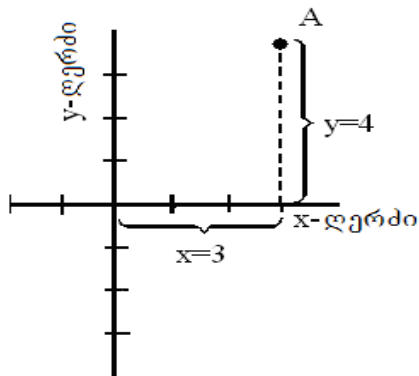


MATLAB-ის გრაფიკული შესაძლებლობები

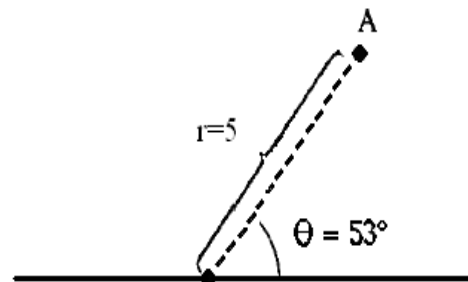
გრაფიკის აგება საშუალებას გვაძლევს შევამოწმოთ და გავანალიზოთ მონაცემები. MATLAB-ს გააჩნია მძლავრი გრაფიკული საშუალებები.

კოორდინატთა მართკუთხა და პოლარული სისტემა

სხვადასხვა ამოცანებში ვიყენებთ მონაცემებს რომელთა წარმოდგენა შესაძლებელია როგორც კოორდინატთა მართკუთხა, ისე პოლარულ სისტემაში. კოორდინატთა მართკუთხა სისტემაში წერტილის ადგილი განისაზღვრება მანძილით სათავიდან ჰორიზონტალური და ვერტიკალური ღერძების გასწვრივ ნახ. 1. პოლარულ კოორდინატთა სისტემაში გრაფიკის ასაგებად გვჭირდება პოლარული მონაცემები – მანძილი კოორდინატთა სათავიდან და კუთხე ნახ. 2.



ნახ. 1. კოორდინატთა მართკუთხა - დეკარტის სისტემა



ნახ. 2. კოორდინატთა პოლარული სისტემა

როგორც უკვე ვიცით დეკარტის მართკუთხა კოორდინატთა სისტემაში გრაფიკების აგება შეიძლება plot ბრძანებით.

ზოგჯერ უფრო მოსახერხებელია მონაცემების პოლარული კოორდინატებში გამოსახვა.

brZaneba polar

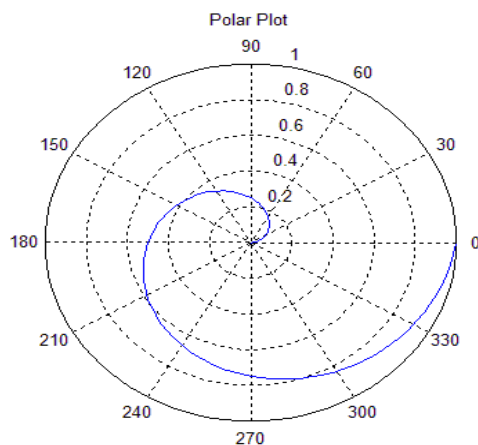
MATLAB-ში პოლარული გრაფიკი აიგება ბრძანებით **polar**.

polar(theta,r) ეს ბრძანება ააგებს პოლარულ გრაფიკს θ კუთხისა და შესაბამისი მოდულისათვის-r

თუ რომელიმე არგუმენტოვანი მატრიცაა, მაშინ გრაფიკი აიგება მატრიცის სვეტების მიხედვით, ერთდროულად მივიღებთ იმდენ მრუდს, რამდენი სვეტიც შედის მატრიცაში. თუ ორივე არგუმენტი ერთნაირი ზომის მატრიცაა, ერთი მათგანის სვეტები აიგება მეორის შესაბამისი სვეტების მიმართ.

დავუშვათ გვინდა ავაგოთ ზრდადი რადიუსიანი წრეწირის ანუ სპირალის გრაფიკი. შევქმნით θ ვექტორს, რომლის ელემენტებია რადიანებში გამოსახული კუთხე 0 დან 2π -მდე და r ვექტორს, რომლის ელემენტები იქნება რადიუსის შესაბამისი მნიშვნელობები 0 დან 1-მდე. 0 შეიცავს გრაფიკს, რომელიც აიგება ბრძანებათა შემდეგი მწკრივით:

```
theta = 0:2*pi/100:2*pi;  
r = theta/(2*pi);  
polar(theta,r),...  
title('Polar Plot'),...  
grid
```



ნახ. 3. ზრდადი რადიუსიანი წრის პოლარული გრაფიკი

ხშირად საჭიროა მონაცემები კოორდინატთა ერთი სისტემიდან მეორეში გადავიყვანოთ. ასეთი ტრანსფორმაცია გამოისახება ფორმულებით:
პოლარული კოორდინატების გადაყვანა მართკუთხა სისტემაში:

$$x = r \sin \theta$$

$$y = r \cos \theta$$

მართკუთხა კოორდინატების გადაყვანა პოლარულ სისტემაში:

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

ტანგენსის შებრუნებული სიდიდის გამოთვლისას სიფრთხილეა საჭირო, რომ სწორად შევარჩიოთ კუთხის მნიშვნელობა. ამ მიზნით MATLAB-ში რეკომენდირებულია **atan2** ფუნქციის გამოყენება.

