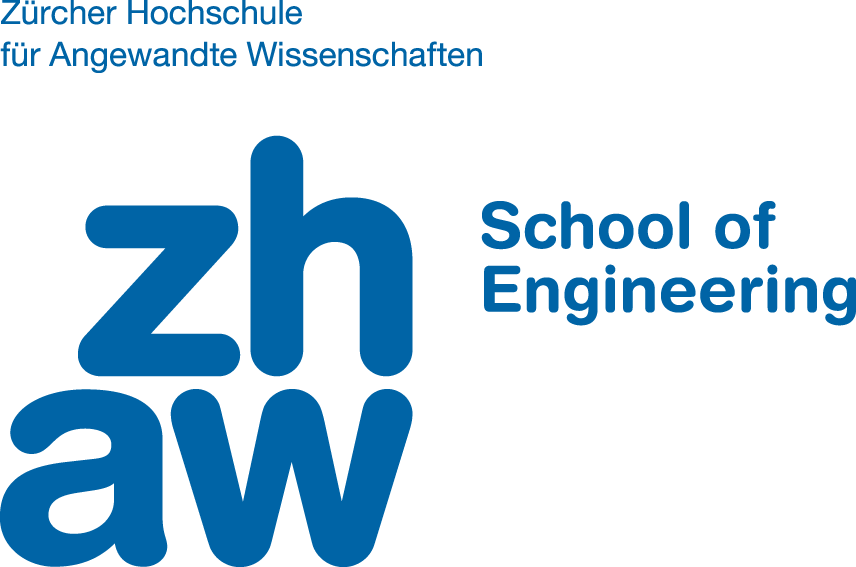
**Laboratory 2B:** Sample Solution

**Introduction (Fourier Analysis and Fourier Synthesis)**

try out the code snippets in a Matlab script : lab2b\_intro.m

% Lab 2B : Fourier Series : Intro

% ========================

clear all, close all, clc;

% Define Constants and time vector

T0 = 2; w0 = 2\*pi/T0; A = 1;

t = -2\*T0:T0/100:2\*T0;

Kmax = 10; % Max number of harmonics

c0 = A/2; % DC-Offset (average value)

% since c0 has a different expression than the other ck, will take c0

% outside of the loop, and combine with the initialisation of x(t)

x\_t = c0\*ones(1,length(t)); % initialise x(t) with DC-content

% for k = -Kmax:1:Kmax

% if k ~= 0

% ck = j\*A/(k\*2\*pi);

% x\_t = x\_t + ck\*exp(j\*k\*w0\*t);

% end

% end

% % Alternative taking ck and c-k together, and loop only with positive k

% for k = 1:Kmax

% ck = j\*A/(k\*2\*pi);

% x\_t = x\_t + ck\*exp(j\*k\*w0\*t) + conj(ck)\*exp(-j\*k\*w0\*t) ;

% end

% Alternative taking ck and c-k together, and writing Ak phik notation

for k = 1:Kmax

ck = j\*A/(k\*2\*pi);

x\_t = x\_t + 2\*abs(ck)\*cos(k\*w0\*t + phase(ck));

