DFT Example : Hand Calculation:

Calculate the Discrete Fourier Transformation (DFT) of a single sine function with f0 = 1Hz , N=4, Ts = ¼ seconds , Fs = 4 Hz .  
In this example the observation window (N\*Ts) is equal to one period of the periodic time function. The definition of the normalised DFT is given below:

x(t)

x[n]

|X[k]|

t (s)

║

n\*Ts = n\*tstep

0 1\*fstep 2\*fstep 3\*fstep 4\*fstep

║ ║

Fs/2 Fs

f (Hz)

║

k\*(Fs /N) = k\*fstep

X[0] =

X[1] =

X[2] =

X[3] =

Let us calculate the complex Fourier coefficients ck for the time function x(t) defined above, and compare them to the numerical approximation X[k] calculated in the previous page.

The ck coefficients are defined as: 

But it is easier to calculate the ck coefficients for a simple harmonic function like this x(t) by comparing it to the Fourier series synthesis formula: 

There are several possibilities to calculate the ck coefficients for this x(t) function:

Try it out one or two different ways, and compare the results with the numerical approximation with the X[k] coefficients.

|  |  |
| --- | --- |
| ***Method*** | ***Start of calculation*** |
| Using the integral definition. Rather long (unnecessarily complicated for this simple x(t) ) |  |
| Comparing to synthesis formula.  Identifying which coefficients are non-zero.  Expand expression using Euler Identity. |  |
| Use symmetry properties of x(t).  For odd function expect purely imaginary ck.  Plus use properties of ck coeffs: |  |
| Compare to the formula expressing the sinus as a combination of two complex exponentials. |  |

All the methods above should deliver the same result. Only c+1 and c-1 have non-zero values, and are equal to:

Draw a graphic of the corresponding double sided spectrum and compare to the X[k] calculated on the previous page.

|ck|

f = k.f0 (Hz)

-3\*f0 -2\*f0 -1\*f0 0 1\*f0 2\*f0 3\*f0

phase(ck) [°]

f = k.f0 (Hz)

-3\*f0 -2\*f0 -1\*f0 0 1\*f0 2\*f0 3\*f0

The concept of the DFT is summarised in the figure below[[1]](#footnote-1):

DFT

N-Points

time

domain

N-Points

frequency

domain

Ts

N.Ts

t (s)

fstep

N.fstep

f (Hz)

Fs

Fs/2

Fs (N-1)/N

t (s)

fmin= fstep=1/(N.Ts) = Fs /N

fstep

The upper half of the X[k] coefficients actually corresponds to the spectrum range [-Fs/2 ; 0)

Figure 1 DFT basic idea: converting N points in time to N points in frequency domain

Fill out the table below for the example which you calculated in the previous page:

x(t)

x[n]

| X[k] |

0,5

t (s)

║

n\*Ts = n\*tstep

0 1\*fstep 2\*fstep 3\*fstep 4\*fstep

║ ║

Fs/2 Fs

f (Hz)

║

k\*(Fs /N) = k\*fstep

**Exercise 1** function with f0 = 1 Hz , N=4, Fs = 4 Hz

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Parameters of the DFT*** | | | | |  | ***Characteristics of the Signal*** | |
| Exercise Nr. | Number of points | Sampling Frequency | Frequency Resolution | Time Step = Time resolution | Observation Window |  | Fundamental Frequency | Period |
| & Time Fct | **N** | **Fs**  [Hz] | **Fs/N = fstep**  [Hz] | **Ts = tstep**  [s] | **N\*Ts = N/Fs**  [s] |  | **f0** [Hz]  and in [k\*fstep] | **T0** [s]  and in [n\*Ts] |
| (1) Single Sinus |  |  |  |  |  |  |  |  |

You can calculate the DFT numerically using the FFT[[2]](#footnote-2) (Fast Fourier Transform) function. An example code is shown below.

It is important to notice that, when you calculate the DFT numerically, you code needs to:

* define the parameters of the DFT
* define time and frequency vectors, based on a common index vector.

clear all, close all, clc;

% PARAMETERS

N = 4;

Fs = 4; %100;

aux = 0:1:N-1;

tstep = 1/Fs;

fstep = Fs/N;

t = aux\*tstep;

f = aux\*fstep;

