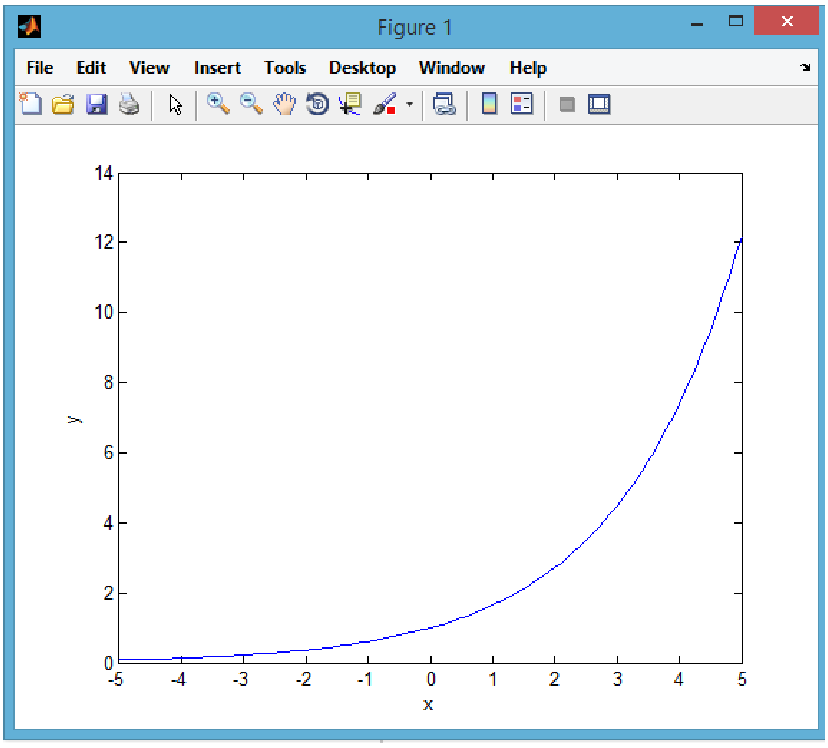
3.9 Grafik Düzenleme

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

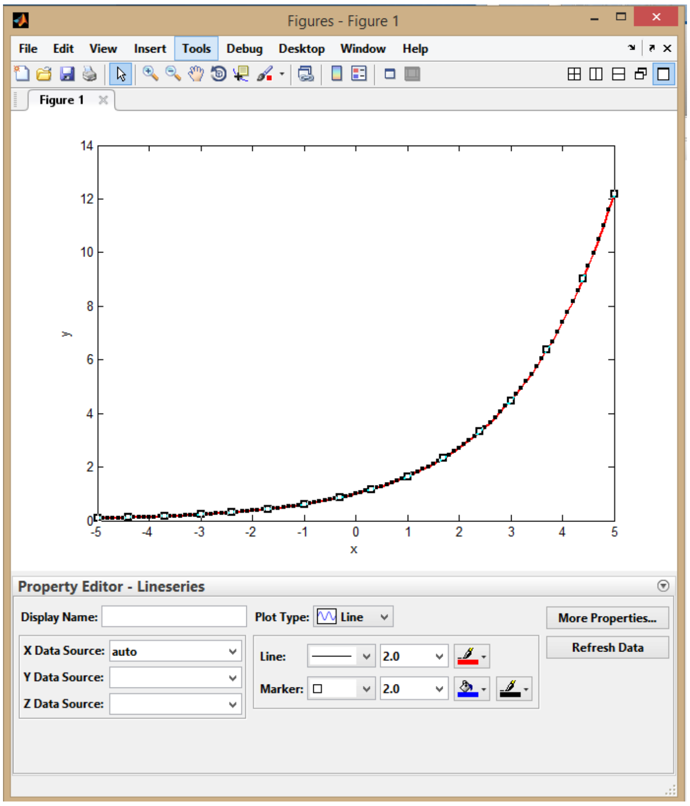
x-y dağılımı türünde bilgilerin çizilmesinde kullanılan en temel MATLAB fonksiyonudur. Bu tür bir çizim yapmak için birbiriyle aynı boyda iki yöney (vektör) girdi olarak sağlanmalıdır. Aşağıda bir x-y çifti için böyle bir çizimi yapan MATLAB ifadeleri verilmiştir.

```
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. 5 Şekil düzenleme

`plot` fonksiyonu ile üretilen şekillerdeki tüm düzenlemeler fonksiyonun kullanımı sırasında da yapılabilmektedir. Üretilen şekildeki tüm bileşenlerin görünümleri geçerli özellik-değer çiftleri ile değiştirilebilmektedir. Çizelge 3.11 bu tür değişikliklerin komut satırında ya da MATLAB programları içerisinde but 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 öğrenebileceği düzeydedir. Dolayısı ile örnekler burada verilenler ile sınırlı tutulacaktır. `plot` fonksiyonunun kullanılması ve penceredeki bileşenlerin düzenlemesi 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. 11 plot fonksiyonu ile üretilen grafiklerin bazı özelliklerinin düzenlemesi

Tanımlayıcı	Açıklama
`plot(x,y,'r')`	Eğri rengi kırmızı olsun. Diğer bazı renkler siyah : `'k'`, mavi : `'b'`, yeşil: `'g'`, sarı: `'y'`, camgöbeği: `'c'`, fuşya: `'m'`, , beyaz: `'w'`
`plot(x,y,'k+')`	Eğri siyah ve veri işaretçisi + olsun. Veri işaretçiler için bazı seçenekler çember : `'o'`, yıldız : `'*'`, çarpı: `'x'`, kare: `'s'`, baklava: `'d'`, nokta: `'.'`
`plot(x,y,'g--s')`	Eğriyi kesikli çizgiler ile göster, yeşil yap ve kare işaretçiler koy. Diğer bazı çizgi türleri: düz çizgi : `'-'`, kesikli `'--'`, kesikli-noktalı : `'-.'`, noktalı: `':'`, çizgi yok: `' '`
`plot(x,y,'b','LineWidth',2)`	Eğri mavi renk olsun, çizgi kalınlığı 2 punto olsun

`t=0:pi/20:2*pi;`
`plot(t,sin(t),'mo',...`
`'LineWidth',2...`
`'MarkerEdgeColor', 'k'...`
`'MarkerFaceColor', 'g'...`
`'MarkerSize', 10);`
	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. 12 Bazı grafik düzenleme deyimleri

Deyim	Açıklama	Örnek

`hold on`	Geçerli grafik penceresini tutarak bundan sonra gelecek çizim işlemlerinin aynı pencereye yapılmasını sağlar. `hold off` ile bu özellik kapatılabilir.	`x=-2*pi:0.01:2*pi;`
		`y1=sin(x);`
		`y2=cos(x);`
		`plot(x,y1,'r')`
		`hold on`
		`plot(x,y2,'b')`

`grid on`	x ve y yönlerinde grid çizgileri eklenir. `grid off` seçeneği ile kapatılabilir.	`x=-2:.15:2;`
		`plot(x,x.^2)`
		`grid on`

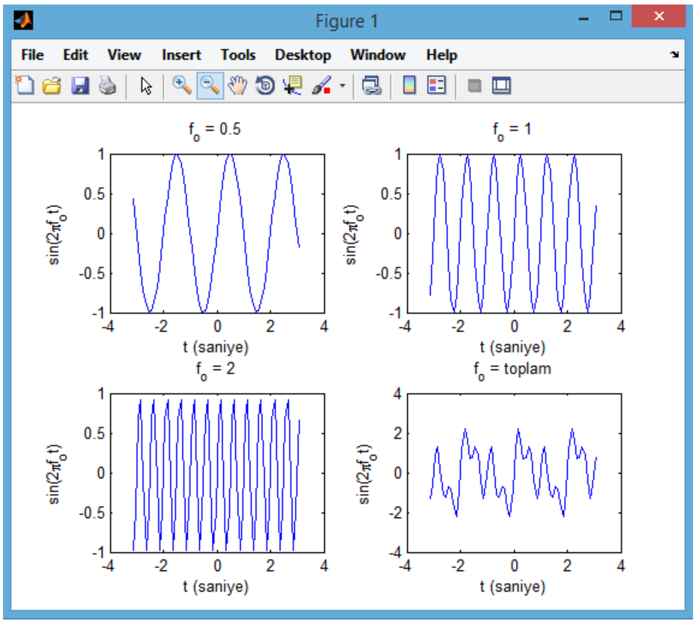
`title`	Grafik penceresi üzerine bir başlık eklenir.	`t=0:5;`
		`plot(t,exp(t))`
		`title('Üstel Fonksiyon')`

`legend`	Grafik üzerindeki verilerin ne anlama geldiğini gösteren bir gösterge (lejand)	`legend('sin(x)', 'cos(x)')`
`figure`	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`, `commet`, `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. 6 `subplot` deyimi ile oluşturulan 2x2 grafik penceresi

4. UYGULAMALAR

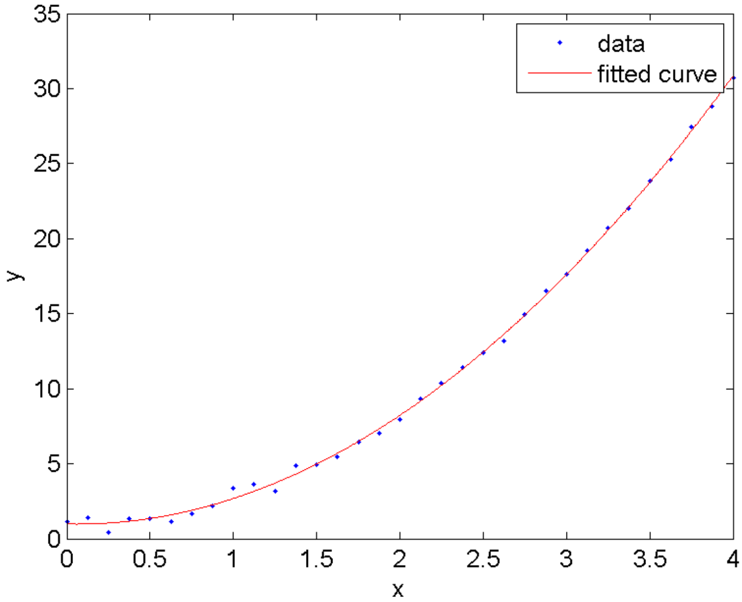
4.1 Aradeğer Bulma ve Veri Gridleme

Günümüz mühendislik ve bilimsel ölçme işlemleri sürekli olayların frekans, uzaklık, zaman gibi bağımsız bir değişkene bağlı olarak ayrık noktalarda örneklenmesi ile yürütülür. Ölçme sıklığının artırılması ile sürekli olaya en yakın bir kayıt elde edilebilir. Bunun tersi olarak yeteri sıklıkta örneklenmeden yapılan bir kayıt, ölçülen süreçten tamamen farklı bir davranış sergileyebilir. Ölçme sıklığı (örnekleme) çok sayıda etmene bağlıdır. Ancak birçok durumda örnek sayısını artırmanın, verileri saklama, ölçme maliyeti, veri işlem gereksinimleri gibi konularda zorluklar çıkaracağı açıktır. Bu nedenlerden, süreci yeterince yakınlıkta tanımlayacak örnekleme ve ölçme aralıklarının bulunması kendi başına bir mühendislik problemi olmaktadır.

Ölçme aralığı ve buna bağlı başka parametreleri saptanarak ayrık noktalarda ölçülmüş veriler, ilgili süreci tanımlama, bir mühendislik ya da bilim problemini çözme, bir durumu ortaya koyma gibi amaçlarla çeşitli süreçlerden geçirilebilmektedir. Bu süreçler sırasında gerek gösterim gerekse hesaplama amaçları ile ölçüm yapılmamış bir zaman, uzaklık ya da frekans değerindeki veriye gereksinim duyulabilmektedir. Ölçülmüş noktalardaki veri davranışından yola çıkılarak ara bir noktadaki değerler çeşitli yollarla hesaplanabilmektedir. Bunlardan ilki veriye davranışına uygun bir sürekli fonksiyon çakıştırmak ve bu sayede istenilen herhangi bir bağımsız değişken değerine karşılık gelen verinin hesaplanmasını olanaklı kılmaktır. MATLAB araç kutuları içerisinde yalnızca bu amaçla geliştirilmiş bir araç kutusu bulunmaktadır (Curve Fitting Toolbox). Bu araç kutusu arayüzüne, komut satırından `cftool` yazılarak erişilebilir. Arayüz açılmadan da araç kutusundaki fonksiyonlar ayrı ayrı çağırılarak kullanılabilmektedir. Aşağıdaki örnekte ikici dereceden bir fonksiyon oluşturularak bir miktar gürültü eklenmiştir. Oluşturulan bu veri, deneme amaçlı kullanılarak ikinci dereceden bir polinomun çakıştırılması ile ilgili MATLAB programı aşağıda verilmiştir. Polinoma çakıştırma işleminin uygulamadaki karşılığı veri ile uyumlu polinomun katsayılarının hesaplanmasıdır. Verinin davranışına bakılarak değişik derecelerden polinomlar çakıştırma fonksiyonu olarak seçilebilir. Polinoma çakıştırma işlemi için Bölüm 3.2'de alıştırmalar bölümünde verilen denklem sisteminin oluşturularak çözülmesi de kullanılabilir.

```
x=0:.125:4;
y=2*x.^2-0.5*x+1;
y1=y+0.25*randn(size(y));
f=fit(x',y1','poly2');%'poly1': doğru 'poly3': üçüncü derece ...
f
plot(f,x,y1)
```

```
f =

     Linear model Poly2:
     f(x) = p1*x^2 + p2*x + p3
     Coefficients (with 95% confidence bounds):
       p1 =       1.978  (1.895, 2.061)
       p2 =     -0.4339  (-0.7795, -0.08833)
       p3 =      0.9811  (0.6824, 1.28)
```



Şekil 4. 1 Bir veri kümesine ikinci dereceden bir polinom çakıştırılması. Veriler mavi noktalar ile çakıştırılan polinom ise düz kırmızı eğri ile gösterilmiştir.

Sayısal veriye `f(x) = p1*x^2 + p2*x + p3` gibi bir polinom çakıştırıldığında x için herhangi bir değer verilerek veriye o noktada bir yaklaşım sağlanabilir. Bu işlemin ne anlama geldiği Şekil 4.1'de verilen veri noktaları ve ona çakıştırılan eğrinin davranışına bakılarak anlaşılabilir. Veriye davranışına bağlı olarak, üstel bir fonksiyon, sinüzoidallerin toplamı, Fourier serisi gibi farklı sürekli fonksiyonlar da çakıştırılabilir.

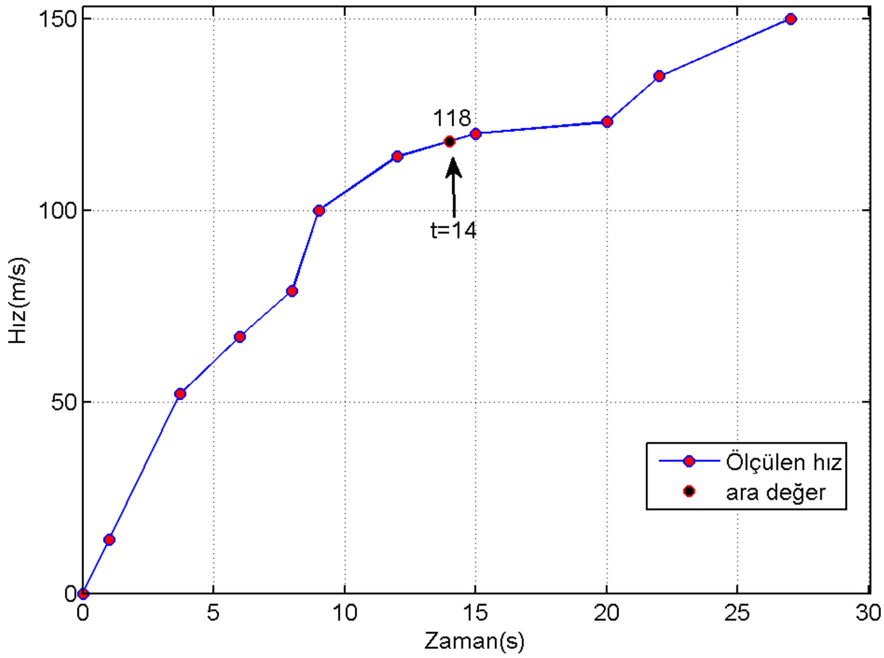
MATLAB kütüphanesinde aradeğer bulma ile ilgili kullanılan bir diğer fonksiyon `interp1` adını taşımaktadır. Fonksiyon doğrusal ara değer bulma yöntemini kullanmaktadır. Fonksiyonun genel kullanımı aşağıdaki gibidir:

```
di = interp(x,y,xi);
```

Yukarıdaki yazılışta `x`'e bağlı `y` değerlerinden yola çıkılarak `xi` gibi bir ara noktadaki veri kestirilmeye çalışılmaktadır. Hesaplanan değer `di` değişkenine aktarılmaktadır. Aşağıdaki örnek kullanımın daha iyi anlaşılmasına yardımcı olacaktır.

