## Лабораторная работа № 1

[ИЗУЧЕНИЕ РАБОЧЕЙ СРЕДЫ MATLAB](" \l "aaa). ЭЛЕМЕНТАРНЫЕ ОПЕРАЦИИ. ВЫЧИСЛЕНИЯ СО СКАЛЯРАМИ И МАТРИЦАМИ

[1. ЦЕЛЬ РАБОТЫ](" \l "aaa)

Изучить рабочую среду MATLAB. Ознакомиться с особенностями выполнения операций для скалярных и матричных данных.

[2. КРАТКИЕ СВЕДЕНИЯ](" \l "aaa)

MATLAB (от англ. MATrix LABoratory) – язык программирования высокого уровня и система инженерных и научных вычислений, разработчиком которого является корпорация MathWorks Inc. (США).

Система MATLAB позволяет:

* + - выполнять математические вычисления;
    - моделировать различные системы;
    - анализировать данные, обрабатывать их и визуализировать;
    - разрабатывать алгоритмы, приложения и пользовательский интерфейс.

При изучении математики используются в той или иной степени:

* *язык* MATLAB (Паскаль- и Си- подобный, объектно-ориентированный),
* *среда* MATLAB (командное окно, редактор, отладчик),
* *управляемая графика* (построение 2D, 3D графиков, создание анимации),
* *библиотека математических функций* (от самых простых (sin(*x*), cos(*x*) и т.п.) до более сложных, как нахождение обратной матрицы, собственных значений, экстремумов, производных и интегралов функций и множество других),
* *программный интерфейс* (набор встроенных процедур, функций и констант).

* Массив – основная форма, используемая MATLAB для работы с данными.

В среде MATLAB можно работать двумя способами:

1. непосредственным набором команд (в этом случае результаты вычисления присваиваются некоторым переменным);
2. с использованием программ, написанных на языке MATLAB, при помощи ввода данных, вычислений и вывода результатов.

Выбор способа работы зависит от решаемой задачи. В обоих случаях используются практически все вычислительные возможности MATLAB. Преимущество второго способа состоит в возможности использования программ в вычислительных экспериментах.

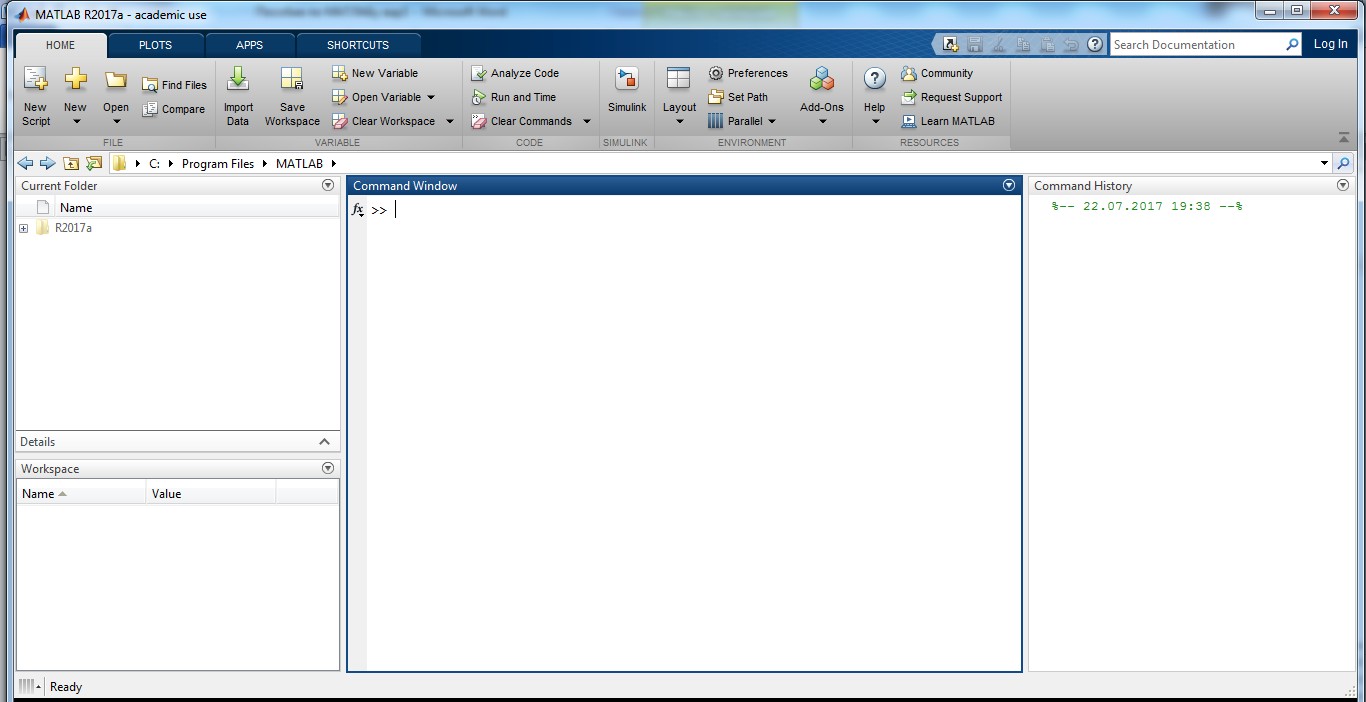
При запуске программы на экране открывается рабочая среда **MATLAB,** изображенная на рисунке 1. Основными элементами рабочей среды являются:

* меню;
* панель инструментов с кнопками;
* окна с вкладками **Workspace** и **Current Folder** для просмотра перемен­ных и установки текущего каталога;
* окно **Command Window,** служащее для ввода команд и вывода результата;
* окно **Command History,** предназначенное для просмотра и повторного выполнения ранее введенных команд (окно **Command History** может быть не пустым, если до этого пакет MATLAB использовался);

Все команды, описанные в настоящем методическом пособии, следует набирать в командной строке. Сам символ >> приглашения командной строки, приведенный в примерах, набирать не нужно. Важно запомнить, что набор любой команды или выражения должен закан­чиваться нажатием на <Enter>, для того чтобы программа MATLAB вы­полнила эту команду или вычислила выражение.

