Subject: Re: Loop Arrays
Posted by Craig Markwardt on Wed, 10 Oct 2001 05:46:14 GMT
View Forum Message <> Reply to Message

Ken Mankoff <mankoff@I.HATE.SPAM.cs.colorado.edu> writes:

> On Tue, 9 Oct 2001, Mark Hadfield wrote:
>
>> From: "Ken Mankoff" <mankoff@lasp.colorado.edu>
>>> I am interested in creating circular arrays, where subscripts that would
>>> be out-of-bounds on a regular array just start indexing on the other side
>>> of the array.
>>
>> You can do quite a lot with ordinary arrays using arrays of indices, eg
>>
>> a = indgen(10)
>> print, a[[0,10,20,100] mod n_elements(a)]
>>
>
> This is the technique I have been using. However there are 2 cases it does
> not cover:
>
> 1) negative indexes require a few more lines of code to get your example
> to work. I would recode it as:
>
> a = indgen(10)
> indexes = [0,10,20,100,-10,-22] ;;; or some other values...
> ind = indexes mod n_elements(a)
> neg = where(ind lt 0, num)
> if (num ne 0) then ind[neg] = ind[neg] + n_elements(a)
> print, a[ind]
>
> 2) subscript ranges. You cannot do:
> print, a[8:12 mod n_elements(a)]
>
> It is these two specific abilities that I would like to have.

Hi Ken--

Like Mark, I too have longed for the ability to index "from the
vright," so to speak, using negative numbers, or some kind of notation.
Unfortunately, negative numbers already have a meaning, or, err,
rather, the already have a non-meaning when used in an index list.
Negative numbers and too-big numbers are clipped when used in an index
list.

However, you can get a little of what you want back by using this
notation:

```
  print, a[ (ii + na) MOD na ]
```

If ii is guaranteed only to be in the range of [-na to +na] then this
will always work. As you pointed out though, you can't do this with
index ranges.

Good luck,
Craig

--

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