Лекция 2. Матричная лаборатория

1. Матрицы и их представление в MATLAB.
2. Элементы матриц.
3. Операции с матрицами.
4. Поэлементные операции с матрицами.
5. Операции с матрицами в задачах линейной алгебры.
6. Матричное сложение, вычитание, умножение и возведение в степень.
7. Транспонирование и эрмитово сопряжение матриц.
8. Вычисление основных характеристик матрицы.
9. Обращение матрицы.
10. Матричное деление.
11. Разложение матриц.
12. Операции с матрицами в задачах математической статистики.

2.1. Матрицы и их представление в MATLAB

Алгоритмический язык MATLAB называют языком "сверхвысокого" уровня за счет *матричной обработки данных*.

Это значит, что *любая переменная по умолчанию считается* ***матрицей***.

В линейной алгебре матрица обычно обозначается заглавной буквой (часто полужирным шрифтом), а ее элементы — строчными буквами с индексами. Запишем матрицу $\mathbf{A}$ с учетом того, что в MATLAB *нижняя граница* индексов равна ***единице***:

$$\mathbf{A}=\begin{bmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1n} \\ \dots & & \dots & & \dots \\ a_{i1} & \dots & a_{ij} & \dots & a_{in} \\ \dots & & \dots & & \dots \\ a_{m1} & \dots & a_{mj} & \dots & a_{mn} \end{bmatrix}.$$

где a_{ij} — элемент матрицы ***i***-й строки и ***j***-го столбца.

Размер матрицы, определяемый количеством строк m и столбцов n, принято записывать в виде произведения $m\times n$.

Имя (идентификатор) матрицы в MATLAB составляется из последовательности латинских букв, цифр и символа подчеркивания и начинается с ***буквы***. *Прописные и строчные буквы различаются*.

Матрица вводится построчно в *квадратных скобках*, элементы строки отделяются пробелом (или запятой), а сами строки — точкой с запятой:

```
>> A = [1 2 3;5 6 7;8 9 7;10 11 12]
A =
     1     2     3
     5     6     7
     8     9     7
    10    11    12
```

Размер матрицы $m\times n$ определяется с помощью функции **`size`**:

```
>> size(A)
ans =
     4     3
```

Хранение матриц в оперативной памяти организовано ***по столбцам***.

Матрицу $\mathbf{A}$ размером $1\times n$ называют ***вектором-строкой***, и его элементы указываются *одним* индексом:

$$\mathbf{A}=\begin{bmatrix} a_1 & \dots & a_i & \dots & a_n \end{bmatrix},$$

Элементы *вектора-строки* вводится в квадратных скобках через пробел (или запятую):

```
>> A = [1 2 3 4 5 6 7]
A =
     1     2     3     4     5     6     7
>> size(A)

ans =

     1     7
```

Матрицу $\mathbf{A}$ размером $m \times 1$ называют ***вектором-строкой***, и его элементы также указываются *одним* индексом:

$$\mathbf{A} = \begin{bmatrix} a_1 \\ \dots \\ a_j \\ \dots \\ a_m \end{bmatrix}$$

Элементы *вектора-столбца* вводятся в квадратных скобках через точку с запятой:

```
>> A = [1;4;5;8]
A =
     1
     4
     5
     8
>> size(A)
ans =
     4     1
```

Длиной вектора называют количество его элементов.

Длина вектора определяется с помощью функции **`length`**:

```
>> A = [1;4;5;8;9;10;11;12];
>> length(A)
ans =
     8
```

Матрицу $\mathbf{A}$ размером 1×1 называют ***скаляром***, и его можно вводить без квадратных скобок:

```
>> A = 7
A =
     7
>> size(A)
ans =
     1     1
```

Матрицу $\mathbf{A}$ размером $n \times n$ называют ***квадратной*** *матрицей* ***порядка*** n:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1i} & \dots & a_{1n} \\ \dots & & \dots & & \dots \\ a_{i1} & \dots & a_{ii} & \dots & a_{in} \\ \dots & & \dots & & \dots \\ a_{n1} & \dots & a_{ni} & \dots & a_{nn} \end{bmatrix}.$$

Главной диагональю квадратной матрицы называют вектор, образованный из ее диагональных элементов a_{ii}, $i = 1, 2, \ldots, n$.

