
Subject: Re: A case where FOR loops are unavoidable?
Posted by news.qwest.net on Mon, 27 Nov 2006 23:26:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

"gknoke" <greg.knoke@gmail.com> wrote in message
news:1164665956.774256.277610@14g2000cws.googlegroups.com...
> $f(x, b1, b2, p) = \text{alog}((1/b1)*(1-p)*\exp(-x/b1) + (1/b2)*p*\exp(-x/b2))$
>
> Here x is a vector roughly of length 2^{20} , b1 and b2 are roughly
> 100-1000 elements, and p is roughly 10-50 elements.

I don't understand what you mean with:

$x/b1$

where x has a million points, and b1 has 100 points.

I guess you mean that you are creating a 4 dimensional surface,
that has dimensions of $[2^{20}, 1000, 1000, 50]$.

That would seem to require 50 terrabytes of numbers, so 200 terrabytes
of data storage or so. Do you really need all those numbers?

How are you storing the results?

Perhaps using some of the well known algorithms to search parameter
space would be more appropriate than calculating the entire thing.

Anyways, to answer your question, vectorize on the largest vectors, and
loop over the smallest vectors, (which is what you are doing by sending
'x' in as a vector). If you can fit it in memory, you can do a two
dimensional

slice by also vectorizing b1, b2, or p (which fits in memory best).

for instance (this removes 2 loops)

for k=1,np

for j=1,nb2

f_out(i,j,k) = total(f(x, b1, b2[j], p[k]), 1)

If you really need the speed, perhaps switching over to fortran would help,
since this is a very simple calculation.

Cheers,
bob
