

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

---

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра инженерной графики

**В.Д. Дубовец, В.А. Столер, В.Ф. Бондаренко**

**ПОСТРОЕНИЕ ГРАФИЧЕСКИХ МОДЕЛЕЙ  
В СРЕДЕ MATLAB**

*Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве учебно-методического пособия по учебной дисциплине «Инженерная компьютерная графика» для специальностей, закрепленных за УМО*

**БГУИР**

Минск 2014

УДК 004.925 (076)  
ББК 32.973.26–018.2я73  
Д79

Рецензенты:  
кафедра автоматизированных информационных систем  
частного учреждения образования «Минский институт управления»  
(протокол №6 от 28.01.2014);  
заведующий кафедрой информационных технологий Республиканского  
института инновационных технологий Белорусского национального  
технического университета,  
кандидат физико-математических наук, доцент В.В.Сидорик

Д79 Построение графических моделей в среде MATLAB : учеб.-  
метод. пособие / В.Д. Дубовец, В.А. Столер, В.Ф. Бондаренко. Минск:  
БГУИР, 2014. – 70 с.

**ISBN 978-985-543-072-9.**

Приведены материалы, необходимые для приобретения основных навыков по формированию в среде MATLAB графических моделей разнообразных функций, объясняющих математические зависимости, а также иллюстрирующих физические, химические, электромагнитные и другие процессы. Полученные знания могут быть использованы при оформлении и защите лабораторных работ, курсовых и дипломных проектов, а также для решения разнообразных задач на практике.

УДК 004.925(076)  
ББК 32.973.26–018.2я73

**978-985-543-072-9**

© Дубовец В. Д., Столер В. А., Бондаренко В. Ф., 2014  
© УО «Белорусский государственный университет  
информатики и радиоэлектроники», 2014

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. Двухмерная (2D) графика.....	5
1.1. Графики в линейном масштабе.....	5
1.2. Специальные виды графиков.....	9
1.2.1. Представление функции в виде дискретных отсчетов.....	9
1.2.2. Лестничные графики.....	11
1.2.3. Графики с указанием погрешности.....	12
1.2.4. Графики в логарифмическом и полулогарифмическом масштабах.....	13
1.3. Графики параметрических функций.....	14
2. Трехмерная (3D) графика.....	14
2.1. Линейчатые поверхности.....	15
2.2. Каркасные поверхности.....	17
2.2.1. Прозрачная каркасная поверхность.....	19
2.2.2. Каркасная поверхность, залитая цветом.....	20
2.3. Контурные графики.....	22
2.4. Сплошная освещенная поверхность.....	25
3. Особенности работы в MatLab.....	26
3.1. Работа с M-файлами.....	26
3.2. Файл-программы.....	27
3.3. Встроенные функции.....	29
4. Представление графических моделей.....	29
4.1. Оформление графиков.....	29
4.2. Сохранение и экспорт графиков.....	31
4.3. Анимация.....	31
ПРИЛОЖЕНИЕ 1 Задания для закрепления теоретического материала, 2D-графика . . . . .	33
ПРИЛОЖЕНИЕ 2. Задания для закрепления теоретического материала, 3D-графика . . . . .	56
ЛИТЕРАТУРА.....	69

## ВВЕДЕНИЕ

Одна из задач курса инженерной компьютерной графики состоит в том, чтобы научить студентов основным навыкам формирования графических моделей разнообразных функций, объясняющих математические зависимости, а также иллюстрирующих физические, химические, электромагнитные и другие процессы. Применение компьютерной графики часто оказывается полезным не только для наглядного представления результатов вычислений, но и для отладки инженерных и научных программ. Такие знания могут быть также эффективно использованы при оформлении и защите лабораторных работ, курсовых и дипломных проектов, а также для решения разнообразных задач на практике.

**MatLab** позволяет пользователю получать результаты в графическом виде, прикладывая минимум усилий. Основную работу, связанную с построением графика, масштабированием осей, подбором цветов, а в случае 3-мерной графики и с углом обзора наблюдаемого объекта берет на себя среда **MatLab**.

В части 1 и 2 приводятся сведения, необходимые для построения графиков следующих типов:

- графики на плоскости в декартовой системе координат;
- трехмерные графики функций двух переменных в декартовой системе координат и особенности их построения;

При создании с помощью компьютера графика той или иной зависимости необходимо иметь в виду, что компьютер способен работать только с числами, но не с непрерывными значениями аргументов и их функций. Поэтому, если порой наблюдателю кажется, что график той или иной функции представлен на экране дисплея весьма плавными кривыми или объемными телами, то это всего лишь иллюзия. Такая иллюзия достигается благодаря тому, что между отдельными значениями аргументов и их функций выбран достаточно малый интервал, а также тем, что современные дисплеи и принтеры обладают настолько высокой разрешающей способностью, что глаз человека не способен воспринимать имеющиеся неоднородности в графической модели. Если этот интервал увеличить, то на графике функции будут видны изломы.

При чтении материала следует учитывать то, что серый цвет в текстах программ использован для выделения фрагментов, которые содержат функции вывода графиков или их оформления.

Изучать графику **MatLab**, пользуясь настоящими материалами, рекомендуется в следующей последовательности:

- загрузить среду **MatLab**;
- ознакомиться с Приложениями 1 и 2;
- приступить к решению задач 2D-графики (Задания для закрепления теоретического материала, 2D-графика), выбирая свой вариант по указанию преподавателя. Перед решением каждой задачи следует ознакомиться с

соответствующим примером из раздела 1. Правильность решения необходимо установить путем визуального сравнения полученного графика с графиком, приведенным рядом с условием решаемой задачи;

– решить задачи 3D-графики (Задания для закрепления теоретического материала, 3D-графика). Перед решением каждой задачи следует ознакомиться с соответствующим примером из раздела 2. Правильность решения необходимо установить путем визуального сравнения полученного графика с графиком, приведенным рядом с условием решаемой задачи.

## 1. Двухмерная (2D) графика

Под 2D-графикой будем понимать визуализацию результатов вычислений с построением графических зависимостей на плоскости в декартовой (прямоугольной) или полярной системе координат. Дальнейший материал ограничен рассмотрением только декартовой системы координат, как наиболее часто употребляемой для решения практических задач. При этом возможно построение графиков функций одной переменной или графиков функций, заданных параметрически, т. е. двумя уравнениями  $x(t)$  и  $y(t)$  при изменении  $t$  в заданных пределах.

При графической визуализации параметрически заданных функций вначале для одинаковых значений  $t$  вычисляются значения функций  $x(t)$  и  $y(t)$ , а затем строится график зависимости  $y(x)$ .

### 1.1. Графики в линейном масштабе

Рассмотрим построение графика на примере простой функции одного аргумента.

Пусть этой функцией будет  $y(t)=e^{-t}\cdot\sin(10\cdot t)$ , где аргумент  $t$  меняется в интервале  $[0,1]$  с шагом 0,01 (рис. 1).

При построении графика этой функции **MatLab** вычисляет ее значения, начиная с точки 0 до точки 1 перемещаясь по оси аргумента  $t$  с шагом 0,01. При этом значения функции  $y(t)$ , вычисленные на каждом шаге, соединяются прямыми линиями.

Если шаг изменения аргумента выбран достаточно малым, то глаз наблюдателя не замечает изломов между значениями функции. Если шаг увеличить, то изломы будут видны.

Построение графика функции одной переменной (или нескольких таких графиков) в линейном масштабе осуществляется с помощью функции **plot(...)**.

Рассмотрим использование функции **plot(...)** на конкретных примерах.

*Пример 1.* Построить график функции  $y(t)=e^{-t}\cdot\sin(10\cdot t)$ , где аргумент  $t$  меняется в интервале  $[0,1]$  с шагом 0,01 (см. рис. 1).

```
t=[0:0.01:1];  
y=exp(-t).*sin(10*t);  
plot(t,y)
```

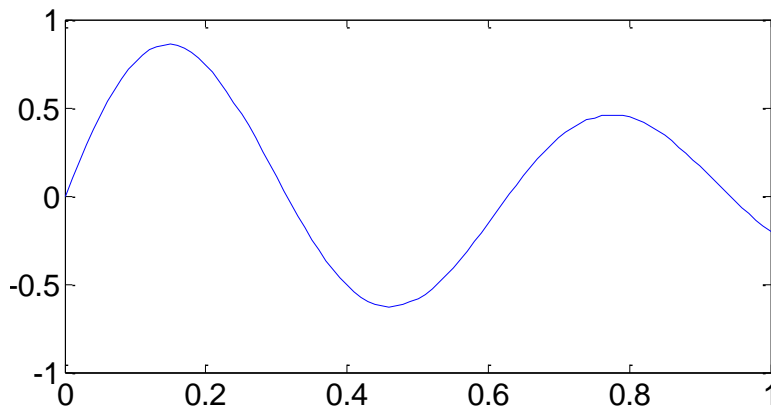


Рис. 1

Обсудим построенный график.

1. В этом примере строка программы **t=[0:0.01:1];** определяет пределы и шаг изменения аргумента (пределы изменения – от 0 до 1, шаг – 0,01). Если после строки стоит знак «;», то вычисляемые значения не выводятся. Если такой знак в конце строки не стоит, то все вычисленные значения будут выведены в командное окно.

Во второй строке **y=exp(-t).\*sin(10\*t);** записана функция, значения которой вычисляются для каждого из значений аргумента **t**.

В третьей строке **plot(t,y)**, выделенной серым цветом, записана встроенная функция **MatLab**, которая выводит в декартовой системе координат график одномерной функции **y** с аргументом **t**. При этом на первом месте в скобках стоит аргумент, а на втором функция.

Пример 2. Построить в одном окне графики двух функций  $x(t)=2 \cdot \sin(50 \cdot t)$  и  $y(t)=e^{-t} \cdot \sin(10 \cdot t)$ , где аргумент **t** меняется в интервале [0,2] с шагом 0,01. (рис. 2).

```
t=[0:0.01:2];  
y=exp(-t).*sin(10*t);  
x=2*sin(50*t);  
plot(t,x,t,y)
```

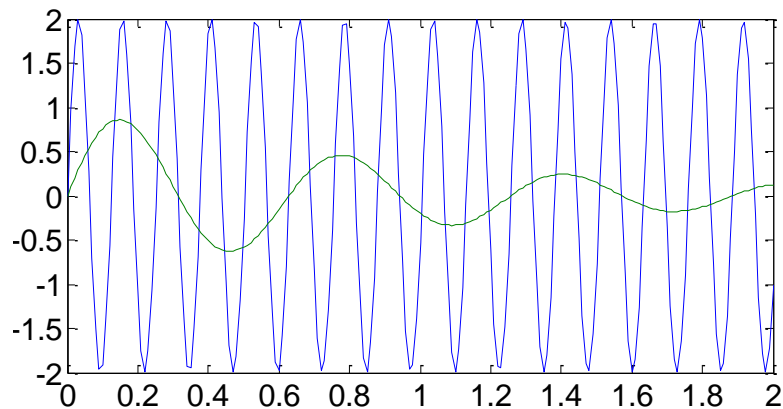


Рис. 2

Обсудим построенные графики.

**MatLab** (если это не указано специально) самостоятельно выбирает цвет графика, масштаб и оцифровывает оси абсцисс и ординат. Из рис. 2 видно, что масштабирование производится по максимальным значениям одной из функций и ее аргумента.

Для функции  $x(t)=2 \cdot \sin(50 \cdot t)$  шаг изменения аргумента оказался не достаточно малым и поэтому на ее графике отчетливо видны изломы.

Иногда требуется сравнить поведение двух функций, значения которых сильно отличаются друг от друга. При этом график функции с небольшими значениями практически сливается с осью абсцисс, и установить его вид не удастся. В таком случае можно воспользоваться функцией **plotyy(...)**, которая

выводит графики в окно с двумя осями ординат, имеющими подходящий масштаб.

*Пример 3.* Увеличим амплитуду синусоиды функции  $x(t)$  из примера 2 в 50 раз, т. е. положим что  $x(t)=100\cdot\sin(50\cdot t)$ , а  $y(t)=e^{-t}\cdot\sin(10\cdot t)$ , где аргумент  $t$  меняется от 0 до 2 с шагом 0,01. Если для построения графиков этих функций использовать функцию **plot(...)**, то получим график, представленный на рис. 3.

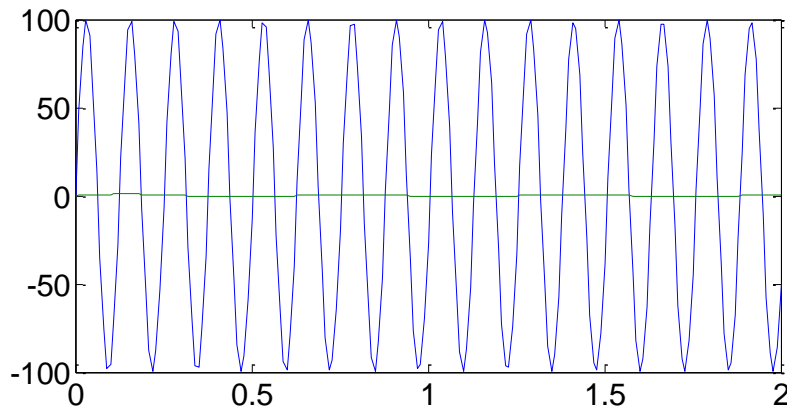


Рис. 3

Выведем теперь графики тех же функций с помощью функции **plotyy(...)** (рис. 4).

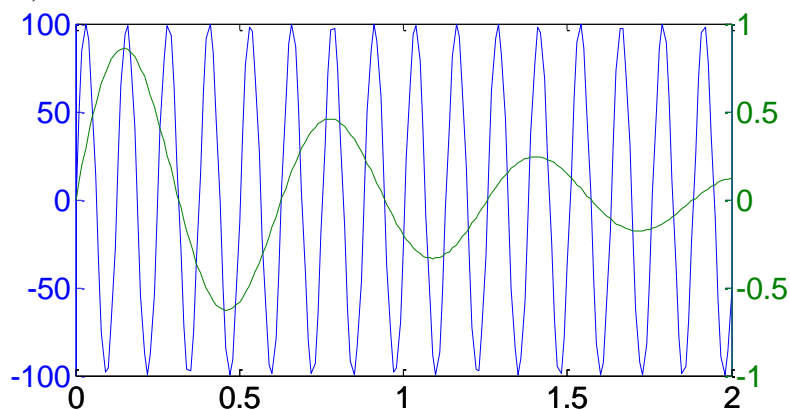


Рис. 4

На рис. 4 проявляются характерные особенности каждой из функций. Кроме того, на цветном мониторе видно, что цвет каждой из вертикальных осей повторяет цвет соответствующего графика. При черно-белом изображении графики различных функций отличаются лишь оттенками серого цвета.

Иногда график функции требуется снабдить заранее заданными отличительными признаками, например, цветом, определенным типом линии, расположить на этой линии специальные символы (маркеры). Для этого также применяется функция **plot(...)**, но более сложной конструкции.

*Пример 4.* Вычертить график функции  $y(t)=e^{-t}\cdot\sin(10\cdot t)$  в виде штриховой линии красного цвета, помеченной маркером типа «кружок». Аргумент  $t$  меняется в интервале  $[0,1]$  с шагом 0,03.

Эта задача решается следующим образом (рис. 5):



```

t=[0:0.03:1];
y=exp(-t).*sin(10*t);
plot(t,y,'ro--')

```

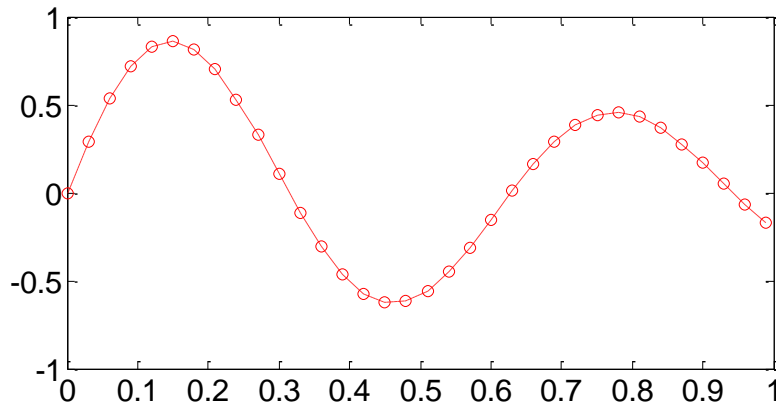


Рис. 5

Полезно сравнить структуру функции **plot(...)** в этом примере и в примере 1. Из сравнения следует, что в функцию **plot(...)** в настоящем примере включен дополнительный параметр, заключенный в апострофы. Этот параметр представляет собой последовательность символов, определяющих цвет линии (**r** – это красный цвет), тип маркера (**o** – это кружок) и стиль линии (двойной дефис, т. е. символ «--» соответствует штриховой линии). Заметим, что внутри апострофов между отдельными символами нет никаких разделительных знаков и, кроме того, последовательность этих символов упорядочена, а именно: цвет, тип маркера, стиль линии. Внутри апострофов могут располагаться один, два или три таких символа.

В табл. 1 содержатся данные, необходимые для определения свойств линий графиков.

Таблица 1

Цвет	Тип маркера	Стиль линии
1	2	3
y желтый	· точка	- сплошная
m розовый	o кружок	: пунктирная
c голубой	x крестик	-. штрих-пунктирная
r красный	+ знак плюс	-- штриховая
g зеленый	s квадрат	
b синий	d ромб	
w белый	v треугольник вершиной вниз	
k черный	^ треугольник вершиной вверх	
	< треугольник вершиной влево	
	> треугольник вершиной вправо	
	p пятиконечная звезда	
	h шестиконечная звезда	

Функция **plot(t,y,S)**, где под **S** понимается дополнительный параметр, заключенный в апострофы (например, **y+** (цвет – желтый, тип маркера – знак плюс, стиль линии – сплошная) или **cd**: (голубой, ромб, пунктирная) или другая комбинация символов из табл. 1), чаще всего применяется при вычерчивании нескольких графиков в одном окне. Поясним это на примере.

*Пример 5.* Вычертить графики трех функций  $y_1(t)=\sin(t)$ ,  $y_2(t)=\sin^2(t)$  и  $y_3(t)=\sin^3(t)$  при изменении **t** в пределах от  $-2\pi$  до  $2\pi$  с шагом  $0,1\pi$  (рис. 6).

```
t=[-2*pi:0.1*pi:2*pi];
y1=sin(t);
y2=sin(t).^2;
y3=sin(t).^3;
plot(t,y1,'-m',t,y2,'-.+r',t,y3,'-ok')
```

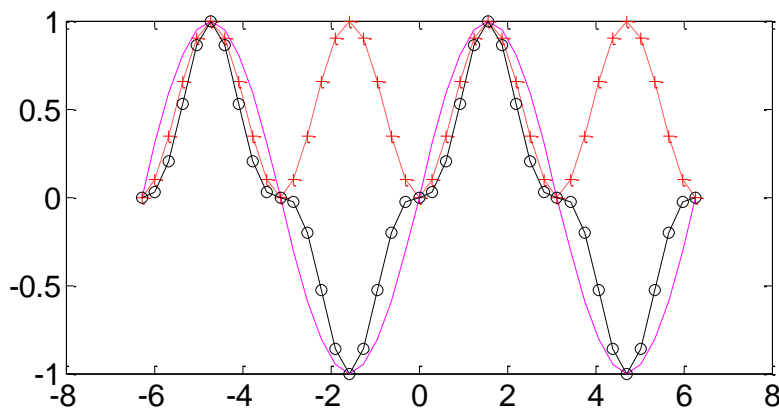


Рис. 6

В данном случае функция **plot(...)** содержит требования по выводу графиков трех функций: **y1**, **y2** и **y3**. При этом параметр **S** для функции **y1** состоит лишь из двух символов. В этом случае **S='-m'**. В то же время для функции **y2** параметр **S= '-.+r'**, а для функции **y3** – **S='-ok'**.

Вывод: при описании вида графиков может быть использовано различное количество символов из табл. 1, важно не нарушать последовательности расположения символов.

## 1.2. Специальные виды 2D-графиков

### 1.2.1. Представление функции в виде дискретных отсчетов

Представить графическую зависимость в виде дискретных отсчетов можно, пользуясь функцией **stem(...)**. В общем случае функция **stem(...)** имеет вид **stem(t,y,S)**, где **S** является дополнительным параметром, который используется так же, как и в функции **plot(...)**. Параметр **S** выбирается из табл. 1.

**stem(Y)** строит зависимость значений элементов вектора от номеров этих значений в виде дискретных отсчетов.

*Пример 6.* Построить график функции  $y(t)=e^{-t}\cdot\sin(10\cdot t)$ , где аргумент  $t$  меняется от 0 до 1 с шагом 0,02. Функцию  $y$  задать как вектор (рис. 7).

```
t=[0:0.02:1];
Y=exp(-t).*sin(10*t);
stem(Y)
```

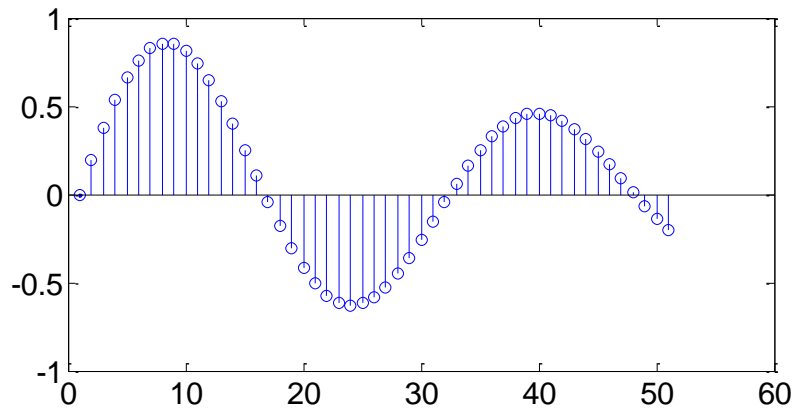


Рис. 7

Цвет и стиль линии, а также форму маркера на рис. 7 **MatLab** выбрал самостоятельно.

**stem(t,y,S)** строит в виде дискретных отсчетов график функции  $y$  с абсциссами  $t$ , где  $S$  – дополнительный параметр, аналогичный применяемому в функции **plot(...)**.

*Пример 7.* Построить график функции  $y(t)=e^{-t}\cdot\sin(10\cdot t)$ , где аргумент  $t$  меняется от 0 до 1 с шагом 0,02 (рис. 8). Дискретные отсчеты закрасить красным цветом, обозначить маркером в виде «\*» и вычертить штриховой линией.

```
t=[0:0.02:1];
Y=exp(-t).*sin(10*t);
stem(t,Y,'r*--')
```

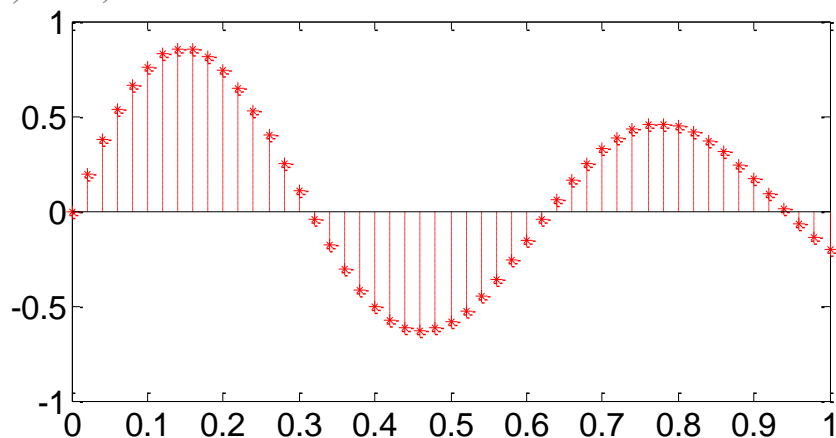


Рис. 8

Обратите внимание, на то, чем отличается на графиках рис. 7 и рис. 8 шкала оси абсцисс. На рис. 7 значения оси абсцисс представляют собой целочисленный ряд 0, 1, 2, ..., 50, т. е. соответствуют номерам элементов вектора  $Y$ , а на рис. 8 значения абсцисс представляют собой определенную на

интервале  $[0,1]$  и возрастающую с шагом 0,02 последовательность действительных чисел.

### 1.2.2. Лестничные графики

Лестничные графики представляют собой ступеньки с огибающей, заданной в виде функции  $y(t)$ . Они используются, например, для наглядного представления функции  $y(t)$ , представленной результатами ряда измерений ее значений. При этом в промежутках между измерениями значения функции считаются постоянными и равными величине последнего результата измерения.

Для построения лестничных графиков используется функция `stairs(...)`. Общий вид аргумента функции `stairs(...)` такой же, как и в функциях `plot(...)` и `stem(...)`. Правила использования дополнительного параметра **S** аналогичны правилам, применяемым для функций `plot(...)` и `stem(...)`.

*Пример 8.* Построить лестничный график функции  $y(t)=e^{0,1t}$ , где аргумент  $t$  меняется от 0 до 20 с шагом 1 (рис. 9). Описание дополнительного параметра **S** (см. табл. 1): цвет – зеленый, тип маркера – квадрат, стиль линии – пунктирная.

Программу для расчета значений функции  $y(t)$  и вывода требуемого графика (рис. 9) представим в двух вариантах:

Вариант 1  
`t=[0:20];`  
`y=exp(0.1*t);`  
`stairs(t,y,'gs:')`

Вариант 2  
`t=[0:20];`  
`stairs(t,exp(0.1*t),'gs:')`

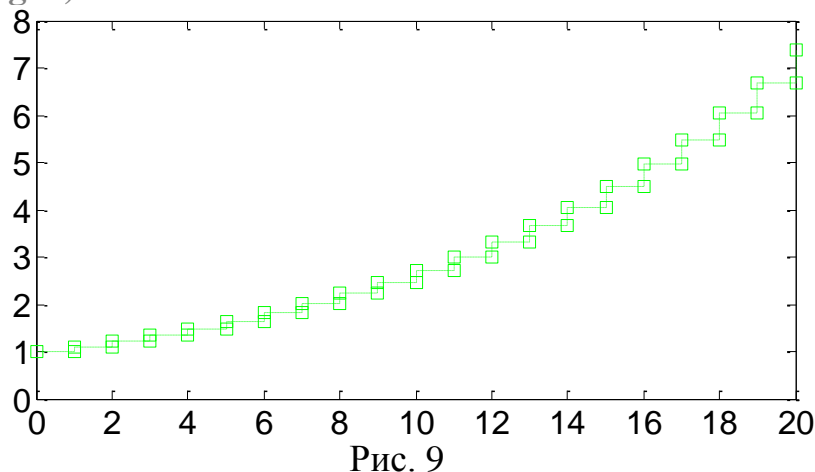


Рис. 9

Программа в варианте 2 на одну строку короче. **MatLab** может вычислять значения  $y(t)$  не только предварительно, но и непосредственно при использовании функции `stairs(...)`. Это оказывается верным также и в случаях применения функций `plot(...)`, `stem(...)` и функции `errorbar(...)`, которая рассматривается ниже.

### 1.2.3. Графики с указанием погрешности

Если при измерении значений функции имеется существенная погрешность, которая часто бывает случайной величиной, то для построения графика с учетом погрешности может быть использована функция **errorbar(...)**. Эта функция позволяет визуально оценить погрешность в каждой точке измерения путем представления ее в виде буквы I. При этом высота буквы I равна удвоенному значению погрешности измерения ординаты функции **y(t)** в каждой точке.

**errorbar(y,t,k,u)** строит график значений функции **y** в зависимости от значений аргумента **t** с указанием нижней **k** и верхней **u**. Значения **k** и **u** могут быть заданы в виде последовательности как детерминированных, так и случайных чисел.

**errorbar(y,t,E)** строит график значений функции **y** в зависимости от значений аргумента **t** с указанием границ погрешности в виде **[y-E,y+E]**.

**errorbar(...,S)** строит график значений функции **y** в зависимости от значений аргумента **t** по одному из указанных выше вариантов, но с применением дополнительного параметра **S**, который используется по тем же правилам, что и в функциях **plot(...)** и **stem(...)**.

*Пример 9.* С помощью функции **errorbar(y,t,E,S)** построить график  $y(t)=e^{0,05t}$ , где аргумент **t** меняется от -2,91 до 55,2 с шагом 2,1 (рис. 10). Параметр **E** является последовательностью случайных чисел, которая задается с помощью встроенных функций **MatLab**, т. е. **rand** и **size** (напомним, что **rand** генерирует последовательность равномерно распределенных случайных чисел, а **size** определяет размер массива сгенерированных чисел). Параметр **E** задать в виде **E=rand(size(4.7\*t))**. Описание дополнительного параметра **S**: цвет – зеленый, тип маркера – треугольник вершиной кверху, стиль линии – пунктирная (см. табл. 1).

```
t=[-2.91:2.1:55.2];  
y=exp(0.05*t);  
E=rand(size(4.7*t));  
errorbar(t,y,E,'g^:')
```

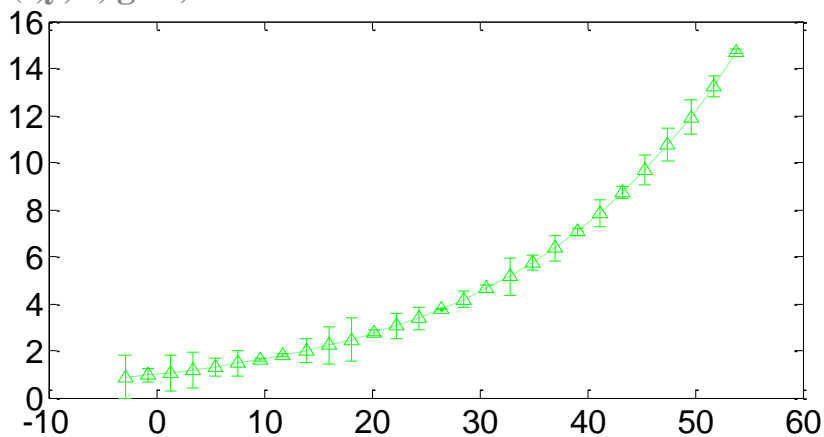


Рис. 10

### 1.2.4. Графики в логарифмическом и полулогарифмическом масштабах

Встречаются случаи, когда диапазон изменения функции настолько велик, что для ее графической визуализации приходится применять логарифмический масштаб. Для построения графиков в логарифмическом и полулогарифмическом масштабе служат следующие функции:

- loglog(...)** – логарифмический масштаб по обеим осям;
- semilogx(...)** – логарифмический масштаб по оси абсцисс;
- semilogy(...)** – логарифмический масштаб по оси ординат.

Аргументы этих функций (все, что стоит внутри скобок) формируются по тем же правилам, что и в функции **plot(...)**.

*Пример 10.* Построить в одном окне в полулогарифмическом масштабе с использованием функции **semilogx(...)** графики зависимостей **f(t)=t·log(t<sup>2</sup>)+1000** и **g(t)=10·t·sin(log(t<sup>2</sup>))** (рис. 11) при условии, что **t** меняется на интервале [10,1000] с шагом 1. График функции **g(t)** выделить красным цветом.

```
t=[10:1000];  
f=t.*log(t.^2)+1000;  
g=10*t.*sin(log(t.^2));  
semilogx(t,f,t,g,'r')
```

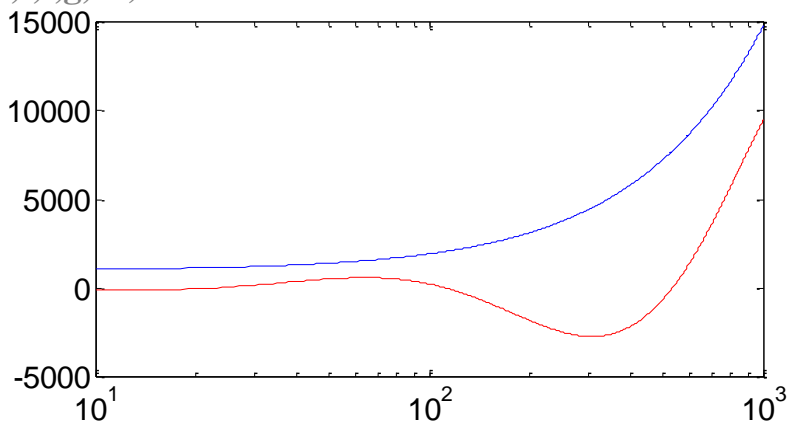


Рис. 11

### 1.3. Графики параметрических функций

Напомним, как образуется график параметрической функции.

Если имеются две функции **x(t)** и **y(t)**, то параметрическая функция задается в виде **y(x)**. Таким образом, для построения графика такой функции необходимо выполнить следующие действия:

- задать пределы и шаг изменения аргумента **t**;
- вычислить значения функций **x(t)** и **y(t)**;
- использовать функцию **plot(...)** в одном из ее вариантов.

*Пример 11.* Используя в качестве исходных функций  $x(t)=\sin(t)$  и  $y(t)=0,1 \cdot \cos(t)$ , построить график (рис. 12) параметрической функции  $y(x)$  при условии, что  $t$  изменяется на интервале  $[0, 2\pi]$  с шагом 0,01.

Вариант 1

```
t=[0:0.01:2*pi];
```

```
x=sin(t);
```

```
y=0.1*cos(t);
```

```
plot(x,y)
```

Вариант 2

```
t=[0:0.01:2*pi];
```

```
plot(sin(t), 0.1*cos(t))
```

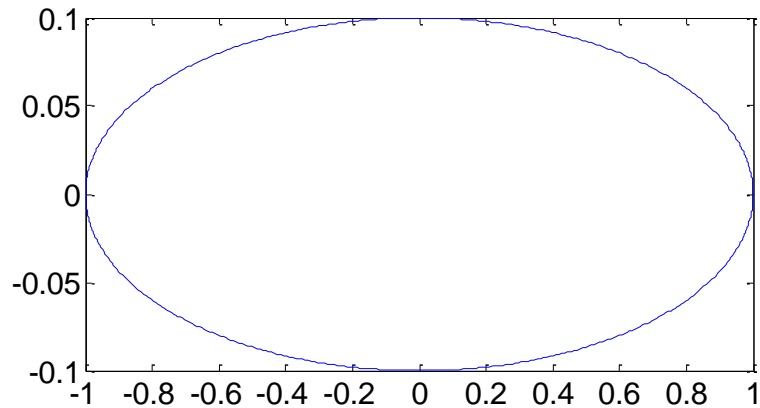


Рис. 12

На этом мы завершим рассмотрение 2D-графики.

Дополнительно отметим, что выше рассмотрены лишь основы графической визуализации на плоскости. На самом деле графические возможности среды **MatLab** значительно шире.

## 2. Трехмерная (3D) графика

Под 3D-графикой понимают построение графической зависимости функции двух переменных  $z(x,y)$ . График такой функции представляет собой изображение некоторой поверхности в трехмерном пространстве и строится с использованием аксонометрического метода.

