

# Новости NCC-L<sup>A</sup>T<sub>E</sub>X

А.И. Роженко

**Аннотация.** В этом документе приведены отличия новой версии макроса NCC от NCC-3.05 (для L<sup>A</sup>T<sub>E</sub>X-2.09).

## Содержание

Введение . . . . .	1
1. Заказ класса NCC . . . . .	2
2. Управление размерами и стилем оформления страницы . . . . .	3
3. Набор текста . . . . .	4
3.1. Составные слова и тире . . . . .	4
3.2. Дополнительные боксы . . . . .	4
3.3. Разное . . . . .	6
4. Математика . . . . .	8
4.1. Выключные формулы . . . . .	8
4.2. Другие математические команды . . . . .	10
4.3. Теоремы . . . . .	12
5. Рубрикация . . . . .	14
6. Рисунки и таблицы . . . . .	14
6.1. Базовые команды . . . . .	15
6.2. Оформление с обтеканием . . . . .	15
6.3. Плавающее оформление рисунка или таблицы . . . . .	16
6.4. Плавающее оформление пары рисунков или таблиц . . . . .	16
7. Графика . . . . .	17

## Введение

Новая версия макроса NCC адаптирована автором под современный L<sup>A</sup>T<sub>E</sub>X. В процессе адаптации макрос был полностью переработан. Команды, представляющие самостоятельный интерес, перенесены в отдельные пакеты коллекции *ncctools*. Их можно использовать совместно с другими классами L<sup>A</sup>T<sub>E</sub>X. Собственно макрос NCC и сопутствующие ему пакеты образуют коллекцию *ncclatex*. Ядро макроса NCC интегрировано пакетом **ncclatex**, который также можно использовать совместно с другими классами.

Настоящая версия документа описывает только основные отличия нового макроса. В дальнейшем ее планируется расширить.

# 1. Заказ класса NCC

```
\documentclass[Список опций]{ncc}
```

Класс NCC содержит 4 стиля оформления документов, заказываемые следующими опциями:

`article` – статья.

`preprint` – препринт. Команда `\preprint{номер}` задает номер препринта. Титул оформляется на отдельной странице.

`book` – монография. Команда `\bookeditor{текст}` позволяет вставить текст под названием книги (обычно это научные редакторы книги).

`report` – отчет. Эквивалентен стилю `book` с опцией `oneside`.

Размеры текстового поля определяются по базовому кеглю документа:

Опция	Ширина текста	Высота текста
10pt	110 мм	157 мм
11pt	126.5 мм	199 мм
12pt	145 мм	233 мм
14pt	160 мм	240 мм

Класс NCC поддерживает все стандартные опции заказа размера страницы: `a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper`, `executivepaper`, `landscape`. Дополнительно имеется опция `a5a4paper`, применяемая для печати текста формата A5 на принтере формата A4, имеющем центрированный карман ввода.

В отличие от стандартных классов, опции заказа размера страницы не изменяют размеров текстового поля. Чтобы пересчитать размеры текстового поля на полную страницу, используйте опцию `fittopage`.

Класс NCC поддерживает следующие стандартные опции: `oneside`, `twoside`, `draft`, `final`, `openright` (в стиле `book` новая глава начинается с нечетной страницы, в стиле `article` пустая четная страница добавляется в конец статьи при необходимости), `openany` (нет выравнивания по нечетной странице), `onecolumn`, `twocolumn`, `openbib`, `fleqn`, `leqno`.

Опция `russian` заказывает русский стиль оформления заголовков, загружает пакет многоязыковой поддержки **Babel** с опцией `russian`, включает `\frenchspacing` (без дополнительных отбивок после конца предложения) и переопределяет команды `\alph`, `\Alph` на алфавитную нумерацию кириллицей (нумерация латиницей сохраняется в командах `\alphlatin`, `\Alphlatin`).

Заказа опции `russian` вполне достаточно при наборе русского текста в стандартной кириллической кодировке Windows (CP1251). Если же текст набран в другой кодировке, то необходимо подгрузить пакет `inputenc`, заказав в параметрах название кодировки. Например, если текст набран в кодировке DOS (CP866), то закажите

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

По умолчанию устанавливаются опции `10pt`, `a4paper`, `article`, `twoside`, `onecolumn`, `final`, `openany`.

## 2. Управление размерами и стилем оформления страницы

Во всех стилях оформления документа, кроме `article`, используется стиль оформления страницы `headings`. В стиле `article` используется `myheadings`. Текстовое поле центрируется в странице без учета размеров поля маргинальных сносок и подножия страницы. Размеры поля текста и его расположение на странице можно изменить командой

```
\ToCenter[hfm]{ширина}{высота}
```

Необязательный параметр `[hfm]` содержит любую комбинацию из этих трех букв, означающих названия полей, которые должны быть учтены при центрировании текста: `h` – учитывать колонтитул (header), `f` – учитывать подножие (footer), `m` – учитывать поле маргинальных сносок.

Вместо команды `\ToCenter` можно использовать команду

```
\FromMargins[hfm]{слева}{справа}{сверху}{снизу}
```

задающую отступы от краев страницы (для четной страницы в режиме `twocolumn` левый и правый отступы меняются местами).

Обе эти команды имеют `*`-форму, не содержащую параметра `[hfm]`. В этом случае размеры и позиция поля текста вычисляются без учета дополнительных полей, устанавливается `\pagestyle{empty}` и подавляется вывод информации в `aux`-файл. Использование: для оформления коротких документов, например, служебных записок.

Команды `\ToCenter` и `\FromMargins` доступны только в преамбуле документа (до команды `\begin{document}`). Эти команды заменяют команду `\field` из старого макроса `NCS`.

