

---

***Example nº 20***  
***Cable – Stayed Bridge Construction***  
***(Example 5)***

---

# CivilFEM Manual of Advanced Examples

## Example nº 20 – Table of Contents

- 20 EXAMPLE Nº 20: CABLE – STAYED BRIDGE CONSTRUCTION (Example 5) 1
  - 20.1 AIM ..... 1
  - 20.2 DESCRIPTION OF THE EXAMPLE ..... 1
  - 20.3 RESULTS TO BE OBTAINED ..... 2
  - 20.4 CALCULATION LOG ..... 3
    - 20.4.1 Introduction ..... 3
    - 20.4.2 Model generation ..... 4
    - 20.4.3 Solution 1 ..... 10
    - 20.4.4 Postprocess 1 ..... 10
    - 20.4.5 Solution 2 ..... 11
    - 20.4.6 Postprocess 2 ..... 12
    - 20.4.7 Postprocess 3 (POST26) ..... 13
    - 20.4.8 Macros for postprocessing ..... 13
  - 20.5 RESULTS ..... 16
    - 20.5.1 Bending moment (MZ) in different phases of the construction 16
    - 20.5.2 Other results in the final state of the structure ..... 21
    - 20.5.3 Envelope of bending moment (MZ) ..... 23
    - 20.5.4 Time history result (POST26) ..... 24

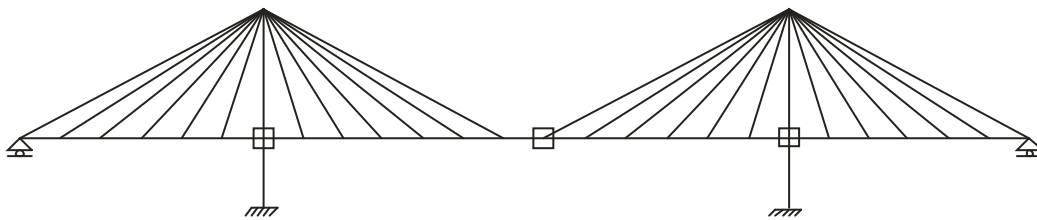
# 20 EXAMPLE N° 20: CABLE – STAYED BRIDGE CONSTRUCTION (Example 5)

## 20.1 AIM

The aim of this example is to demonstrate how to determine the initial cable forces in a cable-stayed bridge. The initial condition for the cable force calculation is that the desired shape of the deck is horizontal (zero vertical deflection) and the towers are vertical (zero horizontal deflection of the tower tops). In the second part the backward construction process analysis is performed on the model. At the right tower the construction is symmetric while at the left tower it is asymmetric.

## 20.2 DESCRIPTION OF THE EXAMPLE

The figure below is a sketch of the bridge model.



The squares represent coupling between the nodes. At the intersections of the deck and the towers all the degrees of freedom are coupled. In the middle of the deck horizontal movement (UX) is allowed and the rest of degrees of freedom are coupled.

### DATA:

#### **GEOMETRY**

HPAbove = 45 m	Height of pylon above deck level
HPBelow = 25 m	Height of pylon below deck level
ACable = 20 cm <sup>2</sup>	Cable area
NCables = 6	Number of cables (one side of a tower)
XDistCable = 14 m	Distance between cables at deck level

#### **CONCRETE**

HA-40 (EHE)

**CABLE STEEL**                      Y1860S7 (EHE)  
**ADDITIONAL DEAD LOAD**      10 kN/m

CALCULATION STEPS:

1. Cable forces are calculated in the final state of the bridge.
  - The deformations of the deck due to the self weight of the structure and additional dead load is calculated.
  - Every cable is prestressed by unit change in the length (-1 m) one by one and the deformations of the deck is calculated for each case.
  - A system of linear equations is assembled:  $Kx = U$ .
    - The response of the structure to the unit cable forces are the elements of the coefficient matrix (K).
    - The deformations (UY of the deck) are the elements of the vector on the right-hand side (U).
  - The system of linear equations is solved  $\rightarrow x$  is determined.
    - The elements of the x vector are the actual cable forces (i.e. cable shortening) to achieve the desired shape of the structure.
  - The x vector is applied as cable forces (i.e. cable shortening) and the structure is solved again.
2. Backward analysis of the construction process ('demolishing' the bridge step by step).
  - Remove additional dead load.
  - Remove the last section of the deck.
  - Loop over the rest of the structure.
    - Remove the cable.
    - Remove segment of the deck.

## 20.3                      RESULTS TO BE OBTAINED

In each construction phase:

1. Deflection of the structure.
2. Bending moments diagram.
3. Axial forces diagram.
4. Cable force (shortening) variation.
5. Envelope of bending moments due to the construction.

## 20.4 CALCULATION LOG

### 20.4.1 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, Example20\_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 (last load step).
+1	- Go to the next phase.
-1	- Go one phase back.
END	- Go to the final phase of the construction (first load step).
UY	- Vertical deflection of the structure.
MZ	- Bending moment about the axis Z.
FX	- Axial force in the deck.
CF	- Cable forces.
CST	- Applied cable shortening.
CF_GRAPH	- Graph showing the variation in the cable forces during the phases of the construction.
ENV_MZ-	- Envelope of negative bending moments.
ENV_MZ+	- Envelope of positive bending moments.

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

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 first solution phase the deflection of the structure due to its self weight and the additional dead load are calculated, and the response of the structure to the unit cable forces (shortening) in each cable is obtained.

In the first postprocessing phase the  $Kx=U$  equation is created and solved.

In the second solution phase gravity is applied, and the additional dead load and the calculated cable forces (shortening) in each cable are placed on the model. This is the final state of the bridge and the start state of the backward construction process analysis. After removing the additional dead load, a step by step demolition of the bridge is done, using the Birth & Death of Elements. On the right side of the bridge the cable and deck segments are “killed” in pairs on both sides of the tower (symmetric construction procedure). On the left side of the bridge the cable and deck segments are “killed”, one-by-one on the two sides of the tower (asymmetric construction procedure).

In the second postprocessing phase the results in the structure are checked for each phase of the construction. To see the cable forces and cable shortening two element tables must be created, since these results of LINK11 elements are not accessible otherwise. To create envelopes of the bending moments the Combination module of CivilFEM is employed.

In the third postprocessing phase the history of the cable force variation is plotted. This needs to be done in the POST26 Time History Postprocessor.

The buttons in the top toolbar can be used for quick access to 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 correspond to the actual construction phase.

## 20.4.2 Model generation

```

FINISH
~CFACTIV,NLBR,Y
~CFCLEAR,,1
/TITLE,Cable-Stayed Bridge Construction Process Example5 (Backward, complete)

!use library of macros
*ULIB,Example20_macros,inp

*ABBR,START,*USE,TO_STEP,NLS
*ABBR,+1,*USE,NEXT
*ABBR,-1,*USE,BACK
*ABBR,END,*USE,TO_STEP,1
*ABBR,RIGHT_END,*USE,TO_STEP,LSRightFirst-1
*ABBR,LEFT_END,*USE,TO_STEP,LSRightLast
*ABBR,UY,*USE,UY
*ABBR,MZ,*USE,MOMENTZ
*ABBR,FX,*USE,FORCEX
*ABBR,CF,*USE,CFORCE,IStep
*ABBR,CST,*USE,CSTROKE,IStep
*ABBR,ENV_MZ-,*USE,ENVELOPE,1
*ABBR,ENV_MZ+,*USE,ENVELOPE,2
*ABBR,CF_+9,*USE,CF_SET,1
*ABBR,CF_-9,*USE,CF_SET,2
*ABBR,CF_GRAPH,*USE,CFORCE_GRAPH

! Setup
! -----
~UNITS,SI

! Parameters
! -----
HPAbove=45                !height of pylon above deck level
HPBelow=25                !height of pylon below deck level

ACable=20E-4              !cable area
NCables=6                 !number of cables
XDistCable=14             !distance between cables at deck level

! Element division numbers
DeckEDiv=3                !deck
PylonEDiv=3               !pylon

! Increment in entity numbering (for the entities of the right side of the bridge)

