

آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

*بخش سوم / رسم نمودار در MATLAB:

نرم افزار MATLAB امکان رسم انواع گوناگونی از نمودارهای دو بعدی و سه بعدی را برای کاربران خود فراهم نموده است. در اینجا به بررسی چند نمونه از این نمودارها و چند دستور که برای ویرایش شکل نمودارها به کار میرود می پردازیم.

الف) رسم خط و منحنی:

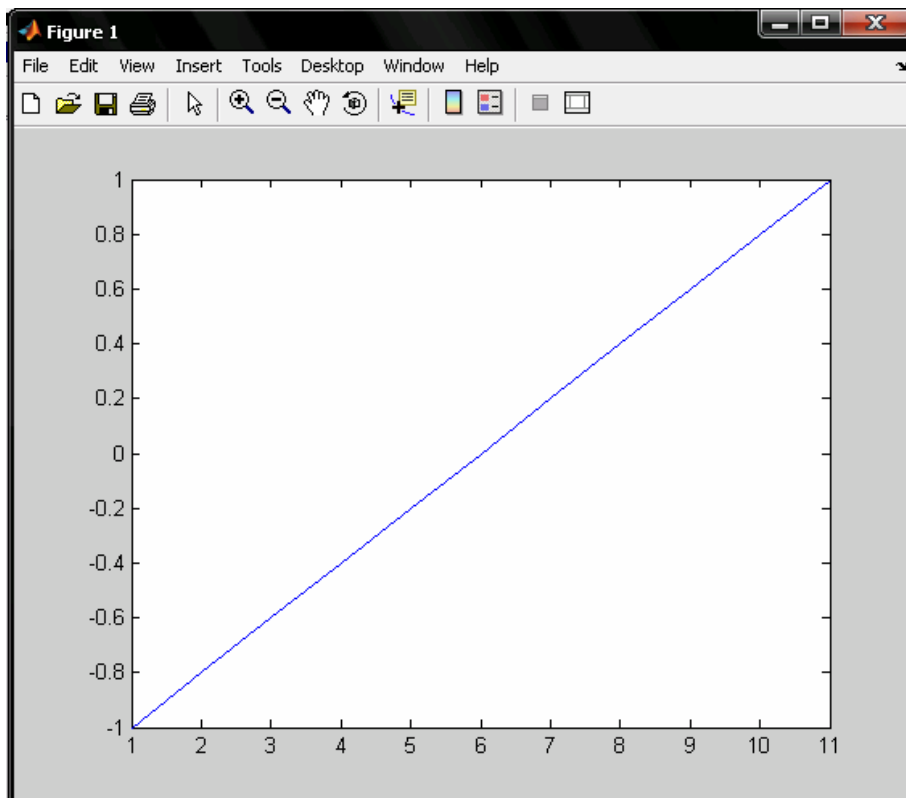
الف-۱: رسم خط و منحنی در صفحه (2-D):

plot: از این دستور برای رسم خط و منحنی در صفحه (دو بعدی) استفاده می شود. این تابع نیز مانند سایر توابع موجود در MATLAB بسته به تعداد و نوع ورودیهایش نتایج متفاوتی ارائه می دهد. اگر تابع یک ورودی از نوع ماتریس سطری داشته باشد یک خط پیوسته در یک محور دو بعدی رسم می شود که محور افقی آن نشان دهنده ی شماره ی درایه و محور عمودی نشان دهنده ی مقدار هر درایه است. مثلاً به نمودار رسم شده برای آرایه ی X توجه کنید:

```
>> X=-1:2:1
```

```
>> plot(X)
```

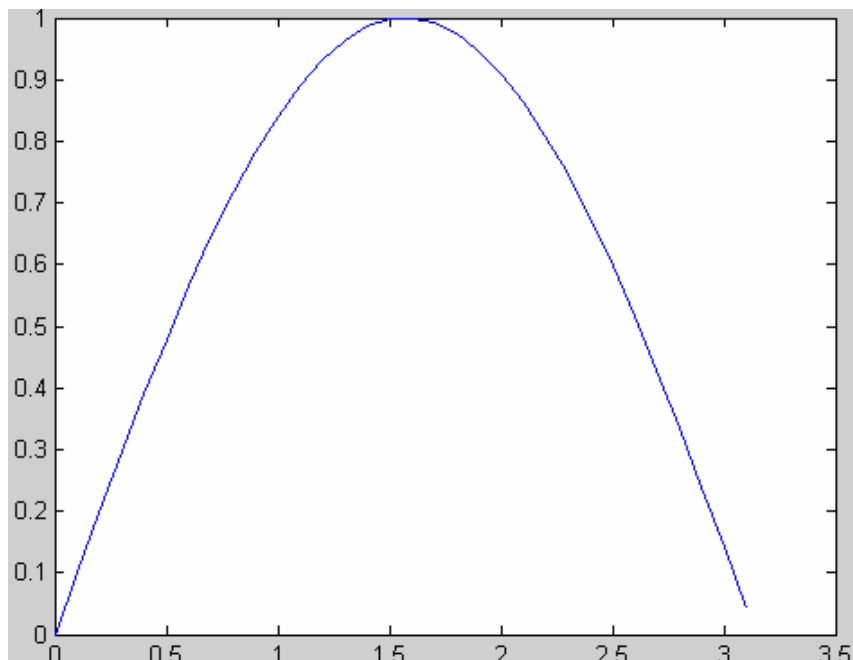
نتیجه ی اجرای دستور، نمودار زیر است که در پنجره ی **figures** رسم می گردد. ما در اینجا نمودار را در پنجره ی **figures** نمایش داده ایم، اما بعد از این برای آنکه نمودارها فضای کمتری اشغال کنند، از نشان دادن خود پنجره ی **figures** صرف نظر نموده و تنها خود نمودارها را نشان می دهیم. با دقت در نمودار ملاحظه می کنید که محور افقی بر اساس تعداد درایه های X شماره گذاری شده و محور عمودی نشان دهنده ی مقادیر درایه ها از کمترین تا بیشترین مقدار است. هر درایه با خطی به درایه ی بعد وصل شده است و در نتیجه یک خط پیوسته بوجود آمده است:



آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

گاهی شما مایلید یک متغیر را به عنوان تابعی از متغیر دیگر رسم کنید. بدین منظور باید ابتدا به متغیر مستقل مقادیری در قالب یک ماتریس سطری اختصاص دهید، سپس تابع آنرا تعریف کنید. حال با استفاده از تابع پلات که متغیر مستقل به عنوان پارامتر ورودی اول و تابع به عنوان آرگومان دوم به آن داده شده است، می توانید نمودار تابع بر حسب متغیر مستقل را رسم کنید. به عنوان مثال اگر y تابعی از x باشد و بخواهیم y را بر حسب x رسم کنیم از `plot(x,y)` استفاده می کنیم. البته اجباری نیست که حتماً y به x وابسته باشد. به هر حال محور طول های نمودار بر اساس x و محور عرض ها خواهد بود:

```
>> x=0:0.1:pi;
>> y=sin(x);
>> plot(x,y)
```



در تابع پلات و توابعی که برای رسم نمودارهای خط و منحنی به کار می روند، علاوه بر ورودی های اصلی، تعدادی ورودی که اغلب از نوع متن هستند نیز می تواند باشد که در واقع شکل ظاهری نمودار را تنظیم می کند. شکل خط میتواند نقطه چین، خط و نقطه، یا دارای برخی علائم مانند $*$ ، $+$ یا برخی اشکال هندسی باشد. همچنین می توانید رنگ خط را نیز تعیین کنید. این چند کار همزمان با نوشتن علامت مورد نظرتان برای رسم خط و حرفی که معرف رنگ خط است و نوشتن نمادی که میخواهید روی خط ظاهر شود، داخل یک ' ' انجام می گیرد. مثلاً اگر بنویسید '`--r*`' آنگاه خط شما بریده بریده و قرمز رنگ خواهد بود و با ستاره هایی روی آن علامتگذاری می شود. علائم روی خط در واقع جایگاه یکایک درایه ها را نشان می دهند.

آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

جداول زیر حرف مربوط به هر رنگ و علائمی را که می توانید استفاده کنید تا خط را رسم و روی آن علامت گذاری کنید ، بیان می کند.

Specifier	Marker Type
+	Plus sign
o	Circle
*	Asterisk
.	Point
x	Cross
'square' or s	Square
'diamond' or d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
'pentagram' or p	Five-pointed star (pentagram)
'hexagram' or h	Six-pointed star (hexagram)

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line

جدول انواع خطوط

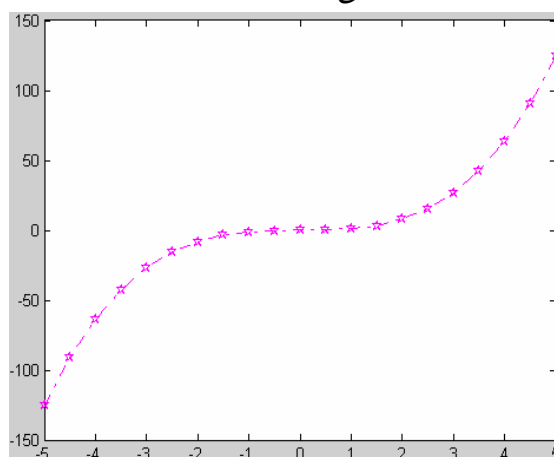
Specifier	Color
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White

جدول انواع علائم برای نشانه گذاری جای داده ها

جدول رنگ ها

مثال زیر نمونه ای از استفاده ی از این جداول را نشان می دهد.

