Санкт-Петербургский государственный политехнический университет

Кафедра «Прикладная математика»

Смирнов А.Б., Смирнова Е.Н.

Методические указания к выполнению заданий курсового проекта по курсу «Математические пакеты программ»

Санкт-Петербург

2013

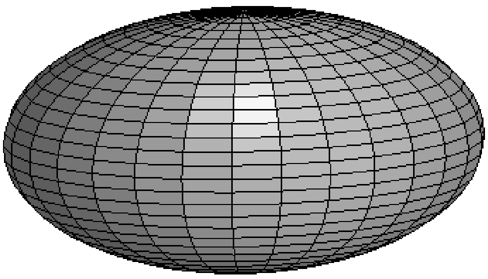
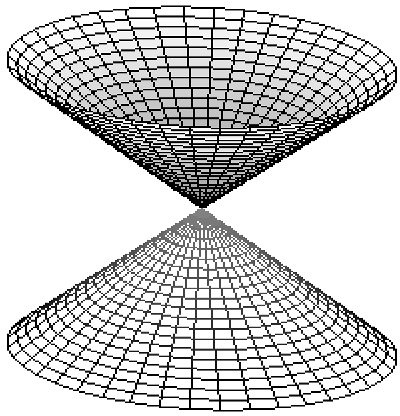
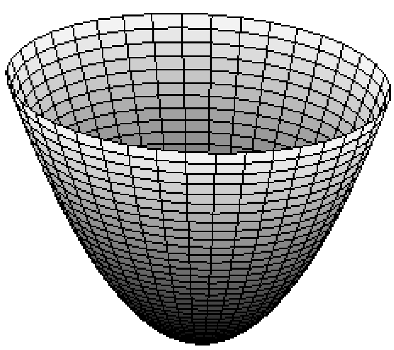
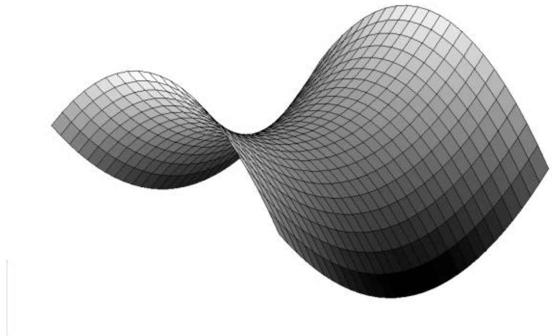
Одной из тем курсового проекта для студентов 1-ого курса при изучении пакета MATLAB является построение взаимного расположения в пространстве двух поверхностей второго порядка с использованием функций ***surf*** и ***mesh*** [1], т.е. без использования специальных функция типа ***ellipsoid*** или ***cylinder.***

Уравнение любой поверхности 2-ого порядка можно записать в виде:

$$\lambda\cdot(x-x_0)^2+\mu\cdot(y-y_0)^2+\nu\cdot(z-z_0)^2+\alpha\cdot(x-x_1)+\beta\cdot(y-y_1)y+\gamma\cdot(z-z_1)=\delta$$

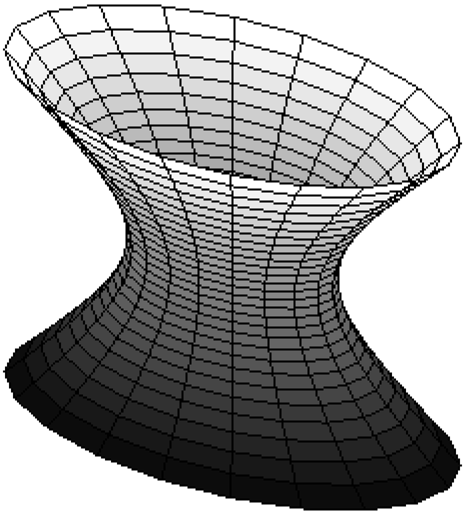
В зависимости от знаков и величины (ноль или нет) коэффициентов $\lambda,\mu,\nu,\ \alpha,\beta,\gamma,\delta$ различают следующие типы поверхностей [2,3] (см. таблица 1). Далее используются общепринятые обозначения коэффициентов.

Таблица 1 Уравнения и вид поверхностей.

