

Разработка конвертера препринтов ИПМ из формата .docx в форматы HTML и JATS XML

Р.Ю. Скорнякова

ИПМ им. М.В. Келдыша РАН

Аннотация. Наряду с традиционной формой электронного представления полных текстов научных статей – форматом PDF – в последние годы получил широкое распространение формат HTML, обладающий для онлайн-публикаций рядом преимуществ за счет имеющихся в нем средств для лучшей структуризации материала, вставки мультимедийного контента и реализации разного рода интерактивных и динамических возможностей. Наиболее распространенным подходом к формированию HTML-версии статьи является предварительное создание ее XML-версии в соответствии с разработанным в США стандартом JATS XML, который помимо основы для создания HTML- и PDF-версий, является также стандартом для обмена содержимым статей и его хранения. Однако конвертация в этот формат набранных в наиболее распространенных форматах .docx и LaTeX научных статей со сложным содержимым, включающим большое количество формул, таблиц и рисунков, является непростой задачей и имеющиеся программные средства либо не справляются с ней в полном объеме, либо обходятся довольно дорого. В работе предложен подход к созданию конвертера научных статей из формата .docx в форматы HTML и JATS XML с использованием инструмента с открытым исходным кодом Mammoth и описан созданный на основании этого подхода прототип конвертера препринтов ИПМ в HTML с последующим преобразованием в JATS XML.

Ключевые слова: HTML-версия журнальной статьи, JATS XML, преобразование форматов научных статей, препринты ИПМ

Development of a converter of KIAM preprints from .docx format to HTML and JATS XML formats

R.Y. Skornyakova

Keldysh Institute of Applied Mathematics

Abstract. Along with the traditional form of electronic presentation of full texts of scientific articles – PDF format – in recent years the HTML format has become widespread. HTML has a number of advantages for online publications

due to the means it has for better structuring of material, insertion of multimedia content and implementation of various interactive and dynamic capabilities. The most common approach to generating an HTML version of an article is to first create its XML version in accordance with the JATS XML standard developed in the USA, which, in addition to the basis for creating HTML and PDF versions, is also a standard for exchanging and storing article contents. However, converting scientific articles with complex content, including a large number of formulas, tables and figures, typed in the most commonly used .docx and LaTeX formats into this format is not an easy task and the available software tools either do not cope with it in full or are quite expensive. The paper proposes an approach to creating a converter of scientific articles from the .docx format to HTML and JATS XML formats leveraging the open source tool Mammoth and describes a prototype converter of KIAM preprints to HTML with subsequent conversion to JATS XML, created based on this approach.

Keywords: journal article HTML-version, JATS XML, conversion of scholarly article formats, KIAM preprints

В настоящее время подавляющее большинство научных журналов имеет онлайн-версии и предоставляет полные тексты статей для открытого доступа или на коммерческой основе. Помимо традиционной формы представления полных текстов – формата PDF – многие издательства публикуют полные версии научных статей в HTML-формате. Каждый из этих форматов имеет свои преимущества и недостатки, подробно изложенные в работе [1]. Основные преимущества HTML-формата

- в лучшей структуризации материала, что позволяет быстрее ориентироваться в нем и находить нужный контент;
- в возможности адаптации под различные размеры экрана;
- в предоставляемой браузерами возможности автоматического перевода на другие языки;
- в наличии форматов масштабируемого представления формул, пригодного для машинной обработки и поиска;
- и, самое существенное, в возможности добавления мультимедийного контента и расширения функционала разного рода интерактивными и динамическими возможностями.

PDF-формат более удобен для чтения офлайн и для обмена содержимым статей. Ни один из форматов PDF или HTML на данном этапе не обладает абсолютным преимуществом перед другим, поэтому издательства стараются предоставлять контент в обоих форматах. В связи с этим весьма актуальной является задача организации процесса получения двух синхронизованных между собой версий научной статьи из материала, присланного автором. Несмотря на то, что история публикаций полных текстов научных статей в HTML-формате насчитывает уже порядка 30 лет, единого подхода к организации этого процесса и доступного для всех инструментария за это время не выработано. В работе предложен один из возможных вариантов организации процесса получения PDF- и HTML-

версий научной статьи из исходного материала в формате текстового процессора Word, ориентированный на малые издательства с ограниченным бюджетом, и описан создаваемый в рамках такого подхода конвертер препринтов ИПИМ из формата .docx в формат HTML и формат JATS XML – NISO-стандарт XML-представления журнальных статей [4].