Главную диагональ можно вывести в виде вектора-столбца с помощью функции **diag**:

```
>> A = [1 2 3;5 6 7;8 9 7]
A =
     1     2     3
     5     6     7
     8     9     7
>> diag(A)
ans =
     1
     6
     7
```

Единичной матрицей называют порядка квадратную матрицу $\mathbf{I}$ порядка n, все элементы которой равны нулю, кроме элементов главной диагонали, равных единице:

$$\mathbf{I} = \begin{bmatrix} 1 & \ldots & 0 & \ldots & 0 \\ \ldots & & \ldots & & \ldots \\ 0 & \ldots & 1 & \ldots & 0 \\ \ldots & & \ldots & & \ldots \\ 0 & \ldots & 0 & \ldots & 1 \end{bmatrix}.$$

Иногда удобно зарезервировать в **Workspace** *имя* матрицы, размер которой и значения элементов заранее неизвестны. Такую матрицу называют *пустой* и вводят в квадратных скобках без содержимого:

```
>> A = [];
>> size(A)
ans =
     0     0
```

2.2. Элементы матриц

В этой лекции будут рассматриваться матрицы, элементами которых являются ***численные константы***. При формировании матрицы эти элементы могут задаваться:

- ☐ непосредственно в виде *численных констант*:

```
>> A = [1 5.7 3.8;17 8 13;0.1 0 19]
A =
    1.0000    5.7000    3.8000
   17.0000    8.0000   13.0000
    0.1000         0   19.0000
```

- ☐ в виде *имен переменных* (скаляров), значения которых известны:

```
>> a = 1; b = 5.7; c = 3.8; d = 17;
>> A = [a b;c d]
A =
    1.0000    5.7000
    3.8000   17.0000
```

- ☐ в виде *арифметических выражений* с известными значениями переменных (скаляров):

```
>> a = 1; b = 5.7; c = 3.8; d = 17;
>> A = [a+sin(b) c+d;a/d sqrt(d)]
A =
    0.4493   20.8000
    0.0588    4.1231
```

- ☐ в виде *имен матриц* с известными элементами:

```
>> a = [1 2;3 4],b =[4 5;6 7],c = [1 1;0 0],d = [10 -10;- 7 7]
a =
     1     2
     3     4
b =
     4     5
     6     7
c =
     1     1
     0     0
d =
    10   -10
    -7     7
>> A = [a b;c d]
A =
     1     2     4     5
     3     4     6     7
     1     1    10   -10
     0     0    -7     7
```

□ в виде ***регулярной сетки*** для *векторов*:
<*начальное значение*>:[<*шаг*>:]<*конечное значение*>

Шаг, равный единице, можно не указывать, условным признаком чего служат квадратные скобки.

Например, для вектора **x** при шаге, равном единице:

```
>> x = 7:10
x =
     7     8     9    10
```

и для того же вектора при шаге, равном 0.01;

```
>> x = 7:0.01:10;
>> length(x)
ans =
   301
```

□ в виде численных констант при *автоматической генерации типовых матриц*.
Типовые матрицы генерируются с помощью *стандартных функций* MATLAB, примеры которых даются в табл. 1.

