ТЕМА 2

ДАННЫЕ И ВЫЧИСЛЕНИЯ

Изучаемый минимум:

- Создание векторов и матриц
  x0:dx:x1 [A B; C D]
  zeros, **ones**, **eye**, **rand**, **randn**
- Извлечение элементов из матрицы
  M(row,col) M(row,:) M(:,col)
  M(end,end-1) M(:)
- Операции над матрицами:
  ПОЭЛЕМЕНТНО: **A+B**, **A-B**, **A.*B**, **A.^B**, **A./B**
  МАТРИЧНЫЕ: **A+B**, **A-B**, **A*B**, **A^n**, **A/B**, **A\B**
- Логические выражения:
  0 – Ложь, **1** (не **0**) – Истина
  Операции: **A&B**, **A|B**, **~A**, **A==B**, **A~=B**

2.1. Арифметические вычисления

- Все данные в MATLAB представляются в виде массивов.
- При записи оператора с явным присваиванием вида **a = x + y** результат присваивается переменной, стоящей слева от знака равенства.
- При записи оператора с неявным присваиванием вида **x + y** результат присваивается автоматически создаваемой системой переменной с именем **ans** (ANSwer).

2.1.1. Рациональные числа

- Диапазон: **realmin**, **realmax**.

2.1.2. Комплексные числа

- Мнимая часть числа: символ **i** либо **j**.
- Комплексные числа при умножении, делении и возведении в степень заключаются в круглые скобки.

- Для вычисления комплексно-сопряженного числа применяется апостроф, который следует набирать сразу за числом, без пробела.

Пример 2.2

```
>> sqrt(-1.0)
ans =
     0 + 1.0000i

>> (2.+3i)*(3+5j)
ans =
     -9.0000 +19.0000i

>> 0+5.6j
ans =
     0 + 5.6000i

>> 5+0i
ans =
     5

>> 3+2i'
ans =
     3.0000 - 2.0000i

>> (sin(2i)+ 3+2i')'
ans =
   3.0000 - 1.6269i
```

2.1.3. Операторы

- Операторы: **+**, **.+**, **–**, **.–**, *****, **.***, **/**, **./**, ****, **.**, **^**, **.^**, **'**, **.'**
- Порядок операций:
  - возведение в степень **^**;
  - умножение и деление *****, **/**, ****;
  - сложение и вычитание **+**, **–**;
  - слева направо.
- Изменение порядка: **()**.

Пример 2.5

```
>> 5+4*3^2
ans =
     41

>> 3^3^3
ans =
     19683

>> 3^(3^3)
ans =
     7.6256e+012
```

2.1.4. Встроенные математические функции

- Список встроенных элементарных функций с их кратким описанием: команда **>> help elfun**.

- Список встроенных специальных математических функций с их кратким описанием: команда **`>> help specfun`**.
- Имена функций набираются строчными буквами.
- Аргументы функций заключаются в круглые скобки и разделяются запятыми.
- Выходные параметры функций заключаются в квадратные скобки и разделяются запятыми или пробелами.

2.2. Массивы

- Справка об элементарных матрицах и матричных манипуляциях: команда **`>> help elmat`**.
- Способы создания массивов:
  - ввод поэлементно всех значений (операция индексации **`()`**);
  - ввод полного списка элементов (операция конкатенации **`[]`**) для векторов и матриц;
  - операция формирования диапазона значений **`:`**;
  - присваивание значения самому последнему элементу массива;
  - встроенные функции генерации массивов.
- Ввод элементов одной строки – через запятую **`,`** или пробел.
- Ввод строк – через точку с запятой **`;`**.
- Скаляр – матрица (1 × 1).
- Вектор-столбец (матрица (n × 1)), вектор-строка (матрица (1 × n)).
- Доступ к элементу – при помощи операции индексации **`()`**:
  - один индекс (порядковый номер элемента);
  - группа индексов, разделяемых запятыми.
- Первое значение индекса – **`1`**, последнее значение – **`end`**.
- В качестве индекса может задаваться вектор.
- Чтение несуществующего элемента: сообщение об ошибке.
- Запись несуществующего элемента: недостающие элементы заполняются нулями.
- Элементы любого массива упорядочены по столбцам.