1. Преимущества и недостатки существующих подходов и программных инструментов

В работах [2, 3] описаны применяемые издательствами подходы к созданию синхронизированных между собой PDF- и HTML-версий журнальных статей и программные инструменты, используемые в случае, если исходные материалы поступают в формате текстового процессора Word.

Наиболее распространенным подходом, получившим название XML-First, является предварительное создание XML-версии журнальной статьи в соответствии со стандартом, принятым в издательстве, с последующим автоматическим преобразованием ее в HTML и PDF (Рис. 1).

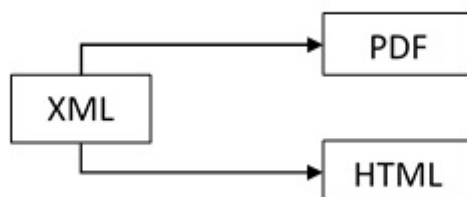


Рис. 1. Подход XML-First

XML-разметка отражает структуру статьи и обычно включает деление на титульную часть, содержащую метаданные, тело статьи – ее содержание – и справочную часть, включающую библиографический список, приложения и т.п. В ней, как правило, предусматриваются отдельные элементы для названия, авторов, аннотации, разделов статьи и их заголовков, рисунков, формул, таблиц, библиографических ссылок и т.д.

Основное преимущество XML-представления статьи в отделении контента от его визуального представления. Это упрощает хранение статей, обмен ими и преобразование в различные форматы. Полные тексты статей в XML-формате можно хранить в базе данных и формировать HTML динамически по запросу. Хранение полных текстов статей в базе данных делает возможным поиск не только по метаданным, но и по содержанию. К преимуществам XML-формата можно отнести также возможность добавления семантики предметных областей за счет включения элементов и атрибутов, отражающих конкретные научные понятия с использованием

определенных словарей и онтологий, что способствует продвижению в направлении формализации научного знания.

Изначально разные издательства использовали разные XML-схемы, но необходимость обмена журнальными статьями и хранения их в электронных библиотеках потребовала выработки для этой цели единого стандарта. Такой стандарт был разработан в Национальной медицинской библиотеке США. Первоначально он предназначался для хранения в архиве PubMed, поддерживаемого этой библиотекой, но затем стал использоваться и другими организациями. В процессе эксплуатации выяснилось, что формат удобен не только для хранения и обмена, но и для подготовки научных статей к публикации. После доработки совместно с другими организациями, этот формат, получивший название Journal Article Tag Suite, сокращенно JATS, стал стандартом NISO. Стандарт включает в себя три модели, имеющие разные назначения:

- «архивно-обменную» – Journal Archiving and Interchange Tag Set – для обмена журнальными статьями и хранения их в репозиториях, объединяющих статьи из разных изданий;
- «издательскую» – Journal Publishing Tag Set – для разметки статьи, публикуемой в конкретном журнале;
- и «авторскую» – Article Authoring Tag Set – для первоначального ввода контента статьи без привязки к конкретному журналу.

Эти модели в значительной степени совпадают, но имеются и отличия, обусловленные их назначением. Например, в «авторской» модели нет таких элементов, как название журнала, ISSN и т.п.; в «архивно-обменной» модели тело статьи не является обязательным элементом – для хранения и обмена могут быть предназначены только метаданные.

Подход XML-First безусловно имеет много преимуществ – при наличии XML-версии преобразование в другие форматы становится делом техники – однако получение XML является довольно сложной задачей. Издательствам было бы удобно, чтобы авторы присылали статьи уже набранные в формате JATS XML, однако на практике добиться этого не удастся [5]: авторам проще набирать тексты в привычной среде Word или TeX, к тому же статья часто пишется до принятия решения, в каком журнале она будет опубликована, а в разных журналах могут быть свои особенности применения стандарта JATS, например, требования к наличию определенных элементов и их порядку (JATS XML в этом смысле является достаточно гибким стандартом). Поэтому издательства либо нанимают дополнительной персонал для ручного ввода в XML, либо используют коммерческие или разработанные собственными силами конвертеры из исходных форматов в XML. Ручной ввод требует значительных финансовых затрат на оплату дополнительного персонала и довольно много времени на сам ввод текста. Использование программ-конвертеров позволяет сократить временные затраты, однако имеющиеся в свободном доступе конвертеры из

