Example nº 18
Cable – Stayed Bridge Construction (Example 3)

CivilFEM Manual of Advanced Examples Example nº 18 – Table of Contents

18 EXAMPLE Nº 18: CABLE – STAYED BRIDGE CONSTRUCTION (Example 3)

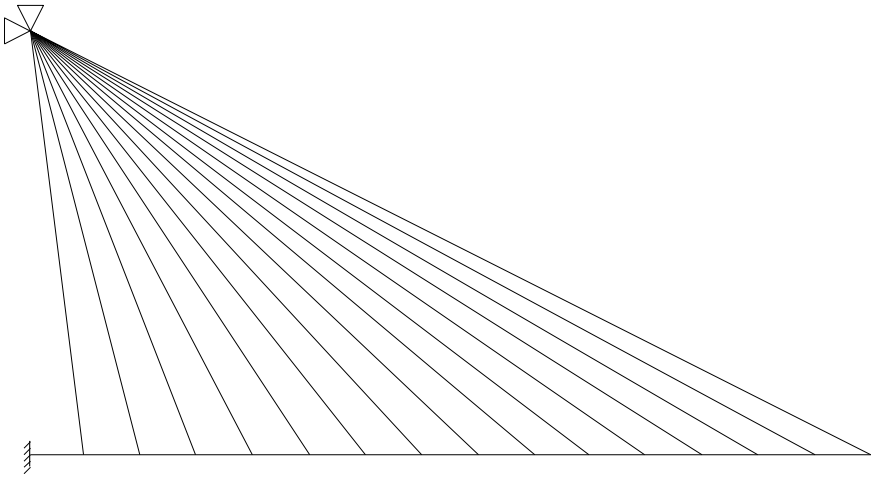
18.1 AIM

The aim of this example is to demonstrate how to determine the initial cable forces in a cable-stayed bridge following the steps of the construction. The initial condition for the cable force calculation is that the desired shape of the deck is horizontal (zero vertical deflection).

18.2 DESCRIPTION OF THE EXAMPLE

This example is based on Example 17. The only difference is that cable sagging is considered using an equivalent modulus of elasticity for the cable elements. This equivalent modulus of elasticity depends on the length and the weight of the cable and on the cable force.

The figure below is a sketch of the bridge model.



DATA:

GEOMETRY

H=45	hight of tower
L=2*H	length of deck
NCables=15	number of cables

ACable=20E-4	cable area
XDistCable=L/NCables	distance between two cables

CONCRETE

HA-40 (EHE)

CABLE STEEL

Y1860S7 (EHE)

CALCULATION STEPS:

1. New section of the deck is added (brought alive) to the structure.
2. Gravity is applied and the model solved.
3. A new cable is added (brought alive) to the structure.
4. Iterative procedure to determine the cable forces:
    - Initial cable forces (i.e. cable shortening) is equal to the vertical deflection of the nodes where the cables are connected to the deck.
    - Model is solved and the new vertical deflection is compared to the desired shape of the structure.
        - If satisfactory the iteration stops.
        - If not the new vertical displacement added to the cable forces (i.e. cable shortening) and the model is solved again with an updated modulus of elasticity.

18.3 RESULTS TO BE OBTAINED

In each construction phase:

1. Deflection of the structure.
2. Bending moment.
3. Axial force.
4. Cable force (shortening).

18.4 CALCULATION LOG

18.4.1 Introduction

A CivilFEM box bridge section is used to model the bridge deck, therefore the Bridges Module needs to be activated.

For the postprocessing of the results a series of macros are created (the macro file, Example18_macros.inp, needs to be placed in the working directory). For each macro a button is placed in the top toolbar:

START	- Go to the first phase of the construction.
NEXT	- Go to the next phase.
BACK	- Go one phase back.
END	- Go to the final phase of the construction.
UY	- Vertical deflection of the structure.
MZ	- Bending moment about the axis Z.
FX	- Axial force in the deck.
CFORCE	- Cable forces.
CSTROKE	- Applied cable shortening.

For the cables LINK11 elements are used. This element allows the application of shortening as an external load.

Since the modulus of elasticity of each cable is updated during the analysis, each cable has to have its own material properties set defined.

A set of arrays is created to store data during the analysis.

The desired shape of the bridge deck is set to a horizontal line.

When the meshing is done, components of elements that correspond to each construction phase are created, both for the deck and the cables.