Окно **Command Window** состоит из следующих элементов:

* заголовка с названием окна и двумя кнопками справа;
* рабочей области с командной строкой, в которой находится мигающий вертикальный курсор;
* полос скроллинга.

Рис. 1. Рабочая среда MATLAB

**Таблица 1.** Специальные символы

|  |  |
| --- | --- |
| ***Символ*** | ***Назначение*** |
| **[ ]** | Квадратные скобки используются при задании матриц и векторов |
|  | Пробел служит для разделения элементов матриц |
| **,** | Запятая применяется для разделения элементов матриц и операторов в строке ввода |
| **;** | Точка с запятой отделяет строки матриц, а точка с запятой в конце оператора (команды) отменяет вывод результата на экран |
| **:** | Двоеточие используется для указания диапазона (интервала изменения величины) и в качестве знака групповой операции над элементами матриц |
| **( )** | Круглые скобки применяются для задания порядка выполнения математических операций, а также для указания аргументов функций и индексов матриц |
| **.** | Точка отделяет дробную часть числа от целой его части, а также применяется в составе комбинированных знаков (.\*, .^, ./, .\) |
| **. . .** | Три точки и более в конце строки отмечают продолжение выражения на следующей строке |
| **%** | Знак процента обозначает начало комментария |
| **!** | Восклицательный знак отмечает начало команды MS DOS, например команда !dir выводит оглавление текущего каталога |
| **‘** | Апостроф указывает на символьные строки, а для включения самого апострофа в символьную строку нужно поставить два апострофа подряд |

**Таблица 2.** Форматы вывода числовых данных

|  |  |  |
| --- | --- | --- |
| ***Символ*** |  | ***Назначение*** |
| short или  пустая строка | Значение по умолчанию – масштабируемый десятичный формат с 4 знаками после запятой | 3.1416 |
| long | Масштабируемый десятичный  формат с 14 знаками после запятой | 3.141592653589793 |
| short e | Экспоненциальный формат с 4 знаками после запятой и двумя разрядами для показателя степени | 3.1416e+00 |
| long e | Экспоненциальный формат с 15 знаками после запятой и двумя разрядами для показателя степени | 3.141592653589793e+00 |
| long g | Десятичный формат с 14 знаками после запятой | 3.14159265358979 |
| short g | Десятичный формат с 4 знаками после запятой | 3.1416 |
| longEng | Инженерный формат с 14 знака ми после запятой и тремя знака ми для показателя степени | 3.14159265358979e+000 |
| shortEng | Инженерный формат с 4 знака ми после запятой и тремя знака ми для показателя степени | 3.1416e+000 |
| hex | Шестнадцатиричный формат | 400921fb54442d18 |
| bank | Десятичный формат с двумя знаками после запятой | 3.14 |
| rat | Формат рациональных чисел | 355/113 |
| + | Положительные, отрицательные или нулевые числа отображаются соответственно знаками +, – и пробел, мнимая часть комплексного числа игнорируется | + |
| compact | Подавление вывода пустых строк результата вычислений в консоли |  |
| loose | Включение вывода пустых строк результата вычислений в консоли |  |

[3. методические указания](" \l "aaa)

**ПРИМЕР 1 Простейшие вычисления**

Арифметические операции в MATLAB выполняются в обычном порядке, свойственном большинству языков программирования:

* + возведение в степень – ^;
  + умножение и деление – \*, /;
  + сложение и вычитание – +, -.

Для изменения порядка выполнения арифметических операторов следует использовать круглые скобки.

При вычислениях возможны некоторые исключительные ситуации, например, деление на ноль, которые в большинстве языков программирования приводят к ошибке. При делении положительного числа на ноль в MATLAB получается inf (бесконечность), а при делении отрицательного числа на ноль получается -inf (минус бесконечность) и выдается предупре­ждение:

>> 1/0

Warning: Divide by zero.

ans =

Inf

При делении нуля на ноль получается NaN (не число) и также выдается пре­дупреждение:

>> 0/0

Warning: Divide by zero.

ans =

NaN

Наберите в командной строке 1+2 и нажмите <Enter>. В результате в команд­ном окне MATLAB отображается следующее:

>> 1 + 2

ans =

3

>>

Программа MATLAB сначала вычислила сумму 1 + 2, за­тем записала результат в специальную переменную ans и вывела ее значе­ние, равное 3, в командное окно. Переменная ans автоматически создается, когда вычисляемое выражение не присваивается некоторой переменной. Информация о переменной ans сразу же появилась в окне Workspace ([рис.](#ris3) 2). В первом столбце Name записано имя переменной. Следующий столбец Value показывает значение переменной, если это возможно. Содер­жимое столбца Size, по существу, демонстрирует основной принцип работы MATLAB. Программа MATLAB *все данные представляет в виде массивов.* Переменная ans является двумерным массивом размера один на один и за­нимает 8 байт памяти, о чем свидетельствует столбец Bytes. Наконец, в по­следнем столбце Class указан тип переменной – double array, т. е. массив, состоящий из чисел двойной точности. Любой столбец можно скрыть или отобразить, если на заголовке окна щелкнуть правой кнопкой и вызвать контекстное меню.

В окне **Command Window** ниже ответа расположена командная строка с ми­гающим курсором, обозначающая, что среда MATLAB готова к дальнейшим вычислениям. Можно набирать в командной строке новые выражения и находить их значения.

