

СОДЕРЖАНИЕ И ТЕХНОЛОГИИ
МАТЕМАТИЧЕСКОГО ОБРАЗОВАНИЯ В ВУЗЕ

УДК 004.912

РАБОТА С МАТЕМАТИЧЕСКИМИ ТЕКСТАМИ
С ПОМОЩЬЮ ПРОГРАММ ЭКРАННОГО
ДОСТУПА

Е. С. Корнев

*Кемеровский государственный университет,
Кемерово, Россия*

q148@mail.ru

Данная работа предназначена для слепых учащихся высших и средних учебных заведений, преподавателей, учёных и аспирантов по математическим специальностям. В работе излагаются трудности и особенности, связанные с чтением и написанием математических текстов при помощи так называемых программ экранного доступа, которые озвучивают информацию с экрана компьютера с помощью синтезатора речи и выводят её шрифтом Брайля на тактильные дисплеи. Описаны технические возможности работы с математическими текстами и связанные с этим проблемы. Данная статья также полезна зрячим преподавателям и учителям, которым приходится преподавать математику слепым учащимся.

Ключевые слова: программа экранного доступа, LaTeX, MathType, обучение людей с нарушением зрения.

1. Введение

Люди, у которых зрение не позволяет различать изображение на экране, работают на компьютере с помощью синтезированной речи и тактильных (брайлевских) дисплеев, которые выводят информацию с экрана посредством шрифта Брайля. Для этого на компьютер устанавливается специальная программа экранного доступа (от английского «Screen Reader»), которая и осуществляет все действия по преобразованию изображения на экране в речь и брайлевский текст.

Из-за этого сразу возникает целый ряд ограничений и проблем у пользователей таких программ, которых нет у пользователей с хорошим зрением. Прежде всего, программы экранного доступа функционируют по прин-

ципу получения информации об объекте не с экрана, а изнутри приложения, т. е. программа экранного доступа не распознаёт графическое изображение на экране, а получает внеэкранный информацию об объекте, такую как код символа в тексте, класс, состояние и имя объекта в структуре приложения (OSM), или содержимое тегов и атрибутов в разметке HTML, XML и MathML. Таким образом, если программа экранного доступа встречает графический объект, такой как рисунок, график, диаграмму, фотографию или графический значок, она не способна передать пользователю содержимое изображения. В лучшем случае сообщаются характеристики графического объекта: размеры, адрес расположения и тому подобное.

Это приводит к тому, что на сайтах, в документах и учебных материалах, где математические формулы и символы внедряются как графические изображения, пользователи программ экранного доступа просто не могут получить информацию об этих объектах.

Кроме того, программы экранного доступа не способны читать отсканированные в графические форматы бумажные книги и документы. Например, часто печатные учебники по математике сканируют в форматы PDF и DJVU, которые представляют из себя фотографическую копию изображения страниц. Прочитать такие файлы пользователи программ экранного доступа не могут.

Существуют технологии оптического распознавания символов (OCR), которые способны распознавать и переводить в электронный текст буквы, цифры, знаки пунктуации и простейшие специальные символы, однако они не способны распознавать и преобразовывать в какой-либо формат сложные математические формулы, графики функций, графы, коммутативные диаграммы и геометрические объекты.

Всё вышесказанное приводит к так называемой проблеме доступности математических текстов для программ экранного доступа, которой и посвящена данная работа.

Для создания математических текстов создано несколько компьютерных сред: LaTeX, MathML и тому подобные. Они создают электронные тексты, где математические символы внедряются не как графические объекты, а как символы юникода. С этими символами программы экранного доступа уже способны работать, описывать и читать.

Методы описания математических выражений, созданных с помощью LaTeX или MathML, разрабатывают сами создатели программ экранного доступа и пользователи на это никак повлиять не могут. Из-за этого уровень доступности и понимания математических текстов зависит от того, насколько качественно проработан этот момент в самой программе экранного доступа.

Тем не менее, в данной работе описано, каким образом авторы математических электронных документов могут улучшить их доступность и читаемость для пользователей программ экранного доступа с полным или частичным отсутствием зрения.

В частности, мы подробно рассмотрим следующие моменты:

- как можно читать математические выражения с помощью синтезированной речи и шрифта Брайля;
- что нужно делать при написании математических текстов для их доступности и читаемости программами экранного доступа;
- как слепые люди могут самостоятельно писать математические тексты для научных журналов, дипломных работ, презентаций и конференций.

На момент написания этой работы в мире широко используются следующие программы экранного доступа для Windows:

- бесплатная программа NVDA¹;
- коммерческая платная программа JAWS².

Локализацию и дистрибьюцию программы JAWS для России осуществляет компания «Элита групп»³.

Для Linux используется программа Orca⁴.

Наиболее качественно поддержка математических выражений реализована в программе JAWS для Windows, поэтому в данной работе будет рассматриваться только эта программа. Также JAWS рекомендуется всем учебным, научным и издательским организациям, работающим с математическими текстами.

Структура работы следующая: в разделе 2 мы опишем взаимодействие JAWS с математическими выражениями и то, как с помощью JAWS можно читать математические символы и формулы. В разделе 3 речь идёт о системе LaTeX и создании с её помощью доступных для программ экранного доступа электронных математических документов. В разделе 4 рассмотрена работа с математическими текстами в Microsoft Word с помощью надстройки MathType. Наконец, в разделе 5 описаны способы взаимодействия слепых учащихся, аспирантов, сотрудников и преподавателей с их зрячими коллегами или учителями при работе с математическими электронными текстами.