Сгенерируем *единичную матрицу* **A** третьего порядка с помощью функции **eye**:

```
>> A = eye(3)
A =
     1     0     0
     0     1     0
     0     0     1
```

Таблица 1. *Функции генерирования типовых матриц*

Функция	**Типовая матрица**
zeros(M,N)	Нулевая матрица M×N
ones(M,N)	Матрица единиц M×N
eye(N)	Единичная матрица порядка N
rand(M,N)	Матрица M×N случайных чисел в диапазоне от 0 до 1, распределенных по *равномерному* закону
randn(M,N)	Матрица M×N случайных чисел, распределенных по *нормальному* закону с математическим ожиданием, равным 0, и дисперсией, равной 1

diag(V)	1. Диагональная матрица — квадратная матрица с нулевыми элементами, кроме элементов главной диагонали, заданными вектором V. 2. Вектор V из элементов главной диагонали квадратной матрицы

Обращение к ***элементу*** матрицы происходит по ее имени с указанием индексов в круглых скобках:

```
>> B = [3 5 7;3 7 9;2 0 1]
B =
     3     5     7
     3     7     9
     2     0     1
>> B(1,1)
ans =
     3
>> i = 2; j = 3; B(i,j)
ans =
     9
```

Обращение к ***строке*** матрицы (выделение строки) происходит по ее имени с указанием *номера строки M* — **A(M,:)**:

```
>> C = [1 5.7 3.8;17 8 13;0.1 0 19]
C =
    1.0000    5.7000    3.8000
   17.0000    8.0000   13.0000
    0.1000         0   19.0000
>> C(2,:)
ans =
    17     8    13
```

Обращение к ***столбцу*** матрицы (выделение столбца) происходит аналогично с указанием *номера столбца N* — **A(:,N)**:

```
>> C(:,3)
ans =
    3.8000
   13.0000
   19.0000
```

Другие разновидности обращений будут рассмотрены на лабораторных занятиях/

2.3. Операции с матрицами

Операции с матрицами принято разделять на *две группы*:

- ***поэлементные*** *операции*;
- ***матричные*** *операции*, в которых выделяют *две подгруппы*:
  - операции с матрицами в задачах ***линейной алгебры***;
  - операции с матрицами в задачах ***математической статистики***.

Рассмотрим данные операции подробнее.

2.4. Поэлементные операции с матрицами

К поэлементным операциям с матрицами относятся:

- арифметические операции: сложение, вычитание, умножение, деление, возведение в степень;

  В поэлементных арифметических операциях матрицы-операнды должны иметь *одинаковый размер*, т. к. в этих операциях синхронно участвуют соответственные элементы матриц.

- вычисление элементарных функций, аргументы которых — матрицы.

При выполнении поэлементных арифметических операций необходимо помнить о:

- наличии ***точки*** перед символами арифметических операций *умножения*, *деления* и *возведения в степень* (табл. 2, второй столбец);
- наличии двух операций деления: левого и правого.

Таблица 2. *Символы арифметических операций в MATLAB*

Операция	**Поэлементная**	**Матричная**
Сложение	**+**	**+**
Вычитание	**–**	**–**
Умножение	**.***	*****
Деление	Левое **.** Правое **./**	Левое **** Правое **/**
Возведение в степень	**.^**	**^**

Рассмотрим выполнение ***поэлементных арифметических операций*** На простых примерах. Сформируем простейшие квадратные матрицы **A** и **B** второго порядка:

```
>> A = [1 10;2 5]
A =
     1    10
     2     5
>> B = [0 1;-1 7]
B =
     0     1
    -1     7
```

Выполним поэлементное ***сложение*** матриц **A** и **B** (пояснить результат):

```
>> C = A+B
C =
     1    11
     1    12
```

Выполним поэлементное ***умножение*** матриц **A** и **B** (пояснить результат):

```
>> D = A.*B
D =
     0    10
    -2    35
```

Уберем *точку* перед операцией умножения:

```
>> D = A*B
D =
   -10    71
    -5    37
```

Получен совсем другой результат — результат *матричного* умножения, о котором пойдет речь далее.

Выполним поэлементное ***правое деление*** матриц **A** и **B** (пояснить, что здесь ***правильный*** результат, Пояснить, откуда `Inf`):

```
>> E = A./B
E =
       Inf   10.0000
    -2.0000    0.7143
>> A = [1 10;2 5]
```

Уберем *точку* перед операцией деления:

```
>> E = A/B
E =
    17    -1
    19    -2
```

Получен другой результат — результат *матричного* деления, о котором пойдет речь далее.

