

Пономарева Ирина Сергеевна,
Зеленухина Виктория Андреевна,
Тарасевич Юрий Юрьевич

РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ MATLAB WEB SERVER



ВВЕДЕНИЕ

Признанным лидером в среде научных исследований является многофункциональный пакет MATLAB. Обширные библиотеки стандартных функций (toolbox) облегчают решение широкого круга задач. MATLAB Web Server (MWS) представляет возможность разрабатывать приложения в режиме удаленного доступа, используя стандартные компоненты MATLAB.

Применение веб-технологий при разработке программных средств учебного назначения позволяет полностью или частично решить несколько важных задач.

1. Учебные компьютерные программы, основанные на веб-технологиях, более доступны для массового использования.

2. Существенно упрощается задача сопровождения программного обеспечения. При обнаружении ошибок в программе, распространяемой на CD, или в случае создания обновленной версии программы необходимо заменять все экземпляры программы. Обновление программы, основанной на веб-технологиях, происходит незаметно для пользователей.

3. Проблема охраны авторских прав существенно упрощается.

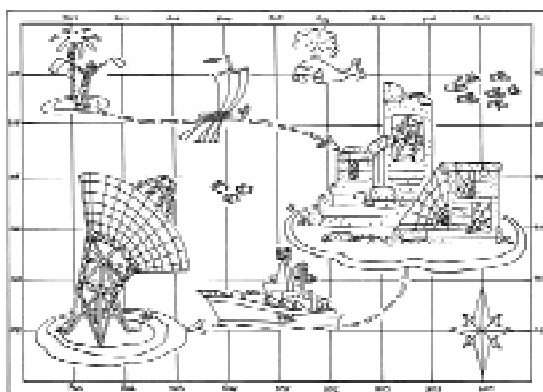


КАК РАБОТАЕТ MWS?

Подробно с принципом работы MWS можно познакомиться по публикациям [1; 2; 3], поэтому мы ограничимся только кратким обзором ключевых положений.

Работа MWS зависит от организации сети TCP/IP для передачи данных между клиентской системой и MATLAB. В упрощенном варианте во всем процессе участвуют два компьютера. Один из них – удаленный компьютер, за которым работает некоторый пользователь. Этот компьютер должен иметь выход в Internet. Пользователь взаимодействует с сервером посредством web-браузера, например, Microsoft Internet Explorer, и при этом ему совершенно не нужны знания о MATLAB. На сервере, в свою очередь, установлен MATLAB Web Server и размещено web-приложение.

В нашем случае используется более сложная организация сети, в которой взаи-



...организация сети... удаленный компьютер, сервер, содержащий веб-приложение, и компьютер, на котором находится MATLAB и MWS.

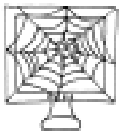
модействуют удаленный компьютер, сервер, содержащий web-приложение, и компьютер, на котором находится MATLAB и MWS (рисунок 1).



ЧТО НЕОБХОДИМО ДЛЯ РАБОТЫ С MWS?

Для работы web-приложения, помимо комплекса MATLAB и пакета MWS, требуется установленный web-сервер, который организует их работу на компьютере. Это может быть, например, Apache, свободно распространяемый в сети Internet (<http://www.apache.org>).

От разработчика приложения требуется знание программирования на MATLAB и базовые знания в области html.



КАКОВА СТРУКТУРА WEB-ПРИЛОЖЕНИЯ И ПРИНЦИП ЕГО РАБОТЫ?

Все приложение будет состоять из нескольких файлов разных типов. Поэтому целесообразно для собственного удобства разложить их по каталогам.