Чтобы заказать подчеркивание колонтитула, подгрузите пакет

```
\usepackage[список стилей страниц]{ncchdr}
```

В дополнительном параметре указывается список стилей страницы, для которых требуется линейка в колонтитуле. Последний из перечисленных стилей заодно устанавливается в качестве текущего стиля. *Примечание: стили `headings` и `myheadings` имеют общую основу в макросе `NCS`. Указав один из них в параметрах загрузки пакета, Вы автоматически заказываете линейку и для другого.*

По умолчанию толщина линейки в колонтитуле равна 0.4pt. В данном документе линейка более толстая. Для переопределения ее толщины использована следующая команда:

```
\renewcommand{\headrulewidth}{0.8pt}
```

Имеется также возможность указать для конкретной страницы специальный колонтитул с помощью команды `\thispageheading{текст}` (в старом макросе эта команда называлась `\thisheading`). В таком колонтитуле допустим многострочный текст.

Техническая рамка страницы, реализованная в старом макросе командой `\border`, в новом макросе заказывается иначе, через механизм поддержки водяных знаков (пакет `watermark` из коллекции `ncctools`):

```
\usepackage{ncscropmark}
\watermark{\cropmark[h]{10mm}{10mm}{10mm}{10mm}{10mm}}
```

Спецификация команды `\cropmark` следующая:

```
\cropmark[hfm]{слева}{справа}{сверху}{снизу}{длина}
```

Необязательный параметр `[hfm]` имеет тот же смысл, что и в командах определения размера текстового поля. Следующие четыре параметра задают размеры отступов от выбранной комбинации полей (в примере это текстовое поле вместе с колонтитулом). Последний параметр задает длину линии в отрисовываемых уголках. Если длина положительна, то уголки “смотрят” внутрь. Если же длина отрицательная, то наружу. Для задания крестов используйте совместно две кропмарки внутри одной водяной марки:

```
\watermark{\cropmark[h]{10mm}{10mm}{10mm}{10mm}{10mm}
\cropmark[h]{10mm}{10mm}{10mm}{10mm}{-10mm}}
```

### 3. Набор текста

В этом пункте описываются команды общего назначения, применяемые при наборе текста.

#### 3.1. Составные слова и тире

В русском языке довольно часто встречаются составные слова, соединенные дефисом или средним тире. Перенос таких слов в  $\text{\LaTeX}$  затруднен (возможен только по дефису или тире). Макрос `NCC` содержит команды, обеспечивающие возможность переноса составных слов (реализация выполнена в пакете `extdash` из коллекции `ncctools`). Имеется две разновидности таких команд: 1) с разрешением переноса после дефиса или тире, 2) с запретом переноса в этой точке (вторая разновидность разрешает перенос обоих составных слов, но запрещает перенос в точке их соединения). Аналогично, имеется две разновидности длинного тире и два варианта маленького растяжимого пробела (стандартные маленькие пробелы  $\text{\LaTeX}$  не растяжимы). Для всех этих команд имеются аббревиатуры:

Описание	Перенос разрешен		Перенос запрещен	
	Команда	Аббрев.	Команда	Аббрев.
Дефис	<code>\hyphdash</code>	<code>\-/</code>	<code>\hyphdash*</code>	<code>\=/</code>
Среднее тире	<code>\endash</code>	<code>\--</code>	<code>\endash*</code>	<code>\==</code>
Длинное тире	<code>\emdash</code>	<code>\---</code>	<code>\emdash*</code>	<code>\===</code>
Полупробел	<code>\halfspace</code>	<code>\-,</code>	<code>\halfspace*</code>	<code>\=,</code>

Команда длинного тире автоматически вставляет растяжимые полупробелы вокруг. Ее рекомендуется использовать для набора тире между словами и для прямой речи. Если Вы хотите, чтобы длинное тире было несколько короче (что соответствует российским полиграфическим традициям), добавьте опцию `syremdash` в заказ класса `NCC`. Неразбиваемый маленький растяжимый пробел `\=,`, можно использовать в инициалах и сокращениях. Например, `t.\=,e.` выглядит так: `t.e.`

#### 3.2. Дополнительные боксы

Макрос `NCC` загружает пакет `nccboxes` из коллекции `ncctools`, определяющий дополнительные боксы:

`\j hbox{прототип}[align]{текст}` размещает *текст* в боксе, ширина которого задается параметром *прототип*. Дополнительный параметр *align* задает выравнивание (`l`, `c`, `r` или `s`; по умолчанию `c`).

`\j vbox{прототип}[align]{текст}` сдвигает текст по вертикали, используя для выравнивания “подпорку”, определяемую вертикальными размерами *прототипа*. Дополнительный параметр *align* задает вертикальное выравнивание относительно подпорки (`t`, `c` или `b`; по умолчанию `c`): `t` означает выравнивание высоты (верха) текста и подпорки, `b` – выравнивание глубины (низа) текста и подпорки, `c` – центрирование текста относительно подпорки. Результирующая высота и глубина бокса вычисляются как максимум между соответствующими параметрами подпорки и выровненного по ней текста.

`\jparbox{прототип}[align]{ширина}{тело}` формирует параграфный бокс требуемой ширины и выравнивает его вертикально относительно прототипа тем же способом, что и в команде `\jvbox`.

`\pbox[align]{тело}` реализует простую одностолбцовую таблицу без обводки и отступов. Параметр *align* может состоять из двух букв, определяющих горизонтальное выравнивание текста внутри столбца (l, c или r) и вертикальное выравнивание таблицы относительно окружающего текста (t, c или b). Например, команда `\pbox[lt]{...}` эквивалентна команде

```
\begin{tabular}[t]{@{}l@{}}... \end{tabular}
```

с одним маленьким исключением, что расстояние между строками в таблице `\pbox` не зависит от значения `\arraystretch`.

`\addbox{сверху}{снизу}{текст}` создает бокс, высота которого увеличена на величину, задаваемую параметром *сверху*, глубина которого увеличена на величину *снизу*. Например, `\addbox{.5ex}{.5ex}{text}` увеличивает высоту и глубину бокса на 0.5ex.

`\picbox{body} = \begin{picture}(0,0)(0,0) body \end{picture}`.

