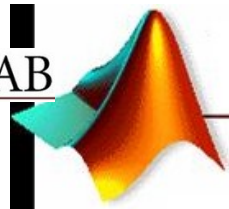
MATLAB

MATLAB, bilim adamları ve mühendislere, Fortran ve C gibi gelenekselleşmiş dillerde program yazmaksızın, matrislere dayalı problemleri çözmede kullanılmak üzere bir sayısal hesaplama kütüphanesi sunmak amacıyla, **MAT**ris **LAB**oratuvarı (**MAT**rix **LAB**oratory) programı olarak tasarlanmıştır. Fakat daha sonra, *Optimization Toolbox* ve *Control System Toolbox* gibi bazı toolbox'lar eklenerek geliştirilmiştir. (Matlab, 1970'lerin sonunda Cleve Moler tarafından yazılmıştır. Cleve Moler aynı zamanda "The Mathworks" firmasının da kurucusudur.)

MATLAB bir yorumlayıcıdır (interpreter); yani sonuç, daha ziyade el tipi hesap makinelerine benzer tarzda ekranda yazılı bir metin olarak alınır. Neticede diğer dillerde olduğu gibi "derleyici"ye (compiler) ihtiyaç yoktur; fakat programlamaya izin vermesinden dolayı da güçlü bir paket programdır.

MATLAB Programının Tipik Kullanım Alanları

- **Matematiksel hesaplama (nümerik ve sembolik ??) işlemleri,**
- **Algoritma geliştirme ve kod yazma (programlama),**
- **Lineer cebir, istatistik, Fourier Analizi, filtreleme, optimizasyon, sayısal integrasyon vb. konularda hazir matematik fonksiyonlara ulaşım,**
- **2D ve 3D grafiklerinin çizimi,**
- **Modelleme ve simülasyon,**
- **Grafiksel arayüz oluşturma,**
- **Veri analizi ve kontrolü.**



MATLAB Ortamının Tanıtımı

MATLAB (Ara Yüz Tanıtımı)

Geliştirme Ortamı

a) **Başlatma Penceresi (Launch Pad)**

b) **Command Window (Komut Penceresi)**

c) **Workspace**

d) **Current Directory**

e) **Command History**

1. Temel Bilgiler

Komut Penceresi: **MATLAB** ile iletişim kurulan ana penceredir. **MATLAB** yorumlayıcısı, kullanıcıdan gelecek komutları kabul etmeye hazır olduğunu gösteren **" >> "** biçiminde bir ileti görüntüler. Örneğin, 4*25+6*52+2*99 gibi basit bir matematiksel işlemi yapmak için

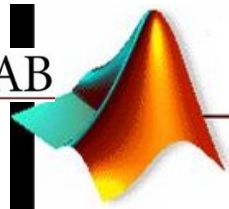
>> 4*25+6*52+2*99 ifadesini yazıp **ENTER** tuşuna basarız.

ans=

610

Komut satırında yanlışların düzeltilmesi: Klavyede yer alan ok tuşları komut satırında yapılan yanlışlıkların düzeltilmesine imkan verir. Bunlar yukarı **"↑"** aşağı **"↓"** sol **"←"** sağ **"→"**. Yukarı tuşu kullanılarak bir önceki satır tekrar görüntülenerek sağ ve sol tuşları ile yanlış yazılı yere kursör taşınarak düzenleme gerçekleştirilir.

Sonucun Ekranda Görüntülenmesini Gizleme: Bir deyim yazıp, **ENTER** tuşuna basınca sonuçlar ekranda otomatik olarak görüntülenir. Buna karşılık, deyimin sonuna **" ; "** ilave edilecek olursak, bu deyim ile yapılan hesaplamalar ekranda görüntülenmez.



2. MATLAB DEĞİŞKENLERİ VE KURALLAR

MATLAB'de diğer programlama dillerinden farklı olarak deyimler tümüyle matrisleri kapsar.

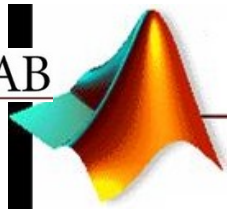
Deyim Oluşturma Gurupları: Değişkenler, rakamlar, işletmenler ve fonksiyonlardır. **MATLAB** deyimleri ise komut ortamında yazılan ve komut satirindan girilen herseydir.

Değişkenler: Deyimler içerisinde sayısal değerlerin yerini alan ifadelerdir. MATLAB bir değişken ile karşı karşıya geldiğinde, otomatik olarak bu değişken oluşturulur ve yeteri kadar bellek ayrılır. Eğer değişken daha önceden tanımlı ise **MATLAB** onun içeriğini değiştirir ve gerekirse yeni bellek ayırır.

Örneğin, >>x =50 komut satırından
işletildiğinde "x" adı altında bir değişken oluşturur ve 50 değeri bu değişkene atanir.

Diğer bilgisayar dillerinde olduğu gibi **MATLAB**'in de değişken isimleri konusunda bazı kuralları vardır.

MATLAB



Bazı Kurallar…….

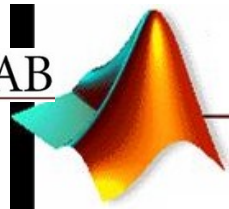
1- Değişken isimleri küçük, büyük harf kullanımına duyarlıdır. Buna göre aynı anlama gelen fakat farklı yazılan “orta”, “Orta”, “orTa” ve “ORTA” kelimeleri **MATLAB** için farklı değişkenlerdir.

2- Değişken isimleri en fazla 63 karakter içerebilir. Bundan fazla olanlar dikkate alınmaz.

3- Değişken isimleri daima bir harf ile başlamalıdır. Bunu harfler, rakamlar veya alt çizgiler ”_” izleyebilir. ***Noktalama işaretleri*** değişken isminde kullanılmaz. Çünkü bunların pek çoğunun **MATLAB** için bir anlamı vardır. Ayrıca değişken adlarında küçük veya büyük “ç ı ö ü ğ ş” Türkçe karakterler kullanılmaz.

Rakamlar: MATLAB rakamlar için artı veya eksi işareti ve tercihli ondalık noktası ile birlikte alışagelmiş ondalık işaretler sistemi kullanır. Kuvvet belirlemek için “e” harfi kullanılır. Sanal rakamlar son takı olarak “ i ” veya “ j ” harfini kullanır.

3, -100, 0.0005, 9.53564 1.456e10, 2.5e-5, 10i, -3.4j, 3e5i



İşletmenler (Operatörler) : MATLAB, deyimler içerisinde alışageldik aritmetik işletmenler ve öncelik kuralları kullanır.

İşlem	Sembol	Örnek
Toplama, a+b	+	2+3
Çıkarma, a-b	-	5-2
Çarpma, a*b	*	3*4
Bölme, a/b	/	14/7
Üs alma, a^b	^	2^3
Parantez a*(b+c)	( )	2*(3+5)

Aritmetik işlemlerde öncelik hakkı: Tek bir matematiksel deyim içinde birden fazla işlem bir arada bulunabildiğine göre hangi işlemin öncelik hakkına sahip olduğunun bilinmesi yerinde olacaktır. Aşağıda, **MATLAB**'de kullanılan işlemlerde, işlemlerin öncelik listesi verilmiştir.

Öncelik	İşlem
1.	Parantez
2.	Üst alma, soldan sağa doğru
3.	Çarpma ve bölme, soldan sağa doğru
4.	Toplama ve çıkarma, soldan sağa doğru

Fonksiyon: MATLAB, sin, abs, sqrt, ve log gibi önemli fonksiyonları da içine alan çok sayıda matematik fonksiyon sağlar. Bu fonksiyonların bazıları aşağıdaki Tablo'da listelenmiştir.

Fonksiyon	Sembol	Örnek
Sinüs	sin	sin(pi)
Kosinüs	cos	cos(pi)
Tanjant	tan	tan(pi)
Arksinüs	asin	asin(0)
Arkkosinüs	acos	acos(0)
Arktanjant	atan	atan(1)

Fonksiyon	Sembol	Örnek
Eksponensiyal, e^x	exp	exp(2)
Doğal logaritma $\ln(x)$	log	log(10)
10 tabanlı logaritma	log10	log10(10)
Kare kök, $\sqrt{x}$	sqrt	sqrt(25)
Mutlak değer, $\|x\|$	abs	abs(3)

sinus, cosinus, tanjant, arksinus, arkkosinus ve arktanjant fonksiyonlari aci degerlerini radyan cinsinden arguman olarak alirlar. Derece cinsinden calismak isterseniz (Derece/180)=(Radyan/pi) formulunden faydalaniniz. sin(30)(Derece)=sin(30*pi/180)

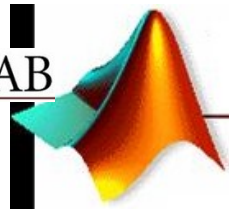
Örnekler:

Matematiksel yazılım	Bilgisayarda Yazılımı
ab-c+d-6+da	a*b-c+d-6+d*a
$b+c^3-d/8-b^2c$	b+c^3-d/8-b^2*c
$\frac{a}{b}+\sqrt{c^3}-bd^2+\frac{2ab}{b^2-4ac}$	a/b+c^(3/2)-b*d^2+(2*a*b)/(b^2-4*a*c)
$a+\frac{(b+c^2).3f^3}{d+\frac{e-f}{3a}}$	???

Uygulama :

$$x=\frac{a+\frac{c}{b-a}}{\sqrt[3]{\frac{1+c^2-\sqrt{a+b^4}}{b+\sqrt[5]{cd^3}}}}+(ac)^3-\frac{\sqrt{a^4}}{\frac{2}{a\sqrt{a}}}$$

işlemini **MATLAB** dilinde kodlayınız.



MATLAB' DE DEĞİŞKEN TİPLERİ

Herşey matris?

- Matris, vektör (sütun ya da satır), string (karakter dizisi), skaler

Double:

Skaler veya vektörlerden oluşan sayısal değişkenleri ifade eder.

Char:

Tek bir karakter veya karakter grubundan oluşan skaler veya dizileri ifade eder.

MATLAB' DE DEĞİŞKEN ATAMALARI

Eşitlik İfadeleri ile Değişken Atamaları

Bu şekildeki bir değişken atamasının genel hali,

>> **değişken = değer**

şeklindedir. Burada "değişken", herhangi bir karakter veya karakter grubu olabilir. "değer" ise, herhangi bir **matematiksel ifade**, **bir karakter dizisi**, **bir sabit**, **bir matris** veya bunların birden fazlasının matematiksel işlemler ile oluşturulmuş kombinasyonları olabilir.

Değişken Örnekleri

- Değişkenler, **skaler**, **vektör**, **matris** veya metin (karakter dizisi) (**string**) olabilir.
- Değişken örnekleri:

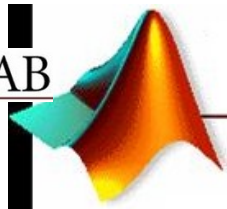
  - **skaler**=1; **kuvvet**=-3.2e3; **rasyonel**=22/5;

  - **metin**='Deniz'

  - **vektor1**=[1 2 3] %satir vektoru

  - **vektor2**=[1;2;3] %sutun vektoru

  - **matris**=[1 2 3;-1 0 1]



Örnek:

Skaler atama

X = 3
A = 5-5i
B = A / 5

!!!!! ans adlı özel değişkenin varlığına dikkat ediniz!!!!

Bir değişkene bir değer atadıktan sonra aynı değişkene farklı bir değer atamak, o değişkenin **ilk değerinin silinmesine**, söz konusu değişkenin bundan sonraki işlemlerde **yeni değeri** ile işlem görmesine neden olmaktadır.

Örnek:

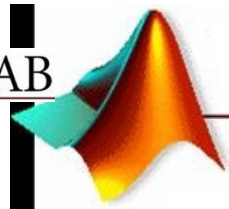
```
>> a=5
>>b=a+2
   b =
        7
```

```
>>a=10
>>b=a+2
   b =
       12
```

```
>>4+4
   ans =
        8
```

```
>>ans*4
   ans =
       32
```

MATLAB



Sayı Formatı

- Bir işlem sonucu, varsayılan (default) olarak 4 ondalık ile gösterilir.
- Sayı gösteriminde hane sayısı format fonksiyonu ile değiştirilir.
  >> format xxx
- **format** veya **format short** : 5 rakamlı (4 ondalık)
- **format bank**: İki ondalıklı sayı
- **format long**: 15 ondalık
- **format rat**: Ondalık sayıları rasyonel sayı olarak gösterir.
- **UYGULAMA 1 :**
  >>a=22/5 i yukarıdaki formatlar için test ediniz.

  UYGULAMA 2 : Ayrıca aşağıdaki komutlar dizisini çalıştırınız.
  >>format rat
  >>a=22/5
  >>b=1/3
  >>c=a+b
  >>format

MATLAB

ÖZEL DEĞERLER VE KALICI DEĞİŞKENLER

MATLAB'in yapısında önceden tanımlanmış, kullanılacakları zaman tekrar tanımlanmalarına gerek olmayan ve herhangi bir anda kullanılmaya hazır bazı özel değerler bulunur.

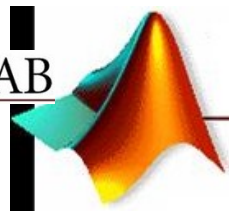
Örneğin:

yarıçapı 2 birim olan bir kürenin alanını

```
>>  r=2;
>>  alan =  4   *  pi   *  r^2
>> alan =

            50.2655
```

```
>> date
ans =
19-Mar-2003
```

```
>> fix(clock)
ans =
     2008     3     4    12    22    45
```



CLC, CLEAR, WHO ve WHOS KOMUTLARI

clc komutu komut penceresi ekranını temizler.

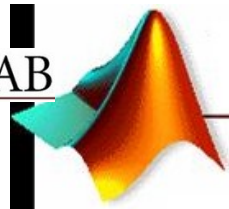
clear komutu butun değişkenleri ve fonksiyonları bellekten siler. (Bütün değişkenleri çalışma alanından (workspace) çıkarır.)

who komutu ile sadece değişken adlarını; whos komutu ile de değişkenlerimizin özelliklerini görebiliriz.

Lütfen Aşağıdaki Komutları Komut Penceresinden Çalıştırın:

```
>>help clc
>>help clear
>>help who
>>help whos
```

Daha detaylı yardım için MATLAB programınız açıkken F1 tuşuna basınız ve gelen yardım penceresinde "Search for" boşluğuna yada "Search" sekmesini tıkladığınızda önünüze gelen boş alana yardım almak istediğiniz komutu yazıp ENTER tuşuna basınız.



BİR DEĞİŞKENE DIŞARDAN BİR DEĞER ATAMAK

“ input " fonksiyonu, komut penceresinde kullanıcıdan bir değişkene bir değer girmesini isteyen bir komut görüntüler ve kullanıcının bu değeri girmesini bekler.

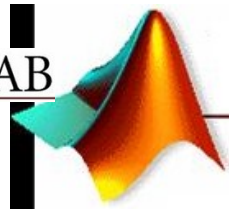
X=input('Bir deger giriniz= ');

```
>> x=input('x degerini giriniz= ')

x degerini giriniz=  10
x =

    10
```

Dışardan karakter dizisi (string) okumak için

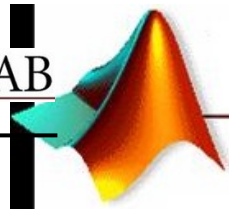
pal=input('Adınızı Giriniz = ', 's');

MATLAB



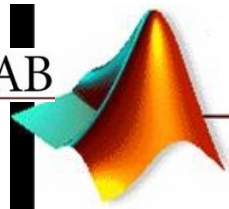
Uygulama :

Dışarıdan input komutuyla girilen 3 sayının ortalamasını bulan bir MATLAB programını komut penceresi yardımıyla yazınız.



