Subject: Imposing inverse fft to be real
Posted by collinritzinger on Mon, 21 Jan 2013 23:41:19 GMT
View Forum Message <> Reply to Message

The issue: I am creating complex arrays with the same amplitude but randomizing the phases, then using inverse FFT.

This is easy, however I would like to impose that they are conjugate symmetric, i.e. Hermitian. The original array ("dn" and its fft "org" as defined below) are real valued and hence will return mostly (i.e. withing floating point precision) real valued FFT results. However, the randomized phase array ("rr", as defined below) should have the conjugate symmetry imposed on it in order to insure that the resulting inverse FFT will have negligible imaginary components. This seems like it would be straight forward....

```
dn=readfits('array.fits.gz')

org=fft(dn,/double,/center)  ; take the fft, center frequencies around origin

imbx=imaginary(org)
rebx=real_part(org)

phi=atan(imbx,rebx,/phase)
phaserand=dblarr(512,512,512)
amp=abs(org)

i_index=phi(sort(randomu(seed,[512,512,512])))                ; sorted random phases....
phaserand[*,*,*]=i_index          ; sorted randomized phases
rr=complex(amp*cos(phaserand),amp*sin(phaserand),/double)

for i=0,255 do begin     ;impose conjugate symmetry for real result
for j=0,255 do begin
for k=0,255 do begin

rr[511-i,511-j,511-k]=conj(rr[i,j,k])

endfor
endfor
endfor

inverse_rr=fft(rr,/inverse,/double,/center)
```

I've tried this using the center option, shifting the arrays...etc etc. But no matter what I try, the inverse of rr has non-negligible imaginary values :(

Any ideas on this? Thanks! :)