MASTER MOSIG M1

Signal Processing

Signal Analysis in the Frequency Domain
Computer Exercises N. 3

Authors
Ky Nguyen
Raquel Oliveira

Grenoble, 24 November 2010

Index

11D Fourier Transform Implementation....................3
2Fourier Transform of an Impulse....................4
3Fourier Transform of a Rectangle....................7
42D Fourier Transform with ImageJ....................14
5Free the cat....................17

1 1D Fourier Transform Implementation

1.1) In the class Signal, complete the Discrete Fourier Transform Java code.

```
public ComplexSignal dft() {
    ComplexSignal result = new ComplexSignal();
    int nbSamples = this.getNbSamples();

    // Write your code here
    Complex c = new Complex();
    double xk;
    for (int n =0; n< nbSamples; n++)
    {
        Complex Xn = new Complex();
        for(int k = 0; k< nbSamples; k++)
        {
            xk  = this.getValueOfIndex(k);
            c = Complex.createFromPolar(1,-(2*Math.PI*k*n)/nbSamples);
            c.multiplyByReal(xk);
            Xn.add(c);
        }
        result.add(Xn);
    }
    return result;
  }
```

1.2) In the class Signal, complete the Inverse Discrete Fourier Transform Java code.

```
  public static Signal idft(ComplexSignal fourier) {
    ComplexSignal result = new ComplexSignal();
    int nbSamples = fourier.getNbSamples();

    // Write your code here
    Complex c = new Complex();
    for (int n =0; n< nbSamples; n++)
    {
        Complex Xk;
        Complex xn = new Complex();
        for(int k = 0; k< nbSamples; k++)
        {
            Xk  = fourier.get(k);
            c = Complex.createFromPolar(1,(2*Math.PI*k*n)/nbSamples);
            c.mul(Xk);
            xn.add(c);
        }
```

```
      xn.multiplyByReal((double)1/(double)nbSamples);
      result.add(xn);
    }
    return result.getRealSignal();
  }
```

2 Fourier Transform of an Impulse

2.1) Calculate the Discrete Fourier Transform $Ƒ_D(\delta_a)[\omega] = \Delta_a[\omega]$ of δ_a for each ω. What happens if a = 0 ?

$Ƒ_D(\delta_a)[\omega] = Ƒ_D(\delta)[\omega-a] = Ƒ_D(S_a\,\delta)[\omega] =$

$$\int_{x=-\infty}^{x=+\infty} \delta_a(x)\, e^{-i\omega x}\, dx \;=\; \int_{x=-\infty}^{x=+\infty} \delta(x-a)\, e^{-i\omega x}\, dx$$

u=x-a, x=u+a

$$\int_{u=-\infty}^{u=+\infty} \delta(u)\, e^{-i\omega(u+a)}\, du \;=\; \int_{u=-\infty}^{u=+\infty} \delta(u)\, e^{-i\omega u}\, e^{-i\omega a}\, du \;=\; e^{-i\omega a}\int_{u=-\infty}^{u=+\infty} \delta(u)\, e^{-i\omega u}\, du \;=$$

$$e^{-i\omega a}\, Ƒ_D(\delta)[\omega] \;=\; e^{-i\omega a}\, 1 \;=\; e^{-i\omega a}$$

if a=0:

$Ƒ_D(\delta_a)[\omega] = e^{-i\omega 0} = 1 = Ƒ_D(\delta)[\omega]$

The Discrete Fourier Transform of the unit impulse function will be the same of the delta function.

2.2)Deduce from the previous expression: ReΔ, ImΔ, MagΔ, PhΔ. What happens if a=0?

$\Delta_a[\omega] = e^{-i\omega a}$ = c (complex number)

Using these formulas: $c = r\, e^{i\Theta}$, re = r (cos Θ) , im = r (sen Θ) we obtain:

MagΔ = r = 1

PhaseΔ = Θ = -ωa

ReΔ = cos (-ωa)

ImΔ = sin (-ωa)

if a=0:

MagΔ = r = 1

PhaseΔ = Θ = 0

ReΔ = cos (0) = 1

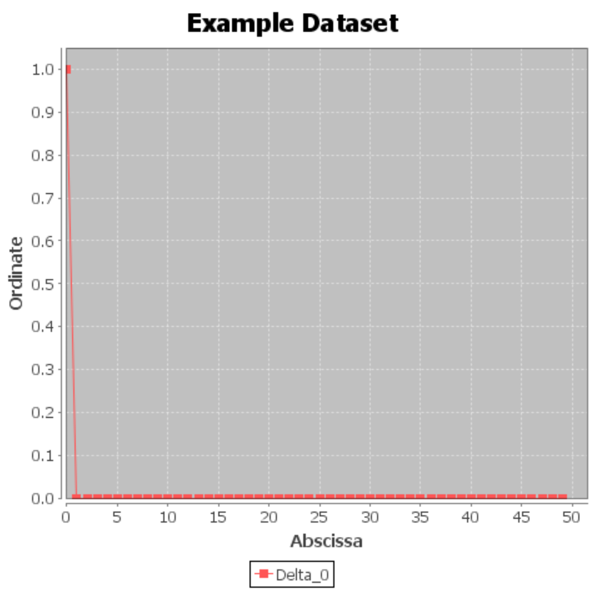
ImΔ = sin (0) = 0

2.3) Complete the function public static Signal generateDelta (int nonZeroSampleNumber, int numberOfSamples) of the class Signal.