В данной работе рассматривается только проблема написания и чтения на компьютере математических текстов, проблема печати математических текстов по Брайлю является отдельной задачей и в данной работе не рассматривается.

¹URL: <https://nvaccess.org>

²URL: <https://freedomscientific.com>

³URL: <https://elitagroup.ru/pages/prod-jaws.php>

⁴URL: <https://help.ubuntu.com/stable/ubuntu-help/a11y-screen-reader.html.ru>

2. Работа с математическим текстом с помощью программы JAWS

JAWS является коммерческим платным продуктом от компании Freedom Scientific и предназначена только для работы в системе Windows. Ссылки на домашнюю страницу и страницу российского дистрибьютера программы JAWS можно найти в разделе 1. Рекомендуется всегда использовать самую последнюю версию JAWS, поскольку эта программа очень часто обновляется, и в каждом обновлении производятся исправления и доработки, в том числе поддержки математических символов.

В JAWS имеется очень подробная справочная система, которая позволяет самостоятельно изучить все функции и возможности JAWS и освоить работу с программой, а также имеется очень большая таблица названий специальных и математических символов. Это позволяет программе правильно произносить символы и буквы в любом месте, где они встречаются. Например, JAWS способна сообщать греческие буквы, различные символы стрелок, графические смайлы, математические символы и знаки операций. Однако для этого необходимо, чтобы эти символы имели именно те юникоды, которые внесены в таблицу символов JAWS. Если из-за особенностей шрифта или транскодирования, юникода символа нет в таблице, JAWS проигнорирует его.

Также, JAWS выводит обозначения наиболее распространённых математических символов по Брайлю на брайлевский (тактильный) дисплей в привычном для русского математического Брайля виде. Для этого необходимо выбрать в JAWS в качестве основной таблицы трансляции Брайля таблицу русского компьютерного Брайля Rus_Alternative. При выборе любой другой таблицы трансляции Брайля брайлевские обозначения математических символов либо не будут выводиться совсем, либо будут отличаться от принятых в русском Брайле обозначений.

Чтение математических символов, стоящих в тексте отдельно, не в виде формулы, выполняется в любых приложениях и в любых типах документов. Например, если в тексте встречается строчная греческая буква α , набранная математическим курсивным шрифтом, JAWS прочитает её как «курсивная строчная альфа». Но, если в тексте встретится формула, где эта же буква является частью математического выражения, JAWS уже не прочитает её отдельно.

Для чтения математических формул и прочих блоков текста, которые внедрены в документ с помощью MathML или MathType, в JAWS используется специальный инструмент — математический просмотр (Math Viewer). В случае, когда JAWS встречает в документе математическое выражение перед этим выражением сообщается: «мат. содержимое», после чего целиком зачитывается всё выражение в описательной форме. Если необходимо подробно изучить это выражение, на веб-страницах или в HTML-документах нужно, наведя системный фокус на выражение, нажать клавишу Enter, а в документе Word, наведя курсор на выражение, нажать на клавиатуре последовательную

ступенчатую команду: сначала одновременно нажать и отпустить клавиши Insert и Пробел, а затем в верхнем цифровом ряду клавиатуры нажать клавишу с символом «Равно». После этого также включится математический просмотр. Если на клавиатуре нет калькуляторного цифрового блока, и в JAWS выбрана раскладка клавиатуры для переносного компьютера (Laptop), то вместо клавиши Insert нужно нажимать CapsLock.

После включения математического просмотра клавиши стрелок влево и вправо переводят между компонентами формулы одного уровня. Нажатие на текущем компоненте формулы клавиши «Стрелка вниз» переводит на дочерний уровень этого компонента. Нажатие клавиши «Стрелка вверх» возвращает на предыдущий уровень. Например, если в режиме математического просмотра вы просматриваете формулу

$$\sqrt{x^2 + 1} + e^{\sin(\alpha) + \cos(\alpha)},$$

то на первом уровне просмотра клавиши стрелок влево и вправо будут перемещать между квадратным корнем и экспонентой. Если же необходимо подробно изучить показатель экспоненты, то нужно, находясь на экспоненте, нажать стрелку вниз, а потом стрелками влево и вправо переходить между синусом и косинусом. Для возврата на первый уровень нужно нажать стрелку вверх. Аналогично можно прочесть более детально подкоренное выражение под знаком радикала. Для выхода из режима математического просмотра нужно нажать клавишу Escape.

Поскольку программа JAWS локализуется на русский язык, описание математических конструкций в русской версии JAWS производится на русском языке с использованием русских названий функций и символов. Вот некоторые примеры такого описания.

Пример 2.1. JAWS прочитает формулу

$$\alpha \otimes \beta \cap \gamma$$

как «альфа произведение в круге бета пересечение гамма».

Пример 2.2. JAWS прочитает формулу

$$\int_0^{\infty} \sqrt{x^2 + 1} dx$$

как «интеграл с ноль снизу по центру и бесконечность сверху по центру квадратный корень из $x^2 + 1$ по центру».

Здесь нужно понимать, что в разных разделах математики некоторые символы могут обозначать разные операции. Поэтому для описания используется описание графического вида символа. Так, например, в тензорном анализе символ \wedge читается как «внешнее произведение», а в математической логике как «конъюнкция» или «логическое и». Поскольку в JAWS нет анализа

контекста, символы, имеющие неоднозначное применение, описываются так, как они выглядят графически.

