

Министерство образования Российской Федерации
Ставропольский государственный университет

**Компьютерная обработка
результатов измерения
(учебное пособие)**

Ставрополь, 2008 г.

Содержание

1	Статистика и вероятность. Случайные величины и распределения	4
1.1	Краткие теоретические сведения	4
1.2	Примеры практических заданий	8
1.3	Задания для самостоятельного выполнения	12
1.4	Контрольные вопросы	13
2	Теория физических измерений. Систематические и случайные погрешности	14
2.1	Краткие теоретические сведения	14
2.2	Примеры практических заданий	19
2.3	Задания для самостоятельного выполнения	22
2.4	Контрольные вопросы	24
3	Теория оценок	25
3.1	Краткие теоретические сведения	25
3.2	Примеры практических заданий	27
3.3	Задания для самостоятельного выполнения	34
3.4	Контрольные вопросы	35
4	Системы линейных уравнений. Степенные уравнения. Дифференциальные уравнения	37
4.1	Краткие теоретические сведения	37
4.2	Примеры практических заданий	40
4.3	Задания для самостоятельного выполнения	47
4.4	Контрольные вопросы	48
5	Анализ временных рядов. Фурье и вейвлет-анализ	50
5.1	Краткие теоретические сведения	50
5.2	Примеры практических заданий	54
5.3	Задания для самостоятельного выполнения	61
5.4	Контрольные вопросы	62
6	Обработка изображений	64
6.1	Краткие теоретические сведения	64
6.2	Примеры практических заданий	69
6.3	Задания для самостоятельного выполнения	77
6.4	Контрольные вопросы	78

Задания для СКР	80
1 Краткие теоретические сведения	80
2 Примеры практических заданий	89
2.1 Общее оформление текста в \LaTeX	89
2.2 Оформление формул	91
2.3 Общее оформление статьи	92
3 Задания для самостоятельного выполнения	93
4 Контрольные вопросы	95
Список литературы	96

Компьютерная обработка

1 Статистика и вероятность. Случайные величины и распределения

1.1 Краткие теоретические сведения

Случайной величиной называется величина X , если все ее возможные значения образуют конечную или бесконечную последовательность чисел x_1, \dots, x_N , и если принятие ею каждого из этих значений есть случайное событие. **Вероятностью** наступления данного случайного события x_k называется предел относительной частоты наступления данного события n_k/N :

$$P(x_k) = \lim_{N \rightarrow \infty} \frac{n_k}{N}.$$

Случайная величина может быть не только дискретной, но и **непрерывной**. В этом случае она может принимать любое значение из заданной области действительных чисел. Непрерывная случайная величина характеризуется **плотностью вероятности**:

$$\varphi(x) = \lim_{\Delta x \rightarrow 0} \frac{P(x < X < x + \Delta x)}{\Delta x} = \frac{dP}{dx}.$$

Вероятность попадания значений X в интервал (x_1, x_2) можно вычислить по формуле:

$$P(x_1 < X < x_2) = \int_{x_1}^{x_2} \varphi(x) dx.$$

Вероятность того, что значения случайной величины не превышают заданного числа x называют **функцией распределения**: $F(x) \equiv P(X \leq x) = \int_{-\infty}^x \varphi(x) dx$. Так как вероятность попадания значений (X) в промежуток $(-\infty, \infty)$ равна единице, плотность вероятности должна быть **нормирована**:

$$\int_{-\infty}^{\infty} \varphi(x) dx = 1.$$

Две случайные величины X и Y называются **независимыми**, если наступление одного из событий x_n никак не сказывается на результате события y_n .

В этом случае вероятность совместного наступления событий x_n и y_n равна $P(x_n y_n) = P(x_n)P(y_n)$. Дальнейшие формулы в этом разделе подразумевают независимость входящих в них случайных величин, если зависимость не указана явно.

Набор случайных величин X можно охарактеризовать **средним арифметическим**: $\langle X \rangle = 1/N \sum_{n=1}^N x_n$. Для малых N данная величина будет отличаться от **математического ожидания**, определяемого по формуле

$$M(X) \equiv \bar{X} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N x_n \quad \text{и} \quad M(X) = \int_{-\infty}^{\infty} x \varphi(x) dx,$$

однако **закон больших чисел** говорит, что чем большей является величина N , тем ближе $\langle X \rangle$ к \bar{X} .

Помимо математического ожидания набор случайных величин характеризуется медианой и модой. **Мода** — значение во множестве наблюдений, которое встречается наиболее часто. Иногда мод может быть больше одной. В этом случае можно сказать, что совокупность *мультимодальна*. Как правило мультимодальность указывает на то, что набор данных не подчиняется нормальному распределению. **Медиана** — возможное значение признака, которое делит отсортированную совокупность на две равные части: 50% «нижних» единиц ряда данных будут иметь значение признака не больше, чем медиана, а 50% «верхних» — значения признака не меньше, чем медиана.

Свойства математического ожидания: $M(\sum \mathbf{c}_n \cdot X_n) = \sum \mathbf{c}_n \cdot M(X_n)$, где \mathbf{c}_n — постоянная величина; $M(\prod X_n) = \prod M(X_n)$; $M(f(x)) = \int_{-\infty}^{\infty} f(x) \varphi(x) dx$.

Если $f(x) = (x - x_0)^n$, говорят, что $M(f(x))$ является **моментом** случайной величины порядка n . При $x_0 = 0$ момент называют **начальным**, а при $x_0 = \bar{X}$ — **центральным**.

Центральный момент второго порядка называют **дисперсией**: $D(X) = \overline{(x - \bar{x})^2}$. Разброс случайной величины относительно математического ожидания характеризуется **средним квадратичным отклонением** $\sigma = \sqrt{D}$. Дисперсию можно вычислить и по упрощенному выражению:

$$D = \overline{(x - \bar{x})^2} = \overline{x^2} - 2\overline{x\bar{x}} + \overline{\bar{x}^2} = \overline{x^2} - \bar{x}^2 \equiv M(x^2) - [M(x)]^2.$$

- ▶ Если функция плотности вероятности случайного события описывается аналитически, говорят, что она подчинена некоторому **распределению**. Наиболее известными видами распределения являются непрерывное, нормальное (гауссово), Пуассона, биномиальное, экспоненциальное и многие другие.

Говорят, что случайная величина имеет непрерывное **равномерное распределение** на отрезке $[a, b]$, где $a, b \in \mathbb{R}$, если ее плотность $\varphi(x)$ имеет вид

$$\varphi(x) = \begin{cases} \frac{1}{b-a}, & x \in [a, b] \\ 0, & x \notin [a, b] \end{cases}.$$

Интегрируя, получим, для $F(x)$:

$$F(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & x \geq b \end{cases}.$$

Математическое ожидание и медиана для равномерного распределения совпадают с серединой отрезка $[a, b]$, модой же является любое значение из этого отрезка.

Важнейшую роль в физике имеет **нормальное** (гауссово) распределение. Физическая величина подчиняется нормальному распределению, когда она подвержена влиянию огромного числа случайных помех. Плотность вероятности нормального распределения имеет вид:

$$\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right),$$

Функция распределения записывается через *интеграл Римана*:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x \exp\left(-\frac{(t-\bar{x})^2}{2\sigma^2}\right) dt.$$

Особенностью нормального распределения является совпадение медианы, моды и математического ожидания.

Вероятность того, что нормальная случайная величина с параметрами \bar{x} и σ попадет в интервал (α, β) равна:

$$P(\alpha < X < \beta) = \Phi\left(\frac{\beta - \bar{x}}{\sigma}\right) - \Phi\left(\frac{\alpha - \bar{x}}{\sigma}\right),$$

где $\Phi(x)$ – **функция Лапласа**:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \exp\left(-\frac{t^2}{2}\right) dt.$$

- Для выяснения зависимости случайных величин X и Y используются такие функции, как корреляция и ковариация. **Ковариация** является мерой линейной зависимости случайных величин и определяется формулой: $\text{cov}(X, Y) = \overline{(X - \bar{X})(Y - \bar{Y})}$. Понятно, что ковариация величины с самой собой есть ее дисперсия. *Ковариация независимых случайных величин равна нулю*, обратное неверно.

Коэффициент корреляции двух величин задается формулой

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sqrt{D(X) \cdot D(Y)}}.$$

Коэффициент корреляции равен ± 1 тогда и только тогда, когда X и Y линейно зависимы. Если они независимы, $\rho_{X,Y} = 0$. Промежуточные значения коэффициента корреляции не позволяют однозначно судить о зависимости случайных величин, но позволяет предполагать степень их зависимости.

Для непрерывных величин аналогом коэффициента корреляции является *корреляционная функция*. Более часто применяют ее разновидность — **автокорреляционную функцию**:

$$\Psi(\tau) = \int f(t)f(t - \tau) dt,$$

показывающую связь сигнала (функции $f(t)$) с копией самого себя, смещенного на величину τ . Автокорреляционная функция имеет обычно максимум при $t = 0$. Имея два сигнала, представляющих собой исходный и сдвинутый на неизвестную величину z , можно определить z из корреляционной функции: эту величину однозначно покажет положение максимума корреляционной функции.

- В теории сигналов вводят понятие шума. **Шум** — беспорядочные колебания различной физической природы, отличающиеся сложностью временной и спектральной структуры. Наиболее общим видом шума является **белый шум** — стационарный шум, спектральные составляющие которого равномерно распределены по всему диапазону задействованных частот. В природе и технике «чисто» белый шум (то есть белый шум, имеющий одинаковую спектральную мощность на всех частотах) не встречается (ввиду того, что *такой сигнал имел бы бесконечную мощность*), однако под категорию белых шумов попадают любые шумы, спектральная плотность которых одинакова (или слабо отличается) в рассматриваемом диапазоне частот. Белым шумом является любой шум с фиксированной дисперсией, нулевым математическим ожиданием и автокорреляционной функцией, имеющей вид *дельта-функции Дирака*. Чаще всего белый шум моделируют гауссовским распределением, т.к. такая модель хорошо подходит для математического описания многих природных процессов.

Зашумленность сигнала характеризуют **отношением сигнал/шум** (SNR):

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right)^2,$$

где P и A — соответственно, средняя мощность и среднеквадратичное значение амплитуды сигнала и шума. Зачастую SNR выражают в децибелах:

$$\text{SNR}(\text{dB}) = 10 \lg \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) = 20 \lg \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right).$$

Причина множителя 10 становится понятной, исходя из приставки «деци», а множитель 20 возникает вследствие умножения на 2, появляющегося при логарифмировании амплитуд.

1.2 Примеры практических заданий

На этом занятии познакомимся со средой матричных вычислений Matlab и научимся выполнять в ней простейшие манипуляции с данными.

Название «Matlab» подразумевает работу с матрицами. Соответственно, все данные в Matlab представлены в матричной форме. *Этого не следует забывать при выполнении различных операций!* Чтобы присвоить переменной A значение 10.5, достаточно написать: $A=10.5$. На экране тут же отобразится значение переменной A . *Для подавления вывода следует заканчивать команды точкой с запятой.* Попробуйте набрать теперь $A=10.5$; заметили разницу?

Если вы хотите инициализировать эту переменную вектором, напишите, например: $A=[1\ 2\ 3]$ или же $A=[1,2,3]$. Вы получите вектор-строку с элементами 1, 2 и 3. Таким образом, перечислять элементы строки можно либо через запятую, либо через пробел. Чтобы получить вектор-столбец, следует разделять его элементы точкой с запятой: $A=[1;2;3]$. И, наконец, для ввода матриц: разделяйте элементы строки запятыми или пробелами, а строки — точкой с запятой. Введем единичную матрицу:

$$A=[1\ 0\ 0;0\ 1\ 0;0\ 0\ 1]$$

на экране отобразится:

```
A=
    1    0    0
    0    1    0
    0    0    1
```


Для обращения к элементу матрицы необходимо набрать его имя и в скобках указать либо абсолютный номер, либо *номер строки*, а затем *номер столбца*. Наберите теперь `A(5)`, а после — `A(2,2)`. На экране отобразится один и тот же текст: `ans=1`. И действительно, пятый по счету элемент соответствует элементу второй строки второго столбца. Программистам следует обратить внимание, что *номера массивов начинаются с единицы*, в отличие от языков программирования, где первому элементу соответствует ноль.

Допустим, у вас есть матрица-строка, а вы хотите сделать матрицу-столбец. Для этого используйте операцию транспонирования: наберите `x=[1 2 3]`, а затем `x'`. Как видите, добавление после имени матрицы апострофа означает операцию транспонирования.

Диапазоны данных можно указывать, разделяя числа двоеточием. Так, `[1:10]` дает набор целых чисел от 1 до 10, а `[1:0.5:10]` — то же, но с шагом в 0.5.

Теперь создайте еще одну матрицу

$$a=[0 \ 1 \ 0;1 \ 0 \ 1;0 \ 1 \ 0]$$

Заодно обратите внимание: в *Matlab* имена переменной зависят от регистра! Для произведения простейших операций над матрицами используйте операторы `+`, `-`, `/` и `*`. Наберите поочередно `A+a`, `A-a`, `A*a`, `A/a`. Обратите внимание, что при этом производятся именно матричные операции. Поэтому необходимо согласовывать размеры матриц (вспомните курс алгебры). Еще одним, чисто матричным оператором, является оператор левого деления `\`, использующийся при решении систем уравнений. Кроме того, используется оператор возведения в степень, `^`.

Но, допустим, вы захотите произвести поэлементное умножение или деление матриц. В этом случае *до символа операции наберите точку*. Например, попробуйте набрать `A./a`. Попробуйте эти операции с более осмысленными векторами или матрицами.

Если вы присвоили переменной *одно* значение, все вычисления с ней выполняются как со скаляром.

Кроме чисел в *Matlab* есть понятия бесконечности (`Inf`) и не-числа (`NaN`). Вы можете присвоить эти величины переменным. Обратите внимание, что `NaN` соответствует неопределенным операциям, например, пределам вида `0/0`, `∞/∞` и т.п. (однако, попробуйте операцию `Inf/0`). В отличие от `Inf`, `NaN` инвариантен относительно любых операций.

Еще одним удобством является дополнение команд и «история команд» в стиле UNIX. Набирая команду и нажимая клавишу `Tab` вы упростите себе работу: если вариант дополнения только один, *Matlab* закончит команду, иначе — выведет окно выбора со списком возможных команд. Клавишами `↑` и `↓` можно

перемещаться по истории команд, что полезно, если вам надо набирать много однотипных длинных команд, отличающихся незначительно.

В Matlab существует огромное количество функций. Обычно они имеют стандартный синтаксис типа $A=funk(b, \dots)$: аргументы функции помещаются в скобки, а ее значение приравнивается матрице. Помощь по функции `funk` можно получить командой `help funk` или нажав клавишу F1 и воспользовавшись гипертекстовой системой помощи.

- Теперь познакомимся с командой `rand`. Эта команда *генерирует равномерно распределенные случайные числа из диапазона $[0, 1]$* . Создайте вектор-строку из ста случайных чисел командой `a=rand(1,100);`. Не забывайте писать в конце команд точку с запятой, иначе вывод (особенно для больших массивов) может занять длительное время. Командой `x=rand(100,100)` можно создать массив из 100×100 случайных чисел. Аналогично ведет себя команда `x=rand(100)`.

Для **отображения графиков** в Matlab есть команда `plot`. Если ей задан один аргумент, она отображает по оси X номер элемента, а по Y — его значение. Если задать аргументами массивы *одинаковой* длины, первый используется как значения оси X, второй — Y. Постройте график `plot(a)`. Убедились, что a — массив совершенно случайных чисел?

Команда `mean(a)` вернет среднее арифметическое значение переменной a по столбцам. Вторым аргументом команды можно указать, по какому именно столбцу мы хотим усреднить массив a . Найдите среднее значение своего вектора. Отобразите на графике все значения и их среднее командой

```
plot([1:100], a, [1:100], mean(a))
```

Если точки двух графиков по оси X совпадают (как в предыдущем случае), можно укоротить запись:

```
plot([1:100],[a; mean(a)])
```

(команда `plot`, если задать ей в качестве аргумента матрицу, изображает разными цветами графики функций, соответствующих каждой строке матрицы).

Функция `normpdf(X, x, s)` позволяет построить график нормального распределения, соответствующий вектору координат X, для математического ожидания x и среднего квадратичного отклонения s . Наберите

```
x=[-70:30]; y=normpdf(x, -20, 20);
```

а затем — `plot(x,y)`. На графике вы увидите плотность вероятности нормально распределенной случайной величины с математическим ожиданием -20 и среднеквадратичным отклонением 20.

Сгенерируем синусоидальный сигнал на участке $[0, 2\pi]$ командами

```
x=[0:pi/50:2*pi]; y=sin(x);
```

Теперь добавим к сигналу гауссов белый шум с амплитудой 10 дБ относительно амплитуды сигнала:

```
y1=awgn(y,10,'measured'); plot(x, [y; y1])
```

Третий параметр (`measured`) обязателен, т.к. без него процесс добавления шума будет несколько иным (мощность сигнала будет считаться равной 0 дБ), можете проверить на синусоиде с амплитудой 10.

- ▶ Для генерации синусоидального сигнала $y_0 = A \sin(2\pi t/T)$ с амплитудной модуляцией по закону $y_1 = f(t)$ необходимо перемножить эти две функции: $y = y_0 \cdot y_1$. Промодулируем синусоиду с периодом $\pi/5$ пилообразным сигналом с периодом 10 на интервале $x \in [0, 20]$. Для генерирования «пилы» используется функция `sawtooth`. Если задать ей один аргумент (вектор x), период будет равен 2π , а сигнал будет изменяться в интервале $[-1, 1]$. Чтобы задать смещение максимума, равное $a \cdot 2\pi$, необходимо указать: `y=sawtooth(x, a)`. Таким образом, чтобы получить «пилу» с интервалом сигнала в $[0, 1]$ и периодом 10, необходимо дать команду `y1=0.5+sawtooth(x*2*pi/10)/2;`. Следовательно, получить наш сигнал можно командой

```
y=sin(x*10).*(0.5+sawtooth(x*pi/5)/2);
```

(не забудьте про точку перед знаком умножения между функциями, иначе получите ошибку, т.к. Matlab попытается перемножить два вектора–строки).

- ▶ Создадим теперь две синусоиды, сдвинутые на три единицы: `x=[0:0.05:20]; y=sin(x); y1=sin(x+3);`. Попробуем определить, на сколько единиц сдвинут первый сигнал относительно второго. Для этого воспользуемся корреляционной функцией. Запишем `Corr=xcorr(y,y1)`. Корреляционная функция в данном случае имеет вдвое большую ширину, чем исходная, т.к. она получается путем последовательного сдвига второй функции относительно первой. Поэтому построим график командой `plot([-20:0.05:20],Corr)`. Воспользовавшись функцией увеличения можно увидеть, что ближайший к нулю максимум соответствует сдвигу одной функции относительно другой. В нашем случае сигнал был периодическим, поэтому при сдвигах на величины, превышающие половину периода, возникает ошибка, кратная полупериоду. Это необходимо учитывать в расчетах.

1.3 Задания для самостоятельного выполнения

1. Найдите сумму, разность, произведение и частное матриц

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 9 & 8 & 7 \\ 5 & 3 & 1 \\ 0 & 2 & 6 \end{pmatrix}.$$

- Найдите определители исходных и получившихся матриц (команда `det(A)`).
2. Найдите значение почленного, матричного и скалярного произведений векторов $a = (2, 5, 7)$ и $b = (11, 13, 17)$. Скалярное произведение найдите двумя способами: путем перемножения векторов и при помощи функции `dot(a, b)`. Найдите векторное произведение $a \times b$ при помощи функции `cross(a, b)`.
3. Постройте график нормального распределения на интервале $[0, 100]$ с математическим ожиданием 50 и дисперсией 100.
4. Получите сигнал с амплитудной модуляцией (из примера). Добавьте к нему гауссов белый шум с SNR 100, 50, 10 и 1 дБ. Постройте отдельно графики всех полученных сигналов. Можно ли сделать какой-либо вывод о виде сигнала при SNR=1? Как вы думаете, можно ли восстановить из него исходный сигнал?
5. Для полученного сигнала найдите следующие характеристики: математическое ожидание (`mean`), среднее квадратичное отклонение (`std`), медиану (`median`) и моду (`mode`). Найдите аналогичные величины для разности между зашумленным и оригинальным сигналом. Сравните полученные величины с теоретическими.
6. Попробуйте определить сдвиг двух синусоид (из примера) при зашумлении:
- только одной с уровнем сигнал/шум 1 дБ;
 - обеих с уровнем SNR=1 дБ;
 - одной с уровнем SNR=0.1 дБ;
 - обеих с уровнем SNR=0.1 дБ.

Постройте один из сигналов с SNR=0.1 дБ. Можно ли определить его период? Можно ли определить период по автокорреляционной функции этого сигнала?

1.4 Контрольные вопросы

1. Что называют случайной величиной? Вероятностью? Плотностью вероятности? В чем отличие непрерывных и дискретных случайных величин?
2. Что такое функция распределения вероятности? Для чего используется нормировка функции распределения?
3. Дайте определения медианы, моды, математического ожидания, дисперсии и среднеквадратичного отклонения.
4. Можно ли считать равными среднее арифметическое и математическое ожидание? Какой закон описывает сходство этих величин?
5. Что называют центральными, начальными моментами? Чему равен центральный момент первого порядка?
6. Какие основные распределения случайных величин вы знаете? Какому распределению подчиняются большинство физически наблюдаемых случайных величин?
7. Для чего используется функция Лапласа?
8. Что такое ковариация, корреляция? Для чего при расчете коэффициента корреляции ковариацию делят на среднее геометрическое дисперсий случайных величин?
9. Для чего при исследовании двух сигналов применяется корреляционная функция?
10. Дайте определение белого шума. Почему шум такого вида не существует в природе?
11. Какая характеристика описывает чистоту сигнала? Объясните, от чего зависит коэффициент при расчете этой характеристики в логарифмической шкале.
12. Дайте определения скалярного, векторного и тензорного произведения двух векторов. Как в MatLab реализуются эти операции?

Компьютерная обработка

2 Теория физических измерений. Систематические и случайные погрешности

2.1 Краткие теоретические сведения

Развитие науки и техники неразрывно связано с точными измерениями, которые дают новую информацию об окружающем физическом мире. Эксперимент служит основной формой целенаправленного движения к познанию окружающего материального мира, то есть важнейшим инструментом науки. Естественным и наиболее простым способом количественного оценивания свойства является прямое сравнение двух вещей в определенном отношении (по степени проявления свойства). Для стандартизации измерений было разработано соглашение о единицах, используемых для измерений. В метрологии их называют **мерами**.

Результатом сравнения оцениваемой вещи с мерой является именованное число, называемое **значением величины**. Измерения одной и той же величины можно проводить как однократно, так и многократно.

Физические величины можно разделить на два типа: постоянные, изменяющиеся и случайные. К постоянным величинам относят инварианты, чье значение доподлинно известно, физические постоянные, а также величины, однозначно не изменяющие своего значения в процессе измерения. Изменяющиеся величины по определенному закону меняют со временем свое значение. Точное значение случайной величины определить невозможно, можно указать лишь ее дисперсию и среднее значение. Таким образом, эксперимент позволяет измерить с заданной точностью лишь постоянные и изменяющиеся величины. Однако, и в этом случае точность измерения будет ограничена случайными ошибками.

В отличие от классической механики, в квантовой механике многие физические величины являются связанными (например, импульс и координата), поэтому, чем больше точность измерения одной из связанных величин, тем меньше точность измерения другой величины (согласно соотношению неопределенностей).

Представление результатов физического измерения имеет немаловажное значение для понимания сути исследуемого процесса. *Результаты измерения можно представить в виде таблиц или графиков.* Наибольшую наглядность имеет графическое представление. При этом немаловажно правильно задать масштаб графика, направление осей и шкалы осей (линейную, экспоненциальную, логарифмическую или иную). Например, для отображения зависимости

активности радиоактивного изотопа от времени, удобно использовать линейную шкалу для активности препарата и логарифмическую — для времени.

Количественной характеристикой неоднозначности результата измерения является **погрешность**. Ее оценивают, исходя из всей информации, накопленной при подготовке и выполнении измерений. Окончательный результат измерения нельзя расценивать как «истинное значение» измеряемой физической величины, так как в этом нет смысла из-за наличия погрешности.