end

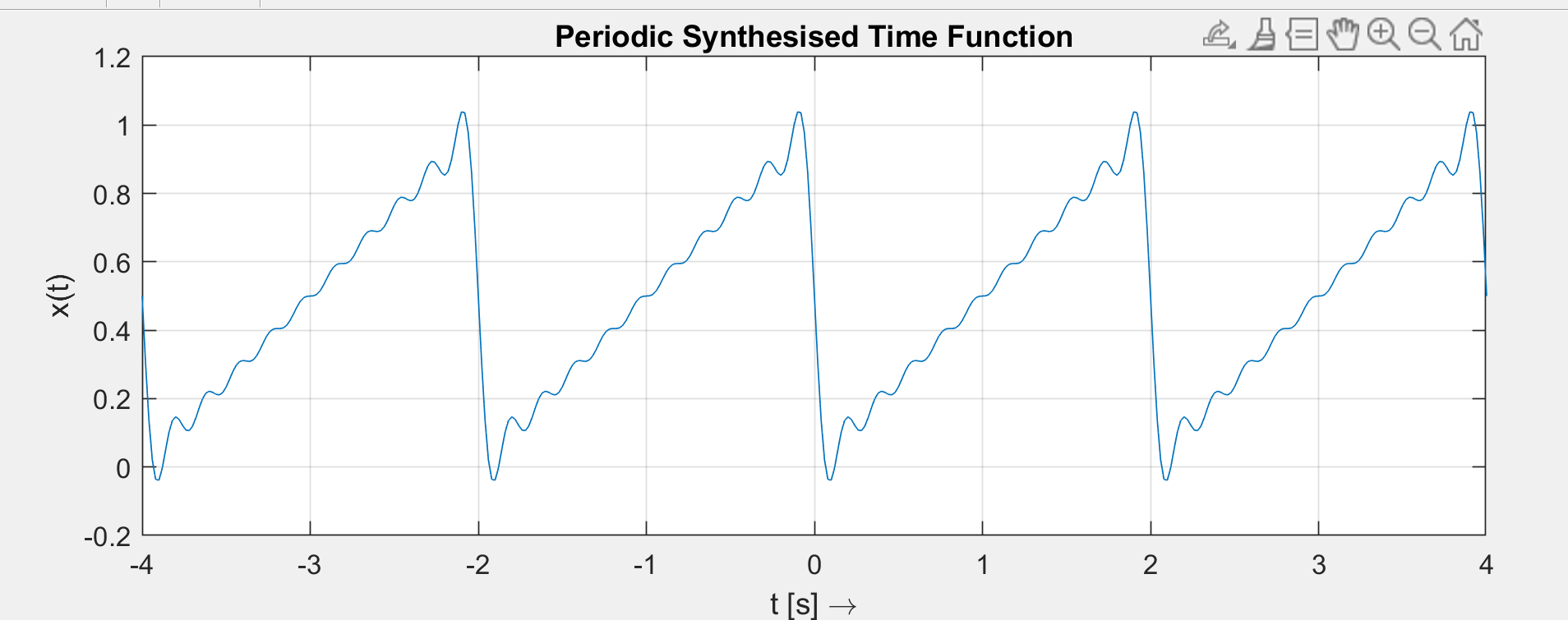
figure(), plot(t,x\_t), grid on

xlabel('t [s] \rightarrow'),ylabel('x(t)')

title('Periodic Synthesised Time Function')

disp('Because of warning, check how big mag(imag(x\_t))')

max(imag(x\_t))



# Fourier Series Coefficients: calculate and plot

Please use the ck integral definition for your calculation:



-T0 0 +T0 t

x(t)