Эллипсоид $\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}+\frac{(z-z_0)^2}{c^2}=1$ (1) 	Конус $\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}-\frac{(z-z_0)^2}{c^2}=0$ (2) 
Параболоид эллиптический $z-z_0=\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}$ (3) 	Параболоид гиперболический $z-z_0=\frac{(x-x_0)^2}{a^2}-\frac{(y-y_0)^2}{b^2}$ (4) 

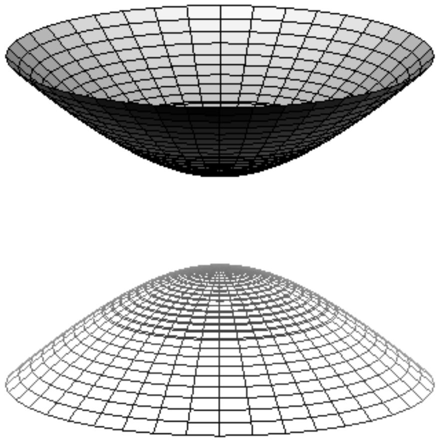
Гиперболоид однополостный

$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}-\frac{(z-z_0)^2}{c^2}=1 \quad (5)$$



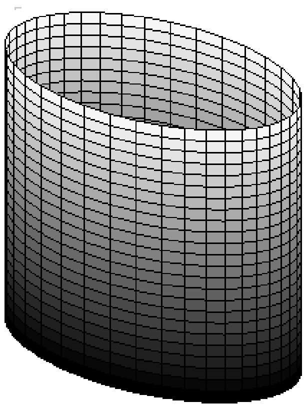
Гиперболоид двуполостный

$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}-\frac{(z-z_0)^2}{c^2}=-1 \quad (6)$$



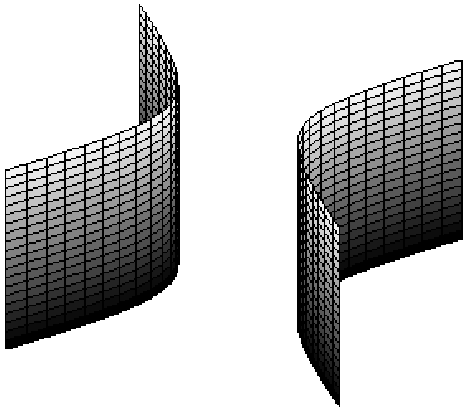
Цилиндр эллиптический

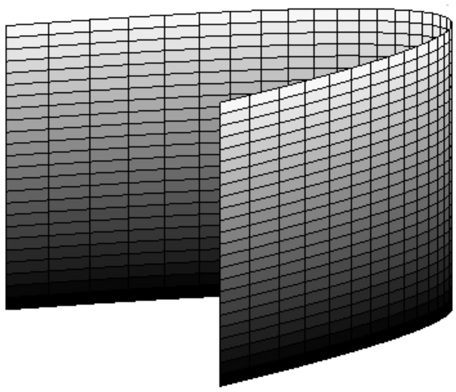
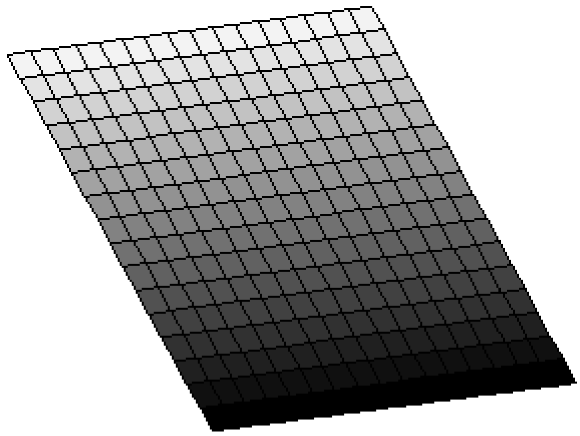
$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}=1 \quad (7)$$



Цилиндр гиперболический

$$\frac{(x-x_0)^2}{a^2}-\frac{(y-y_0)^2}{b^2}=1 \quad (8)$$



Цилиндр параболический	Плоскость (вырожденная поверхность)
$\frac{(x-x_0)^2}{a^2}-2b(y-y_0)=0$ (9)	$\frac{x-x_0}{a}+\frac{y-y_0}{b}+\frac{z-z_0}{c}=1$ (10)
	

