

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И УПРАВЛЕНИЯ В ТЕХНИЧЕСКИХ СИСТЕМАХ

Кафедра информатики и управления в технических системах

ЦИФРОВОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ И СИСТЕМ В MATLAB

Методические указания

к выполнению практических работ по дисциплине
«Цифровое моделирование процессов и систем в MATLAB»

В двух частях

Часть 1

Основы программирования и визуализации данных



Севастополь
СевГУ
2022

УДК 519.67(076.5)
ББК 22.18я73
Ц751

Р е ц е н з е н т :

А. А. Кабанов – канд. техн. наук, доцент,
заведующий кафедрой "Информатика и управление в технических системах"

Составители: В. А. Карапетьян, В. А. Крамарь, В. Н. Мирянова

Ц751 Цифровое моделирование процессов и систем в MATLAB :
методические указания к выполнению практических работ по дисциплине
«Цифровое моделирование процессов и систем в MATLAB». – В 2-х
частях : Ч. 1. Основы программирования и визуализации данных /
Севастопольский государственный университет, Институт
информационных технологий и управления в технических системах,
кафедра Информатики и управления в технических системах ; сост. :
В. А. Карапетьян, В. А. Крамарь, В. Н. Мирянова. – Севастополь : СевГУ,
2022. – 24 с. – Текст : электронный.

Целью методических указаний является оказание помощи обучающимся при выполнении практических работ для формирования умений и приобретения навыков при решении задач в системе MATLAB.

Методические указания предназначены для студентов всех направлений, обучающихся дополнительных профессиональных программ, всех, кто интересуется системой MATLAB.

УДК 519.67(076.5)
ББК 22.18я73

Методические указания рассмотрены и утверждены на заседании кафедры "Информатика и управления в технических системах", протокол № 7 от 23 марта 2022 г.

Методические указания рассмотрены и рекомендованы к изданию на заседании ученого совета Института информационных технологий и управления в технических системах, протокол № 6 от 18 апреля 2022 г.

© Карапетьян В.А., Крамарь В.А.,
Мирянова В.Н., сост., 2022
© ФГАОУ ВО «Севастопольский
государственный университет», 2022

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ	4
2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	4
2.1. Простейшие операции в MATLAB	4
2.1.1. Арифметические операции	4
2.1.2. Форматы данных	4
2.2. Элементарные встроенные математические функции	5
2.3. Имена переменных	6
2.4. Некоторые команды общего назначения	6
3. МАТРИЦЫ	6
3.1. Представление матриц в MATLAB	6
3.2. Действия с матрицами	7
3.3. Некоторые функции формирования матриц. Простейшие операции над матрицами	7
3.4. Собственные числа и собственные векторы матрицы	8
4. МНОГОЧЛЕНЫ	8
5. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ	9
6. СОЗДАНИЕ m-ФАЙЛОВ	10
7. ПРОГРАММИРОВАНИЕ В MATLAB	11
7.1. Отношения и логические операции	11
7.2. Операторы управления вычислительным процессом	12
7.2.1. Условный оператор <i>if</i>	12
7.2.2. Оператор цикла <i>while</i>	13
7.2.3. Оператор цикла <i>for</i>	14
8. ПОСТРОЕНИЕ ГРАФИКОВ	15
8.1. 2-D графики	15
8.2. 3-D графики	16
9. СИМВОЛЬНАЯ МАТЕМАТИКА	19
9.1. Решение уравнений	19
9.2. Символьные вычисления	20
10. ЗАДАНИЕ НА ВЫПОЛНЕНИЕ РАБОТЫ	20
11. СОДЕРЖАНИЕ ОТЧЕТА И ПОРЯДОК ЗАЩИТЫ РАБОТЫ	23
12. КОНТРОЛЬНЫЕ ВОПРОСЫ	23
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	24

1. ЦЕЛЬ РАБОТЫ

Целью работы является ознакомление с системой MATLAB и получение навыков решения с помощью MATLAB математических задач различного типа.

2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

MATLAB (аббревиатура от MATrix LABoratory) это программа для численных вычислений и отображения данных. Рассмотрим, основные команды для организации вычислений в MATLAB.

2.1. Простейшие операции в MATLAB

2.1.1. Арифметические операции

Арифметические операции в MATLAB над скалярами приведены в таблице 2.1.

Таблица 2.1 – Арифметические операции в MATLAB

Арифметическая операция	Символ	Пример
Сложение	+	$6+3=9$
Вычитание	-	$6-3=3$
Умножение	*	$6*3=18$
Правое деление	/	$6/3=2$
Левое деление	\	$6\backslash 3=3/6=1/2$
Возведение в степень	^	$6^3=216$

2.1.2. Форматы данных

MATLAB имеет несколько форматов для отображения данных. Некоторые, основные форматы приведены в таблице 2.2.

Таблица 2.2 – Основные форматы

Тип данных	Описание	Пример
format short	4 знака после запятой	>> 351/7 ans=50.1429
format long	14 знаков после запятой	>> 351/7 ans =50.14285714285715
format short e	Научный формат с 4 знаками после запятой	>> 351/7 ans=5.0143e+001
format long e	Научный формат с 15 знаками после запятой	>> 351/7 ans=5.014285714285715e+001

2.2. Элементарные встроенные математические функции

Таблица 2.3 – Общие математические функции

abs(x)	Вычисляет модуль (абсолютная величина) числа x
sqrt(x)	Вычисляет квадратный корень числа x
round(x)	Вычисляет округление числа x до ближайшего целого
fix(x)	Отбрасывает дробную часть числа x
floor(x)	Округляет число x до целого в меньшую сторону
ceil(x)	Округляет число x до целого в большую сторону
sign(x)	Вычисляет знак числа x
rem(x,y)	Вычисляет остаток от деления x/y . Например, $\text{rem}(25,4)$ даст ответ 1.
exp(x)	Вычисляет e^x
log(x)	Вычисляет натуральный логарифм числа x $\ln x$
log2	Вычисляет двоичный логарифм числа x
log10	Вычисляет десятичный логарифм числа x

