Министерство общего и профессионального образования РФ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

(НГУ)

УДК 025.4.03; 002.53:681.3.016

002.009.7:330.163; 025.4.03

ГРНТИ 20.23.21, 20.51.17

№ госрегистрации

Инв. №

УТВЕРЖДАЮ
Проректор по научной работе, профессор
_____Г.Ю.Шведенков
«_____» _____ _____г.

ОТЧЕТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ
РАЗРАБОТКА ТЕХНОЛОГИИ ПРЕДСТАВЛЕНИЯ НАУЧНО-ТЕХНИЧЕСКИХ ПУБЛИКАЦИЙ В ИНТЕРНЕТЕ И ПЕЧАТНОМ ВИДЕ НА ОСНОВЕ ЕДИНОГО ХРАНИЛИЩА ИНФОРМАЦИИ

(итоговый отчет за 2002 г.)

Зам. проректора
по научной работе _____Ю.Н. Попов

Руководитель темы
доцент, к.т.н. _____В.С. Черкасский

Новосибирск 2002

СПИСОК ИСПОЛНИТЕЛЕЙ

Руководитель работы, доцент, канд. техн. наук	_____	В.С. Черкасский (введение, заключение, раздел 2-3)
Ответственный исполнитель, профессор, д-р физ.-мат. наук	_____	И.А. Котельников (раздел1-2, 4, заключение)
Профессор, д-р физ.-мат. наук	_____	В.И. Яковлев (раздел 4)
Научный сотрудник	_____	А.Н. Матвеенко (раздел 3, 5)
Ведущий программист	_____	Н.А. Витюгова (раздел 5)
Магистрант	_____	П.Д. Рудыч раздел (3, 5)

РЕФЕРАТ

Отчет 65 с., 4 рис., 2 табл., 17 источников.

НАУЧНО-ТЕХНИЧЕСКАЯ ИНФОРМАЦИЯ, ПРЕДСТАВЛЕНИЕ В ИНТЕРНЕТЕ, ФОРМАТ LATEX, PDF, HTML, XML, ПРЕОБРАЗОВАНИЕ В ФОРМАТ HTML, ЭЛЕКТРОННЫЕ ПУБЛИКАЦИИ, ГИПЕРТЕКСТ, ЯЗЫК МАТЕМАТИЧЕСКОЙ РАЗМЕТКИ MATHML, МОДЕЛИРОВАНИЕ, ФИЗИЧЕСКИЕ МОДЕЛИ, МОДЕЛИРОВАНИЕ В ИНТЕРНЕТЕ, MATLAB, WEBSERVER.

Изучаются методы представления научно-технической информации в электронном виде, возможность и способы представления ее в интернете. Проведено сравнение различных методов представления научно-технической информации в интернете, рассмотрены популярные форматы LaTeX, PDF и языки разметки HTML (XHTML), XML. Сделан вывод, что наиболее подходящим способом представления информации является язык разметки HTML (XHTML) с элементами MathML. Для преобразования больших объемов накопленной информации из формата LaTeX в форматы (X)HTML+MathML и XML+MathML выбран пакет TeX4HT. Разработан ряд скриптов, позволяющих улучшить качество преобразования. Отработаны основные элементы технологии такого преобразования на примере первой части курса классической электродинамики. Выявлены слабые места имеющихся средств и намечены пути их улучшения через модернизацию системы навигации и интерфейса преобразованных материалов.

Предложена и апробирована технология разработки интерактивных динамических моделей физических процессов для представления их в интернете. Технология базируется на системе MATLAB и пакете Matlab WEB server, которые позволяют в режиме удаленного доступа отправлять на сервер задания на расчет по разработанным моделям с последующим получением результатов на компьютер пользователя по протоколу HTTP. На основании предложенной технологии разработано 8 задач по классической электродинамике, которые размещены на web-сервере физического факультета НГУ. Сформулированы задачи для дальнейшего совершенствования технологии – повышение скорости генерации анимационных иллюстраций и улучшение интерфейса взаимодействия пользователя с системой.

СОДЕРЖАНИЕ

Введение……..6
1 Представление научных материалов в интернете……9
1.1 Представление научных текстов в формате LaTeX..... 9
1.1.1 TechExplorer..... 9
1.1.2 WebEQ..... 10
1.2 Представление научных текстов в формате PDF ..... …… 10
1.3 Представление научных текстов в формате HTML..... 13
1.3.1 HTML и математика 13
1.3.2 Математический язык разметки 15
1.3.3 Особенности математической нотации 16
1.3.4 История MathML..... 17
1.3.5 Проект MathML..... 18
1.4 Интерактивное представление результатов расчетов (моделирования) в виде графиков и мультфильмов 20
2 Конвертация научных материалов в HTML 21
2.1 Пакет LaTeX2HTM 21
2.1.1 Несколько исторических замечаний 22
2.1.2 Принципы создания Web-документов 22
2.2 Пакет TeX4HT 23
2.2.1 От LaTeXa к DVI..... 24
2.2.2 От DVI к HTML..... 24
2.2.3 От DVI к GIF 25
2.2.4 От DVI к XML 25
2.2.5 XML? Этого мало..... 26
2.2.6 Составные части пакета TeX4HT и вспомогательные программы (MikTeX, ImageMagic, Aladdin GhostScript) 27
2.2.7 Установка рабочей среды для преобразования и ее обновление 28
2.2.8 Локализация и отладка пакета, использование кодировки Unicode 33
2.2.9 Технология преобразования..... 38
2.2.10 Форматы представления графических материалов 40
3 Представление моделей физических явлений в интернете 42
3.1 Специализированные программы, использование web-серверов 43
3.1.1 Выбор языка программирования..... 43

3.1.2 Закрытые и открытые моделирующие программы ... 43
3.1.3 Исполнение программ на стороне клиента и стороне сервера ... 44
3.1.4 Совмещение HTML и Matlab ... 44
3.2 Matlab web-сервер и технология его использования ... 45
3.2.1 Структура каталогов (папок) и технология подготовки моделей ... 45
3.2.2 Статические и динамические модели и создание «мультфильмов» ... 52
4 Подготовка и представление 1-й части курса электродинамики в интернете . 53
4.1 Содержание и структура конспекта ... 53
4.2 Особенности подготовленного текста и их реализация в LaTeX ... 55
4.3 Проблемы и пути их решения ... 56
5 Модели физических явлений в интернете ... 57
Заключение ... 58
Список использованных источников ... 59
Приложение А Скрипт UTF2WIN ... 61
Приложение Б Основной файл учебника ... 63

Введение

В настоящее время все большую актуальность приобретает задача хранения и представления больших объемов накопленной научно-технической информации в единообразном виде, причем это представление (и способ ее хранения) должны предоставлять возможность как воспроизводить эту информацию в печатном виде с типографским качеством, так и представлять эту информацию в интернете. Каждая из этих двух задач (воспроизведение на экране и подготовка к печати), конечно, имеет свои решения, но при раздельных способах хранения и представления информации возникает проблема актуализации этой информации, т.е. внесения соответствующих изменений в обе части фактически одного и того же документа (под документом здесь и далее мы понимаем текстовой документ научно-технического содержания, как правило, в области естественных наук). Поэтому желательно хранить документ в одном виде, а с помощью определенных процедур переводить его в то или иное представление.

Что касается подготовки соответствующих документов, то в значительной части научного и педагогического сообщества принято использовать систему TeX (LaTeX) [1-3], изобретенную более 20 лет назад. В настоящее время LaTeX обеспечивает наилучшее качество печатных текстов научно-технического содержания. За 20 лет накоплены огромные массивы документов, статей, учебных пособий, подготовленных в системе LaTeX. Большинство научных журналов принимают научные статье в электронном виде в формате LaTeX.

С развитием интернета все большее распространение приобретают электронные публикации научных и учебных материалов. Академические и коммерческие исследовательские группы постоянно выпускают большое количество научного материала. Все больше и больше научных публикаций размещается в базах данных, таких как довольно известный архив препринтов по физике и математике Национальной Лаборатории в Лос-Аламосе (Los Alamos). В особенности это относится к некоторым областям физики и математики, где цены на академические журналы весьма высоки. К тому же, базы данных с информацией о математических исследованиях, такие как Mathematical Reviews и Zentralblatt für Mathematik, предлагают огромное количество ссылок на содержащиеся в интернете математические документы. Многие научные журналы публикуют электронные версии печатных изданий. Та же тенденция наблюдается и в области обучения. Зачастую используемый для целей публикации в интернете формат PDF, несмотря на свою кажущуюся универсальность, обладает целым рядом недостатков, а использование HTML для представления материалов с большим количеством математических текстов до настоящего времени весьма ограничено, несмотря на наличие около 20 программных продуктов, предназначенных для этой цели. Постоянно обновляемый список таких продуктов можно найти на сайте консорциума

W3C, например, по адресу http://www.w3.org/Math/iandi/impl-interop20010220.html. Трудно не растеряться в таком многообразии. Некоторые технологии устарели, некоторые решают только очень узкую задачу, некоторые явно отстали из-за неадекватного маркетинга. Другие еще слишком молоды, чтобы понять, будет ли из них толк. Единственное, что можно констатировать с полной уверенностью — так это то, что решение проблемы отображения математических текстов в интернете далеко от идеала. И это несмотря на то, что именно ученые изобрели интернет — принято считать, что интернет родился в ЦЕРНе — крупнейшем международном научном центре. [Для справки: протокол TCP/IP стал преобладающим в ЦЕРНе к 1989 году; презентация Web состоялась в июле 1990 года на семинаре в ЦЕРНе; мировое сообщество физиков, работающих в области высоких энергий, узнало об этом из 204 номера CERN Computer Newsletters в декабре 1991 года].

Основная задача, которая решалась в рамках настоящего проекта в 2002 году – анализ существующих технологий преобразования научно-технических текстов, созданных в формате LaTeX, в формат, подходящий для представления в интернет и выбор базовой технологии, с помощью которой который можно осуществлять такое преобразование. Кроме того, выбранный вариант технологии дорабатывался для нужд представления научных и учебных материалов на примере преобразования первой части учебника по электродинамике, который создавался участниками данного проекта параллельно с разработкой технологии его представления. На основании анализа имеющихся данных авторы пришли к выводу, что наилучшим способом представления данных является формат XHTML с представлением формул с помощью языка разметки MathML, просмотр полученных материалов в интернете с помощью наиболее популярного в настоящее время браузера Microsoft Internet Explorer, дополненного плагином MathPlayer (http://www.dessci.com/webmath/mathplayer/), а наиболее подходящим средством преобразования является пакет программ TeX4HT, разработанный Эйтаном Гурари (http://www.cis.ohio-state.edu/~gurari/TeX4HT/).

Особенностью представления учебных материалов по сравнению с чисто научными является желательность использования помимо статических текстов, формул и графиков еще и интерактивных моделей, которые позволяют обучаемому самостоятельно поработать с моделями описываемых явлений и, тем самым, лучше усвоить изучаемый материал. Для интерактивного взаимодействия через интернет с разработанными моделями использовалась технология, основанная на web-сервере MatLab [4]. Сервер MatLab позволяет по запросу пользователя и в соответствии с заданным им набором параметров динамически проводить расчеты по заранее подготовленной модели, генерировать графические и видео- изображения, которые затем передаются в сервер IIS для передачи пользователю через интернет.

В качестве фактического материала, который использовался для отработки вышеупомянутых технологий, использовалась первая часть курса классической электродинамики, включающая

теоретический материал, моделирующие программы на языке Matlaba, и наборы задач, размещенных в интернете.

1 Представление научных материалов в интернете

1.1 Представление научных текстов в формате LaTeX

Существует несколько приложений, которые способны отображать исходный текст в разметке LaTeX непосредственно в окне Web-навигатора в виде, близком к откомпилированному документу LaTeX. Такие приложения представляют собой либо динамически подключаемый дополнительный программный модуль (plug-in), либо Java- апплет. Программные средства такого рода принимают урезанный в той или иной степени исходный текст LaTeX; они не воспроизводят DVI-файлы (т.е. откомпилированные документы LaTeX) и не используют также шрифты METAFONT, подменяя их шрифтами, имеющимися в операционной системе. В настоящее время ни одно из таких приложений не воспроизводит полностью все возможности LaTeXa, однако целый ряд разработчиков работает в этом направлении [5].

Наиболее часто используемым динамически подключаемым модулем, расширяющим возможности Web-навигаторов в части отображения LaTeXовских текстов, является гипермедийный навигатор techexplorer (http://www-3.ibm.com/software/network/techexplorer/) фирмы IBM. Он предназначен для программ Netscape Navigator и Microsoft Internet Explorer и имеет варианты для нескольких программно-аппаратных платформ. Данный модуль позволяет воспроизводить значительную часть нематематической LaTeXовской разметки и пригоден как для отображения содержащихся в HTML-документах математических формул, так и для воспроизведения документов в целом. Существенным недостатком techexplorer является невозможность отображения им текстов на кириллице.

Для воспроизведения в навигаторе математических формул часто применяется Java-апплет WebEQ. Будучи написанным на языке Java, он совместим со многими браузерами, работающими на различных платформах. Апплет WebEQ обладает богатыми функциональными возможностями в части воспроизведения математических выражений, но в отношении работы с текстом его возможности существенно ограничены.

1.1.1 TechExplorer

Разработанный фирмой IBM плагин techexplorer представляет собой динамически подключаемый модуль, предназначенный для навигаторов Netscape Navigator и Microsoft Internet Explorer. Techexplorer распознает ограниченное подмножество команд TeXa и LaTeXa и может использоваться как для изображения встроенных в HTML-страницу математических выражений, так и для воспроизведения целых документов в окне навигатора.

Плагин techexplorer претендует на большее, нежели просто отображение разметки научного текста. Его вводное издание (которое можно получить на сайте http://www-

3.ibm.com/software/network/techexplorer/), не только воспроизводит документы, но и обладает некоторыми дополнительными функциональными возможностями, например: гипертекстовыми ссылками, просмотром изображений в форматах GIF и JPEG, пользовательскими меню и иерархической навигацией по документу. Профессиональное издание (которое нужно приобретать у фирмы IBM) в дополнение к функциональным возможностям свободно распространяемого вводного издания обеспечивает поддержку вывода на печать, поиска, встроенного видео, интерфейсов для программирования на Java/JavaScript и имеет дополнительные встроенные структуры, обеспечивающие взаимодействие techexplorera с другими приложениями. Существенным ограничением в использовании techexplorera является невозможность отображения символов кириллицы в тех случаях, даже если они находятся вне математических формул.

1.1.2 WebEQ

WebEQ фирмы Design Science (http://www.dessci.com/webmath) — это набор инструментов, включающий Java-апплет для отображения математических выражений. Апплет обеспечивает поддержку математической разметки Mathematical Markup Language (MathML) и набора команд, которому дано имя WebTeX. WebTeX является еще более узким набором команд LaTeX, нежели набор команд, распознаваемых плагином techexplorer. Пакет WebEQ включает в себя редактор, который позволяет создать математическое выражение, а потом сохранить его в виде исходного кода MathML, рисунка в формате JPEG или PNG, либо в виде исходного текста Java-апплета, который можно скопировать и вставить в HTML-страницу. В редакторе предусмотрена также возможность открытия файла, содержащего уже существующее выражение в разметке MathML, но он не позволяет сохранить результаты работы в WebTeX-формате.

Редактор WebEQ создан с использованием входящего в состав пакета WebEQ программного интерфейса для языка Java, который позволяет добавлять к собственным апплетам функции отображения математических выражений и создания рисунков, а также дает возможность управлять WebEQ с помощью JavaScript. Имеется возможность динамически создавать новые математические объекты, а также манипулировать внутренней структурой уже существующих объектов.

Для облегчения вставки математических объектов в Web-страницы имеется Мастер WebEQ. Эта прикладная программа воспринимает в качестве исходных данных файл, содержащий разметку на языке HTML и разметку WebTeX или MathML, и создает новый HTML-файл, содержащий рисунки или элементы апплета WebEQ для математических выражений.

1.2 Представление научных текстов в формате PDF

В настоящее время PDF де-факто стал стандартом при обмене «готовыми» научными публикациями. Редакции журналов присылают авторам статьи на корректуру именно в формате PDF.

На сайтах издательств опубликованные статьи выставляются в формате PDF. Ученые также предпочитают обмениваться уже опубликованными статьями в виде файлов PDF, хотя еще несколько лет назад в подобных случаях приходилось отправлять по электронной почте исходный текст (в разметке LaTeX) и рисунки (в виде отдельных файлов). Обмен печатными материалами в виде PDF гарантирует, что при воспроизведении материала на любом выходном устройстве (принтере) материал будет отображен совершенно одинаково.

Трудно понять, почему PDF занимает почетное, но только второе место после HTML, когда заходит речь о научных публикациях в интернете. Назовем несколько причин, которые, как нам кажется, предопределили сложившуюся ситуацию. Сразу скажем, что ни одна из них не является фатальной для перспектив распространения PDF. Но сумма оказалась внушительной.

