

# ЛАБ. РАБОТА №1. НАЧАЛЬНОЕ ЗНАКОМСТВО С MATLAB

## ФОРМА Бэкуса — Наура.

Форма Бэкуса — Наура (сокр. БНФ) — формальная система описания синтаксиса, в которой одни синтаксические элементы формального языка последовательно определяются через другие элементы.

### Термины БНФ

- *Терминал* (терминальный символ или обобщение понятия «буквы») — объект, непосредственно присутствующий в словах языка, соответствующего грамматике и имеющий конкретное, неизменяемое значение. В формальных компьютерных языках, в качестве терминалов обычно берут стандартные символы ASCII — малые и заглавные латинские буквы, цифры, а также специальные символы.
- *Нетерминал* (нетерминальный символ) — объект, обозначающий какую-либо сущность языка (например: инструкция, формула, арифметическое выражение, и т.д.) и не имеющий конкретного символьного значения. Все нетерминалы записываются с помощью текста раскрывающего их смысл и заключаются в треугольные скобки. Например:

<Натуральное число>

### Специальный нетерминал <Пусто>

Используется для обозначения отсутствия всякого символьного значения.

### Оператор определения.

Используется для присвоения значения терминалу или нетерминалу. Записывается как последовательность следующих терминалов:

::=

### Основное правило БНФ.

Основное правило предполагает, что нетерминал, смысл которого определяется записывается слева от оператора определения, а справа записываются терминалы или нетерминалы смысл которых уже определен или будет определен далее. Например:

<Знак умножения> ::= \*

## **Оператор выбора.**

Используется как способ указать при определении нетерминала выбор только одного нетерминала или терминала из множества в правой части определения. Например:

$\langle \text{Цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

## **Рекурсивное правило БНФ.**

Определение нетерминалов БНФ позволяет определяемый нетерминал записывать как в левой, так и правой части операции определения. Такое правило является рекурсивным и всегда предполагает, что в нем содержится некоторое начальное значение определяемого нетерминала. Например:

$\langle \text{Цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$   
 $\langle \text{Число} \rangle ::= \langle \text{Цифра} \rangle \mid \langle \text{Число} \rangle \langle \text{Цифра} \rangle$

В данном примере часть правила  $\langle \text{Цифра} \rangle$  задает начальное значение для рекурсии, а часть  $\langle \text{Число} \rangle \langle \text{Цифра} \rangle$  показывает правило выполнения самой рекурсии. Наглядно, это можно представить следующими шагами:

Шаг	Часть правила, которая работает	$\langle \text{Цифра} \rangle$	$\langle \text{Число} \rangle$
1	$\langle \text{Цифра} \rangle$	5	5
2	$\langle \text{Число} \rangle \langle \text{Цифра} \rangle$	2	52
3	$\langle \text{Число} \rangle \langle \text{Цифра} \rangle$	7	527
4	$\langle \text{Число} \rangle \langle \text{Цифра} \rangle$	1	5271
5	$\langle \text{Число} \rangle \langle \text{Цифра} \rangle$	8	52718

Продолжая шаги с помощью части правила  $\langle \text{Число} \rangle \langle \text{Цифра} \rangle$  можно сформировать цепочку изображающую натуральное число любой разрядности.

## **Специальное расширение стандарта БНФ**

Помимо классической БНФ в научной литературе также используется расширенная форма. Эта форма вводит некоторые новые правила и обозначения.

В рамках данного цикла лабораторных работ мы позаимствуем из расширенной формы следующее:

### **Условное вхождение.**

Условное вхождение подразумевает возможность использовать или пропускать некоторую часть правила и записывается с помощью квадратных скобок. Квадратные скобки выделяют необязательный

элемент правила, который может присутствовать, а может и отсутствовать. Например, правило вида  $\langle A \rangle ::= [B]$  обозначает, что нетерминал  $A$  либо является пустым, либо состоит из одного символа  $B$ .

