USER MANUAL

OF

USER MATERIAL SUBROUTINE (UMAT)

FOR

A NEW COUPLED MATERIAL MODEL FOR WOVEN FABRICS, INCLUDING A COUPLED NON-ORTHOGONAL HYPOELASTIC CONSTITUTIVE MODEL INTEGRATED WITH A NEW WRINKLING CRITERION

By:

M. Haghi Kashani[1], M. Hejazi[2], A. Hosseini[2,3], F. Sassani[2], F. Ko[3], A.S. Milani[1,*]

[1] Composites Research Network-Okanagan Laboratory, School of Engineering, University of British Columbia, Kelowna, BC, Canada

[2] Department of Mechanical Engineering, University of British Columbia, Vancouver, Canada

[3] Advanced Fibrous Materials Laboratory, University of British Columbia, Vancouver, Canada

*Corresponding Author:abbas.milani@ubc.ca,

Submitted to 32nd ASC Technical Conference, Purdue University

Oct 2017

1. Introduction

This manual contains review and evaluation of an ABAQUS User Subroutine UMAT_Woven Fabric. UMAT_Woven Fabric is a Fortran based code for the purpose of the numerical implementation of a new material model for woven fabrics in ABAQUS. This version of UMAT_Woven Fabric is based on the material model presented in the paper submitted to ASC 2017 Conference (*Toward Enhanced Forming Simulation of Woven Fabrics using a Coupled Non-Orthogonal Hypoelastic Constitutive Model, Integrated with a New Wrinkling Onset Criterion*).

Section 2 of this manual provides context for the subroutine function, input and output parameters. Furthermore, an example is provided in Section 3 of this user manual to illustrate the effectiveness and validity of the newly developed UMAT compared to currently available material model for woven fabrics in ABAQUs. Eventually, the potential capabilities of the subroutine are discussed in Section 3.

2. Description of the Subroutine

2.1 Overview

One of the advantages of woven fabrics over unidirectional prepregs is their superior formability thanks to the large shear deformation capability. There exists, however, a limit on the shear deformation of woven fabrics, namely the wrinkling. Applying tension to delay wrinkling during forming processes, a consequence of the inherent coupling in woven fabrics, is widely known to the industry. Yet, inherent coupling – change in the effective material properties of a given direction of the fabric due to the applied deformation in other directions - has not been fully understood and implemented in the forming simulations of fabric reinforcements. Coupling should be incorporated in numerical optimization routines to accurately predict the deformation of the material under complex forming set-ups, and more importantly to predict a realistic yarn tension level that can suppress wrinkles. Towards this goal, a new coupled non-orthogonal model which predicts not only the stress-strain path, but also the critical point (shear wrinkling) of the woven fabrics should be proposed and implemented in a numerical simulation. The theory of the material model is thoroughly discussed in the submitted article. In summary, a coupled non-orthogonal hypoelastic constitutive model along with a criterion for wrinkling in terms of coupling are offered. To show its application, the model is implemented in ABAQUS via a UMAT code to predict the stress and strain fields as well as the onset of wrinkling under large shear deformations.

The popular user subroutine to study the solid mechanics of woven fabrics is VFabric subroutine, which is deeply explained in ABAQUS (DS Simulia) user manual. In spite of taking non-orthogonality into account, the inherent coupling is ignored in the VFabric subroutine. The code provided here is able to take the inherent coupling into consideration, resulting in more accurate prediction of the response of woven fabrics under forming processes. Moreover, the VFabric subroutine is efficient for dynamic explicit analyses and cannot be used for dynamic implicit and static FE simulations. Although the explicit analysis is more practical for studying forming process; its precision, in particular for wrinkling prediction, is much weaker than implicit analysis. Also, coupled thermal-mechanical analysis to consider the effect of temperature on the mechanical properties of woven fabrics can be performed using the implicit approach. Hence, the presented subroutine has been written for the implicit solutions in ABAQUS. When ABAQUS calls the UMAT_Woven Fabrics, the subroutine is provided with the inputs such as current stress as well as strain components and solution dependent state variables (SDV). The Subroutine undertakes the stress analysis and predicts the response of woven fabrics in the current time increment by providing outputs at the end of the increment. These outputs (stress, strain and SDV) will be used as next increment inputs.