- PDF опоздал с выходом в интернет. Только с 1996 года стало возможным показывать документов PDF в интернет-браузерах, когда место под солнцем уже было занято. К тому же PDF не имеет ощутимых преимуществ (если вообще имеет хоть какие-то преимущества) по сравнению с безраздельно царствовавшим в то время языком разметки HTML, если отвлечься от проблемы отображения математических формул. Да и проблемы-то как таковой нет. По тем временам вполне приемлемым казалось отображение формул в виде рисунков gif.
- Документы в формате PDF обычно имеют больший размер, нежели HTML. Это плата за точность отображения информации. На заре интернета проблема пропускной способности сетей стояла очень остро, а типичная научная статья в формате PDF имеет размер от 0.5 Mb и выше. Сейчас острота проблемы уменьшается, но пользователи интернета свой выбор уже сделали. Поезд ушёл.
- Нужно понимать психологию ученого. Даже человек без специального образования способен быстро научиться «править» документы HTML «вручную», используя простейший текстовый редактор и пару простеньких образцов. Документ PDF изнутри сродни исполняемой программе. Его невозможно редактировать без специальных инструментов. Фирма Adobe, разработавшая формат PDF, предоставила целым спектр таких инструментов, но, пытаясь сохранить свое монопольное положение, не позволила другим фирмам производить такие инструменты. Это находилось в разительном контрасте с положением дел на рынке редакторов HTML.
- Фирма Adobe явно опоздала с введением скриптов в документы PDF. Без них электронный документ мало чем отличается от глиняных табличек с шумерской клинописью. Без скриптов невозможно в полной мере обеспечить интерактивное взаимодействие пользователя с электронным документом (с браузером). Опоздав, фирма Adobe не смогла наверстать упу-

щенное, так как «JavaScript по Adobe» и «JavaScript по Microsoft» — это две большие разницы (см., например, [6]).

- Наконец представим, что ничего из сказанного выше нет и в помине. Представим, что пользователь заполнил в браузере какую-нибудь форму в PDF документе и что указанные им сведения переданы на сервер. И что дальше? А ничего. Если пользователь желает получить ответ, возможно, он его получит, но в формате HTML. Существует очень небольшой набор серверных программ, которые способны динамически генерировать документы PDF сообразно сведениям, полученным от пользователя. А между тем Web-серверы заняты почти исключительно этим делом. При этом обычно используются скрипты ASP и PHP, чтобы обработать на сервере запрос пользователя и отправить в ответ страничку HTML. Нам известны две мощные (и, соответственно, дорогие) системы автоматизации бизнеса компаний SAP и PeopleSoft, которые поддерживают электронные формы в формате PDF и способны динамически генерировать документы PDF на стороне сервера сообразно сведениям, полученным от пользователя, но они мало распространены.
- Другим важнейшим вопросом после интерактивности является вопрос о поиске информации в документах. Можно ли найти документ PDF, если он уже есть на сервере? Можно, и для этого нужно прибегнуть к помощи службы поиска. Тут-то и выясниться, что многие поисковые машины до сих пор не умеют индексировать документы PDF. Проще говоря, не знают, как извлечь полезную информацию из PDF. Одна из новых поисковых машин 2000 году с гордостью информировала интернет-общественность, что способна индексировать документы PDF. Конкуренты молчат до сих пор. Только в 2001 году фирма Adobe выпустила фильтр для индексации документов PDF с помощью службы индексирования Indexing Services, которая является неотъемлемой частью Windows 2000, а в виде отдельного продукта поставляется с 1997 года.

Как уже упоминалось выше, важнейшим вопросом является создание и/или преобразование документов в формат PDF. Редактора PDF как такового не существует. Фирма Adobe (и ряд независимых программистов) создали программы конвертации документов Word и LaTeX в формат PDF [5,6].

Преобразование документа LaTeX в формат PDF осуществляется в «одно касание». Приведем выдержку из книги И.Котельникова и П.Чеботаева «LaTeX», которая готовится к переизданию в издательстве «Бином». Авторы пишут, что с недавних пор традиционный сценарий компиляции исходного текста в документ DVI стал не нужен, так как теперь есть все необходимое для прямого преобразования размеченного текста в формат PDF.

Документ LaTeX нетрудно преобразовать в PDF формат. Проще всего это сделать, заменив

компилятор:

```
pdflatex first.tex
```

Программа `pdflatex` на выходе создаст файл `first.pdf`. Его можно просмотреть на экране или напечатать при помощи программы Acrobat Reader, бесплатно распространяемой фирмой Adobe, или при помощи другой широкой известной программы `Gsview`. Эти программы поставляются отдельно от системы LaTeX. При печати на бумагу документ, переданный в виде pdf-файла, имеет столь же высокое качество, как и dvi-файл, но его неимоверно проще пересылать по электронной почте или экспонировать в интернете, поскольку программа Acrobat Reader может встраиваться в наиболее распространенные интернет-браузеры.

Здесь уместно спросить, а стоило ли тогда вообще упоминать про `dvi` и программу `latex`, если `pdf` и `pdflatex` имеют столь неоспоримое преимущество. Стоило, но оснований для этого остается всё меньше и меньше. Последнее серьезное препятствие для повсеместного перехода от `latex` к `pdflatex` (для русскоязычных пользователей) исчезло в конце 2001 года, когда Владимир Волович создал PostScript-версию русских LH-шрифтов. Поясним, что так называемые `mf`-шрифты, технология которых была разработана Доном Кнутом специально для TeX, мало пригодны для внедрения в документ `pdf` — в Acrobat Reader они выглядят ужасно. Ещё полезно знать, что из `dvi`-файла также можно получить документ `pdf` и что в редких случаях такой многоступенчатый способ получения документа `pdf` более предпочтителен, нежели прямая компиляция файла `tex` в `pdf` посредством программы `pdflatex`.

Все сказанное выше приводит нас к мысли, что в настоящее время формат PDF является предпочтительным для обмена готовыми научными материалами в интернете, но он обладает рядом недостатков, затрудняющих интерактивное взаимодействие с этими материалами в режиме он-лайн.

1.3 Представление научных текстов в формате HTML

1.3.1 HTML и математика

С самого начала World Wide Web, который теперь просто называют вебом, зарекомендовал себя, как весьма эффективный способ сделать информацию доступной большому количеству людей. Однако, даже при том, что веб был изначально задуман и реализован учеными для ученых, возможности для включения математических выражений в HTML были крайне ограничены. В настоящее время, большая часть математической информации в вебе представлена в виде текстов с включением графических изображений математических формул (в виде рисунков GIF или JPEG) или в виде полноформатных документов PDF.

Спрос на эффективные средства обмена научной информацией остается по-прежнему вы-

соким. Все больше и больше исследователей, ученых, инженеров, учителей, студентов, работая в различных местах земного шара, полагаются на электронные средства коммуникации. В тоже время, методы распространения научной нотации, основанные на графическом представлении, преобладающие сейчас в интернете, примитивны и неадекватны. Качество документов оставляет желать лучшего, создание — сложно, а математическая информация, содержащаяся в картинках, не доступна для поиска, индексации и повторного использования в других приложениях.

Существует две основных проблемы с использованием HTML.

Проблема отображения. Рассмотрим уравнение . Размеры изображения подобраны так, чтобы соответствовать окружающему тексту размера 14pt в системе, где оно было создано. Естественно, что в других системах или при другом размере текста уравнение будет выглядеть не так, как задумывал автор. Кроме того, изображение с этим уравнением создавалось в предположении, что у документа будет белый фон. Таким образом, если читатель или браузер установит другой цвет страницы, то в результате вокруг текста возникнет белый "ореол".

Далее рассмотрим уравнение , которое служит примером выравнивания горизонтальной оси изображения по верху строчных букв в окружающем тексте. В этом уравнении присутствует знаменатель, и базисная линия располагается на расстоянии около одной трети

от низа изображения. Это уравнение, например, можно разместить так: , то есть так, чтобы горизонтальная ось изображения и базисная линия уравнения совпадали, но это приведет к проблемам с межстрочными интервалами. В результате этого текст станет тяжело читаемым. Кроме того, вертикальное выравнивание по центру отличается в различных браузерах, делая невозможным гарантировать правильное расположение формул для различных пользователей.

Уравнения, оформленные в виде изображений, обычно труднее воспринимать, чем окружающий текст. Более того, эти проблемы только увеличиваются, если документ напечатать. Разрешение изображений около 70 точек на дюйм, тогда как окружающий текст обычно имеет 300, 600 или более точек. Такая разница в качестве неприемлема для большинства людей.

Проблемы кодирования. Попытаемся найти в этом документе фрагмент, например, '=10' первого уравнения выше. Или попробуем скопировать и вставить уравнение в другое приложение; более того, попробуем скопировать и вставить только фрагмент уравнения. При использовании методов, основанных на графическом представлении, ни одна из этих задач не может быть решена. И хотя использование атрибута `alt` в документе может немного помочь, ясно, что интерактивные веб-документы должны предоставлять более продвинутый интерфейс между браузерами и математической информацией. Иными словами, отображение математических формул в виде рисунков противоречит современной концепции разделения содержания и представления со всеми

вытекающими отсюда последствиями.

Кроме всего сказанного, использование изображений для математических объектов требует высокой пропускной способности канала связи. Представление уравнения разметкой обычно (но не всегда) меньше по объему и лучше сжимаемо по сравнению с представлением в виде изображения. Существенно также, что при использовании разметки для записи уравнений большая часть работы по отображению перекладывается на клиентскую машину.

Проблемы с некачественным отображением математических выражений в документе HTML, вероятно, могут быть решены по мере совершенствования браузеров и стандартизации рендеринга, реализованного в различных типах браузеров. Однако улучшение рендеринга не решит проблему доступа к информации, стуктурирования и поиска информации, содержащейся в математических выражениях. Поэтому улучшение методов, основанных на графическом представлении, представляется малоперспективным. Для полной интеграции математического материала в веб-документ необходимо представление математической нотации с помощью языка разметки.

1.3.2 Математический язык разметки

При создании любого языка разметки необходимо тщательно учесть потребности его потенциальных пользователей. В случае с MathML область применения простирается от сферы образования до сферы исследований и даже бизнеса.

Сфера образования является большой и важной областью, в которой должна быть обеспечена возможность размещения в вебе научных материалов для обучение. В тоже время, преподаватели ограничены во времени и оборудовании, и обычно останавливаются перед сложностью создания технических веб-документов. Студенты и преподаватели должны иметь возможность быстро и легко создавать математические документы, используя интуитивно понятные, легкие для изучения и "дешевые" инструменты.

Электронные учебники — другой способ использования веба, который потенциально очень важен для образования. Известный специалист по управлению Peter Drucker прогнозирует окончание эпохи системы высшего образования, сконцентрированной вокруг больших университетских городков, и начало ее распространения через веб. Электронные учебники должны быть более интерактивными и должны связывать текст с научным программным обеспечением.

Для соответствия требованиям исследовательского сообщества, язык математической разметки должен упростить обслуживание и работу с большими объемами документов, где крайне важны автоматический поиск и индексация. Из-за большого количества уже существующих математических документов, особенно в формате TeX, также немаловажно иметь возможность конвер-

тации между существующими форматами и любым новым. И, наконец, для научных исследований жизненно необходима возможность хранить информацию в архивах.

Многие ученые и инженеры используют в своей работе технические документы для обмена и записи результатов экспериментов, а также для компьютерного моделирования и проверки проведенных вычислений. Для такого рода использования методы размещения математической информации в вебе должны предоставлять стандартный способ совместного ее использования так, чтобы можно было легко ее прочитать, обработать и создавать с использованием доступных и простых инструментов.

Другое общее требование заключается в возможности отображения математического материала в других информационных средах, таких, как речь или шрифт Брайля, которые крайне важны для людей с нарушением зрения.

Коммерческие издательства также заинтересованы в продвижении методов представления математической информации в вебе в различном виде от электронных версий обычных книг до интерактивных учебников и академических журналов. Издательствам требуется такой способ размещения математических документов в вебе, который бы имел возможность для высококачественного вывода, был бы применим для крупномасштабного коммерческого использования и, по возможности, совместим со старыми системами, в частности, на основе SGML.

1.3.3 Особенности математической нотации

Характерной чертой математической информации является использование сложной и высокоразвитой двумерной символьной системы обозначений. Однако, как писал J.R. Pierce в своей книге по теории коммуникации, математика и ее нотация не должны рассматриваться как одно и то же. [7]. Математические идеи существуют независимо от способа их представления. Тем не менее, взаимосвязь между значением и обозначением весьма тонка, и в возможности представлять и манипулировать идеями в символьной форме кроется значительная мощь математического аппарата, как инструмента описания и анализа. Основная трудность при внедрении математики в World Wide Web состоит в том, чтобы зафиксировать как представление, так и содержание (то есть значение) таким образом, чтобы в документах максимально использовать высокоразвитую систему математической нотации и потенциал взаимодействия в электронных средствах информации.

Математическая система обозначений постоянно развивается, так как люди постоянно усовершенствуют способы представления идей. Даже стандартная система обозначения арифметических действий прошла через удивительное многообразие стилей, включая множество ныне несуществующих, поддерживающих математические обозначения своего времени [8]. Современная математическая система обозначений является продуктом вековых усовершенствований, и принятые обозначения для высококачественной печати достаточно сложны. Например, переменные и

буквы, обозначающие числа, сейчас обычно печатаются специальным математическим курсивом чуть отличным от обычного текстового курсива. Пробелы, окружающие символы операций такие как +, -, × и /, немного отличаются от таковых в тексте, отражая соглашения о старшинстве оператора. Целые книги посвящены правилам набора математических текстов, от выравнивания верхних и нижних индексов до правил для выбора размеров круглых скобок и специальных нотационных соглашений для различных областей математики (например, [9-12], или в литературе по TeX [1-3]).

Принятая система обозначений в математике, и в печатном тексте вообще, направлена на то, чтобы зрительно выделить и сделать напечатанные выражения более легкими для чтения и понимания. Хотя это и кажется очевидным, но мы полагаемся на сотни соглашений, таких как параграфы, заглавные буквы, семейства шрифтов, и даже механизм десятичной нумерации разделов, подобный тому, что мы используем в этом документе (заслуга G. Peano, который, вероятно, более известен своими аксиомами для натуральных чисел). Нотационные соглашения, наверное, даже более важны для электронных документов, поскольку для них существуют известные трудности при чтении с экрана.

Внедрение математики в веб — это не просто поиск способов отображения математической информации в окне браузера. Веб представляет фундаментально новый подход к хранению знаний, в котором *взаимосвязь* играет центральную роль. Становится все более и более важно найти способы взаимосвязи математических документов, которые облегчат автоматическую обработку, индексацию и повторное использование в других математических приложениях и контекстах. Подобные усовершенствования в технологии коммуникации позволяют расширить наши возможности в представлении и кодировании математического материала. Мы надеемся, что MathML является важным шагом в этом направлении.

1.3.4 История MathML

Задача представления математической информации для компьютерной обработки и электронных средств коммуникации возникла задолго до появления интернета. Раньше общей практикой для ученых была запись статей в неком виде, основанном на ASCII-символах и дальнейшая пересылка их друг другу по электронной почте. Несколько языков математической разметки, в частности TeX [1], уже широко использовались в 1992 году, еще до того, как интернет занял столь значимое положение [13].

Консорциум World Wide Web (W3C) понимал, что отсутствие основ для научной коммуникации является серьезной проблемой. В 1994 году Dave Raggett внес предложение о включении HTML Math в прототип HTML 3.0. На конференции в Дармшадте в апреле 1995 года был проведен круглый стол по математической разметке. В ноябре того же года представители Wolfram Research

выдвинули предложение команде W3C о реализации поддержки математики в рамках HTML. Важную роль в объединении многих заинтересованных сторон сыграла проведенная в мае 1996 года встреча Digital Library Initiative в Champaign-Urbana. Результатом этой встречи стало формирование редакционного наблюдательного совета по HTML Math. Впоследствии эта рабочая группа разрослась, и в марте 1997 года была формально повторно сформирована как первая W3C Math Working Group. Вторая W3C Math Working Group была сформирована в июле 1998 года.

Проект MathML отражает интересы и мнения различных групп специалистов. Многое в развитии MathML заслуживает специального упоминания. Например, это касается вопроса общедоступности, где были особенно ощутимые затруднения. В этом направлении большую работу проделали T. V. Raman. Neil Soiffer и Bruce Smith из Wolfram Research. Они поделились своим опытом в решении проблем представления математического материала, накопленным при работе над проектом Mathematica 3.0. Их идеи оказали важное влияние на структуру элементов представления. Paul Topping из Design Science также внес свой вклад в математическое форматирование и редактирование. Много пользы извлек проект MathML из партнерства с рядом членов рабочих групп, связанных с другими работами по кодированию математической информации в SGML и в сообществах компьютерной алгебры. В их числе Stephen Buswell из Stilo Technologies, Nico Poppelier из Elsevier Science Stéphane Dalmas из INRIA (Sophia Antipolis), Stan Devitt из Waterloo Maple, Angel Diaz и Robert S. Sutor из IBM, и Stephen M. Watt из University of Western Ontario. Также, на развитие MathML повлиял проект OpenMath, работа рабочей группы ISO 12083 и работа Stilo Technologies над фрагментом DTD для "семантической" математики. Американское математическое общество играло ключевую роль в развитии MathML. Помимо прочего, председателями обеих W3C Math Working Group стали представители этой организации. С мая 1996 по март 1997 года группу вел Ron Whitney. Patrick Ion был сопредседателем группы с марта 1997 по июнь 1998 вместе с Robert Miner из The Geometry Center, а с июля 1998 вместе с Angel Diaz из IBM.

1.3.5 Проект MathML

Для соответствия различным требованиям научного сообщества MathML разрабатывался с учетом следующих условий.

MathML должен:

Представлять математический материал так, чтобы он подходил для обучения и научной коммуникации любого типа.

Представлять как математическую нотацию, так и математическое содержание.

Обеспечивать возможность преобразования между ним и другими математическим форматами, как презентационными, так и семантическими. Форматы вывода должны включать:

1. графическое отображение

2. синтезаторы речи
3. форматы систем компьютерной алгебры
4. форматы других языков, таких как TeX
5. отображение в виде простого текста, например, эмуляторы VT100
6. печатные устройства, включая работающие со шрифтом Брайля

Понятно, что такие преобразования могут привести к некоторой потере информации.

Иметь возможность включения информации, необходимой для определенных средств отображения и других приложений.

Поддерживать корректный просмотр длинных выражений.

Обеспечивать расширяемость.