საგარჯიშო

გადაიყვანეთ მოცემული წერტილები კოორდინატთა მართკუთხადან პოლარულ სისტემაში:

1. (3, -2)
2. (0, -1)
3. (-2, 0)
4. (0.5, 1)

გადაიყვანეთ შემდეგი წერტილები პოლარულიდან მართკუთხა სისტემაში:

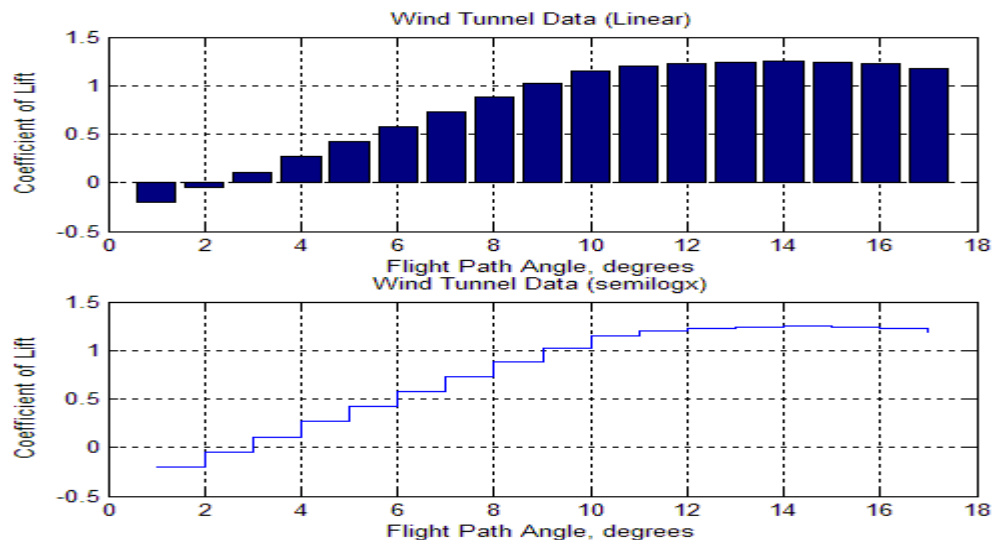
5. (π , 1)
6. ($\pi/2$, 0)
7. (2.3, 0.5)
8. (0.5, 0.5)

bar და stairs გრაფიკები

bar(y)	ააგებს bar გრაფიკს y მნიშვნელობათათვის რომელიც წარმოადგენს გაფერადებულ მართკუთხედთა ერთობლიობას (ნახ.4), x ღერძზე გადაიზომება y ვექტორის ინდექსები და ისინი მიუთითებენ თითოეული მართკუთხედის ცენტრს, y ღერძზე კი მისი მნიშვნელობები და ისინი წარმოადგენენ შესაბამისი მართკუთხედების სიმაღლეს
bar(x,y)	ააგებს bar გრაფიკს მოცემულ x და y მონაცემებისთვის
stairs(y)	ააგებს საფეხუროვან stair გრაფიკს y ვექტორის მნიშვნელობათათვის.
stairs(x,y)	ააგებს საფეხუროვან stair გრაფიკს x და y მონაცემებისთვის

0 წარმოადგენს **bar** და **stairs** გრაფიკებს რომლებიც აგებულია შემდეგი ბრძანებებით:

```
subplot(2,1,1), bar(y)
title('Wind Tunnel Data (Linear)'),...
xlabel('Flight Path Angle, degrees'),...
ylabel('Coefficient of Lift'),...
grid
subplot(2,1,2), stairs(y)
title('Wind Tunnel Data (semilogx)'),...
xlabel('Flight Path Angle, degrees'),...
ylabel('Coefficient of Lift'),...
grid
```



ნახ. 4 bar და stairs გრაფიკები

გრაფიკული ოფციები

MATLAB შეიცავს სხვადასხვა ოფციებს გრაფიკის გასაფორმებლად და გასაუმჯობესებლად. განვიხილოთ ზოგიერთი მათგანი, რომელიც შემდგომში სშირად გამოგვადგება.