Для подготовки “красивых” таблиц можно использовать следующие команды:

`\Strut/число/` задает подпорку, высота и глубина которой вычисляются на основе прототипа (буквы A) следующим образом: если *число* > 0, то высота подпорки увеличивается на величину интерлиньяжа (расстояния между базовыми линиями соседних строк текста), умноженную на это число; если *число* < 0, то глубина подпорки увеличивается на величину интерлиньяжа, умноженную на модуль этого числа. Например, `\Strut/1/` вставляет подпорку, высота которой превышает высоту буквы A на величину одной строки. Команда `\Strut` без параметра эквивалентна команде `\Strut/0/`.

`\tstrut`, `\bstrut`, `\tbstrut` вставляют подпорки, высота, глубина, высота и глубина которых превышает размеры буквы A на специальную величину. Эта величина вычисляется так, чтобы полная высота (высота + глубина) подпорки `\tbstrut` была в 1.5 больше величины интерлиньяжа.

`\cbox/число/[align]{тело}` создает бокс, чье тело есть одностолбцовая таблица. Его высота и глубина увеличиваются с помощью `\tstrut` в начале и `\bstrut` в конце тела. Горизонтальное выравнивание (l, c или r) внутри столбца и вертикальное выравнивание относительно подпорки (t, c или b) задаются параметром *align*. По умолчанию используется центрированное выравнивание. Результирующий бокс вертикально выравнивается относительно подпорки `\Strut/число/` с помощью команды `\jvbox`. Команда `\cbox*` дополнительно обнуляет высоту и глубину результирующего бокса (используется для оформления мультистолбцовых ячеек таблиц).

Команда `\cbox` применяется в “головке” таблиц. Стиль вывода тела команды определяется командой `\cboxstyle`. В макросе NCS эта команда переопределена следующим образом:

```
\renewcommand\cboxstyle{\scriptsize}
```

Продемонстрируем использование подпорок и команды `\cbox` на примере таблицы

Vertically spanned head	Simple head	Very long head of two lines	
	subhead	subhead	subhead
Text	field	field	field
Text	field	field	field
Text	field	field	field

которая была сделана следующим образом:

```

\begin{center}
\renewcommand\cboxstyle{\small\bf}
\setlength{\tabcolsep}{10pt}
\begin{tabular}{|l|c|c|c|}\hline
\cbox*{-1.5/{Vertically}\spanned head} & \cbox{Simple head}
&\multicolumn2{c|}{\cbox{Very long head\of two lines}}\\ \cline{2-4}
&\cbox{subhead} & \cbox{subhead} & \cbox{subhead} \\ \hline
\Strut/1/ Text & field & field & field \\
& Text & field & field & field \\
\bstrut Text & field & field & field \\ \hline
\end{tabular}
\end{center}

```

*Примечание: можно также использовать команды `\tbox`, `\bbox`, `\tbbox` из старого макроса NCC. Однако они считаются устаревшими и поэтому рекомендуется вместо них использовать команды `\tstrut`, `\bstrut`, `\tbstrut`.*

### 3.3. Разное

Новые команды макроса NCC:

`\openrightorany` исполняет команду `\clearpage` или `\cleardoublepage` в зависимости от open-моды, выбранной в опциях класса (`openright` или `openany`). Если создается пустая четная страница, то текст в ее колонтитуле будет пуст.

`\AuthorBeforeTitle`, `\TitleBeforeAuthor` управляют порядком вывода названия и списка авторов командой `\maketitle`. Используются только в преамбуле документа. В стиле `article` сначала идет название, а затем авторы. В остальных стилях все наоборот.

`\noeqbreak`, `\alloweqbreak` управляют возможностью разрыва страницы перед выключной формулой. Первая команда запрещает разрыв, а вторая разрешает. По умолчанию действует `\alloweqbreak`.

`\SetTOCStyle{стиль}` задает стиль оформления содержания, списка рисунков и таблиц. Используется в преамбуле документа. Стиль по умолчанию пуст. Но например, чтобы вывести содержание размером `\small`, воспользуйтесь командой `\SetTOCStyle{\small}`.

`\PnumPrototype{число}` задает прототип номера страницы, используемого для вычисления величины правого отступа в содержании документа. По умолчанию прототип равен 99. Если в документе более 99 страниц, то вставьте команду `\PnumPrototype{999}` в преамбулу документа.

`biblist` позволяет оформлять библиографию без заголовка и начиная с любого указанного номера. Синтаксис аналогичен структуре `thebibliography`, но имеется необязательный параметр, задающий начальный номер для счетчика библиографических ссылок:

```
\begin{biblist}[начальный номер]{прототип} ... \end{biblist}
```

Библиографические ссылки будут нумероваться начиная с номера, на единицу больше заказанного. Стиль оформления библиографии задается командой `\bibliststyle` (по умолчанию ее значение `\small`). Структура `thebibliography` создана на основе `biblist`.

Из старого макроса NCC перешли следующие дополнительные команды:

`desclist` есть расширенная версия структуры `description`:

```
\begin{desclist}{перед}{после}[прототип] ... \end{desclist}
```

Параметры *перед* и *после* задают оформление маркера списка и определяют команды, вставляемые перед и после текста маркера. Необязательный параметр *прототип* задает прототип текста, по которому вычислять левый отступ текста элементов списка.

`\setyear{год}`, `\theyear`: первая команда задает год, а вторая – печатает заданный год. По умолчанию команда `\theyear` печатает текущий год.

`\acknow` начинает новый параграф с заголовком *Acknowledgements* или *Благодарности* (в русском стиле).

`\cref{метка}` выводит ссылку на номер библиографического элемента без квадратных скобок. Используется при подготовке ссылок, содержащих сложные перечисления, например, ссылок вида [2–4, 7].

`\No` выводит символ №. Если опция `russian` не используется, то символ номера эмулируется текстом “No.”.