Для графической визуализации функции двух переменных следует:

а) сформировать матрицу  $[x,y]$  с координатами узлов сетки на прямоугольной области определения функции. Матрица с координатами узлов сетки генерируется с помощью команды `meshgrid(x,y)`. Аргументами  $x$  и  $y$  являются векторы, элементы которых задают координаты узлов прямоугольной сетки. Если область определения функции квадрат, то в команде `meshgrid(...)` можно использовать один аргумент;

б) вычислить значения функции в узлах сетки;

в) использовать для вывода графика одну из функций **MatLab**;

г) используя команды оформления графика, нанести дополнительную информацию.

Как и прежде, главным условием для отображения графических зависимостей функции  $z(x,y)$  является понимание того, как в среде **MatLab** осуществляются поэлементные операции с векторами.

### 2.1. Линейчатые поверхности

Для построения линейчатых поверхностей используются функции `plot3(...)` и `contour3(...)`.

*Пример 12.* Построить график функции  $z(x,y)=x+y^3$  (рис. 13), где переменные  $x$  и  $y$  изменяются на интервале  $[-8,8]$  с шагом 0,5. Для построения графика использовать непрерывные линии черного цвета.

```
[x,y]=meshgrid(-8:0.5:8);
```

```
z=x+y.^3;
```

```
plot3(x,y,z,'k')
```

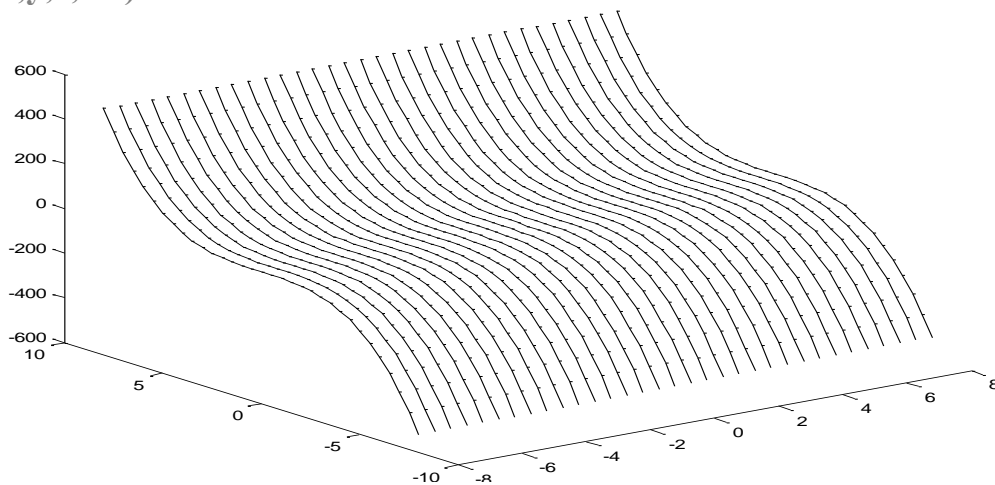


Рис. 13



Обсудим построенный график.

1. Как видно из условия задачи, область определения функции  $z$  представляет собой квадрат. В этом случае в команде **meshgrid(...)** достаточно указать пределы и шаг изменения лишь одного из аргументов функции.

2. Команда **meshgrid(...)** только подготавливает матрицу  $[x,y]$ , но не выводит ее на экран монитора.

3. В функции **plot3(...)** присутствует дополнительный параметр, заключенный в апострофы.

4. Все точки каждой из линий, с помощью которых построен график, находятся на одинаковом расстоянии от плоскости  $ZOY$ . На языке начертательной геометрии они носят название линий профильного уровня.

Следует отметить, что правила применения функции **plot3(...)** аналогичны правилам применения функции **plot(...)** в 2D-графике.

Используем теперь для построения линейчатой поверхности функцию **contour3(...)**.

Пример 13. С использованием функции **contour3(...)** построить линейчатую поверхность функции  $z=x^2+y^2$  (рис. 14). Переменные  $x$  и  $y$  изменяются на интервале  $[-3,3]$  с шагом  $0,1$ . Количество линий на графике – 40.

```
[x,y]=meshgrid(-3:0.1:3);
```

```
z=x.^2+y.^2;
```

```
contour3(x,y,z,40)
```

```
colormap(bone)
```

```
colorbar
```

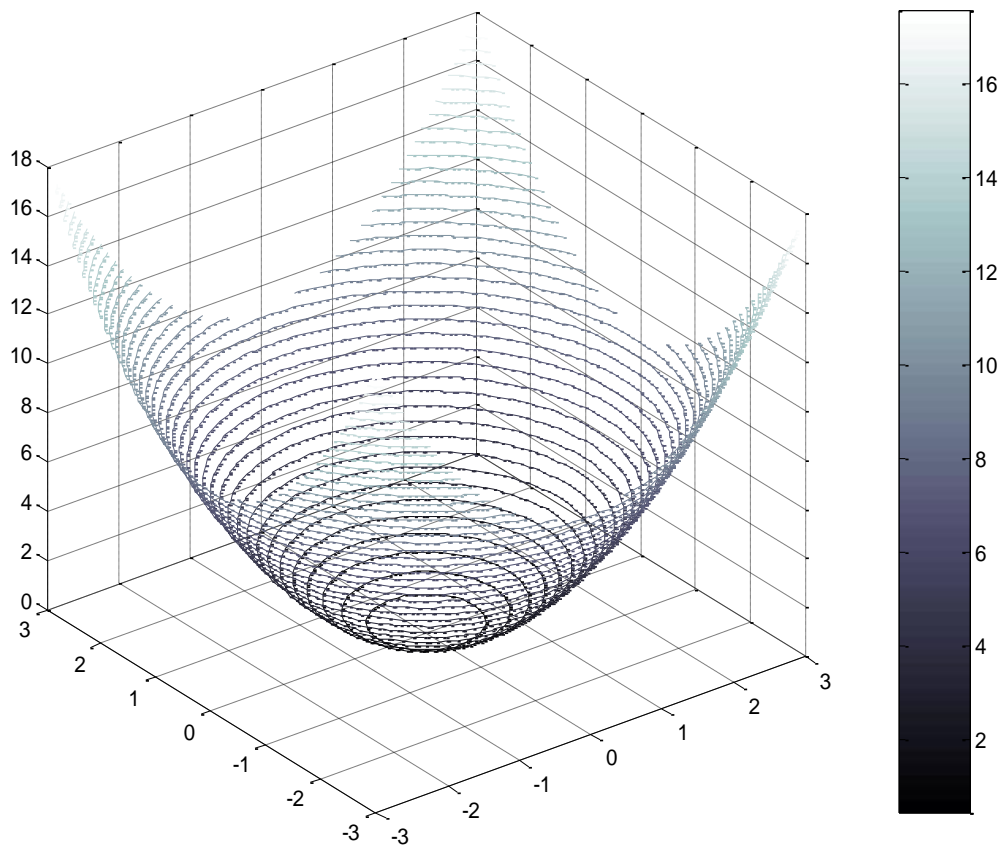


Рис. 14

Из рис. 14 видно, что линии, с помощью которых построена поверхность, параллельны плоскости **XOY**. На языке начертательной геометрии они носят название линий горизонтального уровня.

Использование команд **colormap** и **colorbar** в приведенном тексте программы будет пояснено ниже.

Функция **contour3(...)** может быть использована без указания количества линий горизонтального уровня. В этом случае **MatLab** самостоятельно определяет их количество.

## 2.2. Каркасные поверхности

Команда **plot3(...)** может быть использована для построения так называемой каркасной поверхности, которая представляет собой каркас поверхности геометрического тела, похожий на каркас, сплетенный из проволоки.

*Пример 14.* Построить график каркасной поверхности функции  $z(x,y)=x+y^3$  (рис. 15), где переменные **x** и **y** изменяются на интервале  $[-8,8]$  с шагом 1. Для построения графика использовать непрерывные линии. Линии, расположенные параллельно плоскости **ZOY**, закрасить в черный цвет, а линии, параллельные плоскости **ZOX** – в красный цвет. На плоскостях **XOY**, **ZOX**, **ZOY** сформировать сетку.

```
[x,y]=meshgrid(-8:8);  
z=x+y.^3;  
plot3(x,y,z,'k',y,x,z,'r')  
grid on
```

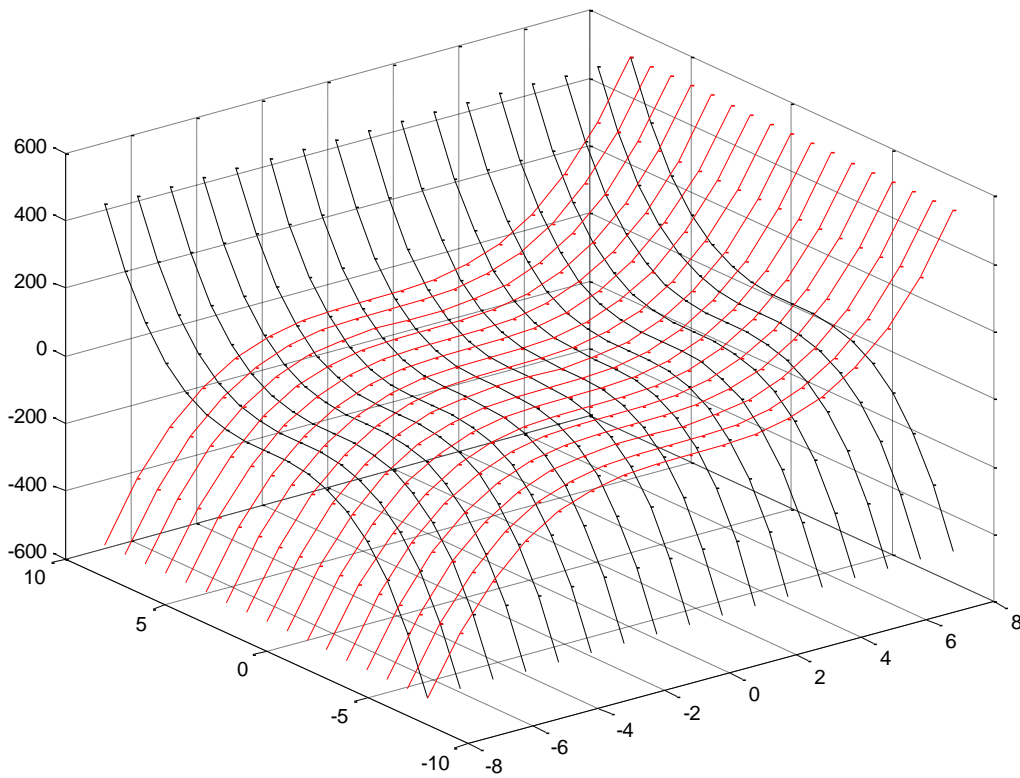


Рис. 15

Как видно из текста программы, для формирования каркасной поверхности в функции **plot3(...)** использованы два аргумента с дополнительными параметрами. При этом во втором аргументе переменные **x** и **y** переставлены местами. Сетка, как и в 2D-графике задается командой **grid on**.

В данном примере для построения каркасной поверхности была использована функция **plot3(...)**. Однако для построения таких поверхностей в **MatLab** имеются и другие функции, например, **mesh(...)** и **surf(...)**.

*Пример 15.* С помощью функции **mesh(...)** построить каркасную поверхность (рис. 16), заданную функцией  $z = \sin(x)/(x^2 + y^2 + 0.3)$ , где переменные **x** и **y** изменяются на интервале  $[-3, 3]$  с шагом 0,1.

```
[x,y]=meshgrid(-3:0.1:3);  
z=sin(x)./(x.^2+y.^2+0.3);  
mesh(x,y,z)
```

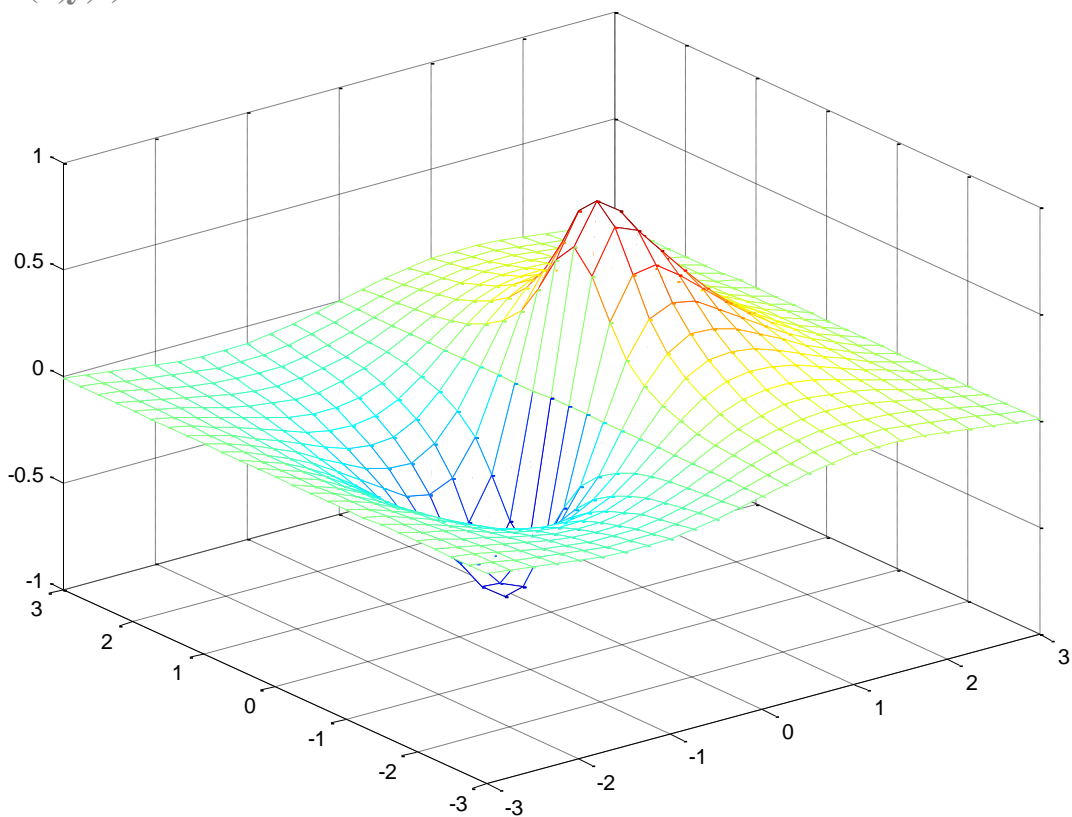


Рис. 16

На экране монитора можно видеть, что цвет линий каркасной поверхности зависит от величины значений функции **z**. Кроме того, функция **mesh(...)** предполагает вывод только видимой части поверхности.

### 2.2.1. Прозрачная каркасная поверхность

Для того, чтобы сделать каркасную поверхность «прозрачной» (рис. 17), следует использовать команду **hidden off**. Предыдущая программа в этом случае имеет следующий вид:

```
[x,y]=meshgrid(-3:0.1:3);  
z=sin(x)./(x.^2+y.^2+0.3);  
mesh(x,y,z)  
hidden off
```

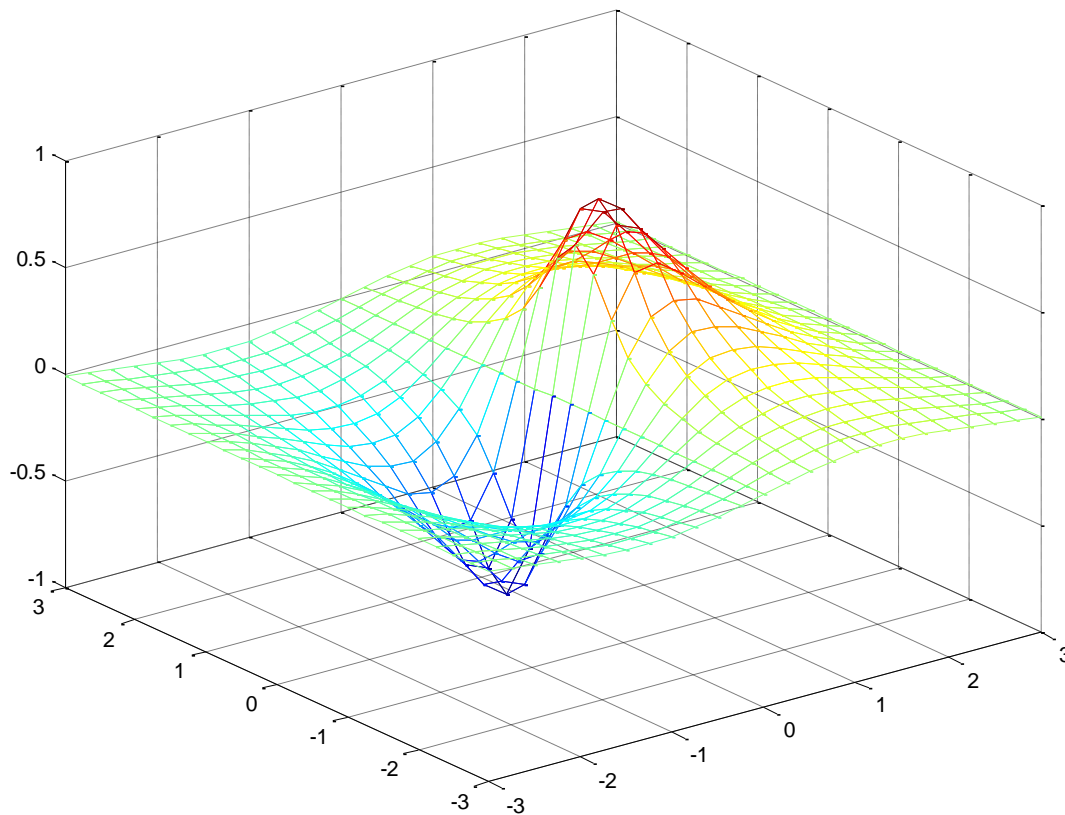


Рис. 17

Команда **hidden on** убирает невидимую часть поверхности, возвращая графику прежний вид.

### 2.2.2. Каркасная поверхность, залитая цветом

В отличие от функции **mesh(...)**, применение функции **surf(...)** позволяет окрасить не только линии графика, но и залить определенным цветом каждую клетку поверхности. Цвет зависит от значений функции в точках, соответствующих углам клетки.

Результат замены функции **mesh(x,y,z)** в примере 15 на функцию **surf(x,y,z)** представлен на рис. 18.

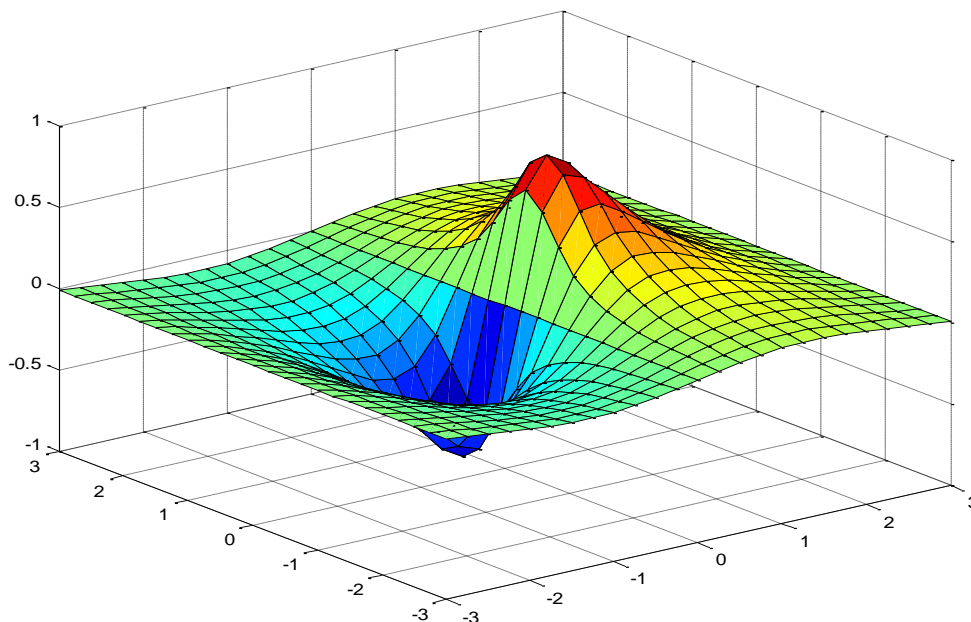


Рис. 18

Трехмерные графики, изображенные на рис. 16-18 позволяют наглядно представить форму поверхности, однако по ним трудно судить о значениях двухмерной функции. Для того, чтобы частично устранить этот недостаток, можно после функции **surf(x,y,z)** применить команду **colorbar**, которая выводит рядом с графиком столбик, устанавливающий соответствие между цветом поверхности и величиной функции. Рис. 19 иллюстрирует получившийся результат.

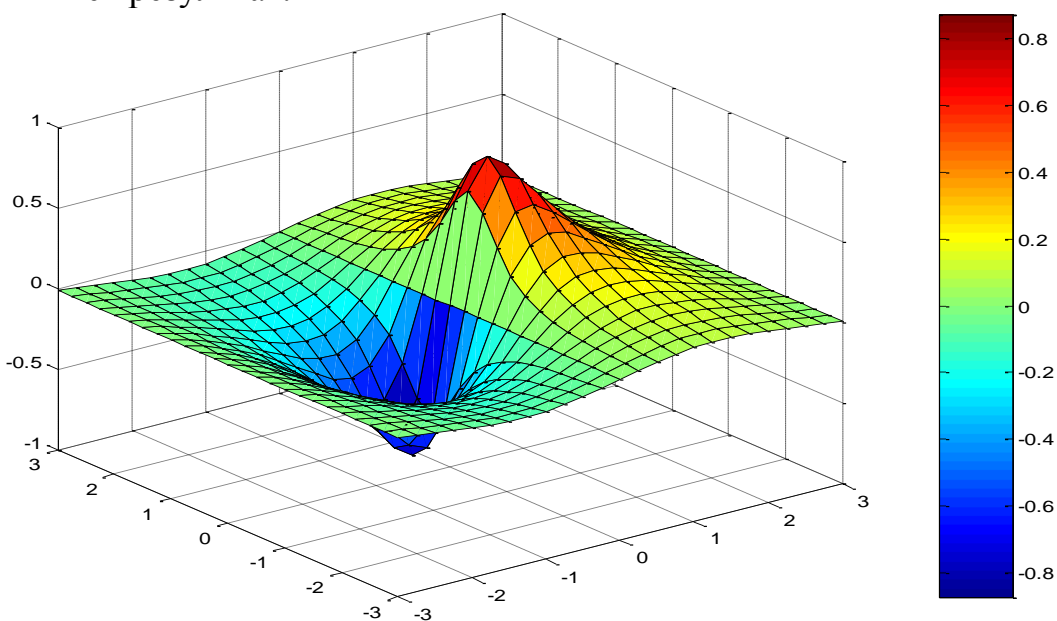


Рис. 19

Команду **colorbar** можно использовать в сочетании со всеми функциями **MatLab**, строящими трехмерные объекты.

Применение после функции **surf(x,y,z)** команды **shading flat** позволяет убрать каркасные линии (рис. 20), а использование команды **shading interp** – получить поверхность, плавно залитую цветом, зависящим от значений функции.

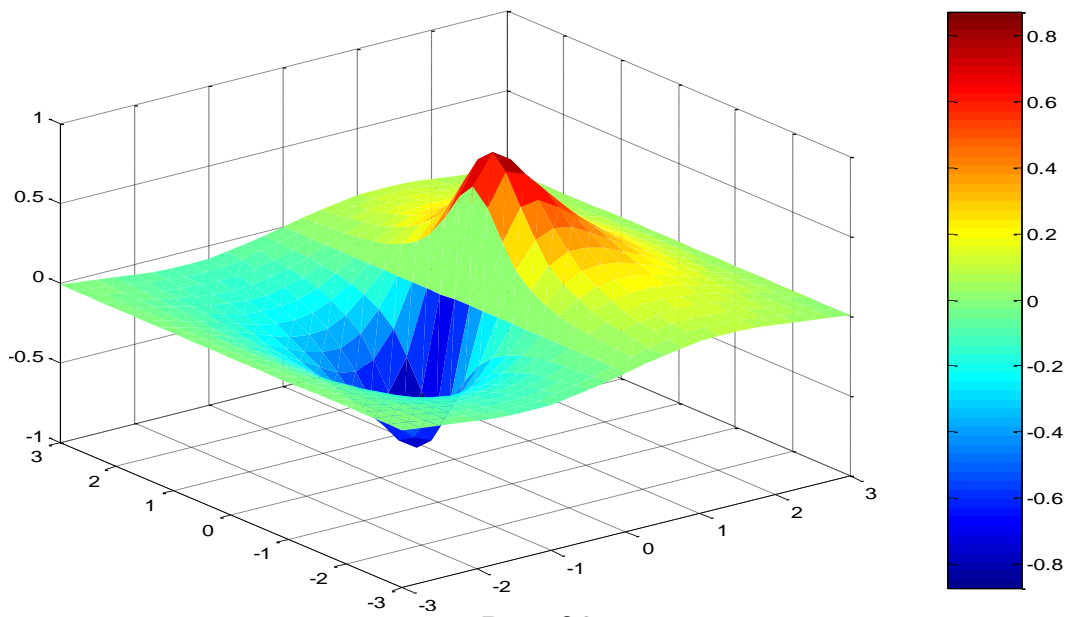


Рис. 20

### 2.3. Контурные графики

Контурные графики, как правило, представляют собой некоторый набор линий на плоскости. Каждая из линий является горизонтальной проекцией линии пересечения заданной поверхности с плоскостью горизонтального уровня.

**MatLab** позволяет получить различные типы контурных графиков при помощи функций `contour(...)` и `contourf(...)`.

*Пример 16.* Построить контурный график функции  $z=x^2+y^2$  (рис. 21) из двадцати линий. Переменные  $x$  и  $y$  изменяются на интервале  $[-3,3]$  с шагом  $0,1$ .

```
[x,y]=meshgrid(-3:0.1:3);
z=x.^2+y.^2;
contour(x,y,z,20)
colorbar
```

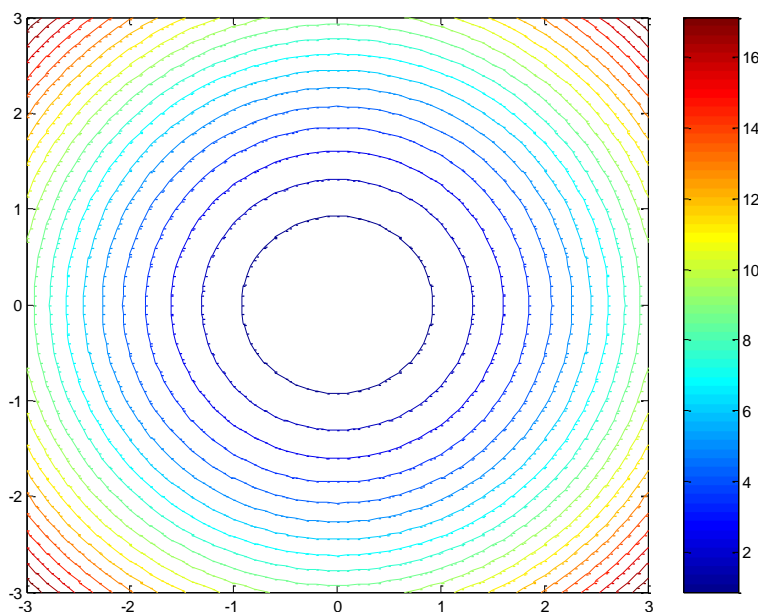


Рис. 21

Такой контурный график дает весьма приблизительное представление о том, какое значение функции соответствует той или иной линии уровня.

**MatLab** позволяет нанести на каждую линию уровня значение, которое принимает на ней исследуемая функция. Для этого необходимо использовать функцию **clabel(...)**.

Функция **clabel(...)** вызывается с двумя аргументами: матрицей, содержащей информацию о линиях уровня, и указателем на график, где следует нанести числовое значение функции.

Пользователю не обязательно самому создавать аргументы **clabel(...)**. Для этого функцию **contour(...)** следует вызвать с двумя выходными аргументами.

Поясним это на примере.

*Пример 17.* Построить контурный график функции  $z=x^2+y^2$  (рис. 22), состоящий из десяти оцифрованных линий уровня. Переменные  $x$  и  $y$  изменяются на интервале  $[-3,3]$  с шагом  $0,1$ .

```
[x,y]=meshgrid(-3:0.1:3);  
z=x.^2+y.^2;  
[C,H]=contour(x,y,z,10);  
clabel(C,H)  
grid on
```

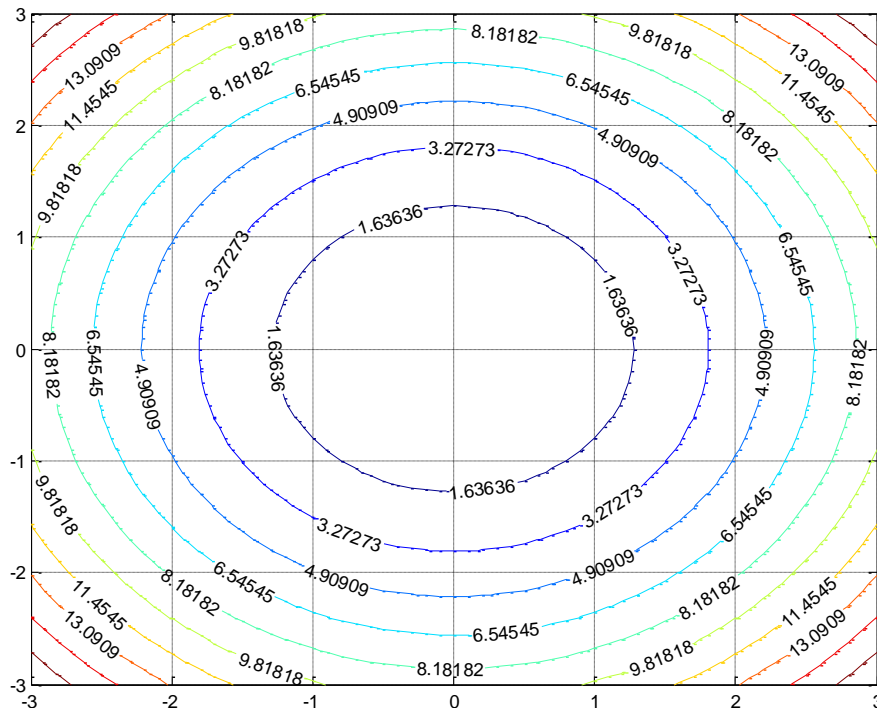


Рис. 22

**MatLab** предоставляет возможность оцифровки линий уровня вручную. Для этого вместо функции **clabel(C,H)** необходимо использовать функцию **clabel(C,H,'manual')**. Такая замена в данном примере приводит к тому, что на экран монитора выводится контурный график без оцифровки линий уровня, но с перекрестьем, которое можно двигать по изображению с помощью мыши. Расположив перекрестье на требуемой линии уровня и нажав левую или правую кнопку мыши, можно получить значение функции, соответствующее выбранной линии. Для того, чтобы завершить оцифровку линий уровня,

перекрестье следует поместить за пределы графика и нажать любую кнопку мыши.

На рис. 23 представлен результат оцифровки с помощью функции `clabel(C,H,'manual')` четырех линий уровня, рассчитанных применительно к условиям примера 17.

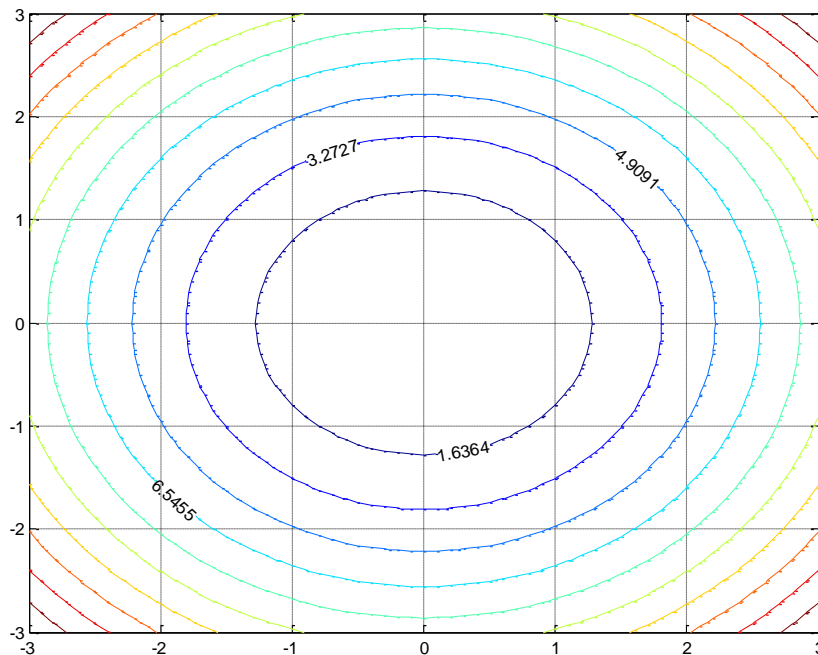


Рис. 23

**MatLab** позволяет строить линии уровня не только на плоскости, но и на графике двумерной функции  $z(x,y)$ . Для этого применяется функция `surfz(x,y,z)`, правила использования которой аналогичны правилам для функции `surf(x,y,z)`.

Если в примере 15 для вывода графика сделать замену функции `mesh(x,y,z)` на функцию `surfz(x,y,z)`, то получим результат, показанный на рис. 24.

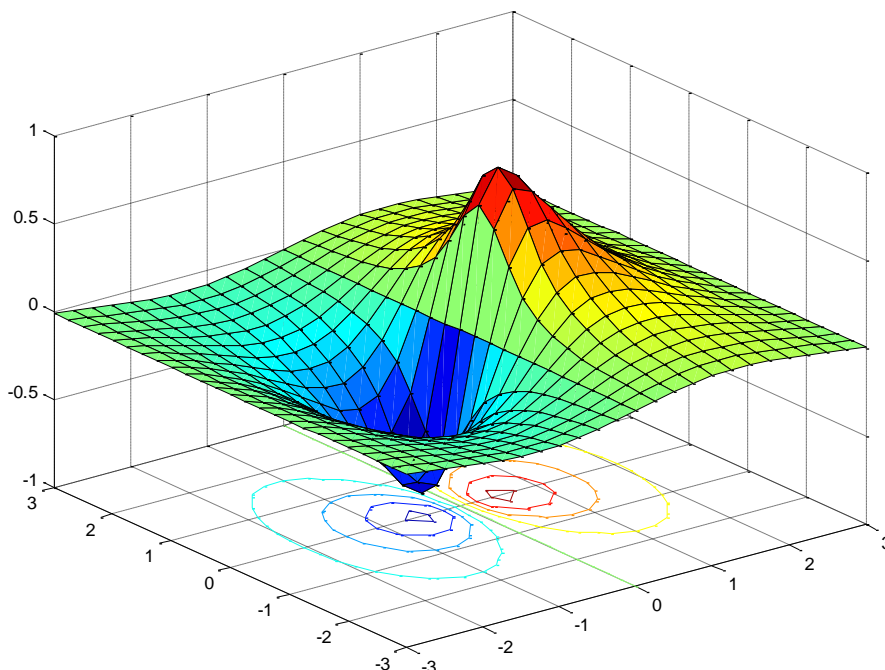


Рис. 24



## 2.4. Сплошная освещенная поверхность

Наиболее реалистичны трехмерные графики поверхностей, в которых имитируется освещение их от некоторого точечного источника, расположенного в определенном месте системы координат. Для получения таких графиков используется функция **surfcl(x,y,z)**.

