

# Обработка массивов

Поиск элементов массива с максимальным и минимальным значениями:

- **max(A)** возвращает наибольший элемент, если **A** – вектор; или возвращает вектор-строку, содержащую максимальные элементы каждого столбца, если **A** – матрица;
- **max(A,B)** возвращает массив того же размера, что и **A** (или **B**), каждый элемент которого есть максимальный из соответствующих элементов этих массивов;
- **max(A,[ ],dim)** возвращает наибольший элемент по столбцам или по строкам матрицы в зависимости от значения скаляра **dim**.

Например, **max(A,[ ],1)** возвращает максимальные элементы каждого столбца матрицы **A**;

- **[C,I] = max(A)** – кроме максимальных значений, возвращает вектор индексов элементов с этими значениями.

Примеры:

```
>> A=magic(7)
```

**A =**

```
30  39  48  1 10  19  28  
38  47  7 9 18  27  29  
46  6 8 17  26  35  37  
5 14  16  25  34  36  45  
13  15  24  33  42  44  4  
21  23  32  41  43  3 12  
231  40  49  2 11  20
```

**>> C = max(A)**

```
C = 46  47  48  49  43  44  45
```

**>> C = max(A,[ ],1)**

```
C = 46  47  48  49  43  44  45
```

```
>> C = max(A,[ ],2)    C =  
48  
47  
46  
45  
44  
43  
49
```

```
>> [C,I] = max(A)
```

```
C = 46    47    48    49    43    44    45
```

```
I = 3     2     1     7     6     5     4
```

Для нахождения элемента массива с минимальным значением служат подобные функции:

```
min(A) min(A,B) min(A,[ ],dim) [C,I] = min(A)
```

# Сортировка элементов массива

**[s, i] = sort (x)**

**[s, i] = sort (x, dim)**

**[s, i] = sort (x, mode)**

**[s, i] = sort (x, dim, mode)**

- **sort(A)** в случае одномерного массива **A** сортирует и возвращает элементы по возрастанию их значений; в случае двумерного массива происходят сортировка и возврат элементов каждого столбца.

**i** - массив индексов исходного массива

**mode='ascend'** - сортировка по возрастанию,

**'descend'** - по убыванию.

**dim=1** для матрицы сортировка по столбцам, **2** - по строкам.

**V =**

<b>-0.51301</b>	<b>-0.52824</b>	<b>-0.21580</b>	<b>-0.64126</b>
<b>0.14051</b>	<b>-0.57329</b>	<b>-0.42226</b>	<b>0.68796</b>
<b>-0.17928</b>	<b>0.36181</b>	<b>0.59941</b>	<b>-0.69113</b>
<b>-0.77139</b>	<b>0.32716</b>	<b>-0.17473</b>	<b>0.51710</b>

**>> [D,I]=sort(V)**

**D =**

<b>-0.77139</b>	<b>-0.57329</b>	<b>-0.42226</b>	<b>-0.69113</b>
<b>-0.51301</b>	<b>-0.52824</b>	<b>-0.21580</b>	<b>-0.64126</b>
<b>-0.17928</b>	<b>0.32716</b>	<b>-0.17473</b>	<b>0.51710</b>
<b>0.14051</b>	<b>0.36181</b>	<b>0.59941</b>	<b>0.68796</b>

**I =**

<b>4</b>	<b>2</b>	<b>2</b>	<b>3</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
<b>3</b>	<b>4</b>	<b>4</b>	<b>4</b>
<b>2</b>	<b>3</b>	<b>3</b>	<b>2</b>

**[s, i] = sortrows (A)**

**[s, i] = sortrows (A, c)**

Сортировка строк, в соответствии с сортировкой первого столбца (по умолчанию) или столбца с номером **c**. (если **c<0** сортировка по убыванию)

**I = >> F=sortrows(I)      >> F1=sortrows(I,-2)**

**F =**

**F1 =**

<b>4</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>4</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>2</b>
<b>3</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>	<b>2</b>	<b>2</b>	<b>3</b>
<b>2</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>4</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>

## Задание

Решить задачу на собственные значения для матрицы

$$\begin{pmatrix} 1 & 0.9 & 0.8 & 0.7 \\ 0.9 & 1 & 0.9 & 0.8 \\ 0.8 & 0.9 & 1 & 0.9 \\ 0.7 & 0.8 & 0.9 & 1 \end{pmatrix}$$

Отсортировать собственные векторы в соответствии с собственными значениями, которые должны быть отсортированы в убывающем порядке.

