

# Diferensiyel Denklem Sistemleri için Sayısal Yöntemler

Erhan Coşkun

Karadeniz Teknik Üniversitesi

Aralık, 2018

- Bu bölümde

- Bu bölümde
- yüksek mertebeden lineer bir denklemin birinci mertebeden bir sisteme dönüştürülmesi,

- Bu bölümde
- yüksek mertebeden lineer bir denklemin birinci mertebeden bir sisteme dönüştürülmesi,
- diferensiyel denklem sistemleri için İleri Euler, Runge-Kutta(II) ve (IV) yöntemlerini ve

- Bu bölümde
- yüksek mertebeden lineer bir denklemin birinci mertebeden bir sisteme dönüştürülmesi,
- diferensiyel denklem sistemleri için İleri Euler, Runge-Kutta(II) ve (IV) yöntemlerini ve
- **MATLAB/OCTAVE çözücülerin sistemler için nasıl uygulanabileceği konularını inceliyoruz .**

# Tipik Uygulama alanları

- Diferensiyel denklem sistemleri, uygulamalı bilimlerin çeşitli alanlarında karşımıza çıkarlar.

# Tipik Uygulama alanları

- Diferensiyel denklem sistemleri, uygulamalı bilimlerin çeşitli alanlarında karşımıza çıkarlar.
- Özellikle fen bilimleri ve mühendislik başta olmak üzere,

# Tipik Uygulama alanları

- Diferensiyel denklem sistemleri, uygulamalı bilimlerin çeşitli alanlarında karşımıza çıkarlar.
- Özellikle fen bilimleri ve mühendislik başta olmak üzere,
- çok sayıda bağımlı değişkenin tek bir bağımsız değişkene bağlı olarak bilinen yasalar çerçevesinde değişiminin söz konusu olduğu her problem bir diferensiyel denklem sistemi biçiminde modellenabilir:



# Tipik Uygulama alanları

- Diferensiyel denklem sistemleri, uygulamalı bilimlerin çeşitli alanlarında karşımıza çıkarlar.
- Özellikle fen bilimleri ve mühendislik başta olmak üzere,
- çok sayıda bağımlı değişkenin tek bir bağımsız değişkene bağlı olarak bilinen yasalar çerçevesinde değişiminin söz konusu olduğu her problem bir diferensiyel denklem sistemi biçiminde modellenebilir:
- **Örneğin birbiri ile rekabet içerisinde yaşayan ve  $t$  anındaki nüfusları sırasıyla  $N_1(t)$  ve  $N_2(t)$  ile gösterilen canlı nüfuslarının zamanla değişimini modelleyen nonlinear lojistik model**

$$\begin{aligned}\frac{dN_1}{dt} &= aN_1 - bN_1N_2 \\ \frac{dN_2}{dt} &= -cN_2 + dN_1N_2\end{aligned}\tag{1}$$

**biçiminde ifade edilebilir ve bu modeli bilinen analitik yöntemler yardımıyla çözemeyiz.**

# Tipik Uygulama alanları: $F=ma$



$$ma = F$$

olarak bilinen Newton hareket kanunu kapsamında değerlendirilebilen her fiziksel olaya ait matematiksel model birinci mertebeden bir lineer diferensiyel denklem sistemine dönüştürülebilir:



$$ma = F$$

olarak bilinen Newton hareket kanunu kapsamında değerlendirilebilen her fiziksel olaya ait matematiksel model birinci mertebeden bir lineer diferensiyel denklem sistemine dönüştürülebilir:

- Öncelikle ideal yaylar için yay salınım modelini gözönüne alalım:

$$\begin{aligned} my'' &= -cy' - ky + f(t) \\ y(0) &= \alpha, y'(0) = \beta \end{aligned} \quad (2)$$

# Tipik Uygulama alanları: $F=ma$



$$ma = F$$

olarak bilinen Newton hareket kanunu kapsamında değerlendirilebilen her fiziksel olaya ait matematiksel model birinci mertebeden bir lineer diferensiyel denklem sistemine dönüştürülebilir:

- Öncelikle ideal yaylar için yay salınım modelini gözönüne alalım:

$$\begin{aligned} my'' &= -cy' - ky + f(t) \\ y(0) &= \alpha, y'(0) = \beta \end{aligned} \quad (2)$$

- Burada  $m$  kütle,  $c$  ortam direnci,  $k$  yayın esneklik düzeyini belirleyen sabit ve  $f(t)$  ise dış kuvveti temsil etmektedir.

## Tipik Uygulama alanları: $F=ma$



$$ma = F$$

olarak bilinen Newton hareket kanunu kapsamında değerlendirilebilen her fiziksel olaya ait matematiksel model birinci mertebeden bir lineer diferensiyel denklem sistemine dönüştürülebilir:

- Öncelikle ideal yaylar için yay salınım modelini gözönüne alalım:

$$\begin{aligned} my'' &= -cy' - ky + f(t) \\ y(0) &= \alpha, y'(0) = \beta \end{aligned} \quad (2)$$

- Burada  $m$  kütle,  $c$  ortam direnci,  $k$  yayın esneklik düzeyini belirleyen sabit ve  $f(t)$  ise dış kuvveti temsil etmektedir.
- Ayrıca  $t$  zaman değişkeni olmak üzere  $y = y(t)$  ise asılı cismin statik denge konumundan yer değiştirmesidir.

# Tipik Uygulama alanları: $F=ma$



$$ma = F$$

olarak bilinen Newton hareket kanunu kapsamında değerlendirilebilen her fiziksel olaya ait matematiksel model birinci mertebeden bir lineer diferensiyel denklem sistemine dönüştürülebilir:

- Öncelikle ideal yaylar için yay salınım modelini gözönüne alalım:

$$\begin{aligned} my'' &= -cy' - ky + f(t) \\ y(0) &= \alpha, y'(0) = \beta \end{aligned} \quad (2)$$

- Burada  $m$  kütle,  $c$  ortam direnci,  $k$  yayın esneklik düzeyini belirleyen sabit ve  $f(t)$  ise dış kuvveti temsil etmektedir.
- Ayrıca  $t$  zaman değişkeni olmak üzere  $y = y(t)$  ise asılı cismin statik denge konumundan yer değiştirmesidir.
- **Aşağı yön pozitif olarak seçilirse, pozitif  $y$ , cismin denge konumunun aşağısında, negatif  $y$  ise yukarısında olduğunu ifade etmektedir.**

## Tipik Uygulama alanları: $F=ma$ (ideal yay)

- (2) başlangıç değer problemi  $u = y, v = y'$  değişken dönüşümü ile

## Tipik Uygulama alanları: $F=ma$ (ideal yay)

- (2) başlangıç değer problemi  $u = y$ ,  $v = y'$  değişken dönüşümü ile
- 

$$\begin{aligned}u' &= v \\v' &= 1/m(-cv - ku + f(t)) \\u(0) &= \alpha, v(0) = \beta\end{aligned}\tag{3}$$

başlangıç değer sistemine dönüştürülür.



## Tipik Uygulama alanları: $F=ma$ (ideal yay)

- (2) başlangıç değer problemi  $u = y$ ,  $v = y'$  değişken dönüşümü ile
- 

$$\begin{aligned}u' &= v \\v' &= 1/m(-cv - ku + f(t)) \\u(0) &= \alpha, v(0) = \beta\end{aligned}\tag{3}$$

başlangıç değer sistemine dönüştürülür.