Заменяем в примере 15 функцию **mesh(x,y,z)** на функцию **surfcl(x,y,z)** и добавим команду **colormap (gray)**, которая задает окраску тонами серого цвета, а также известные нам команды **shading interp** и **colorbar**.

```
[x,y]=meshgrid(-3:0.25:3);  
z=sin(x)./(x.^2+y.^2+0.3);  
surfcl(x,y,z)  
colormap (gray)  
shading interp  
colorbar
```

В результате получим график, представленный на рис. 25.

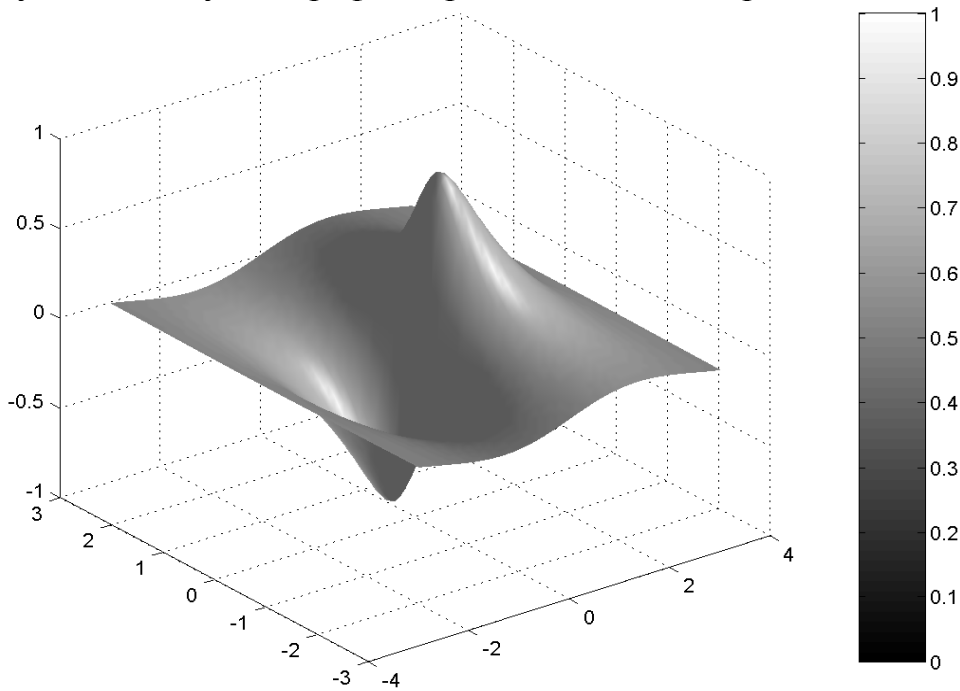


Рис. 25

### 3. Особенности работы в MatLab

#### 3.1. Работа с М-файлами

М-файлы – это обыкновенные текстовые файлы (с расширением **.m**), содержащие команды системы **MatLab**. Подготовленный и записанный на жесткий диск компьютера М-файл становится частью системы и его можно вызывать как из командной строки, так и из другого М-файла. Большая часть обширного набора готовых средств системы **MatLab** – это внешние расширения в виде М-файлов. Благодаря текстовому формату М-файлов пользователь может ввести в систему любую новую команду, оператор или функцию и легко приспособить ее к решению нужных классов задач.

Различают два типа М-файлов: файл-программы (Script M-Files) и файл-функции (Function M-Files).

Для создания и редактирования М-файлов предназначен встроенный модуль **Editor** (Редактор), который можно запустить командой **edit** (для редактирования нового файла). Для запуска модуля **Editor** (Редактор) можно также использовать меню **File** (Файл) или два крайних слева значка на панели инструментов как для создания нового файла, так и для открытия уже существующего.

#### 3.2. Файл-программы

Файл-программы (их также называют *сценариями* или *скриптами*) являются простейшим типом М-файлов. Они содержат последовательность команд без входных и выходных параметров.

В качестве примера составим файл-программу нахождения длины  $d$  радиус-вектора точки  $(x;y;z)$  трехмерного пространства с двумя фиксированными координатами  $y = 3, z = 4$ . В главном меню **MatLab** выберем команду **File => New => M-file** (Файл => Новый => М-файл), в результате чего раскроется окно редактора М-файлов (рис. 26).

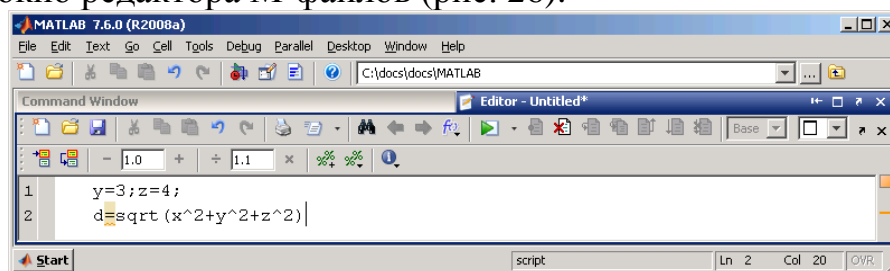


Рис. 26

В этом окне введем следующие строки:

```
x=2;y=3;z=4;
```

```
d=sqrt(x^2+y^2+z^2)
```

Для того, чтобы выполнить эту файл-программу можно с помощью мыши выделить в окне сценарий программы, скопировать и вставить его в командное

окно. После нажатия на клавиатуре кнопки **Enter** файл-программа выполняется и в командном окне выводится результат:

```
d =  
5.3852
```

Обратим внимание, что строка, задающая  $d$ , не завершена точкой с запятой, поэтому результат работы этой строки будет отображаться в командном окне.

Сохраним введенный файл-сценарий в текущем каталоге под именем, например, **myprog.m**. Для этого в окне редактора М-файлов выберем команду **File => Save As** (Файл => Сохранить как). В появившемся диалоговом окне **Save file as** (Сохранить файл как) раскроется подкаталог **work** основного каталога, который **MatLab** по умолчанию определяет в качестве текущего рабочего каталога **Current Directory**. В поле **File name** (Имя файла) введем имя **myprog** вместо отобразившегося по умолчанию имени **Untitled.m** и щелкнем по кнопке **Save** (Сохранить). Расширение **.m** вводить не следует, система **MatLab** добавит его автоматически.

Выполнить сохраненный файл-сценарий можно одним из следующих способов. Первый способ состоит из следующих действий:

1) в редакторе М-файлов выбрать команду **Debug => Run** (Отладка => Запуск);

2) набрать в командной строке имя М-файла (без расширения) и на клавиатуре нажать **Enter**, т. е. выполнить сценарий как команду **MatLab**.

Выполним сценарий вторым способом:

```
>> x=2;  
>> myprog  
d =  
5.3852
```

Ответом является длина 5,3852 радиус-вектора точки (2;3;4).

### 3.3. Встроенные функции

При практическом освоении **MatLab** необходимо использовать встроенные основные элементарные математические функции, которые приведены в табл. 2.

Таблица 2

Основные элементарные функции

<b>exp</b>	экспонента	<b>abs</b>	модуль
<b>log</b>	натуральный логарифм	<b>asin</b>	арксинус
<b>log10</b>	десятичный логарифм	<b>acos</b>	арккосинус
<b>sqrt</b>	квадратный корень	<b>atan</b>	арктангенс
<b>sin</b>	синус	<b>sinh</b>	гиперболический синус
<b>cos</b>	косинус	<b>cosh</b>	гиперболический косинус
<b>tan</b>	тангенс	<b>tanh</b>	гиперболический тангенс

<b>cot</b>	котангенс	<b>asinh</b>	гиперболический арксинус
<b>sec</b>	секанс	<b>acosh</b>	гиперболический арккосинус
<b>csc</b>	косеканс	<b>atanh</b>	гиперболический арктангенс

Аргументами элементарных функций могут быть действительные либо комплексные числа, а также массивы. Если в качестве аргумента функции задан массив, результат представляет собой массив, полученный поэлементным вычислением функции для соответствующих элементов исходного массива. Аргументы тригонометрических функций задаются в радианах. Обратные к ним функции возвращают результат также в радианах.

Примеры:

```
>> acosh(2)
```

```
ans =
```

```
1.3170
```

```
>> sin(1+i)
```

```
ans =
```

```
1.2985 + 0.6350i
```

```
>> exp(0:3)
```

```
ans =
```

```
1.0000 2.7183 7.3891 20.0855
```

Справочная информация по встроенным элементарным функциям содержится в разделе **elfun** справочной системы **MatLab**. Информацию из раздела можно получить, набрав в командной строке команду **doc elfun**.

## 4. Представление графических моделей

### 4.1. Оформление графиков

Для того чтобы графики было удобно использовать в дальнейшей работе, их необходимо соответствующим образом оформить.

На график можно нанести:

- координатную сетку (команда **grid on**);
- подписи к осям абсцисс и ординат (команды **xlabel(...)**, **ylabel(...)**);
- заголовок (команда **title(...)**);
- легенду (команда **legend(...)**).

Перечисленные команды можно использовать для графиков в линейном, логарифмическом или полупологарифмическом масштабах. Если в одном окне отображается несколько графиков, то с помощью команды **legend** можно поместить дополнительную информацию о каждом из графиков.

*Пример 18.* Вычертить графики двух функций  $y_2(t)=\sin(t)$  и  $y_1(t)=\sin^3(t)$  при изменении  $t$  в пределах от  $-2\pi$  до  $2\pi$  с шагом  $0,1\pi$  (рис. 27).

```
t=[-2*pi:0.1*pi:2*pi];  
y1=sin(t);  
y2=sin(t).^3;  
plot(t,y1,'-m',t,y2,'-ok')  
grid on  
title('Периодические функции')  
xlabel('Аргумент -- время(сек)')  
ylabel('Амплитуда (В)')  
legend('10 мая','11 мая',4)
```

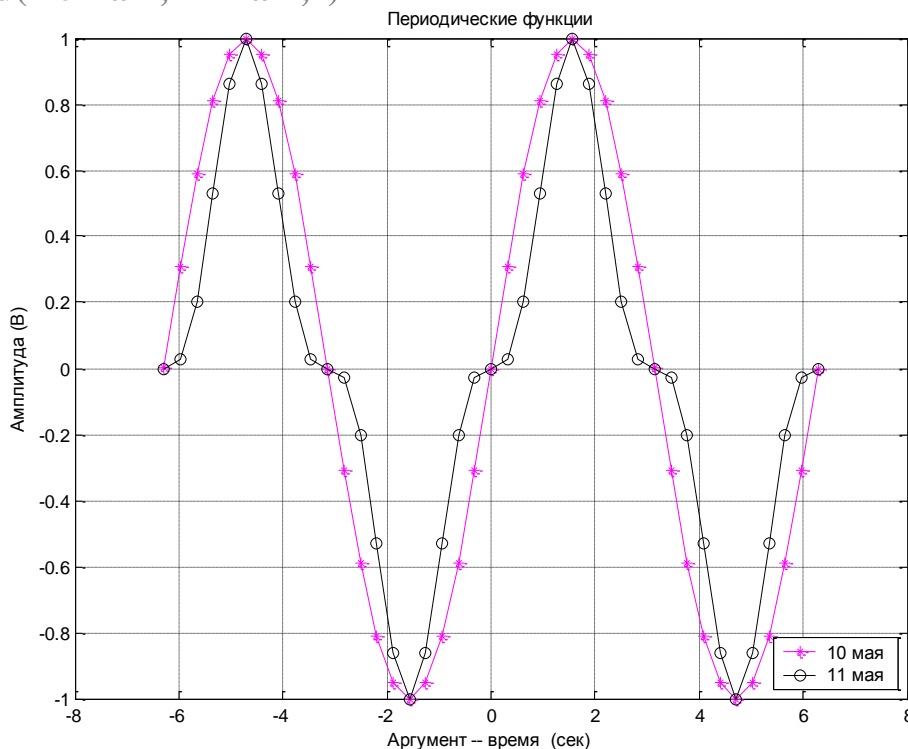


Рис. 27

При составлении легенды следует учитывать, что порядок следования и количество аргументов команды **legend(...)**, заключенных в апострофы, должны быть согласованы с конкретными линиями графика. В данном примере команда **legend('10 мая','11 мая')** и функция **plot(t,y1,'-m',t,y2,'-ok')** согласованы как по порядку следования функций (**y1** и **y2**), так и по их количеству.

Последним дополнительным аргументом **legend(...)** может быть положение легенды в графическом окне, а именно:

- а) -1 – вне графика в правом верхнем углу;
- б) 0 – выбирается лучшее положение в пределах графика так, чтобы как можно меньше перекрывать сами графики;
- в) 1 – в правом верхнем углу графика (это положение используется по умолчанию);
- г) 2, 3, 4 – соответственно, в верхнем левом углу, в нижнем левом углу и в нижнем правом углу графика.

Чтобы перенести легенду, следует нажать левую кнопку мыши и, не отпуская ее, перетащить легенду в нужное место.

Часто бывает удобным снабдить тот или иной график соответствующей надписью. В этом случае можно воспользоваться командой **gtext('...')**, где между апострофами располагают необходимую надпись. Применение команды **gtext('...')** приводит к выводу строки текста, начало которой помещается в том месте, где находится перекрестье маркера.

*Пример 19.* В графике функции  $y_1(t)=\sin(t)$  (пример 18) поместить текст в виде строки «График функции  $\sin(t)$ ». Начало надписи (координаты перекрестья) должно быть расположено (рис. 28) примерно в точке с координатами (3,0,2).

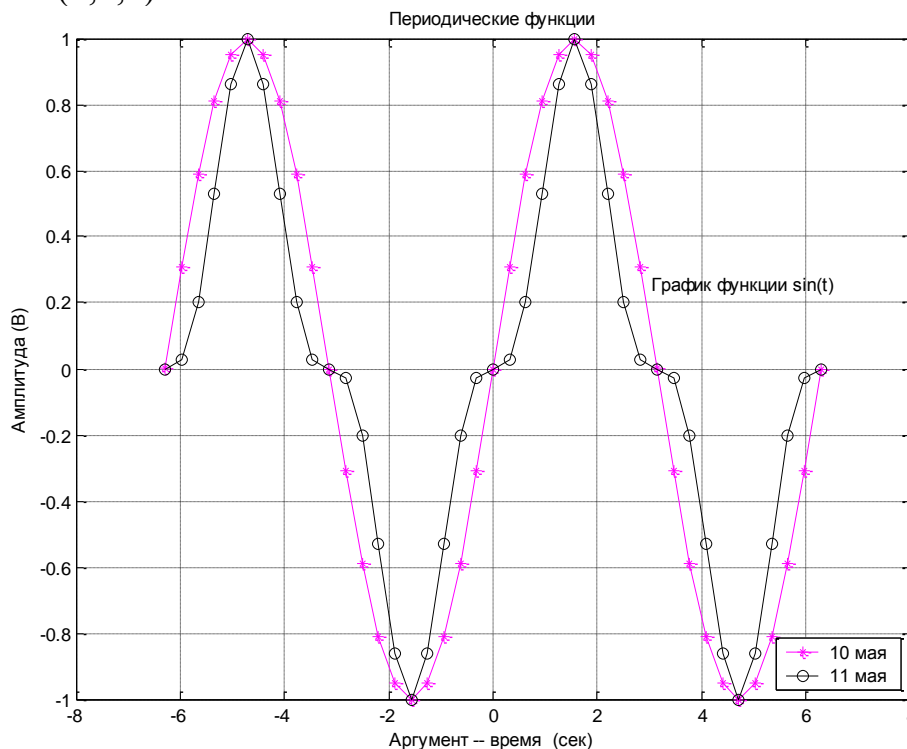


Рис. 28

## 4.2. Сохранение и экспорт графиков

Построенный 2D- или 3D-график можно сохранить в среде **MatLab** или экспортировать (перенести) его в любое другое приложение **Windows**.

Рассмотрим сохранение и экспорт графического файла с помощью построенного в среде **MatLab** (см. рис. 28).

Для сохранения рис. 28 (или любого другого рисунка) в среде **MatLab** необходимо выполнить следующие действия:

1) в падающем меню **File** графического окна выбрать строку **Save** или **Save As...**;

2) в появившемся окне указать, в какой папке и под каким именем пользователь желает сохранить файл.

Для экспорта графика необходимо выполнить следующие действия:

1) установить, с какими расширениями работает приложение **Windows**, в котором пользователь желает сохранить графический файл;

2) в падающем меню **File** графического окна выбрать строку **Export**;

3) в открывшемся окне выбрать папку, задать название графического файла и его тип (расширение). При этом тип файла должен быть таким, чтобы с ним могло работать выбранное приложение **Windows**.

Для тренировки экспортируйте рис. 27 в наиболее распространенное приложение **Windows** – текстовый редактор **Word**. С этой целью в **Word** желательно заранее сформировать нужную папку. При экспорте в **Word** рекомендуется использовать расширения [**\*.tif**] или [**\*.bmp**].

## 4.3. Анимация

В ряде случаев полезно наблюдать за тем, как строится график функции  $y$  с аргументом  $x$  по мере вычисления ее значений. При этом мы можем следить за динамикой некоторого процесса (т. е. за его развитием), а не наблюдать статичную (застывшую итоговую) картинку, как было ранее. С этой целью используется функция **comet(x,y)**, которая позволяет получить анимированный график, на котором кружок, обозначающий точку, перемещается, оставляя за собой след в виде линии – траектории движения. График похож на летящую комету с хвостом. Скоростью движения кружка можно управлять, манипулируя шагом изменения аргумента. Длину хвоста можно задать в виде дополнительного параметра **S**. В этом случае функция **comet(...)** имеет вид **comet(x,y,z)**, где **z** определяет длину хвоста. По умолчанию значение **z** принимается равным 0,1.

*Пример 20.* С использованием функции **comet(x,y)** построить траекторию движения фиксированной точки на окружности, катящейся по прямой (график циклоиды). Циклоида задается в виде параметрической функции, определяемой исходными уравнениями  $x(t)=t\sin(t)$ ,  $y(t)=1-\cos(t)$ . Аргумент  $t$  изменяется на интервале  $[0,10\pi]$  с шагом 0,001.

Вариант 1

```
t=[0:0.001:10*pi];
```

```
x=t-sin(t);
```

```
y=1-cos(t);
```

```
comet(x,y)
```

Вариант 2

```
t=[0:0.001:10*pi];
```

```
comet(t-sin(t),1-cos(t))
```

Мы не приводим графика циклоиды потому, что **MatLab** устроен так, что анимированный график не может быть запомнен в буфере и восстановлен. При изменении размеров анимированного графика, при его повороте, увеличении или уменьшении, его запоминании с последующим восстановлением или других действиях с графиком траектория движения точки пропадает.



## Задания для закрепления теоретического материала (2D-графика)

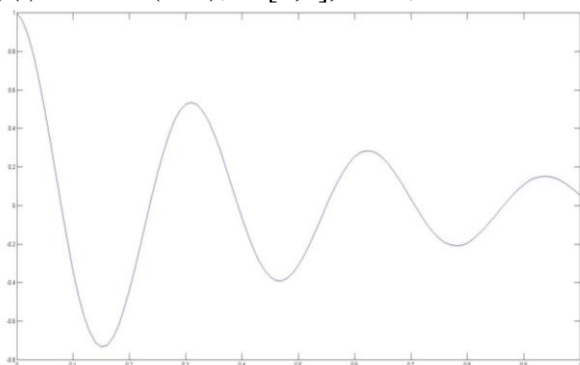
### Задача 1.1

Построить график функции с применением оператора **plot(t,y)**.

#### ВАРИАНТЫ РЕШЕНИЙ

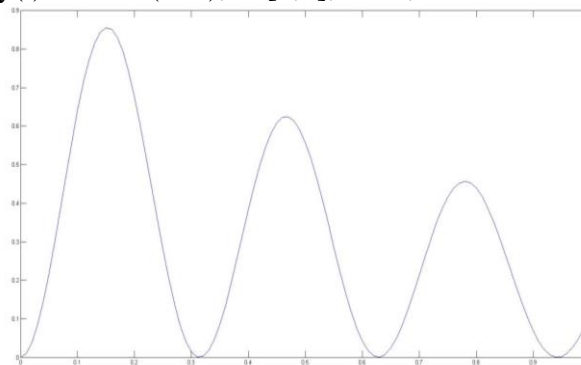
##### Вариант 1

$$y(t)=e^{-2t}\cdot\cos(20\cdot t), t\in[0,1], \Delta t=0,01$$



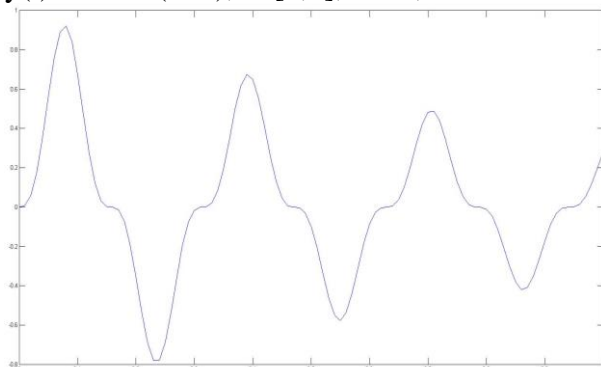
##### Вариант 2

$$y(t)=e^{-t}\cdot\sin^2(10\cdot t), t\in[0,1], \Delta t=0,01$$



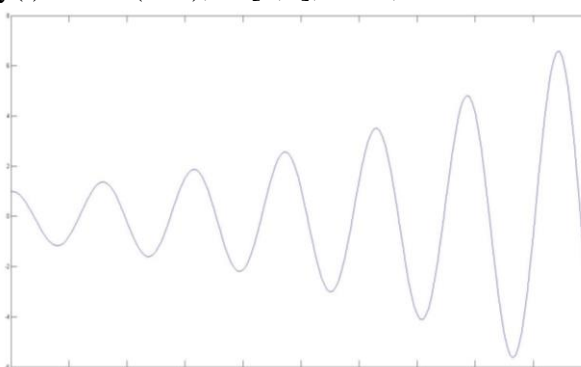
##### Вариант 3

$$y(t)=e^{-t}\cdot\sin^3(20\cdot t), t\in[0,1], \Delta t=0,01$$



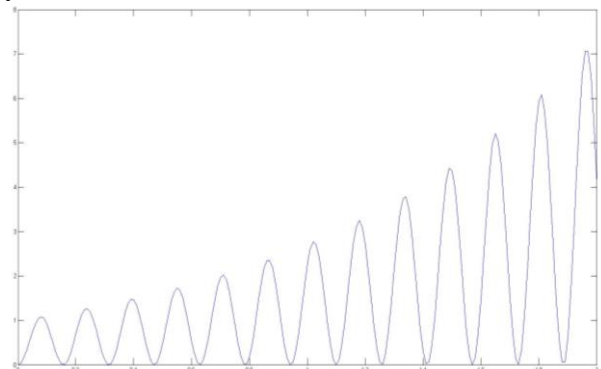
##### Вариант 4

$$y(t)=e^t\cdot\cos(20\cdot t), t\in[0,2], \Delta t=0,01$$



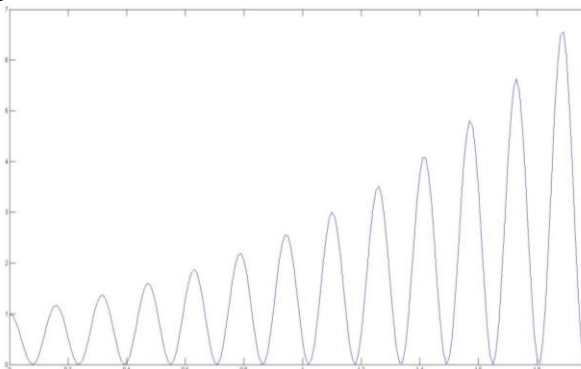
##### Вариант 5

$$y(t)=e^t\cdot\sin^2(20\cdot t), t\in[0,2], \Delta t=0,01$$



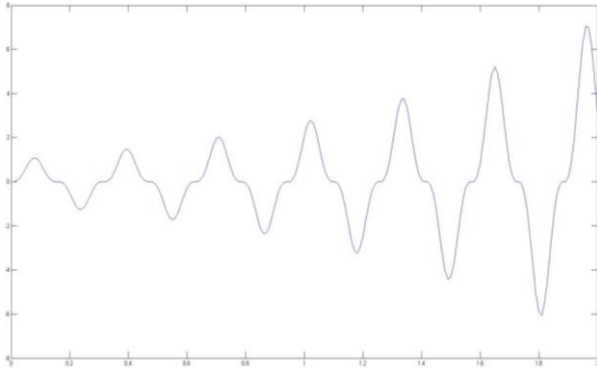
##### Вариант 6

$$y(t)=e^t\cdot\cos^2(20\cdot t), t\in[0,2], \Delta t=0,01$$



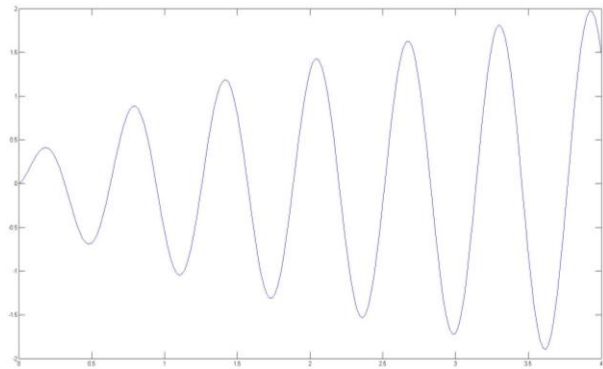
**Вариант 7**

$y(t)=e^t \cdot \sin^3(20 \cdot t), t \in [0,2], \Delta t=0,01$



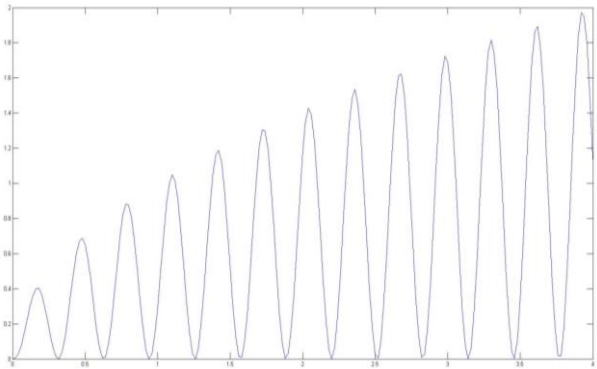
**Вариант 8**

$y(t)=\sqrt{t} \cdot \sin(10 \cdot t), t \in [0,4], \Delta t=0,02$



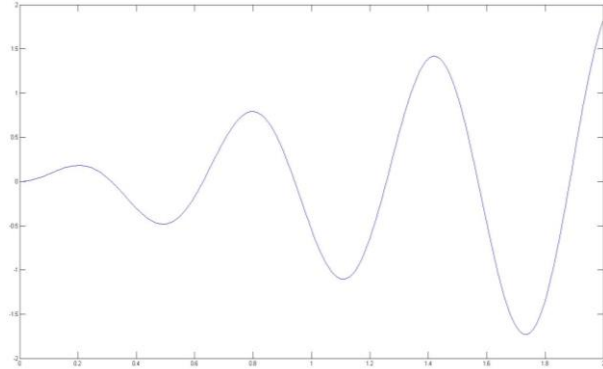
**Вариант 9**

$y(t)=\sqrt{t} \cdot \sin^2(10 \cdot t), t \in [0,4], \Delta t=0,02$



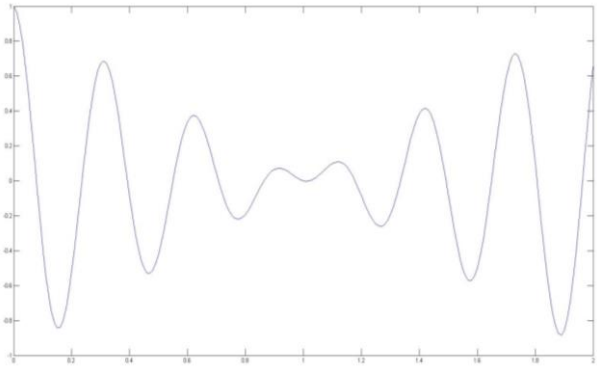
**Вариант 10**

$y(t)=t \cdot \sin^2(10 \cdot t), t \in [0,2], \Delta t=0,01$



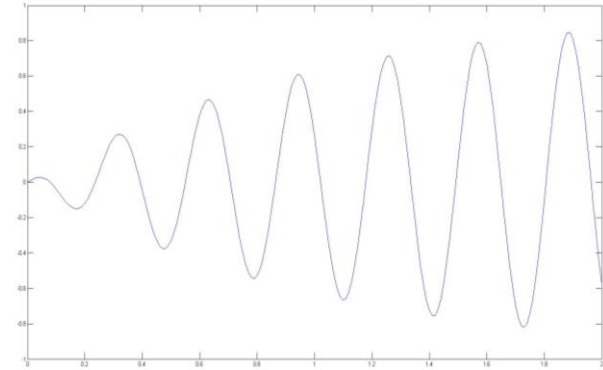
**Вариант 11**

$y(t)=(1-t) \cdot \cos(20 \cdot t), t \in [0,2], \Delta t=0,01$



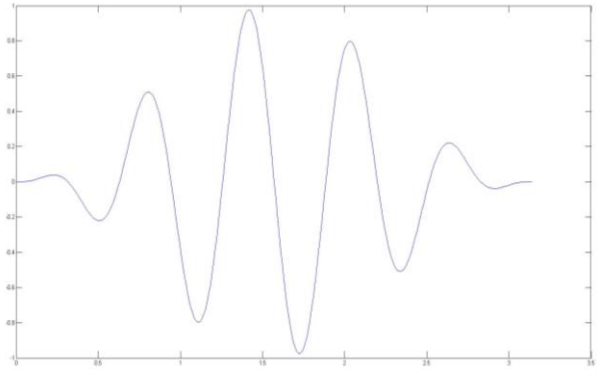
**Вариант 12**

$y(t)=(1-e^{-t}) \cdot \cos(20 \cdot t), t \in [0,2], \Delta t=0,01$



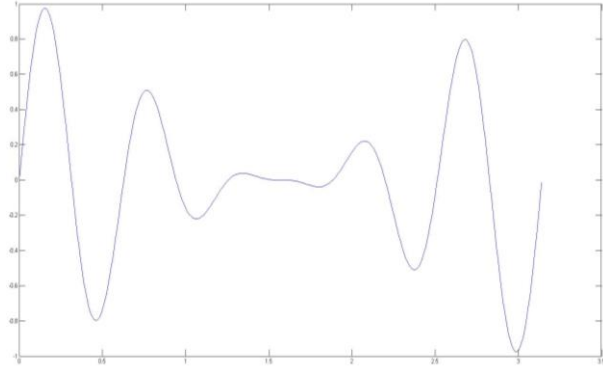
**Вариант 13**

$y(t)=\sin^2 t \cdot \sin(10 \cdot t), t \in [0,\pi], \Delta t=0,01$



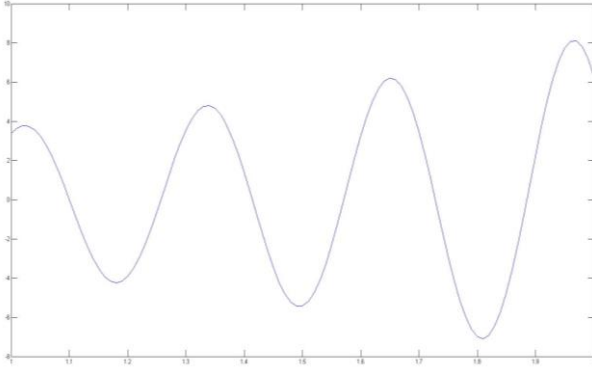
**Вариант 14**

$y(t)=\cos^2 t \cdot \sin(10 \cdot t), t \in [0,\pi], \Delta t=0,01$



*Вариант 15*

$y(t)=(1+e^t)\cdot\sin(20\cdot t)$ ,  $t\in[0,2]$ ,  $\Delta t=0,01$



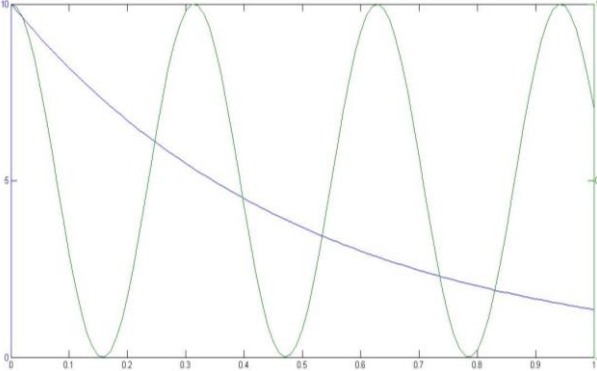
**Задача 1.2**

Построить в одном окне графики двух функций  $y_1(t)$  и  $y_2(t)$  с использованием оператора **plotyy(...)**.