2.2.1. Векторы

`Пример 2.9`

Создание вектор-строки операцией конкатенации. Разделители элементов строки – пробелы или запятые.

```
>> V1 = [ 1 0 3 -5]
V1 =
     1     0     3    -5
```

Создание вектор-строки поэлементно.

```
>> b(1)=1; b(2)=3; b(3)=-5; b(4)=2
b =
     1     3    -5     2
```

Создание вектор-строки при помощи операции формирования диапазона значений.

```
>> V = 1 : 3 : 12
V =
     1     4     7    10
```

Создание вектор-строки присваиванием значения последнему элементу.

```
>> V3(3) = 9
V3 =
     0     0     9
```

Создание вектор-строки операцией конкатенации двух строк.

```
>> V=[V1 V3]
V =
     1     0     3    -5     0     0     9
```

Изменение существующего элемента.

```
>> V(3)=17
V =
     1     0    17    -5     0     0     9
```

Изменение группы элементов.

```
>> V(5:7)=4
V =
     1     0    17    -5     4     4     4
```

Чтение несуществующего элемента.

```
>> V(9)
??? Index exceeds matrix dimensions.
```

Запись несуществующего элемента.

```
>> V(9)=2
V =
   1    0    17    -5    4    4    4    0    2
```

Операция конкатенации.

```
>> V=[V 8 9]
V =
   1    0    17    -5    4    4    4    0    2    8    9
```

Удаление элемента или группы элементов операцией **[]**:

```
>> V(3)=[]
V =
    1     0     -5     4     4     4     0     2     8     9
>> V([1 4 5])=[]
V =
     0     -5      4      0      2      8      9
>> V([1:2:end])=[]
V =
    -5      0      8
```

Пример 2.14

Задание вектор-столбца операцией конкатенации. Элементы разделяются точкой с запятой.

```
>> s1 = [ 1; 3; -12]
s1 =
    1
    3
  -12
```

2.2.2. Операции с векторами

Размеры векторов, к которым применяются поэлементные операции, должны совпадать. Исключение – операции над вектором и числом.

```
>> v1 = [4 2 3];
>> v2 = [3 -2 0];
```

Пример 2.17

Нахождение суммы (**+**) и разности (**–**) двух векторов.

```
>> s = v1+v2
 s =
     7   0   3
```

Пример 2.18

Оператор **.*** : поэлементное умножение векторов:

```
>> u = v1.*v2
 u =
     12   -4   0
```

Пример 2.19

Оператор **.^**: поэлементное возведение в степень:

```
>> u = v1.^3
 u =
     64   8   27
>> u = v1.^v2
 u =
     64   0.2500   1
```

Пример 2.20

Оператор **./**: поэлементное деление элементов первого вектора на соответствующие элементы второго вектора:

```
>> u = v1./v2
 u =
     1.3333   -1   Inf
```

Оператор **.**: обратное поэлементное деление элементов второго вектора на соответствующие элементы первого.

```
>> u = v1.\v2
 u =
     0.75   -1   0
```

Пример 2.21

Сложение вектора и числа прибавляет число к каждому элементу вектора:

```
>> u = v1+10
```

```
 u =
     14   12   13
```

То же самое справедливо и для вычитания.

Пример 2.22

Поэлементное умножение вектора на число как справа, так и слева приведет к одинаковому результату: **v1*10** ≡ **10*v1**. Делить при помощи знака **/** можно вектор на число: **v1/10**. Деление числа на вектор **10/v1** приводит к сообщению об ошибке. Для деления числа на каждый элемент вектора используется операция **./**

```
>> u = 10./v1
 u =
     2.5   5   3.3333
```

или:

```
>> u =v1.\ 10
 u =
     2.5   5   3.3333
```

Пример 2.23

V.', **V'**: транспонирование, комплексно-сопряженное транспонирование.

```
>> V = [3+i; -2-2i; 10];
>> V1=V'
 V =
      3-1i   -2+2i   10
>> V2=V.'
 V2 =
      3+1i   -2-2i   10
```

2.2.3. Создание матриц

Пример 2.26

Создание матрицы операцией конкатенации.

```
>> A=[1 2 3; 4,5,6]
A =
     1     2     3
```

```
     4     5     6
```