формата .docx не справляются с задачей в полном объеме (в них отсутствует возможность конвертации формул, плохо конвертируются таблицы и т.п.), а коммерческие продукты довольно дороги, хотя и обходятся дешевле, чем ручной труд по вводу данных. К тому же, полностью автоматических коммерческих конвертеров произвольных (“сырых”) документов Word в JATS XML не существует. Причина – в отсутствии в формате .docx семантики, отражающей структуру научной статьи. Семантика вносится либо путем разметки документа Word специальными стилями, как в Inera eXtyles [6] – самом распространенном инструменте для преобразования документов Word в JATS XML – либо программным путем с использованием технологий искусственного интеллекта, как в разработке компании Ictect [7]. В первом случае нужна предварительная работа по разметке исходного документа Word, во втором основное время уходит на доработку выходного документа. Стоит отметить также, что основанные на технологиях искусственного интеллекта конвертеры обучались на англоязычном материале и к русскоязычному материалу не применимы.

Несмотря на наличие различных программ-конвертеров и накопленный опыт работы с ними, преобразование текстов научных статей из формата Word в формат JATS XML остается довольно трудоемким и/или финансово затратным. Поэтому, помимо подхода XML-First, рассматриваются и другие подходы. Например, авторы работы [8] предлагают использовать HTML как промежуточный формат при преобразовании в JATS XML, мотивируя это, в частности, тем, что семантику в HTML проще визуализировать. Разработанный ими конвертер с открытым исходным кодом xSweet [9] преобразует “сырые” документы Word в HTML с сохранением всей информации о форматировании. Семантику научной статьи предлагается вносить в HTML с помощью Word-подобного HTML-редактора Wax-JATS, встроенного в издательскую платформу Kotahī. Формулы в формате MathType конвертер xSweet не поддерживает, их можно только повторно ввести в редакторе в формате TeX. Поскольку значительная часть препринтов ИПМ содержит большое число формул, для наших целей этот инструмент не подходит.

2. Схема рабочего процесса для издательства с малым бюджетом

В итоге, после анализа рынка программного обеспечения для автоматизации процесса получения HTML-версий научных статей, нами было принято решение заняться разработкой собственного конвертера препринтов из формата .docx в HTML и JATS XML, отказавшись от подхода XML-first из-за трудностей, связанных с его реализацией.

Так же, как и разработчики Kotahī, мы считаем, что удобнее сначала конвертировать документ Word в HTML, а затем уже HTML конвертировать в JATS XML. Семантику JATS можно реализовать в HTML при помощи

классов и атрибутов, которые при преобразовании в XML перейдут в элементы и атрибуты JATS. В HTML, в отличие от XML, эту семантику можно легко визуализировать, создав, например, отладочный CSS, в котором нужные классы и атрибуты выделяются цветом. Однако, в отличие от подхода, реализуемого в платформе Kotahi, мы предлагаем вносить семантику не в документ HTML, а непосредственно в документ Word при помощи специальных стилей, аналогично подходу, реализованному в инструменте Inera eXtyles, и использовать эти стили при конвертации в HTML. При таком подходе работник редакции остается в привычной среде, ему нет необходимости осваивать новые инструменты.

Рабочий процесс предлагается выстроить следующим образом (Рис. 2).

Шаг 1. Присланный автором документ Word редактируется, преобразуется в PDF и публикуется, так же, как это делалось и ранее, до внедрения HTML-версий.

Шаг 2. Работник редакции копирует содержимое документа Word в специально созданный шаблон, содержащий семантические стили, и производит разметку этими стилями.

Шаг 3. Размеченный семантическими стилями документ Word автоматически преобразуется в HTML; с помощью отладочного CSS выявляются ошибки семантической разметки и исправляются в Word. После этого документ Word снова конвертируется в HTML.

Шаг 4. В отлаженный HTML вручную добавляются метаданные, отсутствующие в исходном документе Word, HTML конвертируется в XML, и оба документа публикуются.

