
Subject: Re: Rebin/Reform/Histogram
Posted by [Mrunmayee](#) on Tue, 21 Sep 2010 13:45:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

One wrong statement in the loop:

```
> for i = 0, n1-1 do begin
>>>   w = rebin(w1[*],i], nv, n2) ; Makes nvxn2 matrix
>   p = w*w2                      ; Performs number 3 above.
>   SF[i,*] = dt # p
> endfor
```

Subject: Re: Rebin/Reform/Histogram
Posted by [rogass](#) on Tue, 21 Sep 2010 20:41:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 21 Sep., 15:45, Mrunmayee <gaur...@gmail.com> wrote:

```
> One wrong statement in the loop:
>
>
>
>> for i = 0, n1-1 do begin
>>>>   w = rebin(w1[*],i], nv, n2) ; Makes nvxn2 matrix
>>   p = w*w2                      ; Performs number 3 above.
>>   SF[i,*] = dt # p
>> endfor
```

Hi, maybe you have to do this in 3D. It can look something like the following (no proof-just an idea):

```
sf = rebin(/sample,reform(/over,transpose(dt),1,n2,n1),nv,n1,n2) # $
(rebin(/sample,reform(w1, nv, 1, n1,/over),nv, n2, n1) * $
transpose(rebin(/sample,reform(w2, nv, 1, n2,/over),nv, n1, n2),
[0,2,1]))
```

However, if you have large matrices then you will run into memory problems. Just reduce redundancy and maximise the work within the loop. Then you won't "feel" the loop overhead. The 'over' keyword transforms the matrices fast in place - so keep it in mind if you like to use them later.

Regards

CR

Subject: Re: Rebin/Reform/Histogram

Posted by [Mrunmayee](#) on Wed, 22 Sep 2010 09:21:59 GMT

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

On Sep 22, 1:41 am, chris <rog...@googlemail.com> wrote:

> On 21 Sep., 15:45, Mrunmayee <gaur...@gmail.com> wrote:

>

> sf = rebin(/sample,reform(/over,transpose(dt),1,n2,n1),nv,n1,n2) # \$

> (rebin(/sample,reform(w1, nv, 1, n1,/over),nv, n2, n1) * \$

> transpose(rebin(/sample,reform(w2, nv, 1, n2,/over),nv, n1, n2),

> [0,2,1]))

>

> However, if you have large matrices then you will run into memory

> problems. Just reduce redundancy and maximise the work within the

> loop. Then you won't "feel" the loop overhead. The 'over' keyword

> transforms the matrices fast in place - so keep it in mind if you like

> to use them later.

Thanks, Chris, I tried this. And I did run into memory problem. I see

what you did above and I was looking for just something like it. I

just needed to tweak nv, n1, n2 in rebin commands to match dimensions

(it doesn't match as you have written). So for nv = 231, n1 = 1547 and

n2 = 1537, I cannot rebin w1 (nv x n1 size) without running into "%

Array has too many elements."

I just read the tip from David about "Memory used to subscript

arrays". I am thinking of keeping the loop, but instead of using "*"

for the subscript, I can just use a pre-created index array. Just to

save some memory. But I probably won't be able to rebin to the desired

dimensions anyway.

If there is a way around, do let me know.
