

Министерство образования Республики Беларусь  
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

---

Республиканский институт инновационных технологий

В.В. Сидорик  
С.Г. Погирницкая

**ПРАКТИКУМ ПО МОДЕЛИРОВАНИЮ  
В СРЕДЕ MATLAB**

Учебно-методическое пособие  
для студентов и слушателей системы повышения  
квалификации и переподготовки

*Учебное электронное издание*

Минск 2012

УДК 004.42

**Авторы:**

*В.В.Сидорик, С.Г. Погирницкая*

**Рецензенты:**

*И.З. Джилавдари*, профессор кафедры информационно-измерительной техники и технологий БНТУ, доктор технических наук, профессор;

*И.В. Никифоров*, доцент кафедры вычислительной математики БГУ, кандидат физико-математических наук, профессор, доцент

В учебно-методическом пособии описаны этапы математического моделирования, приведены сведения о среде моделирования MATLAB. Предлагается большое количество практических задач из различных областей знаний. В рассматриваемых примерах дается описание проблемы, строятся физическая, математическая и алгоритмическая модели, приводится программный код. Для самостоятельной работы предлагается широкая программа исследований.

Издание предназначено для слушателей системы повышения квалификации и переподготовки кадров, а также для студентов и преподавателей.

Белорусский национальный технический университет  
пр-т Независимости, 65, г. Минск, Республика Беларусь  
Тел.(017) 292-77-52 факс (017) 292-91-37  
Регистрационный № БНТУ/РИИТ – 87.2012

© БНТУ, 2012

© Сидорик В.В., Погирницкая С.Г., 2012

© Сидорик В.В., компьютерный дизайн, 2012

## Оглавление

Введение.....	4
ОПИСАНИЕ МАТЛАВ.....	11
ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ.....	23
РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ С ИСПОЛЬЗОВАНИЕМ ВСТРОЕННЫХ ФУНКЦИЙ.....	27
ДВИЖЕНИЕ ЖИДКОСТИ ПО ТРУБОПРОВОДУ.....	33
ЗАПОЛНЕНИЕ РЕЗЕРВУАРА.....	38
РАЗВИТИЕ БИОЛОГИЧЕСКИХ ПОПУЛЯЦИЙ.....	42
ЭПИДЕМИЯ ГРИППА.....	48
ЭКСТРЕМАЛЬНЫЕ ПРЫЖКИ.....	57
ПРЫЖКИ С ПАРАШЮТОМ.....	62
МИРОВОЙ РЕКОРД ПО ПРЫЖКАМ В ДЛИНУ.....	72
МЕХАНИЧЕСКИЕ КОЛЕБАНИЯ.....	76
СПЯЩИЙ ПОЛИЦЕЙСКИЙ.....	81
КОЛЕБАТЕЛЬНЫЙ КОНТУР.....	88
КОЛЕБАНИЯ МЕМБРАНЫ.....	93
РАСПРОСТРАНЕНИЕ ТЕПЛА.....	96
МОДЕЛИРОВАНИЕ В ПОДСИСТЕМЕ SIMULINK.....	101
Список использованных источников.....	114

## Введение

В настоящее время складываются и интенсивно развиваются основы новой методологии научных исследований – математического моделирования на основе компьютерного эксперимента. Сущность этой методологии заключается в замене реального объекта его моделью и исследовании с использованием современного математического аппарата на базе современного программного обеспечения и компьютеров. Математическое моделирование охватывает все новые и новые сферы деятельности, включая не только традиционные приложения к сложным техническим системам, но и различные области экономики, управления и гуманитарные знания.

Рост математической культуры приводит к тому, что изучение общих теоретических положений и методов вычислений не вызывает серьезных затруднений у специалистов в различных предметных областях. Вместе с тем, как показывает практика, знание предметной области и необходимого математического аппарата вовсе недостаточно для решения прикладных задач. Необходимо приобрести навыки в переводе решаемой проблемы на математический язык. В этом и состоит искусство математического моделирования. Проблема усугубляется различием в преподавании математики как фундаментальной науки для специалистов (и не только математики) и использованием математики специалистами в своей профессиональной деятельности.

Современные технические, экономические, экологические, социальные и другие системы в современном обществе настолько сложны, что не поддаются исследованию с нужной степенью точности и полнотой обычными теоретическими методами. Проведение натурального эксперимента в ряде случаев дорого, занимает длительное время, опасно или вообще невозможно. В этой связи математическое моделирование позволяет проводить компьютерные эксперименты с моделями и выступает здесь как новый метод познания, проектирования и конструирования. Математическое моделирование выступает в настоящее время как основа любого современного крупномасштабного технического, экономического, социального и любых других проектов.

В основе методологии математического моделирования лежат понятия предметной модели и математической модели. Моделирование может быть представлено в виде замкнутого циклического процесса, представленного на рис 1.

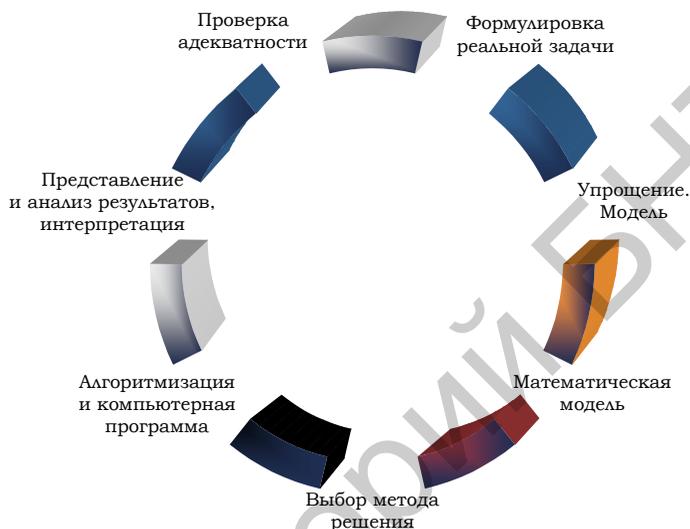


Рис. 1. Этапы математического моделирования

### ***Формулировка реальной задачи – проблема***

В реальной практике математического моделирования исходным пунктом является некоторая реальная эмпирическая ситуация, выдвигающая требования ее описания и ответа на поставленные вопросы (что происходит, как, почему...), а также прогнозирование поведения системы в других условиях.

На этом этапе необходимо установить, в чем именно заключается задача. Ответ на этот вопрос не всегда однозначен и очевиден. Здесь также, как правило, необходимо выявить наиболее существенные особенности и свойства системы, указать на те, которые несущественны. Сложность в том, что заранее нет никаких критериев, позволяющих классифицировать признаки на существенные и несущественные. Приходится опираться на опыт, логические умозаключения, интуи-

цию. Исключение из рассмотрения несущественных признаков приводит к упрощению реальной проблемы, которую и называют моделью. Всякая реальная проблема, упрощаемая в конкретной предметной области, может быть определена как предметная модель (экономическая, биологическая, физическая, химическая, социальная...).

### **Предметная модель**

Слово модель происходит от латинского *modus* (копия, образ, очертание). Моделирование – это процесс замещения некоторого объекта – оригинала (реальная система) другим замещающим его объектом (более простым) – моделью. Предполагается, что изучение модели обеспечивает изучение оригинала.

Целью моделирования является получение, обработка и представление информации, которая может быть перенесена на реальную систему с заданной точностью.

Степень подобия, адекватности модели и системы определяется соответствием множества свойств модели  $M$  множеству свойств системы  $S$  (рис. 2).

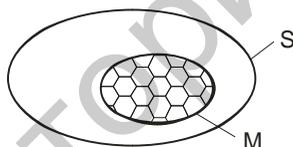


Рис. 2. Реальная система и модель

### **Математическая модель**

Построение математической модели заключается в определении связей между различными свойствами, процессами и явлениями с помощью математического аппарата, позволяющего качественно и количественно с помощью логико-математических конструкций описывать основные свойства системы, параметры, связи.

Математическая модель – это формальная система, представляющая собой конечное собрание символов и совершенно строгих правил оперирования этими символами в совокупности с интерпретацией свойств определенного объекта некоторыми отношениями, символами или константами. Математи-

ческая модель – это совокупность математических объектов и отношений между ними, которая адекватно отражает существенные для данной системы свойства.

Создание математической модели позволяет перевести изучение системы на язык математических выражений, позволяющих количественно описывать поведение и свойства системы на абстрактном математическом языке.

Формы и способы представления математических моделей, а также их решения зависят от ряда факторов. В этой связи математические модели могут быть детерминированные, стохастические, непрерывные, дискретные, статические, динамические и др.

Математическая модель как совокупность математических выражений (уравнений), описывающих поведение системы, представляет не что иное, как уравнение движения системы (в общем случае).

Достоинство математических моделей заключается в том, что они абстрактны, точны и однозначно передают информацию. Точность математической модели позволяет решать уравнения и сопоставлять получаемые результаты с экспериментальными данными.

Идеализация реальной задачи и описание ее на математическом языке имеют смысл лишь тогда, когда математическая модель поддается решению и анализу. Именно по этой причине появились блоки без трения, невесомые и нерастяжимые нити, абсолютно твердые или упругие тела, хищники, питающиеся только одним видом жертв, и другие идеализированные представления об окружающем мире.

### **Вычислительный алгоритм**

Прежде чем применять вычислительные средства решения математических моделей проводится их предварительное исследование методами прикладной математики. Анализируются вопросы множественности решений, устойчивости. Рассматриваются особенности математических моделей. Как результат выполняется построение вычислительного алгоритма, позволяющего решить рассматриваемую математическую модель с необходимой точностью.

## *Программное обеспечение компьютерного эксперимента*

Программное обеспечение компьютерного эксперимента включает различные среды программирования (Pascal, Delphi, Fortran, Visual Basic), системы компьютерной математики (MathCad, MATLAB), специализированные программные продукты, ориентированные на решение конкретных прикладных задач в конкретной области, и др.

В последние годы все большую популярность приобретают системы компьютерной математики – программы, возникшие на стыке математики и информатики и предназначенные для решения и визуализации математических задач. Система Matlab (MATrix LABoratory – матричная лаборатория) занимает среди них особое место как по степени универсальности, так и по сложности. По обилию функций и скорости вычислений MATLAB превосходит большинство подобных систем и является бесспорным лидером в области численных расчетов и математического моделирования различных систем и устройств.

Система MATLAB может быть использована при изучении дисциплины «Математическое моделирование» как для студентов, так и в системе повышения квалификации и переподготовки. Эта система может быть также полезна при изучении дисциплин, где требуется обработка экспериментальных результатов.

Опыт показал, что система MATLAB может быть выбрана и эффективно использована в качестве среды решения реальных инженерных задач. С одной стороны, этот пакет достаточно прост для изучения, имеет удобный для пользователя интерфейс, средства помощи и диагностики ошибок, задачи и решения выражаются в форме, близкой к математической постановке. С другой стороны, он имеет мощный математический аппарат и широкие возможности для представления результатов.

В MATLAB реализованы численные алгоритмы решения задач линейной алгебры, уравнений, нахождения значений определенных интегралов, аппроксимации, решения систем или отдельных дифференциальных уравнений. MATLAB обладает хорошо развитыми возможностями визуализации двумерных и трехмерных данных, позволяет создавать собственный графический интерфейс пользователя.

Работа в среде MATLAB может осуществляться в двух режимах:

- в командном режиме (режиме калькулятора или режиме непосредственных вычислений), т.е. вычисления производятся непосредственно после набора очередного оператора;
- путем вызова программы, записанной на языке MATLAB.

Простой встроенный язык программирования позволяет легко создавать собственные алгоритмы.

Матричная форма представления операций делает программирование в среде MATLAB очень лаконичным, что снижает трудоемкость работ и повышает скорость вычислений.

Существует большое число пакетов прикладных программ, расширяющих и дополняющих возможности MATLAB, предназначенных для решения задач из той или иной области науки или техники.

Для того чтобы решать задачи моделирования и эффективно использовать пакет MATLAB, необходима базовая математическая подготовка в объеме технических вузов. В частности, необходимо иметь представление о следующих понятиях:

- элементы матричной математики (понятия скаляр, вектор, матрица, операции над ними);
- функции одной и нескольких переменных;
- графическое представление функций и поверхностей;
- алгебраические уравнения и их системы;
- интегрирование;
- дифференциальные уравнения (порядок уравнения, обыкновенные дифференциальные уравнения и уравнения в частных производных);
- аппроксимация экспериментальных данных, метод наименьших квадратов.

Кроме того, требуется знание основ алгоритмизации (знание любого алгоритмического языка и базовых алгоритмических конструкций).

### ***Компьютерный эксперимент***

Проведение вычислительного эксперимента предполагает решение математической модели, которая описывает предметную модель, в выбранной программной среде с помощью компьютера (рис.3).



Рис. 3. Компьютерный эксперимент

# ОПИСАНИЕ MATLAB

## Окна системы MATLAB

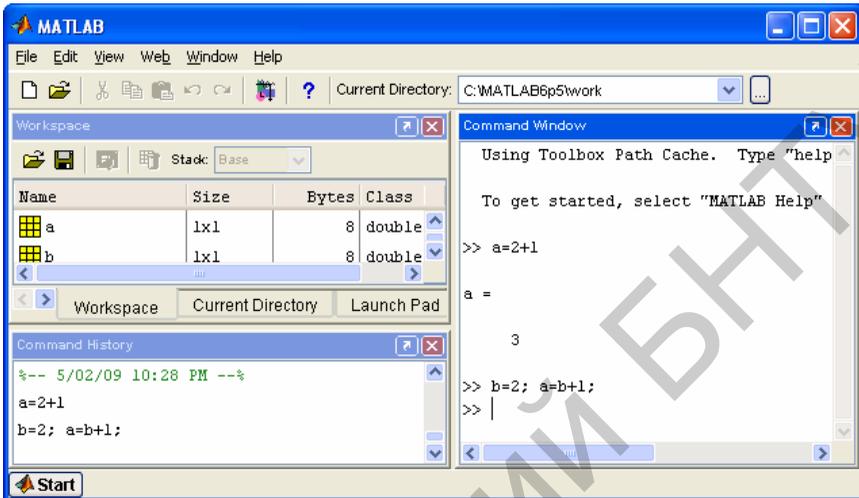


Рис. 4. Основное окно системы MATLAB

После запуска программы MATLAB на экране появляется основное окно системы (рис. 4), составленное из окон следующего назначения:

- Command Window** предназначено для ввода команд и вывода результатов вычислений и сообщений об ошибках;
- Command History** в окне выводится список выполнявшихся ранее команд и время загрузок системы;
- Workspace** обеспечивает просмотр переменных и внесение в них изменений;
- Current Directory** предназначено для просмотра и установки текущей папки;
- Launch Pad** обеспечивает быстрый доступ к установленным расширениям MATLAB, справочной системе и документации.

Вычисления в режиме диалога выполняются в командном окне Command Window.

- >> приглашение для ввода команды. Команда выполняется после нажатия клавиши Enter. Точка с запятой в конце выражения подавляет вывод результата.

Программы создаются в окне Редактора, вызываемого командой File> New> M-File (рис. 5).

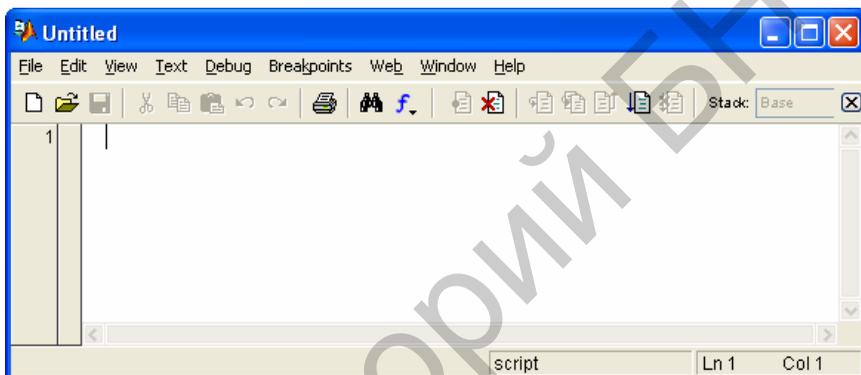


Рис. 5. Окно Редактора MATLAB

### Константы

Константы (постоянные величины) могут быть числовыми и символьными:

- 0.9093 дробное число. Десятичная точка отделяет целую часть числа от дробной;
- 1.6021e-20 число  $1,6021 \cdot 10^{-20}$  в экспоненциальной форме (без пробелов);
- 3i или 3j мнимое число;
- 2+3i комплексное число;
- pi число  $\pi = 3,14159265\dots$ ;
- 'текст' символьная константа;
- %текст текстовый комментарий.

## Переменные

Имена переменных состоят из букв английского алфавита, цифр и символа подчеркивания. Первым символом в имени должна быть буква. MATLAB идентифицирует число символов не более 31. MATLAB различает прописные и строчные буквы.

$a=3$  присвоение переменной  $a$  значения 3;  
 $A=2$  присвоение другой переменной  $A$  значения 2.

## Вектора и матрицы

По умолчанию все числовые переменные в MATLAB считаются матрицами. Матрицей принято называть прямоугольный массив чисел или выражений. Скалярная величина есть матрица порядка  $1 \times 1$ . Вектором является одномерный массив размера  $1 \times n$  (вектор-строка) или  $m \times 1$  (вектор-столбец).

$v=[4 \ 0]$  ввод вектор-строки  $v = (4 \ 0)$ .

$v=[4,0]$  Элементы отделяются пробелами или запятыми.

$u=[5;6]$  ввод вектор-столбца  $u = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$ .

Элементы отделяются точкой с запятой.

$M=[1 \ 2; \ 3 \ 8]$  ввод матрицы  $M = \begin{pmatrix} 1 & 2 \\ 3 & 8 \end{pmatrix}$ .

$p1=M(1,2)$  переменной  $p1$  присваивается значение элемента матрицы  $M$  из 1-й строки 2-го столбца, равное 2.

$p2=M(3)$  значение переменной  $p2$  равно 2. Матрица  $M$  рассматривается как длинный вектор, сформированный из столбцов исходной матрицы.

$r=M(:,1)$  создается вектор-столбец  $r = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$ , элементы которого равны элементам 1-го столбца матрицы  $M$ .

$q=M(2,:)$  создается вектор-строка  $q = (3 \ 8)$ , элементы которого равны элементам 2-й строки матрицы  $M$ .

$A=M$  создается матрица  $A$ , равная матрице  $M$ .

`A(2, :)=[]` удаление 2-й строки матрицы  $A = \begin{pmatrix} 1 & 2 \end{pmatrix}$ .  
`C=[M u]` объединение матрицы  $M$  и вектора  $u$  в одну матрицу  $C = \begin{pmatrix} 1 & 2 & 5 \\ 3 & 8 & 6 \end{pmatrix}$ .  
`x=0:2:6` формирование вектора как диапазона чисел от 0 до 6 с шагом 2  $x = (0 \ 2 \ 4 \ 6)$ .  
`t=1:5` формирование вектор-строки с элементами от 1 до 5 с шагом 1  $t = (1 \ 2 \ 3 \ 4 \ 5)$ .  
`z=linspace(1,9,5)` формирование вектор-строки из 5 элементов, значения которой равномерно распределены на отрезке от 1 до 9.  
`E=eye(3)` задание единичной матрицы размера  $3 \times 3$ .  
`Z=zeros(3)` задание нулевой матрицы размера  $3 \times 3$ .  
`N=ones(3,4)` задание матрицы из единиц размера  $3 \times 4$ .  
`size(N)` размер матрицы  $N$  в виде вектор-строки  $[3 \ 4]$  (3 строки, 4 столбца).  
`zeros(size(N))` нулевая матрица такого же размера, как  $N$ .  
`length(z)` длина вектора  $z$ .

### Функции

MATLAB предоставляет большое количество встроенных математических функций. Имя функции записывается строчными буквами, аргументы указываются в круглых скобках через запятую. В качестве аргументов можно использовать выражения и другие функции.

`abs(a)` абсолютная величина  $(|a|)$ ;  
`sign(a)` знак числа  $a$ ;  
`sqrt(x)` корень квадратный из  $x$  ( $\sqrt{x}$ );  
`exp(x)` экспонента ( $e^x$ );  
`log(x)` натуральный логарифм ( $\ln x$ );  
`log10(x)` десятичный логарифм ( $\lg x$ );  
`sin(x), cos(x)` синус ( $\sin x$ ), косинус ( $\cos x$ );  
`asin(x), acos(x)` арксинус ( $\arcsin x$ ), арккосинус;  
`tan(x), cot(x)` тангенс ( $\operatorname{tg} x$ ), котангенс ( $\operatorname{ctg} x$ );  
`atan(x), acot(x)` арктангенс ( $\operatorname{arctg} x$ ), арккотангенс.

Большинство функций может работать с аргументами в виде векторов и матриц, вычисляя значения для каждого их элемента. Данная операция называется векторизацией и обеспечивает упрощение записи операций, производимых одновременно над всеми элементами векторов и матриц, и существенное повышение скорости их выполнения.

```
>>x=0:0.1:0.5
>>y=sin(x)
y=
    0.0998    0.1987    0.2955    0.3894    0.4794
```

### Справочная система MATLAB

help	список разделов справки;
help раздел	список команд указанного раздела;
help имя_команды	описание команды с указанным именем;
help ops	операторы и специальные символы;
help elfun	элементарные математические функции;
help demos	список примеров;
type имя_m-файла	просмотр текста m.файла.

### Арифметические операции

В MATLAB реализованы два типа арифметических операций: операции над матрицами в соответствии с правилами линейной алгебры и поэлементные операции. Чтобы их различить, поэлементным операциям предшествует точка.

u=[1;3] заданы два вектор-столбца и матрица

v=[2;4]  $u = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, v = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, M = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}.$

M+2 ко всем элементам матрицы  $M$  прибавляется число 2.

u+v складываются соответствующие элементы векторов.

M\*u матричное умножение.

v\*u ошибка. При матричном умножении число столбцов первого сомножителя должно быть равно числу строк второго.

v.\*u поэлементное умножение.

$M \cdot u$	ошибка. Массивы должны быть одинакового размера.
$M^2$	степень матрицы. Если показатель степени целое положительное число, то матрица перемножается сама на себя.
$M.^2$	поэлементное возведение в степень.
$M \setminus u$	левое деление. Равносильно $M^{-1} \cdot u$ .
$u ./ v$	поэлементное деление.
$M'$	транспонирование матрицы. Строки становятся столбцами.
$u'$	транспонирование. Вектор-столбец становится вектор-строкой.

### **Решение системы линейных уравнений**

Систему линейных уравнений

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

...

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

обычно кратко записывают в матричном виде

$$Ax = b$$

Решение системы линейных уравнений в MATLAB находят с помощью левого деления

$$x = A \setminus b.$$

### **Матричные вычисления**

По умолчанию действует следующее правило: вычисляется соответствующая операция для элементов столбцов и результат помещается в вектор-строку. Чтобы проделать вычисления построчно, можно вначале транспонировать исходную матрицу, а затем полученный в результате вектор-столбец превратить в строку. Кроме того, для многомерных массивов можно явно указать размерность, по которой будет действовать операция.

$\text{sum}(u)$	возвращается сумма элементов вектора $u$ .
$\text{sum}(M)$	возвращается вектор-строка, содержащая
или $\text{sum}(M, 1)$	сумму элементов по столбцам матрицы $M$ .

<code>sum(M, 2)</code>	возвращается вектор-столбец с суммой элементов массива $M$ по строкам.
<code>min(M)</code>	определение минимальных элементов столбцов матрицы $M$ .
<code>max(M)</code>	определение наибольших элементов столбцов матрицы $M$ .
<code>[C, I]=max(M)</code>	кроме максимальных значений возвращается вектор индексов этих элементов.
<code>prod(M)</code>	произведение элементов столбцов матрицы.
<code>mean(M)</code>	определение средних элементов матрицы.
<code>sort(M)</code>	сортировка элементов матрицы по возрастанию.

### **Полиномы в MATLAB**

`p=[1 0 -2 -5]` представление полинома  $p(x) = x^3 - 2x - 5$  с помощью вектор-строки, содержащей упорядоченные по убыванию степени коэффициенты полинома.

`polyval(p, x)` вычисляет значения многочлена  $p$  в точках  $x$ .

### **Построение графиков**

<code>x=0:pi/50:2*pi</code>	задание диапазона $0 \leq x \leq 2\pi$ .
<code>y=sin(x)</code>	вычисление функции.
<code>plot(x, y)</code>	создается кусочно-линейный график зависимости компонент вектора $y$ от $x$ .
<code>x2=0:pi/10:2*pi</code>	
<code>y2=sin(x2-0.25)</code>	строятся две кривые на одном графике.
<code>plot(x, y, 'k:', x2, y2, 'ro')</code>	строятся графики с заданными цветом, стилем линий и маркерами (рис. 6).

