
Example nº 21
Concrete tunnel section checking

CivilFEM Manual of Advanced Examples

Example nº 21 – Table of Contents

- 21 EXAMPLE Nº 21: CONCRETE TUNNEL SECTION CHECKING 1
 - 21.1 AIM 1
 - 21.2 DESCRIPTION OF THE EXAMPLE 1
 - 21.3 RESULTS TO BE OBTAINED 2
 - 21.4 CALCULATION LOG 2
 - 21.4.1 Introduction 2
 - 21.4.2 Model generation 4
 - 21.4.3 Solution 8
 - 21.4.4 Postprocess 8
 - 21.5 RESULTS 10

21 EXAMPLE N° 21: CONCRETE TUNNEL SECTION CHECKING

21.1 AIM

The aim of this example is to check the bending reinforcement in a concrete tunnel according to Eurocode 2 and ACI codes. It demonstrates how to create a 2D model of the underground concrete tunnel, how to capture Solid Section and define reinforcement in them and finally how to use a dummy structure to display results stored in element tables.

21.2 DESCRIPTION OF THE EXAMPLE

The tunnel is represented by a circular arch surrounded by soil. The model is fully parametric and the default values of the main parameters are the followings:

DATA:

GEOMETRY:

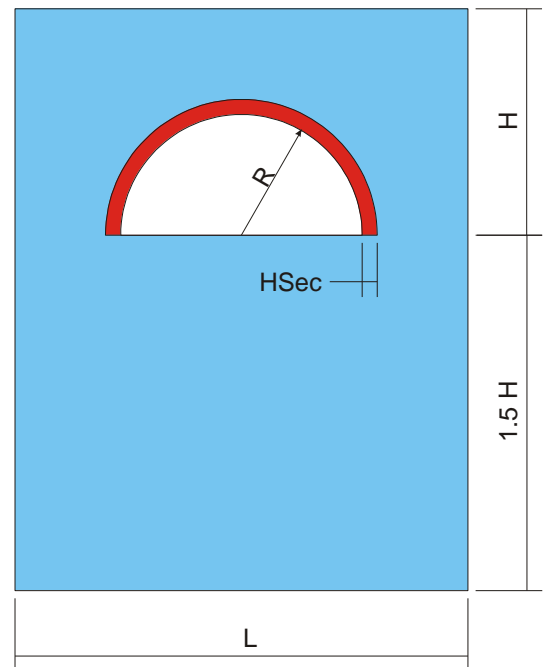
- Height of terrain: $H=30\text{m}$
- Radius of arch: $R=4\text{m}$
- Concrete width: $H\text{Sec}=0.3\text{m}$
- Length of terrain: $L=2*(H+H\text{Sec}+R)$

MATERIALS:

- Soil: Clay Low wl
- Concrete: HA-40 (EHE)
- Reinf. steel: B 500 S (EHE)

LOADS:

- The only load applied to the model is global gravity.



21.3 RESULTS TO BE OBTAINED

The purpose of the example is to calculate the forces and moments in several sections of the arch (every 15 degree along the arch) and check those sections according to Eurocode 2 and ACI 318 codes. The results are displayed over a dummy structure of line elements using element tables:

- Normal force (FX)
- Shear Force (FY)
- Bending Moment (MZ)
- Axial Force + Bending Checking Criteria (CRT_TOT)

21.4 CALCULATION LOG

21.4.1 Introduction

The parameter *SelectCode* determines the design code that is used in the code checking of the bending reinforcement in the Solid Sections. A simple *IF - *ENDIF block is used to select the code depending on the value of *SelectCode*.

