Comparison Between Analytical FT and FFT of smiley Face

1. Comparison of magnitude between analytical FT and FFT

Since Jianhao has finished the smiley face and the analytical Fourier transformation of this sum of a set of Gaussian distributions. Here I give the comparison between the analytical result and the FFT result. By using FFT, the same thing is done as we used before.

(1) Shifted in frequency domain by using fftshift()

(2) Modified in magnitude by dividing it by T^2/N^2,where T is the time domain for sampling like 10(-5~5), N is sampling number like 128 here.

(3) Modified in phase for each FFT value by utilizing the displacement property as the code shows below.

(4) Take the absolute value.

```
G_fft_comp=fftshift(fft2(f))*T^2/N^2;
for i=1:N
    for j=1:N
        G_fft_comp_m(i,j)=G_fft_comp(i,j)*exp(-1i*pi*(i-1)-1i*pi*(j-1));
    end
end
G_fft_abs=abs(G_fft_comp_m);
```

Fig1 is the magnitude distribution in frequency domain of the analytical FT. Fig2 is the magnitude of FFT result. Then I calculate the error between these two methods.
When N=128, the error is up to 3.0611. Compared to the magnitude whose maximum value is 289.0684, the relative error is about 1.06%.

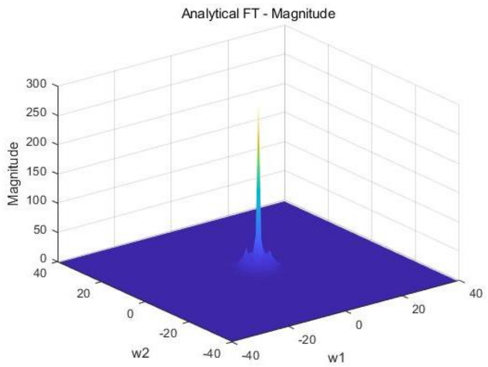

Fig1. Analytical FT magnitude

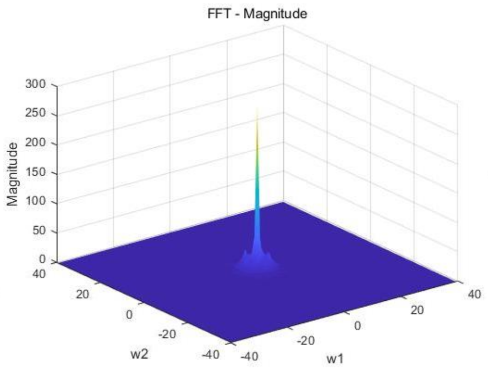

Fig2. FFT magnitude

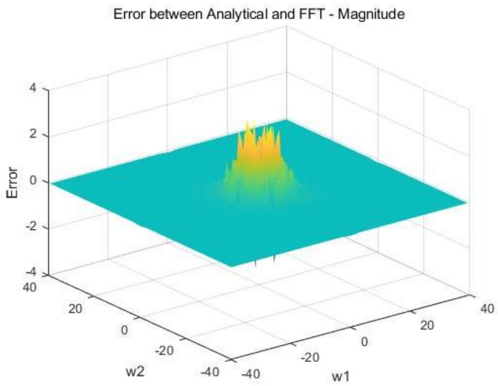

Fig3. Error of magnitude

The error decreases as the sampling number increases. When N=512, maximum error is 0.7045

2. Comparison of phase between analytical FT and FFT

We set some extremely small values of the real part and imaginary part of the FFT results to zero here. Compare the difference of phase of the two methods and we find some points jump out. The same phenomenon appears in 1-D situation before and the error value is approximately 2π or -2π. I think these points should be zero. It is caused by the tolerance to decide whether the extremely small values should be set to zero. It will be explained detailed in next section.

Here I think it does not matter that some points jump out.

Below is the modified part.

```
G_fft_real=real(G_fft_comp_m);
G_fft_imag=imag(G_fft_comp_m);
% index1=find(abs(G_fft_real)<1e-2);
% index2=find(abs(G_fft_imag)<1e-2);
index=find(abs(G_fft_real)<1e-2&abs(G_fft_imag)<1e-2);
G_fft_real(index)=0;
G_fft_imag(index)=0;
G_fft_comp_m2=G_fft_real+1i*G_fft_imag;
```

