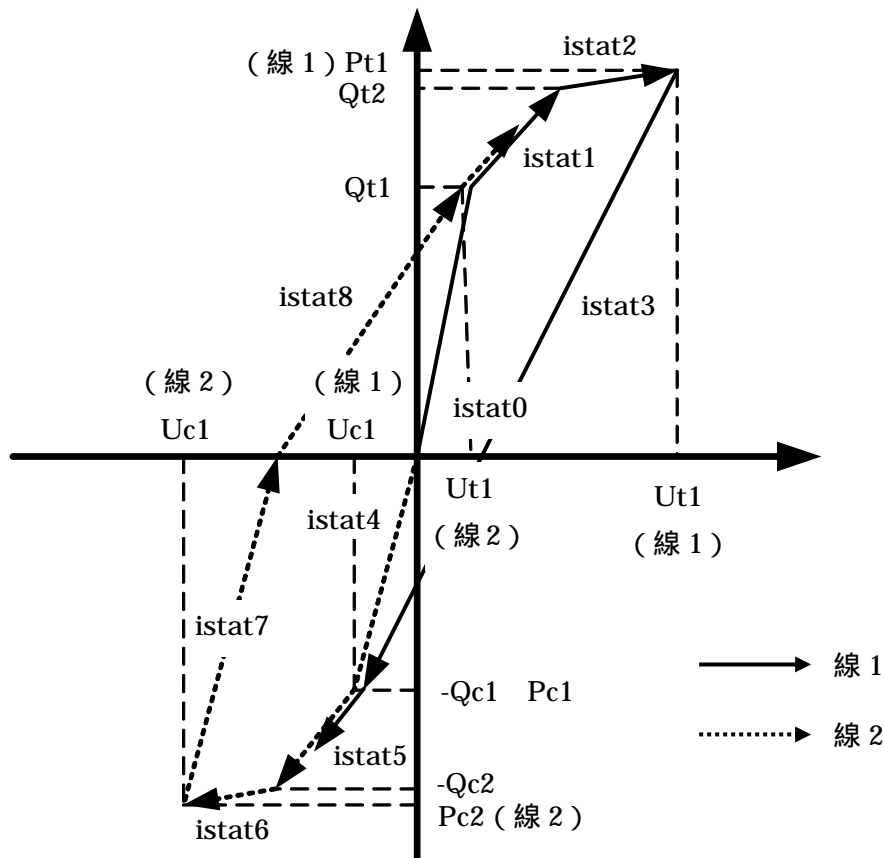


図 4-29 第 1 折れ点を経験して除荷する履歴

線 1 において、ひずみが引張方向に進行し、Qt2 を越えると状態は

istat:2 となる。istat:2 において除荷が生じ、増分ひずみが反転し圧縮方向に進行する場合は、2) の引張側第1折れ点(Q_{t1})を経験して除荷する履歴と同様である。線2 においても、ひずみが圧縮側に進行し Q_{c2} を越えると istat:6 に進行し、除荷が生じ増分ひずみが反転したときは2) の圧縮側第一折れ点(Q_{c1})を経験して除荷する履歴と同様に進行する。

以上の説明をまとめると、木質構造材用の履歴特性は、次の9つの状態で表現される。

- 1) istat:0 引張側の弾性状態であり、接線剛性は引張側の第1勾配 AK_{t1} である。釣合点は $Q_{t1} > 0$ の中で直線的に動き、 Q_{t1} を越えると istat:1 となり、0 を越えると istat:4 となる。
- 2) istat:1 引張側の第2勾配上であり、接線剛性は引張側の第2勾配 AK_{t2} である。増分ひずみが引張方向の時、その境界は $Q_{t2} > 0$ であり、 Q_{t2} を越えると istat:2 となる。増分ひずみが反転し圧縮方向になった時、istat:3 となり増分前の応力を P_{t1} とする。
- 3) istat:2 引張側の第3勾配上であり、接線剛性は引張側の第3勾配 AK_{t3} である。増分ひずみが引張方向の時その境界はなく、増分ひずみが反転し圧縮方向になった時、istat:3 となり、増分前の応力を P_{t1} とする。
- 4) istat:3 圧縮側の最大点を指向する勾配上であり、 $P_{t1} > P_{c1}$ の範囲内で直線的に履歴を描く。引張側の境界値 P_{t1} を越えると istat:1 または istat:2 になり、圧縮側の境界値 P_{c1} を越えると istat:5 または istat:7 となる。
- 5) istat:4 圧縮側の弾性状態であり、接線剛性は圧縮側の第1勾配 AK_{c1} である。釣合点は $0 > -Q_{c1}$ の中で直線的に動き、 $-Q_{c1}$ を越えると istat:5 となり、0 を越えると istat:0 となる。
- 6) istat:5 圧縮側の第2勾配上であり、接線剛性は圧縮側の第2勾配 AK_{c2} である。増分ひずみが圧縮方向の場合、その上限は $-Q_{c2}$ であり、 $-Q_{c2}$ を越えると istat:6 となる。増分ひずみが反転し引張方向になった場合、istat:7 となり、反転した時の増分前の応力を P_{c2} とする。
- 7) istat:6 圧縮側の第3勾配上であり、接線剛性は圧縮側の第3勾配 AK_{c3} である。増分ひずみが圧縮方向の場合、その上限はなく、増分ひずみが反転し引張方向になった場合、istat:7 となり、反転した時の増分前の応力を P_{c2} とする。
- 8) istat:7 圧縮側の除荷時の勾配上であり、接線剛性は圧縮側の除荷

勾配の AKc4 である。釣合点は $0 > p_{c2}$ の範囲内で直線的に履歴を描き、圧縮側の境界値 P_{c2} が $-Q_{c2}$ より小さい時、その値を越えると istat:5 に進み、圧縮側の境界値 P_{c2} が $-Q_{c2}$ より大きい場合は、istat:6 に進行する。増分ひずみが引張方向に進行する時、境界値 0 を越えると istat:8 に進行する。

- 9) istat:8 引張側の最大点を指向する勾配上であり、釣合点は $P_{t1} > P_{c1}$ の範囲内で直線的に履歴を描く。引張側の境界値 P_{t1} が Q_{t2} より小さい時、その値を越えると istat:1 に進み、引張側の境界値 P_{t1} が Q_{t2} より大きい場合は、その境界値を越えると istat:2 に進行する。増分ひずみが圧縮方向に進行する時、境界値 0 を越えると istat:7 に進行する。

次に、上記の履歴ルールにしたがって記述されたサブルーチン Wood_TriLnear() を具体的に見てみよう。

```

c
c      SUBROUTINE/Wood_TriLnear ( 木質構造材用履歴モデル )
c
c      subroutine Wood_TriLinear(ak,istat,akt1,akt2,akt3,qt1,qt2,akc1,
*          akc2,akc3,qc1,qc2,du,p,pt1,pc1,pc2,ug,ut1,uc1)
c      implicit real*8(A-H,O-Z)
c
c      AK          : 接線剛性
c      istat       : 現在の状態(Work)
c      AKt1        : 引張側第一勾配
c      AKt2        : 引張側第二勾配
c      AKt3        : 引張側第三勾配
c      Qt1         : 引張側第一折れ点
c      Qt2         : 引張側第二折れ点
c      AKc1        : 圧縮側第一勾配
c      AKc2        : 圧縮側第二勾配
c      AKc3        : 圧縮側第三勾配
c      Qc1         : 圧縮側第一折れ点
c      Qc2         : 圧縮側第二折れ点
c      du          : 増分変位
c      P           : 現在応力
c      Pt1         : istat8 の折れ点、及び、除荷した時の反曲点
c      Pc1         : istat3 の折れ点、及び、istat7 の折れ点
c      Pc2         : istat7 の折れ点、及び、除荷した時の反曲点
c      ug          : 現在の変位
c
100 continue
c
c          istat=0
c      if(istat.eq.0)then
c          P=AK*du+P
c          if(du.gt.0.)then
c              if(P.lt.Pt1)return

```

```

c                                istat=0 から istat=1 へ
    istat=1                                !3
    du2=(P-Pt1)/AK
    AK=AKt2
    P=AK*du2+Pt1
    return
else
    if(P.gt.0.)return                                !4
c                                istat=0 から istat=4 へ
    istat=4                                !5
    du2=P/AK
    AK=AKc1
    P=du2*AK
    return
end if
c                                istat=1
    else if(istat.eq.1)then                                !6
        if(du.gt.0.)then
            P=AK*du+P
            if(P.lt.Qt2)return                                !7
c                                istat=1 から istat=2 へ
            istat=2                                !8
            du2=(P-Qt2)/AK
            AK=AKt3
            P=AK*du2+Qt2
            return
        else
c                                istat=1 から istat=3 へ
            istat=3                                !9
            Pt1=P                                !10
            ut1=ug
            AKt4=(Pt1-Pc1)/(ut1-uc1)                                !11
            P=AK*du+P
            dp=AK*du
            AK=AKt4
            du=dp/AK
            P=AK*du+Pt1
            return
        end if
c                                istat=2
    else if(istat.eq.2)then                                !12
        if(du.gt.0.)then
            P=AK*du+P
            return
        else
c                                istat=2 から istat=3 へ
            istat=3                                !13
            Pt1=P                                !14
            ut1=ug
            AKt4=(Pt1-Pc1)/(ut1-uc1)                                !15
            P=AK*du+P
            dp=AK*du
            AK=AKt4
            du=dp/AK

```

```

    P=AK*du+pt1
    return
end if
c                                istat=3
else if (istat.eq.3) then      !16
    P=AK*du+P
    if (du.gt.0.) then
    if (P.lt.Pt1) return      !17
    if (Pt1.lt.Qt2) then      !18
c                                istat=3 から istat=1 へ
    istat=1                    !19
    du2=(P-Pt1)/AK
    AK=AKt2
    P=AK*du2+Pt1
    return
    else
c                                istat=3 から istat=2 へ
    istat=2                    !20
    du2=(P-Pt1)/AK
    AK=AKt3
    P=AK*du2+Pt1
    return
    end if
    else
    if (P.gt.Pc1) return      !21
    if (Pc1.eq.-Qc1) then      !22
c                                istat=3 から istat=5 へ
    istat=5                    !23
    du2=(P-Pc2)/AK
    AK=AKc2
    P=AK*du2+Pc2
    return
    else
c                                istat=3 から istat=7 へ
    istat=7                    !24
    du2=P/AK
    AK=AKc1
    P=AK*du2
    return
    end if
    end if
c                                istat=4
else if (istat.eq.4) then      !25
    P=AK*du+P
    if (du.gt.0.) then
    if (P.lt.0.) return      !26
c                                istat=4 から istat=0 へ
    istat=0                    !27
    du2=P/AK
    AK=akt1
    P=du2*AK
    return
    else
    if (P.gt.pc2) return      !28

```

```

c                                istat=4 から istat=5 へ
    istat=5                                !29
    du2=(P-Pc2)/AK
    AK=AKc2
    P=AK*du2+Pc2
    return
end if

c                                istat=5                                !30
else if(istat.eq.5)then
    P=AK*du+P
    if(du.gt.0.)then
c                                istat=5 から istat=7 へ                                !31
        istat=7
        dp=AK*du
        AK=AKc1
        du=dp/AK
        P=P-dp
        Pc2=P                                !32
        P=AK*du+P
        return
    else
        if(p.gt.-Qc2)return                                !33
c                                istat=5 から istat=6 へ                                !34
        istat=6
        du2=(P+Qc2)/AK
        AK=AKc3
        P=AK*du2-Qc2
        return
    end if

c                                istat=6                                !35
else if(istat.eq.6)then
    P=AK*du+P
    if(du.lt.0.)return                                !36
c                                istat=6 から istat=7 へ                                !37
        istat=7
        dp=AK*du
        AK=AKc1
        du=dp/AK
        P=P-dp
        Pc2=P                                !38
        P=AK*du+P
        return

c                                istat=7                                !39
else if(istat.eq.7)then
    P=AK*du+P
    if(du.gt.0.)then
        if(P.lt.0.)return                                !40
c                                istat=7 から istat=8 へ                                !41
        istat=8
        du2=P/AK
        uc1=ug+du-du2                                !42
        Pc1=0
        AKt5=Pt1/(ut1-uc1)                                !43
        AK=AKt5

```

```

    P=AK*du2
    return
  else
    if (P.gt.Pc2) return !44
    if (Pc2.gt.-Qc2) then !45
c      istat=7 から istat=5 へ
      istat=5 !46
      du2=(P-Pc2)/AK
      AK=AKc2
      P=AK*du2+Pc2
      return
    else
c      istat=7 から istat=6 へ
      istat=6 !47
      du2=(P-Pc2)/ak
      AK=AKc3
      P=AK*du2+Pc2
      return
    end if
  end if
c      istat=8
  else if (istat.eq.8) then !48
    P=AK*du+P
    if (du.gt.0.) then
      if (P.lt.Pt1) return !49
      if (Pt1.lt.Qt2) then !50
c        istat=8 から istat=1 へ
        istat=1 !51
        du2=(P-Pt1)/AK
        AK=AKt2
        P=AK*du2+Pt1
        return
      else
c        istat=8 から istat=2 へ
        istat=2 !52
        du2=(P-Pt1)/AK
        AK=AKt3
        P=AK*du2+Pt1
        return
      end if
    else
      if (P.gt.Pc1) return !53
c      istat=8 から istat=7 へ
      istat=7 !54
      du2=P/AK
      AK=AKc1
      P=AK*du2
      return
    end if
c      初期設定
  else !55
    Pt1=Qt1
    Pc1=-Qc1
    Pc2=-Qc1

```

```

    ut1=Pt1/Akt1
    uc1=Pc1/Akc1
    P=0
    if(du.gt.0.)then                                !56
        istat=0
        AK=Akt1
        goto 100
    else                                            !57
        istat=4
        AK=Akc1
        goto 100
    end if
end if
return
end

```

1. 状態が $istat:0$ の場合、以下の処理を行う。
2. 増分ひずみが引張の場合で、増分後の応力が引張側の境界値 $Pt1$ より小さければ、このサブルーチンより戻る。
3. 増分後の応力が $Pt1$ より大きい場合は $istat:1$ となり、以下の処理を行う。
4. 増分ひずみが圧縮の場合、増分後の応力がゼロよりも大きければ、このサブルーチンより戻る。
5. 増分後の応力がゼロよりも小さい場合は、 $istat:4$ となり、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. 増分ひずみが引張の場合で、増分後の応力が引張側の境界値 $Qt2$ より小さければ、このサブルーチンより戻る。
8. 増分後の応力が $Qt2$ より大きい場合は状態が $istat:2$ となり、以下の処理を行う。
9. 増分ひずみが圧縮の場合は $istat:3$ となり、以下の処理を行う。
10. 前ステップまでの応力を $Pt1$ 、ひずみを $ut1$ にセットする。
11. 反曲点の応力 $Pt1$ とひずみ $ut1$ と、指向点の応力 $Pc1$ とひずみ $uc1$ より勾配 $Akt4$ を求める。
12. 状態が $istat:2$ の場合、以下の処理を行う。増分ひずみが引張の場合、応力を計算後、このサブルーチンより戻る。
13. 増分ひずみが圧縮の場合は $istat:3$ となり、以下の処理を行う。
14. 前ステップまでの応力を $Pt1$ 、ひずみを $ut1$ にセットする。
15. 反曲点の応力 $Pt1$ とひずみ $ut1$ と、指向点の応力 $Pc1$ とひずみ $uc1$ より勾配 $Akt4$ を求める。
16. 状態が $istat:3$ の場合、以下の処理を行う。

17. 増分ひずみが引張の場合で、増分後の応力が引張側の境界値 P_{t1} より小さければ、このサブルーチンより戻る。
18. $istat:3$ の境界値 P_{t1} が Q_{t2} より小さければ、以下の処理を行う。
19. 状態を $istat:1$ にする。
20. $istat:3$ の境界値 P_{t1} が Q_{t2} より大きければ、状態を $istat:2$ にし、以下の処理を行う。
21. 増分ひずみが圧縮の場合で、増分後の応力が圧縮側の境界値 P_{c1} より大きければ、このサブルーチンより戻る。
22. $istat:3$ の境界値 P_{c1} が $-Q_{c1}$ と同じであれば、以下の処理を行う。
23. 状態を $istat:5$ にする。
24. $istat:3$ の境界値 P_{c1} が $-Q_{c1}$ でなければ $istat:7$ にし、以下の処理を行う。
25. 状態が $istat:4$ の場合、以下の処理を行う。
26. 増分ひずみが引張の場合で、増分後の応力が境界値 0 より小さければ、このサブルーチンより戻る。
27. 増分後の応力がゼロより大きい場合は $istat:0$ となり、以下の処理を行う。
28. 増分ひずみが圧縮の場合で、増分後の応力が境界値 P_{c2} より大きければ、このサブルーチンより戻る。
29. 増分後の応力が P_{c2} より小さい場合は $istat:5$ となり、以下の処理を行う。
30. 状態が $istat:5$ の場合、以下の処理を行う。
31. 増分ひずみが引張の場合は $istat:7$ となり、以下の処理を行う。
32. 前ステップまでの応力を P_{c2} にセットする。
33. 増分ひずみが圧縮の場合で、増分後の応力が境界値 $-Q_{c2}$ より小さければ、このサブルーチンより戻る。
34. 増分後の応力が $-Q_{c2}$ より大きい場合は $istat:6$ となり、以下の処理を行う。
35. 状態が $istat:6$ の場合、以下の処理を行う。
36. 増分ひずみが圧縮の場合、このサブルーチンより戻る。
37. 増分ひずみが引張の場合は $istat:7$ となり、以下の処理を行う。
38. 前ステップまでの応力を P_{c2} にセットする。
39. 状態が $istat:7$ の場合、以下の処理を行う。
40. 増分ひずみが引張の場合で、尚且つ増分後の応力が境界値 0 より小さければ、このサブルーチンより戻る。
41. 増分後の応力がゼロより大きい場合は $istat:8$ となり、以下の処理を行

- う。
42. 折れ点の変位を $uc1$ に、応力を $Pc1$ をセットする。
 43. 折れ点の応力 $Pc1$ とひずみ $uc1$ と、指向点の応力 $Pt1$ とひずみ $ut1$ より勾配 $AKt5$ を求める。
 44. 増分ひずみが圧縮の場合で、増分後の応力が境界値 $Pc2$ より大きければ、このサブルーチンより戻る。
 45. $istat:7$ の境界値 $Pc2$ が $-Qc2$ より大きければ、以下の処理を行う。
 46. 状態を $istat:5$ にする。
 47. $istat:7$ の境界値 $Pc2$ が $-Qc2$ より小さければ $istat:6$ にし、以下の処理を行う。
 48. 状態が $istat:7$ の場合、以下の処理を行う。
 49. 増分ひずみが引張の場合で、増分後の応力が境界値 $Pt1$ より小さければ、このサブルーチンより戻る。
 50. $istat:8$ の境界値 $Pt1$ が $Qt2$ より小さければ、以下の処理を行う。
 51. 状態を $istat:1$ にする。
 52. $istat:8$ の境界値 $Pt1$ が $Qt2$ より大きければ $istat:2$ にし、以下の処理を行う。
 53. 増分ひずみが圧縮の場合で、増分後の応力が境界値 $Pc1$ より大きければ、このサブルーチンより戻る。
 54. 増分後の応力が $Pc1$ より小さい場合は $istat:7$ となり、以下の処理を行う。
 55. $istat:-1$ のとき（初期）以下の処理を行う。
 56. 初期の増分ひずみが引張の場合、 $istat:0$ とし以下の処理を行う。
 57. 初期の増分ひずみが圧縮の場合、 $istat:4$ とし以下の処理を行う。

4.3 任意型静的縮合モデルにおける接合部モデル

任意型部材モデルにファイバー断面やアナロジーモデルなどの特殊な断面要素を組み込むことができる。ここでは、この任意型部材モデルに組み込むことができる新たな接合部モデルについて説明する。これらの接合部モデルは以下のようなものである。

- 4.3.1 S字型バイリニアスリップモデル
- 4.3.2 S字型トリリニアスリップモデル
- 4.3.3 バイリニア型スリップモデル
- 4.3.4 トリリニア型スリップモデル
- 4.3.5 ボックス型スリップモデル

4.3.1 S字型スリップバイリニア

本節では、接合部に関する履歴ルールで図 4-30 に示す S 字型の対称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 折れ点の応力	: Q_1	正方向最大変位	: $du3$
負方向最大変位	: $du4$	第 1 勾配	: AK_1
第 2 勾配	: AK_2	スリップ勾配	: AK_3
除荷勾配	: AK_4	除荷時の状態	: st