# Статистическая обработка данных

Статистическая обработка данных, представляющих собой реализации случайных величин, заключается в вычислении таких характеристик случайных величин, как **математическое ожидание, дисперсия, среднее квадратическое отклонение, начальные и центральные моменты, коэффициент асимметрии и эксцесса, корреляционная матрица, эмпирический закон распределения.**

# Математическое ожидание (среднее значение)

`mean (x)`

`mean (x, dim)`

`mean (x, opt)`

`mean (x, dim, opt)`

Вычисляется по формуле:

$$m = \frac{1}{n} \sum_{i=1}^n x_i$$

Если  $X$  – матрица, то среднее вычисляется для каждого столбца матрицы. Результат вектор-строка.

**dim=1** (по умолчанию) для матрицы вычисление по столбцам, **2** - по строкам.

**opt** – определяет следующие выборы вычисления:

**“a”** – среднее арифметическое (по умолчанию)

**“h”** – среднее гармоническое  $h = \frac{1}{\frac{1}{n} \sum_{i=1}^n \frac{1}{x_i}}$

**“g”** – среднее геометрическое  $g = \left( \prod_{i=1}^n x_i \right)^{\frac{1}{n}}$   
Элементы  $x$  – положительны.

$$\min(x) \leq h \leq g \leq m \leq \max(x)$$

# Медиана выборки

**median** ( $x$ )

**median** ( $x, dim$ )

Для вычисления медианы нужно отсортировать массив  $X$  ( $s = \text{sort}(X)$ ). Тогда:

$$\text{median} = \begin{cases} s(n / 2), & \text{если } n \text{ нечетно} \\ 0.5(s(n / 2) + s((n + 1) / 2)), & \text{если } n \text{ четно} \end{cases}$$

# Стандартное и среднее квадратическое отклонения

**std** ( $x$ )

**std** ( $x, opt$ )

**std** ( $x, opt, dim$ )

$$std(x) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - m)^2}$$

$$std(x,1) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - m)^2}$$

# Коэффициент асимметрии распределений случайных величин

**skewness** ( $x$ )

**skewness** ( $x, flag$ )

**skewness** ( $x, flag, dim$ )

Асимметрия – мера скошенности графика плотности распределения по отношению к нормальному распределению, вычисляется по формуле:

$$A = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \text{mean}(x))^3}{\text{std}(x)^3}$$

# Коэффициент эксцесса распределения случайной величины

**kurtosis** ( $x$ )

**kurtosis** ( $x, flag$ )

**kurtosis** ( $x, flag, dim$ )

Эксцесс – мера высоты графика плотности распределения по отношению к нормальному распределению (=3), *вычисляется по формуле:*

$$E = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \text{mean}(x))^4}{\text{std}(x)^4}$$

# Квартили

Предоставляют важную информацию о структуре вариационного ряда. Вместе с медианой они делят вариационный ряд на 4 равные части. Квартилей две, их обозначают символами  $Q$ , верхняя и нижняя квартиль. 25% значений меньше, чем нижняя квартиль, 75% значений меньше, чем верхняя квартиль.

**statistics** (*x*)

**statistics** (*x*, *dim*)

Для вектора *X* возвращает следующие статистические характеристики:

***min(x)*, 1-й квартиль, медиану, третий квартиль, *max(x)*, *mean(x)*, *std(x)*, асимметрию, эксцесс**

**Пример**

```
>> X=randn(1,1000);
```

```
>> statistics (X)
```

<b>min</b>	<b>1</b>	<b>med</b>	<b>3</b>
<b>-2.724709</b>	<b>-0.671729</b>	<b>0.029226</b>	<b>0.690743</b>
<b>max</b>	<b>mean</b>	<b>std</b>	<b>ass</b>
<b>2.703021</b>	<b>0.012462</b>	<b>0.963554</b>	<b>-0.037872</b>
<b>2.729541</b>			

```
>> X=randn(1,10000);
```

```
>> statistics (X)
```

min	1	med	3
-3.5173146	-0.6750578	-0.0096442	0.6592135
max	mean	std	ass
3.6071385	-0.0047585	0.9926368	0.0312958
2.9634857			

```
>> X=randn(1,100000); statistics (X)
```

