

Министерство образования и науки РФ  
Федеральное государственное образовательное учреждение  
высшего образования  
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
(ФГБОУ ВО «АмГУ»)

## ПРИКЛАДНЫЕ ПРОГРАММЫ В АВТОМАТИЗАЦИИ

сборник учебно-методических материалов  
для направления подготовки 15.03.04 Автоматизация технологических процессов и производств

Благовещенск, 2017

Печатается по решению  
Редакционно-издательского совета  
Энергетического факультета  
Амурского государственного университета

Составитель: Рыбалев А.Н.

Прикладные программы в автоматизации: сборник учебно-методических материалов для направления подготовки 15.03.04 Автоматизация технологических процессов и производств. – Благовещенск: Амурский гос. ун-т, 2017.

©Амурский государственный университет, 2017  
©Кафедра автоматизации производственных  
процессов и электротехники, 2017  
©Рыбалев А.Н, составитель

## Содержание

Содержание.....	3
ВВЕДЕНИЕ .....	4
1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ .....	5
1. Основные пользовательские интерфейсы системы Matlab .....	5
2. Простейшие векторно-матричные вычисления.....	6
3. Типы данных языка Matlab.....	8
4. Графическая система Matlab .....	10
5. Программирование в Matlab.....	11
6. Аппроксимационные и оптимизационные задачи .....	12
7. Задача на решение линейных дифференциальных уравнений с постоянными коэффициентами и их систем .....	14
7. Задача на численное интегрирование.....	18
8. Построение Simulink-модели системы.....	21
9. Простейшее форматирование документа. Стандарт АмГУ .....	24
10. Форматирование абзацев в Word .....	27
11. Создание списков в Word .....	29
12. Стили и шаблоны в Word .....	30
13. Таблицы в Word.....	32
14. Сноски, ссылки, колонтитулы, оглавление, список литературы в Word.....	34
15. Простейшие вычисления и операции в MathCAD .....	38
16. Преобразование алгебраических выражений в MathCAD .....	41
17. Определение, построение таблиц значений и графиков функций .....	44
19. Действия с матрицами в MathCAD .....	46
20. Решение систем алгебраических уравнений в MathCAD.....	51
21. Решение задач линейного программирования в MathCAD .....	52
22. Аппроксимация эмпирических данных и метод наименьших квадратов в MathCAD. 53	
23. Решение задач математического анализа в MathCAD.....	55
3.МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ . 59	
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	62

## ВВЕДЕНИЕ

Целью дисциплины «Прикладные программы в автоматизации» является приобретение студентами навыков работы с математическими пакетами программ для персонального компьютера и освоении ими методов организации вычислений и обработки информации. В качестве основного математического пакета выбран пакет Matlab фирмы MathWorks Inc., как наиболее приспособленный для решения задач изучаемых дисциплин специальности 15.03.04, – таких как «Теория автоматического управления» «Моделирование систем», «Автоматизация технологических процессов и производств» и др.

Дополнительно изучается физико-математический пакет Mathcad (от компании Mathsoft) для использования его возможностей при выполнении расчётов в курсах инженерных дисциплин. Осваиваются средства автоматизации форматирования текста в редакторе Microsoft Word.

Задачи дисциплины:

получение навыков проведения вычислений с помощью специальных языков высокого уровня;

освоение основ визуализации информации;

изучение современных подходов к моделированию динамических систем;

получение навыков аналитических и численных расчётов в математической программе;

изучение структуры сложного текстового документа формата doc, стили, шаблоны, ссылки, оглавление и пр.

освоение способов быстрого форматирования текстовых документов doc.

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

1) Знать:

основные пользовательские интерфейсы математического пакета;

основные типы данных языка программирования технических расчетов;

принципы организации графической системы математического пакета;

принципы верстки сложных по структуре документов с использованием текстового процессора Word;

возможности и ограничения в использовании математического пакета для расчётов.

2) Уметь:

проводить расчеты в векторно-матричной форме;

проводить расчёты с использованием единиц измерений;

создавать в математическом пакете документы, представляющие собой проводимые расчёты, текстовые и графические пояснения;

визуализировать результаты расчетов с использованием двух- и трехмерной графики;

составлять несложные программы с использованием управляющих конструкций языка программирования и подпрограмм;

аппроксимировать экспериментально полученные зависимости полиномами;

решать задачи поиска экстремума функций одной и нескольких переменных;

решать дифференциальные уравнения с постоянными коэффициентами и их системы с использованием специализированного пакета;

решать задачи на численное интегрирование систем дифференциальных уравнений;

строить простейшие имитационные модели;

создавать в текстовом процессоре Word документы согласно стандарту АмГУ.

3) Владеть навыками проведения расчетов и визуализации их результатов в пакете Matlab. Владеть навыками проведения расчетов, визуализации их результатов и создании отчетов в пакете MathCAD. Владеть навыками создания сложных документов в текстовом процессоре Word.

# 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ

## 1. Основные пользовательские интерфейсы системы Matlab

Основными интерфейсами пользователя Matlab являются:

*Окно команд.*

*Редактор-отладчик.*

*Средство просмотра рабочей области.*

*Средство просмотра и редактирования путей доступа.*

*Окно помощи.*

*Окно команд* – основное окно, в котором работает пользователь. Его закрытие приводит к завершению работы с системой и закрытию остальных окон.

Окно команд обеспечивает

- организацию вычислений;
- доступ ко всем остальным интерфейсам.

Организация вычислений предполагает ввод пользователем выражений в командной строке и вывод результатов в этом же окне (ниже) (результат не выводится, если после выражения стоит ';'') либо в графических окнах.

В качестве выражений могут быть любые допустимые языком Matlab операторы, вызов на исполнение функций и script-файлов (файлов-сценариев).

Script-файлы вызываются на исполнение путем ввода имени файла в командной строке. Например, вызов script-файла myscript.m:

```
>> myscript
```

Файлы-функции вызываются аналогично, но с учетом передачи параметров. Например, вызов функции myfun.m с параметрами 1.4 и 2.5:

```
>> myfun(1.4, 2.5)
```

Кроме того, вызов функции может участвовать в выражениях, например

```
>> a = 3.5/myfun(b+55, 11)
```

Доступ ко всем остальным интерфейсам Matlab осуществляется как из командной строки (исполнением специальных команд типа edit, help), так и с помощью панели инструментов.

*Редактор-отладчик* – средства для набора, редактирования и отладки script-файлов и файлов-функций.

Запускается из Окна команд с помощью кнопок панели инструментов или по команде edit, edit <имя файла> в командной строке.

Позволяет:

- редактировать файлы программ (с выделением ключевых слов и синтаксических ошибок);
- просматривать значения переменных, по их идентификаторам в текстах программ (с помощью мыши);
- запускать программы на исполнение (Пункт "Run" меню "Инструменты");
- производить отладку программ (останов в контрольных точках, по шаговый режим и т.д.).

К сожалению, редактор-отладчик не выводит сообщения об ошибках (они выводятся в Окне команд)

*Средство просмотра рабочей области* – позволяет просмотреть список переменных рабочей области с указанием:

- имен;
- размеров (все переменные в Matlab являются массивами);
- размеров занимаемой памяти в байтах;

– класса (типа).

Ту же информацию можно получить по команде `whos`.

Кроме того, средство позволяет удобно для пользователя представить и отредактировать численные переменные (двойной щелчок мыши по имени переменной - вызов редактора-отладчика).

*Средство просмотра и редактирования путей доступа* – предназначено для просмотра и редактирования путей доступа – списка каталогов (папок), из которых Matlab извлекает файлы на выполнение.

Средство позволяет:

- добавлять и удалять каталоги из списка;
- изменять последовательность поиска.

Доступно из панели инструментов Окна команд или по команде `editpath` в командной строке.

*Окно помощи* – представляет удобный доступ к заголовочным частям библиотечных функций (комментариям).

Доступно из панели инструментов Окна команд или по команде `helpwin` в командной строке.

Справочные и управляющие команды Matlab.

Все они (либо файлы программ, либо файлы помощи для встроенных команд) располагаются в `\Matlab\general`.

Основными являются следующие.

Получение помощи:

`help <имя файла>` – получение помощи (комментария по файлу);

`help <имя каталога>` – получение помощи по функциям каталога (пакета);

`demo` - запуск демонстрационных программ.

Управление переменными:

`who, whos` – информация о переменных рабочей области;

`clear` – удаление переменных и функций из памяти;

`load, save` – загрузка переменных из файлов и запись их в файлы (используются как файлы спец. формата типа MAT, так и обыкновенные текстовые).

Информация по файлам:

`type` – вывод файла на экран;

`what` – список файлов каталога с сортировкой по типу;

`which` – определение пути доступа к файлу;

`edit` – запуск редактора-отладчика.

## 2. Простейшие векторно-матричные вычисления

Элементарные действия над матрицами в Matlab

Ввод матриц (`\MATLAB\elmat`):

матрица-строка

`A = [1 2 3]`

матрица-столбец

`B = [1; 2; 3]`

прямоугольная матрица

`C = [1 2 3; 4 5 6]`

единичная диагональная матрица размером 3 на 3

`E = eye(3)`

диагональная матрица с заданной диагональю

`D = diag([1 2 3])`

то же самое, с учетом, что  $A = [1 \ 2 \ 3]$

$D = \text{diag}(A)$

матрица с нулевыми элементами 3 на 4

$Z = \text{zero}(3, 4)$

Определение размеров матриц (`\MATLAB\elmat`).

Вычисление размеров:

`size(A)`

`[m,n] = size(C)`

Вычисление "длины"

`length(B)`

`n = length(C)`

Поиск максимального, минимального элементов (`\MATLAB\datafun`).

Введем, например, следующую матрицу

$M = [2 \ 4 \ 6 \ 3; \ 6 \ 8 \ 3 \ 0]$

тогда

`max(M)`

`[n,i] = max(M)`

или, допустим, имеем еще

$N = [1 \ 4 \ 3 \ 5; \ 8 \ 4 \ 6 \ 0]$

тогда

`max(M,N)`

Аналогично работает функция `min`.

## 2.2. Математические операции

Введем матрицы

$A = [1 \ 2; \ 3 \ 4]$

$B = [2 \ 3; \ 0 \ 1]$

Умножение:

матричное

$A*B$

поэлементное

$A.*B$

сложение

$A+B$

$A+3$

Возведение в степень

$A^2$

$A.^2$

Транспонирование

$A'$

Логические операции и функции

$A==B$

$A<B$

$A>3$

`all(A>3)`

`any(A>3)`

## 2.3. Линейная алгебра (`\MATLAB\matfun`)

Решим систему линейных уравнений

$$3x_1 + 2x_2 + 4x_3 + 6x_4 = 6$$

$$x_1 - 3x_2 + 2x_3 - 3x_4 = 4$$

$$-2x_1 + 9x_2 - 3x_3 + x_4 = -3$$

$$2x_1 + 2x_2 + 7x_3 + 5x_4 = 1$$

В матричном виде система может быть представлена как

$$AX = B$$

Введем матрицы A и B

$$A = [3 \ 2 \ 4 \ 6; \ 1 \ -3 \ 2 \ -3; \ -2 \ 9 \ -3 \ 1; \ 2 \ 2 \ 7 \ 5]$$

$$B = [6; \ 4; \ -3; \ 1]$$

Решение

$$X = A \setminus B$$

Определитель матрицы A:

$$\det(A)$$

Собственные числа матрицы A:

$$\text{eig}(A)$$

Характеристический полином матрицы A:

$$p = \text{poly}(A)$$

Его корни (\MATLAB\polyfun)

$$\text{roots}(p)$$

### 3. Типы данных языка Matlab

Основные типы языка Matlab:

`double` – массивы, в том числе многомерные, чисел удвоенной точности, в том числе комплексных;

`char` – массивы символов;

`cell` – массивы ячеек, могут содержать данные различных типов;

`struct` – структуры.

*Тип double*

Правила ввода массивов и работы с ними были рассмотрены выше. В основном они справедливы и для многомерных массивов.

Пример:

```
% Создадим нулевой массив 2 на 3 на 4
```

```
M = zeros(2, 3, 4)
```

```
% Заполним четвертую «страницу» массива
```

```
M(:, :, 4) = [1 2 3; 4 5 6]
```

```
% Обратимся к столбцу №3 четвертой «страницы»:
```

```
M(:, 3, 4)
```

Специфические функции многомерных массивов:

`cat` – объединение массивов.

`ndims` – вычисление числа размерностей.

`ndgrid` – формирование массивов в виде расчетной сетки.

`permute` – перестановка размерностей.

`ipermute` – обратная перестановка размерностей.

`shiftdim` – сдвиг размерности.

`squeeze` – удаление единичных размерностей.

В Matlab нет специального типа данных для представления комплексных чисел, – все числа по умолчанию считаются комплексными. Действительные числа рассматриваются как комплексные с нулевой мнимой частью.

При запуске программы автоматически инициализируются мнимыми единицами две переменные: `i` и `j`. В дальнейшем они могут использоваться в этом качестве, а могут быть и переопределены. Определение любой переменной значением мнимой единицы может быть выполнено следующим образом:

$$i = \text{sqrt}(-1)$$

С такой переменной можно задать любое комплексное число:

$$a = 2+i, \quad b = 1+2*i, \quad c = \text{exp}(4+2*i)$$

Альтернативный способ задания комплексных чисел не использует специальных переменных (и знаков умножения) и поэтому является более универсальным:

```
a = 2+1i, b = 1+2j, c = exp(4+2i)
```

В данном случае *i* и *j* – это не переменные, а просто символы.

Специфические операции для работы с комплексными числами:

`abs` – вычисление абсолютной величины (длины вектора) для векторного представления комплексного числа;

`angle` – вычисление фазового угла вектора для векторного представления комплексного числа;

`conj` – вычисление комплексного сопряжения;

`imag` – вычисление мнимой части комплексной переменной;

`real` – вычисление действительной части комплексной переменной.

### *Тип char*

Для задания переменных такого типа используются одинарные кавычки:

```
S = 'String'.
```

В Matlab строковые переменные рассматриваются в виде векторов-строк, содержащих ASCII - коды символов. Поэтому над ними разрешаются все действия, разрешенные для соответствующих «численных» векторов. Однако способ вывода результатов на экран (в численном или строковом виде) зависит от произведенной операции. Приведем наиболее важные функции Matlab для работы со строками:

`abs` – преобразует строку в вектор ASCII - кодов символов;

`setstr` – преобразует вектор ASCII - кодов символов в строку;

`sprintf` – преобразует число в строку-представление заданного формата;

`sscanf` – читает строку-представление числа заданного формата.

Для ознакомления с остальными функциями введите

```
help strfun
```

### *Тип cell*

Cell – массивы ячеек, которые могут содержать в себе данные любых типов. Для работы с ними определена операция `{}`.

Пример:

```
A = [1 2; 3 4] %матрица
```

```
S = 'abc' %строка
```

```
C = {A, S} % массив ячеек
```

```
% Обратимся к матрице из массива ячеек
```

```
C{1}
```

```
% Обратимся к элементу (1,2) матрицы
```

```
C{1}(1,2)
```

О специфических функциях для работы с массивами ячеек можно узнать, введя

```
help datatypes
```

и обратившись к разделу

```
Cell array functions
```

### *Тип struct*

Переменные типа `struct` представляют собой структуры – собрания именованных полей различными типами данных. Для структур определена операция «.».

Пример:

```
A = [1 2; 3 4] %матрица
```

```
S = 'abc' %строка
```

```
% Введем структуру
```

```
ST.Array = A
```

```
ST.String = S
```

```
% Обратимся к матрице из структуры
```

```
ST.Array
% Обратимся к элементу (1,2) матрицы
ST.Array(1,2)
```

О специфических функциях для работы с массивами ячеек можно узнать, введя `help datatypes`

и обратившись к разделу `Structure functions`

Массивы ячеек и структуры используются очень широко при обмене данными с функциями.

#### 4. Графическая система Matlab

В Matlab используется так называемая дескрипторная графика: изображение формируется из объектов, каждый из которых однозначно адресуется дескриптором – уникальным числом. Основные типы графических объектов:

`figures` – фигуры (графические окна), имеют целочисленные дескрипторы, равные номерам фигур;

`axes` – оси, имеют дробные дескрипторы;

`lines` – линии, имеют дробные дескрипторы;

`surfaces` – поверхности, имеют дробные дескрипторы.

Каждый графический объект характеризуется определенным для его типа набором свойств, управляющим его отображением. Для работы со свойствами используются функции `get` и `set`.

Пример:

```
%Создание фигуры 3
figure(3)
% Сделаем белым фон
set(3, 'Color', [0 0 0])
a = axes;
% Узнаем позицию осей
get(a, 'Position')
% Зададим новую позицию и размеры осей
set(a, 'Position', [.5 .5 .4 .4])
% Нанесем на оси линию
l = line([1 2], [3 4])
% Сделаем ее красной
set(l, 'Color', 'red')
```

Основные свойства графических объектов доступны непосредственно в графических окнах.

В основном пользователю системы нет необходимости напрямую работать с графическими объектами, поскольку система имеет достаточно большой набор графических функций высокого уровня. Ниже рассмотрены наиболее часто используемые функции.

*Двумерный график (`\matlab\graph2d`)*

Расчет графика функции  $y = 2\sin(3x^2)$  на интервале от 0 до 3 с шагом 0.01:

```
x = 0:.01:3;
y = 2*sin(3*x.^2)
```

Построение графика и сетки

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

Закроем все графические окна

```
close all
```

*Управление окнами (`\matlab\graphics`)*

Допустим, мы хотим одновременно вывести в разные окна графики  $\sin(x)$ ,  $\cos(x)$  и  $y$ . Для каждого графика создаем новое окно

```
figure, plot(x, sin(x)), grid, title('sin(x)')
```

```
figure, plot(x,cos(x)),grid, title('cos(x)')
```

```
figure, plot(x,y),grid, title('y(x)')
```

Теперь закроем окна 2 и 3

```
close([2 3])
```

выберем окно 1

```
figure(1)
```

добавим туда график cos(x)

```
hold on
```

```
plot(x,cos(x),'r')
```

```
hold off
```

*Трехмерный график (\matlab\graph2d)*

Построим поверхность  $Z = \sin^2(x) + \cos^2(y)$  над прямоугольником  $-5 < x < 5, -4 < y < 4$ :

```
x = -5:.1:5;
```

```
y = -4:.1:4;
```

```
[X,Y] = meshgrid(x,y);
```

```
Z = sin(X).^2 + cos(Y).^2;
```

```
mesh(x,y,Z)
```

Полный список графических функций можно получить, введя

```
help graphics (дескрипторная графика)
```

```
help graph2d (двумерные графики)
```

```
help graph3d (трехмерные графики)
```

Кроме того, полезно ознакомиться с демонстрационными примерами

```
demo (раздел Graphics)
```

## 5. Программирование в Matlab

### Работа с функциями

Пользователь Matlab может создавать свои функции для решения определенных задач. В отличие от Script-файлов, функции имеют некоторые элементы оформления и по умолчанию оперируют локальными переменными.

Например, создадим функцию minsqr для аппроксимации опытных данных многочленом типа  $Z = a_1 + a_2XY$ , использующую метод наименьших квадратов.

Входными аргументами такой функции естественно принять вектора  $x$  и  $y$  (представляющие собой, например, известные значения двух входных воздействий на некоторый объект) и матрицу  $Z$  экспериментальных данных (реакция объекта).

Выходным аргументом функции будет вектор коэффициентов  $a = [a_1; a_2]$ .

```
function a = minsqr(x,y,Z)
```

```
% Аппроксимация опытных данных многочленом
```

```
% типа  $Z = a_1 + a_2 * X * Y$ 
```

```
% с использованием метода наименьших квадратов.
```

```
[X,Y] = meshgrid(x,y);
```

```
[m,n] = size(X);
```

```
A = zeros(2,2); B = zeros(2,1);
```

```
A(1) = m*n;
```

```
A(2) = sum(sum(X.*Y));
```

```
A(3) = A(2);
```

```
A(4) = sum(sum(X.^2.*Y.^2));
```

```
B(1) = sum(sum(Z));
```

```
B(2) = sum(sum(Z.*X.*Y));
```

```
a = A\B;
```

Текст функции помещается в одноименный файл.

Для проверки работоспособности функции сформируем матрицы X и Y и рассчитаем матрицу Z, приняв, например, что  $Z=2+3XY$ .

```
x = 0:.1:1; y = 2:.1:6;
[X,Y] = meshgrid(x,y);
z= 2+3*X.*Y;
```

Если функция работает правильно, результатом ее вызова, очевидно, будет вектор [2;3]:

```
a = minsqr(x,y,z)
```

*Функции inline (\matlab\funfun\@inline)*

В отдельных случаях функции могут оформляться не в виде отдельных файлов, а путем формирования т.н. объектов inline.