Погрешность может быть выражена в единицах измеряемой величины x — в таком случае она обозначается Δx и носит название **абсолютной погрешности**. Однако абсолютная погрешность не отражает в полной мере качества измерений. Действительно, абсолютная погрешность 1 мм при измерении размера комнаты перед оклеиванием обоями свидетельствует о высоком качестве измерения, но та же погрешность неприемлема при измерении, например, расстояний между атомами в металле.

Критерием качества измерения является безразмерное отношение абсолютной погрешности к окончательному результату измерения, $\delta x = \Delta x/x$, которое называют **относительной погрешностью** и используют как в абсолютном, так и в процентном выражении.

Выделяют следующие виды погрешностей:

промахи возникают вследствие неисправности измерительных приборов или ошибок в эксперименте, сделанных по невнимательности;

систематические погрешности как правило, неизвестны и могут быть учтены лишь при выполнении измерений несколькими приборами. Данная погрешность возникает вследствие значительной величины влияния прибора на измеряемую величину, либо же из-за пренебрежения некоторыми величинами на этапе моделирования эксперимента;

случайные погрешности носят случайный характер (чаще всего они имеют гауссово распределение) и возникают из-за незначительных (флуктуационных) изменений хода эксперимента.

Рассмотрим мысленный эксперимент по изменению физической величины x . Пусть в результате n измерений получен ряд значений x_1, \dots, x_n . Для получения действительного значения величины x необходимо, прежде всего выявить промахи, которые могут иметь вид неестественных значений результатов измерения. Далее учитывают систематические (приборные) погрешности, которые имеют вид поправок к результатам. Оставшиеся случайные погрешности обычно характеризуют средним квадратичным отклонением.

Приборные погрешности обычно указываются на лицевой панели прибора или в сопроводительной документации в виде **класса точности** — наибольшей

погрешности в процентном соотношении от предела измерений.

В случае, когда n имеет достаточно большую величину, погрешность результата $\sigma_{\bar{x}}$ уменьшается в \sqrt{n} раз по сравнению с погрешностью отдельного измерения σ , и выражается формулой

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n(n-1)}}.$$

За оценку погрешности окончательного результата многократного измерения примем величину Δx , задающую симметричный относительно \bar{x} интервал значений, называемый **доверительным интервалом**. Вероятность найти значение измеряемой величины в указанном интервале носит название **доверительной вероятности** α . Для нормального распределения существуют таблицы доверительных вероятностей в относительных единицах $\Delta x / \sigma$. Случайную величину обычно записывают в виде

$$x = \bar{x} \pm \Delta x, \quad \alpha = \alpha_0.$$

Идеальным является бесконечное число измерений, однако, обычно ограничиваются пятью-десятью замерами, что приводит к искажению оценки погрешности. В этом случае погрешность рассчитывают по формуле

$$\Delta x_{\text{случ}} = t(\alpha, n)\sigma_{\bar{x}},$$

где $t(\alpha, n)$ – табличные **коэффициенты Стьюдента**.

После измерения случайной погрешности результат записывают в виде $x = \bar{x} \pm \Delta x$, где

$$\Delta x = \sqrt{(\Delta x_{\text{случ}})^2 + \sigma_{\text{приб}}^2}.$$

Следует учесть при выполнении измерений, что *если результаты измерений не выходят за рамки приборной погрешности, можно уверенно считать $\Delta x = \sigma_{\text{приб}}$* . Повышение количества измерений в этом случае не способствует повышению точности эксперимента.

Для записи результата следует согласовывать точность измеренной величины и погрешности: порядок значащих цифр в них должен совпадать. Для определения количества значащих цифр погрешности следует учесть, что *относительная неточность оценивания величины σ составляет примерно $(n-1)^{-1/2}$* . Таким образом, точность оценки погрешности при выполнении десяти измерений составляет около 30%, что делает бессмысленным писать более одной значащей цифры (т.е. правильной будет запись в виде $x = 154 \pm 2$ и неправильной:

$x = 154.3 \pm 2.1$). Для более наглядного представления результата можно вынести за скобки общий множитель. Например, получив значение $\bar{x} = 3954.2 \cdot 10^3$ при погрешности $\Delta x = 126 \cdot 10^2$ на основании десяти измерений, следует записать результат в виде $x = (395 \pm 1) \cdot 10^4$.

Для определения погрешности в случае, когда *физическая величина определяется аналитически исходя из измерения нескольких физических величин*, необходимо придерживаться следующих правил.

1. Предельная **абсолютная погрешность** суммы или разности равна сумме абсолютных погрешностей. Пусть $Y = X_1 + X_2$, $X_1 = \bar{x}_1 \pm \Delta x_1$, $X_2 = \bar{x}_2 \pm \Delta x_2$, тогда $\Delta y = \Delta x_1 + \Delta x_2$. В общем виде:

$$\Delta(\pm \sum a_n) = \sum \Delta a_n.$$

2. Предельная **относительная погрешность** произведения или частного равна сумме относительных погрешностей (при пренебрежении величинами второго и бóльших порядков малости). Как следствие: относительная погрешность n -й степени случайной величины равна произведению относительной погрешности этой величины на n .

$$\prod (a_i \pm \Delta a_i) = \prod a_i \prod (1 \pm \delta a_i) \approx \prod a_i (1 \pm \sum \delta a_i),$$

$$(a[1 \pm \delta a])^n \approx a^n (1 \pm n\delta a).$$

3. В **сложных функциях** вида $y = f(x_1, \dots, x_n)$ можно оценить погрешность, воспользовавшись приближением:

$$\delta y \approx \frac{dy}{y} = \frac{df(x_1, \dots, x_n)}{f(x_1, \dots, x_n)}, \quad (2.1)$$

в котором следует заменить $dx_i/x_i = \delta x_i$ – относительная погрешность измерения величины x_i , $dx_i = \Delta x_i$ – абсолютная погрешность. Все слагаемые, возникающие при расчете по формуле (2.1) необходимо суммировать по абсолютной величине.

- Многие изменяющиеся физические величины имеют линейную зависимость. Оценить параметры такой зависимости можно графически, либо аналитически. Графическая оценка заключается в следующем алгоритме. На график в виде точек наносятся значения измеренных величин. Вокруг каждой точки строится прямоугольник со сторонами, соответствующими погрешностям измеренных величин. Далее между точками проводится прямая так, чтобы по возможности пересечь все прямоугольники. Если на протяжении всей прямой наблюдается примерно одинаковое количество точек над и под ней, и при этом точки располагаются хаотически (т.е. не образуют групп, последовательно расположенных

над или под прямой), данную зависимость можно интерпретировать как линейную. Из коэффициента наклона полученной прямой и точки пересечения ординаты определяются параметры зависимости $X = AY + B$.

Одним из наиболее распространенных приемов статистической обработки экспериментальных данных, относящихся к различным функциональным зависимостям физических величин друг от друга является **метод наименьших квадратов**. В своей простейшей форме он применим к линейной зависимости $y = ax + b$ и позволяет получить достоверные оценки ее параметров, a и b , а также оценить их погрешности.

Пусть проведено n парных измерений величин x и y . По экспериментальным данным необходимо найти оценки параметров a и b , а также оценки их дисперсий σ_a^2 и σ_b^2 . Предположим, что значения x_i известны без погрешностей, а величины y_i распределены нормально относительно величины \bar{y}_i с одинаковыми дисперсиями σ_y^2 . Математические ожидания \bar{y}_i удовлетворяют выражению $\bar{y}_i = ax_i + b$.

В этом случае критерий минимизации суммы квадратов отклонения $Y = \sum (y_i - \bar{y}_i)^2$ позволяет с достаточно большой вероятностью оценить величины a и b . Минимизация Y производится по обоим переменным, т.е. необходимо, чтобы выполнялись равенства: $\frac{\partial Y}{\partial a} = 0$ и $\frac{\partial Y}{\partial b} = 0$. Эти два условия образуют систему уравнений, из которых и находятся искомые неизвестные величины:

$$a = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} = \frac{\overline{xy} - \bar{x} \bar{y}}{\overline{x^2} - (\bar{x})^2}, \quad (2.2)$$

$$b = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{n \sum x_i^2 - (\sum x_i)^2} = \frac{\overline{x^2 y} - \bar{x} \overline{xy}}{\overline{x^2} - (\bar{x})^2}. \quad (2.3)$$

Соответствующие дисперсии равны:

$$\sigma^2 = \frac{n}{n-2} \left(\overline{y^2} - (\bar{y})^2 - a^2 [\overline{x^2} - (\bar{x})^2] \right), \quad \sigma_a^2 = \frac{\sigma^2}{n(\overline{x^2} - (\bar{x})^2)}, \quad \sigma_b^2 = \sigma_a^2 \bar{x}^2.$$

Найденные величины позволяют **аппроксимировать** (подыскать наиболее удовлетворяющую функциональную зависимость для данного набора данных) заданную таблично функцию $y(x)$.

- ▶ В случае функции двух и более переменных $z \equiv z(x, y) = ax + by + c$ коэффициенты a , b и c также можно отыскать методом наименьших квадратов. Единственным отличием в данном случае будет то, что система будет состоять из трех уравнений для трех неизвестных.

Теперь предположим, что у нас имеется нелинейная функция $y = f(x)$. Даже в этом случае возможно использовать метод наименьших квадратов, но формулы (2.2) и (2.3) могут значительно видоизмениться. Некоторые зависимости, однако, можно свести к линейным. Например, функция $y = e^{ax+b}$ после логарифмирования примет вид $\ln y = ax + b$. Заменяя переменную $y_1 = \ln y$ (т.е. прологарифмировав измеренные величины y_i) можно определить коэффициенты a и b по тем же формулам, что и для линейной зависимости. Однако, следует иметь в виду, что в этом случае дисперсии σ , σ_a и σ_b будут вычисляться немного иначе.

2.2 Примеры практических заданий

Случайная погрешность физического измерения имеет природу, аналогичную белому шуму, поэтому для начала рассмотрим простейшие методы очистки одномерных сигналов вида $y = y(t)$ от шумов.

Используем сигнал из задания 4 лабораторной работы №1, но создадим массив из десяти сигналов, зашумленных с одинаковым уровнем SNR:

```
x=[0:0.05:20];  
y=sin(x*10).*(0.5+sawtooth(x*pi/5)/2);  
for a=[1:10]  
y1(a,:)=awgn(y,1,'measured');  
end
```

Вид цикла `for` отличается от языков программирования вроде C: цикл поочередно перебирает все значения переменной `a`. Если бы мы заранее инициализировали ее массивом, можно было бы просто написать `for a`. Цикл `for` заканчивается командой `end` (ею же заканчиваются и многие другие циклы). Двоеточие в адресации `y(a,:)` означает, что мы выбираем **все** элементы по второй координате (т.е. приравнивание производится к целой строке). Еще одним отличием от языков программирования является динамическое расширение матриц: нет необходимости в начале работы с ней сообщать ее предельный размер.

Итак, мы получили массив `y1`, в строках которого содержатся зашумленные варианты одного и того же сигнала. Можете отобразить их все графически командой `plot(x,y1)`. Оценить зашумленность сигнала можно командой `plot(y,y1)`. Если бы сигналы в `y1` совпадали с `y`, мы увидели бы отрезок с коэффициентом наклона 1. Чем дальше форма полученной фигуры от такого отрезка, тем больше зашумленность сигнала.

Для восстановления сигнала из десяти измерений попробуем усреднить наборы сигналов и найти их медиану:

```
y_mean = mean(y1);
y_med = median(y1);
plot(x, [y; y_mean; y_med])
```

Как видите, оба восстановленных сигнала имеют примерно одинаковые величины и довольно близки к реальной функции (особенно на участках с большой амплитудой сигнала). Однако, как мы увидим впоследствии, если к сигналу добавлен шум типа «соль/перец», медианная фильтрация будет работать намного эффективнее фильтрации по среднему арифметическому.

► Теперь допустим, что мы имеем линейную зависимость $y = ax + b$, заданную таблично в виде $y(x)$. Для определения методом наименьших квадратов коэффициентов линейной (а также высших степеней) зависимости служит функция `polyfit(x,y,n)`. Она содержит три аргумента: x – вектор аргумента, y – вектор функции, n – степень аппроксимирующего полинома. Ее результат в простейшем случае представляет собой вектор коэффициентов (начиная со старшей степени). Если функцию вызвать как `[p,S] = polyfit(x,y,n)`, вектор p будет содержать коэффициенты, а в структуре S будут содержаться такие данные, как степени свободы (df) и норма отклонений данных от аппроксимирующей кривой (normr). Для восстановления полученной зависимости используется функция `polyval(p,x)`, где p – полученный функцией `polyfit` вектор коэффициентов, x – вектор аргумента. В таком виде функция возвращает вектор восстановленной функции. В виде `[y, delta] = polyval(p,x,S)` функция возвращает массив погрешностей (т.е. в каждой точке восстановленные значения функции можно представить в виде $y = y \pm delta$, т.е. оценить абсолютную погрешность восстановления можно при помощи команды `mean(delta)`.

Найдем коэффициенты модельной зависимости. Пусть $y = 7.15x + 4.22$. Построим вектора, соответствующие аргументу и функции:

```
x = [0:100]; y = 7.15*x + 4.22;
```

Зашумим сигнал для получения разброса точек y_i :

```
y1 = awgn(y,10,'measured');
```

Отобразим на экране оба ряда: `plot(x,y,x,y1,'.')` (запись `'.'` означает, что график будет отображаться точками). Как видите, разброс данных достаточно велик. Определим коэффициент корреляции: `corrcoef(x,y1)`. Он довольно близок к единице, следовательно, мы можем попытаться получить коэффициенты линейной зависимости и восстановить функцию:

```
[p,S] = polyfit(x,y1,1);          % коэффициенты a и b
[y2, delta] = polyval(p,x,S);    % восстановленный вектор
```

```
plot(x,y1, 'r', x, [y;y2])      % все три графика, зеленый - наш
mean(delta)                    % абсолютная ошибка
mean(delta)/mean(y)           % относительная ошибка
```

- Можно найти приближение методом наименьших квадратов и другим способом. Пусть Y – вектор–столбец значений функции, $A = (a, b)^T$ – вектор–столбец коэффициентов разложения. Тогда условие $y_i = ax_i + b$ можно представить в виде матричного произведения $Y = XA$. Второй столбец матрицы X целиком состоит из единиц, а в первом находится последовательность значений x_i . В этом случае нахождение коэффициентов сводится к решению системы линейных уравнений $y_i = ax_i + b$, дающему минимальную невязку. Такое решение находится при помощи операции левостороннего матричного деления: $X \backslash Y$. Решим предыдущий пример таким способом.

```
X = [x' ones(size(x'))]; % создаем матрицу аргумента
                                % (т.к. x и y1 - строки, транспонируем их)
A = X \ y1'                   % находим коэффициенты
                                % и отображаем их на экране
```

Функция `ones` создает матрицу с единичными элементами размером, соответствующим значению аргументов (или вектор с единичными элементами, если аргумент один). Функция `size` возвращает вектор, каждый элемент которого соответствует количеству элементов аргумента в данной размерности (а количество элементов вектора равно размерности матрицы–аргумента). Все, что следует за знаком `%` Matlab игнорирует, считая это комментарием.

Полученные значения должны быть примерно равны найденным предыдущим способом. Как мы увидим далее, такой способ нахождения корней аппроксимации пригоден не только для полиномиальных, но и для многих других функций.

- Попробуем создать квадратичную зависимость и аппроксимировать ее методом наименьших квадратов. Пусть зависимость на отрезке $[0, 100]$ имеет вид $y = 2.4x^2 - 0.87x + 2.13$. Создадим соответствующие массивы данных, добавим шум с SNR=20 дБ и отобразим оба сигнала на графике:

```
x=[1:100];
y=2.4*x.^2-0.87*x+2.13;
y1=awgn(y,20,'measured');
plot(x,[y;y1]);
```

Теперь создадим вектор коэффициентов аппроксимации полиномом второй степени восстановим функцию и отобразим на графике:

```
[p, S] = polyfit(x, y1, 2);
[y2, DELTA] = polyval(p, x, S);
plot(x, [y1;y2]);
```

Сравните функции y и y_2 : `plot(x, [y;y2])`. Отобразите на экране найденные коэффициенты: p . Рассчитайте среднее квадратичное отклонение аппроксимации (`mean(DELTA)`). Также рассчитайте относительную ошибку аппроксимации `mean(DELTA)/mean(y1)`.

► Однако, чаще всего функциональные зависимости имеют иные виды зависимости. Допустим, нам известно, что измеряемая величина изменяется по закону

$$y = a_0 + a_1 e^{-t} + a_2 t e^{-t}. \quad (2.4)$$

Для аппроксимации такой функцией можно представить уравнение (2.4) в матричном виде $Y = TA$, где T – функциональная матрица, у которой в первом столбце размещены единицы (соответствует умножению на a_0), во втором – функция e^{-t} , а в третьем – $t e^{-t}$. Найти коэффициенты A можно при помощи оператора левого деления: $A = T \backslash Y$. Для закрепления материала выполните следующий пример.

```
t = [0 0.3 0.8 1.1 1.6 2.3]';          % сразу вводим данные в столбцах
y = [0.6 0.67 1.01 1.35 1.47 1.25]';
T = [ones(size(t)) exp(-t) t.*exp(-t)];
A = T \ y
```

Теперь отобразим данные на графике:

```
x = (0:0.1:2.5)';
Y = [ones(size(x)) exp(-x) x.*exp(-x)] * A;
plot(x, Y, '- ', t, y, 'o')
```

Параметр `'-'` означает, что график построится в виде непрерывной линии (его можно опустить); `'o'` выведет график, на котором каждое значение будет обозначено кружками.

2.3 Задания для самостоятельного выполнения

1. На интервале `[0:0.05:20]` задайте функцию $y = \sin(x)$. Создайте три массива: на 10, 100 и 1000 зашумленных копий функции y с $\text{SNR} = -10$ дБ. Отобразите одну из этих функций. Найдите для каждого массива медиану. Постройте графики всех трех медиан. Для отображения трех разных

графиков на одном листе можно воспользоваться функцией `subplot(abc)`, где `a` – количество графиков по вертикали, `b` – количество графиков по горизонтали, `c` – номер текущего графика. Например, чтобы построить три графика один над другим, надо дать последовательность команд:

```
subplot(311); plot(график1)
subplot(312); plot(график2)
subplot(313); plot(график3)
```

Как видите, даже когда амплитуда шума в три раза превышает амплитуду сигнала, можно выделить сигнал чисто статистическими методами, однако, при этом необходимо провести огромное количество измерений.

- Найдите *обоими способами* коэффициенты a и b для таблично представленной зависимости $y(x)$, предполагая, что она имеет линейный вид. Найдите матрицу коэффициентов корреляции x и y (командой `corrcoef(x, y)`). Элемент матрицы с координатами (i, j) соответствует корреляции i -го и j -го столбцов. Таким образом, следует обратить внимание на элементы, расположенные на побочной диагонали. Если они близки к 1, можно говорить о линейной зависимости данных. Данные представлены в таблице:

x	1	2	3	4	5	6	7	8	9	10
y	7.7	13.7	22.0	23.1	23.7	36.7	35.6	47.8	50.2	52.1

($a = 5.1$, $b = 3.4$).

- Известно, что некоторая зависимость (см. таблицу ниже) имеет вид $y = ax \sin(x) - b \ln(x)$. Определите коэффициенты a и b и постройте данную кривую с более детальным отображением (на векторе `[1:0.05:10]`). Подсказка: сразу же задайте вектора x и y как столбцы; матрица X задается командой `X=[x.*sin(x) -log(x)]`.

x	1	2	3	4	5	6	7	8	9	10
y	-0.68	8.41	-23.0	-37.2	-73.2	-39.7	9.14	21.0	7.97	-72.5

($a = 7.72$, $b = 14.8$).

- Про моделируйте эксперимент измерения ста значений функции $y = 1.7x^3 + 3.4x^2 - 2.9x + 9.2$ и восстановления коэффициентов зависимости. Для этого создайте вектор аргумента `x=[1:100]`, получите по формуле соответствующий вектор функции `y_ideal`, а из него — зашумленный результат `y` с $\text{SNR}=25$ дБ.

Методом `polyfit` — `polyval` получите значения коэффициентов. Отобразите на графике точками исходные данные и непрерывной линией полученный аппроксимацией результат.

- Аналогично предыдущему заданию составьте модель эксперимента по измерению амплитуды напряжения в контуре, испытывающем колебания с

основной частотой $\Omega = 1000$ Гц и двумя гармониками $\Omega \pm \omega$, где $\omega = 74$ Гц. Известно, что суммарное колебание описывается приближенной формулой $U = a \sin(\Omega t) + b \sin(\omega t) - c \cos(\omega t)$. Создайте интервал времен $t = [0 : 0.06 : 120]$. Для получения идеальных значений U положите $a = 361$, $b = 117$, $c = 92$. Отношение сигнал/шум при получении зашумленного сигнала выберите равным 20 дБ.

Восстановите значения коэффициентов a , b и c .

2.4 Контрольные вопросы

1. Что такое постоянная, изменяющаяся и случайная величины? Чем отличается представление физических величин в классической и квантовой физике?
2. Какие виды погрешностей вы знаете? Что такое класс точности прибора?
3. Что включает в себя доверительный интервал? Что такое доверительная вероятность?
4. Для чего вводятся коэффициенты Стьюдента?
5. Как определить количество значащих цифр погрешности при записи результата измерения? Можно ли на основе 2000 измерений некоторой величины записать ее значение в виде $x = 1.592 \pm 0.002$? $x = (2.6423 \pm 0.5) \cdot 10^{10}$? $x = 110 \pm 4$?
6. Назовите правила для вычисления погрешности результата косвенных измерений, имеющего простую арифметическую зависимость. Как вычислить погрешность результата косвенных измерений, имеющего сложную зависимость? В чем особенность показательной и экспоненциальной функций, почему экспоненциальная функция имеет наибольшую чувствительность к погрешности аргумента?
7. Дайте определение и область применения метода наименьших квадратов. Для чего применяется аппроксимация рядов данных?
8. Какие функции в наибольшей степени удобны для аппроксимации?
9. Какие два способа линейной аппроксимации существуют в MatLab?
10. Как в MatLab выполнить аппроксимацию сложных функциональных зависимостей?

3 Теория оценок

3.1 Краткие теоретические сведения

Для вычисления вероятности попадания случайной величины, имеющей нормальное распределение, в заданный интервал $[a, b]$, используются функции Лапласа. Вероятность того, что случайная величина x лежит в интервале $[\bar{x} - \delta, \bar{x} + \delta]$ равна

$$P(|x - \bar{x}| < \delta) = 2\Phi(\delta/\sigma).$$

Если $\delta = 3\sigma$, вероятность попадания случайной величины в данный отрезок будет равна 0.9973%. Другими словами, лишь 0.27% наблюдаемых могут располагаться вне этого интервала. Этот вывод называют **правилом трех сигм**: *если случайная величина распределена нормально, то абсолютная величина ее отклонения от математического ожидания не превосходит утроенного среднего квадратичного отклонения.*

Согласно **теореме Ляпунова**, *случайная величина, являющаяся суммой большого числа взаимно независимых случайных величин, имеет нормальное распределение.* Так как результат физического измерения определяется огромным количеством случайных величин (давление, температура, магнитное поле Земли и т.п.), любой результат физического измерения имеет нормальное распределение.

Так как при измерении реальных (эмпирических) случайных величин ограничиваются конечным (и обычно небольшим) количеством измерений, вид распределения данной величины может отличаться от гауссова распределения. Пусть $x_i, i = \overline{1, n}$ — нормальные независимые случайные величины с $\langle x \rangle = 0$ и $\sigma_x = 1$. Тогда сумма квадратов этих величин $\chi^2 = \sum_{i=1}^n x_i^2$ распределена по закону «**хи квадрат**» с $k = n$ степенями свободы. Каждое линейное соотношение между этими величинами уменьшает количество степеней свободы распределения на единицу. Плотность этого распределения:

$$f(x) = \begin{cases} 0, & x \leq 0, \\ \frac{1}{2^{k/2}\Gamma(k/2)} e^{-x/2} x^{k/2-1}, & x > 0, \end{cases}$$

где $\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$ — гамма-функция, в частности, $\Gamma(n+1) = n!$. Отсюда видно, что с увеличением степеней свободы распределение «хи квадрат» приближается к нормальному распределению (что соответствует закону больших чисел).