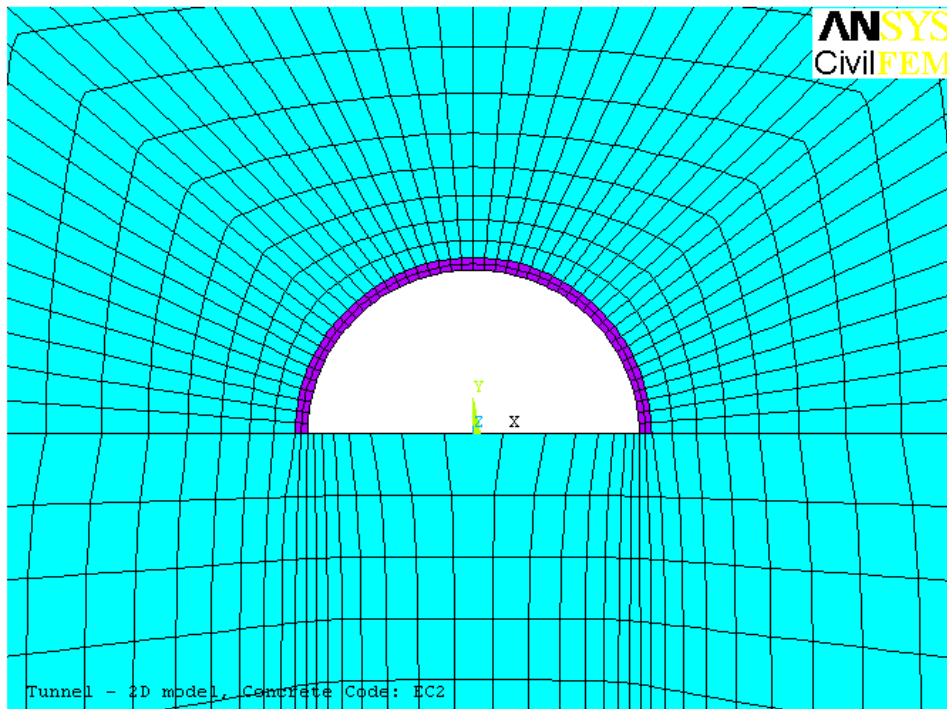
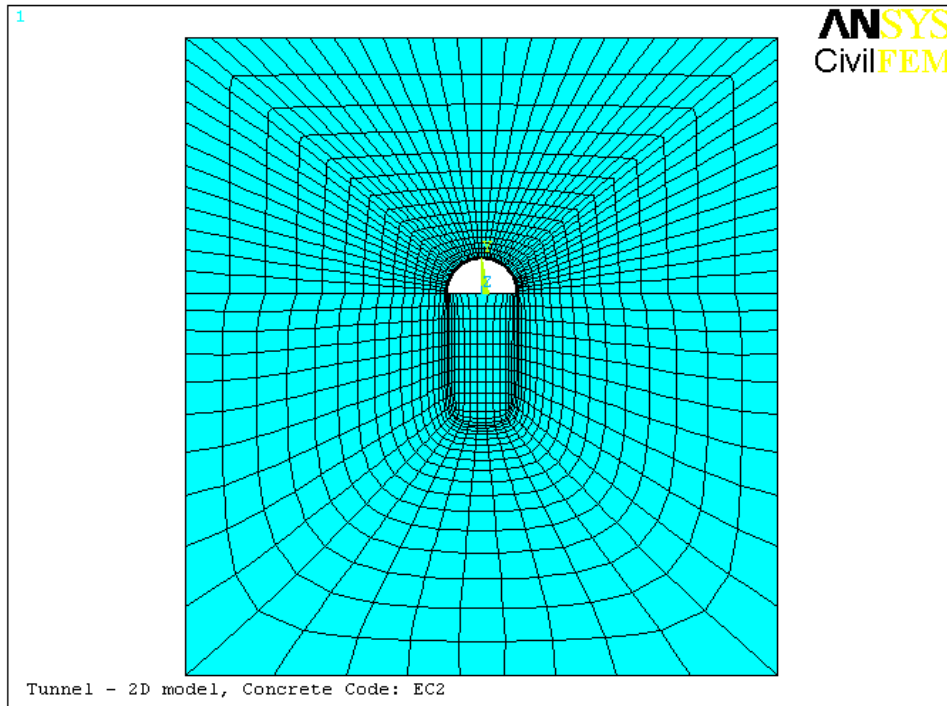
For the definition of reinforcement two parameters are introduced:

- RForceRatio ratio of total reinforcement area and cross section area.
- CCover concrete cover.