`\tl{текст}`, `\tc{текст}`, `\tr{текст}` выключка текста влево, по центру и вправо с помощью пружин `\hspace*{\fill}`. Эти команды используются в таблицах, когда выравнивание в ячейке должно отличаться от выравнивания во всем столбце.

`\bff` включает полное ожирнение текста и математики (устанавливается шрифт `\bf` и `\mathversion{bold}`).

### Команды, специфичные для стиля `article`:

```
\setseries{текст}, \setvolume{том}, \setissue{номер}
```

управляют серией, томом и номером сборника, в котором издается статья. Эти команды обычно задаются в локальном файле `nccadd.sty`, который автоматически загружается в стиле `article`. Вывод этой информации выполняется соответственно командами `\theseries`, `\thevolume`, `\theissue`. В режиме `openright` в конце статьи при необходимости добавляется пустая четная страница.

**Команды, специфичные для стиля `preprint`.** Команда `\preprint{номер}` задает номер препринта. Если номер задан, то под титулом препринта будет добавлен текст

### Препринт номер

**Команды, специфичные для стилей `book` и `report`.** Так же как и в стандартном классе `book`, в макросе NCC можно использовать команды `\frontmatter`, `\mainmatter`, `\backmatter`, определяющие логическое разбиение текста книги на стартовые страницы (с римской нумерацией и нумерованными главами), основную часть (арабская нумерация страниц и нормальная нумерация глав) и завершающие страницы (с нумерованными главами). Команда `\bookeditor{текст}` позволяет вставлять фрагмент текста на титульной странице под заголовком монографии и списком автором. Можно управлять способом вывода тега названия главы в колонтитуле и содержании книги. По умолчанию этот тег состоит из номера главы. Но можно сделать, чтобы перед номером шел префикс “Глава”. Для этого служит команда `\ChapterPrefixStyle{список}`, в списке которой через запятую можно указывать два слова: `header` и `toc`. Указав `header`, Вы включаете вывод префикса в колонтитуле, а указав `toc`, задаете вывод префикса в содержании.

## 4. Математика

### 4.1. Выключные формулы

Новая версия макроса NCC обеспечивает практически полную поддержку выключных формул старого стиля. Однако новая реализация существенно отличается от старой, поскольку базируется на средствах пакета **amsmath**. В этой связи, часть команд старого стиля NCC хотя и реализована, но считается устаревшей и вместо них рекомендуется использовать следующие эквиваленты:

<code>\Eq{...}</code>	→	<code>\eq*{...}</code>
<code>\Eqs{...}</code>	→	<code>\eqs*{...}</code>
<code>\Eqalign{...}</code>	→	<code>\eqalign*{...}</code>
<code>\no{...}</code>	→	<code>\tag*{...}</code>
<code>\eref{...}</code>	→	<code>\eqref{...}</code>

Следующие примеры демонстрируют команды набора выключных формул средствами пакета **amsmath**, которые теперь можно использовать с макросом NCC.

<code>\begin{equation*}</code> <code>a=b</code> <code>\end{equation*}</code>		$a = b$	
<code>\begin{equation}</code> <code>a=b</code> <code>\end{equation}</code>		$a = b$	(1)
<code>\begin{equation}\label{xx}</code> <code>\begin{split}</code> <code>a&amp; =b+c-d\\</code> <code>&amp; \quad +e-f\\</code> <code>&amp; =g+h\\</code> <code>&amp; =i</code> <code>\end{split}</code> <code>\end{equation}</code>		$\begin{aligned} a &= b + c - d \\ &+ e - f \\ &= g + h \\ &= i \end{aligned}$	(2)
<code>\begin{multline}</code> <code>a+b+c+d+e+f\\</code> <code>+i+j+k+l+m+n</code> <code>\end{multline}</code>		$\begin{aligned} a + b + c + d + e + f \\ + i + j + k + l + m + n \end{aligned}$	(3)
<code>\begin{gather}</code> <code>a_1=b_1+c_1\\</code> <code>a_2=b_2+c_2-d_2+e_2</code> <code>\end{gather}</code>		$a_1 = b_1 + c_1$	(4)
		$a_2 = b_2 + c_2 - d_2 + e_2$	(5)
<code>\begin{align}</code> <code>a_1&amp; =b_1+c_1\\</code> <code>a_2&amp; =b_2+c_2-d_2+e_2</code> <code>\end{align}</code>		$a_1 = b_1 + c_1$	(6)
		$a_2 = b_2 + c_2 - d_2 + e_2$	(7)
<code>\begin{align}</code> <code>a_{11}&amp; =b_{11}&amp;</code> <code>  a_{12}&amp; =b_{12}\\</code> <code>a_{21}&amp; =b_{21}&amp;</code> <code>  a_{22}&amp; =b_{22}+c_{22}</code> <code>\end{align}</code>		$a_{11} = b_{11}$	(8)
		$a_{12} = b_{12}$	(8)
		$a_{21} = b_{21}$	(9)
		$a_{22} = b_{22} + c_{22}$	(9)





В  $\TeX$ е используются два размера отступов перед выключными формулами: короткий и нормальный. Если последняя строка параграфа достаточно короткая и размер выключной формулы такой, что между концом текста и формулой имеется достаточно места, то используется короткий отступ. В противном случае используется нормальный отступ. Однако, если оформляется массив формул, то отступ перед ними будет всегда нормальным, вне зависимости от их реальной ширины. Чтобы заставить  $\LaTeX$  использовать короткий отступ, в макросе `NCC` имеется команда `\useshortskip`, действующая на ближайшую выключную формулу.

## 4.2. Другие математические команды

Следующие математические команды перешли из старого макроса `NCC`:

`\nr` имеет точно такой же синтаксис как команда `\l`, но вставляет дополнительное расстояние в `0.5ex`. Эта команда часто используется в массивах выключных формул для дополнительной раздвижки некоторых строк.

`\mrel{строки}` создает новый математический символ из категории символов отношения с помощью одностолбцовой таблицы. Например, `\mrel{\langle\langle[-.7ex]\rangle\rangle}` выводит  $\lesssim$ .