Если требуется продолжить работу с предыдущим выражением, например вычислить (1+2)/4.5, то проще всего воспользоваться уже имеющимся результатом, который хранится в переменной ans. Наберите в командной строке ans/4.5 (при вводе десятичных дробей используется точка) и нажмите <Enter>, получается:

>> ans/4.5

ans =

0.6667

>>

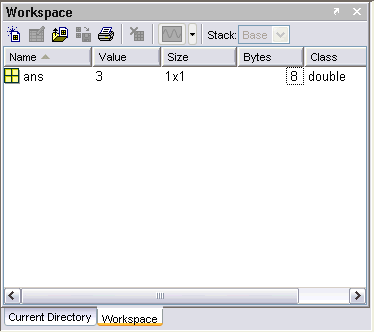


Рис. 2. Окно Workspace с информацией об использованных переменных среды

Если используется операция присваивания Имя\_переменной = Выражение [;]

то результат вычисления выражения помещается не в системную переменную ans, а в переменную с именем «Имя\_переменной». Символ «;» указывает на то, что результат вычисления не будет выводиться на экран. Этот символ используется также и для разделения нескольких операторов в одной строке.

Имена переменных должны формироваться из букв латинского алфавита, цифр, символа подчеркивания и могут начинаться только с буквы. Например,

>> x = sin(1.3\*pi)/log(3.4);

>> y = sqrt(tan(2.75)/tanh(2.75));

>> z = (x + y)/(x - y)

z =

0.0243 - 0.9997i

Последний оператор присваивания не завершается точкой с запятой для того, чтобы сразу получить значение исходного выражения.

## Замечание: заглавные и строчные буквы системой различаются!

**ПРИМЕР 2 Использование элементарных функций**

В MatLab имеются следующие математические функции (Z – в общем случае число комплексное, X и Y – только действительные числа):

|  |  |
| --- | --- |
| abs(Z) | – вычисление модуля комплексного или действительного числа, |
| angle(Z) | – вычисление аргумента, |
| sqrt(Z) | – вычисление квадратного корня, |
| real(Z) | – вычисление действительной части числа, |
| imag(Z) | – вычисление мнимой части числа, |
| round(X) | – округление до целого; |
| fix(X) | – округление до ближайшего целого в сторону нуля, |
| floor(X) | – округление до ближайшего целого в сторону отрицательной  бесконечности, |
| sign(X) | 1, при X  0,    – вычисление функции знака 0, при X  0,     1, при X  0, |
| rem(X, Y) | – вычисление остатка от деления X на Y, |
| exp(Z) | – вычисление экспоненты, |
| log(Z) | – вычисление натурального логарифма, |
| log10(Z) | – вычисление десятичного логарифма, |
| sin(Z) | – вычисление синуса, |
| cos(Z) | – вычисление косинуса, |
| tan(Z) | – вычисление тангенса, |
| asin(Z) | – вычисление арксинуса, |
| acos(Z) | – вычисление арккосинуса, |

|  |  |
| --- | --- |
| atan(Z) | – вычисление арктангенса, |

|  |  |
| --- | --- |
| sinh(Z) | – вычисление гиперболического синуса, |
| cosh(Z) | – вычисление гиперболического косинуса, |
| tanh(Z) | – вычисление гиперболического тангенса, |
| asinh(Z) | – вычисление гиперболического арксинуса, |
| acosh(Z) | – вычисление гиперболического арккосинуса, |
| atanh(Z) | – вычисление гиперболического арктангенса. |

Например,

>> Z=20+20i; asin(Z)

ans =

0.7851 + 4.0355i

Предположим, что требуется вычислить значение следующего выражения:

.

Введите в командной строке это выражение в соответствии с правилами MATLAB и нажмите <Enter>.

>> exp(-2.5)\*log(11.3)^0.3 - sqrt((sin(2.45\*pi) + cos(3.78\*pi))/tan(3.3))

Ответ выводится в командное окно:

ans =

-3.2105

При вводе выражения использованы встроенные функции MATLAB. Аргументы функций заключаются в круглые скобки, имена функций набираются строчными бук­вами. Для ввода числа *π* достаточно набрать pi в командной строке.

**ПРИМЕР 3 Вычисление интегралов**

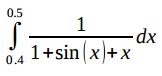
В библиотеке Matlab есть функция **trapz**, которая вычисляет определенный интеграл методом трапеций. Функция trapz имеет следующие синтаксические конструкции:

- **trapz(Y)** — возвращает определенный интеграл, используя интегрирование методом трапеций с единичным шагом между отсчетами. Если Y - вектор, то trapz(Y) возвращает интеграл элементов вектора Y, если Y - матрица, то trapz(Y) возвращает вектор-строку, содержащую интегралы каждого столбца этой матрицы;

- **trapz(X,Y)** — возвращает интеграл от функции Y по переменной X, используя метод трапеций (пределы интегрирования в этом случае задаются начальным и конечным элементами вектора X);

- **trapz(...,dim)** — возвращает интеграл по строкам или по столбцам для входной матрицы в зависимости от значения переменной dim.

Технологию вычисления интеграла в Matlab рассмотрим на примере.  
**Пример** . **Вычислить, разбив на количество отрезков n=100**



Для начала объявим необходимые переменные:

>> n = 100;

>> a1 = 0.4; %Начало определённого интеграла

>> b1 = 0.5; %Конец определённого интеграла

>> h = (b1-a1) / n; %Находим шаг

>> x=a1:h:b1; %задаем шаг

Далее проводим вычисления:

>> y=(1 /(1 + sin(x)+ x));

>> trapz(x,y)

Вывод:  
 **>> ans = 0.0531**

В системе MATLAB методы интегрирования более высоких порядков точности реализуются функциями: *quad* (метод Симпcона) и *quad8* (метод Ньютона-Котеса 8-го порядка точности). Оба этих метода являются к тому же *адаптивными.* Последнее означает, что пользователю нет необходимости контролировать достигнутую точность результата путем сравнения последовательных значений, соответствующих разным шагам интегрирования. Все это указанные адаптивные функции выполняют самостоятельно.