In the solution there are two main loops. The first loop corresponds to each construction phase. In each phase a new segment of the deck and a new cable is added to the structure (brought alive). The second loop is the iteration process to calculate the cable forces (shortening) in each construction phase. The iteration procedure is controlled by two parameters (tolerance and max number of iterations) that are set at the beginning of the log file. These solution steps create a large number of load steps. The step number corresponding to the introduction of a new deck segment and to the end of the cable force iteration is stored in order to make the postprocessing of the results simpler.

In the postprocessing, a list of the relevant load steps are first created.

To see the cable forces and cable shortening two element tables must be created, since these results of LINK11 elements are not accessible otherwise.

The buttons in the top toolbar can be used for quick access of the results. Using the buttons to navigate between the load steps the macro not only reads the results but also automatically selects the elements that corresponds to the actual construction phase.

18.4.2 Model generation

```
  FINISH
  ~CFACTIV,NLBR,Y
  ~CFCLEAR,,1
  /TITLE,Cable-Stayed Bridge Construction Process (Example3)

  !use library of macros
  *ULIB,Example18_macros,inp

  *ABBR,START,*USE,TO_START
  *ABBR,NEXT,*USE,NEXT
  *ABBR,BACK,*USE,BACK
  *ABBR,END,*USE,TO_END
  *ABBR,UY,*USE,UY
  *ABBR,MZ,*USE,MOMENTZ
  *ABBR,FX,*USE,FORCEX
  *ABBR,CFORCE,*USE,CFORCE,IStep
  *ABBR,CSTROKE,*USE,CSTROKE,IStep


! Setup
! ------------------------------------------------
    ~UNITS,SI
    *AFUN,DEG

! Parameters
! ------------------------------------------------
    H=45                        !hight of tower
    L=2*H                       !length of deck

    NCables=15                  !number fo cables
    ACable=20E-4                !cable area
    XDistCable=L/NCables        !distance between two cables

    DeckEDiv=5                  !element division number in the deck

    NFases=NCables              !number of construction phases

    !iterations:
    Tolerance=0.01              !tolerance of norm of diplacement vector
    MaxNIter=100                !Max Number of Iteration in each phase

! arrays to store values of DL and UY in each phase of construction
! DL (cable length change - prestressing, LINK11 element allows
!   the application of element length change as external load)
! UY (deck vertical deflection)
! CForce (cable force increment- calculated from cable length change)
! CForceAccum (accumulated cable force)
    *DO,I,1,NFases
        *DIM,DL_%I%,,I
        *DIM,UY_%I%,,I
        *DIM,CForce_%I%,,I
        *DIM,CForceAccum_%I%,,I
    *ENDDO
```

```
! UYDesign (designed shape of the deck)
  *DIM,UYDesign,,NCables    !location of nodes where the cables
                            !  are connected to the deck
  *DO,I,1,NCables
    UYDesign(I)=0           !horizontal line (any other shape can be definied)
  *ENDDO

! View options
  /VIEW,1,,,1
  /VUP,1,Y

! ------------------------------------------------
! PREPROCESSING
! ------------------------------------------------
/PREP7

! Element types
! ------------------------------------------------
  ET,1,BEAM44
  ET,2,LINK11

! Materials
! ------------------------------------------------
!  MPTEMP,1