Если z – нормальная случайная величина с $\bar{z} = 0$ и $\sigma_z = 1$, а v – независимая от z величина, распределенная по закону χ^2 с k степенями свободы, то величина $T = Z/\sqrt{V/k}$ имеет распределение, которое называют **t-распределением** или **распределением Стьюдента**. Данное распределение при увеличении k тоже приближается к нормальному распределению.

► Из-за того, что количество экспериментов (т.е. наблюдаемых величин) при измерении физической величины ограничено конечным (небольшим) числом, для определения как можно более точного значения математического ожидания данной величины используется *теория оценок*.

Несмещенной называют оценку величины x , совпадающую с \bar{x} , однако, действительные оценки являются смещенными. **Эффективной** называют оценку с наименьшей дисперсией. **Состоятельной** называют оценку, стремящуюся при увеличении количества наблюдаемых к \bar{x} .

Аналогично характеристикам случайных величин, для выборок вводят понятие группового и общего среднего, генеральной и выборочной дисперсии. В качестве оценки дисперсии выборки обычно принимают **исправленную дисперсию** $D = \frac{n}{n-1} D_{\text{выб}}$. Т.е. при расчете дисперсии суммы квадратов отклонений наблюдаемых от их среднего значения необходимо делить на число наблюдаемых без единицы.

Точностью оценки (или доверительным интервалом) δ называют длину полуинтервала, в котором с определенной вероятностью находится измеряемая величина. Соответствующую вероятность γ называют **надежностью** оценки. Т.о., получим соотношение:

$$P(|x - \bar{x}| < \delta) = \gamma.$$

Примем, что если случайная величина x распределена нормально, ее выборочная средняя $\langle x \rangle$ также распределена нормально. В этом случае $\overline{\langle x \rangle} = a$ и $\sigma_{\langle x \rangle} = \sigma/\sqrt{n}$, где a и σ – математическое ожидание и среднеквадратичное отклонение случайной величины x . В этом случае для надежности оценки получим:

$$\gamma \equiv P(\langle x \rangle - \delta < a < \langle x \rangle + \delta) = 2\Phi(\delta\sqrt{n}/\sigma).$$

Однако, чаще всего для данной наблюдаемой σ неизвестно. В этом случае можно оценить доверительный интервал при помощи распределения Стьюдента. В этом случае параметры распределения определяются так:

$$\gamma \equiv P\left(\left|\frac{\langle x \rangle - a}{S/\sqrt{n}}\right| < t_\gamma\right) = 2 \int_0^{t_\gamma} S(t, n) dt, \quad \delta = \frac{t_\gamma S}{\sqrt{n}},$$

3.2 Примеры практических заданий

где $S = \sqrt{D}$ – «исправленное» среднее квадратичное отклонение, параметр t_γ для данных n и γ можно найти в соответствующих таблицах.

Итак, оценку истинного значения измеряемой величины производят при помощи доверительного интервала. При этом параметр a считают равным выборочному среднему $\langle x \rangle$.

- Оценку среднее квадратичного отклонения σ случайной величины x производят при помощи критерия «хи квадрат». Вероятность $\gamma = P(|\sigma - S| < \delta)$ в этом случае равна

$$\gamma = \int_{\frac{\sqrt{n-1}}{1+\delta/S}}^{\frac{\sqrt{n-1}}{1-\delta/S}} R(\chi, n) d\chi \equiv \mathcal{X}(n, S).$$

Вычислив по выборке S и найдя по специальной таблице для S и заданной γ величину $q = \delta/S$, получим искомый доверительный интервал.

Для оценки других характеристик распределения используется **метод моментов**, согласно которому *начальные и центральные эмпирические моменты являются состоятельными оценками соответствующих начальных и центральных теоретических моментов.*

3.2 Примеры практических заданий

Найдем общую среднюю совокупности, состоящей из следующих трех групп:

Группа	I		II		III	
Значение признака	1	3	2	4	3	6
Частота признака	11	34	22	28	31	14
Объем выборки	11 + 34 = 45		22 + 28 = 50		31 + 14 = 45	

Для начала найдем групповые средние: $\langle x_1 \rangle$, $\langle x_2 \rangle$ и $\langle x_3 \rangle$:

```
>> x1 = (11*1 + 34*3)/45
```

```
x1 =  
2.5111
```

```
>> x2 = (22*2 + 28*4)/50
```

```
x2 =  
3.1200
```

```
>> x3 = (31*3 + 14*6)/45
```

```
x3 =  
3.9333
```

Теперь найдем общую среднюю по групповым средним:

```
>> X = (x1*45 + x2*50 + x3*45)/(45 + 50 + 45)
```

```
X =
    3.1857
```

Однако, при работе с большими массивами данных лучше использовать преимущества матричной алгебры:

```
>> xi = [1 3 2 4 3 6];
>> ni = [11 34 22 28 31 14 ] ;
>> N = sum(ni)
N =
    140
>> X = sum(xi.*ni/N)
X =
    3.1857
```

Найдем *генеральную дисперсию* и генеральное среднеквадратичное отклонение данной выборки:

```
>> D = sum(ni.*(xi-X).^2)/N
D =
    1.5369
>> sigma=sqrt(D)
sigma =
    1.2397
```

Кроме того, определить среднеквадратичное отклонение ряда x можно при помощи команды `std(x)`.

- Теперь рассмотрим ряд измерений некоторой физической величины x . Результаты серии измерений заданы таблицей (ν_i – частота соответствующего значения x_i):

x_i	31	28	34	26	35	30	34	32	40	20
ν_i	20	12	10	5	7	20	12	19	4	2

Известно, что некоторые результаты могут быть заведомо ошибочными. Нам необходимо оценить среднее значение данной величины, исключив ошибочные результаты. Составим массивы величины x и соответствующих частот n :

```
>> x=[ 31 28 34 26 35 30 34 32 40 20];
>> n=[ 20 12 10 5 7 20 12 19 4 2];
```

Отобразив данные на графике (`plot(x,n,'o')`) можно заметить, что действительно некоторые значения сильно отклоняются от положения, которое они занимали бы при нормальном распределении.

Найдем среднее значение величины x и ее среднеквадратичное отклонение:

```
>> X=sum(x.*n)/sum(n)
X =
    31.4144
>> sigma=sqrt(sum(n.*(x-X).^2)/sum(n))
sigma =
    3.1891
```

Определим границы доверительного интервала $[a, b]$ в пределах трех σ :

```
>> a=X-3*sigma
a =
    21.8470
>> b=X+3*sigma
b =
    40.9818
```

Теперь исключим из выборки значения, выходящие за пределы интервала. При помощи функции `find` можно найти индексы членов массива, удовлетворяющих заданному условию. Наберите `find(x<a)`. На экране отобразится индекс единственного элемента, выходящего за границы данного интервала: 10. Исключить его из дальнейших расчетов можно, приравняв соответствующую ему частоту нулю: `n(find(x<a))=0;`

Теперь повторим вычисление X и σ , a и b . Для того, чтобы вызвать из истории команд строку, начинающуюся с определенных символов, наберите один-два первых символа и нажмите клавишу «вверх». Таким образом можно быстро вызвать из истории команд нужную вам команду, не перебирая все промежуточные.

Теперь проверим, не влияет ли «подозрительное» значение $x = 40$ на точность измерения. Найдем медиану нашего ряда и оценим доверительный интервал по медиане. Для этого нам необходимо построить новый вектор `newx`, в котором значения величины x будут содержаться столько раз, какова их частота:

```
>> for a = [1:length(n)]
newx = [newx ones(1,n(a)).*x(a)];
end
med = median(newx)
med =
    31
>> b=med+3*sigma
b =
    39.4442
```

```
>> a=med-3*sigma
a =
    22.5558
```

Действительно, значение $x = 40$ выбивается из доверительного интервала. Приравняем его частоту к нулю, и найдем $\langle x \rangle$, близкое к истинному:

```
>> n(find(x>b))=0;
>> X=sum(x.*n)/sum(n)
X =
    31.3048
>> sigma=sqrt(sum(n.*(x-X).^2)/sum(n))
sigma =
    2.3345
>> a=X-3*sigma
a =
    24.3012
>> b=X+3*sigma
b =
    38.3083
>> find(x>b)
ans =
     9
>> find(x<a)
ans =
    10
```

Итак, кроме уже исключенных значений, все значения x_i удовлетворяют критерию «трех сигм», следовательно, можно записать ответ: $x = 31.3 \pm 2.3$.

- ▶ Теперь оценим значения величины $\langle x \rangle$ при помощи доверительного интервала с надежностью 95% при помощи распределения Стьюдента. Для этого в MATLAB существует функция `ttest`. В простейшем случае вида `h=ttest(x)` она возвращает вероятность отклонения гипотезы о нормальном распределении величины x с математическим ожиданием $\bar{x} = 0$. Проверка даст результат: 1. Действительно, математическое ожидание нашей величины далеко не равно нулю. Второй аргумент функции `ttest` задает предполагаемое математическое ожидание. Введите `h=ttest(x,X)`. Вы получите ответ: `h=0`. Т.е., можно принять гипотезу о гауссовой форме распределения величины x около ее среднего значения. Оценить 95%-й доверительный интервал величины x можно при помощи расширенного вывода функции `ttest` в форме `[h,p,ci]=ttest(x,X)`. В этом случае параметр `h` сообщает о степени ненадежности гипотезы, `p` равен

вероятности совпадения величины X с математическим ожиданием ряда x , ci сообщает границы 95%-го доверительного интервала. Определим доверительный интервал для нашего ряда без исключения заведомо ложных результатов и с их исключением:

```
>> [h,p,ci]=ttest(x,X)
```

```
h =
```

```
    0
```

```
p =
```

```
    0.8647
```

```
ci =
```

```
    27.0673    34.9327
```

```
>> [h,p,ci]=ttest(x(1:8),X)
```

```
h =
```

```
    0
```

```
p =
```

```
    0.9622
```

```
ci =
```

```
    28.6157    33.8843
```

Итак, в обоих случаях гипотеза о соответствии распределения величины x нормальному распределению принимается, однако, во втором случае вероятность определения математического ожидания \bar{x} выше, и доверительный интервал уже, что явно свидетельствует о большей надежности вычислений. *Середина доверительного интервала $\bar{x} = 31$ совпадает с медианой ряда, что лишний раз подтверждает большую степень надежности медианного усреднения по сравнению со средним арифметическим.*

- ▶ Matlab предоставляет огромный набор инструментальных средств. Однако, как вы уже могли заметить, при работе с большим количеством однообразных данных приходится много раз повторять одни и те же команды. Эту задачу можно упростить, создав **скрипт** (или **m-файл**). Скрипт представляет собой описание и реализацию пользовательской функции, которая вызывается из командной строки Matlab аналогично любой команде, однако может содержать значительное количество инструкций, облегчающих работу пользователя.

M-файл может содержать любые инструкции. Если он не начинается со слова **function**, выполняется все его содержимое, однако если вы хотите использовать какие-либо переменные в этом файле, их придется инициализировать заранее, причем их имена должны быть абсолютно такими же, как и в m-файле. Удобнее, однако, создать m-файл в виде функции, принимающей в качестве аргументов необходимые переменные и возвращающей определенные величины.

Заголовок файла функции имеет вид

```
%
% Комментарий, отображающийся при введении команды help имя_функции
%
function [возвращаемые величины] = имя_функции(входные, аргументы)
```

Далее следуют операторы, выполняемые в теле функции. Если после команды вы пропустите символ точки с запятой, ее вывод будет отображен на экране.

Итак, создадим m-файл, осуществляющий проверку выборки на корректность при помощи критерия «трех сигм». Зайдите в меню File \Rightarrow New \Rightarrow M-File. Появится окно редактора. Наберите содержимое файла и сохраните его в рабочей папке под именем `three_s.m`.

```
1 % three_s.m
2 % [ X sigma ] = three_s(x, n)
3 % Производит отбор выборки x с соответствующими частотами n
4 % при помощи критерия "трех сигм"
5 % результат: среднее значение X и его среднеквадратичное отклонение, sigma
6
7 function [ X sigma ] = three_s(x, n)
8 newx = []; % вспомогательный массив
9 Data = [x ; n]; % совмещенный массив данных
10 X = sum(x.*n)/sum(n); % среднее арифметическое
11 sigma = sqrt(sum(n.*(x-X).^2)/sum(n)); % среднеквадратичное отклонение
12 down = X-3*sigma; % нижняя граница доверительного интервала
13 up = X+3*sigma; % верхняя граница -//-
14 a=find(x < down); % a и b - массив координат, выходящих за границы
15 b=find(x > up);
16 while (length(a)>0) || (length(b)>0) % пока есть неверные значения
17 Data = Data(:, find(Data(1, find(Data(1,:) >= down)) <= up)); % выбрасываем их
18 x = Data(1,:);
19 n = Data(2,:);
20 X = sum(x.*n)/sum(n);
21 for a = [1:length(n)]
22 newx = [newx ones(1,n(a)).*x(a)];
23 end
24 X = median(newx);
25 sigma = sqrt(sum(n.*(x-X).^2)/sum(n));
26 down = X-3*sigma;
27 up = X+3*sigma;
28 a = find(x < down);
29 b = find(x > up);
30 end
```


Обратите внимание, что при запуске скрипта текущей папкой Matlab должна быть папка, в которую вы его сохранили.

Попробуйте вначале для того, чтобы лучше разобраться с принципом работы данного скрипта, не ставить точку с запятой в конце строк 17, 24, 25, 28 и 29. Запустить скрипт можно командой `[X sigma] = three_s(x,n)`.

- ▶ Зачастую физику-экспериментатору приходится проверять нулевую гипотезу о равенстве средних двух независимых наборов данных. Пусть в результате одного измерения некоторой физической величины x был получен ряд данных:

```
x1 = 47.78 36.40 35.66 8.93 40.42 54.16 51.76 44.32 46.19 50.75
```

Затем было произведено независимое измерение этой же физической величины при других условиях эксперимента. При этом был получен ряд:

```
x2 = 44.09 46.75 44.20 7.99 47.74 75.07 62.48 44.43 34.73 55.26
```

Требуется проверить нулевую гипотезу о равенстве математических ожиданий данных величин.

Для проверки данной гипотезы существует функция Matlab `ttest2`. Наберите `ttest2(x1,x2)`. В ответ вы получите: `ans=0`, т.е. гипотеза о равенстве математических ожиданий наших двух рядов отклонена на 95%-м уровне. Для определения доверительного интервала и вероятности равенства математических ожиданий воспользуемся расширенным выводом команды:

```
>> [h p ci] = ttest2(x1, x2)
h =
    0
p =
    0.5122
ci =
   -19.2074    9.9334
```

Таким образом, вероятность того, что математические ожидания выборок равны, составляет лишь $p = 51\%$, при этом доверительный интервал математического ожидания разности $x_1 - x_2$ достаточно широк: $c_i = [-19.2, 9.9]$, т.е. математические ожидания данных рядов могут различаться на 4.6 со среднеквадратичным отклонением $\sigma = 14.6$.

Большая ширина доверительного интервала говорит о том, что данные в рядах x_1 и x_2 получены с низкой надежностью. Однако, найдя медианы рядов x_1 , x_2 и совмещенного ряда $(x_1; x_2)$ можно попытаться с достаточно высокой степенью вероятности оценить математическое ожидание величины x .

3.3 Задания для самостоятельного выполнения

1. Некоторая совокупность состоит из трех групп: X_1 , X_2 , и X_3 . Группы имеют следующие значения:

```
X1 = 35.04 35.45 35.01 34.94 34.63 35.11 34.41 35.29 35.69
      34.69 35.36 35.53 34.30 34.36 35.23
X2 = 34.30 34.80 34.86 34.81 35.08 34.79 35.04 33.93 34.48
      34.41 33.74 34.60 34.00
X3 = 35.17 34.21 34.78 34.65 34.16 33.62 34.53 34.12 34.82
      34.77 35.29 34.81 34.28 34.72 34.12 34.55 34.53 34.55
```

Найдите: групповые средние (35,00, 34,53, 34,54), общее среднее (34,69), групповые дисперсии (0,19, 0,19, 0,16), генеральную дисперсию (0,22).

2. Усовершенствуйте скрипт `three_s.m` так, чтобы помимо основных вычислений на экране отображались среднее арифметическое значение массива с данными, а также 95%-й доверительный интервал по критерию Стьюдента.
3. Определите математическое ожидание и среднеквадратичное отклонение рядов данных x_1 и x_2 (пример со стр. 33) по-отдельности, а затем вместе. Сравните результаты со значениями, полученными при помощи функции `three_s` (вторым аргументом функции должен быть массив единиц: `ones(1,10)` для проверки рядов по-отдельности). Для совместной проверки рядов формат вызова функции следующий:

```
>> [mean, sigma] = three_s([x1 x2], ones(1,20))
```

Аналогично следует вызывать функцию `std` для определения среднеквадратичного отклонения рядов:

```
>> std(x1)
>> std(x2)
>> std([x1 x2])
```

4. Определите давление в цилиндре с газом, исходя из закона Менделеева–Клапейрона: $pV = mRT/\mu$, если известно, что масса газа $m = 2$ грамма, $\mu = 29$ г/моль, $R = 8.31$, а объем и температуру газа измеряли в течение минуты, получив следующие значения:

Величина	Значение									
V , л	2.27	2.27	2.26	2.25	2.26	2.27	2.29	2.28	2.25	2.28
T , К	399.4	399.1	399.3	396.8	399.5	400.2	400.6	403.0	399.2	401.3

Считайте, что за это время давление газа не успело сколь-нибудь значительно измениться. Определите погрешности измерения величин V и T . Считая, что остальные величины являются постоянными, определите кос-

венную погрешность измерения p .

Для удобства вычислений *создайте скрипт, позволяющий для заданного ряда данных получить математическое ожидание, среднеквадратичное отклонение и относительную ошибку.*

Запишите результат в виде $p = \bar{p} \pm \sigma_p$ ($p = 101 \pm 1$ кПа).

5. Для определения емкости C неизвестного конденсатора при помощи осциллографа исследовали затухающий импульс, возникающий при разрядке конденсатора через резистор $R = 3$ кОм. По показаниям осциллографа были записаны следующие значения тока:

$t, \text{ с}$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$I, \text{ А}$	1.00	0.72	0.52	0.37	0.26	0.19	0.14	0.10	0.07	0.04	0.03

Известно, что погрешность считывания значений тока с экрана осциллографа составляет $\sigma_I = 0.01$ А. Кроме того, известно что сопротивление резистора известно с точностью 3%. Из формулы $I = I_0 \exp(-t/[RC])$ определите погрешность измерения емкости конденсатора.

Методом наименьших квадратов определите значение емкости конденсатора, исходя из уравнения $t = -RC \ln I$ (составьте матрицу $X = -R \cdot \log(I)$) и найдите решение: $C = X \backslash t$ (97 мкФ). Запишите ответ в виде $C = \bar{C} \pm \sigma_C$.

Для увеличения точности эксперимента было проведено еще одно измерение, результаты которого несколько отличались от предыдущих:

$t, \text{ с}$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$I, \text{ А}$	1.00	0.75	0.56	0.41	0.30	0.23	0.17	0.12	0.10	0.07	0.05

Проверьте нулевую гипотезу о равенстве средних в обоих опытах. Определите величину емкости во втором случае (112 мкФ).

Столь большое различие емкостей, полученных в результате двух независимых экспериментов, заставило предположить, что в результате длительной эксплуатации резистор R нагрелся, что вызвало увеличение его сопротивления. Считая емкость конденсатора прежней, определите сопротивление резистора во втором случае (3.5 кОм).

3.4 Контрольные вопросы

1. Для чего применяется правило «трех сигм»?
2. Согласно какой теореме мы имеем право считать подавляющее большинство физических наблюдаемых распределенными нормально?
3. Что описывает распределение «хи квадрат»?
4. Какие величины подчиняются распределению Стьюдента?
5. Назовите основные виды оценок наблюдаемых.

6. Для чего вводят «исправленную» дисперсию?
7. Что такое точность, надежность оценки? Как можно оценить точность оценки для заданной надежности?
8. Почему при расчете математического ожидания и среднеквадратичного отклонения набора случайных величин следует оценивать оба этих параметра? В каком случае эту оценку можно не производить?
9. Сформулируйте определение метода моментов.
10. Назовите два способа вычисления среднего значения и дисперсии случайной величины в MatLab.
11. Что такое весовые коэффициенты и для чего необходимо их использовать при вычислении общей средней, генеральной дисперсии?
12. Какую характеристику случайной величины следует использовать для определения границ доверительного интервала при помощи критерия «трех сигм»? Аргументируйте ответ.
13. В каких случаях для определения центра доверительного интервала можно использовать моду? Медиану? Среднее арифметическое?
14. Какая функция MatLab позволяет оценить доверительный интервал математического ожидания случайной величины?
15. Почему критерий проверки значений наблюдаемой на промахи носит название критерия «трех сигм»? Что особенного в интервале $\bar{x} \pm 3\sigma_x$?

4 Системы линейных уравнений. Степенные уравнения. Дифференциальные уравнения

4.1 Краткие теоретические сведения

Система линейных уравнений для n неизвестных имеет вид:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2; \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases} \quad (4.1)$$

Если представить коэффициенты уравнений в виде квадратной матрицы \mathbf{A} , а свободные члены и неизвестные в виде векторов-столбцов \mathbf{b} и \mathbf{x} , то выражение (4.1) примет вид $\mathbf{Ax} = \mathbf{b}$.

Если существует вектор-столбец \mathbf{x} , обращающий выражение $\mathbf{Ax} = \mathbf{b}$ в тождество, говорят, что \mathbf{x} является **решением** данной системы уравнений. Для того, чтобы решение системы существовало и было единственным, необходимо чтобы определитель матрицы \mathbf{A} не был равен нулю: $|\mathbf{A}| \neq 0$.

Численные методы нахождения решения системы уравнений можно разделить на два класса: *приближенные* и *точные*. Приближенные методы позволяют найти вектор \mathbf{x} , дающий *наименьшую невязку* $\delta = \mathbf{Ax} - \mathbf{b}$. Точные же методы позволяют отыскать единственный возможный вектор решений, обращающий невязку в ноль.

Самым известным точным методом решения систем уравнений является **метод Гаусса**. Прямой ход метода Гаусса приводит систему к верхней треугольной форме, обратный ход позволяет «снизу вверх» найти решения системы: из последнего уравнения корень x_n , из предпоследнего — x_{n-1} и т.д. Реализация прямого хода метода Гаусса требует $N \propto 2n^3/3$, а обратного — $N \propto n^2$ арифметических операций.

Простейшим приближенным методом решения систем уравнений является **метод простой итерации**. Система уравнений в этом методе преобразуется к виду $\mathbf{x} = \mathbf{Vx} + \mathbf{c}$ и ее решение находится как предел последовательности $\mathbf{x}^{n+1} = \mathbf{Vx}^n + \mathbf{c}$.

В MATLAB, как и во многих других математических пакетах, используются особенности матричной алгебры для нахождения решения систем уравнений. Так, решение системы уравнений $\mathbf{Ax} = \mathbf{b}$ получается при помощи «левого» де-

ления матриц: $\mathbf{x} = \mathbf{A} \setminus \mathbf{b}$; решение системы $\mathbf{x}\mathbf{A} = \mathbf{b}$ находится при помощи прямого деления: $\mathbf{x} = \mathbf{b}/\mathbf{A}$.

Матрица коэффициентов \mathbf{A} не обязательно должна быть квадратной. Пусть эта матрица содержит m строк и n столбцов. Тогда, в зависимости от соотношений между m и n получим три случая:

- $m = n$ — квадратная матрица, возможно существование точного решения;
- $m < n$ — недоопределенная система, решение возможно лишь в общем виде с по крайней мере $n - m$ свободных коэффициентов;
- $m > n$ — переопределенная система, приближенное решение которой находится при помощи метода наименьших квадратов (в случае линейной зависимости строк данной системы может существовать и точное решение).