У функции *quad8* более высокий порядок точности по сравнению с функцией *quad*, что очень хорошо для гладких функций, так как обеспечивается более высокая точность результата при большем шаге интегрирования (меньшем объеме вычислений). Как и многие другие функции системы MATLAB, функции *quad* и *quad8* могут принимать различное количество параметров. Минимальный формат вызова этих функций

quad8(name, x1, x2)

включает в себя три параметра: имя подынтегральной функции – *name*, нижний предел интегрирования – *x1* и верхний предел интегрирования – *x2*. Если применяется четвертый параметр, то он является требуемой относительной точностью результата вычислений.

**ПРИМЕР 4. Матричные вычисления**

Поскольку MatLab – матрично-ориентированная система, то в ней даже обычная переменная представляет собой матрицу размерностью 11. Чтобы задать вектор (матрицу из одной строки), состоящий, например, из трех элементов, то их значения необходимо перечислить в квадратных скобках, разделяя пробелами или запятыми:

>> P=[12 25 31] P =

12 25 31

Получить значение каждого элемента вектора можно по его индексу,

например, для второго элемента

>> P(2)

ans =

25

Задание матрицы требует указания различных строк. Для различения

строк используется знак «;», например:

>> Q=[3 6;4 10] Q =

3 6

4 10

>> Q(2,1)

ans = 4

Обращение к элементам матрицы при помощи одного индекса,

поясним на следующем примере.

|  |  |
| --- | --- |
| >> A=[1 2 3; 4 5 6; 7 8 9] A =  1 2 3  4 5 6  7 8 9 | |
| >> A(2)  ans =  4 | >> A(6)  ans = 8 |

Можно осуществлять вывод одной строки или одного столбца матрицы. Для этого используется знак «:», например,

|  |  |
| --- | --- |
| Вывод 2-ой строки  >> A(2,:)  ans =  4 5 6 | Вывод 3-го столбца  >> A(:,3)  ans =  3  6  9 |

Для преобразования матрицы в столбец используется также символ

«:». Например,

>> A(:)

ans =

1

4

7

2

5

Помимо поэлементного ввода векторов и матриц можно задавать значения в виде арифметической прогрессии с шагом 1:

>> x=[1:5]

x =

1 2 3 4 5

с произвольным шагом

>> x=[1:0.5:3]

x =

1.0000 1.5000 2.0000 2.5000 3.0000

или с шагом, заданным при помощи арифметического выражения:

>> x=[0:pi/4:pi] x =

0 0.7854 1.5708 2.3562 3.1416

Следующие матричные функции обеспечивают генерацию некоторых наиболее распространенных видов матриц размерностью M строк на N столбцов:

|  |  |
| --- | --- |
| zeros(M, N) | – генерация матрицы с нулевыми элементами, |
| ones(M, N) | – генерация матрицы с единичными элементами, |
| rand(M, N) | – генерация матрицы с элементами, имеющими  случайные значения, равномерно распределенные в диапазоне от 0 до 1 |
| eye(M, N) | – генерация матрицы с единичными диагональными  элементами. |

Например,

>> Q=eye(3,4) Q =

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |

Транспонирование квадратной матрицы осуществляется при помощи символа апостроф («'»):

3

4

2

4

1

3

>> B=A' B =

1

2

>> A=[1 2;3 4]

A =

Для сложения и вычитания матриц используются обычные знаки «+» и «–». Умножение двух матриц по правилам линейной алгебры осуществляется с использованием знака «\*». В случае, когда требуется перемножить поэлементно две матрицы, используется знак «.\*». Например,