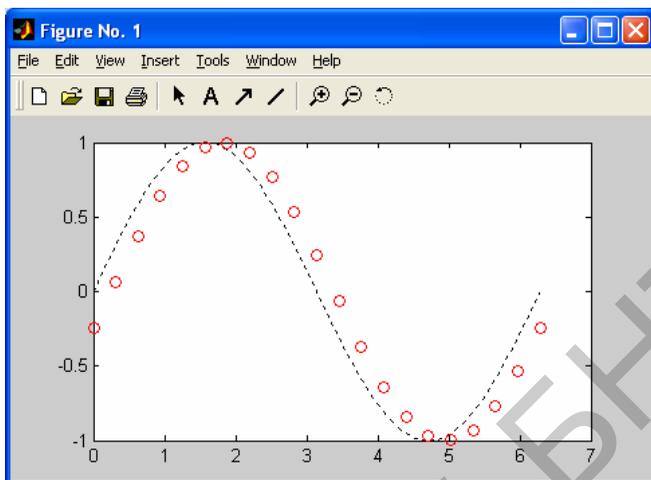


Рис. 6. Построение графиков с заданными свойствами линий

Цвет линий		Стиль линий		Тип маркера	
y	Желтый	-	Сплошная	o	Окружность
m	Фиолетовый	--	Штриховая	x	Крест
c	Голубой	-.	Штрих-пунктир	+	Плюс
r	Красный	:	Двойной пунктир	*	Звездочка
g	Зеленый			s	Квадрат
b	Синий			d	Ромб
w	Белый			v	Треугольник
k	Черный				

`fplot('sin(t)+0.5',[0 6])` строится график функции, заданной в символьном виде, на интервале от 0 до 6.

### Трехмерная графика

`plot3(x,y,z)` трехмерный аналог команды `plot`.  
`[X,Y]=meshgrid(x,y)` формируется прямоугольная сетка, заданная векторами  $x$  и  $y$ .  
`[X,Y]=meshgrid([-3:0.15:3]);`  $-3 \leq x, y \leq 3$

```
Z=X.^2+Y.^2;
```

```
mesh(X,Y,Z)
```

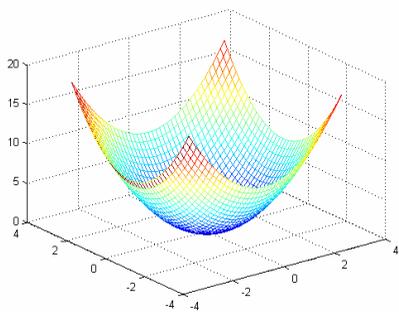
```
surf(X,Y,Z)
```

```
contour(X,Y,Z)
```

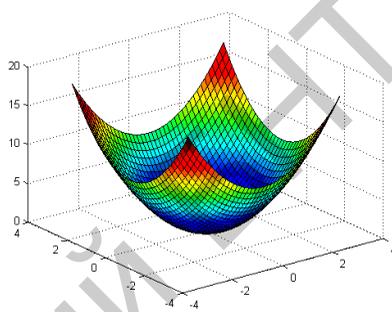
задание функции  $Z(X, Y)$  на сетке  
строится каркасная поверхность  
 $Z(X, Y)$  (рис. 7 а).

строится сплошная цветная поверхность (рис. 7 б).

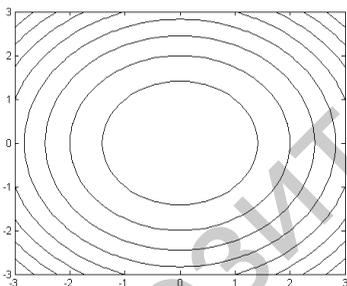
строится контурный график (рис. 7 в).



а)



б)



в)

Рис. 7. Построение трехмерных поверхностей

### **Управление выводом графика**

```
plot(x,y,x2,y2,'r-.')
```

```
title('График','FontName','Arial Cyr')
```

вывод заголовка заданным шрифтом.

```
xlabel('X')
```

маркировка оси  $x$ .

```
ylabel('Y')
```

маркировка оси  $y$ .

<code>text(5.2,4.5,'ТЕХТ')</code>	вывод текста в заданные координаты места графика.
<code>legend('y','y2')</code>	идентификация кривых (легенда).
<code>grid on</code>	нанесение координатной сетки (рис .8).

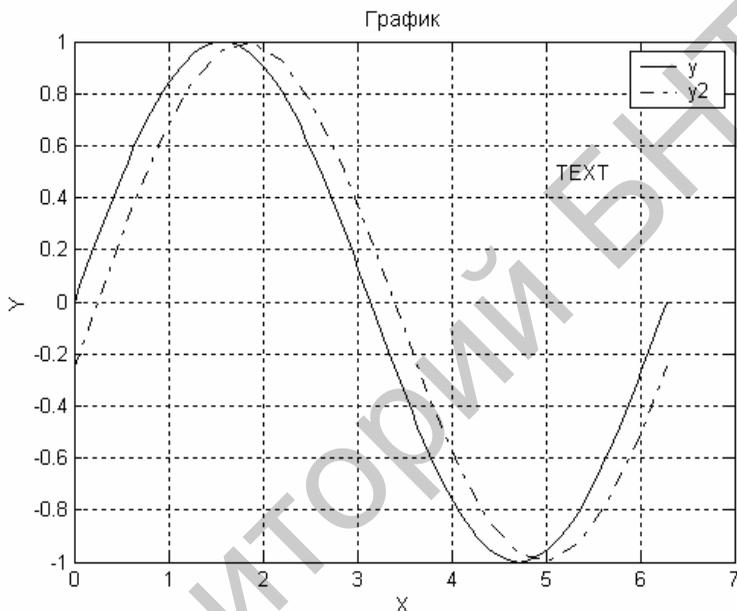


Рис. 8. Нанесение надписей на графике

<code>hold on</code>	продолжение вывода графиков в текущее окно.
<code>hold off</code>	отменяет режим продолжения.
<code>figure</code>	открывается новое графическое окно.
<code>subplot(m,n,p)</code>	разбивает графическое окно на $m \times n$ подокон, $p$ – номер текущего подокна, нумерация идет по строкам (рис. 9).

```

t=0:pi/50:2*pi;
subplot(1,2,1); plot(t,sin(t))
title('Fig.1')
subplot(1,2,2); plot(t,t.^2)
title('Fig.2')

```

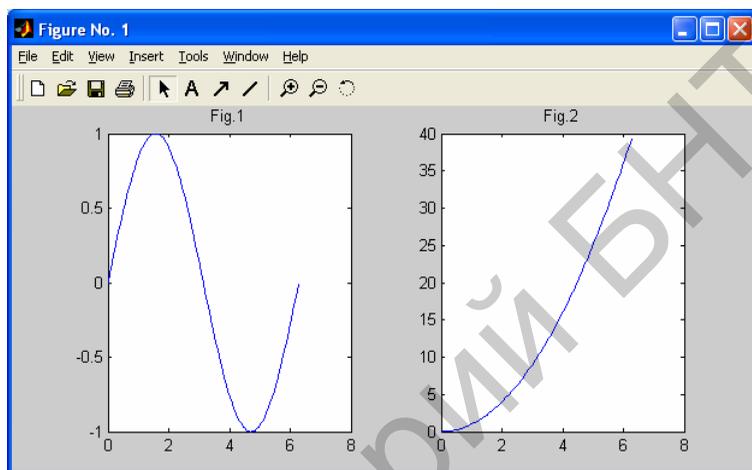


Рис. 9. Вывод графиков в одном графическом окне

### **Программирование в среде MATLAB**

Программы создаются в окне Редактора, вызываемого щелчком по кнопке New M-File на панели инструментов. Программы на языке программирования MATLAB сохраняются в виде текстовых файлов с расширением m (*m-файлы*).

%Комментарий  
операторы

*m-файл* сценарий. Работает с данными из рабочей области

```

function
[v1,v2,...]=fname(p1,p2..)
%Комментарий
операторы
var1=...
var2=...

```

*m-файл* функция *fname* содержит входные (*p1,p2,...*) и выходные (*v1,v2,...*) параметры и использует локальные переменные, доступные только в пределах данной функции.

```
global X Y Z
```

Объявление переменных X, Y, Z глобальными. Чтобы несколько программных модулей могли совместно использовать глобальную переменную, ее идентификатор должен быть объявлен как *global* во всех этих модулях.

### Управляющие конструкции языка программирования

```
s=0;  
for i=1:5  
    s=s+v(i)  
end
```

цикл с определенным числом повторений. вычисление суммы элементов вектора v.

```
x=[]; k=1;  
while k<5  
    x=[x k^3];  
    k=k+1;  
end
```

цикл с неопределенным числом повторений обеспечивает выполнение команд тела цикла, пока истинно проверяемое условие; формирование вектора x с элементами k<sup>3</sup> (k<5).

```
if x<0  
    y=0;  
elseif x>1  
    y=x;  
else  
    y=x.^2;  
end
```

условный оператор выполняет группу операторов, если логическое выражение истинно;

вычисление функции  $y = \begin{cases} 0, & x < 0 \\ x^2, & 0 \leq x \leq 1 \\ x, & x > 1 \end{cases}$

```
r=9; h=10; d=13  
flag=1  
switch flag  
case 1  
    S=pi*r^2  
case 2  
    S=h*d  
end
```

оператор переключения;

вычисление площади фигуры в зависимости от ее типа (1 – круг, 2 - прямоугольник).

## ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ

Дифференциальные уравнения очень часто встречаются при построении моделей, описывающих динамику объектов исследования. Изучение любого процесса или явления сводится к установлению зависимости между величинами, его характеризующими. Для сложных физических процессов, в которых определяющие величины могут существенно меняться в пространстве и времени, установить зависимость между этими величинами очень трудно. В этих случаях на помощь приходит метод математической физики. Суть его в том, что ограничивается промежуток времени, из всего пространства рассматривается лишь элементарный объем (отрезок). В основу вывода уравнений, описывающих физические явления, положены законы сохранения энергии и массы, импульса. В пределах элементарного объема и выбранного малого отрезка времени можно пренебречь изменением некоторых величин и существенно упростить зависимость.

Производной функции  $y = f(x)$  называется предел отношения приращения функции  $\Delta y$  к приращению аргумента  $\Delta x$  при

$\Delta x$ , стремящемся к нулю, т.е.  $y' = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} f(x)$ . Если  $z = f(x, y)$

функция двух переменных  $x$  и  $y$ , то, зафиксировав для  $y$  какое-либо значение, можно дифференцировать  $z$  по  $x$ . Полученная

производная  $\frac{\partial z}{\partial x}$  называется частной производной  $z$  по  $x$ .

В предельном случае при описании процесса, когда изменение независимой переменной (времени, координаты и т.п.) стремится к нулю, искомую зависимость можно записать в виде дифференциального уравнения.

Дифференциальным уравнением называется уравнение, в котором неизвестная функция входит под знаком производной или дифференциала. Наивысший порядок производной, входящей в дифференциальное уравнение, определяет порядок уравнения.

Если неизвестная функция, входящая в дифференциальное уравнение, зависит только от одной независимой переменной, то дифференциальное уравнение называется *обыкновенным*. Если неизвестная функция, входящая в дифференциальное уравнение, является функцией двух или большего числа неза-

висимых переменных, то дифференциальное уравнение называется *уравнением в частных производных*.

Процесс решения дифференциальных уравнений называется интегрированием. Решением дифференциального уравнения является всякая функция, при подстановке которой дифференциальное уравнение обращается в тождество. Общее решение дифференциального уравнения представляет собой семейство функций.

Для того чтобы полностью (однозначно) описать тот или иной физический процесс, необходимо кроме самого уравнения этого процесса задать еще дополнительные условия, называемые *краевыми условиями*: они подразделяются на начальные и граничные условия.

Интегрируя дифференциальные уравнения, можно получить зависимость между величинами для всей области интегрирования и всего рассматриваемого промежутка времени. Искомые функции могут быть найдены как в аналитическом виде, так и в виде таблицы значений.

### **Обыкновенные дифференциальные уравнения**

Обыкновенное дифференциальное уравнение  $n$ -го порядка в общем случае содержит независимую переменную, неизвестную функцию и ее производные или дифференциалы до  $n$ -го порядка включительно и может быть записано в следующем виде:

$$y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)}).$$

Наибольшее распространение имеют задачи Коши, в которых заданы начальные условия (начальное состояние процесса): при  $x = x_0$   $y(x_0) = y_0$ ,  $y'(x_0) = y_0'$ ... . Геометрически задача Коши для уравнения первого порядка состоит в том, чтобы из всего множества интегральных кривых, представляющих собой общее решение, выделить ту интегральную кривую, которая проходит через точку с координатами  $(x_0; y_0)$ .

Дифференциальные уравнения высших порядков решаются в основном сведением к системе уравнений первого порядка путем замены переменных:

$$y_1 = y', y_2 = y'' \text{ и т.д.}$$

При этом дифференциальное уравнение  $n$ -го порядка заменяется системой из  $n$  уравнений:

$$y' = y_1,$$

$$y_1' = y_2$$

.....

$$y_{n-1}' = f(x, y, y_1, y_2, \dots, y_{n-1})$$

Интегрируя дифференциальные уравнения, можно получить зависимость между величинами для всей области интегрирования и всего рассматриваемого промежутка времени.

### **Дифференциальные уравнения в частных производных**

Многие практически важные задачи гидродинамики, тепло- и массопереноса, теплопроводности, диффузии, теории упругости и других областей знаний описываются дифференциальными уравнениями с частными производными второго порядка. Они отражают изменение параметров объекта во времени и пространстве.

Тип уравнения, название	Физическое содержание
Параболический тип Уравнение теплопроводности $d \frac{\partial u}{\partial t} - \nabla \cdot (c \nabla u) + au = f$	Описывает распространение тепла, диффузию и другие процессы переноса
Гиперболический тип Волновое уравнение $d \frac{\partial^2 u}{\partial t^2} - \nabla \cdot (c \nabla u) + au = f$	Описывает процессы, связанные с механическими, электрическими, акустическими и другими видами колебаний
Эллиптический тип Уравнение Пуассона (при $f = 0$ – Лапласа) $- \nabla \cdot (c \nabla u) + au = f$	Описывает стационарное тепловое поле, потенциальное течение жидкости и другие физические явления, связанные с выходом на стационарный режим

Оператор Набла в декартовой системе координат имеет вид

$$\nabla = \frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k},$$

$$\operatorname{div} \vec{a} = \nabla \cdot \vec{a}, \quad \operatorname{grad} \varphi = \nabla \varphi,$$

Оператор Лапласа

$$\operatorname{div}(\operatorname{grad} \varphi) = \nabla \cdot (\nabla \varphi) = \nabla^2 \varphi = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = \Delta \varphi$$

Для того чтобы решить задачу, необходимо задать краевые условия, которые включают в себя:

- начальные условия – значение функции и ее производной в начальный момент времени;
- граничные условия:
  - условие Дирихле – условие, при котором на границе рассматриваемой области искомая функция принимает заданные значения;
  - условие Неймана называется условием, при котором на границе рассматриваемой области нормальная производная искомой функции должна принимать заданные значения;
  - смешанные условия – условия, при которых на границе рассматриваемой области задается линейная комбинация искомой функции и ее нормальной производной.

## РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ С ИСПОЛЬЗОВАНИЕМ ВСТРОЕННЫХ ФУНКЦИЙ

### Решение обыкновенных дифференциальных уравнений

При решении задач моделирования можно выбирать разные процедуры численного решения обыкновенных дифференциальных уравнений. В системе MATLAB имеются встроенные функции: `ode23`, `ode45`, `ode113`, `ode15s`, `ode23s`, `ode23t` и `ode23tb`. Функции с суффиксом `s` предназначены для решения так называемых систем жестких дифференциальных уравнений, а для всех остальных систем дифференциальных уравнений наиболее употребительной является функция `ode45`, реализующая алгоритм Рунге–Кутты 4-5-го порядка.

Эта функция наиболее удобна для первого, «пристрелочно-го», решения большинства задач. Она реализует одношаговый алгоритм для вычисления  $y(t_n)$ , где необходимо знание решения в одной предыдущей точке  $y(t_{n-1})$ .

#### Встроенная функция `ode45`

Функция `ode45` создана для решения задач, представляемых в общем виде дифференциальными уравнениями с начальными условиями вида

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$

Здесь  $t$  независимая переменная (время, координата, объем и др.) и  $y$  вектор зависимых переменных (температура, концентрация, давление и др.). Математически проблема точно описана, если правая часть уравнения выражена в виде функции  $f(t, y)$  и заданы начальные условия  $y = y_0$  в момент времени  $t_0$ .

Функция `ode45` имеет следующий синтаксис:

$$[t, y] = \text{ode45}('fname', tspan, y0, options).$$

**`fname`** имя функции (М-файл), используемой для представления правой части дифференциального уравнения в виде

$$function dydt = fname(t, y).$$

Выводимое выражение `dydt` должно быть вектором такой же размерности, что и  $y$ . Следует отметить, что независимая переменная  $t$  должна обя-

	зательно быть включена, даже если она отсутствует в явном виде.
<i>tspan</i>	2-элементный вектор, задающий интервал интегрирования ( $[t_0 \ t_f]$ ).
<i>y0</i>	вектор начальных условий для зависимой переменной.
<i>options</i>	дополнительный параметр, позволяющий управлять деталями процесса интегрирования. С помощью строки параметров <i>options</i> можно управлять, например, значениями допустимой относительной и абсолютной погрешности интегрирования. Этот параметр можно не указывать, если пользователя устраивают значения погрешностей, заданных по умолчанию, т.е. относительная погрешность интегрирования $1.0e-3$ , а абсолютная (по каждой из переменных состояния) – $1.0e-6$ . В противном случае, перед обращением к процедуре <code>ode45</code> следует указать значения погрешностей при помощи процедуры <code>odeset</code> .
<i>t</i>	значения независимой переменной (вектор-столбец), для которых вычисляются значения зависимой переменной <i>y</i> .
<i>y</i>	вычисленные значения зависимой переменной (массив). Каждый столбец <i>y</i> является отдельной зависимой переменной.

Функция `ode45` осуществляет интегрирование численным методом Рунге–Кутты 4-го порядка, а с помощью метода 5-го порядка контролирует относительные и абсолютные ошибки интегрирования на каждом шаге и изменяет величину шага интегрирования так, чтобы обеспечить заданные пределы ошибок интегрирования.

### **Общие правила определения функции правых частей**

При написании функции правых частей для каждой задачи необходимо руководствоваться следующими правилами:

1. Функция описания правых частей должна содержать не менее двух входных аргументов *t* и *y*, даже если какой-то из них не используется явно при вычислении правых частей.

2. Правые части системы, которые вычисляются функцией `odefun`, должны образовывать вектор-столбец.

3. Любые дополнительные параметры, которые необходимо передавать функции `odefun`, должны быть в конце списка параметров самой функции (после специального параметра *flag*) и в списке аргументов вызова решателя.

### **Интегрирование системы обыкновенных дифференциальных уравнений**

Пусть задана система обыкновенных дифференциальных уравнений

$$\frac{dy_1}{dt} = y_2 y_3$$

$$\frac{dy_2}{dt} = -y_1 y_3$$

$$\frac{dy_3}{dt} = -0.51 y_1 y_2$$

с начальными условиями  $y_1(0) = 0$ ,  $y_2(0) = 1$ ,  $y_3(0) = 1$ .

Для интегрирования данной системы уравнений необходимо создать *m*-файл, который является функцией переменных *t* и *y*. Текст файла

```
function dydt = myfunction(t,y)
dydt=zeros(3,1);
dydt(1)=y(2)*y(3);
dydt(2)=-y(1)*y(3);
dydt(3)=-0.51*y(1)*y(2);
```

Имя файла и функции должны совпадать. Файл надо сохранить с именем `myfunction`.

Для обращения к процедуре интегрирования необходимо создать файл с произвольным именем, например, `main`, в котором следует задать начальные условия и написать обращение к функции `ode45` с необходимыми параметрами:

```
tspan = [0 12]; % интервал интегрирования 0+12 с
y0 = [0 1 1]; % вектор начальных условий
options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5])
[t,y]=ode45('myfunction',[0 12],[0 1 1],options)
```

В этом примере абсолютная и относительная погрешность задается при помощи параметра `options` и команды `odeset` (относительная  $10^{-4}$ , абсолютная соответственно по каждой из переменных вектора состояний  $10^{-4}$ ,  $10^{-4}$ ,  $10^{-5}$ ). Время интегрирования зададим в интервале от 0 до 12, вектор начальных условий  $[0 \ 1 \ 1]$ .

Результатом интегрирования является матрица проинтегрированных значений фазовых переменных  $y$ , в которой каждый столбец соответствует одной из переменных состояния, а строка содержит значения переменных состояния, соответствующих определенному шагу интегрирования, т.е. значению вектора  $t$ .

Графически результаты интегрирования данной системы дифференциальных уравнений (рис. 10) могут быть представлены с помощью команды

```
plot(t,y(:,1), t,y(:,2), t,y(:,3))
```

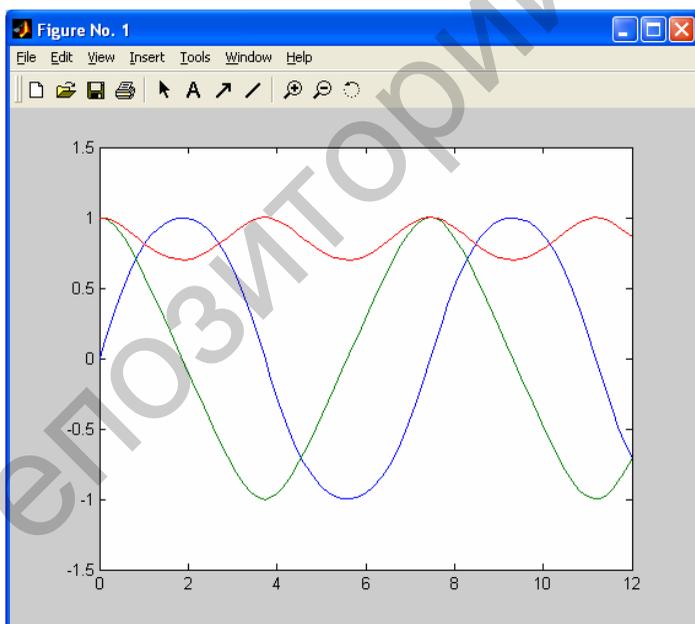


Рис. 10. Решение системы дифференциальных уравнений

## Решение уравнений в частных производных в Matlab

Для решения дифференциальных уравнений в частных производных, определенных на двумерных областях, предназначен пакет расширения Matlab PDE (Partial Differential Equations Toolbox). В пакете реализован метод конечных элементов.

Для решения уравнений параболического типа предназначена функция `parabolic`, гиперболического типа – `hyperbolic`, эллиптического типа – функция `asempde`.

Форматы вызова функций

```
u1=hyperbolic(u0,ut0,tlist,b,p,e,t,c,a,f,d)
u1=parabolic(u0,tlist,b,p,e,t,c,a,f,d)
u=asempde(b,p,e,t,c,a,f)
```

Здесь

<code>u0</code>	значения функции и ее производной в начальный момент времени.
<code>ut0</code>	
<code>b</code>	параметр отвечает за постановку граничных условий. Он может быть матрицей или именем <i>m</i> -файла. Граничные условия могут зависеть от времени <i>t</i> .
<code>p, e, t</code>	описывают сетку расчетной области.
<code>c, a, d, и f</code>	коэффициенты уравнений, они могут также зависеть от времени <i>t</i> .
<code>u1</code>	каждая строка матрицы содержит решение в узлах сетки, координаты которых задаются соответствующим столбцом в <i>p</i> . Каждый столбец в <i>u1</i> является решением в момент времени, который определяется соответствующей координатой вектора <i>tlist</i> .

Графический интерфейс модуля PDE (рис. 11) позволяет выполнить следующие действия:

- определение геометрии области решения (режим рисования);
- задание граничных условий (режим граничных условий);
- выбор коэффициентов, определяющих задачу (режим PDE);
- дискретизация конечных элементов (режим сетки);

- задание начальных условий и решение уравнений (режим решения);
- обработка решения (режим графика).