В случае, когда матрица \mathbf{A} является *вырожденной* (т.е. имеет линейно зависимые строки или столбцы: $|\mathbf{A}| = 0$), решение системы (4.1) либо не существует, либо не является единственным. В случае существования точного решения системы, его значения можно получить, умножив *псевдообратную матрицу* коэффициентов на вектор свободных членов: $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. Псевдообратная матрица вычисляется в MATLAB командой `pinv`. Проверить точность решения можно, подставив найденные корни в изначальное уравнение.

Определить, имеет ли система точные решения можно, приведя к верхней треугольной форме расширенную матрицу $(\mathbf{A}|\mathbf{b})$ (команда `rref([A b])`). Если нижний ряд такой матрицы будет нулевым, за исключением последнего члена, система не будет иметь точных решений.

Переопределенные системы уравнений часто используются для аппроксимации функций методом наименьших квадратов. Имея функциональную зависимость вида $y = f(x; C_1, \dots, C_n)$, где C_i — неизвестные постоянные величины (множители), методом наименьших квадратов можно восстановить их значения. Для этого необходимо составить функциональную матрицу F , каждый столбец которой отражает функциональную связь соответствующего слагаемого. В результате мы получим систему уравнений $\mathbf{y} = \mathbf{F}\mathbf{C}$. При достаточно большом количестве строк данной системы методом наименьших квадратов можно найти коэффициенты: $\mathbf{C} = \mathbf{F} \setminus \mathbf{y}$.

Для недоопределенных систем возможно найти основное решение \mathbf{p} , содержащее по крайней мере m ненулевых компонент. Общее решение можно найти из выражения $\mathbf{x} = \mathbf{p} + \mathbf{Z}\mathbf{q}$, где \mathbf{Z} — базис нулевого подпространства для матрицы \mathbf{A} , \mathbf{q} — произвольный вектор длины n .

- **Степенное уравнение** имеет вид $p_n(x) = 0$, где $p_n(x)$ — полином n -й степени вида $p_n(x) = \sum_{i=0}^n C_n x^n$. Для решения таких уравнений применяются как численные, так и точные методы. Так, известны точные аналитические формы для нахождения корней полиномов до третьей степени включительно.

Численные методы позволяют найти корни полиномов, проводя последовательные итерации на сужающейся сетке.

В MATLAB полиномы задаются вектором коэффициентов $\mathbf{C} = (C_1, \dots, C_n)$. Корни уравнения $p_n(x) = 0$ получим, выполнив $\mathbf{x}=\text{roots}(\mathbf{p})$. Проверить точность найденных корней можно, восстановив коэффициенты полинома по полученным значениям корней: $\text{poly}(\mathbf{x})$. Для восстановления значений полинома на сетке \mathbf{x} служит команда $\text{polyval}(\mathbf{p}, \mathbf{x})$.

Для решения уравнения $p_n(x) = 0$ в случае, когда оно задано численно в виде набора точек (x_i, y_i) , необходимо вначале определить коэффициенты полинома методом наименьших квадратов. Для этих целей служит команда $\text{polifit}(\mathbf{x}, \mathbf{y}, \mathbf{n})$. Она возвращает вектор коэффициентов полинома p_n заданной степени n . Далее поиск корней осуществляется стандартными способами.

► **Обыкновенные дифференциальные уравнения (ОДУ)** порядка n задаются в виде функции $f(x, y, y', \dots, y^{(n)}) = 0$.

Наиболее простым является решение ОДУ первого порядка вида $y' = f(x)$. ОДУ высших порядков при решении численными методами обычно приводятся к системе ОДУ первого порядка.

Нахождение (аналитически) решений ОДУ первого порядка обычно сводится к разделению переменных и последующему интегрированию:

$$y' = f(x, y) \Rightarrow \varphi(y) dy = \psi(x) dx \Rightarrow y = y_0 + \int_0^x \psi(x) dx.$$

Многие физические задачи приводят к ОДУ второго порядка вида

$$Ay'' + By' + Cy + Dx = 0.$$

В случае, когда $D \equiv 0$, а множители A , B и C являются константами, данное ОДУ называется **однородным** с постоянными множителями. Его решение ищется в виде $y = \mathfrak{C}_1 \exp(k_1 x) + \mathfrak{C}_2 \exp(k_2 x)$, где k_1 и k_2 – корни *характеристического уравнения* $Ak^2 + Bk + C = 0$. Если $D \equiv \text{const} \neq 0$, уравнение называется **неоднородным**. Его решение есть сумма общего решения соответствующего однородного уравнения и одного из частных решений неоднородного уравнения.

Методы аналитического решения ОДУ специфичны для каждого класса ОДУ.

Дифференциальные уравнения в частных производных (ЧДУ) для функции $y = y(x_1, x_2, \dots, x_n)$ имеют вид

$$f(y, x_1, \dots, x_n; \frac{\partial y}{\partial x_1}, \dots; \frac{\partial^2 y}{\partial x_1^2}, \dots; \dots; \frac{\partial^m y}{\partial x_1^m}, \dots) = 0.$$

Однако, наиболее часто встречаются ЧДУ первого порядка для функции двух переменных $z = z(x, y)$ вида

$$f\left(z, x, y, \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right) = 0.$$

Нелинейные дифференциальные уравнения содержат некоторые производные функции y не как простые множители, а как аргументы функций (чаще всего — степенных), например: $(y'')^3 - \sin y' = \operatorname{tg}(xy)$. Обычные физические задачи никогда не приводят к таким уравнениям, однако, и их решения вполне можно найти при помощи численных методов.

4.2 Примеры практических заданий

Решим систему уравнений

$$\begin{cases} -x_1 + x_2 + 2x_3 = 10; \\ 3x_1 - x_2 + x_3 = -20; \\ -x_1 + 3x_2 + 4x_3 = 40. \end{cases}$$

Представим ее в виде $\mathbf{Ax} = \mathbf{b}$. Инициализируем постоянные:

```
>> A=[-1 1 2; 3 -1 1; -1 3 4];
>> b=[10; -20; 40];
```

Нам необходимо проверить на вырожденность матрицу \mathbf{A} :

```
>> det(A)
ans =
    10
```

Теперь решить данную систему можно несколькими способами.

1. Через обратную матрицу.

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}, \quad \Rightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}.$$

В MATLAB'е это примет вид:

```
>> x = inv(A)*b
x =
    1.0000
   19.0000
   -4.0000
```

Проверим решение:


```
>> A*x
ans =
    10.0000
   -20.0000
    40.0000
```

2. Метод Гаусса. Приведем к верхней треугольной форме расширенную матрицу ($\mathbf{A} : \mathbf{b}$):

```
>> rref([A b])
ans =
     1     0     0     1
     0     1     0    19
     0     0     1    -4
```

Слева мы получили единичную матрицу, что значительно упрощает вычисления. Однако, если бы матрица не имела нулей в правом верхнем углу, мы все равно могли бы найти корни системы (обратный ход метода Гаусса).

3. Автоматический метод Гаусса. В данном случае необходимо лишь воспользоваться уже известным вам оператором «левого» (или обратного) деления:

```
>> x = A\b
x =
     1.0000
    19.0000
    -4.0000
```

Далее мы будем использовать именно этот способ решения линейных систем уравнений.

- Теперь решим систему уравнений, заданную вырожденной матрицей (естественно, в этом случае мы не претендуем на единственность решения).

```
>> A = [ 1     3     7
        -1    4     4
         1    10    18 ];
>> b = [5; 2; 12];
>> det(A)
ans =
     0
```

Так как определитель матрицы коэффициентов равен нулю, невозможно найти обратную матрицу. Однако, можно воспользоваться способом решения через *псевдообратную матрицу*:

```
>> pinv(A)*b
ans =
    0.3850
   -0.1103
    0.7066
>> % Проверим
>> A*ans
ans =
    5.0000
    2.0000
   12.0000
```

Однако, изменим вектор **b**:

```
>> b = [3;6;0];
>> pinv(A)*b
ans =
   -1.0892
    1.2512
   -0.5235
>> A*ans
ans =
   -1.0000
    4.0000
    2.0000
```

В этом случае решение не будет точным (точнее, оно вообще не является решением данной системы). Проверим, возможно ли найти общее решение данной системы уравнения, приведя к верхней треугольной форме расширенную матрицу (**A : b**):

```
>> rref([A b])
ans =
    1.0000         0    2.2857         0
         0    1.0000    1.5714         0
         0         0         0    1.0000
```

Последняя строка содержит ненулевой элемент лишь в столбце свободных членов, что однозначно свидетельствует об отсутствии решений данной системы уравнений.

- Найдём решение уравнения $2x^2 - 4x + 5 = 0$. Для этого необходимо инициализировать полином набором коэффициентов и найти корни командой `roots`.

```
>> p = [2 -4 5];  
>> roots(p)  
ans =  
    1.0000 + 1.2247i  
    1.0000 - 1.2247i
```

Итак, корни нашего уравнения: $x = 1 \pm 1.2247i$. Точность вычислений MATLAB можно задать явно командой `format`. Для отображения результата в виде рациональных дробей можно указать следующее.

```
>> format rat  
>> ans  
ans =  
     1          + 1079/881i  
     1          - 1079/881i
```

Вернуться к прежнему виду результатов можно командой `format short`.

Теперь найдем корни полинома $p(x) = x^4 + 2x^3 - 3x^2 + 4x + 5$ и получим его график.

```
>> p = [1 2 -3 4 5];  
>> roots(p)  
ans =  
   -3.1825  
    0.9556 + 1.1148i  
    0.9556 - 1.1148i  
   -0.7287  
>> x=[-4:.05:2]; y=polyval(p,x);  
>> plot(x,y)
```

Нарисуем ось X:

```
>> hold on  
>> plot([-4 2], [0 0], 'k')
```

Команда `hold on` позволяет «дорисовать» что-либо на уже имеющемся графике. Буква `'k'` в параметре означает рисование черным цветом. Отключить вывод на один и тот же график можно командой `hold off`.

В MATLAB есть возможность аналитического решения уравнений. Для этого используются интегрированные функции Maple. Найдем корни квадратного уравнения $ax^2 + bx + c = 0$ в общей форме. Для этого нам потребуется ввести пять *символьных* переменных и использовать функцию `solve` для решения уравнения.

```

>> syms a b c x y
>> y=a*x^2+b*x+c;
>> solve(y)
ans =
-1/2*(b-(b^2-4*a*c)^(1/2))/a
-1/2*(b+(b^2-4*a*c)^(1/2))/a
>> pretty(ans)

```

$$\begin{bmatrix} \frac{b - (b^2 - 4ac)^{1/2}}{a} \\ \frac{b + (b^2 - 4ac)^{1/2}}{a} \end{bmatrix}$$

Функция `pretty` выводит результаты в более привычном для глаза виде. Функцию `solve` можно использовать в таких простых случаях и без введения символьных переменных: просто наберите `solve('a*x^2+b*x+c')`. По умолчанию функция `solve(arg)` решает символьное уравнение $arg = 0$ относительно переменной x . Если ее не будет, уравнение решается относительно y или любой другой переменной, которую функция встретит первой. Для указания конкретной переменной, относительно которой мы хотим решить уравнение, вторым аргументом функции можно задать эту переменную.

```

>> solve(y,a)
ans =
-(b*x+c)/x^2
>> solve(y,b)
ans =
-(a*x^2+c)/x
>> solve(y,c)
ans =
-a*x^2-b*x

```

Если переменная не была задана как символьная, необходимо заключать ее (и выражение) в одинарные кавычки.

- ▶ Для решения дифференциальных уравнений в МАТЛАВ существует огромное количество функций, базирующихся на различных численных методах. Наиболее быстрым (но и наименее точным) является метод Рунге–Кутты. Решим

этим методом дифференциальное уравнение ван дер Поля $y'' + \mu(1 - y^2)y' + y = 0$, $\mu > 0$.

Для начала перепишем это уравнение с заменой $y_1 = y$, $y_2 = y_1'$: $y_2' = \mu(1 - y_1^2)y_2 - y_1$. Для простоты примем $\mu = 1$. Введем функцию, описывающую наше уравнение (ее необходимо ввести как новый m-файл и сохранить под именем vdp1.m):

```
% Функция для инициализации решения уравнения ван дер Поля с мю=1
function dydt = vdp1(t,y)
dydt = [y(2); (1-y(1)^2)*y(2)-y(1)];
```

Теперь найдем решение уравнения и отобразим графики функции y и ее первой производной:

```
>> [t, y] = ode45(@vdp1, [0 20], [2; 0]);
>> plot(t, y(:,1), '- ', t, y(:,2), '--')
```

Функция `ode45` в качестве первого параметра требует имя функции, в которой описано дифференциальное уравнение; второй параметр — интервал, в котором изменяется аргумент искомой функции; третий аргумент — начальные условия для функции и ее производной. Возвращаемое значение y содержит два столбца: в первом находится искомая функция, а во втором — ее первая производная.

Итак, для численного решения дифференциального уравнения в MATLAB необходимо сначала представить это уравнение в виде линейной системы

$$\begin{cases} y_1' = f_1(x, y_1, \dots, y_n), \\ y_2' = f_2(x, y_1, \dots, y_n), \\ \dots \\ y_n' = f_n(x, y_1, \dots, y_n). \end{cases}$$

Затем функции f_1, \dots, f_n следует определить как строки специальной функции, которая будет играть роль первого параметра функции, решающей данное уравнение.

Теперь найдем аналитическое решение дифференциального уравнения $y'' + ky = 0$. Для этого обратимся к функциям Maple. Сначала найдем общий вид решения, а затем — при $y(0) = 0$, $y'(0) = 5$.

```
>> dsolve('D2y+k*y=0', 'x')
ans =
C1*sin(k^(1/2)*x)+C2*cos(k^(1/2)*x)
>> dsolve('D2y+k*y=0', 'y(0)=0', 'Dy(0)=5', 'x')
ans =
5/k^(1/2)*sin(k^(1/2)*x)
```

Итак, уравнение колебаний из положения равновесия с начальной скоростью 5 имеет вид $y = \frac{5}{\sqrt{k}} \sin(x\sqrt{k})$.

Теперь найдем решение простейшего уравнения затухающих колебаний $y'' + y' + y = 0$:

```
>> dsolve('D2y+Dy+y=0', 'y(0)=0', 'Dy(0)=5', 'x')
ans =
10/3*3^(1/2)*exp(-1/2*x)*sin(1/2*3^(1/2)*x)
```

Итак, в данном случае решение: $y = \frac{10}{\sqrt{3}} \exp(-x/2) \sin(x\sqrt{3}/2)$.

Построим график данной функции:

```
>> x=[0:.1:30];
>> y=10/3*3^(1/2)*exp(-1/2*x).*sin(1/2*3^(1/2)*x);
>> plot(x,y)
```

Обратите внимание на точку перед знаком умножения на синус. Без нее МАТЛАВ попытался бы выполнить матричное умножение. *Если результаты символьных вычислений не имеют таких слабых мест, их можно использовать напрямую при помощи команды eval.* Например, вместо второй строчки можно было бы записать:

```
>> y=eval(ans);
```

Однако, можно сделать и по-другому: команда `subs` вычисляет данные в символьном виде:

```
>> x=[0:.1:30];
>> syms z;
>> z = dsolve('D2y+Dy+y=0', 'y(0)=0', 'Dy(0)=5', 'x');
>> y=subs(z);
>> plot(x,y)
```

Аналогично, символьные вычисления помогают решать системы уравнений (в том числе и дифференциальных). Так, для системы $\begin{cases} x + 2y = 2, \\ 7x - y = 3; \end{cases}$ получим:

```
>> syms x y
>> [x y] = solve('x+2*y-2', '7*x-y+3')
x =
-4/15
y =
17/15
```

А для системы дифуравнений $\begin{cases} f'(t) = 3f(t) + 4g(t), \\ g'(t) = -4f(t) + 3g(t); \end{cases}$ получим:

```
>> syms f g
>> [f,g] = dsolve('Df=3*f+4*g, Dg=-4*f+3*g', 'f(0) = 0, g(0) = 1')
f =
exp(3*t)*sin(4*t)
g =
exp(3*t)*cos(4*t)
```

4.3 Задания для самостоятельного выполнения

1. Решите систему уравнений

$$\begin{cases} x_1 + x_2/2 + x_3/3 = 1; \\ x_1/2 + x_2/3 + x_3/4 = 0; \\ x_1/3 + x_2/4 + x_3/5 = 0. \end{cases}$$

Обратите внимание, что определитель матрицы коэффициентов $\det(A) = 4.6296e-04$. Такие системы называются **плохо обусловленными**. Их решения сильно осциллируют при малейших изменениях коэффициентов матрицы.

Найдите решения, записав в матрице коэффициенты в виде рациональных дробей вроде $1/3$, $1/4$, $1/5$ (9, -36, 30), затем измените $1/3$ на 0.333 (9.6707, -39.5082, 33.2836), после — на 0.33 (55.5556, -277.7778, 255.5556), а затем — на 0.3 (0.5556, 5.5556, -7.7778).

2. Решите систему уравнений

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 1; \\ 2x_1 - x_2 + 4x_3 = 2; \\ x_1 - 3x_2 + x_3 = 3. \end{cases}$$

(1, π, 0)

3. Решите систему уравнений

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 1; \\ 2x_1 - x_2 + 4x_3 = 2; \\ x_1 - 3x_2 + 2x_3 = 3. \end{cases}$$

(-3.4, -0.8, 2)

4. Решите систему уравнений

$$\begin{cases} x_1 + x_2 + x_3 = 0; \\ 2x_1 + 4x_2 - 4x_3 = 0; \\ x_1 + 1/3x_2 + 3x_3 = 0. \end{cases}$$

$(a, -a/4, -3a/4)$.

5. Решите уравнение $x^7 - 2x^5 + 3x^3 - 4x = 0$. ($0, \pm 1.2848, 0.9304 \pm 0.8313i, -0.9304 \pm 0.8313i$).

6. Решите (аналитически) уравнение $x^3 + ax^2 + bx + c = 0$.

Найдите решение этого уравнения при $a = 1, b = 2, c = 3$ двумя способами: при помощи функции `subs` и функции `roots` ($0.1378 + 1.5273i, 0.1378 - 1.5273i, -1.2757$).

7. Найдите решение уравнения ван дер Поля при $\mu = 5$. Найдите значение функции $y(x)$ при $x = 4$ (1.2350).

8. Постройте график решения задачи Коши методом Рунге–Кутты на интервале $[0, 1]$ для уравнения $y' = x^3 \sin y + 1$ при $y(0) = 0$.

9. Найдите аналитическое решение уравнения $\frac{dy}{dx} = -2xy$, удовлетворяющее начальному условию $y(0) = 1$ ($y = \exp(-x^2)$).

10. Найдите аналитическое решение уравнения

$$(x - 2\sqrt{xy}) \frac{dy}{dx} - y = 0, \quad (x > 0, y > 0)$$

при $y(1) = 10$. (`sin x`, для упрощения выражения после команды `dsolve...` используйте `simplify(ans)`, пока ответ не примет наиболее простую форму). В данном случае `RootOf(_Z^2-1)` означает «корни уравнения $_Z^2 - 1 = 0$, т.е. 1 и -1 . Данное выражение часто встречается, когда Maple не может выразить корни явно.

11. Найдите решение уравнения

$$(1 - x^2) \frac{d^2y}{dx^2} - 2x \frac{dy}{dx} + 12y = 0$$

с начальными условиями $y(0) = 0, y'(0) = -3/2$ ($-3/2x + 5/2x^3$). Найдите значение функции при $x = 1$ (1).

4.4 Контрольные вопросы

1. Почему систему линейных уравнений удобно представлять в матричной форме? Что такое расширенная матрица, соответствующая данной системе?

2. Дайте определение критерия существования и единственности решения системы линейных уравнений.
3. В чем достоинства и недостатки метода Гаусса, почему он не применяется при решении систем даже со сравнительно небольшим количеством неизвестных ($\sim 10^2$)?
4. В каком случае невозможно получить точное решение системы линейных уравнений? Что такое плохо обусловленные системы?
5. В каких случаях система линейных уравнений имеет единственное решение? Бесконечное множество решений? Ни одного решения? Приближенное решение?
6. Чем псевдообратная матрица отличается от обратной?
7. Для каких полиномиальных уравнений существуют точные аналитические решения? Можно ли найти аналитическую форму решения уравнения более высоких порядков?
8. Какое максимальное количество операций необходимо провести для поиска корня степенного уравнения на отрезке $[0, 1]$ с точностью 10^{-9} , используя метод последовательного перебора значений аргумента с заданным шагом? Сколько (максимум) операций необходимо для той же операции при использовании метода деления отрезка пополам?
9. Можно ли сказать, что абсолютно все ОДУ первого порядка имеют элементарные аналитические решения? Какие условия накладываются на подынтегральную функцию при решении ОДУ?
10. Назовите основные методы численного интегрирования и дифференцирования. По какому минимальному количеству точек можно найти десятую производную в центре ряда данных?
11. Почему аналитическое решение однородного ОДУ с постоянными коэффициентами сводится к решению характеристического уравнения?
12. В чем состоит особенность решения ЧДУ? В каких случаях решения ЧДУ имеют наиболее простые формы?
13. Приведите пример физической задачи, сводящейся к решению ОДУ первого, второго, третьего порядка. Какие физические задачи могут привести к решению ЧДУ? Есть ли какие-то отличия дифференцирования функций по пространственным или временным переменным?
14. Объясните, как можно в MatLab получить аналитическую форму решения степенного уравнения четвертой степени.
15. Как, решив дифференциальное уравнение при помощи метода ван дер Поля, можно перейти к равномерной сетке по оси x ?
16. Позволяют ли средства MatLab решать дифференциальные уравнения аналитически?

Компьютерная обработка

5 Анализ временных рядов. Фурье и вейвлет-анализ

5.1 Краткие теоретические сведения

Одним из распространенных и практически важных случаев связи между аргументом и функцией является задание этой связи в виде некоторой таблицы (x_i, y_i) . В таком виде обычно задаются экспериментальные данные.

На практике часто приходится использовать табличные данные для приближенного вычисления при любом значении аргумента (из некоторой области). Этой цели служит задача приближения (аппроксимации) функций: *заданную таблично функцию $f(x)$ требуется приближенно заменить некоторой функцией $g(x)$ так, чтобы отклонение $g(x)$ от $f(x)$ в заданной области было наименьшим*. Функция $g(x)$ при этом называется **аппроксимирующей**. Если приближение строится на заданном дискретном множестве точек $\{x_i\}$, то аппроксимация называется *точечной*. К ней относятся интерполирование, среднеквадратичное приближение и др. При построении приближения на непрерывном множестве точек (например, на отрезке $[a, b]$) аппроксимация называется *непрерывной или интегральной*.

Для восстановления промежуточных значений функции, заданной таблично, используются различные алгоритмы интерполяции. Простейшей является **линейная интерполяция**. При этом функция в промежутках между заданными точками интерполируется отрезками прямых линий. Данный вид интерполяции обладает значительной погрешностью, особенно для быстро изменяющихся функций при редком расположении контрольных точек. Разложим функцию в ряд Тейлора

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \dots + \frac{(x - x_0)^n f^{(n)}(x_0)}{n!} + R_n$$

и ограничимся нулевым и первым членами разложения. Тогда на каждом отрезке $[x_n, x_{n+1}]$ функцию можно задать в виде

$$f_n(x) = y_n + (x - x_n) \frac{y_{n+1} - y_n}{x_{n+1} - x_n},$$

где y_n – табличное значение функции в точке x_n . Значительно меньшую погрешность имеет *интерполяция сплайнами*.

Сплайн — кусочная полиномиальная функция, которая на заданном отрезке может принимать очень простую форму, в то же время имея гибкость

и гладкость. Чаще всего для интерполяции функций используются *кубические сплайны*, представляющие собой набор полиномов третьей степени $p_n(x)$, проходящих через заданные точки. На точках сшивания, в которых один сплайн переходит в другой, на сплайны накладываются следующие ограничения:

- предыдущий сплайн должен переходить в следующий в точках сшивания, т.е. $p_n(x_n) = p_{n+1}(x_n)$;
- интерполированная функция должна быть гладкой, т.е. $p'_n(x_n) = p'_{n+1}(x_n)$;
- точки сшивания являются точками перегиба, т.е. $p''_n(x_n) = 0$ для всех n .