Поддерживать шаблоны и другие средства редактирования математической информации.

Быть понятным человеку и простым для программной обработки.

В независимости от того насколько удачен MathML как язык разметки, он будет полезен только в случае грамотного его применения. W3C Math Working Group определила краткий список дополнительных целей его реализации. В нем описана минимальная функциональность, которую должны обеспечивать программы отображения и обработки MathML.

Выражения MathML внутри страниц HTML (XHTML) должны корректно отображаться в наиболее распространенных браузерах в соответствии с установками читателя и автора, при этом должно обеспечиваться качество, максимально достижимое на данной платформе.

Документы HTML (XHTML), содержащие выражения MathML, должны корректно выводиться на печать с высоким разрешением.

Выражения MathML, включенные в веб-страницы, должны реагировать на действия пользователя, такие как работа с мышью, и осуществлять взаимодействие с другими приложениями через браузер.

Редакторы и конвертеры математических выражений должны разрабатываться с учетом возможности создавать веб-страницы, содержащих выражения MathML.

В ближайшее время для решения проблемы отображения планируется использовать встраиваемые элементы, такие как Java-апплеты, плагины и элементы управления ActiveX. Однако, объем, в котором будет это реализовано, зависит от сотрудничества и поддержки производителей браузеров и другого программного обеспечения. W3C Math Working Group продолжает работу с рабочими группами по Document Object Model (DOM) и Extensible Style Language (XSL), чтобы гарантировать, что нужды научного сообщества будут удовлетворены в будущем. Ясно, что MathML 2.0 является значительным шагом вперед по сравнению с MathML 1.0 Recommendation (Апрель 1998).

Из приведенного анализа ясно, что целесообразно в настоящеее время ориентироваться на

разметку MathML, внедренную в файлы HTML (XHTML), тем более, что в последнее время появились как средства конвертации математических выражений из форматов LaTeX и Word в разметку MathML, так и плагины и апплеты для просмотра получаемых выражений с помощью Microsoft Internet Explorer и других браузеров.

1.4 Интерактивное представление результатов расчетов (моделирования) в виде графиков и мультфильмов

Учебные программы, моделирующие физические явления, которые позволяют представить результаты расчетов в виде графических и анимационных изображений, являются действенным инструментом обучения в арсенале преподавателей естественно-научных дисциплин. Не подменяя живое общение преподавателя и студента, они дополняют традиционное изложение предмета в виде набора простых моделей, которые можно объяснить “на пальцах”, средствами визуализации теоретических построений. Особую ценность представляют программы двойного назначения, которые можно использовать как для обучения студентов на практических занятиях и лекционных демонстраций, так и для самостоятельных занятий обучаемых через интернет при предварительном ознакомлении с материалом или для закрепления полученных знаний. Большое число таких программ по разным разделам физики представлено в настоящее время в сети (см., например, http://www.physicscentral.com/resources/interactive.html, http://physics.nad.ru/Physics/, http://users.erols.com/renau/impedance.html, http://www.physics.ucc.ie/~oll/hysteresis/node1.htm). При этом используются самые разные технологии: от приготовленного заранее видео-фрагмента в одном из стандартных форматов (например, MPEG) с загрузкой его через интернет на локальный компьютер и последующего просмотра с помощью стандартных средств (например, Windows Media Player), до java-апплетов, работой которых можно управлять с локального компьютера после его загрузки из сети. Возможно также размещение в сети на соответствующей странице файла (апплета), подготовленного в таких системах как MathCad (http://www.mathsoft.com/), Mathematica (http://www.wolfram.com/), MatLab (www.matlab.com), Macromedia Flash MX 6 (http://www.macromedia.com/software/studio/productinfo/product_overview) и др., после загрузки которых на локальный компьютер можно запустить соответствующий файл на исполнение. При этом, естественно, предполагается наличие на локальном компьютере соответствующей программы, которая может выполнить требуемые расчеты и с которой можно работать в дальнейшем в автономном режиме. Большое количество таких «электронных книг», как платных так и бесплатных, можно найти на сайтах соответствующих фирм – разработчиков этого математического обеспечения. Наибольшего внимания, с нашей точки зрения, здесь заслуживает технология фирмы Macromedia, которая получила наибольшее распространение в вебе; последние версии Macromedia Flash позволяют создавать интерактивные учебные демонстрации, как показывают примеры, раз-

мещенные в вебе по адресу http://www.physicscentral.com/resources/interactive.html.

Другим заслуживающим, с нашей точки зрения, вариантом для представления в интернете учебных материалов типа «моделирующая программа» является программа, расположенная на сервере, выполняющая все расчеты на стороне сервера и передающая в ответ на запрос по протоколу HTTP результаты расчета по этому же протоколу. При этом запрос на расчет не просто инициирует проведение соответствующего расчета, но позволяет задавать требуемый набор параметров и условия задачи, которые будут использоваться серверной программой. К таким программам относится MatLabWEB-сервер, который мы и использовали для разработки учебных и демонстрационных задач по курсу электродинамики [4].

2 Конвертация научных материалов в HTML

2.1 Пакет LaTeX2HTM

LaTeX, в основе которого лежит система набора текстов TeX, обладает более широкими возможностями, чем большинство других программных средств обработки текстов. В его основном описании [2] LaTeX характеризуется как «Система подготовки документов». Так и LaTeX2HTML —это нечто гораздо большее, чем LaTeXовская опция *Save As HTML (Сохранить в формате HTML).* Лучше всего к нему подходит определение «Система подготовки Web-документов».

LaTeX2HTML — это открытое программное средство, распространяемое по лицензии GNU. В настоящее время используется версия 99.1 или старше. Она работает в среде операционных систем Linux, OS/2, Windows NT, Windows 95 и DOS, а также в большинстве вариантов систем UNIX, для которой она была первоначально разработана. Версия для платформы Macintosh находится в состоянии разработки.

По умолчанию создаваемые HTML-страницы соответствуют спецификации HTML 3.2. Поэтому такие страницы можно просматривать в навигаторах, поддерживающих эту или более новую спецификацию HTML 4.0. При необходимости вместо этого можно выбрать создание страниц, которые соответствуют HTML 2.0. При таком ограничении для более сложных LaTeXовских окружений, таких как таблицы или выравнивания в математических формулах, используется графика. И наоборот, можно выбрать создание дополнительных атрибутов CLASS и ID, соответствующих спецификации HTML 4.0. Они позволяют ассоциировать информацию стиля с конкретными процедурами LaTeX или даже отдельными экземплярами процедур, абзацами и фрагментами текста. Более того, когда в одном документе используется несколько языков, то путем задания атрибутов LANG в версии HTML 4.0 обеспечивается поддержка многих языков и диалектов.

Транслятор LaTeX2HTML написан на языке Perl [14], так что в принципе LaTeX2HTML работает в любой системе, в которой установлен язык Perl. Но для того чтобы LaTeX2HTML мог начать трансляцию, необходимы еще некоторые программные средства. Как и LaTeX, язык Perl и другое необходимое прикладное программное обеспечение являются общедоступными и их легко получить в интернете.

2.1.1 Несколько исторических замечаний

Первоначально LaTeX2HTML был разработан Никосом Дракосом. В 1995 г. Дракос представил свою программу интернет-сообществу с помощью списка рассылки LaTeX2HTML [mailto:latex2html@tug.org]. К концу 1996 г. местонахождением репозитария для использования разработчиками стал Дармштадтский университет Германии, а впоследствии — Бейрутский университет, где его теперь поддерживает в рабочем состоянии Марек Рушаль. С тех пор многие внесли свой вклад в программу, обеспечивая оптимизацию для различных платформ, работая над мобильностью и совершенствуя процедуры инсталляции. Многие из этих разработок были непосредственной реакцией на полученные из списка рассылки запросы о поддержке отдельных LaTeXовских команд и пакетов.

Документация, поддержка LaTeXовских пакетов, стратегии создания изображений и расширения для самых последних спецификаций HTML, а также общая координация работ — все это, главным образом, заслуга Росса Мура.

2.1.2 Принципы создания Web-документов

Первые версии LaTeX2HTML были разработаны Никосом Дракосом для того, чтобы использовать только зарождавшийся тогда веб для компьютеризованного обучения и образования. Всегда требовалось представить наиболее важную информацию. Дракос также понял, что необходимо обеспечить легкое перемещение по представленной информации согласно ее логической структуре. Он перечислил несколько принципов, которые считал необходимыми для любого серьезного программного обеспечения, предназначенного для создания Web-документов (http://www.cbl.leeds.ac.uk/nikos/doc/www94/www94.html):

1. Автоматическое создание гипертекстовых сетей на основе структуры.
2. Гибкость в определении размера узлов.
3. Автоматическое создание средств навигации.
4. Включение сложно форматируемой информации, например рисунков, таблиц, математических уравнений, диаграмм, химических формул и экзотических языков.

Основанные на этих принципах ранние версии LaTeX2HTML были достаточно эффективны, даже при очень ограниченных возможностях первых версий HTML. При введенных последующими

спецификациями HTML расширениях эти же принципы были сохранены и при разработке транслятора LaTeX2HTML, более того, они были в полной мере использованы и расширены. Теперь все те процедуры, которые обычно использовались при подготовке бумажных документов с помощью LaTeXa, автоматически конвертируются в структуры и разметку HTML, обеспечивая практически наивысшее качество результатов, которое может быть достигнуто при конкретной версии HTML.

Главной целью программы конвертации, такой как LaTeX2HTML, является, конечно, принцип 1. Принцип 2 реализуется с помощью переменных конфигурации и опций командной строки, которые подробно описаны в Руководстве пользователя LaTeX2HTML (см., например, [5]).

На наш взгляд, основным недостатком LaTeX2HTML на сегодняшний день является представление математических выражений в виде графических объектов (рисунков gif). При этом не могут быть достигнуты многие важные цели, сформулированные при разработке проекта MathML. Тем не менее, реализуемое в специальном режиме разбиение математических выражений на отдельные элементы представляет собой определенный шаг вперед к созданию полноценной разметки языка MathML. Уже выполнена определенная работа над LaTeX2HTML с целью создания мастера WebEQ, генерирующего Java-апплеты и файлы с разметкой MathML. Для этих целей в HTML 3.2 создаются элементы <APPLET>, а в HTML 4.0 — элементы <OBJECT>. Однако в настоящее время эта процедура работает только с некоторым подмножеством LaTeXовской математики. Предполагалось, что работа в этом направлении будет развиваться, с тем чтобы охватить все стороны LaTeXовской математики, на которые распространяется MathML. Однако в настоящее время, как нам представляется проект LaTeX2HTML заморожен. Последняя информация о системе LaTeX2HTML, которую удается найти в интернете, датирована примерно 1999 годом.

2.2 Пакет TeX4HT

TeX4HT — это система, которая предоставляет способ создания настраиваемых гипертекстовых версий LaTeXовских документов (http://www.cis.ohio-state.edu/ ~gurari/TeX4HT/); она является расширением TEXa вообще и LaTeXa — в частности.

Система состоит из двух частей: набора стилевых файлов, которые дополняют существующие LaTeXовские файлы возможностями HTML, и постпроцессора для выделения HTML-файлов из результатов работы TeXa. Основную задачу преобразования исходных текстов LaTeX в документ иного формата, нежели в DVI -код TeX4HT оставляет самому TeXy. Следовательно, TeX4HT имеет доступ ко всем возможностям TeXa в области работы со шрифтами, макро, переменными, кодами категорий и другим важным свойствам, которые весьма часто остаются спрятанными от пользователя, но существенны для стилевых файлов. В большинстве приложений оказываются необходимыми только несколько основных функций этой системы. При создании

HTML-документов из LaTeXовских источников пользователь вряд ли заметит вмешательство системы.

Несмотря на то, что TeX4HT — это гибкая авторская система с богатыми возможностями, для большинства применений его функциональность остается незаметной. Как правило, компиляция и просмотр документов, использующих TeX4HT, выполняются так же просто, как и без него.

Процесс трансляции исходного документа в гипертекст состоит из трех этапов: компиляции исходного текста TeXовской программой (мы использовали для этой цели MiKTeX) в DVI-код, обработки DVI-кода программой `TeX4HT` и выполнение заключительных процедур, необходимых для завершения трансляции.

2.2.1 От LaTeXa к DVI

Для разрешения перекрестных ссылок в LaTeXe требуется дважды выполнить компиляцию исходного файла с помощью TeXa, а чтобы вместо них при использовании TeX4HT получить все гипертекстовые ссылки, может понадобиться и третья компиляция. Иногда, в случае табличных процедур с большим количеством объединяемых ячеек, может понадобиться выполнить компиляцию большее число раз, чтобы система смогла определить, как должны выглядеть эти ячейки.

Когда LaTeX загружает пакет TeX4HT, он загружает файл TeX4HT.sty и анализирует там всего лишь несколько строк. Затем он формулирует запрос на повторную загрузку этого файла впоследствии, при которой будет просмотрена оставшаяся часть файла. Второй раз загрузка выполняется при обнаружении кода `\begin{document}`, и на этот раз учитываются также требования обусловленные опциями пакета.

Поскольку TeX.4ht «выходит на сцену» только после команды `\begin{document}`, некоторые сделанные ранее пользовательские определения могут оказаться для него недоступными, если только они не были повторно определены в файле конфигурации. Например, могут возникнут трудности при генерации элементов HTML для показателя степени (верхнего индекса) в макро `\newcommand{\x}{a^{b}}`, который был определен до начал окружения `document`.

2.2.2 От DVI к HTML

DVI — это язык описания страницы, содержащий инструкции для определения того, что именно должно появляться в каждом месте, предназначенном для печати документа (носителя информации). И наоборот, HTML — это структурно-ориентированный язык, в котором очень мало внимания уделяется вопросам размещения информации.

Следовательно, преобразование из DVI в HTML — это во многих отношениях обратный процесс, при котором нужно восстановить информацию, возможно, потерянную при преобразовании из LaTeXa в DVI. Этот обратный процесс может оказаться, совершенно неудачным, если DVI-код был получен из исходного документа, в котором нарушено обычное расположение элементов; примером этого может быть `\hspace{-0.6em}`.

В процессе трансляции из DVI в HTML обращения к шрифтам обрабатываются с помощью виртуальных гипертекстовых шрифтов. Если они отсутствуют или непригодны, то пользователь может без особых усилий сформировать новые шрифты.

2.2.3 От DVI к GIF

TeX4HT не содержит средств для преобразования DVI-кода в растровое изображение. В этом он полагается на внешние средства, доступные при выполнении задания. В имеющемся варианте TeX4HT используется двухступенчатое преобразование рисунка в формате DVI в GIF-файл, содержащий растровое изображение. На первом шаге для преобразования рисунка в промежуточный PostScript-файл используется драйвер `dvips`. На втором шаге для завершения работы вызывается программа `convert` (являющаяся частью пакета ImageMagick). При определенной настройке пакета можно повторно графические файлы не конвертировать, если в этом нет необходимости. Поскольку конвертация `ps -> gif` выполняется внешней процедурой из пакета ImageMagick, то можно ее заменить на более старую, но хорошо известную пользователям LaTeXa библиотеку процедуру `netpbm`.

Размеры рисунков зависят от размера исходных документов и, в случае рисованных символов, от размеров используемых шрифтов. Однако размеры и качество рисунков могут также зависеть от выбранных значений установочных параметров для используемых внешних утилит и, конечно, от экрана монитора, используемого для просмотра HTML-страниц.

2.2.4 От DVI к XML

Наиболее значимым достоинством системы TeX4HT является практически неограниченные возможности его конфигурирования. TeX4HT для чтения исходного текста документы использует саму систему LaTeX, поэтому не имеет ограничений на набор команд и процедур — для любой команды и любой процедуры LaTeX можно определить эквивалент на языке разметки, который должен использовать выходной документ (обычно HTML). Вместо HTML выходной документ может быть записан в разметке XML или иной другой — это зависит от того, как определены эквиваленты команд и процедур LaTeXa..

Автор системы TeX4HT Эйтан Гурари разработал несколько вариантов конфигурационных и стилевых файлов, которые позволяют получать на выходе системы файлы в разметке HTML, XHTML, XML в различном сочетании представления математических формул: от рисунков GIF до разметки. Перечисление всех вариантов здесь затруднительно, поскольку, например, при выборе разметки возможны дальнейшие варианты: можно либо ориентироваться на представление разметки MathML в окне браузера с помощью плагина MathPlayer, либо с помощью XSL стилей, разработанных Давидом Карлайлом (David Carlisle), ориентированных на более широкий спектр браузеров. В частности, стили Карлайла позволяют просматривать документы с разметкой MathML в окне браузера Netscape 7, тогда как первый вариант позволяет просматривать результат только с помощью браузера Microsoft Internet Explorer (см., например, http://www.oasis-

open.org/cover/gurari-ml9808.html). Неограниченные возможности пакета TeX4HT в его конфигурировании и предопределили наше решение – использовать его в качестве базового для построения системы преобразования научных и учебных документов из формата LaTeX в формат HTML+MathML. Мы остановили свой выбор на отображении разметки MathML в браузера Internet Explorer с помощью плагина MathPlayer, учитывая что Internet Explorer в настоящее время занимает монопольное положение на рынке браузеров. При появлении других, достаточно широко используемых браузеров, способных отображать разметку MathML, адаптация выходного файла к этим браузерам не будет представлять никаких затруднений. Мы решили не преобразовывать документ LaTeX в документ с разметкой XML, что более соответствовало бы идее разделения содержания и представления, по причинам, которые объясним в следующем разделе.