Итак, пусть создаваемое приложение располагается в каталоге *Model*. В этом каталоге желательно организовать еще три подкаталога. Первый подкаталог назовем *Htmlfiles*. В нем будут храниться только html-файлы, отображающие интерфейс web-приложения. Второй подкаталог *Mfiles* будет предназначаться только для m-файлов. Они производят все вычисления. И, наконец, третий – для временных файлов, создаваемых в качестве результатов в процес-

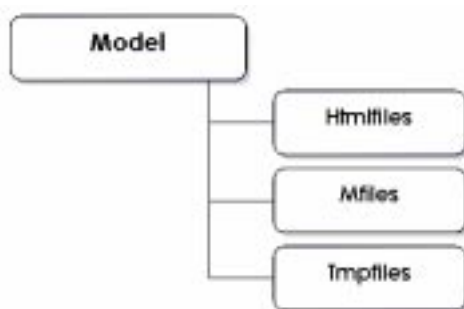


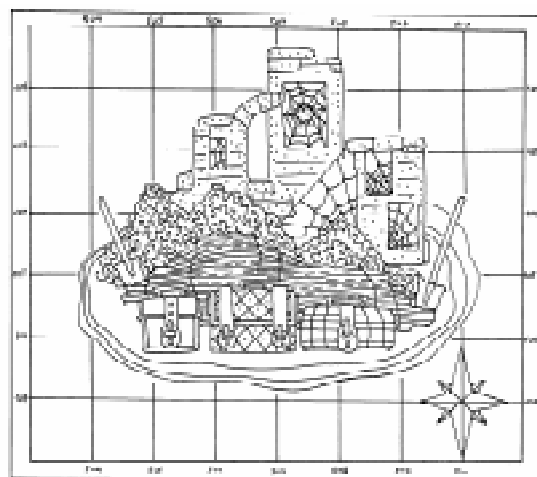
Рисунок 2.



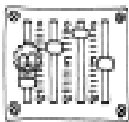
Рисунок 1.

се работы удаленного пользователя с моделью, – назовем *Tmpfiles* (рисунок 2).

Такое количество создаваемых файлов связано с функционированием приложения, принцип которого заключается в следующем: пользователь с удаленного компьютера через web-браузер обращается к модели, то есть к html-файлу из каталога *Htmlfiles*. Он вводит необходимые параметры и делает запрос на выполнение расчетов. Подгружается MWS, вызывая m-файл из каталога *Mfiles*. M-файл считывает данные с html-страницы, производит вычисления и отображает результаты в виде отдельных файлов, которые помещает в каталог *Tmpfiles*. При этом на удаленном компьютере html-страница обновляется и отображает полученные результаты, считывая их из каталога временных файлов *Tmpfiles* (рисунок 3).



Все приложение будет состоять из нескольких файлов разных типов.



НЕКОТОРЫЕ ОСОБЕННОСТИ

Для правильной работы MWS необходимо произвести несколько настроек.

Во-первых, необходимо установить взаимосвязь с MWS. Для этого в конфигурационном файле *matweb.conf*, сопровождающем программу *matweb.exe*, прописывается путь к упомянутому выше каталогу *Mfiles*. Сначала указывается в квадратных скобках имя *m*-файла, в котором впоследствии будут выполняться все вычисления, а затем прописывается путь к каталогу, содержащему этот файл:

```
[pend]
mlserver = localhost
mldir = [имя домена]/model/mfiles
```

В нашем случае прописан путь к файлу *pend.m* находящемуся в каталоге *Mfiles*.

Во-вторых, прежде чем перейти к созданию приложения, необходимо настроить на web-сервере конфигурационный файл, служащий для управления временными файлами.

Если в качестве web-сервера выступает Apache, то конфигурационным файлом будет файл *httpd.conf*. В этом файле необходимо для каталога *Icons* (он создается при установке Apache) прописать имя созданного нами каталога *Tmpfiles*, поскольку мы намеренно хотим, чтобы все временные файлы хранились именно в нем. Для

этого необходимо имеющуюся в файле строку отредактировать, прописав путь к каталогу временных файлов *Tmpfiles*:

```
Alias /icons/ "C:/MATLAB701/toolbox/
webservice/wsdemos/model/tmpfiles"
```

А теперь, когда произведены все соответствующие настройки, можно приступить к созданию приложения для MWS.



