

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
Кафедра приборостроения и информационно-измерительных технологий

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СРЕДСТВ ИЗМЕРЕНИЙ В СРЕДЕ MATLAB

Методические указания к лабораторным работам

Составитель
Н.Ю. МАКАРОВА



Владимир 2011

УДК 519.876.5
ББК 22.18
М 34

Рецензент
Кандидат технических наук, профессор
Владимирского государственного университета
Г. П. Колесник

Печатается по решению редакционного совета
Владимирского государственного университета

М34 Математическое моделирование средств измерений в
среде MATLAB: метод. указания к лаб. работам / Владим.
гос. унт.; сост. Н.Ю. Макарова. – Владимир: Изд-во Владим.
гос. ун-та, 2011. – 82 с.

Приведены методики выполнения лабораторных работ по математическому моделированию в программном комплексе *MATLAB* по дисциплинам «Математическое моделирование средств измерений механических величин», «Математические методы и модели в САПР приборостроения». Изучаются основы программирования в *MATLAB* и *SIMULINK*.

Предназначены для магистрантов 1 и 2-го курса дневного отделения, обучающихся по направлению: 200100 «Приборостроение».

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС 3-го поколения.

Ил. 42. Табл. 6. Библиогр.: 5 назв.

УДК 519.876.5
ББК 22.18

ВВЕДЕНИЕ

Математическое моделирование в настоящее время сопровождает процесс разработки и исследования современных информационно-измерительных систем в учебном процессе и на производстве. Данный курс лабораторных работ посвящен созданию моделей в среде MATLAB и используется в дисциплинах «Математическое моделирование средств измерений механических величин», «Математические методы и модели в САПР приборостроения». Этот курс необходим магистрантам в дальнейшем для последующего решения исследовательских задач в соответствии с квалификационной характеристикой магистра по специальности 200100 «Приборостроение». Целью преподавания дисциплины является ознакомление магистрантов с основными принципами математического моделирования приборов и информационно-измерительных систем, а также с современными программами, предназначенными для инженерного и научного моделирования в приборостроении. Применение моделирования особенно эффективно при исследовании проектируемых систем с целью изучения и прогнозирования различных явлений процессов в этих системах. Приближение к реальным условиям работы проектируемых систем осуществляется при стохастическом моделировании, когда к условиям моделирования добавляются случайные изменения параметров системы, возмущения и шумы измерений физических величин.

В приборостроении актуально моделирование задач управления, получения, передачи и преобразования информации. При этом современные модели везде для описания процессов и систем используют дифференциальные уравнения и линейные матричные преобразования. Развитие современных методов моделирования создало предпосылки для создания и исследования высокоэффективных систем, которые, как правило, ориентированы на цифровые алгоритмы обработки информации с применением современных микропроцессоров, нейрокомпьютеров, процессоров с нечеткой логикой и других современных технологических достижений.

Методические указания предназначены для магистрантов 1-го и 2-го курсов дневного отделения, обучающихся по направлению 200100 «Приборостроение».

Лабораторная работа № 1

ПРОСТЕЙШИЕ ВЫЧИСЛЕНИЯ В MATLAB

Цель работы: ознакомление с программным пакетом MATLAB.

Оборудование: дисплейный класс, программа MATLAB R2008 или выше.

Общие сведения и основы работы в MATLAB

MATLAB (Matrix Laboratory – матричная лаборатория) является мощным средством для решения научно-исследовательских и инженерных задач различной сложности. MATLAB охватывает широкий спектр проблем, связанных с матричным анализом, с обработкой сигналов и изображений, с задачами математической физики, оптимизацией, обработкой и визуализацией данных и многими другими. В состав MATLAB входят дополнительные специализированные пакеты ToolBoxes, которые устанавливаются выборочно. Среди них можно выделить Signal Processing Toolbox, предназначенный для исследования и обработки сигналов (генерация, фильтрация сигналов, спектральный анализ, статистическая обработка сигналов); Image Processing Toolbox, предназначенный для функций цифровой обработки и анализа изображений; Toolbox Optimization, предназначенный для решения основных линейных и нелинейных задач оптимизации (минимизация линейных функций, подбор параметров, минимаксные задачи, задачи о достижении цели); Toolbox Partial Differential Equations (PDE), созданный для решения задач математической физики, описываемых дифференциальными уравнениями и системами в частных производных. В состав MATLAB входит специальное приложение для имитационного моделирования SIMULINK. Визуальная среда GUIDE предназначена для написания приложений с графическим пользовательским интерфейсом. В MATLAB реализованы классические численные алгоритмы решения уравнений, задач линейной алгебры, нахождения значений определенных интегралов, решения дифференциальных уравнений и систем. Простой язык программирования позволяет легко создавать собственные алгоритмы. Кроме того, в MATLAB имеется возможность использовать обширную библиотеку встроенных функций и ToolBoxes. Данное сочетание позволяет достаточно быстро разрабатывать эффективные программы, направленные на решение практически важных задач.

Рабочая среда MATLAB

Рабочая среда MATLAB содержит следующие компоненты (рис. 1):

- меню;
- панель инструментов;
- рабочее окно с активной строкой (Command Window);
- окно с вкладками Launch Pad и Workspace, из которого можно получить доступ к модулям ToolBox и к содержимому рабочей среды;
- окно с вкладками Command History и Current Directory для просмотра и повторного вызова ранее введенных команд и установки текущего каталога.

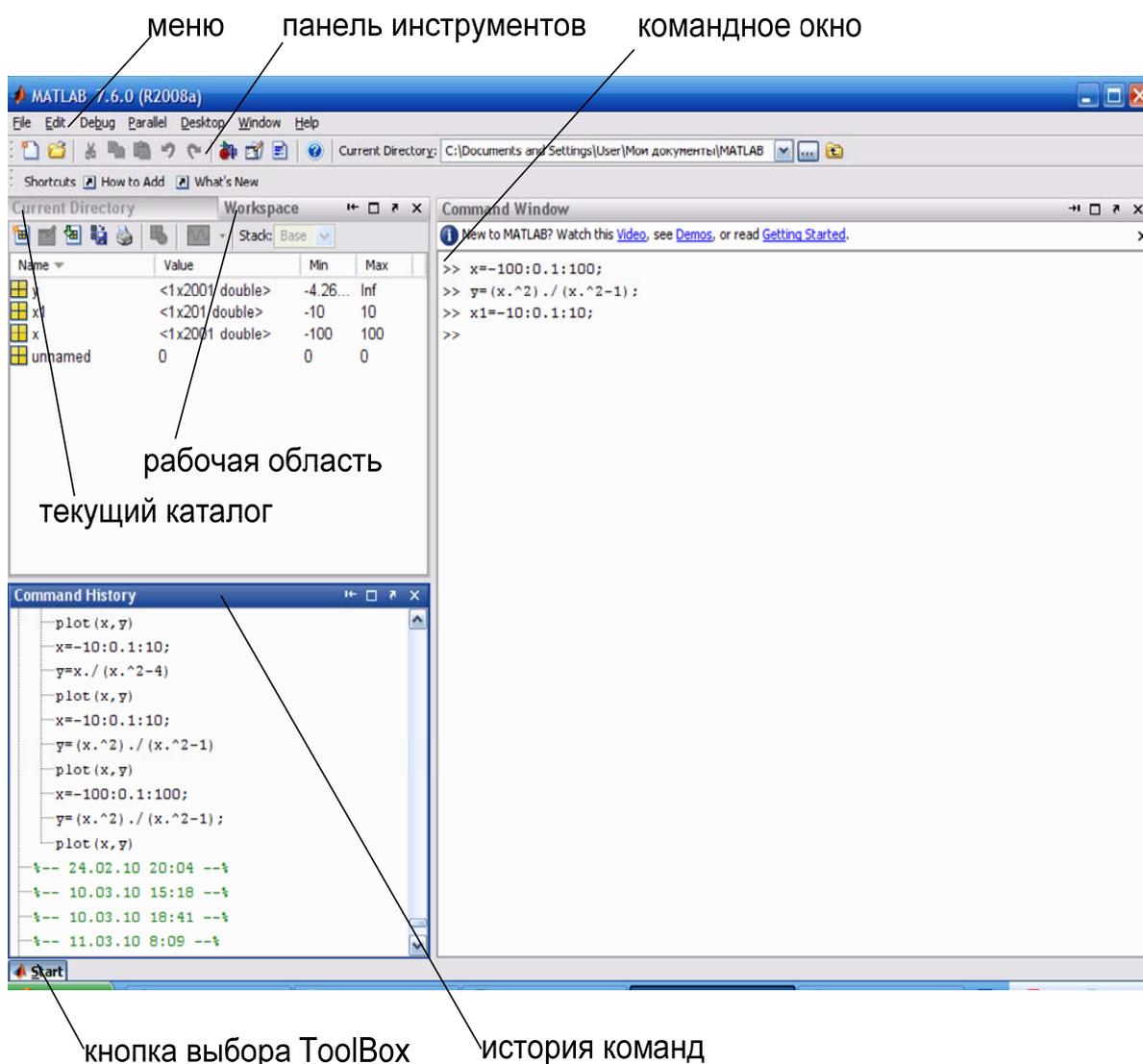


Рис. 1. Рабочая среда MATLAB 7.x (2008)

Формат вывода результата вычислений

Нужный формат вывода устанавливается пользователем из меню MATLAB. Для этого нужно выбрать в меню File вкладку Preferences (свойства), затем вкладку Command Window (рис. 2).

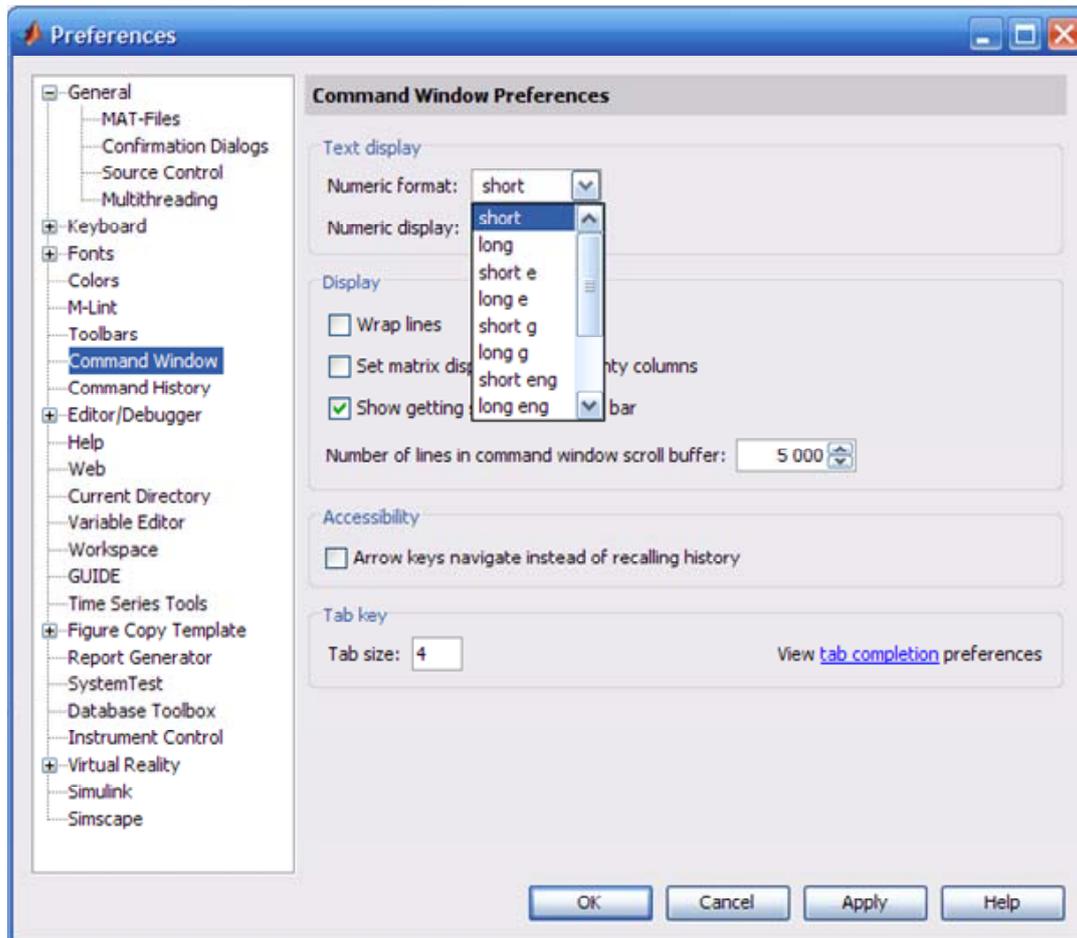


Рис. 2. Диалоговое окно Preferences MATLAB

При установке формата short, результат выводится в виде

```
>> 1/3
```

```
ans =
```

```
0.3333.
```

При установке формата long результат выводится в виде

```
>> 1/3
```

```
ans =
```

```
0.3333333333333333.
```

Результат вычислений может выводиться в экспоненциальной форме при установке формата short e.

```
>> 10000/3
```

ans =

3.3333e+003.

Такая запись эквивалентна $3,3333 \cdot 10^3$.

Простейшие арифметические функции и элементарные вычисления

Встроенные математические функции MATLAB позволяют вычислять различные выражения. Команды MATLAB имеют вид, свойственный всем языкам программирования высокого уровня.

Пример. Наберите в командной строке $5+5 \cdot 8-9/3$ и нажмите Enter. Результат будет представлен в виде

```
>> 5+5*8-9/3
```

ans =

42.

Арифметические операции в MATLAB выполняются в обычном порядке (табл. 1) :

1 - возведение в степень ^;

2 – умножение и деление *, /;

3 – сложение и вычитание +, -.

Для изменения порядка выполнения операторов следует использовать скобки. MATLAB запоминает все вводимые команды. Для повторного их написания можно использовать клавиши <↑>, <↓>.

При вычислении в MATLAB встречаются различные операции. Деление на ноль не приводит к ошибке, в командной строке выдается предупреждение о бесконечном результате:

```
>> 2/0
```

ans =

Inf.

При делении ноля на ноль получается неопределенность – не число NaN (not a number) и также выдается предупреждение:

```
>> 0/0
```

ans =

NaN.

При вычислении корня из отрицательных чисел MATLAB переходит в область комплексных чисел:

```
>> sqrt(-4)
```

ans =

0 + 2.0000e+000i.

Пример. Для того чтобы вычислить значение выражения

$$(\ln 10)^6 \cdot e^{-4} + \sqrt{\frac{\cos 3.55\pi - \sin 8.15\pi}{\operatorname{tg} 6}},$$

в командной строке необходимо набрать

```
>> (log(10))^6*exp(-4)+sqrt((cos(3.55*pi)-sin(8.15*pi))/tan(6))
ans =
    3.7409e+000.
```

Таблица 1. Элементарные функции MATLAB

Обозначение	Пояснение	Обозначение	Пояснение
sin	Синус	cos	Косинус
tan	Тангенс	cot	Котангенс
sec	Секанс	csc	Косеканс
asin	Арксинус	acos	Арккосинус
atan	Арктангенс	acot	Арккотангенс
sinh	Гиперболический синус	cosh	Гиперболический косинус
tanh	Гиперболический тангенс	coth	Гиперболический котангенс
exp	Экспоненциальная функция	log	Натуральный логарифм
log10	Десятичный логарифм	log2	Логарифм по основанию 2
pow2	Возведение числа 2 в степень	sqrt	Квадратный корень
abs, angle	Модуль и фаза комплексного числа	complex	Конструирует комплексное число по его действительной и мнимой части
conj	Возвращает комплексно-сопряженное число	imag, real	Возвращает мнимую и действительную части комплексного числа
fix	Округление до ближайшего целого по направлению к нулю	floor, ceil	Округление до ближайшего целого по направлению к $+\infty$ или $-\infty$
round	Округление до ближайшего целого	mod	Остаток от целочисленного деления со знаком
rem	Остаток от целочисленного деления без знака	sign	Возвращает знак числа

Подробная информация об элементарных функциях и специальных функциях доступна при вызове команд `help elfun` и `help specfun`. Как во всех языках программирования в MATLAB предусмотрена возможность

работы с переменными, причем тип переменных определяется автоматически. В качестве присваивания используется знак =. Строчные и прописные буквы являются двумя разными переменными. Для просмотра информации о переменных используется команда whos.

Графики функций одной переменной

MATLAB обладает обширными возможностями для визуализации графических данных. MATLAB позволяет строить графики функций в линейном, логарифмическом и полулогарифмическом масштабе. Для построения графика функции необходимо:

- 1 – задать вектор значений аргумента x ;
- 2 – вычислить вектор значений функции $f(x)$;
- 3 – вызвать команду `plot(x,y)` для построения графика.

Команды для задания вектора и вычисления функции завершаются точкой с запятой для подавления вывода в командное окно их значений. Для того чтобы задать вектор-строку с указанным шагом, необходимо ввести $x=[\text{начальное значение} : \text{шаг} : \text{конечное значение}]$. В MATLAB все данные представляются в виде массивов, матриц (Matrix Laboratory). Для поэлементной работы с векторами необходимо использовать точку перед операторами $*$, $^$, $/$, \backslash . Пример построения графика функции представлен на рис. 3.

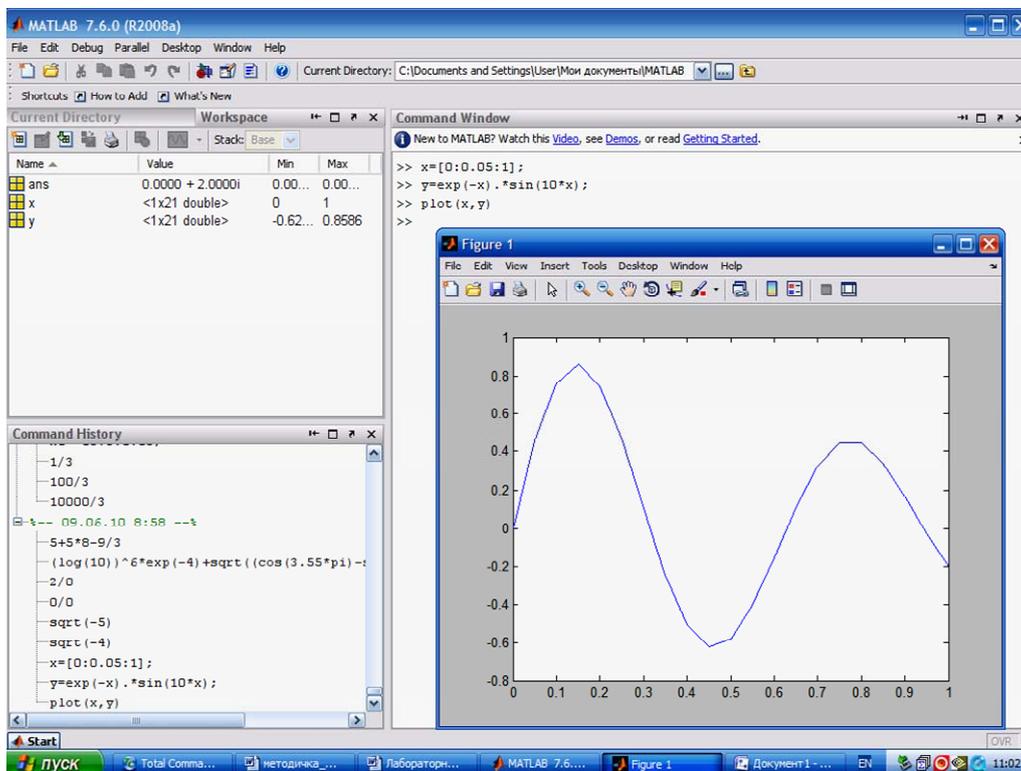


Рис. 3. График функции

Для вывода двух графиков в одних координатных осях (рис. 4) необходимо вызвать команду `plot` с двумя парами аргументов:

```
>> x=[-2*pi:pi/20:2*pi];  
>> f=exp(-0.1*x).*sin(x).^2;  
>> g=exp(-0.2*x).*sin(x).^2;  
>> plot(x,f,x,g).
```

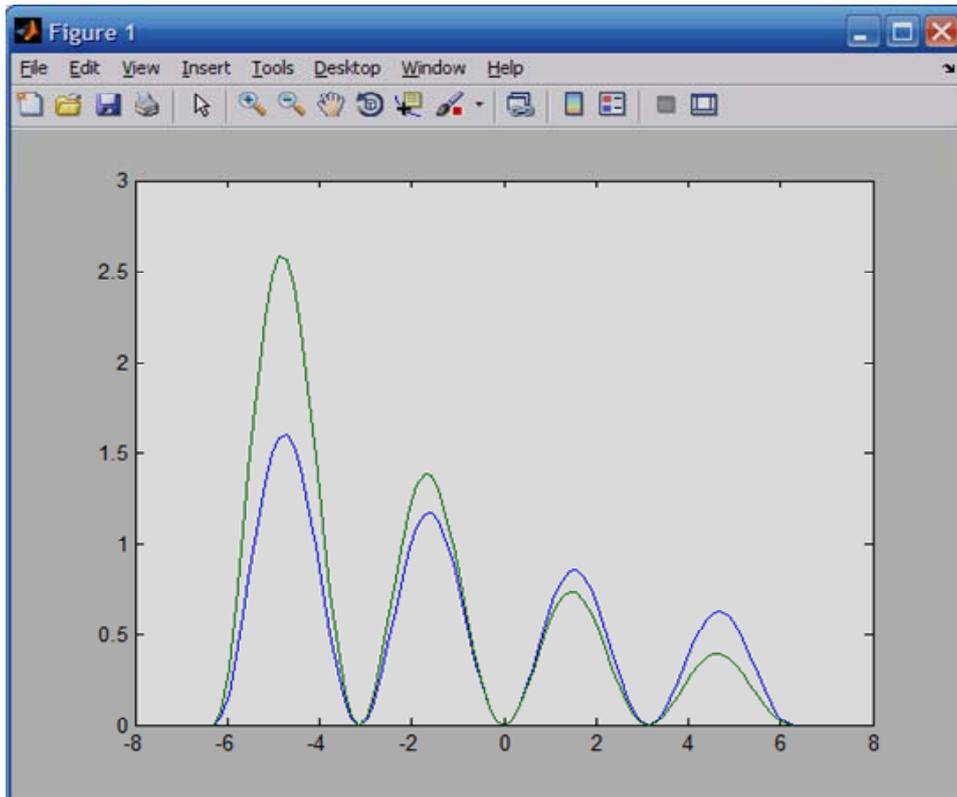


Рис. 4. Графики двух функций

Функции могут иметь разные области изменения входного аргумента, но при этом задаются аналогичным образом. Также можно построить графики с различными осями ординат с помощью функции `plotyy`. Для построения графиков в логарифмических масштабах используются функции:

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

MATLAB позволяет задавать цвет линии графика, тип линии и тип маркера, устанавливая дополнительные аргументы в функции `plot`. Эти аргументы помещаются в апострофах и следуют за парой векторов (табл.2).

