

ВВОДНАЯ ЛАБОРАТОРНАЯ РАБОТА «ОСВОЕНИЕ ИНТЕРФЕЙСА MATLAB»

Цель работы:

Получение общего представления о математическом пакете Matlab – особенностей интерфейса, функциональных основных возможностей, формирования навыков практической работы в среде Matlab, математических вычислений, моделирования, разработки приложений и анализа данных.

Разобранные примеры:

Был разобран пример создания векторов (пример 1).

Пример 1– Задание векторов

```
vectorStroka = [1 2 3]

vectorStroka =
    1     2     3

vectorStolbec = [1;2;3]

vectorStolbec =
    1
    2
    3
```

Были созданы несколько матриц (пример 2).

Пример 2 – Задание матриц

```
matrixfat = [2 4 8; 3 9 27]

matrixfat =
    2     4     8
    3     9    27
```

```
matrixslim = [1 2; 3 4; 7 8; 9 12]
```

```
matrixslim =
```

```

1      2
3      4
7      8
9     12
```

Было разобрано задание вектора и вычисление его значения (пример 3).

Пример 3 – Задание вектора и вычисление вектора

```
vec1 = 0:0.5:5
```

```
vec1 =
```

```

      0      0.5000      1.0000      1.5000      2.0000      2.5000      3.0000
3.5000      4.0000      4.5000      5.0000
```

```
vec2 = cos(vec1)
```

```
vec2 =
```

```

      1.0000      0.8776      0.5403      0.0707     -0.4161     -0.8011     -0.9900     -
0.9365     -0.6536     -0.2108      0.2837
```

Был разобран способ использования функции rand (пример 4).

Пример 4 – Использование функции rand

```
rmat = rand(2)
```

```
rmat =
```

```

0.8147      0.1270
0.9058      0.9134
```

```
rmat = rand([3 2])
```

```
rmat =
```

```

0.6324      0.5469
0.0975      0.9575
0.2785      0.9649
```

Также был разобран пример использования функции randi (пример 5).

Пример 5 – Использование функции randi

```
rmat = randi([-5 5], 5)
```

```
rmat =
```

```

-4    -4    2    3    2
 5    -1   -5    3   -5
 5     5    4   -1   -2
 0     3    5    2   -5
 3     5    2   -4   -4

```

```
rmat = randi([-3 7], 3, 6)
```

```
rmat =
```

```

 6     7     1    -1     4     0
 4    -3     5     2     4     4
 0     1     5     1     5     4

```

Были разобраны способы задания матриц, а также способы обращения к её элементам (пример 6).

Пример 6 – Задание матрицы и обращение к ее элементам

```
M=[1 2 3; 6 5 4; 78 88 89]
```

```
M =
```

```

 1     2     3
 6     5     4
78    88    89

```

```
M(2,2)=44
```

```
M =
```

```

 1     2     3
 6    44     4
78    88    89

```

```
M(3,:)=99
```

```
M =
```

```

 1     2     3
 6    44     4
99    99    99

```

```
M(:,3)=12
```

```
M =
```

```

 1     2    12
 6    44    12
99    99    12

```

Была разобрана возможность изменения фрагмента матрицы (пример 7).

Пример 7 – Изменение фрагмента матрицы

```
M(1:3, 1:2)=3
```

M =

3	3	12
3	3	12
3	3	12

Был построен график с помощью функции plot (рисунок 1, пример 8).

Пример 8 – Построение графика с использованием функции plot

```
ex = [0:0.007:4.9];  
ey = exp(-ex).*sin(8*ex);  
plot(ey)
```