### **Специальное обозначение терминальных символов.**

Поскольку для условного вхождения возникла необходимость в специальном смысле квадратных скобок, то терминальный вид квадратных скобок следует определить другим способом. Для такого определения можно воспользуемся следующей нетерминальной формой:

$$\begin{aligned} \langle ' [ ' \rangle \\ \langle ' ] ' \rangle \end{aligned}$$

Аналогичным образом можно определить терминальные символы для треугольных скобок:

$$\begin{aligned} \langle ' < ' \rangle \\ \langle ' > ' \rangle \end{aligned}$$

Кроме того, символ пробела, который легко теряется при использовании в обычной терминальной форме, будет удобно определить в виде:

$$\langle ' \ ' \rangle$$

## **НАЧАЛЬНОЕ ЗНАКОМСТВО С MATLAB.**

### **Типы арифметических чисел в MATLAB.**

#### **1. Арифметическое число.**

$$\begin{aligned} \langle \text{Арифметическое число} \rangle ::= & \langle \text{Натуральное число} \rangle \\ & | \langle \text{Целое со знаком} \rangle \\ & | \langle \text{Действительное число} \rangle \\ & | \langle \text{Комплексное число} \rangle \end{aligned}$$

#### **2. Натуральные числа.**

$$\begin{aligned} \langle \text{Натуральное число} \rangle ::= & \langle \text{Цифра} \rangle \mid \langle \text{Натуральное число} \rangle \langle \text{Цифра} \rangle \\ \langle \text{Цифра} \rangle ::= & 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{aligned}$$

#### **3. Целые числа со знаком.**

$$\begin{aligned} \langle \text{Целое со знаком} \rangle ::= & \langle \text{integer} \rangle \\ \langle \text{integer} \rangle ::= & [ \langle \text{Знак} \rangle ] \langle \text{Натуральное число} \rangle \\ \langle \text{Знак} \rangle ::= & + \mid - \end{aligned}$$

#### 4. Действительные числа

$\langle \text{Действительное число} \rangle ::= \langle \text{Real} \rangle$   
 $\langle \text{Real} \rangle ::= \langle \text{Regular Real} \rangle \mid \langle \text{Exponential Real} \rangle$

#### 5. Действительные числа в обычной форме.

$\langle \text{Regular Real} \rangle ::= \langle \text{Целое со знаком} \rangle . \langle \text{Число} \rangle$

Например:

1.1  
1.38  
602.21412

#### 6. Действительные числа в экспоненциальной форме.

$\langle \text{Exponential Real} \rangle ::= \langle \text{Мантисса} \rangle \mathbf{e} \langle \text{Порядок} \rangle$   
 $\langle \text{Мантисса} \rangle ::= \langle \text{Regular Real} \rangle$   
 $\langle \text{Порядок} \rangle ::= \langle \text{Целое со знаком} \rangle$

**ВНИМАНИЕ!**  $\mathbf{e}$   $\langle \text{Порядок} \rangle$  это значение  $10^{\langle \text{Порядок} \rangle}$ .

Например:

1,602176565e-19 == 1.602176565\*10<sup>-19</sup> (это элементарный заряд)  
1,380650424e-23 == 1.380650424\*10<sup>-23</sup> (это Постоянная Больцмана)  
6,02214129e23 == 6.02214129\*10<sup>23</sup> (это число Авогадро)

#### 7. Комплексные числа.

$\langle \text{Комплексное число} \rangle ::= [ \langle \text{Real} \rangle + ] \langle \text{Real} \rangle \mathbf{i} \mid [ \langle \text{Real} \rangle + ] \langle \text{Real} \rangle \mathbf{j}$   
где:  $\mathbf{i}^2 = -1$ ;  $\mathbf{j}^2 = -1$ ;

Например:

1.1 + 6.2i  
1.1 + 6.2j

#### 8. Символы специальных значений:

$\langle \text{Символ специального значения} \rangle ::= [\text{Знак}] \text{inf} \mid \text{NaN}$

Например:

inf	% infinity или бесконечность
NaN	% Not-a-Number или неопределенность

## **Комментарий. Константы и переменные. Арифметические операции.**

### **1. Комментарий.**

Комментарий это произвольный текст, который не интерпретируется (игнорируется) парсером MatLab и может размещаться в виде отдельных строк или замыкать отдельные строки.

Формально комментарий можно определить так:

<Комментарий> ::= % <' '><Произвольный текст>

Например:

```
% Это комментарий
```

### **2. Арифметические константы в MatLab.**

Константы в MatLab, это самоопределенные терминалы. Другими словами это терминалы, смысл которых явно следует из их изображения.

Например:

```
-123      % Это числовое значение типа <целое со знаком>
```

Однако, кроме самоопределенных терминалов для обозначения констант используются также символические имена, за которыми закрепляются определенные численные значения. Таким образом, константы можно определить как:

```
<Арифметическая константа> ::=
    <Самоопределенный арифметический терминал>
    | <Имя арифметической константы>

<Имя арифметической константы> ::=  pi
                                     |  eps
                                     |  realmin | realmax
                                     |  i  |  j
```

где:

pi = значение числа ПИ

eps = 2<sup>(-52)</sup> или 2.22e-16

realmin = 2<sup>(-1022)</sup> или 2.2251e-308

realmax = 21024 или 1.7977e+308

К символическим именам констант можно также отнести корень из минус единицы, то есть, **i** или **j**, который использовался в определении комплексных чисел.