Таким образом, для полиномов p_n , определенных на $N - 1$ отрезке имеем $4N - 4$ коэффициентов. Граничные условия дают для этих коэффициентов $3N - 4$ уравнения. Недостающие N уравнений можно получить из дополнительных условий.

Однако, для квадратичных сплайнов перечисленных условий достаточно для формирования системы уравнений, из которой могут быть найдены все коэффициенты.

В двумерном пространстве аналогом кубических сплайнов является **пластинчатый сплайн** — поверхность, проходящая через узловые точки и минимизирующая энергию сгиба

$$I[f(x, y)] = \iint_{\mathbb{R}} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dx dy.$$

- При работе с гармоническими функциями, а также при определении спектра сигнала используются ряды Фурье. **Ряд Фурье** представляет собой бесконечный ряд с членами — гармоническими функциями

$$f(x) = \sum_{n=0}^{\infty} a'_n \cos(nx) + \sum_{n=0}^{\infty} b'_n \sin(nx).$$

Коэффициенты a_n и b_n рассчитываются по формулам

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx) dx, \quad b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx) dx.$$

При разложении функции в ряд Фурье необходимо помнить, что функция при этом определена на конечном отрезке. Вне этого отрезка функция будет повторяться, как периодическая. наилучшим образом разложению в ряд Фурье поддаются гармонические и псевдогармонические функции, при этом получается конечное число членов ряда.

При разложении временного ряда в ряд Фурье следует помнить, что если первая компонента соответствует элементарному колебанию с заданной частотой, остальные компоненты представляют собой высшие гармоники данного колебания, расположенные синфазно основному (косинусные члены) и со сдвигом на 90° (синусные члены). Значит, коэффициенты Фурье представляют собой амплитуды элементарных колебаний, составляющих суммарный сигнал.

Получение набора коэффициентов a_n и b_n на дискретном множестве точек и восстановление функции при помощи ряда Фурье называют **дискретным преобразованием Фурье** (ДПФ). Недостаток ДПФ в том, что при ограничении количества членов ряда результирующая функция заметно искажается.

Непрерывное преобразование Фурье задается формулами

$$F(k) = \mathcal{F}(f(x)) \equiv \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx, \quad f(x) = \mathcal{F}^{-1}(F(k)) \equiv \frac{1}{2\pi} \int_{-\infty}^{\infty} F(k) e^{2\pi i k x} dk.$$

$\mathcal{F}(f)$ называют *прямым*, а $\mathcal{F}^{-1}(F)$ — *обратным преобразованием Фурье*. Физическая величина k имеет обратную по отношению к x размерность. Следовательно, если x имеет значение времени, k является частотой; если x — координата, k — пространственная частота. Эта особенность позволяет применять Фурье-преобразования в спектральном анализе сигналов и изображений. Спектр сигнала совпадает с его преобразованием Фурье. Для сигнала, полученного смешиванием нескольких гармонических сигналов, посредством преобразований Фурье возможно определить частоту основной и побочных гармоник.

Фурье-анализ применяется и для фильтрации сигналов. Например, для отсека высокочастотной составляющей необходимо умножить спектр сигнала на функцию-ступеньку, значения которой до граничной частоты f_0 равно 1, а после нее 0. Последующее обратное преобразование Фурье преобразованного таким образом спектра позволит получить отфильтрованный сигнал.

Удобна Фурье-фильтрация и для преобразования изображений. В этом случае x имеет размерность радиус-вектора заданной точки изображения, а k — пространственной частоты. Примерной аналогией пространственной частоты является количество штрихов на изображении, расположенных поперек заданного направления \vec{k} . Таким образом, например, для удаления из изображения помех можно отсечь высокочастотную составляющую его Фурье-образа. Для этого на образе заменяем нулями значения $F(x)$ во всех точках, лежащих вне окружности заданного радиуса с центром в центре образа. Чем меньше радиус такой окружности, тем на более низких частотах будет происходить фильтрация. Отрицательной стороной такой фильтрации является потеря высокочастотных деталей изображения (мелких деталей, штриховок).

Так как свертка сигналов по свойству преобразования Фурье является произведением образов этих сигналов, $a(t) \otimes b(t) \equiv \mathcal{F}(a) \cdot \mathcal{F}(b) = A(\nu) \cdot B(\nu)$, преобразования Фурье широко используются в оптике и спектроскопии. Любой оптический прибор можно рассматривать как черный ящик, в котором происходит преобразование Фурье входного сигнала $f(t)$, умножение его на аппаратную функцию прибора $A(\nu)$ (фурье-образ передаточной функции прибора), зашумление (т.е. умножение на функцию $1 + \delta(\nu)$, где δ – случайная функция, характеризующая спектральную мощность шума), обратное преобразование Фурье:

$$\begin{aligned} \mathbf{f}(t) &\Rightarrow \mathcal{F}(f(t)) \equiv F(\nu) \Rightarrow B(\nu) = F(\nu) \cdot A(\nu) \Rightarrow \\ &\Rightarrow N(\nu) = B(\nu) \cdot (1 + \delta(\nu)) \Rightarrow \mathbf{f}'(t) \equiv \mathcal{F}^{-1}(N(\nu)). \end{aligned}$$

Спектральный прибор осуществляет пространственное перераспределение входящего излучения, так что выходной сигнал является еще и функцией координаты светоприемника. В этом случае прибор производит преобразование частотной координаты Фурье-спектра к пространственному виду:

$$\begin{aligned} \mathbf{f}(t) &\Rightarrow \mathcal{F}(f(t)) \equiv F(\nu) \Rightarrow B(\nu) = F(\nu) \cdot A(\nu) \Rightarrow \\ &\Rightarrow N(\nu) = B(\nu) \cdot (1 + \delta(\nu)) \Rightarrow \nu \rightarrow x \Rightarrow \mathbf{S}(x). \end{aligned}$$

- В середине прошлого века в математике появились новые методы анализа и восстановления функций — использование вейвлетов (wavelet, в отечественной математике — **функция-всплеск**). *Вейвлеты* — класс функций, использующихся для пространственной и масштабируемой локализации заданной функции. Семейство вейвлетов может быть образовано из функции $\psi(x)$ (ее иногда называют «материнским вейвлетом»), ограниченной на конечном интервале. «Дочерние» вейвлеты $\psi^{a,b}(x)$ образуются из «материнского» путем сдвига и масштабирования.

Вейвлеты широко используются для сжатия изображений, так как они обладают более гибкими свойствами по сравнению с преобразованием Фурье. Так, например, если изображение обладает резкими границами, Фурье-образ этого изображения дает **краевые эффекты** — полученная ступенька сглаживается, что искажает границы изображения, восстановленного из Фурье-образа.

Отдельный вейвлет можно определить как

$$\psi^{a,b}(x) = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right).$$

Тогда **базис вейвлетов**, соответствующих функции $f(x)$ определяется как

$$W_\psi(f)(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt.$$

В вейвлет–анализе важным условием является ортогональность базиса вейвлетов, по которым раскладывается функция.

Одним из известных семейств вейвлетов являются **вейвлеты Хаара**, построенные по аналогии с матричным *преобразованием Адамара* (за это их называют также вейвлетами Адамара–Хаара). «Материнский» вейвлет Хаара имеет вид

$$\psi(x) \equiv \begin{cases} 1, & 0 \leq x \leq 1/2; \\ -1, & 1/2 < x \leq 1; \\ 0, & \text{в других случаях.} \end{cases}$$

«Дочерние» вейвлеты задаются выражением $\psi^{j,k}(x) \equiv \psi(2^j x - k)$. Вейвлеты Хаара являются ортогональными.

Вейвлет–фильтрация имеет бóльшую гибкость по сравнению с Фурье–фильтрацией. Удаляя из вейвлет–образа члены с определенным масштабом можно исключить шумы из сигнала. Однако, если Фурье–фильтрация позволяет исключить только шум, подчиняющийся гармоническому закону, вейвлет–фильтрация очищает сигнал практически от любого шума (необходимо только правильно подобрать базис вейвлетов).

5.2 Примеры практических заданий

Рассмотрим методы аппроксимации, предлагаемые MATLAB. Для аппроксимации одномерных данных используется функция `interp1(x,Y,xi,method)` со следующими аргументами:

- `x` — массив аргументов табличной функции;
- `Y` — массив значений табличной функции;
- `xi` — значение (или массив значений) аргумента, для которого необходимо рассчитать приближенные значения функции;
- `method` — метод интерполяции (в случае отсутствия данного аргумента используется линейная интерполяция):
 - `'nearest'` — замена значения функции ближайшим известным значением;
 - `'linear'` — кусочно–линейная интерполяция;
 - `'spline'` — интерполяция кубическими сплайнами;
 - `'cubic'` — кусочно–кубическая эрмитова интерполяция.

Зададим массив аргументов `x=[1:10]` и функций `y=sin(x)`. Теперь рассмотрим, как действуют различные методы интерполяции.

```
>> xi=[1:.05:10];
>> y_nearest=interp1(x,y,xi,'nearest');
```

```
>> y_linear=interp1(x,y,xi,'linear');
>> y_spline=interp1(x,y,xi,'spline');
>> y_cubic=interp1(x,y,xi,'cubic');
```

Построим графики получившихся функций с подписями, соответствующими каждой интерполяции:

```
>> plot(x,y, 'o', xi, [y_nearest; y_linear; y_spline; ...
    y_cubic],xi,sin(xi),'.')
>> legend('DATA','nearest', 'linear','spline','cubic','sinusoide')
```

Как видите, наиболее близкой к реальной синусоиде оказалась интерполяция сплайнами.

Рассчитаем теперь среднее квадратичное отклонение для каждого вида интерполяции:

```
>> std(y_nearest-sin(xi))
ans =
    0.2031
>> std(y_linear-sin(xi))
ans =
    0.0618
>> std(y_spline-sin(xi))
ans =
    0.0068
>> std(y_cubic-sin(xi))
ans =
    0.0394
```

Действительно, наименьшим отклонением от заданной функции обладает интерполяция сплайнами.

Теперь рассмотрим аппроксимацию функции $y = \exp(-x/10) \cos(x)$ на отрезке $[1, 20]$ по сетке с шагом 0.01:

```
>> x=[1:20];
>> y=exp(-x/10).*cos(x);
>> x_i=[1:.01:20];
>> tic;y_linear=interp1(x,y,xi,'linear');toc
Elapsed time is 0.001192 seconds.
>> tic;y_nearest=interp1(x,y,xi,'nearest');toc
Elapsed time is 0.001147 seconds.
>> tic;y_cubic=interp1(x,y,xi,'cubic');toc
```

```
Elapsed time is 0.003650 seconds.
>> tic;y_spline=interp1(x,y,xi,'spline');toc
Elapsed time is 0.017406 seconds.
```

Функциональные скобки `tic...toc` позволяют засечь время выполнения расположенных между ними команд. Как видите, интерполяция сплайнами происходит в 15 раз медленнее интерполяции ближайшим числом (и такая небольшая разница возникает лишь благодаря высокой оптимизации вычислительных процессов в MATLAB). Проверив значения стандартного отклонения и в этом случае обнаружим значительное превосходство интерполяции сплайнами над другими методами.

- Теперь рассмотрим задачу построения спектра сигналов с амплитудной и частотной модуляциями. Пусть наши сигналы заданы на интервале $[0, 100]$ с с шагом 0.001 с. Оригинальные сигналы представляют собой синусоиды с частотами 10 Гц и 0.1 Гц. Результирующие сигналы — амплитудно- и частотно-модулированный вторым первый сигнал. Получим спектры сигналов:

```
>> x=[0:0.001:99.999]; % Для "ровного" счета
>> y1=sin(x*2*pi*10); y2=sin(x*2*pi/10); % сами сигналы
>> y_AM=y1.*y2; % амплитудно модулированный сигнал
>> y_FM=sin(x*2*pi.*(10+y2/10)); % частотно модулированный сигнал
>> SP_FM=fft(y_FM); % спектр ЧМ-сигнала
>> SP_AM=fft(y_AM); % спектр АМ-сигнала
>> MAX=length(SP_AM); % длина спектрального вектора
>> scale=( [1:MAX]-MAX/2)/MAX/0.001; % шкала частот
>> scale(1)=[]; SP_AM(1)=[]; SP_FM(1)=[]; % коррекция сдвига
```

Особое внимание обратите на построение шкалы частот: для графического отображения преобразования Фурье мы, как и ранее, будем использовать функцию `fftshift`, центрирующую фурье-образ относительно области определения. Таким образом, если l — длина вектора-аргумента, то центр распределения будет в ячейке с номером $l/2$. Частотный предел образа ν_{max} , согласно **теореме Котельникова–Найквиста**, равен половине частоты дискретизации сигнала, т.е. $1/(2\Delta x)$. Следовательно, отцентрированный (т.е. уже поделенный пополам) интервал шкалы частот необходимо еще и умножить на отношение ν_{max}/l , чтобы получить правильные пределы. Подводя итог сказанному, получим формулу для перевода аргумента функции-сигнала в частотный диапазон (n — номер элемента вектора):

$$\nu(n) = \frac{n - l/2}{l \cdot \Delta x}.$$

Постройте график спектра


```
>> plot(scale,fftshift(abs(SP_AM).^2))
```

Как видите, спектр АМ-сигнала представляет собой четыре пика: расстояние между боковыми пиками равно удвоенной частоте модулирующего сигнала, а расстояние между центрами боковых пиков равно удвоенной частоте несущего сигнала (можете это проверить). Вершины боковых пиков отстоят на ± 0.1 Гц от несущего сигнала ± 10 Гц. Постройте на одном графике спектры модулирующего, несущего и АМ сигналов:

```
>> SP1=fft(y1); SP1(1)=[];
>> SP2=fft(y1); SP2(1)=[];
>> plot(scale,[fftshift(abs(SP_AM)); fftshift(abs(SP1));
>> fftshift(abs(SP2))])
```

Теперь отобразим спектр ЧМ-сигнала.

```
>> FM=fftshift(abs(SP_FM).^2); % спектр
>> plot(scale,FM) % график всего спектра
>> plot(scale(50000:50500),FM(50000:50500)) % подробней
```

Теперь спектр представляет собой довольно широкую полосу частот около несущей.

- ▶ Попробуем применить простейшую Фурье-фильтрацию к зашумленному сигналу.

```
>> sig=cos(2*pi*t/30).*sin(2*pi*t*7).*log(1+t); % наш "сигнал"
>> noisy=awgn(sig,10,'measured'); % зашумленная копия сигнала
>> plot(t,[sig;noisy]) % посмотрим, что получилось
>> SP=fft(sig); % спектр сигнала
>> SP_n=fft(noisy); % спектр зашумленного сигнала
>> L=length(SP); % длина вектора данных
>> freq=(1:L)-L/2)/L/.01; % переходим к пространству частот
>> plot(freq,[fftshift(abs(SP));fftshift(abs(SP_n))]); % спектры
>> SP_f=SP_n; % инициализируем отфильтрованный спектр
>> SP_f(find((real(SP).^2+imag(SP).^2)<7))=0; % удаляем составляющие
% с низкой мощностью (менее 7)
>> plot(freq,fftshift(abs(SP_f))); % посмотрим отфильтрованный спектр
>> sig_f=real(ifft2(SP_f)); % восстановим сигнала
>> plot(t,sig_f) % отобразим восстановленный сигнал
>> plot(t,[noisy;sig_f]) % вместе с зашумленным
>> plot(t,[noisy;sig_f;sig]) % и все вместе
>> std(sig-sig_f) % погрешность
```

```
ans =
    0.0744
```

В данном случае мощность шумов была невысокой, поэтому фильтрация мало-мощных сигналов почти не повлияла на основной сигнал.

Другой вариант фильтрации — усечение высокочастотной составляющей. Отсечем все гармоники с частотами выше 15 Гц:

```
>> SP_f1=fftshift(SP_n); % сдвигаем спектр для фильтрации
>> SP_f1(find(abs(freq)>15))=0; % обнуляем высокие частоты
>> sig_f1=real(ifft2(ifftshift(SP_f1))); % восстанавливаем сигнал
>> plot(t,[sig_f1;sig]) % сравниваем с сигналом
>> plot(t,[sig_f1;noisy]) % и с зашумленным сигналом
```

Это аналогично вычислению шума и вычитанию его из сигнала:

```
>> SP_f1=fftshift(SP_n);
>> SP_f1(find(abs(freq)<15))=0;
>> pure_noise=real(ifft2(ifftshift(SP_f1)));
>> sig_f2=noisy - pure_noise;
>> plot(t,[sig_f2-sig_f1])
```

Результат еще лучше мы сможем получить, если извлечем из спектра только узкую область, содержащую максимумы спектральной плотности:

```
>> SP_f1=fftshift(SP_n);
>> SP_f1(find(abs(abs(freq)-7)<1))=0;
>> pure_noise=real(ifft2(ifftshift(SP_f1)));
>> sig_f2=noisy - pure_noise;
>> plot(t,[sig_f2;sig])
```

Можно попытаться выделить низкочастотную составляющую сигнала (осуществить *детектирование АМ сигнала*):

```
>> SP_n=fft(abs(noisy)); % получаем спектр сигнала
>> SP_f1=fftshift(SP_n); % подготавливаем
>> SP_f1(find(abs(freq)>3))=0; % убираем несущую
>> sig_f2=real(ifft2(ifftshift(SP_f1))); % детектируем
>> plot(t,[sig_f2;sig])
```

- ▶ Следующая задача часто встает перед астрофизиком: имея модель спектра в лабораторной системе отсчета и реальный спектр звезды определить смещение спектра звезды относительно лабораторного (т.е. найти лучевую скорость звезды).

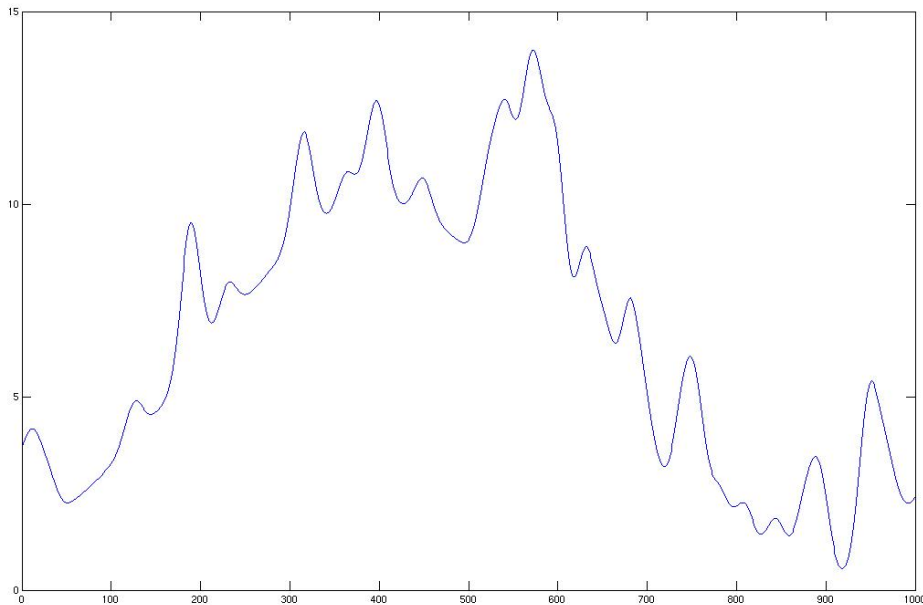
Сгенерируем «псевдоспектр» звезды и две его зашумленные копии с уровнями S/N 50 дБ и 10 дБ. Попробуем определить смещение «спектра» относительно лабораторного при помощи Фурье-анализа.

```

1 >> x=[1:1000]; % абсциссы спектра
2 >> y=normpdf(x,400,200)*5000; % "континуум"
3 >> gaussian=normpdf(x,500,10)*25; % профиль спектральной "линии"
4 >> dotz=[200,310,487,500,512,515,689,891,900,901,950];
5 >> deltaz=zeros(1, 1000);
6 >> deltaz(dotz)=1;
7 >> deltaz(round(dotz/1.1))=1;
8 >> deltaz(round(dotz/1.3))=1;
9 >> deltaz(round(dotz/2))=1;
10 >> deltaz(round(dotz/5))=1;
11 >> deltaz(round(dotz/7))=1;
12 >> deltaz(round(dotz/11))=1;
13 >> Fdeltaz=fft(deltaz);
14 >> Fgaussian=fft(gaussian);
15 >> Spectra=ifft(Fdeltaz.*Fgaussian)+y; % собственно "спектр"
16 >> plot(Spectra)
17 >> Spectra_10=awgn(Spectra,10,'measured');
18 >> Spectra_50=awgn(Spectra,50,'measured');
19 >> Spectra_10_moved=zeros(1,1000);
20 >> Spectra_50_moved=zeros(1,1000);
21 >> Spectra_10_moved(111:end)=Spectra_10(1:890); % сдвинутые копии
22 >> Spectra_50_moved(111:end)=Spectra_50(1:890); % с зашумлением
23 >> [corr,lags] = xcorr(Spectra_50_moved,Spectra);
24 >> plot(lags,corr)
25 >> lags(find(corr==max(corr))) % положение сдвига
26 ans =
27     110
28 >> [corr10,lags10] = xcorr(Spectra_10_moved,Spectra);
29 >> lags10(find(corr10==max(corr10)))
30 ans =
31     109
32 >> [corr15,lags15] = xcorr(Spectra_10_moved,Spectra_50);
33 >> lags15(find(corr15==max(corr15)))
34 ans =
35     109

```

Вектор `dotz` содержит базовые координаты для расчета положения «линий»

Рис. 1: Сгенерированный псевдоспектр (`plot(Spectra)`)

в спектре. Инициализируя некоторые компоненты вектора `deltaz` единицами и сворачивая его с гауссианой, получим профиль «эмиссионных линий», сложив который с «непрерывным спектром» сформируем собственно псевдоспектр. Все используемые команды должны быть вам уже известны. Как видите, даже сигнал с $S/N = 10$ дБ содержит достаточно информации для точного определения сдвига. Координата максимума кросс-корреляционной функции может немного отличаться от 110 из-за того, что сдвинутые компоненты потеряли часть информации (недостающие данные мы просто заменили нулями).

- Теперь попробуем восстановить сигнал при помощи вейвлет-анализа, а также определить сдвиг. Сохраните вектор сдвинутого зашумленного спектра:

```
>> save Spectra_moved Spectra_10_moved
```

Теперь в меню MATLAB *Start* выберите *Toolboxes* \Rightarrow *Wavelet* \Rightarrow *Wavelet toolbox main menu*. В появившемся окне нажмите кнопку *Wavelet 1-D*. Откроется окно одномерных Wavelet-преобразований. Загрузите данные (*File* \Rightarrow *Load* \Rightarrow *Signal* \Rightarrow *Spectra_moved*).

На панели справа из выпадающего меню *Wavelet* выберите *dmey* (дискретные вейвлеты Мейера) — эти вейвлеты лучше всего удовлетворяют задаче фильтрации астрофизических спектров. Из меню *Level* выберите четвертый уровень. Нажмите кнопку *Analyze*. Вы увидите графики аппроксимирующего (a) и детализирующих (d) коэффициентов разложения «спектра» по вейвлетам Мейера. Аппроксимирующий коэффициент несет в себе низкочастотную составляющую (огibaющую), детализирующие — содержат высокочастотные составляющие

щие (причем коэффициент `d1` явно содержит шум). Как видите, аппроксимирующий коэффициент почти соответствует нашему спектру.