Выполним поэлементное ***левое деление*** матриц **A** и **B**:

```
>> P = A.\B
P =
         0    0.1000
   -0.5000    1.4000
```

Результат соответствует поэлементному делению матрицы **B** на матрицу **A** (пояснить).

Вывод: для *поэлементного* деления матриц A на B следует использовать операцию *правого* деления **A./B.**

Выполним поэлементное ***возведение в степень*** для матриц **A** и **B**:

```
>> Q = A.^B
Q =
  1.0e+004 *
    0.0001    0.0010
    0.0001    7.8125
```

Основание задается элементами матрицы **A**, а показатель степени — соответствующими элементами матрицы **B**. Например, проверим для элемента матрицы **A**, равного 5, который возводится в степень 7, заданную соответствующим элементом матрицы **B**:

```
>> 5.^7
ans =
       78125
```

Допустимо выполнение поэлементных матричных операций со ***скаляром***. Например, одновременное возведение всех элементов матрицы **A** в квадрат:

```
>> V = A.^2
V =
     1   100
     4    25
```

Теперь рассмотрим вторую группу поэлементных операций с матрицами — вычисление ***элементарных функций, аргументы которых — матрицы***. Для этого используем сформированные выше матрицы **A** и **B**:

```
>> M = sqrt(sin(A)+cos(B))
M =
   1.3570                  0 + 0.0610i
   1.2040                  0 + 0.4528i
```

Способность одновременного вычисления функции для всех элементов матрицы является уникальным свойством MATLAB, благодаря которому достигается исключительно высокая производительность данной системы.

2.5. Операции с матрицами в задачах линейной алгебры

В разд.2.3 была приведена классификация операций с матрицами. В этом разделе рассматриваются матричные операции из подгруппы "*операции с матрицами в задачах в задачах линейной алгебры*", к которым относятся:

- ☐ арифметические матричные операции: сложение, вычитание, умножение, возведение в степень;
- ☐ транспонирование и эрмитово сопряжение матриц;
- ☐ вычисление основных характеристик матрицы;
- ☐ обращение матрицы;

- матричное деление;
- разложение матриц.

Рассмотрим подробнее.

2.6. Матричное сложение, вычитание, умножение и возведение в степень

Сложение и вычитание матриц возможно только для матриц *одинакового* размера, и тождественно операциям *поэлементного* сложения и вычитания матриц, рассмотренным ранее.

Для операций сложения и вычитания матриц справедливы обычные законы арифметики:

$$\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A};$$

$$\mathbf{A}-\mathbf{B}=-\mathbf{B}+\mathbf{A}.$$

Умножение матрицы на скаляр тождественно операции поэлементного умножения матрицы на скаляр, поэтому символ операции умножения можно использовать и с точкой и без нее:

```
>> A = [1 2;3 4]
A =
     1     2
     3     4
>> A.*2
ans =
     2     4
     6     8
>> A*2
ans =
     2     4
     6     8
```

Умножение матрицы на матрицу возможно только в том случае, если размеры матриц-сомножителей $\mathbf{A}$ и $\mathbf{B}$ согласованы, а именно: число ***столбцов*** n матрицы $\mathbf{A}$ размером $m\times n$ равно числу ***строк*** n матрицы $\mathbf{B}$, т. е. матрица $\mathbf{B}$ имеет размер $n\times p$.

Произведение матриц $\mathbf{A}\times\mathbf{B}$ представляет собой матрицу $\mathbf{C}$ размером $m\times p$, элементы которой c_{ik}, $i=1,2,\ldots,m$, $k=1,2,\ldots,p$, равны *сумме локальных произведений* соответственных элементов *i*-й строки матрицы $\mathbf{A}$ и *k*-го столбца матрицы $\mathbf{B}$:

$$c_{ik}=\sum_{j=1}^{n}a_{ij}b_{jk}.$$

Пример умножения матрицы $\mathbf{A}$ размером $m\times n=2\times3$ на матрицу $\mathbf{B}$ размером $n\times p=3\times2$, Произведение — матрица $\mathbf{C}$ размером $m\times p=2\times2$:

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

```
    22    28
    49    64
```

В общем случае умножение матриц ***не коммутативно***:

$$\mathbf{AB} \neq \mathbf{BA}.$$

Возведение матрицы $\mathbf{A}$ ***в целую положительную степень*** q возможно только для ***квадратных*** матриц и тождественно умножению матрицы $\mathbf{A}$ саму на себя раз q. Например, возведем в квадрат полученную выше матрицу `C`:

```
>> D = C^2
D =
        1856        2408
        4214        5468
```

Напомним, что при наличии точки в символе операции в квадрат возводятся все элементы матрицы:

```
>> D = C.^2
D =
         484         784
        2401        4096
```

2.7. Транспонирование и эрмитово сопряжение матриц

Транспонирование *матрицы* $\mathbf{A}$ — это операция замены ее строк столбцами:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1n} \\ \dots & & \dots & & \dots \\ a_{i1} & \dots & a_{ij} & \dots & a_{in} \\ \dots & & \dots & & \dots \\ a_{m1} & \dots & a_{mj} & \dots & a_{mn} \end{bmatrix} \Rightarrow \mathbf{A}' = \begin{bmatrix} a_{11} & \dots & a_{i1} & \dots & a_{m1} \\ \dots & & \dots & & \dots \\ a_{1j} & \dots & a_{ij} & \dots & a_{mj} \\ \dots & & \dots & & \dots \\ a_{1n} & \dots & a_{in} & \dots & a_{mn} \end{bmatrix},$$

где $\mathbf{A}'$ — *транспонированная* матрица.

В MATLAB для транспонирования матрицы используется символ " ' " (апостроф):

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

Эрмитово сопряжение *матрицы* — это операция транспонирования матрицы с одновременной заменой ее элементов на комплексно сопряженные (пояснить в примере):

```
>> A =[3+2i 4-5i;7-5i 1+i]
A =
   3.0000 + 2.0000i   4.0000 - 5.0000i
   7.0000 - 5.0000i   1.0000 + 1.0000i
>> A'
ans =
   3.0000 - 2.0000i   7.0000 + 5.0000i
   4.0000 + 5.0000i   1.0000 - 1.0000i
```

2.8. Вычисление основных характеристик матрицы

С основными характеристиками матрицы и их вычислением в MATLAB можно познакомиться в [ЦОС. Моделирование в MATLAB], Солонина, Арбузов, 2008. Мы познакомимся с двумя основными характеристиками матрицы:

- определить (детерминант);
- норма.

Определитель (детерминант) *квадратной* матрицы $\mathbf{A}$ порядка n — скаляр — вычисляется с помощью функции **det**:

```
>> A = [1 2 3;4 8 6;7 10 9]
A =
     1     2     3
     4     8     6
     7    10     9
>> det(A)
ans =
   -24
```

Перечислим основные свойства определителей:

1. Определитель *равен нулю* тогда и только тогда, когда столбцы (строки) матрицы *линейно зависимы*, т. е. когда хотя бы один из них может быть представлен в виде линейной комбинации остальных.

   Например, элементы второй строки A равны элементам первой строки, умноженным на два:

   ```
   >> A = [1 2 3;2 4 6;7 10 9]
   A =
        1     2     3
        2     4     6
        7    10     9
   >> det(A)
   ans =
        0
   ```

2. Если хотя бы один столбец (строка) матрицы $\mathbf{A}$ — нулевой, то ее определитель равен нулю:

   ```
   >> A = [1 2 3;2 4 6;0 0 0]
   A =
        1     2     3
        2     4     6
        0     0     0
   >> det(A)
   ans =
        0
   ```

3. Определитель единичной матрицы $\mathbf{I}$ равен единице:

   ```
   >> I = eye(3)
   I =
        1     0     0
        0     1     0
        0     0     1
   >> det(I)
   ans =
        1
   ```

4. При транспонировании матрицы $\mathbf{A}$ ее определитель не меняется:

   ```
   >> A = [1 2 3;2 7 10;5 11 0];
   >> det(A)
   ans =
   ```

```
    -49
>> det(A')
ans =
    -49
```

5. Матрицу $\mathbf{A}$ называют ***вырожденной*** (особенной, сингулярной), если ее определитель равен нулю, и *невырожденной* (не особенной, не сингулярной) в противном случае.