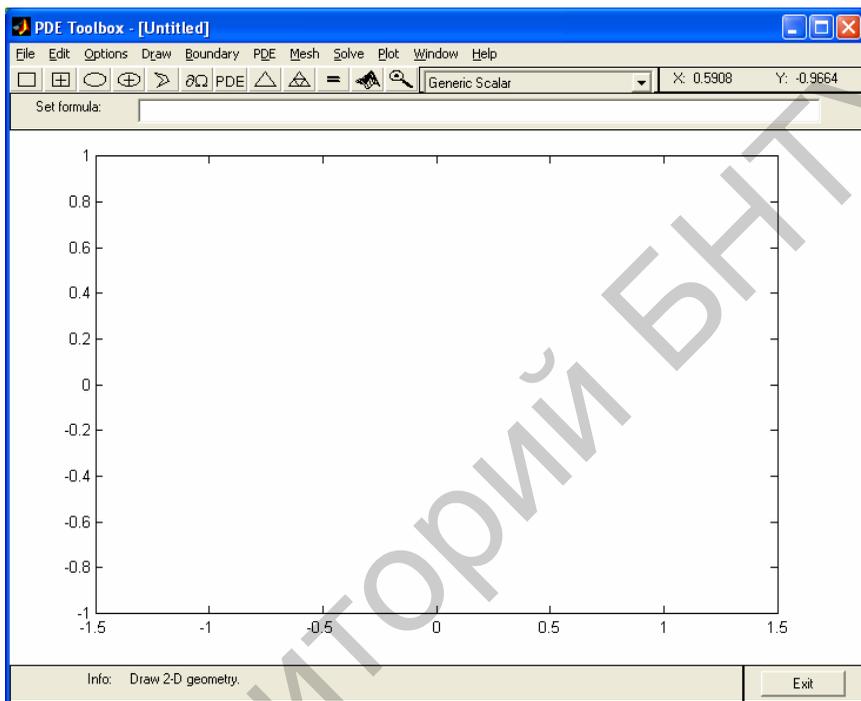


Рис. 11. Графический интерфейс модуля PDE

Подробное описание пакета PDE приводится в задаче о [распространении тепла](#).

## ДВИЖЕНИЕ ЖИДКОСТИ ПО ТРУБОПРОВОДУ

### Формулировка проблемы

Задача важна с точки зрения энергосбережения, т.к. для прокачки жидкостей нагнетательное устройство должно затрачивать энергию. Потери энергии зависят от расходов жидкости и потерь давления в трубопроводе. Жидкость движется по сложному трубопроводу, составленному из двух ветвей. Исследовать, как будут меняться потери давления и распределение расходов в ветвях трубопровода при изменении их параметров.

### Физическая модель

Имеется трубопровод, состоящий из двух параллельно соединенных трубопроводов с круглым сечением (рис.12). Известен общий расход жидкости  $Q$ . Жидкость однородна, несжимаема, плотность жидкости  $\rho$  постоянная. Движение жидкости установившееся.

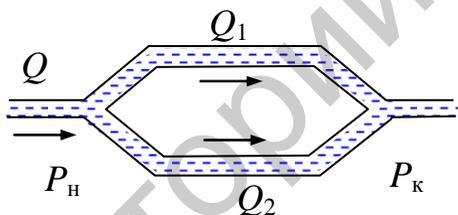


Рис. 12. Схема трубопровода

### Параметры моделирования

Длины ветвей трубопровода  $l_1, l_2$ .

Диаметры труб  $d_1, d_2$ .

Коэффициенты гидравлического трения  $\lambda_1, \lambda_2$ .

Коэффициенты местных сопротивлений  $\xi_{1i}, \xi_{2i}$ .

### Математическая модель

Как следует из гидравлики, потери давления в каждой ветви трубопровода определяются выражением

$$\Delta P_1 = c_1 Q_1^2, \quad \Delta P_2 = c_2 Q_2^2,$$

$$\text{где } c_1 = \left( \lambda_1 \frac{l_1}{d_1} + \sum_{i=1}^{n_1} \xi_{1i} \right) \frac{8\rho}{\pi^2 d_1^4}; \quad c_2 = \left( \lambda_2 \frac{l_2}{d_2} + \sum_{i=1}^{n_1} \xi_{2i} \right) \frac{8\rho}{\pi^2 d_2^4};$$

$Q_1, Q_2$  – расходы жидкости в ветвях трубопровода;

$\lambda_1, \lambda_2$  – коэффициенты гидравлического трения, зависящие от режима течения жидкости и шероховатости труб;

$l_1, l_2$  и  $d_1, d_2$  – длины и диаметры ветвей трубопровода;

$\xi_{1i}, \xi_{2i}$  – коэффициенты местных сопротивлений (участки изменения конфигурации трубопровода, вентили, задвижки и т.п.).

Поскольку на основании законов сохранения энергии  $\Delta P_1 = \Delta P_2$ ,

$Q_1 + Q_2 = Q$ , то получается система уравнений

$$\begin{cases} c_1 Q_1^2 = c_2 Q_2^2 \\ Q_1 + Q_2 = Q \end{cases}$$

Пусть в нашем случае

$$Q = 0,2 \text{ м}^3/\text{с}, \quad \rho = 998 \text{ кг}/\text{м}^3,$$

$$l_1 = 100 \text{ м}, \quad l_2 = 150 \text{ м}, \quad d_1 = 0,05 \text{ м}, \quad d_2 = 0,15 \text{ м},$$

$$\xi_1 = 20, \quad \xi_2 = 35, \quad \lambda_1 = \lambda_2 = 0,02.$$

### **Алгоритмизация математической модели**

Данная система уравнений сводится к системе линейных уравнений путем извлечения квадратного корня из первого уравнения:

$$\begin{cases} \sqrt{c_1} Q_1 - \sqrt{c_2} Q_2 = 0 \\ Q_1 + Q_2 = Q \end{cases}$$

Решение представлено в виде кода в m-файле pipe.m

---

pipe.m

```
Q=0.2; rho=998; lambda=0.02;
L=[100 150]; d =[0.05 0.15];
zeta=[20 35];
c= (lambda.*L./d+zeta).*8* rho./ ( pi ^2*d.^4)
A=[sqrt(c(1)) -sqrt(c(2));1 1]
b=[0;Q]; Qi=A\b
```

### Программа исследований

1. Найдите аналитическое решение системы уравнений и сравните с численным решением.

2. Как будут изменяться потери давления в ветвях трубопровода и расходы при изменении диаметра трубы в первой ветви.

3. Как будут изменяться потери давления в ветвях трубопровода и расходы при изменении длины первой ветви трубопровода.

4. Как будут изменяться потери давления в ветвях трубопровода и расходы, если в первой ветви трубопровода поставить задвижку (табл. 1) (т.е. добавляется местное сопротивление).

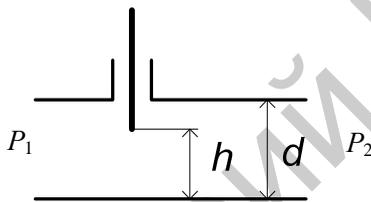


Рис. 13. Схема задвижки

Таблица 1

$h/d$	0,125	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
$\xi$	97,8	35	10	4,6	2,06	0,98	0,44	0,17	0,06	0

5. Определите расход жидкости в ветвях трубопровода для разных режимов течения и труб из различных материалов (исследовать влияние коэффициента гидравлического сопротивления). Режим течения жидкости определяется числом Рейнольдса

$$Re = \frac{\rho v d}{\mu} = \frac{v d}{\nu}$$

в табл. 2.

Таблица 2

Зона сопротивления	Границы зоны	Закон сопротивления
1. Ламинарное течение	$Re < 2300$	$\lambda = 64/Re$ – закон Гагена–Пуазейля
2. Зона гидравлически гладких труб при турбулентном режиме течения	$4000 < Re < 20 \frac{d}{K_9}$	$\lambda = \frac{0,3164}{Re^{0,25}}$ – формула Блазиуса $\lambda = \frac{1}{(1,8 \cdot \lg Re - 1,64)^2}$ – формула Филоненко–Альтшуля
3. Переходная зона (от зоны гидравлически гладких труб к зоне квадратичного сопротивления)	$20 \frac{d}{K_9} < Re < 500 \frac{d}{K_9}$	$\lambda = 0,11 \left( \frac{K_9}{d} + \frac{68}{Re} \right)^{0,25}$ – формула А.Д. Альтшуля
4. Зона квадратичного сопротивления	$500 \frac{d}{K_9} < Re$	$\lambda = \frac{1}{\left( 1,74 + 2 \lg \frac{d}{K_9} \right)^2}$ – формула Прандтля–Никурадзе $\lambda = 0,11 \left( \frac{K_9}{d} \right)^{0,25}$ – формула Б.Л. Шифринсона

Значения эквивалентной шероховатости  $K_s$  для труб из различных материалов представлены в табл. 3.

Таблица 3

Материал труб	Состояние труб	$K_s$ (среднее значение) мм
Тянутые из стекла и цветных металлов	Новые	0,001...0,01 (0,005)
	Новые	0,02...0,05 (0,03)
Стальные бесшовные	Бывшие в употреблении	0,015...0,03 (0,02)
	Новые	0,03...0,1 (0,5)
Стальные сварные	Бывшие в употреблении	0,08...1,5 (1,0)
	Новые	0,01...0,02 (0,15)
Стальные оцинкованные	Бывшие в употреблении	0,4...0,7 (0,5)
	Новые	0,2...0,5 (0,3)
Чугунные	Бывшие в употреблении	0,5...1,5 (1,0)
	Новые	0,05...0,1 (0,085)
Асбоцементные	Бывшие в употреблении	0,6
	При хорошей поверхности с затиркой	0,3...0,8 (0,5)
Бетонные	С грубой (шероховатой) поверхностью	3...9
	Рукава и шланги резиновые	0,03

6. Модифицируйте программу для трех ветвей трубопровода. Расходы жидкостей в ветвях трубопровода определяются системой уравнений

$$\begin{cases} c_1 Q_1^2 = c_2 Q_2^2 \\ c_3 Q_3^2 = c_1 Q_1^2 \\ Q_1 + Q_2 + Q_3 = Q \end{cases}$$

## ЗАПОЛНЕНИЕ РЕЗЕРВУАРА

Рассмотрим правила применения встроенной процедуры *ode45* для решения одного дифференциального уравнения первого порядка. Такое уравнение является математической моделью системы, состоящей из резервуара (танка), наполняемого жидкостью с помощью насоса.

### **Формулировка задачи**

Резервуар (танк) заполняется жидкостью с помощью насоса. В нижней части танка имеется отверстие, через которое жидкость вытекает наружу.

### **Предметная модель**

Танк имеет идеальную форму цилиндра. Цилиндр расположен вертикально. Сечение цилиндра постоянно и не зависит от уровня жидкости. Танк имеет отверстие в нижней части, из которого жидкость самопроизвольно вытекает. Площадь отверстия постоянна. Жидкость однородна. Плотность жидкости постоянна, жидкость несжимаема. Давление жидкости определяется только высотой столба жидкости. Функция скорости подачи жидкости определяется действующим насосом.

### **Параметры моделирования**

Начальный уровень жидкости в танке.  
Функция подачи насосом жидкости в танк.  
Сечение отверстия.

### **Математическая модель**

Изменение количества (массы) жидкости в танке определяется уравнением

$$\frac{dM}{dt} = \rho Q_{in} - \rho Q_{out}.$$

Масса жидкости в танке определяется площадью сечения танка, уровнем и плотностью жидкости:

$$M = \rho Sh.$$

Скорость истечения жидкости

$$Q_{out} = k\sqrt{h} ,$$

где коэффициент  $k$  пропорционален площади отверстия.

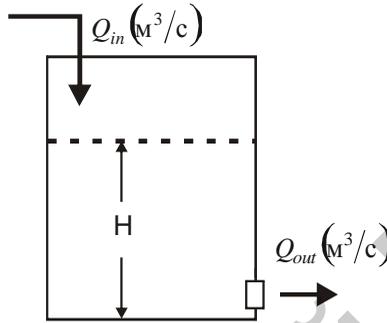


Рис. 14. Заполнение резервуара жидкостью

Соответствующей подстановкой в приближении нами модели это уравнение можно преобразовать

$$\frac{d(\rho Sh)}{dt} = \rho Q_{in} - \rho k\sqrt{h} ,$$

$$\rho S \frac{dh}{dt} = \rho Q_{in} - \rho k\sqrt{h} ,$$

$$\frac{dh}{dt} = \frac{1}{S} (Q_{in} - k\sqrt{h})$$

Изменение высоты  $h$  (уровня) жидкости в танке может быть описано дифференциальным уравнением

$$\frac{dh}{dt} = \alpha(t) - \beta\sqrt{h} .$$

Скорость подачи жидкости  $Q_{in}$  может быть постоянной или зависящей от времени, что более реально. Возьмем для приме-

ра  $Q_{in} = 10 + 5 \sin(t)$ . Скорость истечения жидкости из танка зависит от площади сечения отверстия (определяется коэффициентом  $k$ ) и уровня жидкости в танке. Полагая  $k$  и  $h_0$  равными 1, создадим  $m$ -файл, необходимый для решения дифференциального уравнения в интервале времени 30 секунд.

### Алгоритмизация математической модели

*Шаг 1.* Выясним вид функции  $f(t, y)$  правой части дифференциального уравнения. Функция имеет вид  $f(t, y) \rightarrow f(t, h) = \alpha(t) - \beta\sqrt{h}$ , поэтому

---

```
tankfill.m
function dhdt = tankfill(t,h)
A = 10 + 5*sin(t);
S = 3;
H = 2*sqrt(h);
dhdt = (A - H)/S;
```

---

*Шаг 2.* Используем процедуру *ode45* для решения.

В начальный момент времени  $t = 0$  высота жидкости в танке, по нашим предположениям,  $h_0 = 1$ . Интервал времени для интегрирования взяли от 0 до 30.

Создадим еще один файл, *tank.m*, содержащий следующий код:

---

```
tankfill_main.m
tspan = [0 30]; %интервал интегрирования
h0 = 1; % начальная высота
[t h] = ode45('tankfill', tspan, h0);
plot(t, h) %построение графика h(t)
xlabel ('time, s')
ylabel ('height, m')
```

---

Результаты представлены на рис. 15

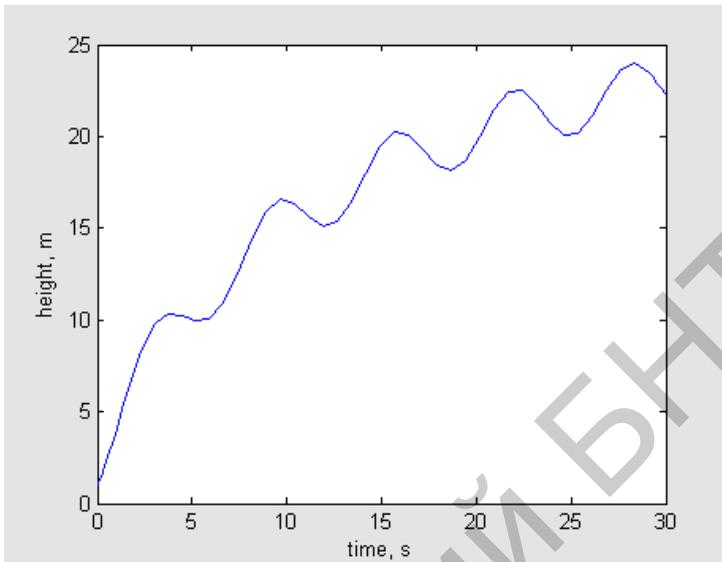


Рис. 15. Зависимость высоты жидкости в танке от времени

### ***Программа исследований***

1. Идеализируйте работу насоса, подающего жидкость в танк. Задайте скорость подачи жидкости постоянной ( $A = 10$ ) и сопоставьте полученный результат с предыдущим. Объясните различия в графиках.
2. Уменьшите площадь сечения выходного отверстия в несколько раз. Добейтесь ситуации, когда уровень жидкости в танке начинает превышать начальный, достигает максимума, а затем кривая выходит на насыщение. Объясните полученный результат.
3. Отключите подачу жидкости в танк ( $A = 0$ ). Наблюдайте кривую истечения жидкости из танка. Объясните вид кривой. При необходимости варьируйте интервал интегрирования.
4. Измените функцию подачи жидкости в танк  $A = 5 |\sin(t)|$ . Сопоставьте полученные результаты.
5. Закройте отверстие. Наблюдайте изменение уровня жидкости в танке.

## РАЗВИТИЕ БИОЛОГИЧЕСКИХ ПОПУЛЯЦИЙ

Первое глубокое математическое исследование закономерностей динамики взаимодействующих популяций дано в книге В. Вольтерра «Математическая теория борьбы за существование» (1931). Крупнейший итальянский математик Вито Вольтерра – основатель математической биологии – предложил описывать взаимодействие видов подобно тому, как это делается в статистической физике и химической кинетике, – в виде мультипликативных членов в уравнениях (произведений численностей взаимодействующих видов). Системы, изученные Вольтерра, состоят из нескольких биологических видов и запаса пищи, который используют некоторые из рассматриваемых видов. Особи различных биологических видов могут питаться одинаковой или различной пищей. В системе одновременно могут существовать несколько биологических видов особей (например, волки, медведи и зайцы). Особи могут конкурировать друг с другом при добывании пищи. Среди особей могут присутствовать хищные виды, поедающие другой вид особей.

### Формулировка проблемы

В озере существуют два вида рыб: рыбы-хищники (щуки) и рыбы-жертвы (караси) (рис. 16). Рассмотреть динамику развития популяций рыб.

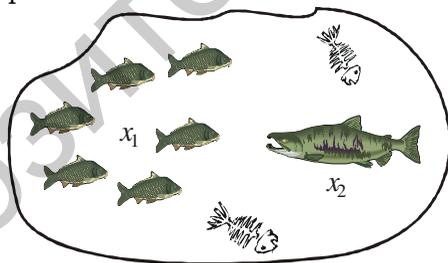


Рис. 16. Популяции рыб в озере

### Модель развития биологических популяций

О компонентах системы формулируются следующие допущения:

1. В системе существуют только два вида биологических особей: хищники (щуки) и жертвы (караси).

2. Щуки питаются только карасями.

3. Караси питаются другой пищей (не щуками). Пища для карасей либо имеется в неограниченном количестве, либо ее поступление с течением времени жестко регламентировано.

4. Особи каждого вида отмирают так, что в единицу времени погибает постоянная доля существующих особей.

5. Хищные виды поедают жертвы, причем, в единицу времени количество съеденных жертв всегда пропорционально вероятности встречи особей этих двух видов, т.е. произведению количества хищников на количество жертв.

6. Если имеются пища в неограниченном количестве и несколько видов, которые способны ее потреблять, то доля пищи, потребляемая каждым видом в единицу времени, пропорциональна количеству особей этого вида, взятого с некоторым коэффициентом, зависящим от вида (модели межвидовой конкуренции).

7. Если вид питается пищей, имеющейся в неограниченном количестве, прирост численности вида за единицу времени пропорционален численности вида.

8. Если вид питается пищей, имеющейся в ограниченном количестве, то его размножение регулируется скоростью потребления пищи, т.е. за единицу времени прирост пропорционален количеству съеденной пищи.

Перечисленные гипотезы позволяют описывать сложные биологические системы при помощи систем обыкновенных дифференциальных уравнений, в правых частях которых имеются суммы линейных и билинейных членов. Такими уравнениями описываются и системы химических реакций.

### ***Параметры моделирования***

Скорость роста каждого вида щук (карасей) при отсутствии другого вида карасей (щук).

Скорость гибели каждого вида при отсутствии другого (естественная, от голода).

Коэффициенты влияния на скорость роста (гибели) каждой популяции.

### ***Математическая модель***

В общем виде, с учетом самоограничения численности по логистическому закону, система дифференциальных уравне-

ний, описывающая взаимодействие двух видов, может быть записана в форме

$$\begin{cases} \frac{dx_1}{dt} = a_1x_1 - b_{12}x_1x_2 \\ \frac{dx_2}{dt} = -a_2x_2 + b_{21}x_1x_2 \end{cases} \quad (1)$$

Скорость изменения популяции данного вида, например, условно названной первой – карасей,  $\frac{dx_1}{dt}$  растет вследствие естественного прироста данной популяции со скоростью  $a_1$  и пропорциональна количеству особей данной популяции  $x_1$ . Скорость изменения популяции данного вида определяется также ее взаимодействием с другой популяцией – щуками, выражаемой константой взаимодействия первого и второго видов  $b_{12}$  и произведением соответствующих  $x_1$  и  $x_2$ . Знак минус при  $b_{12}$  означает, что наличие щук приводит к уменьшению популяции карасей (щуки питаются карасями). В свою очередь, знак минус при  $a_2$  означает, что при отсутствии карасей (пищи для щук) популяция щук убывает.

В общем случае в зависимости от видов особей знаки при коэффициентах  $a_i$  и  $b_{ij}$  могут быть (+) или (-), включая ноль. Соответствие знаков коэффициентов  $b_{ij}$  различным типам взаимодействий приведено в табл. 4.

Таблица 4

Типы взаимодействия видов

Тип взаимодействия	Коэффициент	
	$b_{12}$	$b_{22}$
Симбиоз	$> 0$	$> 0$
Комменсализм	$> 0$	$0$
Хищник-жертва	$> 0$	$< 0$
Аменсализм	$0$	$< 0$
Конкуренция	$< 0$	$< 0$
Нейтрализм	$0$	$0$

Исследование свойств перечисленных в таблице моделей приводит к некоторым важным выводам относительно исхода взаимодействия видов. Так, например, уравнения конкуренции ( $b_{12} > 0$ ,  $b_{21} < 0$ ) предсказывают выживание одного из двух видов в случае, если собственная скорость роста другого вида меньше некоторой критической величины.

*Симбиоз* – длительное тесное сожительство двух популяций разных видов, при котором они приносят друг другу взаимную пользу.

*Комменсализм* – сотрапезничество, нахлебничество, сожительство популяций разных видов, характеризующееся тем, что одна из них (комменсал) постоянно или временно живет за счет другой, не причиняя ей вреда.

*Аменсализм* (от греч. а – отрицательная частица и лат. mensa – стол, трапеза) – форма взаимоотношений между популяциями, полезная для одного вида, но вредная для другого.

Пусть начальные условия в рассматриваемой задаче следующие:

начальное количество щук	500;
начальное количество карасей	25000;
коэффициент прироста количества карасей	5 (1/год);
коэффициент прироста количества щук (при отсутствии карасей хищники гибнут от голода и их популяция уменьшается)	3 (1/год);
влияние популяции щук на развитие популяции карасей	0.01 (1/год);
влияние популяции карасей на развитие популяции щук	0.0001(1/год).

### **Алгоритмизация математической модели**

Приведем систему уравнений (1) к стандартному виду для решения с использованием встроенной функции *ode45*:

$$\frac{dx}{dt} = \frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_1 x_1 - b_{12} x_1 x_2 \\ -a_2 x_2 + b_{21} x_1 x_2 \end{bmatrix}, \quad (2)$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad f(t, x) = \begin{bmatrix} a_1 x_1 - b_{12} x_1 x_2 \\ -a_2 x_2 + b_{21} x_1 x_2 \end{bmatrix} \quad x(0) = x_0 = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}.$$

Создадим файл predator.m, описывающий правую часть системы дифференциальных уравнений:

---

```
predator.m
function dxdt = predator(t,x)
global a1 a2 b12 b21
dxdt = [(a1 - b12*x(2))*x(1); (-a2 + b21*x(1))*x(2)];
```

---

Код m-файла predator\_main.m с процедурой обращения к файлу predator.m имеет вид

---

```
predator_main.m
clc
global a1 a2 b12 b21
a1 = 5;
b12 = -0.01;
a2 = -3;
b21 = 0.0001;
x10 = 25000;
x20 = 500;
x0 = [x10; x20];
dt = [0 5];
opt = odeset('RelTol', 1e-6)

[t x] = ode45(@predator, dt, x0, opt);
%[t x] = ode45(@predator, dt, x0);

subplot(2,1,1)
plot(t,x(:,2))
xlabel('time, year')
ylabel('predator fish')
subplot(2,1,2)
plot(t,x(:,1))
xlabel('time, year')
ylabel('prey fish')
```

---

### *Программа исследований*

1. Выполните приведенную программу. Результат представлен на рис. 17. Обратите внимание на цикличность изменения количества жертв и количества хищников. В какие моменты времени количество жертв минимально? Максимально? Что можно сказать о популяции хищников для этих моментов времени?