რამდენიმე მრუდის აგება ერთიდაიგივე ნახაზზე

არსებობს ერთიდაიგივე ნახაზზე რამდენიმე მრუდის აგების სამი გზა:

- **plot** ბრძანების არგუმენტად ავიღოთ მატრიცები, მაშინ აიგება იმდენი გრაფიკი რამდენი სვეტიც აქვს მატრიცს. თუ `plot(x,y)` ბრძანებაში `x` და `y` ერთიდაიგივე ზომის მატრიცებია აიგება იმდენი გრაფიკი რამდენი სვეტიცაა ამ მატრიცებში და თითოეული გრაფიკი იქნება `x`-სა და `y`-ის შესაბამისი სვეტებით აგებული გრაფიკი;
- `plot` ბრძანებას მივცეთ რამდენიმე არგუმენტი, მაგალითად: `plot(x1,y1,x2,y2)` სადაც `x1`, `y1`, `x2`, `y2` ვექტორებია, MATLAB ააგებს ორივე `x1`, `y1` და `x2`, `y2` გრაფიკს ერთ ნახაზზე (ერთ გრაფიკულ ფანჯარაში). ამ მეთოდის უპირატესობა ისაა, რომ საჭირო არაა წერტილების რაოდენობა ერთმანეთს ემთხვეოდეს. MATLAB ავტომატურად შეარჩევს მათთვის წირის განსხვავებულ ფერს. შესაძლებელია ორივე გრაფიკისთვის თქვენთვის სასურველი ფორმისა და ფერის მიცემაც ბრძანებით `plot(x1,y1,' ',x2,y2,' ')` სადაც აპოსტროფებში ჩაწერთ თქვენთვის სასურველ მახასიათებლებს;
- ერთიდაიგივე ნახაზზე სხვადასხვა მონაცემთა აგება ხორციელდება ბრძანებით **hold** ან **hold on**. ეს ბრძანებები ინარჩუნებენ გრაფიკულ ფანჯარას აქტიურ მდგომარეობაში და ყოველი შემდგომი გრაფიკი ემატება ადრე აგებულ გრაფიკს

შენახულ გრაფიკულ ფანჯარაში, **hold** ბრძანების ხელშეწყობით მიცემა ან **hold off** ბრძანება შეწყვეტს ამ პროცესს.

დერძებზე წრფივი და ლოგარითმული სკალის მოცემა

საზოგადოდ როცა `plot` ბრძანებით ვაგებთ გრაფიკს, იგულისხმება, რომ x და y დერძები დაყოფილია თანატოლ ინტრვალებად. ასეთ გრაფიკულ წარმოდგენას წრფივს უწოდებენ. ზოგჯერ შეიძლება საჭირო იყოს ლოგარითმული სკალის მოცემა ერთ-ერთ ან ორივე დერძზე. 10 ფუძიანი ლოგარითმული სკალა მოხერხებულია, როცა ცვლადის მნიშვნელობები ძლიერ ფართო დიაპაზონში იცვლება.

`semilogx(x,y)` აგებს გრაფიკს ლოგარითმული სკალით x -ზე და წრფივით y -ზე.

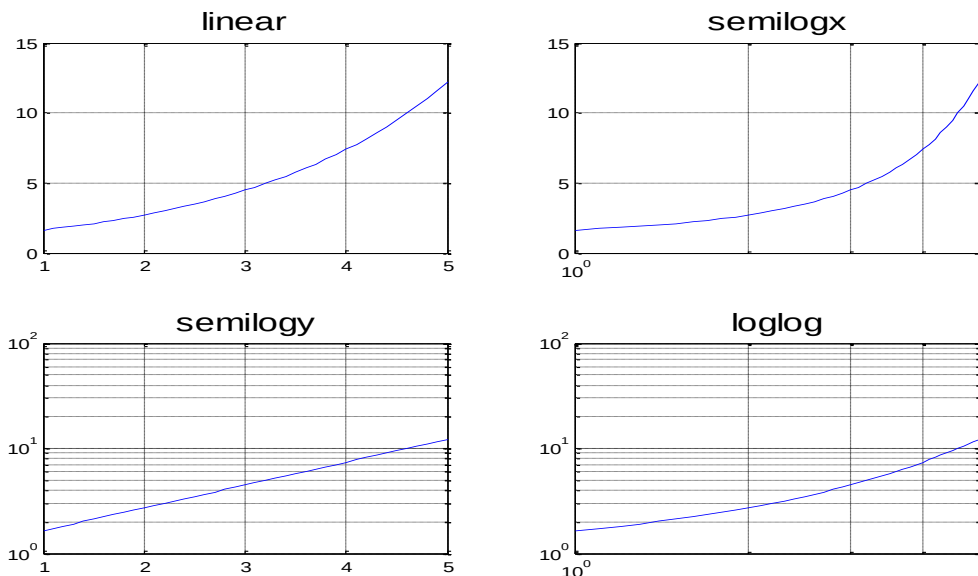
`semilogy(y)` აგებს გრაფიკს ლოგარითმული სკალით y -დერძზე და წრფივით x -ზე.