>> A=[1 2;3 4]; B=[3 2;0 1]; C=A\*B C =

3 4

9 10

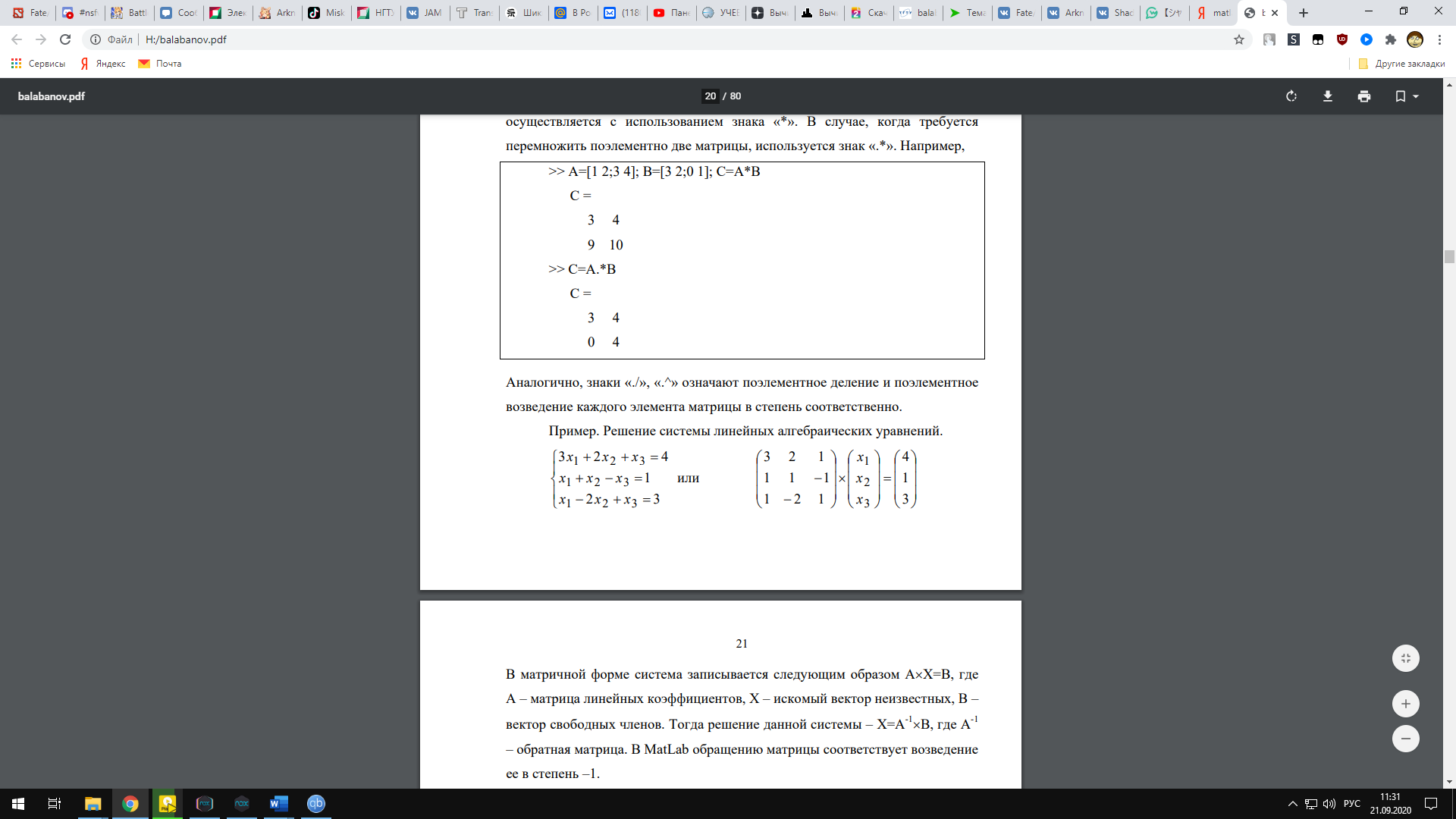
>> C=A.\*B C =

3 4

0 4

Аналогично, знаки «./», «.^» означают поэлементное деление и поэлементное возведение каждого элемента матрицы в степень соответственно.

Пример. Решение системы линейных алгебраических уравнений.



В матричной форме система записывается следующим образом AX=B, где A – матрица линейных коэффициентов, X – искомый вектор неизвестных, B – вектор свободных членов. Тогда решение данной системы – X=A-1B, где A-1

– обратная матрица. В MatLab обращению матрицы соответствует возведение ее в степень –1.

Выполним решение заданной системы уравнений:

>> A=[3 2 1;1 1 -1;1 -2 1]; B=[4;1;3]; X=A^(-1)\*B X =

1.7000

-0.6000

0.1000

Для вычисления определителя квадратной матрицы M используется функция det(M), например:

>> M=eye(3,3)

M =

>> det(M) ans =

1

|  |  |  |
| --- | --- | --- |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

Для определения размерности вектора-строки используется функция length, а размерности матрицы – size. Например,

>> A=[3 2 1];B=[1 2; 3 4;5 6];k=length(A)

k =

3

>> [m,n]=size(B) m =

3

n =

2

Функции min (max) для матрицы возвращают вектор-строку минимальных (максимальных) значений каждого столбца матрицы, например,

>> A=[3 2 1; 1 2 3]; min(A)

ans =

1 2 1

Применение этих функций к векторам-строкам позволяет определить наибольший (наименьший) элемент в строке, например,

>> A=[3 2 1]; max(A)

ans =

3

а также его порядковый номер в строке

>> [M,m]=max(A) M =

3

m =

1

Функции sum, prod, mean и std для матриц возвращают вектор-строку сумм, произведений, средних значений и стандартных квадратичных отклонений каждого столбца матрицы, а для вектора-строки возвращают сумму, произведение, среднее значение и стандартное квадратичное отклонение соответственно.

Для матрицы функция sort возвращает матрицу, у которой отсортированы в порядке возрастания элементы каждого столбца, а для вектора-строки - упорядочивает все его элементы, например,

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 0 |
| 10 | 2 | 3 |
| 23 | 2 | 11 |

Матричная функция find (A) возвращает [индексы](#_bookmark1) ненулевых элементов матрицы A. Например,

>> A=[10 2 11; 23 1 0; 1 2 3]; B=sort(A)

B =

>> A=[10 2 11]; B=sort(A) B =

2 10 11

>>A=[1 0 3; 0 0 4; 2 2 2]; find(A)

ans =

1

3

6

7

8

9

Если в скобках указано логическое условие, то с помощью этой функции осуществляется поиск элементов матрицы, удовлетворяющих этому условию. При этом возвращаются индексы найденных элементов. Например,

|  |  |
| --- | --- |
| >> k=find(A>2) k =  7  8 | >> k=find(A==0)  k =  2  4  5 |

[**4 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**](#aaa)

**Задание 1**. Вычислите значение выражения двумя способами: в одну строчку и с получением промежуточных значений (не менее чем за три шага).



Получите результат в разных форматах и представьте в отчете в виде таблицы:

|  |  |
| --- | --- |
| ***Формат*** | ***Результат*** |
| short |  |
| long |  |
| short e |  |
| long e |  |
| hex |  |
| bank |  |
| rat |  |
| + |  |

Сравните полученные результаты.

**Задание 2**. Вычислите алгебраические выражения при заданных значениях параметров (двумя способами: в одну строчку и с получением промежуточных значений не менее чем за три шага). Сравните полученные результаты.

*Таблица 3*

|  |  |  |  |
| --- | --- | --- | --- |
| Номер  варианта |  | Значения | |
| *а* | *в* |
| 1 |  | 4 | 0,5 |
| 2 |  | 2 | 14,36 |
| 3 |  | 10 | 0,5 |
| 4 |  | 2 | 11,05 |
| 5 |  | 3 | 0,151 |
| 6 |  | 4 | 1,310 |
| 7 |  | 2 | 200,0 |
| 8 |  | 1 | 12,21 |
| 9 |  | 3 | 0,521 |
| 10 |  | 2 | 12,11 |
| 11 |  | 1 | 20,01 |
| 12 |  | 3 | 0,707 |
| 13 |  | 3 | 2,712 |
| 14 |  | 2 | 19,03 |
| 15 |  | 4 | 300,1 |
| 16 |  | 3 | 0,501 |
| 17 |  | 2 | 13,13 |
| 18 |  | 1 | 0,001 |
| 19 |  | 4 | 0,707 |
| 20 |  | 1 | 1,201 |
| 21 |  | 2 | 13,17 |
| 22 |  | 2 | 3,141 |
| 23 |  | 4 | 3,141 |
| 24 |  | 1 | 15,15 |
| 25 |  | 3 | 0,523 |

**Задание 3**. Проверьте корректность тригонометрических тождеств, вычисляя значения левой и правой частей равенства при заданных значениях α и β.