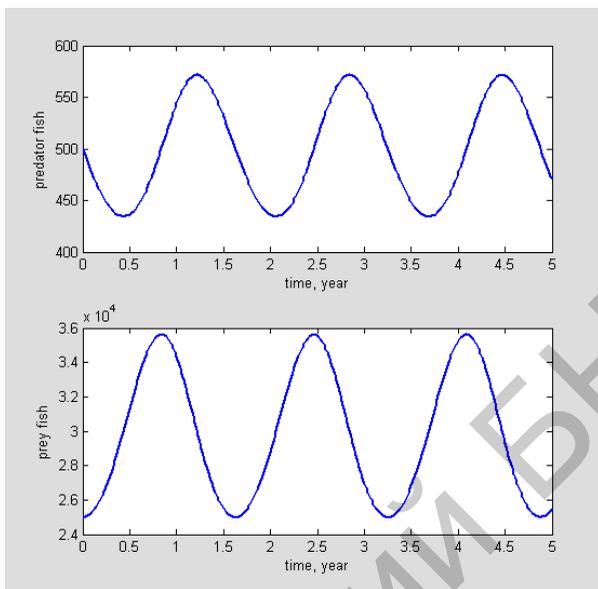


Рис. 17. Изменение численности рыб

2. Измените начальные условия. Рассмотрите случаи количества жертв и хищников в сочетании (40000, 400), (30000, 600). Как изменился вид кривых, описывающих развитие популяции жертв и хищников? Изменился ли интервал времени между максимумами? Сами максимумы? Ответ обоснуйте.

3. Для того чтобы влиять на развитие популяции рыб, разрешите вылов особей рыб-хищников, достигших 2-летнего возраста. Задайте число разрешенных для вылова хищников, например, 200 (1/год). Вылов разрешен в течение 3 месяцев. Модифицируйте вид правой части системы уравнений, описывающих развитие популяции. Подсказка. Добавьте в правую часть уравнения 2 математической модели выражение вида  $-v_2 u$ , где

$$v_2 = 200, \text{ а функция } u \text{ имеет вид } u = \begin{cases} 0, & t < 2 \\ 1, & 2 \leq t \leq 2.25 \\ 0, & t > 2.25 \end{cases}$$

С помощью логического оператора IF модифицируйте логику программного кода predator.m. Выполните программу и объясните полученный результат. Как изменилось развитие популяций после отлова части хищников?

## ЭПИДЕМИЯ ГРИППА

### Формулировка проблемы

В марте 1978 года английский медицинский журнал сообщил об эпидемии гриппа в школе. Численность учащихся составляла 763. Динамика численности заболевших  $I(t)$  по дням представлена в табл. 5.  $S(t)$  – численность здоровых, еще не болевших учащихся.

Таблица 5

День	0	3	4	5	6	7	8	9	10	11	12	13	14
$S(t)$	762	740	650	400	250	120	80	50	20	18	15	13	10
$I(t)$	1	20	80	220	300	260	240	190	120	80	20	5	2

Сделать анализ развития заболевания и прогноз его развития на будущее.

### Модель эпидемии гриппа

Среди учащихся вследствие контакта с инфицированным возможна передача инфекции здоровому учащемуся и, как следствие, его заболевание. Инфицированные учащиеся со временем выздоравливают. После выздоровления учащиеся могут приобрести иммунитет и больше не болеть. Учащиеся не подвергаются контакту с другими инфицированными людьми за пределами школы. Контакты с преподавателями не принимаются во внимание.

### Параметры модели

Количество учащихся в школе.

Вероятность заболевания.

Вероятность выздоровления.

### Математическая модель

Математическая модель развития эпидемии может быть представлена системой из трех дифференциальных уравнений первого порядка:

$$\begin{cases} \frac{dS}{dt} = -aSI \\ \frac{dI}{dt} = +aSI - bI \\ \frac{dR}{dt} = +bI \end{cases} \quad (3)$$

Первое уравнение этой системы описывает скорость заболевания здоровых учащихся, количество которых уменьшается. Вероятность заболевания пропорциональна произведению количества здоровых  $S$  (которые могут заболеть с вероятностью  $a$ ) на количество инфицированных  $I$ . Второе уравнение описывает скорость изменения количества заболевших  $dI/dt$ . Наряду с поступлением инфицированных ( $+aSI$ ) из этой группы выводится часть переболевших и поправившихся с вероятностью  $b$ . Переболевшие попадают в группу восстановившихся после заболевания. Скорость изменения численности учащихся  $R$  в этой группе описывается третьим уравнением системы. Количество инфицированных, восстановившихся и еще не болевших связаны очевидным соотношением  $R(t) = 763 - S(t) - I(t)$ .

### Алгоритм решения

Система уравнений (3) не может быть решена до тех пор, пока не найдены неизвестные коэффициенты  $a$  и  $b$ . Значения этих коэффициентов могут быть определены из следующих рассуждений. Второе уравнение может быть представлено в виде

$$\frac{1}{I} \frac{dI}{dt} = +aS - b. \quad (4)$$

Это означает, что зависимость  $1/I dI/dt$  является линейной функцией от  $S$  с параметрами  $a$  и  $b$ .

Нанесем на график экспериментальную зависимость. Как следует из математики, производная  $dI/dt$  в общем случае может быть аппроксимирована значениями функции  $I(h)$  в соседних точках в виде

$$\frac{dI}{dt} \cong \frac{I(t+h) - I(t-h)}{2h}.$$

Построим график зависимости (4) (рис. 18). Код программы построения графика

---

построение графика

---

```
S = [ 762 740 650 400 250 120 80 50 20 18 15 13 10];  
I = [1 20 80 220 300 260 240 190 120 80 20 5 2];  
days = [0 3 4 5 6 7 8 9 10 11 12 13 14];  
for k = 1:10  
    y(k) = (1/I(k+2))*(I(k+3)-I(k+1))/2;  
    x(k)=S(k+2);  
end  
plot(x,y,'o')
```

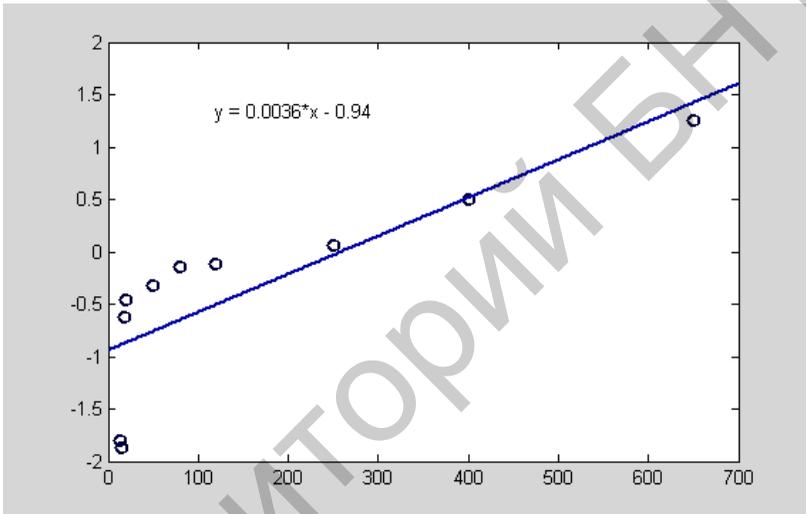


Рис. 18. График зависимости (4)

Интерполируем эту зависимость прямой (вопрос интерполяции будет рассмотрен в [приложении](#)). Линейная интерполяция дает для коэффициентов  $a$  и  $b$  значения 0.0036 и 0.94 соответственно. Следует отметить, что две первые точки на графике существенно выпадают из общего характера расположения других точек, поэтому исключим две последние экспериментальные точки из рассмотрения. Линейная аппроксимация в этом случае дает значения  $a = 0.0026$ ,  $b = 0.51$  (рис.19) (смотри [приложение](#)).

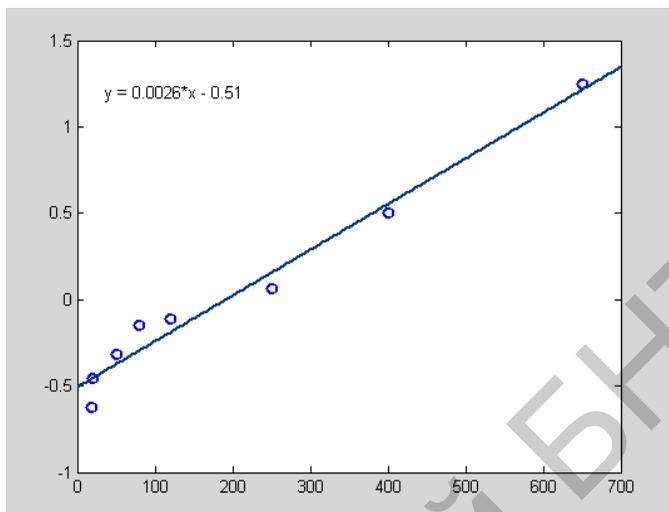


Рис. 19. Линейная интерполяция

### **Алгоритм моделирования**

Построим процедуру вычисления правой части системы (3).

---

`influenza.m`

---

```
function dzdt = influenza(t, y, p)
dzdt = [-p(1)*y(1)*y(2); p(1)*y(1)*y(2)-p(2)*y(2);
p(2)*y(2)];
```

Основной m-файл для обращения к процедуре ode45 и построения графика имеет вид

---

`influenza_main.m`

---

```
clc
S = [ 762 740 650 400 250 120 80 50 20 18 15 13 10];
I = [1 20 80 220 300 260 240 190 120 80 20 5 2];
days = [0 3 4 5 6 7 8 9 10 11 12 13 14];

y0 = [762 1 0];
dt = [0 14];
[t, y] = ode45(@influenza, dt, y0, [], [0.0026 0.51]);
subplot(2,1,1)
```

```

plot(t, y(:,1), days, S, 'o', 'LineWidth', 2)
title('График функции для группы риска',
'FontName','Arial Unicode MS')
ylabel('количество в группе риска',
'FontName','Arial Unicode MS')

subplot(2,1,2)
plot(t, y(:,2), days, I, 'o', 'LineWidth', 2)
title('График функции инфицированных',
'FontName','Arial Unicode MS')
ylabel('количество инфицированных',
'FontName','Arial Unicode MS')

xlabel('время, дни', 'FontName','Arial Unicode MS')

```

Результаты представлены на рис. 20. На графиках сопоставляются теоретические кривые и экспериментальные значения, показанные маркерами «о».

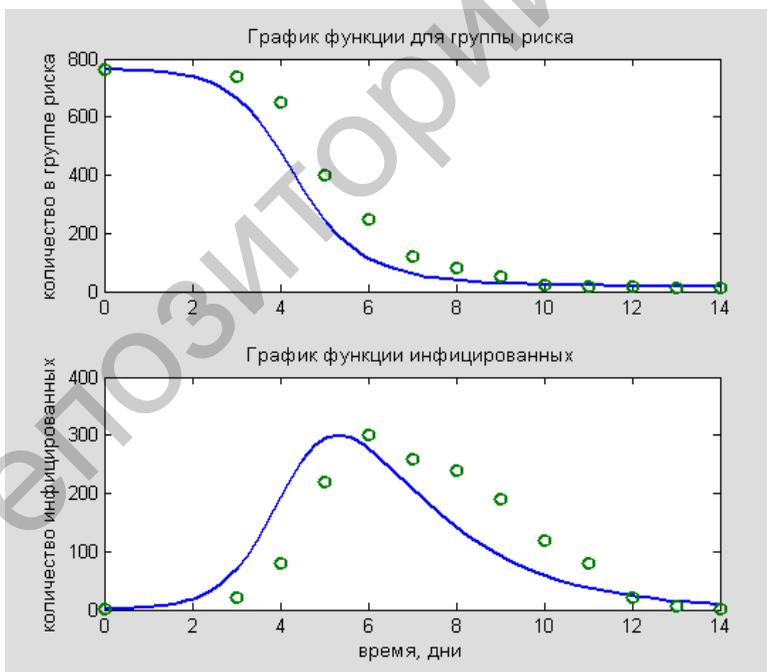


Рис. 20. Теоретические и экспериментальные зависимости

### Программа исследований

1. Постройте график функции  $R(t)$ , отражающий динамику количества школьников, переболевших гриппом.

2. Предположим, что вы воспользовались значениями  $a = 0.0036$ ,  $b = 0.94$  и не стали отбрасывать две экспериментальные точки, существенно не укладывающиеся на прямую. Постройте графики и сопоставьте с более точной аппроксимацией.

3. Более корректный расчет показывает, что параметры аппроксимации следует принимать  $a = 0.0022$ ,  $b = 0.4362$ . Внесите изменения и постройте графики для группы риска и количества инфицированных. Как изменилось соответствие теоретических и экспериментальных данных?

4. Предположим, что вам удалось анализ эпидемии для населения небольшого городка. Математическая модель эпидемии имеет вид

$$\begin{cases} \frac{dx}{dt} = -\beta xy + \gamma \\ \frac{dy}{dt} = \beta xy - \alpha y \\ \frac{dz}{dt} = \alpha y - \gamma \end{cases}$$

5. Численные значения параметров  $\alpha = 0.05$ ,  $\beta = 0.0002$ ,  $\gamma = 4.5$ . Население городка составляет 9875 человек. На момент начала заболевания  $y(0) = 20$ ,  $z(0) = 0$ . Рассмотрите и установите смысл  $x$ ,  $y$ ,  $z$  и параметров  $\alpha$ ,  $\beta$ ,  $\gamma$ . Получите кривые развития эпидемии. Какие преимущества дает вам знание пика эпидемии? Для населения? Для обеспечения аптек лекарствами?

**Аппроксимация экспериментальных данных**

При построении математических моделей часто возникает необходимость описать экспериментальные данные уравнением (эмпирической формулой).

Пусть на числовой оси заданы точки  $x_1, x_2, \dots, x_i, \dots, x_n$  (обычно их называют узлами) и соответствующие им значения  $y_1, y_2, \dots, y_i, \dots, y_n$  (рис.21).

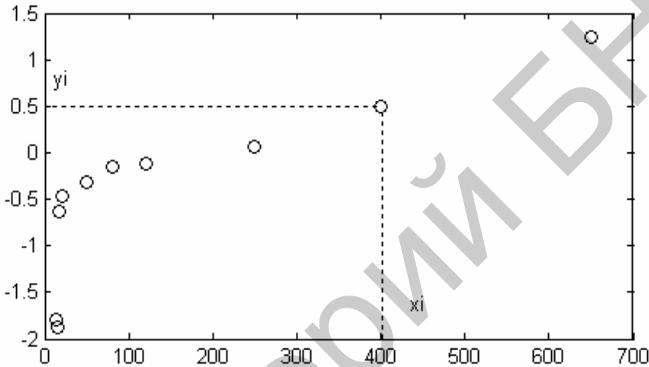


Рис. 21. Экспериментальная зависимость

Задача интерполяции состоит в том, чтобы построить функцию, которая оптимальным образом проходит через заданные точки. Интерполяция позволяет оценивать значение заданной таблично функции в тех точках, для которых данных нет.

Функцию, приближенно описывающую экспериментальные данные, называют аппроксимирующей, а сам процесс построения функции – аппроксимацией.

Первым этапом аппроксимации является выбор вида аппроксимирующей функции (рис.22). На график наносят экспериментальные точки и анализируют вид полученной зависимости, сравнивают с графиками типичных математических функций. Для описания экспериментальной зависимости обычно подбирают легко вычисляемые функции: линейные функции, полиномы, экспоненциальные (для явлений типа распада и роста) и тригонометрические функции (для периодических процессов).

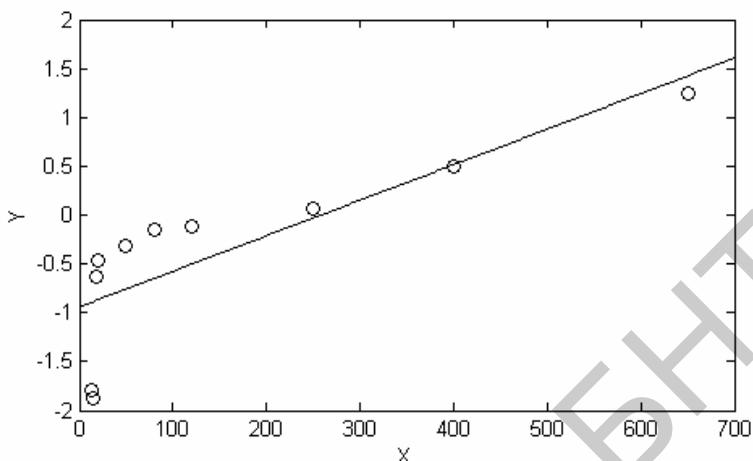


Рис. 22. Построение аппроксимирующей функции

Второй этап – это определение коэффициентов аппроксимирующей функции. Нелинейные зависимости стараются свести к линейным путем замены переменных.

На третьем этапе оценивают степень точности аппроксимации по корреляции между экспериментальными и расчетными данными.

Одним из самых распространенных методов для приближенного представления функций является метод наименьших квадратов.

Метод базируется на том, что сумма квадратов отклонений значений аппроксимирующей функции в узловых точках от экспериментальных должна быть минимальной:

$$S = \sum_{i=1}^n (\Delta y_i)^2 = \sum_{i=1}^n (y_i - y(x_i))^2 = \min ,$$

где  $n$  – это заданное количество точек.

После дифференцирования  $S$  по неизвестным коэффициентам аппроксимирующей функции получают систему линейных уравнений для этих коэффициентов.

В случае линейной аппроксимирующей функции  $y = a + bx$

$$a = \frac{\sum y_i \sum x_i^2 - \sum x_i \sum x_i y_i}{n \sum x_i^2 - (\sum x_i)^2},$$

$$b = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2},$$

где суммирование ведется от 1 до  $n$ .

Аппроксимация полиномами в системе MATLAB осуществляется с помощью функции `polyfit`, которая имеет вид

$$p = \text{polyfit}(x, y, n),$$

где  $x$  – вектор аргументов;

$y$  – вектор значений функции;

$n$  – степень полинома;

$p$  – вектор его коэффициентов.

В случае линейной аппроксимирующей функции  $n = 1$ .

Найдем коэффициенты линейной функции в примере об [эпидемии гриппа](#) и построим графики для экспериментальной и теоретической зависимостей.

```
p=polyfit(x,y,1)
plot(x,polyval(p,x),'-',x,y,'o')
```

Искомый коэффициент  $a$  – это значение элемента  $p(1)$ , искомый коэффициент  $b$  – это значение элемента  $p(2)$ . Таким образом, экспериментальная зависимость описывается уравнением  $y = 0.0036x - 0.94$ .

## ЭКСТРЕМАЛЬНЫЕ ПРЫЖКИ

### Формулировка проблемы

Любитель экстремальных ощущений привязан упругим канатом и прыгает с моста с высоты 80 метров над поверхностью воды. Длина каната 30 метров. В распоряжении прыгуна имеются канаты с тремя различными коэффициентами упругости, приведенными в табл. 6.

Таблица 6

Упругий канат	Коэффициент упругости $k$ , Н/м
А	5
В	40
С	500

Прыгуну необходимо как можно ближе приблизиться к поверхности воды, не ударяясь о воду. На каком канате следует сделать выбор любителю острых ощущений?

### Параметры моделирования

Жесткость каната.

Длина каната.

Масса прыгуна.

Параметры силы сопротивления.

### Физическая модель

Падение происходит строго вертикально и маятниковые качания отсутствуют. Отсутствует боковой ветер.

Подъемной силой и силой Архимеда можно пренебречь.

Массой каната можно пренебречь.

В точке поворота канат не разрывается.

В точке поворота прыгун не испытывает смертельных перегрузок.

Сопротивление воздуха определяется линейной и квадратичной зависимостью от скорости  $R = av + bv^2$ , где параметры  $a$  и  $b$  равны 1.

Совместим начало координат с точкой, где начинает растягиваться канат - 50 м от поверхности воды (рис. 23). Нулевой уровень показан штриховой линией. Ось  $x$  направлена вниз.

На прыгуна в любой точке траектории действуют сила тяжести  $\vec{P} = m\vec{g}$ , направленная вертикально вниз, сила сопротивления воздуха  $\vec{R} = -a\vec{v} - b|\vec{v}|\vec{v}$ , направленная всегда противоположно направлению скорости, и сила упругости  $F = -kx$ , возникающая только при растяжении каната (положение прыгуна ниже нулевой точки показано на рис. 23).

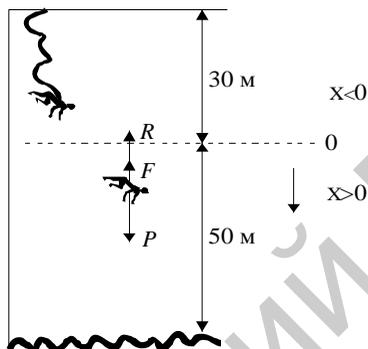


Рис. 23. К постановке задачи о прыгуна

Будем считать, что масса прыгуна с экипировкой составляет 90 кг, а ускорение свободного падения  $g = 10 \text{ м/с}^2$ .

### Математическая модель

Математическая модель (уравнение движения прыгуна) может быть получена на основании второго закона Ньютона применительно к данной системе. По второму закону Ньютона сумма всех сил, действующих на прыгуна, равна произведению его массы на ускорение. С учетом направления движения и сил второй закон Ньютона в данном случае имеет вид

$$ma = P - R - F,$$

$$ma = mg - av - b|v|v - F.$$

Поскольку  $v = \frac{dx}{dt}$  и  $a = \frac{dv}{dt} = \frac{d^2x}{dt^2}$ , получаем уравнение движения в виде дифференциального уравнения второго порядка:

$$\frac{d^2x}{dt^2} = g - \frac{F}{m} - \frac{a}{m} \frac{dx}{dt} - \frac{b}{m} \left| \frac{dx}{dt} \right| \frac{dx}{dt}.$$

### Алгоритмизация математической модели

Для решения математической модели воспользуемся стандартной процедурой ODE45 из пакета Matlab.

Далее необходимо создать функцию odefun. Для этого приведем наше дифференциальное уравнение к виду, удобному для интегрирования. Код программы для решения описанной задачи состоит из процедуры решения дифференциального уравнения bungee.m и m-файла bungee\_main.m, обращающегося к этой процедуре.

---

bungee\_main.m

---

```
figure
[t,xsol]=ode45(@bungee,[0 50],[-30 0],[],5);
plot(t,50-xsol(:,1));

figure
[t,xsol]=ode45(@bungee,[0 50],[-30 0],[],40);
plot(t,50-xsol(:,1));

figure
[t,xsol]=ode45(@bungee,[0 50],[-30 0],[],500);
plot(t,50-xsol(:,1));
```

---

bungee.m

---

```
function dxdt=bungee(t,x,k)
m=90;
g=10;
a=1;
b=1;
P=m*g;
R=a*x(2)+b*abs(x(2))*x(2);
if x(1)>0
    F=k*x(1);
else
    F=0;
end

dxdt=[x(2);(P-F-R)/m];
```

Простая оценка времени свободного падения с высоты 80 метров из формулы  $h = gt^2/2$  дает значение 4 секунды. Исходя из этого для интервала времени процедуры интегрирования возьмем 50 секунд. Этого времени будет достаточно для наблюдения падения даже с квадратичным законом сопротивления воздуха.

### Графическое представление результатов моделирования

На рис. 24-26 представлены результаты моделирования для всех трех канатов. Для удобства восприятия система координат на графиках приведена к случаю, когда ось  $x$  направлена вверх, а нулевой уровень отсчитывается от поверхности воды.

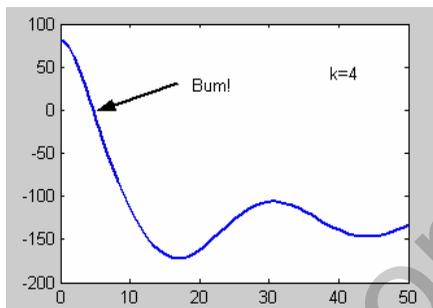


Рис. 24. Канат А. Прыгун ударяется о поверхность воды

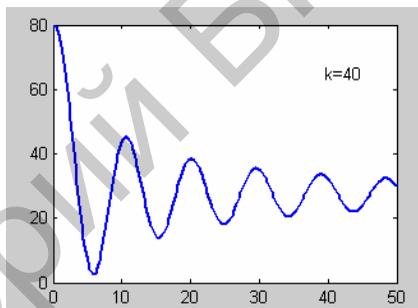


Рис. 25. Канат В. Прыгун достигает своей цели и не разбивается

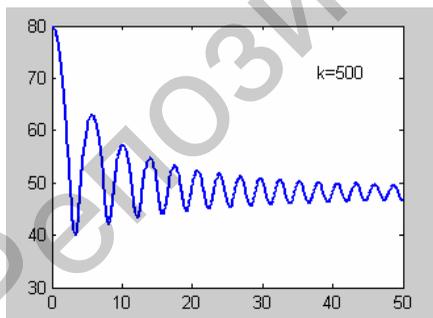


Рис. 26. Канат С. Прыгун не разбивается, но не достигает своей цели. Точка возврата очень далека от поверхности

### Программа исследований

1. Предположим, что канатом решил воспользоваться другой прыгун. Рассмотрите случаи более тяжелого (масса 120 кг) и более легкого (40 кг) прыгунов. К чему приводит увеличение массы прыгуна?