Таблица 2.4 – Тригонометрические функции

cos(x)	Вычисляет косинус x
cot(x)	Вычисляет котангенс x
csc(x)	Вычисляет косеканс x
sec(x)	Вычисляет секанс x
sin(x)	Вычисляет синус x
tan(x)	Вычисляет тангенс x
acos(x)	Вычисляет арккосинус x
acot(x)	Вычисляет арккотангенс x
acsc(x)	Вычисляет арккосеканс x
asec(x)	Вычисляет арксеканс x
asin(x)	Вычисляет арксинус x
atan(x)	Вычисляет арктангенс x
cosh(x)	Вычисляет гиперболический косинус x
coth(x)	Вычисляет гиперболический котангенс x
csch(x)	Вычисляет гиперболический косеканс x
sech(x)	Вычисляет гиперболический секанс x
sinh(x)	Вычисляет гиперболический синус x
tanh(x)	Вычисляет гиперболический тангенс x
acosh(x)	Вычисляет гиперболический арккосинус x
acoth(x)	Вычисляет гиперболический арккотангенс x
acsch(x)	Вычисляет гиперболический арккосеканс x
asech(x)	Вычисляет гиперболический арксеканс x
asinh(x)	Вычисляет гиперболический арксинус x
atanh(x)	Вычисляет гиперболический арктангенс x

Таблица 2.5 – Функции для работы с комплексными числами

Функция	Описание
conj(x)	Вычисляет комплексно сопряжённое число для x . Если $x = a + ib$, то conj(x) даст $a - ib$.
angle(x)	Вычисляет фазу числа x . Значение лежит в интервале от $-\pi$ до π .
real(x)	Вычисляет вещественную часть x .
imag(x)	Вычисляет мнимую часть x .
abs(x)	Вычисляет модуль числа x .

2.3. Имена переменных

Имена переменных в MATLAB состоят из латинских букв или сочетания латинских букв и цифр. Имя переменной не должно начинаться с цифры и содержать более 63 символов. Необходимо отметить, что имена переменных чувствительны к регистру, т.е. XX , Xx , xX , xx – имена четырёх различных переменных. MATLAB содержит, так называемые, предопределённые переменные, такие как число π – “pi”, $\sqrt{-1}$ – “i” или “j” и другие.

По умолчанию любая команда в системе MATLAB приводит к отображению результата в командном окне. Чтобы результат не выводился, достаточно завершить ввод команды точкой с запятой.

2.4. Некоторые команды общего назначения

Команды	Описание
quit	завершает работу в системе MATLAB;
who	выводит список текущих переменных из рабочей области памяти;
clear	производит удаление переменных и функций из памяти;
clc	очищает рабочую область экрана;
disp <символьная строка>	осуществляет вывод строки в командном окне;
pause (<длительность паузы в секундах>)	при отсутствии параметра ожидает нажатия клавиши;

3. МАТРИЦЫ

3.1. Представление матриц в MATLAB

Матрица содержит числа или функции в строках или столбцах. Одномерная матрица описывает вектор-строку или вектор-столбец.

В векторе-строке элементы записываются в квадратных скобках, разделённые пробелами или запятыми.

Например, $x = [7 - 1 \ 2 - 5 \ 8]$.

В векторе-столбце элементы также записываются в квадратных скобках, разделённые символом точка с запятой (;).

Например, $x = [7; -1; 2; -5; 8]$.

Элементы матрицы разделяются пробелами (в строке) и точкой с запятой (в столбце)

Например, матрица $A = \begin{bmatrix} 1 & 3 & -4 \\ 0 & -2 & 8 \end{bmatrix}$, записывается в MATLAB

`>> A=[1 3 -4; 0 -2 8].`

Матрица $B = \begin{bmatrix} -5x & \ln 2x + 7 \sin 3y \\ 3i & 5 - 13i \end{bmatrix}$, записывается в MATLAB

`>> B=[-5*x log(2*x)+7*sin(3*y); 3i 5-13i].`

3.2. Действия с матрицами

В таблице 3.1. приведены основные команды, позволяющие выделять элементы из матрицы. Элемент матрицы A, находящийся на пересечении i-той строки и j-того столбца, обозначается A(i,j).

Таблица 3.1 – Команды выделения элементов

Команда	Описание
A(:,n)	Возвращает все элементы из столбца n матрицы A
A(n,:)	Возвращает все элементы из строки n матрицы A
A(:,m:n)	Возвращает элементы из всех строк между столбцами m и n матрицы A
A(m:n,:)	Возвращает элементы из всех столбцов между строками m и n матрицы A
A(m:n,p:q)	Возвращает элементы из строк от m до n и столбцов от p до q матрицы A

3.3 Некоторые функции формирования матриц. Простейшие операции над матрицами

Таблица 3.2 – Некоторые команды формирования матриц

zeros(m,n)	Создаёт матрицу m×n с нулевыми элементами
ones(m,n)	Создаёт матрицу m×n с единичными элементами.
eye(m,n)	Создаёт матрицу m×n с единичными элементами по главной диагонали и остальными нулевыми элементами.
inv(A)	Вычисляет матрицу, обратную матрице A.
A'	Вычисляет матрицу A^T .
eig(A)	Вычисляет массив собственных чисел матрицы A, т.е. корни характеристического полинома $\det(sI - A)$.
[A B]	Горизонтальная конкатенация (объединение) матриц A и B.

Операция транспонирования матрицы записывается в MATLAB

`>> B=A'`

Операция произведения двух матриц соответствующих размерностей (число столбцов первой матрицы должно быть равно числу строк второй матрицы) записывается в MATLAB

```
>>C=A*B
```

Операция обращения матрицы (матрица должна быть квадратной) записывается в MATLAB

```
>>E=inv(D)
```

Например, заданы две матрицы

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 4 \\ -1 & 6 & 7 \end{bmatrix}, \quad B = \begin{bmatrix} 7 & 4 & 2 \\ 3 & 5 & 6 \\ -1 & 2 & 1 \end{bmatrix}.$$

Необходимо выполнить операцию $A^2 + B^2 - (AB)^T$.

```
>>A=[ 1 0 1; 2 3 4; -1 6 7]
```

```
>>B=[ 7 4 2; 3 5 6; -1 2 1]
```

```
>>K=A^2+B^2-(A*B)'
```

3.4 Собственные числа и собственные векторы матрицы

Рассмотрим следующее уравнение

$$AX = \lambda X \quad (3.1)$$

где, A - $n \times n$ квадратная матрица, X - вектор столбец, соответствующей размерности и λ - скаляр.

