



APDL-Mode APDL and Syntax Highlighting 15cm25cm

Contents

1	WorkBench to APDL Translation	2
1.1	Contacts	2
1.2	Material Models	3
1.3	Boundary conditions	3
2	APDL Reference	6
2.1	Idiosyncrasies	6
2.2	File types	7
2.3	Defining parameters	9
2.4	Erasing variables from memory	12
2.5	Variable substitution	12
2.6	Expressions	14
2.7	Arrays	14
2.8	debugging	15
2.9	Multiple runs, probabilistic design	15
2.10	Undocumented commands	15
3	APDL-Mode Syntax Highlighting Reference	15
3.1	Header	15
3.2	Highlighting APDL specials	17
3.3	Implied (or colon) looping	19
4	And the rest	20

1 WorkBench to APDL Translation

1.1 Contacts

1.1.1 default 3D thermal contact

```

      2333      6      5      6      0      6458      6457      6422      6421      8
-1
keyo,cid,8,2      ! auto create asymmetric contact (from Program Controlled s
keyo,cid,10,2      ! adjust contact stiffness each NR iteration (from Program C
keyo,cid,12,5      ! bonded always
keyo,cid,2,0      ! augmented Lagrange (from Program Controlled setting)
keyo,cid,4,0      ! on Gauss point (from Program Controlled setting)
keyo,cid,9,1      ! ignore initial gaps/penetration
keyo,cid,7,0      ! No Prediction
rmod,tid,3,10. ! FKN
rmod,tid,5,0. ! ICONT
rmod,tid,6,0. ! PINB
rmod,tid,10,0. ! CNOF
rmod,tid,12,0. ! FKT
rmod,tid,36,31 ! WB DSID
rmod,cid,3,10. ! FKN
rmod,cid,5,0. ! ICONT
rmod,cid,6,0. ! PINB
rmod,cid,10,0. ! CNOF
rmod,cid,12,0. ! FKT
rmod,cid,36,31 ! WB DSID
*set,_maxkxx,605000.
rmod,cid,14,_maxkxx/_ASMDIAG      ! TCC, Divide by Length since Traction Based
rmod,tid,14,_maxkxx/_ASMDIAG      ! TCC, Divide by Length since Traction Based
keyo,cid,1,2      ! Pure thermal contact
/com,***** Create Contact "Contact Region 2" *****
/com,      Real Constant Set For Above Contact Is 8 & 7
*set,tid,8
*set,cid,7
r,tid
r,cid
et,tid,170
et,cid,174
eblock,10,,,94
(15i9)

```

2334 8 8 8 0 8439 8419 8443 8443 8

1.2 Material Models

1.2.1 Plasticity

Multilinear Kinematic Hardening (MISO) Implementation:

TB,PLAS,1,1,2,MISO

This means that one must input the curve in plastic strains and true stresses, **not** in the engineering quantities!

1.3 Boundary conditions

forces it is possible to apply time and spatially varying loads either tabular or functional. Example (depending on the coordinate sys.): $=10*\sin(x)$

moment scope: geometric selection, named selection, remote point contact pair: conta174, targe170 and pilot node similar to remote point pilot node is placed at the centre of the geom. curvature moment is applied around the reference coordinate system. pinball radius (might) reduce(s) the memory intensive range of participating elements

remote point contact pair with reference geometry and pilot node(point)

remote force contacts with pilot node MPC formulation (flexible or rigid)

pressure surf154, sf

force surf154, sfe "line force": surf156, sfe

bolt pretension prets179 (WB: select only ONE face for whole stud!)

bearing load surf154, Elements selected in load direction and pressure load applied on projected area (WB: select ALL faces of a cylinder!)

