2. Массивы. Матричные вычисления. Линейная алгебра

2.1. Выражения. Переменные. Массивы. Структуры

a) Логические переменные, константы и выражения

Логические константы: 1 – истина, 0 – ложь.

Логические операции: >, <, <=, >=, ==, ~=, &, |, not

Задание: рассчитать таблицу истинности для 2И-НЕ, и для 2ИЛИ-НЕ

```
x = 1; y = 1;
R1 = not(x & y)
```

b) Массивы

Правила:

1) В MATLAB нижняя граница индексов в массиве **равна 1**.
2) Тип переменной не объявляется, и любая переменная по умолчанию считается матрицей.

Векторы

Вектором называется массив размерности 1 x N.

```
v1 = [1 3 5 7] % вектор-строка
v2 = [2; 4; 6; 8 ] % вектор-столбец
v3 = [0 : 10]
v4 = v3' % транспонирование
x = [0 : 0.1 : 8]; % начало, шаг, конец
x(3)
x(5:7)
x([1 3 5])
r = x; length(r)
r = x(end-2 : end) % размер вектора уменьшился
y = linspace(11, 17, 10) % начало, конец, число точек
```

Полезные функции:

```
clear
R = rand(10,1); % вторая размерность обязательна, иначе получится 10x10
S = sort(R);
[S k] = sort(R); % сохраняет вектор индексов исходного массива
max(R)
min(R)
mean(R)
for i = 1:10; V(i) = i^2; end; % иногда можно использовать и «классику»
```

Задания:

1) Сформировать вектор X от 0 до 20 с шагом 2. Сделать так, чтобы в векторе Y хранились значения вектора X в обратном порядке. Желательно не использовать цикл for.
2) Самостоятельно разобраться с функциями округления: fix, floor, ceil, round, nearest, convergent

```
D = [8.2 8.5 8.7 -8.2 -8.5 -8.7]
```

Тип функции	Функция	Назначение
Округление	`fix(X)`	Округление в направлении нуля — усечение дробной части
	`floor(X)`	Округление в направлении $-\infty$ — округление до ближайшего целого в сторону уменьшения
	`ceil(X)`	Округление в направлении $+\infty$ — округление до ближайшего целого в сторону увеличения
	`round(X)`	Округление до ближайшего целого — при дробной части, равной `0.5`, — в сторону увеличения модуля числа
	`nearest(X)`	Округление до ближайшего целого — при дробной части, равной `0.5`, — в сторону увеличения
	`convergent(X)`	Округление до ближайшего целого — при дробной части, равной `0.5`, — в сторону ближайшего четного числа

Двумерные массивы

Примеры:

```
A = [1 2 3; 4 5 6; 7 8 9]
B = A'
A(2,3)
A(:,3)
A(1,:)
A(:)
B = [ A(1,:); sin(pi/4) round(1) exp(2i)] % как вариант
W = [A B] % получилась матрица 3x6
```

Полезные функции:

```
sum(A) % суммирование по столбцам
sum(A(:)) % суммирование по всем элементам
Z = zeros(3,5) % нулевая
L = ones(2,3) % единичная
P = eye(5) % нулевая с единицами на главной диагонали
R = rand(4) % матрица 4x4
size(Z) % вектор со значениями каждой размерности
length(Z) % максимальная из размерностей
numel(Z) % число элементов в матрице
```

Задания:

```
% 1. сформировать матрицу 3x3, состоящую из чисел pi
%
% 2. даны матрицы
A = [1 2 3; 4 5 6; 7 8 9]; B=[8 2 6; 4 1 3; 7 0 9];
% найти, сколько элементов в этих матрицах совпадают друг с другом
```

Bitmap в виде матриц

В примере используются некоторые функции пакета IPT

```
M = imread('images/rose.tif'); % либо M = imread('images/cktboard.tif');
imshow(M);
imfinfo images/rose.tif
LM = M(300:700,300:700); % выделяем подматрицу
imshow(LM);
imwrite(LM,'rose.jpg'); % записываем матрицу на диск в формате jpg
```

Многомерные массивы

Интерактивно не просматриваются, визуально плохо представляются.

```
% формируем третье измерение
W = A; % предполагаем, что даны матрицы A и B размерности 3x3
W(:, :, 2) = B;
% второй способ
C = rand(3, 5, 3);
size(C)
C(:, :, 1)
```

c) Символьные переменные, константы и выражения

Используется Unicode-кодирование, строка представляет собой вектор символов.