При словесном описании возникают проблемы с обозначениями, когда у символа имеется верхний индекс. Например, JAWS всегда будет читать выражение x^0 как «икс в степени ноль», даже если это обозначает координату точки с координатами (x^0, x^1, \dots, x^n) . Для анализа математического выражения JAWS использует теги MathML. Поэтому, в документах, где формулы созданы не с использованием MathML, JAWS может описывать формулы некорректно или частично. Также JAWS игнорирует любые формулы в документах или на веб-страницах, которые внедрены в текст как графические изображения.

Таким образом, для того, чтобы JAWS читала математический текст максимально корректно, рекомендуется создавать все математические выражения в документе с помощью MathML или средств, которые преобразуют выражения в такие объекты. Например, надстройка MathType для Word позволяет преобразовывать формулы из нотации LaTeX в MathML, это подробно описано в разделе 4. При таком способе внедрения математических выражений в документ Word JAWS будет максимально корректно читать их. Если формулы в документ Word внедрять с помощью редактора формул Word устаревших версий, JAWS будет воспринимать их как графические объекты и не сможет описывать в режиме математического просмотра.

3. Работа с LaTeX

LaTeX — это система макрокоманд, позволяющая создавать электронные документы любой сложности по аналогии с созданием программ. То есть, сначала создаётся исходный файл документа в текстовом формате с расширением `tex`, в этом исходном файле с помощью специального набора команд задаются все свойства оформления и структуры документа, приводится текст самого документа и с помощью специальных команд набирается любой математический текст. Затем, этот исходный файл документа компилируется в файл, где текст уже выглядит именно так, как если бы он был напечатан в типографии или создан в любом другом текстовом редакторе. Исходный файл можно компилировать в документы форматов DVI, PDF, PS и HTML. При этом все математические формулы и символы при компиляции в конечном файле выглядят уже в печатном общепринятом виде. Компиляцию можно запускать как с помощью командной строки, так и с помощью графических интерфейсов через меню и сочетания клавиш в текстовых редакторах, которые рассчитаны на работу с LaTeX.

Основное ядро LaTeX было создано Дональдом Кнутом в 1978 году для работы через командную строку. Затем это ядро стали расширять, надстраивать и адаптировать для различных операционных систем. Для Windows наиболее распространённым дистрибутивом LaTeX с графическим интерфейсом

является среда MiKTeX⁵. В этот дистрибутив уже встроен текстовый редактор для LaTeX с графическим интерфейсом, однако для работы с программами экранного доступа гораздо более удобным является текстовый редактор WinEdt⁶.

В этом текстовом редакторе компиляцию набранного исходного текста документа можно запускать либо через пункты меню, либо с помощью сочетаний клавиш. Кроме того, WinEdt автоматически обнаруживает установленный в Windows дистрибутив LaTeX и настраивается на работу с ним. Например, вы можете написать в окне WinEdt исходный текст документа LaTeX, затем нажать сочетание клавиш Control+Shift+P, и сразу начнётся компиляция этого текста в конечный файл формата PDF. Если компиляция пройдёт без ошибок, скомпилированный документ автоматически откроется в приложении по умолчанию для просмотра файлов PDF, выбранном в Windows.

Для чтения скомпилированных LaTeX документов программами экранного доступа пригоден только формат PDF, поскольку такие форматы как DVI и PS являются графическими и недоступны для чтения программами экранного доступа. Более того, скомпилированные с помощью LaTeX файлы PDF содержат все математические символы как символы с юникодами.

Программа JAWS может корректно читать эти символы, но описывать их с помощью инструмента математического просмотра (см. раздел 2) уже не может. В следствии чего сложные математические формулы могут читаться некорректно или непоследовательно.

Перечислим преимущества создания текстов в системе LaTeX.

1. Все параметры форматирования и оформления документа указываются в исходном файле с помощью текстовых команд. В частности, размер и начертание шрифта, размер полей, интервал и все прочие.
2. Структура документа также полностью задаётся текстовыми командами в исходном файле. Например, заголовки, главы, сноски, таблицы, списки и многое другое.
3. Все математические формулы и символы набираются с помощью текстовых команд, которые читаются любой программой экранного доступа. Кроме того, из вида этих команд полностью понятна структура формулы или вид символа. Не нужно перетаскивать какие-либо символы мышью с панели символов или ограничиваться только теми символами и шрифтами, которые есть в Word или других текстовых редакторах.
4. LaTeX самостоятельно при компиляции документа верстаёт всё оформление, структуру и внешний вид текста. Не нужно вручную что-либо выделять и форматировать как в редакторах типа Word.

⁵URL: <https://miktex.org>

⁶URL: <https://winedt.com>

5. Если в структуре команд допущена ошибка, то LaTeX остановится и выведет сообщение об ошибке с указанием её причины и номера строки в исходном тексте.
6. Оформление шрифта и форматирования в формулировках и доказательствах математических утверждений, таких как теоремы, замечания и определения, происходит автоматически. В исходном файле достаточно только указать тип утверждения, его начало и конец с помощью текстовых команд. Нумерация утверждений происходит автоматически, а также имеется удобная система расстановки закладок и меток, которые позволяют ссылаться на номера утверждений, формул и страниц, не указывая явно их номер.
7. Каждый математический символ или признак указывается в исходном файле с помощью текстовой команды, читаемой любой программой экранного доступа.

В силу вышесказанного, использование LaTeX позволяет пользователю с отсутствием зрения полностью самостоятельно набирать и форматировать любой математический текст без визуального контроля, а после компиляции в формат PDF получать готовый математический текст, который можно демонстрировать в любых ситуациях: на конференциях, докладах, при работе со зрячими коллегами или для публикации в научных журналах.