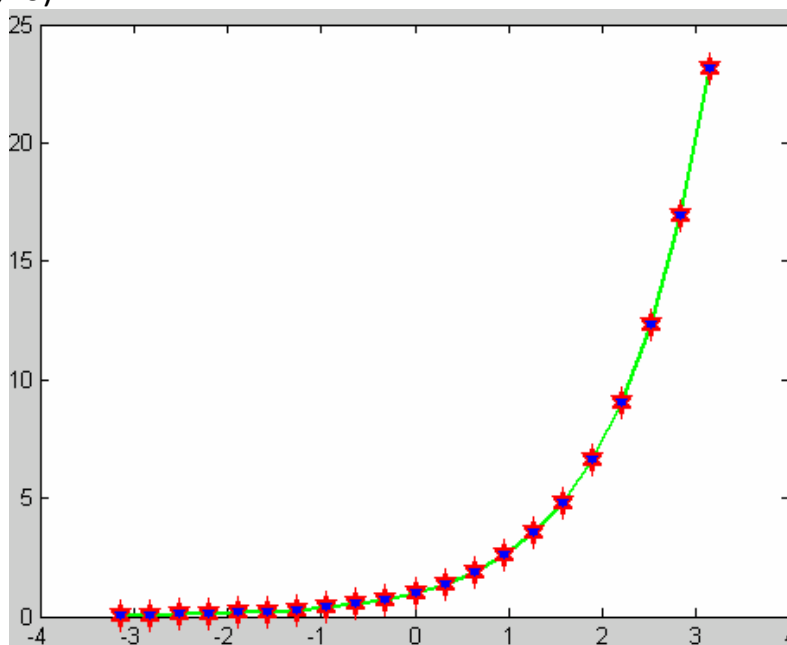
```
>> t=-5:0.5:5;
>> plot(t,t.^3,'-mp')
```



آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

برای سایر اعداد و علائمی که برای ویرایش خط استفاده می شوند ابتدا نام آنها را به صورت متن و سپس عدد یا علامت مربوطه را به عنوان ورودی بعدی وارد می کنید. مثلاً برای تعیین ضخامت خط ابتدا عبارت **'linewidth'** و پس از آن عدد مورد نظر را به تابع می دهیم. عبارات دیگر **'markeredgecolor'** برای تعیین رنگ دور علائم نشانه گذاری، **'markerfacecolor'** برای تعیین رنگ داخل علائم نشانه گذاری و **'markersize'** برای مشخص نمودن اندازه ی علائم نشانه گذاری استفاده میشوند. تاثیر این عبارات را در مثال زیر مشاهده کنید.

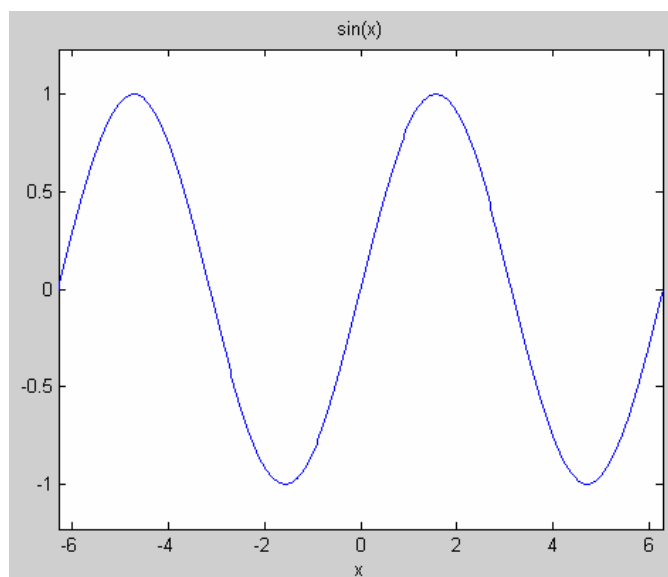
```
>> a=(-1:1:1)*pi;
>> plot(a,exp(a),'-gh','linewidth',2,'markeredgecolor','r','markerfacecolor'...'b','markersize',10)
```



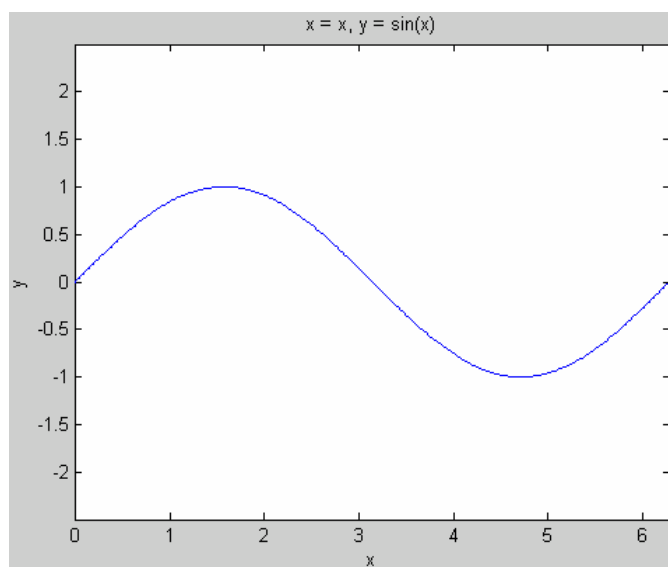
دستور **grid on** باعث می شود که صفحه ی رسم نمودار شما متناسب با اسکال نمودارتان تقسیم بندی گردد. البته روشن و خاموش کردن این تقسیم بندی ها و بسیاری تنظیمات دیگر که در اینجا با استفاده از دستورات اعمال نمودیم از طریق ابزار تنظیماتی که در پنجره ی **figures** قرار داده شده است نیز قابل تنظیم می باشد. با این ابزارها در قسمتهای قبل آشنا شدیم.

ezplot: این تابع همانند تابع **plot** عمل می کند، با این تفاوت که حوزه ی کاربرد آن در مورد متغیرهای **symbolic** است. متغیر اول و دوم همانند تابع **plot** متغیر مستقل، و تابع وابسته به آن هستند. در صورتی که متغیر دیگری وارد نکنیم به طور خودکار نمودار به ازای بازه ی $[0, 2\pi]$ برای متغیر مستقل رسم می شود و اگر ورودی اول و دوم به هم وابسته نباشند آنگاه برای هر یک بازه ی پیش فرض $[-2\pi, 2\pi]$ خواهد بود. اما شما می توانید خودتان یک بازه به عنوان ورودی سوم برای تابع **ezplot** معرفی کنید. مثال:

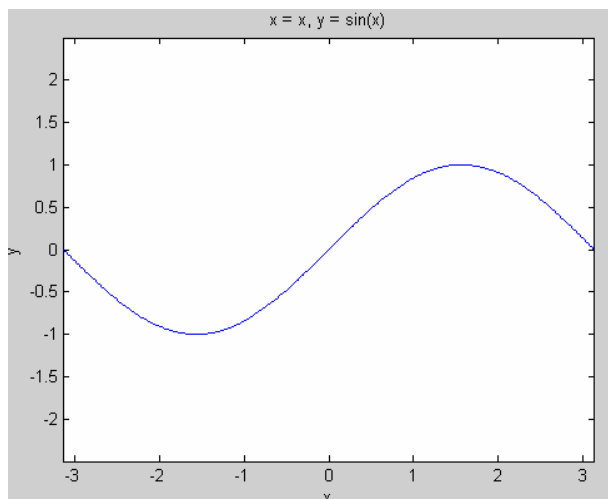
```
>> syms x y
>> ezplot(sin(x))
```



```
>> ezplot(x,sin(x))
```



```
>> y=sin(x);  
>> ezplot(x,y,[-pi,pi])
```



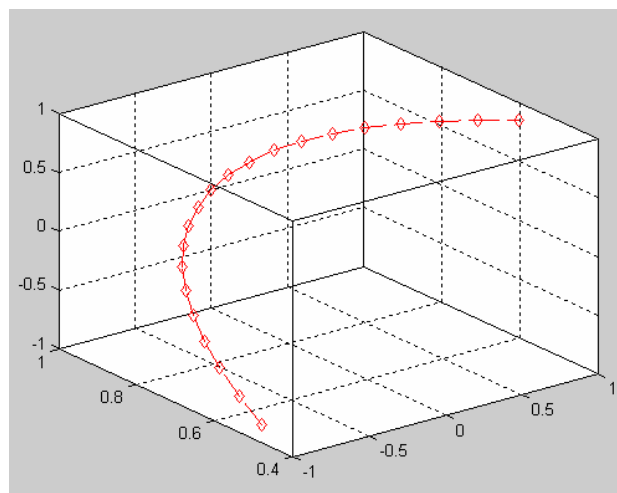
آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

ورودی‌هایی که جهت تنظیم نمودار در مورد تابع `plot` وارد می‌کنیم در مورد `ezplot` قابل استفاده نیست و `error` میدهد. البته روش‌هایی برای تنظیم نمودار در این حالت نیز وجود دارد که به دلیل طولانی شدن موضوع از پرداختن به آن می‌پرهیزم.

