Subject: Re: FOR statement
Posted by Paul van Delst on Thu, 12 Apr 2001 15:57:26 GMT
View Forum Message <> Reply to Message

Eli Beckerman wrote:
>
> Hey,
>
> I just tried running a FOR loop in the hopes
> of incrementing the variable "i" by steps of 0.25 as follows:
>
> radius=fltarr(1000)
> FOR i=0.0, 100.0, 0.25 DO BEGIN
>
> radius(i)=i
>
> ENDFOR
>
> And what I end up with is an array that starts
> with the value 0.75 and is incremented by steps of 1.
>
> I'm following the convention of the FOR statement as
> presented in IDL's online help. What am I doing wrong?!

Try

```
 i=-1L
 FOR xi=0.0, 100.0, 0.25 DO BEGIN & i=i+1L & radius(i)=xi & endfor
```

```
IDL> print, radius[0:10]
     0.00000     0.250000     0.500000     0.750000      1.00000      1.25000
1.50000
     1.75000      2.00000      2.25000      2.50000
```

using i as the array subscript must be converting it to an integer type and screwing stuff up somehow.

Using loop indices like this should be discouraged, I reckon.

paulv

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
paul.vandelst@noaa.gov Alexander Pope.

Subject: Re: FOR statement
Posted by John-David T. Smith on Thu, 12 Apr 2001 16:04:13 GMT
View Forum Message <> Reply to Message

Eli Beckerman wrote:
>
> Hey,
>
> I just tried running a FOR loop in the hopes
> of incrementing the variable "i" by steps of 0.25 as follows:
>
> radius=fltarr(1000)
> FOR i=0.0, 100.0, 0.25 DO BEGIN
>
> radius(i)=i
>
> ENDFOR
>
> And what I end up with is an array that starts
> with the value 0.75 and is incremented by steps of 1.
>
> I'm following the convention of the FOR statement as
> presented in IDL's online help. What am I doing wrong?!

Indexing an array with a floating point number.

try:

FOR i=0.0, 100.0, 0.25 DO BEGIN radius[4*i]=i

or even better:

radius=findgen(401)/4.

JD

Subject: Re: FOR statement
Posted by Liam E. Gumley on Thu, 12 Apr 2001 16:12:11 GMT
View Forum Message <> Reply to Message

Eli Beckerman wrote:
> I just tried running a FOR loop in the hopes
> of incrementing the variable "i" by steps of 0.25 as follows:
>
> radius=fltarr(1000)
> FOR i=0.0, 100.0, 0.25 DO BEGIN
>

> radius(i)=i
>
> ENDFOR
>
> And what I end up with is an array that starts
> with the value 0.75 and is incremented by steps of 1.
>
> I'm following the convention of the FOR statement as
> presented in IDL's online help. What am I doing wrong?!

You are using I as both an array index and a loop variable. This is not a good idea.

Try this instead:

```
nx = 1000    ; number of values required
dx = 0.25    ; step size
x1 = 0.0     ; start value

radius = lindgen(nx) * dx + x1

print, radius[0:5]
```

Bottom line: Don't use loops to create mesh vectors or arrays.

Cheers,
Liam.
http://cimss.ssec.wisc.edu/~gumley/

Subject: Re: FOR statement
Posted by Liam E. Gumley on Thu, 12 Apr 2001 16:15:12 GMT
View Forum Message <> Reply to Message

"Liam E. Gumley" wrote:
> Try this instead:
>
> nx = 1000   ; number of values required

What I meant to say was

nx = 401

but the rest of my post still applies.

Cheers,
Liam.

Subject: Re: FOR statement
Posted by Craig Markwardt on Thu, 12 Apr 2001 16:18:59 GMT
View Forum Message <> Reply to Message

Eli Beckerman <ebeckerman@cfa.harvard.edu> writes:

> Hey,
>
> I just tried running a FOR loop in the hopes
> of incrementing the variable "i" by steps of 0.25 as follows:
>
> radius=fltarr(1000)
> FOR i=0.0, 100.0, 0.25 DO BEGIN
>
> radius(i)=i
>
> ENDFOR
>
>
> And what I end up with is an array that starts
> with the value 0.75 and is incremented by steps of 1.
>
> I'm following the convention of the FOR statement as
> presented in IDL's online help. What am I doing wrong?!

