
Subject: Re: array indexing question

Posted by [mperrin+news](#) on Fri, 12 Jul 2002 01:20:45 GMT

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

Michael A. Miller <mmiller3@iupui.edu> wrote:

> I have three arrays, pa, pb and pab. pa and pb are both 1D
> arrays of length N and pab is a 2D NxN array. I want to
> calculate the sum of $\text{pab}[i,j] * \log(\text{pab}[i,j]) / \text{pa}[i] / \text{pb}[j]$. I know
> that I can do things like $\text{total}(\text{pa} * \log(\text{pa}))$ when I'm dealing
> with a single array. Any suggestions for how to do my first sum
> most efficiently?

How about just

```
total( pab * log(pab) / rebin(pa,N,N) / rebin(transpose(pb),N,N))
```

The `reform()` calls turn your 1D arrays into 2D arrays, so they can then easily be combined with the pab array. The transpose call is needed to get the latter vector changed into a column rather than row vector so the indices work properly.

See the ever-popular

http://www.dfanning.com/tips/rebin_magic.html

for more tricks along these lines.

- Marshall

Subject: Re: array indexing question

Posted by [mmiller3](#) on Fri, 12 Jul 2002 13:35:35 GMT

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

>>>> > "Marshall" == Marshall Perrin <mperrin> writes:

> How about just

> total(pab * log(pab) / rebin(pa,N,N) /
> rebin(transpose(pb),N,N))

That's it!

> See the ever-popular
> http://www.dfanning.com/tips/rebin_magic.html for more
> tricks along these lines.

It has now become even more popular! Thanks :)