الف-۲: رسم خط و منحنی در فضا (3-D):

plot3: تابع `plot3` برای رسم نمودار در فضا به کار می‌رود. توجه داشته باشید که نتیجه‌ی استفاده از این دستور نیز یک خط است، با این تفاوت که این خط دارای بعد بوده و در یک محور سه بعدی رسم می‌گردد. تفاوت کار با این نمودار آنست که سه متغیر به آن داده می‌شود که متغیر اول محور X ، متغیر دوم محور Y و متغیر سوم محور Z را تعیین می‌کند. توجه داشته باشید که هر سه متغیر باید ماتریسهای سطری با طول برابر باشند. در اینجا حتماً باید حداقل سه متغیر مربوط به سه محور وارد شوند. پس از سه ورودی اصلی، می‌توانید عبارات مربوط به ویرایش نمودار را نیز وارد کنید. نمودار حاصله سه بعدی بوده و پس از رسم، با فعال کردن ابزار  از نوار ابزار بالای پنجره `figures` و تغییر شکل موس به صورت یک دایره، می‌توانید کلید سمت چپ موس را در حالی که موس را روی نمودار قرار داده اید نگه دارید، و با حرکت دادن آن به اطراف، نمودارتان را بچرخانید و از جهت‌های دیگر نیز آنرا بررسی کنید. اگر در این شرایط روی نمودار کلیک راست کنید، امکاناتی عبارت از برگرداندن شکل به نمای اولیه (`Reset To Original View`)، دیدن شکل از نمای یکی از صفحه‌ها ($X-Y$ ، $Y-Z$ یا $X-Z$) و امکاناتی برای تنظیم چگونگی چرخش محور و نمودار (`Rotate Option`)، در اختیار شما قرار می‌گیرد. برای آنکه از هم اندازه بودن طول متغیرهای ماتریسمان اطمینان حاصل کنیم می‌توانیم از روشهای گوناگونی استفاده کنیم. قرار دادن یک متغیر واسطه (مثلاً t)، تعریف یک متغیر و تابع قرار دادن بقیه‌ی متغیرها به آن و یا استفاده از توابع `length` و `linspace` از آن جمله‌اند. به نمونه‌هایی از نمودارهایی که با استفاده از دستور `plot3` رسم شده‌اند توجه کنید:

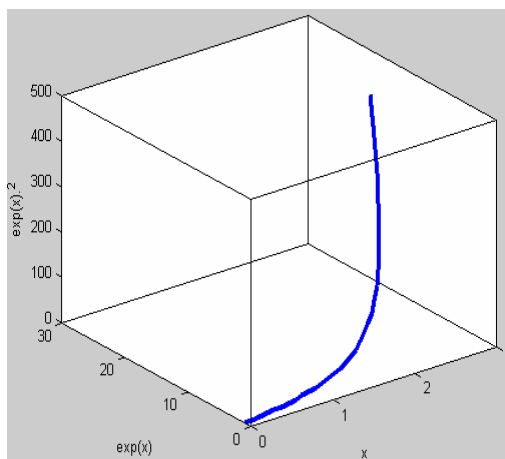
```
>> t=-1:1:1;
>> x=sin(t);
>> y=cos(t);
>> z=t;
>> plot3(x,y,z,'-rd')
>> grid on
>> box on
```



آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

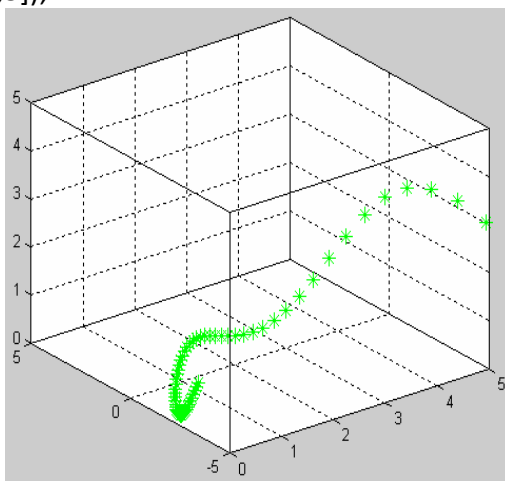
همانطور که می بینید نوشتن دستور `grid on` پس از دستور `plot3` موجب اسکیل بندی صفحه شده است. دستور `box on` باعث می گردد که محور نمودار به صورت یک جعبه ی شفاف در آید که این حالت نمودار، تجسم شکل سه بعدی را راحتتر می کند. توجه داشته باشید که دستورات ویرایش نمودار و یا محور آن، پس از دستور `plot3` باید نوشته شوند تا روی آن تأثیرگذار باشند.

```
>> x=0:.1:3;
>> y=exp(x);
>> z=y.^2;
>> plot3(x,y,z,'linewidth',3)
>> xlabel('x');ylabel('exp(x)');zlabel('exp(x).^2');
>> box on
```



دستورات `xlabel`، `ylabel` و `zlabel` برای آنست که روی محورها همان نام بگذاریم تا روی تصویر بتوانیم آنها را بهتر پیدا کنیم.

```
>> a=-pi:pi/18:pi;
>> b=logspace(-1,1,length(a));
>> c=sin(a)+cos(b);
>> plot3(b,a,c,'g*')
>> grid on
>> box on
>> xlim([0,5]);ylim([-5,5]);zlim([0,5]);
```

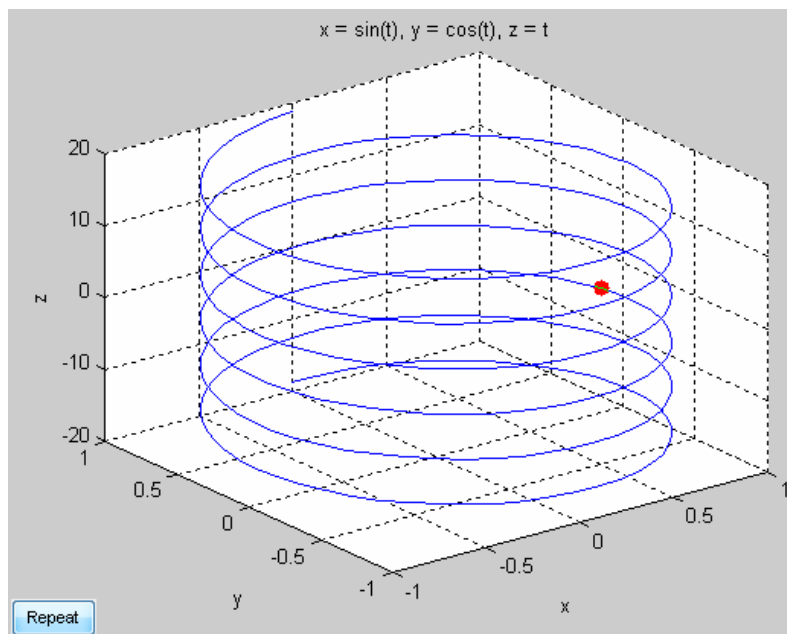


آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

توابع محدود کننده ی محورها را در مثال فوق مشاهده نمودید. این توابع که ورودی آنها حدود هر محور را تعیین می کند، برای یک نمودار سه بعدی `ylim`، `xlim` و `zlim` هستند.

ezplot3: در اینجا شما می توانید به ازای متغیرهای `symbolic` یک منحنی سه بعدی رسم کنید. مانند تابع `ezplot` محدوده ی پیش فرض برای متغیر مستقل -2π تا 2π است. به عنوان مثال اگر متغیر هر سه محور به یک متغیر مانند t وابسته باشند، t در این محدوده می باشد. اگر پس از سه ورودی اصلی، یک بازه قرار دهیم، این بازه حدود t را مشخص میکند. این تابع فقط در این حالت معنا دارد، چرا که `ezplot3` یک نمودار دو متغیره نیست و حتماً باید متغیری باشد که هر سه متغیر ورودی تابع به آن وابسته باشند. مثال:

```
>> syms t
>> ezplot3(sin(t),cos(t),t,[-6*pi,6*pi],'animate')
```

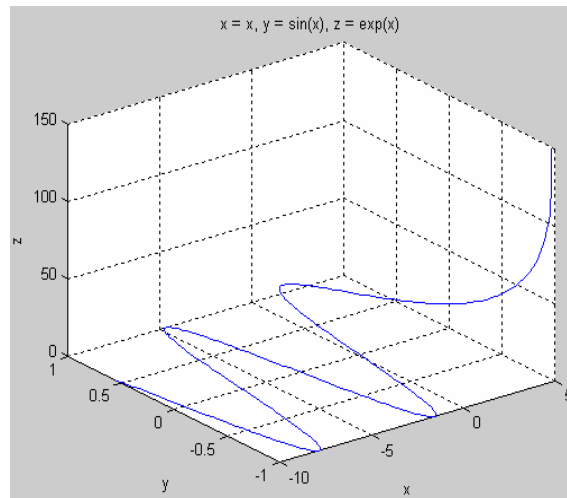


لگه ی قرمزی که روی محور مشاهده می کنید و دکمه ی `Repeat` که در پایین صفحه ظاهر شده است، اثر عبارت `'animate'` در ورودی تابع می باشد. لگه ی قرمز پس از رسم نمودار، از کوچکترین نقطه، شروع به حرکت نموده و روی نمودار تا انتهای آن به مسیر خود ادامه می دهد. دکمه ی `Repeat` برای تکرار این حرکت است. این لگه ی متحرک مخصوصاً در نمودارهای پیچیده شکل نمودار را قابل تجسم تر می سازد.

حال مثالی دیگر:

```
>> syms x
>> ezplot3(x,sin(x),exp(x),[-10,5])
```

آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)



البته توابع دیگری نیز برای رسم انواع خاصی از نمودار خطها و منحنی ها وجود دارد که به علت آنکه کاربرد آنها به گستردگی موارد ذکر شده نیست از بیان آنها خودداری می کنم.