Создание матрицы как вектор-строки, каждый элемент которой является вектор-столбцом.

```
>> A = [[1; 4] [2; 5] [3; 6]]
A =
     1     2     3
     4     5     6
```

Создание матрицы поэлементно.

```
>> A(1,1) = 1;  A(1,2) = 2;  A(1,3) = 3;
>> A(2,1) = 4;  A(2,2) = 4;  A(2,3) = 6
A =
     1     2     3
     4     5     6
```

Создание матрицы присваиванием значения последнему элементу.

```
>> B(2,3)=1
B =
     0     0     0
     0     0     1
```

Создание матрицы операцией конкатенации матриц.

```
>> A1 = [1 2 3; 4 5 6]; A2 = [0 0 0; 1 1 1];
>> A = [A1 A2; A2, A1]
A =
     1     2     3     0     0     0
     4     5     6     1     1     1
     0     0     0     1     2     3
     1     1     1     4     5     6
```

2.2.4. Специальные матрицы

- **`[]`**: пустая матрица (0 × 0);
- **`diag(X,k)`**: создание диагональной матрицы или выделение диагонали;
- **`eye(m,n)`**: (m × n)-матрица с единицами на главной диагонали;
- **`magic(n)`**: (n × n)-матрица магического квадрата с числами от 1 до n^2;
- **`ones(m,n)`**: (m × n)-матрица из единиц;

- **rand(m,n)**: (m × n)-матрица случайных чисел со значениями из интервала [0, 1];
- **randn(m,n)**: (m × n)-матрица равномерно распределенных случайных чисел;
- **tril(X,k)**: выделение нижней треугольной части матрицы **X**;
- **triu(X,k)**: выделение верхней треугольной части матрицы **X**;
- **zeros(m,n)**: (m × n)-матрица из нулей.

Пример 2.27

```
>> A=[]
A =
     []

>> eye(2,3)
ans =
     1     0     0
     0     1     0

>> magic(3)
ans =
     8     1     6
     3     5     7
     4     9     2

>> triu(magic(3))
ans =
     8     1     6
     0     5     7
     0     0     2

>> diag([1 2 3])
ans =
     1     0     0
     0     2     0
     0     0     3

>> rand(3)
ans =
    0.9501    0.4860    0.4565
    0.2311    0.8913    0.0185
    0.6068    0.7621    0.8214
```

2.2.5. Доступ к элементам матрицы

Пример 2.28

```
>> A = [11:19; 21:29]
A =
    11    12    13    14    15    16    17    18    19
    21    22    23    24    25    26    27    28    29
>> A(2,4)
ans =
    24
>> A(8)
```

```
ans =
    24
>> A(1:2:end)
ans =
    11   12   13   14   15   16   17   18   19
>> A(:,2)
ans =
    12
    22
>> A(:,[2 6 4])
ans =
    12   16   14
    22   26   24
>> A(:,1:2:end)
ans =
    11   13   15   17   19
    21   23   25   27   29
>> A(2,:)
ans =
    21   22   23   24   25   26   27   28   29
>> A(:,5:end-1) = []
A =
    11   12   13   14   19
    21   22   23   24   29
>> A(1:3,2:3) = ones(3,2)
A =
    11    1    1   14   19
    21    1    1   24   29
     0    1    1    0    0
>> A = A([1 3],[5 1])
A =
    19   11
     0    0
>> A(3,3)
??? Index exceeds matrix dimensions.
```

```
>> A(3,3) = 100
A =
    19    11     0
     0     0     0
     0     0   100
```

2.2.6. Арифметические операции над матрицами

- **.+**, **.–**, **.*****, **./**, **.**, **.^**: поэлементные операторы сложения, вычитания, умножения, деления и возведения в степень. Если один из операндов – скаляр, то он воздействует на каждый элемент другого операнда;
- **A+B**: матричное сложение, аналог **A.+B**;
- **A–B**: матричное вычитание, аналог **A.–B**;
- **A*B**: матричное умножение (по правилу «строка на столбец»);
- **A^n**: n-кратное матричное умножение;
- **X=B/A**: левостороннее деление, решение системы линейных уравнений **XA=B**;
- **X=A\B**: правостороннее деление, решение системы линейных уравнений **AX=B**;
- **A.'**: транспонирование матрицы;
- **A'**: комплексно-сопряженное транспонирование матрицы.