  ~CFMP,1,LIB,CONCRETE,EHE,HA-40      ! stiffening girder
  !each cable needs its own material defined
  !  since the E modulus is changing
  *DO,I,101,NCables+100
    ~CFMP,I,LIB,PREST,EHE,Y1860S7     ! cables
  *ENDDO

! Bridge cross-sections
! ------------------------------------------------
  ~BRSBOX,1,1,4 ,1.5,2 ,0.2,0.2,0.2,2
  ~BRSMDF,1,BOX,KSYM,,1 ,0,0,0
  ~BRSMDF,1,BOX,WEB,RATS,0.3
  ~BRSMDF,1,BOX,WEB,SLPS,0.15
  ~BRSMDF,1,BOX,WEB,THICK,0.3,2
  ~BRSMDF,1,BOX,WEB,THICK,0.4,3
  ~BRSMDF,1,BOX,WEB,SLOPE,0.5,1
  ~BRSMDF,1,BOX,WEB,DEPTH,1.4,1

! Cross-section
! ------------------------------------------------
  !create cross-section from the bridge cross-section
  ~BRSTOCS,1,1,0,0,0,0                    !stiffening girder


! Beam properties (Real constants)
! ------------------------------------------------
  ~BMSHPRO,1,BEAM,1,1,,,44,1,0            !stiffening girder


! Solid model (KPs and Lines)
! ------------------------------------------------
! Keypoints
    K,1,0,H,0                             !top of tower
    K,2,0,0,0                             !deck starting point

!Lines
    PrevKP=2                              !previous KP of the deck
    *DO,I,1,NCables
        K,,I*XDistCable,0,0
        NewKP=_RETURN                     !new KP of the deck
        L,PrevKP,NewKP                    !line of deck
        CM,L_Deck_%I%,LINE
        LSEL,NONE
```

```
        L,1,NewKP                           !line of cable
        CM,L_Cable_%I%,LINE
        LSEL,NONE
        PrevKP=NewKP
    *ENDDO

! Meshing
! ------------------------------------------------------
! Line divisions for meshing
    LSEL,S,LOC,Y,0                          !deck
    LESIZE,ALL,,,DeckEDiv
    LSEL,INVE                               !cables
    LESIZE,ALL,,,1


! stiffening girder
    TYPE,1
    MAT,1
    REAL,1
    *DO,I,1,NFases
        CMSEL,S,L_Deck_%I%
        LMESH,ALL
        CM,E_Deck_%I%,ELEM              !componenets of the elements in each
                                        !  new step of the construction
        ESEL,NONE
    *ENDDO

! cables
*DIM,CableData,,NCables,3               !to store cable data:
                                        !  ElemID, length, angle (to deck)
    TYPE,2
    *GET,ECable,EX,101                  !initial E modulus
    *GET,NuXYCable,NUXY,101             !Poisson's ration
    *GET,WCable,DENS,101
    WCable=WCable*9.81                  !cable weight per unit length
    *DO,I,1,NCables
        MAT,100+I
        CMSEL,S,L_Cable_%I%
        LSUM
        *GET,LCable,LINE,,LENG
        KCable=ECable*ACable/LCable     !cable stiffness K=EA/L
        R,I+100,KCable                  !real constant
                                        !  different for each cable
        REAL,I+100
        LMESH,ALL
        CM,E_Cable_%I%,ELEM             !componenets of the elements in each
                                        !  new step of the construction
        !fill data vector
        *GET,CableData(I,1),ELEM, ,NUM,MAX
        *GET,CableData(I,2),ELEM,CableData(I,1),LENG
        *GET,LengX,ELEM,CableData(I,1),LPROJ,Y
        CableData(I,3)=ACOS(LengX/CableData(I,2))
        ESEL,NONE
    *ENDDO

    ALLSEL

    !list of nodes where the cables are connected to the deck
    ! the order of nodes follows the numbering of cables (Cable_%I%)
    *DIM,DeckNodes,,NCables
    *DO,I,1,NCables
        DeckNodes(I)=NODE(I*XDistCable,0,0)
    *ENDDO
```