(ب) رسم رویه ها در فضا :

زمانی که تابع ما وابسته به دو متغیر مستقل است، انتظار می رود که نمودار آن سه بعدی و به صورت یک رویه ی فضایی باشد. توابعی که در این بخش برای رسم نمودار معرفی می کنیم از لحاظ عملکرد بسیار مشابه یکدیگر هستند و تنها شکل ظاهری خروجی آنها و نوع ویرایشی که می توانید روی آن اعمال کنید متفاوت است.

ب-۱) توابع دو متغیره ی عددی : در مورد توابع عددی، ابتدا باید دو متغیر مستقلمان را طوری تغییر دهیم که اولاً همبند شوند (تعداد سطر و ستون برابر داشته باشند)، ثانیاً همزمان با هر یک از درایه های یکی از آن دو که در تابع قرار می گیرد، تک تک درایه های دیگری نیز در تابع قرار گرفته و نقطه ی حاصله ثبت گردد. یعنی مثلاً اگر متغیرهای مستقل ما x و y باشند، مقدار تابع به ازای درایه ی اول x و درایه ی اول y ، درایه ی اول x و درایه ی دوم y ، درایه ی اول x و درایه ی سوم y ، الی درایه ی اول x و درایه ی آخر y محاسبه و نقطه ی آن در فضا پیدا شود و به همین ترتیب به ازای هر یک از درایه های x ، تمامی درایه های y در تابع قرار گرفته و مقدار تابع در یک زوج مرتب از x و y پیدا شود. تابع `meshgrid` متغیرهای ما را طوری گسترش می دهد که متغیرهای جدید همبند و قابل استفاده ای پدید آید. برای این کار ابتدا دو متغیر مستقلمان را تعریف و سپس به عنوان ورودی به تابع `meshgrid` می دهیم. تابع را مساوی یک ماتریس دو متغیره که نام دو متغیر جدیدمان را به عنوان درایه های آن در نظر گرفته ایم، قرار می دهیم. مثلاً:

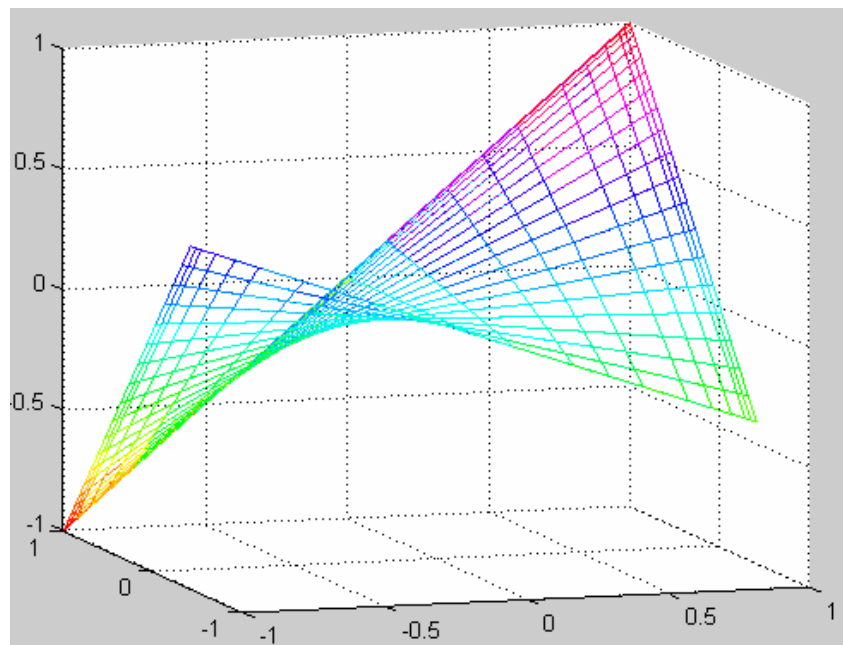
```
>> x=[1,2,3];
>> y=[4,5];
>> [X,Y]=meshgrid(x,y)
X=
     1     2     3
     1     2     3
Y=
     4     4     4
     5     5     5
```

آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

اکنون متغیرهای دگرگون شده می توانند به عنوان متغیر در تعریف تابعی که مایلیم رویه ی فضایی آن را رسم کنیم ، به کار روند.

برخی توابعی که برای رسم رویه به کار می روند عبارتند از `surf`، `surfl`، `mesh`، `meshc`، `meshz` ، `contour3` و `waterfall` و در تمامی این توابع دو متغیر مستقل به عنوان دو ورودی اول و سپس تابع آنها که در واقع مقادیر محور Z را تعیین می کند به عنوان ورودی سوم به تابع داده می شود. دستورات مربوط به تنظیمات و ویرایش نمودار در خطوط بعدی برنامه نوشته می شوند. حال شما را با شکل خروجی این چند تابع و توابعی که برای ویرایش نمودار به کارتان می آید آشنا می کنم. کار را با چند مثال ادامه می دهیم:

```
>> x=sin(-5:.5:5);
>> y=exp(-2:.1:0);
>> [x,y]=meshgrid(x,y);
>> z=x.*y;
>> mesh(x,y,z)
>> hidden off
>> colormap hsv
```

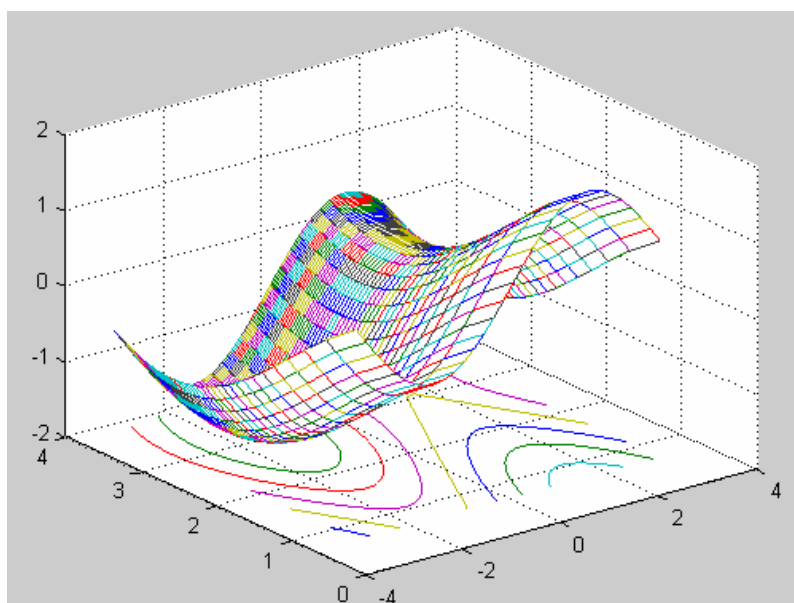


همانطور که می بینید تابع `mesh` یک نمودار مشبک را رسم می کند. دستور `hidden off` باعث می شود که قسمتهای عقب، پشت بخش های جلوتر پنهان نشده و قابل رؤیت باشند.

`colormap` دستوریست برای تعیین طیف رنگی که نمودار شما با آن رنگ آمیزی می شود. انواع گوناگون آن در جدول زیر آورده شده است. با قرار دادن نام هر حالت مقابل تابع `colormap` ، نمودارمان با آن طیف رنگ، رنگ آمیزی میشود :



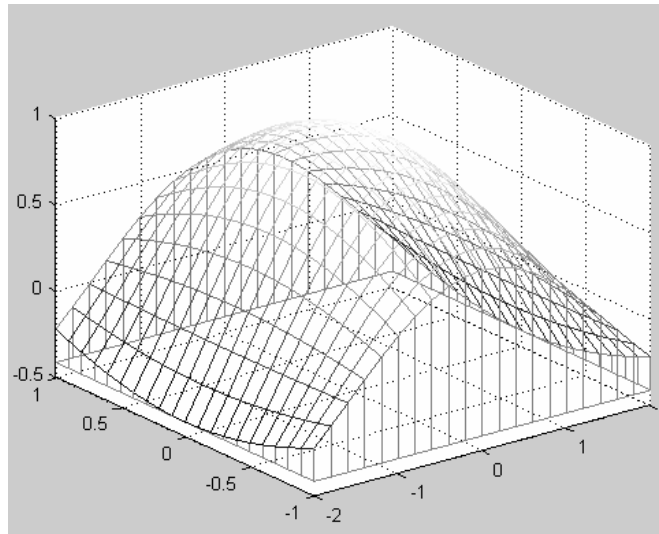
```
>> x=-pi:.2:pi;
>> y=log(2:50);
>> [a,b]=meshgrid(x,y);
>> meshc(a,b,sin(a)+cos(b))
>> colormap lines
```



دیدیم که خروجی تابع `meshc` شبیه به `mesh` است، با این تفاوت که روی صفحه ی زیرین تصویری سایه گونه از نمودار افتاده است. طیف رنگی انتخاب شده نیز `lines` می باشد که تأثیر آن بر شکل کاملاً مشهود است. ضمناً چون دستور `grid off` اینجا به کار نرفته، قسمتهای عقبی پشت بخش های جلو تر پنهان شده است.

