

# Методические материалы по курсу «Компьютерная Физика»

---

*Семинары и задания для студентов 1-го курса*

**В. Н. Задков**

*Физический Факультет МГУ им. М.В.Ломоносова*

Москва ◊ 2000



## Глава 1

# Система компьютерной верстки $\LaTeX$ : Принципы работы и верстка текста

Дан краткий обзор принципов работы  $\LaTeX$ . Описана структура документов  $\LaTeX$  и верстка текстов<sup>1</sup>.

### 1.1 Названия: $\TeX$ и $\LaTeX$

$\TeX$  — это компьютерная программа, созданная Дональдом Кнудом (Donald E. Knuth) [?]. Она предназначена для верстки текста и математических формул. Кнут начал писать  $\TeX$  в 1977 году из-за расстройства от того, что Американское Математическое Общество делало с его статьями в процессе их публикации. Где-то в 1974 году он даже прекратил посылать статьи: «просто мне было слишком больно смотреть на конечный результат».  $\TeX$ , в том виде, в каком мы его сегодня используем, был выпущен в 1982 году и слегка улучшен с годами. Последние несколько лет  $\TeX$  стал чрезвычайно стабилен. Кнут утверждает, что в нем практически нет ошибок. Номер версии  $\TeX$  сходится к  $\pi$  и сейчас равен 3.14159.

$\TeX$  произносится как «тех». В среде ASCII  $\TeX$  нужно писать как `TeX`.

$\LaTeX$  (*LazyTeX*, или `TeX` для ленивых) — макропакет, позволяющий авторам верстать и печатать их работы с высоким типографским качеством при помощи заранее определенных, профессиональных макетов.  $\LaTeX$  был написан Leslie Lamport [?, ?]. В качестве механизма для верстки он использует  $\TeX$ .  $\LaTeX$  работает на большинстве компьютеров, начиная с IBM PC или Mac, и кончая большими системами UNIX или VMS. В университетских сетях система, как правило, уже установлена и готова к работе. В 1994 году предыдущая версия пакета  $\LaTeX$  2.09 была обновлена и стала называться  $\LaTeX$  2 <sub>$\epsilon$</sub> . Здесь мы рассматриваем именно  $\LaTeX$  2 <sub>$\epsilon$</sub> .

$\LaTeX$  произносится как «лэйтех» или как «латех». Если вы ссылаетесь на  $\LaTeX$  в ASCII окружении, пишите `LaTeX`.  $\LaTeX$  2 <sub>$\epsilon$</sub>  пишется как `LaTeX2 $\epsilon$` .

### 1.2 Где взять $\TeX$ / $\LaTeX$ ?

Если вам понадобится любой, относящийся к  $\LaTeX$ , материал, поищите его на одном из ftp архивов СТАН. Ближайший архив расположен на ftp-сервере НИИЯФ МГУ по адресу `ftp.radio-msu.net`.

---

<sup>1</sup> Данный семинар базируется на сокращенной версии руководства “Не очень краткое введение в  $\LaTeX$  2 <sub>$\epsilon$</sub> . Или  $\LaTeX$  2 <sub>$\epsilon$</sub>  за 84 минуты”, Т. Oetker, Н. Partl, I. Нуна, and E. Schlegl (пер. на русский язык Б. Тоботраса).

Для США это — `ctan.tug.org`, для Германии — `ftp.dante.de`, для Великобритании — `ftp.tex.ac.uk`. Если вы хотите установить L<sup>A</sup>T<sub>E</sub>X на вашем собственном компьютере, посмотрите на то, что доступно по адресу `CTAN:systems`.

## 1.3 Исходные файлы L<sup>A</sup>T<sub>E</sub>X

Исходными данными для L<sup>A</sup>T<sub>E</sub>X являются обычный текстовый файл в ASCII. Его можно создать в любом текстовом редакторе. Он содержит текст документа вместе с командами, указывающими L<sup>A</sup>T<sub>E</sub>X, как верстать текст.

### 1.3.1 Пробелы

«Пустые» символы, такие, как пробел или табуляция, трактуются L<sup>A</sup>T<sub>E</sub>X одинаково, как «пробел». *Несколько последовательных* пустых символов трактуются как *один* «пробел». Пустые символы в начале строки обычно игнорируются, а единичный перевод строки воспринимается как «пробел».

Пустая строка между двух строк текста определяет конец абзаца. *Несколько* пустых строк трактуются так же, как *одна* пустая строка. Ниже приведен пример. Справа — текст из входного файла, слева — форматированный вывод.

It doesn't matter how many spaces you include between words: one or several. An empty line starts new paragraph.	height
--	--------

### 1.3.2 Спецсимволы

Следующие символы являются зарезервированными символами, имеющими в L<sup>A</sup>T<sub>E</sub>X специальное значение:

\$ & % # \_ { } ~ ^ \

Если вы введете их в текст напрямую, то они обычно не напечатаются, а заставят L<sup>A</sup>T<sub>E</sub>X сделать что-нибудь, вами вовсе не предусмотренное. Как вы позже увидите, эти символы можно использовать в ваших документах, добавляя к ним префикс «\»:

\$ & % # _ { }	height
----------------	--------

Прочие символы, как и многие, многие другие, можно набрать специальными командами в математических формулах или как акценты. Знак «\» *нельзя* вводить, добавляя перед ним еще один, так как эта команда (\\) используется для разрыва строки<sup>2</sup>.

### 1.3.3 Команды L<sup>A</sup>T<sub>E</sub>X

Команды L<sup>A</sup>T<sub>E</sub>X чувствительны к регистру и имеют одну из двух форм:

- Начинаются с символа backslash «\» и продолжаются именем, состоящим только из букв. Имена команд завершаются пробелом, цифрой или любой другой «не-буквой».

<sup>2</sup>Вместо этого пользуйтесь командой `\backslash`. Она дает «\».

- Состоят из «\» и ровно одного специального символа.

Некоторые команды нуждаются в параметре, который должен быть задан между фигурными скобками «{ }» после имени команды. Некоторые команды имеют необязательные параметры, которые добавляются после имени команды в квадратных скобках «[ ]». Это будет разъяснено позже.

### 1.3.4 Комментарии

Когда в процессе обработки входного файла  $\text{\LaTeX}$  встречается символ %, он игнорирует остаток текущей строки, возврат каретки и все пробелы в начале следующей строки. Этим можно пользоваться для добавления в исходный файл замечаний, которые не будут выводиться на печать.

This is Spercalf<sup>height</sup>fragilisticexpialidocious

Знаком % можно также пользоваться, чтобы разбить длинные строки в тех местах, где не разрешаются пробелы или переводы строк.

## 1.4 Структура входного файла

Когда  $\text{\LaTeX}$  2<sub>ε</sub> обрабатывает входной файл, он ожидает от него определенной структуры. Так, каждый входной файл должен начинаться с команды

```
\documentclass{...}
```

Она указывает, документ какого типа вы собираетесь верстать. После этого, вы можете включать команды, влияющие на стиль документа в целом, или загружать пакеты, добавляющие новые возможности в систему  $\text{\LaTeX}$ . Для загрузки такого пакета используется команда

```
\usepackage{...}
```

Когда вся настройка закончена<sup>3</sup>, тело текста начинается командой

```
\begin{document}
```

Затем вводится текст с командами  $\text{\LaTeX}$ . В конце документа добавляется команда

```
\end{document}
```

Все, что следует после нее,  $\text{\LaTeX}$  игнорирует.

Рис. ?? показывает содержимое минимального файла для  $\text{\LaTeX}$  2<sub>ε</sub>. Несколько более сложный входной файл дан на рис. ??<sup>4</sup>.

<sup>3</sup>Область между `\documentclass` и `\begin{document}` называется *преамбулой*.

<sup>4</sup>Для того, чтобы документы могли содержать русские буквы, необходимо подключить пакет русификации  $\text{\LaTeX}$ . В настоящее время таких пакетов несколько и используемая вами русификация может отличаться от той, что используется в примере. В этом случае вместо пакета `amslatex` вам, возможно, понадобится подключать что-то другое. Ожидается, что в будущем `amslatex` будет стандартным средством локализации  $\text{\TeX}$ .

---

```

\documentclass{article}
\usepackage[russian]{babel}
\begin{document}
Краткость~--- сестра таланта.
\end{document}

```

---

Рис. 1.1: Минимальный файл L<sup>A</sup>T<sub>E</sub>X

## 1.5 Макеты документов

### 1.5.1 Классы документов

Первое, что L<sup>A</sup>T<sub>E</sub>X должен знать при обработке входного файла, это тип создаваемого автором документа. Он задается командой `\documentclass`.

`\documentclass [опции]{класс}`

Здесь *класс* определяет тип создаваемого документа. Таблица ?? перечисляет несколько основных классов документов. В состав L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> входят дополнительные классы для других документов, включая письма и слайды. Параметр *опции* изменяет поведение класса документа. Опции должны разделяться запятыми. В таблице ?? перечислены самые употребительные опции стандартных классов документов.

Таблица 1.1: Классы документов

---

<code>article</code>	для статей в научных журналах, презентаций, коротких отчетов, программной документации, приглашений. . .
<code>report</code>	для более длинных отчетов, содержащих несколько глав, небольших книжек, диссертаций. . .
<code>book</code>	для настоящих книг

---

Пример: Входной файл для документа L<sup>A</sup>T<sub>E</sub>X может, например, начинаться строкой

```
\documentclass[11pt,twoside,a4paper]{article}
```

Она заставляет L<sup>A</sup>T<sub>E</sub>X набирать документ как *статью*, с базовым размером шрифта *одинадцать пунктов* и форматировать документ для *двусторонней* печати на бумаге *формата A4*.

### 1.5.2 Пакеты

В процессе написания вашего документа, вы, вероятно, обнаружите, что в некоторых областях базовый L<sup>A</sup>T<sub>E</sub>X не сможет решить ваши проблемы. Если вы захотите включить в документ графику, цветной текст или исходный код программы из внешнего файла, вам нужно будет расширить

---

```

\documentclass[a4paper,11pt]{article}
\usepackage[cp866]{inputenc}
\usepackage[russian]{babel}
\begin{document}
\author{Б.~Тобограс}
\title{Минимализм}
\frenchspacing
\begin{document}
\maketitle
\tableofcontents
\section{\cyr{}}Начало
Вот тут и начинается моя замечательная статья.
\section{\cyr{}}Конец
\ldots{} а тут она кончается.
\end{document}

```

---

Рис. 1.2: Пример реалистичной журнальной статьи

Таблица 1.2: Опции классов документов

---

<code>10pt, 11pt, 12pt</code>	Устанавливает размер основного шрифта документа. Если ни одна из этих опций не указана, подразумевается <code>10pt</code> .
<code>a4paper, letterpaper...</code>	Определяет размер листа. По умолчанию подразумевается <code>letterpaper</code> .
<code>twocolumn</code>	Заставляет <code>Л<sup>A</sup>T<sub>E</sub>X</code> набирать документ в два столбца.
<code>twoside, oneside</code>	Выбирает одно- или двусторонний вывод. По умолчанию классы <code>article</code> и <code>report</code> используют односторонний вывод, класс <code>book</code> — двусторонний вывод. Заметьте, что опция <code>twoside</code> <i>не</i> заставляет ваш принтер на самом деле печатать с двух сторон.

