
Subject: Vectorize procedure

Posted by [Romolo Politi](#) on Thu, 10 Dec 2009 14:57:52 GMT

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

Hi,

I'm vectorizing a procedure in order to delete three for loops and increase the speed.

I have to calculating the interpolation on a regular grid.

newy=interpol(y,x,newx,/spline)

in the loop manner for each index of the loop there are three different arrays .

in the vectorize manner I have three 3D matrices, But I do not found a way to calculate the interpolation matrix.

Any one have suggestions?

Thanks

Romolo

Subject: Re: Vectorize procedure

Posted by [Jeremy Bailin](#) on Fri, 11 Dec 2009 16:20:13 GMT

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

On Dec 10, 3:00 pm, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:

> In article

> <b599bcd-fd42e-45d8-b643-7ea4620fd...@k19g2000yqc.googlegroup s.com > ,

> Romolo Politi <romolo.pol...@gmail.com> wrote:

>

>> Sorry for the not clarity.

>> my situation is

>> x=fltarr(3200,201,201)

>> y=fltarr(3200,201,201)

>> newx=fltarr(799,201,201)

>> newy=fltarr(799,201,201)

>> for i=0,200 do begin

>> for j=0,200 do begin

>> newy[* ,i,j]=interpol(y[* ,i,j],x[* ,i,j],newx[* ,i,j],/SPLine)

>> endfor

>> endfor

>

> You are doing ~40,000 separate spline fits. I don't think there is

> any way to vectorize this, as each fit is an independent problem.

> Spline fitting is a coupled problem, so you cannot, for example, simply

> string your segments together into one long fit.

>

> If the newx[* ,i,j] are the same for all i and j, you can probably get

> a significant speed-up by calling SPL_INIT once and then calling

> SPL_INTERP for each i and j. Since you are storing newx separately
> for each i and j, I gather this is not the case.
>
> Your program is accessing memory efficiently, but you will get a small
> speed up by saying
>
> newy[0,i,j]=interpol(y[* ,i,j],x[* ,i,j],newx[* ,i,j],/SPLine)
>
> You might also try the other built in spline functions: SPLINE,
> SPLINE_P, SPL_INIT and SPL_INTERP to see whether one is faster.
>
> Finally, I notice that you are downsampling your data substantially.
> Are you sure you really need to use a spline fit to do that?
>
> Ken Bowman

You might be able to fake something in the "stringing the segments together" vein by putting some padding in between the segments that does a linear interpolation between the endpoint of one segment and the beginning of the next, enough padding to make the last old segment and first new segment independent. Of course, you'll get a different answer for those end segments because of that padding than if they didn't exist, so that may not work depending on how much those points are offset, but it should at least be better than sticking them next to each other. What boundary conditions does INTERPOL use with spline fits? Maybe you can design padding that effectively gives you the same boundary conditions?

-Jeremy.
