Subject: Using where() on slices of data cubes
Posted by Conor on Tue, 20 Oct 2009 12:22:17 GMT
View Forum Message <> Reply to Message

I feel like this should be an easy one, but I've never quite figured it out. Let's say I got a data cube and I want to do something on just a slice of it, say I want to turn certain values in a column into something else:

w = where(cube[1,*,*] lt 0)

It seems like you should be able to do something like this:

cube[1,w] = 1e24

But that doesn't work... Somehow I can't quite figure out the right way to do this.

Subject: Re: Using where() on slices of data cubes
Posted by David Fanning on Wed, 21 Oct 2009 22:20:58 GMT
View Forum Message <> Reply to Message

JD Smith writes:

>
> On Oct 20, 4:32 pm, David Fanning <n...@dfanning.com> wrote:
>> JD Smith writes:
>>> you should easily be able to generalize the above arguments to access
>>> these elements
>>
>> I think in this case the word "easily" might be
>> too subtly sarcastic to be easily appreciated by
>> the vast majority of this newsgroup. :-)
>
> (Almost) no sarcasm was intended.
>
> Suppose you have this:
>
> w=where(cube[1,5:*,10:1024] lt 0)
>
> The "slice" is no longer as large as the cube in the yz dimensions,
> and is offset by [5,10] too. So
>
> y_full_cube = slice_column + 5
> z_full_cube = slice_row + 10
>
> and since the slice is smaller than the cube by 5 columns, to convert

> our WHERE index vector w into col,row in the slice, we use
>
> slice_column = w mod (sz[1]-5)
> slice_row = w/(sz[1]-5)
>
> Putting it all together we have:
>
> ind = 1 + sz[0] * (5 + w mod (sz[1]-5) + (10 + w/(sz[1]-5)) * sz[1])

Ah, OK. Even Coyote seems to be catching on now. ;-)

I'm curious if you have a method to test these indices?

Those of us unused to typing (I would say *most* of us,
but I don't want to offend anyone) would find it a challenge,
probably, to type a line of code like this and get it right.

How did you test this code to know it was correct?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Using where() on slices of data cubes
Posted by JDS on Fri, 23 Oct 2009 20:27:23 GMT
View Forum Message <> Reply to Message

On Oct 21, 6:20 pm, David Fanning <n...@dfanning.com> wrote:
> JD Smith writes:
>
>> On Oct 20, 4:32 pm, David Fanning <n...@dfanning.com> wrote:
>>> JD Smith writes:
>>>> you should easily be able to generalize the above arguments to access
>>>> these elements
>
>>> I think in this case the word "easily" might be
>>> too subtly sarcastic to be easily appreciated by
>>> the vast majority of this newsgroup. :-)
>

>> (Almost) no sarcasm was intended.
>
>> Suppose you have this:
>
>> w=where(cube[1,5:*,10:1024] lt 0)
>
>> The "slice" is no longer as large as the cube in the yz dimensions,
>> and is offset by [5,10] too. So
>
>> y_full_cube = slice_column + 5
>> z_full_cube = slice_row + 10
>
>> and since the slice is smaller than the cube by 5 columns, to convert
>> our WHERE index vector w into col,row in the slice, we use
>
>> slice_column = w mod (sz[1]-5)
>> slice_row = w/(sz[1]-5)
>
>> Putting it all together we have:
>
>> ind = 1 + sz[0] * (5 + w mod (sz[1]-5) + (10 + w/(sz[1]-5)) * sz[1])
>
> Ah, OK. Even Coyote seems to be catching on now. ;-)
>
> I'm curious if you have a method to test these indices?
>
> Those of us unused to typing (I would say *most* of us,
> but I don't want to offend anyone) would find it a challenge,
> probably, to type a line of code like this and get it right.
>
> How did you test this code to know it was correct?

My standard test uses a small bindgen array with an analogous offset.
Small so you can actually just check by printing it out. But I'll
admit I didn't even test in this case. Obviously for higher
dimensions this gets less useful.

```
IDL> a=bindgen(3,4,5)
IDL> print,a
  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  25  26
  27  28  29
  30  31  32
  33  34  35

  36  37  38
  39  40  41
  42  43  44
  45  46  47

  48  49  50
  51  52  53
  54  55  56
  57  58  59
IDL> slice=a[1,1:*,2:3]
IDL> print,slice
  28
  31
  34

  40
  43
  46
IDL> w=where(slice gt 0)
IDL> sz=[3,4,5]
IDL> print,1 + sz[0] * (1+ w mod (sz[1]-1) + (2+ w/(sz[1]-1)) * sz[1])
       28          31          34          40          43
46
```