

3.8 Alt programlar (function)

MATLAB programları kod (script) ya da altprogram (function) olarak düzenlenebilir. Özel bir görevi işlevi yerine getiren programların altprogram olarak düzenlenmesinde yarar vardır. Bunun nedeni sıkça başvurulan bu tür işlemler için programın tekrar tekrar yazılması yerine yazılmış bulunan bir altprogramın çağırılmasının daha kolay olmasıdır.

MATLAB dilinde yazılan programlar *.m dosya türü ile saklanmaktadır. MATLAB altprogramları da bu dosya türünde saklanmaktadır. Bir ana program ile bir alt program aynı m-dosyası içinde saklanamaz ancak birden çok altprogram (function) tek bir m-dosyasında saklanılabilir. MATLAB m-dosyalarının çağırılmasında altprogram adını değil dosya adını temel alır. Aşağıda bir MATLAB altprogramının yapısı verilmiştir.

```
function [ort,sapma]=ortalama(x)
X=x(:);
ort=sum(X)/length(X);
sapma=std(X);
return
end
```

Örnekte de görüleceği üzere bir alt program `function` deyimini ile başlar. Ardından bu alt programdan geri döndürülecek değişken ya da değişkenler listelenir. Bir alt programın geriye bir değişken döndürme zorunluluğu yoktur. Durum böyle ise doğrudan fonksiyon ismi ile başlanır. Yalnız bir değişken varsa o değişkenin adı yazılır. Birden çok değişken geri döndürülecekse değişken isimleri köşeli parantez içerisinde listelenir. Değişken isimleri listelendikten sonra bir = işareti konulur. Altprogramın adı bu işarettten sonra yazılır. Fonksiyon adından sonra gelen parantezler içerisine de bu altprograma ana programdan gelen değişkenlerin isimleri yazılır. Ana program ve altprogram arasında gelip-giden değişken isimleri aynı olmak zorunda değildir. Burada belirleyici olan sıralamadır. Aynı sıradaki değişkenler ana ve altprogramda birbirlerinin karşılığı olarak atanmaktadır. Bu bilgiler ışığında yukarıda bir **x** dizeyi içerisindeki sayıların ortalamasını ve standart sapmasını bulan altprogram bir başka program içerisinden aşağıdaki şekilde çağırılabilir:

```
[ort,s]=ortalama(F1);
```

Böylelikle **F1** dizeyinin elemanlarının ortalaması ve standart sapmaları hesaplanarak **ort** ve **s** isimli değişkenlere geri döndürülecektir. Birden çok fonksiyon aynı dosya içerisinde alt alta yazılabilir. Örnek olarak yukarıdaki altprogram içeriğindeki ortalama ve standart sapma hesaplamaları ayrı altprogramlar olarak aynı dosya içerisine yazılabilir:

```

function [ort,sapma]=ortalama(x)
X=x(:);
ort=sum(X)/length(X);
end

function sapma=StdDev(X)
sapma=std(X);
end

```

Alıştırmalar

1. Kendisine girdi olarak gönderilen bir n skalerinin faktoriyelini hesaplayan altprogramı yazınız.
2. $f(x; p) = \sin(p_1x) + \exp(p_2x)$ fonksiyonunu x 'in başlangıç, bitiş ve artım değerlerini ve p_1, p_2 parametrelerini girdi olarak alan, hesaplayan ve ana programa geri gönderen bir altprogram yazınız.
3. Önceki örnekte yazdığımız altprogramı kullanarak verilen fonksiyonu $p_1=2.25, p_2=-0.25$ değerleri ve x 'in $[0\ 5]$ aralığındaki değerleri için 0.1 artımlarla hesaplatınız.
4. $\text{sinc} c = \frac{\sin(2\pi ft)}{2\pi ft}$ şeklinde tanımlanan sinc fonksiyonunu $f=2\text{Hz}$ frekansı için $-5,5$ zaman aralığında 0.1 artımlar için hesaplayan ve ana programa gönderen bir `function` yazınız. $t=0$ anındaki limit durumunun 1 olduğunu dikkate alınız.
5. Girdi olarak gönderilen bir dizeyin içerisindeki elemanlardan sıfırdan küçük olanları NaN ile değiştirerek geri gönderen bir `function` yazınız.
6. Aşağıdaki biçimde düzenlenmiş bir metin dosyasını okuyup, birinci satırda verilen sayıyı nd sonraki satırlarda verilen bilgilerin birinci sütununu x , ikinci sütununu ise G değişkenlerine atayarak ana programa gönderen bir `function` yazınız.

```

21
0      145.65
2      112.46
4      109.13
5      105.02
8      99.74
11     101.16
17     103.47
16     108.61
...     ...

