Subject: Re: More For Loops
Posted by Craig Markwardt on Thu, 13 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:

> 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

---

Subject: Re: More For Loops
Posted by Craig Markwardt on Thu, 13 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

majewski@cygnus.uwa.edu.au_stralia writes:
> Hi
> I'm looking to get rid of the for loops below.
> They make two arrays;
> one containing evey second column and evey second row
> the other containing evey second column and evey 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.

Liam has shown the array subscripting method -- which is fine -- but I will show another.

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,*,*]

I use the /overwrite keyword for memory savings, but this of course

modifies the dimensions of the original data.

Craig

--

Craig Markwardt, Ph.D.
EMAIL: craigm@lheamail.gsfc.nasa.gov

Subject: Re: More For Loops
Posted by Liam E. Gumley on Thu, 13 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

majewski@cygnus.uwa.edu.au_stralia wrote:
> I'm looking to get rid of the for loops below.
> They make two arrays;
> one containing evey second column and evey second row
> the other containing evey second column and evey 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.

The syntax required to sample a multi-dimensional array is not
immediately obvious. For example, consider the following two dimensional
array:

IDL> n = 5
IDL> arr = indgen(n, n)
IDL> print, arr

```
 0      1      2      3      4
 5      6      7      8      9
10     11     12     13     14
15     16     17     18     19
20     21     22     23     24
```

To extract every second element along each dimension, you might try indexing the array as follows:

```
IDL> index = lindgen((n + 1) / 2) * 2
IDL> print, index
           0           2           4
IDL> print, arr[index, index]
      0      12      24
```

However this only extracts every other element along the diagonal. To extract the second element along each dimension, you must sample each dimension in turn, e.g.

```
IDL> sub = arr[index, *]
IDL> print, sub
      0      2      4
      5      7      9
     10     12     14
     15     17     19
     20     22     24
IDL> sub = sub[*, index]
IDL> print, sub
      0      2      4
     10     12     14
     20     22     24
```

With array expression indexing, the same result is obtained with a more compact syntax:

```
IDL> sub = (arr[index, *])[*, index]
IDL> print, sub
      0      2      4
     10     12     14
     20     22     24
```

Cheers,
Liam.
http://cimss.ssec.wisc.edu/~gumley
PS Say hi to MervL and NickB for me.