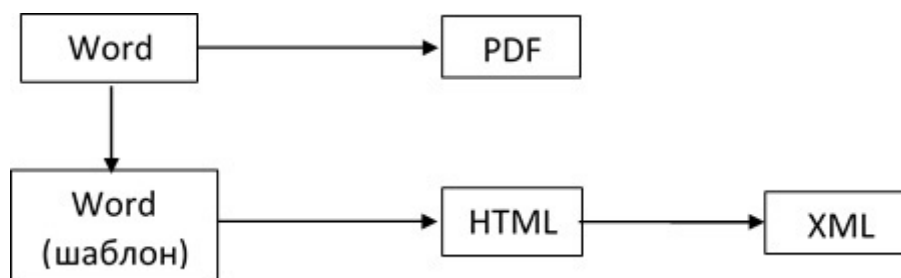


Рис. 2. Схема рабочего процесса

В этой схеме рабочего процесса полученный HTML является окончательным публикуемым документом, а XML создается лишь с целью предоставления возможности желающим иметь копию статьи, пригодную для машинной обработки. В дальнейшем, когда будет накоплено достаточное число XML-документов, может быть создана база данных, при этом полученный конвертацией из Word HTML-документ будет нужен только как промежуточный, а окончательный HTML будет создаваться динамически преобразованием из XML.

3. Палитра стилей и элементы JATS XML

Мы старались минимизировать набор стилей для семантической разметки, чтобы не слишком усложнять работу редактора. За основу были взяты стили из шаблона, рекомендованного авторам для оформления институтских препринтов, большинство из которых являются семантическими. В этом шаблоне стилей немного, порядка десятка, к ним было добавлено еще два десятка. На данный момент используется 24 стиля абзаца, 5 знаковых стилей (Рис. 3) и 3 табличных стили. Табличными стилями помечаются таблицы, используемые для форматирования формул, групп формул и набора рисунков, чтобы отличать их от собственно таблиц.

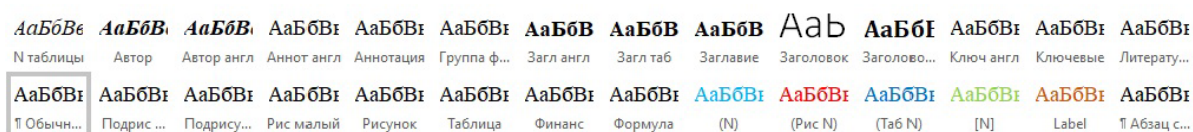


Рис. 3. Стили семантической разметки

Степень гранулированности JATS XML при таком числе стилей получается не очень высокой. Данный набор стилей позволяет конвертеру выделить следующие элементы:

среди метаданных

- заглавие и его английский вариант;
- авторов и перевод их на английский;
- аннотацию и ее английский вариант;
- ключевые слова и их английский вариант;
- сведения о финансировании;

в теле статьи

- разделы с заголовками;
- абзацы;
- рисунки с подрисуночной подписью;
- контейнеры таблиц, включающие номер таблицы, заглавие и собственно таблицу;
- формулы с метками;
- группы формул с метками;
- сноски;
- ссылки на литературу, рисунки, таблицы, формулы и группы формул;

в справочной части

- библиографический список с заголовком;
- отдельные библиографические ссылки с метками.

В дальнейшем набор стилей может быть расширен, однако существенное увеличение числа стилей приведет к тому, что в них сложно

будет ориентироваться, и понадобится специальный плагин для облегчения процесса разметки, как это сделано в Inera eXtyles.

5. Реализация конвертера

За основу для разработки конвертера препринтов ИПМ из формата .docx в HTML был взят инструмент с открытым исходным кодом Mammoth [10], написанный английским программистом Майклом Уильямсоном. При преобразовании Mammoth использует только информацию о стилях, игнорируя такие детали, как шрифт, размер текста, цвет и т.п., например, встроенный стиль «Заголовок 1» по умолчанию преобразуется в элемент h1. Имеется возможность настройки преобразования при помощи таблицы соответствия стилей: стилю можно поставить в соответствие элемент или набор вложенных друг в друга элементов с классами и атрибутами. Например, абзацу со стилем «Рисунок» можно поставить в соответствие элемент div с классом `jats-graphic`, последовательности символов со стилем «[N]» – элемент a с классом `jats-xref` и атрибутом `data-jats-ref-type='bibr'`, который будет работать как ссылка на элемент из списка литературы, и т.д. Тем самым создается возможность внесения семантики, отражающей структуру научной статьи, в HTML-документ. Наличие такой возможности определило выбор этого инструмента среди бесплатных конвертеров .docx в HTML. Отметим также, что Mammoth поддерживает основные структурные единицы документа – списки, изображения, таблицы (правда, без форматирования), сноски – и позволяет произвести настройку преобразования изображений путем переопределения метода `convertImage`.