**ВАРИАНТЫ РЕШЕНИЙ**

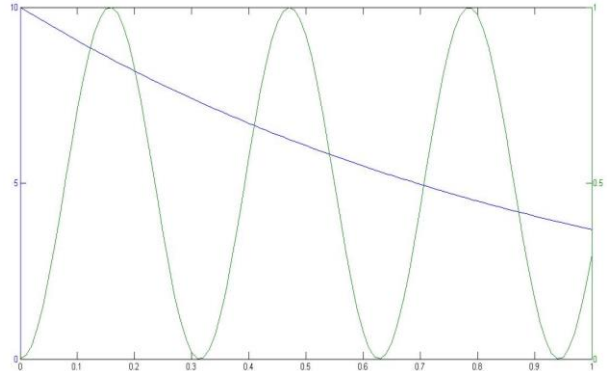
*Вариант 1*

$y_1(t)=10\cdot e^{-2t}$ ,  $y_2(t)=\cos(20\cdot t)$ ,  $t\in[0,1]$ ,  $\Delta t=0,01$



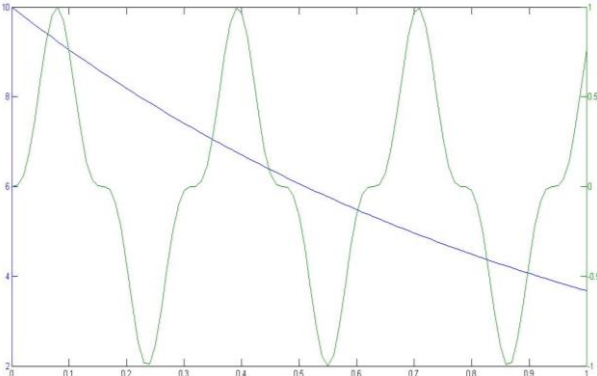
*Вариант 2*

$y_1(t)=10\cdot e^{-t}$ ,  $y_2(t)=\sin^2(10\cdot t)$ ,  $t\in[0,1]$ ,  $\Delta t=0,01$



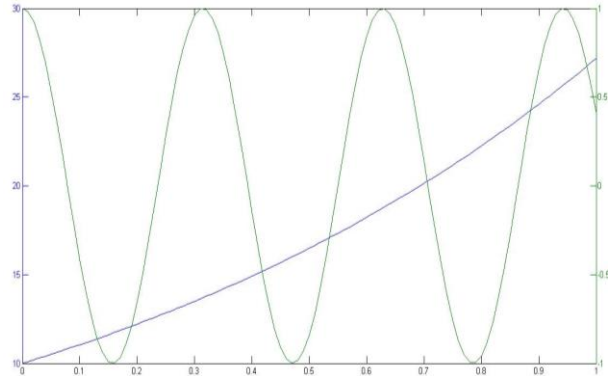
*Вариант 3*

$y_1(t)=10\cdot e^{-t}$ ,  $y_2(t)=\sin^3(20\cdot t)$ ,  $t\in[0,1]$ ,  $\Delta t=0,01$



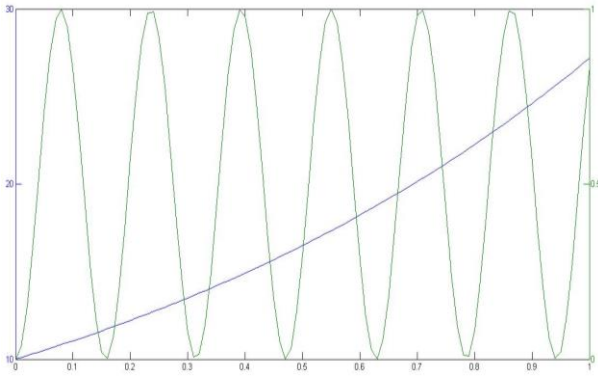
*Вариант 4*

$y_1(t)=10\cdot e^t$ ,  $y_2(t)=\cos(20\cdot t)$ ,  $t\in[0,1]$ ,  $\Delta t=0,01$



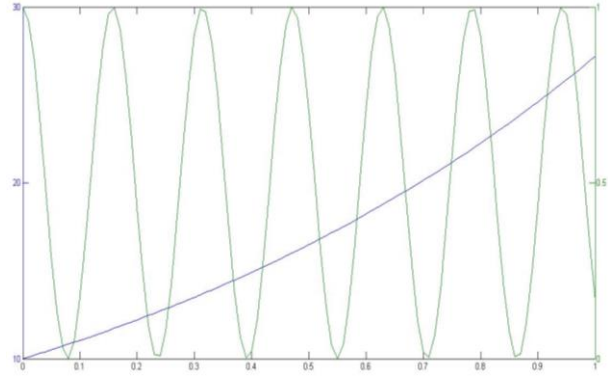
**Вариант 5**

$y_1(t)=10 \cdot e^t, y_2(t)=\sin^2(20 \cdot t), t=[0,1], \Delta t=0,01$



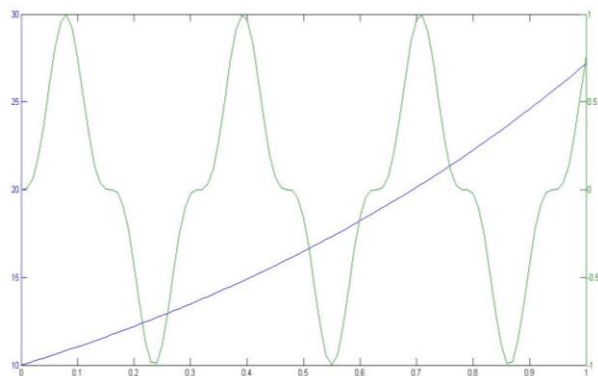
**Вариант 6**

$y_1(t)=10 \cdot e^t, y_2(t)=\cos^2(20 \cdot t), t=[0,1], \Delta t=0,01$



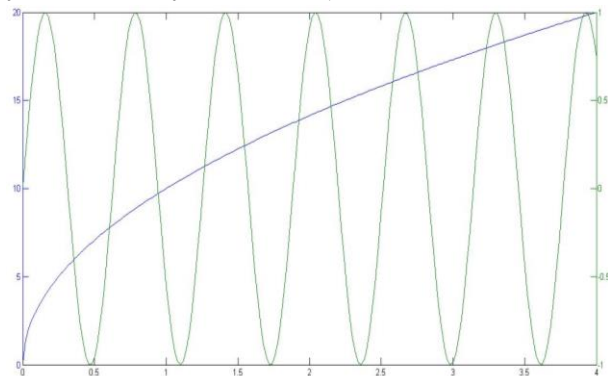
**Вариант 7**

$y_1(t)=10 \cdot e^t, y_2(t)=\sin^3(20 \cdot t), t=[0,1], \Delta t=0,01$



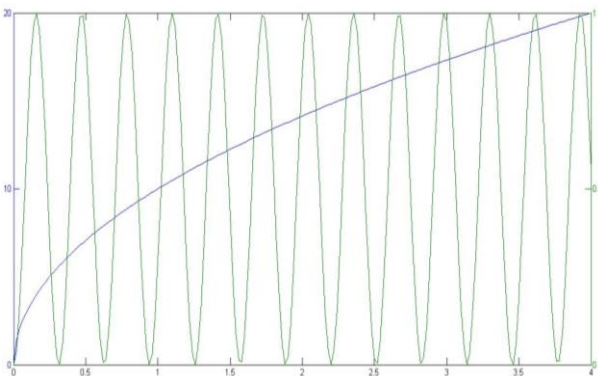
**Вариант 8**

$y_1(t)=10 \cdot \sqrt{t}, y_2(t)=\sin(10 \cdot t), t=[0,4], \Delta t=0,02$



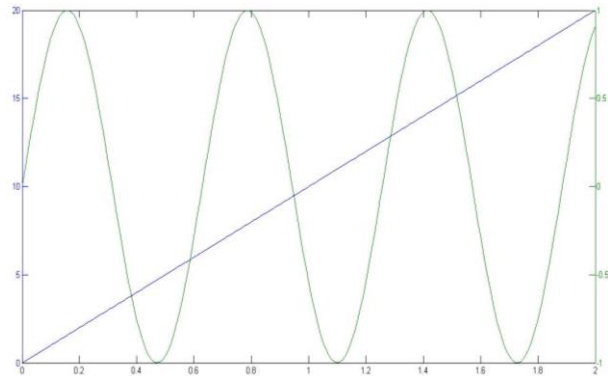
**Вариант 9**

$y_1(t)=10 \cdot \sqrt{t}, y_2(t)=\sin^2(10 \cdot t), t=[0,4], \Delta t=0,02$



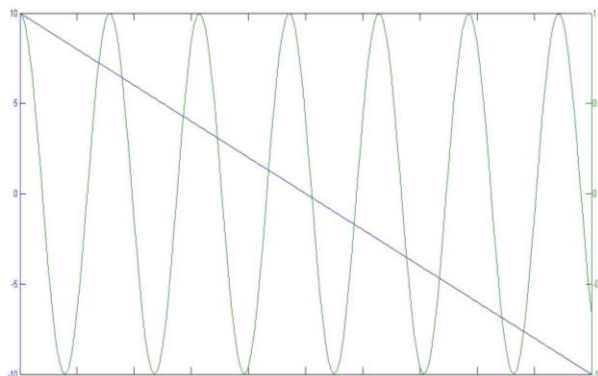
**Вариант 10**

$y_1(t)=10 \cdot t, y_2(t)=\sin(10 \cdot t), t=[0,2], \Delta t=0,01$



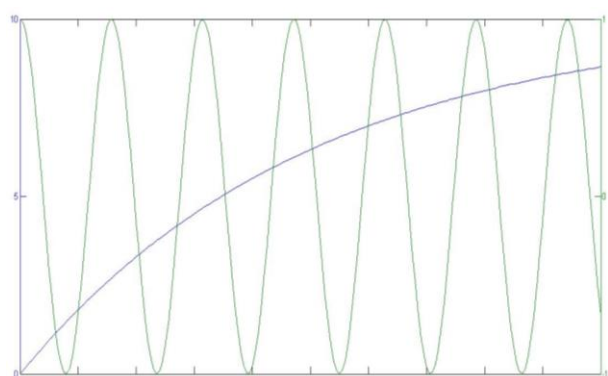
**Вариант 11**

$y_1(t)=10 \cdot (1-t), y_2(t)=\cos(20 \cdot t), t=[0,2], \Delta t=0,01$



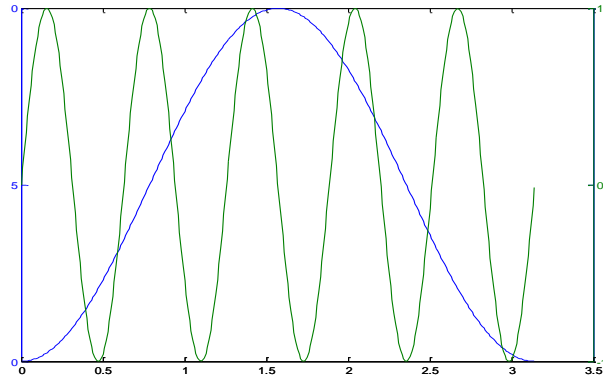
**Вариант 12**

$y_1(t)=10 \cdot (1-e^{-t}), y_2(t)=\cos(20 \cdot t), t=[0,2], \Delta t=0,01$



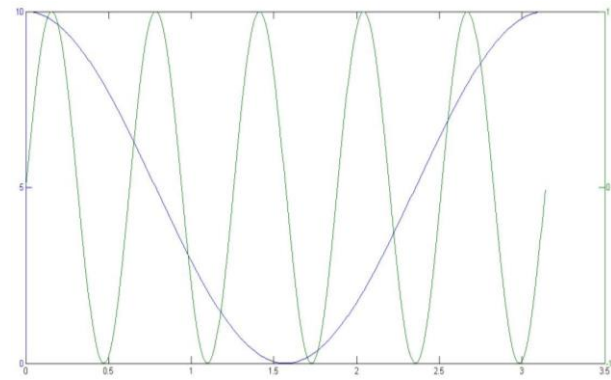
### Вариант 13

$$y_1(t)=10\cdot\sin^2t, y_2(t)=\sin(10\cdot t), t=[0,\pi], \Delta t=0,01$$



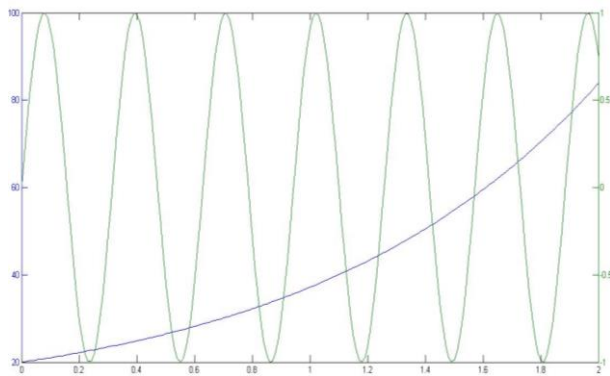
### Вариант 14

$$y_1(t)=10\cdot\cos^2t, y_2(t)=\sin(10\cdot t), t=[0,\pi], \Delta t=0,01$$



### Вариант 15

$$y_1(t)=10\cdot(1+e^t), y_2(t)=\sin(20\cdot t), t=[0,2], \Delta t=0,01$$



## Задача 1.3

С помощью оператора `plot(t,y1,'S1',t,y2,'S2')` построить в одном окне графики двух функций  $y_1(t)$  и  $y_2(t)$ . При построении графиков вид функций, пределы, шаг изменения аргумента и параметр **S** выбрать с использованием табл. 3.

Таблица 3

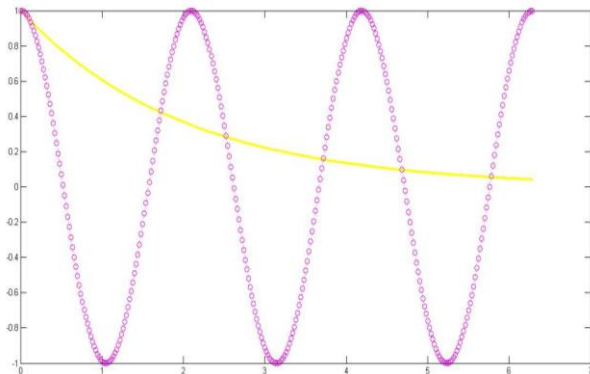
Вариант	Функции	Параметры (цвет линии, тип маркера, тип линии)
1	2	3
1	$y_1(t)=e^{-0,5t}, t\in[0,2\pi], \Delta t=0,2$	Желтый, точка, сплошная
	$y_2(t)=\cos(3\cdot t), t\in[0,2\pi], \Delta t=0,2$	Розовый, кружок, пунктирная
2	$y_1(t)=e^{-t}, t\in[0,2\pi], \Delta t=0,2$	Голубой, крестик, штрих-пунктирная
	$y_2(t)=\sin^2(t), t\in[0,2\pi], \Delta t=0,2$	Красный, знак плюс, штриховая
3	$y_1(t)=e^{-t}, t\in[0,2\pi], \Delta t=0,2$	Зеленый, треугольник вершиной влево, сплошная
	$y_2(t)=\sin^3(t), t\in[0,2\pi], \Delta t=0,2$	Синий, треугольник вершиной вправо, пунктирная
4	$y_1(t)=0,05\cdot e^t, t\in[0,2\pi], \Delta t=0,2$	Черный, пятиконечная звезда, штрих-пунктирная
	$y_2(t)=60\cdot\cos(2\cdot t), t\in[0,2\pi], \Delta t=0,2$	Желтый, знак плюс, штриховая

1	2	3
5	$y_1(t)=0,1 \cdot e^{-t}, t \in [0,2\pi], \Delta t=0,2$	Желтый, квадрат, штрих-пунктирная
	$y_2(t)=60 \cdot \sin^2(t), t \in [0,2\pi], \Delta t=0,2$	Розовый, ромб, штриховая
6	$y_1(t)=0,1 \cdot e^{-t}, t \in [0,2\pi], \Delta t=0,2$	Голубой, треугольник вершиной вниз, сплошная
	$y_2(t)=60 \cdot \cos^2(2 \cdot t), t \in [0,2\pi], \Delta t=0,2$	Красный, треугольник вершиной вверх, пунктирная
7	$y_1(t)=0,1 \cdot e^{-t}, t \in [0,2\pi], \Delta t=0,2$	Розовый, знак плюс, пунктирная
	$y_2(t)=60 \cdot \sin^3(3 \cdot t), t \in [0,2\pi], \Delta t=0,2$	Голубой, пятиконечная звезда, штрих-пунктирная
8	$y_1(t)=\sqrt{t}, t \in [0,2\pi], \Delta t=0,2$	Зеленый, шестиконечная звезда, штриховая
	$y_2(t)=2,5 \cdot \sin(t), t \in [0,2\pi], \Delta t=0,2$	Желтый, точка, сплошная
9	$y_1(t)=\sqrt{t}, t \in [0,2\pi], \Delta t=0,2$	Розовый, кружок, пунктирная
	$y_2(t)=2,5 \cdot \sin^2(t), t \in [0,2\pi], \Delta t=0,2$	Голубой, крестик, штрих-пунктирная
10	$y_1(t)=t, t \in [0,2\pi], \Delta t=0,2$	Красный, знак плюс, штриховая
	$y_2(t)=8 \cdot \sin(10 \cdot t), t \in [0,2\pi], \Delta t=0,2$	Зеленый, треугольник вершиной влево, сплошная
11	$y_1(t)=1-t, t \in [0,2\pi], \Delta t=0,2$	Синий, треугольник вершиной вправо, пунктирная
	$y_2(t)=6 \cdot \cos(t), t \in [0,2\pi], \Delta t=0,2$	Черный, пятиконечная звезда, штрих-пунктирная
12	$y_1(t)=1-e^{-t}, t \in [0,2\pi], \Delta t=0,2$	Желтый, знак плюс, штриховая
	$y_2(t)=2 \cdot \cos(t), t \in [0,2\pi], \Delta t=0,2$	Красный, квадрат, штрих-пунктирная
13	$y_1(t)=\sin^2 t, t \in [0,2\pi], \Delta t=0,2$	Розовый, ромб, штриховая
	$y_2(t)=\sin(10 \cdot t), t \in [0,2\pi], \Delta t=0,2$	Голубой, треугольник вершиной вниз, сплошная
14	$y_1(t)=\cos^2 t, t \in [0,2\pi], \Delta t=0,2$	Красный, треугольник вершиной вверх, пунктирная
	$y_2(t)=\sin(t), t \in [0,2\pi], \Delta t=0,2$	Розовый, знак плюс, пунктирная,
15	$y_1(t)=0,1 \cdot (1+e^t), t \in [0,2\pi], \Delta t=0,2$	Голубой, пятиконечная звезда, штрих-пунктирная
	$y_2(t)=60 \cdot \sin(t), t \in [0,2\pi], \Delta t=0,2$	Зеленый, шестиконечная звезда, штриховая

### ВАРИАНТЫ РЕШЕНИЙ

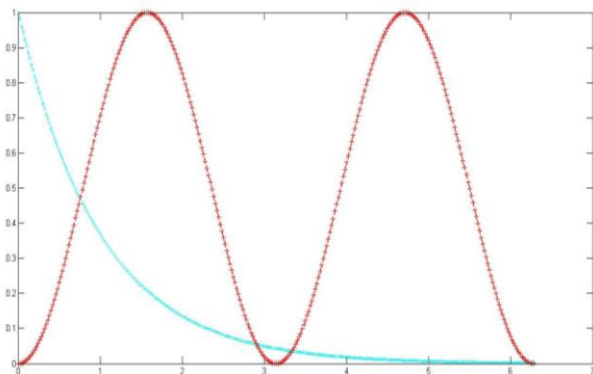
#### Вариант 1

$$y_1(t)=e^{-t}, y_2(t)=\cos(3 \cdot t), t \in [0,2\pi], \Delta t=0,2$$



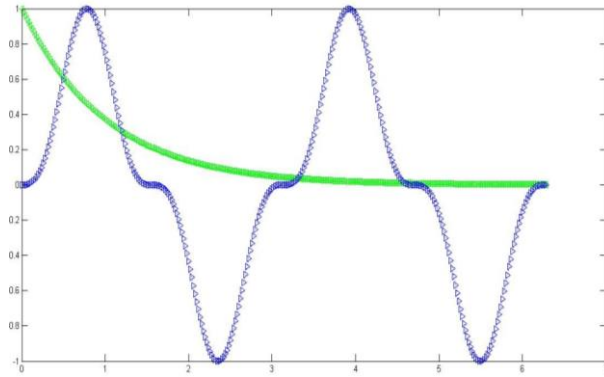
#### Вариант 2

$$y_1(t)=e^{-t}, y_2(t)=\sin^2(t), t \in [0,2\pi], \Delta t=0,2$$



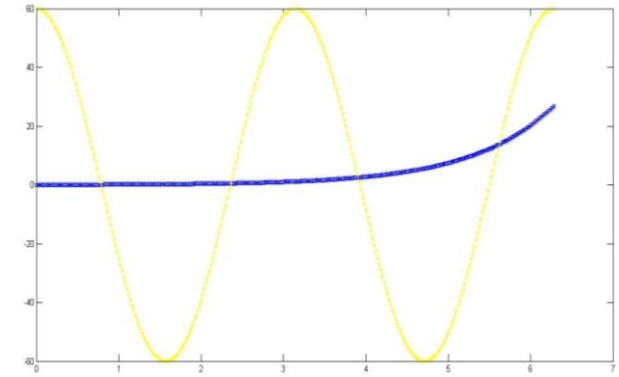
**Вариант 3**

$y_1(t)=e^{-t}, y_2(t)=\sin^3(t), t \in [0, 2\pi], \Delta t=0,2$



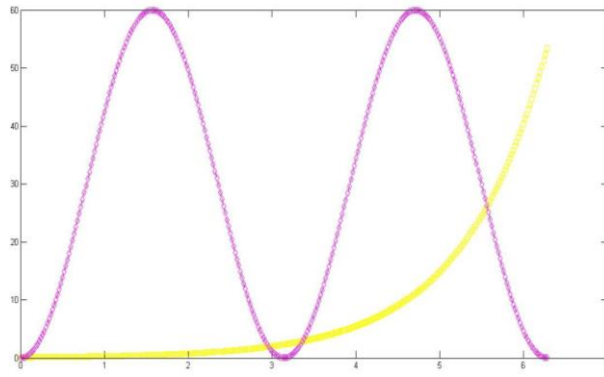
**Вариант 4**

$y_1(t)=0,05 \cdot e^t, y_2(t)=60 \cdot \cos(2 \cdot t), t \in [0, 2\pi], \Delta t=0,2$



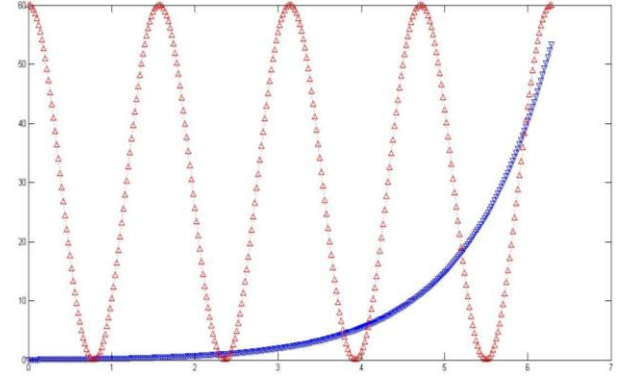
**Вариант 5**

$y_1(t)=0,1 \cdot e^{-t}, y_2(t)=60 \cdot \sin^2(t), t \in [0, 2\pi], \Delta t=0,2$



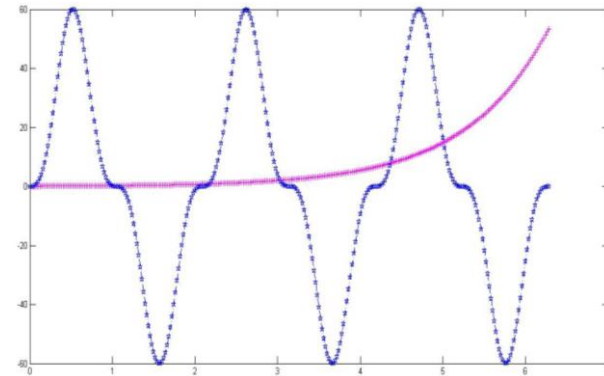
**Вариант 6**

$y_1(t)=0,1 \cdot e^{-t}, y_2(t)=60 \cdot \cos^2(2 \cdot t), t \in [0, 2\pi], \Delta t=0,2$



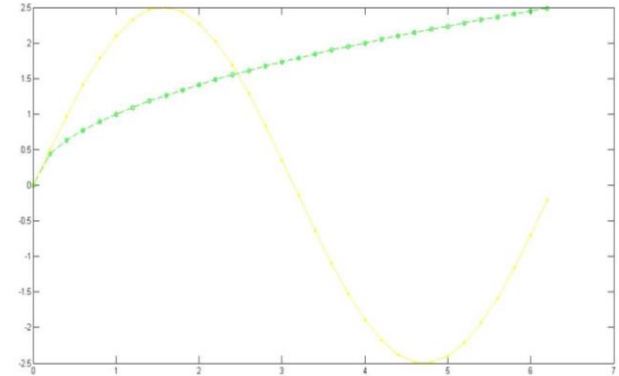
**Вариант 7**

$y_1(t)=0,1 \cdot e^{-t}, y_2(t)=60 \cdot \sin^3(3 \cdot t), t \in [0, 2\pi], \Delta t=0,2$



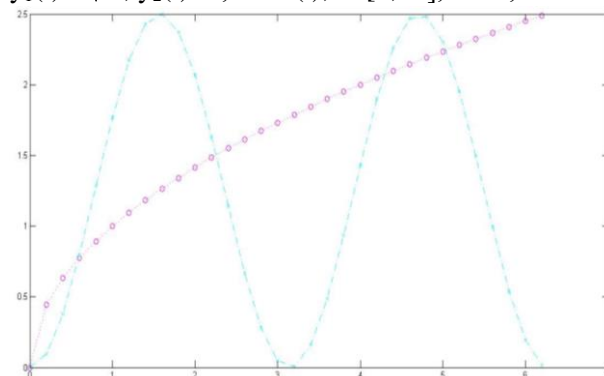
**Вариант 8**

$y_1(t)=\sqrt{t}, y_2(t)=2,5 \cdot \sin(t), t \in [0, 2\pi], \Delta t=0,2$



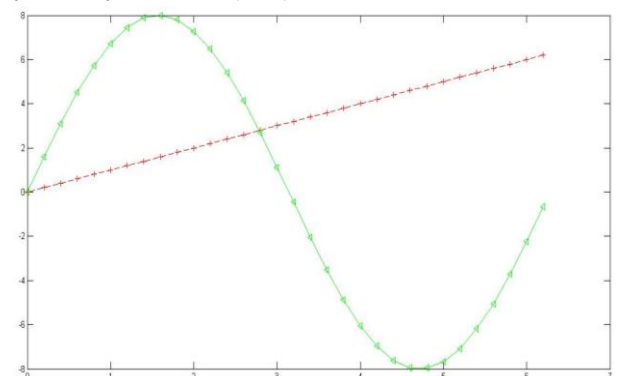
**Вариант 9**

$y_1(t)=\sqrt{t}, y_2(t)=2,5 \cdot \sin^2(t), t \in [0, 2\pi], \Delta t=0,2$



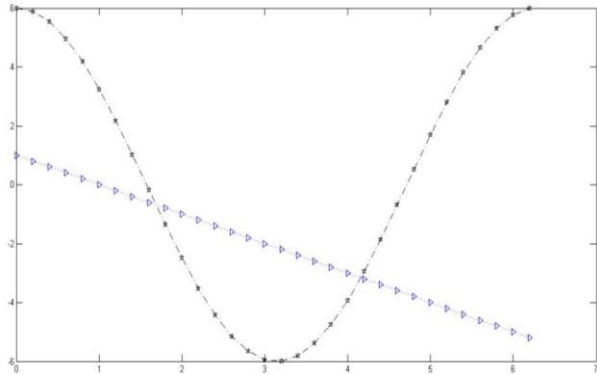
**Вариант 10**

$y_1(t)=t, y_2(t)=8 \cdot \sin(10 \cdot t), t \in [0, 2\pi], \Delta t=0,2$



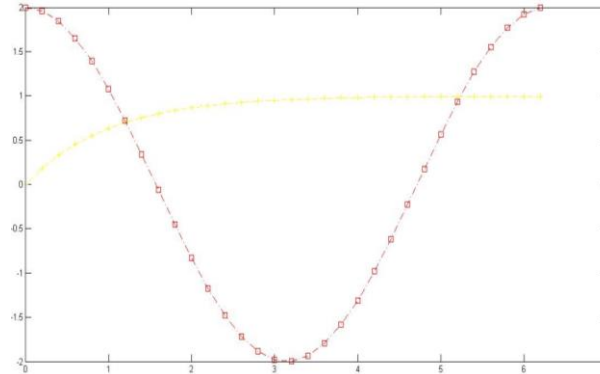
### Вариант 11

$$y_1(t)=1-t, y_2(t)=6 \cdot \cos(t), t \in [0, 2\pi], \Delta t=0,2$$



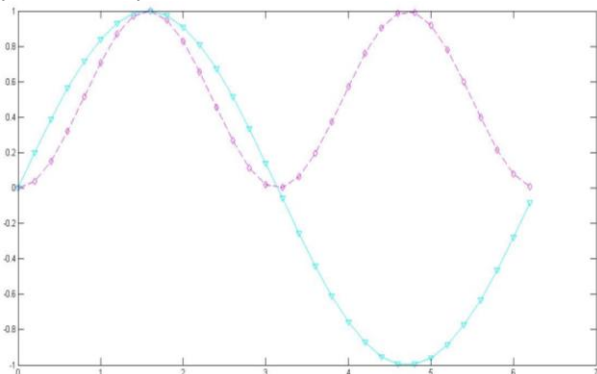
### Вариант 12

$$y_1(t)=1-e^{-t}, y_2(t)=2 \cdot \cos(t), t \in [0, 2\pi], \Delta t=0,2$$



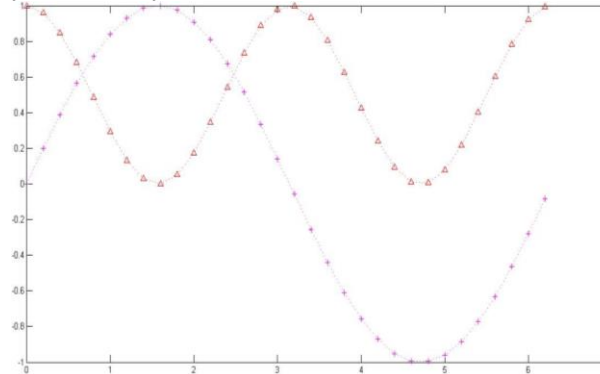
### Вариант 13

$$y_1(t)=\sin^2 t, y_2(t)=\sin(t), t \in [0, 2\pi], \Delta t=0,2$$



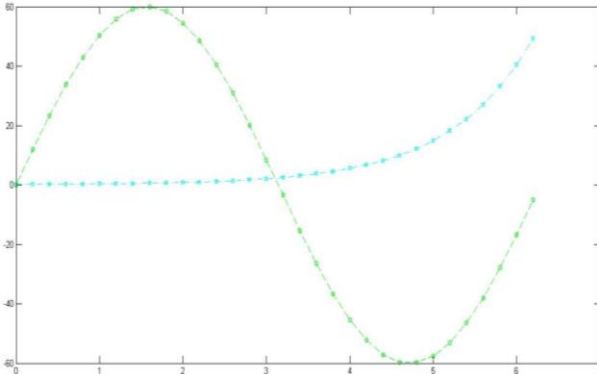
### Вариант 14

$$y_1(t)=\cos^2 t, y_2(t)=\sin(t), t \in [0, 2\pi], \Delta t=0,2$$



### Вариант 15

$$y_1(t)=0,1 \cdot (1+e^t), y_2(t)=60 \cdot \sin(t), t \in [0, 2\pi], \Delta t=0,2$$



## Задача 1.4

С помощью оператора `stem(t,y2,'S2')` построить график функции  $y_2(t)$ . При построении графиков вид функций, пределы, шаг изменения аргумента и параметр **S** выбрать из табл. 4.



