



2020-2021 BAHAR DÖNEMİ

**YMH214
SAYISAL ANALİZ
LAB. DERSİ**

5.DERS

Arş. Gör. Alev KAYA

**02.04.2021
SAAT:16:00-17:00**



Lineer Olmayan Denklem Sistemlerimin Çözümü

- Açık Yöntemler:
- **A**-Newton Raphson Yöntemi
- **B**-Sekant Yöntemi
- **LAB**: Newton Raphson yöntemi ve Sekant yöntemi
Matlab örnek programı

Lineer Olmayan Denklemler Sisteminin Çözümü

Açık Yöntemler

NEWTON - RHAPSON YÖNTEMİ (NEWTON-RHAPSON METHOD)

Kök bulma algoritmalarından en ünlüsü Newton-Rhapson yöntemidir. Çözümüne çok hızlı yakınsayan basit bir programdır. Bu yöntem hem fonksiyonun kendisini hemde türevini içerir. Yöntem aşağıdaki iterasyon ile sonraki noktayı bulmayı hedefler:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Yöntem için adımlar aşağıdaki gibidir:

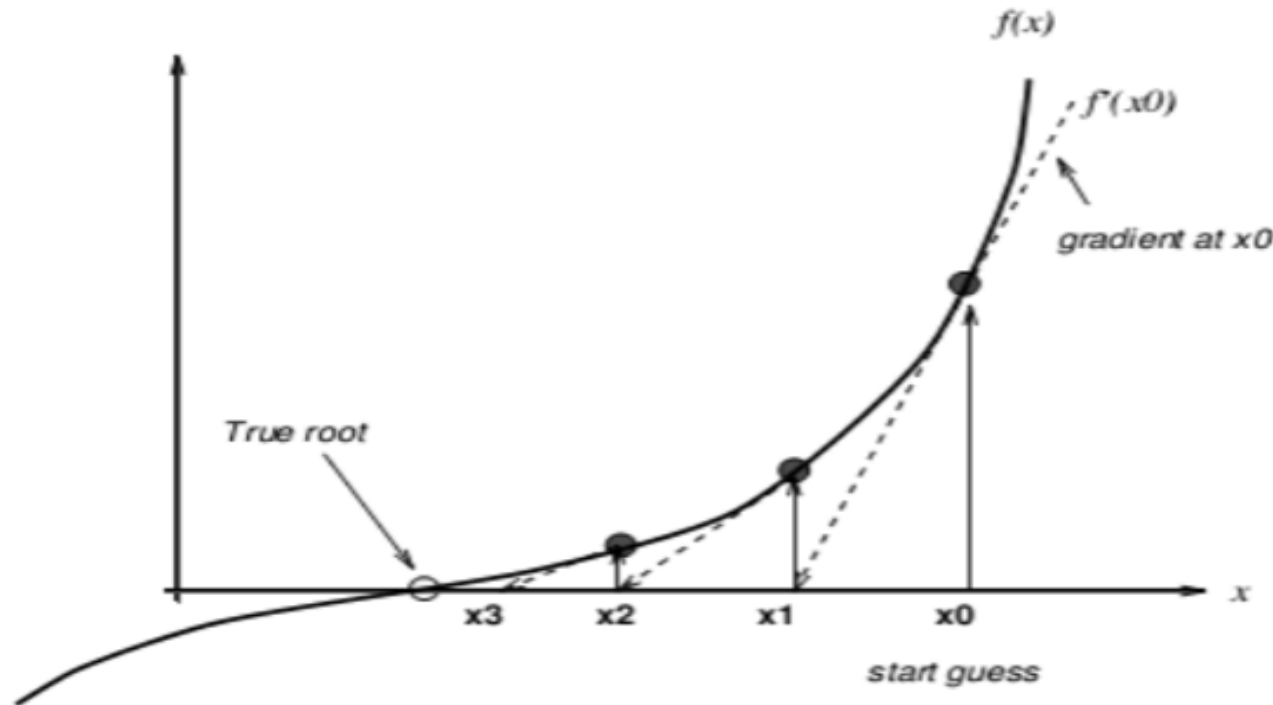
1.Adım: $i=1$ olarak belirle. Eğer $f'(x_i) = 0$ ise programı durdur ve hata mesajı ver.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

2.Adım:

3.Adım: Eğer $f(x_{i+1}) < \text{eps}$ ise x_{i+1} çözümdür ve programdan çık

4.Adım $i \rightarrow i+1$ olarak belirle ve 2.Adıma dön



Örnek: $\tan(\pi-x)-x=0$ fonksiyonunun (1.6,3) aralığındaki kökünü Newton-Rhapon yöntemi ile MATLAB programında yazınız.

