
Subject: Re: Reading fortran

Posted by [David Fanning](#) on Tue, 20 Sep 2005 13:00:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

Andres writes:

```
> Hi all,
>
> does somebody knows how to read this (fortran way)
>
>   do 210 j=1,ng2
>     write(10) real(rhoc(j)),
> 2       real(psi1(j)),real(psi2(j)),real(psi3(j))
>     write(10) aimag(rhoc(j)),
> 2       aimag(psi1(j)),aimag(psi2(j)),aimag(psi3(j))
> 210   continue
>
> in IDL? I managed, but it takes forever for big "ng2". This is what I
> do:
>
> rhoc=fltarr(ng3)
> psi1=fltarr(ng3) & psi2=fltarr(ng3) & psi3=fltarr(ng3)
>
> For i=0l,ng3-1l do begin
>   readu,lun,tmp1,tmp2,tmp3,tmp4
>   rhoc[i]=tmp1
>   psi1[i]=tmp2
>   psi2[i]=tmp3
>   psi3[i]=tmp4
> Endfor
>
> but this takes a long time... Anybody knows a fast way?
```

Here is an article for you to read:

http://www.dfanning.com/tips/ascii_column_data.html

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Reading fortran

David Fanning <davidf@dfanning.com> writes:

> Andres writes:

>

>> Hi all,

>>

>> does somebody knows how to read this (fortran way)

>>

>> do 210 j=1,ng2

>> write(10) real(rhoc(j)),

>> 2 real(psi1(j)),real(psi2(j)),real(psi3(j))

>> write(10) aimag(rhoc(j)),

>> 2 aimag(psi1(j)),aimag(psi2(j)),aimag(psi3(j))

>> 210 continue

>>

>> in IDL? I managed, but it takes forever for big "ng2". This is what I

>> do:

>>

>> rhoc=fltarr(ng3)

>> psi1=fltarr(ng3) & psi2=fltarr(ng3) & psi3=fltarr(ng3)

>>

>> For i=0l,ng3-1l do begin

>> readu,lun,tmp1,tmp2,tmp3,tmp4

>> rhoc[i]=tmp1

>> psi1[i]=tmp2

>> psi2[i]=tmp3

>> psi3[i]=tmp4

>> Endfor

>>

>> but this takes a long time... Anybody knows a fast way?

>

> Here is an article for you to read:

>

> http://www.dfanning.com/tips/ascii_column_data.html

I don't think he's reading ascii data. I think he's reading a long list of floats?

If the problem is that the data is in Fortran order, shouldn't he be reading into a large array and then transposing?

how about (assuming $\text{ng2} * 4 * \text{sizeof(float)} < \text{memory available}$):

pro read_fortran

```
;; The data is 4 columns x ngr rows, but Fortran is stored row major.
data = make_array( ngr2, 4, /float )
openr, lun, "your_file_name", /GET_LUN
readu, lun, data
col_row_data = transpose( data )

rhoc = col_row_data[ 0, * ]
psi1 = col_row_data[ 1, * ]
psi2 = col_row_data[ 2, * ]
psi3 = col_row_data[ 3, * ]
```

end

Is this sort of what you're looking for? Completely untested of course.

Check out this fanning article for colum/row major information/headache.

http://www.dfanning.com/misc_tips/colrow_major.html

Hope this helps.

Matt

--

Matthew Savoie - Scientific Programmer
National Snow and Ice Data Center
(303) 735-0785 <http://nsidc.org>

Subject: Re: Reading fortran
Posted by [David Fanning](#) on Tue, 20 Sep 2005 14:59:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

savoie@nsidc.org writes:

```
> pro read_fortran
>
> ;; The data is 4 columns x ngr rows, but Fortran is stored row major.
> data = make_array( ngr2, 4, /float )
> openr, lun, "your_file_name", /GET_LUN
> readu, lun, data
> col_row_data = transpose( data )
>
> rhoc = col_row_data[ 0, * ]
> psi1 = col_row_data[ 1, * ]
```

```
> psi2 = col_row_data[ 2, * ]
> psi3 = col_row_data[ 3, * ]
>
> end
>
>
> Is this sort of what you're looking for?
```

