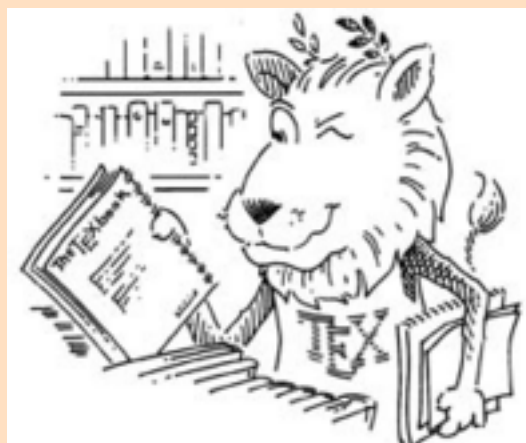


Стефка Караколева

ВЪВЕДЕНИЕ
В ИЗДАТЕЛСКАТА СИСТЕМА
L^AT_EX 2_ε



Русенски Университет «Ангел Кънчев»

Русе, 2005

УДК 004.4+004.9 (075.8)

Ст. Караколева

ВЪВЕДЕНИЕ в издателската система **L^AT_EX 2_ε**

Русенски Университет „Ангел Кънчев“ - Русе, 2005.

Учебното пособие запознава с последната, съвременна версия на системата за текстообработка L^AT_EX. Подробно се разглежда изтегляне и инсталиране на системата от Internet. Включени са всички основни структури за писане на научни документи, както и допълнителни пакети за оформяне на презентации, слайдове, включване на графики, създаване на документи за световната Интернет мрежа във формат pdf. Учебното пособие е първото от този род на български език и може успешно да бъде използвано не само от студенти.

Съдържанието на книгата е съобразено с материала, преподаван в РУ «Ангел Кънчев» и сходни български университети. Включеният материал се изучава по дисциплината «Математически софтуер» за специалност «Информатика» на Русенски Университет «Ангел Кънчев».

Книгата може да бъде използвана от студенти - степен «бакалавър» и «магистър», дипломанти, докторанти, инженери, научни работници.

©Стефка Караколева - автор, 2005.

Рецензент: доц. д-р инж. Ангел Смрикаров

Дизайн и предпечатна подготовка с MiK_TE_X

Увод

L^AT_EX [1] – е система за текстообработка, предназначена за изработване на научни документи с високо типографско качество. L^AT_EX използва T_EX [2] като механизъм за текстообработка.

Съвременната система е напълно подходяща за писане на писма, книги, статии, дисертации, цветни презентации в PDF формат, електронни учебници, хипер-текстови документи за www. Богатите графични възможности на съвременните системи, основани на T_EX, позволяват лесно да се изработват книги от области като математика, физика, химия, техника, музика, шахмат и други.

Това кратко въведение описва L^AT_EX 2_ε и е достатъчно за повечето приложения на L^AT_EX. За пълно описание на системата L^AT_EX вижте [1, 3].

Материалът в книгата е структуриран в шест глави:

Глава 1 разказва за основната структура на L^AT_EX-документите. От нея ще научите малко и за историята на L^AT_EX. След четене на тази глава, трябва да имате груба представа за това, как работи L^AT_EX.

Глава 2 задълбочава в детайли тексто-обработката на вашия документ. Тя обяснява повечето съществени команди и среди в L^AT_EX. След прочитане на тази глава, ще можете да пишете вашите първи документи.

Глава 3 обяснява как да пишете формули в L^AT_EX. Многото примери в тази глава ще ви покажат как да използвате една от най-силните страни на L^AT_EX. В края на тази глава има таблици с повечето достъпни в L^AT_EX математически символи.

Глава 4 разказва за това, как да се генерира азбучен указател и библиография и как да се включват EPS графики. Тя обяснява и как да се създават PDF документи с pdfL^AT_EX и представя няколко удобни разширения, като пакета `hyperref`.

Глава 5 показва как да се използва L^AT_EX за създаване на графики. Вместо да рисувате картинката с някаква графична програма, да я записвате във файл и след това да го включвате в L^AT_EX-файла,

вие описвате картинката, а \LaTeX я рисува вместо вас. Описани са накратко и два елегантни пакета за рисуване.

Глава 6 съдържа опасна информация за това, как може да се изменя стандартния макет на документа, произвеждан от \LaTeX . От нея ще научите как да правите допълнителни настройки според вашите нужди.

Важно е да се четат главите последователно. Все пак, книгата не е толкова голяма. Важно е също да четете и примерите внимателно, защото голямо количество от информацията е дадена в тях.

\LaTeX е достъпен за повечето компютри, от IBM PC или Mac, до големите UNIX или VMS системи. На много университетски компютърни мрежи, \LaTeX вече е установен и готов за работа. Това е така и в компютърната зала на Университетска Библиотека на Русенски Университет «Ангел Кънчев», както и в компютърните зали на катедра «Информатика». Целта на тази книга е *не* да ви обясни как да инсталирате и настройвате системата \LaTeX , а да ви научи как да пишете вашите документи така, че те да могат да бъдат обработени с \LaTeX .

Ако искате да намерите някакъв, отнасящ се до \LaTeX материал, потърсете го на един от ftp архивите Comprehensive \TeX Archive Network (СТАН). Неговият основен адрес е <http://www.ctan.org>. Всички пакети могат също да се получат от ftp-архива <ftp://www.ctan.org> и неговите огледални сайтове по целия свят. Те могат да бъдат намерени например на <ftp://ctan.tug.org> (САЩ), <ftp://ftp.dante.de> (Германия), <ftp://ftp.tex.ac.uk> (Великобритания).

По-нататък в книгата постоянно ще намирате препратки на СТАН, особено насочващи към софтуер и документи, които може да изтеглите. Вместо изписване на пълния запис на конкретен адрес, пише просто СТАН: и след това адреса в дървото на СТАН, където трябва да отидете. Ако искате да имате \LaTeX на вашия собствен компютър, вижте това, което е достъпно на адрес <CTAN:/tex-archive/systems>.

Стефка Караколева <skarakoleva@ru.acad.bg>

Катедра «Числени методи и статистика»
Русенски Университет «А.Кънчев»

Съдържание

Увод	iii
1 Необходима информация	1
1.1 \TeX и \LaTeX	1
1.1.1 \TeX	1
1.1.2 \LaTeX	1
1.2 Основни типографски принципи на \LaTeX	2
1.2.1 Автор, дизайнер и технически изпълнител	2
1.2.2 Дизайн на макета	3
1.2.3 Предимства и недостатъци на \LaTeX	3
1.3 Изходни файлове на \LaTeX	4
1.3.1 Интервали	5
1.3.2 Специални символи	5
1.3.3 Команди на \LaTeX	5
1.3.4 Коментарни редове	6
1.4 Структура на входния файл	7
1.5 Типична сесия при работа с \LaTeX	9
1.6 Макет на документа	10
1.6.1 Класове документи	10
1.6.2 Пакети	10
1.6.3 Стиллове на страницата	12
1.7 Разширения на файлове, които се срещат при работа с \LaTeX	12
1.8 Големи проекти	15
1.9 Инсталиране на MiKTeX	16
1.10 Инсталиране на Shell (обвивка, текстов редактор)	17
1.11 Езикова настройка в MiKTeX	18
2 Обработка на текста	21
2.1 Структура на текста и езика	21
2.2 Разбиване на редове и страници	23
2.2.1 Подравнени параграфи	23
2.2.2 Пренасяне	24
2.3 Специални стрингове	25

2.4	Специални букви и символи	26
2.4.1	Знаци кавички	26
2.4.2	Тире и дефис	26
2.4.3	Тилда (\sim)	26
2.4.4	Знак за градус (\circ)	27
2.4.5	Символ Евро (€)	27
2.4.6	Многоточие (\dots)	27
2.4.7	Акценти и специални символи	28
2.5	Поддръжка на чужди езици	28
2.5.1	Поддръжка на френски език	31
2.5.2	Поддръжка на немски език	32
2.5.3	Поддръжка на Кирилица	33
2.6	Интервали между думите	34
2.7	Заглавия, глави и раздели	35
2.8	Препратки	38
2.9	Забележки под печатното поле	38
2.10	Подчертани думи	39
2.11	Среди	40
2.11.1	Списък, изброяване и описание	40
2.11.2	Подравняване вляво, вдясно и центриране	40
2.11.3	Цитати и стихове	41
2.11.4	Резюме	42
2.11.5	Буквално възпроизвеждане	42
2.11.6	Таблицы	43
2.12	Плаващи обекти	45
3	Набор на математически формули	49
3.1	Общи сведения	49
3.2	Групиране в математически режим	51
3.3	Елементи на математическите формули	52
3.4	Математически интервали	56
3.5	Вертикално разположен материал	56
3.6	Фантоми	58
3.7	Размер на математическия шрифт	59
3.8	Теорема, леми, закони	60
3.9	Дебели (bold) символи	61
3.10	\LaTeX	62
3.10.1	Сравнение между <code>eqnarray</code> и <code>align</code>	62
3.10.2	Команда <code>nonumber</code> и цитиране	63
3.10.3	Команда <code>subequations</code>	63
3.10.4	Дълги формули на няколко реда	63
3.10.5	Случаи (cases) и текст във формули	64
3.10.6	Матрици в \LaTeX	64
3.11	Списък на математическите символи	66

4	Специални възможности	73
4.1	Включване на Encapsulated POSTSCRIPT графики	73
4.2	Библиография	75
4.3	Азбучен указател	76
4.4	Включване на речник	78
4.5	Настройка на колонтитули	79
4.6	Пакет verbatim	80
4.7	Изтегляне и инсталиране на L ^A T _E X-пакети	80
4.8	Работа с pdfL ^A T _E X	81
4.8.1	PDF-документи за WWW	83
4.8.2	Шрифтове	83
4.8.3	Използване на графики	84
4.8.4	Пакет hyperref	85
4.8.5	Проблеми с хипер-заглавията	89
4.9	Съвместимост на изходните текстове в L ^A T _E X и pdfL ^A T _E X	90
4.10	Създаване на презентации с помощта на pdfscreen	91
5	Генериране на математически графики	95
5.1	Обзор	95
5.2	Среда picture	96
5.2.1	Основни команди	96
5.2.2	Отсечки	97
5.2.3	Вектори	98
5.2.4	Окръжности	99
5.2.5	Текст и формули	100
5.2.6	Команди \multiput и \linethickness	100
5.2.7	Овали. Команди \thinlines и \thicklines	101
5.2.8	Създаване и многократно използване на блокове с картинки	102
5.2.9	Квадратични криви на Безие	103
5.3	X _y -pic	104
5.4	Създаване на POSTSCRIPT графики с пакета PSTricks	107
5.4.1	Основни понятия в пакета PSTricks	107
5.4.2	Примери на използване на PSTricks	109
5.4.3	Свързване на текст и графика	112
6	Настройка на L^AT_EX	115
6.1	Нови команди, среди и пакети	115
6.1.1	Нови команди	115
6.1.2	Нови среди	117
6.1.3	Изработване на ваш собствен пакет	118
6.2	Шрифтове и техните размери	118
6.2.1	Команди за смяна на шрифта	118
6.2.2	Логическо маркиране	122

6.3	Интервали	123
6.3.1	Интервали между редовете	123
6.3.2	Форматиране на параграфи	123
6.3.3	Хоризонтални интервали	124
6.3.4	Вертикални интервали	125
6.4	Макет на страницата	126
6.5	Блокове	128
6.6	Линийки и подпори	130
	Библиография	133
	Азбучен указател	135

Списък на фигурите

1.1	Минимален L ^A T _E X-файл	8
1.2	Макет на документ-статия	8
4.1	Пример за настройка на fancyhdr	79
4.2	Пример за входен файл на pdfscreen	92
4.3	Примерен файл за настройка на пакета pdfscreen за български език	93
6.1	Примерен пакет	118
6.2	Параметри на макета на страницата.	127

Списък на таблиците

1.1	Класове документи	10
1.2	Опции на класовете документи	11
1.3	Някои от разпространяваните с \LaTeX пакети	13
1.4	Предопределени стилове на \LaTeX -страницата	14
2.1	Пълен набор на Евро символи	28
2.2	Акценти и специални символи	29
2.3	Специални команди за френски език	32
2.4	Специални символи за немски език	33
2.5	Български, руски и украински езици	34
2.6	Ключове за разрешени позиции на плаващ обект	46
3.1	Акценти в математически режим	66
3.2	Малки гръцки букви	66
3.3	Главни гръцки букви	66
3.4	Бинарни отношения	67
3.5	Бинарни оператори	67
3.6	Големи оператори	68
3.7	Стрелки	68
3.8	Ограничители	68
3.9	Големи ограничители	68
3.10	Други символи	69
3.11	Не-математически символи	69
3.12	Ограничители AMS	69
3.13	Букви от гръцки и иврит AMS	69
3.14	Бинарни отношения AMS	70
3.15	Стрелки AMS	70
3.16	Отрицателни бинарни отношения и стрелки AMS	71
3.17	Бинарни оператори AMS	71
3.18	Други символи AMS	72
3.19	Математически азбуки	72
4.1	Имена на опциите на пакета <code>graphicx</code>	74
4.2	Примерен синтаксис на ключове за указателя	77

6.1	Шрифтове	120
6.2	Размер на шрифта	120
6.3	Абсолютни размери на шрифтовете в стандартните класове	121
6.4	Математически шрифтове	121
6.5	Единици за размерност в \TeX	125

Глава 1

Необходима информация

Началото на тази глава съдържа кратък обзор на историята и философията на \LaTeX . Втората част на главата изяснява основната структура на \LaTeX -документите. След като се запознаете с тази глава, трябва да разберете как работи \LaTeX , което е необходимо, за да четете книгата нататък.

1.1 \TeX и \LaTeX

1.1.1 \TeX

\TeX е компютърна програма, създадена от американския математик и програмист Доналд Кнут (Donald E. Knuth) [2], предназначена за обработка на документи, съдържащи голям брой формули. Кнут започва да пише \TeX през 1977 година с цел, да подобри типографското качество на своите собствени книги и статии. Програмата \TeX , във вида, в който днес се използва, е разработена през 1982 година с малки добавки през 1989. \TeX е известен със своята изключителна стабилност, икономичност, преносимост на различни типове компютри, многоезикова поддръжка и практически пълна липса на грешки. Номерът на версията на \TeX клони към числото π и в момента е равен на 3.141592.

\TeX се произнася «тех», с “ch” като в немската дума “Ach”. В ASCII среда \TeX трябва да се пише `TeX`.

1.1.2 \LaTeX

\LaTeX е макропакет, позволяващ на авторите да обработват и печатат документи с високо типографско качество, с помощта на предварително определени, професионални макети. \LaTeX е създаден от Leslie Lamport [1]. Като механизъм за обработка той използва \TeX . В наши дни \LaTeX се поддържа от Frank Mittelbach.

През 1994 година пакетът \LaTeX е обновен от екипа на международния проект $\LaTeX 3$ под ръководството на Frank Mittelbach. В него са добавени някои подобрения и са обединени всички варианти на \LaTeX , разединени с пускането преди години на версията $\LaTeX 2.09$. За да не се бърка тази нова версия със старата, тя е наречена $\LaTeX 2_{\epsilon}$. Тази книга обхваща именно $\LaTeX 2_{\epsilon}$.

\LaTeX се произнася «лейтех» или «латех». Ако се цитира \LaTeX в ASCII среда, се пише `LaTeX`. $\LaTeX 2_{\epsilon}$ се произнася «лейтех-ту-епс» и се пише `LaTeX2e`.

1.2 Основни типографски принципи на \LaTeX

1.2.1 Автор, дизайнер и технически изпълнител

За да публикуват, авторите дават своите ръкописи в издателството. След това един от дизайнерите на издателството определя макета на документа (ширина на текста, шрифтове, интервали под и над заглавие и т.н.). Дизайнерът записва своите инструкции в ръкописа и го дава на техническия изпълнител, който обработва текста в съответствие с тези инструкции.

В много случаи, човекът-дизайнер се опитва да разбере какво е имал предвид авторът, когато е писал своя ръкопис. Той определя заглавията на главите, цитати, примери, формули и други, използвайки своя професионален опит и съдържанието на ръкописа.

В \LaTeX средата, \LaTeX играе ролята на дизайнер на книгата, използвайки `TeX` като технически изпълнител. Но \LaTeX е “само” програма, и следователно се нуждае от точни инструкции. Авторът е длъжен да предостави допълнителна информация, описваща логическата структура на неговия документ. Тази информация се записва в текста във вид на \LaTeX «команди».

Този принцип на \LaTeX , коренно се различава от WYSIWYG¹ подхода, приет в голяма част от съвременните текстови процесори, като *MS Word* или *Corel WordPerfect*. В тези приложения авторите определят макета на документа интерактивно, докато въвеждат текста в компютъра. В процеса на работа те могат да видят на екрана как ще изглежда документа след отпечатване.

При използване на \LaTeX обикновено не може да се види изходната картина по време на печатането на текста. Тя обикновено се вижда на екрана след обработка на файла с \LaTeX . След това могат да се направят корекции преди документът да се изпрати към принтера.

¹What you see is what you get.

1.2.2 Дизайн на макета

Типографският дизайн е професия. Неопитните автори често допускат сериозни грешки при форматирането, предполагайки, че дизайнът на книгата е въпрос на естетика: „Ако документът изглежда красиво, следователно е с добър дизайн“. Но тъй като документът е предназначен за четене, а не за излагане в художествена галерия, удобството при четене и яснотата са много по-важни, отколкото красивият му вид. Например:

- Размерът на шрифта и номерацията на разделите трябва да се избират така, че да направят структурата на главите и разделите ясна за читателя.
- Редът трябва да бъде достатъчно къс, за да не се напрягат очите на читателя и достатъчно дълъг за красиво запълване на страницата.

С WYSIWYG системите, авторите често създават естетически приятни документи, със слабо изразена или непоследователна структура. \LaTeX предотвратява такива грешки при форматирането, заставляйки автора да обявява *логическата* структура на документа. След това вече \LaTeX избира най-подходящия макет (стил) на документа.

1.2.3 Предимства и недостатъци на \LaTeX

Темата, която често се обсъжда, когато хора от света на WYSIWYG се срещнат с потребителите на \LaTeX е «предимствата на \LaTeX пред стандартните текстови процесори» или обратното. Най-доброто, което може да се направи, ако започне такава дискусия, е тя да се избегне, защото често е безмислена. Но не винаги може да се отклони спора. . .

Ето ви няколко аргумента. Основните преимущества на \LaTeX пред обикновените текстови процесори:

- Достъпни са професионално изработени макети, които могат да направят вашия документ да изглежда наистина «като издание».
- Сложни формули се записват лесно и удобно в линеен вид.
- Потребителят трябва да научи само няколко лесно разбираеми команди, които определят логическата структура на документа. Почти никога не се налага той да се грижи за макета на документа.
- Лесно се изработват дори сложни структури, като съдържание, забележки, литература, индекс, списъци на фигури и таблици и други.
- За решение на много типографски задачи, които не се поддържат директно от основния \LaTeX , има свободно разпространявани допълнителни пакети. Например, съществуват пакети за включване

на POSTSCRIPT-графики или за оформяне на библиография в точно съответствие с конкретни стандарти. Много от тези допълнителни пакети са описани в *The L^AT_EX Companion* [3].

- L^AT_EX поощрява авторите да пишат добре структурирани документи, тъй като именно така работи L^AT_EX — чрез дефиниране на структури.
- T_EX, форматиращото сърце на L^AT_EX 2_ε, е изключително мобилен и свободно достъпен. Затова системата работи фактически на всички съществуващи платформи.
- Сорс-файловете, писани с L^AT_EX, са изключително малки по обем и могат да се пишат с какъвто и да е текстов редактор.

L^AT_EX има също и някои недостатъци, но е трудно да се намерят сред тях съществени. ; -)

- Много хора казват, че L^AT_EX е труден за усвояване.

Забележително изказване по този въпрос е направено от Лесли Лампорт, авторът на най-разпространения макропакет L^AT_EX :

«Да го използвате е много лесно, ако принадлежите към тези два процента от населението, които са способни да мислят логически и могат да прочетат ръководството на потребителя. Останалите деветдесет и осем процента го определят като изключително труден и невъзможен за използване.»

- Поради това, че вградените макети имат множество параметри за настройване, изработването на напълно нов макет на документа не е много просто и отнема много време, затова оставете това на специалистите.
- Изключително трудно се пишат неструктурирани и неорганизиранни документи.

1.3 Изходни файлове на L^AT_EX

Входен файл за L^AT_EX е обикновен текстов ASCII файл. Той може да бъде създаден с какъвто и да е текстов редактор. Той съдържа текста на документа и командите, които указват на L^AT_EX как да обработва текста.

1.3.1 Интервали

«Празните» символи, такива като интервал и табулиране, се интерпретират от L^AT_EX еднакво – като «интервал». Няколко последователни празни символи се възприемат като *един* «интервал». Празни символи в началото на реда обикновено се игнорират, а единично прекъсване на реда се възприема като «интервал».

Празен ред между два реда в текста определя края на параграфа. Няколко празни реда се възприемат също като *един* празен ред. Долу е приведен пример. Вляво е текстът от входния файл, вдясно — форматирания изход.

Няма значение дали оставяте един
или няколко интервала
между думите.

Празен ред поставя началото
на нов параграф.

Няма значение дали оставяте един или няколко интервала между думите.

Празен ред поставя началото на нов параграф.

1.3.2 Специални символи

Следващите символи са резервирани символи, които или имат в L^AT_EX специален смисъл, или не са достъпни във всички шрифтове. Ако ги въведете в текста направо, те обикновено не се печатат, а заставят L^AT_EX да направи нещо, което вие все още не знаете, но ще научите, ако внимателно четете тази книга.

\$ % ^ & _ { } ~ \

Ако искате да отпечатате тези символи във вашия документ, добавете пред тях префикс «\»:

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

\$ % ^ & _ { } ~

Тези символи, както и много, много други, могат да бъдат отпечатани със специални команди в математически формули или като акценти. Знакът «\» *не може* да се въвежда два пъти, тъй като тази команда (\\) се използва за прекъсване на реда.¹

1.3.3 Команди на L^AT_EX

Командите на L^AT_EX са чувствителни към горен и долен регистър и приемат една от следните две форми:

¹Вместо това използвайте командата `\backslash`. Тя дава «\».

- Те започват със символа backslash «\» и продължават с името, състоящо се само от букви. Имената на командите завършват с интервал, цифра, или всяка друга «не-буква».
- Те се състоят от «\» и точно един не-буквен символ.

L^AT_EX игнорира интервалите след командата. Ако искате да получите интервал след команда, трябва да поставите или «{ }» и интервал, или специална команда за интервал след името на командата. «{ }» не позволява на L^AT_EX да игнорира всички интервали след името на командата.

Кнут е разделял
хората, работещи с \TeX{} на
\TeX{}ници и \TeX перти.\n
Днес е \today

Кнут е разделял хората, работещи с T_EX на
T_EXници и T_EXперти.
Днес е 18 януари 2006 г.

Някои команди се нуждаят от параметър, който трябва да бъде зададен между фигурни скоби «{ }» след името на командата. Има команди, които поддържат незадължителни параметри, които се добавят след името на командата в квадратни скоби «[]». Следният пример използва някои команди на L^AT_EX. Не се замисляйте над тях, те ще бъдат разяснени по-късно.

Бъдете сигурни, че
\textsl{тази книга}
ще ви помогне да
научите \LaTeX.

Бъдете сигурни, че *тази книга* ще ви помогне да научите L^AT_EX.

Следващият текст е на нов ред.\newline
Текст, текст, \ldots

Следващият текст е на нов ред.
Текст, текст, ...

1.3.4 Коментарни редове

Когато в процеса на обработка на входния файл L^AT_EX срещне символа %, той игнорира остатъка от текущия ред, преминаването на нов ред и всички интервали в началото на следващия ред.

Това може да се използва за добавяне в изходния файл на бележки, които няма да се извеждат за печат.

Това е %глупав,%
%но все пак показателен%
пример: Непротиво%
конституцион%
ствувателствайте

Това е пример: Непротивоконституционствувателствайте

Знакът % може също да се използва за разбиване на даден ред на части на такива места, където не се допускат интервали или нов ред.

За по-дълги коментари може също да се използва средата `comment`, предоставена от пакета `verbatim`. За да се използва средата `comment`, трябва в преамбюла на документа да се добави командата `\usepackage{verbatim}`:

```
Ето още един
\begin{comment}
достатъчно банален, но полезен
\end{comment}
пример за поставяне
на коментари в документа.
```

Ето още един пример за поставяне на коментари в документа.

Важно е да се знае, че това няма да работи в сложна среда, например математическа формула.

1.4 Структура на входния файл

Когато $\text{\LaTeX} 2_{\epsilon}$ обработва входния файл, той очаква от него да следва определена структура. Така всеки входен файл трябва да започва с командата

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

Тя указва какъв тип документ започвате да пишете. След тази команда вие можете да включите и други команди, които влияят на стила на целия документ, или да зареждате пакети, добавящи нови възможности в системата \LaTeX . За зареждане на такъв пакет се използва командата

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

Когато цялата настройка е завършена,¹ започва тялото на документа след записване на командата

```
\begin{document}
```

След тази команда се въвежда текста, заедно с някои полезни команди на \LaTeX . В края на документа се добавя командата

```
\end{document}
```

Всичко, което следва след нея, се игнорира от \LaTeX .

Фигура 1.1 показва съдържанието на минималния $\text{\LaTeX} 2_{\epsilon}$ -файл. Малко по-сложен входен файл е даден на Фигура 1.2.²

¹Областта между `\documentclass` и `\begin{document}` се нарича *преамбюл*.

²В дадения пример е добавен минимален набор команди, необходими за поддръжка на български език. Тук се използва пакета `babel`, който е стандартно средство за езикова локализация на \TeX . Ако е необходимо, се консултирайте с вашия *Local Guide* [6] или с администратора. — Бел. ред.

```
1 \documentclass{article}
2 \usepackage[T2A]{fontenc}
3 \usepackage[cp1251]{inputenc}
4 \usepackage[bulgarian]{babel}
5 \begin{document}
6 Кратко и ясно.
7 \end{document}
```

Фигура 1.1: Минимален L^AT_EX-файл

```
1 \documentclass[a4paper,11pt]{article}
2 \usepackage[T2A]{fontenc}
3 \usepackage[cp1251]{inputenc}
4 \usepackage[bulgarian]{babel}
5 \begin{document}
6 % дефиниране на заглавието
7 \author{Д.~Иванов}
8 \title{Моята първа \LaTeX -статия}
9 \frenchspacing
10 \begin{document}
11 %генериране на заглавието
12 \maketitle
13 %поставяне на съдържание
14 \tableofcontents
15 \section{Начало}
16 Ето тук започва моята забележителна статия.
17 \section{Край}
18 \ldots{} и тук тя завършва.
19 \end{document}
```

Фигура 1.2: Макет на документ-статия

1.5 Типична сесия при работа с L^AT_EX

Сигурно вече нямате търпение да изпробвате малкия пример на L^AT_EX-файла, приведен на страница 8. Нека да опитаме.

Сам по себе си L^AT_EX не включва графичен интерфейс. Той е просто програма, обработваща указания файл. Някои дистрибутиви на L^AT_EX включват графична обвивка, където можете да изберете с мишката компилация на указания файл. В други системи трябва да набирате командите в команден режим. Следващото описание предполага, че на вашия компютър вече има работеща L^AT_EX-инсталация¹.

1. Редактирайте или създайте входния L^AT_EX-файл. Този файл трябва да бъде плосък ASCII текст. В Windows трябва да се убедите, че съхранявате файла във формат ASCII, или *Plain text*. Избирайки име на файла, добавете разширение `.tex`.
2. Компилирайте вашия входен файл с L^AT_EX. Ако всичко мине без грешки, ще получите файл `.dvi`. За да получите коректно съдържание и вътрешни препратки, може да се наложи да стартирате L^AT_EX няколко пъти. Ако във входния файл има грешка, L^AT_EX ще ви я покаже и ще спре обработката на входния файл. За да се върнете в командния ред, натиснете `ctrl-D`.

```
latex name.tex
```

3. Сега можете да видите файла `.dvi`. Това може да се направи по различни начини. Ако работите под Windows, опитайте YAP („yet another previewer“). Този визуализатор е включен в дистрибутива на MiKTeX. Можете също да преобразувате DVI-файла в POSTSCRIPT за печат или преглеждане с помощта на Ghostscript:

```
dvips -Pcmz name.dvi -o name.ps
```

Ако сте късметлия, вашата L^AT_EX система дори ще съдържа инструмента `dvipdf`, който позволява да преобразувате вашите `.dvi`-файлове направо в PDF.

```
dvipdf name.dvi
```

¹Вижте раздел "Инсталиране на MiKTeX".

1.6 Макет на документа

1.6.1 Класове документи

Първото, което L^AT_EX трябва да знае при обработката на входния файл, е типа на създавания от автора документ. Той се задава с командата `\documentclass`.

```
\documentclass[опции]{клас}
```

Тук *клас* определя типа на създавания документ. Таблица 1.1 изброява класовете документи, разглеждани в тази книга. Дистрибуцията на L^AT_EX 2_ε осигурява допълнителни класове за други документи, включително писма и слайдове. Параметърът *опции* изменя поведението на класа документ. Опциите трябва да се разделят със запетая. В Таблица 1.2 са изброени най-употребяваните опции на стандартните класове документи.

Таблица 1.1: Класове документи

<code>article</code>	За статии и научни журналы, презентации, кратки отчети, програмна документация, покани. . .
<code>report</code>	За по-дълги отчети, съдържащи няколко глави, неголеми книжки, дисертации. . .
<code>book</code>	За книги
<code>slides</code>	За слайдове. Използват се големи букви без пренасяне. Вместо това може да се използва FoilL ^A T _E X. ^a

^aCTAN:/tex-archive/macros/latex/contrib/supported/foiltex

Пример: Входният файл на L^AT_EX-документа може да започва с реда

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

Той заставя L^AT_EX да набира документа като *статия*, с основен размер на шрифта *единадесет точки* и да форматира документа за *двустранен* печат на хартия с *формат A4*.

1.6.2 Пакети

Докато пишете вашия документ, вероятно ще забележите, че има някои области, в които базовия L^AT_EX не може да реши вашите проблеми. Ако

Таблица 1.2: Опции на класовете документи

<code>10pt, 11pt, 12pt</code>	Установява размера на основния шрифт на документа. Ако нито една от тези опции не е зададена, се подразбира <code>10pt</code> .
<code>a4paper, letterpaper...</code>	Определя размера на страницата. По подразбиране е <code>letterpaper</code> . Също могат да бъдат указани <code>a5paper, b5paper, executivepaper</code> и <code>legalpaper</code> .
<code>fleqn</code>	Формулите на отделен ред да се отпечатват подравнени вляво, а не центрирани.
<code>leqno</code>	Поставя номерата на формулите вляво, а не вдясно.
<code>titlepage, notitlepage</code>	Определя дали трябва да започне нова страница след заглавието на документа или не. По подразбиране класът <code>article</code> не започва нова страница, а <code>report</code> и <code>book</code> — започват нова страница.
<code>onecolumn, twocolumn</code>	Инструктира L ^A T _E X да отпечата документа в една колона или в две колони.
<code>twoside, oneside</code>	Избира едно- или двустранен изход. По подразбиране класът <code>article</code> и <code>report</code> използват едностранен изход, класът <code>book</code> — двустранен изход. Забележете, че тези опции се отнасят само до типа на документа. Опцията <code>twoside</code> не застава вашия принтер да печата двустранно.
<code>landscape</code>	Сменя положението на страницата на ландшафтно.
<code>openright, openany</code>	Установява главите да започват или само на дясната страница, или на първата достъпна. Това не работи с класа <code>article</code> , тъй като той не знае нищо за главите. Класът <code>report</code> по подразбиране започва глава на следващата страница, а класът <code>book</code> — на дясната.

искате да включите в документа графика, цветен текст или изходен код на програма от външен файл, вие трябва да разширите възможностите на \LaTeX . Такива разширения се наричат пакети. Пакетите се активират с командата

```
\usepackage[опции]{пакет}
```

където *пакет* е името на пакета, а *опции* – списък от ключови думи, включващи специални възможности на пакета. Някои пакети са включени в основния дистрибутив на $\LaTeX 2_{\epsilon}$ (виж Таблица 1.3). Други се предоставят отделно. Допълнителна информация за установените пакети може да намерите във вашия *Local Guide* [6]. Основен източник на информация за \LaTeX -пакетите е *The \LaTeX Companion* [3]. Тя съдържа описание на стотици пакети, както и описание на това, как да пишете ваши собствени разширения за $\LaTeX 2_{\epsilon}$.

1.6.3 Стилоре на страницата

\LaTeX поддържа три предопределени комбинации на горния и долния колонтитул на страницата, така наречените стилове на страницата. Параметърът *стил* на командата

```
\pagestyle{стил}
```

определя кой от тях да се използва. Предопределените стилове на страницата са изброени в Таблица 1.4.

Възможно е да се смени стила на текущата страница с командата

```
\thispagestyle{стил}
```

Описание на това, как да създавате ваши собствени стилове на колонтитулите има в *The \LaTeX Companion* [3], а също и в Раздел 4.5 на страница 79.

1.7 Разширения на файлове, които се срещат при работа с \LaTeX

Работейки с \LaTeX , скоро ще се окажете в лабиринт от файлове с различни разширения и вероятно – никакво указание. Следният списък обяснява различните типове файлове, използвани при работа с \TeX .

.tex Входен файл на \LaTeX или \TeX . Може да бъде компилиран с командата `latex`.

Таблица 1.3: Някои от разпространяваните с L^AT_EX пакети

<code>doc</code>	Позволява да се документират програми на L ^A T _E X. Описан в <code>doc.dtx</code> ^a и в <i>The L^AT_EX Companion</i> [3].
<code>exscale</code>	Предоставя мащабираните версии на разширени математически шрифтове. Описан в <code>ltxscale.dtx</code> .
<code>fontenc</code>	Определя каква кодировка на шрифта трябва да използва L ^A T _E X. Описан в <code>ltoutenc.dtx</code> .
<code>ifthen</code>	Предоставя команди от вида “ако ..., то изпълни ..., иначе изпълни ...”. Описан в <code>ifthen.dtx</code> и в <i>The L^AT_EX Companion</i> [3].
<code>latexsym</code>	За включване на шрифт със специални символи на L ^A T _E X, трябва да се използва пакета <code>latexsym</code> . Описан в <code>latexsym.dtx</code> и в <i>The L^AT_EX Companion</i> [3].
<code>makeidx</code>	Предоставя команди за генерация на азбучен указател. Описан в Раздел 4.3 и в <i>The L^AT_EX Companion</i> [3].
<code>syntonly</code>	Обработва документа, без да го отпечата.
<code>inputenc</code>	Позволява да се указва входната кодировка, такава като ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows или определена от потребителя. Описан в <code>inputenc.dtx</code> .

^aТози файл трябва да бъде инсталиран във вашата система и вие можете да получите `dvi` файл, като дадете командата `latex doc.dtx` във всеки каталог, където имате права за запис. Същото се отнася за всички подобни файлове, споменати в тази таблица.