Норма матрицы $\mathbf{A}$ — это скаляр, с помощью которого интегрально оцениваются значения элементов матрицы.

Среди норм матрицы $\mathbf{A}$ выделим следующие основные:

□ норма $\|\mathbf{A}\|_1$ — это максимальная сумма модулей элементов в *столбце*:

$$\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^{n} |a_{ij}| ;$$

□ норма $\|\mathbf{A}\|_\infty$ — это максимальная сумма модулей элементов *в строке*:

$$\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^{m} |a_{ij}| ;$$

□ норма $\|\mathbf{A}\|_2$ (***евклидова норма***) — это корень квадратный из суммы квадратов модулей всех элементов матрицы:

$$\|\mathbf{A}\|_2 = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} .$$

Аналогичные нормы существуют для *векторов*, которые будут вычисляться на лабораторных занятиях.

Норма матрицы и вектора вычисляется с помощью функции:

`norm(A,p)`

где `p` — параметр, указывающий норму и принимающий значения: `1` — для $\|\mathbf{A}\|_1$; `2` — для $\|\mathbf{A}\|_2$ (*по умолчанию*); `inf` — для $\|\mathbf{A}\|_\infty$:

Вычислим нормы матрицы **`A`** (пояснить):

```
> >> A = [1 2;4 -5]
A =
     1     2
     4    -5
>> NORMS = [norm(A,1) norm(A,inf) norm(A)]
NORMS =
    7.0000    9.0000    6.4787
```

2.9. Обращение матрицы

Матрицу $\mathbf{B}$ называют *обратной* к матрице $\mathbf{A}$, если произведение этих матриц дает *единичную* матрицу $\mathbf{I}$:

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I} .$$

Матрицу $\mathbf{B}$, обратную к матрице $\mathbf{A}$, обозначают как $\mathbf{A}^{-1}$:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I} .$$

Операция вычисления матрицы $\mathbf{A}^{-1}$, называемая ***обращением*** матрицы $\mathbf{A}$, возможна только для *квадратной* матрицы с *определителем*, *не равным нулю*.

Обращение матрицы выполняется с помощью функции **`inv`**:

```
>> A=[1 2;5 8]
A =
```

```
     1     2
     5     8
>> det(A)
ans =
    -2
>> B = inv(A)
B =
   -4.0000    1.0000
    2.5000   -0.5000
>> A*B
ans =
     1     0
     0     1
```

2.10. Матричное деление

В табл. 2 представлены две операции матричного деления:

- левое матричное деление — `A\B`, эквивалентное операции $\mathbf{A}^{-1}\mathbf{B}$, т. е. `inv(A)*B`;
- правое матричное деление — `A/B`, эквивалентное операции $\mathbf{AB}^{-1}$, т. е. `A*inv(B)`.

Символ ***левого*** матричного деления "\" используют при решении систем линейных алгебраических уравнений (СЛАУ):

$$\mathbf{Ax} = \mathbf{b}, \tag{2.1}$$

где $\mathbf{A}$ — матрица коэффициентов при неизвестных; $\mathbf{b}$, $\mathbf{x}$ — векторы-столбцы свободных членов и неизвестных соответственно.