hydrostatic pressure surf154, sfgad and sf

Pressure Surface elements **surf154** and surface loads on elements (**sfe**)

```
/com,***** Define Pressure Vector Using Surface Effect Elements *****
local,12,0,0.,0.,0.,0.,0.,0.
csys,0
et,2,154
```

```

eblock,10,,,10
(15i9)
61      2      2      2      12      116      107      67      66      398
62      2      2      2      12      115      114      107      116      407
63      2      2      2      12      107      108      68      67      396
64      2      2      2      12      114      113      108      107      406
65      2      2      2      12      108      109      69      68      399
66      2      2      2      12      113      112      109      108      405
67      2      2      2      12      109      110      70      69      401
68      2      2      2      12      112      111      110      109      404
69      2      2      2      12      110      106      71      70      395
70      2      2      2      12      111      105      106      110      394
-1
esel,s,type,,2
keyop,2,2,1      ! Apply load in local coordinate system
keyop,2,11,2     ! Use real and not project area
esel,all

*DIM,_loadvari28x,TABLE,2,1,1,TIME,
! Time values
_loadvari28x(1,0,1) = 0.
_loadvari28x(2,0,1) = 1.
! Load values
_loadvari28x(1,1,1) = 0.
_loadvari28x(2,1,1) = -1.

*DIM,_loadvari28y,TABLE,2,1,1,TIME,
! Time values
_loadvari28y(1,0,1) = 0.
_loadvari28y(2,0,1) = 1.
! Load values
_loadvari28y(1,1,1) = 0.
_loadvari28y(2,1,1) = -1.

*DIM,_loadvari28z,TABLE,2,1,1,TIME,
! Time values
_loadvari28z(1,0,1) = 0.
_loadvari28z(2,0,1) = 1.

```

```
! Load values
_loadvari28z(1,1,1) = 0.
_loadvari28z(2,1,1) = 0.
```

```
...
```

```
esel,s,type,,2
nsle
sfe,all,1,pres,1,%_loadvari28x%
sfe,all,2,pres,1,%_loadvari28y%
sfe,all,3,pres,1,%_loadvari28z%
nsel,all
esel,all
```

displacements

remote displacement x,y,z, displacements are for the pilot node NOT the entire area!

simply supported hinge for beams (rotations are free)

elastic support surf154, r(4)=foundation stiffness, default thickness=1, damping possible

cylindrical support rotation with nmodif,node,x,y,z,thxy,...

compression only support rigid surf2surf contacts

Frictionless Support Fixing normal displacement with (d), possibly modifying nodal coordinate system (nmodif), turning into normal direction.

```
/com,***** Frictionless Supports X *****
```

```
CMBLOCK,_FRICSUX,NODE, 48
```

```
(8i10)
```

61	62	63	64	65	66	115	116	
	117	118	119	120	121	122	123	124
							
	408	409	410	411	412	413	414	415
	416	417	418	419	420	421	422	423

```
cmsel,s,_FRICSUX
```

```
d,all,ux,0
```

```

nset,all
nset,all
/com,***** Node Rotations *****
nmod,61,,,,-43.6672749853753,0.,0.
...

```

2 APDL Reference

2.1 Idiosyncrasies

- You can only store character variables of 8 characters, strings of 32 characters and for a maximum of 248 characters you need to create a string array!
- No proper function definitions <- write 'command' files (suffix: .mac), or call a macro (arbitrary suffix) with '*use', something close is to fill a "table" array, interpolating values and possible real indexing A(0.3).
- you can get table array values with real index values but must use integers for assigning them the values, the same goes for *vplot: it needs the array indices in integers and is, moreover, only capable to plot the columns and not their line values!
- *vplot does only plot the columns of arrays, it is not possible to specify rows
- No direct array values to file export in GUI mode <- write command file for *vwrite, or use a (lookup) table for this purpose
- *vwread does not work with C format specifiers in contrast to *vwrite
- Still (v15) no **round** function in sight, but something like `nint(max*1e3)/1e3` might do for you
- One cannot directly get the variable value, either one must assign the variable to another one, or use the '*stat' command
- Operators > and <: $1 < 2 = 1$; $2 < 1 = 1$; $2 > 1 = 2$:TODO check
- Inconsistent naming: `/ {x,y} range` but `/axlab, {x,y}`,
- Inconsistent comma usage: `/com` and `c***` are working with and without a subsequent comma (`/com` without a comma is eating one character)

- The `/contour` command does not work on device `/show,PNG`
- Deletion of array parameters without warning is only possible with an undocumented option: `*del,Array,,nopr`
- `*cfwrite` does parameter substitution without `%%`: `*cfwrite, X_points = NoN,*cfwrite`, the same as `X_points = %NoN%????`
- No direct operations on arrays, like `A=A*3`, takes a detour with `*voper` or `*toper`
- Load symbol vectors `/pbc,all,,1` in `/prep7` are uniform in contrast to the more realistic ones in `/solu`
- You can send only a complete block structure to the solver