`loglog(x,y)` აგებს გრაფიკს ლოგარითმული სკალით ორივე დერძზე

მნიშვნელოვანია გავიხსენოთ, რომ ნულის ტოლი ან ნულზე ნაკლები სიდიდის ლოგარითმი არ არსებობს, ამიტომ თუ მონაცემები, რომლებიც უნდა ავაგოთ **semilog** ან **loglog** ბრძანებით, შეიცავს უარყოფით ან ნულის ტოლ მნიშვნელობებს, MATLAB მიგვითითებს შეცდომაზე და გვამცნობს, რომ ეს მნიშვნელობები გამოტოვებული იქნება გრაფიკის აგების დროს.

0 5-ზე წარმოდგენილია წრფივი, `semilogx`, `semilogy` და `loglog` ბრძანებებით აგებული გრაფიკები:

```
x=1:0.1:5; y=exp(x/2);
figure
subplot(2,2,1)
plot(x,y), grid, title('linear','FontSize',17)
subplot(2,2,2)
semilogx(x,y), grid, title('semilogx','FontSize',17)
subplot(2,2,3)
semilogy(x,y), grid, title('semilogy','FontSize',17)
subplot(2,2,4)
loglog(x,y), grid, title('loglog','FontSize',17)
```



ნახ. 5. წრფივი და ლოგარითმულ სკალაში აგებული გრაფიკები

სავარჯიშო

SeadgineT x veqtori 0 dan 20-mde bijiT 0,5 da gamoiTvaleT Sesabamisi y rogorc x-is funqcia: $y = 5x^2$

1. ააგეთ ამ მონაცემთა წრფივი გრაფიკი
2. ააგეთ ამ მონაცემთა გრაფიკი ლოგარითმული სკალით x-ის მიმართ
3. ააგეთ ამ მონაცემთა გრაფიკი ლოგარითმული სკალით y-ის მიმართ
4. ააგეთ ამ მონაცემთა გრაფიკი ლოგარითმული სკალით ორივე ღერძის მიმართ
5. SeadareT grafikebi erTmaneTs, aRwereT TiToeulis upiratesoba da nakli

გრაფიკული გამოსახულების საზღვრების მოცემა

MATLAB ავტომატურად ირჩევს ღერძების სკალას. მაგრამ შესაძლებელია მიუთითოთ ღერძების საზღვრები ბრძანებით **axis**:

axis	ეს ბრძანება გრაფიკულ ფანჯარაზე 'გაყინავს' ღერძების არსებულ სკალას, ავტომატურ რეჟიმში მის დასაბრუნებლად საჭიროა იგივე ბრძანებას გამეორება
axis(v)	სადაც v Semdegi saxis ოთხ ელემენტიანი ვექტორია [xmin, xmax, ymin, ymax], წარმოგიდგენს მოცემული გრაფიკის იმ ნაწილს რომელიც მოქცეულია x RerZze [xmin, xmax] საზღვრებში და y ღერძზე [ymin, ymax] საზღვრებში
axis('square')	წარმოადგენს გრაფიკს კვადრატულ 'ჩარჩოში', სადაც x-სა და y-ის ცვლილების რეალური მნიშვნელობების მიუხედავად ისინი ერთნაირი ზომისად იქნება წარმოდგენილი
axis('normal')	აღადგენს ღერძების ნორმალურ თანაფარდობას

ბრძანება subplot

ბრძანება **subplot** საშუალებას გვაძლევს გრაფიკული ფანჯარა დავეოთ რამდენიმე ქვეფანჯარად. 0 და 0 შექმნილია ამ ბრძანების გამოყენებით. **subplot** გააჩნია 3 არგუმენტი subplot (m,n,p), სადაც m,n განსაზღვრავს რამდენ ქვეფანჯარადაა დაყოფილი გრაფიკული ფანჯარა და რა რიგით, ხოლო p მიუთითებს რომელ ფანჯარაში აიგოს მოცემული გრაფიკი. ფანჯრები დანომრილია მარცხნიდან მარჯვნივ და ზემოდან ქვემოთ.

