

FIZ220__EST__Odev__05__LineerCebirUygulamalari

June 8, 2020

1 Ödev: 5

1.1 FİZ220 - Bilgisayar Programlama II | 01/06/2020

Lineer Cebir Uygulamaları

Son gönderim tarihi: 7 Haziran Pazar, 23:59

Gönderim şekli: FIZ220__Odev__05__Grup__#.ipynb isimli jupyter ipynb formatında dosyayı ödev sayfasından göndermek suretiyle

Gönderecek kişi: Grup temsilcisi

Dr. Emre S. Taşcı, emre.tasci@hacettepe.edu.tr

Fizik Mühendisliği Bölümü

Hacettepe Üniversitesi

1.2 1. Soru: Eğri aslında aynı eğri

Ölçeği ayarlayıp, aşağıdaki iki grafikte verilen yayı aynı uzunluğa getirin. Eğrileri döndürmeniz ya da değerler üzerinde değişim yapmanız istenmemektedir: grafiklerden birinde (ya da dilerseniz ikisinde) x ve y eksenlerinin ölçeğini değiştirip, sonuç olarak öyle iki grafik elde edeceksiniz ki, çıktılarını alıp, sadece eğrileri kesip, üst üste koyduğumuzda birebir (/mümkün mertebe) örtüşecek boyutta olacaklar.

```
[2]: import numpy as np
import matplotlib.pyplot as plt

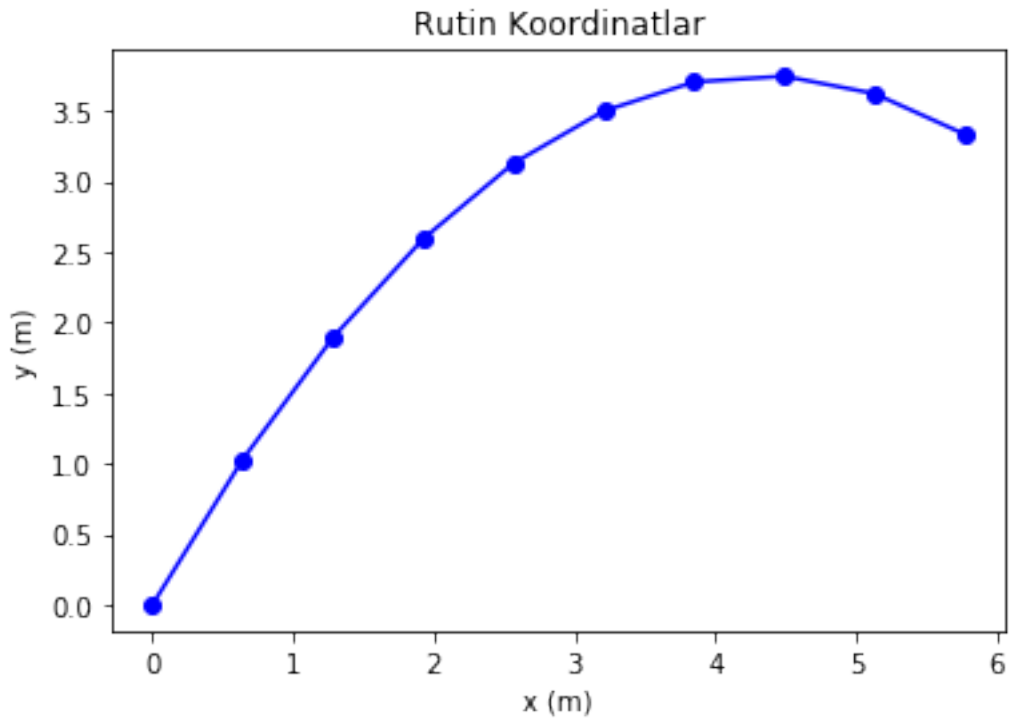
alpha = np.deg2rad(30) # derece -> radyan
beta = np.deg2rad(30) # derece -> radyan
V0 = 10 # m/s
g = 10 # m/s^2

t0 = (2*V0/g)*(np.sin(beta)/np.cos(alpha))

t = np.linspace(0,t0,10)
x = V0*np.cos(alpha+beta)*t
y = V0*np.sin(alpha+beta)*t - 0.5*g*t**2

plt.plot(x,y,"o-b")
```

```
plt.xlabel("x (m)")
plt.ylabel("y (m)")
plt.title("Rutin Koordinatlar")
#plt.xlim(0,8)
#plt.ylim(0,4)
plt.show()
```



```
[3]: import numpy as np
import matplotlib.pyplot as plt

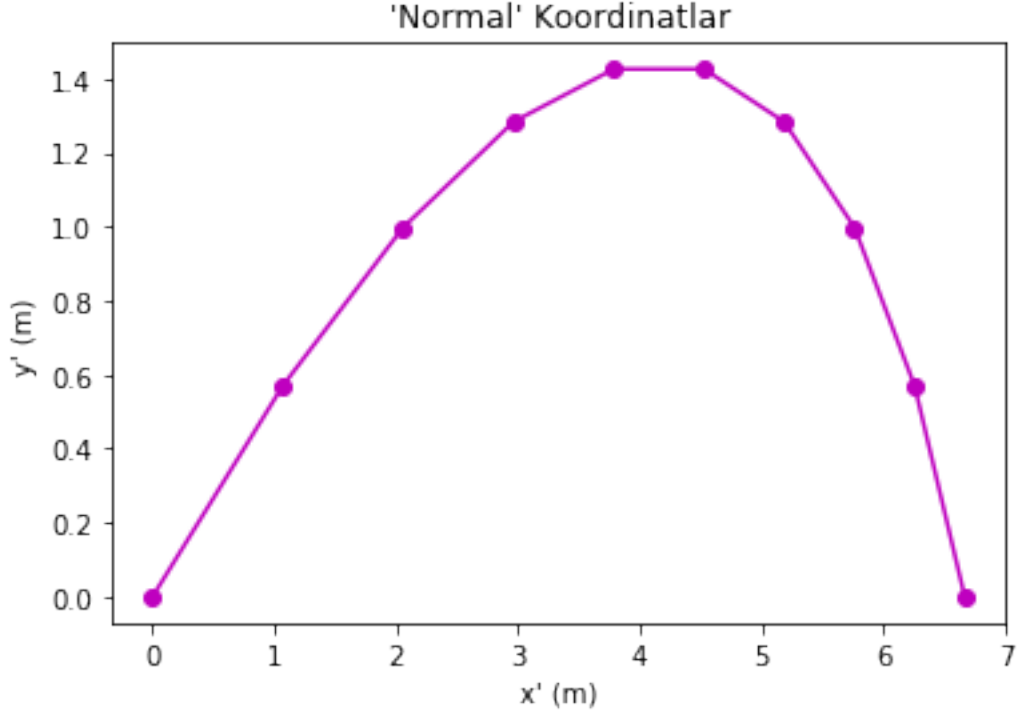
alpha = np.deg2rad(30) # derece -> radyan
beta = np.deg2rad(30) # derece -> radyan
V0 = 10 # m/s
g = 10 # m/s^2

t0 = (2*V0/g)*(np.sin(beta)/np.cos(alpha))

t = np.linspace(0,t0,10)
xp = V0*np.cos(beta)*t - 0.5*g*np.sin(alpha)*t**2
yp = V0*np.sin(beta)*t - 0.5*g*np.cos(alpha)*t**2

plt.plot(xp,yp,"o-m")
plt.xlabel("x' (m)")
```

```
plt.ylabel("y' (m)")
plt.title("'Normal' Koordinatlar")
plt.show()
```