Пример изменения свойств линий приведен на рис. 5.

```
plot(x,f, 'ro-',x,g,'kp:')
```

Таблица 2. Свойства линии

Цвет		Тип маркера		Тип линии	
y	Желтый	.	Точка	-	Сплошная
m	Розовый	o	Кружок	:	Пунктирная
c	Голубой	x	Крестик		
r	Красный	+	Плюс	.	Штрих-пунктирная
g	Зеленый	*	Звездочка	--	Штриховая
b	Синий	s	Квадрат		
w	Белый	d	Ромб		
k	Черный	v	Треугольник		

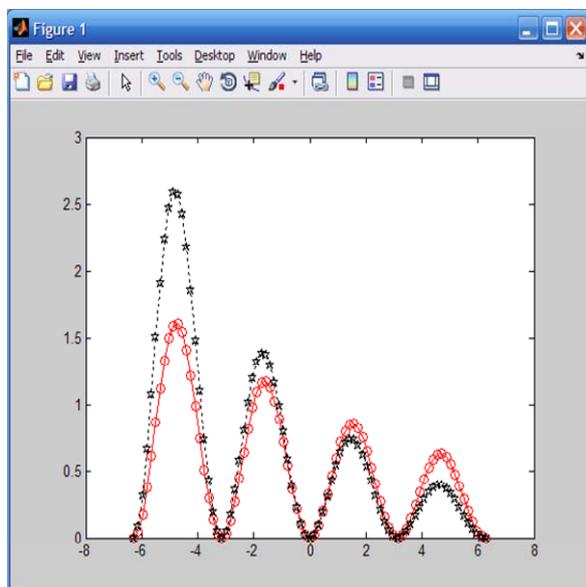


Рис. 5. Изменение свойства линий

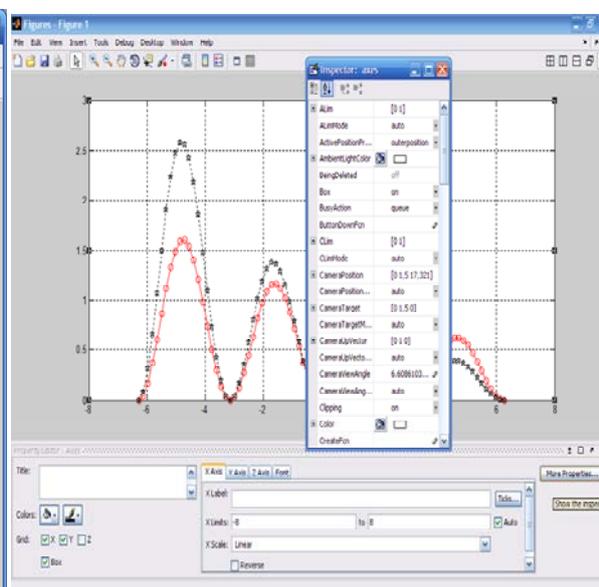


Рис. 6. Редактирование свойств графика

Изменить свойства линии, вывести на график обозначения осей, легенду, включить сетку можно с помощью свойств графика (рис. 6). Для этого необходимо нажать на пиктограмму Edit Plot и два раза щелкнуть левой кнопкой мыши на поле графика.

Графики параметрических и кусочно-заданных функций

Для построения параметрических функций необходимо задать вектор значений аргумента, вычислить значения функции и построить график.

Пример. График эллипса, заданного выражениями

$$x(t)=0.5\sin(t), y(t)=0.7\cos(t) \text{ для } t [0, 2\pi].$$

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

```
>> x=0.5*sin(t);
>> y=0.7*cos(t);
>> plot(x,y)
```

Для построения кусочно-заданной функции необходимо вычислить каждый участок, полученные пары массивов объединить в векторы абсцисс и ординат и построить график (рис. 7).

Пример.

$$y(x) = \begin{cases} \pi \cdot \sin(x), & -2\pi \leq x \leq -\pi; \\ \pi - |x|, & -\pi < x < \pi; \\ -\pi \cdot \sin^3 x, & \pi \leq x \leq 2\pi. \end{cases}$$

```
>> x1=[-2*pi:0.01:-pi];
>> y1=pi*sin(x1);
>> x2=[-pi:0.01:pi];
>> y2=pi-abs(x2);
>> x3=[pi:0.01:2*pi];
>> y3=-pi*sin(x1).^3;
>> x=[x1 x2 x3];
>> y=[y1 y2 y3];
>> plot(x,y)
или
>> plot(x1,y1,x2,y2,x3,y3).
```

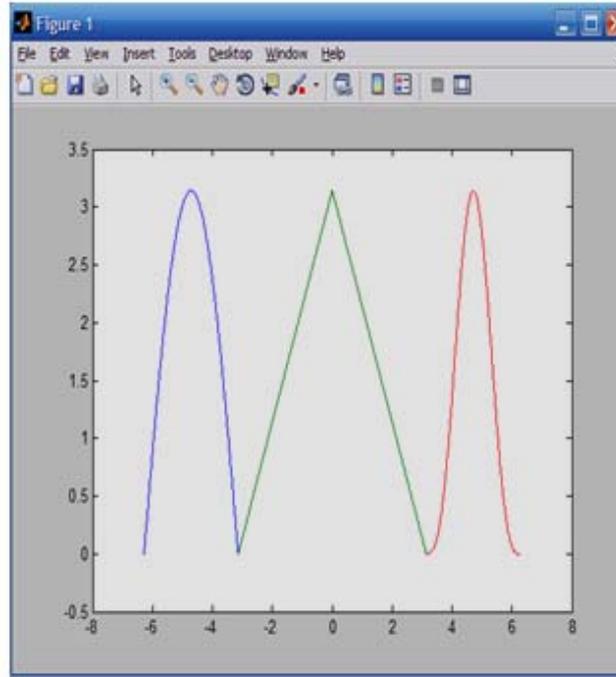


Рис. 7. График кусочно-линейной функции

Графики функций двух переменных

Для построения трехмерных графиков в MATLAB используются различные функции. Для визуализации функций двух переменных необходимо

1. Сгенерировать матрицы с координатами узлов сетки на прямоугольной области определения функции.
2. Вычислить функцию в узлах сетки.
3. Выбрать графическую функцию.

Сетка генерируется при помощи команды `meshgrid`.

Пример. Построить график функции

$$Z = 3(1 - X)^2 e^{-X^2 - (Y+1)^2} - 10\left(\frac{X}{5} - X^3 - Y^5\right) e^{-X^2 - Y^2} - 1/3 \cdot e^{-(X+1)^2 - Y^2},$$

заданной на прямоугольной области определения $X \in [-2, 2]$ и $Y \in [-2, 2]$.

```

>> [X,Y]=meshgrid(-2:0.05:2,-2:0.05:2);
>> Z = 3*(1-X).^2.*exp(-(X.^2) - (Y+1).^2) ...
    - 10*(X/5 - X.^3 - Y.^5).*exp(-X.^2-Y.^2) ...
    - 1/3*exp(-(X+1).^2 - Y.^2);
>> mesh (X,Y,Z)

```

Для построения каркасной поверхности используется функция mesh. Функция surf строит каркасную поверхность графика функции и заливает каждую клетку поверхности определенным цветом, зависящим от значения функции в точках. Команда shading interp позволяет убрать каркасные линии на графике. Функция contour3 строит поверхность, состоящую из линий уровня.

В MATLAB можно изменять цветовую палитру с помощью команды colormap (...) (табл. 3).

Таблица 3. Палитры цвета

Палитра	Цвет
autumn	Плавное изменение: красный-оранжевый-желтый
bone	Легкий оттенок синего
cool	Оттенки голубого и пурпурного цветов
gray	Оттенки серого
hsv	Цвета радуги
spring	Оттенки пурпурного и желтого

Несколько графиков в одном окне

MATLAB позволяет разбить графическое окно на несколько подграфиков со своими осями. Для этого используется команда subplot (i, j, n), где i – число графиков по вертикали (рис. 8), j – число графиков по горизонтали, n - текущий график.

Пример вывода графиков в нескольких окнах.

```

>> [x,y]=meshgrid(-2:0.5:2,-2:0.5:2);
>> z = 3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...
    - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) - 1/3*exp(-(x+1).^2 - y.^2);
>> subplot(3,2,1)
>> mesh(x,y,z)
>> title ('mesh')
>> subplot(3,2,2)
>> surf(x,y,z)
>> title ('surf')
>> subplot(3,2,3)
>> meshc(x,y,z)

```

```

>> title ('meshc')
>> subplot(3,2,4)
>> surfc(x,y,z)
>> title ('surfc')
>> subplot(3,2,5)
>> contour3(x,y,z)
>> title ('contour3')
>> subplot(3,2,6)
>> surf(x,y,z)
>> shading interp
>> title ('surf!')

```

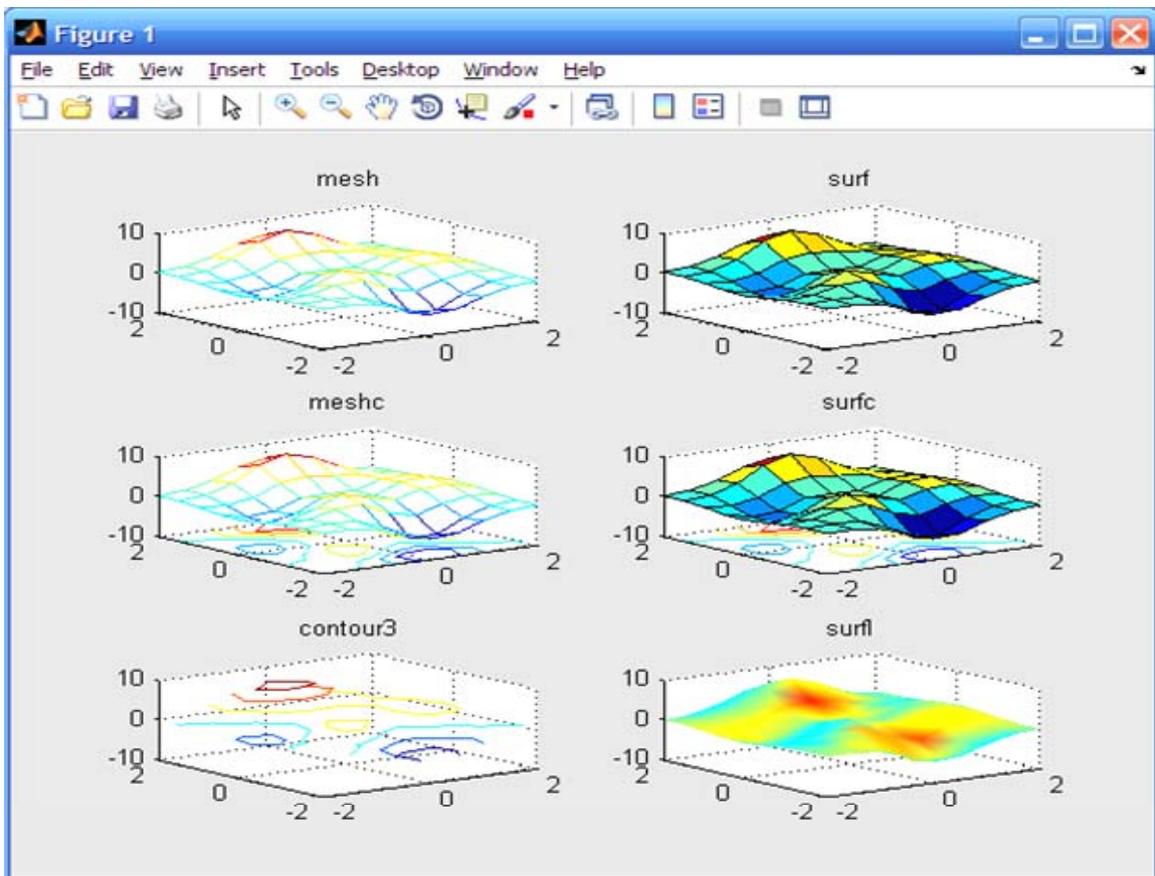


Рис. 8. Использование подграфиков

Задания к лабораторной работе №1

Задание 1

- Задать изменение аргумента.
- Вычислить значения функций 1 и 2 для аргумента в заданном интервале.

- Вывести графики функций одновременно на одном графике в декартовых координатах. Для разных графиков использовать разный тип линий.
- Вывести графики функций в двух подокнах на одном графике.

Варианты заданий:

№	Функция 1	Функция 2	a	b	h
1	$y = \sin(x)$	$z = \exp(x+3)/5000 - 1$	-2π	2π	$\pi/20$
2	$y = \cos(x)$	$z = 0.00025e^3 - x - 0.6$	-2π	2π	$\pi/20$
3	$y = \operatorname{tg}(x) + 0.1$	$z = (1+x)^6$	-2π	2π	$\pi/20$
4	$y = (x^2-1)/15$	$z = 1+\sin(x)$	-2π	2π	$\pi/20$
5	$y = (x^3-2)/15$	$z = 5\cos(x)$	-2π	2π	$\pi/20$
6	$y = x^2 - 10$	$z = 0.025\exp(-1.2x)$	-5	5	1
7	$y = 3\sin(x)$	$z=0.015x^3$	-5	5	1
8	$y = 4\sin(x)$	$z = 0.05x^2$	1	10	1
9	$y = 6\sin(x)$	$z = 0.01x^3$	-10	10	1
10	$y = 2+\cos(x)$	$z = -0.05(x^2 + 10\cos(x))$	-8	8	1
11	$y = \sin^2(x/3)$	$z = 0.01(x^2 - 40\sin(x))$	-8	8	1
12	$y = \cos^3(x)$	$z = \sin(x) + \sin(2x)$	$-\pi$	π	$\pi/8$
13	$y = 0.5x + \cos^2(x)$	$z = \sin^2(x) + \cos(x)$	$-\pi$	π	$\pi/8$
14	$y = \sin(x) + \cos^2(2x)$	$z = x(0.5 + x)\exp(0.1x)$	$-\pi$	π	$\pi/8$
15	$y = \sin(x) \exp(x/2)$	$z = 5x - x^{1.5} + \sin(x)$	0	5	0.5
16	$y = \sin(x) + \cos(x)$	$Y = 1 - 2\cos(x)$	$-\pi$	π	$\pi/8$
17	$Y = \cos^2(x)$	$Y = x\sin(x)$	-2π	2π	$\pi/20$
18	$Y = (x-2)/(x+2)$	$Y = \operatorname{cosec}(x)$	-5	5	1
19	$Y = 1/2(x+ x)$	$Y = x^2 + 1/x$	-8	8	1
20	$Y = 10^{1/x}$	$Y = 10/(x^2+1)$	-10	10	1

Задание 2

Построить графики функций в полярной системе координат (1-10) и функций, заданных параметрически (11-20)

№	Функция	№	Функция
1	$r = \varphi/2$	11	$X = t^3, y = t^2$
2	$r = e^\varphi$	12	$X = 10\cos(t), y = \sin(t)$
3	$r = \pi/\varphi$	13	$X = 10\cos^3 t, y = 10\sin^3 t$
4	$r = 2\cos(\varphi)$	14	$X = a(\cos t + t \sin t), y = a(\sin t - t \cos t)$
5	$r = 1/\sin(\varphi)$	15	$X = at/(1+t^3), y = at^2/(1+t^3)$
6	$r = \sec^2 \varphi/2$	16	$X = 2^t + 2^{-t}, y = 2^t - 2^{-t}$
7	$r = 10\sin(3\varphi)$	17	$X = 2\cos^2 t, y = 2\sin^2 t$
8	$r = 1/\sin(\varphi)$	18	$X = t - t^2, y = t^2 - t^3$
9	$r = a(1 + \cos(\varphi))$	19	$X = a(2\cos t - \cos(2t)), y = a(2\sin t - \sin(2t))$
10	$r^2 = a^2 \cos(2\varphi)$	20	$X = a/\sqrt{1+t^2}, y = at/(\sqrt{1+t^2})$

Задание 3. Трехмерная функция и объемные графики в своих окнах.

- Ввести исходные данные.
 - Вычислить функцию.
 - Вывести функцию в виде 5 трехмерных графиков разного типа.
 - Вывести функцию в виде 2 контурных графиков разного типа.
- Поверхностный и контурный графики. Для формирования поверхностного или контурного графика необходимо:
- задать сетку по координатам X и Y,
 - вычислить функцию $Z=f(X,Y)$,
 - ввести номер графического окна, вывести туда график выбранного типа:
 - трехмерный с аксонометрией, функция $\text{plot3}(X,Y,Z)$,
 - трехмерный с функциональной окраской, функция $\text{mesh}(X,Y,Z)$,
 - трехмерный с функциональной окраской и проекцией, функция $\text{meshc}(X,Y,Z)$,
 - трехмерный с функциональной окраской и проекцией, функция $\text{surf}(X,Y,Z)$,
 - контурный, функция $\text{contour}(X,Y,Z)$,
 - объемный контурный, функция $\text{contour3}(X,Y,Z)$,
 - трехмерный с освещением, функция $\text{surf1}(X,Y,Z)$.

Варианты заданий:

№	Функция	Пределы изменения	
		x	y
1	$z=\sin(x)\cos(y)$	от -2π до 2π	от -2π до 2π
2	$z=\sin(x/2)\cos(y)$	от -2π до 2π	от -2π до 2π
3	$z=\sin(2x)\cos(y)$	от -2π до 2π	от -2π до 2π
4	$z = \sin(x)\cos(y/2)$	от -2π до 2π	от -2π до 2π
5	$z = \sin(x/2)\cos(2y)$	от -2π до 2π	от -2π до 2π
6	$z = \sin(2x)\cos(2y)$	от -2π до 2π	от -2π до 2π
7	$z = (1+\sin(x)/x)(\sin(y)/y)$	от -2π до 2π	от -2π до 2π
8	$z = (\sin(x)/x)\cos(y)$	от -2π до 2π	от -2π до 2π
9	$z = (\sin(x)/x) \cos(y) $	от -2π до 2π	от -2π до 2π
10	$z = (\sin(x)/x)y$	от -2π до 2π	от -2π до 2π
11	$z = (\sin(x)/x) y $	от -2π до 2π	от -2π до 2π
12	$z = (\sin(x)/x)\sin(y)$	от -2π до 2π	от -2π до 2π

13	$z = (\sin(x)/x) \sin(y) $	от -2π до 2π	от -2π до 2π
14	$z = (\sin(x)/x)(1-y)$	от -2π до 2π	от -2π до 2π
15	$z = (\sin(x)/x) y+0.5 $	от -2π до 2π	от -2π до 2π
16	$Z = \sin(x)/x * \cos(y)/y$	от -2π до 2π	от -2π до 2π
17	$z = \sin(2x)\cos(4y)/y$	от -2π до 2π	от -2π до 2π
18	$z = (\cos(x)/x)(1-y)$	от -2π до 2π	от -2π до 2π
19	$z = (1+\sin(x)/x)(1+\cos(y)/y)$	от -2π до 2π	от -2π до 2π
20	$z = (\cos(x)/x) y+7 $	от -2π до 2π	от -2π до 2π

Содержание отчета

Отчет оформляется каждым студентом самостоятельно. Защита происходит в начале каждого следующего занятия с демонстрацией работы программы на ЭВМ. Студент, не подготовивший или не защитивший отчет по работе, к следующей лабораторной работе не допускается.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Вариант задания.
4. Текст программы в MATLAB и полученные графики.
5. Выводы по работе.

Контрольные вопросы и задания

1. Приведите примеры научно-исследовательских и инженерных задач, которые могут быть решены в программе MATLAB?
2. Назовите и поясните основные компоненты рабочей среды MATLAB?
3. Каким образом можно изменить формат данных в MATLAB?
4. Как можно изменить свойства линии на графике в MATLAB?
5. Какие функции применяются для визуализации функций одного и двух аргументов в MATLAB?
6. Как организовать вывод нескольких графиков в одном окне?

Лабораторная работа № 2 РАБОТА С МАССИВАМИ В MATLAB

Цель работы: научиться выполнять вычисления с векторами и массивами в MATLAB.

Оборудование: дисплейный класс, программа MATLAB R2008 или выше.

Работа с векторами

Представление данных в виде матриц используется при обработке какого-то массива экспериментальных данных, полученных в ходе эксперимента, также при исследовании и анализе различных преобразовательных процессов для представления вектора входных воздействий и получении вектора выходных величин.

Элементы вектора-столбца отделяются точкой с запятой, а элементы строки отделяются пробелом или запятой:

```
>> a=[5 6 7]
```

```
a =
```

```
    5    6    7
```

```
>> b=[1; 2; 3]
```

```
b =
```

```
    1
```

```
    2
```

```
    3
```

Транспонирование векторов осуществляется с помощью апострофа.

```
>> a'
```

```
a =
```

```
    5
```

```
    6
```

```
    7
```

Для определения длины векторов используется функция `length`. Для определения размерности массивов используются функции `ndims` и `size`.

В MATLAB операции сложения, вычитания и элементарные функции выполняются с массивами поэлементно.

```
>> sin(a)
```

```
ans =
```

```
-9.5892e-001
```

```
-2.7942e-001
```

```
 6.5699e-001
```

```
>> a+b
ans =
    6
    8
   10
```

При умножении векторов a и b без точки перед знаком умножения MATLAB выдаст сообщение об ошибке, так как при произведении матрицы на матрицу необходимо согласование размерностей.

```
>> a*b
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

При поэлементных операциях необходимо ставить точку перед знаками $*$, $^$, $/$, \backslash .

```
>> a.*b
ans =
    5
   12
   21
```

Перемножение элементов вектора осуществляется функцией `prod`, суммирование элементов вектора – функцией `sum`. Минимум и максимум находятся с помощью функций `max` и `min`. Упорядочение вектора по возрастанию элементов осуществляется функцией `sort(a)`, по убыванию – `sort(-a)`.

Для вычисления скалярного произведения нужно использовать функцию `sum` и поэлементное умножение. Для векторного произведения применяется функция `cross`.

Двумерные массивы, матрицы

При задании матриц элементы в строках вводятся через пробел или запятую, а строки друг от друга отделяются точкой с запятой.

```
Пример.  $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ 
>> A=[1 2 3; 4 5 6; 7 8 9]
A =
    1    2    3
    4    5    6
    7    8    9
```

Обращение к элементам матрицы осуществляется с помощью двух индексов – номеров строки и столбца в круглых скобках.

```
>> A(2,3)
ans =
    6
```

При сложении или вычитании матрицы должны быть одного размера, иначе MATLAB выдаст сообщение об ошибке. Транспонирование матрицы производится так же, как транспонирование вектора, при помощи символа «'».

Для решения систем линейных уравнений используется символ \.

$$\text{Пример. } \begin{cases} 1.2x_1 + 0.3x_2 - 0.2x_3 = 1.3 \\ 0.5x_1 + 2.1x_2 + 1.3x_3 = 3.9 \\ -0.9x_1 + 0.7x_2 + 5.6x_3 = 5.4 \end{cases}$$

При решении указанной системы необходимо ввести матрицу коэффициентов перед x-A, матрицу правой части-B.

```
>> A=[1.2 0.3 -0.2; 0.5 2.1 1.3; -0.9 0.7 5.6]
```

```
A =
    1.2000    0.3000   -0.2000
    0.5000    2.1000    1.3000
   -0.9000    0.7000    5.6000
```

```
>> b=[1.3; 3.9; 5.4]
```

```
b =
    1.3000
    3.9000
    5.4000
```

```
>> x=A\b
```

```
x =
    1.0000
    1.0000
    1.0000
```

Проверка b=A*x.

```
>> A*x
ans =
    1.3000
    3.9000
    5.4000
```

MATLAB позволяет конструировать блочные матрицы, но при этом соответствующие размеры блоков должны совпадать. Для выделения блоков матрицы осуществляется индексация при помощи двоеточия.

```
>> A=[1.2 0.3 -0.2; 0.5 2.1 1.3; -0.9 0.7 5.6]
```

```
A =
```

```
1.2000 3.0000 -2.0000
5.0000 2.1000 1.3000
-9.0000 7.0000 5.6000
```

```
>> A1=A(2:3, 2:3)
```

```
A1 =
2.1000 1.3000
7.0000 5.6000
```

Для удаления строки или столбца необходимо использовать пустые парные квадратные скобки []. Удалим первый столбец матрицы A.

```
>> A(:,1)=[]
```

```
A =
3.0000-2.0000
2.1000 1.3000
7.0000 5.6000
```

С помощью индексации можно заполнять матрицы нужными элементами.

```
>> A(:,1)=1
```

```
A =
1.0000-2.0000
1.0000 1.3000
1.0000 5.6000
```

В MATLAB имеется большая коллекция матриц специального вида.

Матрица, состоящая из нулей, вызывается функцией `zeros`, в которой задается количество строк и столбцов.

```
>> Z=zeros(4,3)
```

```
Z =
0 0 0
0 0 0
0 0 0
0 0 0
```

Если нужна квадратная матрица, то достаточно указывать один аргумент.

```
>> Z=zeros(3)
```

```
Z =
0 0 0
0 0 0
0 0 0
```

Единичная матрица содержит единицы на главной диагонали и вызывается командой `eye`.

```
>> E=eye(3,6)
```

```
E =  
 1  0  0  0  0  0  
 0  1  0  0  0  0  
 0  0  1  0  0  0
```

Матрица из единиц задается функцией `ones`.

```
>> T=ones(3,6)  
T =  
 1  1  1  1  1  1  
 1  1  1  1  1  1  
 1  1  1  1  1  1
```

Матрица из чисел, расположенных случайным образом между 0 и 1, задается функцией `rand`.

```
>> R=rand(3)  
R =  
 0.9649  0.9572  0.1419  
 0.1576  0.4854  0.4218  
 0.9706  0.8003  0.9157
```

Матрица из случайных нормально распределенных чисел задается функцией `randn`.