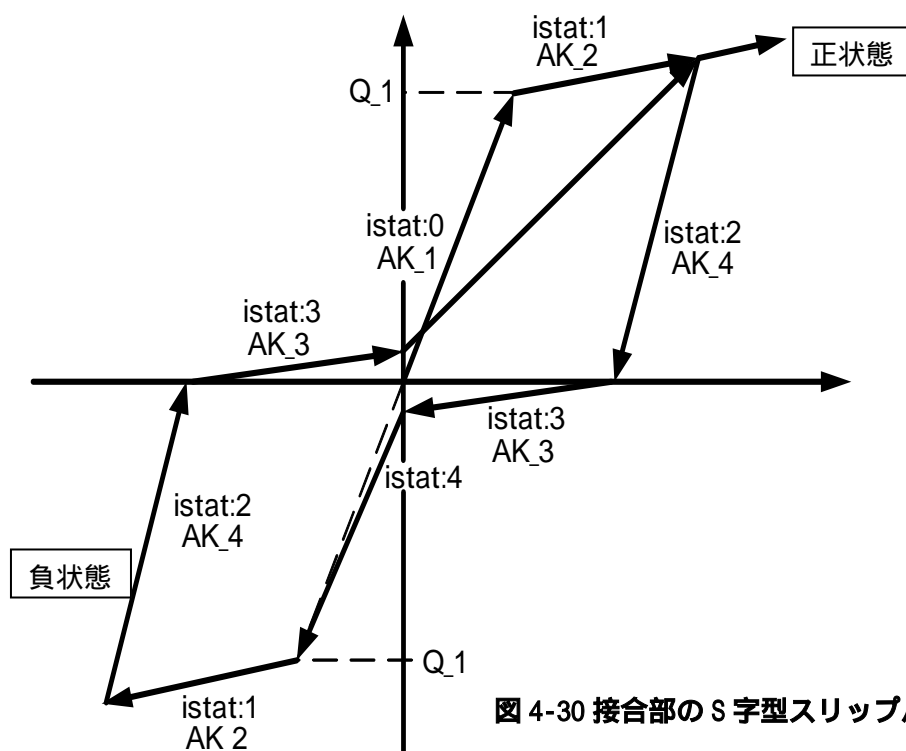


図 4-30 接合部の S 字型スリップバイリニア履歴特性

この履歴特性は木質構造物の接合部の解析を行うために作成した履歴特性であるが、コンクリートのせん断型の履歴としても用いられる事がある。正負対称のバイリニアの骨格曲線であり、除荷時の剛性が初期の剛性と異なる場合においても対応でき、また、スリップ時の剛性もユーザー側で指定できるモデルである。弾性状態は、 $istat:0$ であり、 $istat:2$ は除荷された時の状態を表している。

降伏以降の状態は、 $istat:1$ で表しており、 $istat:3$ は残留変位によるスリップ状態を表す。 $istat:4$ で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、図 4-2 を用いてこの履歴について説明する。図 4-2 は、変位が正方向に進み Q_1 を越えて履歴を描く場合を示す図である。初期では状態 0 であり、この状態の境界値 $Q_1 > -Q_1$ の範囲内では変位が反転してもこの直線上に履歴は描かれる。この履歴特性は正負対称の履歴であり、ここでは正方向の境界値 Q_1 を越えて進行する履歴について説明する。

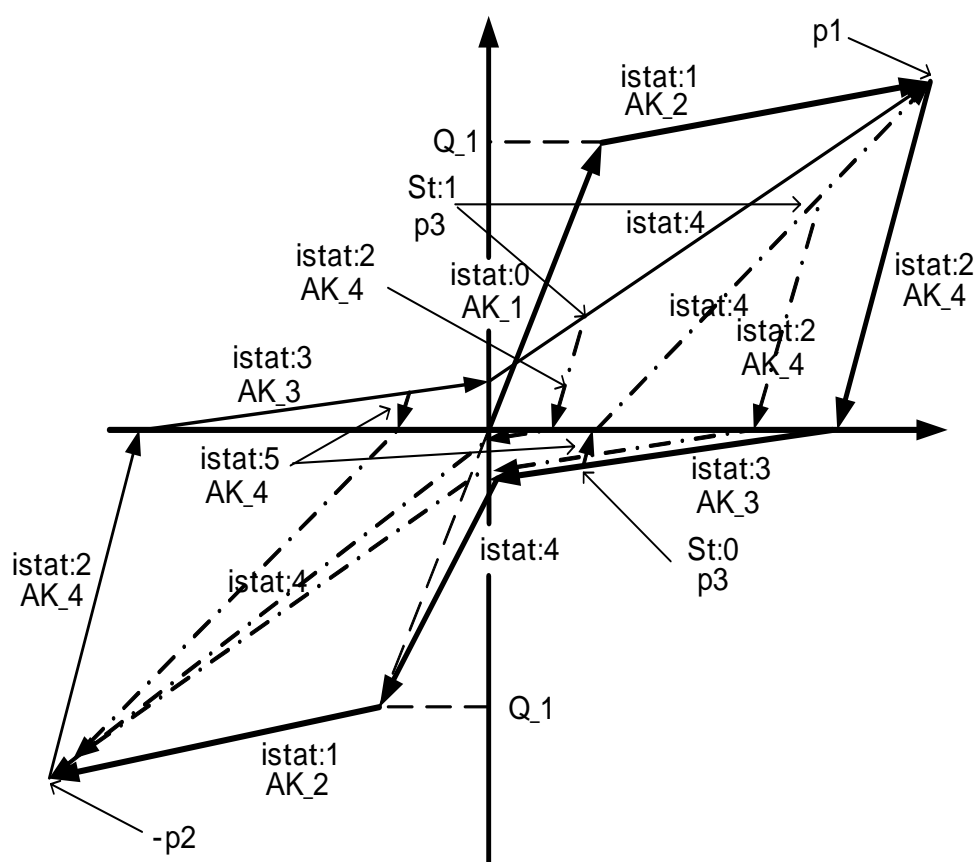


図 4-31 第1折れ点 Q_1 を越えて除荷する履歴

初めの増分変位が正方向に進行し、正方向境界値 Q_1 を越えると状態は 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は 2 となる。除荷時の応力を p_1 として $p_1 > 0$ を境界値として直線的に進行する履歴となる。状態 2 において、増分変位が反転して正方向に進み境界値 p_1 を越えると状態 1 に戻る。また、状態 2 において増分変位が負方向に進み境界値 0 を越えると状態は 3 となり、スリップを表す履歴となる。状態 3 は変位を境界値として、負方向に進行する場合は $u > 0$ の範囲内で直線的に進む。増分変位が反転し正方向に進行する場合は状態が 5 となり、除荷時の状態を $st:0$ 、除荷時の応力を p_3 として境界値 $0 > p_3$ の範囲内で直線履歴を描く。状態 5 以降の履歴については後述する。

状態 3 において境界値の変位 $u = 0$ を越えると状態は 4 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 $-Q_1$ 、負方向の履歴を経験していれば $-p_2$ の範囲内で直線に進行する。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 $(-du_4, -Q_1)$ もしくは $(-du_4, -p_2)$ を指向する勾配を持って進行する。また、状態 4 において増分変位が反転し、正方向に増分変位が進行すると状態は 2 となる。この時、除荷時の状態を $st:1$ 、除荷時の応力を p_3 として状態 2 の境界条件とする。この状態 2 の履歴については後述する。

状態 4 において負方向の境界値 $-Q_1$ もしくは $-p_2$ を越えると状態は 1 に進行する。この場合の状態 1 は負方向の境界条件はなく、正方向に増分変位が反転すると、除荷時の応力を $-p_2$ として、境界値 $0 > -p_2$ の範囲内で直線的に履歴を描く。状態 1 は正方向の場合と同様に負方向の境界値 $-p_2$ を越えると状態は 1 となり、正方向の境界値 0 を越えると状態 3 となってスリップを表す履歴に進む。状態 3 は正方向に進む場合、境界値を変位によって表し、 $0 > u$ の範囲内で直線的に進行する履歴となる。この時、増分変位が反転すると状態は 5 となり、先に述べた場合と同様に境界値を得る。状態 3 において正方向の境界値 $u=0$ を越えると状態は 4 となって、正側の最大点座標 (du_3, p_1) を指向して進行する。

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、除荷したときの応力を p_3 として状態 5 の境界値とする。図 4-31 中、に示した負側の状態 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と、状態 5 の境界値は $0 > p_3$ 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 を越えると状態は 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応力を p_3 として状態 2 に進行し、 $p_3 > 0$

の範囲内で直線的に履歴を描く。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、除荷したときの応力を p_3 として状態 2、5 の境界値とする。図 4-31 中、に示した正側の状態 4 より除荷され、状態 2、5 に進行する履歴について説明すると、図では状態 4 で除荷されると状態 2 に進行しているが、状態 4 から除荷した場合、2 通りの履歴を描く。1 つ目は除荷され応力が 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。

1. 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。除荷した時の応力を p_3 とし、除荷時の状態を $st:1$ とする。状態 2 の境界値は $p_3 > 0$ 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。

2. 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。除荷した時の応力を p_3 とし、除荷時の状態を $st:1$ とする。状態 5 の境界値は $p_3 > 0$ 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 : 0 弾性状態であり、接線剛性は第 1 勾配 AK_1 である。 $Q_1 > 0$ の場合、 Q_1 の中で直線的に動き、 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、接線剛性は AK_2 である。応力が正の場合、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、状態が 2 となり、反転した時の増分前の応力を p_1 にする。

応力が負の場合は、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、状態が2となって反転した時の増分前の応力を p_2 とする。

istat : 2 状態 1 で除荷された時の状態であり、接線剛性は除荷勾配の AK_4 である。応力が正の場合は $p_1 > 0$ の中で直線的に動き、正方向に増分変位が進み、 p_1 を越えると状態は 1 に進む。逆に、負方向に増分変位が進み、0 を越えると状態は 3 に進む。

応力が負の場合は $0 > -p_2$ の中で直線的に動き、負方向に増分変位が進み、 $-p_2$ を越えると状態は 1 に進む、正方向に増分変位が進み、0 を越えると状態は 3 に進む。

istat : 3 スリップ時の勾配であり、接線剛性は 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、応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、0 を越えると状態は 4 となる。増分変位が反転し負方向になった時、状態が 5 となり除荷時の状態を表す st を 0 にして増分前の応力を p_3 とする。

応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、0 を越えると状態は 4 となる。増分変位が反転し正方向になった時、状態は 5 となり除荷時の状態を表す st を 0 にして増分前の応力を p_3 とする。

istat : 4 正負共に最大点を指向する勾配であり、応力が正の場合は増分変位が正方向の時その境界は $p_1 > 0$ 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 にして増分前の応力を p_3 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。

応力が負の場合は増分変位が負方向の時、その境界は $0 > -p_2$ 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 にして増分前の応力を p_3 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、接線剛性は AK_4 である。応力が正の場合は $p_3 > 0$ の中で直線的に動き、正方向に増分変位が進み、 p_3 を越えると st が 0 の時は状態 3 に進行し、 st が 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、0 を越えると状態は 4 に進む。

応力が負の場合は $0 > -p_3$ の中で直線的に動き、負方向に

増分変位が進み、 p_3 を越えると st が0の時は状態3に進行し、
 st が1の時は状態4に進行する。正方向に増分変位が進み0
を越えると状態は4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_BiLiner ( S字型スリップバイリニア履歴モデル )
C
      subroutine S_slip_BiLiner(AK, istat, AK_1, AK_2, AK_3, AK_4, Q_1, du,
*                               P, P1, P2, ug, du3, du4, st)
      implicit real*8(A-H, O-Z)

C-----
c   AK           : 接線剛性
c   istat        : 現在の状態(Work)
c   AK_1         : 第1勾配
c   AK_2         : 第2勾配
c   AK_3         : スリップ時の剛性
c   AK_4         : 除荷剛性
c   Q_1          : 第1折れ点の応力
c   du           : 増分変位
c   P            : 現在の応力(Work)
c   P1           : istat=1 正側における反曲点上端(Work)
c   P2           : istat=1 負側における反曲点上端(Work)
c   P3           : st が1の時の istat=2、5の反曲点上端(Work)
c   st           : 除荷状態を表すパラメーター
C-----
100  continue
C-----istat0
      if (istat.eq.0) then
         p=AK*du+p
         if(du.gt.0) then
            if(p.lt.p1) return
            !1
C
            !2
            !3
            istat=1
            du2=(p-p1)/AK
            AK=AK_2
            p=AK*du2+p1
            return
         else
C-----istat0
            if(p.gt.-p2) return
            !4
C
            !5
            istat=1
            du2=(p+p1)/AK
            AK=AK_2
            p=AK*du2-p2
            return
            endif
C-----istat1

```



```

endif
C                                     istat2(負側)
else                                     !20
p=AK*du+p
if(du.lt.0)then
if(st.eq.0)then
if(p.gt.-p2) return                                     !21
C                                     istat2 から istat=1 へ(負側)
istat=1                                     !22
du2=(p+p1)/AK
AK=AK_2
p=AK*du2-p2
return
else
if(p.gt.p3) return                                     !23
C                                     istat2 から istat=4 へ(負側)
st=0                                     !24
istat=4                                     !25
du2=(p-p3)/AK
AK=AK_5
p=AK*du2+p3
return
endif
else
if(p.lt.0)return                                     !26
C                                     istat2 から istat=3 へ(正側)
istat=3                                     !27
du2=p/AK
AK=AK_3
p=AK*du2
return
endif
endif
C                                     istat3(正側)
elseif(istat.eq.3)then                                     !28
p=AK*du+p                                     !29
u=ug+du
if(p.gt.0)then
if(du.gt.0)then
if(u.lt.0)return                                     !30
istat=4                                     !31
du1=du-u                                     !32
p3=-AK*du1+p                                     !33
AK=(p1-p3)/du3                                     !34
p=AK*u+p3                                     !35
return
else
C                                     istat3 から istat5(除荷)
st=0                                     !36
istat=5                                     !37
dp=AK*du
AK_5=AK                                     !38
AK=AK_4
du=dp/AK

```

```

        p=p-dp
        p3=p
        p=AK*du+p
        return
    endif
    else
C
        istat3 ( 負側 )
        if (du.lt.0) then
        if (u.gt.0) return
C
            istat3 から istat4 ( 負側 )
            istat=4
            du1=du-u
            p3=-AK*du1+p
            AK=(p2+p3)/du4
            p=AK*u+p3
            return
        else
C
            istat3 から istat5(除荷)
            st=0
            istat=5
            dp=AK*du
            AK_5=AK
            AK=AK_4
            du=dp/AK
            p=p-dp
            p3=p
            p=AK*du+p
            return
        endif
    endif
C
        istat4
        elseif (istat.eq.4) then
            p=AK*du+p
            if (p.gt.0) then
            if (du.gt.0) then
            if (p.lt.p1) return
C
                istat4 から istat=1 へ(正側)
                istat=1
                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
                dp=AK*du
                AK_5=AK
                AK=AK_4
                du=dp/AK
                p=p-dp
                p3=p
                p=AK*du+p3
                du5=-p3/AK_4+ug
                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_4
            du=dp/AK
            p=p-dp
            p3=p                                !62
            p=AK*du+p3
            du5=-p3/AK_4+ug                    !63
            if(du5.lt.0) then
                istat=2                        !64
            else
                istat=5                        !65
            endif
            return
        endif
    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
                if(p.lt.p3) return            !67
C                                     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
        endif
    endif
C-----istat5

```