2.2.5 XML? Этого мало

Если принять к исполнению концепцию отделения содержания от представления и довести ее до логического завершения, следовало бы один раз преобразовать исходный текст, размеченный командами LaTeX, в разметку XML и далее работать только с документом XML. При этом преобразование документа XML к виду, удобному для представления в печатном или электронном виде, можно производить при помощи наложения на документ XML различных стилей XSL, спроектированных индивидуально для каждого типа представления. Основное удобство такого подхода состоит в том, что изменения, вносимые в исходный XML документ, отображаются в документах любого типа представления без их дополнительного редактирования. Такой подход также позволяет легко преодолеть проблемы, связанные с различием возможностей браузеров разных типов, поскольку XSL стиль можно разработать отдельно для каждого типа браузеров.

Способы разработки стилей XSL для трансформации документа XML в документ HTML хорошо отработаны (см., например, http://msdn.microsoft.com/library/default.asp ?url=/nhp/default.asp?contentid=28000438). Имеются также наработки по преобразованию документов XML с математическими формулами в разметке MathML в разметку LaTeX, что позволит в перспективе использовать компилятор LaTeX для подготовки печатных документов из XML. В этом направлении следует обратить внимание на работы Василия Ярошевича (см. http://www.raleigh.ru/MathML/mmltex/).

Однако использование документов XML+MathML в качестве хранилища исходных текстов имеет ряд недостатков.

Во-первых, многие пользователи LaTeX освоили язык разметки LaTeX настолько хорошо, что предпочитают набирать математические формулы непосредственно в командах LaTeXa, не прибегая к помощи визуальных редакторов. Для них необходимость переучивания на MathML стало бы серьезным препятствием.

Вторая проблема представляется более серьезной. LaTeX имеет гибкую систему перекрестного цитирования, которая позволяет делать ссылки на уравнения, задачи, теоремы, параграфы, главы и вообще любые нумерованные объекты не только назад по тексту, но и вперед по тексту. Это достигается использованием меток нумерованных объектов и многократной компиляцией документа. Каждому нумерованному объекту, на который нужно дать ссылку из другой части данного документа или даже из другого документа, можно присвоить метку. Эту метку можно использовать для ссылки на объект. При первой компиляции LaTeX последовательно нумерует все объекты в порядке их появления в документе и составляет таблицу соответствия меток и номеров объектов. При второй компиляции компилятор считывает эту таблицу и вставляет номера объектов в ссылки, используя соответствие меток и номеров. Такая организация перекрестного цитирования требует двукратного компилирования документа, а иногда и трехкратного (если, например, номер уравнения составлен из номера другого уравнения, к которому присоединен штрих). Однако она позволяет поддерживать правильное соответствие номеров объектов и ссылок на них в ходе всей работы над документом, поскольку при добавлении, удалении или перестановке объектов, они автоматически перенумеровываются, а ссылки автоматически обновляются. Это существенно упрощает работу, особенно работу над большим документом.

Поскольку для сохранения подобной системы организации перекрестных ссылок требуется многократная компиляция документа, простое "обувание" документа XML в стиль XSL недостаточно, так как это принципиально одноразовое действие. Таким образом, для сохранения системы перекрестного цитирования, имеющейся в системе LaTeX, необходимо написать компилятор документов XML, который бы составлял таблицу соответствия меток нумерованных объектов и их номеров, и только затем проводил "обувание", либо … использовать для этой цели компилятор LaTeX. Мы выбрали именно этот второй путь, не исключая, однако, что в будущем мы примем решение вернуться к первому варианту и написать компилятор документов XML.

2.2.6 Составные части пакета TeX4HT и вспомогательные программы (MikTeX, ImageMagic, Aladdin GhostScript)

Как было описано выше, пакет TeX4HT использует для своей работы дополнительные процедуры и утилиты и прежде всего сам LaTeX (точнее, компилятор MiKTeX для операционной системы Windows). Для работы с пакетом TeX4HT нами использовались:

1. Пакет MiKTeX версии 2.0 и старше; пакет MiKTeX можно взять на собственном сайте, посвященном этому пакету www.miktex.org или любом сайте-зеркале CTAN, (например, в НГУ это ftp://ftp.nsu.ru/mirrors/ftp.dante.de/pub/tex/systems/win32/miktex или www.tutornet.ru/TEX/soft/miktex);

2. пакет ImageMagick для преобразования изображений; пакет ImageMagick можно взять с сайта, посвященного этому пакету www.imagemagick.org или в НГУ это проще сделать по адресу http://www.tutornet.ru/TeX/WebTools/TeX4HT/ImageMagick-win2k.zip
3. пакет GhostScript; пакет GhostScript вы можете взять либо в НГУ по адресу http://www.tutornet.ru/TeX/Soft/Aladdin, либо на сайте, посвященном этому пакету http://www.cs.wisc.edu/~ghost/index.html — кликните на ссылке "Obtaining AFPL GhostScript 7.00", затем найдите раздел для MS-Windows, и загрузите саморазворачивающийся архив, в настоящее время это `gs700w32.exe`
4. пакет TeX4HT; пакет TeX4HT и его последние обновления лучше всего взять с сайта автора Эйтана Гурари http://www.cis.ohio-state.edu/~gurari/TeX4HT/mn.html (это файлы `wfiles.zip` и `newt4ht.zip`) или в НГУ по адресу http://www.tutornet.ru/TeX/WebTools/TeX4HT. Кроме того, может понадобиться файл `TeX4HT-miktex.zip`, созданный П. Витоном, который можно взять либо на его собственной странице http://facweb.arch.ohio-state.edu/pviton/support/TeX4HT.html, либо в НГУ на сервере www.tutornet.ru.

2.2.7 Установка рабочей среды для преобразования и ее обновление

Установка MiKTeX

Как упоминалось выше, прежде всего необходимо установить MiKTeX. Инструкцию по установке можно получить по сети по адресу http://tutornet.ru/tex/Soft/miktex/HowToInstall/default.htm или непосредственно с сайта, посвященного пакету MiKTeX (www.miktex.org). Далее предполагается, что уже установлен MiKTeX 2.1 (в директорию, которая рекомендуется по умолчанию, `c:\texmf`). Кроме того предполагается, что при инсталляции MiKTeXa добавлен путь в директорию с каталогом `...\bin`. Установка MiKTeX 2.2 ничего не меняет в дальнейшем изложении, что же касается установки TeX4HT из меню установки MiKTeX, которая появилась в последнее время в версии 2.2, то наш опыт говорит, что предлагаемая процедура установки пока еще не совершенна и лучше ее не использовать.

Если используется MiKTeX 2.0, то при его установке была какая-то аномалия, потому что этот пакет устанавливался по умолчанию в `c:\Program Files\miktex`, а имя каталога, которое содержит пробел, усложняло установку TeX4HT. Версия 2.1 возвращается к стандарту `c:\texmf`, как было в ранних версиях; но при желании использовать каталог с именем, содержащим пробел, необходимо при установке TeX4HT использовать «короткую форму» имени этих каталогов.

Установка GhostScript

За исключением директории по умолчанию, описанное далее относится к GhostScript ver.6.0+. Для инсталляции необходимо запустить исполняемую программу для дистрибутива

gs700w32.exe, который содержит встроенную процедуру установки. Будет задан вопрос куда устанавливать и рекомендуется принять значения, которые система рекомендует по умолчанию, тогда исполняемые модули будут помещены в директорию c:\gs\gs7.00\bin. Пакет со шрифтами GhostScript'а удалять не нужно — они могут понадобиться. *Не рекомендуется устанавливать GhostScript в директорию, путь к которой содержит пробелы* — TeX4HT не сможет обработать такой путь.

Установка ImageMagick

Написанное далее относится только к версии 5.20+. Необходимо разархивировать (unzip) дистрибутив на диск c:\. В настоящий момент она будет помещена в папку c:\ImageMagick. *Не рекомендуется размещать ImageMagick в директорию, путь к которой содержит пробел — TeX4HT не сможет обрабатывать ее.*

Для того, чтобы ImageMagick мог найти свои delegates-файлы, необходимо установить переменную окружения magick_delegate_path чтобы она указывала на директорию, содержащую delegates.mgk, в текущем дистрибутиве — это c:\ImageMagick. Необходимо также установить переменную magick_module_path таким образом, чтобы она указывала на положение файла modules.mgk, который, как правило, располагается в том же месте, что и delegates.mgk.

Установка TeX4HT

Для установки TeX4HT необходимо разархивировать wfiles.zip в c:\TeX4HT и соответствующие поддиректории, которые сохранены в архиве. В этом случае файлы будут размещены в директории c:\TeX4HT и поддиректориях. При обновлении TeX4HT файлы с обновлениями необходимо разархивировать в c:\TeX4HT, и новые файлы заменят старые.

Начиная с мая 2001, файлы обновления распространяются с атрибутом «только для чтения» (read-only); и это может вызвать проблемы. Некоторые программы-архиваторы, WinZip например, игнорируют это, так что можно не обращать на это внимание. Но в любом случае это желательно проверить следующим образом:

1. Запустить DOS-сессию и перейти в основную директорию TeX4HT. Набрать и выполнить команду attrib TeX4HT.4ht. Если в атрибутах будет буква R (read-only), тогда выполнить команду attrib -r *.* /s. Это снимет атрибут read-only со всех файлов в директории и со всех файлов в поддиректориях (где хранятся гипертекстовые шрифты). Можно снять этот атрибут и работая в каком-нибудь файл-менеджере, например, в FAR.

Стилевые файлы TeX4HT (*.4ht) перенести в LaTeX-дерево MiKTeX'а, для чего:

2. Создать новую директорию в дереве LaTeX'a; если MiKTeX установлен в его директорию по умолчанию, то это будет директория:

- для MiKTeX 2.1, 1.20 или более ранней версии или для любой инсталляции MiKTeX'a, чьи директории верхнего уровня *не* содержат пробела: `c:\texmf\tex\latex\TeX4HT`
- для MiKTeX 2.0 или любой инсталляции MiKTeX'a, чьи директории верхнего уровня *содержат* пробел: `c:\Program_Files\MiKTeX\tex\latex\TeX4HT`

3. Скопировать `*.sty` и `*.4ht` из `c:\TeX4HT` в эту директорию. Удалите их из директории `c:\TeX4HT`.
4. Необходимо обновить базу данных MiKTeX'a, иначе TeX не найдет эти файлы. Для этого необходимо использовать `Start Menu -> Programs -> MikTeX 2 -> MiKTeX Options`. На закладке `General` нажмите кнопку `Refresh Now`. Другой путь — запустить из командной строки DOS (в окне DOS-сессии) команду `initexmf -u`.

На следующем шаге необходимо сконфигурировать `TeX4HT.env`. Если установлен MiKTeX 2.0 в свою директорию по умолчанию, или любая другая версия MiKTeX'a установлена в директорию, имя которой содержит пробел, то необходимо перейти к разделу «Специальные инструкции для MiKTeX 2.0»

В противном случае необходимо открыть файл `TeX4HT.env` в любом текстовом редакторе и:

- В строке `tc:\path\tfm!` заменить `path` на путь к директории со шрифтами MiKTeX'a. По умолчанию это `miktex\fonts`, так что вся строка будет иметь вид `tc:\miktex\fonts\tfm!`
- Добавьте еще одну `t` строку, указывающую на директорию, в которой располагаются ваши «локальные» метрические файлы: по умолчанию эта строка будет выглядеть так `tc:\Localtexmf\fonts\tfm!` (помните, первой должна быть буква `t`, и последний символ — восклицательный знак).

  Заметим, что директория `c:\Localtexmf\fonts` может вообще отсутствовать, если вам никогда не надо было создавать дополнительные метрические файлы кроме тех, которые инсталлируются вместе с MiKTeX. (Вы можете искусственно ее создать пропустив через LaTeX короткий документ со шрифтом 12 pt). Как бы там ни было, если директория `c:\Localtexmf\` существует, то необходимо добавить вышеупомянутую `t` строку, даже если на нее нет пока явной ссылки.
- Если MiKTeX установлен так, что он включает другие директории LaTeX, содержащие `tfm`-файлы, также необходимо добавить соответствующие `t` строки.
- В строке, начинающейся с `Ggswin32c`, необходимо заменить `gswin32c` на полный путь к `gswin32c.exe`: в установке GhostScript 7.00 по умолчанию это будет `Gc:\gs\gs7.00\bin\gswin32c`. Внимание! Не удалите начальное `G` в этой строке —

это необходимо для TeX4HT.

- В строке, начинающейся с `Gconvert`, необходимо заменить `convert` на полный путь к утилите `convert` пакета ImageMagick: при установке по умолчанию это будет `Gc:\ImageMagick\convert`. Внимание! Не удалите начальное `G` в этой строке — это необходимо для TeX4HT.

После редактирования необходимо сохранить `TeX4HT.env`.

Далее, необходимо добавить директорию TeX4HT (т.е. `c:\TeX4HT`) в ваш путь. Это можно сделать с помощью Windows NT Control Panel или путем редактирования файла `autoexec.bat` при работе под Windows 95/98 (помните, если вы отредактировали `autoexec.bat`, то для того, чтобы новые установки вступили в силу необходимо перегрузить компьютер). Наконец, необходимо скопировать файлы `*.tab` (`ht`, `htlatex`, `httex`, и `httexi`) в файлы с именем `*.bat` и перенесите их в какую-нибудь директорию, находящуюся в вашем пути. Поскольку директория MikTeX'a ...`\bin` находится в пути, то можно их всех поместить туда.

Специальные инструкции для MiKTeX 2.0

Далее описана инсталляция TeX4HT в том случае, когда директория, в которую установлен MiKTeX, содержит пробел. В этом случае необходимо использовать «краткую форму» этой директории при установке TeX4HT. Можно убедиться в том, что краткая форма имени директории выбрана правильно, запустив сессию DOS и выполнив команду `dir /p` в окне сессии DOS при использовании Windows 95/98/ME или `dir /x/p` при использовании Windows NT4/2000.

Необходимо открыть файл `TeX4HT.env` в любом текстовом редакторе и:

- В строке `tc:\path\tfm!` замените `path` «краткой формой» пути к директории со шрифтами MiKTeX. По умолчанию в MiKTeX 2.0 такой директорией является `c:\Progra~1\miktex\fonts`, так что новая строка будет иметь вид `tc:\Progra~1\miktex\fonts\tfm!`. Если соответствующая шрифтовая директория будет расположена в `c:\Program Files\miktex\fonts\tfm`.
- Необходимо добавить еще одну `t` строку, указывающую на директорию, в которой находятся метрические файлы «локальных» шрифтов, аналогично описанному выше: при установке MiKTeX 2.0 по умолчанию необходимо добавить строку `tc:\Localt~1\fonts\tfm!` (обратите внимание на то, что строка начинается с буквы `t` и в конце стоит восклицательный знак), что соответствует пути `c:\Local TeXMF\fonts\tfm`.

  Отметим, что директория `c:\LocalTeXMF\fonts` может еще не существовать, если вам ни разу не приходилось генерировать дополнительные метрические файлы кроме

тех, которые устанавливаются при инсталляции MiKTeX. (Вы можете искусственно вызвать создание такой директории и метрических файлов запустив компиляцию в LaTeX небольшого документа со шрифтом 12 пунктов). Если `c:\LocalTeXMF\` существует, то необходимо добавить `t` строку даже в том случае, если директория, на которую необходимо сослаться, еще не существует.

1. Если при установке MiKTeX использовались директории, ранее существовавшие в дереве LaTeX, необходимо добавить соответствующие `t` строки с указание на пути к этим `tfm` файлам также.
2. В строке, которая начинается `Ggswin32c`, необходимо замените `gswin32c` на полный путь (drive/path) к программе `gswin32c.exe`: в установленном по умолчанию пакете GhostScript 7.00 это `Gc:\gs\bin\gs7.00\gswin32c`. Обратите внимание на первую букву `G`, которая необходима для TeX4HT.
3. В строке, которая начинается `Gconvert`, необходимо заменить `convert` на полный путь (drive/path) к программе `convert` пакета ImageMagick: в установленном по умолчанию пакете это `Gc:\ImageMagick\convert`. Обратите внимание на первую букву `G`, которая необходима для TeX4HT.

В конце необходимо сохранить `TeX4HT.env`.

Далее, необходимо добавить директорию TeX4HT (например `c:\TeX4HT`) к вашему пути. Это можно сделать через Панель управления (Control Panel) при работе с Windows NT/Win 2000 или отредактировав `autoexec.bat` при работе в Windows 95/98 (помните, что если вы отредактировали `autoexec.bat`, вам необходимо перегрузить компьютер для того, чтобы ваши изменения вступили в силу). Наконец, необходимо скопировать файлы с расширением `*.tab` (`ht`, `htlatex`, `httex`, и `httexi`), заменив расширение на `.bat`, и переместите их в директорию, путь к которой проложен. Поскольку путь к директории MikTeX'a уже обычно проложен, то можно их поместить в эту директорию.

Обновление пакета GhostScript

При обновлении версии GhostScript скорее всего изменится имя директории, в которой будут располагаться соответствующие файлы. В этом случае необходимо внести изменения в следующие файлы, содержащие ссылки на место размещения:

`c:\TeX4HT\TeX4HT.env` (в той части, где содержатся `G`-скрипты)

Обновление пакета ImageMagick

При обновлении пакета ImageMagick скорее всего изменится имя директории, в которой будут располагаться соответствующие файлы. В этом случае необходимо внести изменения в следующие файлы, содержащие ссылки на место размещения ImageMagick:

- `c:\TeX4HT\TeX4HT.env` (в той части, где содержатся `G`-скрипты)
- `autoexec.bat` если добавлялись переменные окружения `magick_delegate_path` и `magick_module_path`. В случае если вы работаете под WinNT, необходимо изменить установки с помощью `System -> Environment` в Панели Управления (Control Panel).

Обновление пакета TeX4HT

Для установки обновленных файлов необходимо разархивировать (unzip) дистрибутив обновления в `c:\TeX4HT`, сохраняя структуру директорий. После этого необходимо проверить эту директорию в соответствии со следующим планом.