1.3 2. Soru: O öyle değil miydi ki hakikaten?

Bizim bildiğimiz -saatin tersi yönünde döndüren- dönüş matrisi:

$$R_{\alpha} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$$

şeklinde değil miydi (yani “ $-\sin \alpha$ ” sağ üstte; “ $\sin \alpha$ ” sol altta olacak şekilde)?

Ama sistemimizi saatin tersi yönünde döndürmemize rağmen, neden 1. örnek olan eğik düzleme atış probleminde (ve diğer iki örnekte de) işlemi yaparken “ $-\sin \alpha$ ”yı sol altta, “ $\sin \alpha$ ”yı sağ üstte aldık?... (Açıklayın)

1.4 3. Soru: ???

2. örnek olan sürtünmeli eğik düzlemde kayan kütlenin ‘normal’ koordinat sistemindeki hareket denklemleri olan:

$$\mu F_N - mg \sin \theta = m\ddot{x}' F_N - mg \cos \theta = m\ddot{y}' \ddot{y}' = 0$$

üç denklemi lineer denklem seti olarak matris çarpımı ile temsil ederken:

$$\begin{pmatrix} m & 0 & -\mu \\ 0 & m & -1 \\ 0 & 220 & 0 \end{pmatrix} \begin{pmatrix} \ddot{x}' \\ \ddot{y}' \\ F_N \end{pmatrix} = \begin{pmatrix} -mg \sin \theta \\ -mg \cos \theta \\ 0 \end{pmatrix}$$

şeklinde yazdık. En alt satırın orta sütunundaki “220” değeri nereden geldi? Dersimizin koduyla aynı oluşu tesadüf mü?

4. Soru: [Soruları severim ama, cevaplarım ama, sağlama yapmasak?](#)

$$\begin{pmatrix} -(k+K) & K \\ K & -(k+K) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = -m\omega^2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

şeklinde verilen üç yay, iki cisim sistemi denklem takımını normal koordinatlar ifade etmek için özdeğer ve özvektörlerini hesaplatmıştık:

```
[4]: import numpy as np

m = 1 # kg
k = 10 # N/m
K = 30 # N/m

A = np.array([[-(k+K), K], [K, -(k+K)]])

[l,u] = np.linalg.eig(A)
u_inv = np.linalg.inv(u)
D = np.diag(l)

print(l)
print(D)
print(u)
```

```
[-10. -70.]
[[-10.  0.]
 [ 0. -70.]]
[[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]
```

İlgili özdeğer ve özvektörlerin:

$$A\vec{u}_i = \lambda_i\vec{u}_i$$

özdeğer denklemini sağladığını teyit edin.

Bonus: Soruya ait A matrisini, özvektörler matrisinden ve köşegenleştirilmiş formu kullanarak geri üretin.