

Верстка научных отчетов в L^AT_EX

Филиппенко Павел, Сибгатуллин Булат

30 января 2022 г.

Содержание

1	И_AT_EX что это такое?	3
1.1	Введение	3
1.2	Преимущества И _A T _E X	4
1.3	Как оно работает???	4
2	Редакторы кода	4
3	Overleaf	5
3.1	Регистрация	5
3.2	Знакомство с интерфейсом	6
4	Начало работы	7
5	Работа с текстом	8
6	Математика	11
7	Рисунки и таблицы	13
8	Ссылки	17

1 ЛАТЭХ что это такое?

1.1 Введение

Любую лабораторную работу или научную статью важно не только сделать правильно и грамотно, но и добиться того, чтобы она была презентабельной и хорошо выглядела. В этом нам поможет культовая среда для верстки ЛАТЭХ. Для оформления лабораторных работ или иных документов многие используют Word или LibreOffice Writer. Однако обе эти программы имеют ряд недостатков и в вопросе верстки научного текста значительно уступают ЛАТЭХ.

Что такое ЛАТЭХ? Согласно Википедии: ЛАТЭХ – наиболее популярный набор макрорасширений (или макропакет) системы компьютерной вёрстки ТЭХ, который облегчает набор сложных документов. Пакет позволяет автоматизировать многие задачи набора текста и подготовки статей, включая набор текста на нескольких языках, нумерацию разделов и формул, перекрёстные ссылки, размещение иллюстраций и таблиц на странице, ведение библиографии и др. Возможно данное определение звучит несколько сложновато. Не забывая голову лишним, можете считать ЛАТЭХинструментом оформления научных отчетов. Более глубокое понимание придет с практикой.

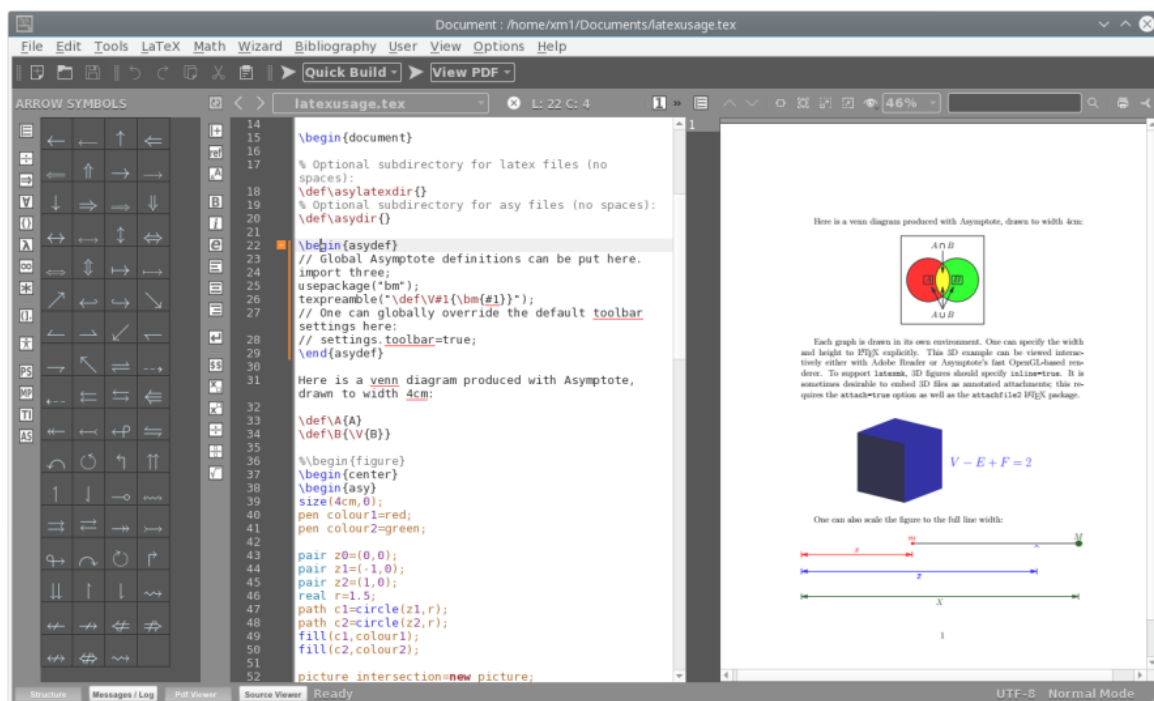


Рис. 1:

1.2 Преимущества \LaTeX

Чем же так хорош \LaTeX ? В целом можно выделить 4 основных преимущества

1. Подгоняет документ ко всем типографским стандартам. Таким образом, можно полностью сконцентрироваться на содержании документа, а не на его оформлении.
2. Автоматизация процессов: автоматическая нумерация разделов и формул, автоматическое составление оглавления и многое другое. Меньше рутины.
3. Открытая среда. Обширное community. В интернете можно найти ответы на любые ваши вопросы, большое количество информации, а так же пакетов, расширений и фич.
4. Кроссплатформенность.

1.3 Как оно работает???

В некотором смысле работа \LaTeX похож на известные вам языки программирования. На вход подается исходный код, затем происходит компиляция и на выходе мы получаем тепленький красивый pdf-документ.

2 Редакторы кода

Для работы с \LaTeX необходимо 2 вещи: специальный компилятор, который будет магическим образом превращать ваши исходники в pdf и редактор кода, в котором вы будете непосредственно работать.



Рис. 2:

Существует большое количество программ, которые представляют собой визуальную графическую среду для создания и редактирования документов \LaTeX . Наиболее распространенным вариантом являются программы `TexMaker` и `TexStudio`. Это кросс-платформенные редакторы \LaTeX с открытым кодом. Данные программы являются интегрированными средами для создания \LaTeX документов и включают такие возможности,

как интерактивная система проверки правописания, сворачивание блоков текста, подсветка синтаксиса и многое другое.

Мы же будем использовать веб-редактор латех документов OverLeaf. Его основное преимущество заключается в том, что данный редактор не требует никакой установки, воспользоваться им можно с любой платформы на любом устройстве с выходом в интернет. Кроме того, этот редактор позволяет нескольким пользователям редактировать один и тот же документ одновременно и просматривать изменения друг друга в режиме реального времени.

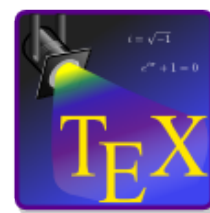


Рис. 3:

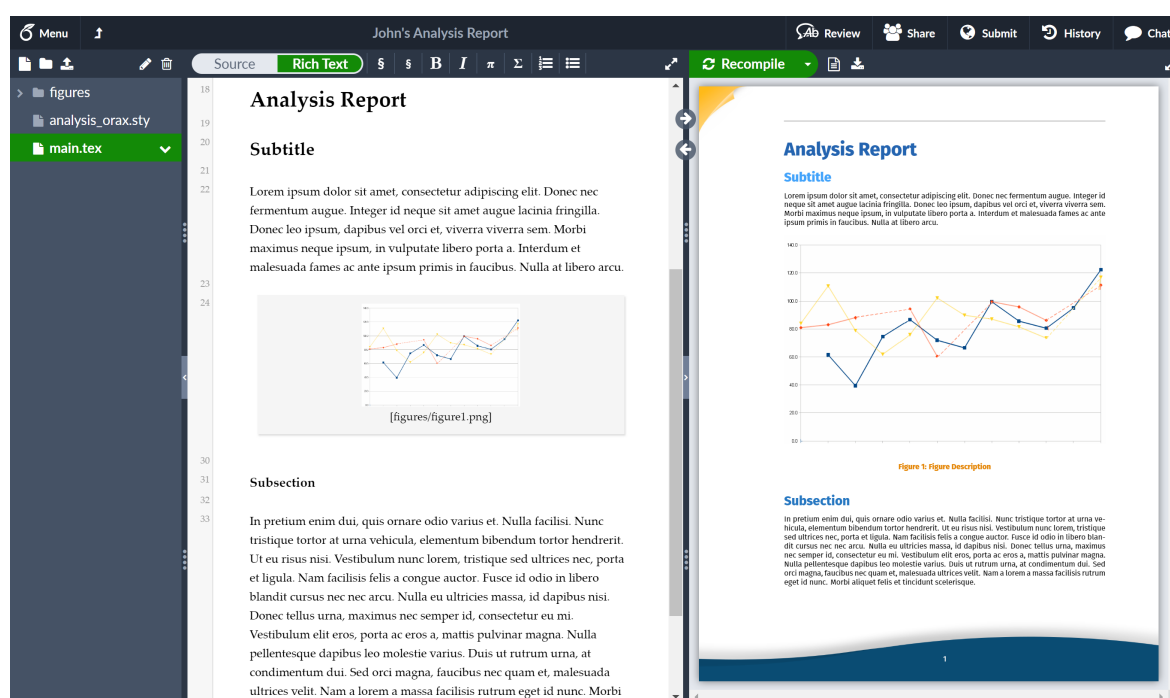


Рис. 4:

3 Overleaf

3.1 Регистрация

Давайте же познакомимся с нашим текстовым редактором. Для этого перейдем на сайт <https://www.overleaf.com/login> и сразу попадем на страницу регистрации.

Чтобы зарегистрироваться, необходимо внизу экрана выбрать пункт Register.

Вводим адрес электронной почты и придумываем пароль. После этого на вас предложат создать ваш первый проект, а на почту придет письмо с просьбой подтвердить адрес. Абсолютно ничего сложного, подтверждаем адрес и создаем наш первый проект Blank Project.

3.2 Знакомство с интерфейсом

После того, как вы создали ваш первый в жизни tex-project вас перенесет в рабочее пространство OverLeaf. Здесь будет проходить большая часть вашей работы.

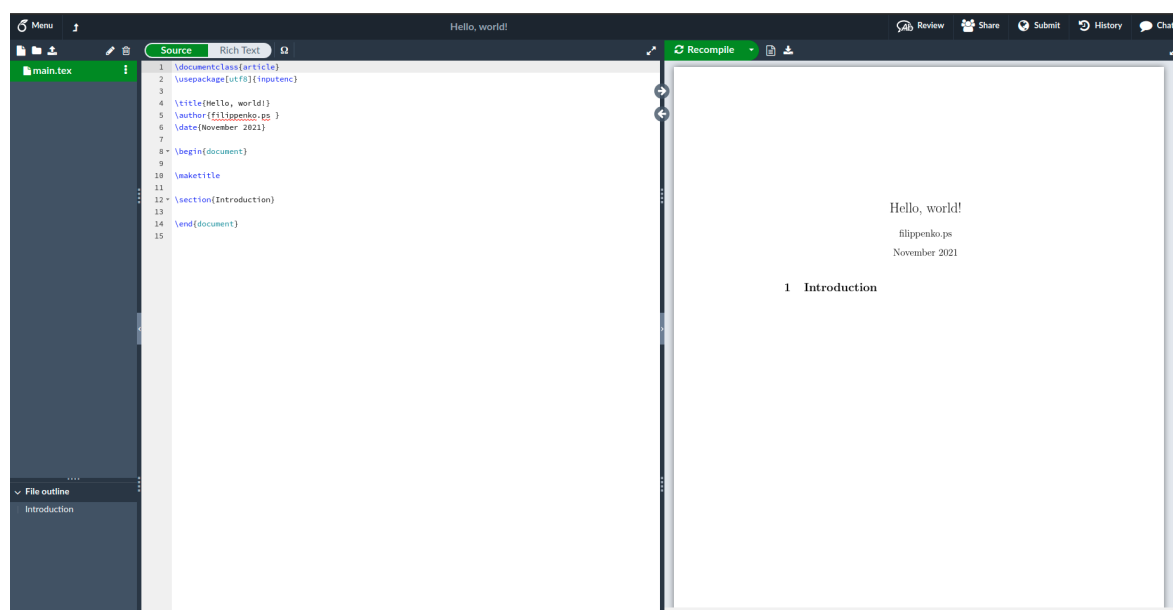


Рис. 5:

Все пространство можно условно разделить на 3 главные смысловые области.

Первая область – рабочая, здесь вы будете писать теховский код, который потом отправится на обработку компилятору.

Вторая область – область визуализации, здесь вы можете посмотреть текущий вид вашего pdf-документа.

Третья область – вспомогательная, в этой области вы можете видеть все файлы и папки вашего текущего проекта, а так же мини-оглавление вашей работы.