```
clear all;close all;clc
% Bir aracın zamana karşı ölçülmüş hız değerleri
t=[0 1 3.7 6 8 9 12 15 20 22 27];
v=[0 14 52 67 79 100 114 120 123 135 150];

%Aracın t=14 saniyedeki hızını bulunuz.
ti=14;
vi=interp1(t,v,ti);%t=14 deki aradeğerin hesaplanması
```

```
plot(t,v,'b-o','MarkerFaceColor','r')
hold on
plot(14,vi,'r o','MarkerFaceColor','k')
%grafiğin düzenlenmesi
grid on;box on
xlabel('Zaman(s)')
ylabel('Hız(m/s)')
legend('Ölçülen hız','ara değer')
```

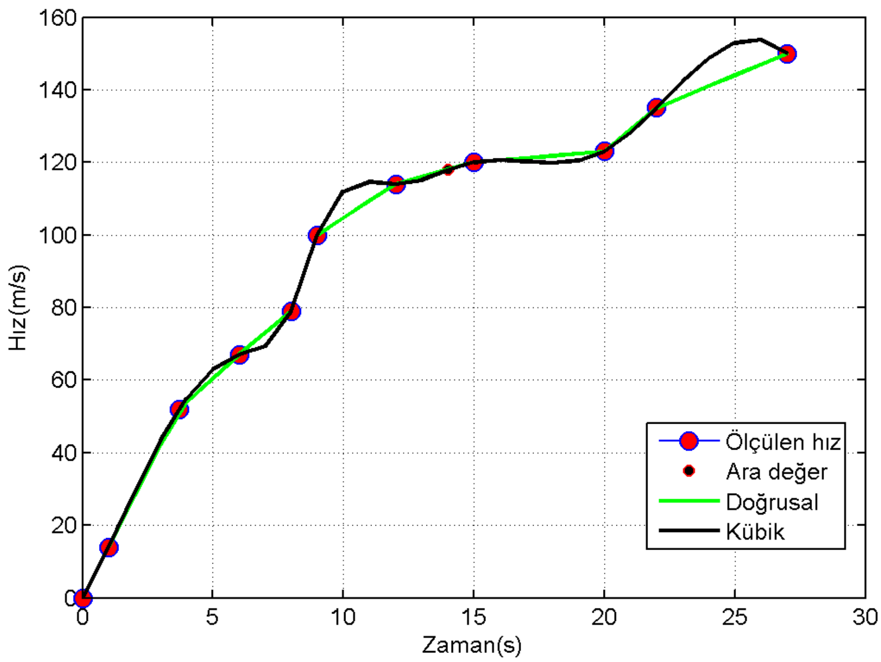


Şekil 4. 2 Bir aracın hızının ölçülmeyen bir ara zaman değerinde hesaplanması

Bu amaçla kullanılabilecek bir diğer MATLAB fonksiyonu `spline` ismi ile kütüphanede yer almaktadır. Fonksiyonun, birbirine komşu iki veri noktası arasında doğrusal değil üçüncü derece bir polinom davranışı gösterdiği varsayımı ile çalışır. Kübik ara değer bulma fonksiyonu önceki örnek için

```
vii=spline(t,v,ti)
vii=
    117.7473
```

şeklinde çağırılarak kullanılabilir.

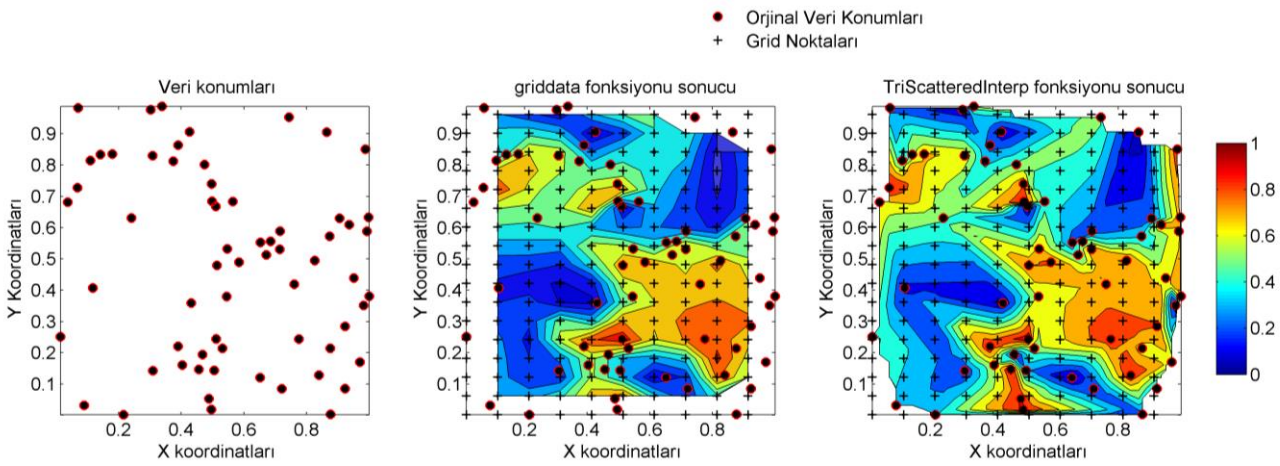


Şekil 4. 3 Doğrusal ve kübik aradeğer bulma fonksiyonları ile yeniden örneklenen veri

Birden çok parametre, konum ya da değişkene bağlı verilerin belirli bir sistematik ile ara değerlerinin bulunması işlemi yukarıda verilen örneklerden biraz daha karışık hale gelebilmektedir. Örnek olarak bir çalışma alanında x ve y konumlarına bağlı ölçülen bir jeofizik veriyi ele alabiliriz. Bu tür bir veri kümesinden beklenilen çalışma alanını x ve y yönlerinde eşit aralıklarla taramasıdır. Ancak bu çok zaman mümkün olmaz. Ölçülen verilerin görüntülenmesi ya da işlenmesi sırasında bu verilerin bir grid (ızgara) üzerine düşen noktalardaki değerlerine gerek duyulur. Bu amaçla veri gridleme şeklinde adlandırdığımız yöntem grubuna başvururuz. Burada işlemin ayrıntılarına girilmeyip yalnızca MATLAB dilindeki uygulamasına örnek verilecek ve gerekli açıklamalar örnek üzerinde yapılacaktır.

```
% Veri Gridleme ve Konturlama Uygulaması
clear all;close all;clc
% Sınama verisi
% Rastgele x,y konumlarında ölçüldüğü varsayılan z değerleri üretiliyor
x=rand(64,1);y=rand(64,1);z=rand(64,1);
dx=.1; % x yönünde grid aralığı
dy=.06;% y yönünde grid aralığı
% Izgaranın düğüm noktaları oluşturuluyor
xi=min(x):dx:max(x);
yi=min(y):dy:max(y);
%grid ızgarası oluşturuluyor
[X,Y]=meshgrid(xi,yi);
%Veri griddata fonksiyonu ile gridleniyor x ve y sırası ile verilerin konumları, z
bu konularda tanımlı veri, X ve Y ise verilerin yeniden gridleneceği ızgara
noktalarıdır.
Z=griddata(x,y,z,X,Y,'cubic'); Tırnak içinde gridleme işlemi ile ilgili bir seçenek
yazılmıştır. Diğer seçenekler 'linear' ve 'nearest' şeklindedir.
```

```
%verilerin konumları çizdiriliyor
subplot(131)
plot(x,y,'r o','MarkerFaceColor','k');box on;axis image
xlabel('X koordinatları');ylabel('Y Koordinatları')
%grid ızgarası çizdiriliyor
subplot(132)
plot(x,y,'r o','MarkerFaceColor','k');box on;axis image
hold on
plot(X,Y,'k +')
plot(X,Y,'k .')
%gridlenmiş değerler çizdiriliyor
contourf(xi,yi,Z)
alpha(.75);
legend('Orjinal Veri Konumları','Grid Noktaları')
xlabel('X koordinatları');ylabel('Y Koordinatları')
```



Şekil 4. 4 Bir düzlemde dağınık olarak bulunan veri konumları kullanılarak yapılan gridleme işleminin sonucu

```
%griddata fonksiyonu MATLAB'ın sonraki sürümlerinde kaldırılacaktır. Onun
%yerine TriScatteredInterp fonksiyonu aşağıdaki gibi kullanılabilecektir.
F = TriScatteredInterp(x,y,z);
[qx,qy] = meshgrid(sort(x),sort(y));
qz = F(qx,qy);
%Sonuçların çizdirilmesi
subplot(133)
contourf(sort(x),sort(y),qz);hold on
plot(x,y,'r o','MarkerFaceColor','k');box on;axis image
plot(X,Y,'k +')
plot(X,Y,'k .')
```

Yukarıdaki örnekte, verilerin ölçüldüğü noktalar dışında kalan ve bir ızgaranın köşe noktalarında yer alan yeni noktalarda veriye bir yaklaşım yapılmaktadır. Bu örnekte birbirine komşu noktalar üçgenler oluşturacak şekilde birleştirilmekte ve üçgenin kenarları üzerinde kübik yaklaşımla aradeğerler hesaplanmaktadır. Programda kullanılan `meshgrid` fonksiyonu verilen x ve y değerlerinin olası tüm birleşimlerinden oluşan bir grid ağı ya da ızgarası oluşturmaktadır. `contourf` fonksiyonu ise xi ve yi konumlarına bağlı olarak hesaplanan yeni değerleri renklendirilmiş bir kontur çizgileri grubu ile görselleştirmek amacıyla kullanılmıştır. Bu fonksiyonlar hakkında geniş bilgi için komut satırında `doc contourf` ve `doc meshgrid` yazılabilir. Örnekte 2-boyutta veri gridlemek için kullanılan griddata fonksiyonuna benzer olarak 3-boyutlu veri gridleme işlemi için `griddata3` fonksiyonu aşağıdaki yazımla kullanılabilmektedir.

```
w = griddata3(x,y,z,v,xi,yi,zi)
```

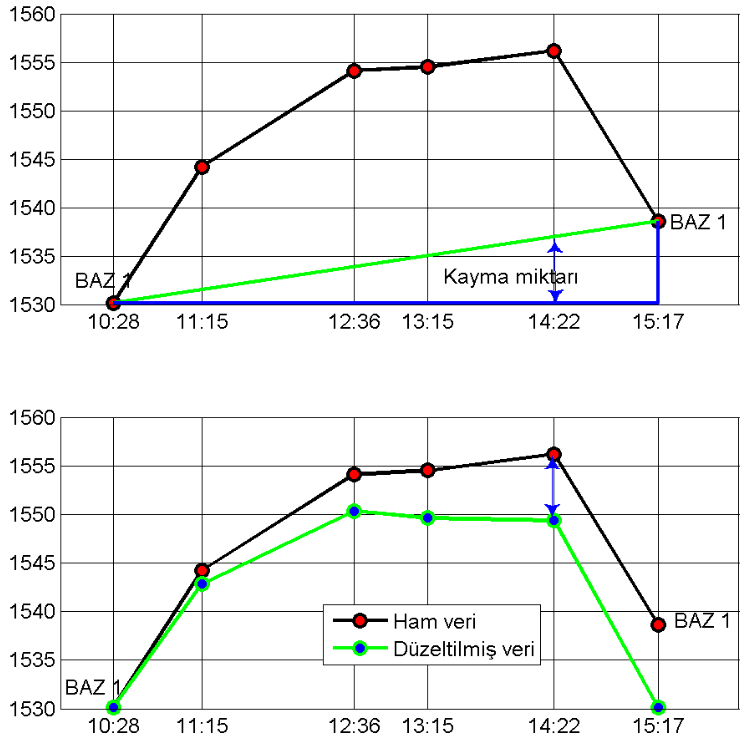
Burada `x,y` ve `z` verinin ait olduğu konum koordinatları, `v` verinin kendisi, `xi,yi` ve `zi` ise verinin yeniden gridleneceği koordinatlardır. `griddata` ve `griddata3` MATLAB'ın yeni sürümlerinde kaldırılacağından `TriScatteredInterp` fonksiyonunun kullanımı da ayrıca verilmiştir.

Veri gridleme işleminde var olan konum ve bilgilerden yola çıkılarak veri bulunmayan bir konumdaki veriye yaklaşım yapıldığından hesaplama yapılan nokta yakınında veri bulunması gereklidir. Başka bir deyişle gridleme yapılmak istenen nokta ya da noktalar veri noktaları arasında kalmıyor ve çalışma alanı dışına taşıyorsa bu noktalarda gridleme yapılamayacağı gözönüne alınmalıdır.

Alıştırmalar

1. Gravite ölçümlerinde genellikle ölçüm cihazından kaynaklanan bazı hatalar ortaya çıkar. Bu hata belirli aralıklarla baz olarak adlandırılan sabit bir noktaya dönerek ölçümün tekrarlanması yoluyla düzeltilir. Beklenti, aynı konumda aynı okumanın tekrarlanabilmesidir. Ancak bu gerçekleşmez. Baz noktasına tekrarlı gelişlerde okunan değerlere bir doğru çakıştırarak yapılacak düzeltme hesaplanabilir. Geliş zamanları arasındaki okumalara bu düzeltme eklenir ya da çıkarılır. Aşağıdaki okuma değerleri için bu düzeltmeleri yapan bir program yazınız.

İSTASYON	Saat	Okuma
BAZ1	**10:28**	**1530.15**
STA1	11:15	1544.20
STA2	12:36	1554.10
STA3	13:15	1554.50
STA4	14:22	1556.20
BAZ1	**15:17**	**1538.60**



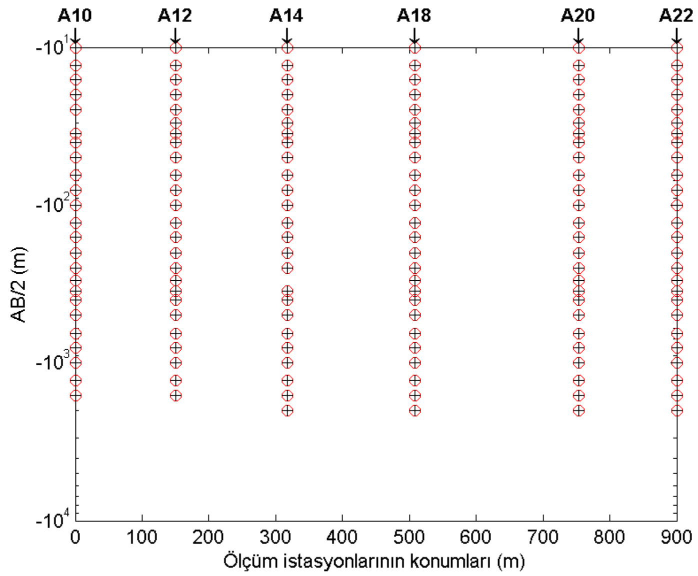
Şekil 4. 5 Gravite ölçümlerine yapılan kayma (drift) düzeltmesi

2. http://goo.gl/e3B1Uy adresinden indireceğiniz sıkıştırılmış dizin içinde sistematik olarak isimlendirilmiş veri dosyaları bulunmaktadır. Dosyalardan birinin içeriği aşağıdaki gibidir:

```
0  0  51
24
10          2        46
13          2        45
16          2        42.5
20          2        42
25          2        41.5
35          2        41
40          2        44
50          2        44
65          10       38
80          10       35
100         10       30
130         10       25.5
160         10       23
200         10       20
250         40       18.5
300         40       17
350         40       17.5
400         40       17
500         80       18.5
650         80       20.5
800         80       18
1000        200      15
1300        200      15.5
1600        200      16.5
```

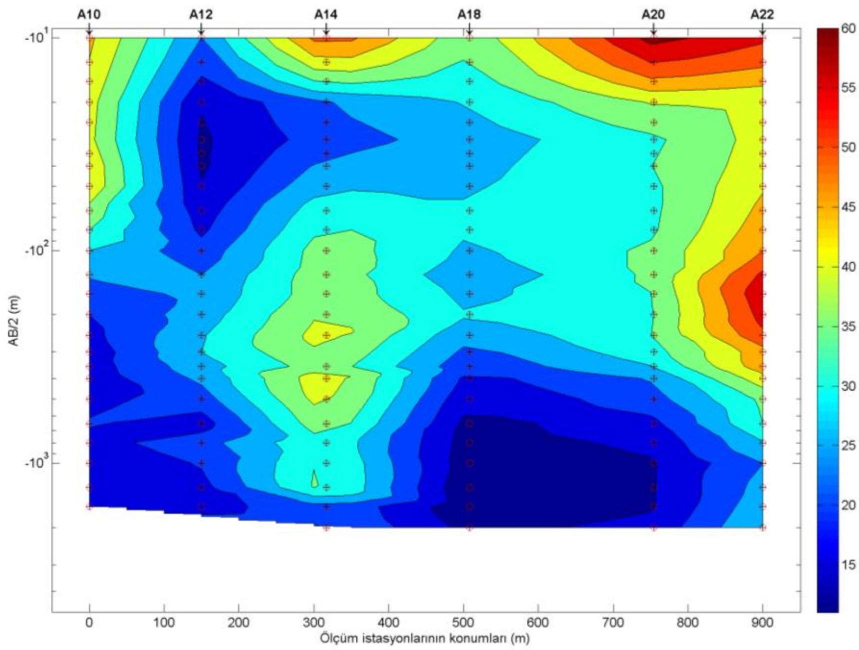
Dosya bir istasyonda ölçülmüş düşey elektrik sondajı verilerini içermektedir. Birinci satırda ölçüm istasyonunun x,y ve z konumları verilmektedir. İkinci satırda kaç adet veri olduğu bilgisi yer almaktadır. Sonraki satırların sütunlarında sırasıyla AB/2 (düşey eksen değerleri), MN/2 problemin çözümü ile ilgisiz bir ölçme parametresi ve ölçülen görünür özdirenç yer almaktadır.

Ölçüm noktalarının yatay ve düşey konumlarının görünümü aşağıdaki şekilde verilmiştir.



Şekil 4. 6 Ölçümlerin yatay ve düşey konumları

Verilen bilgiler ışığında 6 istasyonda ölçülmüş olan görünür özdirenç değerlerini yatay eksen istasyon konumları, düşey eksen AB/2 olacak şekilde gridleyiniz. Yatay grid aralığı 50 m, düşey grid aralığı ise 10m alınız. Sonuçları contourf fonksiyonu ile görselleştiriniz. Yükseklik değerlerini dikkate almayınız. Tüm işlemleri doğru yaptığınızda aşağıdaki gibi bir grafik elde etmeniz beklenmektedir.



Şekil 4. 7 Verilerin gridlenerek renklendirilmiş kontur alalıkları ile görselleştirilmesi

4.2 2B ve 3B Grafikler

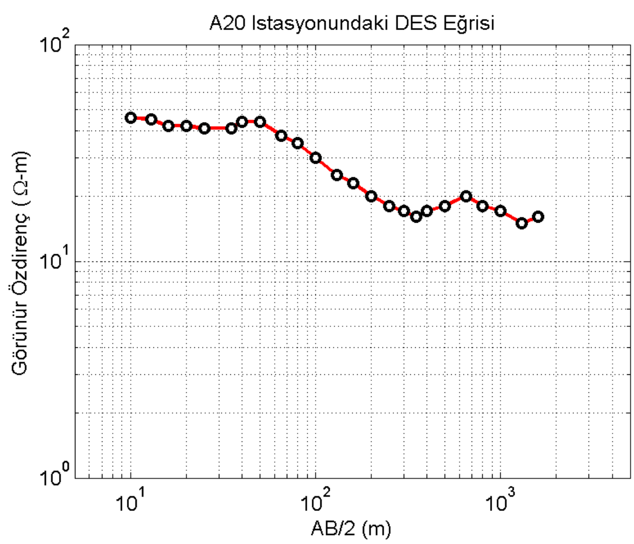
MATLAB grafikleri genel olarak çizgisel, yüzeysel ya da hacimsel olarak sınıflandırılabilir. Çizgisel grafikler bir x-y dağılımı veya bir büyüklüğün veya fonksiyonun bağımsız bir değişkene göre değişimini göstermek için kullanılabilir. Alan ya da yüzey türü grafikler ise genellikle x ve y gibi iki bağımsız değişkene bağlı bir büyüklüğün ya da fonksiyonun değişimini görselleştirmekte kullanılır. Hacim türü grafikler ise x,y ve z gibi üç bağımsız değişkene bağlı bir veri ya da fonksiyonun değişim ve dağılımını göstermek üzere kullanılmaktadır. Çizgisel grafikler üretmek için MATLAB'da sıkça başvurulan `plot` fonksiyonu daha önce çeşitli uygulama ve örnekler ile açıklanmıştır. Bunun dışındaki fonksiyonlar ile üretilen örnek grafikler ve açıklamaları aşağıda verilmiştir.

loglog

Her iki ekseni logaritmik değişen çizgisel grafikler oluşturmak için kullanılan bir fonksiyondur. Kullanılışı `plot` fonksiyonu ile aynıdır. Jeofizik verilerden düşey elektrik sondajı (DES) eğrileri çift logaritmik eksen üzerinde grafiklenirler. Bu şekilde bir örnek aşağıda verilmiştir. Tablonun solundaki hücrede yatay ve düşey eksen değerlerine karşılık gelen AB/2 ve görünür özdirenç değerlerini içeren dosya bulunmaktadır.

