Subject: Re: Plotting one point per loop
Posted by Paul Van Delst[1] on Wed, 08 Jul 2015 18:33:45 GMT
View Forum Message <> Reply to Message

On 07/08/15 13:52, wdolan@oxy.edu wrote:
> Hi Paul,
>
> It is set to plot to a postscript file. That seems like a good idea,
> but I am not sure how to store those values until the end. Because if
> I did an array, I would have to specify a size, and each run has a
> different number of scans.

Fair enough.

Craig answered your actual question (thank goodness! :o). I went off on a tangent about efficient ways to increase array sizes when you don't know the final size, so that you can then plot (mostly because you said you were a new user :o)

Keep reading only for sh*ts and giggles.

As an exercise (my actual work today is a bit repetitive) I came up with the code way down below. When I run it I get the following:

```
IDL> .run blah.pro
% Compiled module: $MAIN$.
% Time elapsed: 16.238165 seconds.
X               FLOAT     = Array[300000]
% Time elapsed: 0.20202398 seconds.
X               FLOAT     = Array[300000]
```

The doubling method is pretty efficient. Way better than simple concatenation. If you increase the number of scans to 1000000 then the first method will be done sometime tomorrow....

```
; The (generally unknown) number of scans for this example
n_scans = 300000L


; METHOD #1: CONTINUALLY CONCATENATE ONTO ARRAY

; Specify an empty array
x = []

; Loop over your (unknown) number of scans
scan_count = 0
```

```
tic
repeat begin
   ; Keep track of the scan count
   scan_count++
   ; Accumulate an array of numbers to plot
   x = [x,randomn(seed,1)]
endrep until scan_count eq n_scans
toc
help, x


; METHOD #2: DOUBLE SIZE OF ARRAY AS NEEDED
; (HAM FISTED CODE, BUT YOU GET THE IDEA)

; Specify an empty array and initial size
n_size = 10000L
x = fltarr(1000)

; Loop over your (unknown) number of scans
scan_count = 0
tic
repeat begin
   ; Keep track of the scan count
   scan_count++
   ; Double array size if necessary
   if ( n_elements(x) lt scan_count ) then begin
     n = n_elements(x)
     x = [temporary(x),fltarr(n)]
   endif
   ; Insert value into array
   x[scan_count-1] = randomn(seed,1)
endrep until scan_count eq n_scans
; Truncate array as necessary
x = x[0:scan_count-1]
toc
help, x

end
```

cheers,

paulv