MATLAB R2020a

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Script New Live Script New Open Find Files Compare Import Data Save Workspace New Variable Open Variable Clear Workspace Favorites Run and Time Clear Commands Simulink Layout Preferences Set Path Parallel Add-Ons Help Community Request Support Learn MATLAB

Search Documentation Sign In

Current Folder: C:\Matlab_Dersler

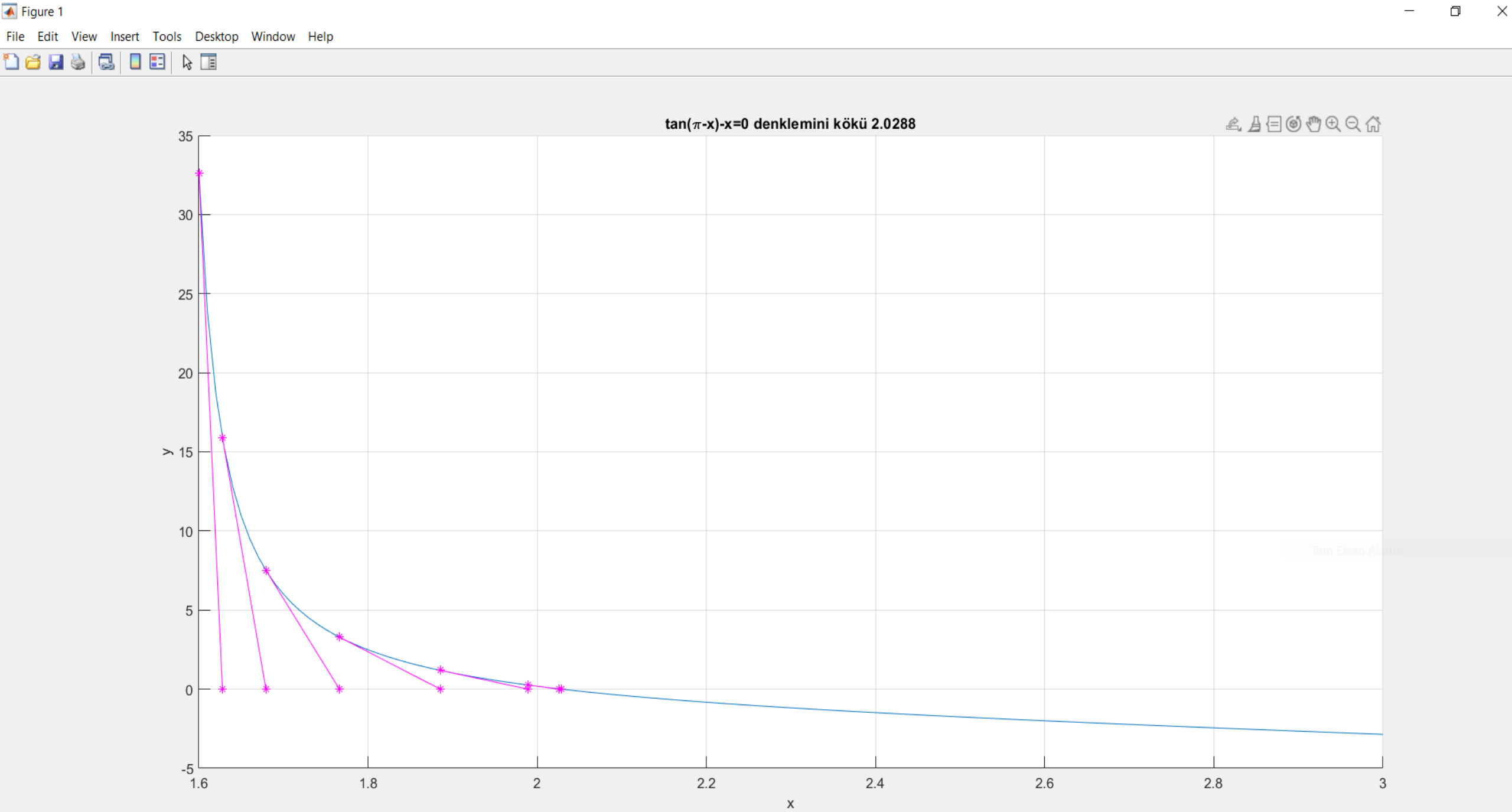
Editor - C:\Matlab_Dersler\newton_rhapson.m

```
1 function newton_rhapson %newton_rhapson method
2 clc;clear all;
3 a=1.6;b=3;
4 xx=[a:0.01:b];
5 y=f(xx);
6 it=1;
7 x(it)=a;
8 while abs(f(x(it)))>eps
9     if abs(df(x(it)))<eps
10         disp('Program türev sıfır durumunda hesaplanamaz!')
11         break;
12     else
13         line([x(it) x(it)-f(x(it))/df(x(it))],[f(x(it)),0],[1 1],'Marker','*','LineStyle','-','Color','m');
14         hold on;
15         x(it+1)=x(it)-f(x(it))/df(x(it));
16         it=it+1;
17     end
18 end
19 plot(xx,y);
20 xlabel('x');
21 ylabel('y');
22 title(['tan(\pi-x)-x=0 denklemini kökü ',num2str(x(it))])
23 grid on
24 fprintf('tan(pi-x)-x=0 denkleminin kökü %6.4f dir ve iterasyon %6.4f dir',x(it),it)
25
26 function y=f(x)
27 y=tan(pi-x)-x;
28
29 function y=df(x)
30 y=-(1+(tan(pi-x)).^2)-1;
```