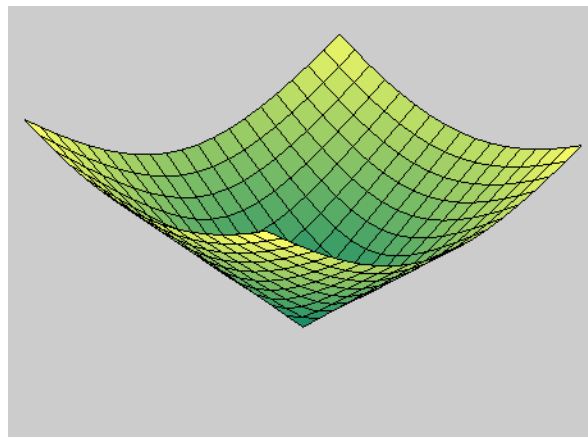
```
>> x=-1:.1:1;
>> y=-2:.2:2;
>> [x,y]=meshgrid(x,y);
>> meshz(y,x,cos(x).*cos(y))
>> hidden off
>> colormap gray
```

آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)



meshz نیز نوع دیگری از mesh است که نقاط لبه ای آن با خطوطی عمود به صفحه ی زیرین متصل شده اند.

```
>> x=-5:5:5;
>> y=-4:5:4;
>> [x,y]=meshgrid(x,y);
>> surf(x,y,(x.^2+y.^2).^0.5)
>> colormap summer
>> axis off
```

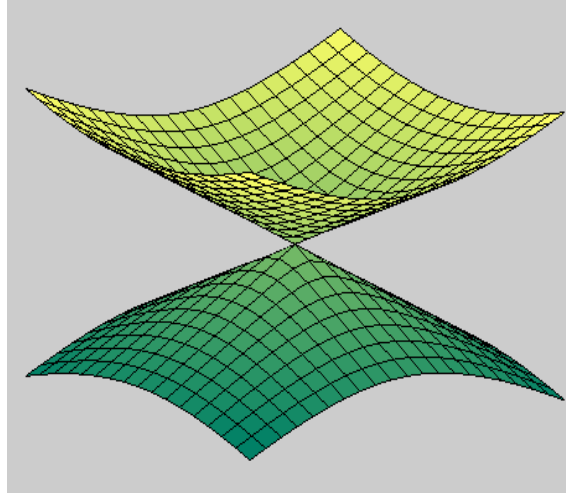


همانطور که ملاحظه می کنید تابع surf یک نمودار مشبک با شبکه های تو پر تولید می کند. دستوری که باعث ناپدید شدن محورها شده است دستور axis off است. تابع axis نیز مانند colormap تنظیمات گوناگونی را می پذیرد که همگی در شکل و اندازه ی محورهای نمودار تأثیر می گذارند. می توانید با مراجعه به help انواع این دستور را ملاحظه کنید. می دانید که تابع به کار رفته در این مثال تابع نمودار یک مخروط است. فرض کنید بخواهیم بخش منفی این نمودار را هم همزمان با بخش بالایی مشاهده کنیم. برای اینکار لازم است اول بخش بالایی رسم و بعد در حالی که این بخش را نگه میداریم بخش پایینی را رسم کنیم. هر گاه بخواهیم نموداری در همان صفحه ی نمودار قبل و با حضور آن رسم گردد از دستور hold on استفاده می کنیم.

به عنوان مثال، این دستورات را در ادامه ی دستورات قبل می نویسیم:

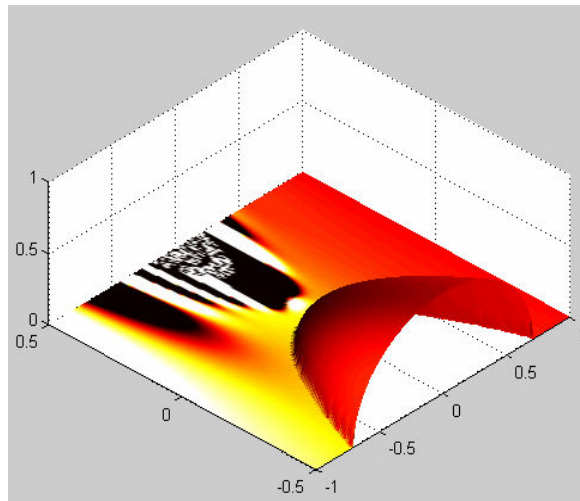
آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

```
>> hold on
>> surf(x,y,-(x.^2+y.^2).^0.5)
```



surf: این دستور را با یک مثال بیان می کنم:

```
>> x=-1:.01:1;
>> y=-.5:.01:.5;
>> [a,b]=meshgrid(x,y);
>> surfl(a,b, (-a.^2-b).^0.5)
>> colormap hot
>> shading interp
```



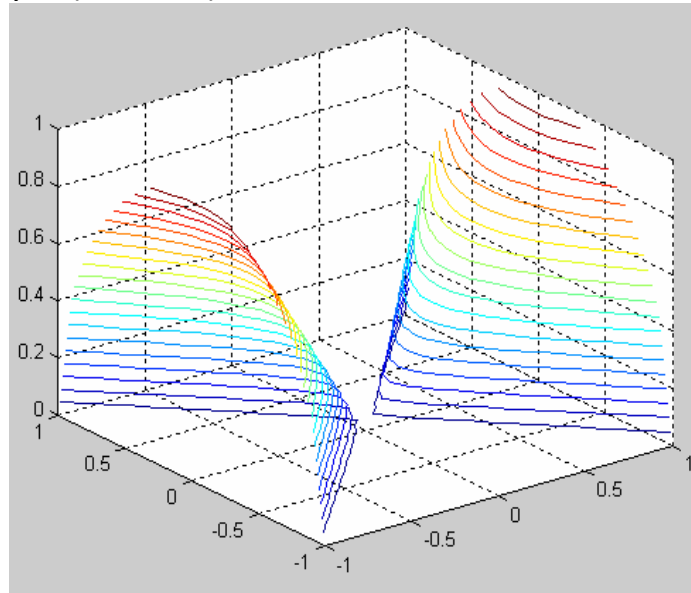
surfl باعث می شود که از یک سمت نور به نمودار تابیده شود و سطح نمودار سایه و روشن گردد. دستور shading که سه حالت flat, faceted و interp دارد، برای محو کردن خطوط روی سطح نمودار استفاده می شود. کاملترین حالت برای interp رخ می دهد که هم خطوط محو می شوند و هم تیزی های سطح نمودار گرفته شده و سطحی صاف خواهیم داشت.

surf نیز همانند meshc تصویری از نمودار بر صفحه ی زیر آن خواهد داشت.

آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

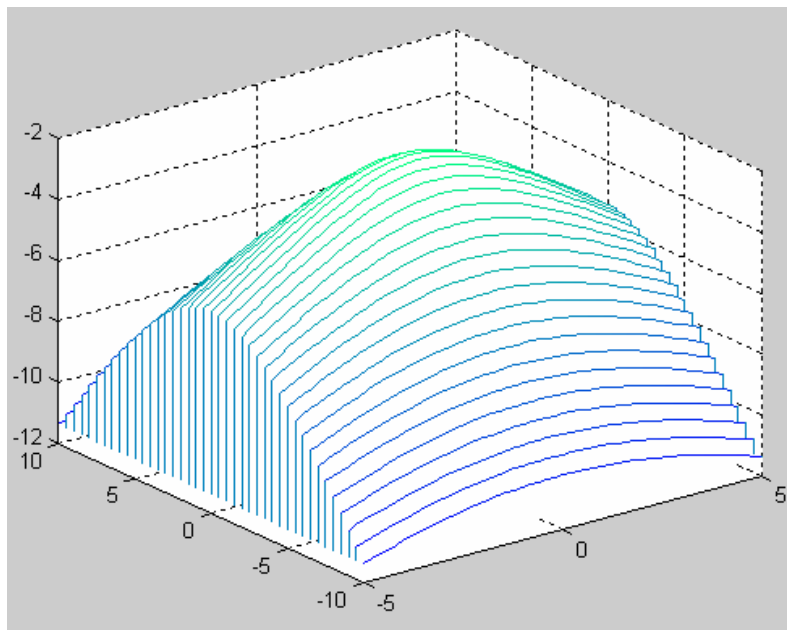
contour3 که یک عدد طبیعی به عنوان ورودی چهارم خود می پذیرد، به اندازه ی این عدد نمودار را برش داده و تنها مقاطع بریده شده را نمایش می دهد.

```
>> x=-1:1:1;
>> [x,y]=meshgrid(x);
>> contour3(x,y,(x.^2-y.^2).^0.5,20)
```



نمودار بعدی که ما بررسی می کنیم نمودار حاصل از اجرای دستور **waterfall** است. همانگونه که از نامش نیز پیداست نموداری آبشاری شکل رسم می کند.

```
>> x=-10:5:10;
>> y=-5:2:5;
>> [x,y]=meshgrid(y,x);
>> waterfall(x,y,-(x.^2+y.^2+4).^0.5)
>> colormap winter
```