MATLAB'DE DİZİLER

- Dizi (array), en genel tanımı ile nümerik veya metinsel değerler topluluğudur. (veri yapısı – data structure) MATLAB'de herşey bir dizi olarak işleme konur ve dizi en temel veri elemanıdır.
  - –Reel ile kompleks sayıları ifade eden çift kat veya nümerik diziler (double veya numeric array)
  - –Metin ifade eden diziler, karakter dizileri (char array)



MATLAB'DE DİZİLER (devam)

- Bir nümerik dizi, skaler, vektör veya matris olabilir ve tüm nümerik diziler double array formatındadır.
- 1x1 dizisi, bir skaler (scalar) gösterir. (a=3, b=-6.5)
- mx1 veya 1xn dizisi, bir vektör (vector) gösterir.
- mxn veya nxm dizisi,bir matris (matrix) gösterir.
- Bu çerçevede 1x1 dizisi sabit matris veya tek elemanlı matris, nx1dizisi sütun matrisi ve 1xn dizisi ise satır matrisi olarak da düşünülebilir.
- Bir dizinin eleman sayısı, satır ile sütun sayısının çarpımıdır.

MATLAB

MATLAB'DE DİZİLER (devam)

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}$$ 3x2 matrix → 6 eleman

b=[1 2 3 4] 1x4 array → 4 eleman, **satır vektörü**

$$c=\begin{bmatrix}1\\ 3\\ 5\end{bmatrix}$$ 3x1 array → 3 eleman, **sütun vetörü**

A(2,2)=4 b(3)=3 c(1)=1

Satır # Sütun #

MATLAB

VEKTÖRLER

1xn veya mx1 şeklinde tek boyutlu dizi olan vektörleri oluşturmanın iki temel yolu vardır:

i) Direkt olarak (**[...]** ile**)**

- **Satır vektörleri: >> f = [f1 f2 f3 ... fn] veya**
  >> f = [f1,f2,f3, ...,fn]
- **Sütun vektörleri: >> f = [f1; f2; f3; ...; fn]**

ii) Eşit aralıklı elemanlar kullanarak (**:** ile)

- **f = İlkDeğer : DeğişimMiktarı : SonDeğer**

Değişim miktarı belirtilmezse İlkDeger'den sonra 1'er artım olacağını ifade eder.

ÖRNEK : a=1:10 veya b=1:5:25

MATLAB

MATRİSLER

$$F = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1n} \\ F_{21} & F_{22} & \cdots & F_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ F_{m1} & F_{m2} & \cdots & F_{mn} \end{bmatrix}_{mxn}$$

Matrisleri oluşturmanın en temel yolu […] kullanmaktır. mxn boyutundaki bir matrisin genel formatı:

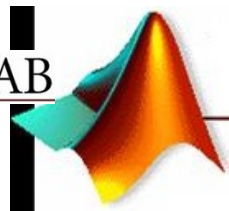
- **F = [F11 F12 ... F1n ; F21 F22 ... F2n ; ..... ; Fm1 Fm2 ... Fmn]**

veya

- **F = [F11,F12, ... ,F1n ; F21,F22, ... ,F2n; ..... ;Fm1, Fm2, ... , Fmn]**

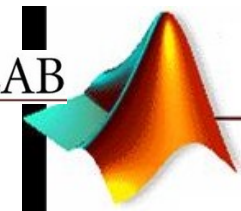
DİZİ EDİTÖRÜ (ARRAY EDITOR)

Çalışma alanında (workspace) herhangi bir değişkenin üzerini çift tıklarsanız Microsoft Excel tablosuna (spreadsheet) benzer bir pencere açılır sizin için. Buna dizi editörü ya da array editor denir. Değiskeninizin elemanlarını bu pencere yardımıyla da modifiye edebilirsiniz.



FONKSİYONLAR

- **Fonksiyonlar da bilgisayar programlarıdır. Bilgisayarlar yardımıyla çözülmeye çalışılan problemler fonksiyonlar sayesinde yönetilmesi daha kolay, küçük parçacıklara bölünürler. Her fonksiyon:**
  - **Kendine özgü bir isme sahiptir.**
  - **Kendine, üzerinde işlem yapacağı bir argüman (parametre) ya da argümanlar (parametreler) alır.**
  - **GENELLİKLE geriye bir değer döndürür. (skaler, vektör ya da matris.)**
  - **Örnek : rand(n,m)**



MATLAB'DE DİZİLER (devam)

Hem vektörler hem de matrisler yardimci (utility) fonksiyonlar (**zeros**, **ones** ve **rand**) kullanılarak da oluşturulabilir:

- **zeros(1,n) veya zeros(n,1)**
- **zeros(n) veya zeros(n,n)**
- **zeros(n,m)**
- **ones(1,n) veya ones(n,1)**
- **ones(n) veya ones(n,n)**
- **ones(n,m)**
- **rand(n,m)**
- **rand(n) veya rand(n,n)**
- **round(rand(n,m))**
- **fix(rand(n,m)) (Nasıl bir çıktı ????)**

MATLAB

MATLAB'DE RASTGELE SAYI ÜRETİMİ

- rand(n,m) fonksiyonu MATLAB'de nxm boyutunda ve elemanlari rastgele sayilar olan bir matris olusturur. Olusturulan bu rastgele sayilar 0 ile 1 arasindadir ve duzgun (uniformly) dagilimlidir.
- Rastgele sayilardan olusan bu matrisin tum elemanlarini k gibi bir sayi ile carparak sayilari 0 ile k araligina cekebiliriz.
- Ondalik sayilardan olusmus bir matrisin elemanlarini yuvarlayip tam sayi yapmak icin round fonksiyonunu kullanabiliriz.
- a=rand(1,10) ??
- b=round(40*rand(5,3)) ?? c= 40*round(rand(5,3)) ??
- d=round(10+40*rand(5,3)) ?? e=10+40*round(rand(5,3))??
- f=round(50+250*rand(3,4)) ?? g=round(1000*rand(1)) ??

DİZİLERE UYGULANABİLEN BAZI FONKSİYONLAR

- length(a): a vektörünün eleman sayısı
- sum(a): a vektörünün elemanlarının toplamı
- max(a): a vektörünün maksimum elemani
- min(a): a vektörünün minimum elemani
- size(b): b matrisinin boyutu (satır ve sütun)
- size(b,1): b matrisinin satır sayisi
- size(b,2): b matrisinin sütun sayisi
- sum(b): b matrisinin sütun başına eleman toplamı (!!! Satır Vektörü!!!)
- sum(sum(b)): ?????????
  - a(n): a vektorunun n. eleman değeri
  - b(n,:): b matrisinin sadece n. satırı
  - b(:,n): b matrisinin sadece n. sütunu
  - b': b matrisinin transpozesi yani devriği
- det(b): b matrisinin determinantı !!!! (Kare matrisler için)!!!!
- inv(b) : b matrisinin tersi !!!!(Kare matrisler için)!!!!
- diag(b): b matrisinin ana diagonel (çapraz) elemanları
- Elemanter işlemler eleman eleman işlem demektir ve . işleci kullanılır:
  Çarpma: .* , Bölme: ./ , Üs alma: .^
  Örnek : Bir matrisin tüm elemanlarının karesini alma (a^2=a*a) ????

MATLAB

UYGULAMA

- **Soru:** **MATLAB'in size fonksiyonunu kullanarak bir a vektörünün boyutunu nasıl bulursunuz?**

- **Cevap:** **max(size(a)) = length(a)**

MATLAB

MATLAB'DE PROGRAMLAMA

Bilgisayar programlamasında genel olarak belirli kalıp ve kurallara uyulur. Bir bilgisayar yazılımının oluşturulmasında genel olarak aşağıda sıralanan prosedüre uyulur:

- Problemin tanımı (Ne-Niçin)
- Çözüm yönteminin ve adımlarının belirlenmesi (algoritma: akış şemaları veya pseudo-kod)
- Kodlama (Programlama diline çevirme)
- Sınama (Test) (Programı çalıştırma)
- Güncelleştirme

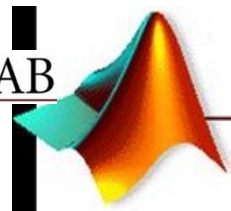
MATLAB

MATLAB'DE PROGRAMLAMA

- MATLAB'de programlama en genel olarak iki yolla yapılır:
  - Komut satırında (in-line) programlama
  - m-dosyalarıyla (m-files) programlama

- m-dosyalarının da iki türü vardır:
  - Düzyazı (script) m-dosyaları
  - Fonksiyon (function) m-dosyaları

- m dosyaları oluşturabilmek için bir metin editörüne ihtiyaç vardır.

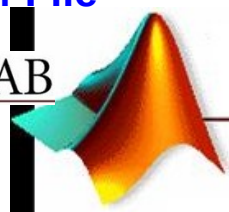


MATLAB' de Programlama

M-dosyası (M-File)

Bir senaryo dosyası (script file) özel bir görevi yerine getirmek için gerekli MATLAB komutlarının saklandığı bir metin programıdır. Başka bir ifadeyle; bir hesaplamayı gerçekleştirmek için yazılacak bir çok komut dizisi, komut penceresinden tek tek girmek yerine bir dosyada saklanır daha sonra bu dosya çalıştırılarak bu komutlar icra edilir. Bu dosyaların MATLAB'in çalıştığı dizinde (current directory) "dosya_adı.m" uzantısıyla saklanmaları gerekir. DOSYA ADLARININ İÇİNDE TÜRKÇE KARAKTERLER VE BOŞLUKLAR KULLANILAMAZ.

Senaryo dosyalarının (M-dosyalarının) oluşturulması ve yazılması için MATLAB bir metin hazırlayıcısı (text editor) sunmaktadır. Bu senaryo dosyaları Windows'da Notepad gibi herhangi bir metin hazırlayıcısında da yazılabilirler. MATLAB metin hazırlayıcısı ya "current directory" penceresinde bos bir alana sag tiklayip "New, M-File" ile ya da kısaca "File" menüsünden "New, M-File" ibaresini seçerek etkin hale getirilebilir.



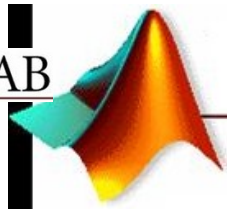
M-dosyalarının Gerekliliği:

- ✓Değişken sayısının fazla olması durumunda
- ✓Akış diyagramlarının uygulanmasında
- ✓Programdaki değişikliklerin kolayca yapılmasında

Örnek:

Dışarıdan klavye yoluyla girilen dairenin yarıçapına göre alanini ve çevresini hesaplayan bir MATLAB programini "alanVeCevreHesapla.m" adinda bir M-dosyasi icinde yaziniz ve komut penceresinden "alanVeCevreHesapla" komutunu yazarak calistiriniz.

```
clc;
clear;
r = input('Dairenin yarıçapını giriniz =  ');
alan=pi*r^2;
cevre=2*pi*r;
alan,cevre
```



MATEMATİKSEL VE MANTIKSAL OPERATÖRLER

Program dallandıran bir çok yapıda, işlemler, sonucun "**doğru**“ **(TRUE)** (**1**) veya **"yanlış“ (FALSE)** (**0**) olması durumları ile kontrol edilir. MATLAB'de '**doğru**' veya '**yanlış**' ifadeleri ile sonuçlar üreten **iki çeşit** operatör vardır. Bunlar **matematiksel** ve **mantıksal** operatörlerdir. (Bir onceki derste **aritmetik** operatorleri gormustuk. Dolayisiyla MATLAB'de su ana kadar 3 tip operator (isletmen) gormus olduk.)

Matematiksel Karşılaştırma Operatörleri

Bu operatörler iki değişkenin değer bakımından karşılaştırmasını yaparlar ve üretilen sonucun **doğru** (**1**) veya **yanlış** (**0**) durumuna göre sonuçlar üretirler.

Genel kullanımları **a1** **işlem** **a2** şeklindedir. Burada a1 ve a2, *aritmetik değerler*, değişkenler veya karakter dizileri olabilir, "**işlem**" ise, sözkonusu matematiksel kıyaslama operatörlerinden biri olabilir. Eğer a1 ve a2 arasındaki ilişki operatörün belirttiği şekilde ise **sonuç 1** değerini alır. Eğer operatörün belirttiği durumdan farklı bir durum söz konusu ise **sonuç 0** değerini alır.

==	Eşittir	5**==**6	→	0
~=	Eşit değildir	5**~=**6	→	1
>	Büyüktür	5**>**6	→	0
>=	Büyük veya eşittir	5**>=**6	→	0
<	Küçüktür	5**<**6	→	1
<=	Küçük veya eşittir	5**<=**6	→	1

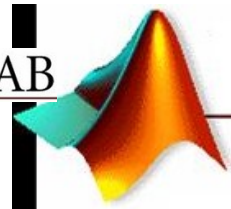
Eşitlik durumlarında verilen işaret iki adet eşittir "**==**" işaretinden oluşur. Oysa değişken atamalarında kullandığımız eşittir "**=**" bir tanedir. Bu ikisi birbirlerinden farklı operatörlerdir. "**==**" operatörü, ***kıyaslama durumlarında*** kullanılır ve mantıksal bir sonuç üretir. "Eşit ise", "eşit midir?" şeklindeki durumlarda kullanılır. "**=**" işareti ise, bir ***değişkene bir değer atamada (atama operatoru)*** kullanılır, örneğin **MATLAB** komut penceresinde; 3**=**5 yazdığımızda; program hata üretir. Oysa 3**==**5 yazdığımızda bu ***" 3, 5'e eşit midir?*** " anlamına gelir, kıyaslama yanlıştır ve **MATLAB** bu durum için "0" cevabını üretir. Yeni başlayanlar için bir karşılaştırma durumunda tek eşittir "**=**" işareti kullanmak, sık yapılan bir hatadır.

```
>> 3==5

ans =

     0
```

```
>> 3 = 5
       ??? 3=5
           !
Error: The expression to the left of the equals
sign
       is not a valid target for an assignment.
```

```
>>  x=0;
>>  y=sin(pi);
```

şeklinde yapılan iki değişken atamasını takiben yapılan x**==**y karşılaştırmasının sonucu olarak **1** cevabı beklenirken, **0** cevabı alınır. **MATLAB**, bu iki değeri farklı olarak algılamıştır. Çünkü MATLAB'de sin(pi) değeri 0'a eşit değildir, sin(pi), yuvarlamadaki farklılıktan ötürü 1.2246e-016 değerine sahiptir ve **0**'a eşit değildir. Yani teorik olarak birbirine eşit iki değer, aslında farklı sayılara tekabül etmektedir.



```
>>x=0;
>>y=sin(pi);
>>x==y
            ans  = 0
```

Sık yapılabilecek diğer bir hata da ***karşılaştırma*** operatörlerinin ***aritmetik*** operatörlerden **daha sonra** değerlendirildikleri durumunu ihmal etmektir. Yani parantezlerden yararlanılmadığı durumlarda bile aritmetik işlemler, **öncelikle** yapılır.

2+8 > 8+3

(2+8) > (8+3)

Bu iki durumda da **MATLAB**'in üreteceği cevap **0**'dır.