2. Постройте график ускорения движения прыгуна. Обратите внимание на вид кривой вблизи максимального растяжения каната. По графику оцените максимальное значение ускорения и сопоставьте его с ускорением свободного падения (помните, что если ускорение превосходит  $5g$ , это может быть опасно для жизни).

3. Оцените максимальную силу, действующую на прыгуна в момент максимального растяжения каната ( $F = ma_{\max}$ ). Как вы думаете, будет прыгун испытывать болезненные ощущения?

## ПРЫЖКИ С ПАРАШЮТОМ

### Формулировка проблемы

Катапульта выбрасывает парашютиста (манекен человека) с высоты 6000 метров. Рассмотреть 2 случая: 1) катастрофа – парашют не раскрылся; 2) удачное приземление – с раскрытым парашютом на безопасной скорости. Исследовать кинематику и динамику движения, условия безопасного приземления парашютиста.

### Физическая модель

Наиболее упрощенная физическая модель – это движение материальной точки, имеющей определенную массу, под действием гравитационного поля Земли. В этом случае сопротивление среды не учитывается, т.к. материальная точка не обладает размерами. Тело движется только под действием силы тяжести. Движение материальной точки в гравитационном поле Земли в вертикальном направлении является равноускоренным с ускорением свободного падения  $g \approx 9,81 \text{ м/с}^2$ . Такая модель очень грубая, т.к. не ограничивает скорость парашютиста ( $v = gt$ ).

В действительности сила сопротивления воздуха препятствует бесконечному увеличению скорости (даже при нераскрытом парашюте!) и ограничивает предельную скорость падения и безопасность приземления. Схема действия сил на парашютиста показана на рис. 27.

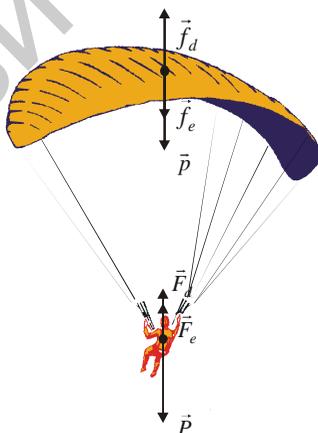


Рис. 27. К постановке задачи о парашютисте

Сила сопротивления среды движущемуся телу растет с ростом скорости. При относительно малых скоростях величина силы сопротивления линейно зависит от скорости:

$$F_{\text{сопр}} = k_1 \cdot v,$$

где  $k_1$  определяется свойствами среды и формой тела.

Например, для шарика

$$k_1 = 6\pi\eta r - \text{формула Стокса,}$$

где  $\mu$  – динамическая вязкость среды;

$r$  – радиус шарика. Для воздуха при  $t = 20^\circ\text{C}$  и давлении 1 атм  $\mu = 0,0182 \text{ Н}\cdot\text{с}\cdot\text{м}^{-2}$ , для воды  $\mu = 1,002 \text{ Н}\cdot\text{с}\cdot\text{м}^{-2}$ , для глицерина –  $\mu = 1480 \text{ Н}\cdot\text{с}\cdot\text{м}^{-2}$ .

С ростом скорости сила сопротивления пропорциональна квадрату скорости:

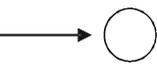
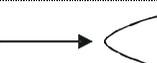
$$F_{\text{сопр}} = k_2 \cdot v^2.$$

Разумеется, линейная по скорости часть силы сопротивления формально также сохранится, но если  $k_2 \cdot v^2 \gg k_1 \cdot v$ , то вклад линейного члена можно пренебречь (это конкретный пример ранжирования факторов). Величина  $k_2$  пропорциональна площади сечения тела  $S$ , поперечного по отношению к потоку, плотности среды и зависит от формы тела. Обычно принимают  $k_2 = 0,5cS\rho_{\text{среды}}$ , где  $c$  – коэффициент лобового сопротивления (безразмерен). Значения  $c$  зависят от формы тела и для некоторых (не очень больших) скоростей приведены в табл. 7.

С учетом этого формулировка физической модели включает следующие предположения. На любой высоте вязкость воздуха (динамическая) постоянна. На парашют и парашютиста действуют силы вязкого трения. Плотность воздуха и температура также не меняются. Парашют раскрывается сразу после выхода парашютиста из самолета (в реальности составляет около 2 с). В момент катапультирования горизонтальная составляющая скорости отсутствует, парашютист движется только в вертикальном направлении и начинает движение с нулевой начальной скоростью. Гравитационное поле однородное. Парашютист и парашют связаны жесткими стропами. Воздушные потоки в различных направлениях отсутствуют.

Таблица 7

Значения коэффициента лобового сопротивления для некоторых тел, поперечное сечение которых имеет указанную на рисунке форму

	Диск	$c = 1,11$
	Полусфера	$c = 1,33$
	Полусфера	$c = 0,55$
	Шар	$c = 0,4$
	Каплевидное тело	$c = 0,045$
	Каплевидное тело	$c = 0,01$

### Параметры задачи

Масса парашютиста и парашюта.

Коэффициенты вязкого трения для парашютиста.

Начальная высота.

Прыжок совершается над поверхностью Земли в однородном гравитационном поле.

### Математическая модель

Математическая модель решения поставленной задачи может быть получена на основании второго закона Ньютона:

$$m\mathbf{a} = m\mathbf{g} - F_{\text{сопр}}, \quad F_{\text{сопр}} = -k_1\mathbf{v} - k_2\mathbf{v}^2,$$

$$m \frac{d^2y}{dt^2} = mg - k_1v - k_2v^2.$$

После замены переменных дифференциальное уравнение второго порядка сводится к системе двух дифференциальных уравнений первого порядка:

$$\begin{cases} \frac{dy}{dt} = -v \\ \frac{dv}{dt} = \frac{mg - k_1v - k_2v^2}{m} \end{cases} \quad (5)$$

Поясним смысл и некоторые значения параметров, входящих в уравнение движения (5).

$t$  – время движения, изменяющееся от нуля с момента катапультирования до момента приземления ( $t = 0 \div t_{fin}$ ).

$h$  – высота, изменяющаяся от начальной высоты катапультирования до 0 ( $h = h_0 \dots 0$ , по условию  $h_0 = 6000$  м).

$v$  – скорость движения манекена в вертикальном направлении ( $v_0 = 0$ ).

$m$  – масса парашютиста со снаряжением.

$g$  – ускорение свободного падения ( $g \approx 9,81$  м/с<sup>2</sup>).

$k_1$  и  $k_2$  – коэффициенты, определяющие вклад линейной и квадратичной составляющей скорости в сопротивление среды:

$$k_1 = 6\mu r, \quad (6)$$

$$k_2 = 0,5cS\rho_{\text{среды}}, \quad (7)$$

где  $\mu$  – динамическая вязкость среды (для воздуха  $\mu = 0,0182$  Н·с·м<sup>-2</sup>);

$r$  – радиус тела;

$c$  – коэффициент лобового столкновения;

$S$  – площадь сечения тела, поперечного по отношению к потоку;

$\rho_{\text{среды}}$  – плотность среды.

Оценим значения  $m$ ,  $r$ ,  $c$  и  $S$ .

*Случай 1.* Рассмотрим падение парашютиста без парашюта («лежа», раскинув руки в стороны). Средний рост человека 1,7 м, полуобхват грудной клетки приблизительно 0,4 м. Выберем в качестве коэффициента лобового сопротивления число  $c = 1,2$  как среднее между коэффициентами для диска

и для полусферы (выбор для качественной оценки правдоподобен). Оценим площадь:  $S = 1,7 \cdot 0,4 = 0,7 \text{ м}^2$ .

*Случай 2.* В случае, если парашют раскрылся, коэффициент лобового сопротивления составит  $c = 1,33$  (полусфера). Для оценки  $m$ ,  $S$  и безопасной скорости приземления были проанализированы характеристики различных парашютов (табл. 8). Было принято значение  $m = 120$ , площадь  $S$  можно будет вводить из командной строки (15...83 м<sup>2</sup>). Безопасная скорость приземления – не более 5 м/с.

Таблица 8

Тип парашюта	Масса парашютиста, кг	Минимальная безопасная высота, м	Площадь парашюта, м <sup>2</sup>	Средняя вертикальная скорость снижения, м/с
Д-6	140	150	83	5,0
Д-1-5-У	120	180	82,5	5,1
УР-15	100	150	50	5,75
ПО-16	100	600	22	5,0
МАЛЬВА-24	100	300	24,2	нет данных
Радар-М	95	300	15,6	нет данных
ПЗ-81	100	150	27	5,62
РЕЗЕРВ-16	100	150	15,6	нет данных

### Программный код

```
fall_main.m
clc
global S m g mu
m=120;
g=9.81;
mu=0.0182;
v = 0;
h = 6000;
% площадь сечения парашютиста и парашюта
S1=0.7;
S2=82,5;
% S1=input('input square of a man body (0.5...1.0) ');
% S2=input('input square of a parachute (15...83) ');
tspan=[0:2000];
```

```

% расчет для нераскрывшегося парашюта
% 2-мерная матрица высоты и скорости – начальные условия
y0=[h,v];
S=S1;
[t1,y1]=ode45('fall',tspan,y0);

% расчет для раскрывшегося парашюта
S=S2;
[t2,y2]=ode45('fall',tspan,y0);

% создание 2 окон вывода графиков,
% расположенных одно под другим
subplot(2,1,1)

% график высоты от времени
imax=find(y1(:,1)>0);
indexmax=max(imax);
plot(t1(1:indexmax),y1((1:indexmax),1),'b--','LineWidth',2)

% удержание графика в окне
hold on
% 2-й график зависимости высоты от времени
imax=find(y2(:,1)>0);
indexmax=max(imax);

plot(t2((1:indexmax)),y2((1:indexmax),1),'g-','LineWidth',2)
legend('без парашюта', 'с парашютом')
hold on
% отображение сетки
grid on
% заголовок верхнего окна, подписи осей
title('Dependence of height from time')
ylabel('Height H,m')
xlabel('time t,s')

% создание нижнего окна для вывода зависимости
% скорости от времени; вывод графиков и подписи осей
subplot(2,1,2)

imax=find(y1(:,1)>0);
indexmax=max(imax);
plot(y1(1:indexmax,1),y1(1:indexmax,2),'b--','LineWidth',2)

```

```

% реверс оси абсцисс
set(gca, 'XDir', 'reverse')
hold on

imax=find(y2(:,1)>0);
indexmax=max(imax);
plot(y2(1:indexmax,1),y2(1:indexmax,2),'g-', 'LineWidth',2)
hold on
grid on
title('Dependence of velocity from height')
ylabel('Velocity v,m/s')
xlabel('height h,m')

```

---

```

fall.m

```

---

```

function dydt = fall(t,x)
global S m g mu

% выбор коэффициента сопротивления
if S < 15
    c=1.2;
else
    c=1.33;
end

% затяжной прыжок, парашют раскрывается через 20 с
% if t>20
%     S=82.5;
% else
%     S=0.7;
% end

ro=1.225;
% плотность воздуха как функция h
% ro=1.225*(1-x(1)/44300)^4.256;

% условный радиус тела
r=sqrt(S/pi);

% коэффициенты сопротивления
k1=6*pi*mu*r;
k2=c*S*ro;
dydt=[-x(2); (m*g-k1*x(2) - k2*x(2)^2)/m];

```

### Программа исследований

1. Теоретически оцените, при какой скорости для падающего вертикально шара в линейном приближении сила сопротивления сравняется с силой тяжести (движение станет равномерным). Используйте выражение  $6\pi r v = mg$ .

2. Теоретически оцените численное значение предельной скорости падения в квадратичном приближении  $v = \sqrt{2mg/cSp}$ , полученной при пренебрежении линейным членом.

3. Для проверки адекватности модели выполните программу исследований с параметрами, заданными в программном коде (масса парашютиста  $m = 120$  кг, площадь парашюта  $S = 82,5$  м<sup>2</sup> (парашют Д-1-5-У для начинающих), начальная высота  $h_0 = 6000$  м, начальная скорость  $v_0 = 0$ ) (рис. 28).

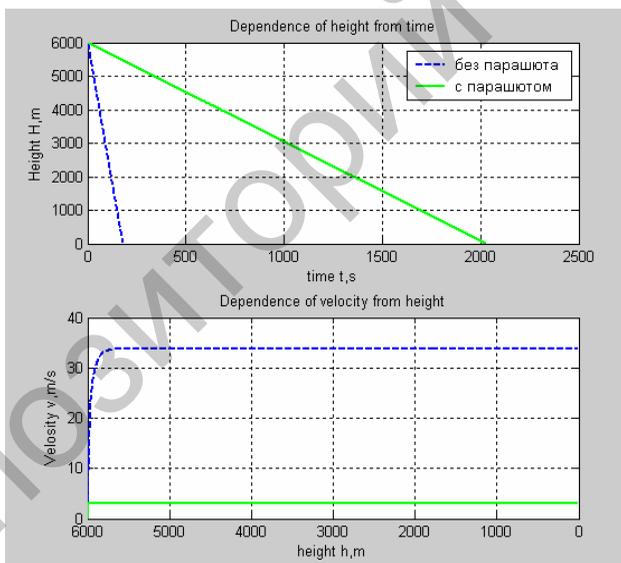


Рис. 28. Результаты компьютерного эксперимента

4. Обратите внимание на разницу во времени достижения поверхности Земли с парашютом и без парашюта. При падении без парашюта парашютист сталкивается с поверхностью на скорости более 30 м/с, что составляет более 100 км/ч.

5. Безопасным считается приземление, при котором скорость приземления  $v \approx 5$  м/с. Проведите компьютерный эксперимент с различными парашютами и оцените скорость приземления.

6. Задайте значение коэффициента  $k_2 = 0$  и рассмотрите случай движения при наличии только линейного сопротивления (рис. 29). Сопоставьте с результатами при  $k_2 \neq 0$ . Сделайте вывод о влиянии сил трения на кинематику и динамику движения парашютиста.

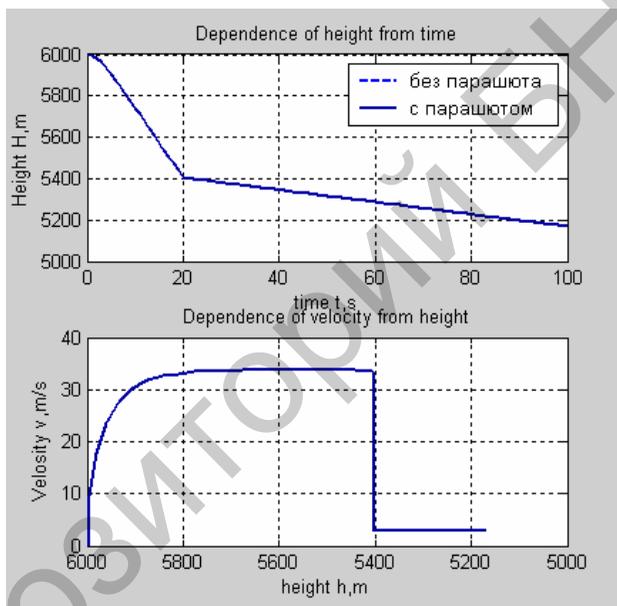


Рис. 29. Результаты компьютерного эксперимента после изменения параметров

7. Рассмотрите случай раскрытия парашюта в затяжном прыжке, например, парашют раскрывается через 20 секунд после начала движения. Для этого в программном коде процедуры fall.m снимите комментарии с соответствующих строк. Обратите внимание на скачок скорости при раскрытии парашюта. Попробуйте оценить ускорение, а значит и перегрузку, испытываемую парашютистом.

8. Уточненные данные показывают, что плотность воздуха не может быть принята постоянной по всей высоте, т.к. зависимость плотности от высоты выражается формулой  $\rho = \rho_0 \cdot (1 - h/44300)^{4,256}$ . На высоте  $h = 6000$  м плотность воздуха составляет  $\rho = 0,6594$  кг/м<sup>3</sup>. Снимите комментарии в программном коде и проведите компьютерные эксперименты с учетом зависимости  $\rho(h)$ . Оцените значение предельной скорости в этом случае.

9. Обратите внимание на табличное значение минимально безопасной высоты парашютиста. На этой высоте парашют уже должен полностью раскрыться, чтобы скорость в момент приземления была безопасной. Задайте начальную высоту 500 м и установите скорость приземления для различных парашютов.

10. Сопоставляя теоретические и экспериментальные данные, приведенные в табл. 8, сделайте вывод об адекватности рассмотренной в данной работе модели.

## МИРОВОЙ РЕКОРД ПО ПРЫЖКАМ В ДЛИНУ

### *Формулировка проблемы*

В 1968 году на олимпийских играх в Мехико Боб Бимон установил мировой рекорд по прыжкам в длину 8,90 м. Результат превысил прежний мировой рекорд на 0,80 м. С 1968 года результат Боба Бимона был превышен только раз – спустя 23 года на соревнованиях в Токио Майк Пауэлл прыгнул на 8,95 м. После прыжка Бимона некоторые предполагали, что выдающийся результат Бимона обусловлен высокогорными условиями. Мехико расположен на высоте 2250 м над уровнем моря. Воздух на этой высоте более разрежен. Исследовать правильность такого утверждения.

### *Физическая модель прыжка*

Гравитационное поле считается однородным. Движение тела прыгуна можно рассматривать и описывать как движение центра масс. Встречный или попутный ветер отсутствует. Сила аэродинамического сопротивления одинакова на всех стадиях прыжка. Подъемная сила отсутствует. Прыжок совершается в двух измерениях. Начальная скорость прыжка и начальный угол одинаковы на высоте Мехико и на уровне моря.

### *Параметры модели*

Начальная скорость движения ( $v_0$ ).

Начальный угол прыжка ( $\theta_0$ ).

Плотность воздуха ( $\rho$ ).

Ускорение свободного падения ( $g$ ).

Площадь поперечного сечения с учетом геометрии фигуры ( $s$ ).

Масса прыгуна ( $m$ ).

Коэффициент аэродинамического сопротивления ( $D$ ).

### *Математическая модель*

Математическая модель прыжка в предположении, что сила аэродинамического сопротивления  $F_c = -Dv^2$  описывается двумя дифференциальными уравнениями второго порядка:

$$\frac{d^2x}{dt^2} = -\frac{D}{m}v_x\sqrt{v_x^2 + v_y^2},$$

$$\frac{d^2y}{dt^2} = -g - \frac{D}{m} v_y \sqrt{v_x^2 + v_y^2}.$$

Здесь

$$\frac{dx}{dt} = v_x \quad \frac{dy}{dt} = v_y, \quad D = \frac{c\rho s}{2}.$$

Эти уравнения заменой переменных сводятся к системе из четырех дифференциальных уравнений первого порядка:

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \frac{d}{dt} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} u_3 \\ u_4 \\ -\frac{D}{m} u_3 \sqrt{u_3^2 + u_4^2} \\ -g - \frac{D}{m} u_4 \sqrt{u_3^2 + u_4^2} \end{pmatrix}. \quad (8)$$

Параметры моделирования имеют следующие значения: ускорение свободного падения  $g = 9,8 \text{ м/с}^2$ , масса прыгуна  $m = 80 \text{ кг}$ , коэффициент  $c = 0,72$ , площадь поперечного сечения  $s = 0,50 \text{ м}^2$ , начальный угол при отталкивании  $\theta_0 = 22,5^\circ = \pi/8$  рад, начальная скорость прыжка  $v_0 = 10 \text{ м/с}$ .

На высоте Мехико плотность воздуха  $\rho = 0,94 \text{ кг/м}^3$ , на уровне моря –  $\rho = 1,29 \text{ кг/м}^3$ .

### **Алгоритм моделирования**

Построим процедуру вычисления правой части системы (8). Создадим функцию `jump.m`.

---

`jump.m` функция вычисления правой части уравнений

---

```
function dzdt = jump(t, z)
global m c s g v0 teta ro
```

```
x = z(1);
y = z(2);
vx = z(3);
```

```

vy = z(4);

xdot = vx;
ydot = vy;
b = c*ro*s/(2*m);
v = sqrt(z(3)^2 + z(4)^2);
vxdot = -b * vx * v;
vydot = -g -b * vy * v;

dzdt =[xdot; ydot; vxdot; vydot];

```

Основной файл `jump_main.m` для обращения к процедуре `ode45` и построения графика имеет вид

---

```

jump_main.m


---


clc
global m c s g v0 teta ro

m = 80;
c = 0.72;
s = 0.5;
g = 9.81;
v0 = 11;
ro = 1.29;
ro = 0.94;

teta = pi/8;
tmax = 1;
dt = [0 tmax];

z0 = [0 0 v0*cos(teta) v0*sin(teta)];

opt = odeset('RelTol', 1e-8, 'AbsTol', 1e-8);
[t z] = ode23(@jump, dt, z0, opt);

figure
plot(z(:,1), z(:,2), 'LineWidth', 2)
title('Зависимость H(x)', 'FontName','Arial Unicode MS')
ylabel('Высота, м', 'FontName','Arial Unicode MS')
xlabel('Расстояние, ', 'FontName','Arial Unicode MS')
grid on

```

На рис. 30 представлена зависимость высоты прыжка от расстояния. Начало отсчета соответствует точке отталкивания.

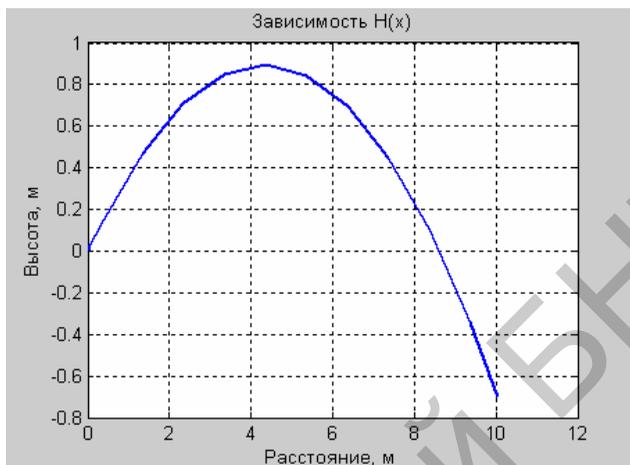


Рис. 30. Зависимость высоты прыжка от расстояния

### **Программа исследований**

1. Определите по графику дальность прыжка.
2. Внесите дополнения в программный код, чтобы исключить вывод на график (или вычисления!!!) значений  $H < 0$ . Значение  $H = 0$  соответствует точке приземления.
3. Исходя из значения начальной скорости обоснуйте взятое для расчета максимальное время  $t$  с.
4. При заданных параметрах измените значение плотности воздуха на значение, соответствующее уровню Мехико. Как изменилась дальность прыжка?
5. Варьируя начальную скорость, добейтесь значения дальности прыжка, соответствующего рекорду Бимона. Сопоставьте полученное значение дальности со значением, полученным при плотности воздуха на уровне моря.
6. Сделайте вывод о том, какой из двух параметров (начальная скорость или плотность воздуха) оказывает наиболее существенное влияние на дальность прыжка.
7. Известно, что с высотой ускорение свободного падения убывает. Обоснуйте, почему в указанном компьютерном эксперименте и на уровне моря, и на высоте Мехико были взяты одинаковые значения  $g$ .

## МЕХАНИЧЕСКИЕ КОЛЕБАНИЯ

Задачи о колебаниях встречаются во всех областях техники. Во многом колебания различных физических объектов сходны друг с другом и описываются одинаковыми дифференциальными уравнениями. Простейшие примеры – колебания маятника и электрические колебания в цепи, состоящей из конденсатора и катушки.

Колебания могут быть *вынужденными*, т.е. совершаемые под действием внешних сил, или *свободными*, совершаемые под действием внутренних сил системы после того, как система была выведена из положения равновесия.

### Формулировка проблемы

Исследовать колебательные процессы на примере пружинного маятника. Построить график зависимости смещения и скорости от времени.