```

```

NumIncr=1000

! additional dead load
GsLoad=10e3                !10 kN/m

! View options
/VIEW,1,0,0,1
/VUP,1,Y

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

! Element types
! -----
ET,1,BEAM44                !stiffening girder
ET,2,BEAM4                 !pylon
ET,3,LINK11                !cables

! Materials
! -----
~CFMP,1,LIB,CONCRETE,EHE,HA-40    ! pylon, stiffening girder
~CFMP,2,LIB,REINF,EHE,B 500 S
~CFMP,3,LIB,PREST,EHE,Y1860S7    ! cables

! Bridge cross-sections
! -----
~BRSBOX,1,1,4 ,1.5,2 ,0.2,0.2,0.2,2
~BRSMDF,1,BOX,KSYM,,1
~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
~CSECDMS,2,REC,1,5,2.5     !pylon

! Beam properties (Real constants)
! -----
~BMSHPRO,1,BEAM,1,1,,,44,1,0    !stiffening girder
~BMSHPRO,2,BEAM,2,2,,, 4,1,0    !pylon

! Solid model (KPs and Lines)
! -----

! Keypoints
K,1,0,HPAbove,0            !top of left pylon
K,2,0,0,0                 !deck starting point (left pylon)
K,3,0,0,0                 !KP3 is identical to KP2 (KP of the pylon)
K,4,0,-HPBelow,0
XDistPylon=(2*NCables+1)*XDistCable !distance between the two pylons
K,NumIncr+1,XDistPylon,HPAbove,0 !top of left pylon
K,NumIncr+2,XDistPylon,0,0     !deck starting point (left pylon)
K,NumIncr+3,XDistPylon,0,0
K,NumIncr+4,XDistPylon,-HPBelow,0

! Lines

```

```

!left pylon
L,4,3
L,3,1
CM,L_Pylon_Left,LINE
LSEL,NONE

! deck and cables of left pylon
KPStart=2
PrevKPLeft=KPStart           !starting KP of the deck (previous KP)
PrevKPRight=KPStart
*DO,I,1,NCables
  !right
  K,,I*XDistCable,0,0
  NewKP=_RETURN             !new KP of the deck
  L,PrevKPRight,NewKP      !line of deck
  CM,L_Deck_%I*2-1%,LINE
  LSEL,NONE
  L,1,NewKP                 !line of cable
  CM,L_Cable_%I*2-1%,LINE
  LSEL,NONE
  PrevKPRight=NewKP

  !left
  K,,-I*XDistCable,0,0
  NewKP=_RETURN             !new KP of the deck
  L,NewKP,PrevKPLeft       !line of deck
  CM,L_Deck_%I*2%,LINE
  LSEL,NONE
  L,1,NewKP                 !line of cable
  CM,L_Cable_%I*2%,LINE
  LSEL,NONE
  PrevKPLeft=NewKP

*ENDDO

NUMSTR,KP,NumIncr+1
NUMSTR,LINE,NumIncr+1

!right pylon
L,NumIncr+4,NumIncr+3
L,NumIncr+3,NumIncr+1
CM,L_Pylon_Right,LINE
LSEL,NONE

! deck and cables of right pylon
KPStart=NumIncr+2
PrevKPLeft=KPStart         !starting KP of the deck (previous KP)
PrevKPRight=KPStart
*DO,I,1,NCables
  !left
  K,,XDistPylon-I*XDistCable,0,0
  NewKP=_RETURN             !new KP of the deck
  L,NewKP,PrevKPLeft       !line of deck
  CM,L_Deck_%NumIncr+I*2-1%,LINE
  LSEL,NONE
  L,NumIncr+1,NewKP        !line of cable
  CM,L_Cable_%NumIncr+I*2-1%,LINE
  LSEL,NONE
  PrevKPLeft=NewKP

  !right
  K,,XDistPylon+I*XDistCable,0,0
  NewKP=_RETURN             !new KP of the deck
  L,PrevKPRight,NewKP      !line of deck
  CM,L_Deck_%NumIncr+I*2%,LINE
  LSEL,NONE
  L,NumIncr+1,NewKP        !line of cable
  CM,L_Cable_%NumIncr+I*2%,LINE

```

```

LSEL,NONE
PrevKPRight=NewKP

*ENDDO

!closing segment of the deck
NUMSTR,LINE,1

! the segment is NOT connected to the rest of the model
K,,NCables*XDistCable,0,0
KPLeft=_RETURN
K,,XDistPylon-NCables*XDistCable,0,0
KPRight=_RETURN
L,KPLeft,KPRight           !line of deck
CM,L_Deck_Close,LINE

ALLSEL

! Meshing
! -----
! Line divisions for meshing
LSEL,S,LOC,Y,0
LESIZE,ALL,,,DeckEDiv      !stiffening girder
LSEL,INVE
CMSEL,U,L_Pylon_Left
CMSEL,U,L_Pylon_Right
LESIZE,ALL,,,1             !cables
CMSEL,S,L_Pylon_Left
CMSEL,A,L_Pylon_Right
LESIZE,ALL,,,PylonEDiv     !cables

ALLSEL

! stiffening girder
TYPE,1
MAT,1
REAL,1
!of the left pylon
*DO,I,1,NCables*2
  CMSEL,S,L_Deck_%I%
  LMESH,ALL
  CM,E_Deck_%I%,ELEM
  ESEL,NONE
*ENDDO
!component of all the deck elements of the left pylon
ESEL,ALL
CM,E_Deck_Left,ELEM
ESEL,NONE

!of the right pylon
*DO,I,1,NCables*2
  CMSEL,S,L_Deck_%NumIncr+I%
  LMESH,ALL
  CM,E_Deck_%NumIncr+I%,ELEM
  ESEL,NONE
*ENDDO

!component of all the deck elements of the right pylon
ESEL,ALL
CMSEL,U,E_Deck_Left
CM,E_Deck_Right,ELEM
ESEL,NONE

!closing segment
CMSEL,S,L_Deck_Close
LMESH,ALL

```

```

!component of closing segment elements
CM,E_Deck_Close,ELEM

! merge nodes on the left end of the closing segemnt
NSEL,S,LOC,X,NCables*XDistCable
NUMMRG,NODE, , , ,LOW

! on the right side couple the nodes
NSEL,S,LOC,X,XDistPylon-NCables*XDistCable
*GET,N1,NODE,,NUM,MIN
*GET,N2,NODE,,NUM,MAX

CP,NEXT,UY,N1,N2
CP,NEXT,UZ,N1,N2
CP,NEXT,ROTZ,N1,N2
CP,NEXT,ROTX,N1,N2
CP,NEXT,ROTY,N1,N2

!component of all the deck elements
ESEL,ALL
CM,E_Deck_All,ELEM
ESEL,NONE

! cables
TYPE,3
MAT,3
*GET,ECable,EX,3

*DIM,CableData,,NCables*4,3           !to store cable data: ComponentID, ElemID,
DeckNodeID
!of the left pylon
*DO,I,1,NCables*2
  CMSEL,S,L_Cable_%I%
  LSUM
  *GET,LCable,LINE,,LENG
  KCable=ECable*ACable/LCable       !cable stiffness K=EA/L
  R,I+2,KCable                       !real constant - different for each cable
  REAL,I+2
  LMESH,ALL
  CM,E_Cable_%I%,ELEM
  !fill data vector
  CableData(I,1)=I
  *GET,CableData(I,2),ELEM, ,NUM,MAX
  ESEL,NONE
*ENDDO

!component of all the cable elements of the left pylon
ESEL,ALL
CMSEL,U,E_Deck_All
CM,E_Cable_Left,ELEM
ESEL,NONE

!of the right pylon
*DO,I,1,NCables*2
  CMSEL,S,L_Cable_%NumIncr+I%
  LSUM
  *GET,LCable,LINE,,LENG
  KCable=ECable*ACable/LCable       !cable stiffness K=EA/L
  R,I+2,KCable                       !real constant - different for each cable
  REAL,I+2
  LMESH,ALL
  CM,E_Cable_%NumIncr+I%,ELEM
  !fill data vector
  CableData(2*NCables+I,1)=NumIncr+I
  *GET,CableData(2*NCables+I,2),ELEM, ,NUM,MAX
  ESEL,NONE
*ENDDO
!component of all the cable elements of the right pylon