```

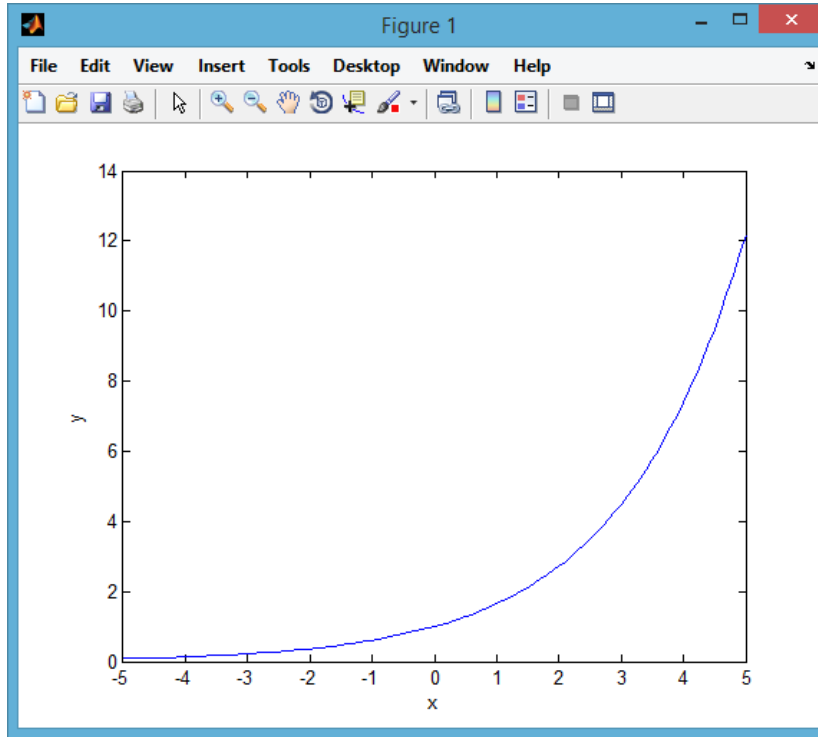
3.9 Grafik Düzenleme

Sayısal ve mühendislik hesaplamalarda ara adımların veya sonuçlarının görsel sunumu problemin çözümünde ve karar vermede büyük kolaylık sağlamaktadır. Bazı durumlarda birkaç sayfa yazı ile anlatılabilecek bilgi ya da sonuç bir şekil ile basit ve anlaşılır olarak ortaya konulabilir. Bu nedenle iyi bir mühendis ya da bilim adamı gerek problemin tanımlanmasında gerekse çözümü ile ilgili durumları ortaya koymada uygun grafikleme ve çizim araçlarını kullanabilmelidir. MATLAB bu çerçevede geniş bir grafik kütüphanesi barındırmaktadır. Bunlardan en temel olanları burada verilecek daha ileri düzey görselleştirme ve grafikleme araçları uygulamalar bölümünde verilmiştir.


plot

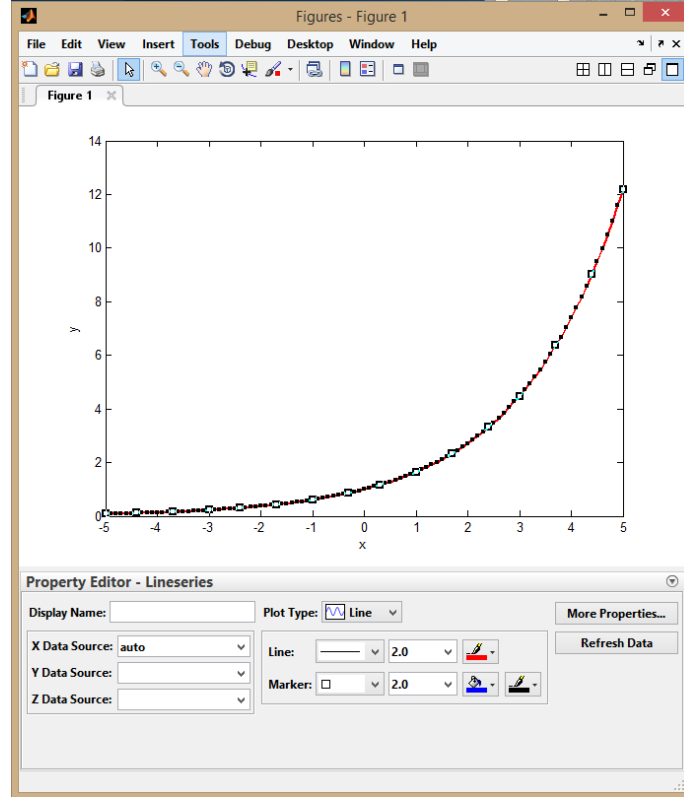
x-y dağılımı türünde bilgilerin çizilmesinde kullanılan en temel MATLAB fonksiyonudur. Bu tür bir çizim yapmak için birbiriyle aynı boyda iki vektör (girdi olarak sağlanmalıdır). Aşağıda bir x-y çifti için böyle bir çizimi yapan MATLAB ifadeleri verilmiştir.

```
x=-5:.1:5;
y=exp(0.5*x);
plot(x,y)
xlabel('x') %x-ekseni etiketi ayarlanıyor
ylabel('y') %y-ekseni etiketi ayarlanıyor
```