`\underrel{база}{подножие}` служит двойником к команде `\overrel`. Например, `\A\underrel{\longrightarrow}{x\to 0}B` выводит  $A \xrightarrow{x \rightarrow 0} B$ .

`\mop{name}` используется для кодирования нестандартных математических операций типа `\sin`. Например, `\k=\mop{sgn}x` выводит  $k = \operatorname{sgn} x$  с правильным выбором шрифта математической операции и отступов. Эта команда эквивалентна команде `\operatorname` из пакета `amsmath`.

`\tg`, `\ctg`, `\arctg`, `\arcctg` реализуют одноименные математические операции. Это весь дополнительный набор математических операций, имеющихся в новом макросе `NCC` (другие операции легко сделать с помощью команды `\mop`). В стиле `russian` определяются и другие математические операции (подробности см. в описании опции `russian` пакета `babel`).

Команда `\bm{формула}` (ожирение математической формулы) старого макроса `NCC` не реализована в новом варианте, поскольку в  $\LaTeX$ е имеется соответствующий пакет. Для использования команды `\bm` подгрузите пакет `bm`:

```
\usepackage{bm}
```

Поскольку в макросе `NCC` используется пакет `amsmath`, то при наборе математических формул можно использовать все средства этого пакета. Вот часть из них:

`\text{текст}` используется для оформления текста в формулах (вместо команды `\mbox`). Ее основная особенность – скалируемость. Например, текст в индексах будет иметь размеры, соответствующие стилю `\scriptstyle`.

`\displaybreak[число]` принудительный разрыв страницы после строки выключной формулы, в которой эта команда используется. Необязательное *число* в диапазоне 0–4 задает силу разрыва как в команде `\pagebreak`. Ее можно использовать в любой выключной формуле, где эта команда имеет смысл. В частности, можно устроить разрыв страницы внутри группы формул `eqnarray` (в `amsmath` это было не возможно, но в макросе `NCC` структура `eqnarray` переопределена в соответствии с другими структурами подготовки выключных формул). Команда `\displaybreak`, примененная внутри `\eqalign` приведет к разрыву страницы после завершения этой группы уравнений. Ее также можно применять внутри одиночной выключной формулы (`\eq` или ее эквивалента), чтобы сделать разрыв после нее.

`\allowdisplaybreaks[число]` управляет разрывами между формулами группы выключенных формул. Необязательное *число* в диапазоне 1–4 задает силу разрешения на разрыв страницы после формулы в группе: чем больше это число, тем нежелательнее разрывы между формулами в группе. Число 0 означает запрет разрывов между формулами в группе. При этом необходимо явно указать `\displaybreak`, чтобы разрешить разрыв страницы. В макросе NCC используется режим `\allowdisplaybreaks[4]`.

`subequations` – эта структура замораживает номер счетчика уравнений и включает буквенную нумерацию внутренних уравнений, т. е. реализует подуравнения. Внутри этой структуры все подуравнения будут нумероваться в виде (5а), (5б), (5в) и т. д. Метка сразу после `\begin{subequations}` будет ссылаться на номер базовой формулы (т. е. на номер 5), а метка внутри выключной формулы ссылается на номер подуравнения (т. е. на номер 5а).

`\boxed{формула}` рисует рамку вокруг формулы. Например, формула  $a = b + c$  получена с помощью `\boxed{a=b+c}`.

`\frac{верх}{низ}`, `\dfrac{верх}{низ}` оформляют дроби в конкретной математической моде: первая – в `\textstyle`, вторая – в `\displaystyle`.

`\binom{верх}{низ}`, `\tbinom{верх}{низ}`, `\dbinom{верх}{низ}` суть эквиваленты команды `{верх \choose низ}`, использовавшейся для оформления биномиальных коэффициентов.

`\varliminf`, `\varlimsup` следует использовать вместо `\limi` и `\lims` из старого макроса NCC. Эти команды рисуют черту под и над оператором `lim` соответственно.

`\iint`, `\iiint`, `\iiiint`, `\idotsint` задают кратные интегралы: двойной, тройной, четверной, и с многоточиями между интегралами. Вместо команд старого макроса NCC `\dint` и `\tint` следует использовать `\iint` и `\iiiint` соответственно.

Команды `\cases`, `\matrix`, `\pmatrix`, пришедшие из Plain TeX, в пакете **amsmath** заменены на аналогичные структуры и их набор существенно расширен. При переходе со старого макроса необходимо использовать структурный вариант этих команд. Например, команду `\cases` надо заменить на структуру `cases` следующим образом:

```
\cases{... & ... \cr ... & ...} →
\begin{cases} ... & \text{...} \\ \end{cases}
```

Из этого примера видны отличия структуры от команды. В старой команде второй столбец оформлялся в текстовой моде, а в новой структуре – в математической моде. Переход к новой строке выполняется теперь с помощью команды `\\` [расстояние], а не `\cr`.

В пакете **amsmath** введены следующие структуры для оформления матриц с разными вариантами обрамления: `matrix` задает “чистую” матрица без обрамления; `pmatrix`, `bmatrix`, `Vmatrix`, `vmatrix`, `Vmatrix` задают матрицы с обрамлением `( )`, `[ ]`, `{ }`, `| |`, `|| ||` соответственно. Число столбцов в таких матрицах не более 10 (его можно увеличить, изменив значение ответственного за это счетчика `MaxMatrixCols`). Оформление этих матриц очевидное: как в структуре `array`, но параметр спецификации столбцов отсутствует.

Имеется также структура `smallmatrix`, оформление полей в которой выполняется мелким математическим стилем в зависимости от места ее использования. Например, матрица  $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$  оформлена следующим образом:

```
\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)
```

Внутри матриц можно использовать команду `\hdotsfor[множитель]{число столбцов}`, позволяющую напечатать многоточие в указанном числе столбцов. Необязательный параметр *множитель* задает во сколько раз больше сделать разрядку между точками по сравнению с

обычной разрядкой. Так множитель 1.5 означает увеличение расстояния между точками в 1.5 раза.

