



2020-2021 BAHAR DÖNEMİ

**YMH214
SAYISAL ANALİZ
LAB. DERSİ**

4.DERS (GECE GRUBU)
Arş. Gör. Alev KAYA

**26.03.2021
SAAT:16:00-17:00**



Lineer Olmayan Denklem Sistemlerimin Çözümü

➤ Kapalı Yöntemler :

A-Regula Falsi yöntemi

➤ Açık Yöntemler :

A-Fixed point Iteration yöntemi

➤ **LAB:** Regula Falsi yöntemi ve Fixed Point yöntemi Matlab örnek programı

Lineer Olmayan Denklem Sistemlerinin Çözümü

➤ Kapalı Yöntemler :

A-Regula Falsi yöntemi (Yer Değiştirme Methodu)

- İkiye bölme yöntemi, köklerin bulunması için geçerli bir yöntem olmasına rağmen '**kaba kuvvet**' yaklaşımı oldukça verimsiz,
- İkiye bölme yönteminin eksiklerinden biri, $x_{üst}$ ile x_{alt} arasında kalan aralığı ikiye bölerken $f(x_{alt})$ ve $f(x_{üst})$ değerlerini göz önüne almamasıdır.
- **Örneğin;** $f(x_{alt})$ sıfıra $f(x_{üst})$ ' den daha yakınsa, kökün x_{alt} değerine , $x_{üst}$ değerinden daha yakın olma olasılığı mevcuttur.

Lineer Olmayan Denklem Sistemlerimin Çözümü

➤ Kapalı Yöntemler :

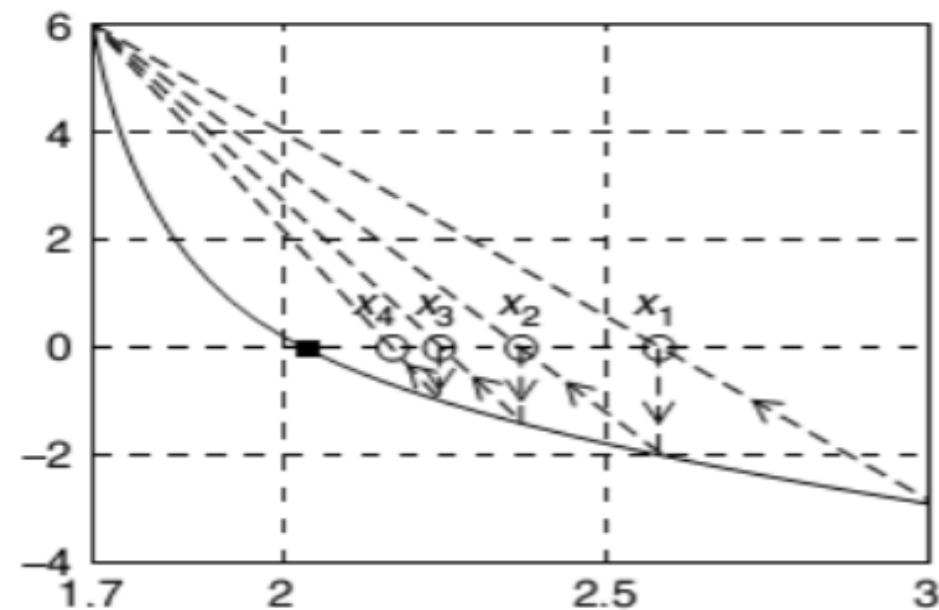
A-Regula Falsi yöntemi (Yer Değiştirme Methodu)

- Yer değiştirme formülü kullanılarak hesaplanan x_{tahmin} değeri daha sonra lineer olmayan fonksiyon $f(x_{tahmin})$ ile aynı işaretli yapan x_{alt} ve $x_{üst}$ değeri ile yer değiştirilir.
- Böylece x_{alt} ve $x_{üst}$ değerleri her zaman gerçek kökü kısıkaca alır.
- Bu yöntem, kapalı yöntemler arasında tercih edilen bir çözüm yaklaşımıdır.

$$➤ x_{tahmin} = x_{üst} - \frac{f(x_{üst})(x_{alt} - x_{üst})}{f(x_{alt}) - f(x_{üst})}$$

REGULA FALSİ YÖNTEMİ (REGULA FALSİ - FALSE POSITION METHOD)

Eğer $f(x)=0$ denkleminin (a,b) aralığında kökü olması için $f(a).f(b)<0$ koşulu sağlanması gerekmektedir.