Workspace

Name	Value
Tam Ekran Alıntısı	

Command Window



Lineer Olmayan Denklem Sistemlerimin Çözümü

Açık Yöntemler

KİRİŞLER YÖNTEMİ (SECANT METHOD)

Kirişler yöntemi Newton-Rhapson yöntemindeki türevin sonlu fark yaklaşımı yazılması ile elde edilir. Newton rhapson yönteminde türev yerine

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

yazılarak iterasyonu

$$x_{k+1} \leftarrow x_k - \frac{f(x_k)}{\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}} \quad \rightarrow \quad x_{k+1} \leftarrow x_k - \frac{(x_k - x_{k-1})f(x_k)}{f(x_k) - f(x_{k-1})}$$


$$x_{k+1} \leftarrow \frac{x_{k-1}f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})}$$

Olarak yazarız.

Yöntem için adımlar aşağıdaki gibidir:

Algoritma

1.Adım: $i=1$ olarak belirle.


$$x_{k+1} \leftarrow \frac{x_{k-1}f(x_k) - x_kf(x_{k-1})}{f(x_k) - f(x_{k-1})}$$

2.Adım:

3.Adım: Eğer $f(x_{i+1}) < \text{eps}$ ise x_{i+1} çözümdür ve programdan çık

4.Adım $i \rightarrow i+1$ olarak belirle ve 2.Adıma dön

Örnek: $f(x)=x^3-3x+2$ fonksiyonunun $x_0=-2.6$, $x_1=-2.4$ vererek kökünü Kirişler yöntemi ile MATLAB programında yazınız.

MATLAB R2020a

HOME PLOTS APPS EDITOR PUBLISH VIEW

Search Documentation Sign In

FILE NAVIGATE EDIT BREAKPOINTS RUN

Current Folder: C:\Matlab_Dersler

Editor - C:\Matlab_Dersler\kirisler_yontemi.m

```
1 function kirisler_yontemi
2 clc;clear all;
3 it=1;
4 x(it)=-2.6;
5 it=it+1;
6 x(it)=-2.4;
7 xx=[x(1):0.01:-1.98];
8 y=f(xx);
9 while abs(f(x(it)))>0.001
10     x(it+1)=(x(it-1)*f(x(it))-x(it)*f(x(it-1)))/(f(x(it))-f(x(it-1)));
11     line([x(it-1) x(it+1)], [f(x(it-1)) f(x(it+1))], [1 1], 'Marker', '*', 'LineStyle', '-', 'Color', 'm');
12     hold on;
13     it=it+1;
14 end
15 plot(xx,y);
16 xlabel('x');
17 ylabel('y');
18 title(['x.^3-3*x+2=0 denklemini kökü ',num2str(x(it))])
19 grid on
20 fprintf('x.^3-3*x+2=0 denkleminin kökü %6.4f dir ve iterasyon %6.4f dir',x(it),it)
21
22 function y=f(x);
23 y=x.^3-3*x+2;
```

Workspace

Name	Value
------	-------

Tam Ekran Alıntısı

Command Window

