
Subject: Re: Cumulative max() in *arbitrary* dimension?

Posted by [JDS](#) on Thu, 08 Mar 2012 18:33:39 GMT

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

On Tuesday, February 28, 2012 1:57:20 PM UTC-5, Gianguido Cianci wrote:

> On Monday, February 27, 2012 6:09:38 PM UTC-6, JDS wrote:

>> In fact, I was very surprised to find

>> that, when working along the final dimension of large arrays (of a few

>> hundred million elements), MAX_CUMULATIVE is ~2x faster than its

>> MAX(DIMENSION=) analog, which produces a subset of the information!

>

> Mind. Blown.

I've since tuned this up a bit more, saving 1/2 of the index computation during each step of the loop by incrementing a running index array. It's now (rather remarkably) >5x faster than MAX(DIMENSION=3) for me with large 3D arrays. And of course it gives all the intermediate cumulative max values.

You can easily show how inefficient MAX is on the final dimension with a simple example:

```
IDL> a=byte(randomu(sd,300,400,3000)*256)
```

```
IDL> t=systime(1) & b=max(a,DIMENSION=3) & print,systime(1)-t
```

```
IDL> t=systime(1) & b2=a[*,* ,0] & for i=1,3000-1 do b2>=a[*,* ,i] & print,systime(1)-t
```

```
IDL> print,array_equal(b,b2)
```

I can only presume the built-in MAX has some design limitations for final dimension looping. I presume this all works the same way for MIN, BTW.
