Subject: Re: how to calculate a running total of a vector
Posted by VUKOVIC on Sun, 04 Dec 1994 16:54:55 GMT
View Forum Message <> Reply to Message

In <CHASE.94Dec2163443@retro.jhuapl.edu> chase@retro.jhuapl.edu writes:

>
> Mirko> Suppose I have a vector v. I want to generate a vector vv
> Mirko> whose i-th element is a sum of the first i elements of v. How
> Mirko> to do it fast -- without loops?
>
>
> This will work, but it is overkill also. I probably does the same
> amount of computations as your matrix multiply solution.
>
> running_sum = (convol([0,v],replicate(1.,n_elements(v)),center=0,/edge_tru n))(1|
> :*)
>
Tried your suggestion, and it works. But it is damn slow compared to a simple
loop algorithm (.4 vs 4 sec for a 1024 length vector on an old vaxstation):

```
FUNCTION RSUMV,V
;+
; Produces a running total of a vector where result(i) is total(v(0:i))
;-

nn=n_elements(v)
res=fltarr(nn)

res(0)=v(0)
for ii=1,nn-1 do res(ii)=res(ii-1)+v(ii)

return,res
end
```

My final thought is to go maybe fia the FFT. One can maybe approximate the sum
by an integral, and do the integration fia FFT.

```
FFT(FFT(V,-1)/W,1)
```

where W is a frequency like vector. I may try that later.

Mirko

Mirko Vukovic | vukovic@uwmfe.neep.wisc.edu
Dept. of Nucl. Eng. and Eng. Phys. | mvukovic@wiscmacc.bitnet
U of Wisconsin -- Madison | phone: (608) 265-4069
1500 Johnson Drive; Madison WI 53706| fax: (608) 265-2364

Subject: Re: how to calculate a running total of a vector
Posted by sjt on Mon, 05 Dec 1994 10:33:15 GMT
View Forum Message <> Reply to Message

Chris Chase S1A (chase@retro.jhuapl.edu) wrote:

: In article <3bnmri$e91@news.doit.wisc.edu> VUKOVIC@uwmfe.neep.wisc.edu (Mirko Vukovic) writes:

: Mirko> Suppose I have a vector v. I want to generate a vector vv
: Mirko> whose i-th element is a sum of the first i elements of v. How
: Mirko> to do it fast -- without loops?

: Mirko> I thought of multiplying a matrix with 1's on and bellow its
: Mirko> diagonal by v. That would do it, but seems like a bit of
: Mirko> overkill.

: This will work, but it is overkill also. I probably does the same
: amount of computations as your matrix multiply solution.

: running_sum = (convol([0,v],replicate(1.,n_elements(v)),center=0,/edge_tru n))(1:*)

: Chris Chase
: --
: ============================
: Bldg 24-E188
: The Applied Physics Laboratory
: The Johns Hopkins University
: Laurel, MD 20723-6099
: (301)953-6000 x8529
: chris.chase@jhuapl.edu

I recall trying various things for this one a while back (on about IDL2.0.13 or thereabouts, running on a VAX) and found the quickest was a single loop:

if DA is your data array:

PS = fltarr(n_elements(DA))

ps(0)=da(0)
for j=1l,n_elements(da)-1 do ps(j)=ps(j-1)+da(j)

I know IDL loops are slow but the other methods tend to have so much overhead of memory allocation etc that it just won. (That said, I haven't tried Chris's method)

--

+----------------------+------------------------------- --+---------+
James Tappin,	School of Physics & Space Research	O__
sjt@star.sr.bham.ac.uk	University of Birmingham	-- \/`
"If all else fails--read the instructions!"	
+-- --+---------+