Şekil 3.5 `plot` fonksiyonuyla oluşturulan x-y grafiği

Grafik penceresinde görülen eğri, çizgi, yazı gibi tüm bileşenlerin özellikleri değiştirilebilmektedir. Pencere üzerinde etkileşimli olarak düzenlemeyi etkin hale getirmek için araç çubuğu üzerinde bulunan  simgesi seçilebilir. Düzenleme etkin iken özelliği değiştirilmek istenen bileşene çift tıklatılarak değiştirilebilecek özelliklere erişim sağlanabilir. Burada seçilen bileşene bağlı olarak, yazı tipi, eğri rengi, işaretçi türü, eksen sınırları, eksen etiketleri, grafik başlığı, grafik arka plan rengi gibi çok sayıda özellik değiştirilebilmektedir. Aşağıdaki şekilde eğri rengi ve veri noktalarına konulan işaretçilerin şekli gibi özelliklerin değiştirildiği pencere görülebilir.



Şekil 3.1 Şekil düzenleme

`plot` fonksiyonu ile üretilen şekillerdeki tüm düzenlemeler fonksiyonun kullanımı sırasında da yapılabilmektedir. Üretilen şekildeki tüm bileşenlerin görünimleri geçerli özellik-değer çiftleri ile değiştirilebilmektedir. Çizelge 3.11 bu tür değişikliklerin komut satırında ya da MATLAB programları içerisinde bu tür düzenlemelerin nasıl yapıldığına dair örnekleri içermektedir. Burada verilen örnekler dışında çok sayıda özellik benzer kod yazımları ile değiştirilebilmektedir. Ancak bunlar kullanıcının merak ve gereksinimlerine bağlı olarak ulaşıp öğrenilebileceği düzeydedir. Dolayısıyla örnekler burada verilenler ile sınırlı tutulacaktır. `plot` fonksiyonunun kullanılması ve penceredeki bileşenlerin düzenlenmesi ile ilgili daha fazla örnek ve bilgi MATLAB yardım belgeleri arasında bulunabilir. MATLAB komut satırına `doc plot` yazılarak örnek çizim ve düzenlemelere ulaşılabilir.

Çizelge 3.1 plot fonksiyonu ile üretilen grafiklerin bazı özelliklerinin düzenlemesi

Tanımlayıcı	Açıklama
<code>plot(x,y,'r')</code>	Eğri rengi kırmızı olsun. Diğer bazı renkler siyah : 'k', mavi : 'b', yeşil: 'g', sarı: 'y', camgöbeği: 'c', fuşya: 'm', , beyaz: 'w'
<code>plot(x,y,'k+')</code>	Eğri siyah ve veri işaretçisi + olsun. Veri işaretçiler için bazı seçenekler çember : 'o', yıldız : '*', çarpı: 'x', kare: 's', baklava: 'd', nokta: '.'
<code>plot(x,y,'g--s')</code>	Eğriyi kesikli çizgiler ile göster, yeşil yap ve kare işaretçiler koy. Diğer bazı çizgi türleri: düz çizgi : '-', kesikli '--', kesikli-noktalı : '-.', noktalı: ':', çizgi yok: ' '
<code>plot(x,y,'b','LineWidth',2)</code>	Eğri mavi renk olsun, çizgi kalınlığı 2 punto olsun
<pre>t=0:pi/20:2*pi; plot(t,sin(t),'mo',... 'LineWidth',2... 'MarkerEdgeColor','k'... 'MarkerFaceColor','g'... 'MarkerSize',10);</pre>	Eğri rengi fuşya, işaretçi çember, çizgi kalınlığı 2 punto, işaretçi çizgi rengi siyah, işaretçi dolgu rengi yeşil ve işaretçi büyüklüğü 10 punto olsun.

Hangi çizim aracının kullanıldığından bağımsız olarak grafikleme ile ilgili sık başvurulan bazı MATLAB deyimlerini tanımlamak yerinde olacaktır. Aşağıdaki çizelge çeşitli grafik penceresi düzenleme deyimleri ve kullanımlarını özetlemektedir.

