Subject: Inverse FFT
Posted by aultc on Mon, 16 Dec 2002 16:33:39 GMT
View Forum Message <> Reply to Message

Hi,

I hope someone help me with a problem I am having with the FFT function.

I have a signal f_t, which I then take the FFT of to produce its corresponding spectral components. I then want to manually compute its inverse FT, rather than using the IDL FFT(.../inverse) function.

The reason for this is that I want each spectral component to propogate at different velocitys over a time period t. Hence, when the signal is recombined t seconds later, the signal *should* look different.

I am not having much luck at the moment, so any suggestions on this problem will be gratefully received.

Thanks,
Colin

Subject: Re: Inverse FFT
Posted by R.G. Stockwell on Mon, 16 Dec 2002 17:53:33 GMT
View Forum Message <> Reply to Message

Colin Ault wrote:

> Hi,
>
> I hope someone help me with a problem I am having with the FFT
> function.
>
> I have a signal f_t, which I then take the FFT of to produce its
> corresponding spectral components. I then want to manually compute its
> inverse FT, rather than using the IDL FFT(.../inverse) function.
>
> The reason for this is that I want each spectral component to
> propogate at different velocitys over a time period t. Hence, when the
> signal is recombined t seconds later, the signal *should* look
> different.
>
> I am not having much luck at the moment, so any suggestions on this
> problem will be gratefully received.
>

> Thanks,
> Colin

To "manually compute" the inverse fft in the same way IDL does,
make sure you have use the + argument in your exp(), and calculate
the straight sum (i.e. do not divide by N).
Having said that, there is absolutely no difference in fft(/inv) and
your manual method, so I suggest using the fft method. The only difference
is in the algorithm for calculating the result.

To "propogate at different velocitys over a time period t", perhaps you
can implement that by adjusting the phase of your fft componets (using
the fft shift theorem) to get your desired results.

Cheers,
bob

Subject: Re: Inverse FFT
Posted by Streun Andreas on Mon, 16 Dec 2002 18:01:04 GMT
View Forum Message <> Reply to Message

Colin Ault wrote:
> I have a signal f_t, which I then take the FFT of to produce its
> corresponding spectral components. I then want to manually compute its
> inverse FT, rather than using the IDL FFT(.../inverse) function.
>
> The reason for this is that I want each spectral component to
> propogate at different velocitys over a time period t. Hence, when the
> signal is recombined t seconds later, the signal *should* look
> different.
>

Hi Colin,

I'm not sure if I understood the problem, but I don't see the reason
why you can't use the standard FFT: You transform to frequency
domain, apply your function to the spectrum, for example
a filter, and transform back to time domain. Of course, you have
to work on the complex frequencies, not on the absolute values
(i.e. power spectrum) in order not to loose phase informations
for backtransformation.
Attached find a code fragment, where I wanted to see how a
time signal looks, if I consider only significant frequencies
(i.e.with absolute values above some threshold). Maybe that's
related to your problem.

Best regards,
Andreas

```
; get profile of image
    p =total(roi,1)

; and show it
    plot, p

; make fft
    f =fft(p,1)

; get absolute value of [complex] fft
    fa=abs(f)

; define filter threshold relative to maximum
    filter=0.1

; filter fft by deleting all frequencies lower than threshold
    ff=f
    ff[where(fa lt filter*max(fa))]=0.0

; inverse fft to make filtered profile
    pf=fft(ff,-1)

; show filtered profile
    oplot, pf, color=250

; show absolute fft, but only the first channels
    plot, fa[0:100]

; show filtered fft too
    ffa=abs(ff)
    oplot, ffa[0:100], color=250
```

Subject: Re: Inverse FFT
Posted by jeyadev on Mon, 16 Dec 2002 23:56:24 GMT
View Forum Message <> Reply to Message

In article <3DFE131D.6000005@noemail.now>,
R.G. Stockwell <sorry@noemail.now> wrote:
> Colin Ault wrote:
>
>> Hi,
>>
>> I hope someone help me with a problem I am having with the FFT

>> function.
>>
>> I have a signal f_t, which I then take the FFT of to produce its
>> corresponding spectral components. I then want to manually compute its
>> inverse FT, rather than using the IDL FFT(.../inverse) function.
>>
>> The reason for this is that I want each spectral component to
>> propogate at different velocitys over a time period t. Hence, when the
>> signal is recombined t seconds later, the signal *should* look
>> different.