LaTeX является открытым продуктом и сторонние разработчики создали для него множество расширений, которые называются пакетами. Подключая различные пакеты в начале исходного файла документа, можно добавлять в базовые возможности LaTeX различные команды и навыки. Также любой желающий может создать для LaTeX свой собственный пакет или шаблон документа. Для тех, кто не видит математический текст на экране, LaTeX фактически является линейным текстовым представлением математических документов. Если изучить синтаксис и команды LaTeX, то читая исходный файл документа, а не его скомпилированную в формат электронного документа версию, можно полностью понять смысл, вид и структуру любых математических конструкций, даже таких как графы, диаграммы, матрицы и многоуровневые формулы.

Важнейшим преимуществом LaTeX является использование так называемых математической моды и текстовой моды. Все символы в текстовой моде при компиляции выглядят в конечном документе точно так, как они написаны в исходном файле. А специальные символы и команды в математической моде обрабатываются при компиляции и дают в конечном документе графические символы и выражения, в привычном для математической записи в литературе виде.

Для включения и выключения математической моды в LaTeX служат специальные символы или команды. Например, если в строку обычного текста нужно вставить математический символ или выражение, то в начале и конце этого выражения нужно поставить знак $\$, а математические симво-$

лы набрать с помощью специальных текстовых команд. Признаком начала команды LaTeX служит символ `\`. Например, чтобы вставить в строку текста выражение $\alpha^2 + \beta^2$ нужно в тексте исходного файла с расширением `tex` написать

```
\alpha^2+\beta^2
```

(здесь команды `\alpha` и `\beta` печатают в конечном документе греческие буквы α и β , а символ `^` обозначает, что стоящий после него символ должен быть напечатан как верхний правый индекс над предшествующим символом).

Аналогично, символ `_` в исходном файле документа указывает на то, что при компиляции, стоящий после него символ должен быть напечатан как нижний правый индекс под предшествующим символом, т. е. выражение `\$x_n\$` в исходном файле после компиляции в файле PDF даст x_n .

Чтобы напечатать в конечном документе математическое выражение в отдельной строке с указанием номера, или без него, нужно заключить в исходном файле документа это выражение между командами начала и конца вынесенной математической формулы. Таких команд существует несколько, и их можно найти в [1,2]. Например, любое математическое выражение, набранное в исходном файле между двойным символом `$$`, печатается в отдельной строке математическим шрифтом без номера.

Таким образом, зная команды и синтаксис LaTeX, любой математический текст можно написать следующим образом: создаём текстовый файл с расширением `tex` в любом текстовом редакторе, адаптированном под LaTeX, в частности в WinEdt, записываем в него весь текст документа с использованием команд и конструкций LaTeX, отправляем этот исходный файл на компиляцию в формат PDF с помощью меню или сочетания клавиш, после компиляции получаем полностью готовый корректно оформленный документ с математическими выражениями в привычном для математической записи виде.

Теперь приведём примеры оформления в LaTeX математических выражений и утверждений.

Пример 3.1. Чтобы получить после компиляции в документе теорему с доказательством с автоматически проставленным номером и правильным визуальным оформлением, нужно в исходном файле документа написать примерно следующую конструкцию:

```
\begin{theorem} Любая непрерывная на отрезке  $[a, b]$  функция
интегрируема по Риману.
\end{theorem}
```

```
\begin{proof}
Пусть  $f$  - непрерывная на отрезке  $[a, b]$  функция...
\end{proof}
```

После компиляции получим в файле PDF следующий текст:

Теорема 1. *Любая непрерывная на отрезке $[a, b]$ функция интегрируема по Риману.*

Доказательство. Пусть f - непрерывная на отрезке $[a, b]$ функция... ■

Здесь видно, что слова «Теорема» и «Доказательство» напечатаны автоматически в месте указания команд, начинающих формулировку и доказательство теоремы, номер теоремы подставлен автоматически, слово «Теорема» набрано полужирным шрифтом, а сама формулировка теоремы и слово «Доказательство» — курсивным шрифтом. И всё это LaTeX делает автоматически, безо всякого вмешательства пользователя. Конечно, при желании можно номера теорем проставлять вручную или полностью отключить нумерацию утверждений.

Как было сказано выше, в LaTeX можно подключать множество сторонних пакетов, позволяющих использовать специальные команды, которых нет в базовом дистрибутиве LaTeX. Подключение любого пакета с названием, скажем `mypack`, происходит стандартным образом: в начальном разделе исходного файла, где прописываются все глобальные свойства документа, получаемого после компиляции, этот раздел называется преамбулой документа, нужно вписать строку:

```
\usepackage{mypack}
```

Пример 3.2. Подключим в преамбуле исходного файла пакет `amsmath`, который вводит много дополнительных полезных команд для набора математических конструкций. В частности, этот пакет добавляет отдельные команды для круглых, квадратных и прямых скобок для матриц. Для подключения этого пакета в преамбуле вписываем строку

```
\usepackage{amsmath}
```

Теперь напишем в исходном файле конструкцию, которая после компиляции даст квадратную матрицу в круглых скобках из трёх строк и трёх столбцов с коэффициентами, имеющими двойной нижний индекс.

```
$$
\begin{pmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{pmatrix}
$$
```

Здесь символ `&` указывает начало новой ячейки таблицы, команда `\\` указывает начало новой строки таблицы, а в фигурные скобки заключены все выражения, которые должны стоять в нижнем индексе у буквы a . Фигурные скобки используются для того, чтобы LaTeX было понятно, что после символа `_` нужно ставить в нижний индекс не только первую цифру, а обе цифры, записанные в фигурных скобках.

