
Subject: Re: Sum along diagonals

Posted by [Craig Markwardt](#) on Fri, 25 Aug 2000 07:00:00 GMT

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

mole6e23@hotmail.com (Todd Clements) writes:

...

> Every once in a while (not often enough to make me worry about optimizing
> too much), I want to take a not necessarily square matrix and get the sum
> along the diagonals, such as the following, with the theoretical function
> sum_diag:

>

> IDL> blah = indgen(4, 4)

> IDL> print, blah

> 0 1 2 3

> 4 5 6 7

> 8 9 10 11

> 12 13 14 15

> IDL> print, sum_diag(blah)

> 0 5 15 30 30 25 15

>

> which is the series [0, 4+1, 8+5+2, 12+9+6+3, ...]

>

> Of course, to be difficult, I'd like it to work for non-square matrices as well:

...

How about this solution. It's not a one-liner, and it uses two loops, but remember loops are not always bad if you can do a lot of work inside one iteration. This one makes $NX+NY-1$ iterations.

```
;; Set up the problem with some fake data, a NX x NY array
```

```
nx = 4 & ny = 3 & mm = indgen(nx, ny)
```

```
;; Output array
```

```
tt = fltarr(nx+ny-1)
```

```
;; Do the work
```

```
ll = lindgen(nx>ny)
```

```
for i = 0, ny-1 do tt(i) = total((mm(0+ll,i-ll))(0:i<(nx-1)))
```

```
for i = 1, nx-1 do tt(i+ny-1) = total((mm(i+ll,ny-1-ll))(0:(nx-1-i)<(ny-1)))
```

For the gobledy-gook impaired, first note that the two loops march down the left side of the array and then across the bottom, respectively. Second, I'm using array indexing along two dimensions simultaneously. The expression, `mm(0+ll,i-ll)`, is the actual diagonal of interest. The bit at the end of the expression, `(0:i<(nx-1))`, is used to trim off the end, which may contain the wrong data if the array is not square.

There you go! This is a speedy devil on my machine.

Good luck,
Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