Конвертер Mammoth имеет реализации на нескольких языках программирования: Python, JavaScript, Java и C#, однако последняя не написана вручную, а получена автоматически из реализации на Java. Код C#, полученный автоматическим преобразованием, не читабелен, его трудно модифицировать вручную, из-за чего теряются преимущества открытого исходного кода. К тому же, обновления C#-реализации производятся с большим опозданием по сравнению с другими реализациями. Поэтому мы от нее отказались. Из реализаций, написанных вручную, для нас более удобной оказалась реализация на Java.

Разрабатываемый нами конвертер использует Mammoth на первом шаге для получения документа HTML, а затем осуществляет автоматическую постобработку для достижения полного соответствия документа HTML нужной структуре. Постобработкой осуществляется разбиение на разделы, удаление таблиц, которые применялись для форматирования формул, объединение рисунков с их подписями в одном контейнере, расстановка ссылок на литературу, рисунки, таблицы, формулы и т.д.

Mammoth не поддерживает формулы, поэтому конвертацию формул пришлось реализовывать самим. Был написан конвертер формул, основанный на C#-библиотеке MathType SDK, который каждую формулу формата MathType из документа Word преобразует в формат MathML и записывает в отдельный текстовый файл. Эти файлы используются затем конвертером из .docx в HTML: формулы считываются из файлов и записываются в нужные места.

На сегодняшний день реализован прототип конвертера .docx в HTML, который работает с документами Word, при условии соблюдения ряда ограничений:

- подписи к рисункам должны быть расположены под рисунками;
- номера формул должны быть справа от формул;
- номера и заглавия таблиц должны быть сверху от таблиц;
- библиографические ссылки должны либо все иметь метки, либо каждая библиографическая ссылка должна содержаться в одном абзаце;
- не допускаются многоуровневые разделы;
- не допускаются рисунки в формате wmf.

Последние два ограничения в дальнейшем могут быть сняты. Конвертер не поддерживает OLE-объекты, фигуры SmartArt и т.п. Подобные объекты должны быть преобразованы в изображения или таблицы.

Прототип конвертера .docx в HTML находится в стадии тестирования. Конвертер из HTML в JATS XML на данный момент реализован частично. Планируем в течение года создать рабочие версии обоих конвертеров и начать внедрение.

4. Пример разметки и результат преобразования

На рисунке 4 изображен фрагмент препринта в редакторе Word, а на рисунках 5, 6 результат его автоматического преобразования в HTML: код и его представление в браузере.

Форматирование группы формул автор препринта выполнил при помощи таблицы. Перед преобразованием в HTML редактор применил к этой таблице специальный табличный стиль, соответствующий группе формул, так что форматизирующая таблица оказалась выделенной цветом. Специальные семантические стили были применены к метке с номером формулы, ссылкам на формулы и на литературу. Они также выделены цветом.