```

      else
      if(p.gt.0)return                                !72
C                                     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
      endif
C-----istat5
      else
      p=AK*du+p
      if(du.lt.0)then
      if(p.gt.p3)return                                !77
C                                     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
C                                     初期設定(正側)
      else                                           !87
      istat=0
      AK=AK_1
      p=0
      p1=Q_1                                       !88
      p2=Q_1                                       !89
      du3=p1/AK_1
      du4=du3
      st=0                                           !90
      goto 100
      end if
      end

```

1. 状態が $istat : 0$ の場合、以下の処理を行う。
2. 増分変位が正の場合 ($du > 0$) で、増分後の応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。
3. 増分後の応力が $p1$ より大きい場合は状態が 1 となり、以下の処理を行う。
4. 増分変位が負の場合 ($du < 0$) で、増分後の応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。
5. 増分後の応力が $-p3$ よりも小さい場合は状態が 1 となり、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. 現在の応力が 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。
8. 現在の応力が 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。
9. 現在の応力が 0 よりも大きく増分変位が負 ($du < 0$) であるか、現在の応力が 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。
10. 状態 1 で除荷したため、現在の応力が $p1$ 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 $du3$ に記憶する。また、増分前の応力を求めて正側の最大応力として $p1$ に記憶する。
11. 状態 1 で除荷したため、現在の応力が $-p2$ 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 $du4$ に記憶する。また、増分前の応力を求めて負側の最大応力として $p2$ に記憶する。
12. 状態が $istat : 2$ の場合、以下の処理を行う。
13. 現在の応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、増分後の応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。
14. 現在の応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、増分後の応力が正方向の境界値 $p1$ より大きければ、状態は 1 となり以下の処理を行う。
15. 現在の応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、増分後の応力が正方向の境界値 $p3$ より小さければ、このサブルーチンより戻る。
16. 現在の応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、増分後の応力が正方向の境界値 $p3$ 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。

17. 状態を 4 にする。
18. 現在の応力が正で増分変位が負 ($du < 0$) の場合で、増分後の応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。
19. 現在の応力が正で増分変位が負 ($du < 0$) の場合で、増分後の応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。
20. 現在の応力が負であるならば以下の処理を行う。
21. 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、増分後の応力が負側の境界値 $-p_2$ より大きければ、このサブルーチンより戻る。
22. 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、増分後の応力が負側の境界値 $-p_2$ より小さければ、状態を 1 として以下の処理を行う。
23. 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、増分後の応力が負側の境界値 $-p_3$ より大きければ、このサブルーチンより戻る。
24. 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、増分後の応力が負側の境界値 $-p_3$ より小さければ、 st を 0 にして以下の処理を行う。
25. 状態を 4 にする。
26. 増分変位が正 ($du > 0$) の場合で、増分後の応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。
27. 増分変位が正 ($du > 0$) の場合で、増分後の応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。
28. 状態が $istat : 3$ の場合、以下の処理を行う。
29. 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。
30. 現在の応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。
31. 現在の応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。
32. 増分変位から増分後の変位を引いて、増分前の変位から変位 0 までの変位量を求め、 du_1 にセットする。
33. 32 で求めた du_1 を用いて変位 0 における応力を求め、 p_3 にセットする。
34. 33 で求めた 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。
35. 求めた接線剛性に増分後の変位を掛けて、33 で求めた p_3 に足し合わせ増分後の応力を計算する。
 36. 現在の応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。
 37. 状態を 5 にする。
 38. 現在の接線剛性 A K を AK_5 にセットする。
 39. 増分前の応力を状態 5 の境界値 p_3 にセットする。
 40. 現在の応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。
 41. 現在の応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。
 42. 増分変位から増分後の変位を引いて、増分前の変位から変位 0 までの変位量を求め、 du_1 にセットする。
 43. 42 で求めた du_1 を用いて変位 0 における応力を求め、 p_3 にセットする。
 44. 43 で求めた 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。
 45. 求めた接線剛性に増分後の変位を掛けて、43 で求めた p_3 に足し合わせ増分後の応力を計算する。
 46. 現在の応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。
 47. 現在の接線剛性 A K を AK_5 にセットする。
 48. 増分前の応力を状態 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. 現在の応力が正で増分変位が正 ($du > 0$) の場合、増分後の応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. 現在の応力が正で増分変位が正 ($du > 0$) の場合、増分後の応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. 現在の応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 AK を AK_5 にセットする。
 54. 増分前の応力を境界値 p_3 にセットする。
 55. p_3 と除荷剛性 AK_4 を用いて、除荷して応力が 0 となる時の残留変位 du_5 を求める。
 56. 55 で求めた 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. 55 で求めた 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. 現在の応力が負で増分変位が負 ($du < 0$) の場合、増分後の応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. 現在の応力が負で増分変位が負 ($du < 0$) の場合、増分後の応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. 現在の応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 AK を AK_5 にセットする。
62. 増分前の応力を境界値 p_3 にセットする。
63. p_3 と除荷剛性 AK_4 を用いて、除荷して応力が 0 となる時の残留変位 du_5 を求める。
64. 63 で求めた 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. 63 で求めた 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. 現在の応力が正で増分変位が正 ($du > 0$) の場合、増分後の応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. 現在の応力が正で増分変位が正 ($du > 0$) の場合、増分後の応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. 現在の応力が正で増分変位が正 ($du > 0$) の場合、増分後の応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 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 $A K$ を記憶しておいた AK_5 にする。
72. 現在の応力が正で増分変位が負 ($du < 0$) の場合、増分後の応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. 現在の応力が正で増分変位が負 ($du < 0$) の場合、増分後の応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 - du_4 と応力が 0 となる点の変位を求め、その間の変位量を計算し du_5 にセットする。
76. 75 で求めた du_5 を用いて応力が 0 になる点から負側の最大点座標

- (- du_4 , - p_2) を指向する接線剛性を計算し A_K にセットする。
77. 現在の応力が負で増分変位が負 ($du < 0$) の場合、増分後の応力が負方向の境界値 p_3 よりも大きければ、このサブルーチンより戻る。
 78. 現在の応力が負で増分変位が負 ($du < 0$) の場合、増分後の応力が負方向の境界値 p_3 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. 現在の応力が負で増分変位が負 ($du < 0$) の場合、増分後の応力が負方向の境界値 p_3 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 AK を記憶しておいた AK_5 にする。
 82. 現在の応力が負で増分変位が正 ($du > 0$) の場合、増分後の応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. 現在の応力が負で増分変位が正 ($du > 0$) の場合、増分後の応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 du_3 と応力が 0 となる点の変位を求め、その間の変位量を計算し du_5 にセットする。
 86. 85 で求めた du_5 を用いて応力が 0 になる点から正側の最大点座標 (du_3, p_1) を指向する接線剛性を計算し A_K にセットする。
 87. 状態が $istat$: - 1 ならば以下の処理を行う。
 88. 正負共に第 1 折れ点の応力をセットする。
 89. 正側の第 1 折れ点の変位を du_3 にセットし、負側の第 1 折れ点の変位を du_4 にセットする。
 90. 除荷状態を表す st をゼロセットする。

4.3.2 S字型スリップ
トリリニア

本節では、接合部に関する履歴ルールで、図 4-32 に示す S 字型の対称スリップトリ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。

第一折れ点の応力	: Q_1	正方向最大変位	: du3
第二折れ点の応力	: Q_2	負方向最大変位	: du4
第一勾配	: AK_1	第二勾配	: AK_2
第三勾配	: AK_3	スリップ勾配	: AK_4
除荷勾配	: AK_5	除荷時の状態	: st

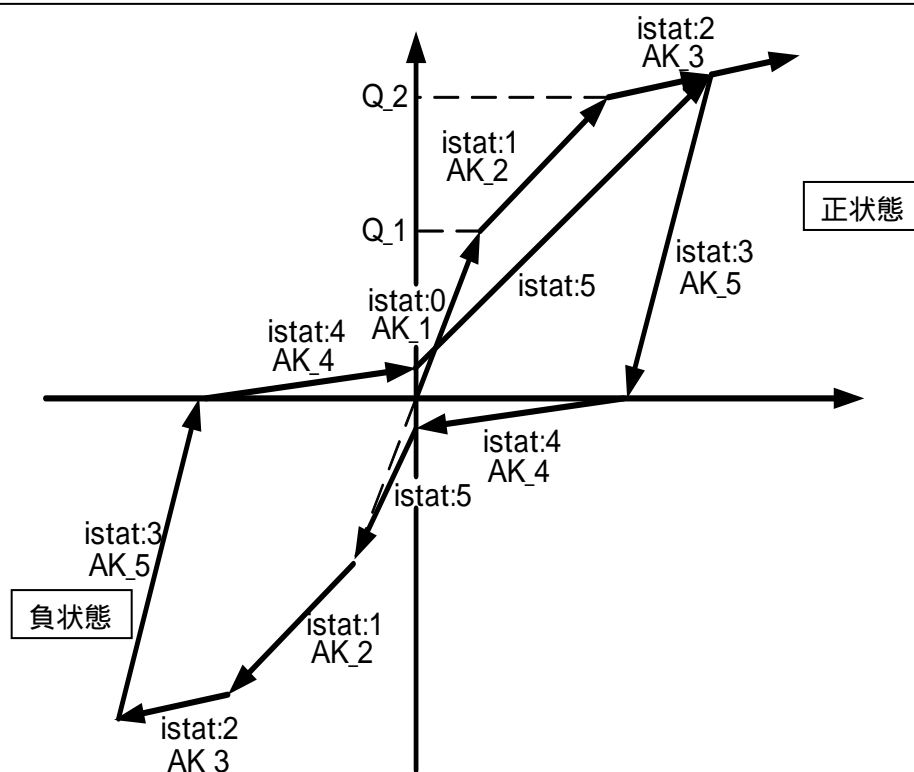


図 4-32 接合部の S 字型スリップトリリニア履歴特性

この履歴特性は木質構造物の接合部の解析を行うために作成した履歴特性であるが、コンクリートのせん断型の履歴としても用いられる事がある。正負対称のトリリニアの骨格曲線であり、除荷時の剛性が初期の剛性と異なる場合においても対応でき、また、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:0 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 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 4-33 を用いて

履歴全体の流れについて説明する。

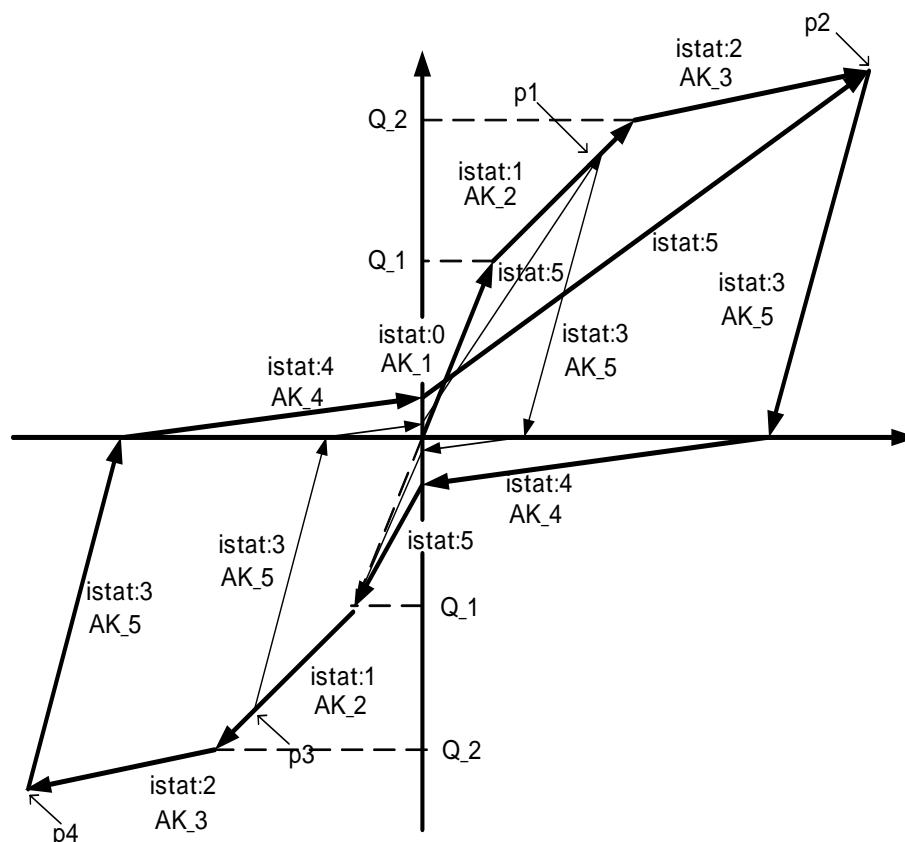


図 4-33 S字型スリプトリニア履歴特性の全体の流れ

図 4-33 の細線で表した履歴は第 1 折れ点を経過して除荷された場合の履歴であり、太線の履歴は第 2 折れ点を経過して除荷された履歴である。まずこの 2 通りの履歴について説明する。どちらの履歴においても初期の状態 0 の履歴は同じであり、状態 0 は境界値 $Q_1 > -Q_1$ 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。

細線で表した履歴について説明する。状態 0 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 $Q_2 > 0$ 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は 3 となり除荷時の応力を p_1 として $p_1 > 0$ 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 p_1 を越えると状態 1 に戻る。また、状態 2 において増分変位が負方向に進み、境界値 0 を越えると状態は 4 となり、スリップを表す状態に入る。

状態 4 は変位を境界値として、負方向に進行する場合は $u(\text{変位}) > 0$

の範囲内で直線的に進む。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。状態4において境界値の変位 $u=0$ を越えると状態は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 $> -Q_1$ 、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 $> -p_3$ 、第2折れ点を経験しておれば負方向の境界値を $> -p_4$ 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 $(-du_4, -Q_1)$ もしくは $(-du_4, -p_3)$ $(-du_4, -p_4)$ を指向する勾配をもって進行する。また、状態5において増分変位が反転し、正方向に増分変位が進行すると状態は3となる。状態3以降の履歴については後述する。

状態5において、負方向の境界値(図4-43の場合は $-Q_1$) 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 $> -Q_2$ 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1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。除荷時の応力を $-p_3$ として、境界値 $0 > -p_3$ の範囲内で直線に履歴を描く。状態3から境界値 $-p_3$ を越えると状態は1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、境界値を変位によって表し $0 > u$ の範囲内で直線的に進行する履歴となる。この時、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。状態4において変位 $u=0$ を超えると状態は5となって、正側の最大点座標 (du_3, p_1) を指向して進行する。

次に太線で表した履歴について説明する。細線と同様に正方向に増分変位が進行し、状態0から状態1に入ったとする。さらに状態1で正方向の境界値 Q_2 を越えると状態2に進行する。状態2は正方向に増分変位が進む場合、その境界はなく、逆に増分変位が反転して負方向に進行すると状態は3となる。この時、除荷したときの応力を p_2 として状態3の境界値とする。したがって、状態3の境界値は $p_2 > > 0$ となり、境界値 p_2 を越えると状態は2に戻り、境界値 0 を越えるとスリップを表す状態4に進行する。状態4は細線で述べた境界条件を持っており、境界値 $u=0$ を越えると状態は5となって、負方向の最大点座標を指向して進行する。状態4で増分変位が反転し正方向に進行すると状態は6となるが、この履歴については後にまとめる。

状態5が負方向に進行し状態1となり状態1の負方向境界値 $-Q_2$ を越えると状態は2となる。状態2において増分変位が反転し正方向に進行すると状態は3となり、除荷時の応力を $-p_4$ として状態3の境界値と

する。状態 3 以降の履歴については細線と同様の履歴となる。

次に、状態 4、5 から除荷した時の履歴について、図 4-33 を用いて説明する。

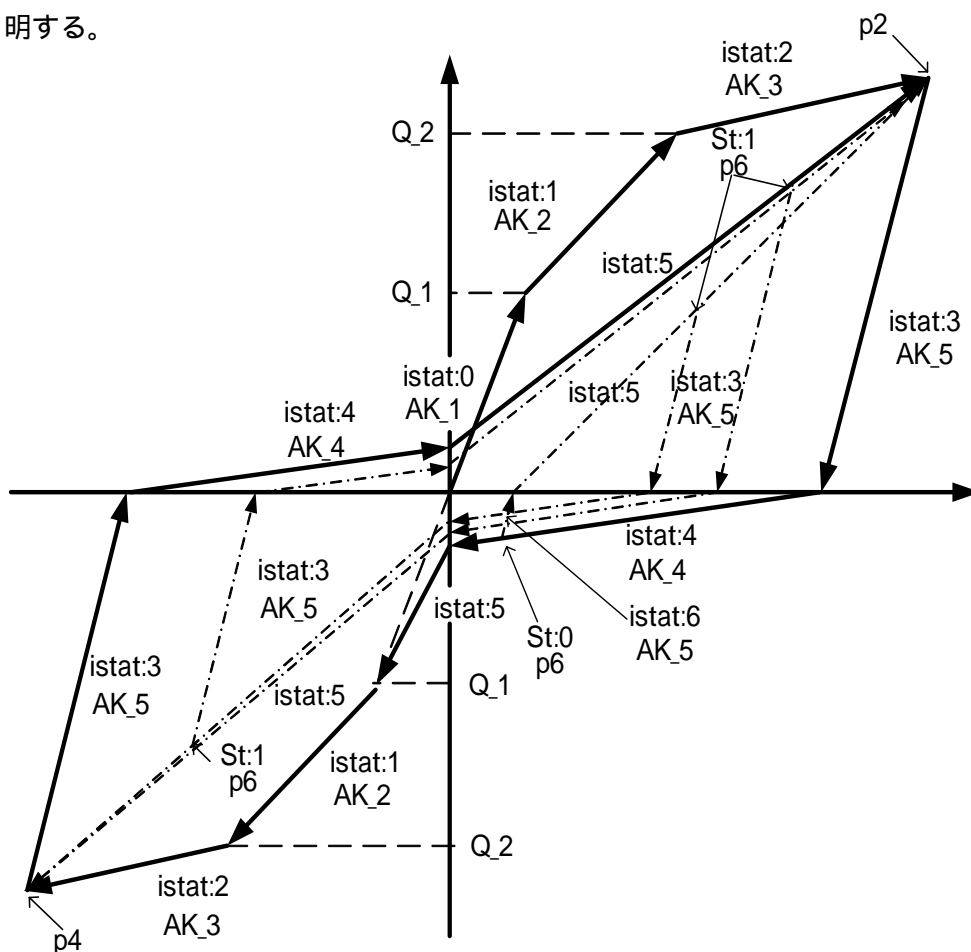


図 4-34 状態 4、状態 5 から除荷した時の履歴

まず初めに、図 4-34 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は 6 になり、除荷時の応力を p_6 、除荷時の状態を $st:0$ とする。状態 6 の境界値は $0 > \gamma > p_6$ 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、除荷時の応力を 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 $p_6 > \gamma > 0$ 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、この後の流れは細線の履歴と同様に描く。

次に、図 4-34 の正状態で状態 4 から状態 5 に進行し、除荷された履

履歴について説明する。この時状態 5 から除荷すると 2 通りの履歴を描かれる。1 つ目は除荷されて応力が 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。除荷した時の応力を p_6 とし、除荷時の状態を $st:1$ とする。状態 3 の境界値は $p_6 > 0$ 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。除荷した時の応力を p_6 とし、除荷時の状態を $st:1$ とする。状態 6 の境界値は $p_6 > 0$ 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、S 字型の対称スリップトリリニア履歴特性は、次の 6 つの状態で表現される。

istat : 0 弾性状態であり、接線剛性は第 1 勾配 AK_1 である。 $Q_1 > 0$ の範囲内で直線的に動き、 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、接線剛性は AK_2 である。応力が正の場合、増分変位が正方向に進行する時 $Q_2 > 0$ 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、状態が 3 となって反転した時の増分前の応力を p_1 にする。

応力が負の場合は、増分変位が負方向に進行する時 $Q_2 < 0$ の範囲内で直線的に動き、除荷されて増分変位が反転し正方向に進行すると、状態が 3 となって反転した時の増分前の応力を p_3 とする。

istat : 2 正負共に第三勾配であり、接線剛性は AK_3 である。応力が正の場合、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、状態が 3 となって反転した時の増分前の応力を p_2 にする。

応力が負の場合は、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、状態が 3 となって反転した時の増分前の応力を p_4 とする。

istat : 3 状態 1、2 で除荷された時の状態であり、接線剛性は除荷勾配の AK_5 である。応力が正の場合は $p_1 > 0$ もしくは $p_2 > 0$ 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 を越えると状態は 1 か 2 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。

応力が負の場合は $0 > -p_3$ もしくは $0 > -p_4$ 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 $-p_4$ を越えると状態は 1 か 2 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 3 に進む。

istat : 4 スリップ時の勾配であり、接線剛性は AK_4 である。状態 4 における境界条件は変位 u によって得られており、応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、0 を越えると状態は 5 となる。増分変位が反転し負方向になった時、状態が 6 となり除荷時の状態を表す st を 0 にして増分前の応力を p_6 とする。

応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、0 を越えると状態は 5 となる。増分変位が反転し正方向になった時、状態は 6 となり除荷時の状態を表す st を 0 にして増分前の応力を p_6 とする。

istat : 5 正負共に最大点を指向する勾配であり、応力が正の場合は増分変位が正方向の時その境界は $p_1 > 0$ もしくは $p_2 > 0$ 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 にして増分前の応力を p_6 とし状態 3、6 に進行する。状態が 3 もしくは 6 に進行する条件については、先に状態 4 から状態 5 に進行する履歴で述べた。

応力が負の場合は増分変位が負方向の時、その境界は $0 > -p_3$ もしくは $0 > -p_4$ 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 にして増分前の応力を p_6 とし状態 3、6 に進行する。状態が 3 もしくは 6 に進行する条件については、先に状態 4 から状態 5 に進行する履歴で述べた。

istat : 6 状態 4、5 から除荷する時の状態であり、接線剛性は AK_4 である。応力が正の場合は $p6 > 0$ の中で直線的に動き、正方向に増分変位が進み、 $p6$ を越えると st が 0 の時は状態 4 に進行し、 st が 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、0 を越えると状態は 5 にむ。

応力が負の場合は $0 > p6$ 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 が 0 の時は状態 4 に進行し、 st が 1 の時は状態 5 に進行する。正方向に増分変位が進み、0 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_TriLinear ( S字型スリプトリニア履歴モデル )
C
      subroutine Smallslip_TriLinear(AK,istat,AK_1,AK_2,AK_3,AK_4,AK_5,
*                                   AK_6,Q_1,Q_2,du,P,P1,P2,P3,P4,P6,
*                                   ug,du3,du4,st)
      implicit real*8(A-H,O-Z)
C
C      AK           : 接線剛性
C      istat        : 現在の状態(Work)
C      AK_1         : 第 1 勾配
C      AK_2         : 第 2 勾配
C      AK_3         : 第三勾配
C      Q_1          : 第 1 折れ点
C      Q_2          : 第 2 折れ点
C      du           : 増分変位
C      P            : 現在の応力(Work)
C      P1           : istat=1 正側における反曲点上端(Work)
C      P2           : istat=2 正側における反曲点上端(Work)
C      P3           : istat=1 負側における反曲点上端(Work)
C      P4           : istat=2 負側における反曲点上端(Work)
C      P6           : st が 1 の時の istat=2,5 の反曲点上端(Work)
C      st           : 除荷状態を表すパラメーター
C
100  continue
C                                     istat0
      if (istat.eq.0)then
      p=AK*du+p
      if(du.gt.0)then
      if(p.lt.p1) return
C                                     istat0 から istat=1 へ ( 正側 )
      istat=1
      du2=(p-p1)/AK
      !1
      !2
      !3

```

	AK=AK_2		
	p=AK*du2+p1		
	return		
	else		
C	if(p.gt.-p3) return	istat0 から istat=1 へ (負側)	!4
	istat=1		!5
	du2=(p+p3)/AK		
	AK=AK_2		
	p=AK*du2-p3		
	return		
	endif		
C		istat1	
	elseif(istat.eq.1) then		!6
	p=AK*du+p		
	if(p.gt.0) then		
	if(du.gt.0) then		
C	if(p.lt.p2) return	istat1 から istat=2 へ (正側)	!7
	istat=2		!8
	du2=(p-p)/AK		
	AK=AK_3		
	p=AK*du2+p2		
	return		
C	else	istat1 から istat=3 へ (除荷)	
	istat=3		!9
	dp=AK*du		
	AK=AK_5		
	du=dp/AK		
	p=p-dp		
	du3=ug		!10
	p1=p		!11
	p=AK*du+p		
	return		
	endif		
	else		
	if(du.lt.0) then		
C	if(p.gt.-p4) return	istat1 から istat=2 へ (負側)	!12
	istat=2		!13
	du2=(p+p4)/AK		
	AK=AK_3		
	p=AK*du2-p4		
	return		
	else		
C		istat1 から istat=3 へ (除荷)	
	istat=3		!14
	dp=AK*du		
	AK=AK_5		
	du=dp/AK		
	p=p-dp		
	du4=-ug		!15
	p3=-p		!16

	p=AK*du+p return endif endif		
C	istat2		
	elseif(istat.eq.2)then		!17
	p=AK*du+p		
	if(p.gt.0)then		
	if(du.gt.0)return		!18
C	istat2 から istat=3 へ(除荷)		
	istat=3		!19
	dp=AK*du		
	AK=AK_5		
	du=dp/AK		
	p=p-dp		
	du3=ug		!20
	p2=p		!21
	p=AK*du+p		
	return		
	else		
	if(du.lt.0)return		!22
C	istat2 から istat=3 へ(除荷)		
	istat=3		!23
	dp=AK*du		
	AK=AK_5		
	du=dp/AK		
	p=p-dp		
	du4=-ug		!24
	p4=-p		!25
	p=AK*du+p		
	return		
	endif		
C	istat3		
	elseif(istat.eq.3)then		!26
	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		!27
C	istat3 から istat=1 へ(正側)		
	istat=1		!28
	du2=(p-p1)/AK		
	AK=AK_2		
	p=AK*du2+p1		
	return		
	else		
	if(p.lt.p2)return		!29
C	istat3 から istat=2 へ(正側)		
	istat=2		!30
	du2=(p-p2)/AK		
	AK=AK_3		
	p=AK*du2+p2		

```

        return
    endif
    else
        if(p.lt.p6)return                                !31
C          istat3 から istat=5 へ(除荷)

        st=0                                             !32
        istat=5                                         !33
        du2=(p-p6)/AK
        AK=AK_6
        p=AK*du2+p6
        return
    endif
    else
        if(p.gt.0)return                                !34
C          istat3 から istat=4 へ(正側)

        istat=4                                         !35
        du2=p/AK
        AK=AK_4
        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                    !36
C          istat3 から istat=1 へ(負側)

                    istat=1                              !37
                    du2=(p+p3)/AK
                    AK=AK_2
                    p=AK*du2-p3
                    return
                else
                    if(p.gt.-p4)return                    !38
C          istat3 から istat=2 へ(負側)

                    istat=2                              !39
                    du2=(p+p4)/AK
                    AK=AK_3
                    p=AK*du2-p4
                    return
                endif
            else
                if(p.gt.p6)return                          !40
C          istat3 から istat=5 へ(除荷)

                st=0                                     !41
                istat=5                                  !42
                du2=(p-p6)/AK
                AK=AK_6
                p=AK*du2+p6
                return
            endif
        else
            if(p.lt.0)return                                !43