```

```

ESEL,ALL
CMSEL,U,E_Deck_All
CMSEL,U,E_Cable_Left
CM,E_Cable_Right,ELEM

!component of all the cable elements
ESEL,ALL
CMSEL,U,E_Deck_All
CM,E_Cable_All,ELEM
ESEL,NONE

! pylons
TYPE,2
MAT,1
REAL,2
CMSEL,S,L_Pylon_Left
LMESH,ALL
CM,E_Pylon_Left,ELEM
ESEL,NONE
CMSEL,S,L_Pylon_Right
LMESH,ALL
CM,E_Pylon_Right,ELEM
CMSEL,A,E_Pylon_Left
CM,E_Pylon_All,ELEM

ALLSEL

!list of nodes where the cables connects to the deck
*DO,I,1,NCables*4
    ESEL,S,ELEM,,CableData(I,2)
    NSLE,S,ALL
    NSEL,R,LOC,Y,0
    *GET,CableData(I,3),NODE,,NUM,MAX
*ENDDO

ALLSEL

!list of the nodes of the pylon top
*DIM,NPylonTop,,2
NPylonTop(1)=NODE(0,HPAbove,0)
NPylonTop(2)=NODE(XDistPylon,HPAbove,0)

EPLOT

! Boundary conditions
! -----
! pylon base
NSEL,S,LOC,Y,-HPBelow
D,ALL,ALL

! two ends of stiffening girder
NSEL,S,LOC,X,-NCables*XDistCable
NSEL,A,LOC,X,XDistPylon + NCables*XDistCable
CM,N_DeckEnds,NODE
D,ALL,UY

! connection between the pylons and the stiffening girder
! left pylon
NSEL,S,LOC,X,0
NSEL,R,LOC,Y,0
CP,NEXT,UX,ALL
CP,NEXT,UY,ALL
! right pylon
NSEL,S,LOC,X,XDistPylon
NSEL,R,LOC,Y,0
CP,NEXT,UX,ALL
CP,NEXT,UY,ALL

```

ALLSEL

### 20.4.3 Solution 1

```
! SOLUTION
!-----
! 1. small displacement analysis for cable force calculation
!   in the final stage of the bridge

/SOLU
ANTYPE,STATIC
EQSLV,JCG
! an iterative equation solver is used to overcome the bad
! conditioning of the stiffness matrix
OUTRES,ALL,ALL

!additional dead load (Gs)
CMSEL,S,E_Deck_All
SFBEAM,ALL,2,PRES,GsLoad

ALLSEL
ACEL,,9.81,
SOLVE      !LS1

!left
*DO,I,1,NCables*4
    ACEL,,                                !remove gravity effect
    SFEDELE,ALL,ALL,ALL
    SFE,CableData(I,2),,PRES,, -1        !apply unit elongation on cable
    SOLVE
*ENDDO
```

### 20.4.4 Postprocess 1

```
! POSTPROCESSING
!-----
! 1. calculate the cable forces using 'K stiffness matrix'
/POST1

*DIM,KMat,,NCables*4,NCables*4  !K matrix
*DIM,UYVect,,NCables*4         !UY vector - vertical deflection
*DIM,DLInitVect,,NCables*4     !Initial DL vector - cable shortening

!results of LS1
SET,1

!vertical displacement of the pylon at the deck level
UYPylon=UY(NODE(0,0,0))

!get and store the vertical deflection of the deck due to gravity
!compensate for the shortening of the pylon
*DO,I,1,NCables*4
    UYVect(I)=UY(CableData(I,3))-UYPylon
*ENDDO
!the horizontal displacement of the pylon top (store it in the same vector)
*DO,I,1,2
    UYVect(I*NCables*2)=UX(NPylonTop(I))
*ENDDO

!get and store the vertical deflection of the deck due to the unit cable forces
*DO,ICable,1,NCables*4
```

```

SET,1+ICable
*DO,INode,1,NCables*4
    KMat(INode,ICable)=UY(CableData(INode,3))
*ENDDO
!add horizontal displacement of the pylon top to K
!(replace relevant members of the matrix)
*DO,I,1,2
    KMat(I*NCables*2,ICable)=UX(NPylonTop(I))
*ENDDO
*ENDDO

!solve the linear system to get cable forces at the final stage of the construction
*MOPER,DLInitVect,KMat,SOLV,UYVect
    
```

## 20.4.5 Solution 2

```

! SOLUTION
!-----
! 2. the demolishing process
/SOLU
ANTYPE,STATIC
EQSLV,JCG
    ! an iterative equation solver is used to overcome the bad
    ! conditioning of the stiffness matrix
NLGEOM,ON
NROPT,FULL
OUTRES,ALL,ALL

SFEDELE,ALL,ALL,ALL

!modify boundary conditions
! two ends of stiffening girder
D,N_DeckEnds,UY,UYPylon

!additional dead load (Gs)
CMSEL,S,E_Deck_All
SFBEAM,ALL,2,PRES,GsLoad

ALLSEL

*DO,I,1,NCables*4
    SFE,CableData(I,2),,PRES,,DLInitVect(I)
*ENDDO

NLS=0      !counter of load steps
ACEL,,9.81,
!LS1 - final state of the bridge
ESEL,S,LIVE
CM,E_Live_LS_%NLS+1%,ELEM      !create componenet of LIVE elements (same for each LS)
ALLSEL
SOLVE
NLS=NLS+1

!LS2 - remove additonal dead load
SFEDELE,E_Deck_All,ALL,ALL
ESEL,S,LIVE
CM,E_Live_LS_%NLS+1%,ELEM      !create componenet of LIVE elements (same for each LS)
ALLSEL
SOLVE
NLS=NLS+1

!LS3 - remove the closing segment of the bridge deck
! and the verical supports at the two ends
EKILL,E_Deck_Close
    
```

```

ESEL,S,LIVE
CM,E_Live_LS_%NLS+1%,ELEM
ALLSEL
SOLVE
NLS=NLS+1

! two different processes are modelled on the two parts of the bridge:
! - right part: symmetric
! - left part: non-symmetric
!right part
LSRightFirst=NLS+1
*DO,I,NCables*2,1,-2
!kill cable
EKILL,E_Cable_%NumIncr+I%
EKILL,E_Cable_%NumIncr+I-1%
ESEL,S,LIVE
CM,E_Live_LS_%NLS+1%,ELEM
ALLSEL
SOLVE
NLS=NLS+1

!kill deck
EKILL,E_Deck_%NumIncr+I%
EKILL,E_Deck_%NumIncr+I-1%
ESEL,S,LIVE
CM,E_Live_LS_%NLS+1%,ELEM
ALLSEL
SOLVE
NLS=NLS+1
*ENDDO
LSRightLast=NLS

!left part
*DO,I,NCables*2,1,-1
!kill cable
EKILL,E_Cable_%I%
ESEL,S,LIVE
CM,E_Live_LS_%NLS+1%,ELEM
ALLSEL
*IF,I,EQ,1,THEN
    D,NODE(XDistCable,0,0),UY !temporary support
*ENDIF
SOLVE
NLS=NLS+1

!kill deck
EKILL,E_Deck_%I%
ESEL,S,LIVE
CM,E_Live_LS_%NLS+1%,ELEM
ALLSEL
*IF,I,GT,1,THEN
    SOLVE
    NLS=NLS+1
*ENDIF
*ENDDO