18.4.3 Solution

```
! SOLUTION
!------------------------------------------------------------
/SOLU
    ANTYPE,STATIC
    NLGEOM,ON
    NROPT,FULL
    OUTRES,ALL,ALL

!parameters and arrays to check the solution procedure and
!  provide useful data for postprocessing
    NLS=0                          !parameter to keep track of the Load Step Number
    *DIM,NIter,,NFases             !to store the number of iterations in each fase
    *DIM,NLS_Gravity,,NFases       !to store the Number of Load Step when the
                                   !  gravity is applied
    *DIM,NLS_Prestress,,NFases     !to store the Number of Load Step of the last
                                   !  step of the Iteration
    *DIM,LS_List,,NFases*2         !series of loadstep numbers referring to the two
                                   !  load steps defined above

    SFCUM,PRES,ADD                 !set to add surface loads to previous values
                                   !  (accumulate)

!START of soultion loop
    *DO,IFase,1,NFases

! Boundary condition (fixed nodes)
        DDELE,ALL,ALL
        NSEL,S,LOC,X,0
        D,ALL,ALL,0

        ALLSEL
        EKILL,ALL                          !kill all elements

    ! initial state of the current phase (state at the end of the previous phase)
    ! bring elements alive from the previous phases
        *IF,IFase,GT,1,THEN,
            *DO,ISegment,1,IFase-1
                EALIVE,E_Deck_%ISegment%
                EALIVE,E_Cable_%ISegment%
            *ENDDO
        *ENDIF

    ! stage #1: new section of deck (cantilever - no new cable attached yet)
        EALIVE,E_Deck_%IFase%

        ACEL,,9.81                     !apply gravity
        SOLVE
        NLS=NLS+1
        NLS_Gravity(IFase)=NLS         !store Load Step number for postprocessing

    ! get vertical displacements of deck - UY
        *DO,INode,1,IFase
            UY_%IFase%(INode)=UY(DeckNodes(INode))
        *ENDDO

    ! stage #2: new cable attached
        EALIVE,E_Cable_%IFase%

        RUN=1                          !parameter to control the iteration procedure
        Failed=0                       !1 if the iteration number exeeds MaxNIter

    !START of iteration loop
        *DOWHILE,RUN                   !stops when RUN = 0

        !do iteration to adjust the DL values using a constant UY values
```

```
        !  of each iteration as DL values
        *DO,INode,1,IFase
            DL_%IFase%(INode)=DL_%IFase%(INode)-( UYDesign(INode)-UY_%IFase%(INode)
)
        *ENDDO

        ! solve with calculated cable prestressing (length change)
        !   from all previous phases
        SFEDELE,ALL,ALL,ALL
        *DO,JFase,1,IFase                   !go through the previous phases
            *DO,ICable,1,JFase
                SFE,E_Cable_%ICable%,,PRES,,DL_%JFase%(ICable)
            *ENDDO
        *ENDDO
!update the E modulus and K of the cables
        *DO,ICable,1,IFase
            !get total cable shortening
            *GET,CableDL,ELEM,CableData(ICable,1),PRES,1
            LengX=CableData(ICable,2)*COS(CableData(ICable,3))
            !POSITIVE SIGN for cable tension
            T=-CableDL*ECable*ACable/CableData(ICable,2)
            Sigma=T/ACable
            Temp1=ECable*Wcable*Wcable*LengX*LengX
            Temp2=12*Sigma*Sigma*Sigma
            Temp3=Temp1/Temp2+1
            !E equivalent (tangent modulus)
            ECableNew=ECable/Temp3
            MP,  EX,100+ICable,ECableNew
            !apply new E
            KCableNew=ECableNew*ACable/CableData(ICable,2)
            RMODIF,100+ICable,1,KCableNew
        *ENDDO
        SOLVE
        NLS=NLS+1

        ! get vertical displacements of deck - UY
        *DO,INode,1,IFase
            UY_%IFase%(INode)=UY(DeckNodes(INode))
        *ENDDO

        !calculate norm of the vertical displacement vector
        Norm=0
        *DO,INode,1,IFase
            Norm=Norm + (UYDesign(INode)-UY_%IFase%(INode))**2
        *ENDDO
            Norm=SQRT(Norm)

        !check if the iteration can be stopped
        RUN=RUN+1
        *IF,Norm,LT,Tolerance,THEN     !if the shape of the deck is acceptable
                                       !  -> stop iteration
            NIter(IFase)=RUN-1
            NLS_Prestress(IFase)=NLS   !store Load Step number for postprocessing
            RUN=0
        *ELSEIF,RUN,GT,MaxNIter,THEN   !if there are too many itarations
                                       !  -> stop calculation
            NIter(IFase)=9900000+RUN-1 !use this number to easily identify when
                                       !  the iteration failed
            Failed=IFase
            *EXIT
        *ENDIF

    !END of iteration loop
        *ENDDO

    *IF,Failed,GT,0,THEN
        *EXIT
    *ENDIF
```

```
!END of solution loop
    *ENDDO

FINISH
```