```

C	istat3 から istat=4 へ (負側)	
	istat=4	!44
	du2=p/AK	
	AK=AK_4	
	p=AK*du2	
	return	
	endif	
	endif	
C	istat4	
	elseif(istat.eq.4) then	!45
	p=AK*du+p	
	u=ug+du	
	if(p.gt.0) then	
	if(du.gt.0) then	
	if(u.lt.0) return	!46
C	istat4 から istat=5 へ (正側)	
	istat=5	!47
	du2=du+ug	
	p5=-AK*u+p	!48
	if(p2.eq.Q_2) then	
	AK=(p1-p5)/du3	!49
	else	
	AK=(p2-p5)/du3	!50
	endif	
	p=AK*du2+p5	
	return	
	else	
C	istat4 から istat6(除荷)	
	St=0	!51
	istat=6	!52
	dp=AK*du	
	AK_6=AK	!53
	AK=AK_5	
	du=dp/AK	
	p=p-dp	
	p6=p	!54
	p=AK*du+p	
	return	
	endif	
	else	
	if(du.lt.0) then	
	if(u.gt.0) return	!55
C	istat4 から istat=5 へ (負側)	
	istat=5	!56
	du2=du+ug	
	p5=-AK*u+p	!57
	if(p4.eq.Q_2) then	
	AK=(p3+p5)/du4	!58
	else	
	AK=(p4+p5)/du4	!59
	endif	
	p=AK*du2+p5	
	return	
	else	

C	istat4 から istat=6 へ (除荷)	
st=0		!60
istat=6		!61
dp=AK*du		
AK_6=AK		!62
AK=AK_5		
du=dp/AK		
p=p-dp		
p6=p		!63
p=AK*du+p		
return		
endif		
endif		
C	istat5 (正側)	
elseif(istat.eq.5) then		!64
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		!65
C	istat5 から istat=1 へ (正側)	
istat=1		!66
du2=(p-p1)/AK		
AK=AK_2		
p=AK*du2+p1		
return		
else		
if(p.lt.p2) return		!67
C	istat5 から istat=2 へ (正側)	
istat=2		!68
du2=(p-p2)/AK		
AK=AK_3		
p=AK*du2+p2		
return		
endif		
else		
C	istat5 から istat=3,6 へ (除荷)	
st=1		!69
dp=AK*du		
AK_6=AK		!70
AK=AK_5		
du=dp/AK		
p=p-dp		
p6=p		!71
p=AK*du+p6		
du5=-p6/AK_5+ug		!72
if(du5.gt.0) then		
istat=3		!73
else		
istat=6		!74
endif		
return		
endif		
C	istat5 (負側)	

	else		
	if(du.lt.0)then		
	if(p4.eq.Q_2)then		
	if(p.gt.-p3)return		!75
C		istat5 から istat=1 へ(負側)	
	istat=1		!76
	du2=(p+p3)/AK		
	AK=AK_2		
	p=AK*du2-p3		
	return		
	else		
	if(p.gt.-p4)return		!77
C		istat5 から istat=2 へ(負側)	
	istat=2		!78
	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		!79
	dp=AK*du		
	AK_6=AK		!80
	AK=AK_5		
	du=dp/AK		
	p=p-dp		
	p6=p		!81
	p=AK*du+p6		
	du5=-p6/AK_5+ug		!82
	if(du5.lt.0)then		
	istat=3		!83
	else		
	istat=6		!84
	endif		
	return		
	endif		
C		istat6 (正側)	
	elseif(istat.eq.6)then		!85
	if(p.gt.0)then		
	p=AK*du+p		
	u=ug+du		
	if(du.gt.0)then		
	if(p.lt.p6)return		!86
C		istat6 から istat=4/5 へ(正側)	
	if(st.eq.0)then		
	istat=4		!87
	else		
	istat=5		!88
	endif		
	st=0		!89
	du2=(p-p6)/AK		
	AK=AK_6		!90

```

    p=AK*du2+p6
    return
    else
    if(p.gt.0)return                                !91
C          istat6 から istat=5 へ( 除荷 )
    st=0                                            !92
    istat=5                                         !93
    du2=p/AK
    du5=du4+u+du2                                  !94
    if(p4.eq.Q_2)then
    AK=p3/du5                                       !95
    else
    AK=p4/du5                                       !96
    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                                !97
C          istat6 から istat=4/5 へ( 負側 )
    if(st.eq.0)then
    istat=4                                         !98
    else
    istat=5                                         !99
    endif
    st=0                                            !100
    du2=(p-p6)/AK
    AK=AK_6                                         !101
    p=AK*du2+p6
    return
    else
    if(p.lt.0)return                                !102
C          istat6 から istat=5 へ( 除荷 )
    st=0                                            !103
    istat=5                                         !104
    du2=p/AK
    du5=du3-u-du2                                  !105
    if(p2.eq.Q_2)then
    AK=p1/du5                                       !106
    else
    AK=p2/du5                                       !107
    endif
    p=AK*du2
    return
    endif
    endif
C          初期設定
    else
    istat=0                                         !108
    AK=AK_1                                         !109

```

```

p=0
p1=Q_1
p2=Q_2
p3=Q_1
p4=Q_2
du3=p1/AK_1
du4=du3
st=0
goto 100
endif
end

```

1. 状態が $istat : 0$ の場合、以下の処理を行う。
2. 増分変位が正の場合 ($du > 0$) で、増分後の応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。
3. 増分変位が正の場合 ($du > 0$) で、増分後の応力が $p1$ より大きい場合は状態が 1 となり、以下の処理を行う。
4. 増分変位が負の場合 ($du < 0$) で、増分後の応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。
5. 増分変位が負の場合 ($du < 0$) で、増分後の応力が $-p3$ よりも小さい場合は状態が 1 となり、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. 現在の応力が正で、増分変位が正 ($du > 0$) の時、増分後の応力が正方向の境界値 $p2$ より小さければ、このサブルーチンより戻る。
8. 現在の応力が正で、増分変位が正 ($du > 0$) の時、増分後の応力が正方向の境界値 $p2$ より大きければ、状態を 2 にして以下の処理を行う。
9. 現在の応力が正で、増分変位が負 ($du < 0$) の時、状態を 3 にして以下の処理を行う。
10. 状態 1 で除荷したため、最大変位として除荷したときの変位を $du3$ に記憶する。
11. 増分前の応力を求めて正側の最大応力として $p1$ に記憶する。
12. 現在の応力が負で、増分変位が負 ($du < 0$) の時、増分後の応力が負方向の境界値 $-p4$ より大きければ、このサブルーチンより戻る。
13. 現在の応力が負で、増分変位が負 ($du < 0$) の時、増分後の応力が負方向の境界値 $-p4$ より小さければ、状態を 2 にして以下の処理を行う。
14. 現在の応力が負で、増分変位が正 ($du > 0$) の時、状態を 3 にして以

下の処理を行う。

15. 状態 1 で除荷したため、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。
16. 増分前の応力を求めて負側の最大応力として p_3 に記憶する。
17. 状態が $istat : 2$ の場合、以下の処理を行う。
18. 現在の応力が正で増分変位が正 ($du > 0$) ならば、このサブルーチンより戻る。
19. 現在の応力が正で増分変位が負 ($du < 0$) の時、状態を 3 にして以下の処理を行う。
20. 状態 2 で除荷したため、最大変位として除荷したときの変位を du_3 に記憶する。
21. 増分前の応力を求めて正側の最大応力として p_2 に記憶する。
22. 現在の応力が負で、増分変位が負 ($du < 0$) ならば、このサブルーチンより戻る。
23. 現在の応力が負で、増分変位が正 ($du > 0$) の時、状態を 3 にして以下の処理を行う。
24. 状態 2 で除荷したため、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。
25. 増分前の応力を求めて負側の最大応力として p_3 に記憶する。
26. 状態が $istat : 3$ の場合、以下の処理を行う。
27. 現在の応力が正で増分変位が正 ($du > 0$) の時、除荷状態を表す st が 0 で p_2 が第 2 折れ点 Q_2 に等しく現在の応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。
28. 現在の応力が正で増分変位が正 ($du > 0$) の時、除荷状態を表す st が 0 で、 p_2 が第 2 折れ点 Q_2 に等しく現在の応力が正方向の境界値 p_1 より大きければ、状態を 1 にして以下の処理を行う。
29. 現在の応力が正で増分変位が正 ($du > 0$) の時、除荷状態を表す st が 0 で、 p_2 が第 2 折れ点 Q_2 に等しくなく、現在の応力が正方向の境界値 p_2 より小さければ、このサブルーチンより戻る。
30. 現在の応力が正で増分変位が正 ($du > 0$) の時、除荷状態を表す st が 0 で、 p_2 が第 2 折れ点 Q_2 に等しくなく、現在の応力が正方向の境界値 p_2 より大きければ、状態を 2 にして以下の処理を行う。
31. 現在の応力が正で増分変位が正 ($du > 0$) の時、除荷状態を表す st が 1 で、現在の応力が正方向の境界値 p_6 より小さければ、このサブルーチンより戻る。
32. 現在の応力が正で増分変位が正 ($du > 0$) の時、除荷状態を表す st

- が 1 で、現在の応力が正方向の境界値 p_6 より大きければ、除荷状態を表す st を 0 にする。
33. 状態を 5 にする。
 34. 現在の応力が正で増分変位が負 ($du < 0$) の時、増分後の応力が負方向の境界値 0 より大きければ、このサブルーチンより戻る。
 35. 現在の応力が正で増分変位が負 ($du < 0$) の時、増分後の応力が負方向の境界値 0 より小さければ、状態を 4 にして以下の処理を行う。
 36. 現在の応力が負で増分変位が負 ($du > 0$) の時、除荷状態を表す st が 0 で、 p_4 が第 2 折れ点 Q_2 に等しく現在の応力が負方向の境界値 $-p_3$ より大きければ、このサブルーチンより戻る。
 37. 現在の応力が負で増分変位が負 ($du > 0$) の時、除荷状態を表す st が 0 で、 p_4 が第 2 折れ点 Q_2 に等しく現在の応力が負方向の境界値 $-p_3$ より小さければ、状態を 1 にして以下の処理を行う。
 38. 現在の応力が負で増分変位が負 ($du > 0$) の時、除荷状態を表す st が 0 で、 p_4 が第 2 折れ点 Q_2 に等しくなく、現在の応力が負方向の境界値 $-p_4$ より大きければ、このサブルーチンより戻る。
 39. 現在の応力が負で増分変位が負 ($du > 0$) の時、除荷状態を表す st が 0 で p_4 が第 2 折れ点 Q_2 に等しくなく、現在の応力が負方向の境界値 $-p_4$ より小さければ、状態を 2 にして以下の処理を行う。
 40. 現在の応力が負で増分変位が負 ($du > 0$) の時、除荷状態を表す st が 1 で、現在の応力が負方向の境界値 p_6 より大きければ、このサブルーチンより戻る。
 41. 現在の応力が負で増分変位が負 ($du > 0$) の時、除荷状態を表す st が 1 で、現在の応力が負方向の境界値 p_6 より小さければ、除荷状態を表す st を 0 にして以下の処理を行う。
 42. 状態を 5 にする。
 43. 現在の応力が負で増分変位が正 ($du > 0$) の時、増分後の応力が正方向の境界値 0 より小さければ、このサブルーチンより戻る。
 44. 現在の応力が負で増分変位が正 ($du > 0$) の時、増分後の応力が正方向の境界値 0 より大きければ、状態を 4 にして以下の処理を行う。
 45. 状態が $istat : 4$ の場合、以下の処理を行う。
 46. 現在の応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。
 47. 現在の応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 5 にして以下の処理を行う。
 48. 増分後の変位 u に $-AK$ をかけて増分後の応力に足し合わせ、変位が

- 0 となる点の応力 p_5 を求める。
48. p_2 が第2折れ点 Q_2 に等しい時、48 で求めた p_5 を用いて、座標 $(0, p_5)$ 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。
50. p_2 が第2折れ点 Q_2 に等しくない時、48 で求めた p_5 を用いて、座標 $(0, p_5)$ から正方向の最大点座標 (du_3, p_2) を指向する接線剛性を計算し、AK にセットする。
51. 現在の応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。
52. 状態を 6 にする。
53. 現在の接線剛性 $A K$ を AK_6 にセットする。
54. 増分前の応力を状態 6 の境界値 p_6 にセットする。
55. 現在の応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。
56. 現在の応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 5 にして以下の処理を行う。
57. 増分後の変位 u に $-AK$ をかけて増分後の応力に足し合わせ、変位が 0 となる点の応力 p_5 を求める。
58. p_4 が第2折れ点 Q_2 に等しい時、57 で求めた p_5 を用いて、座標 $(0, p_5)$ から正方向の最大点座標 $(-du_4, -p_3)$ を指向する接線剛性を計算し、AK にセットする。
59. p_4 が第2折れ点 Q_2 に等しくない時、57 で求めた p_5 を用いて、座標 $(0, p_5)$ から正方向の最大点座標 $(-du_4, -p_4)$ を指向する接線剛性を計算し、AK にセットする。
60. 現在の応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、以下の処理を行う。
61. 状態を 6 にする。
62. 現在の接線剛性 $A K$ を AK_6 にセットする。
63. 増分前の応力を状態 6 の境界値 p_6 にセットする。
64. 状態が $istat : 5$ の場合、以下の処理を行う。
65. 現在の応力が正で増分変位が正 ($du > 0$) の時、 p_2 が第2折れ点 Q_2 に等しく、増分後の応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。
66. 現在の応力が正で増分変位が正 ($du > 0$) の時、 p_2 が第2折れ点 Q_2 に等しく、増分後の応力が正方向の境界値 p_1 より大きければ、状態を 1 にして以下の処理を行う。

67. 現在の応力が正で増分変位が正 ($du > 0$) の時、 p_2 が第2折れ点 Q_2 に等しくなく、増分後の応力が正方向の境界値 p_2 より小さければこのサブルーチンより戻る。
68. 現在の応力が正で増分変位が正 ($du > 0$) の時、 p_2 が第2折れ点 Q_2 に等しくなく、増分後の応力が正方向の境界値 p_2 より大きければ、状態を2にして以下の処理を行う。
69. 現在の応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。
70. 現在の接線剛性 A_K を AK_6 にセットする。
71. 増分前の応力を状態6の境界値 p_6 にセットする。
72. p_6 と除荷剛性 AK_5 を用いて、除荷して応力が0となる時の残留変位 du_5 を求める。
73. 72で求めた 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3とする。
74. 72で求めた 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6とする。
75. 現在の応力が負で増分変位が負 ($du < 0$) の時、 p_4 が第2折れ点 Q_2 に等しく、増分後の応力が負方向の境界値 $-p_3$ より大きければこのサブルーチンより戻る。
76. 現在の応力が負で増分変位が負 ($du < 0$) の時、 p_4 が第2折れ点 Q_2 に等しく、増分後の応力が負方向の境界値 $-p_3$ より小さければ状態は1となり以下の処理を行う。
77. 現在の応力が負で増分変位が負 ($du < 0$) の時、 p_4 が第2折れ点 Q_2 に等しくなく、増分後の応力が負方向の境界値 $-p_4$ より大きければこのサブルーチンより戻る。
78. 現在の応力が負で増分変位が負 ($du < 0$) の時、 p_4 が第2折れ点 Q_2 に等しくなく、増分後の応力が負方向の境界値 $-p_4$ より小さければ状態は2となり以下の処理を行う。
79. 現在の応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。
80. 現在の接線剛性 AK を AK_6 にセットする。
80. 増分前の応力を境界値 p_6 にセットする。
82. p_6 と除荷剛性 AK_5 を用いて、除荷して応力が0となる時の残留変位 du_5 を求める。
83. 82で求めた 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3とする。

84. 82 で求めた 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 6 とする。
85. 状態が $istat : 6$ の場合、以下の処理を行う。
86. 現在の応力が正で増分変位が正 ($du > 0$) の場合、増分後の応力が正方向の境界値 p_6 よりも小さければ、このサブルーチンより戻る。
87. 現在の応力が正で増分変位が正 ($du > 0$) の場合、増分後の応力が正方向の境界値 p_6 よりも大きく、除荷状態を表す st が 0 の時、状態を 4 にする。
88. 現在の応力が正で増分変位が正 ($du > 0$) の場合、増分後の応力が正方向の境界値 p_6 よりも大きく、除荷状態を表す st が 1 の時、状態を 5 にする。
89. 除荷状態を表す st を 0 にする。
90. 接線剛性 $A K$ を記憶しておいた AK_6 にセットする。
91. 現在の応力が正で増分変位が負 ($du < 0$) の場合、増分後の応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。
92. 現在の応力が正で増分変位が負 ($du < 0$) の場合、増分後の応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。
93. 状態を 5 にする。
94. 負方向の最大変位 - du_4 と応力が 0 となる点の変位を求め、その間の変位量を計算し du_5 にセットする。
95. p_4 が第 2 折れ点 Q_2 に等しい時、94 で求めた du_5 を用いて応力が 0 になる点から負側の最大点座標 (- du_4 , - p_3) を指向する接線剛性を計算し $A K$ にセットする。
96. p_4 が第 2 折れ点 Q_2 に等しくない時、94 で求めた du_5 を用いて応力が 0 になる点から負側の最大点座標 (- du_4 , - p_4) を指向する接線剛性を計算し $A K$ にセットする。
97. 現在の応力が負で増分変位が負 ($du < 0$) の場合、増分後の応力が負方向の境界値 p_6 よりも大きければ、このサブルーチンより戻る。
98. 現在の応力が負で増分変位が負 ($du < 0$) の場合、増分後の応力が負方向の境界値 p_6 よりも小さく、除荷状態を表す st が 0 の時、状態を 4 にする。
99. 現在の応力が負で増分変位が負 ($du < 0$) の場合、増分後の応力が負方向の境界値 p_6 よりも小さく、除荷状態を表す st が 1 の時、状態を 5 にする。
100. 除荷状態を表す st を 0 にする。
101. 接線剛性 AK を記憶しておいた AK_6 にする。

102. 現在の応力が負で増分変位が正 ($du > 0$) の場合、増分後の応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。
103. 現在の応力が負で増分変位が正 ($du > 0$) の場合、増分後の応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。
104. 状態を 5 にする。
105. 正方向の最大変位 $du3$ と応力が 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。
106. が第 2 折れ点 Q_2 に等しい時、105 で求めた $du5$ を用いて応力が 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 $A K$ にセットする。
107. $p2$ が第 2 折れ点 Q_2 に等しくない時、105 で求めた $du5$ を用いて応力が 0 になる点から正側の最大点座標 ($du3, p2$) を指向する接線剛性を計算し $A K$ にセットする。
108. 状態が $istat : -1$ の場合、以下の初期設定を行う。
109. 状態を $istat : 0$ にセットする。
110. $p1$ と $p3$ に Q_1 をセットし、 $p2$ と $p4$ に Q_2 をセットする。
111. 正側の第 1 折れ点の変位を $du3$ にセットし、負側の第 1 折れ点の変位を $du4$ にセットする。
112. 除荷状態を表す st を 0 セットする。

4.3.3 スリップバイ
リニア

本節では、接合部に関する履歴ルールで、図 4-35 に示す対称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。

第一折れ点の応力	: Q_1		
正方向残留変位	: $du3$	負方向残留変位	: $du4$
第一勾配	: AK_1	第二勾配	: AK_2
スリップ勾配	: $AK_1 \times 0.000000001$		