Примеры:

```
S1 = 'строка';
length(S1)
S2 = [S1 ' символов']
findstr(S1,'сим') % поиск подстроки в строке
```

d) Примеры преобразования типов

```
X = 254; dec2bin(X), dec2hex(X)
str2num('0.25') % надёжнее использовать str2double
M = uint16(S1) % аналогично сработают single(S1) либо double(S1)
char(M) % восстановили строку
char(1610) % символ по его коду
```

e) Структуры

В одной переменной типа «структура» допускается группирование различных типов данных. Удобно для хранения различных сложно структурированных информационных ресурсов. Пример (из каталога *lum-exp*):

```
load('matlab2.mat')
comp.descr
comp.meas1.descr
comp.meas1.data
plot(comp.meas1.data(:, 1), comp.meas1.data(:, 2))
hold on % о формате графического вывода будет сказано позже
plot(comp.meas2.data(:, 1), comp.meas2.data(:, 2),'r')
```

f) Смешанные массивы

Смешанным массивом (cell array) называют многомерный массив, элементами которого являются копии других массивов, которые могут принадлежать различным типам данных.

```
c = {'test', [3e-5 1e-12 5e7 120], 3+2i}
format SHORT E
c{1}
c{2}
c{3}
cellplot(c,'legend') % графическое изображение массива
```

g) Символические вычисления

```
solve('sin(1/x)=pi/4')
solve('x^4=5')
```

2.2. Операции линейной алгебры

1. Суммой (разностью) матриц **A** и **B** размера $m \times n$ называется матрица **C** того же размера с элементами, равными сумме (разности) соответствующих элементов матриц **A** и **B**. Примеры:

```
A = [1 2 3; 2 5 4; 7 8 9]; B = [1 2 3; 3 2 1; 0 2 3];
V = [1 2 3]; W = [4 5 6];
A + B, A – B
V + W, V – W
```

2. Произведением матрицы **A** размера $m \times n$ на матрицу **B** размера $n \times p$ называется матрица **C** размера $m \times p$, элемент i-й строки и k-го столбца которой равен сумме произведений соответственных элементов i-й строки матрицы **A** и k-го столбца матрицы **B**: $C_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk}$; $i = 1, 2, \ldots, m$; $k = 1, 2, \ldots, p$.

Примеры:

```
V*W % сообщение об ошибке
V*W' % одно число
W'*V % матрица 3*3
W.*V % поэлементное умножение
A * B, A * B', A' * B, A .* B % четыре разных результата
B * A % умножение матриц не коммутативно
A^3 % операция эквивалентна A*A*A
%
% получение комплексно сопряжённой матрицы
Q=[1+2i 7-3i;-1-5i 1+2i;8+i 1-8i]
Qs = (Q').' % эрмитово сопряжённая и затем транспонированная
```

3. Матрица **D** называется обратной к матрице **A**, если произведение этих матриц дает единичную матрицу **I**. Матрицы должны быть квадратными. Примеры:

```
D = A^-1 % либо D = inv(A)
A*D % либо D*A
```

4. Определитель (детерминант) матрицы – многочлен, комбинирующий элементы квадратной матрицы таким образом, что его значение сохраняется при транспонировании и линейных комбинациях строк или столбцов.

```
det(A), det(B)
```

5. Деление матриц, примеры:

```
A \ B % левое матричное деление эквивалентно inv(A)*B
A / B % правое матричное деление эквивалентно A*inv(B)
W./V  % поэлементное деление
```

2.3. Матричные вычисления в задачах линейной алгебры и математической статистики

1. Норма матрицы

Для расчёта абсолютной и относительной погрешностей вычислений часто используют понятия нормы вектора и матрицы. Существуют различные определения норм, к числу часто употребляемых относятся следующие три нормы:

Векторная норма $\left\|\vec{X}\right\|_1$ равна сумме модулей элементов вектора:

$$\left\|\vec{X}\right\|_1 = \sum_{i=1}^{m} |x_i|.$$

Соответствующая норма матрицы $\left\|\hat{A}\right\|_1$ определяется как максимальная сумма модулей элементов в столбце:

$$\left\|\hat{A}\right\|_1 = \max_j \sum_{i=1}^{m} \left|A_{ij}\right|.$$

Векторная норма $\left\|\vec{X}\right\|_\infty$ равна максимальному элементу вектора:

$$\left\|\vec{X}\right\|_\infty = \max_i |x_i|.$$

Соответствующая норма матрицы определяется как максимальная сумма модулей элементов в строке:

$$\left\|\hat{A}\right\|_\infty = \max_i \sum_{j=1}^{n} \left|A_{ij}\right|.$$

Евклидова норма вектора $\left\|\vec{X}\right\|_2$ вводится выражением:

$$\left\|\vec{X}\right\|_2 = \sqrt{\sum_{i=1}^{m} |x_i|^2}.$$

Для матрицы евклидова норма выглядит следующим образом:

$$\left\|\hat{A}\right\|_e = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} \left|a_{ij}\right|^2}.$$

При анализе результатов эксперимента либо в численных расчётах абсолютная и относительная погрешности для вектора данных представляются как:

$$\Delta\vec{X} = \left\|\vec{X} - \vec{X}^*\right\|, \qquad \partial\vec{X} = \frac{\left\|\vec{X} - \vec{X}^*\right\|}{\left\|\vec{X}\right\|},$$

где $\vec{X}$ – точное, а $\vec{X}^*$– приближённое решение задачи. Абсолютная и относительная погрешности для матрицы вводятся аналогично погрешностям вектора.

Примеры:

```
X = [0.02 1e-12 3.33e-5 0 0.034]
N = [norm(X, 1) norm(X, inf) norm(X)] % евклидова – по умолчанию (2)
mean(X) % матожидание (среднее по элементам) - довольно мало
```

2. Элементарные статистические функции. Примеры:

```
R = rand(100,1);
% другие сл. распределения normrnd(0.5, 0.2);
max(R), min(R), mean(R)
std(R), var(R) % среднеквадратичное отклонение и дисперсия
%
x = [0 : 0.1 : 10];
y = normpdf(x, 0.5, 0.2) % статистическое распр. Гаусса
```

3. Решение систем линейных алгебраических уравнений

Система из m линейных уравнений с n неизвестными может быть описана при помощи матриц $\hat{A} \cdot \vec{x} = \vec{b}$, где $\vec{x}$ – вектор неизвестных, $\hat{A}$ – матрица коэффициентов при неизвестных или матрица системы, $\vec{b}$ – вектор свободных членов системы или вектор правых частей. Численным решением системы являются такие значения вектора $\vec{x}$, что выполняется уравнение $\hat{A} \cdot \vec{x} - \vec{b} = 0$ с заданной точностью. Пример: пусть дана СЛАУ:

$$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = 8 \\ x_1 - 3x_2 - 6x_4 = 9 \\ 2x_2 - x_3 + 2x_4 = -5 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = 0 \end{cases}$$

```
A = [2 1 -5 1; 1 -3 0 -6; 0 2 -1 2; 1 4 -7 6];
b = [8; 9; -5; 0];
% первый способ - стандартный
x = linsolve(A, b)
R = A*x-b % проверка невязок
% второй способ – по определению СЛАУ
x = inv(A)*b
x = A \ b % то же, но в других терминах
% третий способ – метод Гаусса
G = [A b] % расширенная матрица
G = rref(G) % верхнетреугольная матрица
x = G( : , end) % простая формальность: ответ - в последнем столбце
```

Задание: решить СЛАУ несколькими способами и определить точность решения:

$$\begin{cases} 5x_1 + 2x_2 - 7x_3 - 0,5x_4 + 9 = 0; \\ x_1 - 0,3x_2 + 9x_3 + 5 = 0; \\ 6x_1 + x_2 - 8x_3 - 19 = 0; \\ 3x_2 + 4x_3 - 2x_4 = 0 \end{cases}$$