Значения λ , для которых X является ненулевым, называется собственными числами матрицы A , а соответствующие значения X называются собственными векторами матрицы A .

Уравнение (3.1) может быть представлено в виде

$$(A - \lambda I)X = 0, \quad (3.2)$$

где I - $n \times n$ единичная матрица. Уравнение (3.2) имеет нетривиальное решение, только, если определитель равен нулю:

$$|A - \lambda I| = 0. \quad (3.3)$$

Уравнение (3.3) называется характеристическим уравнением матрицы A . Решение уравнения (3.3) даёт собственные числа матрицы A .

MATLAB позволяет вычислить и собственные числа и собственные векторы для заданной матрицы A .

eig(A) - вычисляет вектор столбец, содержащий собственные числа матрицы A .

[Q,d] = eig(A) - вычисляет квадратную матрицу Q , столбцами которой являются собственные векторы матрицы A и квадратную матрицу d , содержащую собственные числа матрицы A на главной диагонали.

4. МНОГОЧЛЕНЫ

Многочлен – это функция одной переменной, которая может быть записана в следующем виде

$$f(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_{n-1} x + a_n.$$

В MATLAB векторы коэффициентов представляют многочлены. Например, рассмотрим многочлен $5s^5 + 7s^4 + 2s^2 - 6s + 10$. Для представления его в MATLAB, мы введём вектор

```
>>x=[ 5 7 0 2 -6 10]
```

Функция **roots(a)** – определяет корни многочлена, коэффициенты которого заданы в векторе a.

```
>> roots ([5 7 0 2 -6 10])
```

или

```
>> x=[5 7 0 2 -6 10]
```

```
>>r=roots(x)
```

Функция **poly(r)** – восстанавливает многочлен по его известным корням, которые определены в векторе r.

Значение многочлена в заданной точке может быть определено при помощи функции **polyval(a,x)**, где a – вектор коэффициентов многочлена, а x – заданная точка. Например, для нахождения значения многочлена $5s^5 + 7s^4 + 2s^2 - 6s + 10$ в точке $s=2$, выполним следующую команду

```
>> x=polyval ([5 7 0 2 -6 10], 2)
```

Для перемножения двух многочленов используем команду **conv(a,b)**. Перемножим многочлены $x_1 = 2x + 5$ и $x_2 = x^2 + 3x + 7$.

```
>>x1=[2 5];
```

```
>>x2=[1 3 7];
```

```
>>z=conv(x,y)
```

Для деления двух многочленов используется команда **deconv(a,b)**.

5. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Систему линейных алгебраических уравнений можно представить в матричном виде $Ax = B$, где A – невырожденная матрица, т.е. её определитель не должен быть равен нулю, B – вектор правых частей.

Решение матричного уравнения $Ax = B$ может быть получено в следующем виде:

$$A^{-1} Ax = A^{-1} B.$$

$$Ex = A^{-1} B,$$

где E (в других обозначениях I) – единичная матрица.

Тогда, окончательно получаем

$$x = A^{-1} B.$$

В MATLAB указанное вычисление будет реализовано в следующем виде

```
>> x=inv(A)*B
```

Другим методом решения указанного матричного уравнения является матричное деление $x = A \setminus B$

>>x=A\B

Например, решим следующую систему уравнений

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 5x_4 = 21 \\ -2x_1 + 5x_2 + 7x_3 - 9x_4 = 18 \\ 5x_1 + 7x_2 + 2x_3 - 5x_4 = 25 \\ -x_1 + 3x_2 - 7x_3 + 7x_4 = 30 \end{cases}$$

```
>> A = [1 2 3 5; -2 5 7 -9; 5 7 2 -5; -1 -3 -7 7];
>> B = [21; 18; 25; 30];
>> S = A\B
```

6. СОЗДАНИЕ m-ФАЙЛОВ

Создание программы в среде MATLAB осуществляется либо при помощи собственного встроенного, либо стороннего текстового редактора, который вызывается автоматически, если он предварительно установлен с помощью команды Preferences меню File. Чтобы создать новый М-файл, нужно вызвать команду меню File→New→Script. Или File→New→Function.

В языке MATLAB имеются программы двух типов: так называемые Script-файлы (файлы-сценарии или управляющие программы) и файлы-функции (процедуры). Файлы обоих типов имеют расширение ".m". При помощи Script-файлов оформляются основные программы, управляющие от начала и до конца организацией всего вычислительного процесса, или отдельные части основных программ. Как файлы-функции оформляются и отдельные процедуры и функции (т.е. такие части программы, которые рассчитаны на неоднократное использование Script-файлами или другими процедурами при изменяемых значениях входных параметров).

Главным отличием текстов этих двух видов файлов является то, что файлы-функции имеют первую строку вида:

function <ПКВ> = <имя процедуры> (<ПВВ>)

где ПКВ – Перечень Конечных Величин, ПВВ – Перечень Входных Величин.

Script-файлы такой строки не имеют.

Основные особенности записи текста М-файла:

- обычно каждый оператор записывается в отдельной строке текста программы. Признаком конца оператора является символ (он не появляется в окне) возврата каретки и перехода на следующую строку, который вводится в программу при нажатии клавиши [Enter];

- можно размещать несколько операторов в одной строке. Тогда предыдущий оператор должен заканчиваться символом точка с запятой (;);

- длинный оператор можно записывать в несколько строк. При этом предыдущая строка оператора должна заканчиваться тремя точками "...";

- если оператор не заканчивается символом ";", результат его действия при выполнении программы будет выведен в командное окно MATLAB;
- строка программы, начинающаяся с символа "%" не выполняется. Эта строка воспринимается системой MATLAB как комментарий;
- строки комментария, предшествующие первому выполняемому оператору программы, воспринимаются системой MATLAB как описание программы.
- в программах на языке MATLAB отсутствует оператор окончания текста программы;
- в языке MATLAB переменные не описываются и не объявляются. Любое новое имя, появляющееся в тексте программы, воспринимается системой MATLAB как имя матрицы. Размер этой матрицы устанавливается при предварительном вводе значений её элементов либо определяется действиями по установлению значений её элементов, описанными в предыдущем операторе или процедуре. В языке MATLAB невозможно использование матрицы или переменной, в которой предварительно не введены или не вычислены значения её элементов (а значит, и не определены размеры этой матрицы). В этом случае при выполнении программы MATLAB выдаст сообщение об ошибке "Переменная не определена".