---

возможности L<sup>A</sup>T<sub>E</sub>X. Такие расширения называются пакетами. Пакеты активизируются командой

```
\usepackage[опции]{пакет}
```

Где *пакет* — это имя пакета, а *опции* — список ключевых слов, включающих специальные свойства пакета. Огромное число пакетов можно найти на CTAN архивах [?].

### 1.5.3 Стили страницы

L<sup>A</sup>T<sub>E</sub>X поддерживает три предопределенных комбинации верхнего колонтитула и нижнего колонтитула — так называемые стили страницы. Параметр *стиль* команды

```
\pagestyle{стиль}
```

определяет, какой из них использовать. Предопределенные стили страницы перечислены в таблице ??.

Таблица 1.3: Предопределенные стили страницы L<sup>A</sup>T<sub>E</sub>X

<code>plain</code>	печатает номера страниц внизу страницы в середине нижнего колонтитула. Этот стиль установлен по умолчанию.
<code>headings</code>	печатает название текущей главы и номер страницы в верхнем колонтитуле каждой страницы, а нижний колонтитул остается пустым. (Этот стиль использован в данном документе.)
<code>empty</code>	делает и верхние, и нижние колонтитулы пустыми.

Можно сменить стиль текущей страницы командой

```
\thispagestyle{стиль}
```

## 1.6 Структура текста

Самый важный элемент текста в L<sup>A</sup>T<sub>E</sub>X (и в наборе вообще) — абзац. Он называется «элементом текста», потому что абзац является той типографской формой, в которую облачается одна связная мысль или идея. В следующих разделах вы узнаете, как можно вызвать перевод строки, например, при помощи `\\` и начало абзаца, например, оставив пустую строку. Поэтому, если начинается новая мысль, должен начинаться новый абзац, а в противном случае — используйте переводы строки.

Следующая, более маленькая, единица текста — предложение. В английском тексте после точки, завершающей предложение, ставится бóльший пробел, чем после точки, стоящей после



сокращения.  $\LaTeX$  старается определить, какую из них вы имели в виду. Если он ошибается, вы должны ему подсказать. Это объяснено далее в этой главе.

Наконец, абзацы текста должны быть также логично структурированы на более высоком уровне, объединяясь в главы, разделы, и так далее. Типографский эффект записи, например, `\section{The Структура текста и языка}` обычно достаточно очевиден, чтобы было сразу ясно, как использовать эти высокоуровневые структуры.

## 1.7 Разбиение на строки и страницы

### 1.7.1 Выровненные абзацы

Книги часто верстаются так, чтобы все строчки имели одинаковую длину.  $\LaTeX$  вставляет необходимые разрывы строк и пробелы между словами, оптимизируя форматирование абзаца как целого. При необходимости он также переносит слова, которые не помещаются на строке. От класса документа зависит, как верстаются абзацы. Обычно каждый абзац начинается с красной строки, а дополнительного интервала между ними нет.

В некоторых случаях может быть необходимо указать  $\LaTeX$  разорвать строку:

```
\\ или \newline
```

начинают новую строку, не начиная нового абзаца.

```
\newpage
```

начинает новую страницу.

$\LaTeX$  всегда пытается производить наилучшее из возможных разбиений строк. Если он не может найти способ разбить строки в соответствии со своими стандартами, он позволяет одной строке выступать из абзаца вправо.  $\LaTeX$  затем выводит диагностику («`overfull hbox`») во время обработки входного файла. Чаще всего это случается, когда  $\LaTeX$  не может найти место для переноса слова.

### 1.7.2 Переносы

$\LaTeX$  переносит слова, когда это необходимо. Если алгоритм переносов не находит правильных точек переноса, вы можете исправить положение, сказав об исключении при помощи следующих команд.

Команда `\-` вставляет в слово выборочный перенос. Он также становится единственной разрешенной точкой переноса в этом слове. Эта команда в особенности полезна для слов, содержащих специальные символы (например, символы с акцентами), потому что  $\LaTeX$  не переносит такие слова автоматически.

I think this is: supercalifragilisticexpial-  
idocious height

Несколько слов можно удержать вместе на одной строке командой

```
\mbox{текст}
```

Она будет в любом случае сохранять свой аргумент вместе.

My phone number will be changed height  
shortly. It should be 0116 291 2319.  
Parameter *file name* has to contain a  
file name.

## 1.8 Специальные буквы и символы

### 1.8.1 Знаки кавычек

Для набора кавычек вы *не должны* использовать знак ", как на пишущей машинке. При верстке существуют специальные знаки открывающей и закрывающей кавычек. В L<sup>A</sup>T<sub>E</sub>X используйте два знака ‘ в качестве открывающей кавычки и два знака ’ в качестве закрывающей.<sup>5</sup>

“Please, press key ‘x’ ” height

### 1.8.2 Тире и дефисы

L<sup>A</sup>T<sub>E</sub>X знает про четыре вида тире. Три из них вы можете получить различным числом последовательных знаков -. Четвертое на самом деле не тире вовсе, а математический знак минус:

one-to-two, X-rays height  
pages 13–67  
yes — or not?  
–1, 0, 1

Эти тире называются так: - дефис, -- короткое тире, --- длинное тире и \$-\$ знак минуса.

### 1.8.3 Многоточие ( ... )

Для многоточия есть специальная команда, называемаяся

```
\ldots
```

This is wrong: New-York, Tokyo, ... height  
This is correct: New-York, Tokyo, ...

---

<sup>5</sup>При верстке обычно требуются две пары кавычек, на случай «„кавычек“ в кавычках». При верстке на русском языке обычно вместо “таких” и “таких” кавычек используют «эти» и „эти“. Какими командами они задаются — зависит от используемой русификации, но обычно это `\flqq`, `\frqq` (вместо этой пары часто используют лигатуры « и ») и `\glqq`, `\grqq`.

### 1.8.4 Акценты и специальные символы

Л<sup>A</sup>T<sub>E</sub>X поддерживает использование акцентов и специальных символов из многих языков. Таблица ?? показывает всевозможные акценты в применении к букве o. Понятно, что на ее месте могут быть и другие буквы.

Чтобы поместить знак акцента над буквами i или j, точки над ними должны быть удалены. Это достигается набором \i и \j.

Hôtel, naïve, élève, height  
 smørrebrød, !'Señorita!,  
 Schönbrunner Schloß Straße

Таблица 1.4: Акценты и специальные символы

ò	\‘o	ó	\’o	ô	\~o	õ	\~o
ō	\=o	ò	\.o	ö	\"o	ç	\c c
ǒ	\u o	ǒ	\v o	ǒ	\H o	ø	\c o
o	\d o	o	\b o	oo	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	ł	\l	Ł	\L
ı	\i	İ	\j	!'	!'	?'	?'

## 1.9 Поддержка иностранных языков

Если вам нужно писать документы на отличных от английского языках, то Л<sup>A</sup>T<sub>E</sub>X должен быть соответствующим образом сконфигурирован при установке системы. Если ваша система уже соответствующим образом сконфигурирована, вы можете активизировать пакет babel добавлением команд

```
\usepackage[cp866]{inputenc} \usepackage[russian]{babel}
```

после команды \documentclass. Параметр cp866 пакета inputenc задает при этом кодировку текста в исходном файле (в данном случае это альтернативная кодировка MS DOS). Поддерживаются все основные языки (свыше 40) и кодировки.

### 1.10 Пробелы между словами

Знак «\» перед пробелом дает в результате пробел, который не будет увеличен. Знак «~» дает пробел, который не может увеличиться и который, кроме того, запрещает разрыв строки. Ко-

манда `\@` перед точкой указывает, что эта точка заканчивает предложение, несмотря на то, что стоит за буквой верхнего регистра.

```
Mr. Smith was happy to see her      height
cf. Fig. 5
I like BASIC. What about you?
```

Дополнительный пробел после точек можно запретить командой

```
\frenchspacing
```

которая указывает L<sup>A</sup>T<sub>E</sub>X не вставлять пробела после точки более, чем после обычных символов. Это обычно для языков, отличных от английского, за исключением библиографий. Если вы используете `\frenchspacing`, команда `\@` не нужна.

## 1.11 Интервалы

### 1.11.1 Интервалы между строками

Если вам нужны большие интервалы между строками, то их значение можно изменить помещением в преамбулу команды

```
\linespread{коэффициент}
```

Для печати «через полтора интервала» пользуйтесь `\linespread{1.3}`, для печати «через два интервала» — `\linespread{1.6}`. По умолчанию этот коэффициент равен 1.

### 1.11.2 Форматирование абзацев

Два параметра в L<sup>A</sup>T<sub>E</sub>X влияют на верстку абзацев. Поместив в преамбулу определения вида

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

вы измените внешний вид абзацев. Эти две строчки увеличивают расстояние между абзацами и устанавливают абзацный отступ равным нулю.

Если вы хотите сделать абзацный отступ в не имеющем его абзаце, то вставьте в начало абзаца команду

```
\indent
```

Понятно, что эффект от нее будет только если `\parindent` не установлен равным нулю.

Для создания абзаца без отступа первой командой абзаца можно сделать

```
\noindent
```

Это может быть удобно, когда вы начинаете документ с текста, а не с команды секционирования.

Таблица 1.5: Единицы размерности в T<sub>E</sub>X

mm	миллиметр $\approx 1/25$ дюйма	□
cm	сантиметр = 10 mm	□
in	inch = 25.4 mm	□
pt	пункт $\approx 1/72$ дюйма $\approx \frac{1}{3}$ mm	□
em	примерная ширина буквы 'M' текущего шрифта	□
ex	примерная высота буквы 'x' текущего шрифта	□

### 1.11.3 Горизонтальные интервалы

L<sup>A</sup>T<sub>E</sub>X автоматически определяет пробелы между словами и предложениями. Чтобы добавить горизонтальный пробел, пользуйтесь

```
\hspace{длина}
```

Если такой интервал должен быть выдержан, даже если он приходится на начало или конец строки, используйте `\hspace*`, а не `\hspace`. В простейшем случае *длина* — это просто число и единица измерения. Наиболее важные единицы перечислены в таблице ??.

Here is a  space in 1,5 cm.