Однако, при помощи *Wavelet toolbox* можно осуществлять wavelet-декомпозицию и в командной строке при помощи команд `wavedec` и `wrcoef`. Создадим вектор-огibaющую «зашумленную спектра»:

```
>> [C,L]=wavedec(Spectra_10_moved,4,'dmey'); %декомпозиция сигнала
>> approx10=wrcoef('a',C,L,'dmey',4); %аппроксимирующий коэффициент
>> plot(approx10) % посмотрим, что получилось
```

а также вектор-огibaющую «лабораторного спектра»:

```
>> [C,L] = wavedec(Spectra,4,'dmey');
>> approx = wrcoef('a',C,L,'dmey',4);
>> plot(approx)
```

и проверим, чему равен «сдвиг»

```
>> [corrw,lagsw] = xcorr(approx10,approx);
>> lagsw(find(corrw==max(corrw)))
ans =
    109
```

5.3 Задания для самостоятельного выполнения

1. Затабулируйте показательную функцию на отрезке $[6, 14]$ с шагом 0.1. Вычислите теперь при помощи этой таблицы значение произведения $X = 1097 \cdot 1013$ (воспользуйтесь свойством логарифмов: $\ln X = \ln 1097 + \ln 1013$). Для нахождения значений $\ln 1097$ и $\ln 1013$ воспользуйтесь аппроксимацией сплайнами. Аналогично, при помощи аппроксимации сплайнами вычислите X . Для замены экспоненциальной функции на логарифмическую при вычислении приближенного значения логарифмов меняйте местами x и y при вызове функции `interp1`. Значение X найдите, интерполируя показательную функцию.
2. Создайте вектор-сигнал, представляющий собой сумму двух синусоид с $\nu_1 = 50$ Гц и $\nu_2 = 120$ Гц на промежутке $t \in [0, 0.25]$ с с периодом дискретизации 0.001 с. Добавьте к нему аддитивного нормального шума:

```
y = y + 2*randn(size(t));
```

Постройте спектр итогового сигнала, определите по спектру частоты исходных сигналов.

3. Выполните упражнение по фильтрации сигнала (стр. 57), задав в качестве граничной мощности 30. Сравните полученный сигнал с оригиналом. Рассчитайте среднеквадратичное отклонение отфильтрованного сигнала от оригинала (0.06).
4. Создайте спектр как в упражнении со стр. 59 и его зашумленную копию с $S/N = 0$ дБ, сдвинутую на 110 единиц вправо. Определите этот сдвиг при помощи кросс-корреляционной функции (117).
5. Теперь определите для этого же спектра сдвиг при помощи вейвлет-декомпозиции вейвлетами Мейера на четвертом уровне, построив кросс-корреляционную функцию для четвертых аппроксимирующих коэффициентов (118). Затем — используя кросс-корреляционную функцию суммы четвертого аппроксимирующего и четвертого детализирующего коэффициентов (115). Детализирующие коэффициенты строятся путем указания ключа 'd':

```
>> approxd = wrcoef('d',C,L,'dmey',4);
```

6. Создайте спектр, сдвинутый на 2 единицы относительно исходного (зашумленного с $S/N = 0$ дБ):

```
>> Spectra_00_moved=zeros(1,1000);
>> Spectra_00_moved(3:end)=Spectra_0(1:998);
```

При помощи вейвлет-декомпозиции на пятом уровне с использованием только пятого аппроксимирующего коэффициента определите сдвиг относительно модельного спектра (3). Также определите сдвиг из кросс-корреляционной функции (9).

7. Создайте зашумленную копию лабораторного сигнала с $S/N = -10$ дБ. Проведите вейвлет-анализ этого сигнала при помощи интерактивного окна Wavelet toolbox. Сдвиньте его на 57 единиц вправо и попытайтесь определить сдвиг при помощи кросс-корреляционной функции аппроксимирующих коэффициентов вейвлет-декомпозиции получившегося и лабораторного сигналов.

5.4 Контрольные вопросы

1. Дайте определение операциям аппроксимации, интерполяции, экстраполяции.
2. В чем сходство и в чем различие аппроксимации и интерполяции? Возможно ли, чтобы аппроксимирующая кривая совпадала с интерполяционной?
3. Какие функциональные зависимости в наилучшей степени подлежат ли-

- нейной интерполяции? В каких случаях линейная интерполяция даст значительную ошибку?
4. Перечислите основные граничные условия, позволяющие произвести интерполяцию квадратичными, кубическими сплайнами. Можно ли считать данные виды интерполяции однозначными?
 5. Напишите формулы, используемые для разложения функции в ряд Фурье, формулы прямого и обратного непрерывного преобразования Фурье.
 6. Назовите достоинства и недостатки Фурье–анализа. Почему Фурье–анализ имеет очень широкое применение, несмотря на свои недостатки?
 7. В чем заключается преимущество вейвлет–анализа над Фурье–анализом?
 8. Как строится базис дочерних вейвлетов? Всегда ли необходимо соблюдать ортогональность базиса вейвлетов?
 9. Как определить глубину рекурсии при вейвлет–анализе заданного ряда данных?
 10. Какую информацию несет набор аппроксимирующих коэффициентов при вейвлет–анализе ряда данных? Что содержит в себе набор последних детализирующих коэффициентов?
 11. Можно ли использовать вейвлет–анализ для частотной фильтрации сигнала?
 12. Сформулируйте теорему Котельникова–Найквиста. Для чего при дискретизации звукозаписей частоту дискретизации устанавливают равной $44.1 \div 48$ кГц, несмотря на то, что человек не слышит частот свыше $10 \div 20$ кГц (и звуковая аппаратура даже Hi-End класса имеет глубокий завал на частотах свыше 18 кГц)?
 13. Как при помощи Фурье–фильтрации можно вычистить из сигнала его среднюю величину? Очистить от шумов? Провести АМ–детектирование?
 14. Почему даже при сильном зашумлении двух копий одного и того же сигнала из вида их корреляционной функции можно определить сдвиг между сигналами?

Компьютерная обработка

6 Обработка изображений

6.1 Краткие теоретические сведения

Изображение представляет собой двумерную функцию $f(x, y)$, где x и y — пространственные координаты, а уровень f называется **интенсивностью** или **уровнем серого** изображения в данной точке (цветное изображение является совокупностью по крайней мере трех функций $r(x, y)$, $g(x, y)$ и $b(x, y)$). Если величины x , y и f принимают дискретные значения, говорят о *цифровом изображении*. Элементарная единица цифрового изображения называется **пикселем** (pixel — picture element). Под *цифровой обработкой изображений* подразумевается обработка цифровых изображений с помощью ЭВМ.

В отличие от человека, чье зрение приспособлено лишь к сравнительно узкому частотному диапазону, ЭВМ все равно, каковы частоты в обрабатываемом ею изображении. Т.е. к цифровым изображениям следует относить также: ультразвуковые, рентгеновские, электронные (микроскопия), магнитнорезонансные (томография) и прочие изображения.

Несмотря на то, что в окружающем нас мире нет ничего непрерывного (абсолютно все в нашей Вселенной подвержено квантованию), на уровне классической физики основные наблюдаемые можно считать непрерывными, т.к. по сравнению с их величиной величина кванта наблюдаемой пренебрежимо мала (сравните средний рост человека и квант пространства). Однако, даже самый современный суперкомпьютер по своей производительности не может сравниться с человеческим мозгом. Поэтому для обработки изображения при помощи компьютера его необходимо **квантовать**, т.е. преобразовать из квазинепрерывной формы в дискретную. Квантование производится как в пространстве координат, так и в пространстве амплитуд. Следовательно, процедуру квантования (*дискретизации*) квазинепрерывного изображения $I_0(X, Y)$ в его дискретную форму $I(x, y)$ можно представить в виде:

$$I(x, y) = \text{round} \left(\frac{2^N - 1}{I_{max}} \int_{S_{x,y}} I_0(X, Y) dX dY \right) + \delta_{x,y}. \quad (6.1)$$

Здесь I_{max} — максимальная интенсивность дискретизируемого сигнала, N — разрядность по интенсивности, интегрирование ведется по площади пикселя с координатами (x, y) , $\delta_{x,y}$ — шум дискретизации, возникающий при преобразовании сигнала.

В MATLAB изображению соответствует матрица. Если изображение серое, матрица двумерная. Координаты изображения отсчитываются от первого элемента матрицы, первая координата (x) соответствует номеру строки матрицы, вторая — номеру (y) столбца. Цветному изображению соответствует трехмерная матрица, в которой каждый из слоев соответствует цветовой компоненте R, G или B.

Простейшими методами преобразования изображений являются манипуляции в *пространственной области*, т.е. непосредственные операции с пикселями изображений, которые можно обозначить выражением $g(x, y) = T[f(x, y)]$, где T — некоторый оператор над изображением. Оператор T может применяться как ко всему изображению, так и к окрестности отдельной точки (x_0, y_0). Оператор T принимает простейший вид при линейных преобразованиях яркости изображения: $g(x, y) = \lambda \cdot f(x, y)$, либо же при более сложных функциональных зависимостях, *одинаковых для всех пикселей изображения*.

Некоторые типы изображений для более наглядного отображения необходимо логарифмировать. Человеческие органы чувств работают в шкале логарифмов возмущений (именно поэтому мощности сигналов по отношению к воздействию на человеческие органы чувств принято измерять в децибеллах), поэтому в логарифмической шкале некоторые изображения могут выглядеть значительно четче. Логарифмические преобразования выполняются при помощи выражения $g(x, y) = \mathfrak{C} \ln(1 + f(x, y))$. Основное применение логарифмического преобразования состоит в сжатии динамического диапазона. Особенно полезным является логарифмическое преобразование Фурье–спектра изображения (т.к. динамический диапазон Фурье–спектра может быть довольно значительным и его линейное преобразование не позволит в должной степени отразить переходы между величинами повышенной и пониженной амплитуд).

- ▶ **Гистограммой** цифрового изображения, число возможных уровней яркости L которого лежит в диапазоне $[0, G]$, называется дискретная функция $h(r_k) = n_k$, где r_k — k -й уровень яркости, n_k — число пикселей изображения с уровнем яркости r_k . Часто бывает удобно работать с *нормированными гистограммами*, которые получаются делением $h(r, k)$ на общее количество пикселей изображения.

Изображения могут иметь низкий динамический диапазон, особенно если они получены в неподходящих условиях съемки. Для улучшения качества изображения можно применять **эквализацию** его гистограммы. Эквализация заключается в том, что гистограмма изображения расширяется на весь возможный динамический диапазон яркостей. Т.е., минимальная яркость исходного изображения заменяется на нуль, а максимальная — на предельное значение яркости. Остальные величины яркости равномерно распределяются в расширенном динамическом диапазоне в соответствии с первоначальной гистограммой.

Кроме того, существуют методы «подгонки» гистограммы, когда гистограмма изображения подгоняется под заданную функцию, т.е. когда уровни яркости отдельных пикселей меняются в соответствии с их положением на новой гистограмме.

- ▶ Для коррекции изображений зачастую применяется матричная пространственная фильтрация, заключающаяся в свертке исходного изображения с некоторой маской, задающей фильтр. Так, например, фильтр Лапласа, позволяющий получить лапласиан исходного изображения, реализуется при помощи свертки с маской $\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, которая получается из дискретного приближения второй производной.

Нелинейные фильтры определяют значение пикселя нового изображения исходя из результата упорядочивания значений в некоторой окрестности соответствующего пикселя исходного изображения. Самым известным из нелинейных фильтров является медианный фильтр, позволяющий бороться с импульсными шумами «соль и перец» (данный шум изменяет значения яркости в некоторых точках изображения так, что они принимают нулевое либо максимальное значение).

- ▶ **Двумерное дискретное преобразование Фурье (ДПФ)** изображения задается выражением

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i(\frac{ux}{M} + \frac{vy}{N})}, \quad u = \overline{0, M-1}, \quad v = \overline{0, N-1}.$$

Частотной областью называется координатная система, задающая аргументы $F(u, v)$ частотными переменными u и v . Обратное ДПФ задается выражением

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{2\pi i(\frac{ux}{M} + \frac{vy}{N})}, \quad x = \overline{0, M-1}, \quad y = \overline{0, N-1}.$$

Даже если изображение f вещественно, его преобразование Фурье F , как правило, является комплексным. Для визуализации F используется вычисление абсолютной величины этой функции (*спектра* изображения). Фурье-образ изображения обладает периодичностью с периодом, равным размеру изображения. Поэтому для большего удобства визуализации его можно привести к системе координат с нулем в центре изображения. Для этого в спектре производится перестановка первого и третьего квадрантов и второго и четвертого.

Фурье-фильтрация изображения формально ничем не отличается от фильтрации одномерных сигналов. Т.е. если известна функция фильтра $g(x, y)$, филь-

трованное изображение получается путем свертки

$$f'(x, y) = f(x, y) \otimes g(x, y),$$

или, учитывая свойства Фурье–преобразований,

$$f'(x, y) = \mathcal{F}^{-1}(\mathcal{F}(f(x, y)) \cdot \mathcal{F}(g(x, y))) = \mathcal{F}^{-1}(F(u, v)G(u, v)).$$

Функцию $G(u, v)$ принято называть **передаточной функцией** фильтра.

Фурье–преобразования некоторых периодических функций могут иметь артефакты, называемые **краевыми эффектами**: между смежными периодами ДПФ могут возникнуть эффекты интерференции, искажающие вид спектра изображения. Для избавления от краевых эффектов следует дополнить изображение (размера $M \times N$) по краям нулями, сформировав матрицу размером $P \times Q$, где $P \geq 2M - 1$, $Q \geq 2N - 1$. Алгоритм ДПФ работает быстрее, когда P и Q являются степенями двойки, поэтому эти числа (особенно для больших изображений) стараются выбрать именно исходя из критерия скорости.

Наиболее распространенными видами фильтрации в частотной области являются низкочастотная (НЧ) и высокочастотная (ВЧ) фильтрация. При НЧ фильтрации осуществляется размытие изображения, ВЧ фильтрация, наоборот, повышает резкость изображения.

Легко преобразовать НЧ фильтр в соответствующий ему ВЧ фильтр и наоборот. Пусть $LP(u, v)$ – передаточная функция НЧ фильтра, тогда передаточная функция соответствующего ему ВЧ фильтра вычисляется как $HP = 1 - LP$. Так как ВЧ фильтры отбрасывают нулевую составляющую изображения (среднюю интенсивность), необходимо добавлять некоторое смещение к отфильтрованному изображению. ВЧ усиление имеет передаточную функцию $HP_E(u, v) = a + b \cdot HP(u, v)$, где a – смещение, b – коэффициент усиления.

- ▶ При получении изображения всегда используются реальные приборы, которые неизменно добавляют к изображению тот или иной вид шума. Используя свойства Фурье–преобразований, можно записать спектр полученного изображения в виде

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \Leftrightarrow g(x, y) = h(x, y) \otimes f(x, y) + n(x, y).$$

Функция $h(x, y)$ называется **функцией рассеяния точки** (ФРТ), а ее Фурье–образ $H(u, v)$ – **оптической передаточной функцией** (ОПФ). Функция $n(x, y)$ характеризует шум.

Для оцифрованных изображений характерен периодический шум

$$n(x, y) = A \sin \left[\frac{1}{M} 2\pi u_0 (x + B_x) + \frac{1}{N} 2\pi v_0 (y + B_y) \right],$$

т.е. обратное Фурье–преобразование комплексно–сопряженных единичных импульсов, находящихся в точках (u_0, v_0) и $-u_0, -v_0$. Параметры периодического шума можно оценить по виду Фурье–спектра изображения: такой шум порождает частотные всплески, которые можно обнаружить даже визуально. Кроме того, в данном случае хорошие результаты дает исследование изображения при помощи его центральных моментов. Наиболее часто используются первый нулевой (математическое ожидание) и второй центральный (дисперсия) моменты.

Еще одним видом фильтрации является *адаптивная медианная фильтрация*. Ее отличие от «обычной» медианной фильтрации в том, что размер окрестности текущего пикселя, по которой проводится усреднение, изменяется в зависимости от содержимого этой окрестности. Чем более равномерным является распределение интенсивности вокруг данной точки, тем большим будет размер окрестности для фильтрации, и наоборот. Адаптивная медианная фильтрация помогает избавиться от шумов типа «соль и перец», не смазывая мелких деталей изображения.

- ▶ Самый простой подход к восстановлению изображения состоит в построении приближения вида $\hat{F}(u, v) = G(u, v)/H(u, v)$, после чего $\hat{f}(x, y) \approx \mathcal{F}^{-1}(F)$. Такой метод называется **инверсной фильтрацией**. Однако, вблизи нулей H этот метод дает большие проблемы, кроме того он не позволяет полностью избавиться от шума.

Более надежным методом является **винеровская фильтрация**. Винеровский фильтр ищет приближение \hat{f} , минимизирующее функцию $D = \langle (f - \hat{f})^2 \rangle$. Решение этой задачи в частотной области зависит от спектра искажающей функции и энергетического спектра шума.

Аналогично одномерным сигналам, в двумерном пространстве также вводятся вейвлет-преобразования. Фильтрация двумерными вейвлетами позволяет зачастую добиться результатов, недоступных Фурье–фильтрации.

- ▶ Особой проблемой в области цифровых изображений является распознавание объектов. Распознавание объектов применяется в системах распознавания печатного текста, системах электронного зрения, охранных системах, системах дорожного видеонаблюдения и многих других. Методика распознавания базируется как на корреляционных функциях (поиск максимума корреляции изображения с образцом), так и на адаптивных системах, использующих элементы искусственного интеллекта.

Многие системы распознавания основаны на изучении **скелетов** изображения — своего рода каркасов, на которые «натянуты» детали изображения. Со скелетами проще проводить операции аффинных преобразований, что помогает распознавать повернутый или имеющий другой масштаб образец.

6.2 Примеры практических заданий

Для загрузки изображений в рабочее пространство MATLAB используется функция `imread`:

```
>> My_image = imread('/Data/images/myimage.xbm');
```

За отображение изображения на экране отвечает функция `imshow(Im, Gray)`. Первый аргумент `Im` — матрица-изображение, второй аргумент не является обязательным, он содержит число уровней яркости для изображения. Если написать `imshow(Im, [])`, для отображения будет использован весь диапазон значений яркостей изображения `Im`. Сохранить изображение можно при помощи команды `imwrite`.

Для просмотра, загрузки и сохранения изображений можно использовать также `imtool` — графический интерфейс пакета *Image Processing Toolbox*, который вызывается командой `imtool`. Запись `imtool(f)` практически аналогична `imshow(f)`.

Откройте какое-нибудь из изображений, хранящихся на жестком диске (их можно найти при помощи поиска: командой `locate .jpg`, или через графический поиск (например, при помощи *Beagle*), нажав сочетание клавиш `<Meta> + <F>`). Если изображение цветное, его необходимо свести к черно-белому виду и преобразовать тип данных в `int8` (если изображение содержит другой тип данных) для упрощения работы. Затем можно построить спектр изображения. Действия могут быть аналогичны следующему:

```
>> RGB = imread('/Data/Misc/html/pict/6.jpg'); % считываем
>> BW = (RGB(:,:,1) + 2*RGB(:,:,2) + RGB(:,:,3))/4; % преобразуем
>> imshow(BW) % смотрим, что получилось
```

Скорее всего, гистограмма нового изображения будет узкой. Впоследствии мы преобразуем его гистограмму.

- ▶ Алгоритм ДПФ изображений реализован в MATLAB при помощи функции `fft2`. Для получения спектра Фурье необходимо выполнить команду `S=abs(F)`, а для его визуализации — `imshow(fftshift(S), [])`. Для лучшей визуализации ДПФ можно представить его в логарифмическом масштабе. Если вы выполнили центрирование изображения при помощи функции `fftshift`, вернуть его к оригинальному виду (например, для выполнения каких-либо вычислений) можно при помощи команды `ifftshift`. Обратное преобразование Фурье вычисляется при помощи команды `ifft2`.

Построим Фурье-спектр изображения и его логарифмическое преобразование, а также логарифмическое преобразование изображения. Логарифмическое преобразование яркости изображения реализуется в MATLAB при помощи команд

```
>> g = im2uint8(mat2gray(log(1+double(f))));
```

Таким образом, необходимо ввести такой набор команд:

```
>> BW_sp = fft2(double(BW));
>> imshow(fftshift(real(BW_sp)), [])
>> imshow(fftshift(abs(BW_sp)))
>> BW_sp_log = im2uint8(mat2gray(log(1 + abs(BW_sp))));
>> imshow(fftshift(BW_sp_log))
```

- ▶ Далее отобразим гистограмму изображения. При работе в MATLAB следует помнить, что индексы массивов начинаются с 1, а не с 0, как принято в большинстве языков программирования. Поэтому элементу гистограммы с нулевой яркостью соответствует первый элемент вектора-гистограммы. Простейший способ расчета гистограммы осуществляется при помощи функции `h = imhist(f)`. Гистограмму можно отобразить при помощи *столбчатых диаграмм*: `bar(horz, v, width)`, где `v` – вектор-строка, график которого нужно построить, `horz` – (необязательный) вектор приращений по горизонтальной шкале (т.е. координаты середины соответствующего столбца), необязательный параметр `width` (от 0 до 1) задает относительную ширину столбца. Итак, для отображения гистограммы необходимо набрать

```
>> h = imhist(BW);
>> bar(h)
```

Эквализацию гистограммы изображения можно осуществить при помощи функции `histeq`, возвращающей эквализированное изображение. Первым параметром функции является исходное изображение, а вторым — количество уровней яркости для эквализации. Если второй параметр является вектором, гистограмма подгоняется под задаваемую им функцию. Построим гистограмму нашего изображения и его спектра и попробуем эквализировать ее:

```
>> BW_h = histeq(BW);
>> bar(imhist(BW_h))
>> bar(imhist(real(BW_sp)));
>> BW_sp_h = histeq(real(BW_sp), 256);
>> imshow(BW_sp_h)
>> bar(imhist(real(BW_sp_h)));
```

- ▶ Линейная пространственная фильтрация в пакете Image Processing Toolbox реализуется функцией `imfilter`. Медианная фильтрация осуществляется при помощи функции `medfilt2(f, [m, n])`. Зашумление изображения можно реализовать при помощи функции `imnoise(f, 'salt & pepper', probab)`.

Создадим зашумленное изображение и попробуем отфильтровать его.

```
>> BW_n = imnoise(BW, 'salt & pepper', 0.1); % зашумляем BW
>> imshow(BW_n) % смотрим зашумленное изображение
>> BW_n_med0 = medfilt2(BW_n); % фильтруем по квадрату 3x3
>> imshow(BW_n_med0) % смотрим, что получилось
>> BW_n_med5 = medfilt2(BW_n,[5 5]); фильтруем по квадрату 5x5
>> imshow(BW_n_med5) % размытое изображение
```

- ▶ Для свертки изображений разных размеров необходимо привести их к единому масштабу. Вычислив размер расширенного изображения $P \times Q$, можно выполнить его Фурье-преобразование при помощи функции

```
>> F = fft2(f, P, Q);
```

При использовании расширения функция фильтра в частотной области также должна иметь размер $P \times Q$.

Для вычисления размера расширенной матрицы удобно использовать следующую функцию

```
function PQ = paddedsize(AB, CD, PARAM)
% paddedsize
% вычисляет размеры дополненной матрицы для фильтрации
% на основе ДПФ
%
% PQ = paddedsize(AB, 'pwr2')
% вычисляет PQ как ближайшая степень 2 к max(AB)
%
% PQ = paddedsize(AB, CD)
% вычисляет наименьшие целые четные числа,
% меньшие или равные AB+CD+1
%
% PQ = paddedsize(AB, CD, 'pwr2')
% вычисляет PQ как ближайшие степени 2 к max([AB CD])
%
if nargin == 1 % paddedsize(AB)
    PQ = 2*AB;
elseif nargin == 2 & ~ischar(CD) % paddedsize(AB,CD)
    PQ = AB + CD + 1;
    PQ = 2 * ceil(PQ/2); % ближайшее четное
elseif nargin == 2 % paddedsize(AB, 'pwr2')
    m = max(AB);
    P = 2^nextpow2(2*m); % ближайшая степень 2 к 2m
```

```

    PQ = [P, P];
elseif nargin == 3
    m = max([AB CD])
    P = 2^nextpow2(2*m);
    PQ = [P, P];
else
    error('Неверное число аргументов')
end

```

В комментариях описан синтаксис функции. Заметьте, что необязательно в качестве текстового аргумента писать 'pwr2' — он может быть любым символом, например, 'w'.