```
>> R=randn(3)  
R =  
 -0.4326  0.2877  1.1892  
 -1.6656 -1.1465 -0.0376  
  0.1253  1.1909  0.3273
```

Диагональная матрица с необходимыми элементами по диагонали и остальными нулевыми элементами формируется функцией `diag`.

```
>> d=[1 2 3 4];  
>> D=diag(d)  
D =  
 1  0  0  0  
 0  2  0  0  
 0  0  3  0  
 0  0  0  4
```

Для заполнения не главной, а побочной диагонали используется второй аргумент в функции `diag(d,2)`.

Магическая матрица (магический квадрат), у которой суммы элементов в строках и столбцах равны, вызывается функцией `magic`.

```
>> M=magic(3)  
M =  
 8  1  6
```

3 5 7
4 9 2

В MATLAB заложены функции по матричному анализу.

Функцией `det` можно вычислить определитель матрицы. Функция `rank` вычисляет ранг матрицы. Функция `norm` выдает векторные и матричные формы.

Как их правильно использовать и какие задавать аргументы, можно узнать из справки.

```
>> help det
```

```
DET Determinant.
```

```
DET(X) is the determinant of the square matrix X.
```

```
Use COND instead of DET to test for matrix singularity.
```

Задания к лабораторной работе № 2

Задание 1. Задать матрицу A с помощью операции конкатенации:

A1=3,25 1,07 2,34

A2=10,10 0,25 4,78

A3=5,04 7,79 3,31

Причем, для четных номеров варианта A1, A2, A3 – строки матрицы, для нечетных – столбцы.

Задание 2. Сгенерируйте массив B размером 3x3 со случайными элементами, равномерно распределенными на интервале от 0 до 10.

Задание 3. Выполните действия:

1) $A+10B$

2) $A*B$

3) B^T

4) почленно умножить A на B,

5) расположить элементы матрицы A по возрастанию (по столбцам),

6) определить максимальный и минимальный элементы матрицы B,

7) вычислить определитель матрицы B,

8) $A \cdot B - B \cdot A$

9) создать случайную матрицу 4x4 M и вычислить

$(2 \cdot M^3 - 3 \cdot (M^{-1})^2 + E) \cdot M^T$, где E – единичная матрица

Задание 4. Задать массив C из трех элементов.

Задание 5. Решить систему алгебраических линейных уравнений и осуществить проверку.

$A \cdot X = C$.

Задание 6. Определить массив D:

$$D = |\sin(A) + B^{3/5}|.$$

Задание 7. Для двух векторов:

$$a = \{a_1 \ a_2 \ a_3\} \quad \text{и} \quad b = \{b_1 \ b_2 \ b_3\}$$

определите векторное произведение векторов, скалярное произведение векторов, модули векторов и угол между ними.

Задание 8. Сформировать магическую матрицу размером 10X10 и вычислить её ранг. Отсортировать элементы столбца в порядке возрастания.

Содержание отчета

Отчет оформляется каждым студентом самостоятельно. Защита происходит в начале каждого следующего занятия с демонстрацией работы программы на ЭВМ. Студент, не подготовивший или не защитивший отчет по работе, к следующей лабораторной работе не допускается.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Вариант задания.
4. Текст программы в MATLAB и полученные графики.
5. Выводы по работе.

Контрольные вопросы и задания

1. Как формируются в MATLAB массивы и матрицы?
2. Как организуются поэлементные операции с массивами в MATLAB?
3. С помощью каких операций осуществляется решение линейных уравнений?
4. Расскажите о матрицах специального вида в MATLAB.
5. Какие функции матричного анализа являются библиотечными в MATLAB?

Лабораторная работа № 3

ОСНОВЫ ПРОГРАММИРОВАНИЯ В MATLAB

Цель работы: научиться создавать файлы-функции и файлы-программы с циклами и ветвлениями в MATLAB.

Оборудование: дисплейный класс, программа MATLAB R2008 или выше.

М-файлы

При реализации алгоритма функции преобразования какого-либо измерительного прибора или исследования влияния внешнего или внутреннего параметра на результаты измерения необходимо написать программу в MATLAB.

Для написания текста программ в MATLAB используется редактор М-файлов (рис. 9). При помощи редактора можно создавать собственные функции, вызывать их, в том числе и из командной строки. Для вызова редактора М-файлов нужно выбрать в меню File основного меню MATLAB вкладку New, а в ней M-file.

Файлы программ сохраняются в MATLAB с расширением *.m. Комментарии в тексте программы начинаются со знака %.

В MATLAB М-файлы делятся на два вида: файл-функции (functions) и файл-сценарии (scripts). Сценарии содержат определенную последовательность команд, не имеют входных и выходных аргументов. Функции напротив производят определенные действия с входными аргументами и возвращают результат в выходных аргументах.

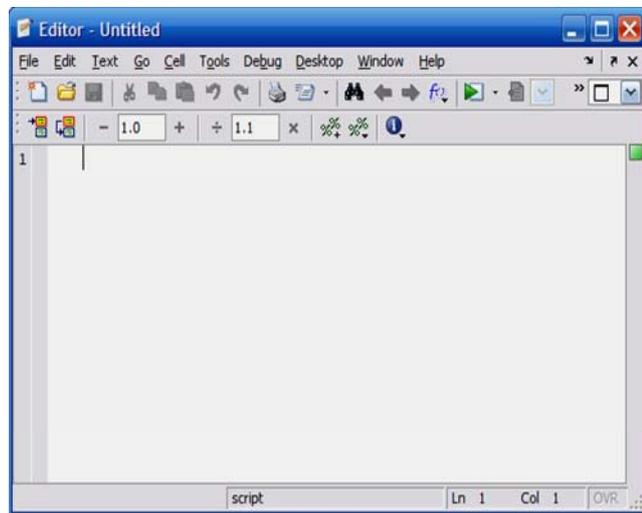


Рис. 9. Редактор М-файлов MATLAB R2008

Слово function определяет, что данный файл содержит функцию. Первая строка является заголовком функции, в которой размещается имя функции и список входных и выходных аргументов. Далее следует тело функции

```
function f=myfunction(x)
f=log(x)/sin(x)
```

В рассмотренной функции с именем myfunction входным аргументом является x, а результат вычислений – это выходной аргумент f. Файл-функцию следует сохранять под тем же именем, которое указано в первой строке файла. Созданную функцию можно вызвать в командной строке или в другой функции.

```
>>myfunction(2)
```

```
f =  
    0.7623
```

Входных и выходных аргументов может быть несколько
function [X, Y, Z] = name (x, y, z)

Операторы цикла

Цикл for

Этот цикл предназначен для выполнения определенного числа повторяющихся действий. Его структура выглядит следующим образом

```
for count = start : step : final  
    команды MATLAB  
end
```

В данном примере count – счетчик цикла, start – начальное значение счетчика, step – шаг изменения счетчика, final – конечное значение счетчика. Счетчик может принимать не только целые значения, но и вещественные.

ПРИМЕР. Выведем на график семейство кривых для $x \in [0, 2\pi]$, которое задано функцией, зависящей от параметра a

$$y(x, a) = e^{-ax} \sin(x) .$$

Параметр изменяется от -0,1 до 0,1.

Текст программы выглядит следующим образом:

```
function graphic  
x=[0:pi/30:2*pi];  
for a=-0.1:0.02:0.1  
    y=exp(-a*x).*sin(x)  
    hold on  
    plot(x,y)  
end
```

Семейство кривых представлено на рис. 10

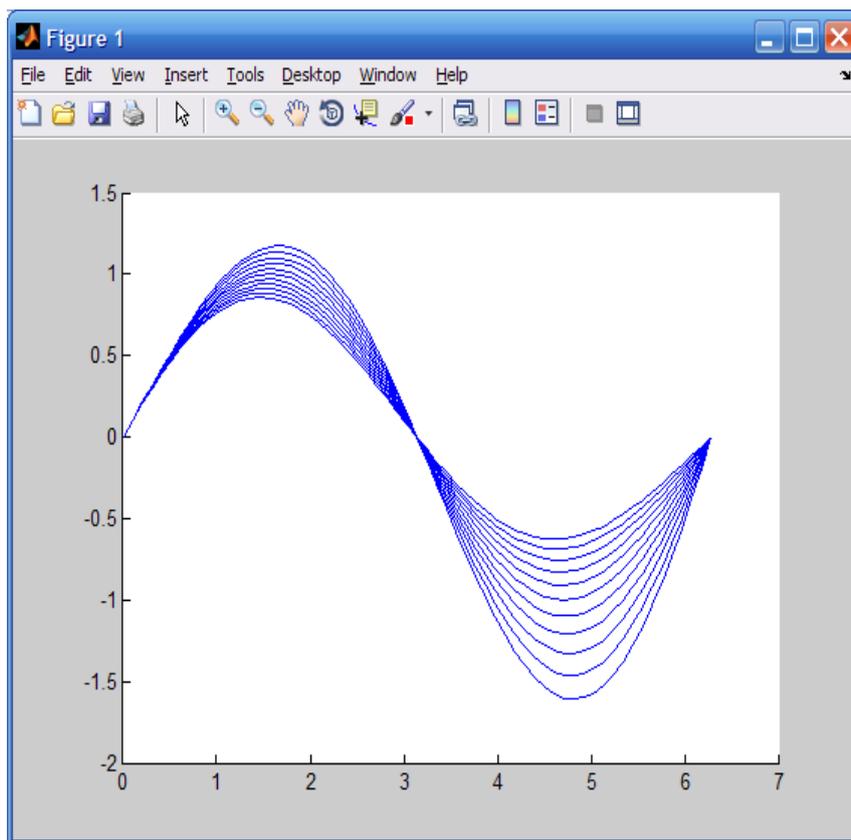


Рис. 10. Семейство кривых

Цикл for используется для вычисления накопления, суммы. При этом вначале сумма обнуляется, затем в каждом цикле i -тый элемент складывается с предыдущим.

Пример. Вычислить сумму $S = \sum_{k=1}^{10} \frac{1}{k!}$

```
function summa
S=0;
for k=1:10
    S=S+1/factorial(k);
end
```

Ответ

S=
1.7183

Циклы for могут быть вложены друг в друга, при этом переменные вложенных циклов должны быть разными.

Цикл while

Цикл `while` используется тогда, когда неизвестно количество необходимых итераций в теле цикла. Он выполняется до тех пор, пока выполняется условие цикла:

```
while условие цикла
    команды MATLAB
end
```

Для задания условий используются следующие операторы (табл. 4).

Таблица 4. Операции соотношения

Обозначение	Операции отношения	Обозначение	Операции отношения
<code>==</code>	Равенство	<code><=</code>	Меньше или равно
<code><</code>	Меньше	<code>>=</code>	Больше или равно
<code>></code>	Больше	<code>~=</code>	Не равно

Если необходимо задать сложное условие, то можно воспользоваться одним из предложенных ниже способов (табл. 5).

Таблица 5. Логические соотношения

Оператор	Условие	Способ 1	Способ 2
Логическое И	<code>x<3</code> и <code>y=4</code>	<code>and (x<3, y==4)</code>	<code>(x<3) & (y==4)</code>
Логическое ИЛИ	<code>x=1</code> или <code>x=2</code>	<code>or (x==1, x==2)</code>	<code>(x==1) (x==2)</code>
Отрицание НЕ	<code>a≠1.9</code>	<code>not (a==1.9)</code>	<code>~(a==1.9)</code>

Пример. Необходимо вычислить сумму ряда (разложение в ряд $\sin x$)

$$S(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$$

Будем вычислять ряд с точностью 10^{-10} , это значит накапливать и суммировать слагаемые, которые меньше по модулю этого значения.

Текст программы

```
function y=mysin(x)
s=0;
k=0;
while abs(x.^(2*k+1)/factorial(2*k+1)) > 1.e-10
    s=s+(-1)^k*(x.^(2*k+1)/factorial(2*k+1));
    k=k+1;
end
```

Проверим вычисленное значение созданной функции и библиотечной функции $\sin(x)$

```
>> mysinus(pi/2)
```

```
ans =
```

```
1.0000
```

```
>> sin(pi/2)
```

```
ans =
```

```
1
```

Условный оператор if

Условный оператор `if` применяется для реализации разветвленных алгоритмов. Обобщенная его структура такова:

```
if условие
    команды MATLAB
elseif условие
    команды MATLAB
else
    команды MATLAB
end
```

Пример использования оператора ветвления.

```
function vybor(a)
```

```
if (a==0)
```

```
    disp('a равно нулю')
```

```
elseif a==1
```

```
    disp ('a равно единице')
```

```
elseif a==2
```

```
    disp ('a равно двум')
```

```
elseif a>=3
```

```
    disp ('a больше или равно трем')
```

```
else
```

```
    disp ('a меньше трех, но не единица, не ноль и не два')
```

```
end
```

Вызов этой функции в командном окне

```
>> vybor(5)
```

```
а больше или равно трем
```

```
>> vybor(1)
```

```
а равно единице
```

```
>> vybor(-4)
```

```
а меньше трех, но не единица, не ноль и не два.
```

Оператор `disp` выводит на экран необходимое сообщение. Оператор `if` полезен для построения кусочно-заданных функций.

Пример. Построить функцию, заданную кусочно:

$$y(x) \begin{cases} \sin(x) - 1, & x < \pi \\ x / \pi, & -\pi \leq x < \pi \\ -\cos(x), & x \geq \pi. \end{cases}$$

Текст программы в MATLAB

```
function y=kus(x)
if x<pi
    y=sin(x)-1;
elseif x<pi % проверка x>pi не нужно
    y=x/pi;
else % здесь x>pi
    y=-cos(x);
end
```

При вызове функции `plot` в командном окне кусочная функция не построится.

```
>> x=[-3*pi:0.1:3*pi];
>> y=kus(x);
>> plot(x,y)
```

Правильнее в данном случае использовать функцию `fplot`

```
>> fplot('kus',[-3*pi, 3*pi])
```

График кусочно-заданной функции приведен на рис. 11.

Оператор switch

Оператор переключения `switch` позволяет разделить алгоритм программы на равнозначные ветви, каждая из которых выполняется, когда переменная оператора принимает одно из значений из списка. После выполнения какой-либо из ветвей происходит выход из `switch`. Если подходящих значений для переменной не нашлось, то выполняется ветвь программы, соответствующая `otherwise`.

Пример.

```
switch a
case -1
    disp('a = -1')
case 0
    disp('a = 0')
case 1
    disp('a = 1')
case (2, 3, 4)
```

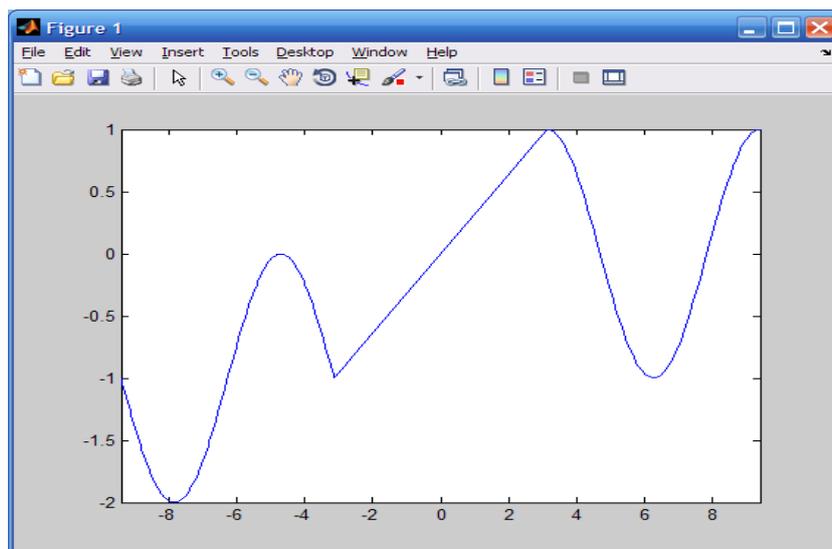


Рис. 11. Кусочно-заданная функция

```

disp ('а равно 2, 3 или 4')
otherwise
    disp ('а не равно -1, 0, 1')
end

```

Прерывание цикла. Исключительные ситуации

При организации циклических вычислений желательно, чтобы внутри цикла не возникало ошибок. Предотвратить выполнение цикла можно с помощью оператора `break`. Он заканчивает цикл и переходит к оператору `end` и осуществляются вычисления, следующие за `end`.

Если в программе появляются ошибки, приводящие к окончанию программы, то для избежания выхода из программы следует применить оператор `try...catch`. Структура его такова:

```

try
    операторы, выполнение которых может привести к ошибке
catch
    операторы, которые следует выполнить при возникновении ошибки
    в блоке между try и catch.

```

Запись в текстовый файл

Работа с файлами состоит из трех этапов:

- открытие файла;
- считывание/запись данных;
- закрытие данных.

Команда `fopen` открывает файл. Имя файла указывается в апострофах первым входным аргументом. Второй аргумент задает способ доступа к файлу.

- `'rt'` – открываемый текстовый файл предназначен только для чтения;
- `'rt+'` – открываемый текстовый файл предназначен для чтения и записи;
- `'wt'` – создаваемый пустой текстовый файл предназначен только для записи;
- `'wt+'` – создаваемый пустой текстовый файл предназначен для чтения и записи.

Желательно указывать полное имя и путь файла, при задании только имени MATLAB ищет файл только в текущем каталоге.

Информация выводится в текстовый файл при помощи функции `fprintf`.

```
fprintf (F, 'строка добавлена командой fprintf.')
```

Следующая команда выводит заданную строку сразу за предыдущей, а не на новой строке.

```
fprintf (F, 'еще одна строка.')
```

Для вывода текста с новой строки следует добавить символ перевода строки `\n` в начало новой строки после апострофа:

```
fprintf (F, '\n текст с новой строки.')
```

Символ перевода строки `\n` можно разместить в конце строки, после которой текст должен начинаться с новой строки.

```
function zapis_tabl
```

```
% открытие нового файла для записи
```

```
[F, mes] = fopen ('table.dat', 'w');
```

```
% печать в файл заголовка таблицы
```

```
fprintf (F, 'таблица значений функции \n');
```

```
% печать в файл шапки таблицы
```

```
fprintf (F, ' -----\n');
```

```
fprintf (F, ' |   x   |   y   |\n');
```

```
fprintf (F, ' -----\n');
```

```
% закрытие файла
```

```
fclose (F),
```

Функция `fprintf` позволяет управлять форматом записываемых чисел в файл.