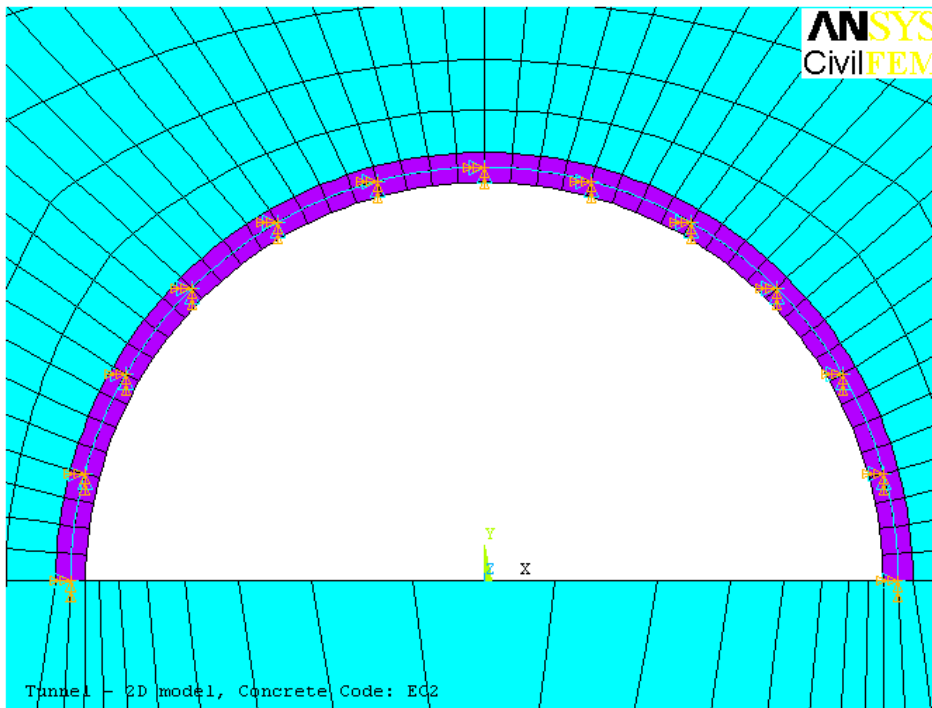
An additional element type (PIPE18) needs to be defined for the dummy structure. Since these elements are not part of the tunnel structure, they are simply used for displaying results in the Solid Sections, no results need to be stored for them.

The meshing of the model is fully controlled by mesh parameters associated with lines of the solid model. The aim is to create a mapped mesh in the whole structure. For this, some lines also need to be concatenated since a mapped mesh can only be generated in areas defined by 3 or 4 lines. The figures bellow show the final mesh of the model.

To capture Solid Sections a local coordinate system needs to be defined in the plane of the nodes that forms the section. The working plane (WP) is rotated and moved to the right place and a local coordinate system is created in the center of the WP. The nodes in the plane and the elements on one side of it are selected to capture the Solid Sections. In each section faces are defined and then reinforcement groups are defined using the faces.



A series of PIPE elements are created connecting all the locations of the Solid Sections in the tunnel. The elements are fixed at each node. The elements and the boundary conditions are shown in the figure below. PIPE elements have been chosen as dummy elements (for plotting results only) because they can have curved shapes.



Global gravity (direction Y) is applied to the structure and it is solved.

In the postprocessing CivilFEM results are collected (~CFGET command) and stored in vectors and transferred (*VPUT command) into Element Tables. These Element Tables can be then displayed over the dummy structures of PIPE elements. This is the way to display results for Solid Sections in the same way as in a beam structure.

21.4.2 Model generation

```

FINISH
~CFACTIV,CIVILFEM,Y
~CFCLEAR,,1

! SETUP
SelectCode=2 !1 (default) - EC2, 2-ACI
*IF, SelectCode,EQ,2,THEN
  ~CODESEL,EC3-92,ACI,ACI,,EC8-94
  /TITLE, Tunnel - 2D model, Concrete Code: ACI
*ELSE
  ~CODESEL,EC3-92,EC2-91,EC2-91,,EC8-94
  /TITLE, Tunnel - 2D model, Concrete Code: EC2
*ENDIF
~UNITS,SI

*ABBR,RELOAD,/INPUT,tunnel_example,inp

/PREP7

!INPUT PARAMETERS
H=30 !m, hight of terrain
HBelow=1.5*H !hight of terrain below the tunnel
R=4 !m, radius of tunnel
HSec=0.3 !m, hight of concrete section
L=2*(H+HSec+R) !width of terrain