2.2 File types

the whole zoo is documented in the operations guide

No	Type	Name	temp.	Rem
1	abort	.abt		
2	graphics annotation commands	.ano	yes	
3	neutral file format	.anf	no	
4	animation	.anim		
5		.ans_log		
6	input data copied from batch input file /batch	.bat	yes	
7	sparce solver	.bcs	no	run
8	interpolated body forces (bfint)	.bfin	no	
9		.cdb		
10	sparce solver	.dsp		run
11	interpolated DOF data (cbdof)	.cbdo	no	
12	color map	.cmap	no	
13	default command file suffix (*copen, *cfwrite)	.cmd	no	
14	component mode synthesis	.cms	no	
15	nonlinear diagnostics file (nldiag)	.cnd	no	
16	pcg solver	.pcs		run
17	workbench solver input	.dat		
18	database	.db		
19	db backup	.dbb		
20	databas from vmseh failure in batch mode	.dbe	no	
21	fortran solution information	.dbg	no	
22	Do-loop nesting	.do#	yes	
23	scratch file modal analysis	.dscr	yes	
24		.D#		
25	perfomance information sparse solver distributed	.dsp	no	
26	scratch file distributed sparse solver	.dsp#		
27	Superelement DOF solution from use pass	.dsub	no	
28	Element definitions (EWRITE)	.elem	no	
29	element matrices	.emat		
30	element saved data	.esav		
31	errors and warnings	.err		
32	distributed memory	#.err		
33	rotated element matrices	.erot	yes	
34	Element saved data ESAV files created by nonlinear analyses	.esav	yes	
35	scratch file PCG Lanczos eigensolver	.evc	yes	
36	scratch file PCG Lanczos eigensolver	.evl	yes	
37		.ext		
38		.exti		
39	local results file distributed memory	#.ext		
40	stiffness-mass matrices	.full		
41	Fatigue data [FTWRITE]	.fatg	no	
42	neutral graphics file	.grph	no	
43	Graphical solution tracking file	.gst	no	
44	IGES file from Ansys solid model data [IGESOUT]	.iges	no	
45	initial state	.ist		
46	Loading and bc of load steps (used for multiframe restart)	.ldhi		
47	Database command log file [LGWRITE]	.lgw	no	

- .mac
- .db
- .dbb

2.3 Defining parameters

up to 5000

2.3.1 Double, char38, char8, logical, TODO

in table only 8 chars?

2.3.2 Variable names

Called 'parameter' in the Ansys manual. All numeric values are stored as double precision values. Not defined variables are assigned a tiny value near zero. The interpreter is not case sensitive :TODO except in strings?

- Parameters must begin with a letter or an underscore

```
1ansys = 3           !is not a valid variable name
a1nsys = 3           !a1nsys is a valid variable name
A1NSys = 4           !this is the same variable
A1NSys = Temp        !'Temp' is not defined, near zero
```

The following text is the respective Ansys solver/interpreter output.

```
BEGIN:
  1ansys = 3           !is not a valid variable name
PARAMETER 1Ansys =    3.000000000
*** ERROR ***                CP =    0.259    TIME= 18:06:41
Invalid character in parameter name.
  The setting of parameter= 1Ansys is ignored.
BEGIN:
  a1nsys = 3           !a1nsys is a valid variable name
PARAMETER A1NSYS =    3.000000000
BEGIN:
  A1NSys = 4           !this is the same variable
PARAMETER A1NSYS =    4.000000000
BEGIN:
```

```

A1NSys = Temp                !'Temp' is not defined
*** WARNING ***              CP =          0.260    TIME= 18:06:56
Unknown parameter name= TEMP. A value of 7.888609052E-31 will be used.
PARAMETER A1NSYS =          0.7888609052E-30
BEGIN:

```

Should not begin with an underscore This convention is used in naming variables in Ansys supplied macros and the GUI.

```

_ansys = 3    !'_ansys' represents a reserved variable in Ansys supplied macros
_ = 3         ! a single underscore definition is valid
X = _
_ = 3 !the single underscore represents also a 'variable' in APDL