- (2) lineer bir denklem ve dolayısıyla da (3) lineer bir diferensiyel denklem sistemidir ve analitik çözümleri mevcuttur.

## Tipik Uygulama alanları: $F=ma$ (İdeal olmayan yay)

- (2) başlangıç değer problemi  $u = y, v = y'$  değişken dönüşümü ile

## Tipik Uygulama alanları: $F=ma$ (İdeal olmayan yay)

- (2) başlangıç değer problemi  $u = y$ ,  $v = y'$  değişken dönüşümü ile
- Ancak yayın boyundaki değişim ile oluşan geri çağırma veya itme kuvvetini ilişkilendiren Hook yasası,  $F_{yay} = -ku$ , ideal olmayan yaylar için geçerli değildir.

## Tipik Uygulama alanları: $F=ma$ (İdeal olmayan yay)

- (2) başlangıç değer problemi  $u = y$ ,  $v = y'$  değişken dönüşümü ile
- Ancak yayın boyundaki değişim ile oluşan geri çağırma veya itme kuvvetini ilişkilendiren Hook yasası,  $F_{yay} = -ku$ , ideal olmayan yaylar için geçerli değildir.
- Bu durumda örneğin

$$F_{yay} = -ku - \delta u^3, k > 0, \delta > 0$$

biçiminde bir yaklaşımın kullanılması halinde elde edilen (2) ve (3) sırasıyla nonlinear denklem denklem sistemine dönüşürler ki bu sistemin elemanter fonksiyonlar cinsinden analitik çözümü mevcut değildir.

# Tipik Uygulama alanları: $F=ma$ (Asmalı köprü askı sistemleri)

- Daha değişik konfigürasyonlar da (3) türünde modeller yardımıyla ifade edilebilirler.

## Tipik Uygulama alanları: $F=ma$ (Asmalı köprü askı sistemleri)

- Daha değişik konfigürasyonlar da (3) türünde modeller yardımıyla ifade edilebilirler.
- Örneğin boğaz köprüsü gibi bir asmalı köprünün askıları germeye karşı direnç gösteren ve fakat sıkışmaya karşı direnç göstermeyen yaylar olarak düşünülebilirler.

## Tipik Uygulama alanları: $F=ma$ (Asmalı köprü askı sistemleri)

- Daha değişik konfigürasyonlar da (3) türünde modeller yardımıyla ifade edilebilirler.
- Örneğin boğaz köprüsü gibi bir asmalı köprünün askıları germeye karşı direnç gösteren ve fakat sıkışmaya karşı direnç göstermeyen yaylar olarak düşünülebilirler.
- Bu durumda askıda oluşan geri çağırma kuvveti

$$F_{yay} = -ku^+, u^+ = \begin{cases} u, & u > 0 \\ 0 & u \leq 0 \end{cases}$$

olarak ifade edilebilir ve

# Tipik Uygulama alanları: $F=ma$ (Asmalı köprü askı sistemleri)

- Daha değişik konfigürasyonlar da (3) türünde modeller yardımıyla ifade edilebilirler.
- Örneğin boğaz köprüsü gibi bir asmalı köprünün askıları germeye karşı direnç gösteren ve fakat sıkışmaya karşı direnç göstermeyen yaylar olarak düşünülebilirler.
- Bu durumda askıda oluşan geri çağırma kuvveti

$$F_{yay} = -ku^+, u^+ = \begin{cases} u, & u > 0 \\ 0 & u \leq 0 \end{cases}$$

olarak ifade edilebilir ve

- bu kuvvet formülasyonunun kullanılması halinde oluşan nonlinear modelin de analitik çözümü mevcut değildir.



- En genel halde  $n$ -inci basamaktan

$$a_0(t)y^{(n)} + a_1(t)y^{(n-1)} + \cdots + a_{(n-1)}y = f(t), \quad (4)$$

$$y(0) = b_0, y'(0) = b_1, \dots, y^{(n-1)}(0) = b_{n-1}$$

başlangıç değer problemi

# $n$ -inci basamaktan lineer denklemler

- En genel halde  $n$ -inci basamaktan

$$a_0(t)y^{(n)} + a_1(t)y^{(n-1)} + \dots + a_{(n-1)}y = f(t), \quad (4)$$

$$y(0) = b_0, y'(0) = b_1, \dots, y^{(n-1)}(0) = b_{n-1}$$

başlangıç değer problemi

- $$u_1 = y, u_2 = y', \dots, u_n = y^{(n-1)}$$

isimli yeni değişkenler ile  $n$  bilinmeyenli  $n$  adet denklemden oluşan

# $n$ -inci basamaktan lineer denklemler

- En genel halde  $n$ -inci basamaktan

$$a_0(t)y^{(n)} + a_1(t)y^{(n-1)} + \dots + a_{(n-1)}y = f(t), \quad (4)$$

$$y(0) = b_0, y'(0) = b_1, \dots, y^{(n-1)}(0) = b_{n-1}$$

başlangıç değer problemi

- $u_1 = y, u_2 = y', \dots, u_n = y^{(n-1)}$

isimli yeni değişkenler ile  $n$  bilinmeyenli  $n$  adet denklemden oluşan

- $u_1' = u_2,$   
 $u_2' = u_3,$   
...  
 $u_n' = 1/a_0(t)(f(t) - a_1(t)u_n - \dots - a_{n-1}(t)u_1),$  (5)  
 $u_1(0) = b_0, u_2(0) = b_1, \dots, u_n(0) = b_n$

bir başlangıç değer sistemine dönüştürülebilir.

- Özel bazı durumlar hariç, değişken katsayılı ve (5) biçiminde ifade edilebilen genel bir sistemin de çözümünü analitik olarak ifade edebilmek mümkün değildir.

- Özel bazı durumlar hariç, değişken katsayılı ve (5) biçiminde ifade edilebilen genel bir sistemin de çözümünü analitik olarak ifade edebilmek mümkün değildir.
- **Bu durumda sayısal yöntemlerin kullanılması kaçınılmazdır.**

- Özel bazı durumlar hariç, değişken katsayılı ve (5) biçiminde ifade edilebilen genel bir sistemin de çözümünü analitik olarak ifade edebilmek mümkün değildir.
- Bu durumda sayısal yöntemlerin kullanılması kaçınılmazdır.
- İlk olarak önceki bölümde incelediğimiz İleri Euler yönteminin denklem sistemleri için nasıl uygulanabildiğini görelim.

# İki bilinmeyenli sistemler



$$\begin{aligned}\frac{dx}{dt} &= f(t, x, y) \\ \frac{dy}{dt} &= g(t, x, y) \\ x(0) &= x_1, y(0) = y_1, t \in [0, b]\end{aligned}\tag{6}$$

başlangıç değer problemini gözönüne alalım.

# İki bilinmeyenli sistemler



$$\begin{aligned}\frac{dx}{dt} &= f(t, x, y) \\ \frac{dy}{dt} &= g(t, x, y) \\ x(0) &= x_1, y(0) = y_1, t \in [0, b]\end{aligned}\tag{6}$$

başlangıç değer problemini gözönüne alalım.

- Vektör notasyonu yardımıyla

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, F(X) = \begin{bmatrix} f(t, x, y) \\ g(t, x, y) \end{bmatrix}, X^{(1)} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

olarak tanımlayalım.