Нажав на кнопочку «Menu» на верхней панели слева, вам откроется меню детальной настройки документа (что-то страшное, трогать пока не будем). А нажав на стрелочку рядом, мы выйдем в главное меню управления проектами.

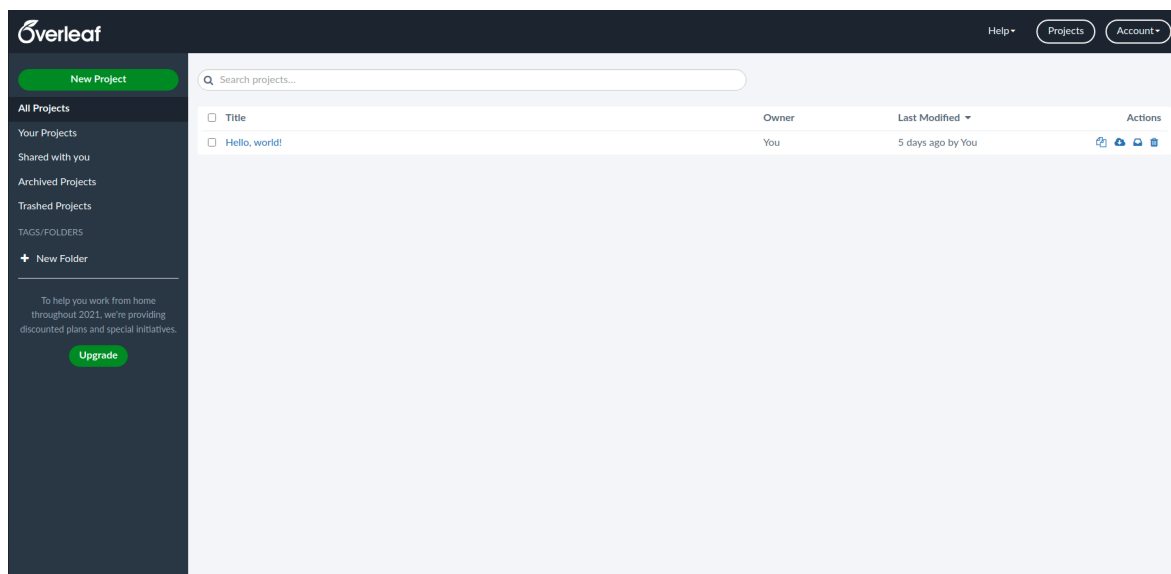


Рис. 6:

В этом меню вам доступно управление вашим аккаунтом, а так же управление всеми вашими латеховскими проектами проектами.

4 Начало работы

Наконец, мы приступаем к написанию нашего документа. Перейдем в наш «проект Hello world!» и в рабочей области удалим весь сгенерированный по умолчанию код.

Перед тем, как наполнять наш документ содержанием и смыслом, необходимо настроить его вид и подключить используемые пакеты.

```
\documentclass[a4paper,12pt]{article}

% Русский язык
\usepackage[T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english,russian]{babel}

% Математика
\usepackage{amsmath,amsfonts,amssymb,amsthm,mathtools}
```

Первой командой мы требуем, чтобы \LaTeX сгенерировал наш документ типа `article` (статья) на бумаге размера A4, со стандартным

шрифтом размера 12. Следующие 3 строчки подключают пакеты для работы с русским языком, а самая последняя строчка подключает математические пакеты. Кстати, как вы могли заметить, значком % обозначаются комментарии. Комментарии помогут вам легче ориентироваться в вашем коде, все что начинается со значка % будет игнорироваться компилятором.

Итак, мы подключили базовые пакеты, без которых невозможна работа в латех. Далее мы подключим еще несколько пакетов, которые понадобятся нам в дальнейшем, а затем перейдем к написанию статьи.

```
\usepackage{graphicx} % импорт изображений
\graphicspath{images/} % папка с картинками
\usepackage{caption}

% центрирование подписи к картинке
\captionsetup{justification=centering}

\usepackage{hyperref} % работа с ссылками

% настроим поля и колонтитулы страницы
\usepackage{geometry} %
\geometry{top=25mm}
\geometry{bottom=35mm}
\geometry{left=35mm}
\geometry{right=20mm}
```

Подготовительная работа закончена и мы можем приступить к написанию документа.