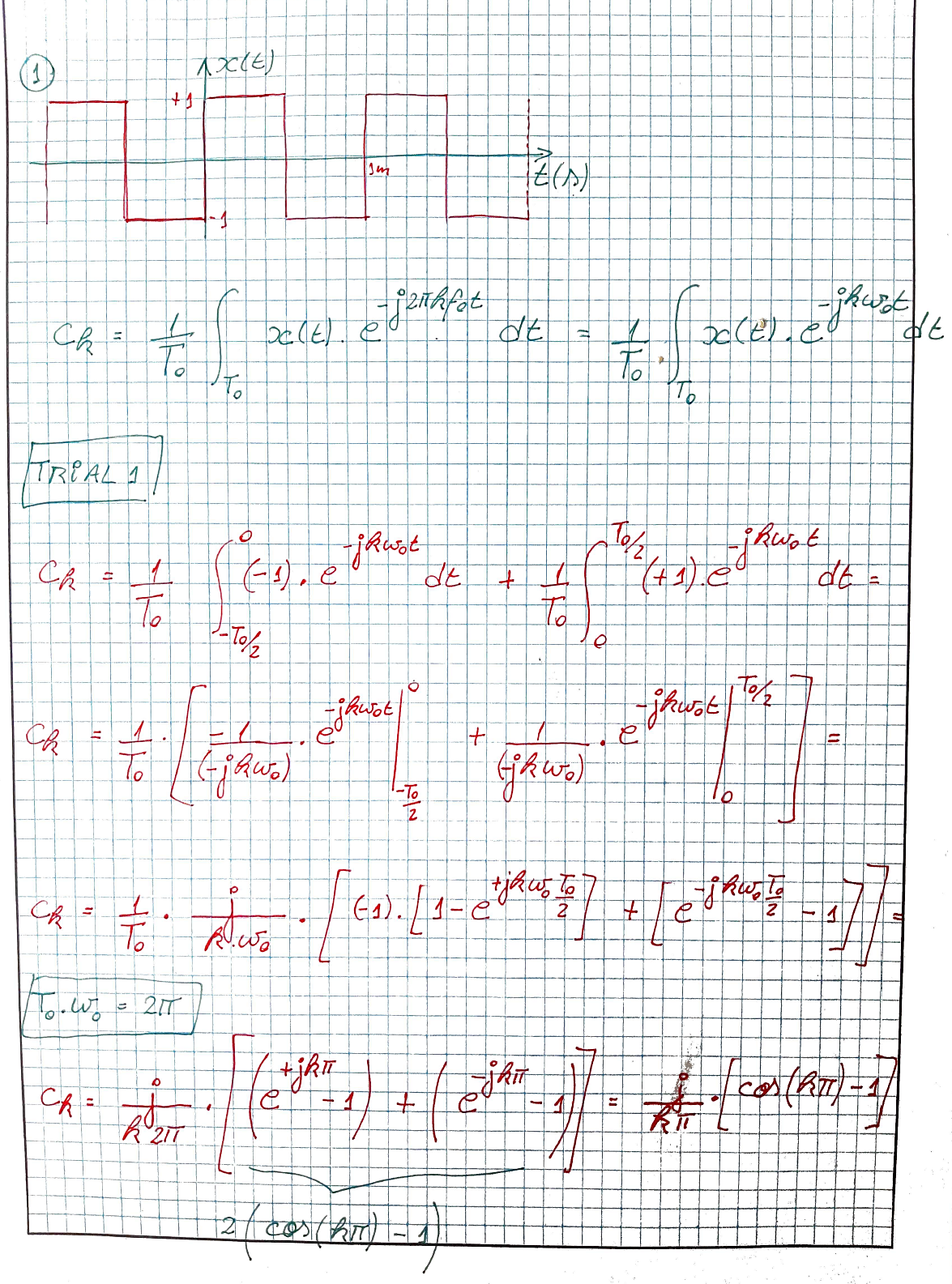
A

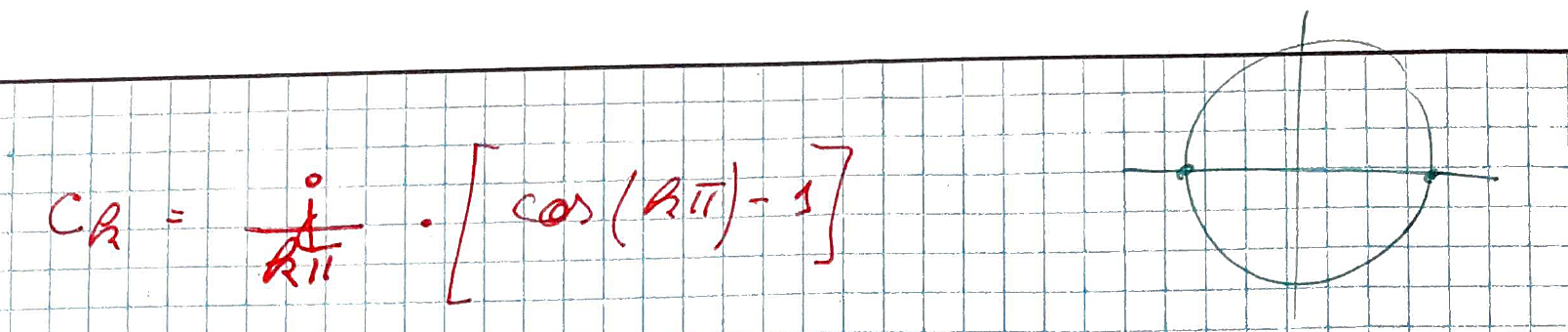
-A

Two alternatives to calculate the ck coefficients:

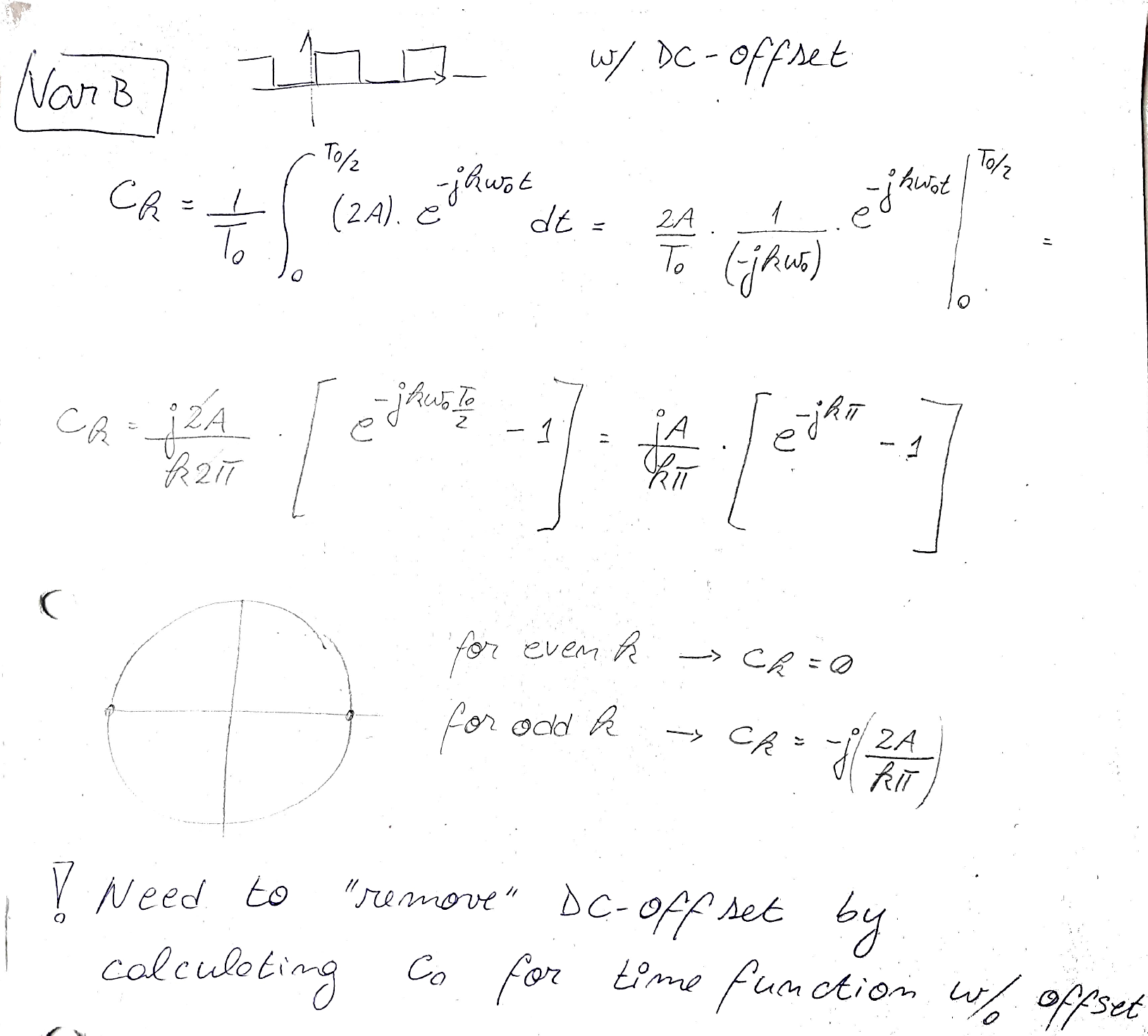
* Directly with the 2 intervals -T0/2 till 0 and 0 till T0/2
* First adding an offset, to have one of these intervals disappearing (cause amp=0)  
  In this alternative remember to remove DC-offset in the end