# İki bilinmeyenli sistemler



$$\begin{aligned}\frac{dx}{dt} &= f(t, x, y) \\ \frac{dy}{dt} &= g(t, x, y) \\ x(0) &= x_1, y(0) = y_1, t \in [0, b]\end{aligned}\tag{6}$$

başlangıç değer problemini gözönüne alalım.

- Vektör notasyonu yardımıyla

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, F(X) = \begin{bmatrix} f(t, x, y) \\ g(t, x, y) \end{bmatrix}, X^{(1)} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

olarak tanımlayalım.

- Bu durumda (6) sistemini

$$\frac{dX}{dt} = F(t, X), X(0) = X^{(1)}\tag{7}$$

biçiminde yazabiliriz.

# Sistemler için İleri Euler yöntemi

- $h$  adım uzunluğu ile (7) için İleri Euler yaklaşımları skaler denklemlere benzer biçimde

$$X^{(i+1)} = X^{(i)} + hF(t_i, X^{(i)}), \quad i = 1, 2, \dots, n \quad (8)$$

olarak tanımlanır.

# Sistemler için İleri Euler yöntemi

- $h$  adım uzunluğu ile (7) için İleri Euler yaklaşımları skaler denklemlere benzer biçimde

$$X^{(i+1)} = X^{(i)} + hF(t_i, X^{(i)}), \quad i = 1, 2, \dots, n \quad (8)$$

olarak tanımlanır.

- (8) yaklaşımını her bir bileşen için ayrı ayrı yazarak,

# Sistemler için İleri Euler yöntemi

- $h$  adım uzunluğu ile (7) için İleri Euler yaklaşımları skaler denklemlere benzer biçimde

$$X^{(i+1)} = X^{(i)} + hF(t_i, X^{(i)}), \quad i = 1, 2, \dots, n \quad (8)$$

olarak tanımlanır.

- (8) yaklaşımını her bir bileşen için ayrı ayrı yazarak,
- 

$$\begin{aligned}x_{i+1} &= x_i + hf(t_i, x_i, y_i), \\y_{i+1} &= y_i + hg(t_i, x_i, y_i), \\x(0) &= x_1, y(0) = y_1, \quad i = 1, \dots, n\end{aligned}$$

elde ederiz.

## Örnek 1

$$x' = -2x + 2y$$

$$y' = -x$$

$$x(0) = -4, y(0) = 0$$

*başlangıç değer problemi verilsin. Verilen problemin*

- *analitik çözümünü belirleyiniz,*

## Örnek 1

$$x' = -2x + 2y$$

$$y' = -x$$

$$x(0) = -4, y(0) = 0$$

*başlangıç değer problemi verilsin. Verilen problemin*

- *analitik çözümünü belirleyiniz,*
- *$h = 0.1$  için  $[0, 5]$  aralığındaki yaklaşık çözümlerini ileri Euler yöntemiyle belirleyerek*

## Örnek 1

$$x' = -2x + 2y$$

$$y' = -x$$

$$x(0) = -4, y(0) = 0$$

*başlangıç değer problemi verilsin. Verilen problemin*

- *analitik çözümünü belirleyiniz,*
- *$h = 0.1$  için  $[0, 5]$  aralığındaki yaklaşık çözümlerini ileri Euler yöntemiyle belirleyerek*
- *$[-4, 4] \times [-4, 4]$  aralığında uygun bazı başlangıç değerleri ile çözüm eğrilerinin grafiğini çizdiriniz.*

# Sistemler için İleri Euler yöntemi

- $X = \begin{bmatrix} x \\ y \end{bmatrix}$ ,  $A = \begin{bmatrix} -2 & 2 \\ -1 & 0 \end{bmatrix}$ ,  $X^{(1)} = X(0) = \begin{bmatrix} -4 \\ 0 \end{bmatrix}$  notasyonu ile verilen denklem sistemini



# Sistemler için İleri Euler yöntemi

- $X = \begin{bmatrix} x \\ y \end{bmatrix}$ ,  $A = \begin{bmatrix} -2 & 2 \\ -1 & 0 \end{bmatrix}$ ,  $X^{(1)} = X(0) = \begin{bmatrix} -4 \\ 0 \end{bmatrix}$  notasyonu ile verilen denklem sistemini

$$\frac{dX}{dt} = AX$$

biçiminde ifade edilebiliriz.

# Sistemler için İleri Euler yöntemi

- $X = \begin{bmatrix} x \\ y \end{bmatrix}$ ,  $A = \begin{bmatrix} -2 & 2 \\ -1 & 0 \end{bmatrix}$ ,  $X^{(1)} = X(0) = \begin{bmatrix} -4 \\ 0 \end{bmatrix}$  notasyonu ile verilen denklem sistemini

- $$\frac{dX}{dt} = AX$$

biçiminde ifade edilebiliriz.

- $A$  matrisinin özdeğerleri

$$\lambda_1 = -1 + i, \lambda_2 = -1 - i$$

dir.

# Sistemler için İleri Euler yöntemi

- $X = \begin{bmatrix} x \\ y \end{bmatrix}$ ,  $A = \begin{bmatrix} -2 & 2 \\ -1 & 0 \end{bmatrix}$ ,  $X^{(1)} = X(0) = \begin{bmatrix} -4 \\ 0 \end{bmatrix}$  notasyonu ile verilen denklem sistemini

$$\frac{dX}{dt} = AX$$

biçiminde ifade edilebiliriz.

- $A$  matrisinin özdeğerleri

$$\lambda_1 = -1 + i, \lambda_2 = -1 - i$$

dir.

- Ayrıca  $\lambda_1 = -1 + i$  ye karşılık gelen özvektör

$$V_1 = \begin{bmatrix} 1 \\ (1 + i)/2 \end{bmatrix}$$

dir.

- Buna göre denklemin bir çözümü

$$e^{\lambda_1 t} V_1 = e^{(-1+i)t} \begin{bmatrix} 1 \\ (1+i)/2 \end{bmatrix} = e^{-t} \begin{bmatrix} \cos t + i \sin t \\ 1/2(\cos t - \sin t) + i/2(\cos t - \sin t) \end{bmatrix}$$

olarak tanımlanır.

# Sistemler için İleri Euler yöntemi

- Ayrıca

$$\text{Reel}(e^{\lambda_1 t} V_1) = e^{-t} \begin{bmatrix} \cos t \\ 1/2(\cos t - \sin t) \end{bmatrix}$$

ve

# Sistemler için İleri Euler yöntemi

- Ayrıca

$$\text{Reel}(e^{\lambda_1 t} V_1) = e^{-t} \begin{bmatrix} \cos t \\ 1/2(\cos t - \sin t) \end{bmatrix}$$

ve

- $$\text{Sanal}(e^{\lambda_1 t} V_1) = e^{-t} \begin{bmatrix} \sin t \\ 1/2(\cos t + \sin t) \end{bmatrix}$$

de çözüm olup, genel çözüm reel ve sanal kısımların kombinasyonu olarak

# Sistemler için İleri Euler yöntemi

- Ayrıca

$$\text{Reel}(e^{\lambda_1 t} V_1) = e^{-t} \begin{bmatrix} \cos t \\ 1/2(\cos t - \sin t) \end{bmatrix}$$

ve

- 

$$\text{Sanal}(e^{\lambda_1 t} V_1) = e^{-t} \begin{bmatrix} \sin t \\ 1/2(\cos t + \sin t) \end{bmatrix}$$

de çözüm olup, genel çözüm reel ve sanal kısımların kombinasyonu olarak

- 

$$\begin{aligned} X(t) &= C_1 \text{Reel}(e^{\lambda_1 t} V_1) + C_2 \text{Sanal}(e^{\lambda_1 t} V_1) \\ &= C_1 e^{-t} \begin{bmatrix} \cos t \\ 1/2(\cos t - \sin t) \end{bmatrix} + C_2 e^{-t} \begin{bmatrix} \sin t \\ 1/2(\cos t + \sin t) \end{bmatrix} \end{aligned}$$

elde ederiz.