5 Работа с текстом

Наш основной код начинается с окружения

```
\begin{document} % начало документа
\end{document} % конец документа
```

эти строчки обозначают соответственно начало и конец нашего основного документа, именно внутри этого окружения мы будем писать наш основной контент.

Теперь займемся смысловым наполнением нашего документа. Между командами `begin` и `end` напомним нашу первую строчку:

```
\begin{document} % начало документа
Наша первая строчка.
\end{document} % конец документа
```

Для того, чтобы скомпилировать полученный код, нажмем сочетание клавиш **Ctrl + S**, или зеленую кнопку **Recompile** на страничке редактора Overleaf.

В области визуализации pdf-документа можем убедиться, что написанная нами строчка появилась в нашем будущем документе.

Попробуем написать вторую строчку под первой, а затем, заново скомпилировать документ. Здесь мы сталкиваемся с первой проблемой: вопреки нашему желанию вторая строчка напечаталась рядом с первой, а не под ней, так, как это было в исходнике.

<pre>\begin{document} Наша первая строчка. Вторая строчка. \end{document}</pre>	<pre>Наша первая строчка. Вторая строчка</pre>
---	--

Для того, чтобы перейти на новую строку в \LaTeX есть несколько способов

1. Если вы хотите начать новый абзац, то нужно оставить одну пустую строку.

<pre>\begin{document} Наша первая строчка. Вторая строчка. \end{document}</pre>	<pre>Наша первая строчка. Вторая строчка</pre>
--	--

2. Если вы хотите просто перенести текст на следующую строку, не начиная при этом новый абзац, То в конце строки необходимо поставить `\\`.

<code>\begin{document}</code>	
Наша первая строка.\\	Наша первая строка.
Вторая строка.	Вторая строка
<code>\end{document}</code>	

3. Если вы ходите получить вертикальный отступ, какого-то фиксированного размера, вы можете указать этот размер в квадратных скобках, после `\\`.

<code>\begin{document}</code>	Наша первая строка.
Наша первая строка.\\[1 cm]	
Вторая строка.	Вторая строка
<code>\end{document}</code>	

Похожая ситуация в \LaTeX с горизонтальными пробелами. Если между двумя словами, вы поставите 5 пробелов, в конечном документе, между этими словами все равно будет один пробел.

<code>\begin{document}</code>	
The first the second.	The first the second.
<code>\end{document}</code>	

Для того, чтобы сделать горизонтальный отступ заданного размера, применим команду `\hspace{}`

<code>\begin{document}</code>	
The first <code>\hspace{5 mm}</code>	The first the second.
the second.	
<code>\end{document}</code>	

Приведем в табличке некоторые команды оформления текста, поэкспериментировать с ними вы можете самостоятельно.

Стиль оформления	команда	результат
Выделить жирным	<code>\textbf{}</code>	text
Выделить курсивом	<code>\textit{}</code>	<i>text</i>
Подчеркнуть	<code>\underline{}</code>	<u>text</u>
Взять в рамочку	<code>\fbox{}</code>	text

Закончить раздел работы с текстом, мне бы хотелось кратким описанием разделов документа. \LaTeX предлагает удобную иерархию создания разделов, подразделов, подпод разделов и так далее.

Для того, чтобы создать в документе новый раздел необходимо воспользоваться командой $\text{\backslash section}\{\}$, подраздел $\text{\backslash subsection}\{\}$, ну а подподраздел $\text{\backslash section}\{\}$. В фигурных скобочках пишется названия раздела, причем, \LaTeX автоматически нумерует его. При этом, если вы хотите создать нумерованный раздел или подраздел, необходимо поставить символ * после слова section.