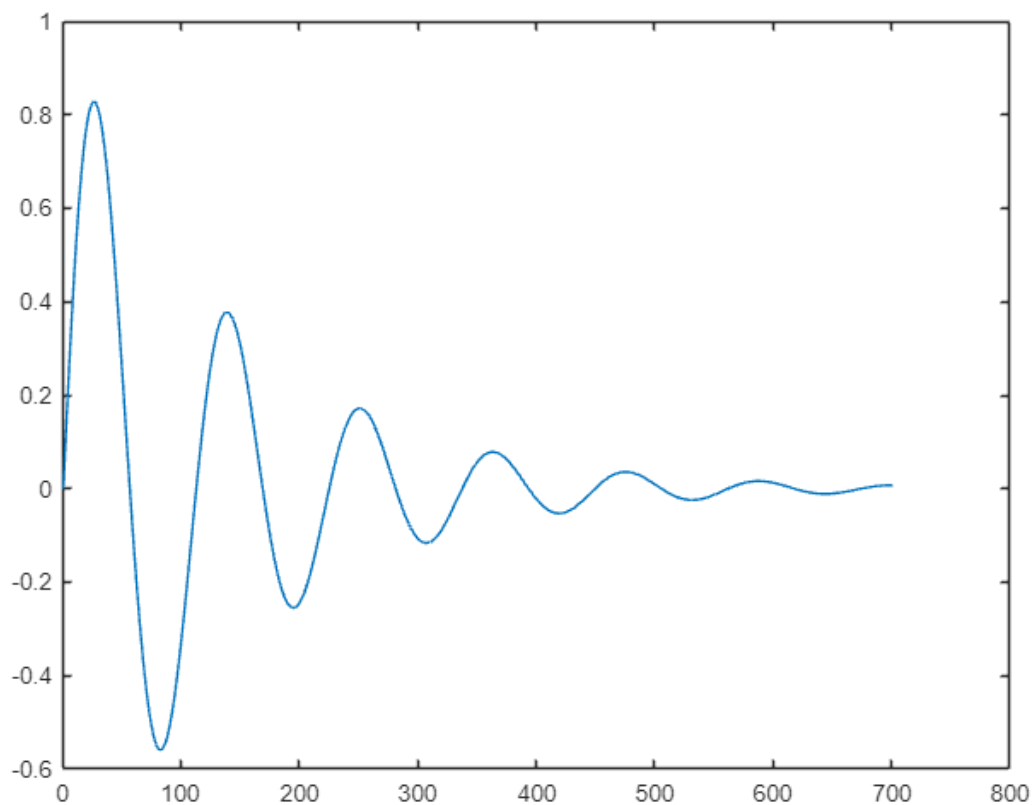


Рисунок 1 – Построение графика с использованием функции plot

Также были построены два графика функций в одних осях (пример 9, рисунок 2).

Пример 9 – Два графика функции в одних осях с помощью hold on и plot

```
ex = [0:0.007:4.9];
ey = exp(-ex).*sin(8*ex);
ez = exp(-ex).*cos(10*ex);
plot(ex, ey)
hold on
plot(ex, ez)
plot(ex, ey, ex, ez)
```

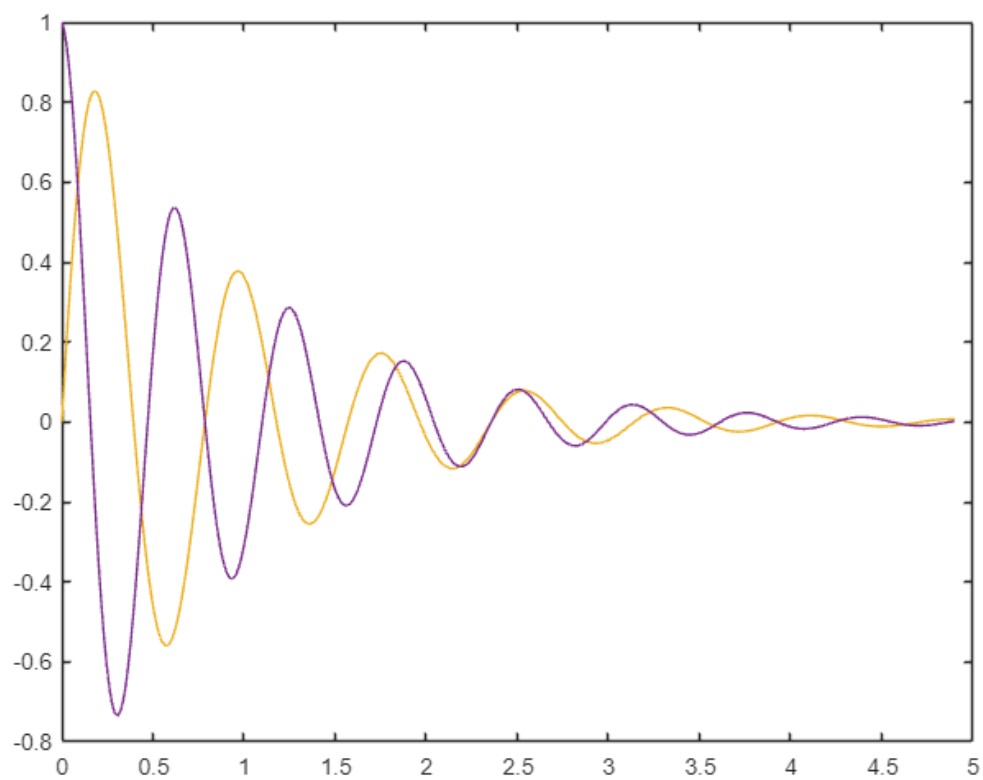


Рисунок 2 – Два графика функции в одних осях

Была разобрана возможность задания цвета и типа линий для графика (пример 10, рисунок 3)