```
%AB/2   Gör. Özd.
10      46
13      45
16      42
20      42
25      41
35      41
40      44
50      44
65      38
80      35
100     30
130     25
160     23
200     20
250     18
300     17
350     16
400     17
500     18
650     20
800     18
1000    17
1300    15
1600    16
```

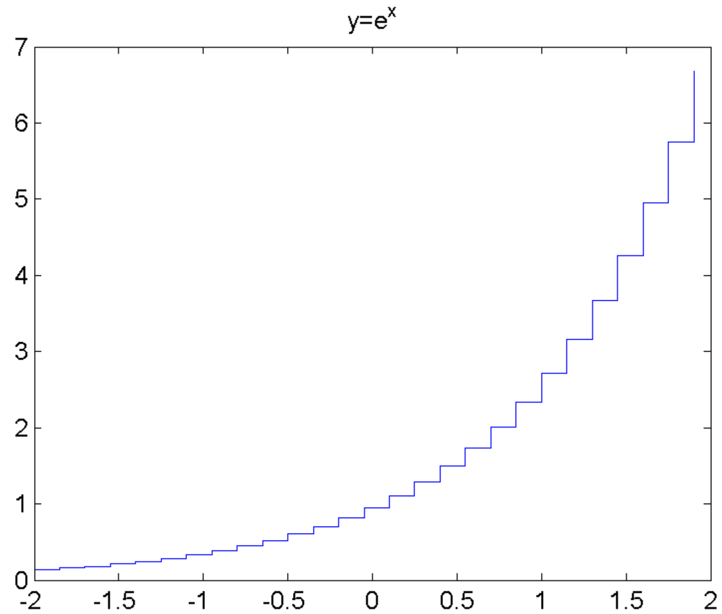
```
load des.txt
loglog(des(:,1),des(:,2),'ro','LineWidth',2,...
'MarkerFaceColor','w','MarkerEdgeColor','k');
axis([5 5000 1 100])
grid on
xlabel('AB/2 (m)')
ylabel('Görünür Özdirenç ( \Omega-m)')
title('A20 Istasyonundaki DES Eğrisi')
```



`semilogx` ve `semilogy` fonksiyonları da yukarıdaki örneğe benzer şekilde kullanılabilir. Bu fonksiyonlardaki fark, sırasıyla yalnız x- ve yalnız y-eksenlerinin logaritmik olmasıdır. Bu tür çizimler yarı logaritmik grafikler olarak adlandırılırlar.

stairs

Kelime anlamı basamaklar olan `stairs` kelimesi ile adlandırılan bu fonksiyon da çizgisel grafikler elde etmek için kullanılır. Aşağıdaki şekilde $y = e^x$ fonksiyonunun [-2,2] aralığındaki değişimi `stairs` fonksiyonu ile görüntülenmiştir.



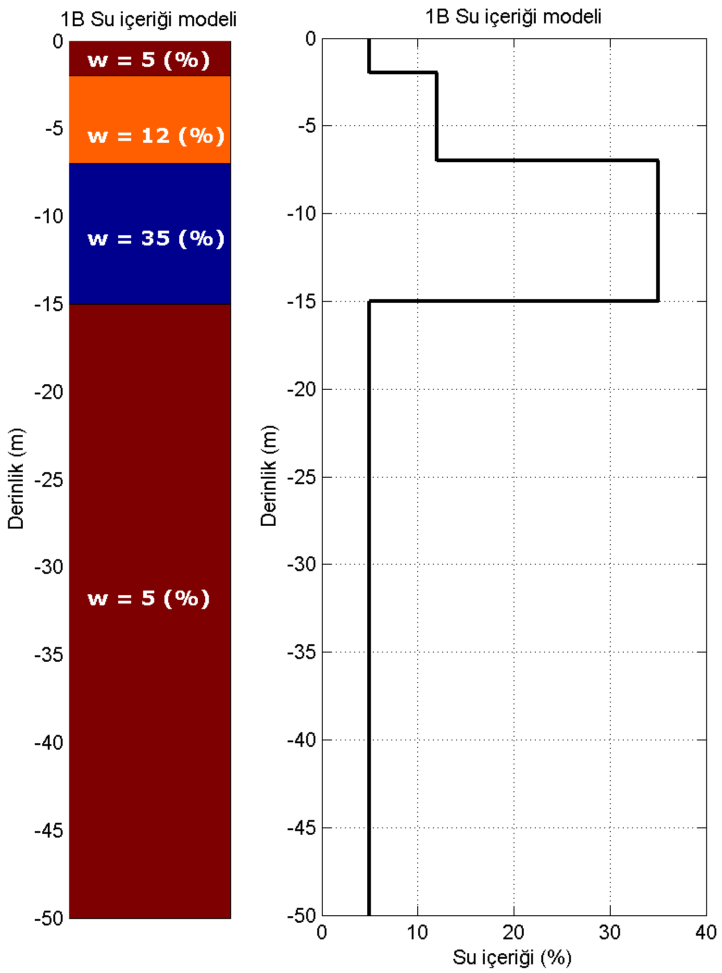
Şekil 4. 8 `stairs` MATLAB fonksiyonu ile oluşturulan çizgisel grafik

Fonksiyonun görsellik açısından uygun durumlarda kullanımı tercihe bırakılabilir. Ancak jeofizik veri ve modellerin görselleştirilmesi açısından bir boyutlu (1B) uygulamaarda daha kullanışlıdır. Fonksiyonun uygulaması olarak 1B jeofizik modellerin çizdirilmesi örneği aşağıda verilmiştir.

Bir boyutlu jeofizik model, yer içindeki fiziksel parametrenin yalnızca düşey yönde (yüzeyden yerin içine doğru) değiştiğinin kabul edildiği kavramsal bir modeldir. Geometri olarak yatay ve tekdüze katmanlardan oluştuğu var sayılır. Bu durumda model, katmanların kalınlıkları ve bu katmanların fiziksel özelliği ile tanımlanır. Yer yarı sonsuz kabul edildiğinden son katmanın kalınlığının da sonsuz olduğu varsayılır. Aşağıda yer içindeki katmanların hacimsel olarak su içeriğini gösterir 1B modelin parametreleri, bu parametreler kullanılarak su içeriğinin derinlikle değişimini gösteren grafiği çizdiren MATLAB programı ve programın çıktısı verilmiştir. Burada fonksiyonun belirtilen amaçla kullanılabilmesi için birinci fiziksel parametre iki kez tekrarlanmış ve kalınlığı sonsuz kabul edilen katmanın kalınlığı için de düşey eksenin en büyük değeri tekrar edilmiştir.

Katman	Su içeriği	Kalınlık
1	5	2
2	12	5
3	35	8
4	5	∞

```
p=[5 12 35 5];%Katman fiziksel parametreleri
thk=[2 5 8];  %Katman kalınlıkları
zmax=50;      %Grafiğin düşey eksen sınırı
renk='k';     %Çizgi rengi 'k': siyah

zp=cumsum(thk);
h1=stairs([p(1) p],[0 -zp -zmax],...
'LineWidth',2,'color',renk);
xlabel('Su içeriği (%)');
title('1B su içeriği modeli')
ylabel('Derinlik (m)');
grid on
axis([0 40 -50 0])
```



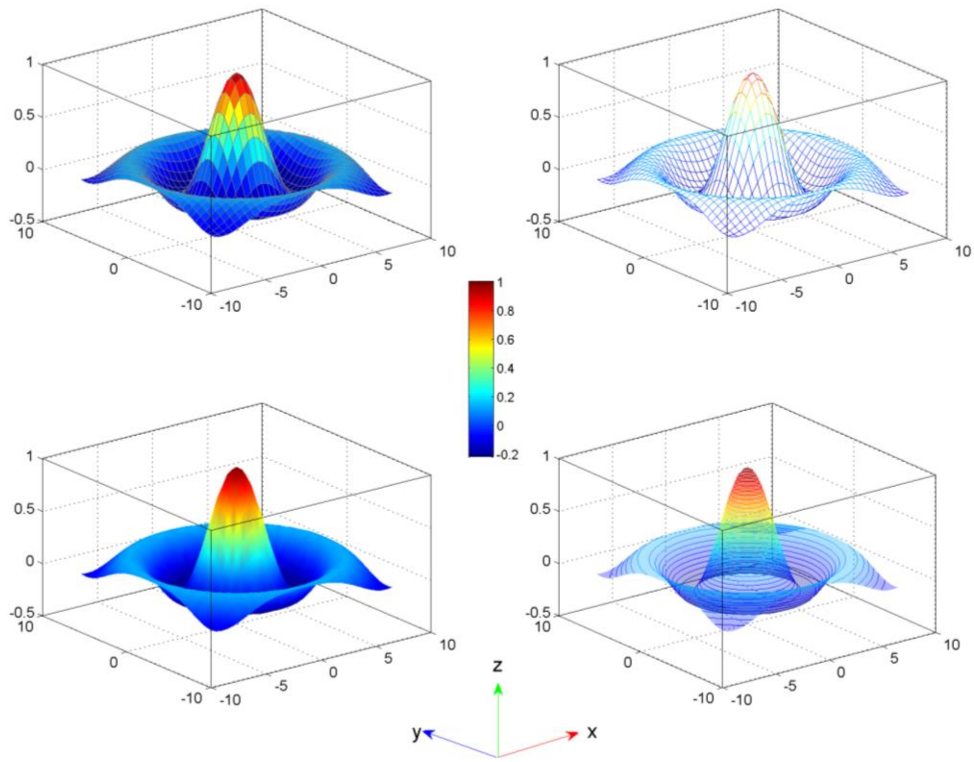
Şekil 4. 9 1B yer modelinin parametreleri ve `stairs` fonksiyonu ile çizimi

surf-mesh-contour3

Üç-boyutlu gölgelendirilmiş yüzey çizen MATLAB fonksiyonlarıdır. Bu tür çizimler genellikle bir arazi topoğrafyasının gösterilmesinde, ölçülen bir fiziksel büyüklüğün (manyetik alan, yer çekim ivmesi, radyoaktivite gibi) enlem ve boylama göre değişimini göstermek üzere hazırlanırlar. Dolayısı ile girdi olarak iki yatay eksen konumu ile bu konumda ölçülen fiziksel büyüklüğe gereksinim vardır. Fonksiyonların kullanımları aşağıdaki örnek üzerinde açıklanmıştır.

```
close all,clear all
% Sınama verisi x,y koordinatları üretiliyor
[x,y] = meshgrid(-8:.5:8);
r = sqrt(x.^2 + y.^2) + eps;
% sinc fonksiyonu iki boyutta hesaplanıyor
z = sin(r)./r;
% surf ile yüzey oluşturuluyor
subplot(2,2,1)
surf(x,y,z,'EdgeColor',[.5 .5 .5]);box on

% Yüzey ağ olarak çiziliyor
subplot(2,2,2)
mesh(x,y,z);
box on

% Özel gölgelendirme ve ışıklandırma seçenekleri
subplot(2,2,3)
surf(x,y,z,'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong')
box on

% Yüzeyle birlikte 3B kontur çizgileri ekleniyor
subplot(2,2,4)
surf(x,y,z)
shading interp % Diğer seçenekler shading flat | shading faceted
box on
alpha(.5) % Saydamlık ayarlanıyor
hold on
% 3B kontur çizgileri ekleniyor
contour3(x,y,z,32);
box on
```

Örnekte sinc fonksiyonunun x ve y koordinatlarına bağlı değerleri hesaplanarak kullanılmıştır. Fonksiyonların kullanımları ile ilgili çeşitli seçenek ve ayarlar ile ilgili açıklamalar yukarıdaki program parçası içerisinde verilmiştir.



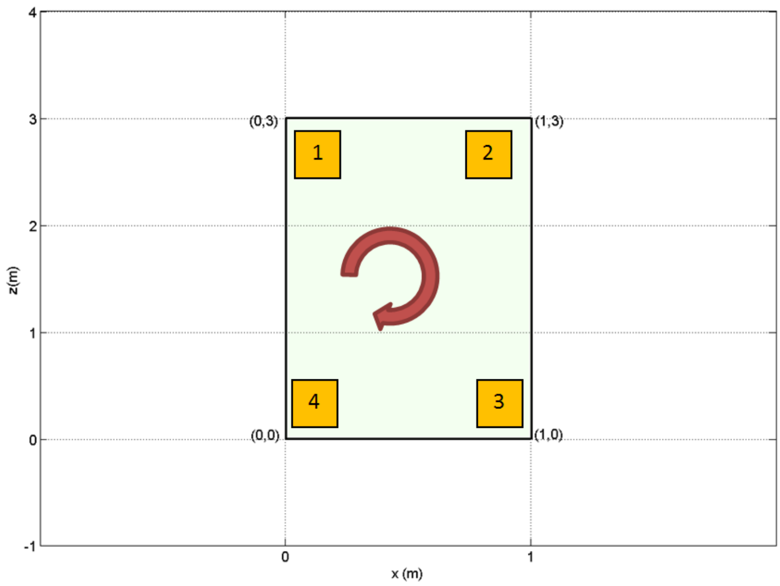
Şekil 4. 10 surf, mesh ve contour3 fonksiyonları ile üretilen yüzey grafikleri

patch

MATLAB kütüphanesinde 2B harita ve modellerin çiziminde başlıca, `contourf, imagesc, surf, pcolor` gibi kütüphane fonksiyonları kullanılmaktadır. Bu fonksiyonlar tüm meslek ve bilim gruplarının kullanımı için genel ilkelerle tasarlandığından jeofizik model ve verilerin görselleştirilmesi ile ilgili kimi gereksinimleri karşılayamazlar. Örneğin `imagesc` görüntü işleme araç kutusundan bir fonksiyon olup asıl olarak fotoğraf ve görüntülerin görselleştirilmesi için kullanılmaktadır. Ancak birçok jeofizik model ve haritanın hızlı bir görselinin hazırlanmasında kullanılabilecek bir fonksiyondur. Fonksiyon her bir x-y konum çifti için eşit büyüklükte bir hücre (pixel gibi düşünülebilir) tanımlar. Bu hücre, ön tanımlı bir renk skalasının atanan fiziksel büyüklüğe karşılık gelen rengi ile boyanır. Tüm hücreler eşit boyutlarda olduğundan x ve z yönlerinde eşit aralıklarla tanımlanmamış veri noktalarının çiziminde görsel yanılgılara neden olabilir. Diğer taraftan jeofizik modellerde topoğrafik değişimler gibi dikdörtgen geometriyi bozucu bilgilerin de temsil edilmesi gerekebilir. Bu nedenle diğer fonksiyonlar için bir uygulama veya örnek verilmeden jeofizik görüntüleme ile ilgili genel gereksinimleri karşılayabilen bir fonksiyon olan `patch` ile ilgili uygulamalara değinilecektir.

Kelime anlamı yama olan `patch` fonksiyonu kapalı ve renklendirilmiş poligonlar çizdirilmesi temelinde çalışmaktadır. Poligonlar istenilen sayıda köşe ile tanımlanabileceğinden geometri ile ilgili genel bir esneklik elde edilmiş olacaktır. Kapalı poligonlar ya da onların birleşiminden üç boyutlu (bir küpün yüzeyleri gibi) şekiller de elde edilebileceğinden fonksiyon 3B çizimlerde de kullanılabilmektedir. Fonksiyonun çok fazla seçenek ile kullanımı söz konusudur. Ancak burada basit, anlaşılır ve karşılaşılma sıklığına bağlı olarak çeşitli örnekler verilecektir.

Dikdörtgen şeklindeki bir poligon tanımlamak ve bir parametre değerini bu poligona atamak için aşağıdaki basit örnek ile başlanabilir. x-z düzleminde yer alan bir dikdörtgen aşağıdaki gibi olsun



Şekil 4. 11 Dikdörtgen şeklinde bir poligonun köşe noktalarının numaralandırılması

Dört köşesi olan bu poligon 4 adet x-z koordinat çifti ile tanımlanabilir. Grafikteki konumları incelendiğinde aşağıdaki MATLAB deyimleri ile bu tanımlama yapılabilir. Genel bir kural olmamakla birlikte numaralandırmada saat yönünde ilerlenmesi önerilmektedir.

```
x=[0 1 1 0]';% Koordinatlar sütunlarda verilmelidir. Bu yüzden transpozu alınıyor
z=[3 3 0 0]';
```

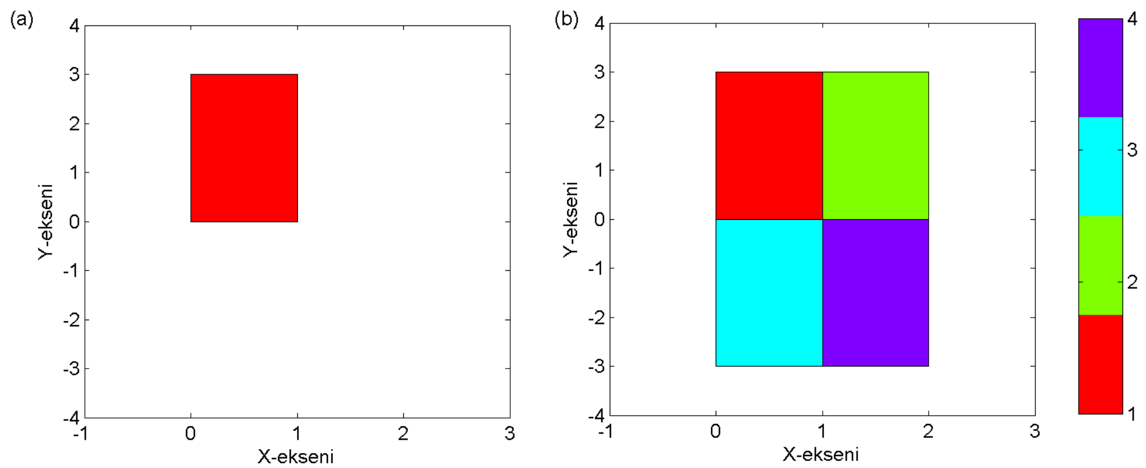
Bu poligonun her bir tanımlı (x,z) noktasına görüntülenmek istenen bir sayısal büyüklük atanabilir. Atanacak parametrelerin değerleri gösterim amaçlı 1 olsun.