С точки зрения иерархии типов данных Matlab объекты inline представляют собой пользовательский класс данных (user), несмотря на то, что их поддержка обеспечена самими производителями программы Matlab. Для данного класса определена структура данных и переопределены некоторые функции Matlab (например, feval). Подробнее см.

```
help inline
methods inline
```

При формировании объектов inline необходимо явно задать строку вычисления функции, например:

```
y = inline('sin(x)*cos(x)')
```

Если входных аргументов более одного, необходимо также указать их имена:

```
z = inline('sin(x)*cos(y)', 'x', 'y')
```

В большинстве случаев вызов функции inline не отличается от вызова обычных функций-файлов:

```
y(pi/3)
z(pi/3,pi/6)
```

Необходимо лишь помнить, что, например, в данном случае y и z – имена функций.

Управляющие конструкции

Как и во всех языках высокого уровня в Matlab имеется полный набор управляющих конструкций:

Условные переходы

```
if...elseif...else...end;
```

циклы

```
while...end,
```

```
for...end;
```

переключатель

```
switch...case...otherwise...end
```

О правилах их использования можно узнать из справочной системы, введя, например

```
help if
```

## 6. Аппроксимационные и оптимизационные задачи

Аппроксимация зависимости  $y(x)$  полиномом (метод наименьших квадратов). Используется функция

```
p = polyfit(x,y,n)
```

где x,y - вектора (экспериментальные данные), n - порядок полинома, p - вектор коэффициентов полинома, начиная с коэффициента при старшей степени (\matlab\polyfun).

Пример:

Рассчитаем некоторую функцию-полином

```
x = 0:.1:10;
```

```
y = 3*x.^2 + 2*x + 1;
```

Аппроксимируем данные полиномами первого и второго порядков:

```
p1 = polyfit(x, y, 1)
```

```
p2 = polyfit(x, y, 2)
```

Вычислим значения полиномов в точках  $x$  и построим соответствующие графики:

```
y1 = polyval(p1, x);
```

```
y2 = polyval(p2, x);
```

```
plot(x, y1, x, y2), grid
```

6.2. Минимизация функции одной переменной

```
xmin=fminbnd(N_of_F, x1, x2)
```

```
xmin=fminbnd(N_of_F, x1, x2, options)
```

где  $N\_of\_F$  - имя минимизируемой функции,  $x1, x2$  - границы по  $x$ , в пределах которых ищется минимум,  $options$  - строка опций процедуры.

Значения опций по умолчанию можно получить с помощью функции `foptions`, см.

```
help foptions
```

Изменить опции, можно, например, так (зададим "терминальную" точность вычислений  $10^{-5}$ ):

```
options = foptions
```

```
options(2) = 1e-5
```

```
(\matlab\funfun)
```

Пример

Определим минимум функции  $y = \sin(x)/(5x^2-x)$  на интервале  $x = 3..6$ .

Зададим функцию и построим ее график на интервале

```
y = inline('sin(x) ./ (5.*x.^2-x)')
```

```
x = 3:.05:6;
```

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

Определим координаты минимума

```
xmin = fminbnd(y, 3, 6)
```

Минимизация функции нескольких переменных (симплекс-метод)

```
xmin= fminsearch (N_of_F, x0)
```

```
xmin= fminsearch (N_of_F, x0, options),
```

```
xmin= fminsearch (N_of_F, x0, options, [], P1, P2, ...),
```

где  $N\_of\_F$  - имя минимизируемой функции,  $x0$  - вектор начальных приближений,  $P1, P2, \dots$  доп. параметры для минимизируемой функции.

Функция `fmins` требует описания минимизируемой функции в виде

```
y = N_of_F(x),
```

```
y = N_of_F(x, P1, P2, ...),
```

где  $x$  - вектор аргументов, по которым ищется минимум (`\matlab\funfun`).

Пример:

Найти минимум функции  $y = (x_1-1)^2 + (x_2+2)^2$  (естественно, он находится в точке  $x_1=1, x_2=-2$ ).

Опишем функцию как объект `inline`:

```
y = inline('(x(1)-1)^2+(x(2)+2)^2')
```

Найдем координаты минимума, начиная поиск с точки  $x_1=0, x_2=0$ .

```
xmin = fminsearch (y, [0 0])
```

Функция `fminsearch` удобна для аппроксимации данных различными аналитическими зависимостями. В этом случае минимизировать необходимо «разницу» между экспериментальными значениями неизвестной функции и ее аналитически вычисленными значениями. При этом аналитическая зависимость может иметь любой вид (в отличии, например, от метода наименьших квадратов).

Пример:

Аппроксимировать данные многочленом  $a_1 + a_2x$ .

Минимизируемую функцию можно разместить в файле `eg`:

```
function err = er(a,x,y,z)
err = sum(sum((z-a(1)-a(2)*x.*y).^2));
```

либо задать ее как объект inline:

```
er=inline('sum(sum((z-a(1)-...
          a(2)*x.*y).^2))','a','x','y','z')
```

Минимизируемая функция представляет собой сумму квадратов разности между  $z$  и  $a_1+a_2x$  по всем точкам. Минимум находится по вектору  $a$ .

Для проверки метода сформируем соответствующие матрицы и найдем  $a_1$  и  $a_2$ , соответствующие минимуму функции er:

```
x = 0:.1:1; y = 2:.1:6;
[X,Y] = meshgrid(x,y);
Z= 2+3*X.*Y;
```

Найдем  $a_1$  и  $a_2$ :

```
a = fminsearch('er',[0 0],[[]],[[]],X,Y,Z)
```

– если er – файл с функцией,

```
a = fmins(er,[0 0],[[]],[[]],X,Y,Z)
```

– если er – inline-объект

Более широкие возможности по решению оптимизационных задач предоставляет пакет Optimization Toolbox. Для обзора рекомендуется запустить демонстрационный файл

```
tutdemo
```

7. Задача на решение линейных дифференциальных уравнений с постоянными коэффициентами и их систем

Для решения линейных дифференциальных уравнений и их систем могут использоваться функции пакета Control Systems. Данный пакет в основном предназначен для исследования и синтеза линейных систем управления, однако его возможности позволяют решать линейные дифференциальные уравнения и их системы безотносительно проблемам теории управления.

Пакет Control Systems построен на основе объектно-ориентированной технологии и включает несколько классов, представляющих математическое описание линейных объектов. Мы будем использовать классы

tf (transfer function) – передаточные функции;

ss (state space) – описание в пространстве состояний;

Эти классы являются потомками виртуального класса lti (linear time-invariant – линейные системы с постоянными параметрами) и наследуют от него поля (свойства) для описания объектов и методы (функции) для работы с ними.

Объекты tf, ss описываются соответствующими структурами, поля которых заключают свойства объекта. Набор свойств для всех классов можно просмотреть введя:

```
help ltiprop
```

Доступ к полям объектов в Matlab осуществляется с помощью функций set (установить) и get (получить) и путем непосредственного обращения к полю. Однако в большинстве случаев необходимости в этом нет, так как основные поля заполняются и изменяются функциями пакета.

Для детального ознакомления с функциями воспользуйтесь справочной системой:

меню окна управления Matlab;

командами помощи: help control, help 'function\_name';

командой демонстрации: demo.

Большинство функций пакета Control работает со всеми тремя классами объектов, меняя алгоритм своей работы в зависимости от типа объекта. Для создания объектов применяется функции-конструкторы, название которых совпадает с именем класса: `tf`, `ss`.

Передаточные функции

Передаточной функцией системы называют отношение изображения по Лапласу выходной переменной к изображению по Лапласу входной переменной при нулевых начальных условиях:

$$W(p) = \frac{x_{\text{вых}}(p)}{x_{\text{вх}}(p)}$$

при

$$x_{\text{вых}}(0) = \frac{dx_{\text{вых}}}{dt}(0) = \frac{d^2x_{\text{вых}}}{dt^2}(0) = \dots = \frac{d^{n-1}x_{\text{вых}}}{dt^{n-1}}(0) = 0,$$

где  $n$  – порядок дифференциального уравнения, описывающего звено или систему;  $p$  – оператор Лапласа.

Как известно, изображение по Лапласу функции  $f(t)$  находят по формуле:

$$f(p) = \int_0^{\infty} f(t)e^{-pt} dt.$$

Обратное преобразование вычисляется следующим образом:

$$f(t) = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} f(p)e^{pt} dp.$$

Зная передаточную функцию системы и входное воздействие  $x_{\text{вх}}(t)$ , можно определить его (ее) реакцию на это воздействие при нулевых начальных условиях. Для этого нужно, предварительно рассчитав изображение входной величины  $x_{\text{вх}}(p)$ , определить изображение выходной величины

$$x_{\text{вых}}(p) = x_{\text{вх}}(p)W(p)$$

и далее с помощью обратного преобразования (4) вычислить  $x_{\text{вых}}(t)$ .

Линейные звенья и системы описываются дифференциальными уравнениями вида

$$\begin{aligned} a_n \frac{d^n x_{\text{вых}}}{dt^n} + a_{n-1} \frac{d^{n-1} x_{\text{вых}}}{dt^{n-1}} + \dots + a_1 \frac{dx_{\text{вых}}}{dt} + a_0 x_{\text{вых}} = \\ = b_m \frac{d^m x_{\text{вх}}}{dt^m} + b_{m-1} \frac{d^{m-1} x_{\text{вх}}}{dt^{m-1}} + \dots + b_1 \frac{dx_{\text{вх}}}{dt} + b_0 x_{\text{вх}}. \end{aligned}$$

В соответствии с одной из теорем операционного исчисления изображение  $k$ -й производной функции  $f(t)$  при нулевых начальных условиях:

$$\frac{d^k f}{dt^k} \Rightarrow p^k f(p)$$

при

$$f(0) = \frac{df}{dt}(0) = \frac{d^2 f}{dt^2}(0) = \dots = \frac{d^{k-1} f}{dt^{k-1}}(0) = 0.$$

Кроме того, известно, что изображение линейной комбинации функций равно линейной комбинации изображений.

Поэтому, преобразовав по Лапласу левую и правую части уравнения, получим:

$$\begin{aligned} a_n p^n x_{\text{вых}}(p) + a_{n-1} p^{n-1} x_{\text{вых}}(p) + \dots + a_1 p x_{\text{вых}}(p) + a_0 x_{\text{вых}}(p) = \\ = b_m p^m x_{\text{вх}}(p) + b_{m-1} p^{m-1} x_{\text{вх}}(p) + \dots + b_1 p x_{\text{вх}}(p) + b_0 x_{\text{вх}}(p), \end{aligned}$$

откуда

$$W(p) = \frac{x_{\text{вых}}(p)}{x_{\text{вх}}(p)} = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0}.$$

Таким образом, зная дифференциальное уравнение линейного звена (системы), можно легко получить его передаточную функцию, и наоборот. С этой точки зрения дифференциальные уравнения и передаточные функции, – эквивалентные понятия.

Рассмотрим теперь дифференциальное уравнение в том виде, в каком оно рассматривается в математике:

$$a_n \frac{d^n y}{dx^n} + a_{n-1} \frac{d^{n-1} y}{dx^{n-1}} + \dots + a_1 \frac{dy}{dx} + a_0 y = F(x).$$

Положив  $x = t$ ,  $y = x_{\text{вых}}$ ,  $F(x) = F(t) = x_{\text{вх}}$ , получим

$$a_n \frac{d^n x_{\text{вых}}}{dt^n} + a_{n-1} \frac{d^{n-1} x_{\text{вых}}}{dt^{n-1}} + \dots + a_1 \frac{dx_{\text{вых}}}{dt} + a_0 x_{\text{вых}} = x_{\text{вх}}.$$

Передаточная функция, составленная по уравнению имеет вид

$$W(p) = \frac{x_{\text{вых}}(p)}{x_{\text{вх}}(p)} = \frac{1}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0}.$$

Ввести объект – передаточную функцию в Matlab можно с помощью функции `tf`, например, для дифференциального уравнения

$$\frac{d^3 y}{dx^3} + 2 \frac{d^2 y}{dx^2} + 2 \frac{dy}{dx} + y = 0,5x + \sin^2(x) = F(x)$$

```
sys = tf(1, [1 2 2 1])
```

Решим уравнение на интервале  $x = 0 \dots 10$  при нулевых начальных условиях:

1) зададим вектор  $x$ , например, от нуля до десяти с шагом 0,1:

```
x = 0:.1:10
```

2) рассчитаем вектор значений правой части:

```
f = .5*x + sin(x).^2
```

3) рассчитаем вектор  $y$ :

```
y = lsim(sys, f, x)
```

4) построим график  $y(x)$

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

Смотрите также

```
help lsim
```

Для решения дифференциального уравнения при ненулевых начальных условиях, оно должно быть преобразовано в систему уравнений первого порядка. Например, для уравнения это можно сделать, положив

$$y_1 = y, y_2 = \frac{dy}{dx}, y_3 = \frac{d^2 y}{dx^2},$$

тогда получим систему

$$\begin{cases} \frac{dy_1}{dt} = y_2; \\ \frac{dy_2}{dt} = y_3; \\ \frac{dy_3}{dt} = F(x) - y_1 - 2y_2 - 2y_3. \end{cases}$$

с начальными условиями

$$y_1(0) = y(0), y_2(0) = \frac{dy}{dx}(0), y_3(0) = \frac{d^2y}{dx^2}(0)$$

Ее решение производится с помощью ss-модели.

Описание систем в пространстве состояний

State Space-объект описывается с помощью матриц уравнений состояний и выхода систему с несколькими входами и выходами

Любой объект (система) описывается в пространстве состояний уравнениями вида:

$$\begin{aligned} \dot{X} &= F_1(X, U), \\ Y &= F_2(X, U), \end{aligned}$$

где  $X$  – вектор независимых координат объекта (системы), однозначно описывающих его (ее) состояние (вектор переменных состояния);  $Y$  – вектор выходных (измеряемых) величин;  $U$  – вектор входных воздействий. В общем случае вектор  $U$  может включать как управляющие (задающие), так и возмущающие воздействия, однако часто возмущения «выносятся» в отдельный вектор.

Для линейных объектов и систем уравнения принимают вид:

$$\begin{aligned} \dot{X} &= AX + BU, \\ Y &= CX + DU, \end{aligned}$$

где  $A$  – квадратная матрица состояний;  $B$  – матрица управления;  $C$  – матрица выхода;  $D$  – так называемая матрица «прямого обхода» (управление  $U$  как бы «в обход» внутренних состояний непосредственно действует на выход  $Y$ ). Для реальных объектов и систем практически всегда  $D = 0$ .

Размеры матриц  $A$ ,  $B$ ,  $C$  и  $D$  определяются размерностями векторов состояния, управления и выхода. Если  $X \in R^n$ ,  $U \in R^m$ ,  $Y \in R^\ell$ , то матрицы имеют следующие размеры:  $A - [n \times n]$ ,  $B - [n \times m]$ ,  $C - [\ell \times n]$ ,  $D - [\ell \times m]$ .

Пусть, например, имеется система уравнений

$$\begin{cases} \frac{dx_1}{dt} = -2x_1 + x_2 - x_3; \\ \frac{dx_2}{dt} = -x_1 - x_2 + 2t; \\ \frac{dx_3}{dt} = -3x_1 - x_3 + 2\sin(t); \end{cases}$$

Необходимо найти функции  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  на интервале  $t = 0 \dots 10$  при начальных условиях  $x_1(0)=1$ ,  $x_2(0)=0$ ,  $x_3(0)=-1$ .

Для приведения к нужной форме представим систему в виде

$$\begin{cases} \frac{dx_1}{dt} = -2x_1 + x_2 - x_3; \\ \frac{dx_2}{dt} = -x_1 - x_2 + u_1; \\ \frac{dx_3}{dt} = -3x_1 - x_3 + u_2; \end{cases}$$

$$y_1 = x_1; y_2 = x_2; y_3 = x_3.$$

Система – уравнения состояний, три равенства – уравнения выхода.

В матричном виде уравнения:

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \begin{bmatrix} -2 & 1 & -1 \\ -1 & -1 & 0 \\ -3 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.)$$

Зададим матрицы описания:

A = [-2 1 -1; -1 -1 0; -3 0 -1]

B = [0 0; 1 0; 0 1]

C = eye(3)

D = zeros(3, 2)

Создадим ss – объект:

sys1 = ss(A, B, C, D)

Сформируем  $u_1(t)$  и  $u_2(t)$  и объединим их в вектор-столбец U

t = 0:.1:10

u1 = 2\*t;

u2 = 2\*sin(t)

U = [u1;u2]

Сформируем начальные условия

x0 = [1;0;-1]

Найдем  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$ :

X = lsim(sys1,U,t,x0)

Смотрите

Построим графики функций:

plot(t,X), grid

## 7. Задача на численное интегрирование

Для решения систем обыкновенных дифференциальных уравнений в Matlab имеется ряд функций-«решателей»: ode45, ode23, ode113, ode23t, ode15s, ode23s, ode23tb, которые различаются точностью вычислений и возможностями.

Часть из них предназначена для решения нежестких систем, часть - для решения жестких систем. (Жесткими системами называются системы, у которых незначительное изменение переменных вызывает значительное изменение их производных. В частности это системы с разрывными правыми частями).

Подробнее о решателях можно узнать по

help funfun

Все функции-решатели предназначены для решения проблемы

$dx/dt = F(t,X)$  – т.е. системы обыкновенных дифференциальных уравнений в форме Коши.

Некоторые из них (ode23t,ode15s,ode23s,ode23tb), могут также решать задачу

$Mdx/dt = F(t,X)$ ,

где M – так называемая «массовая матрица» (квадратная).

Кроме того, функции ode15s,ode23t,ode23tb решают ту же задачу и в случае нестационарной матрицы M:

$M(t)dx/dt = F(t,X)$

Описание системы диф. уравнений должно быть помещено в отдельную функцию. Минимальные требования к оформлению такой функции состоят в том, что она, принимая в качестве входных аргументов значение независимой переменной (как, правило, такой переменной является время) и вектор переменных системы, должна вычислить вектор производных переменных системы по независимой переменной (времени). Т.е., коротко говоря, функция описания должна вычислить правые части системы.

Таким образом, минимальный формат описания является следующим:

```
function pX = Name_of_fun(t,X)
```

где  $t$  - независимая переменная,  $X$  - вектор переменных системы,  $pX$  - вектор производных переменных системы по времени.

Вызов решателя для системы, описанной в файле `Name_of_fun`, в простейшем случае производится так:

```
[t,x] = solver('Name_of_fun',tspan,x0)
```

где

`solver` – функция-«решатель», например, `ode45`;

`tspan` - вектор моментов времени, в которые должно быть определено решение. Первый и последний элементы `tspan` определяют начальное и конечное время интегрирования. Промежуточные значения времени могут быть опущены, тогда `tspan = [t0 tfinal]`, а промежуточные значения выбираются решателем самостоятельно;

$x_0$  – вектор начальных значений переменных системы;

$t$  – вектор-столбец расчетных моментов времени;

$x$  – матрица, состоящая из столбцов - решений для каждой переменной системы в расчетные моменты времени.

Пример:

Решить систему дифференциальных уравнений

$$dx_1/dt = -3x_1 + 2x_2x_3;$$

$$dx_2/dt = x_1 - 2x_2 + x_3;$$

$$dx_3/dt = -x_3 + \sin(2t).$$

Создадим функцию описания системы `sys_fun` и поместим ее в одноименный `m`-файл:

```
function px = sys_fun(t,x)
px = [-3*x(1)+2*x(2)*x(3);...
      x(1) - 2*x(2) + x(3);...
      -x(3) + sin(t)];
```

Решим систему на промежутке  $t = 0..10$  с нулевыми начальными условиями. (Система, похоже, нежесткая, - применим `ode45`.)

```
[t,x] = ode45('sys_fun',[0 10],[0 0 0]);
```

Построим графики изменения переменных системы во времени

```
plot(t,x), grid
```

Конечно, в приведенном выше примере предпочтительнее описать функцию как объект `inline` (проделайте самостоятельно).

Дополнительные возможности при решении систем дифференциальных уравнений открываются, если функция-решатель вызывается в форматах

```
[t,x] = solver('Name_of_fun',tspan,x0,options)
```

```
[t,x] = solver('Name_of_fun',tspan,x0,options,p1,p2)
```

где `options` - структура, описывающая опции решателя;  $p_1$ ,  $p_2$  - доп. параметры, передаваемые функции описания системы

Структура `options` создается функцией `odeset`, см.

```
help odeset
```

Другой формат вызова некоторых решателей

```
[t,x,te,xe,ie] = solver('Name_of_fun',tspan,x0,options)
```

позволяет отслеживать события - переход каких-либо величин через ноль:

te - время возникновения события;

xe - значение вектора x во время события;

ie - "индекс" события.

