

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**

**Реферат по научному программированию**  
**на тему:**  
**«Средства построения трехмерных графиков»**

**Выполнила:**  
**Студент группы НФИбд-01-21**  
**Студенческий билет №: 1032217044**  
**Коняева Марина Александровна**

**Москва 2023**

## **Оглавление**

<b>Введение .....</b>	<b>3</b>
<b>Построение графиков поверхностей ..... Ошибка! Закладка не определена.</b>	
<b>Построение графиков поверхностей, заданных параметрически Ошибка! Закладка не определена.</b>	
<b>Дополнительные возможности при построении графиков ..... Ошибка! Закладка не определена.</b>	
<b>Заключение .....</b>	<b>12</b>
<b>Список литературы .....</b>	<b>12</b>

## **Введение**

В современном мире научного программирования невозможно обойтись без построения различных графиков. В околонуточных языках таких как Python C/C++ существуют много написанных библиотек, для построения графиков. В данной работе, будет рассмотрено построения графиков в семействе языков MATLAB, которые уже обладают встроенным функционалом. В частности будет рассмотрена реализация на языке Octave.

Цель данной работы: изучить способы построения трехмерных графиков в Octave

Задачи данной работы:

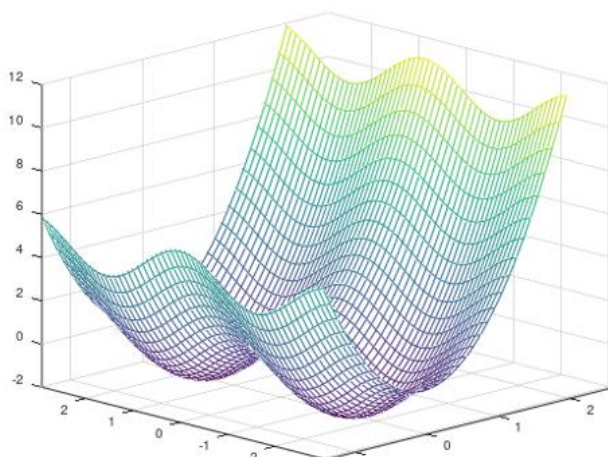
- 1) Разобрать построение графиков поверхностей
- 2) Разобрать построение графиков поверхностей, заданных параметрически
- 3) Разобрать дополнительные возможности при построении графиков

## Построение графиков поверхностей

Дадим определение прямоугольной (или декартовой) системы координат в пространстве. Прямоугольная система координат в пространстве состоит из заданной фиксированной точки  $O$  пространства, называемой началом координат, и трех перпендикулярных прямых пространства  $OX$ ,  $OY$  и  $OZ$ , не лежащих в одной плоскости и пересекающихся в начале координат, их называют координатными осями ( $OX$  – ось абсцисс,  $OY$  – ось ординат,  $OZ$  – ось аппликат). Положение точки  $M$  в пространственной системе координат определяется значением трех координат и обозначается  $M(x,y,z)$ . Три плоскости, содержащие пары координатных осей, называются координатными плоскостями  $XY$ ,  $XZ$  и  $YZ$ .

Величина  $z$  называется функцией двух величин  $x$  и  $y$ , если каждой паре чисел, которые могут быть значениями переменных  $x$  и  $y$ , соответствует одно или несколько определенных значений величины  $z$ . При этом переменные  $x$  и  $y$  называют аргументами функции  $z(x,y)$ . Пары тех чисел, которые могут быть значениями аргументов  $x$ ,  $y$  функции  $z(x,y)$ , в совокупности составляют область определения этой функции.

