

Содержание

1	Советы по оформлению	2
1.1	Основные рекомендации и советы	2
1.2	Формулы	4
1.2.1	Основы	4
1.2.2	Советы	5
1.3	Таблицы и рисунки	9
1.3.1	Основы	9
1.3.2	Советы	10

Примечание

Материалы созданы для помощи студентам кафедры ММП. Дата создания: 2025 год.

1. Советы по оформлению

Многие советы взяты из практики и книги [1].

1.1 Основные рекомендации и советы

Ниже представлены основные рекомендации, которые могут быть полезны при работе с TeX'ом и сделают вашу работу красивее!

1. Нужно структурировать большой проект по файлам¹. Вы можете каждую секцию хранить в отдельном файле. Например, `introduction.tex`, `chapter1.tex`, `chapter2.tex` и так далее. Это упрощает навигацию и работу в команде. Тогда ваш документ будет выглядеть так:

```
\begin{document}
  \input{introduction.tex}
  \input{chapter1.tex}
  \input{chapter2.tex}
\end{document}
```

2. Используйте пакет `hyperref` (с настройкой `colorlinks=true`), чтобы все перекрестные ссылки в PDF становились кликабельными.
3. Для более аккуратной работы с переносом строк стоит изучить `\sloppy` и другие режимы работы с пробелами в строках. Подробнее по этой [ссылке](#).
4. Для цитирования используйте пакет **BibTeX**, все источники храните в файле `.bib`. Подробнее тут https://www.overleaf.com/learn/latex/Bibliography_management_with_bibtex. **BibTeX** почти всегда следует получать из <https://scholar.google.com/> и почти никогда не следует писать самостоятельно.
5. Для цитирования литературы используйте только `\cite{}` или `\citet{}` или `\citer{}`. Для цитирования в тексте лучше использовать `\citep{}`. Часто поведение цитирования задается файлом стиля (обычно это файл с расширением `.sty`), который у каждого журнала/конференции свой. Подробнее про цитирование тут https://ru.overleaf.com/learn/latex/Natbib_citation_styles.
6. Для отчетов и статей высокого уровня **никогда не использовать** вертикальные линии и `\hline` в таблицах. `\toprule`, `\midrule`, `\bottomrule` — золотой стандарт. Обычно `\hline` и особенно вертикальные линии выглядят **некрасиво!** Подробнее про таблицы в секции 1.3.

¹Каждый решает сам, но если вы пишете конспект лекций и работаете параллельно — бывает очень полезно.

7. Используйте **длинное** тире --- как тире в русском языке. Например, Земля — это планета. Вместо: Земля - это планета.
8. После кавычек " нужно добавить ~. Иначе пропадет пробел после закрывающей кавычки. Например: Книга "Стража! Стража!" очень интересная. Такие кавычки в целом лучше **не использовать!**
 Для кавычек в **русском языке** лучше использовать << >>. Например: Книга «Стража! Стража!» очень интересная.
 Для кавычек в **английском языке** лучше использовать ‘ ‘ ’ ’. Например: Книга “Стража! Стража!” очень интересная.
9. pandas умеет **конвертировать** pd.DataFrame в TeX. Подробнее по ссылке https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_latex.html.
10. В TeX’e можно рисовать диаграммы графики и так далее, но делать это нужно кодом и не всегда удобно. Подробно вы можете об этом почитать в книге [1]. Мы рекомендуем использовать специальные редакторы, например, [drawio](#) или [mathcha](#). Если вы видите в работах красивую диаграмму или что-то визуально симпатичное, то скорее всего, это можно сделать в этих редакторах. Ещё они поддерживают математический режим с TeX’ом формул.
11. Чтобы вместить текст в лимит по страницам, можно использовать \vspace. Можно заменить окружение *itemize* на • \$\bullet\$ с \vspace{-...pt} и так далее. **Использовать осторожно!** Менее агрессивно — посмотреть в сторону пакета *enumitem*.
12. Если вам нужно выбрать цвета графиков, а у вас очень плохо со вкусом, то лучше использовать шпаргалки по выбору цветов, например <https://infogra.ru/design/krutaya-shpargalka-po-sochetaniyu-tsvetov>.
13. Если вам очень хочется, то вы можете обернуть плавающий объект текстом с помощью пакета *wrapfig*. Код для примера ниже:

```
\begin{wrapfigure}{l}{0.2\textwidth}
\centering
\includegraphics[width=1\linewidth]{Mmp_logo.png}
\caption{Это лого ММП}
\end{wrapfigure}
```

Текст, цель которого показать, как выглядит текст, который содержится вокруг фигуры.



