Subject: Re: multiplication
Posted by Craig Markwardt on Tue, 28 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Carsten Dominik <dominik@astro.uva.nl> writes:
>
> Well, it depends very much on the size of the array. Loops in IDL are
> indeed very slow. Try the following: Set N to a large number
> (e.g. 10 000 000) and execute the following lines:
>
> x=fltarr(n)*0.+1.000001 & p=1 & for i=0.,1.*n_elements(x)-1 do p=p*x[i] & print,p
>
> x=fltarr(n)*0.+1.000001 & p=exp(total(alog(x)))&print,p
>
> You'll get a surprise, I promise.

One way to speed things up is to use some sort of a divide and conquer algorithm. Which is to say, divide the array into two segments and multiply them element-by-element. Keep doing this until you get down to a single element.

```
FUNCTION CMPRODUCT, ARRAY
      X = ARRAY
      N = N_ELEMENTS(X)
      WHILE N GT 1 DO BEGIN
         IF (N MOD 2) EQ 1 THEN X(0) = X(0) * X(N-1)  ;; When N is odd!!
         N2 = FLOOR(N/2)
         X = X(0:N2-1) * X(N2:*)  ;; Don't worry if N is odd here.

         ;; X keeps shrinking by a factor of two each time
         N = N2
      ENDWHILE
 RETURN,X(0)
END
```

Disadvantages are that it may be slower when n_elements(array) is small. Also, the round-off error can grow to significance, as I think Carsten was trying to say, but this will happen with most approaches unfortunately. Double precision can help.

Craig

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