Yöntem için adımlar aşağıdaki gibidir:

Algoritma

1.Adım: $i=1$ olarak belirle

$$x^* = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

2.Adım:

3.Adım: Eğer $f(x^*) < \text{eps}$ veya $|b-a|/2 < \text{eps}$ ise x^* çözümdür ve programdan çık

4.Adım: Eğer $f(a)f(x^*) < 0$ ise $b \rightarrow x^*$ olarak belirle, Eğer $f(x^*)f(b) < 0$ ise $a \rightarrow x^*$ olarak belirle

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

Örnek: $\tan(\pi-x)-x=0$ fonksiyonunun (1.6,3) aralığındaki kökünü Regula Falsi 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 New Variable Open Variable Clear Workspace Favorites Run and Time Clear Commands Simulink Layout Preferences Set Path Add-Ons Help Community Request Support Learn MATLAB

FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES

Current Folder: C:\Matlab_Dersler

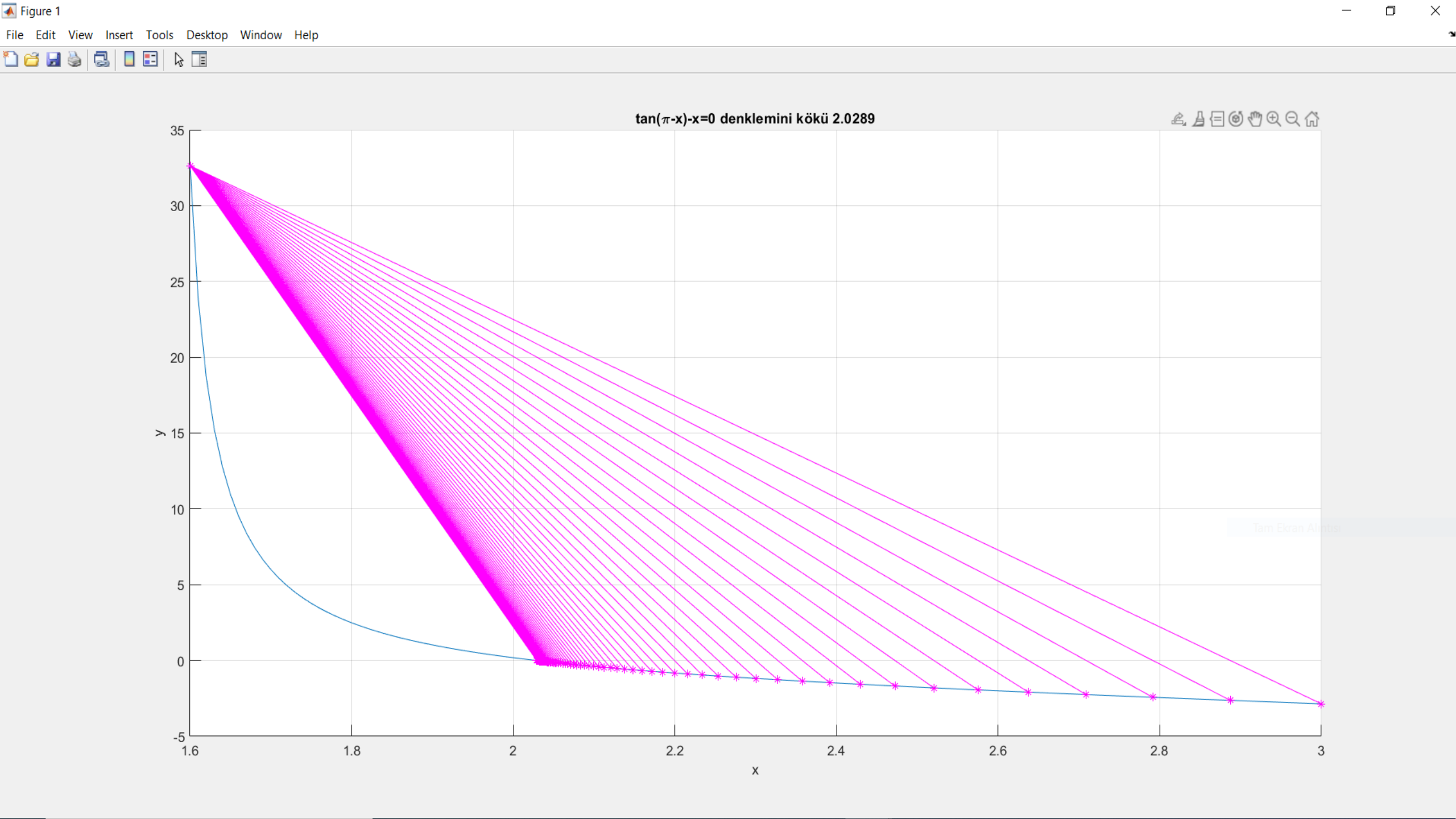
Editor - Untitled*

```
1 function regula_falsi%regula_falsi method
2 clc;clear all;
3 a=1.6;b=3;
4 x=[a:0.01:b];
5 y=f(x);
6 it=1;
7 if f(a)*f(b)>0
8     fprintf(' (tan(\pi-x)-x=0) denkleminin (%4.2f,%4.2f) aralığında kökü yoktur',a,b)
9 else
10     m=(a*f(b)-b*f(a))/(f(b)-f(a));
11     while abs(f(m))>0.001
12         line([a b],[f(a) f(b)],[1 1],'Marker','*','LineStyle','-','Color','m');
13         hold on;
14         it=it+1;
15         if f(a)*f(m)<0 b=m;
16         else if
17             f(m)*f(b)<0 a=m;
18         end
19         m=(a*f(b)-b*f(a))/(f(b)-f(a));
20     end
21     plot(x,y);
22     xlabel('x');
23     ylabel('y');
24     title(['tan(\pi-x)-x=0 denklemini kökü ',num2str(m)])
25     grid on
26     fprintf('tan(pi-x)-x=0 denkleminin kökü %6.4f dir ve iterasyon %6.4f dir',m,it)
27 end
28 function y=f(x)
29 y=tan(pi-x)-x;
30
31
32
33
34
```