Еще одна удобная функция `lpfilter` реализует передаточные функции некоторых низкочастотных фильтров

```

function [H, D] = lpfilter(type, M, N, D0, n)
% H = lpfilter(type, M, N, D0, n)
% вычисляет передаточную функцию H низкочастотного фильтра
% типа type размера MxN
% значения type:
% 'ideal' идеальный НЧ фильтр с частотой отсечки D0>0
%           n необязательно
%
% 'btw' НЧ фильтр Баттеуоса (Butterworth) порядка n (по умолч. 1)
%           с частотой отсечки D0>0
%
% 'gaussian' НЧ фильтр Гаусса с частотой отсечки D0>0, n необязательно
%
% расчет маски для измерения расстояний от центра:
[U, V] = dftuv(M,N);
D = sqrt(U.^2 + V.^2);
switch type % выбираем тип фильтра
    case 'ideal'
        H = double(D<=D0);
    case 'btw'
        if nargin == 4
            n = 1;
end
H = 1./(1 + (D./D0).^(2*n));
case 'gaussian'
    H = exp(-(D.^2)./(2*(D0^2)));

```



```
        otherwise
            error('Неизвестный тип фильтра')
end
```

Для нее необходима еще одна функция: `dftuv` (она строит координатную матрицу в частотном диапазоне, аналогично построению вектора частот в предыдущей лабораторной работе):

```
function [U, V] = dftuv(M, N)
% dftuv создает частотную матрицу
% размера M x N
%
% инициализируем переменные
u = [0:M-1];
v = [0:N-1];
% вычисляем индексы для построения координатной матрицы
idx = find(u > M/2);
u(idx) = u(idx) - M;
idy = find(v > N/2);
v(idy) = v(idy) - N;
% Создаем матрицу
[V, U] = meshgrid(v,u);
```

Для большего удобства необходимо создать еще и функцию `dftfilt`, позволяющую применять к изображению заданный вид частотного фильтра:

```
function g = dftfilt(f, H)
% dftfilt осуществляет частотную фильтрацию
% f - фильтруемое изображение,
% H - передаточная функция фильтра
%
% выполняем Фурье-преобразование расширенного изображения
F = fft2(f, size(H, 1), size(H,2));
% выполняем фильтрацию
g = real(ifft2(H.*F));
% возвращаем к первоначальному размеру
g = g(1:size(f, 1), 1:size(f, 2));
```

Получить передаточную функцию фильтра можно при помощи функции `lpfilter`, а отфильтровать изображение внутренними средствами MATLAB (с использованием расширения нулями) — при помощи команд `fspecial` и `imfilter`. *Основные шаги фильтрации в частотной области:*

1. получить параметры расширения с помощью `paddedsz`:

```
PQ = paddedsz(size(f));
```

2. построить преобразование Фурье с расширением:

```
F = fft2(f, PQ(1), PQ(2));
```

3. сгенерировать функцию фильтра G размером $PQ(1) \times PQ(2)$, если фильтр был центрирован, до фильтрации необходимо выполнить `G=fftshift(G)`;

4. произвести свертку в Фурье-пространстве:

```
FF = G.*F;
```

5. найти вещественную часть обратного преобразования Фурье от FF :

```
ff = real(iff2(FF));
```

6. вырезать верхний левый прямоугольник исходных размеров:

```
ff = ff(1:size(f,1), 1:size(f,2));
```

Попробуем на нашем изображении некоторые ВЧ фильтры.

```
>> PQ = paddedsz(size(BW));
>> D0 = .05*PQ(1);
>> HBW = 1 - lpfilter('btw', PQ(1), PQ(2), D0, 2);
>> H = .5 + 2*HBW;
>> gbw = dftfilt(BW, HBW);
>> gbw = im2uint8(mat2gray(gbw));
>> ghf = dftfilt(BW, H);
>> ghf = im2uint8(mat2gray(ghf));
>> ghe = histeq(ghf, 256);
>> imshow(ghe);
>> figure, imshow(BW)
```

► Функция `statmoments(p, n)` позволяет вычислить центральные статистические моменты матрицы p до величины n .

Применим теперь к нашему зашумленному изображению адаптивную медианную фильтрацию. Попробуем несколько различных значений окна усреднения. Функция `adpmedian(g, Smax)` позволяет осуществлять адаптивную медианную фильтрацию изображения g с максимальным размером окна $Smax$. Листинг функции:

```

function f = adpmedian(g, Smax)
% adpmedian осуществляет адаптивную медианную фильтрацию
% Медианный фильтр начинает с квадрата 3 x 3 и
% увеличивается до размера Smax x Smax
% Smax должно быть НЕЧЕТНЫМ целым, большим 1
%
if (Smax <= 1) | (Smax/2 == round(Smax/2)) | Smax ~= round(Smax)
    error('Smax должно быть нечетным целым больше 1')
end
[M, N] = size(g);
f = g;
f(:) = 0;
alreadyProcessed = false(size(g));
% Начинаем фильтрацию
for k = [ 3:2:Smax]
    zmin = ordfilt2(g, 1, ones(k, k), 'symmetric');
    zmax = ordfilt2(g, k*k, ones(k, k), 'symmetric');
    zmed = medfilt2(g, [k k], 'symmetric');
    processUsingLevelB = (zmed > zmin) & (zmax > zmed) & ...
        ~alreadyProcessed;
    zB = (g > zmin) & (zmax > g);
    outputZxy = processUsingLevelB & zB;
    outputZmed = processUsingLevelB & ~zB;
    f(outputZxy) = g(outputZxy);
    f(outputZmed) = zmed(outputZmed);
    alreadyProcessed = alreadyProcessed | processUsingLevelB;
    if all(alreadyProcessed(:))
        break;
    end
end
end
f(~alreadyProcessed) = zmed(~alreadyProcessed);

```

Теперь посмотрим, как работает адаптивная медианная фильтрация.

```

>> BW_n_amed3 = adpmedian(BW_n,3);
>> BW_n_amed5 = adpmedian(BW_n,5);
>> BW_n_amed7 = adpmedian(BW_n,7);
>> BW_n_amed15 = adpmedian(BW_n,15);
>> imshow(BW_n_amed3)
>> figure, imshow(BW_n_amed5)
>> figure, imshow(BW_n_amed7)

```

```
>> figure, imshow(BW_n_amed15)
```

Сравним с «обычной» медианной фильтрацией с квадратом 5×5 :

```
>> figure, imshow(BW_n_med5)
```

- ▶ При сравнении различных подходов полезно работать с одним и тем же тестовым изображением. Для этих целей удобно изображение «шахматной доски», генерируемое функцией `checkerboard`:

```
>> C = checkerboard(NP, M, N);
```

генерирует изображение «доски» с длиной клетки NP, числом строк M и числом столбцов N.

Построим зашумленное изображение смазанной шахматной доски:

```
>> f = checkerboard(8); % шахматная доска полями 8x8
>> PSF = fspecial('motion', 7, 45); % ФРТ смазывания на 7 точек
>> h = imfilter(f, PSF, 'circular'); % циклическая свертка с фильтром
>> noise = imnoise(zeros(size(f)), 'gaussian', 0, .001); % шум
>> g = h + noise; % зашумленная и смазанная "доска"
```

- ▶ Винеровская фильтрация реализуется функцией `deconvwnr`. Вызванная в виде `deconvwnr(g, PSF)`, она предполагает равным нулю отношение шум/сигнал изображения. В виде `deconvwnr(g, PSF, NSPR)` она полагает, что шум/сигнал равен константе (или массиву) NSPR. Команда вида `deconvwnr(g, PSF, NCORR, FCORR)` предполагает известными автокорреляционные функции NCORR шума и FCORR неискаженного изображения.

Попробуем применить винеровский фильтр к полученному изображению:

```
>> fr1 = deconvwnr(g, PSF); % простейшая фильтрация
>> Sn = abs(fft2(noise)).^2; % мощность шума
>> nA = sum(Sn(:))/prod(size(noise)); % средняя мощность шума
>> Sf = abs(fft2(f)).^2; % мощность изображения
>> fA = sum(Sf(:))/prod(size(f)); % средняя мощность изображения
>> R = nA/fA; % отношение шум/сигнал
>> fr2 = deconvwnr(g, PSF, R); % восстановление с учетом шума
>> NCORR = fftshift(real(ifft2(Sn))); % автокорреляционные функции
>> FCORR = fftshift(real(ifft2(Sf))); % - // -
>> fr3 = deconvwnr(g, PSF, NCORR, FCORR);
>> imshow(fr1)
>> figure, imshow(fr2)
>> figure, imshow(fr3)
>> figure, imshow(g)
```

Последнее изображение оказывается очень близким к идеалу, однако, не всегда известны все параметры, необходимые для точного восстановления. Поэтому зачастую приходится подбирать их экспериментально.

- ▶ Двумерные вейвлет преобразования реализуются в MATLAB при помощи уже известного вам Wavelet Toolbox. Можно сразу использовать консольные функции, а можно при помощи графического интерфейса пакета сделать предварительные оценки: выбрать тип вейвлетов для фильтрации, определить порядок фильтра и т.п. Откройте построенное изображение (предварительно сохранив его на диск) в окне *Wavelet 2D*. Поэкспериментируйте с различными вейвлетами, выбрав наиболее подходящий метод фильтрации.

6.3 Задания для самостоятельного выполнения

1. Откройте какое-либо изображение при помощи `imtool` (либо `imread`). Если это изображение цветное, приведите его к черно-белому виду. Назовите изображение `Im`. Постройте спектр изображения, назвав его, например, `Im_sp`. Осуществите логарифмическое преобразование к изображению и его спектру, отобразите полученное на экране.
2. Для пробного изображения постройте преобразования методом эквализации гистограммы с функциональными зависимостями для функций $\sin(x)$, $\cos(x)$, $\exp(x)$, $\exp(-x^2)$, $\ln(x)$ ($x = [1:256]$). Например, для первой функции это выполняется при помощи команды

```
>> Im_sin = histeq(Im, sin([1:256]));
```

3. Создайте копию изображения, зашумленную шумом «соль и перец» с вероятностью (`probab`), равной 0.07. Методом медианной фильтрации получите два очищенных изображения: в первом фильтрацию проводите по стандартному квадрату 3×3 , а во втором — по прямоугольнику 2×4 . Отобразите результаты на экране.
4. Наберите `m`-файлы, необходимые для осуществления фильтрации в частотной области. Постройте для изображения-образца фильтрованные копии с применением идеального НЧ фильтра, идеального ВЧ фильтра, НЧ и ВЧ фильтров Баттеюса, НЧ и ВЧ фильтров Гаусса (передаточные функции ВЧ и НЧ фильтров в сумме дают единицу, поэтому для ВЧ фильтра передаточная функция $HP = 1 - LP$, где LP — передаточная функция соответствующего НЧ фильтра).
5. Создайте `m`-файл `adpmedian.m`. Очистите свое зашумленное изображение при помощи адаптивного медианного фильтра. Сравните результат с медианной фильтрацией.

6. Создайте изображение шахматной доски с размером ячейки 20×20 пикселей. Смажьте изображение на 11 точек под углом 45° . Добавьте гауссова шума с математическим ожиданием 0 и среднеквадратичным отклонением 0.01. Отфильтруйте изображение при помощи простого винеровского фильтра и винеровского фильтра с учетом автокорреляционных функций. Сравните результаты.
7. Откройте зашумленное и смазанное изображение шахматной доски в окне *Wavelet Toolbox*. Попробуйте при помощи Вейвлет-анализа улучшить изображение, применяя к нему различные методы фильтрации разных порядков.

6.4 Контрольные вопросы

1. Как в математике рассматривается любое изображение? Чем серое изображение отличается от цветного?
2. Что такое цифровое изображение? Дайте определения основных характеристик цифрового изображения: разрядности, размера, масштаба.
3. Объясните физический смысл формулы (6.1), что означают входящие в него переменные?
4. Что такое преобразования изображения в пространственной области? Относятся ли к ним такие операции, как изменение масштаба, поворот, Фурье-фильтрация, размытие, корреляционные преобразования?
5. Что такое гистограмма? Как при помощи операций с гистограммой можно улучшить качество изображения?
6. Относятся ли нелинейные и дифференциальные фильтры к преобразованиям в пространственной области?
7. Дайте определение двумерного дискретного преобразования Фурье. Каков будет размер Фурье-спектра изображения размером $M \times N$ точек?
8. Что характеризует амплитуда Фурье-спектра изображения? Какой физический смысл несет фаза Фурье-спектра?
9. Чему равно значение Фурье-спектра изображения в точке с координатами $(0, 0)$?
10. Назовите основные недостатки двумерного дискретного преобразования Фурье.
11. Почему, несмотря на свои недостатки, Фурье-преобразования так активно используются при обработке изображений в частотной области?
12. Какой физический смысл несут функция рассеяния точки и оптическая передаточная функция; как эти две характеристики связаны между собой?
13. Под какую характеристику изображения адаптируется адаптивная меди-

- анная фильтрация?
14. Каковы достоинства и недостатки инверсной фильтрации?
 15. Какой аналог винеровской фильтрации существует в одномерной области?
 16. В чем заключается основная проблема распознавания изображений? Почему до сих пор нет автоматических систем, распознающих рукописные тексты?
 17. Что такое скелет изображения?
 18. Какие основные виды аффинных преобразований вы знаете?
 19. Сформулируйте основные шаги фильтрации в частотной области.

Задания для СКР

1 Краткие теоретические сведения

В 1977 году профессором Стэнфордского университета Д.Э. Кнудом была создана система профессиональной верстки $\text{T}_{\text{E}}\text{X}$ (произносится как «тех»). Создать эту систему его подтолкнуло полное отсутствие компьютерных систем верстки, позволяющих создавать математические тексты любой сложности, в которых он мог бы набирать свое знаменитое творение «Искусство программирования». Однако, писать тексты на чистом $\text{T}_{\text{E}}\text{X}$ е довольно сложно, поэтому в начале 80-х годов Л. Лампорт разработал макроязык на основе $\text{T}_{\text{E}}\text{X}$ и назвал его $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ (произносится как «латех»). Для работы с библиографией позднее было разработано такое расширение, как $\text{VibT}_{\text{E}}\text{X}$. Существует множество других расширений $\text{T}_{\text{E}}\text{X}$ а, но они имеют еще более узкую специализацию.

Исходные файлы $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ обычно имеют расширение `.tex`, однако, никто не заставляет вас работать именно с этим расширением, вы можете назвать файл как угодно, хоть `my.new.source`, главное — указать компилятору $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ имя именно этого файла. Вызов компилятора производится командой `latex`, в качестве параметра которой указывается имя файла с исходным кодом документа. Если параметр не указать, можно будет компилировать текст прямо из командной строки. Результатом компиляции является файл с расширением `.dvi`. Для подготовки текста к печати его необходимо преобразовать в PS или же PDF формат. При компиляции появляются также дополнительные файлы, содержащие данные о библиографическом указателе, предметном указателе, оглавлении, перекрестных ссылках.

Тело файла в $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ состоит из двух частей: **преамбулы**, в которой обозначены класс документа, его кодировка, язык, используемые шрифты, а также могут быть определены новые макрокоманды; и **тела документа**, заключенного в между командами `\begin{document}` и `\end{document}`. Обратный слеш перед комбинацией из нескольких букв или одной не-буквы (русские буквы, кстати, тоже относятся к «не-буквам») указывает, что данная последовательность является **командой**. **Аргументы** команды передаются далее в фигурных скобках (обязательные аргументы) или в квадратных скобках (необязательные аргументы). У многих команд есть *вариант со звездочкой*, например: `\section{Раздел}` и `\section*{Раздел}` (вариант со звездочкой в данном случае не включает раздел в сквозную нумерацию и в оглавление).

Главная задача $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ — упростить (да и сделать вообще возможным) набор и форматирование сложных документов. Поэтому, одним из его ценных свойств является *автоматизация рубрикации документа, перечней, перекрестных ссылок, подписей, библиографических ссылок, нумерации формул и многого, мно-*

того другого.

- ▶ *Самой первой командой преамбулы должно быть указание его класса.* Это осуществляется командой `\documentclass [опции]{класс}`, где опции (перечисляются через запятую) могут быть следующими:

`10pt`, `11pt`, `12pt` — указание размера шрифта данного документа (в пунктах);

`a4paper` — размер бумаги для форматирования страницы;

`twoside` — принудительная двухсторонняя печать (для односторонних классов);

`titlepage` — принудительное формирование титула на отдельной странице.

Кроме того, могут быть и другие опции. Класс документа должен быть одним из следующих:

`book` — книга (двухсторонняя печать, наиболее крупная рубрикация, заголовок на отдельной странице);

`report` — доклад (то же, что и книга, но нет понятия «Часть»);

`article` — статья (односторонняя печать, заголовок сразу над основным материалом, нет понятий «Часть», «Глава»);

`slide` — презентация (особое форматирование для создания мультимедийных презентаций).

За определением класса в преамбуле следуют определения новых макрокоманд пользователя и включения **пакетов**: `\usepackage{пакет}`. Пакет представляет собой перечень новых часто используемых макрокоманд (пользователь тоже может создать такой пакет, его расширением будет `.sty`). Главными пакетами являются `inputenc`, задающий входную кодировку документа, и `babel`, задающий язык документа (для автоматической расстановки переносов и указания правил расстановки пробелов между словами и знаками пунктуации).

- ▶ **Тело** документа содержит уже те последовательности символов и команд, которые будут непосредственно формировать выходной файл. Кроме того, в теле можно указывать и определения новых макрокоманд. Однако, следует помнить, что *локальная макрокоманда существует лишь внутри группы после момента ее объявления.* **Группой** является последовательность символов, заключенная в фигурные скобки, а также в скобки, создающие **окружения**: `\begin{что-то}... \end{что-то}`. То есть, тело документа тоже является своего рода группой. **Абзацы** внутри документа должны разделяться пустой строкой.

Также следует помнить, что, как и в любом языке программирования, в \LaTeX и \TeX есть понятие **комментария** — текста, не влияющего на работу компилятора. Комментарием является любой текст, размещенный в строке после символа процента.

Итак, исходный код минимального документа–статьи будет выглядеть примерно так:

```
\documentclass[12pt]{article} % Статья, кегль = 12 пунктов
\usepackage[koi8-r]{inputenc} % Входная кодировка - КОИ8-Р
\usepackage[russian, english]{babel} % Основных языка 2
\begin{document} % Начало тела документа
\section{Пробный раздел} % Раздел документа
Привет, мир!!! % эта надпись будет первым абзацем нашего раздела

% Пустая строка необходима для указания конца абзаца
Hello, world!!! % Можем напечатать и по-английски
% В конце документа необязательно вставлять пустую строку
\end{document} % Конец тела документа
```

Открыть dvi–файл можно при помощи какого-либо просмотрщика документов этого класса, например, `kdvi`, `xdvi` или `evince`. Для конвертирования dvi в ps используется команда `dvips`. Если указать ей лишь один параметр (название файла), сразу после конвертации постскрипт отправится на принтер по умолчанию (т.е. вывод команды напрямую перенаправляется на ввод команды печати `lpr`). Для сохранения постскрипт–файла необходимо указать параметр `-o файл.ps` после имени dvi–файла. Т.е., для получения ps–файла `a.ps` из исходного `a.tex` необходимо в простейшем случае выполнить в командной строке:

```
latex a && dvips a -o a.ps
```

(расширения файлов `dvi` и `tex` можно не указывать).

- ▶ Для создания заголовка документа используется команда `\maketitle`. Однако, для того, чтобы она знала, какую информацию необходимо поместить в заголовок, до ее использования нужно указать:

```
\author{А.В.~Тор1 \and А.В.~Тор2} % Авторы
\date{Дата} % то, что будет помещено как дата публикации
\thanks{Комментарии} % сноска внизу титульной страницы
\title{Название нашей публикации}
```

Обратите внимание на знак тильды (~): он обозначает **неразрывный пробел** (т.е. место, где нельзя разорвать строку).

Рубрикация документа осуществляется (в зависимости от его класса) командами: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph` для части, главы, раздела, подраздела, подподраздела, абзаца,

подабзаца. Если вам необходимо включить в конце текста приложение, его следует определять теми же командами рубрикации (т.е. `\chapter{}` или `\section{}` в зависимости от стиля документа), но до первого приложения указать команду `\appendix`, меняющую нумерацию частей на буквенные обозначения, а их название на «Приложение».

- Для указания **размера** шрифта используются команды `Huge`, `huge`, `LARGE`, `Large`, `large`, `normalsize`, `small`, `footnotesize`, `scriptsize`, `tiny`, например:

`Huge huge LARGE Large large norm small foot script tiny`

Изменение **начертания шрифта** производится командами: `\bf` (**жирный**), `\tt` (печатная машинка), `\sl` (*наклонный*), `\sc` (МАЛЫЕ ЗАГЛАВНЫЕ), `\it` (*курсив*), `\rm` (нормальный). Следует заметить, что перечисленные только что команды *действуют на весь текст, следующий за ними*, поэтому для возврата в нормальный режим необходимо либо воспользоваться командой `\rm`, либо заключить нужный участок текста в группу (между фигурными скобками, как окружение, либо как параметр другой команды). Кроме того, командой `\underline` можно подчеркнуть текст, а командой `\fbox` поместить текст в рамку.

Для оформления **перечней** чаще всего применяют окружения `itemize` (для маркированного перечня) и `enumerate` (для нумерованного перечня).

Л^AT_EX позволяет создавать автоматические перекрестные ссылки. Для того, чтобы создать ссылку на определенную часть публикации, нужно воспользоваться командой `\label{метка}`. В другом месте текста можно указать для ссылки на страницу с меткой команду `\pageref{метка}`, для ссылки на номер раздела (или пункта перечня) — `\ref{метка}`, а для ссылки на формулу — `\eqref{метка}` (в последнем случае номер формулы будет заключен в круглые скобки). Так как сведения о метках заносятся в специальный файл, а затем считываются из него, для правильного отображения ссылок *необходимо компилировать такой документ дважды*.

- **Формулы** в тексте делятся на *выключные* (стоящие отдельной строкой) и *внутритекстовые*. Внутритекстовые формулы обрамляются по краям знаками доллара (например, $\sin x=0$), а выключные — удвоенными знаками доллара (например, $\sin x=0$).

Верхние и нижние **индексы** задаются в Л^AT_EX при помощи символов циркумфлекса и подчеркивания, соответственно. При этом, если в индексе должно содержаться более одного символа, необходимо объединить их в группу. Например:

$$\int_0^{2\pi} \sin x \, dx = 0 \quad \Rightarrow \quad \int_0^{2\pi} \sin x \, dx = 0,$$

$$\frac{1+i}{1-i} = \frac{1+2i-1}{2} = i \Rightarrow \frac{1+i}{1-i} = \frac{1+2i-1}{2} = i,$$

(команда `\`, позволяет вставить **тонкий пробел**, необходимый для отделения дифференциала от подынтегральной функции).

Греческие буквы задаются командой, соответствующей английскому названию данной буквы, например: ρ – `\rho`, χ – `\chi`. Заглавные греческие задаются подобным образом: Σ – `\Sigma`, Ξ – `\Xi`.

Далее приведена краткая таблица–сводка основных математических функций и операций (за исключением простейших типа `\sin`).