$$X(0) = \begin{bmatrix} C_1 \\ C_1 \frac{1}{2}(1) + C_2 \frac{1}{2}(1) \end{bmatrix} = \begin{bmatrix} -4 \\ 0 \end{bmatrix}$$

olup, buradan

$$C_1 = -4, C_2 = 4$$

elde ederiz.





$$X(0) = \begin{bmatrix} C_1 \\ C_1 \frac{1}{2}(1) + C_2 \frac{1}{2}(1) \end{bmatrix} = \begin{bmatrix} -4 \\ 0 \end{bmatrix}$$

olup, buradan

$$C_1 = -4, C_2 = 4$$

elde ederiz.

- O halde aradığımız çözüm

$$x(t) = 4e^{-t}(sint - cost)$$

$$y(t) = 4e^{-t}sint$$

olarak elde ederiz.

```
(%i1) denk1:'diff(x(t),t)=-2*x(t)+2*y(t);
```

```
(%o1)  $\frac{d}{dt} x(t) = 2 y(t) - 2 x(t)$ 
```

```
(%i2) denk2:'diff(y(t),t)=-x(t);
```

```
(%o2)  $\frac{d}{dt} y(t) = -x(t)$ 
```

```
(%i3) atvalue(x(t),t=0,-4);
```

```
(%o3) -4
```

```
(%i4) atvalue(y(t),t=0,0);
```

```
(%o4) 0
```

```
(%i5) desolve([denk1,denk2],[x(t),y(t)]);
```

```
(%o5) [x(t)=%e-t(4 sin(t)-4 cos(t)),y(t)=4 %e-t sin(t)]
```

# Sistemler için İleri Euler yöntemi

- $h$  adım uzunluğu ile ileri Euler yöntemini  $[0, 5]$  aralığında uygulayalım.

# Sistemler için İleri Euler yöntemi

- $h$  adım uzunluğu ile ileri Euler yöntemini  $[0, 5]$  aralığında uygulayalım.
- $(i + 1)$ –inci adımdaki yaklaşımlar için

$$x_{i+1} = x_i + 2h(-x_i + y_i)$$

$$y_{i+1} = y_i + h(-x_i)$$

fark denklemlerini elde ederiz.

# Sistemler için İleri Euler yöntemi

- $h$  adım uzunluğu ile ileri Euler yöntemini  $[0, 5]$  aralığında uygulayalım.
- $(i + 1)$ -inci adımdaki yaklaşımlar için

$$x_{i+1} = x_i + 2h(-x_i + y_i)$$

$$y_{i+1} = y_i + h(-x_i)$$

fark denklemlerini elde ederiz.

- Örneğin  $h = 0.1$  ve  $(x_1, y_1) = (-4, 0)$  için ilk yaklaşım

$$\begin{aligned}x_2 &= x_1 + 0.1 \times 2 \times (-x_1 + y_1) \\ &= -4 + 0.1 \times 2 \times (-4 + 0) = -3.2\end{aligned}$$

$$\begin{aligned}y_2 &= y_1 + 0.1(-x_1) \\ &= 0 + 0.1 \times (-(-4)) = 0.4\end{aligned}$$

olarak elde edilir.

- Benzer biçimde

$$\begin{aligned}x_3 &= x_2 + 0.1 \times 2 \times (-x_2 + y_2) \\ &= -3.2 + 0.1 \times 2 \times (-3.2 + 0.4) \\ &= -2.48 \\ y_3 &= y_2 + 0.1 \times (-x_2) \\ &= 0.4 + 0.1 \times (-(-3.2)) \\ &= 0.72\end{aligned}$$

olarak elde edilir.

# Sistemler için İleri Euler yöntemi

- $h = 0.1$  adım uzunluğu ile  $[0, 1]$  aralığında elde ettiğimiz yaklaşım değerleri  $(x_i, y_i)$ ,  $i = 1, \dots, 11$  ile Tablo 1 de sunulmaktadır.

$T$	$x_i$	$y_i$
0	-4.0000	0
0.1	-3.2000	0.4000
0.2	-2.4800	0.7200
0.3	-1.8400	0.9680
...	...	...
1	0.6621	1.3260

Tablo 1: Örnek 1 için yaklaşımlar

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`



# Sistemler için İleri Euler yöntemi(kod)

- function [T,X]=ieulers(h,Tmax)
- % İleri Euler yöntemi için Vektör Cebirsel Kod

# Sistemler için İleri Euler yöntemi(kod)

- function [T,X]=ieulers(h,Tmax)
- % İleri Euler yöntemi için Vektör Cebirsel Kod
- t=0;T=0; x1=4;y1=-3;

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`

# Sistemler için İleri Euler yöntemi(kod)

- function [T,X]=ieulers(h,Tmax)
- % İleri Euler yöntemi için Vektör Cebirsel Kod
- t=0;T=0; x1=4;y1=-3;
- X=[x1,y1];
- U=X';
- while (t<Tmax)

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`
- `t=t+h;`

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`
- `t=t+h;`
- `X=[X;U'];`



# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`
- `t=t+h;`
- `X=[X;U'];`
- `T=[T;t];`

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`
- `t=t+h;`
- `X=[X;U'];`
- `T=[T;t];`
- `end`

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`
- `t=t+h;`
- `X=[X;U'];`
- `T=[T;t];`
- `end`
- `plot(X(:,1),X(:,2));`

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`
- `t=t+h;`
- `X=[X;U'];`
- `T=[T;t];`
- `end`
- `plot(X(:,1),X(:,2));`
- `function z=f(t,X)`

# Sistemler için İleri Euler yöntemi(kod)

- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`
- `t=t+h;`
- `X=[X;U'];`
- `T=[T;t];`
- `end`
- `plot(X(:,1),X(:,2));`
- `function z=f(t,X)`
- `x=X(1);y=X(2);`

# Sistemler için İleri Euler yöntemi(kod)

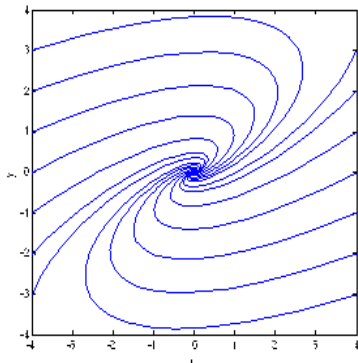
- `function [T,X]=ieulers(h,Tmax)`
- `% İleri Euler yöntemi için Vektör Cebirsel Kod`
- `t=0;T=0; x1=4;y1=-3;`
- `X=[x1,y1];`
- `U=X';`
- `while (t<Tmax)`
- `U=U+h*f(t,U);`
- `t=t+h;`
- `X=[X;U'];`
- `T=[T;t];`
- `end`
- `plot(X(:,1),X(:,2));`
- `function z=f(t,X)`
- `x=X(1);y=X(2);`
- `z=[-2*x+2*y;-x];`

# Sistemler için İleri Euler yöntemi

- Farklı başlangıç değerleri için Örnek 1 e ait  $(x(t), y(t))$  çözüm eğrileri Şekil 1 de sunulmaktadır.

