

2.Soru Açıklama

Burada xn katsayılarını buldurmak için find_xn fonksiyonu kullanıldı.

find_xn(func, Order, Period) fonksiyonu

```
function xn = find_xn(func,Order,Period)
j = sqrt(-1);
w0 = 2*pi/Period;
xn = [-Order:Order];
counter = 1;
for n=[-Order:Order]
    fx = @(x) (func(x)).*exp(-j*n*w0*x);
    xn(counter) = (1/Period)*integral(fx,0,1);
    counter = counter + 1;
end
end
```

Bu fonksiyondaki func bir fonksiyondur. Örnek olarak:

$f = @(x) x^2 + x$

gibi.

Order kaçınıcı dereceden fourier serisine açılacağını belirten değer. Period ise isminden de belli olduğu üzere Perioddur.

$$x[n] = \frac{1}{T} \int_T x(t) e^{-j n w_0 t}, w_0 = 2\pi/T$$

formülü bu fonksiyonda yapılmıştır. Integral için fonksiyon belli olduğundan (sadece veriden ibaret olmadığından) MatLab'ın build in fonksiyonu olan integral kullanılmıştır. İntegral'de ilk argüman fonksiyon ikinci argüman integral başlangıç değeri üçüncü argüman ise integral son değeridir. Fonksiyonumuz 0 dan 1 e tanımlı olduğundan period 1 kabul edilmiş ve fonksiyonumuzun 0 dan 1 e integrali alınmıştır. fx ile

$$x(t) e^{-j n w_0 t}$$

değeri sağlanmıştır. Bu fonksiyonun integrali alındıktan sonra Perioda bölünmüştür ve ilgili n değerine göre xn vektörüne atanmıştır.

find_xt(xn,Order,t,Period)

```
function xt = find_xt(xn,Order,t,Period)
j = sqrt(-1);
w0 = 2*pi/Period;
xt = 0;
counter = 1;
for n=[-Order:Order]
    xt = xt + xn(counter)*exp(j*w0*n*t);
    counter = counter + 1;
end
end
```

Bu fonksiyon x[n] lerden verilen Order değerine göre x(t) yi çiziyor.

$$x(t) = \sum_{n=-\text{Order}}^{n=\text{Order}} x[n]e^{jn\omega_0 t}, \quad \omega_0 = 2\pi/T$$

Formülünü kullanmaktadır. For döngüsü içinde gerekli toplama işlemi yapılmaktadır.

GUI Kısmı:

GUI kısmı guide kullanılarak yapıldı.

GUI kısmında çizimleri yaptırmak için 2 fonksiyon kullanıldı.

draw_xn_axis(handles) fonksiyonu

```
function draw_xn_axis(handles)
    global T;
    global func1;
    global order;
    global xn;
    xn = find_xn(func1,order,T);

    xn_ax = [-(length(xn)-1)/2 : (length(xn)-1)/2];

    axes(handles.axes3);
    stem(xn_ax,abs(xn),'filled','Color','black');
    xlim([xn_ax(1)-1 xn_ax(end)+1]);
    title('x[n] Magnitude');
```

