Subject: Re: Challenging question - array curve fitting
Posted by Craig Markwardt on Mon, 23 Apr 2007 02:49:56 GMT
View Forum Message <> Reply to Message

Sorry for neglecting your post for so long!

"Qing" <csis@bigpond.net.au> writes:
> Hello Craig,
>
> Thanks a lot for your comments and tips. It is intriguing for
> "grouping multiple time series into a single large vector...". I can
> manage to transform/ reform the data array into a large vector, but
> my brain just can't think of a way to model the concatenated vector
> independently. For example, I am using a Gaussian curve model with 3
> fitting parameters for each curve. Typically Nx=Ny=128 and the
> number of time points Nt=60. The thing is that my computer has two
> CPUs, and it only uses about 50% total CPU when fitting the curve by
> looping through each pixel.

Your model function would still need to compute each light curve
separately, which may involve a loop. But, for example, you could
loop over time sample instead of light curve number, and in each
iteration compute 128x128 model values at once (or fewer).

Example:
```
; Compute NX x NY x NT light curve samples
; Model is simple linear P0 + P1*T
; Parameters are arranged like this:
;    P0 = P(0:(NX*NY-1))  ;; For each pixel
;    P1 = P(NX*NY:*)      ;; For each pixel
function lcmod, t, p, nx=nx, ny=ny
  ntot = nx*ny
  p0 = reform(p(0:ntot-1),nx,ny)
  p1 = reform(p(ntot-1:*),nx,ny)
  nt = n_elements(t)
  model = fltarr(nx,ny,nt)
  for i = 0, nt-1 do model(0,0,i) = p0 + p1*t(i)
  return, model
end
```

This only works because NX*NY is much larger than NT.

> I though usually array operation is more efficient than looping
> throug all elements individually, but I was not sure if that is the
> case for a non-linear fitting task. Or at least, using array
> operation can get better use of the CPUs upto 100%. Do you thing
> using a large vector would be as efficient as using array?

It all depends on how much work is done per iteration of the loop. If you can accomplish a lot of work in one iteration, then you will not save by vectorizing the loop. Since MPFIT has a lot of set-up and tear-down expenses, then I suspect you could indeed gain by grouping a several time series together.

> Why does "the number of arithmetic operations required to perform
> the fit scales as the number of pixels *cubed*"?, I thought it would
> be a linear relation if using array just like looping through all
> pixels one-by-one. Am I missing something?

Actually it scales as M N^2 where M is the number of data points and N is the number of parameters. However, since this example involves grouping independent light curves with independent parameters into one block, M is also proportional to N, hence an overall N^3 dependence.

Hope you succeeded!
Craig

--

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