I think he probably wants a function. ;-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Reading fortran
Posted by [Paul Van Delst\[1\]](#) on Tue, 20 Sep 2005 15:16:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

savoie@nsidc.org wrote:

```
> David Fanning <davidf@dfanning.com> writes:
>
>
>> Andres writes:
>>
>>
>>> Hi all,
>>>
>>> does somebody knows how to read this (fortran way)
>>>
>>>     do 210 j=1,ng2
>>>       write(10) real(rhoc(j)),
>>> 2         real(psi1(j)),real(psi2(j)),real(psi3(j))
>>>       write(10) aimag(rhoc(j)),
>>> 2         aimag(psi1(j)),aimag(psi2(j)),aimag(psi3(j))
>>> 210   continue
>>>
>>> in IDL? I managed, but it takes forever for big "ng2". This is what I
>>> do:
>>>
>>> rhoc=fltarr(ng3)
>>> psi1=fltarr(ng3) & psi2=fltarr(ng3) & psi3=fltarr(ng3)
```

```

>>>
>>> For i=0l,ng3-1l do begin
>>>   readu,lun,tmp1,tmp2,tmp3,tmp4
>>>   rhoc[i]=tmp1
>>>   psi1[i]=tmp2
>>>   psi2[i]=tmp3
>>>   psi3[i]=tmp4
>>> Endfor
>>>
>>> but this takes a long time... Anybody knows a fast way?
>>
>> Here is an article for you to read:
>>
>> http://www.dfanning.com/tips/ascii\_column\_data.html
>
>
>
> I don't think he's reading ascii data. I think he's reading a long list of
> floats?

```

Same principle though.

```

> If the problem is that the data is in Fortran order, shouldn't he be reading
> into a large array and then transposing?
>
> how about (assuming ngr2 * 4 * sizeof(float) < memory available):

```

The data is written 4 numbers at a time, with 2 sets of ngr2 sets of numbers. Additionally, each set of 4 numbers is a separate record (with the attendant record markers). So how about:

```

; Open the file as "direct" access.
openr, lun, filename, /get_lun ; No /f77_unformatted

; create the data array and read the data
; Dimension 1 indices 0 and 5 are for the Fortran record markers
; Dimension 2 indices are for the real and imaginary bits
data = fltarr( 6, 2, ngr2 )
readu, lun, data

; strip out the real numbers
rrhoc = reform(data[1,0,*])
rpsi1 = reform(data[2,0,*])
rpsi2 = reform(data[3,0,*])
rpsi3 = reform(data[4,0,*])

; strip out the imaginary numbers
irhoc = reform(data[1,1,*])

```

```
ipsi1 = reform(data[2,1,*])
ipsi2 = reform(data[3,1,*])
ipsi3 = reform(data[4,1,*])
```

Also, totally untested, but you get the idea. You can also transpose the data array before stripping out the real and imaginary bits so you don't need a REFORM on every line to remove unity dimensions.

paulv

```
>
>
> pro read_fortran
>
> ;; The data is 4 columns x ngr rows, but Fortran is stored row major.
> data = make_array( ngr2, 4, /float )
> openr, lun, "your_file_name", /GET_LUN
> readu, lun, data
> col_row_data = transpose( data )
>
> rhoc = col_row_data[ 0, * ]
> psi1 = col_row_data[ 1, * ]
> psi2 = col_row_data[ 2, * ]
> psi3 = col_row_data[ 3, * ]
>
> end
>
>
> Is this sort of what you're looking for? Completely untested of course.
>
> Check out this fanning article for colum/row major information/headache.
>
> http://www.dfanning.com/misc\_tips/colrow\_major.html
>
>
> Hope this helps.
>
> Matt
>
```

--

Paul van Delst
CIMSS @ NOAA/NCEP/EMC

Subject: Re: Reading fortran
Posted by [panblosky](#) on Tue, 20 Sep 2005 17:07:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Paul, with your way it works! I don't need to make the reform
though... Thanks to all!!

Subject: Re: Reading fortran
Posted by [panblosky](#) on Tue, 20 Sep 2005 17:07:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Paul, with your way it works! I don't need to make the reform
though... Thanks to all!!
