Course in ANSYS

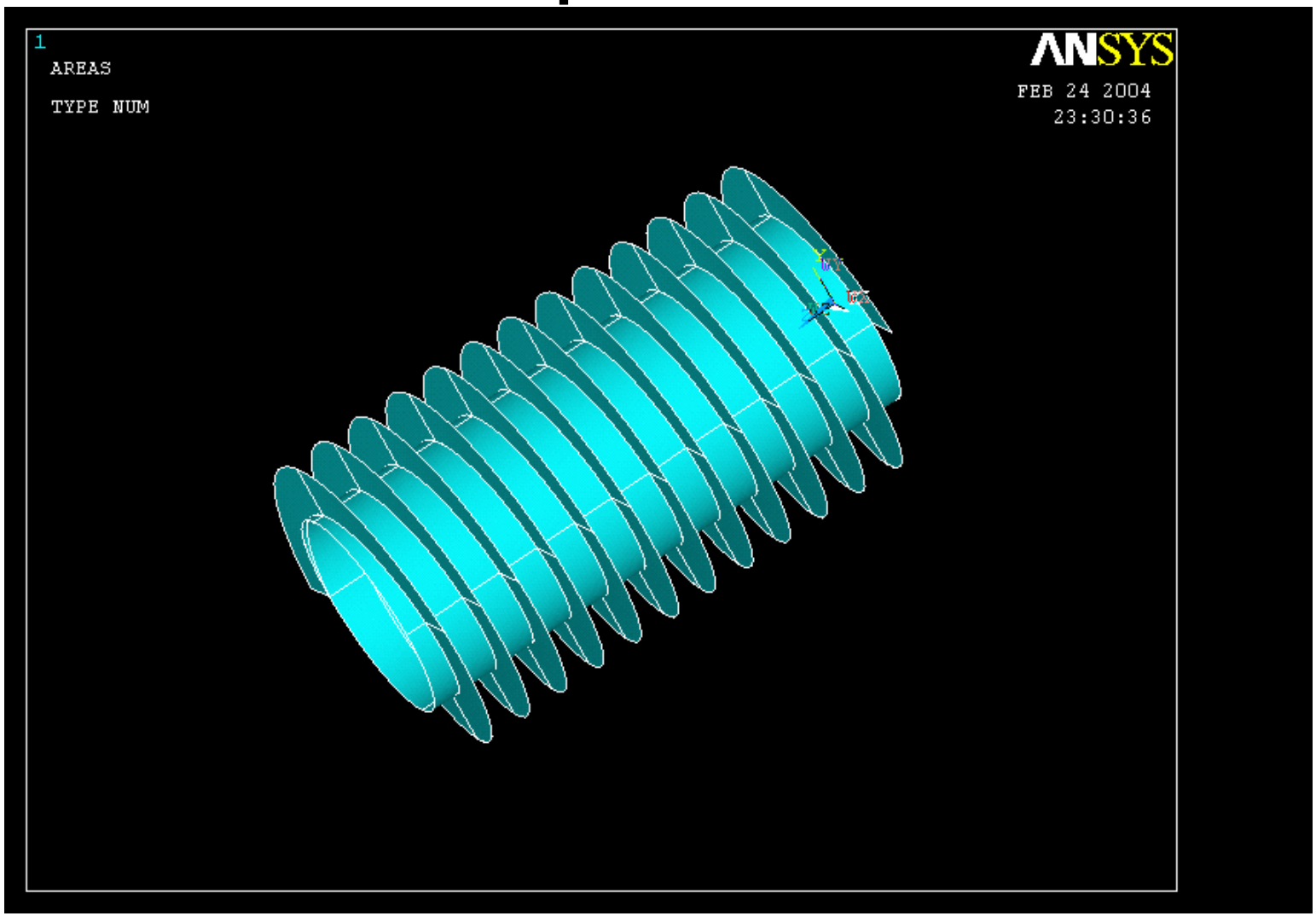
Example0700

Example – Macro

Objective:
A reusable macro that can be used to create geometry not readily accessible in the preprocessor.