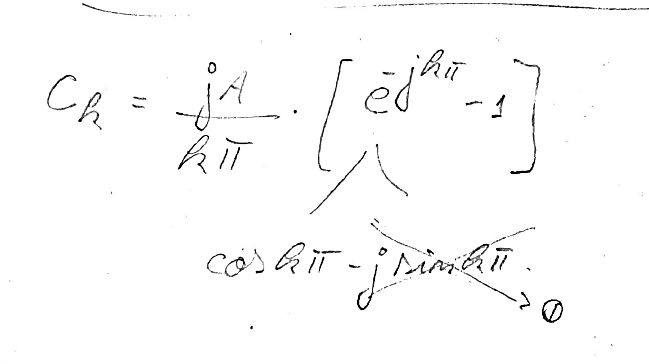
1. First alternative





Second alternative





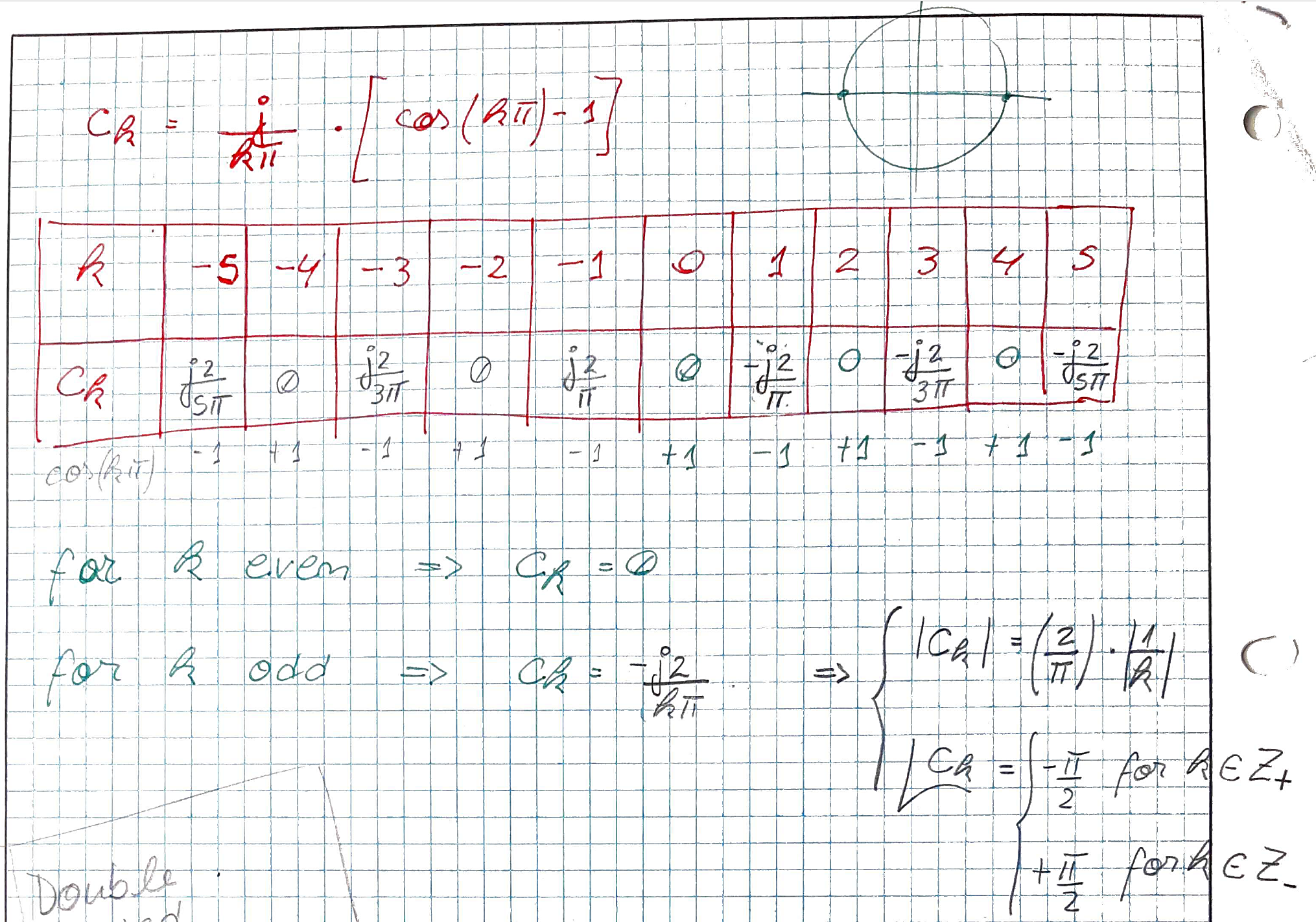
1. Do the values of ck depend on T0 ? No

Is it possible to have a simpler expression for all odd, and all even ck coefficients?

Yes (see previous page)

1. Check what are the numerical values of ck for k within [-5 ; +5] .

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 |
| ck |  |  |  |  |  |  |  |  |  |  |  |



1. Are the non-zero ck coefficients pure real or pure complex numbers? What does this mean compared to the real coefficients ak and bk ?

The ck coefficients are pure imaginary.

Expect only sine shaped harmonics (sine with phase =0 ), cause odd symmetry.

1. Check the values of your ck coefficient with a Fourier Synthesis in a Matlab script.

Using same script from introduction and changing the ck expression (lab2b\_exer\_1.m)

c0 = 0; % DC-Offset (average value)

for k = -Kmax:1:Kmax

if k ~= 0

ck = j/(k\*pi) \* (cos(k\*pi) -1);

x\_t = x\_t + ck\*exp(j\*k\*w0\*t);

end

end

**Alternative:**

% Alternative differentiating even and odd coeffs

for k = -Kmax:1:Kmax

if ((k ~= 0) & mod(k,2)~=0 )

ck = -2\*j/(k\*pi);

x\_t = x\_t + ck\*exp(j\*k\*w0\*t);

end

end

1. Prepare a plot of the corresponding Double Sided Spectrum in Matlab. You can use k (index of harmonics) or frequency in Hz for the horizontal axis.

*Hint*: you can define your f vector as f = k\*f0 , where k is the index vector for the ck coefficients and f0 is the fundamental frequency of the periodic signal x(t).

k\_vec = -Kmax:1:Kmax;