Значения арифметических констант можно переопределять и восстанавливать. Например:

```

>> pi = 10          % Переопределим значение pi
pi =
    10
>> clear pi         % Восстановим значение pi
>> pi
ans =
    3.1416

% Еще один пример

>> i = 10           % Переопределим значение i
i =
    10
>> clear i          % Восстановим значение i
>> i
ans =
    0 + 1.0000i

```

### 3. Переменные в MatLab.

Забегая несколько вперед, отметим, что все переменные MatLab являются матрицами (детальнее см. лабораторная работа №2). Иными словами, даже переменная, предназначенная для хранения обычного натурального числа это матрица размером 1x1, то есть, матрица с одной ячейкой. Все переменные в MatLab обозначаются именами.

```

<Имя переменной> ::= <Идентификатор>
<Идентификатор> ::= <Буква> | <Идентификатор><Буква>
                    | <Идентификатор><Цифра>
<Буква> ::= <Малые и заглавные буквы латинского алфавита> | <_>

```

### **Выражение. Простое арифметическое выражение.**

Выражения это важнейший способ представить в MatLab символическую запись формул, которые вычисляются для арифметических, логических, текстовых и других значений. Основой для определения синтаксиса выражений служат простые выражения порядок вычисления которых регулируется круглыми скобками или старшинством операций. Поскольку формы Бэкуса – Наура ориентированы только на синтаксическое конструирование, а правила порядка вычисления выражений относятся к семантике (то есть смысловой нагрузке синтаксических форм) такие правила описываются либо дополнительными грамматиками либо реализуются в виде блок – алгоритмических схем. В части дополнительных грамматик, полезно ознакомиться с языком Дика (англ. Dusk language, [https://ru.wikipedia.org/wiki/Язык\\_Дика](https://ru.wikipedia.org/wiki/Язык_Дика)), который удобно использовать при вычислении выражений в скобках.

В нашем случае, чтобы избежать тяжеловесных цепочек форм Бэкуса – Наура мы будем раскрывать общий синтаксис выражения

постепенно. Итак, вначале формализуем общий синтаксис нетерминала выражение.

### 1. Общий синтаксис нетерминала <Выражение>

```
<Выражение> ::= <Простое выражение>
               | <Одноместная операция><Простое выражение>
               | <Выражение><Двухместная операция><Простое выражение>
               | ( <Выражение> )
```

```
<Простое выражение> ::= <Операнд>
                       | <Одноместная операция><Операнд>
                       | <Операнд><'>
                       | <Операнд><Двухместная операция><Операнд>
```

```
<Операнд> ::= <Константа>
              | <Матрица>
              | <Вызов функции>
```

```
<Вызов функции> ::= <Имя функции>
                    | <Имя функции> ( )
                    | <Имя функции> ( <Список параметров> )
```

```
<Имя функции> ::= <Идентификатор>
```

```
<Список параметров> ::= <Параметр>
                       | <Параметр> , <Список параметров>
```

Приведенная формализация не раскрывает символического определения нетерминалов <Одноместная операция> и <Двухместная операция>. Дело в том, что операции в MatLab обладают весьма высоким полиморфизмом, то есть, смысл их выполнения и даже состав существенно зависит от типа операндов. Ранее мы уже подчеркивали, что все переменные MatLab являются матрицами.

В рамках данной лабораторной работы мы пока будем рассматривать только арифметические выражения для простых арифметических операций, то есть, операций операндами которых, являются арифметические константы, арифметические матрицы размером не более чем 1x1 или соответствующие вызовы функций. Другими словами, наши операнды будут неотличимы от операндов с которыми работает обычный калькулятор.

### 2. Синтаксис простого арифметического выражения.

Синтаксис арифметического выражения полностью будет совпадать с общим синтаксисом выражения, если заменить нетерминал <Операнд> на нетерминал <Простой арифметический операнд>, а также добавить нетерминалы <Простая одноместная арифметическая операция> и <Простая двухместная арифметическая операция>. При этом, приставка «простой» или «простая» будет означать операнды размером не более чем размерностью 1x1 или операции над

соответствующими операндами. Соответственно нетерминал <Матрица> заменим на нетерминал <Имя переменной (размерности 1x1)>:

<Простой арифметический операнд> ::= <Арифметическая константа>  
| <Имя переменной (размерности 1x1)>  
| <Вызов функции (с результатом размерности 1x1)>

