ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Physics

Α.Μ: ***201100079***	**Επώνυμο:** ***Κορομηλάς***	**Όνομα:** ***Χρήστος***
Ημ/νία(Ημέρα,Ώρα) εκτέλεσης της άσκησης: ***13/04/2022 (Τετάρτη, 15:30-18:00)***		
Ημ/νία παράδοσης της άσκησης: 20/04/2022		

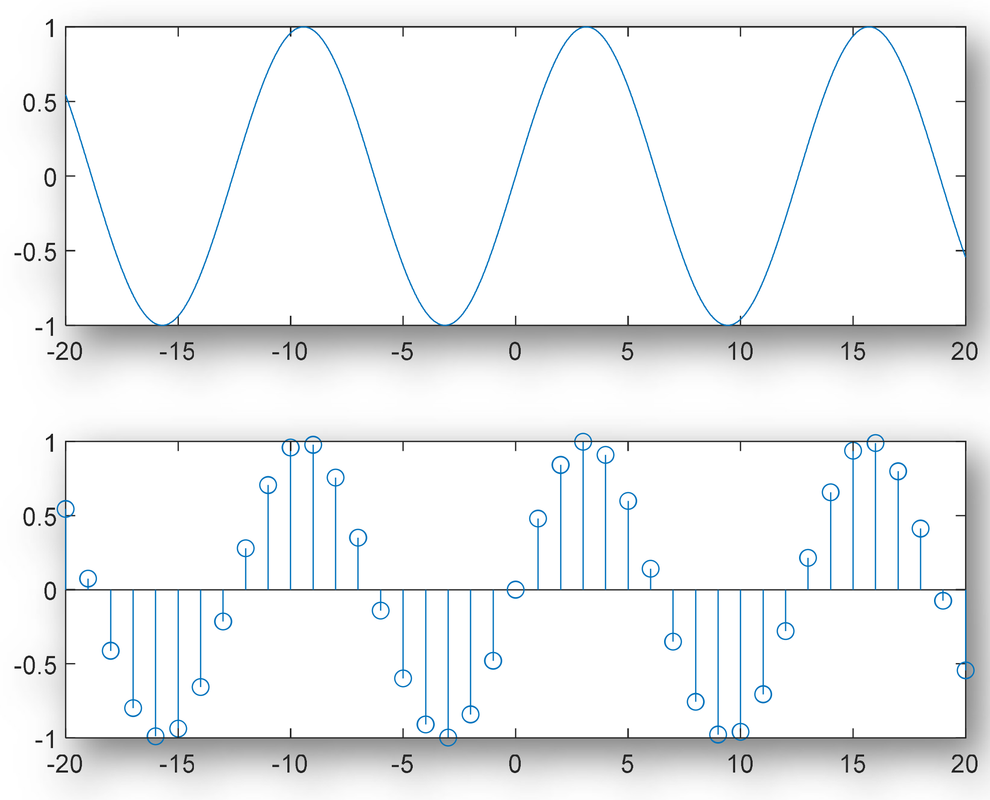
ΣΥΝΕΧΗ ΚΑΙ ΔΙΑΚΡΙΤΑ ΣΗΜΑΤΑ – ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER

Μια εργασία η οποία αποσκοπεί στο να μας εξοικειώσει περισσότερο με την πλατφόρμα MATLAB, ενώ παράλληλα εφαρμόζονται προσεγγιστικές μέθοδοι υπολογισμού του μετασχηματισμού Fourier

➢ ***Άσκηση 1η :***

Ακολουθεί το πρόγραμμα $\boldsymbol{cont\_discr1.m}$ στη λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης:

```
%% Practice code for Function Plotting

close all;
clc;
clear all;

t = -20:0.001:20;
ft = sin(t/2);

n = -20:1:20;
fn = sin(n/2);

figure(1);
subplot(2,1,1);
plot(t,ft);

subplot(2,1,2);
stem(n,fn);

print -depsc pr11;
```



Απάντηση ερωτήσεων:

Ερώτημα 1:

Το πρόγραμμα καθώς ξεκινάει δημιουργεί ένα μονοδιάστατο πίνακα όπου βάζει τις τιμές του $\boldsymbol{t}$ από το $-\mathbf{20}$ έως το $+\mathbf{20}$ με βήμα $\mathbf{0.001}$. έπειτα με αυτές τις τιμές υπολογίζει τα αντίστοιχα $\boldsymbol{f}(\boldsymbol{t}) = \boldsymbol{sin}(\boldsymbol{t}/\mathbf{2})$ τα οποία καταχωρούνται σε ένα νέο μονοδιάστατο πίνακα. Ακολουθείται η ίδια διαδικασία για τις τιμές $\boldsymbol{n}$ από το $-\mathbf{20}$ έως το $+\mathbf{20}$ αλλά τώρα με βήμα διακριτής μορφής $\mathbf{1}$ και υπολογίζεται αντίστοιχα το $\boldsymbol{f}(\boldsymbol{n}) = \boldsymbol{sin}(\boldsymbol{n}/\mathbf{2})$. Έπειτα, δημιουργείται ένα σχήμα που του δίνουμε τη τιμή $\mathbf{1}$ έτσι ώστε το $\boldsymbol{Matlab}$ να ονομάσει $\boldsymbol{Figure}\ \mathbf{1}$, με τις εντολές $\boldsymbol{subplot}$ χωρίζουμε αυτό το σχήμα σε κομμάτια και μπορούμε να επιλέγουμε σε ποιο από αυτά τα κομμάτια θα μπαίνουν οι γραφικές μας, συνεπώς επιλέγουμε σε ποιο από αυτά τα κομμάτια θα μπαίνουν οι γραφικές μας, συνεπώς επιλέγουμε στο πάνω μισό του χώρου μας, να αναπαρασταθεί με γραφική παράσταση συνεχούς μορφής η συνάρτηση $\boldsymbol{f}(\boldsymbol{t})$, ενώ στο κάτω μισό του χώρου, με διακριτής μορφής αναπαράσταση τύπου stem η συνάρτηση $\boldsymbol{f}(\boldsymbol{n})$. Τέλος το πρόγραμμα πριν τερματίσει εκτυπώνει τα δεδομένα μας σε αρχείο τύπου .eps με την ονομασία $\boldsymbol{pr}\mathbf{11}.\boldsymbol{eps}$.