```

## 20.4.6 Postprocess 2

```

! POSTPROCESSING
!-----
/POST1

/ESHAPE,0
IStep=1

```

```
!element tables
*DO,I,1,NLS
  SET,I
  !cable forces
  ETABLE,CF_%I%,SMISC, 1
  !applied cable shortening
  ETAB,CST_%I%,NMISC, 3
*ENDDO

!combinations:
~TRGDEF,1,CROSS,M,Z,MIN
~TRGDEF,2,CROSS,M,Z,MAX
/TITLE,Construction
~CMBDEF,1,OPTION,NLS-1,0,1
*DO,I,1,NLS-1
  ~STSTDEF,1,I,LSTEP,I,0,1,1,1,1
*ENDDO
~COMBINE
```

## 20.4.7 Postprocess 3 (POST26)

```
! POSTPROCESSING
!-----
/POST26
NUMVAR,200
! Change of cable forces
*DO,ICable,1,NCables*4
  ESOL,ICable+1,CableData(ICable,2),CableData(ICable,3),SMISC,1,CForce_%ICable%
*ENDDO
```

## 20.4.8 Macros for postprocessing

File name: Example20\_macros.inp

```
TO_STEP
ISTEP=ARG1
/EOF

NEXT
*IF, IStep,GT,1,THEN
  IStep=IStep-1
*ENDIF
/EOF

BACK
*IF, IStep,LT,NLS,THEN
  IStep=IStep+1
*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
~CMBDAT,1
~CFSET,0,IStep,LAST,
/PLOPTS,INFO,2
/PLOPTS,LEG2,0
~PLLSFOR,M,Z,-1,
/EOF

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

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

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

CF_SET
*IF,ARG1,EQ,1,THEN
    *IF,CF_Set*9,LT,NCables*4,THEN
        CF_Set=CF_Set+1
    *ENDIF
*ELSE
    *IF,CF_Set,GT,1,THEN
        CF_Set=CF_Set-1
    *ENDIF
*ENDIF

/EOF

CFORCE_GRAPH
/POST26
ALLSEL
*DO,ICable,1,NCables*4
    ESOL,ICable+1,CableData(ICable,2),CableData(ICable,3),SMISC,1,CForce_%ICable%
*ENDDO
/PLOPTS,INFO,3
Coff=(CF_Set-1)*9

```

```
PLVAR, Coff+2, Coff+3, Coff+4, Coff+5, Coff+6, Coff+7, Coff+8, Coff+9, Coff+10, Coff+11
/EOF

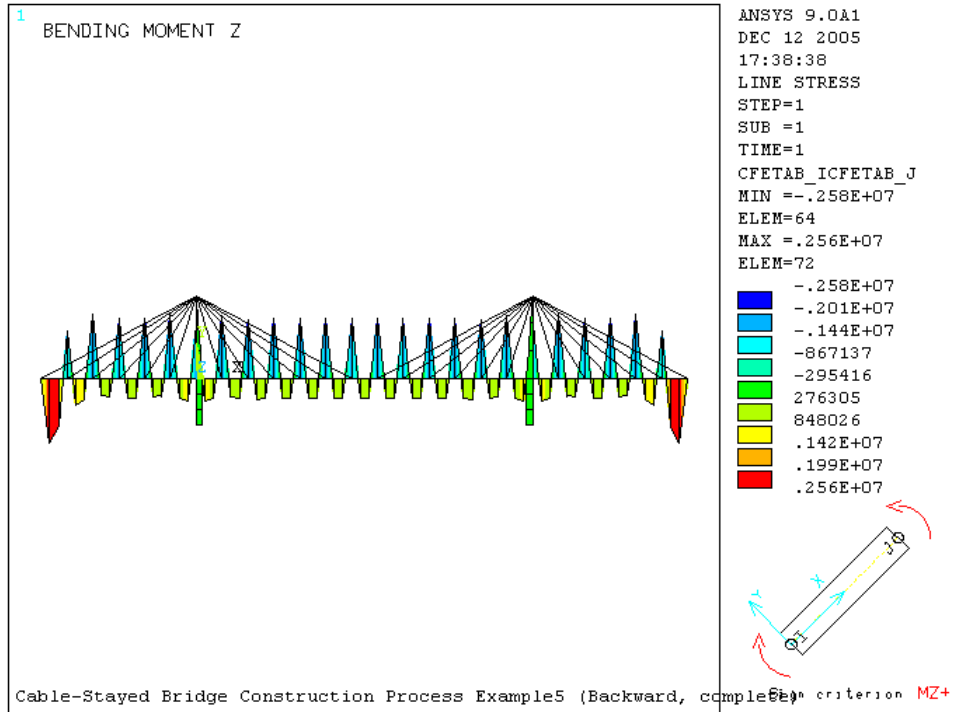
ENVELOPE
/POST1
ALLSEL
~CMBDAT, 2
*IF, ARG1, EQ, 1, THEN
  ~CMB, 1, CROSS, M, Z, MIN
*ELSE
  ~CMB, 1, CROSS, M, Z, MAX
*ENDIF
/PLOPTS, INFO, 2
/PLOPTS, LEG2, 0
~PLLSFOR, M, Z, -1,
/EOF

ELEMSEL
!select elements that are alive during the loadstep
CMSEL, S, E_Live_LS_%ISTEP%
```