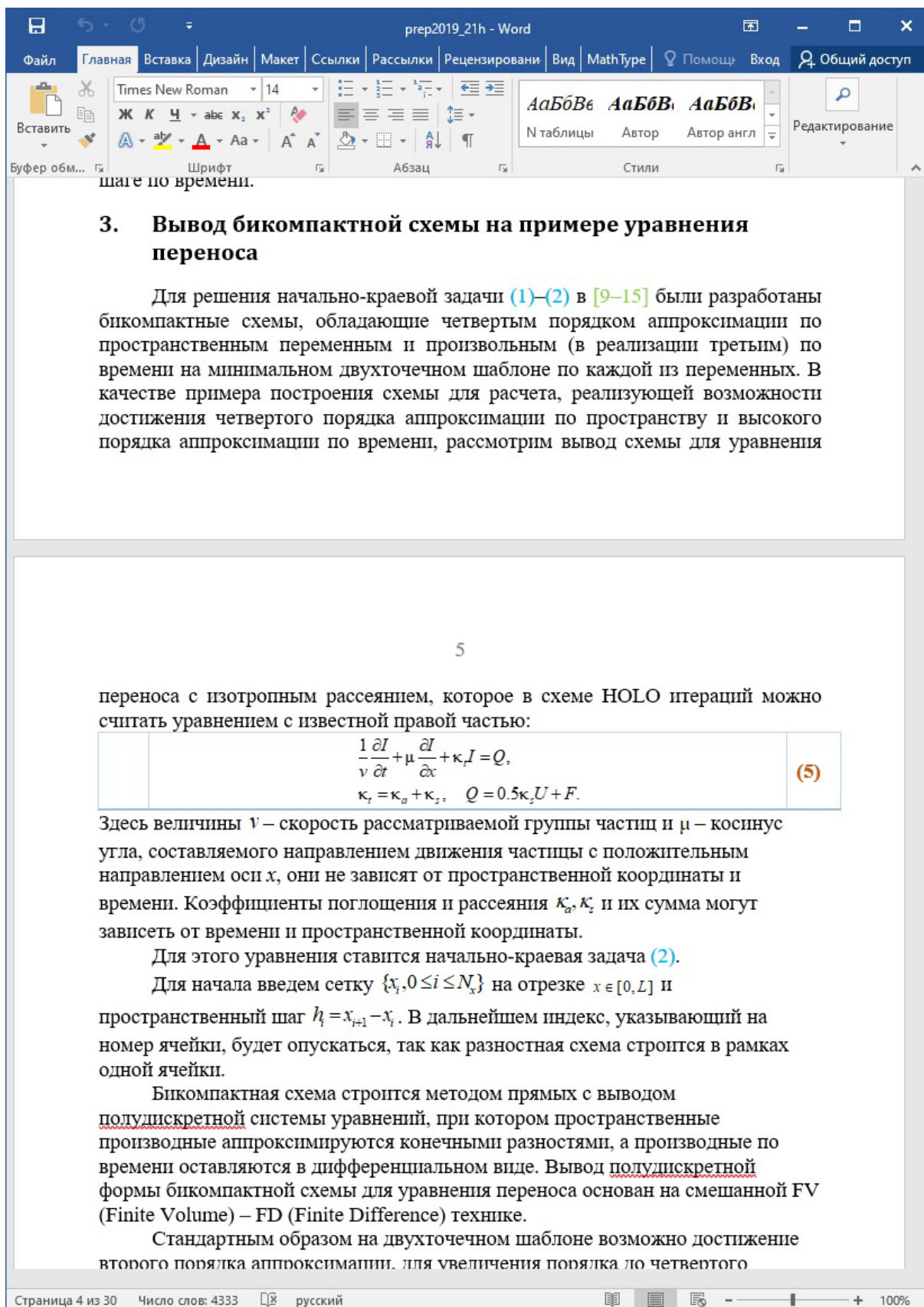


Рис. 4. Фрагмент препринта в редакторе Word

```

<section class="jats-sec" id="sec3">
  <h2 class="jats-title">3. Вывод бикompактной схемы на примере уравнения переноса</h2>
  <p>
    Для решения начально-краевой задачи
    <a class="jats-xref" data-jats-ref-type="disp-formula" href="#fm11">(1)</a>
    <strong>-</strong>
    <a class="jats-xref" data-jats-ref-type="disp-formula" href="#fm12">(2)</a> в
    <a class="jats-xref" data-jats-ref-type="bibr" href="#bibr9">[9-15]</a>
    были разработаны бикompактные схемы, обладающие четвертым порядком аппроксимации по
  </p>
  <figure class="jats-disp-formula-group" id="fm15">
    <div>
      <div>
        <p><span><math>...</math></span></p>
      </div>
      <div>
        <p><span class="jats-label">(5)</span></p>
      </div>
    </div>
  </figure>
  <p>
    Здесь величины <span><math>...</math></span> – скорость рассматриваемой группы части
    <span><math>...</math></span> – косинус угла, составляемого направлением движения ча
    <span><math>...</math></span> и их сумма могут зависеть от времени и пространственн
  </p>
  <p>Для этого уравнения ставится начально-краевая задача
    <a class="jats-xref" data-jats-ref-type="disp-formula" href="#fm12">(2)</a>.
  </p>
  <p>
    Для начала введем сетку <span><math>...</math></span> на отрезке
    <span><math>...</math></span> и пространственный шаг <span><math>...</math></span>.
    В дальнейшем индекс, указывающий на номер ячейки, будет опускаться, так как разности
  </p>
</section>

```

Рис. 5. Фрагмент результата автоматической конвертации в HTML. Код

В результате автоматического преобразования таблица с группой формул перешла в элемент `figure` с классом `jats-disp-formula-group` и атрибутом `id="fm15"`, формулы были преобразованы в MathML, а номер формулы – в элемент `span` с классом `jats-label`. Ссылки на формулы и литературу перешли в элементы `a` с классом `jats-xref` и атрибутами `data-jats-ref-type="disp-formula"` и `data-jats-ref-type="bibr"` соответственно, а также атрибутом `href`, указывающим на соответствующую формулу или элемент библиографического списка. Тем самым автоматически реализована возможность переходить по ссылкам на формулы и литературу. В дальнейшем, наличие подобных ссылок на литературу позволяет написать JavaScript-код, реализующий всплывающие подсказки с текстом библиографической ссылки при наведении мыши на ее номер внутри основного текста препринта.