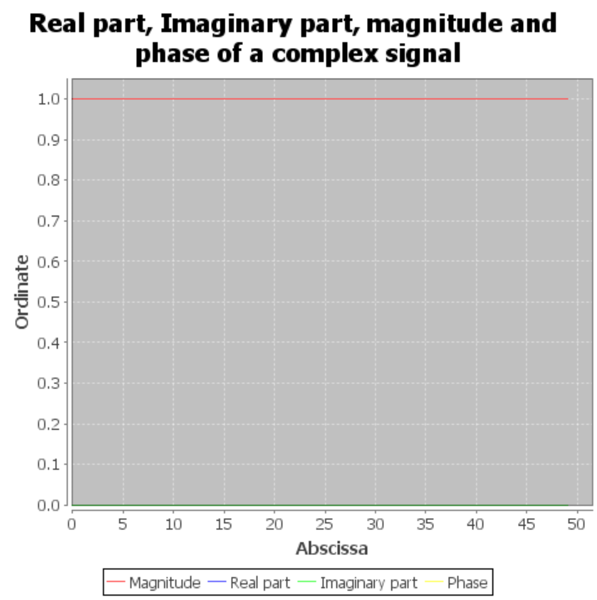
```
public static Signal generateDelta(int nonZeroSampleNumber, int
numberOfSamples) {
   Signal signal = new Signal();
   signal.settName("Delta_" + nonZeroSampleNumber);
   // Write your code here
   for (int n=0; n<=numberOfSamples-1; n++){
      if (n==nonZeroSampleNumber)
         signal.addElement(n, 1);
      else
         signal.addElement(n, 0);
   }
   return signal;
  }
```

2.4) Apply Discrete Fourier Transform on an impulse of 50 samples, with the non-zero sample at index 0. Verify the previous questions and comment what you obtain.

Delta function



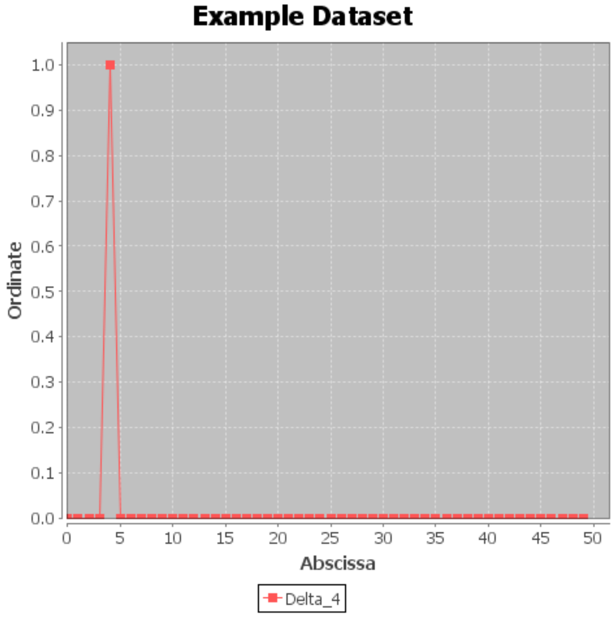
DFT of Delta function



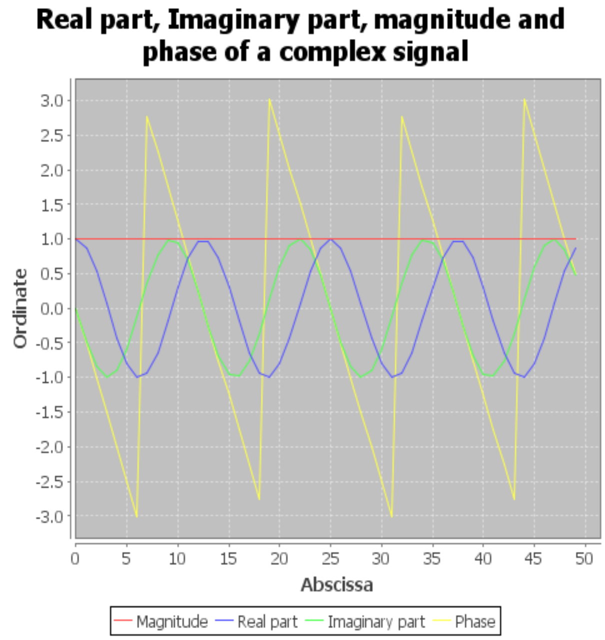
With the non-zero sample at index 0, the Discrete Fourier Transform of the function is 1, with the real part equals 1, the imaginary part equals 0, the magnitude equals 1 and the phase equals 0.

2.5) What happens if the non-zero sample is at index 4 ?