2.2 Definitions of Subroutine Functions, Inputs and Outputs

The subroutine includes four basic modules. In the first module the stress and strain arrays are transformed from the orthogonal global coordinate system to the local non-orthogonal coordinate system. The material stiffness matrix will be determined and assembled in the second module based on local strain array. The third module transfers the local stiffness matrix to the global stiffness matrix. Finally, the last module computes the global stress array and SDVs for next increment.

2.2.1 Subroutine Parameters

The subroutine was written in accordance ABAQUS User Material Manual Guide. The following lines were imported from ABAQUS to provide proper context for UMAT coding:

```
  SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
   1 RPL,DDSDDT,DRPLDE,DRPLDT,
          2STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
        3NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
         4CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,JSTEP,KINC)
C
   INCLUDE 'ABA_PARAM.INC'
C
   CHARACTER*80 CMNAME
   DIMENSION STRESS(NTENS),STATEV(NSTATV),
   1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
         2STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
         3PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
  4 JSTEP(4)
```

We also need to define the dimension of arrays used in the subroutine code (the STRANL and STRESSL are local strain and stress arrays, the rest of the arrays have been defined in the code, Appendix 1):

```
DIMENSION DOBAR(4), DCBAR(4), DOSBAR(1), DCSBAR(1),
  1STRANL(NTENS),STRESSL(NTENS),TRANSF(NTENS,NTENS),
  2 TRANSFT(NTENS,NTENS),
        3DBAR(NTENS,NTENS),DSTRANL(NTENS),DENTAL(NTENS,NTENS),
  4 DDSDDET(NTENS,NTENS), DELTA(1)
```

The coeficint and constant numbers that are used to define the stiffness function are coded as follows:

```
PARAMETER(ONE=1.D0, TWO=2.D0, THREE=3.D0, FOUR=4.D0, FIVE=5.D0,
     1 AII=0.4518D12, BII=3.2748D10, CII=7.3238D8, DII=6.6648.D0,

     ……………….
```

The basic arrays in this subroutine are:

TRANSF- the transformation matrix form local coordinate ot global

TRANFT- the transformation matrix from global coordinates to local

DDSDDET- the local material stifness matrix

STATEV(1)-ϵ_1

STATEV(2)- ϵ_2

STATEV(3)- ϵ_{12}

STATEV(4)- σ_1

STATEV(5)- σ_2

STATEV(6)- σ_{12}

STATEV (NSTATV)- the wrinkeling indicator (1=wrinkled, 0=Normal)

2.2.2 Transformation from global orthogonal coordinate system to local non-orthogonal coordinate system

The details of the transformation between the orthogonal and non-orthogonal coordinate systems are presented in the submitted article. The transformatiom matrix can be written as:

$$T = \begin{bmatrix} cos^2\alpha & cos^2(\alpha+\theta) & 2\cos\alpha\cos(\alpha+\theta) \\ sin^2\alpha & sin^2(\alpha+\theta) & 2\sin\alpha\sin(\alpha+\theta) \\ \sin\alpha\cos\alpha & \sin(\alpha+\theta)\cos(\alpha+\theta) & \sin(2\alpha+\theta) \end{bmatrix}$$

This matrix implemented in the code as:

```
TRANSF(1,1)=COS(STRAN(3)/TWO)**TWO
TRANSF(1,2)=SIN(STRAN(3)/TWO)**TWO
TRANSF(1,3)=SIN(STRAN(3))
TRANSF(2,1)=SIN(STRAN(3)/TWO)**TWO
```

```
 TRANSF(2,2)=COS(STRAN(3)/TWO)**TWO
 TRANSF(2,3)=SIN(STRAN(3))
 TRANSF(3,1)=SIN(STRAN(3)/TWO)*COS(STRAN(3)/TWO)
 TRANSF(3,2)=SIN(STRAN(3)/TWO)*COS(STRAN(3)/TWO)
 TRANSF(3,3)=ONE
```