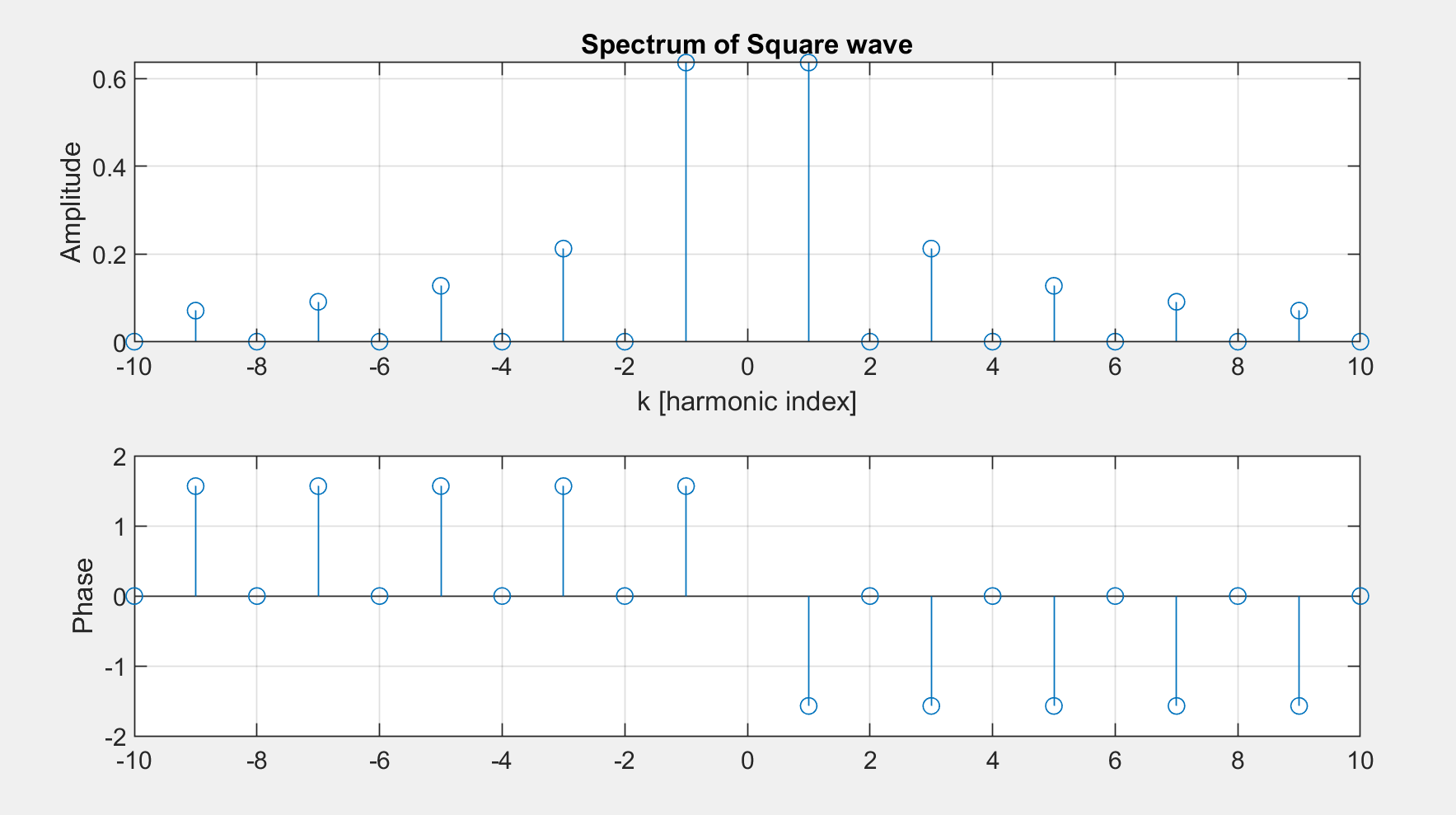
ck\_vec = j./(k\_vec\*pi) .\* (cos(k\_vec\*pi) -1);

figure(),

subplot(211),stem(k\_vec, abs(ck\_vec)), grid on, ylabel('Amplitude')

subplot(212),stem(k\_vec, angle(ck\_vec)), grid on, ylabel('Phase')

subplot(211),title('Spectrum of Square wave'), xlabel('k [harmonic index]')



# Numerical Approximation for Fourier Series with FFT

1. Start a new script in Matlab, and insert the code shown below: lab2b\_exer\_2.m
2. Execute the code and analyse the following points:
   1. How is the periodic square function x(t) defined?

Matlab function square()

x\_t = square(2\*pi\*t/T0,50); % periodic square with 50% duty cycle and period T0

* 1. With how many points per period?

N points per period. Time interval shows 1 period

N = 2^8; % number of points for the FFT

aux = 0:1:N-1; % auxiliary index vector

% Select the sampling period, to have exactly N-points/period (ideal situation)

tstep = 1\*T0/N; % resolution in the time domain

t = tstep\*aux; % time vector

* 1. Which function is used to calculate the numerical approximation of the ck coefficients?

Matlab function fft()

c\_k = (1/N)\*fft(x\_t); % normalised FFT = approx(ck coefficients)

* 1. Which ck coefficients are visible after the zoom (with xlim() function) ?

The 1st 20 harmonics ( k ϵ [0,20] )

* 1. What is plotted: amplitude-part or phase-part of the spectrum ?