Delta function



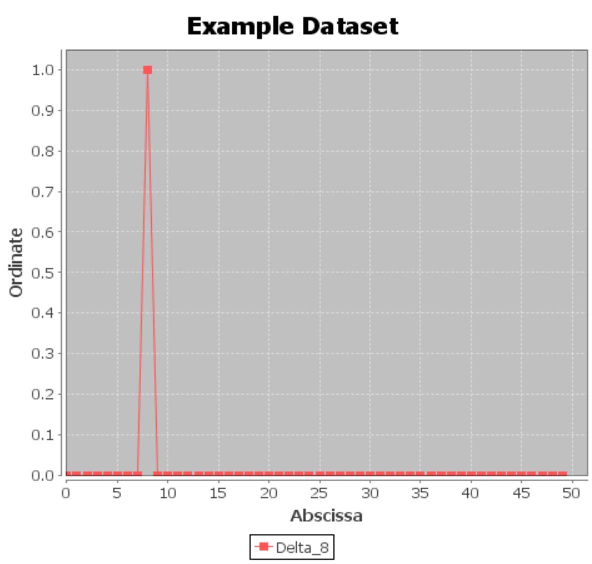
DFT of Delta function



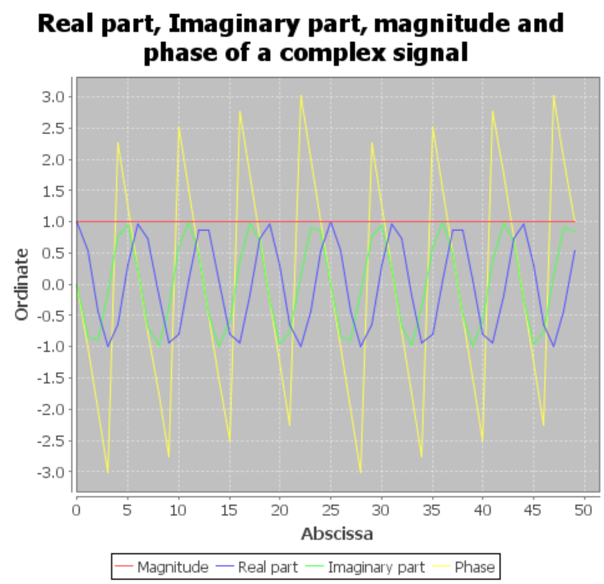
The DFT becomes very different from the previous one, since the phase and the imaginary part is not 0 (zero) anymore, and the real part is not 1 anymore. The shape of the real part is a shape's cosine, the shape of the imaginary part is a shape's sine, the magnitude is constant equals to 1, the graphic of the phase is a shape with 4 peaks (the same non-zero index of the delta function). The shape of the phase can be flipped twice, since the middle of the image, horizontally and vertically.

2.6) What happens if the non-zero sample is at index 8 ?

Delta Function



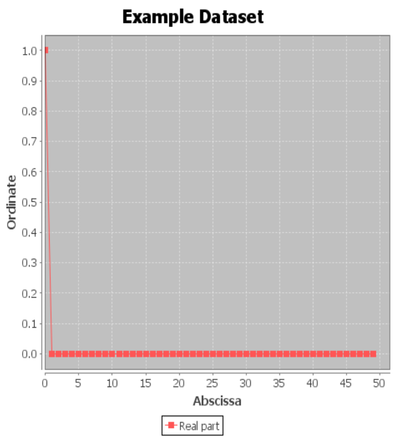
DFT of Delta Function



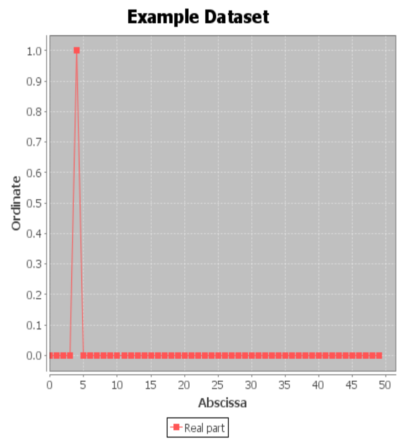
When the index of the non-zero value increases from 4 to 8, the frequency increases also and the number of peaks follows the value of the index (is 8 now).

2.7) Compute the Discrete Inverse Fourier Transform of your previous result signals. What do you observe? Open the result files in a text editor. What do you observe?

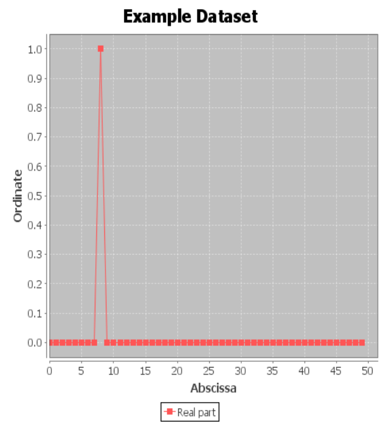
IDFT of Delta Function-0



IDFT of Delta Function-4



IDFT of Delta Function-8



When we compute the Discrete Inverse Fourier Transform of the result of the Discrete Fourier Transform, the result is the original function: $\mathcal{F}_D^{-1}(\mathcal{F}_D f)[x] = f[x]$. Opening the result files of the IDFT, we observe that the samples are very close to the original samples (that ones that were applied at the DFT). For instance, when a=4, the samples of the signal generated for the delta functions are 0 at all the points except at the point 4, when is 1. The samples of the IDFT signal is this case is 1 at point 4 and values approximately equal to 0 at the others (example of some values: 2.886579864025407E-17, -8.881784197001253E-17). The results are not exactly equals to 0 because of some rounding that the function performs.

3 Fourier Transform of a Rectangle

3.1) Calculate the Continuous Fourier Transform of II.

$$II(x) = \begin{cases} 0 & \text{if } x< -0,5 \text{ or } x>0,5, \\ 1 & \text{otherwise} \end{cases}$$

$$\mathcal{F}(II)(\omega) = \int_{x=-\infty}^{+\infty} II(x) e^{-i\omega x}\, dx$$

=> If II(x)=0:

$$\mathcal{F}(II)(\omega) = \int_{x=-\infty}^{-1/2} 0 = 0 \quad \text{and} \quad \mathcal{F}(II)(\omega) = \int_{x=+1/2}^{+\infty} 0 = 0$$

=> If II(x)=1:

$$Ƒ(II)(\omega)=\int_{x=-1/2}^{+1/2} 1\, e^{-i\omega x}\, dx$$

Using the tip of the paper:

Tip:

$$\int_a^b e^{\alpha x}dx = \left[\frac{1}{\alpha}e^{\alpha x}\right]_a^b = \frac{1}{\alpha}e^{\alpha b} - \frac{1}{\alpha}e^{\alpha a}$$

$$Ƒ(II)(\omega) = \left[\frac{1}{-i\omega}e^{-i\omega x}\right] \text{from } -1/2 \text{ to } 1/2 = \frac{1}{-i\omega}e^{-i\omega/2} - \frac{1}{-i\omega}e^{i\omega/2} = \frac{(e^{-i\omega/2}-e^{i\omega/2})}{-i\omega} = \frac{(e^{i\omega/2}-e^{-i\omega/2})}{i\omega}$$

3.2) Prove that |Ƒ(II)| = |sinc|.

$$Ƒ(II)(\omega)=\frac{(e^{i\omega/2}-e^{-i\omega/2})}{i\omega}$$

We can write the transform in terms of the oscillation frequency x instead of angular frequency ω=2лx (http://mathworld.wolfram.com/FourierTransform.html).

$$= \frac{(e^{i2лx/2}-e^{-i2лx/2})}{i2лx} = \frac{(e^{iлx}-e^{-iлx})}{i2лx}$$

Using the Euler's relations: $\sin(\Theta)=\frac{(e^{i\Theta}-e^{-i\Theta})}{2i}$

$$= \sin\frac{(лx)}{лx} = sinc(x)$$

3.3) If ´Y is a real or a complex number, ´Y ≠1, and* p *and* q *are any integers, show that (see paper):

In order to prove the equation $\sum_{n=p}^{q} ´Y^n=\frac{(´Y^p-´Y^{q+1})}{(1-´Y)}$ we will use this mechanism:

$$\sum_{n=p}^{q} ´Y^n=\sum_{n=0}^{q} ´Y^n-\sum_{n=0}^{p-1} ´Y^n$$

We know that:

Tip: Recall that $\sum_{n=0}^{N}\gamma^n = \frac{1-\gamma^{N+1}}{1-\gamma}$.