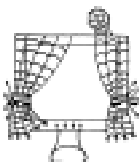
СОЗДАНИЕ WEB-ПРИЛОЖЕНИЯ

МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

В качестве примера рассмотрим модель больших колебаний маятника с затуханием, описываемую дифференциальным уравнением в безразмерном виде:

$$x'' + 2\beta x' + \sin(x) = 0$$

Здесь *x* – координата, β – декремент затухания. Необходимо изучить поведение модели, зависящей от единственного параметра β и начальных условий (угла и скорости).



ИНТЕРФЕЙС WEB-ПРИЛОЖЕНИЯ

Создание Internet-версии данной модели условно можно разбить на несколько этапов. На первом этапе продумывается интерфейс web-приложения.

При работе с локальной версией модели у пользователя имеется возможность (рисунок 4):

- производить динамический ввод начальных условий с помощью мыши;
- изменять параметр уравнения;
- получать фазовый портрет;
- получать график зависимости координаты от времени;
- проводить имитационное моделирование с помощью Simulink;
- получать теоретические сведения и справку по работе с моделью.

Хотелось бы, чтобы Internet-версия модели не теряла всех этих возможностей. Таким образом, нашей целью было создание простого и в то же время удобного ин-

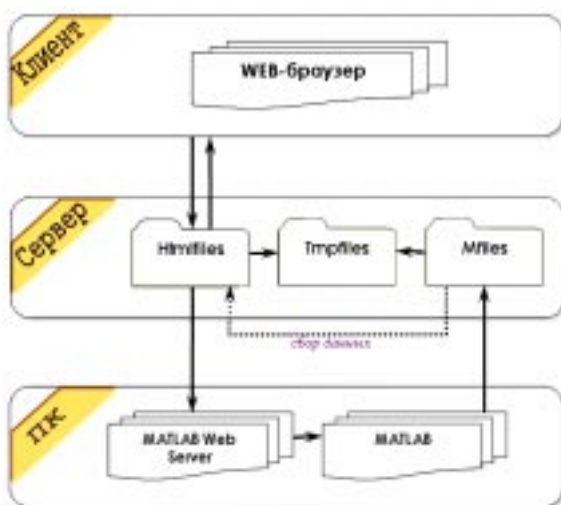


Рисунок 3.

терфейса, который был бы понятен любому пользователю, ограничиваясь средствами html, но максимально приближенного к интерфейсу локальной версии модели.

В итоге интерфейс включил несколько элементов управления: окно ввода для единственного параметра дифференциального уравнения β , два поля ввода для задания начальных условий «угол» и «скорость», два переключателя «Очистка» и «Дорисовка» и кнопку «Нарисовать». Помимо этого, страницу снабдили ссылками на источник теоретического материала и справку, которая была бы в помощь пользователю при работе со страницей. В поля ввода изначально введены некоторые значения (рисунок 5).

Ниже на странице предполагается расположить два графических изображения, которые бы отображали результат работы с моделью при введенных начальных условиях и параметре уравнения. Это будут координатные оси, на одной из которых изображается фазовый портрет, а на второй график зависимости одной из координат от времени.

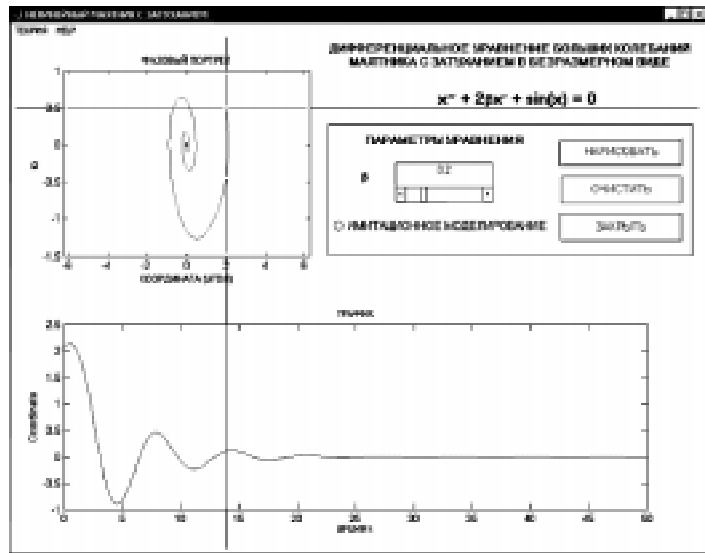


Рисунок 4. Интерфейс локального приложения.