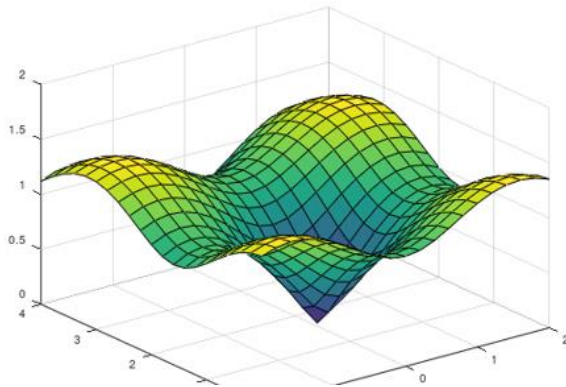
Для формирования прямоугольной сетки в Octave есть функция `meshgrid`. Рассмотрим построение 3-х мерного графика на следующем примере.



$$z(x, y) = 3x^2 - 2\sin^2 y$$

```
>> [x y]=meshgrid(-2:0.1:2,-3:0.1:3);  
>> z=3*x.*x-2*sin(y).^2;  
>> mesh(x,y,z);
```

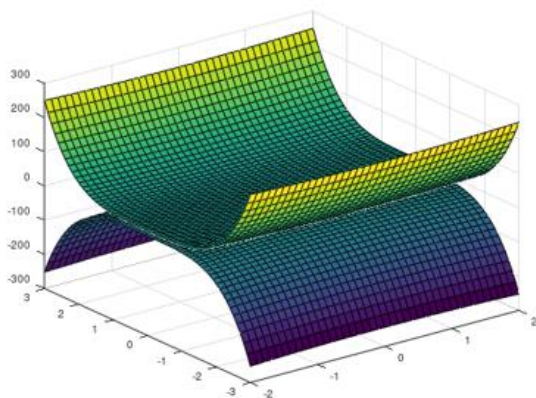
Для построения поверхностей, кроме функции mesh, есть функция surf, которая строит каркасную поверхность, заливая ее каждую клетку цветом, который зависит от значения функции в узлах сетки.



$$z(x, y) = \sqrt{\sin^2 x + \cos^2 y}$$

```
>> [x y]=meshgrid(-2:0.2:2,0:0.2:4);
>> z=sqrt(sin(x).^2+cos(y).^2);
>> surf(x,y,z);
```

Построение графики двух поверхностей в одной системе координат (hold on, surf и mesh)



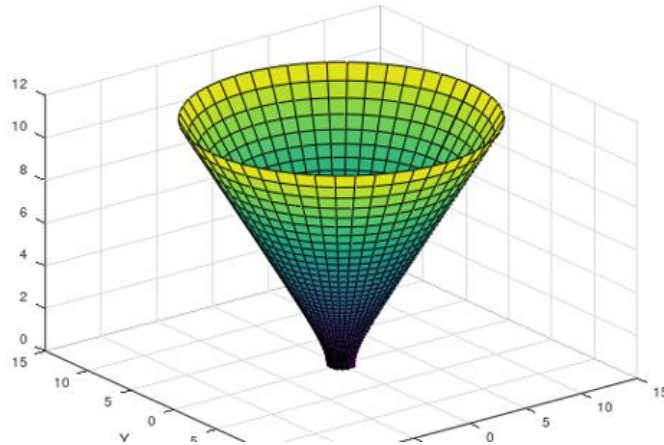
$$z(x, y) = \pm(2x^2 + 3y^4) - 1$$

```
>> h=figure();
>> [x y]=meshgrid(-2:0.1:2,-3:0.1:3);
>> z=2*x.^2+3*y.^4-1;
>> z1=-2*x.^2-3*y.^4-1;
>> surf(x,y,z);
>> hold on
>> surf(x,y,z1);
```

Любой трёхмерный график можно вращать, используя мышку.