Для оформления многострочных индексов в суммах (внутри подындкса и надындкса) используются команда `\substack{формулы}` и структура `\begin{subarray}{align} формулы \end{subarray}`. Последняя есть обобщение первой. Параметр *align* может принимать значения *l*, *c*, *r*, означающие левостороннее, центрированное и правостороннее выравнивание соответственно.

В определение команды `\smash` в **amsmath** добавлен необязательный параметр: `\smash[tb]{формула}`, в котором можно указать что именно необходимо обнулить: высоту формулы (*t*) или ее глубину (*b*).

Здесь приведены далеко не все возможности пакета **amsmath**. Более полное описание можно найти в файле `amsltdoc.dvi`, находящемся в директории `texmf/doc/latex/amsmath`.

### 4.3. Теоремы

Принципы оформления математических утверждений полностью перешли из старого макроса, но сделаны более совместимыми со стандартными средствами. Наряду с командой `\newtheorem` имеется команда `\renewtheorem`, позволяющая переопределять уже определенные математические утверждения. С помощью *\**-формы этих команд определяются математические утверждения стиля `apar` (номер перед названием). Предопределены два типа вывода математических утверждений, называемые `theorem` и `remark`. Утверждение первого типа выводится курсивом, а второго – прямым шрифтом. Можно вводить новые типы и переопределять имеющиеся.

Имеется 8 предопределенных математических утверждений в трех стилях: автоматическая нумерация, ручная нумерация, нумерация в стиле `apar`. Каждое математическое утверждение можно оформлять любым из двух способов: в виде `\команда ... \qef` или в виде `\begin{команда} ... \end{команда}`.

**Теоремы с автоматической нумерацией** приводятся здесь в структурном стиле:

<code>\begin{theorem}[Комментарий]</code>	<i>тело</i>	<code>\end{theorem}</code>
<code>\begin{lemma}[Комментарий]</code>	<i>тело</i>	<code>\end{lemma}</code>
<code>\begin{proposition}[Комментарий]</code>	<i>тело</i>	<code>\end{proposition}</code>
<code>\begin{corollary}[Комментарий]</code>	<i>тело</i>	<code>\end{corollary}</code>
<code>\begin{definition}[Комментарий]</code>	<i>тело</i>	<code>\end{definition}</code>
<code>\begin{statement}[Комментарий]</code>	<i>тело</i>	<code>\end{statement}</code>
<code>\begin{example}[Комментарий]</code>	<i>тело</i>	<code>\end{example}</code>
<code>\begin{remark}[Комментарий]</code>	<i>тело</i>	<code>\end{remark}</code>

Четыре первых утверждения оформляются как теоремы, а последние – по типу замечаний. Счетчики всех утверждений совпадают с названиями структур и являются динамическими. Стиль их нумерации устанавливается командой `\countstyle` в преамбуле документа:

```
\countstyle{базовый счетчик}
```

Если базовый счетчик указан, то все динамические счетчики ему подчиняются и имеют составное имя при печати. Если базовый счетчик отсутствует, то динамические счетчики имеют простую нумерацию. По умолчанию в стилях `article` и `preprint` установлен `\countstyle{}`, а в стилях `book` и `report` – `\countstyle{chapter}`.

Для конкретных счетчиков можно указать стиль, отличающийся от общего. Для этого надо их указать в дополнительном параметре команды `\countstyle` (это список счетчиков через запятую). Например, команда

```
\countstyle[equation]{section}
```

заказывает посекционную нумерацию уравнений.

**Теоремы с ручной нумерацией** приводятся здесь в структурном стиле:

```

\begin{Theorem}{Номер}[Комментарий]      тело \end{Theorem}
\begin{Lemma}{Номер}[Комментарий]        тело \end{Lemma}
\begin{Proposition}{Номер}[Комментарий]   тело \end{Proposition}
\begin{Corollary}{Номер}[Комментарий]     тело \end{Corollary}
\begin{Definition}{Номер}[Комментарий]    тело \end{Definition}
\begin{Statement}{Номер}[Комментарий]     тело \end{Statement}
\begin{Example}{Номер}[Комментарий]       тело \end{Example}
\begin{Remark}{Номер}[Комментарий]        тело \end{Remark}

```

Если параметр *Номер* пуст, то печатается утверждение без номера. Тип оформления этих утверждений аналогичен предыдущему.

Если необходимо оформить нестандартное утверждение как теорему, то можно воспользоваться командой `\liketheorem`. Аналогично, для оформления утверждения как замечания имеется команда `\likere remark`. Их синтаксис:

```

\liketheorem{Заголовок}{Номер}[Комментарий]  тело \qef
\likere remark{Заголовок}{Номер}[Комментарий]  тело \qef

```

**Теоремы стиля arar** приводятся здесь в командном стиле:

```

\atheorem[Комментарий]      тело \qef
\alemma[Комментарий]        тело \qef
\aproposition[Комментарий]   тело \qef
\acorollary[Комментарий]     тело \qef
\adefinition[Комментарий]    тело \qef
\astatement[Комментарий]     тело \qef
\anexample[Комментарий]      тело \qef
\aremark[Комментарий]       тело \qef

```

Все эти утверждения нумеруются с помощью динамического счетчика `arar`. Для оформления нестандартных утверждений в этом стиле используются команды

```

\liketheorem*{Заголовок}[Комментарий]  тело \qef
\likere remark*{Заголовок}[Комментарий]  тело \qef

```

Имеется еще команда `\arar`, позволяющая начать новый нумерованный параграф с заданным заголовком или без него:

```
\arar[Заголовок]
```

**Оформление доказательств** выполняется с помощью команды `\rproof[Комментарий]`. Завершается доказательство обычно командой `\qed`. Например, команда

```
\rproof[теоремы 1] Некоторый текст \qed
```

выглядит так:

**Доказательство теоремы 1.** Некоторый текст □

Команда `\qed` рисует квадрат, прижатый к правому краю страницы. Это квадрат задается командой `\qedsymbol`, которую можно использовать, например, в поле номера в выключной формуле, если доказательство заканчивается ею.