Команда

```
\stretch{n}
```

генерирует специальный «резиновый» пробел. Он растягивается, заполняя все оставшееся места на строке. Если на одной строке встречаются две команды `\hspace{\stretch{n}}`, то они растягиваются пропорционально своим коэффициентам.

x  x  x  height

### 1.11.4 Вертикальные интервалы

Интервалы между абзацами, разделами, подразделами и т.п. определяются L<sup>A</sup>T<sub>E</sub>X автоматически. При необходимости дополнительный пробел *между двумя абзацами* можно добавить командой

```
\vspace{длина}
```

Обычно эта команда вставляется между двумя пустыми строчками. Если это пространство должно сохраняться вверху или внизу страницы, используйте вариант команды со звездочкой: `\vspace*`.

Команду `\stretch` вместе с `\pagebreak` можно применять для верстки текста на последней строке страницы или для вертикального центрирования текста на странице.

Некий текст\ldots

`\vspace{\stretch{1}}`

А это окажется на последней строке страницы.\pagebreak

Дополнительный пробел между двумя строками *одного* абзаца или внутри таблицы указывается командой

`\[длина]`

## 1.12 Заголовки, главы и разделы

Чтобы помочь читателю ориентироваться в вашей работе, вы должны разделять ее на главы, разделы и подразделы. L<sup>A</sup>T<sub>E</sub>X поддерживает это специальными командами, принимающими в качестве аргумента заголовки раздела. Ваше дело — использовать их в надлежащем порядке.

Класс `article` включает следующие команды секционирования:

<code>\section{...}</code>	<code>\paragraph{...}</code>
<code>\subsection{...}</code>	<code>\subparagraph{...}</code>
<code>\subsubsection{...}</code>	<code>\appendix</code>

В классах `report` и `book` вы можете использовать две дополнительные команды:

<code>\part{...}</code>	<code>\chapter{...}</code>
-------------------------	----------------------------

L<sup>A</sup>T<sub>E</sub>X создает оглавление, беря заголовки разделов и номера страниц из предыдущего цикла компиляции документа. Команда

`\tableofcontents`

вставляет оглавление в то место, где она вызвана. Чтобы получить правильное оглавление, новый документ должен быть обработан L<sup>A</sup>T<sub>E</sub>X дважды. В особых случаях может быть необходим и третий проход. Когда это потребуется, L<sup>A</sup>T<sub>E</sub>X вас предупредит.

Все вышеперечисленные команды секционирования существуют также в вариантах со звездочкой. Такой вариант получается добавлением `*` к имени команды. Они генерируют заголовки разделов, которые не нумеруются и не включаются в оглавление. Например, команда `\section{Справка}` становится `\section*{Справка}`.

Титульный лист документа генерируется при помощи команды

`\maketitle`

Его содержимое должно быть определено командами

`\title{...}, \author{...} и \date{...}`

до момента вызова `\maketitle`. Аргумент команды `\author` может содержать несколько имен, разделенных командами `\and`.

Пример некоторых из упомянутых команд может быть найден на иллюстрации ??.

## 1.13 Перекрестные ссылки

В книгах, отчетах и статьях часто встречаются перекрестные ссылки на иллюстрации, таблицы и отдельные части текста. Для этого ЛАТЭХ предоставляет следующие команды:

```
\label{метка}, \ref{метка} и \pageref{метка}
```

где *метка* — выбранный пользователем идентификатор. ЛАТЭХ заменяет `\ref` номером раздела, подраздела, иллюстрации, таблицы или уравнения, где была использована соответствующая команда `\label`. `\pageref` печатает номер страницы, соответствующей команде `\label`<sup>6</sup>. Так же, как и в случае с заголовками разделов, здесь также используются номера из предыдущего прохода.

Reference to this section looks as height  
follows: «see. section ?? on page ??.»

## 1.14 Сноски

Команда

```
\footnote{текст сноски}
```

печатает сноску внизу текущей страницы. Сноски всегда должны помещаться после слова, к которому они относятся:

ЛАТЭХ users frequently use footnotes<sup>a</sup>. height

---

<sup>a</sup>This is a footnote.

## 1.15 Выделенные слова

В рукописи, напечатанной на машинке, **важные слова выделяются подчеркиванием**. В печатных изданиях эти слова выделяются *курсивом*. Команда для переключения на шрифт *выделения* называется

```
\emph{текст}
```

Ее аргументом является текст для выделения. Что на самом деле делает эта команда, зависит от контекста:

If you *emphasize the text that has* height  
*been already emphasized, ЛАТЭХ uses*  
roman *font*.

---

<sup>6</sup>Заметьте, что эти команды не знают, на что именно они ссылаются. `\label` просто сохраняет последний автоматически генерируемый номер.

## 1.16 Шрифты и их размеры

Л<sup>A</sup>T<sub>E</sub>X выбирает подходящее начертание и размер шрифта, основываясь на логической структуре документа (разделы, сноски, ...). Иногда может быть желательно сменить шрифт вручную. Для этого вы можете пользоваться командами, перечисленными в таблицах ?? и ?. Действительный размер каждого шрифта определяется дизайном и зависит от класса и опций документа. Таблица ?? показывает абсолютные размеры, соответствующие этим командам в стандартных классах документов.

The small and **bold** Romans ruled all of <sup>height</sup> great big *Italy*.

Таблица 1.6: Шрифты

<code>\textrm{...}</code>	прямой шрифт	<code>\textsf{...}</code>	без засечек
<code>\texttt{...}</code>	пишущая машинка		
<code>\textmd{...}</code>	нормальный	<code>\textbf{...}</code>	<b>полужирный</b>
<code>\textup{...}</code>	прямой шрифт	<code>\textit{...}</code>	<i>курсив</i>
<code>\textsl{...}</code>	<i>наклонный шрифт</i>	<code>\textsc{...}</code>	КАПИТЕЛЬ
<code>\emph{...}</code>	<i>выделенный шрифт</i>	<code>\textnormal{...}</code>	обычный

Таблица 1.7: Размеры шрифта

<code>\tiny</code>	крошечный	<code>\Large</code>	еще больше
<code>\scriptsize</code>	очень маленький	<code>\LARGE</code>	очень большой
<code>\footnotesize</code>	довольно маленький	<code>\huge</code>	огромный
<code>\small</code>	маленький	<code>\Huge</code>	громадный
<code>\normalsize</code>	нормальный		
<code>\large</code>	большой		

В связи с командами смены размера шрифта заметную роль играют фигурные скобки. Они используются для построения *групп*. Группы ограничивают область действия большинства команд Л<sup>A</sup>T<sub>E</sub>X:

He likes <sup>height</sup> large and <sub>small</sub> characters.

Команды, влияющие на размер шрифта, влияют также на расстояние между строками, но только если соответствующий абзац заканчивается внутри области действия команды. Поэтому



Таблица 1.8: Абсолютные размеры шрифтов в стандартных классах

<i>Размер</i>	<i>10pt (по умолчанию)</i>	<i>опция 11pt</i>	<i>опция 12pt</i>
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

закрывающая фигурная скобка } не должна стоять слишком рано. Заметьте положение команды `\par` в следующих двух примерах:

Do not read this! This is <sup>height</sup>  
rubbish. Trust me!

This is also rubbish. <sup>height</sup>

Если вы хотите применить команду изменения размера к целому абзацу текста или больше того, то для этого лучше использовать синтаксис окружения:

This is not truth. But what is <sup>height</sup>  
truth nowadays...

Это избавит вас от подсчета множества фигурных скобок.

Для завершения нашего путешествия в мир шрифтов и их размеров, позвольте дать вам один совет:

**Помните!** Чем Б ЛЬ шрифтов ВЫ используете в вашем документе, тем легче ЕГО  
ЧИТАТЬ и тем он будЕТ.

## 1.17 Окружения

Для верстки специальных видов текста ЛАТЭХ определяет множество окружений для разных типов форматирования:

```
\begin{название} текст \end{название}
```

где *название* определяет окружение. Окружения можно вызывать внутри окружений, соблюдая порядок вызова и возврата:

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

В следующих разделах рассказывается обо всех важных окружениях.

### 1.17.1 Список, перечисление и описание

Окружение `itemize` подходит для простых списков, окружение `enumerate` — для нумерованных списков, а окружение `description` — для описаний.

height

1. You can mix lists:
  - But this could look not so good.
  - With minus.

2. Therefore, remember:

**Foolish** stuff will not be clever if you put them in a list.

**Clever** stuff, however, you could typeset in the list.

### 1.17.2 Выравнивание влево, вправо и по центру

Окружения `flushleft` и `flushright` форматируют абзацы, выровненные влево или вправо. Окружение `center` дает центрированный текст. Если вы не используете `\` для указания разрывов строк, L<sup>A</sup>T<sub>E</sub>X определит их автоматически.

This text  
is left-aligned. L<sup>A</sup>T<sub>E</sub>X does not try to  
make all the lines of the same length.

height

This text  
is right-aligned. L<sup>A</sup>T<sub>E</sub>X does not try to  
make all the lines of the same length.

height

In the center of  
Earth

height

### 1.17.3 Буквальное воспроизведение

Текст, заключенные между `\begin{verbatim}` и `\end{verbatim}` будет напрямую напечатан, как набранный на пишущей машинке, со всеми пробелами и возвратами каретки, без выполнения каких бы то ни было команд L<sup>A</sup>T<sub>E</sub>X.

Внутри абзаца аналогичную функцию выполняет команда

```
\verb+текст+
```

Здесь «+» — это только пример символа-ограничителя. Вы может использовать любой символ, кроме букв, «\*» или пробела. Многие примеры на L<sup>A</sup>T<sub>E</sub>X в данном документе набраны этой командой.

Command `\ldots` ... height

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

`verbatim_environment` height  
`with_asterik_show`  
`spaces_in_the_text`

#### 1.17.4 Таблицы

Окружение `tabular` используют для верстки таблиц, возможно, с горизонтальными и вертикальными линиями. L<sup>A</sup>T<sub>E</sub>X автоматически определяет ширину столбцов.

Аргумент *спецификация* команды

```
\begin{tabular}{спецификация}
```

определяет формат таблицы. Используйте `l` для столбца текста, выровненного влево, `r` для текста, выровненного вправо и `c` для центрированного текста, `r{ширина}` для столбца, содержащего выровненный текст с переносом строк, и `|` для вертикальной линии.

Внутри окружения `tabular` знак «&» переходит к следующему столбцу, команда `\\` начинает новую строку, а `\hline` вставляет горизонтальную линию:

height

54	hex
124	octal
1010100	binary
84	decimal

height

Welcome to the paragraph bounded by a box. How do you feel yourself here?
---

## 1.18 Плавающие объекты

Иллюстрации или таблицы являются т.н. плавающими объектами L<sup>A</sup>T<sub>E</sub>X, поскольку не уместяющаяся на текущей странице таблица или рисунок, может ‘плавать’, перемещаясь на следующую страницу в процессе заполнения текстом текущей. L<sup>A</sup>T<sub>E</sub>X предлагает для плавающих объектов два окружения — для таблиц и иллюстраций.