```
p=[1 1 1 1]';
```

Çizim için gerekli geometrik ve fiziksel tanımlamalar yapıldıktan sonra patch fonksiyonu aşağıdaki gibi çağrılarak grafik oluşturulabilir:

```
patch(x,z,p)
```

Bu işlemin sonucunda Şekil 4.12. a'da verilen grafik elde edilir. Yukarıda verilen poligona benzer olarak başka poligonlar tanımlanarak bir örüntü oluşturulabilir. Burada verilen örnekler hücrelerden oluşan 2B jeofizik modellerin çizimine temel alınabilir.

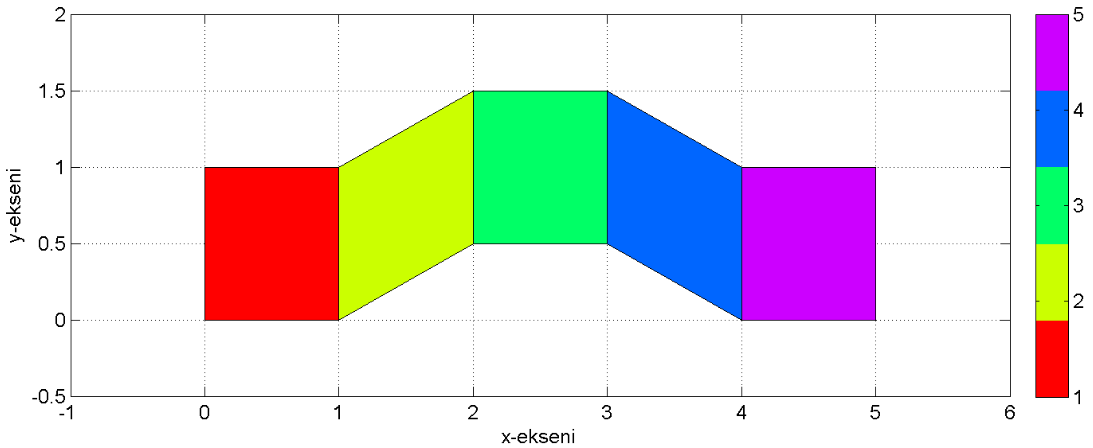


Şekil 4. 12 patch fonksiyonu ile oluşturulan (a) bir hücreden ve (b) dört hücreden oluşan model parçaları

Jeofizik modellerin geometrisi kimi zaman topoğrafik bilgi ile birlikte şekillenir. Bu durumda tüm hücreler dikdörtgen olamaz. `patch` fonksiyonu ile bu tür bir geometriyi oluşturmak olanaklıdır. Aşağıda bu şekilde bir geometri ve bu geometriye bağlı parametre dağılımını oluşturmak için yazılan MATLAB programı verilmiştir:

```
x=[0 1 2 3 4;1 2 3 4 5;1 2 3 4 5;0 1 2 3 4]
z=[1 1 1.5 1.5 1;1 1.5 1.5 1 1;0 .5 .5 0 0;0 0 .5 .5 0]
p=repmat([1 2 3 4 5],4,1)
patch(x,z,p)
```

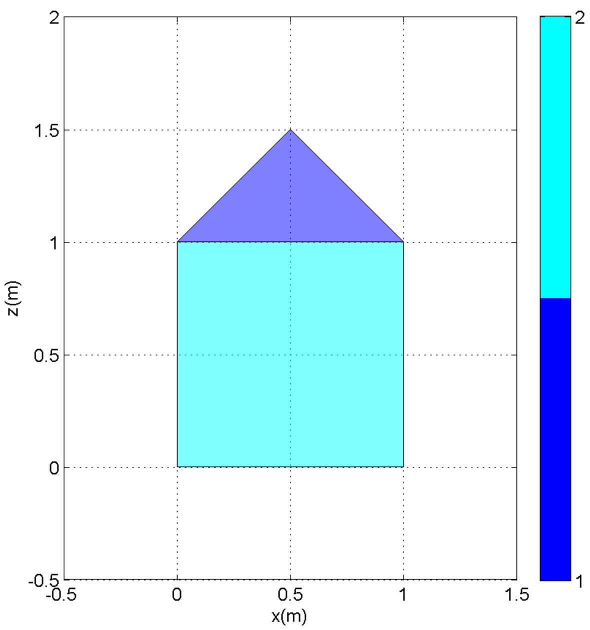
```
x =

     0     1     2     3     4
     1     2     3     4     5
     1     2     3     4     5
     0     1     2     3     4


z =

    1.0000    1.0000    1.5000    1.5000    1.0000
    1.0000    1.5000    1.5000    1.0000    1.0000
         0    0.5000    0.5000         0         0
         0         0    0.5000    0.5000         0


p =

     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
```



Şekil 4. 13 patch fonksiyonu ile farklı geometride dörtgenlerden oluşan bir grafik

Farklı sayıda köşesi olan poligonlardan bir çizim oluşturmak da olanaklıdır. Buna göre bir üçgen ve bir dörtgenden oluşan bir şekil aşağıdaki biçimde çizilebilir:

```
xu=[0 1 .5];     % Ucgenin x koordinatları
zu=[1 1 1.5];    % Ucgenin z koordinatları
xd=[0 1 1 0];    % Dörtgenin x koordinatları
zd=[0 0 1 1];    % Dörtgenin z koordinatları
patch(xu,zu,[1 1 1]') %Ucgen ciziliyor
hold on
patch(xd,zd,[2 2 2 2]')%Dörtgen çiziliyor
colormap(jet(2))
axis image;box on;grid
xlabel('x(m)');ylabel('z(m)')
alpha(.5) %Saydamlık ayarlanıyor
```



Şekil 4. 14 Farklı geometride poligonlardan oluşan grafiğin `patch` fonksiyonu ile oluşturulması

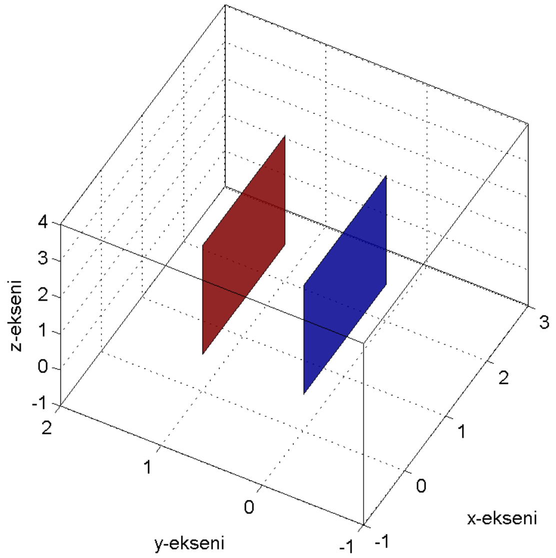
İki-boyutlu düzlemde verilen örnekler farklı birleşimlerle çoğaltılabilir ya da çeşitlendirilebilir. Bu noktadan sonra üç-boyutlu poligonlar ile yapılabilecek çizimlere örnekler verilerek yine 3B jeofizik modellerin oluşturulmasına yardım edecek bilgiler ortaya konulacaktır. Üç boyutlu uzayda bir P noktasının konumu (x,y,z) şeklinde üç koordinat ile belirlenebilir. Dolayısı ile önceki çizimden farklı olarak poligonların köşe noktalarının y koordinatları da belirtilmek durumundadır. Buna göre y-yönünde birbirinden 1 birim uzakta iki dörtgen poligon (Şekil 4.15) çizilmek istenirse

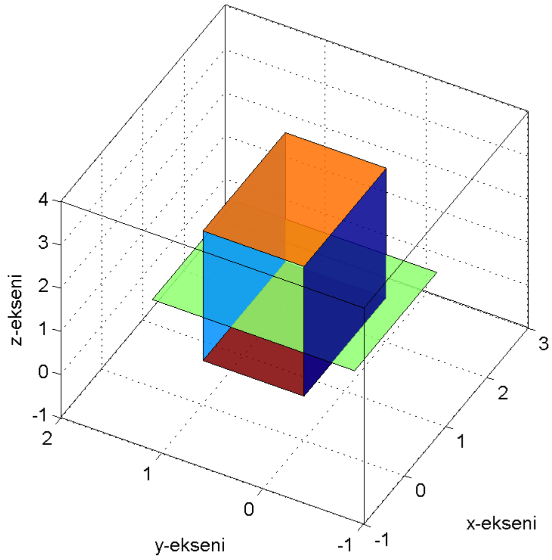
```
close all;clear all
% Şekil 4.15'de verilen şeklin verisi. Poligonların köşelerinin x,y ve z         %
koordinatları tanımlanmalıdır.
subplot(121)
x=[0 2 2 0;0 2 2 0]';
y=[0 0 0 0;1 1 1 1]';
z=[3 3 0 0;3 3 0 0]';
p=[1 1 1 1;2 2 2 2]';
patch(x,y,z,p)
box on;grid on;axis([-1 3 -1 2 -1 4])
% Şekil 4.14.b'da verilen şeklin verisi. Önceki poligonlara ek olarak yatay
% düzlemde yer alan 3 yeni poligon eklenmiştir.
subplot(122)
x=[0 2 2 0;0 2 2 0;0 2 2 0;0 2 2 0;0 2 2 0]';
y=[0 0 0 0;1 1 1 1;-.5  -.5 1.5 1.5;  0 0 1 1;0 0 1 1]';
z=[3 3 0 0;3 3 0 0;1 1 1 1;3 3 3 3;0 0 0 0]';
p=[1 1 1 1;2 2 2 2;3 3 3 3;4 4 4 4;5 5 5 5]';
patch(x,y,z,p)
box on;grid on;axis([-1 3 -1 2 -1 4])
```

Şekil 4.15'de verilen poligonların x,y,z koordinatları ile her bir poligona atanan değerler aşağıdaki gibidir:

```
x =
     0     0     0     0     0
     2     2     2     2     2
     2     2     2     2     2
     0     0     0     0     0


y =
     0    1.0   -0.5      0      0
     0    1.0   -0.5      0      0
     0    1.0    1.5    1.0    1.0
     0    1.0    1.5    1.0    1.0


z =
     3     3     1     3     0
     3     3     1     3     0
     0     0     1     3     0
     0     0     1     3     0
```

```
p =

     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
```





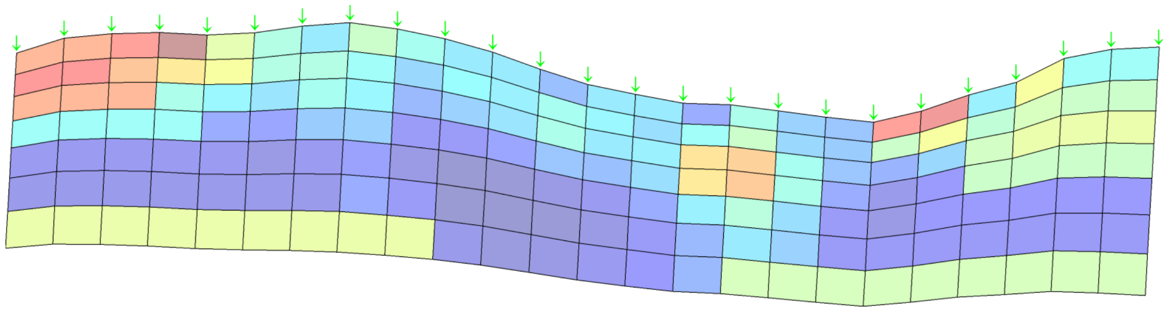
Şekil 4. 15 patch fonksiyonu ile 3B uzayda yer alan poligonlar ve bunlara atanan fiziksel parametrelerin görüntülenmesi

Jeofizik Modellerin Görselleştirilmesi

MATLAB'da hazır bulunan `patch` fonksiyonu kullanılarak bir jeofizik modelin 2B ve 3B çizimleri yukarıda anlatılan çerçevede yapılabilir. Bu konudaki örnekler jeofizik yöntemlerden doğru akım özdirenç ile elde edilen modellerden verilecektir. Ancak aynı uygulamalar diğer jeofizik model ve verilerin çiziminde de benzer şekilde kullanılabilir.

Elektrik yöntemlerin iki boyutlu uygulamalarında yöntemin duyarlı olduğu yer içi bölümü sonlu sayıda hücreye bölünerek ayrıklaştırılır. Yöntemden elde edilen veriler ile bu hücrelere ait fiziksel parametrenin değeri çözülür. Sonuçların görselleştirilmesi ise modelin geometrisi ve her bir hücre için hesaplanan özdirenç değeri kullanılarak gerçekleştirilir (Şekil 4.16). Model hücreleri çalışma alanındaki topoğrafik değişimlere bağlı olarak dikdörtgen ya da dörtgen olabilir. Verilerin değerlendirilmesinde kullanılan yöntem, yazılım, ayrıklaştırma türü gibi parametrelere bağlı olarak model geometrisi ve özdirençler farklı biçimlerde karşımıza çıkabilir. Burada kullanılan yöntem iyi anlaşılmalı ve verilen bilgilerden hücrelerin konumları ve onlara atanan özdirenç parametresi yukarıdaki örneklere benzer biçime getirilebilmelidir.

Şekil 4.16'de örnek olarak verilen 2B özdirenç modeli, x-z düzleminde bir fiziksel parametrenin değişimini göstermektedir. Bu nedenle daha önce söz edilen `contourf`, `imagesc`, `pcolor` gibi MATLAB fonksiyonları da çizim için kullanılabilir. Bu fonksiyonların kullanımı görünüm açısından kullanıcı tercihi olabilir. Ancak topoğrafik bilgilerin de modele katılması gibi durumlar da düşünüldüğünde genel tüm amaçlar için `patch` fonksiyonu kullanılması düşünülebilir. Şekil 4.16'de verilen örnek çizimde bu tür bir model görülebilir.

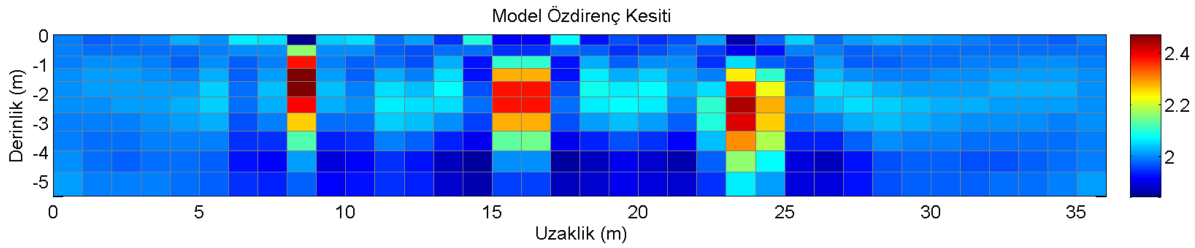


Şekil 4. 16 2B yer modellerinin görselleştirilmesi. Düşey eksen derinlik yatay eksen ise uzaklığı göstermektedir. Dörtgen hücrelere atanan renkler fiziksel parametrenin değeri ile ilişkili olup seçilen bir renk skalasına göre renklendirilmektedir.

Yukarıdaki örneğe benzer şekilde bir çizim yapılmak istenirse modeli oluşturacak hücrelerin dört köşe koordinatları ve her bir hücrenin dört köşesine atanacak fiziksel parametre değerleri gereklidir. Bir hücrenin içeriğini tek bir parametre ile tanımlamak istersek dört köşesine de aynı değeri atamak gerekir. Bu işlemleri gerçekleştirerek böyle bir kesit çizdirebilmek için gerekli bilgileri içeren veri dosyasına http://goo.gl/Ak4zop adresinden ulaşılabilir. Bu örnekte topoğrafik bilgi bulunmamaktadır. Bu nedenle yukarıdan aşağıya bir sütunda yer alan tüm hücrelerin köşelerinin x koordinatları aynı olacaktır. Çizimi yapmak için gerekli bilgiler bir *.mat dosyası içerisinde saklanmıştır. Aşağıda verilen programda bu dosyanın içeriği belleğe aktarılmakta, her bir hücrenin dört köşe koordinatı hesaplanmakta ve model patch fonksiyonu ile çizdirilmektedir. Gerekli açıklamalar program içerisinde verilmiştir.

```
% nx  : x yönünde nokta sayısı      nz : z yönünde nokta sayısı
%  z  : z yönünde derinlikler      rho : fiziksel parametre
%  x  : noktaların x konumları
% Yukarıdaki bilgiler res2d.mat dosyası içeriğinde bulunmaktadır.
% load deyimi ile aynı isimli değişkenler belleğe yüklenilebilir.
load res2d.mat
say=1;
for k=1:nx-1
    for m=1:nz-1
        xp(say,:)=[x(k) x(k+1) x(k+1) x(k)];
        zp(say,:)=[z(m) z(m) z(m+1) z(m+1)];
        say=say+1;
    end
end
patch(xp',zp',repmat(rho',4,1));
xlabel('Uzaklık (m)');
ylabel('Derinlik (m)');
title('Model Özdirenç Kesiti' );
axis image
```

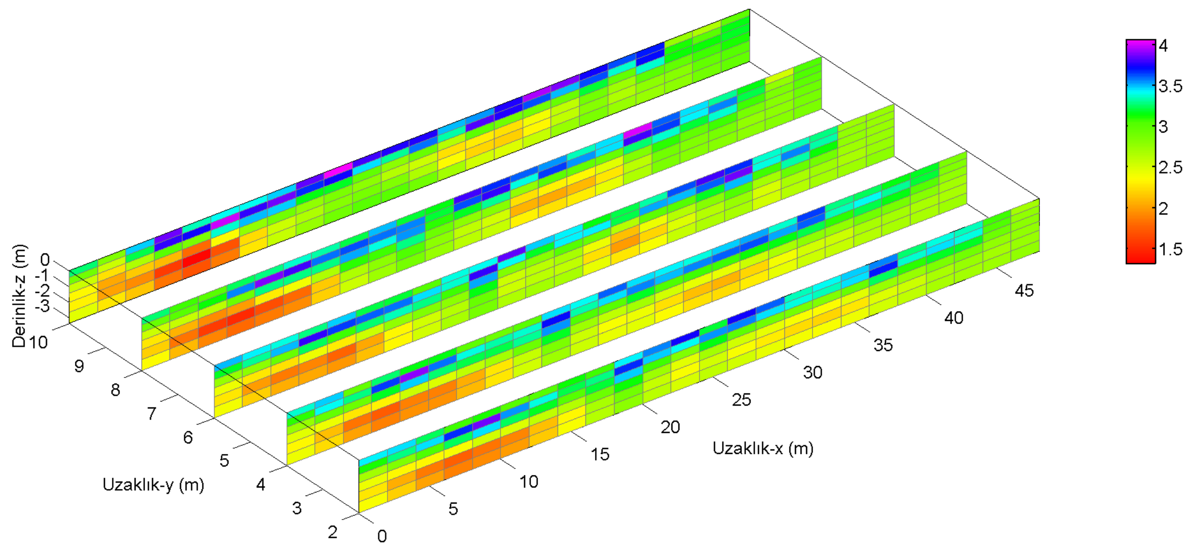
Program çalıştığında üretilen grafik Şekil 4.17'de verilmiştir. Modelde düşey yönde 10 yatay yönde 36 olmak üzere 360 adet hücre bulunmaktadır. Hücre numaraları (sıraları) üst sol köşeden başlayarak aşağıya doğru ilerlemektedir.



Şekil 4. 17 patch fonksiyonu ile çizdirilmiş 2B yer elektrik modeli

Örnek gösterim için 2B özdirenç modeli seçilmiş olmakla birlikte x-z veya x-y düzlemlerinde bir fiziksel parametrenin dağılımını göstermek üzere diğer jeofizik yöntemlere ait modellerin çizilmesinde de aynı program kullanılabilir.

Yalnızca koordinatlara üçüncü bir boyut eklenerek diğer yöndeki değişimler de aynı grafik üzerinde gösterilebilir. Şekil 4.15'de basit örneklerinin verildiği 3B kesitler de hazırlanabilir. Bu amaçla bir çalışma alanına ait birden çok 2B model x,y,z koordinatları dikkate alınarak Şekil 4.18'de gösterilmiştir. Burada derinlik (z), soldan sağa doğru olan eksen x-eksenini, diğer yatay eksen ise y-eksenini tanımlamaktadır.



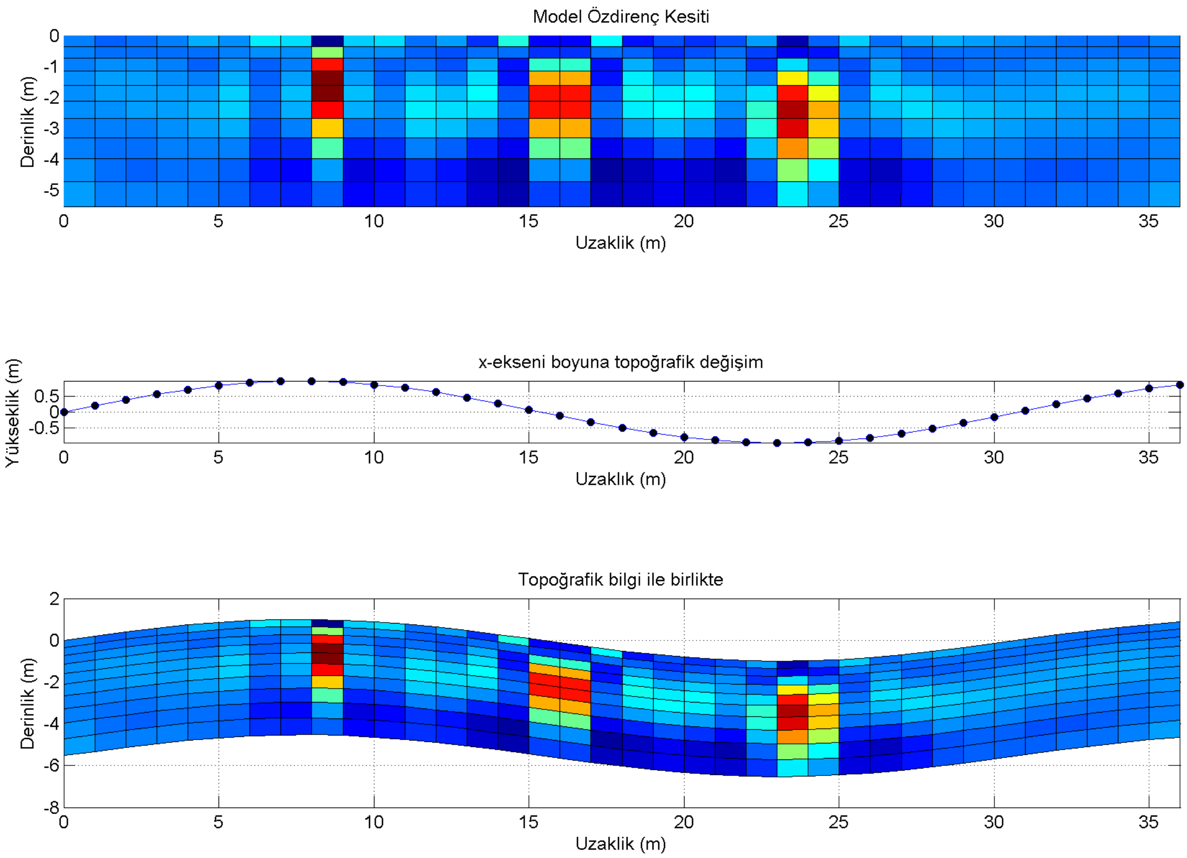
Şekil 4. 18 (x,y,z) koordinatları ile tanımlanmış hücreler kullanılarak patch fonksiyonu ile oluşturulan gösterim

Yukarıdaki çizimin hazırlanmasında kullanılan verilere http://goo.gl/edfTPU adresinden ulaşılabilir. Dosyalarda gerçek bir elektrik özdirenç çalışmasından elde edilen 2B modellere ait bilgiler bulunmaktadır (Akca Öztürk, 2011). Her bir tekil kesitin oluşturulmasında kullanılan hücrelerin dört köşe koordinatları ve bu hücrelere atanan özdirenç değerleri dosya içeriğinde bulunmaktadır. Aşağıdaki program a5-1ws1.mat, a5-1ws2.mat,.. şeklinde sırayla numaralanmış 5 dosyadan bu verileri okumaktadır. Kesitlerin üçüncü (y) eksendeki yerlerini konumlandırmak için üçüncü bir konum değişkeni tanımlanmakta ve tüm bilgiler kullanılarak şekil çizdirilmektedir. Gerekli açıklamalar aşağıdaki program dosyası içerisinde verilmiştir.