## 5. Рубрикация

Прежде всего отметим команды задания стиля вывода рубрик документа. Их действие распространяется на все выключные рубрики документа (т. е. на `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`). Стиль вывода этих рубрик определяется командой

```
\sectionstyle{стиль}
```

где *стиль* – один из следующих вариантов:

`hangindent` – стандартный стиль L<sup>A</sup>T<sub>E</sub>X.

`hangindent*` – аналогично предыдущему, но без правого выравнивания.

`parindent` – рубрики выключены на величину параграфного отступа.

`parindent*` – аналогично предыдущему, но без правого выравнивания.

`center` – рубрики центрируются.

Команды `\indentaftersection` и `\noindentaftersection` управляют отступом после секций. Первая разрешает параграфный отступ, вторая – запрещает. По умолчанию используется стиль `hangindent*` без параграфного отступа после секций.

По сравнению со старой версией макроса удалены команды ручной рубрикации (начинающиеся с большой буквы). Другими словами, в новой версии макроса есть только команды стандартного оформления рубрик документа. Однако возможности этих команд существенно расширены с помощью команд модификации, которые помещаются непосредственно перед рубрикой, которую следует вывести несколько иначе.

- Если Вы не хотите помещать текст рубрики в колонтитул, напишите `\norunninghead` перед командой рубрикации.
- Если Вы хотите заменить текст, поступающий в колонтитул, на другой текст, используйте команду `\runninghead{другой текст}`.
- Если Вы хотите сделать нумерованную рубрику, которая появится в колонтитуле и содержании документа, воспользуйтесь модификатором `\noheadingtag`.
- Если Вы хотите оформить рубрику с нестандартным номером и поместить ее в колонтитул и содержание, воспользуйтесь командой `\headingtag{другой номер}`. Счетчик рубрики в этом случае не изменяется.
- Если Вы хотите полностью подменить оформление номера рубрики на собственное оформление, воспользуйтесь командой `\headingtag*{другой номер}`. При этом текст рубрики не будет поступать в колонтитул и содержание.
- Если Вы хотите подавить запись рубрики в содержание, используйте модификатор `\skipwrittingtoaux`.

Модификаторы, связанные с оформлением номера рубрики, действуют и на команду `\caption`. Более того, команда `\caption` имеет \*-форму как и другие команды рубрикации.

## 6. Рисунки и таблицы

В стандартных классах L<sup>A</sup>T<sub>E</sub>X плавающее оформление рисунков и таблиц выполняется в структурах `figure` и `table`. Эти структуры обеспечивают плавающее размещение, но не задают стиль оформления. В макросе NCC стиль оформления задается специальными средствами в основном с помощью надстройки дополнительных команд над командами плавающего оформления.

## 6.1. Базовые команды

Команда `\FloatingStyle{style}` определяет стиль оформления рисунков и таблиц (мы называем их в дальнейшем плавающими элементами, хотя далеко не всегда к ним применяется плавающее оформление). По умолчанию в макросе NCC установлен стиль

```
\FloatingStyle{\footnotesize\centering}
```

Базовые команды `\minifig` и `\minitabl` обеспечивают подготовку материала в виде мини-страницы и разрешают использование команды `\caption` внутри. Синтаксис этих команд соответствует расширенному синтаксису команды `\parbox`:

```
\minifig[pos][высота][inner-pos]{ширина}{тело}
\minitabl[pos][высота][inner-pos]{ширина}{тело}
```

Параметр *pos* задает вертикальное выравнивание (*t*, *b* или *c*) по отношению к окружающему тексту; параметр *высота* задает требуемую высоту мини-страницы; параметр *inner-pos* определяет вертикальное выравнивание содержимого внутри мини-страницы (*t*, *b*, *c* или *s*); наконец, параметр *ширина* задает ширину мини-страницы. *Тело* плавающего элемента оформляется в соответствии с заданным стилем и может содержать команду `\caption` внутри.

Все другие команды плавающего оформления в макросе NCC базируются на `\minifig` и `\minitabl`.

## 6.2. Оформление с обтеканием

Наиболее сложным является оформление плавающих элементов с “обтеканием” текстом. Хотя существуют пакеты, обеспечивающие обтекание, но они не дают полного контроля над этим процессом. Именно поэтому в макросе NCC реализованы средства ручного оформления. В старом макросе приходилось реализовывать обтекание рисунка с помощью `\minifig` и команды `\parbox`. Это было довольно утомительно. В новом макросе имеются специальные средства для облегчения этого процесса.

По правилам российской полиграфии в режиме обтекания рисунок должен располагаться на внешней стороне страницы (слева – на четной и справа – на нечетной странице при двухсторонней печати).

Такое размещение обеспечивается следующими командами:

```
\sidefig[pos](w1)(w2){рисунок}{текст}
\sidefig*[pos](w1)(w2){рисунок}{текст}
\sidetabl[pos](w1)(w2){таблица}{текст}
\sidetabl*[pos](w1)(w2){таблица}{текст}
```

Команды без звездочки анализируют номер текущей страницы и текущий режим оформления документа (*twoside* или *oneside*, *twocolumn* или *onecolumn*). В случае двухстороннего одностолбцового оформления они выводят плавающий элемент на внешней стороне страницы, а в остальных случаях – справа. Команды со звездочкой выводят плавающий элемент на противоположной стороне. Они нужны, поскольку после анализа текущего номера страницы может оказаться, что в процессе форматирования этот элемент перейдет на следующую страницу и его оформление окажется неверным.

В процессе оформления текстового элемента в этих командах вставляется параграфный отступ, если команда начинает новый параграф. В начало и конец текста вставляются подпорки `\strut`, чтобы обеспечить нормальный интерлиньяж со строками текста, обтекающими такую команду сверху и снизу.