Mantıksal Operatörler

Bu operatörler, bir veya iki mantıksal anlamı olan ve mantıksal bir sonuç üreten operatörlerdir. Üç tane çiftli mantık operatörü vardır. Bunlar; "**AND**", "**OR**" ve "**XOR**" dur. Diğer bir mantık operatörü tekli yapıya sahip olan "**NOT**" operatörüdür. Çiftli yapıdaki operatörlerin genel kullanımı **a işlem b** şeklinde iken tekli bir operatör olan "**NOT**" genel kullanımı **işlem a** şeklindedir. Burada **a** ve **b** ***değişkenler*** iken **işlem**, aşağıdaki mantık operatörlerinden biridir, **a** ile **b**'nin arasındaki ilişki, operatörün belirttiği şekilde ise sonuç **1**, eğer değilse sonuç **0** olur.

Operatörler	Komut karşılığı	İşlevleri
a&b	AND	VE
a|b	OR	VEYA
xor(a,b)	XOR	Özel Veya
~a	NOT	Değil

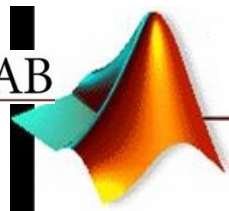
Örnek:

```
>> k=4; m=5;
>> (k>6) and (m<8)
??? (k>6) and (m<8)
         |
Error: Unexpected MATLAB expression

>> (k>6) & (m<8)

ans =

     0
```

→ **HATALI YAZIM**

0<=x<9 ifadesinin MATLAB'deki karşılığı:

(0<=x) & (x<9)



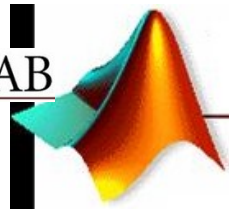
MANTIKSAL OPERATORLERIN DOGRULUK TABLOLARI

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A | B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	xor(A,B)
0	0	0
0	1	1
1	0	1
1	1	0

A	~A
0	1
1	0



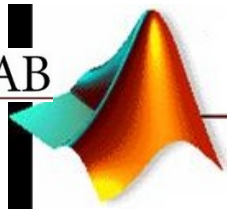
Kontrol Yapıları

IF

Switch

For Loop

While Loops



if Şartlı deyimi (Conditional Statement)

Bir mantıksal ifadeyi kontrol ederek bunun sonucuna göre mümkün seçeneklerden birini icra edebilen bir komuttur.

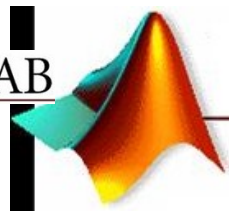
if Deyiminin Üç Farkli Kullanim Sekli vardır.

```
if Şart
   1. işlem
   2. işlem
   3. işlem
      :
end
```

```
if Şart
   1.işlem;
else
   2. işlem
end
```

```
if Şart
...
elseif ...
...
else
...
end
```

Şart dediğimiz şey bir karşılaştırma ifadesi (a>b gibi) ya da bir mantıksal ifadedir (a&b gibi).



Örnek: Girilen (okutulan) x ve y değerlerine göre aşağıdaki sonuc değerini bulan bir MATLAB programını bir M-dosyasi icine yazınız ve komut penceresinden dosya adi ile calistiriniz.

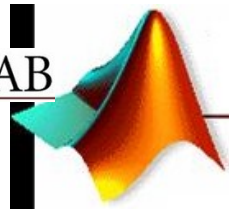
x>y ise sonuc= $\sqrt{(x-y)}$

x=y ise sonuc=$(x+y)^7$

x<y ise sonuc=x*y

ÇÖZÜM:

```
clc;
clear;
x=input('x değeri=');
y=input('y değeri=');
if x>y
        sonuc=sqrt(x-y)
elseif x==y
        sonuc=(x+y)^7
else
        sonuc=x*y
end
```



Uygulama:

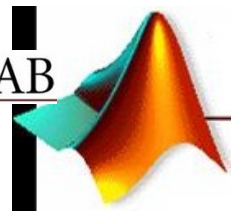
Dışarıdan girilen x değerlerine göre aşağıdaki fonksiyonun değerini hesaplayan bir **MATLAB** programını bir M-dosyası içine yazınız ve komut penceresinden dosya adı ile çalıştırınız.

1 < = x < 10

$$F(x) = x + x^2 + x^3 + x^4 + x^5$$

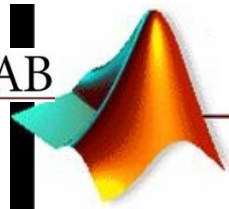
10 < = x

$$F(x) = x + x^2 + \frac{\ln(x)+1}{\log_{10}(x)} + \tan(x) + \sqrt{x} + \frac{3 \cdot x + 5}{x^4}$$



disp VEYA fprintf KOMUTLARIYLA EKRANA MESAJ YAZDIRMA

Daha önceki derslerimizde input komutu ile klavye üzerinden MATLAB programlama ortamına veri girmeyi öğrenmiştik. Bu dersimizde ise disp veya fprintf komutları yardımıyla ekrana nasıl mesaj yazdırıldığını öğreneceğiz.



disp KOMUTUYLA EKRANA MESAJ YAZDIRMA

disp ile Metinsel Çıkış

```
>>disp('Bu bir metin ciktisidir.')
```

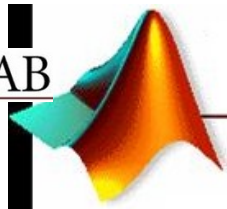
disp ile Nümerik Değişken Çıkışı

```
>>disp(numerik degisken)
```

disp ile Metinle Beraber Nümerik Degişken Çıkışı

```
>>disp(['Bu bir metin ciktisidir.' num2str(numerik degisken)])
```

ÖNEMLİ NOT : Bu slaytta "numerik degisken" dediğimiz şey bir skaler, bir vektör ya da bir matris olabilir.



fprintf KOMUTUYLA EKRANA MESAJ YAZDIRMA

GENEL KULLANIM

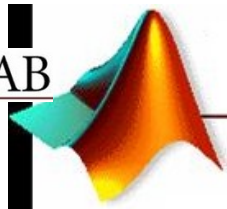
fprintf('ekrana basılacak açıklama %X \n', deger)

Burada %X Kısmında Kullanabileceğimiz Seçenekler :

%c : degerin tek bir karakter olduğunu gösterir.
%s : degerin bir string olduğunu gösterir.
%d : degerin bir tam sayı olduğunu gösterir.
%f : degerin ondalıklı bir sayı olduğunu gösterir.
%e : degeri 10'un kuvveti üstel olarak gösterir.
%g : degeri yazılabilecek en kısa formda gösterir.
(Gereksiz sıfırlar atılır mesela.)

Diğer yandan :

\n : degeri ekrana yazdıktan sonra imleci bir satır atlatır.
\t : degeri ekrana yazdıktan sonra imleci bir TAB kadar atlatır.
\'': ekrana tek tırnak işareti basar.

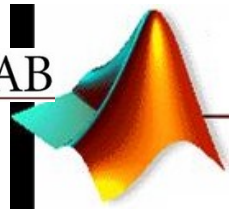


ÖRNEK

Girilen iki sayının oranını bulan ve payda sıfır girildiğinde ekrana "uzgunum, sifira bolum hatasi var." mesajı yazdıran program.

$$oran = \frac{pay}{payda}$$

```
clc;
clear;

pay = input('payi giriniz: ');
payda = input('paydayi giriniz: ');

if   payda==0
        disp('uzgunum, sifira bolum hatasi var.');
else
        oran=pay/payda;
        fprintf('oran = %f \n',oran);
end
```



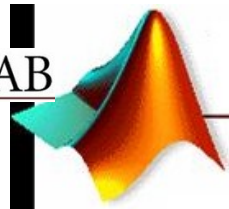
DÖNGÜLER (LOOPS)

Bir çok uygulamada bazı işlemlerin tekrar tekrar gerçekleştirilmesi gerekir. Bilgisayar programlama dillerinde, bu tür işlemleri çok sayıda tekrar etme imkanı sağlayan yapılara **ÇEVRİM**, **DÖNGÜ** veya **LOOP** denir.

- Çevrim, bir tekrarlı işlem yapısıdır.

✓ **Çevrimdeki işlem sayısını önceden belirleyerek ve bu sayıya ulaşıp ulaşmadığını bir sayaç ile denetleyerek gerçekleştirilen çevrim yapıları (for döngüsü)**

✓ **Çevrimin sona ermesini bir koşula bağlı olarak kontrol eden çevrim yapıları (while döngüsü)**



DÖNGÜLER (devam)

for Döngüsü

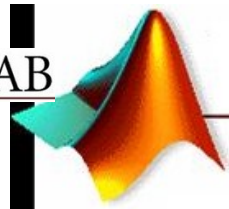
Bir for (için) döngünün genel formatı

```
for döngü değişkeni=ifade
    deyimler
end
```

while Döngüsü

Bir while (süresince veya iken) döngüsünün genel formatı

```
while ifade
    deyimler
end
```



for DÖNGÜSÜ

for ***döngü değişkeni*** **=** ***başlangıç*****:*****artış miktarı*****:*****bitiş***

komutlar

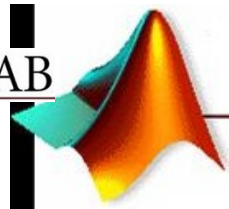
end

for ***döngü değişkeni*** **=** ***başlangıç*****:*****bitiş***

komutlar

end

artış miktarı otomatik olarak 1 atanır.

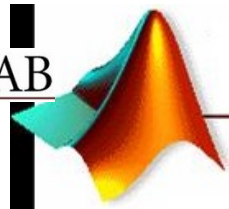


Örnek: 1 ile klavyeden girilen herhangi bir sayı arasındaki sayıların toplamını ve çarpımını hesaplayarak ekrana basan bir MATLAB programı yazınız.

```
clc;
clear;
n=input('lutfen bir sayi giriniz= ');
toplam=0; % toplamada etkisiz eleman
carpim=1; % carpmada etkisiz eleman
for i=1:n
        toplam=toplam+i ;
        carpim=carpim*i ;
end
toplam % Gauss Teoremi mi??????
carpim % n sayisinin faktoriyeli mi?????
```

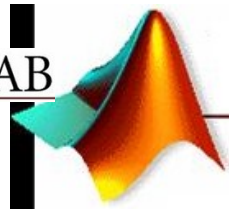
UYGULAMA

```
fprintf(‘for dongusu testi \n');
for i = 4:-1:1
        fprintf(‘for dongusu icindeyiz: i = %d \n',i);
end
fprintf('\nfor dongusunun sonu \n');
```



while DÖNGÜSÜ

```
while şart
        ifade_1
        ifade_2

        ifade_n
end
```



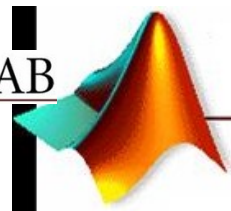
UYGULAMA

Ekrana adınızı ve soyadınızı 10 kez yazan bir MATLAB programını while döngüsü kullanarak oluşturunuz.

```
clc;
clear;

a=1;
while a<=10
        disp('Deniz Dal');
        a=a+1;
end
```

for Döngüsü ile Çözüm

```
clc;
clear;

for i=1:10
        disp('Deniz Dal');
end
```

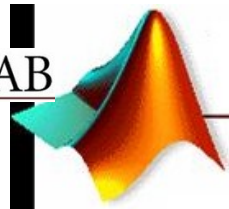


Uygulama

Asağıdaki while döngüsü kaç kere işletilir?

```
devamEt=1;

a=0;

while devamEt
        disp('Deniz Dal');
        a=a+1;
end
```

Yandaki programı ekrana 10 kere "Deniz Dal" yazacak şekilde nasıl modifiye edersiniz??????



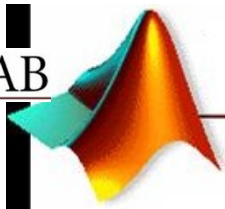
break DEYİMİ

for veya while döngülerinde program akışını kontrol edebilmenin diğer bir yolu da break deyimini kullanmaktır. break deyimini döngü gövdesi içerisinde kullanmak, döngünün durmasını ve döngüden sonra gelen ilk ifade veya komutun işletilmesini sağlar.

Örnek:

```
clc;clear;
for i=1:10
        if  i==4
                break;
        end
        fprintf('i= %d \n', i);
end
disp('dongu break deyimi ile sonlandirildi.');
```

EKRAN ÇIKTISI ???



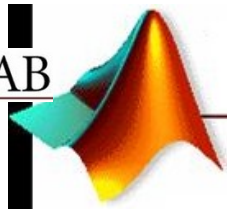
continue DEYİMİ

SADECE for döngülerinde, program akışı ayrıca continue deyimi ile kontrol edilir. continue deyimi işletildiğinde sayaç değişkeni bir sonraki değerine artırılır, continue deyiminden sonraki bütün herşey ihmal edilir ve program for döngüsünün ilk deyiminden itibaren tekrar çalışmaya başlar.

Örnek:

```
clc;clear;
for i=1:10
        if  i==4
                continue;
        end
        fprintf('i= %d \n', i);
end
disp('dongunun isleyisi continue deyimi ile degistirildi.');
```

EKRAN ÇIKTISI ???



BİR SATIR VEKTÖRÜNÜ BİLGİ DEPOLAMAK İÇİN KULLANMA

```
A=[ ]; %Baslangicta Bos
```

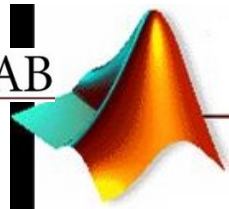
1. METOD :

```
A=[A x]; %A=[x]
A=[A y]; %A=[x y]
.
.
```

2. METOD :

```
A(1)=x; %A=[x]
A(2)=y; %A=[x y]
.
.
```

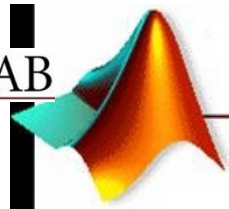
Başlangıçta boş olan bir satır vektörünün içine sırasıyla x ve y elemanları ekleniyor.



İçiçe for Döngüleri

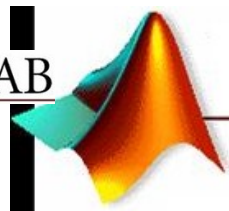
```
clc;
clear;
carpim=[ ];  % Bos Vektor
for i = 1:5
        toplam = 0;
        for j = 1:5
                toplam = toplam + j;
        end
        x = toplam * i;
        carpim=[carpim x];
end
carpim
```

EKRAN ÇIKTISI ???