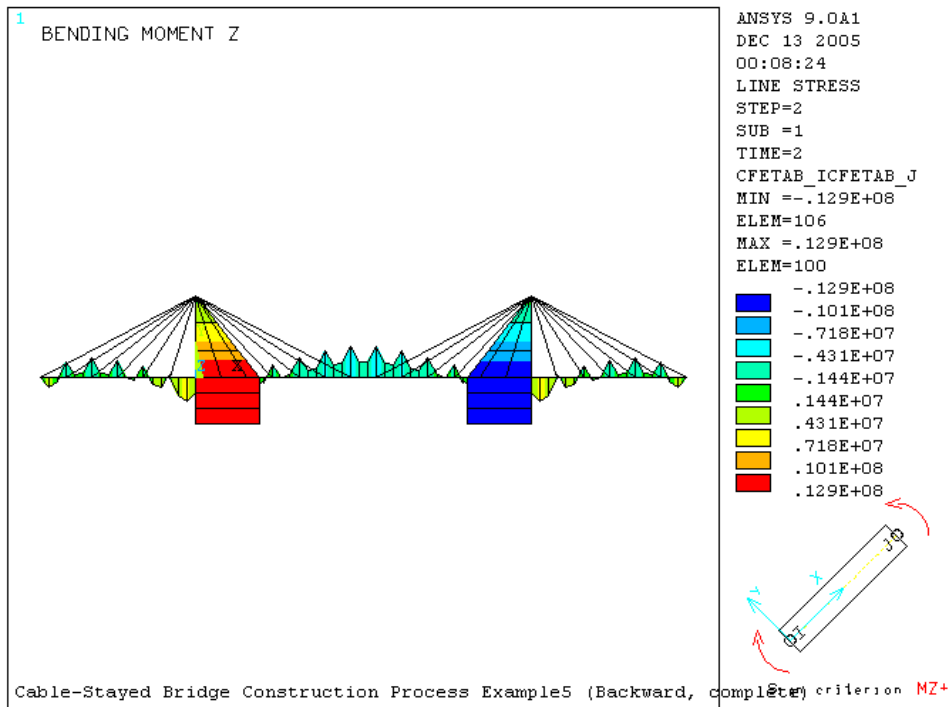
## 20.5 RESULTS

### 20.5.1 Bending moment (MZ) in different phases of the construction

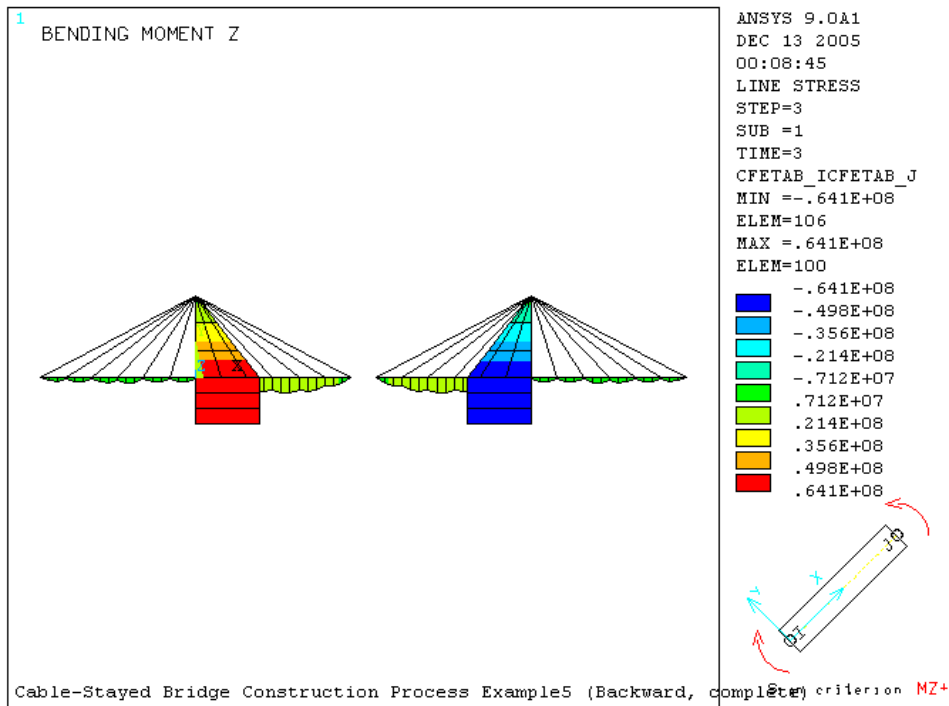
In the final state of the bridge (LS1):



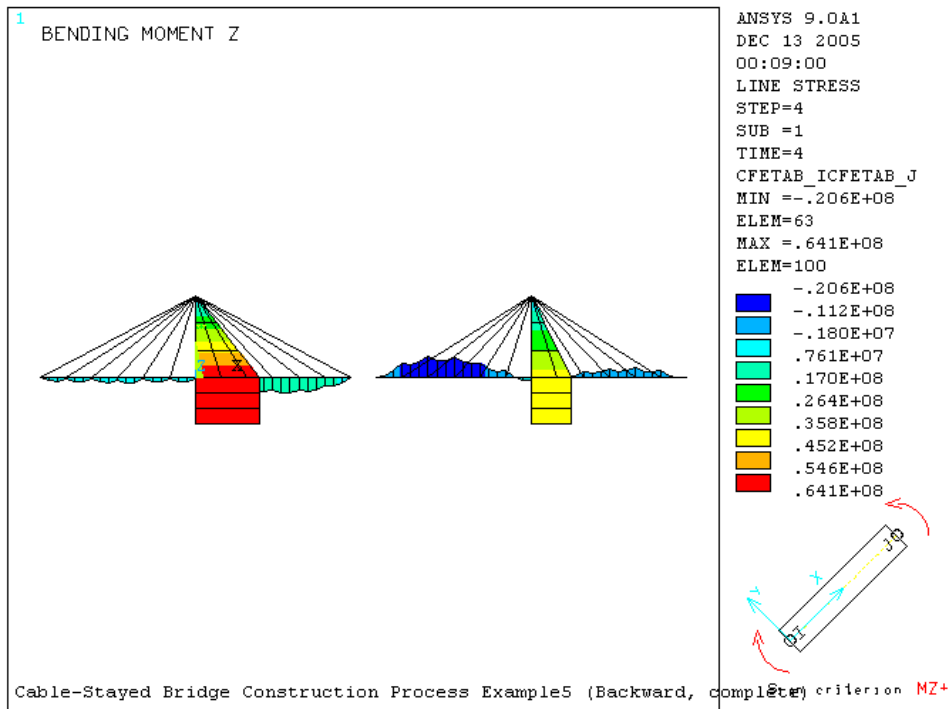
After removing the additional deadload (LS2):



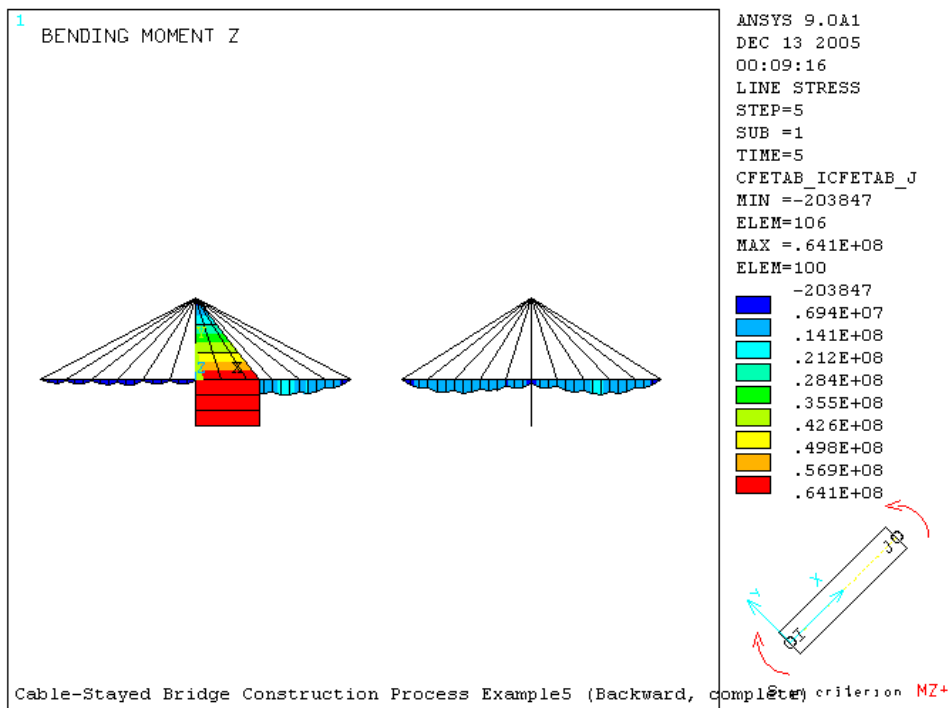
After removing the closing segment of the deck (LS3):



After removing two cables of the right tower (LS4):

