Subject: Re: Using MIN on arrays : Exorcising loops?
Posted by A. D. &amp; J.C. Cool on Wed, 10 Oct 2001 07:35:30 GMT
View Forum Message <> Reply to Message

Hi Guys,

Well, when I say Martin's method didn't work, that's based on the test program I
was using. That was a simple thingy that defined three arrays of randomly seeded noise,
bytscaled that, then added three blotches of recognisable clutter - radar's the game here.

I displayed the calculated "minimum" of the three arrays using the simple double loop
method, and then in another window the "minimum" using Martin's method. The results
were clearly visually not equivalent.

Dropping in Craig's method where he initialises min_array to data_array(*,*,0) produced
the same results visually, and this was confirmed independently using WHERE to check
the two "minimum" arrays.

I didn't investigate further, as Craig's method is also more suitable for other circumstances
where we need to find the minimum from up to 20 arrays sized 640x500.

For interest, use of the < operator speeds up the process by a factor of almost 22 whether
you're using 3 arrays or 20.

Andrew Cool
andrew.cool@dsto.defence.gov.au
Defence Science & Technology Organisation
Salisbury Adelaide, South Australia

Craig Markwardt wrote:

```
>  andrew cool <andrew.cool@dsto.defence.gov.au> writes:
>
>>  Martin Downing wrote:
>>>  For a simple case like this, why not just use:
>>>      Min_array = data_array[*,*,0] < data_array[*,*,1] < data_array[*,*,2]
>>>
```

```
>>>  Martin
>>
>>  G'day Martin,
>>
>>        Now that I'm back at work, I regret to advise that your approach
>>        doesn't work. Craig's, however, does :-
>>
>>>     data_array = Fltarr(640,500,NZ)
>>>     Min_array = data_array(*,*,0)
>>>
>>>     for i = 1, NZ-1 do $
>>>        min_array = min_array < data_array(*,*,i)
>>
>>
>>        It seems that you need to have an initial test condition before you
>>        start applying those < operators. Not being a math-head, that might
>>        not be the right jargon to describe it.
>
>  I would have thought both approaches would have worked, and been about
>  the same speed.  That is curious.  The advantage to my approach is
>  that NZ, the size of the third dimension, can be variable; and the
>  advantage of Martin's is that it all fits one one line (but the number
>  is hardcoded).
>
>  Craig
>
>  --
>   --------------------------------------------------------- --------------
>  Craig B. Markwardt, Ph.D.         EMAIL:    craigmnet@cow.physics.wisc.edu
>  Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
>   --------------------------------------------------------- --------------
```