
Subject: Re: Vectorizing Code

Posted by [James Tappin](#) on Thu, 28 Feb 2002 16:35:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

Marc Schellens wrote:

```
>>> sa = 10.^2*EXP(-ABS(REBIN(LINDGEN(nstate), nstate, nvec, /SAMPLE) - $
>>>   REBIN(REFORM(LINDGEN(nvec), 1, nvec), nstate, nvec, /SAMPLE))*dz/h)
>>
>> I am afraid I'd rather stick with the optimized loops than something that
>> takes me five minutes to figure out :-(
>> Pavel
>
> but unfortunately in IDL it may save you more than five
> minutes at run-time if you do figure it out...
> :-) marc
>
```

Loops aren't as bad as they used to be. Many years ago (IDL 2.x on VMS) I wrote a very clever (IMHO) routine for time-averaging irregularly sampled data which speeded things up about 10-fold. However recently (now with IDL 5.x on Linux and Solaris) it became clear that this same time averaging routine was a major bottleneck. Replacing it with a version with one more loop and many less temporary arrays speeded up 100-fold in some cases.

The moral: Nowadays loops may be slow but creating and destroying workspace arrays is often even slower.

James

--

```
+-----+-----+-----+
| James Tappin      | School of Physics & Astronomy | O__ |
| sjt@star.sr.bham.ac.uk | University of Birmingham   | -- V` |
| Ph: 0121-414-6462. Fax: 0121-414-3722      |      |
+-----+-----+-----+
```