```
fprintf (имя файла, 'формат', список переменных);
```

```
fprintf (F, '| %7.4f| %11.8f|', x, y);
```

Обозначение %7.4f задает формат вывода переменной x, которая расположена на первом месте в списке вывода. Знак % обозначает начало формата, цифра 7 обозначает количество позиций под целое значение x, цифра 4 – количество цифр после точки, буква f указывает на то, что выводится число в формате с плавающей точкой.

Задания к лабораторной работе № 3

Задание 1. Составить файл-функцию для вычисления кусочно-заданной функции:

- 1) для аргумента x, вводимого с клавиатуры, вычислить значения y;
- 2) для массива x вычислить значения y и вывести график.

№ п/п	Функция	№ п/п	Функция
1	$f(x) = \begin{cases} 4 * \sin(x) - 1, & \text{при } x < 1 \\ \ln(x) + 5, & \text{при } x \geq 1 \end{cases}$	7	$f(x) = \begin{cases} \sqrt{4 * x^4 + 3}, & \text{при } x < 2 \\ e^{2*x} - 0.5, & \text{при } x \geq 2 \end{cases}$
2	$f(x) = \begin{cases} 25 - x, & \text{при } x < 0 \\ \cos^2(x) - x, & \text{при } x \geq 0 \end{cases}$	8	$f(x) = \begin{cases} (3 * x^3 + 5) / (3 * x + 1), & \text{при } x < -1 \\ \ln x + 4 , & \text{при } x \geq -1 \end{cases}$
3	$f(x) = \begin{cases} \sqrt{4 * x^2 + 1}, & \text{при } x < 2 \\ e^x - 1, & \text{при } x \geq 0 \end{cases}$	9	$f(x) = \begin{cases} 2.5 * x / (6 * x + 4), & \text{при } x < -2 \\ (x^2 - 2) / 5, & \text{при } x \geq -2 \end{cases}$
4	$f(x) = \begin{cases} (x^2 + 5) / (x + 1), & \text{при } x < -1 \\ \ln x + 2 , & \text{при } x \geq -1 \end{cases}$	10	$f(x) = \begin{cases} -2 * x + 6, & \text{при } x < 10 \\ \cos^2(3 * x) + 2 * x, & \text{при } x \geq 10 \end{cases}$
5	$f(x) = \begin{cases} x / (3 * x + 2), & \text{при } x < -2 \\ x^2 - 2, & \text{при } x \geq -2 \end{cases}$	11	$f(x) = \begin{cases} \sqrt{(4 * x^2 + 3)} / 8, & \text{при } x < 3 \\ 3 * x * e^{5*x} + 1, & \text{при } x \geq 3 \end{cases}$
6	$f(x) = \begin{cases} -x + 5, & \text{при } x < 10 \\ \cos^2(x) + 2 * x, & \text{при } x \geq 10 \end{cases}$	12	$f(x) = \begin{cases} 3 - 5 * x * \cos(x), & \text{при } x < 5 \\ (x^2 + 2) / 6 * x, & \text{при } x \geq 5 \end{cases}$

№ п/п	Функция	№ п/п	Функция
13	$f(x) = \begin{cases} \sqrt{4 * x^2 + 3}, & \text{при } x < 3 \\ e^x + 1 & \text{при } x \geq 3 \end{cases}$	17	$f(x) = \begin{cases} 7 * \cos(x) + 2, & \text{при } x < 1 \\ 2 + (x - 1)^3, & \text{при } x \geq 1 \end{cases}$
14	$f(x) = \begin{cases} 1 - \cos(x), & \text{при } x < 5 \\ x^2 + 2 & \text{при } x \geq 5 \end{cases}$	18	$f(x) = \begin{cases} \sin(x - \pi/4), & \text{при } x < 0 \\ 2 + x - x^2, & \text{при } x \geq 0 \end{cases}$
15	$f(x) = \begin{cases} 6 * \cos(x) + 1, & \text{при } x < 3 \\ \ln(2 * x) + 15, & \text{при } x \geq 3 \end{cases}$	19	$f(x) = \begin{cases} (2x - 3) / (3x + 2), & \text{при } x < -1 \\ \ln x + 6 , & \text{при } x \geq -1 \end{cases}$
16	$f(x) = \begin{cases} (30 - 8 * x) / 1.5, & \text{при } x < 0 \\ \cos^2(x) - 6 * x, & \text{при } x \geq 0 \end{cases}$	20	$f(x) = \begin{cases} 2x^2 - x^4, & \text{при } x < 2 \\ \operatorname{tg}^2(x), & \text{при } x \geq 2 \end{cases}$

Задание 2. Составить программу для таблицы аргументов и значений функции Z на отрезке $[x_0, x_k]$ с шагом Δx , записать таблицу в файл.

№ п/п	Функция
1	$Z = 4 * (1 - y^2), \text{ где } y_0 = 1.5, \Delta y = 0.5, y_k = 4$
2	$Z = (x + 1)^3 - 2, \text{ где } x_0 = 0.5, \Delta x = 0.5, x_k = 4$
3	$Z = 1 + 2 * x - x^2, \text{ где } x_0 = 0, \Delta x = 0.5, x_k = 4.5$
4	$Z = \sqrt{ x + 1 }, \text{ где } x_0 = -4, \Delta x = 1, x_k = 4$
5	$Z = \sqrt{4 * x^2 + 1}, \text{ где } x_0 = 0, \Delta x = 0.2, x_k = 2.4$
6	$Z = e^{x^2} - 1, \text{ где } x_0 = -2, \Delta x = 0.2, x_k = 0$
7	$Z = 2 * \sqrt{x + 1} * \cos(x), \text{ где } x_0 = 0, \Delta x = 0.25, x_k = 3.5$
8	$Z = \ln(x) + 5 * x, \text{ где } x_0 = 0.5, \Delta x = 0.5, x_k = 6$
9	$Z = 6 * (1 - y^4), \text{ где } y_0 = 1, \Delta y = 0.5, y_k = 4.5$
10	$Z = (x + 1)^3 - 4 * x, \text{ где } x_0 = 0, \Delta x = 0.5, x_k = 4$
11	$Z = 4 + 2 * x - 3 * x^2, \text{ где } x_0 = 0, \Delta x = 0.5, x_k = 4.5$
12	$Z = \sqrt{ 2 * x + 1 }, \text{ где } x_0 = -4, \Delta x = 1, x_k = 4$
13	$Z = \sqrt{(4 * x^2 + 3) / x}, \text{ где } x_0 = 0, \Delta x = 0.2, x_k = 2.4$
14	$Z = 2 * x * e^{x^2} - 1, \text{ где } x_0 = -2, \Delta x = 0.2, x_k = 0$

№ п/п	Функция
15	$Z = 2 * \sqrt{x+1} * \sin(2 * x)$, где $x_0 = 0$, $\Delta x = 0.25$, $x_k = 3.5$
16	$Z = \ln(2 * x) + 5 * x$, где $x_0 = 0.5$, $\Delta x = 0.5$, $x_k = 6$
17	$Z = x / \sqrt[3]{x^2 - 4}$, где $x_0 = 0$, $\Delta x = 0.2$, $x_k = 3$
18	$Z = x(x-1)^2(x-2)^3$, где $x_0 = 0$, $\Delta x = 0.25$, $x_k = 4$
19	$Z = 2e^{x^2-4x}$, где $y_0 = 1$, $\Delta y = 0.5$, $y_k = 5$
20	$Z = 2x^3 + 3x^2 - 12x + 5$, где $x_0 = 0$, $\Delta x = 0.4$, $x_k = 4$

Задание 3. Составить простой цикл с предусловием для вычисления суммы ряда с указанной точностью и определить количество итераций.

№ п/п	Ряд	Замечания
1	$\sum_{n=0}^{\infty} \frac{x^n}{n}$	Подобрать различные x Точность 10^{-10}
2	$\sum_{n=0}^{\infty} (-1)^{n-1} \frac{x^n}{n}$	Подобрать различные x Точность 10^{-10}
3	$\sum_{n=0}^{\infty} \frac{x^{2n-1}}{2n-1}$	Подобрать различные x Точность 10^{-10}
4	$\sum_{n=0}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{2n-1}$	Подобрать различные x Точность 10^{-10}
5	$\sum_{n=0}^{\infty} (n+1)x^n$	Подобрать различные x Точность 10^{-10}
6	$\sum_{n=0}^{\infty} (-1)^{n-1} (2n-1)x^{2n-2}$	Подобрать различные x Точность 10^{-10}
7	$\sum_{n=0}^{\infty} n(n+1)x^{n-1}$	Подобрать различные x Точность 10^{-10}
8	$\sum_{n=0}^{\infty} n/x^n$	Подобрать различные x Точность 10^{-10}
9	$\sum_{n=0}^{\infty} x^{4n-3} / (4n-3)$	Подобрать различные x Точность 10^{-10}
10	$\sum_{n=0}^{\infty} \frac{(-1)^{n-1}}{(2n-1)3^{n-1}}$	Точность 10^{-9}

№ п/п	Ряд	Замечания
11	$\sum_{n=0}^{\infty} \frac{2n-1}{2^n}$	Точность 10^{-8}
12	$\sum_{n=0}^{\infty} \frac{(1+n)}{3^n}$	Точность 10^{-9}
13	$\sum_{n=0}^{\infty} \frac{(-1)^n - n}{3^n}$	Точность 10^{-8}
14	$\sum_{n=1}^{\infty} \frac{(1+n)}{3^n}$	Точность 10^{-9}
15	$\sum_{n=1}^{\infty} \frac{n}{(2n+1)5^n}$	Точность 10^{-8}
16	$\sum_{n=1}^{\infty} \frac{1}{n(n+1)}$	Точность 10^{-9}
17	$\sum_{k=1}^{\infty} (2^{k+1}\sqrt{x} - 2^k\sqrt{x})$	Подобрать различные x Точность 10^{-10}
18	$\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n}$	Точность 10^{-8}
19	$\sum_{n=1}^{\infty} \frac{1}{n!2^n}$	Точность 10^{-9}
20	$\sum_{n=1}^{\infty} \frac{(-1)^n}{n!}$	Точность 10^{-8}

Содержание отчета

Отчет оформляется каждым студентом самостоятельно. Защита происходит в начале каждого следующего занятия с демонстрацией работы программы на ЭВМ. Студент, не подготовивший или не защитивший отчет по работе, к следующей лабораторной работе не допускается.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Вариант задания.
4. Текст программы в MATLAB и полученные графики.
5. Выводы по работе.

Контрольные вопросы и задания

1. Опишите редактор М-файлов в MATLAB.
2. Объясните, как Вы понимаете отличие файл-функции

от файл-сценария?

3. Как задаются входные и выходные аргументы функции?

4. В чем отличие цикла for от цикла while?

5. Каков синтаксис оператора ветвления в MATLAB?

Лабораторная работа № 4

ИССЛЕДОВАНИЕ ФУНКЦИЙ НА ОСНОВЕ ЧИСЛЕННЫХ МЕТОДОВ В MATLAB

Цель работы: научиться использовать встроенные функции для анализа функций в MATLAB.

Оборудование: дисплейный класс, программа MATLAB R2008 или выше.

Решение уравнений

При анализе экспериментальных данных и при исследовании измерительных преобразований возникает потребность использовать в программе численные функции, реализующие численные методы.

В MATLAB используется множество встроенных функций, реализующих численные методы и охватывающих широкий спектр задач: решение уравнений, нахождение максимума или минимума функции, интегрирование, интерполирование, решение дифференциальных уравнений. Для работы со встроенными функциями необходимо сначала написать файл-функцию для вычисления исследуемой функции.

Нахождение корней произвольных уравнений осуществляет встроенная функция fzero, для определения корней полиномов применяется roots.

Функция fzero позволяет приближенно вычислить корень уравнения по заданному начальному приближению (значение или отрезок).

fzero (имя функции, приближение).

Имя функции задается в апострофах или начинаясь с символа @.

Пример. Найти корни уравнения $\sin(x) - x^2 \cos(x) = 0$ на отрезке $[-5, 5]$.

Сначала необходимо построить график функции. Для этого зададим файл-функцию.

Текст программы:

```
function uavnenie
```

```
% рисуем график функции
```

```
fplot(@myfunction,[-5 5])
```

```
% чертим сетку на графике
```

```
grid on
```

```
% значение корня вблизи -5
```

```
x1=fzero(@myfunction,-5)
```

```
% вычислим значение функции в точке x1
```

```

y1=myfunction(x1)
% поиск корня на интервале
x2=fzero(@myfunction, [-3 -1])

% значение корня и значение функции
[x2, f]=fzero(@myfunction, [-3 -1])
function y=myfunction(x)
y=sin(x)-x.^2.*cos(x);
Результаты вычислений в программе
x1 =
    -4.7566
y1 =

```

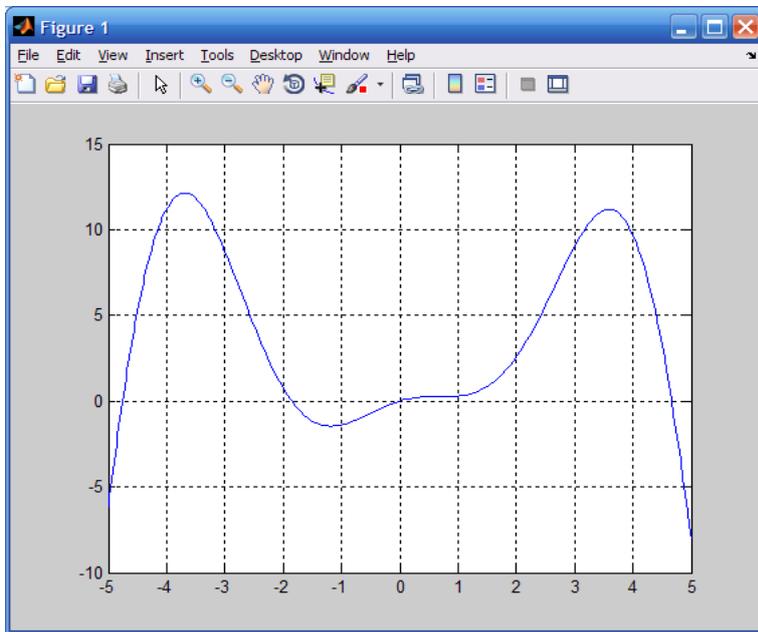


Рис. 12. График функции

```

p=x^7+3,2x^5-5,2x^4+0,5x^2-3
p=[1 0 3.2 -5.2 0 0.5 1 -3]

```

С помощью функции `polyval` можно вычислить значение полинома от некоторого аргумента.

```

polyval(p,0)
ans =
    -3

```

Функция `roots` вычисляет корни данного уравнения, в том числе комплексные.

```

>> r=roots(p)
r =
    -0.5668 + 2.0698i
    -0.5668 - 2.0698i

```

```

2.6645e-015
x2 =
    -1.8539
x2 =
    -1.8539
f =
    -2.2204e-016

```

График функции представлен на рис. 12.

Корни полинома находят с помощью функции `roots`, в качестве аргумента которой вводится вектор коэффициентов полинома. Полином в MATLAB задается вектором его коэффициентов

```

-0.6305 + 0.5534i
-0.6305 - 0.5534i
1.2149
0.5898 + 0.6435i
0.5898 - 0.6435i

```

Поиск минимумов и максимумов функции

Поиск локального минимума осуществляется функцией `fminbnd`. Перед нахождением минимума нужно построить график.

```

function minimum
% рисуем график функции
fplot(@myfunction,[0 2])
% чертим сетку на графике
grid on
% значение минимума на интервале 0,7...2
x1=fminbnd(@myfunction,0.7,2.0)
% значение минимума и значение функции
[x2, f]=fminbnd(@myfunction, 0.7, 2.0)
% значение максимума и значение функции
x3=fminbnd(@myfunction1, 0, 0.4)
% значение функции в точке максимума
f3=-myfunction1(x3)
function y=myfunction(x)
y=exp(-x).*sin(3*pi*x);
function y=myfunction1(x)
y=-(exp(-x).*sin(3*pi*x));
Результаты вычислений

```

```

x1 =
    1.1554
x2 =
    1.1554
f =
   -0.3132
x3 =
    0.1555
f3 =
    0.8513

```

График функции изображен на рис. 13.

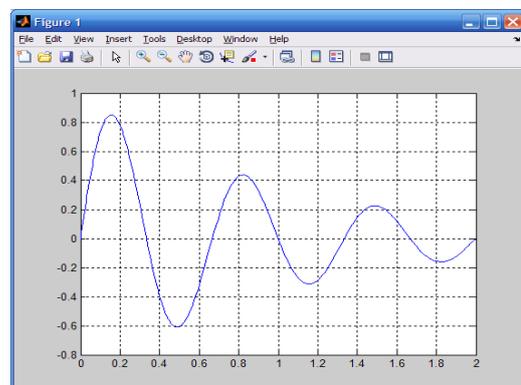


Рис. 13. График функции

Интегрирование функции

В MATLAB имеется возможность вычисления определенных интегралов и двойных определенных интегралов с заданной точностью. При этом можно выбирать наиболее подходящий метод численного интегрирования.

Для вычисления определенного интеграла применяется функция `quad`. Ее первый аргумент – интегрируемая функция, второй и третий аргументы – нижний и верхний пределы интегрирования.

Пример

$$\int_{-1}^1 e^{-x} \sin x dx$$

Текст программы

```
function integral
I=quad(@fint,-1,1)
function f=fint(x)
f=exp(-x).*sin(x)
```

Результат вычисления

```
I =
-0.663491467853105
```

По умолчанию задана точность 10^{-3} . Если необходимо повысить точность, то нужно задать дополнительный четвертый аргумент функции `quad`.

```
I1=quad(@fint,-1,1, 1.0e-9)
```

```
I1 =
-0.663493666622457
```

Двойные интегралы вычисляются с помощью функции `dblquad`.

Пример

$$\int_0^1 \int_{-\pi}^{\pi} (e^x \sin y + e^{-x} \cos y) dx dy$$

Текст программы:

```
function integral
I2=dblquad(@fint2,-pi, pi, 0,1)
function f=fint2(x,y)
f=exp(-x).*sin(y).^2+exp(-x).*cos(y).^2;
```

Результат вычислений

```
I2 =
23.097478731539596
```

Решение дифференциальных уравнений

Для решения обыкновенных дифференциальных уравнений произвольного порядка и систем с начальными условиями (задач Коши) предназначены специальные встроенные функции-солверы. MATLAB имеет достаточно большой выбор солверов, реализующих различные методы. Пользователю необходимо выбрать подходящий солвер в зависимости от свойств решаемой задачи и произвести необходимые установки, обеспечивающие получение решения с заданной точностью.

Какие солверы используются, можно посмотреть в справочной системе MATLAB.

ODE solvers: ode45 ode23, ode113, ode15s, ode23s, ode23t, ode23tb
implicit ODEs: ode15i
options handling: odeset, odeget
output functions: odeplot, odephas2, odephas3, odeprint
evaluating solution: deval
ODE examples: rigidode, ballode, orbitode
function handles: function_handle

Схема решения состоит из следующих этапов:

1. Приведение дифференциального уравнения к системе дифференциальных уравнений первого порядка.
2. Написание файл-функции для системы уравнений.
3. Вызов подходящего солвера.
4. Визуализация результата.

Рассмотрим в качестве примера уравнение

$$y'' + 2y' + 10y = \sin(t),$$

начальные условия $y(0)=1, y'(0)=0$.

Вводим вспомогательные функции y_1 и y_2 .

$$y_1=y \quad y_2=y'$$

$$\begin{cases} y_1' = y_2 \\ y_2' = -2y_2 - 10y_1 + \sin t \end{cases} \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Текст программы:

```
function diffur
% начальные условия
Y0=[1;0];
% вызов солвера от созданной файл-функции,
% задание начального и конечного момента времени и % вектора
начальных условий
[T, Y]=ode45(@func,[0 15], Y0);
% вывод графика решения исходного дифференциального уравнения
```

```

plot(T, Y(:,1),'r')
hold on
% вывод на график производной
plot (T, Y(:,2),'k--')

```

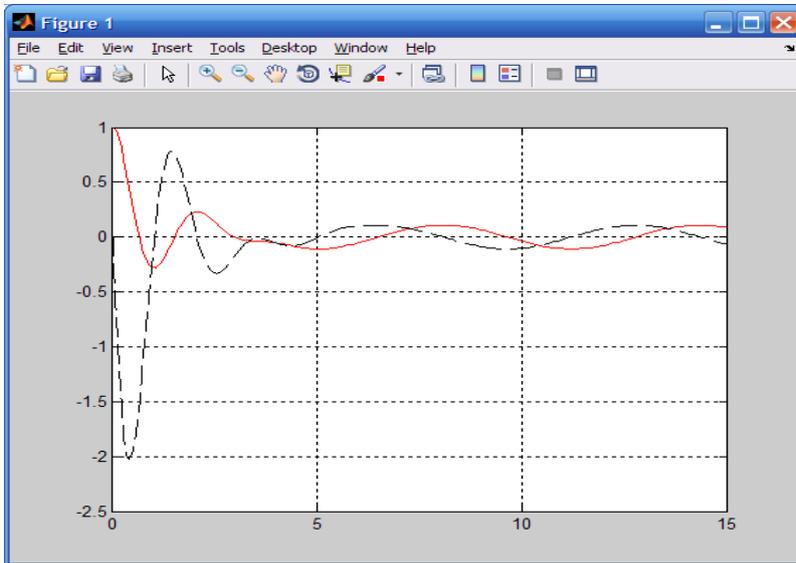


Рис. 14. График функции и ее производной