Workspace

Name	Value
Tam Ekran Alıntısı	

Select a file to view details



BÖLÜM 3 - $F(x)=0$ FORMUNDA LİNEER OLMAYAN DENKLEMLERİN ÇÖZÜMLERİ

$f(x)=0$ denkleminin çözümüne, denklemin kökleri veya fonksiyonun sıfırları denir. Bir denklemin bir yada birden fazla kökü olabildiği gibi hiç kökü de yoktur.

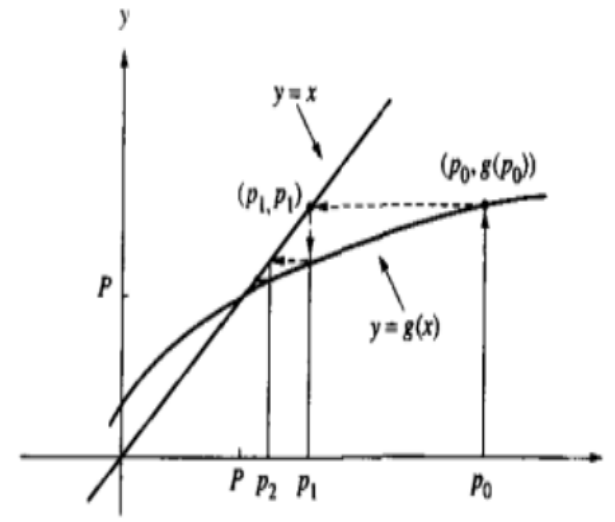
Örneğin $\sin x - x = 0$ denkleminin bir tek $x=0$ kökü varken,

$\tan x - x = 0$ denkleminin $x = 0, \pm 4.493, \pm 7.725, \dots$ şeklinde sonsuz kökü vardır. Bu bölümde denklemin kökünün bulunması için yöntemler verilip MATLAB kodu yazılacaktır.