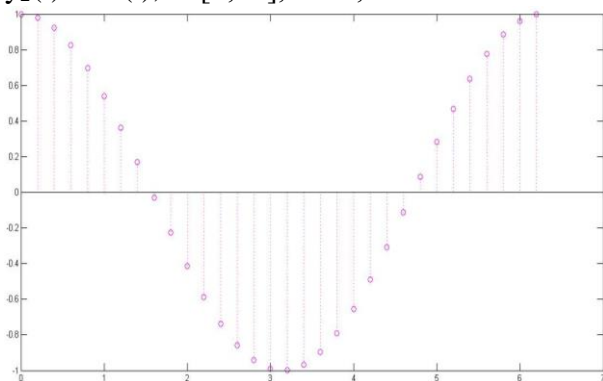
Таблица 4

Вариант	Функции	Параметры (цвет линии, тип маркера, тип линии)
1	2	3
1	$y_2(t)=\cos(t), t \in [0, 2\pi], \Delta t=0,2$	Розовый, кружок, пунктирная
2	$y_2(t)=\sin^2(t), t \in [0, 2\pi], \Delta t=0,2$	Красный, знак плюс, штриховая
3	$y_2(t)=\sin^3(t), t \in [0, 2\pi], \Delta t=0,2$	Синий, треугольник вершиной вправо, пунктирная
4	$y_2(t)=60 \cdot \cos(2t), t \in [0, 2\pi], \Delta t=0,2$	Зеленый, знак плюс, штриховая
5	$y_2(t)=60 \cdot \sin^2(t), t \in [0, 2\pi], \Delta t=0,2$	Розовый, ромб, штриховая
6	$y_2(t)=60 \cdot \cos^2(2t), t \in [0, 2\pi], \Delta t=0,2$	Красный, треугольник вершиной вверх, пунктирная
7	$y_2(t)=60 \cdot \sin^3(3t), t \in [0, 2\pi], \Delta t=0,2$	Голубой, пятиконечная звезда, штрих-пунктирная
8	$y_2(t)=2,5 \cdot \sin(t), t \in [0, 2\pi], \Delta t=0,2$	Желтый, точка, сплошная
9	$y_2(t)=2,5 \cdot \sin^2(t), t \in [0, 2\pi], \Delta t=0,2$	Голубой, крестик, штрих-пунктирная
10	$y_2(t)=8 \cdot \sin(10t), t \in [0, 2\pi], \Delta t=0,2$	Зеленый, треугольник вершиной влево, сплошная
11	$y_2(t)=6 \cdot \cos(t), t \in [0, 2\pi], \Delta t=0,2$	Черный, пятиконечная звезда, штрих-пунктирная
12	$y_2(t)=2 \cdot \cos(t), t \in [0, 2\pi], \Delta t=0,2$	Красный, квадрат, штрих-пунктирная
13	$y_2(t)=\sin(10t), t \in [0, 2\pi], \Delta t=0,2$	Голубой, треугольник вершиной вниз, сплошная
14	$y_2(t)=\sin(t), t \in [0, 2\pi], \Delta t=0,2$	Розовый, знак плюс, пунктирная,
15	$y_2(t)=60 \cdot \sin(t), t \in [0, 2\pi], \Delta t=0,2$	Зеленый, шестиконечная звезда, штриховая

### ВАРИАНТЫ РЕШЕНИЙ

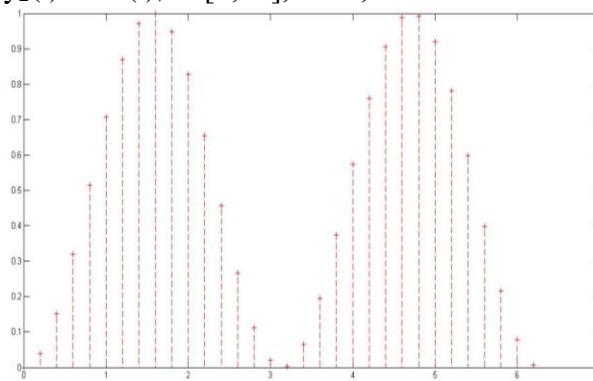
#### Вариант 1

$$y_2(t)=\cos(t), t \in [0, 2\pi], \Delta t=0,2$$



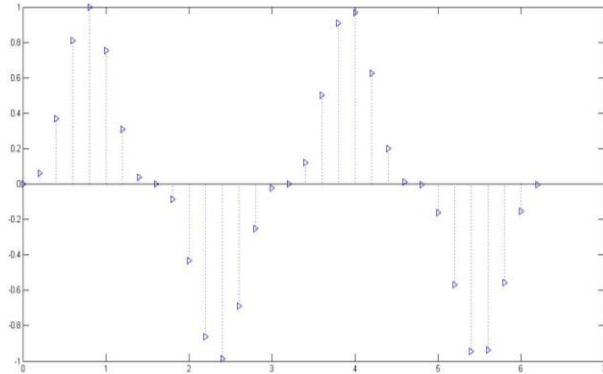
#### Вариант 2

$$y_2(t)=\sin^2(t), t \in [0, 2\pi], \Delta t=0,2$$



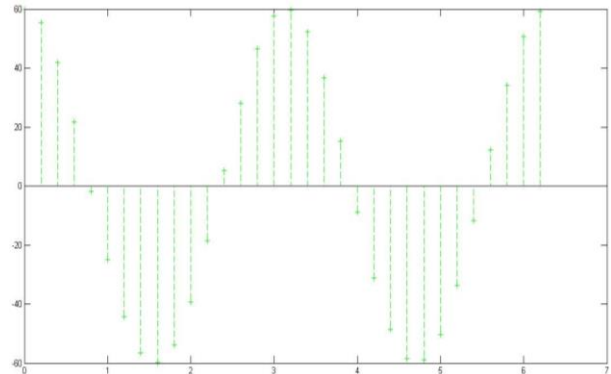
**Вариант 3**

$y_2(t) = \sin^3(t), t \in [0, 2\pi], \Delta t = 0,2$



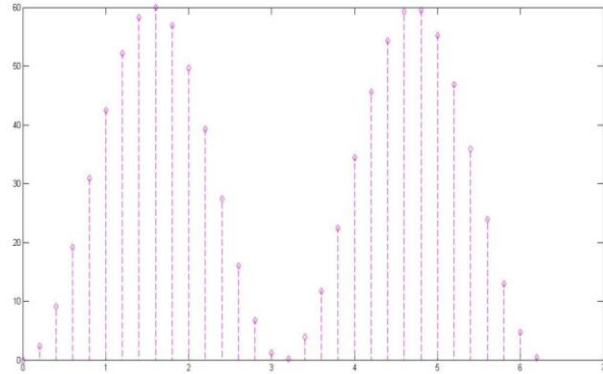
**Вариант 4**

$y_2(t) = 60 \cdot \cos(2t), t \in [0, 2\pi], \Delta t = 0,2$



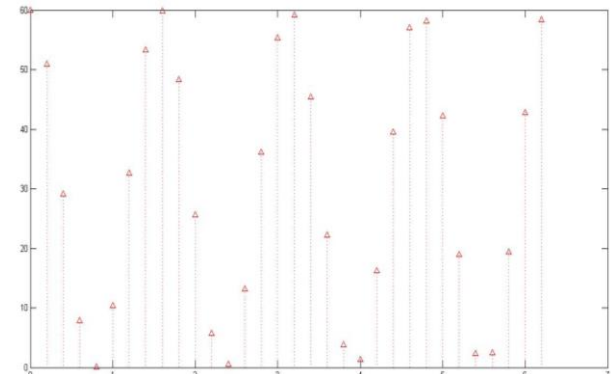
**Вариант 5**

$y_2(t) = 60 \cdot \sin^2(t), t \in [0, 2\pi], \Delta t = 0,2$



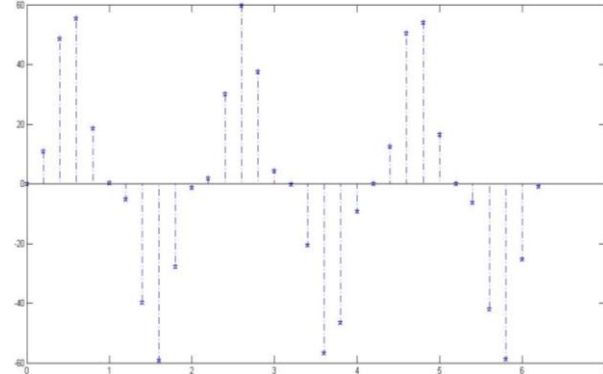
**Вариант 6**

$y_2(t) = 60 \cdot \cos^2(2t), t \in [0, 2\pi], \Delta t = 0,2$



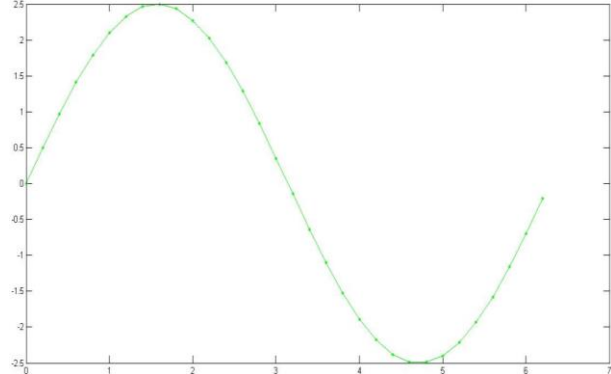
**Вариант 7**

$y_2(t) = 60 \cdot \sin^3(3t), t \in [0, 2\pi], \Delta t = 0,2$



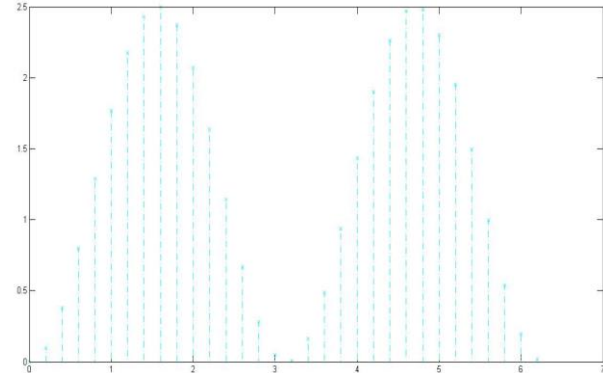
**Вариант 8**

$y_2(t) = 2,5 \cdot \sin(t), t \in [0, 2\pi], \Delta t = 0,2$



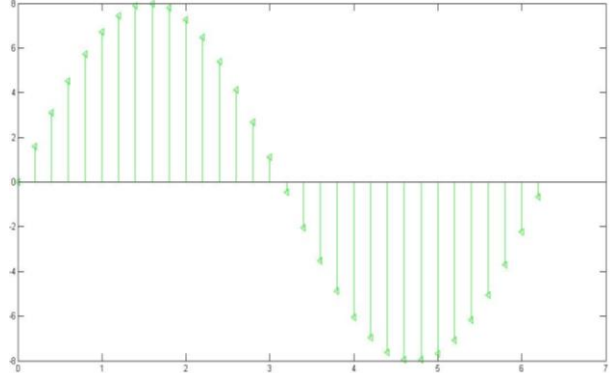
**Вариант 9**

$y_2(t) = 2,5 \cdot \sin^2(t), t \in [0, 2\pi], \Delta t = 0,2$



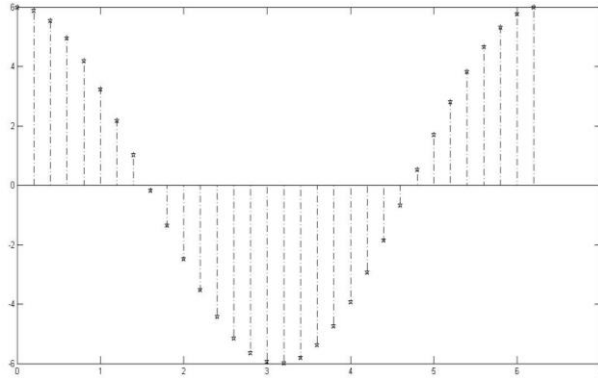
**Вариант 10**

$y_2(t) = 8 \cdot \sin(10t), t \in [0, 2\pi], \Delta t = 0,2$



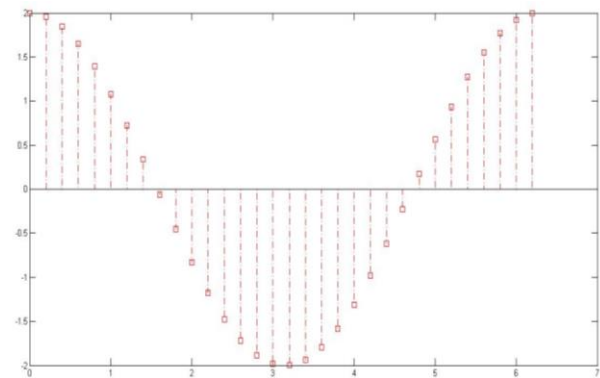
*Вариант 11*

$y_2(t)=6 \cdot \cos(t), t \in [0, 2\pi], \Delta t=0,2$



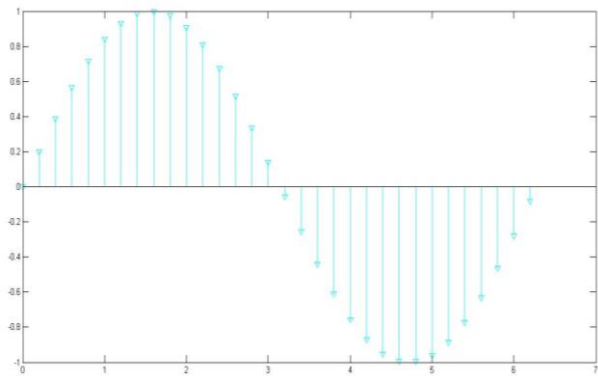
*Вариант 12*

$y_2(t)=2 \cdot \cos(t), t \in [0, 2\pi], \Delta t=0,2$



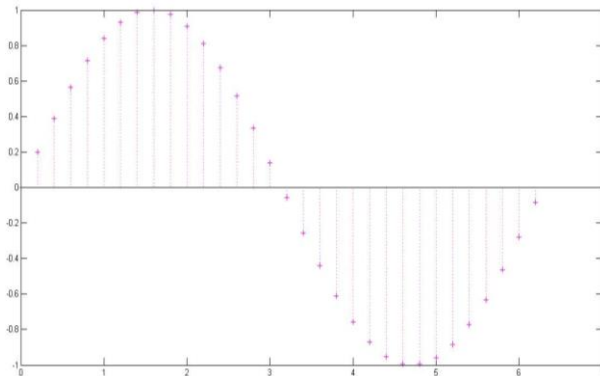
*Вариант 13*

$y_2(t)=\sin(10 \cdot t), t \in [0, 2\pi], \Delta t=0,2$



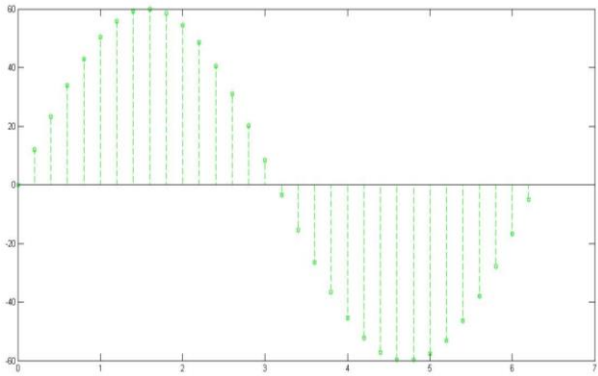
*Вариант 14*

$y_2(t)=\sin(t), t \in [0, 2\pi], \Delta t=0,2$



*Вариант 15*

$y_2(t)=60 \cdot \sin(t), t \in [0, 2\pi], \Delta t=0,2$



**Задача 1.5**

С помощью оператора **stairs(t,y1,'S')** построить график функции **y1(t)**. При построении графиков вид функций, пределы, шаг изменения аргумента и параметр **S** выбрать из табл. 5.

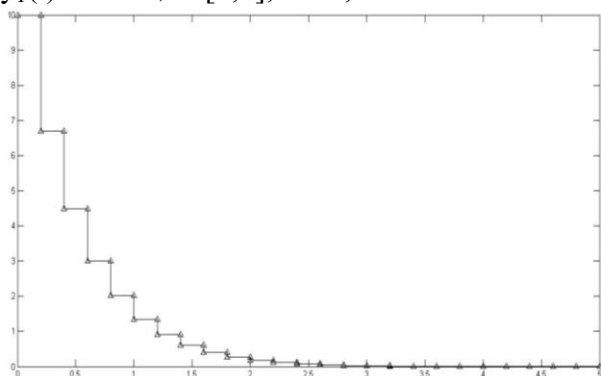
Таблица 5

Вариант	Функции	Параметры (цвет линии, тип маркера, тип линии)
1	2	3
1	$y_1(t)=10 \cdot e^{-2t}, t \in [0,5], \Delta t=0,2$	Черный, треугольник вершиной вверх, сплошная
2	$y_1(t)=e^t, t \in [0,4], \Delta t=0,2$	Черный, шестиконечная звезда, пунктирная
3	$y_1(t)=\sqrt{t}, t \in [0,100], \Delta t=5$	Красный, ромб, штриховая
4	$y_1(t)=1-e^{-2t}, t \in [0,5], \Delta t=0,2$	Черный, треугольник вершиной вверх, сплошная
5	$y_1(t)=t^2, t \in [0,10], \Delta t=0,6$	Красный, кружок, штриховая
6	$y_1(t)=\sin(t), t \in [0,2\pi], \Delta t=0,08$	Зелёный, точка, сплошная
7	$y_1(t)=t^{\frac{2}{3}}, t \in [0,10], \Delta t=0,6$	Голубой, треугольник вершиной вверх, сплошная
8	$y_1(t)=t^{-2}, t \in [0,2,1], \Delta t=0,06$	Красный, треугольник вершиной вправо, пунктирная
9	$y_1(t)=\arcsin(t), t \in [0,1], \Delta t=0,08$	Голубой, треугольник вершиной вправо, штриховая
10	$y_1(t)=\arccos(t), t \in [0,1], \Delta t=0,06$	Зеленый, треугольник вершиной влево, сплошная
11	$y_1(t)=\operatorname{arctg}(t), t \in [0,5], \Delta t=0,4$	Красный, ромб, пунктирная
12	$y_1(t)=\operatorname{acot}(t), t \in [0,1,5], \Delta t=0,4$	Красный, квадрат, штрих-пунктирная
13	$y_1(t)=\arcsin( t ), t \in [-1,1], \Delta t=0,08$	Красный, пятиконечная звезда, штриховая
14	$y_1(t)=\arccos( t ), t \in [-1,1], \Delta t=0,08$	Синий, квадрат, штриховая
15	$y_1(t)=1-t^2, t \in [0,5], \Delta t=0,35$	Красный, пятиконечная звезда, штрих-пунктирная

### ВАРИАНТЫ РЕШЕНИЙ

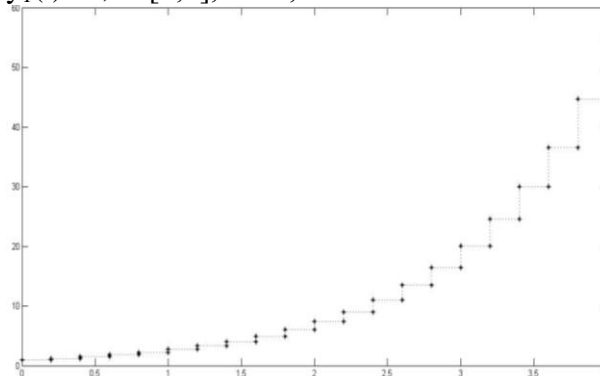
#### Вариант 1

$$y_1(t)=10 \cdot e^{-2t}, t \in [0,5], \Delta t=0,2$$



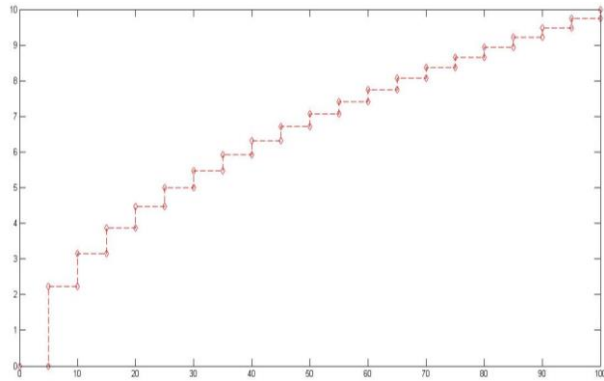
#### Вариант 2

$$y_1(t)=e^t, t \in [0,4], \Delta t=0,2$$



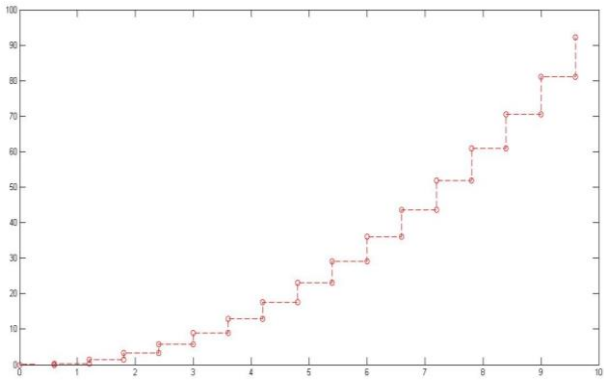
**Вариант 3**

$y_1(t)=\sqrt{t}$ ,  $t \in [0,100]$ ,  $\Delta t=5$



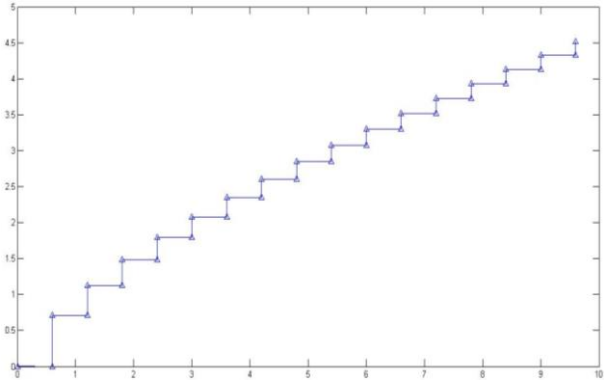
**Вариант 5**

$y_1(t)=t^2$ ,  $t \in [0,10]$ ,  $\Delta t=0,6$



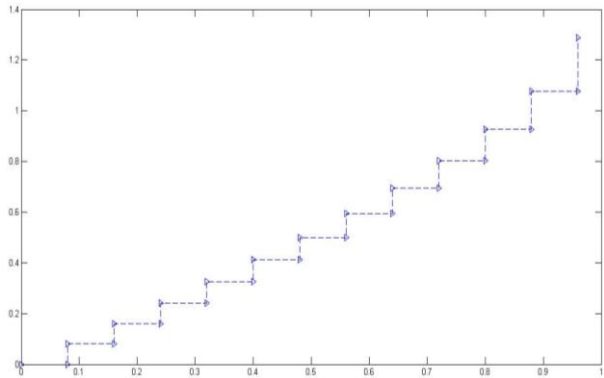
**Вариант 7**

$y_1(t)=t^{\frac{2}{3}}$ ,  $t \in [0,10]$ ,  $\Delta t=0,6$



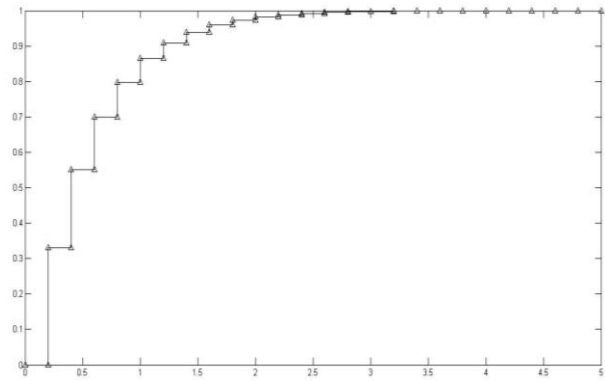
**Вариант 9**

$y_1(t)=\arcsin(t)$ ,  $t \in [0,1]$ ,  $\Delta t=0,08$



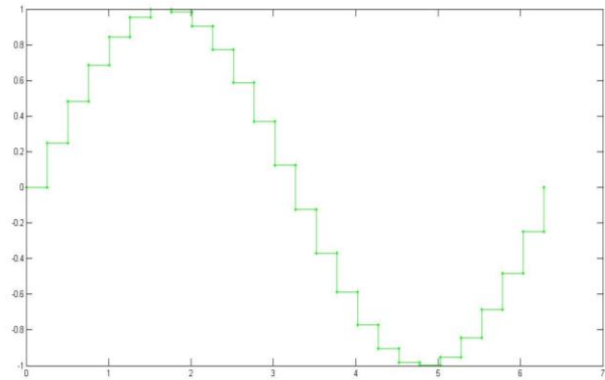
**Вариант 4**

$y_1(t)=1-e^{-2t}$ ,  $t \in [0,5]$ ,  $\Delta t=0,2$



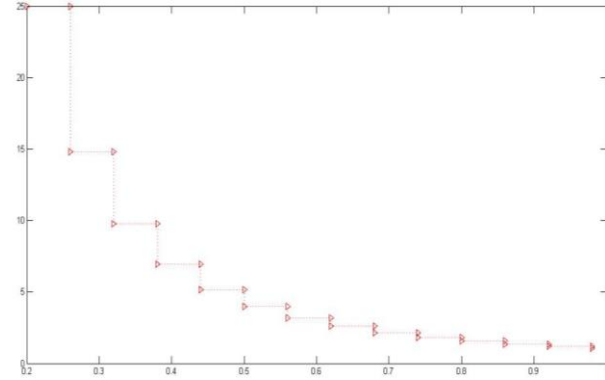
**Вариант 6**

$y_1(t)=\sin(t)$ ,  $t \in [0,2\pi]$ ,  $\Delta t=0,08$



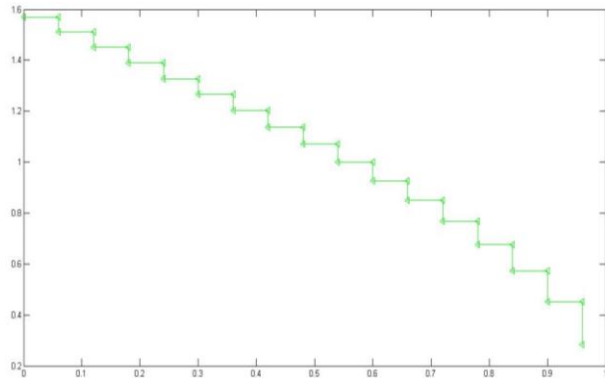
**Вариант 8**

$y_1(t)=t^{-2}$ ,  $t \in [0,2,1]$ ,  $\Delta t=0,06$



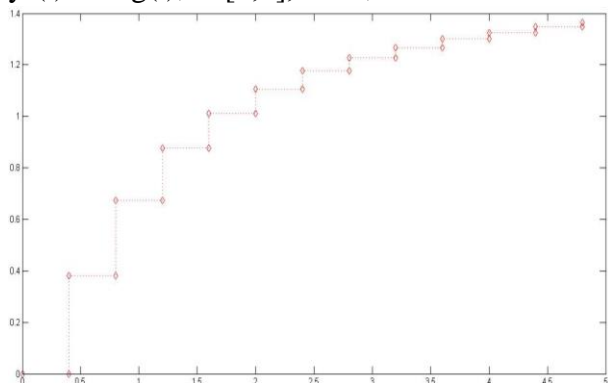
**Вариант 10**

$y_1(t)=\arccos(t)$ ,  $t \in [0,1]$ ,  $\Delta t=0,06$



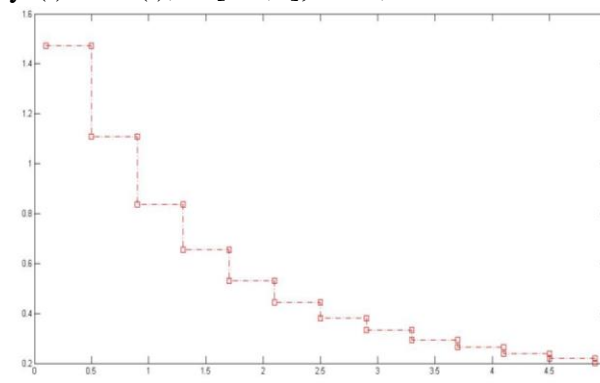
### Вариант 11

$$y_1(t) = \arctg(t), t \in [0, 5], \Delta t = 0,4$$



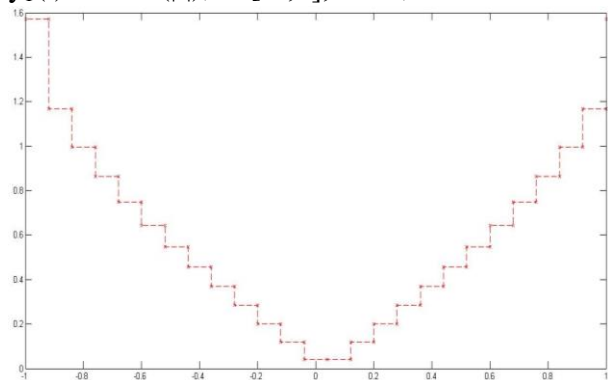
### Вариант 12

$$y_1(t) = \text{acot}(t), t \in [0, 1, 5], \Delta t = 0,4$$



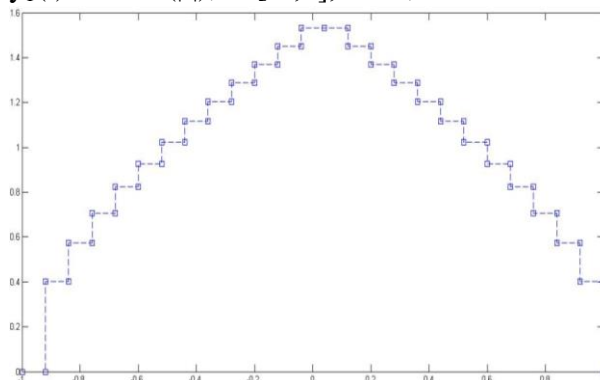
### Вариант 13

$$y_1(t) = \arcsin(|t|), t \in [-1, 1], \Delta t = 0,08$$



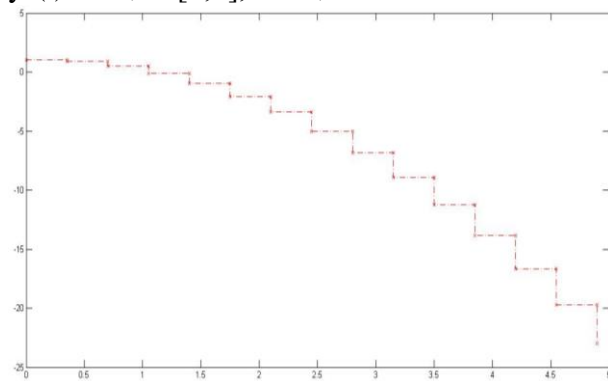
### Вариант 14

$$y_1(t) = \arccos(|t|), t \in [-1, 1], \Delta t = 0,08$$



### Вариант 15

$$y_1(t) = 1 - t^2, t \in [0, 5], \Delta t = 0,35$$



## Задача 1.6

С помощью оператора **errorbar(t,y,E,'S')** построить график функции  $y_1(t)$ . При построении графиков вид функций, пределы, шаг изменения аргумента и параметр **S** выбрать из табл. 6.

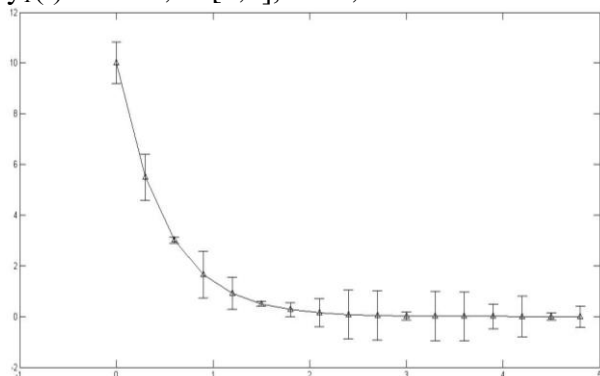
Таблица 6

Вариант	Функции	Параметры (цвет линии, тип маркера, тип линии)
1	2	3
1	$y_1(t)=10 \cdot e^{-2t}, t \in [0,5], \Delta t=0,3$	Черный, треугольник вершиной вверх, сплошная
2	$y_1(t)=e^t, t \in [0,4], \Delta t=0,2$	Черный, шестиконечная звезда, пунктирная
3	$y_1(t)=\sqrt{t}, t \in [0,60] \Delta t=4$	Красный, ромб, штриховая
4	$y_1(t)=1-10 \cdot e^t, t \in [0,5], \Delta t=0,3$	Черный, треугольник вершиной вверх, сплошная
5	$y_1(t)=t^2, t \in [0,5], \Delta t=0,3$	Красный, кружок, штриховая
6	$y_1(t)=2 \cdot \sin(t), t \in [0,2\pi], \Delta t=0,4$	Зеленый, точка, сплошная
7	$y_1(t)=t^{2/3}, t \in [0,10], \Delta t=0,6$	Голубой, треугольник вершиной вверх, сплошная
8	$y_1(t)=t^{-2}, t \in [0,2,1], \Delta t=0,06$	Красный, треугольник вершиной вправо, пунктирная
9	$y_1(t)=3 \cdot \arcsin(t), t \in [0,1], \Delta t=0,06$	Голубой, треугольник вершиной вправо, штриховая
10	$y_1(t)=4 \cdot \arccos(t), t \in [0,1], \Delta t=0,06$	Зеленый, треугольник вершиной влево, сплошная
11	$y_1(t)=4 \cdot \arctg(t), t \in [0,5], \Delta t=0,3$	Красный, ромб, пунктирная
12	$y_1(t)=3 \cdot \operatorname{acot}(t), t \in [0,1,5], \Delta t=0,4$	Красный, квадрат, штрих-пунктирная
13	$y_1(t)=2 \cdot \arcsin( t ), t \in [-1,1], \Delta t=0,1$	Красный, пятиконечная звезда, штриховая
14	$y_1(t)=3 \cdot \arccos( t ), t \in [-1,1], \Delta t=0,12$	Синий, квадрат, штриховая
15	$y_1(t)=1-t^2, t \in [0,5], \Delta t=0,3$	Красный, пятиконечная звезда, штрих-пунктирная

### ВАРИАНТЫ РЕШЕНИЙ

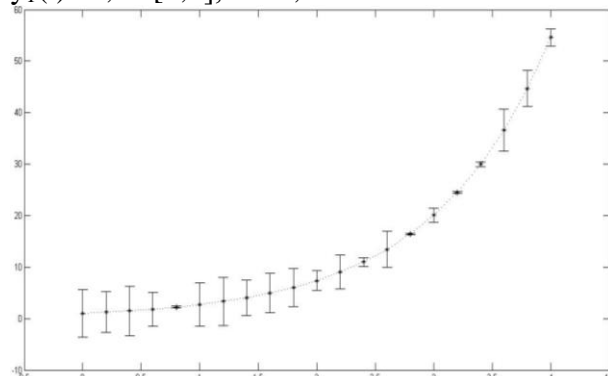
#### Вариант 1

$$y_1(t)=10 \cdot e^{-2t}, t \in [0,5], \Delta t=0,3$$



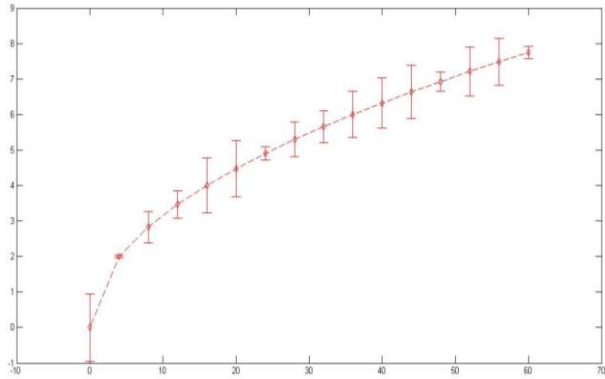
#### Вариант 2

$$y_1(t)=e^t, t \in [0,4], \Delta t=0,2$$



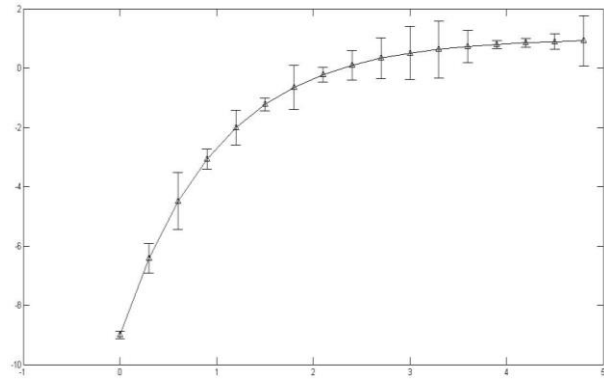
**Вариант 3**

$y_1(t) = \sqrt{t}$ ,  $t \in [0, 60]$ ,  $\Delta t = 4$



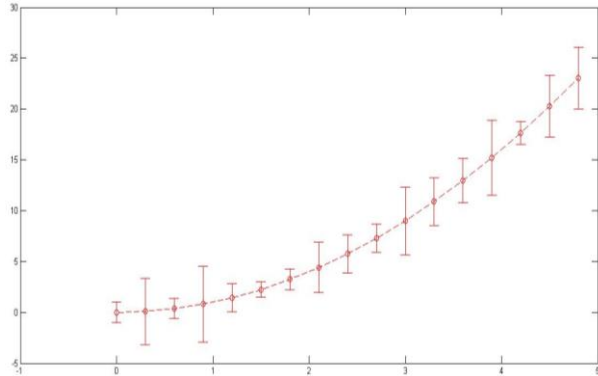
**Вариант 4**

$y_1(t) = 1 - 10 \cdot e^{-t}$ ,  $t \in [0, 5]$ ,  $\Delta t = 0,3$



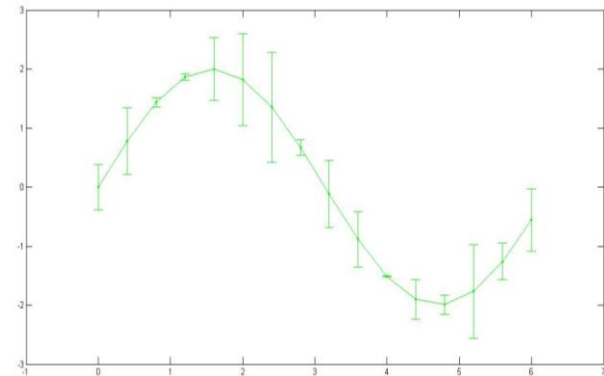
**Вариант 5**

$y_1(t) = t^2$ ,  $t \in [0, 5]$ ,  $\Delta t = 0,3$



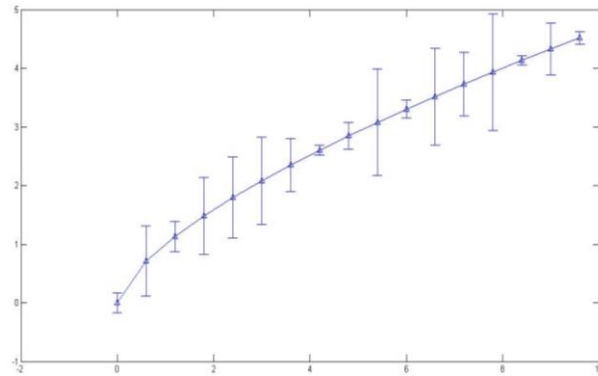
**Вариант 6**

$y_1(t) = 2 \cdot \sin(t)$ ,  $t \in [0, 2\pi]$ ,  $\Delta t = 0,4$



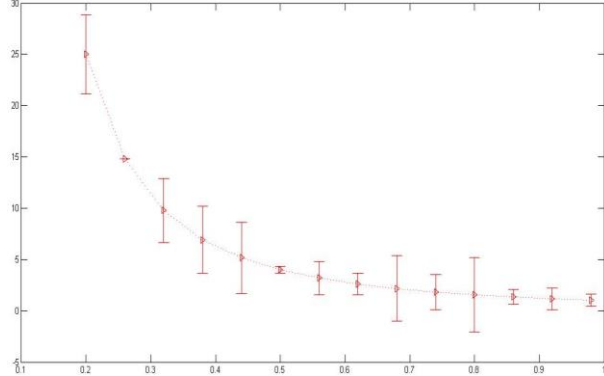
**Вариант 7**

$y_1(t) = t^{2/3}$ ,  $t \in [0, 10]$ ,  $\Delta t = 0,6$



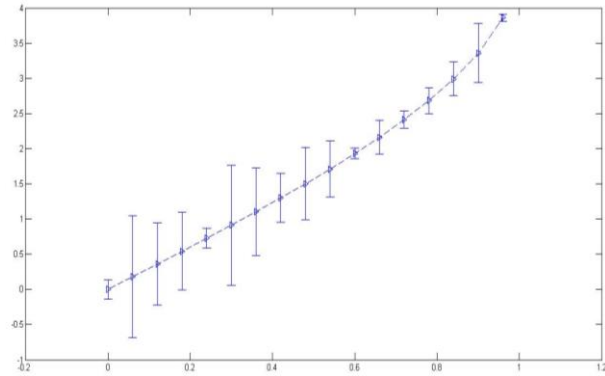
**Вариант 8**

$y_1(t) = t^{-2}$ ,  $t \in [0, 2, 1]$ ,  $\Delta t = 0,06$



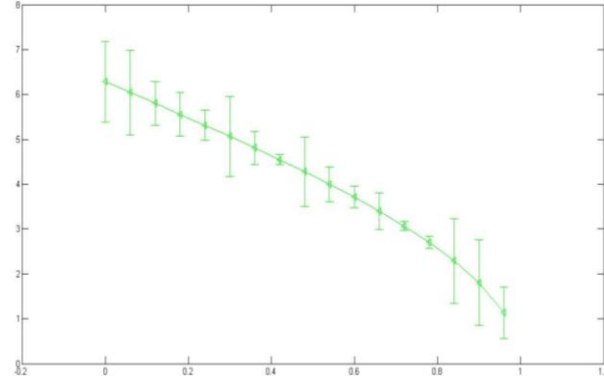
**Вариант 9**

$y_1(t) = 3 \cdot \arcsin(t)$ ,  $t \in [0, 1]$ ,  $\Delta t = 0,06$



**Вариант 10**

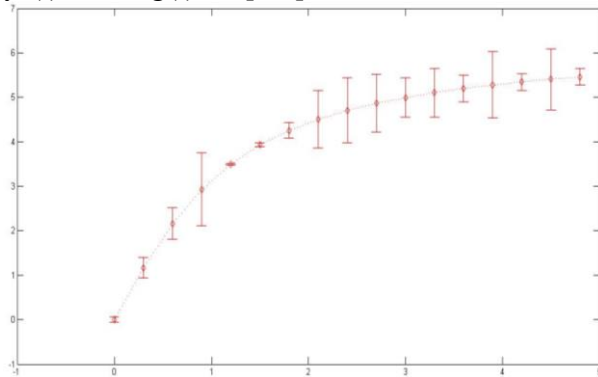
$y_1(t) = 4 \cdot \arccos(t)$ ,  $t \in [0, 1]$ ,  $\Delta t = 0,06$





