Tip 68: Writing Array Columns to Output or to Files

The *VWRITE command can be used to output an array column, in addition to scalar parameters. The array position from which the printing will start must be indicated when executing the *VWRITE command. As mentioned above, the *VWRITE command cannot be executed inside the GUI, it has to be executed from an input file or macro. The *VWRITE command prints the data from the starting position on down to the end of the column. The output that results can be re-directed with the /OUTPUT or with the *CFOPEN and *CFCLOS commands. The following two macros can be used to make calling the *VWRITE command easy. The array must exist, having been created with a *DIM command. The first macro works on a 1-dimensional array parameter. Note the instruction on how to call the macro, with the array parameter name surrounded by single quotes in order to delay the evaluation.

```
! This macro will print a 1-dimensional array
! according to the starting position indicated.
! If this macro is called WRITEAR1.MAC and an
! array called COL1DATA is to be printed from
! position COL1DATA(1) to the end of the array
! then call this macro with the statement:
!      WRITEAR1,'COL1DATA',1
! setting the name of the array in single quotes.
! The user may wish to change the FORMAT statement.

*vwrite,arg1(arg2)
(E16.8)
```

The second works on a 2-dimensional array parameter. The macro call will include the row and column position from which to start. The *VWRITE statement will cause printing of a column of the 2-dimensional array. When calling the macro, the array parameter name is, as above, enclosed with single quotes to delay evaluation.

```
! This macro will print a column of a 2-dimensional array
! according to the starting position indicated.
! If this macro is called WRITEAR2.MAC and an
! array called MYDATA2D is to be printed from
! position MYDATA2D(1,2) to the end of column 2 then call
! this macro with the statement:
!      WRITEAR2,'MYDATA2D',1,2
! setting the name of the array in single quotes.
! The user may wish to change the FORMAT statement.

*vwrite,arg1(arg2,arg3)
(E16.8)
```

Given that all ANSYS arrays are implicitly 3-dimensional, the second macro above could be used to print out a 1-dimensional array if the second calling parameter is set to one. A similar macro can be written to print a "column" of a 3-dimensional array. If a term in the array is MYARRAY(III,JJJ,KKK) then the *VWRITE command will cycle through the values of the III index when printing out data. The macro for a 3-dimensional array could be written so that it tests ARG2, ARG3, and ARG4 to see if they are zero. If they are zero, then they presumably were not entered, and the correct form of a *VWRITE command could be used to print a scalar, 1-D array, 2-D array, or 3-D array, as appropriate. Such a macro is illustrated below. Its use would be very error prone without error checking code. A scalar need not have its name enclosed in single quotes in calling this macro, but an array would have to be enclosed in single quotes as in the above examples. A user may want to customize this macro to change the FORMAT statements, or to remove the /NOPR and /GOPR commands.

```
! Macro to write a scalar or an array column, as appropriate.
! Indicate the starting position for *VWRITE if an array is used.
! Enclose an array parameter name in single quotes.   #################
! Examples, if this macro is called WRITER.MAC:
!    writer,aaa             ! if aaa is a scalar
!    writer,'bbb',1,3,2     ! if bbb is a 3-D array parameter
! Note the /NOPR and /GOPR commands. They will overwrite user settings.

/nopr                               ! reduce the amount printed to /OUTPUT
*if,arg2,ne,0,then                  ! if nonzero an array is used
   *if,arg3,eq,0,then               ! if not a 2-D array
       arg3=1
   *endif
   *if,arg4,eq,0,then               ! if not a 3-D array
       arg4=1
   *endif
   *vwrite,arg1(arg2,arg3,arg4)
   (E16.8)
*else                               ! if a scalar
   *vwrite,arg1
   (E16.8)
*endif
/gopr                               ! switch on /OUTPUT
```

Test these macros thoroughly before use. Note that they contain no error handling code. Warning: It is particularly difficult to remember to surround the name of the array parameter with single quotes.