Пакет MATLAB имеет развитые средства для графического представления данных. Изображение поверхности с помощью указанных функций основано на использовании ее уравнения в явном виде

$$z = f(x,y) \quad \text{, или } y = f(x,z) \quad \text{, или } x = f(y,z) \qquad (11)$$

Рассмотрим некоторые вопросы, вызывающие затруднения у студентов при выполнении этого задания:

1. Некоторые уравнения не содержат одну из независимых переменных (например, цилиндр – переменную z), и студенты не понимают, как задать точку на поверхности.

2. Если приводить часть уравнений к виду (11) потребуется извлекать квадратные корни, и, иногда подкоренные выражения оказываются отрицательными, что приводит к комплексным числам.

3. При изображении взаимного расположения нескольких поверхностей определенные трудности студенты испытывают с выбором цветовой палитры, т.к. стандартные средства MATLAB изменения палитры применяются ко всем объектам изображения.

4. Определенные трудности испытывают студенты при выборе масштаба изображаемых объектов и точки обзора.

Далее приведены примеры, показывающие как справиться с возникающими проблемами.. Для иллюстрации используется эллиптический цилиндр, описываемый уравнением (7), и плоскость (10). Считается, что студенты знакомы с функциями

MATLAB ***surf*** и ***mesh*** [1] в простейшей форме, необходимыми для выполнения данного задания.

Начнем с построения отдельной поверхности. Пусть требуется построить эллиптический цилиндр (7) со следующими параметрами: $a = 5.2, \quad b = 4.3, \quad c = 3,$ $x_0 = 0.7, \quad y_0 = 1.2$.

1. Обсудим, как задать высоту цилиндра. Отсутствие координаты z в уравнении (7) означает, что любое сечение фигуры плоскостью, перпендикулярной оси z ,выглядит одинаково, то есть не зависит от координаты z ,поэтому высоту цилиндра вдоль оси z назначим по своему усмотрению. Будем задавать точку, лежащую на поверхности уравнением $y = f(x, \mathrm{z}) = \mathrm{f(x)}$ (в нашем случае функция не зависит от переменной z). Заметим также, что ось цилиндра смещена относительно начала координат и проходит через точку (x_0, y_0).

2. Рассмотрим, что делать с комплексными числами при извлечении корней. Попробуем сначала построить цилиндр, используя декартову систему координат. Выразим одну из координат из уравнения (7), например, y через - x, (z в уравнение не входит).

Так как функция ***surf***, предназначенная для построения поверхностей, не работает с комплексными числами, то перед обращением к ней комплексные элементы массивов Yc1 и Yc2 необходимо заменить на NaN (Not-a-Number)

Следующий фрагмент кода с комментариями реализует задачу:

```
% задаем параметры цилиндра
a=5.2;
b=4.3;
c=3;
x0=0.7;
y0=1.2;
% создаем сетку по осям X и Z
x=-6:0.25:6;
z=-8:0.5:8;
 [X,Z]=meshgrid(x,z);
% вычисляем значения Y
Yc1=sqrt(1-(X-x0).^2/a^2)*b;
Yc2=-sqrt(1-(X-x0).^2/a^2)*b;
% исключаем комплексные элементы массивов Yc1 и Yc2
%
% находим индексы комплексных элементов массива Yc1
ind = abs(imag(Yc1))>0;
% и заменяем их значением NaN
Yc1 (ind) = NaN;
% находим индексы комплексные элементов массива Yc2
ind = abs(imag(Yc2))>0;
% и заменяем их значением NaN
Yc2 (ind) = NaN;
% строим цилиндр в декартовых координатах
figure
surf(X,Yc1,Z)
hold on
surf(X,Yc2,Z)
```

Замечание. Обратите внимание, как заполняются элементы массивов значениями ***NaN***. Вместо использования поэлементного заполнения в цикле, например,

```
[n,m]=size(Yc1)
for i=1:n
    for j=1:m
        if abs(imag(Yc1(i,j)))>0
        Yc1(i,j)=NaN;
        end
    end
end
```