ეკრანის კონტროლი

როგორც ვიცით MATLAB-ში ინფორმაცია შეიძლება წარმოვადგინოთ ორ ძირითად ფანჯარაში ბრძანებათა და გრაფიკულ ფანჯარაში. შემდეგი ბრძანებები საშუალებას გვაძლევს შევარჩიოთ და გავასუფთაოთ გრაფიკული ფანჯარა:

shg	ეკრანზე გამოიყვანს გრაფიკულ ფანჯარას
ნებისმიერი კლავიში	დააბრუნებს ეკრანზე ბრძანებათა ფანჯარას
clf	ასუფთავებს გრაფიკულ ფანჯარას

გრაფიკის გაფორმება ტექსტით

adre Cven ukve gavecaniT Tu rogor SeiZleba warwerebis keTeba RerZebisa da grafikis Tavze (brZanebebi: xlabel, ylabel, title). grafikze sxvadxva saxis teqstis datana SeiZleba Semdegi brZanebebis gamoyenebiTac

text(x,y,'text')	ბრჭყალებში ჩაწერილ ტექსტი წაეწერება გრაფიკის წერტილში, რომლის კოორდინატებია x,y. თუ x ,y ვექტორებია, წარწერას გააკეთდება ყოველ წერტილში
text(x,y,'text','sc')	ბრჭყალებში ჩაწერილ ტექსტი წაეწერება გრაფიკის წერტილში, რომლის კოორდინატებია x,y ისე, რომ ქვედა მარცხენა კუთხის კოორდინატები მიიჩნევა (0,0)-ის ტოლად, ზედა მარჯვენასი კი (1,1)-ს ტოლად
gtext('text')	წააწერს ტექსტს გრაფიკზე იმ წერტილში, რომელსაც მაუსით

მივუთითებთ

ბრძანება ginput

ეს ბრძანება საშუალებას გვაძლევს ავიღოთ კოორდინატები პირდაპირ გრაფიკული ფანჯარიდან მაუსის მეშვეობით:

[x,y]=ginput საშუალებას გვაძლევს შევარჩიოთ წერტილთა შეუზღუდავი რაოდენობა გრაფიკული ფანჯარიდან მაუსის საშუალებით. შეიქმნება x, y ვექტორები კოორდინატთა შესაბამისი მნიშვნელობებით. ბრძანების მოქმედება შეწყდება enter კლავიშზე დაჭერისას

[x,y]=ginput(n) საშუალებას გვაძლევს შევარჩიოთ n წერტილი გრაფიკული ფანჯარიდან მაუსის საშუალებით. შეიქმნება x, y ვექტორები კოორდინატთა შესაბამისი მნიშვნელობებით.

სამგანზომილებიანი გრაფიკი

MATLAB-ში სამგანზომილებიანი გრაფიკის აფგების სხვადასხვა გზა არსებობს. შესაძლებელია ავაგოთ ბადურა (mesh) ან ჩვეულებრივი (surf) ზედაპირი სამგანზომილებიან სივრცეში. შეგვიძლია ასეთ ზედაპირს შევხედოთ გრაფიკულად ნებისმიერი მიმართულებიდან, შეგვიძლია განვსაზღვროთ ღერძების სკალა. სამგანზომილებიანი ზედაპირი შეიძლება წარმოვადგინოთ კონტურული გრაფიკების მეშვეობითაც.

ზედაპირის აგება mesh და surf ბრძანებებით

mesh და **surf** ზედაპირი აიგება მატრიცის სახით წარმოდგენილი მონაცემების საფუძველზე. მატრიცის თითოეული ელემენტი შესაბამება წერტილს ზედაპირზე რომლის შესაბამისი კოორდინატებია x – მატრიცის შესაბამისი ელემენტის სტრიქონის ინდექსი, y – მატრიცის შესაბამისი ელემენტის სვეტის ინდექსი, z – მატრიცის შესაბამისი ელემენტის მნიშვნელობა. **mesh**-ით იგება მხოლოდ ბადურა ზედაპირი **surf**-ით კი ჩვეულებრივი ზედაპირი.

სამგანზომილებიანი ზედაპირისათვის მონაცემების შექმნა შეიძლება სხვაგვარადაც: ჯერ შეგვიძლია განვსაზღვრაოთ x და y ცვლადები და შემდეგ გამოვითვალოთ z სიდიდეები, შესაბამისი x, y ცვლადების ფუნქციის სახით. x და y ისე უნდა შევარჩიოთ, რომ მათი მნიშვნელობები თანაბრად იყოს განაწილებული x - y სიბრტყეზე. დავუშვათ გვინდა ავაგოთ ფუნქცია

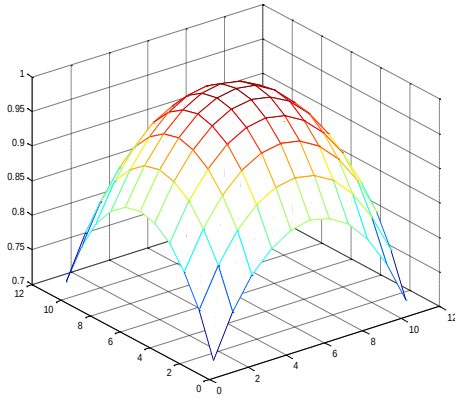
$$f(x, y) = z = \sqrt{1 - x^2 - y^2}$$

ისე, რომ $-0.5 \leq x \leq 0.5$ და $-0.5 \leq y \leq 0.5$. ეს ფუნქცია წარმოადგენს ერთეულოვანი რადიუსის მქონე სფეროს ზედაპირის განტოლების სახეცვლილებას:

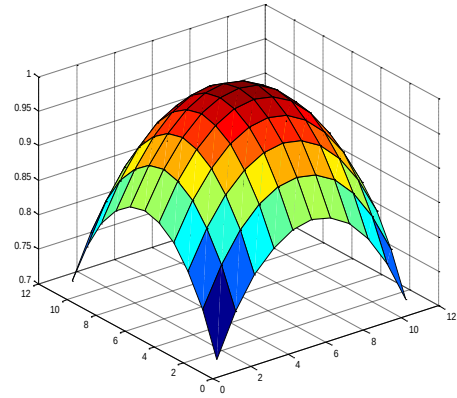
$$x^2 + y^2 + z^2 = 1$$

რადგან $f(x, y)$ იყენებს კვადრატული ფესვის მხოლოდ დადებით მნიშვნელობებს, იგი ამ ფუნქციის მხოლოდ ზედა ნახევარს წარმოადგენს. იმისათვის, რომ ავაგოთ სამგანზომილებიანი ზედაპირი შემდეგნაირად უნდა მოვიქცეთ:

```
n=0;
for x = -0.5:0.1:0.5,
    n=n+1;
    m=0;
    for y = -0.5:0.1:0.5,
        m=m+1;
        z(n,m) = abs(sqrt(1 - x.^2 - y.^2));
    end
end
mesh(z), title('3-D Plot'), grid
```



ნახ. 6ა. სფეროს ნაწილის
სამგანზომილებიანი mesh გრაფიკი



ნახ. 6ბ. სფეროს ნაწილის
სამგანზომილებიანი surf გრაფიკი

ასეთი z ვექტორის აგება უფრო მოხერხებულია meshgrid ბრძანების საშუალებით რომელიც აწარმოებს 2 მატრიცას X -ს და Y -ს x და y ღერძის ჩვენთვის სასურველი მნიშვნელობების მიხედვით. მაგალითად

$[X,Y]=\text{meshgrid}(-2:2,-2:2)$

აწარმოებს შემდეგი სახის ორ X და Y მატრიცას

X	-2	-1	0	1	2
	-2	-1	0	1	2
	-2	-1	0	1	2
	-2	-1	0	1	2
	-2	-1	0	1	2
Y	-2	-2	-2	-2	-2
	-1	-1	-1	-1	-1
	0	0	0	0	0
	1	1	1	1	1
	2	2	2	2	2

მაშინ სფეროს ზემოთაღწერილი ნაწილი ასეც შეგვიძლია ავაგოთ:

$[X,Y]=\text{meshgrid}(-0.5:0.1:0.5,-0.5:0.1:0.5)$

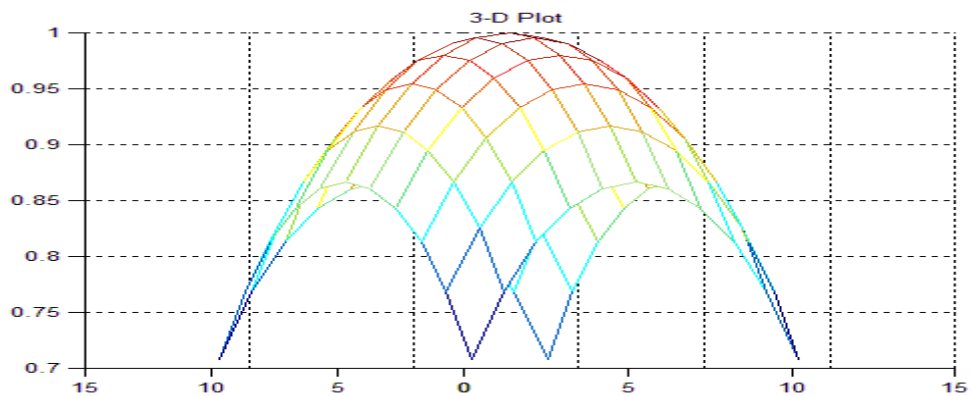
$Z=\text{sqrt}(\text{abs}(1-X.^2-Y.^2));$