Рис. 1: Это лого ММП

Дальше текст, сгенерированный через \lipsum[1] Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.2 Формулы

1.2.1 Основы

1. Для набора формул существует математический режим, например чтобы набрать $\int_0^{2\pi} \frac{\sin x}{x} dx$, то нужно ввести `\int_0^{2\pi}\{\frac{\sin x}{x}\}dx \sin x`. Можно ещё написать `\(\int_0^{2\pi}\{\frac{\sin x}{x}\}dx \sin x\)`. Вариант с `$ $` — более старый и имеет один недостаток: начало конструкции совпадает с её концом.
2. Для набора формул в отдельной строке можно использовать окружение *equation*:

```
\begin{equation}
\int_0^{2\pi}\{\frac{\sin x}{x}\}dx. \label{eq:tag}
\end{equation}
```

$$\int_0^{2\pi} \frac{\sin x}{x} dx. \quad (1)$$

Если вам необходимо убрать тег в выражении **1**, то стоит добавить суффикс `*` к названию окружения *equation* или:

```
\begin{equation}
\int_0^{2\pi}\{\frac{\sin x}{x}\}dx. \notag
\end{equation}
```

$$\int_0^{2\pi} \frac{\sin x}{x} dx.$$

Правило с тегом работает для многих окружений, например *align*, *gather* и другие.

3. Для набора формул в несколько строк можно использовать *align*:

```
\begin{align}
&x = 10, \\
&y = 100, \\
&z = 1000.
\end{align}
```

$$x = 10, \quad (2)$$

$$y = 100, \quad (3)$$

$$z = 1000. \quad (4)$$

Мы используем `&` для выравнивания. Можно использовать окружение *gather* для центрирования:

```

\begin{gather}
x = 10, \\
y = 100, \\
z = 1000.
\end{gather}

```

$$x = 10, \tag{5}$$

$$y = 100, \tag{6}$$

$$z = 1000. \tag{7}$$

4. Можно использовать окружение *split*, оно действует как *align*, но тег один.

```

\begin{equation} \label{eq1}
\begin{split}
A &= \frac{\pi r^2}{2} \\
&= \frac{1}{2} \pi r^2
\end{split}
\end{equation}

```

$$\begin{aligned}
A &= \frac{\pi r^2}{2} \\
&= \frac{1}{2} \pi r^2
\end{aligned}
\tag{8}$$

5. Примеры различных отступов в математическом режиме:

<code>x\negthickspace x</code>	<code>xx</code>
<code>x\negmedspace x</code>	<code>xx</code>
<code>x\!x</code>	<code>xx</code>
<code>xx</code>	<code>xx</code>
<code>x\,x</code>	<code>xx</code>
<code>x\:x</code>	<code>xx</code>
<code>x\;x</code>	<code>xx</code>
<code>x\quad x</code>	<code>x x</code>
<code>x\qquad x</code>	<code>x x</code>

1.2.2 Советы

1. Знаки препинания в формулах: читайте их, как часть текста, даже если формула выносится в отдельную строку. Если идет несколько формул подряд или в фигурных скобках случаи перебираются, стоит ставить ; в конце пункта и точку в конце, если дальше начинается новое предложение. Если предложение не заканчивается, то ставить запятую или соответствующий знак препинания.

Пример плохой: Далее мы решаем задачу:

$$\min_{\theta} \sum_{i=1}^n \mathcal{L}_{\theta}(\mathbf{x}_i) \quad (9)$$

θ — веса нейронной сети, \mathcal{L} — функция потерь.

Пример хороший: Далее мы решаем задачу:

$$\min_{\theta} \sum_{i=1}^n \mathcal{L}_{\theta}(\mathbf{x}_i), \quad (10)$$

где θ — веса нейронной сети, \mathcal{L} — функция потерь.

2. Часто абзац начинается с отступа, но этот отступ не нужен, если предложение не закончено, но прервано формулой. В этом случае после **\$\$ не нужно делать пропуск строки.**
3. Когда вы ссылаетесь на формулу, стоит добавлять неразрывный пробел `~` и написать, что вы ссылаетесь на формулу. Без неразрывного пробела возможны случаи, когда ссылка и формула будут на разных строках. **Это правило касается: формул, таблиц, фигур, ссылок на источники, секции и тд.**

Пример плохой:

Из `\ref{eq:min}` следует, что мы решаем задачу минимизации.