используются функции MATLAB, ориентированные на работу с массивами

```
ind = abs(imag(Yc1)) > 0;
Yc1 (ind) = NaN;
```

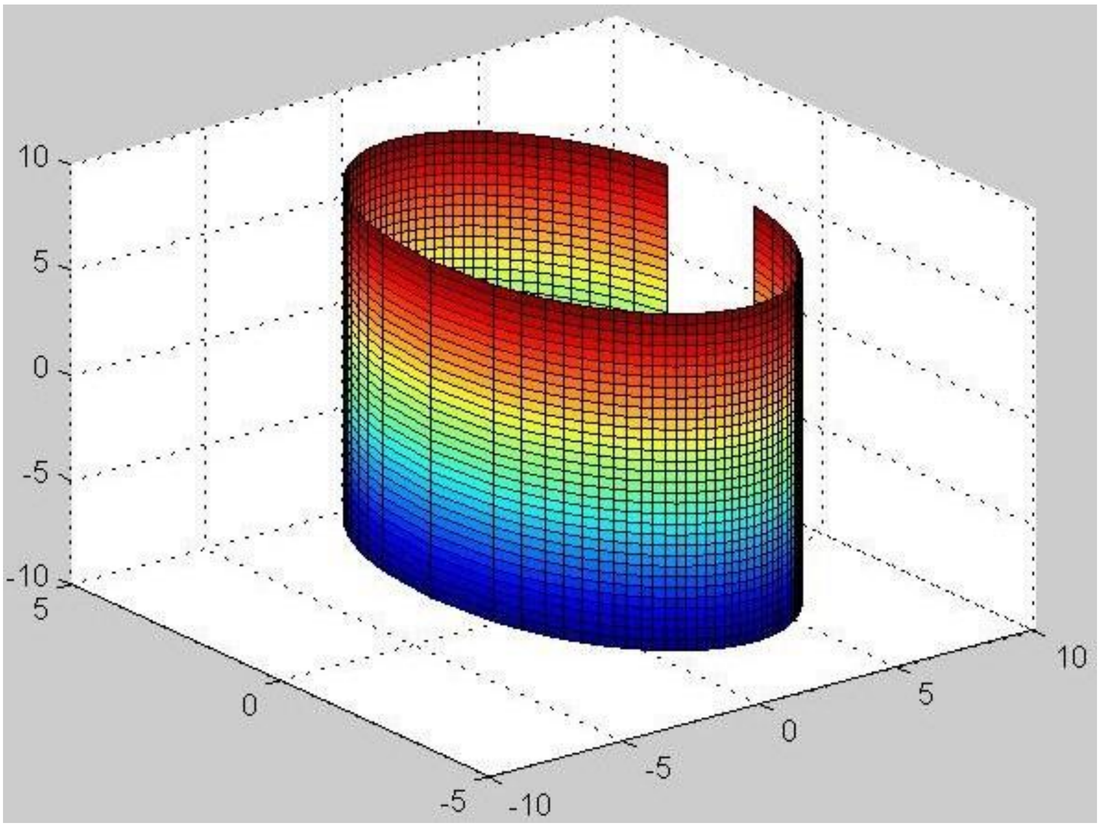


Рис.1. Построение цилиндра с исключением комплексных значений

Видно, что изображение получается искаженным – границы половинок цилиндра не смыкаются, так как получившиеся комплексные значения заменяются ***NaN*** и не изображаются на рисунке. Можно попробовать изменить рисунок, задав более мелкую сетку путем выбора шага вдоль осей, но, как правило, это не приводит к желаемому результату, хотя неотображаемая область уменьшится.

Одним из возможных подходов является применение параметрического задания поверхности. Для поверхностей второго порядка можно использовать связь декартовых координат с цилиндрическими или сферическими в зависимости от уравнения.