## Построение графиков поверхностей, заданных параметрически

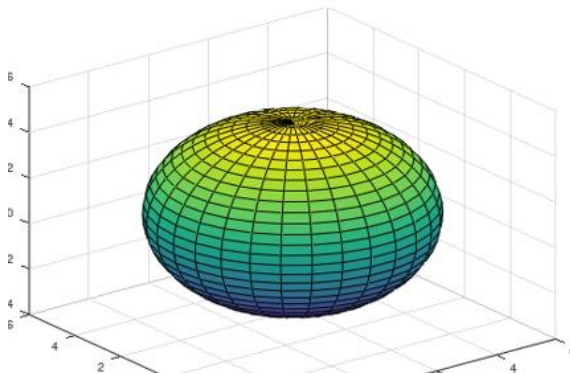
Построение поверхности однополостного гиперболоида,  
 $x(u,v)=ch(u)\cos(v)$ ,  $y(u,v)=ch(u)\sin(v)$ ,  $z(u,v)=sh(u)$ ,  $u \in [0, \pi]$ ,  $v \in [0, 2\pi]$



```
>> h=3.14/50;
>> u=[0:h:3.14]';
>> v=[0:2*h:6.28];
>> X=cosh(u)*cos(v);
>> Y=cosh(u)*sin(v);
>> Z=sinh(u)*ones(size(v));
>> surf(X,Y,Z);
>> grid on;
>> title('Plank hyperboloid');
>> xlabel('X');
>> ylabel('Y');
>> zlabel('Z');
```

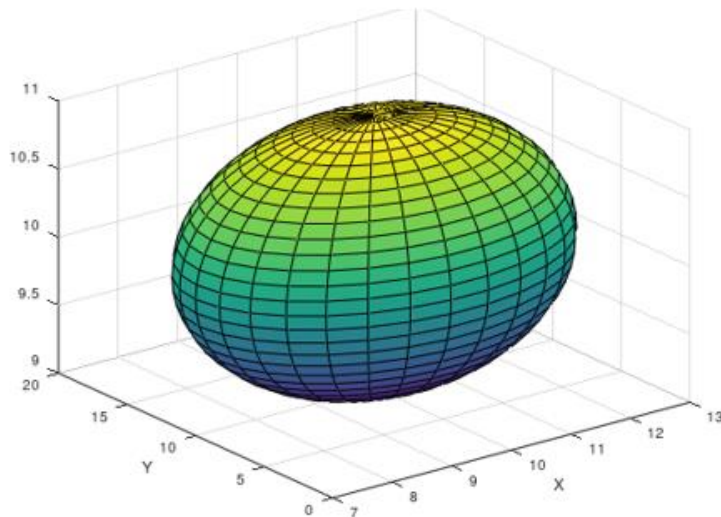
Построение поверхности сферы с центром  $(x_0, y_0, z_0)$  и радиусом  $R$  параметрически (или  $[X,Y,Z]=\text{sphere}(n)$ ;  $\text{surf}(X,Y,Z)$ )

$$x(u, v) = x_0 + R \sin(u) \cos(v), y(u, v) = y_0 + R \sin(u) \sin(v), z(u, v) = z_0 + R \cos(u)$$



```
>> h=pi/30;
>> u=[-0:h:pi]';
>> v=[0:2*h:2*pi];
>> x=1+4*sin(u)*cos(v);
>> y=1+4*sin(u)*sin(v);
>> z=1+4*cos(u)*ones(size(v));
>> surf(x,y,z);
>> grid on;
>> title('SPHERE');
>> xlabel('X');
>> ylabel('Y');
>> zlabel('Z');
```

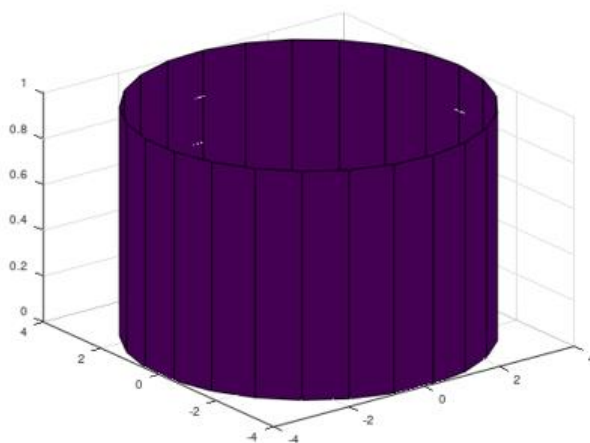
Построение поверхности эллипсоида, где  $a, b, c$  – полуоси эллипсоида,  
 $x_0, y_0, z_0$  – центр эллипсоида



$$\begin{aligned}x(u, v) &= x_0 + a \sin(u) \cos(v), \\y(u, v) &= y_0 + b \sin(u) \sin(v), \\z(u, v) &= z_0 + c \cos(u)\end{aligned}$$

```
>> h=pi/30;
>> u=[-0:h:pi]';
>> v=[0:2*h:2*pi];
>> a=3;b=7;c=1;
>> x0=y0=z0=10;
>> x=x0+a*sin(u)*cos(v);
>> y=y0+b*sin(u)*sin(v);
>> z=z0+c*cos(u)*ones(size(v));
>> surf(x,y,z);
>> grid on;
>> title('ELLIPSOID');
>> xlabel('X');
>> ylabel('Y');
>> zlabel('Z');
```