This is quite all right. Consider the problem of the continuity in one
dimension

dq/dt + K dE/dx = 0

where q(x,t) and E(x,t) are functions of both variables of 'x' and 't'
and the derivatives are partials. Also the driving field E(x,t) is a
linear function of q(x,t) in the sense that

E(x,t) = Integral[q(x,t)P(x), x]

The solution is obtained exactly as you say. Taking the Fourier
Transform of the entire equation respect to 'x' separates the problem in
'k' space and each 'k' component q(k,t) propagages with a time constant
that depends on 'k', i.e. you get something like

dq(k,t)/dt + q(k,t)/a(k) = 0

where a(k) is the 'k' dependent time constant.

To solve this, you proceed exactly as you did. Just make sure that
you put the right flag for the reverse transform.

>> I am not having much luck at the moment, so any suggestions on this
>> problem will be gratefully received.

One thing that you should check is whether you are packing the
arrays for the transform in the correct fashion. One of the most
common mistakes is the treatment of the negative frequencies. (I
myself have been guilty of messing them up!) As you are in IDL,
it is easy to check and see if they are as expected.

Good luck.
--

Surendar Jeyadev jeyadev@wrc.xerox.bounceback.com

Subject: Re: Inverse FFT
Posted by aultc on Tue, 17 Dec 2002 11:06:18 GMT
View Forum Message <> Reply to Message

Hi,

Thanks for the comments and suggestions.

I am packing the negative frequencies correctly (I hope!) - just using the same method in the online FFT example. I call this array k_points, and this ranges thus 0, 0.01, 0.02.....,0.50, -0.49, ..., -0.02, -0.01. This is for 100 points sampled at T=1.0 seconds.

I then compute the FFT via the normal method, FFT(function), and obtain my expected spectral pattern. So far, so good!

I then use the following code to compute (manually) the inverse:

```
FOR j=0, n-1 DO BEGIN

 spec_sig = FT*exp(2*pi*k_points * t[j]/n)

 new_signal[j] = TOTAL(spec_sig)

ENDFOR
```

FT is an array holding the fourier transform of my function
n is the number of points (100)
k_points is the same as mentioned above
t is an array from 0,1,..100, i.e the times at which the function is sampled.

I then just carry out the summation and put it in new_signal. This is then plotted.

Unfortunately this still doesn't work! Any further suggestions would be greatly welcomed.

Colin

then just carries out the summation

aultc@astro.warwick.ac.uk (Colin Ault) wrote in message
news:<24be9e8e.0212160833.7d214a6a@posting.google.com>...
> Hi,
>
> I hope someone help me with a problem I am having with the FFT
> function.
>
> I have a signal f_t, which I then take the FFT of to produce its
> corresponding spectral components. I then want to manually compute its
> inverse FT, rather than using the IDL FFT(.../inverse) function.
>
> The reason for this is that I want each spectral component to
> propogate at different velocitys over a time period t. Hence, when the
> signal is recombined t seconds later, the signal *should* look
> different.
>
> I am not having much luck at the moment, so any suggestions on this
> problem will be gratefully received.
>
> Thanks,
> Colin

Subject: Re: Inverse FFT
Posted by aultc on Tue, 17 Dec 2002 11:10:06 GMT
View Forum Message <> Reply to Message

I should also add that the piece of code in my follow-up message is
strictly just to recreate the original signal. Once I can do that, I
can then add in the velocity / propagation terms.

aultc@astro.warwick.ac.uk (Colin Ault) wrote in message
news:<24be9e8e.0212160833.7d214a6a@posting.google.com>...
> Hi,
>
> I hope someone help me with a problem I am having with the FFT
> function.
>
> I have a signal f_t, which I then take the FFT of to produce its
> corresponding spectral components. I then want to manually compute its
> inverse FT, rather than using the IDL FFT(.../inverse) function.
>

> The reason for this is that I want each spectral component to
> propogate at different velocitys over a time period t. Hence, when the
> signal is recombined t seconds later, the signal *should* look
> different.
>
> I am not having much luck at the moment, so any suggestions on this
> problem will be gratefully received.
>
> Thanks,
> Colin

Subject: Re: Inverse FFT
Posted by R.G. Stockwell on Tue, 17 Dec 2002 14:42:27 GMT
View Forum Message <> Reply to Message

Colin Ault wrote:

> Hi,
>
> Thanks for the comments and suggestions.
>
> I am packing the negative frequencies correctly (I hope!) - just using
> the same method in the online FFT example. I call this array k_points,
> and this ranges thus 0, 0.01, 0.02.....,0.50, -0.49, ..., -0.02,
> -0.01. This is for 100 points sampled at T=1.0 seconds.
>
> I then compute the FFT via the normal method, FFT(function), and
> obtain my expected spectral pattern. So far, so good!
>
> I then use the following code to compute (manually) the inverse:
>
> FOR j=0, n-1 DO BEGIN
>
> spec_sig = FT*exp(2*pi*k_points * t[j]/n)
>
> new_signal[j] = TOTAL(spec_sig)
>
> ENDFOR
>
> FT is an array holding the fourier transform of my function