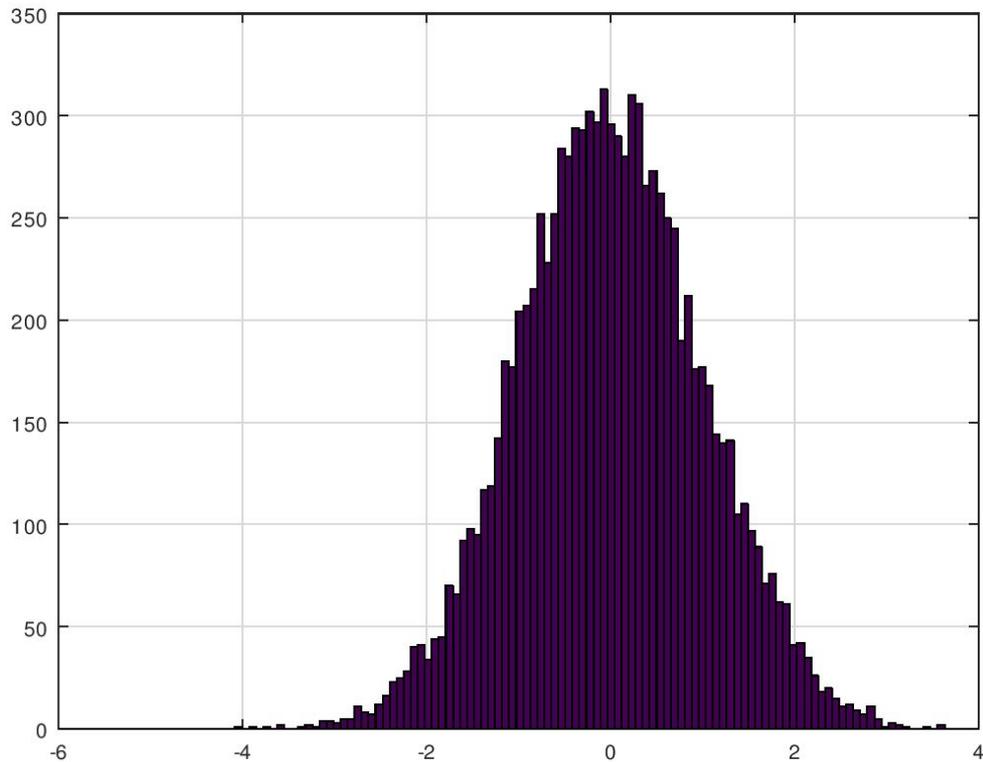
min	1	med	3
-4.2626814	-0.6733263	0.00220971	0.67382284
max	mean	std	ass
4.8990076	0.0016832	0.99919802	-0.00013567
2.99073352			

# Построение гистограммы

**hist** (*y*)

**hist** (*y*, *nbins*)

```
>> X=randn(1,10000); hist (X,101)
```



# Дискретное преобразование Фурье

В цифровой обработке сигналов широкое применение находит преобразование Фурье, которое сопоставляет каждой функции времени ( $t$ ) функцию от частоты ( $f$ ).

Это преобразование задается парой соотношений:

$$y(f) = \int_{-\infty}^{\infty} s(t) e^{-j \cdot 2 \cdot \pi \cdot t \cdot f} dt$$

$$s(t) = \int_{-\infty}^{\infty} y(f) e^{j \cdot 2 \cdot \pi \cdot t \cdot f} df$$

На ЭВМ выполнить операции над непрерывными функциями невозможно, поэтому непрерывные функции дискретизируются по времени и по частоте и преобразование Фурье выполняется над векторами: ( $j$  – мнимая единица)

$$s_i = s(i \cdot \Delta t), y_k = y(k \cdot \Delta f)$$

$$y_k = \sum_{i=0}^{N-1} s_i \cdot e^{-j2\pi \frac{i \cdot k}{N}}, s_i = \frac{1}{N} \sum_{k=0}^{N-1} y_k \cdot e^{-j2\pi \frac{i \cdot k}{N}}$$