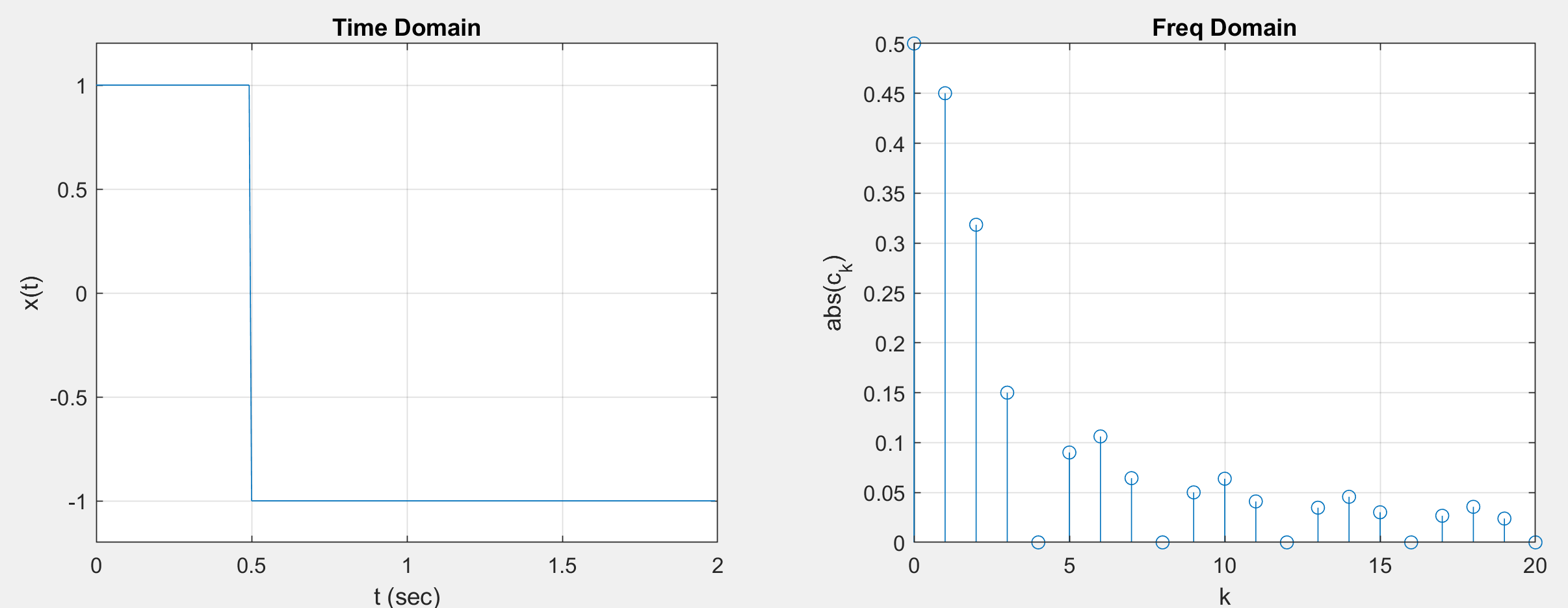
Amplitude spectrum:

stem(f,abs(c\_k)),

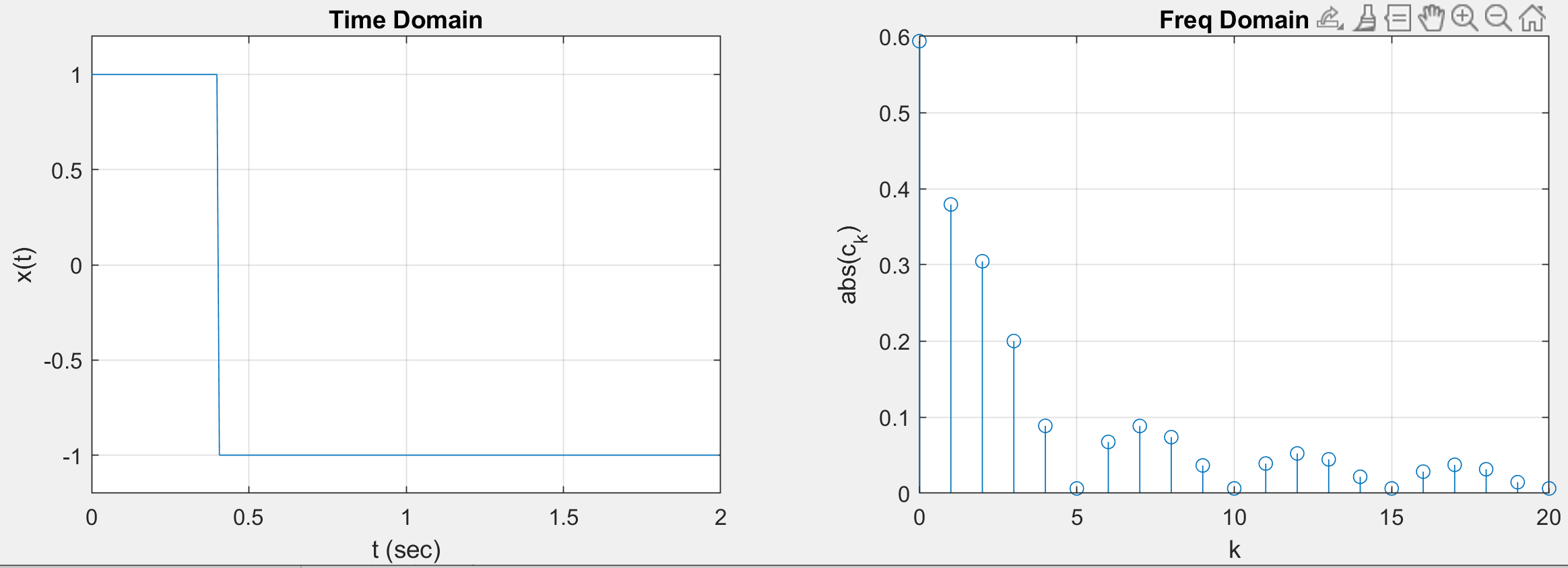
1. Vary the duty cycle of the periodic square x(t) (e.g. 50%, 25%, 20%, 10%). What happens with the corresponding spectra? Find out what are the zero-crossings of the amplitude spectrum for the different values of the duty cycle. For example for:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Duty cycle | 50% or | 25% or | 20% | 10% |
| Zero-crossings  ck = 0 for … | ck = 0 for  k = 2\*n  with n ϵ Z  ck ≠ 0 for  k = 1 ; 3 ; 5 ; 7 …  or k = 2\*n +1 | ck = 0 for  k = 4\*n  with n ϵ Z | ck = 0 for  k = 5\*n  with n ϵ Z | ck = 0 for  k = 10\*n  with n ϵ Z |