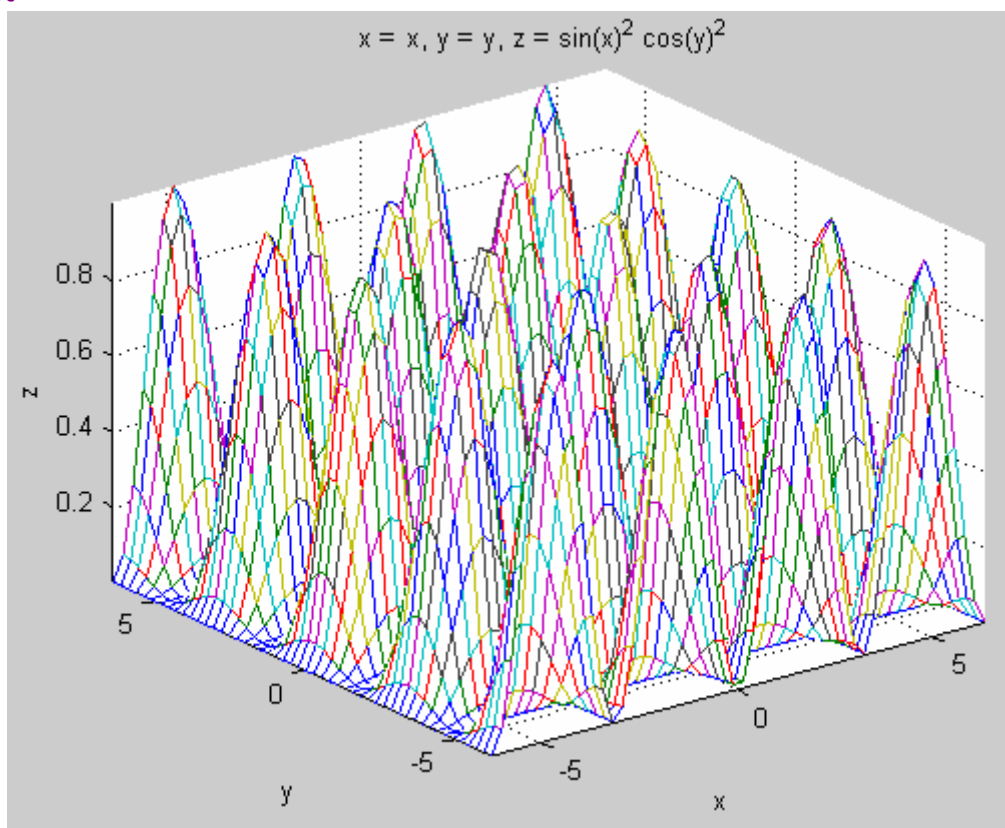


آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

ب-۲) توابع دو متغیره **ن سیمبلیک (غیر عددی یا پارامتری):** برخی از انواع نمودارهایی که برای توابع دو متغیره ی عددی قابل رسم بودند برای توابع دو متغیره ی symbolic نیز قابل استفاده هستند، با این تفاوت که به ابتدای نام آنها ez اضافه شده و متغیرهایی که به آنها داده می شود دیگر ماتریسهای عددی نیستند، بلکه متغیرهای پارامتری هستند. این توابع عبارتند از ezsurf, ezmeshc, ezmesh و ezsurf. وارد کردن متغیرها همانند توابع بخش عددی است.

پس از وارد کردن متغیرهای اصلی، ورودیهای دیگری نیز می توان به تابع داد. به عنوان مثال به صورت پیش فرض متغیرهای مستقل بین -2π و 2π در نظر گرفته می شود، اما شما می توانید یک ماتریس سطری دو متغیره برای تعیین حدود هر دو متغیر به صورت یکسان و یا یک ماتریس چهار متغیره که به ترتیب مقدار مینیمم و ماکزیمم متغیر اول و مقدار مینیمم و ماکزیمم متغیر دوم را نشان میدهد به تابع دهید. همچنین به صورت پیش فرض نمودار به گونه ای رسم می شود که از در کنار هم قرار گرفتن 60 بند تشکیل گردد. شما می توانید یک عدد طبیعی وارد کنید که تعیین کننده ی تعداد بندها به دلخواه شماست. بدیهیست هر چه تعداد این بندها بیشتر باشد شکل کاملتر است. تمامی ویرایشهایی که در مورد رویه های توابع عددی به کار می بردیم اینجا نیز قابل اعمال هستند. به مثالهای زیر که مطالب فوق را پوشش می دهند توجه کنید:

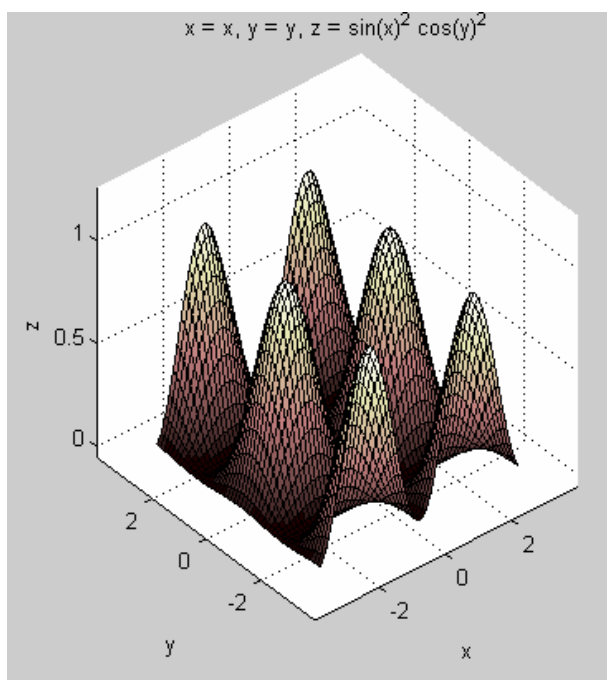
```
>> syms x y
>> z=sin(x)^2*cos(y)^2;
>> ezmesh(x,y,z)
>> colormap lines
>> axis tight
```



آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

همانطور که مشاهده نمودید x و y هر دو در بازه $[-2\pi, 2\pi]$ قرار دارند. دقت کنید که اینجا دیگر نیازی به تابع `meshgrid` نداریم. همچنین لازم نیست از عملگرهای ریاضی ماتریسها (* ، $./$ و $^{\wedge}$) استفاده کنیم. دستور `axis tight` که قبلاً نیز با نوع دیگری از آن آشنا شده بودیم (`axis off`) مرزهای محور نمایش را طوری تنظیم می کند که با لبه های نمودار رسم شده هم اندازه باشند و به این ترتیب اضافه ی محور که از حدود نمودار تجاوز می نمود حذف می گردد. حالا نمودار `ezsurf` را برای همین تابع ولی با بازه ی دیگری از متغیرها رسم می کنیم و ویرایشهای دیگری را روی آن اعمال می کنیم:

```
>> syms x y
>> z=sin(x)^2*cos(y)^2;
>> ezsurf(x,y,z,[-3,3])
>> colormap pink
>> axis square
```

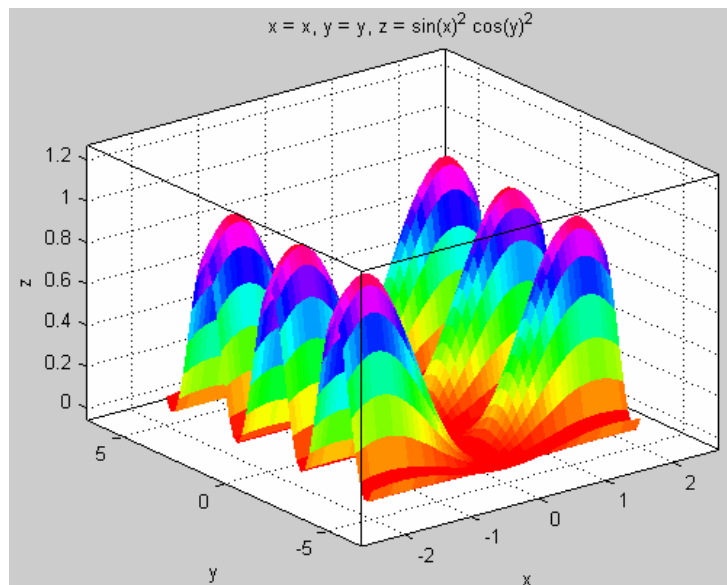


می بینید که مینیمم مقدار برای x و y ، -3 و مقدار ماکزیمم 3 است. در ضمن استفاده از دستور `axis square` محور را به صورت یک مکعب مربع در می آورد.

اکنون این نمودار را با همین دستور اما به ازای دو محدوده ی متفاوت برای x و y رسم می کنیم. از دستور `shading flat` هم برای محو کردن خطوط روی نمودار استفاده می کنیم.

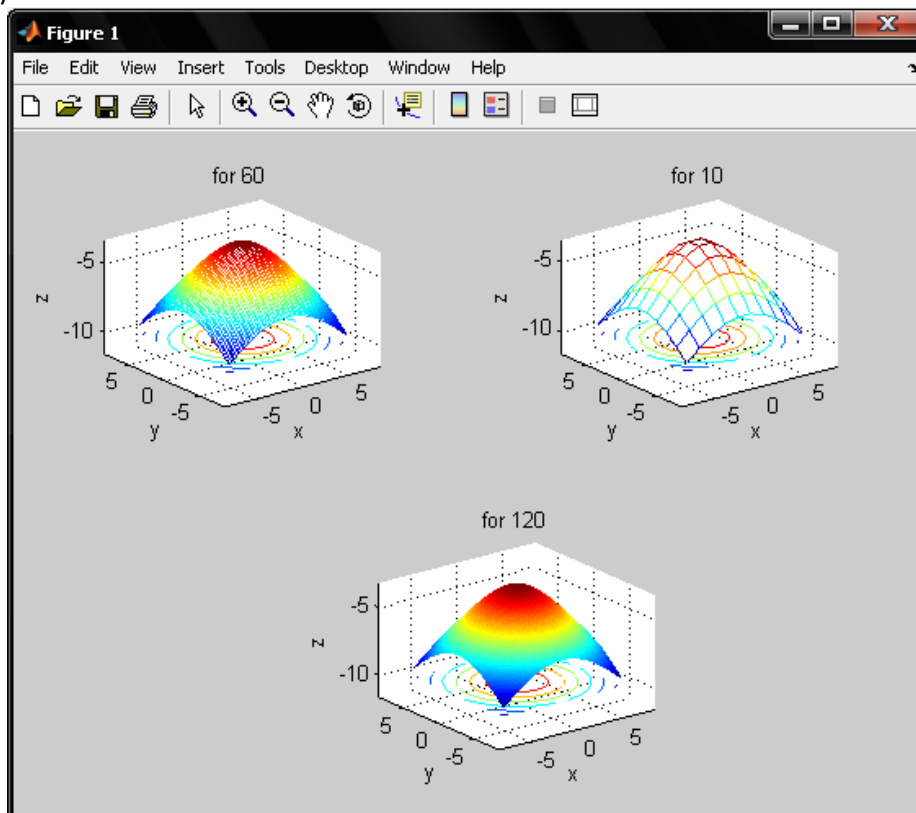
```
>> syms x y
>> z=sin(x)^2*cos(y)^2;
>> ezsurf(x,y,z,[-3,3,-4,4])
>> colormap hsv
>> box on
>> shading flat
```

آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)



در مثال بعد می خواهیم استفاده از تابع `ezmeshc` و تأثیر تغییر عدد پیش فرض 60 را مشاهده کنیم. توجه کنید:

```
>> syms t s
>> axes('position',[.1,.6,.3,.3])
>> ezmeshc(t,s,-(t^2+s^2+9)^0.5)
>> title('for 60')
>> axes('position',[.6,.6,.3,.3])
>> ezmeshc(t,s,-(t^2+s^2+9)^0.5,10)
>> title('for 10')
>> axes('position',[.4,.1,.3,.3])
>> ezmeshc(t,s,-(t^2+s^2+9)^0.5,120)
>> title('for 120')
```



آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

در دستورات و نمودارهای فوق چند نکته ی آموزشی مهم وجود دارد:

۱- تأثیر دادن اعداد طبیعی گوناگون به تابع `ezmeshc` و توابع مشابه را ملاحظه می کنید که چگونه باعث تغییر جزئیات رویه می گردد. همانطور که قبلاً هم گفتیم اگر این عدد طبیعی را وارد نکنید به صورت پیش فرض نمودار از کنار هم قرار گرفتن 60 ردیف تشکیل می گردد و هر عددی که وارد کنید این تعداد تغییر می کند. در این مثال می توانید سه نمودار رسم شده به ازای مقدار پیش فرض 60 و مقادیر انتخابی ما که 10 و 120 هستند را مقایسه نمائید.

۲- حلقه های رنگی روی صفحه ی پایینی، حاصل اضافه شدن حرف `c` به آخر دستور `ezmesh` و تبدیل آن به دستور `ezmeshc` است و نتیجه ای مشابه با دستور `meshc` در بخش نمودارهای عددی دارد.

۳- همانطور که می بینید هر نمودار روی یک محور جداگانه رسم شده است و این به دلیل اجرای دستور `axes` قبل از اجرای دستور رسم هر نمودار می باشد. نوشتن عبارت '`position`' و آرایه ی چهار درایه ای بعد از آن در پرانتز جلوی این تابع، موقعیت قرار گیری هر محور در صفحه را تعیین می کند. درایه ی اول فاصله ی محور از سمت چپ، درایه ی دوم فاصله ی محور از پایین، درایه ی سوم پهنای محور و درایه ی چهارم ارتفاع محور در صفحه را مشخص می کند. این اعداد باید کسری از واحد باشند و به گونه ای انتخاب شوند که محور ها روی هم و یا خارج از کادر پنجره ی `figures` قرار نگیرند. دقت کنید که دستور `axes` را با `axis` اشتباه نگیرید. `axes` برای ایجاد یک محور جدید در مکان دلخواه از صفحه استفاده شده و همواره قبل از دستور رسم نمودار می آید، در حالی که `axis` پس از دستور رسم نمودار آمده و محور نمودار رسم شده را به فرم دلخواه ما که می تواند برگرفته از مشخصات خود نمودار باشد، تغییر می دهد.

۴- نوشتن دستور `title` پس از دستور رسم یک نمودار و با نوشتن یک عبارت دلخواه به عنوان ورودی آن برای نامگذاری نمودار به کار می رود تا بتوان آن نمودار را میان سایر نمودار ها مشخص کرد. اگر شما خودتان نمودار را با استفاده از دستور `title` نامگذاری نکنید، خود نرم افزار مقدار پارامتری محورها را به عنوان نام نمودار رسم شده انتخاب می کند که ممکن است چندان مفید نباشد. به عنوان مثال در نمونه ی فوق اگر خودمان نمودارهایمان را نامگذاری نمی کردیم، نام هر سه نمودار به خاطر یکسان بودن توابع و متغیرهای پارامتری ورودیشان یکسان و عبارت بود از $x = t, y = s, z = -(t^2 + s^2 + 9)^{1/2}$ و به این ترتیب مشخص نبود که هر نمودار مربوط به کدام دستور رسم نمودار است.

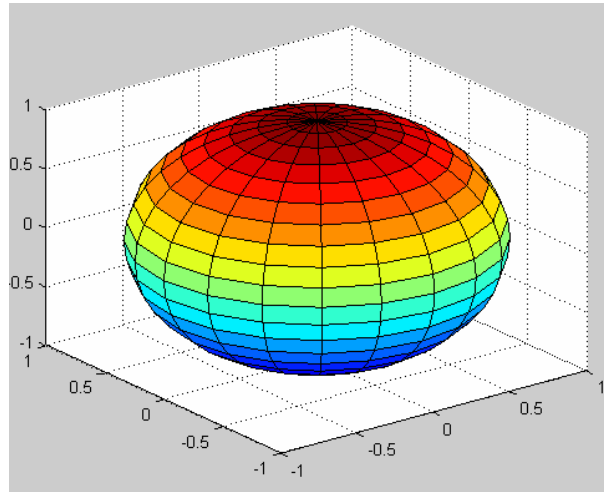
ب-۳) نمودارهای خاص: در MATLAB دستوراتی وجود دارد که موجب رسم اشکال خاصی می گردد. این اشکال عبارتند از کره و استوانه.

کره: برای رسم کره از تابع `sphere` استفاده می کنیم. به صورت پیش فرض تابع `sphere` یک کره متشکل از 20 بند تولید می کند. اما شما می توانید خودتان تعداد بند دلخواهتان را به تابع داده و شکل را به کره ی کامل نزدیکتر کنید و یا آنرا به یک چند وجهی مشخص تبدیل نمایید. همچنین شما می توانید دستور را با یک آرایه ی سه درایه ای از متغیرهای مورد نظر خود مساوی قرار داده و سپس با استفاده از این متغیرها، کره را با هر کدام از دستوراتی که از سری دستورات رسم رویه

آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

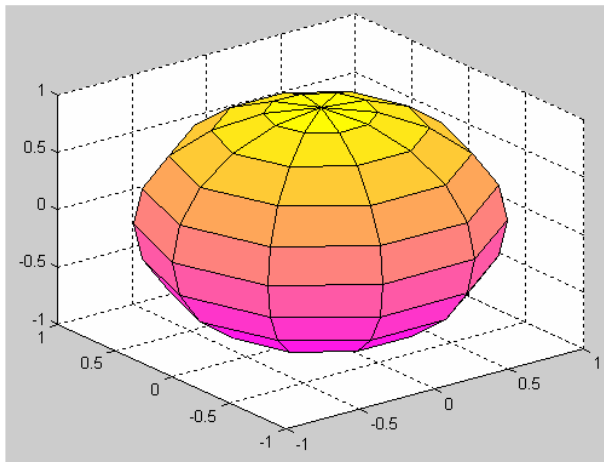
می شناسید رسم کنید و یا حتی از این متغیرها برای رسم شکلهای متفاوت بهره بگیرید. دستور sphere به تنهایی برای رسم یک کره از نوع رویه های surf طراحی شده است. به مثالهای زیر توجه کنید:

```
>> sphere
```



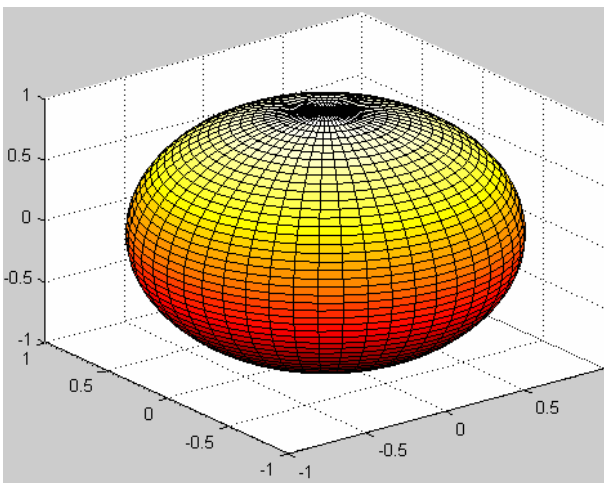
```
>> sphere(10)
```

```
>> colormap spring
```