# Sistemler için İleri Euler yöntemi

- Farklı başlangıç değerleri için Örnek 1 e ait  $(x(t), y(t))$  çözüm eğrileri Şekil 1 de sunulmaktadır.

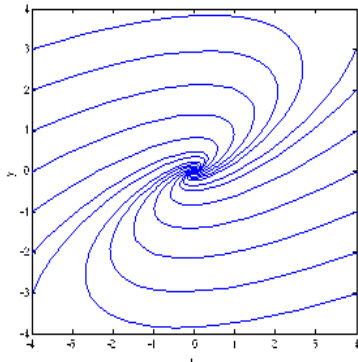


Şekil 1: Örnek 1 e ait  $(x(t), y(t))$  çözüm eğrileri



# Sistemler için İleri Euler yöntemi

- Farklı başlangıç değerleri için Örnek 1 e ait  $(x(t), y(t))$  çözüm eğrileri Şekil 1 de sunulmaktadır.



Şekil 1: Örnek 1 e ait  $(x(t), y(t))$  çözüm eğrileri

- Her bir çözüm eğrisinin  $(0, 0)$  denge noktasına yakınsadığını gözlemleyiniz

## Sistemler için II. Basamaktan Runge-Kutta yöntemi

- Sistemler için Runge-Kutta yöntemleri skaler başlangıç değer problemlerine benzer biçimde tanımlanır:

## Sistemler için II. Basamaktan Runge-Kutta yöntemi

- Sistemler için Runge-Kutta yöntemleri skaler başlangıç değer problemlerine benzer biçimde tanımlanır:
- Örneğin (7) için ikinci mertebeden Runge-Kutta yöntemi(RKIIs) aşağıdaki gibi ifade edilebilir:

## Sistemler için II. Basamaktan Runge-Kutta yöntemi

- Sistemler için Runge-Kutta yöntemleri skaler başlangıç değer problemlerine benzer biçimde tanımlanır:
- Örneğin (7) için ikinci mertebeden Runge-Kutta yöntemi(RKIIs) aşağıdaki gibi ifade edilebilir:



$$\begin{aligned}M_1 &= F(t_i, X^{(i)}); \\M_2 &= F(t_i + h, X^{(i)} + hM_1); \\M &= (M_1 + M_2)/2; \\X^{(i+1)} &= X^{(i)} + hM, i = 1, 2, \dots\end{aligned}\tag{9}$$

## Sistemler için II. Basamaktan Runge-Kutta yöntemi

- Sistemler için Runge-Kutta yöntemleri skaler başlangıç değer problemlerine benzer biçimde tanımlanır:
- Örneğin (7) için ikinci mertebeden Runge-Kutta yöntemi(RKIIs) aşağıdaki gibi ifade edilebilir:



$$\begin{aligned}M_1 &= F(t_i, X^{(i)}); \\M_2 &= F(t_i + h, X^{(i)} + hM_1); \\M &= (M_1 + M_2)/2; \\X^{(i+1)} &= X^{(i)} + hM, i = 1, 2, \dots\end{aligned}\tag{9}$$

- Burada  $M_i$  lerin eğimlerden oluşan vektörler olduğuna dikkat edelim.

## Örnek 2

$$dN_1/dt = aN_1 - bN_1N_2$$

$$dN_2/dt = -cN_2 + dN_1N_2$$

$$N_1(0) = 10; N_2(0) = 5$$

*modelini*

- *a = 0.2, b = 0.1, c = 0.1, d = 0.3 parametreleri ile gözönüne alalım ve*

## Örnek 2

$$dN_1/dt = aN_1 - bN_1N_2$$

$$dN_2/dt = -cN_2 + dN_1N_2$$

$$N_1(0) = 10; N_2(0) = 5$$

*modelini*

- $a = 0.2, b = 0.1, c = 0.1, d = 0.3$  parametreleri ile gözönüne alalım ve
- *ikinci Mertebeden Runge-Kutta RKII yöntemi yardımıyla  $N_1(t)$  ve  $N_2(t)$  nüfusları için uygun sayısal yaklaşımları*

## Örnek 2

$$dN_1/dt = aN_1 - bN_1N_2$$

$$dN_2/dt = -cN_2 + dN_1N_2$$

$$N_1(0) = 10; N_2(0) = 5$$

*modelini*

- $a = 0.2, b = 0.1, c = 0.1, d = 0.3$  parametreleri ile gözönüne alalım ve
- *ikinci Mertebeden Runge-Kutta RKII yöntemi yardımıyla  $N_1(t)$  ve  $N_2(t)$  nüfusları için uygun sayısal yaklaşımları*
- $[0, 15]$  aralığında  $h = 0.1$  adım uzunluğu ile hesaplayınız



## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanım,h), tanım=[t1,tson]

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- **function RK2s(tanim,h)**

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- $N=[10 \ ;5]; N1=N(1); N2=N(2); t=tanim(1); tson=tanim(2); T=t;$

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;
- **N=N+h\*M; t=t+h;**

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;
- N=N+h\*M; t=t+h;
- T=[T;t];N1=[N1;N(1)];N2=[N2;N(2)];



## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;
- N=N+h\*M; t=t+h;
- T=[T;t];N1=[N1;N(1)];N2=[N2;N(2)];
- end

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;
- N=N+h\*M; t=t+h;
- T=[T;t];N1=[N1;N(1)];N2=[N2;N(2)];
- end
- plot(T,N1,'.-k'); hold on;plot(T,N2,'.-r');

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;
- N=N+h\*M; t=t+h;
- T=[T;t];N1=[N1;N(1)];N2=[N2;N(2)];
- end
- plot(T,N1,'.-k'); hold on;plot(T,N2,'.-r');
- function yp=F(t,N)

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;
- N=N+h\*M; t=t+h;
- T=[T;t];N1=[N1;N(1)];N2=[N2;N(2)];
- end
- plot(T,N1,'.-k'); hold on;plot(T,N2,'.-r');
- function yp=F(t,N)
- yp=[a\*N(1)-b\*N(1)\*N(2); -c\*N(2)+d\*N(1)\*N(2)];

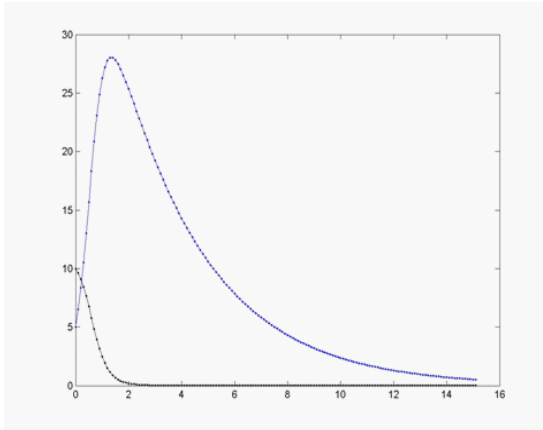
## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;
- N=N+h\*M; t=t+h;
- T=[T;t];N1=[N1;N(1)];N2=[N2;N(2)];
- end
- plot(T,N1,'.-k'); hold on;plot(T,N2,'.-r');
- function yp=F(t,N)
- yp=[a\*N(1)-b\*N(1)\*N(2); -c\*N(2)+d\*N(1)\*N(2)];
- end