So:

$$\sum_{n=0}^{q} ´Y^n = \frac{(1-´Y^{q+1})}{(1-´Y)} = \frac{(1-(´Y^q ´Y^1))}{(1-´Y)}$$

$$\sum_{n=0}^{p-1} \text{´Y}^n = \frac{(1-\text{´Y}^{p-1+1})}{(1-\text{´Y})} = \frac{(1-\text{´Y}^p)}{(1-\text{´Y})}$$

Using the mechanism:

$$\sum_{n=p}^{q} \text{´Y}^n = \frac{(1-(\text{´Y}^q\text{´Y}))}{(1-\text{´Y})} - \frac{(1-\text{´Y}^p)}{(1-\text{´Y})} = \frac{(1-\text{´Y}^q\text{´Y}-1+\text{´Y}^p)}{(1-\text{´Y})} = \frac{(\text{-´Y}^q\text{´Y}+\text{´Y}^p)}{(1-\text{´Y})} =$$

$$= \frac{(\text{´Y}^p\text{-´Y}^{q+1})}{(1-\text{´Y})}$$

3.4) Prove that for each n, $Ƒ_D(II_m)[k] = sinc_m[k]$.

=> Developing $Ƒ_D(II_m)[k]$:

$$Ƒ_D(II_m)[k] = \sum_{k=0}^{N-1} (II_m)\, e^{-i2\text{л}kn/N} = \sum_{k=0}^{N-1} e^{-i2\text{л}kn/N}$$

We know that the rectangle function has the value 1 only when -m≤x<≤m. So:

$$Ƒ_D(II_m)[k] = \sum_{k=-m}^{m} e^{-i2\text{л}kn/N} = \sum_{k=-m}^{m} (e^{-i2\text{л}k/N})^n$$

We apply the previous proof:

$$\sum_{n=p}^{q} \gamma^n = \frac{\gamma^p - \gamma^{q+1}}{1-\gamma}$$

$$= \frac{((e^{-i2\text{л}k/N})^{-m} - (e^{-i2\text{л}k/N})^{m+1})}{(1-e^{-i2\text{л}k/N})} = \frac{((e^{i2\text{л}km/N}) - (e^{-i2\text{л}k/N})^{m+1})}{(1-e^{-i2\text{л}k/N})} \quad (1)$$

=> Developing $sinc_m[k]$:

$$sinc_m[k] = \frac{(\sin(2\text{л}k/N\,(m+1/2)))}{(\sin(\text{л}k/N))}$$

Using the Euler's relation:

$$= \frac{\dfrac{(e^{i(2\text{л}k/N(m+1/2))} - e^{-i(2\text{л}k/N(m+1/2))})}{2i}}{\dfrac{(e^{i\text{л}k/N} - e^{-i\text{л}k/N})}{2i}} = \frac{(e^{i(2\text{л}k/N(m+1/2))} - e^{-i(2\text{л}k/N(m+1/2))})}{(e^{i\text{л}k/N} - e^{-i\text{л}k/N})} = \frac{(e^{i2\text{л}km/N}\, e^{i\text{л}k/N} - e^{-i2\text{л}km/N}\, e^{-i\text{л}k/N})}{(e^{i\text{л}k/N} - e^{-i\text{л}k/N})}$$

$$= \frac{(e^{i\text{л}k/N}(e^{i2\text{л}km/N} - e^{-i2\text{л}km/N}\, e^{-i2\text{л}k/N}))}{(e^{i\text{л}k/N}(1-e^{-i2\text{л}k/N}))} = \frac{(e^{i2\text{л}km/N} - e^{-i2\text{л}km/N}\, e^{-i2\text{л}k/N})}{(1-e^{-i2\text{л}k/N})} = \frac{(e^{i2\text{л}km/N} - (e^{-i2\text{л}km/N})^{m+1})}{(1-e^{-i2\text{л}k/N})} \quad (2)$$

(1) = (2)

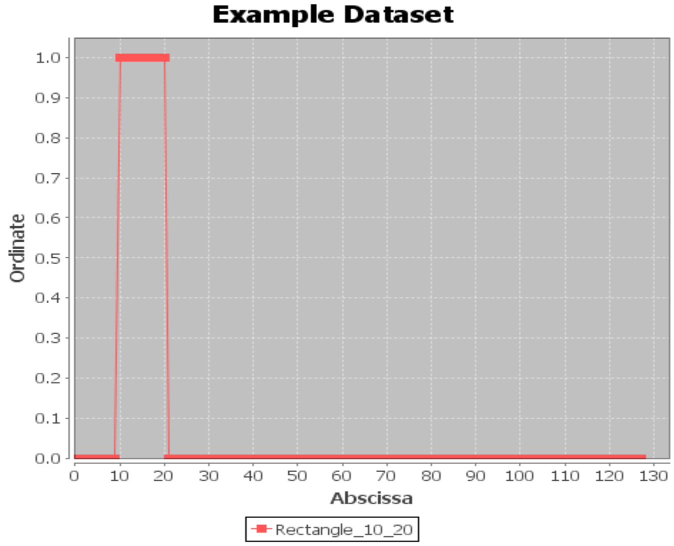
3.5) Implement the function public static Signal generateRectangle(int firstNonZeroSample, int lastNonZeroSample, int numberOfSamples) in the class Signal (refer to the comments for parameters definitions).

```
public static Signal generateRectangle(int firstNonZeroSample, int
lastNonZeroSample, int numberOfSamples) {
    Signal signal = new Signal();
    signal.settName("Rectangle_" + firstNonZeroSample + "_" +
lastNonZeroSample);

    // Write your code here
    for (int n=0; n<=numberOfSamples-1; n++){
        if ((n<firstNonZeroSample) || (n>lastNonZeroSample))
                signal.addElement(n, 0);
        else
            signal.addElement(n, 1);
    }
    return signal;
  }
```

