
Subject: Re: FOR loops removal

Posted by [ben.bighair](#) on Fri, 22 Aug 2008 10:50:36 GMT

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

On Aug 22, 5:08 am, loebasboy <stijn....@gmail.com> wrote:

> On Aug 21, 8:34 pm, Chris <beaum...@ifa.hawaii.edu> wrote:

>

>

>

>> What about something like this?

>

>> s= 2*(array le 2)-1

>> array = (array ne 0)*array + (array eq 0)*(shift(s,1)+shift(s,-1))

>

>> You'll have to manually fix the endpoints, but the rest should be

>> good. I tested this and the first loop on an array of 400,000

>> elements. The first loop ran in .54 seconds, while the second ran in .

>> 054s.

>

>> Also, you should be a little careful when testing whether array equals

>> zero. If the array isn't an integer type (int, byte, long, etc), then

>> its possible to have numbers you think ought to be zero but, because

>> of finite precision arithmetic, have very small nonzero values. If you

>> suspect that this is happening, you can try a test like abs(array) le

>> eps, where eps is something like 10^-5 or something big enough to

>> cover roundoff error but small enough not to treat nonzero data you

>> care about as zero

>

>> chris

>

> Hello, thank you both for the input. Chris, yours doesn't go faster

> also despite it is really elegant (and pointing me to the shift

> function, which I didnt know exist and can come in handy). The reason

> for the lack of speed profit is that I haven't a large array there

> that needs to be processed, instead I have many small array that need

> to be processed (the whole of my program exist of one BIG for loop)

> and there are relatively a lot of zeros in the processed array (which

> is why the WHERE function solution doesn't work either)

>

Hi,

Given your response to Chris I think the teeny suggestion I have to share is a moot point. But I'll put it out there because it can be handy. If I understand Chris' code snippet correctly, there can be a speeded up by using something like this...

```
mask1 = array ne 0
```

```
mask0 = ~mask1
array = mask1*array + mask0*(shift(s,1)+shift(s,-1))
```

It probably is a slim gain but can add up to a lot if you are looping. In my simple minded test below I see better than 2x gain usually.

```
a = RANDOMN(seed, 2000, 2000) > 0.0
b = A GT 0
```

```
PRINT, "~B test"
t0 = SYSTIME(/sec)
c = ~B
print, systime(/sec) - t0
```

```
PRINT, "B EQ 0 test"
t0 = SYSTIME(/sec)
c = b EQ 0
print, systime(/sec) - t0
```

```
IDL> @not_test
~B test
    0.056261063
B EQ 0 test
    0.13446379
```

Cheers,
ben