2.2.3 Stiffness functions in the non-orthogonal coordiate system

The coupled stiffness functions in the local non-orthogonal coordinate system are determined based the local strains. For a new woven fabric, a user needs onlt to modify this section based on the characterization results. As a matter of fact, this section of the code is the main body of the code which presents the coupled contitutive model in the nonorthogonal coordinate system.

```
      DOBAR(1)=-AII*STATEV(1)**FIVE+BII*STATEV(1)**FOUR
      1 -CII*STATEV(1)**THREE+DII*STATEV(1)**TWO-EII*STATEV(1)+FII
   IF(STATEV(1).GT.EPSILONII) THEN
    DOBAR(1)=-AII*EPSILONII**FIVE+BII*EPSILONII**FOUR
          1 -CII*EPSILONII**THREE+DII*EPSILONII**TWO-EII*EPSILONII+FII
   END IF
   IF(STATEV(1).LT.ZERO) THEN
    DOBAR(1)=FII
   END IF
…………
.
.
.
.
.
.
.
.
.
C
   IF(STATEV(1).LT.ZERO.OR.STATEV(1).EQ.ZERO) THEN
    IF(STATEV(2).LT.ZERO.OR.STATEV(2).EQ.ZERO) THEN
     DOSBAR(1)=ZPTZS
     IF(STATEV(3).GT.ZPZS)Then
```

```
        DOSBAR(1)=DOSBAR(1)-ZPOZE
      END IF
      DCSBAR(1)=ONE
    END IF
  END IF
C
```

2.2.4 Generating the local stifness matrix

The local stifness matrix is assmebled as follows :

```
        DBAR(1,1)=DOBAR(1)*DCBAR(1)
        DBAR(1,2)=DOBAR(3)*DCBAR(3)
        DBAR(1,3)=ZERO
        DBAR(2,1)=DOBAR(4)*DCBAR(4)
        DBAR(2,2)=DOBAR(2)*DCBAR(2)
        DBAR(2,3)=ZERO
        DBAR(3,1)=ZERO
        DBAR(3,2)=ZERO
        DBAR(3,3)=DCSBAR(1)*DOSBAR(1)
```

2.2.5 Stress Anlysis in Global Coordinate System

After trasnfering stiffness matrix form local to global coordinate system (by using TRANSF array), the stress analysis and updating the state varibales are as follows:

```
 DO K1=1, NTENS
    DO K2=1, NTENS
     DDSDDE(K1,K2)=DDSDDET(K1,K2)
    END DO
  END DO

  DO K1=1, NTENS
   DO K2=1, NTENS
    STRESS(K1)=STRESS(K1)+DDSDDE(K1,K2)*DSTRAN(K2)
   END DO
  END DO
```

Updating SDVs:

```
DO K1=1, NTENS
       STATEV(K1)=STRANL(K1)
       STATEV(NTENS+K1)=STRESSL(K1)
     END DO
```

2.2.6 Wrinkeling prediction

The criterion for prdiction of wrinkling occurrence is developed in the submitted article. The last step of the material subroutine is to predict wrinkling initiation which is coded as:

```
DELTA(1)=ABS(STRESSL(1))**(ONE+HALF)/(ABS(STRANL(1))**(ONE+HALF)+
    1 STRANL(1)**HALF-TANH(ABS(STRANL(1))**HALF))
     IF(STATEV(3).LT.-ACOS(ANGS).OR.STATEV(3).GT.DELTA(1)) THEN
       STATEV(3*NTENS)=ONE
     END IF
              STATEV(NSTATV)=DELTA(1)
```

3. Use of the Subroutine in ABAQUS

To use the UMAT_Woven Fabrics, it is requierd to select the user material in the defining properties section. Afterwards, when it comes to define the job, we should provide the path for the Fortran file of UMAT_Woven Fabrics in the special section of the Job bar.

3.1 An example to demonstarte the advantage of the new subroutine

To prove the effectiveness and validity of the written subroutine, an example is provided below. Firstly, 2% tension in the transverse direction is applied and then tension in the main direction is applied up to 4%. Figure 1 demonstartes the higher accuracy of the coupled model (the new model) compared to the uncoupled model (The currently available woven fabrics subroutine in ABAQUS - VFabric).