Умножив обе части (2.1) на $\mathbf{A}^{-1}$, получим решение СЛАУ в виде:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}. \tag{2.2}$$

В MATLAB это соответствует операции `inv(A)*B`, т. е. ***левому*** матричному делению **`x = A\b`**:

Поясним на примере решения СЛАУ:

$$\begin{cases} 2x_1 - x_2 + 4x_3 = 9; \\ x_1 - 2x_2 - 3x_3 = -2; \\ 4x_1 - x_2 - x_3 = 10, \end{cases} \tag{2.3}$$

где:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 4 \\ 1 & -2 & -3 \\ 4 & -1 & -1 \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 9 \\ -2 \\ 10 \end{bmatrix}. \tag{2.4}$$

Сформируем матрицу $\mathbf{A}$ и вектор $\mathbf{b}$, и используем операцию *левого* деления для решения СЛАУ (2.3) в виде (2.2) — определения вектора $\mathbf{x}$:

```
>> A = [2 -1 4;1 -2 -3;4 -1 -1]
A =
     2    -1     4
     1    -2    -3
     4    -1    -1
>> b = [9;-2;10]
b =
     9
    -2
```

```
    10
>> x = A\b
x =
     3
     1
     1
```

Проверим правильность решения. Выполнив умножение в левой части (2.1), получим вектор **b**:

```
>> A*x
ans =
     9
    -2
    10
```

Преимуществом MATLAB является *возможность одновременного решения **нескольких** СЛАУ*, отличающихся вектором свободных членов. В этом случае вектор **b** будет представлен ***матрицей*** свободных членов, каждый столбец которой соответствует вектору свободных членов одной СЛАУ:

```
>> A=[2 -1 4;1 -2 -3;4 -1 -1];
>> b = [9 3 7;-2 1 0;10 15 3]
b =
     9     3     7
    -2     1     0
    10    15     3
>> x = A\b
x =
    3.0000    4.2432    0.7027
    1.0000    2.6757   -1.2703
    1.0000   -0.7027    1.0811
```

Первый столбец матрицы соответствует решению СЛАУ (2.2).

2.11. Разложение матриц

Разложением матрицы называют ее представление в виде *произведения матриц*.

Разложение матриц используется во многих приложениях. Мы познакомимся с одним из видов разложения — *LU-разложением*, которое, в частности, используется при решении СЛАУ.

Основными преимуществами решения СЛАУ на основе *LU-разложения* являются:

- существенно меньший объем вычислений, чем при обращении матрицы в (2.2);

  Это крайне важно при реализации численного метода решения СЛАУ на цифровом устройстве, например, ЦПОС

- меньшая чувствительность решения к погрешностям исходных данных, т. е. небольшим изменением элементов матрицы $\mathbf{A}$ и вектора $\mathbf{b}$ в (2.1).

В MATLAB *LU-разложение* матрицы $\mathbf{A}$ представлено в виде:

$$\mathbf{PA} = \mathbf{LU}, \tag{2.5}$$

где:

$\mathbf{L}$ — нижняя (lower) треугольная матрица с единицами в главной диагонали и *ненулевыми* элементами *ниже* нее;

$\mathbf{U}$ — верхняя (upper) треугольная матрица с *ненулевыми* элементами на главной диагонали и *выше* нее;

$\mathbf{P}$ — вспомогательная матрица нулей и единиц, формирующая столбцы матрицы произведения $\mathbf{PA}$ так, чтобы *первым* элементом каждого столбца был ***главный элемент*** — наибольший по модулю, что повышает точность решения СЛАУ.

LU-разложение матрицы выполняется с помощью функции:

```
[L,U,P]=lu(A)
```

Например, выполним LU-разложение матрицы **A** в (2.4) (пояснить **L** и **U**):

```
>> A=[2 -1 4;1 -2 -3;4 -1 -1]
A =
     2    -1     4
     1    -2    -3
     4    -1    -1
>> [L,U,P]=lu(A)
L =
    1.0000         0         0
    0.2500    1.0000         0
    0.5000    0.2857    1.0000
U =
    4.0000   -1.0000   -1.0000
         0   -1.7500   -2.7500
         0         0    5.2857
P =
     0     0     1
     0     1     0
     1     0     0
```

В MATLAB решение СЛАУ (2.1) на основе *LU-разложения* реализуется с помощью функции:

```
x = linsolve(A,b)
```

Решив СЛАУ (2.3), получим тот же результат:

```
>> A=[2 -1 4;1 -2 -3;4 -1 -1]
A =
     2    -1     4
     1    -2    -3
     4    -1    -1
>> b = [9;-2;10]
b =
     9
    -2
    10
>> x = linsolve(A,b)
x =
     3
     1
     1
```

Более подробно с разложением матриц можно познакомиться в [ЦОС. Моделирование в MATLAB], 2008.