18.4.4 Postprocess

```
! POSTPROCESSING
!------------------------------------------------------
/POST1

/ESHAPE,0

!fill LS_List array
*DO,I,1,NFases
    LS_List(I*2-1)=NLS_Gravity(I)
    LS_List(I*2)=NLS_Prestress(I)
*ENDDO

!element tables
*DO,I,1,NFases*2
    IStep=LS_List(I)
    SET,IStep
    !cable forces
    ETABLE,CF_%IStep%,SMISC, 1
    !applied cable shortening
    ETAB,CST_%IStep%,NMISC, 3
*ENDDO

!display vertical displacements
IFase=NFases*2
IStep=LS_List(IFase)
*USE,UY
```

18.4.5 Macros for postprocessing

File name: Example18_macros.inp

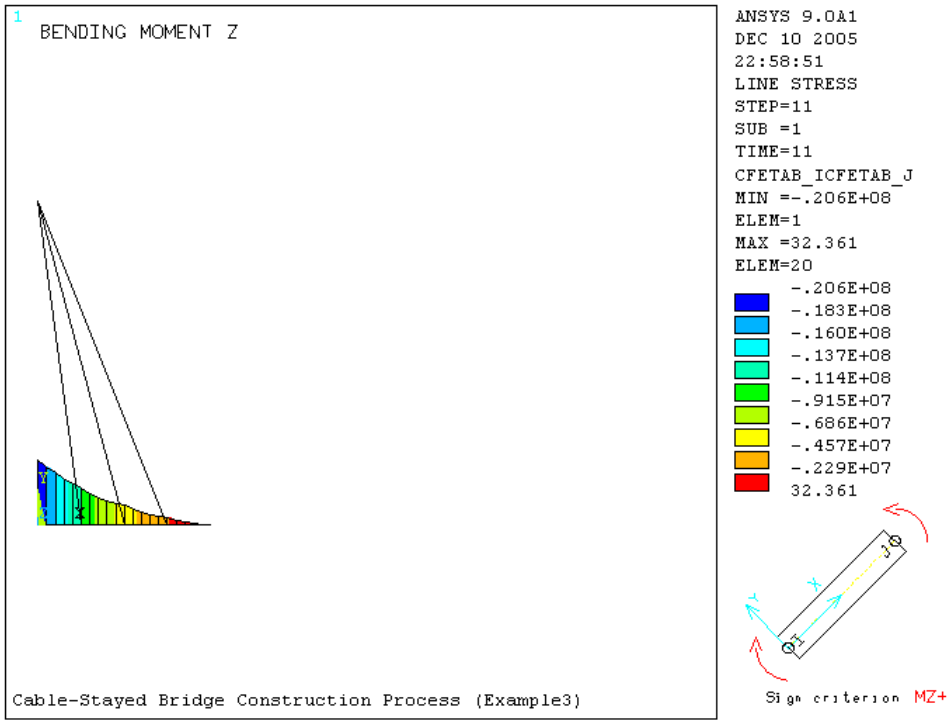
```
TO_START
IFase=1
IStep=LS_List(IFase)
/EOF

TO_END
IFase=NFases*2
IStep=LS_List(IFase)
/EOF

NEXT
*IF,IFase,LT,NFases*2,THEN
    IFase=IFase+1
    IStep=LS_List(IFase)
*ENDIF
/EOF

BACK
*IF,IFase,GT,1,THEN
    IFase=IFase-1
```

```
    IStep=LS_List(IFase)
*ENDIF
/EOF


UY
!vertical deflection
/POST1
*USE,ELEMSEL
SET,IStep
/PLOPTS,INFO,3
/PLOPTS,LEG2,0
/DSCALE,ALL,AUTO
/EFACET,1
PLNSOL, U,Y, 0,1.0
/EOF

MOMENTZ
!bending moment
/POST1
*USE,ELEMSEL
~CFSET,0,IStep,LAST,
/PLOPTS,INFO,2
/PLOPTS,LEG2,0
~PLLSFOR,M,Z,-1,
/EOF

FORCEX
!axial force
/POST1
*USE,ELEMSEL
~CFSET,0,IStep,LAST,
/PLOPTS,INFO,2
/PLOPTS,LEG2,0
~PLLSFOR,F,X,-1,
/EOF

CFORCE
!cable forces
/POST1
*USE,ELEMSEL
ESEL,R,TYPE,,2
/PLOPTS,INFO,3
/PLOPTS,LEG2,0
PLETAB,CF_%ARG1%,NOAV
/EOF

CSTROKE
!applied cable shortening
/POST1
*USE,ELEMSEL
ESEL,R,TYPE,,2
/PLOPTS,INFO,3
/PLOPTS,LEG2,0
PLETAB,CST_%ARG1%,NOAV
/EOF


ELEMSEL
!select elements that are alive during the loadstep
ESEL,NONE
*DO,I,1,IFase
    *IF,MOD(I,2),EQ,1,THEN
        CMSEL,A,E_DECK_%(I+1)/2%
    *ELSE
        CMSEL,A,E_Cable_%I/2%
    *ENDIF
*ENDDO
```