Пример 10 – Задание цвета и типа линии для графика

```
ex = [0:0.007:4.9];
ey = exp(-ex).*sin(8*ex);
ez = exp(-ex).*cos(10*ex);
plot(ex, ey, 'y:x')
```

```
plot(ex, ez, 'm-.x')
```

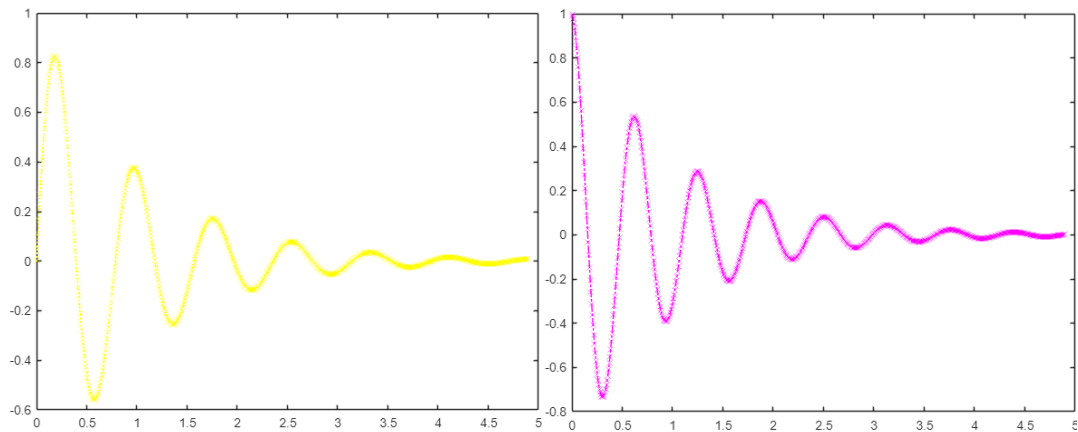


Рисунок 3 – Графики с изменённым цветом и типом линии

Была разобрана возможность использования легенды и подписей осей (пример 11, рисунок 4).

Пример 11 – Использование легенды и подписи осей

```
ex = [0:0.007:4.9];
ey = exp(-ex).*sin(8*ex);
ez = exp(-ex).*cos(10*ex);
plot(ex, ey, 'y:', ex, ez, 'm-.')
legend('ey = exp(-ex).*sin(8*ex);', 'ez = exp(-ex).*cos(10*ex);')
grid on
xlabel('ex')
ylabel('ey')
```

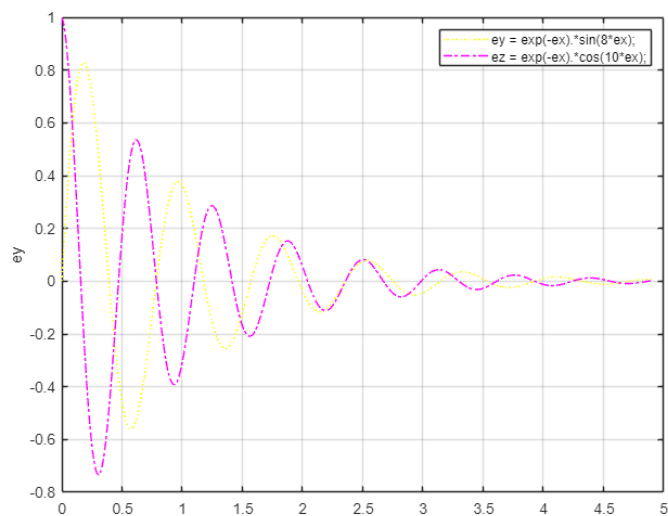


Рисунок 4 – Использование легенды и подписи осей

Также была разобрана возможность использования легенды и подписей осей на нескольких графиках в одном графическом окне (пример 12, рисунок 5).

