Subject: Re: Vectorize procedure
Posted by Kenneth P. Bowman on Thu, 10 Dec 2009 20:00:41 GMT
View Forum Message <> Reply to Message

In article
<b599bcdf-d42e-45d8-b643-7ea4620fd7ed@k19g2000yqc.googlegroups.com>,
 Romolo Politi <romolo.politi@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