Любой материал, включенный в окружения `figure` или `table`, трактуется как плавающий. Оба окружения имеют необязательный параметр

```
\begin{figure}[спецификация размещения] или
\begin{table}[спецификация размещения]
```

называемый *спецификацией размещения*. Этот параметр используется для указания L<sup>A</sup>T<sub>E</sub>X, куда можно перемещать плавающий объект. *Спецификация размещения* конструируется путем собирания в строчку *ключей размещения плавающего объекта* (табл. ??).

Таблица 1.9: Ключи размещения плавающего объекта

Ключ	Разрешает помещать объект ...
<code>h</code>	<i>здесь же</i> , в том самом месте текста, где он появился. Обычно используется для маленьких объектов.
<code>t</code>	<i>наверху</i> страницы
<code>b</code>	<i>внизу</i> страницы
<code>p</code>	на <i>специальной странице</i> , содержащей только плавающие объекты.

Например, таблицу можно начать следующей строкой:

```
\begin{table}[hbp]
```

Спецификация размещения `[hbp]` позволяет L<sup>A</sup>T<sub>E</sub>X разместить таблицу прямо по месту (`h`), или внизу той же страницы (`b`), или на выделенной странице (`p`). Если никакой спецификации размещения не задано, стандартные классы предполагают `[tbp]`.

Командой

```
\caption{текст заголовка}
```

вы можете задать заголовок для объекта. Увеличивающийся номер и строка «Рисунок» или «Таблица» добавляются L<sup>A</sup>T<sub>E</sub>X.

Две команды

```
\listoffigures и \listoftables
```

работают аналогично команде `\tableofcontents`, печатая список иллюстраций или таблиц, соответственно.

При помощи `\label` и `\ref` можно делать ссылки из вашего текста на плавающий объект.

Следующий пример рисует квадрат и вставляет его в документ.

```

Рисунок~\ref{white} является примером Поп-Арта.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Пять на пять сантиметров} \label{white}
\end{figure}
\\

```

## 1.19 Включение EPS графики

Л<sup>A</sup>T<sub>E</sub>X имеет базовые средства для работы с плавающими объектами, такими, как иллюстрации и таблицы, при помощи рассмотренных нами окружений `figure` и `table`.

Существует также несколько способов создавать собственно графику средствами базового Л<sup>A</sup>T<sub>E</sub>X или его расширений. К сожалению, большинство пользователей находит этот способ сложным и не универсальным, поэтому мы его здесь не рассматриваем. Более простой метод получения графики в документе — это создавать ее специализированными программными пакетами<sup>7</sup> и включать в документ готовую графику. Пакеты Л<sup>A</sup>T<sub>E</sub>X предлагают множество способов это делать. Здесь обсуждается только использование графики в формате Encapsulated POSTSCRIPT (EPS), поскольку это довольно просто делается и широко распространено. Чтобы использовать картинку в формате EPS, вам нужно использовать для вывода POSTSCRIPT принтер<sup>8</sup>.

Хороший набор команд включения графики входит в пакет `graphicx`. Он является частью целого семейства пакетов, называющегося комплектом “`graphics`”<sup>9</sup>. В предположении, что вы работаете в системе с доступным для вывода POSTSCRIPT принтером, и с установленным пакетом `graphicx`, для включения картинки в ваш документ вы можете использовать следующую пошаговую инструкцию:

1. Экпортируйте картинку из вашей графической программы в формате EPS.<sup>10</sup>
2. В преамбуле документа загрузите пакет `graphicx` при помощи

```
\usepackage[драйвер]{graphicx}
```

где *драйвер* — это название вашего конвертера DVI в POSTSCRIPT. Самый широко используемый называется `dvips`. Он используется по умолчанию.

3. Для включения *файла* в ваш документ пользуйтесь командой

```
\includegraphics[ключ=значение, ...]{файл}
```

Необязательный параметр принимает вид списка разделенных запятыми пар *ключей* и соответствующих *значений*. *Ключи* можно использовать для изменения ширины, высоты, поворота включаемой графики. Таблица ?? перечисляет основные ключи.

<sup>7</sup>Таковыми, как Origin, CorelDraw, Freehand, Gnuplot, ...

<sup>8</sup>Другая возможность вывода POSTSCRIPT заключается в использовании программы GHOSTSCRIPT, доступной с CTAN: `support/ghostscript`.

<sup>9</sup>CTAN: `macros/latex/packages/graphicx`.

<sup>10</sup>Если ваша программа не имеет экспорта в EPS, можно попробовать установить драйвер PostScript принтера (скажем, какого-нибудь Apple LaserWriter) и печатать в файл через этот драйвер. Альтернативно, можно использовать программу GHOSTSCRIPT. Заметьте, что EPS обязан содержать не больше одной страницы.

Таблица 1.10: Названия ключей для пакета `graphicx`

---

<code>width</code>	масштабирует графику до указанной ширины
<code>height</code>	масштабирует графику до указанной высоты
<code>angle</code>	поворачивает графику против часовой стрелки
<code>scale</code>	масштабирует графику

---

Возможно, пример сделает это понятнее:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test.eps}
\end{center}
\end{figure}
```

Здесь включается рисунок, записанный в файл `test.eps`. Он *сначала* поворачивается на 90 градусов и *затем* масштабируется до конечной ширины в 0.5 ширины стандартного абзаца. Пропорции сохраняются, поскольку не указано конкретной высоты. Параметры высоты и ширины могут также быть указаны в абсолютных размерах.

## 1.20 Библиография

Окружение `thebibliography` генерирует библиографию. Каждый элемент начинается с

```
\bibitem{маркер}
```

Затем *маркер* используется для дальнейших ссылок на книгу, статью или труд:

```
\cite{маркер}
```

Элементы библиографии нумеруются автоматически. Максимальную ширину этих номеров устанавливает параметр после команды `\begin{thebibliography}`. В следующем примере {99}

указывает  $\text{\LaTeX}$ , что ни один из номеров элементов не будет шире, чем число ‘99’.

Partl [?] suggested that... height

## Литература

- [1] H. Partl: *German  $\text{\TeX}$* , TUGboat  
Vol. 9, No. 1 ('88)

## Глава 2

# Система компьютерной верстки $\text{\LaTeX}$ : Набор математических формул

Рассмотрена верстка формул и других математических выражений в  $\text{\LaTeX}$ . Хотя здесь дан только поверхностный обзор, для большинства задач изложенных здесь вещей будет достаточно<sup>1</sup>.

### 2.1 Общие сведения

$\text{\LaTeX}$  включает в себя специальный режим для верстки математики. Математический текст внутри абзаца вводится между  $\backslash($  и  $\backslash)$ , между  $\$$  и  $\$$  или между  $\backslash\text{\begin{math}}$  или  $\backslash\text{\end{math}}$ :

Adding  $a$  squared with  $b$  squared, we  
receive  $c$  squared. This can be written as  
follows:  $c^2 = a^2 + b^2$

Большие математические уравнения или формулы предпочтительнее «выключать», то есть верстать их на отдельных строчках. Для этого заключайте их между  $\backslash[$  и  $\backslash]$  или между  $\backslash\text{\begin{displaymath}}$  и  $\backslash\text{\end{displaymath}}$ . Это дает вам не нумерованные формулы:

Adding  $a$  squared with  $b$  squared, we  
receive  $c$  squared. This can be written as  
follows:

$$c^2 = a^2 + b^2.$$

This is another line...

Если вы хотите, чтобы  $\text{\LaTeX}$  их нумеровал, пользуйтесь окружением `equation`. При помощи `\label` и `\ref` вы можете в тексте ссылаться на уравнение:

$$\epsilon > 0 \tag{2.1}$$

From (??) follows ...

---

<sup>1</sup>Данный семинар базируется на сокращенной версии руководства “Не очень краткое введение в  $\text{\LaTeX} 2_{\epsilon}$ . Или  $\text{\LaTeX} 2_{\epsilon}$  за 84 минуты”, Т. Oetker, Н. Partl, I. Нупа, and E.Schlegl (пер. на русский язык Б. Тоботраса).



Заметьте, что выражения в выключных формулах верстаются по-разному:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad \text{height}$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad \text{height}$$

*Математический режим* отличается от *текстового режима*. В *математическом режиме* все пробелы должны в явном виде задаваться командами вроде `\,`, `\quad` или `\qquad`. *Пустые строки недопустимы*. Каждая формула занимает только один абзац. Каждая буква считается именем переменной и верстается в этом качестве. Если вы хотите ввести в формулу нормальный текст (нормальный прямой шрифт с нормальными пробелами), то вам нужно ввести его командами `\text{trm}{...}`:

$$x^2 \geq 0 \quad \text{for all } x \in \mathbf{R} \quad (2.2) \quad \text{height}$$

## 2.2 Группировка в математическом режиме

Большинство команд математического режима действует только на следующий символ. Так что, если вы хотите, чтобы команда влияла на несколько символов, вам нужно сгруппировать их вместе при помощи фигурных скобок `{...}`:

$$a^x + y \neq a^{x+y} \quad (2.3) \quad \text{height}$$

## 2.3 Составляющие математической формулы

Здесь мы рассмотрим наиболее важные команды, используемые в математической верстке. Детальный перечень команд для набора математических символов приведен в разделе ?? на странице ??.

**Строчные греческие буквы** вводятся как `\alpha`, `\beta`, `\gamma`, ..., прописные буквы вводятся как `\Gamma`, `\Delta`, ...:

$$\lambda, \xi, \pi, \mu, \Phi, \Omega \quad \text{height}$$

**Верхние и нижние индексы** вводятся при помощи символов `^` и `_`:

$$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3 \quad \text{height}$$

$$e^{x^2} \neq e^{x^2}$$

**Квадратный корень** вводится как `\sqrt`, корень  $n$ -ной степени печатается при помощи `\sqrt[n]`. Размер знака корня выбирается ЛАТЭХ автоматически:

$$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2} \quad \text{height}$$

$$\sqrt{[x^2 + y^2]}$$

Команды `\overline` и `\underline` создают **горизонтальные линии** сразу над или под выражением:

$$\overline{m+n} \quad \text{height}$$

Команды `\overbrace` и `\underbrace` создают длинные **горизонтальные фигурные скобки** сразу над или под выражением:

$$\underbrace{a+b+\dots+z}_{26} \quad \text{height}$$

Для добавления к переменным знаков математических акцентов, таких, как маленькие стрелки или знака тильда, вы можете пользоваться командами, перечисленным в таблице ???. Широкие «шляпки» и тильды, охватывающие несколько символов, генерируются командами `\widetilde` и `\widehat`. Символ «'» дает знак производной.

$$y = x^2 \quad y' = 2x \quad y'' = 2 \quad \text{height}$$

**Векторы** часто указываются добавлением маленьких стрелок стрелки над переменной. Это делается командой `\vec`. Для обозначения вектора от  $A$  до  $B$  полезны две команды `\overrightarrow` и `\overleftarrow`:

$$\vec{a} \quad \overrightarrow{AB} \quad \text{height}$$

**Имена функций** типа `lg` часто верстаются прямым шрифтом, а не курсивом, как переменные. Поэтому L<sup>A</sup>T<sub>E</sub>X содержит следующие команды для набора имен наиболее важных функций:

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>	<code>\min</code>	<code>\sinh</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>	<code>\Pr</code>	<code>\sup</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>	<code>\sec</code>	<code>\tan</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\sin</code>	<code>\tanh</code>

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \quad \text{height}$$

Двухъярусная **дробь** верстается командой `\frac{...}{...}`. Часто предпочтительнее ее форма с косой чертой  $1/2$ , потому что она смотрится лучше для коротких дробей:

$$1\frac{1}{2} \text{ hours} \quad \text{height}$$

$$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

**Оператор интеграла** печатает команда `\int`, **оператор суммы** — команда `\sum`. Верхние и нижние пределы указываются при помощи знаков «`^`» и «`_`», так же, как верхние и нижние индексы:

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}}$$

Для **скобок** и прочих ограничителей в Т<sub>Е</sub>X существует множество символов (скажем, `[` `<` `||` `↑`). Круглые и квадратные скобки можно вводить соответствующими клавишами, фигурные скобки — `\{`, прочие ограничители — специальными командами (например, `\updownarrow`). Список доступных ограничителей смотрите в таблице ?? на странице ??.

$$a, b, c \neq \{a, b, c\}$$

Если вы поместите перед открывающим ограничителем команду `\left`, или перед закрывающим — `\right`, то Т<sub>Е</sub>X автоматически выберет правильный размер ограничителя. Заметьте, что вы должны каждый `\left` закрывать соответствующим `\right`, и что размер определяется корректно только если оба они набраны на одной строке. Если вы не хотите иметь одного из ограничителей, используйте невидимый ограничитель `\right.` или `\left.`:

$$1 + \left( \frac{1}{1-x^2} \right)^3$$

Чтобы ввести в формулу **многоточие**, есть несколько команд. `\ldots` верстает точки на базовой линии, `\cdots` — центрированные. Кроме того, существуют команды `\vdots` для вертикальных и `\ddots` для диагональных точек.

$$x_1, \dots, x_n \quad x_1 + \cdots + x_n$$

## 2.4 Математические пробелы

Если выбранные Т<sub>Е</sub>X пробелы внутри формул неудовлетворительны, вы можете их подстраивать с использованием команд управления пробелами. Команды для маленьких пробелов: `\`, для  $\frac{3}{18}$  quad (U), `\:` для  $\frac{4}{18}$  quad (U) и `\;` для  $\frac{5}{18}$  quad (U). Экранированный символ пробела `\_` дает средних размеров пробел, а `\quad` (U) и `\qqquad` (U) дают большие пробелы. Размер `\quad` примерно соответствует ширине буквы 'M' в текущем шрифте. Команда `\!` производит отрицательный пробел размером  $-\frac{3}{18}$  quad (U):

$$\iint_D g(x, y) dx dy$$

instead

$$\int \int_D g(x, y) dx dy$$

Заметьте, что ‘d’ в дифференциале обычно верстается прямым шрифтом.

## 2.5 Вертикально расположенный материал

Для верстки **матриц** пользуйтесь окружением `array`. Его работа напоминает окружение `tabular`. Для разрыва строки используется команда `\\`.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \text{height}$$

Окружение `array` можно также использовать для верстки выражений, имеющих один большой ограничитель, подставляя «. » в качестве невидимого правого ограничителя:

$$y = \begin{cases} a & \text{if } d > c \\ b + x & \text{in the mornings} \\ l & \text{during the rest of the day} \end{cases} \quad \text{height}$$

Для формул, занимающих несколько строк или для систем уравнений вместо `equation` пользуйтесь окружениями `eqnarray` и `eqnarray*`. В `eqnarray` каждая строка получает отдельный номер уравнения. В `eqnarray*` номера не ставятся.

Окружения `eqnarray` и `eqnarray*` работают наподобие таблицы из трех столбцов формата `{rc1}`, где средний столбец используется для знака равенства, или знака неравенства, или другого подходящего знака. Команда `\\` разбивает строки.

$$\begin{aligned} f(x) &= \cos x & (2.4) \\ f'(x) &= -\sin x \\ \int_0^x f(y)dy &= \sin x & (2.5) \end{aligned} \quad \text{height}$$

Заметьте, что по обеим сторонам средней колонки, знаков равенства, слишком много свободного места. Оно может быть уменьшено установкой `\setlength\arraycolsep{2pt}`, как в следующем примере. Команда `\nonumber` заставляет L<sup>A</sup>T<sub>E</sub>X не генерировать номер для соответствующей строки уравнения.