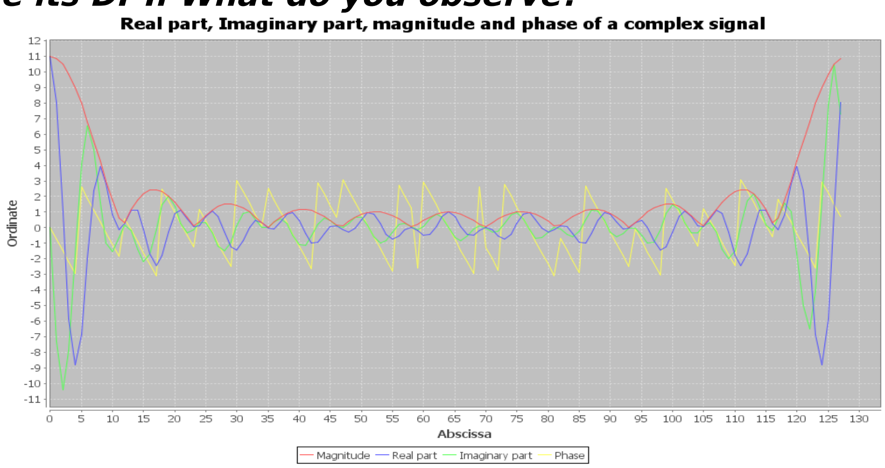
3.6) Implement the function public static ComplexSignal generateComplexSinc(int numberOfSamples, int m) in the class Signal (refer to the comments for parameters definitions).

```
public static ComplexSignal generateComplexSinc(int numberOfSamples, int m) {
    ComplexSignal signal = new ComplexSignal();

    // Write your code here
    signal.add(Complex.createFromPolar((double)m, 0.0));
    for (int n=1; n<=numberOfSamples-1; n++){
        double mag=Math.sin( (2.0 * Math.PI * (double)n / (double)
numberOfSamples) * (((double) m) + 0.5)  );
        mag = mag / ( Math.sin(Math.PI * (double)n/(double)numberOfSamples) );
        signal.add(Complex.createFromPolar(mag, 0.0));
    }
    return signal;
  }
```

3.7) Generate a rectangular signal with value 0 from sample number 0 to sample number 9, value 1 from sample number 10 to sample number 20 and value 0 from sample number 21 to sample number 127.



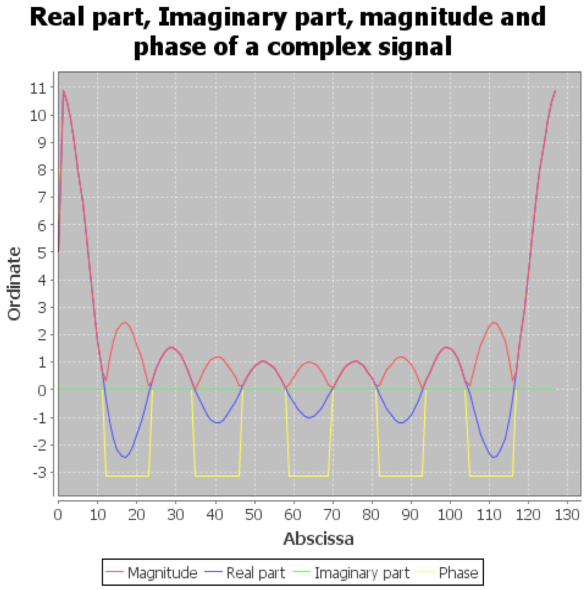
3.8) Compute its DFT. What do you observe?



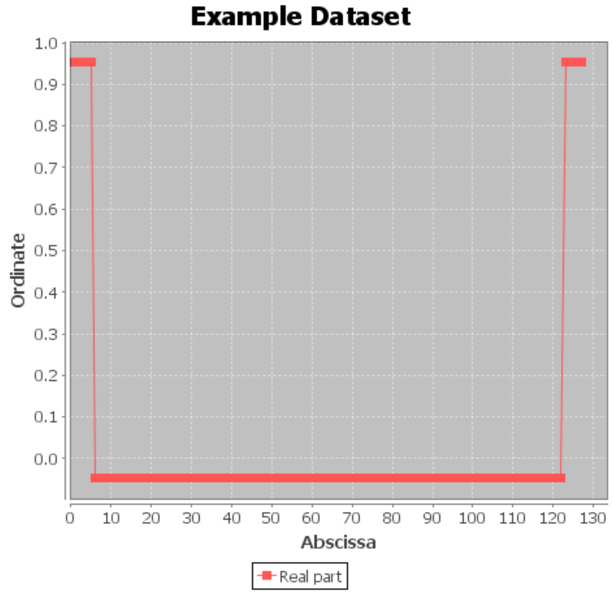
The Fourier Transform is the frequency spectrum of a rectangular wave signal which is varying in time. The shape is symmetric, the number of peaks of the DFT is the width of the rectangle. The height of the first peak is approximately equals to the width of the rectangle. The next peaks becomes flatter until the middle of the shape, when it can be flipped horizontally. The magnitude's shape follows the variation of the real and the imaginary parts.

3.9) Generate a sinc$_m$ signal with m equal to the half width of the previous signal (i.e. m = 5) on 128 samples. Compute its Discrete Inverse Fourier Transform. What do you observe?