```

grid on
function F=func(t,y)
F=[y(2); -2*y(2)-
10*y(1)+sin(t)];

```

График функций (рис. 14).

Задания к лабораторной работе № 4
Задание 1. Решение нелинейного уравнения.

1. Создать Mat-функцию для функции $f_1(x)$.
2. Создать файл программы.

Ввести текст заглавия задачи как комментарий. Ввести в него аргументы в заданных пределах.

3. Вывести $y(x)=f_1(x)$ в виде XY графика. По нему определить приближенно корни уравнения $y(x)=0$. Если корни на графике не просматриваются, то изменить пределы изменения аргумента и повторить операции.

4. Для каждого корня найти точное значение, используя функцию `fzero`.

Задание 2. Решение системы из двух нелинейных уравнений.

1. Создать Mat-функции для функций $f_2(x)$ и $f_3(x) = f_1(x) - f_2(x)$.
2. Создать файл программы. Ввести текст заглавия задачи как комментарий. Ввести в него аргументы в заданных пределах.
3. Вывести $f_1(x)$ и $f_2(x)$ в виде XY графиков. По нему определить приближенно корни системы уравнений как координаты точек пересечения графиков $f_1(x)$ и $f_2(x)$. Если корни на графике не просматриваются, то изменить пределы изменения аргумента и повторить операции.
4. Для каждого корня найти точное значение, используя функцию `fzero` к переменной $f_3(x)$.

Варианты заданий:

№ п/п	f1(x)- полином 3-й степени с коэффициентами а				f2(x)
	а3	а2	а1	а0	
1	0	-1	4	-1	0.2exp(x)-20
2	0	2	-2	-15	40 cos(x)
3	0	1	4	-1	10ln(x+5.5)
4	0	9	-8	-70	100 sin(x)
5	0	-4	4	50	70cos(x)
6	1	-5	4	40	60exp(0.1*x)-100
7	2	-3	2	30	20sin(2x)
8	3	-6	1	50	exp(x)sin(2x)
9	4	-9	1	70	exp(x)cos(3x)
10	5	-7	5	60	-60 cos(x)
11	-1	-4	9	60	15log(x+5.1)
12	-2	-6	-7	55	-50ln(x+5.1)
13	-3	-9	-8	75	-100 cos(x)
14	-4	7	8	-75	100sin(x/2)
15	-5	1	4	-1	40cos(x/2)
16	-0,85	-2	7	2	6cos(x)-5
Для заданий 17-20 решить уравнения и системы уравнений					
	Уравнения		Системы уравнений		
17	$2x^2-5x+2$		$xy=6$	$x^2+y^2=13$	
18	X^3+x-1		$x^2-x+y=4$	$y^2-2x=0$	
19	$lgx=0.1x$		$x^2+y=10$	$x+y^2=6$	
20	$x=1+0.5\sin(x)$		$y=\sin(x)$	$y=\cos(x)$	

Задание 3 Исследовать на экстремумы функции.

1	$2 \cos \frac{x}{2} - x^2 + 0.4x + 1.4$	6	$2 \cos \frac{x}{2} + 0,5x^2 - 3x + 2,88$
2	$2 \sin \frac{x}{2} - x^2 + 3x - 1.25$	7	$2 \sin \frac{x}{2} + x^2 + x - 2$
3	$\ln(2x+3) - x^2 + 10x - 24$	8	$\ln(x+3) + x^2 - 4,4x - 3$
4	$\ln(2-x) - 0.5x^2 - x$	9	$\ln(3-x) + x^2 + 4,4x - 3$
5	$2 \sin^2 \frac{x}{2} - x^2 - 5x - 4.76$	10	$2 \cos^2 \frac{x}{2} + x^2 + 5x + 3,69$

11	$-\sin \frac{x}{2} - 2x + 0,4$	16	$-\sin \frac{x}{2} + x - 3$
12	$\cos \frac{x}{2} - 2x + 3$	17	$\cos \frac{x}{2} + 2x + 1$
13	$\frac{2}{2x+3} - 2x + 10$	18	$\frac{1}{x+3} + 2x - 4,4$
14	$\frac{1}{x-2} - x - 1$	19	$\frac{1}{x-3} + 2x + 4,4$
15	$\sin x - 2x - 5$	20	$-\sin x + 2x + 5$

Задание 4. Интегрирование функций.

Вычислить определенный интеграл, предварительно задав необходимую функцию

1	$\int_1^2 (x^2 - 2x + 3) dx$	11	$\int_0^{\sqrt{2}/2} \frac{1}{\sqrt{1-x^2}} dx$
2	$\int_0^8 (\sqrt{2x} + \sqrt[3]{x}) dx$	12	$\int_2^{2.5} \frac{1}{\sqrt{5+4x-x^2}} dx$
3	$\int_1^4 \frac{1+\sqrt{y}}{y^2} dy$	13	$\int_0^1 \frac{y^2}{\sqrt{y^6+4}} dy$
4	$\int_2^6 \sqrt{x-2} dx$	14	$\int_0^{\pi/4} \cos^2 x dx$
5	$\int_0^{-3} \frac{1}{\sqrt{25+3x}} dx$	15	$\int_0^{\pi/2} \sin^3 x dx$
6	$\int_0^1 \frac{x}{x^2+3x+2} dx$	16	$\int_e^e \frac{1}{x \ln x} dx$
7	$\int_{-1}^1 \frac{y^5}{y+2} dy$	17	$\int_1^e \frac{\sin(\ln(x))}{x} dx$
8	$\int_0^1 \frac{1}{x^2+4x+5} dx$	18	$\int_{-\pi/4}^{\pi/4} tg x dx$
9	$\int_3^4 \frac{1}{x^2-3x+2} dx$	19	$\int_0^1 \frac{e^x}{1+e^{2x}} dx$
10	$\int_0^1 \frac{z^3}{z^8+1} dz$	20	$\int_{\pi/6}^{\pi/3} ctg^4 x dx$

Задание 5. С применением интегрального исчисления вычислить и представить графически следующие задачи

1	Вычислить площадь, ограниченную параболой $y=4x-x^2$ и осью абсцисс
2	Вычислить площадь, ограниченную кривой $y=\ln x$, осью абсцисс и прямой $x=e$
3	Вычислить площадь, ограниченную кривой $y=x(x-1)(x-2)$ и осью абсцисс
4	Вычислить площадь, ограниченную кривой $y^3=x$, прямой $y=1$, вертикалью $x=8$
5	Вычислить площадь, ограниченную одной полуволной синусоиды $y=\sin(x)$ и осью абсцисс
6	Вычислить площадь, ограниченную кривой $y=\operatorname{tg}x$, осью абсцисс и прямой $x=\pi/3$
7	Вычислить площадь, ограниченную кривой $y=x^3$, осью ординат и прямой $y=8$
8	Вычислить площадь, ограниченную кривой $y=2x-x^2$ и прямой $y=-x$
9	Вычислить площадь, ограниченную кривой $y=3-2x$ и параболой $y=x^2$
10	Вычислить площадь, ограниченную кривыми $y=x^2$, $y=x^2/2$ и прямой $y=2x$
11	Вычислить площадь, ограниченную кривыми $y=x^2/3$ и $y=4-2/3x^2$
12	Вычислить площадь, ограниченную кривыми $y=1/(1+x^2)$ и $y=x^2/2$
13	Вычислить площадь, ограниченную кривыми $y=e^x$, $y=e^{-x}$ и прямой $x=1$
14	Вычислить площадь, ограниченную кривой $y=1/x^2$, осью абсцисс и прямой $x=1$
15	Вычислить площадь, содержащуюся внутри кривой $(x/5)^2+(y/4)^{2/3}=1$
16	Вычислить площади двух частей, на которые круг $x^2+y^2=8$ разделен параболой $y^2=2x$
17	Найти объем тела, образованного при вращении вокруг оси OX кривой $y=\sin^2 x$ в промежутке от $x=0$ до $x=\pi$
18	Найти объем тела, образованного при вращении площади, ограниченной полукубической параболой $y^2=x^3$, осью OX и прямой $x=1$, вокруг оси OX
19	Найти объем тела, образованного при вращении площади, ограниченной полукубической параболой $y^2=x^3$, осью OX и прямой $x=1$, вокруг оси OY
20	Найти объем тела, образованного при вращении вокруг оси X площади, содержащейся между параболой $y=x^2$ и $y=\sqrt{x}$

Задание 6. Решение задачи Коши.

1. Приведение дифференциального уравнения к системе дифференциальных уравнений первого порядка.
2. Написание специальной файл-функции для системы уравнений.
3. Вызов подходящего солвера.
4. Визуализация результата.

1	$(1+x^2)y''-2xy'=0 \quad y=0 \quad y'=3 \quad \text{при } x=0$
2	$1+y'^2=2yy'' \quad y=1 \quad y'=1 \quad \text{при } x=1$
3	$yy''+y'^2=y'^3 \quad y=1 \quad y'=1 \quad \text{при } x=0$
4	$xy''=y' \quad y=0 \quad y'=0 \quad \text{при } x=0$
5	$y''y^3=1 \quad y=1 \quad y'=1 \quad \text{при } x=1/2$
6	$yy''-2xy'=0 \quad y=0 \quad y'=3 \quad \text{при } x=0$
7	$xy''\sqrt{1+y'^2}=0 \quad y=0 \quad \text{при } x=1 \quad y=1 \quad \text{при } x=e^2$
8	$y''(1+\ln x)+1/x*y'=2+\ln x \quad y=1/2 \quad y'=1 \quad \text{при } x=1$
9	$y''=y'/x*(1+\ln(y'/x)) \quad y=1/2 \quad y'=1 \quad \text{при } x=1$
10	$y''-y'^2+y'(y-1)=0 \quad y=2 \quad y'=2 \quad \text{при } x=0$
11	$3y'y''=y+y'^3+1 \quad y=-2 \quad y'=0 \quad \text{при } x=0$
12	$y^2+y'^2-2yy''=0 \quad y=1 \quad y'=1 \quad \text{при } x=0$
13	$y'y''+y'^2+yy''=0 \quad y=1 \quad \text{при } x=0 \quad y=0 \quad \text{при } x=-1$
14	$2y'+(y'^2-6x)y''=0 \quad y=0 \quad y'=2 \quad \text{при } x=2$
15	$y'y'^2+yy''-y'^2=0 \quad y=1 \quad y'=2 \quad \text{при } x=0$
16	$2yy''-3y'^2=4y^2 \quad y=1 \quad y'=0 \quad \text{при } x=0$
17	$2yy''+y^2-y'^2=0 \quad y=1 \quad y'=1 \quad \text{при } x=0$
18	$y''=y'^2-y \quad y=-1/4 \quad y'=1/2 \quad \text{при } x=1$
19	$1+yy''+y'^2=0 \quad y=0 \quad y'=1 \quad \text{при } x=1$
20	$(1+yy')y''=(1+y'^2)y' \quad y=1 \quad y'=1 \quad \text{при } x=0$

Содержание отчета

Отчет оформляется каждым студентом самостоятельно. Защита происходит в начале каждого следующего занятия с демонстрацией работ программы на ЭВМ. Студент, не подготовивший или не защитивший отчет по работе, к следующей лабораторной работе не допускается.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Вариант задания.
4. Текст программы в MATLAB и полученные графики.
5. Выводы по работе.

Контрольные вопросы и задания

1. Какая функция применяется для нахождения корней произвольных уравнений?
2. Объясните, как организовать вызов небиблиотечной функции в MATLAB?
3. Как найти максимум функции?
4. Как можно увеличить точность вычисления интеграла?
5. Опишите, какие решатели дифференциальных уравнений используются в MATLAB и в чем особенность каждого из них?

Лабораторная работа № 5

СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ В MATLAB

Цель работы: научиться создавать приложения с графическим пользовательским интерфейсом в MATLAB.

Оборудование: дисплейный класс, программа MATLAB R2008 или выше.

Принципы создания приложений с GUI

Создание графических приложений позволяет сделать программу более наглядной и понятной для независимого пользователя. Ему не нужно вникать в смысл каждой строки программы и не нужно разбираться, где поменять входные данные.

Приложения MATLAB являются графическими окнами, содержащими элементы управления (кнопки, списки, переключатели, флаги, полосы скроллинга, области ввода, меню), а также оси и текстовые области для вывода результатов.

Приложения – GUI (graphical user interfaces – графические пользовательские интерфейсы) создаются в среде GUIDE (рис. 15). Для запуска этой среды нужно нажать пиктограмму GUIDE в панели command window или набрать guide в командной строке. Далее во вкладке Create New Gui выбираем Blank GUI.

Редактор GUIDE содержит:

- строку меню;
- панель инструментов управления (вверху);

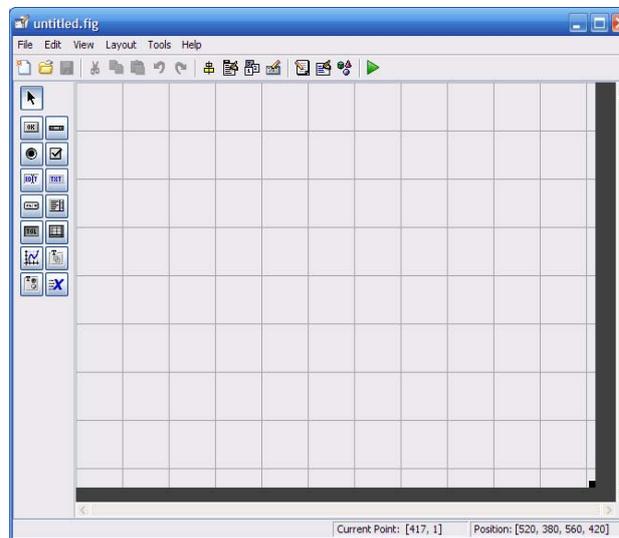


Рис. 15. Среда GUIDE MATLAB R2008

- заготовку окна приложения с нанесенной сеткой;
- вертикальную и горизонтальную линейки прокрутки;
- панель инструментов объектов интерфейса (слева) (рис. 16)

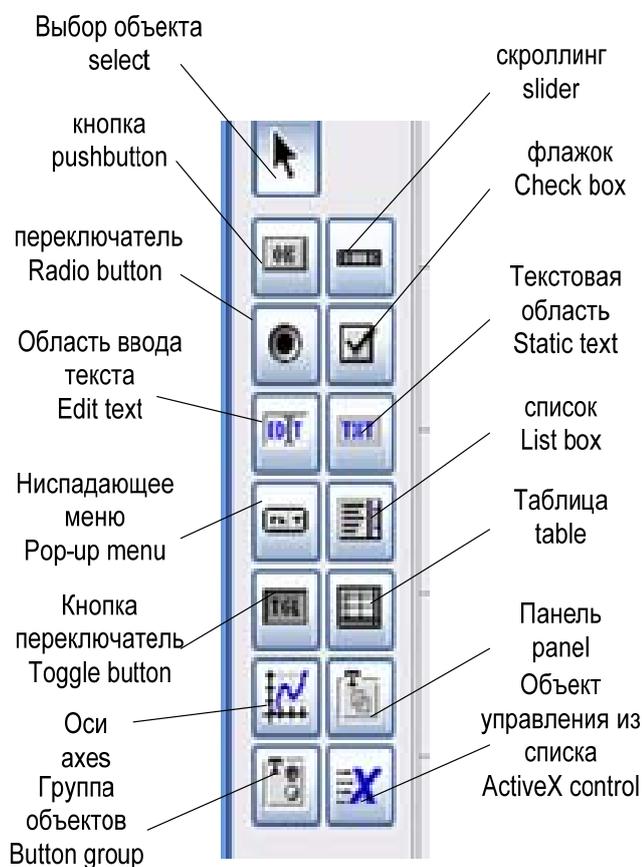


Рис. 16. Панель инструментов для создания интерфейса

Создадим простое приложение, содержащее оси и две кнопки построить график и очистить график. Сначала выберем Axes из панели инструментов и разместим их в центре окна. Затем выберем две кнопки Push Button и разместим их под графиком. При нажатии на кнопку правой клавиши мыши вызовем инспектор свойств Property Inspector. В нем выберем свойство String и впишем «Построить» и свойство Tag и впишем btnplot. Удобно задавать имена, часть которых соответствует типу объекта управления. Для второй кнопки в поле свойства String внесем «Очистить», а в поле свойства Tag – btnclear (рис. 17). Свойство String соответствует надписи на кнопке, а свойство Tag соответствует имени или тегу объекта в программе.

Затем сохраним файл и запустим программу, но при нажатии на кнопки ничего не происходит, так как не прописана обработка событий.

При сохранении GUI приложения автоматически генерируется m-файл, в котором необходимо прописывать обработку событий при нажатии на кнопку. Для этого в файле найдем функцию обработки событий при нажатии на кнопку «Построить» и пропишем построение графика.

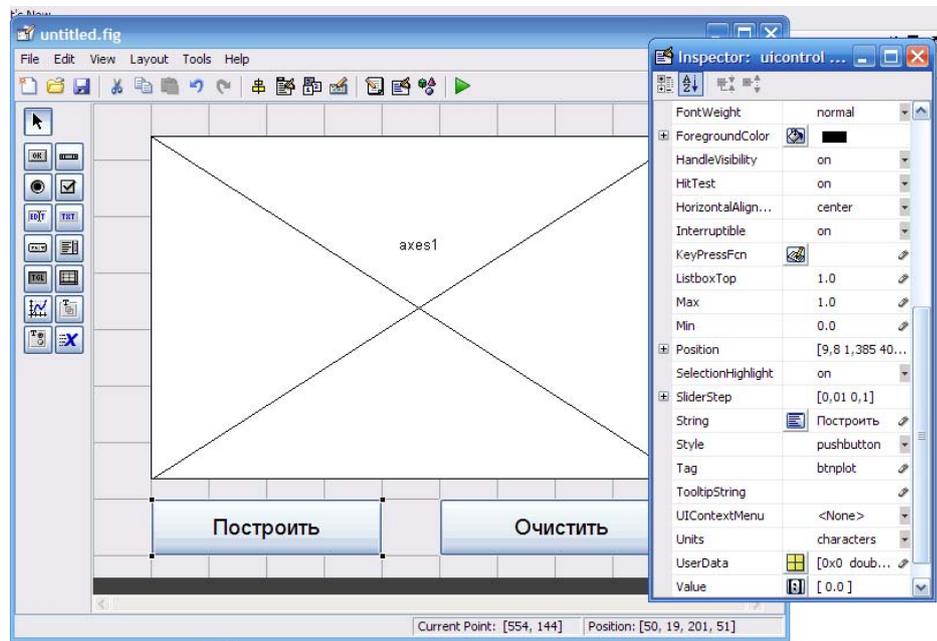


Рис. 17. Окно приложения с двумя кнопками

```
function btnplot_Callback(hObject, eventdata, handles)
% hObject   handle to btnplot (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
x=[-2:0.2:2];
y=exp(-x.^2);
plot(x,y)
```

Обработка события при нажатии на кнопку «Очистить» с помощью функции `cla` (clear axes).

```
function btnclear_Callback(hObject, eventdata, handles)
% hObject   handle to btnclear (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
cla
```

Управлять свойствами объектов можно в m-файле. Сделаем так, чтобы в любой момент времени была доступна только одна кнопка. Если графика нет, то доступна кнопка «Построить», если график нарисован, то доступна кнопка «Очистить». Это делается с помощью свойства «Enable» -

значение `on` – разрешает доступ, `off` – запрещает. Установка нужных свойств устанавливается функцией `set`. При изменении свойств любого объекта используем указатель на этот объект `handles`.

```
function btnplot_Callback(hObject, eventdata, handles)
% hObject handle to btnplot (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
x=[-2:0.2:2];
y=exp(-x.^2);
plot(x,y)
% кнопка должна быть недоступной после вывода графика
set(hObject, 'Enable', 'on')
% кнопка «Очистить» должна стать доступной
set(handles.btnclear, 'Enable', 'off')
% --- Executes on button press in btnclear.
function btnclear_Callback(hObject, eventdata, handles)
% hObject handle to btnclear (see GCBO)
% eventdata reserved-to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
cla
% кнопка должна быть недоступной после очистки осей
set(hObject, 'Enable', 'on')
% кнопка «Построить» должна стать доступной
set(handles.btnplot, 'Enable', 'off')
```

Теперь разместим в окне приложения два флажка (`check box`). Один будет управлять сеткой по оси `X`, а другой – сеткой по оси `Y`. В свойствах установим в `String` – сетка по `X` или сетка по `Y`, в свойстве `Tag` – `chbxGridX` и `chbxGridY`. Нас интересует еще свойство `Value`, которое принимает значение `1` при включении флага и `0` при выключении.

Сделаем так, чтобы при нажатии пользователем кнопки «Построить» происходило отображение линий сетки в зависимости от установленных флагов, а нажатие кнопки «Очистить» приводило к скрытию сетки. Блок обработки события `Callback` кнопки «Построить» нужно дополнить проверкой состояния флагов. Указатели на флаги содержатся в полях `chbxGridX` и `chbxGridY` структуры `handles`. Состояние флагов определяет значение свойств `XGrid` и `YGrid` осей.

```
function btnplot_Callback(hObject, eventdata, handles)
% hObject handle to btnplot (see GCBO)
% eventdata reserved-to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```

x=[-2:0.2:2];
y=exp(-x.^2);
plot(x,y)
% проверка флага Сетка по X
if get(handles.chbxGridX, 'Value')
    % флаг включен, строим сетку по X
    set(gca, 'XGrid','on')
else
    % флаг выключен, убираем сетку по X
    set(gca, 'XGrid','off')
end
% проверка флага Сетка по Y
if get(handles.chbxGridY, 'Value')
    % флаг включен, строим сетку по Y
    set(gca, 'YGrid','on')
else
    % флаг выключен, убираем сетку по Y
    set(gca, 'YGrid','off')
end

```

Смена состояния флагов сетки не приводит к немедленным изменениям на графике. Необходимо проверять состояние флагов для отображения или скрытия сетки.

```

function chbxGridX_Callback(hObject, eventdata, handles)
% hObject handle to chbxGridX (see GCBO)
% eventdata reserved-to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of chbxGridX
% проверка флага по оси X
if get(hObject, 'Value')
    % флаг включен, следует добавить линии сетки
    set(gca, 'XGrid', 'on')
else
    % флаг выключен, следует убрать линии сетки
    set(gca, 'XGrid','off')
end

% --- Executes on button press in chbxGridY.
function chbxGridY_Callback(hObject, eventdata, handles)
% hObject handle to chbxGridY (see GCBO)

```

```

% eventdata reserved-to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of chbxGridY
% проверка флага по оси Y
if get (hObject, 'Value')
    % флаг включен, следует добавить линии сетки
    set (gca, 'YGrid', 'on')
else
    % флаг выключен, следует убрать линии сетки
    set (gca, 'YGrid','off')
end

```

Приложение после проделанных изменений выглядит следующим образом (рис. 18).

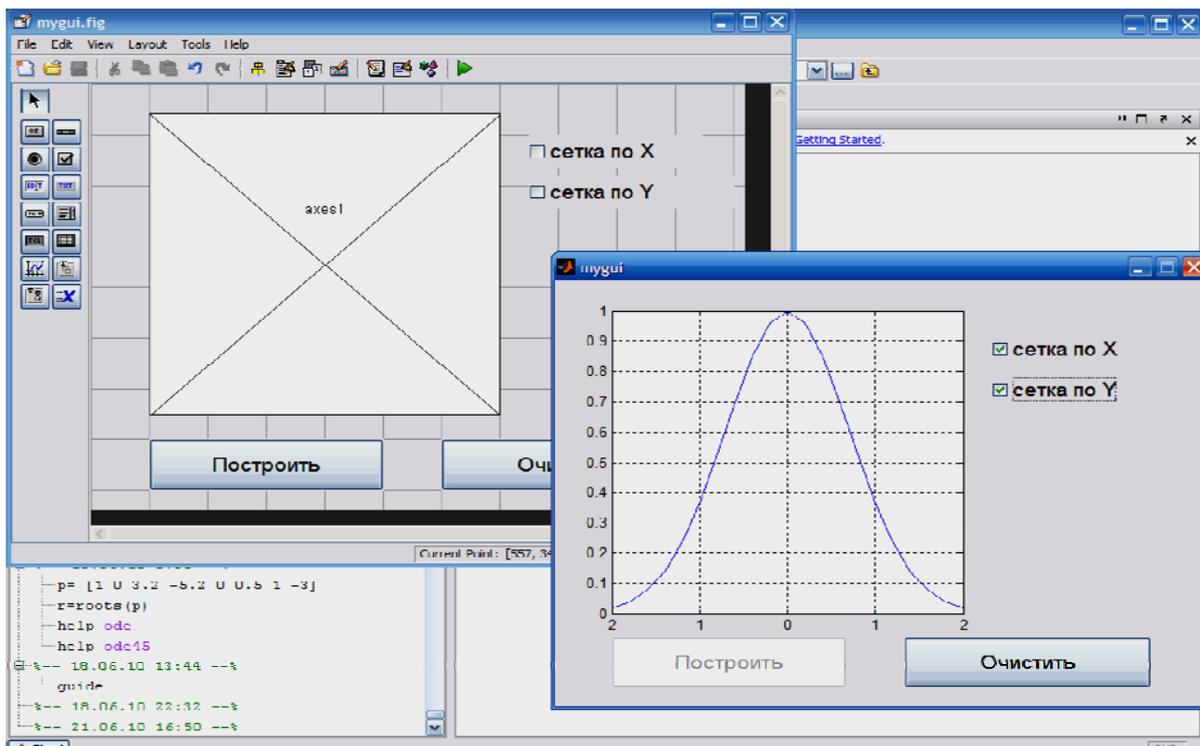


Рис. 18. Пользовательский интерфейс

Флаги предоставляют возможность выбора одной или сразу нескольких опций. Одновременный выбор только одной опции осуществляется при помощи переключателей.

Переключатели группируются по назначению. Применим переключатели для выбора типа маркера (кружок, квадрат или отсутствие маркера). Добавьте в окно приложения три переключателя и установите свойствам

Tag значения rbMarkCirk, rbMarkSq, rbMarkNone, а String – маркеры-круги, маркеры-квадраты и без маркеров (рис. 19).

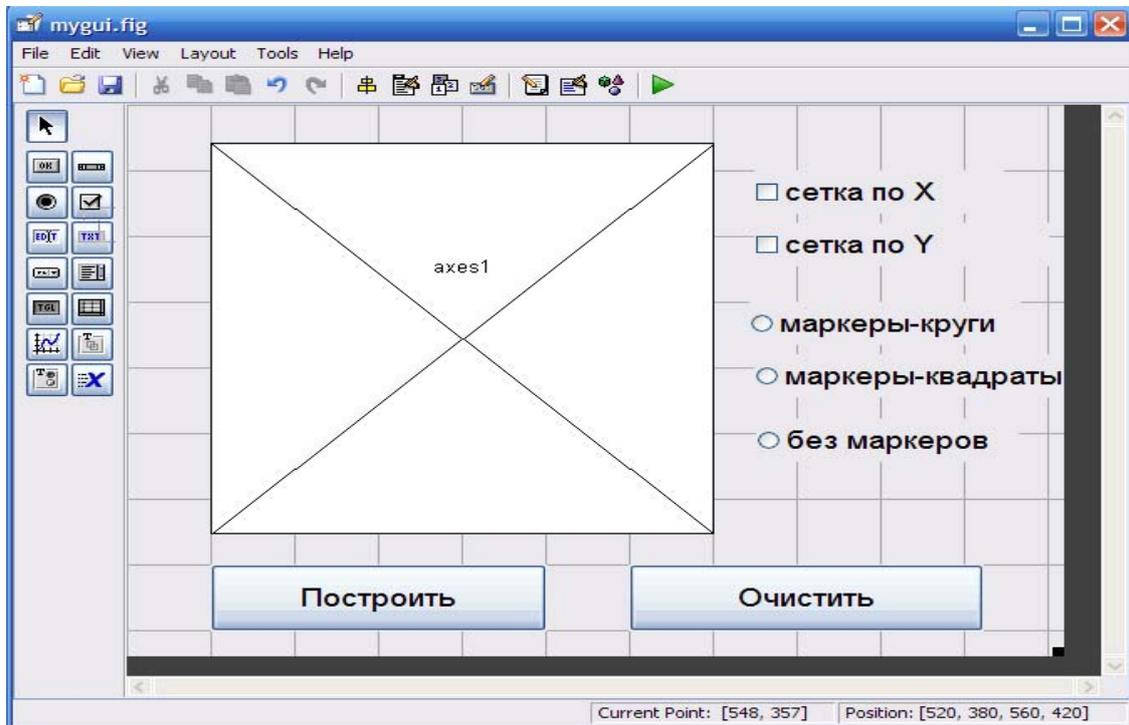


Рис. 19. Добавление переключателей

Состояние переключателя определяется свойством Value, если Value равно 1, то переключатель включен, нулю – выключен. Можно установить необходимые первоначальные значения этого свойства. Дальнейшее управление значением переключателей осуществляется программно. Если происходит переключение одного из radio button, то изменяется тип маркеров линии и выключаются два остальных переключателя. Изменение типа маркера происходит при обращении к свойству линии Marker. Указатель на линию возвращает plot в выходном аргументе.

```
function btnplot_Callback(hObject, eventdata, handles)
% hObject handle to btnplot (see GCBO)
% eventdata reserved-to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
x=[-2:0.2:2];
y=exp(-x.^2);
% запись указателя в поле line структуры handles
handles.line=plot(x,y);
% сохранения структуры handles для использования в других под-
функциях
```

```
guidata (gcbo, handles);
```

Обработка событий переключателя выглядит следующим образом.

```
function rbMarkCirc_Callback(hObject, eventdata, handles)
```

```
% hObject handle to rbMarkCirc (see GCBO)
```

```
% eventdata reserved-to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of rbMarkCirc
```

```
% выбран переключатель маркеры-круги
```

```
set(handles.line,'Marker','o') % размещение маркеров-кругов на линии
```

```
% установка остальных переключателей в положение выключено
```

```
set(handles.rbMarkSq, 'Value', 0)
```

```
set(handles.rbMarkNone, 'Value', 0)
```

```
% --- Executes on button press in rbMarkSq.
```

```
function rbMarkSq_Callback(hObject, eventdata, handles)
```

```
% hObject handle to rbMarkSq (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of rbMarkSq
```

```
% выбран переключатель маркеры-квадраты
```

```
set(handles.line,'Marker','s') % размещение маркеров-квадратов на ли-
```

нии

```
% установка остальных переключателей в положение выключено
```

```
set(handles.rbMarkCirc, 'Value', 0)
```

```
set(handles.rbMarkNone, 'Value', 0)
```

```
% --- Executes on button press in rbMarkNone.
```

```
function rbMarkNone_Callback(hObject, eventdata, handles)
```

```
% hObject handle to rbMarkNone (see GCBO)
```

```
% eventdata reserved-to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of rbMarkNone
```

```
% выбран переключатель без маркеров
```

```
set(handles.line,'Marker','none') % удаление маркеров с линии
```

```
% установка остальных переключателей в положение выключено
```

```
set(handles.rbMarkSq, 'Value', 0)
```

```
set(handles.rbMarkCirc, 'Value', 0)
```

Проверьте работоспособность приложения. Маркеры линии меняются, но имеются некоторые недостатки интерфейса. Во-первых, при измене-

нии маркеров на пустых осях MATLAB выдает ошибку обращения к несуществующему объекту. Во-вторых, нажатие на кнопку «Построить» приводит к построению графика без маркеров вне зависимости от установленного переключателя. И, наконец, повторный щелчок на переключатель приводит к его выключению, но всегда должен быть включен один из переключателей.

Первый недостаток устраняется при запрещении доступа к переключателям, если нет линии на графике и разрешении его после ее появления. Это делается в функциях обработки событий при нажатии на кнопки «Построить» и «Очистить». Нужно установить свойство Enable в нужное значение 'on' или 'off'. При запуске приложения все переключатели должны быть недоступны, так как пользователь еще не построил график функции.

Для того чтобы при построении графика тип маркера отвечал установленному переключателю, нужно при обработке события при нажатии на кнопку «Построить» включить опрос переключателей.

Для того чтобы всегда был активным один переключатель и при повторном щелчке он не выключался, нужно воспользоваться еще одним возможным значением свойства Enable – inactive. Переключатель со значением inactive выглядит в рабочем положении как доступный переключатель, но попытка его изменить не приводит к успеху. Необходимо проработать следующие изменения в программе:

1. Свойство Enable переключателя, событие которого обрабатывается, должно иметь значение inactive свойства Enable, а для остальных двух on.
2. При нажатии на кнопку «Построить» свойству Enable всех переключателей присваивается значение on, а затем определяется установленный в данный момент переключатель и в Enable заносится inactive.

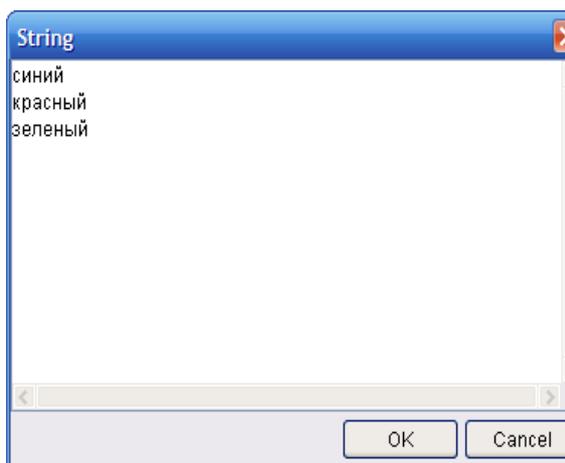


Рис. 20. Окно String

новленный в данный момент переключатель и в Enable заносится inactive.

Альтернативный способ выбора пользователем только одной из предлагаемых опций реализуется с помощью раскрывающегося списка.

Добавим в интерфейс возможность выбора цвета линии графика из раскрывающегося списка (синий, красный, зеленый). Перейдите в режим редактирования и добавьте при помощи панели управления раскры-

вающийся список. В свойстве Tag запишите pmColor, в свойстве String-синий, красный, зеленый (рис. 20).

Обработка события Callback раскрывающегося списка состоит в определении выбора пользователя и соответствующем изменении цвета линии. Свойство списка Value содержит номер выбранной строки. С помощью оператора Switch устанавливают цвет линии в зависимости от номера выбранной строки списка.

