Subject: Please help me avoid loops and conditionals
Posted by pford on Tue, 09 Sep 2003 16:44:34 GMT
View Forum Message <> Reply to Message

Greetings,

I use IDL just frequently enough to know that my ingrained programming style is sub optimal for IDL but not frequently enough to clearly see how to improve it. What I want to do is have one object inside the other, in this case two concentric ellipses, and fill them with different values. This function will be invoked thousands of times if not more, so any small improvement here will show significant..

In the example below I see 3 items that sould slow this down, the array declaration, the loops and the conditional statements. (Note: I have not tried running this yet since I don't currently have access to the machine with IDL, so there are likely typo bugs, etc.)

```
function elp2, a, b, box_dim, vval, e_a,e_b, I_ratio
x_box = box_dim/2
box = intarr(box_dim,box_dim)
o_val = fix(vval / I_ratio)
v = fix(vval)
for i = 0, box_dim-1 do begin
    for j = 0, box_dim-1 do begin
        x = float(i - x_box)
        y = float(j - x_box)
        if( ((x/(a+e_a))^2 + (y/(b+e_b))^2) LE 1.0) then $
  if( ((x/a)^2 + (y/b)^2) LE 1.0) then box(i,j) = o_val $
  else box(i,j) = v
 endfor; j = 0, box_dim-1 do
endfor; i = 0, box_dim-1 do
return, box
end
```

So how do I go about converting this into a Boolean matrix operation that avoids all of this? Would it be faster to create a mask array such as:

```
 x = float((indgen(box_dim) ï¿½ box_dim/2) # replicate(1, box_dim))
 y = (transpose(x) / b)^2
 x = (x / a)^2
       mask  =  where((x + y) LE 1.0)
```

?

Thanks