## Sistemler için II. Basamaktan Runge-Kutta yöntemi(Kod)

- % Yazılımı: RK2s(tanim,h), tanim=[t1,tson]
- function RK2s(tanim,h)
- N=[10 ;5];N1=N(1);N2=N(2);t=tanim(1);tson=tanim(2);T=t;
- a=0.2;b=0.1;c=0.1;d=0.3;
- while t<tson
- M1= F(t,N); M2= F(t+h,N+h\*f(t,N)); M=(M1+M2)/2;
- N=N+h\*M; t=t+h;
- T=[T;t];N1=[N1;N(1)];N2=[N2;N(2)];
- end
- plot(T,N1,'.-k'); hold on;plot(T,N2,'.-r');
- function yp=F(t,N)
- yp=[a\*N(1)-b\*N(1)\*N(2); -c\*N(2)+d\*N(1)\*N(2)];
- end
- end

- Çözüm eğrilerinden  $N_2(t)$  avcısının  $N_1(t)$  yi tüketmek suretiyle nüfusunu çoğalttığını, daha sonra da tek besin kaynağı olan  $N_1(t)$  nüfusunun tükenmesiyle  $N_2(t)$  nüfusunun da hızla azalarak yok olmaya başladığı görülmektedir, Şekil 2



Şekil 2: Canlı nüfusunun zamana göre değişimi ( $N_1(t)$  (siyah),  $N_2(t)$  (mavi))



- Benzer biçimde (7) için dördüncü mertebeden Runge-Kutta yöntemi(RKIVs) aşağıdaki gibi ifade edilebilir:

$$\begin{aligned}M_1 &= F(t_i, X^{(i)}); \\M_2 &= F(t_i + h/2, X^{(i)} + h/2M_1); \\M_3 &= F(t_i + h/2, X^{(i)} + h/2M_2); \\M_4 &= F(t_i + h, X^{(i)} + hM_3); \\M &= (M_1 + 2M_2 + 2M_3 + M_4)/6; \\X^{(i+1)} &= X^{(i)} + hM, i = 1, 2, \dots\end{aligned}\tag{10}$$

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(örnek)



$$u'' + \alpha u'(u^2 - 1) + u = 0$$

$$u(0) = u_0, u'(0) = u_1$$

başlangıç değer problemi Van der Pol problemi olarak bilinir ve  $u(t)$  nonlineer devrelerde elektrik akımını temsil eder.

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(örnek)



$$\begin{aligned}u'' + \alpha u'(u^2 - 1) + u &= 0 \\u(0) &= u_0, u'(0) = u_1\end{aligned}$$

başlangıç değer problemi Van der Pol problemi olarak bilinir ve  $u(t)$  nonlineer devrelerde elektrik akımını temsil eder.

- $u' = v$  dönüşümü ile

$$\begin{aligned}u' &= v \\v' &= \alpha v(1 - u^2) - u \\u(0) &= u_0, v(0) = u_1\end{aligned}\tag{11}$$

birinci mertebeden nonlineer sistemini elde ederiz.

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(örnek)



$$\begin{aligned}u'' + \alpha u'(u^2 - 1) + u &= 0 \\u(0) &= u_0, u'(0) = u_1\end{aligned}$$

başlangıç değer problemi Van der Pol problemi olarak bilinir ve  $u(t)$  nonlinear devrelerde elektrik akımını temsil eder.

- $u' = v$  dönüşümü ile

$$\begin{aligned}u' &= v \\v' &= \alpha v(1 - u^2) - u \\u(0) &= u_0, v(0) = u_1\end{aligned}\tag{11}$$

birinci mertebeden nonlinear sistemini elde ederiz.

- $u(0) = 2, v(0) = 0.5$  ve  $\alpha = 0.2$  değerleri için  $[0, 15]$  aralığında  $h = 0.1$  adım uzunluğu ile Runge-Kutta(IV) ve/veya MATLAB/Octave ode45 yaklaşımların grafiklerini çizdiriniz.

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- `function RK4s(tanim,h)`

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- $U=[2 ;0.5];u=U(1);v=U(2);$

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- `function RK4s(tanim,h)`
- `U=[2 ;0.5];u=U(1);v=U(2);`
- `t=tanim(1);tson=tanim(2); T=t; alfa=0.2;`

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- `function RK4s(tanim,h)`
- `U=[2 ;0.5];u=U(1);v=U(2);`
- `t=tanim(1);tson=tanim(2); T=t; alfa=0.2;`
- `while t<tson`



# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- U=[2 ;0.5];u=U(1);v=U(2);
- t=tanim(1);tson=tanim(2); T=t; alfa=0.2;
- while t<tson
- **M1=F(t,U);**

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- `function RK4s(tanim,h)`
- `U=[2 ;0.5];u=U(1);v=U(2);`
- `t=tanim(1);tson=tanim(2); T=t; alfa=0.2;`
- `while t<tson`
- `M1=F(t,U);`
- `M2=F(t+h/2,U+h/2*m1);`

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- U=[2 ;0.5];u=U(1);v=U(2);
- t=tanim(1);tson=tanim(2); T=t; alfa=0.2;
- while t<tson
- M1=F(t,U);
- M2=F(t+h/2,U+h/2\*m1);
- **M3=F(t+h/2,U+h/2\*m2);**

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- $U=[2 ;0.5];u=U(1);v=U(2);$
- $t=tanim(1);tson=tanim(2); T=t; alfa=0.2;$
- while  $t < tson$
- $M1=F(t,U);$
- $M2=F(t+h/2,U+h/2*m1);$
- $M3=F(t+h/2,U+h/2*m2);$
- $M4=F(t+h,U+h*m3);$

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- `function RK4s(tanim,h)`
- `U=[2 ;0.5];u=U(1);v=U(2);`
- `t=tanim(1);tson=tanim(2); T=t; alfa=0.2;`
- `while t<tson`
- `M1=F(t,U);`
- `M2=F(t+h/2,U+h/2*m1);`
- `M3=F(t+h/2,U+h/2*m2);`
- `M4=F(t+h,U+h*m3);`
- `M=(M1+2*M2+2*M3+M4)/6;`

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- U=[2 ;0.5];u=U(1);v=U(2);
- t=tanim(1);tson=tanim(2); T=t; alfa=0.2;
- while t<tson
- M1=F(t,U);
- M2=F(t+h/2,U+h/2\*m1);
- M3=F(t+h/2,U+h/2\*m2);
- M4=F(t+h,U+h\*m3);
- M=(M1+2\*M2+2\*M3+M4)/6;
- U=U+h\*M; t=t+h;

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- U=[2 ;0.5];u=U(1);v=U(2);
- t=tanim(1);tson=tanim(2); T=t; alfa=0.2;
- while t<tson
- M1=F(t,U);
- M2=F(t+h/2,U+h/2\*m1);
- M3=F(t+h/2,U+h/2\*m2);
- M4=F(t+h,U+h\*m3);
- M=(M1+2\*M2+2\*M3+M4)/6;
- U=U+h\*M; t=t+h;
- T=[T;t];u=[u;U(1)];v=[v;U(2)];