Пример 2.31

```
>> A = [1 2; 3 4];  B=eye(2);
```

```
>> A+1
ans =
     2     3
     4     5
```

```
>> A.*3
ans =
     3     6
     9    12
```

```
>> A.^3
ans =
     1     8
    27    64
```

```
>> A*A
ans =
     7    10
    15    22
```

```
>>X = A\B
X =
   -2.0    1.0
    1.5   -0.5
```

```
>> A.'
ans =
     1     3
     2     4
```

```
>> A+B
ans =
     2     2
     3     5

>> A.*A
ans =
     1     4
     9    16

>> A./B
Warning: Divide
by zero.
ans =
     1   Inf
   Inf     4
```

2.2.7. Операции отношения и логические операции

- **>**, **>=**, **<**, **<=**, **==**, **~=**: операторы отношения, аналоги функций **gt**, **ge**, **lt**, **le**, **eq**, **ne**. Поэлементные сравнения матриц одинакового размера. Скаляр сравнивается с каждым элементом массива. Результат – логическая матрица.
- Операции **==**, **~=** проводят сравнения вещественных и мнимых частей комплексных чисел, а операции **>**, **<**, **>=**, **<=** – только вещественных частей.
- **&**, **|**, **~**: логические операторы, аналоги функций **and**, **or**, **not**, **xor**.
- **find**: выделение части массива, элементы которого удовлетворяют определенному условию.

Пример 2.32

```
>> A = [0 1; 2 3];  B=eye(2);

>> A<=B
ans =
     1     0
     0     0

>> A&B
ans =
     0     0
     0     1

>> A(find((A>2)|(A<1)))=9
A =
     9     2
     3     9
```

2.3. Лабораторный практикум № 2

2.3.1. Задания

Задание 2.6

Вычислите значения:

a) $\dfrac{1}{\frac{\sqrt{2}}{2}+i\frac{\sqrt{2}}{2}}$; b) $\dfrac{2+3i}{4-i}+\dfrac{2-3i}{1+4i}$; c) $\sin\left(1+\sqrt[3]{1-i}\right)$.

Задание 2.8

Вычислите значения **e^(1+2i)**, **(1+2i)^e**, **(1+2i)^(1+2i)**.

Задание 2.10

Создайте матрицу **b** как вектор-строку 1×4 **b=(1 2 3 4)**. Транспонируйте вектор-строку **b** в вектор-столбец **c**.

Задание 2.11

Создайте вектор-строку 1 × 4 **x=(1+2i 2-j 3 4)**. Введите команды и определите, в чем состоит их отличие?

```
>> y=x'
>> z=x.'
```

Преобразуйте вектор-строку **x** в вектор-строку **X=(1-2i 2+i 3 4)**.

Задание 2.13

Создайте матрицу **A** размером 4 × 3. $A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\\ 10 & 11 & 12\end{pmatrix}$.

Выполните следующие действия (каждое действие выполняется над исходной матрицей **A**):

- выделите третью строку матрицы;
- удалите второй столбец;
- поменяйте местами первый и второй столбцы;
- измените порядок следования строк матрицы на обратный;
- доступ к элементу матрицы выполняется при помощи индексов

  >> a=A(2,3) или порядкового номера **>> a=A(N)**

  Значение **N** равно…?

Задание 2.14

Создайте матрицы размером 4 × 4 при помощи встроенных функций **zeros**, **ones**, **eye**, **rand** и **randn**.

Задание 2.19

Задайте команды

```
>> A = [1, 2, 3; 4 5 6; 7, 8, 9];
```

Последовательно введите команды, выполняющие доступ к элементам матрицы, и изучите результаты их выполнения.

```
>> b=A(3,:)
>> b=A(:,2)
>> b=A(1:3,2)
>> B=A(1:3,2:3)
>> B=A([1 3],:)
>> B=A([3 2 1],:)
>> B=A(:,[3:-1:1]);
>> A(:)
>> p=[3 1 2];
>> n=[2 3 1];
>> A(p,:)
>> A(n,:)
>> A(:,p)
>> A(:,n)
>> A(p,p)
>> A(n,n)
```

