Subject: A more efficient way of multiplying this
Posted by stefano.rgc on Sun, 02 Aug 2015 02:44:38 GMT
View Forum Message <> Reply to Message

Hello, I have learnt a lot in this site. Now I will ask my first question.

Imagine you have a vector [x,y,z] and an array [[A1,A2,A3,...,An],[B1,B2,B3,...,Bn],[C1,C2,C3,...,Cn]]. What I need is multiply x by the vector [A], y by vector [B] and z by vector [C]. Currently I do this with a "for loop" (just 3 loops in this example) but this is very inefficient when the dimensions get bigger.

Some of you know a way to do this without resorting to a "for loop"?

I will appreciate any reply. Thanks.

Subject: Re: A more efficient way of multiplying this
Posted by Heinz Stege on Sun, 02 Aug 2015 13:14:48 GMT
View Forum Message <> Reply to Message

Hello,

I modified Nikola's Code a little (see below). First I excluded the creation of the arrays a and b from the time measurement. Second I added the Reform/Rebin method. You can find a lot of discussions in other threads of this group and in the internet. (Thanks to JD Smith and David Fanning.)

Here are my results:

```
IDL> .run multiply_arrays_example
% Compiled module: $MAIN$.
Slow Loop:       7.8099990
Fast Loop:      0.48500013
"# operator":   0.31200004
reform/rebin:   0.14099979
```

and here is the file multiply_arrays_example.pro:

```
a = randomu(seed,3)
b = RANDOMU(seed, 150, 3)

seconds0 = SYSTIME(1)
for k=0,9999 do begin   ; "Slow Loop" (scaled by 10)
  c1 = FLTARR(150, 3)
  FOR i = 0, 2 DO FOR j = 0, 149 DO c1[j, i] = a[i]*b[j, i]
  end
seconds1 = SYSTIME(1)
```

```
for k=0,99999 do begin   ; "Fast Loop"
  c2 = FLTARR(150, 3)
  FOR i = 0, 2 DO c2[*, i] = a[i]*b[*, i]
  end
seconds2= SYSTIME(1)

for k=0,99999 do begin   ; "# operator"
  aa = (FLTARR(150)+1)#a
  c3 = aa*b
  end
seconds3 = SYSTIME(1)

for k=0,99999 do begin   ; "reform/rebin"
  c4 = rebin(reform(a,1,3),150,3,/sample)*b
  end
seconds4 = SYSTIME(1)

print, 'Slow Loop:   ', (seconds1-seconds0)*10
if array_equal(c2,c1) then print, 'Fast Loop:   ', seconds2-seconds1
if array_equal(c3,c1) then print, '"# operator":', seconds3-seconds2
if array_equal(c4,c1) then print, 'reform/rebin:', seconds4-seconds3

end   ; end of file multiply_arrays_example.pro
```

Cheers, Heinz

Subject: Re: A more efficient way of multiplying this
Posted by stefano.rgc on Mon, 03 Aug 2015 02:00:50 GMT
View Forum Message <> Reply to Message

Thanks for the quick replies.

@karo03de. What you write is basically the same I wrote: "Currently I do this with a "for loop" (just 3 loops in this example)"

@Nikola Vitas. Thank you for the "# operator" way. It was very instructive.

@Heinz Stege. As you did, I also had figured the way with rebin/reform but I was afraid of how big the "rebin(reform(a,1,3),150,3,/sample)" variable could have been; that because the actual dimensions were 7000 and 5000 instead of 3 and 150 (as you used). But now I realized the amount of memory is only 3 MB so it is such not a big deal. Also thanks for the /sample keyword; I didn't know it.

I will manage it with the rebin/reform way as @Heinz Stege suggested it. But I thought that there

was a way to do this without the need of resorting to "copy" something many times, as it does rebin or the line "aa = FINDGEN(100)#(FLTARR(150)+1)" at @Nikola Vitas code.

Thanks again.