```
function pmColor_Callback(hObject, eventdata, handles)
% hObject   handle to pmColor (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns pmColor contents as cell
array
%         contents {get(hObject,'Value')} returns selected item from pmColor

% определение номера строки
N=get(hObject,'String');
switch N
    case 1
        % выбрана первая строка, делаем линию синей
        set(handles.line,'Color','b');
    case 2
        % выбрана вторая строка, делаем линию красной
        set(handles.line,'Color','r');
    case 3
        % выбрана третья строка, делаем линию зеленой
        set(handles.line,'Color','g');
end
```

Проверьте, меняется ли цвет линии графика. Но в таком виде интерфейс имеет ряд недостатков.

1. Повторное построение графика не учитывает текущий выбор цвета в раскрывающемся списке.
2. Выбор цвета на несуществующем графике приводит к ошибке.
3. Список нужно озаглавить.

Первый недостаток устраняется следующим образом. Необходимо добавить в функцию обработки события при нажатии на кнопку «Построить» опрос состояния раскрывающегося списка.

Второй недостаток устраняется, если запретить доступ пользователя к списку при отсутствии графика и разрешить при его появлении.

Третий недостаток устраняется с помощью добавления в интерфейс статического текста.

Теперь работающее приложение выглядит так (рис. 21).

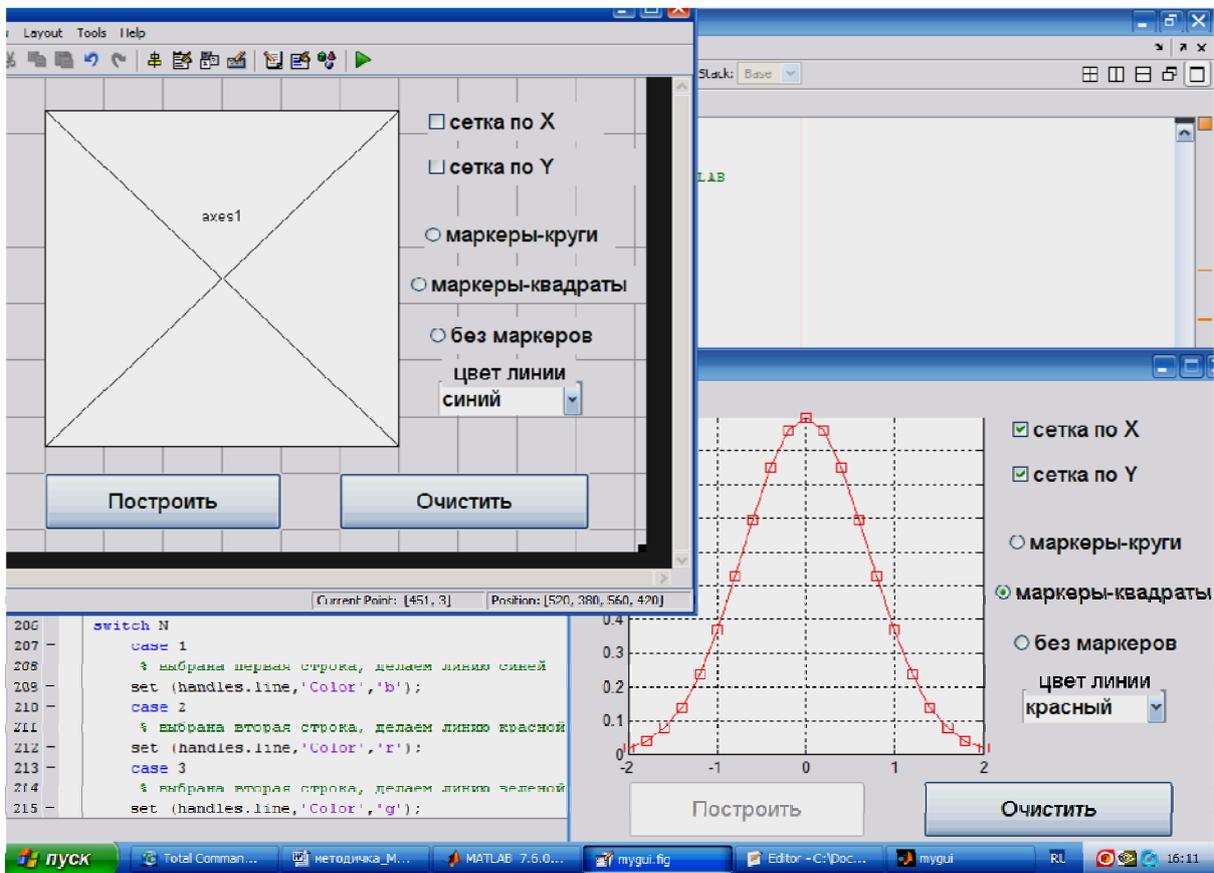


Рис. 21. Добавление раскрывающегося списка и статического текста

Используем полосу скроллинга для изменения толщины линии графика. Добавим ее на поле интерфейса и внесем изменения в свойство Tag – scrWidth. Добавим пояснительный текст Static Text – «Толщина линии». В инспектор свойств вносим изменения:

1. Максимальное значение max – 10, минимальное значение min-1. Определяются границы изменения Value при перемещении бегунка.
2. Начальное положение Value установить равным 1.
3. Свойство SliderStep определяет, насколько значение Value изменится при нажатии на кнопки со стрелками (первое значение) и при перетаскивании бегунка мышью (второе значение) (рис. 22).

Теперь нужно ввести обработку событий при обращении к скроллингу. Округленное значение свойства Value будет определять значение толщины линии.

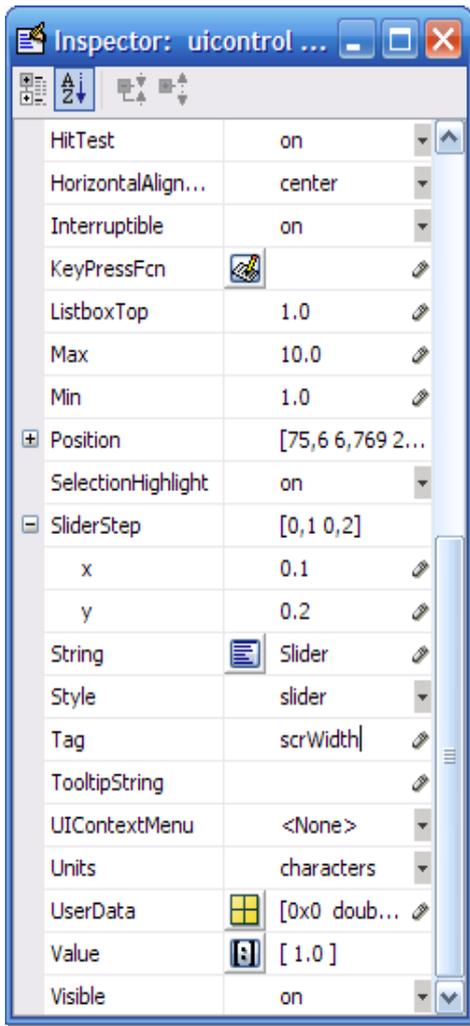


Рис. 22. Изменение свойств Slider

Изменение в тексте программы
`function scrWidth_Callbac(hObject,
eventdata, handles)`
`% hObject handle to scrWidth (see
GCBO)`
`% eventdata reserved-to be defined
in a future version of MATLAB`
`% handles structure with handles
and user data (see GUIDATA)`

`% Hints: get(hObject,'Value') re-
turns position of slider`
`% get(hObject,'Min') and
get(hObject,'Max') to determine range of
slider`

`% получение толщины линии в
зависимости от положения бегунка
скроллинга`

`w=get(hObject,'Value')`
`% установка толщины линии`
`set(handles.line,'LineWidth',
round(w))`

Теперь интерфейс будет выгля-
деть следующим образом (рис. 23).

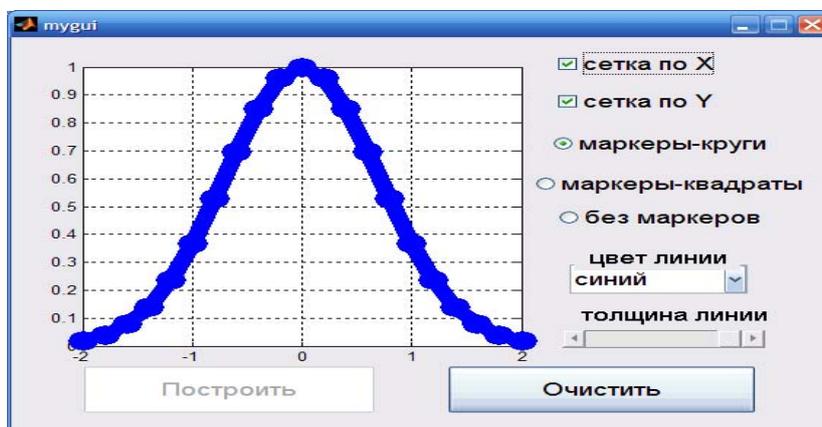


Рис. 23. Добавление в интерфейс Slider

Таблица 6. Описание свойств объектов

Свойства объектов Static Text		Свойства объектов Edit Text	
String	Tag	String	Tag
a=	text_a	a=	edit_a
b=	text_b	b=	edit_b
a+b=	text_a+b	a+b=	edit_a+b
a*b=	text_ab	a*b=	edit_ab

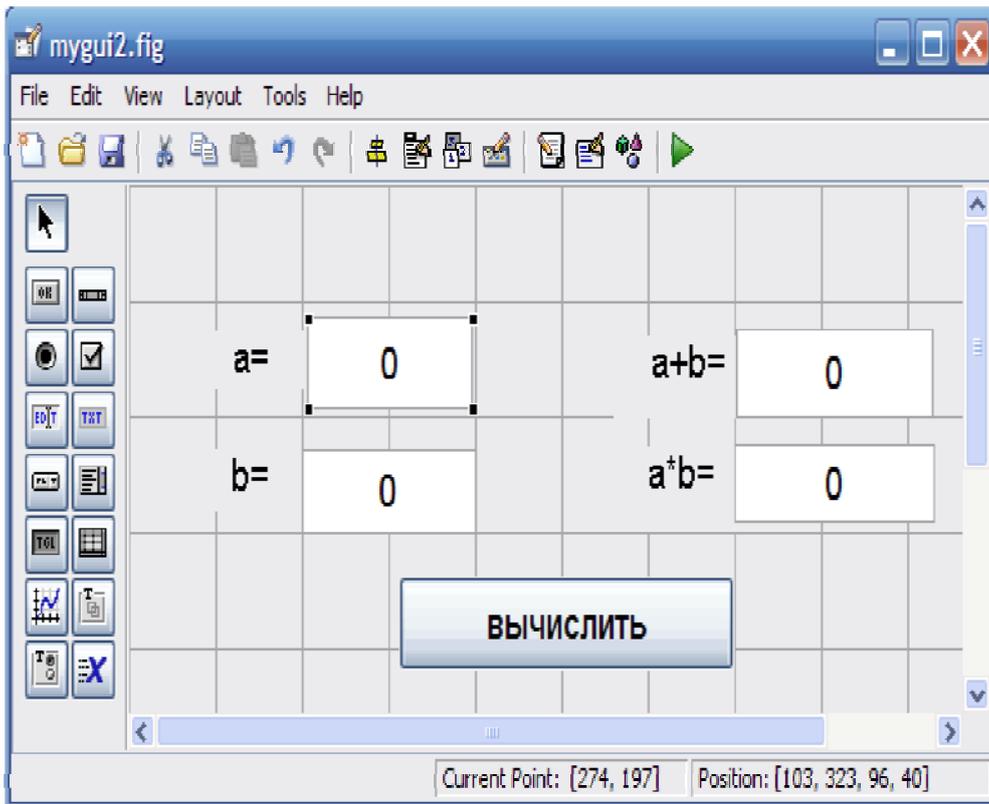


Рис. 24. Интерфейс с изменяющимся и статическим текстом

Рассмотрим пример интерфейса для ввода цифровых значений, их обработки и вывода результата.

Создадим интерфейс, содержащий 4 элемента Static Text, 4 элемента Edit Text и кнопку Push Button (рис. 24 и табл. 6)

Для обработки событий в программе необходимо произвести следующие изменения. Для использования в программе числа, вводимого в

поле `a=` , необходимо в функции `edit_a_Callback` добавить следующие строки:

```
function edit_a_Callback(hObject, eventdata, handles)
% hObject   handle to edit_a (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit_a as text
% str2double(get(hObject,'String')) returns contents of edit_a as a double
% преобразуем строку в вещественное число и присваиваем его пе-
ременной a
```

```
a=str2double(get(hObject,'String'));
% создаем указатель на переменную a
handles.a=a;
% сохраняем структуру handles для использования в других под-
функциях
```

```
guidata (gcbo, handles)
Аналогичные действия необходимо проделать в функции ed-
it_b_Callback
```

```
function edit_b_Callback(hObject, eventdata, handles)
% hObject   handle to edit_b (see GCBO)
% eventdata reserved-to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_b as text
% str2double(get(hObject,'String')) returns contents of edit_b as a double
b=str2double(get(hObject,'String'));
handles.b=b;
guidata (gcbo, handles)
```

Для обработки событий при нажатии на кнопку «Вычислить»:

```
function pbn_calculate_Callback(hObject, eventdata, handles)
% hObject   handle to pbn_calculate (see GCBO)
% eventdata reserved-to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
% вычисляем сумму a+b
c=handles.a+handles.b;
% запись суммы в поле a+b=
set(handles.edit_a_b, 'String', c)
% вычисляем произведение a*b
d=handles.a*handles.b;
% запись произведения в поле a*b=
```

set(handles.edit_ab, 'String', d)

Теперь интерфейс выглядит следующим образом (рис. 25).

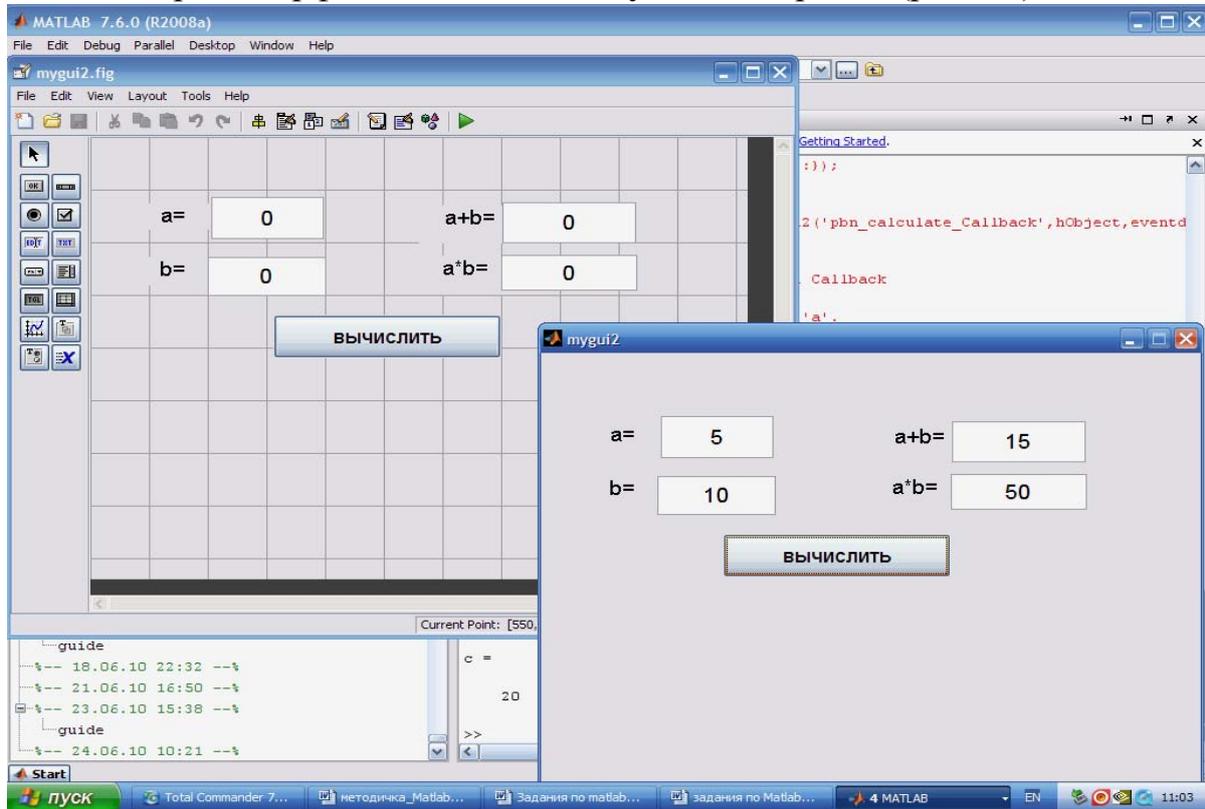


Рис. 25. Интерфейс для вычисления суммы и произведения

Задания к лабораторной работе № 5

Проектирование интерфейса пользователя в среде GUIDE.

Во всех заданиях разместить оси – Axes с размещением на них графического представления объекта с управлением включением сетки.

В некоторых заданиях наложить запрет на ввод нечисловых и отрицательных значений переменных.

Создать кнопку reset.

1. Составить алгоритм вычисления площади треугольника со сторонами a, b, c по формуле Герона и периметра.
2. Составить алгоритм вычисления площади ромба по диагоналям d_1, d_2 и периметра.
3. Составить алгоритм вычисления площади кольца радиусами r_1, r_2 .
4. Составить алгоритм вычисления площади квадрата по диагонали d и периметра.
5. Составить алгоритм вычисления расстояния между двумя точками на плоскости с координатами $A(x, y), B(a, b)$. Нарисовать окружность с центром в точке A и радиусом AB и посчитать площадь и периметр.

6. Составить алгоритм вычисления площади треугольника со сторонами a, b и углом между ними A и периметра.
7. Составить алгоритм вычисления площади трапеции со сторонами a, b и высотой h и периметра.
8. Составить алгоритм вычисления длины окружности и площади круга по его радиусу R .
9. Составить интерфейс для исследования пассивного ФНЧ. Вводим параметры R и C , вычисляем постоянную времени фильтра, строим АЧХ, ФЧХ (2 графика).
10. Составить интерфейс для исследования пассивного ФВЧ. Вводим параметры R и C , вычисляем постоянную времени фильтра, строим АЧХ, ФЧХ (2 графика).
11. Исследуем гармоническую модуляцию. Задаем параметры входного сигнала (2 окна ввода), на графике отражаем вид сигнала. На другом графике – вывод модулированного сигнала. Выбор модуляции из списка (амплитудная, частотная, фазовая).
12. Исследуем импульсную модуляцию. Задаем параметры входного сигнала (2 окна ввода), на графике отражаем вид сигнала. На другом графике – вывод модулированного сигнала. Выбор модуляции из списка (амплитудно-импульсная, частотно-импульсная, широтно-импульсная).
13. Составить интерфейс для исследования универсального вольтметра. Задаем параметры входного сигнала (2 окна ввода), на графике отражаем вид сигнала. Создаем окна вывода результата: 1) для действующего значения, 2) для максимального значения, 3) для среднего значения.
14. Создать интерфейс для расчета массы тела по введенным габаритам ($a \times b \times c$) – три окна ввода и по плотности (плотность выбирается из ниспадающего списка – 5 разных веществ).
15. Создать интерфейс для выбора и вывода на экран графиков синусоиды и косинусоиды с заданием амплитуды, периода, фазы.

Дополнительное задание

Разработать в среде GUIDE приложение для интерактивного управления следующими заданиями.

1. Используя датчик случайных чисел, сформировать массивы координат вершин выпуклого и произвольного полигонов.
2. Реализовать метод горизонтального луча для анализа принадлежности точки полигону.

3. Изобразить полигоны. Закрасить области внутри и вне полигона. Разными маркерами изобразить точки, принадлежащие и не принадлежащие полигону.

Дополнительные указания

1. Воспользуйтесь средой разработки приложения GUIDE и разместите на фигуре следующие элементы интерфейса: поле графика (координатная плоскость), две группирующие рамки и две полосы прокрутки.

2. В одной рамке разместите два окна редактирования и кнопку. Одно окно редактирования используйте для ввода граничных значений полярного радиуса и приращения полярного угла, второе окно – для ввода начальной точки (потребуется при формировании массивов вершин полигонов). Кнопку используйте для чтения содержимого окон редактирования в переменные.

3. В другой рамке разместите три переключателя, два индикатора неальтернативных вариантов и две простые кнопки. Переключатели используйте для определения составляющей цвета (Red, Green, Blue), интенсивность которой меняется с помощью полосы прокрутки. Одну кнопку используйте для построения изображения полигонов, другую – для вычисления координат случайной точки, анализа принадлежности точки полигону и для изображения точки. Индикаторы используйте для определения типа изображаемого полигона (выпуклый, произвольный).

4. Реализуйте в виде М-файлов алгоритмы формирования массивов вершин полигонов и анализа принадлежности точки полигону.

5. Заполните шаблоны функций обработчиков событий, формируемых средой GUIDE.

6. С помощью инспектора свойств интерфейсных элементов установите свойства элементов интерфейса (размеры, расположение, цвет, имя и другие свойства).

Содержание отчета

Отчет оформляется каждым студентом самостоятельно. Защита происходит в начале каждого следующего занятия с демонстрацией работы программы на ЭВМ. Студент, не подготовивший или не защитивший отчет по работе, к следующей лабораторной работе не допускается.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Вариант задания.

4. Текст программы в MATLAB и полученные графики.
5. Выводы по работе.

Контрольные вопросы и задания

1. Какие элементы управления можно использовать при создании графического интерфейса MATLAB?
2. Какие компоненты содержит редактор GUIDE?
3. Как организуется указатель на какой-либо объект в m-файле MATLAB?
4. Назначение и синтаксис функций set и get.
5. Как организовать в интерфейсном окне объекты ввода и вывода численных данных?

Лабораторная работа № 6

АППРОКСИМАЦИЯ И ИНТЕРПОЛЯЦИЯ ДАННЫХ В MATLAB

Цель работы: научиться работать с полиномами, аппроксимировать и интерполировать данные в *MATLAB*.

Оборудование: дисплейный класс, программа MATLAB R2008 или выше.

Аппроксимация и интерполяция часто используются при обработке экспериментальных данных. При этом точность и время расчета будут зависеть от выбранного вида интерполяции.

Операции с полиномами

Полиномы в MATLAB представляются в виде вектора коэффициентов. Задача интерполяции связана с построением полинома, наилучшим образом приближающим двух- и трехмерные табличные данные, полученные в ходе того или иного эксперимента.

Задаются полиномы с помощью матрицы коэффициентов.

Пример. Полиномы $p(x)=x^5+x^3+1$ и $q(x)=x^2+2x+3$.

```
>> p=[1 0 1 0 0 1]
```

```
p =  
    1    0    1    0    0    1
```

```
>> q=[1 2 3]
```

```
q =  
    1    2    3
```

Для умножения двух полиномов применяется операция conv.