$\text{\backslash section*}\{\text{SectionName}\}$

6 Математика

Теперь, когда вы умеете работать с текстом вам нужно понять, как в \LaTeX записываются и оформляются математический выражения. Для выделения формул в \LaTeX используются две конструкции. Если получившиеся выражение вам важно и вы хотите его отметить (чтобы использовать в дальнейшем или по другой причине), то вам нужно использовать такую запись:

$$y = f(x) \tag{1}$$

Вот так будет выглядеть ее код:

```
\begin{equation}
  y = f(x)
\end{equation}
```

Видно, что данное выражение обрaмлено в $\text{\backslash begin}\{equation\} \dots \text{\backslash end}\{equation\}$, так компилятор понимает, что мы будем записывать математические символы. Заметьте, что при такой форме записи \LaTeX самостоятельно записывает номер данного выражения, а следовательно, если вы хотите обратиться к какой-то формуле, уже записанной у вас в документе, вы можете просто записать её номер.

В случае если ваша выражение является результатом промежуточных вычислений и не будет использоваться в дальнейшем, то стоит показать это при помощи второго вида записи:

$$y = f(x)$$

И сделать это можно при помощи `$$`:

$$y = f(x)$$

Также, если вы хотите использовать в тексте какое-либо математическое выражение вам нужно выделить его `$` (теперь только одним):

Some text: e^x . И вот так это выглядит в коде:

Some text: `$e^x`

Теперь посмотрим, как в \LaTeX будут записываться некоторые математические символы:

$$e^x, e^{abc}, e^a bc$$

$$e_x, e_{abc}, e_a bc$$

$$e_{abc}^{xyz}$$

$$e^x, e^{\{abc\}} e^{abc}$$

$$e_x, e_{\{abc\}}, e_{abc}$$

$$e^{\{xyz\}}_{\{abc\}}$$

Здесь вы можете видеть, как в \LaTeX записывается степень и индекс, а также их комбинация. Также обратите внимание на то, что если в степени или индексе больше одного символа, то их нужно брать в фигурные скобки.

Часто бывает нужно (при использовании знака суммы или интеграла), чтобы степень и индекс перешли в пределы (например суммирования или интегрирования). Сделать это мы можем при помощи выражения `\limits`, записанного после нужного нам символа, но перед его степенью и индексом:

$$\int_0^{x_{\text{макс}}}$$

`$$\int\limits^{x_{\text{макс}}}_0$$`

Обратите внимание, что здесь мы использовали русский текст в записи выражения. Для его использования вам нужно будет написать `\text{}` и текст в фигурных скобках. При использовании английского языка можно просто писать на нем без использования `\text{}`.

Стоит отметить, что в \LaTeX часто приходится использовать формулы раной "высоты поэтому здесь есть выражения для подгона высоты скобок под высоту формулы. Посмотрите как она реализована на примере обычной дроби:

$$\left(\frac{x}{y}\right)$$

`$$\left(\frac{x}{y}\right)$$`

Дробь здесь была задана выражением `\frac{ }{ }` (в первой скобке записывается числитель, а во второй знаменатель. За подгон высоты скобок к высоте выражения отвечают надписи `\right` и `\left` перед скобками. Их вы можете использовать с любым видом скобок (фигурными, прямоугольными и т.д.).

На этом блок посвященный математике подходит к концу, и если вы ожидали увидеть тут разбор существующих в \LaTeX символов, то мне стоит вас огорчить, так как этого тут не будет. Использование символов сводится к простому поиску кода символа и не представляет из себя ничего сложного. Сложности обычно вызывают использование символов одновременно со степенями, дробями (в первую очередь из-за скобок), а это мы уже с вами разобрали.

7 Рисунки и таблицы

В составлении отчета важное место занимают различные таблицы с данными, графики, схемы установок и другие таблицы и рисунки,

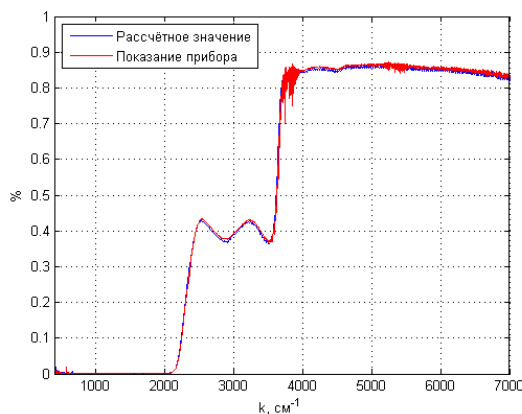


Рис. 7: Коэффициент пропускания стекла

поэтому в \LaTeX реализована работа как с таблицами, так и со вставкой изображений. Для начала рассмотрим изображения:

Вот как выглядит код для такого изображения:

```
\begin{figure}[h!]
  \centering
  \includegraphics[width=0.5\linewidth]{9.png}
  \caption{Коэффициент пропускания стекла}
\end{figure}
```

Посмотрим, что происходит в данном фрагменте кода. Все функции обрамлены `\begin{figure}[h!] ... \end{figure}`, так компилятор понимает, что здесь мы будем вставлять изображение, `h!` в квадратных скобках означает, что изображение должно стоять в первом подходящем месте. Подключение изображения реализуется функцией `\includegraphics` в квадратных скобках при помощи `width` прописывается длина изображения по горизонтали. Ее можно задать например при помощи `*\linewidth`, где `\linewidth` - горизонтальный размер страницы с учетом отступов, а `*` - число, на которое длина этой строки будет умножаться в нашем случае оно равно 0.5, а значит что изображение будет занимать только половину страницы. Имя изображения записывается в фигурных скобках (`{9.png}`), про папку из которой берутся изображение мы говорили вначале методички, если такая папка не задана, то по умолчанию используется папка, в которой хранится файл с кодом.

При помощи `\caption{}` мы задаем описание к изображению, причем номер рисунка перед ним прописывается автоматически. `\centering`

располагает изображение и все его дополнительные свойства по центру (в нашем случае описание).

Если вам нужно расположить рядом несколько графиков, то вы можете сделать расположив их в ряд в одной строке. Для этого можно использовать функцию `minipage`:

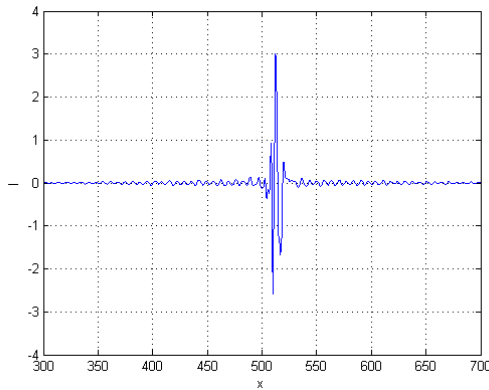


Рис. 8: Экспериментально полученная интерферограмма для пустого канала

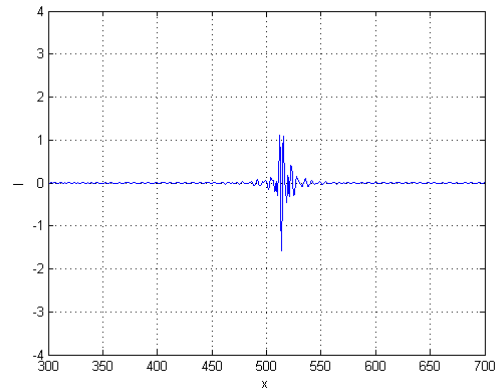


Рис. 9: Экспериментально полученная интерферограмма для стекла

Разберем код данного фрагмента:

```
\begin{figure}[h!]
  \begin{center}
    \begin{minipage}[h!]{0.48\linewidth}
      \includegraphics[width=1\linewidth]{2.png}
      \caption{Экспериментально полученная
        интерферограмма для пустого канала}
    \end{minipage}
    \hfill
    \begin{minipage}[h!]{0.48\linewidth}
      \includegraphics[width=1\linewidth]{3.png}
      \caption{Экспериментально полученная
        интерферограмма для стекла}
    \end{minipage}
  \end{center}
\end{figure}
```

В данном случае `\centering` нам будет недостаточно, так как он располагает по центру только изображение и все что с ним связано, а у нас используется два изображения. Поэтому вместо него мы используем знакомое нам выражение `\begin{center} ... \end{center}`. Далее мы используем `\begin{minipage}[h!]{0.48\linewidth} ... \end{minipage}`. Как понятно из названия `minipage` означает маленькую страницу, горизонтальный размер которой задается уже аргументом во второй фигурной скобке. `\hfill` автоматически определяет нужно делать перенос строки или можно уместить изображение на этой же строчке. Также не ставьте значение горизонтального размера `minipage` равным `1 / (количество изображений на строку)`, так как между мини-страницами существует минимальное расстояние. Если вы ну будете учитывать его компилятор все еще сможете уместить все изображения в одну строку, но центрирование сойдет.