```
clear all
close all
clc
pname='a5-1ws';
% 2B modellerden 3B görüntü elde etme
XP=[];ZP=[];PRHO=[];YP=[];
for k=1:5
    %Sistematik numaralandırılmış mat-dosyaları okunuyor
    data=load([pname,num2str(k),'.mat']);
    %Herbir kesitte yer alan hücrelerin x ve z koordinatları
    %birleştiriliyor
    XP=[XP;data.xp];
    ZP=[ZP;data.zp];
    %Model özdirençleri tüm kesitler için birleştiriliyor
    PRHO=[PRHO;data.prho];
    %Üçüncü eksene ait koordinat değerleri oluşturuluyor.
    %Ardışık kesitlerin araları 2m olduğundan bir kesitteki tüm hücrelerin
    %y eksen değerleri aynıdır. Y yönünde ilerlendikçe sırasıyla 0,2,4,6,8
    %şeklinde değerler almaktadır.
    YP=[YP;repmat((k-1)*2,length(data.prho),4)];
end
%Tüm hücrelerin x,y,z koordinatları ve özdirençleri ayrı ayrı olmak üzere
%birer değişkende toplandı.
%Çizim yapılıyor:
hh=patch(XP',YP',ZP',repmat(log10(PRHO'),4,1));
%Dış çerçeve ekleniyor
box on
axis image
%Hücre çerçeve rengi griye ayarlanıyor [R G B] değerler sırası ile kırmızı,
%yeşil ve mavi renklerin ağırlığını belirliyor 0-1 aralığında giriliyor.
set(hh,'EdgeColor',[.5 .5 .5]);
set(gca,'DataAspectRatio',[1 .35 1])
view(3)
colorbar
xlabel('Uzaklık-x (m)');ylabel('Uzaklık-y (m)');zlabel('Derinlik-z (m)');
```

Jeofizik çalışmaların yapıldığı alanlar her zaman düz olmayabilir. Engebeli alanlarda modelin gerçekçi çizimi için topoğrafyadaki değişimlerin de model geometrisine eklenmesi gereklidir. Şekil 4.17'dr verilen 2B yer elektrik modeline varsayımsal bir topoğrafik bilgi eklenerek model geometrisinin buna göre düzenlenmesi aşağıdaki program aracılığı ile gerçekleştirilebilir. Programın üreteceği grafik Şekil 4.19'da alt sırada verilmiştir. Programı çalıştırmak için önceki uygulamadaki veri dosyasının kullanılması gerekecektir.

```
load res2d.mat
say=1;
% Topoğrafik bilgi varsayımsal olarak bir sinüs fonksiyonundan üretiliyor
y=sin(0.65*pi*x);
for k=1:nx-1
    for m=1:nz-1
        xp(say,:)=[x(k) x(k+1) x(k+1) x(k)];
        % z koordinatları her bir x konumunda tanımlı yüksekliğe göre kaydırılıyor
        zp(say,:)=[z(m)+y(k) z(m)+y(k+1) z(m+1)+y(k+1) z(m+1)+y(k)];
        say=say+1;
    end
end
patch(xp',zp',repmat(rho',4,1));
xlabel('Uzaklik (m)');
ylabel('Derinlik (m)');
title('Model Özdirenç Kesiti');
box on
grid on
```



Şekil 4. 19 2B jeofizik modele topoğrafya bilgisinin eklenmesi

slice

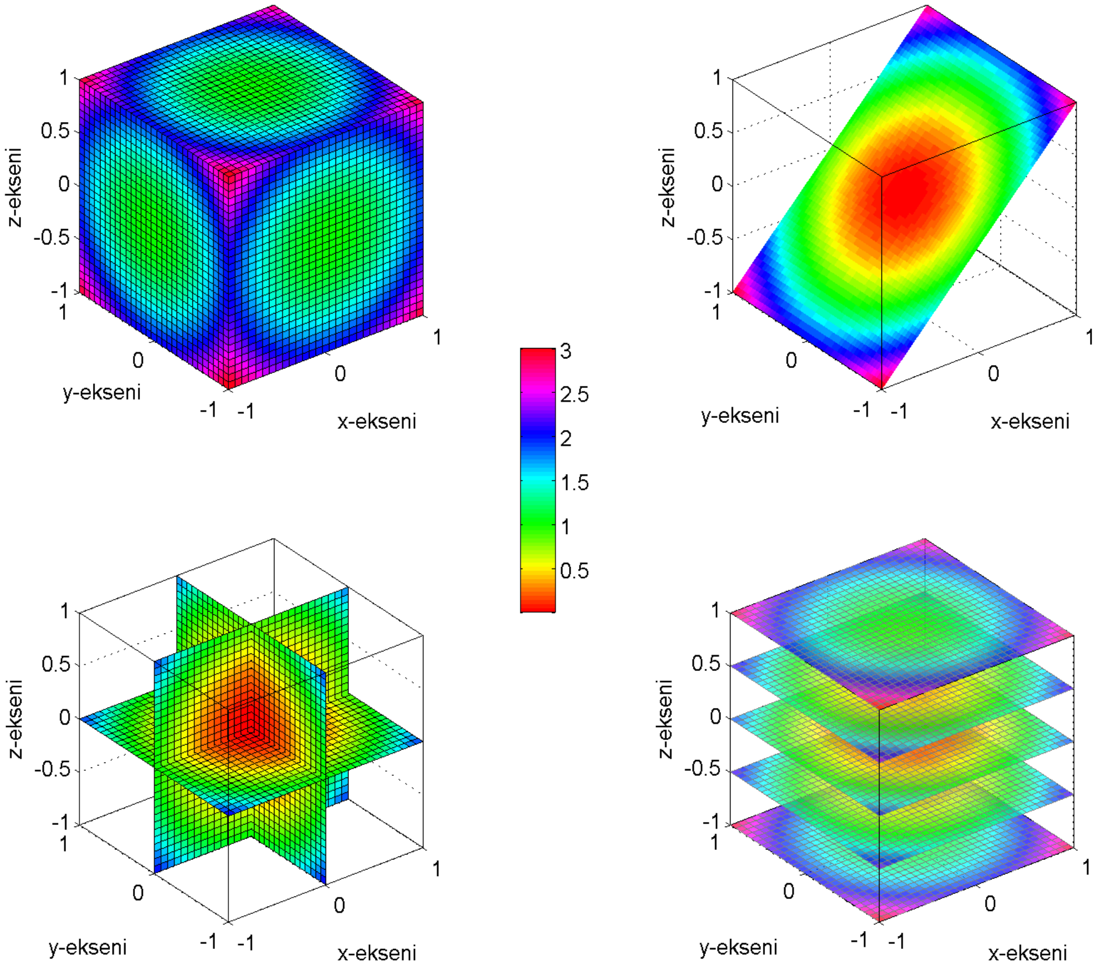
Kelime anlamı dilim olan `slice`, hacimsel bir veri kümesinin gösterimi ve bu hacmi kesen düzlemlerde kesitler alınabilmesini sağlayan bir MATLAB fonksiyonudur. Bir hacmin tanımlanabilmesi için üç boyut gerektiği açıktır. Fonksiyonun genel kullanımı

```
slice(x,y,z,v,sx,sy,sz)
```

şeklindedir. Burada x,y ve z değişkenlerinde 3B uzayda veri değeri atanmış her bir noktanın koordinatları yer almaktadır. Bu değişkenler üç boyutlu olmalıdır. Belirlenmiş koordinatlarda çizdirilecek fiziksel büyüklüğün değerleri ise v değişkeninde saklanmaktadır. v değişkeni de üç boyutlu olup x,y ve z ile aynı büyüklükte olmalıdır. Yazılışta yer alan sx,sy ve sz değişkenleri ise her üç eksende kesit alınacak düzlem ya da düzlemleri göstermektedir. Aynı eksende birden çok düzlemde kesit alınabilir. Aşağıdaki örnekte x-,y- ve z-eksenleri [-1,1] aralığında 30 eşit parçaya bölünerek örneklenmiş ve her bir x,y,z konumuna deneme amaçlı bir değer atanmıştır. Bu değer tüm koordinatların karelerinin toplamı şeklindedir. Çizimde kullanılacak gerçek bir verinin verilen x-y-z gridi için tanımlanmış olması ya da yeniden gridlenmesi gerekeceği hatırlanmalıdır. Gerekli diğer açıklamalar program içerisinde verilmiştir.

```
function dilimle
% 3B varsayımsal veri üret
[x,y,z] = meshgrid(linspace(-1,1,30));
v = x.^2 + y.^2 + z.^2;

% Dilimleri çiz
subplot(2,2,1)
slice_ciz(x,y,z,v,[-1 1],[-1 1],[-1 1])

% eğik bir düzlem tanımla
[xi, yi] = meshgrid(linspace(-1,1,50));zi = xi;
subplot(2,2,2)
slice_ciz(x,y,z,v,xi,yi,zi);

% x=0,y=0 ve z=0 düzlemlerinden dilimler
subplot(2,2,3)
slice_ciz(x,y,z,v,0,0,0)

% z-yönünde 0.5 birim aralık ile tekrarlı dilimler
subplot(2,2,4)
slice_ciz(x,y,z,v,[],[],-1:.5:1);

function slice_ciz(x,y,z,v,xk,yk,zk)
slice(x,y,z,v,xk,yk,zk);
axis image;box on;xlabel('x-ekseni');ylabel('y-ekseni');zlabel('z-ekseni');
colormap(hsv)
```

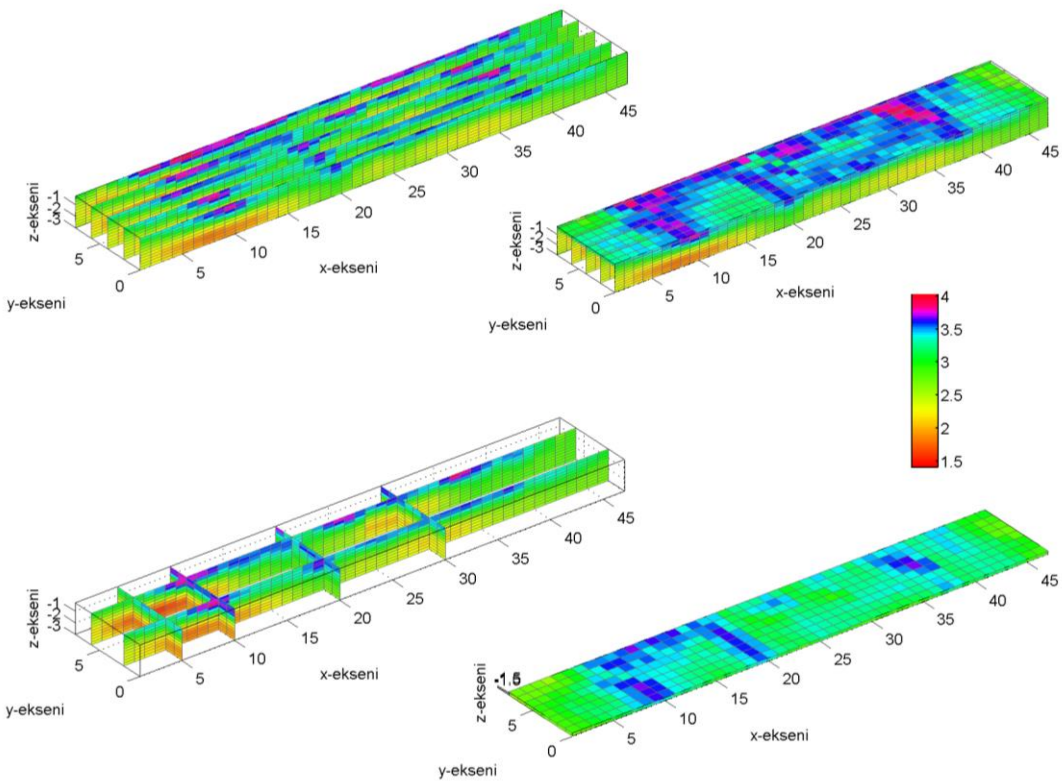
Yukarıda verilen program çalıştırıldığında çeşitli olasılıklar gözetilerek bir hacimde tanımlanmış veriden kesitler alınarak aşağıdaki grafik üretilir.



Şekil 4. 20 slice fonksiyonu ile hacimsel bir veriden alınan düzlemsel kesitler

Burada verilen uygulama örneğinden yola çıkarak Şekil 4.18'de patch fonksiyonu ile görselleştirilen veriler için benzer çizimler yapılabilir. Çizim sırasında kullanılacak verilerin yeniden düzenlemesi gerekecektir. Bahsedilen örnekte koordinatlar her bir hücrenin dört köşesi için tanımlanmışken burada her bir veri için birer adet x,y,z koordinatı gerekmektedir. Hücreler için tanımlanmış fiziksel parametrenin her bir hücrenin merkezine ait olduğu kabul edilerek bu tür bir dönüşüm yapılabilir. Bu durumda hücrelerin dört köşe koordinatlarından merkezlerine ait koordinatların hesaplanması yeterli olacaktır. Diğer taraftan her bir kesiti bir düzlemde tanımlı olan verilerin de birleştirilerek 3B hacimde tanımlı hale getirilmesi ve gereksinimlere göre daha sık gridlenmesi gerekecektir. Aşağıdaki program özetlenen işlemleri yürüterek verileri `slice` fonksiyonu ile çizilebilecek şekilde düzenlemektedir. Bu işlemin sağlayacağı yarar gerçekte veri olmayan düzlemlerde fiziksel parametrenin dağılımı ile ilgili bilgi sağlaması olacaktır. Örnekte Şekil 4.18'nin üretilmesinde kullanılan veriler kullanılacaktır. Bu amaçla yazılan program aşağıda verilmiştir. Programın çıktısı Şekil 4.21'de gösterilmiştir.

```
function patch2slice
clear all;close all;clc
X=[];Y=[];Z=[];P=[];
prf_ara=2; % Profiller arasındaki uzaklık
dx=1; dy=1; dz=0.25; %x, y ve z yönlerinde gridleme aralığı

%Dosyalar okunup koordinat ve veriler birleştiriliyor
for k=1:5
    load(['a5-1ws',num2str(k),'.mat'])
    %hücrelerin merkez koordinatları
    x=(xp(:,1)+xp(:,2))/2;             X=[X;x];
    z=(zp(:,1)+zp(:,3))/2;             Z=[Z;z];
    %y yönünde profil aralığı kadar uzaklık
    y=(k-1)*prf_ara*ones(size(x));     Y=[Y;y];
    %fiziksel parametreler
    P=[P;prho];
end
% Verilerin yeniden gridleneceği ızgara oluşturuluyor
xgrid=min(X):dx:max(X);
ygrid=min(Y):dy:max(Y);
zgrid=min(Z):dz:max(Z);
[XG,YG,ZG]=meshgrid(xgrid,ygrid,zgrid);
% Fiziksel parametrenin değerleri yeniden gridleniyor
PG=griddata(X,Y,Z,P,XG,YG,ZG);

% Y-yönünde kesitler alınıyor
subplot(2,2,1)
slice_ciz(XG,YG,ZG,log10(PG),[],[0:2:8],[])
% Y- ve z- yönünde kesitler alınıyor
subplot(2,2,2)
slice_ciz(XG,YG,ZG,log10(PG),[],[0:2:8],[-.5])
% x ve y-yönünde kesitler alınıyor
subplot(2,2,3)
slice_ciz(XG,YG,ZG,log10(PG),[5 10 20 30],[2 6],[])

% z-yönünde kesitler alınıyor
subplot(2,2,4)
slice_ciz(XG,YG,ZG,log10(PG),[],[],[-1 -1.25])


function slice_ciz(x,y,z,v,xk,yk,zk)
h=slice(x,y,z,v,xk,yk,zk);
axis image;box on;xlabel('x-ekseni');ylabel('y-ekseni');zlabel('z-ekseni');
colormap(hsv)
shading faceted
set(h,'EdgeColor',[.25 .25 .25]+.2)
```



Şekil 4. 21 slice fonksiyonu ile üretilen ve çeşitli düzlemlerde alınan jeofizik kesitlerin görüntüsü

isosurface

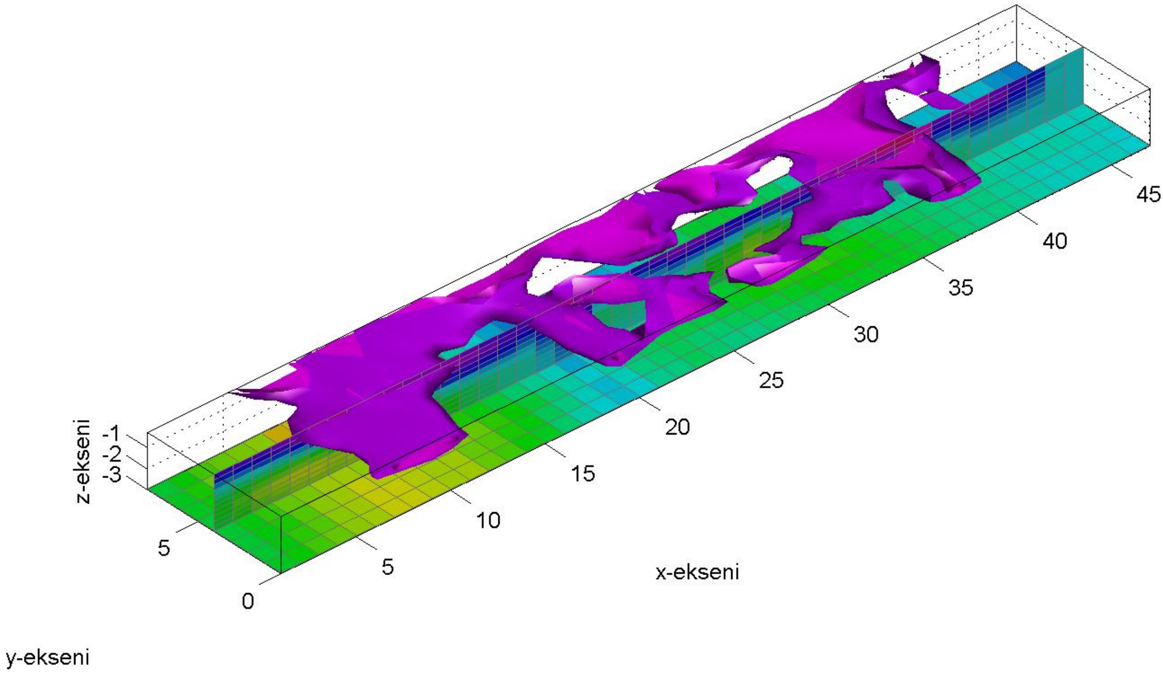
Üç boyutlu çizim oluşturulmasında kullanılan diğer birçok MATLAB grafik fonksiyonu arasından son olarak `isosurface` fonksiyonuna değinilecektir. Bu fonksiyon ile yine bir hacim içerisinde tanımlanmış veriler için eş-değer yüzeyleri oluşturulmaktadır. Bunun anlamı çizilecek yüzey üzerindeki tüm noktalarda fiziksel parametrenin değerinin aynı olmasıdır. Kullanımı `slice` fonksiyonu ile büyük benzerlik göstermekte olup aşağıdaki gibidir:

```
isosurface(x,y,z,v,sp)
```

Burada x,y ve z bir hacim içerisinde verilerin ait olduğu koordinatları, v bu koordinatlarda tanımlı veriyi, sp ise eş-değer yüzeyin oluşturulmasında kullanılacak parametre değeridir. Önceki örnekteki veri kullanılarak değeri 3.6 olan eş yüzey aşağıdaki gibi oluşturulabilir.