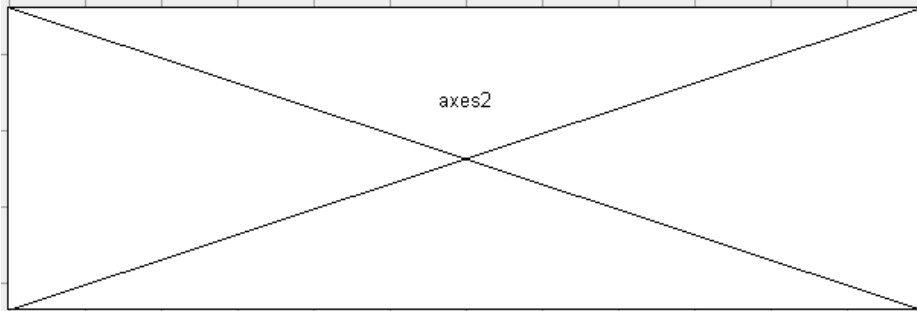
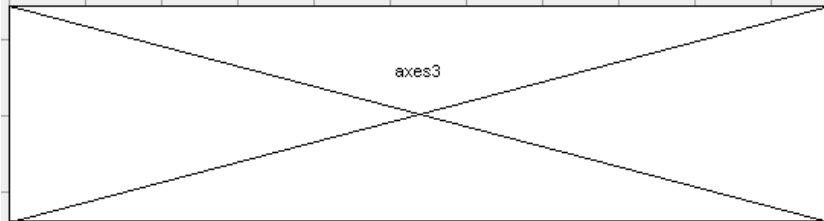
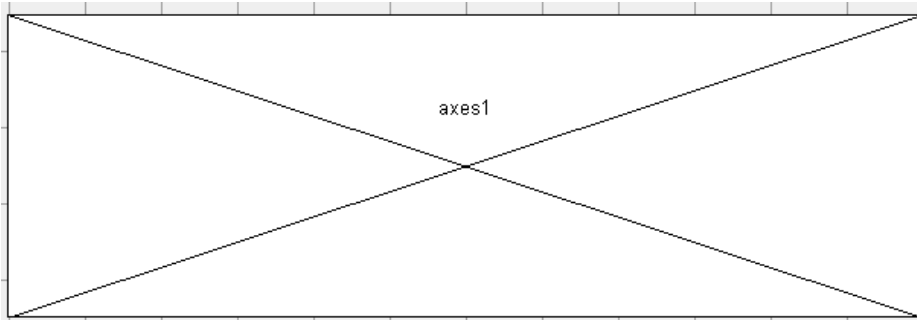
Bu fonksiyonda find_xn fonksiyonu kullanılarak x[n] değerleri order'a göre elde edildi. xn vektörüne göre xn_ax olan xn vektörünün eksenini elde edildi. Eksen 0 ortada olmak üzere iki tarafa simetrik olacak şekilde elde edildi. Daha sonra x[n] değerlerinin **Büyüklik** değerleri çizildi.

draw_axis(handles) fonksiyonu

```
function draw_axis(handles)
global t;
global y;
global T;
global funcl;
global order;
global xn;
global xt;
global dsply_original;
xn = find_xn(funcl,order,T);
xt = find_xt(xn,order,t,T);

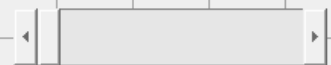
axes(handles.axes2);
if dsply_original
    cla;
    plot(t,y);
    hold on;
    plot(t,real(xt));
    legend('Original',string(order)+' Approximation');
else
    cla;
    plot(t,real(xt));
    legend(string(order)+' Approximation');
end
title(string(order) + ' Approximation');
```

Bu fonksiyonda find_xn fonksiyonu kullanılarak xn vektörü hesaplandı ve bu vektör kullanılarak xt vektörü hesaplandı. Daha sonra ise dsply_original fonksiyonun içindeki değer 1 olması durumunda x(t) fonksiyonu orijinal fonksiyonla beraber çizildi. dsply_original fonksiyonunun içindeki değer 0 olması durumunda ise sadece x(t) fonksiyonu çizildi.



2nd Order

9th Order



Order =

☐ Plot with Original

```

function homewrok2_2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to homewrok2_2 (see VARARGIN)

global func1;
global increment;
global t;
global y;
global T;
global dsply_original;
dsply_original = 0;
T = 1;
func1 = @(x) -(x.^3)/3 + (x.^2)/2 -3*x/16 + 1;

increment = 0.0001;
t = [0:increment:1];
y = func1(t);
axes(handles.axes1);
plot(t,y);
title('Original Function');

```

Program başlangıcında soruda belirtilen fonksiyon global bir değer olan func1 olarak atandı. Bu fonksiyonu verilerle elde etmek için t eksenini increment değerine göre belirlendi ve y değerine func1 fonksiyonu ve t verisi kullanılarak veriler yazıldı. Orijinal fonksiyon ilk eksene çizildi. dsply_original değeri ilk olarak 0 verildi. Daha checkbox'dan alınan verilere göre değişecektir.

```

% --- Executes on button press in second_order_button.
function second_order_button_Callback(hObject, eventdata, handles)
% hObject    handle to second_order_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
global order;
order = 2;
set(handles.order_slider,'Value',order);
set(handles.order_text,'String','Order = ' + string(order));
draw_xn_axis(handles)
draw_axis(handles)