Для изображения цилиндра (7) ведем обозначения

$$x = x_0 + a\cos(\varphi)$$
$$y = y_0 + b\sin(\varphi)$$
$$z = z$$

Параметр $\varphi \in [0, \pi]$ определяет угловую координату в плоскости xOy. Уравнение (7) выполняется автоматически при любых φ и z. Поэтому массивы точек в пространстве, лежащих на поверхности задаются элементарно. Следующий фрагмент кода с комментариями иллюстрирует построение заданного цилиндра.

```
% построение с использованием параметрического задания
a=5.2;
b=4.3;
c=3;
x0=0.7;
y0=1.2;
phi=0:pi/20:2*pi;
z=-8:0.5:8;
[PHI,Z]=meshgrid(phi,z);
X=x0+a*cos(PHI);
Y=y0+b*sin(PHI);
figure
surf(X,Y,Z)
colormap(spring)
```

В результате получим следующее изображение (рис.2). По сравнению с предыдущим рисунком дополнительно изменена цветовая гамма.

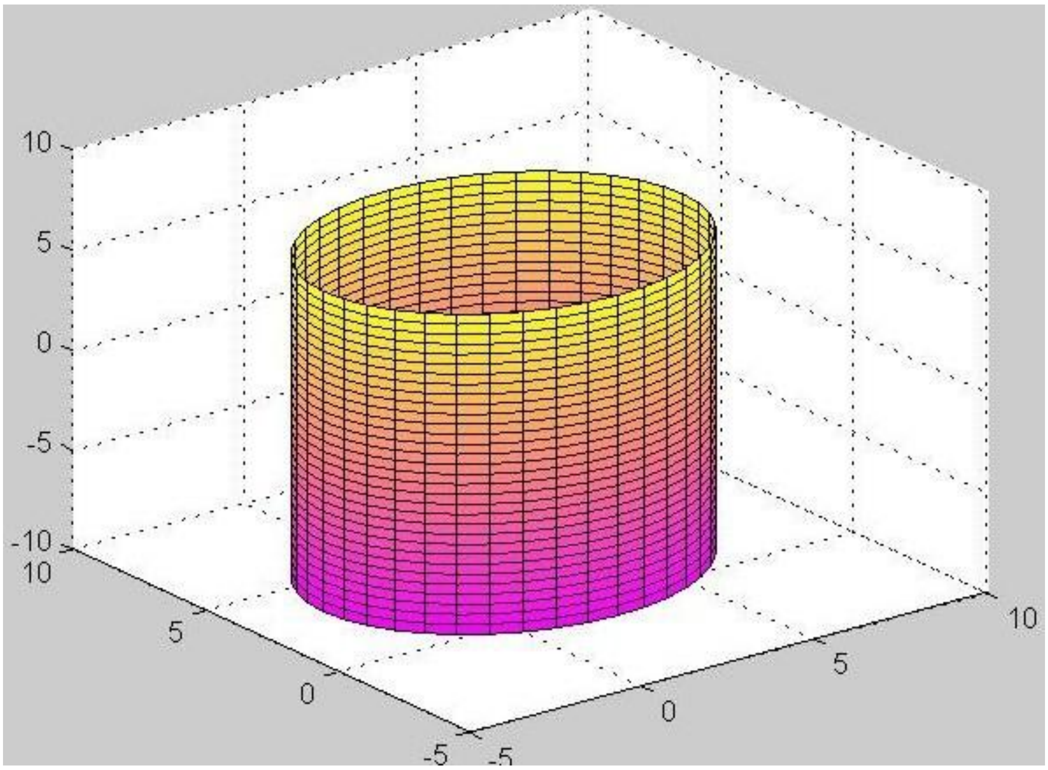


Рис.2. Построение цилиндра с использованием параметрического задания

3. Проиллюстрируем один из приемов использования разных цветовых палитр для двух графических объектов в одних осях. Добавим к ранее построенному цилиндру еще один цилиндр, повернутый относительно оси X на угол $\alpha = \dfrac{\pi}{3}$.

```
% определим цилиндр в системе координат, повернутой вокруг оси x  на угол alf
alf=pi/3;
cs=cos(alf);
sn=sin(alf);
% задаем  высоту нового цилиндра и пересчитываем сетку
phi=0:pi/20:2*pi;
zn=-9:0.5:9;
[PHIN,ZN]=meshgrid(phi,zn);
XP=x0+a*cos(PHIN);
YP=y0+b*sin(PHIN);
% пересчитываем координаты повернутого цилиндра
Yr=YP.*cs+ZN*sn;
Zr=-YP*sn+ZN*cs;
% задаем палитру для повернутого цилиндра
[nz,mz]=size(Zr);
colo=zeros(nz,mz,3);
for j=1:mz
  colo(:,j,:)=summer(nz);
end
% рисуем повернутый цилиндр вместе с предыдущим
hold on
surf(XP,Yr,Zr,colo)
```