Ερώτημα 2:

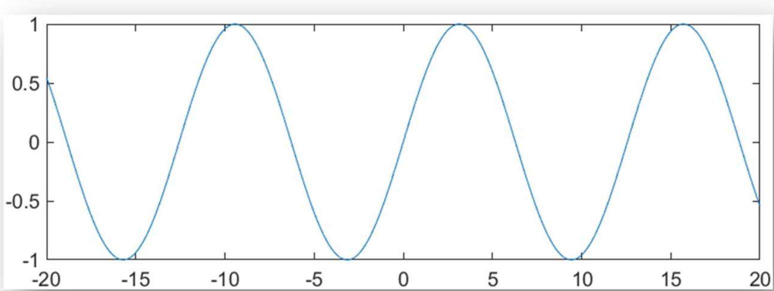
Η γραμμή 5 του εργαστηριακού οδηγού είναι η:

```
t = -20:0.001:20;
```

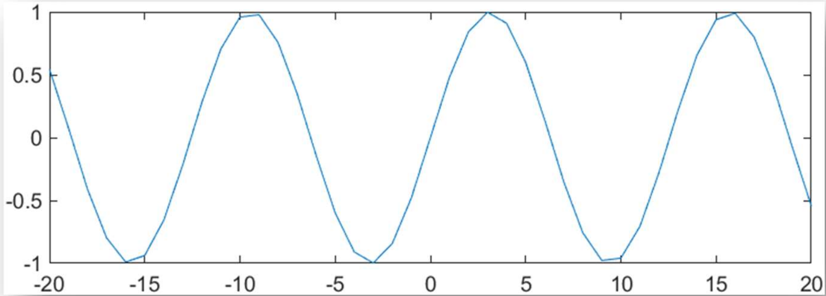
Χρησιμοποιούμε αυτή τη γραμμή κώδικα , για να μπορέσουμε να πάρουμε πολλές ισαπέχουσες τιμές του $\boldsymbol{t}$ από το $-\mathbf{20}$ έως το $+\mathbf{20}$ με βήμα $\mathbf{0.001}$, οι οποίες να έχουν τη μορφή της συνεχούς συνάρτησης(σε αντίθεση με το $\boldsymbol{n}$) και να τοποθετήσουμε στο μονοδιάστατο πίνακα $\boldsymbol{t}$.

Ερώτημα 3:

Εάν πραγματοποιούσαμε αυτή την αλλαγή ο πίνακας $\boldsymbol{t}$ θα έπαιρνε τιμές με βήμα $\mathbf{0.1}$ και κατά τη διαδικασία του $\boldsymbol{plotting}$ δεν θα βλέπαμε διαφορά όμως σε μία άλλη συνάρτηση ίσως και να μην προσέγγιζε τόσο τη συνέχεια ή ίσως να μην αντιπροσώπευε και τη πραγματική συνάρτηση που θα θέλαμε.

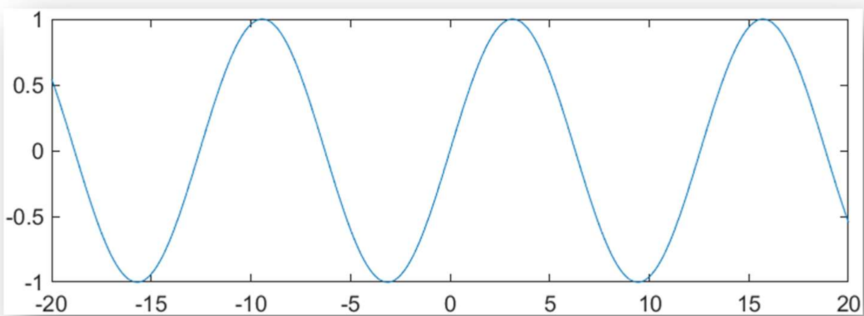


Αν πχ βάλουμε βήμα $\mathbf{1}$ *όπως στη διακριτή εκδοχή του* $\boldsymbol{n}$*:*