```

```

!section parameters
ArchDivAngle=15                !element division angle
ArchNSegments=180/ArchDivAngle !number of segments
NSections=ArchNSegments+1      !number of cross sections

!mesh parameters
SectionDiv=2                    !division number along HSec
SegmentDiv=4                    !division number in one segment of the arch

!reinforcement
RForceRatio=6/1000             !reinforcement area / cross section area ratio
                                ! for automatic definition of reinforcement
CCover=0.04                    ! concrete cover: 40mm

! MATERIALS
!soil
~CFMP,1,LIB,SOIL,,CL
!concrete
~CFMP,2,LIB,CONCRETE,EHE,HA-40
!reinforcement
~CFMP,3,LIB,REINF,EHE,B 500 S

! ELEMENT TYPE
ET,1,PLANE82
KEYOPT,1,3,2                    !plane strain
ET,2,PIPE18,,,,,,,,,1          !dummy element for postprocessing,
                                ! no printout of element solution
r,2,0.1,0.01,R

! SOLID MODEL
!areas of terrain
BLC4,-L/2,0,L,H                !upper part, result: area #1
PCIRC,R+Hsec, ,0,180           !tunnel, result: area #2
ASBA,1,2                       !subtract, result: area #3, terrain above tunnel
BLC4,-L/2,-HBelow,L,HBelow    !lower part, result: area #1, terrain under tunnel
!areas of arch
ASEL,NONE
PCIRC,R+Hsec,R ,0,180          !arch, area #2
!divide into segments
WPSYLE,0.05,0.1,-1,1,0.003,0,0,,5 !display WP (Working Plane)
WPROTA,90,,90
*DO,i,1,ArchNSegments-1
  WPROTA,,ArchDivAngle
  ASBW,ALL                      !divide areas by the rotating WP
*ENDDO
WPCSYS,-1,0

ALLSEL,,
AGLUE,ALL                      !glue areas to create common lines

!create components
CSYS,1                          !cylindrical csys
ASEL,S,LOC,X,R,R+HSec
CM,ArchAreas,AREA              !component of areas of the arch segments

ALLSEL,,
CSYS,0
ASEL,S,LOC,Y,0,H              !select the areas of the uppert part of the model
                                ! (soil + arch)

CMSEL,U,ArchAreas
CM,UpSoilArea,AREA
ALLSEL,,
CMSEL,U,ArchAreas
CMSEL,U,UpSoilArea,
CM,LowSoilArea,AREA

```

```

! MESH SIZE CONTROL
CSYS,1                                !cylindrical csys
CMSEL,S,ArchAreas
ALLSEL,BELOW,AREA
LSEL,R,LOC,X,R                        !select only the two circular lines
LSEL,A,LOC,X,R+HSec
LESIZE,ALL,,SegmentDiv
LSEL,R,LOC,X,R+HSec                  !reselect the outer lines only
LCCAT,ALL                             !concatenates lines to allow mapped meshing
! of the upper soil area

ALLSEL,BELOW,AREA
LSEL,U,LOC,X,R                        !unselect the two circular lines
LSEL,U,LOC,X,R+HSec
LESIZE,ALL,,SectionDiv
CSYS,0

CMSEL,S,UpSoilArea
ALLSEL,BELOW,AREA
LSEL,U,LOC,Y,0,R+HSec                !only the outer lines of the area are selected
LCCAT,ALL                             !concatenates lines
ALLSEL,BELOW,AREA
LSEL,R,LOC,Y,0
LESIZE,ALL,,15,10,,0

CMSEL,S,LowSoilArea
ALLSEL,BELOW,AREA
LSEL,R,LOC,Y,0
LSEL,R,LOC,X,-R,+R
LESIZE,ALL,,15,-10,,0

ALLSEL,BELOW,AREA
LSEL,R,LOC,Y,0
LCCAT,ALL                             !concatenates lines

ALLSEL,BELOW,AREA
LSEL,R,LOC,Y,-HBelow
LESIZE,ALL,,9,-.5,,0

ALLSEL,BELOW,AREA
LSEL,R,LOC,Y,-HBelow/2
LSEL,R,LOC,X,-L/2
LESIZE,ALL,,10,4,,0

ALLSEL,BELOW,AREA
LSEL,R,LOC,Y,-HBelow/2
LSEL,R,LOC,X,L/2
LESIZE,ALL,,10,1/4,,0

ALLSEL

!MESHING
!-----
CMSEL,S,ArchAreas
TYPE,1
MAT,2
MSHAPE,0,2D                          !quad elements
MSHKEY,1                              !0-free, 1-mapped
AMESH,ALL
CM,ArchNodes,NODE
CM,ArchElements,ELEM

CMSEL,S,UpSoilArea
TYPE,1
MAT,1
MSHAPE,0,2D                          !quad elements