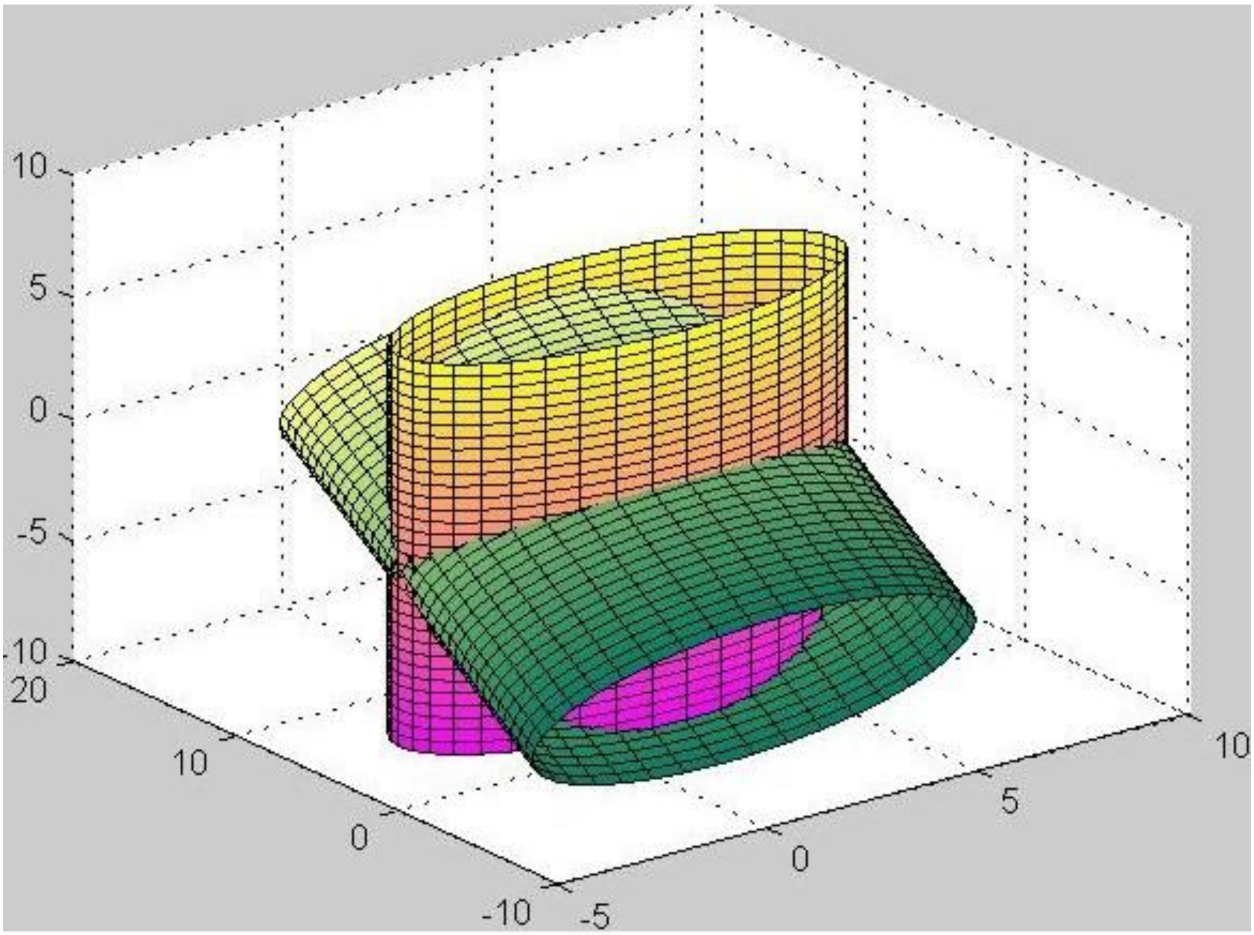


Рис.3. Использование разных цветовых палитр для изображения двух цилиндров

Для задания палитры для второго цилиндра, формируется массив colo, размерности [nz,mz,3], где [nz,mz]=size(Zr), и использующийся в качестве параметра цветовой палитры в обращении к функции surf [1].