Örnek: **Çarpım tablosunu ekrana basan bir MATLAB programı yazınız.**

```
clc;
clear;
for a=1:10
        for b=1:10
                carpim=a*b;
                fprintf('%d * %d = %d \n', a, b, carpim);
        end
end
```



ÖRNEK

Yarıçap değerleri 1,2,3,4,5 için bir kürenin hacmini ($H=4/3\pi r^3$) hesaplayan bir MATLAB programı yazınız.

elemanter yöntem:

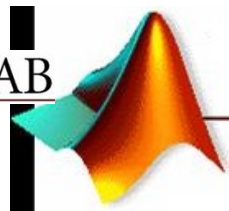
```
r=1:5;
hacim=(4/3)*pi*r.^3;
disp([r;hacim])
```

for yöntemi:

```
for r=1:5
   hacim=(4/3)*pi*r^3;
   disp([r,hacim])
end
```

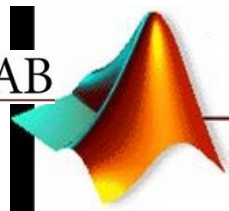
while yöntemi:

```
r=1;
while r<=5
   hacim=(4/3)*pi*r^3;
   disp([r,hacim])
   r=r+1;
end
```



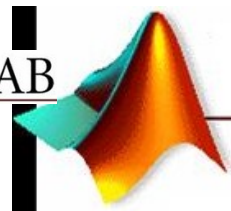
Örnek: Girilen ay'a göre gün sayısını hesaplayan MATLAB programı.

```
clc;
clear;

ay=input('Hangi ayin gun sayisini ogrenmek istiyorsunuz (1-12)=  ' );

if ay==1 | ay==3 | ay ==5 | ay==7 | ay==8 | ay==10 | ay==12
        disp('Bu Ay 31 gunden olusur.');
elseif  ay==2
        disp('Bu Ay 28 gunden olusur.');
else
        disp('Bu Ay 30 gunden olusur.');
end
```



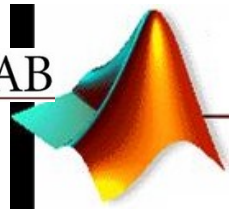
switch Şartlı Deyimi

```
switch(durum)
        case{durum1}
                işlemler
        case{durum2}
                işlemler
        .
        .
        otherwise          % istege bagli
end
```



Örnek: 1 ile 10 arasında girilen bir sayının tek ya da çift olduğunu ekrana basan bir MATLAB programını **switch** deyimini kullanarak yazınız.

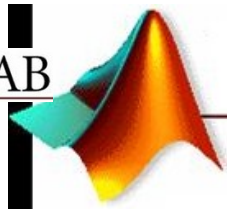
```
clc;clear;
sayi=input('1 ile 10 arasinda bir sayi giriniz= ');
switch(sayi)
        case{1,3,5,7,9}
                disp('Bu sayi tektir.');
        case{2,4,6,8,10}
                disp('Bu sayi cifttir.');
        otherwise
                disp('Bu sayi 1 ile 10 araliginin disinda.');
end
```



UYGULAMA

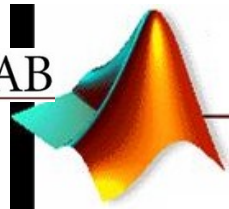
Girilen ay numarasına göre, o ayın gün sayısını veren MATLAB programını **switch** yapısını kullanarak yazınız.

```
clc;clear;
ay = input('Bir ay numarasi giriniz (1-12)= ');
switch(ay)
    case{1,3,5,7,8,10,12}
        disp('Bu Ay 31 gunden olusur.');
    case{2}
        disp('Bu Ay 28 gunden olusur.');
    case{4,6,9,11}
        disp('Bu Ay 30 gunden olusur.');
    otherwise
        disp('Yanlis bir ay no girdiniz.');
end
```



HATA AYIKLAMA (DEBUGGING)

M dosyası editörünün hata ayıklamaya (debugging) yarayan fonksiyonlarını ve calışma alanı (workspace) penceresini kullanarak çalışan programınızdaki (!! BEKLEDİĞİNİZ GİBİ ÇALIŞMAYAN !!) hataları bulabilirsiniz. Bunun için herhangi bir program satırına kesme noktası (breakpoint) koymalı ve kodunuzu satır satır işletmelisiniz. (Derste bu konuyu özetleyen canlı bir uygulama yapılacak.)

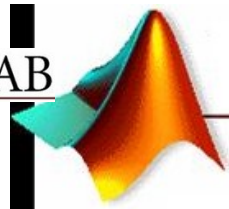


HATA AYIKLAMA (DEBUGGING)

Hata Ayıklama (Debugging) modundan komut satırında return komutunu işleterek çıkabilirsiniz. Ayrica M dosyasının içindeki kesme noktasını üzerini tıklayarak kaldırmalısınız.

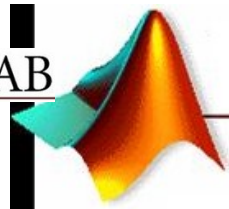
K>>return

>>



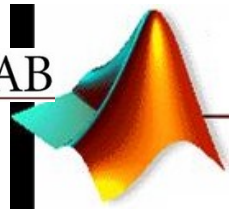
UYGULAMA

İçi rastgele oluşturulmuş tam sayılarla dolu (1 ile 50 arasında) 20 elemanlı bir satır vektörünün en küçük ve en büyük elemanını bularak ekrana basan bir MATLAB programı yazınız. **(min ve max fonksiyonlarini kullanmiyoruz ☺ Kendi min ve max fonksiyonlarımızı yazmaya çalışıyoruz.)**



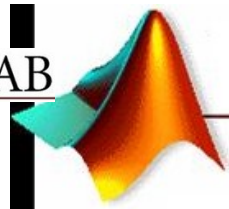
ÇÖZÜM

```
clc;clear;
A=round(1+49*rand(1,20));
%Dizinin En Kucuk ve En Buyuk Elemaninin Dizinin
%Ilk Elemani Oldugunu Farzet
enKucuk=A(1);
enBuyuk=A(1);
for i=2:length(A)
    if A(i)<enKucuk
       enKucuk=A(i);
    end
    if A(i)>enBuyuk
        enBuyuk=A(i);
    end
end
A
disp(['Bu Dizinin En Kucuk Elemani : ' num2str(enKucuk)]);
disp(['Bu Dizinin En Buyuk Elemani : ' num2str(enBuyuk)]);
```



Bilgisayar Programlama

(4. Hafta)



return Komutu

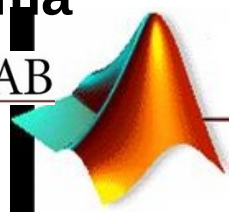
- Yazdığınız MATLAB programını herhangi bir anda (programın normalde sona erdiği noktanın haricinde - early termination) sona erdirmek için return komutunu kullanabilirsiniz.

- Bu işlem için break komutunu KULLANMAYINIZ.

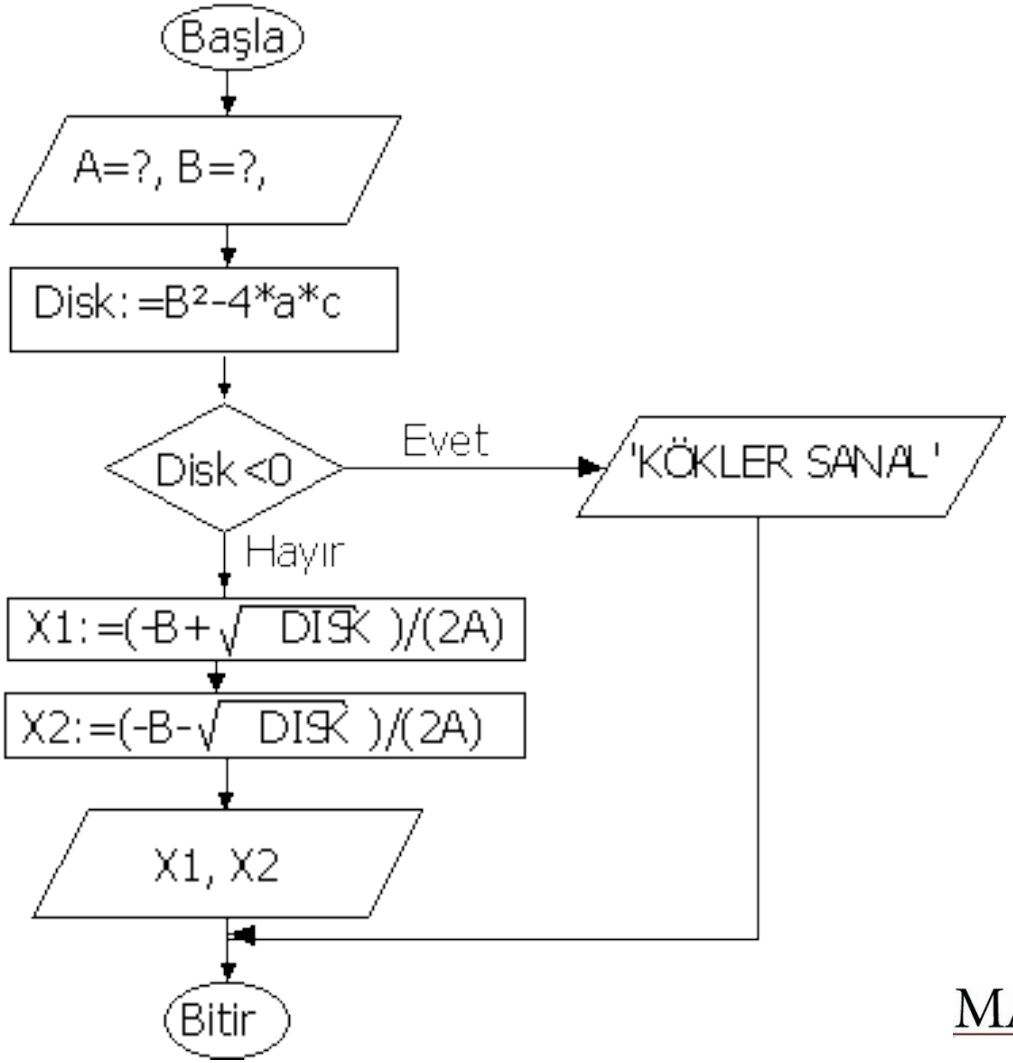
PROGRAMINIZI ANİ SONLANDIRMAK İSTEDİĞİNİZ YER BİR DÖNGÜNÜN İÇİ İSE NE OLUR?????????

- break komutunu sadece for ve while döngüleri içinde kullanınız.

- Eğer return komutu ana program içerisinde kullanılmışsa, kontrol komut penceresindeki klavyeye geçer. Eğer return komutu bir fonksiyon içerisinde kullanılmışsa, kontrol bu fonksiyonu çağıran ana fonksiyona devredilir. (Fonksiyonları daha sonraki derslerimizde göreceğiz.) Bir önceki dersimizde ise return komutunu komut satırından çalıştırarak hata ayıklama modundan çıkmıştık.



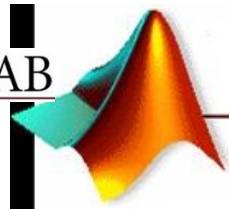
Örnek : $ax^2+bx+c=0$ şeklinde verilen 2. derece denklemin köklerini bulan programın akış diyagramını çiziniz. (Bu denklemin ikinci derece olmadığı uyarısını nasıl verirsiniz?)



MATLAB

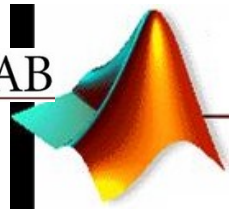
PROBLEMİN MATLAB'DE PROGRAMLANMASI

```
clc;clear;
a=input('a katsayisini giriniz= ');
b=input('b katsayisini giriniz= ');
c=input('c sabitini giriniz= ');
delta=b^2-4*a*c;
if delta<0
        disp('Kokler Sanal');
        return; %Programi Ani Sonlandir
else
        disp('Kokler Reel');
end
X1=(-b+sqrt(delta))/(2*a);
X2=(-b-sqrt(delta))/(2*a);
fprintf('1. Kok : %g \n', X1);
fprintf('2. Kok : %g \n', X2);
```



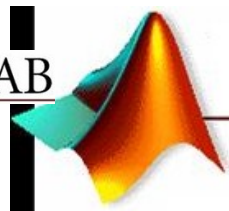
UYGULAMA : **1x5 boyutunda (1 satır ve 5 sütun) bir dizinin (satır vektörü) elemanlarını klavye yoluyla kullanıcıdan alan ve en sonunda bu diziyi ekrana basan MATLAB programı.**

```
clc;clear;
A=[ ];
for i=1:5
        fprintf('A(%d) = ',i);
        A(i)=input(' ');
end
A
```



UYGULAMA : **2x3 boyutunda (2 satır ve 3 sütun) bir dizinin (matris) elemanlarını klavye yoluyla kullanıcıdan alan ve en sonunda bu diziyi ekrana basan MATLAB programı.**

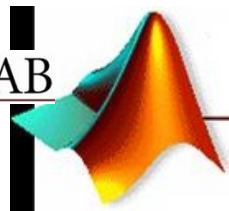
```
clc;clear;
B=[ ];
for i=1:2      % Distaki for satirlar icin
   for j=1:3   % Icteki for sutunlar icin
       fprintf('B(%d,%d) = ',i,j);
       B(i,j)=input(' ');
   end
end
B
```



UYGULAMA

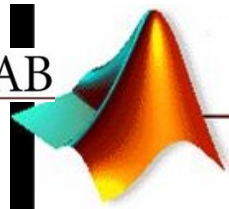
5 öğrencinin herhangi bir dersten aldıkları vize ve final notlarını klavye yoluyla kullanıcıdan alarak herbir öğrencinin ortalamasını hesaplayan ve aşağıdaki formata göre ekrana yazdıran bir MATLAB programı.

Sira	Vize	Final	Ortalama
1. Ogrenci	35	40	37.5
2. Ogrenci	22	95	58.5
:	:	:	:



Çözüm

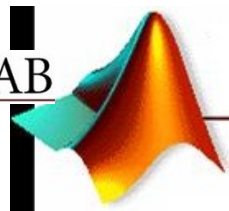
```
clc;clear;
vize=[ ];final=[ ];ort=[ ];
for i=1:5  %Dizilere Disaridan Veri Girisi
   fprintf('%d. Ogrencinin Vize Notu=', i); vize(i)=input(' ');
   fprintf('%d. Ogrencinin Final Notu=', i); final(i)=input(' ');
   ort(i)=(vize(i)+final(i))/2;
end
fprintf('Sira        Vize       Final      Ortalama\n');
fprintf('------      -------    --------   ----------\n');
for i=1:5 %Dizi Elemanlarini Ekrana Yazdirma
   fprintf('%d. Ogrenci     %g         %g         %f\n', i, vize(i),

   final(i),ort(i));
end
```



MATRİSLERDE İŞLEMLER

Matrislerin kendilerine ait cebirleri vardır. Ama biz özellikle matrislerin aşağıda sıralanan işlemleriyle ilgileneceğiz.

- **Matrislerin Bir Skalerle Çarpımı**
- **Matrislerde Toplama**
- **Matrislerde Çıkarma**
- **Matrislerde Çarpma**
- **Matrislerin Transpozu**



Matrislerin Bir Skalerle Çarpımı

UYGULAMA : **Aşağıdaki şekliyle verilen bir A matrisinin her bir elemanını 2 rakamı ile çarpıp bir C matrisine atayan ve en sonunda bu C matrisini ekrana basan MATLAB programı.**

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$$

```
clc;clear;
C=[ ];
A=[2  3 ; 1   4];
[m n]=size(A); %m satir ve n sutun
for i=1:m
    for j=1:n
        C(i,j)=2*A(i,j);
    end
end
A
C
```

MATLAB

Matrislerde Toplama ve Çıkarma İşlemi

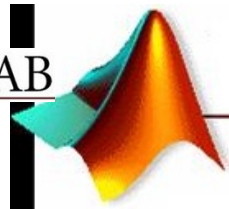
İki matrisin toplanabilmesi veya çıkarılabilmesi için boyutlarının (yani satır ve sütun sayıları) eşit olması gerekir.

$$A=\begin{bmatrix}2 & 3\\ 1 & 4\end{bmatrix} \qquad B=\begin{bmatrix}3 & 6\\ 4 & 5\end{bmatrix}$$

İki matrisin toplamı

$$C=\begin{bmatrix}2 & 3\\ 1 & 4\end{bmatrix}+\begin{bmatrix}3 & 6\\ 4 & 5\end{bmatrix}=\begin{bmatrix}5 & 9\\ 5 & 9\end{bmatrix}$$

```
clc;clear;
A=[2 3; 1 4];
B=[3 6; 4 5];
C=[ ];
for i=1:2
    for j=1:2
        C(i,j)=A(i,j)+B(i,j);
    end
end
A
B
C
```



Matrislerde Çarpma İşlemi

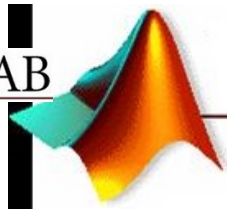
A ve **B** gibi iki matrisin çarpılabilmesi için **A** matrisinin sütun sayısının **B** matrisinin satır sayısına eşit olması gerekmektedir. **A** matrisi **mxn**, **B** matrisi **nxk** ise bu çarpma işlemi sonucunda elde edilecek **C** matrisinin boyutu **mxk** olacaktır.