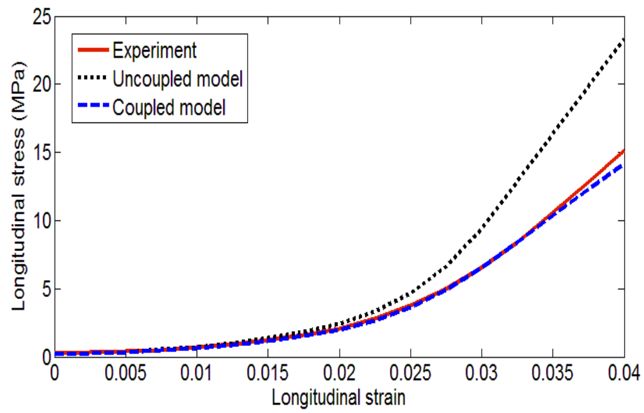

Figure 1. Comparison between the newly dveloped material model for woven fabrics and the currently available woven fabrics model – Vfabric - in ABAQUS.

3.3 Potential Application

The results showed the advantages of the presented UMAT for woven fabrics over the VFabric subroutine which is available in ABAQUS. In fact, the new subroutine predicts the behavior of woven fabrics under forming processes with higher precision in comparison with VFabric subroutine. Moreover, thanks to taking the inherent coupling into account, the manufacturing process parameters such as the applied pressure on the blank holder can be optimized using this new UMAT to prevent wrinkling. Furthermore, since the presented code was written for implicit solutions, coupled thermal-mechanical analyses to capture the effect of temperature on the mechanical properties is also possible. The last, but not least, it can be used for various types of woven fabrics by just modifying the stiffness functions section based on the characterization results of the given woven fabrics.