4. Покажем, как выбрать области для построения объектов для получения влияет приемлемого результата. Рассмотрим построение на одном рисунке ранее построенного цилиндра и плоскости, заданной уравнением.

$$0.3x + y + 0.3z - 2 = 0 \quad .$$

Дадим несколько рекомендаций:

- Первоначально следует выбрать заведомо большую область построения, чтобы все объекты были видны и, если они пересекаются, добиться видимости области пересечение объектов. Это позволяет визуально выбрать наиболее рациональную область в окончательном варианте.
- Следует использовать функцию ***linspace*** [1], чтобы число точек на поверхности не зависело от выбранной области построения, т.е. шаг дискретизации вычисляется автоматически.
- Поскольку в пакете MATLAB происходит автоматическое масштабирование по осям, для более реального изображения можно воспользоваться функциями ***xlim, ylim, zlim***, [1], чтобы уменьшить искажение объектов, связанное с масштабированием.

```
% построение цилиндра и плоскости на одном рисунке
% параметры поверхностей и области построения
a=5.2;
b=4.3;
c=3;
x0=0.7;
y0=1.2;
xl = -7;
xh= 7;
yl = -10;
yh= 20;
zl = -50;
zh = 20;
nx = 20;
ny = 20;
nz = 20;
nphi = 20;
% создаем сетку для построения цилиндра
phi=linspace(0, 2*pi, nphi);
z=linspace(xl, zh ,nz);
[PHI,Z]=meshgrid(phi,z);
X=x0+a*cos(PHI);
Y=y0+b*sin(PHI);
% строим цилиндр
figure
surf(X,Y,Z);
colormap(winter);
hold on
% строим сетку для плоскости
x=linspace(xl, xh, nx);
z=linspace(yl, yh, ny);
```

```
[Xf,Zf]=meshgrid(x,z);
Yf=2-0.3*Zf-0.3*Xf;
%задаем палитру для плоскости
[nf,mf]=size(Zf)
colorf=zeros(nf,mf,3);
for j=1:mf
  colorf(:,j,:)=gray(nf);
end
% строим плоскость
grid on
surf(Xf,Yf,Zf,colorf)
al = min([xl , yl]);
ah = max([yh , xh]);
xlim ([al , ah])
ylim ([al, ah])
```