Ερώτημα 4*:*

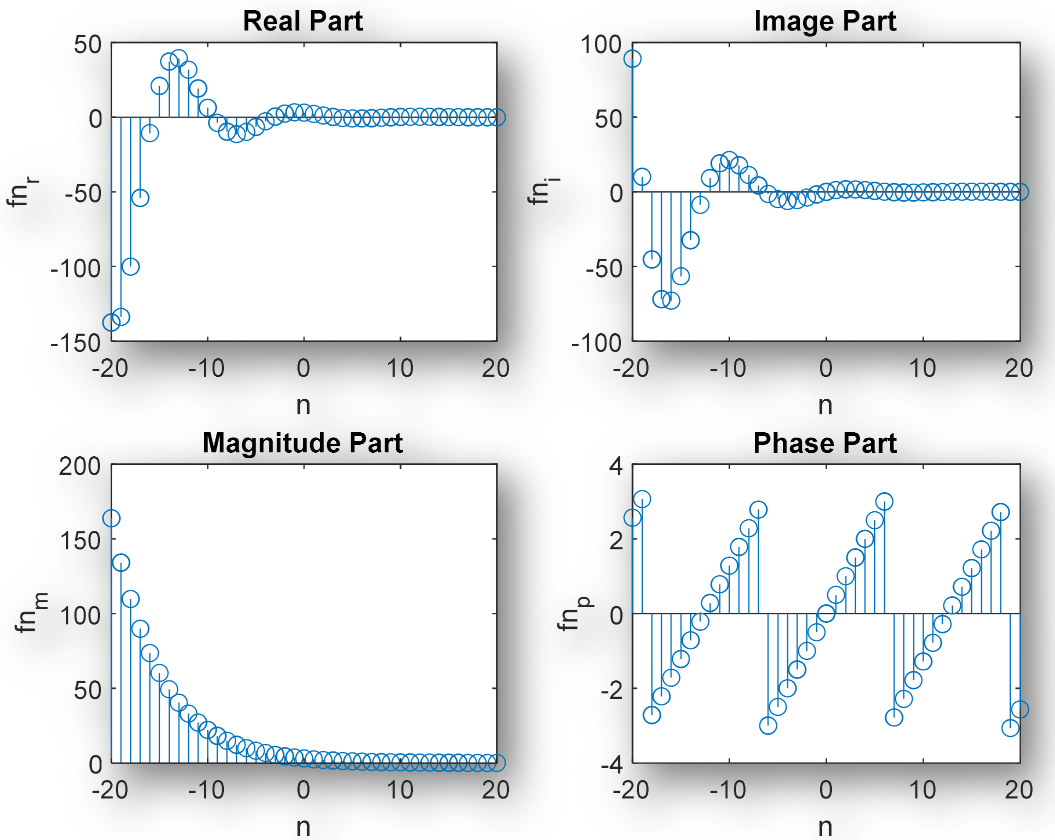
Εάν πραγματοποιούσαμε αυτή την αλλαγή ο πίνακας $\boldsymbol{t}$ *θα έπαιρνε τιμές με βήμα* $\mathbf{0.00001}$ *και κατά τη διαδικασία του* $\boldsymbol{plotting}$ *θα βλέπαμε ότι η συνάρτηση θα προσέγγιζε πιο συνεχής μορφή. Προφανώς μικρότερο βήμα προσεγγίζει πιο πολύ την συνάρτηση που πάμε να απεικονίσουμε, με μεγαλύτερη ακρίβεια.*



➢ ***Άσκηση 2η :***

Ακολουθεί το πρόγραμμα $\boldsymbol{discr2.m}$ στην λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης:

```
%%Practice code for complex Funcion analysis

close all;
clc;
clear all;

n = [-20:1:20];

fn = 3*exp((-0.2+0.5*i).*n);

fn_r = real(fn);
fn_i = imag(fn);
fn_m = abs(fn);
fn_p = angle(fn);

figure(1);

subplot(2,2,1);
stem(n,fn_r);
title('Real Part');
ylabel('fn_r');
xlabel('n');

subplot(2,2,2);
stem(n,fn_i);
title('Image Part');
ylabel('fn_i');
xlabel('n');

subplot(2,2,3);
stem(n,fn_m);
title('Magnitude Part');
ylabel('fn_m');
xlabel('n');

subplot(2,2,4);
stem(n,fn_p);
title('Phase Part');
ylabel('fn_p');
xlabel('n');

print -depsc pr11b;
```



Απάντηση ερωτήσεων:

Ερώτημα 1:

Αρχικά το πρόγραμμα δημιουργεί έναν μονοδιάστατο πίνακα n όπου τοποθετεί τις διακριτές της μεταβλητής n από το $-\mathbf{20}$ έως το $+\mathbf{20}$ με βήμα $\mathbf{1}$. Στη συνέχεια υπολογίζεται για κάθε τιμή του $\boldsymbol{n}$ η συνάρτηση $\boldsymbol{f(n) = 3 * e^{-0.2+0.5*i} * n}$

Και οι τιμές της ταξινομούνται μέσα στο πίνακα $\boldsymbol{fn}$. Προχωρώντας , υπολογίζονται οι πραγματική τιμή της συνάρτησης, οι μιγαδική της, η απόλυτη τιμή της, και η φάση της, και τοποθετούνται τα αποτελέσματα σε αντίστοιχους πίνακες. Δημιουργείται, όπως και στο προηγούμενο πρόγραμμα ένα παράθυρο $\boldsymbol{Figure}$ το οποίο τώρα χωρίζεται στα 4 και στο καθένα από αυτά αναπαρίστανται αυτές οι τέσσερεις ποσότητες που υπολογίστηκαν στο προηγούμενο βήμα με $\boldsymbol{plotting}$ τύπου s$\boldsymbol{tem}$. Η εντολή $\boldsymbol{title}$ ονοματίζει την γραφική παράσταση ενώ οι $\boldsymbol{ylabel}$ και $\boldsymbol{xlabel}$ δίνουν τίτλο στον $\boldsymbol{y}$ και $\boldsymbol{x}$ άξονα αντίστοιχα. Καθώς το πρόγραμμα εκτελείται δημιουργείται ένα αρχείο και τυπώνει τα δεδομένα εκεί μέσα με τύπο $\boldsymbol{.eps}$ και ονομασία $\boldsymbol{pr11b.eps}$

Ερώτημα 2:

Δεν εμφανίζεται καμία διαφορά αν αλλάξουμε το συμβολισμό του $\boldsymbol{i}$ με το $\boldsymbol{j}$.