$\text{mesh}(Z)$

grid

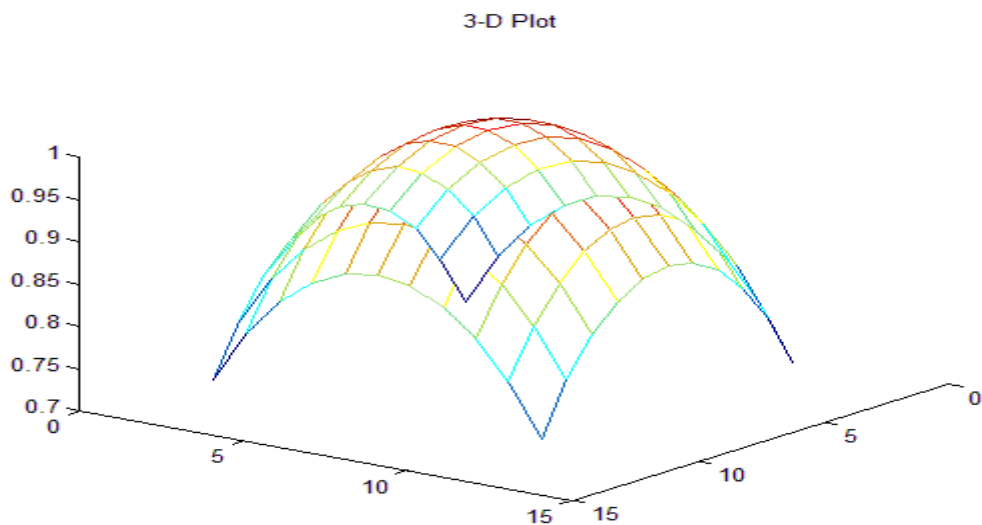
შეიძლება სამგანზომილებიანი გრაფიკის აგებისას საჭირო იყოს აგებულ ზედაპირს რაიმე გარკვეული მიმართულებიდან შევხედოთ. ხედვის მიმართულება განისაზღვრება გრადუსებში გამოხატული აზიმუტისა (ჰორიზონტალური) და სიმაღლის (ვერტიკალური) მიხედვით. აზიმუტის დადებითი მიმართულება საათის ისრის საწინააღმდეგო მიმართულებით იცვლება x ღერძიდან. დადებითი სიმაღლეები გვიჩვენებს ზედაპირს ზემოდან, უარყოფითი – ქვემოდან. თუ არ მიეთითება ხედვის კუთხეები მაშინ MATLAB ავტომატურად აყენებს მათ შემდეგ მნიშვნელობებს: აზიმუტი -37.5 გრადუსი და სიმაღლე 30 გრადუსი, 0 .

ხედვის კუთხე განისაზღვრება $\text{view}(\text{აზიმუტი}, \text{სიმაღლე})$ ბრძანებით. მაგალითად, $\text{view}([-37.5, 0])$ განსაზღვრავს ხედვის შემდეგ კუთხეს $[-37.5, 0]$



ნახ. 7. სფეროს ნაწილი. ხედვის კუთხე $[-37.5, 0]$

მე-8 ნახაზზე გრაფიკი ნაჩვენებია მიმართულებიდან: $[-37.5, -30]$



ნახ. 8. სფეროს ნაწილი. ხედვის კუთხე $[-37.5, -30]$

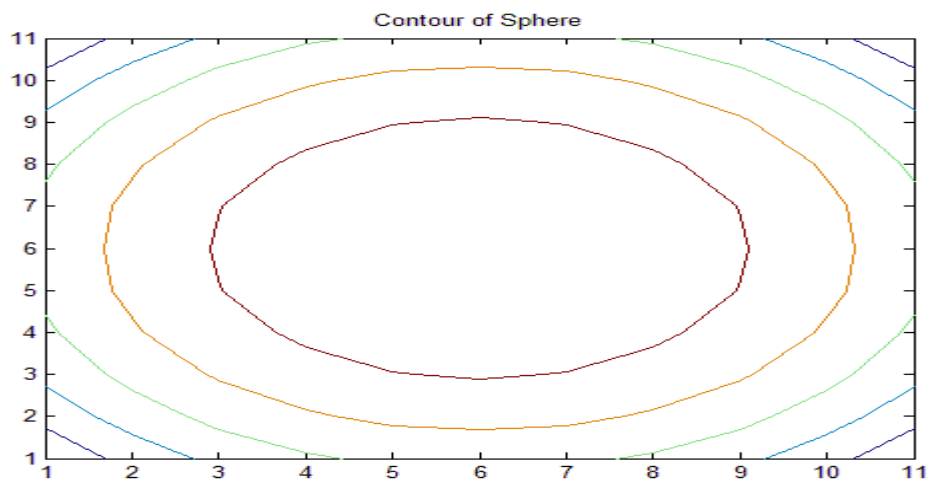
კონტურული გრაფიკი

konturuli ruqa Seicavs wirebis jgufts, romlebic aerTeben erTnair simaRleze myof wertilebs. Tu dedamiwis fizikur ruqaze aseTi wirebiT SevaerTebT zRvis donidan erTiYsimarlis mqone wertilebs, SegviZlia vimsjeloT maTi realuri simaRlis Sesaxeb. MATLAB-is saSualebiT matricis saxiT mocemuli zedapirisaTvis SegviZlia avagoT Sesabamisi konturuli gamosaxuleba. amisaTvis arsebobs brZaneba **contour**.