```

```

MSHKEY,1          !0-free, 1-mapped
AMESH,ALL

CMSEL,S,LowSoilArea
MSHAPE,0,2D      !quad elements
MSHKEY,1          !0-free, 1-mapped
!SMRT,3
AMESH,ALL

ALLSEL,,

! BOUNDARY CONDITIONS
NSEL,S,LOC,X,-L/2
NSEL,A,LOC,X,L/2
D,ALL,UX

NSEL,S,LOC,Y,-HBelow
D,ALL,UY

ALLSEL,,

EPLLOT

/NUMBER,1
/PNUM,MAT,1
/REPLOT

!capture solid sections along the arch
WPROTA,90,,
!the definition of the last section is different to the rest of the sections
*DO,i,0,NSections-1
  ALLSEL,,
  CMSEL,S,ArchNodes
  CMSEL,S,ArchElements
  CSYS,1
  *IF,i,GT,0,THEN
    WPROTA,ArchDivAngle
  *ENDIF
  SecNode=NODE(R+HSec/2,i*ArchDivAngle,0) !node in the middle of the section
  NWPAVE,SecNode                          !move WP to node

  *IF,i,EQ,NSections-1,THEN
    StartAngle=(i-1)*ArchDivAngle        !the last section
  *ELSE
    StartAngle=i*ArchDivAngle
  *ENDIF
  !select lines in the segment
  LSEL,S,LOC,Y,StartAngle+1,StartAngle+ArchDivAngle-1
  LSEL,R,LOC,X,R                          !select inside line
  ASLL,S                                  !select area attached to the line

  CSWPLA,11+i,0,1,1, !new CSYS aligned with WP

  NSEL,R,LOC,X,0
  ESLN,S                                  !select elements attached to the nodes
  ESLA,R                                  !reselect elements attached to the selected area
  ~SLDSEC,i+1,1,11+i,Section_%i%        !create solid section
  ~SECMDF,i+1,FACES,PT_ADD,LOC,,1,HSec/2,-0.5 !faces of a Generic cross section
  ~SECMDF,i+1,FACES,PT_ADD,LOC,,1,HSec/2,0.5 ! need to be defined manually
  ~SECMDF,i+1,FACES,PT_ADD,LOC,,2,-HSec/2,0.5
  ~SECMDF,i+1,FACES,PT_ADD,LOC,,2,-HSec/2,-0.5
  ~CFGET,SecArea,SECTION,i+1,MECHPROP,A,,1 !area of the cross section
  ~RNFDEF,i+1,1,3,1,0,CCover,SecArea*RForceRatio*0.5 !reinforcement at face 1
  ~RNFDEF,i+1,2,3,2,0,CCover,SecArea*RForceRatio*0.5 !reinforcement at face 2
*ENDDO

CSYS,1

```

```

!create a PIPE element model of the arch just for postprocessing purposes
TYPE,2
REAL,2

*DIM,PipeElements,,ArchNSegments      !array of Element IDs
*DIM,PipeSections,,ArchNSegments,2    !array to associate sections to the nodes
                                        ! of the pipe element:
                                        !   SecID at node 'i', SecID at node 'j'

NUMSTR,ELEM,10001    !numbering of elemets are important
NUMSTR,NODE,10001   ! for KLOOP parameter of *VPUT command

NSEL,NONE
ESEL,NONE
N,,0,0,0
NAux=NODE(0,0,0)
N,,R+HSec/2,0,0
Node1=NODE(R+HSec/2,0,0)
*DO,i,1,ArchNSegments
    N,,R+HSec/2,i*ArchDivAngle,0
    Node2=NODE(R+HSec/2,i*ArchDivAngle,0)
    E,Node1,Node2,NAux
    Node1=Node2
    *GET,MaxElemID,ELEM,,NUM,MAX
    PipeElements(i)=MaxElemID
    PipeSections(i,1)=i
    PipeSections(i,2)=i+1
*ENDDO
NSEL,U,,NAux
CM,PipeNodes,NODE
CM,PipeElements,ELEM
D,ALL,ALL

ALLSEL,,

!NArchCent=NODE(R,90,0)

CSYS,0

WPSTYLE,,,,,,,,,0    !do not display WP anymore
/REPLOT

FINISH