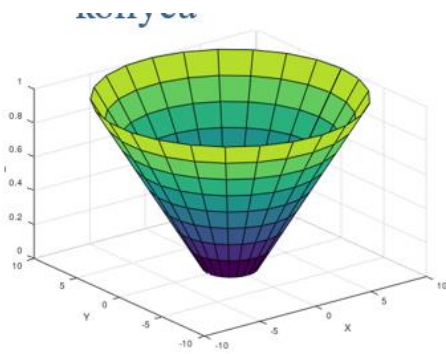
Построение поверхности цилиндра радиуса R=4 и высотой h=1



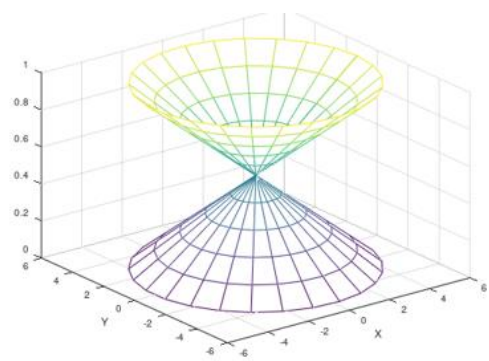
```
>> [x, y, z] = cylinder ([4,4],25);
>> grid on;
>> surf(x, y, z);
>> title ("Cylinder")
```

Построение поверхности усеченного кругового конуса и кругового конуса





```
>> [x, y, z] = cylinder (2:1:10,25);
>> grid on;
>> surf (x, y, z);
>> title ("Cone")
>> xlabel('X');
>> ylabel('Y');
>> zlabel('Z');
```



```
>> [x, y, z] = cylinder ([5,4,3,2,1,0,1,2,3,4,5],25);
>> grid on;
>> mesh(x, y, z);
>> title ("Cone")
>> xlabel('X');
>> ylabel('Y');
>> zlabel('Z');
```

## Дополнительные возможности при построении графиков

Каждый график изображать с помощью функции `plot(x,y)`, но перед обращением к функциям `plot(x2,y2)`, `plot(x3,y3)`, ..., `plot(xn,yn)` вызвать команду `hold on`, которая блокирует режим очистки

Octave представляет дополнительные возможности для оформления графиков:

- команда `grid on` (`grid`) наносит сетку на график, `grid off` убирает сетку с графика;
- функция `axis[xmin, xmax, ymin, ymax]` выводит только часть графика, определяемую прямоугольной областью  $x_{\min} \leq x \leq x_{\max}$ ,  $y_{\min} \leq y \leq y_{\max}$ ;
- функция `title('Заголовок')` предназначена для вывода заголовка графика;
- функции `xlabel('Подпись под осью x')`, `ylabel('Подпись под осью y')` служат для подписей осей  $x$  и  $y$  соответственно;
- функция `text(x,y,'текст')` выводит текст левее точки с координатами  $(x,y)$ ;
- функция `legend('легенда1', 'легенда2', ..., 'легендаn', m)` выводит легенды для каждого из графиков, параметр  $m$  определяет месторасположение легенды в графическом окне: 1 – в правом верхнем углу графика (значение



по умолчанию); 2 – в левом верхнем углу графика; 3 – в левом нижнем углу графика; 4 – в правом нижнем углу графика.

В строке могут участвовать символы, отвечающие за тип линии, маркер, его размер, цвет линии и вывод легенды. Попробуем разобраться с этими символами. За сплошную линию отвечает символ «-». За маркеры отвечают следующие символы (см. табл 1).