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- U=[2 ;0.5];u=U(1);v=U(2);
- t=tanim(1);tson=tanim(2); T=t; alfa=0.2;
- while t<tson
- M1=F(t,U);
- M2=F(t+h/2,U+h/2\*m1);
- M3=F(t+h/2,U+h/2\*m2);
- M4=F(t+h,U+h\*m3);
- M=(M1+2\*M2+2\*M3+M4)/6;
- U=U+h\*M; t=t+h;
- T=[T;t];u=[u;U(1)];v=[v;U(2)];
- end



# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- `function RK4s(tanim,h)`
- `U=[2 ;0.5];u=U(1);v=U(2);`
- `t=tanim(1);tson=tanim(2); T=t; alfa=0.2;`
- `while t<tson`
- `M1=F(t,U);`
- `M2=F(t+h/2,U+h/2*m1);`
- `M3=F(t+h/2,U+h/2*m2);`
- `M4=F(t+h,U+h*m3);`
- `M=(M1+2*M2+2*M3+M4)/6;`
- `U=U+h*M; t=t+h;`
- `T=[T;t];u=[u;U(1)];v=[v;U(2)];`
- `end`
- `plot(T,u,'.-k');hold on; plot(T,v,'.-b');`

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- `function RK4s(tanim,h)`
- `U=[2 ;0.5];u=U(1);v=U(2);`
- `t=tanim(1);tson=tanim(2); T=t; alfa=0.2;`
- `while t<tson`
- `M1=F(t,U);`
- `M2=F(t+h/2,U+h/2*m1);`
- `M3=F(t+h/2,U+h/2*m2);`
- `M4=F(t+h,U+h*m3);`
- `M=(M1+2*M2+2*M3+M4)/6;`
- `U=U+h*M; t=t+h;`
- `T=[T;t];u=[u;U(1)];v=[v;U(2)];`
- `end`
- `plot(T,u,'-k');hold on; plot(T,v,'-b');`
- `function Yp=F(t,U)`

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- U=[2 ;0.5];u=U(1);v=U(2);
- t=tanim(1);tson=tanim(2); T=t; alfa=0.2;
- while t<tson
- M1=F(t,U);
- M2=F(t+h/2,U+h/2\*m1);
- M3=F(t+h/2,U+h/2\*m2);
- M4=F(t+h,U+h\*m3);
- M=(M1+2\*M2+2\*M3+M4)/6;
- U=U+h\*M; t=t+h;
- T=[T;t];u=[u;U(1)];v=[v;U(2)];
- end
- plot(T,u,'-k');hold on; plot(T,v,'-b');
- function Yp=F(t,U)
- Yp=[U(2);alfa\*U(2)\*(1-U(1)^2)-U(1)];

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- U=[2 ;0.5];u=U(1);v=U(2);
- t=tanim(1);tson=tanim(2); T=t; alfa=0.2;
- while t<tson
- M1=F(t,U);
- M2=F(t+h/2,U+h/2\*m1);
- M3=F(t+h/2,U+h/2\*m2);
- M4=F(t+h,U+h\*m3);
- M=(M1+2\*M2+2\*M3+M4)/6;
- U=U+h\*M; t=t+h;
- T=[T;t];u=[u;U(1)];v=[v;U(2)];
- end
- plot(T,u,'-k');hold on; plot(T,v,'-b');
- function Yp=F(t,U)
- Yp=[U(2);alfa\*U(2)\*(1-U(1)^2)-U(1)];
- end

# Sistemler için IV. Basamaktan Runge-Kutta yöntemi(Kod)

- function RK4s(tanim,h)
- U=[2 ;0.5];u=U(1);v=U(2);
- t=tanim(1);tson=tanim(2); T=t; alfa=0.2;
- while t<tson
- M1=F(t,U);
- M2=F(t+h/2,U+h/2\*m1);
- M3=F(t+h/2,U+h/2\*m2);
- M4=F(t+h,U+h\*m3);
- M=(M1+2\*M2+2\*M3+M4)/6;
- U=U+h\*M; t=t+h;
- T=[T;t];u=[u;U(1)];v=[v;U(2)];
- end
- plot(T,u,'.-k');hold on; plot(T,v,'.-b');
- function Yp=F(t,U)
- Yp=[U(2);alfa\*U(2)\*(1-U(1)^2)-U(1)];
- end
- end

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.

# Sistemler için MATLAB/OCTAVE ODE çözücüleri

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.

# Sistemler için MATLAB/OCTAVE ODE çözücüleri

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.
- `function testode()`



# Sistemler için MATLAB/OCTAVE ODE çözücüleri

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.
- function testode()
- **b=15;alfa=0.2;**

# Sistemler için MATLAB/OCTAVE ODE çözücülerini

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.
- function testode()  
• b=15;alfa=0.2;  
•  $Y_0=[2 \ 0.5];$

# Sistemler için MATLAB/OCTAVE ODE çözücülerini

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.
- function testode()  
• b=15;alfa=0.2;  
• Y0=[2 0.5];  
• tt=0:0.1:b;

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.
- `function testode()`
- `b=15;alfa=0.2;`
- `Y0=[2 0.5];`
- `tt=0:0.1:b;`
- `fvdP = @(T,Y) [Y(2); alfa*(1 - Y(1)^2) * Y(2) - Y(1)];`

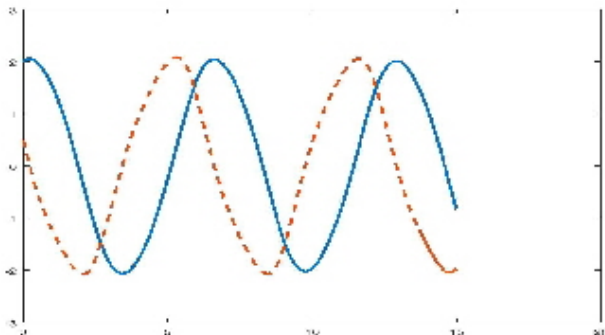
- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.
- function testode()
  - b=15;alfa=0.2;
  - Y0=[2 0.5];
  - tt=0:0.1:b;
  - fvdP = @(T,Y) [Y(2); alfa\*(1 - Y(1)^2) \* Y(2) - Y(1)];
  - [T,Y] = ode45(fvdP, tt, Y0);

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.
- `function testode()`
- `b=15;alfa=0.2;`
- `Y0=[2 0.5];`
- `tt=0:0.1:b;`
- `fvdp = @(T,Y) [Y(2); alfa*(1 - Y(1)^2) * Y(2) - Y(1)];`
- `[T,Y] = ode45(fvdp, tt, Y0);`
- `plot(T,Y(:,1),'-g','linewidth',2); hold on;`

- Octave diferensiyel denklem sistemlerinin çözümü için MATLAB benzeri ode23 ve ode45 gibi alternatif sunar.
- Van der Pol problemi için testode isimli uygulaması aşağıda sunulmaktadır.
- `function testode()`
- `b=15;alfa=0.2;`
- `Y0=[2 0.5];`
- `tt=0:0.1:b;`
- `fvdp = @(T,Y) [Y(2); alfa*(1 - Y(1)^2) * Y(2) - Y(1)];`
- `[T,Y] = ode45(fvdp, tt, Y0);`
- `plot(T,Y(:,1),'-g','linewidth',2); hold on;`
- `plot(T,Y(:,2),'-r','linewidth',2);`