После компиляции получим:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

LaTeX не может самостоятельно расщеплять длинные математические выражения на строки. Поэтому нужно делать это вручную, контролируя результат в скомпилированном файле PDF. Для расщепления длинных формул можно использовать окружения `split` и `aligned`. Однако место расщепления на строки можно указать только вручную.

Пример 3.3. Чтобы расщепить формулу на строки с выравниванием по одной общей позиции в строке, можно воспользоваться окружением `split`. Однако это окружение не воспринимается, если переключение в математическую моду происходит с помощью команды `$$`. Поэтому, это окружение следует использовать с такими командами перехода в математическую моду, как `\[\]` или `\(\)`. Если в исходном файле документа написать следующую конструкцию:

```
\[
\begin{split}
a_1(x)&=f_1(x)+g_1(x), \\
a_2(x)&=f_2(x)+g_2(x), \\
a_3(x)&=f_3(x)+g_3(x).
\end{split}
\]
```

после компиляции получим:

$$\begin{aligned} a_1(x) &= f_1(x) + g_1(x), \\ a_2(x) &= f_2(x) + g_2(x), \\ a_3(x) &= f_3(x) + g_3(x). \end{aligned}$$

Здесь символ `&` указывает точку выравнивания в каждой строке, а команда `\\` указывает место начало новой строки.

Приведём обязательные рекомендации для корректной работы программ экранного доступа с MiKTeX в Windows. На момент написания статьи MiKTeX является Java-приложением. Поэтому для того, чтобы программы экранного доступа могли озвучивать графический интерфейс, необходимо установить в Windows компонент Java Runtime версии 8 или выше, и включить функцию Access Bridge. Для этого скачайте текущую версию компонента JAVA с сайта www.java.com и установите стандартным образом.

После окончания установки в Windows 10 или выше нажмите кнопку «Пуск», через поиск найдите и щёлкните пункт «Панель управления». Откроется панель управления Windows. Зайдите в раздел «Специальные возможно-

сти», далее «Центр специальных возможностей», далее «Использование компьютера без дисплея». В открывшемся окне установите флажок «Enable Java Access Bridge», перезапустите вашу программу экранного доступа и откройте настройки Java или MiKTeX. Если ваша программа экранного доступа привычным образом озвучивает элементы окна, значит Java Access Bridge включён и работает.

Ещё одна проблема при чтении файлов PDF, скомпилированных с помощью LaTeX, связана с некорректным чтением программами экранного доступа кириллических символов. Если при создании документа на русском языке используется пакет babel с опцией russian, то в конечном файле PDF русские буквы будут кодироваться таким образом, что программы экранного доступа будут читать бессмысленные названия символов.

Как было описано в разделе 2, программы экранного доступа распознают символы по их коду, а не графическому представлению. Поэтому, чтобы русский текст правильно читался программами экранного доступа, необходимо, чтобы кириллические символы имели стандартные коды. Чтобы устранить эту проблему, достаточно подключить в исходном файле документа пакет spsr, который делает кодировку кириллических символов в файле PDF корректной для чтения программами экранного доступа. Для этого в преамбуле документа добавьте строку

```
\usepackage{spsr}
```

В разделе 5 мы приведём инструкцию, как с помощью LaTeX создавать доступные для чтения программой JAWS веб-страницы, где математические выражения внедряются в формате MathML. Более подробно о работе с LaTeX и его средствах создания математических текстов можно узнать в [1] или [2]. Руководство по MiKTeX см. в [3].

4. Работа в Microsoft Word

В текстовом редакторе Microsoft Word имеется собственный редактор формул, однако работа с ним с помощью программ экранного доступа практически недоступна. Поэтому для создания, читаемых программой JAWS математических текстов, необходимо установить в Windows программу MathType⁷. MathType имеет собственный язык создания математических выражений, однако он недоступен для программ экранного доступа. Вместе с MathType устанавливаются надстройки в программах Word и Powerpoint, которые позволяют внедрять в файлы этих программ математические выражения, которые читаются JAWS и доступны для математического просмотра JAWS (см. раздел 2).

⁷URL: <https://www.wiris.com/en/mathtype/>

Надстройка MathType в Word позволяет писать любые математические выражения прямо в документе Word на языке LaTeX, а затем преобразовывать их в привычные математические выражения, которые читаются математическим просмотром JAWS. Для полноценной работы с Word требуется самая последняя версия программы JAWS, совместимая с вашей версией Word версия MathType, версия Microsoft Word 2019 или выше, или Microsoft 365 Word.

Чтобы убедиться, что в Word подключена надстройка MathType, после установки в Windows MathType откройте Word, перейдите на верхнюю ленту и убедитесь, что на ленте появился пункт «MathType». Если этого пункта нет, разрешите в настройках безопасности Word использование сторонних макросов и позвольте доверять надстройкам и макросам от разработчика Design Science.

Если надстройка MathType в Word подключена, то необходимо настроить трансляцию кода LaTeX в формулы MathType.

Для этого сделайте следующее: на верхней ленте нажмите пункт «MathType», далее нажмите «Format», затем «Convert Equations». В открывшемся окне установите флажки для всех предлагаемых типов формул. Переключите переключатель «Convert to» в положение «MathType Translator Text», и в списке с названиями трансляторов выберите «AMSLaTeX».

