

Ad-Soyad:

Numara:

**MAT412 BİLGİSAYAR PROGRAMLAMA II FİNAL SINAV SORULARI**

1) 
$$\begin{cases} 2x_1 + 4x_2 - 6x_3 = -4 \\ x_1 + 5x_2 + 3x_3 = 10 \\ x_1 + 3x_2 + 2x_3 = 5 \end{cases}$$

denklem sistemini Gauss Yoketme yöntemi ile çözünüz.

2)  $\int_0^1 e^{-x^2} dx$  integralinin yaklaşık değerini 8 aralık için Simpson yöntemi ile hesaplayan ve sonucu ekrana 6 basamak duyarlılık ile yazdıran bir kod yazınız.

3)

x	1	3	4
y	0.841471	0.141120	-0.756802

Yukarıdaki verilen tablodan yararlanarak  $x = 2$  değerine karşılık gelen  $y$  değerini Kuadratik Enterpolasyon kullanarak hesaplayan ve sonucu ekrana 6 basamak duyarlılık ile yazdıran bir kod yazınız.

4)  $u' = xe^{-u} + \frac{x}{1+x^2}$ ,  $u(0) = 1$  başlangıç değer probleminin  $[0,1]$  aralığındaki çözümünü 0.1 adım uzunluğu ile İkinci Mertebeden Runge Kutta yöntemi ile hesaplayan ve sonuçları ekrana 3 basamak duyarlılık ile yazdıran bir kod yazınız.

5)  $y' = 1 + \frac{y}{t}$ ,  $y(1) = 2$  başlangıç değer probleminin  $[1,2]$  aralığındaki çözümünü 0.25 adım uzunluğu ile İkinci Mertebeden Taylor Serisi yöntemi ile hesaplayan ve sonuçları ekrana 4 basamak duyarlılık ile yazdıran bir kod yazınız.

Süre: 90 dakikadır. Başarılar.

21.05.2019

Prof. Dr. Vedat Suat ERTÜRK

# MAT 412 BİLGİSAYAR PROGRAMLAMA II FINAL CEVAP ANAHTARI

1) Sistemin genişletilmiş katsayılar matrisi

$$\left[ \begin{array}{ccc|c} 2 & 4 & -6 & -4 \\ 1 & 5 & 3 & 10 \\ 1 & 3 & 2 & 5 \end{array} \right]$$

şekindedir. İkinci ve üçüncü satıra birinci satırın  $-1/2$  katı ilave edilirse

$$\left[ \begin{array}{ccc|c} 2 & 4 & -6 & -4 \\ 0 & 3 & 6 & 12 \\ 0 & 1 & 5 & 7 \end{array} \right]$$

elde edilir. Üçüncü satıra ikinci satırın  $-1/3$  katı ilave edilerek

$$\left[ \begin{array}{ccc|c} 2 & 4 & -6 & -4 \\ 0 & 3 & 6 & 12 \\ 0 & 0 & 3 & 3 \end{array} \right]$$

elde edilir. Son satırdan dolayı  $3x_3 = 3 \Rightarrow x_3 = 1$  olacağından  
 $3x_2 + 6x_3 = 12 \Rightarrow x_2 = 2$  olur. Son olarak  $2x_1 + 4x_2 - 6x_3 = -4$  den  $x_1 = -3$  bulunur

2) Program

```
import numpy as np, math
n=input('aralık sayısını giriniz... ');
n=int(n)
if n%2!=0:
    print('aralık sayısı çift olmalı!')
    n=input('aralık sayısını tekrar giriniz... ');
a,b=input('sırasıyla a ve b değerlerini \
    'virgül ile ayırarak giriniz:').split(', ');
a,b=int(a),int(b)
h=(b-a)/n
def f(x):
    'integral içindeki fonksiyonu tanımlayınız'
    return math.exp(-x*x)
t1=0
for i in np.arange(1,n,2):
    t1=t1+f(a+i*h)
```

t2=0

for j in np.arange(2,n-1,2):

t2=t2+f(a+i\*h)

print('integralin yaklaşık değeri: ', '%.6f' % ((h/3)\*(f(a)+f(b)+4\*t1+2\*t2)))

aralık sayısını giriniz... 8

sırasıyla a ve b değerlerini virgül ile ayırarak giriniz: 0,1

### 3) Program

import math

x0, x1, x2, x, y0, y1, y2 = input('x0, x1, x2, x, y0, y1 ve y2 değerlerini sırasıyla '\n virgül ile ayırarak giriniz: ').split(',')

x0, x1, x2, x, y0, y1, y2 = int(x0), int(x1), int(x2), int(x), float(y0), float(y1), float(y2)

a0 = y0

a1 = (y1 - y0) / (x1 - x0)

a2 = ((y2 - y0) / (x2 - x0) - (y1 - y0) / (x1 - x0)) / (x2 - x1)

p = a0 + a1 \* (x - x0) + a2 \* (x - x0) \* (x - x1)

print('%.6f' % p)

x0, x1, x2, x, y0, y1 ve y2 değerlerini sırasıyla virgül ile ayırarak giriniz: 1,3,4,  
2, 0.841471, 0.141120, -0.756802

### 4) Program

import numpy as np, math

x = np.empty(20)

u = np.empty(20)

def f(x,u):

'f(x,u) fonksiyonu tanımlanıyor'

return x \* math.exp(-u) + x / (1 + x \* x)

n = 10

a = 0

b = 1

h = (b - a) / n

x[0] = 0

u[0] = 1

```

print('x', 'u')
print('...', '...')
print('%0.1f' % x[0], '%.3f' % u[0])
j=1
while j<=n:
    x[j]=x[0]+j*h
    p1=h*f(x[j-1], u[j-1])
    p2=h*f(x[j-1]+h, u[j-1]+p1)
    u[j]=u[j-1]+(p1+p2)/2
    print('%0.1f' % x[j], '%.3f' % u[j])
    j+=1

```

### 5) Program

```

import numpy as np, math
t=np.empty(20)
y=np.empty(20)
def f(t,y):
    'f(t,y) fonksiyonu tanımlanıyor'
    return 1+y/t
def ft(t,y):
    'f(t,y) nin t ye göre türevi tanımlanıyor'
    return -y/(t*t)
def fy(t,y):
    'f(t,y) nin y ye göre türevi tanımlanıyor'
    return 1/t

```

```

t[0]=1
y[0]=2
h=0.25
n=4
print('t', 'y')
print('...', '...')
for i in np.arange(0, n+1):
    t[i+1]=t[0]+i*h
    y[i+1]=y[i]+(t[i+1]-t[i])*f(t[i], y[i]) + ((t[i+1]-t[i])**2/2)*\
(f_t(t[i], y[i]) + f_y(t[i], y[i])*f(t[i], y[i]))
    print('%0.1f' % t[i+1], '%.4f' % y[i+1])

```