Çizelge 3.2 Bazı grafik düzenleme deyimleri

Deyim	Açıklama	Örnek
<code>hold on</code>	Geçerli grafik penceresini tutarak bundan sonra gelecek çizim işlemlerinin aynı pencereye yapılmasını sağlar. <code>hold off</code> ile bu özellik kapatılabilir.	<pre>x=-2*pi:0.01:2*pi; y1=sin(x); y2=cos(x); plot(x,y1,'r') hold on plot(x,y2,'b')</pre>
<code>grid on</code>	x ve y yönlerinde grid çizgileri eklenir. <code>grid off</code> seçeneği ile kapatılabilir.	<pre>x=-2:.15:2; plot(x,x.^2) grid on</pre>
<code>title</code>	Grafik penceresi üzerine bir başlık eklenir.	<pre>t=0:5; plot(t,exp(t)) title('Üstel Fonksiyon')</pre>
<code>legend</code>	Grafik üzerindeki verilerin ne anlama geldiğini gösteren bir gösterge (lejang)	<pre>legend('sin(x)', 'cos(x)')</pre>
<code>figure</code>	Yeni bir grafik penceresi açar	

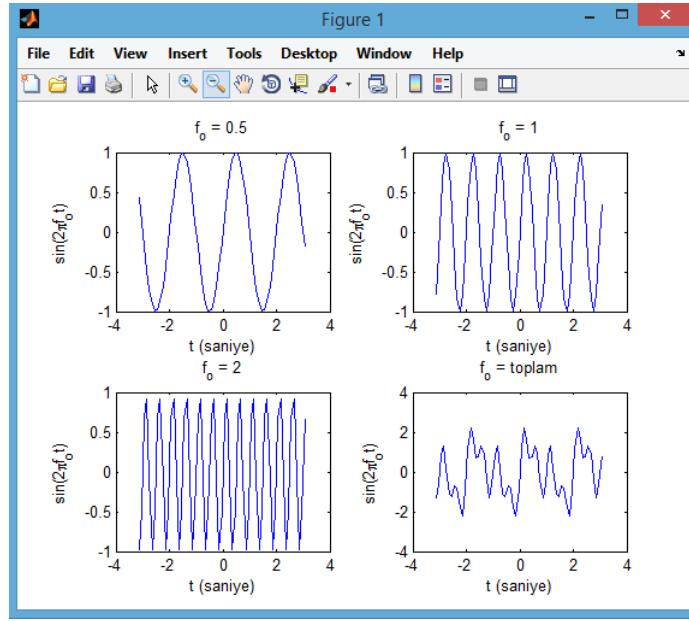
`plot` fonksiyonu grafik özelliklerinin nasıl düzenlendiğinin anlaşılması için seçilmiştir. MATLAB kütüphanesinde çizgisel grafiklerle ilgili çok sayıda fonksiyon bulunmaktadır. Bunlar `semilogx`, `semilogy`, `loglog`, `errorbar`, `comet`, `stem`, `stairs` şeklinde sıralanabilir. Bunların her birinde veri girişleri ile ilgili değişik biçimler gerekiyor olabilir. Sayılan fonksiyonlardan bazıları ile ilgili uygulamalar dördüncü bölümde verilecektir. Fonksiyonların kullanımları ile ilgili bilgi ve örnekler MATLAB yardım belgeleri içinde aranarak bulunabilir.

Bir pencerede birden çok grafiğin görüntülenmesi zaman zaman başvurulan bir durumdur. Bu amaçla MATLAB kütüphanesinde `subplot` fonksiyonu kullanılmaktadır. Bu fonksiyon ile grafik penceresi bir matris gibi düzenlenebilmektedir. Genel yazımı `subplot(m,n,p)` şeklindedir. Bu yazım ile m satır n sütun bir grafik alanı açılır ve p numaralı alan aktif hale getirilir. Pencerede alanlar soldan sağa doğru numaralandırılmaktadır. Aşağıda buna bir örnek verilmiştir.

```
function cokluciz
fo_1=0.5; fo_2=1; fo_3=2;
x=-pi:.1:pi;

y1=sin(2*pi*x*fo_1); y2=sin(2*pi*x*fo_2); y3=sin(2*pi*x*fo_3); y4=y1+y2+y3;
ciz(x,y1,2,2,1,fo_1); ciz(x,y2,2,2,2,fo_2); ciz(x,y3,2,2,3,fo_3);
ciz(x,y4,2,2,4,'toplam')

function ciz(X,Y,m,n,p,fo)
subplot(m,n,p)
plot(X,Y);
title(['f_o = ',num2str(fo)])
xlabel('t (saniye)')
ylabel('sin(2\pif_ot)')
```



Şekil 3. 2 subplot deyimi ile oluşturulan 2x2 grafik penceresi