7. ПРОГРАММИРОВАНИЕ В MATLAB

7.1. Отношения и логические операции

Операции отношения сравнивают два числа, выдавая ответ истинна или ложь (true/false). Логические операции рассматривают true/false утверждения и выполняют действия, которые соответствуют true или false операторам. Отношения и логические операции используются в математических выражениях и в комбинациях с другими командами. MATLAB имеет шесть операций отношения, которые приведены в таблице 7.1

Таблица 7.1 – Операции отношения

Операции отношения	Описание
<	Меньше
<=	Меньше или равно
>	Больше
>=	Больше или равно
==	Равно
~=	Не равно

Основные логические операции приведены в таблице 7.2.

Таблица 7.2 – Логические операции

Логическая операция	Имя	Эквивалентная запись логической операции	Описание

$\&$ Пример A&B	AND	and (A,B)	Если оба операнда (A и B) true, то и результат true. В противном случае false.
$ $ Пример A B	OR	or(A,B)	Если один или второй операнд true, то и результат true. В противном случае false.
$\sim$ Пример $\sim A$	NOT	not(A)	true, если операнд A false и false, если операнд A true.

Приоритет логических операций следующий:

NOT – самый высокий приоритет;

AND – средний приоритет;

OR – самый низкий приоритет.

7.2 Операторы управления вычислительным процессом

7.2.1. Условный оператор *if*

В MATLAB можно использовать условные операторы. В самом простом случае синтаксис данного оператора *if* имеет вид:

```
if <выражение>
<операторы>
end
```

Если значение параметра "выражение" соответствует значению true, то выполняется оператор, иначе он пропускается программой. Следует отметить, что "выражение" является условным выражением, в котором выполняется проверка некоторого условия при помощи операторов отношения, приведённых в таблице 7.1.

Например, **if a<b**.

Например, приведём реализации функции **sign()**, которая возвращает +1, если число больше нуля, -1 – если число меньше нуля и 0, если число равно нулю:

```
x = 5;
if x > 0
disp(1);
end
if x < 0
disp(-1);
end
if x == 0
disp(0);
```

В приведённом примере видно, что все эти три условия являются взаимоисключающими, т.е. при срабатывании одного из них нет необходимости проверять другие. Реализация именно такой логики позволит увеличить скорость выполнения программы. Этого можно добиться путём использования конструкции

```
if <выражение>
<операторы1>    % выполняются, если истинно условие
else
<операторы2>    % выполняются, если условие ложно
end
```

Тогда приведённый выше пример можно записать следующим образом:

```
x = 5;
if x > 0
    disp(1);
else
    if x < 0
        disp(-1);
    else
        disp(0);
    end
end
```

Приведённый выше пример можно записать в более простой форме, используя ещё одну конструкцию оператора **if** языка MATLAB:

```
if <выражение1>
<операторы1>    % выполняются, если истинно выражение1
elseif <выражение2>
<операторы2>    % выполняются, если истинно выражение2
...
elseif <выражениеN>
<операторыN>    % выполняются, если истинно выражениеN
end
```

```
x = 5;
if x > 0
    disp(1);          % выполняется, если x > 0
elseif x < 0
    disp(-1);         % выполняется, если x < 0
else
    disp(0);          % выполняется, если x = 0
end
```

Рассмотрим **пример** использования составных условий. Пусть требуется проверить попадание переменной *x* в диапазон от 0 до 3. Программа будет иметь следующий вид:

```
x = 1;
if x >= 0 & x <= 3
    disp('x принадлежит диапазону от 0 до 3');
else
    disp('x не принадлежит диапазону от 0 до 3');
end
```

7.2.2 Оператор цикла *while*

Язык программирования MATLAB имеет два оператора цикла: **while** и **for**. С их помощью, например, выполняется программирование рекуррентных алгоритмов, подсчёта суммы ряда, перебора элементов массива и многое другое.

В самом простом случае цикл в программе организуется с помощью оператора **while**, который имеет следующий синтаксис:

```

while <условие>
    <операторы>
end

```

Здесь <условие> означает условное выражение подобное тому, которое применяется в операторе **if**, и цикл **while** работает до тех пор, пока это условие **true**. Если условие будет **false** до начала выполнения цикла, то операторы, входящие в цикл, не будут выполнены ни разу.

Например, запишем программу для подсчёта суммы ряда $S = \sum_{i=1}^{30} i$:

```

S = 0;           % начальное значение суммы
i=1;            % счётчик суммы
while i <= 30    % цикл (работает пока i <= 30)
    S=S+i;      % подсчитывается сумма
    i=i+1;      % увеличивается счётчик на 1
end            % конец цикла
disp(S);       % отображение суммы на экране

```

Подсчитаем сумму ряда $S = \sum_{i=1}^{30} i$, пока $S \leq 30$. Здесь в операторе цикла получается два условия: либо счётчик по i доходит до 30, либо значение суммы S превысит 30. Данную логику можно реализовать с помощью составного условного выражения в операторе цикла **while**:

```

S = 0;           % начальное значение суммы
i=1;            % счётчик суммы
while i <= 30 & S <= 30 % цикл (работает пока i<=30 и S<=30)
    S=S+i;      % подсчитывается сумма
    i=i+1;      % увеличивается счётчик на 1
end            % конец цикла
disp(S);       % отображение суммы на экране

```

Работу любого оператора цикла, в том числе и **while**, можно принудительно завершить с помощью оператора **break**.

7.2.3. Оператор цикла for

Синтаксис оператора цикла **for** имеет следующий вид:

```

for <счётчик> = <нач. значение>:<шаг>:<кон. значение>
    <операторы цикла>
end

```

Рассмотрим работу данного цикла на примере реализации алгоритма поиска максимального значения элемента в векторе:

```

a = [5 3 7 4 2 9 5 4 1 2];
m = a(1);           % текущее максимальное значение
for i=2:length(a)   % цикл от 2 до конца вектора с
                    % шагом 1 (по умолчанию)
    if m < a(i)      % если a(i) > m,
        m = a(i);   % то m = a(i)
    end
end

```

```
end                % конец цикла for
disp(m) ;
```

Если величина шага не указывается явно, то он берётся по умолчанию равным 1. Для реализации цикла со счётчиком от большего значения к меньшему, нужно явно указывать шаг, например, -1. Если этого не сделать, то цикл сразу завершит свою работу.