*Таблица 4*

|  |  |  |  |
| --- | --- | --- | --- |
| № | тождество | α | β |
| 1 |  | π/12 |  |
| 2 |  | π/12 |  |
| 3 |  | 24◦ |  |
| 4 |  | 24◦ |  |
| 5 | http://fizmat.by/pic/MATH/page323/im5.jpg | π/6 | 3π/8 |
| 6 | http://fizmat.by/pic/MATH/page323/im3.jpg | π/6 | 3π/8 |
| 7 | http://fizmat.by/pic/MATH/page323/im3.jpg | π/6 | 3π/8 |
| 8 | http://fizmat.by/pic/MATH/page323/im3.jpg | π/6 | 3π/8 |
| 9 | http://fizmat.by/pic/MATH/page323/im3.jpg | π/6 | 3π/8 |
| 10 | http://fizmat.by/pic/MATH/page323/im5.jpg | π/6 | 3π/8 |
| 11 | http://fizmat.by/pic/MATH/page323/im5.jpg | π/6 | 3π/8 |
| 12 | http://fizmat.by/pic/MATH/page323/im5.jpg | π/6 | 3π/8 |
| 13 | http://fizmat.by/pic/MATH/page323/im7.jpg | π/12 |  |
| 14 | http://fizmat.by/pic/MATH/page323/im7.jpg | π/12 |  |
| 15 | http://fizmat.by/pic/MATH/page323/im8.jpg | π/6 | 3π/8 |
| 16 | http://fizmat.by/pic/MATH/page323/im8.jpg | π/6 | 3π/8 |
| 17 | http://fizmat.by/pic/MATH/page323/im8.jpg | π/6 | 3π/8 |
| 18 | http://fizmat.by/pic/MATH/page323/im4.jpg | π/12 |  |
| 19 | http://fizmat.by/pic/MATH/page323/im4.jpg | π/12 |  |
| 20 | http://fizmat.by/pic/MATH/page323/im6.jpg | 2π/3 |  |

**Задание 4**. Вычислите значение определенного интеграла по формуле приближенного метода трапеций, где Δt = (q-p)/(n-1), n выбрать самостоятельно.

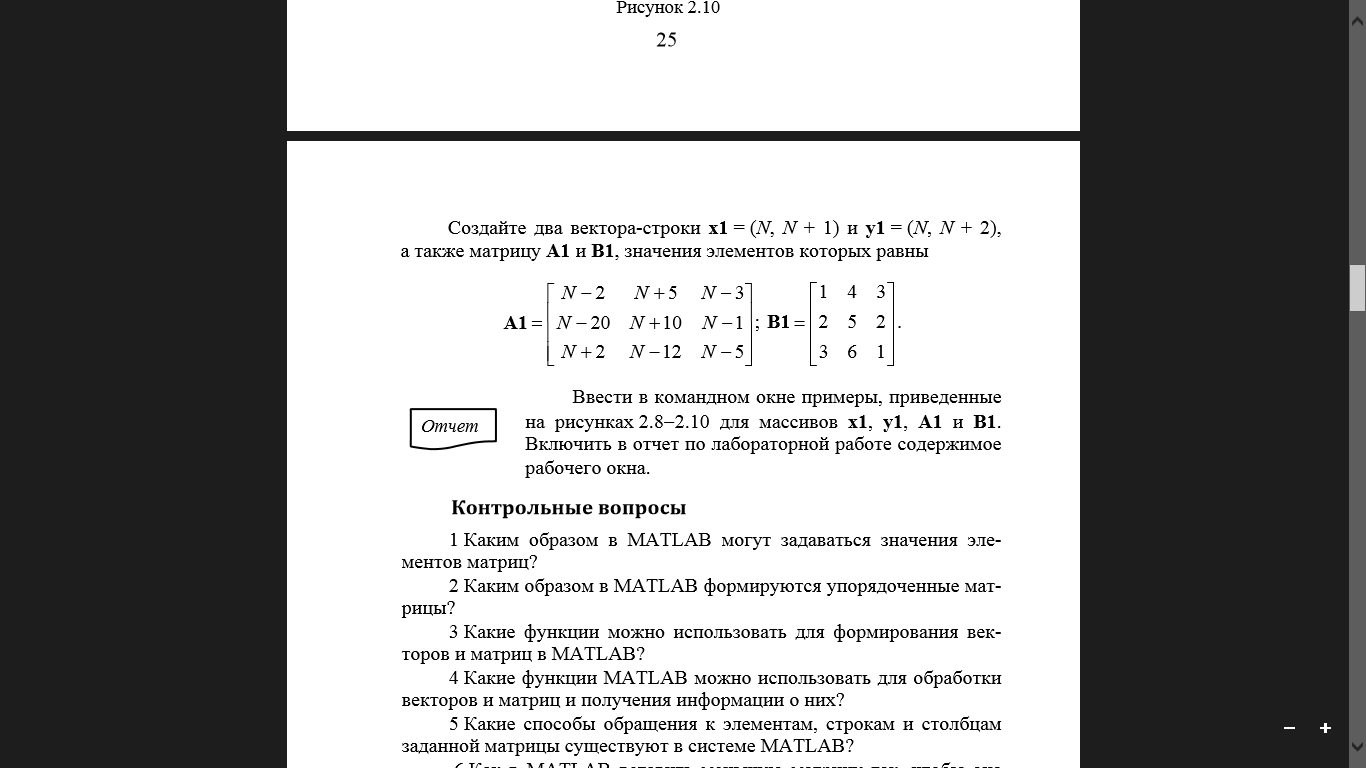
. *Указание:* значение подынтегральной функции в точках получите в виде вектора-строки, а их сумму вычислите, используя функцию ***sum(v)*** – сумма элементов вектора v. Сравните полученные результаты с точным значением, полученным с использованием функции Matlab ***trapz, quad*** и ***quad8.*** Полученные результаты представьте в виде таблицы.