Appendix 1 : UMAT_WOVEN FABRICS

```
C ****************************************************************
c ****************************************************************
C                  UMAT FOR Woven Fabircs
c                   UMAT_WOVEN FABRICS
c          User Material Subroutine for Coupled Non-Orthogonal
c             Constitutive Modeling of Woven Fabrics
c
c   January, 2017
c   Version  1.1
c ****************************************************************
c   Applications:
c            2-D Planer Stress/Strain
c            Conventional Shell Element S4
c ****************************************************************
c   Authors:
c            Masoud Hejazi & Masoud Haghi Kashani
c            Abbas Hosseini
C             Farrokh Sassani
c            Frank ko
C             Abbas Milani
c
c   Department of Mechanical Engineering,
C    University Of British Columbia, Canada
c
C
C -------------------------------------------------------------
C
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,JSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
     4 JSTEP(4)
c
C LOCAL ARRAYS
C -------------------------------------------------------------
C DOBAR -  PURE TENSILE STIFNESS MODULES
```

```
C DCBAR -  COUPLING TENSILE STIFNESS INDUCED MODULUS
C DOSBAR-  PURE SHEAR STIFNESS MODULES
C DCSBAR-  COUPLING SHEAR STIFNESS INDUCED MODULUS
C TRANSF-  TRANSFORMATION MATRIX
C DBAR-    JACOBIAN NON-ORTHOGONAL MATRIX
C STRANL-  NON-ORTHOGONAL STRAIN ARRAY
C STREESL- NON-ORTHOGONAL STRESS ARRAY
C
      DIMENSION DOBAR(4), DCBAR(4), DOSBAR(1), DCSBAR(1),
     1 STRANL(NTENS), STRESSL(NTENS), TRANSF(NTENS,NTENS),
     2 TRANSFT(NTENS,NTENS),
     3 DBAR(NTENS,NTENS), DSTRANL(NTENS), DENTAL(NTENS,NTENS),
     4 DDSDDET(NTENS,NTENS), DELTA(1)
C
C
      PARAMETER(ONE=1.D0, TWO=2.D0, THREE=3.D0, FOUR=4.D0, FIVE=5.D0,
     1 AII=0.4518D12, BII=3.2748D10, CII=7.3238D8, DII=6.6648D6,
     2 EII=1.3571D4, FII=0.3344D2, GII=27.47D0,
     3 EPSILONII=0.03D0, OPTH=1.3D0, ZPSF=0.83D0, ZPTE=0.28D0,
     4 ZPTF=0.025D0, FIFFOUR=54.D0, TPS=2.6D0, THUND=1200.D0,
     5 ZPZT=0.02D0, THOO=211.D0, ZPTZS=0.1587D0, ZPOZE=0.1071D0,
     6 ZPZS=0.07D0, SVHU=1700.D0, TAH=2.5D0, OFFN=14597.D0,
     7 OPTT=1.22D0, TWTH=12.D3, HALF=0.5D0, OFSZ=1570.D0,
     8 OFSE=15685.D0, OOTF=11353.D0, ZPZOS=0.017D0, ZERO=0.D0,
     9 TOL=1.D-10, ANGS=0.921D0)
C
C TRANSFORMATION MATRIX, ORTHOGONAL TO NONORTHOGONAL
C---------------------------------------------------------------
C
      TRANSF(1,1)=COS(STRAN(3)/TWO)**TWO
      TRANSF(1,2)=SIN(STRAN(3)/TWO)**TWO
      TRANSF(1,3)=SIN(STRAN(3))
      TRANSF(2,1)=SIN(STRAN(3)/TWO)**TWO
      TRANSF(2,2)=COS(STRAN(3)/TWO)**TWO
      TRANSF(2,3)=SIN(STRAN(3))
      TRANSF(3,1)=SIN(STRAN(3)/TWO)*COS(STRAN(3)/TWO)
      TRANSF(3,2)=SIN(STRAN(3)/TWO)*COS(STRAN(3)/TWO)
      TRANSF(3,3)=ONE


      DO K1=1, NTENS
       DO K2=1, NTENS
       TRANSFT(K1,K2)=TRANSF(K2,K1)
       END DO
      END DO
C
```

```
C TRANSFORMING TO NONORTHOGONAL STRANL AND DSTRANL
C -----------------------------------------------------------
C
      DO K1=1, NTENS
        STRANL(K1)=ZERO
        DSTRANL(K1)=ZERO
      END DO

      DO K1=1, NTENS
        DO K2=1, NTENS
          STRANL(K1)=STRANL(K1)+TRANSFT(K1,K2)*(STRAN(K2)+DSTRAN(K2))
        END DO
      END DO

      DO K1=1, NTENS
        DO K2=1, NTENS
