Subject: PROCEDURE: Read arbitrary table of data
Posted by rsmith on Fri, 14 Jul 1995 07:00:00 GMT
View Forum Message <> Reply to Message

IDL has a handy ability to read a table of numbers of n columns by m rows, simply by defining
IDL> a = fltarr(n,m)
IDL> readf,1,a

However, one doesn't always know m and n. I wrote this piece of code to solve the problem of reading a table of data (floats or ints) with a fixed (but unknown) number of rows and columns, separated by whitespace and returns.

I'm releasing it into the public domain; if you find any bugs or make any improvements, please send them to me and I'll fix or incorporate them.

Randy Smith rsmith@wisp5.physics.wisc.edu

----------cut here----------

```
PRO read_array,filename,array,SKIP=skip,INTEGER=integer
;+
; NAME:
; READ_ARRAY
;
; PURPOSE:
;       Reads a file into an array variable.  The file is assumed to
;       consist of lines of numbers, separated by tabs or spaces,
;       with the same number of values on each line.  The file length
;       is arbitrary.
;
; CATEGORY:
;       PROG
;
; CALLING SEQUENCE:
;       READ_ARRAY, filename, array
;
; INPUTS:
;       filename: The name of the file to be read.
;       array:    The variable to hold the data.
;
; KEYWORD PARAMETERS:
; SKIP:    The number of lines of "header" information in the
;          file to skip.
;
; INTEGER: If the data in the file is integer.  The default is
```

```
;         floating point.
;
; OUTPUTS:
;       Returns a two-dimensional array whose first index is the
;       number of elements per line, and second index is the number of
;       lines in the file.  Also outputs these numbers to the screen.
;
; RESTRICTIONS:
;       Not tested all that much.  Does not read double precision data.
;
; EXAMPLE:
;       READ_ARRAY, 'spectra.dat', spectrum, SKIP=3
;       Reads the file spectra.dat, skipping the first 3 lines, creating
;       the array variable spectrum.
;
; MODIFICATION HISTORY:
;  Written by: Randall Smith, 6/19/95
;-
if (N_params() lt 2) then begin
    print,'Call with'
    print,'"Read_Array,"filename",array,[skip=n],[/integer]'
    print,'where "filename" is the file to be read'
    print,'      "array" is the variable to put the data into and'
    print,'      /skip=n where n is the number of lines at the beginning ' + $
      'to skip and '
    print,'      /integer is used if the data is integer, not float.'
    return
endif
;
; Check to see if file exists and open file
;
result = findfile(filename,count=ct)
if (ct eq 0) then begin
    print,'File : '+filename+' not found.'
    return
endif
if (ct gt 1) then begin
    print,'Multiple files match that name:'
    print,result
    return
endif
get_lun,lun
openr,lun,filename
;
; Skip any lines?
;
if (keyword_set(skip) ne 0) then begin
    line = 'string'
```

```
   for i=0,skip-1 do readf,lun,line
endif
;
; Calculate the number of elements per line
;
tab = string(9B)
space = ' '
first = 1
line = ' string '
readf,lun,line
pos = strpos(line,tab)
while (pos ne -1) do begin
   strput,line,space,pos       ; Convert tabs
   pos = strpos(line,tab)
endwhile
line = strtrim(line,2)  ; Remove extra spaces
line = line+' '    ; Guarantee at least one space found
while (strlen(line) gt 0) do begin
   pos = strpos(line,space)
   if (pos ne -1) then begin
      if (keyword_set(integer)) then begin
         number = int(strmid(line,0,pos))
      endif else begin
         number = float(strmid(line,0,pos))
      endelse
      if (first eq 1) then begin
         first = 0
         arrayline = number
      endif else begin
         arrayline = [arrayline,number]
      endelse
      nline = strmid(line,pos,strlen(line))
      line = nline
   endif
   line = strtrim(line,1)      ; Get rid of excess white space
endwhile
array = arrayline
nperline = n_elements(arrayline)
print,'Number of elements per line: ',strtrim(string(nperline),2)
numline = 1
;
; Read the file
;
if (keyword_set(integer)) then begin
   a = intarr(nperline)
endif else begin
   a = fltarr(nperline)
endelse
```

```
while (not(eof(lun))) do begin
    readf,lun,a
    array = [[array],[a]]
    numline = numline + 1
endwhile
print,'Number of lines in file: ',strtrim(string(numline),2)
;
; Clean up
;
free_lun,lun

return
end
```