Все параметры в квадратных и круглых скобках необязательны. Их смысл:

*pos* определяет вертикальное выравнивание плавающего элемента относительно текста (*t*, *b* или *c*; по умолчанию *c*). Может также содержать дополнительные буквы: *j* означает растяжку последней строки текста (используется, если параграф продолжается вне элемента с обтеканием), *n* отменяет вставку подпорок `\strut`;

*w*<sub>1</sub> задает ширину плавающего элемента (число) в миллиметрах (в единицах `\unitlength` которые в макросе `NCS` равны миллиметрам);

*w*<sub>2</sub> задает ширину текстового бокса в миллиметрах.

Если оба параметра ширины опущены, то ширины текста и плавающего элемента будут равны  $(\text{\linewidth}-1.5\text{em})/2$ . Если *w*<sub>2</sub> отсутствует, то ширина текстового бокса вычисляется как  $\text{\linewidth}-w_1\text{\unitlength}-1.5\text{em}$ . Суммарная ширина рисунка и плавающего элемента могут превышать ширину страницы. В этом случае будет наложение этих боксов.

При подготовке плавающих элементов с обтеканием иногда требуется знать где именно окажется плавающий элемент – слева или справа и выполнять несколько различное оформление в разных случаях. Команда

```
\ifleftsidefloat{слева}{справа}
```

дает такую информацию. Она исполняет параметр `{слева}`, если плавающий элемент оказался слева и `{справа}` в противном случае. Эта команда, естественно, используется внутри команд оформления с обтеканием.

### 6.3. Плавающее оформление рисунка или таблицы

```
\fig[placement](w){рисунок}
\fig*[placement](w){рисунок}
\tabl[placement](w){таблица}
\tabl*[placement](w){таблица}
```

Параметр *placement* задает предпочтительные варианты размещения плавающего элемента на странице. По умолчанию используется размещение `htp`. Необязательный параметр (*w*) задает ширину плавающего элемента в миллиметрах (вещественное число). При этом соответственно уменьшается ширина подрисуночной подписи в команде `\caption`.

Команды `\fig` и `\tabl` суть надстройки над структурами `figure` и `table`. Их *\**-формы являются надстройками над *\**-формами этих структур, т. е. используются при оформлении плавающих элементов, занимающих на всю ширину страницы в двухстолбцовой моде.

### 6.4. Плавающее оформление пары рисунков или таблиц

```
\figs[placement](w1)(w2){рисунок 1}{рисунок 2}
\figs*[placement](w1)(w2){рисунок 1}{рисунок 2}
\tabls[placement](w1)(w2){таблица 1}{таблица 2}
\tabls*[placement](w1)(w2){таблица 1}{таблица 2}
```

Эти команды размещают пару плавающих элементов рядом. Параметр *placement* имеет тот же смысл, что и выше, а параметры (*w*<sub>1</sub>) и (*w*<sub>2</sub>) задают ширину левого и правого плавающего элементов. Если один из них или оба этих параметра опущены, то ширина оставшихся элементов вычисляется по тем же правилам, что и в командах оформления с обтеканием, только расстояние между плавающими элементами будет `1em`, а не `1.5em`. Наложение плавающих элементов также возможно, если суммарная их ширина превышает ширину текстового поля страницы.



## 7. Графика

Вставка графических объектов в макросе NCC реализована на базе стандартного пакета **graphicx**. Для основных типов просмотрщиков в макросе заданы списки графических расширений, которые они понимают. Это позволяет не указывать расширение файла в команде `\includegraphics`, и обеспечивает трансляцию документа разными трансляторами без изменения его содержимого. Необходимо только чтобы для каждого транслятора была подготовлена версия графического файла, который он понимает.

Речь прежде всего идет о просмотрщиках `dvips`, `Yap` и трансляторе `pdflatex`. Транслятор `pdflatex` понимает графические форматы `.png`, `.mps`, `.tif`, `.pdf`, `.jpg`. Драйверы `dvips` и `Yap` понимают форматы `.eps`, `.ps`, `.bmp`, `.msp`, `.psx`, `.pict`, `.pntg` (лучше использовать первых 4 формата, в частности, `.eps` или `.bmp`, поскольку в остальных случаях просмотрщик `Yap` “тормозит”, так как на лету конвертирует рисунки в `bmp`).

В макросе NCC установлено название директории, в которой предпочтительно размещать все рисунки. Рекомендуется помещать их в поддиректорию `graphics` относительно директории, в которой находится транслируемый документ.

Для управления вставкой графиков в макросе NCC имеются следующие команды:

`\pic{filename}` загружает файл с именем `filename.pic` из поддиректории `graphics`. В старой версии макроса NCC `.pic` файлы либо содержали картинки, подготовленные в TeXCADe, либо команду `\putimage` вставки графического объекта.

`\putimage(x,y)[x_r,y_r](x_s,y_s){filename}` эмулирует команду вставки графиков в стиле старого макроса NCC. Более полное описание команды находится в пакете **nccpic**.

`\draftgraphics`, `\finalgraphics` переключают режим вывода графических объектов. В режиме `\draftgraphics` вместо графического объекта рисуется прямоугольник нужных размеров, а в режиме `\finalgraphics` графический файл вставляется в текст документа. Отметим, что опции `draft` и `final` команды `\documentclass` включают режимы `\draftgraphics` и `\finalgraphics` соответственно. Именно поэтому последние команды были добавлены в макрос NCC, чтобы можно было более гибко управлять режимом вывода графических объектов. Отметим также, что в драфтовом режиме команда `\putimage` не проверяет существует ли указанный графический файл или нет.

*Примечание: Если Вы планируете конвертировать документ в формат pdf и хотите нарисовать рисунок в TeXCADe, то не используйте режим `emlines` в нем, поскольку транслятор `pdflatex` не понимает графики `emTeX`.*