          DSTRANL(K1)=DSTRANL(K1)+TRANSFT(K1,K2)*DSTRAN(K2)
        END DO
      END DO
C
C
C------------------------------------------------------------
C -----------------------------------------------------------
C -----------------------------------------------------------
C FUNCTIONS DOBAR DCBAR DOSBAR DC
C      -----------------------------------------------
C DOBAR..
C -------------------------------------------------------------
       DOBAR(1)=-AII*STATEV(1)**FIVE+BII*STATEV(1)**FOUR
     1 -CII*STATEV(1)**THREE+DII*STATEV(1)**TWO-EII*STATEV(1)+FII
      IF(STATEV(1).GT.EPSILONII) THEN
        DOBAR(1)=-AII*EPSILONII**FIVE+BII*EPSILONII**FOUR
     1 -CII*EPSILONII**THREE+DII*EPSILONII**TWO-EII*EPSILONII+FII
      END IF
      IF(STATEV(1).LT.ZERO) THEN
        DOBAR(1)=FII
      END IF

C
      DOBAR(2)=-AII*STATEV(2)**FIVE+BII*STATEV(2)**FOUR
     1 -CII*STATEV(2)**THREE+DII*STATEV(2)**TWO-EII*STATEV(2)+FII
      IF(STATEV(2).GT.EPSILONII) THEN
        DOBAR(2)=-AII*EPSILONII**FIVE+BII*EPSILONII**FOUR
     1 -CII*EPSILONII**THREE+DII*EPSILONII**TWO-EII*EPSILONII+FII
      END IF
      IF(STATEV(2).LT.ZERO) THEN
```

```
   DOBAR(2)=FII
  END IF

C
  DOBAR(3)=(TPS+THUND*STATEV(2)-THUND*(STATEV(2)-ZPZT))/OPTH
  IF(STATEV(2).LT.ZPZT) THEN
   DOBAR(3)=(TPS+THUND*STATEV(2))/OPTH
  END IF
  IF(STATEV(2).LT.ZERO) THEN
   DOBAR(3)=TPS
  END IF
C
  DOBAR(4)=(TPS+THUND*STATEV(1)-THUND*(STATEV(1)-ZPZT))/OPTH
  IF(STATEV(1).LT.ZPZT) THEN
   DOBAR(4)=(TPS+THUND*STATEV(1))/OPTH
  END IF
  IF(STATEV(1).LT.ZERO) THEN
   DOBAR(4)=TPS
  END IF
C ---------------------------------------------------------------
C
C DCBAR
C ---------------------------------------------------------------
  DCBAR(1)=ONE-GII*(STATEV(2)+TOL)**OPTH
  IF(STATEV(1).GT.ZPTF) THEN
   DCBAR(1)=DCBAR(1)-ZPSF*(STATEV(2)+TOL)**ZPTE
  END IF
  IF(STATEV(2).GT.ZPTF) THEN
   IF(STATEV(1).LT.EPSILONII.OR.STATEV(1).EQ.EPSILONII) THEN
    DCBAR(1)=DCBAR(1)+(TWO-FIFFOUR*STATEV(1))
   END IF
   IF(STATEV(1).GT.EPSILONII) THEN
   DCBAR(1)=DCBAR(1)+(TWO-FIFFOUR*EPSILONII)
   END IF
  END IF


  DCBAR(2)=ONE-GII*(STATEV(1)+TOL)**OPTH
  IF(STATEV(2).GT.ZPTF) THEN
   DCBAR(2)=DCBAR(2)-ZPSF*(STATEV(1)+TOL)**ZPTE
  END IF
  IF(STATEV(1).GT.ZPTF) THEN
   IF(STATEV(2).LT.EPSILONII.OR.STATEV(2).EQ.EPSILONII) THEN
    DCBAR(2)=DCBAR(2)+(TWO-FIFFOUR*STATEV(2))
   END IF
   IF(STATEV(2).GT.EPSILONII) THEN
```

```
        DCBAR(2)=DCBAR(2)+(TWO-FIFFOUR*EPSILONII)
        END IF
      END IF

      DCBAR(3)=ONE+THOO*STATEV(1)
C
      DCBAR(4)=ONE+THOO*STATEV(2)

      IF(STATEV(2).LT.ZERO.OR.STATEV(1).LT.ZERO) THEN
       DO K1=1, 4
         DCBAR(K1)=ONE
       END DO
      END IF
C ----------------------------------------------------------------
C ----------------------------------------------------------------
C
C DOSBAR AND DCSBAR
C ----------------------------------------------------------------
      DOSBAR(1)=ZPTZS
C
      DCSBAR(1)=ONE+SVHU*(STATEV(1)+STATEV(2))
      If(STATEV(1).GT.ZPTF) THEN
        DCSBAR(1)=DCSBAR(1)+OFFN*(STATEV(1)-ZPTF)**OPTT
      END IF
C
      IF(STATEV(2).GT.ZPTF) THEN
        DCSBAR(1)=DCSBAR(1)+OFFN*(STATEV(2)-ZPTF)**OPTT
C
        IF(STATEV(1).GT.ZPTF) THEN
          DCSBAR(1)=DCSBAR(1)
     1    +TWTH*((STATEV(1)-ZPTF)*(STATEV(2)-ZPTF))**HALF
        END IF
C
      END IF
C
C
      IF(STATEV(3).GT.ZPZOS) THEN