SABİT NOKTA İTERASYONU (FIXED POINT ITERATION)

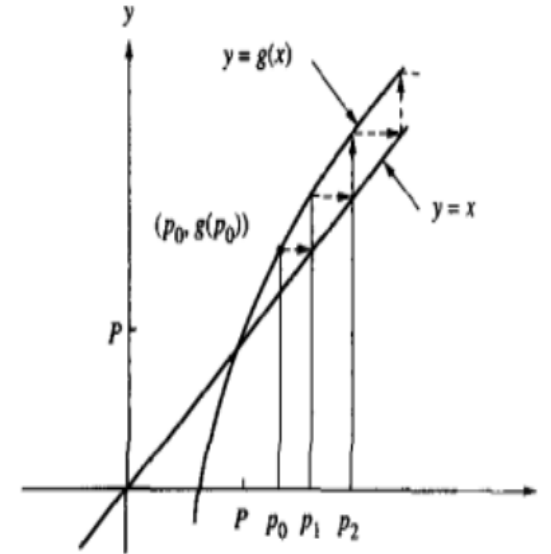
Eğer $g(x)$ fonksiyonu ve $g'(x)$ fonksiyonu sürekli ve $|g'(x)| < 1$ ise $x=g(x)$

denkleminin bir tek çözümü vardır ve çözüm $x_{k+1} = g(x_k)$ iterasyon yöntemi ile elde edilir.

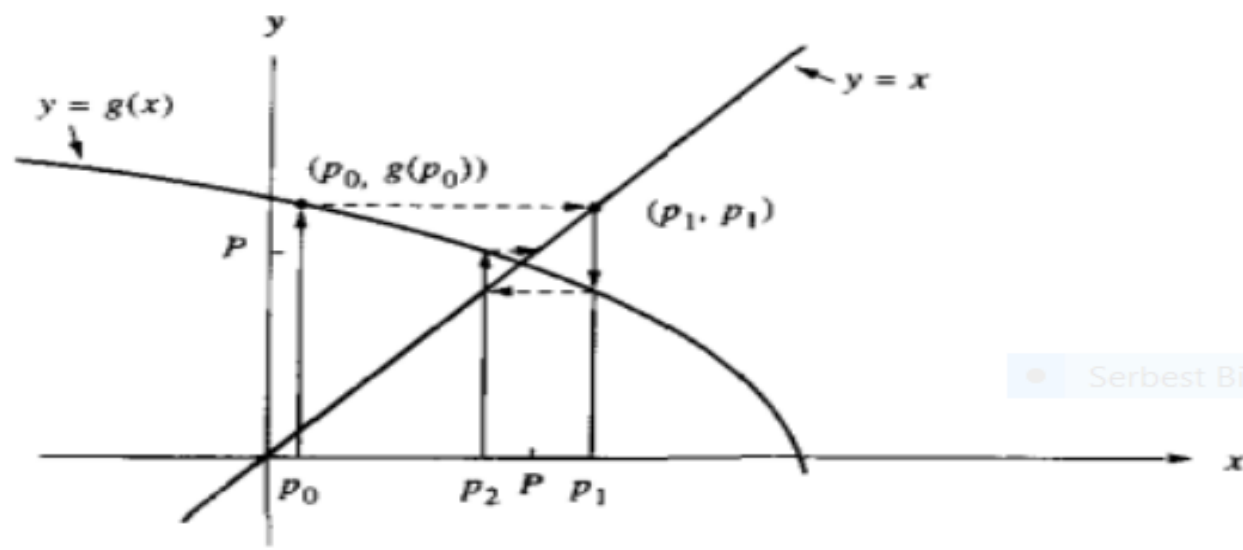


• Serbest Biçimli Ekran Alıntısı

→ Yakınsaklık: $0 < g'(P) < 1$ durumu



→ Iraksak: $1 < g'(P)$.