Örnek

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 3 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 2 \\ 1 & 5 \\ 2 & 3 \end{bmatrix}$$

$$C = A \times B = \begin{bmatrix} 5 & 15 \\ 1 & 12 \end{bmatrix}$$

```
clc;clear;
A=[1 2 1;0 3 -1];
B=[1 2; 1 5;2 3];
if size(A,2) ~= size(B,1)
   disp('Carpim Illegal');
   return;
end
C=zeros(size(A,1),size(B,2));
for i=1:2
   for j=1:2
      for k=1:3
         C(i,j)=C(i,j)+A(i,k)*B(k,j);
      end
    end
end
A
B
C
```



Matrislerde Tranzpoz İşlemi

Transpoz, matrislerde satır ile sütunun yer değiştirmesi işlemidir. Yani A=2x3'lük bir matrisin transpozu alındığı zaman $B=A^T$=3x2'lik bir matris elde edilir.

```
clc;clear;
A=[1 3 0; 4  2 -3];
B=[ ];
[m n]=size(A);
for i=1:n
   for j=1:m
         B(i,j)=A(j,i);
   end
end
A
B
```

A =

1	3	0
4	2	-3

$B = A^T$

1	4
3	2
0	-3

Komut satırında A' yı test ediniz.

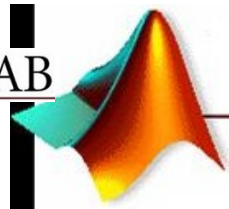
Hangi metod daha kolay ? ☺

MATLAB

UYGULAMA

B=[-45 0 5 10 -91 2] dizisinin (satır vektörünün) elemanlarını tersten başka bir diziye aktaran ve bu yeni diziyi ekrana yazdıran MATLAB programı.

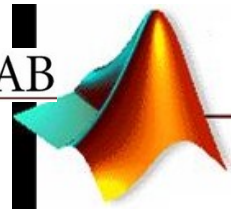
```
clc;clear;
B=[-45  0   5  10  -91  2];
C=[ ];
k=length(B);
for i=1:length(B)
    C(k)=B(i);
    k=k-1;
end
B
C
```



UYGULAMA

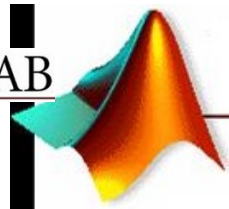
K=[-4 3 0 ; 2 0 4] matrisindeki sıfırların sayısını ve yerini (satır ve sütun numaralarını) ekrana basan bir MATLAB programı.

```
clc;clear;
K=[-4   3   0 ;   2   0  4 ];
[m n]=size(K);
sifirSayisi=0;
for i=1:m
   for j=1:n
      if K(i,j)==0
         sifirSayisi=sifirSayisi+1;
         fprintf('K(%d,%d) = 0 \n',i,j);
      end
   end
end
K
fprintf('K Matrisindeki SIFIR Sayisi = %d \n',sifirSayisi);
```



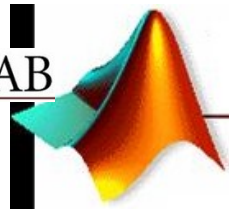
UYGULAMA

İçi rastgele oluşturulmuş tam sayılarla dolu (1 ile 50 arasında) 20 elemanlı bir satır vektörünün elemanlarını büyükten küçüğe veya küçükten büyüğe sıralayan ve ekrana basan bir MATLAB programı yazınız. (Orjinal vektörü de ekrana basmayı unutmayınız.) (2. bir vektör kullanımına izin yoktur.) **(sort fonksiyonunu kullanmiyoruz ☺)**

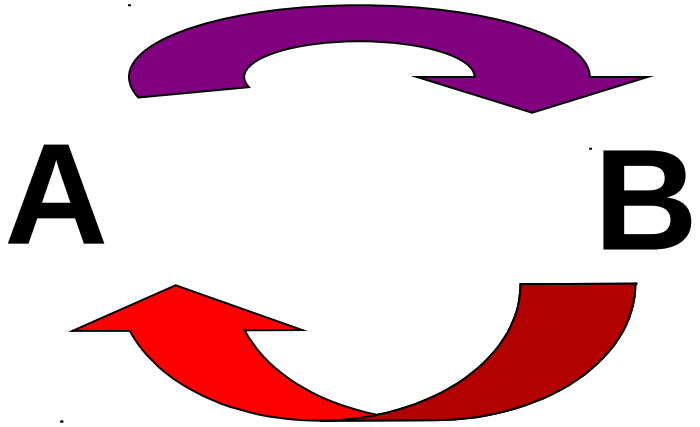


Bilgisayar Programlama

(5. Hafta)



İKİ MATLAB DEĞİŞKENİNİN DEĞERLERİNİN YER DEĞİŞTİRMESİ (SWAPPİNG)




```
geciciDegisken=A;
A=B;
B=geciciDegisken;
```

!!! GEÇİCİ BİR DEĞİŞKENE İHTİYAÇ VAR !!!

MATLAB

Sıralama yani Sorting Önemli Bir Problem

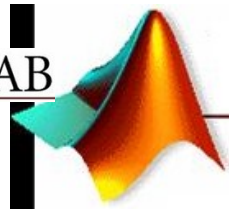
1 - Küçükten Büyüğe Sıralama (Artan) (Ascending Sort)
2 - Büyükten Küçüğe Sıralama (Azalan) (Descending Sort)

A=[3 1 2]

Artan Sıralı A=[1 2 3] >>sort(A) veya sort(A,'ascend')

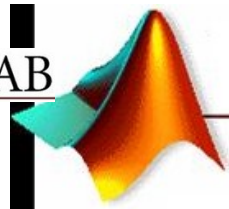
Azalan Sıralı A=[3 2 1] >>sort(A,'descend')

(MATLAB'de tanımli bu sort fonksiyonunu kullanmayacağız bizler. Kendi sort programlarımızı yazacağız.)



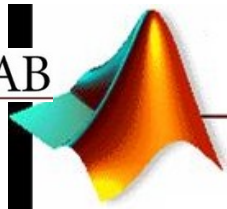
“Selection Sort” (Sıralama Algoritması)

Bu algoritma birinci elemandan başlayarak son elemana kadar, sıralanmamış DİZİYİ parça parça sıralar. Önce dizideki en küçük eleman bulunur ve dizinin ilk elemanı ile yer değiştirilir (swap). Sonraki aşamada dizinin sıralanmamış olan parçası içindeki en küçük eleman bulunur ve ikinci elemanla yer değiştirilir. Bu işlemi N defa tekrarladığımızda N elemanlı bir diziyi sıralamış oluruz.



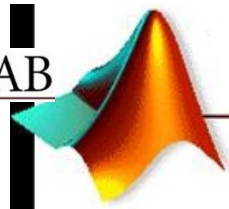
"Selection Sort" ile Küçükten Büyüğe Sıralama

```
clc;clear;
A=round(1+49*rand(1,20)) %Siralanmamis Diziyi Yazdir
tic %Siralamaya Baslamadan Once Kronometreyi Calistir
for i=1:(length(A)-1)
    minimum=i;
    for j=(i+1):length(A)
        if A(j)<A(minimum)
            minimum=j;
        end
    end
    %yer degistir (swap)
    geciciDegisken=A(i);
    A(i)=A(minimum);
    A(minimum)=geciciDegisken;
end
toc %Siralama Bittikten Sonra Kronometreyi Durdur
A %Siralanmis Diziyi Yazdir
```



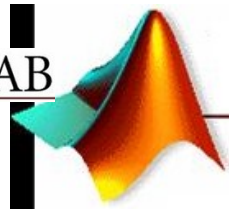
“Bubble Sort” (Sıralama Algoritması)

Bu algoritma DİZİ elemanlarını adım adım ve ikili olarak karşılaştırarak gerektiğinde (karşılaştırılan iki eleman yanlış sırada ise) yerlerini değiştirmek (swap) prensibi ile çalışır.



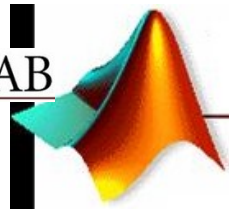
“Bubble Sort" ile Büyükten Küçüğe Sıralama

```
clc;clear;
A=round(1+49*rand(1,20)) %Siralanmamis Diziyi Yazdir
tic %Siralamaya Baslamadan Once Kronometreyi Calistir
for i=1:(length(A)-1)
   for j=length(A):-1:(i+1)
      if A(j-1)<A(j)
         %yer degistir (swap)
         geciciDegisken=A(j-1);
         A(j-1)=A(j);
         A(j)=geciciDegisken;
      end
   end
end
toc %Siralama Bittikten Sonra Kronometreyi Durdur
A %Siralanmis Diziyi Yazdir
```



“Selection Sort” “Bubble Sort" A KARŞI ☺

Bu iki algoritmanın performansını ve kod yazım kolaylığını karşılaştırınız. Hangisi daha hızlı? Bunun sebepleri üzerinde düşününüz.



FONKSİYONLAR

Fonksiyonlar da bilgisayar programlarıdır. Bilgisayarlar yardımıyla çözülmeye çalışılan problemler fonksiyonlar sayesinde yönetilmesi daha kolay, küçük parçacıklara bölünürler. Bu metoda **"Böl ve Yönet"** ya da **"Divide and Conquer"** adı verilir. Her fonksiyon:

- Kendine özgü bir isme sahiptir.
- GENELLİKLE kendine, üzerinde işlem yapacağı bir argüman (parametre) ya da argümanlar (parametreler) alır.
- GENELLİKLE geriye bir değer döndürür. (skaler, vektör ya da matris.)

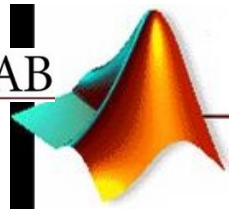
MATLAB

FONKSİYON M DOSYALARI

Bir Fonksiyon M dosyasının iki bileşeni vardır :

1 – Fonksiyonun imzası ya da prototipi
(İlk satırda tanımlanır.)

2 – Fonksiyonun tanımı (yapması gereken iş)
(İkinci satırdan başlar ve devam eder.)



FONKSİYON İMZASI YADA PROTOTİPİ

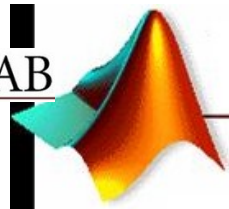
function cikis parametresi = fonksiyon_adi (giris parametreleri 1, 2, …n)

function [cikis parametreleri 1, 2,…, n] = fonksiyon_adi (giris parametreleri 1, 2, …n)

function cikis parametresi = fonksiyon_adi ( )

function cikis parametresi = fonksiyon_adi
function fonksiyon_adi (giris parametreleri 1, 2, …n)

Fonksiyon M dosyalarının ilk satırı MUHAKKAK yukarıda tanımladığımız gibi fonksiyonların imzası ya da prototipi dediğimiz satırlardan oluşmalıdır. Bir fonksiyonu yazmaya başlamadan önce onun imzası hakkında iyice düşünmeliyiz. (Giriş parametresi(leri) almalı mı, geriye bir değer döndürmeli mi?)



M-Fonksiyonlar Kullanılırken Dikkat Edilecek Hususlar :

1 - Her MATLAB fonksiyonu function anahtar kelimesi ile başlamalıdır.

2 - fonksiyon_adı m-dosyasına verilen isimle aynı olmalıdır.

3 - Bir MATLAB fonksiyonu komut penceresinden fonksiyon adı ve varsa eğer parantez içerisinde birbirlerinden virgülle ayrılmış parametrelerle çağrılır.

4 - Parametre aktarımı olması durumunda alt ve ana programda eşit sayıda giriş parametresi olmalıdır.

Örnek: İki nokta arasındaki uzaklığı bulan programı m-fonksiyon (alt program) kullanarak yazınız.

x1 = 1.noktanın x koordinati; x2 = 2.noktanın x koordinati

y1 = 1.noktanın y koordinati; y2 = 2.noktanın y koordinati

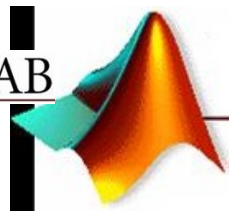
FONKSİYON ALT PROGRAMI (uzaklik.m):

```
function mesafe = uzaklik(x1,y1,x2,y2) %imza
mesafe=sqrt((x2-x1)^2+(y2-y1)^2); %tanım
```

Bu M fonksiyonu **uzaklik.m** olarak kaydedilir.

ANA PROGRAM (KOMUT PENCERESİ):

```
>> ax=3; ay=4; bx=1; by=2;
>> sonuc = uzaklik(ax,ay,bx,by) % ya da
>> uzaklik(3,4,1,2)
```



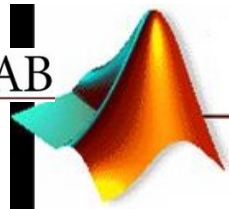
BİR ÖNCEKİ ÖRNEĞİN DÜZ M DOSYASI İLE ÇÖZÜMÜ

uzaklik.m

```
clc;clear;
x1=input('x1 koordinatini gir');
y1=input('y1 koordinatini gir');
x2=input('x2 koordinatini gir');
y2=input('y2 koordinatini gir');
mesafe=sqrt((x2-x1)^2+(y2-y1)^2);
fprintf('Girilen 2 nokta arasi mesafe : %g \n',mesafe);
```

KOMUT PENCERESİ

```
>>uzaklik
```



!!! ÖNEMLİ !!!

Bir fonksiyon M dosyasının içinde birden fazla fonksiyon tanımlayabilirsiniz. fonksiyonA.m adında bir fonksiyon M dosyasının içeriği aşağıdaki gibi olabilir.

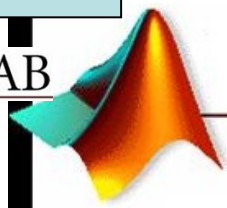
fonksiyonA.m

```
function sonuc1=fonksiyonA(n)
sonuc1=fonksiyonB(n)+5;


function sonuc2=fonksiyonB(n)
sonuc2=n*sqrt(n);
```

fonksiyonA temelde ne iş yapıyor?

Çağrılma Sırası

Komut Penceresi -> fonksiyonA -> fonksiyonB -> fonksiyonB(sonuc) -> fonksiyonA(sonuc)->Komut Penceresi(sonuc)



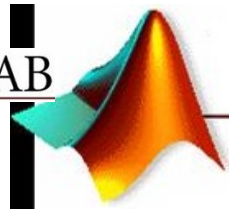
UYGULAMA

Dışarıdan girilen x ve y değerlerine göre aşağıdaki fonksiyonun değerini hesaplayan bir MATLAB programını fonksiyon M dosyası formatında yazınız. (Dosya adı olarak fonksiyon.m kullanınız.)

$$F(x,y) = x^2 \cdot y + \sqrt{x.y} + \frac{\ln(x)+1}{\log_{10}(y)} + \tan\left(\frac{x}{y}\right)$$

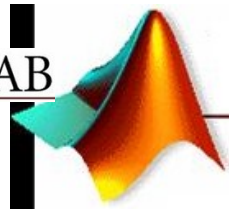
Bu fonksiyon M dosyasının imzası ne olmalı?