Изучить влияние массы груза  $m$  и жесткости пружины  $k$  на период и амплитуду колебательных процессов. Построить графики влияния массы груза  $m$  и жесткости пружины  $k$  на период и амплитуду колебательных процессов, провести анализ и сделать выводы. Изучить явление резонанса при приближении частоты вынуждающей силы к собственной частоте колебательной системы.

### Физическая модель

Пружина абсолютно упругая, сила трения линейно зависит от скорости, вынуждающая сила гармоническая (рис. 31).

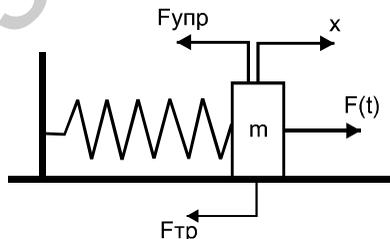


Рис. 31. Модель пружинного маятника

### Параметры модели

Масса груза  $m$ .

Жёсткость пружины  $k$ .

Закон изменения вынуждающей силы.

### Математическая модель

Математическая модель пружинного маятника, совершающего колебания под действием упругой силы пружины  $F_{\text{упр}}$  при наличии в системе силы трения  $F_{\text{тр}}$  и вынуждающей силы  $F$ , составлена на основе второго закона Ньютона и имеет вид

$$m \frac{d^2 x}{dt^2} = -F_{\text{упр}} - F_{\text{тр}} + F.$$

В простейших случаях сила упругости пропорциональна смещению  $kx$ , сила трения пропорциональна скорости  $r \frac{dx}{dt}$ , а вынуждающую силу задают по гармоническому закону:  $F = F_0 \cos(\omega t)$  с частотой  $\omega$ . В этом случае уравнение колебаний пружинного маятника принимает вид

$$m \frac{d^2 x}{dt^2} = -kx - r \frac{dx}{dt} + F_0 \cos(\omega t),$$
$$\frac{d^2 x}{dt^2} = -\frac{k}{m} x - \frac{r}{m} \frac{dx}{dt} + \frac{F_0}{m} \cos(\omega t).$$

### Алгоритмизация математической модели

Заменой переменных сведём дифференциальное уравнение 2-го порядка к системе из двух дифференциальных уравнений первого порядка:

$$\begin{cases} \frac{dx}{dt} = x_1 \\ \frac{dx_1}{dt} = -\frac{k}{m} x - \frac{r}{m} \frac{dx}{dt} - \frac{F_0}{m} \cos(\omega t) \end{cases}.$$

В качестве программного кода создадим два m-файла: vibration.m и vibration\_main.m. Файл vibration.m вычисляет правую часть системы дифференциальных уравнений, файл

vibration\_main.m задает начальные условия, обращается к функции ode45 и строит графики.

---

vibration.m

---

```
function dydt = vibration(t,y)
global k m F0 omega; % объявление глобальных переменных
dydt=[y(2,1); -(k /m)*y(1,1) + F0*cos(omega*t)/m];
```

---

vibration\_main.m

---

```
clc % очистка экрана
clear ; % очистка переменных

global k m F0 omega; % объявление глобальных переменных

% задание начальных параметров
k = 0.1; % коэффициент жесткости пружины
m = 1; % масса груза
F0 = 0; % вынуждающая сила
omega = 1; % частота вынуждающей силы

% цикл для наглядности изменений на графике
%for i=0:1:50;

% при изменении жесткости пружины
% m = m + i;

% при изменении жесткости пружины
% k = k - 0.001 * i;

figure(1);
clf; % очистка графического окна 1

tspan=[0 100]; % интервал интегрирования (время)
y=[ 0; 1 ]; % начальные параметры y
% начальные условия
% y (1,1) - смещение (первая строка, первый столбец)
% y (2,1) - скорость (вторая строка, первый столбец)

% обращение к встроенной функции решения дифференц.
уравнений
[t,y]=ode45('vibration', tspan,y);
```

```

plot(t,y)           % вывод графика
grid on            % включение сетки на графике

% обозначение осей x, y, название графика, легенда
xlabel ('Время, t', 'FontName', 'Arial Unicode MS')
ylabel ('Смещение груза, y(t)', 'FontName', 'Arial Unicode MS')
title('Модель пружинного маятника', 'FontName', 'Arial Unicode MS')
legend ('y (смещение)', 'v(t) (скорость)')

pause (0.001);

% конец цикла
% end;

```

### Программа исследований

1. Задайте  $k=0.1$ ,  $m=1$ ,  $F_0=0$ ,  $\omega=1$ ,  $\Delta t=1..100$ ,  $x_0=0$ ,  $v_0=1$  и начните исследование со случая отсутствия трения и вынуждающей силы. В этом случае маятник будет совершать свободные незатухающие колебания (рис.32). Задайте параметры (все значения параметров даны в системе СИ).

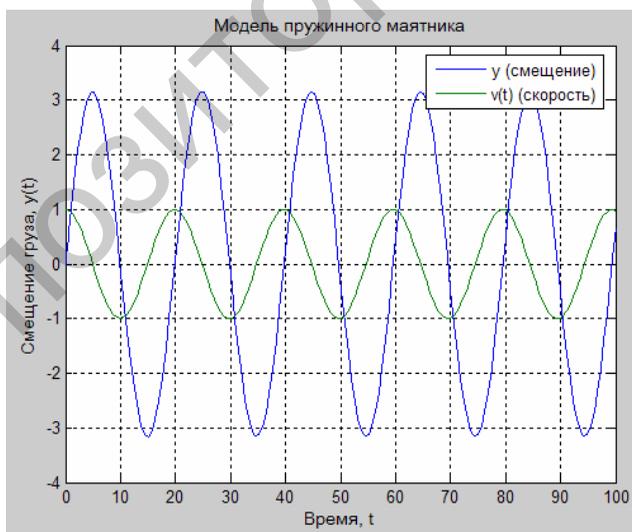


Рис. 32. Свободные незатухающие колебания

2. Измените коэффициент жесткости пружины  $k = 0.3$ . Наблюдайте изменение частоты колебаний. Уменьшите значение коэффициента жесткости  $k = 0.07$ . Как в этом случае изменится частота колебаний? Что будет, если коэффициент жесткости приравнять нулю?

3. Увеличьте массу груза  $m = 3$ . Как влияет изменение массы груза на частоту колебаний (рис. 33)?



Рис. 33. Колебания после изменения параметров

4. Введите в систему силу трения. Задайте значение  $r = 1$  и, варьируя  $r$  (увеличивая, уменьшая в несколько раз) наблюдайте затухание колебаний.

5. Введите в систему вынуждающую силу. Задайте значения  $F_0 = 1$ ,  $\omega = 1$ ,  $r = 0$  и наблюдайте развитие колебательного процесса (переходный процесс). Увеличьте время эксперимента. Что можно сказать об амплитуде колебаний? Приблизьте значение частоты вынуждающей силы к собственной частоте колебательной системы  $\omega_0 = \sqrt{k/m}$ . Задайте  $\omega = 0,9 \cdot \omega_0$ .

6. Введите в систему трение и проанализируйте влияние коэффициента трения на амплитуду колебаний.

## СПЯЩИЙ ПОЛИЦЕЙСКИЙ

### Формулировка проблемы

Автомобиль движется по ровной дороге и наезжает на искусственное препятствие – «спящий полицейский». Исследовать кинематику и динамику движения автомобиля.

### Модель автомобиля

Ограничимся рассмотрением наезда на препятствие только одного колеса. Горизонтальная составляющая скорости автомобиля не меняется. Воздействие препятствия сводится только к возбуждению вертикального перемещения автомобиля. Колесо при движении полностью повторяет профиль препятствия. Подвеска состоит из упругой пружины и демпфера (рис.34).

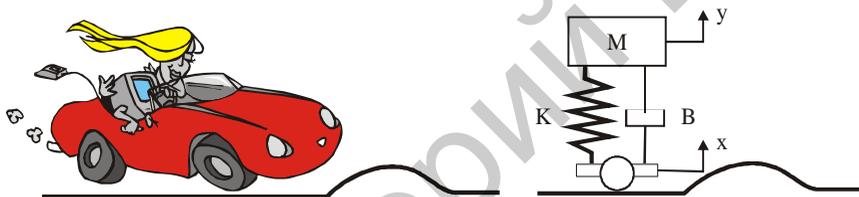


Рис. 34. Модель автомобиля

### Параметры моделирования

Масса автомобиля.

Скорость автомобиля.

Параметры подвески: коэффициент упругости и коэффициент демпфирования.

Параметры препятствия: ширина, высота, профиль.

Упругие свойства и параметры демпфирования колеса.

### Математическая модель

При наезде колеса на препятствие колесо перемещается в вертикальном направлении. Это перемещение описывается переменной  $x$ . На кузов автомобиля воздействие со стороны дороги передается посредством подвески (пружины с жесткостью  $K$  и демпфера с коэффициентом демпфирования  $B$ ). Силовое воздействие посредством пружины определяется относительным смещением кузова, описываемым переменными  $x$  и

$y$ , силовое воздействие со стороны демпфера – относительной скоростью этих перемещений  $dx/dt$  и  $dy/dt$ . В уравнении движения постоянное воздействие на пружину кузова, компенсируемого равной и противоположно направленной силой упругости пружины  $Mg = K\Delta x$ , не будем учитывать.

С учетом сделанных предположений в соответствии со вторым законом Ньютона уравнение движения (математическая модель) рассматриваемой системы имеет вид

$$M \frac{d^2 y}{dt^2} = B \left( \frac{dx}{dt} - \frac{dy}{dt} \right) + K(x - y).$$

$$\frac{d^2 y}{dt^2} + \frac{B}{M} \frac{dy}{dt} + \frac{K}{M} y = \frac{B}{M} \frac{dx}{dt} + \frac{K}{M} x.$$

Вертикальное ускорение кузова автомобиля  $d^2 y/dt^2$  является функцией скорости автомобиля в горизонтальном направлении, т.к. горизонтальное перемещение вследствие неровности сопровождается вертикальным перемещением. При этом профиль дороги (кривизна, ширина, высота) играет существенную роль.

Выберем в качестве математической модели неровности дороги функцию

$$x(s) = \frac{H}{2} \left( 1 - \cos \left( \frac{2\pi s}{L} \right) \right) \quad 0 \leq s \leq L.$$

Вид этой функции для  $H = 10$  см и  $L = 100$  см представлен на рис. 35.

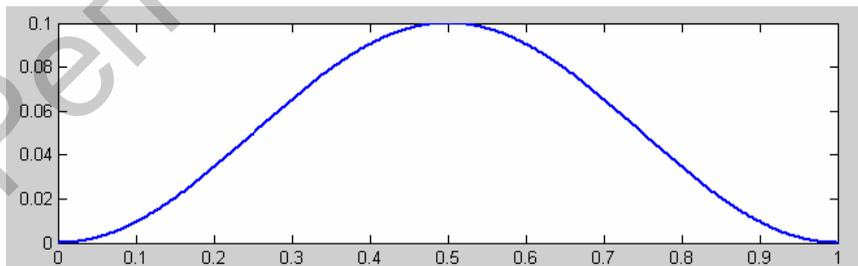


Рис. 35. Модель неровности дороги

Следует обратить внимание на то, что радиус кривизны колеса не должен быть больше радиуса кривизны кривой в каждой точке траектории.

При постоянной скорости автомобиля в горизонтальном направлении  $s = V_0 t$  функция профиля и дороги и ее производная принимают вид

$$x(t) = \frac{H}{2} \left( 1 - \cos \left( \frac{2\pi V_0 t}{L} \right) \right)$$

$$\frac{dx}{dt} = \frac{H}{L} \pi V_0 \sin \left( \frac{2\pi V_0 t}{L} \right)$$

Используем замену

$$y = z_1 \quad \frac{dy}{dt} = \frac{dz_1}{dt} = z_2 \quad \frac{d^2 y}{dt^2} = \frac{dz_2}{dt} = -\frac{B}{M} \frac{dy}{dt} - \frac{K}{M} y + \frac{B}{M} \frac{dx}{dt} + \frac{K}{M} x$$

и приведем уравнение движения к виду, удобному для интегрирования с помощью процедуры `ode45`:

$$\frac{d^2 y}{dt^2} = -\frac{B}{M} \frac{dy}{dt} - \frac{K}{M} y + \frac{B}{M} \frac{dx}{dt} + \frac{K}{M} x$$

$$\begin{cases} \frac{dz_1}{dt} = z_2 \\ \frac{dz_2}{dt} = -\frac{B}{M} \frac{dy}{dt} - \frac{K}{M} y + \frac{B}{M} \frac{dx}{dt} + \frac{K}{M} x \end{cases}$$

$$\frac{d}{dt} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} z_2 \\ -\frac{B}{M} z_2 - \frac{K}{M} z_1 + \frac{B}{M} \frac{dx}{dt} + \frac{K}{M} x \end{pmatrix}.$$

### Алгоритмизация математической модели

Создадим процедуру `polizei.m` для вычисления правой части и m-файл `polizei_main.m`, обращающийся к этой процедуре.

---

polizei.m

---

```
function dzdt = polizei(t, z)
global M K B V0 H L
g = 9.8; % ускорение свободного падения
dzdt = zeros(2,1); % вектор-столбец

s = V0 * t;
x = H / 2 * (1 - cos(2 * pi * s / L));
xdot = H / L * pi * V0 * sin(2 * pi * V0 * t / L);

% неровность дороги локализована в области  $0 \leq s \leq L$ 
if s > L
    x = 0;
    xdot = 0;
end

dzdt(1) = z(2);
dzdt(2) = - B / M * z(2) - K / M * z(1) + B / M *
xdot + K / M * x;
% return
```

---

polizei\_main.m

---

```
clc
global M K B V0 H L
M = 450;
K = 35000;
B = 7300;
z10 = 0;
z20 = 0;

accel = zeros(1000);
i = 1;

V0 = 20;
H = 0.05;
L = 0.8;

%L/V0-время движения по неровности

tmax = 2*L/V0;
z0 = [z10; z20];
dt = [0 tmax];
dt = 0:tmax/500:tmax;
```

```

ii = 0;
aa = zeros(1000,1);

opt = odeset('RelTol', 1e-8);
[t z] = ode45(@polizei, dt, z0, opt);

subplot(2,2,1)
plot(t,z(:,1))
title('displacement')
grid on

subplot(2,2,2)
plot(t,z(:,2))
title('velocity')
grid on

subplot(2,2,3)
plot(t,aa(1:size(t)))
grid on

% построение графика ускорения
[p n] = size(z);
i=1

accel = zeros(1,p);
i=1;
a = ones(p);
tt = 0
zf = zeros(2,1);
zf = z0
while i < p
    a = polizei(tt, zf);
    accel(i) = a(2);
    i = i+1;
    tt = tt + tmax/p;
    zf=[z(i,1); z(i,2)];
end

subplot(2,2,4)
plot(t,accel)
title('acceleration')
grid on

```

## Программа исследований

1. Введите и выполните приведенную программу со следующими начальными условиями.

Таблица 9

Начальная скорость	20 м/с
Высота неровности	5 см
Ширина неровности	80 см
Масса автомобиля	450 кг
Коэффициент упругости подвески	35000 Н/м
Постоянная демпфирования	7300 Н/(м/с)
Скорость вертикального перемещения в момент наезда на препятствие	0 м/с
Вертикальное смещение в момент наезда на препятствие	0 м
Время эксперимента	$L/V_0$

Графики представлены на рис. 36. Для того чтобы получить график ускорения вам предстоит модифицировать приведенный программный код. Во всех последующих экспериментах для наблюдения процесса после проезда автомобилем препятствия варьируйте время (например,  $2L/V_0$ ,  $5L/V_0$ ,  $10L/V_0$ ).

2. Измените массу автомобиля. Увеличьте до 980 кг. Как изменилась амплитуда вертикального перемещения? Скорость? Ускорение?

3. Уменьшите жесткость (упругость) подвески до 20000 Н/м. Сопоставьте графики перемещения, скорости, ускорения.

4. Уменьшите постоянную демпфирования в 3 раза, до нуля. Как изменился процесс при движении по препятствию? После?

5. Особое внимание уделите графику и абсолютному значению ускорения при проезде препятствия. Превышение  $5g$  ( $g = 9,8 \text{ м/с}^2$ ) становится чрезвычайно опасным для жизни!

6. Измените параметры препятствия. Рассмотрите варианты узкого и высокого препятствия, широкого и высокого, широкого и низкого (высота и ширина: 10 см и 40 см, 10 см и 100 см, 4 см и 100 см).

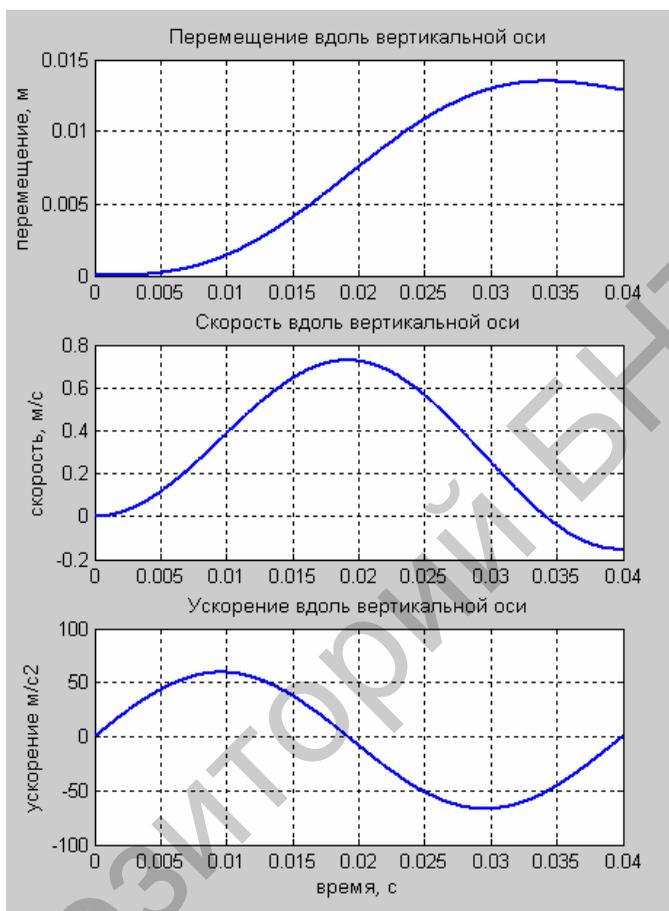


Рис. 36. Результаты компьютерного моделирования

7. Сделайте выводы об условиях безопасного проезда «спящего полицейского».

8. Создайте более совершенную модель. Учтите упругие свойства колеса. В этом случае в модель необходимо добавить еще один демпфер и упругий элемент. Возьмите массу колеса 18 кг, упругую постоянную 70000 Н/м, коэффициент демпфирования 233 Н/(м/с). Сделайте рисунок модели. Составьте математическую модель. Модель должна включать два дифференциальных уравнения второго порядка (одно для кузова, второе для колеса).

## КОЛЕБАТЕЛЬНЫЙ КОНТУР

### Формулировка задачи

Колебательный контур содержит катушку индуктивности, конденсатор, резистор и источник. Рассмотреть электромагнитные процессы в контуре.

### Физическая модель

Рассматривается модель последовательного колебательного контура с сосредоточенными параметрами (рис. 37). Резистор имеет только активное сопротивление, катушка – индуктивное, конденсатор – емкостное. Источник генерирует синусоидальную ЭДС, внутреннее сопротивление источника пренебрежимо мало по сравнению с сопротивлением цепи. Контур не излучает.

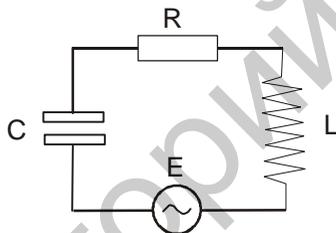


Рис. 37. Модель колебательного контура

### Параметры моделирования

Емкость конденсатора.

Индуктивность катушки.

Сопротивление резистора.

Амплитуда и частота ЭДС источника.

### Математическая модель

В соответствии с законом Кирхгофа алгебраическая сумма падений напряжения на всех элементах контура равна ЭДС источника. Поэтому математическая модель колебательного контура имеет вид

$$L \frac{dI}{dt} + RI + \frac{1}{C} \int Idt = E ,$$

$$L \frac{dI}{dt} + RI + \frac{1}{C} \int Idt = E_0 \cos(\omega t).$$

Для того чтобы избавиться от интегрирования, продифференцируем левую и правую часть уравнения по времени:

$$L \frac{d^2 I}{dt^2} + R \frac{dI}{dt} + \frac{1}{C} I = E_0 \omega \cos(\omega t),$$

$$\frac{d^2 I}{dt^2} + \frac{R}{L} \frac{dI}{dt} + \frac{1}{LC} I = \frac{\omega}{L} E_0 \cos(\omega t),$$

где  $R$  – активное сопротивление резистора, Ом;

$L$  – индуктивность катушки, Гн;

$C$  – емкость конденсатора, Ф;

$I$  – ток в контуре, А;

$E = E_0 \cos(\omega t)$  – ЭДС источника; как функция времени изменяется с частотой  $\omega$  и амплитудным значением  $E_0$ .

Заменой переменных

$$I = z_1 \quad \frac{dI}{dt} = \frac{dz_1}{dt} = z_2 \quad \frac{d^2 I}{dt^2} = \frac{dz_2}{dt} = -\frac{R}{L} \frac{dI}{dt} - \frac{1}{LC} I + E_0 \cos(\omega t)$$

приведем уравнение к системе двух дифференциальных уравнений порядка

$$\frac{dz_1}{dt} = z_2$$

$$\frac{dz_2}{dt} = -\frac{R}{L} dz_2 - \frac{1}{LC} z_1 + \frac{\omega}{L} E_0 \cos(\omega t)$$

$$\frac{d}{dt} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} z_2 \\ -\frac{R}{L} z_2 - \frac{1}{LC} z_1 + \frac{\omega}{L} E_0 \cos(\omega t) \end{pmatrix}.$$

## Алгоритмизация математической модели

Создадим процедуру `contur.m` для вычисления правой части системы дифференциальных уравнений. М-файл `contur_main.m` с использованием функции `ode45` обращается к этой процедуре.

---

```
% contur.m - процедура вычисления правой части
function dzdt = contur(t, z)
global R C L E0 w
dzdt = [z(2); -R/L*z(2) - 1/(L*C)*z(1) + w/L*E0 *
cos(w*t)];
```

---

```
% contur_main.m - основной файл
clc
global R C L E0 w

R = 0.3;
L = 1;
C = 0.25;
z0 = [1; 0];
t0 = 0;
w0 = 1/sqrt(L*C);
T = 2*pi/w0;

E0 = 0;
w = 1;

tmax = 5*T;
dt = [t0 tmax];
[t z] = ode45(@contur, dt, z0);

subplot(3,1,1)
plot(t,z(:,1), 'LineWidth', 2)
title('Ток в контуре', 'FontName','Arial Unicode MS')
ylabel('I, A', 'FontName','Arial Unicode MS')
grid on

subplot(3,1,2)
UL = z(:,2)*L;
plot(t, UL, 'LineWidth', 2)
title('Напряжение на катушке', 'FontName','Arial
Unicode MS')
```

```

ylabel('UL, В', 'FontName','Arial Unicode MS')
grid on
subplot(3,1,3)
UC = z(:,1)*R;
plot(t, UC, 'LineWidth', 2)
title('Напряжение на резисторе', 'FontName','Arial
Unicode MS')
ylabel('UR, В', 'FontName','Arial Unicode MS')
xlabel('время, с', 'FontName','Arial Unicode MS')
grid on

```

Результаты выполнения программы представлены на рис 38.

