

Лабораторная работа №4

Работа с графикой в LaTeX

Абрамян Артём Арменович

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Цель данной работы — изучить возможности включения и манипулирования графикой в LaTeX, освоить различные способы позиционирования изображений, научиться работать с перекрестными ссылками.

2 Задание

Выполнить следующие задания:

- Попробовать включить собственное изображение, заменив стандартные примеры.
- Исследовать возможности параметров `height`, `width`, `angle` и `scale`.
- Использовать параметр `width` для установки размера графики относительно `textwidth` и другой графики относительно `linewidth`. Протестировать их поведение с опцией `twocolumn` и без нее.
- Создать демонстрацию с использованием `lipsum` и попробовать различные спецификаторы позиционирования флоатов. Изучить, как различные спецификаторы взаимодействуют друг с другом.
- Добавить новые нумерованные части (разделы, подразделы, нумерованные списки) в тестовый документ и определить, сколько компиляций необходимо для корректной работы команд `label`.
- Добавить флоаты и проверить, что происходит при размещении `label` до `caption` вместо после; предсказать результат.
- Выяснить, что происходит при размещении `label` для уравнения после `end equation`.

3 Теоретическое введение

Для включения графики в LaTeX используется пакет `graphicx`, который добавляет команду `includegraphics`. Этот пакет поддерживает различные форматы изображений: EPS, PNG, JPG и PDF.

Изображения в LaTeX обычно включаются как плавающие объекты (floats), чтобы избежать больших пустых промежутков на странице. Для управления позиционированием используются спецификаторы:

- `h` – „Here“ (здесь, если возможно)
- `t` – Top (верх страницы)
- `b` – Bottom (низ страницы)
- `p` – Page (отдельная страница только для флоатов)
- `H` – абсолютное позиционирование (требует пакет `float`)

Система перекрестных ссылок в LaTeX работает с помощью команд `label` и `ref`. Для корректной работы требуется минимум две компиляции документа.

Более подробно про работу с графикой в LaTeX см. в [[kotelnikov_chebotaev_book_latex2_ru](#), [lvovsky_book_latex_ru](#)].

4 Выполнение лабораторной работы

4.1 Базовое включение графики

Начнем работу с базового включения изображений в документ. Для этого используется пакет `graphicx` и команда `includegraphics`. На рисунке ?? показан простейший пример включения изображения в документ.

```
\begin{figure}[!h]
    \centering
    \includegraphics[width=0.8\textwidth]{fig01.png}
    \caption{Базовый пример включения изображения}
    \label{fig:fig01}
\end{figure}
```

Рисунок 4.1: Базовый пример включения изображения

4.2 Манипулирование размерами графики

4.2.1 Параметр `width`

Параметр `width` позволяет задать ширину изображения. Можно использовать как абсолютные значения (например, `5cm`), так и относительные (например, `0.5` от ширины текста). На рисунке ?? показан пример использования параметра `width`.

```
\centering
\includegraphics[width=0.8\textwidth]{fig02.png}
```

Рисунок 4.2: Изображение с параметром `width`

4.2.2 Параметр `height`

Аналогично ширине, можно задать высоту изображения с помощью параметра `height`. LaTeX автоматически сохраняет пропорции изображения (рис. ??).

```
\centering
\includegraphics[height=5cm]{fig03.png}
```

Рисунок 4.3: Изображение с параметром `height`

4.2.3 Параметр `scale`

Параметр `scale` позволяет масштабировать изображение пропорционально. Например, `scale=0.5` уменьшит изображение в два раза (рис. ??).

```
\centering
\includegraphics[scale=0.5]{fig04.png}
```

Рисунок 4.4: Использование параметра `scale`

4.2.4 Поворот изображения

С помощью параметра `angle` можно повернуть изображение на заданный угол в градусах. На рисунке ?? показан пример поворота изображения на 45 градусов.

```
\centering
\includegraphics[angle=45,width=0.6\textwidth]{fig05.png}
```

Рисунок 4.5: Поворот изображения с помощью `angle`

4.3 Разница между `textwidth` и `linewidth`

Параметр `textwidth` — это ширина текстового блока на физической странице, тогда как `linewidth` — это текущая ширина, которая может локально отличаться от `textwidth`.

чаться. На рисунке ?? показан пример использования `textwidth`.

```
\texttt{\textbackslash textwidth} --- это ширина текстового блока на физической странице,  
тогда как \texttt{\textbackslash linewidth} --- это текущая ширина, которая может локально отличаться.
```

Рисунок 4.6: Пример с `textwidth`

Разница наиболее заметна при использовании опции класса `twocolumn`. В режиме двух колонок `linewidth` адаптируется к ширине колонки, а `textwidth` всегда относится к полной ширине страницы (рис. ??).

```
Разница наиболее заметна при использовании опции класса \texttt{twocolumn}.  
В режиме двух колонок \texttt{\textbackslash linewidth} адаптируется к ширине колонки,  
а \texttt{\textbackslash textwidth} всегда относится к полной ширине страницы.
```

Рисунок 4.7: Демонстрация разницы в режиме `twocolumn`

В обычном одноколоночном режиме эти значения одинаковы, но в `twocolumn` изображение с `width=0.5 linewidth` будет занимать половину ширины колонки, а с `width=0.5 textwidth` — половину всей страницы.

4.4 Позиционирование флоатов

4.4.1 Спецификаторы позиции

Можно использовать различные спецификаторы позиции для управления размещением изображений. На рисунке ?? показан пример кода с различными спецификаторами.

```
\begin{itemize}  
    \item \texttt{h} --- 'Here' (здесь, если возможно)  
    \item \texttt{t} --- Top (верх страницы)  
    \item \texttt{b} --- Bottom (низ страницы)  
    \item \texttt{p} --- Page (отдельная страница только для флоатов)  
\end{itemize}
```

Рисунок 4.8: Пример с различными спецификаторами позиции