# Sistemler için ODE çözücüler

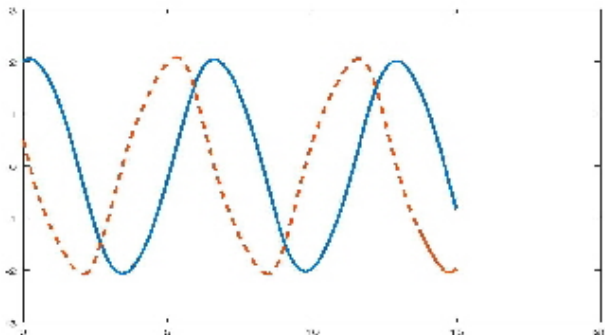
- $u(0) = 2, v(0) = 0.5$  ve  $\alpha = 0.2$  değerleri için  $[0, 15]$  aralığında  $h = 0.1$  ile ODE45 yaklaşımları





# Sistemler için ODE çözücüler

- $u(0) = 2, v(0) = 0.5$  ve  $\alpha = 0.2$  değerleri için  $[0, 15]$  aralığında  $h = 0.1$  ile ODE45 yaklaşımları



- $u$  (düz çizgi) ve  $v = du/dt$  (kesikli çizgi)

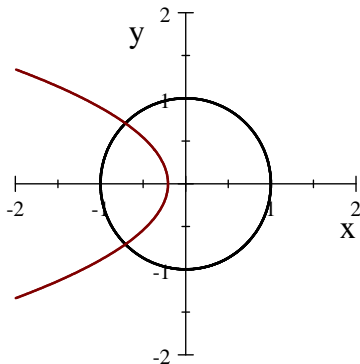
## Örnek 3

$$\begin{aligned}\frac{dx}{dt} &= x^2 + y^2 - 1 & (12) \\ \frac{dy}{dt} &= x + y^2 + (\sqrt{2} - 1)/2 \\ x(0) &= -3, y(0) = -4\end{aligned}$$

*başlangıç değer problemi verilmiş olsun.*

# Sistemler için uygulamalar

- Verilen problemin denge noktalarının Şekil 3 de gösterilen eğrilerin arakesit noktaları olarak  $(-1/\sqrt{2}, -1/\sqrt{2})$  ve  $(-1/\sqrt{2}, 1/\sqrt{2})$  olduğunu gösteriniz.



Şekil 3: Örnek 3 e ait denge noktaları

- $(-1/\sqrt{2}, -1/\sqrt{2})$  noktasının asimtotik kararlı(denge noktasında Jacobien reel kısımları negatif özdeğerlere sahip) ve  $(-1/\sqrt{2}, 1/\sqrt{2})$  noktasının ise eyer noktası(denge noktasında Jacobien farklı işaretli reel özdeğerlere sahip) olduğunu gösteriniz.
- $(-1/\sqrt{2}, -1/\sqrt{2})$  komşuluğunda verilen probleme karşılık gelen lineer problemi belirleyiniz.
- Verilen problem ve karşılık gelen lineer problemin  $h = 0.1$  adım uzunluğu ile  $[0, 4]$  aralığındaki yaklaşımları RKII yardımıyla elde ederek çözüm bileşenlerinin grafiğini( $x = x(t), y = y(t)$ ) ve faz diyagramını( $(x(t), y(t))$  grafiğini) çizdiriniz.

# Sistemler için uygulamalar

- $x^2 + y^2 - 1 = 0$  ve  $x + y^2 + (\sqrt{2} - 1)/2 = 0$  denklem sistemini çözerek,  $(-1/\sqrt{2}, -1/\sqrt{2})$  ve  $(-1/\sqrt{2}, 1/\sqrt{2})$  reel çözümlerini elde ederiz.
- $J(x, y) = \begin{bmatrix} 2x & 2y \\ 1 & 2y \end{bmatrix}$  olup,  $(-1/\sqrt{2}, -1/\sqrt{2})$  noktasında

$$J = \begin{bmatrix} -\sqrt{2} & -\sqrt{2} \\ 1 & -\sqrt{2} \end{bmatrix}$$

elde ederiz ve Jacobien matrisinin özdeğerlerini

$\lambda_{1,2} = -1.4142 \pm 1.1892i$  olarak elde ederiz. O halde

$(-1/\sqrt{2}, -1/\sqrt{2})$  denge noktası asimtotik kararlı bir denge noktasıdır. Öte yandan  $(-1/\sqrt{2}, 1/\sqrt{2})$  noktasında

$$J = \begin{bmatrix} -\sqrt{2} & \sqrt{2} \\ 1 & \sqrt{2} \end{bmatrix}$$

olup,  $\lambda_1 = -1.8478, \lambda_2 = 1.8478$  dir. Reel ve farklı işaretli özdeğerler  $(-1/\sqrt{2}, 1/\sqrt{2})$  noktasının eyer noktası olduğunu ifade eder.

- $(x^*, y^*) = (-1/\sqrt{2}, -1/\sqrt{2})$  asimtotik kararlı denge noktasında

$$f(x, y) \approx f(x^*, y^*) + \frac{\partial f}{\partial x} \Big|_{(x^*, y^*)} (x - x^*) + \frac{\partial f}{\partial y} \Big|_{(x^*, y^*)} (y - x^*)$$

$$g(x, y) \approx g(x^*, y^*) + \frac{\partial g}{\partial x} \Big|_{(x^*, y^*)} (x - x^*) + \frac{\partial g}{\partial y} \Big|_{(x^*, y^*)} (y - x^*)$$

lineer yaklaşımları kullanılarak

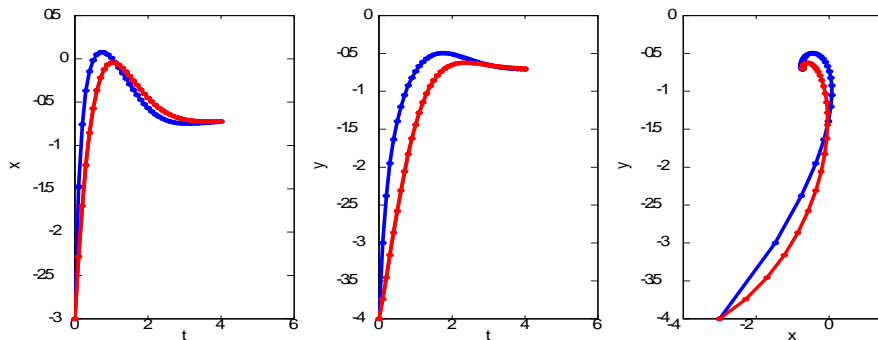


$$\begin{aligned} \frac{dx}{dt} &= -\sqrt{2}(x - x^*) - \sqrt{2}(y - y^*) \\ \frac{dy}{dt} &= (x - x^*) - \sqrt{2}(y - y^*) \\ x(0) &= -3, y(0) = -4 \end{aligned} \tag{13}$$



lineer yaklaşım modelini elde ederiz.

# Sistemler için uygulamalar

- (12) ve karşılık gelen (13) lineer problemin RKII yöntemiyle  $[0, 4]$  aralığında ve  $h = 0.1$  adım uzunluğu ile elde edilen  $x(t)$  ve  $y(t)$  çözüm yaklaşımları Şekil 4 da sunulmaktadır.



Şekil 4: Nonlineer problem(mavi) ve karşılık gelen lineer modelin çözümleri(kırmızı)

-  Granwille Sewell, The numerical solution of ordinary and partial differential equations, World Scientific.
-  Erhan Coşkun, Diferensiyel Denklemler için Fark yöntemleri, KTÜ, 2018, Ders notu.