*Таблица 5*

|  |  |
| --- | --- |
| Вариант | Интеграл |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 |  |

**Задание 5**. Научиться выполнять операции над матрицами.

5.1. Создайте вектор-строку **x1 = (N, N + 1, N + 3)** и вектор-столбец   
**y1 = (N - 1, N - 2, N),** а также матрицы A1 и В1, значения элементов которых:



Вычислите значения выражений:

1. Определители и обратные матрицы для А1 (используя функцию inv(А1) или А1-1) и для В1. Сделайте выводы по результатам. Проверьте, что полученные матрицы (если они существуют) действительно являются обратными.

2. А1+В1, А1-В1, А1\*В1, А1/В1, B1/A1, x1\*y1, y1\*x1.

3. Определить значения k и p, где k – остаток от деления N на 5, увеличенный на 2, р – остаток от деления N на 3, увеличенный на 3.

Создать матрицы zeros(k, p), ones(p, k), eye (k+p),

4. Создать матрицу 5×5 и выполнить для нее операции из табл. 7

*Таблица 6*

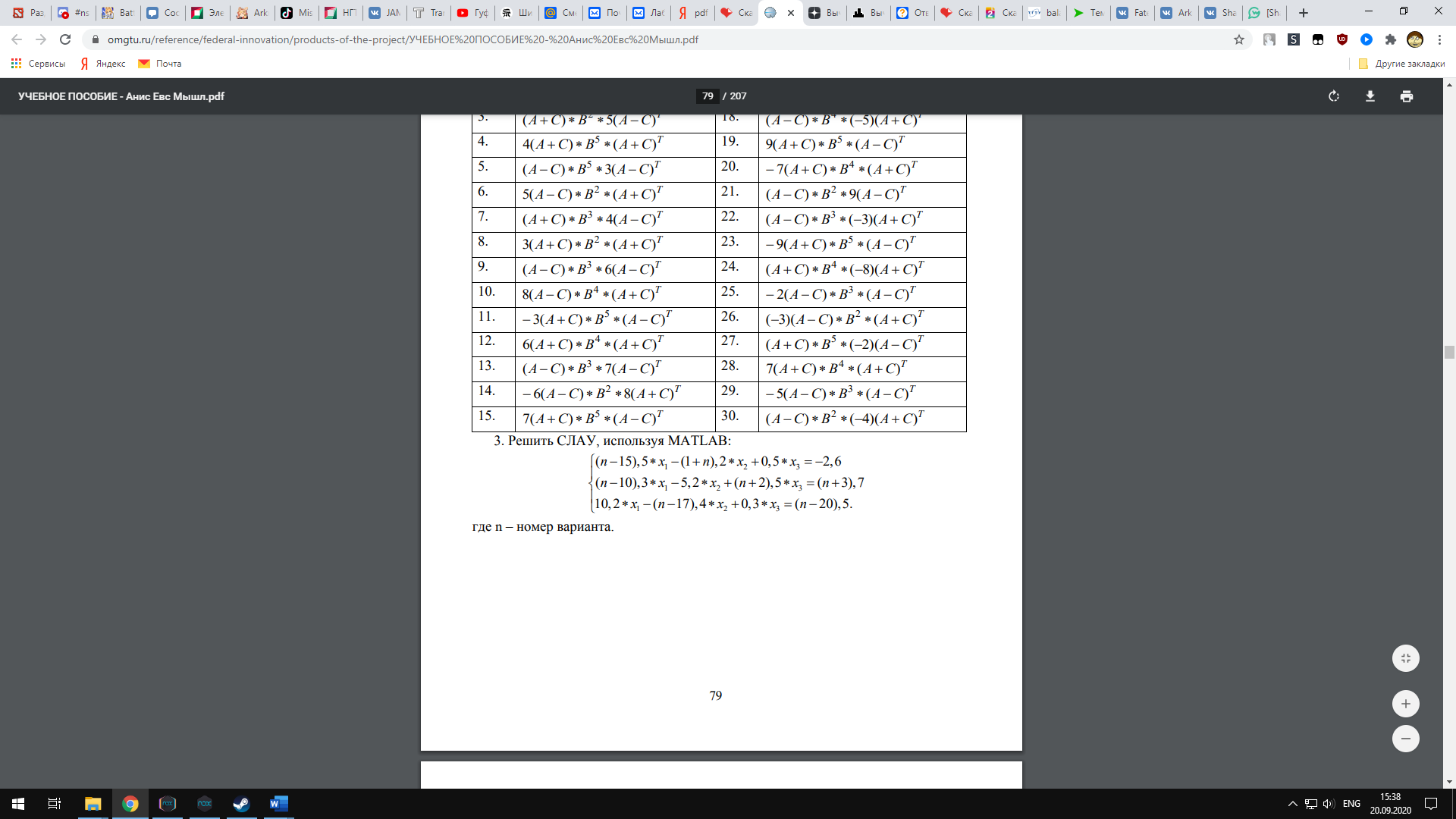
|  |  |
| --- | --- |
| № | Задание |
| 1 | Найти определитель матрицы и сложить его со всеми элементами матрицы |
| 2 | Транспонировать матрицу |
| 3 | Найти наибольшую сумму элементов столбца и сложить его с матрицей |
| 4 | Найти наименьшее произведение элементов строк и умножить на него матрицу |
| 5 | Умножить матрицу на число и найти определитель матрицы |
| 6 | Найти наибольшую сумму элементов строки и сложить его с матрицей |
| 7 | Найти наименьшую сумму элементов строки и сложить его с матрицей |
| 8 | Умножить матрицу на вектор |
| 9 | Поменять местами 2 строку и 2 столбец и подсчитать определитель |
| 10 | Найти наибольшее произведение элементов строк и умножить на него матрицу |
| 11 | Найти наибольшее произведение элементов столбца и сложить его с матрицей |
| 12 | Сложить матрицу с числом |
| 13 | Сложить последнюю строку с последним столбцом, умноженным на 2 |
| 14 | Найти наибольшую сумму элементов строки и поделить на него матрицу |
| 15 | Найти наименьшее произведение элементов строки и поделить на него матрицу |
| 16 | Разделить матрицу на число и найти ее определитель |
| 17 | Вычесть из первого столбца последнюю строку |
| 18 | Поменять местами первый столбец и последнюю строку, умноженную на 2 |
| 19 | Сложить каждый столбец с каждой строкой |
| 20 | Разделить матрицу на ее определитель |
| 21 | Найти сумму максимальных элементов строк и вычесть его из матрицы |
| 22 | Найти произведение минимальных элементов столбцов и сложить с матрицей |
| 23 | Найти произведение максимальных элементов строк и поделить на него матрицу |
| 24 | Найти сумму минимальных элементов столбцов и умножить на него матрицу |
| 25 | Умножить матрицу на ее определитель |