2.13. Операции с матрицами в задачах математической статистики

В разд.2.3 была приведена классификация операций с матрицами. В этом разделе рассматриваются матричные операции из подгруппы "*операции с матрицами в задачах математической статистики*".

Для решения задач математической статистики в MATLAB предусмотрен большой набор встроенных функций, некоторые из которых приведены в табл. 3.

Таблица 3. *Функции математической статистики*

Функция	Назначение
max(A)	Максимальные элементы столбца
min(A)	Минимальные элементы столбца
sum(A)	Сумма элементов столбца
mean(A)	Математическое ожидание (среднее значение) элементов столбца
var(A,1)	Дисперсия элементов столбца, вычисляемая по формуле: $$\sigma_j^2 = \frac{\left(a_{1j} - \overline{a}_j\right)^2 + \left(a_{2j} - \overline{a}_j\right)^2 + ... + \left(a_{mj} - \overline{a}_j\right)^2}{m}$$

Проиллюстрируем использование данных функций на примерах решения простейших задач математической статистики.

Запишем матрицу $\mathbf{X}$ размером $M \times N$ из M *последовательностей* случайных чисел длины N:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1N} \\ \dots & & \dots & & \dots \\ x_{i1} & \dots & x_{ij} & \dots & x_{iN} \\ \dots & & \dots & & \dots \\ x_{M1} & \dots & x_{Mj} & \dots & x_{MN} \end{bmatrix}.$$

В качестве последовательностей случайных чисел выберем последовательности чисел, распределенных по *нормальному* закону с математическим ожиданием, равным 0, и дисперсией, равной 1.

Такие последовательности называют *нормальным белым шумом* (дискретным).

Одну (любую) строку матрицы $\mathbf{X}$ называют ***реализацией*** нормального белого шума, а совокупность всех M строк (M реализаций) — ***ансамблем реализаций*** нормального белого шума.

Сгенерируем матрицу $\mathbf{X}$ размером 100×100 с помощью функции **randn** (табл. 1):

```
>> X = randn(100,100);
```

и определим *статистические характеристики* нормального белого шума:

- математическое ожидание;

  Математическим ожиданием случайной последовательности называют ее среднее значение по ансамблю реализаций.

  Вычислим математическое ожидание **M_X** .с помощью функции **mean(X)** для *столбцов* матрицы, а затем их среднее значение **M** с помощью той же функции:

  ```
  >> M_X = mean(X); M = mean(M_X)
  M =
      0.0135
  ```

- дисперсию.

  Дисперсией случайной последовательности называют ее среднеквадратическое отклонение от среднего по ансамблю реализаций.

  Вычислим дисперсию **D_X** .с помощью функции **var(X,1)** для *столбцов* матрицы, а затем их среднее значение **D** с помощью функции **mean(D_X)**:

  ```
  >> D_X = var(X,1); D = mean(D_X)
  D =
      0.9667
  ```

Полученные значения математического ожидания и дисперсии отличаются от истинных значений, поэтому их называют ***оценками*** математического ожидания и дисперсии.

Оценки статистических характеристик будут стремиться к истинным значениям при $M \to \infty$.

Теперь вместо 100 реализаций белого шума длины 100 рассмотрим одну его реализацию большой длины $100 \times 100 = 1 \times 10000$. Сгенерируем нормальный белый шум в виде вектора **Y** и определим ***оценки*** математического ожидания и дисперсии:

```
>> Y = randn(1,10000); M = mean(Y), D = var(Y,1)
M =
    0.0027
D =
    1.0090
```

Оценки статистических характеристик будут стремиться к истинным значениям при $N \to \infty$.