

فصل دوازدهم: نمودارهای سه بعدی



MATLAB®

کلاس آموزشی

www.MatlabKar.com

فصل دوازدهم: نمودارهای سه‌بعدی

۱۱-۱- خمهای فضایی - تابع `plot3`

با استفاده از تابع `plot3` در متلب می‌توان یک منحنی را در فضای سه‌بعدی ترسیم کرد. روش استفاده از این تابع بسیار شبیه تابع `plot` است. جز اینکه برای هر منحنی به سه بردار هم طول نیاز است.

مثال: رسم یک فنر با شعاع برابر با یک:

$$x=t$$

$$y= \sin(t)$$

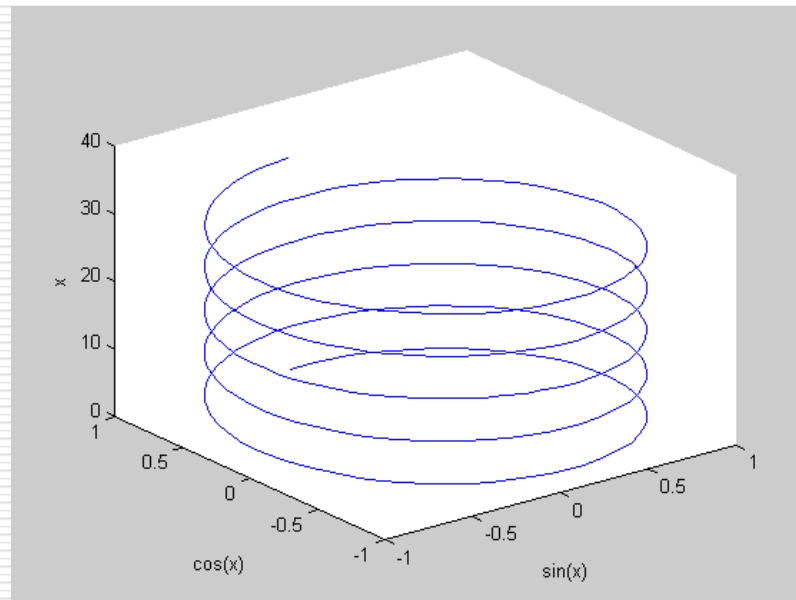
$$z=\cos(t)$$

$$t \in R$$

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۱- خمهای فضایی-ادامه

```
>>t=0: pi/50:10*pi;  
>>plot3(sin(t) , cos(t) , t); xlabel('sin(x)'); ylabel('cos(x)'); zlabel('x')
```



فصل دوازدهم: نمودارهای سه‌بعدی

۱۱-۲- تغییر زاویه دید

با استفاده از دکمه فشاری **Rotate 3D** بر روی هر پنجره شکل متلب و یا با استفاده از دستور **view** می‌توان زاویه دید را تغییر داد. همچنین در منوی **Tools** با استفاده از زیرمنوهای **Camera Motion** می‌توان در فضای سه‌بعدی حرکت کرد.

شکل کلی استفاده از دستور **view** بصورت زیر است:

view([az , el])

که در این رابطه **az** و **el** به ترتیب زاویه دوربین نسبت به صفحه **XY** و بخش منفی محور **Y** است.

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۳- نمودارهای شبکه‌ای: توابع `mesh`, `meshc`, `meshz`

با استفاده از این توابع می‌توان سطوح شبکه‌ای (یا توری) ایجاد کرد. شکل کلی استفاده از تابع `mesh` بصورت زیر است:

`mesh(x,y,z)`

که در این رابطه Z تابعی دو متغیره از متغیرهای X و Y می‌باشد. بنابراین لازم است که Z یک ماتریس دو بعدی باشد که تعداد سطرهای آن برابر با تعداد عناصر Y و تعداد ستونهایش برابر با تعداد عناصر X باشد. X و Y باید بردار باشند اگرچه می‌توانند ماتریس‌هایی هم‌بعد نیز باشند بدینصورت که بردار X به تعداد عناصر بردار Y بصورت سطری تکرار شود و بردار Y به تعداد عناصر X بصورت ستونی تکرار گردد. که در اینصورت دو ماتریس هم‌بعد خواهیم داشت. تابع `meshgrid` می‌تواند این عمل را انجام دهد:

`[x_new,y_new]=meshgrid(x,y);`

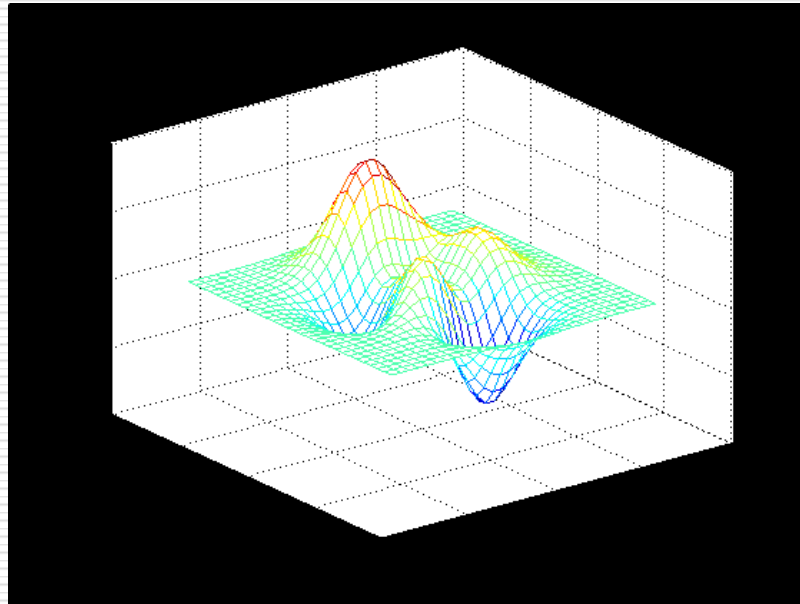
تابع `meshc` علاوه بر نمودار شبکه‌ای، نمودارهای تراز را نیز رسم می‌کند.
تابع `meshz` دیواره‌هایی را در پایین نمودار به سمت صفحه $X-Y$ رسم می‌کند.

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۳- نمودارهای شبکه‌ای - ادامه -

مثال ۱ : تابع **peaks** یکی از توابع متلب است که یک مدل ریاضی از پیش تعریف شده را ایجاد می‌کند:

```
>> [x ,y,z]= peaks(30);  
>> mesh(x,y,z);
```

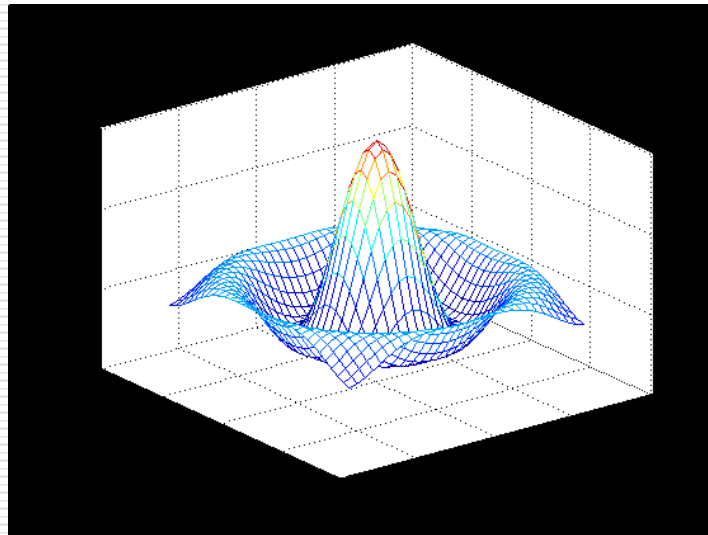


فصل دوازدهم: نمودارهای سه بعدی

۱۱-۳- نمودارهای شبکه‌ای - ادامه -

مثال ۲: رسم یک تابع دو بعدی به فرمول $z = \sin(r)/r$ که $r = \sqrt{X^2 + Y^2}$

```
>> x=-7.5: 0.5: 7.5; y=x;  
>> [x_new,y_new]=meshgrid(x,y);  
>> r = sqrt(x_new.^2 + y_new.^2) + eps;  
>> z=sin(r) ./ r; mesh(x_new , y_new , z) یا mesh(x,y,z);
```



فصل دوازدهم: نمودارهای سه بعدی

۱۱-۳- نمودارهای شبکه‌ای - ادامه -

به صورت پیش فرض متلب نقاطی از نمودار سه بعدی که پشت نقاط جلویی قرار گرفته‌اند را مخفی می‌کند. با استفاده از تابع **hidden** می‌توان این رفتار را تغییر داد:

>> hidden off

>> hidden on

>> hidden

فصل دوازدهم: نمودارهای سه‌بعدی

۱۱-۴- نمودارهای رویه: surf و surfl و surfc

تفاوت نمودارهای رویه با نمودارهای شبکه‌ای آن است که در اینجا بجای خطوط شبکه‌ای، وصله‌ها یا فواصل بین شبکه‌ها ترسیم می‌گردد.