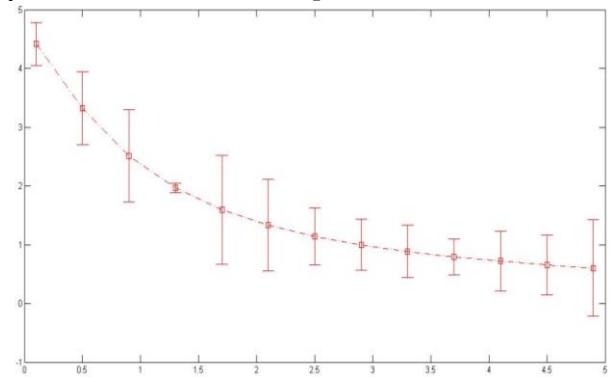
*Вариант 11*

$y_1(t)=4 \cdot \arctg(t), t \in [0,5], \Delta t=0,3$



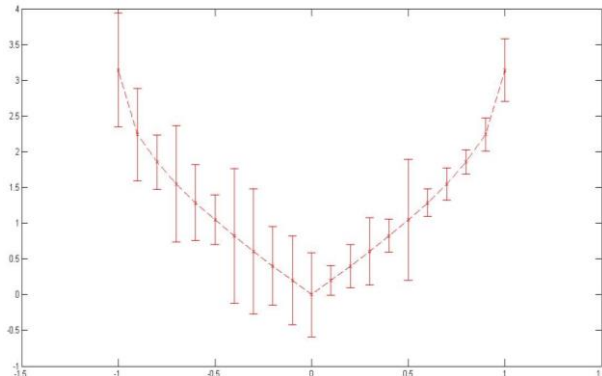
*Вариант 12*

$y_1(t)=3 \cdot \operatorname{acot}(t), t \in [0.1,5], \Delta t=0,4$



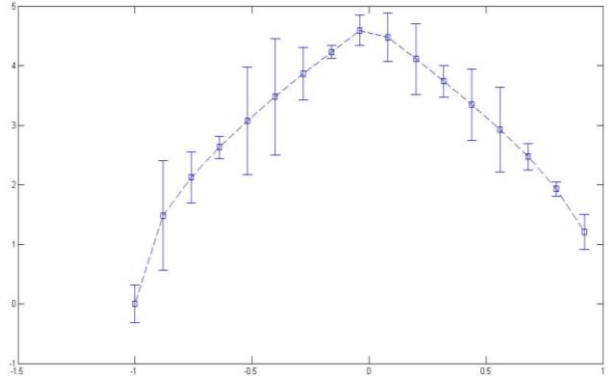
*Вариант 13*

$y_1(t)=2 \cdot \arcsin(|t|), t \in [-1,1], \Delta t=0,1$



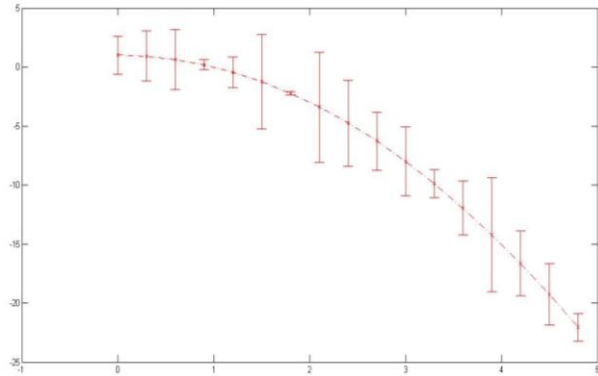
*Вариант 14*

$y_1(t)=3 \cdot \arccos(|t|), t \in [-1,1], \Delta t=0,12$



*Вариант 15*

$y_1(t)=1-t^2, t \in [0,5], \Delta t=0,3$



**Задача 1.7**

С помощью операторов **loglog(...)**, **semilogx(...)**, **semilogy(...)** построить график функции **y(t)**. При построении графиков вид функций, пределы и шаг изменения аргумента выбрать из табл. 7.

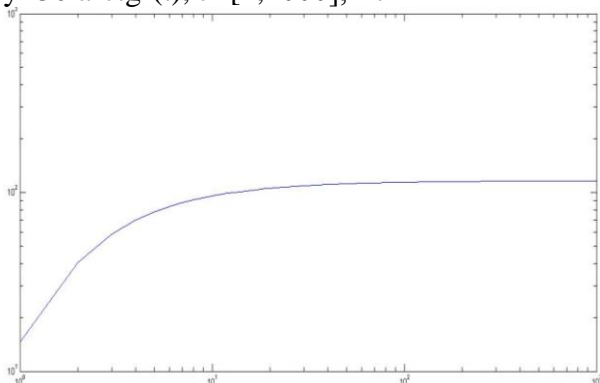
Таблица 7

Вариант	Функция	Оператор вывода
1	2	3
1	$y=30 \cdot \arctg^3(t), t \in [1, 1000], \Delta t=1$	Loglog
2	$y=(1-t) \cdot  \sin(0,05 \cdot t)^{\cos(0,01t)} , t \in [1, 1000], \Delta t=1$	Semilogx
3	$y=\ln(t^2) - \operatorname{tg}^2(0,1 \cdot t), t \in [1, 1000], \Delta t=1$	Semilogy
4	$y=\sin^4(\sqrt{t}), t \in [1, 1000], \Delta t=1$	Loglog
5	$y=0,01 \cdot \arctg^2(\cos(0,03 \cdot t)), t \in [1, 1000], \Delta t=1$	Semilogx
6	$y=\tan(0,05 \cdot \sin^2(0,1 \cdot t)), t \in [1, 300], \Delta t=1$	Semilogy
7	$y=\ln(t) - \cos^2(\sqrt{t}), t \in [1, 1000], \Delta t=1$	Loglog
8	$y= \arccos^3(0,01 \cdot t) , t \in [1, 1000], \Delta t=1$	Semilogx
9	$y=\ln(t) - \sin^2(0,05 \cdot t), t \in [1, 1000], \Delta t=1$	Semilogy
10	$y=\operatorname{tg}^2(\sqrt{t}), t \in [1, 1000], \Delta t=1$	Loglog
11	$y=(1000-t) \cdot \cos(\sin(0,05 \cdot t)), t \in [1, 1000], \Delta t=1$	Semilogx
12	$y=\ln(\cos^2(15 \cdot t)) - \sqrt{t}, t \in [1, 1000], \Delta t=1$	Semilogy
13	$y=(3-2 \cdot \sin(0,1 \cdot t)) \cdot  \sin(0,05 \cdot t) , t \in [1, 1000], \Delta t=1$	Loglog
14	$y=\ln(t) \cdot  \sin(0,05 \cdot t) , t \in [1, 1000], \Delta t=1$	Semilogx
15	$y=(1-t) \cdot \sin(0,05 \cdot t)^{\sin(0,05t)} , t \in [1, 1000], \Delta t=1$	Semilogy

### ВАРИАНТЫ РЕШЕНИЙ

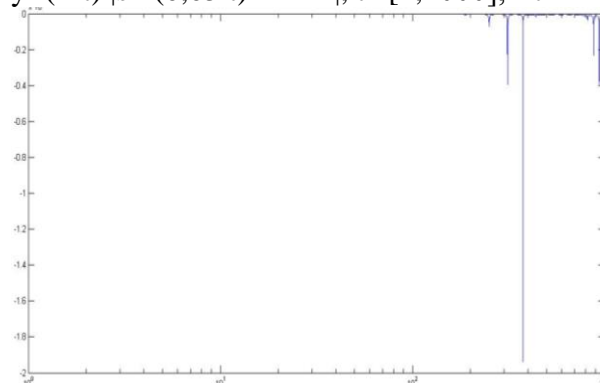
#### Вариант 1

$$y=30 \cdot \arctg^3(t), t \in [1, 1000], \Delta t=1$$



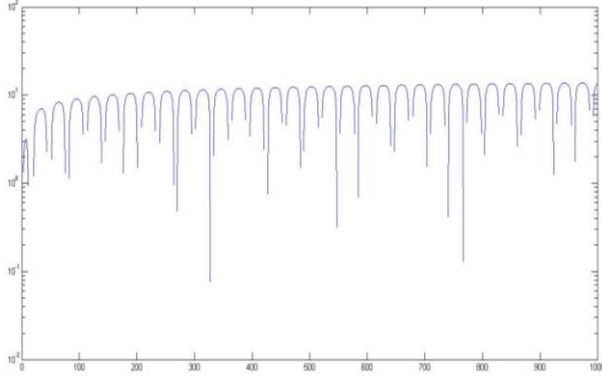
#### Вариант 2

$$y=(1-t) \cdot |\sin(0,05 \cdot t)^{\cos(0,01t)}|, t \in [1, 1000], \Delta t=1$$



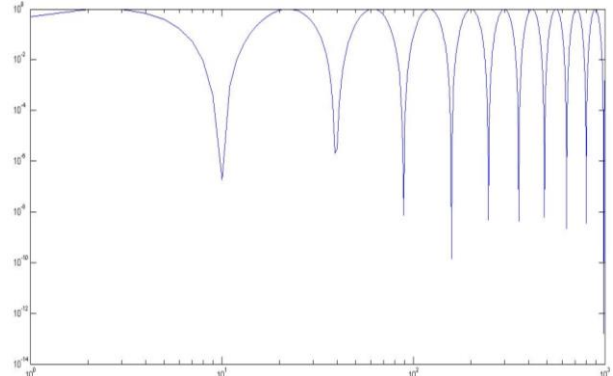
**Вариант 3**

$y = \ln(t^2) - \text{tg}^2(0,1 \cdot t), t \in [1, 1000], \Delta t = 1$



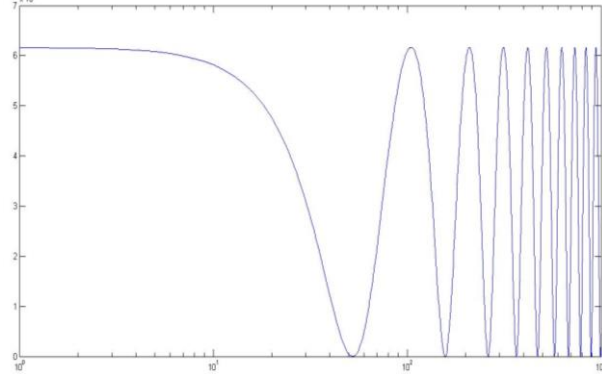
**Вариант 4**

$y = \sin^4(\sqrt{t}), t \in [1, 1000], \Delta t = 1$



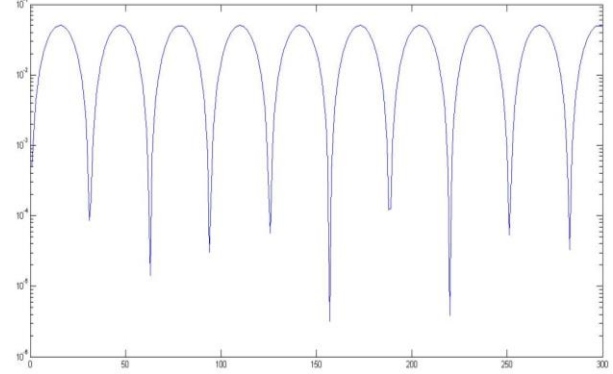
**Вариант 5**

$y = 0,01 \cdot \arctg^2(\cos(0,03 \cdot t)), t \in [1, 1000], \Delta t = 1$



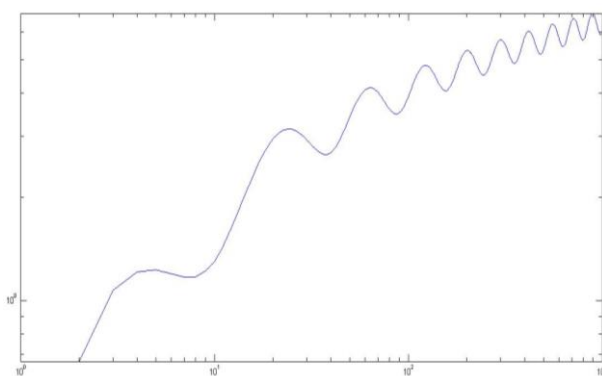
**Вариант 6**

$y = \tan(0,05 \cdot \sin^2(0,1 \cdot t)), t \in [1, 300], \Delta t = 1$



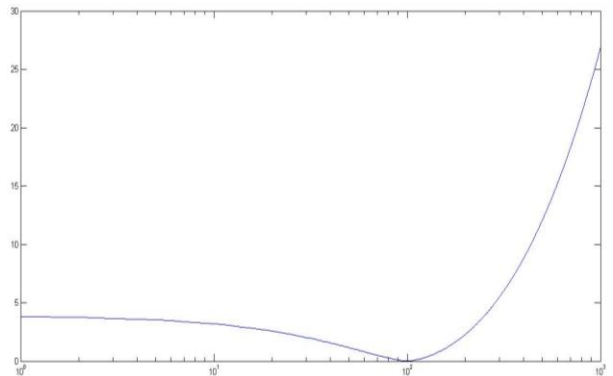
**Вариант 7**

$y = \ln(t) - \cos^2(\sqrt{t}), t \in [1, 1000], \Delta t = 1$



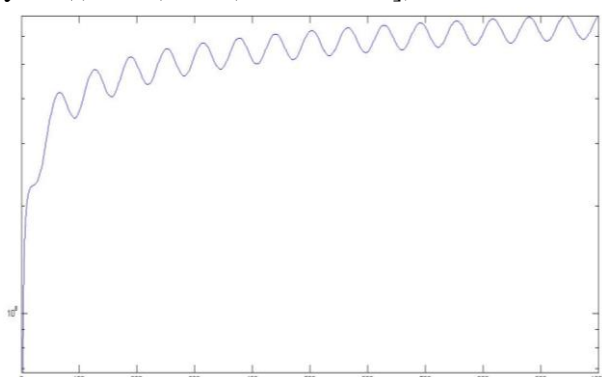
**Вариант 8**

$y = |\arccos^3(0,01 \cdot t)|, t \in [1, 1000], \Delta t = 1$



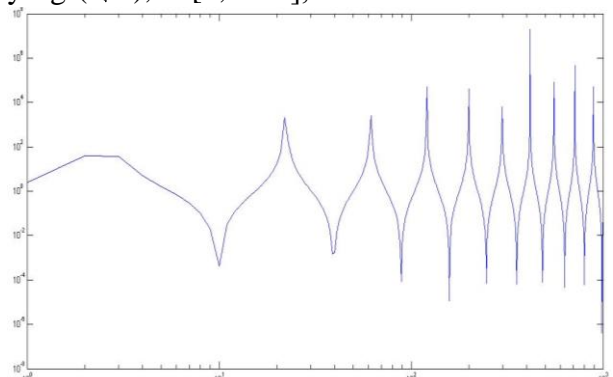
**Вариант 9**

$y = \ln(t) - \sin^2(0,05 \cdot t), t \in [1, 1000], \Delta t = 1$



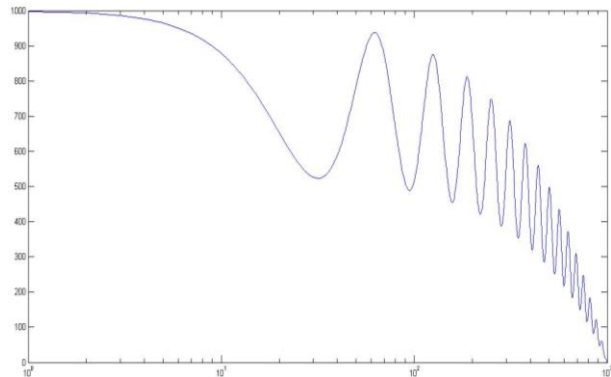
**Вариант 10**

$y = \text{tg}^2(\sqrt{t}), t \in [1, 1000], \Delta t = 1$



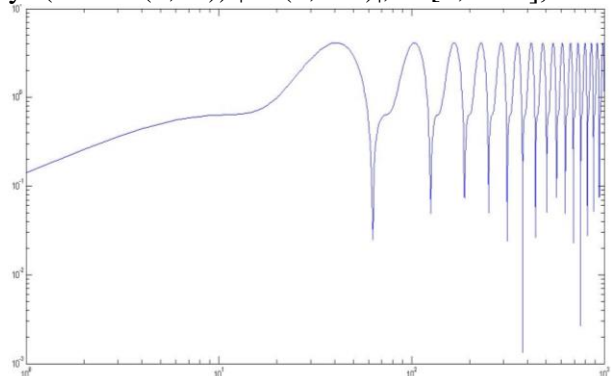
### Вариант 11

$$y=(1000-t)\cdot\cos(\sin(0,05\cdot t)), t\in[1,1000], \Delta t=1$$



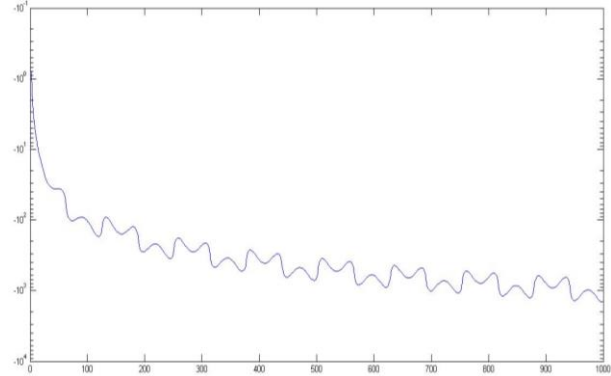
### Вариант 13

$$y=(3-2\cdot\sin(0,1\cdot t))\cdot|\sin(0,05\cdot t)|, t\in[1,1000], \Delta t=1$$



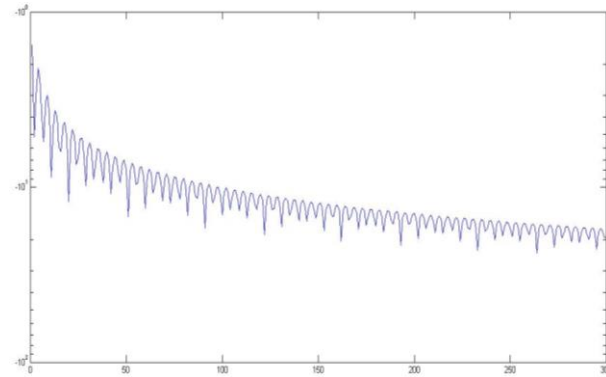
### Вариант 15

$$y=(1-t)\cdot\sin(0,05\cdot t)^{\sin(0,05t)}, t\in[1,1000], \Delta t=1$$



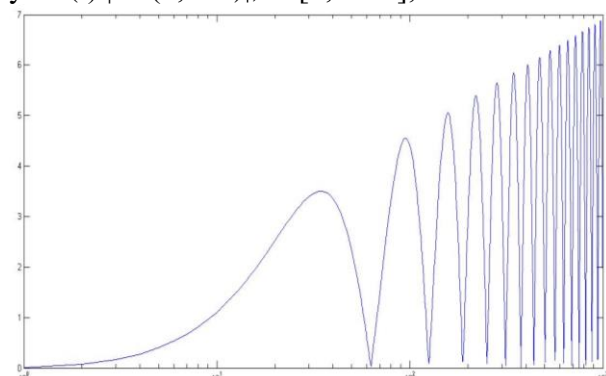
### Вариант 12

$$y=\ln(\cos^2(15\cdot t))-\sqrt{t}, t\in[1,1000], \Delta t=1$$



### Вариант 14

$$y=\ln(t)\cdot|\sin(0,05\cdot t)|, t\in[1,1000], \Delta t=1$$



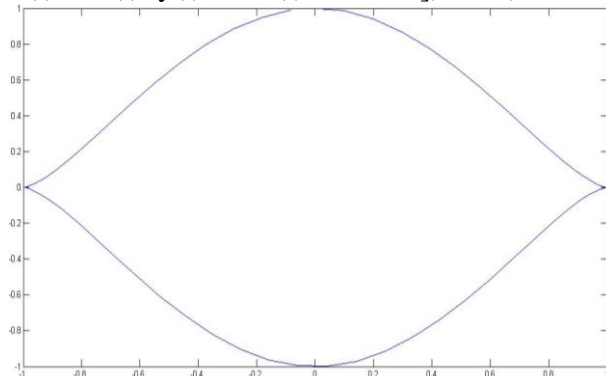
## Задача 1.8

Построить график параметрической функции  $y(x)$  с применением оператора `plot(...)`.

## ВАРИАНТЫ РЕШЕНИЙ

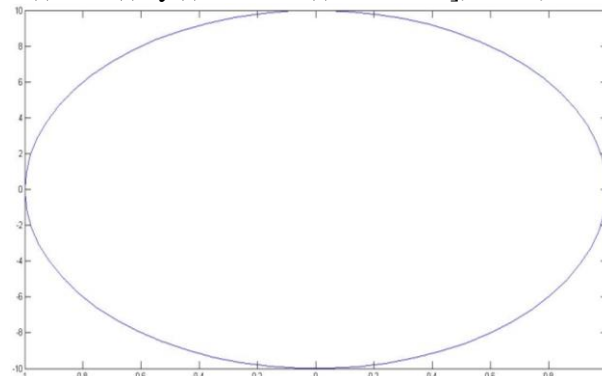
### Вариант 1

$$x(t)=\sin(t), y(t)=\cos^3(t), t \in [0, 2\pi], \Delta t=0,1$$



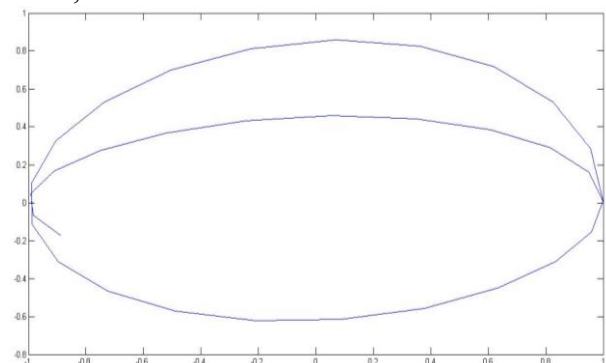
### Вариант 2

$$x(t)=\sin(t), y(t)=10 \cdot \cos(t), t \in [0, 2\pi], \Delta t=0,1$$



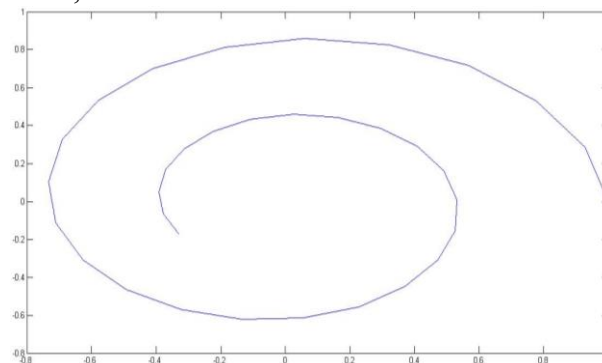
### Вариант 3

$$x(t)=\cos(10 \cdot t), y(t)=e^{-t} \cdot \sin(10 \cdot t), t \in [0, 2\pi], \Delta t=0,03$$



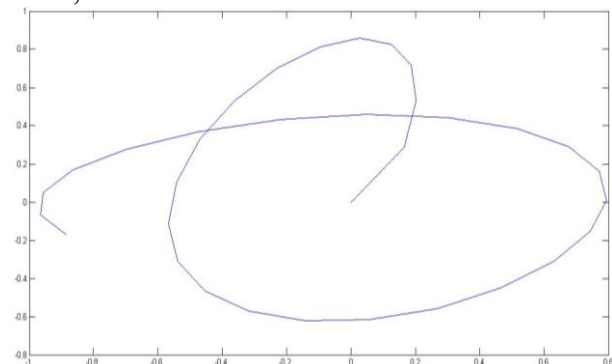
### Вариант 4

$$x(t)=e^{-t} \cdot \cos(10 \cdot t), y(t)=e^{-t} \cdot \sin(10 \cdot t), t \in [0, 1], \Delta t=0,03$$



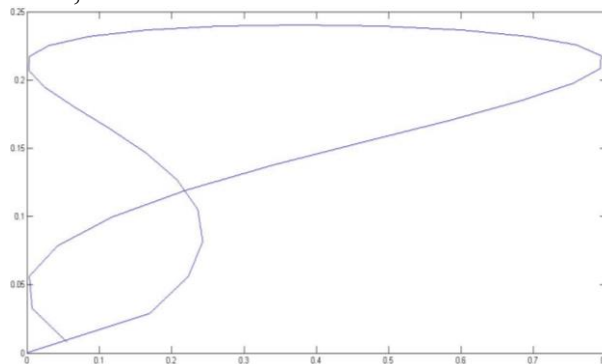
### Вариант 5

$$x(t)=\sqrt{t} \cdot \cos(10 \cdot t), y(t)=e^{-t} \cdot \sin(10 \cdot t), t \in [0, 1], \Delta t=0,03$$



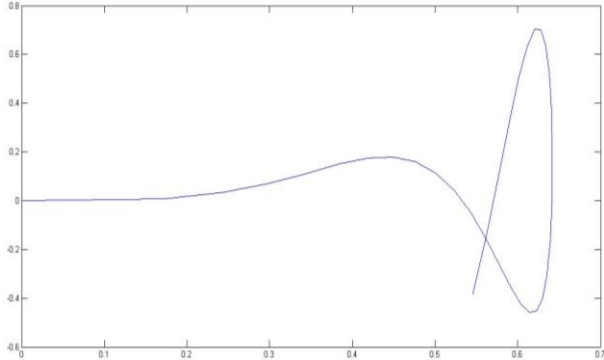
### Вариант 6

$$x(t)=\sqrt{t} \cdot \cos^2(5 \cdot t), y(t)=(1-t) \cdot \sin(t), t \in [0, 1], \Delta t=0,03$$



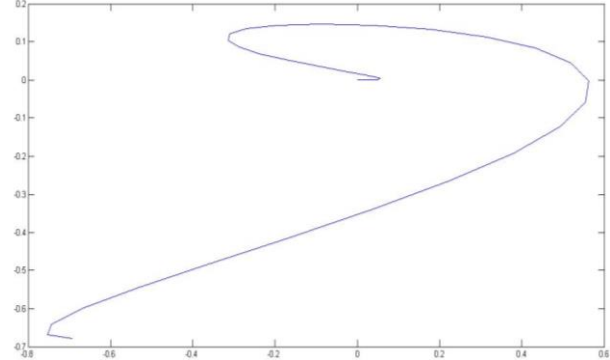
*Вариант 7*

$x(t)=\sqrt{t} \cdot \cos(t)$ ,  $y(t)=\sin^2(t) \cdot \sin(10 \cdot t)$ ,  $t \in [0,1]$ ,  
 $\Delta t=0,03$



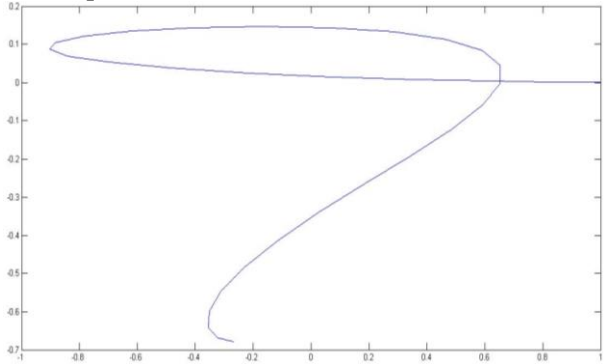
*Вариант 8*

$x(t)=\arctg(t) \cdot \cos(10 \cdot t)$ ,  $y(t)=\sin^2(t) \cdot \sin(5 \cdot t)$ ,  
 $t \in [0,1]$ ,  $\Delta t=0,03$



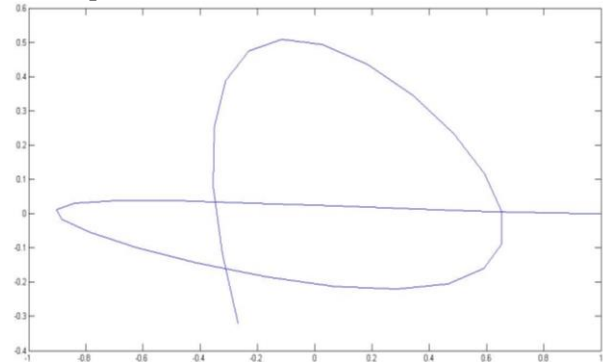
*Вариант 9*

$x(t)=\cos^2(t) \cdot \cos(10 \cdot t)$ ,  $y(t)=\sin^2(t) \cdot \sin(5 \cdot t)$ ,  
 $t \in [0,1]$ ,  $\Delta t=0,03$



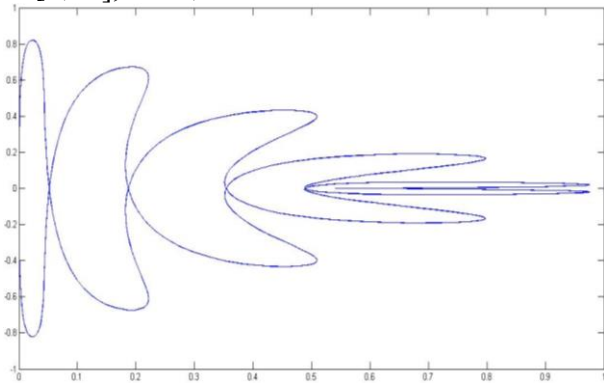
*Вариант 10*

$x(t)=\cos^2(t) \cdot \cos(10 \cdot t)$ ,  $y(t)=\sin^2(t) \cdot \sin(10 \cdot t)$ ,  
 $t \in [0,1]$ ,  $\Delta t=0,03$



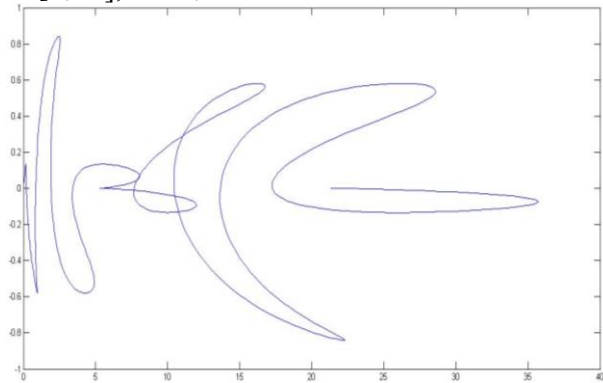
*Вариант 11*

$x(t)=\cos^2(t) \cdot \cos(\cos(10 \cdot t))$ ,  
 $y(t)=\sin^2(t) \cdot \sin(\sin(10 \cdot t))$ ,  
 $t \in [0,2\pi]$ ,  $\Delta t=0,01$



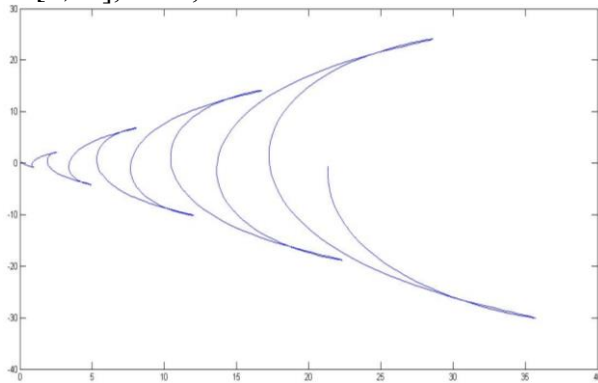
*Вариант 12*

$x(t)=\sin^2(t) \cdot \cos(\cos(10 \cdot t))$ ,  
 $y(t)=\sin^2(t) \cdot \sin(\sin(10 \cdot t))$ ,  
 $t \in [0,2\pi]$ ,  $\Delta t=0,01$



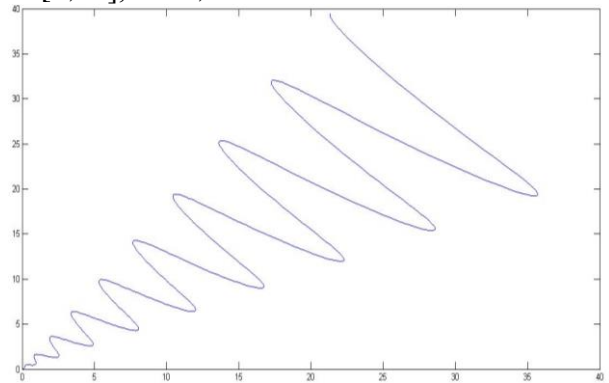
*Вариант 13*

$x(t)=t^2 \cdot \cos(\cos(5 \cdot t))$ ,  $y(t)=t^2 \cdot \sin(\sin(5 \cdot t))$ ,  
 $t \in [0, 2\pi]$ ,  $\Delta t=0,01$



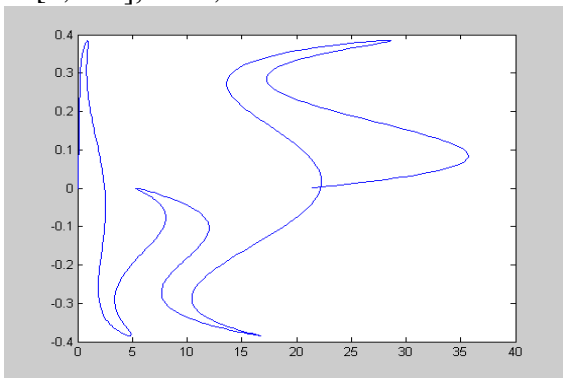
*Вариант 14*

$x(t)=t^2 \cdot \cos(\cos(5 \cdot t))$ ,  $y(t)=t^2 \cdot \cos(\sin(5 \cdot t))$ ,  
 $t \in [0, 2\pi]$ ,  $\Delta t=0,01$



*Вариант 15*

$x(t)=t^2 \cdot \cos(\cos(5t))$ ,  
 $y(t)=\sin^2(t) \cdot \cos(t)$ ,  
 $t \in [0, 2\pi]$ ,  $\Delta t=0,01$



## Задания для закрепления теоретического материала (3D-графика)

### Задача 2.1

С помощью оператора **plot3(...)** построить линейчатый график функции **z(x,y)**. Дополнительный параметр **S** задать с помощью табл. 8. Программу вычисления значений функции **z** выбрать в соответствии со своим вариантом.