```

21.4.3 Solution

```

/SOLU

ACEL,0,9.81,0,    !global gravity

SOLVE             ! LS1

FINISH

```

21.4.4 Postprocess

```

/POST1

!arrays to store data temporarily
NResultsMax=3 !the max number of results to store in each step of postprocessing
*DIM,ResultLabels,CHAR,NResultsMax

```

EXAMPLE N° 21: CONCRETE TUNNEL SECTION CHECKING

```

*DIM,ETabLabels,CHAR,NResultsMax !these lables cannot have more than 6 character
! since we add _I or _J and the limit is 8

!two arrays are needed:
! - results at node I of elements
! - results at node J of elements
*DIM,PipeResultI,,NSections !array of results in each section
*DIM,PipeResultJ,,NSections !array of results in each section

~CFSET,0,1,LAST, !read results: LS1

/GRAPHICS, FULL !*VPUT is not supported for PowerGraphics displays

NResults=3 !(FX,FY,MZ)
ResultLabels(1)='FX','FY','MZ'
ETabLabels(1)='FX','FY','MZ'

CMSEL,S,PipeElements

*DO,i,1,NResults
!two parameters just to keep the code shorter
ELab=ETabLabels(i)
Lab=ResultLabels(i)

ETABLE,%ELab%_I
ETABLE,%ELab%_J

*DO,j,1,ArchNSegments
SecI=PipeSections(j,1) !Sec ID at node I
SecJ=PipeSections(j,2) !Sec ID at node J
~CFGET,PipeResultI(j),SOLID,SecI,FORCE,%Lab%
~CFGET,PipeResultJ(j),SOLID,SecJ,FORCE,%Lab%
*ENDDO

*VPUT,PipeResultI(1),ELEM,PipeElements(1),ETAB,%ELab%_I
*VPUT,PipeResultJ(1),ELEM,PipeElements(1),ETAB,%ELab%_J

*ENDDO

!plot results
!axial force
PLLS,FX_I,FX_J,0.1,0

!shear force
PLLS,FY_I,FY_J,0.1,0
!moment Mz
PLLS,MZ_I,MZ_J,0.1,0

!2D AXIAL + BENDING CHECKING
!-----
~CHKCON,2DB, ,0
~CFSET,1!read first alternative

NResults=1 !(CRT_TOT)
ResultLabels(1)='CRT_TOT'
ETabLabels(1)='AB_CRT', !Axial force + bending CRT_TOT

CMSEL,S,PipeElements

*DO,i,1,NResults
!two parameters just to keep the code shorter
ELab=ETabLabels(i)
Lab=ResultLabels(i)

ETABLE,%ELab%_I
ETABLE,%ELab%_J