Ερώτημα 3:

Σύμφωνα με την αρίθμηση του εργαστηριακού οδηγού στις γραμμές 10-11 υπολογίζουμε τη συνάρτηση πραγματικών τιμών της $\boldsymbol{f}(\boldsymbol{n})$ και τη συνάρτηση φανταστικών αριθμών της $\boldsymbol{f}(\boldsymbol{n})$ αντίστοιχα. Στις γραμμές 13-14 υπολογίζουμε το μέτρο της συνάρτησης $\boldsymbol{f}(\boldsymbol{n})$ ή αλλιώς απόκριση πλάτους και τη φάση ή γωνία φάσης της $\boldsymbol{f}(\boldsymbol{n})$ αντίστοιχα.

```
10 fn_r = real(fn);
11 fn_i = imag(fn);
12
13 fn_m = abs(fn);
14 fn_p = angle(fn);
```

Ερώτημα 4:

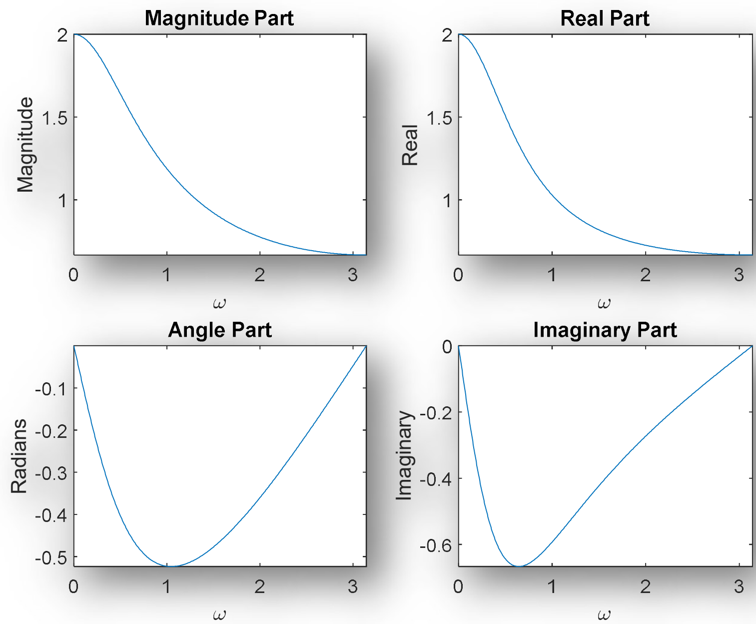
Θα μπορούσαμε να αντικαταστήσουμε αυτή τη γραμμή του κώδικα με τις παρακάτω :

```
for k = 1:1:41;
    n(k) = k-21;
end
```

- ***Άσκηση 3η :***

Ακολουθεί το πρόγραμμα: $\boldsymbol{Fourier1.m}$ στην λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης:

```
%%First Fourier practice program

clear all;
close all;
clc;

w = [0:1:500]*pi/500;
X = exp(i*w)./(exp(i*w)-0.5*ones(1,501));

magX = abs(X); angX = angle(X);
realX = real(X); imagX = imag(X);

figure(1);

subplot(2,2,1);
plot(w,magX);
xlabel('\omega');
ylabel('Magnitude');
title('Magnitude Part');
axis([0 pi min(magX) max(magX)]);

subplot(2,2,2);
plot(w,realX);
xlabel('\omega');
ylabel('Real');
title('Real Part');
axis([0 pi min(realX) max(realX)]);

subplot(2,2,3);
plot(w,angX);
xlabel('\omega');
ylabel('Radians');
title('Angle Part');
axis([0 pi min(angX) max(angX)]);

subplot(2,2,4);
plot(w,imagX);
xlabel('\omega');
ylabel('Imaginary');
title('Imaginary Part');
axis([0 pi min(imagX) max(imagX)]);

print -depsc progr23;
```



Απάντηση ερωτήσεων:

Ερώτημα 1:

Αρχικά το πρόγραμμα καθώς εκκινεί τοποθετεί στο πίνακα $\boldsymbol{w}$ τις τιμές που θα έχει το $\boldsymbol{\omega}$ οι οποίες κυμαίνονται από τη τιμή $\mathbf{0}{-}{>}\ \boldsymbol{\pi}$ (501 στο σύνολό τους και ισαπέχουν). Έπειτα, υπολογίζεται ο μετασχηματισμός $\boldsymbol{Fourier}$ διακριτού χρόνου της συνάρτησης $\boldsymbol{x}(\boldsymbol{n}) = \mathbf{0.5}^{\boldsymbol{n}}\ \boldsymbol{u}(\boldsymbol{n})$ (όπως περιγράφεται και στον εργαστηριακό οδηγό) για όλες τις τιμές του ω που έχουν άστατο πίνακα $\boldsymbol{Q}$ και με τη βοήθεια αυτού, υπολογίζονται οι πραγματικές τιμές της συνάρτησης , οι μιγαδικές, οι απόλυτες και οι φάσεις της κάθε τιμής. Εκτυπώνουμε όπως και προηγουμένως τις αντίστοιχες γραφικές με τη μορφή συνεχούς παράστασης $\boldsymbol{plot}$. Η εντολή axis οριοθετεί τα όρια του x και του y άξονα που θα εμφανίζονται στις γραφικές μας παραστάσεις. Καθώς το πρόγραμμα εκτελείται δημιουργείται ένα αρχείο και τυπώνει τα δεδομένα εκεί μέσα με τύπο $\boldsymbol{.eps}$ και ονομασία $\boldsymbol{progr23}$.