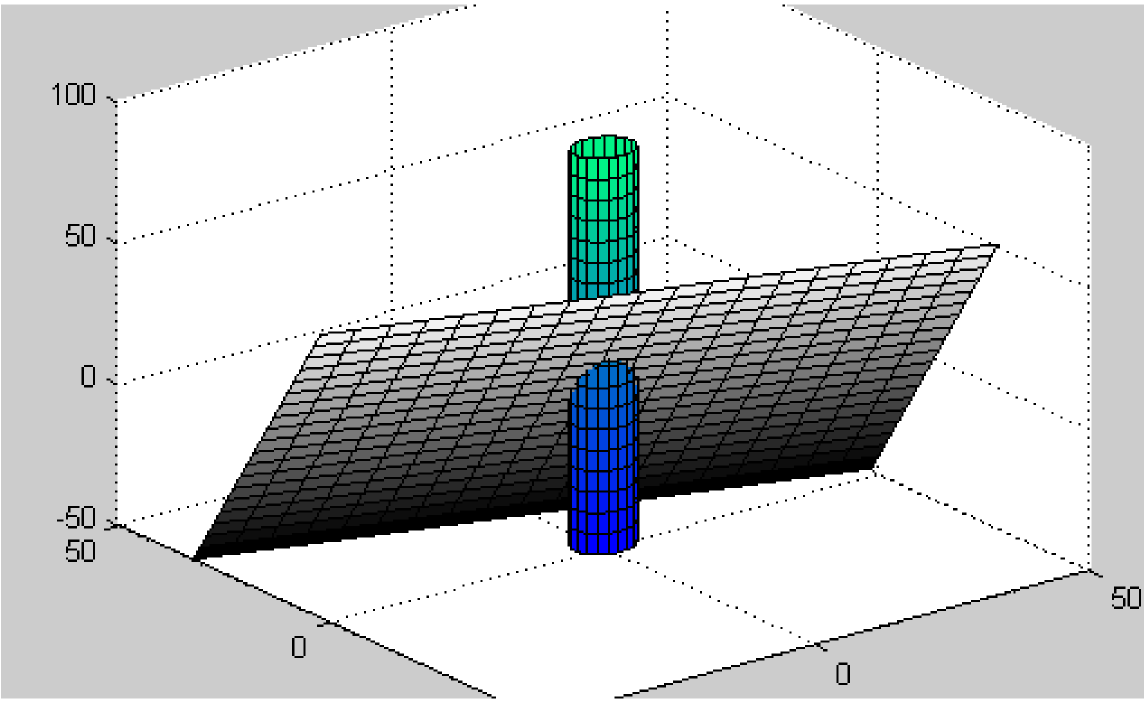


Рис.4. Плоскость и цилиндр – предварительный результат

В силу бесконечной протяженности цилиндра в направлении оси z и плоскости во всех направлениях, границы по этим направлениям можно задать любыми. Задание пределов области в виде переменных позволяет воспользоваться тем же кодом с заменой числовых значений. Рисунок 4 дает возможность визуально выбрать подходящую область построения, изменив только переменные, определяющие область

```
xl = -7;  xh= 7;
yl = -10; yh= 20;
zl = -50; zh = 20;
```

Результат представлен на рис.5. Дополнительно можно воспользоваться функцией `view[1],` для изменения точки обзора на объекты.

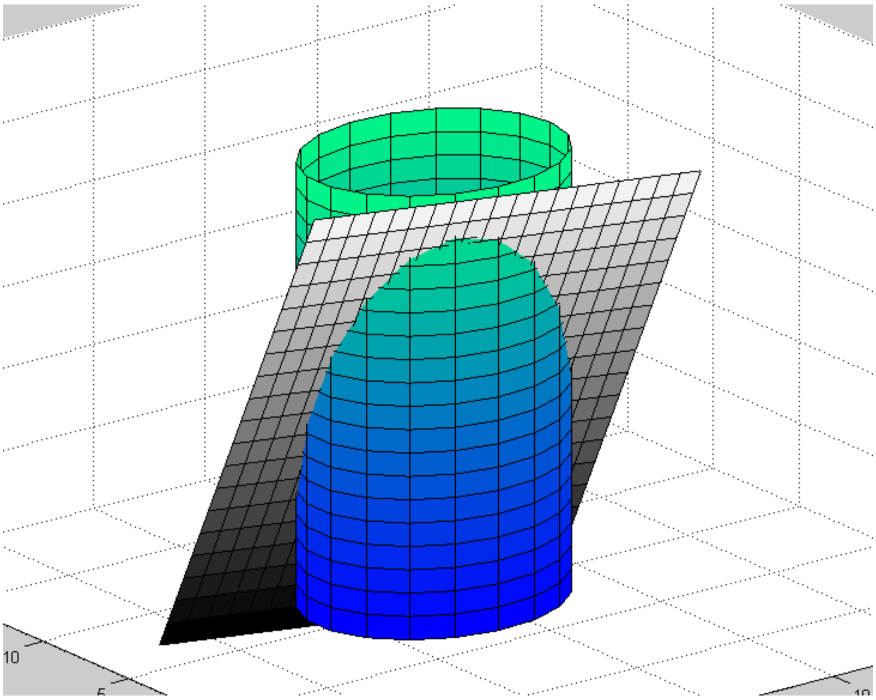


Рис.5 Плоскость и цилиндр – окончательный вариант.

По аналогии строятся другие поверхности второго порядка.

ЛИТЕРАТУРА

1. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н.. MATLAB 7. – СПб: БХВ-Петербург, 2005.-1104с.
2. Ильин В. А., Позняк Э. Г - Аналитическая геометрия, Физматлит,2009.-224с.
3. Привалов И.И. Аналитическая геометрия. - Лань, 2008.- 304 с.