You appear to be mixing the semantics of indexing with assignment.
What do you think happens when you do radius(0.25) = 0.25, which is
followed by radius(0.5) = 0.5 and radius(0.75) = 0.75? IDL
automatically converts floating point array indices into integers by
rounding down, so all of the statements I just mentioned affect the
same array element, and 0.75 supercedes because it is last.

You probably want:

FOR i=0.0, 100.0, 0.25 DO radius(i*4)=i

Or better yet:

FOR i=0, 999 do radius(i) = i * 0.25
(which avoids the ambiguity of indexing by a floating point number)

Or even better:

radius = findgen(400)*0.25
(avoids FOR loops altogether)

Also, why do you have a 1000 element array when you only fill the
first 400 elements?

Craig

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

Subject: Re: FOR statement
Posted by Eli Beckerman on Thu, 12 Apr 2001 19:41:16 GMT
View Forum Message <> Reply to Message

Thanks All.
In defense of myself, I wasn't using the FOR loop to make an array
(I just excluded all the unnecessary stuff from my posting, and
seemingly confused everyone even further!)

But clearly, I was mixing the semantics of indexing with assignment
as Craig pointed out!

FOR i=0, 999 do radius(i) = i * 0.25 is the way to go for what I
wanted. Thanks for seeing through my morning foggyness.

-Eli

Subject: Re: FOR statement
Posted by Liam E. Gumley on Thu, 12 Apr 2001 21:04:14 GMT
View Forum Message <> Reply to Message

Eli Beckerman wrote:
> In defense of myself, I wasn't using the FOR loop to make an array
> (I just excluded all the unnecessary stuff from my posting, and
> seemingly confused everyone even further!)
>
> But clearly, I was mixing the semantics of indexing with assignment
> as Craig pointed out!
>
> FOR i=0, 999 do radius(i) = i * 0.25 is the way to go for what I
> wanted. Thanks for seeing through my morning foggyness.

Eli,

No matter how look at it, your example shows you *are* using a loop to
create an evenly spaced mesh vector. I guess I was trying to say that
using a loop is not the best way to do this in IDL. One of the wisest

pieces of advice I ever received about IDL programming was

"Try to think like an IDL programmer, not a Fortran or C programmer".

In this spirit, I submit that the following method is preferable:

```
nx = 1000    ; number of values required
dx = 0.25    ; step size
x1 = 0.0     ; start value
radius = lindgen(nx) * dx + x1
```

You might try changing the number of values to 10,000,000 and seeing
which method is faster.

Cheers,
Liam.

Subject: Re: FOR statement
Posted by Eli Beckerman on Fri, 13 Apr 2001 20:17:35 GMT
View Forum Message <> Reply to Message

> Eli,
>
> No matter how look at it, your example shows you *are* using a loop to
> create an evenly spaced mesh vector. I guess I was trying to say that
> using a loop is not the best way to do this in IDL. One of the wisest
> pieces of advice I ever received about IDL programming was
>
> "Try to think like an IDL programmer, not a Fortran or C programmer".
>
> In this spirit, I submit that the following method is preferable:
>
> nx = 1000   ; number of values required
> dx = 0.25   ; step size
> x1 = 0.0   ; start value
> radius = lindgen(nx) * dx + x1
>
> You might try changing the number of values to 10,000,000 and seeing
> which method is faster.
>
> Cheers,
> Liam.

Well, here's my FOR loop with some background info... The radius
vector was what was hanging me up -- it was not what the FOR loop was FOR!

I can't reason out any way of doing this without the loop, but I

imagine it can be done...

```
dist = sqrt((x-xc)^2 + (y-yc)^2)    ;xc & yc are the center positions
bin = 0.25                          ;of a distribution of points
psf = histogram(dist,binsize=bin)   ;psf is the radial profile
norm = total(psf)
maxR = max(dist)

ee=fltarr((maxR/bin)+1)
rval=fltarr(maxR/bin)+1)

FOR i=0.0, maxR/bin DO BEGIN
  radius = i * bin
  insidecounts = n_elements(where(dist le radius))
  rval(i) = radius
  ee(i) = insidecounts/norm        ;to establish the number of counts
                                   ;within a given radius
ENDFOR
```