## 2.6 Полужирные символы

Команда смены шрифта `\mathbf` дает полужирные символы, но они обычные (прямые), тогда как математические символы обычно курсивные. Существует команда `\boldmath`, но *она может использоваться только вне математического режима*. То же относится и к символам.

$$\mu, M \quad \mathbf{M} \quad \boldsymbol{\mu}, \boldsymbol{M} \quad \text{height}$$

Заметьте, что запятая тоже полужирная, что может быть нежелательным.

## 2.7 Список математических символов

В следующих таблицах вы найдете все символы, известные обычно в *математическом режиме*.

Таблица 2.1: Акценты математического режима

$\hat{a}$	<code>\hat{a}</code>	$\check{a}$	<code>\check{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>	$\acute{a}$	<code>\acute{a}</code>
$\grave{a}$	<code>\grave{a}</code>	$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\widehat{A}$	<code>\widehat{A}</code>	$\widetilde{A}$	<code>\widetilde{A}</code>

Таблица 2.2: Строчные греческие буквы

$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$\omicron$	<code>o</code>	$\upsilon$	<code>\upsilon</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$\phi$	<code>\phi</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\varphi$	<code>\varphi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\chi$	<code>\chi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\psi$	<code>\psi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\omega$	<code>\omega</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>		
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>	$\tau$	<code>\tau</code>		

Таблица 2.3: Прописные греческие буквы

$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		

Таблица 2.4: Не-математические символы<sup>1</sup>

†	<code>\dag</code>	§	<code>\S</code>	©	<code>\copyright</code>
‡	<code>\ddag</code>	¶	<code>\P</code>	£	<code>\pounds</code>

<sup>1</sup>Эти символы можно использовать и в текстовом режиме.

Таблица 2.5: Бинарные отношения

Вы можете получить соответствующие отрицания добавлением перед следующими символами команды `\not`.

<	<	>	>	=	=
≤	<code>\leq</code> или <code>\le</code>	≥	<code>\geq</code> или <code>\ge</code>	≡	<code>\equiv</code>
≪	<code>\ll</code>	≫	<code>\gg</code>	≐	<code>\doteq</code>
⋖	<code>\prec</code>	⋗	<code>\succ</code>	≈	<code>\sim</code>
⋔	<code>\preceq</code>	⋕	<code>\succeq</code>	≈	<code>\simeq</code>
⊂	<code>\subset</code>	⊃	<code>\supset</code>	≈	<code>\approx</code>
⊆	<code>\subseteq</code>	⊇	<code>\supseteq</code>	≅	<code>\cong</code>
⊆	<code>\sqsubset</code> <sup>1</sup>	⊆	<code>\sqsupset</code> <sup>1</sup>	⋈	<code>\Join</code> <sup>1</sup>
⊆	<code>\sqsubseteq</code>	⊆	<code>\sqsupseteq</code>	⋈	<code>\bowtie</code>
∈	<code>\in</code>	∋	<code>\ni</code> , <code>\owns</code>	∝	<code>\propto</code>
⊥	<code>\vdash</code>	⊥	<code>\dashv</code>	⊨	<code>\models</code>
	<code>\mid</code>	∥	<code>\parallel</code>	⊥	<code>\perp</code>
☺	<code>\smile</code>	☹	<code>\frown</code>	∞	<code>\asymp</code>
:	:	∉	<code>\notin</code>	≠	<code>\neq</code> или <code>\ne</code>

<sup>1</sup>Для доступа к этому символу пользуйтесь пакетом `amssymb`.

Таблица 2.6: Бинарные операторы

+	+	-	-		
±	<code>\pm</code>	∓	<code>\mp</code>	◁	<code>\triangleleft</code>
·	<code>\cdot</code>	÷	<code>\div</code>	▷	<code>\triangleright</code>
×	<code>\times</code>	\	<code>\setminus</code>	*	<code>\star</code>
∪	<code>\cup</code>	∩	<code>\cap</code>	*	<code>\ast</code>
⊔	<code>\sqcup</code>	⊓	<code>\sqcap</code>	○	<code>\circ</code>
∨	<code>\vee</code> , <code>\lor</code>	∧	<code>\wedge</code> , <code>\land</code>	•	<code>\bullet</code>
⊕	<code>\oplus</code>	⊖	<code>\ominus</code>	◇	<code>\diamond</code>
⊙	<code>\odot</code>	⊗	<code>\otimes</code>	⊕	<code>\uplus</code>
⊗	<code>\otimes</code>	○	<code>\bigcirc</code>	∏	<code>\amalg</code>
△	<code>\bigtriangleup</code>	▽	<code>\bigtriangledown</code>	†	<code>\dagger</code>
◁	<code>\lhd</code> <sup>1</sup>	▷	<code>\rhd</code> <sup>1</sup>	‡	<code>\ddagger</code>
⊆	<code>\unlhd</code> <sup>1</sup>	⊇	<code>\unrhd</code> <sup>1</sup>	∫	<code>\wr</code>

Таблица 2.7: Большие операторы

∑	<code>\sum</code>	∪	<code>\bigcup</code>	∨	<code>\bigvee</code>	⊕	<code>\bigoplus</code>
∏	<code>\prod</code>	∩	<code>\bigcap</code>	∧	<code>\bigwedge</code>	⊗	<code>\bigotimes</code>
∏	<code>\coprod</code>	⊔	<code>\bigsqcup</code>			⊙	<code>\bigodot</code>
∫	<code>\int</code>	∫	<code>\oint</code>			⊕	<code>\biguplus</code>

Таблица 2.8: Ограничители

(	(	)	)	↑	<code>\uparrow</code>	↑	<code>\Uparrow</code>
[	[ или <code>\lbrack</code>	]	] или <code>\rbrack</code>	↓	<code>\downarrow</code>	↓	<code>\Downarrow</code>
{	<code>\{</code> или <code>\lbrace</code>	}	<code>\}</code> или <code>\rbrace</code>	↕	<code>\updownarrow</code>	↕	<code>\Updownarrow</code>
⟨	<code>\langle</code>	⟩	<code>\rangle</code>		или <code>\vert</code>		<code>\ </code> или <code>\Vert</code>
⌊	<code>\lfloor</code>	⌋	<code>\rfloor</code>	⌈	<code>\lceil</code>	⌋	<code>\rceil</code>
/	/	\	<code>\backslash</code>				



Таблица 2.9: Стрелки

$\leftarrow$	<code>\leftarrow</code> или <code>\gets</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
$\rightarrow$	<code>\rightarrow</code> или <code>\to</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\Rrightarrow$	<code>\Rrightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>	$\nearrow$	<code>\nearrow</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>	$\searrow$	<code>\searrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\swarrow$	<code>\swarrow</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\iff$ (бóльший пробел)	<code>\iff</code> (бóльший пробел)	$\leadsto$	<code>\leadsto</code> <sup>1</sup>

<sup>1</sup>Для доступа к этому символу пользуйтесь пакетом `amssymb`.

Таблица 2.10: Прочие символы

$\dots$	<code>\dots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>
$\hbar$	<code>\hbar</code>	$\imath$	<code>\imath</code>	$\jmath$	<code>\jmath</code>	$\ell$	<code>\ell</code>
$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>	$\aleph$	<code>\aleph</code>	$\wp$	<code>\wp</code>
$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\mho$	<code>\mho</code> <sup>1</sup>	$\partial$	<code>\partial</code>
$'$	<code>'</code>	$\prime$	<code>\prime</code>	$\emptyset$	<code>\emptyset</code>	$\infty$	<code>\infty</code>
$\nabla$	<code>\nabla</code>	$\triangle$	<code>\triangle</code>	$\square$	<code>\square</code> <sup>1</sup>	$\diamond$	<code>\diamond</code> <sup>1</sup>
$\perp$	<code>\perp</code>	$\top$	<code>\top</code>	$\sphericalangle$	<code>\sphericalangle</code>	$\surd$	<code>\surd</code>
$\diamondsuit$	<code>\diamondsuit</code>	$\heartsuit$	<code>\heartsuit</code>	$\clubsuit$	<code>\clubsuit</code>	$\spadesuit$	<code>\spadesuit</code>
$\neg$	<code>\neg</code> или <code>\lnot</code>	$\flat$	<code>\flat</code>	$\natural$	<code>\natural</code>	$\sharp$	<code>\sharp</code>

<sup>1</sup>Для доступа к этому символу пользуйтесь пакетом `amssymb`.

Таблица 2.11: Математические алфавиты

Пример	Команда	Требуемый пакет
$\mathrm{ABCdef}$	<code>\mathrm{ABCdef}</code>	
$\mathit{ABCdef}$	<code>\mathit{ABCdef}</code>	
$\mathnormal{ABCdef}$	<code>\mathnormal{ABCdef}</code>	
$\mathcal{ABC}$	<code>\mathcal{ABC}</code>	
$\mathcal{ABC}$	<code>\mathcal{ABC}</code>	eucal с опцией: <code>mathcal</code> или
$\mathscr{ABC}$	<code>\mathscr{ABC}</code>	eucal с опцией: <code>mathscr</code>
$\mathfrak{ABCdef}$	<code>\mathfrak{ABCdef}</code>	eufrak
$\mathbb{ABC}$	<code>\mathbb{ABC}</code>	amfonts или amssymb

## Глава 3

# Система компьютерной верстки $\LaTeX$ : Как работать над проектом?

До настоящего момента мы рассматривали как оформлять документы на  $\LaTeX$ , не касаясь конкретных реализаций  $\TeX$  и  $\LaTeX$  и приемов работы с ними. В данном разделе рассматривается конкретная реализация  $\TeX$  для Windows —  $\text{MiKTeX}$  — и основные приемы работы с ним. Описан весь цикл работы над проектом: его ввод и редактирование в программе  $\text{WinEdt}$ , трансляция и отладка, просмотр результата трансляции на экране и его вывод на принтер (утилита  $\text{Yap}$ ).

### 3.1 $\text{MiKTeX}$ и его друзья

$\text{MiKTeX}$  это пакет программ для операционных систем MS Windows включающий транслятор с  $\TeX$ ,  $\LaTeX$  и многие другие  $\TeX$ -приложения. Он доступен и распространяется бесплатно (см. ниже).

#### 3.1.1 Где получить $\text{MiKTeX}$ ?

Дистрибутив пакета  $\text{MiKTeX}$  можно найти на любом официальном сервере CTAN (см. раздел ??) в директории CTAN: `systems/win32/miktex/`. Кроме  $\text{MiKTeX}$  вам могут понадобиться другие пакеты, которые используются вместе с  $\text{MiKTeX}$ . Их список приведен ниже:

**Aladdin Ghostscript** — интерпретатор языка  $\text{PostScript}$ . Он бесплатно распространяется и доступен по адресу <http://www.cs.wisc.edu/~ghost/aladdin/index.html>.

**Yap** — утилита для просмотра и печати файлов `*.dvi`, являющихся результатом трансляции документов на  $\TeX$  и  $\LaTeX$  (см. раздел ??).  $\text{Yap}$  использует интерпретатор  $\text{Ghostscript}$  для работы с EPS-графикой. Утилита включена в стандартный дистрибутив  $\text{MiKTeX}$ .

**WinEdt** — графическая оболочка для работы с  $\TeX$  под Windows (см. раздел ??). Бесплатно распространяется и доступна по адресу <http://www.winedt.com>.

## 3.2 Последовательность работы с проектом на $\text{T}_\text{E}\text{X}$ или $\text{L}\text{A}\text{T}_\text{E}\text{X}$

Последовательность действий при работе с проектом на  $\text{T}_\text{E}\text{X}$  или  $\text{L}\text{A}\text{T}_\text{E}\text{X}$  показана на рис. ?? и описана ниже:

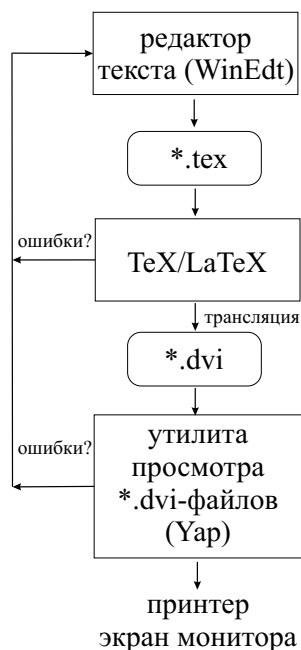


Рис. 3.1: Последовательность работы с проектом на  $\text{T}_\text{E}\text{X}$  или  $\text{L}\text{A}\text{T}_\text{E}\text{X}$ .

**Шаг 1. Ввод и реактирование входных документов.** Входной документ или документы на  $\text{T}_\text{E}\text{X}/\text{L}\text{A}\text{T}_\text{E}\text{X}$ , составляющие проект, являются обычными текстовыми файлами с расширением `tex`. Соответственно, они могут быть созданы и могут редактироваться с помощью любых редакторов текста, включая MS WinWord.

**Шаг 2. Трансляция проекта** выполняется с помощью транслятора с  $\text{T}_\text{E}\text{X}/\text{L}\text{A}\text{T}_\text{E}\text{X}$ . При трансляции образуется файл с расширением `dvi`, содержащий результат трансляции. Естественно, при трансляции возможны ошибки. Часть из них являются фатальными и приводят к прекращению дальнейшей трансляции входного проекта, другие позволяют продолжить трансляцию (команда `v` транслятора). При возникновении ошибок трансляции их необходимо исправить, вернувшись к редактору текста и отредактировав исходный файл (файлы).

**Шаг 3. Просмотр и печать результата трансляции проекта** осуществляется на заключительной стадии работы над проектом с помощью утилиты просмотра `*.dvi` файлов. На этом этапе вы также можете обнаружить ошибки набора и вернуться к шагу 1.

Работу над проектом на  $\text{T}_\text{E}\text{X}/\text{L}\text{A}\text{T}_\text{E}\text{X}$ , сводящуюся к многократному выполнению шагов 1–3, можно существенно упростить и ускорить при использовании графической оболочки WinEdt, рассмотренной в разделе ???. Вызываемая ей утилита Yap для просмотра и печати `*.dvi` файлов описана в разделе ???.

### 3.2.1 Графическая оболочка WinEdt для работы с T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X

Работа с проектом на T<sub>E</sub>X или L<sup>A</sup>T<sub>E</sub>X может быть полностью выполнена в программе WinEdt, графический интерфейс которой показан на рис. ???. Она построена по принципу Turbo-систем для работы с языками программирования C или Pascal и включает в себя встроенный редактор текста, быстрый набор команд T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X, панель инструментов для быстрого выполнения шагов 1–3 процедуры работы над проектом и многое другое.

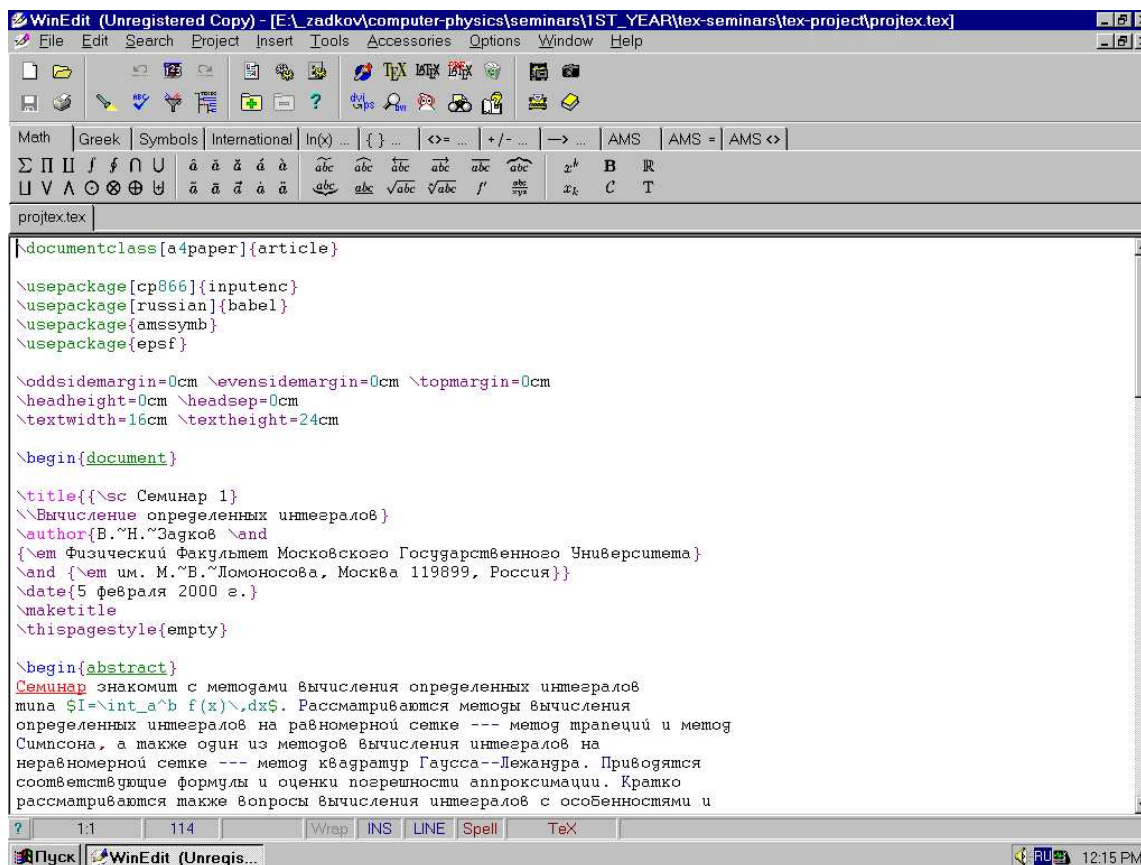


Рис. 3.2: Графический интерфейс WinEdt.

Ввод и редактирование текста проекта на T<sub>E</sub>X или L<sup>A</sup>T<sub>E</sub>X осуществляется в окне редактора. Редактор имеет интуитивный интерфейс, схожий со встроенными в Turbo-системы редакторами. Работа с текстом проекта облегчается также тем, что команды T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X выделяются цветом на экране. Одновременно можно работать с несколькими файлами проекта, при этом каждый из них будет открыт в своем окне и обозначен ярлыком с именем файла в верхней части окна редактора. Переключение между различными окнами осуществляется при помощи щелчка мышью на соответствующий ярлык с именем файла.

В верхней части окна графического интерфейса WinEdt расположено меню, под ним — панель инструментов и панели символов (рис. ???).

Панели символов в нижней части рис. ?? содержат все основные символы T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X, используемые для формульного набора, и набор специальных символов, используемых при наборе текста. Кроме того, три дополнительных панели дают возможность доступа к большинству ма-



Рис. 3.3: Панель инструментов и таблицы символов WinEdt.

тематических символов  $\text{AMSTeX}$ <sup>1</sup>. Ввод необходимого символа в текущую позицию в тексте в открытом окне редактора осуществляется щелчком мыши по иконке символа.

Панель инструментов в верхней части рис. ?? содержит множество иконок для выполнения различных функций по работе или управлению проектом на  $\TeX$ / $\LaTeX$ . Не рассматривая стандартные для  $\text{Windows}$ -систем функции типа создания нового файла, записи файла, открытия файла или его печати, мы остановимся только на основных функциях работы с проектом.

$\TeX$  выполняет трансляцию проекта на  $\TeX$ . Если проект состоит из нескольких файлов, необходимо указать головной файл с помощью иконки со знаком “+”. При этом независимо от того, с каким из открытых файлов в редакторе вы в данное время работаете, будет первым транслироваться указанный вами головной файл. При трансляции на экране появляется окно, в котором отображаются сообщения транслятора, включая ошибки трансляции.

$\LaTeX$  выполняет трансляцию проекта на  $\LaTeX$ . Если проект состоит из нескольких файлов, необходимо указать головной файл с помощью иконки со знаком “+”. При этом независимо от того, с каким из открытых файлов в редакторе вы в данное время работаете, будет первым транслироваться указанный вами головной файл. При трансляции на экране появляется окно, в котором отображаются сообщения транслятора, включая ошибки трансляции.

$\text{DVI Preview}$  вызывает утилиту  $\text{Yap}$  для просмотра и печати результата трансляции (см. раздел ??).

$\text{DVIPS}$  производит перевод  $\text{dvi}$ -файла результата трансляции в формат  $\text{PostScript}$  (файл с расширением  $\text{ps}$ ).

$\text{GSView}$  вызывает утилиту  $\text{GSView}$  для просмотра и печати  $\text{PostScript}$  файлов. Используется в комбинации с предыдущей функцией.

$\text{Erase Working Files}$  уничтожает все рабочие файлы, полученные в результате работы над проектом.

Описание множества других функций программы WinEdt вы можете найти в разделе  $\text{Help}$  меню.

### 3.2.2 Утилита $\text{Yap}$ для просмотра и печати результата трансляции

Просмотр и печать результата трансляции проекта может быть выполнена с помощью утилиты  $\text{Yap}$ , графический интерфейс которой показан на рис. ?. Она позволяет просматривать на экране и печатать файлы, полученные в результате трансляции проектов на  $\TeX$  и  $\LaTeX$  (файлы с расширением  $\text{dvi}$ ). При этом она поддерживает следующие графические форматы рисунков, включенных в ваши проекты:  $\text{EMF}$  (Enhanced Metafiles, \*.emf),  $\text{WMF}$  (Windows Metafiles, \*.wmf),  $\text{BMP}$  (Device-Independent Bitmap, \*.bmp),  $\text{EPS}$  (Encapsulated Postscript, \*.eps),  $\text{GIF}$  (Graphics Interchange Format, \*.gif) и др.

<sup>1</sup>Для пользования данными наборами символов необходимо использовать соответствующие пакеты.

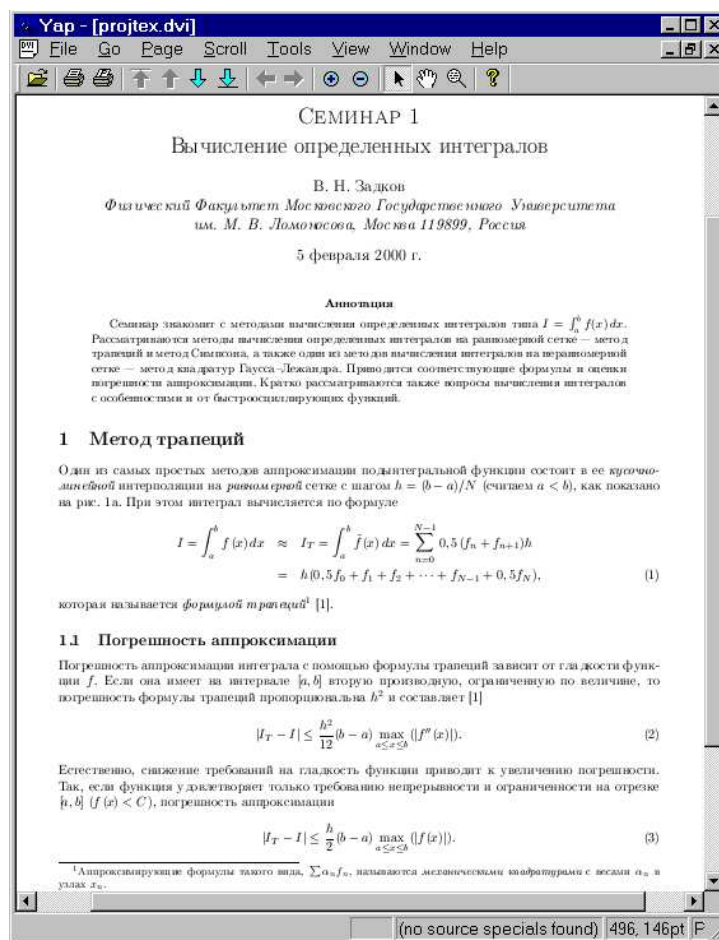


Рис. 3.4: Графический интерфейс утилиты Yap.

Большинство функций утилиты Yap, необходимые для просмотра и печати \*.dvi файлов доступны через панель инструментов, расположенную в верхней части интерфейса под меню. Слева направо расположены следующие иконки, кликая мышкой на которые вы можете выполнить необходимые действия по просмотру и печати документа:

**Open** позволяет открыть \*.dvi файл, являющийся результатом трансляции проекта на T<sub>E</sub>X или L<sup>A</sup>T<sub>E</sub>X.

**Print** позволяет напечатать результат трансляции целиком или заданный диапазон страниц на принтер. При этом рисунки, включенные в тело документа, будут печататься за исключением рисунков в формате PostScript (вместо них будет оставлено пустое место). Для печати документов, включающих PostScript-графику, используется следующая команда.

**Print PostScript** позволяет напечатать результат трансляции целиком или заданный диапазон страниц на PostScript принтер или на любой принтер при помощи программы GhostView, которая должна быть установлена в системе.

**First Page** позволяет перейти на первую страницу документа.

**Previous Page** позволяет перейти на предыдущую страницу документа.

**Next Page** позволяет перейти на следующую страницу документа.

**Last Page** позволяет перейти на последнюю страницу документа.

**Back** и **Forward** при просмотре документов, как правило, не используются.

**Zoom In** позволяет увеличить отображаемую страницу на экране (без изменения ее реальных размеров).

**Zoom Out** позволяет уменьшить отображаемую страницу на экране (без изменения ее реальных размеров).

**Standard Pointer** используется в качестве обычного указателя или курсора.

**Hand Tool** используется для перемещения отображаемой страницы документа в пределах области **Yap** в случае, если страница не помещается целиком на экране.

**Magnifying Glass** используется для увеличения небольшого фрагмента текста вокруг указателя. Полезно применять для просмотра деталей формул без увеличения страницы целиком. Заметим, что данная функция не распространяется на картинки в формате **PostScript**.

**About** выдает информацию о программе и ее версии.

## Глава 4

# Система компьютерной верстки $\text{\LaTeX}$ : Пример проекта

В качестве примера проекта по  $\text{\LaTeX}$  рассмотрена верста текста одного из семинаров с использованием класса `article`. Приведен исходный текст проекта на  $\text{\LaTeX}$  и результат его трансляции, выведенный на печать.



## 4.1 Исходный текст на L<sup>A</sup>T<sub>E</sub>X

---