Элементы «очистка» и «дорисовка» созданы для желаемого отображения фазового портрета. Понятно, что при каждом новом вводе начальных условий и параметра уравнения модели пользователь будет получать новые результаты, то есть новую фазовую траекторию и новый график. Для того чтобы следить за изменением поведения модели в зависимости от новых параметров, необходимо выбрать переключатель «Дорисовка». Это означает, что в процессе построения все новых и новых фазовых тра-

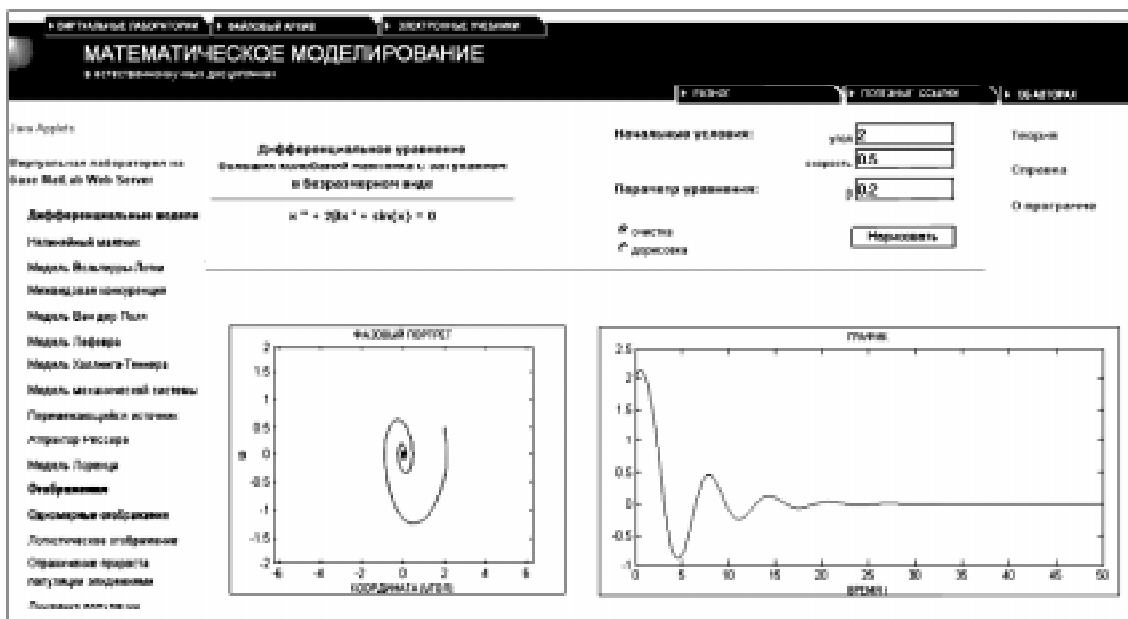


Рисунок 5. Интерфейс web-приложения.

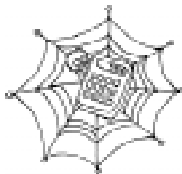
екторий будет происходить их дорисовка на уже имеющиеся. В случае выбора переключателя «Очистка» фазовый портрет будет обновляться, а на экран будет выводиться лишь последняя фазовая траектория.



HTML-ФАЙЛЫ

Определившись с интерфейсом web-приложения, переходим к созданию html-файла. Как уже отмечалось выше, он должен располагаться в каталоге *Htmlfiles*. Назовем его *pendulum_out.html* (листинг 1).

Через специальные символы \$ задается свойство *value*. Это необходимо для того, чтобы html-страница и m-файл могли обмениваться данными для соответствующих элементов управления.