<code>\sqrt{x}</code> $\Rightarrow \sqrt{x}$	<code>\sum A</code> $\Rightarrow \sum A$	<code>\prod A</code> $\Rightarrow \prod A$
<code>\vec{E}</code> $\Rightarrow \vec{E}$	<code>\frac{a}{b}</code> $\Rightarrow \frac{a}{b}$	<code>A\times B</code> $\Rightarrow A \times B$
<code>A\cdot B</code> $\Rightarrow A \cdot B$	<code>a\cdots b</code> $\Rightarrow a \cdots b$	<code>a\ldots b</code> $\Rightarrow a \dots b$
<code>\hat{A}</code> $\Rightarrow \hat{A}$	<code>a\le b\ge c</code> $\Rightarrow a \leq b \geq c$	<code>\ddot{y}</code> $\Rightarrow \ddot{y}$
<code>X\equiv Y</code> $\Rightarrow X \equiv Y$	<code>\rightarrow</code> $\Rightarrow \rightarrow$	<code>\dot{a}</code> $\Rightarrow \dot{a}$
<code>a\propto b</code> $\Rightarrow a \propto b$	<code>x\pm y</code> $\Rightarrow x \pm y$	<code>a\sim b</code> $\Rightarrow a \sim b$
<code>\partial z</code> $\Rightarrow \partial z$	<code>f\otimes g</code> $\Rightarrow f \otimes g$	<code>\infty</code> $\Rightarrow \infty$
<code>\Re y</code> $\Rightarrow \Re y$	<code>\Im y</code> $\Rightarrow \Im y$	<code>a\in b</code> $\Rightarrow a \in b$
<code>\int_a^b f(x)\,dx</code> $\Rightarrow \int_a^b f(x) dx$	<code>\oint_a^b f(x)\,dx</code> $\Rightarrow \oint_a^b f(x) dx$	
<code>\lim_{x\to 0} f(x)</code> $\Rightarrow \lim_{x \rightarrow 0} f(x)$	<code>\underbrace{x}_0</code> $\Rightarrow \underbrace{x}_0$	
<code>\overbrace{y}^1</code> $\Rightarrow \overbrace{y}^1$	<code>a^{b^c}</code> , <code>a^{b^c}^d</code> $\Rightarrow a^{b^{c^d}}, a^{b^{c^d}}$	

Принудительно задать промежутки можно при помощи команд:

<code>\,</code> \Rightarrow малый: <code><></code>	<code>\:</code> \Rightarrow средний: <code><></code>	<code>\;</code> \Rightarrow чуть больше: <code><></code>
<code>\quad</code> \Rightarrow большой: <code>< ></code>	<code>\quad\quad</code> \Rightarrow двойной: <code>< ></code>	<code>\!</code> \Rightarrow отрицательный: <code><></code>

Для ввода скобок переменной высоты можно использовать команды `\left`, `\right` и `\mid` для левых, правых и средних ограничителей, соответственно (если нужен только один ограничитель, необходимо ограничить область, к которой он относится командой `\right`. или `\left`.). Кроме того, можно и принудительно указать размер ограничителей командами `\bigl`, `\bigr`, `\bigrm`, ..., `\Bigl`, `\Bigr`, `\Biggr`.

$$\Bigl(\bigr[a\bigm|b\Bigr]\Biggr) \Rightarrow \left(\left[a \mid b \right] \right);$$

$$\left(\frac{\left(x^2+y_1\right)^3}{x\bigm|y^3}\right) \Rightarrow$$

$$\left(\frac{(x^2 + y_1)^3}{x | y^3} \right).$$

- Для задания **матриц и систем уравнений** существует несколько способов. Наиболее удобны способы, предлагаемые пакетами **AMS-LATEX**: `amsmath` (оформление формул), `amfonts` (специальные шрифты), `amssymb` (специальные символы).

Задать матрицу можно при помощи окружения `array`, для задания нумерованных формул следует использовать окружение `equation`:

```
\begin{equation}\begin{array}{ccc}
\sin x& 0& \cos x \\
0& \tg x & 0 \\
\operatorname{cosec} x & 0 & \operatorname{sec} x
\end{array} \\
\label{matrix}\end{equation}
```

$$\begin{array}{ccc} \sin x & 0 & \cos x \\ 0 & \operatorname{tg} x & 0 \\ \operatorname{cosec} x & 0 & \operatorname{sec} x \end{array} \quad (1)$$

Следует заметить, что при использовании окружения `equation` не надо использовать знаки доллара для ограничения формул. В окружении `array` вторым аргументом следует указать выравнивание столбцов с данными: `c` соответствует выравниванию посередине, `l` — по левому краю, `r` — по правому краю. Содержимое каждого столбца отделяется друг от друга знаком амперсанда, а конец строки обозначается знаком двойного обратного слеша (конец последней строки обозначать необязательно).

На выключную формулу, определенную таким образом, можно сослаться (1): определенную таким образом можно сослаться `\eqref{matrix}`.

Можно задать и системы уравнений при помощи окружения `array`:

```
$$\left\{ \begin{array}{rcl}
a_1 \cdot x^2 + b_1 \cdot x & = & c_1, \\
a_2 \cdot x^2 + b_2 \cdot x & = & c_2.
\end{array} \right. $$
```

Однако, в этом случае пробелы около знака равенства получаются не совсем правильными. Используя пакет `amsmath` воспользуемся окружением `aligned`:

```
$$\left\{ \begin{aligned}
a_1 \cdot x^2 + b_1 \cdot x & = c_1, \\
a_2 \cdot x^2 + b_2 \cdot x & = c_2.
\end{aligned} \right. $$
```

Сравните: слева приведен результат использования окружения `array`, а справа — `aligned`:

$$\begin{cases} a_1 \cdot x^2 + b_1 \cdot x = c_1, \\ a_2 \cdot x^2 + b_2 \cdot x = c_2. \end{cases} \quad \begin{cases} a_1 \cdot x^2 + b_1 \cdot x = c_1, \\ a_2 \cdot x^2 + b_2 \cdot x = c_2. \end{cases}$$

Задать кусочную формулу можно следующим образом:

```

 $\psi(x) \equiv \begin{cases} 1, & \text{если } 0 \leq x \leq 1/2; \\ -1, & \text{если } 1/2 < x \leq 1; \\ 0, & \text{в других случаях.} \end{cases}$ 

```

$$\psi(x) \equiv \begin{cases} 1, & \text{если } 0 \leq x \leq 1/2; \\ -1, & \text{если } 1/2 < x \leq 1; \\ 0, & \text{в других случаях.} \end{cases}$$

Если формула не вмещается в отдельную строку, ее можно задать при помощи окружения `multline`:

```

 $\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots - \frac{x^{97}}{97!} + \frac{x^{99}}{99!} - \frac{x^{101}}{101!} + \dots - \frac{x^{999}}{999!} + \frac{x^{1001}}{1001!}$ 

```

$$\begin{aligned} \sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \\ + \frac{x^{97}}{97!} - \frac{x^{99}}{99!} + \frac{x^{101}}{101!} - \dots \\ - \frac{x^{999}}{999!} + \frac{x^{1001}}{1001!} \end{aligned}$$

Окружения «со звездочкой» не приводят к автоматической генерации формул. Как видите, окружение `multline` выравнивает первую строку уравнения по левому краю, последнюю — по правому краю, а внутренние — посередине.

Если вы хотите выравнивать части формулы по какому-либо знаку, можно воспользоваться окружениями `align` или `split`. В каждой строке окружения `split` должно быть не более одного амперсанда, задающего точку выравнивания.

- Если вы хотите вставить в публикацию **графический файл**, его необходимо сохранить в формате *encapsulated postscript* (с расширением `.eps`). Экспортировать графики и изображения в такой формат умеет большинство специализированных программ (например, SciLab, MatLab, Gimp, QCad, Dia, OpenOffice).

Для того, чтобы вставить в данное место текста рисунок, воспользуйтесь командой

```
\includegraphics[width=xcm,height=ycm]{path/to/pict.eps}.
```

В необязательном параметре можно принудительно указать размер рисунка (а также некоторые другие его свойства), в обязательном параметре команды содержится путь к рисунку и его название. Учтите, что при указании пути к рисунку нельзя использовать сокращение `~` для домашней папки.

Чтобы задать плавающий рисунок, необходимо поместить его внутрь окружения `figure`, при помощи команды `\caption{name}` можно задать подпись к рисунку, а пометив его меткой (после команды `\caption`), можно ссылаться на данный рисунок в других частях текста. При создании плавающих объектов следует помнить, что они могут появиться не совсем там, где вы планировали, особенно, если документ перегружен такими объектами.

- Верстка **таблиц** осуществляется при помощи окружения `tabular`, а чтобы сделать плавающую таблицу, необходимо заключить ее в окружение `table`:

№п/п	Описание	ПО
1	Операционная система	Linux
2	Пакет профессиональной верстки	Л ^A T _E X
3	Свободный математический пакет	SciLab
4	Проприетарный математический пакет	MatLab

Эта таблица была сформирована при помощи команд:

```
\begin{tabular}{|c||c|c|}
\hline
\bf \No p/p& \bf Описание& \bf ПО\\
\hline\hline
1& Операционная система& Linux\\
2& Пакет профессиональной верстки& \LaTeX\\
3& Свободный математический пакет& SciLab\\
4& Проприетарный математический пакет& MatLab\\
\hline
\end{tabular}
```

Окружение `tabular` подобно окружению `array` за тем исключением, что оно позволяет создавать значительно более сложные таблицы. Горизонтальная линия таблицы задается при помощи команды `\hline`, а вертикальные определяются в преамбуле символом вертикальной черты.

- Для создания **нумерованных и маркированных списков** воспользуйтесь окружениями `enumerate` и `itemize` соответственно. Команда `\item` позволяет вставить в список следующий элемент. Если указать `\item[метка]`, вместо стандартной отметки для данного пункта будет использоваться *метка*. Например,

```
\begin{enumerate}
\item Первый пункт \item Второй пункт
\begin{itemize}
\item Первый подпункт \item Второй подпункт
\end{itemize}\item Третий пункт
\end{enumerate}
```

1. Первый пункт
2. Второй пункт
 - Первый подпункт
 - Второй подпункт
3. Третий пункт

Простейший **список литературы** можно сформировать при помощи окружения `thebibliography`. Каждый источник задается командой `\bibitem{метка}`, где *метка* используется в дальнейшем при цитировании. Вторым параметром окружения `thebibliography` является текстовый блок, определяющий максимальную ширину метки для обозначения источника. Например [1],

Список литературы

- [1] Сергиенко А.Б. Цифровая обработка сигналов. — СПб.: Питер, 2005. — 604 с.: ил.
- [2] У. Савич. Программирование на C++. 4-е изд. — СПб.: Питер; Киев: Издательская группа ВНУ, 2004. — 781 с.: ил.

Этот текст получен при помощи команд:

```
Например~\cite{Serg},\@[1em]
\begin{thebibliography}{99}
\bibitem{Serg} Сергиенко~А.Б. Цифровая обработка сигналов.~---
СПб.: Питер, 2005.~--- 604~с.: ил.
```



```
\bibitem{} У. Савич. Программирование на C++. 4-е изд. --- СПб.:  
Питер; Киев: Издательская группа ВНУ, 2004. --- 781 с.: ил.  
\end{thebibliography}
```

- Наконец, рассмотрим как создать **оглавление**. Если для обозначения разделов текста использовать специально предназначенные для этого команды (`\chapter`, `\section` и т.д.), сведения о всех подразделах документа заносятся при его компиляции в специальный файл, откуда могут быть считаны для формирования оглавления, списка рисунков или списка таблиц.

Оглавление создается командой `\tableofcontents`, список иллюстраций — `\listoffigures`, а список таблиц — `\listoftables`. Если вам необходимо внести в один из этих списков дополнительную информацию, воспользуйтесь командами `\addcontentsline{file}{depth}{text}`. Параметр `file` описывает, в какой из трех файлов следует внести информацию: в содержание (`toc`), список иллюстраций (`lof`) или список таблиц (`lot`). Параметр `depth` задает глубину вложения, как должен появиться текст в оглавлении (например, `part`, `chapter`,...). Параметр `text` и является текстом, который должен появиться в данном списке.

Обратите внимание, что при создании оглавления сведения о номерах страниц разделов берутся из `toc`-файла. Следовательно, при первом компилировании (до существования этого файла), оглавление будет пустым (в это время сведения о документе только заносятся в `toc`-файл). При втором компилировании каждому разделу из `toc`-файла сопоставляется страница, на которой он находится, ее номер и заносится в `toc`-файл рядом с наименованием раздела. Учитывая эту особенность \LaTeX а, *если оглавление расположено в конце документа, его необходимо компилировать дважды, если же оно располагается в начале документа — трижды*. Для надежности, если вы используете и оглавление, и перекрестные ссылки, лучше после каждого значительного изменения компилировать документ трижды.

2 Примеры практических заданий

2.1 Общее оформление текста в \LaTeX

Создайте новый текстовый файл, сохраните его с расширением `.tex`. Занесите в него стандартную преамбулу для статьи со шрифтом размера 12pt (см. стр. 82).

Задайте в преамбуле название документа «Лабораторные работы» и свое Ф.И.О. в качестве автора:

```
\title{Лабораторные работы}  
\author{Фамилия Имя Отчество}
```

В теле документа первой записью внесите

```
\section{Общее оформление текста}\subsection{Первое задание}
```

Наберите часть текста данного пособия со страницы 81:

```
1 \next\label{first}
2 Самой первой командой преамбулы должно быть указание его\ж класса\n. Это
3 осуществляется командой \verb'\documentclass[опции]{класс}',
4 где \verb'опции' (перечисляются через запятую) могут быть следующими:
5 \begin{itemize}
6 \item[\tt 10pt, 11pt, 12pt] --- указание размера шрифта данного документа
7 (в пунктах);
8 \item[\tt a4paper] --- размер бумаги для форматирования страницы;
9 \item[\tt twoside] --- принудительная двухсторонняя печать (для
10 односторонних классов);
11 \item[\tt titlepage] --- принудительное формирование титула
12 на отдельной странице.
13 \end{itemize}
14 Кроме того, могут быть и другие опции. Класс документа должен быть
15 одним из следующих:
16 \begin{itemize}
17 \item[\tt book] --- книга (двухсторонняя печать, наиболее крупная рубрикация,
18 заголовок на отдельной странице);
19 \item[\tt report] --- доклад (то же, что и книга, но нет понятия <<Часть>>);
20 \item[\tt article] --- статья (односторонняя печать, заголовок сразу над
21 основным материалом, нет понятий <<Часть>>, <<Глава>>);
22 \item[\tt slide] --- презентация (особое форматирование для создания
23 мультимедийных презентаций).
24 \end{itemize}
```

Откомпилируйте документ. Сравните результат с оригиналом.

► Теперь перед `\end{document}` добавьте таблицу со стр. 27:

```
1 \subsection{Второе задание}
2 Найдем общую среднюю совокупности, состоящей из следующих трех групп:
3 $$\renewcommand{\arraystretch}{0}
4 \begin{tabular}{|r|c|c|c|c|c|}
5 \hbox to 5cm{}&\hbox to 1.3cm{}&\hbox to 1.3cm{}&\hbox to 1.3cm{}&
6 \hbox to 1.3cm{}&\hbox to 1.3cm{}&\hbox to 1.3cm{}\\
7 \hline
8 \strut Группа&\multicolumn{2}{|c|}{I} &\multicolumn{2}{|c|}{II} &
9 \multicolumn{2}{|c|}{III} \\
10 \hline
11 \strut Значение признака&1&3&2&4
12 &3&6\\
13 \strut Частота признака&11&34&22&28&31&14\\\hline
```

```

14 \strut Объем выборки&\multicolumn{2}{|c|}{\$11+34=45\$}&
15 \multicolumn{2}{|c|}{\$22+28=50\$}&
16 \multicolumn{2}{|c|}{\$31+14=45\$}\\
17 \hline
18 \end{tabular}
19 $$

```

Как видите, знаками `$$` можно обозначать не только выключную формулу, но и любой другой объект, который должен быть отцентрован и отделен небольшими промежутками от текста (это может быть и текст, и рисунок, и таблица).

Команда `\renewcommand{\cmd}{someth}` предназначена для переопределения уже существующей команды `\cmd`. Для определения новой макрокоманды воспользуйтесь командой `\newcommand{\cmd}{someth}`, где `someth` — любой набор команд, который повторяется довольно часто, и вы хотели бы поместить его в обертку-заменитель `\cmd`. `\arraystretch` содержит интервал между ячейками таблицы.

Команда `\hbox{}` определяет текстовый блок (бокс), который является самодостаточной единицей и не может быть разорван. Указывая `\hbox to 1cm{}` мы определяем пустой бокс шириной 1 см (это необходимо для задания минимальной ширины столбцов).

Команда `\strut` предназначена для вставки «распорки», не позволяющей горизонтальным линиям наползать на текст.

Команда `\multicolumn{num}{format}{someth}` позволяет объединять соседние столбцы в текущей строке. `num` — число, определяющее количество объединенных столбцов, `format` — новый формат для объединенной ячейки, `someth` — то, что заносится в данную ячейку.

В самом начале тела документа добавьте оглавление и содержание:

```
\maketitle \tableofcontents
```

Откомпилируйте и сравните с результатом.

2.2 Оформление формул

В конце текста своего документа добавьте

```
\section{Оформление формул}\subsection{Первое задание}
```

Вставьте в текст кусочную формулу со стр. 86. Далее вставьте систему уравнений со стр. 37:

```

1 \begin{equation}
2 \left\{
3 \begin{aligned}
4 a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n&=b_1;\\
5 a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n&=b_2;\\
6 \cdots\\
7 a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n&=b_n.
8 \end{aligned}
9 \right.
10 \label{lin_eq}
11 \end{equation}

```

Сделайте где-нибудь в уже набранном тексте ссылку на эту систему при помощи команды `\eqref{lin_eq}`.

Задайте название подсекции «Второе задание» и наберите формулу со стр. 26:

```

1 $$
2 \gamma\equiv P(\bigl|\frac{\text{aver}\{x\}-a\}{S/\sqrt{n}}\bigr|<t_\gamma)=2\int_0^{t_\gamma}
3 S(t,n)\,dt,\quad
4 \delta=\frac{t_\gamma S}{\sqrt{n}},
5 $$

```

Наберите формулы со стр. 53:

```

$$\psi^{a,b}(x)=|a|^{-1/2}\psi\Bigl(\frac{x-b}{a}\Bigr).$$
$$W_\psi(f)(a,b)=\text{rev}\{\sqrt{a}\int_{-\infty}^{\infty}
f(t)\psi\Bigl(\frac{t-b}{a}\Bigr)\,dt.

```

Трижды откомпилируйте документ. Проверьте на правильность.

2.3 Общее оформление статьи

Зачастую стандартные стили документов ЛАТЭХ формируют страницу со размерами, не удовлетворяющими требованиям редакции научных журналов. В этом случае нужно указать в преамбуле документа размеры страницы и колонтитулов. Пусть мы хотим, чтобы страница нашего документа имела ширину текста, равную 17.5 см, высоту — 24 см, отступ верхнего колонтитула от края страницы 1.5 см, поля на нечетных страницах (при двойной печати) 2.2 см, а на четных — 1.3 см. В этом случае в преамбуле документа следует записать:

```

\textwidth=17.5cm
\oddsidemargin=-.3cm
\evensidemargin=-1.2cm
\topmargin=-1cm
\textheight=24cm

```

Не удивляйтесь, что в командах приведены другие числа. Дело в том, что `margin`-параметры отсчитываются от базовой линии, находящейся на расстоянии одного дюйма (2.54 см) от края страницы.

Научная статья обычно имеет следующую структуру:

- Название статьи.
- Авторы (с указанием учреждений, в которых они работают).
- Аннотация.

(Опционально) Ключевые слова.

(Опционально) Содержание.

- Текст статьи.
- Список литературы (оформленный по ЕСКД).

Итак, для оформления нашего документа как статьи остается добавить аннотацию и список литературы.

Для добавления аннотации воспользуемся окружением `abstract`, которое необходимо добавить между командами `\maketitle` и `\tableofcontents`:

```
1 \begin{abstract}
2 Наш материал посвящен тренировке для обучения {\bf основным} навыкам
3 работы с системой \LaTeX. Мы объясним, как сверстать правильно оформленную
4 научную статью.
5 \end{abstract}
```

В результате должно получиться следующее:

Аннотация

Наш материал посвящен тренировке для обучения **основным** навыкам работы с системой \LaTeX . Мы объясним, как сверстать правильно оформленную научную статью.

В конце документа добавим список литературы аналогично примеру со стр. 88. Добавьте на свой вкус еще 2–3 источника (например, из списка литературы данного пособия). Пометьте некоторые источники и сделайте на них ссылки в тексте «статьи».

3 Задания для самостоятельного выполнения

1. Создайте документ класса `report`, в котором опишите цели своего обучения в ВУЗе.
2. Наберите таблицу из $5 \div 7$ строк, в столбцы которой занесите:
 - Номер по порядку.

- Величайшее на ваш взгляд научное открытие.
- Ученого, благодаря которому произошел этот прорыв в науке.

(Опционально) Дату открытия.

Образец шапки таблицы:

№п/п	Открытие	Ученый	Дата

3. Наберите формулу (6.1) (стр. 64).

4. Наберите таблицу, в которой перечислите несколько известных неопределенностей (типа $\sin x/x$). В первом столбце запишите формулу, описывающую данную неопределенность, во второй — предел, к которому стремится аргумент функции при получении неопределенности, в третьей — значение предела.

Образец шапки таблицы:

$f(x)$	x_0	$\lim_{x \rightarrow x_0} f(x) =$
$\frac{\sin x}{x}$	0	1

5. Наберите формулу

$$\exp\left(\frac{\sin(x^2 - x_0^2)}{x_0^2}\right) \cdot \frac{1 + \frac{\sin x}{x}}{1 + \frac{1}{1 + \frac{1}{1 + \sin x}}} - \int_0^{\cos^2 x_0} \ln\left[1 + \frac{\operatorname{tg} x}{x_0} \cdot (1 + e^x)\right] dx.$$

Подсказка: для формирования цепной дроби используйте команду `\cfrac` расширения **AMS-L^AT_EX**.

6. Изучите литературу по **L^AT_EX**. Определите наиболее употребимые для физико-математических дисциплин команды и окружения **L^AT_EX**. Результат оформите в виде статьи, приведя в ее конце список использованной литературы (в т.ч. и интернет-источников). В статье при перечислении команд или окружений, описания которых взяты из определенного источника, до описания цитируйте источник.
7. Отформатируйте статью следующим образом: ее текст должен иметь ширину 17 см, высоту 22 см, располагаться на 4 см от верхнего и на 3 см от левого края страницы. Задайте полуторный межтекстовый интервал:

`\renewcommand{\baselinestretch}{1.5}`

Трижды откомпилируйте работу. Для защиты СКР предоставьте результаты в виде dvi-файлов и исходных текстов.

4 Контрольные вопросы

1. Какой минимальный набор макрокоманд необходимо знать для составления простейшего документа в \LaTeX ?
2. Какие основные классы документов вы знаете?
3. Какие команды \LaTeX используются для оформления разделов документа?
4. Как в \LaTeX задать нумерованные, маркированные и вложенные перечни?
5. Чем внутритекстовые формулы отличаются от выключных (сравните также особенности внешнего вида формул)?
6. Какие окружения применяются в \LaTeX для задания матриц, систем уравнений, кусочных формул?
7. Как в документ \LaTeX можно вставить графический файл?
8. Опишите, как используется окружение `tabular` для верстки таблиц.
9. Чем построение простейшего списка литературы отличается от построения нумерованного списка?
10. Опишите механизм создания оглавления в \LaTeX .
11. Объясните принцип формирования базы данных для заполнения перекрестных ссылок, библиографических ссылок и оглавления в документе: почему зачастую документ приходится теховать дважды или даже трижды?

Список литературы

Основная литература:

- [1] *Интернет-энциклопедия*: <http://wikipedia.org> (Википедия).
 - [2] Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. — М.: Техносфера, 2006 — 616 с.
 - [3] Гмурман В. Е. Теория вероятностей и математическая статистика. Учеб. пособие для вузов. — Изд. 7-е, стер. — М.: Высш. шк., 2001. — 479 с.
 - [4] Говорухин В., Цибулин В. Компьютер в математическом исследовании. Учебный курс. — СПб.: Питер, 2001. — 624 с.
 - [5] Сергиенко А. Б. Цифровая обработка сигналов. — СПб.: Питер, 2005. — 604 с.
 - [6] Чен К., Джигблин П., Ирвинг А. MATLAB в математических исследованиях: Пер. с англ. — М.: Мир, 2001. — 346 с.
- ### Дополнительная литература:
- [7] Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. — М.: Высш. шк., 1987. — 630 с.
 - [8] Кнут Д. Э. Все про $\text{T}_{\text{E}}\text{X}$./ Пер. с англ. М. В. Лисиной. — Протвино: АО $\text{RD}_{\text{E}}\text{X}$, 1993. — 592 с.: ил.
 - [9] Львовский С. М. Набор и верстка в системе $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. — 3-е изд., испр. и доп. — М.: МЦНМО, 2003. — 448 с.
 - [10] Pan G. W. Wavelets in electromagnetic and device modeling. — John Wiley & Sons, Inc., Hobocen, New Jersey, 2003. — 531 p.

Компьютерная обработка
результатов измерения
(учебное пособие)

Емельянов Эдуард Владимирович

Емельянов Эдуард Владимирович

**Компьютерная обработка
результатов измерения
(учебное пособие)**

Подписано в печать 6 ноября 2008 г. Гарнитура Computer Modern.
Формат 60 × 80 1/16. Тираж 5 экз. Цена договорная.

Отпечатано в домашней типографии Емельянова Э.В.



© Емельянов Э.В., 2008