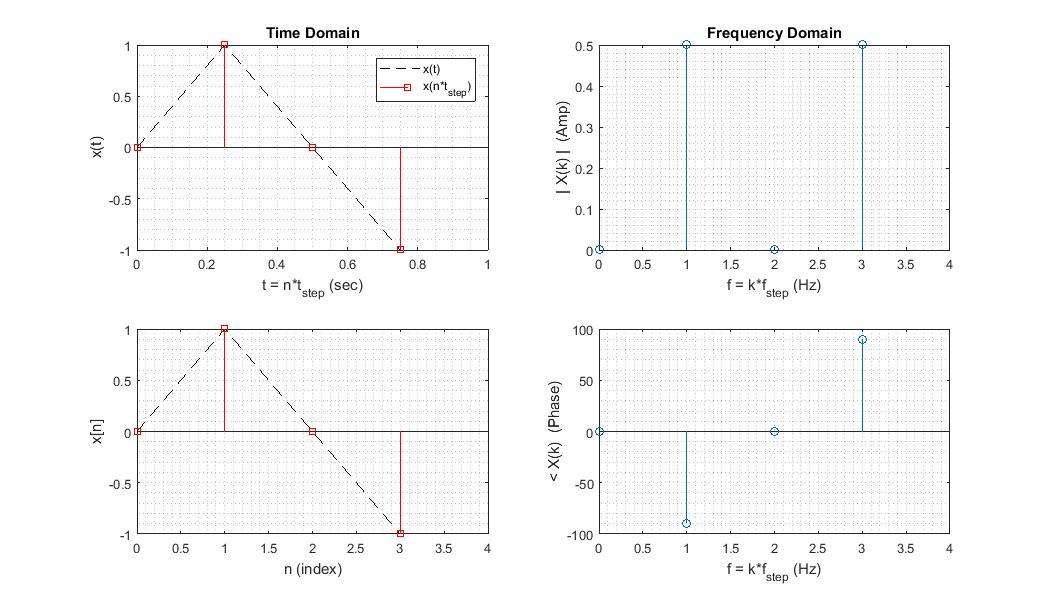
% FUNCTIONS

x\_t = sin(2\*pi\*1\*t);

X\_k = (1/N)\*fft(x\_t);

% PLOTS

….



The complete Matlab script is uploaded as *DFT\_example\_basic.m*

Read and understand the code lines describing the plot commands and the “goodies for a clearer plot”.

Then change the parameters to match the request of the following exercises. You can add for each exercise the corresponding plot as a jpg figure, to document your work.

**Exercise 2** function with f0 = 25 Hz , N=4, Fs = 100 Hz

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Parameters of the DFT*** | | | | |  | ***Characteristics of the Signal*** | |
| Exercise Nr. | Number of points | Sampling Frequency | Frequency Resolution | Time Step = Time resolution | Observation Window |  | Fundamental Frequency | Period |
| & Time Fct | **N** | **Fs**  [Hz] | **Fs/N = fstep**  [Hz] | **Ts = tstep**  [s] | **N\*Ts = N/Fs**  [s] |  | **f0** [Hz]  and in [k\*fstep] | **T0** [s]  and in [n\*Ts] |
| (2) Single Sinus |  |  |  |  |  |  |  |  |

**Exercise 3** function with f0 = 25 Hz , N=16, Fs = 100 Hz

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Parameters of the DFT*** | | | | |  | ***Characteristics of the Signal*** | |
| Exercise Nr. | Number of points | Sampling Frequency | Frequency Resolution | Time Step = Time resolution | Observation Window |  | Fundamental Frequency | Period |
| & Time Fct | **N** | **Fs**  [Hz] | **Fs/N = fstep**  [Hz] | **Ts = tstep**  [s] | **N\*Ts = N/Fs**  [s] |  | **f0** [Hz]  and in [k\*fstep] | **T0** [s]  and in [n\*Ts] |
| (3) Single Sinus |  |  |  |  |  |  |  |  |

**Exercise 4** function with f0 = 25 Hz , N=16, Fs = 400 Hz

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Parameters of the DFT*** | | | | |  | ***Characteristics of the Signal*** | |
| Exercise Nr. | Number of points | Sampling Frequency | Frequency Resolution | Time Step = Time resolution | Observation Window |  | Fundamental Frequency | Period |
| & Time Fct | **N** | **Fs**  [Hz] | **Fs/N = fstep**  [Hz] | **Ts = tstep**  [s] | **N\*Ts = N/Fs**  [s] |  | **f0** [Hz]  and in [k\*fstep] | **T0** [s]  and in [n\*Ts] |
| (4) Single Sinus |  |  |  |  |  |  |  |  |

Try now changing the time function for a cosinus, a triangle, a sawtooth and a square function, and check the corresponding spectra.

1. The concept of the DFT is discussed in chapter 2 of the script. [↑](#footnote-ref-1)
2. The FFT is an algorithm optimisation for the DFT calculation, which is specially efficient when N = 2P . It uses symmetry properties of the phasors to significantly reduce the number of factors to be calculated. [↑](#footnote-ref-2)