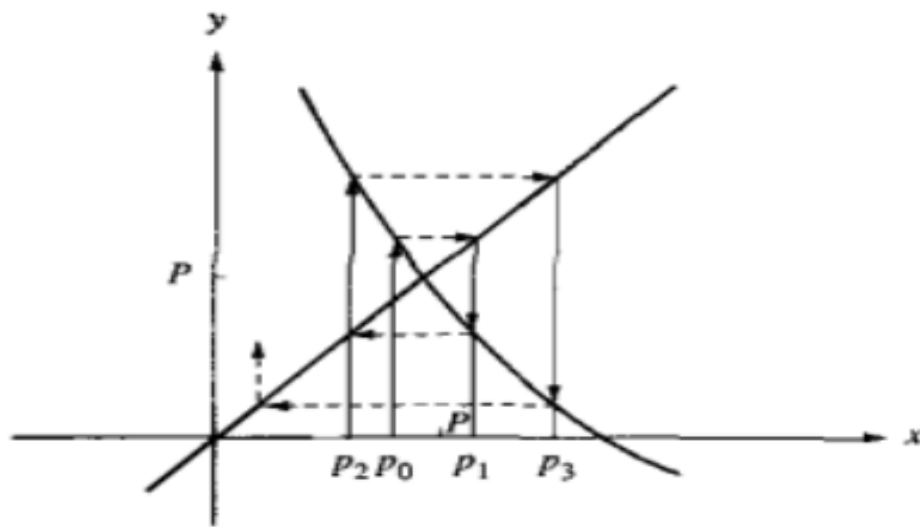


● Serbest Biçimli Ekran Alıntısı

→ Yakınsaklık

$$-1 < g'(P) < 0.$$

durumu



→ Iraksaklık: $g'(P) < -1$. durumu

Alıştırma 1. $x = 2^{-x}$ denkleminin...

Örnek: $x=2^{-x}$ denkleminin kökünü $[0,1]$ aralığında bulacak şekilde sabit nokta iterasyonunu MATLAB kodunu 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

Analyze Code Run and Time Clear Commands

Simulink Layout Preferences Set Path Parallel Add-Ons Help

Community Request Support Learn MATLAB

Search Documentation

Current Folder

Name

sabit_nokta.mat

sabit_nokta.m

ikiye_bolme.m

ikiye_bolme.m (Function)

ikiye_bolme()

f(x)

Workspace

Name Value

Editor - C:\Matlab_Dersler\sabit_nokta.m

```
1 function sabit_nokta
2 clc;clear all;
3 %x=2^(-x), [0,1]
4 xx=linspace(0,1,100);
5 gx=2.^(-xx);
6 x0=0;
7 plot(x0,0,'or');hold on;
8 x1=2^(-x0);
9 plot(x1,0,'or');
10 line([x0 x0],[0,x1],[1 1],'Marker','*','LineStyle','-','Color','m');
11 hold on;
12 it=1;
13 while abs(x1-x0)>eps
14     x0=x1;
15     x1=2^(-x0);
16     plot(x1,0,'or');
17     line([x0 x0],[0,x1],[1 1],'Marker','*','LineStyle','-','Color','m');
18     hold on;
19     it=it+1;
20 end
21 plot(xx,xx,gx);
22 xlabel('x');
23 ylabel('y');
24 title(['g(x)=2^{-x} eğrisi ile f(x)=x doğrusunun kesimnoktası ',num2str(x1)])
25 fprintf('g(x)=2^{-x} eğrisi ile f(x)=x doğrusunun kesim noktası %6.4f dir ve iterasyon %6.4f dir',x1,it)
26 save('sabit_nokta.mat','it','x1','x0')
```

Command Window



Figure 1



File Edit View Insert Tools Desktop Window Help