Ερώτημα 2:

Ο πίνακας $\boldsymbol{w}$ περιέχει 501 στοιχεία τα οποία ισαπέχουν μεταξύ τους και κυμαίνονται από το $\mathbf{0}{-}{>}\ \boldsymbol{\pi}$. Ο λόγος που είναι 501 τα στοιχεία του πίνακα είναι επειδή έχουμε πάρει τιμές από το 0 έως και το 500 με βήμα 1.

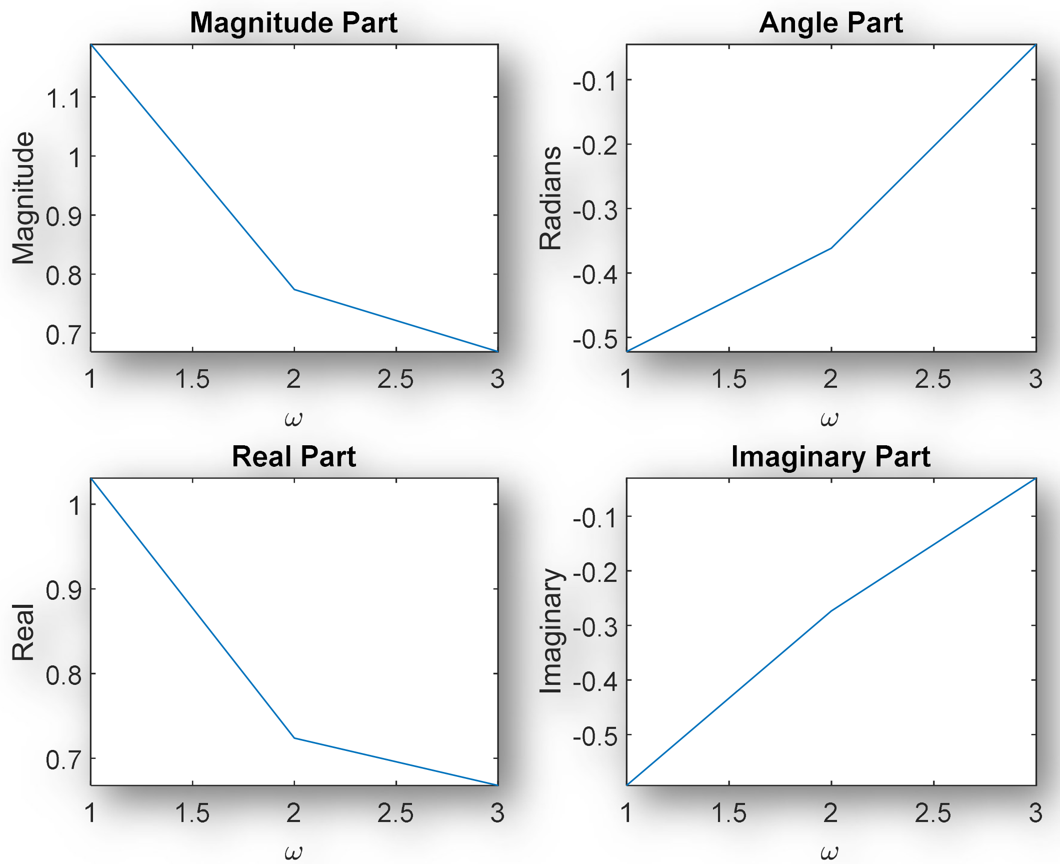
Ερώτημα 3:

Η εντολή $\boldsymbol{ones}(\,)$ επιστρέφει πίνακα μοναδιαίων στοιχείων. Για παράδειγμα εάν γράφουμε $\boldsymbol{ones}(\boldsymbol{N})$ αυτή η εντολή θα επιστρέψει μοναδιαίο πίνακα $\boldsymbol{NxN}$, ενώ αν γράψουμε $\boldsymbol{ones}(\boldsymbol{M},\boldsymbol{N})$ θα επιστρέψει μοναδιαίο $\boldsymbol{MxN}$.

- ***Άσκηση 4η :***

Ακολουθεί το πρόγραμμα: $\boldsymbol{Fourier2.m}$ στην λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης:

```
%%Second Fourier practice program

clear all;
close all;
clc;

N=8;
w = [1:1:3];
a0 = 0;
X=a0;

for(n=0:1:N)
    X = a0+((0.5)*exp(-i*w)).^n;
    a0 = X;
end

magX = abs(X); angX = angle(X);
realX = real(X); imagX = imag(X);

figure(1);

subplot(2,2,1);
plot(w,magX);
xlabel('\omega');
ylabel('Magnitude');
title('Magnitude Part');
axis([1 3 min(magX) max(magX)]);

subplot(2,2,2);
plot(w,angX);
xlabel('\omega');
ylabel('Radians');
title('Angle Part');
axis([1 3 min(angX) max(angX)]);

subplot(2,2,3);
plot(w,realX);
xlabel('\omega');
ylabel('Real');
title('Real Part');
axis([1 3 min(realX) max(realX)]);

subplot(2,2,4);
plot(w,imagX);
xlabel('\omega');
ylabel('Imaginary');
title('Imaginary Part');
axis([1 3 min(imagX) max(imagX)]);

print -depsc prog2;
```



Απάντηση ερωτήσεων:

Ερώτημα 1:

Οι τιμές που θα λαμβάνει το $\boldsymbol{N}$ θα είναι ακαριαίες και αρκετά μεγάλες για να υπάρχει ακρίβεια στο τελικό αποτέλεσμα. Γιαν να υπολογίσουμε αυτά τα $\boldsymbol{N}$ ελέγχουμε το κριτήριο σύγκλισης του μετασχηματισμού όπου σύμφωνα με τον εργαστηριακό οδηγό είναι $\boldsymbol{0.5^N = e^{-5}}$. Λύνοντας τη σχέση ως προς $\boldsymbol{N}$ προκύπτει:

$$\boldsymbol{N = 7.2134}$$

Συνεπώς μεγάλη τιμή έτσι ώστε να υπάρχει ακρίβεια μπορεί να θεωρηθεί οποιαδήποτε τιμή μεγαλύτερη του 8, όμως για να εξοικονομήσουμε επεξεργαστική ισχύ επιλέγουμε $\boldsymbol{N = 8}$.