```

- Variable names with a trailing underscore These are hidden from the '*status' command output and can be deleted as a group with '*del'.

```

ansys_ = 3          !this is a 'hidden' variable from *status
*status             !does not show 'ansys_'
                   ,PRM_          !show variables with trailing underscore
*del,,PRM_          !delete all variables with trailing underscore

```

```

BEGIN:
ansys_ = 3
PARAMETER Ansys_ =      3.000000000
BEGIN:
*status
ABBREVIATION STATUS-
  ABBREV    STRING
  SAVE_DB   SAVE
  RESUM_DB  RESUME
  QUIT      Fnc_/EXIT
  POWRGRPH  Fnc_/GRAPHICS

```

```

PARAMETER STATUS-      (          5 PARAMETERS DEFINED)
(INCLUDING              4 INTERNAL PARAMETERS)

```

NAME	VALUE	TYPE	DIMENSIONS
X	3.00000000	SCALAR	

```

BEGIN:
,PRM_

```

```
PARAMETER STATUS- PRM_      (      5 PARAMETERS DEFINED)
(INCLUDING      4 INTERNAL PARAMETERS)
```

NAME	VALUE	TYPE	DIMENSIONS
Ansys_	3.00000000	SCALAR	

BEGIN:

- Must contain only letters, numbers and underscores

```
!! only letters, numbers and underscores are allowed
ain$sys = 3          !this is not a valid variable name
ain_sys = 3          !this is a valid variable name
```

the Ansys interpreter output looks like this:

```
BEGIN:
  ain$sys = 3          !this is not a valid variable name
*** ERROR ***          CP =      0.256    TIME= 17:35:07
Invalid character in parameter name.
The setting of parameter= A1N$SYS is ignored.
BEGIN:
  ain_sys = 3          !this is a valid variable name
PARAMETER A1N_SYS =    3.000000000
BEGIN:
```

- Must contain no more than 32 characters

```
!! The following is not a valid variable name
v23456789_123456789_123456789_123 = 3
!! The following is a valid variable name
v23456789_123456789_123456789_12 = 3
```

- Local Variables

```
Depth = ARG1 !ARG{1-9}, AR{10-19} = "*use" variables
AR18 = AR19
*stat, argx
```

2.3.3 Character strings

Must not contain more than 32 characters

```
! character string variables are enclosed with ''
Yc = '012345678901234567901234567890123' !not a character variable any more
Symetry = 'yes'
```

2.4 Erasing variables from memory

```
!! defining
Scalar = 3                !the '=' assignment is a shorthand for '*set'
*set,Scalar,4             !reassignment
*set,Vector,1,2,3,4,5,6,7,8,9,10
Vector = 0,1,2,3,4,5,6,7,8,9,10,11,12 !TODO:
Vector = 4                !TODO:
!! deleting
Scalar =                  !this is not a variable any more
*set,Scalar               !alternative to 'Scalar ='
*del,all                  !delete all variables!
*del,Vector               !TODO:
```

2.5 Variable substitution

with '%'

2.5.1 Substitution of Numeric Variables

In "string commands" like '/com', where a string follows the command name one can force the substitution of a parameter name to its value. Other examples are

```
Steel = 1
/com,Material %Steel% is steel
!! ATTENTION: in the following situation!
/com,%Steel% does NOT substitute variable Steel
/com, %Steel% does substitute variable Steel
/com,Stuff like %Steel+1% returns 2
```

2.5.2 Substitution of Character Variables

It is possible to substitute a command name

```
R='RESUME'
%R%,MODEL,DB
```

```
!! string, message commands and comment behaviour && $$$% %% :bla: &&&
```

```
/com, bla = %bla%
igesin,'test','%iges%'
```

```

/title,Nothing in %particular%
!! in "string commands" are no code comments possible
/com,beam3 %YES% ! this is *really not commented out!!!! &
c*** *beam3 !otto *otto %neither% here !!!!!!! &
/com, beamr laskf %otto% !%otto% we are here

```

- In certain 'string commands'

/title and /com are string commands similar to c***

```

right = 'wrong'
/title, the value of right is %right%
/com, this is %right%: /com does expand parameters as well