8. ПОСТРОЕНИЕ ГРАФИКОВ

8.1. 2-D графики

Основной функцией, осуществляющей построение графиков на экране дисплея, является функция **plot**. Общая форма обращения к данной функции такова:

```
plot(x1,y1,linespec1,x2,y2, linespec2,...)
```

где x_i – массив значений аргумента, а y_i – массив значений функции, соответствующей i -тому графику. Параметры $linespec_i$ являются символьными (их указание необязательно). Параметр $linespec_i$ может содержать три специальных символа, определяющих тип линии, тип точки и цвет линии графика.

Команда **plot**(y) строит график вектора y , используя в качестве аргумента порядковый номер элементов вектора. Если y – вектор комплексных чисел, то команда **plot**(y) отображает на комплексной плоскости элементы вектора, т. е. действует аналогично команде **plot**($\text{real}(y), \text{imag}(y)$).

Ниже приведены некоторые процедуры оформления графиков:

grid – нанесение координатной сетки;

title – заголовок графика;

xlabel – название оси x ;

ylabel – название оси y .

Чтобы создать несколько графических окон, необходимо воспользоваться командой **figure**, которая создаёт новое графическое окно, оставляя предыдущие.

Чтобы разместить несколько графиков в одном окне, необходимо воспользоваться командой **subplot**. Обращение к этой процедуре должно предшествовать обращению к процедуре построения графика и выглядеть следующим образом:

```
subplot(n,m,p)
```

где число n указывает, на сколько частей делится поле графика по вертикали; m – по горизонтали; p – номер графического подокна, в котором будет строиться график. Подокна нумеруются слева направо, сверху вниз.

Например, построим три графика

$$y_1 = \sin t;$$

$$y_2 = t;$$

$$y_3 = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!}; \quad 0 \leq t \leq 2\pi.$$

Для этого используем следующую программу в MATLAB

```
% using the plot command
t = linspace(0, 2*pi, 100);
```

```

y1 = sin(t);
y2 = t;
y3 = t - (t.^3)/6 + (t.^5)/120 - (t.^7)/5040;
plot(t, y1, t, y2, t, y3, 'o')
axis([0 5 -1 5]); %ограничиваем поле графика
xlabel('t')
ylabel('sin(t) approximation')
title('sin(t) function')
text(3.5,0, 'sin(t)')
gtext('Linear approximation')
gtext('4-term approximation')

```

В результате, получаем графики, приведённые на рисунке 8.1.

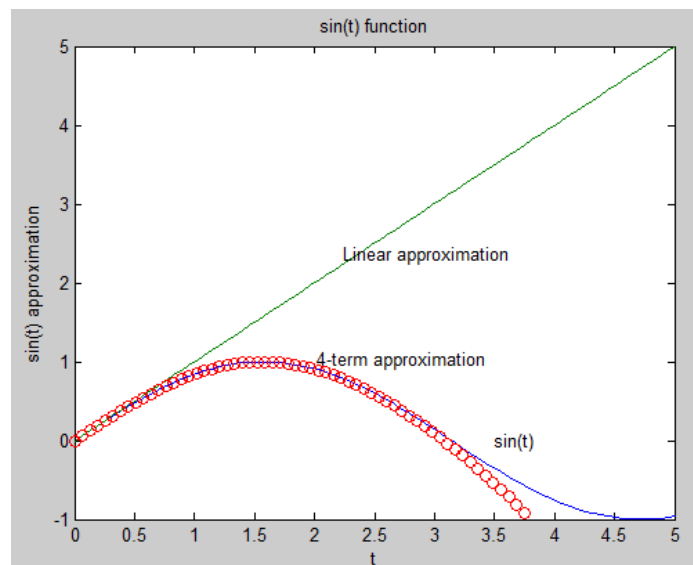


Рисунок 8.1 – Графики заданных функций

8.2. 3-D графики

MATLAB обладает большим набором средств трёхмерной графики. Прежде всего, это различные средства построения поверхностей, описываемых функцией двух переменных

В самом простом случае, для визуализации графика в трёхмерных координатных осях, используется функция `plot3(X, Y, Z)`. В качестве первых двух аргументов она принимает матрицы с координатами точек по осям Ox и Oy соответственно, а в качестве третьего аргумента ей передаётся матрица значений точек по оси Oz .

Например, построим график функции $z(x,y) = \exp(-x^2 - y^2)$, при $x = -1, -0,9, \dots, 1$ и $y = -2, -1,9, \dots, 2$.

Сформируем матрицы X и Y , содержащие координаты точек данного графика по осям Ox и Oy соответственно. Данные матрицы нужны для того, чтобы функция `plot3()` определяла, какие реальные координаты соответствуют точке $z(i,j)$ матрицы значений по оси Oz . Для этого достаточно взять i -ю и j -ю компоненту матриц $x = X(i, j)$; $y = Y(i, j)$.

Формирование матриц X и Y можно осуществить с помощью функции

[X,Y]=meshgrid(x,y) .

