Subject: Re: Avoiding a for cicle
Posted by Med Bennett on Thu, 06 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hi Ricardo,

You can use the SHIFT command in this situation - do something like

max_pos = where(data gt shift(data,-1) and data gt shift(data,1))

of course you have to check if the edge effects make sense in your case.

Ricardo Fonseca wrote:

> Hi
>
> I'm looking for a more efficient way of implementing the following (i.e.
> avoiding the for cycle) which is a routine for finding local maximuns
>
> ; Data is a 1D Array
>
> s = Size(Data)
>
> nx = s[1]
>
> max_pos = [-1]
>
> for i = 1, nx-1 do $
> if ((Data[i] gt Data[i-1]) and (Data[i] gt Data[i+1])) then $
> max_pos = [[max_pos],i]
>
> ; and then throw away the first element...
>
> Any ideas? Thanks in advance, Ricardo

Subject: Re: Avoiding a for cicle
Posted by John-David T. Smith on Fri, 07 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Ricardo Fonseca wrote:
>
> Hi
>
> I'm looking for a more efficient way of implementing the following (i.e.
> avoiding the for cycle) which is a routine for finding local maximuns
>
> ; Data is a 1D Array

>
> s = Size(Data)
>
> nx = s[1]
>
> max_pos = [-1]
>
> for i = 1, nx-1 do $
> if ((Data[i] gt Data[i-1]) and (Data[i] gt Data[i+1])) then $
> max_pos = [[max_pos],i]
>
> ; and then throw away the first element...

max_pos = where(data gt median(data,3))

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Avoiding a for cicle
Posted by landsman on Fri, 07 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

In article <B51278FB.4C3D%zamb@physics.ucla.edu>,
 Ricardo Fonseca <zamb@physics.ucla.edu> wrote:
> Hi
>
> I'm looking for a more efficient way of implementing the following
(i.e.
> avoiding the for cycle) which is a routine for finding local maximuns
>
> ; Data is a 1D Array
>
> s = Size(Data)
>
> nx = s[1]
>
> max_pos = [-1]
>
> for i = 1, nx-1 do $
> if ((Data[i] gt Data[i-1]) and (Data[i] gt Data[i+1])) then $
> max_pos = [[max_pos],i]
>

> ; and then throw away the first element...
>
> Any ideas? Thanks in advance, Ricardo
>

And JD Smith gave the solution:

> max_pos = where(data gt median(data,3))

Cool! I think this can be described as an almost literal example of
"thinking outside of the box"

Wayne Landsman landsman@mpb.gsfc.nasa.gov

Subject: Re: Avoiding a for cicle
Posted by John-David T. Smith on Tue, 11 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> J.D. Smith (jdsmith@astro.cornell.edu) writes:
>
>> I'll reserve my entry until I see the contenders. Points are awarded for
>> unusual use of obscure IDL functions, brevity, style, lip synch, and
>> congeniality.
>
> This contest is rigged. I don't see any possible way I
> can win without points also being awarded for humor and
> general naivetï¿½. :-(
>

Any humor embedded within IDL statements will be accepted. An example might be the use of widget_info() or routine_names() for this calculation. Naivetï¿½ falls under the congeniality category, also known as the not-yet-a-grumpy-old-IDL-hack category.

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Avoiding a for cicle
Posted by davidf on Tue, 11 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith (jdsmith@astro.cornell.edu) writes:

> I'll reserve my entry until I see the contenders. Points are awarded for
> unusual use of obscure IDL functions, brevity, style, lip synch, and
> congeniality.

This contest is rigged. I don't see any possible way I
can win without points also being awarded for humor and

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Avoiding a for cicle
Posted by John-David T. Smith on Tue, 11 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
>
> Benno Puetz <puetz@mpipsykl.mpg.de> writes:
>> "J.D. Smith" wrote:
>>> Ricardo Fonseca wrote:
>>>>
>>>> Hi
>>>>
>>>> I'm looking for a more efficient way of implementing the following (i.e.
>>>> avoiding the for cycle) which is a routine for finding local maximuns
>>>>
> ...
>>>> for i = 1, nx-1 do $
>>>> if ((Data[i] gt Data[i-1]) and (Data[i] gt Data[i+1])) then $
>>>> max_pos = [[max_pos],i]
>>>>
>>>> ; and then throw away the first element...
>>>