Ερώτημα 2:

Όπως δείξαμε και στο προηγούμενο ερώτημα, μπορούμε να σταματήσουμε τη διαδικασία σε συγκεκριμένο $\boldsymbol{N}$ γιατί ικανοποιείται το κριτήριο σύγκλισης και ο υπολογισμός μας έχει μεγάλη ακρίβεια.

***Ερώτημα 3**:*

Μπορούμε να παρατηρήσουμε άμεσα τον αριθμό υπολογισμών που απαιτείται σε αυτό το πρόγραμμα και στο προηγούμενο, καθώς σε αυτό με 8 υπολογισμούς μπορούμε να πετύχουμε επαρκή ακρίβεια, ενώ ο προηγούμενος κώδικας κάνει 501 υπολογισμούς για τις τιμές του $\boldsymbol{Q}$.

***Ερώτημα 4**:*

Πρέπει να εξεταστεί εάν η σειρά του μετασχηματισμού συγκλίνει δηλαδή εάν εκπληρώνεται η συνθήκη $\sum_{n=-\infty}^{\infty}|x(n)| < \infty$. Πραγματοποιώντας του υπολογισμούς μας έχουμε:

$$\sum_{n=-\infty}^{\infty}|x(n)| = \sum_{n=-\infty}^{\infty} 0.1^{n} * u(n) = \sum_{n=0}^{\infty} 0.1^{n} = \frac{1}{1-0.1} < \infty$$

Αφού $0.1 < 1,\ n \to \infty \ => \ 0.1^{n} \to 0$

Συνεπώς μπορεί να πραγματοποιηθεί ο μετασχηματισμός Fourier διακριτού χρόνου.

***Ερώτημα 5**:*

Πρέπει να εξεταστεί εάν η σειρά του μετασχηματισμού συγκλίνει δηλαδή εάν εκπληρώνεται η συνθήκη $\sum_{n=-\infty}^{\infty}|x(n)| < \infty$. Πραγματοποιώντας του υπολογισμούς μας έχουμε:

$$\sum_{n=-\infty}^{\infty}|x(n)| = \sum_{n=-\infty}^{\infty} 1.1^{n} * u(n) = \sum_{n=0}^{\infty} 1.1^{n} \to \infty$$

Αφού $1.1 > 1,\ n \to \infty \ => \ 1.1^{n} \to \infty$

Συνεπώς δεν μπορεί να πραγματοποιηθεί ο μετασχηματισμός Fourier διακριτού χρόνου.

***Ερώτημα 6**:*

Πρέπει να εξεταστεί εάν η σειρά του μετασχηματισμού συγκλίνει δηλαδή εάν εκπληρώνεται η συνθήκη $\sum_{n=-\infty}^{\infty}|x(n)| < \infty$. Πραγματοποιώντας του υπολογισμούς μας έχουμε:

$$\sum_{n=-\infty}^{\infty}|x(n)| = \sum_{n=-\infty}^{\infty} 1^{n} * u(n) = \sum_{n=0}^{\infty} 1^{n} \to ?$$

Αφού $n \to \infty \ => \ 1^{n} \to 1$

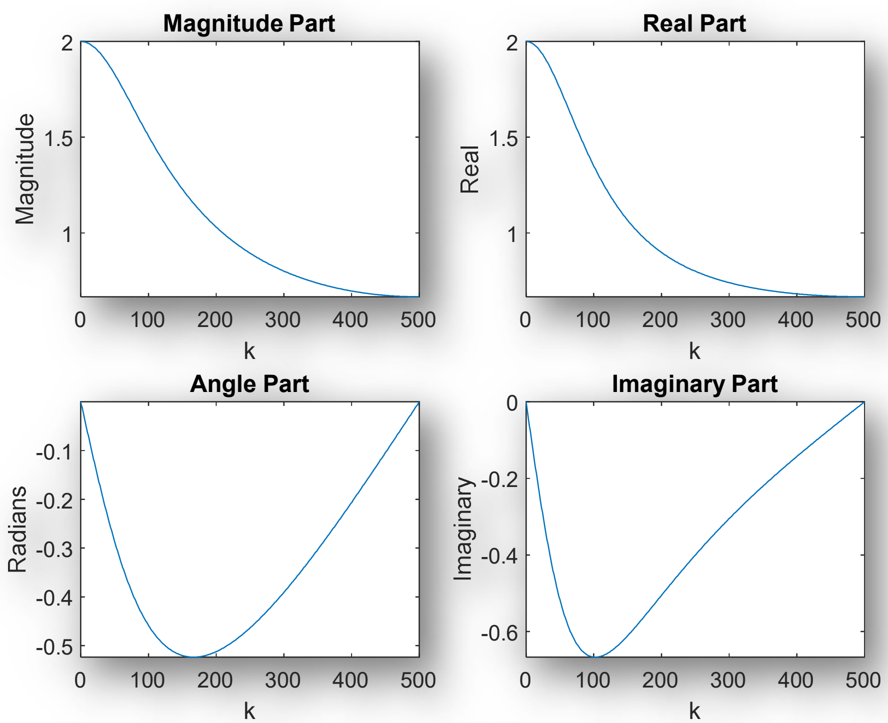
Συνεπώς δεν μπορούμε να αποφανθούμε.