Пример хороший:

Из формулы`~\ref{eq:min}` следует, что мы решаем задачу минимизации.

На английском это бы выглядело так: From Equation`~\ref{eq:min}`. Обратите внимание на **большую** букву.

4. Не бойтесь использовать `\cdot`, это может сделать формулу значительно понятнее и читаемее. Особенно если вы умножаете скаляры на векторы или матрицы.

Плохой пример:

$$\sum c_{ij}(\theta, \phi) \mathbf{f}(x_i, y_j) + \sum a_{ij}(\theta, \phi) \mathbf{g}(x_i, y_j).$$

Хороший пример:

$$\sum c_{ij}(\theta, \phi) \cdot \mathbf{f}(x_i, y_j) + \sum a_{ij}(\theta, \phi) \cdot \mathbf{g}(x_i, y_j).$$

5. Для многоточия в формулах лучше использовать `\ldots` или `\cdots`. Например: $i \in \{1, \ldots, K\}$ или $i \in \{1, \cdots, K\}$.

6. Для оформления текста в формулах используйте `\text{...}`, для подписей к математическим объектам `\mathrm{...}`.

Плохой пример:

$$\min_{\theta} \mathcal{L}_{\text{adv}}(\theta), \theta - \text{weights.}$$

$$\min_{\theta} \mathcal{L}_{adv}(\theta), \theta - weights. \quad (11)$$

Хороший пример:

$$\min_{\theta} \mathcal{L}_{\mathrm{adv}}(\theta), \theta \text{ --- weights}.$$

$$\min_{\theta} \mathcal{L}_{\text{adv}}(\theta), \quad \theta - \text{weights.} \quad (12)$$

7. Используйте команды и создавайте файлы нотаций. Пример:

```
\newcommand{\bx}{\mathbf{x}}
\newcommand{\E}{\mathbb{E}}
```

Тогда вместо $\mathbb{E}x$ $\mathbf{E}x$, вы можете написать $E\ x$, что делает TeX читаемым. Кроме того, TeX выглядит более естественно и осмысленно, что позволит писать его проще и быстрее.

8. Для большей наглядности можно красить формулы $x + f(x)$ `$\textcolor{red}{x + f(x)}$` . Бывает удобно, когда нужно показать что поменялось при переходе от одной формулы к другой.

9. Используйте скобки разного размера $\left(\left(\left(\right.\right.\right.\$ \Biggl(\biggl(\bigl(\$$, так читаемость будет значительно лучше.

10. Использование команд `\left` и `\right` для автоматического подбора размера скобок, что часто лучше, чем вручную выбирать `\big`, `\Big`.

11. Если вы используете **большие операторы**, например, \int, \sum, \prod, \cup , то вы можете переставлять пределы, добавив `\limits`. Это может улучшить читаемость, например, $\sum_{i=1}^n$, $\sum\limits_{i=1}^n$.

12. Для большинства известных математических функций есть зарезервированные названия.

```
\begin{equation}
\sin x, \backslash sin x, \backslash quad log x, \backslash log x
\end{equation}
```

$$\sin x, \sin x, \quad \log x, \log x \quad (13)$$

13. При описании гиперпараметров не забывайте оборачивать текст внутри математического режима в `\text{}`. Гиперпараметры лучше писать не чистым текстом, а с использованием математического режима: 10^{-3} вместо `1e-3`; β_{\max} `\beta_{\text{max}}` вместо `beta_max` и так далее.

14. Иногда бывает удобно использовать `\underbrace{ }_{ }`. Например:

```
\begin{equation}
\min_{\mathbf{w}} \underbrace{\|X\mathbf{w} - \mathbf{y}\|_2^2}_{\mathcal{L}(\mathbf{w})}
\end{equation}
```

$$\min_{\mathbf{w}} \underbrace{\|X\mathbf{w} - \mathbf{y}\|_2^2}_{\mathcal{L}(\mathbf{w})} \quad (14)$$

15. Для скалярного произведения лучше использовать `\langle \rangle` `\rangle` `\langle x, x \rangle`, а не `<>`

16. Можно по-разному писать транспонирование `A^T`, `A^{\top}` `A^T`, `A^{\top}`.

17. Можно использовать `\substack{ }` для записи формул в столбец, бывает удобно для суммирования и других операторов.

```
\begin{equation}
\sum_{\substack{1 \leq i \leq n \\ i \neq j}}
\end{equation}
```

$$\sum_{\substack{1 \leq i \leq n \\ i \neq j}} \quad (15)$$

18. Если вдруг забыли как написать определенный символ или оператор, то всегда можно обратиться к сайтам с распознаванием, например, к [detexify](#).