1. Проверить, снят ли атрибут "read-only" выполнив, например, в DOS-сессии команду `attrib TeX4HT.4ht`. Если результат содержит букву `R` ("read-only"), то можно запустить команду `attrib -r *.* /s` (Или просто запустить приведенную команду без всякой проверки). Можно изменить этот атрибут и с помощью какого-либо файлового менеджера, например FAR.
2. Если в обновлении имеются какие-либо `*.sty` файлы, необходимо скопировать их в поддиректорию TeX4HT в дереве LaTeX, которая была создана при инсталляции ситемы, и удалить их из `c:\TeX4HT`.
3. Если в обновлении имеются какие-либо `*.4ht` файлы, необходимо скопировать их в поддиректорию TeX4HT в дереве LaTeX, которая была создана при инсталляции ситемы, и удалить их из `c:\TeX4HT`.
4. Теперь необходимо обновить базу данных файлов MiKTeX, выполнив `Пуск -> Программы -> MikTeX 2 -> MiKTeX Options (Start Menu -> Programs -> MikTeX 2 -> MiKTeX Options)`. На вкладке `General`, нажмите кнопку `Refresh Now`.
5. Теперь можно удалить все `*.c` файлы.
6. Если в обновлении имеются новые `*.tab` файлы, необходимо переименовать их в `*.bat` и переместить в директорию, путь к которой известен, и где они заменят помещенные там ранее одноименные файлы.
7. Необходимо также удалить файл `c:\TeX4HT\TeX4HT.fls`: это файл, содержащий описание используемых файлов для минимизации дискового поиска, при появлении новых `*.htf` файлов должен быть пустым.

2.2.8 Локализация и отладка пакета, использование кодировки Unicode

Прежде чем приступать к преобразованию сложных документов необходимо проверить ра-

ботоспособность системы на простых примерах, поставляемых вместе с системой. Дистрибутив TeX4HT содержит два тестовых файла. Каждый создает HTML файл, содержащий одну строку текста, один одиночный символ в виде gif-файла (“glyph-gif”), и один многосимвольный gif-файл. Исходные тестовые файлы приведены в таблице 1.

Таблица 1 Тестовые файлы

Testa.tex	Testb.tex
\documentclass {article} \usepackage {TeX4HT} \begin {document} A single $\heartsuit$ and a full suit: \(\clubsuit \diamondsuit \heartsuit \spadesuit \). \end {document}	\documentclass {article} \begin {document} A single $\heartsuit$ and a full suit: \(\clubsuit \diamondsuit \heartsuit \spadesuit \). \end {document}

Предполагая, что все переменные окружения и пути установлены правильно, для тестирования необходимо:

- Запустить DOS-сессию и перейти в какую-нибудь рабочую директорию, *отличную* от `c:\TeX4HT4;`
- Скопировать `testa.tex` и `textb.tex` из `c:\TeX4HT` в эту директорию;
- Прежде всего необходимо оттестировать вариант, который не требует явного вызова пакета LaTeX

  `htlatex testb`

По этой команде сначала запускается LaTeX после чего дважды вызывается ImageMagick/GhostScript для генерации gif-файлов. После запуска получившегося файла `testb.htm` в результате в окне Internet Explorera появится картинка

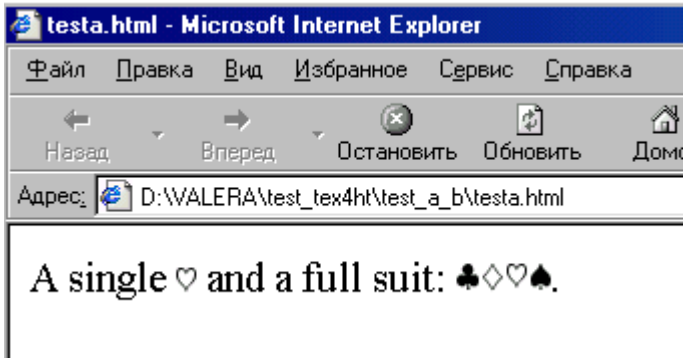

Рисунок 1 – Экранный вывод

- Далее, тестируется второй вариант, при использовании которого пакет TeX4HT явно указывается в исходном файле (см. testa.tex, строку `\usepackage {TeX4HT}`). Для этого используется команда

  `ht latex testa`

По этой команде будет запущен LaTeX, но поскольку “glyph-gif” уже имеются, будет только одно обращение к ImageMagick/GhostScript. Получится файл `testa.htm`, который при воспроизведении даст такую же картинку.

Дальнейшее тестирование проводилось с файлом, `testbold.tex`, который был написан нами для проверки возможности использования русского языка и определения качества вывода формул. Текст файла приведен далее в таблице 2. Там же приведен конфигурационный файл `win2.cfg`, который пришлось написать для корректного отражения русского языка в получающемся `html`-файле и правильного размещения формул в строке.

Таблица 2 Тестирование русского языка и формул

Testbod.tex	Win2.cfg
\documentclass {article} \usepackage[cp1251]{inputenc} \usepackage[english,russian]{babel} \usepackage[win2,html,3,sections+]{TeX4HT} \begin {document} Hello world!\\ \textbf{Hello world!}\\ Привет, \(\delta=(c^2/2\pi \omega \sigma)^{1/2} \) мир! (this is usual font)\\ \textbf{Привет, мир!} (this is bold font)\\	% win1251.cfg \Preamble{} \begin{document} \Configure{charset}{charset=windows-1251} \Configure{PicMath} {}{}{} { class=\string"math\string" align \string"absbottom\string"} \EndPreamble

$\delta=(c^2/2\pi \omega \sigma)^{1/2}$ \textit{Привет, мир!} (this is italic font)\\ $$ \delta=(c^2/2\pi \omega \sigma)^{1/2} $$ $\pm$ $\S$ \end {document}	

Преобразование осуществлялось с помощью команды

```
ht latex testbold.tex
```

Полученный в результате преобразования файл `testbold.html` после воспроизведения в браузере имеет вид, показанный на рисунке 2.

Следует отметить, что в первоначальном варианте пакет TeX4HT не поддерживал русский язык, а точнее, отсутствовали соответствующие шрифты. Когда после предварительной отладки это было обнаружено, Эйтан Гурари изготовил файл `larm.htf` — описатель русских фонтов типа `larm` и т.д. Мы его поместили в директорию `c:\TeX4HT\ht fonts\alias\cyrillic` под именем `la.htf`. Это позволило нам правильно использовать все шрифты, начинающиеся на `la*` (`labx*.tfm, larm*.tfm, latt*.tfm` и т.д.).

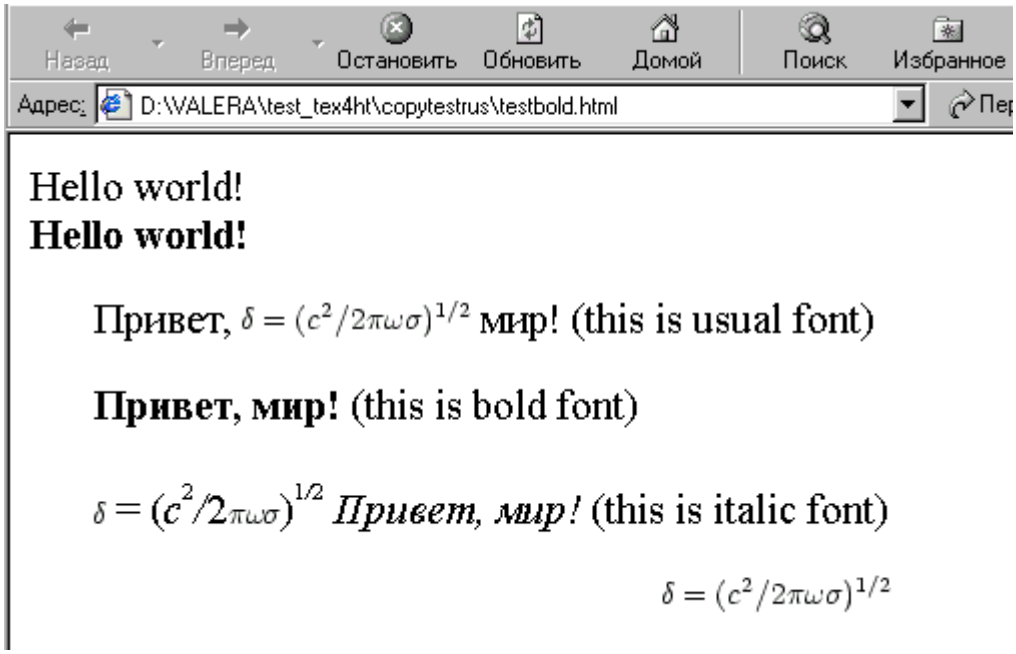

Рисунок 2 - Вывод текста на русском языке и простейших формул

Абсолютно этот же файл был использован нами для воспроизведения шрифтов типа

mgm*.tfm. Для этого мы скопировали полученный от Гурари шрифт larm.ht под именем mgm.ht и переписали соответствующие (первую и последнюю) строки внутри этого файла. Таким образом, мы получили доступ к использованию шрифтов mgm*.tfm (это Garamond). Впоследствии аналогично нами были сделаны шрифты Monotype (mbk*.tfm) и Paragraph (tdy*.tfm, tzc*.tfm). Основной шрифт la.htf теперь входит в стандартную поставку TeX4HT, а остальные можно сделать по мере надобности.

Следует также отметить, что в своем простейшем варианте, описанном выше, TeX4HT преобразует математические формулы в виде рисунков, gif-файлов. Но дальнейшая работа с пакетом показала, что он имеет возможность конвертировать формулы в язык разметки MathML. Для этого требуется более тонкая настройка пакета и использование его продвинутых возможностей.

По целому ряду соображений, которые будут ясны из дальнейшего, при преобразовании сложных документов с большим количеством формул, использующих русский язык в качестве основного языка текста для получения формул, написанных с помощью языка разметки MathML, пришлось перейти на UNICODE-кодировку. Не вдаваясь в описание самой этой кодировки (см., например http://www.unicode.org/), скажем только, что текст в формате XHTML состоял из «entities», т.е. каждая буква полученного текста на русском языке и каждый элемент формулы состоял из изображения кода буквы в формате UNICODE. Например

```
<head>
<title>1
&#x041A;&#x043B;&#x0430;&#x0441;&#x0441;&#x0438;&#x0444;&#x0438;&#x043
A;                              &#x0430;&#x0446;&#x0438;&#x044F;&#x0444;
&#x0438;&#x0437;&#x0438;&#x0447;&#x0435;&#x0441;&#x043A;
&#x0438;&#x0445;&#x0432;&#x0435;&#x043B;&#x0438;&#x0447;
&#x0438;&#x043D;&#x044B;        &#x0438;        &#x0442;&#x0435;&#x043D;
&#x0437;&#x043E;&#x0440;&#x044B;
</title>
```

Этот загадочный код соответствовал русской части текста

```
<head>
<title>1
Классификация физических величины и тензоры
</title>
```

При этом браузер понимал этот текст и воспроизводил его правильно. (Здесь и везде далее мы проводили все работы с браузером Microsoft Internet Explorer версии 5.5 и 6.0). Конечно, с таким положением можно было бы смириться, но помимо разбухания текста — по восемь символов на одну букву — такой текст невозможно править в формате HTML, необходимо постоянно исполь-

зовать кодировочную таблицу. Аналогичным же образом выводились греческие буквы и ряд специальных символов внутри формул, записанных в разметке MathML. Латинские буквы при этом воспроизводились правильно. Для приведения этого текста (XHTML) в нормальный вид был разработан скрипт `utf2win2.wsf (Приложение A)`, который работал в режиме постпроцессора. После получения файлов, преобразованных из LaTeX в XHTML, они все обрабатывались этим скриптом, в результате чего русские буквы переводились в кодировку win-1251, а большинство греческих букв и символов из загадочной конструкции `"&#x0391;"` переводилась в понятную аббревиатуру `"&Alpha;"`, которую браузер понимал также хорошо. В результате такого преобразования размер файлов значительно сокращался, а при необходимости внесения небольших правок это можно было легко сделать с помощью любого текстового редактора

2.2.9 Технология преобразования

Для дальнейшей отработки технологии преобразования текстов из формата LaTeX в формат XHTML использовался текст первой части пособия по классической электродинамике, написанного проф. Котельниковым И.А. и Яковлевым В.И. Исходный документ был разработан в системе LaTeX2e и содержал более 50 файлов основного текста, и свыше 100 рисунков в формате eps. Помимо этого для написания пособия был разработан собственный класс документов TextBook.cls, а также целый ряд макроопределений и новых окружений. При преобразовании ставилась задача получить гипертекстовый документ, который можно с высоким качеством воспроизводить с помощью стандартных средств.

Поскольку к настоящему моменту только мало распространенный браузер Mozila может воспроизводить формулы в разметке MathML, мы в своей работе ориентировались на появившийся в 2002 году плагин MathPlayer фирмы Design Science (http://www.dessci.com/en/products/mathplayer/default.htm), который позволяет воспроизводить математические формулы в браузере Microsoft Internet Explorer 5.5 и 6.0. Этот плагин является бесплатным и каждый может его скачать с указанной выше страницы. Для того чтобы формулы в разметке MathML воспроизводились, необходимо чтобы тексты в формате XHTML удовлетворяли определенным условиям, с которыми можно познакомиться на странице, посвященной плагину MathPlayer. Автор пакета TeX4HT включил соответствующую опцию в настройки пакета. Таким образом, для осуществления преобразования нами в директорию с исходными TeX-файлами был помещен файл XHMLATEX2EM.BAT, который запускал нужные программы пакета. Это файл был разработан П. Витоном, и его можно свободно загрузить со страницы http://www.arch.ohio-state.edu/crp/faculty/pviton/support/TeX4HT.html. Единственное изменение, которое мы внесли в этот файл – мы убрали загрузку «на лету» переменных окружения, необходимых для работы системы. Эти переменные мы устанавливаем в файле `autoexec.bat`.

Использование этого запускающего файла позволило нам также все настройки пакета TeX4HT вынести в специальный конфигурационный файл, а не помещать в исходный файл LaTeX. Используемый нами конфигурационный файл `mat.cfg` имеет вид,

```
% Configuration file for Mathplayer
\Preamble{htm,mathplayer,3,notoc*,sections+}
    \begin{document}
    \EndPreamble
```

Опции пакета TeX4HT перечислены в фигурных скобках после команды `\Preamble`. Первые две не нуждаются в комментариях, число 3 (возможны варианты 1,2,3 и 4) задает желаемую глубину иерархии гипертекстовых страниц. Эта иерархия отражает логическую структуру содержимого, определенную командами секционирования `\part`, `\chapter`, `\section` и т.д. Опция `sections+` обеспечивает наличие гипертекстовых ссылок не только от элементов оглавления к соответствующим разделам, но и гиперссылок от оглавлений разделов назад к оглавлению.

Особо следует сказать об опции `notoc*`, которая была добавлена в пакет автором по нашей инициативе. Дело в том, что при тестировании сложного документа обнаружились проблемы с конструкция LaTeX вида

```
\section*{Векторный анализ}
\addcontentsline{toc}{section}{Векторный анализ},
```

которая используется для присвоения соответствующему разделу имени «`Векторный анализ`» но без нумерации (наличие *). При использовании такой конструкции соответствующее название не включается в содержание. Для включения его в содержание используется вторая строка кода, которая его включает непосредственно. При переводе такой конструкции в HTML заголовок «`Векторный анализ`» дважды попадает на соответствующую страницу и генерирует 2 ссылки на одно и тоже место – страницу с основным текстом. Можно решить, конечно, эту проблему закомментировав подобные строки в исходном коде, но мы старались по возможности таким образом построить преобразование исходного текста в HTML, чтобы не вносить никаких изменений в исходный текст. По нашей просьбе автор пакета добавил опцию `notoc*`, которая и решает эту проблему – при использовании этой опции удвоение в содержании и ссылках не происходит. С подробным описанием возможных опций пакета можно познакомиться либо на сайте автора пакета, где имеется руководство по использованию пакета (http://www.cis.ohio-state.edu/~gurari/TeX4HT/), либо в книге [5].

При первоначальной попытке преобразовать текст мы столкнулись с целым рядом проблем. Одна из них — отсутствие на первом этапе отладки секционирования текста и его структурирования по главам, параграфам и т.д. Оказалось, что разработанный нами для написания учеб-

ника класс документов TextBook.cls не был известен системе. По нашей просьбе автор разработал дополнительный стилевой файл `TextBook.4ht`, который и решил эту проблему. При отладке и настройке мы также обнаружили, что пакет не понимает конструкцию

```
\begin{subequation}
....
\end{subequation}
```

Соответствующие изменения также были включены в стилевой файл.

Определенные проблемы возникли с использованием полужирного математического шрифта. После анализа возможностей MathPlayer мы пришли к выводу, что команда `\mathbf` может быть реализована на языке разметки MathML с помощью тага `<m:mathstyle>` и Гурари внес соответствующие изменения в стилевые файлы, но при этом еще в версии *.4ht от 5.09.2002 содержалась ошибка, которая приводила к разбиению операторов типа `lim`, `grad` и т.д. на отдельные буквы и неверно воспроизводилась с помощью MathPlayer. В настоящее время эти ошибки устранены и более поздние версии обновлений шрифтов и стилевых файлов позволяют использовать полужирный шрифт в математических выражениях.

Таким образом, после всех исправлений и дополнений процесс преобразования запускается с помощью файла RUN.BAT, который имеет вид

```
call clean.bat
call XHMLATEX2EM.bat webbook.tex mat.cfg
WSCRIPT c:\utils\UTF2WIN2.WSF . /sub
```

Файл `clean.bat` очищает директорию от служебных файлов, сгенерированных при прошлой компиляции. Файл `mat.cfg` содержит настройки TeX4HT, как это было описано выше. После получения HTML-файлов в кодировке UNICODE с помощью скрипта `UTF2WIN2.WSF` выполняется преобразование русских букв в кодировку win-1251. Отдельного обсуждения заслуживает вопрос о преобразовании графических файлов и размещении их на web-страницах.