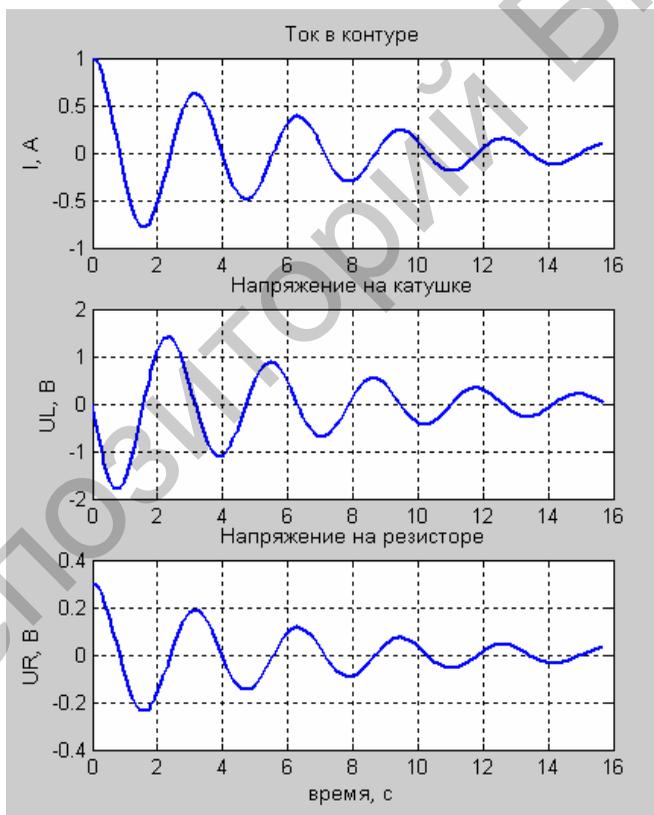


Рис. 38. Электромагнитные процессы в контуре

### Программа исследований

1. При значении  $R = 0.3$  получили затухающий процесс. Задайте величину сопротивления резистора  $R = 0$ . Получите графики и сделайте вывод о влиянии сопротивления резистора на характер колебаний.

2. Увеличьте сопротивление резистора и сделайте вывод о степени затухания в зависимости от величины активного сопротивления.

3. Увеличьте величину индуктивности контура в 4 раза. Как изменилась частота колебаний? Аналогично проведите исследования с емкостью.

4. Задайте сопротивление резистора больше критического  $R_{кр} = 2\sqrt{L/C}$  и наблюдайте превращение затухающего процесса в аperiodический.

5. Включите источник (задайте значение  $E_0 > 0$ ). Наблюдайте переходные процессы в контуре. Постепенно увеличивая время эксперимента  $t_{max}$ , попробуйте получить установившиеся колебания.

6. Постепенно приближайте частоту  $\omega$  изменения ЭДС источника к собственной частоте контура  $\omega_0 = 1/\sqrt{LC}$ . Наблюдайте увеличение амплитуды колебаний (резонанс). Сопоставьте значение ЭДС источника и значение амплитуды напряжения на катушке.

7. Постройте график напряжения на конденсаторе. Для этого внесите дополнения в код программы. Напряжение на конденсаторе определяется выражением  $U_C = \frac{1}{C} \int Idt$ .

8. Постройте график энергии магнитного поля, создаваемого током в катушке  $W_M = LI^2/2$ . Сопоставьте частоту изменения энергии магнитного поля и напряжения на катушке.

## КОЛЕБАНИЯ МЕМБРАНЫ

### Формулировка проблемы

Рассматривается процесс колебания мембраны.

### Физическая модель

Мембрана имеет квадратную форму  $-1 \leq x, y \leq 1$ .

Левая и правая границы мембраны закреплены, верхняя и нижняя границы свободны. Колебания происходят в направлении, перпендикулярном поверхности мембраны. Задана начальная форма мембраны. Задана начальная скорость точек мембраны.

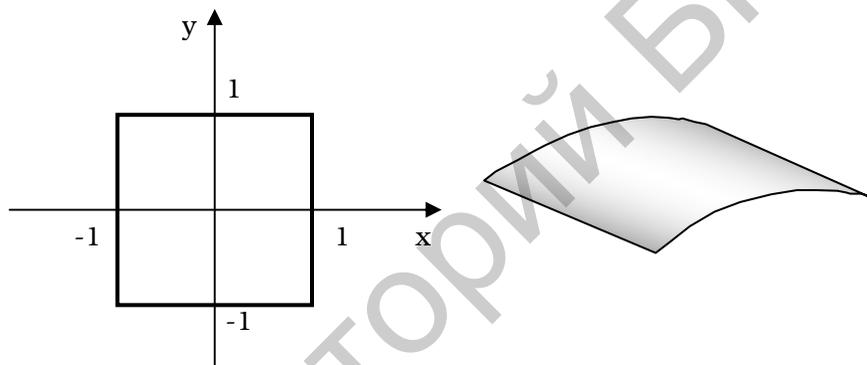


Рис. 39. Модель мембраны

### Параметры моделирования

Коэффициенты  $c, a, f, d$ .

Начальные условия.

### Математическая модель

Колебание мембраны в общем случае описывается уравнением

$$d \frac{\partial^2 u}{\partial t^2} - \nabla \cdot (c \nabla u) + au = f.$$

В нашем случае уравнение имеет вид

$$\frac{\partial^2 u}{\partial t^2} - \operatorname{div}(\operatorname{grad}(u)) = 0.$$

На левой и правой границах задается условие Дирихле

$$u = 0 \quad \text{для} \quad x = \pm 1,$$

на верхней и нижней границах задается условие Неймана

$$\frac{\partial u}{\partial n} = 0 \quad \text{для} \quad y = \pm 1 \text{ (squareb3)}.$$

Задаются начальные условия:

значение функции  $u$  в начальный момент времени:

$$u(0) = \arctan(\cos(\pi x));$$

значение производной функции в начальный момент времени:

$$\partial u(0)/\partial t = 3 \sin(\pi x) \exp(\cos(\pi y)).$$

### **Алгоритмизация математической модели**

Исходное уравнение является уравнением в частных производных гиперболического типа. Будем решать его с помощью функции `hyperbolic`.

Согласно обозначениям в Matlab, коэффициенты уравнения имеют следующие значения:  $c = 1$ ;  $a = 0$ ;  $f = 0$ ;  $d = 1$ .

Область решения – квадрат (геометрия описана в файле `squareg.m`).

Граничные условия описаны в файле `squareb3.m`.

Решение будем искать на временном интервале от 0 до 5 секунд.

Запишем программу в m-файле `membran.m`.

---

```
membran.m
```

```
[p,e,t]=initmesh('squareg');
x=p(1,:); y=p(2,:);
u0=atan(cos(pi/2*x));
```

```

ut0=3*sin(pi*x).*exp(cos(pi*y));
tlist=linspace(0,5,31);
uu=hyperbolic(u0,ut0,tlist,'squareb3',p,e,t,1,0,0,1);
umax=max(max(uu)); umin=min(min(uu)); i=1;
pdeplot(p,e,t,'xydata',uu(:,i),'zdata',uu(:,i), 'mesh',
'on','colorbar','off');
axis([-1 1 -1 1 umin umax]); caxis([umin umax]);

```

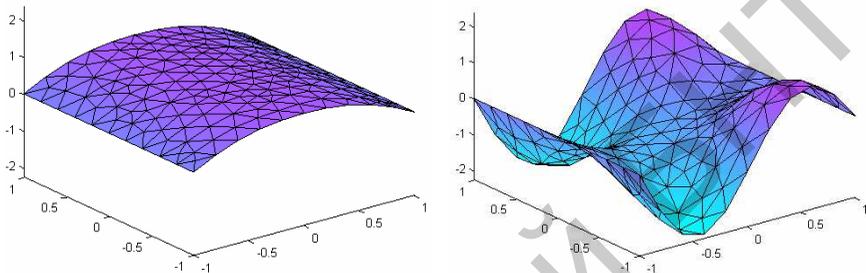


Рис. 40. Колебания мембраны

### ***Программа исследований***

1. Рассмотрите поведение мембраны в различные промежутки времени, изменяя значение индекса  $i = 1$ .
2. Измените начальные условия  $u_0$ .
3. Измените значение коэффициента  $c$ .
4. Введите коэффициент  $a$ , отличный от нуля.

## РАСПРОСТРАНЕНИЕ ТЕПЛА

### Формулировка проблемы

Рассматривается процесс распространения тепла в стенке кирпичной дымовой трубы (рис. 41).

### Физическая модель

Заданы форма и размеры поперечного сечения трубы (кольцо с внешним диаметром  $d_1$  и внутренним  $d_2$ ). Высота трубы намного больше поперечных размеров. Труба выполнена из красного кирпича. Известны физические свойства этого материала (плотность  $\rho$ , коэффициент теплопроводности  $\lambda$  и теплоемкость  $c$ ). Рассмотрим два случая: 1) заданы значения температур на стенках: температура внутренней стенки равна температуре дымовых газов  $T_{\text{дым}}$ , температура наружной стенки равна температуре окружающей среды  $T_{\text{окр}}$ ; 2) задан тепловой поток через наружную стенку трубы. Необходимо найти распределение температур в поперечном сечении трубы.



Рис. 41. К постановке задачи

### Параметры моделирования

Условия на границах сечения.

Физические свойства материала и его параметры.

### Математическая модель

Процесс распространения тепла в твердом теле описывается уравнением с частными производными

$$\rho c \frac{\partial T}{\partial t} = \operatorname{div}(\lambda \operatorname{grad} T) + Q,$$

где  $T$  – температура;

$\rho$  – плотность материала трубы (кирпич);

$c$  – теплоемкость материала;

$\lambda$  – теплопроводность материала;

$Q$  – внутренний источник теплоты.

Для стационарного (установившегося) процесса уравнение принимает вид

$$\operatorname{div}(\lambda \operatorname{grad} T) + Q = 0.$$

В нашем случае нет внутренних источников теплоты, т.е.  $Q = 0$ .

Чтобы однозначно определить процесс, необходимо задать граничные условия:

*1-й случай.* Условие Дирихле для внешней границы  $\Omega_1$   $T|_{\Omega_1} = T_{\text{окр}}$  и внутренней  $\Omega_2$   $T|_{\Omega_2} = T_{\text{дым}}$ .

*2-й случай.* Условие Неймана для внешней границы  $\lambda \frac{\partial T}{\partial n} + \alpha(T - T_{\text{окр}})|_{\Omega_1} = 0$ , где  $\alpha$  – коэффициент теплоотдачи.

Для нестационарного процесса необходимо задать значение температуры в начальный момент времени  $T_0$ .

### **Алгоритмизация математической модели в Matlab**

Для решения дифференциальных уравнений с частными производными, определенных на двумерных областях, предназначен пакет расширения Matlab PDE (Partial Differential Equations Toolbox).

Пакет позволяет пройти все шаги решения задачи в визуальном режиме.

1. Запустить модуль PDETool из окна Launch Pad системы MATLAB (либо дать команду `pdetool` в командном окне).

Появится графическое окно PDE Toolbox ([рис. 11](#)).

2. Определение геометрии области решения задачи.

Конструируем область, используя кнопки на панели инструментов. Первые две позволяют создать прямоугольник (от угла, от центра), а третья и четвертая – эллипс. С помощью пятой можно сконструировать область, имеющую форму многоугольника.

Двойным щелчком мыши по заданной области можно перейти в окно Object Dialog и установить точные размеры и координаты, дать имя фигуре.

Из геометрических примитивов можно составить сложную область, задав в строке Set Formula их комбинацию. Знак плюс означает объединение объектов, а минус – вычитание.

Нарисуем два круга радиуса  $r_1 = 0,6$  и  $r_2 = 0,4$  с центром в начале координат. Кольцо зададим формулой  $E1 - E2$  (рис. 42).

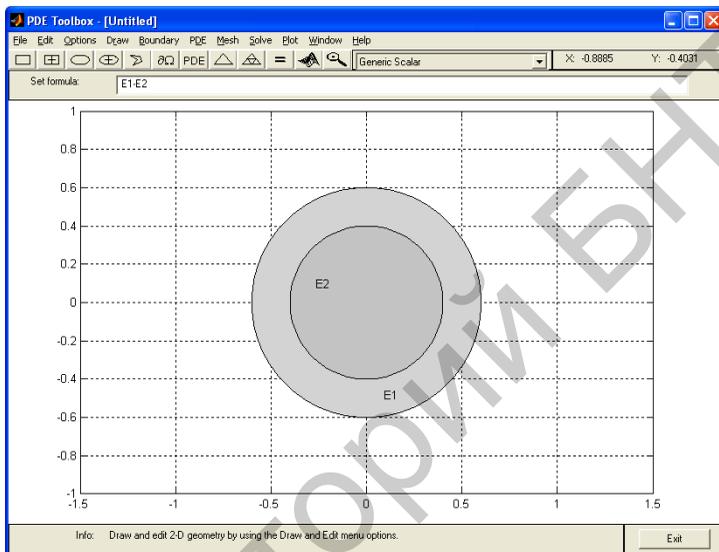


Рис. 42. Конструирование области решения задачи

3. Определение уравнения. Из раскрывающегося списка выбрать тип решаемой задачи. Пункт Heat Transfer соответствует задаче о переносе тепла.

Далее следует задать коэффициенты дифференциального уравнения. Для этого нужно перейти в режим дифференциального уравнения, выбрав пункт PDE Mode в меню PDE или нажав кнопку PDE (рис. 43).

Выбор эллиптического типа уравнения соответствует стационарному, а параболического – нестационарному уравнению теплопроводности. Выберем тип Elliptic.

Введем значения коэффициентов уравнения

$$\text{div}(k \cdot \text{grad}(T)) = Q + h \cdot (T - T_{\text{ext}}),$$

$$Q = 0, \quad h = 0, \quad k = \lambda = 0,77.$$

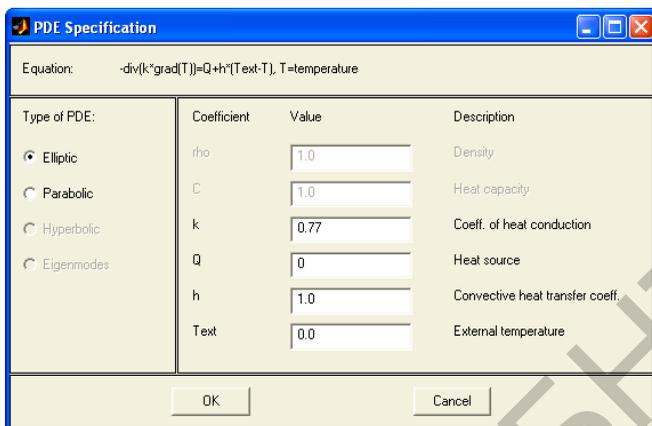


Рис. 43. Ввод коэффициентов уравнения

4. Задание граничных условий. Нажав кнопку  $\partial\Omega$  (Boundary Conditions), переходим в режим установки граничных условий. Для каждого сегмента границы можно определить свои условия. Четыре дуги окружности выделяем с нажатой клавишей Shift. После двойного щелчка для выделенного участка границы открывается окно диалога Boundary Conditions. Устанавливаем для внешней границы значение температуры, равное 30 (условие Дирихле), для внутренней – 400.

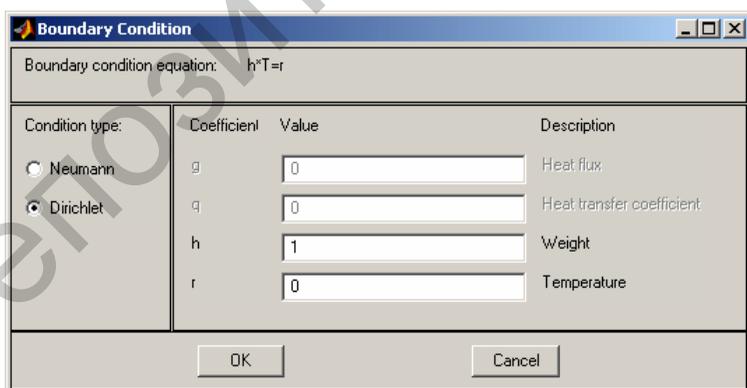


Рис. 44. Задание граничных условий

5. Триангуляция области. После нажатия кнопки  область покрывается сеткой, состоящей из треугольников. Для уменьшения размеров сетки следует нажать на кнопку Refine Mesh.

6. Решение. Решение задачи производится выбором пункта Solve PDE меню Solve или нажатием на кнопку . Найденное распределение температуры отображается в окне PDE Toolbox.

7. Параметры графика устанавливаются в окне Plot Selection.

Изменение геометрии области, граничных условий, типа уравнения и его коэффициентов может быть выполнено, даже если решение уже найдено. По команде File\Save As можно сохранить геометрию области, тип и коэффициенты уравнения и граничных условий.

### **Программа исследований**

1. Решите построенную модель со следующими параметрами: наружный радиус трубы  $r_1 = 0,6$  м, внутренний радиус трубы  $r_2 = 0,4$  м;

коэффициент теплопроводности кирпича  $\lambda = 0,77$  Вт/(м·°С);

плотность кирпича  $\rho = 1700$  кг/м<sup>3</sup>;

теплоемкость кирпича  $c = 880$  Дж/(кг·°С);

температура наружного воздуха  $T_{\text{окр}} = 30$  °С;

температура дымовых газов  $T_{\text{дым}} = 400$  °С.

2. Измените граничные условия на наружной стенке трубы. Задайте условие Неймана  $n \cdot k \cdot \text{grad}(T) + q \cdot T = g$ . Коэффициент теплоотдачи  $q = \alpha$  задайте равным 20 Вт/(м<sup>2</sup>·°С).

3. Поднимите температуру дымовых газов до 600 °С.

4. Пусть труба сильнее обдувается ветром. Увеличьте коэффициент теплоотдачи до 30 Вт/(м<sup>2</sup>·°С).

5. Модифицируйте модель процесса. Рассмотрите его как нестационарный.

Нажмите кнопку PDE и измените тип уравнения на параболический. В окне диалога Solve Parameters положите начальную температуру равной температуре окружающей среды. Найдите распределение температуры в моменты времени 10, 100, 1000, 10000.

## МОДЕЛИРОВАНИЕ В ПОДСИСТЕМЕ SIMULINK

В состав системы MATLAB входит подсистема Simulink – интерактивная среда для моделирования и анализа широкого класса динамических систем, использующая графический язык блок-диаграмм. Моделирование основывается на компонентах – графических блоках, которые содержатся в библиотеках и являются основой для создания модели в виде графического представления – блок-схемы. Такое представление автоматизирует один из наиболее трудоемких этапов моделирования – алгоритмизацию и решение сложных систем алгебраических и дифференциальных уравнений, функционально описывающих данную модель.

Компоненты моделей являются графическими блоками и моделями, которые содержатся в ряде библиотек и с помощью мыши могут переноситься в основное окно и соединяться друг с другом необходимыми связями. В состав моделей могут включаться источники сигналов различного вида, виртуальные регистрирующие приборы, графические средства анимации и др.

На всех этапах работы в Simulink практически нет необходимости заниматься обычным программированием. Добавление в модель любого нового блока, задание его параметров (свойств модели) и соединений приводит к автоматической генерации и изменению кода.

Simulink – это специальный инструмент исследования, который не требует от пользователя знания языков программирования, позволяет приобрести визуальные ассоциации и уяснить аналогии между структурой математического описания объекта и структурой блок-схемы, реализующей данную модель. Визуальные блоки облегчают восприятие причинно-следственных отношений между цепочкой математических понятий: независимая переменная, дифференциальное уравнение, вынуждающая функция, начальные условия, решение дифференциального уравнения и соответствующей цепочкой понятий «аналогового» моделирования – время моделирования, входной сигнал, структурная схема объекта, начальные условия на интеграторах, выходной сигнал.

Рассмотрим решение проблемы моделирования на примере колебательной системы (рис. 45), состоящей из груза, упругого элемента и гасителя колебаний (демпфера). Груз подвергается действию силы.

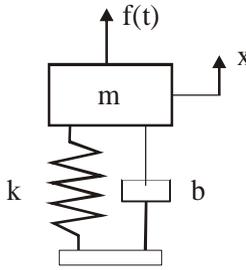


Рис. 45. Модель колебательной системы

Модель описывается дифференциальным уравнением второго порядка

$$m \frac{d^2 x}{dt^2} = -b \frac{dx}{dt} - kx + f(t), \quad (9)$$

где  $m$  – масса груза;

$x$  – смещение;

$\frac{dx}{dt}$  – скорость;

$\frac{d^2 x}{dt^2}$  – ускорение;

$f(t)$  – действующая сила;

$b$  – коэффициент демпфирования;

$k$  – коэффициент упругости.

Преобразуем уравнение движения модели к виду, который является основой для дальнейшего использования в Simulink:

$$\frac{d^2 x}{dt^2} = \frac{1}{m} \left[ -b \frac{dx}{dt} - kx + f(t) \right]. \quad (10)$$

### Запуск Simulink

В командном окне MATLAB наберите Simulink и нажмите клавишу «Ввод» (рис. 46).

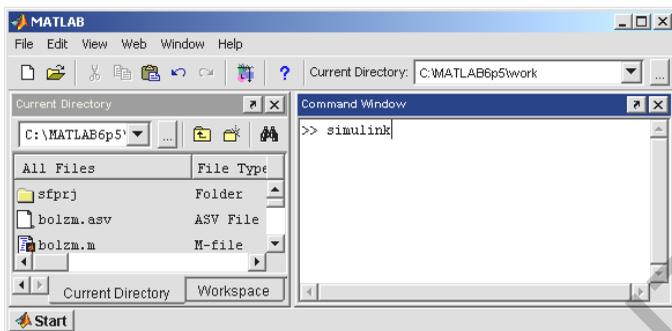


Рис. 46. Запуск Simulink

Появится окно браузера библиотек Simulink (рис. 47).

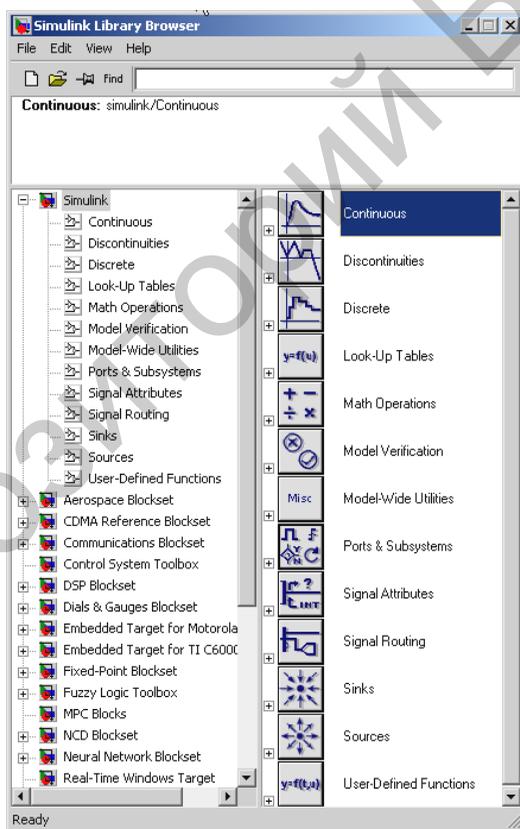


Рис. 47. Окно браузера библиотек

Последовательным выполнением File > New > Model создайте новое окно для модели (рис. 48).

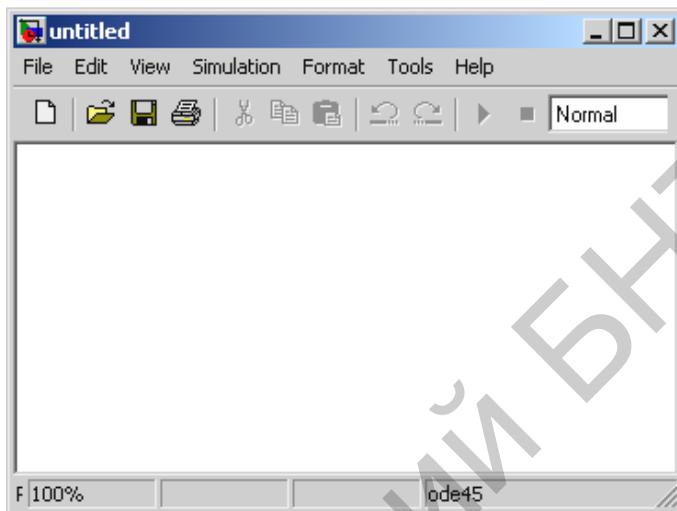


Рис. 48. Новое окно модели

Рассмотрим процедуру представления уравнения движения модели к блочному виду.

В уравнении (10) в правой части в квадратных скобках суммируются три выражения – три сигнала: сила –  $f(t)$ , производная смещения  $\frac{dx}{dt}$  и смещение  $x$ . При этом и  $\frac{dx}{dt}$ , и  $x$  умножаются на константы. Сумма в квадратных скобках также умножается на  $1/m$ .

Для составления модели понадобится блок суммирования Sum с тремя входами для сложения  $f(t)$ ,  $\frac{dx}{dt}$  и  $x$ , три блока уси-

ления Gain для умножения  $b\frac{dx}{dt}$ ,  $kx$  и  $\frac{1}{m}\left[-b\frac{dx}{dt} - kx + f(t)\right]$ . Для выполнения процедуры интегрирования – два блока интегрирования Integrator и блок визуализации Scope.

