
Subject: More For Loops

Posted by [majewski](#) on Thu, 13 Apr 2000 07:00:00 GMT

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

Hi

I'm looking to get rid of the for loops below.

They make two arrays;

one containing every second column and every second row

the other containing every second column and every alternate row

```
-----  
x_data = 3072  
y_data = 512  
DATA_size = [x_data/2,y_data]  
Data_sets_ev = bytarr(DATA_size[0],DATA_size[1])  
Data_sets_od = bytarr(DATA_size[0],DATA_size[1])  
  
for i = 0, (DATA_size[0]/2)-1 do begin  
  for j = 0, DATA_size[1]-1 do begin  
    Data_sets_ev[i,j] = my_data[(2*i),(2*j)]  
    Data_sets_od[i,j] = my_data[(2*i),(2*j+1)]  
  endfor  
endfor  
-----
```

This is to extract a couple of NOAA14 AVHRR Sea Surface Temperature measurements. The different spectral bands overlap in the data file, hence they need to be separated by the above.

Thanks

Leon Majewski

Leon Majewski

Remote Sensing & Satellite Research Group
Curtin University of Technology, Perth, Australia

email: majewski@ses.curtin.edu.au

Subject: Re: More For Loops

Posted by [majewski](#) on Sat, 15 Apr 2000 07:00:00 GMT

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

On 13 Apr 2000 14:56:09 -0500, Craig Markwardt
<craigmnet@cow.physics.wisc.edu> wrote:

Thankyou very much,
This works perfectly, and rather fast too...

For loop - 13.5 seconds
Reform method - 0.45 seconds (just a bit of a difference - when doing
50 or so data sets)

I ended up needing to throw in a couple more reform functions to get
it in the desired form.

```
my_data = reform(my_data, 2, x_data/2, y_data, /overwrite)
data_sets_ev = REFORM(my_data[0,*,*])
data_sets_od = REFORM(my_data[1,*,*])
```

```
>
>> majewski@cygnus.uwa.edu.au_stralia writes:
> ...
>>> for i = 0, (DATA_size[0]/2)-1 do begin
>>>   for j = 0, DATA_size[1]-1 do begin
>>>     Data_sets_ev[i,j] = my_data[(2*i),(2*j)]
>>>     Data_sets_od[i,j] = my_data[(2*i),(2*j+1)]
>>>   endfor
>>> endfor
> ...
>>
>> Keep in mind that a (2M) x N array can be thought of as a 2 x M x N
>> array -- or an M x N array of pairs. IDL can reform the first kind of
>> array into the second, and then it's a simple matter of extracting
>> what you want. The "_ev" is the first of each pair, the "_od" is the
>> second.
>>
>> my_data = reform(my_data, 2, x_data/2, y_data, /overwrite)
>>
>> data_sets_ev = my_data[0,*,*]
>> data_sets_od = my_data[1,*,*]
>
> Ah, replying to myself. I must be getting older.
>
> I see now that I didn't understand the layout of your original array.
> Your my_data is really a (2*M*2) x N array. That is, the even and odd
> rows are interleaved. This is still no problem. The revised form is:
>
> my_data = reform(my_data, 2, x_data/4, 2, y_data, /overwrite)
> ;           pair    row pair of rows array
>
> data_sets_ev = my_data[0,*,0,*]
> data_sets_od = my_data[0,*,1,*]
>
```

> In this case it appears that you are only interested in the first of
> each pair of elements, hence the [0,...].
>
> Craig
>
> --
> -----
> Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
> -----

Leon Majewski

Remote Sensing & Satellite Research Group
Curtin University of Technology, Perth, Australia

email: majewski@ses.curtin.edu.au