M-ФАЙЛ

Следующий этап – это создание m-файла, в котором будут происходить все расчеты, строиться фазовый портрет и гра-

фик. Общий принцип работы m-файла следующий:

- считывает данные, введенные удаленным пользователем в форме;
- проводит вычисления;
- результаты вычислений отображает на двух координатных осях;
- на основе координатных осей создает два отдельных графических файла в каталоге *Tmpfiles*.

Для написания программы используются стандартные команды MATLAB (листинг 2).



ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Мы говорили об удобстве использования модели не случайно. В локальной версии модели имеется возможность динамического ввода начальных условий с помощью мыши посредством команды *ginput*. Html такой возможности не дает. Поэтому возникает необходимость создания полей ввода, в которых пользователь вручную за-

Листинг 1.

```
<html><body>
//Указываем путь к программе matweb.exe, чтобы обеспечить связь с MATLAB
<form action="http:// [имя домена]/
cgi-bin/matweb.exe" method="POST" name="form1">
//Указываем имя файла m-файла, в котором будут производиться расчеты
<input type="hidden" name="mlmfile" value="pend">
//Создаем поля ввода для начальных условий и параметра уравнения
угол <input type="text" name="corner" value=$corner$>
скорость <input type="text" name="v" value=$v$>
параметр  $\beta$  <input type="text" name="beta" value=$beta$>
//Создаем взаимосвязанные переключатели, задав им одинаковые имена. Первому присвоили значение равное 1, а второму 0 (эти значения будут необходимы при проверке условия в m-файле для сохранения положения точки при обновлении страницы)
<input type="radio" name="rad" value=0 $Radio1$>очистка
<input type="radio" name="rad" value=1 $Radio2$>дорисовка
//Создаем кнопку
<input type="submit" value="Нарисовать">
//Создаем два графических изображения


</form>
</body></html>
```

Листинг 2.

```

%Определяем имя функции
function rs = pend(h)
%Определяем глобальную переменную, отвечающую за параметр дифференциального уравнения, описывающего исследуемую модель
global beta
%Создаем дополнительную переменную для создания уникальных имен графических файлов
mlid = h.mlid;
%Делаем рабочим каталог Htmlfiles
cd('C:/MATLAB701/toolbox/websserver/wsdemos/model/htmlfiles');
%Проверяем в структуре строки ввода наличие всех параметров, от которых зависит построение модели
if (isfield(h, 'x')) & (isfield(h, 'y')) & (isfield(h, 'beta')) & (isfield(h, 'rad'))
%Если все переменные заданы, то соответствующим дескрипторам присваиваются их значения
h.x = str2double(h.x);
h.y = str2double(h.y);
beta = str2double(h.beta); h.beta = beta;
rb = str2double(h.rad);
%Если хотя бы одна переменная не задана или задана неверно, то присваиваем значения переменных по умолчанию
else
h.x = 0.5;
h.y = 2;
beta = 0.2; h.beta = beta;
rb = 0;
end
%Решаем дифференциальное уравнение
options = odeset('OutputFcn','','RelTol',1e-4);
[t,y] = ode45(@nldpe,[0 50],[h.corner h.v],options);
%Делаем рабочим каталог временных файлов Tmpfiles
cd('C:/MATLAB701/toolbox/websserver/wsdemos/model/tmpfiles');
%Уничтожаем старые рисунки, которым больше часа в целях избежания переполнения каталога
wscleanup('ml*pendulum1.jpeg', 1);
wscleanup('ml*pendulum2.jpeg', 1);
%Проверяем на наличие в каталоге Tmpfiles фигуры, на которой будет строиться новый фазовый портрет
m = exist('pendout.fig');
%Если таковая отсутствует, то создаем ее под именем pendout.fig, делая невидимой
if m == 0
f = figure('name','pendout','visible','off');
%Если такой файл существует, то его открываем, не отображая на экране
else
f = open ('pendout.fig');
set(f,'Visible', 'off')
end
%Проверяем, какой из переключателей выбран пользователем по переменной rb, которая может принимать (в нашем случае) значение 0 или 1, в зависимости от положения точки. В случае если выбран переключатель «очистка», то ему присваиваем свойство 'checked', что означает «выбранный», а для второго переключателя это свойство отменяется. Здесь же происходит очистка осей командой hold off. Это означает, что на осях будет строиться лишь

```


Продолжение листинга 2.