Sinc$_m$



IDFT



Since the sinc is the Fourier Transform of the rectangle ($\mathcal{F}$(II) = sinc), the Inverse Fourier Transform of the sinc should be the rectangle function ($\mathcal{F}^{-1}$(sinc) = II). But we observe with this signal that it's not exactly the rectangle function, but the rectangle function with a shifting of size m, where m is the half width of the rectangle.

3.10) Why is the Discrete Inverse Fourier Transform of your sinc$_m$ function not the rectangular signal you created ? If you try to see it as a rectangular signal, what can you say?

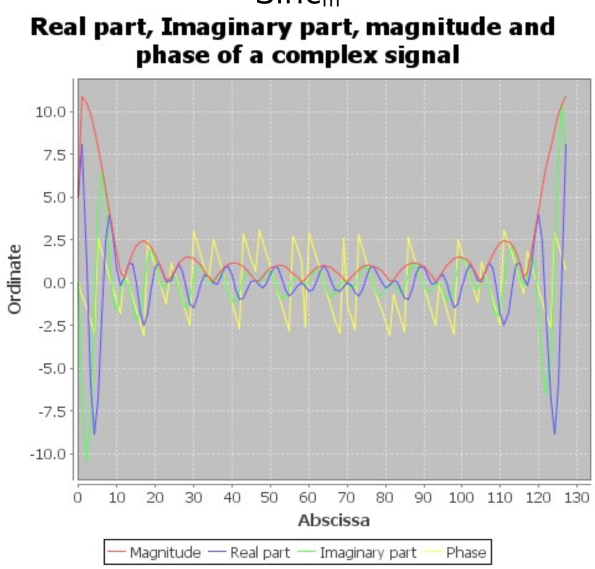
Because the rectangular function is normally centralized at the axis y. The signal we created before is not centralized at the axis y, but instead of that it is shifted by 15 (we first shift the negative part of the function until reach the ordinate 0, then we shift more 10 points right). The Fourier Transform of a m-shifted function is the Fourier of (the function $_x$ $e^{im2\pi n/N}$). In order to be the same, we should multiple the sinc$_m$ function for $e^{im2\pi n/N}$ and apply the DIFT of the result. So we'll have the shifted rectangular function.

3.11) Modify the $sinc_m$ signal generation function so that its Discrete Inverse Fourier Transform resembles to the signal you create question 3.7.

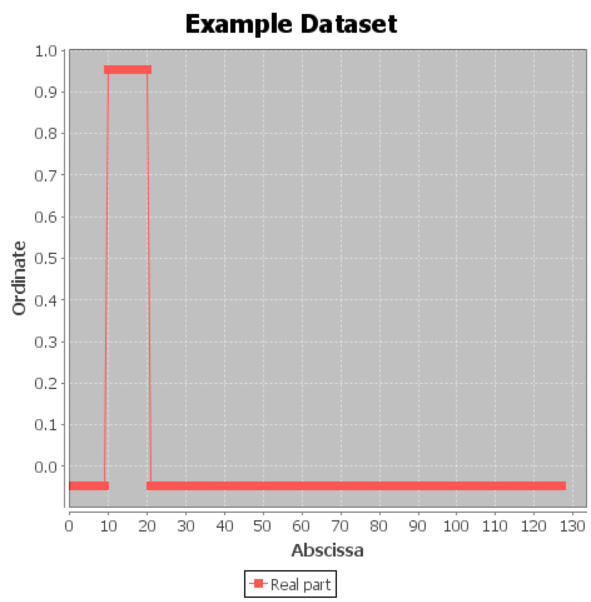
```
public static ComplexSignal generateComplexSinc(int numberOfSamples, int m) {
    ComplexSignal signal = new ComplexSignal();

    // Write your code here
    signal.add(Complex.createFromPolar((double)m, 0.0));
    for (int n=1; n<=numberOfSamples-1; n++){
        double mag=Math.sin( (2.0 * Math.PI * (double)n / (double)
numberOfSamples) * (((double) m) + 0.5)  );
        mag = mag / ( Math.sin(Math.PI * (double)n/(double)numberOfSamples) );
        Complex x;
        x=Complex.createFromPolar(mag, 0.0);

        /*Question 3.11*/
        double SHIFT=-15;
        Complex x2=Complex.createFromPolar(1,
(2*Math.PI*n*SHIFT)/numberOfSamples);
        x.mul(x2);
        /*End*/

        signal.add(x);
    }
    return signal;
  }
```

3.12) What do you observe?

$Sinc_m$



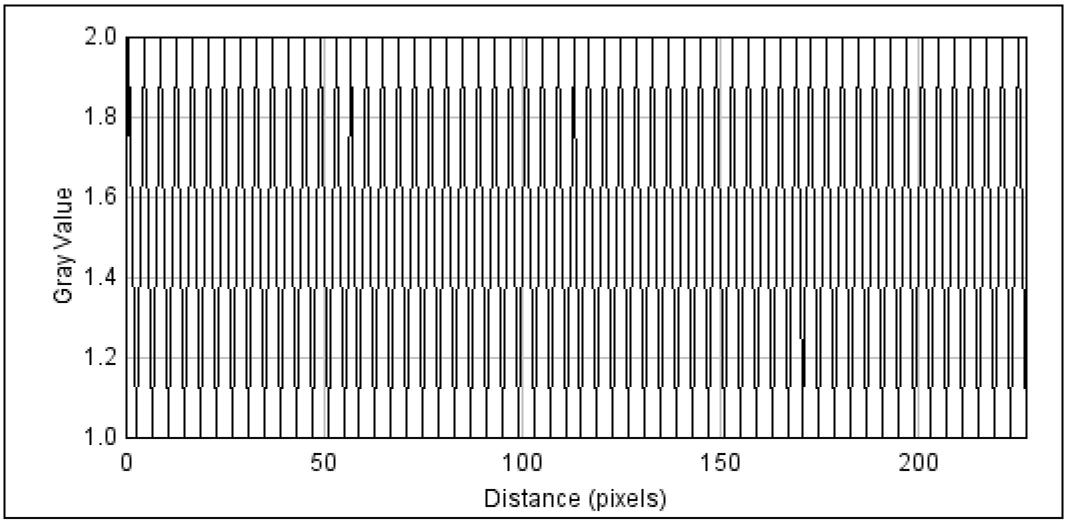
IDFT



After multiplying the $sinc_m$ function for $e^{im2\pi n/N}$ and applying the DIFT of the result, the generated signal is equal to the shifted rectangular function of the question 3.7.