*Таблица 1. Символы маркеров*

Символ маркера	Изображение маркера
.	точка
*	✱
x	×
+	+
o	○
s	■
d	◆
v	▼
^	▲
<	▽
>	△
p	□
h	◇

Цвет линии определяется буквой латинского алфавита (см. табл 2), можно использовать и цифры, но на взгляд авторов использование букв более логично (их легче запомнить по английским названиям цветов).

Таблица 2. Цвета линии

Символ	Цвет линии
y	желтый
m	розовый
c	голубой
r	красный
g	зеленый
b	синий
w	белый

При выводе текста с помощью функций xlabel, ylabel, title, text можно выводить греческие буквы (см. табл 3), использовать символы верхнего и нижнего индекса.

Таблица 3. Греческие буквы

Команда	Символ	Команда	Символ
<code>\alpha</code>	$\alpha$	<code>\upsilon</code>	$\upsilon$
<code>\beta</code>	$\beta$	<code>\phi</code>	$\phi$
<code>\gamma</code>	$\gamma$	<code>\chi</code>	$\chi$
<code>\delta</code>	$\delta$	<code>\psi</code>	$\psi$
<code>\epsilon</code>	$\epsilon$	<code>\omega</code>	$\omega$
<code>\zeta</code>	$\zeta$	<code>\Gamma</code>	$\Gamma$
<code>\eta</code>	$\eta$	<code>\Delta</code>	$\Delta$
<code>\theta</code>	$\theta$	<code>\Theta</code>	$\Theta$
<code>\iota</code>	$\iota$	<code>\Lambda</code>	$\Lambda$
<code>\kappa</code>	$\kappa$	<code>\Xi</code>	$\Xi$
<code>\lambda</code>	$\lambda$	<code>\Pi</code>	$\Pi$
<code>\mu</code>	$\mu$	<code>\Sigma</code>	$\Sigma$
<code>\nu</code>	$\nu$	<code>\Upsilon</code>	$\Upsilon$
<code>\xi</code>	$\xi$	<code>\Phi</code>	$\Phi$
<code>\pi</code>	$\pi$	<code>\Psi</code>	$\Psi$
<code>\rho</code>	$\rho$	<code>\Omega</code>	$\Omega$
<code>\sigma</code>	$\sigma$	<code>\forall</code>	$\forall$
<code>\varsigma</code>	$\varsigma$	<code>\exists</code>	$\exists$
<code>\tau</code>	$\tau$	<code>\approx</code>	$\approx$
<code>\int</code>	$\int$	<code>\in</code>	$\in$
<code>\wedge</code>	$\wedge$	<code>\sim</code>	$\sim$
<code>\vee</code>	$\vee$	<code>\leq</code>	$\leq$
<code>\pm</code>	$\pm$	<code>\leftrightarrow</code>	$\leftrightarrow$
<code>\geq</code>	$\geq$	<code>\leftarrow</code>	$\leftarrow$
<code>\infty</code>	$\infty$	<code>\uparrow</code>	$\uparrow$
<code>\partial</code>	$\partial$	<code>\rightarrow</code>	$\rightarrow$

## **Заключение**

В данной работе были изучены способы построения графиков в Octave

Были выполнены следующие задачи:

- 1) Разобрать построение графиков поверхностей
- 2) Разобрать построение графиков поверхностей, заданных параметрически
- 3) Разобрать дополнительные возможности при построении графиков

## **Список литературы**

1. Введение в Octave для инженеров и математиков: / Е. Р. Алексеев, О. В. Чеснокова — М.: ALT Linux, 2012. — 368 с.: ил. — (Библиотека ALT Linux).
2. Программирование на Octave[wiki]//Построение графиков:  
[https://ru.wikibooks.org/wiki/Программирование\\_на\\_Octave/Построение\\_графиков](https://ru.wikibooks.org/wiki/Программирование_на_Octave/Построение_графиков)
3. Программирование на Octave[wikibooks]:  
[https://ru.abcdef.wiki/wiki/Scientific\\_programming\\_language](https://ru.abcdef.wiki/wiki/Scientific_programming_language)