последняя фазовая траектория. И наоборот, если выбран переключатель «дорисовка», то именно ему (Radio2) присваивается свойство 'checked', а команда *hold on* позволит дорисовывать на оси новые фазовые траектории.

```

if rb= ==0
    h.Radio1 = 'checked';
    h.Radio2 = '';
    hold off
    cla
else
    h.Radio2 = 'checked';
    h.Radio1 = '';
    hold on
end
%Строим новую фазовую траекторию. Ее цвет генерируется произвольно, что облегчает
визуальное восприятие картинки
    plot(y(:,1),y(:,2), 'Color',rand(3,1));
%Определяем постоянные границы отображения координатных осей
    set(gca, 'XLimMode', 'manual', 'XLim', [-2*pi,2*pi], 'YLim', [-2,2]);
%Создаем заголовок и подписи к осям
    title('Фазовый портрет');
    xlabel('Координата (угол)');
    ylabel('upsilon', 'FontName', 'Symbol',);
%Определяем размер отображения осей
    set(gcf, 'PaperPosition', [0 0 10 8]);
%Обновляем фигуру
    drawnow
%Сохраняем текущие оси как output.fig
    saveas(gcf, 'output', 'fig');
%Создаем новую переменную GraphFileName, которая соответствует уникальному имени с
помощью переменной mlid
    h.GraphFileName = sprintf('%spendulum1.jpeg',mlid);
%На основе фигуры f создаем jpeg-файл с уникальным именем, заложенным в переменную
GraphFileName
    wsprintjpeg(f,h.GraphFileName);
%Закрываем фигуру f
    close(f)
%Открываем невидимое графическое окно для второго графика
    f1 = figure('visible','off');
%Строим график зависимости координаты от времени с заголовком и подписями к осям
    plot(t,y(:,1));
    title('График');
    ylabel('Угол');
    xlabel('Время');
%Задаем размер изображения
    set(gcf, 'PaperPosition', [0 0 18 8])
%Обновляем фигуру
    drawnow
%Создаем новую переменную GraphFileName1, которая соответствует уникальному имени с
помощью переменной mlid
    h.GraphName1 = sprintf('%spendulum2.jpeg',mlid);

```

Продолжение листинга 2.

```
% На основе фигуры f1 создаем jpeg-файл с уникальным именем, заложенным в переменную GraphFileName1
    wsprintjpeg(f1,h.GraphFileName1);
%Закрываем фигуру f1
    close(f1)
%Вновь делаем рабочим каталог Htmlfiles
    cd('C:/MATLAB701/toolbox/webserver/wsdemos/model/htmlfiles')
%Вызываем изменение шаблона выходного файла
    rs = htmlrep(h,'pendulum_out.html');
%Описываем функцию для дифференциального уравнения модели
    function dydt = nldpe(t,y)
    global beta
    dydt = zeros(2,1);
    dydt(1) = y(2);
    dydt(2) = -2*beta*y(2)-sin(y(1));
```

дает интересующие его начальные условия. Однако с помощью JavaScript возможно очень просто реализовать способ динамического ввода начальных данных. Для этого в коде html-страницы потребуется преобразовать строку вывода графического изображения фазового портрета, а также добавить простой скрипт.

```

<script language="JavaScript"><!--
function coord()
{
//Считываем координаты мыши относительно
картинки в пикселях
    corner0 = window.event.offsetX;
    v0 = window.event.offsetY;
//Перемасштабируем их соответственно координатным осям
    n1 = document.form1.img1.width/16.15;
    n2 = document.form1.img1.height/4.9;
    corner1 = corner0-
document.form1.img1.width/2-6)/n1;
    v1 = (document.form1.img1.height/2-v0-5.5)/
n2;
//Отображаем в поля ввода
    document.form1.corner.value = corner1;
    document.form1.v.value = v1;
}
--></script>
```