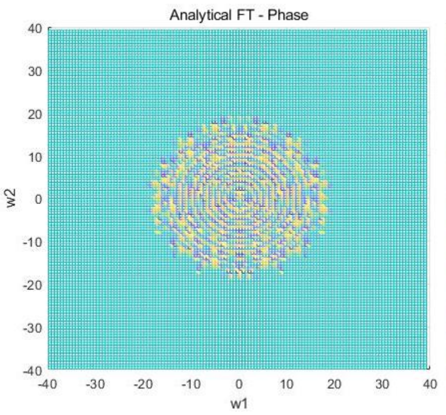

Fig4. Analytical FT phase

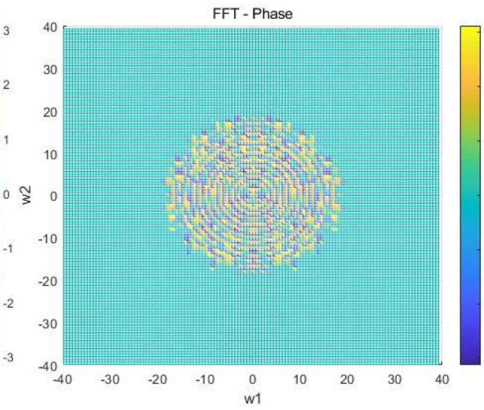

Fig5. FFT phase

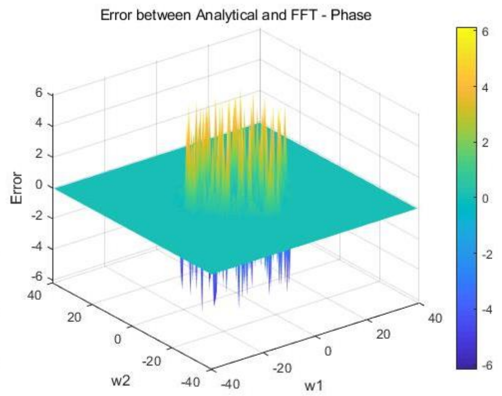

Fig6. Error of phase

3. Comparison of magnitude between analytical Gaussian and IFFT

Although the errors in magnitude and phase when calculating separately, the first 4 modifying steps above are correct theoretically. But if we want to extract phase alone and see its distribution, the machine error in MATLAB is not negligible. These extremely small value of the real part and imaginary part of the FFT result is in the same order of magnitude, so problems arise when using atan2() directly. Therefore, we need to modify these extremely small values to zero. But the tolerance of this small value should depend on different μ, Σ and N.

But the displacement term we product here do take effect. The phase is surely modified to be no problem if we don't see it alone. What's more, the phase difference here is π for each point, so there is only a difference in its sign. The key point is still the magnitude of real and imaginary part.

Anyway, after the whole procedure (without modifying small value to zero), I take a look into the inverse procedure like

```
f_ifft=abs(ifftshift(ifft2(G_fft_comp_m)))*N^2/T^2;
```

Then we get the error between the Gaussian distribution of smiley face and the inversed value, which is about $\mathbf{10^{-13}}$. We can totally dismiss them.

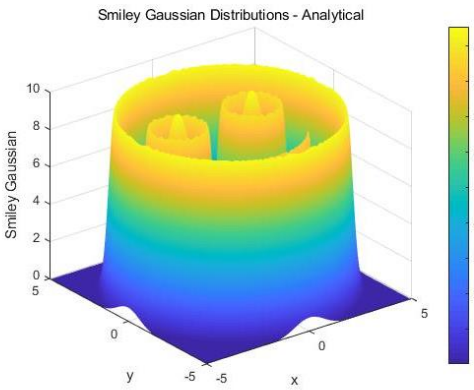

Fig7 Analytical Gaussian

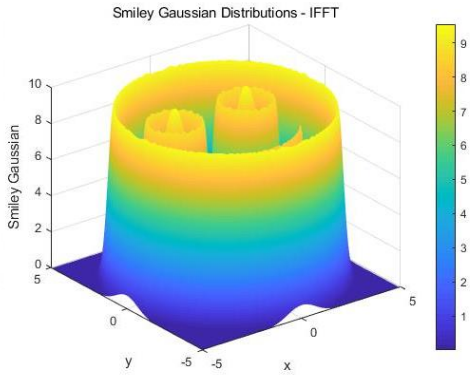

Fig8. IFFT Gaussian

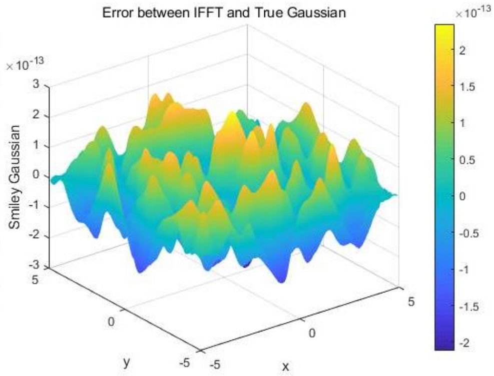

Fig9. Error of Gaussian

I also take a look in to what will happen if I use the modified result along with modify small value to zero. It is shown below. The error is a little larger but is still under control. It means those points contain very little valuable information.