➢ ***Άσκηση 5η :***

Ακολουθεί το πρόγραμμα: ***Fourier3.m*** στην λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης:

```
%%Third Fourier exercise

clear all;
close all;
clc;

M = 500;
n1 = 0;
n2 = 1000;
n0 = 0;
k = [0:1:M];
[u_step,n] = step_f(n0,n1,n2);

x = (0.5.^n).*u_step;
nT=n.';
arg1 = -i*pi/M;
X = x*((exp(arg1)).^(nT*k));

magX = abs(X); angX = angle(X);
realX = real(X); imagX = imag(X);

figure(1);

subplot(2,2,1);
plot(k,magX);
xlabel('k');
ylabel('Magnitude');
title('Magnitude Part');
axis([0 M min(magX) max(magX)]);

subplot(2,2,3);
plot(k,angX);
xlabel('k');
ylabel('Radians');
title('Angle Part');
axis([0 M min(angX) max(angX)]);

subplot(2,2,2);
plot(k,realX);
xlabel('k');
ylabel('Real');
title('Real Part');
axis([0 M min(realX) max(realX)]);

subplot(2,2,4);
plot(k,imagX);
xlabel('k');
ylabel('Imaginary');
title('Imaginary Part');
axis([0 M min(imagX) max(imagX)]);

print -depsc prog24;
```

```
%%Step function
function [u,n] = step_f(n0,n1,n2);
n = [n1:1:n2];
u = [(n-n0)>=0];
```



Απάντηση ερωτήσεων:

Ερώτημα 1:

Το πρόγραμμα καθώς εκκινεί αρχικοποιεί τις μεταβλητές $\boldsymbol{M = 500, n1 = 0\ , n2 = 1000\ , n0 = 0}$ και δημιουργεί ένα μονοδιάστατο πίνακα k στον οποίο βρίσκονται τιμές από το $\mathbf{0}$ έως και το $\mathbf{500}$ με βήμα $\mathbf{1}$. Στη συνέχεια αποθηκεύει στο πίνακα $\boldsymbol{u_{step}}$ τα $\mathbf{1001}$ στο πλήθος στοιχεία της βηματικής συνάρτησης $\boldsymbol{u(n)}$. Για κάθε στοιχείο του $\boldsymbol{u_{step}}$ και για κάθε στοιχείο του n υπολογίζεται η σχέση $\boldsymbol{x = (0.5^n) * u_{step}}$ και τα στοιχεία της καταχωρούνται στο πίνακα x. Έπειτα, μετατρέπεται ο πίνακας n σε πίνακα στήλης και τον τοποθετεί μέσα στο $\boldsymbol{nT}$. Ορίζεται μία νέα μεταβλητή $\boldsymbol{arg1}$ η οποία αρχικοποιείται με τη τιμή $\boldsymbol{-i * \pi/M}$ οποία θα αποτελέσει τον συντελεστή μέσα στο εκθετικό του επόμενου υπολογισμού. Υπολογίζεται ο μετασχηματισμός $\boldsymbol{Fourier}$, $\boldsymbol{X = x * \left((e^{arg1})^{nT*k}\right)}$ όπου όλες οι τιμές που προκύπτουν καταχωρούνται στο πίνακα Q. Υπολογίζονται όπως ακριβώς γίνεται και στα προηγούμενα προγράμματα οι πραγματικές τιμές της συνάρτησης, ,οι μιγαδικές, η απόλυτες τιμές και οι φάσεις της, μετά τυπώνονται όλες αντίστοιχα σε ένα παράθυρο που δημιουργείται μέσω της $\boldsymbol{Matlab}$. Καθώς το πρόγραμμα εκτελείται δημιουργείται ένα αρχείο και τυπώνει τα δεδομένα εκεί μέσα με τύπο $\boldsymbol{.eps}$ και ονομασία $\boldsymbol{prog24}$.

Ερώτημα 2:

Δεν θα είχε ιδιαίτερο νόημα να θέσουμε $\boldsymbol{n1 = -1000}$ *και* $\boldsymbol{n2 = 1000}$, *γιατί εάν παρατηρήσουμε μέσα στη συνάρτηση* $\boldsymbol{step\_f.m}$ *και πιο συγκεκριμένα στο πίνακα* $\boldsymbol{u}$ *αποθηκεύονται μόνο θετικές τιμές του* $\boldsymbol{n}$ *ως τη τιμή* $\mathbf{1}$, *ενώ οι αρνητικές παίρνουν τη τιμή* $\mathbf{0}$. *Συνεπώς όταν θα πραγματοποιηθεί η πράξη* $\boldsymbol{x = 0.5^n * u_{step}}$ *τα σημεία από το* $\mathbf{-1000}$ *έως το* $\mathbf{-1}$ *θα έχουν επιστρέψει* $\mathbf{0}$ *και δεν θα επηρεάσουν το πίνακα x πέρα του γεγονότος ότι θα τον μεγαλώσουν κατά* $\mathbf{1000}$ *θέσεις.*

Ερώτημα 3:

Θα έπρεπε να θέσουμε τα $\boldsymbol{n1 = -1000}$ *και* $\boldsymbol{n2 = 0}$ *ενώ ακόμη να αλλάξουμε τη συνθήκη μέσα στη* $\boldsymbol{step\_f.m}$ *και πιο συγκεκριμένα θα είχε τη μορφή:*

```
%%Step function
function [u,n] = step_f(n0,n1,n2);
n = [n1:1:n2];
u = [(n-n0)<=0];
```