Таблица 1.4: Предопределени стилове на L^AT_EX-страницата

<code>plain</code>	Печата номера на страницата долу, в средата на долния колонтитул. Този стил е по подразбиране.
<code>headings</code>	Печата името на текущите глава или раздел и номера на текущата страница в горния колонтитул, докато долният колонтитул остава празен. (Този стил е използван в този документ.)
<code>empty</code>	Установява и долния, и горния колонтитул на страницата да са празни.

`.sty` Макро-пакет на L^AT_EX. Този файл можете да заредите във вашия L^AT_EX-документ с използване на командата `\usepackage`.

`.dtx` Документиран T_EX. Това е основен формат на разпространяваните стилови файлове на L^AT_EX. Ако обработите файла `.dtx` с L^AT_EX, ще получите документиран макро-код на L^AT_EX-пакета, съдържащ се в `.dtx` файла.

`.ins` Инсталатор за файловете, съдържащи се в съответния файл `.dtx`. Ако изтеглите пакет от Интернет, обикновено получавате файлове `.dtx` и `.ins`. Компилирайте с L^AT_EX файла `.ins`, за да разопаковате файла `.dtx`.

`.cls` Файл за класа, определящ вида на вашия документ. Избира се с командата `\documentclass`.

`.fd` Файл с описание на шрифт, описващ на L^AT_EX нов шрифт.

Следващите файлове се генерират, когато L^AT_EX обработва входния файл:

`.dvi` Device Independent file (файл, независещ от устройството). Това е основен резултат от компилирането с L^AT_EX. Съдържанието му може да се види с програмата за визуализиране на DVI, или като изпрати към принтера с `dvips` или подобно приложение.

`.log` Дава детайлен отчет за това, какво се е случило при последната компилация.

`.toc` Съхранява заглавията на всички раздели. Чете се при следващата компилация и се използва при генерацията на съдържанието.

`.lof` Подобен на `.toc` списък от фигури.

- .lot И отново същото за списък от таблици.
- .aux Още един файл, предаващ информация от една компилация към следващата. Освен всичко друго, .aux файла се използва за съхраняване на информация, свързана с препратките.
- .idx Ако вашият документ има азбучен указател, L^AT_EX съхранява всички думи за указателя в този файл. Обработете този файл с програмата `makeindex`. Подробно вижте в Раздел 4.3 на страница 76.
- .ind Обработен файл .idx, готов за включване във вашия документ при следващата компилация.
- .ilg Журнал за работата на `makeindex`.

1.8 Големи проекти

При създаване на големи документи (доклади, книги), е удобно отделните части да се създават самостоятелно, след което да се свържат в единен документ посредством *Главен файл*. L^AT_EX има две команди, които ще ви помогнат да направите това. Командата

```
\include{filename}
```

се използва в тялото на *Главния файл*, за да се включи в него съдържанието на файл *filename.tex*. Забележете, че L^AT_EX започва нова страница, преди да обработва материала, съдържащ се във *filename.tex*.

Следващата команда може да се използва само в преамбюла. Тя указва на L^AT_EX да чете само някои от включените (с `\include`) файлове.

```
\includeonly{filename1,filename2,...}
```

След задаване в преамбюла на документа, тази команда ще изпълнява само тези команди `\include`, чиито имена на файлове са изброени в аргумента на командата `\includeonly`. Важно е да се отбележи, че между имената на файловете и запетаите не трябва да има празно място.

Командата `\include` започва обработка на включения текст на нова страница. Това е удобно при използване на `\includeonly`, тъй като границите на страниците няма да се преместват, дори ако някои от включените файлове се пропускат. Когато това е нежелателно, се използва командата

```
\input{filename}
```

Тя просто включва съдържанието на указания файл от същата позиция (а не на нова страница).

За бърза проверка на синтаксиса на командите в проекта, се използва пакетът `syntonly`. Той заставя \LaTeX да пробяга по документа, проверявайки синтаксиса и използваните команди, без да генерира изходен файл DVI. \LaTeX работи по-бързо в този режим, икономисвайки ценното ви време. Синтаксисът е много прост:

```
\usepackage{syntonly}
\syntonly
```

След корекция на евентуални грешки, за получаване на изходен файл, коментирайте с `%` втория ред, и компилирайте отново с \LaTeX .

1.9 Инсталиране на MiKTeX

MiKTeX е дистрибутив на \LaTeX за платформа Win32, включващ трансляторите на \TeX и \LaTeX с пълен набор стандартни пакети и множество допълнителни програми:

YAP – удобен визуализатор за преглеждане на DVI-файлове;

pdfTeX и **pdfLaTeX** – за получаване на изходен документ във формат PDF (вместо DVI);

METAPOST — мощна система за създаване на векторни рисунки във формат POSTSCRIPT;

dvips – за конвертиране на DVI-файл в POSTSCRIPT-файл;

Огромно количество документация.

Изтеглянето на MiKTeX от www.miktex.org е лесно, но за пълноценна работа на системата е необходимо да се инсталират допълнителни програми.

Последователността на инсталиране е: **Ghostscript** и **GSView**, **Adobe Acrobat Reader**, **MiKTeX**, **Shell**.

- Най-напред трябва да се установят **GSView** и **Ghostscript**, предназначени за работа с POSTSCRIPT шрифтове и векторни картинки във формат `.ps` и `.eps`. Програмата **Ghostscript** е необходима и за коректна работа на dvi-визуализатора YAP.

Дистрибутивът на **Ghostscript** е в каталога <http://www.cs.wisc.edu/~ghost/>. Програмата е безплатна.

- Необходимо е и инсталирането на програмата **Adobe Acrobat Reader** от каталога <http://www.adobe.com/products/acrobat/> за преглеждане на PDF документи, станали фактически стандарт за представяне на научно-техническа информация (и в Internet). PDF документи се получават с помощта на програмите **pdfTeX** и **pdfLaTeX** от дистрибуцията на MiKTeX. Програмата е безплатна.
- Следва инсталиране на **MiKTeX**. В началото на 2005 г. на официалния сайт на проекта <http://www.miktex.org> текуща версия е 2.4. За инсталация трябва да се стартира файлът **setup.exe**. Инсталирането се ръководи от **MiKTeX Setup Wizard**. Има три типа установяване: **Small**, **Large** и **Total**. По възможност се избира установяване от типа **Total**, включващ **Type1** шрифтове за кирилица **cm-super**.

При инсталирането, **MiKTeX** изгражда две важни дървета:
`c:\texmf` и `c:\localtexmf`

MiKTeX е безплатен и свободно разпространяван продукт.

1.10 Инсталиране на Shell (обвивка, текстов редактор)

Идеологията и устройството на TeX са такива, че е без значение как се получава изходният `tex`-файл. Няма значение и операционната система. Изискването е това да е обикновен текстов файл. По тази причина в дистрибутива на MiKTeX няма никаква обвивка (за да не се натрапва на потребителя интерфейс).

Съществуват десетки специализирани текстови редактори, специално разработени за набор в TeX. Голяма част от тях са безплатни:

WinEdt (<http://www.winedt.com>)

WinEdt е една от най-удобните и използвани обвивки за работа с TeX. WinEdt се интегрира автоматично с вече инсталирания MiKTeX. Настройва се всичко: проверка на синтаксиса, включване на речници, възможност за писане на макрокоманди, удобство при набора на формули. Продуктът има само един недостатък – ползва се свободно един месец, след което продължава да работи, но напомня, че е необходима регистрация.

Конфигурирането на WinEdt за връзка със системата MiKTeX се извършва, като се редактира файлът
`c:\texmf\miktex\config\miktex.ini`.

Под [TeX] - **environment** променливите има списък с редактори. Редът, съдържащ WinEdt трябва да е некомментиран, като буквата

Е на `Editor` е в първа колона. Коментира се редът на съществуващия редактор (вероятно `Notepad`).

TeXshell (<http://www.projectory.de>)

TeXshell е безплатна обвивка, която по своите интерфейсни възможности силно отстъпва на `WinEdt`, но има малък обем и е проста в настройката. Има едно предимство – не е необходима инсталация. Достатъчно е да се копира на диска в съответния каталог.

TeXnicCenter (<http://www.toolscenter.org>)

TeXnicCenter е безплатен продукт. По удобство за набор на формули много отстъпва на `WinEdt`, но не е необходима инсталация.

WinShell (<http://www.winshell.de>)

WinShell е безплатна обвивка. Интерфейсните му възможности са по-развити от тези в `TeXShell`, малък е по обем. Има проверка на синтаксиса и меню за стартиране на външни програми.

Още над 40 препратки към различни обвивки и полезните конвертори `word2tex` и `tex2word` се намират на адрес <http://home.germany.net/100-122054/texwin.htm>

1.11 Езикова настройка в MiKTeX

Използването в MiKTeX на език, различен от английски, е усъвършенствано чрез пакета `babel` (Johannes Braams) за повече от 50 езика. За съжаление обаче, в дистрибуцията на MiKTeX 2.4, файлът `bghyph.tex` за правилата на пренасяне в български език е погрешно заменен с друг, което създава допълнителни трудности.

Ако искате да пишете документ на език, различен от английски (в случая български) трябва да се погрижите за следното:

- MiKTeX да бъде конфигуриран в две насоки:
 1. Всички автоматично генерирани текстови стрингове (Таблица, Съдържание, Фигура, Литература и др.) трябва да бъдат преведени на съответния език;
 2. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ трябва да узнае правилата за пренасяне на езика. Установяването на тези правила в MiKTeX е по-сложно и включва обновяване на форматните файлове.
- Редакторът `WinEdt` трябва да бъде конфигуриран за кирилица;
- В преамбула на всеки документ да се включват команди за използване на пакетите `babel`, `inputenc` и `fontenc` със съответните кодировки за кирилица.

Подробно, последователността от стъпки, чрез които успешно може да се инсталира български език, е:

Конфигуриране на MiKTeX

- Добавя се нов ред във файла
c:\texmf\tex\generic\config\language.dat

```
bulgarian bghyph.tex
```

Коментират се редовете с ненужните езици, съхранява се;

- Създават се две папки:
c:\localtexmf\tex\generic\babel\
c:\localtexmf\tex\generic\hyphen\
- Копират се файловете `bulgarian.ins` и `bulgarian.dtx`
от c:\texmf\source\generic\babel\
в c:\localtexmf\tex\generic\babel\
и файлът `bulgarian.sty` от c:\texmf\tex\generic\babel\
се копира в c:\localtexmf\tex\generic\babel\
- Копират се файловете
`bghyphen.tex`, `bghyphsi.tex`, `mik2t2.tex`, `catmik.tex`
от c:\texmf\source\generic\babel\
в c:\localtexmf\tex\generic\hyphen\
- Редактира се c:\texmf\tex\generic\babel\bulgarian.ldf
и се поправя (ако има синтактична грешка)
ред 218: \def\@SS{\mathchar''7019}
- От командния прозорец се изпълняват командите:

```
cd c:\localtexmf\tex\generic\babel
latex.exe bulgarian.ins
```

- Обновяват се базите данни:
Start→**Programs**→**MiKTeX**→**MiKTeX Options**→**Refresh**
- За да се използва в MiKTeX български език, се създават наново форматните файлове:

Start→**Programs**→**MiKTeX**→**MiKTeX Options**

– **Languages**→**New**

* Добавя се нов език **bulgarian**

- * Указват се правилата за пренасяне:
с:\localtexmf\tex\generic\hyphen\bghyphen.tex
- * Изключват се ненужните езици, като се оставят english,
german, french, nohyphen;
- **Formats**
Премахват се ненужните формати, като се оставят **tex**, **latex**,
pdftex, **pdflatex**.
- Обновяват се всички форматни файлове.

General→**Update Now**

Установяване на Cyrillic Font в WinEdt

От менюто на WinEdt: **Options**→**Fonts**→**Courier Cyrillic**

Глава 2

Обработка на текста

След прочитане на първата глава, вие трябва да знаете от какво се състои \LaTeX -документа. Втората глава ще добави в тази структура всичко, от което се нуждаете, за да изработвате истински статии или книги с \LaTeX 2_ϵ .

2.1 Структура на текста и езика

Основната цел на автора при написването на текст, е да се предадат на читателя идеи, информация или знания. Читателят по-добре ще разбере текста, ако тези идеи са структурирани, и най-добре той ще види и почувства тези структури, ако типографската форма отразява логическата и смислова структура на съдържанието.

\LaTeX се различава от другите системи за обработка по това, че вие трябва само да му задавате логическата и смислова структура на текста. Той след това избира типографската форма на текста в съответствие с «правила», зададени във файла на класа на документа и в различни стилни файлове.

Най-важният елемент на текста в \LaTeX (и в типографията въобще) е параграфът. Той се нарича «елемент на текста», защото параграфът е тази типографска форма, в която се формулира една свързана мисъл или идея.

В следващите раздели вие ще разберете как можете да отивате на нов ред, например, с помощта на \backslash и как да определите началото на параграф, например като оставите празни редове. Основно правило: ако започвате нова мисъл, трябва да започнете нов параграф, а ако не – трябва да преминете на нов ред. Ако се съмнявате във вашето разбиране на параграфи, помислете за текста като носител на идеи и мисли. Ако сте започнали нов параграф, а продължавате изложението на същата мисъл, то махнете началото на новия параграф. Ако вътре в параграфа започва свършено нова мисъл, разделете го на два параграфа.

Много хора изобщо не разбират колко е важно разумното разбиване на текста на параграфи. Много от тях не се замислят за значението на прекъсването на параграфа или, особено в \LaTeX , започват параграфи, без да осъзнават това. Последната грешка се прави лесно, когато в текст се включват уравнения. Вижте следните примери и ще разберете, защо понякога се използват празни редове (прекъсване на параграфа), а понякога – не (ако вие още не разбирате всички команди достатъчно, за да разберете примерите, прочетете тази и следващите глави и след това се върнете към примерите отново.)

```
\ldots Квадратният тричлен
е многочлен от втора степен
\begin{equation}
ax^2+bx+c\; ,
\end{equation}
където  $a \neq 0$ .
```

... Квадратният тричлен е многочлен от втора степен

$$ax^2 + bx + c, \quad (2.1)$$

където $a \neq 0$.

```
Числата на Фибоначи се
задават с начални стойности
 $u_1=u_2=1$  и рекурентна
зависимост
\begin{equation}
u_{n+1}=u_{n-1}+u_n \; .
\end{equation}
```

Първите 14 числа на Фибоначи са публикувани през 1228 от \ldots

Числата на Фибоначи се задават с начални стойности $u_1 = u_2 = 1$ и рекурентна зависимост

$$u_{n+1} = u_{n-1} + u_n. \quad (2.2)$$

Първите 14 числа на Фибоначи са публикувани през 1228 от ...

```
\ldots е уравнение на
елипсоид.
```

```
\begin{equation}
\frac{x^2}{a^2}+
\frac{y^2}{b^2}+
\frac{z^2}{c^2} =1
\end{equation}
се казва, че е в
каноничен вид
\ldots
```

... е уравнение на елипсоид.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \quad (2.3)$$

се казва, че е в каноничен вид ...

Следващата, по-малка единица на текста е изречението. В английския текст след точка, завършваща изречение, се слага по-голям интервал, отколкото след точка, стояща след съкращение. \LaTeX се старая да определи, коя от тях сте имали предвид. Ако той греша, вие трябва да

му подскажете. Това е обяснено по-нататък в тази глава на стр. 34. Всички параграфи в текста също трябва да бъдат логически структурирани на по-високо ниво, обединявайки се в глави, раздели, и така нататък. Команда от типа `\section{Структура на текста и езика}` поставя начало на нов раздел. По същия начин се използват командите за начало на нова глава, или подраздел, за които ще стане въпрос по-нататък.

2.2 Разбиване на редове и страници

2.2.1 Подравнени параграфи

Книгите често се оформят така, че всички редове да имат еднаква дължина. \LaTeX поставя необходимите прекъсвания на редовете и интервали между думите, като оптимизира форматирането на параграфа като цяло. Ако е необходимо, той освен това пренася думи, които не се събират на реда. От класа на документа зависи това, как ще се форматира параграфа. Обикновено първият ред на всеки параграф започва с отстъп, а допълнително разстояние между два параграфа няма. Подробности вижте в Раздел 6.3.2.

В някои случаи може да е необходимо да се укаже на \LaTeX да прекъсне реда. След командите

```
\ или \newline
```

започва нов ред, без да започва нов параграф. Командата

```
\*
```

забранява освен това прекъсването на страницата след поставеното прекъсване на реда. Командата

```
\newpage
```

започва нова страница.

Всяка от следните команди съответно

```
\linebreak[n], \nolinebreak[n], \pagebreak[n] и \nopagebreak[n]
```

прекъсва реда, забранява прекъсването на реда, прекъсва страницата и забранява прекъсването на страницата. Те дават възможност на автора да влияе на тяхното действие, посредством незадължителния аргумент n , който може да бъде равен на число от нула до четири. Ако дадете за n число, по-малко от 4, вие оставяте на \LaTeX правото на избор, да игнорира вашите команди, ако резултатът бъде много лош. Не бъркайте

командите за “прекъсване“ (`break`) с командите за “начало“ (`new`). Дори ако давате команда «прекъсване», \LaTeX ще се опита да подравни дясната граница на страницата и общата височина на страницата, както е описано в следващия раздел. Ако вие действително искате да започнете «нов ред», използвайте съответните команди.

\LaTeX винаги се опитва да избере най-добрия от възможните варианти за форматиране на реда. Ако не може да намери начин да форматира реда в съответствие със своите стандарти, той позволява на един ред да отстъпи от параграфа надясно. След това \LaTeX извежда диагностика («`overflow hbox`») по време на обработката на входния файл. Най-често това се случва, когато \LaTeX не може да намери подходящо място за пренасяне на дума.¹ Можете да инструктирате \LaTeX да отслаби малко своите стандарти, ако дадете команда `\sloppy`. Това ще предотврати такива прекалено дълги редове, чрез увеличаване на интервалите между думите, дори ако крайният резултат не е оптимален. В този случай потребителят получава предупреждение («`underfull hbox`»). В голяма част от случаите резултатът не изглежда много добре. Командата `\fussy` връща \LaTeX към неговото поведение по подразбиране.

2.2.2 Пренасяне

\LaTeX пренася думите, когато това е необходимо. Ако алгоритъмът за пренасяне не може да намери точните места за пренасяне, можете да се справите със ситуацията, информирайки \TeX за изключенията, с помощта на следните команди.

Командата

```
\hyphenation{списък от думи}
```

предизвиква пренасяне на думите, изброени в нейния аргумент, само на местата, отбелязани с «-». Аргументът на командата трябва да съдържа само думи, състоящи се от обикновени букви, или по-скоро знаци, които се разглеждат от \LaTeX като нормални букви. Правилата на пренасяне се запомнят за езика, който е активен в момента на обръщане към командата `\hyphenation`. Това означава, че ако поставите командата за пренасяне в преамбюла на документа, тя ще влияе на английските преноси. Ако вие я поставите след командата `\begin{document}` и използвате при това някакъв пакет за поддръжка на национален език, например `babel`, то правилата за пренасяне ще бъдат активни за езика, активиран с `babel`.

¹Макар, че \LaTeX ви дава предупреждение и номера на реда, такива редове не винаги се откриват лесно. Ако използвате в командата `\documentclass` опция `draft`, такива редове ще бъдат отбелязани с дебели черни линии в десните полета.

В следващия пример се разрешава пренасяне в думата «hyphenation», както и в думата «Hyphenation», и се забранява пренасянето в думите «MATLAB», «Matlab» и «matlab». В аргумента не се допускат специални символи.

Пример:

```
\hyphenation{MATLAB Hy-phen-a-tion}
```

Командата \- поставя в думата избран пренос. Тя също така установява единствените разрешени точки за пренасяне в тази дума. Тази команда е особено полезна за думи, съдържащи специални символи (например, символи с акценти), защото ЛАТ_ЕХ не пренася такива думи автоматично.

```
Дали това е про\~ти\~во\~%
кон\~сти\~ту\~ци\~он\~но или
не\~про\~ти\~во\~кон%
\~сти\~ту\~ци\~он\~но.
```

Дали това е противоконституционно или не-противоконституционно.

Няколко думи могат да се задържат заедно на един ред с командата

```
\mbox{текст}
```

Тя заставя ЛАТ_ЕХ да държи аргумента цял, при всякакви обстоятелства.

```
Телефонният ми номер е
\mbox{0324 222 336677}.
```

Телефонният ми номер е 0324 222 336677.

```
Параметърът
\mbox{\emph{брой аргументи}}
е незадължителен.
```

Параметърът *брой аргументи* е незадължителен.

Командата \fbox е аналогична на командата \mbox, но в добавка, около нейния аргумент се рисува видима рамка.

2.3 Специални стрингове

В някои от примерите на предните страници, вече сте виждали някои много прости команди на ЛАТ_ЕХ за отпечатване на специални текстови стрингове:

Команда	Пример	Описание
\today	18 януари 2006 г.	Текуща дата на текущия език
\TeX	Т _Е Х	Име на любимата ви издателска система
\LaTeX	Л _А Т _Е Х	Името на Л _А Т _Е Х
\LaTeXe	Л _А Т _Е Х 2 _ε	Текуща версия на Л _А Т _Е Х

2.4 Специални букви и символи

2.4.1 Знаци кавички

За набора на кавички *не трябва* да използвате знака `"`, както на пишеща машина. При обработка на текста съществуват специални знаци отварящи и затварящи кавички. В \LaTeX използвайте два знака `'` за отварящи кавички и два знака `'` - за затварящи. За единични кавички, просто използвайте един от тях.

`'`след команда, натиснете
клавиш `'Enter'` `'`

“след команда, натиснете клавиш `'Enter'`”

При обработка обикновено се изискват две двойки кавички, при случай на „кавички“ в кавички». При обработка на български език обикновено вместо “такива” и ‘такива’ кавички се използват «тези» и „тези“. Те се задават обикновено с командите `\flqq`, `\frqq` (или `«` и `»`) и `\glqq`, `\grqq`.

2.4.2 Тире и дефис

\LaTeX «знае» за четири вида тире. Три от тях можете да получите с различен брой последователни знаци `-`. Четвъртото всъщност не е тире изобщо, а математическия знак минус:

`dauther-in-low`, X-лъчи`\`
страници 13--67`\`
да~--- или не?`\`
`0`, `1` и `-1`

`dauther-in-low`, X-лъчи
страници 13-67
да — или не?
0, 1 и -1

Имената на тези тирета са: `-` дефис, `--` късо тире, `---` дълго тире и `$-$` знак минус.

2.4.3 Тилда (`~`)

Често срещан символ в WWW-адреси е символът тилда. За неговото генериране в \LaTeX може да се използва `\~`, но резултатът няма да бъде това, което желаете. Можете да използвате това:

`http://www.bulgaria.bg/\~{}photos` `\`
`http://www.dechica.com/\simdemo`

`http://www.bulgaria.bg/~photos`
`http://www.dechica.com/~demo`

2.4.4 Знак за градус (°)

Ето как се печата знак за градус в L^AT_EX:

Температурата вече е
`$$-30\text{\textcircled{C}}$`.

Температурата вече е -30°C .

Пакетът `textcomp` позволява достъп до символа градус с командата `\textcelsius`.

2.4.5 Символ Евро (€)

Когато пишете за пари в наши дни, се нуждаете от символа на евро-валутата. Много съвременни шрифтове съдържат символа Евро. След зареждане на пакета `textcomp` в преамбюла на документа:

```
\usepackage{textcomp}
```

можете да използвате командата

```
\texteuro
```

за достъп до този символ.

Ако вашият шрифт няма собствен *Евро* символ, или на вас не ви харесва как той изглежда, имате още два варианта.

Първият е пакета `eurosym`. Той осигурява официалния символ Евро:

```
\usepackage[official]{eurosym}
```

Ако предпочитате символ Евро, който съответства на вашия шрифт, то вместо опцията `official` използвайте опция `gen`.

Ако на вашата система е инсталиран Adobe Eurofonts (той е свободно достъпен от <ftp://ftp.adobe.com/pub/adobe/type/win/all>), вие можете да използвате или пакета `europs` и командата `\EUR` (за Евро символ, съответстващ на текущия шрифт), или пакета `eurosans` и командата `\euro` (за официалния символ Евро).

Пакетът `marvosym` осигурява множество различни символи, в това число и Евро, под името `\EUR`. Неудобно е това, че не се предоставят наклонени и дебели варианти на символа Евро.

2.4.6 Многоточие (...)

На пишеща машина, точка или запетая заемат същото пространство, както и всяка друга буква. Напечатани в книга, тези символи заемат

Таблица 2.1: Пълен набор на Евро символи

пакет	команда	roman	sans-serif	typewriter
eurosym	<code>\euro</code>	€	€	€
[gen]eurosym	<code>\euro</code>	€	€	€
europs	<code>\EUR</code>	€	€	€
eurosans	<code>\euro</code>	€	€	€
marvosym	<code>\EUR</code>	€	€	€

малко място и се разполагат много близо до предходната буква. Следователно, вие не можете да въведете многоточие просто като напечатате три точки, тъй като разстоянието между тях ще бъде неправилно. Затова, за многоточие има специална команда

```
\ldots
```

```
не така ... , а така:\  
едно, две, три, \ldots
```

```
не така ... , а така:  
едно, две, три, ...
```

2.4.7 Акценти и специални символи

L^AT_EX поддържа използването на акценти и специални символи за много езици. Таблица 2.2 показва всевъзможните акценти, приложени към буквата о. Ясно е, че на нейно място могат да бъдат и други букви.

За да се постави знак за акцент над буквите *i* или *j*, точките над тях трябва да бъдат премахнати. Това става с командите `\i` и `\j`.

```
H\^otel, na\"i ve, \'el\'eve,\  
sm\o rrebr\o d, !'Se\~norita!,\  
Sch\"onbrunner Schlo\ss}  
Stra\ss e
```

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

2.5 Поддръжка на чужди езици

Когато пишете документ на език, различен от английски език, L^AT_EX трябва да бъде конфигуриран по подходящ начин в три насоки:

1. Всички автоматично генерирани текстови стрингове¹ трябва да бъдат преведени на другия език. За много езици тези изменения се постигат с използване на пакета `babel` (автор Johannes Braams).
2. \LaTeX трябва да знае правила за пренасяне за новия език. Включването на правилата за пренасяне в \LaTeX е малко по-сложно. То включва обновяване на форматния файл с други разрешени образци на пренасяне. Вашият *Local Guide* [6] трябва да съдържа повече информация за това.
3. Специфичните за езика типографски правила. Например, за френския език, преди всеки символ двоеточие (:) трябва да стои интервал.

Ако вашата система вече е конфигурирана по съответния начин, можете да активирате пакета `babel` с добавяне на командата

```
\usepackage[език]{babel}
```

след командата `\documentclass`. Списък на всички *езици*, които поддържа вашата \LaTeX система, се изобразява всеки път, когато се стартира компилатора. `Babel` автоматично активира правилата за пренасяне на избрания от вас език. Ако форматния файл на вашия \LaTeX не поддържа пренасянето за избрания език, `babel` ще работи, но ще забрани пренасянето, което ще се отрази отрицателно на външния вид на документа.

`Babel` освен това въвежда нови команди за някои езици, които опростяват въвеждането на специални символи. Например, немският език съдържа много знаци `umlaut` (äöü). С използване на `babel`, вие можете да въведете ö, печатайки "o вместо `\"o`.

¹Съдържание, Списък на фигурите, Библиография ...

Таблица 2.2: Акценти и специални символи

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

Ако извикате `babel`, посочвайки няколко езика,

```
\usepackage[език1,език2]{babel}
```

то за избор на текущия език се използва командата

```
\selectlanguage{език1}
```

Голяма част от съвременните компютърни системи позволяват да се въвеждат специални символи директно от клавиатурата. За да обработва различните входни кодировки, използвани за различните групи езици и/или на различни компютърни платформи, L^AT_EX използва пакета `inputenc`:

```
\usepackage[кодировка]{inputenc}
```

Когато използвате този пакет, вие трябва да имате предвид, че е възможно, други хора да не могат да видят коректно вашите входни файлове на своя компютър, поради използването на друга кодировка. Например, немският акцент `ä` в система OS/2 се кодира като 132, на Unix системи, използващи ISO-LATIN 1, се кодира като 228, докато в `cyrillic encoding cp1251` за Windows, този символ не съществува изобщо. Следователно, трябва да използвате тези възможности с внимание. В зависимост от типа на използваната от вас система, могат да се използват следните кодировки:¹

Операционна система	кодировка	
	western latin	cyrillic
Mac	applemac	macukr
Unix	latin1	koi8-ru
Windows	ansinew	cp1251
DOS, OS/2	cp850	cp866nav

Важна е и кодировката на шрифта. Тя определя, в каква позиция вътре в T_EX-шрифта, се намира всяка буква.

Кодировките на шрифтовете се обработват с пакета `fontenc`:

```
\usepackage[кодировка]{fontenc}
```

където *кодировка* е изискваната кодировка на шрифта. Могат едновременно да се зареждат няколко кодировки.

¹За по-задълбочено изучаване на поддържаните входни кодировки за езици, основани на Латиница и Кирилица, прочетете съответната документация `inputenc.dtx` и `cyinrenc.dtx`. Раздел 4.7 обяснява как се получава печатна версия на документацията.

Кодировката на \LaTeX по подразбиране е OT1, кодировката на оригиналния шрифт на \TeX -Computer Modern. Тя съдържа само 128 символи от 7-битния набор символи ASCII. Когато се изискват акцентирани символи, \TeX ги създава, комбинирайки нормален символ с акцент. Въпреки, че резултатът изглежда идеално, този подход спира автоматичното пренасяне вътре в дума, съдържаща акцентирани символи. Освен това, някои латински букви не могат да се създадат чрез комбиниране на нормален символ с акцент, да не говорим за буквите на не-латинските азбуки, като Гръцката азбука или Кирилицата.

За преодоляване на тези ограничения, са били създадени няколко 8-битни набори шрифтове, подобни на CM. *Extended Cork* (EC) шрифтовете в кодировка T1 съдържат букви и препинателни знаци за повечето европейските езици, основани на Латиница. Наборът шрифтове LN съдържа необходимите букви за отпечатване на документи на езици, използващи Кирилица. Поради големия брой глифове на кирилица, те са организирани в четири кодировки на шрифта: T2A, T2B, T2C и X2.¹ Пакетът CB съдържа шрифтове в кодировка LGR за набор на гръцки текстове.

Използвайки тези шрифтове, вие можете да подобрите, или въобще да направите възможно пренасянето в не-английски документи. Още едно преимущество при използването на новите CM-подобни шрифтове е, че те предоставят шрифтове от фамилията CM от всички плътности, форми и оптически мащабирани размери на шрифта.

2.5.1 Поддръжка на френски език

Ето как могат да се създават в \LaTeX френски документи. Можете да заредите поддръжката на френски език с командата

```
\usepackage[frenchb]{babel}
```

По исторически причини името на опцията в `babel` за френски език е `frenchb` или `français`, но не `french`.

Горната команда включва френските правила за пренасяне, ако вашата \LaTeX -система е правилно настроена. Освен това, всички автоматично генерирани стрингове стават на френски: `\chapter` отпечатва `Chapitre`, `\today` отпечатва текущата дата на френски и така нататък. Става достъпен също и набор от нови команди, позволяващи леко да се набират френски текстове. За пояснение, вижте Таблица 2.3.

Когато превключите на френски език, ще забележите също и изменение във вида на списъците. За повече информация за това, какво прави `frenchb` опцията на `babel` и как можете да измените нейното поведение,

¹Езиците, поддръжани от всяка от тези кодировки, са изброени в [12].

Таблица 2.3: Специални команди за френски език

<code>\og guillemets \fg{}</code>	guillemets
<code>M\up{me}, D\up{r}</code>	M ^{me} , D ^r
<code>1\ier{ }, 1\iere{ }, 1\ieres{ }</code>	1 ^{er} , 1 ^{re} , 1 ^{res}
<code>2\ieme{ } 4\iemes{ }</code>	2 ^e 4 ^{es}
<code>\No 1, \no 2</code>	N ^o 1, n ^o 2
<code>20~\degres C, 45\degres</code>	20 °C, 45°
<code>\bsc{M. Durand}</code>	M. DURAND
<code>\nombre{1234,56789}</code>	1,234.567,89

компилирайте с L^AT_EX файла `frenchb.dtx` и прочетете получения файл `frenchb.dvi`.

2.5.2 Поддръжка на немски език

Ако искате да създавате с L^AT_EX немски документи, можете да заредите поддръжката на немски език с командата

```
\usepackage[german]{babel}
```

Това включва немските правила за пренасяне, ако вашата система L^AT_EX е правилно настроена. Освен това, автоматично генерираните стрингове ще станат на немски, например, “Chapter” ще стане “Kapitel”. Става достъпен и набор от нови команди, които ви дават възможност да пишете на немски по-бързо, дори когато не използвате пакета `inputenc`. За пояснение, вижте Таблица 2.4. С използване на `inputenc` всичко това става ненужно, но вашият текст ще може да се превключва на различни кодировки.

В немските книги често се срещат френските знаци кавички («guillemets»). Немските типографи обаче, ги използват по друг начин. Цитат в немска книга обикновено изглежда »така«. В немско-говорящите части на Швейцария, типографите използват «guillemets» така, както и във Франция.

Главният проблем е свързан с команди от типа `\flq`. Ако използвате шрифтовете OT1 (а те се включват по подразбиране), тези кавички изглеждат като математическия символ “<<”, от което на типографите им прилошава. Шрифтовете в кодировка T1, от друга страна, съдържат

Таблица 2.4: Специални символи за немски език

"a	ä	"s	ß
"‘	”	"’	“
"< или \flqq	«	"> или \frqq	»
\flq	<	\frq	>
\dq	"		

необходимите символи. Така че, ако са ви необходими такива кавички, убедете се, че сте включили кодировка T1 (`\usepackage[T1]{fontenc}`).

2.5.3 Поддръжка на Кирилица

Пакетът `babel`, започвайки от версия 3.7, включва поддръжка на кодировка T2* и поддръжка на обработката на български, руски и украински текстове с използване на кирилически букви.

Поддръжката на Кирилицата е основана на стандартния механизъм на L^AT_EX: пакетите `fontenc` и `inputenc`. Но, ако трябва да използвате Кирилица в математически режим, трябва да заредите пакета `mathtext` преди пакета `fontenc`:¹

```
\usepackage{mathtext}
\usepackage[T1,T2A]{fontenc}
\usepackage[cp1251]{inputenc}
\usepackage[english,russian,bulgarian]{babel}
```

Казано въобще, `babel` автоматично избира кодировката на шрифта по подразбиране. За трите изброени езици това е T2A. Обаче, документите не са ограничени само с една кодировка на шрифта. За многоезични документи, използващи езици, основани на Кирилица и Латиница, има смисъл явно да се включи латинската кодировка на шрифтовете. `babel` се грижи за превключването на съответната кодировка, когато се избират различни езици в документа.