Использование опций, не принятых по умолчанию, позволяет:

– задавать точность вычислений (AbsTol, RelTol);

– задавать функцию вывода результатов решателя на каждом шаге вычислений (OututFcn);

– выводить статистику по окончанию расчета (Stats);

– использовать матрицу Якоби (dF/dx) для ускорения расчета (Jacobian, JConstant, JPattern);

– отслеживать «события» – переход каких-либо величин через ноль, и даже останавливать расчет при возникновении событий (Events);

– решать уравнения с «массовой матрицей» при векторе производных (Mass, MassConstant);

– задавать максимальный и начальный шаг интегрирования, а также максимальный порядок метода (MaxStep, InitialStep и MaxOrder )

и др.

В ряде случаев файл описания системы имеет более сложный вид и использует систему «флагов», подфункции и case-конструкцию. См.

help odefile

odefile.m – шаблон для создания файлов универсального описания систем.

Пример:

Пусть необходимо решить систему уравнений

$$dx_1/dt + e^{-0,02t} dx_2/dt = -2x_1 - 0,05x_1x_2;$$

$$3dx_1/dt - dx_2/dt = -x_2^2 + x_1 + 10.$$

Расчет провести до момента времени, когда координата  $x_2$ , снижаясь, достигнет значения 3, кроме того, фиксировать событие  $x_1 = 1$ .

При расчете использовать матрицу Якоби.

Создадим файл-описание системы, используя odefile.m как шаблон.

```
function varargout = sys_fun(t,x,flag)
switch flag
case ''
    varargout{1} = f(t,x);
case 'init'
    [varargout{1:3}] = init;
case 'jacobian'
    varargout{1} = jacobian(t,x);
case 'mass'
    varargout{1} = mass(t,x);
case 'events'
    [varargout{1:3}] = events(t,x);
otherwise
    error(['Unknown flag '' flag ''.']);
end
```

%---- Вычисление правых частей----

```
function righth_parts = f(t,x)
righth_parts = [-2*x(1) - .05*x(1)*x(2); ...
                -x(2)*x(2) + x(1) + 10];
```

```

%-----Инициализация-----

function [tspan,x0,options] = init
tspan = [0 10];
x0 = [0 0];
options = odeset('Jacobian','on','Mass','on', ...
'MassConstant', 'off', 'Events', 'on', ...
'OutputFcn', 'odeplot', 'Stats', 'on');

% -----Расчет матрицы Якоби-----
function dfdx = jacobian(t,x)
dfdx = [-2-.05*x(2) -.05*x(1);1 -2*x(2)];

% --- Расчет "массовой" матрицы -----
function M = mass(t,x)
M = [1 exp(-.02*t)-2; 3 -1];

%-----Контроль событий-----
function [value,isterminal,direction] = events(t,x)
value = [x(1)-1 x(2)-3] % условие: x1 = 1, x2 = 3
isterminal = [0 1]; % окончить ли расчет? [Нет Да]
direction = [0 -1]; % направление изменения
% координаты ? [Любое Вниз]

Расчет и построение графика
[t,x,te,xe,ie] = ode15s('sys_fun');
Evevts_t_x1_x2_num =[ te, xe, ie]

```

## 8. Построение Simulink-модели системы

Система Simulink – программный продукт семейства Matlab предназначенный для моделирования динамических систем. Она работает в тесном взаимодействии со средой Matlab, обмениваясь с ней данными.

Создание модели системы в Simulink происходит в наиболее простой и естественной форме - путем сборки модели из блоков, представляющих динамические звенья, статические характеристики и другие алгоритмы преобразования сигналов (в том числе и нелинейные). Расчет моделей производится с помощью методов численного интегрирования.

При запуске системы Simulink формируется пустое окно для набора модели системы и окно базовой библиотеки блоков Simulink .

Базовая библиотека Simulink содержит 13 разделов, важнейшими из которых являются

Continues – блоки непрерывной обработки;

Discontinues – блоки, реализующие разрывные функции;

Discrete – блоки дискретной обработки;

Math Operations – математические функции;

Ports & Subsystems – средства создания подсистем;

Signal Routings – управление распространением сигналов;

Sources – источники сигналов;

Sinks – приемники сигналов.

Кроме того, пакеты прикладных программ Matlab добавляют свои разделы блоков.

Модель системы набирается перетаскиванием мышью блоков из разделов библиотеки в окно набора модели и внутри его с последующим соединением блоков стрелками, показывающими распространение сигналов от блока к блоку. Копирование блоков в пределах окна модели также осуществляется перетаскиванием с помощью правой кнопки мыши.

Модель обязательно должна содержать хотя бы один блок-источник сигналов из раздела Sources, например, блок Step (ступенчатое изменение входного сигнала). Для «измерения» и запоминания контролируемых переменных (сигналов) модели применяются блоки-«приемники сигналов» раздела Sinks, чаще всего блок Scope (вывод временного графика изменения сигнала).

Модель динамики исследуемых объектов, процессов и систем набирается с помощью блоков разделов Continues, Discontinues, Discrete, Math Operations, Ports & Subsystems, Signal Routings. В последнем разделе представлены блоки для выполнения вспомогательных операций по соединению блоков, мультиплексированию сигналов и созданию подсистем.

После помещения необходимых блоков в окно набора модели необходимо задать их параметры: значения коэффициентов, временные задержки и т.д. Доступ к параметрам возможен по двойному щелчку левой кнопкой мыши на интересующем блоке. При щелчке правой кнопкой мыши появляется меню, позволяющее работать с буфером обмена Windows, а также настроить внешний вид блока (например, изменить его ориентацию в окне модели).

Линии связи (стрелки), соединяющие блоки модели, могут показывать направление передачи как одного (скалярного) сигнала, так и нескольких (векторного сигнала). В последнем случае блоки, как правило, обрабатывают одинаковым образом все компоненты вектора.

Некоторые блоки, однако, специально предназначены для обработки векторов, например, операции матричного умножения или свертки векторов.

Просмотреть «ширину» линии можно, пометив пункт Signal Dimensions меню Format. Выделить векторные линии жирным, можно, пометив пункт Wide Vector Lines.

После набора модели можно запустить процедуру расчета, предварительно задав ее параметры (при необходимости). Под параметрами в данном случае понимаются название функции-решателя, начальное и конечное время интегрирования, а также параметры обмена данными между системой Simulink и средой Matlab.

В Simulink используется два типа функций-решателей (методов численного интегрирования): с постоянным и с переменным шагом. При выборе первого типа пользователь может задать величину шага, которая и будет определять точность расчета.

Для методов с переменным шагом можно задать максимальное и минимальное значение шага, а также абсолютную и относительную точность вычислений. Методы с переменным шагом условно разделяются на методы решения «нежестких» задач и методы решения «жестких» задач.

В названии последних присутствует буква «s».

Все методы, доступные в Simulink, отличаются друг от друга точностью и скоростью вычислений. В качестве практической рекомендации по их выбору можно привести следующее.

Если модель не содержит разрывных функций (нежесткая задача) следует пользоваться методами ode45 или ode23 (методы Рунге-Кутты 4-5 и 2-3 порядков соответственно), в противном случае (жесткая задача) следует пользоваться методами ode15s и ode23s, специально разработанными для жестких систем.

И, наконец, если в модели возможны переключения с бесконечной теоретически скоростью (скользящие режимы), единственным решением будет применение методов с постоянным шагом, так как другие методы останавливают вычисления, бесконечно уменьшая шаг.

Подробнее о методах см. справочную систему Matlab.

Обмен данными между системой Simulink и средой Matlab позволяет формировать «внешнее управление» моделью из среды и выводить расчетные данные в среду для их дальнейшей обработки. При этом необходимо задать имена соответствующих переменных (вкладка Workspace I/O) и, в случае необходимости, режим внешнего управления, пометив пункт External меню Simulation

Запуск, приостановка расчета и досрочная остановка осуществляются с помощью кнопок панели инструментов окна модели. Просмотреть графики изменения переменных в окнах блоков Scope можно как после окончания процедуры вычисления, так и во время ее выполнения.

Модели и библиотеки Simulink хранятся в текстовых файлах специального формата с расширением .mdl. Запуск системы и открытие окна модели производится путем ввода имени модели в командной строке Matlab.

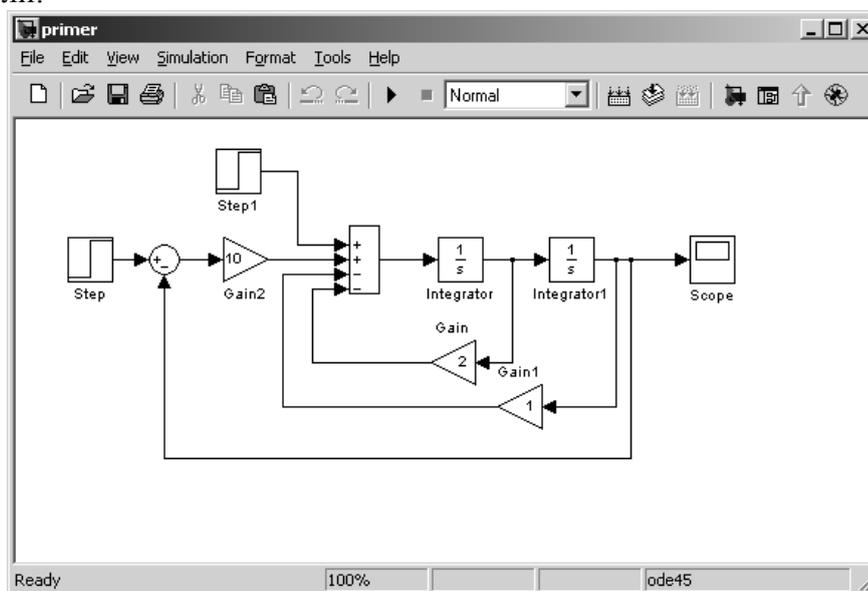
В качестве упражнения постройте модель системы

$$\begin{cases} \dot{x}_1 = -2x_1 + k(y_{зад} - x_1) + f, \\ \dot{x}_2 = x_1. \end{cases}$$

$$y = x_1.$$

и осуществите расчет ее реакции на ступенчатое изменение сначала сигнала  $u_{зад}$ , а потом сигнала  $f$ . Обратите внимание, что для этого необходимо настроить параметры блоков Step.

Окно модели:



Расчет производился при следующих параметрах:

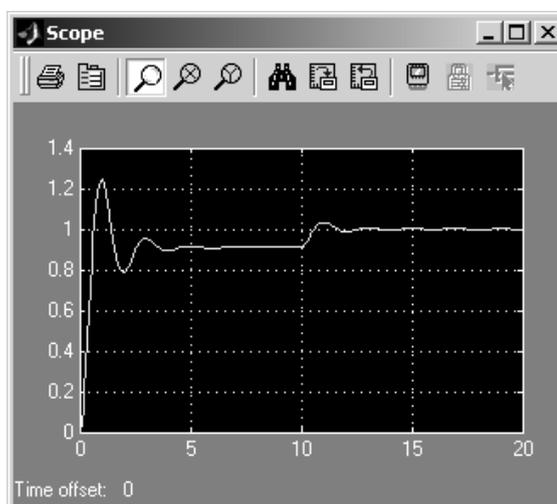
Step: Step time = 0, Initial Value = 0, Final value = 1;

Step1: Step time = 10, Initial Value = 0, Final value = 1;

Simulation parameters: Start time = 0, Stop time = 20, Solver Type – ode45,

Relative tolerance =  $1e-6$ .

Результаты расчета:



## 9. Простейшее форматирование документа. Стандарт АмГУ.

Форматирование документа – это изменение его внешнего вида. WORD обеспечивает форматирование документов на пяти различных уровнях:

на уровне символов (изменение гарнитуры, начертания, размера и цвета шрифта, межбуквенного интервала в слове, анимации и т.д.);

на уровне абзацев (выравнивание по левому краю, по правому краю, по центру и по ширине; отступы справа и слева; отступ первой строки; отступы до и после абзаца; межстрочный интервал, управление разбивкой на страницы и т.д.);

на уровне страниц (параметры страниц, ориентация страниц, рамка, колонтитулы первой страницы, четных и нечетных страниц и т.д.);

на уровне разделов (формирование разделов со следующей страницы или на текущей странице, разбиение текста на колонки и т.д.);

на уровне документа (номера страниц, оглавление и т.д.).

Абзац – это фрагмент текста, который заканчивается непечатаемым символом или фрагмент текста, процесс ввода которого заканчивается нажатием на клавишу ввода Enter. К абзацу относятся: абзац основного текста, заголовки, оглавление, списки (нумерованные и маркированные) и т.д.

Для форматирования символов применяется диалоговое окно Шрифт (Формат / Шрифт) или команды на панели инструментов форматирования. Форматирование абзацев осуществляется с помощью окна диалога Абзац, которое открывается командой Формат / Абзац.

Посмотреть форматирование выделенного текста можно в области задач, выполнив команду Формат / Показать форматирование.

### *Стили*

Стили предназначены для внешнего оформления документа и его абзацев, т.е. стили используются для форматирования документа. Стил – это набор формирующих команд, сохраняемый под уникальным именем для многократного использования. Форматирование текста с помощью стили значительно быстрее, чем форматировать вручную каждый элемент текста, так как одна команда (стиль) автоматически формирует группу параметров текста.

Существует три основных типа стилей:

Стил символа содержит параметры форматирования символов, включая шрифт, размер, начертание, положение и интервалы.

Стил абзаца содержит параметры форматирования абзацев, такие как междустрочные интервалы, отступы, выравнивание и позиции табуляции. Стили абзацев также могут содержать стили или параметры форматирования символов. Большинство стилей, используемых в Word, являются стилями абзацев.

Стил таблицы содержит параметры форматирования таблиц (при вставке таблицы, ей по умолчанию назначается стил – сетка таблицы).

При создании нового документа на базе шаблона Обычный, он получает копию набора стилей из базового набора стилей общего назначения (из встроенных стилей): Обычный, Заголовки 1, 2, 3.

После того как ввод текста в документ завершен, и текст отредактирован, целесообразно воспользоваться командой Автоформат для изменения внешнего вида всего документа.

При автоматическом форматировании документа каждому абзацу назначается один из стилей Word. Например, заголовку может быть назначен стил Заголовков 1 или Заголовков 2, абзацу основного текста – стил Основной текст, а абзацу, входящему в список – стил Список. Примечаниям назначается стил Текст Примечание, верхним колонтитулам – стил Верхний колонтитул, а номерам страниц – стил Номер страницы и т.д.

Назначение абзацам документа стандартных стилей Word обеспечивает быстрое изменение внешнего вида созданного документа за счет применения стилей из библиотеки стилей. Встроен-

ные стили обеспечивают форматирование заголовков различных уровней и позволяют просматривать иерархическую структуру документа в режиме структура, а также быстро строить оглавление.

Кроме того, если для большинства абзацев документа используется стиль Основной текст, а не Обычный, вы сможете легко переформатировать только стиль основного текста, не затрагивая при этом стиль остального текста.

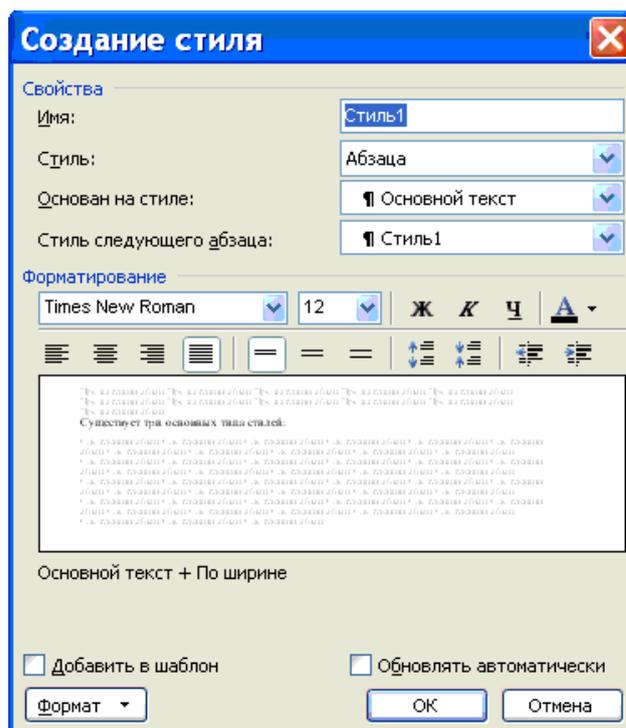
#### *Операции со стилями в области задач Стили и форматирование*

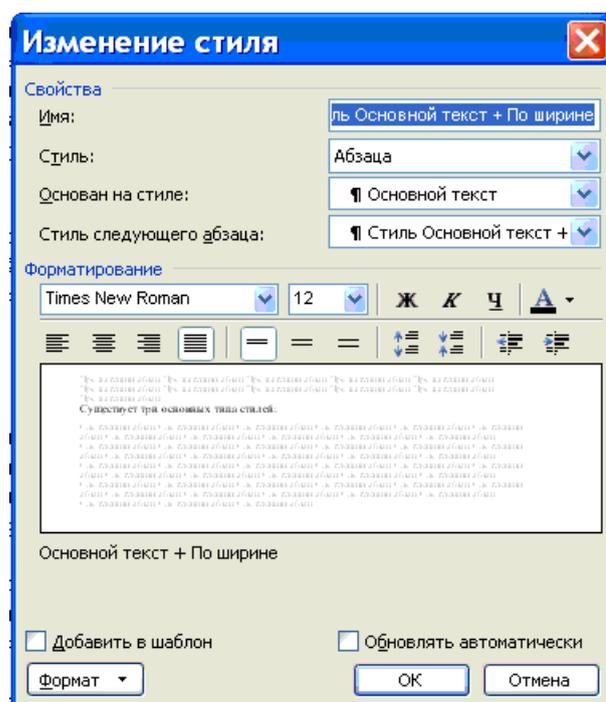
Стили можно изменять и создавать новые. Кроме того, можно применить другой стиль (наложить стиль) к уже отформатированному тексту, т.е. переформатировать его.

Для этого необходимо выбрать команду Формат / Стили и форматирование, появится область задач в режиме «Стили и форматирование». Установите курсор в текст, который необходимо переформатировать, и в области задач в разделе «Выберите форматирование для применения» щелкните на требуемый стиль, абзац под курсором переформатируется на выбранный стиль.

Если необходимо изменить стиль или создать новый стиль, то установите курсор в текст, стиль который необходимо изменить, в области задач в разделе «Форматирование выделенного текста» будет отображаться стиль текста под курсором. Далее необходимо подвести указатель мыши к названию стиля и щелкнуть на появившейся справа стрелке, откроется список команд: Очистить формат, Изменить стиль, Создать стиль. Выберите требуемую команду (изменить или создать стиль), появится соответствующее окно диалога, в котором можно осуществить изменение или создание стиля.

Стиль Обычный является основой для большинства других стилей абзаца, поэтому при изменении стиля Обычный изменятся все основанные на нем стили. Стиль Обычный не основан ни на одном из стилей.





Элементы текстовых документов, которые вводятся в документ в процессе его редактирования и форматирования:

Колонтитулы (Вид / Колонтитулы).

Символ (Вставка / Символ).

Ссылка (сноски; названия рисунков, таблиц, формул; перекрестные ссылки; оглавление и указатели). Вызывается командой Вставка / Ссылка.

Примечание (Вставка / Примечание).

Гиперссылка (Вставка / Гиперссылка).

Закладка (Вставка / Закладка).

Фоны, подложки и темы (Формат / Фон, Формат / Тема).

Колонтитул — это текст или рисунок (номер страницы, дата печати документа, название документа, фамилия автора, рисунок и т. д.), который печатается внизу или вверху каждой страницы документа. В зависимости от места расположения (на верхнем или на нижнем поле страницы) колонтитулы бывают верхними и нижними.

Для создания колонтитула необходимо выполнить команду Вид / Колонтитулы. Колонтитулы, как и фон документа, а также поля, размер и ориентация страниц, номера страниц, разрывы страниц и разделов относятся к изменению внешнего вида страниц и документа.

**Проверка правописания (орфографии и грамматики) в Word**

Проверка орфографии (правописание) – это средство редактора, позволяющее проверить и исправить написание слов в документе. Редактор сравнивает слова в проверяемом документе со словарем, причем неизвестные слова при этом выделяются. После этого можно пропустить слово, откорректировать его или занести в словарь.