После этого верните положение переключателя «Convert to» в положение «MathType Equations (OLE Objects)» и нажмите кнопку начала конвертирования. Эту настройку достаточно выполнить один раз, далее все произведённые здесь настройки будут применяться по умолчанию при конвертировании формул.

MathType понимает не все команды перехода в математическую моду LaTeX. Например, математические выражения, вынесенные в отдельную строку, нужно заключать только в командные скобки $\[\]$, а команды $\$$ и $\(\)$ MathType игнорирует. Для преобразования математических выражений, встроенных в строку текста, следует стандартным образом заключать их между символами $\$$.

Чтобы преобразовать выражение в коде LaTeX в графическую формулу Word, необходимо навести курсор на текст выражения LaTeX и нажать сочетание клавиш $\text{Alt}+\backslash$. На месте текста LaTeX появится графическая формула, которую можно детально прочитать в математическом просмотре JAWS, если находясь на этой формуле сначала одновременно нажать и отпустить сочетание клавиш $\text{Insert}+\text{Пробел}$, а затем нажать в верхнем ряду клавиатуры клавишу с символом $=$.

Эта ступенчатая команда работает только при запущенной программе JAWS и только на формуле в документе Word. Если повторно нажать на графической формуле сочетание клавиш $\text{Alt}+\backslash$, формула снова будет отображена в виде текста LaTeX. Это позволяет редактировать формулы даже после их преобразования в графический вид. Рассмотрим этот процесс подробно на примерах.

Пример 4.1. Чтобы вставить в документ Word с помощью MathType строку текста «Пусть φ - угол между вектором v и осью абсцисс.», наберите следующее:

Пусть φ - угол между вектором v и осью абсцисс.

Затем наведите курсор на выражение φ и нажмите сочетание клавиш Alt+\. Вместо текста LaTeX появится греческая буква φ . Затем наведите курсор на выражение v и нажмите Alt+\. в строке появится строчная латинская буква v , набранная наклонным математическим шрифтом.

Пример 4.2. Чтобы вставить в документ Word с помощью MathType отдельную строку с равенством

$$\sin^2(\alpha) + \cos^2(\alpha) = 1$$

наберите:

`\[\sin^2(\alpha)+\cos^2(\alpha)=1\]`

Наведите курсор на любой символ в этой строке и нажмите сочетание клавиш Alt+\. На месте текста LaTeX появится графическая формула. При наведении курсора на эту формулу JAWS целиком будет зачитывать её. Если необходимо просмотреть эту формулу подробнее с помощью математического просмотра JAWS, наведите курсор на формулу, нажмите и отпустите Insert+Пробел, а затем нажмите клавишу с символом =.

Рекомендуется конвертировать любые выражения в коде LaTeX сразу после их набора, нажимая сочетание клавиш Alt+\. Если набрать в документе много текста в формате LaTeX, а потом попытаться сконвертировать в графические формулы через описанный выше пункт верхней ленты «Convert Equations», конвертация может зависнуть или не сработать.

MathType также позволяет обратное конвертирование графических формул редактора формул Word в текст формата LaTeX. Для этого нажмите описанным выше способом в верхней ленте пункт «Convert Equations», в открывшемся окне установите переключатель «Convert to» в положение «MathType Translator Text» и нажмите кнопку начала конвертации.

При большом количестве формул в документе конвертация также может не сработать, или некоторые формулы не будут сконвертированы. Если в качестве транслятора MathType в настройках конвертации формул выбран AMSLaTeX, как описано выше, то все примеры набора матриц и прочих математических конструкций из раздела 3 будут корректно преобразованы в графические формулы Word.

5. Создание и чтение доступных математических документов

Здесь будут даны рекомендации по созданию математических текстов, доступных для чтения как программами экранного доступа, так и зрячими пользователями с экрана. Мы также обсудим варианты и способы работы

пользователей программ экранного доступа при проведении презентаций, выступлений с защитой своих работ или с докладами на конференциях.

Зрячим пользователям, работающим со слепыми учащимися или сотрудниками, будет полезно узнать, как они могут создавать математические тексты и веб-страницы, доступные для чтения программами экранного доступа.

5.1. Что делать слепым пользователям

Прежде всего изучите команды и структуру LaTeX, используя электронные версии книг и руководств по LaTeX, например, электронные версии публикаций [1, 2]. Если вы хотите полностью самостоятельно создавать математические документы, такие как статьи для журналов, презентации докладов, диссертации и дипломные работы, установите последнюю версию MiKTeX и WinEdt. Используя эту связку вы абсолютно самостоятельно сможете создавать полностью готовые, правильно оформленные, математические документы. Подробнее об этом см. раздел 3.

Если вам нужно создавать учебные пособия для школы или организаций, где требуют только документы в формате Word, установите Microsoft Word версии 2019 или выше и программу MathType. Это позволит, используя команды LaTeX создавать документы Word с графическими математическими формулами. Подробнее об этом см. раздел 4. Создание презентаций с помощью LaTeX мы отдельно рассмотрим позднее.

Если вы учитесь в вузе, где читаются лекции по математическим дисциплинам, то прежде всего просите лекторов прочитывать всё, что они пишут на доске или выводят на экран проектора. Также вы можете попросить у преподавателя, который читает лекции дать вам исходные файлы лекций в формате LaTeX. Это позволит вам читать любые математические выражения в полностью доступном для программ экранного доступа виде, как обычный текст.