Таблица 10

Библиотека	Символ и название в библиотеке	Название блока на схеме	Математическое представление	Функциональное назначение
Math Operations	 Sum		$-b \frac{dx}{dt} - kx + f(t)$	Суммирование правой части уравнения (10)
Math Operations	 Gain	$kx$	$\times k$	Умножение входного сигнала на коэффициент упругости
Math Operations	 Gain	$bx'$	$\times b$	Умножение входного сигнала на коэффициент демпфирования
Math Operations	 Gain		$\frac{1}{m}$	Правая часть уравнения (10) делится на массу
Source	 Step	$f(t)$	$f(t)$	Генерирует силу
Sinks	 Scope	Scope		Визуализация сигнала $x(t)$
Continuous	 Integrator	Integrator1	$\frac{d^2x}{dt^2} \rightarrow \int \rightarrow \frac{dx}{dt}$	Интегрирование второй производной смещения
Continuous	 Integrator	Integrator	$\frac{dx}{dt} \rightarrow \int \rightarrow x$	Интегрирование первой производной смещения

В библиотеке Sources найдем источник сигнала для силы Step, в библиотеке Math Operations – блок для усиления сигнала Gain, для суммирования используем блок Sum из этой же библиотеки.

Удерживая левую кнопку мыши, перетащим символ блока Step из окна библиотеки Simulink – Sources в окно модели (рис. 49).

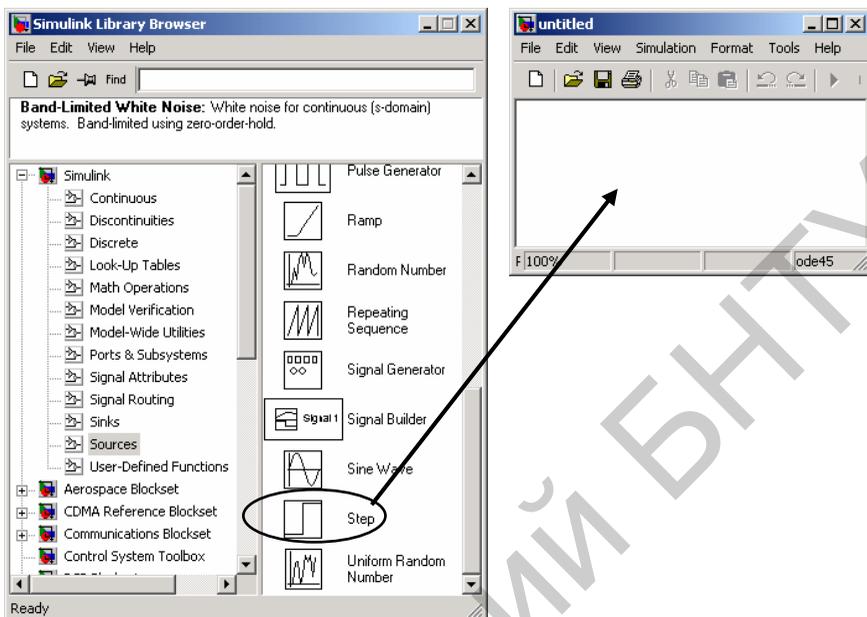


Рис. 49. Перетаскивание блока из библиотеки

Аналогично будем поступать и с другими блоками. Каждый блок имеет точки входа и выхода, указанные стрелками. Контекстное меню помогает изменять параметры блоков.

Используя контекстное меню (щелчок правой кнопки по блоку Sum), изменим параметры блока Sum (рис. 50).

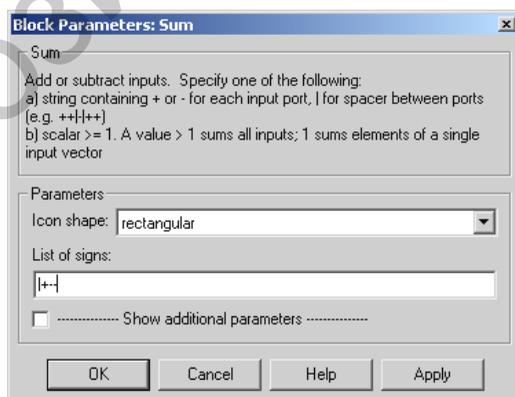


Рис. 50. Окно параметров блока Sum

Изменим форму блока `round` на `rectangular`, знак « + » заменим на знак « - » и добавим еще один знак « - » (еще одна точка входа). Знаки « + » и « - » соответствуют знакам слагаемых в математической модели.

Для изменения названия блока `Step` на  $f(t)$  достаточно щелкнуть мышью по слову `Step` и редактировать его.

Будем использовать для обозначения производной –  $x'$ , второй производной –  $x''$ .

Используя мышью, соединим блоки линиями (рис. 51). Щелчком по названию блока изменим имена блоков в соответствии с их назначением.

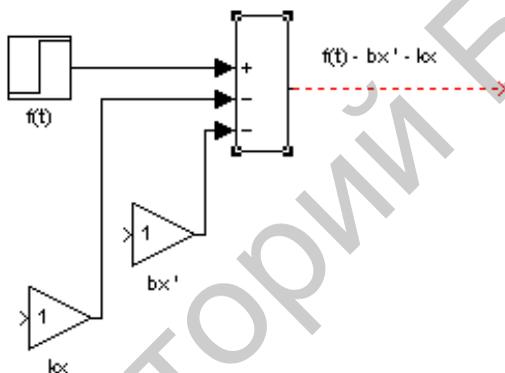


Рис. 51. Суммирование входных сигналов

Выражение в квадратных скобках умножается на  $1/m$ . Поэтому к выходу с блока суммирования подключим еще один блок умножения. На выходе с этого блока получаем вторую производную (рис. 52).

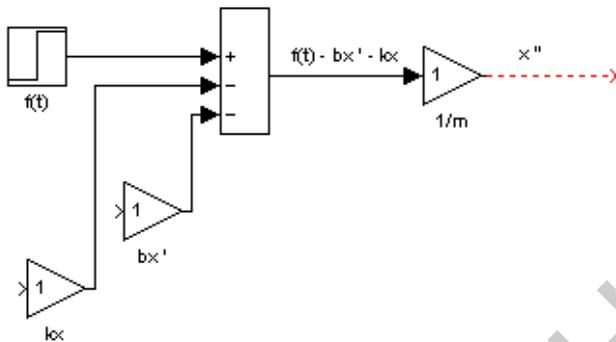


Рис. 52. Применение блока Gain к результату суммирования

Нас интересует смещение  $x$ . Чтобы получить смещение проинтегрируем сигнал  $x''$  последовательно два раза, используя блок интегрирования Integrator из библиотеки Continuous. После первого интегрирования получаем сигнал – первую производную  $x'$ , после второго – искомое смещение  $x$  (рис. 53).

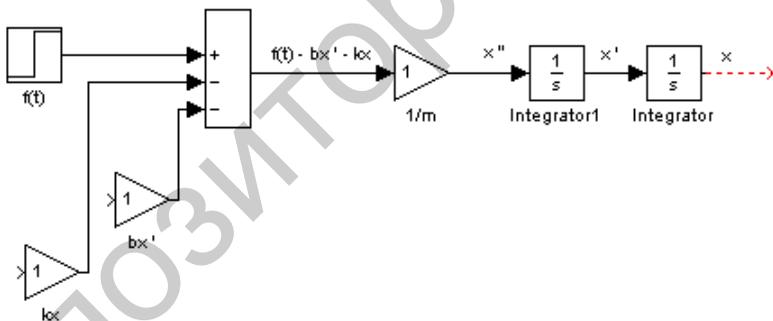


Рис. 53. Интегрирование выходного сигнала (дважды) для получения  $x$

Используя контекстное меню, повернем блоки умножения для демпфирования и упругости на 180 градусов и соединим с соответствующими выходами после интеграторов.

Для визуализации смещения  $x$  добавим блок Scope из библиотеки Sinks (рис. 54).

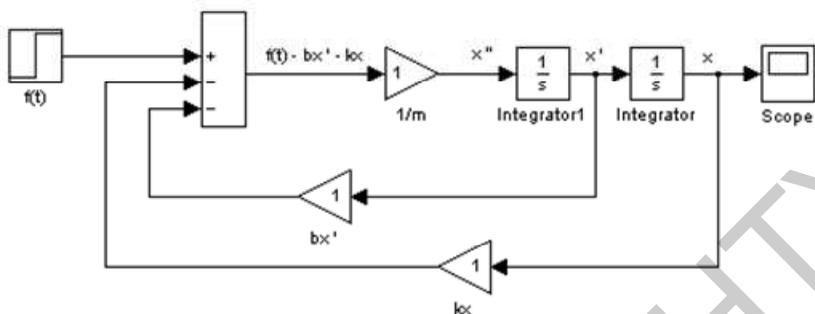


Рис. 54. Окончательная модель с визуализацией смещения

Формирование функциональной схемы модели из блоков завершено. Далее необходимо задать численные значения параметров моделирования. Будем использовать значения, представленные в табл. 11.

Таблица 11

Step блок	Step time	0
	Initial value	0
	Final value	1
Gain блок-масса	Gain	1
Gain блок-демпфирование	Gain	5
Gain блок-упругость	Gain	100
Integrator интегрирование (оба)	Initial condition	0

Значения параметров модели вводятся после двойного щелчка по блоку в соответствующие окна. В результате получаем следующую модель, представленную на рис. 55.

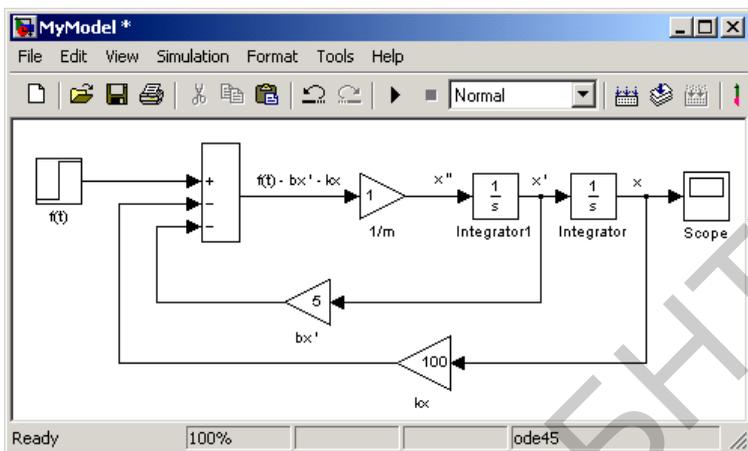


Рис. 55. Окончательная модель с численными значениями параметров

Запуск модели осуществляется из меню Simulation > Start.

Для наблюдения графика смещения используйте двойной щелчок по блоку Scope (рис. 56). Используйте иконку Winocular для автоматического масштабирования графика. Имеется также дополнительная возможность изменения масштаба с использованием лупы.

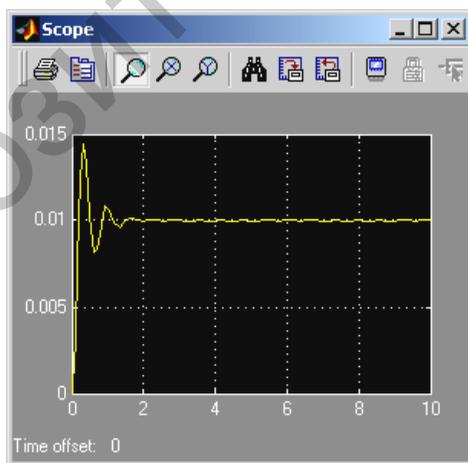


Рис. 56. Окно осциллографа Scope

Сохраните созданную модель стандартным способом через меню File. Присвойте файлу имя MyModel.mdl. Обратите внимание, что расширение файла модели Simulink задается как .mdl.

### **Программа исследований**

1. Запустите модель на выполнение Simulation > Start с заданными параметрами. Двойным щелчком по Scope и щелчком по Binocular наблюдать график смещения.

2. Измените параметры колебательной системы. Увеличьте (уменьшите) в два, четыре, десять раз коэффициент упругости. Как изменился вид графика смещения? Сделайте анализ и выводы.

3. Увеличьте (уменьшите) в два, четыре, десять раз коэффициент демпфирования. Как изменился вид графика смещения? Сделайте анализ и выводы.

4. Подключите дополнительно источник визуализации Scope для того, чтобы наблюдать график первой производной, второй производной. В какое место блок-схемы вы его должны подключить?

5. Исследуйте эффект влияния вида вынуждающей силы на характер колебательных процессов. Удалите источник Step и замените его на Sine из той же библиотеки. Варьируя частотой источника вблизи значения  $\omega_0 = \sqrt{k/m}$ , наблюдайте изменения характера колебаний. Что происходит при приближении частоты вынуждающей силы (источника) к собственной частоте колебательной системы  $\omega_0$ ?

6. Электрический контур, содержащий резистор, конденсатор и источник ЭДС, может быть описан дифференциальным уравнением

$$\frac{du}{dt} + \frac{1}{RC}u = \frac{1}{RC}u_0(t),$$

где  $u(t)$  – выходное напряжение;

$\frac{du}{dt}$  – скорость изменения выходного напряжения;

$R$  – сопротивление резистора;

$C$  – емкость конденсатора;  
 $u_0(t)$  – входное напряжение.

Создайте в Simulink модель контура (рис. 57). Исследуйте зависимость выходного напряжения от произведения  $RC$ . В качестве начального значения рассмотрите значение  $RC = 100$ .

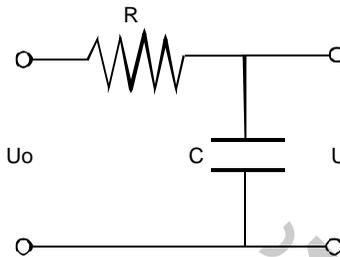


Рис. 57. Модель электрического контура

7. Замените источник Step на источник синусоидального сигнала Sine. Проведите моделирование с численными значениями частоты  $RC = 500$  при частотах входного напряжения 10 и 200 Гц. Сопоставьте амплитуду выходного сигнала (напряжения) в обоих случаях. Оцените численные значения выражения  $20 \log_{10} \left( \frac{\text{амплитуда } U(t)}{\text{амплитуда } U_0(t)} \right)$  в децибелах. Если вы получили, что для частоты 200 Гц это значение в несколько раз меньше, чем для 20 Гц, это означает, что такая схема выполняет функции фильтра нижних частот.

8. Составьте модель, которая описывает прыжки на упругом канате, описанные ранее. Выясните для себя функциональное назначение каждого блока. Блок ограничения Saturation представляет нелинейное устройство и требует для решения данной модели задать нижнее значение Lower limit = 0. Верхнее значение Upper limit задайте очень большим. На выходе этого блока сигнал равен входному в пределах ограничения. Функциональный блок Fcn содержит функцию, описывающую квадратичную зависимость силы сопротивления от скорости.

9. Установите для себя назначение блока с константой 50.

10. Перед запуском модели задайте в командной строке MATLAB

```
>> k=40; m=90; g=9.8; a1=1; a2=1
```

В этом случае Simulink будет использовать указанные значения для переменных. Преимущество такого подхода заключается в том, что блоки содержат, как это видно из рис. 58, переменные и выражения с переменными, а не их значения.

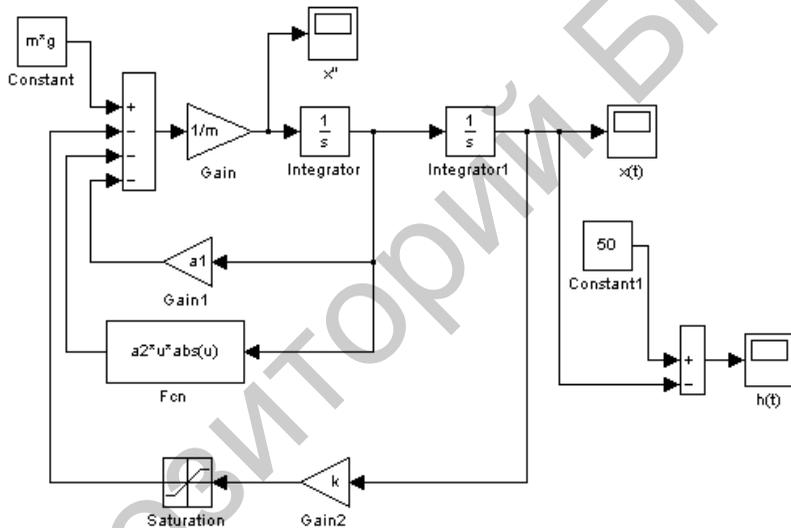


Рис. 58. Модель контура в Simulink

## Список использованных источников

1. Самарский, А.А. Математическое моделирование: Идеи. Методы. Примеры/ А.А. Самарский, А.П. Михайлов. – М.: Наука; Физматлит, 1997. – 320 с.
2. Советов, Б.Я. Моделирование систем: учебник для вузов/ Б.Я. Советов, С.А. Яковлев.– 3-е изд. – М.: Высшая школа, 2001.
3. Советов, Б.Я. Моделирование систем. Курсовое проектирование: учебное пособие для вузов / Б.Я. Советов, С.А. Яковлев. – М.: Высшая школа, 1988.
4. Самарский, А.А. Математическое моделирование/ А.А. Самарский, А.П. Михайлов. – М.: Физматлит, 2001.
5. Ануфриев, И.Е. MATLAB 7/ И.Е. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова. – СПб.: БХВ-Петербург, 2005.
6. Дьяконов, В.П. MATLAB: учебный курс/ В.П. Дьяконов. – СПб.: Питер. – 2001.
7. Дьяконов, В.П. MATLAB 6.5 SP1/7 + Simulink 5/6(r). Основы применения/ В.П. Дьяконов. – М.: СОЛОН-Пресс, 2005.
8. Дьяконов, В.П. MATLAB 6.5 SP1/7 + Simulink 5/6(r) в математике и моделировании. Серия «Библиотека профессионала» / В.П. Дьяконов. – М.: СОЛОН-Пресс, 2005.
9. Кетков, Ю.Л., MATLAB 7: программирование, численные методы / Ю.Л. Кетков, А.Ю. Кетков, М.М. Шульц. – СПб.: БХВ-Петербург, 2005.
10. Потемкин, В.Г. Система инженерных и научных расчетов MATLAB 5.x: в 2 т / В.Г. Потемкин. – М.: ДИАЛОГ-МИФИ, 1999.
11. Говорухин, В. Компьютер в математическом моделировании: учебный курс/ В.Говорухин, В. Цибулин. – СПб.: Питер, 2001.
12. Васильков, Ю.В. Компьютерные технологии вычислений в математическом моделировании: учебное пособие/ Ю.В.Васильков, Н.Н. Василькова. – М.: Финансы и статистика, 2001.
13. Половко, А.М. MATLAB для студента/ А.М. Половко, П.Н. Бутусов. – СПб.: БХВ-Петербург, 2005.
14. Краков, М.С. Численные методы и визуализация данных в среде MATLAB: методическое пособие/ М.С. Краков, И.В. Никифоров. – Минск, 2001.

15. Сидорик, В.В. Математическое моделирование в среде Matlab: учебно-методическое пособие / В.В. Сидорик, С.Г. Погирницкая. – Минск: БНТУ, 2008. – 112 с.

16. Математический энциклопедический словарь / под ред. Ю.В. Прохорова. – М.: Советская энциклопедия, 1988.

17. Бахвалов, Н.С. Численные методы/ Н.С. Бахвалов. – М.: Наука, 1973.

18. Марчук, Г.И. Методы вычислительной математики/ Г.И. Марчук. – М.: Наука; Физматлит, 1989.

19. Львовский, Е.Н. Статистические методы построения эмпирических формул/ Е.Н. Львовский. – М.: Высшая школа, 1988.

20. Повх, И.А. Техническая гидромеханика/ И.А. Повх. – Л.: Машиностроение, 1976. – 502 с.

21. Идельчик, И.Е. Справочник по гидравлическим сопротивлениям/И.Е. Идельчик – М.: Машиностроение, 1975.–559 с.

22. Исаченко, В.П. Теплопередача: учебник для вузов/ В.П. Исаченко, В.А. Осипова, А.С. Сукомел. – 4-е изд., перераб. и доп. – М.: Энерго-издат, 1981.

23. Краснощеков, Е.А. Задачник по теплопередаче: учебное пособие для вузов/ Е.А. Краснощеков, А.С. Сукомел. – М.: Энергия, 1980.

24. Роуч, П. Вычислительная гидродинамика/ П. Роуч. – М.: Мир, 1980. – 616 с.

25. Пасконов, В.М. Численное моделирование процессов тепло- и массообмена /В.М. Пасконов, В.И. Полежаев, Л.А. Чудов. – М.: Наука, 1984. – 288 с.

26. Дульнев, Г.Н. Применение ЭВМ для решения задач теплообмена/ Г.Н. Дульнев, В.Г. Парфенов, А.В. Сигалов. – М.: Высшая школа, 1990.

27. Патанкар, С. Численные методы решения задач теплообмена и динамики жидкости/ С. Патанкар. – М.: Энергоиздат, 1984.

28. <http://www.math.duke.edu/education/ccp/materials/diffeq/predprey/contents.html>

29. <http://www.math.duke.edu/education/ccp/materials/diffeq/>

30. Тарасевич, Ю.Ю. Математическое и компьютерное моделирование/ Ю.Ю. Тарасевич. – М.: УРСС, 2003.

31. Шеннон, Р. Имитационное моделирование систем – искусство и наука/ Р. Шеннон. – М.: Мир, 1978.

32. Айзерман, М.А. Классическая механика/М.А. Айзерман. – М.: Наука, 1980. – 368 с.

33. Амелькин, В.В. Дифференциальные уравнения в приложениях/ В.В. Амелькин. – М.: Наука, 1987. – 160 с.

34. Краснощеков, П.С. Принципы построения моделей/ П.С. Краснощеков, А.А. Петров – М.: Изд-во МГУ, 1983. – 264 с.

35. Математическое моделирование / под ред. Дж. Эндрюса, Р. Мак-Лоуна; пер. с англ. – М.: Мир, 1979. – 278 с.

36. Попов, Ю.П. Вычислительный эксперимент/ Ю.П. Попов, А.А. Самарский. – М.: Знание, 1983. – 64 с.

37. Ризниченко, Г.Ю. Математические модели биологических репродукционных процессов/ Г.Ю. Ризниченко, А.Б. Рубин. – М.: Изд-во МГУ, 1993. – 300 с.

38. Самарский, А.А. Введение в численные методы/ А.А. Самарский. – М.: Наука, 1982. – 272 с.

39. Седов, Л.И. Механика сплошной среды в 2 т. /Л.И. Седов. Т. 1. – М.: Наука, 1973. – 536 с.

40. Седов, Л.И. Методы подобия и размерности в механике/ Л.И. Седов. – М.: Наука, 1981. – 448 с.

41. Bender, E.A. An Introduction to Mathematical Modelling/ E.A. Bender. – N.Y.: Wiley, 1978.

42. Cross, M., Learning the Art of Mathematical Modelling/ M. Cross, A.O. Moscardini. – N.Y.: Wiley, 1985. – 154 p.

43. Dym, C.L., Principles of Mathematical Modelling/ C.L.. Dym, E.S. Ivey. – N.Y.: Academic Press, 1980. – 256 p.

44. Jacoby, S.L.S. Mathematical Modelling with Computers/S.L.S. Jacoby, J.S. Kowalik. – Englewood Cliffs, N.J.: Prentice Hall, Inc., 1980. – 292 p.

45. Lehman, R.S. Computer, Simulation and Modelling: An Introduction / R.S. Lehman. – N.Y.: Wiley, 1977.

46. Дэбни, Дж. Simulink 4. Секреты мастерства /Дж. Дэбни, Т. Харман. – М.: Бином. Лаборатория знаний, 2003. – 404 с.

47. Поршнев, С.В. MATLAB 7. Основы работы и программирования: учебник / С.В. Поршнев. – М.: Бином. Лаборатория знаний, 2006. – 320 с.

48. Иглин, С.П. Математические расчеты на базе Matlab / С.П. Иглин. – М.: BHV-Санкт-Петербург, 2005. – 649 с.

49. Гандер, В., Решение задач в научных вычислениях с применением Maple и MATLAB/ В. Гандер, И. Гржебичек. – М.: Вассамедина, 2005. – 520 с.

50. Новгородцев, А. Расчет электрических цепей в MATLAB / А.Новгородцев. – СПб.: Питер, 2004.

51. Черных, И. Simulink. Среда создания инженерных приложений / И. Черных. – М.: Диалог-МИФИ, 2004.

52. Kwon, Y.W. The Finite Element Method using MATLAB / Y.W. Kwon. – Boca Raton a. o.: CRC Press, 1997.

Репозиторий БНТУ