Проверка грамматики – это проверка грамматических и стилистических правил письма (несогласованность подлежащего и сказуемого, неверные предложные сочетания).

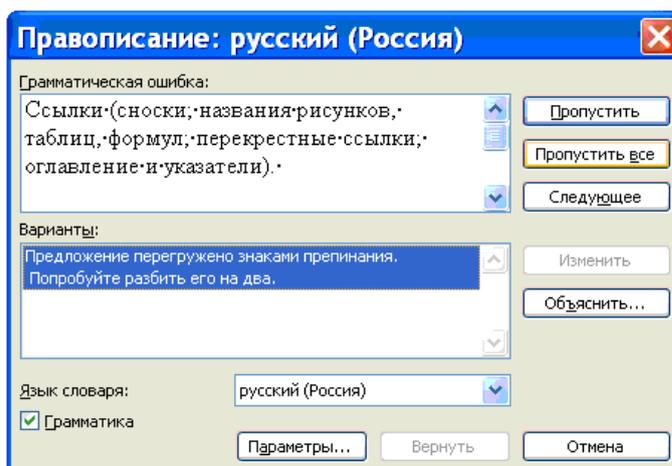
Существуют несколько способов проверки правописания:

Проверка правописания и грамматики при вводе текста.

Ручная проверка орфографии и грамматики.

Чтобы установить автоматическую проверку правописания и грамматики, выполните следующее: Выберите команду Сервис/параметры и в появившемся окне щелкните на вкладке Правописание. Установите флажок автоматически проверить орфографию и автоматически проверить грамматику.

Для ручной проверки орфографии и грамматики необходимо выбрать команду Сервис/Правописание, появится окно диалога Правописание, с помощью которого можно выполнить требуемую проверку.



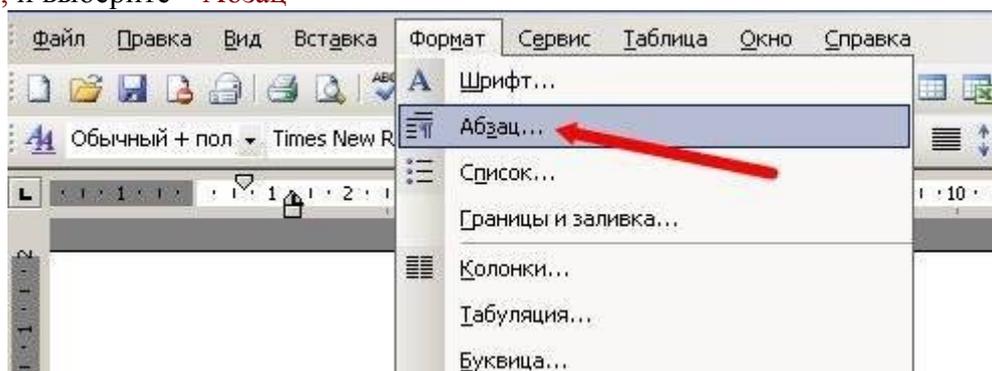
## 10. Форматирование абзацев в Word

Форматирование абзацев в Word. В любом документе не обойтись без форматирования текста. Основной смысловой единицей текста обычно является абзац. Поэтому команды выравнивания и операции форматирования предназначены для изменения внешнего вида отдельных абзацев. Без абзаца текст будет смотреться не красиво, да и для чтения он мало пригоден. Вы, наверное, сами убедились в этом? На сайтах, где текст идет сплошной стеной, очень быстро устают глаза. И каким бы интересным он ни был, посетители бросают читать его на полуслове и уходят. Вот поэтому грамотный веб-дизайнер обязательно разбивает текст на небольшие абзацы. Книги, написанные без абзацев, теряют свою привлекательность и чаще всего остаются не востребованными.

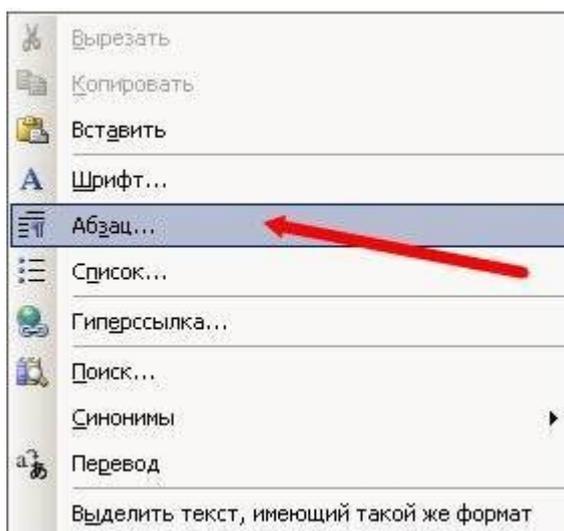
Поэтому абзацы должны быть обязательно. В Word есть возможность облегчить себе работу с абзацами. Для этого необходимо настроить текстовый редактор «под себя».

Настройка форматирования абзацев в Ворде для всех версий программы одинакова, поэтому по этому поводу можете не заморачиваться.

Для полного форматирования абзаца используют диалоговое окно «Абзац». Войдите в меню – **Формат**, и выберите – **Абзац** –



или щелкните правой кнопкой мыши по полю документа и в выпадающем контекстном меню выберите пункт **Абзац**.



В открывшемся окне «Абзац» на вкладке «Отступы и интервалы» настройте Выравнивание абзаца

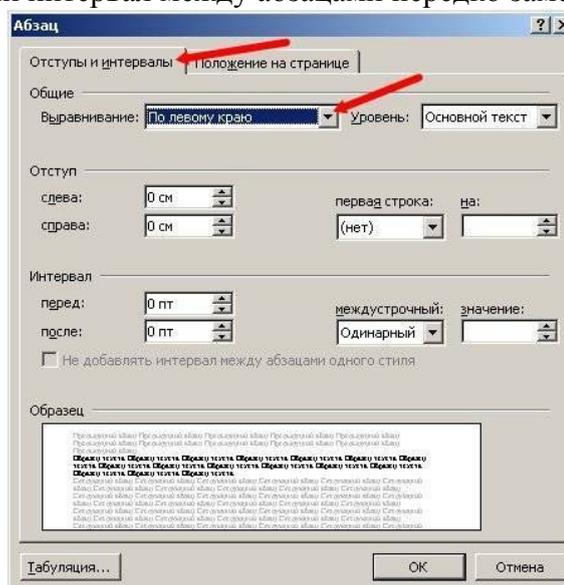
- По левому краю
- По центру
- По правому краю
- По ширине

В раскрывающемся списке Уровень можете задать смысловый уровень абзаца в общей структуре текста.

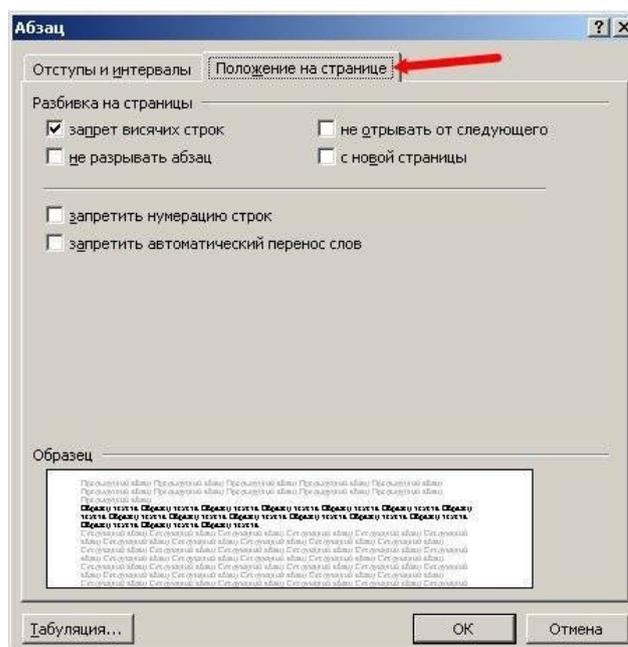
Панель Отступ определяет правую и левую границу абзаца относительно правой и левой границ страницы.

В раскрывающемся списке Первая строка можете задать наличие и размеры «красной строки» (абзацного отступа).

Панель Интервал позволяет задать промежутки между абзацами, а также между строками данного абзаца. Увеличенный интервал между абзацами нередко заменяет абзацный отступ.



Вкладка **Положение на странице** предназначена для форматирования абзацев, попадающих на границу между страницами. Здесь можно запретить отрывать от абзаца одну строку, потребовать, чтобы абзац размещался на одной странице целиком, «присоединить» следующий абзац к данному или начать текущим абзацем новую страницу.



С помощью окна **Абзац** очень просто выполнять форматирование, но делать этого не стоит. Оно полезно для изучения приёмов форматирования, но для практической работы лучше им не пользоваться.

## 11. Создание списков в Word

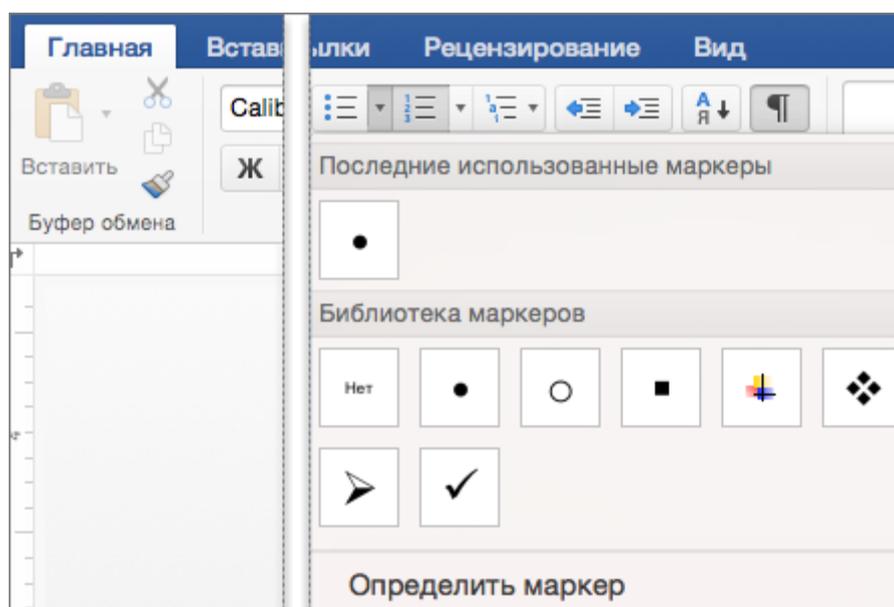
Чтобы превратить строку текста в маркированный или нумерованный список, выделите текст и на вкладке **Главная** в группе " **Абзац** ", нажмите кнопку **маркеры** или **Нумерация**. Каждую строку или абзац становится маркированных или нумерованных элементов.

### *Создание списка с нуля*

Поместите курсор в место, где вы хотите маркированного или нумерованного списка.

На вкладке **Главная** щелкните стрелку рядом с кнопкой **маркеры** или **Нумерация**.

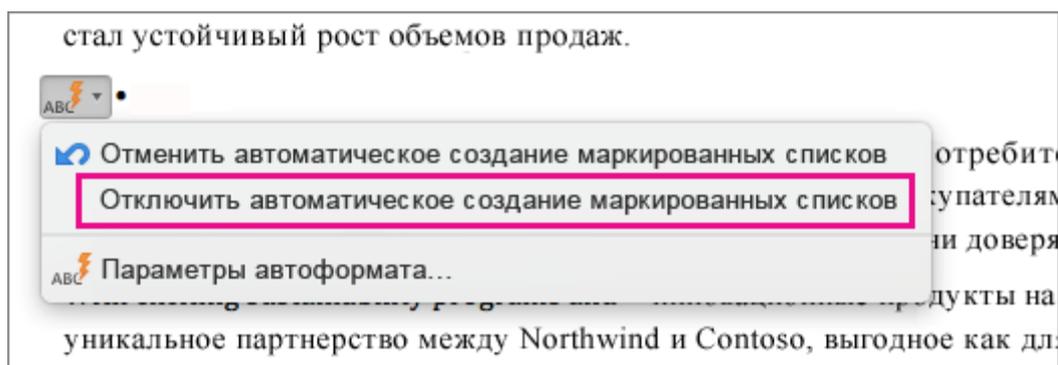
Выберите стиль и начните вводить текст.



Каждый раз при нового маркера или номера, или нажмите клавишу ВВОД, чтобы завершить список, дважды нажмите клавишу ВВОД.

Совет: Когда вы начинаете абзац со звездочки с пробелом (\*) или числа 1 с точкой (1.), Word отображает кнопку Параметры автозамены и начинает создавать маркированный или нумерованный список. Если вам не нужен маркированный или нумерованный список, нажмите кнопку

Параметры автозамены и выберите команду Отключить автоматическое создание маркированных списков или Отключить автоматическое создание нумерованных списков.



### Начало нумерации с 1

Щелкните элемент, который должен быть первым элементом нового списка.

Сочетание клавиш CTRL + нажмите или щелкните правой кнопкой мыши элемент и нажмите кнопку **Нумерация**.

### Настройка отступов элементов списка

Выделите строки, для которых нужно настроить отступ.

На вкладке **Главная** в группе **Абзац** нажмите кнопку **Увеличить отступ** .

### Удаление элементов из списка

Выделите элементы, которые нужно удалить.

Нажмите клавишу DELETE.

### Завершение маркированного или нумерованного списка

Выполните одно из указанных ниже действий.

В конце маркированного или нумерованного списка дважды нажмите клавишу RETURN.

Выделите строки текста, что вам не нужно в списке и на вкладке "**Главная**" в группе **Абзац** нажмите кнопку **Маркированный список**  или **Нумерованный список** .

## 12. Стили и шаблоны в Word

*Шаблон документа* — это файл с расширением dot, в котором содержатся набор стилей, а также настройки меню, панелей инструментов и сочетаний клавиш, элементы автотекста и макросы. Шаблон может также включать текст, предназначенный для вставки в создаваемые на его основе документы.

Каждый документ Word связан с определенным шаблоном, который выбирается при его создании, используется по умолчанию или присоединяется к нему позднее. По умолчанию новые документы Word основываются на шаблоне Normal.dot (Обычный). Далее мы будем называть его просто Обычный.

*Стиль* — это именованный набор параметров форматирования, которые можно применить к выбранному фрагменту текста. Очевидно, что перечень параметров, применимых к конкретному фрагменту текста, зависит от его типа, поэтому различают четыре вида стилей:

*стиль знака* — содержит только параметры шрифта;

*стиль абзаца* — содержит полный набор сведений, необходимых для форматирования абзаца, то есть параметры абзаца и параметры шрифта;

*стиль таблицы* — содержит информацию о границах таблицы, цветах текста ее элементов, заливке, выравнивании и шрифтах;

*стиль списка* — содержит сведения о форматировании маркированного или нумерованного списка.

Стили (то есть их имена и определения) хранятся в файлах документов и шаблонов. При создании документа в него копируется набор стилей из того шаблона, на котором он основан. Фор-

матируя документ, можно использовать стили, содержащиеся в нем самом, а также стили из шаблона Normal и встроенные стили Word. Кроме того, можно применять стили из так называемых общих шаблонов. *Общие шаблоны* не используются в качестве основы для конкретных документов, они загружаются автоматически при запуске программы или вручную после ее запуска, после чего хранящиеся в них стили становятся доступными во всех открываемых в программе документах.

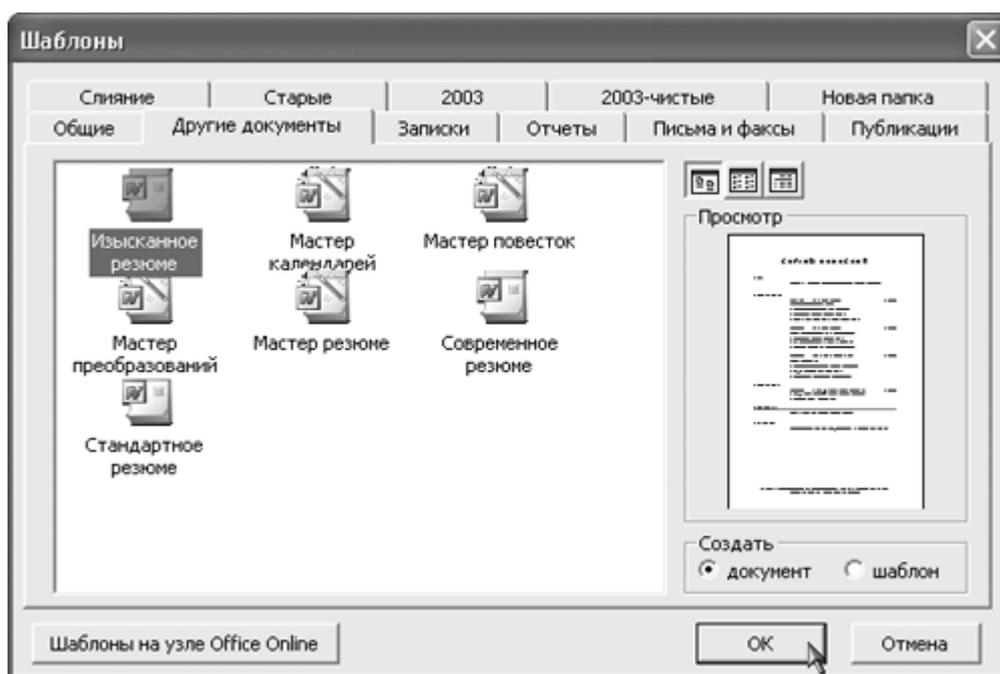
Программа Word дает возможность не только изменять, переименовывать и удалять стили, содержащиеся в документе или шаблоне, но и создавать в нем новые стили. Кроме того, она позволяет копировать стили из одного файла в другой, в том числе из шаблона в шаблон, из шаблона в документ, из документа в шаблон и из документа в документ.

Все сказанное здесь о доступе к стилям и выполняемых над ними операциях относится и к другим элементам, которые могут храниться в шаблонах документов Word.

Создание документа на основе шаблона

Посредством команды *Файл* -> *Создать* откройте область задач *Создание документа* и в области Шаблоны щелкните на ссылке *На моем компьютере*. Программа отобразит диалоговое окно *Шаблоны*, показанное на рисунке, где вы сможете выбрать один из имеющихся на компьютере шаблонов.

Вместе с Word устанавливается большой набор шаблонов для создания разнообразных документов личного и делового характера. Поэтому шаблоны на вашем компьютере есть обязательно.



Когда вы щелкаете на значке шаблона, справа появляется образец документа, который будет создан на его основе. Создайте несколько документов на основе готовых шаблонов, чтобы получить представление о том, как они могут выглядеть.

Если щелкнуть на ссылке *Новый документ* в области задач Создание или выбрать значок с тем же именем в диалоговом окне Шаблоны, программа создаст новый документ на основе шаблона Normal. То же самое происходит после щелчка на кнопке *Создать* панели инструментов *Стандартная*. Вы не должны забывать, что при работе над документом доступны все сочетания клавиш, стили и другие элементы, входящие в состав шаблона Normal, даже если, создавая документ, вы выбрали иной шаблон.

Помимо шаблонов в диалоговом окне *Шаблоны* представлены так называемые *мастера создания документов* — мини-программы для создания стандартных документов. Каждая из них вводит серию диалоговых окон, где можно выбрать разные варианты оформления документа и

ввести сведения для заполнения его основных полей (таких, как имя адресата письма или реквизиты составителя резюме). Файлы мастеров, имеющие расширение *wiz*, присоединяются к документам подобно шаблонам, но их содержимое нельзя изменять. Набор стандартных мастеров устанавливается вместе с Word, а кроме того, их могут создавать разработчики программного обеспечения.

#### *Форматирование текста с помощью стилей*

Предположим, вы назначили некоторому фрагменту текста стиль. Если теперь изменить параметры форматирования данного стиля в документе, внешний вид текста автоматически изменится. То же самое произойдет, если скопировать в документ одноименный стиль с другими параметрами из шаблона данного документа, другого шаблона или документа. При этом программа дает возможность задать для документа установку, которая обеспечивает автоматическое обновление стилей при его открытии — стили заново копируются в документ из его шаблона.

### 13. Таблицы в Word

В программе Microsoft Word можно не только напечатать текст, но и сделать таблицу. Она вставляется в текст при помощи определенных кнопок, о которых мы сейчас будем говорить. Вы можете выбрать для своей таблицы нужно количество строк и столбцов, редактировать ее как угодно – расширять, сужать, объединять ячейки, удалять, печатать внутри текст и многое другое.

Но тут есть один важный момент. В ней нельзя будет произвести вычисления. То есть если требуется не только расчертить таблицу со словами и цифрами, но еще и произвести с цифрами какие-либо действия (сложить, умножить, вычислить процент и т.д), то тогда Вам нужно воспользоваться другой программой - Microsoft Excel.

