Subject: for loops for total newbies
Posted by Anna on Fri, 13 Jul 2007 13:54:57 GMT
View Forum Message <> Reply to Message

Dear experts,

[skippable] I'm a physics student currently working on my bachelor thesis research. To prepare me for the programming aspects, I've been taught C/C++ by an assistant professor in C.S. who has advised a gazillion Master students on their thesis, as well as several Ph.D's, and can spot a syntax error from across the computer lab, and LabView by a grad student in physics whom we suspect was caught while getting coffee and forced into teaching the course although he unfortunately doesn't speak the language in which the manual was written (our mother tongue), and prefers to keep his comments to "Your program, it does not work".

We suspect that the philosophy behind these teaching methods is that we're now supposed to be able to work with all languages by applying linear combinations of our experiences. However, I am now supposed to use IDL to analyse my raw data, and I'm stuck. I've tried asking some astronomy majors but they turned completely white and ran away. [/ skippable]

For my research project, I have to analyse sets of png images (each set consisting of 1000 210x820 images). I have an IDL program which can be coerced into working on a good day, but running it over the data of one day takes a night and a morning, which means that I can't measure the next morning (never mind what the hardworking sysadmins think of me). I thought I could combine two things: optimising my program, and teaching my C-trained, for-loop-loving mind the power of IDL by asking you to look at two pieces of code:

```
====================================
cor_row= fltarr(max_cor_shift)
   for i = 0, number_of_images-3 do begin
      print, i
      for j = 0, length-1 do begin
            for lag = -10, 10 do begin
            cor_row(lag+10)= c_correlate(data(i
+1,*,j),data(i,*,j),-lag)
         endfor
         cor_array(i,j,*)=cor_row
      endfor
   endfor
====================================
```

'data' is an array containing all images, of dimension

data(number_of_images, width, length) = (1000, 210, 820).
'cor_array' is supposed to contain all correlation data at the end,
and has dimenson (number_of_images - 2, lenght, max_cor_shift).
max_cor_shift is a constant and in this case 21.

This is reasonably fast, but I'd like to know whether it can be
written differently, both to make it faster and as a learning
experience...

The real cycle eater is the fitting function:

```
==========================
crtot= fltarr(sz-1,(size(cor_array))[2],(size(cor_array))[3])
crtot(0:sz-3,*,*)= cor_array(0:sz-3,*,*)

meanarr = fltarr((size(cor_array))[2])
j= fltarr(1000*max_cor_shift)
for i=0,((1000*max_cor_shift)-1) do j(i)=i/1000.
avgz= fltarr(sz-2)
fit = fltarr(sz-2, max_cor_shift*1000)

for i = 0, (size(cor_array))[2]-1 do begin
 print, i
 for f=0, number_of_images-3 do begin
  wait,.0001
  maxcr=max(crtot(f,i,*),k)
  k2=(k+1+max_cor_shift) mod max_cor_shift
  k1=(k-1+ max_cor_shift) mod max_cor_shift
  a= crtot(f,i,k2)/2+ crtot(f,i, k1)/2- maxcr
  b= crtot(f,i,k2)/2- crtot(f,i, k1)/2
  c= maxcr
  fit(f,*)= a*(j-k)^2 + b*(j-k) + c
  kaas= max(fit(f,*), s)
  avgz(f)= s/(float(sz)) -10
 endfor
 meanarr(i)=mean(avgz)
endfor
===============================
```

This takes about two hours per data set. I tried averaging earlier,
but that had too much of an impact on the quality of the fits.

If you could please help me, I'd be very, very much obliged!

Thank you,

Anna Baas