```

```

*DO,j,1,ArchNSegments
  SecI=PipeSections(j,1)    !Sec ID at node I
  SecJ=PipeSections(j,2)    !Sec ID at node J
! get result from CivilFEM database and store it in the PipeResult arrays
  ~CFGET,PipeResultI(j),SOLID,SecI,RESULT,%Lab%
  ~CFGET,PipeResultJ(j),SOLID,SecJ,RESULT,%Lab%
*ENDDO

! copy results into element tables
*VPUT,PipeResultI(1),ELEM,PipeElements(1),ETAB,%ELab%_I
*VPUT,PipeResultJ(1),ELEM,PipeElements(1),ETAB,%ELab%_J

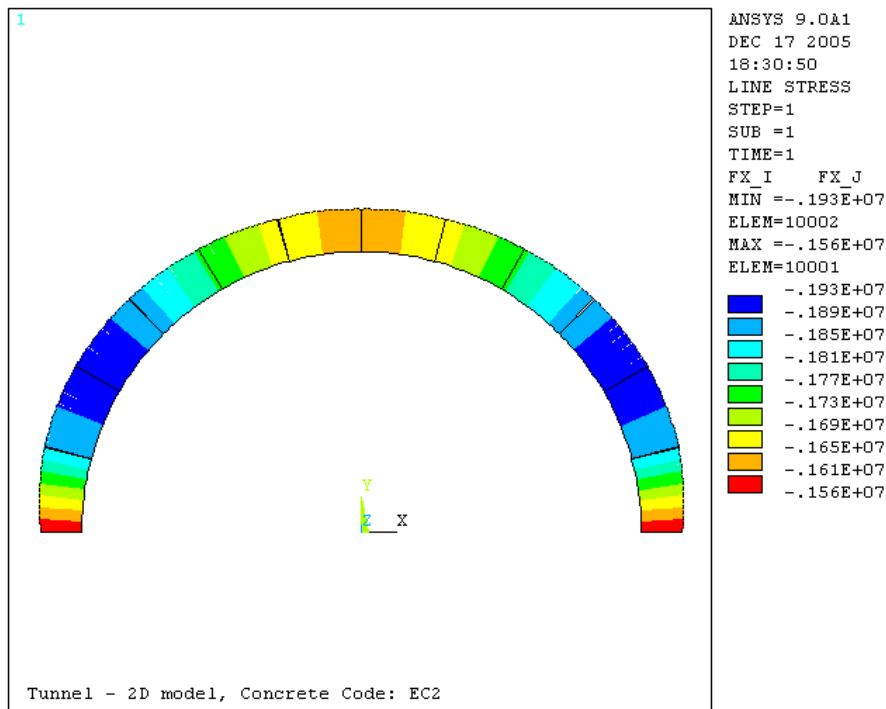
*ENDDO

! CRT_TOT
PLLS,AB_CRT_I,AB_CRT_J,0.1,0

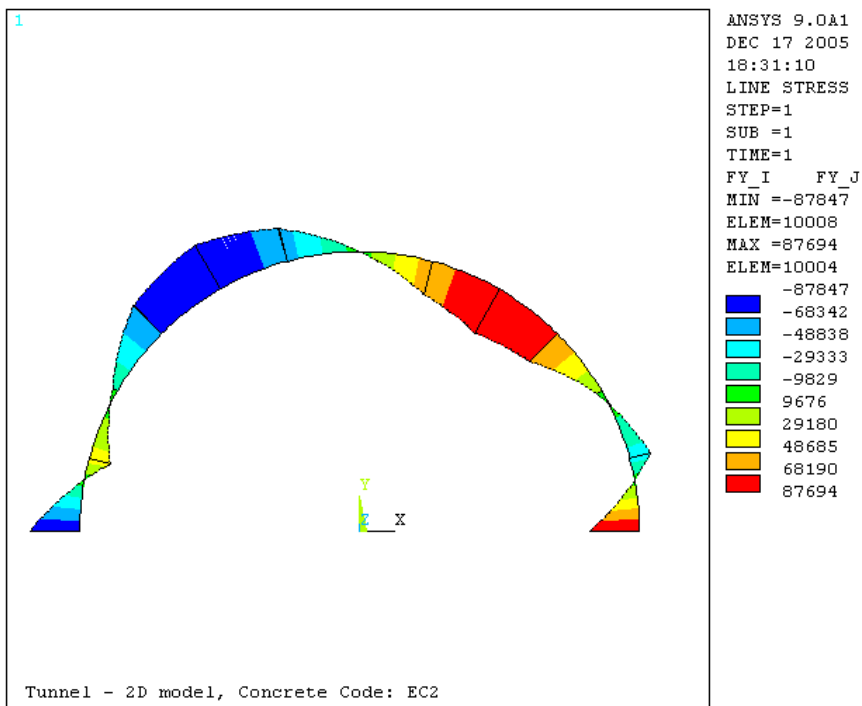
```