*Стоит отметить, что в Word'e есть специальная кнопка для вставки таблиц Excel. Но с ними нужно уметь правильно обращаться.*

Пример таблицы, созданной в Word:

<b>Наименование</b>	<b>Количество</b>	<b>Цена</b>
Хлеб	5	1,5
Молоко	2	2
Масло	3	3

#### *Как вставить таблицу в программе Microsoft Word*

Сначала нужно поставить мигающий курсор (мигающую палочку) туда, где должна быть таблица. Для его смещения вниз нажмите кнопку Enter на клавиатуре. А чтобы поднять мигающий курсор выше – Backspace (кнопку удаления).

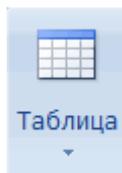
В Microsoft Word более старой версии (2003 года и ранее) для вставки таблиц есть специальный пункт в самом верху с правой стороны. Называется он «Таблица».

Таблица

А в программе Word более современной версии (2007-2016) для добавления таблицы нужно нажать на надпись «Вставка» в левом верхнем углу программы.

Вставка

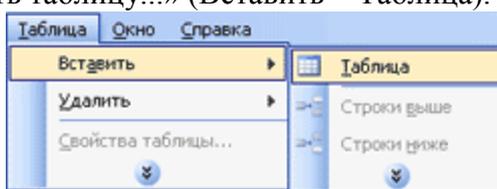
Когда Вы нажмете на эту надпись, появятся новые кнопки редактирования. Среди них есть «Таблица».



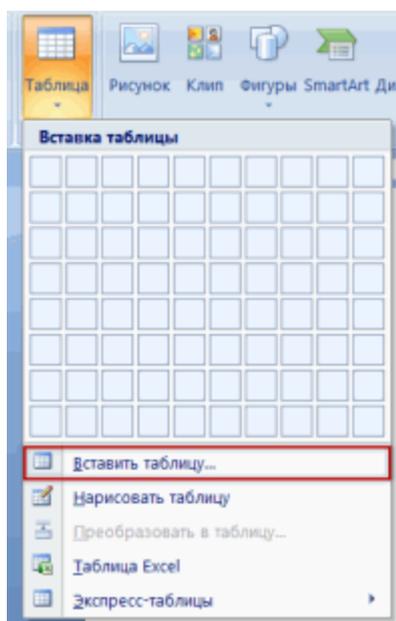
#### *Как добавить таблицу*

Если Вы нажмете на надпись «Таблица», откроется список, в котором программа Word предложит несколько вариантов ее вставки. Для начала поговорим о классическом способе.

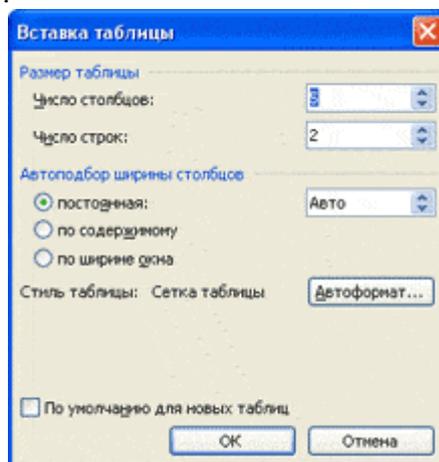
Нажмите на пункт «Вставить таблицу...» (Вставить – Таблица).



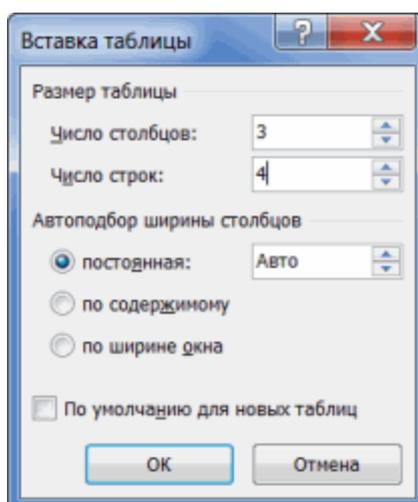
ИЛИ



Откроется небольшое окошко.



В верхней части нужно напечатать, сколько столбцов и строк должно быть в нашей таблице. Я, например, указал три столбца и четыре строки. Также обратите внимание на нижнюю часть окошка. Точка должна стоять на пункте «Постоянная», и в соседнем белом поле должно быть указано значение «Авто».



*Другие настройки ширины столбцов применяются крайне редко, в исключительных случаях, и рассматривать их в рамках данного урока мы не будем.*

И, наконец, нажмите на кнопку ОК.

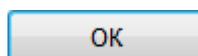
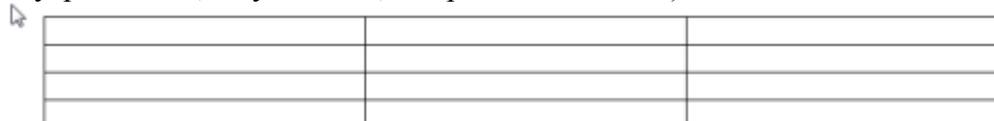


Таблица с указанным нами количеством столбцов и строк должна вставиться в документ (на лист). В моем случае вот что получилось:


*Как удалить таблицу*

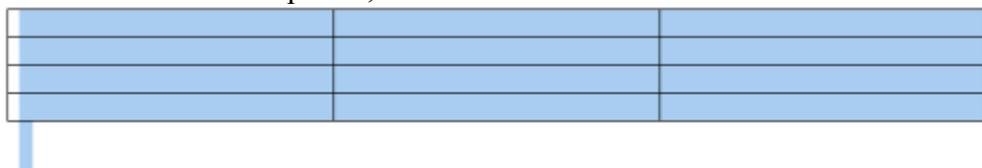
Удалить таблицу, созданную в программе Word, можно разными способами. Сейчас мы научимся это делать самым, на мой взгляд, простым, при помощи которого можно удалить таблицу в любой версии программы.

Сначала нужно ее выделить. Для этого наведите курсор на первую ячейку первой строки (только не внутрь ячейки, а чуть левее, за пределы таблицы).



*Ячейка* – это отдельный прямоугольник, из которого состоит таблица.

Нажмите левую кнопку мышки и, не отпуская ее, тяните в самый конец. Вот тут важный момент – тянуть нужно до последней строки и чуть ниже. Выделиться таблица должна особым образом, с маленьким «хвостиком» после нее.



А теперь нажмите кнопку Delete (Del) на клавиатуре. Если Вы выделили таблицу правильно – так, как написано чуть выше, – то она исчезнет.

#### 14. Сноски, ссылки, колонтитулы, оглавление, список литературы в Word

При наборе книг и рефератов часто возникает необходимость использования сносок. В Word имеется два типа сносок: стандартные, располагаемые в конце той страницы, на которой расположена ссылка, и общие для документа, располагаемые в его конце. Первые обычно исполь-

зуют для пояснения по ходу изложения, а наиболее типичное применение концевых сносок – ссылки на источник (например, на используемую литературу).

Для создания сноски установите курсор в то место, где должна будет находиться ссылка, и выберите пункт Сноска из меню Вставка ? Ссылка. В открывшемся окне, если вы создаете обычную сноску, сразу жмите ОК, и на месте курсора появится номер сноски, набранный в режиме верхнего индекса. Затем, если вы работали в режиме разметки, внизу страницы будет поставлен тот же номер, и курсор ввода переместится туда, чтобы вы смогли набрать поясняющий текст. Если же документ находится в стандартном режиме, то внизу будет открыто дополнительное окно, в котором и следует редактировать текст сносок. В том случае, когда требуется создать концевую сноску, достаточно переключить опцию вставки сноски на положение концевые сноски.

Если вы будете вставлять сноски регулярно, то заметите, что нумерация у них общая, то есть если на первой странице было три сноски, то на второй первая же сноска появится под номером четыре. Чтобы нумерация была индивидуальной для каждой страницы, нажмите в окне Сноски на кнопку Параметры, в результате чего получите возможность изменить ряд дополнительных опций.

В частности, можно указать, чтобы нумерация начиналась заново в каждом разделе или для каждой страницы (последний вариант – только для обычных сносок). Кроме того, предлагается определить, должны ли сноски находиться в самом конце страницы или располагаться сразу за текстом. Помимо этого можно поменять и формат номера: кроме арабских цифр можно использовать римскую или буквенную нумерацию или даже нумерацию при помощи специальных значков. Если же нумерация вообще не нужна, то допустимо ограничиться отметкой в виде символа, который либо вводится вручную, либо выбирается из таблицы символов непосредственно в окне Сноски.

#### *Номера страниц и колонтитулы*

Любому многостраничному документу положено иметь нумерацию страниц. Средствами Word создать нумерацию очень просто: для этого достаточно зайти в меню Вставка и выбрать пункт Номера страниц. После этого в появившемся диалоговом окне остается лишь выбрать, сверху или внизу страницы должны быть цифры и по какому краю следует их выравнивать. При желании можно провести нумерацию буквами или римскими цифрами; для этого следует нажать на кнопку Формат и в окне формата номера страницы указать подходящий формат числа. В этом же окне указывают и номер первой страницы. Например, если часть текста, скажем, с 1-й по 20-ю страницу, находится в одном файле, а нумерацию следует произвести в файле с продолжением, то укажите в этом окне в поле Начать с число 21.

Еще одной опцией является включение номера главы в дополнение к колонцифре (колонцифра – типографское название номера страницы). Для этого прежде всего следует пронумеровать названия глав. Несмотря на то что вы попытаете включить эту опцию для документа без нумерации глав, Word попросит вас использовать команду Нумерация заголовков из меню Формат, на самом деле это делается при помощи команды Список из того же меню. Открыв окно списка, перейдите на страницу Многоуровневый и выберите один из вариантов с нумерацией глав. Например, чтобы пронумеровать все заголовки первого уровня, выберите крайний правый вариант из нижнего ряда, после чего нажмите ОК. В зависимости от мощности ПК и объема документа программа может задуматься на несколько секунд или даже минут, после чего ко всем заголовкам будут добавлены номера. Теперь остается вернуться в окно формата номера страницы и отметить опцию Включать номер главы. После этого к номеру страницы будет добавлен и номер главы.

Хотя колонцифра и колонтитул похожи, как близнецы-братья, для установки колонтитула придется воспользоваться пунктом Колонтитулы из меню Вид. При этом документ переключится в вид Разметка страницы, в каком бы виде он до этого ни находился, основной текст станет серым, а область колонтитула будет выделена пунктирной рамкой. Заодно откроется панель инструментов Колонтитулы, содержащая кнопки для вставки автотекста, номера страницы и общего их числа, даты, времени, а также для вызова диалогов Формат номера страницы и Параметры страницы.

Если все, что требуется от колонтитула, – это нести статический текст, то просто наберите его с клавиатуры. При помощи кнопки автотекста можно добавить и динамическое, то есть изменяющееся со временем или от страницы к странице, содержание. Например, выбрав из кнопки-меню Вставить автотекст пункт Стр. <№> из <всего>, вы получите нумерацию страниц, показанную на рисунке. При этом вставка собственно номера страницы как такового вам уже не потребуется. Завершив правку колонтитула, нажмите Заккрыть. Впоследствии можно поправить колонтитулы, вновь обратившись к меню Вид либо дважды щелкнув по самому колонтитулу в режиме разметки (в обычном режиме колонтитулы не отображаются).

Следует учитывать, что колонтитулы будут одинаковыми для всего документа, если только вы не указали в параметрах страницы опцию Зеркальные поля. В таком случае колонтитулы будут разными для четной и нечетной страниц. А чтобы создать разные надписи в разных частях документа, следует разбивать его на разделы.

#### *Разделы*

Для разбивки документа используют пункт Разрыв из меню Вставка, открывающий диалоговое окно Разрыв. Два из первых трех пунктов вставляют в документ специальные символы – начала новой страницы и новой строки, заставляющие редактор принудительно перейти к новой странице или строке соответственно, а пункт Новая колонка служит для установки разрыва колонки в определенном месте.

Более интересной возможностью является создание нового раздела. Разделы предназначены для отделения значительных частей текста (например, глав книги) друг от друга. При этом становится возможным в рамках одного файла применять различные колонтитулы, использовать отдельную нумерацию страниц и заголовков и т. д. При выборе варианта Со следующей страницы новый раздел начнется со страницы, следующей за той, на которой в настоящий момент находится курсор ввода, причем весь текст, находящийся за ним, будет перенесен в новый раздел. Если же выбрать На текущей странице, то разрыв будет вставлен непосредственно в том месте, где находится курсор, а визуально это будет выглядеть как завершение строки. А при выборе создания раздела с четной или с нечетной страницы раздел начнется соответственно с ближайшей четной или нечетной страницы.

#### *Разделение на колонки*

Если вы работаете с крупноформатной бумагой либо вам требуется разместить на странице мелкий текст, то для удобочитаемости рекомендуется разбивать его на две колонки или более. Это обусловлено тем, что строку, содержащую очень много символов, трудно воспринять как целое, в результате чего взгляд будет «блуждать», срываясь между строками.

Чтобы разделить текст на колонки, щелкните по кнопке Столбцы, расположенной на стандартной панели инструментов. Прямо под ней появится окошко, при помощи которого, просто перемещая указатель мышки, можно выбрать количество столбцов. Щелкните по подходящему вам варианту, и документ (или текущий раздел, если разделов несколько) будет разбит на указанное число колонок.

Текст в колонках переходит с нижней части одной колонки к верхней части следующей, расположенной правее на той же странице. При этом следует учитывать, что в обычном режиме Word показывает только одну колонку, а для того чтобы увидеть содержимое так, как оно будет выглядеть при печати, следует перейти в режим разметки. Впрочем, если вы работаете с Word XP, то переход в режим разметки будет произведен автоматически сразу после того, как вы создадите колонки.

Когда возникает необходимость перенести текст колонки не в конце страницы, а в произвольном месте, используйте разрыв колонки (Вставка /Разрыв). Если надо озаглавить текст, находящийся в колонках, таким образом, чтобы заголовок располагался над ними, то опять-таки воспользуйтесь разрывом, установив новый раздел на текущей странице.

При создании колонок с помощью кнопки Столбцы их ширина будет равной. Чтобы изменить размеры колонок, достаточно передвинуть разделитель на линейке. А если дважды по нему щелкнуть, откроется окно Колонки, в котором можно задать точные параметры. Это окно доступно также из меню Формат, пункт Колонки; с его помощью можно не только изменять параметры

уже имеющихся столбцов, но и создавать новые. Изменению поддается число колонок, их ширина и ширина межколонного интервала (промежутка). Допустимо установить и разделитель – вертикальную черту между колонками. Все ваши настройки, как обычно, будут схематически отображаться.

#### *Создание оглавления и указателей*

Для объемистых документов принято составлять предметные указатели, содержащие ключевые слова и фразы. Когда-то эту утомительную работу приходилось выполнять вручную, но с помощью Word ее можно до известной степени автоматизировать.

Создание предметного указателя начинается с выбора слов, предназначенных для включения в указатель. Делается это вручную, когда вы выделяете нужное слово и нажимаете Alt+Shift+X, чтобы открыть окно Определение элемента указателя. Выделенный текст окажется в строке основного элемента, при этом вы можете его сразу отредактировать (например, сменить число или падеж). Завершается определение нажатием на кнопку Пометить, после чего Word может перейти в режим показа непечатных знаков, так что не надо пугаться появлению в тексте {XE "слово"} – это всего-навсего служебный элемент, который не будет выведен на печать. Таким вот образом придется пройтись по всему документу, выделяя слова для внесения в указатель. Процедура это достаточно долгая, но иногда – просто необходимая. Кроме того, собственно сборка указателя будет произведена программой быстро и точно.

В то же время имеется способ автоматической пометки, при котором, правда, может быть потеряна точность указателя (то есть может появиться «мусор» из неполных частей слов и иные лишние ссылки). Для автоматической пометки следует сначала создать отдельный текстовый файл (хоть в Word, хоть в Блокноте) и ввести в него слова, которые должны быть включены в указатель для основного документа. После того как вы такой список подготовили, сохраните и закройте файл списка и вернитесь к тому документу, которому требуется указатель. После такой подготовки, как и в первом случае, останется лишь обратиться к окну Оглавление и указатели (меню Вставка / Ссылка).

Если вы предпочли использовать второй способ пометки указателей, то прежде всего нажмите на кнопку Автопометка, в диалоге открытия файла укажите документ, содержащий предварительно введенные ключевые слова, и нажмите Открыть. После этого программа сама отметит совпадающие слова в документе.

Вставка собственно указателя в помеченный тем или иным способом документ не представляет никакой сложности: достаточно в окне оглавления и указателей выбрать число колонок и стиль оформления (стиль выбирается из ниспадающего списка Форматы). Если включить опцию Номера страниц по правому краю, то становится возможным указать еще и заполнитель (точки или линии), который будет вставлен между колонкой со словами указателя и колонкой с номерами страниц. В результате будет создан новый раздел документа, содержащий предметный указатель указанного вида. При изменении номеров страниц и редактировании текста самого документа указатель также будет обновляться. Если же потребуются внести в указатель еще несколько слов, то лучше не вводить их в него вручную, а создать весь указатель заново.

Значительно проще создается оглавление. Дело в том, что, выделяя текст заголовка тем или иным стилем, от Заголовков 1 до Заголовков 9 (вот еще один важный момент использования именно стилей, а не произвольного изменения оформления шрифта!), вы даете программе знать, что это именно заголовок и, следовательно, данный текст является потенциальным кандидатом на включение в оглавление. Таким образом, помечать уже ничего не надо – достаточно вызвать окно Оглавление и указатели и перейти на закладку Оглавление.

Выбор параметров оглавления сводится к тому, что вы указываете, следует ли показывать номера страниц и по какому краю их выравнивать, а также заголовки до какого уровня следует включать в оглавление. Из списка Форматы (или Вид) выбирают стиль оформления. Как обычно, примерный внешний вид отражается в образце (в Word 2000 и XP образца два – для обычного документа и в варианте для Web).

Подобно указателю, оглавление вставляется в отдельный раздел документа, создаваемый автоматически в том месте, где находился курсор ввода в момент обращения к диалогу создания

оглавления. В Word 2000 все элементы оглавления являются гиперссылками, поэтому щелчок левой клавишей мышки по строке оглавления приведет не к помещению в это место курсора ввода, а к перемещению в место, на которое ссылается данный пункт оглавления. Поэтому для правки оглавления в Word 2000 придется щелкать по полю рядом со строкой и лишь затем, при помощи стрелок на клавиатуре, передвигать курсор на нужную позицию, что весьма неудобно. Поэтому в Word XP щелчок по ссылке, приводящей к перемещению в документе, срабатывает только тогда, когда на клавиатуре нажата клавиша Ctrl.

Кроме указателей и оглавления в этом же окне можно создавать еще и список иллюстраций. Он создается подобно оглавлению, с той лишь разницей, что выбирается текст со стилем не заголовков, а названия объектов.

Кроме уже перечисленных, имеется еще один специальный тип указателя – таблица ссылок, представляющая собой список упоминаемых в документе законодательных актов, судебных решений и т. д. Впрочем, эта возможность ориентирована прежде всего на американских пользователей.

## 15. Простейшие вычисления и операции в MathCAD

Основные математические вычисления проводятся пользователем с помощью операторов.

**Операторы** – это условные символы, служащие для выполнения определенных задач, например: -, +, , /, =. Знаки -, +, = не меняют своего значения как при вводе с клавиатуры, так и при отображении на экране. При введении с клавиатуры знака \* (звездочка) на экране появляется привычный для математики знак умножения. При введении знака / (деление), на экране появляется математическая десятичная дробь.

При выполнении символьных операций, числа используются и только в виде символов. А при выводе результатов они приобретают свои численные значения.

**Операторы** используются для составления математических выражений – формул. При написании формул знак равенства (=) используется как знак вывода результатов вычислений, а для присваивания значения переменной следует применять знак присваивания (:=).

Например: для присваивания переменным а и b числовых значений, следует ввести: a:=3b:=5, а для вычисления суммы а и b вставить знак равенства (=) a+b=

Если переменной присваивается начальное значение при помощи оператора (:=), или даже =, то такое присваивание называется *локальным*. До этого присваивания переменную нельзя использовать, т.к. она не определена. Однако с помощью знака  $\equiv$  (три горизонтальные черточки) можно обеспечить *глобальное* присваивание. Оно может производиться в любом месте документа. К примеру, если провести присваивание таким образом в конце документа, то она будет иметь тоже значение, что и в начале документа. Но в дальнейшем значение этой переменной может изменяться при помощи локального присваивания.