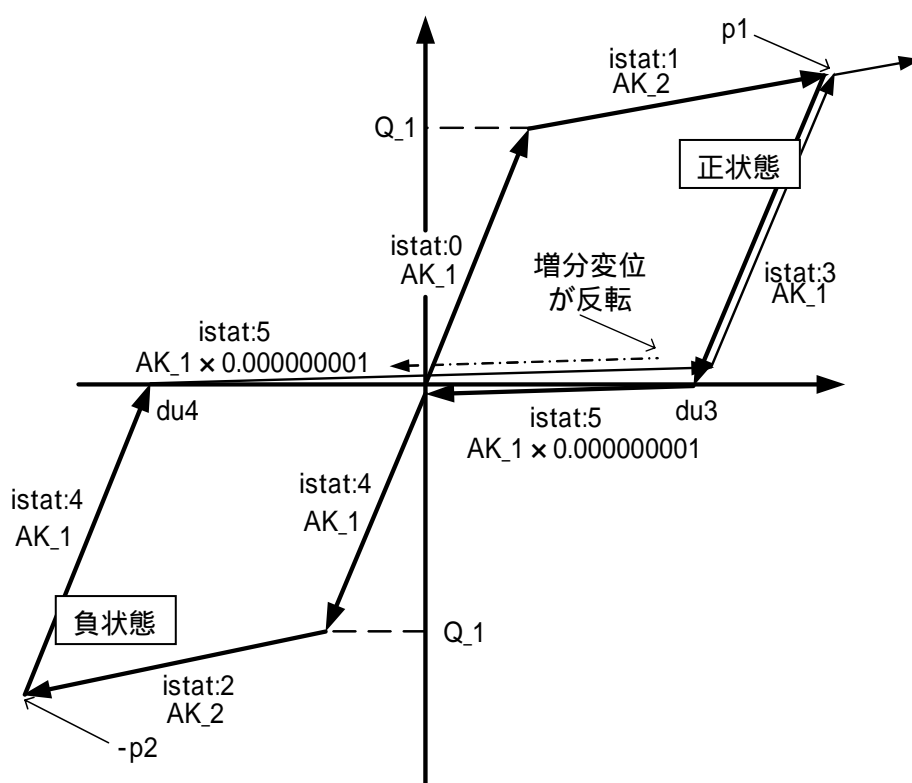


図 4-35 接合部のスリップバイリニア履歴特性

この履歴特性は木質構造物の接合部の解析を行うために作成した履歴特性であるが、X型のブレースの復元力特性としても用いられる事がある。正負対称のバイリニアの骨格曲線であり、除荷時の剛性は初期の剛性と同じにしており、スリップ時の剛性は極力 0 に近い値として初期の剛性に 0.000000001 倍する事とした。弾性状態は、istat:0 であり、istat:3, 4 は除荷された時の弾性状態を表す。降伏以降の状態は、istat:1, 2 で表しており、istat:5 は残留変位によるスリップ状態を表す。

ここで、図 4-35 を用いてこの履歴について説明する。同図は、変位

が正方向に進み Q_1 を越えて履歴を描く場合を示す。初期では状態 0 であり、この状態の境界値 $Q_1 > -Q_1$ の範囲内では変位が反転しても直線的に履歴を描く。この履歴特性は正負対称の履歴であり、ここでは正方向の境界値 Q_1 を越えて進行する履歴について説明する。

初めの増分変位が正方向に進行し、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無く、増分変位が反転し負方向に進むと状態は 3 となり、除荷時の応力を p_1 として $p_1 > 0$ 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 p_1 を越えると状態 1 に戻る。また、状態 3 において増分変位が負方向に進み境界値 0 を越えると、応力が 0 となった点の変位を du_3 として状態は 5 となり、スリップを表す履歴となる。

状態 5 では極力剛性が 0 に近い値として第 1 剛性 AK_1 に 0.000000001 倍した値を剛性としており、状態 5 を進行している間、応力は 0 であると仮定する。状態 5 は変位 u を境界値としており、 $du_3 > u > du_4$ の範囲内で直線に動く履歴となっている。増分変位が反転して正方向に進行し、正方向の境界値 $u=du_3$ を越えると状態は 3 に戻る。また、増分変位が負方向に進行し境界値 $u=du_4$ を越えると負側の状態 4 に進行する。

1 度も負側の履歴を経験していない場合、 du_4 は 0 となっており、変位が 0 を越えると状態 4 に進行する。状態 4 では境界値 $0 > -p_2$ の範囲内で直線的に進行する履歴となっており、1 度も負側の履歴を経験していなければ、境界値は $0 > -Q_1$ となる。状態 4 で増分変位が反転し正方向に進行して境界値 0 を越えると状態は 5 に戻り、応力が 0 となった点の変位を du_4 にセットする。また、状態 4 において増分変位が負方向に進行し、境界値 $-p_2$ を越えると状態は 2 となる。状態 2 は負方向には上限がなく、正方向に進む場合除荷したときの応力を $-p_2$ として状態 4 にセットする。

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 : 0 | 弾性状態であり、接線剛性は第 1 勾配 AK_1 である。 $Q_1 > -Q_1$ の中で直線的に動き、 Q_1 、 $-Q_1$ を越えると状態 1 となる。 |
| istat : 1 | 正側の第 2 勾配であり、接線剛性は AK_2 である。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、状態が 3 となって反転した時の増分前の応力を p_1 にする。 |

- istat : 2 負側の第2勾配であり、接線剛性はAK_2である。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、状態が4となって反転した時の増分前の応力を - p2 にする。
- istat : 3 正側の除荷勾配であり、スリップ後の弾性域となる。接線剛性はAK_1であり、境界条件は $p1 > 0$ となる。増分変位が正方向に進行し、境界値 p1 を越えると状態は1となり、逆に、増分変位が負方向に進行し、境界値 0 を越えると状態は5に進行する。この時、応力が0となった時の変位を正側の残留変位 du3 とする。
- istat : 4 負側の除荷勾配であり、スリップ後の弾性域となる。接線剛性はAK_1であり、境界条件は $0 > -p2$ となる。増分変位が負方向に進行し、境界値 - p2 を越えると状態は2となり、逆に、増分変位が正方向に進行し、境界値 0 を越えると状態は5に進行する。この時、応力が0となった時の変位を正側の残留変位 du4 とする。
- istat : 5 残留変位間のスリップを表す状態であり、境界条件は変位 u によって表され $du3 > u > du4$ となる。増分変位が正方向に進行し、境界値 du3 を越えると正側の弾性域である状態3に進行し、逆に、増分変位が負方向に進行し、境界値 du4 を越えると負側の弾性域である状態4に進行する。

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 / Slip_BiLinear_f (スリップバイリニア型)
C
      subroutine Slip_BiLinear_f(AK,istat,AK_1,AK_2,Q_1,du,P,P1,P2,ug,
*                               du3,du4)
      implicit real*8(A-H,O-Z)

C
C      AK           :接線剛性
C      istat        :現在の状態(Work)
C      AK_1         :第1勾配
C      AK_2         :第2勾配
C      Q_1          :第1折れ点のせん断力
C      du           :増分変位
C      P            :現在のせん断力(Work)
C      P1           :istat=3における反曲点上端(Work)
C      P2           :istat=2における反曲点上端(Work)

```

```

C
100 continue
    if(istat.eq.0) then                                !1
        p=AK*du+p
        if(du.gt.0.) then
            if(p.lt.p1) return                        !2
        c                                             istat = 0 から istat=1 へ (正側)
        istat=1                                       !3
        du2= (p - p1)/AK
        AK=AK_2
        p = AK*du2+p1
        return
        else
            if(p.gt.-p2) return                        !4
        c                                             istat = 0 から istat=2 へ (負側)
        istat=2                                       !5
        du2= (p+p2)/AK
        AK=AK_2
        p =AK*du2-p2
        return
        endif
    c                                             istat =1 (正側)
    elseif(istat.eq.1) then                            !6
        p = AK*du + p
        if(du.ge.0.) return                            !7
    c                                             istat =1 から istat=3 へ (除荷)
    istat=3                                           !8
    dp = AK*du
    AK=AK_1
    du=dp/AK
    p=p-dp
    p1 = p                                           !9
    p = AK*du + p
    return
    c                                             istat =2 (負側)
    elseif(istat.eq.2) then                            !10
        p = AK*du + p
        if(du.le.0.) return                            !11
    c                                             istat =2 から istat=4 へ (除荷)
    istat=4                                           !12
    dp = AK*du
    p=p-dp
    AK=AK_1
    du=dp/AK
    p2=-p                                           !13
    p = AK*du + p
    return
    c                                             istat =3 (正側)
    elseif(istat.eq.3) then                            !14
        p = AK*du + p
        if(du.gt.0.) then
            if(p.lt.p1) return                        !15
        c                                             istat =3 から istat=1 へ (正側)
        istat=1                                       !16

```

```

    du2= (p - p1)/AK
    AK=AK_2
    p = AK*du2+p1
    return
  else
    if(p.gt.0)return !17
c      istat =3 から istat=5 へ (負側) !18
    istat=5
    du2= p/AK
    du3=du-du2+ug !19
    AK=AK_1*0.000000001 !20
    p=0 !21
    return
  endif
c      istat =4 (負側) !22
  elseif(istat.eq.4) then !23
    p = AK*du + p
    if(du.lt.0.) then !24
    if(p.gt.-p2)return !25
c      istat =4 から istat=2 へ (負側) !26
    istat=2
    du2= (p+p2)/AK
    AK=AK_2
    p=AK*du2-p2
    return
  else
    if(p.lt.0)return !27
c      istat =4 から istat=5 へ (正側) !28
    istat=5
    du2= p/AK
    du4=du-du2+ug !29
    AK=AK_1*0.000000001 !30
    p=0 !31
    return
  endif
c      istat =5 (スリップ) !32
  elseif(istat.eq.5) then !33
    u=ug+du !34
    if(du.gt.0.) then
    if(u.lt.du3)return
c      istat =5 から istat=3 へ (正側) !35
    istat=3
    du2=u-du3
    AK=AK_1
    p=AK*du2
    return
  else
    if(u.gt.du4)return !36
c      istat =5 から istat=4 へ (負側) !37
    istat=4
    du2=u-du4
    AK=AK_1
    p=AK*du2
    return
  endif

```

```

      endif
      else
c          初期設定
      istat = 0                                !35
      AK=AK_1
      p1=Q_1                                  !36
      p2=p1
      p=0.
      du3=0                                    !37
      du4=0
      goto 100
      endif
      return
      end

```

1. 状態が $istat : 0$ の場合、以下の処理を行う。
2. 増分変位が正の場合 ($du > 0$) で、増分後の応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。
3. 増分後の応力が $p1$ より大きい場合は状態が 1 となり、以下の処理を行う。
4. 増分変位が負の場合 ($du < 0$) で、増分後の応力が負方向の境界値 $-p2$ よりも大きければ、このサブルーチンより戻る。
5. 増分後の応力が $-p2$ よりも小さい場合は状態が 2 となり、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. 増分変位が正 ($du > 0$) ならば、このサブルーチンより戻る。
8. 増分変位が負 ($du < 0$) ならば、状態を 3 にし以下の処理を行う。
9. 増分前の応力を求めて $p1$ にセットする。
10. 状態が $istat : 1$ の場合、以下の処理を行う。
11. 増分変位が負 ($du < 0$) ならば、このサブルーチンより戻る。
12. 増分変位が正 ($du > 0$) ならば、状態を 4 にし以下の処理を行う。
13. 増分前の応力を求めて $-p2$ にセットする。
14. 状態が 3 の場合、以下の処理を行う。
15. 増分変位が正の場合 ($du > 0$) で、増分後の応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。
16. 増分後の応力が $p1$ より大きい場合は状態が 1 となり、以下の処理を行う。
17. 増分変位が負の場合 ($du < 0$) で、増分後の応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。
18. 増分変位が負の場合 ($du < 0$) で、増分後の応力が負方向の境界値 0

- よりも小さければ、状態は5とし以下の処理を行う。
19. 応力が0となる点の変位を求め、 du_3 にセットする。
 20. 接線剛性 AK を極力0に近い値とし、第1勾配 AK_1 に 0.000000001 倍して求める。
 21. 増分後の応力を0とする。
 22. 状態が $istat : 4$ の場合、以下の処理を行う。
 23. 増分変位が負の場合 ($du < 0$) で、増分後の応力が負方向の境界値 $-p_2$ よりも大きければ、このサブルーチンより戻る。
 24. 増分変位が負の場合 ($du < 0$) で、増分後の応力が負方向の境界値 $-p_2$ よりも小さければ、状態は2とし以下の処理を行う。
 25. 増分変位が正の場合 ($du > 0$) で、増分後の応力が正方向の境界値0より小さければ、このサブルーチンより戻る。
 26. 増分後の応力が0より大きい場合は状態が5となり、以下の処理を行う。
 27. 応力が0となる点の変位を求め、 du_4 にセットする。また接線剛性 AK を第1勾配 AK_1 に 0.000000001 倍して求める。
 28. 増分後の応力を0とする。
 29. 状態が $istat : 5$ の場合、以下の処理を行う。
 30. 前ステップまでの総変位 u_g に増分変位 du を足し合わせて、増分後の変位 u を求める。
 31. 増分変位が正の場合 ($du > 0$) で、変位 u による正方向境界値 du_3 より小さければ、このサブルーチンより戻る。
 32. 増分変位が正の場合 ($du > 0$) で、変位 u による正方向境界値 du_3 より大きければ、状態を3とし以下の処理を行う。
 33. 増分変位が負の場合 ($du < 0$) で、変位 u による負方向境界値 du_4 より大きければ、このサブルーチンより戻る。
 34. 増分変位が負の場合 ($du < 0$) で、変位 u による負方向境界値 du_4 より小さければ、状態を4とし以下の処理を行う。
 35. 状態が -1 であるならば、初期設定として状態0とし、以下の処理を行う。
 36. p_1 と p_2 に第1折れ点 Q_1 をセットする。
 37. 残留変位 du_3 と du_4 をゼロセットする。

4.3.4 スリップトリリニア

本節では、接合部に関する履歴ルールで、図 4-36 に示す対称スリップトリ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折れ点の応力	: Q_1	正方向残留変位	: du3
第2折れ点の応力	: Q_2	負方向残留変位	: du4
第1勾配	: AK_1	第2勾配	: AK_2
第3勾配	: AK_3	スリップ勾配	: AK_1 × 0.000000001

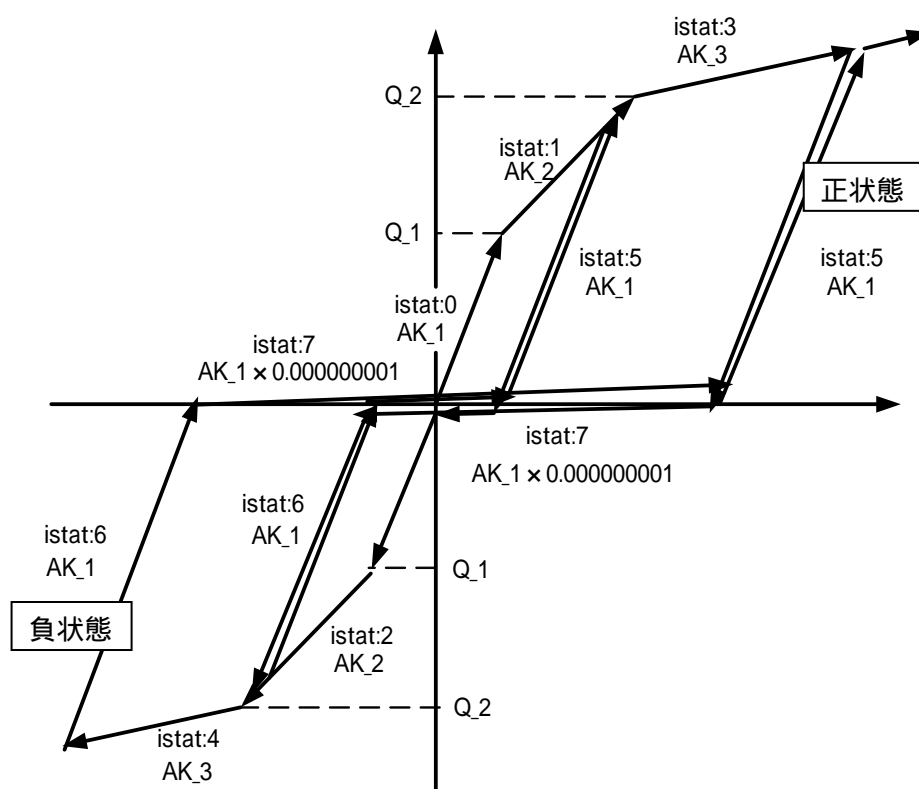


図 4-36 接合部のスリップトリリニア履歴特性

この履歴特性は木質構造物の接合部の解析を行うために作成した履歴特性であるが、多くは X 型のブレースの復元力特性としても用いられる。骨格曲線は正負対称のトリリニアであり、除荷時の剛性が初期の剛性と同じである。スリップ時の剛性は極力 0 に近い値とし、第 1 勾配の 0.000000001 倍とした。弾性状態の指標は istat:0 であり、また、istat:5、6 は除荷時の弾性状態を表す。

降伏以降の状態は、istat:1 から istat:4 で表し、istat:7 は残留変位によるスリップ状態を表す。

ここで、図 4-37 を用いてこの履歴について説明する。

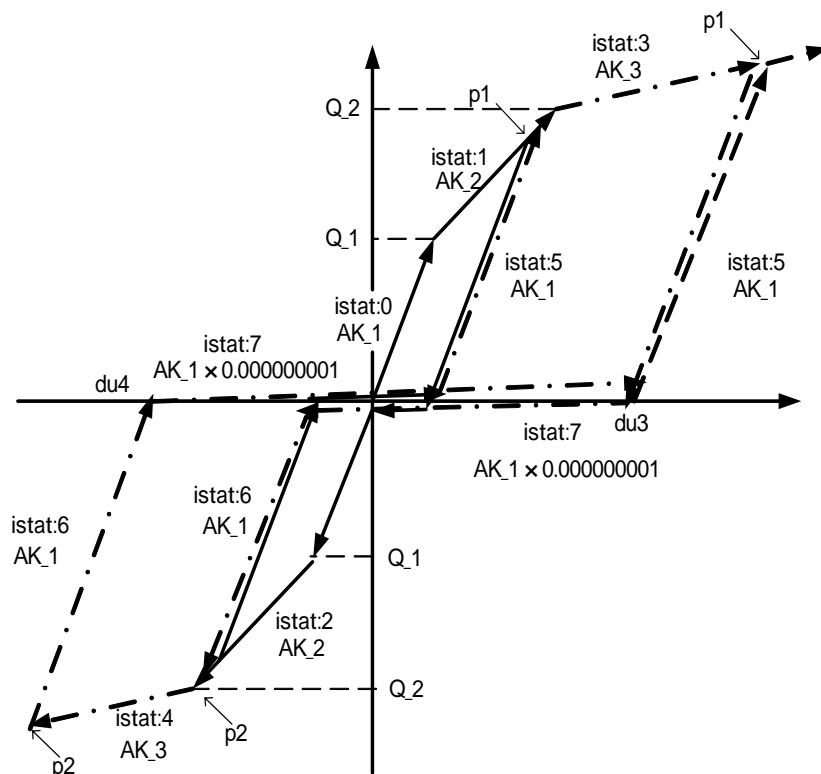


図 4-37 スリップトリリニア履歴特性の全体の流れ

図 4-37 ではスリップトリリニア履歴特性の流れを 2 通りに分けて示す。この 2 通りの履歴を、第一折れ点を経験して履歴を描く場合と第二折れ点を経験して履歴を描く場合に分けて説明する。

第 1 折れ点を経験して履歴を描く場合

初期の状態 0 では、境界値 $1 > -Q_1$ の範囲内で増分変位が反転したとしても直線的に履歴を描く。増分変位が正方向に進行し正方向境界値 Q_1 を越えて進行すると状態は 1 に進む。反対に状態 0 で負方向に進行し負方向境界値 $-Q_1$ を越えると状態は 2 に変化する。

まず、状態 1 に進行する履歴について説明する。状態 1 は正方向に進む場合境界値は $Q_2 >$ であり、正方向境界値 Q_2 を越えると状態 3 となる。状態 3 以降の履歴についてはで説明する。状態 2 において除荷され増分変位が反転して負になると状態は 5 となる。除荷された時の応力を $p1$ とし状態 5 の境界値とする。状態 5 の境界値は $p1 > 0$ の範囲内で直線的に進行する履歴となる。状態 5 で増分変位が反転して正方向に進行し、正方向境界値 $p1$ を越え

ると状態は1に戻る。また、状態5において増分変位が負方向に進行し、境界値0を越えると応力が0となる点の変位を du_3 にセットし、スリップを表す状態7に入る。状態7は変位 u を境界値として、 $du_3 > u > du_4$ の範囲内で直線に進行し、接線剛性は AK_1 に 0.000000001 倍した値とする。状態7で増分変位が反転して正方向に進行し、境界値 $u=du_3$ を越えると状態5に戻る。また、状態7において増分変位が負方向に進行し、境界値 $u=du_4$ を越えると状態6に進行し、負側の弾性状態となる。

状態6の境界条件は $0 > \sigma > -Q_1$ であり、増分変位が反転して正方向境界値0を越えるとスリップを表す状態7に戻る。また、状態6で負方向に進行し境界値 $-Q_1$ を越えると、状態は2に進行する。状態2は負方向に進行する場合、境界値は $-Q_2$ であり $-Q_2$ を越えると状態は4になる。状態4以降の履歴については 3.2 で説明する。状態2で増分変位が反転し正方向に進行すると、除荷したときの応力を $-p_2$ として状態6に入る。この時の状態6の境界値は $0 > \sigma > -p_2$ であり、正方向に進行し正方向境界値0を越えると状態7に入り、増分変位が反転し負方向の境界値 $-p_2$ を越えると状態2に戻る。

第2 折れ点を経験して履歴を描く場合

状態0から状態1に進行し、正方向の境界値 Q_2 を越えると状態3に入る。状態3は正方向に進む場合上限はなく、増分変位が反転して負方向に進行すると、除荷したときの応力を p_1 として状態5に進む。状態5では境界値を $p_1 > \sigma > 0$ とし、この範囲内で直線に動く。状態5で増分変位が反転して正方向に進行し、正方向境界値 p_1 を越えると状態は3に戻る。反転せずに負方向に進行し、負方向の境界値0を越えると、応力が0となる点の変位を du_3 にセットし、状態7に変化する。

状態7の接線剛性は AK_1 の 0.000000001 倍とし、状態7の境界値は変位 u として $du_3 > u > du_4$ の範囲内で直線的に動く。状態7は増分変位が反転して正方向に進行し、正方向の境界値 $u=du_3$ を越えると状態5に戻る。反転せずに負方向に進行し、負方向の境界値 du_4 を越えると、状態6に進行する履歴となる。この時の状態6では、 $0 > \sigma > -Q_1$ の範囲内で直線的に動き、増分変位が反転して正方向に進み0を越えると状態7に戻り、増分変位が反転せずに負方向に進んで $-Q_1$ を越えると状態2に進行する。状態2の履歴は 3.2 で説明している。状態2から状態4に進行したとすると、状態4は負方

向に進行する時上限はなく、増分変位が反転して正方向に進行すると状態は6に入り、除荷した時の応力を $-p_2$ にセットする。状態6は で説明した履歴と同じである。

以上の説明をまとめると、対称スリップトリリニア履歴特性は、次の7つの状態で表現される。

- istat : 0 弾性状態であり、接線剛性は第1勾配 AK_1 である。 $Q_1 > 0$ - Q_1 の中で直線的に動き、 Q_1 を越えると状態1に、 $-Q_1$ を越えると状態2となる。
- istat : 1 正側の第2勾配上であり、接線剛性は AK_2 である。増分変位が正方向に進行する時 $Q_2 > 0$ 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、状態が5となって反転した時の応力を p_1 にする。
- istat : 2 負側の第2勾配上であり、接線剛性は AK_2 である。増分変位が負方向に進行する時 $-Q_2 > 0$ の範囲内で直線的に動き、除荷されて増分変位が反転し正方向に進行すると、状態が6となって反転した時の応力を $-p_2$ とする。
- istat : 3 正側の第3勾配であり、接線剛性は AK_3 である。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、状態が5となって反転した時の応力を p_1 にする。
- istat : 4 負側の第3勾配であり、接線剛性は AK_3 である。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、状態が6となって反転した時の応力を $-p_2$ とする。
- istat : 5 正側の状態1、3で除荷された時の状態であり、接線剛性は第1勾配と同じ AK_1 である。 $p_1 > 0$ の中で直線的に動き、正方向に増分変位が進み p_1 を越えると状態は1か3に進む。また、負方向に増分変位が進み0を越えると応力が0となる点の変位を du_3 にセットして状態は7に進む。
- istat : 6 負側の状態2、4で除荷された時の状態であり、接線剛性は第1勾配と同じ AK_1 である。 $0 > -p_2$ の中で直線的に動き、負方向に増分変位が進み、 $-p_2$ を越えると状態は2か4に進む。また、正方向に増分変位が進み0を越えると応力が0となる点の変位を du_4 にして状態は7に進む。
- istat : 7 スリップ時の勾配であり、接線剛性は AK_1 に 0.000000001 倍

した値である。状態7における境界条件は変位 u によって得られており、その境界条件は $du_3 > u > du_4$ である。増分変位が正方向の時、 du_3 を越えると状態は5となる。また、増分変位が負方向の時、境界値 du_4 を越えると状態6となる。

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 /Slip_Trilinear_f (スリップ対称トリリニア型)
C
C      subroutine Slip_Trilinear_f(AK,istat,AK_1,AK_2,AK_3,Q_1,Q_2,du,P,
*      P1,P2,ug,du3,du4)
C      implicit real*8(A-H,O-Z)
C
C      AK          :接線剛性
C      istat       :現在の状態(Work)
C      AK_1        :第一勾配
C      AK_2        :第二勾配
C      Q_1         :第一折れ点のせん断力
C      du          :増分変位
C      P           :現在のせん断力(Work)
C      P1          :istat=3における反曲点上端(Work)
C      P2          :istat=2における反曲点上端(Work)
C
100 continue
   if(istat.eq.0) then                                !1
      p=AK*du+p
      if(du.gt.0.) then
         if(p.lt.p1) return                            !2
C                                     istat = 0 から istat=1 へ (正側)
         istat=1                                       !3
         du2= (p-p1)/AK
         AK=AK_2
         p=AK*du2+p1
         return
      else
         if(p.gt.-p2) return                            !4
C                                     istat = 0 から istat=2 へ (負側)
         istat=2                                       !5
         du2=(p+p2)/AK
         AK=AK_2
         p=AK*du2-p2
         return
      endif
C                                     istat =1 (正側)
   elseif(istat.eq.1) then                            !6
      p=AK*du+p
      if(du.gt.0.) then
         if(p.lt.Q_2) return                            !7

