Subject: Re: Vectorizing Code
Posted by Jeff Guerber on Thu, 28 Feb 2002 00:00:15 GMT
View Forum Message <> Reply to Message

On Wed, 27 Feb 2002, Steve Jones wrote:

```
>  Is it possible to vectorize a simple double for-loop?
>
>  for i=0,nstate-1 do begin
>    for j=0,nvec-1 do begin
>      sa(i,j)=10.^2*exp(-abs(i-j)*dz/h)
>    endfor
>  endfor
>
>  I tend to write a large number of such loops and my indexes have been
>  steadily increasing of late... Is there a faster alternative?  Thanks
>  in advance
```

I'm not sure about completely vectorizing this particular case, but two things that should help:

1) Exchange your i and j loops, since in IDL as in Fortran the leftmost index varies fastest. (See the current thread titled "IDL2MATLAB".)
2) Move the invariant terms outside the loops:

```
d=lonarr(nstate,nvec)
for j=0,nvec-1 do begin
   for i=0,nstate-1 do begin
      d(i,j) = i - j
   endfor
endfor
sa = 10.^2*exp(-abs(d)*dz/h)
```

(Hmmm. You may be able to create (using indgen and replicate) two appropriate nstate-by-nvec arrays, one for i and one for j, then subtract those... In this case, I'm not sure that would be faster, though.)

By the way, ^ has higher precedence than *, so I think you may want to say 10.^(2*exp(...)) [note the extra set of parentheses] instead. What you wrote is equivalent to 100.*exp(...).

Another tip: If your indices might exceed 32767, be sure to write your loops "for i=0L, ...", otherwise the index will be an int instead of a long. That one's bitten me, and it's painful!

Jeff Guerber