### **Переменные.**

**Переменные** - это поименованные объекты, которым можно присваивать различные значения. Имена констант, переменных и объектов называются идентификаторами. Идентификаторы в системе Mathcad могут иметь практически любую длину. При их задании можно использовать латинские и греческие буквы, а также цифры. Однако начинаться идентификатор может только с латинской буквы, например: z, w1, CTY, L\_123p, SUMM. Пробел, недопустим, его заменяют значком \_.

В математике часто возникает необходимость в задании некоторого ряда значений – чаще всего упорядоченного. Например, для вычисления факториала  $N! = 1*2*3*...*(N-1)*N$  нужно сформировать ряд целых чисел от 0 до N с шагом 1. Для создания таких рядов в Mathcad используется **ранжированная переменная**.

**Способ ее задания:** для задания переменной интервала значений от 1 до 5 с шагом 0,5 набирают: n:=1,1.5..5. После знака присваивания ставится начальное значение интервала, запятая, следующее значение переменной, т.е. после 1 с учетом шага 0.5 следует число 1.5, после этого нажимают на кнопку задания диапазона  (панель **Матрицы**), и пишется конечное значение пе-

ременной. Для того, чтобы просмотреть значения ранжированной переменной, надо написать= , после этого, на листе появится табличка с вычисленными данными.

### **Функции.**

Функции в Mathcad записываются в обычной для математика форме:

$f(x, \dots)$  – функция;

$f$ – имя функции;

$x, \dots$  - список переменных.

В Mathcad формально можно разделить функции на два типа:

Встроенные функции (например,  $\sin(x)$ );

Функции, определенные пользователем (например,  $f(x,y)=x \cdot \cos(x+y)$ ).

Система Mathcad содержит расширенный набор встроенных элементарных функций. Имя функции отождествляется с выполняемой ей математической функцией, например:  $\sin(x)$  – функция, вычисляющая синус аргумента  $x$ , где  $\sin$  – имя функции. Полный перечень вычисляемых функций можно просмотреть, нажав на пользовательской панели инструментов на кнопку  $f(x)$  или выбрав пункт меню **Вставка** подменю **Функция**.

Аргумент и значение функции могут быть действительным или комплексным числом  $z$ . Ниже приведены функции разбитые на 6 групп.

Тригонометрические:  $\sin(z), \cos(z), \tan(z), \sec(z), \csc(z), \cot(z)$ .

Гиперболические:  $\sinh(z), \cosh(z), \tanh(z), \operatorname{sech}(z), \operatorname{csch}(z), \operatorname{coth}(z)$ .

Обратные тригонометрические:  $\operatorname{asin}(z), \operatorname{acos}(z), \operatorname{atan}(z)$ .

Обратные гиперболические:  $\operatorname{asinh}(z), \operatorname{acosh}(z), \operatorname{atanh}(z)$ .

Показательные и логарифмические:  $\exp(z), \ln(z), \log(z)$ .

Функции комплексного аргумента:  $\operatorname{Re}(z)$  – выделение действительной части  $z$ ,  $\operatorname{Im}(z)$  – выделение мнимой части  $z$ ,  $\operatorname{arg}(z)$  – вычисление аргумента (фазы).

### **Символьный вывод.**

Наряду с численным выводом, в Mathcad имеется возможность символьного, или аналитического, вычисления значения выражения. Для символьных вычислений имеется ряд специальных средств, самое простое из них – это оператор **символьного вывода**. Если численный  $\rightarrow$  (вывод – это «запрограммированный» расчет по формулам, то символьный вывод – это результат работы символьного процессора. Работа символьного процессора заключается в анализе самого текста математических выражений.

Чтобы вычислить символьно математическое выражение, необходимо выполнить следующее:

Ввести это выражение;

Ввести оператор символьного вывода нажатием соответствующей кнопки  на панели

### **Символы.**

После этого справа от символа оператора символьного вывода появится определенное аналитическое значение выражения. Если символьному процессору Mathcad не удастся аналитически упростить выражение, то в том же виде, что и слева,  $\rightarrow$  оно будет выдано справа от знака

Пример:

$$b \cdot \sin(\operatorname{asin}(c \cdot x)) \rightarrow b \cdot c \cdot x$$

$$x^2 \cdot \cos(x + y) \rightarrow x^2 \cdot \cos(x + y)$$

### **Вычислительные операторы.**

Вычислительные операторы вставляются в документы при помощи панели инструментов **Вычисления**. При нажатии любой из кнопок в документе появляется символ соответствующего математического действия, снабженный несколькими местозаполнителями (рис.3).



После ввода какого-либо вычислительного оператора имеется возможность вычислить его значение либо численно, нажатием клавиши  $\langle \Rightarrow \rangle$ , либо символично, с помощью оператора символического вывода.

**Задание:**

Перед выполнением каждого задания, необходимо сначала писать само задание.

**Например:**

1. Простейшие арифметические операции

$$\begin{aligned} a &:= 2 & b &:= 3 \\ a + b &= 5 \\ a - b &= -1 \end{aligned}$$

Задайте переменным  $a$  и  $b$  числовые значения и выполните над ними элементарные операции: сложение, вычитание, умножение, деление, возведение  $a$  в степень  $b$ , корень из  $a$ .

Вычислить тригонометрические функции при заданном самостоятельно значении  $x$ , используя пункт Меню **Вставить/Функция**

$$\sin(x) \quad \cos(x) \quad \tan(x) \quad a \tan(x) \quad \sec(x) \quad \cos(x)$$

Вычислить натуральный и десятичный логарифмы при заданном  $x$ .

Вычислить производные первого, второго и третьего порядка от заданной функции

$$f(x) = 2x + 3 + \cos(x)$$

Вычислить определенный (пределы интегрирования задать самостоятельно) и неопределенный интегралы двух функций:

$$f(x) = 2x + 3 + \cos(x)$$

$$f(x) = e \cdot \cos(x)$$

Вычислить пределы функций

$$\lim_{x \rightarrow 0} \frac{\sin(x)}{x} \quad \lim_{x \rightarrow \infty} \frac{\sqrt{x^2 + 2}}{3 \cdot x + 6}$$

Найти сумму и произведение. Число  $m$  – задать самостоятельно.

$$\sum_{n=1}^m \frac{2 + (-1)^n}{n - \ln(n)}$$

Для присвоения значений от 1 до 5 использовать Булево равенство  $\mathbf{=}$  из панели **Булевы операторы**.

$$x^4 \cdot \prod_{(n=1..5)} \frac{1}{x + n}$$

Пользуясь мастером функций получить значение синуса и гиперболического арккосинуса

n := 1..5		m := -4, -2.8..2	
n =	acosh(n) =	m =	sin(m) =
1	0	-4	0.757
2	1.317	-2.8	-0.335
3	1.763	-1.6	-1
4	2.063	-0.4	-0.389
5	2.292	0.8	0.717
		2	0.909

## 16. Преобразование алгебраических выражений в MathCAD

Выражения в Mathcad часто имеют не тот вид, в котором Вы хотели бы их видеть. Поэтому в программе есть несколько инструментов, позволяющих преобразовывать выражения путем перестановки или замены.

Рассмотрим простой пример – разложение в ряд натурального логарифма в окрестности точки  $x=1$ :

$$\ln(x) \xrightarrow{\text{series}, 3, x=1} -1 + x - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3}$$

Все слагаемые, кроме первых двух, кратны  $(x-1)$ . По уравнению видно, что первые слагаемые тоже можно привести к такому виду. Самый простой путь сделать это – вырезать и вставить нужный фрагмент:

$$-1 + x - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3}$$

Скопировать:  $(x-1)$

Удалить:  $-1 + x$

$$- \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3}$$

Вставить:  $(x-1)$

$$(x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3}$$

У такого способа есть недостаток – очень легко сделать ошибку: можно забыть скопировать знак «минус», или вставить не то выражение... Простой путь проверить, что ошибки нет – это присвоить переменной некоторое значение (лучше всего – неправильную дробь) и проверять значение выражения после каждого шага:

$$x := \frac{11}{7} \quad -1 + x - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} = 0.47$$

$$(x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} = 0.47$$

Если получен тот же результат, то можно быть уверенным, что ошибки нет. Однако, теперь Вы задали переменной  $x$  значение. Если Вы хотите продолжить аналитические преобразования, нужно предварительно удалить присваивание, т.е. очистить переменную, с помощью функции **clear**:

$$x = 1.571 \quad \text{clear}(x) \quad x = ?$$

Можно одним действием очистить сразу несколько переменных:

$$y := 4 \quad z := -3 \quad \text{clear}(y, z) \quad y = ? \quad z = ?$$

### Ключевые слова для приведения к нужному виду

Mathcad может сделать кое-что сам для преобразования выражения с помощью ключевых слов:

“*simplify*”, “*expand*”, “*factor*” и “*parfrac*” – для преобразования самого выражения;

“*float*” – для преобразования результата вычисления;

“*assume*” (с модификаторами) – ограничивает диапазон возможных значений переменных.

*Символьное вычисление*

Начальная точка всех символьных вычислений – это операция аналитического преобразования. Эта операция приводит степени и множители, а также сокращает общие множители:

$$x \cdot x + 4 \cdot x - 2 \cdot x \rightarrow x^2 + 2 \cdot x \quad \frac{(1+x) \cdot (2-x)}{1+x} \rightarrow 2-x$$

Эта операция также подставляет известные соотношения для переменных. Например, есть выражение:

$$Y \cdot X$$

Нужно заменить переменные выражениями:

$$X := X - a \quad Y := Y + b$$

Mathcad выдает сообщение, что переменные не определены, но для символьных операций это не имеет значения:

$$X \rightarrow X - a \quad Y \rightarrow Y + b$$

В результате получаем:

$$X \cdot Y \rightarrow (X - a) \cdot (Y + b)$$

Для подстановки служит также ключевое слово “*substitute*”, но метод, описанный выше, более гибкий. Mathcad не может заменить выражение выражением, как в символьных математических пакетах, таких как Maple.

*Упростить* – “*simplify*”

С помощью ключевого слова “*simplify*” можно упрощать тригонометрические выражения:

$$\sin(x)^2 + \cos(x)^2 \xrightarrow{\text{simplify}} 1$$

$$\sin(x) \cdot \cos(x) \xrightarrow{\text{simplify}} \frac{\sin(2 \cdot x)}{2}$$

Также можно упрощать многие произведения:

$$e^x \cdot e^y \xrightarrow{\text{simplify}} e^{x+y}$$

(С помощью оператора аналитического преобразования так сделать нельзя.)

Раскрыть скобки – “*expand*”

Ключевое слово “*expand*” раскладывает выражение путем перемножения элементов произведения и возведения их в степень:

$$(x+2) \cdot (x-3) \xrightarrow{\text{expand}} x^2 - x - 6$$

$$\frac{(x+2) \cdot (x-3)}{(2 \cdot x - 7) \cdot (x+2)} \xrightarrow{\text{expand}} \frac{x}{2 \cdot x - 7} - \frac{3}{2 \cdot x - 7}$$

Разложить на множители – “*factor*”

Это ключевое слово раскладывает на множители многочлены и дроби с многочленами:

$$x^2 - x - 6 \xrightarrow{\text{factor}} (x+2) \cdot (x-3)$$

$$\frac{x}{2 \cdot x - 7} - \frac{3}{2 \cdot x - 7} \xrightarrow{\text{factor}} \frac{x-3}{2 \cdot x - 7}$$

“*factor*” – обратное по отношению к “*expand*” действие. Однако результат этого действия не всегда полезен:

$$x^2 - 1 \xrightarrow{\text{factor}} (x-1) \cdot (x+1) \quad , \text{ но } \quad x^2 + 1 \xrightarrow{\text{factor}} x^2 + 1$$

Разложение на дроби – “*parfrac*”

Это ключевое слово раскладывает выражение в сумму дробей:

$$\frac{x - x^2 + 2}{2 \cdot x^2 - x^3 + 5 \cdot x - 6} \xrightarrow{\text{parfrac}} \frac{1}{3 \cdot (x-1)} + \frac{4}{15 \cdot (x+2)} + \frac{2}{5 \cdot (x-3)}$$

Вычисление с плавающей запятой – “*float*”

Это ключевое слово выводит доступный числовой результат в десятичном формате. Количество отображаемых цифр – 20:

$$\pi \xrightarrow{\text{float}} 3.1415926535897932385$$

Такое количество может быть слишком большим, но его можно уменьшить с помощью дополнительного местозаполнителя с модификатором:

$$\pi \xrightarrow{\text{float}, 4} 3.142$$

Предположить – “*assume*”

Ключевое слово “*assume*” позволяет определить диапазон переменных. На простом примере показано символьное вычисление квадратного корня:

$$\sqrt{x^2} \rightarrow \sqrt{x^2}$$

$$\sqrt{x^2} \xrightarrow{\text{assume}, x > 0} x \quad \sqrt{x^2} \xrightarrow{\text{assume}, x < 0} -x$$

Вы можете сочетать “*assume*” с другими ключевыми словами, как мы это уже делали, когда искали действительные корни уравнения:

$$(x^3 - 1) \cdot (x^2 - 2) = 0$$

$$(x^3 - 1) \cdot (x^2 - 2) = 0 \xrightarrow{\text{solve, assume, } x = \text{real}} \begin{bmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \end{bmatrix}$$

В этом примере важен порядок ключевых слов.

### Другие ключевые слова

В меню Математика → Символьные операции находится большое число ключевых слов и модификаторов. Одна группа, о которой следует упомянуть, – это прямые и обратные преобразования:

Преобразование Лапласа	$\xrightarrow{\text{laplace}} ?$	$\xrightarrow{\text{invlaplace}} ?$
Преобразование Фурье	$\xrightarrow{\text{fourier}} ?$	$\xrightarrow{\text{invfourier}} ?$
z-преобразование	$\xrightarrow{\text{ztrans}} ?$	$\xrightarrow{\text{invztrans}} ?$

$$\sin(\omega \cdot t) \xrightarrow{\text{laplace}} \frac{\omega}{s^2 + \omega^2} \xrightarrow[\text{assume, } \omega > 0]{\text{invlaplace}} \sin(t \cdot \omega)$$

## 17. Определение, построение таблиц значений и графиков функций

Для того чтобы представить функцию в виде таблицы значений, в качестве её аргумента используют ранжированную переменную. (Поэтому часто употребляется термин «дискретный аргумент»)

ПРИМЕР\_Получить таблицу значений функции  $y = \arctg^3 x + b$

для  $b = 1,75$ ;  $x \in [0,1; 2,2]$  с шагом  $\Delta x = 0,3$ .

Решение:

1) присвоить заданное значение переменной b:

$b := 1.75$

2) задать аргумент x как ранжированную переменную:

$x := 0.1, 0.4 .. 2.2$

3) ввести выражение для функции y:

– после имени функции в скобках обязательно указать тот аргумент, который задан как ранжированная переменная (в данном примере это x):

$y(x) := \blacksquare$

– в маркер ввода ввести шаблон встроенной функции  $\text{atan}(\blacksquare)$  (это arctg на языке MathCAD), с помощью одного из инструментов Вставка функции

$y(x) := \text{atan}(\blacksquare)$

– в маркер ввода ввести аргумент x, затем оставшуюся часть выражения:

$y(x) := \text{atan}(x)^3 + b$

4) в следующей строке ввести с клавиатуры  $x=$ . При этом сразу появится таблица значений аргумента  $x$ . Затем в этой же строке ввести  $y(x) =$ , появится таблица значений функции  $y$ :

$x =$	$y(x) =$
0.1	1.751
0.4	1.805
0.7	1.978
1	2.234
1.3	2.516
1.6	2.787
1.9	3.032
2.2	3.248

Построение графиков

ПРИМЕР. Построить график функции

$$f(x) = \sqrt{x} e^{ax} - 71 \text{ при } a=0,55;$$

$$x \in [0,5; 8] \text{ с шагом } \Delta x = 0,25.$$

Решение:

1) присвоить заданное значение переменной  $a$ :

$$a := 0.55$$

2) задать аргумент  $x$  как ранжированную переменную

$$x := 0.5, 0.75 \dots 8$$

3) ввести функцию  $f(x)$  (использовать встроенную функцию  $\exp(\blacksquare)$ ):

$$f(x) := \sqrt{x} \exp(ax) - 71$$

4) вставить шаблон двумерного декартова графика. При этом появляется пустая область графика с двумя маркерами ввода;

5) в маркер ввода под ось абсцисс в центре ввести имя аргумента  $x$ ; в маркер ввода левее оси ординат ввести  $f(x)$ ; щёлкнуть вне области графика.

6) увеличить область графика. Для этого выделить ее штриховой рамкой. Навести указатель мыши на нижний правый угол рамки, добившись, чтобы он принял форму  $\swarrow$ . При нажатой ЛКМ протягивать указатель вниз и вправо.

Форматирование графика

Форматирование графика подразумевает изменение внешнего вида, оформления элементов графической области (осей, координатной сетки, кривой, отображающей зависимость  $y(x)$ , шкалы, маркеров и т.д.).

Для форматирования графика следует:

1) выделить область графика щелчком (появляется синяя рамка);

2) с помощью одного из инструментов: меню Декартов график  $\Rightarrow$  Формат или двойного щелчка по области графика вызвать диалоговое окно Форматирование текущего X-Y графика;

3) на вкладке X-Y Оси щелчком мыши установить параметры: наличие линий сетки и число интервалов сетки на осях; выделение координатной оси абсцисс (Репер); наличие цифровых данных по осям и т. д.;

4) на вкладке Графики установить параметры: тип графика, стиль, цвет, толщина линии, наличие маркеров;

5) на вкладке Надписи ввести заголовок к графику и надписи к осям.

ПРИМЕР. Построить в той же графической области (созданной в примере 2) график функции  $g(x) = x^3 \cos 3x$

Решение:

1) охватить синим курсором-рамкой ранее введенное обозначение функции  $f(x)$  слева от оси ординат (рис. 1.4);

2) нажать клавишу  $\boxed{\cdot}$ . При этом ниже появляется маркер ввода.

3) в маркер ввода ввести  $g(x)$  и щёлкнуть вне области графика.

### 19. Действия с матрицами в MathCAD

Рассмотрим несколько матриц:

$$M := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad S := \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad R(\theta) := \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Как видно, они могут включать в себя числа, символы и даже функции. Они также могут содержать текстовые элементы (строки).

Элемент матрицы можно вывести, используя подстрочные индексы:

$$M_{2,2} \rightarrow 4 \quad S_{1,2} \rightarrow 1 \quad R\left(\frac{\pi}{6}\right)_{1,1} \rightarrow \frac{\sqrt{3}}{2}$$

Матрицы выше являются квадратными  $2 \times 2$ , но у них может быть любой размер по строкам и столбцам:

$[-3]$	массив $1 \times 1$ : обычное число
$u := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$	массив $3 \times 1$ : вектор-столбец
$v := [4 \ 5 \ 6]$	массив $1 \times 3$ : вектор-строка
$W := \begin{bmatrix} 1 & -7 & 13 \\ 3 & 2 & 6 \\ -4 & 11 & 0 \end{bmatrix}$	матрица $3 \times 3$
$\begin{bmatrix} p & q \\ r & s \\ t & u \\ v & w \end{bmatrix}$	матрица $4 \times 2$

Запомните: первое число – номер строки (или их количество), второе – столбца.

Элементы, выделенные с помощью подстрочных индексов:

$$u_{2,1} = 2 \quad v_{1,3} = 6 \quad W_{3,2} = 11$$

Для вектор-столбца второй индекс можно опустить, но не для вектор-строки:

$$u_2 = 2 \quad v_2 = ?$$

Значение должно быть вектором.