Tasks:
In this example, an user frequently creates helical fins on a cylindrical body. However, no helical 2D primitive exists to facilitate the modeling. A macro was created to build the cylindrical body with a helical fin based on parameters specified by the user. The macro, named helical_fins.mac, uses the cylinders length, radius, the ribs radius and pitch as parameters with 1, 1, 2 and .2 being the default values, respectively. If zero is given for any parameter, the macro will prompt the user for a value.

Example - Macro



Example - Macro

- **Process**:
  - Step 1: Determine if the given parameters are valid.
  - Step 2: Group all existing geometry/mesh entities and unselect. This will allow for an uncluttered view of the new geometry.
  - Step 3: Create the Cylinder
  - Step 4: Create the Helical Fin
  - Step 5: Glue all of the resulting areas together. This will insure mesh connectivity.
  - Step 6: Clean up all internal parameters.

Example - Macro

- Step 1: Determine if the given parameters are valid.

```
! USAGE:
! fins, length, cylinder radius, fin radius, fin pitch
/nopr ! Suppresses printing to output window
! Check to see if users needs to enter data
*if,arg1,eq,0,then
*ask,length_,What is the cylinders length?,1
*else
length_=arg1
*endif
```

Example - Macro

- Step 1: Determine if the given parameters are valid.

```
*if,arg2,eq,0,then
*ask,crad_,What is the cylinders radius?,1
*else
crad_=arg2
*endif
*if,arg3,eq,0,then
*ask,frad_,What is the fin radius?,2
*else
frad_=arg3
*endif
*if,arg4,eq,0,then
*ask,pitch_,What is the fin pitch?,.2
*else
pitch_=arg4
*endif
```

Example - Macro

- Step 2: Group all existing geometry/mesh entities and unselect. This will allow for an uncluttered view of the new geometry.

```
cm,k_,kp
cm,l_,line
cm,a_,area
cm,v_,volu
cm,n_,node
cm,e_,elem
cmgrp,all_,k_,l_,a_,v_,e_,n_
cmsel,u,all_
```

Example - Macro

- Step 3: Create the Cylinder

```
/prep7
/view,1,1,1,1! Change to an iso view
numcmp,all ! Compress numbers of all entities
wpcsys,-1,0 ! Changes the WP to global CS but keeps
! the isoview.
! create cylinder
k,,0,0,0
kpnc_=_return ! Returns kp number of kp just created
kplo
circle,kpnc_,crad_,
*get,l4_,line,0,num,max
l3_=l4_-1
l2_=l4_-2
l1_=l4_-3
k,,0,0,length_
kpnl_=_return
l,kpnc_,kpnl_
lextr_=_return
adrag,l1_,l2_,l3_,l4_,,,lextr_
```

Example - Macro

- Step 4: Create the Helical Fin

```
! create helixes using polar coordinates
wpstyl,,,,,,1 ! WP uses polar c.s.
csys,4 ! Change local c.s. to the WP c.s.
k,,crad_,0,0
k1_=_return
k,,frad_,0,0
k2_=_return
l,k1_,k2_
l1_=_return
ii_=length_/pitch_ !determines the number of fin areas f
!looping
*do,i_,1,ii_,1
k,,frad_,90*i_,pitch_*i_
k3_=_return
l,k2_,k3_
l2_=_return
adrag,l1_,,,,,,l2_
k2_=k3_
l1_=l2_+1
*enddo
```

Example - Macro

- Step 5: Glue all of the resulting areas together. This will insure mesh connectivity.

```
aglue,all
csys
```

Example - Macro

– Step 6: Clean up all internal parameters.

```
! Clean up all variables
! $ allows multiple commands on one line
length_= $ crad_= $ frad_= $ pitch_=
l1_= $ l2_= $ l3_= $ l4_=
kpnc_= $ kpnl_= $ lextr_=
k1_= $ k2_= $ k3_=
i_= $ ii_=
/gopr ! Resumes printing to the output window
```