В допълнение към включването на правилата за пренасяне, превеждането на автоматично генерираните стрингове и активирането на някои специфични за езика типографски правила (например, `\frenchspacing`), `babel` предоставя няколко команди за обработка в съответствие със стандартите на български, руски или украински езици.

¹ Ако използвате пакетите *A_MS-L_AT_EX*, заредете ги също преди пакетите `fontenc` и `babel`.

И за трите езика се предоставя специфична пунктуация: кирилическо тире за текста (то е малко по-дълго от латинското тире и е оградено с малки интервали), тире за пряка реч, кавички и команди за управление на пренасянето, вж. Таблица 2.5.

Таблица 2.5: Допълнителни дефиниции, въвеждани с опциите на `babel` за български, руски и украински езици

"	забранява лигатура в тази позиция.
"-	дефис, разрешаващ да се пренася остатъкът от думата.
"---	кирилическо тире в обикновен текст.
"--~	кирилическо тире в съставни фамилии.
"--*	кирилическо тире за означаване на пряка реч.
" "	като "-", но не генерира знака дефис (за съставни думи с дефис, например <code>x-"u</code> или друг знак като "disable/enable").
"~	означава съставна дума без разделяне.
"="	означава съставна дума с разделяне, с разрешаване пренасянето в думите-компоненти.
" ,	малък интервал за инициали с разрешаване на пренасянето в следващата фамилия.
"“	немски леви двойни кавички (изглежда така: „).
"’	немски десни двойни кавички (изглежда така: “).
"<	френски леви двойни кавички (изглежда така: <<).
">	френски десни двойни кавички (изглежда така: >>).

Руската и украинската опции на пакета `babel` въвеждат команди `\Asbuk` и `\asbuk`, които работят аналогично на командите `\Alpha` и `\alpha`, но генерират малки и главни букви от руската или украинската азбуки (в зависимост от текущия език в документа). Българската опция на `babel` въвежда команди `\enumBul` и `\enumLat` (`\enumEng`), които заставят `\Alpha` и `\alpha` да генерират букви или от българската, или от латинската (английската) азбуки. По подразбиране се генерират български букви.

2.6 Интервали между думите

За получаване на равен десен край на извеждане, \LaTeX поставя различни интервали между думите. В края на изречението той слага малко по-голям интервал, което прави текста по-удобен за четене, в съответствие с традициите, приети в английския език). \LaTeX предполага, че изречението завършва с точка, въпросителен или възклицателен знак. Ако точката следва след буква в горен регистър, тя не се счита за край на изречението, тъй като точки след букви в горен регистър обикновено се използват за съкращение.

Всяко изключение от тези предположения трябва да бъде явно определено от автора. Знакът «\» пред интервал генерира интервал, който не може да бъде увеличен. Знакът тилда «~» генерира интервал, който не може да бъде увеличен и който, освен това, забранява прекъсването на реда. Командата \@ пред точка определя, че тази точка е край на изречение, дори ако тя стои след буква в горен регистър.

```
Mr.~Smith е трудолюбив човек.\@
вж.~Fig.~5\@
Аз харесвам MATLAB\@. А ти?
```

```
Mr. Smith е трудолюбив човек.
вж. Fig. 5
Аз харесвам MATLAB. А ти?
```

Допълнителен интервал след точка може да бъде забранен с командата

```
\frenchspacing
```

която указва на L^AT_EX да *не* въвежда по-голям интервал след точка, отколкото интервал след обикновен символ. Това е често срещано в езици, различни от английския, освен в библиографията. Ако използвате \frenchspacing, командата \@ не е необходима.

2.7 Заглавия, глави и раздели

За да помогнете на читателя да се ориентира във вашия документ, вие трябва да го разделите на глави, раздели и под-раздели. L^AT_EX поддържа това със специални команди, които имат за аргумент заглавието на раздела. Вашата задача е да ги използвате в точна последователност.

В класа `article` са достъпни следните команди за секциониране:

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

Ако искате да разбийте вашия документ на части, без изменение на номерацията на разделите/главите, използвайте командата

```
\part{...}
```

Когато работите в класовете `report` или `book`, можете да използвате

допълнителна команда за секционирание

```
\chapter{...}
```

Тъй като класът `article` не знае нищо за главите (chapters), статиите доста лесно се добавят в книга като глави. Интервалите между разделите, номерацията и размера на шрифта на заглавията се установяват от `LaTeX` автоматично.

Две от командите за секционирание са особени:

- Командата `\part` не влияе на последователността на номериране на главите.
- Командата `\appendix` няма аргумент. Тя просто започва да номерира главите с букви вместо цифри.¹

`LaTeX` създава Съдържание, като взема заглавията на разделите и номерата на страниците от предходния цикъл на компилация на документа. Командата

```
\tableofcontents
```

извежда Съдържанието на това място, където тя се е появила. Нов документ трябва да бъде компилиран с `LaTeX` два пъти, за да се получи правилно Съдържание. Понякога може да се наложи да се компилира документа и трети път. `LaTeX` ще ви каже, когато това е необходимо.

Всички изброени по-горе команди за секционирание съществуват също и във варианти със звездичка. Вариант със звездичка се получава чрез добавяне на `*` към името на командата. Така се генерират заглавия на раздели, които не се номерират и не се появяват в Съдържанието.

Например, ако се използва командата

```
\chapter*{Увод},
```

заглавието на тази глава в текста се отпечатва без номер и не се появява в съдържанието.

Ако все пак искаме да се появи заглавието на тази глава в съдържанието, се използва командата

```
\addcontentsline{toc}{chapter}{Увод},
```

където `toc` е разширението на файл със същото име като вашия документ, в който `LaTeX` записва заглавията на главите и разделите, които след това ще се отпечатват в Съдържанието.

¹В класа `article` се изменя номерацията на разделите.

Обикновено заглавията на разделите се появяват в съдържанието точно в същия вид, в който те са въведени в текста. Понякога това е невъзможно поради това, че заглавието е твърде дълго, за да се постави в съдържанието. В този случай, елементът на съдържанието може бъде определен като незадължителен аргумент пред самото заглавие.

```
\chapter[Заглавие за съдържанието]{Това е твърде дълго
и отегчително заглавие, появяващо се в текста}
```

Командите

```
\tableofcontents
\listoffigures
\listoftables
```

генерират съответно Съдържание (toc), Списък на Фигурите (lof) и Списък на Таблиците (lot).

Заглавната страница на целия документ се генерира с командата

```
\maketitle
```

Нейното съдържание трябва да бъде определено с командите

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

преди извикване на `\maketitle`. Аргументът на командата `\author` може да съдържа няколко имена, разделени с командата `\and`.

Пример на някои от посочените команди може да бъде намерен на [Фигура 1.2](#) на [страница 8](#).

Освен описаните по-горе команди за секционирание, L^AT_EX 2_ε въвежда три допълнителни команди за използване с класа `book`. Те са полезни при разделяне на книгата на части според типа на информацията. Тези команди изменят заглавията на главите и номерацията на страниците така, както това се очаква от книга:

`\frontmatter` трябва да бъде първата команда след `\begin{document}`.

Тя превключва номерацията на страниците с използване на римски цифри, а разделите не са номерирани. Това е подобно на използването на команди за секционирание със звездички (например, `\chapter*{Увод}`), но основната разлика е, че заглавията на разделите ще се появяват в Съдържанието.

`\mainmatter` се поставя непосредствено преди първата глава на книгата.

Тя включва номерацията на страниците с арабски цифри и изчислява брояча на страниците.

`\appendix` отбелязва началото на допълнителния материал във вашата книга. След тази команда главите ще се номерират с букви.

`\backmatter` трябва да бъде въведена преди последните части на вашата книга, такива като Библиография и Азбучен Указател. В стандартните класове документи, няма видим ефект от това.

2.8 Препратки

В книги, доклади и статии често се срещат препратки към формули, фигури, таблици и специални части на текста. \LaTeX предоставя следните команди за препратки:

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

където *етикет* е идентификатор, избран от потребителя.

\LaTeX заменя `\ref` с номера на раздела, под-раздела, фигурата, таблицата или теоремата, след която е била използвана съответната команда `\label`. `\pageref` печата номера на тази страница, на която се е появила съответната команда `\label`. Така, както и в случая със заглавията на раздели, тук също се използват номерата на етикетите, «запомнени» при предишната компилация.

Препратката на този
раздел `\label{sec:this}`
изглежда така: <<вж.
Раздел `\ref{sec:this}` на
стр. `\pageref{sec:this}`.>>

Препратката на този раздел изглежда така:
«вж. Раздел 2.8 на стр. 38.»

2.9 Забележки под печатното поле

С командата

```
\footnote{Текст на забележката}
```

се печата забележка под линия на текущата страница.

Забележката под печатното поле трябва винаги да се поставя след думата или изречението, към което се отнася. Следователно, забележки, които се отнасят към изречение, или част от него, трябва да се поставят след точката или запетаята.¹

¹Да отбележим, че забележките отвличат читателя от основния текст на документа. Затова, най-добре е да се постараете да включите всичко, което искате да кажете, в тялото на документа.

Потребителите на `\LaTeX{}` често употребяват забележки под поле. `\footnote{%
Това е забележка.}`

Потребителите на `LaTeX` често употребяват забележки под поле.^a

^aТова е забележка.

2.10 Подчертани думи

В ръкописи, напечатани на пишещата машина, важните думи се отделят с подчертаване:

```
\underline{текст}
```

В печатните издания обаче, тези думи се подчертават като се отпечатват в наклонен шрифт (*курсив*). `LaTeX` предоставя командата

```
\emph{текст}
```

за *подчертаване* на текст. Нейният аргумент е текста за подчертаване. Какво всъщност прави тази команда със своя аргумент, зависи от контекста:

Ако използвате `\emph{подчертаване вътре във вече подчертан текст}`, то `LaTeX{}` използва `\emph{прав}` шрифт.}

Ако използвате *подчертаване вътре във вече подчертан текст*, то `LaTeX` използва *прав шрифт*.

Забележете разликата между командите за *подчертаване* и смяна на *шрифта*:

```
\textit{Можете също така да  
  \emph{подчертаете} текст,  
  ако го наберете в italic,}  
\textsf{ в sans-serif}  
  \emph{шрифт}  
\texttt{или в  
  \emph{typewriter} стил.}
```

Можете също така да подчертаете текст, ако го наберете в italic, в sans-serif шрифт или в typewriter стил.

2.11 Среди

За обработка на специални видове текст, \LaTeX дефинира множество среди за различни видове форматиране:

```
\begin{среда} текст \end{среда}
```

където *среда* определя името на средата. Среда може да се извиква от друга среда, като се съблюдава реда на отваряне и затваряне на средата:

```
\begin{a}... \begin{b}... \end{b}... \end{a}
```

В следващите под-раздели се разказва за някои важни среди.

2.11.1 Списък, изброяване и описание

Средата `itemize` е подходяща за прости списъци, средата `enumerate` – за номерирани списъци, а средата `description` – за описание.

```
\flushleft
\begin{enumerate}
\item Можете по ваш вкус да
смесвате средите на списъците:
\begin{itemize}
\item Но това може да изглежда
глупаво.
\item[-] с тире.
\end{itemize}
\item Затова запомнете:
\begin{description}
\item[Глупавите] неща няма да
станат по-значителни от това,
че са в списък.
\item[Умните] неща обаче,
могат да бъдат представени
по-добре в списък.
\end{description}
\end{enumerate}
```

1. Можете по ваш вкус да смесвате средите на списъците:

- Но това може да изглежда глупаво.
- с тире.

2. Затова запомнете:

Глупавите неща няма да станат по-значителни от това, че са в списък.

Умните неща обаче, могат да бъдат представени по-добре в списък.

2.11.2 Подравняване вляво, вдясно и центриране

Средите `flushleft` и `flushright` генерират параграфи, които са подравнени вляво или вдясно. Средата `center` генерира центриран текст. Ако не използвате `\` за определяне на местата за прекъсване на редовете, \LaTeX ги определя автоматично.

```
\begin{flushleft}
  Този текст\\ е подравнен вляво.
  \LaTeX{} не се стареа да направи
  всички редове с еднаква дължина.
\end{flushleft}
```

Този текст
е подравнен вляво. \LaTeX не се стареа да
направи всички редове с еднаква дължина.

```
\begin{flushright}
Този текст\\ е подравнен вдясно.
\LaTeX{} не се стареа да направи
всички редове с еднаква дължина.
\end{flushright}
```

Този текст
е подравнен вдясно. \LaTeX не се стареа да
направи всички редове с еднаква дължина.

```
\begin{center}
  В центъра\\на страницата.
\end{center}
```

В центъра
на страницата.

2.11.3 Цитати и стихове

Средата `quote` е полезна за цитати, важни фрази и примери.

```
Типографско правило за
дължина на реда:
\begin{quote}
  По правило, редът трябва
  да съдържа не повече от
  66~символа.
\end{quote}
Затова \LaTeX{} прави такива
широки полета на страницата.
Затова във вестниците често се
използва набор в няколко колони.
```

Типографско правило за дължина на реда:

По правило, редът трябва да съ-
държа не повече от 66 символа.

Затова \LaTeX прави такива широки полета
на страницата. Затова във вестниците често
се използва набор в няколко колони.

Съществуват още две подобни среди: `quotation` и `verse`. Средата `quotation` е полезна за по-дълги цитати, състоящи се от няколко параграфа, защото тя прави отстъп на първия ред от всеки параграф. Средата `verse` се използва за стихотворения, при които са важни прекъсванията на редовете. Редовете се отделят чрез въвеждане на `\\` в края на реда и празен ред след всяка строфа.

Помня една детска песничка.

Тя е за Зайченцето бяло:

```
\begin{flushleft}
\begin{verse}
Зайченцето бяло\\
цял ден си играло\\
в близката горичка\\
със една сърничка.
\end{verse}
\end{flushleft}
```

Помня една детска песничка. Тя е за Зайченцето бяло:

```
Зайченцето бяло
цял ден си играло
в близката горичка
със една сърничка.
```

2.11.4 Резюме

В научните публикации е прието да се започва с резюме, което дава на читателя бърз преглед на това, какво да очаква. \LaTeX осигурява за тази цел средата `abstract`. Обикновено `abstract` се използва в документи, които се обработват в класа `article`.

```
\begin{abstract}
Това е резюме.
\end{abstract}
```

Това е резюме.

2.11.5 Буквално възпроизвеждане

Текст, заграден с `\begin{verbatim}` и `\end{verbatim}` ще бъде директно напечатан, като набран на пишеща машина, с всички интервали и нови редове, без изпълняване на каквито и да било команди на \LaTeX .

Вътре в параграфа, аналогична функция изпълнява командата

```
\verb+текст+
```

Тук «+» е само пример на символ-ограничител. Вие можете да използвате всеки символ, освен букви, «*» или интервал. Много \LaTeX -примери в тази книжка са набрани с тази команда.

Командата `\verb|\ldots| \ldots`

```
\begin{verbatim}
1; function y=f(x)
2; y=sin(x)+exp(-x)+x.^3-5;
\end{verbatim}
```

Командата `\ldots ...`

```
1; function y=f(x)
2; y=sin(x)+exp(-x)+x.^3-5;
```

```
\begin{verbatim*}
Вариантът на средата
verbatim      със
звездичка подчертава
интервалите в текста.
\end{verbatim*}
```

```
Вариантът на средата
verbatim      със
звездичка подчертава
интервалите в текста.
```

Командата `\verb` също може да се използва по аналогичен начин със звездичка:

```
\verb*|ето така :-) |
```

```
ето така :-)
```

Средата `verbatim` и командата `\verb` не могат да се използват вътре в параметри на други команди.

2.11.6 Таблици

Средата `tabular` може да се използва за отпечатване на красиви таблици, със или без хоризонтални и вертикални линии. \LaTeX определя ширината на колоните автоматично.

Аргументът *спецификация* на командата

```
\begin{tabular}[позиция]{спецификация}
```

дефинира формата на таблицата. Използвайте `l` за колона с ляво-подравнен текст, `r` – за дясно-подравнен текст и `c` – за центриран текст, `p{ширина}` – за колона, съдържаща свързан текст с пренасяне на думите, и `|` – за вертикална линия.

Ако текстът в колоната е твърде широк за страницата, \LaTeX няма да може автоматично да го пренесе. Използвайки `p{ширина}`, вие можете да дефинирате специален тип колона, в която текстът ще се пренася като в нормален параграф.

Аргументът *позиция* определя вертикалното положение на таблицата относно базовата линия на околния текст. Използвайте една от буквите `t`, `b` и `c`, за да позиционирате таблицата горе, долу или в центъра спрямо основната линия.

Вътре в средата `tabular` знакът «&» прехвърля към следващата колона, командата `\\` започва нов ред, а `\hline` въвежда хоризонтална линия. Можете да добавяте непълни линии с помощта на командата `\cline{j-i}`, където `j` и `i` са номерата на колоните, над които трябва да се простира линията.

```
\begin{tabular}{|r|l|}
\hline
7C0 & шестнадесетично \\
3700 & осмично \\
11111000000 & двоично \\
\hline
1984 & десетично \\
\hline
\end{tabular}
```

7C0	шестнадесетично
3700	осмично
11111000000	двоично
1984	десетично

```
\begin{tabular}{|p{4.7cm}|}
\hline
Този параграф е в рамка. \\
\hline
\end{tabular}
```

Този параграф е в рамка.

Разделителят на колоните може да се задава с конструкцията $\@{...}$. Тази команда премахва интервала между колоните и го заменя с това, което е включено между фигурните скоби. Едно от честите приложения на тази команда е показано по-долу, в задачата за подравняване по десетичната точка. Друго възможно приложение е премахването на водещия интервал в таблица с $\@{}$:

```
\begin{tabular}{@{} l @{} }
\hline
няма водещ интервал \\
\hline
\end{tabular}
```

няма водещ интервал

```
\begin{tabular}{l}
\hline
водещ интервал вляво и вдясно \\
\hline
\end{tabular}
```

водещ интервал вляво и вдясно

Тъй като няма вграден способ за подравняване на числови колони по десетичната точка, ние можем да «надхитрим» \TeX и да направим това с помощта на две колони: дясно-подравнена цяла част и ляво-подравнена дробна. Командата $\@{.}$ в реда $\begin{tabular}$ заменя нормалния интервал между колоните просто с «.», давайки ефект на една колона, подравнена по десетичната точка. Не забравяйте да замените във вашите числа десетичната точка с разделител между колони ($\&$)! Надпис над колона може да се постави над нашата числова «колона» с командата \multicolumn :


```

\begin{tabular}{c r @{.} l}
Израз с  $\pi$  & & 
\multicolumn{2}{c}{Стойност} \\
\hline
 $\pi$  & 3.1416 & \\
 $\pi^{\pi}$  & 36.46 & \\
 $(\pi^{\pi})^{\pi}$  & 80662.7 & 
\end{tabular}

```

Израз с π	Стойност
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

```

\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Материали} \\
\hline
Вид 1 & Вид 2 \\
\hline
\end{tabular}

```

Материали	
Вид 1	Вид 2

Материал, набран в средата `tabular`, винаги е неделим и се разполага на една страница. Ако искате да набирате дълги таблици, може да видите пакетите `supertabular` или `longtable`.

2.12 Плаващи обекти

В наши дни повечето публикации съдържат много фигури и таблици. Тези елементи на текста се нуждаят от специално отношение, защото те не могат да бъдат разбити между страниците. Един изход би бил да се започва нова страница всеки път, когато се срещне фигура или таблица, твърде голяма, за да се събере на текущата страница. Този подход би довел до това, страниците да остават частично празни, което изглежда много зле.

Решението на този проблем е: всяка фигура или таблица, която не се събира на текущата страница да „плава“ в документа, премествайки се на следващата страница, а в същото време текущата страница да се запълва с текст. \LaTeX позволява на фигурите и таблиците да се преместват («плават») на различни позиции на страницата, минимизирайки празното пространство и максимизирайки количеството текст на страницата.

\LaTeX предлага две среди за плаващите обекти, една за таблици и друга – за фигури. За да се възползвате напълно от тези две среди, е важно да разберете приблизително, как \LaTeX обработва плаващите обекти. Иначе те могат да станат източник на разочарование поради това, че \LaTeX ги поставя не там, където вие искате.

Концепцията на «плаващите» фигури е особено трудна за WYSIWYG (what you see is what you get) - потребителите, които използ-

ват точно фиксирани позиции на графиките и фигурите при създаването на документа. Важно е да се запомнят две важни неща:

- Никога не свързвайте думите в текста с позицията на фигурата. Използването на фразите „Тази фигура. . .“ или „Следващата фигура“ изискват фигурата да следва непосредствено.

Използването на фразата „Фигура 5. . .“ позволява фигурата да бъде разположена къде да е на страницата (дори и на друга страница).

- Отпуснете се! Някои автори много се притесняват, когато фигурата не се появява точно където искат. Позицията на фигурите е **работа** на L^AT_EX, не се грижете за това.

Нека да разгледаме командите, предлагани от L^AT_EX за плаващите обекти:

Всеки материал, включен в средата `figure` или `table`, се определя като «плаващ» обект. И двете среди имат незадължителен параметър

```
\begin{figure}[спецификация на позицията] или
\begin{table}[спецификация на позицията]
```

наречен *спецификация на позицията*. Този параметър се използва за указване на L^AT_EX, къде може да се премества плаващия обект. *Спецификация на позицията* се конструира като поредица от *ключове за разрешени позиции на плаващия обект*, (вж. Таблица 2.6).

Например, дадена таблица може да започва със следния ред:

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

Таблица 2.6: Ключове за разрешени позиции на плаващ обект

Ключ	Разрешава да се разположи обекта . . .
<code>h</code>	<i>тук</i> , на същото място в текста, където се е появил плаващият обект. Обикновено се използва за малки обекти.
<code>t</code>	<i>горе</i> на страницата
<code>b</code>	<i>долу</i> на страницата
<code>p</code>	на <i>специална страница</i> , съдържаща само плаващи обекти.
<code>!</code>	без да се разглеждат повечето вътрешни параметри, ^a които могат да попречат за разполагането на този обект.

^aТакива, като максимален брой плаващи обекти, разрешени на една страница

Спецификацията на позицията `[!hbp]` позволява на \LaTeX да постави таблицата точно на това място (`h`) или долу на същата страница (`b`), или на отделна страница (`p`) с плаващи обекти, дори всичко това да не изглежда много добре (!). Ако не е зададена никаква спецификация на позицията, стандартните класове предполагат `[tbp]`.

\LaTeX разполага всеки срещнат плаващ обект в съответствие със зададената от автора спецификация. Ако обектът не може да се събере на текущата страница, той се отстранява и се поставя на опашка от фигури или на опашка от таблици.¹ Когато започне нова страница, \LaTeX проверява, може ли да запълни специална страница с плаващи обекти от опашките. Ако това не е възможно, първият обект от всяка опашка се разглежда като току що появил се в текста; \LaTeX отново се опитва да ги постави в съответствие с тяхната спецификация (с изключение на „h“, което вече е невъзможно). Всеки нов плаващ обект, който се появи в текста, се поставя на съответната опашка. \LaTeX стриктно спазва реда, в който са били срещнати плаващи обекти от съответния тип. Затова фигура, която не може да се позиционира, отхвърля всички следващи фигури към края на документа. Следователно:

Ако \LaTeX не позиционира плаващите обекти както вие очаквате, то често това се дължи само на един обект, който е «задръстил» едната или другата опашка.

Макар и да е възможно да се задават в \LaTeX единични спецификации на позицията на плаващ обект, това може да доведе до проблеми. Ако обектът не се събира на посоченото място, той остава на опашката, като блокира следващите плаващи обекти. В частност, никога не трябва да използвате опцията `[h]`; това е толкова лошо, че в съвременните версии на \LaTeX той автоматично се заменя на `[ht]`.

След обясненията на тези трудности остава да отбележим още няколко неща за средите `table` и `figure`. Чрез командата

```
\caption{текст на заглавието}
```

можете да задавате заглавие на плаващия обект. Нарастващият номер и надписът «Фигура» или «Таблица», се добавят от \LaTeX .

Двете команди

```
\listoffigures и \listoftables
```

работят аналогично на командата `\tableofcontents`, печатайки съответно списък на фигурите или таблиците. Тези списъци включват целия

¹Тези опашки се подчиняват на дисциплината *FIFO*: ‘първ влязъл – първ излязъл’.

аргумент на командата `\caption`, така че, ако сте склонен да използвате дълги заглавия, трябва да предоставите техен кратък вариант за включване в списъците. Това се осъществява чрез въвеждане на кратка версия в квадратни скоби след командата `\caption`.

```
\caption[Кратко]{Ддддъъъъъъъъъъъълго}
```

С помощта на `\label` и `\ref` могат да се правят препратки от вашия текст към плаващия обект.

Следният пример рисува квадрат и го поставя в документа. Тази техника може да се използва, за да се остави в документа място за изображение, което ще се постави по-късно.

```
1 На Фигура~\ref{white} е даден
2 квадрат със страна $5\, cm$.
3 \begin{figure}[!hbp]
4 \makebox[\textwidth]{%
5 \framebox[5cm]{\rule{0pt}{5cm}}}
6 \caption{Пет на пет в сантиметри}
7 \label{white}
8 \end{figure}
```

В горния пример \LaTeX *много силно* (!) ще се старее да позиционира фигурата точно *на място* (h).¹ Ако това е невъзможно, той ще се опита да я разположи *долу на страницата* (b). Ако не му се отдаде да постави фигурата на текущата страница, той ще провери, може ли да създаде страница с плаващи обекти, съдържаща тази фигура и, вероятно, някои таблици от опашката с таблици. Ако няма достатъчно материал за специална страница с плаващи обекти, \LaTeX започва нова страница и отново разглежда фигурата, все едно че тя току що се е появила в текста.

В определени случаи може да се наложи да се използва команда

```
\clearpage или дори \cleardoublepage
```

Командата `\clearpage` заповядва на \LaTeX , незабавно да постави всички плаващи обекти, останали в опашките, след което да започне нова страница. `\cleardoublepage`, освен това, започва нова дясно-стояща страница.

¹Предполага се, че опашката от фигури е празна.

Глава 3

Набор на математически формули

В тази глава ще се срещнем с най-голямата сила на \TeX : математическата обработка. Но имайте предвид, че тази глава не навлиза дълбоко в същността на темата. Макар, за много хора, изложените тук неща да са достатъчно, не се отчайвайте, ако не успеете да намерите решение, съответстващо на вашите нужди за обработка на математически формули. Твърде вероятно е, вашият проблем да се решава в $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ ¹.

3.1 Общи сведения

$\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ има специален режим за обработка на математика. Математически формули могат да бъдат набрани вътре в параграфа, или параграфът може да бъде разбит на части, за да се отпечатаат те отделно. Математически текст вътре в параграф се въвежда между $\backslash($ и $\backslash)$, $,$ между $\$$ и $\$$ или между $\backslash\text{begin}\{\text{math}\}$ и $\backslash\text{end}\{\text{math}\}$.

Събирайки $\$a\$$ на квадрат с $\$b\$$ на квадрат, получаваме $\$c\$$ на квадрат. Или на езика на математиката: $\$a^{\{2\}}+b^{\{2\}}=c^{\{2\}}\$$

Събирайки a на квадрат с b на квадрат, получаваме c на квадрат. Или на езика на математиката: $a^2 + b^2 = c^2$

¹ *Американското математическо общество* е изработило мощно разширение на $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. Много примери в тази глава използват това разширение. То е включено във всички съвременни дистрибутиви на \TeX . Ако във вашето то отсъства, можете да го получите от адрес CTAN:/tex-archive/macros/latex/required/amslatex.

```
\TeX{} се произнася като
\(\tau\epsilon\chi\).\[6pt]
100~\m$^{3}$ вода.\[6pt]
Това излиза от моето
\begin{math}\heartsuit\end{math}
```

```
TeX се произнася като  $\tau\epsilon\chi$ .
100 м3 вода.
Това излиза от моето ♡
```

Когато искате да «отделите» големи математически уравнения или формули от останалата част от параграфа, е за предпочитане те да се поставят на отделни редове, без да се прекъсва параграфа. За да направите това, можете да ги оградите с `\[` и `\]` или с `\begin{displaymath}` и `\end{displaymath}`.

```
Събирайки  $a$  на квадрат с
 $b$  на квадрат, получаваме
 $c$  на квадрат. Или
на езика на математиката:
\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}
или можете да изразите това
по-кратко:  $[a^2+b^2=c^2]$ 
```

```
Събирайки  $a$  на квадрат с  $b$  на квадрат, по-
лучаваме  $c$  на квадрат. Или на езика на ма-
тематиката:
```

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

```
или можете да изразите това по-кратко:
```

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

Ако искате \LaTeX да номерира вашите уравнения, можете да използвате средата `equation`. При това можете да поставите на уравнението етикет `\label` и да правите препратки към него от произволно място в текста с командите `\ref` или `\eqref`:

```
\begin{equation}
\label{eq:eps}
\varepsilon > 0
\end{equation}
От (\ref{eq:eps})
следва \ldots{}
От \eqref{eq:eps}
също следва \ldots{}

```

$$\varepsilon > 0 \tag{3.1}$$

```
От (3.1) следва ... От (3.1) също следва ...
```

Забележете разликата в стила на отпечатване на уравнения вътре в параграф и когато те са «отделени»:

```
 $\lim_{k \to \infty}
\sum_{n=1}^k \frac{1}{n^3}$ 
```

$$\lim_{k \rightarrow \infty} \sum_{n=1}^k \frac{1}{n^3}$$

```
\begin{displaymath}
\lim_{k \to \infty}
\sum_{n=1}^k \frac{1}{n^3}
\end{displaymath}
```

$$\lim_{k \rightarrow \infty} \sum_{n=1}^k \frac{1}{n^3}$$

Има разлики между *математически режим* и *текстов режим*. Например, в *математически режим*:

1. Повечето интервали и прекъсвания на редове не се вземат под внимание, тъй като всички интервали или произлизат от логиката на математическите изрази, или трябва да се задават в явен вид с команди като `\`, или `\quad`, или `\qquad`.
2. Празни редове не се допускат. Всяка формула заема само един параграф.
3. Всяка буква се разглежда като име на променлива и се отпечатва като такава. Ако искате да въведете нормален текст вътре във формула (нормален прав шрифт с нормални интервали), то е необходимо да го въведете с командите `\textrm{...}` (вж. също Раздел 3.7 на стр. 59).

```
\begin{equation}
  \forall x \in \mathbf{R}:
  \quad x^2 \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R} : \quad x^2 \geq 0 \quad (3.2)$$

```
\begin{equation}
  x^2 \geq 0 \qquad
  \textrm{за всяко } x \in
  \mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \quad \text{за всяко } x \in \mathbf{R} \quad (3.3)$$

Математиците са много строги по отношение на използваните символи: прието е да се използват “ажурни дебели символи“, които се получават с командата `\mathbb` от пакетите `amsmath` или `amssymb`. Последният пример сега изглежда така:

```
\begin{displaymath}
  x^2 \geq 0 \qquad
  \textrm{за всяко } x \in
  \mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \quad \text{за всяко } x \in \mathbb{R}$$

3.2 Групиране в математически режим

Повечето команди в математически режим действат само на следващия символ. Така че, ако искате командата да влияе на няколко символа, трябва да ги групирате заедно с помощта на фигурни скоби: `{...}`.

```
\begin{equation}
  a^x+y \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \quad (3.4)$$

3.3 Елементи на математическите формули

В този раздел ще бъдат описани най-важните команди, използвани в математическата обработка. Подробен списък на командите за набор на математически символи вижте в Раздел 3.11 на страница 66.

Малки гръцки букви се въвеждат с командите `\alpha`, `\beta`, `\gamma`, ..., главните букви се въвеждат с командите `\Gamma`, `\Delta`, ...¹

```

 $\lambda, \xi, \pi, \mu, \Phi, \Omega$ 
 $\Phi, \Omega$ 

```

$$\lambda, \xi, \pi, \mu, \Phi, \Omega$$

Горни и долни индекси се въвеждат с помощта на символите «`^`» и «`_`».

```

 $a_1$   $\quad$   $x^2$   $\quad$   $e^{-at}$   $\quad$   $a_{ij}^3$ 
 $e^{x^2} \neq e^{x^2}$ 

```

$$a_1 \quad x^2 \quad e^{-at} \quad a_{ij}^3$$

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

Квадратен корен се въвежда с командата `\sqrt`, корен n -ти се генерира с `\sqrt[n]`. Размерът на знака за корен се избира от L^AT_EX автоматично. Ако ви е нужен само знак за корен, използвайте `\surd`.

```

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

```

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

$$\sqrt[3]{2} \quad \sqrt[3]{2}$$

$$\sqrt[3]{2} \quad \sqrt[3]{2}$$

Командите `\overline` и `\underline` създават **хоризонтални линии** непосредствено над или под израза.

```
 $\overline{m+n}$ 
```

$$\overline{m+n}$$

Командите `\overbrace` и `\underbrace` създават дълги **хоризонтални фигурни скоби** непосредствено над или под израза.

```

 $\underbrace{a+b+\cdots+z}_{26}$ 

```

$$\underbrace{a+b+\cdots+z}_{26}$$

¹В L^AT_EX 2_ε не се дефинира главна буква «алфа», защото тя изглежда по същия начин, както и латинската «А».