Во вкладке Математика  $\rightarrow$  Операторы и символы  $\rightarrow$  Операторы  $\rightarrow$  Векторы и матрицы Вы найдете команды для выделения столбцов и строк:

Операторы	Символы	if	π	x→	Об
<b>Алгебра</b>					
$ x $	+	,	/	$x^y$	
$x!$	()	÷	•	–	
$\sqrt{x}$	%				
<b>Векторы и матрицы</b>					
$\times$	$\ x\ $	$[ ]$	$M^{(i)}$	$M,$	
$M^{\wedge}$	$M^T$	1..n	1,3..n	$\vec{v}$	
<b>Математический анализ</b>					
$\oplus$	*	$d/dx$	$\int dx$	$\lim$	
=	$f'$	$\prod$	$\sum$		
<b>Определение и вычисление</b>					
$:=$	=	$\equiv$	$\rightarrow$		
<b>Проектирование</b>					
$\bar{z}$	°	$\angle$	•		
<b>Сравнение</b>					
$\in$	=	$\oplus$	>	$\geq$	
<	$\leq$	$\wedge$	$\neg$	$\vee$	
$\neq$					

$$W^{(1)} = \begin{bmatrix} 1 \\ 3 \\ -4 \end{bmatrix} \quad W^{\wedge} = [1 \ -7 \ 13]$$

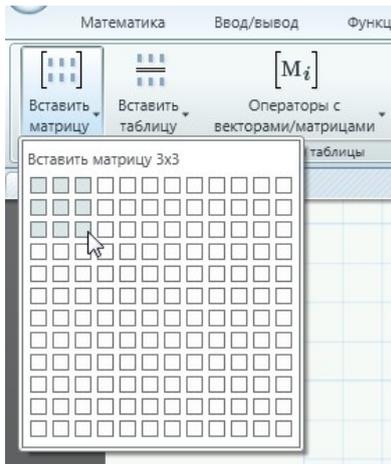
Многие операции для векторов и матриц аналогичны операциям для обычных чисел, переменных и функций: сложение, вычитание, некоторые виды умножения. Поиск обратной матрицы близко к операции деления. Вы можете записать эти операторы, используя имена векторов и матриц. В качестве примера рассмотрим векторное произведение матрицы и вектора:

$$W \cdot u = \begin{bmatrix} 26 \\ 25 \\ 18 \end{bmatrix}$$

Мы рассмотрим эту операцию подробнее позже. Однако стоит заметить, что она требует девять операций умножения и девять – сложения. Расписывать их утомительно и чревато ошибками – для больших матриц сделать это очень трудно.

Применение векторов очень широко. Вспомните пиксели на экране монитора – их могут быть миллионы. Они обрабатываются с помощью операций с матрицами.

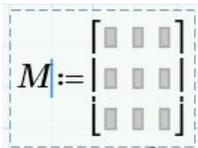
Для создания вектора или матрицы откройте вкладку Матрицы/таблицы. Когда курсор находится в пустой области щелкните по самой левой кнопке «Вставить матрицу». Появится сетка с маленькими квадратами:



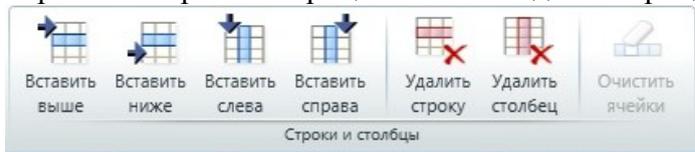
Перемести указатель на сетку, выберите желаемый размер матрицы, затем щелкните левой кнопкой мыши. Появится пустая матрица:



Матрице можно присвоить имя, щелкнув на левую скобку, нажав  $[:]$  для оператора присваивания и введя имя:



Вставку и удаление строк и столбцов легко осуществлять с помощью команд из меню «Операторы с векторами/матрицами» на вкладке Матрицы и таблицы:



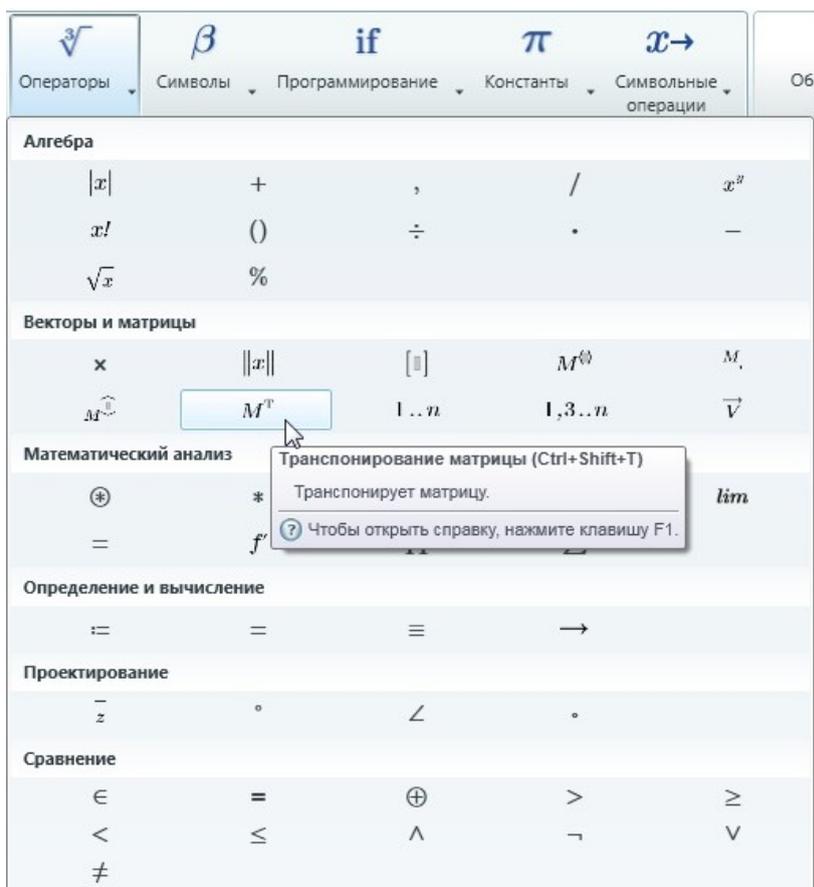
### Операции с матрицами

Эффект от различных операций с матрицами и векторами будет проще понять, используя символы. Будем использовать две матрицы и два вектора:

$$A := \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad B := \begin{bmatrix} p & q \\ r & s \end{bmatrix} \quad v := \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad w := \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

#### Транспонирование

Оператор транспонирования находится на вкладке Математика → Операторы → Векторы и матрицы:



Щелкните по правой границе матрицы и примените оператор. Он работает как для символьных, так и для числовых матриц:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T \rightarrow \begin{bmatrix} a & c \\ b & d \end{bmatrix} \quad \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}^T = [1 \ 2 \ 3]$$

### Поэлементные операции

Часто операции в векторах приходится совершать поэлементно. Для этого служит оператор векторизации. Операции в Excel зачастую являются поэлементными, они также важны и в Mathcad. Чтобы перемножить два вектора поэлементно, сначала введите простое умножение:

$$v \cdot w$$

Затем выберите все выражение и примените векторизацию:

$$\vec{v \cdot w}$$

Вычислите, чтобы посмотреть результат: первый элемент умножается на первый, второй – на второй, и т.д.:

$$\vec{v \cdot w} \rightarrow \begin{bmatrix} a \cdot p \\ b \cdot q \\ c \cdot r \end{bmatrix}$$

Другие поэлементные операции:

$$\vec{v} + \vec{w} \rightarrow \begin{bmatrix} a+p \\ b+q \\ c+r \end{bmatrix} \quad \vec{v} - \vec{w} \rightarrow \begin{bmatrix} a-p \\ b-q \\ c-r \end{bmatrix} \quad \begin{matrix} \vec{v} \\ \vec{w} \end{matrix} \rightarrow \begin{bmatrix} a \\ p \\ b \\ q \\ c \\ r \end{bmatrix}$$

Поэлементные операции применимы только к массивам одного размера.

*Сложение и вычитание*

Сложение и вычитание выполняется поэлементно:

$$\vec{v} + \vec{w} \rightarrow \begin{bmatrix} a+p \\ b+q \\ c+r \end{bmatrix} \quad \vec{v} - \vec{w} \rightarrow \begin{bmatrix} a-p \\ b-q \\ c-r \end{bmatrix}$$

Эта операция также применима лишь к массивам одного размера.

С помощью оператора суммирования можно найти сумму всех элементов вектора (не матрицы):

$$\sum v \rightarrow a + b + c$$

*Скалярное произведение*

Умножение на константу работает так:

$$3 \cdot A \rightarrow \begin{bmatrix} 3 \cdot a & 3 \cdot b \\ 3 \cdot c & 3 \cdot d \end{bmatrix}$$

При скалярном умножении матриц происходит умножение строк на столбцы. При этом используется тот же символ, что и при обычном умножении. Эта операция допустима только для тех матриц, в которых число строк в первой матрице равно числу столбцов во второй. Для наших матриц 2x2:

$$A \cdot B \rightarrow \begin{bmatrix} a \cdot p + b \cdot r & a \cdot q + b \cdot s \\ c \cdot p + d \cdot r & c \cdot q + d \cdot s \end{bmatrix}$$

Заметьте, что последовательность множителей играет роль:

$$B \cdot A \rightarrow \begin{bmatrix} a \cdot p + c \cdot q & b \cdot p + d \cdot q \\ a \cdot r + c \cdot s & b \cdot r + d \cdot s \end{bmatrix}$$

Скалярное произведение не коммутативно, за исключением особых случаев:

$$A \cdot B \neq B \cdot A$$

Скалярное произведение двух векторов дает результат с комплексно-сопряженными числами (с чертой сверху). Для действительных чисел на это можно не обращать внимания:

$$v \cdot w \rightarrow a \cdot \bar{p} + b \cdot \bar{q} + c \cdot \bar{r} \quad w \cdot v \rightarrow p \cdot \bar{a} + q \cdot \bar{b} + r \cdot \bar{c}$$

*Векторное произведение*

Этот оператор применим только для двух вектор-столбцов, состоящих из трех элементов:

$$v \times w \rightarrow \begin{bmatrix} b \cdot r - c \cdot q \\ c \cdot p - a \cdot r \\ a \cdot q - b \cdot p \end{bmatrix} \quad w \times v \rightarrow \begin{bmatrix} c \cdot q - b \cdot r \\ a \cdot r - c \cdot p \\ b \cdot p - a \cdot q \end{bmatrix}$$

Векторное произведение имеет широкое применение в механике, гидродинамике, электромагнетизме и в других областях.

*Обратная матрица*

Обратная матрица определяется только для квадратных матриц:

$$A^{-1} \rightarrow \begin{bmatrix} \frac{d}{a \cdot d - b \cdot c} & -\frac{b}{a \cdot d - b \cdot c} \\ -\frac{c}{a \cdot d - b \cdot c} & \frac{a}{a \cdot d - b \cdot c} \end{bmatrix}$$

Произведение матрицы и ее обратной матрицы является *единичной* матрицей:

$$A \cdot A^{-1} \xrightarrow{\text{simplify}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Произведение матрицы и единичной матрицы дает изначальную матрицу:

$$A \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

*Определитель*

$$A \rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad |A| \rightarrow a \cdot d - b \cdot c$$

Определитель можно найти только для квадратной матрицы. Его значение может быть равно нулю, даже если все элементы матрицы не равны нулю. Обратная матрица содержит дроби, в знаменателе которых находится определитель:

$$A^{-1} \rightarrow \begin{bmatrix} \frac{d}{a \cdot d - b \cdot c} & -\frac{b}{a \cdot d - b \cdot c} \\ -\frac{c}{a \cdot d - b \cdot c} & \frac{a}{a \cdot d - b \cdot c} \end{bmatrix} \quad \begin{bmatrix} \frac{d}{|A|} & -\left(\frac{b}{|A|}\right) \\ -\left(\frac{c}{|A|}\right) & \frac{a}{|A|} \end{bmatrix}$$

Если определитель равен нулю, обратной матрицы не существует, а матрица является сингулярной. Вспомните деление на ноль в обычной алгебре. Mathcad сообщит, если матрица является сингулярной:

Для скаляра определитель равен его модулю:

$$|-3| = 3$$

Для вектора команда Определитель вычисляет длину вектора:

$$v \rightarrow \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad |v| \rightarrow \sqrt{|a|^2 + |b|^2 + |c|^2}$$

## 20. Решение систем алгебраических уравнений в MathCAD

Рассмотрим системы линейных алгебраических уравнений в Mathcad в векторно-матричной форме  $A \cdot x = b$ , где  $A$  – квадратная матрица коэффициентов при неизвестных, причем определитель матрицы должен быть отличным от нуля;  $x$  – вектор неизвестных;  $b$  – вектор свободных членов. Решение данной сводится к следующему. Если определитель матрицы  $A$  отличен от нуля, то матрица  $A$  обратима. Тогда, умножив левую и правую часть и сходного уравнения на обратную мат-

рицу (A-1), получаем решение в виде  $x=A^{-1} \cdot b$ . Реализовать полученное решение средствами Mathcad не представляет сложностей. Для случая бесконечного множества решений получаем сингулярную матрицу, Mathcad выдает сообщение “Matrix is singular. Cannot compute its inversy – Матрица сингулярная. Нельзя вычислить эту инверсию” и прерывает вычисления. На листинге представлен пример решения систем линейных алгебраических уравнений.

$$\begin{aligned} x + 3 \cdot y &= 5 \\ 3 \cdot x - 5 \cdot y &= 1 \end{aligned} \quad A := \begin{pmatrix} 1 & 3 \\ 3 & -5 \end{pmatrix} \quad b := \begin{pmatrix} 5 \\ 1 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} 0.357 & 0.214 \\ 0.214 & -0.071 \end{pmatrix} \quad z := A^{-1} \cdot b \quad z = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \text{solution} \quad x=2 \ y=1$$

## 21. Решение задач линейного программирования в MathCAD

В Mathcad к классу задач линейного программирования относятся задачи, в которых требуется оптимизировать (определить максимум или минимум) целевую функцию вида

$$z = f(x_1, x_2, \dots, x_n) = c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n$$

при следующих ограничениях

$$a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \dots + a_{in} \cdot x_n - b_i \geq 0, \quad (i = 1, 2, \dots, m)$$

$$x_k \geq 0, \quad (k = 1, 2, \dots, n)$$

Одним из способов решение подобных задач в Mathcad является использование блока **given** с функциями **minimize** и **maximize**. В качестве примера рассмотрим задачу планирования производства красок. Суть задачи в следующем. Фабрика выпускает два типа красок - **I** и **E**. Для производства красок используются два компонента - **A** и **B**. Максимальные суточные запасы этих компонентов: компонент **A** - **6 тонн**; компонент **B** – **8 тонн**. Расходы компонентов **A** и **B** на производство 1 тонны краски следующие: для краски **I** - **A/B= 2/1**; для краски **E** - **A/B = 1/2**. Суточный спрос на краску **I** никогда не превышает спрос на краску **E** более, чем на **1** тонну. Спрос на краску **I** никогда не превышает **2 тонн** в сутки. Оптовые цены на краску **I** – **2000 рублей** за тонну, а на краску **E** – **3000** рублей за тонну. Определить максимальный доход фабрики от продажи краски. Обозначив суточный объем выпуска краски **I** за **x1**, а суточный объем выпуска краски **E** за **x2**, получаем экономико - математическую модель задачи, вид которой и листинг ее решения в Mathcad приведены на листинге.

## Решение задачи линейного программирования.

ORIGIN := 1      Задание целевой функции  
 $F(x) := 2000 \cdot x_1 + 3000 \cdot x_2$   
 Задание начальных приближений       $x := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$   
 Given  
 Ввод ограничений  
 $x_2 + 2 \cdot x_1 \leq 6$   
 $2 \cdot x_2 + x_1 \leq 8$   
 $x_1 - x_2 \leq 1$   
 $x_1 \leq 2$   
 $x_1 \geq 0$   
 $x_2 \geq 0$   
 Решение  
 $x := \text{Maximize}(F, x) \quad x = \begin{pmatrix} 1.333 \\ 3.333 \end{pmatrix}$   
 Значение функции цели       $F(x) = 1.267 \times 10^4$

## 22. Аппроксимация эмпирических данных и метод наименьших квадратов в MathCAD

Наиболее распространенным методом аппроксимации экспериментальных данных является метод наименьших квадратов. Метод позволяет использовать аппроксимирующие функции произвольного вида и относится к группе глобальных методов. Критерием близости в методе наименьших квадратов является требование минимальности суммы квадратов отклонений от аппроксимирующей функции до экспериментальных точек:

$$\Phi = \sum_{i=1}^n (y_i - f(x_i))^2 \rightarrow \min$$

Таким образом, не требуется, чтобы аппроксимирующая функция проходила через все заданные точки, что особенно важно при аппроксимации данных, заведомо содержащих погрешности.

Важной особенностью метода является то, что аппроксимирующая функция может быть произвольной. Ее вид определяется особенностями решаемой задачи, например, физическими соображениями, если проводится аппроксимация результатов физического эксперимента. Наиболее часто встречаются аппроксимация прямой линией (линейная регрессия), аппроксимация полиномом (полиномиальная регрессия), аппроксимация линейной комбинацией произвольных функций. Кроме того, часто бывает возможно путем замены переменных свести задачу к линейной (провести линеаризацию). Например, пусть аппроксимирующая функция ищется в виде  $y = A \exp(kx)$ . Прологарифмируем это выражение и введем обозначения  $z = \ln(y)$ ,  $a = \ln(A)$ . Тогда в новых обозначениях задача сводится к отысканию коэффициентов линейной функции  $z = a + kx$ .

Пусть данные некоторого эксперимента представлены в виде таблицы значений переменных  $x$  и  $y$ .

Можно поставить задачу об отыскании аналитической зависимости между переменными, то есть некоторой формулы  $y = f(x)$ , явным образом выражающей зависимость от  $x$ . Естественно требовать, чтобы график искомой функции  $y = f(x)$  изменялся плавно и не слишком уклонялся от экспериментальных точек  $(x_i, y_i)$ . Поиск такой функциональной зависимости называют “сглаживанием” экспериментальных данных.

Задачу о сглаживании экспериментальных данных можно решать методом наименьших квадратов. Согласно методу наименьших квадратов, указывается формула:

$$y = Q(x, a_0, a_1, \dots, a_n),$$

где  $a_0, a_1, \dots, a_n$  - числовые параметры.

Наилучшими значениями параметров  $a_0, a_1, \dots, a_n$  (которые обозначим  $\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n$ ) считаются те, для которых сумма квадратов отклонений функции  $Q(x, a_0, a_1, \dots, a_n)$  от экспериментальных точек  $(x_i, y_i)$  является минимальной, то есть функция  $S(a_0, a_1, \dots, a_n)$ , которая в точке  $(\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n)$  достигает минимума.

$$S(a_0, a_1, \dots, a_n) = \sum_{i=1}^n (Q(x_i, a_0, a_1, \dots, a_n) - y_i)^2$$

Отсюда, используя необходимые условия экстремума функции нескольких переменных, получаем систему уравнений для определения параметров  $\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n$ :

$$\frac{\partial S}{\partial a_0} = 0, \frac{\partial S}{\partial a_1} = 0, \dots, \frac{\partial S}{\partial a_n} = 0.$$

Если система имеет единственное решение  $\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n$ , то оно является искомым и аналитическая зависимость между экспериментальными данными определяется формулой  $y = f(x) = Q(x, \tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n)$ . Заметим, что в общем случае эта система нелинейная.

Рассмотрим подробнее аппроксимирующие зависимости с двумя параметрами:  $y = Q(x, \alpha, \beta)$ . Используя соотношения и опуская несложные выкладки, получим систему двух уравнений с двумя неизвестными  $\alpha$  и  $\beta$ :

$$\begin{cases} \sum_{i=1}^n (Q(x_i, \alpha, \beta) - y_i) \frac{\partial Q(x_i, \alpha, \beta)}{\partial \alpha} = 0, \\ \sum_{i=1}^n (Q(x_i, \alpha, \beta) - y_i) \frac{\partial Q(x_i, \alpha, \beta)}{\partial \beta} = 0 \end{cases}$$

В частном случае аппроксимации экспериментальных данных с помощью линейной функции имеем:

$$y = Q(k, x, b) = kx + b, \frac{\partial Q}{\partial k} = x, \frac{\partial Q}{\partial b} = 1$$

Система для этого случая является линейной относительно неизвестных  $k$  и  $b$ :

$$\begin{cases} \sum_{i=1}^n (kx_i + b) - y_i = 0, \\ \sum_{i=1}^n (kx_i + b) \cdot x_i - y_i x_i = 0 \end{cases} \Rightarrow \begin{cases} bn + k \sum_{i=1}^n x_i = \sum_{i=1}^n y_i, \\ b \sum_{i=1}^n x_i + k \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i \end{cases}$$

Если для переменных  $x$  и  $y$  соответствующие значения экспериментальных данных  $(x_i, y_i)$  не располагаются вблизи прямой, то выбирают новые переменные так, чтобы преобразованные экспериментальные данные  $X_i = \phi(x_i, y_i), Y_i = \psi(x_i, y_i)$  в новой системе координат  $(x, y)$  давали бы точки  $(x_i, y_i)$ , менее уклоняющиеся от прямой.

$$X = \phi(x, y), Y = \psi(x, y)$$

Для аппроксимирующей прямой  $Y = kX + B$  коэффициенты можно определить из уравнений (6), где вместо  $x$  и  $y$  подставляют соответствующие значения  $X_i$  и  $Y_i$ . Нахождение зависимостей вида (7) называют, выравниваем экспериментальных данных. Функциональная зависимость  $y = f(x)$  определена неявно уравнением  $\psi(x; y) = k\psi(x; y) + b$  разрешимым относительно  $y$  в частных случаях.