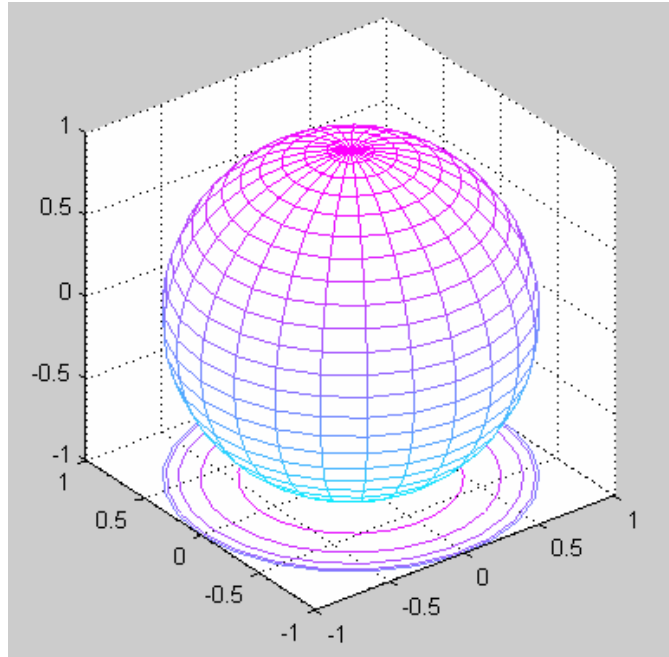


```
>> sphere(50)
```

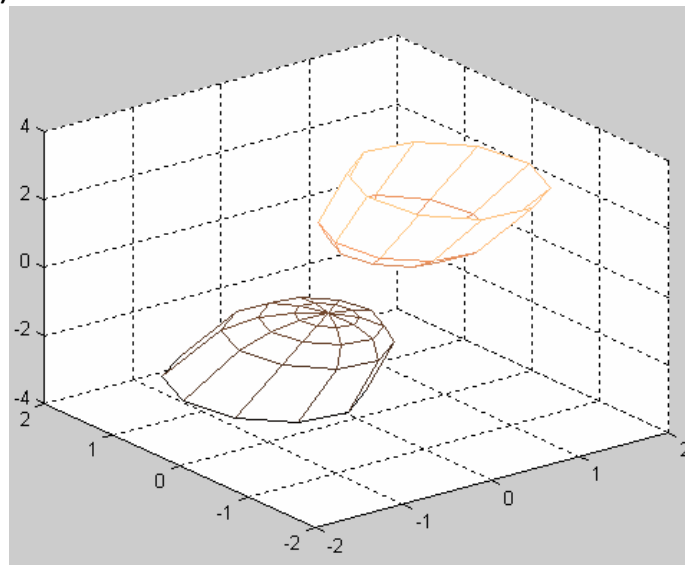
```
>> colormap hot
```



```
>> [x,y,z]=sphere(25);
>> meshc(x,y,z)
>> axis square
>> colormap cool
```



```
>> [x,y,z]=sphere(10);
>> h=mesh(x,y,1./z);
>> colormap copper
>> rotate(h,[1,1,0],15)
```

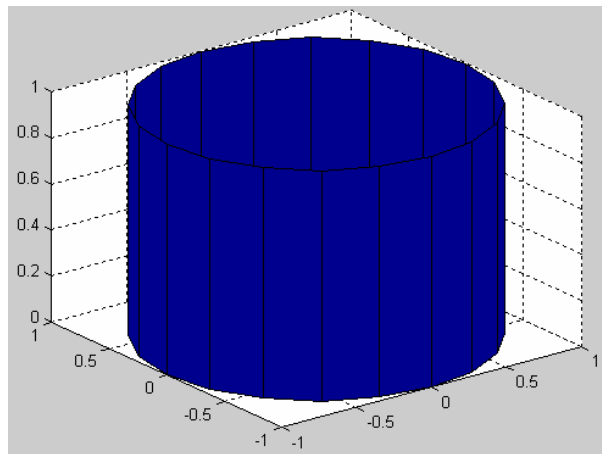


بد نیست در اینجا علاوه بر اینکه رسم یک نمودار متفاوت با استفاده از متغیرهای یک کره را می بینید، شما را با تابع `rotate` نیز آشنا کنم. شما می توانید به نموداری که می خواهید رسم کنید یک نام اختصاص دهید و با دادن این نام به تابع `rotate` نمودارتان را حول خطی در امتداد بردار مورد نظرتان به اندازه ی درجه ی دلخواه بچرخانید. در مثال فوق نمودارمان را `h` نامیدیم و حول امتداد بردار `[1,1,0]` به اندازه ی 15 درجه منحرف ساختیم.

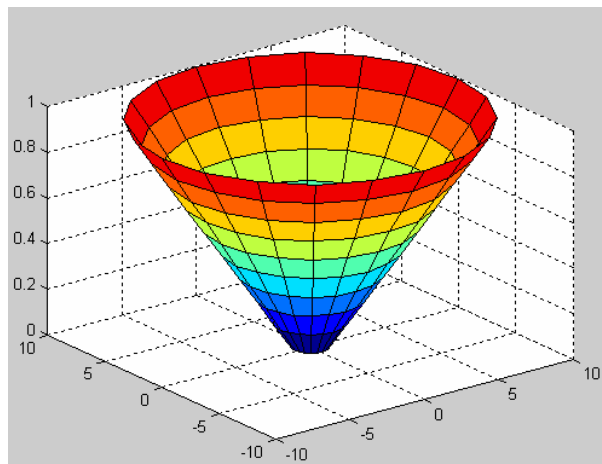
آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

استوانه: در مورد استوانه دستور `cylinder` استفاده می شود که به تنهایی یک رویه ی استوانه ای با شعاع یک و 20 وجه و ارتفاع 1 بوجود می آورد. اگر شما یک ورودی به تابع دهید با آن به عنوان شعاع استوانه برخورد می کند که می تواند ثابت یا متغیر باشد. اگر دو ورودی دهید، ورودی اول را شعاع و ورودی دوم را تعداد وجوه در نظر می گیرد. همچنین می توانید استوانه را مساوی با یک آرایه ی سه درایه ای از نام متغیرهای دلخواه قرار داده و با استفاده از مقادیر ذخیره شده در متغیرها نمودارهای گوناگونی رسم کنید. مثالهای زیر دربرگیرنده ی موارد فوق می باشند:

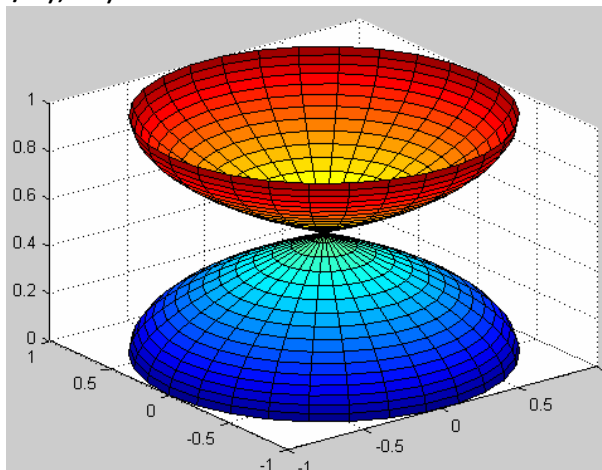
```
>> cylinder
```



```
>> cylinder(1:10)
```

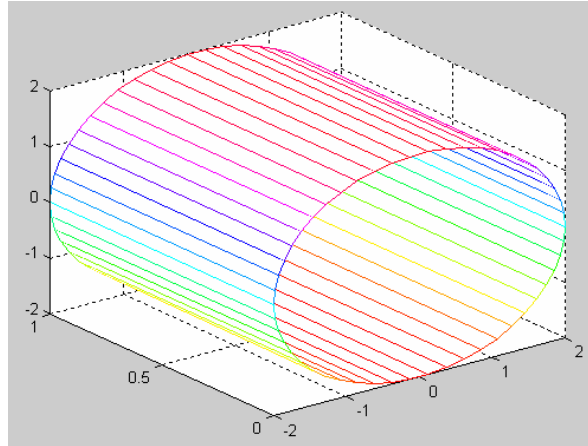


```
>> cylinder(sin(-pi/2:.1:pi/2),30)
```

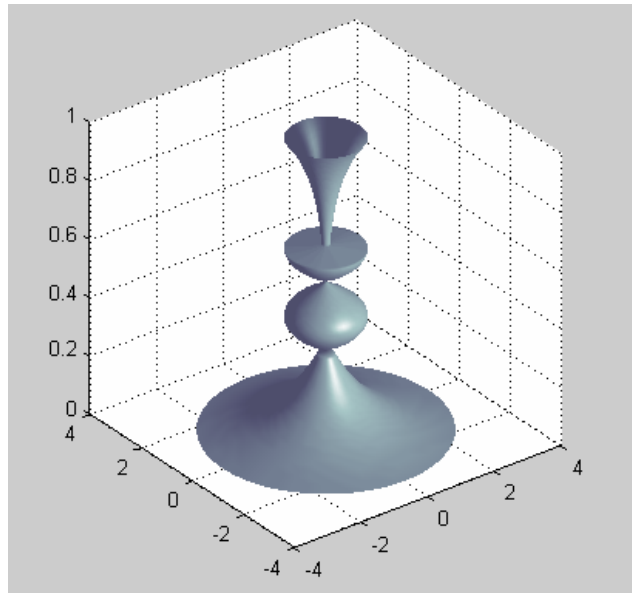


آموزش نرم افزار MATLAB (آشنایی با نرم افزار و کاربرد آن در ریاضیات)

```
>> [x,y,z]=cylinder(2,50);
>> mesh(x,z,y)
>> colormap hsv
```



```
>> t=[tan(2*pi/5:-0.03:0.2) sin(-pi:1:pi/2) logspace(-1,0,50)];
>> [x,y,z]=cylinder(t,50);
>> surf(x,y,z)
>> shading interp
>> colormap bone
>> axis square
```



می بینید که با دادن شعاع متغیر به استوانه می توان اشکال زیبایی آفرید. توجه کنید که خود نرم افزار طوری دایره های حاصل از مقادیر متغیر اندازه ی شعاع را در کنار هم قرار می دهد که اولین شعاع در ارتفاع 0 و آخرین شعاع در ارتفاع 1 قرار گیرد، و بقیه ی حلقه ها با فواصل یکسان این بین قرار می گیرند.