Subject: FOR loops and efficiency
Posted by Rachel on Thu, 21 May 2009 16:04:43 GMT
View Forum Message <> Reply to Message

It was impressed upon me sometime or another that the use of FOR loops in IDL applications tends to be an inefficient solution for doing many tasks, yet sometimes I have difficulty finding a reasonable alternative to the FOR loop. I was wondering if anyone could give me advice on the following example code.

I am trying to make a function that takes arrays of parameters and then generates a mathematical model. In the following example I use gaussian curves, but generally I would want to expand an implementation to other mathematical functions (gaussians are just easy for this example).
So basically I can accomplish what I want to do using something like the following:

```
x = findgen(2000)*0.1 + 900.0
y = fltarr(2000)+1.0

lam0 = findgen(10)*50.0 + 900.0
depth = findgen(10)/10.0
width = findgen(10)

for i = 0,n_elements(lam0)-1 do y = y *(1.0 - depth[i]*exp(-(x-width
[i])^2/2.0/width[i]))
```

I was thinking about how one might accomplish the same things without a for loop and I came up with the following... problem being that for large arrays of lam0 this is actually more inefficient (I'm assuming because of the use of extraordinarily large arrays)

```
n = n_elements(x)
nlines = n_elements(lam0)
y = product(1.0 - rebin(transpose(depth),n,nlines)*exp(-(rebin
 (x,n,nlines)-rebin(transpose(lam0),n,nlines))^2/2.0/rebin(tr anspose
(width),n,nlines)),2)
```

any advise?

Thanks!
Josh