4 2D Fourier Transform with ImageJ

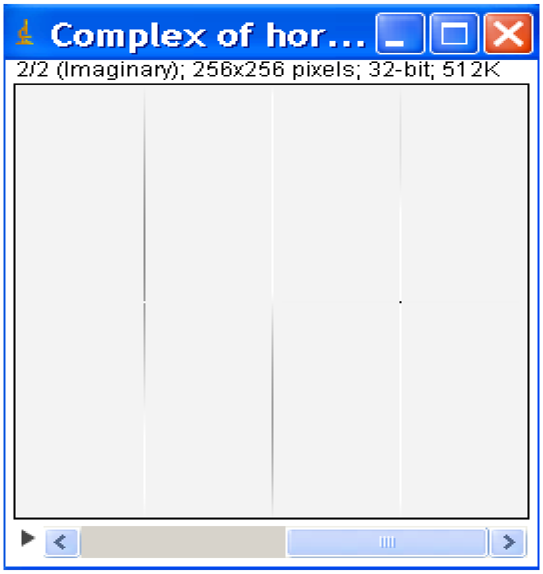
4.1) Open the file horizontal.gif with imageJ. Display the profile of a line of this image. Explain what you see.



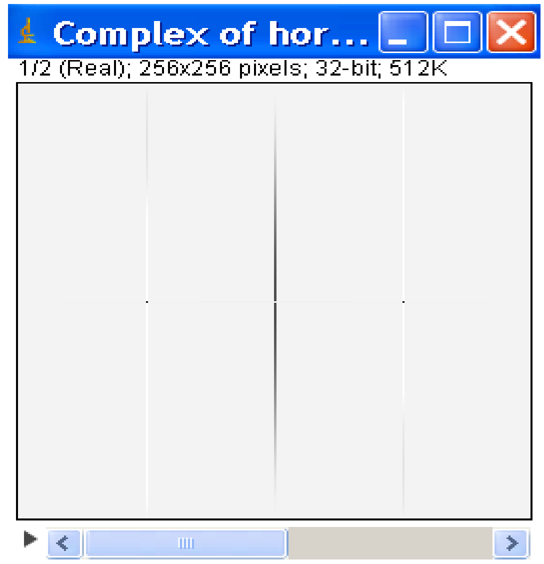
ImageJ displays a graph of with x-axis is distance of pixels, and y-axis is gray value. The length of x-axis is from 0 to 250, the length of y-axis is from 0 to 2. In the graph, there are four intensity of gray color of the line which they represent (real part, imaginary part, phase part and magnitude part of Fourier transform) frequencies of signal (sine wave). Therefore, the x-axis represents the frequency Hz of the signal, and y-axis represents the amplitude of the signal.

4.2) Compute the FFT of the image horizontal.gif. Explain what you see.

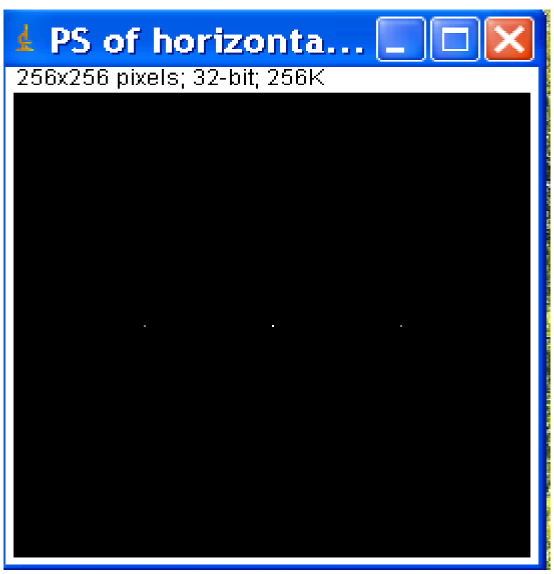
Imaginary Part



Real Part



Raw Power Spectrum



Fast Fourier Transform

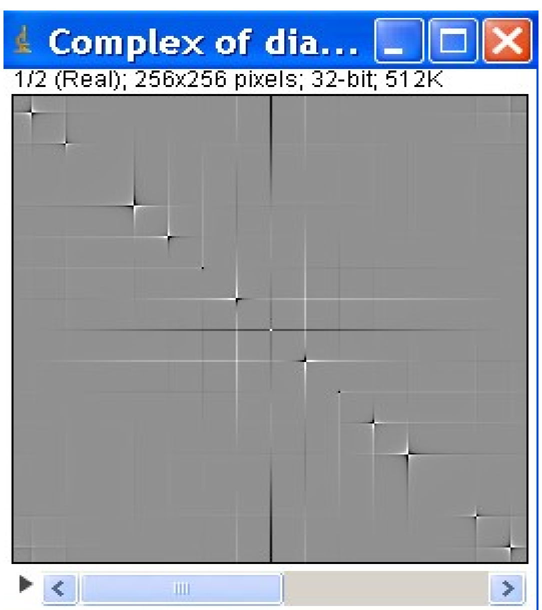
Three images are displayed on the screen. First one is the Complex of horizontal (Phase) with 32-bit image is a stack with two slices for real part and imaginary part of the FFT. Second one is raw power spectrum (Magnitude) is power spectrum without logarithm scaling. Last one is the FFT of horizontal, it contains 8-bit image of the power spectrum. As we know, the energy content of the signal over its frequencies is given the power spectrum. The power spectrum is simply the magnitude of the total spectrum.