Пример 12 – Использование легенды и подписи осей

```
t=0:0.002:10*pi;  
k=6
```

```
k =
```

```
6
```

```
x11=(k-1)*(cos(t)+cos((k-1)*t)/(k-1));  
y11=(k-1)*(sin(t)-sin((k-1)*t)/(k-1));  
k=7;  
x21=(k-1)*(cos(t)+cos((k-1)*t)/(k-1));  
y21=(k-1)*(sin(t)-sin((k-1)*t)/(k-1));  
k=8;  
x31=(k-1)*(cos(t)+cos((k-1)*t)/(k-1));  
y31=(k-1)*(sin(t)-sin((k-1)*t)/(k-1));  
k=4.3;  
y41=(k-1)*(sin(t)-sin((k-1)*t)/(k-1));  
x41=(k-1)*(cos(t)+cos((k-1)*t)/(k-1));  
subplot(2, 2, 1)  
plot(x11, y11)  
legend('k=6')  
subplot(2, 2, 2)  
plot(x21, y21)  
legend('k=7')  
subplot(2, 2, 3)  
plot(x31, y31)  
legend('k=8')  
subplot(2, 2, 4)  
plot(x41, y41)  
legend('k=4.3')
```

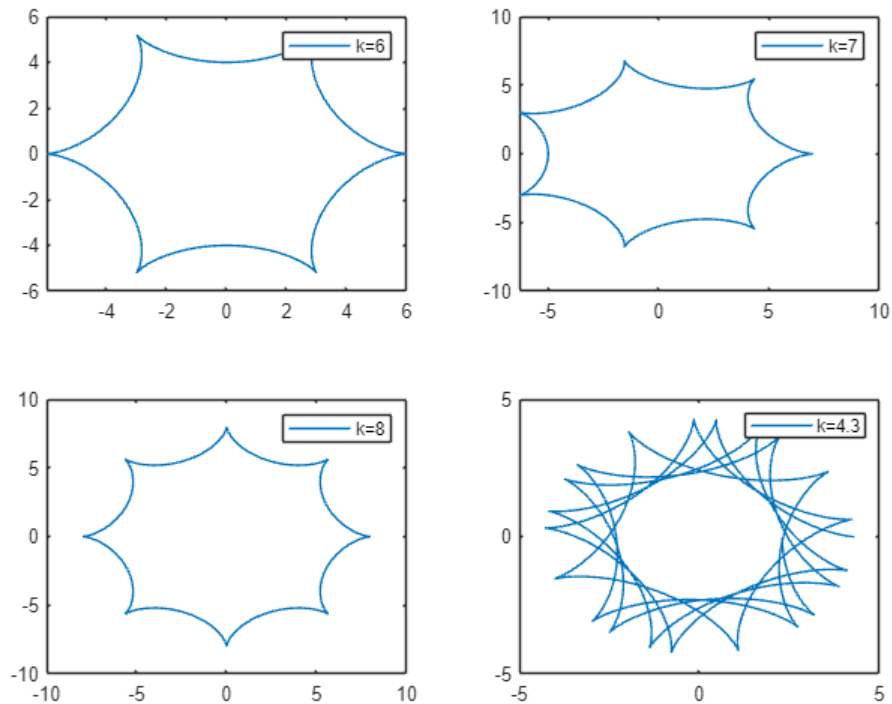


Рисунок 5 – Использование легенды и подписи осей в нескольких графиках

Была создана, а после чего и вызвана процедура (пример 13).

Пример 13 – Создание и вызов процедуры

```
function [x1,x2] = func(a,b,c)
x1=cos(b);
x2=(c+a)/(4*b);
end
[r1, r2]=func(2,3,4)
```

```
r1 =
-0.9900
```

```
r2 =
0.5000
```

Также был построен график созданной нами функции (пример 14, рисунок 6).

Пример 14 – Построение графика функции

```
function y = func(x)
```



```

y(:,1)=345*sin(x)./x;
y(:,2)=x.^3;
end
func(0.3)

```

```
ans =
```

```
339.8482    0.0270
```

```
fplot(@func, [-10,10])
```

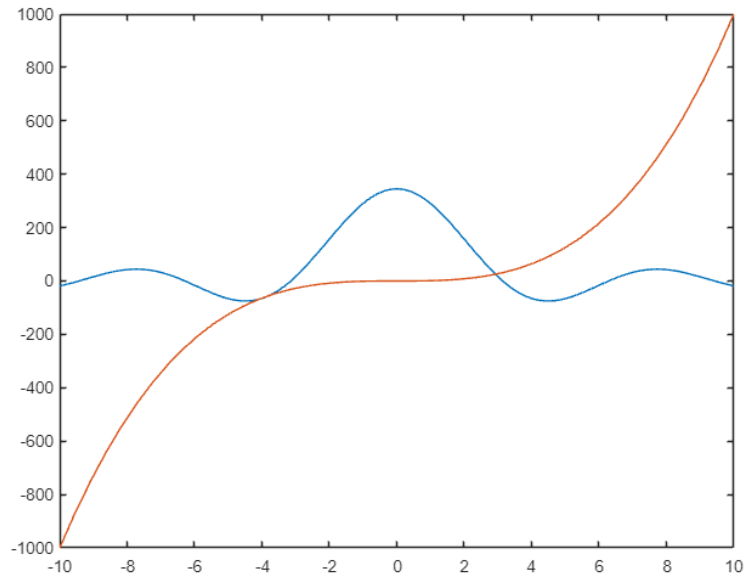


Рисунок 6 – Построенный график функции

Краткое описание изученных библиотек:

- 1) ops – содержит операторы и некоторые специальные символы(plus, minus, ...);
- 2) relop – содержит операторы сравнения(eq, gt, ...);
- 3) slash – содержит операторы деления матриц;
- 4) elfun – содержит элементарные математические функции(sin, cos, ...);
- 5) specfun – содержит специальные математические функции(bessely, beta, ...);
- 6) plot – содержит инструменты для работы с графиками.

Задание по варианту:

Вычислить значения функции $f(x)$ на отрезке $[a; b]$ с шагом h (рисунок 7).

8	$x^3 - 3x + \frac{x^3 - 0,3x}{\sqrt{1+2x}}$	0	4.5	0.5
---	---	---	-----	-----

Рисунок 7 – Задание по варианту

Выполнение задания представлено в листинге 1.

Листинг 1 – Задание по варианту

```
x=[0:0.5:4.5];
y=x.^3-3*x+(x.^3-0.3*x)./sqrt(1+2*x)

y =

    0    -1.3927    -1.5959     0.3375     5.3094    14.1977    27.8649
47.1624    72.9333    106.0143
plot(x, y, 'y:x')
xlabel('x')
ylabel('y')
legend('y=x^3 - 3x + (x^3 - 0.3x)/sqrt(1+2x)')
```

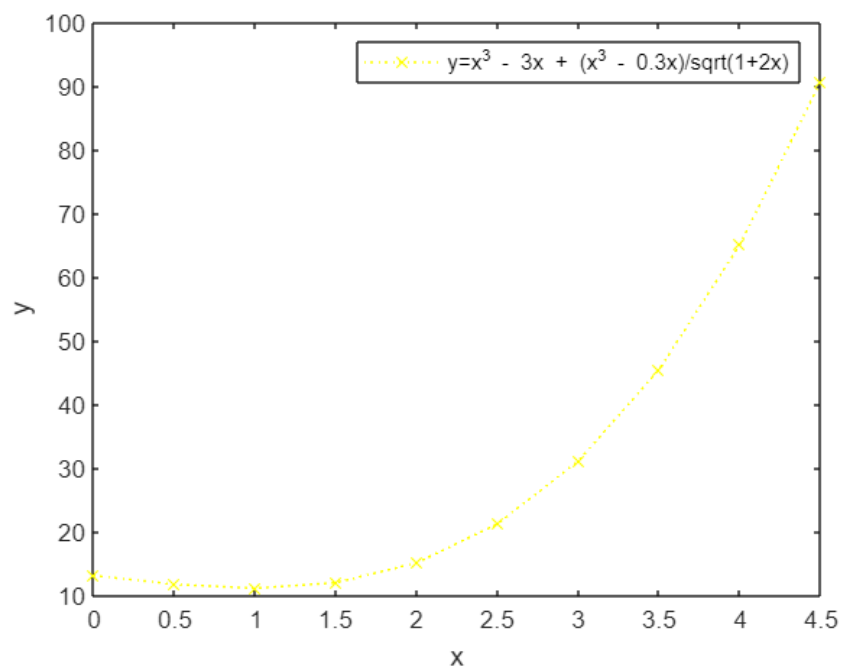


Рисунок 8 – График задания по варианту

Вывод:

В ходе лабораторной работы было получено общее представление о математическом пакете Matlab – особенностях интерфейса, функциональных основных возможностях. Также были сформированы навыки: практической работы в среде Matlab, математических вычислений, моделирования, разработки приложений и анализа данных.

Ответы на контрольные вопросы:

1)MATLAB — язык программирования и пакет прикладных программ для решения задач технических вычислений;

2)С помощью команды help;

3)Для ввода векторов и матриц используются квадратные скобки []. Также есть функции для формирования специальных матриц: linspace, ones, zeros, eye, magic;

4)Для построения графиков используются функции: plot, loglog, semilogx, semilogy, polar;

5)Функциями legend, xlabel, ylabel;

6)Да, с помощью hold on или plot;

7)Разбить окно на несколько подокон с помощью subplot, после чего с помощью этой же функции выбрать окно, в котором будет построен следующий график.