### 23. Решение задач математического анализа в MathCAD

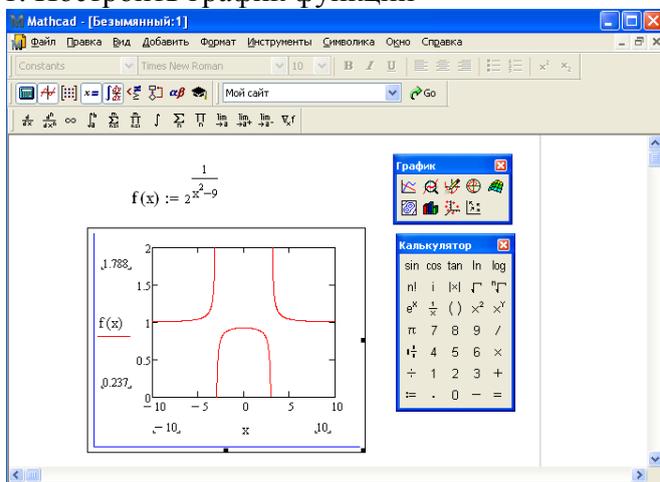
Задание №1:

Исследовать на непрерывность функцию  $f(x)$  в точках 3, -3, 0.

1. Необходимо построить график функции.
2. Исследовать на сходимость в каждой точке при помощи пределов функций в данных точках.

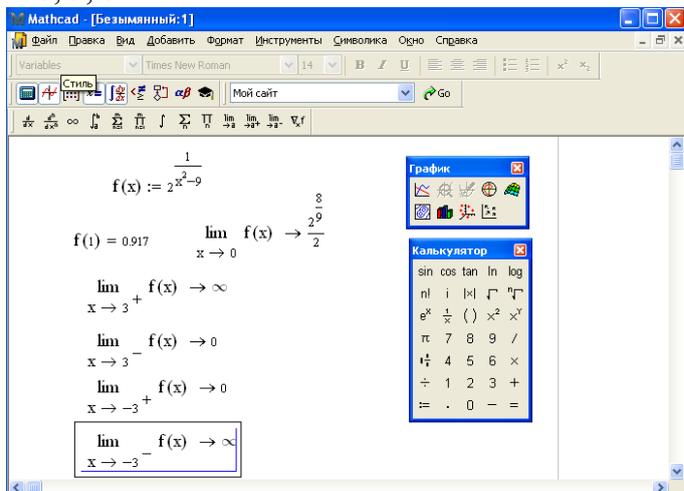
$$f(x) = 2^{x^2 - 9}$$

1. Построить график функции



2. Исследовать значение функции на концах отрезка, при помощи пределов функции можно

с помощью кнопок на панели инструментов  $\lim_{x \rightarrow a}$   $\lim_{x \rightarrow a^+}$   $\lim_{x \rightarrow a^-}$  :  
 $x=0, 3, -3$



Задание №2:

Записать уравнение касательной к функции  $f(x)$  в точке  $x_0$ .

1. Задать функцию и точку касания.

2. Найти значение производной функции в точке. Производную можно найти, воспользо-

вавшись кнопками на панели инструментов  $\frac{d}{dx}$   $\frac{d^n}{dx^n}$ :

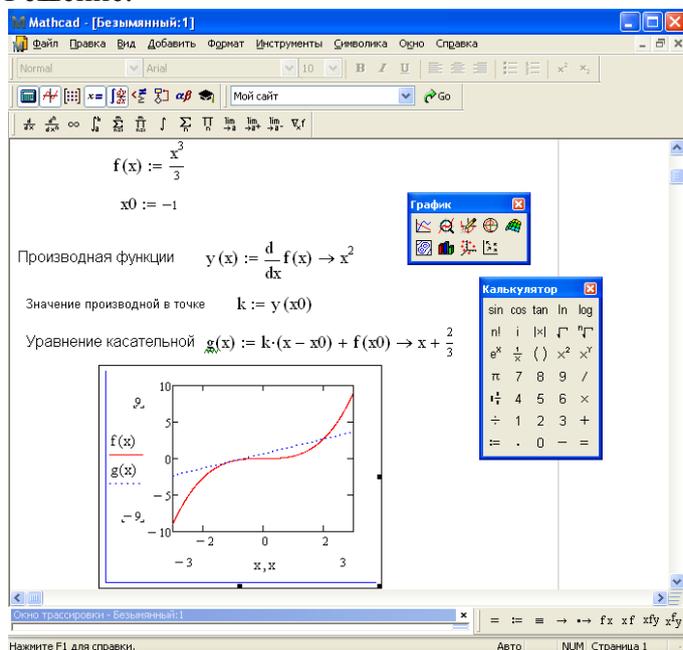
3. Составить уравнение касательной функции.

4. Построить график функции касательной.

$$f(x) = \frac{x^3}{3} \text{ в точке с абсциссой } x = -1$$

Написать уравнение касательной к кривой

Решение:



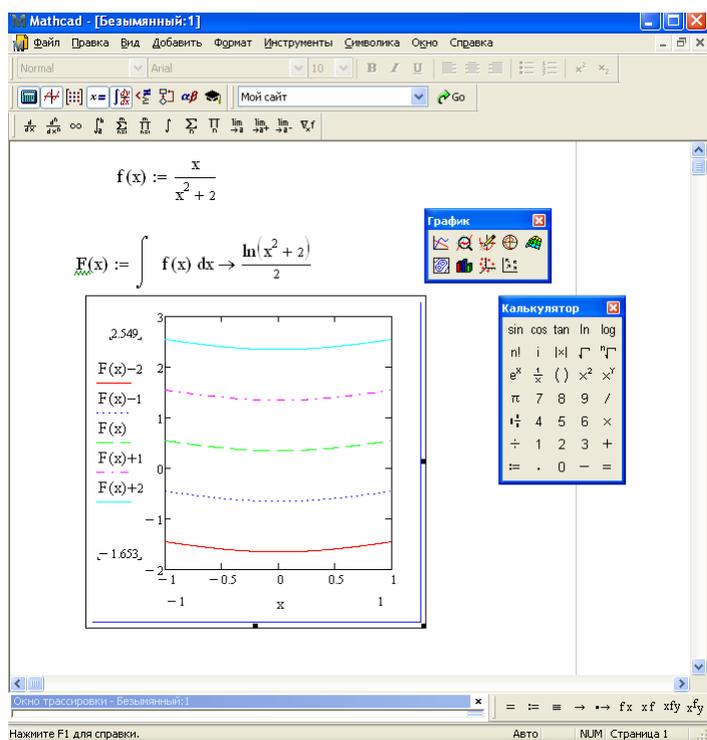
Задание №3:

Построить семейство интегральных линий функции  $f(x)$ .

$$f(x) := \frac{1}{\sqrt{1+4x-3x^2}}$$

( решение на рис.). Интеграл можно найти при помощи кнопок на

панели инструментов:  $\int_a^b$ ,  $\int$ .



Задания для самостоятельного выполнения:

1. Написать уравнение касательных к гиперболе  $y = \frac{4}{x}$  в точках 1 и 4.
2. Написать уравнение касательной и нормали к кривой  $y = \frac{8}{4+x^2}$  в точке 2.
3. Вычислить площадь фигуры, ограниченной линиями  $y^2 = 9x$  и  $y = 3x$ . Площадь фигуры вычислить с помощью определенного интеграла.
4. Исследовать функцию  $f(x) = e^{(x+1/x)}$  на непрерывность в точках  $x=0$  и  $x=1$ . Построить график.
5. Написать уравнение касательной, проведенной к кривой  $y(x) = 2x^2 - 5x + 1$ . Построить график функции и касательной.
6. Найти производную функции  $y(x)$ , заданной параметрическими уравнениями  $x = \ln(1+t^2)$ ,  $y = t - \arctg(t)$ .
7. Найти производную функции  $y$ , заданной неявно  $\sin(x \cdot y) + \cos(x \cdot y) = \tg(x + y)$ .
8. Провести полное исследование функции  $y(x) = x^4 + 8x^3 + 18x^2 + 8$ . Найти точки пересечения с осями, точки экстремума, интервалы выпуклости и вогнутости и асимптоты, построить график.
9. Провести полное исследование функции  $y(x) = (x+2)e^{-x}$ . Найти точки пересечения с осями, точки экстремума, интервалы выпуклости и вогнутости и асимптоты, построить график.
10. Построить семейство интегральных линий функции  $f(x) = \ln(1+x^2)$ .
11. Вычислить площадь фигуры, ограниченной линиями  $y^2 = 2x+1$  и  $x - y - 1 = 0$ .
12. Найти интервал сходимости ряда  $\sum_{n=0}^{\infty} \frac{x^n}{n \cdot 3^n}$ .
13. Вычислить интеграл:

a)  $\int \frac{\cos(x)}{x} dx$

б)  $\int_e^{e^2} \frac{\sqrt{\ln x}}{x} dx$

в)  $\int_0^1 \frac{2x}{1+x^2} dx$

г)  $\int_1^{\sqrt{2}} \frac{xdx}{\sqrt{4-x^2}}$

д)  $\int_1^2 \ln(3x+2) dx$

### 3.МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная работа представляет собой особую, высшую степень учебной деятельности. Она обусловлена индивидуальными психологическими различиями обучающегося и личностными особенностями и требует высокого уровня самосознания, рефлексивности. Самостоятельная работа может осуществляться как во внеаудиторное время (дома, в лаборатории), так и на аудиторных занятиях в письменной или устной форме.

Самостоятельная работа обучающихся является составной частью учебной работы и имеет целью закрепление и углубление полученных знаний и навыков, поиск и приобретение новых знаний, в том числе с использованием автоматизированных обучающих систем, а также выполнение учебных заданий, подготовку к предстоящим занятиям, зачетам и экзаменам. Организуется, обеспечивается и контролируется данный вид деятельности студентов соответствующими кафедрами.

Самостоятельная работа предназначена не только для овладения дисциплиной, но и для формирования навыков самостоятельной работы вообще, в учебной, научной, профессиональной деятельности, способности принимать на себя ответственность, самостоятельно решить проблему, находить конструктивные решения, выход из кризисной ситуации и т. д. Значимость самостоятельной работы выходит далеко за рамки отдельного предмета, в связи с чем выпускающие кафедры должны разрабатывать стратегию формирования системы умений и навыков самостоятельной работы. При этом следует исходить из уровня самостоятельности абитуриентов и требований к уровню самостоятельности выпускников, с тем, чтобы за весь период обучения достаточный уровень был достигнут.

При проведении самостоятельной работы, связанной с проработкой теоретического материала, студентам предлагается законспектировать рассматриваемый вопрос, в случае необходимости задать возникшие вопросы на практическом занятии или на консультации.

При изучении дисциплины практикуются следующие виды и формы самостоятельной работы студентов:

- подготовка к устному опросу по темам лабораторных работ;
- выполнение индивидуальных заданий по темам лабораторных работ;
- подготовка к зачету.

Самостоятельная работа тесно связана с контролем (контроль также рассматривается как завершающий этап выполнения самостоятельной работы), при выборе вида и формы самостоятельной работы следует учитывать форму контроля.

Формы контроля при изучении дисциплины:

- устный опрос;
- проверка выполнения индивидуальных заданий по темам лабораторных работ.

Самостоятельная работа проводится в виде подготовительных упражнений для усвоения нового, упражнений при изучении нового материала, упражнений в процессе закрепления и повторения, упражнений проверочных и контрольных работ, а также для самоконтроля.

Для организации самостоятельной работы необходимы следующие условия:

- готовность студентов к самостоятельному труду;
- наличие и доступность необходимого учебно-методического и справочного материала;
- консультационная помощь.

Самостоятельная работа может проходить в лекционном кабинете, компьютерном зале, библиотеке, дома. Самостоятельная работа тренирует волю, воспитывает работоспособность, внимание, дисциплину и т.д.

Рекомендации по организации аудиторной самостоятельной работой

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Основными видами аудиторной самостоятельной работы являются:

- выполнение лабораторных работ по инструкциям;
- работа с литературой и другими источниками информации, в том числе электронными;

само- и взаимопроверка выполненных заданий;

Выполнение лабораторных работ осуществляется на лабораторных занятиях в соответствии с графиком учебного процесса. Работа с литературой, другими источниками информации, в т.ч. электронными может реализовываться на лекционных и практических занятиях. Данные источники информации могут быть представлены на бумажном и/или электронном носителях, в том числе, в сети Internet. Преподаватель формулирует цель работы с данным источником информации, определяет время на проработку документа и форму отчетности.

Само- и взаимопроверка выполненных заданий чаще используется на лекционном, практическом занятии и имеет своей целью приобретение таких навыков как наблюдение, анализ ответов сокурсников, сверка собственных результатов с эталонами.

Рекомендации по организации внеаудиторной самостоятельной работы

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя, но без его непосредственного участия.

При предъявлении видов заданий на внеаудиторную самостоятельную работу рекомендуется использовать дифференцированный подход к уровню подготовленности обучающегося. Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит консультацию с определением цели задания, его содержания, сроков выполнения, ориентировочного объема работы, основных требований к результатам работы, критериев оценки, форм контроля и перечня литературы. В процессе консультации преподаватель предупреждает о возможных типичных ошибках, встречающихся при выполнении задания.

Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня подготовленности обучающихся.

Видами заданий для внеаудиторной самостоятельной работы могут быть:

для овладения знаниями: чтение текста (учебника, первоисточника, дополнительной литературы); составление плана текста; графическое изображение структуры текста; конспектирование текста; выписки из текста; работа со словарями и справочниками; учебно-исследовательская работа; использование аудио- и видеозаписей, компьютерной техники и Интернет-ресурсов и др.;

для закрепления и систематизации знаний: работа с конспектом лекции (обработка текста); повторная работа над учебным материалом (учебника, первоисточника, дополнительной литературы, аудио- и видеозаписей); составление плана и тезисов ответа; составление таблиц, глоссария для систематизации учебного материала; изучение словарей, справочников; ответы на контрольные вопросы; аналитическая обработка текста (аннотирование, рецензирование, реферирование, контент-анализ и др.); подготовка сообщений к выступлению на семинаре, конференции; подготовка рефератов, докладов; составление библиографии, заданий в тестовой форме и др.;

для формирования умений: решение задач и упражнений по образцу; решение вариативных задач и упражнений; составление схем; проектирование и моделирование разных видов и компонентов профессиональной деятельности и др.

Для обеспечения внеаудиторной самостоятельной работы по дисциплине преподавателем разрабатывается перечень заданий для самостоятельной работы, который необходим для эффективного управления данным видом учебной деятельности обучающихся.

Преподаватель осуществляет управление самостоятельной работой, регулирует ее объем на одно учебное занятие и осуществляет контроль выполнения всеми обучающимися группы. Для удобства преподаватель может вести ведомость учета выполнения самостоятельной работы, что позволяет отслеживать выполнение минимума заданий, необходимых для допуска к итоговой аттестации по дисциплине.

В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления и становится активным самостоятельным субъектом учебной деятельности.

Обучающийся самостоятельно определяет режим своей внеаудиторной работы и меру труда, затрачиваемого на овладение знаниями и умениями по каждой дисциплине, выполняет внеа-

удиторную работу по индивидуальному плану, в зависимости от собственной подготовки, бюджета времени и других условий.

Ежедневно обучающийся должен уделять выполнению внеаудиторной самостоятельной работы в среднем не менее 3 часов.

При выполнении внеаудиторной самостоятельной работы обучающийся имеет право обращаться к преподавателю за консультацией с целью уточнения задания, формы контроля выполненного задания.

Контроль результатов внеаудиторной самостоятельной работы студентов может проводиться в письменной, устной или смешанной форме с представлением продукта деятельности обучающегося. В качестве форм и методов контроля внеаудиторной самостоятельной работы могут быть использованы зачеты, тестирование, самоотчеты, контрольные работы, защита творческих работ и др.

Методические рекомендации по изучению теоретических основ дисциплин

Изучение теоретической части дисциплин призвано не только углубить и закрепить знания, полученные на аудиторных занятиях, но и способствовать развитию у студентов творческих навыков, инициативы и организовать свое время.

Самостоятельная работа при изучении дисциплин включает:

- чтение студентами рекомендованной литературы и усвоение теоретического материала дисциплины;
- знакомство с Интернет-источниками;
- подготовку к различным формам контроля (тесты, опрос по темам лабораторных работ);
- подготовку ответов на вопросы по различным темам дисциплины в той последовательности, в какой они представлены.

Планирование времени, необходимого на изучение дисциплин, студентам лучше всего осуществлять весь семестр, предусматривая при этом регулярное повторение материала.

Материал, законспектированный на лекциях, необходимо регулярно прорабатывать и дополнять сведениями из других источников литературы, представленных не только в программе дисциплины, но и в периодических изданиях.

При изучении дисциплины сначала необходимо по каждой теме прочитать рекомендованную литературу и составить краткий конспект основных положений, терминов, сведений, требующих запоминания и являющихся основополагающими в этой теме для освоения последующих тем курса. Для расширения знания по дисциплине рекомендуется использовать Интернет-ресурсы; проводить поиски в различных системах и использовать материалы сайтов, рекомендованных преподавателем.

При ответе на зачете необходимо: продумать и четко изложить материал; дать определение основных понятий; дать краткое описание явлений; привести примеры. Ответ следует иллюстрировать схемами, рисунками и графиками.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ощепков А. Ю. Системы автоматического управления: теория, применение, моделирование в MATLAB [Электронный ресурс] : учебное пособие. — Электрон. дан. — СПб. : Лань, 2013. — 208 с. — Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=5848](http://e.lanbook.com/books/element.php?pl1_id=5848) - ЭБС «Лань»
2. Информатика и программирование: компьютерный практикум : учеб. пособие : рек. УМО / А. Н. Гуда [и др.] ; под общ. ред. В. И. Колесникова. - М. : Дашков и К, 2009. - 238 с.
3. Острейковский, В.А. Информатика: учеб. : рек. Мин. обр. РФ / В.А. Острейковский. - 5-е изд., стер. - М. : Высш. шк., 2009. - 512 с.
4. Информатика. Базовый курс: учеб. пособие : рек. Мин. обр. РФ / под ред. С. В. Симоновича. - 2-е изд. - М. : Питер Пресс, 2009- 640 с.
5. Макаров Е.Г. Mathcad: Учебный курс (+CD). – СПб.: Питер, 2009. – 384 с.
6. Несен А.В. Microsoft Word 2010: от новичка к профессионалу. – М.: СОЛОН-ПРЕСС, 2011. – 448 с.
7. Рыбалев, А.Н. Информатика. Специальные главы: пособие к выполнению лаб. работ / А. Н. Рыбалев ; АмГУ, Эн.ф. - Благовещенск : Изд-во Амур. гос. ун-та, 2007. - 42 с.
8. Масловская, А.Г. Методы вычислений: реализация алгоритмов в MATLAB: практикум / А. Г. Масловская , Т. К. Барабаш, Л. В. Чепак ; АмГУ, ФМиИ. - Благовещенск: Изд-во Амур. гос. ун-та, 2010. - 204 с.
9. Шампайн, Л. Ф. Решение обыкновенных дифференциальных уравнений с использованием MATLAB: учеб. пособие / Л. Ф. Шампайн, И. Гладвел, С. Томпсон ; пер. с англ. И. А. Макаров. - СПб. : Лань, 2009. - 300 с.
10. Плохотников, К.Э. Вычислительные методы. Теория и практика в среде Matlab: курс лекций: учеб. пособие : рек. УМО / К.Э. Плохотников. - М. : Горячая линия -Телеком, 2009. - 496 с.
11. Масловская, А.Г. Численные методы. Моделирование на базе MATLAB: практикум: рек. ДВ РУМЦ / А. Г. Масловская, Л. В. Чепак ; АмГУ, ФМиИ. - Благовещенск : Изд-во Амур. гос. ун-та, 2007. - 108 с. : рис., табл. - Библиогр.: с. 105
12. Герман-Галкин, С.Г. Виртуальные лаборатории полупроводниковых систем в среде Matlab-Simulink [Электронный ресурс] : учебно-методическое пособие. — Электрон. дан. — СПб.:Лань,2013.—443 с. — Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=36998](http://e.lanbook.com/books/element.php?pl1_id=36998) — ЭБС «Лань»
13. Охорзин, В.А. Прикладная математика в системе MATHCAD [Электронный ресурс] : учебное пособие. — Электрон. дан. — СПб. : Лань, 2009. — 349 с. — Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=294](http://e.lanbook.com/books/element.php?pl1_id=294) — ЭБС «Лань»