Если вы хотите прочитать статью или книгу, которая имеется только в электронном формате PDF, то попробуйте написать автору и попросить исходный файл в формате TeX, объяснив, что программа экранного доступа технически не позволяет вам полностью понять математические выражения в электронном документе. Если вам необходимо прочитать книгу, которая имеется только в отсканированном в формате PDF или DJVU виде файла, содержащего фотографическую копию страниц бумажной книги, то никаких технических средств распознать математические выражения и правильно вывести их в текстовом формате не существует. В этом случае вы можете попросить зрячих людей, начитывать голосом текст книги на любое записывающее устройство типа диктофона, рекордера и тому подобного. Чтобы зрячие чтецы правильно называли математические символы, греческие буквы и обозначения, они могут воспользоваться любым математическим справочником, Википедией и прочими доступными источниками, например [4]. Кроме того,

они могут просто снять символ или формулу на камеру смартфона, отправить это в интернет и быстро найти название и описание символа или обозначения. Также в интернете можно найти уже готовые таблицы математических символов с их названиями и соответствующими командами LaTeX.

Если вы хотите читать математические формулы в Википедии в формате LaTeX, то сделайте следующее:

1. Зарегистрируйтесь в Википедии и войдите со своей учётной записью.
2. Зайдите в настройки вашего профиля и найдите настройку, отвечающую за вывод формул.
3. По умолчанию будет выбран графический формат. Измените значение на «Вывод в формате LaTeX» и сохраните изменения.

После этого в любом веб-браузере все математические выражения будут отображаться в виде команд LaTeX, которые полностью читаются любой программой экранного доступа.

Если вам нужно прочитать документ или веб-страницу, где математические выражения созданы с помощью языка разметки MathML, то установите последнюю версию программы JAWS и воспользуйтесь встроенным в JAWS инструментом «Математический просмотр». Подробнее об этом см. в разделах 2, 4. Если в документе или на веб-сайте математические выражения внедрены в виде графических изображений, то прочитать их с помощью программ экранного доступа невозможно. В этом случае придётся искать альтернативный текст в формате LaTeX или MathML.

5.2. Что делать зрячим преподавателям

Зрячим преподавателям и сотрудникам, которым нужно взаимодействовать со слепыми или слабовидящими людьми, необходимо создавать все электронные математические документы в таком виде, чтобы их могли читать и обычные зрячие пользователи, и пользователи программ экранного доступа. Делается это очень просто. Если вы хотите создать документ в формате PDF с помощью LaTeX, то просто подключите в исходном файле документа два пакета: `smap` и `axessibility`. Оба этих пакета можно подключить с помощью добавления в преамбулу документа одной строки вида:

```
\usepackage{smap,axessibility}
```

Подключение этих пакетов никак не влияет на визуальное представление документа и не изменяет зрительного представления формул. Пакет `smap` позволяет программам экранного доступа правильно читать кириллические символы в скомпилированных файлах PDF, а пакет `axessibility` добавляет в файл PDF невидимые теги, в которых содержится исходный код всех математических символов и выражений в виде кода LaTeX.

Программа JAWS способна обнаруживать и читать содержимое таких скрытых тегов. Таким образом, зрячие пользователи смогут привычным образом читать документ с экрана, а пользователи программ экранного доступа

будут вместо графических формул и символов получать их текстовое представление в формате LaTeX. Это позволит пользователям программ экранного доступа полностью понимать структуру и смысл математических выражений.

Если вы хотите создать документ в формате Word, который можно было бы прочитать с помощью программы JAWS, то установите программу MathType, включите в настройках конвертации формул конвертацию в формат MathML и выполните конвертацию всех графических формул в этот формат с помощью надстройки MathType для Word. Более подробно это описано в разделе 4. Если все формулы будут корректно сконвертированы в MathML, то пользователи программы JAWS смогут читать их и детально анализировать с помощью упомянутого выше средства «Математический просмотр».

Если вы хотите создать веб-сайт с математическим содержанием, которое пользователи программ экранного доступа могли бы читать, то создавайте все математические выражения на сайте в формате MathML и не используйте графических изображений формул или символов в виде внедрённых графических файлов или ссылок на них.

5.3. Создание доступных веб-страниц с математическими выражениями

Как было описано в разделе 2, программа экранного доступа JAWS имеет встроенный инструмент «Математический просмотр», который позволяет читать и детально исследовать математические выражения на веб-страницах, внедрённые с помощью разметки MathML. Здесь мы рассмотрим подробную инструкцию, как создать файл HTML, содержащий математические выражения в формате MathML, с помощью LaTeX. Мы будем рассматривать в качестве примера операционную систему Windows и дистрибутив MiKTeX для Windows.

Предположим, что у нас есть исходный файл документа в формате LaTeX под именем Article.tex, и мы хотим сконвертировать его в файл формата HTML, где все математические выражения внедрены в формате MathML. Установите самый последний дистрибутив MiKTeX (см. раздел 3). В современные дистрибутивы MiKTeX входит компилятор Make4ht, который мы и будем использовать.

В папке, где находится файл Article.tex и все служебные файлы документа создайте пустой текстовый файл, переименуйте его в Compile.bat. Откройте этот пустой файл в любом текстовом редакторе, впишите в него следующий текст:

```
start make4ht.exe -u Article.tex "mathml"
```

и сохраните файл. Если имя файла исходного текста документа содержит пробелы, то обязательно заключите его в файле Compile.bat в кавычки.

Теперь запустите файл `Compile.bat` и дождитесь окончания компиляции. В папке с файлом `Article.tex` будут созданы файлы `Article.html`, `Article.css` и другие служебные файлы. Приведённый выше синтаксис командного файла создаёт файл формата HTML5 в кодировке UTF8 с математическими выражениями в формате MathML. Если при открытии полученного файла HTML в браузере заголовок страницы отображается некорректно, то откройте файл `Article.html` в любом текстовом редакторе и вручную перепишите содержимое тега `<title> </title>`.

