
Subject: Re: Writing arrays to text file - format code trickery?

Posted by [rjp23](#) on Mon, 28 Nov 2011 13:39:28 GMT

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

On Nov 28, 1:29 pm, Yngvar Larsen <larsen.yng...@gmail.com> wrote:

> On Nov 28, 1:36 pm, Rob <rj...@le.ac.uk> wrote:

>

>> I've inherited some code which does something along the lines of the
>> following:

>

>> FOR i = 0L, 359 DO printf,unit,A[0,0,i],

>> A[0,1,i],A[0,2,i],A[0,3,i],A[0,4,i],A[0,5,i],format='(6(2X, E15.8))'

>

>> where A is [2, 6, 360]

>

>> Now this is done quite a lot and becomes horribly slow. Is there a
>> clever way I can use the format code to do it with fewer (one?) print
>> statements? (Maybe by juggling the array around first?)

>

> In this case:

>

> printf, unit, reform(a[0,*,*]), format='(6(2X, E15.8))'

>

> Not sure what you mean by "along the lines of ...", though. If you
> also need to write, e.g., a[1,*,*], you may have to rearrange your
> array with TRANSPOSE and/or modify the format description, depending
> on which order the rows are to be written and what each row in your
> file should contain.

>

> Also, you don't save much processing time with only 360 rows, see
> below. If the loop was much longer, you would save a lot.

>

> IDL> .r test_ascii

> % Compiled module: \$MAIN\$.

> Average processing time, method #0: 0.0038861408

> Average processing time, method #1: 0.0029639530

>

> I used the following test script:

> 8<-----

> A = randomu(seed,2,6,360)

> nTests = 1000L

>

> t0 = systime(/seconds)

> for j=0, nTests-1 do begin

> openw, unit, '/tmp/test0.txt', /get_lun

> for i = 0L, 359 do \$

>

> printf,unit,A[0,0,i],A[0,1,i],A[0,2,i],A[0,3,i],A[0,4,i],A[0,5,i],\$

```

>      format='(6(2X, E15.8))'
>   free_lun, unit
> endfor
> t1 = systime(/seconds)
> print, 'Average processing time, method #0:', (t1-t0)/nTests
>
> t0 = systime(/seconds)
> for j=0, nTests-1 do begin
>   openw, unit, '/tmp/test1.txt', /get_lun
>   printf, unit, reform(a[0,*,*]), format='(6(2X, E15.8))'
>   free_lun, unit
> endfor
> t1 = systime(/seconds)
> print, 'Average processing time, method #1:', (t1-t0)/nTests
> 8<-----
>
> --
> Yngvar

```

Thanks, that's very helpful.

By "along the lines of", I just meant that I simplified it down a bit.

I seem to see a speed increase of about 3x by removing the loop.
