Subject: Re: Help optimizing the nested for loops
Posted by Messon Gbah on Tue, 01 Mar 2005 21:16:00 GMT
View Forum Message <> Reply to Message

Hi Craig:

Thanks for your reply and suggestions.

V. Best,
Messon

Craig Markwardt wrote:
> Messon Gbah <gbah@umich.edu> writes:
>
>
>
>> Could some one help get rid of nested for loops in the following statements?
>> The original code was written in C and I'm trying to port it to IDL.
>>
>> indx = indgen(n)
>> indx[0] = 3
>> index[3] = 0
>> array = dblarr(n,n)
>> alpha = dblarr(n,n)
>> beta = dblarr(n)
>>
>> ;Loop 1
>> trace = double(0.0)
>> for j=0, n-1 do begin
>> vj = indx[j]
>> trace += alpha[vj,vj]
>> alpha[vj,vj] += flamda ;flamda = a constant
>> for k=0,j do alpha[vj,indx[k]] = alpha[indx[k],vj]
>> endfor
>
>
> You can remove one level of loops,
>
> trace = total(alpha[indx,indx])
> alpha[indx,indx] += flamda
> for j = 0, n-1 do alpha(indx[j],indx) = alpha(indx,indx[j])
>
> If you can convert to unpermuted matrices, then you can use the
> TRANSPOSE function,
> alpha_prime = (alpha[indx,*])[*,indx]
> alpha_prime_transpose = transpose(alpha_prime)
>
>

>
>
>> ;Loop 2
>> for j=0, n-1 do begin
>>   vj = indx[j]
>>   for k=0,n-1 do array[indx[k],j] =
>> alpha[indx[k],vj]/sqrt(alpha[vj,vj]*alpha[indx[k],indx[k]])
>> endfor
>
>
> Again, it's probably worth converting to unpermuted matrices.
>
> alpha_prime_diag = alpha[indx]
> for j = 0, n-1 do $
>   array_prime[*,j] = alpha_prime[*,j] / sqrt(alpha[j,j]*alpha_prime_diag)
>
>
>
>> ;Loop 3
>> for j=0, n-1 do begin
>>   vj = indx[j]
>>   b[vj] = T[vj] ;T is some init value
>>   for k=0, n-1 do begin
>>   vk = indx[k]
>>   b[vj] += beta[vk]*array[k,j]/sqrt(alpha[vj,vj]*alpha[vk,vk])
>>   endfor
>> endfor
>
>
> Again, converting to unpermuted matrices,
>
> for j = 0, n-1 do $
>   b[j] = T[j] + total(beta*array_prime(*,j)/sqrt(alpha_prime_diag[j]*alpha_p rime_diag))
>
> ...although it's a little confusing which of your matrices are
> permuted and which are not, so that may require some tweaking.
>
> Good luck,
> Craig
>