```

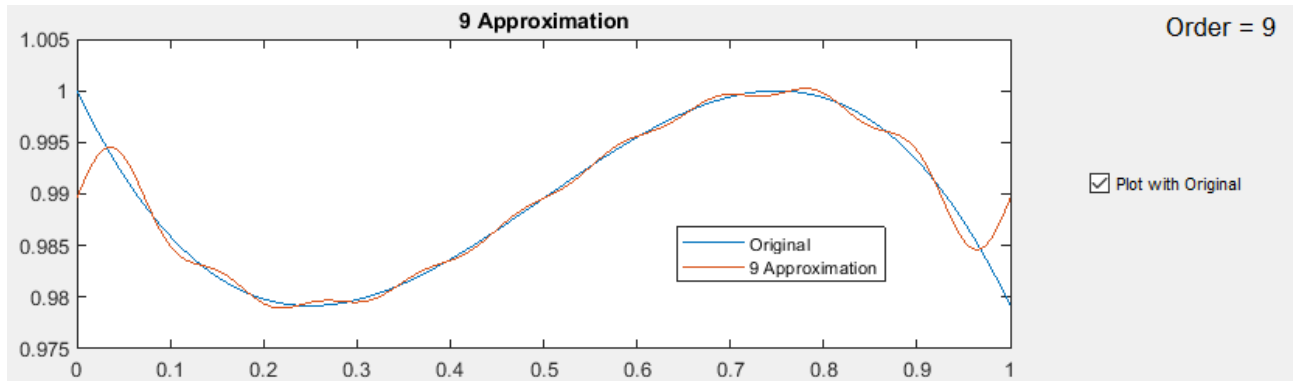
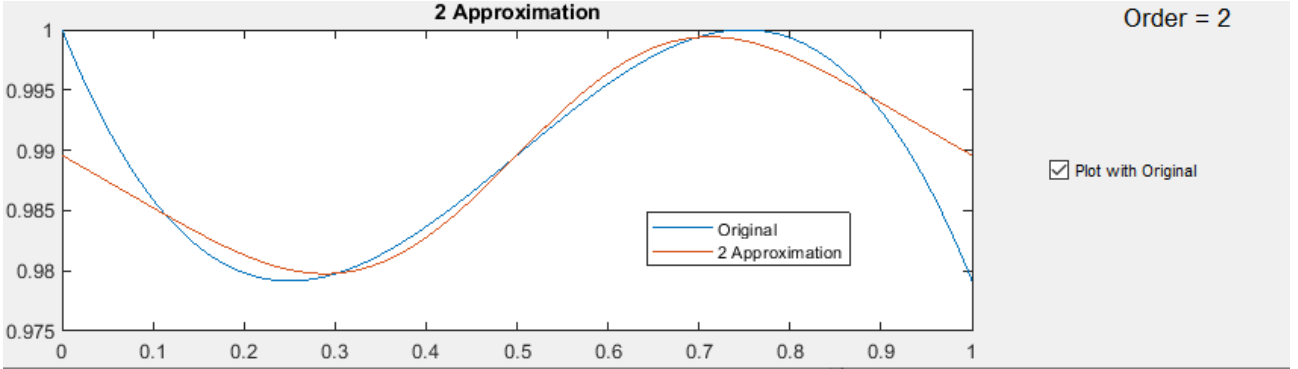
-> Bu butonda order 2 yapıldı ve slider değeri ve order'ı gösteren static text değeri buna göre güncellendi. Sonra draw_xn_axis fonksiyonu ile draw_axis fonksiyonları ile ilgili hesaplamalar yapıp eksenlere ilgili veriler çizdirildi.

-> 9th Order butonunda ise 2nd Order butonundaki işlemlerin aynısı yapıldı farklı olarak order'a 9 atandı.

-> Slider callback'inde order slider'ın deęerine gre integer olarak alındı ve integer olarak slider deęeri set edildi. Sonra draw_xn_axis ve draw_axis fonksiyonları ile ilgili hesaplamalar yapıp fonksiyon izdirildi.

-> Plot with Original checkbox'ının callback'inde global bir variable olan dsply_original deęerleri checkbox'ın durumuna gre gncellendi.

SONULARI DEęERLENDİRME



9. yaklaşımın 2. yaklaşıma gre daha iyi sonu verdięi ıktılardan anlaşılmaktadır. Fourier serisi daha fazla aıldığında orijinal fonksiyona yaklaşımda o kadar artmaktadır.

MUHAMMED HALİT TOKLUOęLU - 161110070