
Subject: Re: FOR loops and efficiency

Posted by [Craig Markwardt](#) on Fri, 22 May 2009 06:38:47 GMT

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

On May 21, 12:04 pm, Rachel <lotsofche...@gmail.com> wrote:

> 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)

>

A FOR loop will only be slow(er) when the time spent executing the
loop overhead is much more than the time spent doing the computations
in one loop iteration. A simple test would be to execute a dummy
loop:

```
NMAX = 100000L
```

```
FOR I = 0L, NMAX do begin & dummy = 1
```

Keep raising the value of NMAX until the execute time of the loop is

perceptible. Don't bother trying to optimize loops smaller than this.

In your case, you are only doing ten iterations, and each iteration does a lot of work, so you won't gain by removing the loop.

Craig