Здесь x и y – одномерные векторы значений координат по осям Oх и Oу соответственно, которые можно сформировать как

x=-1:0.1:1; % координаты точек по оси Oх y=-2:0.1:2; % координаты точек по оси Oу и,

затем, вычислить матрицы

[X,Y]=meshgrid(x,y); % матрицы коорд. точек по осям Oх и Oу

В результате, матрицы X и Y будут содержать следующие первые восемь значений по строкам и столбцам:

Матрица X:

-1	-0,9	-0,8	-0,7	-0,6	-0,5	-0,4	-0,3
-1	-0,9	-0,8	-0,7	-0,6	-0,5	-0,4	-0,3
-1	-0,9	-0,8	-0,7	-0,6	-0,5	-0,4	-0,3
-1	-0,9	-0,8	-0,7	-0,6	-0,5	-0,4	-0,3
-1	-0,9	-0,8	-0,7	-0,6	-0,5	-0,4	-0,3
-1	-0,9	-0,8	-0,7	-0,6	-0,5	-0,4	-0,3
-1	-0,9	-0,8	-0,7	-0,6	-0,5	-0,4	-0,3
-1	-0,9	-0,8	-0,7	-0,6	-0,5	-0,4	-0,3

Матрица Y:

-2	-2	-2	-2	-2	-2	-2	-2
-1,9	-1,9	-1,9	-1,9	-1,9	-1,9	-1,9	-1,9
-1,8	-1,8	-1,8	-1,8	-1,8	-1,8	-1,8	-1,8
-1,7	-1,7	-1,7	-1,7	-1,7	-1,7	-1,7	-1,7
-1,6	-1,6	-1,6	-1,6	-1,6	-1,6	-1,6	-1,6
-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5
-1,4	-1,4	-1,4	-1,4	-1,4	-1,4	-1,4	-1,4
-1,3	-1,3	-1,3	-1,3	-1,3	-1,3	-1,3	-1,3

Используя данные матрицы, можно вычислить значения матрицы Z, следующим образом:

Z=exp(-X.^2-Y.^2) ;

и отобразить результат на экране (рисунок 8.2)

plot3(X,Y,Z) ;

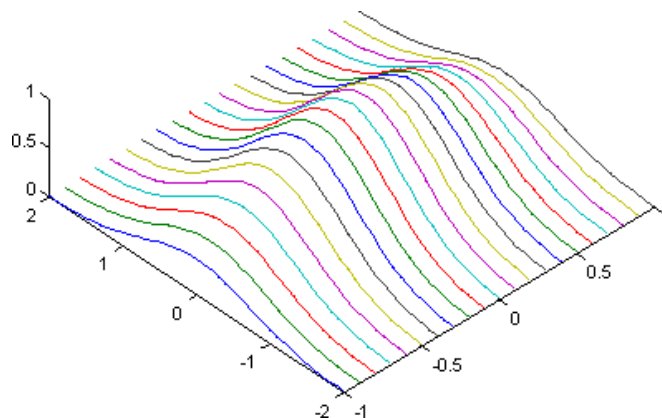


Рисунок 8.2 – Пример отображения графика с помощью функции plot3()

Функция `plot3()` отображает график в виде набора линий, каждая из которых соответствует сечению графика функции $z(x,y) = \exp(-x^2 - y^2)$, вдоль оси Oy . Такое представление графика не всегда удобно, т.к. набор одномерных кривых не даёт полное представление о характере поверхности. Более лучшей визуализации (рисунок 8.3) можно добиться, используя функцию

`mesh(X,Y,Z);` % отображение графика в виде сетки

В MATLAB предусмотрена функция визуализации непрерывной поверхности в трёхмерных осях

`surf(X,Y,Z);` % отображение непрерывной поверхности

В результате получается график, представленный на рисунке 8.4.

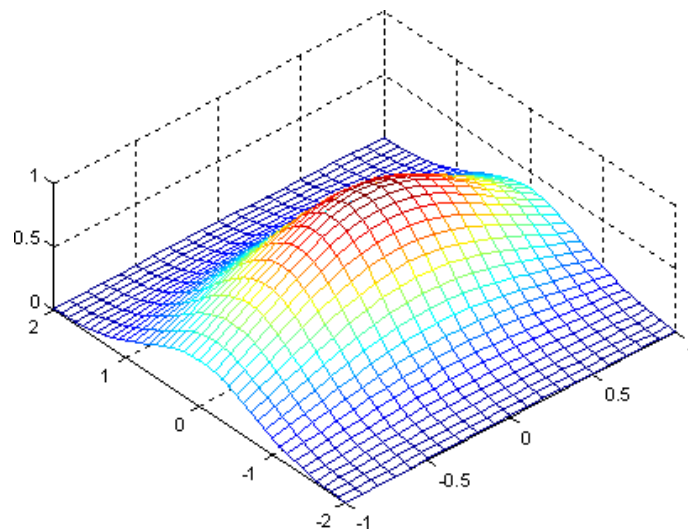


Рисунок 8.3 – Результат работы функции `mesh()`

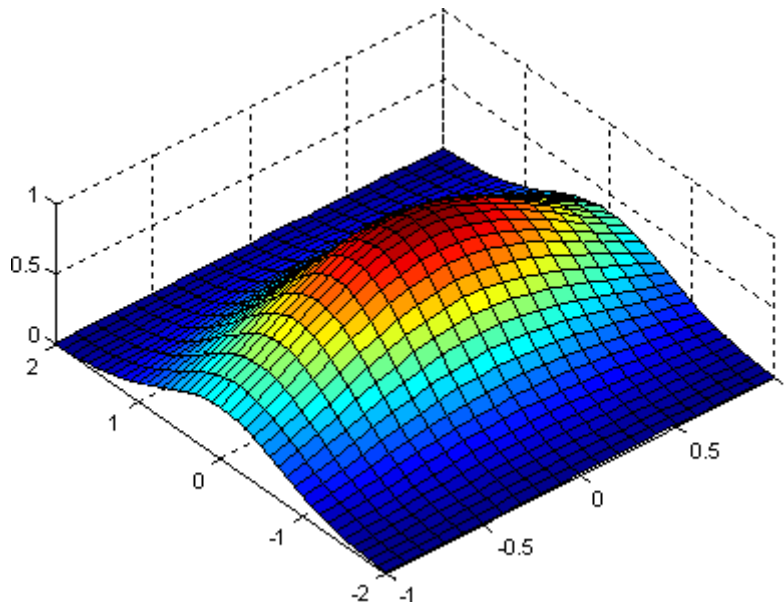


Рисунок 8.4 – Результат работы функции `surf()`

Для оформления трёхмерных графиков можно пользоваться описанными ранее функциями: **`text`**, **`xlabel`**, **`ylabel`**, **`zlabel`**, **`title`**, **`grid [on/off]`**, **`subplot`**.

Наконец, для трёхмерных графиков существует возможность изменять точку их обзора, т.е. положение виртуальной камеры с помощью функции

view([az el]);

где **az** – угол азимута; **el** – угол возвышения.