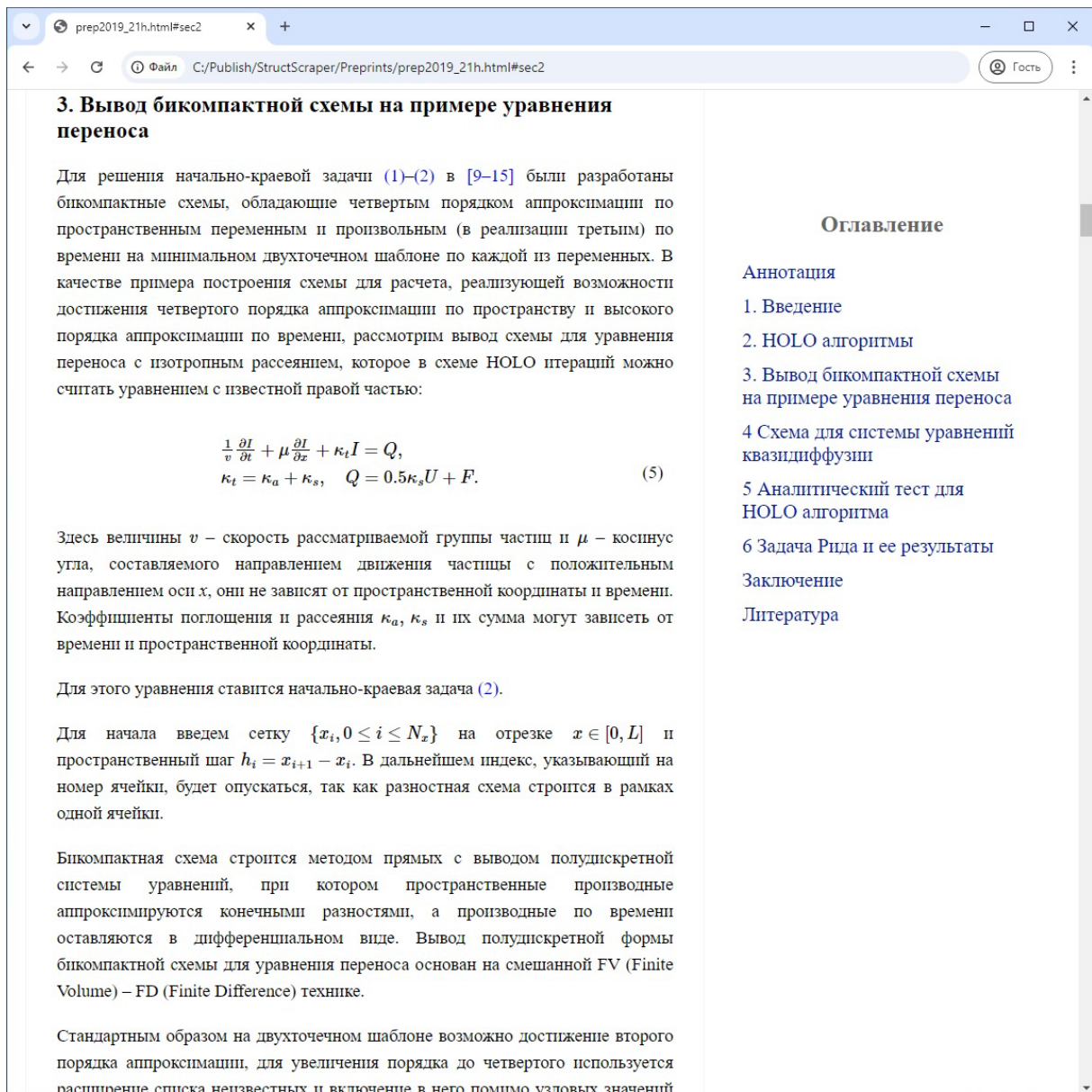


Рис. 6. Фрагмент результата автоматической конвертации в HTML

Литература

1. Чебуков Д.Е. Об HTML версии полного текста научной статьи // Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции (17–22 сентября 2018 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2018. — С. 487–498. — <https://doi.org/10.20948/abrau-2018-16>
2. Скорнякова Р.Ю. Методы и инструменты, используемые при подготовке публикаций научных статей в формате HTML // Электронные библиотеки. 2023. Том 26, № 2 (май 2023), С. 252–302. — <https://doi.org/10.26907/1562-5419-2023-26-2-252-302>, <https://rdl-journal.ru/article/view/774>