```
>> s=conv(p,q)
```

```
s =
```

1 2 4 2 3 1 2 3

Деление полиномов осуществляется с помощью функции `deconv`. Для сложения и вычитания полиномов нужно использовать функцию `polysum(p,q)` и `polysum(p,-q)`. Только предварительно необходимо преобразовать полиномы до одинаковой длины. В MATLAB есть функция вычисления производной полиномов-`polyder`.

Интерполирование

Часто возникает задача представления табличных данных в виде приближенной функции полиномиальной или кусочно-полиномиальной. Самым простым способом интерполирования, предлагаемым в MATLAB, является приближение полиномом методом наименьших квадратов.

Построение полинома заданной степени, который приближает функцию одной переменной, заданную таблицей значений, производится при помощи функции `polyfit`.

Пример. Функция, заданная таблично. Необходимо ее приблизить полиномами четвертой и шестой степени.

x_i	0.1	0.2	0.4	0.5	0.6	0.8	1.2
y_i	-3.5	-4.8	-2.1	0.2	0.9	2.3	3.7

Сначала найдем коэффициенты приближающих полиномов нужной степени (функция `polyfit`), затем вычисляем и строим графики полученных функций (операция `polyval`).

```
function interpolation
x=[0.1 0.2 0.4 0.5 0.6 0.8 1.2];
y=[-3.5 -4.8 -2.1 0.2 0.9 2.3 3.7];
% строим график табличной функции
plot(x,y,'ko')
% вычисляем коэффициенты приближающих полиномов
p4=polyfit(x,y,4);
p6=polyfit(x,y,6);
% построение графиков полиномов
t=[0.1:0.01:1.2];
P4=polyval(p4,t);
P6=polyval(p6,t);
hold on
plot(t,P4,'k-',t,P6,'k-.')
```

Полученный график представлен на рис. 26.

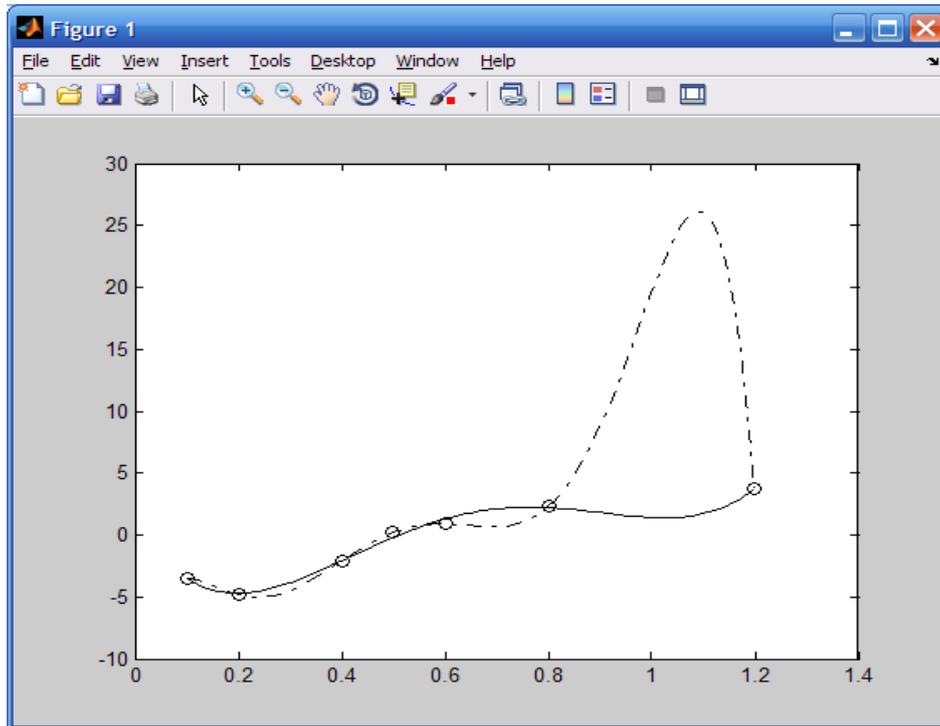


Рис. 26. Приближение полиномами по методу наименьших квадратов

Приближение методом наименьших квадратов дает хороший результат не всегда, в данном примере при увеличении степени полинома приближение ухудшается. Чаще для интерполяции таблично заданной функции применяют сплайны.

Интерполяция по соседним элементам приводит к приближению, при котором значение в каждой промежуточной точке принимается равным ближайшему значению, заданному в таблице. Линейная интерполяция приводит к соединению соседних точек отрезками прямых согласно табличным данным. Интерполяция кубическими сплайнами применяется для создания более гладкой функции.

Все эти способы реализуются встроенной функцией `interp1`, для которой указываются координаты абсцисс промежуточных точек, в которых вычисляются значения интерполянта и способ интерполирования.

‘nearest’ – интерполяция по соседним элементам;

‘linear’ – линейная интерполяция;

‘spline’ – интерполяция кубическими сплайнами;

function interpolation;

`x=[0.1 0.2 0.4 0.5 0.6 0.8 1.2];`