```
\documentclass[a4paper]{article}
\usepackage[cp866]{inputenc}
\usepackage[russian]{babel}
\usepackage{amssymb}
\usepackage{graphicx}
\oddsidemargin=0cm \evensidemargin=0cm \topmargin=0cm
\headheight=0cm \headsep=0cm
\textwidth=16cm \textheight=24cm

\begin{document}

\title{{\sc Семинар 1}
\\Вычисление определенных интегралов}
\author{В.~Н.~Задков \and
{\em Физический Факультет Московского Государственного Университета}
\and {\em им. М.~В.~Ломоносова, Москва 119899, Россия}}
\maketitle
\pagestyle{empty}\thispagestyle{empty}

\begin{abstract}
Семинар знакомит с методами вычисления определенных интегралов
типа  $I = \int_a^b f(x) dx$ . Рассматриваются методы вычисления
определенных интегралов на равномерной сетке --- метод трапеций и метод
Симпсона, а также один из методов вычисления интегралов на
неравномерной сетке --- метод квадратур Гаусса--Лежандра. Приводятся
соответствующие формулы и оценки погрешности аппроксимации. Кратко
рассматриваются также вопросы вычисления интегралов с особенностями и
от быстроосциллирующих функций.
\end{abstract}
```

---

---

```
\section{Метод трапеций}
```

Один из самых простых методов аппроксимации подынтегральной функции состоит в ее кусочно-линейной интерполяции на равномерной сетке с шагом  $h=(b-a)/N$  (считаем  $a < b$ ), как показано на рис. [\ref{fig-int-ts}](#)a. При этом интеграл вычисляется по формуле

```
\begin{equationarray}
I=\int_a^b f(x)\,dx\approx I_T=\int_a^b \tilde{f}(x)\,dx=\sum_{n=0}^{N-1}
0,5(f_n+f_{n+1})h\nonumber\
&= h(0,5f_0+f_1+f_2+\cdots+f_{N-1}+0,5f_N),
\end{equationarray}
```

которая называется формулой трапеций [\footnote{Аппроксимирующие формулы такого вида,  \$\sum \alpha\_n f\_n\$ , называются механическими квадратурами с весами  \$\alpha\_n\$  в узлах  \$x\_n\$ .} \cite{fedorenko}](#).

```
\begin{figure}
\begin{center}
\includegraphics[width=12.cm]{figure.eps}
\end{center}
\caption{Аппроксимация подынтегральной функции  $f(x)$  с помощью (а)
кусочно-линейной (метод трапеций) и (б) кусочно-квадратичной (метод
Симпсона) интерполирующих функций на равномерной сетке. Р2 ---
интерполяционный полином Лагранжа второй степени.}
\label{fig-int-ts}
\end{figure}
```

```
\subsection{Погрешность аппроксимации}
```

Погрешность аппроксимации интеграла с помощью формулы трапеций зависит от гладкости функции  $f$ . Если она имеет на интервале  $[a,b]$  вторую производную, ограниченную по величине, то погрешность формулы трапеций пропорциональна  $h^2$  и составляет [\cite{fedorenko}](#)

```
\begin{equation}
|I_T-I|\leq \frac{h^2}{12}(b-a)\max_{a\leq x\leq b}(|f''(x)|).
\end{equation}
```

Естественно, снижение требований на гладкость функции приводит к увеличению погрешности. Так, если функция удовлетворяет только требованию непрерывности и ограниченности на отрезке  $[a,b]$  ( $f(x) < C$ ), погрешность аппроксимации

```
\begin{equation}
|I_T-I|\leq \frac{h}{2}(b-a)\max_{a\leq x\leq b}(|f(x)|).
\end{equation}
```

---

---

```
\section{Формулы Ньютона-Котеса}
```

Продолжая улучшать формулы трапеций и Симпсона на равномерной сетке, можно, идя по тому же пути, получить аппроксимирующие формулы более высоких порядков. Семейство этих формул называется формулами Ньютона-Котеса~\cite{korn}. Не обсуждая их здесь детально, все же приведем таблицу соответствующих весов функций, используемых в аппроксимирующих формулах (см. табл.~\ref{weights}). На практике, однако, аппроксимирующие формулы высоких порядков (более 4--5) обычно не используются из-за существенных ошибок округления.

```
\begin{table}[h]
\caption{Веса в формулах интегрирования Ньютона-Котеса.}
\smallskip
\begin{center}
\begin{tabular}{ccllllll}
\hline
Степень & Название & Веса & & & & \\
полинома & формулы & & & & & \\
1 & Формула трапеций & h/2 & h/2 & & & \\
2 & Формула Симпсона & h/3 & 4h/3 & h/3 & & \\
3 & Формула 3/8 Симпсона & 3h/8 & 9h/8 & 9h/8 & 3h/8 & \\
4 & Формула Милна & 14h/45 & 64h/45 & 24h/45 & 64h/45 & 14h/45 \\
\hline
\end{tabular}
\end{center}
\label{weights}
\end{table}
```

```
\begin{thebibliography}{9}
\bibitem{fedorenko}
Р. П. Федоренко, {\em Введение в вычислительную физику}. --- М.: Изд-во Моск. физ.-техн. ин-та, 1994.
\bibitem{korn}
Г. Корн, Т. Корн, {\em Справочник по математике для научных работников и инженеров}. --- М.: Физматгиз, 1968.
\end{thebibliography}
```

```
\end{document}
```

---

## 4.2 Результат трансляции

# СЕМИНАР 1

## Вычисление определенных интегралов

В. Н. Задков

*Физический Факультет Московского Государственного Университета  
им. М. В. Ломоносова, Москва 119899, Россия*

4 марта 1999 г.

### Аннотация

Семинар знакомит с методами вычисления определенных интегралов типа  $I = \int_a^b f(x) dx$ . Рассматриваются методы вычисления определенных интегралов на равномерной сетке — метод трапеций и метод Симпсона, а также один из методов вычисления интегралов на неравномерной сетке — метод квадратур Гаусса–Лежандра. Приводятся соответствующие формулы и оценки погрешности аппроксимации. Кратко рассматриваются также вопросы вычисления интегралов с особенностями и от быстроосциллирующих функций.

## 1 Метод трапеций

Один из самых простых методов аппроксимации подынтегральной функции состоит в ее *кусочно-линейной* интерполяции на *равномерной* сетке с шагом  $h = (b - a)/N$  (считаем  $a < b$ ), как показано на рис. 1а. При этом интеграл вычисляется по формуле

$$\begin{aligned} I = \int_a^b f(x) dx &\approx I_T = \int_a^b \tilde{f}(x) dx = \sum_{n=0}^{N-1} 0,5(f_n + f_{n+1})h \\ &= h(0,5f_0 + f_1 + f_2 + \dots + f_{N-1} + 0,5f_N), \end{aligned} \quad (1)$$

которая называется *формулой трапеций*<sup>1</sup> [1].

### 1.1 Погрешность аппроксимации

Погрешность аппроксимации интеграла с помощью формулы трапеций зависит от гладкости функции  $f$ . Если она имеет на интервале  $[a, b]$  вторую производную, ограниченную по величине, то погрешность формулы трапеций пропорциональна  $h^2$  и составляет [1]

$$|I_T - I| \leq \frac{h^2}{12} (b - a) \max_{a \leq x \leq b} (|f''(x)|). \quad (2)$$

Естественно, снижение требований на гладкость функции приводит к увеличению погрешности. Так, если функция удовлетворяет только требованию непрерывности и ограниченности на отрезке  $[a, b]$  ( $f(x) < C$ ), погрешность аппроксимации

$$|I_T - I| \leq \frac{h}{2} (b - a) \max_{a \leq x \leq b} (|f(x)|). \quad (3)$$

<sup>1</sup>Аппроксимирующие формулы такого вида,  $\sum \alpha_n f_n$ , называются *механическими квадратурами* с весами  $\alpha_n$  в узлах  $x_n$ .

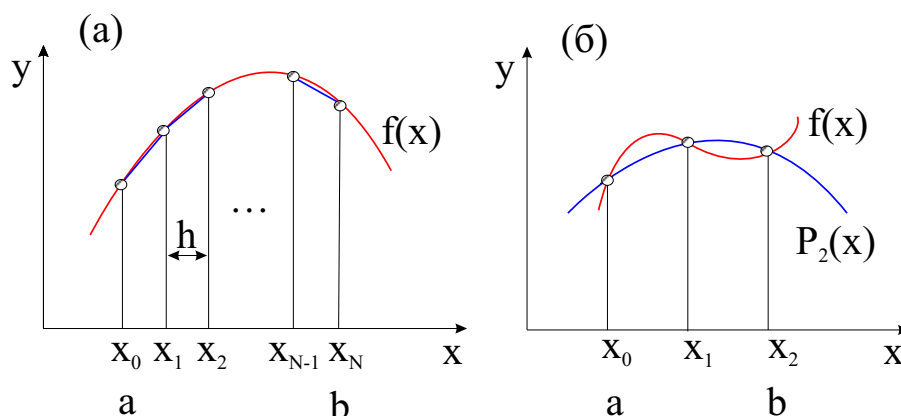


Рис. 1: Аппроксимация подынтегральной функции  $f(x)$  с помощью (а) кусочно-линейной (метод трапеций) и (б) кусочно-квадратичной (метод Симпсона) интерполирующих функций на равномерной сетке.  $P_2$  — интерполяционный полином Лагранжа второй степени.

## 2 Формулы Ньютона-Котеса

Продолжая улучшать формулы трапеций и Симпсона на равномерной сетке, можно, идя по тому же пути, получить аппроксимирующие формулы более высоких порядков. Семейство этих формул называется формулами Ньютона-Котеса [2]. Не обсуждая их здесь детально, все же приведем таблицу соответствующих весов функций, используемых в аппроксимирующих формулах (см. табл. 1). На практике, однако, аппроксимирующие формулы высоких порядков (более 4–5) обычно не используются из-за существенных ошибок округления.

Таблица 1: Веса в формулах интегрирования Ньютона-Котеса.

Степень полинома	Название формулы	Веса				
1	Формула трапеций	$h/2$	$h/2$			
2	Формула Симпсона	$h/3$	$4h/3$	$h/3$		
3	Формула 3/8 Симпсона	$3h/8$	$9h/8$	$9h/8$	$3h/8$	
4	Формула Милна	$14h/45$	$64h/45$	$24h/45$	$64h/45$	$14h/45$

## Список литературы

- [1] Р. П. Федоренко, *Введение в вычислительную физику*. — М.: Изд-во Моск. физ.-техн. ин-та, 1994.
- [2] Г. Корн, Т. Корн, *Справочник по математике для научных работников и инженеров*. — М.: Физматгиз, 1968.

# Литература

- [1] Donald E. Knuth. *The TeXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9; Русский перевод — Дональд Е. Кнут. *Все про TeX*. Протвино, РДTeX, 1993.
- [2] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- [3] С. М. Львовский. *Набор и верстка в пакете L<sup>A</sup>T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*. М.: Космоинформ, 199, 374 с.
- [4] Graham Williams. *The TeX Catalogue* полный список множества пакетов, имеющих отношение к TeX и L<sup>A</sup>T<sub>E</sub>X. Доступен в Интернет по адресу `CTAN:help/Catalogue/catalogue.html`