Bu fonksiyon M dosyasının tanımı ne olmalı?

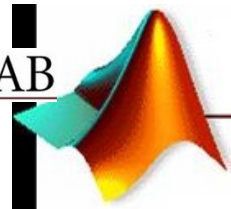


Bilgisayar Programlama

(6. Hafta)



FONKSİYON M DOSYALARI İLE UYGULAMALAR



UYGULAMA 1

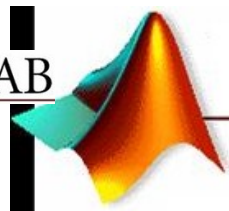
Kendisine argüman olarak aldığı x ve y adlarında 2 MATLAB değişkeninin değerlerini değiştiren (swapping) swap.m adında bir fonksiyon M dosyası yazınız.

swap.m

```
function [x y]=swap(x,y)
geciciDegisken=x;
x=y;
y=geciciDegisken;
```

KOMUT PENCERESİ

```
>>x=5;y=6;
>>[x y]=swap(x,y)
x=
     6
y=
     5
```



UYGULAMA 2

Kendisine argüman olarak aldığı pozitif bir n tamsayısının faktoriyelini hesaplayan ve geriye döndüren faktoriyel.m adında bir fonksiyon M dosyası yazınız.

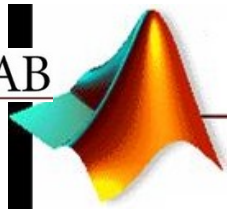
faktoriyel.m

```
function sonuc=faktoriyel(n)
sonuc=1;
for i=1:n
    sonuc=sonuc*i;
end
```

KOMUT PENCERESİ

```
>>faktoriyel(5)
ans=
        120
```

Giriş argümanı olan n sayısı negatif ise eğer bu fonksiyon geriye hangi değeri döndürür?



UYGULAMA 3

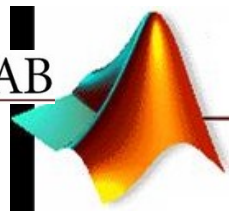
Kendisine arguman olarak aldığı pozitif bir n tamsayısının bir asal sayı olup olmadığını bulan asalMi.m adında bir fonksiyon M dosyası yazınız. Bu fonksiyon geriye hiçbir değer döndürmeyecek, ekrana disp komutuyla çıktı verecektir. (İPUCU: Bir asal sayının iki böleni vardır.)

asalMi.m

```
function asalMi(n)
bolenleri=[ ];
for i=1:n
    if mod(n,i) == 0
        bolenleri=[bolenleri i];
    end
end
if length(bolenleri) == 2
    disp([num2str(n) ‘ asal ‘]);
else
    disp([num2str(n) ‘ asal degil ‘]);
end
```

KOMUT PENCERESİ

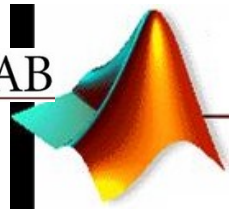
```
>>asalMi(2)
2 asal

>>asalMi(4)
4 asal degil

>>primes(16) ?????????
```



asallar.m

```
function asal=asallar(n)
asal=[ ];
for i=2:n
   if asalMi(i)
       asal=[asal i];
   end
end

function sonuc=asalMi(n)
bolenleri=[ ];
for i=1:n
     if mod(n,i) == 0
          bolenleri=[bolenleri i];
     end
end
if length(bolenleri) == 2
     sonuc=1;
else
      sonuc=0;
end
```

Bizim yazdığımız **asallar.m** adındaki bu fonksiyon M dosyası MATLAB' de zaten tanımlı **primes** adlı fonksiyonun yaptığı işi yapıyor.



UYGULAMA 4

Kendisine argüman olarak aldığı karakter dizisinin (string) bir palindrome olup olmadığını bulan **palindrome.m** adında bir fonksiyon M dosyası yazınız. Bu fonksiyon geriye 0 (string palindrome değil) ya da 1 (string palindrome) çevirecektir. (**HATIRLATMA:** Sağdan ve soldan aynı okunan kelimelere palindrome denir. Örnek: "ey edip adanada pide ye" ya da "kabak" gibi.)

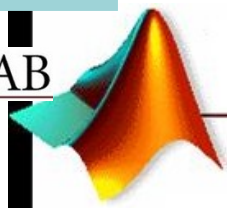
palindrome.m

```
function pal = palindrome(kelime)
boyut=length(kelime); %cift ya da tek
yariyaBol=fix(boyut/2);

pal=1; %kelimeyi palindrome kabul et

for i=0:yariyaBol
    if  kelime(i+1)==kelime(boyut-i)
        continue;
    else
        pal=0;
        break;
    end
end
```

KOMUT PENCERESİ

```
>>palindrome('kabak')
ans =
        1

>>palindrome('baba')
ans =
        0
```

MATLAB



UYGULAMA 5

Kendisine karakter dizisi (string) formatında argüman olarak aldığı ikili sayı sistemindeki bir sayıyı ondalık sayı sistemine çeviren binary.m adında bir fonksiyon M dosyası yazınız.

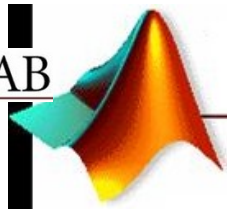
binary.m

```
function ondalik = binary(ikili)
ondalik=0;
boyut=length(ikili);
kuvvet=(boyut-1);
for i=1:boyut
        ondalik=ondalik+str2num(ikili(i))*2^kuvvet;
        kuvvet=kuvvet-1;
end
```

KOMUT PENCERESİ

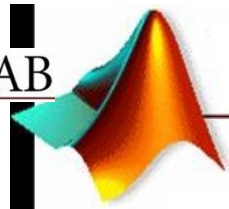
```
>>binary('111')
ans =
           7
```

111=1*2^2+1*2^1+1*2^0



Bilgisayar Programlama

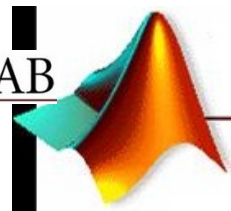
(7. Hafta)



DOSYA YÖNETİMİ

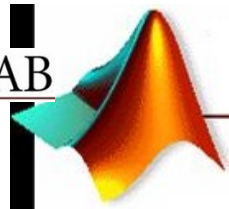
- **Şu ana kadar bir programda hesaplanan veya elde edilen veriler RAM'de saklanacak şekilde tanımlanmıştı. Yani, bilgisayar kapatıldığında veya MATLAB programı sona erdirildiğinde veriler de kaybolmaktaydı.**

- **Oysa pek çok uygulamada, elde edilen verilerin daha sonra kullanılmak üzere saklanması gerekir. Bunun için bu verilerin bir dosyaya yazılması gerekmektedir.**



3 ADIMDA DOSYA YÖNETİMİ

1. **İşleyeceğin dosyayı aç. (fopen)**
2. **Açtığın dosyayı oku ya da açtığın dosyaya yaz. (fscanf, fprintf)**
3. **Açtığın dosyayı kapat. (fclose)**



MATLAB ’de VERİ GİRİŞ-ÇIKIŞ FONKSİYONLARI

Dosya açma: fopen

MATLAB’deki fopen komutu, bilgi kaydetmek veya bilgi okumak için, varolan bir veri dosyasını yazmaya/okumaya açar veya varolmayan bir dosyayı sıfırdan oluşturur. Yazım formatı:

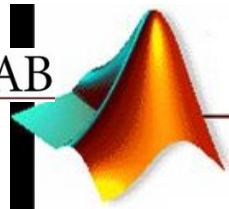
dosya_degiskeni = fopen(‘dosya adı’,’mod’)

Burada:

dosya_degiskeni-> Dosya kontrolünde kullanılan MATLAB değişkeni

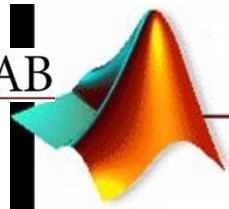
dosya adı-> Verilerin kaydedileceği/okunacağı dosyanın adı

mod-> Dosya işleme modu : r,w,.., vs.



DOSYA İŞLEME MODLARI:

MOD	AÇIKLAMA
‘r’	Dosyayı sadece okumaya açar. (yazmaya izin vermez.)
‘r+’	Dosyayı okumaya ve yazmaya açar.
‘w’	Varolan bir veri dosyasının içindekini siler, dosya yoksa oluşturur ve dosyayı yazmaya açar.
‘w+’	Varolan bir veri dosyasını yazmak için açar, dosya yoksa oluşturur ve dosyayı okumaya ve yazmaya açar.
‘a’	Varolan bir veri dosyasını yazmak için açar, dosya yoksa oluşturur ve girilecek bilgileri dosya sonuna ekler. (append)
‘a+’	Varolan bir veri dosyasını okumak ve yazmak için açar, dosya yoksa oluşturur ve girilecek bilgileri dosya sonuna ekler. (append)

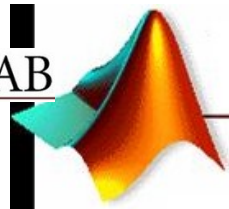


Dosya Kapama: fclose

Yazma ve/veya okumaya açılan dosyanın MATLAB'den ilişkisini kesmeye (dosyayı kapamaya) yarayan komuttur.

Kullanımı:

fclose(dosya_degiskeni)

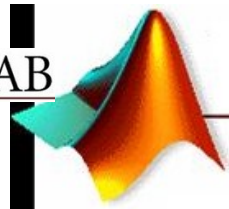


Dosyaya Bilgilerin Kaydedilmesi: fprintf

Verilerin ekrana yazılmasında kullanılan fprintf fonksiyonu, aynı zamanda fopen komutuyla açılmış dosyaya veri yazmak için de de kullanılır.

Kullanımı:

```
fprintf(dosya_degiskeni,'%format %format',degisken1,degisken2);
```



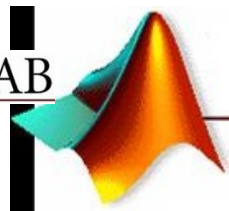
Örnek

Aşağıda verilen M dizisini (satır vektörünü) bir dosyaya kaydeden bir MATLAB programı yazınız.

```
M=[1  2  3  4];
dosya=fopen('veri.txt','w');

fprintf(dosya,'%d %d %d %d',M);

fclose(dosya); %dosyayi kapat
```

For Döngüsü ile:

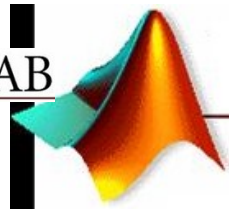
```
M=[1  2  3  4];
dosya=fopen('veri.txt','w');
for i=1:4
   fprintf(dosya,'%d ',M(i));
end
fclose(dosya);
```



Örnek

Aşağıda verilen matrisi “veri.txt” adında bir dosyaya kaydeden bir MATLAB programı yazınız.

```
M=[1  5  11 ; 2  4  5];
dosya=fopen('veri.txt','w');
for i=1:2     %satir
    for j=1:3 %sutun
        fprintf(dosya,'%d ',M(i,j));
    end
     fprintf(dosya,'\n');
end
fclose(dosya); %dosyayi kapat
```



Dosyadan Bilgilerin Okunması: fscanf

fscanf: Formatlanmış verileri içeren dosyadan bilgi okumaya yardımcı MATLAB komutudur.

[degisken, sayi] = fscanf(dosya_degiskeni,'format',alan)

Burada;

degisken: Okunan değerlerin atandığı değişken.

sayi : Okunan data sayısı.

alan : inf (infinity) ile verilerin tamamının okunması sağlanır.

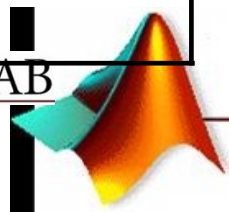
Örnek

Aşağıda verilen vektörü **"veri.txt"** adlı dosyadan okuyup **a** değişkenine atayan bir MATLAB programı yazınız.

veri.txt

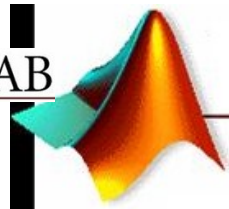
1 5 11 2 4 5

```
clc;clear;
dosya=fopen('veri.txt', 'r' );
[a,sayi]=fscanf(dosya,'%d ',inf);
fclose(dosya);
a
sayi
```



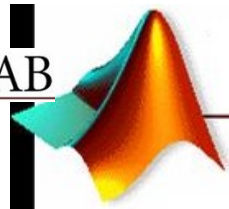
Uygulama

Dışardan girilen öğrenci no ve döneme ait aldığı ders isimlerini ve notunu kaydeden bir MATLAB programı yazınız.



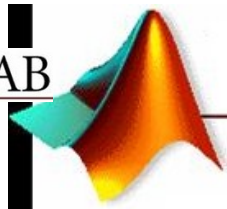
Bilgisayar Programlama

(8. Hafta)



MATLAB'DE 2 BOYUTLU GRAFİK TÜRLERİ

1. Bir grafik penceresinde tek bir grafik (**plot** komutuyla)
2. Bir grafik penceresinde birden fazla grafik (Grafik elemanlarını **plot** komutu içinde virgülle ayırarak ya da **hold** komutuyla)
3. Birden fazla grafik farklı grafik pencereleri içerisinde (**figure** komutuyla)
4. Birden fazla grafik tek bir grafik penceresi içinde ama farklı grafikler halinde (**subplot** komutuyla)



MATLAB'de GRAFİK İŞLEMLERİ

MATLAB, diğer programlama dillerinden farklı olarak oldukça güçlü bir grafik "araç kutusuna" (toolbox) sahiptir.

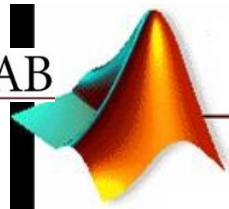
MATLAB'de 2 boyutlu (2D) grafik çizebilmek için **plot** komutu kullanılır. **plot** komutunun açtığı yeni grafik penceresi üzerinde grafiğiniz ile ilgili değişiklikler yapabilir ve hatta grafiğinizi farklı formatlarda **(bmp veya jpeg gibi)** saklayabilirsiniz. (Menüleri inceleyiniz.)

Örnek: $y = x^2 - 9x - 20$

fonksiyonunun herhangi bir aralıktaki grafiği aşağıda verilen **MATLAB** komutlarının çalıştırılması ile elde edilir.

grafik1.m

```
clear;
x=0:20;            % x ve y satir vektorleri
y=x.^2-9*x-20; % . operatorune dikkat !!!
plot(x,y);
```



GRAFİK DÜZENLEYEN KOMUTLAR

Çizeceğiniz her bir grafik için aşağıda verilen tanımlamalar mevcut olmalıdır:

1. Grafiğin başlığı
2. Eksen takımlarının isimleri

bold face: kalın harfle yaz

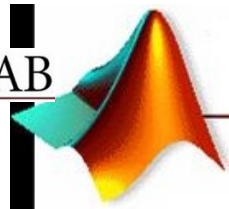
Grafiğe bir isim, başlık vermek için **title** komutu kullanılır.

x eksenine bir eksen ismi vermek için **xlabel** komutu kullanılır.

y eksenine bir eksen ismi vermek için **ylabel** komutu kullanılır.

grafik2.m

```
clear;
x=0:20;
y=x.^2-9*x-20;
plot(x,y);
title('\bf   x^2-9x-20 Fonksiyonun Grafigi');
xlabel('\bf   x');
ylabel('\bf   y');
```



ÇOKLU GRAFİKLER

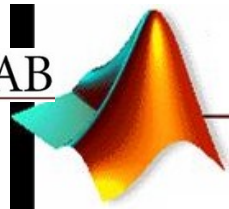
MATLAB'de tek bir grafik penceresinde birden fazla grafik çizdirmek mümkündür.