>>> max_pos = where(data gt median(data,3))
>>>
>>
>> While this is rather efficient concerning code length,
>>
>>
>>   maxpos = WHERE(TEMPORARY(data[0:nx-3]) LT TEMPORARY(data[1:nx-2]) AND $
>>
>>   TEMPORARY(data[1:nx-2]) GT TEMPORARY(data[2:nx-1])) + 1
>>
>> should execute faster, especially for longer arrays
>
> And the code-shortened version of this is:
>
> maxpos = where((data LT data(1:*)) AND (data(1:*) GT data(2:*))) + 1
>
> There are two key things to note here. First, TEMPORARY is not needed
> when you are indexing an array, since subscripted array expressions
> are already considered temporary. Second, IDL automatically truncates
> 1-D arrays in binary operations. So, the finite difference expression
> normally written like this:
>
>   diff = data(1:nx-1) - data(0:nx-2)
>
> can be written like this:
>
>   diff = data(1:*) - data
>
> The two data arrays are of different length, so IDL takes the shortest
> of each. Saves some keystrokes, and the calculation of NX.
>

Alright code slingers... new challenge... find location of all peaks in a region
of n points (n odd), monotonically decreasing away from the peak. I.e. find
peaks of width n.

e.g. an n=5 peak:

```
  -
 - -
-   -
```

I'll reserve my entry until I see the contenders. Points are awarded for
unusual use of obscure IDL functions, brevity, style, lip synch, and
congeniality.

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Avoiding a for cicle
Posted by Craig Markwardt on Tue, 11 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Benno Puetz <puetz@mpipsykl.mpg.de> writes:
> "J.D. Smith" wrote:
>> Ricardo Fonseca wrote:
>>>
>>> Hi
>>>
>>> I'm looking for a more efficient way of implementing the following (i.e.
>>> avoiding the for cycle) which is a routine for finding local maximuns
>>>
...
>>> for i = 1, nx-1 do $
>>> if ((Data[i] gt Data[i-1]) and (Data[i] gt Data[i+1])) then $
>>> max_pos = [[max_pos],i]
>>>
>>> ; and then throw away the first element...
>>
>> max_pos = where(data gt median(data,3))
>>
>
> While this is rather efficient concerning code length,
>
>
> maxpos = WHERE(TEMPORARY(data[0:nx-3]) LT TEMPORARY(data[1:nx-2]) AND $
>
> TEMPORARY(data[1:nx-2]) GT TEMPORARY(data[2:nx-1])) + 1
>
> should execute faster, especially for longer arrays

And the code-shortened version of this is:

maxpos = where((data LT data(1:*)) AND (data(1:*) GT data(2:*))) + 1

There are two key things to note here. First, TEMPORARY is not needed
when you are indexing an array, since subscripted array expressions
are already considered temporary. Second, IDL automatically truncates

1-D arrays in binary operations. So, the finite difference expression
normally written like this:

  diff = data(1:nx-1) - data(0:nx-2)

can be written like this:

  diff = data(1:*) - data

The two data arrays are of different length, so IDL takes the shortest
of each. Saves some keystrokes, and the calculation of NX.

Craig

--
 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------

Subject: Re: Avoiding a for cicle
Posted by Benno Puetz on Tue, 11 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
"J.D. Smith" wrote:
<blockquote TYPE=CITE>Ricardo Fonseca wrote:

>

> Hi

>

> I'm looking for a more efficient way of implementing the following
(i.e.

> avoiding the for cycle) which is a routine for finding local maximuns

>

> ; Data is a 1D Array

>

> s = Size(Data)

>

> nx = s[1]

>

> max_pos = [-1]

>

> for i = 1, nx-1 do $

>   if ((Data[i] gt Data[i-1]) and (Data[i] gt Data[i+1]))
then $

>   max_pos = [[max_pos],i]

>

> ; and then throw away the first element...
<p>max_pos = where(data gt median(data,3))

 </blockquote>
While this is rather efficient concerning code length,

<pre> maxpos = WHERE(TEMPORARY(data[0:nx-3]) LT TEMPORARY(data[1:nx-2])
AND $</pre>

<pre>
    TEMPORARY(data[1:nx-2]) GT TEMPORARY(data[2:nx-1])) + 1</pre>

<pre></pre>
should execute faster, especially for longer arrays

--
Benno Puetz
Kernspintomographie
Max-Planck-Institut f. Psychiatrie
;   ;
Tel.: +49-89-30622-413
Kraepelinstr. 10   &nbs
p;   &nbs p;   &nbs
p;   &nbs p;
Fax : +49-89-30622-520
80804 Muenchen, Germany

 </html>