19. Можно воспользоваться [mathpix](#) для конвертации PDF и изображений в TeX.

1.3 Таблицы и рисунки

1.3.1 Основы

1. Таблицы и рисунки являются плавающими объектами, то есть их расположение вы задаете сами. Следовательно крайне важно и часто очень сложно правильно их расположить.
2. К каждому плавающему объекту нужно указывать подпись (*caption*). Всегда подписывайте рисунки и таблицы так, чтобы по описанию можно было без контекста довольно хорошо понять, что происходит (и просто всегда подписывайте).
3. Для создания таблицы:

```
\begin{table}[h]
  \centering
  \begin{tabular}{cc} % c - center align
    a & b \\
    d & c
  \end{tabular}
  \caption{Тут ваша подпись}
  \label{tab:base} % это для ссылки на caption
\end{table}
```

a	b
d	c

Таблица 1: Тут ваша подпись

4. Для создания фигуры:

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.5\linewidth]{ }
  \caption{Тут ваша подпись}
  \label{fig:base} % это для ссылки на caption
\end{figure}
```

5. Если возможно, то обязательно используйте векторные рисунки, то есть расширения файлов **.pdf**, **.svg**. Это бывает затруднительно и излишне, если на рисунке изображение или очень много точек. В иных случаях почти всегда лучше **.pdf**, **.svg**.
6. Для создания таблиц бывает проще использовать сторонние сайты, например <https://www.tablesgenerator.com/>.

7. Для точного размещения плавающих объектов в квадратных скобках можно указать одну или несколько следующих букв: **t** — разрешение на верху страницы, **b** — разрешение внизу страницы, **p** — на отдельной странице (почти всегда иметь целую страницу с рисунком/таблицей очень плохая идея), **h** — разместить сразу же после появления в исходном тексте. Например, команда `\begin{figure}[hp]` означает, что данный рисунок хотелось бы видеть размещенным непосредственно здесь, а если это не получится, то на новой отдельной странице. Подробнее можно почитать по ссылке https://ru.overleaf.com/learn/latex/Errors/%60!h%27_float_specifier_changed_to_%60!ht%27.

1.3.2 Советы

1. Таблицы можно размещать вместе с рисунками, чтобы не было много пустого места по краям. Для этого лучше использовать окружения *minipage*, *subfigure*, *subtable*.
2. Всегда указывайте ширину таблицы и рисунка, это делается в квадратных скобках, например для рисунков, `\includegraphics[width=.5\linewidth]`, то есть ширина будет 50% от длины строки.
3. Для выравнивания таблиц внутри *minipage* или других окружений, можно использовать `\vspace{10pt}`, который делает вертикальный отступ на 10pt.

```
\begin{figure}[ht]
  \centering
  \begin{subfigure}[t]{0.35\linewidth}
    \vspace{0pt} % Без этого не будет align сверху
    \centering
    \includegraphics[width=1.00\linewidth]{image1.pdf}
    \caption{Caption 1}
    \label{fig:sub1}
  \end{subfigure}
  \hfill
  \begin{subfigure}[t]{0.62\linewidth}
    \vspace{0pt} % Без этого не будет align сверху
    \centering
    \includegraphics[width=1.00\linewidth]{image2.pdf}
    \caption{Caption 2}
    \label{fig:sub2}
  \end{subfigure}
  \caption{Main caption}
  \label{fig:main}
\end{figure}
```

4. По умолчанию таблицы не масштабируются в ширину, то есть возможны случаи, когда таблица вылезает за границы заданные вами или даже за пределы страниц. Для этого стоит обернуть окружение *tabular* в `\resizebox{\textwidth}{!}{ }`. Для примера сравните таблицу 2 с таблицей 3. В общем виде это выглядит так:

```

\begin{table}[h]
  \centering
  \resizebox{\textwidth}{!}{
    \begin{tabular}{...}
      ....
    \end{tabular}
  }
  \caption{...}\label{...}
\end{table}

```

| AA | BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Таблица 2: Плохая таблица

| AA | BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB |

Таблица 3: Хорошая таблица

5. Чтобы не засорять \TeX таблицами удобно описать окружение **tabular** в отдельном файле с удобным названием, например **tables/exp1_lr_loss.tex**. При наличии более 3 больших таблиц это бывает очень удобно. Тогда для отображения таблицы можно сделать:

```

\begin{table}[h]
  \centering
  \resizebox{\textwidth}{!}{
    \input{tables/exp1_lr_loss}
  }
  \caption{...}\label{...}
\end{table}

```

6. Обычно лучший по метрике результат в таблице выделяют **жирным**, второй в списке лучших либо *курсивом*, либо подчёркиванием; так таблицу проще читать.
7. Если после добавления таблицы или рисунка стало много свободного места (по бокам, снизу, сверху), то нужно подумать над расположением и улучшить его.
8. Чем дальше таблица от текста, который на неё ссылается, тем менее, что вероятно кто-то найдет и прочтёт таблицу. Всегда ссылайтесь на таблицу в тексте.
9. Для более красивых таблиц можно использовать `\toprule`, `\midrule`, `\bottomrule` для этого достаточно добавить в документ `\usepackage{booktabs}`. Пример в таблице 4.

```

\begin{table}[]
  \centering

```

```

\begin{tabular}{cc}
\toprule
Method & Accuracy \\ \midrule
KNN & 0.90 \\
Linreg & \textbf{0.95} \\
\bottomrule
\end{tabular}
\caption{Красивая таблица}\label{tab:beatiful}
\end{table}

```

Method	Accuracy
KNN	0.90
Linreg	0.95

Таблица 4: Красивая таблица

10. Пример работы с **minipage**. Вы можете посмотреть на таблицу 5 и рисунок 2 и заметить, как плохо это выглядит. Как минимум рисунок и таблица должны быть выровнены друг относительно друга. Такой пример например в таблице 6 и рисунок 3, для этого мы использовали `\vspace{115pt}`. Ниже код для хорошего примера:

```

\begin{figure}[p]
\centering
\begin{minipage}[t]{0.30\linewidth}
\vspace{0pt}
\centering
\includegraphics[width=\linewidth]{Mmp_logo.png}
\caption{Это лого ММП в хорошем примере}
\label{fig:good2}
\end{minipage}
\hfill
\begin{minipage}[t]{0.4\linewidth}
\vspace{115pt}
\centering
\begin{tabular}{cc}
Method & Accuracy \\ \midrule
KNN & 1.0
\end{tabular}
\captionof{table}{Это таблица в хорошем примере}
\label{tab:good2}
\end{minipage}
\end{figure}

```

11. Иногда бывает что-то проще выразить в виде псевдокода, например, с помощью пакетов *algpseudocode*, *algorithm*. Вы можете посмотреть на Алгоритм 1 и ниже его код:

```
\begin{algorithm}
\caption{Ваш первый алгоритм}\label{alg:cap}
\begin{algorithmic}
\Require  $n \geq 0$ 
\Ensure  $y = x^n$ 
\State  $y \leftarrow 1$ 
\State  $X \leftarrow x$ 
\State  $N \leftarrow n$ 
\While{ $N \neq 0$ }
\If{ $N$  is even}
\State  $X \leftarrow X \times X$ 
\State  $N \leftarrow \frac{N}{2}$  \Comment{This is a comment}
\ElseIf{ $N$  is odd}
\State  $y \leftarrow y \times X$ 
\State  $N \leftarrow N - 1$ 
\EndIf
\EndWhile
\end{algorithmic}
\end{algorithm}
```



Рис. 2: Это лого ММП в плохом примере



Рис. 3: Это лого ММП в хорошем примере

Method	Accuracy
KNN	1.0

Таблица 5: Это таблица в плохом примере

Method	Accuracy
KNN	1.0

Таблица 6: Это таблица в хорошем примере

Algorithm 1 Ваш первый алгоритм бинарного возведения в степень!

Require: $n \geq 0$

Ensure: $y = x^n$

$y \leftarrow 1$

$X \leftarrow x$

$N \leftarrow n$

while $N \neq 0$ **do**

if N is even **then**

$X \leftarrow X \times X$

$N \leftarrow \frac{N}{2}$

else if N is odd **then**

$y \leftarrow y \times X$

$N \leftarrow N - 1$

end if

end while

▷ This is a comment

Благодарности

Материалы оформил Александр Оганов (tg: @welmud). Отдельная благодарность за рекомендации Владимиру Богачеву, Дмитрию Феокистову, Андрею Охотину, Денису Ракитину. Материалы созданы при поддержке команды Практикума на ЭВМ кафедры ММП [github](#).

Список литературы

- [1] Андрей Викторович Столяров. Сверстай диплом красиво: Latex за три дня. <http://www.stolyarov.info/books/pdf/latex3days.pdf>, 2010.