2.2.10 Форматы представления графических материалов

Существует множество разнообразных графических программ, и всё время появляются новые сообразно развитию компьютерной техники. Все драйверы, с которыми работает LaTeX, умеют вставлять рисунки, но обычно это умение распространяется на 3-4 наиболее популярных формата графических файлов в той операционной системе, для которой реализован драйвер. Если работа выполняется в операционной системой MS DOS, то используемые программы просмотра и печати dvi-файла, вероятно, взяты из набора emTeX. Выбор соответствующего драйвера emTeX почти не оставляет поля для маневра — импортируемый рисунок должен иметь формат PCX. Значительно лучше обстоит дело, если работа выполняется в Win-

dows и используется программа dviwin для просмотра и печати документа. Она может импортировать практически любой рисунок (нужно только найти подходящий *графический фильтр,* причем подходят все графические фильтры от текстового редактора Word for Windows 3.x). К сожалению, dviwin не умеет вращать рисунки. Наиболее популярным в настоящее время при работе в системе Windows является драйвер dvips. Он умеет импортировать рисунки на языке PostScript, и еще кое-каких других форматов. В частности, драйвер dvips умеет делать всё то, что умеют драйверы emTeX. При работе с этим драйвером, а мы ориентировались именно на него, наиболее естественным выбором должен быть перевод всех импортируемых рисунков в формат PS, а еще лучше — в формат EPS. Подготовленный нами конспект лекций содержал более 100 рисунков в формате EPS. Основным же форматом при воспроизведении страниц в формате HTML является формат GIF и/или JPEG. Формат JPEG является наиболее предпочтительным при воспроизведении многоцветных картинок типа фотографий, в то время как основная масса иллюстраций в научно-технических документах представляет собой черно-белые (двухцветные или с оттенками серого) графики, схемы или диаграммы. Для качественного воспроизведения таких графических материалов подходит формат GIF. В пакете TeX4HT для преобразования графических материалов используется пакет ImageMagic, обращение к которому закладывается при настройке TeX4HT в файле TeX4HT.ENV. Эти строки имеют вид

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% dvi-to-gif
% dvips options
%   magnification: -x mag  (e.g., -x 1200)
%   page size:     -T x,y  (e.g., -T 14in,14in)
Gif exist zz%%4.ps DEL zz%%4.ps >nul
Gif exist zz%%4.ppm DEL zz%%4.ppm >nul
Gif exist %%3 DEL %%3 >nul
Gdvips  -Pcmz -Pamz -mode ibmvga -D 110 -f %%1 -pp %%2 > zz%%4.ps
Gc:\gs\gs6.50\bin\gswin32c     -dNOPAUSE     -sDEVICE=ppmraw     -
dTextAlphaBits=4        -dGraphicsAlphaBits=4        -r110x110        -
sOutputFile=zz%%4.ppm -q zz%%4.ps -c quit
Gc:\ImageMagick\convert  -crop  0x0  -density  110x110  -transparent
#FFFFFF zz%%4.ppm %%3
Gif exist zz%%4.ps DEL zz%%4.ps >nul
Gif exist zz%%4.ppm DEL zz%%4.ppm >nul
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Как видно из приведенного фрагмента, последовательность преобразования графических

материалов имеет вид DVI -> PS -> PPM -> GIF, при этом происходит переименование графических файлов по схеме, которую создает пакет TeX4HT. Следует отметить, что вместо пакета ImageMagic возможно использование пакета NETPBM, который хорошо известен тем, кто широко использует в своей работе LaTeX. Мы провели такие эксперименты, но какого-либо преимущества у этого пути не нашли.

Еще один вопрос, который возникает при преобразовании сложного документа с большим количеством иллюстраций, это повторное преобразование графических файлов из одного формата в другой, хотя они уже имеются в нужном виде. Этого можно избежать, если внести соответствующие настройки в конфигурационный файл. Была сделана попытки устранения этого дублирования, но при использовании такого конфигурационного файла вопрос оказался более сложным – он связан с тем, что TeX4HT переименовывает все файлы, как графические, так и файлы, соответствующие отдельным web-страницам. При этом ссылки внутри соответствуют новым именам. Это необходимо учитывать при сохранении и использовании ранее преобразованных графических файлов, что составит предмет дальнейших исследований.

3 Представление моделей физических явлений в интернете

Программы, моделирующие физические явления, которые позволяют представить результаты расчетов в виде графических и анимационных изображений, являются действенным инструментом обучения в арсенале преподавателей естественно-научных дисциплин. Не подменяя живое общение преподавателя и студента, они дополняют традиционное изложение предмета в виде набора простых моделей, которые можно объяснить “на пальцах”, средствами визуализации теоретических построений. Особую ценность представляют программы двойного назначения, которые можно использовать как для обучения студентов на практических занятиях и лекционных демонстраций, так и для самостоятельных занятий обучаемых через интернет при предварительном ознакомлении с материалом или для закрепления полученных знаний. На сегодняшний день в интернете размещено большое количество учебного иллюстративного материала как посвященного физике, так и другим дисциплинам. Большая часть этого материала представлена в сети в виде схем, графиков и анимированных иллюстраций. Как правило, при этом используются апплеты, анимированные gif-файлы, различные видеофрагменты, т.е. все то, что может быть воспроизведено в браузере стандартными средствами этого браузера или дополнительными плагинами. В последнее время все шире используется флеш-технология. К сожалению, большинство таких иллюстраций лишены интерактивности, т.е. пользователь знакомится с иллюстрацией того или иного явления при значениях параметров, выбранных авторами. Оптимальным с точки зрения обучения

является интерактивное взаимодействие пользователя с моделью, которое позволяет ему, меняя параметры модели, всесторонне ознакомиться с изучаемым явлением.

3.1 Специализированные программы, использование web-серверов

3.1.1 Выбор языка программирования

Для разработки моделирующих программ мы использовали язык программирования MatLab [15-17], который является мощным интерпретирующим языком сверхвысокого уровня. Использование подобного языка-интерпретатора позволяет достаточно быстро разрабатывать и отлаживать приложения, а благодаря наличию средств создания графического интерфейса создавать удобные в использовании продукты без особых усилий по собственно программированию. MatLab WEB Server позволяет использовать разработанные для локальной работы программы с минимальными изменениями для предоставления доступа к ним через интернет.

3.1.2 Закрытые и открытые моделирующие программы

При разработке учебных и/или демонстрационных программ первый вопрос, который приходится решать – разрабатываемая программа должна быть открытой или закрытой. Закрытой программой мы называем такую программу, при использовании которой возможно задание различных параметров, определяющих моделируемый процесс, а сам алгоритм производимых вычислений и программа являются неизменными и, как правило, недоступными. Открытой же программой мы называем такую программу, которую может видоизменять пользователь для получения тех или иных ответов. Если говорить об обучении предметной области (физике, математике, химии), то предпочтительным является тщательно разработанная закрытая программа с удобным графическим интерфейсом, использование которой не требует знания языка программирования. Достаточно научиться пользоваться такой программой. Такая же программа является предпочтительной при проведении лекционных демонстраций. Если же целью обучения является обучение моделированию, то предпочтительной является открытая программа, текст которой доступен пользователю, и он может сам видоизменять ее, решая новые задачи.

Поскольку целью нашей работы была отработка технологии создания моделирующих (обучающих) программ и/или демонстрационных программ по физике (в данном конкретном случае — по электродинамике) с предоставлением к ним доступа через интернет, то нами был выбран вариант разработки закрытой программы, текст которой недоступен пользователю.

3.1.3 Исполнение программ на стороне клиента и стороне сервера

Второй вопрос состоит в том, где проводить собственно расчеты — на стороне клиента или на стороне сервера. Расчеты на стороне клиента сужают спектр возможных применений, ограничивая платформы и требуя наличия у клиента соответствующей системы (Matlab, MathCad, Mathematica) или передачи клиенту соответствующего exe-файла, что является, вообще говоря, нарушением идеологии интернет, особенно в связи с развитием вирусов, и воспринимается большинством пользователей неоднозначно. В результате учета этих обстоятельств нами было принято решение вычисления проводить на стороне сервера, а клиенту передавать результаты в виде статических или анимированных изображений.

3.1.4 Совмещение HTML и Matlab

Благодаря наличию в системе `Matlab` версии 5.3 и старше пакета MatLabWebServer, удалось реализовать работу с закрытой от пользователей программой на языке MatLab по протоколу HTTP (см. рис.3.). Пользователь, используя приготовленную HTML-страницу, заполняет форму, которая по его команде передает данные по протоколу HTTP программе matweb.exe, которая обрабатывает их, запускает интерпретатор MatLab, загружает в него требуемое приложение и передает ему полученные данные.

В рамках запущенной программы производятся вычисления и построение соответствующих графиков, после чего они передаются в выходную HTML-страницу, которую и получает пользователь.

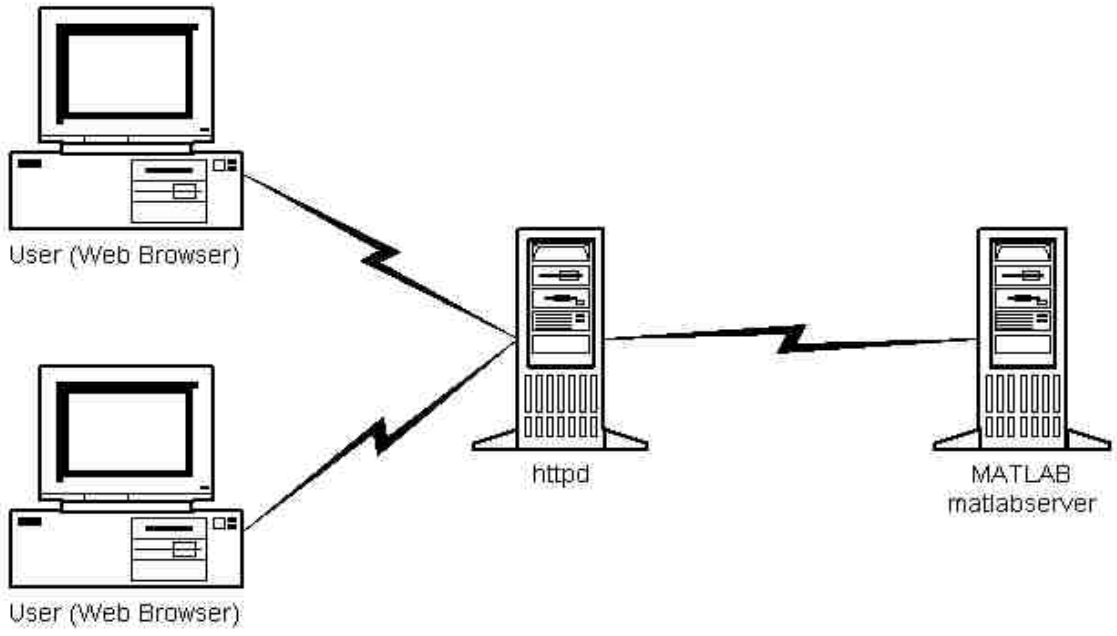

Рисунок 3 – Принципиальная схема взаимодействия пользователя с MatlabWEB-сервером

3.2 Matlab web-сервер и технология его использования

Реализованная нами система работала на сервере под управлением операционной системы Windows 2000 (или NT), на которой был установлен web-сервер IIS (или Apache). Использовалась также конфигурация, где Web-сервер и Matlab Web Server были установлены на разных компьютерах в пределах единой локальной сети.

В отличие от предоставляемых фирмой MathWork образцов, где все задачи вместе со своими html и m-файлами располагаются в одном каталоге, нами была разработана типовая система каталогов с соответствующими правами доступа. Каждая задача размещалась в отдельном каталоге, внутри которого размещался каталог для m-файлов (исполняемых файлов MatLab) и, возможно, каталоги для хранения временных файлов (графических или анимационных, сгенерированных программой), файлов справки с описанием задачи т.д. Такая структура представляется нам более целесообразной. Для ввода параметров задачи и вывода результатов расчетов были разработаны html-шаблоны, обеспечившие дополнительную интерактивность по сравнению с типовыми шаблонами, поставляемыми в составе программного обеспечения MatLab Web Server.

3.2.1 Структура каталогов (папок) и технология подготовки моделей

Создание структуры
1. В консоли веб-сайта необходимо создать виртуальный каталог

```
/cgi-bin
```

Этот каталог должен иметь разрешение Web-сервера на исполнение программ. Каталог может иметь имя, отличное от /cgi-bin, но тогда везде ниже нужно сделать соответствующие поправки. Имя каталога /cgi-bin указывается в аргументе action тага form в HTML-файле, принимающем расчетные параметры от пользователя. Например:

```
<form action="/cgi-bin/matweb.exe" method="POST">
```

Необходимо поместить в /cgi-bin файлы

```
matweb.exe
matweb.conf
```

После стандартной установки MatLab эти файлы располагаются в папке matlab\webserver\bin, где matlab - корневая папка установленного пакета MatLab, например: C:\Matlab6p1. Типичный сценарий конфигурирования Web-сайта состоит в том, что папка matlab\webserver\bin объявляют виртуальным каталогом /cgi-bin Web-сайта. В другом типичном сценарии создают папку cgi-bin в корневом каталоге web-сайта. При этом созданная

папка автоматически станет виртуальным каталогом с абсолютным адресом /cgi-bin.

Исполняемый файл matweb.exe является клиентом сервера MatLab. Он передает запросы, полученные Web-сервером, серверу MatLab, используя настройки, заданные в конфигурационном файле matweb.conf.

2. Необходимо для каждой вновь создаваемой задачи редактировать конфигурационный файл matweb.conf. Конфигурационный файл состоит из секций, которые начинаются с названия MatLab-приложения, заключенного в квадратные скобки:

```
[matlab_application_name]
mlserver=matlabserver_host_name

mldir=mfiles_folder_name
```

Имя MatLab-приложения matlab_application_name должно совпадать с именем m-файла, являющегося основной исполняемой программой для данного приложения. Указанный m-файл должен находится в папке mfiles_folder_name, где могут также находится другие m-файлы, используемые MatLab-приложением. Сетевое имя matlabserver_host_name сервера MatLab можно указать в виде доменного имени компьютера, где установлен сервер MatLab (и где находится папка mfiles_folder_name), либо в виде его IP адреса. Например:

```
[skin]
mlserver=matlab.fija.nsu.ru
mldir=E:\webmatlab\ldin\skin\mfiles
```

3. На компьютере matlabserver_host_name создается папка mfiles_folder_name. В папке mfiles_folder_name\.., родительской по отношению к mfiles_folder_name, создается папка tmpfiles для хранения временных файлов, которые может создавать приложение, а также папка help для документации. Например:

```
E:\webmatlab
     |--eldin
        |--skin
           |--mfiles
           |--tmpfiles
           |--help
```

Далее предполагается, что Web-сервер сконфигурирован таким образом, что E:/webmatlab является корневым каталогом сайта matlab.tutornet.ru. Поэтому приведенному выше дереву папок сопоставлено следующее дерево виртуальных каталогов:

```
/
|--eldin
   |--skin
      |--mfiles
      |--tmpfiles
      |--help
```

При этом родительская папка MatLab-приложения `mfiles_folder_name\..` (`E:\webmatlab\eldin\skin`) одновременно является (корневой) папкой Web-приложения `web_applicaion_name (/eldin/skin)`. Убедитесь, что виртуальный каталог `web_applicaion_name/tmpfiles` имеет разрешение на запись для сервера MatLab. Рекомендуется, чтобы все html-файлы, используемые Web-приложением (в том числе `default.htm`, `panel.htm`, `info.htm`, `warning.htm`, `output.htm`), были размещены в (корневом) каталоге Web-приложения, т.е в родительской папке `mfiles_folder_name\..` MatLab-приложения (`E:/webmatlab/eldin/skin`.).

4. Необходимо далее разместить в каталоге `mfiles_folder_name` m-файлы MatLab-приложения и отредактировать корневой m-файл MatLab-приложения. Он должен начинаться с декларации функции, принимающей объект с входными параметрами `objIn`:

```
function rs = skin(objIn, outFile)
```

Название функции (в данном случае — `skin`) не обязательно должно совпадать с названием приложения (с названием m-файла), однако рекомендуется следовать такому правилу. Объект `objIn` создается программой `matweb.exe`. Он имеет поименованные свойства с именами, совпадающими с именами параметров, переданных из панели управления `panel.htm` Web-приложения (см. ниже). Например, если в `panel.htm` задан параметр `par1`, то в m-файле его значение содержит свойство `objIn.par1`. Объект `objIn` имеет также встроенные свойство `mldir`, содержащее значение `mfiles_folder_name`, заданное в конфигурационном файле `matweb.conf` в соответствующей секции `[matlab_application_name]`. Параметр `outFile` используется при отладке приложения и далее не обсуждается.

Непосредственно после декларации функции рекомендуется выполнить переход в каталог `tmpfiles`, предназначенный для хранения временных файлов, сгенерированных приложением. Например:

```
cd(objIn.mldir);
cd('../tmpfiles');
```

Первая команда cd здесь выполняет переход в папку mfiles_folder_name. В соответствии с рекомендованной структурой каталогов (см. п.3), эта папку является сестринской для папки tmpfiles. Вторая команда cd делает tmpfiles рабочей папкой.