Örnek

$y(t) = 3t^2 - 5t + 8$ fonksiyonu ile türevi olan $y'(t) = 6t - 5$ fonksiyonunun t'ye göre değişimlerini aynı grafik üzerinde gösterebilmek için aşağıda verilen **MATLAB** programı çalıştırılır:

grafik3.m

```
clear;
t=0:0.1:5;%artis miktari grafigin seklini etkiler !!!
y1=3*t.^2-5*t+8;
y2=6*t-5;
plot(t,y1,t,y2);
```



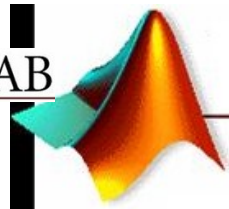
UYGULAMA

0 ile **360 derece** arasındaki açı değerleri için **sin(x)** ve **cos(x)** fonksiyonlarını aynı grafik penceresinde çizen bir **MATLAB** programı yazınız.

(Derece/180)=(Radyan/pi)
Radyan=(Derece*pi)/180

grafik4.m

```
clear;
derece=0:10:360; %artis miktari 60 deneyiniz☺
s=sin(derece*pi/180);
c=cos(derece*pi/180);
plot(derece,s,derece,c);
```

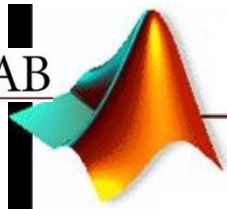


GRAFİKLERİ DÜZENLEME

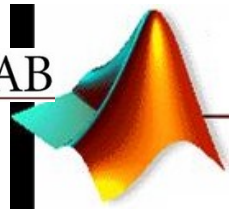
Çizeceğimiz grafiklerde aşağıda belirtilen türlerde düzenlemeler yapabiliriz:

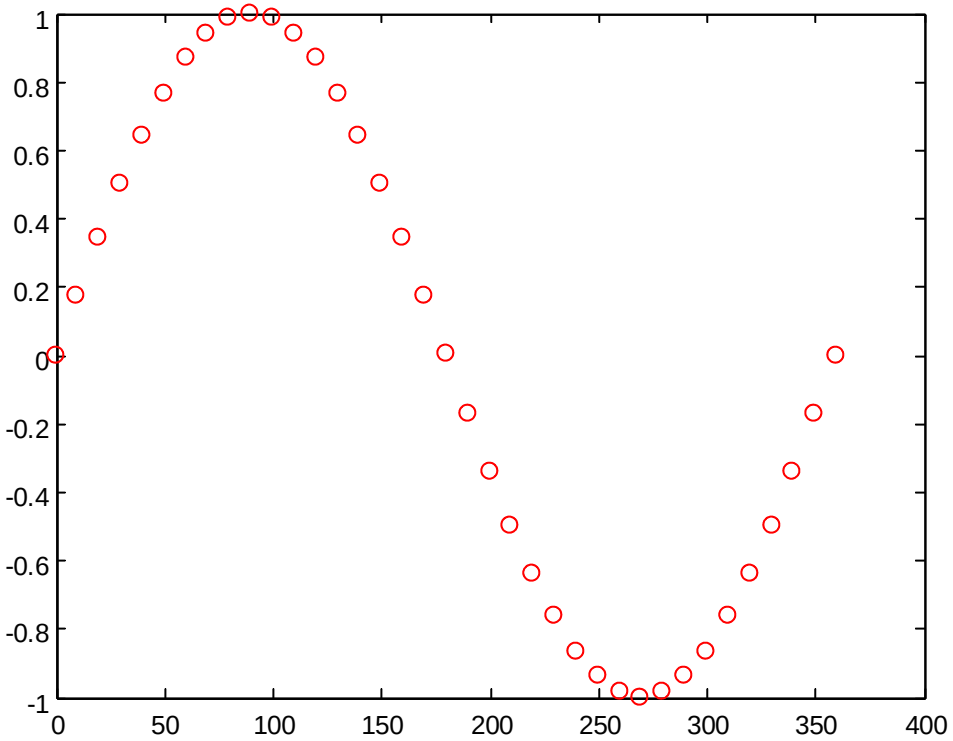
1. Çizgi rengi ve tipini değiştirmek
2. x değeri ile fonksiyon değerinin kesiştiği noktaları işaretlemek
3. Grafiklere açıklama eklemek

plot(x,y,'r-') şeklinde formatlı **MATLAB** komutu, x ve y vektörlerinin grafik çizgi renginin **kırmızı** ve stilinin **düz** olmasını sağlar.



Renk	İşaretleme Tipi	Çizgi Tipi
y: yellow (sarı)	**.** : nokta	**-** : sürekli çizgi
m: magenta (mor)	**o** : yuvarlak	**:** : nokta nokta
b: blue (mavi)	**x** : x işareti	**-.** : kesikli çizgi ve nokta
r: red (kırmızı)	**+** : artı işareti	**--** : kesikli çizgi
g: green (yeşil)	***** : yıldız işareti	
w: white (beyaz)	**s** : square (kare)	
k: black (siyah)	**d** : diamond (elmas)	
	v : aşağı üçgen	
	^ : yukarı üçgen	
	< : sola üçgen	
	> : sağa üçgen	
	p : pentagram (beşgen)	






```
clear;
derece=0:10:360;
s=sin(derece*pi/180);
plot(derece,s, 'ro-');
```

renk

işaretleme tipi

çizgi tipi

MATLAB

legend fonksiyonu ile hangi eğrinin hangi fonksiyona ait olduğu belirtilir.

>>doc legend

>>help legend

grafik5.m

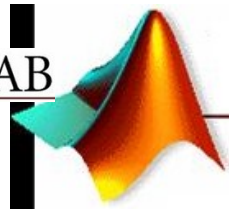
```
clear;
x=0:pi/10:2*pi;
y1=sin(x);
y2=cos(x);
plot(x,y1,'bo:',x,y2, 'rs-')
xlabel('x Degisimi');
ylabel('Fonksiyon Degisimi');
title('sin(x) ve Turevinin Degisimi');
legend('sin(x)','cos(x)',-1); %grafik ekrani disinda ve sagda (-1)
```

figure Fonksiyonu ile Çoklu Grafikler

Birden fazla grafik penceresi açmak için **figure**(**n**) komutu kullanılır. Burada **n** grafik penceresinin adını belirtmektedir.

grafik6.m

```
clear;
x=0:pi/30:2*pi;
y1=sin(x);
y2=cos(x);
figure(1);
plot(x,y1,'bo:');
figure(2);
plot(x,y2,'r*-');
title('cos(x) grafigi');%sadece figure 2'ye ait baslik
```



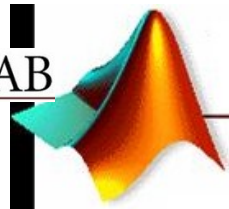
subplot Fonksiyonu ile Alt Grafikler

Aynı grafik penceresinde birden fazla grafik çizmek için **subplot(a,b,c)** fonksiyonu kullanılır. Burada:

a : Grafik penceresinin satır sayısıdır.

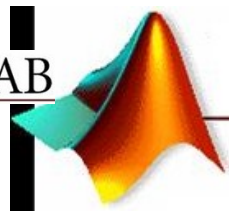
b : Grafik penceresinin sütun sayısıdır.

c : Alt pencere numarasıdır.



grafik7.m

```
clear;
x=0.1:pi/10:2*pi;
y1=sin(x);
y2=cos(x);
y3=tan(x);
y4=cot(x);
subplot(2,2,1);
plot(x,y1);
title('f(x)=sin(x)');
subplot(2,2,2);
plot(x,y2);
title('f(x)=cos(x)');
subplot(2,2,3);
plot(x,y3);
title('f(x)=tan(x)');
subplot(2,2,4);
plot(x,y4);
title('f(x)=cot(x)');
```



hold Komutu

Aynı eksen takımında birden fazla grafik **hold** komutu yardımıyla da çizilebilir. **figure** fonksiyonu kullanılmadığı sürece işletilen herbir **plot** komutu aynı grafik penceresinde işlem görür. Aynı grafik ekranına çizilecek fonksiyonların sayısı fazla ise eğer **plot** komutu içinde fonksiyonları birbirinden virgülle ayırmak sıkıcı olabilir. Bu tür durumlarda **hold** komutu kullanılmalıdır.

grafik8_1.m

```
clear;
x=0:pi/30:2*pi;
y1=sin(x);
y2=exp(2*sin(x));
plot(x,y1,'r*:');
hold;
plot(x,y2,'bo--');
legend('sin(x)','e^{2*sin(x)}');
```

grafik8_2.m

```
clear;
x=0:pi/30:2*pi;
y1=sin(x);
y2=exp(2*sin(x));
plot(x,y1,‘r*:',x,y2,‘bo--');
legend('sin(x)','e^{2*sin(x)}');
```

hold komutunu kaldırsak ne olur?

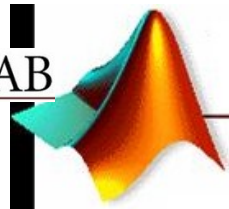
Hayat Bu Kadar Kolay Değil ☺

Bugünkü dersimizde gördüğümüz bütün örneklerde 2 boyutlu grafiğimizin x eksenindeki değerleri saklayacak satır vektörünü kolayca oluşturduk. Çoğu zaman bu işlemi de **MATLAB**'de yazmamız gereken bir program yardımıyla gerçekleştirmemiz gerekebilir. Örneğin x ekseninini tanımlayan satır vektörünün içinde aralığı dışardan girilecek asal sayıların olması istenebilir. Aynı durum y ekseni için de geçerlidir.

Eğlenceli Bir Uygulama

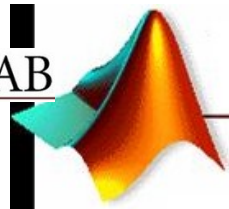
grafik9.m

```
clear;
x=[10,5,3,2,9,14,17,20,25,27,28,29,30,38,45,49,52,54,58,...
59,60,62,66,72,78,81,82,84,87,90,97,102,106,109,112,119,...
125,128,126,122,118,117,121,134,154,174,190,194,194,185];

y=[16,50,70,104,106,104,95,80,67,59,87,124,153,157,144,127,...
109,90,71,100,134,163,178,179,174,161,141,117,93,76,89,105,...
123,140,153,156,144,128,106,86,65,48,30,17,24,29,25,21,16,7];

plot(x,y);
```



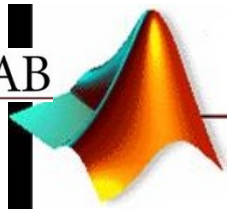
Bilgisayar Programlama

(9. Hafta)



Diziler (Vektörler) İçin Tanımlı Bazı Fonsiyonlar

- **max(x):** x dizisinin en büyük elemanını bulur.
- **min(x):** x dizisinin en küçük elemanını bulur.
- **mean(x):** x dizisine ait elemanların aritmetik ortalamasını bulur.
- **std(x):** x dizisine ait elemanların standart sapmasını bulur.
- **sort(x):** x dizisinin elemanlarını küçükten büyüğe doğru sıralar.
- **median(x):** Sıralanmış x dizisinin (kendi sıralar) orta elemanını bulur.
- **sum(x):** x dizisinin elemanlarının seri toplamını bulur.
- **prod(x):** x dizisinin elemanlarının seri çarpımını bulur.
- **cumsum(x):** x dizisinin elemanlarının kümülatif toplamından oluşan yeni bir dizi oluşturur.
- **diff(x):** x dizisinin elemanlarının farklarından oluşan yeni bir dizi oluşturur.



Diziler (Vektörler) İçin Tanımlı Bazı Fonsiyonlar

max(x)

```
clc; clear;
x = [8,4,6,3,5,14,7,9,10]
enBuyuk = max(x);
enBuyuk
```

min(x)

```
clc;clear;
x = [8,4,6,3,5,14,7,9,10]
enKucuk = min(x);
enKucuk
```

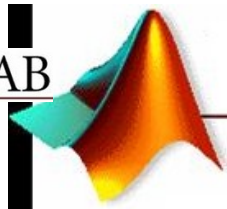
mean(x)

```
clc;clear;
x = [8,4,6,3,5,14,7,9,10]
ortalama = mean(x);
ortalama
```

std(x)

```
clc;clear;
x = [8,4,6,3,5,14,7,9,10]
sapma = std(x);
sapma
```

mean(x) = (sum(x)/length(x)) ???



Diziler (Vektörler) İçin Tanımlı Bazı Fonsiyonlar

sort(x)

```
clc; clear;
x = [8,4,6,3,5,14,7,9,10]
artanSirali = sort(x);
artanSirali
```

sort(x,'ascend')

```
clc; clear;
x = [8,4,6,3,5,14,7,9,10]
artanSirali = sort(x,'ascend');
artanSirali
```

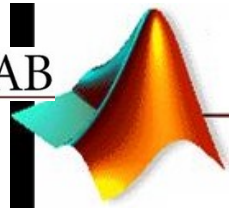
sort(x,'descend')

```
clc; clear;
x = [8,4,6,3,5,14,7,9,10]
azalanSirali = sort(x,'descend');
azalanSirali
```

median(x)

```
clc; clear;
x = [8,4,6,3,5,14,7,9,10]
ortaEleman = median(x);
ortaEleman
```

MATLAB



Diziler (Vektörler) İçin Tanımlı Bazı Fonsiyonlar

sum(x)

```
clc; clear;
x = 1:10
seriToplam = sum(x);
seriToplam
```

prod(x)

```
clc;clear;
x = 1:5
faktoriyel = prod(x);
faktoriyel
```

cumsum(x)

```
clc; clear;
x = 1:10
kumulatifToplam = cumsum(x);
kumulatifToplam
```

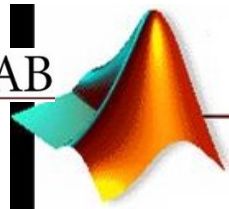
diff(x)

```
clc; clear;
x = [1,4,2,9,6,18,7,3,11,8]
farklar = diff(x);
farklar
```

```
toplam=0;
for i=1:n
    toplam=toplam+i
end
```

Bu toplam nasıl bir toplam ???

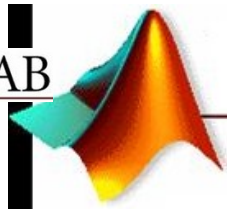
MATLAB



Matlab’da Polinomlar

Sabit katsayılı n. dereceden bir polinomun genel hali:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0$$



Matlab'da Polinomlar

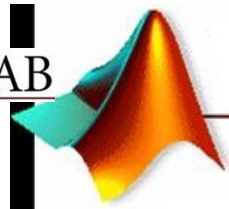
$$s^4 + 3s^3 - 15s^2 - 2s + 9$$

katsayilarVektoru = [1, 3, -15, -2, 9]

$$s^4 + 1$$

katsayilarVektoru = [1, 0, 0, 0, 1]

Boyutu (n+1) olan bir vektör n. dereceden bir polinomun katsayılarını saklar.