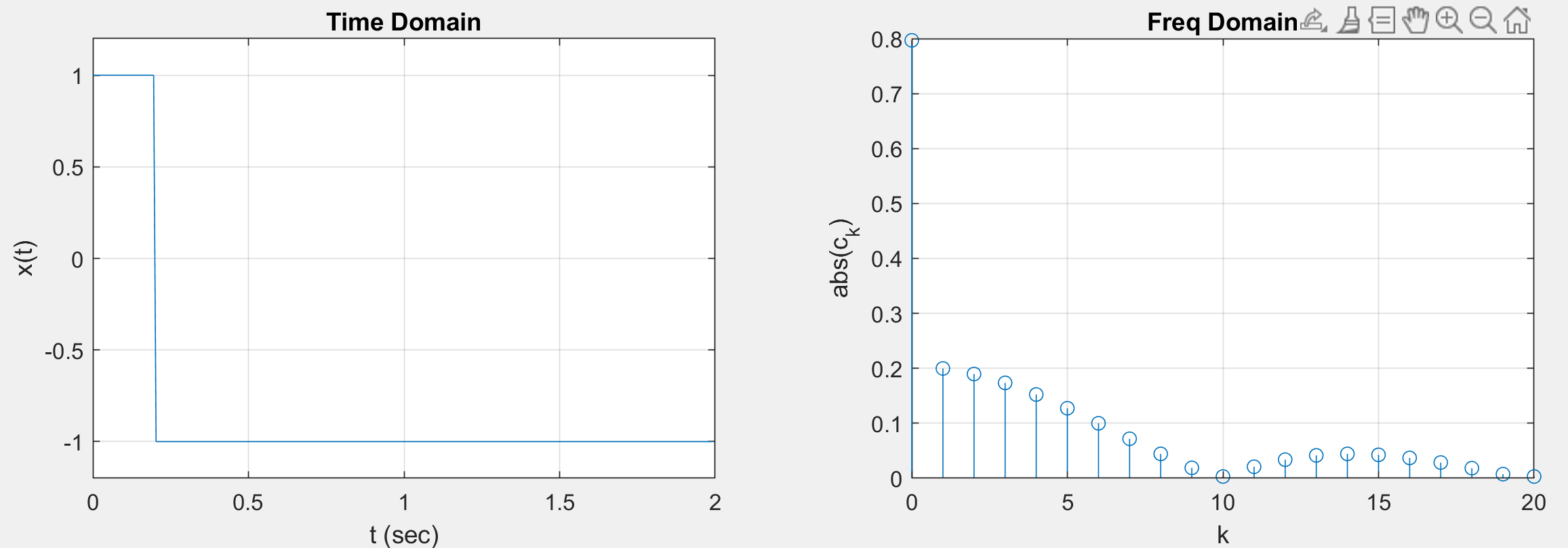
x\_t = square(2\*pi\*t/T0,25);



x\_t = square(2\*pi\*t/T0,20);



x\_t = square(2\*pi\*t/T0,10);



1. Change x(t) for a periodic ramp (use in Matlab the function sawtooth() ) . Which changes can you observe in the spectrum?

Amplitude spectrum with all ck ≠ 0 . The amplitude decays as the k increases.

Agrees with expression given in page 2