21.5 RESULTS

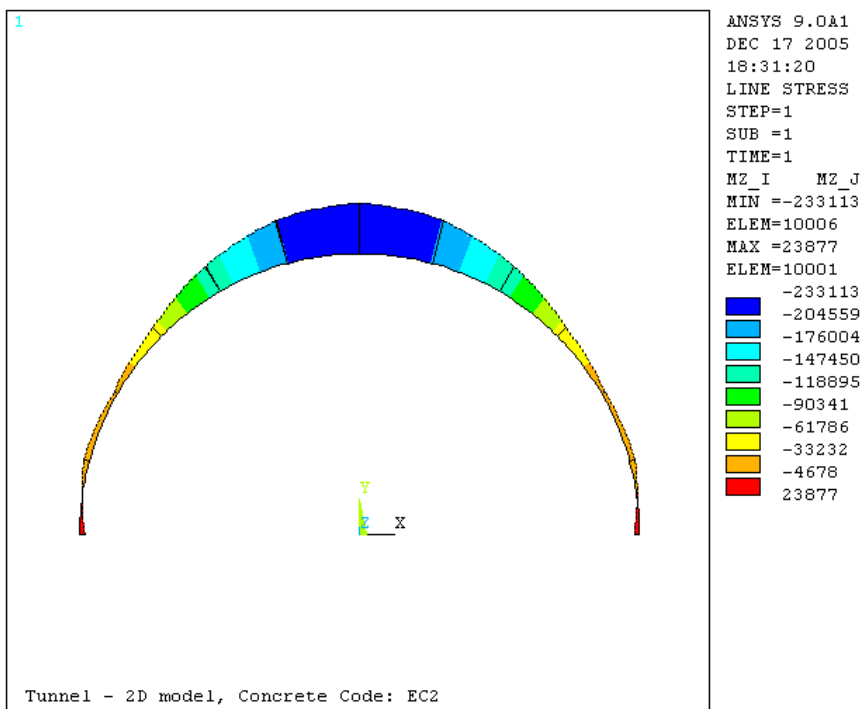
Axial force (FX)



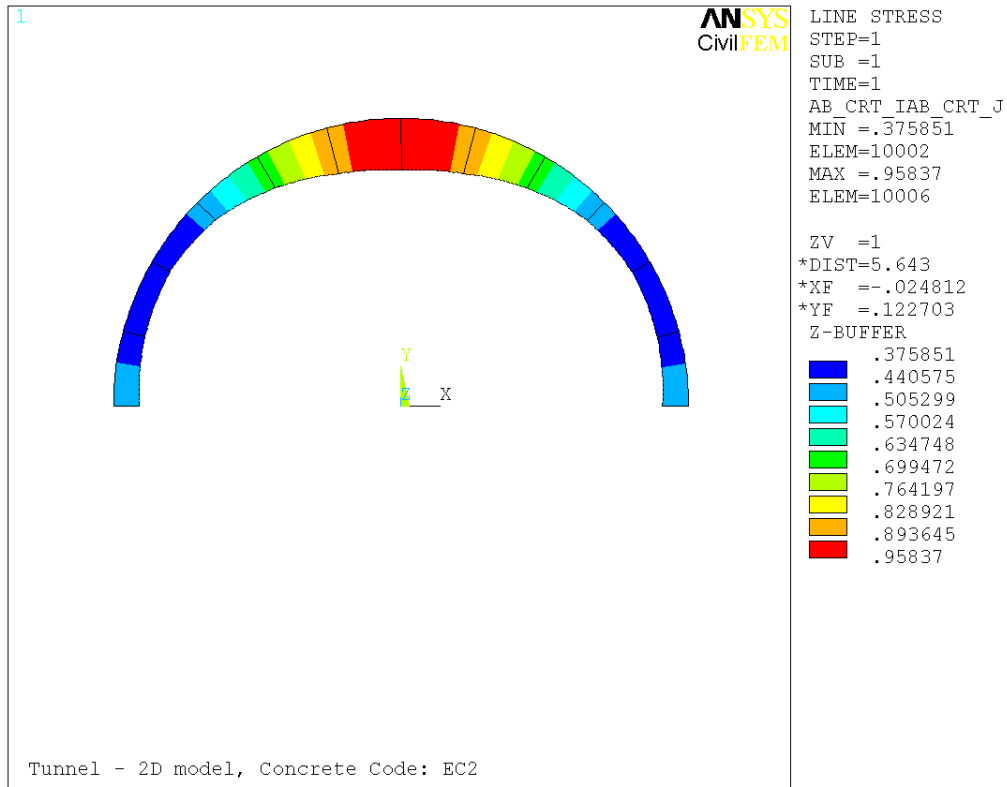
Shear force (FY)



Bending moment (MZ)



EC2: Axial Force + Bending Checking Criteria (CRT_TOT)



ACI: Axial Force + Bending Checking Criteria (CRT_TOT)