За добавяне към променливите на знаци за математически акценти, такива, като малки стрелки или знак тилда, можете да използвате командите, изброени в Таблица 3.1 на страница 66. Широки «шапки» и тилди, обхващащи няколко символи, се генерират с командите `\widetilde` и `\widehat`. Символът «'» дава знак за производна.

```
\begin{displaymath}
y=x^2\quad y'=2x \quad \quad y''=2
\end{displaymath}
```

$$y = x^2 \quad y' = 2x \quad y'' = 2$$

Векторите се означават чрез добавяне на малък знак стрелка над променливата. Това се прави с командата `\vec`. За означаване на вектор от A до B са полезни двете команди `\overrightarrow` и `\overleftarrow`.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

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

Обикновено знакът точка, означаващ операция умножение, не се намира в явен вид. Обаче, понякога той се записва, за да помогне на читателя да групира формулата. В тези случаи се използва командата `\cdot`:

```
\begin{displaymath}
\xi = {\lambda}_1 \cdot {\lambda}_2
\varphi_1 \cdot \varphi_2
\end{displaymath}
```

$$\xi = \lambda_1 \cdot \lambda_2 \varphi_1 \cdot \varphi_2$$

Имената на функции от типа lg често се записват в прав шрифт, а не в наклонен като променливите. Затова L^AT_EX предлага следните команди за набор на имена на най-важните функции:

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

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

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

За функцията модул има две команди: `\bmod` за бинарен оператор « $a \bmod b$ » и `\pmod` за изрази от вида « $x \equiv a \pmod{b}$ ».

```
$a\bmod b$\
$x\equiv a \pmod{b}$
```

$$a \bmod b \\ x \equiv a \pmod{b}$$

Обикновена **дроб** се записва с командата `\frac{...}{...}`. Често се предпочита форма с кóса черта $1/2$, защото тя се вижда по-добре при малко количество „дробен материал“.

```
$1\frac{1}{2}$~часа
\begin{displaymath}
\frac{x^2}{k+1} \quad \quad
x^{\frac{2}{k+1}} \quad \quad
x^{1/2}
\end{displaymath}
```

$$1\frac{1}{2} \text{ часа}$$

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

За отпечатване на биномни коефициенти или подобни структури може да се използва командата `\binom` от пакета `amsmath`.

```
\begin{displaymath}
\binom{n}{k} \quad \quad \mathrm{C}_n^k
\end{displaymath}
```

$$\binom{n}{k} \quad C_n^k$$

За бинарни отношения е удобно да се разполагат символи един над друг. Командата `\stackrel` намира символа, зададен като първи аргумент, в шрифт с размер на индекс, и го разполага над втория аргумент, отпечатван в обичайна позиция:

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

Оператор за интеграл се генерира с командата `\int`, **оператор за сума** – с командата `\sum`, **оператор за произведение** – с командата `\prod`. Горните и долните граници се указват с помощта на знаците « \wedge » и « \underset », като горни и долни индекси¹.

```
\begin{displaymath}
\sum_{i=1}^n \quad \quad \int_0^{\frac{\pi}{2}} \quad \quad \prod_{\epsilon}
\end{displaymath}
```

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

За да имате по-голям контрол при разполагането на индексите в сложни изрази, `amsmath` предоставя още два инструмента: командата `\substack` и средата `subarray`:

¹`AMS-LATEX`, освен това, поддържа многоредови горни и долни индекси.

```

\begin{displaymath}
\sum_{\substack{0<i<n \\ 1<j<m}}
P(i,j) =
\sum_{\begin{subarray}{l} i \in I \\ 1<j<m \end{subarray}}
Q(i,j)
\end{displaymath}

```

$$\sum_{\substack{0<i<n \\ 1<j<m}} P(i,j) = \sum_{\substack{i \in I \\ 1<j<m}} Q(i,j)$$

TeX предоставя много символи за **скоби** и други ограничители (например [< || ↓]). Кръгли и квадратни скоби могат да се въвеждат със съответните клавиши, фигурни скоби – с \{, но всички други ограничители се генерират със специални команди (например \updownarrow). Списък на достъпните ограничители вижте в Таблица 3.8 на страница 68.

```

\begin{displaymath}
\{a,b,c\} \neq \{a,b,c\}
\end{displaymath}

```

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

Ако поставите пред отварящ ограничител командата \left или пред затварящ – \right, то TeX автоматично избира точния размер на ограничителя (разтяга го). Забележете, че трябва да затваряте всяко \left със съответното \right и че размерът на ограничителите се определя коректно само ако двата се отпечатват на един и същ ред. Ако не искате да имате десен ограничител, използвайте невидимия ограничител „\right.“!

```

\begin{displaymath}
1 + \left( \frac{1}{1-x^2} \right)^3
\end{displaymath}

```

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

В някои случаи е необходимо да се определи ръчно точният размер на математическия ограничител. Това може да се направи чрез командите \big, \Big, \bigg и \Bigg, използвани като префикси към повечето команди за ограничители.

```

\Big( (x+1)(x-1) \Big)^2 \\
\big(\Big(\bigg(\Bigg(\quad \\
\big)\Big)\bigg)\Big)\quad \\
\big\|\Big\|\bigg\|\Big\|\$

```

$$\left((x+1)(x-1) \right)^2 \\ \left(\left(\left(\left(\right) \right) \right) \right) \quad \Big\| \Big\| \bigg\| \Big\|$$

Има няколко команди за въвеждане на **три точки** във формула. \ldots отпечатва точките на основната линия, а \cdots ги центрира. Освен това, съществуват команди \vdots за вертикални и \ddots за диагонални точки. В Раздел 3.5 ще намерите друг пример.

```
\begin{displaymath}
x_{1}, \dots, x_{n} \quad \quad
x_{1} + \dots + x_{n}
\end{displaymath}
```

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

3.4 Математически интервали

Ако избраните от $\text{T}_\text{E}\text{X}$ интервали вътре във формула не ви удовлетворяват, можете да ги промените чрез използване на специални команди за интервали. Има няколко команди за малки интервали: `\`, за $\frac{3}{18}$ quad (`\!`), `\!`: за $\frac{4}{18}$ quad (`\!`) и `\!`: за $\frac{5}{18}$ quad (`\!`). Екранният символ за интервал `\!` дава среден размер интервал, а `\quad` (`\!`) и `\qquad` (`\!`) генерират по-големи интервали. Размерът `\quad` примерно съответства на ширината на буквата ‘M’ в текущия шрифт. Командата `\!` произвежда отрицателен интервал с размер $-\frac{3}{18}$ quad (`\!`).

```
\newcommand{\upd}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int\int_{D} g(x,y)
\quad \quad \quad \upd x \, \upd y
\end{displaymath}
вместо
\begin{displaymath}
\int\int_{D} g(x,y)\upd x \upd y
\end{displaymath}
```

вместо

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

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

Прието е, ‘d’ в диференциала да се отпечата в прав шрифт.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{I}\mathcal{A}\mathcal{T}\text{E}\mathcal{X}$ поддържа друг начин за тънка настройка на интервалите между няколко знака за интеграл: командите `\iint`, `\iiint`, `\iiiint` и `\idotsint`. Със зареждане на пакета `amsmath` горният пример може да бъде набран така:

```
\newcommand{\upd}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \quad \quad \quad \upd x \, \upd y
\end{displaymath}
```

$$\iint_D \, dx \, dy$$

За повече подробности вижте електронния документ `textmath.tex` (от документацията на $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{I}\mathcal{A}\mathcal{T}\text{E}\mathcal{X}$) или Глава 8 на книгата *The $\mathcal{I}\mathcal{A}\mathcal{T}\text{E}\mathcal{X}$ Companion* [3].

3.5 Вертикално разположен материал

За отпечатване на **матрици** се използва средата `array`. Тя работи подобно на средата `tabular`. Командата `\!` се използва за прекъсване на реда.

```

\begin{displaymath}
\mathbf{X} = \left(
\begin{array}{ccc}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{array}
\right)
\end{displaymath}

```

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

Средата `array` може също да се използва за отпечатване на изрази, които имат един голям ляв ограничител и се поставя «.» като невидим десен ограничител:

```

\begin{displaymath}
y = \left\{
\begin{array}{ll}
a & \text{ако } d > c \\
b + x & \text{сутрин} \\
l & \text{през другото време}
\end{array}
\right.
\end{displaymath}

```

$$y = \begin{cases} a & \text{ако } d > c \\ b + x & \text{сутрин} \\ l & \text{през другото време} \end{cases}$$

Точно както и в средата `tabular`, можете да рисувате и линии в средата `array`, например, разделящи елементите на матрица:

```

\begin{displaymath}
\left( \begin{array}{c|c}
1 & 2 \\ \hline
3 & 4
\end{array} \right)
\end{displaymath}

```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array} \right)$$

За формули, заемачи няколко реда или за системи уравнения могат да се използват средите `eqnarray` и `eqnarray*` вместо `equation`. В `eqnarray` всеки ред получава отделен номер на уравнение. В `eqnarray*` номерът не се поставя никъде.

Средите `eqnarray` и `eqnarray*` работят подобно на таблици с три стълба с формат `{rcl}`, където средният стълб се използва за знак за равенство, знак за неравенство, или друг подходящ знак. Командата `\\` прекъсва реда.

```

\begin{eqnarray}
f(x) & = & \cos x \\
f'(x) & = & -\sin x \\
\int_0^x f(y)dy & = & \sin x
\end{eqnarray}

```

$$f(x) = \cos x \quad (3.5)$$

$$f'(x) = -\sin x \quad (3.6)$$

$$\int_0^x f(y)dy = \sin x \quad (3.7)$$

Забележете, че от двете страни на средната колона, около знаците за равенство, има твърде много свободно място. То може да бъде редуцирано с установяването `\setlength\arraycolsep{2pt}`, както в следващия пример.

Дълги уравнения не се разбиват автоматично на равни части. Авторът трябва да посочи, къде да бъдат разбити и как да се подравнят. Най-често се използват следните два метода:

```
\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
& & \frac{x^7}{7!} + \dots
\end{eqnarray}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (3.8)$$

```
\begin{eqnarray}
\left. \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \right\}
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \quad (3.9)$$

Командата `\nonumber` заставя L^AT_EX да не генерира номер за това уравнение.

С тези методи може би е сложно да се получат добре изглеждащи вертикално подравнени уравнения; по-мощна алтернатива предоставя пакета `amsmath` (вж. средите `align`, `flalign`, `gather`, `multiline` и `split`).

3.6 Фантоми

L^AT_EX позволява използване на фантоми (невидими обекти) за някои интересни пространствени трикове при разполагане на видимите обекти.

Когато разполагате текст по вертикала с помощта на командите `^` и `_`, можете да използвате командата `\phantom`, за да запазите пространство за символи, които всъщност няма да се появят на страницата. По-добре това ще бъде разбрано от следните примери.

```
\begin{displaymath}
{}^{12}_6\text{C} \quad \phantom{{}^{12}_6\text{C}} \quad \text{вместо} \quad {}^{12}_6\text{C}
\end{displaymath}
```

$${}^{12}_6\text{C} \quad \text{вместо} \quad {}^{12}_6\text{C}$$

Това е един от примерите, когато са необходими по-големи скоби, отколкото предоставяните стандартни `\left[\right]`. Използват се командите `\biggl` и `\biggr` съответно за лява и дясна скоби.

3.8 Теорема, лема, закони, ...

При писане на математически документи, често се налага да се пишат «леми», «определения», «аксиоми» и аналогични структури. \LaTeX поддържа това с командите

```
\newtheorem{име}[брояч]{текст}[раздел]
```

Аргументът *име* е кратка ключова дума, използвана за идентификация на «теоремата». Аргументът *текст* определя истинското име на «теоремата», което ще се отпечата в документа.

Аргументите в квадратни скоби са незадължителни. И двата се използват за определяне на това, как да се номерира «теоремата». С аргумента *брояч* може да се укаже *име* на по-рано обявена «теорема». Новата «теорема» тогава ще се номерира в същата последователност. Аргументът *раздел* (`chapter`, `section`) позволява да определите тип на номерацията на «теоремата», при който автоматично поставяния номер се образува от номера на раздела (главата) и поредния номер на «теоремата».

След като поставите в преамбюла на документа всички команди `\newtheorem`, можете използвате следната команда вътре в текста:

```
\begin{име}[текст]
Това е интересна теорема.
\end{име}
```

Тази теория трябва да е достатъчно. Следващите примери трябва да изяснят всичко казано по-горе и окончателно да ви убедят, че средата `\newtheorem` не е чак толкова сложна за разбиране.


```

% определения за
% преамбюла на документа
\newtheorem{law}{Правило}
\newtheorem{jury}[law]{Извод}
% в тялото на документа
\begin{law} \label{law:box}
Който не работи, не огладнява!
\end{law}
\begin{jury}[Народен]
Ако човек огладнява, трябва да работи!
(вж. Правило~\ref{law:box})
\end{jury}
\begin{law}
Ситият на гладния не вярва.
\end{law}

```

Правило 1. *Който не работи, не огладнява!*

Извод 2 (Народен). *Ако човек огладнява, трябва да работи! (вж. Правило 1)*

Правило 3. *Ситият на гладния не вярва.*

Теоремата «Извод» използва същия брояч, както и теоремата «Правило». Следователно, тя получава пореден номер заедно с другите теореми «Правило». Аргументът в квадратни скоби в текста, указва специфично заглавие на теоремата, или нещо подобно.

```

\flushleft
\newtheorem{mur}{Мърфи}[section]
\begin{mur}
Ако съществуват два или повече
начина да се направи нещо, и
един от тях може да доведе до
катастрофа, то някой непременно
ще направи точно това.
\end{mur}

```

Мърфи 3.8.1. *Ако съществуват два или повече начина да се направи нещо, и един от тях може да доведе до катастрофа, то някой непременно ще направи точно това.*

Теоремата «Мърфи» получава номер, свързан с номера на текущия раздел. Можете да използвате и друга структурна единица, например, глава или под-раздел.

3.9 Дебели (bold) символи

В \LaTeX далеч не е просто да се получат дебели символи. Това, вероятно, е направено умишлено, защото е непрофесионално твърде често да се злоупотребява с тях. Командата за смяна на шрифта $\text{\code{\mathbf}}$ дава дебели символи, но те са голям (прав) шрифт, докато математическите символи обикновено са наклонени. Съществува команда $\text{\code{\boldmath}}$, но *тя може да бъде използвана само извън математически режим*. Същото се отнася и за символите.

```
\begin{displaymath}
\mu, M \quad \mathbf{M} \quad \boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

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

Пакетът `amsbsy` (включван с пакета `amsmath`), както и пакетът `bm` (от инструментите `tools`), включват командата `\boldsymbol`.

```
\begin{displaymath}
\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}$$

3.10 $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$

$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ е набор от макроси, създаден от Американското Математическо Общество (American Mathematics Society). Пакетът $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ предлага изключително удобни команди за писане на математически текст. За подробно запознаване с $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, вижте [CTAN:/tex-archive/macros/latex/required/amslatex](http://ctan.org/tex-archive/macros/latex/required/amslatex).

За да се използват тези макроси, трябва да се зареди съответният пакет в преамбула с командата `\usepackage`.

В Раздел 3.11 на стр. 66 има таблици с най-често използваните символи в пакета $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$.

3.10.1 Сравнение между `eqnarray` и `align`

$\mathcal{A}\mathcal{M}\mathcal{S}$ средата `align` дава по-добро разстояние между символите в уравнение, в сравнение със стандартната среда `eqnarray`.

```
Сравнете
\begin{eqnarray}\label{eq:2}
2x+3&=&7\\
2x+3-3&=&7-3\\
2x&=&4
\end{eqnarray}
с
\begin{align}\label{eq:3}
2x+3&=&7\\
2x+3-3&=&7-3\\
2x&=&4.
\end{align}
```

Сравнете

$$2x + 3 = 7 \quad (3.11)$$

$$2x + 3 - 3 = 7 - 3 \quad (3.12)$$

$$2x = 4 \quad (3.13)$$

с

$$2x + 3 = 7 \quad (3.14)$$

$$2x + 3 - 3 = 7 - 3 \quad (3.15)$$

$$2x = 4. \quad (3.16)$$

Номерирането на всички уравнения в масив може да се забрани, ако вместо `align` се използва средата `align*`.

3.10.2 Команда `nonumber` и цитиране

Номерирането на отделни уравнения в средата `align` може да се забрани чрез командата `\nonumber`, а ако трябва да се цитират отделни уравнения, се поставя етикет чрез командата `\label`. Ето пример:

Разглежда се

```
\begin{align}
2x+3&=7\label{ani}\!
2x+3-3&=7-3\nonumber \!
2x&=4\nonumber \!
\frac{2x}{2}&=\frac{4}{2}\nonumber \!
x&=2.\label{tony}
\end{align}
```

Уравнение `\eqref{ani}` е
еквивалентно на
Уравнение `\eqref{tony}`.

Разглежда се

$$2x + 3 = 7 \quad (3.17)$$

$$2x + 3 - 3 = 7 - 3$$

$$2x = 4$$

$$\frac{2x}{2} = \frac{4}{2}$$

$$x = 2. \quad (3.18)$$

Уравнение (3.17) е еквивалентно на Уравнение (3.18).

За да получите коректни цитати на формулите, трябва да компилирате файла два пъти с $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$.

3.10.3 Команда `subequations`

В средата `subequations` се форматира номерирането на «подуравнения»:

Разглежда се системата

```
\begin{subequations}\label{system}
\begin{align}
2x+3y&=7\label{sys1}\!
3x-4y&=11\label{sys2}
\end{align}
\end{subequations}
```

В системата `\eqref{system}`, първо се решава Уравнение `\eqref{sys1}` за x .

Разглежда се системата

$$2x + 3y = 7 \quad (3.19a)$$

$$3x - 4y = 11 \quad (3.19b)$$

В системата (3.19), първо се решава Уравнение (3.19a) за x .

Командата `\label` непосредствено след `\begin{subequations}` поставя общ номер на двойката уравнения. Цитирането на отделните уравнения произвежда «под-номер».

3.10.4 Дълги формули на няколко реда

Използвайте $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ средата `multline` за изобразяване на няколко реда на дълги формули. Въпреки, че следващият пример математически не е верен, той дава ясна представа как се използва средата `multline`.

Използвайте `copy` и `paste`, за да получите по-лесно следния запис.

```
\begin{multline}
e^x=1+x+\frac{x^2}{2!}+\dots \\
1+x+\frac{x^2}{2!}+\dots \\
1+x+\frac{x^2}{2!}+\dots \\
1+x+\frac{x^2}{2!}+\dots
\end{multline}
```

Използвайте `copy` и `paste`, за да получите по-лесно следния запис.

$$\begin{aligned}
e^x &= 1 + x + \frac{x^2}{2!} + \dots \\
&1 + x + \frac{x^2}{2!} + \dots \\
&1 + x + \frac{x^2}{2!} + \dots \\
&1 + x + \frac{x^2}{2!} + \dots \quad (3.20)
\end{aligned}$$

Забележете, че първият ред е разположен вляво (с малко разстояние), средните редове са в средата, а последният ред е вдясно (също с малко разстояние).

3.10.5 Случаи (cases) и текст във формули

По части зададена функция се въвежда лесно в $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ средата `cases`. Освен това, командата `\text` улеснява въвеждането на текст вътре в математически изрази.

Дадена е функция, зададена на части по следния начин:

```
\begin{equation}
f(x)=
\begin{cases}
2x-3 & \text{ако } x < 0, \\
3 & \text{ако } x \geq 0.
\end{cases}
\end{equation}
```

Функциите f и g са обратни тогава и само тогава, когато

```
$$
f(g(x))=x \quad \text{и} \quad g(f(x))=x.
$$
```

Дадена е функция, зададена на части по следния начин:

$$f(x) = \begin{cases} 2x - 3 & \text{ако } x < 0, \\ 3 & \text{ако } x \geq 0. \end{cases} \quad (3.21)$$

Функциите f и g са обратни тогава и само тогава, когато

$$f(g(x)) = x \quad \text{и} \quad g(f(x)) = x.$$

3.10.6 Матрици в $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{I}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$

Пакетът `amsmath` поддържа няколко среди за матрици, подобни на основната среда `array` в $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. Това са средите `pmatrix`, `bmatrix`, `Vmatrix`, `vmatrix` и `Vmatrix`, които имат вградени ограничители съответно (\cdot) , $[\cdot]$, $\{\cdot\}$, $|\cdot|$ и $\|\cdot\|$.

Нека A е $m \times n$ матрица

```

 $A=$ 
\begin{pmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \dots & a_{mn}
\end{pmatrix}.

```

Нека A е $m \times n$ матрица

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}.$$

Нека A е $m \times n$ матрица

```

 $A=$ 
\begin{bmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \dots & a_{mn}
\end{bmatrix}.

```

Нека A е $m \times n$ матрица

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

Нека A е $m \times n$ матрица

```

 $A=$ 
\begin{Bmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \dots & a_{mn}
\end{Bmatrix}.

```

Нека A е $m \times n$ матрица

$$A = \left\{ \begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right\}.$$

Нека A е $m \times n$ матрица

```

 $A=$ 
\begin{vmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \dots & a_{mn}
\end{vmatrix}.

```

Нека A е $m \times n$ матрица

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix}.$$

Нека A е $m \times n$ матрица

```

 $A=$ 
\begin{Vmatrix}
a_{11} & a_{12} & \dots & a_{1n} \\
a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \dots & a_{mn}
\end{Vmatrix}.

```

Нека A е $m \times n$ матрица

$$A = \left\| \begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right\|.$$

3.11 Списък на математическите символи

В следващите таблици ще намерите всички символи, достъпни обикновено в *математически режим*.

За достъп до символите, изброени в Таблици 3.12–3.16, в преамбюла на документа трябва да бъде зареден пакета `amssymb` и в системата трябва да бъдат установени математическите шрифтове AMS. Ако пакетът и шрифтовете AMS не са установени във вашата система, вижте на CTAN:/tex-archive/macros/latex/required/amslatex. Още по-пълнен списък на символите можете да намерите на адрес CTAN:/tex-archive/info/symbols/comprehensive.

Таблица 3.1: Акценти в математически режим

\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 3.2: Малки гръцки букви

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	υ	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		

Таблица 3.3: Главни гръцки букви

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

Таблица 3.4: Бинарни отношения

Можете да получите съответните отрицания чрез добавяне пред следните символи на командата `\not`.

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> или <code>\le</code>	\geq	<code>\geq</code> или <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset ¹	<code>\sqsubset</code> ¹	\sqsupset ¹	<code>\sqsupset</code> ¹	\bowtie	<code>\bowtie</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\propto	<code>\propto</code>
\in	<code>\in</code>	\ni , \owns	<code>\ni</code> , <code>\owns</code>	\models	<code>\models</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\perp	<code>\perp</code>
$ $	<code>\mid</code>	\parallel	<code>\parallel</code>	\asymp	<code>\asymp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\neq	<code>\neq</code> или <code>\ne</code>
$:$	<code>:</code>	\notin	<code>\notin</code>		

¹За достъп до този символ ползвайте пакета `latexsym`.

Таблица 3.5: Бинарни оператори

$+$	<code>+</code>	$-$	<code>-</code>	\triangleleft	<code>\triangleleft</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\triangleangleright	<code>\triangleangleright</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\star	<code>\star</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\ast	<code>\ast</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\circ	<code>\circ</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\bullet	<code>\bullet</code>
\vee , \lor	<code>\vee</code> , <code>\lor</code>	\wedge , \land	<code>\wedge</code> , <code>\land</code>	\diamond	<code>\diamond</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\uplus	<code>\uplus</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\amalg	<code>\amalg</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\dagger	<code>\dagger</code>
\bigtriangleup	<code>\bigtriangleup</code>	\bigtriangledown	<code>\bigtriangledown</code>	\ddagger	<code>\ddagger</code>
\triangleleft ¹	<code>\triangleleft</code> ¹	\triangleright ¹	<code>\triangleright</code> ¹	\wr	<code>\wr</code>
\untriangleleft ¹	<code>\untriangleleft</code> ¹	\untriangleright ¹	<code>\untriangleright</code> ¹		

Таблица 3.6: Големи оператори

\sum	<code>\sum</code>	\cup	<code>\bigcup</code>	\vee	<code>\bigvee</code>	\oplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\cap	<code>\bigcap</code>	\wedge	<code>\bigwedge</code>	\otimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\sqcup	<code>\bigsqcup</code>			\odot	<code>\bigodot</code>
\int	<code>\int</code>	\oint	<code>\oint</code>			\oplus	<code>\bigoplus</code>

Таблица 3.7: 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>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</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 (голям интервал)	<code>\iff</code> (голям интервал)	\leadsto	<code>\leadsto</code> ¹

¹За достъп до този символ ползвайте пакета `latexsym`.

Таблица 3.8: Ограничители

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
$[$	<code>[</code> или <code>\lbrack</code>	$]$	<code>]</code> или <code>\rbrack</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
$\{$	<code>\{</code> или <code>\lbrace</code>	$\}$	<code>\}</code> или <code>\rbrace</code>	\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> </code> или <code>\vert</code>	$\ $	<code>\ </code> или <code>\Vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>				

Таблица 3.9: Големи ограничители

$\left($	<code>\lgroup</code>	$\right)$	<code>\rgroup</code>	$\left\{$	<code>\lmoustache</code>	$\right\}$	<code>\rmoustache</code>
$\left $	<code>\arrowvert</code>	$\right $	<code>\Arrowvert</code>	$\left\{$	<code>\bracevert</code>	$\right\}$	

Таблица 3.10: Други символи

\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>	ℓ	<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 ¹	<code>\mho</code>	∂	<code>\partial</code>
$'$	<code>'</code>	$'$	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>	\square	<code>\Box</code> ¹	\diamond	<code>\Diamond</code> ¹
\perp	<code>\bot</code>	\top	<code>\top</code>	\sphericalangle	<code>\angle</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>

¹ За достъп до този символ ползвайте пакета `latexsym`.

Таблица 3.11: Не-математически символи

Тези символи могат да се използват и в текстов режим.

\dagger	<code>\dag</code>	\S	<code>\S</code>	\copyright	<code>\copyright</code>	\textregistered	<code>\textregistered</code>
\ddagger	<code>\ddag</code>	\P	<code>\P</code>	\pounds	<code>\pounds</code>	$\%$	<code>\%</code>

Таблица 3.12: Ограничители AMS

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>
\lvert	<code>\lvert</code>	\rvert	<code>\rvert</code>	\lVert	<code>\lVert</code>	\rVert	<code>\rVert</code>

Таблица 3.13: Букви от гръцки и иврит AMS

\digamma	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>	\gimel	<code>\gimel</code>	\daleth	<code>\daleth</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	----------	---------------------	-----------	----------------------

Таблица 3.14: Бинарни отношения AMS

\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\doteqdot или <code>\Doteq</code>	<code>\doteqdot</code> или <code>\Doteq</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\eqslantless	<code>\eqslantless</code>	\eqslantgtr	<code>\eqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll или \llless	<code>\lll</code> или <code>\llless</code>	\ggg или \gggtr	<code>\ggg</code> или <code>\gggtr</code>	\circeq	<code>\circeq</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\thicksim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\gtreqqless	<code>\gtreqqless</code>	\thickapprox	<code>\thickapprox</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\approxeq	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsim	<code>\backsim</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\subseteqeq	<code>\subseteqeq</code>	\supseteqeq	<code>\supseteqeq</code>	\Vdash	<code>\Vdash</code>
\Subset	<code>\Subset</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\backepsilon	<code>\backepsilon</code>
\therefore	<code>\therefore</code>	\because	<code>\because</code>	\varpropto	<code>\varpropto</code>
\shortmid	<code>\shortmid</code>	\shortparallel	<code>\shortparallel</code>	\between	<code>\between</code>
\smallsmile	<code>\smallsmile</code>	\smallfrown	<code>\smallfrown</code>	\pitchfork	<code>\pitchfork</code>
\vartriangleleft	<code>\vartriangleleft</code>	\vartriangleright	<code>\vartriangleright</code>	\blacktriangleleft	<code>\blacktriangleleft</code>
\trianglelefteq	<code>\trianglelefteq</code>	\trianglerighteq	<code>\trianglerighteq</code>	\blacktriangleright	<code>\blacktriangleright</code>

Таблица 3.15: Стрелки AMS

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>	\multimap	<code>\multimap</code>
\leftleftarrows	<code>\leftleftarrows</code>	\rightrightarrows	<code>\rightrightarrows</code>	\upuparrows	<code>\upuparrows</code>
\leftrightarrows	<code>\leftrightarrows</code>	\rightleftarrows	<code>\rightleftarrows</code>	\downdownarrows	<code>\downdownarrows</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\upharpoonleft	<code>\upharpoonleft</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>	\upharpoonright	<code>\upharpoonright</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>	\downharpoonleft	<code>\downharpoonleft</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>	\downharpoonright	<code>\downharpoonright</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>		
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>		

Таблица 3.16: Отрицателни бинарни отношения и стрелки AMS

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\nsubseteqeq
\nleqslant	\ngeqslant	\nsupseteqeq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\nleqq	\ngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\nprecneqq	\nsuccneqq	\nvDash
\nprecnsim	\succnsim	\nVdash
\nprecnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\nsubseteqeq	\nsupseteqeq	\ntrianglelefteq
\subseteqeq	\supseteqeq	\ntrianglerighteq
\nleftarrow	\rightarrow	\nleftrightarrow
\nLeftarrow	\Rightarrow	\nLeftrightarrow

Таблица 3.17: Бинарни оператори AMS

\dotplus	\centerdot	\intercal
\ltimes	\rtimes	\divideontimes
\Cup или \doublecup	\Cap или \doublecap	\smallsetminus
\veebar	\barwedge	\doublebarwedge
\boxplus	\boxminus	\circleddash
\boxtimes	\boxdot	\circledcirc
\leftthreetimes	\rightthreetimes	\circledast
\curlyvee	\curlywedge	

Таблица 3.18: Други символи AMS

\hbar	<code>\hbar</code>	\hbar	<code>\hslash</code>	\mathbb{k}	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\textcircled{S}	<code>\circledS</code>
\triangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
∇	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\supset	<code>\Game</code>
\diamond	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\star	<code>\bigstar</code>
\sphericalangle	<code>\angle</code>	\sphericalangle	<code>\measuredangle</code>	\sphericalangle	<code>\sphericalangle</code>
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\backprime	<code>\backprime</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\varnothing	<code>\varnothing</code>
\eth	<code>\eth</code>	\mho	<code>\mho</code>		

Таблица 3.19: Математически азбуки

Пример	Команда	Изискван пакет
ABCdef	<code>\mathrm{ABCdef}</code>	
ABCdef	<code>\mathit{ABCdef}</code>	
\mathnormal{ABCdef}	<code>\mathnormal{ABCdef}</code>	
\mathcal{ABC}	<code>\mathcal{ABC}</code>	euscript с опция: mathcal
\mathscr{ABC}	<code>\mathscr{ABC}</code>	mathrsfs
\mathfrak{ABCdef}	<code>\mathfrak{ABCdef}</code>	eufrak
\mathbb{ABC}	<code>\mathbb{ABC}</code>	amsfonts или amssymb

Глава 4

СПЕЦИАЛНИ ВЪЗМОЖНОСТИ

При работа с голям документ, \LaTeX ще ви помогне с някои допълнителни възможности, като вмъкване на графики, генериране на Азбучен указател, управление на Библиография и други подобни. По-подробно описание на специалните възможности и разширения на \LaTeX има в *L^AT_EX Manual* [1] и в *The L^AT_EX Companion* [3].

4.1 Включване на Encapsulated POSTSCRIPT графики

\LaTeX дава основните средства за работа с плаващи обекти, такива, като фигури и таблици, с помощта на средите `figure` и `table`.

Съществуват също няколко способа за създаване на собствена графика със средствата на базовия \LaTeX или негови разширения, описани в Глава 4.10. Допълнителна информация е дадена в *The L^AT_EX Companion* [3] и в *L^AT_EX Manual* [1].

Много по-лесен начин за получаване на графики в документа е те да се създадат със специализирани програмни пакети¹ и да се включват в документа готовите картинки.

Също и тук, \LaTeX -пакетите предлагат множество начини да се прави това, но в тази глава се обсъжда само използване на графики във формат Encapsulated POSTSCRIPT (EPS), тъй като това се прави лесно и е широко разпространено.

За да се използват картинки във формат EPS, трябва да има на разположение POSTSCRIPT принтер за печат.²

¹Такива, като XFig, CorelDraw, Freehand, Gnuplot, ...

²Друга възможност за извеждане на POSTSCRIPT е чрез използване на програмата GHOSTSCRIPT, достъпна от CTAN:/tex-archive/support/ghostscript. Потребителите на Windows и OS/2 могат да обърнат внимание на програмата GSVIEW.

Добър набор команди за включване на графики има в пакета `graphicx` (автор D. P. Carlisle). Той е част от цяло семейство пакети, наречени комплект “graphics”.¹

При предположение, че работите в система с достъпен за извеждане POSTSCRIPT-принтер и с инсталиран пакет `graphicx`, можете да използвате следната постъпкова инструкция за включване на картинка във вашия документ:

1. Експортирайте картинката от вашата графична програма във формат EPS.
2. В преамбюла на документа заредете пакета `graphicx` с командата

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

където *драйвер* е името на вашата конвертираща програма от DVI в POSTSCRIPT. Най-широко използвания конвертор се нарича `dvips`. Името на драйвера се изисква затова, защото не съществува стандарт за включване на графика в TeX. Знаейки името на *драйвера*, `graphicx` избира точния метод за включване на информация за графиката в `.dvi` файла така, че принтерът да я разбере и да успее коректно да включи `.eps` файла.

3. Използвайте командата

```
\includegraphics[опция=стойност, ...]{файл}
```

за включване на *файл* във вашия документ. Незадължителният параметър допуска списък от разделени със запетаи *опции* и съответните им *стойности*. *Опциите* могат да се използват за изменение на ширината, височината или ъгъла на завъртане на включваната графика. Таблица 4.1 изброява най-важните опции.

Таблица 4.1: Имена на опциите на пакета `graphicx`

<code>width</code>	мащабира графиката до указаната ширина
<code>height</code>	мащабира графиката до указаната височина
<code>angle</code>	завърта графиката обратно на часовата стрелка
<code>scale</code>	мащабира графиката

¹CTAN:/tex-archive/macros/latex/required/graphics

Следващият пример прави нещата по-ясни:

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

Тук се включва графика, записана във файл `test.eps`. Тя *отначало* се завърта на 90 градуса и *след това* се мащабира до крайна ширина 0.5 от ширината на стандартния параграф. Пропорциите се съхраняват, защото не е указана конкретна височина.

Параметрите *височина* и *ширина* могат също да бъдат указани в абсолютни размери. Вижте Таблица 6.5 на страница 125 за повече информация. Ако искате да знаете за това повече, прочетете [10] и [14].

4.2 Библиография

Библиография се генерира в средата `thebibliography`. Всеки елемент започва с

```
\bibitem[етикет]{маркер}
```

След това *маркер* се използва за цитиране на книгата, или статията в документа.

```
\cite{маркер}
```

Ако не използвате опцията *етикет*, елементите на библиографията се номерират автоматично. Параметърът след командата `\begin{thebibliography}` определя колко място да се резервира за номерата на етикетите.

В следващия пример, `{99}` указва на L^AT_EX, че нито един от номерата на етикетите няма да е по-широк от числото ‘99’.

```
Knuth~\cite{knu} e
предложил \ldots
```

```
\begin{thebibliography}{99}
\bibitem{knu} D.~Knuth:
\emph{The \TeX book},
Addison-Wesley, Reading,
second edition, 1984.
\end{thebibliography}
```

Knuth [1] е предложил ...

Библиография

[1] D. Knuth: *The T_EXbook*, Addison-Wesley, Reading, second edition, 1984.

За големи проекти се използва програмата BibT_EX. Тя се включва във всички съвременни дистрибутиви на T_EX.

BibT_EX позволява да поддържате библиографска база данни и да извличате от нея елементи, свързани с това, което цитирате във вашия документ. Визуалното представяне на библиографията е в различни стилове, което позволява да се създава библиография, като се използва широк асортимент от установени стилове.

4.3 Азбучен указател

Всяка истинска книга има една полезна особеност – Азбучен указател. С помощта на L^AT_EX и съпровождащата програма `makeindex`, указателят може да се създава сравнително лесно.

Тук са дадени само базовите команди за генерация на указателя. По-задълбочено изложение вижте в *The L^AT_EX Companion* [3].