        DCSBAR(1)=DCSBAR(1)-OFSZ*(STATEV(1)+STATEV(2))
C
        If(STATEV(1).GT.ZPTF) THEN
          DCSBAR(1)=DCSBAR(1)-OFSE*(STATEV(1)-ZPTF)**OPTH
        END IF
C
        IF(STATEV(2).GT.ZPTF) THEN
          DCSBAR(1)=DCSBAR(1)-OFSE*(STATEV(2)-ZPTF)**OPTH
C
```

```
          IF(STATEV(1).GT.ZPTF) THEN
            DCSBAR(1)=DCSBAR(1)
     1       -OOTF*((STATEV(1)-ZPTF)*(STATEV(2)-ZPTF))**HALF
C
          END IF
        END IF
      END IF
C
      IF(STATEV(1).LT.ZERO.OR.STATEV(1).EQ.ZERO) THEN
        IF(STATEV(2).LT.ZERO.OR.STATEV(2).EQ.ZERO) THEN
          DOSBAR(1)=ZPTZS
          IF(STATEV(3).GT.ZPZS)Then
            DOSBAR(1)=DOSBAR(1)-ZPOZE
          END IF
          DCSBAR(1)=ONE
        END IF
      END IF
C

C ----------------------------------------------------------------------
C ----------------------------------------------------------------------
C
C
C ASSEMBELING NONORTHOGONAL JACOBIAN MATRIX
C ----------------------------------------------------------------------
      DBAR(1,1)=DOBAR(1)*DCBAR(1)
      DBAR(1,2)=DOBAR(3)*DCBAR(3)
      DBAR(1,3)=ZERO
      DBAR(2,1)=DOBAR(4)*DCBAR(4)
      DBAR(2,2)=DOBAR(2)*DCBAR(2)
      DBAR(2,3)=ZERO
      DBAR(3,1)=ZERO
      DBAR(3,2)=ZERO
      DBAR(3,3)=DCSBAR(1)*DOSBAR(1)

C-----------------------------------------------------------------------
C ----------------------------------------------------------------------
C ----------------------------------------------------------------------

C CALCULATING ORTHOGONAL JACOBIAN MATRIX AND STRESS
C -------------------------------------------------------------------
C
      DO K1=1, NTENS
c        STRESS(K1)=ZERO
        DO K2=1, NTENS
         DDSDDET(K1,K2)=ZERO
```

```
        DENTAL(K1,K2)=ZERO
       END DO
      END DO



      DO K1=1, NTENS
       DO K2=1, NTENS
        DO K3=1, NTENS
         DENTAL(K1,K2)=DENTAL(K1,K2)+TRANSF(K1,K3)*DBAR(K3,K2)
        END DO
       END DO
      END DO


      DO K1=1, NTENS
       DO K2=1, NTENS
        DO K3=1, NTENS
         DDSDDET(K1,K2)=DDSDDET(K1,K2)+DENTAL(K1,K3)*TRANSFT(K3,K2)
        END DO
       END DO
      END DO

       DO K1=1, NTENS
       DO K2=1, NTENS
        DDSDDE(K1,K2)=DDSDDET(K1,K2)
       END DO
      END DO

      DO K1=1, NTENS
       DO K2=1, NTENS
        STRESS(K1)=STRESS(K1)+DDSDDE(K1,K2)*DSTRAN(K2)
       END DO
      END DO
C CALCULATING NONORTHOGONAL STRESSES
C-------------------------------------------------------------------------
      DO K1=1, NTENS
       STRESSL(K1)=ZERO
      END DO
      DO K1=1, NTENS
       DO K2=1, NTENS
        STRESSL(K1)=STRESSL(K1)+TRANSFT(K1,K2)*STRESS(K2)
       END DO
      END DO
C------------------------------------------------------------------------
C------------------------------------------------------------------------
```

```
C
      DO K1=1, NTENS
        STATEV(K1)=STRANL(K1)
        STATEV(NTENS+K1)=STRESSL(K1)
      END DO
C WRINKLE PREDECTION
C-----------------------------------------------------------------------
      DELTA(1)=ABS(STRESSL(1))**(ONE+HALF)/(ABS(STRANL(1))**(ONE+HALF)+
     1 STRANL(1)**HALF-TANH(ABS(STRANL(1))**HALF))
      IF(STATEV(3).LT.-ACOS(ANGS).OR.STATEV(3).GT.DELTA(1)) THEN
        STATEV(3*NTENS)=ONE
      END IF
      STATEV(NSTATV)=DELTA(1)
C -----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
      RETURN
      END
```