Σκοπός είναι να επιστρέφονται οι τιμές $\mathbf{1}$ *για τα αρνητικά* $\boldsymbol{n}$ *και* $\mathbf{0}$ *για τα θετικά.*

Ερώτημα 4:

Θα μπορούσαμε να αλλάξουμε απλά τη σχέση :

```
απο
x = (0.5.^n).*u_step;
σε
x = (1.1.^n).*u_step;
```

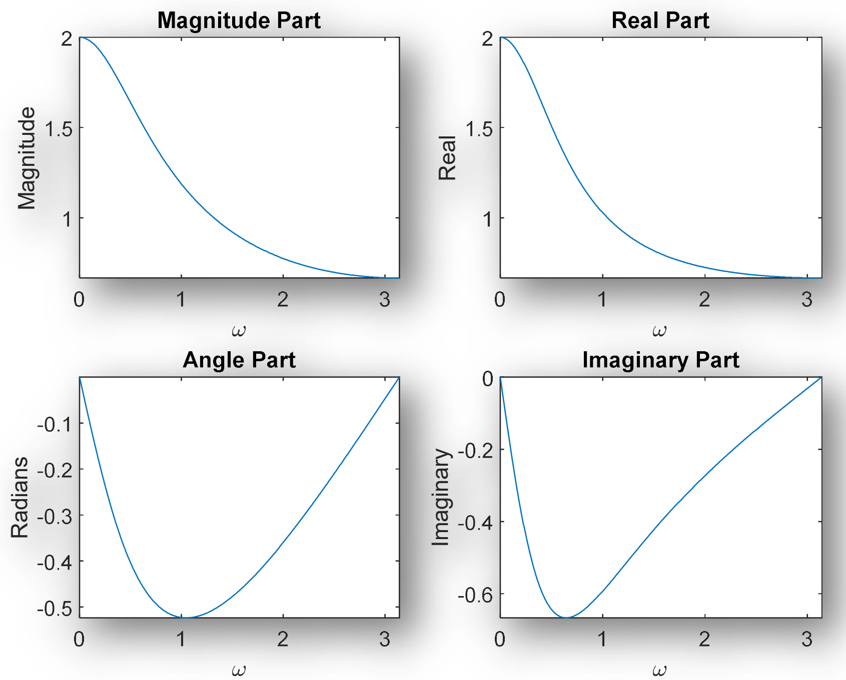
Και να υπολογιστεί σωστά ο μετασχηματισμός μας καθώς με αυτό το τρόπο επίλυσης το άθροισμα:

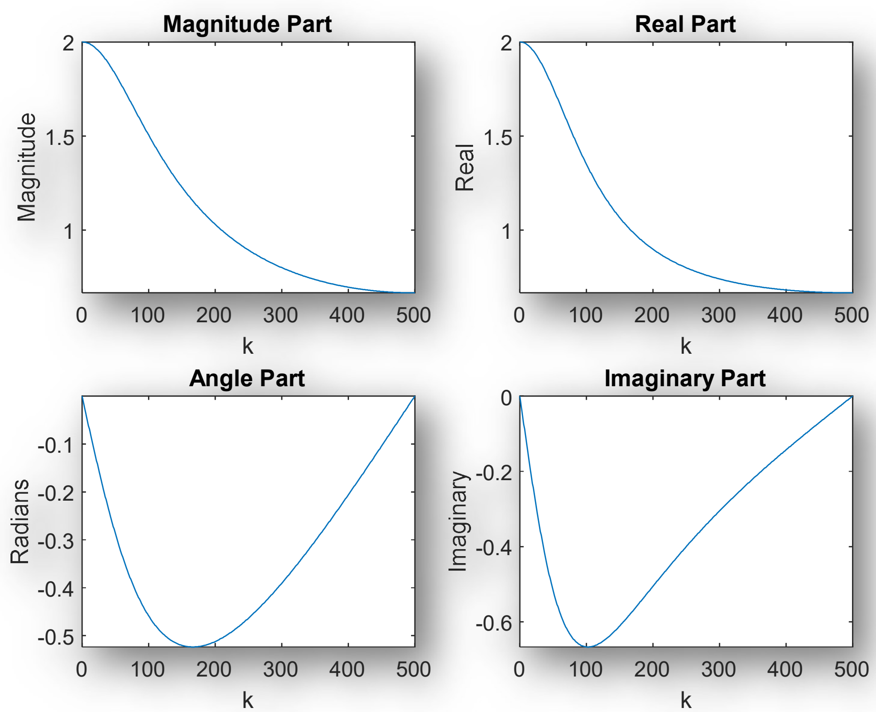
$$\sum_{n=n1}^{n2} 1.1^{\mathrm{n}} * u(n) < \infty$$

και άρα δεν υπάρχει κανένα πρόβλημα σύγκλισης.

***Ερώτημα 5**:*

Οι δύο αντίστοιχες γραφικές:





Παρατηρούμε εύκολα πως υπάρχει ταύτιση στα αποτελέσματά μας. Μία προφανή διαφορά όμως είναι ότι η μία έχει στον άξονα $\boldsymbol{x}$ *τις τιμές του* $\boldsymbol{k}$ *και η άλλη από το* $\boldsymbol{\omega}$ *αλλά υπάρχει σχέση μεταξύ τους. Όμως επειδή :*

$$X(k) = X(\omega/(\pi/500)\,)$$

Και :

$$X(\omega) = X(k * \pi/500\,)$$

Ο μετασχηματισμός Fourier πραγματοποιείται χωρίς κανένα σφάλμα και επιστρέφει τα ίδια αποτελέσματα.