За включване на възможностите на L^AT_EX, в преамбюла на документа трябва да се зареди пакетът `makeidx`:

```
\usepackage{makeidx}
```

и трябва да бъдат разрешени специалните команди на указателя чрез поставяне в преамбюла на командата

```
\makeindex
```

Съдържанието на указателя се формира в тялото на документа с

Таблица 4.2: Примерен синтаксис на ключове за указателя

Пример	Вид на указателя	Коментар
<code>\index{уравнение}</code>	уравнение, 1	обикновен елемент
<code>\index{уравнение!корен}</code>	корен, 3	Подчинен на ‘уравнение’
<code>\index{Гаус@\textsl{Гаус}}</code>	<i>Гаус</i> , 2	форматиран ключ
<code>\index{Нютон@\textbf{Нютон}}</code>	Нютон , 7	същото като горното
<code>\index{Крамер textbf}</code>	Крамер, 3	форматирана страница
<code>\index{Якоби textit}</code>	Якоби, 5	същото като горното
<code>\index{коса@кос\'а}</code>	ко ^с а, 4	акцентирани букви

командата

```
\index{ключ}
```

където *ключ* е елемент на указателя. Командите `\index` се поставят в тялото на документа, на онези места в текста, където трябва да сочи съответният елемент на указателя. Таблица 4.2 обяснява синтаксиса на аргумента *ключ* с няколко примера.

При обработката на входния файл с \LaTeX , всяка команда `\index` записва в специален файл съответния елемент на указателя, заедно с номера на текущата страница. Файлът има същото име като входния \LaTeX -файл, но друго разширение (`.idx`). Този `.idx`-файл след това се обработва с програмата `makeindex`.

```
makeindex filename
```

Програмата `makeindex` генерира сортиран указател със същото име, но този път – с разширение `.ind`.

Ако след това отново се компилира входния файл, този сортиран указател се включва в документа на това място, където \LaTeX намира командата

```
\printindex
```

Пакетът `showidx`, влизащ в $\text{\LaTeX} 2_{\epsilon}$, печата всички елементи на указателя в лявото поле на текста. Това е особено полезно при проверка на текста и сверяване с указателя.

Да отбележим, че командата `\index`, ако се използва невнимателно, може да повлияе на вида на отпечатване.

Важна дума `\index{дума}`. За разлика от `тази дума\index{дума}`. Забележете положението на точката.

Важна дума `.` За разлика от тази дума. Забележете положението на точката.

4.4 Включване на речник

В стандартния дистрибутив на обвивката WinEdt са включени речници за автоматична проверка на командите на \LaTeX , както и различни речници за английски език. Речници за други езици има на адрес CTAN:/tex-archive/systems/win32/winedt/dict. Ако желаете да включите речник за друг език, например български, следвайте следната последователност:

1. Създайте текстов файл `bulgaria.dic`, във всеки ред на който има само по една българска дума, записана от първа позиция. Създаването на такъв файл ще ви отнеме много време, ако искате речника ви да включва повече думи, затова най-добре ми пишете на електронния адрес, посочен в началото на книгата и аз ще ви го изпратя.
2. В директорията `... \WinEdt Team \WinEdt \Dict` се създава нова директория `bulgarian`.
3. Копирайте файла `bulgaria.dic` в създадената директория `bulgarian`.
4. От менюто на WinEdt `-> Options -> Dictionary`
 - (a) Insert
 - Dictionaries `-> Bulgarian (Bulgaria)`
 - Browse `-> ... \Dict \bulgaria.dic`
 - Enabled `-> Bg`
 - Load on Start
 - Save on Exit
 - Use for Completion
 - (б) Update
 - (в) Extract `-> Dictionaries.dat -> Save`
 - (г) OK
5. Вътре във файл на български език:
 - Document Settings `-> Document Mode -> TeX:Bg`

4.5 Настройка на колонтитули

Пакетът `fancyhdr`, написан от Piet van Oostrum, предоставя няколко прости команди, които позволяват да настройвате горния и долния колонтитули на документа. Ако сега погледнете горе на тази страница, ще видите едно от възможните приложения на този пакет.

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% С това сме сигурни, че заглавията на глави и
% раздели са в долен регистър.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} %изтрива текущите установявания за колонтитулите
\fancyhead[LE,R0]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} %оставя място за линия
\fancypagestyle{plain}{%
  \fancyhead{} %изчиства колонтитулите на обикновените страници
  \renewcommand{\headrulewidth}{0pt} % и линията
}

```

Фигура 4.1: Пример за настройка на `fancyhdr`

Сложността в настройката на колонтитулите е в това, че се включват неща, касаещи заглавията на разделите и главите. \LaTeX реализира това посредством двуетапен подход. При дефинирането на колонтитулите се използват командите `\rightmark` и `\leftmark`, за определяне съответно на заглавията на текущите Раздел и Глава. Стойностите на тези две команди се изменят при обработка с командите `\chapter` или `\section`.

За по-голяма гъвкавост, командата `\chapter` и нейните подобни не предефинират `\rightmark` и `\leftmark` сами, а извикват друга команда, наречена `\chaptermark`, `\sectionmark` или `\subsectionmark`, отговаряща за преопределяне на `\rightmark` и `\leftmark`.

Така че, ако искате да измените вида на названията на главите в горния колонтитул, вие просто преопределяте командата `\chaptermark`.

Фигура 4.1 показва, как може да се настрои пакета `fancyhdr` така, че колонтитулите да изглеждат почти така, както изглеждат в тази книга.

Най добре е обаче преди да настройвате колонтитулите, да се запознаете с документацията към пакета на адреса [CTAN:/tex-archive/macros/latex/contrib/supported/fancyhdr](http://ctan.org/tex-archive/macros/latex/contrib/supported/fancyhdr).

4.6 Пакет `verbatim`

По-горе в тази книга се запознахте със *средата* `verbatim`. Съществува също и *пакет* `verbatim`. Пакетът `verbatim` представлява повторна реализация на средата `verbatim`, с поправки на някои негови ограничения и добавяне на нови възможности. Интересно е това, че пакетът `verbatim` включва командата `\verbatim@font`, позволяваща да се използва произволен шрифт, например, кирилица, което е невъзможно в средата `verbatim` без модификация на стандартното поведение на \LaTeX .

Освен това, в него е добавена възможност за отпечатване на листинги на програми, на различни езици за програмиране, с възможности да се поставят в рамка, да се отпечатват номерата на редовете, името на файла. Пакетът `verbatim` предоставя също командата

```
\verbatiminput{файл}
```

която позволява да се включва текстов файл в документа, така, както ако неговото съдържание би се намирало вътре в средата `verbatim`.

Тъй като пакетът `verbatim` е част от комплекта ‘tools’, вие трябва да го намерите инсталиран на повечето системи. Ако искате да знаете за този пакет повече, обезателно прочетете [11].

4.7 Изтегляне и инсталиране на \LaTeX -пакети

Повечето дистрибутиви на \LaTeX включват голям набор установени стилни пакети, но много повече са достъпни в мрежата. Основно място за търсене на стилни пакети в Интернет е CTAN (<http://www.ctan.org/>).

Пакети като `geometry`, `hyphenat` и много други, обикновено се състоят от два файла: файл с разширение `.ins` и друг – с разширение `.dtx`. Често към тях се прилага файл `readme.txt` с кратко описание на пакета.

Веднага след като копирате файловете от пакета на вашия компютър, трябва да ги обработите така, че (а) вашият \TeX да узнае за наличието на нов пакет и (б) да получите необходимата документация. Ето как се прави (а):

1. Обработете с \LaTeX файла `.ins`. Като резултат ще получите файл `.sty`.
2. Преместете файла `.sty` там, където вашият дистрибутив търси тези файлове. Обикновено това са под-директории на каталога

`.../localtexmf/tex/latex`. (Потребителите на Windows и OS/2 трябва да попълнят пълната директория).

3. Обновете базата с имената на файловете във вашия дистрибутив. Командата зависи от използвания \LaTeX -дистрибутив: за teTeX , fpTeX – `texhash`, за web2c – `maktexlsr`, за MiKTeX – `initexmf-update-fndb` или с използване на графичния интерфейс (GUI).

Следващата стъпка е получаването на документацията от файла `.dtx`:

1. Обработете с \LaTeX файла `.dtx`. Като резултат ще получите файл `.dvi`. Може да се наложи да компилирате с \LaTeX няколко пъти, за да получите правилни препратки.
2. Проверете, дали \LaTeX е генерирал файл `.idx`, освен всички други получени файлове. Ако не видите такъв файл, преминете на точка 5.
3. За да генерирате Азбучния указател, напишете командата:

```
makeindex -s gind.ist име
```

(където *име* е името на основния файл без разширение).
4. Отново компилирайте с \LaTeX файла `.dtx`
5. Накрая, за удобство при четене, генерирайте файла `.ps` или `.pdf`.

В някои случаи може да се генерира и файл `.glo` (глосарий). В този случай между стъпките 4 и 5 изпълнете командата:

```
makeindex -s gglo.ist -o име.gls име.glo
```

и поне още веднъж компилирайте с \LaTeX файла `.dtx` преди преминаване на стъпка 5.

4.8 Работа с pdf \LaTeX

Създаването на електронни учебници (e-books) с \LaTeX е усъвършенствано с пакета `hyperref`. Като резултат се получава файл във формат **PDF** (*Portable document format*). Програмата `PDF \LaTeX` генерира PDF директно от изходния текст, написан с \LaTeX , по същия начин, както програмата \LaTeX генерира файл DVI. В получения PDF-файл, всички препратки към формули, заглавия или литература автоматично се преобазуват в хипер-връзки, което означава, че кликуване с мишката върху такава хипер-връзка автоматично ни изпраща в целта. Освен това, отделните единици на Съдържанието и Азбучния указател също стават елементи на хипер-връзки.

Тук се дават само основни команди; подробности и допълнителни екстри има в *The L^AT_EX Web Companion*, [5], както и в документацията.

PDF е формат на хипертекстови документи. Така, както и на web страниците, някои думи в документа се означават като хипервръзки. Те водят към други точки в документа или дори към други документи. Ако кликнете с мишката върху такава хипервръзка, ще попаднете в точката-цел на връзката. В контекста на L^AT_EX това означава, че всички срещнати в текста команди `\ref` и `\pageref` стават хипервръзки. Освен това, Съдържанието, Азбучният указател и други аналогични структури стават набор от елементи на хипервръзки.

Голяма част от web-страниците днес се пишат на езика HTML (*HyperText Markup Language*). За научни документи този формат има два сериозни недостъка:

1. Включването в HTML на математически формули, най-общо казано, не се поддържа. Независимо от наличието на такъв стандарт, повечето използвани днес Интернет - навигатори не го поддържат или нямат необходимите шрифтове.
2. Отпечатване на HTML документи е възможен, но резултатът силно зависи от използваната платформа и навигатора. Резултатът изобщо не е с това качество, което сме свикнали да очакваме в света на L^AT_EX.

Има много опити да се създадат транслатори от L^AT_EX в HTML. Някои от тях дори са твърде успешни, в този смисъл, че те могат да генерират коректни web - страници от обикновени файлове на L^AT_EX. Но всички те правят множество опростявания, за да получат резултата. Веднага щом започнете да използвате сложни възможности на L^AT_EX с включване на допълнителни пакети, нещата започват да се разпадат и транслаторът дава грешки. Авторите, желаещи да съхранят уникалното типографско качество на своите документи, публикувани дори на WWW, предпочитат PDF (*Portable Document Format*), който съхранява вида на документа и позволява да се използва хипертекстова навигация. Повечето съвременни web - навигатори имат вградени средства за изобразяване на документи във формат PDF.

Въпреки, че има средства за преглеждане на формат DVI и POSTSCRIPT почти на всяка платформа, много по-разпространени са Acrobat Reader и xpdf за преглеждане на PDF-документи. Затова, предоставяйки PDF-версии на вашите документи, вие ги правите много по-достъпни за потенциалните читатели.

4.8.1 PDF-документи за WWW

Създаването на PDF-файл от изходния \TeX -файл, е много просто, благодарение на програмата pdf \TeX , разработена от Hàn Thê Thành. pdf \TeX генерира PDF по аналогичен начин, както \TeX генерира DVI. Съществува също pdf \LaTeX , генериращ PDF от изходния \LaTeX -текст.

И двете програми, pdf \TeX и pdf \LaTeX , се инсталират автоматично при повечето съвременни дистрибуции на \TeX , като te \TeX , fp \TeX , Mik \TeX , \TeX Live и CMac \TeX .

За генериране на PDF вместо DVI е достатъчно вместо командата `latex file.tex` да се използва командата `pdflatex file.tex`.

В \LaTeX можете да определите размера на хартията с добавяне на незадължителен параметър в командата `\documentclass`, например, `a4paper` или `letterpaper`. Този механизъм работи и в pdf \LaTeX , но освен това, pdf \TeX се нуждае и от физическия размер на хартията, за да определи физическия размер на страницата в pdf-файла.

Ако използвате пакета `hyperref` (вж. стр. 85), размерът на хартията ще бъде установен автоматично. В други случаи трябва да направите това ръчно, като поставите в преамбула на документа следните редове:

```
\pdfpagewidth=\paperwidth
\pdfpageheight=\paperheight
```

Следващите раздели по-детайлно разглеждат разликата между «нормалния» \LaTeX и pdf \LaTeX . Основните разлики засягат три области: използваните шрифтове, форматите на включваните изображения и ръчното оформяне на хипервръзки.

4.8.2 Шрифтове

pdf \LaTeX може да се справя с всякакви видове шрифтове (PK bitmaps, TrueType, POSTSCRIPT type 1...), но основния за \LaTeX шрифтов формат, PK bitmaps, дава много лош резултат, когато документът се изобразява на екрана с Acrobat Reader. Най-добре е да се използват шрифтове POSTSCRIPT Type 1, особено за генериране на добре изглеждащи документи. Модерните \TeX дистрибуции се установяват така, че това става автоматично. Най-добре е да опитате. Ако това е така, можете да прескочите този раздел.

POSTSCRIPT Type 1 версии на шрифтовете Computer Modern и AMSFonts са произведени от компаниите Blue Sky Research и Y&Y, Inc., които след това са предали авторските права върху тях на Американското Математическо Общество (AMS). В началото на 1997 година тези шрифтове са били направени публично достъпни, а в днешно време са включени в повечето дистрибутиви на \TeX .

Обаче, ако използвате \LaTeX за създаване на документи на езици, различни от английски, може би ще искате да използвате шрифтове EC, LH или CB (вж. дискусията за шрифтовете OT1 на стр. 30).

Владимир Волович е създал шрифтовия пакет `cm-super`, който обхваща пълните набори шрифтове EC/TC, EC Concrete, EC Bright и LH. Той е достъпен на адрес CTAN:/tex-archive/fonts/ps-type1/cm-super, а освен това е включен в дистрибутивите `TeXLive7` и `MikTeX`.

Аналогично, гръцките шрифтове CB във формат Type 1, създадени от Apostolos Syropoulos, са достъпни на адрес CTAN:/tex-archive/fonts/greek/cb.

Друго решение е да се превключи на други POSTSCRIPT Type 1 шрифтове. Въсъщност, някои от тях дори се включват във всяко копие на Acrobat Reader. Тъй като тези шрифтове имат друга метрика на символите, отпечатването на текста на страницата ще се измени. Обикновено текста започва да заема повече място, тъй като CM шрифтовете са много по-компактни. Освен това, общата визуална съгласуваност на документа ще се влоши, тъй като шрифтовете Times, Helvetica и Courier (най-подходящи за такава замяна) не са били проектирани да работят в хармония в един документ.

За тази цел има два готови набора шрифтове: пакета `pxfonts`, който е базиран основно на *Palatino* като основен шрифт в текста, и най-добре – пакета `txfonts`, базиращ се на шрифта *Times*. За да използвате тези пакети, е достатъчно да поставите в преамбюла следните редове:

```
\usepackage[T1]{fontenc}
\usepackage{pxfonts}
```

Забележка: След компилация на изходния текст, може да видите в `.log`-файла редове, наподобяващи следния:

```
Warning: pdftex (file eurmo10): Font eurmo10 at ... not found
```

Те означават, че не е бил намерен някакъв шрифт, използван в документа. Вие трябва да отстраните този проблем, тъй като полученият PDF-документ може *изобщо да не показва страниците с отсъстващите символи*.

4.8.3 Използване на графики

Включването на графики в документ работи най-добре с пакета `graphicx` (вж. стр. 73). С използване на стойност `pdftex` в опцията `driver`, този пакет успешно работи с `pdf \LaTeX` :

```
\usepackage[pdftex]{color,graphicx}
```

В горния пример е включен също пакета `color`, тъй като използването на цвят в документи, изобразявани в мрежата, е нормално явление.

За съжаление, има един проблем при работа с графики и PDF. Графики във формат Encapsulated POSTSCRIPT не работят с pdfL^AT_EX. Ако не зададете разширение на файла в командата `\includegraphics`, пакетът `graphicx` избира подходящия файл, основавайки се на стойността на опцията *driver*. За `pdftex` това са формати `.png`, `.pdf`, `.jpg`, `.mps` (METAPOST) и `.tif`, но *не* и `.eps`.

Прост начин за заобикаляне на този проблем е да се конвертират вашите EPS-файлове във формат PDF с инструмента `epstopdf`, присъстващ в много системи.

1. Намерете файла `epstopdf.exe` и го копирайте в директорията, в която се намират вашите eps-файлове с картинки.
2. В командния прозорец дайте командата

```
epstopdf filename.eps
```

3. Включването на картинката в pdf-документа става с командата

```
\includegraphics{filename}
```

Забележете, че в горната команда, файлът се използва без разширение. Това позволява да се обработва един и същ файл както с L^AT_EX, така и с pdfL^AT_EX. Всяка програма търси файл по подразбиране: за L^AT_EX това е `.eps`, а за pdfL^AT_EX – `.pdf`.

4.8.4 Пакет `hyperref`

Именно пакетът `hyperref` реализира поддръжката на хипер-връзки и полета със заглавия, които могат да се използват при преглеждане на PDF-документа в *Acrobat Reader*. Пакетът автоматично превръща всички вътрешни препратки в документа в текстови хипер-връзки. Освен това, елементи на хипер-връзки стават и всички единици от съдържанието на материала, номерата на страниците в азбучния указател, цитираните литературни единици, списъците на фигури и таблици, както и препратките към web-сайтове.

За да работи това както трябва, е необходимо командата

```
\usepackage[pdftex]{hyperref}
```

да бъде *последна* команда в преамбула на документа.

L^AT_EX създава препратки в документа по следния начин. Командата `\label{name}` поставя етикет с име *name*. Под това име в допълнителен aux-файл се запомня стойността на брояча на текущия номериран обект (раздел, уравнение, таблица) и номера на страницата, на която попада

съответната команда `\label`. Командите `\ref{name}` и `\pageref{name}` печатат съответно номера на обекта, вътре в който стои етикетът `name`, и номера на страницата с етикет `name`.

Пакетът `hyperref` преопределя горните команди на \LaTeX по такъв начин, че номерата на обектите и страниците, които печатат командите `\ref` и `\pageref`, стават хипервръзки, а етикетите, които поставя командата `\label`, стават мишени (`targets`), към които се прави скок при активиране на препратките. Хипер-препратки към друг документ се правят след като се зареди пакетът `xr-hyper` преди пакета `hyperref`. В резултат на това, описаните по-горе команди създават мишена в другия документ чрез командата `\label`.

Имената на `aux`-файловете на всички такива документи се обявяват чрез декларацията

```
\externaldocument[prefix]{file}[URL]
```

в преамбула на текущия входен файл, където `file` е името на файла без разширение, `prefix` позволява да се изключи съвпадението на етикети в различните документи, а `URL` указва адреса на готовия документ.

Друг начин за създаване на хипервръзки в документи, е чрез командата

```
\hypertarget{name}{text}
```

която създава мишена `name` за хипервръзка. Аргументът `text` или не се въвежда, или е обикновен текст.

В случай, че мишената е в същия документ, то посредством командата

```
\hyperlink{name}{text}
```

се създава хипервръзка към мишената `name`, където `text` се оформя като текст на хипервръзката. Ако мишената е в друг документ, чрез командата

```
\href{URL}{text}
```

се създава хипервръзка, като `URL` задава адреса на ресурса, включително името на мишената `name`, а `text` се изобразява като текст на хипервръзката.

Командата `\href{URL}{text}` създава хипер-връзка към документ от произволен тип, съхраняван в мрежата на адрес `URL`. Аргументът `text` се оформя като текст на хипер-връзката. Освен това може да се

използва и по-кратката команда `\href{URL}`, като *URL* се оформя като текст на хипер-връзката.

За допълнителна настройка на поведението на пакета `hyperref` могат да се използват:

- или списък от разделени със запетая опции на пакета `hyperref`, след опцията `pdftex`
`\usepackage[pdftex]{hyperref}`
- отделни редове с командата `\hypersetup{опции}`.

Единствена задължителна опция е `pdftex`, останалите само позволяват да се изменя поведението на `hyperref` по подразбиране. В следващия списък, стойностите по подразбиране са в прав шрифт:

`bookmarks (=true, false)` Показва или пречи на полето със заглавия при изобразяване на документа;

`unicode (=false, true)` Позволява да се използват нелатински символи в заглавията на Acrobat;

`pdftoolbar (=true, false)` Показва или пречи на линията с инструментите на Acrobat;

`pdfmenubar (=true, false)` Показва или пречи на менюто на Acrobat;

`pdfwindow (=true, false)` Изменя началното увеличаване на документа (размера на страницата в «прозореца» на Acrobat);

`pdftitle (=text)` Дефинира заглавие, което се изобразява от Acrobat в прозореца `Document Info`;

`pdfauthor (=text)` Името на автора на PDF-документа;

`pdfnewwindow (=true, false)` Определя дали да се отваря нов прозорец, ако хипер-връзката води извън текущия документ;

`colorlinks (=false, true)` Огражда връзките с цветни рамки (`false`) или изменя цвета на хипер-връзките (`true`). Цветът може да се настройва със следните опции (указаните стойности са по подразбиране):

`linkcolor (=red)` Цвят на вътрешните хипер-връзки (раздели, страници и др.);

`citecolor (=green)` Цвят на връзки към Литературата;

`filecolor (=magenta)` Цвят на връзки към файлове;

`urlcolor (=cyan)` Цвят на връзки към URL (електронна поща, WWW).

Ако ви устройват стойностите по подразбиране, използвайте:

```
\usepackage[pdftex]{hyperref}
```

За да се отвори полето на заглавията и да се оцветят хипер-връзките (стойностите `=true` могат да се изпускат):

```
\usepackage[pdftex,bookmarks,colorlinks]{hyperref}
```

Когато PDF-документът е предназначен за печат, по-добре е да не се използват цветни препратки, защото те ще се окажат на хартията сиви, което затруднява четенето. По-добре е да се използват цветни рамки, които не се печатат:

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

или да се направят препратките черни:

```
\usepackage{hyperref}
\hypersetup{colorlinks,%
            citecolor=black,%
            filecolor=black,%
            linkcolor=black,%
            urlcolor=black,%
            pdftex}
```

Ако искате да дадете информация за частта Document Info на PDF-файла:

```
\usepackage[pdftauthor={Ivan Petrov}%
            pdftitle={Introduction...},%
            pdftex]{hyperref}
```

В допълнение към автоматично генерираните хипер-връзки, е възможно да се правят връзки в явен вид с командата

```
\href{url}{text}
```

Кодът

```
Web-страницата \href{http://www.ctan.org}{CTAN}.
```

генерира на изхода “СТАН”; при кликване с мишката върху думата “СТАН”, попадете на web-страницата СТАН.

Ако препратката води не на URL, а на локален файл, можете да използвате командата `\href`:

Пълният текст на документа се намира `\href{guide.pdf}{тук}`.

което генерира текст “Пълният текст на документа се намира [тук](#).” Кликването върху думата “[тук](#)” отваря файла `guide.pdf`. (Името на файла се разглежда относно положението на текущия документ.)

Авторът на статията може да облекчи читателя при изпращане на писма с мнения, използвайки командата `\href` вътре в командата `\author` на заглавната страница на документа:

```
\author{Stefka Karakoleva %<\href{mailto:skarakoleva@ru.acad.bg}%
      {skarakoleva@ru.acad.bg}>%}
```

По такъв начин, препратката с пощенски адрес допълва адреса, приведен на самата страница. Това е направено така, защото препратката `\href{mailto:skarakoleva@ru.acad.bg}{Stefka Karakoleva}` е удобна в Acrobat Reader, но ще бъде невидима след печат на документа на хартия.

4.8.5 Проблеми с хипер-заглавията

Текстът в хипер-заглавията не винаги изглежда така, както на вас би ви се искало. В хипер-заглавията са допустими доста по-малък набор символи, отколкото в нормалния L^AT_EX, тъй като те са “просто текст”. Обикновено `hyperref` забелязва свързаните с това проблеми и извежда предупреждение:

```
Package hyperref Warning: Token not allowed in a
PDFDocEncoded string:
```

Вие можете да се справите с този проблем, предоставяйки текстов вариант за хипер-заглавието вместо проблемния текст:

```
\texorpdfstring{текст на TEX}{текст на заглавието}
```

Математическите изрази са първите кандидати за такава замяна:

```
\section{\texorpdfstring{ $E=mc^2$ }%
      {E\ =\ mc\textttwosuperior}}
```

което превръща `\section{ $E=mc^2$ }` в “E=mc²” за извеждане в заглавието.

Смяната на цвета също върви лошо в заглавията:

```
\section{\textcolor{red}{Red !}}
```

извежда “redRed!”. Командата `\textcolor` се игнорира, но нейният аргумент (`red`) се печата.

По-добър резултат се получава така:

```
\section{\texorpdfstring{\textcolor{red}{Red !}}{Red\ !}}
```

4.9 Съвместимост на изходните текстове в \LaTeX и $\text{pdf}\LaTeX$

В идеалния случай вашият документ трябва еднакво добре да се компилира както с \LaTeX , така и с $\text{pdf}\LaTeX$. Основният проблем тук е включването на графики. Просто решение е *системното премълчаване* на разширенията на файловете в командата `\includegraphics`. В този случай компилаторът автоматично избира файл с изисквания формат от текущия каталог. Всичко, което трябва да се направи е да се създадат подходящи версии на графичните файлове. \LaTeX ще избира `.eps`, а $\text{pdf}\LaTeX$ ще се постарее да включи файлове с разширения `.png`, `.pdf`, `.jpg`, `.mps` или `.tif` (в указания ред).

За случаите, когато ще ви е необходимо да използвате различни кодове за обикновената и PDF-версията на документа, може някъде в самото начало на документа да направите следното:

```
\newif\ifPDF
\ifx\pdfoutput\undefined\PDFfalse
\else\ifnum\pdfoutput >0\PDFtrue
  \else\PDFfalse
  \fi
\fi
```

Така се определя специална команда, позволяваща лесно да се пише условен код:

```
\ifPDF
  \usepackage[T2A]{fontenc}
  \usepackage[pdftex]{graphicx,color}
  \usepackage[pdftex]{hyperref}
\else
  \usepackage[T2A]{fontenc}
  \usepackage[dvips]{graphicx}
  \usepackage[dvips]{hyperref}
\fi
```

В горния пример пакетът `hyperref` е включен дори и в не-PDF-версията. Това е направено, за да работи командата `\href` във всеки случай и нейното използване да не се отхвърля в условните оператори.

Да отбележим, че в съвременните дистрибутиви на \TeX (например, в \TeX Live), изборът на `pdftex` или `dvips` при извикване на `graphicx` и `color` става автоматично в съответствие с настройките на конфигурационните файлове `graphics.cfg` и `color.cfg`.

4.10 Създаване на презентации с помощта на pdfscreen

Съществуват много презентационни пакети, които могат да се изтеглят от CTAN: `beamer`, `foillhtml`, `foiltex`, `ha-prosper`, `ifmslide`, `pdfscreen`, `pdfslide`, `prosper`, `seminar`, `slidenotes`, `slides`, `uwmslide`. $\text{pdf}\LaTeX$ в комбинация с пакета `pdfscreen`, позволява да създавате презентации във формат PDF, толкова красиви и живи, както и в *PowerPoint*, но много по-мобилни, тъй като Acrobat Reader съществува на много повече системи.

Класът `pdfscreen` използва пакетите `graphicx`, `color` и `hyperref` с опции, настроени за екранна презентация. За създаване на такъв тип документ обикновено се използва класът `article`. Фигура 4.2 показва примерен входен файл.

Пакетът `pdfscreen` има следните опции:

`screen` за екранна презентация. За хартиена версия използвайте опцията `print`.

`panelright` разполага навигационния панел в дясната част на екрана.

Ако панелът трябва да е вляво, използвайте опцията `panelleft`.

Ако той не ви е нужен въобще, използвайте `nopanel`.

`french` или всеки друг поддържан език ще представи текста на навигационните бутони на съответния език. Тази опция е независима от установените опции на пакета `babel`. Ако вашият език не се поддържа от `pdfscreen`, вие все пак можете да локализирате бутоните на панела с помощта на файла `pdfscreen.cfg`, вж. например `pdfscreen.cfg.specimen` и Фигура 4.3.

`chocolate` Цветова схема на навигационния панел. Други възможности са `gray`, `orange`, `palegreen`, `bluelace` и `blue`, използван по подразбиране.

След това се настройва форматът на изобразяване. Тъй като презентацията винаги се представя на пълен екран, това може да се използва за настройка на размера на шрифта:

`\panelwidth` определя ширината на навигационния панел.

```
\documentclass[pdftex,12pt]{article}
%%% пакети %%
%\usepackage[latin1]{inputenc}
%\usepackage[english]{babel}
%\usepackage[T1]{fontenc}
\usepackage[T2A]{fontenc}
\usepackage[cp1251]{inputenc}
\usepackage[english,bulgarian]{babel}
%%% pdfscreen %%
\usepackage[screen,panelleft,chocolate]{pdfscreen}
% Screen Format
\panelwidth=25mm
%%          height width
\screensize{150mm}{200mm}
%%          left right top bottom
\marginsize{42mm}{8mm}{10mm}{10mm}
% Color or image for background
\overlayempty
\definecolor{mybg}{rgb}{1,0.9,0.7}
\backgroundcolor{mybg}
% Logo
\emblema{MyLogo}
%%% For PPower4 (post-processor) %%
\usepackage{pause}
\begin{document}
\begin{slide}
\begin{itemize}
\item Добри новини\dotsc \pause \item Лоши новини
\end{itemize}
\end{slide}
\end{document}
```

Фигура 4.2: Пример за входен файл на pdfscreen


```

%% pdfscreen.cfg
%% настройка за Български език
\paneltitlepagename{\cyr Заглавие}
\panlabstractname{Резюме}
\panelcontentsname{Съдържание}
\panelfullscreenname{Екран}
\panelhomepagename{У дома}
\panelgobackname{Назад}
\panelclosename{Затвори}
\panelquitname{Край}
\panelpagename{Стр.}
\panelofname{от}
%% ----- various colors -----
\definecolor{buttonbackground}{rgb}{.902,.902,.980}
\definecolor{panelbackground}{gray}{.8}
\definecolor{buttonshadow}{gray}{.2}
\definecolor{section0}{rgb}{0,.5,.1}
\definecolor{section1}{rgb}{0,.5,1}
\definecolor{section2}{rgb}{0,.5,.5}
\definecolor{section3}{rgb}{0,.5,.4}
\definecolor{section4}{rgb}{.4,.5,.2}
\definecolor{section5}{rgb}{.5,.5,.3}
\emailid{skarakoleva@ecs.ru.acad.bg}
\urlid{www.ru.acad.bg}
%\IfFileExists{univ.jpg}{\emlema{univ.jpg}}{}
\def\@linkcolor{red}
\def\@anchorcolor{black}
\def\@citecolor{blue}
\def\@filecolor{cyan}
\def\@urlcolor{magenta}
\def\@menucolor{red}
\def\@pagecolor{red}
\def\FontTitle{%
\usefont{T2A}{fad}{m}{n}\fontsize{18pt}{16pt}\selectfont}
\def\@secCNTformat#1{\llap{\scshape\color{section\thesection@level}
\csname the#1\endcsname.\hspace*{6pt}}}
\def\pdfscreen{{\tt pdfscreen.sty}\xspace}
\pagedissolve{Wipe /D 1}
\endinput
%% end of file 'pdfscreen.cfg'

```

Фигура 4.3: Примерен файл за настройка на пакета pdfscreen за български език

`\screensize{ширина}{височина}` определя ширината и височината на екрана, включително навигационния панел.

`\marginsize{ляво}{дясно}{горе}{долу}` определя полетата на документа. В дадения пример документът не се центрира, затова номерата на разделите остават в лявото поле.

Може да се използва фоново изображение във всеки от поддържаните от pdfTeX формати с използване на командата

```
\overlay{изображение}
```

или, ако предпочитате обикновен фон, може да зададете неговия цвят с командата

```
\background{цвет}
```

Ако искате да поставите в навигационния панел логото на вашата организация, използвайте командата

```
\emblema{logo}
```

За контролиране на това, какво да се изобразява на всеки отделен слайд, използвайте средата `\begin{slide} ... \end{slide}`. Съдържанието на всеки слайд се изобразява на своята страницата вертикално центрирано.

Ако искате съдържанието да се изобразява направо в навигационния панел, може да използвате опция `paneltoc` при извикване на `pdfscreen`. Разбира се, трябва съдържанието да включва малко на брой кратки заглавия. За целта можете да задавате кратки заглавия в квадратни скоби.

Ако желаете отделните елементи на изложението да се представят последователно, може да използвате пакета `pause`. Той включва командата `\pause`, която се поставя на онези места, където искате Acrobat да задържи за момент изображението на документа. За да използвате този пакет трябва да инсталирате Java Virtual Mashine на вашия компютър. Пакетът `pause` е част от системата PPower4 (P^4 : *PDF Presentation Post-Processor*), която обработва документа, генериран с pdfTeX посредством командата:

```
ppower4 xy.pdf xyz.pdf
```

Това кратко описание на презентационния пакет `pdfscreen` е само повърхностно запознаване с неговите възможности. В дистрибутивите на LaTeX, както и в архива CTAN има подробно ръководство.

Глава 5

Генериране на математически графики

Съществуват няколко способа за създаване на собствена графика със средствата на базовия \LaTeX или негови разширения (пакети). Обикновено авторите предпочитат следния подход: графиката се създава чрез специализиран пакет и се включва в готов вид в документа. Базовият \LaTeX предлага възможност за генериране на графика по текстово описание (по координати). Този подход от една страна спестява памет, документите са компактни, а от друга страна – графиките са точни (до части от милиметъра). В тази глава ще научите как да създавате собствена графика с \LaTeX или някои специализирани пакети.

5.1 Обзор

Средата `picture` позволява да се програмират картинки направо в средата на \LaTeX . Подробно описание се привежда в *\LaTeX Manual* [1]. От една страна, в тази среда има твърде строги ограничения, например, както наклоните на отсечки, така и радиусите на кръговете са тясно ограничени по отношение на възможни стойности. От друга страна, средата `picture` на $\LaTeX 2_{\epsilon}$ въвежда командата `\qbezier`, («q» означава «квадратичен»). Много от често използваните криви, като окръжности, елипси или дъги, могат да се апроксимират задоволително с квадратични криви на Безие, макар за това да се изискват известни математически усилия.

Въпреки, че програмирането на картинки направо в \LaTeX е строго ограничено и често твърде уморително, има основания да се прави това. Получените по такъв начин документи стават малки по обем и не е необходимо да се прилагат графични файлове към документа.

Пакети, като `epic` и `eepic` (описани, например в *The \LaTeX Companion* [3]) или `pstricks` помагат да се елиминират ограниченията,

които затрудняват потребителя в оригиналната среда `picture`, и силно разширяват графичните възможности на \LaTeX .

Докато първите два пакета просто подобряват средата `picture`, пакетът `pstricks` има собствена среда за рисуване, `pspicture`. Силата на `pstricks` произтича от това, че той разширява възможностите на `POSTSCRIPT`. Освен този пакет, съществуват и много други, написани за конкретни цели. Един тях е пакета `Xy-pic`, описан в тази глава. Много такива графични пакети са подробно описани в *The \LaTeX Graphics Companion* [4], а още повече информация за разнообразни графични пакети с различно предназначение има на www.ctan.org.

Вероятно най-мощният графичен инструмент, свързан с \LaTeX , е `METAPOST`, програма-близък на създадената от Donald E. Knuth програма `METAFONT`. `METAPOST` използва мощен, математически строг език, `METAFONT`. За разлика от `METAFONT`, генериращ растери, `METAPOST` генерира файлове `Encapsulated POSTSCRIPT`, които могат да се импортират в \LaTeX . За начално запознаване, вижте *Ръководство за потребителя на MetaPost* [16], или ръководството *Графика в \LaTeX 2 ϵ* [18].

Много подробно работата с графики (и шрифтове) в \LaTeX и `TeX` е описана в *TeX Unbound* [17].

5.2 Среда `picture`

5.2.1 Основни команди

Средата `picture` се създава с една от двете команди:

```
\begin{picture}(x,y)...\end{picture}
```

или

```
\begin{picture}(x,y)(x_0,y_0)...\end{picture}
```

Числата x , y , x_0 , y_0 се задават в размерност `\unitlength`, която може да се изменя във всеки момент (но не вътре в средата `picture`) с команди, като

```
\setlength{\unitlength}{1.2cm}
```

Стойността на `\unitlength` по премълчаване е `1pt`. Първата двойка числа, (x, y) , определя резервираното за картинката правоъгълно пространство в документа. Незадължителната втора двойка числа, (x_0, y_0) , задава произволни координати на долния ляв ъгъл на резервирания правоъгълник.

Повечето команди за рисуване имат една от следните две форми:

```
\put(x,y){обект}
```

или

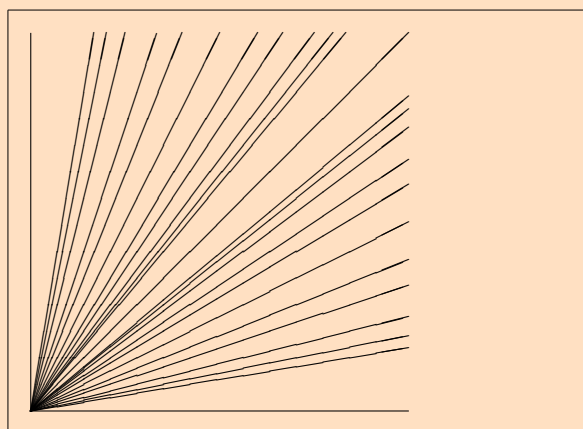
```
\multiput(x,y)(\Delta x,\Delta y){n}{обект}
```

Кривите на Безие са изключение от горното парвило. Те се рисуват с командата

```
\qbezier(x1,y1)(x2,y2)(x3,y3)
```

5.2.2 Отсечки

```
\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(0,1){1}}
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(1,1){1}}
  \put(0,0){\line(1,2){.5}}
  \put(0,0){\line(1,3){.3333}}
  \put(0,0){\line(1,4){.25}}
  \put(0,0){\line(1,5){.2}}
  \put(0,0){\line(1,6){.1667}}
  \put(0,0){\line(2,1){1}}
  \put(0,0){\line(2,3){.6667}}
  \put(0,0){\line(2,5){.4}}
  \put(0,0){\line(3,1){1}}
  \put(0,0){\line(3,2){1}}
  \put(0,0){\line(3,4){.75}}
  \put(0,0){\line(3,5){.6}}
  \put(0,0){\line(4,1){1}}
  \put(0,0){\line(4,3){1}}
  \put(0,0){\line(4,5){.8}}
  \put(0,0){\line(5,1){1}}
  \put(0,0){\line(5,2){1}}
  \put(0,0){\line(5,3){1}}
  \put(0,0){\line(5,4){1}}
  \put(0,0){\line(5,6){.8333}}
  \put(0,0){\line(6,1){1}}
  \put(0,0){\line(6,5){1}}
\end{picture}
```



Отсечки се рисуват с командата

```
\put(x,y){\line(x1,y1){дължина}}
```

Командата `\line` има два аргумента:

1. вектор на направление,
2. дължина.

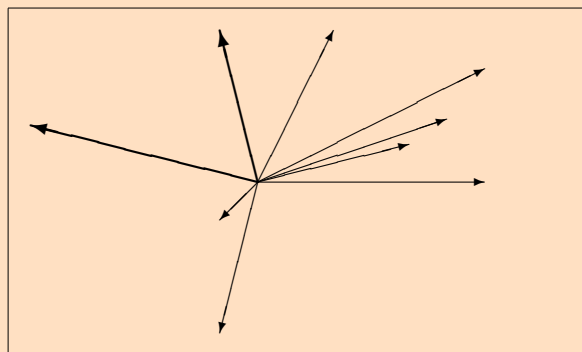
Компонентите на вектора на направление са ограничени; те могат да бъдат измежду целите числа

$$-6, -5 \dots, 5, 6,$$

и трябва да бъдат взаимно прости (да нямат общ делител, освен 1). Фигурата показва всичките 25 възможни стойности на наклона в първи квадрант. Дължината се изразява в единици `\unitlength`. Аргументът *дължина* е вертикалната координата в случай на вертикална отсечка и хоризонталната – във всички останали случаи.

5.2.3 Вектори

```
\setlength{\unitlength}{1mm}
\begin{picture}(60,40)
  \put(30,20){\vector(1,0){30}}
  \put(30,20){\vector(4,1){20}}
  \put(30,20){\vector(3,1){25}}
  \put(30,20){\vector(2,1){30}}
  \put(30,20){\vector(1,2){10}}
  \thicklines
  \put(30,20){\vector(-4,1){30}}
  \put(30,20){\vector(-1,4){5}}
  \thinlines
  \put(30,20){\vector(-1,-1){5}}
  \put(30,20){\vector(-1,-4){5}}
\end{picture}
```



Вектор се рисува с командата

```
\put(x,y){\vector(x1,y1){дължина}}
```

За векторите, стойностите на вектора на направление са още по-ограничени, отколкото за отсечките, а именно – те могат да бъдат измежду целите числа

$$-4, -3, \dots, 3, 4.$$

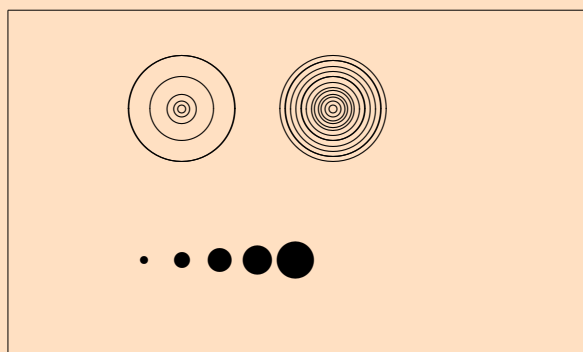
Компонентите на вектора също трябва да бъдат взаимно прости (да нямат общ делител, освен 1). Забележете ефекта на командата `\thicklines` на двата вектора, сочещи наляво.

5.2.4 Окръжности

```

\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
  \put(20,30){\circle{1}}
  \put(20,30){\circle{2}}
  \put(20,30){\circle{4}}
  \put(20,30){\circle{8}}
  \put(20,30){\circle{16}}
  \put(20,30){\circle{32}}
  \put(40,30){\circle{1}}
  \put(40,30){\circle{2}}
  \put(40,30){\circle{3}}
  \put(40,30){\circle{4}}
  \put(40,30){\circle{5}}
  \put(40,30){\circle{6}}
  \put(40,30){\circle{7}}
  \put(40,30){\circle{8}}
  \put(40,30){\circle{9}}
  \put(40,30){\circle{10}}
  \put(40,30){\circle{11}}
  \put(40,30){\circle{12}}
  \put(40,30){\circle{13}}
  \put(40,30){\circle{14}}
  \put(15,10){\circle*{1}}
  \put(20,10){\circle*{2}}
  \put(25,10){\circle*{3}}
  \put(30,10){\circle*{4}}
  \put(35,10){\circle*{5}}
\end{picture}

```



Командата

```
\put( $x, y$ ){\circle{диаметър}}
```

рисува окръжност с център в точката (x, y) и диаметър (не радиус!) *диаметър*. Средата `picture` допуска диаметри до около 14 мм, и дори в тези граници не са допустими всички диаметри. Командата `\circle*` рисува кръгове (запълнени окръжности).

Както и в случая на отсечки, можете да се обръщате към допълнителни пакети, такива, като `eeepic` или `pstricks`. Подробно описание на тези пакети е дадено в *The L^AT_EX Graphics Companion* [4].

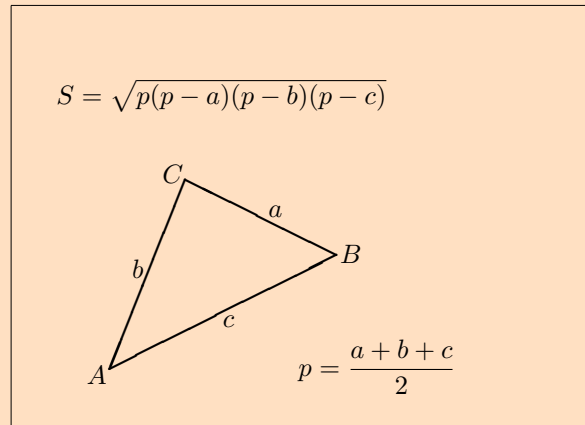
Съществува също и изход в рамките на средата `picture`. Ако не се страхувате да направите необходимите изчисления (или ги възлагате на програма), то можете да изобразявате произволни окръжности и елипси с помощта на криви на Безие. Примери и изходни текстове на Java са дадени в *Графика в L^AT_EX 2_ε* [18].

5.2.5 Текст и формули

```

\setlength{\unitlength}{1cm}
\begin{picture}(6,5)
  \thicklines
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.7,0.3){\text{\$A\$}}
  \put(4.05,1.9){\text{\$B\$}}
  \put(1.7,2.95){\text{\$C\$}}
  \put(3.1,2.5){\text{\$a\$}}
  \put(1.3,1.7){\text{\$b\$}}
  \put(2.5,1.05){\text{\$c\$}}
  \put(0.3,4){\text{\$S=}
    \sqrt{p(p-a)(p-b)(p-c)}\$}
  \put(3.5,0.4){\text{\$\\displaystyle
    p=\frac{a+b+c}{2}\$}}
\end{picture}

```



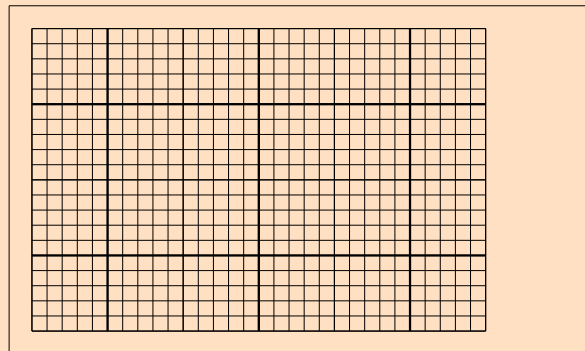
Както показва горния пример, текст и формули могат да се разполагат в средата `picture` по стандартния начин — с командата `\put`.

5.2.6 Команди `\multiput` и `\linethickness`

```

\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){31}%
    {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
    {\line(1,0){30}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){7}%
    {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
    {\line(1,0){30}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){3}%
    {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
    {\line(1,0){30}}
\end{picture}

```



Командата

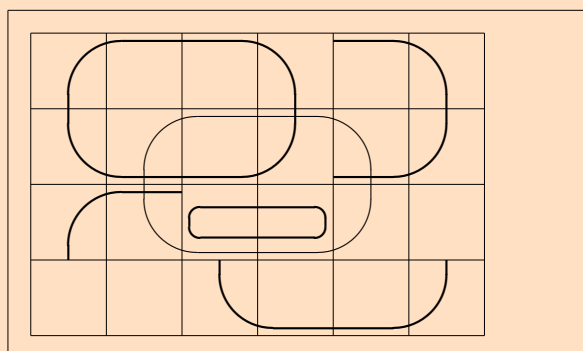
$$\text{\code{\multiput}(x,y)(\Delta x,\Delta y)\{n\}\{обект\}}$$

има 4 аргумента: начална точка (x, y) , вектор на прехода от един обект

към следващия $(\Delta x, \Delta y)$, брой обекти n и самия *обект* за рисуване. Командата `\linethickness` се прилага към хоризонтални и вертикални отсечки, но никога – към наклонени отсечки или окръжности. Тя може да се прилага и към квадратични криви на Безие.

5.2.7 Овали. Команди `\thinlines` и `\thicklines`

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}%
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}%
    {\line(1,0){6}}
  \thicklines
  \put(2,3){\oval(3,1.8)}
  \thinlines
  \put(3,2){\oval(3,1.8)}
  \thicklines
  \put(2,1){\oval(3,1.8)[t]}
  \put(4,1){\oval(3,1.8)[b]}
  \put(4,3){\oval(3,1.8)[r]}
  \put(3,1.5){\oval(1.8,0.4)}
\end{picture}
```



Командата

```
\put(x,y){\oval(w,h)}
```

или

```
\put(x,y){\oval(w,h)[позиция]}
```

произвежда овал с център в (x, y) , имащ дължина w и височина h . Незадължителният аргумент *позиция* може да приема стойности **b** - долу (bottom), **t** - горе (top), **l** - ляво (left) и **r** - дясно (right), които могат да се комбинират, както е показано в дадения пример.

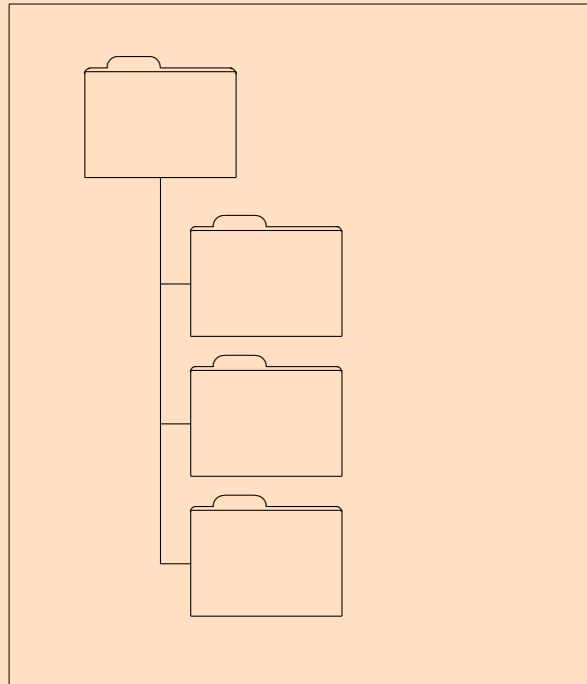
Дебелината на линията може да се контролира с два вида команди: `\linethickness{дължина}` от една страна, и `\thinlines` и `\thicklines` – от друга. Докато командата `\linethickness{дължина}` се прилага само към хоризонтални и вертикални линии (и квадратични криви на Безие), командите `\thinlines` и `\thicklines` се прилагат към наклонени отсечки, окръжности и овали.

5.2.8 Създаване и многократно използване на блокове с картинки

```

\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}% обявяване
\savebox{\foldera}
  (40,32)[bl]{% определяне
  \multiput(0,0)(0,28){2}
    {\line(1,0){40}}
  \multiput(0,0)(40,0){2}
    {\line(0,1){28}}
  \put(1,28){\oval(2,2)[t1]}
  \put(1,29){\line(1,0){5}}
  \put(9,29){\oval(6,6)[t1]}
  \put(9,32){\line(1,0){8}}
  \put(17,29){\oval(6,6)[tr]}
  \put(20,29){\line(1,0){19}}
  \put(39,28){\oval(2,2)[tr]}
}
\newsavebox{\folderb}% обявяване
\savebox{\folderb}
  (40,32)[l]{% определяне
  \put(0,14){\line(1,0){8}}
  \put(8,0){\usebox{\foldera}}
} \put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
  {\usebox{\folderb}}
\end{picture}

```



Блок с картинка може да бъде *обявен* с командата

```
\newsavebox{име}
```

а след това *дефиниран* с командата

```
\savebox{име}(ширина,височина)[позиция]{съдържание}
```

и накрая, многократно *нарисуван* с командата

```
\put(x,y)\usebox{име}
```

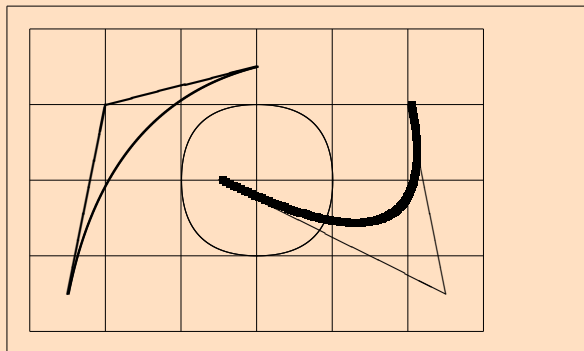
Незадължителният аргумент *позиция* определя точката на закрепване на блока. В дадения пример той е установен `bl`, което поставя точката на закрепване в долния ляв ъгъл на блока. Други варианти са стойностите `t` (горе) и `r` (дясно).

Аргументът *име* всъщност става команда на L^AT_EX (затова в разглеждания пример пред него има обратна наклонена черта). Блоковете могат да бъдат вложени. В примера, `\foldera` се използва вътре в дефиницията на `\folderb`.

Командата `\oval` тук е необходимо да се използва, защото командата `\line` не работи, ако дължината на отсечката е по-малка от 3 мм.

5.2.9 Квадратични криви на Безие

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}
    {\line(1,0){6}}
  \thicklines
  \put(0.5,0.5){\line(1,5){0.5}}
  \put(1,3){\line(4,1){2}}
  \qbezier(0.5,0.5)(1,3)(3,3.5)
  \thinlines
  \put(2.5,2){\line(2,-1){3}}
  \put(5.5,0.5){\line(-1,5){0.5}}
  \linethickness{1mm}
  \qbezier(2.5,2)(5.5,0.5)(5,3)
  \thinlines
  \qbezier(4,2)(4,3)(3,3)
  \qbezier(3,3)(2,3)(2,2)
  \qbezier(2,2)(2,1)(3,1)
  \qbezier(3,1)(4,1)(4,2)
\end{picture}
```



Както показва този пример, разбиването на окръжност на четири квадратични криви на Безие, не дава удовлетворителен резултат. Изискват се минимум осем. Фигурата отново показва влиянието на командата `\linethickness` на хоризонталните и вертикални линии, както и на командите `\thinlines` и `\thicklines` — на наклонените отсечки. Тя също показва, че и двете команди оказват влияние на квадратичните криви на Безие, както и че всяка следваща команда отменя предишните.

Нека $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ задават крайните точки, а m_1, m_2 — съответните наклони на допирателните към квадратичната крива на Безие. Тогава междинната управляваща точка $S = (x, y)$ се задава с уравнението

$$\begin{cases} x = \frac{m_2 x_2 - m_1 x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y = y_i + m_i(x - x_i) \quad (i = 1, 2). \end{cases} \quad (5.1)$$

В *Графика в L^AT_EX 2_ε* [18] е дадена Java-програма, която генерира необходимите `\qbezier` командни редове.

5.3 Xy-pic

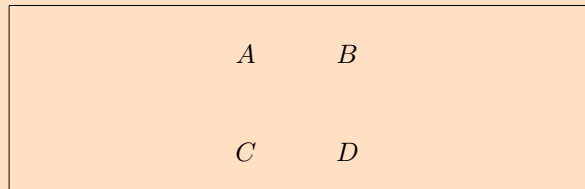
`xy` е специален пакет за рисуване на диаграми. За да го използвате, добавете към преамбюла на документа следния ред:

```
\usepackage[опции]{xy}
```

където *опции* е списък от функции на Xy-pic, които искате да заредите. Опцията `all`, инструктира L^AT_EX да зареди всички команди на Xy.

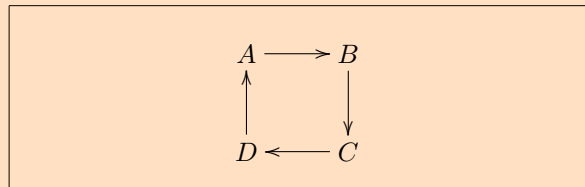
Диаграмите на Xy-pic се рисуват в матричен вид, като всеки елемент на диаграмата се поставя в определена клетка на матрица:

```
\begin{displaymath}
\xymatrix{A & B \\
          C & D }
\end{displaymath}
```



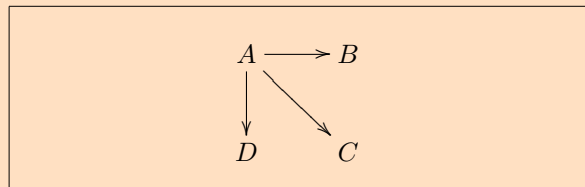
Командата `\xymatrix` трябва да се използва в математичен режим. Тук са зададени два реда и два стълба. За да се получи от тази матрица диаграма, се добавят стрелки-вектори с командата `\ar`.

```
\begin{displaymath}
\xymatrix{ A \ar[r] & B \ar[d] \\
          D \ar[u] & C \ar[l] }
\end{displaymath}
```



Командата за рисуване на вектора се поставя в клетката, от която той излиза. Аргумент е направлението, в което е насочен векторът: `u` (up), `d` (down), `r` (right) и `l` (left).

```
\begin{displaymath}
\xymatrix{
A \ar[d] \ar[dr] \ar[r] & B \\
D & C }
\end{displaymath}
```

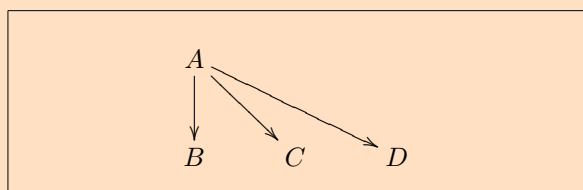


За рисуване на диагонали, укажете повече от едно направление. Можете също така да повторите знак за направление за рисуване на подълги вектори.

```

\begin{displaymath}
\xymatrix{
  A \ar[d] \ar[dr] \ar[dr] & & \\
  B & & C & D }
\end{displaymath}

```

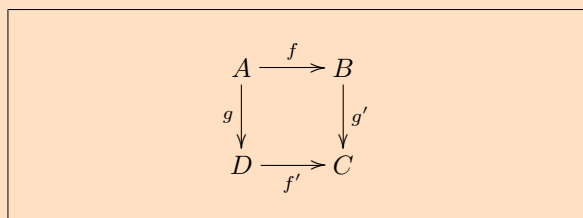


За поставяне на етикети до, над или под векторите, се използват обикновените оператори за долни и горни индекси.

```

\begin{displaymath}
\xymatrix{
  A \ar[r]^f \ar[d]_g & \\
  B \ar[d]^{g'} & \\
  D \ar[r]_{f'} & C }
\end{displaymath}

```

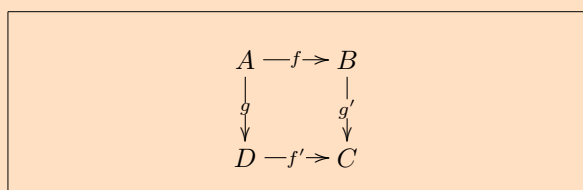


Както се вижда, тези оператори се използват по същия начин както в математичен режим. Единствената разлика е в това, че горният индекс означава «над стрелката на вектора», а долният – «под стрелката». Съществува и трети оператор, вертикална черта: |. Той поставя текста в стрелката.

```

\begin{displaymath}
\xymatrix{
  A \ar[r]|f \ar[d]|g & \\
  B \ar[d]|{g'} & \\
  D \ar[r]|{f'} & C }
\end{displaymath}

```



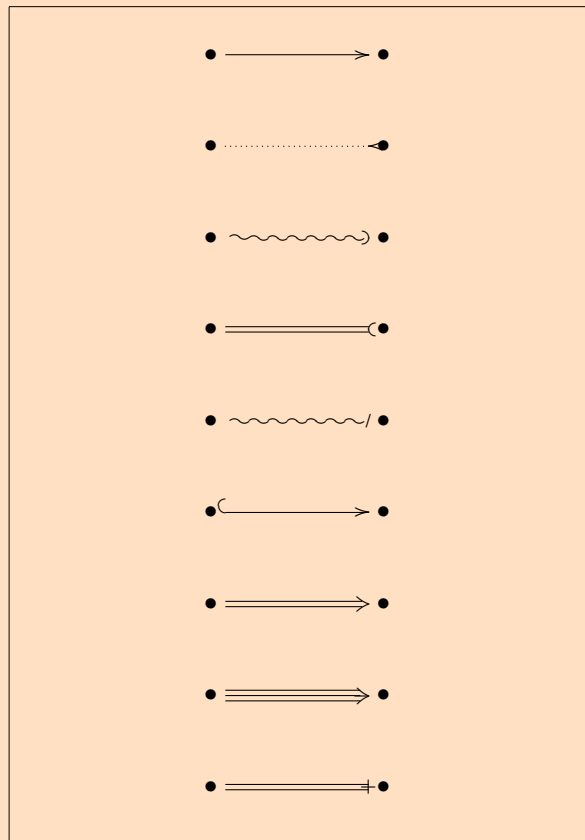
За да нарисувате стрелка с интервал в нея, използвайте командата `\ar[...]|hole`.

В някои случаи е необходимо да се използват различни по вид стрелки. Това може да се направи и чрез промяна на вида им:

```

\shorthandoff{"}
\begin{displaymath}
\xymatrix{
\bullet\ar@{->}[rr] && \bullet\\
\bullet\ar@{.<}[rr] && \bullet\\
\bullet\ar@{~}[rr] && \bullet\\
\bullet\ar@{=}[rr] && \bullet\\
\bullet\ar@{~/}[rr] && \bullet\\
\bullet\ar@{~{({})->}}[rr]&& \bullet\\
\bullet\ar@2{->}[rr] && \bullet\\
\bullet\ar@3{->}[rr] && \bullet\\
\bullet\ar@{=+}[rr] && \bullet
}
\end{displaymath}
\shorthandon{"}

```

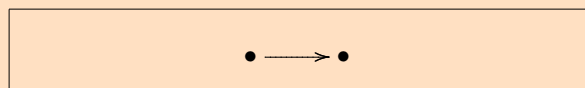


Забележете разликата между следните две диаграми. Модификаторите между наклонените черти определят, как да се рисуват кривите.

```

\begin{displaymath}
\xymatrix{
\bullet \ar[r]
\ar@{.>}[r] &
\bullet }
\end{displaymath}

```



```

\begin{displaymath}
\xymatrix{
\bullet \ar~/~/[r]
\ar@/_/@{.>}[r] &
\bullet }
\end{displaymath}

```



Xy-pic предлага много способи за въздействие върху начина на рисуване на кривите; подробности вижте в документацията на Xy-pic.

5.4 Създаване на POSTSCRIPT графики с пакета PSTricks

PSTricks е колекция от \TeX макроси, основани на POSTSCRIPT, които са съвместими с повечето \TeX макропакети, включително Plain \TeX , \LaTeX , \AMSTeX и $\AMS\LaTeX$. PSTricks ви дава цвят, графики, ротация, манипулация с текст, обработка на данни, графи, дървета и други специални възможности. PSTricks е включен във всеки съвременен дистрибутив на \LaTeX и на адрес CTAN:/tex-archive/graphics/pstricks

5.4.1 Основни понятия в пакета PSTricks

Пакетът PSTricks се зарежда в преамбюла на документа с командата

```
\usepackage{pstricks}
```

Повечето команди в пакета PSTricks рисуват определен обект в точка с координати, отчитани относно текущата точка. За рисунки обикновено се използва специалната \LaTeX среда

```
\begin{pspicture}*[settings] [baseline] (x0,y0) (x1,y1) ... \end{pspicture}
```

Първата двойка координати (x_0, y_0) не е задължителна, по подразбиране е $(0, 0)$. \TeX отделя място за правоъгълник с долен ляв ъгъл (x_0, y_0) и горен десен ъгъл (x_1, y_1) . Вариантът със * отсича онези части от графичните обекти, които излизат извън границите на посочената област.

Команди и аргументи

Почти всички команди в пакета PSTricks имат подобна (сложна) структура. Те изискват няколко или всички следващи аргументи, всеки от които има свои собствени ограничения. Задължителните аргументи са заградени с фигурни скоби $\{\arg\}$. Незадължителните аргументи са заградени с квадратни скоби $[\text{par1}=\text{val1}, \dots]$. Координатите се задават в кръгли скоби: (x, y) . Общият синтаксис на командите е:

```
\command*[settings]{arrows/parameters}(coordinates)
```

Вариант на команда със * означава, че изображеният обект е плътен, а не контурен. Незадължителната настройка (*settings*) в квадратни скоби, се състои от набор двойки от типа $[\text{par1}=\text{val1}, \dots]$. Задължителните аргументи могат да бъдат различни по форма стрелки и параметри за конкретния обект, като дебелина на линията, ъгъл на завъртане и други.

Задаване на графични параметри

PSTricks използва системата «*параметър=стойност*». Тези параметри могат да се задават по два начина:

- за всеки обект с помощта на незадължителни параметри от типа «*параметър=стойност*» в квадратни скоби, разделени със запетая. В този случай тяхното действие е локално за дадения обект.
- за някаква област в текущата среда с помощта на командата

```
\psset{par1=value1,par2=value2,...}
```

Графичните параметри, които могат да се установяват за обекти са дадени в Таблица 4.2 на стр.188 в книгата *The L^AT_EX Graphics Companion* [4]

Координати и единици за измерване

По подразбиране, единицата на измерване в PSTricks е 1 *cm*, но може да се променя чрез параметрите `unit`, `xunit`, `yunit` и др. Например:

```
\psset{\xunit=2mm,\yunit=1mm}
```

Цветове

Стандартният пакет `color` и `pstricks` могат да се заредят едновременно чрез зареждане на пакета `pstcol`:

```
\usepackage{pstcol}
```

Пакетът `pst-grad` добавя възможност за плавно изменение на цвета при оцветяване на обекта. Предопределени цветове са `black`, `darkgray`, `gray`, `lightgray`, `white`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow`.

Позициониране и въртене на обект

Командата

```
\rput*[settings][refpoint]{angle}(x0,y0){stuff}
```

позиционира всякакъв обект (включително и обикновен L^AT_EX материал) в точка (x_0, y_0) . Вариантът със * рисува около *stuff* рамка `\psframebox`. Ако е зададен ъгъл (*angle*), материалът (*stuff*) се завърта (в градуси). Аргументът *refpoint* описва точката на прикрепване на материала. По подразбиране това е центърът на бокса. Това може да се измени, ако

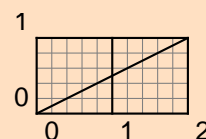
на *refpoint* се зададе една буква l-ляво, r-дясно, t-горе, b-долу, B-базова линия) или комбинация от две букви tl, tr, bl, br, Bl, Br.

5.4.2 Примери на използване на PSTricks

Ето няколко примера за това, как да се използва пакета PSTricks. Подробности и още примери има в *The L^AT_EX Graphics Companion* [4].

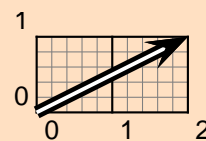
Да започнем с линия. Начална точка е (0, 0); стойностите на всички параметри са по премълчаване.

```
\usepackage{pstricks}
\begin{pspicture}(0,0)(2,1)\psgrid
  \psline(2,1)
\end{pspicture}
```



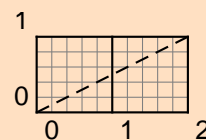
Командата `\psline` има незадължителен параметър, позволяващ завършване на линията със символ от типа на стрелка, а външният вид на линията се изменя с параметрите `linewidth`, `doubleline`, `doublesep`:

```
\usepackage{pstcol}
\begin{pspicture}(0,0)(2,1)\psgrid
  \psline[linewidth=0.6mm,
         doubleline=true,
         doublesep=0.5mm]{->}(2,1)
\end{pspicture}
```



Външният вид на линията може да се променя с параметъра `linestyle`:

```
\usepackage{pstcol}
\begin{pspicture}(0,0)(2,1)\psgrid
  \psline[linestyle=dashed](2,1)
\end{pspicture}
```

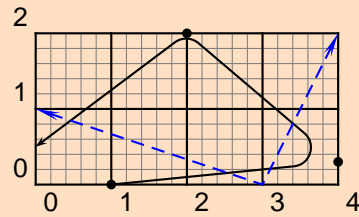


Както се вижда от следващия пример, командата `\psline` може да има много координати за аргументи. Едната от двете линии има променен цвят, дължина на стрелките и стила ѝ е пунктиран. При другата, за изменение на външния вид на траекторията около точките, се използва параметърът `linearc`. Параметърът `showpoints` указва на PSTricks да чертае точките, дори ако линията не минава през тях.

```

\usepackage{pstcol}
\begin{pspicture}(0,0)(4,2)\psgrid
\psline[linearc=0.25,showpoints=true]%
{->}(1,0)(4,0.3)(2,2)(0,0.5)
\psline[linestyle=dashed,
linecolor=blue,
arrowlength=3]{<->}(0,1)(3,0)(4,2)
\end{pspicture}

```

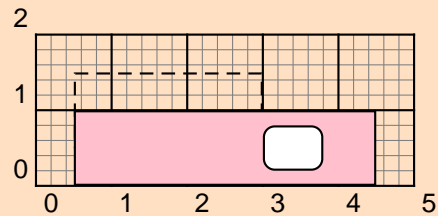


Най-лесният начин да се нарисова «кутийка», е като се използва командата `\psframe`.