Имя временного графического или анимационного файла, генерируемого приложением и записанного в папку tmpfiles, необходимо передать функции htmlrep (см. ниже), генерирующей html-вывод, в виде значения свойства objOut.GraphFileName выходного объекта objOut. Например:

```
objOut.GraphFileName = sprintf("%sskin.jpeg', mlid);
mpgwrite(frame, map, objOut.GraphFileName, [1, 0, 1, 0, 10, 16,
20, 50]);
```

Библиотечная функция sprintf используется для генерации уникального имени файла: вместо параметра %s она подставляет уникальный номер процесса mlid, чтобы исключить порчу графического файла, сгенерированного в другом процессе. Рекомендуется включать расширение имени графического файла в значение objOut.GraphFileName, чтобы по возможности использовать единый шаблон выходного файла output.htm для разных приложений, а также в тех случаях, когда приложение может генерировать графические изображения разных типов в зависимости от входных параметров.

Приложение завершается вызовом библиотечной функции htlmrep, которая генерирует html-вывод, направляемый пользователю в окно с именем frmMain (см. ниже). Перед вызовом функции htlmrep необходимо перейти в родительскую папку приложения, где размещен шаблон вывода output.htm. Например:

```
cd('..');
if (nargin == 1)
rs = htmlrep(objOut, 'output.htm');
elseif (nargin == 2)
rs = htmlrep(objOut, 'output.htm', outFile);
end
```

5. Необходимо отредактировать html-файлы в родительской папке mfiles_folder_name/.. приложения. Эта папка одновременно является родительским каталогом web-приложения, поскольку в соответствии со структурой каталогов, рекомендованной в п.3, все папки, относящиеся к одному приложению, находятся в одной родительской папке.

Файл default.htm определяет общий вид окна ввода-вывода, с которым работает пользователь. Этот файл определяет, что окно содержит 3 фрейма: левый более узкий фрейм

frmPanel с панелью ввода параметров, правый более широкий фрейм frmMain для вывода результатов расчета и верхнюю панель навигации frmNav по сайту:

```
<frameset cols="220,*" SCROLLING="auto" frameborder="yes">
<FRAME SRC="panel.htm" NAME="frmPanel" id="frmPanel"
MARGINHEIGHT=0 MARGINWIDTH=0 BORDER=1 FRAMEBORDER=0 FRAMESPAC-
ING=0>
<frameset rows="23,*" framespacing="0" frameborder="0" bor-
der="0">
<frame src="/common/html/nav.html" name="frmNav" id="frmNav"
frameborder="0" scrolling="no" noresize marginwidth="0" margin-
height="0">
<FRAME SRC="info.htm" name="frmMain" id="frmMain" FRAMEBORDER=0
FRAMESPACING=0 SCROLLING="auto">
</frameset>
</frameset>
```

Как правило, default.htm не нуждается в редактировании. Он загружает в левый фрейм frmPanel файл panel.htm, в правый фрейм frmMain -- файл info.htm, а верхний фрейм frmNav -- файл /common/html/nav.html, расположенный в виртуальном каталоге /common/html/, где хранятся файлы, используемые всеми приложениями.

Файл panel.htm загружен в левый фрейм в течении всего цикла работы Web-приложения. В этом файле необходимо задать имя MatLab-приложения matlab_application_name и название фрейма frmMain, куда следует направить результат расчета. Например:

```
<form  name=panelForm
action="http://matlab.fija.nsu.ru/cgi-bin/matweb.exe"
method="POST"
target="frmMain"
onSubmit="return checkForm(this);"
>
<input type="hidden" name="mlmfile" value="skin" />
  . . .
</form>
```

Имя фрейма frmMain для вывода результатов передается в виде значения атрибута target. Имя MatLab-приложения matlab_application_name передается в виде значения атрибута value скрытого поля ввода input с именем name="mlmfile". Обработчик события onSubmit="return checkForm(this);" указывает, что перед отправкой введенных пользователем данных выполняется функция checkForm, которая осуществляет проверку введенных значений и выполняет некоторые другие действия (см. ниже). Функция checkForm, должна быть определена в блоке

```
<SCRIPT LANGUAGE=JSCRIPT><!--
```

```
function checkForm(f) {
. . .
}
--></SCRIPT>
```

Ее содержание зависит от набора полей ввода параметров. Примеры полей ввода параметров и функции их проверки содержатся в файлах panel.htm для отлаженных задач.

Типовая функция checkForm содержит скрипт, который заменяет в левом фрейме файл info.htm, первоначально загруженный туда, согласно определению этого фрейма (см. значение атрибута src тага frame в файле default.htm). Файл info.htm должен содержать краткую постановку задачи и, возможно, ссылки на более развернутое описание. Функция checkForm загружает в правый фрейм файл warning.htm, содержащий предупреждение, что расчет может занять длительное время.

После завершения работы MatLab-приложения в левый фрейм загружается файл, сгенерированный MatLab-приложением из шаблона output.htm. Этот файл выводит сообщение, что расчет окончен и производится загрузка сгенерированного рисунка или видео-изображения. В момент окончания загрузки графического файла это сообщение исчезает, а вместо него выводится само изображение. В настоящее время для вывода изображений в формате gif используется объект img

```
<img
src="http://www.phys.nsu.ru:8000/eldin/skinest/tmpfiles/$GraphFileName
$"
     id="oImg"
     align="left"
     style="border:1px solid; background-color:#ffffff;"
>
```

а при выводе видео-изображения в формате mpeg используется объект embed.

```
<EMBED
src="http://matlab.fija.nsu.ru/eldin/skin/tmpfiles/$GraphFileName$"
     id="oImg"
     align="left"
     loop="true"
     style="border:1px solid; background-color:#ffffff;">
</EMBED>
```

Для указания имени сгенерированного файла в шаблоне следует использовать обозначение `$GraphFileName$`. Этому обозначению должен предшествовать абсолютный адрес виртуального каталога `web_applicaion_name/tmpfiles`, соответствующего папке `tmpfiles` для хранения временных файлов. Абсолютный адрес должен начинаться с указания протокола (`http://`) и доменного имени сервера `matlabserver_host_name (matlab.fija.nsu.ru)`. Использование полной адресации обязательно для тех приложений, Web-интерфейс которых может располагаться на других Web-сайтах.

Безопасность

Хороший стиль администрирования Web-сервера предполагает, что m-файлы должны быть недоступны по протоколу http. С другой стороны, удобно хранить все файлы, относящиеся к одному приложению в одном каталоге, как это сделано в схеме, рекомендуемой настоящей инструкцией. В связи с этим для предотвращения доступа пользователей к текстам m-файлов, необходимо в консоли Web-сервера явно запретить чтение файлов из каталога `mfiles` в родительском каталоге каждого приложения.

Панель управления

Файл `panel.htm` содержит скрипт, который проверяет, что панель управления встроена в левый фрейм, и при необходимости загружает в окно браузера файл `default.htm`. Поэтому если просто ввести имя файла `panel.htm` в строке адресов браузера, то будут загружены все необходимые фреймы. Правильная работа панели управления гарантируется при точном следовании рекомендациям по наименованию фреймов, изложенных в данном документе.

Помимо приема от пользователя значений входных параметров задачи, типовая панель управления определяет свойства оборудования и программного обеспечения, имеющегося у пользователя. Эти сведения можно использовать в MatLab-приложении для автоматического выбора размера и разрешения графического файла. Ниже приводится список свойств входного объекта, которые можно использовать для указанных целей, и значение этих свойств, которые присваиваются в panel.htm:

```
objIn.availHeight = window.screen.availHeight % высота окна
внутри браузера
objIn.availWidth = window.screen.availWidth   % ширина окна
внутри браузера
objIn.bufferDepth = window.screen.bufferDepth % число битов
цвета в буфере
objIn.colorDepth = window.screen.colorDepth   % число битов
```

```
цветовой палитры
objIn.height = window.screen.height           % полная высота
экрана
objIn.width = window.screen.width             % полная ширина
экрана
objIn.cpuClass = window.navigator.cpuClass    % тип процессора
objIn.platform = window.navigator.platform      % тип операционной
системы
```

Приложение не должно использовать переменные с указанными именами для иных целей.

Панель вывода

В этом разделе приведены дополнительные сведения о содержании файла `output.htm`. Файл `output.htm` содержит скрипт, который проверяет, что панель вывода встроена в левый фрейм, и при необходимости загружает в окно браузера файл `default.htm`. Поэтому если просто ввести имя файла `output.htm` в строке адресов браузера, то будут загружены все необходимые фреймы. Правильная работа панели вывода гарантируется при точном следовании рекомендациям по наименованию фреймов, изложенных в данном документе.

3.2.2 Статические и динамические модели и создание «мультфильмов»

Большинство представляющих интерес демонстраций и, соответственно, программ, обеспечивающие получение этих демонстраций, содержат динамические изображения в качестве результата расчета. В настоящее время в системе Matlab существует два способа получения анимированных изображений - анимация "на лету" с помощью изменения свойств соответствующих графических объектов [15-17], и анимация, получаемая с помощью создания соответствующей матрицы с последующим ее выводом с помощью процедуры `movie`. Ни один из этих способов в чистом виде не годится для использования в интернете. Последовательная пересылка и загрузка последовательности `html`-страниц для получения анимации, конечно, реализовать невозможно. Пересылка же матрицы для функции `movie` и ее последующее воспроизведение на стороне клиента требует наличия у клиента установленной системы `Matlab`, от чего мы отказались с самого начала. Поэтому нами было принято решение генерировать на стороне сервера средствами `Matlab` стандартный для интернет анимированный графический файл. Система `Matlab` до версии 6.0 таких средств в качестве встроенных функций не содержала. Поэтому после проведенного поиска мы остановились на программе `mpgwrite`, взятой нами из архива `MathWorks`, и пакета программ `makemovie`, разработанного A.Weigman (wiegmann@math.lbl.gov).

Программа `mpgwrite` использует матрицу, подготовленную с помощью стандартных

средств Matlab для использования с функцией movie. Но дело в том, что функция getframe, используемая для этих целей, не работает на стороне сервера и команда M(j)=getframe возвращает пустую матрицу. Пришлось создавать матрицу M с помощью функции print.

```
print -djpeg -r72 skin.jpg
OutImage = imread('skin.jpg');%
frame(j1).cdata=OutImage;
frame(j1).colormap=[];
```

После чего использовать матрицу frame в качестве аргумента функции mpgwrite для создания mpeg-файла.

При использовании функции makemovie в пакете поставки имеется функция makeframe, которая создает и записывает на диск промежуточные файлы с мгновенными изображениями. В этом пакете имеются варианты создания gif-файлов и mpeg-файлов, но вариант с созданием mpeg-файлов не работает, поэтому нами в настоящее время используются оба пакета один - для создания gif-файлов и второй - mpeg-файлов. Необходимость предоставлять пользователю возможность выбора типа генерируемого файла связана с тем, что просмотр этих файлов осуществляется разными средствами по разному. Gif-файл просматривается на экране средствами браузера (во всяком случае это справедливо для Microsoft Internet Explorer) и полученная анимация прокручивается заранее заданное число раз. Mpeg-файл просматривается с помощью проигрывателя Windows Media (на компьютерах, на которых установлена операционная система Windows) и этим просмотром можно управлять (останавливать, повторять, просматривать по кадрам). Кроме того, качество получаемых изображений различно в зависимости от типа выводимого рисунка.

4 Подготовка и представление 1-й части курса электродинамики в интернете

4.1 Содержание и структура конспекта

В настоящее время проф. И.А.Котельников и проф. В.И.Яковлев подготовили 1-ю часть конспекта лекций по классической электродинамике, которая преподается на 2 курсе физического факультета НГУ. Для отработки технологии преобразования LaTeX – XHTML были выбраны 9 разделов и 52 параграфа, названия которых приведены далее.

Предисловие

§ Как читать эту книгу

§ Что нужно знать заранее

§ Индукция или дедукция

1 Электростатика

§1 Закон Кулона

§2 Электрический заряд

§3 Электрическое поле

§4 Принцип суперпозиции

§5 Теорема Гаусса

§6 Дивергенция электрического поля

§7 Электрический потенциал

§8 Градиент потенциала

§9 Силовые линии и эквипотенциали

§10 Уравнение Пуассона

§11 Общее решение уравнения Пуассона

§12 Проводник в электрическом поле

§13 Уравнение Лапласа

§14 Стандартные задачи электростатики

§15 Энергия взаимодействия электрических зарядов

§16 Плотность энергии электрического поля

§17 Мультипольное разложение

§18 Емкость системы проводников. Емкостные и потенциальные коэффициенты

§19 Экспериментальная проверка закона Кулона

2 Электрический ток

§20 Плотность тока

§21 Закон сохранения заряда. Уравнение непрерывности

§22 Закон Ома. Проводимость металлов. Условие применимости закона Ома

§23 Закон Джоуля-Ленца

§24 Электрические цепи ЭДС. Законы Кирхгофа

§25 Граничные условия для полей при наличии тока

§26 Об аналогии и характерном отличии между полями в диэлектриках и проводниках

§27 Релаксация зарядов в среде

§28 Ток в газе и жидкости. Подвижности ионов и электронов. Несамостоятельный разряд в газе

§29 Несамостоятельный разряд между двумя параллельными электродами

§30 Ток в вакууме. «Закон 3/2»

3 Магнитостатика

§31 Сила Лоренца

§32 Закон Био–Савара

§33 Теорема Стокса

§34 Уравнения магнитостатики

§35 Геометрия магнитного поля

§36 Векторный потенциал

§37 Основное уравнение магнитостатики

§38 Магнитный диполь

§39 Момент сил и сила, действующая на магнитный диполь в магнитном поле

§40 Прецессия магнитного момента. Гиромагнитное отношение. Магнитный резонанс

`A Системы измерений`

§1 СГС

§2 Международная система единиц (СИ)

`B Криволинейные координаты`

§3 Классификация физических величины и тензоры

§4 Важнейшие формулы и теоремы векторного анализа

`C Ответы к задачам и упражнениям`

`D Из истории великих открытий`

§5 Электрический заряд

§6 Закон Кулона

`E Из жизни великих людей`

§7 Кавендиш

§8 Кулон

§9 Б. Франклин

`F Наследие`

§10 Г.Кавендиш: Об определение плотности Земли

§11 Ш.Кулон: О фундаментальном законе электростатики

§12 Б.Франклин: О статическом электричестве

Исходный текст был набран в LaTeXe и представлял собой довольно сложную структуру. В главном файле (см. Приложение Б), содержащем преамбулу и определения новых команд и окружений с помощью команды `\include{}` загружались требуемые файлы разделов, каждый из которых в свою очередь содержал общий для данного раздела текст и вызываемые с помощью команды `\input{}` отдельные файлы, содержащие текст соответствующего параграфа.

4.2 Особенности подготовленного текста и их реализация в LaTeX

Для написания учебника был разработан новый класс `TextBook.cls`, который базиру-

ется на классе `Book.cls`. Помимо этого было разработано окружение для печати задач и ответов в специальном формате. Кроме этого, как легко видеть из приведенного текста, был введен целый ряд переопределений в стиле написания математических выражений, определены новые команды и т.д. Следует сказать, что TeX4HT в основном хорошо справился со столь трудной задачей. Как упоминалось ранее, автору пришлось написать новый стилевой файл в связи с созданием нового класса документов. Кроме того, определенные проблемы вызвало переопределение оператора `\vec{}`. Приведенное в преамбуле переопределение, которое по замыслу авторов должно вместо стрелки над векторами выводить вектора полужирным шрифтом, не было реализовано. Сперва выяснилось, что TeX4HT не строит адекватные возможностям MathPlayer конструкции, которые могут реализовать команду `\mathbf{}`. После устранения этого недостатка оказалось, что ряд переопределений недостаточно сделать в преамбуле файла LaTeX, но их необходимо продублировать в конфигурационном файле `mat.cfg`. После проверки работоспособности такого подхода мы пришли к выводу, что с эстетической точки зрения на экране вектора смотрятся лучше со стрелкой вверху и оставили их в таком виде.

Как упоминалось ранее, все иллюстрации были подготовлены в формате EPS и TeX4HT в автоматическом режиме перевел их в формат GIF. При размещении рисунков в тексте использовались как стандартный пакет `graphicx`, так и пакет `floatflt`, позволяющий вставлять обтекаемые текстом рисунки.

4.3 Проблемы и пути их решения

Полученная совокупность HTML-файлов была размещена в интернете по адресу http://www.phys/nsu/ru/cherk/eldinfirst/webbook.html. Изучение полученных результатов показало, что при всей своей успешности система преобразования нуждается в доработке. Во-первых, получаемая в автоматическом режиме система навигации требует доработки как с эстетической точки зрения, так и в плане повышения наглядности. Для совершенствования этой системы необходимо опробовать имеющиеся в системе возможности вывода в режиме фрейма. Необходимо доработать внешний вид кнопок перехода, снабдить их системой подсказок и улучшить динамическое представление содержания текущего, просматриваемого раздела.

В процессе массового преобразования обнаружились некоторые достаточно изощренные конструкции LaTeX, которые компилятор MikTeX воспроизводил правильно, в то время как в формате HTML они выглядели неудовлетворительно. Это в первую очередь относится к использованию акцентов, интервалов между некоторыми элементами в сложных математических выражениях, сложной нумерации формул (окружение `subequations`) и ряд других.

5 Модели физических явлений в интернете

По описанной выше технологии к настоящему времени разработаны 8 задач по электродинамике: «Диаграмма направленности антенн», «Токи на поверхности кубического резонатора», «Диаграмма направленности излучения релятивистской частицы», «Движение релятивистской частицы в поле сильной электромагнитной волны», «Нестационарный скин-эффект», «Скин-эффект внутри проводящего полого цилиндра», «Излучение вектора Герца» и «Электростатическое поле точечных зарядов». С этими задачами можно познакомиться на сайтах http://matlab.tutornet.ru, http://phys.nsu.ru:8000.