```

- Unfortunately here is no expansion possible

neither with c*** nor with /sys

```

right = 9
c***,this is %right%: c*** allows no parameter expansion
/sys,ls "*.mac" %otto% &
/syp,ls, %otto% !this is not working, no substitution!
I = 1
otto = 'file00%I%.eps'
/syp,ls, otto !this is working as intended

```

2.5.3 Dynamic Substitution of Numeric or Character Variables

or forced substitution (deferred)

```

Case = 'case 1'
/title,This is %Case%
!! /stitle
!! *ask
!! /tlabel
!! /an3d
!! in tables TODO:
aplot
Case = 'case 2'
!! not necessary to reissue /title, "This is case 2"
!! will appear on subsequent plots
aplot

```

2.6 Expressions

2.6.1 Exponentiation Operator

is '**'

2.6.2 Multiplication Expression

Beware of the oldstyle Ansys asterisk comment!

```
otto = 3 * 4 COMMENT! The value of otto is actually 3!
var1 = sinh(cos(3 *5)) ! old style Ansys comment!!!!
var2 = sinh(cos(3*5)) ! this is valid code
fini * comment
!!
```

2.6.3 Operators

'<' and '>' :TODO

```
otto = 1.82
karl = 1.97
margret = otto < karl !margret = otto
maria = karl < otto !maria = otto
*status,karl > otto
```

2.7 Arrays

4 types: array, char of 8 characters, table and string 128 chars

2.7.1 Specifying array element values

2.7.2 APDL Math

APDL Math works in its own workspace independent of the APDL environment!

```
No = 100
Pi = acos(-1)
Dat = cos(0:2*Pi:(2*Pi/No))+ cos(0:2*Pi*10:(2*Pi/No))
Dat = 0:2*Pi:2*Pi/No
*vfun
*vec,import,apdl,Dat
*fft,Forw,Dat,OutDat,,,Full !what's the difference?
```

```
*fft, ,Dat,OutDat,,,Part !what's the difference?
*export,OutDat,apdl,APDLOutDat
```

2.8 debugging

```
debug !TODO: undocumented?
```

2.9 Multiple runs, probabilistic design

```
PDEXE, Slab, MRUN, NFAIL, FOPT, Fname
in V11: *mrun !TODO:
```

2.10 Undocumented commands

```
!undocumented commands are highlighted differently
/xml !undocumented command /xml
/xfrm !documented command /xfrm
```

3 APDL-Mode Syntax Highlighting Reference

3.1 Header

```
!! -----
!! @ --- header ---
!! -----
!! Time-stamp: <2012-06-22 16:42:24 uidg1626>
!! NOTE: This is APDL pseudo code, checking
!! APDL-Mode's highlighting capabilities and
!! certain aspects of the language
!! Please see further below.
```

```
/units,mpa !indicate mm-t-s unit system
!@ --- Preprocessing ---
/prep7
!@@ -- Elements --
Steel = 1
ID = Steel
real = Steel
et,ID,solid186 !3d, 20 node
!@@ -- Material --
mp,nuxy,Steel,0.3 ! Poisson No
```

```

mp,ex,Steel,200000 ! Elastic modulus
!@@ -- Modeling --
block,0,1,0,1,0,1
!@@ -- Meshing --
vmesh,all
!@@ -- BCs, Loads --
nsel,s,loc,x,0
d,all,all
nsel,s,loc,x,1
d,all,uy,-.1
allsel
save
!@ --- Solving ---
/solu
solve
!@ --- Postprocessing --
/post1
/view,,1,1,1
plnsol,u,sum,2
/image,save,test !save XWindow Dump xwd (or bmp on Windows)
/image,capture !TODO: what is this: file0001.xwd?
/sys,convert test test.png
/upwind !TODO: 2d-graphics library? dated?
*fft !TODO: :-)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Please put the the cursor below the next paragraph of emacs lisp
!! code and type "C-x C-e" to change the setting of
!! 'ansys-highlighting-level' and 'ansys-dynamic-highlighting-flag'
!! change the level from 0 to 2 and toggle the flag from 't' to
!! 'nil'. Browse the file to check the differences.

(progn
  (when
    (featurep 'ansys-mode)
    (unload-feature 'ansys-mode))
  (setq
    ansys-highlighting-level 2
    ansys-dynamic-highlighting-flag t)
  (load-file "ansys-mode.el"))