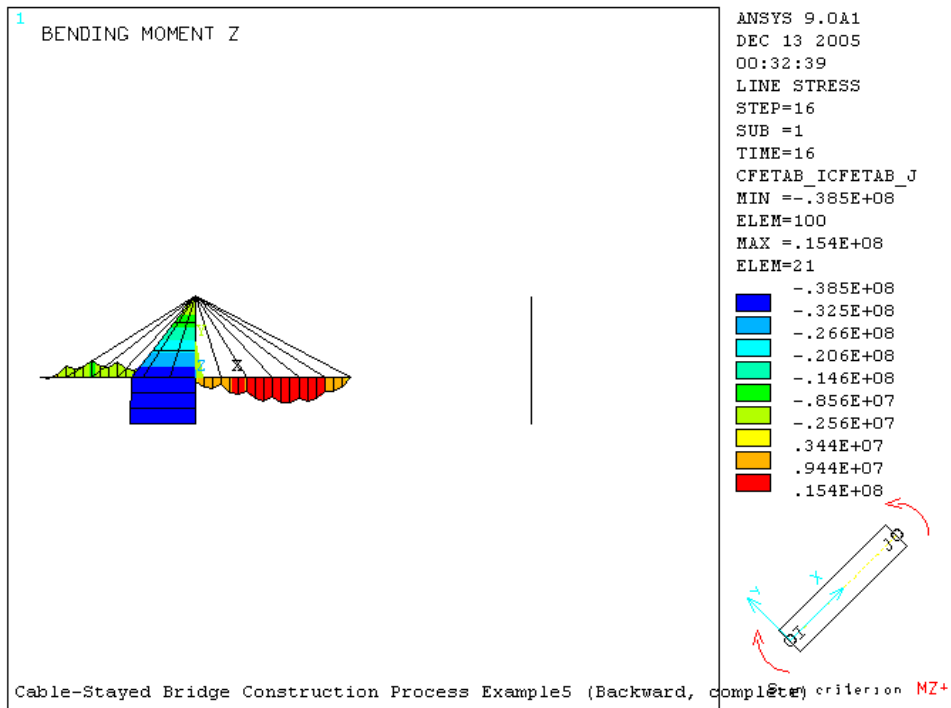


After removing two deck segments of the right tower (LS5):

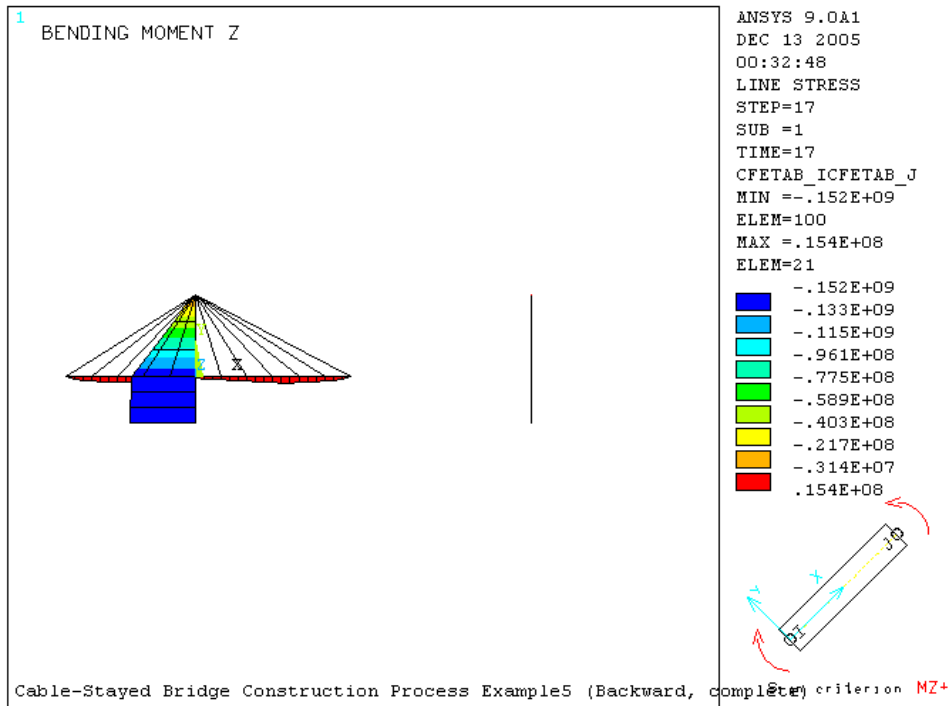


Two following results show the state of the structure after the complete “demolishing” of the right side of the bridge

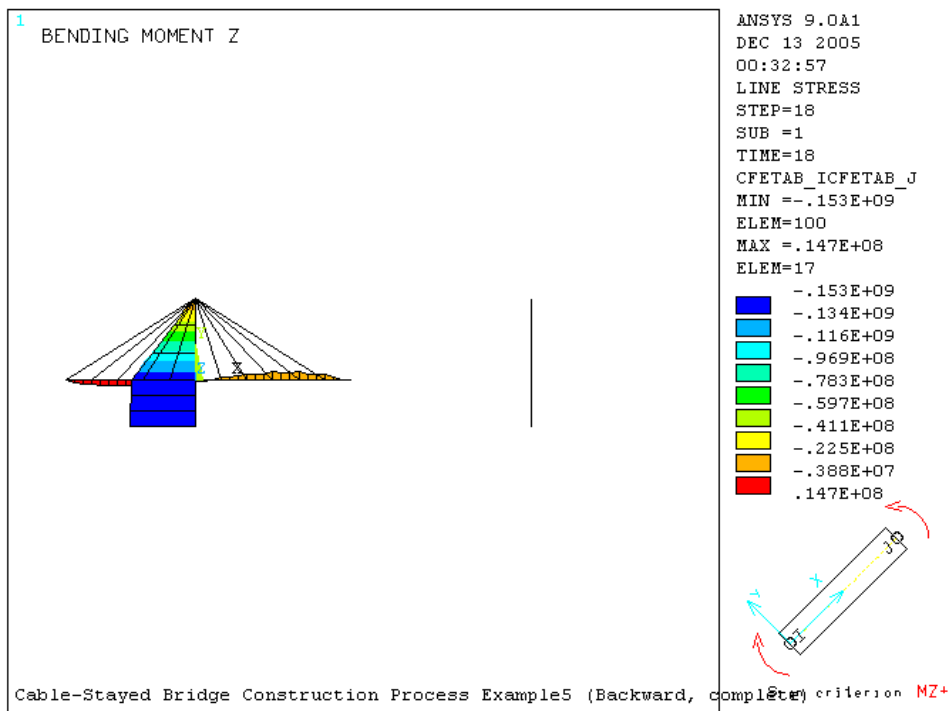
After removing a cable on the left of the left tower (LS16):