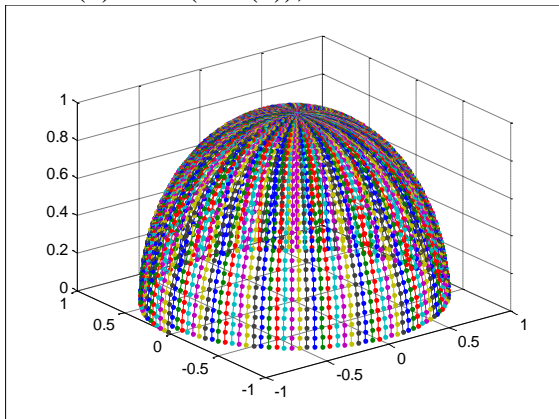
Таблица 8

Вариант	Параметры (цвет линии, тип маркера, тип линии)
1	Точка, сплошная
2	Розовый, кружок
3	Голубой, крестик, штрих-пунктирная
4	Красный, знак плюс, штриховая
5	Зеленый, треугольник вершиной влево
6	Синий, пунктирная
7	Черный, пятиконечная звезда, штрих-пунктирная
8	Точка, сплошная
9	Розовый, квадрат, штрих-пунктирная
10	Сплошная
11	Голубой, треугольник вершиной вниз, сплошная
12	Красный, треугольник вершиной вверх, пунктирная
13	Знак плюс, штриховая
14	Голубой, пятиконечная звезда, штрих-пунктирная
15	Зеленый, шестиконечная звезда, штриховая

### ВАРИАНТЫ РЕШЕНИЙ

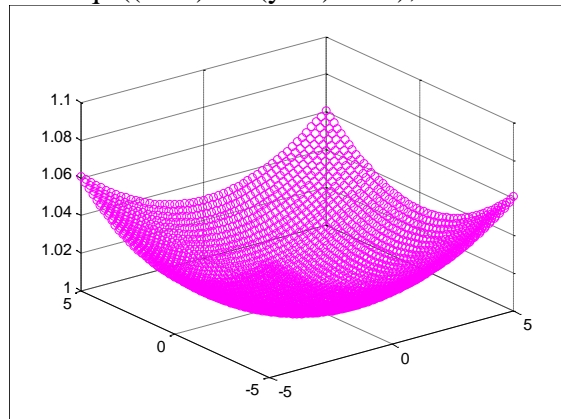
#### Вариант 1

```
u=[0:0.05:pi]'; v=[0:0.05:pi];
x=cos(u)*cos(v);
y=cos(u)*sin(v);
z=sin(u)*ones(size(v));
```



#### Вариант 2

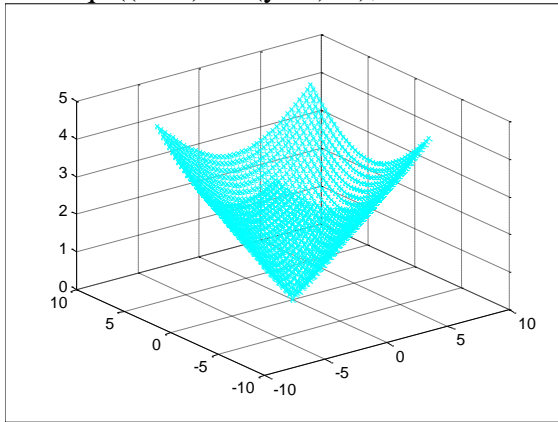
```
[x,y]=meshgrid(-5:0.2:5);
c=1; a=20; b=20;
A=a^2; B=b^2;
z=c*sqrt((x.^2)/A+(y.^2)/B+1);
```





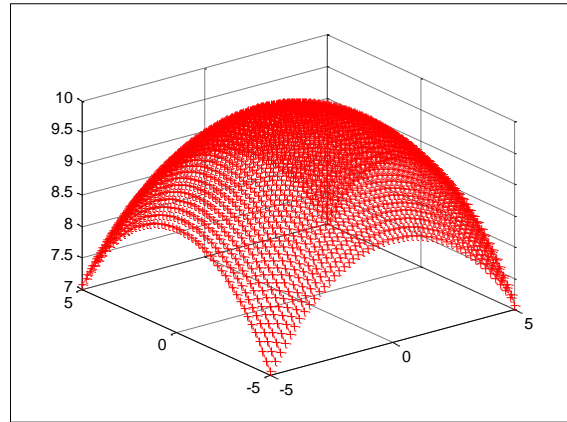
### Вариант 3

```
[x,y]=meshgrid(-2*pi:0.1*pi:2*pi);  
a=2; b=2; c=1;  
A=a^2; B=b^2;  
z=c*sqrt((x.^2)/A+(y.^2)/B);
```



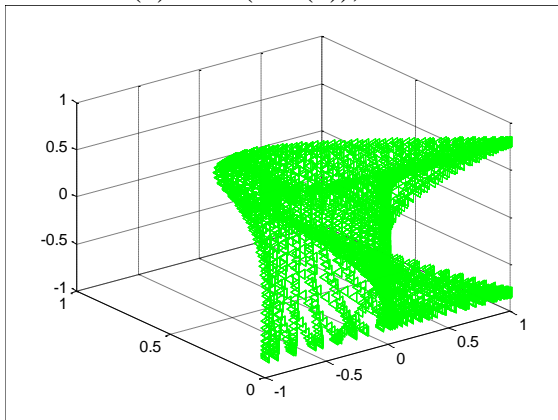
### Вариант 4

```
[x,y]=meshgrid(-5:0.2:5);  
z=sqrt(100-x.^2-y.^2);
```



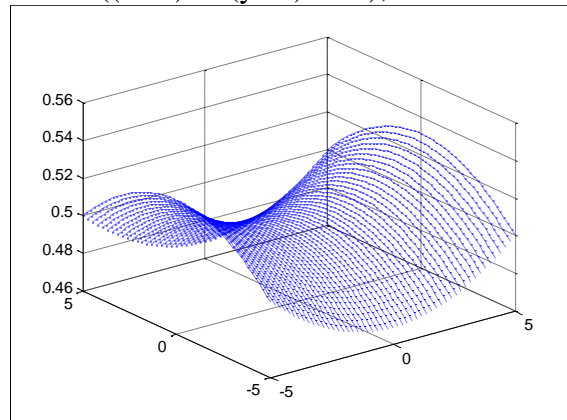
### Вариант 5

```
u=[-pi:0.02*pi:pi]'; v=[-pi:0.02*pi:pi];  
x=(sin(u).^2)*(sin(v.^2));  
y=0.9*(cos(u).^2)*(sin(v).^2);  
z=0.8*sin(u)*ones(size(v));
```



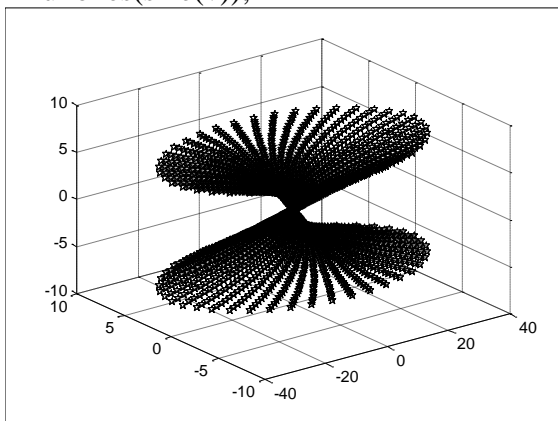
### Вариант 6

```
[x,y]=meshgrid(-5:0.2:5);  
a=20; b=20; c=1;  
A=a^2; B=b^2;  
z=0.5*((x.^2)/A-(y.^2)/B+1);
```



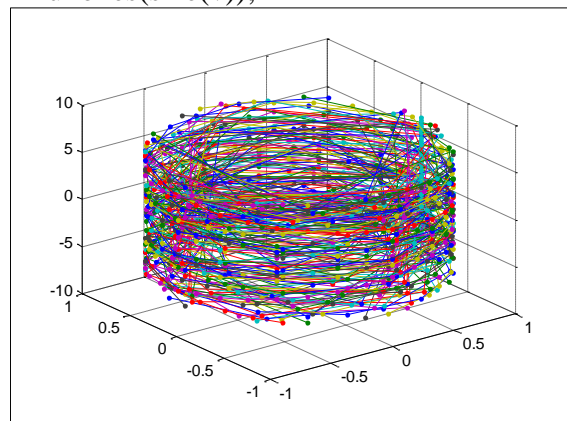
### Вариант 7

```
u=[-2*pi:0.05*pi:2*pi]';  
v=[-2*pi:0.05*pi:2*pi];  
X=(u.^2)*cos(v); Y=u*sin(v);  
Z=u*ones(size(v));
```



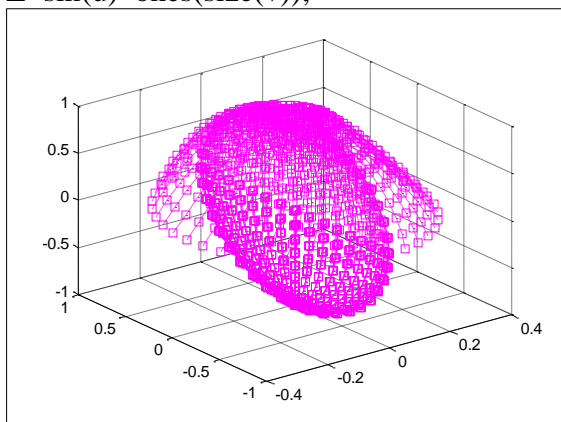
### Вариант 8

```
u=[-2*pi:0.1*pi:2*pi]';  
v=[-2*pi:0.2*pi:2*pi];  
X=cos(u*v); Y=sin(u*v);  
Z=u*ones(size(v));
```



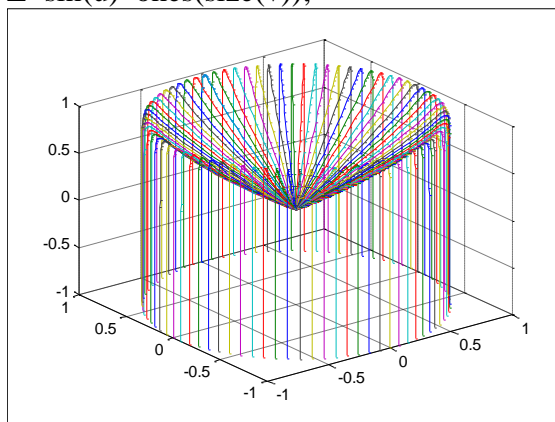
### Вариант 9

$u=[0:0.05*\pi:2*\pi]'$ ;  
 $v=[0:0.05*\pi:2*\pi]'$ ;  
 $X=\text{airy}(u)*\cos(v)$ ;  $Y=\cos(u)*\sin(v)$ ;  
 $Z=\sin(u)*\text{ones}(\text{size}(v))$ ;



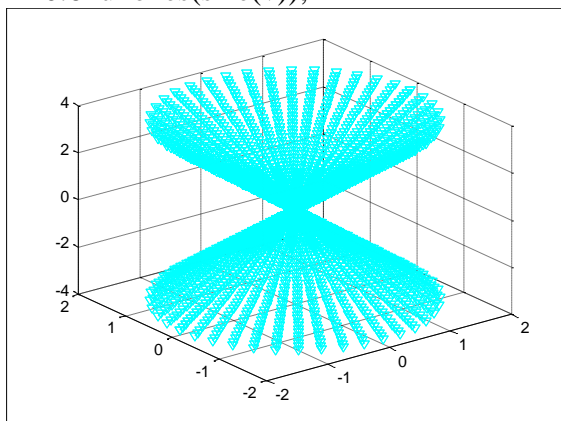
### Вариант 10

$u=[0:0.025*\pi:2*\pi]'$ ;  
 $v=[0:0.025*\pi:2*\pi]'$ ;  
 $X=\text{erf}(u)*\sin(v)$ ;  $Y=\text{erf}(u)*\cos(v)$ ;  
 $Z=\sin(u)*\text{ones}(\text{size}(v))$ ;



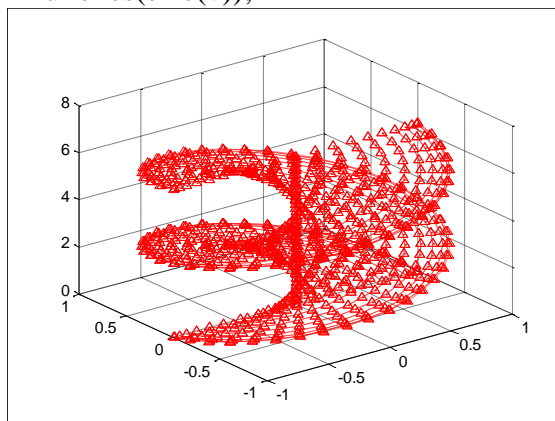
### Вариант 11

$u=[-2*\pi:0.05*\pi:2*\pi]'$ ;  
 $v=[-2*\pi:0.05*\pi:2*\pi]'$ ;  
 $X=0.3*u*\cos(v)$ ;  $Y=0.3*u*\sin(v)$ ;  
 $Z=0.6*u*\text{ones}(\text{size}(v))$ ;



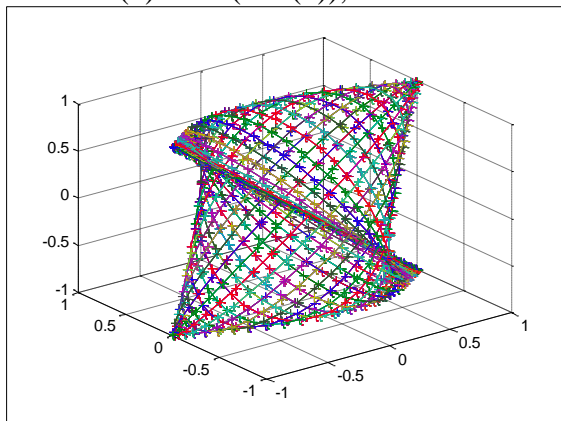
### Вариант 12

$u=[0:0.05*\pi:2*\pi]'$ ;  
 $v=[0:0.05*\pi:2*\pi]'$ ;  
 $X=\cos(u)*\cos(v)$ ;  $Y=\sin(u)*\cos(v)$ ;  
 $Z=u*\text{ones}(\text{size}(v))$ ;



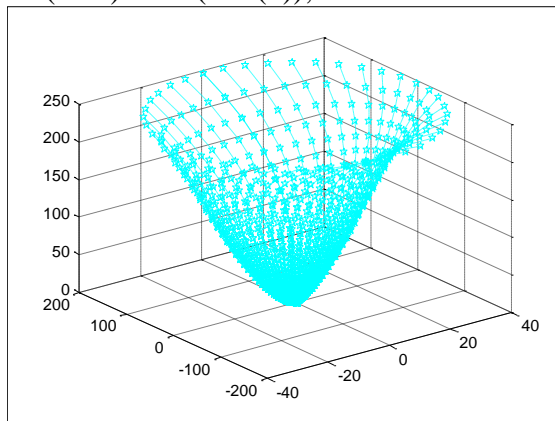
### Вариант 13

$u=[-2*\pi:0.05*\pi:2*\pi]'$ ;  
 $v=[-2*\pi:0.05*\pi:2*\pi]'$ ;  
 $R=1$ ;  $X=R*\sin(u)*\sin(v)$ ;  
 $Y=R*\cos(u)*\cos(v)$ ;  
 $Z=R*\sin(u)*\text{ones}(\text{size}(v))$ ;



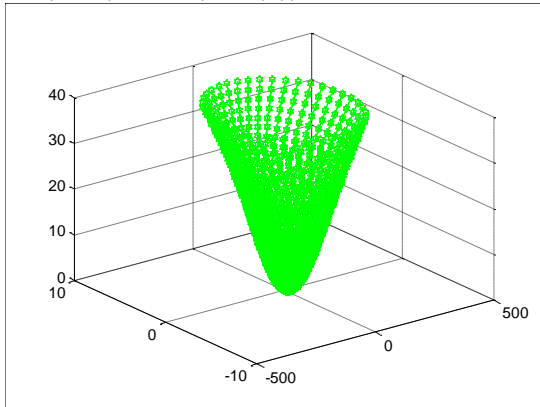
### Вариант 14

$u=[0:0.05*\pi:2*\pi]'$ ;  
 $v=[0:0.05*\pi:2*\pi]'$ ;  
 $X=(u.^2)*\sin(v)$ ;  
 $Y=\text{gamma}(u)*\cos(v)$ ;  
 $Z=(u.^3)*\text{ones}(\text{size}(v))$ ;



### Вариант 15

```
u=[-2*pi:0.05*pi:2*pi];  
v=[-2*pi:0.05*pi:2*pi];  
X=(u.^3)*cos(v);  
Y=u*sin(v);  
Z=(u.^2)*ones(size(v));
```



### Задача 2.2

С помощью оператора **contour3(...)** построить линейчатый график функции **z(x,y)**.

1. Количество линий на графике подобрать самостоятельно (главным критерием здесь является наглядность изображения).

Программу вычисления значений функции **z** выбрать в соответствии со своим вариантом. При построении графика используйте координатную сетку.

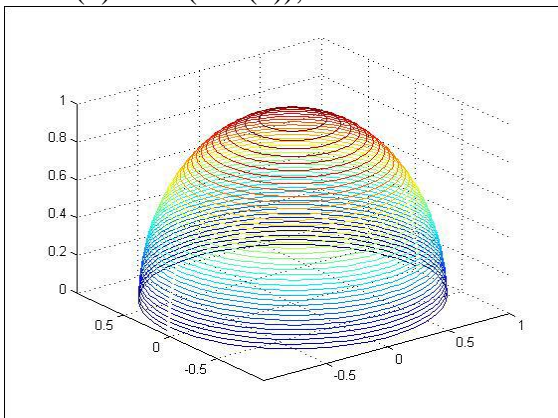
2. Увеличьте выбранное количество линий в десять раз и понаблюдайте за тем, как изменится структура поверхности.

3. Для вывода на бумажный носитель используйте первый вариант графика.

### ВАРИАНТЫ РЕШЕНИЙ

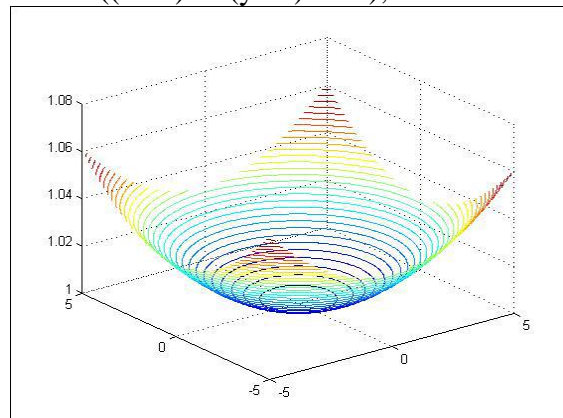
#### Вариант 1

```
u=[0:0.05:pi];  
v=[0:0.05:pi];  
x=cos(u)*cos(v); y=cos(u)*sin(v);  
z=sin(u)*ones(size(v));
```



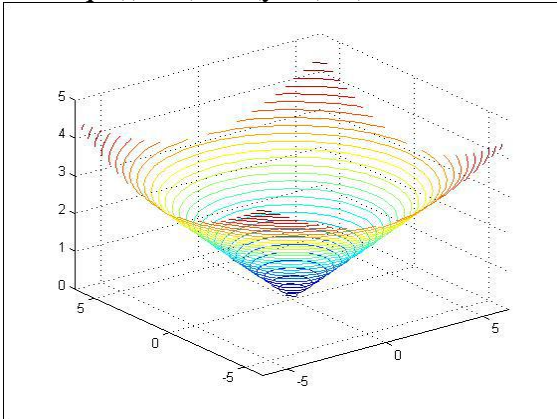
#### Вариант 2

```
[x,y]=meshgrid(-5:0.05:5);  
a=20; b=20; c=1;  
A=a^2; B=b^2;  
z=0.5*((x.^2)/A-(y.^2)/B+1);
```



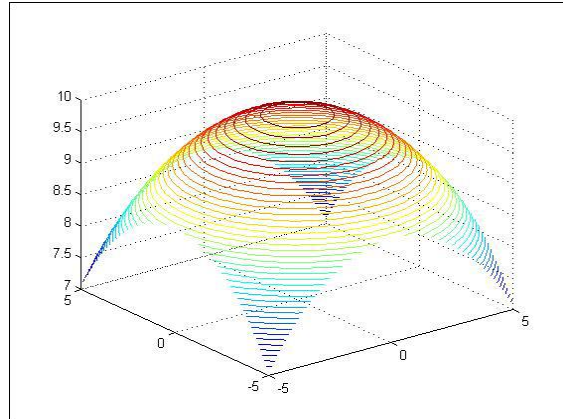
### Вариант 3

```
[x,y]=meshgrid(-2*pi:0.1*pi:2*pi);  
a=2; b=2; c=1; A=a^2; B=b^2;  
z=c*sqrt((x.^2)/A+(y.^2)/B);
```



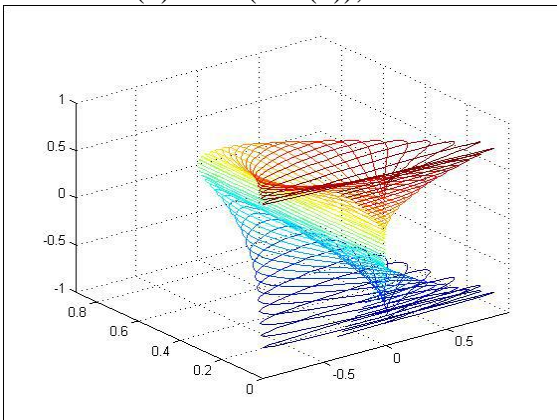
### Вариант 4

```
[x,y]=meshgrid(-5:0.2:5);  
z=sqrt(100-x.^2-y.^2);
```



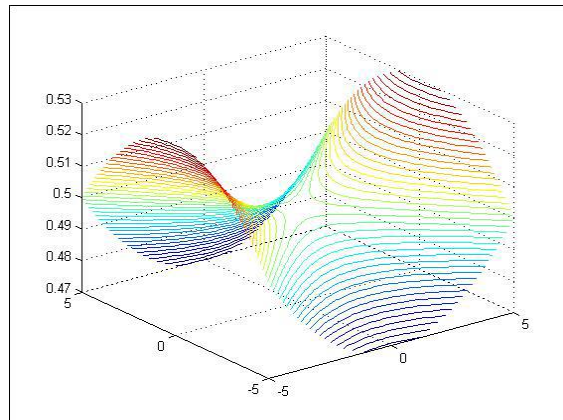
### Вариант 5

```
u=[-pi:0.001*pi:pi]';  
v=[-pi:0.02*pi:pi];  
X=(sin(u).^2)*(sin(v.^2));  
Y=0.9*(cos(u).^2)*(sin(v).^2);  
Z=0.8*sin(u)*ones(size(v));
```



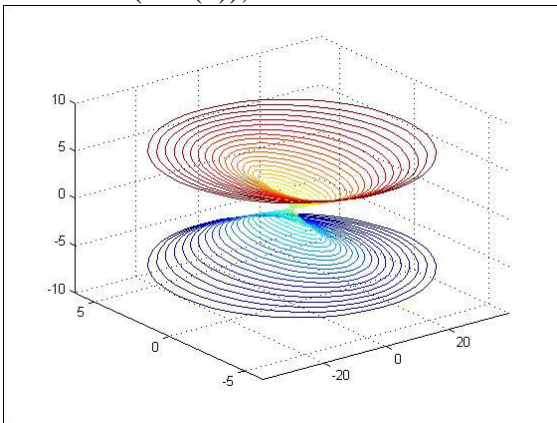
### Вариант 6

```
[x,y]=meshgrid(-5:0.2:5);  
a=20; b=20; c=1;  
A=a^2; B=b^2;  
z=0.5*((x.^2)/A-(y.^2)/B+1);
```



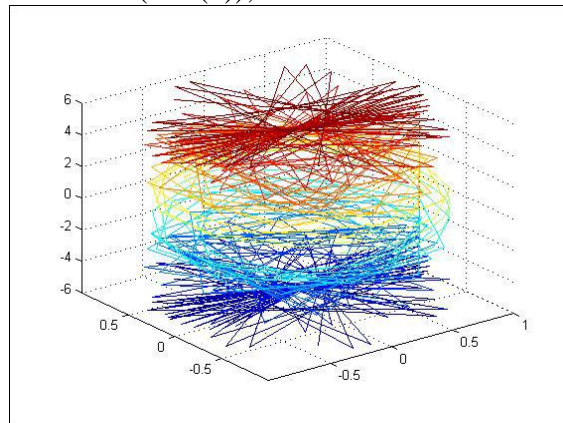
### Вариант 7

```
u=[-2*pi:0.05*pi:2*pi]';  
v=[-2*pi:0.05*pi:2*pi];  
X=(u.^2)*cos(v); Y=u*sin(v);  
Z=u*ones(size(v));
```



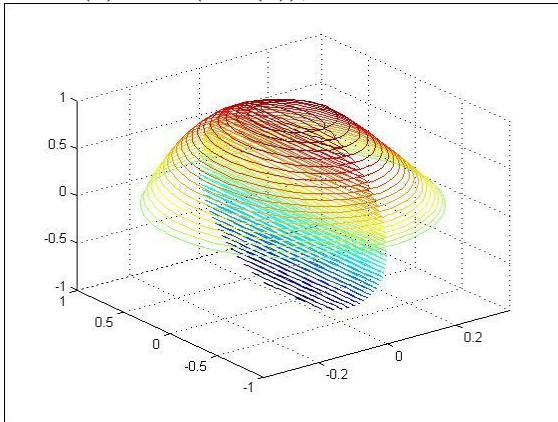
### Вариант 8

```
u=[-2*pi:0.1*pi:2*pi]';  
v=[-2*pi:0.2*pi:2*pi];  
X=cos(u*v); Y=sin(u*v);  
Z=u*ones(size(v));
```



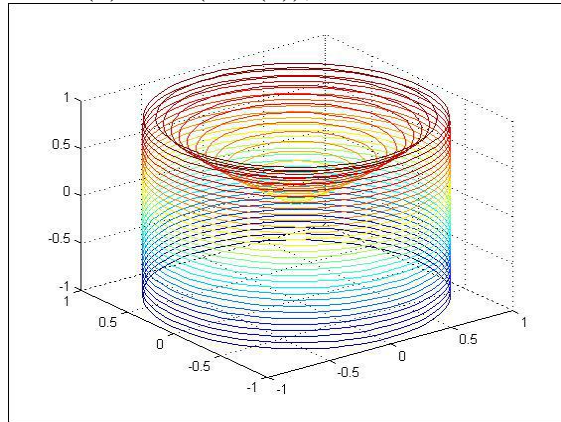
*Вариант 9*

$u=[0:0.025*\pi:2*\pi]'$ ;  
 $v=[0:0.025*\pi:2*\pi]$ ;  
 $X=\text{airy}(u)*\cos(v)$ ;  $Y=\cos(u)*\sin(v)$ ;  
 $Z=\sin(u)*\text{ones}(\text{size}(v))$ ;



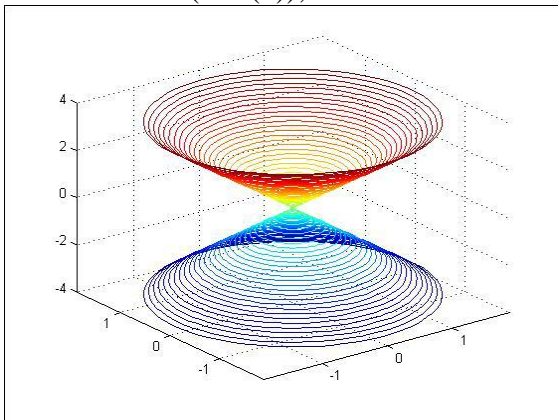
*Вариант 10*

$u=[0:0.025*\pi:2*\pi]'$ ;  
 $v=[0:0.025*\pi:2*\pi]$ ;  
 $X=\text{erf}(u)*\sin(v)$ ;  $Y=\text{erf}(u)*\cos(v)$ ;  
 $Z=\sin(u)*\text{ones}(\text{size}(v))$ ;



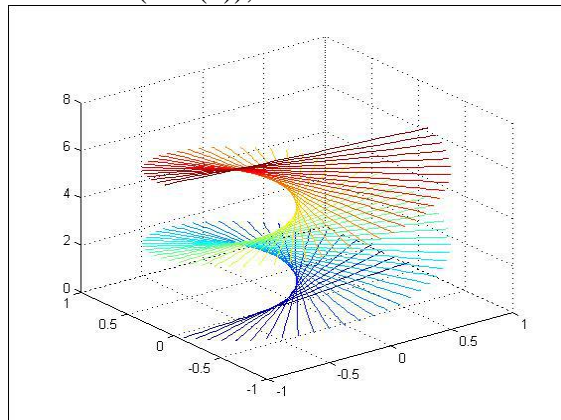
*Вариант 11*

$u=[-2*\pi:0.025*\pi:2*\pi]'$ ;  
 $v=[-2*\pi:0.025*\pi:2*\pi]$ ;  
 $X=0.3*u*\cos(v)$ ;  $Y=0.3*u*\sin(v)$ ;  
 $Z=0.6*u*\text{ones}(\text{size}(v))$ ;



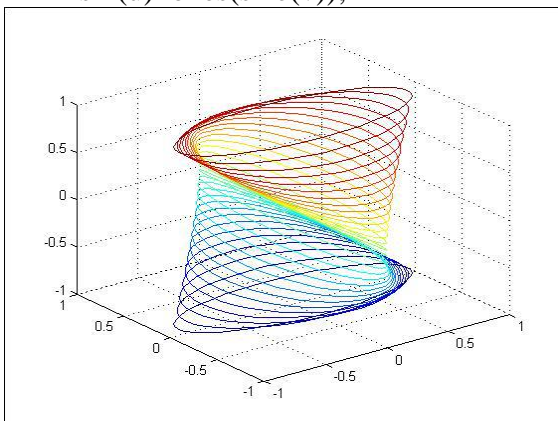
*Вариант 12*

$u=[0:0.025*\pi:2*\pi]'$ ;  
 $v=[0:0.025*\pi:2*\pi]$ ;  
 $X=\cos(u)*\cos(v)$ ;  $Y=\sin(u)*\cos(v)$ ;  
 $Z=u*\text{ones}(\text{size}(v))$ ;



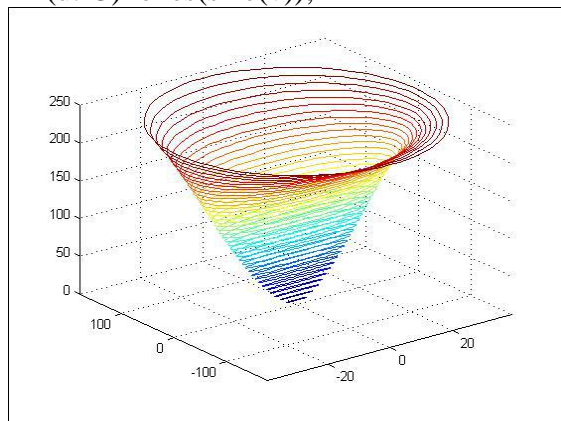
*Вариант 13*

$u=[-2*\pi:0.05*\pi:2*\pi]'$ ;  
 $v=[-2*\pi:0.05*\pi:2*\pi]$ ;  
 $R=1$ ;  $X=R*\sin(u)*\sin(v)$ ;  
 $Y=R*\cos(u)*\cos(v)$ ;  
 $Z=R*\sin(u)*\text{ones}(\text{size}(v))$ ;



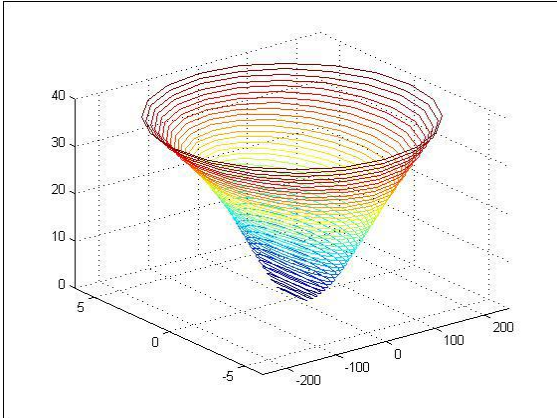
*Вариант 14*

$u=[0:0.025*\pi:2*\pi]'$ ;  
 $v=[0:0.025*\pi:2*\pi]$ ;  
 $X=(u.^2)*\sin(v)$ ;  
 $Y=\text{gamma}(u)*\cos(v)$ ;  
 $Z=(u.^3)*\text{ones}(\text{size}(v))$ ;



### Вариант 15

```
u=[-2*pi:0.1*pi:2*pi]';  
v=[-2*pi:0.1*pi:2*pi]';  
X=(u.^3)*cos(v);  
Y=u*sin(v);  
Z=(u.^2)*ones(size(v));
```



### Задача 2.3

С помощью операторов **mesh(...)** и **surf(...)** построить каркасную поверхность, заданную функцией **z(x,y)**, и соответствующий этой поверхности контурный график.