```



```
(ansys-mode))

:TODO !! ----- /units,mpa !indicate mm-t-s unit sys-
tem c

!@@ -- Ignored characters and condensed input line ($ operator)
finishThisNightmare $ /cle !/clear
f $ fi $ fin $ fini $ finis $ finish $ finisher
```

3.2 Highlighting APDL specials

3.2.1 Reserved words and `_RETURN` statements

```
!!
N      = _RETURN      !return value of certain commands
Alpha2 = +360./(2*N)
Xc     = !empty rhs clears variables
```

3.2.2 `_RETURN` values of macros

```
*return      !TODO: what is this?
*status,_RETURN !0 normal
!1 note
!2 warning
!3 error
!4 fatal
```

3.2.3 Old style APDL comments

```
var1 = sinh(cos(3 *5)) ! old style Ansys comment!!!!
var2 = sinh(cos(3*5)) ! this is valid code
fini * comment
otto = 3 * 4 comment, the value of otto = 3!
!!
```

3.2.4 Ignored characters behind commands

```
f $ fi $ fin $ fini $ finis $ finish $ finisher
!!
```

3.2.5 The End Of File command

```
/eof --- WARNING: /eof crashes the Ansys GUI in interactive mode ---
!!
```

```
!@@ -- function names --
Pi=acos(-1) $ True=1 $ False=0 $ Nn=3.1
Alpha1 = rotx( 14.5) - 360./ (2*Nn)
```

3.2.6 Ignored characters behind commands

```
f $ fi $ fin $ fini $ finis $ finish $ finisher
a $ al $ all $ alls $ allse $ allsel $ allsellllllll
rectngaaaaa,var1,_X2,var2,X2 ! 2d rectangle
!!
```

3.2.7 The End Of File command

```
/eof --- WARNING: /eof crashes the Ansys GUI in interactive mode ---
!! /eof stops input for batch runs and the solver returns error code 8
/exit,nosave !default is save the model data
!!
```

3.2.8 Current element types and deprecated elements

```
!! A current element type:
et,10,solid186
!! deprecated element types:
et,Steel,beam3 $ et,Alu,shell191
!!
```

Let's change the element types to current ones!

```
!! Complete the following element fragments to current ones!
!!
et,Steel,beam $ et,Alu,shell
```

For example select the following elements

```
et,Steel,beam188 $ et,Alu,shell128
```

and you are getting a different element highlighting.

```
!@@ -- default commands
nsel,s,loc,y,0
      ,a,loc,y,1
      ,r,loc,x,0
d,all,all
```

3.3 Implied (or colon) looping

```
!@@ ::: implicit : (colon) looping ::::
!! (n1:n2:dn)
lfillt,(1:2),(3:4),5
!! one subscript per array
bf,(1:10),temp,Tarray(1:10)
b(1:5) = 10,20,30,40,50 !TODO: creates this an array?
!! The *get command and get functions are allowed
*get,Fx(1:10),node,(1:10),f,fz !TODO:
a(1:5) = nx(1:5)
!! TODO:
Fx(1:10) = (1:100:10)    !is this working? :-)
!! alternative to *vfill
*vfill,Fx,ramp,1,10

!! looping
*get,Dim
*if,Dim,le,1,then
    *dim,Reaction,array,Ns,1
*endif
*do,I,1,Ns
    set,Ls,I
    fsum
    *get,Fx,fsum,,item,fx
    Reaction(I)=Fx
*enddo

!@@ -- multiline *msg formatting with the & operator
*MSG,UI,Vcoilrms,THTAv,Icoilrms,THTAi,Papprnt,Pelec,PF,indctnc
Coil RMS voltage, RMS current, apparent pwr, actual pwr, pwr factor: %/ &
Steel = %G A (electrical angle = %G DEG) %/ &
_Power factor: %G %/ &
Inductance = %G %/ &
VALUES ARE FOR ENTIRE COIL (NOT JUST THE MODELED SECTOR)
```