```

y=[-3.5 -4.8 -2.1 0.2 0.9 2.3 3.7];
% строим график табличной функции;
plot (x,y, 'ko');
% задаем промежуточные точки для интерполирования;
xi=[x(1):0.01:x(length(x))];
% вычисляем значения интерполирующих функций;
ynear=interp1(x,y,xi,'nearest');
yline=interp1(x,y,xi,'linear');
yspline=interp1(x,y,xi,'spline');
hold on;
plot (xi, ynear, 'k',xi, yline, 'k:', xi, yspline, 'k-.');
title ('Различные способы интерполяции функций');
legend ('табличная функция', 'по соседним элементам (nearest)',... ;
'линейная (linear)', 'кубические сплайны (spline)',4).

```

График с различными способами интерполяции представлен на рис. 27.

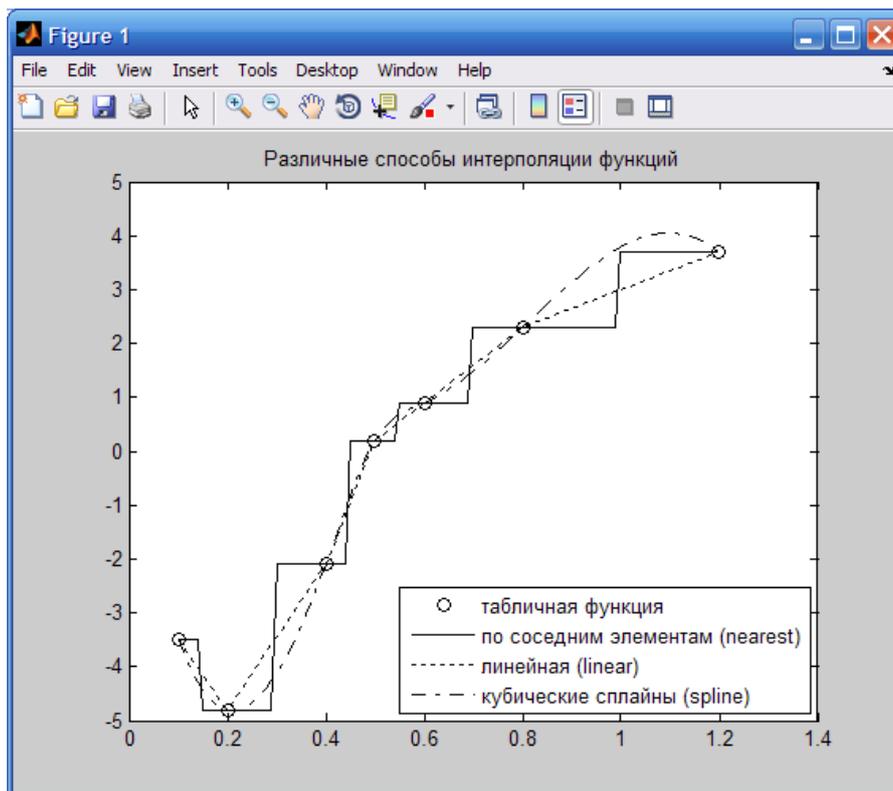


Рис. 27. Различные способы интерполяции функций

MATLAB позволяет интерполировать не только одномерные, но также и двумерные данные.

Содержание отчета

Отчет оформляется каждым студентом самостоятельно. Защита происходит в начале каждого следующего занятия с демонстрацией работы программы на ЭВМ. Студент, не подготовивший или не защитивший отчет по работе, к следующей лабораторной работе не допускается.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Вариант задания (таблица данных для аппроксимации выдается преподавателем).
4. Текст программы в MATLAB и полученные графики.
5. Выводы по работе.

Контрольные вопросы и задания

1. Каким образом задаются полиномы в MATLAB?
2. Для чего используется функция `polyfit`?
3. В чем отличие интерполяций ‘nearest’, ‘linear’ и ‘spline’?
4. Возможно ли выполнить в MATLAB интерполяцию двумерных данных?

Лабораторная работа № 7

МОДЕЛИРОВАНИЕ В ПРИЛОЖЕНИИ SIMULINK

Цель работы: научиться создавать и исследовать модели в приложении для имитационного моделирования Simulink.

Оборудование: дисплейный класс, программа MATLAB R2008 или выше.

Simulink – достаточно самостоятельный инструмент MATLAB, и при работе с ним совсем не требуется знать MATLAB и остальные его приложения. С другой стороны, доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в Simulink. Часть входящих в состав пакетов имеет инструменты, встраиваемые в Simulink, имеются также дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset – набор блоков для разработки цифровых устройств и т.д).

При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков.

Simulink запускается одним из трех способов:

- нажать кнопку (Simulink) на панели инструментов командного окна MATLAB;

- в командной строке главного окна MATLAB напечатать Simulink и нажать клавишу Enter на клавиатуре;

- выполнить команду Open в меню File и открыть файл модели (mdl - файл).

Первоначально открывается набор библиотек Simulink (рис. 28). В его состав входят:

1. Заголовок с названием окна – Simulink Library Browser.
2. Меню с командами File, Edit, View, Help.
3. Панель инструментов с ярлыками наиболее часто используемых команд.
4. Окно комментария для вывода поясняющего сообщения о выбранном блоке.
5. Список разделов библиотеки, реализованный в виде дерева.
6. Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков).

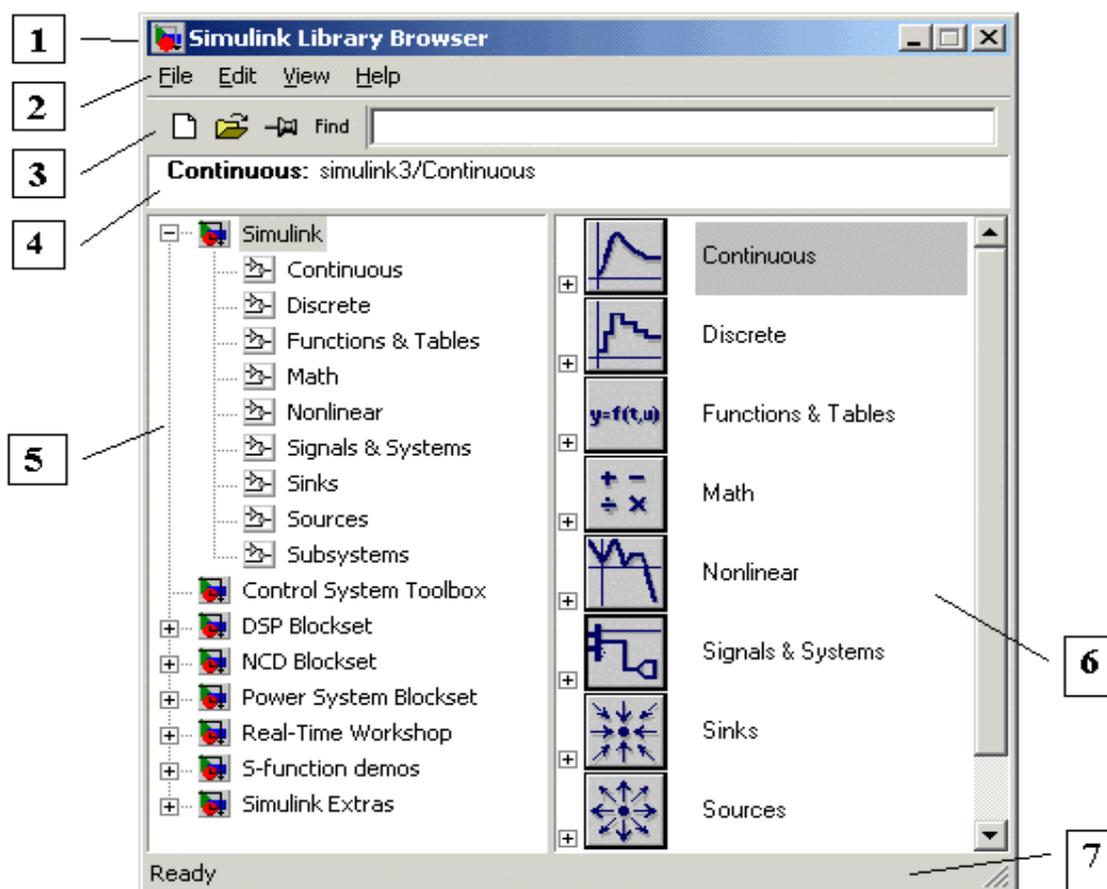


Рис. 28. Набор библиотек Simulink

7. Строка состояния, содержащая подсказку по выполняемому действию.

Библиотека Simulink содержит следующие основные разделы:

- **Continuous** – линейные блоки.
- **Discrete** – дискретные блоки.
- **Functions & Tables** – функции и таблицы.
- **Math** – блоки математических операций.
- **Nonlinear** – нелинейные блоки.
- **Signals & Systems** – сигналы и системы.
- **Sinks** - регистрирующие устройства.
- **Sources** — источники сигналов и воздействий.
- **Subsystems** – блоки подсистем.

Для работы с окном используются команды, собранные в меню. Меню обозревателя библиотек содержит следующие пункты:

- **File (Файл)** — работа с файлами библиотек.
- **Edit (Редактирование)** — добавление блоков и их поиск (по названию).
- **View (Вид)** — управление показом элементов интерфейса.
- **Help (Справка)** — вывод окна справки по обозревателю библиотек.

Кнопки панели инструментов (рис. 29) имеют следующее назначение:

1. Создать новую S-модель (открыть новое окно модели).
2. Открыть одну из существующих S-моделей.
3. Изменить свойства окна обозревателя. Данная кнопка позволяет установить режим отображения окна обозревателя ”*поверх всех окон*”. Повторное нажатие отменяет такой режим.
4. Поиск блока по названию (по первым символам названия). После того как блок будет найден, в окне обозревателя откроется соответствующий раздел библиотеки, а блок будет выделен. Если же блок с таким названием отсутствует, то в окне комментария будет выведено сообщение **Not found <имя блока>** (Блок не найден).



Рис. 29. Панель инструментов

Создание модели

Создать новый файл модели с помощью команды **File/New/Model** или используя кнопку на панели инструментов. Новый файл модели представлен на рис. 30.

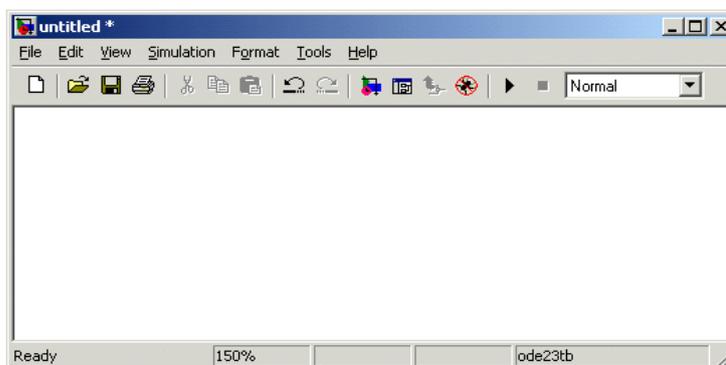


Рис. 30. Новый файл для создания модели

Для того чтобы расположить блоки в окне модели, необходимо от-

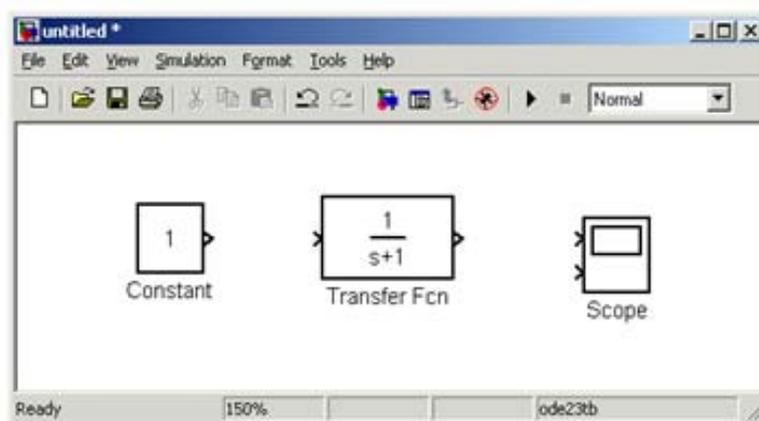


Рис. 31. Объекты Simulink

крыть соответствующий раздел библиотеки (например **Sources - Источники**), далее, указав курсором на требуемый блок и нажав на левую клавишу “мыши”, “перетащить” блок в созданное окно (рис. 31). *Клавишу мыши нужно держать нажатой.*

Если нужно изменить параметры блока, установленные программой “по умолчанию”, необходимо дважды

щелкнуть левой клавишей “мыши”, указав курсором на изображение блока. Откроется окно редактирования параметров данного блока (рис. 32). При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой **ОК**. На рис. 32 в качестве примера показаны блок, моделирующий передаточную функцию, и окно редактирования параметров данного блока.

После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы. Для соединения блоков необходимо указать курсором на “выход” блока, а затем нажать и, не отпуская левую клавишу “мыши”, провести линию к входу другого блока. После чего отпустить клавишу. В случае правильного соединения изображение стрелки на входе блока изменяет цвет. Для создания точки разветвления в соединительной линии нужно подвести курсор к предполагаемому узлу и нажать *правую* клавишу “мыши”. Для удаления линии требуется выбрать линию (так же, как это выполняется для блока), а затем нажать клавишу **Delete** на клавиатуре.

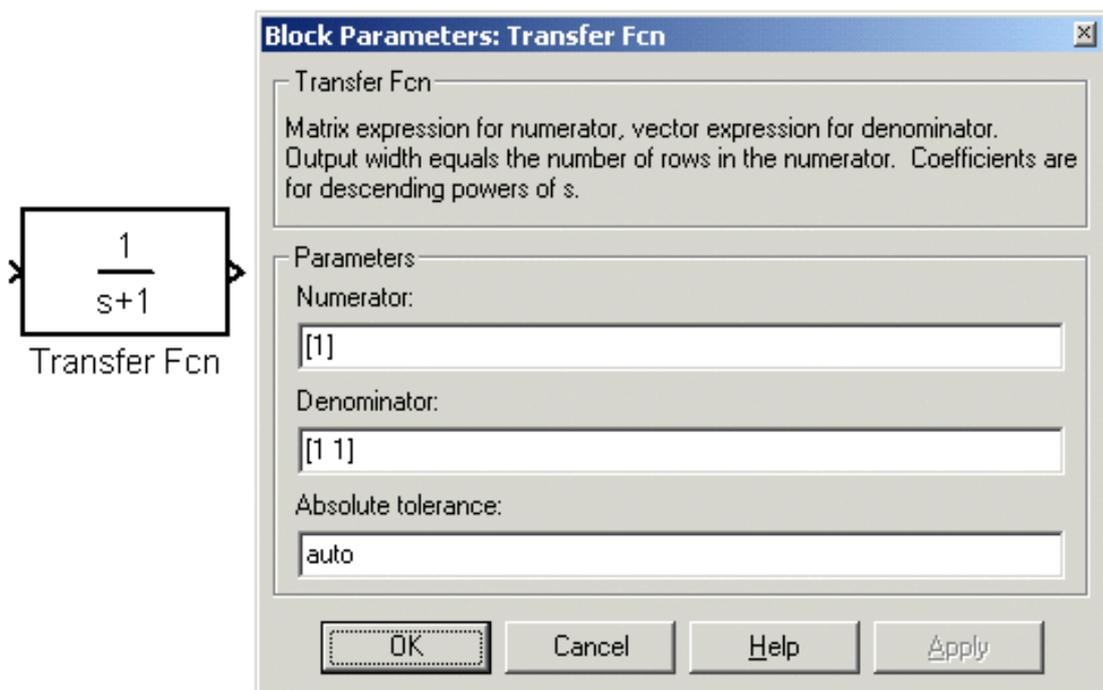


Рис. 32. Изменение параметров блока Simulink

Окно модели содержит следующие разделы (рис. 33).

1. Заголовок с названием окна. Вновь созданному окну присваивается имя **Untitled** с соответствующим номером.
2. Меню с командами **File**, **Edit**, **View** и т.д.

3. Панель инструментов.
4. Окно для создания схемы модели.
5. Строка состояния, содержащая информацию о текущем состоянии модели.

Меню окна содержит команды для редактирования модели, ее настройки и управления процессом расчета, работы файлами и т.п.:

File (Файл) — работа с файлами моделей.

Edit (Редактирование) — изменение модели и поиск блоков.

View (Вид) — управление показом элементов интерфейса.

Simulation (Моделирование) — задание настроек для моделирования и управление процессом расчета.

Format (Форматирование) — изменение внешнего вида блоков и модели в целом.

Tools (Инструментальные средства) — применение специальных средств для работы с моделью (отладчик, линейный анализ и т.п.).

Help (Справка) — вывод окон справочной системы.

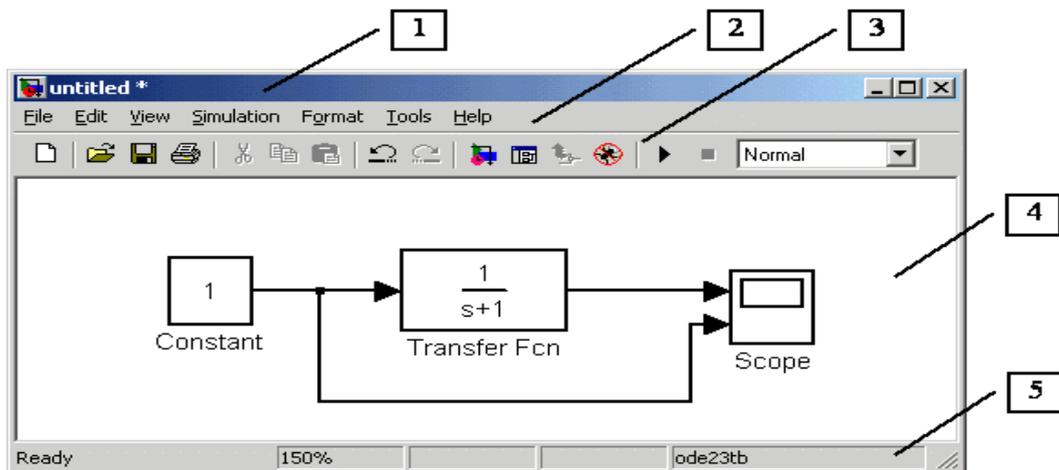


Рис. 33. Окно модели Simulink

Панель инструментов окна модели (рис. 34) содержит следующие пиктограммы:

1. **New Model** — открыть новое (пустое) окно модели.
2. **Open Model** — открыть существующий **mdl**-файл.
3. **Save Model** — сохранить **mdl**-файл на диске.
4. **Print Model** — вывод на печать блок-диаграммы модели.
5. **Cut** — вырезать выделенную часть модели в буфер промежуточного хранения.
6. **Copy** — скопировать выделенную часть модели в буфер промежуточного хранения.

7. **Paste** — вставить в окно модели содержимое буфера промежуточного хранения.
8. **Undo** — отменить предыдущую операцию редактирования.
9. **Redo** — восстановить результат отмененной операции редактирования.
10. **Library Browser** — открыть окно обозревателя библиотек.
11. **Toggle Model Browser** — открыть окно обозревателя модели.
12. **Go to parent system** — переход из подсистемы в систему высшего уровня иерархии (“родительскую систему”). Команда доступна, если открыта подсистема.
13. **Debug** — запуск отладчика модели.
14. **Start/Pause/Continue Simulation** — запуск модели на исполнение (команда **Start**);
15. **Stop** — закончить моделирование. Кнопка становится доступной после начала моделирования, а также после выполнения команды **Pause**.
16. **Normal/Accelerator** — обычный/ускоренный режим расчета.

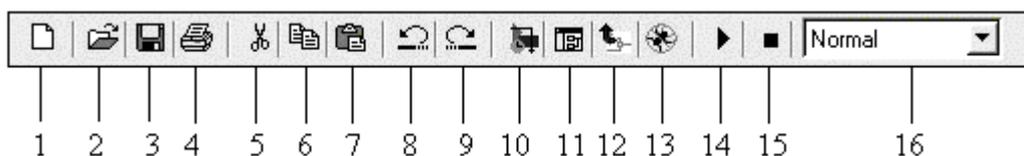


Рис. 34. Панель инструментов окна модели

Для повышения наглядности модели удобно использовать текстовые надписи. Для создания надписи нужно указать мышью место надписи и дважды щелкнуть левой клавишей мыши. После этого появится прямоугольная рамка с курсором ввода. Аналогичным образом можно изменить и подписи с блоками моделей. На рис. 35 показаны текстовая надпись и изменение надписи в блоке передаточной функции.

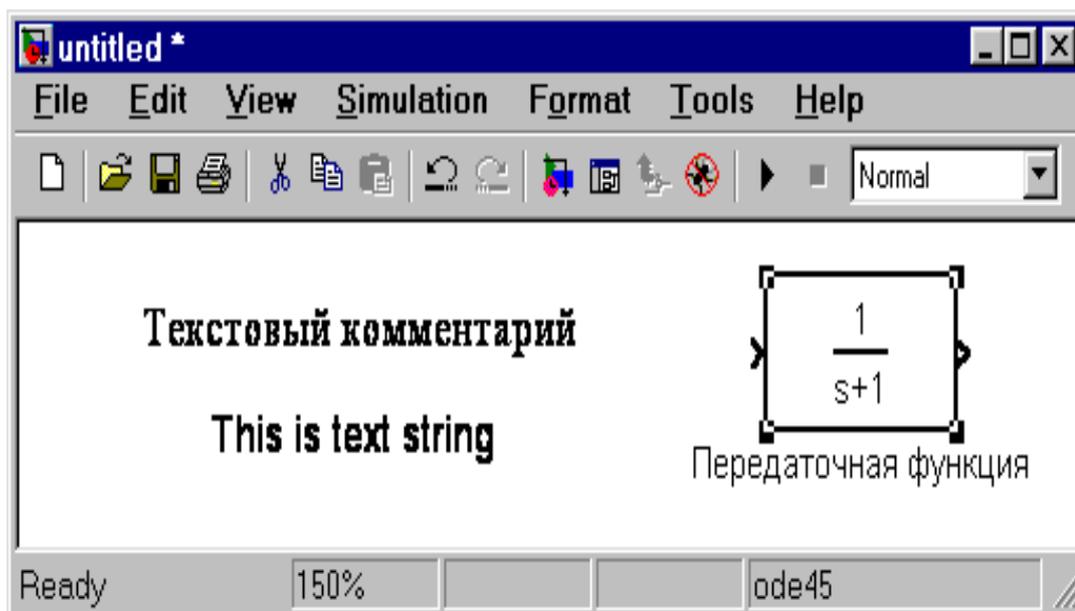


Рис. 35. Добавление надписи в модель

Для изменения размера блока он выделяется, после чего курсор мыши надо установить на один из маркеров по углам блока. После превращения курсора в двустороннюю стрелку необходимо нажать левую клавишу мыши и растянуть (или сжать) изображения блока. На рис. 36 показан этот процесс. Размеры надписей блока при этом не изменяются.

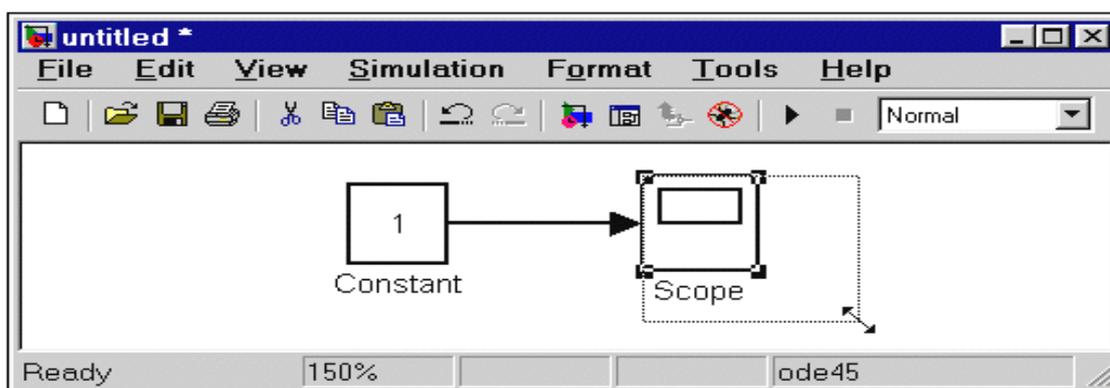


Рис. 36. Изменение размера объекта

Форматирование объектов

1. Изменение отображения надписей:

- **Font** — форматирование шрифта надписей и текстовых блоков.
- **Text alignment** — выравнивание текста в текстовых надписях.
- **Flip name** — перемещение подписи блока.

2. Изменение цветов отображения блоков:

- **Foreground color** — выбор цвета линий для выделенных блоков.

- **Background color** — выбор цвета фона выделенных блоков.
 - **Screen color** — выбор цвета фона для всего окна модели.
3. Изменение положения блока и его вида:
- **Flip block** – зеркальное отображение относительно вертикальной симметрии.
 - **Rotate block** – поворот блока на 90° по часовой стрелке.
 - **Show drop shadow** — показ тени от блока.
 - **Show port labels** — показ меток портов.
4. Прочие установки:
- **Library link display** — показ связей с библиотеками.
 - **Sample time colors** — выбор цвета блока индикации времени.
 - **Wide nonscalar lines** — увеличение/уменьшение ширины не скалярных линий.
 - **Signal dimensions** — показ размерности сигналов.
 - **Port data types** — показ данных о типе портов.

Установка параметров моделирования

Окно настройки параметров расчета (рис. 37) имеет 4 вкладки:

- **Solver (Расчет)** — установка параметров расчета модели.

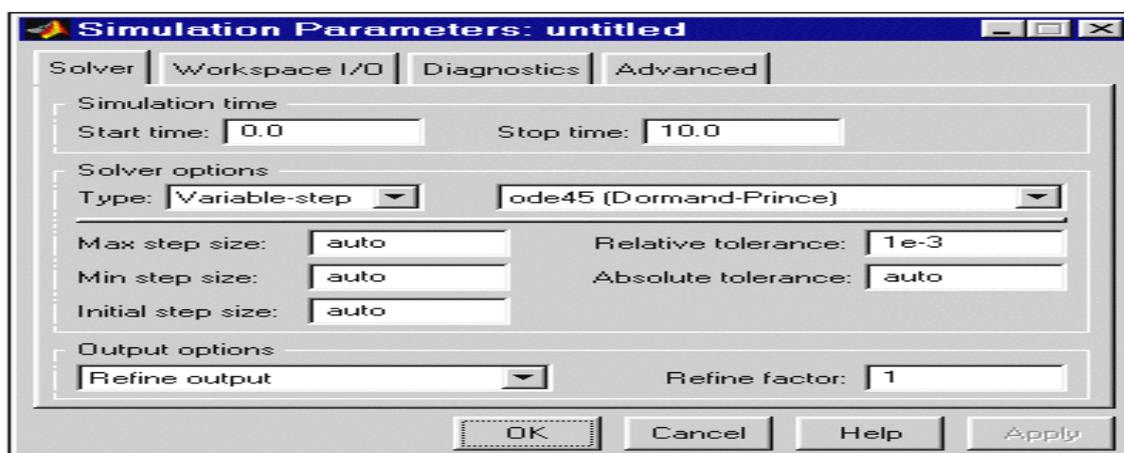


Рис. 37. Установка параметров моделирования

- **Workspace I/O (Ввод/вывод данных в рабочую область)** — установка параметров обмена данными с рабочей областью MATLAB.
- **Diagnostics (Диагностика)** — выбор параметров диагностического режима.
- **Advanced (Дополнительно)** — установка дополнительных параметров.

Установка параметров расчета выполняется с помощью элементов управления, размещенных на вкладке **Solver**. Они разделены на три груп-

пы: **Simulation time** (интервал моделирования), **Solver options** (параметры расчета), **Output options** (параметры вывода).

Элементы, позволяющие управлять вводом и выводом в рабочую область MATLAB промежуточных данных и результатов моделирования, расположены на вкладке **Workspace I/O**.

- **Load from workspace** (загрузить из рабочей области). Если флажок **Input** (входные данные) установлен, то в расположенном справа текстовом поле можно ввести формат данных, которые будут считываться из рабочей области MATLAB. Установка флажка **Initial State** (начальное состояние) позволяет ввести в связанном с ним текстовом поле имя переменной.
- **Save to workspace** (записать в рабочую область) – позволяет установить режим вывода значений сигналов в рабочую область MATLAB и задать их имена.
- **Save options** (параметры записи) – задает количество строк при передаче переменных в рабочую область. Если флажок **Limit rows to last** установлен, то в поле ввода можно указать количество передаваемых строк (строки считают от момента завершения расчета). Если флажок не установлен, то передаются все данные.

Моделирование датчика в Simulink (MATLAB)

Исходная модель приведена на рис. 38. Основная модель состоит из трех подсистем. Она позволяет моделировать выходной сигнал датчика Холла при изменении чувствительности, входного тока, угла между вектором индукции и площадью чувствительного элемента, отслеживать изменение выходного сигнала при перемещении магнита относительно датчика Холла на расстояние x и при соответствующем приложенном давлении (рис. 39 – 41).

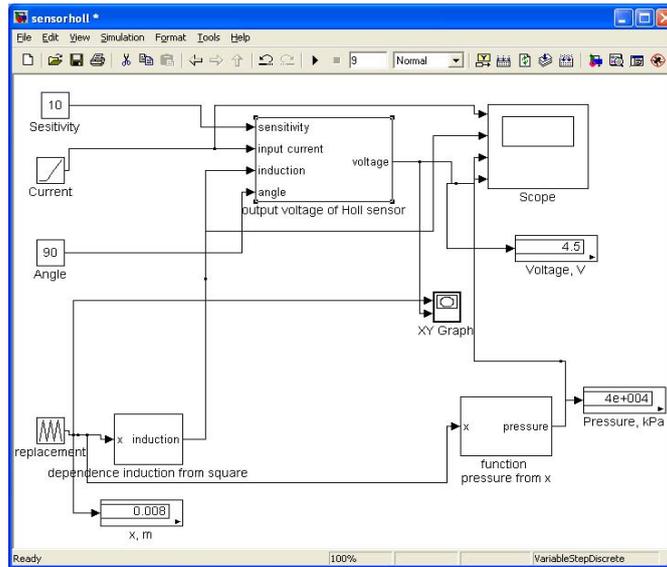


Рис. 38. Основная модель датчика Холла

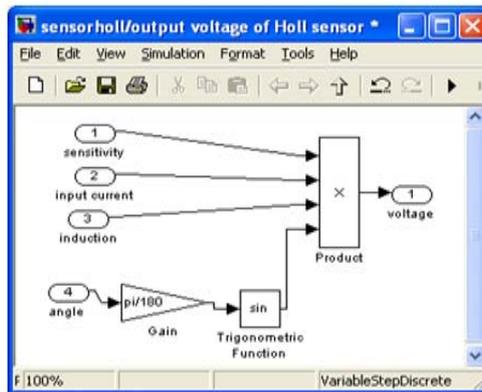


Рис. 39. Подсистема 1, вычисляющая напряжение Холла

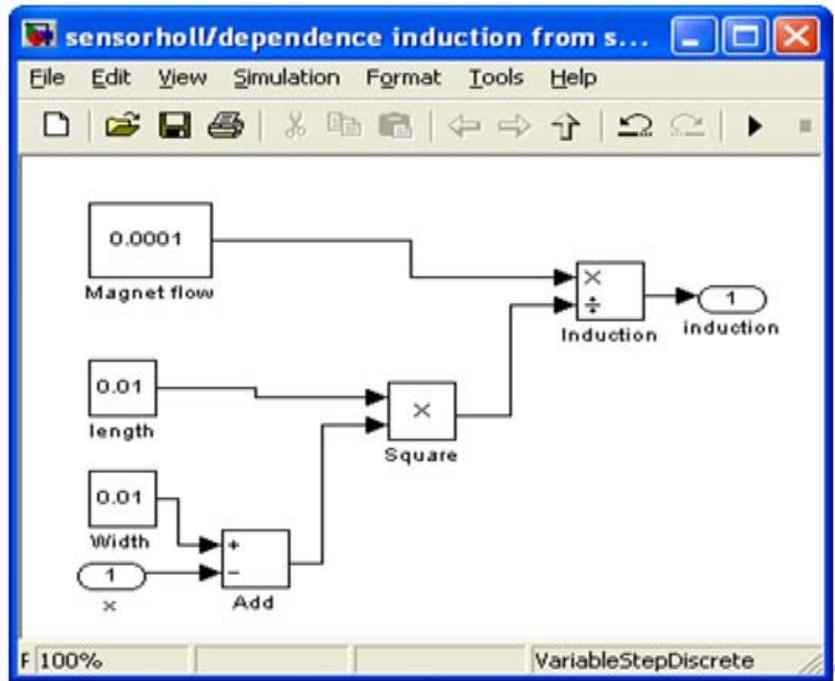


Рис. 40. Подсистема 2, вычисляющая индукцию магнитного поля, действующую на датчик Холла при перемещении магнита

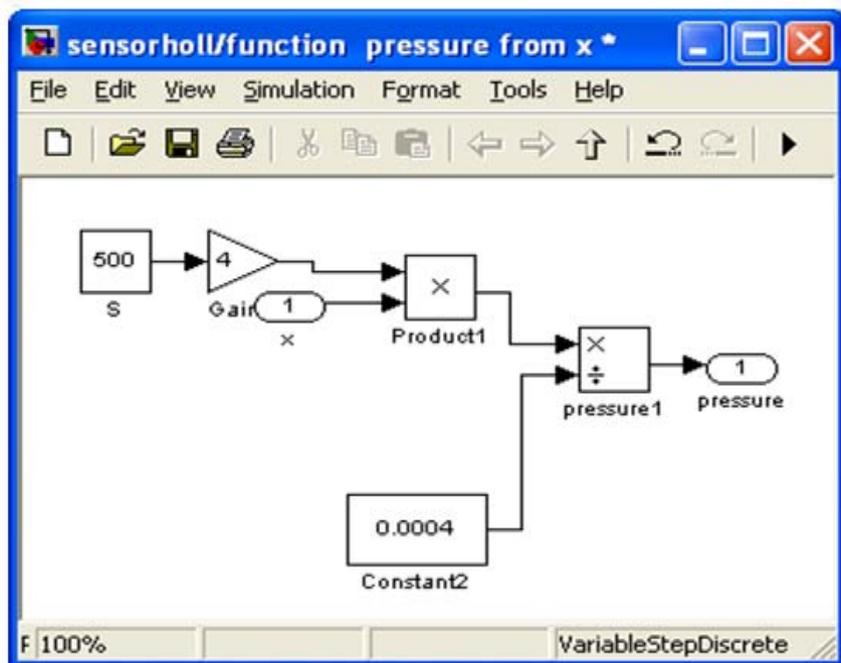
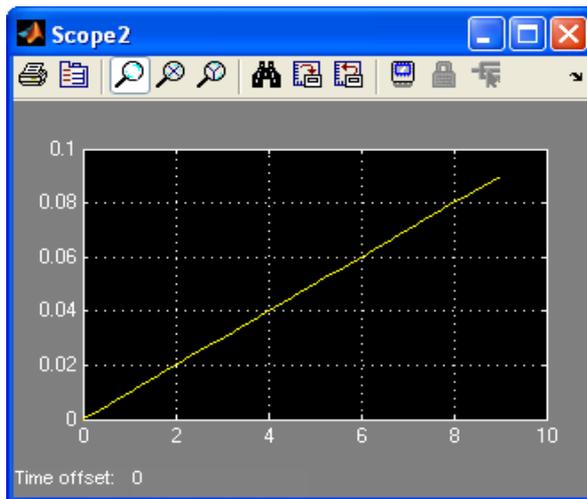
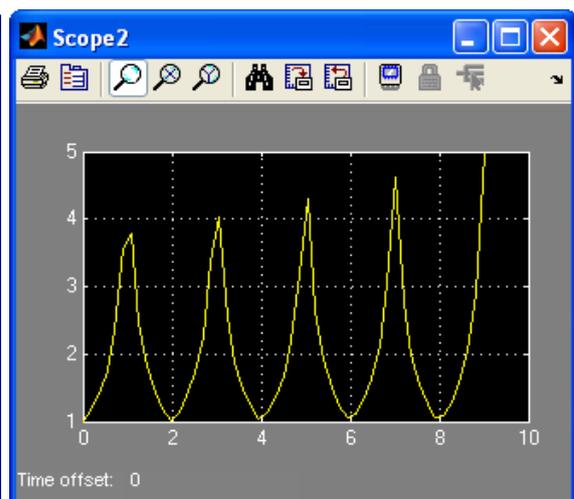


Рис. 41. Подсистема 3, вычисляющая давление, прикладываемое к магниту для перемещения его на расстояние x

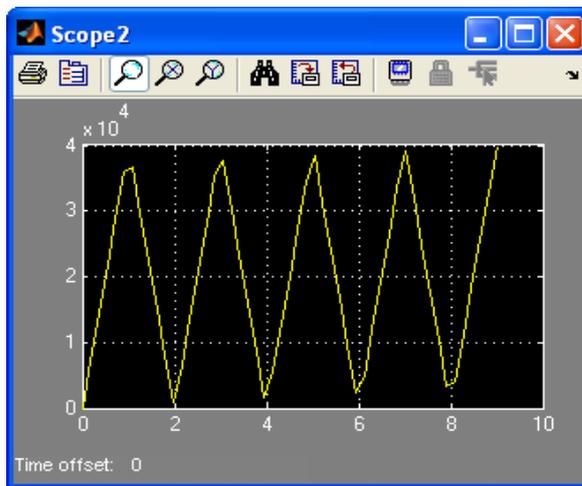
Результаты моделирования представлены на рис. 42.



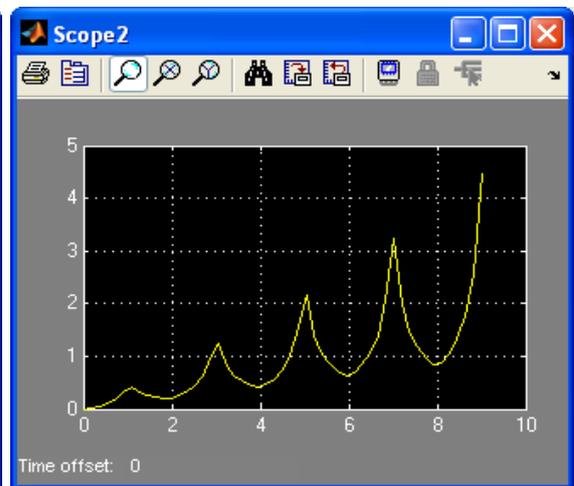
Изменение входного тока



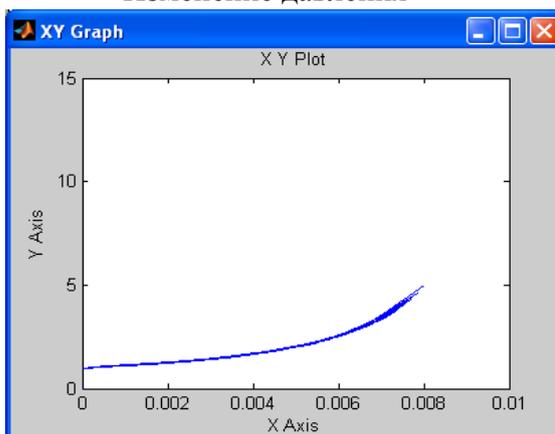
Изменение индукции в соответствии с изменением x



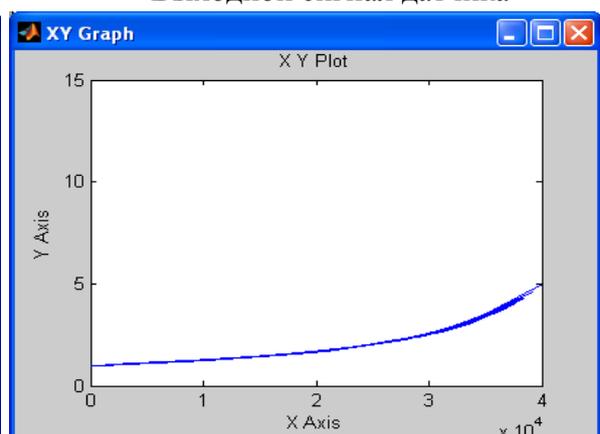
Изменение давления



Выходной сигнал датчика



Зависимость выходного напряжения (y) от смещения магнита (x)



Зависимость выходного напряжения от приложенного давления

Рис. 42. Результаты моделирования

Содержание отчета

Отчет оформляется каждым студентом самостоятельно. Защита происходит в начале каждого следующего занятия с демонстрацией работы программы на ЭВМ. Студент, не подготовивший или не защитивший отчет по работе, к следующей лабораторной работе не допускается.

Содержание отчета:

1. Титульный лист.
2. Цель работы.
3. Вариант задания (задание выдается преподавателем).
4. Модель в Simulink и полученные графики.
5. Выводы по работе.

Контрольные вопросы и задания

1. Для чего предназначено приложение Simulink в MATLAB?
2. Перечислите компоненты, входящие в набор библиотек Simulink?
3. Каким образом формируется модель в Simulink?
4. Расскажите, как изменить параметры блоков моделирования?
5. Как организуется подсистема в Simulink?

Список рекомендуемой литературы

1. *Ануфриев, И.* MatLab 7 : [наиболее полное руководство] / И. Ануфриев, А. Смирнов, Е. Смирнова. – СПб.: БХВ-Петербург, 2005. – 1090 с.– ISBN 5-94157-494-0.
2. *Лазарев, Ю.* Моделирование процессов и систем в MatLab : учеб. курс / Ю. Лазарев . – Спб : Питер, 2005. – 511 с. – ISBN 5-469-00600-X.
3. *Кетков, Ю. Л.* MatLab 7 : программирование, численные методы / Ю.Л. Кетков, А. Ю. Кетков, М. М. Шульц-СПб : БХВ – Петербург, 2005. – 737с. – ISBN 5-94157-347-2.
4. *Дьяконов, В. П.* MatLab 6.5 SP 1/7 + Simulink 5/6/. Основы применения / В. П. Дьяконов. – М. : Солон-Пресс, 2005. – 798 с. – ISBN 5-98003-181-2.
5. *Черных, И. В.* Simulink: среда создания инженерных приложений / И. В. Черных ; под. ред. В. Г. Потемкина. – М. : Диалог – МИФИ, 2004. – 491 с. – ISBN 5-86404-186-6.

ОГЛАВЛЕНИЕ

Введение.....	3
Лабораторная работа № 1. Простейшие вычисления в MATLAB.....	4
Лабораторная работа № 2. Работа с массивами в MATLAB.....	18
Лабораторная работа № 3. Основы программирования в MATLAB.....	25
Лабораторная работа № 4. Исследование функций на основе численных методов в MATLAB	37
Лабораторная работа № 5. Создание пользовательских интерфейсов в MATLAB	47
Лабораторная работа № 6. Аппроксимация и интерполяция данных в MATLAB.....	64
Лабораторная работа № 7. Моделирование в приложении Simulink.....	68
Список рекомендуемой литературы.....	81

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СРЕДСТВ ИЗМЕРЕНИЙ
В СРЕДЕ MATLAB

Методические указания к лабораторным работам

Составитель

Макарова Наталья Юрьевна

Ответственный за выпуск – зав. кафедрой профессор В. П. Легаев

Подписано в печать 12.11.11.

Формат 60x84/16. Усл. печ. л. 4,88. Тираж 60 экз.

Заказ

Издательство

Владимирского государственного университета.
600000, Владимир, ул. Горького, 87.