Subject: Re: All day FFT....
Posted by Robert Stockwell on Wed, 06 Feb 2002 23:30:04 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:

> Hey there,
>
> I've had this process running for about, oh, 4-5 hours now where I'm reducing the resolution of
> an input spectrum of about 500K points (i.e. a lot, but not terribly so.). Earlier I did the
> same for a spectrum of about 700K points. Something has apparently gone haywire in the
second
> one - is there anything I can do to interrupt the process, check out some numbers to see if
> everything is o.k. and if it is, start it up where it left off? I think that's what a ^C does
> but I thought I better consult the idl wizards out there...
>
> thanks for any info.
>
> paulv

I'm betting that ^C will interrupt the process as soon
as it finishes with the FFT. LOL! I'd just kill IDL.

My guess, regarding slowness, is "Prime Number" (number of points
in your time series). Try zeropadding up to, or truncating
down to, a nice factorable number.

I've attached my hackware factors.pro which will return the
factors of a number. (and its recursive, COOL!)`

FYI, 500K should take seconds: Here is a quicky example:

```
IDL> a = lindgen(1025L^2)
IDL> help,a
A               LONG      = Array[1050625]
IDL> tic & b = fft(a) & toc
% Compiled module: TIC.
% Compiled module: TOC.
Elapsed time:       3.9645100 Seconds.
IDL> print,factors(n_elements(a))
% Compiled module: FACTORS.
      5.00000      5.00000      5.00000      5.00000      41.0000      41.0000
```

Cheers,
bob stockwell

```
; do factoring of a function

; development interrupted when i realized I didn't need it
; drops the last number

function factors, n,prevfactors=prevfactors


maxfactor = fix(sqrt(n))
if maxfactor le 1 then begin
 if keyword_set(prevfactors) then begin
   prevfactors = [prevfactors,n]
   return,n
 endif else begin
  return,n
 endelse
endif

fac = findgen(maxfactor-1)+2 ; 2 -- sqrt(n)

doloop = 1
factorflag = 0
counter = 0


while doloop do begin
 if n mod fac(counter) eq 0 then begin
  factorflag = 1
  newfactor = fac(counter)
  if keyword_set(prevfactors) then prevfactors = [prevfactors,newfactor] $
   else prevfactors = newfactor
  newnumber = n/newfactor
  ; to iterate is human, to recurse is divine
  r = factors(newnumber,prevfactors=prevfactors)
  doloop = 0
 endif
 counter = counter+1
 if counter ge maxfactor-1 then doloop = 0
endwhile
```

```
if n_elements(prevfactors) eq 0 then prevfactors = n else begin
 ; only if n is prime do we add it here
 if not(factorflag) then prevfactors = [prevfactors,n]
endelse



return,prevfactors


end


;;;;_____ test code here _____

n = 5001

r = factors(n)


print
print
print,'Finished calculating factors_____'
print,'Number: ',n
print,'Factors:'
print,r



end
```

File Attachments

1) factors.pro, downloaded 88 times