Subject: Re: Cumulative total
Posted by bowman on Fri, 18 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

In article <3602B1F3.210@cdc.noaa.gov>, Andrew Loughe <afl@cdc.noaa.gov> wrote:

> Nice, elegant solution, Eddie.

>> here is the way i do it:
>>
>> IDL> A = findgen(10)
>> IDL> N = n_elements(A)
>> IDL> result = A # (lindgen(N,N) ge transpose(lindgen(N,N)))
>> IDL> print,result
>> 0.00000
>> 1.00000
>> 3.00000
>> 6.00000
>> 10.0000
>> 15.0000
>> 21.0000
>> 28.0000
>> 36.0000
>> 45.0000

In my opinion this is a baroque construction that reveals a shortcoming in IDL. (Hey, not a major shortcoming. I'm not committing heresy here!)

To compute the cumulative sum of a vector of length N, i.e.,

x_cum[0] = x[0]
FOR i = 1, N-1 DO x_cum[i] = x_cum[i-1] + x[i]

should require N loads, N flops (adds), and N stores. This may not optimize well, since each result depends on the previous one, but I suspect most modern Fortran or C compilers would do pretty well.

The IDL approach above requires creating an N^2 matrix filled with integers, performing an if test on every element of that array and its transpose, and then performing a matrix-vector multiply (N^2 multiplies and N^2 adds). What do you do when N = 100,000? It seem a silly way to do a simple task.

I'm not trying to pick on Eddie. It works for him. I wonder which method is faster?

Maybe I should write a Fortran function to do it ... yuck -- highly non-portable.

So, note to RSI: Add CUMULATIVE function to next release of IDL and make it a good IDL function so that one can specify which dimension of a possibly multidimensional array to accumulate over, etc. It should be quite useful with HISTOGRAM.

Ken

--

Kenneth P. Bowman, Professor 409-862-4060
Department of Meteorology 409-862-4466 fax
Texas A&M University bowmanATcsrp.tamu.edu
College Station, TX 77843-3150 Change the AT to @