Appendix 2: The Input File of the Example

```
*Heading
** Job name: picturerame Model name: Model-1
** Generated by: Abaqus/CAE 6.14-2
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Part-1
*Node
      1,           0.,           0.
      2,           1.,           0.
      3,           2.,           0.
      4,           3.,           0.
      5,           4.,           0.
      6,           5.,           0.
      7,           0.,           1.
      8,           1.,           1.
      9,           2.,           1.
     10,           3.,           1.
     11,           4.,           1.
     12,           5.,           1.
     13,           0.,           2.
     14,           1.,           2.
     15,           2.,           2.
     16,           3.,           2.
     17,           4.,           2.
     18,           5.,           2.
```

```
     19,           0.,           3.
     20,           1.,           3.
     21,           2.,           3.
     22,           3.,           3.
     23,           4.,           3.
     24,           5.,           3.
     25,           0.,           4.
     26,           1.,           4.
     27,           2.,           4.
     28,           3.,           4.
     29,           4.,           4.
     30,           5.,           4.
     31,           0.,           5.
     32,           1.,           5.
     33,           2.,           5.
     34,           3.,           5.
     35,           4.,           5.
     36,           5.,           5.
*Element, type=CPS4
 1,  1,  2,  8,  7
 2,  2,  3,  9,  8
 3,  3,  4, 10,  9
 4,  4,  5, 11, 10
 5,  5,  6, 12, 11
 6,  7,  8, 14, 13
 7,  8,  9, 15, 14
 8,  9, 10, 16, 15
 9, 10, 11, 17, 16
```

```
10, 11, 12, 18, 17
11, 13, 14, 20, 19
12, 14, 15, 21, 20
13, 15, 16, 22, 21
14, 16, 17, 23, 22
15, 17, 18, 24, 23
16, 19, 20, 26, 25
17, 20, 21, 27, 26
18, 21, 22, 28, 27
19, 22, 23, 29, 28
20, 23, 24, 30, 29
21, 25, 26, 32, 31
22, 26, 27, 33, 32
23, 27, 28, 34, 33
24, 28, 29, 35, 34
25, 29, 30, 36, 35
*Nset, nset=Set-1, generate
  1,  36,   1
*Elset, elset=Set-1, generate
  1,  25,   1
** Section: Section-1
*Solid Section, elset=Set-1, material=Material-1
1.3,
*End Part
**
**
** ASSEMBLY
**
```

```
*Assembly, name=Assembly
**
*Instance, name=Part-1-1, part=Part-1
*End Instance
**
*Nset, nset=Set-1, instance=Part-1-1, generate
  1,  31,   6
*Elset, elset=Set-1, instance=Part-1-1, generate
  1,  21,   5
*Nset, nset=Set-2, instance=Part-1-1, generate
 1,  6,  1
*Elset, elset=Set-2, instance=Part-1-1, generate
 1,  5,  1
*Nset, nset=Set-3, instance=Part-1-1, generate
  6,  36,   6
*Elset, elset=Set-3, instance=Part-1-1, generate
  5,  25,   5
*Nset, nset=Set-4, instance=Part-1-1, generate
 31,  36,   1
*Elset, elset=Set-4, instance=Part-1-1, generate
 21,  25,   1
*Nset, nset=Set-5, instance=Part-1-1
 1,
*End Assembly
**
** MATERIALS
**
*Material, name=Material-1
```

```
*Depvar
    30,
*User Material, constants=1
1.,
** ----------------------------------------------------------------
**
** STEP: Step-1
**
*Step, name=Step-1, nlgeom=NO
*Static
0.0625, 1., 1e-05, 0.0625
**
** BOUNDARY CONDITIONS
**
** Name: BC-3 Type: Displacement/Rotation
*Boundary
Set-3, 2, 2, 1.
** Name: BC-4 Type: Displacement/Rotation
*Boundary
Set-4, 1, 1, 1.
** Name: BC-5 Type: Displacement/Rotation
*Boundary
Set-5, 1, 1
Set-5, 2, 2
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
```

```
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-3
**
*Output, history
*Element Output, elset=Part-1-1.Set-1
E11, E12, E22
**
** HISTORY OUTPUT: H-Output-2
**
*Element Output, elset=Part-1-1.Set-1
S11, S12, S22, SDV
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step
```