contour(z) აგებს z მატრიცით მოცემული ზედაპირის კონტურს. კონტურული წირების რაოდენობა და მათი მნიშვნელობები ავტომატრად შეირჩევა MATLAB-ის მიერ.

contour(z,n) აგებს z მატრიცით განსაზღვრული ზედაპირის n კონტურს

contour(z,v) აგებს z მატრიცით განსაზღვრული ზედაპირის კონტურულ გამოსახულებას კონტურული წირებით, რომელთა დონეები მოცემულია v ვექტორის სახით



ნახ. 9-ის გრაფიკი სფეროს ზედაპირის კონტურების

შემდეგი ბრძანებები აგებს სფეროს ზედაპირის კონტურს ნახ. 9:

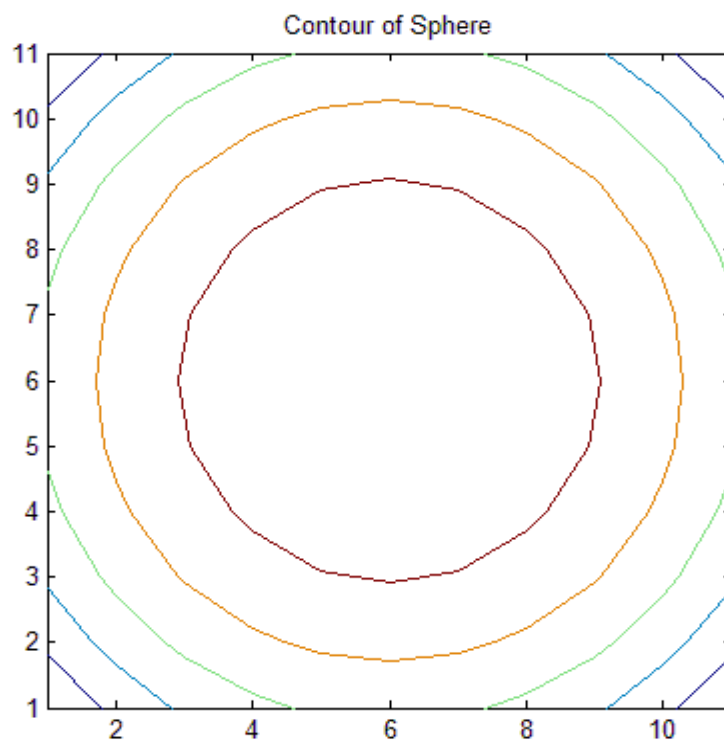
```
[X,Y]= meshgrid(-0.5:0.1:0.5,-0.5:0.1:0.5);
```

```
Z=sqrt(abs(1-X.^2-Y.^2));
```

```
contour(Z,5),...
```

```
title('Contour of Sphere')
```

თუ ამ ბრძანებებს დავამატებთ `axis('square')` ბრძანებას მივიღებთ 0.10-ზე წარმოდგენილ სურათს



ნახ. 10. სფეროს ზედაპირის კონტური ღერძების კვადრატული თანაფარდობით

ბრძანებები და ფუნქციები

axis	აკონტროლებს ღერძების მასშტაბს
bar	აგებს bar გრაფიკს
clf	ასუფთავებს გრაფიკულ ფანჯარას
contour	აგებს კონტურულ გრაფიკს
ginput	იღებს კოორდინატებს გრაფიკული ფანჯრიდან
grid	გრაფიკს უმატებს საკოორდინატო ბადეს
gtext	საშუალებას გვაძლევს ჩავწეროთ ტექსტი უშუალოდ გრაფიკულ ფანჯარაში
hold	ამატებს მიმდინარე გრაფიკს შენახულ გრაფიკულ ფანჯარაში
loglog	აგებს log-log გრაფიკს
mesh	აგებს სამგანზომილებიან mesh გრაფიკს
surf	აგებს სამგანზომილებიან surf გრაფიკს
meshgrid	აწარმოებს x, y მატრიცებს surf და mesh გრაფიკებისთვის
polar	აგებს პოლარულ გრაფიკს
semilogx	აგებს ლოგარითულ-წრფივ გრაფიკს
semilogy	აგებს წრფივ-ლოგარითულ გრაფიკს
shg	ეკრანზე გამოყავს გრაფიკული ფანჯარა
stairs	აგებს საფეხუროვან (stair) გრაფიკს
subplot	დაყოფა გრაფიკულ ფანჯარას რამდენიმე ქვეფანჯარად
text	ჩაწერს მითითებულ ტექსტს გრაფიკულ ფანჯარაში
view	განსაზღვრავს სამგანზომილებიანი გრაფიკის ხედვის კუთხეს