```

```

c                                istat =1 から istat=3 へ (正側)
    istat=3                                !8
    du2= (p-Q_2)/AK
    AK=AK_3
    p=AK*du2+Q_2
    return
else
c                                istat=1 から istat=5 へ (除荷)
    istat=5                                !9
    dp=AK*du
    AK=AK_1
    du=dp/AK
    p=p-dp
    p1=p                                !10
    p=AK*du+p
    return
endif
c                                istat =2 (負側)
elseif(istat.eq.2) then                                !11
    p = AK*du + p
    if(du.lt.0.) then
        if(p.gt.-Q_2) return                                !12
c                                istat = 2 から istat=4 へ (負側)
        istat=4                                !13
        du2=(p+Q_2)/AK
        AK=AK_3
        p=AK*du2-Q_2
        return
    else
c                                istat =2 から istat=6 へ (負側)
        istat=6                                !14
        dp=AK*du
        p=p-dp
        AK=AK_1
        du=dp/AK
        p2=-p                                !15
        p=AK*du+p
        return
    endif
c                                istat =3 (正側)
elseif(istat.eq.3) then                                !16
    p=AK*du+p
    if(du.ge.0.) return                                !17
c                                istat =3 から istat=5 へ (除荷)
    istat=5                                !18
    dp=AK*du
    AK=AK_1
    du=dp/AK
    p=p-dp
    p1=p                                !19
    p=AK*du + p
    return
c                                istat =4 (負側)
elseif(istat.eq.4) then                                !20

```

```

    p=AK*du+p
    if(du.le.0.) return !21
c          istat =4 から istat=6 へ ( 除荷 )
    istat=6 !22
    dp=AK*du
    p=p-dp
    AK=AK_1
    du=dp/AK
    p2=-p !23
    p=AK*du+p
    return
c          istat =5 ( 正側 )
    elseif(istat.eq.5) then !24
    p=AK*du+p
    if(du.gt.0.) then
    if(p.lt.p1)return !25
c          istat =5 から istat=1or3 へ ( 正側 )
    du2=(p-p1)/AK
    if(p1.lt.Q_2)then !26
    istat=1
    AK=AK_2
    else
    istat=3 !27
    AK=AK_3
    endif
    p=AK*du2+p1
    return
    else
    if(p.gt.0)return !28
c          istat =5 から istat=7 へ ( 除荷 )
    istat=7 !29
    du2= p/AK
    du3=du-du2+ug !30
    AK=AK_1*0.000000001 !31
    p=0 !32
    return
    endif
c          istat =6 ( 負側 )
    elseif(istat.eq.6) then !33
    p = AK*du + p
    if(du.lt.0.) then
    if(p.gt.-p2)return !34
c          istat =6 から istat=2or4 へ ( 負側 )
    du2= (p+p2)/AK
    if(p2.lt.Q_2)then !35
    istat=2
    AK=AK_2
    else
    istat=4 !36
    AK=AK_3
    endif
    p=AK*du2-p2
    return
    else

```

```

      if(p.lt.0)return                                     !37
c                                     istat =6 から istat=7 へ ( 除荷 )
      istat=7                                             !38
      du2= p/AK
      du4=du-du2+ug                                       !39
      AK=AK_1*0.000000001                                !40
      p=0                                                 !41
      return
    endif
c                                     istat =7 ( スリップ )
      elseif(istat.eq.7) then                             !42
      u=ug+du
      if(du.gt.0.) then
      if(u.lt.du3)return                                  !43
c                                     istat =7 から istat=5 へ ( 正側 )
      istat=5                                             !44
      du2=u-du3
      AK=AK_1
      p=AK*du2
      return
      else
      if(u.gt.du4)return                                  !45
c                                     istat =7 から istat=6 へ ( 負側 )
      istat=6                                             !46
      du2=u-du4
      AK=AK_1
      p=AK*du2
      return
    endif
    else
c                                     初期設定
      istat = 0                                           !47
      AK=AK_1
      p1=Q_1                                              !48
      p2=p1
      p=0.
      du3=0                                              !49
      du4=0
      goto 100
    endif
    return
  end

```

1. 状態が $istat : 0$ の場合、以下の処理を行う。
2. 増分変位が正の場合 ($du > 0$) で、増分後の応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。
3. 増分後の応力が $p1$ より大きい場合は状態が 1 となり、以下の処理を行う。
4. 増分変位が負の場合 ($du < 0$) で、増分後の応力が負方向の境界値 $-p2$ よりも大きければ、このサブルーチンより戻る。

5. 増分後の応力が $-p_2$ よりも小さい場合は状態が 2 となり、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. 増分変位が正の場合 ($du > 0$) で、増分後の応力が正方向の境界値 Q_2 より小さければ、このサブルーチンより戻る。
8. 増分後の応力が Q_2 より大きい場合は状態が 3 となり、以下の処理を行う。
9. 増分変位が負 ($du < 0$) ならば、状態を 5 にし、以下の処理を行う。
10. 増分前の応力を求めて p_1 にセットする。
11. 状態が $istat : 2$ の場合、以下の処理を行う。
12. 増分変位が負の場合 ($du < 0$) で、増分後の応力が負方向の境界値 $-Q_2$ よりも大きければ、このサブルーチンより戻る。
13. 増分後の応力が $-Q_2$ よりも小さい場合は状態が 4 となり、以下の処理を行う。
14. 増分変位が正 ($du > 0$) ならば、状態を 6 にし、以下の処理を行う。
15. 増分前の応力を求めて $-p_2$ にセットする。
16. 状態が 3 の場合、以下の処理を行う。
17. 増分変位が正 ($du > 0$) ならば、このサブルーチンより戻る。
18. 増分変位が負 ($du < 0$) ならば、状態を 5 にし、以下の処理を行う。
19. 増分前の応力を求めて p_1 にセットする。
20. 状態が $istat : 4$ の場合、以下の処理を行う。
21. 増分変位が負 ($du < 0$) ならば、このサブルーチンより戻る。
22. 増分変位が正 ($du > 0$) ならば、状態を 6 にし以下の処理を行う。
23. 増分前の応力を求めて $-p_2$ にセットする。
24. 状態が $istat : 5$ の場合、以下の処理を行う。
25. 増分変位が正の場合 ($du > 0$) で、増分後の応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。
26. 増分後の応力が p_1 より大きい場合で、 p_1 が第 2 折れ点 Q_2 より小さければ、状態を 1 にして以下の処理を行う。
27. 増分後の応力が p_1 より大きい場合で、 p_1 が第 2 折れ点 Q_2 より大きければ、状態を 3 にして以下の処理を行う。
28. 増分変位が負の場合 ($du < 0$) で、増分後の応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。
29. 増分変位が負の場合 ($du < 0$) で、増分後の応力が負方向の境界値 0 よりも小さければ、状態は 7 とし以下の処理を行う。
30. 応力が 0 となる点の変位を求め、正側の残留変位 du_3 にセットす

- る。
31. スリップ剛性は極力 0 に近い値とするため、接線剛性 AK に $AK_1 \times 0.000000001$ 倍した値をセットする。
 32. 応力を 0 にする。
 33. 状態が $istat : 6$ の場合、以下の処理を行う。
 34. 増分変位が負の場合 ($du < 0$) で、増分後の応力が負方向の境界値 $-p_2$ よりも大きければ、このサブルーチンより戻る。
 35. 増分後の応力が負方向の境界値 $-p_2$ よりも小さい場合で、 p_2 が Q_2 よりも小さければ、状態は 2 とし以下の処理を行う。
 36. 増分後の応力が負方向の境界値 $-p_2$ よりも小さい場合で、 p_2 が Q_2 よりも大きければ、状態は 4 とし以下の処理を行う。
 37. 増分変位が正の場合 ($du > 0$) で、増分後の応力が正方向の境界値 0 より小さければ、このサブルーチンより戻る。
 38. 増分後の応力が 0 より大きい場合は状態が 7 となり、以下の処理を行う。
 39. 応力が 0 となる点の変位を求め、正側の残留変位 du_4 にセットする。
 40. スリップ剛性は極力 0 に近い値とするため、接線剛性 AK に $AK_1 \times 0.000000001$ 倍した値をセットする。
 41. 応力を 0 にする。
 42. 状態が $istat : 7$ の場合、前ステップまでの総変位 u_g に増分変位 du を足し合わせて、増分後の変位 u を求め、以下の処理を行う。
 43. 増分変位が正の場合 ($du > 0$) で、変位 u による正方向境界値 du_3 より小さければ、このサブルーチンより戻る。
 44. 増分変位が正の場合 ($du > 0$) で、変位 u による正方向境界値 du_3 より大きければ、状態を 5 とし、以下の処理を行う。
 45. 増分変位が負の場合 ($du < 0$) で、変位 u による負方向境界値 du_4 より大きければ、このサブルーチンより戻る。
 46. 増分変位が負の場合 ($du < 0$) で、変位 u による負方向境界値 du_4 より小さければ、状態を 4 とし、以下の処理を行う。
 47. 状態が -1 であるならば、初期設定として状態 0 とし、以下の処理を行う。
 48. p_1 と p_2 に第一折れ点 Q_1 をセットする。
 49. 残留変位 du_3 と du_4 をゼロセットする。

4.3.6 ボックス型 スリップモデル

本節では、接合部に関する履歴ルールで、図 4-38 に示すボックス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である。

第 1 折れ点の応力	: Q_1	応力が正の正方向最大変位	: du3
第 2 折れ点の応力	: Q_2	応力が負の負方向最大変位	: du4
除荷時の折れ点の応力	: Q_3	応力が負の正方向最大変位	: du5
第 1 勾配	: AK_1	応力が正の負方向最大変位	: du6
第 2 勾配	: AK_2	正方向スリップ変位の最大値	: u_slip1
除荷勾配	: AK_3	負方向スリップ変位の最大値	: u_slip2
最高点を指向する勾配	: AK_4	除荷時の状態	: st
スリップ剛性パラメーター	: SP	ただし、AK_1 < AK_3 である事。	

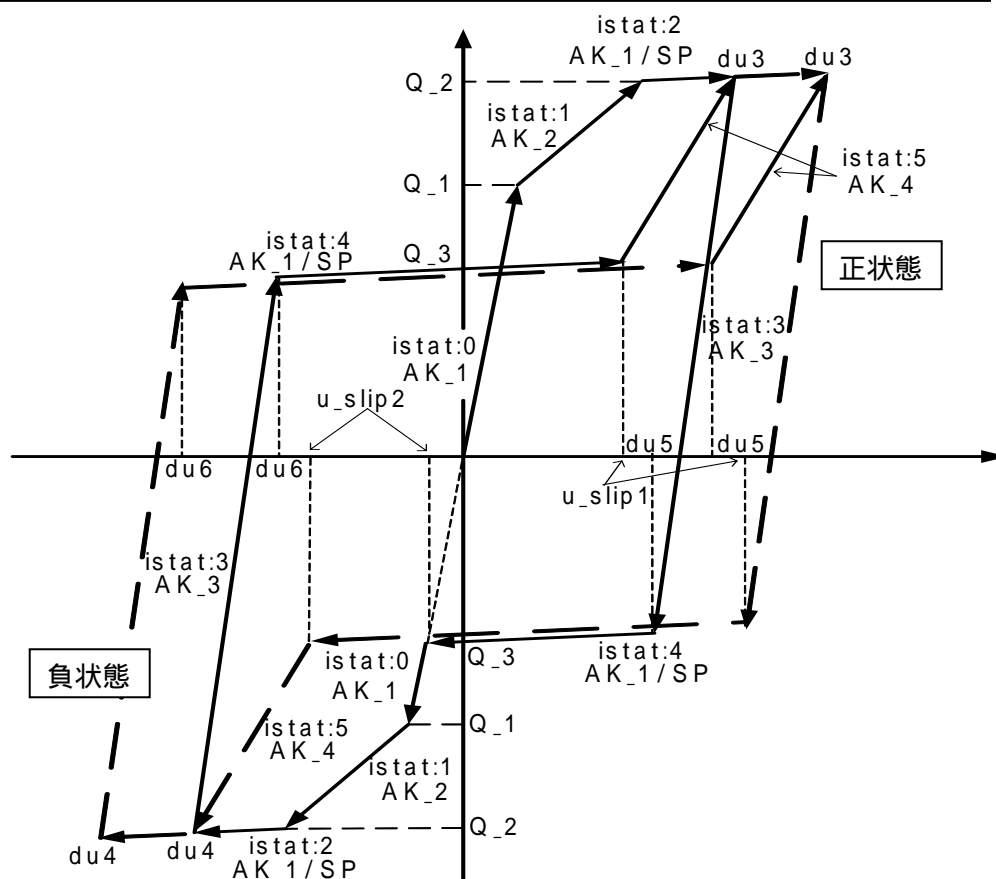


図 4-38 接合部のボックス型スリップモデル履歴特性

この履歴特性は木質構造物の接合部の解析を行うために作成した履歴特性である。正負対称のトリリニアの骨格曲線を持っているが、除荷した時にボックスを持つ履歴となる。除荷時の剛性に初期の剛性と異なる値を用いる事ができ、スリップ時の剛性と第3勾配にはパラメ

ーターSP を用いて、ユーザー側で自由に設定できる。弾性状態は、 $istat:0$ 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 $istat:2$ 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。 $istat:5$ は状態 4 でスリップ勾配を終えてから最高点に向けて指向する状態を表す。

ここで、図 4-38 を用いて全体の流れについて説明する。初期の状態 0 では、境界値 $Q_1 > -Q_1$ の範囲内で直線的に進行するが、増分変位が反転した時、状態を 3 にして増分時の応力を正の時は p_1 、負の時は p_4 にセットする。除荷されて状態 3 に進行する履歴は後にまとめる。増分変位が正方向に進行し正方向境界値 Q_1 を越えて進行すると正側の状態 1 に進行する。反対に状態 0 で負方向に進行し負方向境界値 $-Q_1$ を越えると負側の状態 1 に入る。

ここでは、正方向に進んで正側の状態 1 に進行する履歴について説明する。状態 1 は正方向に進む場合、境界値は $Q_2 >$ であり、正方向境界値 Q_2 を越えると状態 2 となる。また、状態 1 において除荷され増分変位が反転して負になると状態は 3 となり、除荷された時の応力を p_1 とし状態 3 の境界値とする。除荷して状態 3 に進む履歴については後述する。

状態 1 から正方向に増分変位が進行し、状態 2 に進む場合を説明する。状態 2 の接線剛性は第 1 勾配 AK_1 をスリップ剛性パラメーターSP で除した値とし、増分変位が正方向に進行する場合、その上限はない。増分変位が反転して負方向に進行する場合、状態は 3 となって除荷時の応力を p_3 とし、除荷時の変位を正側の最大変位 du_3 にセットする。この時の状態 3 は $p_3 > p_4$ の範囲内で直線に進行するが、 p_4 を状態 3 に入った時に求める必要がある。これについては、後ほどプログラム内で説明する。状態 3 において増分変位が反転し正方向に進み、正方向境界値 p_3 を越えると状態 2 に戻る。また、増分変位が反転せずに負方向境界値 p_4 を越えると、状態はスリップを表す状態 4 に進行する。状態 4 で増分変位が反転して正方向に進行すると状態は 3 となる。しかし、この時状態 3 は複数の履歴を持つため、後にまとめて説明する。

負側状態 4 で増分変位が反転せずに負方向に進行し、状態 0 と交差すると状態は 0 となる。これは、一度も負方向の履歴を経験していない場合であり、もしも一度負方向の状態 2 まで経験していれば、負方向の最大点を指向する状態 5 と交差する点まで状態 4 は進行する。状態 5 と交差すると状態 5 になり、負側の最大点を指向して進む。