Изменение первого угла означает вращение плоскости xOy вокруг оси Oz против часовой стрелки. Угол возвышения есть угол между направлением на камеру и плоскостью xOy.

9. СИМВОЛЬНАЯ МАТЕМАТИКА

В разделах приведённых выше, мы использовали MATLAB численных вычислений. Однако, MATLAB позволяет производить вычисления и в символьном виде для решения уравнений, систем линейных уравнений, дифференциальных уравнений, интегральных уравнений и др.

Символьные выражения хранятся в MATLAB в формате *character string*. Одинарные кавычки используются для определения символьного выражения.

Например,

'sin(y/ x)'; 'x^4 + 5 * x^3 + 7 * x^2 – 7'

Независимая переменная во многих функциях определяется как дополнительный аргумент. Если независимая переменная не задаётся, то MATLAB выберет существующую переменную. Если существует несколько переменных, то MATLAB выберет переменную записанную маленькими буквами (за исключением i и j), которая наиболее близка к x в алфавите. Независимая переменная возвращается функцией **symvar(s)**.

Например,

Expression (s)	symvar(s)
'5 * c * d + 34'	d
'sin(y/ x)'	x

9.1 Решение уравнений

Для решений алгебраических уравнений и систем алгебраических уравнений используются следующие функции:

solve(f) – решает символьное уравнение $f=0$ для заданной символьной переменной.

solve(f1,...,fn) – решает систему уравнений, представленных $f1,...,fn$.

Символьная функция для решения обыкновенного дифференциального уравнения - **dsolve('equation', 'condition')**, где уравнение определяется параметром **'equation'**, а аргумент **'condition'** определяет либо граничные, либо начальные условия.

Символьное уравнение использует символ D для обозначения операции дифференцирования по отношению к независимой переменной. Следовательно, Du определяет dy/dx и D2y определяет d^2y/dx^2 . Например, задано обыкновенное дифференциальное уравнение второго порядка

$$\frac{d^2x}{dt^2} + 5\frac{dx}{dt} + 3x = 7$$

с начальными условиями $x(0)=0$ и $x'(0)=1$.

В MATLAB выражение, которое определяет символьное решение приведённого дифференциального уравнения будет иметь вид:

```
x = dsolve('D2x=-5*Dx-3*x+7','x(0)=0','Dx(0)=1')
```

9.2. Символьные вычисления

Приведём функции для организации символьных вычислений в MATLAB. **diff(f)** – возвращает производную выражения f по отношению к независимой переменной по умолчанию.

diff(f,'t') – возвращает производную выражения f по отношению к переменной t .

diff(f,n) – возвращает n -ую производную выражения f по отношению к независимой переменной по умолчанию.

diff(f,'t',n) – возвращает n -ую производную выражения f по отношению к переменной t .

int(f) – возвращает интеграл выражения f по отношению к независимой переменной по умолчанию.

int(f,'t') – возвращает интеграл выражения f по отношению к переменной t .

int(f,a,b) – возвращает интеграл выражения f по отношению к независимой переменной по умолчанию на интервале $[a, b]$, где a и b – числа.

int(f,'t',a,b) – возвращает интеграл выражения f по отношению к переменной t на интервале $[a, b]$, где a и b – числа.

Например, вычислим производную следующей функции $S = 5x^3 - 7x^2 + 3x + 6$

```
>> syms x  
>> S=5*x^3-7*x^2+3*x+6;  
>> diff (S)
```

Вычислим интеграл $\int_0^{\pi} (\cos y + 7y^2) dy$

```
>>syms y  
>> S2=cos (y)+7*y^2  
>> int (S2,0,pi)
```

10. ЗАДАНИЕ НА ВЫПОЛНЕНИЕ РАБОТЫ

1. Рассчитать следующее выражение, используя командное окно MATLAB

$$1.1 \frac{17(\sqrt{5}-1)}{\lceil 15^2 - 13^2 \rceil} + \frac{5^7 \log_0(e^3)}{\pi \sqrt{121}} + \ln(e^4) + \sqrt{11},$$

$$1.2 B = \frac{\tan x + \sin 2x}{\cos x} + \log_{10} |x^5 - x^2| + \cosh x - 2 \tanh x; \quad x = 5\pi/6,$$

1.3 $x = a + \frac{ab(a+b)}{c\sqrt{|ab|}} + c^a + \frac{14b}{e^{3c}} + \ln(2) + \frac{\log_{10} c}{\log_{10}(a+b+c)} + 2\sinh a - 3\tanh b,$
 $a=1, b=2, c=1.8.$

2. Задана функция $y = (x^{\sqrt{2}+0.02} + e^x)^{1.8} \ln x$. Определите значение y для следующих значений $x=2, 3, 8, 10, -1, -3, -5, -6.2$. Решите задачу, создав вектор x и создав вектор y , используя поэлементное вычисление.

3. Введите следующие три матрицы

$$A = \begin{bmatrix} 1 & 2 & 3 \\ -8 & 5 & 7 \\ -8 & 4 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} 12 & -5 & 4 \\ 7 & 11 & 6 \\ 1 & 8 & 13 \end{bmatrix}, \quad C = \begin{bmatrix} 7 & 13 & 4 \\ -2 & 8 & -5 \\ 9 & -6 & 11 \end{bmatrix}.$$

Выполните действия: $(A+B)^T C$, ABC , $A^{-1}B + C$

4. Заданы две матрицы

$$A = \begin{bmatrix} 3 & 2\pi \\ 5j & 10 + \sqrt{2}j \end{bmatrix}, \quad B = \begin{bmatrix} 7j & -15j \\ 2\pi & 18 \end{bmatrix}$$

Выполните действия: $A+B$; AB ; A^2 ; B^{-1} ; $B^T A^T$; $A^2 + B^2 - AB$.