На этом мы заканчиваем работу с изображениями и переходим к таблицам. Вся работа с таблицами сводится к такому коду:

Пример таблицы	Пример таблицы
Пример таблицы	Пример таблицы

Ее код:

```
\begin{center}
  \begin{tabular}{|c|c|}
    \hline
    Пример таблицы & Пример таблицы \\
    \hline
    Пример таблицы & Пример таблицы \\
    \hline
  \end{tabular}
\end{center}
```

Таблица задается `\begin{tabular}{|c|c|} ... \end{tabular}`. Здесь `tabular` дает компилятору понять, что мы имеем дело с таблицей. Символ `c` в `{|c|c|}` имеет смысл столбца, то есть каждая буква `c` это один столбец. То, как столбцы разделяются задается символами `|` между буквами `c`, их количество можно менять, либо вовсе убирать их. `\hline` - горизонтальная линия, разделяющая строки таблицы (их количество

также можно менять, либо вовсе убирать их). Информация о переходе строки передается компилятору при помощи `\\`, а о следующем столбце при помощи `&`. Учитывайте, что если в строке будет недостаточно `&` или в таблице будет недостаточно `\\`, то компилятор выдаст уведомление об ошибке.

Кстати внутри таблицы можно использовать формулы, текст и вставлять изображения.

8 Ссылки

Ссылки в \LaTeX помогают не только переходить на различные сайты, но и могут направить на нужную страницу, формулу, изображение и любой набор символов. Мы уже сталкивались с таким видом ссылок (называемых гиперссылками) в начале, когда оформляли оглавление, рассмотрим как еще можно с ними работать.

Ссылки на страницы редко когда используются, поэтому здесь будут рассмотрено подключение ссылок к изображениям и формулам. Посмотрим на данное изображение:

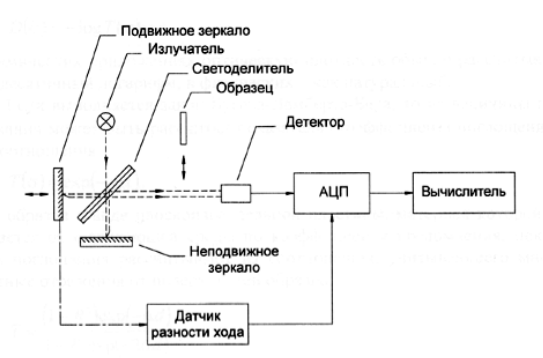


Рис. 10: Схема устройства Фурье-интерферометра

И его код:

```
\begin{figure}[h!]
  \centering
  \includegraphics[width=0.48\linewidth]{1.png}
  \caption{Схема устройства Фурье-интерферометра}
  \label{picture_1}
\end{figure}
```

В коде появилось новое, незнакомое нам выражение:

```
\label{picture_1}
```

`\label{}` задает метку, с именем прописанным внутри фигурных скобок. На это имя теперь можно ссылаться, например так: [1](#). Чтобы это сделать, нужно просто написать такой код:

```
\hyperref[picture_1]{1}
```

Внутри квадратных скобок в данном выражении, вместо `picture_1` может быть имя любой другой метки, а внутри фигурных номер, который вы хотите видеть у данного изображения. Кстати, хочу обратить ваше внимание, что `\label{}` должен идти после подключения изображения, а не после описания к нему, так как в таком случае любые ссылки на него будут переводить вас к описанию.

При создании меток на формулу также используется `\label{}`, но его расположение в коде отличается от такового у изображений:

$$\Delta I = I(x) - I(0) \tag{2}$$

Код к (2):

```
\begin{equation} \label{formula_1}
\Delta I = I(x) - I(0)
\end{equation}
```

Заметьте, что у формул метка находится сразу после `\begin{equation}`, так происходит потому что в обрамлении для формулы обычно записывается только формула и больше ничего. Следовательно удобнее всего ставить метку на обрамление, как на начало формулы. Создание ссылки на такую формулу задается таким выражением:

```
\eqref{formula_1}
```

Здесь надпись в фигурных скобках это имя метки.