Задание 2.20

Последовательно введите команды, выполняющие арифметические действия над матрицами, и изучите результаты их выполнения.

```
>> A = [1; 2; 3; 4; 5; 6]
>> B = A'
>> C = A.'
>> D= -A
>> B = A + 2*D
>> C = A' * A
>> a = [1:4]
```

```
>> E = a' * a
>> f = a * a'
>> d = A/2
>> G = A^2
>> B = A.^2
>> B = A.^A
>> C = A.*A
>> b=diag(A)
```

Задание 2.21

Задайте магическую матрицу **A** 5 × 5. Какие значения принимают следующие выражения?

```
a) A(2,:)      b) A(:,4)        c) A(end,:)

d) A([1,5])    e) A(1,5:-2:1)   f) A(5:-1:1,5:-1:1)

g) A(1:2:end)  h) A(:)          i) A(:,end)
```

Выделите из матрицы 5 × 5 верхнюю и нижнюю треугольные матрицы при помощи встроенных функций **tril** и **triu**.

Задание 2.25

Задайте две матрицы **A** и **B** размером 5 × 5. Последовательно введите команды, выполняющие логические действия и операции отношения над матрицами, и изучите результаты их выполнения.

```
>> A>B         A==B          A~=B         A<=B
>> spy(A>0)    spy(A<0.1)    spy(A>=B)
```

Задание 2.26

Для 4-х матриц и векторов **A** 2 × 3, **B** 3 × 3, **v** 3 × 1, **w** 2 × 1 выполните последовательно операции с целью контроля их легитимности:

```
>> A+B   A+A   B+B    v+w    v-w
>> A*B   B*A   A'*B   B*A'   A*A   A'*A   A*A'   B*B
>> A*v   v*A   v*A'   A'*v   w*A'   v*v'   v'*v   B*v   B*w
```

Задание 2.33

Создайте матрицы **A** размером 5 × 6 и **B** размером 3 × 2.

a) Выделите из матрицы **A** подматрицу **S** с 2-й по 4-ю строку и с 3-го по 6-й столбец.

b) Замените элементы в матрице **A** элементами матрицы **B**, начиная со 2-й строки и 4-го столбца.

c) Создайте матрицу **D** вида $D = \begin{pmatrix} A & 0 \\ 0 & A^T \end{pmatrix}$.

Задание 2.34

Создайте целочисленную случайную матрицу **A** размером 5 × 5. Разделите ее на четыре матрицы: **B** размером 3 × 3, **S** размером 3 × 2, **D** размером 2 × 3, **E** размером 2 × 2.

Задание 2.36

Игра «Жизнь» (Conway's Game of Life). Задан торический мир X размером M × M клеток (M = 101), первоначально заселенный живыми организмами с вероятностью 10 %. В каждой клетке может находиться ровно один живой организм. В каждом следующем поколении: организм зарождается в пустой клетке, вокруг которой собрались ровно 3 организма, или организм погибает:

- от ТОСКИ, если у него меньше двух соседей;
- ГОЛОДА, если у него больше трех соседей.

Смена поколений равна 100. Каждое поколение отображать графически. Прервать надоевшую эпоху можно, нажав клавиши **Ctrl-C**.

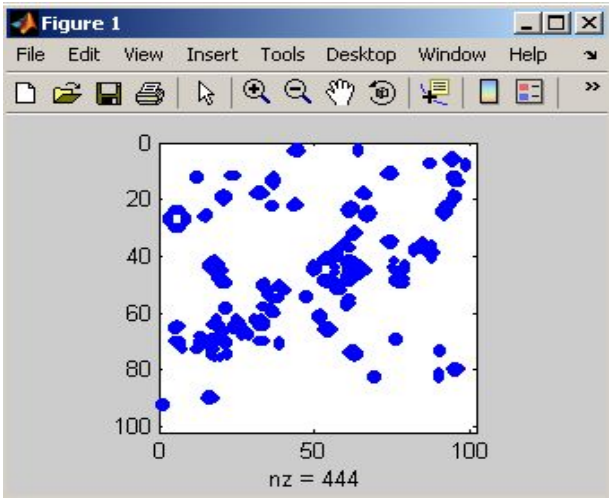

Рис. 2.1. Игра «Жизнь»