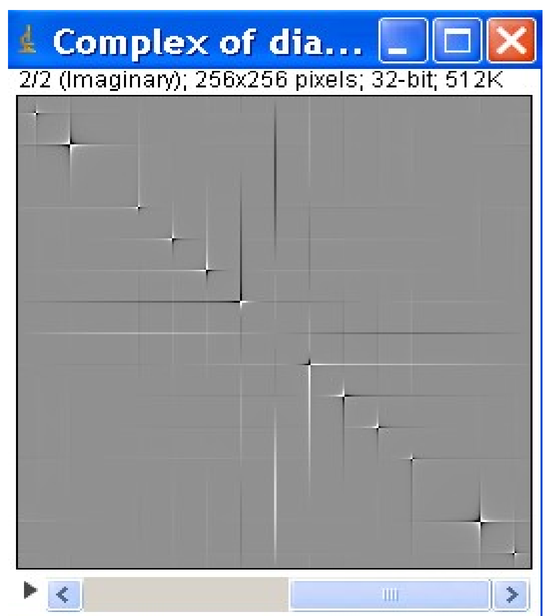
In the Magnitude picture there are 3 dots. The central dot is known as Direct Current value of image or the pixel is zero frequency which is the average color of the image. If we are changing the DC value in FFT image, we are changing the whole image. Actually any change in the DC value (the difference) will be added (or subtracted) from each and every pixel in the resulting image. The other two dots represent the perfect sine wave that the FFT operator found in the image, But why two pixels? Because a single wave can be described in two completely different ways, (one with a negative direction and phase). Both descriptions are mathematically correct, and a Fourier transform does not distinguish between them.

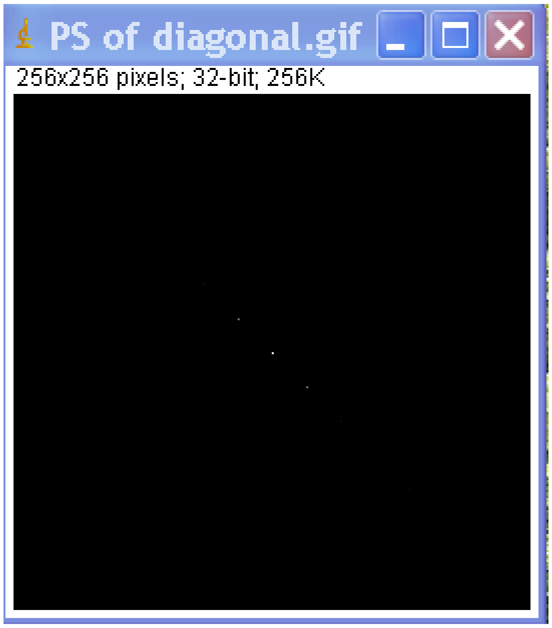
The magnitude of the original sine-save is really 1/2 but the Fourier transform divided the magnitude into two, sharing the results across both plotted frequency waves, so each one of the two components only has a magnitude of 1/4.
This duality of positive and negative frequencies in FFT images explains why all FFT image spectrum is always symmetrical about the center. For every dot on one side of the image, you will always get a similar 'dot' mirrored across the center of the image.

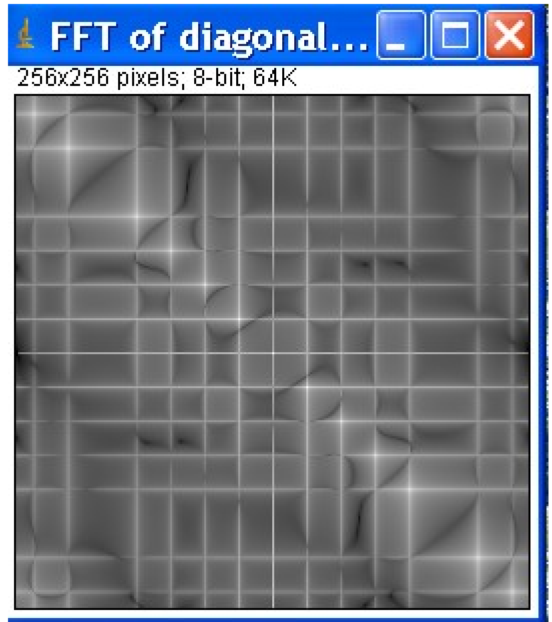
The same thing happens with the 'phase' component of FFT image pair, but with a 180 degree shift in value as well.

4.3) Open the file diagonal.gif with imageJ and compute its FFT. Explain what you see.









Three images are also displayed on the screen as above question 4.2. The first one is Complex of diagonal (Phase) of FFT, the second one is Raw Power Spectrum (Magnitude) of FFT, the last one is power spectrum image of FFT. There are also three dots in the Magnitude of FFT, the Direct Current value is in the center of the image; the two other dots are a symmetrical diagonal through central dot, because the original image is an angle sine wave. Therefore, when we convert the Magnitude image to power spectrum image, we see that many pixels are symmetrical diagonal through the central dot.

5 Free the cat

5.1) Where are the vertical bars of the cage located in the log-scale FFT magnitude?

The vertical bars of the cage located in the log-scale FFT magnitude is the white vertical line and the white horizontal line which they cut together at the central pixel of image.

Question To go further

What would you do to remove them (vertical bar, horizontal bar) in the frequency domain so that they do not appear any more after an Inverse Fourier Transform?

As we know, the simple sine wave is a repeated pattern and shows up 3 dots in the spectrum. In this case, the vertical bar (horizontal bar) is displayed by 3 dots in the spectrum. Therefore, in order to remove the vertical bar (horizontal bar) we will manually mask out the dot or lines in the magnitude image. We do this by transforming to the frequency domain, create a gray scale version of the spectrum, mask the dots or lines, threshold it, multiply the binary mask image with the magnitude image and then transform back to the spatial domain.