負側で状態 0 から状態 2 まで進行したとすると、これまでに除荷さ

れ状態 3 に入る。履歴については後にまとめる。状態 2 で除荷される状態 3 に進行する時、除荷された時の応力を p_6 とし、変位を負側の最大変位 du_4 にセットする。この時の状態 3 は境界値を $p_1 > p_6$ とするが、 p_1 を除荷したときに求める必要がある。これについてはプログラム内で説明する。状態 3 で増分変位が反転し負方向に進行する時、負方向の境界値 p_6 を越えると状態は 2 に戻り、増分変位が反転せずに正方向に進行し、正方向境界値 p_1 を越えるとスリップを表す状態 4 に変化する。状態 4 で増分変位が反転し負方向に進行する時、その履歴は複数あり、後にまとめて説明する。状態 4 で増分変位が反転せずに正方向に進行し、正方向の最大点を指向する状態 5 と交差すると状態は 5 になる。状態 5 で増分変位が反転して負方向に進行すると、状態は 3 となる。また、状態 5 で増分変位が反転せずに正方向に進行し、正方向の境界値 p_3 を越えると状態 2 となる。

これまでで、ボックス型スリップモデルの全体の流れを説明した。しかし、この履歴特性は各状態から除荷した時に複雑な履歴を描く特性を持っている。したがって、この履歴について 正側から除荷した場合、と 負側から除荷した場合、に分けて説明する。

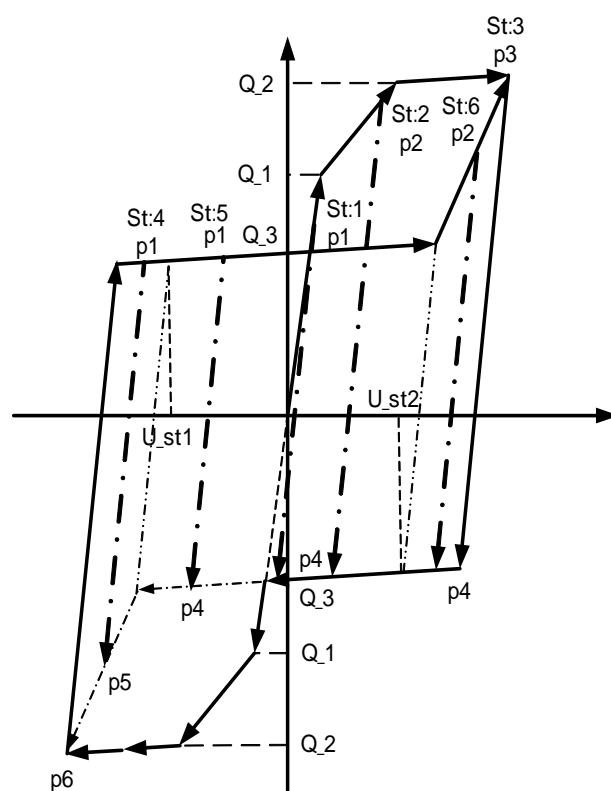


図4-39 正側から除荷した場合の履歴

正側から除荷した場合の履歴

図 4-39 を用いて、正側から除荷した場合の履歴について示す。
除荷の状態を st によって分別し、 st の値によってどのような履歴を描くのかを以下にまとめる。

1. $st : 1$ の時

状態 0 から除荷され状態 3 に進行した時、除荷状態を表す st を 1 に除荷時の応力を p_1 とし、状態 3 の境界値は $p_1 > p_4$ となる。この時、負方向の境界値 p_4 に値はなく、状態 0 から状態 3 に進行する時に求める必要があり、 p_1 の点の変位にスリップ剛性を掛けて $-Q_3$ に足す事によって求める。状態 3 で増分変位が負方向に進行し、負方向の境界値 p_4 を越えるとスリップを表す状態 4 に進み、状態 3 で増分変位が正方向に進行し、正方向の境界値 p_1 を越えると状態 0 に戻る。

2. $st : 2$ の時

状態 1 から除荷され状態 3 に進行した時、除荷状態を表す st を 2 に除荷時の応力を p_2 とし、状態 3 の境界値は $p_2 > p_4$ となる。この時、負方向の境界値 p_4 に値はなく、状態 1 から状態 3 に進行する時に求める必要があり、 p_2 の点の変位にスリップ剛性を掛けて $-Q_3$ に足す事によって求める。状態 3 で増分変位が負方向に進行し、負方向の境界値 p_4 を越えるとスリップを表す状態 4 に進み、状態 3 で増分変位が正方向に進行し、正方向の境界値 p_2 を越えると状態 1 に戻る。

3. $st : 3$ の時

状態 2 から除荷され状態 3 に進行した時、除荷状態を表す st を 3 に除荷時の応力を p_3 とし、状態 3 の境界値は $p_3 > p_4$ となる。この時、負方向の境界値 p_4 に値はなく、状態 2 から状態 3 に進行する時に求める必要があり、 p_3 に Q_2 と Q_3 を引いた値が p_4 となる。状態 3 で増分変位が負方向に進行し、負方向の境界値 p_4 を越えるとスリップを表す状態 4 に進み、状態 3 で増分変位が正方向に進行し、正方向の境界値 p_3 を越えると状態 2 に戻る。

4. $st : 4$ の時

負側で状態 2 まで経験しており、除荷されて正側のスリップを表す状態 4 に進行した時、この状態 4 で除荷されて状態 3 に入る履歴を負側の最大変位 du_4 を用いて、除荷時の状態 st を 2 つに分ける。この 2 つに分ける条件として u_st1 という変数を用いる。 u_st1 は du_4 に u_1 と u_2 を足し合わせる事によって求められる。 u_st1 と u_1 、 u_2 についてはプログラム内で詳しく説明する。この

u_{st1} より小さい変位 u で除荷された時、除荷状態 st は 4 となり、除荷時の応力を $p1$ にして状態 3 に進行する。この時の状態 3 は $p1 > -p5$ の範囲内で直線に進むが、負方向の境界値 $p5$ には値がなく、 $p6$ を用いて求める必要がある。この計算についてはプログラム内で説明する。増分変位が正方向に進行し、正方向境界値 $p1$ を越えると状態 4 に戻り、負方向に進行し、負方向境界値 $p5$ を越えると状態 5 となって負側の最大点を指向して進む。

5. $st : 5$ の時

st が 4 となった時の条件である u_{st1} より大きい変位 u で除荷された時、除荷状態 st は 5 となり、除荷時の応力を $p1$ にして状態 3 に進行する。この時の状態 3 は $p1 > -p4$ の範囲内で直線に進むが、負方向の境界値 $p4$ には値がなく、 $p1$ から $2 \times Q_3$ 引いて求める必要がある。増分変位が正方向に進行し、正方向境界値 $p1$ を越えると状態 4 に戻り、負方向に進行し、負方向境界値 $p4$ を越えると負側のスリップを表す状態 4 となる。

6. $st : 6$ の時

状態 5 から除荷され状態 3 に進行する時、除荷状態を表す st を 6 に除荷時の応力を $p2$ とし、状態 3 の境界値は $p2 > -p4$ となる。この時、負方向の境界値 $p4$ に値はなく、状態 2 から状態 3 に進行する時に求める必要があり、 $p2$ の点の変位にスリップ剛性を掛けて $-Q_3$ に足す事によって求める。状態 3 で増分変位が負方向に進行し、負方向の境界値 $p4$ を越えるとスリップを表す状態 4 に進み、状態 3 で増分変位が正方向に進行し、正方向の境界値 $p2$ を越えると状態 5 に戻る。

負側から除荷した場合の履歴

図 4-39 に負側から除荷した場合の履歴について示す。と同様に除荷状態 st によって、どのような履歴を描くか説明する。

1. $st : 7$ の時

状態 0 から除荷され状態 3 に進行した時、除荷状態を表す st を 7 に除荷時の応力を $p4$ とし、状態 3 の境界値は $p1 > -p4$ となる。この時、正方向の境界値 $p1$ に値はなく、状態 0 から状態 3 に進行する時に求める必要があり、 $p4$ の点の変位にスリップ剛性を掛けて Q_3 から引く事によって求める。状態 3 で増分変位が正方向に進行し、正方向の境界値 $p1$ を越えるとスリップを表す状態 4 に進み、状態 3 で増分変位が負方向に進行し、負方向の境界値 $p4$ を越えると状態 0 に戻る。

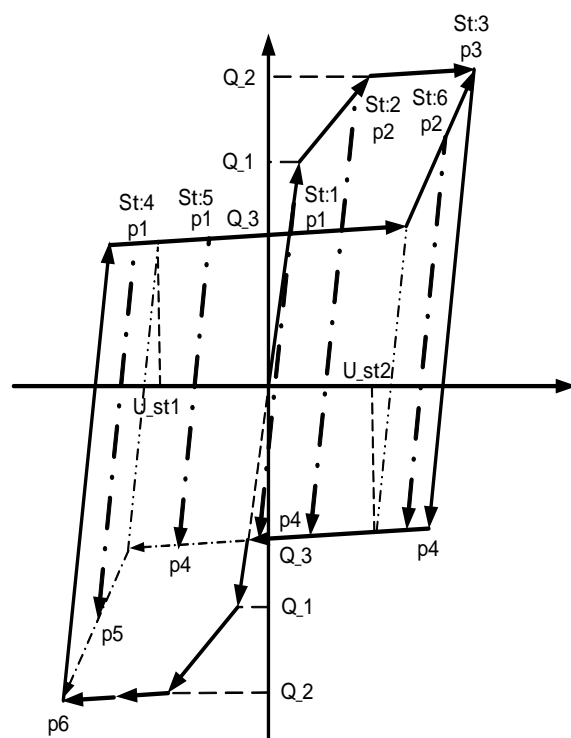


図 4-40 負側から除荷した場合の履歴

2. st : 8 の時

状態 1 から除荷され状態 3 に進行した時、除荷状態を表す st を 8 に除荷時の応力を p_5 とし、状態 3 の境界値は $p_1 > p_5$ となる。この時、正方向の境界値 p_1 に値はなく、状態 1 から状態 3 に進行する時に求める必要があり、 p_5 の点の変位にスリップ剛性を掛けて Q_3 から引く事によって求める。状態 3 で増分変位が正方向に進行し、正方向の境界値 p_1 を越えるとスリップを表す状態 4 に進み、状態 3 で増分変位が負方向に進行し、負方向の境界値 p_5 を越えると状態 1 に戻る。

3. st : 9 の時

状態 2 から除荷され状態 3 に進行した時、除荷状態を表す st を 9 に除荷時の応力を p_6 とし、状態 3 の境界値は $p_1 > p_6$ となる。この時、正方向の境界値 p_1 に値はなく、状態 2 から状態 3 に進行する時に求める必要があり、 p_6 に Q_2 と Q_3 を足した値が p_1 となる。状態 3 で増分変位が正方向に進行し、正方向の境界値 p_1 を越えるとスリップを表す状態 4 に進み、状態 3 で増分変位が負方向に進行し、負方向の境界値 p_6 を越えると状態 2 に戻る。

4. st : 10 の時

正側で状態 2 まで経験しており、除荷されて負側のスリップを表す状態 4 に進行した時、この状態 4 で除荷されて状態 3 に入る履歴を正側の最大変位 du_3 を用いて、除荷時の状態 st を 2 つに分ける。この 2 つに分ける条件として u_{st2} という変数を用いる。 u_{st2} は du_3 から u_1 と u_2 を引く事によって求められる。 u_{st2} と u_1 、 u_2 についてはプログラム内で詳しく説明する。この u_{st2} より大きい変位 u で除荷された時、除荷状態 st は 10 となり、除荷時の応力を p_4 にして状態 3 に進行する。この時の状態 3 は $p_2 > p_4$ の範囲内で直線に進むが、正方向の境界値 p_2 には値がなく、 p_3 を用いて求める必要がある。この計算についてはプログラム内で説明する。増分変位が負方向に進行し、負方向境界値 p_4 を越えると状態 4 に戻り、正方向に進行し、正方向境界値 p_2 を越えると状態 5 となって正側の最大点を指向して進む。

5. st : 11 の時

st が 10 となった時の条件である u_{st2} より小さい変位 u で除荷された時、除荷状態 st は 11 となり、除荷時の応力を p_4 にして状態 3 に進行する。この時の状態 3 は $p_1 > p_4$ の範囲内で直線に進むが、正方向の境界値 p_1 には値がなく、 p_4 に $2 \times Q_3$ を足して求める必要がある。増分変位が負方向に進行し、負方向境界値 p_4 を越えると状態 4 に戻り、正方向に進行し、正方向境界値 p_1 を越えると正側のスリップを表す状態 4 となる。

6. st : 12 の時

状態 5 から除荷され状態 3 に進行した時、除荷状態を表す st を 12 に除荷時の応力を p_5 とし、状態 3 の境界値は $p_1 > p_5$ となる。この時、正方向の境界値 p_1 に値はなく、状態 2 から状態 3 に進行する時に求める必要があり、 p_5 の点の変位にスリップ剛性を掛けて Q_3 から引く事によって求める。状態 3 で増分変位が正方向に進行し、正方向の境界値 p_1 を越えるとスリップを表す状態 4 に進み、状態 3 で増分変位が負方向に進行し、負方向の境界値 p_2 を越えると状態 5 に戻る。

以上の説明をまとめると、ボックス型スリップモデル履歴特性は、次の 5 つの状態で表現される。

- istat : 0 弾性状態であり、接線剛性は第 1 勾配 AK_1 である。 $Q_1 > -Q_1$ の中で直線的に動き、増分変位が反転すると応力が正の時は除荷時の応力を p_1 とし、応力が負の時は除荷時の応力を p_4 として状態 3 となる。増分変位が正方向に進み Q_1 を越えると正側状態 1 に、 $-Q_1$ を越えると負側状態 1 となる。
- istat : 1 正負共に第 2 勾配上であり、接線剛性は AK_2 である。応力が正で増分変位が正方向に進行する時 $Q_2 > -Q_2$ 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、状態が 3 となって反転した時の応力を p_2 にする。
- また、応力が負の場合で増分変位が負方向に進行する時 $-Q_2 > Q_2$ 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、状態が 3 となって反転した時の応力を p_5 にする。
- istat : 2 正負共に第 3 勾配上であり、接線剛性は AK_1/SP である。応力が正で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、状態が 3 となり、反転した時の応力を p_3 にする。
- 応力が負で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、状態が 3 となり、反転した時の応力を p_6 にする。
- istat : 3 正負共に状態 0, 1, 2, 4 で除荷された時の状態であり、接線剛性は除荷勾配 AK_3 である。除荷状態 st によって履歴が異なり、境界条件は st が 1, 5, 7, 11 の時 $p_1 > p_4$ となり、 st が 4, 8, 12 の時 $p_1 > p_5$ 、 st が 9 の時 $p_1 > p_6$ 、 st が 2, 6, 10 の時 $p_2 > p_4$ 、 st が 3 の時 $p_3 > p_4$ となる。増分変位が正で正方向の境界値を越える場合、 st が 1 の時状態は 0 に、 st が 2 の時状態は 1 に、 st が 3 の時状態は 3 に、 st が 6, 10 の時状態は 5 に進む。その他の st については、全て状態 4 に進行する。
- 増分変位が負で負方向の境界値を越える場合、 st が 7 の時状態は 0 に、 st が 8 の時状態は 1 に、 st が 9 の時状態は 2 に、 st が 4, 12 の時状態は 5 に進む。その他の st については、全て状態 4 に進行する。
- istat : 4 正負共にスリップの状態であり、接線剛性は AK_1/SP である。この状態では境界条件は変位 u によって得られており、応力が正の場合は増分変位が正方向の時、その境界条件は

$u_slip1 > u$ である。 u_slip1 の求め方についてはプログラム内で説明する。増分変位が反転して負方向に進行する時、状態は3になり、 u_st1 より変位が大きければ st は5、小さければ st は4になる。

応力が負の場合は増分変位が負方向の時、その境界条件は $u_slip2 > u$ である。 u_slip2 の求め方についてはプログラム内で説明する。増分変位が反転して正方向に進行する時、状態は3になり、 u_st2 より変位が大きければ st は10、小さければ st は11になる。

istat : 5 正負共に最高点を指向する勾配であり、接線剛性は AK_4 である。応力が正の場合で増分変位が正の時 $p3 >$ の中で直線的に動き、正方向に増分変位が進み、 $p3$ を越えると状態は2に進む。また、負方向に増分変位が進むと状態は3になり、除荷時の応力を $p2$ に、除荷時の状態 st を6にする。

応力が負の場合で増分変位が負の時 $>p6$ の中で直線的に動き、負方向に増分変位が進み $p6$ を越えると状態は2に進む。また、正方向に増分変位が進むと状態は3になり、除荷時の応力を $p5$ に、除荷時の状態 st を12にする。

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 /box_slip ( ボックス型スリップ履歴モデル )
C
      subroutine box_slip (AK,istat,AK_1,AK_2,AK_3,AK_4,Q_1,Q_2, Q_3,
*                          du,P,P1,p2,p3,p4,p5,p6,ug,du3,du4,SP)
      implicit real*8(A-H,O-Z)

C
C      AK              : 接線剛性
C      istat           : 現在の状態 (Work)
C      AK_1            : 第一勾配
C      AK_2            : 第二勾配
C      AK_3            : 除荷勾配
C      AK_4            : 最高点指向勾配
C      Q_1             : 第一折れ点
C      Q_2             : 第二折れ点
C      Q_3             : 除荷勾配の折れ点
C      du              : 増分変位
C      P               : 現在の応力 (Work)
C      P1 から p6      : 反曲点上端 (Work)
C      SP              : スリップ剛性パラメーター
C      du3             : 正側最大変位

```

```

c    du4                :負側最大変位
c    u_slip1            :正方向スリップ変位の最大値
c    u_slip2            :負方向スリップ変位の最大値
c    u_st1              :正側状態4の除荷状態判定パラメーター
c    u_st2              :負側状態4の除荷状態判定パラメーター
C
100  continue
C                                istat0
      if (istat.eq.0)then                                !1
      p=AK*du+p
      if(p.gt.0)then
      if(du.gt.0)then
      if(p.lt.Q_1) return                                !2
C                                istat0 から istat=1 へ (正側)
      istat=1                                            !3
      du2=(p-Q_1)/AK
      AK=AK_2
      p=AK*du2+Q_1
      return
      else
C                                istat0 から istat=3 へ (除荷)
      st=1                                              !4
      dp=AK*du
      AK=AK_3
      du=dp/AK
      p=p-dp
      u=ug+du
      p1=p                                              !5
      p4=(AK_1/SP)*u-Q_3
      istat=3                                           !6
      p=AK*du+p
      return
      endif
      else
      if(du.lt.0)then
      if(p.gt.-Q_1)return                                !7
C                                istat0 から istat=1 へ (負側)
      istat=1                                            !8
      du2=(p+Q_1)/AK
      AK=AK_2
      p=AK*du2-Q_1
      return
      else
C                                istat0 から istat=3 へ (除荷)
      st=7                                              !9
      dp=AK*du
      AK=AK_3
      du=dp/AK
      p=p-dp
      u=ug+du
      p4=p                                              !10
      p1=(AK_1/SP)*u+Q_3                                !11
      istat=3                                           !12
      p=AK*du+p

```

```

    return
    endif
    endif

C                                     istat1
    elseif(istat.eq.1)then
    p=AK*du+p
    if(p.gt.0)then
    if(du.gt.0)then
    if(p.lt.p3)return
    C                                     istat1 から istat=2 へ (正側)
    istat=2
    du2=(p-p3)/AK
    AK=AK_1/SP
    p=AK*du2+p3
    return
    C                                     istat1 から istat=3 へ (除荷)
    else
    st=2
    dp=AK*du
    AK=AK_3
    du=dp/AK
    p=p-dp
    u=ug+du
    p2=p
    p4=(AK_1/SP)*u-Q_3
    istat=3
    p=AK*du+p
    return
    endif
    else
    if(du.lt.0)then
    if(p.gt.p6)return
    C                                     istat1 から istat=2 へ (負側)
    istat=2
    du2=(p-p6)/AK
    AK=AK_1/SP
    p=AK*du2+p6
    return
    C                                     istat1 から istat=3 へ (除荷)
    else
    st=8
    dp=AK*du
    AK=AK_3
    du=dp/AK
    p=p-dp
    u=ug+du
    p5=p
    p1=(AK_1/SP)*u+Q_3
    istat=3
    p=AK*du+p
    return
    endif
    endif

