Subject: Re: vectorising versus loops
Posted by nasalmon on Fri, 26 Mar 2004 19:34:51 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@REMOVEcow.physics.wisc.edu> wrote in message news:<ond67ozi1w.fsf@cow.physics.wisc.edu>...
> nasalmon@onetel.net.uk (Neil) writes:
> ...
>>
>> However, there is one small outstanding problem in the vectorisation,
>> and this involves vectorisation of a routine that uses the cross or
>> vector product, the IDL routine being CROSSP, generating a vector from
>> the cross product of two vectors.
>>
>> Currently i am forced to use this in a loop, as my vectorising
>> attempts (see below) have failed. The loop statement below gives the
>> correct result (i being the counter of one of the outer loops):
> ...
>
> Neil, how about using CROSSPN, a vectorized cross product function for
> just such purposes?
>
> Craig
>
> P.S. http://cow.physics.wisc.edu/~craigm/idl/idl.html (under Math)

Well thanks very much for CROSSSP, it worked like a dream, fitted in and sailed through perfectly.

In the main programme structure there were loops nested 4 deep, so vectorising the inner most loop, now leaves only 3 nested loops. Vectorisation used lots of WHERE statements. As the inner loop dealt with vectors, inner products of these could be conveniently replaced with matrix multiplications and transpositions.

However, the improvement factor in speed is only between of one and five, which is small, but certainly worth having. One of the reasons why it is this low is because there is lots of stuff in the other loops and sometimes the inner most loop is very short, may be only 3 elements, while some of the outer loops need to munch thousands of times.

So i was wondering whether there were any other tricks to vectorisation that i had missed, that i could use to improve speed? I can imagine that trying to vectorise the second to inner most loop may be possible. Are there any strategies or special routines that could be used in the vectorisation of loops higher in a hierarchy of loops?

many thanks,
Neil