```
slice_ciz(XG,YG,ZG,log10(PG),[],[4],[-3])
hold on
isosurface(XG,YG,ZG,log10(PG),3.6)
camlight
lighting gouraud
axis image;box on
```

verilen yazılış ile aşağıda verilen şekil üretilecektir.



Şekil 4. 22 Eş değer kontur yüzeyleri oluşturan isosurface fonksiyonunun önceki örnek için uygulanması

Alıştırmalar

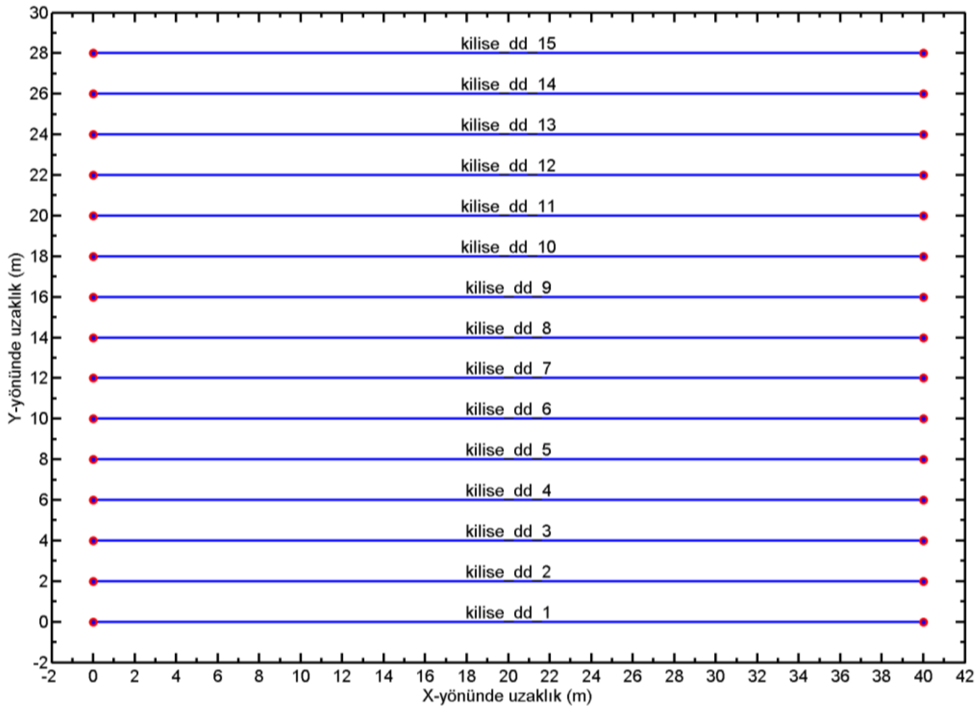
1. Aşağıda katman parametreleri verilen 1B sismik hız modelini, hızın derinlikle değişimini gösterecek şekilde çizdiriniz.

Katman	Vp (km/s)	Kalınlık(m)
1	1.5	2
2	2.0	5
3	1.7	4
4	35	8
5	5	∞

2. http://goo.gl/JKvyIT adresinden indireceğiniz dosya içeriğinde bir fiziksel parametrenin x ve z koordinatlarına bağlı olarak değişimleri verilmiştir. Birinci sütunda verilerin x-koordinatları, ikinci sütunda z-koordinatları ve üçüncü sütunda bu koordinat çiftinde ölçülen değerler verilmiştir. Verileri okuyarak x yönünde 0.5m z yönünde 0.5m aralıklarla yeniden gridleyip çizdiriniz. Çizimde hesapladığınız değerlerin 10 tabanından logaritmasını kullananınız.

3. http://goo.gl/k4A1iq adresinde bir çalışma alanında birbirine paralel ve 2m aralıklarla yer alan profiller üzerinde ölçülen görünür özdirenç verileri bulunmaktadır. On beş veri dosyası sistematik olarak isimlendirilmiştir. Verilerin ölçüldüğü hatların konumları aşağıdaki şekilde

verilmiştir. Her bir veri dosyasında şekilde gösterilen hatlar üzerinde görünür özdirencin düşey (z) yönündeki değişimine dair bilgi bulunmaktadır.

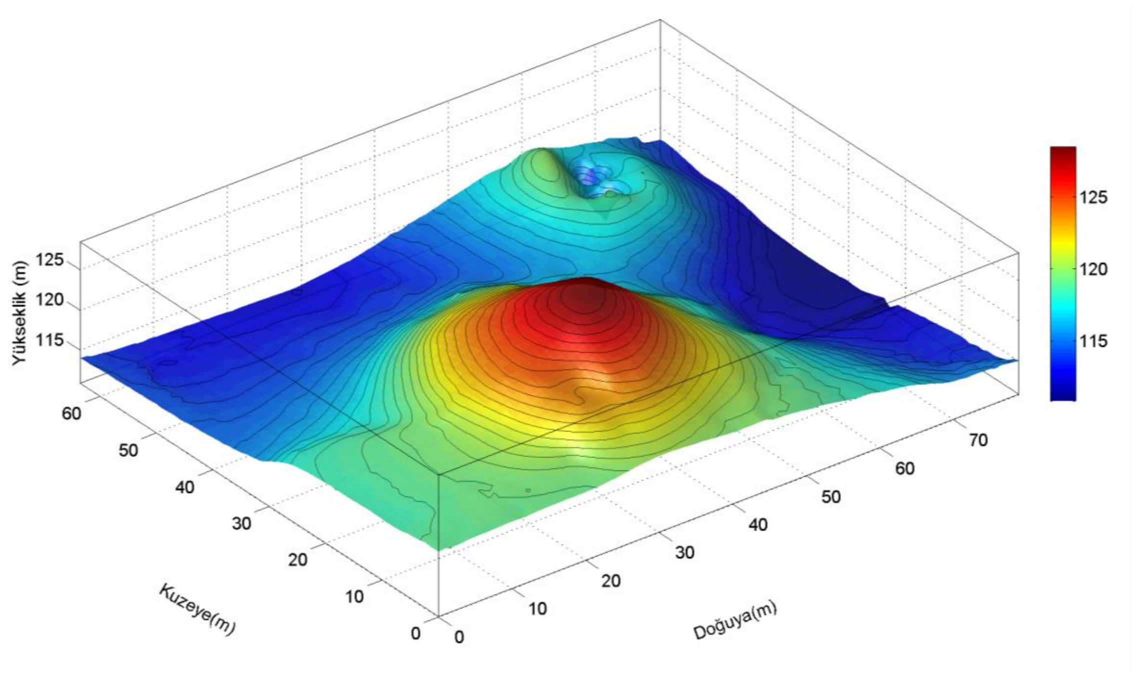


Dosya içeriğindeki verilerin açıklaması aşağıdaki gibidir:

```
kilise_dd_3               % Ölçüm hattı adı
2.0                       % x-yönünde ölçüm aralığı
3                         % Ölçümlerle ilgili bir parametre
150                       % Veri sayısı
1                         % Ölçümlerle ilgili bir parametre
0                         % Ölçümlerle ilgili bir parametre
3.00   2.00 1 111.96 % Veri x-konumu | mn | ölçüm seviyesi | görünür özdirenç
4.00   2.00 2 101.42
5.00   2.00 3 101.98
6.00   2.00 4 102.24
7.00   2.00 5 100.73
8.00   2.00 6 105.40
.
.
.
```

Buna göre tüm dosyaları okuyarak her bir ölçüm seviyesindeki özdirenç dağılımını gösteren haritaları oluşturunuz. Böyle bir haritanın tüm dosyalardan aynı seviyeye ait bilgilerin derlenmesi ile yapılabileceğini göz önünde bulundurun.

4. http://goo.gl/ZrOvM9 adresinden indireceğiniz dosya içerisinde bir çalışma alanında enlem ve boylama bağlı olarak ölçülen yükseklik değerleri bulunmaktadır. Dosya içeriğindeki bilgiler sütunlar halinde ve x,y,z sırasında verilmiştir. Buna göre verilen topoğrafik bilgileri uygun MATLAB fonksiyonlarını kullanarak çizdiriniz. surf ve contour3 fonksiyonları kullanılarak çizilen bir örnek aşağıda verilmiştir.



4.3 Hızlı Fourier Dönüşümü (FFT)

Fourier dönüşümü jeofizik verilerin ölçülmesi, değerlendirilmesi ve işlenmesi sırasında sıkça başvurulan bir spektral analiz işlemidir. Zaman bölgesindeki olayların frekans içeriklerinin görüntülenebilmesi için kullanılır. İşlem geri dönüşlüdür. Yani frekans bölgesinden zaman bölgesine geçiş de olanaklıdır. Bu işleme ise ters Fourier dönüşümü adı verilir. Konunun ayrıntısına girilmeden önce örnekler verilecek ardından da uygulaması verilecektir. `fft` fonksiyonunun genel kullanımı

```
y=fft(x,n)
```

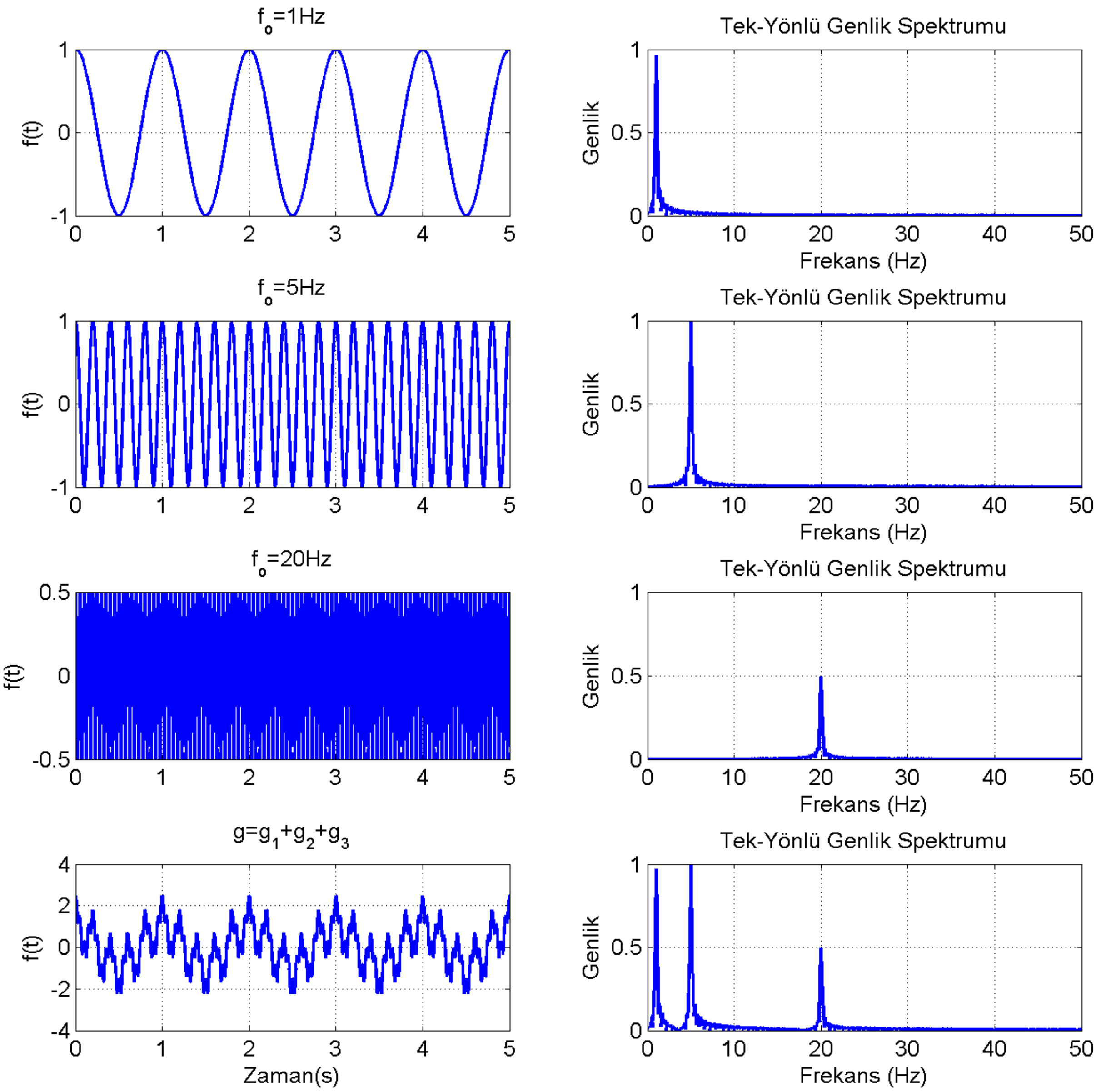
şeklinde verilebilir. Burada x, Fourier dönüşümü alınacak yöneyi, n ise hesaplamanın kaç ayrık noktada yapılacağını göstermektedir. Hızlı Fourier dönüşümünün algoritması giriş verisindeki örnek sayısının 2'nin tam kuvvetleri olmasını gerektirmektedir. Bu nedenle giriş verisinin boyu n'den küçükse bir sonraki tam kuvvete kadar sıfırlarla doldurulabilir. Tersi olarak giriş verisinin boyu n'den büyükse bu kez n'inci elemandan sonrası işleme katılmaz.

Sinüzoidaller ve onların toplamları Fourier dönüşümü başta olmak üzere spektral analiz uygulamalarının daha iyi anlaşılabilmesi için sık kullanılan örneklerdir. Farklı frekanslarda ve genliklerde giriş sinyali üretmek kolaydır. Aşağıdaki örnekte sırasıyla 1, 5 ve 20Hz frekanslarında üç kosinüs fonksiyonu üretilmekte ve dördüncü bir fonksiyon bunların toplamlarından oluşmaktadır. Bu dört fonksiyon dt=0.001s zaman aralığı ile örneklenmektedir. Bu dört zaman serisinin Fourier dönüşümleri MATLAB kütüphanesinden çağırılan `fft` fonksiyonu ile hesaplanmıştır. Fonksiyonların sayısal değerlerini ve bunların Fourier dönüşümleri hesaplayan, zaman ve frekans bölgesindeki davranışlarını çizen MATLAB programı aşağıda verilmiştir. Gerekli açıklamalar program içerisinde bulunabilir.

```
function fft_1
%Farklı frekansta üç cos fonksiyonu ve onların toplamının Fourier
%dönüşümleri
dt = 0.001;                         % Zamanda örnekleme aralığı
FS = 1/(2*dt);                      % Nyquist frekansı
t  = 0:dt:5;                        % Zaman değerleri
f1 = 1.0;                           % Birinci sinüzoidal frekansı
f2 = 5.0;                           % İkinci sinüzoidal frekansı
f3 = 20;                            % Üçüncü sinüzoidal frekansı
% Sınama verileri oluşturuluyor
g1 = cos(2*pi*f1*t);
g2 = cos(2*pi*f2*t);
g3 = 0.5*cos(2*pi*f3*t);
g  = g1+g2+g3;

fft_ornek(t,g1,FS,1)
fft_ornek(t,g2,FS,2)
fft_ornek(t,g3,FS,3)
fft_ornek(t,g ,FS,4)

