
Subject: Programming style

Posted by [lars](#) on Wed, 18 Jan 1995 15:33:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Does anybody out there know if IDL passes arguments to functions by value or by reference? I have developed a programming-style where I tend to pass huge datastructures as parameters and not through common blocks. This is obviously inefficient in the pass-by-value-case, so I might have to change my style.

Greetings,

Lars

Subject: Re: Programming style

Posted by [J.D. Smith](#) on Fri, 15 May 1998 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Patrick V. Ford wrote:

>
> I have a general style/algorithm question.
>
> I want to plot in a 3-D array an ellipsoid within an in ellipsoid where the
> voxels between the boundaries are non-zero and else where zero.
>
> The general function is $(x/a)^2 + (y/b)^2 + (z/c)^2 = 1.0$
>
> I have already done this using for loops and conditional statements but it
> occurred to me that there may be some IDL matrix-boolean logic combination
> that could accomplish this in a faster and more 'elegant' fashion.
>
> I am now open for suggestions?
>
> Thanks in advance.
>
> Patrick Ford, MD
> Department of Radiology
> Baylor College of Medicine
> pford@bcm.tmc.edu

How about:

recast as:

Eq. 1: $z^2 + (x/e11)^2 + (y/e21)^2 z^2 = c1^2$ (outer ellipse)

Eq. 2: $z^2 + (x/e12)^2 + (y/e22)^2 z^2 = c2^2$ (inner ellipse)

let your array be nx by ny by nz.

Then:

```
z=findgen(nx*ny*nz)
x2=((z mod (nx*ny)) mod nx-nx/2)^2
y2=((z mod (nx*ny))/nx-ny/2)^2
z2=((temporary(z)/(nx*ny)) - nz/2)^2
array=bytarr(nx,ny,nz)
array[where(z2+x2/e11^2+y2/e21^2 le c1^2 AND z2+x2/e12^2+y2/e22^2 ge
c2^2)]=1b
```

Note the two ellipses are centered on the midpoint of the array and are concentric. This can be modified by changing the subtracted value in each of x2,y2,z2. Definitely faster than loops. Elegance is in the eye of the beholder, though.

NB: The x,y, and z index vectors must be floats, since for 3-d data, indices get large pretty quick. E.g. 100x100x100 would choke with longs (since $100^3 = 10^6 = 2^{19.86}$). This introduces some "fuzziness" at the boundaries due to roundoff. You can throw in a floor() statement to eliminate this if you really want.

JD

--

J.D. Smith	*	WORK: (607) 255-5842
Cornell University Dept. of Astronomy	*	(607) 255-4083
206 Space Sciences Bldg.	*	FAX: (607) 255-5875
Ithaca, NY 14853	*	

Subject: Re: Programming style
Posted by [steinhh](#) on Sat, 16 May 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

J.D. Smith wrote:

```
> z=findgen(nx*ny*nz)
> x2=((z mod (nx*ny)) mod nx-nx/2)^2
> y2=((z mod (nx*ny))/nx-ny/2)^2
> z2=((temporary(z)/(nx*ny)) - nz/2)^2
> array=bytarr(nx,ny,nz)
> array[where(z2+x2/e11^2+y2/e21^2 le c1^2 AND z2+x2/e12^2+y2/e22^2 ge
> c2^2)]=1b
>
> Note the two ellipses are centered on the midpoint of the array and are
> concentric. This can be modified by changing the subtracted value in
```

> each of x2,y2,z2. Definitely faster than loops. Elegance is in the eye
> of the beholder, though.

My \$0.02 modification (speed/elegance?)

```
x2 = rebin((findgen(nx,1,1)-nx/2)^2,nx,ny,nz)
y2 = rebin((findgen(1,ny,1)-ny/2)^2,nx,ny,nz)
z2 = rebin((findgen(1,1,nz)-nz/2)^2,nx,ny,nz)
array = z2 + x2/e11^2 + y2/e21^2 LE c1^2 AND $
      z2 + x2/e12^2 + y2/e22^2 GE c2^2
```

In case nx/ny/nz are large, this should save considerable time..

Since we're not using such huge indices any longer, we could also substitute FINDGEN with INDGEN (if you don't need fractional pixel positioning of the zero point (center) of the x2,y2,z2 arrays).

Stein Vidar

Subject: Re: Programming style
Posted by [pford](#) on Sat, 16 May 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <355CAD61.6BF75748@astrosun.tn.cornell.edu>, "J.D. Smith" <jdsmith@astrosun.tn.cornell.edu> wrote:

> Patrick V. Ford wrote:
>>
>> I have a general style/algorithm question.
>>
>> I want to plot in a 3-D array an ellipsoid within an in ellipsoid where the
>> voxels between the boundaries are non-zero and else where zero.
>>
>> The general function is $(x/a)^2 + (y/b)^2 + (z/c)^2 = 1.0$
>>
>> I have already done this using for loops and conditional statements but it
>> occurred to me that there may be some IDL matrix-boolean logic combination
>> that could accomplish this in a faster and more 'elegant' fashion.
>>
>> I am now open for suggestions?
>>
>> Thanks in advance.
>>
>> Patrick Ford, MD
>> Department of Radiology

```

>> Baylor College of Medicine
>> pford@bcm.tmc.edu
>
> How about:
>
> recast as:
>
> Eq. 1:  $z^2 + (x/e11)^2 + (y/e21)^2 = c1^2$  (outer ellipse)
> Eq. 2:  $z^2 + (x/e12)^2 + (y/e22)^2 = c2^2$  (inner ellipse)
>
> let your array be nx by ny by nz.
>
> Then:
>
> z=findgen(nx*ny*nz)
> x2=((z mod (nx*ny)) mod nx-nx/2)^2
> y2=((z mod (nx*ny))/nx-ny/2)^2
> z2=((temporary(z)/(nx*ny)) - nz/2)^2
> array=bytarr(nx,ny,nz)
> array[where(z2+x2/e11^2+y2/e21^2 le c1^2 AND z2+x2/e12^2+y2/e22^2 ge
> c2^2)]=1b
>
> Note the two ellipses are centered on the midpoint of the array and are
> concentric. This can be modified by changing the subtracted value in
> each of x2,y2,z2. Definitely faster than loops. Elegance is in the eye
> of the beholder, though.
>
> NB: The x,y, and z index vectors must be floats, since for 3-d
> data, indices get large pretty quick. E.g. 100x100x100 would choke with
> longs (since  $100^3 = 10^6 = 2^{19.86}$ ). This introduces some "fuzziness"
> at the boundaries due to roundoff. You can throw in a floor() statement
> to eliminate this if you really want.
>
> JD
>
> --
> J.D. Smith |*| WORK: (607) 255-5842
> Cornell University Dept. of Astronomy |*| (607) 255-4083
> 206 Space Sciences Bldg. |*| FAX: (607) 255-5875
> Ithaca, NY 14853 |*|

```

Interesting. I am going to have to think about this since it is not immediately obvious to me. The maximum size of the cube I am working with is 64X64X64 (x8 since it is a cyclical dynamic objec) with the actual object a lot smaller. I am actually using a hemi ellipsoid to grossly represent the heart muscle of the left ventricular of the heart in a radionuclide myocardial perfusion image. Fuzziness for this 'model' is actually an advantage.

--
Patrick Ford
pford@bcm.tmc.edu