4 And the rest

*taxis only for 3 dimension? table(0,1) = 3 is working as well

```
!@@ --! multiline message format command this is tricky: use M-o M-o
*MSG,UI,Vcoilrms,THTAi,Icoilrms,THTAi,Papprnt,Pelec,PF,indctnc
Coil RMS voltage, RMS current, apparent pwr, actual pwr, pwr factor: %/ &
Steel = %G A (electrical angle = %G DEG) %/ &
_Power factor: %G %/ &
Inductance = %G %/ &
VALUES ARE FOR ENTIRE COIL (NOT JUST THE MODELED SECTOR)
aldk this is not any longer in the *msg format construct
/com this is not any longer in the *msg format construct
```

```
*vwrite,B(1,1),B(2,1),%yes%
alkd %D &
%E%/E
```

```
!! commands which do not allow arguments
/prep7 $ FINISH !$ means nothing behind
/prep7 !still nothing behind
/prep7 * old style comment, this is allowed
/prep7 this is an error
```

```
nsel,s,loc,x,1
nsel = 3 !you CAN have variable names clashing with commands
```

```
!@@ -- Goto branching --
*go,:branch
aselsalsdkfjaölsdkfjaölskdjf,all
:branch
```

```
!-----
! mdlbl.mac
! Puts Modal Info on Plot
!-----
/post1
set,last
*get,nmd,active,,set,sbst
pfct= $ ffrq= $ adir=
nsel,s,1
```

```

*dim,pfct,,nmd,6
,
,ffrq,,nmd
,adir,char,nmd

adir(1) = 'X','Y','Z','ROTX','ROTY','ROTZ'
*stat,adir
*do,i,1,nmd
  *get,ffrq(i),mode,i,freq
  *do,j,1,6
    *get,pfct(i,j),mode,i,pfact,,direc,adir(j)
  *enddo
*enddo
/annot,delete
/plopt,info,0
/plopt,minm,off
/triad,off
/erase
iadd = arg1
*if,iadd,eq,0,then
  iadd = 1
*endif
/tspe,15,1,1,0,0
/TSPE, 15, 1.000, 1, 0, 0
xx = 1.05
yy = .9
! Change the window settings if you need different
! aspect ratios for your geometry
/win,1,-1,1,.5,1
,2,-1,1,0,.5
,3,-1,1,-.5,0
,4,-1,1,-1,-.5
!
/win,2,off
/win,3,off
/win,4,off

*get,vx,graph,1,view,x

```

```

*get,vy,graph,1,view,y
*get,vz,graph,1,view,z
*get,va,graph,1,angle
*get,vd,graph,1,dist
*do,i,2,4
    /view,i,vx,vy,vz
    /dist,i,vd
    /angle,i,va
*enddo

*do,i,1,4
    ii = i - 1 + iadd
    set,1,ii
    plnsol,u,sum
    *if,i,eq,1,then
        /noerase
    *endif
    /win,i,off
    *if,i,ne,4,then
        /win,i+1,on
    *endif
*enddo

*do,i,1,4
    ii = i - 1 + iadd
    /TLAB, xx, yy ,Mode: %ii%
    yy = yy - .05
    /TLAB, xx, yy,Freq: %ffrq(ii)%
    yy = yy - .05
    *do,j,1,6
        /TLAB, xx, yy ,PF %adir(j)%: %pfct(ii,j)%
        yy = yy - .05
    *enddo
    yy = yy - .11
*enddo
/erase
/annot,delete
sz = .8
xloc = 0
yloc = 0

```

```

*dim,data,,5
data(1) = 12,15,28,10,32
hsz = sz/2

/pspec,0,1,1
/poly,4,xloc-hsz,yloc-hsz,1.8*(xloc+hsz),yloc-hsz,
1.8*(xloc+hsz),yloc+hsz,xloc-hsz,yloc+hsz

x0 = xloc + hsz
y0 = yloc + .7*hsz
lof = .05

*vscfun,dsum,sum,data(1)
/LSPE, 15, 0, 1.000
/TSPEC, 15, 0.700, 1, 0, 0
ang1 = 0
*do,i,1,5
  ang2 = ang1 + (360*data(i)/dsum)
  /PSPE, 2*i, 1, 1
  /PWED, xloc,yloc,sz*.4, ang1,ang2
  /poly,4,x0,y0,x0+lof,y0,x0+lof,y0+lof,x0,y0+lof
  pv1 = 100*data(i)/dsum
  /tlab, x0+1.5*lof,y0, %pv1% %

  y0 = y0 - 1.5*lof
  ang1 = ang2
*enddo
/eof

```