<Простая одноместная арифметическая операция> ::= -

<Простая двухместная арифметическая операция> ::= -  
| + | \* | / | ^

При относительной тяжеловесности синтаксических определений, конечный результат их применения хорошо знаком нам еще со школьной скамьи. Приведем несколько примеров, которые можно проверить прямо в строке приглашения Command Window:

### 3. Примеры простейших операций без использования имен переменных.

```
>> 2+2
>> 2-2
>> 2*2
>> 4/2
>> 2^2      % возведение во вторую степень
>> sqrt(4)  % корень квадратный из 4
```

### 4. Примеры простейших операций с использованием имен переменных.

```
>> x = 3.2
>> y = -2.5
>> x+y
>> x-y
>> x*y;
>> x/y;
>> x^2;
>> sqrt(x/y);
```

### 5. Популярные алгебраические функции (краткий список)

```
sqrt(x);
exp(x);
log(x);
log10(x);
log2(x);
```



## 6. Константа pi и популярные тригонометрические функции (краткий список)

Аргументы прямых тригонометрических функций и результаты обратных представляются в радианах.

```
pi = 3,1415926535 8979323846 2643383279 5028841971 6939937510.....
```

```
sin(x); asin(x);  
cos(x); acos(x);  
tan(x); atan(x);  
cot(x); acot(x);
```

Подробнее о функциях см. HELP (MATLAB Functions: Functions - By Category: Mathematics).

### ПРИМЕРЫ.

#### Примеры арифметических выражений.

##### 1. Нахождение корней квадратного уравнения:

Алгебраическая запись:

$$a * x^2 + b * x + c = 0$$
$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4 * a * c}}{2 * a}$$

Запись в MatLab:

```
a*x^2+b*x+c  
x=(-b+sqrt(b^2-4*a*c))/(2*a)  
  
% Корень действительное число  
>> a=4  
>> b=16  
>> c=2  
>> (-b+sqrt(b^2-4*a*c))/(2*a)  
% Результат:  
ans =  
    -0.1292  
  
% Корень мнимое число  
>> a=4  
>> b=2  
>> c=2  
>> (-b+sqrt(b^2-4*a*c))/(2*a)  
% Результат:  
ans =  
    -0.2500 + 0.6614i
```

## **2. Проверка корней квадратного уравнения:**

```
% Проверка с действительным корнем
>> a=4
>> b=16
>> c=2
>> x=-0.1292
>> a*x^2+b*x+c
% Результат:
ans =
    -4.2944e-004
```

```
% Проверка с мнимым корнем
>> a=4
>> b=2
>> c=2
>> x=-0.2500 + 0.6614i
>> a*x^2+b*x+c
% Результат:
ans =
    2.0016e-004
```

## **3. Преобразование в радианы (R) угла заданного в градусах (D) и обратно.**

```
>> R = pi*D/180
>> D = 180*R/pi
```

## **4. Нахождение проекций вектора заданного длиной L и углом поворота D (в градусах).**

```
>> L=4
>> D=30
>> R = pi*D/180
>> x = L*cos(R)
>> y = L*sin(R)
% Результат:
R =
    0.5236
x =
    3.4641
y =
    2.0000
```

## **5. Нахождение длины L и угла поворота D (в градусах) для вектора заданного проекциями (x, y).**

```
>> x=3.4641
>> y=2.0000
>> L=sqrt(x^2+y^2)
>> R=atan(y/x)
>> D=180*R/pi
```

% Результат:

L =

4.0000

R =

0.5236

D =

30.0000

## ПРАКТИЧЕСКАЯ РАБОТА.

1. Составьте собственные имена переменных и выполните примеры приведенные в разделе «Арифметические операции»

2. Вычислите корни квадратных уравнений, а также выполните несколько операций с проекциями произвольных векторов.

3. Вычислите с помощью MatLab объемы простейших геометрических тел

Фигура	Формула	Обозначения
Куб	$c^3$	$c$ — ребро куба
Призма	$Sh$	$S$ — площадь основания, $h$ — высота призмы
Цилиндр	$\pi r^2 h$	$r$ — радиус, $h$ — высота цилиндра
Шар	$\frac{4}{3}\pi r^3$	$r$ — радиус
Эллипсоид	$\frac{4}{3}\pi abc$	$a, b, c$ — главные оси
Пирамида	$\frac{1}{3}Ah$	$A$ — площадь основания, $h$ — высота пирамиды
Конус	$\frac{1}{3}\pi r^2 h$	$r$ — радиус основания, $h$ — высота конуса

---

Финальная редакция материала 23.08.2018г.

Воронов С.И.