```

\usepackage{pstcol}
\definecolor{pink}{rgb}{1,.75,.8}
\begin{pspicture}(0,0)(4.5,2)
\psgrid
\psframe[linestyle=dashed]%
(0.5,0.5)(3,1.5)
\psframe[fillstyle=solid,%
fillcolor=pink]%
(.5,0)(4.5,1)
\psframe[fillstyle=solid,%
fillcolor=white,%
framearc=0.5]%
(3,0.2)(3.8,0.8)
\end{pspicture}

```

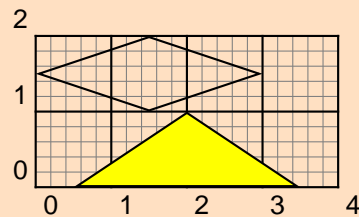


PSTricks рисува и други многоъгълници и тела, освен тези, показани на картинката:

```

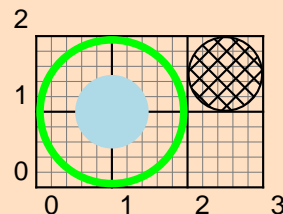
\usepackage{pstcol}
\begin{pspicture}(0,0)(4,2)
\psgrid
\psdiamond(1.5,1.5)(1.5,0.5)
\pstriangle[fillstyle=solid,%
fillcolor=yellow]%
(2,0)(3,1)
\end{pspicture}

```



Обектите могат да се запълват с цвят по различни начини:

```
\usepackage{pstcol}
\definecolor{lightblue}{rgb}{.68, .85, .9}
\begin{pspicture}(0,0)(3,2)\psgrid
  \pscircle[linewidth=1mm,linecolor=green]
    (1,1){1}
  \pscircle[linestyle=none,fillstyle=solid,
    fillcolor=lightblue](1,1){0.5}
  \pscircle[fillstyle=crosshatch](2.5,1.5)
    {0.5}
\end{pspicture}
```

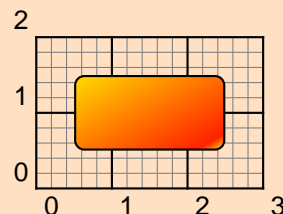


Зареждането на пакета `pst-grad` чрез командата

```
\usepackage{pstcol,pst-grad}
```

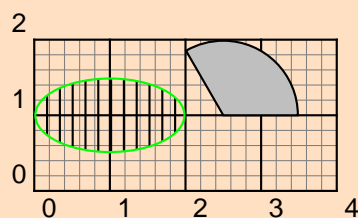
в преамбула на документа, разширява възможностите на PSTricks за плавно запълване с цвят:

```
\usepackage{pstcol,pst-grad}
\begin{pspicture}(0,0)(3,2)\psgrid
  \psframe[framearc=0.25,
    fillstyle=gradient,
    gradangle=30,
    gradbegin=yellow,
    gradend=red]
    (0.5,0.5)(2.5,1.5)
\end{pspicture}
```



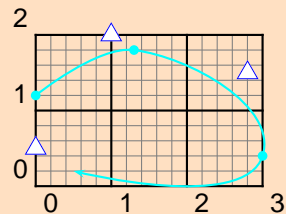
Ето и различни стилове на заштриховане под различни ъгли:

```
\usepackage{pstcol}
\begin{pspicture}(0,0)(4,2)\psgrid
  \psellipse[linecolor=green,
    fillstyle=vlines,
    hatchangle=0]
    (1,1)(1,0.5)
  \pswedge[fillstyle=solid,
    fillcolor=lightgray]
    (2.5,1){1}{0}{120}
\end{pspicture}
```



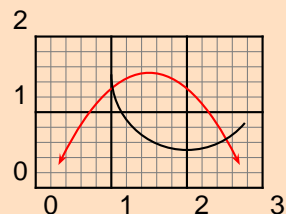
Командата `\psdots` поставя посочения символ в изброените точки:

```
\usepackage{pstcol}
\begin{pspicture}(0,0)(3,2)\psgrid
  \psdots[linecolor=blue,dotstyle=triangle,
    dotscale=2](0,0.5)(1,2)(2.8,1.5)
  \pscurve[linecolor=cyan,showpoints=true]
    {->}%
    (0,1.2)(1.3,1.8)(3,0.4)(0.5,0.2)
\end{pspicture}
```



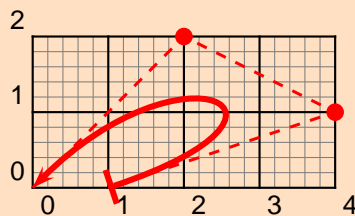
Командата `\parabola` рисува парабола от точка, с максимум в друга точка, а `\psarc` рисува част от окръжност със зададени център, радиус, начална и крайна стойност на ъгъла на завъртане:

```
\usepackage{pstcol}
\begin{pspicture}(0,0)(3,2)\psgrid
  \parabola[linecolor=red]{<->}%
    (0.3,0.3)(1.5,1.5)
  \psarc(2,1.5){1}{180}{320}
\end{pspicture}
```



Крива на Безие по четири контролни точки се рисува с командата `\psbezier`:

```
\usepackage{pstcol}
\begin{pspicture}(0,0)(4,2)\psgrid
  \psbezier[linewidth=0.8mm,
    linecolor=red,
    showpoints=true]{|->}%
    (1,0)(4,1)(2,2)(0,0)
\end{pspicture}
```



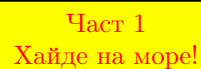
5.4.3 Свързване на текст и графика

Ето няколко примера за свързване на текст и графика със средствата на PSTricks:

```
\usepackage{pstricks}
\pscirclebox[doubleline=true]
  {\bfseries STOP!}
```



```
\usepackage{pstcol}
\psshadowbox[fillstyle=solid,
             fillcolor=yellow]
  {\color{red}
  \begin{tabular}{c}
    Част 1\\ Хайде на море!
  \end{tabular}}
```



```
\usepackage{pstcol}
\psdblframebox[linecolor=green,
               linestyle=dotted,linewidth=3pt,
               dotstyle=*]
  {\color{red} PSTricks е супер!}
```



```
\usepackage{pstricks}
\psdiabox{\sffamily Не паркирай!}
```



За изпълнение на следващия пример, освен пакета `pst-col`, в преамбюла на документа се зарежда и пакетът `pst-grad` чрез командата `\usepackage{pst-grad}`.

```
\usepackage{pstcol,pst-grad}
\pstribox[shadow=true,fillstyle=gradient,
          gradbegin=green,gradend=red]
{\color{white}\Huge
 $\frac{\alpha}{\Omega}$}
```



```
\usepackage{pstcol,pst-grad}
\psovalbox[linecolor=red]
  {\color{blue} \Large Важно!}
```



В следващия пример, освен пакета `pst-col`, в преамбюла на документа се зарежда и пакетът `pst-text` с командата `\usepackage{pst-text}`.

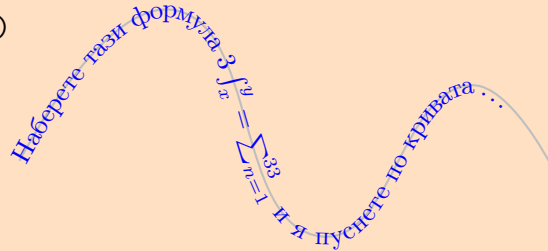
```

\usepackage{pstcol,pst-text}
\begin{pspicture}(-4,-3.2)(3,0.2)
\psset{linecolor=lightgray}
\pstextpath
  {\pscurve(-4,-2)(-2,0)%
   (0,-3)(2,-1)(3,-2)}
  {\color{blue}
Наберете тази формула

$$\int_3 \int_2 \int_1^{33} x^y = \sum_{n=1}^{33}$$

и я пуснете по кривата
\ldots}
\end{pspicture}

```

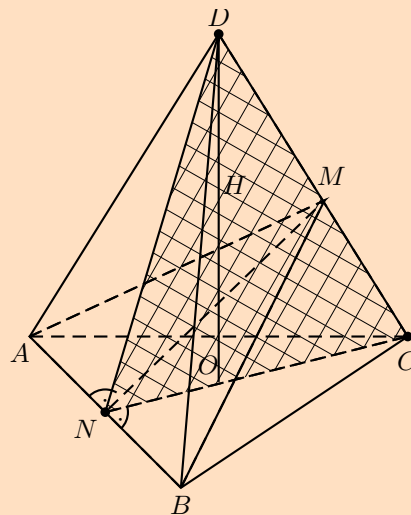


Ето пример и за чертане на пространствена фигура и сечение с равнина.

```

\usepackage{pstricks}
\psset{unit=1mm}
\begin{pspicture}(0,-3)(50,60)
\psline(0,20)(20,0)(50,20)%
  (25,60)(0,20)
\psline(20,0)(25,60)(25,14)
\qline(20,0)(39,38)
\qline(10,10)(25,60)
\rput[t](20,-1){$B$}
\rput[t](50,18){$C$}
\rput[tr](0,19){$A$}
\rput[b](25,61){$D$}
\rput[l](25.5,40){$H$}
\rput[br](25,15){$O$}
\rput[tr](9,9){$N$}
\rput[bl](38,40){$M$}
\psarc(10,10){3}{-45}{15}
\psarc(10,10){3}{60}{135}
\rput(10,11.5){$\cdot$}
\rput(12,9){$\cdot$}
\psset{linestyle=dashed}
\psline(0,20)(39,38)%
  (10,10)(50,20)
\qline(0,20)(50,20)
\psline[showpoints=true,%

```



Глава 6

Настройка на L^AT_EX

Документите, изработени с използване на разгледаните до сега команди, изглеждат чудесно, четат се леко и отговарят на типографските изисквания. Всички въведени до тук команди са напълно достатъчни за широката аудитория.

Има обаче ситуации, в които L^AT_EX не предоставя команда или среда, удовлетворяващи вашите нужди, или резултатът при печат не отговаря на вашите изисквания.

В тази глава ще разберете как да научите L^AT_EX на нови трикове, и как да направите неговото извеждане по-различно от това по подразбиране.

6.1 Нови команди, среди и пакети

Често при писане на големи документи (книги, дисертации) се налага да се извършват еднотипни манипулации с даден тип материал, например дадена дума-понятие да се пише с наклонен шрифт, да се загради в рамка и да се въведе в Азбучния Указател в края на книгата.

Вместо да се използват, за всеки отделен случай, съответните команди на L^AT_EX, може да се дефинира нова команда (или среда), която да извършва всички тези действия в документа. Дефинирането на нови команди и среди се извършва в преамбюла на документа, а се използват вътре в документа.

6.1.1 Нови команди

Създаването на собствена ваша L^AT_EX команда става в преамбюла на документа с командата:

```
\newcommand{име}[брой-арг]{определение}
```

Командата `\newcommand` има два задължителни аргумента: *име* на командата, която създавате и *определение* на командата. Аргументът

брой-арг в квадратните скоби е незадължителен и определя броя на аргументите, които изисква новата команда (допускат се до 9). Ако *брой-арг* липсва, по подразбиране е 0, т.е. не се допускат аргументи.

Следващият пример дефинира нова команда, наречена `\latcomp`. Това е съкращение на «The ЛАТ_EX Companion». Такава команда е удобна, ако се налага често да пишете името на тази книга.

```
%в преамбюла
\newcommand{\latcomp}{%
\emph{The \LaTeX{} Companion}}
% в тялото на документа
‘‘\latcomp’’ \dots{}

‘‘\latcomp’’
```

“*The ЛАТ_EX Companion*” ...
“*The ЛАТ_EX Companion*”

Команди с параметър

Следващите примери показват как да се дефинира команда с един или няколко аргументи. Етикетът `#1` се заменя със задания аргумент. Ако новата команда има два аргумента, използвайте `#1` и `#2`.

Командата `\rowvec` дефинира n - мерен вектор като ключова дума.

```
\documentclass{article}
\newcommand{\rowvec}{(x_1,x_2,\ldots,x_n)}
\begin{document}
Даден е вектор  $\rowvec$ .
\end{document}
```

Ако искате да използвате например α вместо x , тогава:

```
\documentclass{article}
\newcommand{\rowvec1}[1]{({#1}_1,{#1}_2,\ldots,{#1}_n)}
\begin{document}
Дадени са векторите  $\rowvec1{\alpha}$ ,
 $\rowvec1{\beta}$ , and  $\rowvec1{\gamma}$ .
\end{document}
```

[1] в командата `\newcommand{\rowvec1}[1]` уведомява ЛАТ_EX, че командата `\rowvec1` има един параметър за заместване, означен с `#1`. Следователно, командата `\rowvec1{\alpha}` «хваща» параметъра `\alpha` и го поставя на мястото на `#1` в `{({#1}_1,{#1}_2,\ldots,{#1}_n)}`.

В случай, че искате да сменяте и индекса на последната компонента на вектора, тогава:

```
\documentclass{article}
\newcommand{\rowvec2}[2]{({#1}_1,{#1}_2,\ldots,{#1}_{#2})}
```



```
\begin{document}
Дадени са вектори  $\vec{\alpha}_{2k+1}$ 
и  $\vec{\beta}_s$ .
\end{document}
```

[2] в `\newcommand{\rowvec2}[2]` съобщава на L^AT_EX, че командата `\rowvec` очаква два параметъра. След командата `\rowvec2{\alpha}{2k+1}` L^AT_EX «хваща» параметрите `\alpha` и `2k+1` и ги поставя съответно на мястото на `#1` и `#2` в `{({#1}_1, {#1}_2, \ldots, {#1}_{#2})}`.

L^AT_EX няма да ви разреши да създадете нова команда, която би измения вече съществуваща. Има обаче специална команда, в случай, че категорично искате да измените съществуваща команда: `\renewcommand`. Тя има същия синтаксис, както и командата `\newcommand`.

В някои случаи може също да се използва командата `\providecommand`. Тя работи така, както `\newcommand`, но, ако командата вече е дефинирана, L^AT_EX 2_ε мълчаливо я игнорира.

6.1.2 Нови среди

Аналогично на командата `\newcommand`, съществува команда, `\newenvironment`, с която можете да създавате ваша собствена среда:

```
\newenvironment{име}[брой-арг]{преди}{след}
```

Както и командата `\newcommand`, командата `\newenvironment` може да има незадължителен аргумент, който се използва по същия начин, както в командата `\newcommand`. Материалът, зададен в аргумента *преди*, се обработва преди обработката на текста вътре в средата. Материалът, включен в аргумента *след*, се обработва, когато се срещне командата `\end{име}`. Следващият пример илюстрира използването на командата `\newenvironment`.

```
\newenvironment{important}
{\rule{2.5ex}{1ex}%
 \hspace{\stretch{1}}}
{\hspace{\stretch{1}}%
 \rule{2.5ex}{1ex}}
```

```
\begin{important}
Важна информация\ldots
\end{important}
```

■ Важна информация... ■

L^AT_EX контролира дали не определяте вече съществуваща среда. Ако искате все пак да промените съществуваща среда, можете да използвате командата `\renewenvironment`. Тя има същия синтаксис като `\newenvironment`.

Командите, използвани в този пример, ще бъдат разяснени по-късно: описание на командата `\rule` има на стр. 130, командата `\stretch` е описана на стр. 124, а описанието на командата `\hspace` е на стр. 124.

6.1.3 Изработване на ваш собствен пакет

Ако дефинирате много нови команди и среди, преамбюлт на вашия документ ще стане прекалено дълъг. Разумно е да създадете L^AT_EX-пакет, който съдържа дефинициите на всички ваши команди и среди, както и използваните от вас допълнителни пакети. След това можете да използвате вашия нов пакет, като го декларирате в преамбюла на документа с командата `\usepackage`. Така новият пакет става достъпен в документа.

```
% Примерен собствен пакет
\ProvidesPackage{mypack}
\newcommand{\latcomp}{\emph{The \LaTeX{} Companion}}
\newcommand{\rowvec}{(x_1,x_2,\ldots,x_n)}
\newcommand{\rowvec1}[1]{({#1}_1,{#1}_2,\ldots,{#1}_n)}
\newcommand{\rowvec2}[2]{({#1}_1,{#1}_2,\ldots,{#1}_{#2})}
\newenvironment{important}%
{\rule{2.5ex}{1ex}\hspace{\stretch{1}}}
{\hspace{\stretch{1}}\rule{2.5ex}{1ex}}
```

Фигура 6.1: Примерен пакет

Създаването на пакета основно се състои в пренасяне на съдържанието на вашия преамбюл в отделен файл с разширение `.sty` и име по ваш избор, в случая `mypack`. Единствената специална команда в този стилев файл е командата

```
\ProvidesPackage{име на пакета}
```

която трябва да се постави в самото начало на файла `mypack.sty`. Командата `\ProvidesPackage` уведомява L^AT_EX за съществуването на нов пакет. Фигура 6.1 показва малък примерен стилев файл, съдържащ дефинираните в горните примери команди.

6.2 Шрифтове и техните размери

6.2.1 Команди за смяна на шрифта

L^AT_EX избира подходящия вид и размер на шрифта, основавайки се на логическата структура на документа (раздели, препратки, ...). В някои

случаи може да искате да смените шрифта ръчно. За да направите това, използвайте командите, дадени в Таблица 6.1 и Таблица 6.2. Действителният размер на всеки шрифт се определя от дизайна и зависи от класа на документа и неговите опции. Таблица 6.3 показва абсолютните точкови размери за онези команди, които се използват като инструменти в стандартните класове документи.

Съществуват три семейства (*families*) шрифт. Командите `\textrm`, `\textsf` и `\texttt` превключват печатането съответно на Roman, Sans Serif и Typewriter семейства, като Roman е по подразбиране.

Това е текст от семейство
`\textsf{Sans Serif}`, а това --
`\texttt{Typewriter}`
 семейство шрифт.

Това е текст от семейство Sans Serif, а това
 – Typewriter семейство шрифт.

Има две *serifs* шрифт. Командите `\textmd` и `\textbf` включват съответно Medium и **Boldface** серия, като Medium е по подразбиране.

Това е `\textbf{bold}`,
 докато `\textbf{това е}`
`\textmd{medium}` серия
 шрифт}.

Това е **bold**, докато **това е medium серия**
шрифт.

Има четири шрифтови наклона (*shapes*). Командите `\textup`, `\textit`, `\textsl` и `\textsc` превключват съответно на Upright, *Italic*, *Slant* и SMALL CAPS наклони, като Upright е наклон по подразбиране.

Важна роля при използване на командите за промяна на шрифта, играят т.нар. фигурни скоби. Те се използват за построяване на *групи*. Групирането по този начин ограничава областта на действие командите на L^AT_EX.

Има `\textsc{разлика}` между
`\textit{italic}` и `\textsl{slant}`
 наклони. `\textit{Ако текстът`
 вече е *italic*, може да се смени
 с `\textup{upright}`, за да се
 откроява думата в изречението}.
 Могат да се комбинират различни
 текстови команди като например
`\textsl{в \textbf{тази} фраза}`.
 Авторите често използват
 подчергани `\emph{думи}` или
`\emph{фрази}`, с използване на
 командата `\verb+\emph+`.

Има РАЗЛИКА между *italic* и *slant* наклони.
 Ако текстът вече е *italic*, може да се сме-
 ни с upright, за да се откроява думата в
 изречението. Могат да се комбинират раз-
 лични текстови команди като например в
тази фраза. Авторите често използват под-
 чергани *думи* или *фрази*, с използване на
 командата `\emph`.

Важна особеност на L^AT_EX 2_ε е това, че атрибутите на шрифта са независими. Това означава, че можете да давате команди за смяна на

размера или дори на шрифта, съхранявайки при това установените порано атрибути за наклон или наситеност.

В *математически режим* можете да използвате *команди* за смяна на шрифта, за да излезете временно от *математически режим* и да влезете в нормален текст. Ако искате да превключите на друг шрифт за математическо отпечатване, се нуждаете от друг, специален набор команди; вж. Таблица 6.4.

Командите за смяна размера на шрифта, променят и разстоянието между редовете, но само ако параграфа завършва в областта на действие на командата за смяна размера на шрифта. Затова затваряща фигурна скоба } не трябва да се слага прекалено рано.¹:

{\Large Забележете положението на командата \verb|\par| в този пример!\par}

Забележете положението на командата \par в този пример!

¹Командата \par е еквивалентна на празен ред.

Таблица 6.1: Шрифтове

<code>\textrm{...}</code>	прав шрифт	<code>\textsf{...}</code>	sans serif
<code>\texttt{...}</code>	пишеща машина		
<code>\textmd{...}</code>	нормален	<code>\textbf{...}</code>	плътен
<code>\textup{...}</code>	прав шрифт	<code>\textit{...}</code>	<i>курсив</i>
<code>\textsl{...}</code>	<i>наклонен шрифт</i>	<code>\textsc{...}</code>	МАЛКИ ГЛАВНИ
<code>\emph{...}</code>	<i>отделен шрифт</i>	<code>\textnormal{...}</code>	нормален

Таблица 6.2: Размер на шрифта

<code>\tiny</code>	мъничък	<code>\Large</code>	ПО-ГОЛЯМ
<code>\scriptsize</code>	много малък	<code>\LARGE</code>	най-ГОЛЯМ
<code>\footnotesize</code>	твърде малък	<code>\huge</code>	едър
<code>\small</code>	малък	<code>\Huge</code>	ГРАМАДЕН
<code>\normalsize</code>	нормален		
<code>\large</code>	ГОЛЯМ		

Таблица 6.3: Абсолютни размери на шрифтовете в стандартните класове

<i>Размер</i>	<i>10pt (по преобладаване)</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

Таблица 6.4: Математически шрифтове

<i>Команда</i>	<i>Пример</i>	<i>Извеждане</i>
<code>\mathcal{...}</code>	<code>\$\$\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$\$\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$\$\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$\$\mathsf{G}\times R\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$\$\mathtt{L}(b,c)\$</code>	$L(b, c)$
<code>\mathnormal{...}</code>	<code>\$\$\mathnormal{R_{19}}\neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$\$\mathit{ffi}\neq ffi\$</code>	$ffi \neq ffi$

```
{\Large Забележете положението
на командата \verb|\par| в този
пример!}\par
```

Забележете положението на командата `\par` в този пример!

Ако искате командата за изменение на размера на шрифта да се приложи към цял параграф или към няколко параграфа, то тогава можете да използвате синтаксис на среда за смяна размера на шрифта.

```
\begin{Large}
Това е параграф, в който текстът
е с размер \verb|Large|.
\end{Large}
```

Това е параграф, в който текстът е с размер `Large`.

Този подход ще ви спести писането на много фигурни скоби.

6.2.2 Логическо маркиране

Важно е да се отбележи, че е опасно да се претрупват вашите документи с явни команди, като описаните по-горе, защото това противоречи на основната идея на L^AT_EX: разделяне на логическата и визуалната структура на документа. Това означава, че ако използвате едни и същи команди за смяна на шрифта на няколко места в документа за отпечатване на специален вид информация, вие трябва да използвате `\newcommand`, за да дефинирате «логическа команда» за смяна на шрифта или неговия размер.

Да предположим например, че пишете книга по биология и искате имената на видовете да се появяват в наклонен шрифт навсякъде в текста. В никакъв случай не използвайте `\textsl` за всяко име на вид; това е нерационално. Вместо това, използвайте следния подход.

```
\newcommand{\genus}[1]{\textsl{#1}}
\genus{Делфинът} е бозайник \ldots
```

Делфинът е бозайник ...

По-късно, ако вашият издател препоръча да смените имената на видовете от наклонен на *italic shape*, ще трябва да смените само една команда, а не дълго да се ровите в целия документ, търсейки стотиците имена на видове в текста.

Този пример ясно показва основното предимство на системи от типа на L^AT_EX пред WYSIWYG (what you see is what you get) word-процесори.

6.3 Интервали

6.3.1 Интервали между редовете

Ако искате да използвате по-големи интервали между редовете, можете да промените тяхната стойност с поставянето на командата

```
\linespread{коэффициент}
```

в преамбула на документа. Използвайте `\linespread{1.3}` за междуредов интервал «едно и половина», а за «двоен» интервал – `\linespread{1.6}`. Нормално редовете не са «дебели», така че по подразбиране коефициента на дебелина на редовете е равен на 1.

Да отбележим, че ефектът от командата `\linespread` е твърде драстичен и затова тя не е подходяща за публикации. Затова, ако имате сериозни основания за промяна на междуредовия интервал, по-добре използвайте следната команда:

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
  {1.5\baselineskip}
Този параграф е отпечатан с интервал
1.5 пъти по-голям отколкото е бил
преди. Обърнете внимание на командата
\par{ } в края на параграфа.\par}
```

След затваряне на фигурната скоба, всичко се връща в нормално състояние.

Този параграф е отпечатан с интервал 1.5 пъти по-голям отколкото е бил преди. Обърнете внимание на командата

в края на параграфа.

След затваряне на фигурната скоба, всичко се връща в нормално състояние.

6.3.2 Форматиране на параграфи

В \LaTeX има два параметъра, които влияят на вида на параграфа. Поставяйки дефиниция от вида

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

в преамбула на входния файл, вие можете да смените оформлението на параграфите. Тези две команди увеличават разстоянието между параграфите и установяват параграфния отстъп равен на нула.

Във втората команда, `plus` и `minus` указват на \TeX , че може да увеличава и намалява интервала между параграфите с указаните величини,

ако това е необходимо за правилното разполагане на параграфите на страницата.

В Европа, параграфите често се отделят с интервали и не се прави никакъв отстъп. Обаче имайте предвид, че това влияе и на Съдържанието: неговите редове също стават по-разредени. За да се избегне това, съответните команди могат да се пренесат от преамбюла на документа някъде след `\tableofcontents`, или най-добре да не се използват изобщо, защото повечето професионални книги използват «отстъп» (а не интервали, за отделяне на параграфите).

Ако искате да направите параграфен отстъп в параграф, който няма такъв, поставете командата¹

```
\indent
```

в началото на параграфа. Разбира се, ефект от нея ще има само ако `\parindent` не е установен на нула.

За създаване на параграф без отстъп, може да използвате

```
\noindent
```

като първа команда в параграфа. Това е удобно, когато започвате документа с текст, а не с команди за секциониране.

6.3.3 Хоризонтални интервали

Л_AT_EX определя автоматично интервалите между думите и изреченията. За да добавите хоризонтален интервал, използвайте

```
\hspace{дължина}
```

Ако такъв интервал *трябва* да бъде оставен, дори ако той попада в началото или края на реда, използвайте `\hspace*` вместо `\hspace`. В най-простите случаи *дължина* е просто число плюс единица за измерване. Най-важните единици са изброени в Таблица 6.5.

Тук `\hspace{1.5cm}` интервалът е 1,5см.

Тук интервалът е 1,5см.

¹За добавяне на отстъп към първия параграф след заглавие на раздел, се използва пакета `indentfirst` от комплекта 'tools'.

Таблица 6.5: Единици за размерност в Т_ЕX

mm	милиметър $\approx 1/25$ in	□
cm	сантиметър = 10 mm	□
in	inch = 25.4 mm	□
pt	точка $\approx 1/72$ in $\approx \frac{1}{3}$ mm	□
em	примерна ширина на буквата ‘M’ в текущия шрифт	□
ex	примерна височина на буквата ‘x’ в текущия шрифт	□

Командата

`\stretch{n}`

генерира специален «еластичен» интервал. Той се разтяга, докато се запълни цялото оставащо място на реда. Ако на един ред се използват две команди `\hspace{\stretch{n}}`, то те се разтягат пропорционално на своите коефициенти.

`a\hspace{\stretch{1}}`
`a\hspace{\stretch{3}}a`

a a a

При използване на хоризонтални интервали заедно с текст, има смисъл да се генерира интервал, който регулира своя размер според размера на текущия шрифт. Това може да се направи чрез относителните единици за размерност `em` и `ex`:

`{\Large}big\hspace{1em}y\`
`{\tiny}tin\hspace{1ex}y`

big y
 tin y

6.3.4 Вертикални интервали

Интервалите между параграфи, раздели, подраздели, ... се определят от Л_АT_ЕX автоматично. Ако е необходимо, допълнителен вертикален интервал *между два параграфа* може да се добави с командата

`\vspace{дължина}`

Тази команда трябва да се поставя между два реда текст. Ако това пространство трябва да се остави горе или долу на страницата, използвайте вариант на командата със звездичка: `\vspace*`.

Командата `\stretch`, в комбинация с `\pagebreak`, може да се използва за отпечатване на текст на последния ред на страницата или за вертикално центриране на текст на страницата:

Някакъв текст\ldots

`\vspace{\stretch{1}}` Това ще се окаже на последния ред на страницата. `\pagebreak`

Допълнителен интервал между два реда в *един* параграф или вътре в таблица се определя с командата

`\[дължина]`

С помощта на `\bigskip`, `\medskip` и `\smallskip` можете да оставите предопределени вертикални интервали, без да се замисляте за конкретни размери.

6.4 Макет на страницата

L^AT_EX 2_ε ви позволява да определите размер на хартията в командата `\documentclass`. Това автоматично определя съответните текстови полета. Но понякога предопределените стойности може да не ви устройват. Естествено, вие можете да ги измените. Фигура 6.2 показва всички параметри, които могат да бъдат променени. Тя е генерирана с помощта на пакета `layout` от комплекта ‘tools’.¹

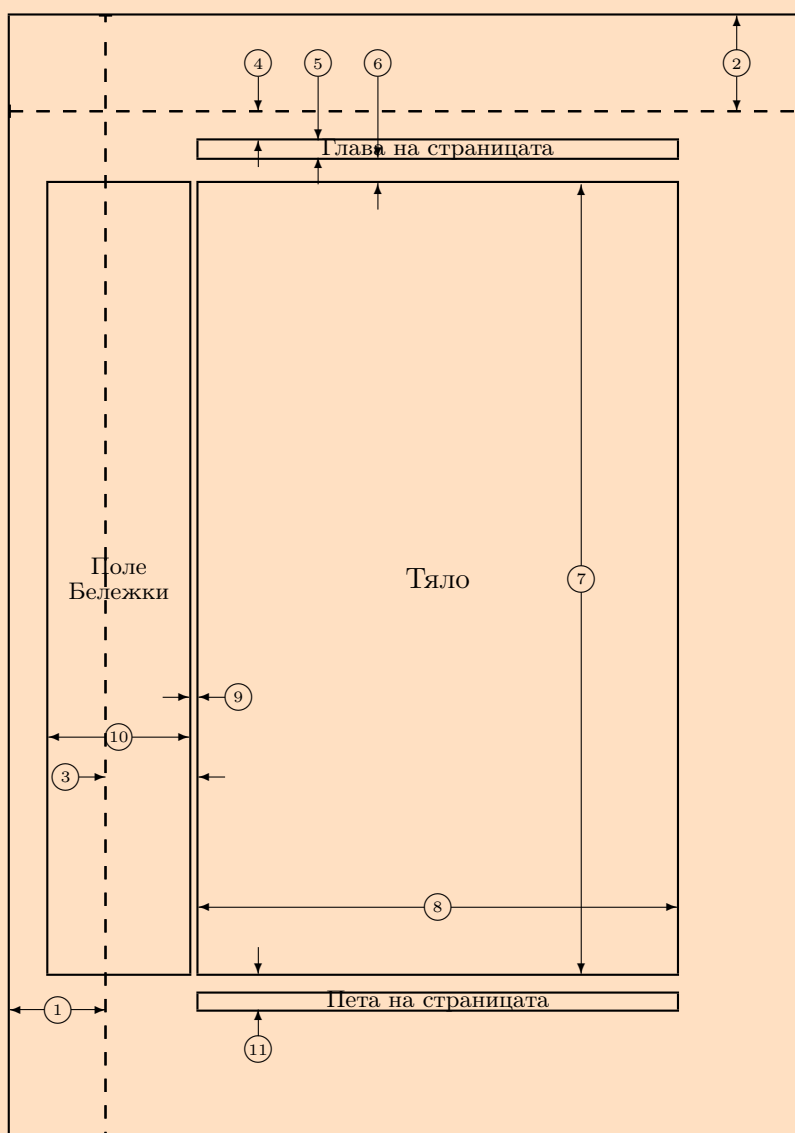
Почакайте! . . . преди веднага да се хвърлите в безумието да правите «тази твърде тясна страница малко по-широка», нека малко да помислим. Подобно на много други неща, има сериозни основания за това, макетът на страницата в L^AT_EX да е такъв, какъвто е.

Безусловно, ако направите сравнение със страница, отпечатана с ново-инсталиран MS Word, то страниците на L^AT_EX изглеждат ужасно тесни. Обаче, погледнете вашата любима книга² и пребройте символите на един стандартен текстов ред. Ще откриете, че на всеки ред има не повече от 66 символа. Сега повторете това със страниците на L^AT_EX. Ще видите, че и тук също има около 66 символа на ред. Опитът показва, че четенето става толкова по-трудно, колкото повече символи има на един обикновен ред. Това е така, защото за очите е трудно да се преместват от края на един ред към началото на следващия. Именно затова вестниците често се отпечатват в няколко колони.

Така че, ако увеличите ширината на вашия текст, имайте предвид, че затруднявате живота на вашите читатели.

¹CTAN:/tex-archive/macros/latex/required/tools

²Има се предвид истинска печатна книга, издадена от уважавано издателство.



1	1 инч + <code>\hoffset</code>	2	1 инч + <code>\voffset</code>
3	<code>\oddsidemargin = 22pt</code> или <code>\evensidemargin</code>	4	<code>\topmargin = 22pt</code>
5	<code>\headheight = 13pt</code>	6	<code>\headsep = 19pt</code>
7	<code>\textheight = 595pt</code>	8	<code>\textwidth = 360pt</code>
9	<code>\marginparsep = 7pt</code>	10	<code>\marginparwidth = 106pt</code>
11	<code>\footskip = 27pt</code>		<code>\marginparpush = 5pt</code> (не е показано)
	<code>\hoffset = 0pt</code>		<code>\voffset = 0pt</code>
	<code>\paperwidth = 597pt</code>		<code>\paperheight = 845pt</code>

Фигура 6.2: Параметри на макета на страницата.

Ето все пак как можете да промените макета...

ЛАТ_ЕX предоставя две команди за изменение на параметри те за промяна на макета, които обикновено се използват в преамбюла на документа.

Първата команда присвоява фиксирана стойност на всеки от параметрите:

```
\setlength{параметър}{дължина}
```

Втората команда добавя дължина към всеки от параметрите:

```
\addtolength{параметър}{дължина}
```

Втората команда всъщност е по-полезна, отколкото `\setlength`, защото ви позволява да правите настройка относно съществуващите установявания. За да се добави един сантиметър към общата ширина на текста, например, в преамбюла се поставя следното:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

В тази връзка, полезен пакет е `calc`. Той позволява да се използват аритметични операции в аргумента на `\setlength` и в други команди, при които могат да се въвеждат числени стойности в нейните аргументи.

6.5 Блокове

ЛАТ_ЕX построява страниците, използвайки за строителен материал различни по големина блокове.

Така всяка буква е малък блок, който се залепва към други букви, формирайки дума. Думата се залепва за други думи, но чрез специално «еластично» лепило, което може да се разтяга или свива така, че точно да запълни реда на страницата.

Това обяснение наистина е твърде елементарно, но идеята е в това, че Т_ЕX винаги работи с блокове и лепило. Не само буквата може да бъде блок. Вие можете да поставите в блок практически всичко, дори и други блокове. Всеки блок след това се обработва с ЛАТ_ЕX, така, сякаш това е отделна буква.

В предните глави ние вече срещнахме някои блокове. Примери за това са средите `tabular` и `\includegraphics`, и двете произвеждащи блок. Това означава, че вие лесно можете да подредите две таблици или фигури една до друга. Само трябва да проверите, дали тяхната обща ширина не надвишава `\textwidth`.

Всеки параграф също може да се опакова в блок, или с командата

```
\parbox[позиция]{ширина}{текст}
```

или в средата

```
\begin{minipage}[позиция]{ширина} текст \end{minipage}
```

Параметърът *позиция*, който може да е една от буквите *c*, *t* или *b*, контролира вертикалното позициониране на блока по отношение на базовата линия на околния текст.

Основната разлика между `\minipage` и `\parbox` е в това, че вътре в `\parbox` не могат да се използват всички команди и среди, докато вътре в `\minipage` може практически всичко.

