Subject: Re: How to rebin complex array?
Posted by Mark Hadfield on Wed, 14 Apr 2004 00:10:39 GMT
View Forum Message <> Reply to Message

Yunxiang Zhang wrote:
>
> I wanna do the following thing as fast as possible.
>
> ;creat a complex array
> a=complexarr(na)
> ;replicate the array k times
> b=rebin(a,na,k)
>
> But rebin won't accept complex array. :( What should I do?

Extract the real & imaginary parts, rebin each one & recombine them?

> Another similiar question is what is the best solution to replicate a 2d
> image eg dist(512,512) k times to get a k-frame "still movie" as a 3d array
> movie(512,512,k) without using any loops?

You can use REBIN in this case.

You might want to look at JD's dimension juggling tutorial, on David Fanning's site:

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

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: How to rebin complex array?
Posted by Peter Mason on Wed, 14 Apr 2004 00:47:46 GMT
View Forum Message <> Reply to Message

Yunxiang Zhang wrote:
> Folks,
>
> I wanna do the following thing as fast as possible.
>
> ;creat a complex array
> a=complexarr(na)
> ;replicate the array k times
> b=rebin(a,na,k)

>
> But rebin won't accept complex array. :( What should I do?

Further to what Mark has written, if this *really* is what you want to do
and if you can live with REBIN's /SAMPLE option (which is faster than its
default bilinear interpolation anyway), then you might try something like
the following:

```
d=double( a, 0, na )
r=rebin( temporary(d), na, k, /SAMPLE )
b=complex( temporary(r), 0, na, k )
```

Of course, this'll only work for single-precision complex "a". (8 bytes
per value, same as "non-complex" double precision.)

Peter Mason

---

Subject: Re: How to rebin complex array?
Posted by Timm Weitkamp on Wed, 14 Apr 2004 07:39:21 GMT
View Forum Message <> Reply to Message

On 13.04.04 at 15:57 -0700, Yunxiang Zhang wrote:

> I wanna do the following thing as fast as possible.
>
> ;creat a complex array
> a=complexarr(na)
> ;replicate the array k times
> b=rebin(a,na,k)
>
> But rebin won't accept complex array. :( What should I do?

This here is probably what you want:

```
a=complexarr(na)   ;  create a complex array
u = 1 + FLTARR(k)  ;  create unit vector with k elements
b = a # u          ;  matrix multiplication of a and u
```

By the way, there is a good reason for REBIN not to work on complex
arrays, and that is the fact that there is not really any proper way of
interpolating between two complex values. (You could argue that
next-neighbor sampling should still be possible.)

> Another similiar question is what is the best solution to replicate a 2d
> image eg dist(512,512) k times to get a k-frame "still movie" as a 3d array
> movie(512,512,k) without using any loops?

As previous posters have said, there's no reason why REBIN shouldn't work

here, given that DIST yields a real-valued result. But *if* you did have a complex array that you wanted to replicate in the manner described above, or if you want to avoid REBIN for some other reason, and not use loops, then I'd probably do this:

```
s = SIZE(image)
movie = MAKE_ARRAY(s[1], s[2], TYPE=s[3])
movie[*] = image[*] # (1.0 + FLTARR(k))
```

Cheers,

Timm

--
Timm Weitkamp <http://people.web.psi.ch/weitkamp>

---

Subject: Re: How to rebin complex array?
Posted by James Kuyper on Wed, 14 Apr 2004 13:47:31 GMT
View Forum Message <> Reply to Message

Timm Weitkamp wrote:
...
> u = 1 + FLTARR(k) ; create unit vector with k elements

A unit vector is a vector such that total(u*u) eq 1.0" (or for complex numbers, total(u*conj(u)) eq 1.0). What you've described is a vector whose elements are all 1, which has no special name.

...
> By the way, there is a good reason for REBIN not to work on complex
> arrays, and that is the fact that there is not really any proper way of
> interpolating between two complex values.

Why in the world not? All of the normal interpolation formulas work perfectly well when applied seperately to the real and imaginary components of a table of complex values. The formulas just involve multiplcation, division, addition, and subtraction, all of which are perfectly well defined for complex numbers, and they produce reasonable results (well, not quite - in some higher-order interpolation algorithms it's necessary to replace c^2 with c*conj(c) in some places).

---

Subject: Re: How to rebin complex array?
Posted by R.G. Stockwell on Wed, 14 Apr 2004 16:31:28 GMT
View Forum Message <> Reply to Message

"James Kuyper" <kuyper@saicmodis.com> wrote in message
news:407D40F3.F045EC70@saicmodis.com...
> Timm Weitkamp wrote:
> ...
...
>> By the way, there is a good reason for REBIN not to work on complex
>> arrays, and that is the fact that there is not really any proper way of
>> interpolating between two complex values.
>
> Why in the world not?

In the 2D complex space, it is ambiguous as how to interpolate.

Breifly, one could interpolate real and imaginary seperately, as
suggested in this thread. Or one could average the lengths of
the vector, and the angle of the vector.

For any rotating vector time series, component-wise interpolation
will almost always underestimate the amplitude of the interpolated value.

Consider two subsequent values, [1,0] and [0,1].
If this is a unit vector rotating, then the "correct" interpolated
value would be [1/root 2,1/root 2] = [.707,.707]. The componently
interpolated value is [.5,.5].

So it really depends on what the type of time series one has.
It may be that a "norm preserving interpolation" may be required.
(average the lengths of the vector rather than just average components).

Cheers,
bob

---

Subject: Re: How to rebin complex array?
Posted by Yunxiang Zhang on Wed, 14 Apr 2004 20:01:06 GMT
View Forum Message <> Reply to Message

Folks,

I did some test to rebin a complex array. Thanks for all you guys' help!:)

speed test result: a for loop is not as worse as I would imagine in
this case.

"#" > "for loop" > "special rebin" > "component rebin"

for loop 0.25066495

component rebin      0.53236008
special rebin      0.37491202
#      0.16867399

I also found using fltarr is better than using replicate
fltarr > replicate

replicate     0.039183140
fltarr     0.027969122

```
;here's my testing code
aa=bytscl(reform(ran2_normal(-738345,dim=5000),2,2500))
a=reform(complex(aa[0,*],aa[1,*]))

na=n_elements(a) & k=na

;using for loop
t0=systime(1)
b1=complexarr(na,k)
for i=0,k-1 do begin
b1[*,i]=a
endfor
print, 'for loop', systime(1)-t0

;using component rebin
t0=systime(1)
ra=real_part(a)
ia=imaginary(a)
rb=rebin(ra,[na,k])
ib=rebin(ia,[na,k])
b2=complex(rb,ib)
print,'component rebin', systime(1)-t0

;i call it special rebin, please refer to Peter's post
t0=systime(1)
d=double(a,0,k)
r=rebin(temporary(d),na,k,/sample)
b3=complex(temporary(r),0,na,k)
print,'special rebin', systime(1)-t0

;using #
t0=systime(1)
b4=a#(1+fltarr(k))
print,'#', systime(1)-t0


t0=systime(1)
for i=0,1000 do aaa=replicate(1.,10000)
```

```
print,'replicate', systime(1)-t0

t0=systime(1)
for i=0,1000 do bbb=1+fltarr(10000)
print,'fltarr', systime(1)-t0

end
```