C                                     istat2 (正側)

```

```

elseif(istat.eq.2)then                                !26
  p=AK*du+p
  if(p.gt.0)then
    if(du.gt.0)return                                !27
C                                                    istat2 から istat=3 へ(除荷)
    st=3                                              !28
    dp=AK*du
    AK=AK_3
    du=dp/AK
    p=p-dp
    u=ug+du
    p3=p                                              !29
    istat=3
    p=AK*du+p
    if(u.lt.du3)return                                !30
    du3=u                                              !31
    p4=p3-Q_2-Q_3                                    !32
    u_slip1=-u1+du3                                  !33
    u_st2=du3-u2-u1                                  !34
    return
C                                                    istat2 ( 負側 )
  else
    if(du.lt.0)return                                !35
C                                                    istat2 から istat=3 へ(除荷)
    st=9                                              !36
    dp=AK*du
    AK=AK_3
    du=dp/AK
    p=p-dp
    u=ug+du
    p6=p                                              !37
    istat=3                                           !38
    p=AK*du+p
    if(u.gt.du4)return                                !39
    du4=u                                              !40
    p1=p6+Q_2+Q_3                                    !41
    u_slip2=u1+du4                                  !42
    u_st1=du4+u2+u1                                  !43
    return
  endif
C                                                    istat3
elseif(istat.eq.3)then                                !44
  p=AK*du+p
C                                                    istat3(正方向)
  if(du.gt.0)then
    if(st.eq.1.or.st.eq.4.or.st.eq.5.or.st.eq.7.or.st.eq.8.
*      or.st.eq.9.or.st.eq.11.or.st.eq.12)then
    if(p.lt.p1) return                                !45
C                                                    istat3 から istat=0/4 へ ( 正側 )
    if(st.eq.1)then
      istat=0                                          !46
      du2=(p-p1)/AK
      AK=AK_1
      p=AK*du2+p1

```

```

elseif(st.eq.4.or.st.eq.5.or.st.eq.7.or.st.eq.8.
*                               or.st.eq.9.or.st.eq.11.or.st.eq.12) then
    istat=4                                !47
    du2=(p-p1)/AK
    AK=AK_1/SP
    p=AK*du2+p1
    endif
    return
elseif(st.eq.2.or.st.eq.6.or.st.eq.10) then
    if(p.lt.p2) return                                !48
C                               istat3 から istat=1/5 へ (正側)
    if(st.eq.2) then
        istat=1                                !49
        du2=(p-p2)/AK
        AK=AK_2
        p=AK*du2+p2
    elseif(st.eq.6.or.st.eq.10) then
        istat=5                                !50
        du2=(p-p2)/AK
        AK=AK_4
        p=AK*du2+p2
    endif
    return
elseif(st.eq.3) then
    if(p.lt.p3) return                                !51
C                               istat3 から istat=2 へ (正側)
    istat=2                                !52
    du2=(p-p3)/AK
    AK=AK_1/SP
    p=AK*du2+p3
    return
    endif
C                               istat3 (負方向)
else
    if(st.eq.1.or.st.eq.2.or.st.eq.3.or.st.eq.5.or.st.eq.6.
*                               or.st.eq.7.or.st.eq.11.or.st.eq.10) then
        if(p.gt.p4) return                                !53
C                               istat3 から istat=0/4 へ (負側)
        if(st.eq.7) then
            istat=0                                !54
            du2=(p-p4)/AK
            AK=AK_1
            p=AK*du2+p4
        elseif(st.eq.1.or.st.eq.2.or.st.eq.3.or.st.eq.5.
*                               or.st.eq.6.or.st.eq.10.or.st.eq.11) then
            istat=4                                !55
            du2=(p-p4)/AK
            AK=AK_1/SP
            p=AK*du2+p4
            endif
            return
        elseif(st.eq.4.or.st.eq.8.or.st.eq.12) then
            if(p.gt.p5) return                                !56
C                               istat3 から istat=1/5 へ (負側)

```

```

    if(st.eq.8)then
    istat=1
    du2=(p-p5)/AK
    AK=AK_2
    p=AK*du2+p5
    elseif(st.eq.4.or.st.eq.12)then
    istat=5
    du2=(p-p5)/AK
    AK=AK_4
    p=AK*du2+p5
    endif
    return
    elseif(st.eq.9)then
    if(p.gt.p6)return
    C
    istat=2
    du2=(p-p6)/AK
    AK=AK_1/SP
    p=AK*du2+p6
    return
    endif
    C
    elseif(istat.eq.4)then
    if(p.gt.0)then
    if(du.gt.0)then
    p=AK*du+p
    u=ug+du
    if(u.lt.u_slip1)return
    C
    if(p3.eq.Q_2)then
    istat=0
    du2=u-u_slip1
    p=-AK*du2+p
    AK=AK_1
    p=AK*du2+p
    return
    else
    istat=5
    du2=u-u_slip1
    p=-AK*du2+p
    AK=AK_4
    p=AK*du2+p
    return
    endif
    C
    else
    p=AK*du+p
    u=ug+du
    if(u.lt.u_st1)then
    st=4
    dp=AK*du
    AK=AK_3
    du=dp/AK
    istat3 から istat=2 へ (負側)
    istat4 (正側)
    istat4 から istat=0 へ (正側)
    istat4 から istat=3 へ (除荷)

```

```

    p=p-dp
    p1=p !66
    p_st=(-du4-u2+ug)*AK_4 !67
    p5=p6+p_st !68
    istat=3 !69
    p=AK*du+p
    return
  else
    st=5 !70
    dp=AK*du
    AK=AK_3
    du=dp/AK
    p=p-dp
    p1=p !71
    p4=(-2*Q_3)+p1 !72
    istat=3 !73
    p=AK*du+p
    return
  endif
endif
else
C                                     istat4 ( 負側 )
    if (du.lt.0) then
    p=AK*du+p
    u=ug+du
    if (u.gt.u_slip2) return !74
C                                     istat4 から istat=0 へ ( 負側 )
    if (p6.eq.-Q_2) then
    istat=0 !75
    du2=u-u_slip2
    p=-AK*du2+p
    AK=AK_1
    p=AK*du2+p
    return
    else
    istat=5 !76
    du2=u-u_slip2
    p=-AK*du2+p
    AK=AK_4
    p=AK*du2+p
    return
    endif
  else
C                                     istat4 から istat=3 へ ( 除荷 )
    p=AK*du+p
    u=ug+du
    if (u.gt.u_st2) then
    st=10 !77
    dp=AK*du
    AK=AK_3
    du=dp/AK
    p=p-dp
    p4=p !78
    p_st=(du3-u2-ug)*AK_4 !79

```

```

        p2=p3-p_st                                !80
        istat=3
        p=AK*du+p
        return
    else
        st=11                                    !81
        dp=AK*du
        AK=AK_3
        du=dp/AK
        p=p-dp
        p4=p                                    !82
        p1=(2*Q_3)+p4
        istat=3                                    !83
        p=AK*du+p
        return
    endif
endif
endif
C                                istat5 ( 正側 )
    elseif (istat.eq.5) then                    !84
        p=AK*du+p
        if (p.gt.0) then
            if (du.gt.0) then
                if (p.lt.p3) return                !85
            C                                istat5 から istat=2 へ ( 正側 )
                istat=2                            !86
                du2=(p-p3)/AK
                AK=AK_1/SP
                p=AK*du2+p3
                return
            else
                C                                istat5 から istat=3 へ ( 除荷 )
                st=6                                !87
                dp=AK*du
                AK=AK_3
                du=dp/AK
                p=p-dp
                u=ug+du
                p2=p                                !88
                p4=(AK_1/SP)*u-Q_3                !89
                istat=3                            !90
                p=AK*du+p
                return
            endif
        else
            C                                istat5 から istat=2 へ ( 負側 )
                if (du.lt.0) then
                    if (p.gt.p6) return                !91
                C                                istat5 から istat=2 へ ( 負側 )
                    istat=2                            !92
                    du2=(p-p6)/AK
                    AK=AK_1/SP
                    p=AK*du2+p6
                    return

```

```

C      else
C                                     istat0 から istat=3 へ ( 除荷 )
      st=12                                     !93
      dp=AK*du
      AK=AK_3
      du=dp/AK
      p=p-dp
      u=ug+du
      p5=p                                     !94
      p1=(AK_1/SP)*u+Q_3                       !95
      istat=3                                   !96
      p=AK*du+p
      return
    endif
  endif

C                                     初期設定
  else
    istat=0                                     !97
    AK=AK_1
    p=0                                         !98
    p1=Q_3
    p2=Q_1
    p3=Q_2
    p4=-Q_3
    p5=-Q_1
    p6=-Q_2
    st=0                                     !99
C                                     判定用初期設定
    u_slip1=Q_3/AK_1                           !100
    u_slip2=-u_slip1
    u1=(Q_2-Q_3)/AK_4                           !101
    u2=(Q_3+Q_2)/AK_3
    goto 100
  end if
end

```

1. 状態が $istat : 0$ の場合、以下の処理を行う。
2. 現在の応力が正で増分変位が正の場合 ($du > 0$)、増分後の応力が正方向の境界値 Q_1 より小さければ、このサブルーチンより戻る。
3. 現在の応力が正で増分後の応力が Q_1 より大きい場合は状態が正側の 1 となり、以下の処理を行う。
4. 現在の応力が正で増分変位が負の場合 ($du < 0$)、除荷時の状態を表す st を 1 にし、以下の処理を行う。
5. 除荷時の応力を p_1 に代入し、状態 3 の負方向境界値となる p_4 をスリップ勾配 AK_1/SP に増分後の変位 u を掛け、 $-Q_3$ から引いて求める。
6. 状態を 3 にする。
7. 現在の応力が負で増分変位が負の場合 ($du < 0$)、増分後の応力が負

- 方向の境界値 - Q_1 より大きければ、このサブルーチンより戻る。
8. 現在の応力が負で増分後の応力が負方向の境界値 - Q_1 より小さければ状態が負側の 1 となり、以下の処理を行う。
 9. 現在の応力が負で増分変位が正の場合 ($du > 0$) 除荷時の状態を表す st を 7 にし、以下の処理を行う。
 10. 除荷時の応力を $p4$ に代入する。
 11. 状態 3 の正方向境界値となる $p1$ をスリップ勾配 AK_1/SP に増分後の変位 u を掛け、 Q_3 に足して求める。
 12. 状態を 3 にする。
 13. 状態が $istat : 1$ の場合、以下の処理を行う。
 14. 現在の応力が正で増分変位が正の場合 ($du > 0$) 増分後の応力が正方向の境界値 $p3$ より小さければ、このサブルーチンより戻る。
 15. 現在の応力が正で増分後の応力が Q_1 より大きい場合は状態が正側の 2 となり、以下の処理を行う。
 16. 接線剛性 AK に第 1 勾配 AK_1 から $S P$ を除した値をセットする。
 17. 現在の応力が正で増分変位が負の場合 ($du < 0$) 除荷時の状態を表す st を 2 にし、以下の処理を行う。
 18. 除荷時の応力を $p2$ に代入する。
 19. 状態 3 の負方向境界値となる $p4$ をスリップ勾配 AK_1/SP に増分後の変位 u を掛け、 $-Q_3$ から引いて求める。
 20. 状態を 3 にする。
 21. 現在の応力が負で増分変位が負の場合 ($du < 0$) 増分後の応力が負方向の境界値 $p6$ より大きければ、このサブルーチンより戻る。
 22. 現在の応力が負で増分後の応力が負方向の境界値 $p6$ より小さければ状態が負側の 2 となり、以下の処理を行う。
 23. 現在の応力が負で増分変位が正の場合 ($du > 0$) 除荷時の状態を表す st を 8 にし、以下の処理を行う。
 24. 除荷時の応力を $p5$ に代入し、状態 3 の正方向境界値となる $p1$ をスリップ勾配 AK_1/SP に増分後の変位 u を掛け、 Q_3 に足して求める。
 25. 状態を 3 にする。
 26. 状態が $istat : 2$ の場合、以下の処理を行う。
 27. 現在の応力が正で増分変位が正の場合 ($du > 0$) このサブルーチンより戻る。
 28. 現在の応力が正で増分変位が負の場合 ($du < 0$) 除荷時の状態を表す st を 3 にし、以下の処理を行う。

29. 除荷時の応力を p_5 に代入し、状態を 3 にする。
30. 増分後の変位 u が正側の最大変位 du_3 より小さければ、このサブルーチンより戻る。
31. 増分後の変位 u が正側の最大変位 du_3 より大きければ、 du_3 に u を代入する。
32. 状態 3 の負方向境界値 p_4 を p_3 から Q_2 と Q_3 を引いて求める。
33. 正方向でスリップが生じる場合に必要となる正方向スリップ変位の最大値 u_{slip1} を求めて代入する。
34. この時の状態 3 が負方向の境界値 p_4 を越えて状態 4 に進行し、この状態 4 で除荷する場合、除荷状態を判定するパラメーター u_{st2} が必要になるため求めて代入する。
35. 現在の応力が負で増分変位が負の場合 ($du < 0$)、このサブルーチンより戻る。
36. 現在の応力が負で増分変位が正の場合 ($du > 0$)、除荷状態を表す st を 9 にし、以下の処理を行う。
37. 除荷時の応力を p_6 に代入する。
38. 状態を 3 にする。
39. 増分後の変位 u が負側の最大変位 du_4 より大きければ、このサブルーチンより戻る。
40. 増分後の変位 u が負側の最大変位 du_4 より小さければ、 du_4 に u を代入する。
41. 状態 3 の正方向境界値 p_1 を p_6 に Q_2 と Q_3 足す事で求める。
42. 負方向でスリップが生じる場合に必要となる負方向スリップ変位の最大値 u_{slip2} を求めて代入する。
43. この時の状態 3 が正方向の境界値 p_1 を越えて状態 4 に進行し、この状態 4 で除荷する場合、除荷状態を判定するパラメーター u_{st1} が必要になるため求めて代入する。
44. 状態が 3 の場合、以下の処理を行う。
45. 増分変位が正 ($du > 0$) で、除荷状態を表す st が 1, 4, 5, 7, 8, 9, 11, 12 のいずれかであり、増分後の応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。
46. 増分後の応力が正方向の境界値 p_1 より大きく除荷状態を表す st が 1 の場合、状態を 0 にして以下の処理を行う。
47. 増分後の応力が正方向の境界値 p_1 より大きく除荷状態を表す st が 4, 5, 7, 8, 11, 12 の場合、状態を 4 にして以下の処理を行う。
48. 増分変位が正 ($du > 0$) で、除荷状態を表す st が 2, 6, 10 のいずれ

- かであり、増分後の応力が正方向の境界値 p_2 より小さければ、このサブルーチンより戻る。
49. 増分後の応力が正方向の境界値 p_2 より大きく除荷状態を表す st が 2 の場合、状態を 1 にして以下の処理を行う。
50. 増分後の応力が正方向の境界値 p_2 より大きく除荷状態を表す st が 6, 10 の場合、状態を 5 にして以下の処理を行う。
51. 増分変位が正 ($du > 0$) で、除荷状態を表す st が 3 であり、増分後の応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。
52. 増分後の応力が正方向の境界値 p_3 より大きい場合、状態を 2 にして以下の処理を行う。
53. 増分変位が負 ($du < 0$) で、除荷状態を表す st が 1, 2, 3, 5, 6, 7, 10, 11 のいずれかであり、増分後の応力が負方向の境界値 p_4 より大きければ、このサブルーチンより戻る。
54. 増分後の応力が負方向の境界値 p_4 より小さく除荷状態を表す st が 7 の場合、状態を 0 にして以下の処理を行う。
55. 増分後の応力が負方向の境界値 p_4 より小さく除荷状態を表す st が 1, 2, 3, 5, 6, 10, 11 の場合、状態を 4 にして以下の処理を行う。
56. 増分変位が負 ($du < 0$) で、除荷状態を表す st が 4, 8, 12 のいずれかであり、増分後の応力が負方向の境界値 p_5 より大きければ、このサブルーチンより戻る。
57. 増分後の応力が負方向の境界値 p_5 より小さく除荷状態を表す st が 8 の場合、状態を 1 にして以下の処理を行う。
58. 増分後の応力が負方向の境界値 p_5 より小さく除荷状態を表す st が 4, 12 の場合、状態を 5 にして以下の処理を行う。
59. 増分変位が負 ($du < 0$) で、除荷状態を表す st が 9 であり、増分後の応力が負方向境界値 p_6 より大きければ、このサブルーチンより戻る。
60. 増分後の応力が負方向の境界値 p_6 より小さい場合、状態を 2 にして以下の処理を行う。
61. 状態が $istat : 4$ の場合、以下の処理を行う。
62. 現在の応力が正で増分変位が正 ($du > 0$) の時、増分後の変位 u が正方向スリップ変位の最大値 u_{slip1} より小さければ、このサブルーチンより戻る。
63. 増分後の変位 u が正方向スリップ変位の最大値 u_{slip1} より大きい時、 p_3 が第 2 折れ点 Q_2 に等しい場合は状態を 0 にして以下の

処理を行う。

64. 増分後の変位 u が正方向スリップ変位の最大値 u_{slip1} より大きい時、 p_3 が第2折れ点 Q_2 に等しくない場合は状態を 5 にして以下の処理を行う。
65. 現在の応力が正で増分変位が負 ($du < 0$) の時、増分後の変位 u が除荷時の状態を判定する u_{st1} より小さい場合、状態を 4 にして以下の処理を行う。
66. 除荷時の応力を p_1 に代入する。
67. 負側から除荷されて進行してきた状態 3 から状態 4 に折れた点と、今回除荷された点までの変位量を計算し、最大点を指向する AK_4 にかけて p_{st} を求める。
68. p_{st} を負側の最大応力 p_6 に足し合わせ、状態 3 における負方向境界値 p_5 とする。
69. 状態を 3 にする。
70. 現在の応力が正で増分変位が負 ($du < 0$) の時、増分後の変位 u が除荷時の状態を判定する u_{st1} より大きい場合、状態を 5 にして以下の処理を行う。
71. 除荷時の応力を p_1 に代入する。
72. 状態 3 の負方向境界値となる p_4 を p_1 から $2 \times Q_3$ を引いて求める。
73. 状態を 3 にする。
74. 現在の応力が負で増分変位が負 ($du < 0$) の時、増分後の変位 u が負方向スリップ変位の最大値 u_{slip2} より大きければ、このサブルーチンより戻る。
75. 増分後の変位 u が負方向スリップ変位の最大値 u_{slip2} より小さく、 p_6 が負側第2折れ点 Q_2 と等しければ、状態を 0 にして以下の処理を行う。
76. 増分後の変位 u が負方向スリップ変位の最大値 u_{slip2} より小さく、 p_6 が負側第2折れ点 Q_2 と等しくなければ、状態を 5 にして以下の処理を行う。
77. 現在の応力が負で増分変位が正 ($du > 0$) の時、増分後の変位 u が u_{st2} より大きければ、除荷状態 st を 10 にして以下の処理を行う。
78. 除荷時の応力を p_4 に代入する。
79. 正側から除荷されて進行してきた状態 3 から状態 4 に折れた点と、今回除荷された点までの変位量を計算し、最高点を指向する AK_4 にかけて p_{st} を求める。
80. p_{st} を正側の最大応力 p_3 から引き、状態 3 における正方向境界値

- p2 とする。更に状態を 3 にする。
81. 現在の応力が負で増分変位が正 ($du > 0$) の時、増分後の変位 u が u_{st2} より小さければ、除荷状態 st を 11 にして以下の処理を行う。
 82. 除荷時の応力を $p4$ に代入し、 $p4$ に $2 \times Q_3$ を足して $p1$ を求める。
 83. 状態を 3 にする。
 84. 状態が $istat : 5$ の場合、以下の処理を行う。
 85. 現在の応力が正で増分変位が正 ($du > 0$) の時、増分後の応力が正方向の最大応力 $p3$ より小さければ、このサブルーチンより戻る。
 86. 増分後の応力が正方向の最大応力 $p3$ より大きければ状態を 2 にして以下の処理を行う。
 87. 現在の応力が正で増分変位が負 ($du < 0$) の時、除荷状態を表す st を 6 にして、以下の処理を行う。
 88. 除荷時の応力を $p2$ に代入する。
 89. 状態 3 の負方向境界値となる $p4$ をスリップ勾配 AK_1/SP に増分後の変位 u を掛け、 $-Q_3$ から引いて求める。
 90. 状態を 3 にする。
 91. 現在の応力が負で増分変位が負 ($du < 0$) の時、増分後の応力が $p6$ より大きければ、このサブルーチンより戻る。
 92. 増分後の応力が $p6$ より小さければ状態は 2 となり、以下の処理を行う。
 93. 現在の応力が負で増分変位が正 ($du > 0$) の時、除荷状態を表す st を 12 にして以下の処理を行う。
 94. 除荷時の応力を $p5$ に代入する。
 95. 状態 3 の正方向境界値となる $p1$ をスリップ勾配 AK_1/SP に増分後の変位 u を掛け、 Q_3 から足して求める。
 96. 状態を 3 にする。
 97. 状態が -1 の時、状態を 0 にして以下の処理を行う。
 98. 応力をゼロセットし、境界値を表す $p1$ に Q_3 、 $p2$ に Q_1 、 $p3$ に Q_2 、 $p4$ に $-Q_3$ 、 $p5$ に $-Q_1$ 、 $p6$ に $-Q_2$ をセットする。
 99. 除荷状態を表す st をゼロセットする。
 100. Q_1 を第 1 折れ点 AK_1 で除した値を計算し、正側の正方向スリップ最大変位 u_{slip1} にセットする。また、 u_{slip2} は $-u_{slip1}$ としてセットする。
 101. 後に計算で必要となる $u1$ と $u2$ を求めてセットしておく。