Perhaps it was a typo, but don't you want

spec_sig = FT*exp(complex(0,1)*2*!pi*k_points * t[j]/n)

(And make sure FT is complex)

Cheers,
bob stockwell

Subject: Re: Inverse FFT
Posted by R.G. Stockwell on Tue, 17 Dec 2002 15:02:57 GMT
View Forum Message <> Reply to Message

Here is an (main level) example that hopefully does what you want.
Note the slightly better precision of the fft method, due to the superior
method of calculating the same thing. Also, there is a huge difference
in speed, especially as N gets larger than say 10 or so.
On my computer 1.6Ghz athlon, the times are about a factor of 20.
(i.e. fft 20 times faster than "by hand").
The point I am getting to is "don't inverse fft by hand".

And note that I left the resulting
arrays as complex, but it is equal to the original time series, since the
imaginary part is zero. You may want to cast them to float (or double)

Cheers,
bob stockwell

```
; make a time series
len = 16
a = randomn(seed,len)
a = double(a)

; calc spectrum
ft = fft(a)

; inverse by fft FAST!
tic = systime(1)
ift = fft(ft,/inverse)
toc = systime(1)
print,'fft time = ', (toc - tic)*1000d ; microseconds

; inverse by hand SLOW!
tic = systime(1)
byhand = dcomplexarr(len)
t = dindgen(len)
for i = 0,len-1 do begin
```

```
 byhand = byhand + ft[i]*exp(complex(0,1)*2*!dpi*t*i/len)
endfor
toc = systime(1)
print,'by hand time = ', (toc - tic)*1000d

; print out results
print,'original time series'
print,a
print,'inverse by fft'
print,ift
print,'inverse by hand'
print,byhand

; plot out results
!p.multi = [ 0,1,3]
plot,a,tit='timeseries'
plot,ift,tit='by fft'
oplot,imaginary(ift),linestyle=2
plot,byhand,tit='byhand'
oplot,imaginary(byhand),linestyle=2
end
```

Subject: Re: Inverse FFT
Posted by aultc on Wed, 18 Dec 2002 09:19:51 GMT
View Forum Message <> Reply to Message

Hi Bob,

Fantastic! Thank you very much for your comments and examples - the manual calculation of the IFT is now working as it should. Also, thanks for your comments on the time comparison, it is definitely more desirable to use the FFT(/inv) method, rather than "by hand". However, I think in some of the things I want to do I will need the manual method.

Thanks once again for your help,

Colin

"R.G. Stockwell" <sorry@noemail.now> wrote in message news:<3DFF3CA1.7010206@noemail.now>...
> Here is an (main level) example that hopefully does what you want.

> Note the slightly better precision of the fft method, due to the superior
> method of calculating the same thing. Also, there is a huge difference
> in speed, especially as N gets larger than say 10 or so.
> On my computer 1.6Ghz athlon, the times are about a factor of 20.
> (i.e. fft 20 times faster than "by hand").
> The point I am getting to is "don't inverse fft by hand".
>
>
> And note that I left the resulting
> arrays as complex, but it is equal to the original time series, since the
> imaginary part is zero. You may want to cast them to float (or double)
>
> Cheers,
> bob stockwell
>
>
>
> ; make a time series
> len = 16
> a = randomn(seed,len)
> a = double(a)
>
> ; calc spectrum
> ft = fft(a)
>
> ; inverse by fft FAST!
> tic = systime(1)
> ift = fft(ft,/inverse)
> toc = systime(1)
> print,'fft time = ', (toc - tic)*1000d ; microseconds
>
> ; inverse by hand SLOW!
> tic = systime(1)
> byhand = dcomplexarr(len)
> t = dindgen(len)
> for i = 0,len-1 do begin
> byhand = byhand + ft[i]*exp(complex(0,1)*2*!dpi*t*i/len)
> endfor
> toc = systime(1)
> print,'by hand time = ', (toc - tic)*1000d
>
> ; print out results
> print,'original time series'
> print,a
> print,'inverse by fft'
> print,ift
> print,'inverse by hand'
> print,byhand

```
>
>  ; plot out results
>  !p.multi = [ 0,1,3]
>  plot,a,tit='timeseries'
>  plot,ift,tit='by fft'
>  oplot,imaginary(ift),linestyle=2
>  plot,byhand,tit='byhand'
>  oplot,imaginary(byhand),linestyle=2
>  end
```