При использовании компилятора `Make4ht` все изображения, внедрённые в файл исходного текста документа корректно встраиваются в файл HTML в виде ссылок на исходные файлы изображений. Это позволяет внедрять в файл HTML графики, рисунки и диаграммы. Однако, программа `JAWS` не способна описывать такие графические объекты, встроенные в веб-страницу.

Вы в дальнейшем можете использовать командный файл `Compile.bat` для преобразования других исходных файлов формата `tex` в веб-страницы, изменяя на соответствующее имя файла в тексте файла `Compile.bat`. Кроме того, вы можете не использовать командный файл, а вводить приведённую в нём последовательность команд в командной строке `Windows`. Однако в этом случае, нужно указывать полный путь к исходному файлу формата `tex`, заключая его в кавычки.

5.4. Создание презентаций с помощью LaTeX

Рассмотрим ситуацию, когда слепому или слабовидящему пользователю нужно самостоятельно создать презентацию доклада на конференции, защите диплома или курсовой работы или прочих публичных выступлений перед зрячей аудиторией.

Проще всего создать презентацию в формате PDF, используя специальный пакет `Beamer`. Этот пакет позволяет создавать презентации в виде набора слайдов формата PDF, используя все возможности набора математических выражений LaTeX. В документации по пакету `Beamer` подробно описаны все его возможности и настройки. Здесь мы только отметим, что исходный файл презентации вручную разбивается на слайды с помощью команд начала и конца слайда `\begin{frame} \end{frame}`.

После компиляции исходного файла презентации получится файл PDF, слайды в котором можно переключать с помощью клавиш со стрелками влево или вправо на клавиатуре компьютера или с помощью кнопок перехода назад и вперёд на беспроводном пульте проектора. Таким образом, достаточно иметь под руками исходный текст презентации. Читая его, вы можете знать, где начинается и заканчивается слайд, а также читать всё содержимое слайдов в формате LaTeX. Например, если у вас есть брайлевский дисплей, то при проведении выступления с показом слайдов можно вывести на брайлевский дисплей исходный файл формата TeX, голосом читая с него содержимое, а на экране будут демонстрироваться скомпилированные слайды.

Можно выписать или распечатать содержимое исходного файла презентации на брайлевскую бумагу и при выступлении читать содержимое с неё, переключая слайды презентации в нужных местах. Можно заучить места смены слайдов и переключать их в ходе рассказа, если вы помните на каком тексте заканчивается один слайд и начинается другой.

Вот пример слайда с математическим текстом в исходном файле презентации:

```
\begin{frame}
\frametitle{Итоговый результат}
```

В итоге мы доказали следующую ключевую теорему:

```
\begin{theorem}
```

Ядро любой регулярной ненулевой кососимметричной 2-формы на вещественном многообразии нечётной размерности $2n+1$ имеет также нечётную размерность $k: 1 \leq k \leq 2n-1$.

```
\end{theorem}
\end{frame}
```

После компиляции в файл PDF получим следующий слайд с заголовком:

Итоговый результат

В итоге мы доказали следующую ключевую теорему:

Теорема 2. *Ядро любой регулярной ненулевой кососимметричной 2-формы на вещественном многообразии нечётной размерности $2n+1$ имеет также нечётную размерность $k: 1 \leq k \leq 2n-1$.*

Пакет Beamer позволяет внедрять в файл PDF презентации гиперссылки, перекрёстные ссылки на текст и другие объекты, включая рисунки, диаграммы, графы и так далее. Однако следует помнить, что при превышении размера слайда, весь превышающий размер текст, будет обрезан и не попадёт на слайд.

LaTeX не умеет автоматически переносить текст при переполнении слайда. Поэтому необходимо контролировать текст после компиляции, читая полученный файл PDF, вручную перемещая команды начала и конца слайдов. Все команды, конструкции и правила набора математических выражений LaTeX работают при подключении пакета Beamer точно также, как при создании обычного документа с помощью LaTeX.

Литература

- [1] Львовский С. М. Набор и вёрстка в системе LaTeX (5-е издание, переработанное). Москва: МЦНМО, 2014. 398 с.
- [2] Котельников И. А., Чеботаев П. З. LaTeX-2e по-русски. Настольная издательская система. 4-е издание. Санкт-Петербург: Корона-Век, 2011. 496 с.
- [3] MiKTeX Manual. URL: <https://docs.miktex.org/manual/>.
- [4] Система обозначений по математике, физике, химии и астрономии: учебное пособие / сост.: Быков А. Г., Егоров М. И., Морозова Г. Б., Проскуряков И. В. (2-е изд., перераб. и доп.) Москва: ВОС, 1982. 450 с.

Поступила 27.12.2023

WORKING WITH MATHEMATICAL TEXTS USING SCREEN READING SOFTWARE

E. S. Kornev

This work is intended for blind students of higher and secondary educational institutions, teachers, scientists and graduate students in mathematical specialties. The paper outlines the difficulties and features associated with reading and writing mathematical texts using so-called screen reading software, which read information from the computer's screen using a speech synthesizer and display it in Braille on tactile displays. The technical capabilities of working with mathematical texts and related problems are described. This work is also useful for sighted university and high school teachers who have to teach mathematics to blind students.

Keywords: screen reading software, LaTeX, MathType, training for people with visual impairments.