*Указание:* Определитель матрицы А – функция det(A), максимальные (минимальные) элементы по каждому столбцу – функции max(А) (min(А)), для получения максимальных (минимальных) элементов по строкам сначала транспонируйте исходную матрицу. Сумма (произведение) элементов матрицы по каждому столбцу – функции sum(А) (prod(А)). Сумму (произведение) элементов вектора v получаете, используя функцию sum(v) (prod(v)). В поэлементных матричных арифметических операциях два знака, первый из которых – точка.

5.2. Создать матрицы А2х3, В3х3, С2х3 из случайных чисел, равномерно распределенных в диапазоне от 0 до 1. Найти, используя MATLAB, значение заданного выражения:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Выражение | № | Выражение |
| 1. | ( *A*  *C*)  *B*4  2( *A*  *C*)*T* | 11. |  4( *A*  *C*)  *B*2  ( *A*  *C*)*T* |
| 2. | 2( *A*  *C*)  *B*3  ( *A*  *C*)*T* | 12. | ( *A*  *C*)  *B*3  (2)(*A*  *C*)*T* |
| 3. | ( *A*  *C*)  *B*2  5( *A*  *C*)*T* | 13. | ( *A*  *C*)  *B*4  (5)(*A*  *C*)*T* |
| 4. | 4( *A*  *C*)  *B*5  ( *A*  *C*)*T* | 14. | 9( *A*  *C*)  *B*5  ( *A*  *C*)*T* |
| 5. | ( *A*  *C*)  *B*5  3( *A*  *C*)*T* | 15. |  7( *A*  *C*)  *B*4  ( *A*  *C*)*T* |
| 6. | 5( *A*  *C*)  *B*2  ( *A*  *C*)*T* | 16. | ( *A*  *C*)  *B*2  9( *A*  *C*)*T* |
| 7. | ( *A*  *C*)  *B*3  4( *A*  *C*)*T* | 17. | ( *A*  *C*)  *B*3  (3)(*A*  *C*)*T* |
| 8. | 3( *A*  *C*)  *B*2  ( *A*  *C*)*T* | 18. |  9( *A*  *C*)  *B*5  ( *A*  *C*)*T* |
| 9. | ( *A*  *C*)  *B*3  6( *A*  *C*)*T* | 19. | ( *A*  *C*)  *B*4  (8)(*A*  *C*)*T* |
| 10. | 8( *A*  *C*)  *B*4  ( *A*  *C*)*T* | 20. |  2( *A*  *C*)  *B*3  ( *A*  *C*)*T* |

5.3. Решить СЛАУ, используя MATLAB:



Где n – номер варианта.

**5. СОДЕРЖАНИЕ ОТЧЕТА**

5.1. Цель работы.

5.2. Задания.

5.3. Последовательность действий в окне Matlab (скриншот).

5.4. Полученные результаты.

5.5. Выводы.

[6. КОНТРОЛЬНЫЕ ВОПРОСЫ](" \l "aaa)

1. Перечислите основные элементы рабочей среды MATLAB. Из каких элементов состоит окно Command Window?
2. Перечислите основные окна в MATLAB и объясните их назначение.
3. Откуда берется переменная ans?
4. Какие значения принимает переменная ans при делении 1/0 и 0/0?
5. Какие системные переменные MATLAB вы знаете?
6. Что используется в качестве оператора присваивания в среде MATLAB?
7. Каким образом формируется очередная команда в MATLAB?
8. Как вызвать предыдущую команду в MATLAB?
9. Какой символ используется для подавления вывода результатов выполнения операторов?
10. Чем определяются форматы представления чисел при выводе результатов вычислений в MATLAB?
11. Из каких символов может состоять имя переменной в MATLAB?
12. Как вводится комментарий в MATLAB?
13. Какие операции и встроенные функции применяются в MATLAB?
14. Каким образом сохраняются значения переменных в файле и как можно восстановить значения переменных, используемых в предыдущих сеансах?
15. Что такое массив?
16. Какие способы ввода матриц Вы знаете?
17. Каким образом осуществляется доступ к элементам матрицы?
18. Каким критериям должны отвечать матрицы при сложении, вычитании и перемножении?
19. Какие символы предназначены для умножения, транспонирования и возведения в степень матриц?