При эксплуатации задач выяснилась следующая проблема — аварийный останов программы (деление на ноль, несовпадение размерности массивов и т.д.), т.е. те ошибки, которые при локальном исполнении приводят к останову исполнения задачи и переходу в режим командного окна, приводит к зависанию программы `matweb.exe` и необходимости его перезапуска. Кроме того, использованная система генерации графических файлов с записью на диск промежуточных картинок является достаточно сложной и относительно медленной. Выходом может быть разработка функции, генерирующей графические анимированные файлы в памяти. Возможно, появившаяся в версии 6.1 функция `avifile` поможет решить эту проблему.

В настоящее время все разработанные задачи могут работать также в локальном варианте, если на компьютера пользователя установлен `MatLab`. При работе в пределах Новосибирского научного центра время получения результата расчета составляет для разных задач от 30 секунд до 2 минут.

Заключение

Проведенный анализ литературы, как печатной так и сетевой, привел нас к выводу, что на сегодняшний день для представления научных материалов в интернете наиболее приемлемым является формат HTML (XHTML) с включенным в него языком разметки MathML для воспроизведения формул. Большой объем накопленных научных, учебных и учебно-методических материалов, подготовленных в системе LaTeX, выдвигает в качестве актуальной задача перевода этих материалов в формат HTML, с помощью которого эти материалы можно адекватно представлять в интернете. Для выполнения такого преобразования наиболее подходящим на сегодняшний день является пакет TeX4HT, который совместно с пакетом LaTeX и другими вспомогательными пакетами программ позволяют получить из исходного документа гипертекстовый документ в формате XHTML, который может просматриваться с помощью популярных браузеров с использованием дополнительного плагина MathPlayer.

Для отработки технологии первая часть конспекта лекций по классической электродинамике (9 разделов и 52 параграфа) была преобразована в формат XHTML с математическими формулами в разметке MathML. В процессе преобразования ряд проблем удалось решить — использование русского языка, преобразование файлов из кодировки UNICODE в win-1251, создание системы гипертекстовых ссылок, преобразование рисунков в формат `gif`. В то же самое время выявился ряд проблем, которые необходимо решать в процессе дальнейшей работы — создание более адекватной системы гиперссылок, разработка системы динамической генерации содержания, отработка связи разрабатываемых динамических моделей и текста, дальнейшее расширение представленных материалов в интернете.

Для представления интерактивных моделей физических явлений в интернете разработана технология создания таких моделей в системе MatLab и их перенос в интернет с помощью MatLab Web Server. Основная проблема создания подобных моделей — быстрая и качественная генерация динамических рисунков на стороне сервера и их передача по сети с приемлемой скоростью и качеством. Для решения этой задачи предполагается использование новых графических форматов, которые появились в последнее время в системе MatLab, а также оптимизация генерируемых иллюстраций по параметру время генерации-объем файлов. Планируется также при продолжении работы увеличить количество разработанных и представленных в интернете моделей.

Список использованных источников

1. Donald E. Knuth. *The TeXbook*. Addison-Wesley, Reading, Massachusetts, 1986. Revised to cover, 1991. Имеется перевод: Д.Е. Кнут. *Все про TeX*. Протвино: Издательство АО RDTEX, 1993.
2. Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesle Reading, Massachusetts, second edition, 1994/1985.
3. И. Котельников, П. Чеботаев. *Издательская система LaTeX2e*. Новосибирск: Сибирский хронограф, 1998.
4. Котельников И.А., Матвеенко А.Н., Черкасский В.С. *Разработка обучающих программ с использованием MATLAB Web Server* . Тезисы докладов Всероссийской научной конференции "Проектирование научных и инженерных приложений в среде MATLAB" (28-29 мая 2002 г.). М.: ИПУ РАН. 2002. 207 С.
5. M.Goosens, S. Rahtz. *The LaTeX Web Companion. Integrating TeX, HTML, and XML*. Addison-Wesley, Reading, Massachusetts, 1999. Имеется перевод: М.Гуссенс, С. Ратц. *Путеводитель по пакету LaTeX и его Web-приложениям.* М.: «Мир», 2001.
6. Практический курс Adobe Acrobat 3.0: Пер. с англ.- М.: КУбК-а, 1997.
7. Pierce, John R. *An Introduction to Information Theory. Symbols, Signals and Noise.. Revised edition of Symbols, Signals and Noise: the Nature and Process of Communication* (1961). Dover Publications Inc., New York, 1980
8. Cajori, Florian. *A History of Mathematical Notations*, vol. I & II. Open Court Publishing Co., La Salle Illinois, 1928 & 1929 republished Dover Publications Inc., New York, 1993.
9. Chaundy, T.W., P.R. Barrett, and C. Batey; *The Printing of Mathematics. Aids for authors and editors and rules for compositors and readers at the University Press*. Oxford. Oxford University Press, London, 1954.
10. Swanson, Ellen. *Mathematics into type: Copy editing and proofreading of mathematics for editorial assistants and authors*. Revised edition. American Mathematical Society, Providence, R.I., 1979.
11. Swanson, Ellen. *Mathematics into type: Updated Edition.* American Mathematical Society, Providence, R.I., 1999.
12. Higham, Nicholas J.; *Handbook of writing for the mathematical sciences*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1993.
13. Poppelier, N.A.F.M., E. van Herwijnen, and C.A. Rowley. *Standard DTD's and Scientific Publishing*, EPSIG News 5 (1992) #3, September 1992, 10-19.
14. Wall L., Christiansen T., Schwartz R.L. *Programming Perl.* 2nd edition. Sebastopol, Calif.: O'Reilly&Assoc. Inc. 1996.
15. Г.Л. Коткин, В.С. Черкасский, *Компьютерное моделирование физических процессов с исполь-*

зованием MATLAB. Новосиб. Ун-т. Новосибирск, 2001.

16. Н.Н. Мартынов, А.П. Иванов. *MATLAB 5.X. Вычисление, визуализация, программирование*. М.: КУДИЦ-ОБРАЗ, 2000.
17. Marchand Patrik. *Graphics and GUI with MATLAB*. Second edition. CRC Press LLC New York; Washington, 1999.

Приложение А

Скрипт UTF2WIN

Скритп UTF2WIN выполняет роль пост-процессора при конвертации документа LaTeX в HTML. Первоначально он был создан для замены символов, представляемых в виде уникодов, литерами русского языка. Позднее его функции были расширены. В частности, он анализирует таги в заголовочной части выходных файлов и производит необходимые изменения. Скрипт может выполнять также внутреннее форматирование выходных файлов, чтобы облегчить чтение кода HTML. Скритп UTF2WIN выполняет те функции, которые могли бы быть осуществлены постпроцессором TeX4HT, но при своей реалзиции потребовали бы более трудоемких изменений в стилевых и конфигурационных файлах TeX4HT. Например, чтобы заменить уникоды русскими буквами, сохранив уникоды для математических символов, необходимо было бы создать новые htшрифты (ht-шрифты, по сути являются таблицами перекодировки, которые используются системой TeX4HT, но создаются отдельно для каждого шрифты или семейства шрифтов). Скрипт UTF2WIN выполняет подобную замену с помощью регулярных выражений. Операция замены в скрипте UTF2WIN, написанном на языке Javascript, выполняется при помощи метода replace объекта типа String:

```
regEx = /&#x\w{4};/g;
txt = txt.replace(regEx, stripUTF)
```

где txt – переменная, содержащая целиком весь текст обрабатываемого файла. Метод replace передает часть строки, выделенную регулярным выражением regEx функции stripUTF:

```
function stripUTF(sMatchedString)
{
  return aEnc[sMatchedString] || sMatchedString;
}
```

Она возвращает замену выделенной части строки, если таковая имеется в массиве `aEnc`, либо саму выделенную строку, если для нее нет замены в указанном массиве. Ниже приведена небольшая часть определения массива `aEnc`:

```
var aEnc = []
aEnc["&#x0192;"]="&fnof;" //(Latin small f with hook)
aEnc["&#x0391;"]="&Alpha;"
aEnc["&#x0392;"]="&Beta;"
aEnc["&#x0393;"]="&Gamma;"
aEnc["&#x0394;"]="&Delta;"
aEnc["&#x0395;"]="&Epsilon;"
aEnc["&#x0396;"]="&Zeta;"

. . .
```

```
aEnc["&#x0040;"]="@"  //#COMMERCIAL AT
aEnc["&#x0041;"]="A"  //#LATIN CAPITAL LETTER A
aEnc["&#x0042;"]="B"  //#LATIN CAPITAL LETTER B
aEnc["&#x0043;"]="C"  //#LATIN CAPITAL LETTER C
aEnc["&#x0044;"]="D"  //#LATIN CAPITAL LETTER D
aEnc["&#x0045;"]="E"  //#LATIN CAPITAL LETTER E
aEnc["&#x0046;"]="F"  //#LATIN CAPITAL LETTER F
aEnc["&#x0047;"]="G"  //#LATIN CAPITAL LETTER G
aEnc["&#x0048;"]="H"  //#LATIN CAPITAL LETTER H
aEnc["&#x0049;"]="I"  //#LATIN CAPITAL LETTER I
. . .
```

Скритп UTF2WIN работает под управлением Windows Script Host, который имеется встроен во все версии операционной системы Windows, начиная с Windows 98. Корпорация Microsoft позиционирует Windows Script Host как замену bat-файлам. Скритп UTF2WIN без перекомпиляции может работать как режиме консольного, так и оконного приложения. Он принимает аргументы командной строки, а также поддерживает интерфейс drag-and-drop. Одновременно он может обрабатывать все файлы в одном или нескольких каталогах, а также в их дочерних каталогах. В простейшем случае достаточно в проводнике Windows захватить мышью нужный каталог и опустить его на иконку скрипта. Представление о других возможностях скрипта дает окно справки, приведенное на рис. 4, которое появляется при клике на иконке скрипта.

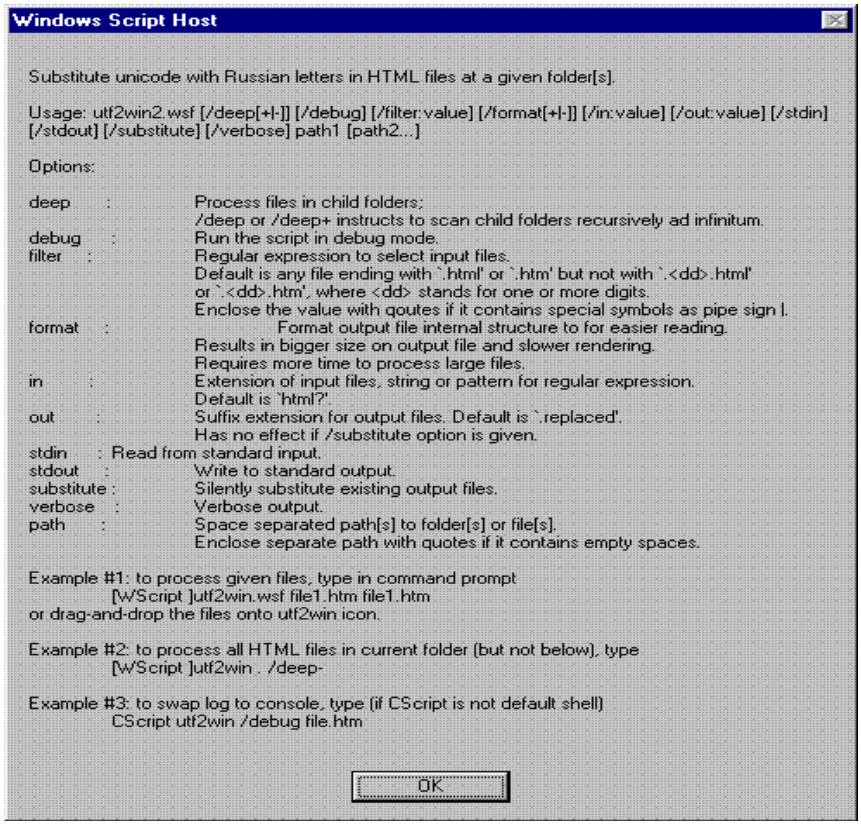

Рисунок 4- Окно справки скрипта UTF2WIN

В будущем мы планируем переписать bat-файлы, используемые пакетом TeX4HT в скрипты Windows Script Host, чтобы расширить функциональные возможности пакета.

Приложение Б

Основной файл учебника

```
\documentclass[10pt,openany,twoside,a5paper]{TextBook}[1998/10/25]
%% Here we load the russian package and change main font of
%% the document.
\usepackage[cp1251]{inputenc}
\usepackage[english,russian]{babel}
\usepackage{indentfirst}
%% Declare plain style of pages instead headings style used by default
\pagestyle{plain}
%% Final page setup to meet requirements of the NSU publishing office
\addtolength{\textheight}{\topmargin}\setlength{\topmargin}{0pt}
\addtolength{\marginparwidth}{-9pt}\addtolength{\textwidth}{9pt}
\addtolength{\textwidth}{42pt}\addtolength{\hoffset}{-21pt}% сдвиг на
50\% добавки
\addtolength{\textheight}{104pt}\addtolength{\voffset}{-52pt}% сдвиг
на 50\% добавки
%% Load graphics package
\usepackage{graphicx}\graphicspath{{eps/}}%\graphicspath{{eps/}{pcx/}}
\DeclareGraphicsRule{.gif}{bmp}{}{}% declare GIF filename extension
%% Redefine some math symbols to AmS-alphabets to provide
%% more plain style of notations.
    \let\|=\shortparallel
    \let\parallel=\shortparallel
    \let\le=\leqslant
    \let\leq=\leqslant
    \let\ge=\geqslant
    \let\geq=\geqslant
    \let\vec=\bm
%\usepackage{makeidx}\makeindex
%% Declare usefull trikcs
%% To typeset vectors, partial derivative, Hamiltonian
\newcommand{\parder}[2]{\ensuremath{\frac{\partial #1}{\partial #2}}}
\newcommand{\Ham}{\ensuremath{\mathcal{H}}}
%% To typeset energy and power few commands are available
\newcommand{\E}{\ensuremath{\mathcal{E}}} % Energy of a single parti-
cle
\newcommand{\W}{\ensuremath{W}}     % Energy per unit volume
\newcommand{\J}{\ensuremath{J}}     % Power (eg, radiated) by a parti-
cle
\newcommand{\Power}{\ensuremath{P}}     % Power per unit volume
\newcommand{\const}{\mathrm{const}}
\DeclareMathOperator{\sinc}{sinc}
\DeclareMathOperator{\sign}{sign}
\let\div=\undefined
\DeclareMathOperator{\div}{div} \DeclareMathOperator{\rot}{rot}
\DeclareMathOperator{\erf}{erf} \DeclareMathOperator{\grad}{grad}
\DeclareMathOperator{\diag}{diag} \DeclareMathOperator{\e}{e}
```

```
\newcommand{\dif}{\mathrm{d}}
%% Declare more flexible hyphenation
%% and prohibit wrong hyphenation of the word "объем"
\tolerance500
%\hyphenation{объем объема объеме объем-чик объем-чи-ка объем-чи-ке}
%% ------------------------------------------------------------
\includeonly{ch0, ch1,      ch3, ch4, % ch5, ch6, ch7, ch8, % For Tu-
torNet only
app1, app2, app3, app4, app5, app6, biblio, SeriesInfo}
%% ------------------------------------------------------------
%% Declare how to print Problems
\usepackage{answers}
\Newassociation{solution}{Solution}{ans}
\renewcommand{\solutionextension}{ans}
%% However, we are using even more sophisticated method:
%IAK>
\newcounter{problem}[section] % for the number of exercises in the
current chapter
%\renewcommand{\theproblem}{\thesection.\arabic{problem}}
\makeatletter\renewcommand{\theproblem}{
  {\ifnum \c@section>\z@ \@arabic\c@section.\fi \@arabic\c@problem}
}\makeatother

\newcommand{\problemname}{Задача}
\newenvironment{problem}[1][\problemname]{\medbreak%
    \refstepcounter{problem}%
    \noindent\llap{$\blacktriangleright$\kern.15em}% triangle in mar-
gin
    \begin{itshape}%
    %\fontfamily{tdy}\selectfont %Academy fonts inside the environment
    \fontfamily{mbk}\selectfont %Mobotype Book Manual
    \textbf{#1~\theproblem}\par\nobreak\noindent
    }{\end{itshape}\medbreak}%
%% ------------------------------------------------------------
%% The \Newassociation{solution}{Solution}{ans} command creates
%% a solution and Solution environments. Firts one must be used in
sourse text
%% to accept text to be written into ans han\dif{l}e. Wriiten text is
wrapped by
%% Solution env. We redefine the latter as we want it to have optional
argument.
%%
\newcommand{\solutionname}{Решение}
\renewenvironment{Solution}[1]{\par\noindent
\begin{upshape}\textbf{#1} \ignorespaces}{\end{upshape}\par\medbreak}
%% ------------------------------------------------------------
%% Declare how to print Solution if it should not go here-and-now.
%% For here-and-now solution we define starred version of the solution
environment.
%%
\newenvironment{solution*}[1][\solutionname:]{\par\noindent
\begin{upshape}\textbf{#1} \ignorespaces}{\end{upshape}\par}
%% ------------------------------------------------------------
```

```
%% That's all. Note the use of \summation command.
%% This is to count figures and problems. Note also that
%% the last included file does not contain things to count.
% ------------------------------------------------------------
\begin{document}
\frontmatter
\include{frontCover}
\include{ch0}% Предисловие
\mainmatter
\include{ch1}% Электростатика
\include{ch3}% Электрический ток
\include{ch4}% Магнитостатика
\appendix
\include{app1}% Системы единиц
\include{app2}% Криволинейные координаты
\include{app3}% Ответы к задачам
\include{app4}% Из истории великих открытий
\include{app5}% Из жизни великих людей
\include{app6}% Наследие
\backmatter
\include{biblio}% Список литературы
\end{document}
```