С помощью команды **colorbar** установить соответствие между цветом и значениями функции на каркасной поверхности.

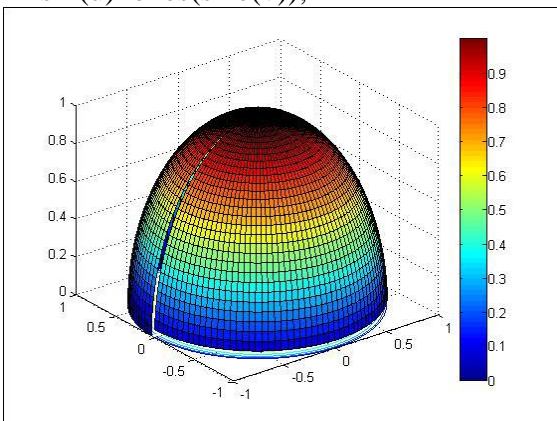
Программу вычисления значений функции **z** выбрать в соответствии со своим вариантом.

При построении графика использовать координатную сетку.

### ВАРИАНТЫ РЕШЕНИЙ

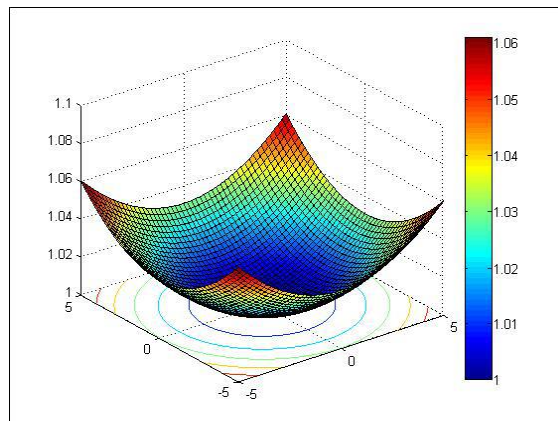
#### Вариант 1

```
u=[0:0.05*pi]';  
v=[0:0.05*pi]';  
X=cos(u)*cos(v);  
Y=cos(u)*sin(v);  
Z=sin(u)*ones(size(v));
```



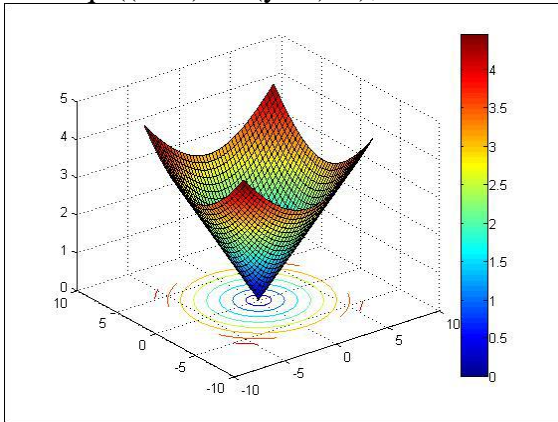
#### Вариант 2

```
[x,y]=meshgrid(-5:0.2:5);  
a=20; b=20; c=1;  
A=a^2; B=b^2;  
z=c*sqrt((x.^2)/A+(y.^2)/B+1);
```



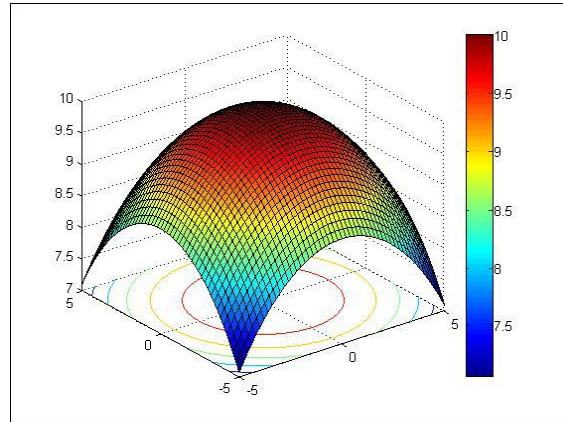
### Вариант 3

```
[x,y]=meshgrid(-2*pi:0.1*pi:2*pi);  
a=2; b=2; c=1;  
A=a^2; B=b^2;  
z=c*sqrt((x.^2)/A+(y.^2)/B);
```



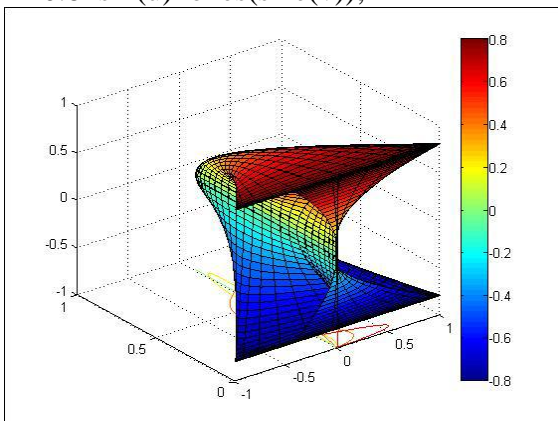
### Вариант 4

```
[x,y]=meshgrid(-5:0.2:5);  
z=sqrt(100-x.^2-y.^2);
```



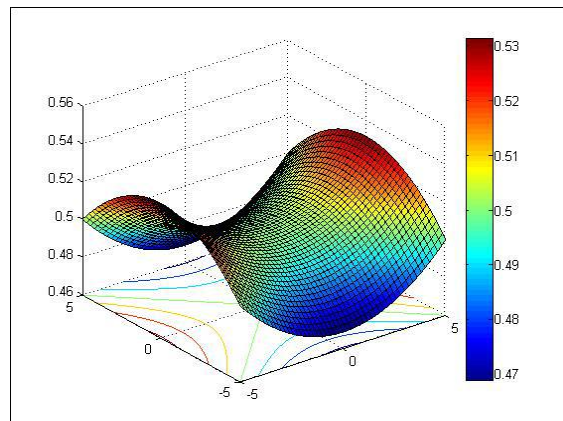
### Вариант 5

```
u=[-pi:0.02*pi:pi]';  
v=[-pi:0.02*pi:pi];  
X=(sin(u).^2)*(sin(v.^2));  
Y=0.9*(cos(u).^2)*(sin(v).^2);  
Z=0.8*sin(u)*ones(size(v));
```



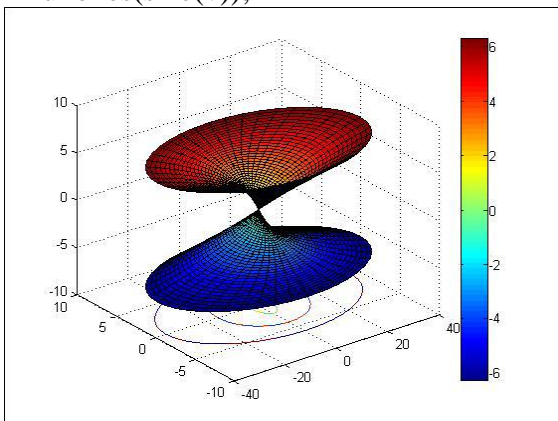
### Вариант 6

```
[x,y]=meshgrid(-5:0.2:5);  
a=20; b=20; c=1;  
A=a^2; B=b^2;  
z=0.5*((x.^2)/A-(y.^2)/B+1);
```



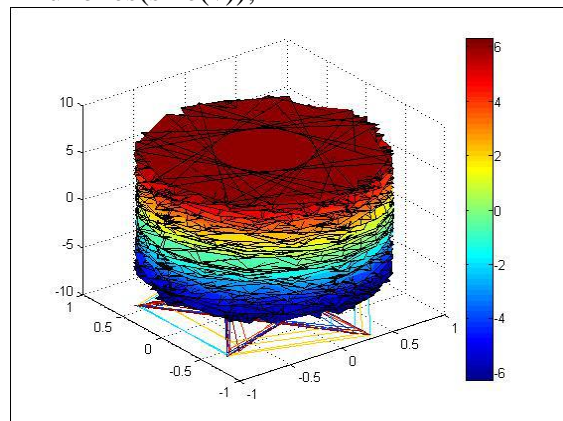
### Вариант 7

```
u=[-2*pi:0.05*pi:2*pi]';  
v=[-2*pi:0.05*pi:2*pi];  
X=(u.^2)*cos(v); Y=u*sin(v);  
Z=u*ones(size(v));
```



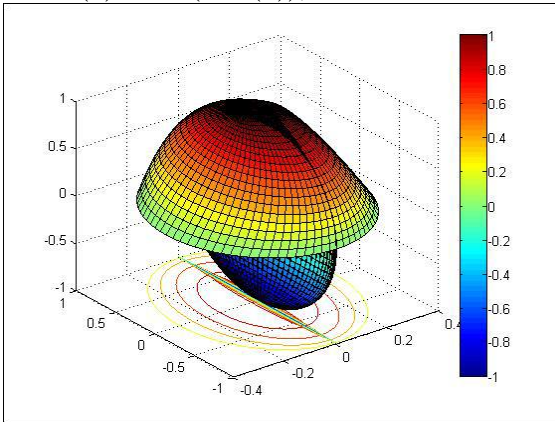
### Вариант 8

```
u=[-2*pi:0.1*pi:2*pi]';  
v=[-2*pi:0.2*pi:2*pi];  
X=cos(u*v); Y=sin(u*v);  
Z=u*ones(size(v));
```



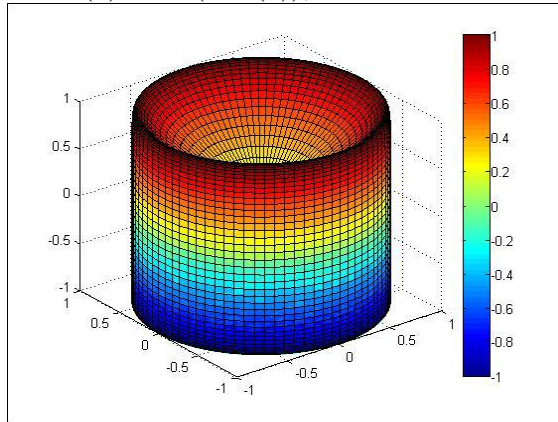
### Вариант 9

```
u=[0:0.025*pi:2*pi]';  
v=[0:0.025*pi:2*pi]';  
X=airy(u)*cos(v); Y=cos(u)*sin(v);  
Z=sin(u)*ones(size(v));
```



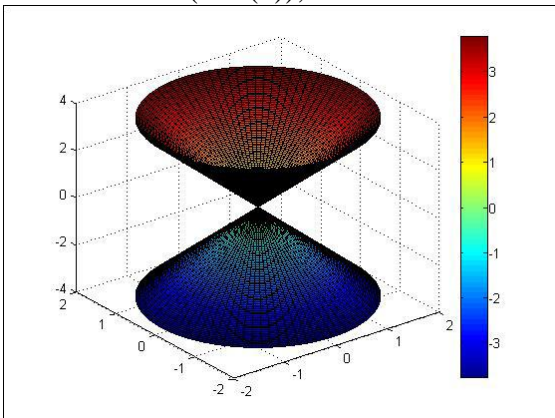
### Вариант 10

```
u=[0:0.025*pi:2*pi]';  
v=[0:0.025*pi:2*pi]';  
X=erf(u)*sin(v); Y=erf(u)*cos(v);  
Z=sin(u)*ones(size(v));
```



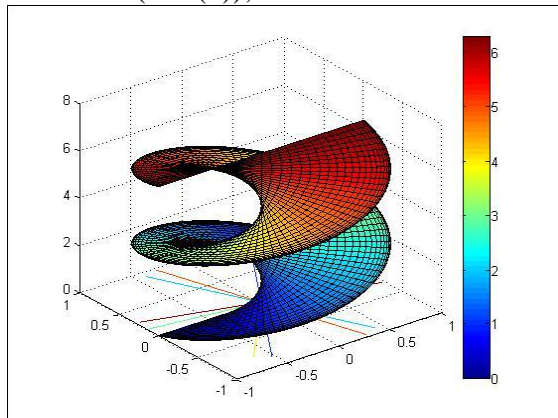
### Вариант 11

```
u=[-2*pi:0.025*pi:2*pi]';  
v=[-2*pi:0.025*pi:2*pi]';  
X=0.3*u*cos(v); Y=0.3*u*sin(v);  
Z=0.6*u*ones(size(v));
```



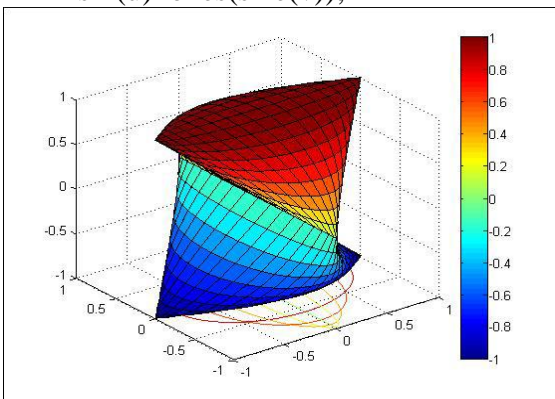
### Вариант 12

```
u=[0:0.025*pi:2*pi]';  
v=[0:0.025*pi:2*pi]';  
X=cos(u)*cos(v); Y=sin(u)*cos(v);  
Z=u*ones(size(v));
```



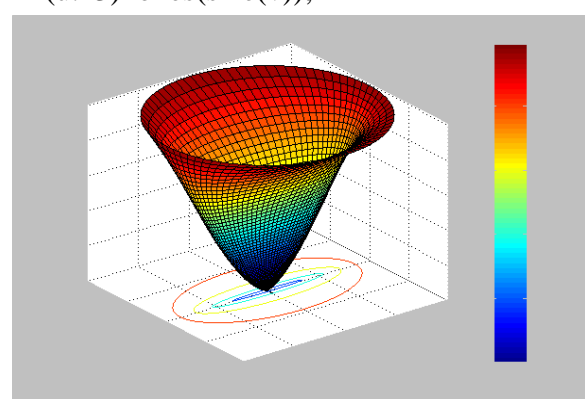
### Вариант 13

```
u=[-2*pi:0.05*pi:2*pi]';  
v=[-2*pi:0.05*pi:2*pi]';  
R=1; X=R*sin(u)*sin(v);  
Y=R*cos(u)*cos(v);  
Z=R*sin(u)*ones(size(v));
```



### Вариант 14

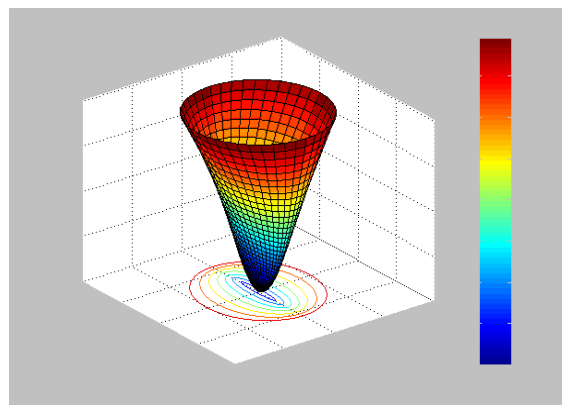
```
u=[0:0.025*pi:2*pi]';  
v=[0:0.025*pi:2*pi]';  
X=(u.^2)*sin(v);  
Y=gamma(u)*cos(v);  
Z=(u.^3)*ones(size(v));
```





### Вариант 15

```
u=[-2*pi:0.05*pi:2*pi]';
v=[-2*pi:0.05*pi:2*pi]';
X=(u.^3)*cos(v);
Y=u*sin(v);
Z=(u.^2)*ones(size(v));
```



### Задача 2.4

С помощью оператора **surf(...)** и команды **shading interp** построить плавно залитую цветом поверхность, заданную функцией **z(x,y)**.

Цветовую палитру окраски поверхности выбрать с помощью оператора **colormap(...)** и табл. 9, добившись наиболее реалистичного вида поверхности.

С применением команды **colorbar** установить соответствие между цветом и значениями функции на каркасной поверхности.

Программу вычисления значений функции **z** выбрать в соответствии со своим вариантом.

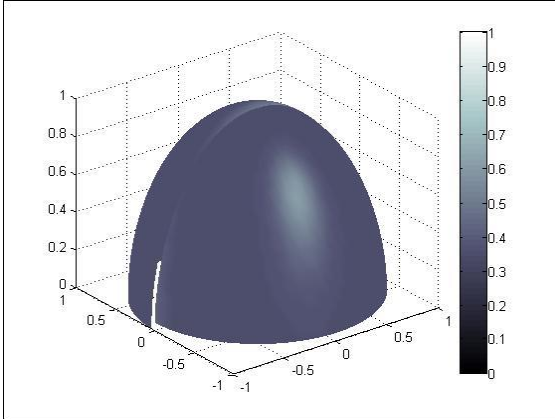
Таблица 9

Вариант	Палитра	Изменение цвета
1	bone	Похожа на палитру gray, но с легким оттенком синего
2	colorcube	Каждый цвет изменяется от темного к яркому
3	cool	Оттенки голубого и пурпурного цветов
4	copper	Оттенки медного цвета
5	flag	Циклические изменения: красный–белый–синий–черный
6	gray	Оттенки серого
7	hot	Плавное изменение: черный–красный–оранжевый–желтый–белый
8	hot	Плавное изменение: черный–красный–оранжевый–желтый–белый
9	jet	Плавное изменение: синий–голубой–зеленый–желтый–красный
10	hsv	Плавное изменение (как цвета радуги)
11	pink	Похожа на палитру gray, но с легким оттенком коричневого
12	pink	Похожа на палитру gray, но с легким оттенком коричневого
13	prism	Циклические изменения: красный–оранжевый–желтый–зеленый–синий–фиолетовый
14	spring	Оттенки пурпурного и желтого
15	summer	Оттенки зеленого и желтого

## ВАРИАНТЫ РЕШЕНИЙ

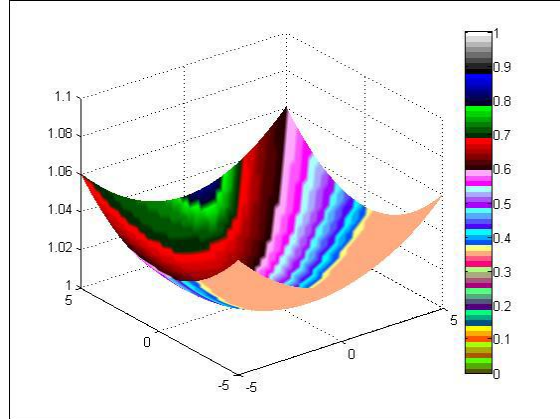
### Вариант 1

```
u=[0:0.05:pi]';
v=[0:0.05:pi]';
X=cos(u)*cos(v); Y=cos(u)*sin(v);
Z=sin(u)*ones(size(v));
```



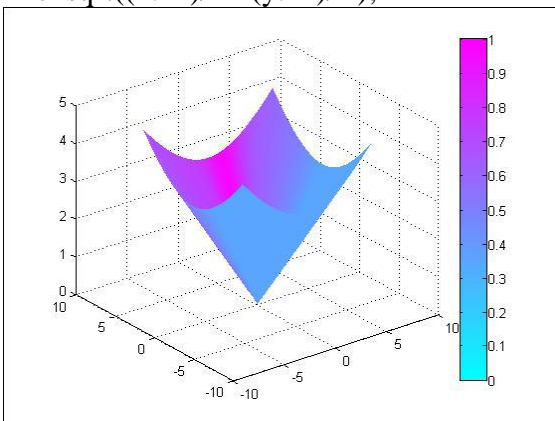
### Вариант 2

```
[x,y]=meshgrid(-5:0.2:5);
a=20; b=20; c=1;
A=a^2; B=b^2;
z=c*sqrt((x.^2)/A+(y.^2)/B+1);
```



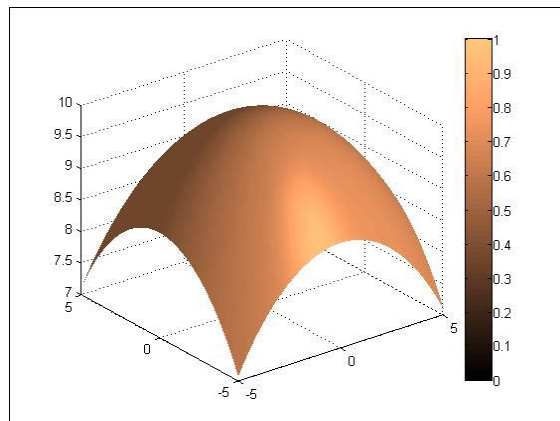
### Вариант 3

```
[x,y]=meshgrid(-2*pi:0.1*pi:2*pi);
a=2; b=2; c=1; A=a^2; B=b^2;
z=c*sqrt((x.^2)/A+(y.^2)/B);
```



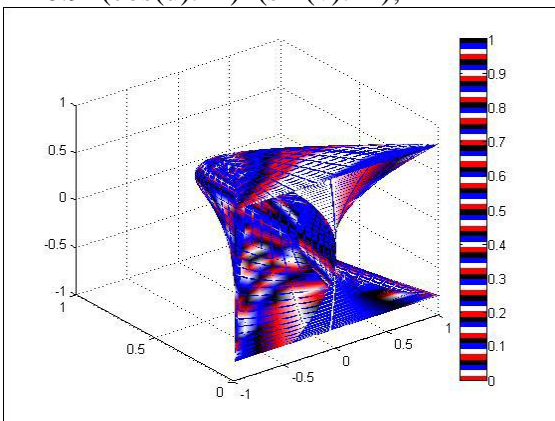
### Вариант 4

```
[x,y]=meshgrid(-5:0.2:5);
z=sqrt(100-x.^2-y.^2);
```



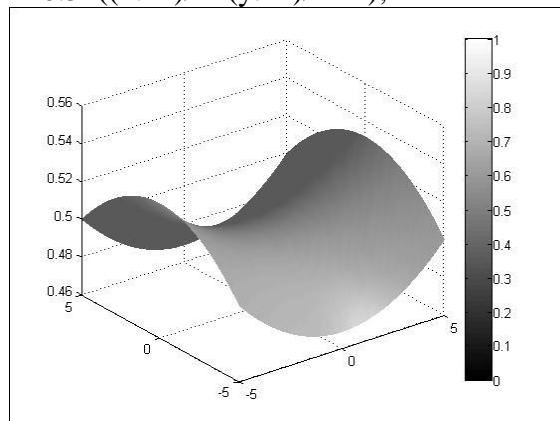
### Вариант 5

```
u=[-pi:0.02*pi:pi]';
v=[-pi:0.02*pi:pi]';
X=(sin(u).^2)*(sin(v.^2));
Y=0.9*(cos(u).^2)*(sin(v).^2);
```



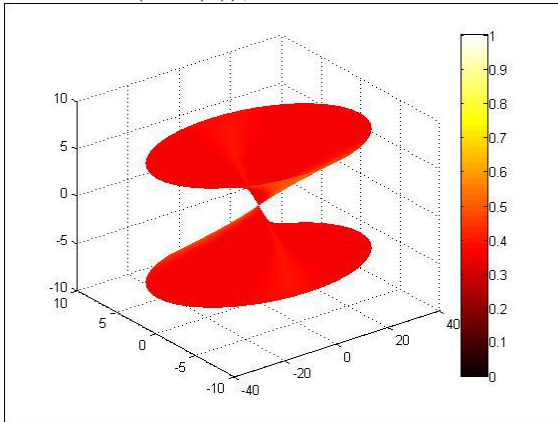
### Вариант 6

```
[x,y]=meshgrid(-5:0.2:5);
a=20; b=20; c=1;
A=a^2; B=b^2;
z=0.5*((x.^2)/A-(y.^2)/B+1);
```



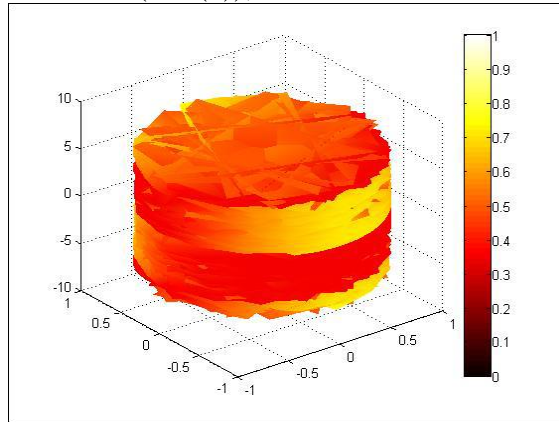
*Вариант 7*

$u = [-2\pi:0.05\pi:2\pi]'$ ;  
 $v = [-2\pi:0.05\pi:2\pi]'$ ;  
 $X = (u.^2) \cdot \cos(v)$ ;  $Y = u \cdot \sin(v)$ ;  
 $Z = u \cdot \text{ones}(\text{size}(v))$ ;



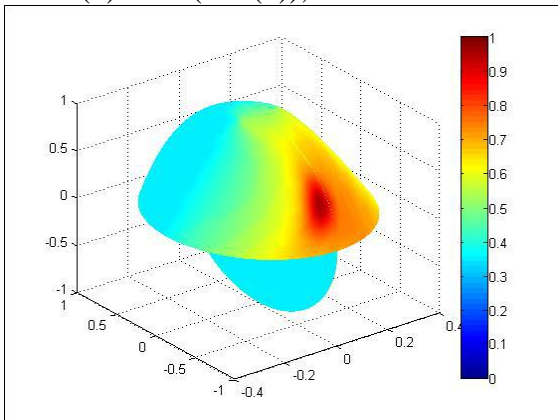
*Вариант 8*

$u = [-2\pi:0.1\pi:2\pi]'$ ;  
 $v = [-2\pi:0.2\pi:2\pi]'$ ;  
 $X = \cos(u \cdot v)$ ;  $Y = \sin(u \cdot v)$ ;  
 $Z = u \cdot \text{ones}(\text{size}(v))$ ;



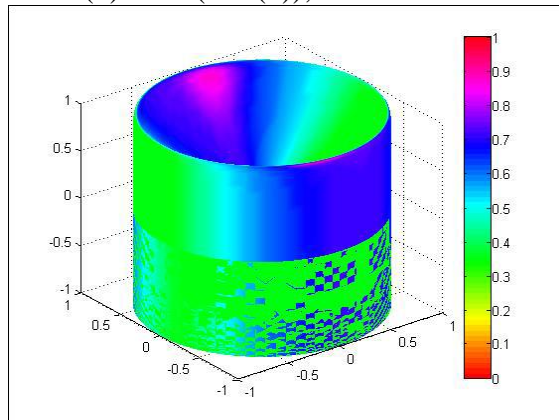
*Вариант 9*

$u = [0:0.025\pi:2\pi]'$ ;  
 $v = [0:0.025\pi:2\pi]'$ ;  
 $X = \text{airy}(u) \cdot \cos(v)$ ;  $Y = \cos(u) \cdot \sin(v)$ ;  
 $Z = \sin(u) \cdot \text{ones}(\text{size}(v))$ ;



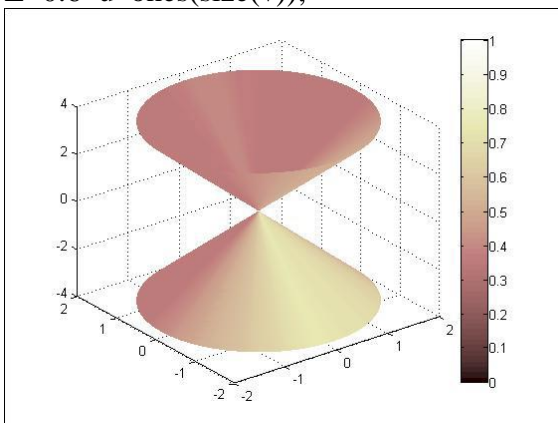
*Вариант 10*

$u = [0:0.025\pi:2\pi]'$ ;  
 $v = [0:0.025\pi:2\pi]'$ ;  
 $X = \text{erf}(u) \cdot \sin(v)$ ;  $Y = \text{erf}(u) \cdot \cos(v)$ ;  
 $Z = \sin(u) \cdot \text{ones}(\text{size}(v))$ ;



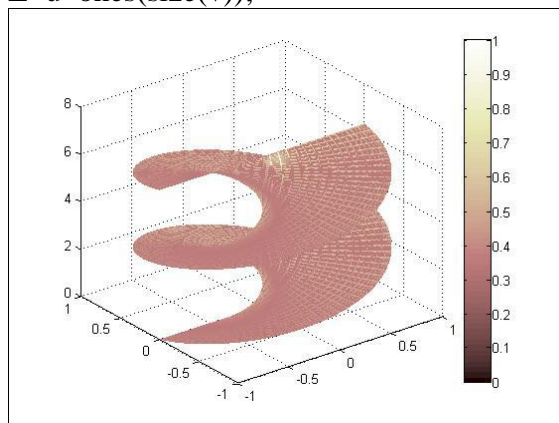
*Вариант 11*

$u = [-2\pi:0.025\pi:2\pi]'$ ;  
 $v = [-2\pi:0.025\pi:2\pi]'$ ;  
 $X = 0.3 \cdot u \cdot \cos(v)$ ;  $Y = 0.3 \cdot u \cdot \sin(v)$ ;  
 $Z = 0.6 \cdot u \cdot \text{ones}(\text{size}(v))$ ;



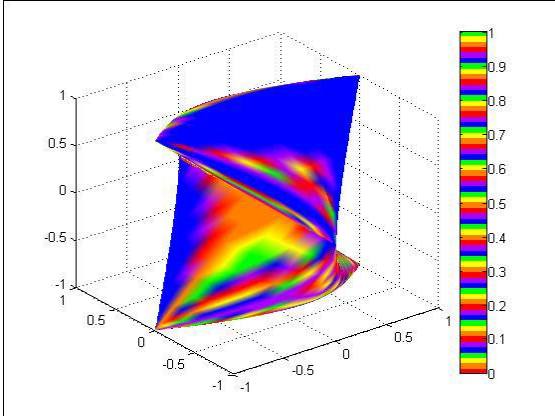
*Вариант 12*

$u = [0:0.025\pi:2\pi]'$ ;  
 $v = [0:0.025\pi:2\pi]'$ ;  
 $X = \cos(u) \cdot \cos(v)$ ;  $Y = \sin(u) \cdot \cos(v)$ ;  
 $Z = u \cdot \text{ones}(\text{size}(v))$ ;



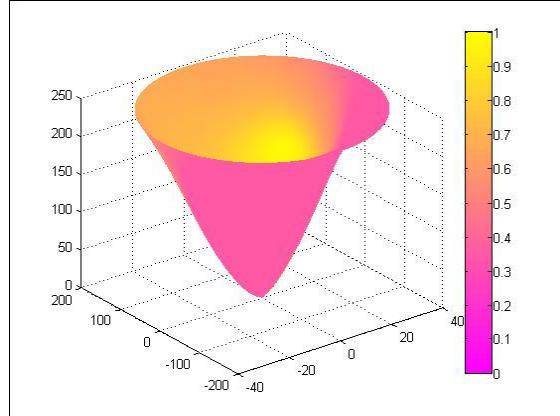
### Вариант 13

```
u=[-2*pi:0.05*pi:2*pi]';  
v=[-2*pi:0.05*pi:2*pi];  
R=1;  
X=R*sin(u)*sin(v);  
Y=R*cos(u)*cos(v);  
Z=R*sin(u)*ones(size(v));
```



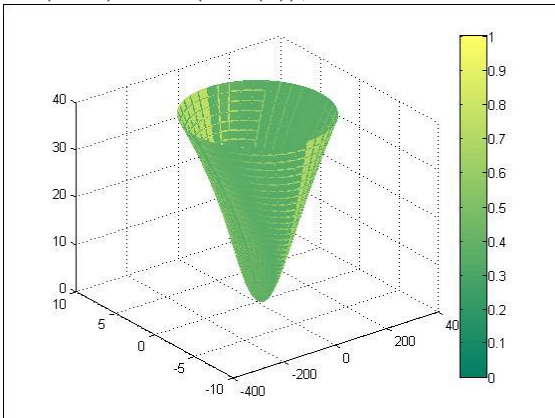
### Вариант 14

```
u=[0:0.025*pi:2*pi]';  
v=[0:0.025*pi:2*pi];  
X=(u.^2)*sin(v);  
Y=gamma(u)*cos(v);  
Z=(u.^3)*ones(size(v));
```



### Вариант 15

```
u=[-2*pi:0.05*pi:2*pi]';  
v=[-2*pi:0.05*pi:2*pi];  
X=(u.^3)*cos(v);  
Y=u*sin(v);  
Z=(u.^2)*ones(size(v));
```



## ЛИТЕРАТУРА

1. Ануфриев И. Е. Самоучитель MatLab 5.3/6.x.- СПб. : БХВ-Петербург, 2002. – 710 с.
2. Дьяконов В.П. Компьютерная математика. Теория и практика.- М.: Нолидж, 2001. – 1295с.
3. Дьяконов В.П., Абраменкова И.В. MATLAB. Обработка сигналов и изображений. Специальный справочник.- СПб.: ПИТЕР, 2002. – 602с.
4. Бондаренко В.Ф., Дубовец В.Д. MATLAB. Основы работы и программирования. Компьютерная математика. Учебный курс / В.Ф. Бондаренко, В.Д. Дубовец.– Минск: Харвест, 2010. – 256с.

*Учебное издание*

**Дубовец Валерий Денисович  
Столер Владимир Алексеевич  
Бондаренко Валентин Федорович**

## **ПОСТРОЕНИЕ ГРАФИЧЕСКИХ МОДЕЛЕЙ В СРЕДЕ MATLAB**

Учебно-методическое пособие  
по дисциплине «Инженерная компьютерная графика»

Редактор  
Корректор  
Компьютерная вёрстка

Подписано в печать  
Гарнитура «Таймс».  
Уч.-изд. л. 4,3.

Формат  
Отпечатано на ризографе.  
Тираж 250 экз.

Бумага офсетная.  
Усл. печ. л.  
Заказ .

---

Издатель и полиграфическое исполнение: учреждение образования  
"Белорусский государственный университет информатики и радиоэлектроники"  
ЛИ №02330/0494371 от 16.03.2009. ЛИ №02330/0494175 от 03.04.2009.  
220013, Минск, П. Бровки, 6