5. Заданы матрицы

$$A = \begin{bmatrix} 2 & 1 \\ 0 & 5 \\ 7 & 4 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 3 \\ -5 & -2 \\ 0 & 3 \end{bmatrix}.$$

Вычислить:

$$(AC^T)^{-1}; \quad (C^T A)^{-1}$$

6. Заданы многочлены

$$p_1(x) = x^5 + 2x^4 - 3x^3 + 7x^2 - 8x + 7;$$

$$p_2(x) = x^4 + 3x^3 - 5x^2 + 9x + 11;$$

$$p_3(x) = x^3 - 2x^2 - 3x + 9.$$

Рассчитайте значение многочленов в точке $x=2$

7. Вычислите нули следующих многочленов

$$p_1(x) = x^7 + 8x^6 + 5x^5 + 4x^4 + 3x^3 + 2x^2 + x + 1;$$

$$p_2(x) = x^6 - 7x^5 + 7x^4 + 15x^3 - 10x^2 - 8x + 15;$$

$$p_3(x) = x^5 - 13x^4 + 10x^3 + 12x^2 + 8x - 15.$$

8. Решите системы алгебраических уравнений

$$\begin{cases} x_2 - 3x_3 = -7 \\ 2x_1 + 3x_2 - x_3 = 9 \\ 4x_1 + 5x_2 - 2x_3 = 15 \end{cases},$$

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 5x_4 = 21 \\ -2x_1 + 5x_2 + 7x_3 - 9x_4 = 18 \\ 5x_1 + 7x_2 + 2x_3 - 5x_4 = 25 \\ -x_1 + 3x_2 - 7x_3 + 7x_4 = 30 \end{cases}, \quad \begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 8 \\ 2x_1 - 2x_2 - x_3 - x_4 = -3 \\ x_1 - 3x_2 + 4x_3 - 4x_4 = 8 \\ 2x_1 + 2x_2 - 3x_3 + 4x_4 = -2 \end{cases}.$$

9. Постройте график функции $y = |x| \cos(x)$, $-200 \leq x \leq 200$.10. Постройте графики функций $\sin^2(x)$, $\cos^2(x)$ на интервале $0 \leq x \leq 2\pi$.11. Задана функция $z = 0.56 \cos(xy)$. Постройте её график при $x \in [0, 10]$, $y \in [0, 100]$.

12. Найдите производную для следующих функций

$$S_1 = e^{x^8}, \quad S_2 = 3x^3 e^{x^5}, \quad S_3 = 5x^3 - 7x^2 + 3x + 6.$$

13. Вычислите первую и вторую производные следующих функций

$$F(x) = x^5 - 8x^4 + 5x^3 - 7x^2 + 11x - 9,$$

$$F(x) = (x^3 + 3x - 8)(x^2 + 21),$$

$$F(x) = (3x^3 - 8x^2 + 5x + 9) / (x + 2),$$

$$F(x) = (x^5 - 3x^4 + 5x^3 + 8x^2 - 13)^2,$$

$$F(x) = (x^2 + 8x - 11) / (x^7 - 7x^6 + 5x^3 + 9x - 17).$$

14. Вычислите интеграл для заданных функций:

$$|x| \text{ на интервале } [0.2, 0.7]; \quad \cos y + 7y^2 \text{ на интервале } [0, \pi]; \quad \sqrt{x}, \quad \cos x, \\ 7x^5 - 6x^4 + 11x^3 + 4x^2 + 8x + 9 \text{ на интервале } [0, 2\pi].$$

15. Решите следующие дифференциальные уравнения

$$\frac{dy}{dt} = 5t - 6y, \quad \frac{d^2y}{dt^2} + 3\frac{dy}{dt} + y = 0.$$

16. Решите следующие дифференциальные уравнения, удовлетворяющие начальным условиям

$$\frac{dy}{dx} = -7x^2, \quad y(1) = 0.7; \quad \frac{dy}{dx} = 5x \cos^2 y, \quad y(0) = \pi/4.$$

11. СОДЕРЖАНИЕ ОТЧЕТА И ПОРЯДОК ЗАЩИТЫ РАБОТЫ

Отчёт должен включать в себя следующие разделы:

- цель работы;
- постановка задачи – номер варианта, исходные данные, формула вычисления и значения входных переменных;
- краткие теоретические сведения, содержащие описания используемых функций MATLAB (назначение функции, перечень и назначение входных и выходных параметров, основные варианты вызова функции);
- тексты программ;
- результаты работы программ – числовые данные, графики;
- общие выводы по работе;
- перечень использованной литературы.

Выполнение и защита практической (лабораторной) работы производится каждым обучающимся индивидуально. Защита результатов работы осуществляется при наличии работающих программ, полностью оформленного отчёта и состоит в следующем:

- представление работающих программ на компьютере;
- предъявление отчёта, оформленного в соответствии с указанными требованиями;
- ответы на вопросы преподавателя по теоретической и практической части работы.

12. КОНТРОЛЬНЫЕ ВОПРОСЫ

Во время защиты практической (лабораторной) работы необходимо уметь выполнять все задания, приведённые в методических указаниях без использования справочной литературы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кошкидько, В. Г. Основы программирования в системе MATLAB: Учебное пособие / Кошкидько В.Г., Панычев А.И. - Таганрог: Южный федеральный университет, 2016. - 84 с.: ISBN 978-5-9275-2048-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/991834>. – Режим доступа: по подписке.
2. Галушкин, Н. Е. Высокоуровневые методы программирования. Язык программирования MatLab. Часть 1: учебник / Н. Е. Галушкин. - Ростов н/Д: Издательство ЮФУ, 2011. - 182 с. - ISBN 978-5-9275-0810-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/550402>. – Режим доступа: по подписке.
3. Ревинская, О. Г. Символьные вычисления в MatLab : учебное пособие для вузов / О. Г. Ревинская. — Санкт-Петербург : Лань, 2020. — 528 с. — ISBN 978-5-8114-5490-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/149344>. — Режим доступа: для авториз. пользователей.
4. Трошина, Г. В. Численные расчеты в среде MatLab : учебное пособие / Г. В. Трошина. — Новосибирск : НГТУ, 2020. — 72 с. — ISBN 978-5-7782-4092-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/152243>. — Режим доступа: для авториз. пользователей.

ЦИФРОВОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ И СИСТЕМ В MATLAB

Часть 1

Основы программирования и визуализации данных

Методические указания

Составители: **Карапетьян** Валерий Артемович,
Крамарь Вадим Александрович, **Мирянова** Вера Николаевна

В авторской редакции

Изд. № 108/2022. Объем 1,5 п.л. Усл. печ. л. 1,39. Уч.-изд. л. 1,47.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»