function fft_ornek(t,g,FS,ax)
cizim_yeri=(2*ax)-1;
L = length(t);                    % Sinyal Boyu
% Zaman fonksiyonunu çiziliyor
subplot(4,2,cizim_yeri)
plot(t,g)
grid on
xlabel('Zaman(s)');ylabel('f(t)');title('Zaman serisi')
% hızlı Fourier dönüşümü
NFFT = 2^nextpow2(L);             % 2 nin sonraki tam kuvveti
Y = fft(g,NFFT)/L;
% Tek taraflı gösterime göre çiz
subplot(4,2,cizim_yeri+1)
f = FS*linspace(0,1,NFFT/2+1);
plot(f,2*abs(Y(1:NFFT/2+1)))
axis([0 50 0 1])
grid on
title('Tek-Yönlü Genlik Spektrumu')
xlabel('Frekans (Hz)');ylabel('Genlik')
```

Aşağıdaki örnekte giriş verisinin 5001 elemanı bulunmaktadır. Bu durumda n ya 4096 olarak belirlenip 5001'e kadar olan elemanlar kullanılmayacak veya n 8192 olarak belirlenip 5002-8192 aralığındaki elemanlara 0 değeri atanacaktır. Örnekte veri `n` boyundan büyük 2 nin ilk tam katı alınmış ve bu sayıyı bulmak için `nextpow2` fonksiyonu kullanılmıştır. Zaman bölgesindeki bir verinin frekans bölgesindeki karşılığı karmaşık sayılardan oluşmaktadır. Dolayısı ile gerçel ve sanal kısımları olan bir karmaşık sayının genliği ve fazından söz edilebilir. Zaman bölgesindeki bir verinin frekans bölgesindeki karşılığı genliği ile gösterilir. Karmaşık bir sayının genliği MATLAB'da `abs` fonksiyonu ile alınabilir. Bu fonksiyon gerçel sayıların mutlak değerini verirken, karmaşık sayıların genliğini verir. Karmaşık sayıların genliği, gerçel ve sanal kısımlarının karelerinin toplamının kareköküdür.

Diğer taraftan frekans bölgesinde tek taraflı ve çift taraflı gösterim seçenekleri bulunmaktadır. Tek taraflı gösterimde frekans ekseni $[0 \;\; 2f_n]$ aralığında iken çift taraflı gösterimde bu aralık `[-fn fn]` şeklindedir. Çift taraflı gösterim bir simetri gösterir ve genlikler 0 frekansının sağ ve soluna paylaştırıldığından ikiye bölünmektedir. MATLAB `fft` fonksiyonu varsayılan olarak çift taraflı gösterime göre çıktı üretmektedir. Örnekte tercih edilen tek taraflı gösterim için fonksiyon çıktısının birinciden ortadaki elemanına kadar alınmakta ve genliği de 2 ile çarpılmaktadır. Verilen örnekteki fonksiyonlar zaman ve frekans bölgesinde Şekil 4.23'de gösterilmiştir.



Şekil 4. 23 Sol sütunda yukarıdan aşağıya g1, g2, g3 ve bunların toplamından oluşan g fonksiyonu zaman bölgesinde, sağ sütunda ise frekans bölgesindeki karşılıkları gösterilmektedir

Frekans bölgesinden zaman bölgesine geçiş için ise ters Fourirer dönüşümü kullanılmaktadır. Bu işlem MATLAB kütüphanesindeki `ifft` fonksiyonu ile gerçekleştirilebilmektedir. Ancak tek-çift taraflı gösterim ile ilgili düzenlemeden dolayı genliklerde bir düzenlemeye gidilmelidir. Yukarıdaki örnekte g olarak isimlendirilen fonksiyonun Fourier dönüşümü Y değişkeninde saklandığı düşünülürse ters Fourier dönüşümü

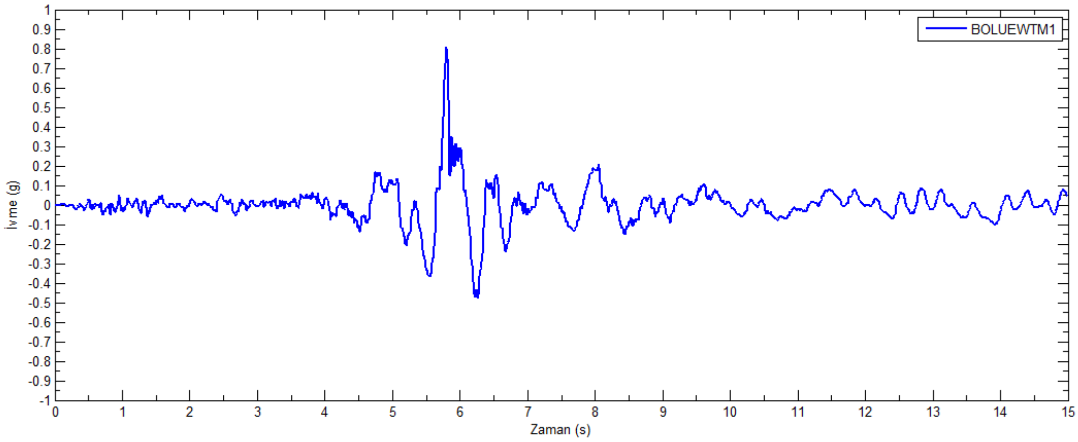
```
T=NFFT*ifft(Y)/2;
```

şeklinde alınabilir.

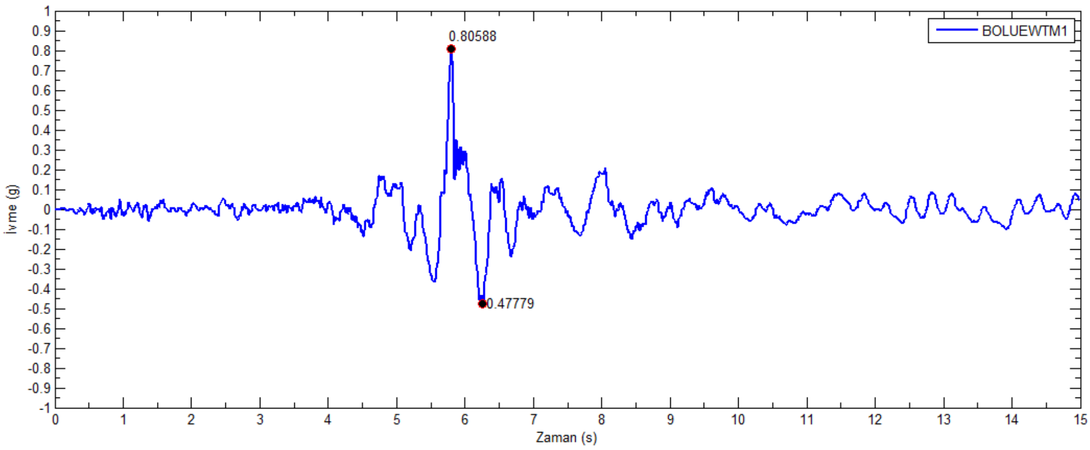
Alıştırmalar

1. Bir deprem gözlem istasyonunda kaydedilmiş depremin doğu batı yönündeki ivme verisini (Başokur vd., 2003) http://goo.gl/iVKG4p adresinden indiriniz. Dosya içeriğinde birinci sütunda saniye cinsinden zaman değerleri, ikinci sütunda ise bu zaman değerlerine karşılık gelen yer ivme değerleri verilmiştir.

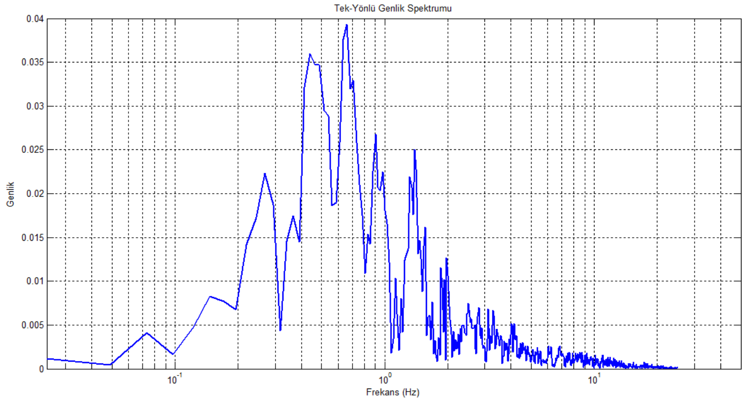
a. Veriyi okutarak aşağıdaki biçimde çizdiriniz



b. İvmenin (+) ve (-) yönde en yüksek değerlerini bularak grafik üzerine aşağıdaki gibi işaretleyiniz



c. İvme verisinin frekans içeriğini görmek üzere verinin Fourier dönüşümünü fft fonksiyonu ile hesaplayıp genliğini aşağıdaki şekilde çizdiriniz. Çizimde tek yönlü gösterim kullanılacak ve yatay eksen logaritmik seçilecektir.



2. $f(t) = 2\sin(\pi \mathrm{t}) + 0.5\sin(10\pi \mathrm{t}) + \sin(20\pi t)$ şeklinde tanımlanan fonksiyonu [0 10] saniye zaman aralığında hesaplayınız. Örnekleme aralığını fonksiyonların frekanslarını göz önüne alarak seçiniz. Hesapladığınız zaman serisinin Fourier dönüşümünü alınız. Fonksiyonun zaman ve frekanstaki davranışlarını yan yana çizdiriniz.

4.4 Sayısal Evrişim

Aynı örnekleme aralığı ile örneklenmiş b ve g gibi iki veri kümesinin sayısal evrişimi

$$f(i) = \sum_{j=1}^{m} \mathrm{b}(j)\,\mathrm{g}(i - j + 1)$$

bağıntısı ile hesaplanabilir. b ve g dizeylerinin eleman sayıları sırası ile m ve n olmak üzere bu iki yöneyin evrişiminden k=n+m-1 adet çıkış elde edilir. Bu da yukarıdaki eşitliğin sağ yanının, i indeksinin her değeri için bir kez yürütülmesi gerektiğini ortaya koyar. Çıkışın ilk yatay eksen değeri $tf_1 = tb_1 + tg_1$ bağıntısı ile hesaplanabilir. Sonraki yatay eksen değerleri bundan örnekleme aralığı (dt) kadar aralıklarla sıralanacaktır. Yukarıdaki işlemi yürüten MATLAB fonksiyonu

```
function C=evrisim(g,b)
n=length(g);
m=length(b);
k=n+m-1;
tic
for i=1:k
    top=0;
    for j=1:m
        nn=i-j+1;
        if nn>1&&nn<n
            top=top+b(j)*g(nn);
        end
    end
    C(i)=top;
end
```

şeklinde yazılabilir. Evrişim işlemi için yazılan fonksiyon ile aynı işlevi yerine getiren MATLAB kütüphane fonksiyonu `conv` da kullanılabilir. Fonksiyonun kullanımı

```
f=conv(b,g);
```

şeklindedir. Evrişimde fonksiyonların sırasının değişmesinin sonucu değiştirmeyeceği hatırlanmalıdır. Evrişim birçok sayısal yöntem içerisinde kullanılan bir işlemdir. Bunlardan biri de doğrusal süzgeçlerdir. Belirli frekansların veriden uzaklaştırılması, giriş verisinin türevinin ya da integralinin alınması gibi işlemler bu amaçlar için tasarlanmış süzgeç katsayıları ile bir giriş fonksiyonun evrişimi ile gerçekleştirilebilir. Süzgeçlerin kurulması, katsayılarının hesaplanması ve kullanımı ile ilgili geniş bilgi için Başokur (2007)'ye bakılabilir.

Aşağıda doğrusal bir süzgecin tasarlanarak bir giriş verisi ile evriştirilmesi ve bunun sonucunda yüksek frekansların veriden uzaklaştırılması ile ilgili bir örnek verilmiştir. Alçak geçişli bir süzgeç veriden belirli bir frekansın üstündeki bileşenlerin uzaklaştırılmasını sağlar. f_L gibi bir kesme frekansının altını geçiren ve bundan büyük tüm frekanslardaki olayları veriden uzaklaştıran doğrusal süzgecin katsayıları

$$b_i = \frac{\sin(2\pi f_L t_i)}{2\pi f_N t_i}, \quad i = 1, N$$

bağıntısı ile hesaplanabilir. Burada f_N Nyquist frekansı, t_i katsayıların hesaplandığı ayrık yatay eksen değerleri, N ise katsayıların sayısıdır. Süzgeç katsayılarının giriş verisi ile aynı örnekleme aralığı için hesaplanması gereklidir. Süzgeç katsayıları ile giriş verisinin sayısal evrişimi sonucunda süzgeçleme gerçekleşmiş olur. Bu işlem sembolik olarak

$$f_F = f * b$$

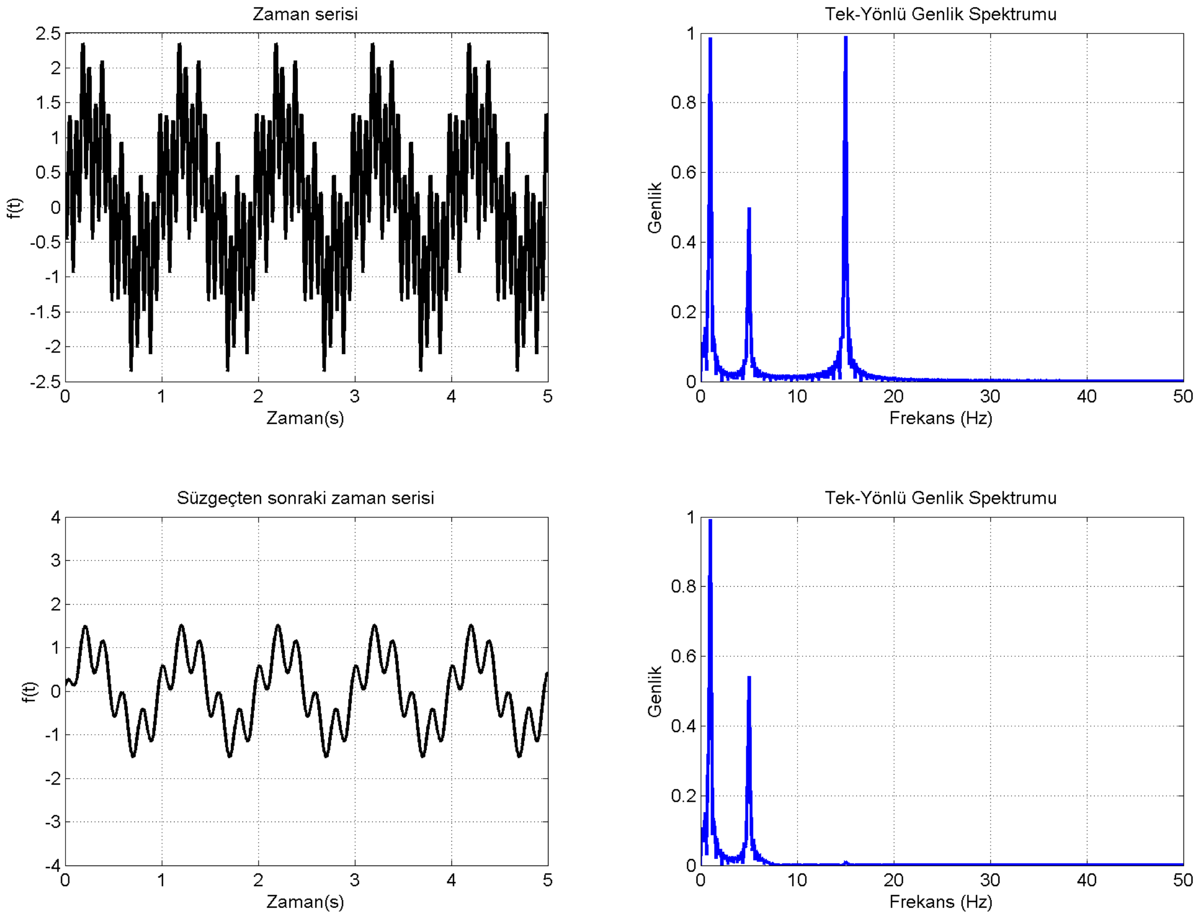
şeklinde gösterilebilir. Aşağıdaki ifade ile tanımlanan 1Hz, 5Hz ve 15Hz frekanslarında titreşen sinüzoidallerin toplamından oluşan

$$f(t) = \sin(2\pi t) + 0.5\cos(10\pi t) - \sin(30\pi t)$$

fonksiyonu dx=0.01 örnekleme aralığı ile örneklenerek sınama verisi olarak kullanılacaktır. Bu veriden 7Hz ve daha yüksek frekanstaki olayların uzaklaştırılması için bu amaçla hesaplanmış süzgeç katsayıları ile evriştirilmesi yeterli olacaktır. Süzgeç katsayılarının hesaplanmasında kullanılan bağıntıda f_L yerine 7, f_N yerine de 50 yazılabilir. Aşağıda verilen MATLAB programı sınama verisinin oluşturulması, süzgeç katsayılarının hesaplanması ve giriş verisi ile süzgeç katsayılarının evriştirilmesi ile zaman bölgesinde süzgeçleme işlemini gerçekleştirmektedir. Verinin frekans içeriğindeki değişimi göstermek için Fourier dönüşümü de alınmış ve süzgeçlemeden önce ve sonraki durumlar ortaya konulmuştur.

```
function evrisim
% Zamanda sınama verisi üretilip frekans bölgesi karşılığı ile çiziliyor
dt = 0.001;                                      % Örnekleme Aralığı zaman
t=0:dt:5;
%Nyquist frekansı
fn=1/2/dt;
L = length(t);                                   % Sinyal Boyu
ft=sin(2*pi*t)+0.5*cos(10*pi*t)-sin(30*pi*t);    % Sınama verisi üretildi
% Zaman fonksiyonunu çiziliyor
subplot(221)
zamanda_ciz(t,ft)
title('Zaman serisi')

subplot(222)
fft_hesapla_ciz(t,ft,L,fn)
% Süzgeç katsayıları hesaplanıyor
fL=7;
ts=-.3:dt:.3;
for k=1:length(ts)
    if ts(k)~=0
        b(k)=sin(2*pi*fL*ts(k))./2/pi/fn./ts(k);
    else
        % Paydanın sıfır olması durumunda
        b(k)=fL/fn;
    end
end
%Süzgeç katsayıları ile zaman fonksiyonu evriştiriliyor
fF=conv(ft,b);

% Çıkış sayısı
k=L+length(ts)-1;

% Çıkışın yatay eksen değerleri
tF(1)=ts(1)+t(1);
for m=2:k
    tF(m)=tF(m-1)+dt;
end
%Sonuçlar çiziliyor
subplot(223)
zamanda_ciz(tF,fF)
axis([0 5 -4 4]);
title('Süzgeçten sonraki zaman serisi')

subplot(224)
fft_hesapla_ciz(tF,fF,L,fn)

function fft_hesapla_ciz(tF,fF,L,fn)
NFFT = 2^nextpow2(length(tF)); % 2 nin sonraki tam kuvveti
```