Докато `\parbox` опакова целия параграф, разбивайки редовете, съществува клас блокови команди, обработващи само хоризонтално разположен материал. Една от тях е познатата вече команда `\mbox`, която просто опакова последователност от блокове в един блок. Тази команда се използва най-често за предотвратяване разделянето от L^AT_EX на две свързани думи.

```
\makebox[ширина][позиция]{текст}
```

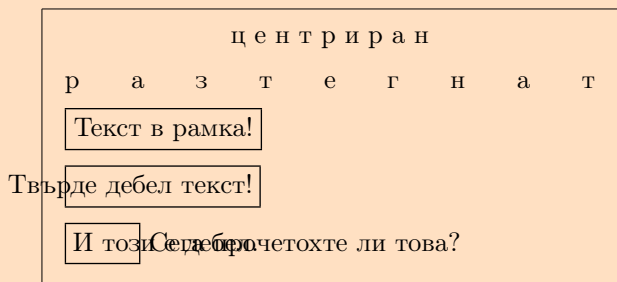
Ширина определя ширината на получения блок. Освен числови изрази за дължина, могат да се използват командите `\width`, `\height`, `\depth` и `\totalheight` в параметъра *ширина*. Те приемат стойности, равни на стойностите, получени при измерването на отпечатвания *текст*.

Параметърът *позиция* приема еднобуквена стойност: *c* – центриран, *l* – подравнен вляво, *r* – подравнен вдясно или *s* (*spread*) – равномерно се запълва блока с текст.

Командата `\framebox` работи точно така, както `\makebox`, но рисува рамка около текста.

Следващият пример показва някои възможности на използването на командите `\makebox` и `\framebox`.

```
\makebox[\textwidth]{%
  ц е н т р и р а н } \par
\makebox[\textwidth][s]{%
  р а з т е г н а т } \par
\framebox[1.1\width]{Текст
  в рамка!} \par
\framebox[0.8\width][r]{%
  Твърде дебел текст!} \par
\framebox[1cm][l]{И този
  е дебел.} Сега прочетохте ли това?
```



Подобна команда има и за управление на вертикалното разположение на материала. Командата

```
\raisebox{преместване}[разтег-над-оснлин][разтег-под-оснлин]{текст}
```

позволява да се определят вертикалните характеристики на блока. В първите три параметъра могат да се използват командите `\width`, `\height`, `\depth` и `\totalwidth`, за да се въздейства на размера на блока, съдържащ материала, вътре в аргумента *текст*.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{a}%
\raisebox{-2.2ex}{a}%
\raisebox{-4.5ex}{a}}
викал той, но дори стоящият до
него не забелязал, че с него се
е случило нещо ужасно.
```

Aaaaаааа викал той, но дори стоящият до него не забелязал, че с него се е случило нещо ужасно.

6.6 Линийки и подпори

Сигурно сте забелязали на страница 117 командата

```
\rule[преместване]{ширина}{височина}
```

При обикновено използване тя генерира просто черен блок.

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



Тази команда е полезна за рисуване на вертикални и хоризонтални линии. Например, линията на заглавната страница е нарисувана с командата `\rule`.

Специален случай е линийката, която няма ширина, но има определена височина. В професионалната текстообработка я наричат *подпора*. Тя се използва, за гарантиране на определена минимална височина на даден елемент на страницата. Бихте могли да я използвате в средата `tabular`, за да сте сигурни, че даден ред има определена минимална височина.

```
\begin{tabular}{|c|}  
\hline  
\rule{1pt}{4ex}Материали \ldots\  
\hline  
\rule{0pt}{4ex}Стомана\  
\hline  
\end{tabular}
```

Материали ...
Стомана

Библиография

- [1] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The TeXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.
- [3] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, USA, 2004, ISBN 0-201-36299-6.¹
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The LaTeX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.²
- [5] Michel Goossens and Sebastian Rahtz, *The LaTeX Web Companion: Integrating TeX, HTML and XML*, Addison-Wesley, Reading, Massachusetts, 1999, ISBN 0-201-43311-7.³
- [6] Всеки дистрибутив на LaTeX трябва да съдържа така наречения *LaTeX Local Guide*, обясняващ особеностите на локалната система. Той трябва да се намира във файл, наречен `local.tex`.
- [7] LaTeX3 Project Team. *LaTeX 2_ε for authors*. Включен в дистрибутива на LaTeX 2_ε като `usrguide.tex`.
- [8] LaTeX3 Project Team. *LaTeX 2_ε for Class and Package writers*. Включен в дистрибутива на LaTeX 2_ε като `clsguide.tex`.
- [9] LaTeX3 Project Team. *LaTeX 2_ε Font selection*. Включен в дистрибутива на LaTeX 2_ε като `fntguide.tex`.

¹Издаден на руски език: М.Гуссенс, Ф.Миттелбах, А.Самарин. *Путеводитель по пакету LaTeX и его расширению LaTeX 2_ε*. Мир, 1999, ISBN 5-03-003325-4.

²Издаден на руски език: М.Гуссенс, С.Ратц и Ф.Миттелбах. *Путеводитель по пакету LaTeX и его графическим расширениям*. Мир, 2002, ISBN 5-03-003388-2.

³ Издаден на руски език: Гуссенс, М., Ратц, С. *Путеводитель по пакету LaTeX и его Web – приложениям*, М., Мир, 2001

- [10] D. P. Carlisle. *Packages in the 'graphics' bundle*. Влиза в състава на комплекта 'graphics' като `grfguide.tex`, достъпен в дистрибутива на \LaTeX .
- [11] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of \LaTeX 's verbatim Environments*. Влиза в състава на комплекта 'tools' като `verbatim.dtx`, достъпен в дистрибутива на \LaTeX .
- [12] Vladimir Volovich, Werner Lemberg and \LaTeX 3 Project Team. *Cyrillic languages support in \LaTeX* . Включен в дистрибутива на \LaTeX 2 ϵ като `cyrguide.tex`.
- [13] Graham Williams. *The TeX Catalogue* пълен списък на множество пакети, свързани с \TeX и \LaTeX . Достъпен в Интернет на адрес [CTAN:/tex-archive/help/Catalogue/catalogue.html](http://ctan.org/tex-archive/help/Catalogue/catalogue.html)
- [14] Keith Reckdahl. *Using EPS Graphics in \LaTeX 2 ϵ Documents* обяснява всичко, което бихте искали да знаете за EPS файловете и тяхното използване в \LaTeX -документите. Достъпен в Интернет на адрес [CTAN:/tex-archive/info/epslatex.ps](http://ctan.org/tex-archive/info/epslatex.ps)
- [15] Kristoffer H. Rose. *Xy-pic User's Guide*. Достъпен в Интернет на адреса на CTAN заедно с дистрибутива на Xy-pic.
- [16] John D. Hobby. *A User's Manual for MetaPost*. Достъпен на адрес <http://cm.bell-labs.com/who/hobby/>
- [17] Alan Hoenig. *TeX Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)
- [18] Urs Oswald. *Graphics in \LaTeX 2 ϵ* , съдържа набор изходни текстове на Java за генерация на произволни окръжности и елипси от средата `picture`, и *MetaPost - A Tutorial*. И двата са достъпни на адрес <http://www.ursoswald.ch>
- [19] Tobias Oetiker, H. Partl, I. Hyna, E. Schlegl. *The not so Short Introduction in \LaTeX 2 ϵ* . Достъпен на адрес; [CTAN:/tex-archive/tex-archive/info/lshort](http://ctan.org/tex-archive/info/lshort)

Азбучен указател

- \!, 56
- ", 26
- "', 34
- "-, 34
- "---, 34
- "<, 34
- "=, 34
- ">, 34
- "‘, 34
- \$, 49
- ’, 53
- \(, 49
- \), 49
- \,, 51, 56
- , 26
- , 26
- \-, 25
- , 26
- , 26
- ..., 28
- \:, 56
- \;, 56
- «, 26
- », 26
- \@, 35
- \[, 50
- Интервал
 - в началото на реда, 5
- Команди, 5
 - \!, 56
 - \(, 49
 - \), 49
 - \,, 51, 56
 - \-, 25
 - \:, 56
 - \;, 56
 - \@, 35
 - \[, 50
 - \addtolength, 128
 - \Alph, 34
 - \alph, 34
 - \and, 37
 - \appendix, 36, 38
 - \ar, 104
 - \arccos, 53
 - \arcsin, 53
 - \arctan, 53
 - \arg, 53
 - \Asbuk, 34
 - \asbuk, 34
 - \author, 37, 89
 - \background, 94
 - \backmatter, 38
 - \backslash, 5
 - \begin, 40, 96
 - \bibitem, 75
 - \Big, 55
 - \big, 55
 - \Bigg, 55
 - \bigg, 55
 - \bigskip, 126
 - \binom, 54
 - \bmod, 53
 - \boldmath, 61
 - \boldsymbol, 62
 - \caption, 47
 - \cdot, 53

- `\cdots`, 55
- `\chapter`, 36, 79
- `\chaptermark`, 79
- `\circle`, 99
- `\circle*`, 99
- `\cite`, 75
- `\cleardoublepage`, 48
- `\clearpage`, 48
- `\cline`, 43
- `\cos`, 53
- `\cosh`, 53
- `\cot`, 53
- `\coth`, 53
- `\csc`, 53
- `\date`, 37
- `\ddots`, 55
- `\deg`, 53
- `\depth`, 129, 130
- `\det`, 53
- `\dim`, 53
- `\displaystyle`, 59
- `\documentclass`, 10, 14, 24, 83
- `\dq`, 33
- `\emlema`, 94
- `\emph`, 39, 120
- `\end`, 40, 96
- `\enumBul`, 34
- `\enumEng`, 34
- `\enumLat`, 34
- `\eqref`, 50
- `\EUR`, 27
- `\euro`, 27
- `\exp`, 53
- `\externaldocument`, 86
- `\fbox`, 25
- `\flq`, 33
- `\flqq`, 26, 33
- `\foldera`, 103
- `\folderb`, 103
- `\footnote`, 38
- `\footskip`, 127
- `\frac`, 54
- `\framebox`, 129
- `\frenchspacing`, 33, 35
- `\frontmatter`, 37
- `\frq`, 33
- `\frqq`, 26, 33
- `\fussy`, 24
- `\gcd`, 53
- `\glqq`, 26
- `\grqq`, 26
- `\headheight`, 127
- `\headsep`, 127
- `\height`, 129, 130
- `\hline`, 43
- `\hom`, 53
- `\href`, 86, 88, 89, 91
- `\hspace`, 118, 124
- `\hyperlink`, 86
- `\hypertarget`, 86
- `\hyphenation`, 24
- `\idotsint`, 56
- `\iiiint`, 56
- `\iiint`, 56
- `\iint`, 56
- `\include`, 15
- `\includegraphics`, 74, 85, 90, 128
- `\includeonly`, 15
- `\indent`, 124
- `\index`, 77
- `\inf`, 53
- `\input`, 15
- `\int`, 54
- `\item`, 40
- `\ker`, 53
- `\label`, 38, 50, 63, 86
- `\latcomp`, 116
- `\LaTeX`, 25
- `\LaTeXe`, 25
- `\ldots`, 28, 55
- `\left`, 55
- `\leftmark`, 79
- `\lg`, 53
- `\lim`, 53
- `\liminf`, 53
- `\limsup`, 53
- `\line`, 98, 103

`\linebreak`, 23
`\linespread`, 123
`\linethickness`, 100, 101, 103
`\listoffigures`, 37, 47
`\listoftables`, 37, 47
`\ln`, 53
`\log`, 53
`\mainmatter`, 37
`\makebox`, 129
`\makeindex`, 76
`\maketitle`, 37
`\marginparpush`, 127
`\marginparsep`, 127
`\marginparwidth`, 127
`\marginwidth`, 94
`\mathbb`, 51
`\mathrm`, 59
`\max`, 53
`\mbox`, 25, 129
`\medskip`, 126
`\min`, 53
`\minipage`, 129
`\multicolumn`, 44
`\multirow`, 97, 100
`\newcommand`, 115, 117
`\newenvironment`, 117
`\newline`, 23
`\newpage`, 23
`\newsavebox`, 102
`\newtheorem`, 60
`\noindent`, 124
`\nolinebreak`, 23
`\nonumber`, 58, 63
`\nopagebreak`, 23
`\not`, 67
`\oddsidemargin`, 127
`\oval`, 101, 103
`\overbrace`, 52
`\overlay`, 94
`\overleftarrow`, 53
`\overline`, 52
`\overrightarrow`, 53
`\pagebreak`, 23
`\pageref`, 38, 82
`\pagestyle`, 12
`\panelwidth`, 91
`\paperheight`, 127
`\paperwidth`, 127
`\parabola`, 112
`\paragraph`, 35
`\parbox`, 129
`\parindent`, 123
`\parskip`, 123
`\part`, 35, 36
`\pause`, 94
`\phantom`, 58
`\pmod`, 53
`\Pr`, 53
`\printindex`, 77
`\prod`, 54
`\providecommand`, 117
`\ProvidesPackage`, 118
`\psarc`, 112
`\psbezier`, 112
`\psdots`, 112
`\psframe`, 110
`\psframebox`, 108
`\psline`, 109
`\psset`, 108
`\put`, 97–102
`\qbezier`, 95, 97, 104
`\qquad`, 51, 56
`\quad`, 51, 56
`\raisebox`, 130
`\ref`, 38, 50, 82
`\renewcommand`, 117
`\renewenvironment`, 117
`\right`, 55
`\right.`, 55
`\rightmark`, 79
`\rput*`, 108
`\rule`, 118, 130
`\savebox`, 102
`\screensize`, 94
`\scriptscriptstyle`, 59
`\scriptstyle`, 59
`\sec`, 53
`\section`, 35, 79

- `\sectionmark`, 79
`\selectlanguage`, 30
`\setlength`, 96, 123, 128
`\sin`, 53
`\sinh`, 53
`\sloppy`, 24
`\smallskip`, 126
`\sqrt`, 52
`\stackrel`, 54
`\stretch`, 118, 125
`\subparagraph`, 35
`\subsection`, 35
`\subsectionmark`, 79
`\substack`, 54
`\subsubsection`, 35
`\sum`, 54
`\sup`, 53
`\tableofcontents`, 36, 37
`\tan`, 53
`\tanh`, 53
`\TeX`, 25
`\texorpdfstring`, 89
`\text`, 64
`\textbf`, 119
`\textcelsius`, 27
`\texteuro`, 27
`\textheight`, 127
`\textit`, 119
`\textmd`, 119
`\textrm`, 59, 119
`\textsc`, 119
`\textsf`, 119
`\textsl`, 119
`\textstyle`, 59
`\texttt`, 119
`\textup`, 119
`\textwidth`, 127
`\thicklines`, 98, 101, 103
`\thinlines`, 101, 103
`\thispagestyle`, 12
`\title`, 37
`\today`, 25
`\topmargin`, 127
`\totalheight`, 129
`\totalwidth`, 130
`\underbrace`, 52
`\underline`, 39, 52
`\unitlength`, 96, 98
`\usebox`, 102
`\usepackage`, 12, 14, 27, 29, 30, 85, 118
`\vdots`, 55
`\vec`, 53
`\vector`, 98
`\verb`, 42, 43
`\verbatim@font`, 80
`\verbatiminput`, 80
`\vspace`, 125
`\widehat`, 53
`\widetilde`, 53
`\width`, 129, 130
`\xunit`, 108
`\xymatrix`, 104
`\yunit`, 108
- Пакети
- `ambsy`, 62
`amfonts`, 51, 72
`amsmath`, 54, 56, 58, 59, 62, 64
`amssymb`, 51, 66
`Babel`, 29
`babel`, 7, 18, 24, 29–31, 33, 34, 91
`bm`, 62
`calc`, 128
`color`, 84, 91, 108
`doc`, 13
`epic`, 95, 99
`epic`, 95
`eufrak`, 72
`europs`, 27
`eurosans`, 27
`eurosym`, 27
`euscript`, 72
`exscale`, 13
`fancyhdr`, 79
`fontenc`, 13, 30, 33
`geometry`, 80
`graphicx`, 74, 84, 85, 91

- hyperref, 83, 85–87, 89, 91
 - hyphenat, 80
 - ifthen, 13
 - indentfirst, 124
 - inputenc, 13, 30, 32, 33
 - latexsym, 13
 - layout, 126
 - longtable, 45
 - makeidx, 13, 76
 - marvosym, 27
 - mathrsfs, 72
 - mathtext, 33
 - mypack, 118
 - pause, 94
 - pdfscreen, 91, 94
 - pst-col, 113
 - pst-grad, 108, 111, 113
 - pst-text, 113
 - pstcol, 108
 - PSTricks, 107–112
 - pstricks, 95, 96, 99, 108
 - pxfonts, 84
 - showidx, 77
 - supertabular, 45
 - syntonly, 13, 16
 - textcomp, 27
 - txfonts, 84
 - verbatim, 7, 80
 - xr-hyper, 86
 - xy, 104
- Разширение
- .aux, 15
 - .cls, 14
 - .dtx, 14
 - .dvi, 14, 74
 - .eps, 16, 74
 - .fd, 14
 - .idx, 15, 77
 - .ilg, 15
 - .ind, 15, 77
 - .ins, 14
 - .lof, 14
 - .log, 14
 - .lot, 15
 - .ps, 16
 - .sty, 14, 80, 118
 - .tex, 9, 12
 - .toc, 14
- Съдържание, 36
- Среди
- Bmatrix, 64
 - Vmatrix, 64
 - abstract, 42
 - align*, 62
 - align, 62, 63
 - array, 56, 57, 64
 - bmatrix, 64
 - cases, 64
 - center, 40
 - comment, 7
 - description, 40
 - displaymath, 50
 - enumerate, 40
 - eqnarray, 57, 62
 - equation, 50, 57
 - figure, 46, 47, 73
 - flushleft, 40
 - flushright, 40
 - itemize, 40
 - math, 49
 - minipage, 129
 - multline, 63
 - picture, 95, 96, 99, 100
 - pmatrix, 64
 - pspicture, 96
 - quotation, 41
 - quote, 41
 - subarray, 54
 - subequations, 63
 - table, 46, 47, 73
 - tabular, 43, 56, 128
 - thebibliography, 75
 - verbatim, 42, 80
 - verse, 41
 - vmatrix, 64
- Шрифт
- \footnotesize, 120
 - \Huge, 120

- `\huge`, 120
- `\LARGE`, 120
- `\Large`, 120
- `\large`, 120
- `\mathbf`, 121
- `\mathcal`, 121
- `\mathit`, 121
- `\mathnormal`, 121
- `\mathrm`, 121
- `\mathsf`, 121
- `\mathtt`, 121
- `\normalsize`, 120
- `\scriptsize`, 120
- `\small`, 120
- `\textbf`, 120
- `\textit`, 120
- `\textmd`, 120
- `\textnormal`, 120
- `\textrm`, 120
- `\textsc`, 120
- `\textsf`, 120
- `\textsl`, 120
- `\texttt`, 120
- `\textup`, 120
- `\tiny`, 120
- акцент, 28
 - математически, 53
- акценти
 - acute, 29
 - grave, 29
 - umlaut, 29
- азбучен указател, 76
- ажурни дебели символи, 51
- български език, 33
- библиография, 75
- букви
 - европейски, 29
- дебели символи, 51, 61
- дефис, 26
- дълги уравнения, 58
- дълго тире, 26
- дължина, 124
- долен индекс, 52
- дроб, 54
- други езици, 28
- дума, 78
- две колони, 11
- двустранен изход, 11
- единици, 124, 125
- една колона, 11
- едностранен изход, 11
- фигури, 47
- фигурни скоби, 6, 119
 - хоризонтални, 52
- формули, 49
- френски, 31
- функцията модул, 53
- горен индекс, 52
- графика, 73
- гръцки букви, 52
- групиране, 119
- хипертекст, 82
- интервал, 5
 - двоен, 123
 - хоризонтален, 124
 - междуредов, 123
 - след команда, 6
 - вертикален, 125
- кавички, 26
- късо тире, 26
- клас
 - article, 10
 - book, 10
 - report, 10
 - slides, 10
- кодировка
 - шрифт
 - LGR, 31
 - OT1, 31
 - T1, 31, 33
 - T2*, 33
 - T2A, 31, 33
 - T2B, 31
 - T2C, 31
 - X2, 31
 - входна
 - ansinew, 30
 - applemac, 30

- cp1251, 30, 33
- cp850, 30
- cp866nav, 30
- koi8-ru, 30
- latin1, 30
- macukr, 30
- кодировка на шрифт
 - LGR, 31
 - OT1, 31
 - T1, 31, 33
 - T2*, 33
 - T2A, 31, 33
 - T2B, 31
 - T2C, 31
 - X2, 31
- кодировка на шрифта, 13, 30
- колонтитул
 - долен, 12
 - горен, 12
- коментарни редове, 6
- курсив, 120
- квадратен корен, 52
- квадратни скоби, 6
- линии
 - хоризонтални, 52
- макет на страницата, 126
- малки главни, 120
- математически
 - функции, 53
 - интервал, 56
 - минус, 26
 - ограничител, 55
- математика, 49
- многоточие, 28
- наклонен шрифт, 120
- немски, 32
- немският език, 29
- незадължителни параметри, 6
- ограничители, 55
- опции, 10
- оператор
 - интеграл, 54
 - произведение, 54
 - сума, 54
- пакет, 7, 10
- пакет makeidx, 76
- параметър, 6
- плаващи обекти, 45
- плътен, 120
- подчертаване, 39
- подпора, 130
- подравняване
 - по десетичната точка, 44
 - вдясно или вляво, 40
- полета, 126
- прав шрифт, 120
- правила за пренасяне, 29
- празни символи, 5
- преамбюл, 7
- предимствата на L^AT_EX, 3
- прекъсване на реда, 23
- препратки, 38
- програма makeindex, 76
- производна, 53
- размер на хартията, 83, 126
- размер на основния шрифт, 11
- размер на страница, 11
- разширение, 12
- руски език, 33
- символи
 - резервирани, 5
- системи уравнения, 57
- скоби, 55
- специални символи, 28
- спецификация на позицията, 47
- среда, 40
- стил на страницата
 - empty, 12
 - headings, 12
 - plain, 12
- стилове на страницата, 12
- страница
 - заглавна, 11, 37
 - A4, 11
 - A5, 11
 - B5, 11
 - executive, 11
 - legal, 11

- letter, 11
- структура на файла, 7
- шрифт, 118
 - документ, размер, 11
 - математически, размер, 59
 - размер, 118, 120
- таблицы, 47
- тилда, 26, 53
- тилда (~), 35
- типове файлове, 12
- тире, 26
 - дълго, 26
 - късо, 26
- точка, 27
- точка, интервал след, 35
- точки
 - диагонални, 55
 - хоризонтални, 55
 - вертикални, 55
- три точки, 55
- украински език, 33
- вектор, 53
- входен файл, 7
- входна кодировка
 - ansinew, 30
 - applemac, 30
 - cp1251, 30, 33
 - cp850, 30
 - cp866nav, 30
 - koi8-ru, 30
 - latin1, 30
 - macukr, 30
- заглавие на документа, 11
- заглавна страница, 37
- запетая, 27
- знак минус, 26
- знак стрелка, 53
- знак за градус, 27
- \\, 23, 40, 41, 43, 126
- *, 23
- \\], 50
- ^, 52
- , 52
- ~, 35
- Ghostscript , 16
- abstract, 42
- Acrobat Reader, 82
- \\addtolength, 128
- Adobe Acrobat Reader , 16
- æ, 29
- align, 62, 63
- align*, 62
- \\Alph, 34
- \\alph, 34
- amsbsy, 62
- amsmath, 51, 72
- amsmath, 54, 56, 58, 59, 62, 64
- amssymb, 51, 66
- \\and, 37
- ansinew, 30
- \\appendix, 36, 38
- applemac, 30
- \\ar, 104
- \\arccos, 53
- \\arcsin, 53
- \\arctan, 53
- \\arg, 53
- array, 56, 57, 64
- \\Asbuk, 34
- \\asbuk, 34
- \\author, 37, 89
- Babel, 29
- babel, 7, 18, 24, 29–31, 33, 34, 91
- \\background, 94
- \\backmatter, 38
- backslash, 6
- \\backslash, 5
- \\begin, 40, 96
- \\bibitem, 75
- \\Big, 55
- \\big, 55
- \\Bigg, 55
- \\bigg, 55
- \\bigskip, 126
- \\binom, 54
- bm, 62

- Bmatrix, 64
- bmatrix, 64
- \bmod, 53
- \boldmath, 61
- \boldsymbol, 62

- calc, 128
- \caption, 47
- cases, 64
- \cdot, 53
- \cdots, 55
- center, 40
- \chapter, 36, 79
- \chaptermark, 79
- \circle, 99
- \circle*, 99
- \cite, 75
- \cleardoublepage, 48
- \clearpage, 48
- \cline, 43
- cm-super, 17
- color, 84, 91, 108
- comment, 7
- \cos, 53
- \cosh, 53
- \cot, 53
- \coth, 53
- cp1251, 30, 33
- cp850, 30
- cp866nav, 30
- \csc, 53

- \date, 37
- \ddots, 55
- \deg, 53
- \depth, 129, 130
- description, 40
- \det, 53
- \dim, 53
- displaymath, 50
- \displaystyle, 59
- doc, 13
- \documentclass, 10, 14, 24, 83
- doubleline, 109

- doublesep, 109
- \dq, 33
- DVI, 16
- dvi, 16

- eepic, 95, 99
- \emblema, 94
- \emph, 39, 120
- empty, 12
- Encapsulated POSTSCRIPT, 73, 85, 96
- \end, 40, 96
- \enumBul, 34
- \enumEng, 34
- enumerate, 40
- \enumLat, 34
- epic, 95
- eqnarray, 57, 62
- \eqref, 50
- equation, 50, 57
- eufrak, 72
- \EUR, 27
- \euro, 27
- europs, 27
- eurosans, 27
- eurosym, 27
- euscript, 72
- \exp, 53
- exscale, 13
- \externaldocument, 86

- fancyhdr, 79
- \fbox, 25
- figure, 46, 47, 73
- \flq, 33
- \flqq, 26, 33
- flushleft, 40
- flushright, 40
- foiltex, 10
- \foldera, 103
- \folderb, 103
- fontenc, 13, 30, 33
- \footnote, 38
- \footnotesize, 120

- `\footskip`, 127
- `\frac`, 54
- `\framebox`, 129
- `\frenchspacing`, 33, 35
- `\frontmatter`, 37
- `\freq`, 33
- `\frqq`, 26, 33
- `\fussy`, 24
- `\gcd`, 53
- geometry, 80
- GhostScript, 73
- `\glqq`, 26
- graphicx, 74, 84, 85, 91
- `\grqq`, 26
- GSView, 16
- `\headheight`, 127
- headings, 12
- `\headsep`, 127
- `\height`, 129, 130
- `\hline`, 43
- `\hom`, 53
- `\href`, 86, 88, 89, 91
- `\hspace`, 118, 124
- `\Huge`, 120
- `\huge`, 120
- `\hyperlink`, 86
- hyperref, 83, 85–87, 89, 91
- `\hypertarget`, 86
- hyphenat, 80
- `\hyphenation`, 24
- и j без точка, 29
- `\idotsint`, 56
- ifthen, 13
- `\iiiint`, 56
- `\iiint`, 56
- `\iint`, 56
- `\include`, 15
- `\includegraphics`, 74, 85, 90, 128
- `\includeonly`, 15
- `\indent`, 124
- indentfirst, 124
- `\index`, 77
- `\inf`, 53
- `\input`, 15
- inputenc, 13, 30, 32, 33
- `\int`, 54
- `\item`, 40
- itemize, 40
- `\ker`, 53
- Knuth, Donald E., 1
- koi8-ru, 30
- `\label`, 38, 50, 63, 86
- Lamport, Leslie, 1
- `\LARGE`, 120
- `\Large`, 120
- `\large`, 120
- `\latcomp`, 116
- `\LaTeX`, 25
- LaTeX 2.09, 2
- LaTeX 2_ε, 2
- LaTeX3, 2
- `\LaTeXe`, 25
- latexsym, 13
- latin1, 30
- layout, 126
- `\ldots`, 28, 55
- `\left`, 55
- `\leftmark`, 79
- `\lg`, 53
- LGR, 31
- `\lim`, 53
- `\liminf`, 53
- `\limsup`, 53
- `\line`, 98, 103
- linearc, 109
- `\linebreak`, 23
- `\linespread`, 123
- linestyle, 109
- `\linethickness`, 100, 101, 103
- linewidth, 109
- `\listoffigures`, 37, 47
- `\listoftables`, 37, 47
- `\ln`, 53
- `\log`, 53

- longtable, 45
- macukr, 30
- \mainmatter, 37
- \makebox, 129
 - makeidx, 13, 76
- \makeindex, 76
- \maketitle, 37
- \marginparpush, 127
- \marginparsep, 127
- \marginparwidth, 127
- \marginwidth, 94
 - marvosym, 27
- math, 49
- \mathbb, 51
- \mathbf, 121
- \mathcal, 121
- \mathit, 121
- \mathnormal, 121
- \mathrm, 59, 121
 - mathrsfs, 72
- \mathsf, 121
 - mathtext, 33
- \mathtt, 121
- \max, 53
- \mbox, 25, 129
- \medskip, 126
 - METAPOST, 16
 - METAPOST, 85
 - MiKTeX, 16, 17
- \min, 53
- \minipage, 129
 - minipage, 129
 - Mittelbach, Frank, 2
 - Mittelbach, Frank, 1
- \multicolumn, 44
- \multirow, 97, 100
 - multline, 63
- mypack, 118
- \newcommand, 115, 117
- \newenvironment, 117
- \newline, 23
- \newpage, 23
- \newsavebox, 102
- \newtheorem, 60
- \noindent, 124
- \nolinebreak, 23
- \nonumber, 58, 63
- \nopagebreak, 23
- \normalsize, 120
- \not, 67
- \oddsidemargin, 127
 - œ, 29
 - OT1, 31
- \oval, 101, 103
- \overbrace, 52
 - overfull hbox, 24
- \overlay, 94
- \overleftarrow, 53
- \overline, 52
- \overrightarrow, 53
- \pagebreak, 23
- \pageref, 38, 82
- \pagestyle, 12
- \panelwidth, 91
- \paperheight, 127
- \paperwidth, 127
- \parabola, 112
- \paragraph, 35
- \parbox, 129
- \parindent, 123
- \parskip, 123
- \part, 35, 36
 - pause, 94
- \pause, 94
 - PDF, 16, 81
- pdfL^AT_EX, 83, 91
- pdflatex, 16
- pdfscreen, 91, 94
- pdftex, 16
- pdfL^AT_EX, 83
- pdfT_EX, 83
- \phantom, 58
- picture, 95, 96, 99, 100
- plain, 12

- pmatrix, 64
- \pmod, 53
- POSTSCRIPT, 4, 9, 16, 73, 74, 82–84, 96, 107
 - Encapsulated, 73, 85, 96
- \Pr, 53
- \printindex, 77
- \prod, 54
- \providecommand, 117
- \ProvidesPackage, 118
- \psarc, 112
- \psbezier, 112
- \psdots, 112
- \psframe, 110
- \psframebox, 108
- \psline, 109
- pspicture, 96
- \psset, 108
- pst-col, 113
- pst-grad, 108, 111, 113
- pst-text, 113
- pstcol, 108
- PSTricks, 107–112
- pstricks, 95, 96, 99, 108
- \put, 97–102
- pxfonts, 84
- \qbezier, 95, 97, 104
- \qqquad, 51, 56
- \quad, 51, 56
- quotation, 41
- quote, 41
- \raisebox, 130
- \ref, 38, 50, 82
- \renewcommand, 117
- \renewenvironment, 117
- \right, 55
- \right., 55
- \rightmark, 79
- \rput*, 108
- \rule, 118, 130
- sans serif, 120
- \savebox, 102
- \screensize, 94
- \scriptscriptstyle, 59
- \scriptsize, 120
- \scriptstyle, 59
- \sec, 53
- \section, 35, 79
- \sectionmark, 79
- \selectlanguage, 30
- \setlength, 96, 123, 128
 - showidx, 77
 - showpoints, 109
- \sin, 53
- \sinh, 53
- \sloppy, 24
- \small, 120
- \smallskip, 126
- \sqrt, 52
- \stackrel, 54
- \stretch, 118, 125
 - subarray, 54
 - subequations, 63
- \subparagraph, 35
- \subsection, 35
- \subsectionmark, 79
- \substack, 54
- \subsubsection, 35
- \sum, 54
- \sup, 53
- supertabular, 45
- syntonly, 13, 16
- T1, 31, 33
- T2*, 33
- T2A, 31, 33
- T2B, 31
- T2C, 31
- table, 46, 47, 73
- \tableofcontents, 36, 37
- tabular, 43, 56, 128
- \tan, 53
- \tanh, 53
- \TeX, 25
- \texorpdfstring, 89
- \text, 64

- `\textbf`, 119, 120
- `\textcelsius`, 27
- `textcomp`, 27
- `\texteuro`, 27
- `\textheight`, 127
- `\textit`, 119, 120
- `\textmd`, 119, 120
- `\textnormal`, 120
- `\textrm`, 59, 119, 120
- `\textsc`, 119, 120
- `\textsf`, 119, 120
- `\textsl`, 119, 120
- `\textstyle`, 59
- `\texttt`, 119, 120
- `\textup`, 119, 120
- `\textwidth`, 127
- `thebibliography`, 75
- `\thicklines`, 98, 101, 103
- `\thinlines`, 101, 103
- `\thispagestyle`, 12
- `\tiny`, 120
- `\title`, 37
- `\today`, 25
- `\topmargin`, 127
- `\totalheight`, 129
- `\totalwidth`, 130
- `txfonts`, 84

- `\underbrace`, 52
- `underfull hbox`, 24
- `\underline`, 39, 52
- `\unitlength`, 96, 98
- `URL`, 26
- `\usebox`, 102
- `\usepackage`, 12, 14, 27, 29, 30, 85, 118

- `\vdots`, 55
- `\vec`, 53
- `\vector`, 98
- `\verb`, 42, 43
- `verbatim`, 7, 80
- `verbatim`, 42, 80
- `\verbatim@font`, 80

- `\verbatiminput`, 80
- `verse`, 41
- `Vmatrix`, 64
- `vmatrix`, 64
- `\vspace`, 125

- `\widehat`, 53
- `\widetilde`, 53
- `\width`, 129, 130
- `WinEdt`, 17
- `www`, 26
- `WYSIWYG`, 2, 3

- `X2`, 31
- `xpdf`, 82
- `xr-hyper`, 86
- `\xunit`, 108
- `xy`, 104
- `\xymatrix`, 104

- `YAP`, 16
- `\yunit`, 108

Стефка Караколева

ВЪВЕДЕНИЕ

в издателската система

L^AT_EX 2_ε

Първо издание

Формат: 70/100/16

Печатни коли: 20

Издателски коли: 10

Тираж: 50

Печатна база при Русенски Университет „Ангел Кънчев“

Русе, 2005