**fft** ( $x$ )                     $N$  – число элементов вектора  $x$

**fft** ( $x, n$ )                 $N=n$

**fft** ( $x, n, dim$ )         $N=n$

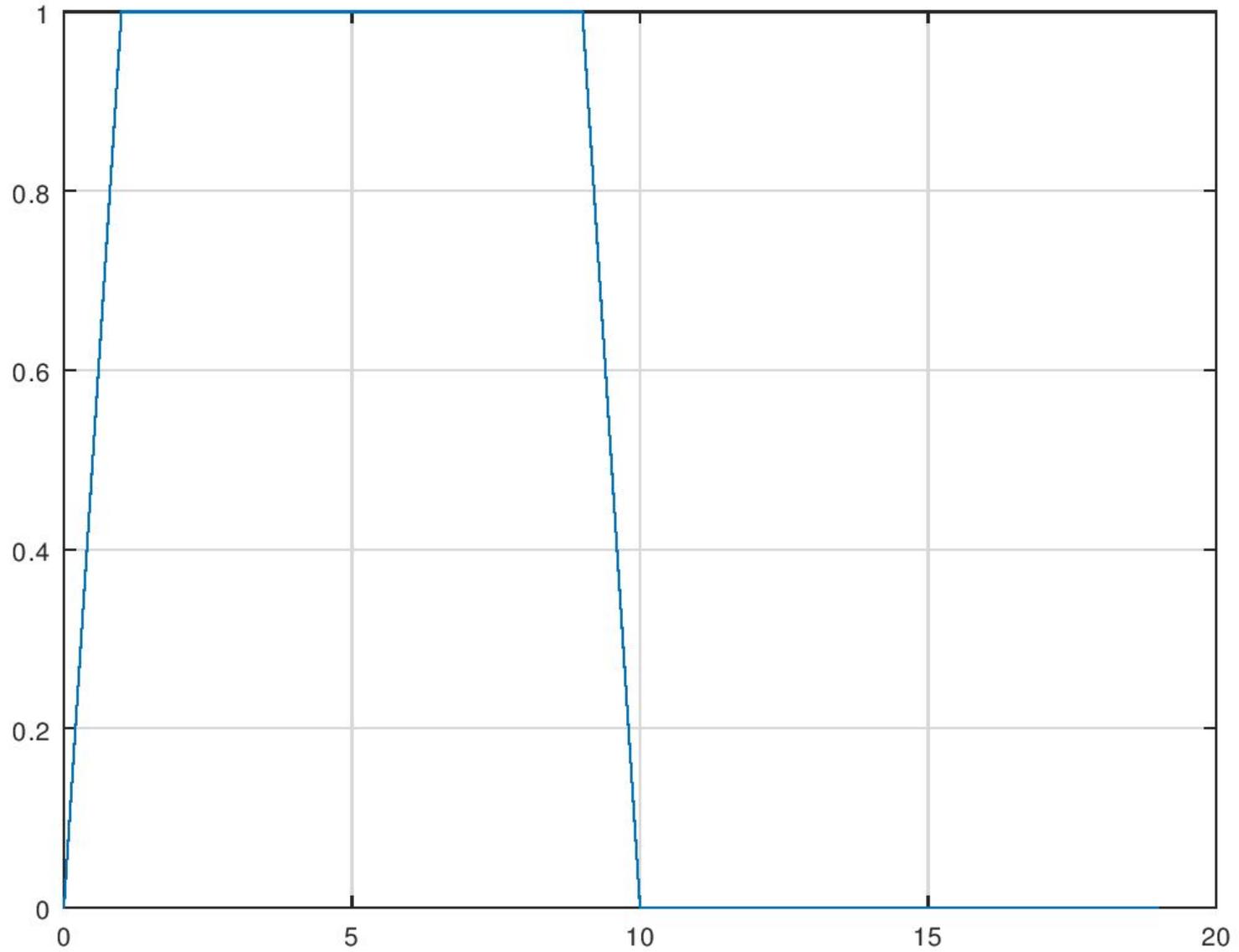
**Задание** Найти преобразование Фурье сигнала, задаваемого соотношением:

$$s(t) = \begin{cases} 1, & t \in [0, \tau] \\ 0, & t \notin [0, \tau] \end{cases}$$

Выберите интервал дискретизации по времени  $dt$ , время наблюдения  $T > 2 \cdot \tau$ , число отсчетов сигнала  $n = \tau / dt$ , общее число отсчетов  $N = T / dt$

Задайте анонимную функцию, вычисляющую сигнал, выполните преобразование Фурье и постройте график модуля преобразования Фурье.

# Сигнал



# Модуль преобразования Фурье