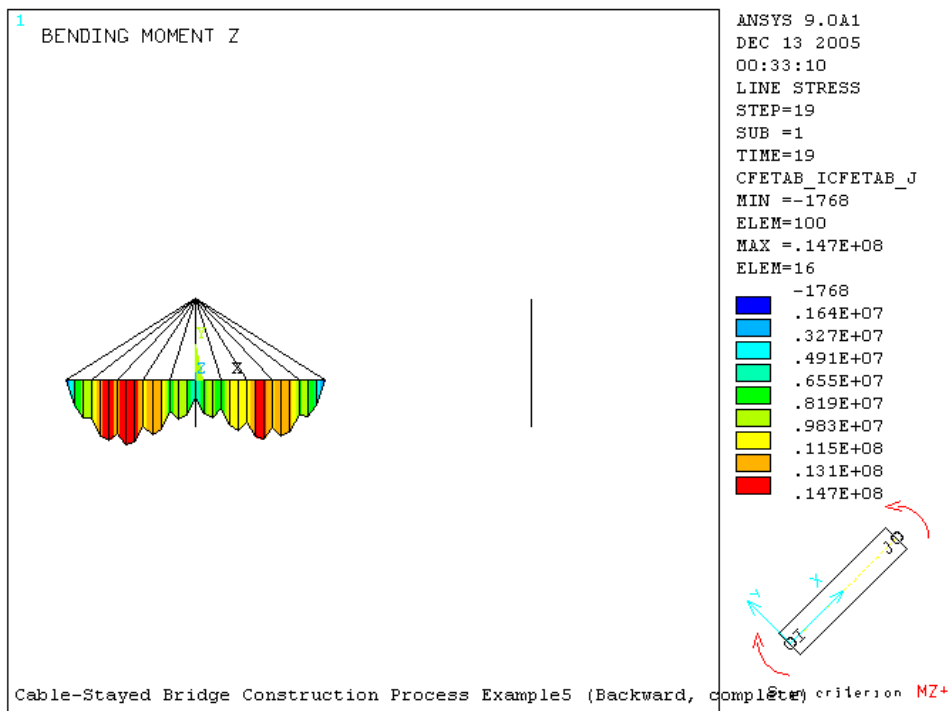
After removing a deck segment on the left of the left tower (LS17):



After removing a cable on the right of the left tower (LS18)

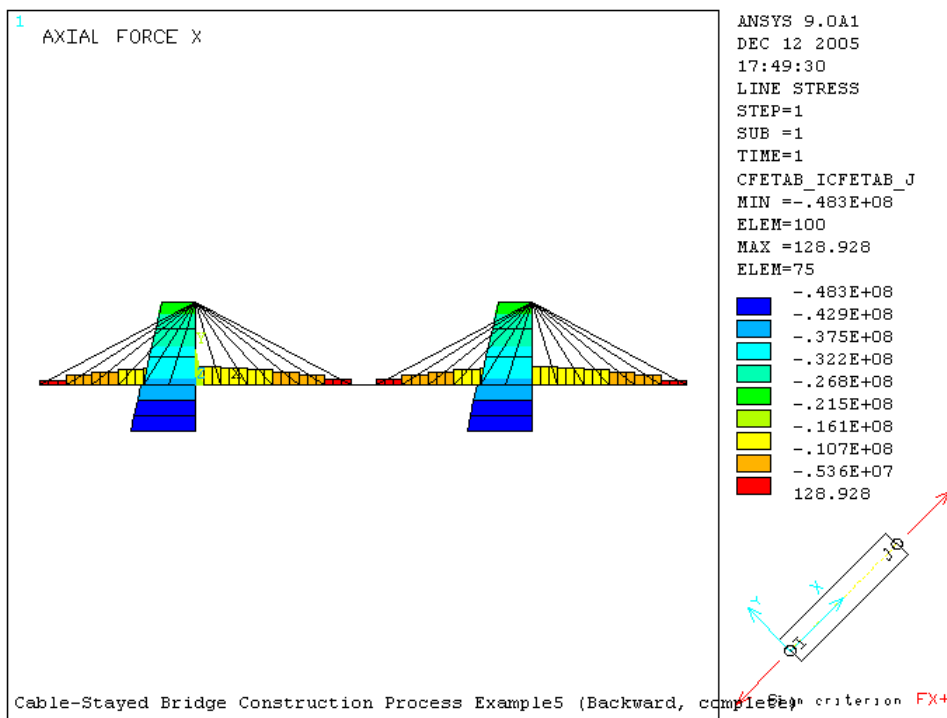


After removing a deck segment on the right of the left tower (LS19):

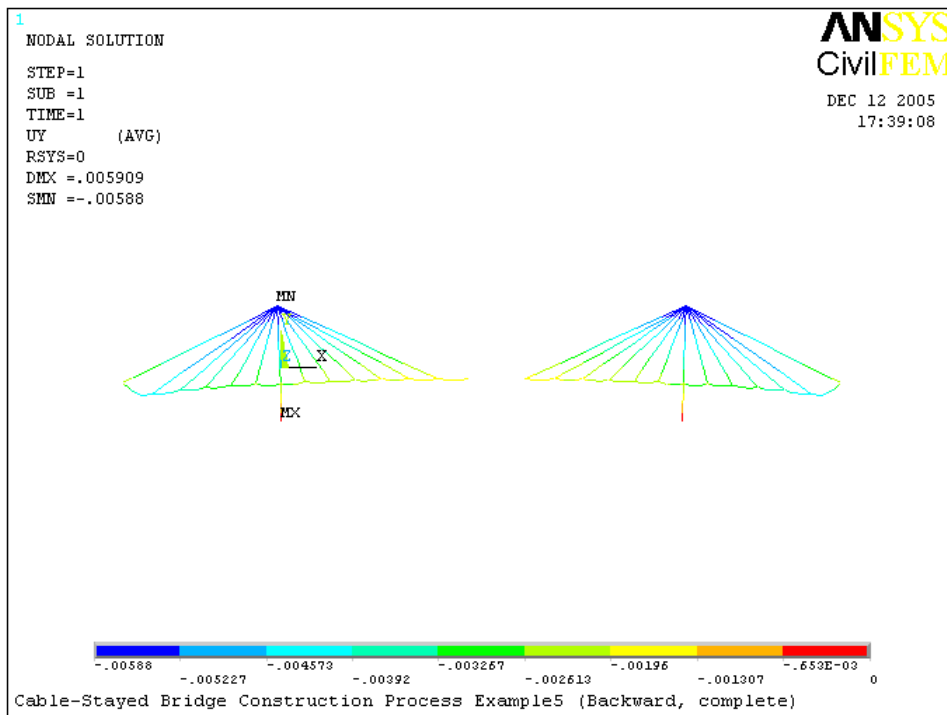


### 20.5.2 Other results in the final state of the structure

Axial force in the deck:

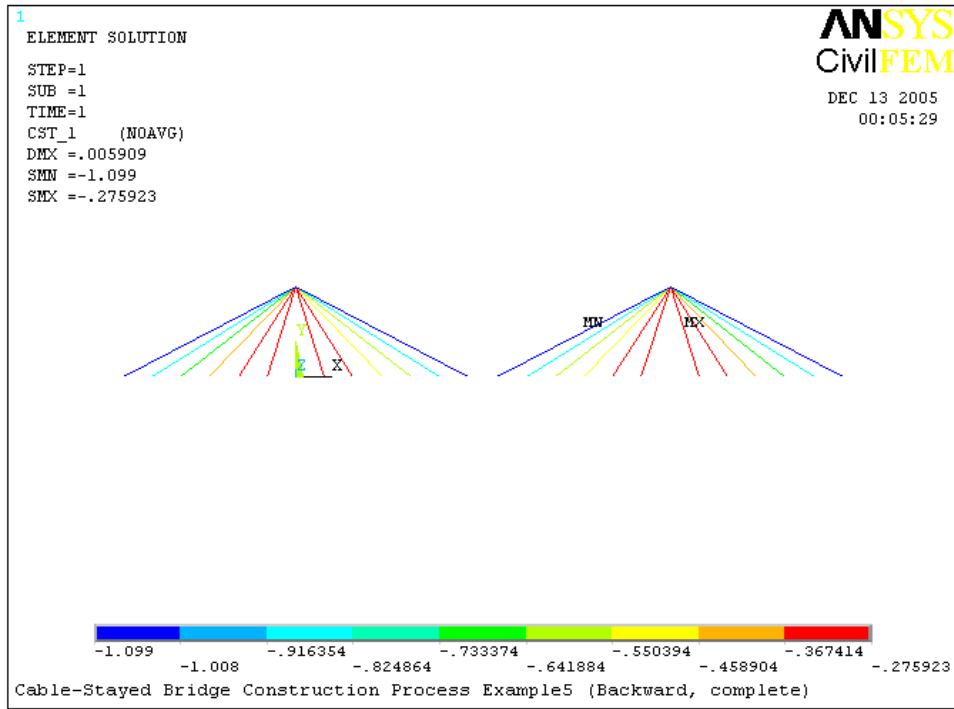


Vertical deflection of the deck:

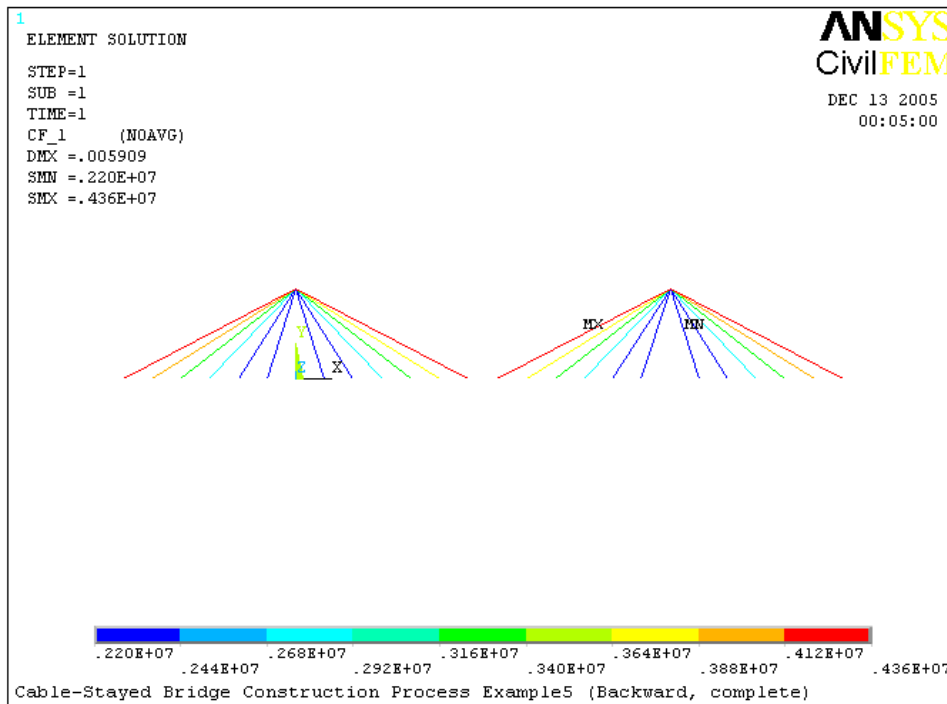


(the gap between the two sides is due to the coupling of nodes: UX movement of the coupled nodes is free)

Cable stroke (shortening):

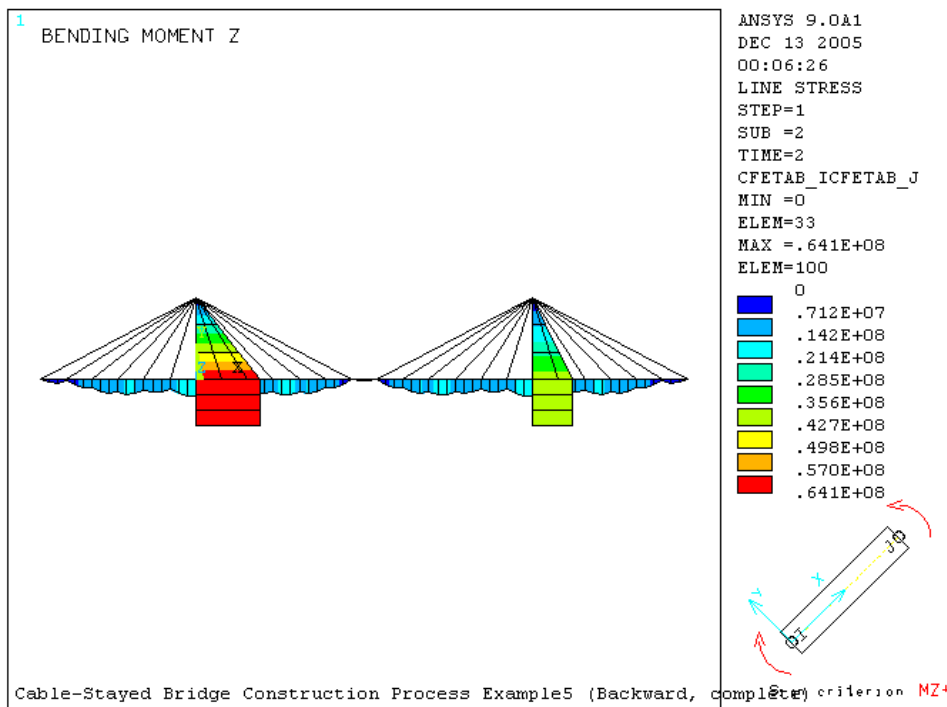


Cable force:

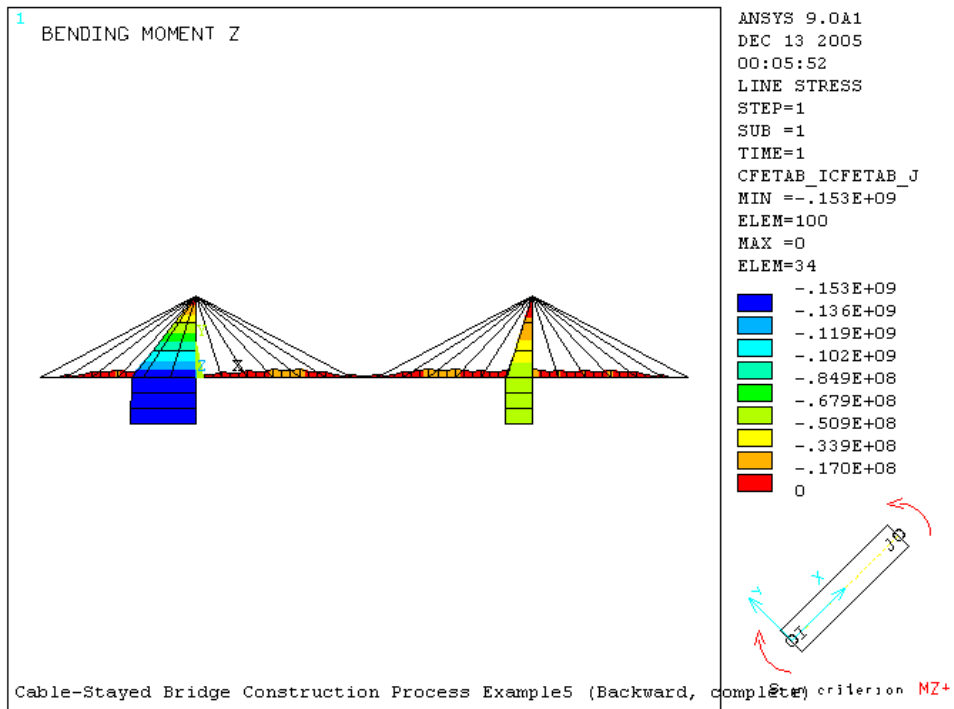


### 20.5.3 Envelope of bending moment (MZ)

Positive bending moment (MZ+):



Negative bending moment (MZ-):



### 20.5.4 Time history result (POST26)

Variation of cable forces during construction (Cables 1 to 10):