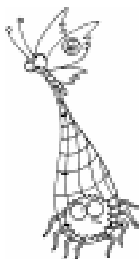
При первоначальной загрузке web-приложения страница должна содержать два графических изображения: фазовый портрет и график. Согласно рассматриваемой модели, кривые, отображаемые на них, зависят от декремента затухания β , начальной скорости v и угла отклонения от положения равновесия $corner$. В случае если удаленному пользователю уже известны эти переменные, ему предоставляется возможность уже при первоначальной загрузке модели получить результат, соответствующий заданным им параметрам. Для этого достаточно в строке запроса задать специальную команду:

```
http://[умя домена]/cgi-bin/
matweb.exe?mlmfile=pend&corner=2&v=
0.5&beta=0.2.
```

В противном случае разработчик web-приложения должен позаботиться о значении переменных по умолчанию.

Отметим еще один момент при создании модели: проблему некорректных входных данных. Пользователь может ввести недопустимые значения параметра. Для устранения этого недостатка к скрипту добавляется еще одна функция, которая ограничивает значение параметра β в пределах от 0 до 1. В случае ввода пользователем недопустимого значения выводится сообщение об ошибке. При этом строка ввода кнопки «Нарисовать» преобразуется добавлением функции *onMousedown="value_beta()*».


```
function value_beta()
{
item1=document.form1.beta.value;
if (item1<0)
{alert('Введено недопустимое значение.
Введите значение от 0 до 1');
document.form1.beta.value=0;}
if(item1>1)
{alert('Введено недопустимое значение.
Введите значение от 0 до 1');
document.form1.beta.value=1;}
}
```



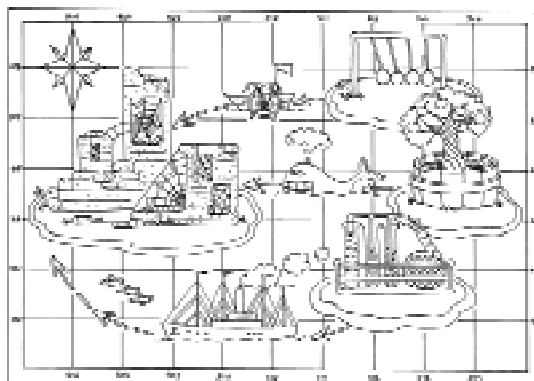
ЗАКЛЮЧЕНИЕ

В настоящее время во многих учебных заведениях происходит реформация средств организации учебного процесса, интенсивное внедрение дистанционного образования. Для физико-математических специальностей организация дистанционного обучения является достаточно сложной. Отсюда понятна актуальность разработки электронных учебников и практикумов.

Такие математические пакеты как Mathcad, Maple, MATLAB позволяют создавать web-приложения, ориентированные, в

том числе, и на сложные вычисления. Используя их широкие возможности, преподаватели могут значительно облегчить организацию своей работы, особенно преподаватели физико-математических специальностей.

С использованием MWS нами был создан целый комплекс классических моделей из физики, химии, биологии, объединенных в виртуальный лабораторный практикум. Все эти модели вы можете исследовать на сайтах <http://mathmod.aspu.ru> и <http://mathmod.exponenta.ru>.



... создан целый комплекс классических моделей из физики, химии, биологии, объединенных в виртуальный лабораторный практикум.

Литература

1. MATLAB Web Server. The MathWorks, Inc. 2002.
2. Котельников И.А., Черкасский В.С. MATLAB Web Server: Вычисления в Интернете // Exponenta Pro. Математика в приложениях. 2004. № 1(4). С. 4–11.
3. Потемкин В.Г. Инструментальные средства MATLAB 5.x. М.: Диалог-МИФИ, 2000. 241 с.

Пономарева Ирина Сергеевна,
ассистент кафедры прикладной
математики и информатики
Астраханского государственного
университета (АГУ),

Зелепухина Виктория Андреевна,
ведущий программист отдела
Internet-технологий АГУ,

Тарасевич Юрий Юрьевич,
кандидат физико-математических
наук, доцент кафедры
теоретической физики и методики
преподавания физики АГУ.



Наши авторы, 2005.
Our authors, 2005.