3. Скорнякова Р.Ю. Обзор программных средств для создания HTML-версии журнальной статьи из исходного материала в формате Word // Научный сервис в сети Интернет: труды XXV Всероссийской научной конференции (18-21 сентября 2023 г., онлайн). — М.: ИПМ им. М.В.Келдыша, 2023. — С. 332-344. — <https://doi.org/10.20948/abrau-2023-38>
4. Standardized Markup for Journal Articles: Journal Article Tag Suite (JATS) // NISO website, 07.07.2021. — <https://www.niso.org/standards-committees/jats>
5. Kasdorf W.E. Getting from Word to JATS XML // The Association of Learned and Professional Society Publishers blog. 18.10.2018. — <https://blog.alpsp.org/2018/10/getting-from-word-to-jats-xml.html>
6. Inera JATS Solutions. — <https://www.inera.com/jats-solutions/>
7. Ictect Intelligent Content for Journals. — <https://ictect.com/JATS-XML>
8. Visel D., Hyde A., Whitmore B. Kotahi: a new JATS production system // Proceedings of Journal Article Tag Suite Conference (JATS-Con), Bethesda (MD), USA, April 25–26, 2022. — <https://www.ncbi.nlm.nih.gov/books/NBK425546/>
9. xSweet. The open .docx to HTML conversion tool. — <https://xsweet.org/>
10. Mammoth. .docx to HTML converter — <https://mike.zwobble.org/projects/mammoth/>

References

1. Chebukov D.E. Ob HTML versii polnogo teksta nauchnoi stati // Nauchnyi servis v seti Internet: trudy XXI Vserossiiskoi nauchnoi konferentsii (17–22 sentiabria 2018 g., g. Novorossiisk). — М.: ИПМ им. М.В.Келдыша, 2018. — С. 487–498. — <https://doi.org/10.20948/abrau-2018-16>
2. Skorniakova R.Iu. Metody i instrumenty, ispolzuemye pri podgotovke publikatsii nauchnykh statei v formate HTML // Elektronnye biblioteki. 2023. Tom 26, № 2 (mai 2023), S. 252–302. — <https://doi.org/10.26907/1562-5419-2023-26-2-252-302>, <https://rdl-journal.ru/article/view/774>
3. Skorniakova R.Iu. Obzor programmnykh sredstv dlia sozdaniia HTML-versii zhurnalnoi stati iz iskhodnogo materiala v formate Word // Nauchnyi servis v seti Internet: trudy XXV Vserossiiskoi nauchnoi konferentsii (18-21 sentiabria 2023 g., onlain). — М.: ИПМ им. М.В.Келдыша, 2023. — С. 332-344. — <https://doi.org/10.20948/abrau-2023-38>
4. Standardized Markup for Journal Articles: Journal Article Tag Suite (JATS) // NISO website, 07.07.2021. — <https://www.niso.org/standards-committees/jats>

5. Kasdorf W.E. Getting from Word to JATS XML // The Association of Learned and Professional Society Publishers blog. 18.10.2018. — <https://blog.alpsp.org/2018/10/getting-from-word-to-jats-xml.html>
6. Inera JATS Solutions. — <https://www.inera.com/jats-solutions/>
7. Ictect Intelligent Content for Journals. — <https://ictect.com/JATS-XML>
8. Visel D., Hyde A., Whitmore B. Kotahi: a new JATS production system // Proceedings of Journal Article Tag Suite Conference (JATS-Con), Bethesda (MD), USA, April 25–26, 2022. — <https://www.ncbi.nlm.nih.gov/books/NBK425546/>
9. xSweet. The open .docx to HTML conversion tool. — <https://xsweet.org/>
10. Mammoth. .docx to HTML converter — <https://mike.zwobble.org/projects/mammoth/>