Subject: Re: simple vectorizing problem
Posted by JD Smith on Thu, 11 Sep 2008 18:56:50 GMT
View Forum Message <> Reply to Message

On Sep 10, 7:53 pm, dpm314 <meich...@gmail.com> wrote:
> Hi everyone and thanks a lot for your suggestions so far. That is a
> cute trick with the Total function, and I was just not aware that you
> could specify to total only rows or columns like that. This will
> work, and I have tried it but didn't get a chance to do the timing
> test.
> However, my first post must not be clear, because this doesn't really
> answer my question.
>
> I have these image arrays and need to perform several functions on
> them which return a scalar (or a 6 element array) for each image in a
> large stack. Is there *in general* a way to so something like:
>
> results = fltarr(num_images)
> results = myFunctThatDoesSomeAnalysis(image)
> ;now results holds an array of scalars for the result of
> myFunctThatDoesSomeAnalysis() on each image
>
> without either using a loop here or inside the analysis function
> iterating over each frame in the image?
>
> again, I've tried many things like:
>
> a = indgen(num_image)
> results(a) = myFunctThatDoesSomeAnalysis(image(*,*,a))
> which I thought would work but it does not.
>
> It is necessary to use a loop like this then in general :
> for i 0, num_images-1 do results(i) =
> myFunctThatDoesSomeAnalysis(image(*,*,i))
>
> ???
>
> I thought the trick with an indexing array ('a' in the example above)
> would work because if I say
> b = indgen(101)/(!pi/100)
> print, cos(b)
>
> I get not one number out but an array of cos() evaluated from 0 to pi
>
> I guess I don't understand how the cos() example gives me an array
> back, and
> myFunctThatDoesSomeAnalysis(image(*,*,a)) gives me just one number.
>

> If this question makes no sense I apologize, and I could provide more
> code to illustrate the problem if that would help. I am still trying
> to wrap my head around vectorizing code, which (I at least) find quite
> different from purely procedural languages like c and Fortran which
> I've written in for years.

There is no general mechanism to "thread" arbitrary code over arbitrary dimensions within an array in IDL. Some languages have such general operator threading (but they're not exactly fast). There *are* a variety of internal IDL routines which can do such threading, including TOTAL, and various others, which can be used together when vectorizing many types of problems (certainly more than are immediately obvious). Another way of saying that is that all IDL vectorization, aside from basic array arithmetic, is explicit, in the sense that it is specifically requested in the called procedure or function (like TOTAL(array, dimension)).

There is such a thing as premature vectorization, however. In your case, if the images in your stack are big enough (at least few hundred or thousand pixels), simply looping through each one and accumulating the result will likely perform just as well as any vectorized method could. It's when you're looping over individual array elements that you should be concerned.... "Embodiment of Pure Evil" notwithstanding.