```
Y = fft(fF,NFFT)/L;
ft = fn*linspace(0,1,NFFT/2+1);
% Tek taraflı gösterime göre çiz
plot(ft,2*abs(Y(1:NFFT/2+1)))
grid
axis([0 50 0 1]);title('Tek-Yönlü Genlik Spektrumu')
xlabel('Frekans (Hz)');ylabel('Genlik')
l=findall(gcf,'Type','line');
set(l,'LineWidth',2,'color','k')
function zamanda_ciz(t,ft)
plot(t,ft)
grid on
xlabel('Zaman(s)'); ylabel('f(t)');
```



Şekil 4. 24 Zaman serisinin süzgeç katsayıları ile evrişimi ile süzgeçleme

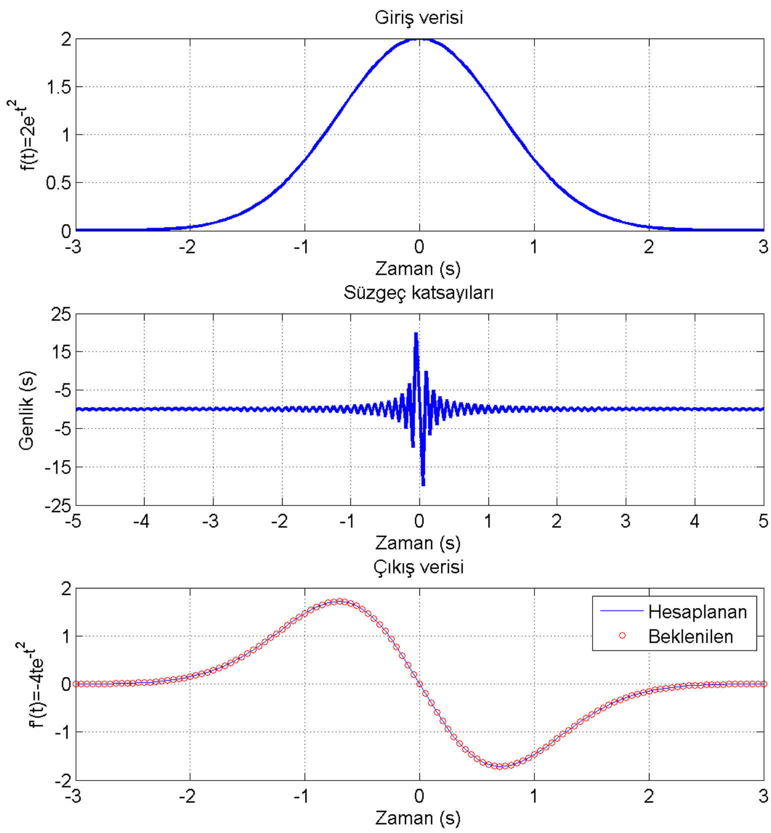
Alıştırmalar

1. Sayısal bir verinin birinci türevi doğrusal süzgeç kuramından yararlanarak $f' = f * b$ evrişimi ile hesaplanabilir. Burada b birinci türev süzgecinin katsayıları, f giriş verisi f' ise giriş verisinin birinci türevidir. Birinci türev süzgecinin katsayıları

$$b_i = \frac{\cos(2\pi f_N t_i)}{t_i} - \frac{\sin(2\pi f_N t_i)}{2\pi f_N t_i^2}$$

bağıntısı ile hesaplanabilir (Başokur, 2007). Burada f_N Nyquist frekansıdır. Sayısal evrişimin başarılı sonuç vermesi için verilerin zaman sınırlı olması gerektiği hatırlanmalıdır. Zaman sınırlılık –t ve +t yönlerine gidildiğinde fonksiyonun sıfıra yaklaşmasıdır. Diğer bir deyişle bir başlangıç ve bitiş zamanının olması, bu zaman aralığı dışında olayın gerçekleşmiyor olması gerekmektedir. Bu bilgiler ışığında

a. Süzgeç katsayılarını dt=0.05 örnekleme aralığı için hesaplatınız
b. $f(t) = 2e^{-t^2}$ fonksiyonunun sayısal değerlerini aynı örnekleme aralığı için hesaplayınız
c. Süzgeç katsayıları ile giriş verisini evriştirerek sayısal türevi hesaplayınız.
d. Sonuçları aşağıdaki gibi beklenilen (analitik) türev ile birlikte karşılaştırmalı olarak çizdiriniz.

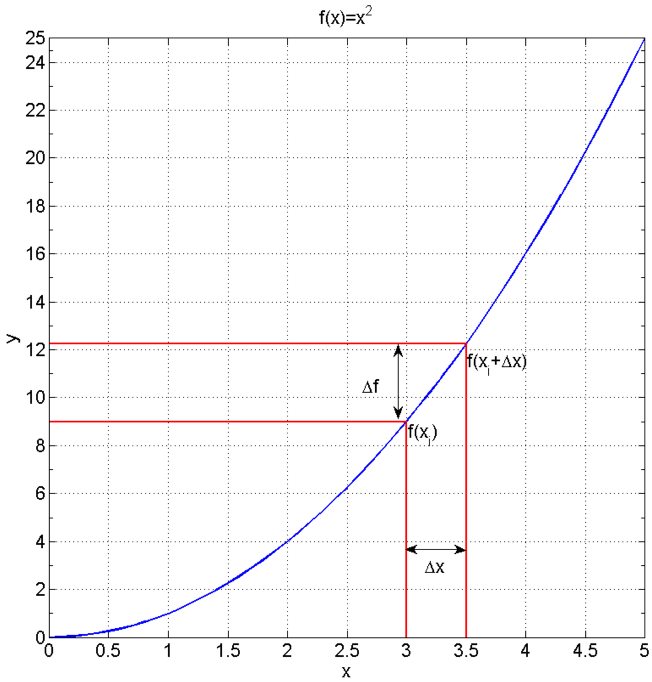


2. Bölüm 4.3 sonunda yer alan Alıştırma 1'de verilen deprem ivme kaydından 5Hz ve daha yüksek frekanslı olayları uzaklaştırınız. Süzgeçleme öncesi ve sonrasını zaman ve frekans ortamında çizdiriniz.

4.5 Sayısal Türev ve Integral

4.5.1 Sayısal Türev

Sürekli fonksiyonların türevleri analitik olarak hesaplanabilmektedir. Ancak birçok mühendislik probleminde sayısal veriler üzerinde çalışılır. Sayısal veriler bir olayın ayrık zaman, uzaklık veya frekans değerlerinde örneklenmiş yanıtıdır. Rüzgar hızının zamana bağlı ölçülmüş değerleri buna örnek verilebilir. Sayısal verilerin türev ya da integrallerinin fiziksel anlamları bulunabilir. Bu anlamlar da ilgili problemin çözümünde kullanılabilir. Dolayısı ile sayısal türev ve integral işlemleri mühendislik hesaplamalarında sıkça kullanılmaktadır. Türevin anlamı aşağıdaki grafikte gösterilmeye çalışılmıştır. $f(x)=x^2$ fonksiyonunun analitik türevi kolaylıkla alınabilir. İşlem sayısal olarak yürütülmek istendiğinde fonksiyonun ayrık yatay eksen noktalarındaki değerleri gereklidir.



Şekil 4. 25 Sayısal türevin hesaplanmasında kullanılan büyüklüklerin karşılığı

Sayısal türev yatay eksen değerindeki küçük bir değişikliğin fonksiyonun değerini nasıl değiştirdiğini göstermektedir. Bu değişikliğin farklı veriler üzerindeki anlamları farklıdır. Örneğin hareket halindeki bir cismin zamana karşı ölçülen hızlarının türevi cismin hareketinin ivmesini verir. Sayısal türev en basit hali ile

$$f\,' = \frac{\Delta f}{\Delta x}$$

bağıntısı ile ifade edilebilir. Bağıntın sağ yanındaki fark ifadeleri çeşitli yollarla hesaplanabilmektedir:

1. İleri farklar

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x}$$

2. Merkezi farklar

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x_i + \Delta x) - f(x_i - \Delta x)}{2\Delta x}$$

3. Geri farklar

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x_i) - f(x_i - \Delta x)}{\Delta x}$$

f (x) fonksiyonu bir dx aralığı ile örneklenmiş ise veya sayısal olarak kaydedilmiş verilerin örnekleme aralığı dx ise bu durumda bir noktadaki sayısal türev o noktaya komşu değerlerden yola çıkılarak hesaplanabilmektedir. İleri farklar için bu durum

$$f_i^{'} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

ifadesi ile açıklanabilir. Aynı şekilde bir x_i konumundaki sayısal türev, merkezi farklar için

$$f_i^{'} = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}$$

ve geri farklar için

$$f_i^{'} = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

şeklinde yazılabilir. Aşağıda bir sınama verisi için her üç yöntemle sayısal türev hesaplayan bir MATLAB programı verilmiştir. Bu programda matematiksel bir fonksiyonun MATLAB'da tanımlanması ile ilgili bir örnek de verilmiştir. Program içerisinde yer alan `fonk = @(x) exp(-x.^2).*sin(x);`

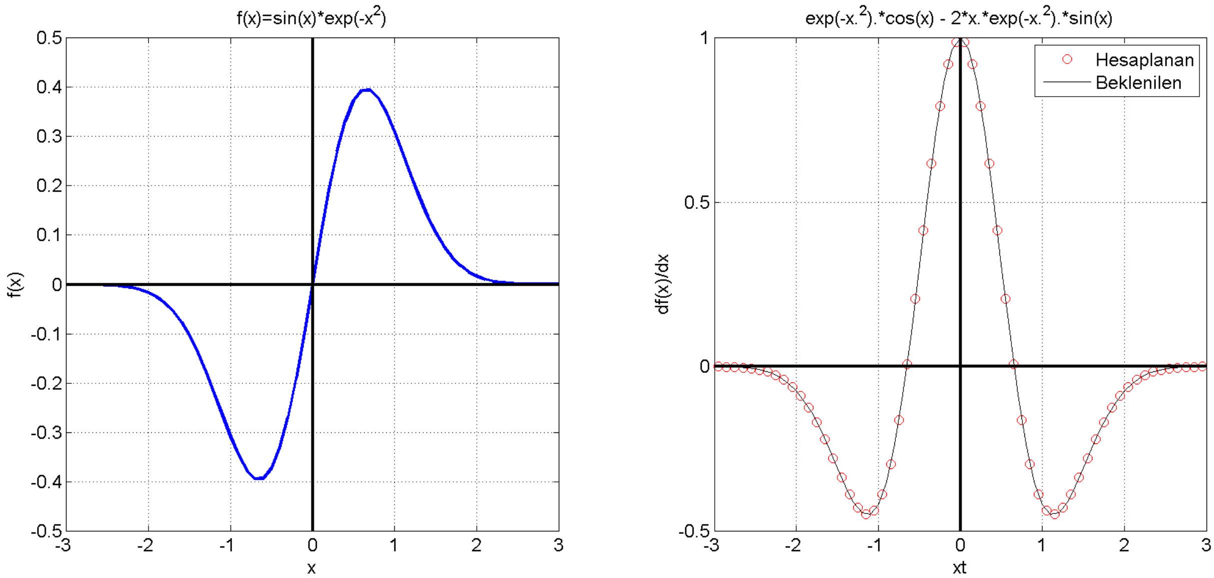
ifadesi bağımsız değişkeni x olan bir fonksiyon tanımlamaktadır. Bu tanımdan sonra `f=fonk(x)` yazımı ile x dizeyinde bulunan tüm sayısal değerler için hesaplanabilir.

```
clear all
close all
% Sınama fonksiyonu üretiliyor
dx = 0.025;
x1=-3;x2=3;
x  = x1:dx:x2;
fonk = @(x) exp(-x.^2) .* sin(x);

f=fonk(x);
% İleri farklar
            df_i = 1/dx*(fonk(x+dx)-fonk(x));
% Geri farklar
           df_g = 1/dx*(fonk(x)-fonk(x-dx));
% Merkezi farklar
            df_m = 1/(2*dx)*(fonk(x+dx)-fonk(x-dx));
```

Sayısal türev hesaplama ile ilgili daha önce `diff` MATLAB fonksiyonundan yararlanılmıştı. Bu fonksiyon genel kullanım ve sadelik açısından yine tercih edilebilir. Bu fonksiyon ardışık dizey elemanları arasındaki farkı hesapladığından bir türev işleci olarak da görev yapmaktadır. Dolayısı ile sayısal türev almanın pratik bir yolu olarak kullanılabilir. Yukarıda verilen örnek sınama fonksiyonu için sayısal türev `diff` fonksiyonu kullanılarak aşağıdaki gibi hesaplanabilir:

```
clear all
close all
%% Sınama fonksiyonu üretiliyor
dx = 0.1;
x1=-3;x2=3;
x  = x1:dx:x2;
fonk = @(x) exp(-x.^2) .* sin(x);
f=fonk(x);
%% Türev hesaplanıyor
df=diff(f)./diff(x);
xt=(x1+dx/2):dx:x2-dx/2;

%% Sonuçlar çiziliyor
subplot(211)
plot(x,f,'LineWidth',2);grid on
hold on
xlabel('x');ylabel('f(x)');title('f(x)=sin(x)*exp(-x^2)')
subplot(212)
plot(xt,df,'r o')
```

```
hold on
xlabel('xt');ylabel('df(x)/dx');
title('exp(-x.^2).*cos(x) - 2*x.*exp(-x.^2).*sin(x)')
%% beklenilen (analitik) türev
fx=exp(-x.^2).*cos(x) - 2*x.*exp(-x.^2).*sin(x);
plot(x,fx,'k');grid on
legend('Hesaplanan','Beklenilen')
```



Şekil 4. 26 Bir sınama verisi ve onun sayısal türevi

Bir sayısal verinin ikinci dereceden türevinin hesaplamak için aynı fonksiyon `d2f=diff(f,2)` şeklinde çağırılabilmektedir. Burada 2 türevin derecesini belirtmektedir. Daha yüksek dereceden türevler için bu girdi istenilen şekilde değiştirilebilir.

4.5.2 Sayısal Integral

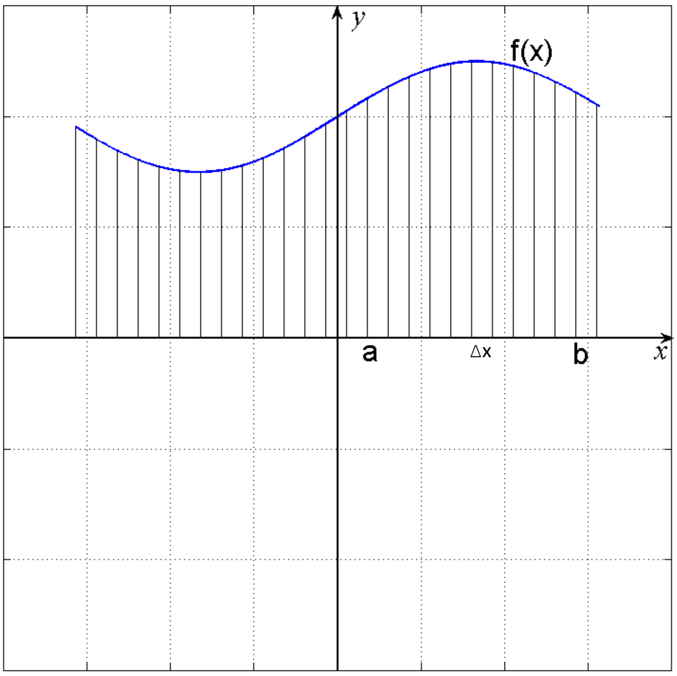
Sayısal integral bağımsız bir değişkene bağlı sayısal değerlerin tanımladığı eğri ile yatay eksen (y=0) arasında kalan alanın toplamına eşittir. Farklı veriler için konuşulduğunda integralin fiziksel anlamı da değişebilir. Örneğin hareketli bir cismin zamana göre hızının integrali cismin toplam yer değiştirmesini verecektir. Sayısal integral ile hesaplanmak istenen

$$I = \int_a^b f(x)dx$$

ifadesidir. Bu hesaplama için çeşitli yöntemler önerilmiştir. Yamuk kuralı (trapez), Simpson yöntemi, Gauss integrasyonu, Romberg integrasyonu gibi yöntemler bunlara örnek verilebilir. Bunlardan en sık kullanılanlardan biri yamuk kuralıdır. Buna göre eğri ile x ekseni arasında kalan alan küçük yamuklara ayrılarak hesaplanır (Şekil 4.27). Bir yamuğun alanı

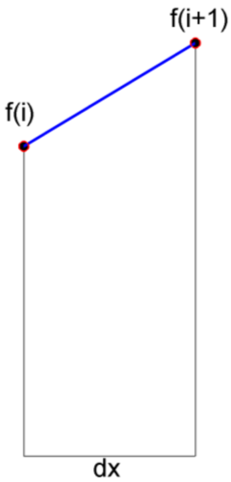
$$A = \frac{(h_1 + h_2)}{2} dx$$

bağıntısı ile verilir. Burada h_1 ve h_2 yamuğun birbirine paralel kenarlarının uzunlukları, dx ise yüksekliğidir. Şekil 4.27'da eğri ile x ekseni arasında kalan alanın, dx adımlarla ayrıklaştırılması ile oluşan yamuklar görülebilmektedir. Buradaki varsayım seçilen dx aralığında veri ya da fonksiyon değerlerindeki değişimin doğrusal olduğudur.



Şekil 4. 27 Yamuk kuralı ile integral alma

(i) fonksiyonun ayrık değerlerinin veya sayısal verilerin sıra numarasını göstermek üzere i ve i+1 numaralı veriler arasında oluşan yamuk aşağıdaki gibidir. Bu yamuğun birbirine paralel olan kenarlarının uzunlukları sırası ile f(i) ve f(i+1)'dir. Yamuğun yüksekliği ise dx kadardır (Şekil 4.28)



Şekil 4. 28 Yamuk kuralında alanın yamuklara bölünmesi

Verilen yamuğun alanı

$$A_i = \frac{f(i) + f(i+1)}{2} dx$$

bağıntısı ile hesaplanabilir. Eğri altında kalan tüm yamukların alanlarını ise

$$I \cong \frac{f(1) + f(2)}{2} dx + \frac{f(2) + f(3)}{2} dx + ... + \frac{f(\text{n}-2) + f(\text{n}-1)}{2} dx + \frac{f(\text{n}-1) + f(\text{n})}{2} dx$$

toplamı ile verilebilir. Burada toplamı oluşturan terimler incelendiğinde f(1) ve f(n) dışındaki tüm ayrık değerlerin iki kez tekrarlandığı görülebilir. Buna göre düzenleme yapıldığında

$$I \cong \frac{dx}{2} \left[f(1) + f(n) + 2\sum_{i=2}^{n-1} f_i \right]$$

ifadesi elde edilir. Yukarıdaki ifade ayrık noktalarda tanımlı f fonksiyonunun integralini yamuk kuralı ile vermektedir. Buradaki yaklaşık eşit ifadesi fonksiyonun tüm x(i), x(i+1) aralıklarında doğrusal davranması durumunda eşittir ifadesine dönüşecektir. Yamuk kuralı ile sayısal integral hesaplanması için verilen ifade aşağıda bir örnek için işletilmiştir.

```
% Sınama verisi üretiliyor
dx = .01;
x  = -pi/2:dx:pi/2;
f  = cos(x);
% Sayısal integral (yamuk kuralı ile)
I = (dx/2)*(f(1)+f(end)+2*sum(f(2:end-1)))
% Analitik değer
A = sin(pi/2)-(sin(-pi/2))
```

```
I =

    2.0000


A =

     2.0000
```

Görüleceği üzere sayısal integral ile beklenilen değerler uyumludur. Burada verilen hesaplamayı yürüten MATLAB kütüphanesinde `trapz` isimli bir fonksiyon bulunmaktadır. Fonksiyonun genel kullanımı

```
I = trapz(x,y)
```

şeklindedir. Burada x, yatay eksen değerleri, y ise bu yatay eksen değerlerindeki sayısal verilerdir. Yukarıda kullanılan sınama verisi için

```
IT=trapz(x,f)
IT=
        2.0000
```

şeklinde çağırılarak kullanılabilir. Sayısal veriler yerine bir fonksiyonun belirli bir aralıktaki integrali hesaplanmak isteniyorsa Simpson yöntemini kullanan MATLAB fonksiyonu `quad` kullanılabilir.

```
f = @(x) exp(-x)
q = quad(f,0,3)
q =
   0.9502
```

program parçası $f(x)=e^{-x}$ fonksiyonunun [0 3] aralığındaki integralini hesaplamaktadır. Burada anlatılanlar dışında sayısal türev ve integral işlemlerinde kullanılabilecek çeşitli MATLAB fonksiyonları ve kısa kullanım amaçları aşağıdaki çizelgede verilmiştir.

Çizelge 4. 1 MATLAB kütüphanesinde yer alan çeşitli sayısal türev ve integrasyon fonksiyonları

Fonksiyon adı	Kullanım amacı
`cumtrapz`	Yamukların alanlarını birikimli olarak veren integral alma fonksiyonu
`gradient`	Sayısal gradyen
`del2`	Ayrık Laplacien
`integral`	Sayısal integral
`integral2`	İki katlı integral
`integral3`	Üç katlı integral

Alıştırmalar

1. $f(x)=1-\mathrm{e}^{-x}$ fonksiyonunun sayısal türevini [0 3] aralığında dx=0.01 adım boyu için hesaplayınız. Fonksiyonun analitik türevi $f^{'}(x)=\mathrm{e}^{-x}$ bağıntısı ile hesaplanabilir. Buna göre analitik türev ile sayısal türev arasındaki hatayı

$$\chi=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(h_i-a_i)^2}$$

bağıntısı ile hesaplayınız. Burada n veri sayısını, h hesaplanan türev değerlerini, a ise analitik türev değerlerini göstermektedir. Aynı hata hesaplamasını dx'in 0.5, 0.1, 0.05 0.01 değerleri için de yaparak dx ile hesaplama hatası arasındaki ilişkiyi çizdiriniz.

2. Düz bir yol üzerinde hareket eden bir cismin zamana karşı ölçülen hız değerleri aşağıdaki gibidir. Cismin toplam yer değiştirmesini sayısal integrasyon ile bulan bir MATLAB programı yazınız.

t(s)	0	1	2.2	3.1	4.2	5	6.5	7.7	9	10
V (m/s)	0	2.81	6.22	8.81	12.01	14.35	18.80	22.41	26.36	29.44

3. http://goo.gl/2lGFPk adresinden indireceğiniz mat-dosyası içeriğinde bir deprem sırasında zamana karşı ölçülmüş hız değerleri bulunmaktadır (t ve v değişkenlerinde). Hızın zamana göre türevinin hareketin ivmesini verdiği bilgisinden yola çıkarak aynı istasyondaki yer ivme değerlerini hesaplayarak çizdiriniz. Hesaplamalarda deprem kaydının ilk 200 saniyesini kullanınız.

KAYNAKLAR

Akca Öztürk, C. 2011. Antik yerleşim alanlarında uygulanan jeoarkeolojik-yeni arkeojeofizik teknikler: Psidia Antioacheia örneği. Süleyman Demirel Üniversitesi, Doktora tezi

Başokur, A. T. 2007. Spektral Analiz ve Sayısal Süzgeçler, JFMO Eğitim Yayınları No: 8.

Başokur, A. T., Dikmen, Ü. and Tokgöz O. E., 2003. Time-frequency representation of strong-motion records for damage appraisal: examples from the Düzce earthquake (Turkey), Near Surface Geophysics, 1, 95-101. (TÜBİTAK 985-Deprem sonrasi acil araştırma programı).

http://www.mathworks.com/matlabcentral/fileexchange/ MATLAB kullanıcıları paylaşım merkezi, Erişim tarihi: 1 Mart, 2015.