تابع **surf** دقیقاً مانند تابع **mesh** بکار برده می‌شود.

تابع **surfl** علاوه بر تواناییهای تابع **surf** می‌تواند خصوصیات نوردهی شیء سه‌بعدی را نیز تنظیم کند. از قبیل جهت منبع نوری و خصوصیات انعکاسی شیء

تابع **surfc** مانند تابع **meshc** عمل می‌کند

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۴- نمودارهای رویه-ادامه

تابع shading :

با استفاده از تابع shading می توان نوع سایه رنگ نمودار را تعیین کرد. این تابع باید پس از یکی از توابع surf بیاید و با یکی از سه پارامتر زیر قابل فراخوانی است:

shading flat

shading interp

shading faceted

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۴- نمودارهای رویه-ادامه

تابع colormap :

با استفاده از این تابع می توان رنگهای بکار برده شده برای ترسیم نمودار رویه را تعریف کرد تنها پارامتر مورد نیاز می تواند یکی از ثابتهای زیر باشد:

hsv, cool , hot, prism, gray

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۵- چرخش دوربین در فضای سه بعدی

با استفاده از تابع `camorbit` می توان پس از رسم نمودار سه بعدی، زاویه دید دوربین نسبت به شیء را تغییر داد:

`camorbit(theta,phi)`

`theta`: زاویه چرخش افقی

`phi`: زاویه چرخش عمودی

و یا می توانید از این فرم تابع استفاده کنید:

`camorbit(theta,phi,'محور چرخش');`

آرگومان سوم محوری است که چرخش حول آن انجام می شود را مشخص می کند که به صورت پیش فرض محور Z است.

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۵- چرخش دوربین در فضای سه بعدی-ادامه...

مثال:

```
surf(peaks)
axis vis3d
axis off
for i=1:360
    camorbit(2,0,'data')
    drawnow
end
```

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۶- تعیین موقعیت دوربین

به منظور تعیین موقعیت دوربین می توان از تابع `campos` استفاده کرد:
آرگومان ورودی این تابع موقعیت دوربین را در فضای `campos([x,y,z])` سه بعدی مشخص می کند.

مثال:

```
surf(peaks)
axis vis3d off
for x = -200:5:200
    campos([x,5,10])
    drawnow
end
```

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۷- نمایش میله رنگ

با استفاده از تابع `colorbar` می توان پس از نمایش نمودار، رنگهای بکار برده شده در آن را بصورت یک میله رنگ در کنار نمودار نمایش داد:

```
>> surf(peaks)
```

```
>> colorbar
```

اعداد نمایش داده شده در کنار میله رنگ مقداری را که هر رنگ به آن اشاره می کند(به صورت پیش فرض مقدار Z) را نمایش می دهد.

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۸- نمودارهای تراز- contour3 , contour

نمودارهای تراز خطوط و منحنی‌های بسته‌ای می‌باشند که برای نمایش ارتفاع‌های مختلف یک نمودار سه بعدی، در فضای دوبعدی یا سه بعدی بکار برده می‌شوند. در این نمودارها، نواحی هم‌ارتفاع توسط خطوطی به یکدیگر متصل می‌شوند.

نحوه استفاده از توابع contour و contour3 نیز دقیقاً شبیه تابع mesh است.

مثال:

```
[c,h] = contour(peaks); clabel(c,h), colorbar
```


فصل دوازدهم: نمودارهای سه‌بعدی

۱۱-۹- تابع pcolor

این تابع داده‌های مربوط به یک نمودار سه‌بعدی را در فضای دوبعدی رسم می‌کند و برای نمایش ارتفاع (مقدار Z) از تغییرات رنگ استفاده می‌نماید. نتیجه حاصل از این تابع، مشابه تصویر از بالای تابع surf است.

مثال ۱:

```
>> pcolor(peaks)
```

مثال ۲:

```
t=tril(ones(10));  
ts=[t,flipr(t)];  
tss=[flipud(ts);ts];  
pcolor(tss); colormap(gray(2))
```

فصل دوازدهم: نمودارهای سه بعدی

۱۱-۹- اجرای برنامه‌های نمایشی متلب در زمینه ترسیم سه بعدی

demo های Graphics در متلب راهنمای بسیار خوبی برای آموزش قابلیت‌های گرافیکی متلب می‌باشد. این برنامه‌ها از منوی **Start** متلب در شاخه **Demo-> Graphics** در دسترسند.