Polinomlar İçin Tanımlı Bazı Fonksiyonlar

roots: Vektör şeklinde tanımlanmış katsayılar polinomunun köklerini hesaplar.

poly: Kökler bilindiğinde polinom katsayıları vektörünü hesaplar.

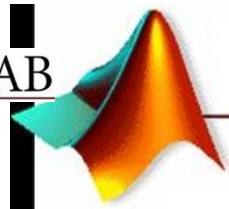
polyval: Polinom değişkeninin herhangi bir değeri için polinomun değerini bulur.

conv: İki polinomun çarpımından (konvolüsyonundan) oluşan katsayılar polinomunu verir.

deconv: Bir polinom başka bir polinoma bölündüğünde bölüm ve kalan polinomlarını hesaplar.

polyder: Bir polinomun türevini alır.

polyint: Bir polinomun integralini alır.

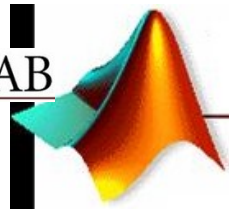


roots

(x^2-2x+1=0)

polinom1.m

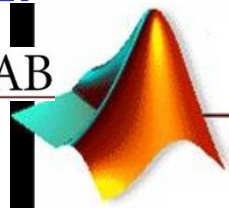
```
polinom=[1,-2,1];
polinomKokleri=roots(polinom) %veya
%roots([1,-2,1])
```



poly

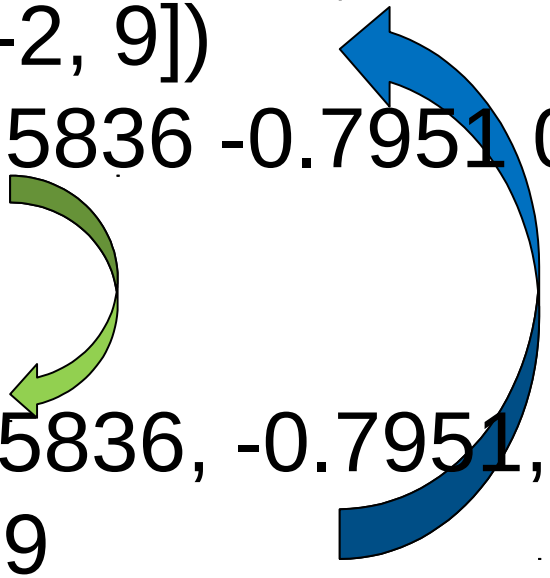
polinom2.m

```
kokler=[-5.5745, 2.5836, -0.7951, 0.7860];

polinom=poly(kokler) %veya

%poly([-5.5745, 2.5836, -0.7951, 0.7860])
```



roots ve poly

```
roots([1, 3, -15, -2, 9])
ans = -5.5745 2.5836 -0.7951 0.7860
```



```
poly([-5.5745, 2.5836, -0.7951, 0.7860 ])
ans = 1 3 -15 -2 9
```

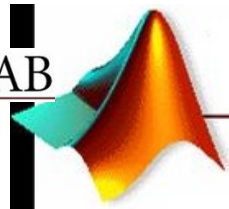
MATLAB

polyval

$p(x)=3x^5 -4x^4+ 8x^3 +6x^2 -x+1$ ise $p(1)=?$

polinom3.m

```
p=[3,-4,8,6,-1,1];
sonuc=polyval(p,1) %veya
%polyval([3,-4,8,6,-1,1],1)
```



polyval

$$p(x)=3x^5 -4x^4+ 8x^3 +6x^2 -x+1$$

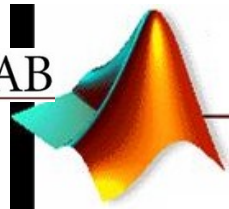
derece = [5 4 3 2 1 0];

p = [3 -4 8 6 -1 1];

polinomDegeri=p(a)=?

?????? derece = (length(p)-1):-1:0 ??????

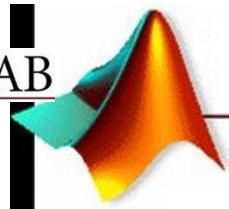
```
polinomDegeri=0;
for i=1:length(p)
        polinomDegeri = polinomDegeri+ p(i)*a^(derece(i));
end
```



conv (Polinom Çarpımı)

```
>> x = [1 2];                % (x+2)

>> y = [1 4 8];              % (x²+4x+8)

>> z = conv(x,y)             % (x+2)*(x²+4x+8)

z =
     1     6    16    16     % x³+6x²+16x+16
```

conv fonksiyonunun yaptığı işi yapan bir fonksiyon M dosyasını siz kendiniz yazabilir misiniz?



deconv

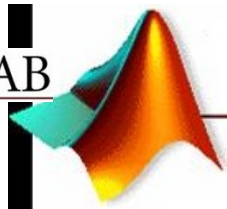
conv fonksiyonu yardımıyla iki polinomun çarpımı ile elde edilecek bir polinoma ait katsayılar bulunur. **deconv** fonksiyonu ise bir polinom ikinci bir polinoma bölündüğünde bölüm ve kalan polinomlarını hesaplar.

$$f(x)= x^2+2x+3$$

$$g(x)=4x^2+5x+6$$

şeklinde iki polinom verilmiş olsun:

```
f=[1 2 3]; g=[4 5 6];
carpim=conv(f,g)
[bolum  kalan]=deconv(carpim,f)
[bolum  kalan]=deconv(carpim,g)
```



polyder

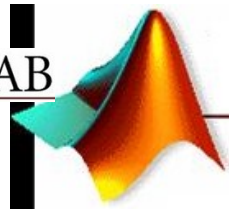
$$p(x)=x^5-2x^4+2x^3+3x^2+x+4$$

polinomunun türevi:

$$dp/dx=5x^4-8x^3+6x^2+6x+1$$

polinom4.m

```
clc; clear;

p=[1  -2   2   3   1   4]

turev=polyder(p)
```



polyder

$$p(x)=x^5-2x^4+2x^3+3x^2+x+4$$

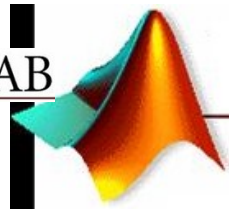
derece = [5 4 3 2 1 0]

p = [1 -2 2 3 1 4]

polyder(p) = [5 -8 6 6 1]

?????? derece = (length(p)-1):-1:0 ??????

polyder(p) = p.*derece ???????



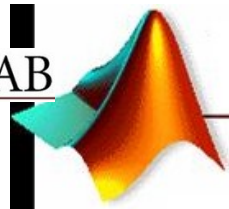
polyint

$$p(x)= x^5-2x^4+2x^3+3x^2+x+4$$

polinomunun integrali:

$$Q(x)=\int (x^5-2x^4+2x^3+3x^2+x+4)dx$$

```
clc; clear;

format rat;

p=[1  -2   2   3   1   4];

integral=polyint(p)

format;
```



polyint

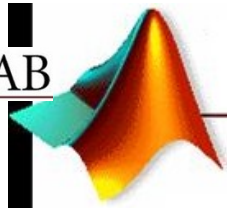
$$p(x)=x^5-2x^4+2x^3+3x^2+x+4$$

derece = [5 4 3 2 1 0]

p = [1 -2 2 3 1 4]

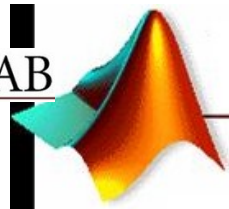
polyint(p) = [1/6 -2/5 2/4 3/3 1/2]

polyint(p)(1) = p(1)/(derece(1)+1) ???????



Bilgisayar Programlama

(10. Hafta)



SEMBOLİK (SİMGESEL) DEĞİŞKENLER VE İŞLEMLER

Daha ilk dersimizde MATLAB yardımıyla hem nümerik hem de sembolik işlemler yapabileceğimizi söylemiştik. MATLAB'de yapılacak sembolik (simgesel) işlemlerde Symbolic Math Toolkit kullanılır. Bu toolkit içinde mevcut fonksiyonlar

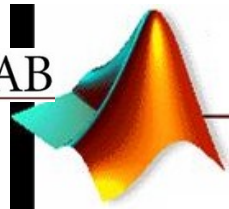
>>help symbolic

ile görüntülenebilir.

Bundan önceki derslerimizden de bildiğimiz gibi sayısal (nümerik) işlemlerimizde değişkenleri kullanmadan önce tanımlama gereği duymuyorduk. Oysa sembolik işlemlerde kullanacağımız değişkenleri sayısal değişkenlerden ayırabilmek için önceden tanımlama ihtiyacı duyarız. Örneğin a gibi bir sembolik değişken

>>syms a

ile tanımlanmalıdır kullanılmadan önce.



BİRDEN FAZLA SEMBOLİK DEĞİŞKENİN syms KOMUTU İLE AYNI ANDA TANIMLANMASI

Bir sembolik ifade içinde kullanılacak değişkenler x, y ve z ise eğer

>>syms x y z

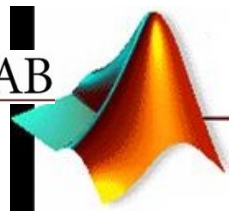
şeklindeki MATLAB deyimi bize kısa yoldan bu sembolik değişkenleri tanımlama imkanını verir.

Artık bu noktadan sonra yukarıda tanımladığımız sembolik değişkenleri sembolik ifadeler içinde kullanabiliriz.

>>sembolik=x^2-y^2

sembolik=

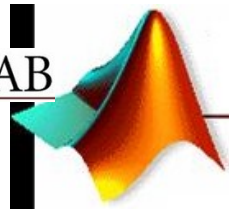
x^2-y^2



SEMBOLİK İFADELERE UYGULANABİLECEK MATLAB FONKSİYONLARI

Bir önceki sembolik ifadeyi faktörlerine (çarpanlarına ayırmak) için:

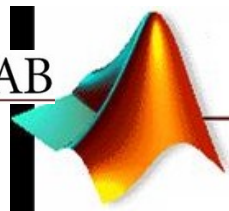
```
>>factor(sembolik)
ans=
        (x-y)*(x+y)
>>factor(x^3-3*x^2+3*x-1)
ans=
        (x-1)^3
>>sembolik^3
ans=
        (x^2-y^2)^3
>>expand(ans) % sembolik ifadenin acik formu
ans=
        x^6-3*x^4*y^2+3*x^2*y^4-y^6
```



SEMBOLİK İFADELERE UYGULANABİLECEK MATLAB FONKSİYONLARI

Bir sembolik ifadeyi sadeleştirmek için simplify fonksiyonu kullanılır.

```
>>sembolik=(x^3-4*x)/(x^2+2*x)
ans=
        (x^3-4*x)/(x^2+2*x)
>>simplify(sembolik)
ans=
        x-2
```



SEMBOLİK İFADELERİN İNTEGRALİNİ ALMA

Bu işlem için sembolik integral alma fonksiyonu olan int kullanılır.

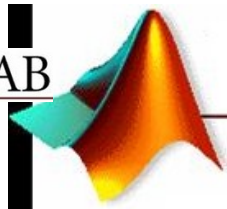
int(s,x) : s adlı sembolik ifadenin x'e göre belirsiz integralini alır.

int(s,x,a,b) : s adlı sembolik ifadenin x'e göre a'dan b'ye kadar belirli integralini alır.

ÖRNEK : $Q(x)=\int (x^5-2x^4+2x^3+3x^2+x+4)dx$ integralinin hesaplanması

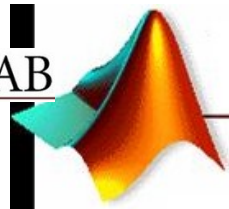
```
>>syms x
>>sembolik = x^5-2*x^4+2*x^3+3*x^2+4;
>>int(sembolik,x) %veya int(sembolik)
ans =
                1/6*x^6-2/5*x^5+1/2*x^4+x^3+4*x
>>pretty(int(sembolik)) % daha guzel bir integral sonucu ciktisi icin
                   6         5         4    3
              1/6 x  - 2/5 x  + 1/2 x  + x  + 4 x
>>int(sembolik,x,1,2)
ans=
                83/5
```

MATLAB



SEMBOLİK İFADELERİN İNTEGRALİNİ ALMA

```
>>syms x
>>sembolik = x*log(x)+2*cos(x)+exp(x);
>>int(sembolik)
ans =
        1/2*x^2*log(x)-1/4*x^2+2*sin(x)+exp(x)
```



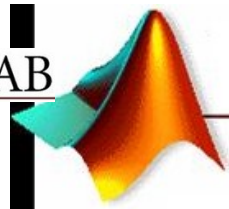
SEMBOLİK İFADELERİN TÜREVİNİ ALMA

Bu işlem için sembolik türev alma fonksiyonu olan diff kullanılır.

```
>>syms x

>>sembolik=sin(exp(x));

>>diff(sembolik)
ans =
                cos(exp(x))*exp(x)
```

ÖRNEK: f(x,y) = x^3*y^4+y*sin(x)

```
>>syms x y
>>sembolik= x^3*y^4+y*sin(x);
>>diff(sembolik,x) % x'e gore kismi turev
ans =
                3*x^2*y^4+y*cos(x)
>>diff(sembolik,y) % y'ye gore kismi turev
ans =
                4*x^3*y^3+sin(x)
```



LİNEER DENKLEM SİSTEMLERİNİN ÇÖZÜMÜ

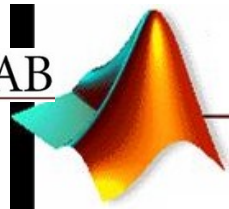
n. dereceden bir lineer denklemler sistemi aşağıdaki gibi tanımlanır:

$$a_{11}x_1+a_{12}x_2+\ldots\ldots+a_{1n}x_n = b_1$$
$$a_{21}x_1+a_{22}x_2+\ldots\ldots+a_{2n}x_n = b_2$$
$$\text{----------}$$
$$\text{----------}$$
$$a_{n1}x_1+a_{n2}x_2+\ldots\ldots+a_{nn}x_n = b_n$$

Bu denklem sistemi matris biçiminde $[A]*[X]=[B]$ şeklinde de yazılabilir. Burada A katsayılar matrisi, B sonuç vektörü (sütun), X ise çözümü istenen değişkenler vektörü (sütun) olarak adlandırılır.

$$A = \begin{bmatrix} a_{11} & a_{12} \ldots\ldots a_{1n} \\ a_{21} & a_{22} \ldots\ldots a_{2n} \\ \text{----------} & \\ \text{----------} & \\ a_{n1} & a_{n2} \ldots\ldots a_{nn} \end{bmatrix}_{nxn} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ \cdots \\ x \end{bmatrix}_{nx1} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ \cdots \\ b \end{bmatrix}_{nx1}$$

MATLAB



LİNEER DENKLEM SİSTEMLERİNİN ÇÖZÜMÜ

[A]*[X] = [B] ise => X = inv(A)*B

(A matrisinin tersinin alınabilmesi için kare matris olması gerekir.)

ÖRNEK: **Aşağıda verilen lineer denklem sistemini çözünüz.**

$$x_1 + 4x_2 - x_3 + x_4 = 2$$
$$2x_1 + 7x_2 + x_3 - 2x_4 = 16$$
$$x_1 + 4x_2 - x_3 + 2x_4 = 1$$
$$3x_1 - 10x_2 - 2x_3 + 5x_4 = -15$$

```
>>A=[1 4 -1 1;2 7 1 -2;1 4 -1 2;3 -10 -2 5];
>>B=[2;16;1;-15];
>>X=inv(A)*B
```

```
X=
    2.000
    1.000
    3.000
   -1.000
```