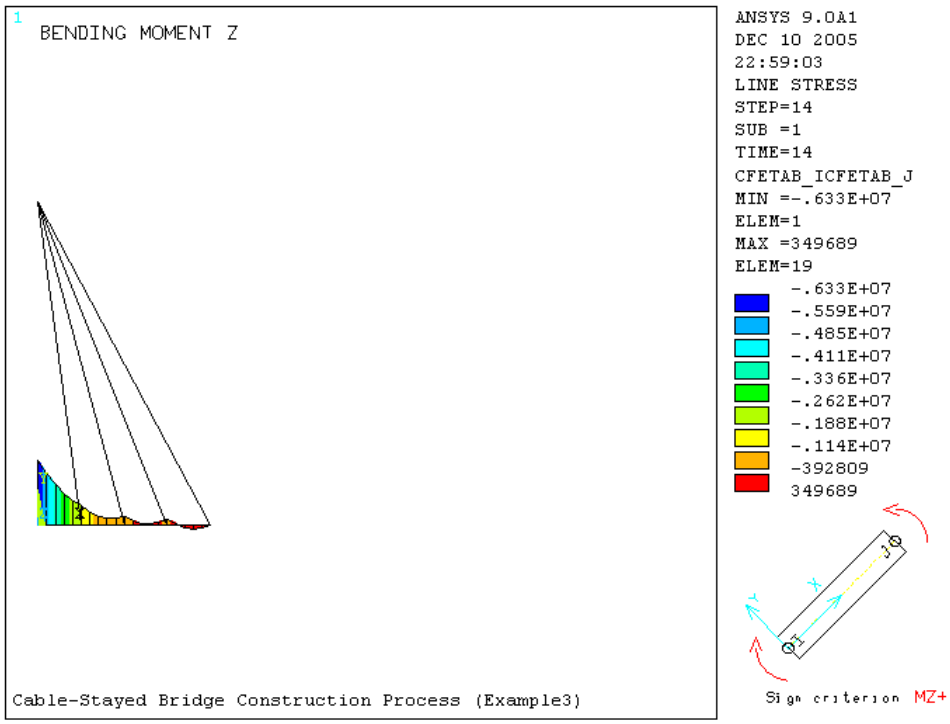
18.5 RESULTS

18.5.1 Bending moment (MZ) in different phases of the construction

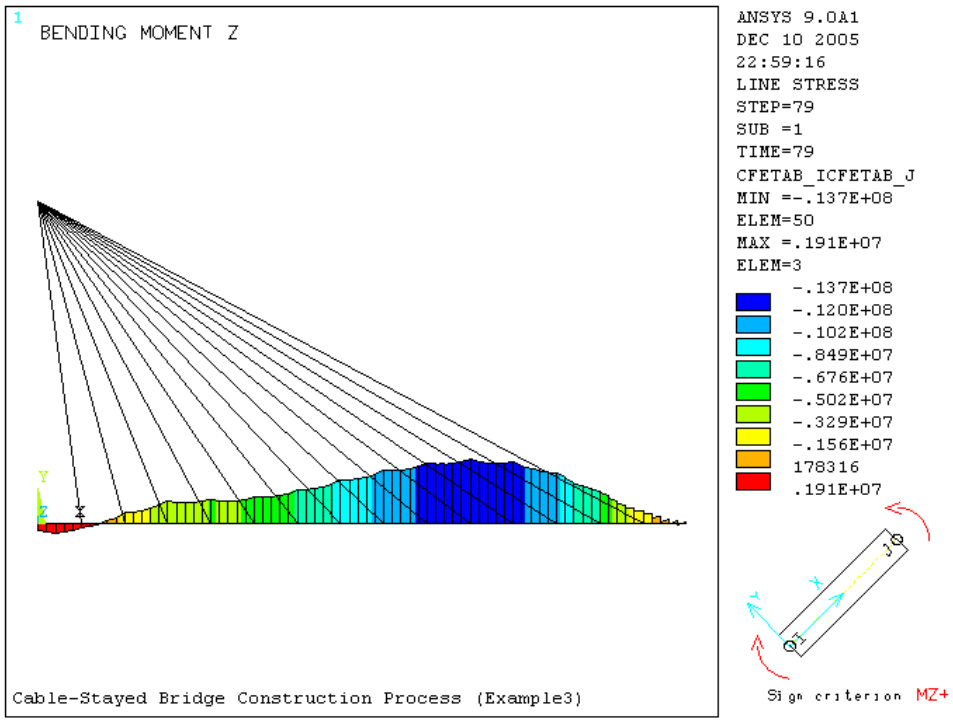
After the 4th deck segment is added:



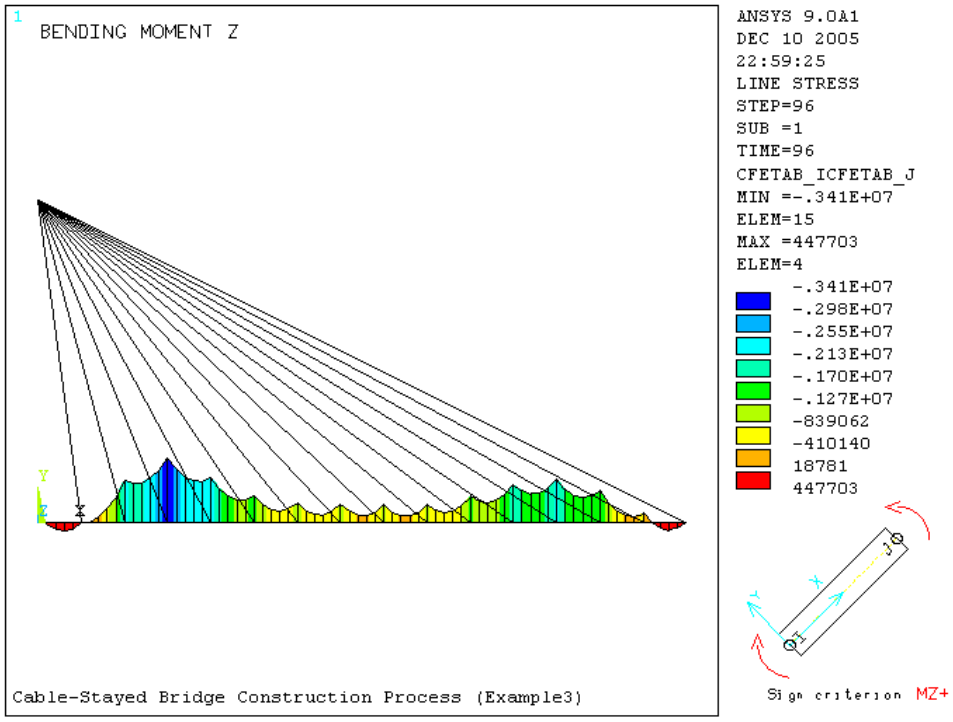
After the 4th cable is added and the cable forces are adjusted:



After the last deck segment is added:

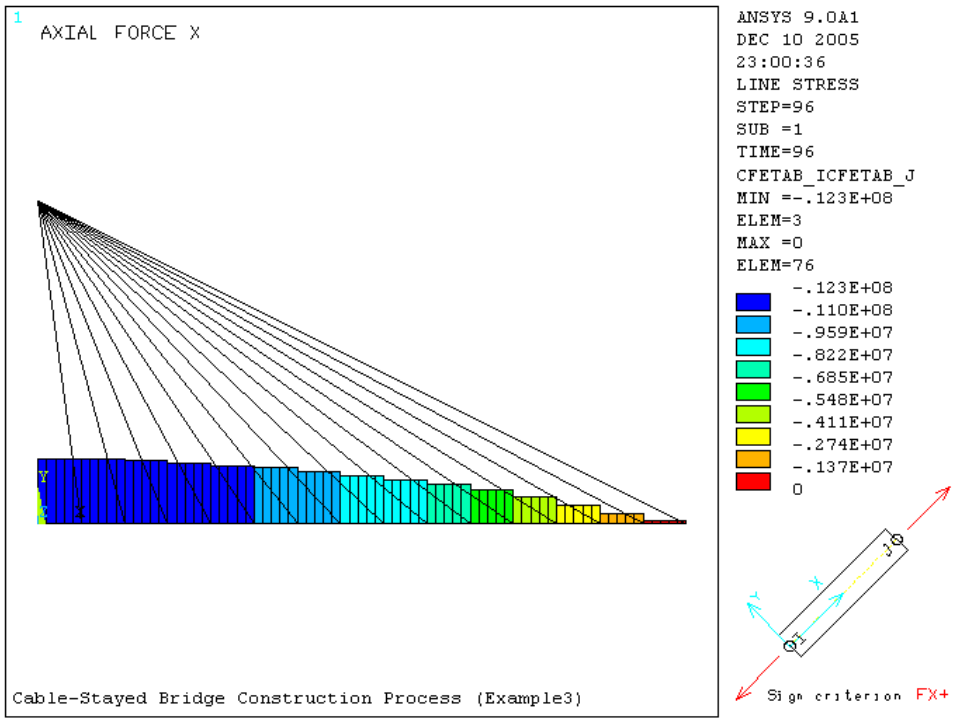


After the last cable is added and the cable forces are adjusted:

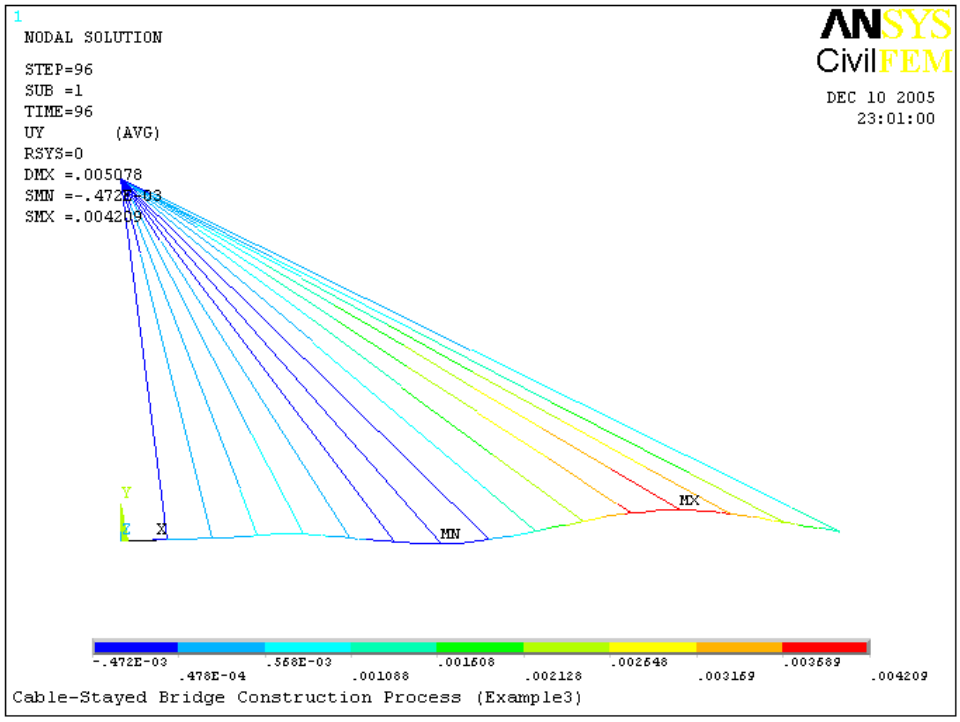


18.5.2 Other results in the final state of the structure

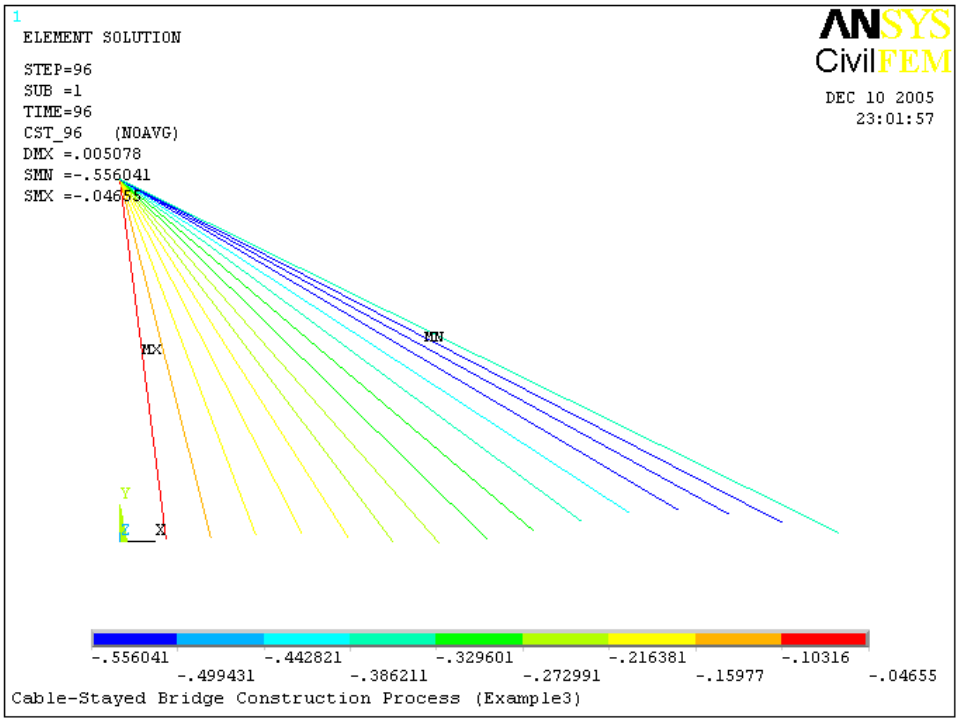
Axial force in the deck:



Vertical deflection of the deck:



Cable stroke (shortening):



Cable force: