Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

ВЫЧИСЛИТЕЛЬНАЯ ФИЗИКА

сборник учебно-методических материалов

для направления подготовки

03.03.02 – Физика

Благовещенск 2017

Печатается по решению
редакционно-издательского совета
инженерно-физического факультета
Амурского государственного
Университета

Составители: Нещименко В.В., Юрина В.Ю.
Вычислительная физика: сборник учебно-методических материалов для направления под-готовки 03.03.02. – Благовещенск: Амурский гос. ун-т, 2017.

© Амурский государственный университет, 2017
© Кафедра физики, 2017
© Нещименко В.В., Юрина В.Ю., составление

ВВЕДЕНИЕ

Целью дисциплины «Вычислительная физика (Практикум на ЭВМ)» является получение основ классификации и методик построения математических моделей физических явлений, освоение основных принципов программных реализаций используемых аналитических или численных методов, а также анализ решения физической задачи, полученной в математических терминах.

Основными задачами дисциплины является формирование у студентов представлений о методах исследования физических объектов или процессов (построение физической модели), математическое описание задачи; выбор метода решения и исследования задачи (построение математической модели), разработка и выбор оптимального алгоритма решения конкретных задач; обработка и анализ полученных результатов (проведение вычислительного эксперимента), корректировка способа решения при наличии особенностей задачи, анализ вопроса устойчивости и сходимости метода решения, оценка границ применимости построенной математической модели.

Для формирования умений и навыков в учебной программе дисциплины предусмотрены практические работы, а также самостоятельная работа студентов, роль которой в настоящее время в системе высшего образования значительно возросла и является формой самообразования.

В процессе обучения происходит закрепление и систематизация знаний, углубление теоретических знаний, развитие умений работать с различными источниками информации и как результат – освоения основных компетенций.

Дисциплина «Вычислительная физика (Практикум на ЭВМ)» относится к дисциплинам вариативной части учебного плана данной образовательной программы.

1 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ И ЛАБОРАТОРНЫМ РАБОТАМ

Лабораторный практикум выполняется по индивидуальному графику бригадами, состоящими из 2 студентов. График выполнения лабораторных работ формируется преподавателем в начале каждого семестра и представляется студентам на первом аудиторном занятии лабораторного практикума. Выполнение лабораторных работ предполагает значительную самостоятельную работу как на этапе предварительной подготовки к работе, так и при выполнении работы, оформлении отчета и подготовки к «защите» работы.

1.1 Подготовка к выполнению лабораторной работы

Подготовка к лабораторной работе осуществляется студентом *до аудиторных занятий,* в часы, отведенные на самостоятельную работу (см. пункт 2.3).

Студент обязан приходить на занятие подготовленным. Наличие «заготовки» к лабораторной работе является обязательным условием допуска студента к выполнению лабораторной работы.

Студенты, не имеющие подготовки, к выполнению лабораторной работы не допускаются.

1.2 Выполнение лабораторной работы в лаборатории

На выполнение каждой лабораторной работы отводится 2 часа аудиторного времени, в это время включается: получение допуска к работе, выполнение необходимых измерений и «защита» работы выполненной на предыдущем занятии.

Перед выполнением работы преподаватель проверяет степень подготовленности каждого студента. Для этого студент должен предоставить «заготовку» отчета в индивидуальном лабораторном журнале ответить на следующие вопросы:

– Какова цель экспериментальной задачи? Каковы основы теории изучаемого явления, основные понятия и формулы?

– Каков принцип работы экспериментальной установки? Перечислите основные этапы эксперимента.

Получив допуск к выполнению лабораторной работы, студент должен ознакомиться с измерительными приборами, используемыми в процессе выполнения работы, получить у лаборанта необходимое дополнительное оборудование, подготовить оборудование к проведению эксперимента согласно методическому руководству. После чего предъявить подготовленное к работе оборудование для проверки лаборанту или преподавателю. Только после получения разрешения от преподавателя или лаборанта можно приступать к выполнению измерений.

При выполнении работы следует соблюдать правила техники безопасности, обращаться с приборами и оборудованием следует бережно и аккуратно, применять приборы только в соответствии с их назначением.

Выполнив все измерения, выключить установку, предъявить преподавателю результаты измерений для проверки. Если при записи результатов или в ходе эксперимента была допущена ошибка, опыт повторяется вновь. Если результаты удовлетворительны, преподавателем делается отметка о выполнении студентом лабораторной работы (ставится подпись и дата в отчете студента).

Отчеты без подписи преподавателя в дальнейшем к «защите» не принимаются.

После подписи результатов преподавателем, студенту необходимо привести лабораторную установку в исходное состояние, сдать лаборанту выданное дополнительное оборудование и привести в порядок рабочее место.

1.3 Оформление отчета и подготовка к «защите» лабораторной работы

Оформление отчета и подготовка к «защите» лабораторной работы осуществляется студентом в часы, отведенные на самостоятельную работу (см. пункт 2.3).

После оформления отчета студент готовится к «защите» лабораторной работы, изучая теоретические основы данной темы, ориентируясь на контрольные вопросы, приведенные в методических указаниях. Для получения зачета по лабораторной работе студент представляет преподавателю оформленный отчет со всеми необходимыми расчетами и «защищает» его в ходе последующего собеседования.

1.4 Предлагаемые к выполнению лабораторные и практические работы и методические указания к ним

Практическая работа № 1. Основные приемы работы с пакетом Matlab. Арифметические операции. Встроенные скалярные функции.

Matlab является интерактивным, матрично-ориентированным пакетом, предназначенным для выполнения научных и инженерных расчетов. Название пакета MATLAB происходит от MATrix LABoratory (Матричная лаборатория). Познакомимся с необходимыми сведениями для начала работы с пакетом. Запустите пакет. Командное окно пакета находится в одном окне, а редактор текста – в другом.

Во время работы с системой можно пользоваться встроенной оперативной помощью, которая содержит дополнительную информацию о командах, опциях и примерах. Команда **help <имя_группы>** выведет на экран список функций, размещенных в этой группе с краткими пояснениями. Команда lookfor позволяет выполнить поиск m-функции по ключевому слову, при этом анализируется первая строка комментария и она же выводится на экран, если в ней встретилось ключевое слово.

Matlab можно использовать как обыкновенный калькулятор, который выполняет основные арифметические операции: + (сложение), - (вычитание), * (умножение), / (деление), ^ (возведение в степень) и содержит набор констант (pi. e, i).

Пример.

```
>> (2+3*pi) / 2
ans =
          5.7124
```

Для более сложных вычислений в пакете предусмотрены дополнительные операции. Кратко перечислим основные функции, имеющиеся в пакете. Описание других можно найти, используя справку в диалоговом режиме.

Функция	Описание	Функция	Описание
abs ()	абсолютное значение числа	**cos ()**	cos(x)
sin ()	sin(x)	**tan ()**	tg(x)
exp ()	e^x	**log()**	ln(x) - натуральный логарифм
acos()	arccos(x)	**sqrt()**	Vx - извлечение кв. корня
atan ()	arctg(x)	**round()**	Округление числа до ближайшего целого

ceil()	Наибольшее целое для данного действительного числа	**floor()**	Наименьшее целое число для данного действительного числа
rem ()	Остаток от деления двух чисел	**sign()**	Знак числа (1, если знак + и -1, если знак-)

Matlab является интерпретирующим языком непосредственных вычислений. Операторы пакета обычно имеют форму:

```
>> имя_переменной=выражение
или просто
>> выражение
```

Выражение формируется из операторов, функций и имен переменных. Если имя переменной и знак равенства в левой части отсутствуют, автоматически генерируется переменная **ans**, которой присваивается результат вычислений. Matlab различает строчные и прописные буквы в именах команд, функций и переменных!

Например, для присвоения выражениям различных имен используется знак равенства.

```
>> s=3-floor(exp(3.3))
s =
      -24
```

Точка с запятой позволяет задать несколько выражений и вывод значений результата после нажатия клавиши <Enter> не производится:

```
>> d=sin (5.6);                    Задание d
>> a=cos (6.5);                    Задание a
>> a+d                             Вывод на экран a+d
ans =
    0.3453
```

Поскольку все вычисления в Matlab выполняются с двоичной точностью, формат вывода может изменяться с помощью следующих команд:

Название формата	Описание формата представления числа
format short	С фиксированной точкой и 4 знаками после точки (по умолчанию)
format long	С фиксированной точкой и 14 знаками после точки
format short e	Научная нотация с 4 десятичными знаками

format long e	Научная нотация с 15 десятичными знаками
format short g	С фиксированной точкой и 3 знаками после точки
format long g	Научная нотация с 13 десятичными знаками

Например,

```
>> format long
>> 3*cos(1.2)
ans =
1.08707326343002
```

Практическая работа № 2. Создание матриц. Операции с матрицами и массивами. Векторные и матричные функции

Matlab работает с одним видом объектов – числовыми прямоугольными матрицами, элементами которых могут быть в общем случае комплексные числа. Все переменные представляют собой матрицы, матрицы 1×1 интерпретируются как скаляры, матрицы с одной строкой или одним столбцом - как вектора. В системе Matlab матрицы могут быть созданы разными способами.

1. Введены явно с помощью списка элементов.
2. Сгенерированы встроенными операторами или функциями.
3. Созданы в т-файлах.
4. Загружены из внешнего файла данных.

Например, в результате выполнения оператора

```
>> A=[1 2 3; 3 2 1; 4 5 6]
или
>> A=[1 2 3
      3 2 1
      4 5 6]
```

Matlab создает матрицу 3×3 и присваивает ее значение переменной A.

```
A =
       1   2   3
       3   2   1
       4   5   6
```

В пакет Matlab встроен ряд функций, позволяющих создавать матрицы специального вида. Приведем некоторые из них.

1. Функция **rand (n)** создает матрицу размера n×n (функция **rand (m,n)** - матрицу размера m×n), каждый элемент которой - случайное число с равномерным законом распределения в диапазоне [0,1].

2. Функция **magic (n)** создает матрицу размера n×n, которая является магическим квадратом.

3. Функция **zeros (m, n)** создает нулевую матрицу размера n×n.

4. Функция **ones (m, n)** создает матрицу размера m×n, каждый элемент которой равен единице.

5. Функция **diag (x)** создает матрицу, у которой на главной диагонали стоят элементы вектора x.

6. Функция **diag (A)** создает матрицу, у которой на главной диагонали стоят диагональные элементы матрицы *A*.

Ссылки на отдельные элементы матриц и векторов осуществляются с помощью индексов в круглых скобках.

Приведем несколько примеров задания матриц, выделения элементов и подматриц.

```
>> Z=zeros (4, 5);
```
Создание нулевой матрицы размера 4×5

```
>> 0=ones (3, 2) ;
```
Создание единичной матрицы размера 3×2

```
>> X=0 : 0.5 : 2
```
Вывод на экран вектора X размера 1×5

```
X =
   0   0.5000   1.0000   1.5000   2.0000
>> Y=sin (X);
```
Создание вектора Yразмера 1 x5, каждый элемент которого является sin от X

```
>> A (2 , 3)
ans = 1
>> A (8)
ans = 1
```
Выделение элемента (2,3) из матрицы A Альтернативное выделение элемента(2,3) как 8-го по счету, если записать матрицу в один вектор-столбец

```
>> A (1 : 2 , 1 : 3)
ans =
        1  2  3
        3  2  1
>> A ( [1  2] , [1  3] )
ans =
          1    3
          3    1
```
Выделение подматриц A

```
>> A (2 , 2) =round (tan (8) )
ans =
          1    2    3
          3   -7    1
          4    5    6
```
Замена элемента матрицы A

```
>> A (: , 2)
ans =
         2
        -7
         5
```
Выделение 2-го столбца

```
>> A (2 , :)
ans =
         3   -7   1
```
Выделение 2-й строки

В пакете Matlab доступны следующие матричные операции: + (сложение), - (вычитание), * (умножение), ^ (возведение в степень, применима только для квадратных матриц!), ' (транспонирование), / (правое деление), \ (левое деление). Если размерность матриц не соответствует выполняемой операции, то система генерирует сообщение об ошибке. Приведенные операции могут стать поэлементными, если перед ними поставить точку. Приведем примеры использования матричных операций.

```
>> D= [1  2 ; 4  8] ;
>> C=D'
C =
         1     4
         2     8
>> 5*(D*C) ^ 2
ans =
    2125   8500
    8500   34000
>> C ^ 2                    Произведение матриц C*C
ans =
        9     36
        18    72
>> C. ^ 2                   Квадрат каждого элемента матрицы C
ans =
         1    16
         4    64
>> sin (D. / 2)
ans =
         0.4794     0.8415
         0.9093     0.7568
```

Часто возникает необходимость использования векторных или матричных функций.

Аргументами векторных функций являются векторы (строки или столбцы). Если в качестве аргумента указана матрица размера m×n, то данная функция действует постолбцово, т.е. результатом действия является вектор-столбец, каждый элемент которого – результат действия этой функции на соответствующий столбец. Например, максимальный элемент прямоугольной матрицы находится помощью команды **max (max (A))**. Наиболее употребительными матричными функциями являются:

inv (X) – обратная матрица,

det (X) – определитель матрицы,

size (X) – размерность матрицы,

norm (X) – норма вектора или матрицы,

rank (X) – ранг матрицы,

cond (X) – число обусловленности, ...

Циклы. Условные операторы и операторы отношения Операторы управления Matlab и операторы отношения при использовании работают также, как и в большинстве языков программирования.

Оператор цикла **for** в общем виде записывается как:

for <переменная_цикла>=<выражение_цикла>

<Выполняемые_операторы>

end

Оператор цикла **while** записывается следующим образом:

while <условие>

<Выполняемые_операторы>

end

Условный оператор **if** имеет следующий синтаксис:

if <условие>

<Выполняемые_операторы1>

else

<Выполняемые_операторы2>

end

В пакете Matlab возможно также множественное ветвление.

В пакете Matlab используются следующие операторы отношений и логические операторы:

= =	Равно	~	Не
~ =	Не равно	&	И
<	Меньше	|	ИЛИ
>	Больше		
<=	Меньше или равно		
>=	Больше или равно		

Приведем несколько примеров использования циклов и условий.

Пример. Найдем значение 5!.

```
>> a=1;
>> for i=l : 5
a=a*i;
end
>> a
a =
     120
```

Пример. Для различных значений x, полученных делением на 3 чисел от 0 до 10, выведем значения квадратов и кубов этих чисел.

```
>> п=10;
>> к=0;
>> while  к<=n
x=к / 3;
disp ( [x   x^2   x^3])
к=к+1;
end
```

Пример. В зависимости от *s* присвоим переменной *a* разные значения.

```
>> s=10;
>> if (s<0) |(s>10)
a=s^2+5;
else
a=0;
end
>> a
a =
   0
```

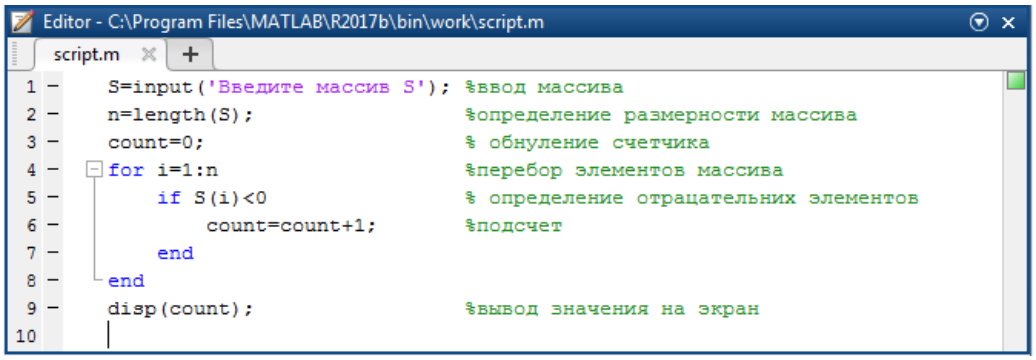
М-файлы

Matlab может последовательно выполнять последовательность операторов, записанных в файл на диске. Имена таких файлов имеют вид **<имя>.m** и такие файлы называются m-файлами. Большая часть работы в Matlab заключается в создании, редактировании и выполнении таких m-файлов. Существуют два типа m-файлов: файлы-программы (сценарии) и файлы-функции.

Практическая работа № 3. Файлы-программы

Файлы-программы состоят из последовательности обычных операторов пакета Matlab. Если m-файл с таким сценарием имеет имя, например, start.m, то команда **start**, введенная в командной строке, вызовет выполнение последовательности операторов этого файла. Переменные в программе являются глобальными и могут изменить значения переменных с теми же именами в командном окне. В таких файлах легко исправить ошибки, внутри файла можно ссылаться на другие m-файлы и рекурсивно на этот же файл.

Приведем пример программы, которая подсчитывает число отрицательных элементов массива, сохраненной под именем script.m.



Вызов в командной строке

```
>> script
```

приведет к выполнению операторов программы:

```
>> script
Введите массив S [2  -7  4  -8  2  -56]
          3
```

Файлы-функции

Файлы-функции дают возможность расширять Matlab, поскольку определенные пользователем функции имеют тот же статус, что и другие функции Matlab. Переменные по умолчанию являются локальными и не влияют на имена и значения переменных в текущей рабочей области Matlab.

Общий вид задания:

function [xl, x2 ,...] =name (y1, y2 ,...)

<тело функции>

Первая строка функции содержит объявление выходных аргументов, имени функции и входные аргументов, без этой строки файл является программой, но не функцией.

Совпадение имени функции и имени файла является обязательным условием в Matlab.

Символ % указывает на то, что вся строка после него является комментарием и игнорируется при выполнении программы.

Приведем пример функции, которая производит подсчет количества элементов массива, кратных заданному числу, и вычисляет сумму таких элементов.

Editor - C:\Program Files\MATLAB\R2017b\bin\work\fun.m

script.m | fun.m

```
function[ml,S]=fun(P,q)   %Задание функции
n=length(P);              % Определение размерности массива P
ml=0;                     %Обнуление счетчиков
S=0;
for i=1:n
    if rem(P(i),q)==0     %Поиск кратных числу q элементов
        ml=ml+1;          %Подсчет количества таких элементов
        S=S+P(i);         %Подсчет их сумм
    end
end
```

Вызов в командной строке (после задания массива P):

```
>> [m, S] = fun (P, 3)
m =
     2
S =
     12
```

присвоит переменным *m* и *S* значения выходных параметров.

Для вычисления значения функции, имя которой передается в функцию через строковую переменную, часто используется функция `feval`, аргументами которой являются имя функции и координата точки, в которой вычисляется значение данной функции. Приведем пример использования функции `feval`. Зададим свою функцию Ra изменений аргумента.

Editor - C:\Program Files\MATLAB\R2017b\bin\work\Ra.m

med.m | Ra.m

```
function z=Ra(x)
z=x^3-2*x+x+5;
```

И пусть, например, требуется обратиться к этой функции med в другом файле-функции, вычисляющем значение функции в середине указанного отрезка:

```
function f=med(func,a,b)
c=(a+b)/2;
f=feval(func,c);
```

Результат обращения к функции med будет иметь вид:

```
>> med('Ra',-5,8)

ans =

    6.8750

>>
```

Текстовые строки. Сообщения об ошибках

Текстовые строки вводятся в Matlab в виде текста в одинарных кавычках: m=’Mistake’ Вывод текстовой строки осуществляется при помощи оператора **disp**. Оператор disp (m) выведет на экран сообщение:

Mistake

Сообщение об ошибках лучше выводить при помощи функции error, так как после обращения к данной функции выполнение m-файла будет прекращено. Для выхода из цикла (и из программы) можно воспользоваться командой **break**.

В m-файле запрос на ввод данных можно организовать при помощи оператора **input**. При вводе оператора `m=input('введите размерность массива')` на экран выводится запрос и выполнение работы файла приостанавливается до ввода требуемых данных с клавиатуры. После нажатия клавиши <Enter> данные присваиваются переменной m.

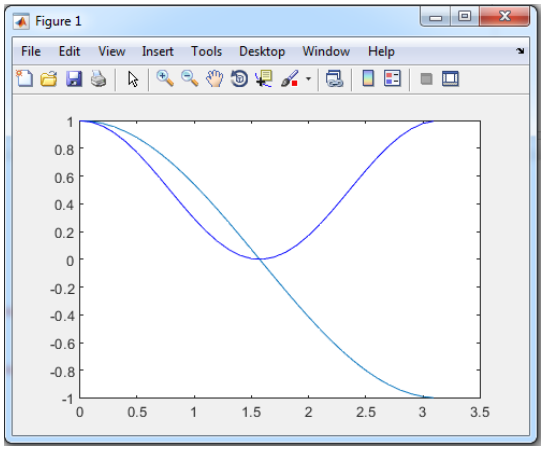
Пакет Matlab позволяет строить двух- и трехмерные графики кривых и поверхностей. Дополнительные возможности и описание графиков можно найти, используя справку в диалоговом режиме.

Практическая работа № 4. Построение графиков

Команду `plot` используют для построения графика двумерной функции. Следующий пример иллюстрирует, как построить графики функций $y = cos(x)$ и $y = cos^2(x)$ на интервале [0, π].

```
>> x=0:0.1:pi;
>> y=cos(x);
>> z=cos(x).^2;
>> plot(x,y,x,z,'b')
```

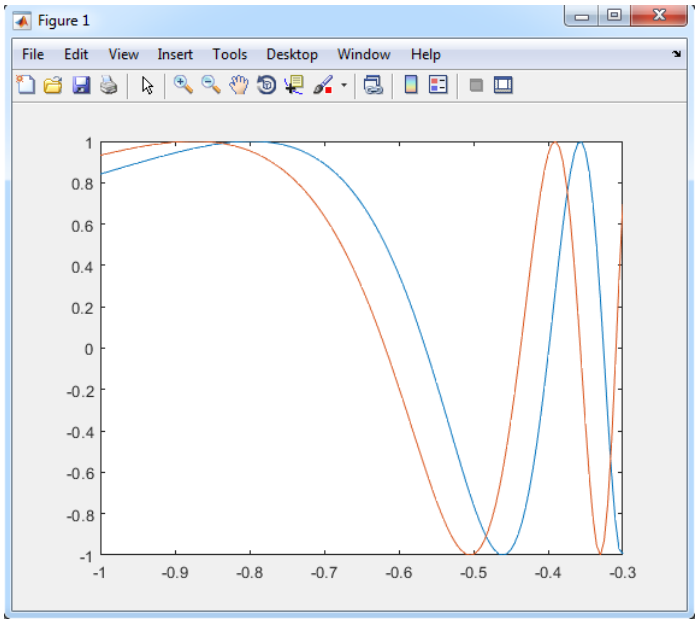
В первой строке программы задается область с шагом 0.1. В следующих двух строках задаются две функции. Первые три строки заканчиваются точкой с запятой - это предотвращает вывод матриц x, y, z на экран. Команда plot строит графики сплошной линией синего цвета.



Сравнение нескольких функций удобно производить, отобразив их графики на одних осях. Например, построим на отрезке [-1,-0.3] графики функций

$f(x) = sin\left(\frac{1}{x^2}\right)$, $g(x) = sin\left(\frac{1.2}{x^2}\right)$.

```
>> x=-1:0.005:-0.3;
>> f=sin(x.^-2);
>> g=sin(1.2*x.^-2);
>> plot(x,f,x,g)
```



Команда `plot'` позволяет легко задать стиль (тип маркера и тип линий) и цвет линий, например,

```
>> plot(x,f,'k-',x,g,'k:')
```

осуществляет построение первого графика сплошной черной линией, а второго - черной пунктирной.

В таблице приведены возможные значения аргументов с указанием результата.

Цвет		Тип маркера		Тип линии	
y	желтый		точка	-	сплошная
m	розовый	o	кружок	:	пунктирная
c	голубой	×	крестик	-.	штрихпунктирная
r	красный	+	знак «плюс»	—	штриховая
g	зеленый	*	звездочка		
b	синий	s	квадрат		
w	белый	d	ромб		
k	черный	v,∧,<,>, ∨	треугольники		
		p,h	звезды		

Удобство оформления графиков зависит от дополнительных элементов оформления: координатной сетки, подписей к осям, заголовка и легенды. Сетка наносится командой `grid on`, функции `xlabel`, `ylabel` служат для размещения подписей к осям, `title` - для заголовка, `legend` - сопровождение легендой.

Графики функций двух переменных

Для отображения функции двух переменных следует:

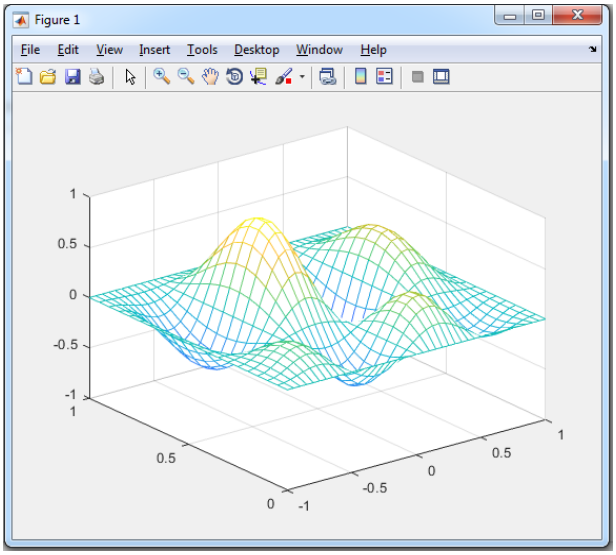
1. Сгенерировать матрицы с координатами узлов сетки на прямоугольной области определения функции.

2. Вычислить функцию в узлах сетки и записать полученные значения в матрицу.

3. Использовать одну из встроенных функций пакета.

Сетка генерируется функцией `meshgrid`, вызываемой с двумя входными аргументами:

```
>> [X,Y]=meshgrid(-1:0.05:1, 0:0.05:1);
>> Z=4*sin(2*pi*X).*cos(1.5*pi*Y).*(1-X.^2).*Y.*(1-Y);
>> mesh(X,Y,Z)
```



Функция `surf` строит каркасную поверхность графика функции и залвает каждую клетку определенным цветом, зависящим от значения функции в точках, соответствующим углам клетки.

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

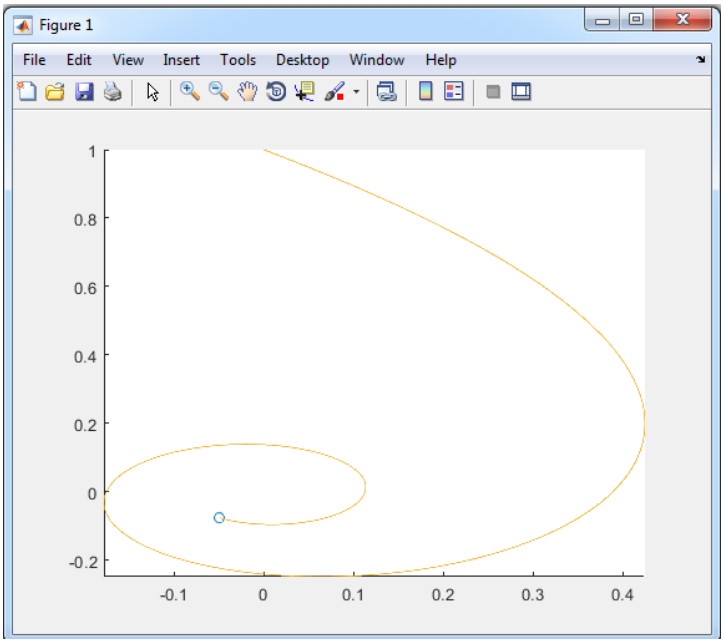
Команда `shading flat` позволяет убрать каркасные линии, а команда `shading interp` позволяет получить поверхность плавно-залитую цветом. Команда `colorbar` выводит рядом с графиком цветовую шкалу. Линии уровня можно построить с помощью команд `meshc, surfc`, которые имеют те же входные аргументы, что и `mesh, surf`.

Анимированные графики

При изучении движения точки на плоскости или в пространстве полезно не только построить траекторию точки, но и следить за движением точки по этой тра-

ектории. Для построения анимированных графиков применяются функции `comet` и `comet3`.

```
>> t=[0:0.001:10];
>> x=sin(t)./(t+1);
>> y=cos(t)./(t+1);
>> comet(x,y)
```



Практическая работа № 5. Разработка интерфейса пользователя

Одной из неотъемлемых частей любой прикладной программы является интерактивный интерфейс пользователя, который разрабатывается для неоднократно решаемых задач с несколькими входными параметрами. Наличие интерфейса освобождает пользователя от необходимости вникать в детали программного продукта и дает возможность в реальном времени наблюдать за состоянием решения задачи. В настоящее время разработаны общие требования, которым должны удовлетворять программные интерфейсы, определены основные элементы управления графическим интерфейсом:

1. Одно или несколько окон для визуализации результатов расчетов.

2. Редактируемые окна, позволяющее задавать значения входных параметров задачи.

3. Управляющие кнопки для запуска и остановки процесса расчета, вывода результатов, завершения работы с программой.

4. Поясняющие надписи.

Для упрощения процедуры создания интерфейса пользователя в пакете Matlab, как и во всех современных объектно-ориентированных программных средах, существует возможность визуального программирования. Приложения MATLAB с графическим интерфейсом являются графическими окнами, содержащими элементы управления (кнопки, списки, переключатели, флаги, полосы скроллинга, области ввода, меню), а также оси и текстовые области для вывода результатов работы. Создание приложений включает следующие основные этапы - расположение нужных элементов интерфейса в пределах графического окна и программирование событий, которые возникают при обращении пользователя к данным объектам, например, при нажатии кнопки. Процесс работы над приложением допускает постепенное добавление элементов в графическое окно, запуск, тестирование приложения и возврат в режим редактирования. Конечным результатом является программа с графическим интерфейсом пользователя (GUI), содержащаяся в одном или нескольких файлах, запуск которой производится указанием ее имени в командной строке или в другом приложении MATLAB. Элементы управления являются графическими объектами -

потомками графического окна в иерархии объектов. Они могут быть созданы при помощи специальных функций низкоуровневой графики. Однако гораздо эффективнее воспользоваться визуальной средой GUIDE, которая позволяет создать заготовку окна приложения, разместить на ней элементы управления, запрограммировать события и их взаимосвязь.

Познакомимся с основными приемами работы с GUIDE на примере разработки интерфейса программы моделирования упругих колебаний без сопротивления при наличии периодической внешней силы, описываемой обыкновенным дифференциальным уравнением второго порядка:

$$x'' + qx = a \sin \omega t, \qquad (1)$$

где ω - частота вынуждающей силы;

a - амплитуда внешнего периодического воздействия;

q - собственная частота колебаний.

Для замыкания математической формулировки уравнение (1) следует также дополнить начальными условиями:

$$x(t = 0), x'(t = 0) = v(t = 0). \qquad (2)$$

Задачу Коши (1-2) будем решать, преобразуя уравнение (1) к системе дифференциальных уравнений первого порядка:

$$\begin{cases} y_2 = \frac{dy_1}{dt} \\ \frac{dy_2}{dt} = -qy_1 + a\sin(\omega t) \end{cases},$$

используя встроенный решатель дифференциальных уравнений ode45.

Перед созданием графического интерфейса желательно разработать его проект. Для рассматриваемой задачи интерфейс должен содержать: шесть окон для ввода параметров a, q, ω, начальных условий $x(t = 0)$, $v(t = 0)$ и правой границы интервала интегрирования дифференциального уравнения (1); соответственно шесть поясняющих надписей и две-три функциональные кнопки, позволяющие строить график колебаний (координаты или скорости), зависимость скорости от координаты и кнопка, позволяющая закончить работу приложения.

Для создания графического интерфейса (GUI - Graphic User Interface) с помощью средств визуального программирования необходимо запустить соответствующую программу командой **guide** или через меню: **File - New - GUI**. Появляется окно GUIDE Quick Start, изображенное на рис. 1, которое помогает настроить визуальную среду на создание приложения нужного типа или открыть существующее для продолжения работы. Шаблон для нового приложения выбирается в списке GUIDE templates на вкладке Create New GUI. Возможен один из следующих вариантов:

1) •Blank GUI (Default) — пустое окно приложения без элементов управления, осей и меню;

2) •GUI with Uicontrols — окно приложения со строками ввода, переключателями и кнопками;

3) •GUI with Axes and Menu— окно приложения, содержащее оси, раскрывающийся список, кнопки и меню;

4) •Modal Question Dialog — модальное диалоговое окно с кнопками Yes и No.

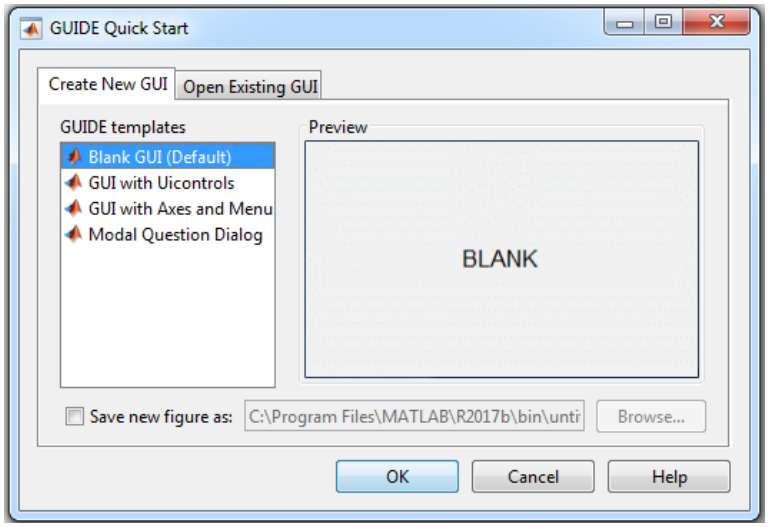


Рис. 1. Окно старта GUIDE.

Установка флага Save on startup as позволяет задать имя файла, в котором будет содержаться окно приложения.

Выберите опцию Blank GUI и нажмите кнопку OK. Появляется редактор среды GUIDE, заголовок которого untitled.fig означает, что в нем открыт новый файл, увеличьте немного размеры окна так, как изображено на рис. 2.

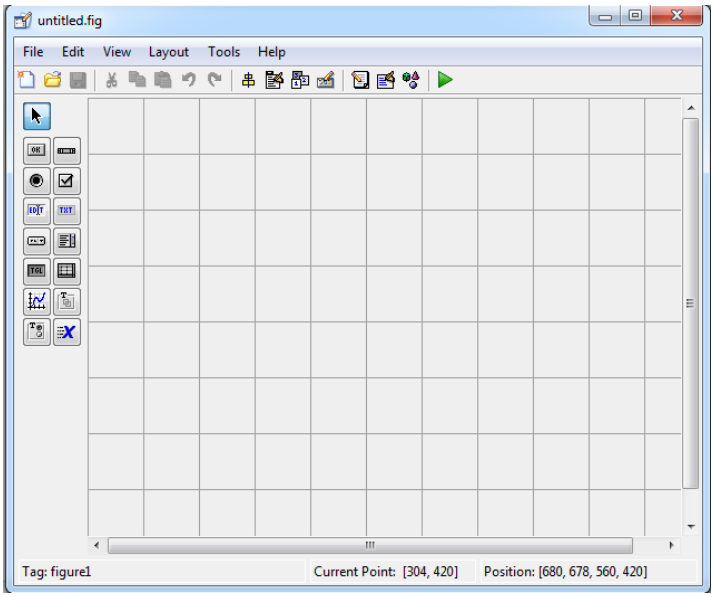


Рис. 2. Редактор среды GUIDE.

Редактор среды GUIDE содержит: строку меню; панель инструментов управления приложением; заготовку окна приложения с нанесенной сеткой; панель инструментов для добавления элементов интерфейса на окно приложения.

Редактор позволяет разместить различные элементы интерфейса (рис. 3). Для этого требуется нажать соответствующую кнопку на панели инструментов и поместить выбранный объект щелчком мыши на заготовку окна приложения.

Другой способ состоит в задании прямоугольной области объекта перемещением мыши по области заготовки окна с удержанием левой кнопки. Размер и положение добавленных объектов изменяются при помощи мыши. Перед изменением размера следует выбрать режим выделения объектов и сделать объект текущим, щелкнув по нему кнопкой мыши. Названия элементов управления появляются на всплывающих подсказках при наведении курсора мыши на кнопки панели инструментов.

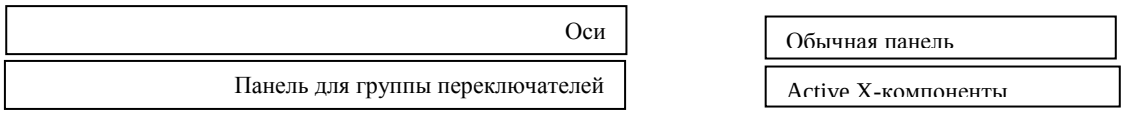


Рис. 3. Панель инструментов.

Приложение в данный момент находится в режиме редактирования. Любой объект можно удалить с окна, выделив его и нажав <Delete>. Запуск приложения производится при помощи кнопки Run. При нажатии на нее появляется диалоговое окно GUIDE, которое сообщает о том, что запуск приложения приведет к его сохранению. Пользователь может нажимать на кнопки, устанавливать флаги, переключатели, обращаться к спискам, разумеется, при этом ничего полезного не происходит. Однако недостаточно разместить элементы интерфейса в окне приложения, следует позаботиться о том, чтобы каждый элемент выполнял нужные функции при обращении к нему пользователя.

Практическая работа № 6. Программирование событий

Приведем объяснения принципа программирования событий в среде GUIDE. Приложение в MATLAB хранится (по умолчанию) в двух файлах с расширением fig и m, первый из них содержит информацию о размещенных в окне приложения объектах, а второй является М-файлом с основной функцией и подфункциями. Добавление элемента интерфейса из редактора приложения приводит к автоматическому созданию соответствующей подфункции. Данную подфункцию следует наполнить содержимым – операторами, которые выполняют обработку события, возникающего при обращении пользователя к элементу интерфейса.

Создание приложения продемонстрируем на примере разработки интерфейса программы, моделирующей колебания. Окно приложения содержит: оси, окна ввода, текстовые надписи, а также три кнопки, предназначенные для построения графика изменения координаты, скорости и завершения работы приложения. Для ввода названия формы откройте окно Inspector, сделав двойной клик в любом месте формы, затем установите новое значение поля Name, как показано на рис. 4.

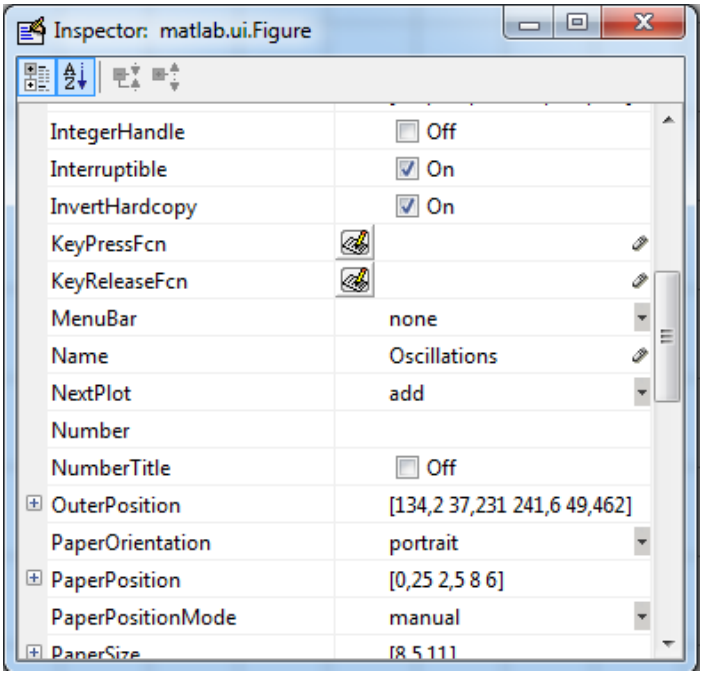


Рис. 4. Задание имени формы.

Сохраните проект на диске под таким же именем - Oscillations. Расположите на форме оси, окна ввода, текстовые подписи и кнопки. На кнопках автоматически размещаются надписи Push Button1, Push Button2, Push Button3, на осях - axes 1, на

окнах ввода – Edit1, Edit2, Edit3, Edit4, Edit5, Edit6, на текстовых подписях - Static Text. Текстовые надписи должны пояснять, какой параметр требуется ввести пользователю.

Кнопки, оси, окна ввода являются элементами интерфейса, которым следует дать имена, уникально идентифицирующие их. Щелчком мыши выделите кнопку Push Button1 и вызовите инспектор свойств Inspector при помощи панели инструментов управления приложением (либо, используя контекстное меню кнопки) (рис. 5).

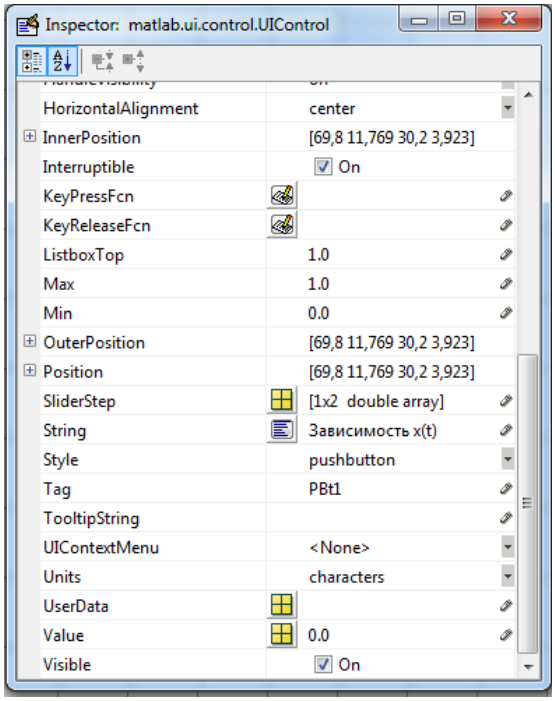


Рис. 5. Окно инспектора свойств объекта Push Button.

Появляется окно инспектора свойств, в котором содержится таблица названий свойств кнопки и их значений. Напомним, что элементы управления являются объектами U1 Objects, которые принадлежат графическому окну согласно иерархии графических объектов MATLAB. Важнейшим свойством в данный момент является Tag, которое содержит тэг или имя объекта. При добавлении элементов управления им автоматически присваиваются имена, причем по умолчанию они состоят их типа объекта и номера (в нашем случае pushbutton1). Дадим кнопке информативное имя, которые говорит, как о типе элемента, так и о его назначении. Эта кнопка будет служить для построения графика, что следует отразить в ее имени. Установите свойство Tag в значение PBt1, для чего щелкните мышью по строке справа от назва-

ния свойства, наберите требуемое значение и нажмите <Enter>. У остальных объектов оставьте имена, предложенные по умолчанию. Следует различать имя объекта и надпись на нем. Кнопка с именем PBt1 содержит текст Push Button. Измените его, установив в инспекторе свойств Inspector свойство string в значение «Зависимость x(t)». Теперь надпись на кнопке соответствует ее назначению. Для окон ввода сохранять подпись необязательно, поэтому строку String оставьте пустой, удалив подпись в окне свойств объектов Edit, как показано на рис. 6.

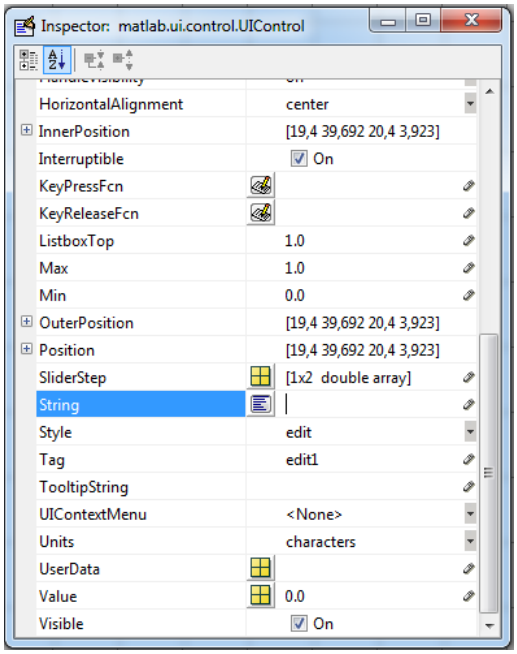


Рис. 6. Окно инспектора свойств объекта Edit.

При создании приложений с графическим интерфейсом вам придется часто прибегать к инспектору свойств Inspector для установки свойств объектов в подходящие значения. Установите имена, свойства String на остальных объектах самостоятельно, так, чтобы получился проект интерфейса вида, представленного на рис. 7. При нажатии кнопок (например, кнопки BPt1) в работающем приложении генерируется событие Callback данного элемента управления. При этом вызывается подфункция BPt1Callback. Имя подфункции образовано именем (тегом) кнопки и названием события. Данная подфункция не содержит операторов, и при нажатии на кнопку в запущенном приложении ничего не происходит.

Основной этап состоит в задании действий, которые выполняются при нажатии пользователем на кнопки. Запрограммируйте подфункцию BPt1 Callback обра-

ботки события Callback (рис. 8) кнопки «Зависимость x(t)» с именем BPt1 в соответствии с листингом (автоматически созданные комментарии убирать не обязательно).

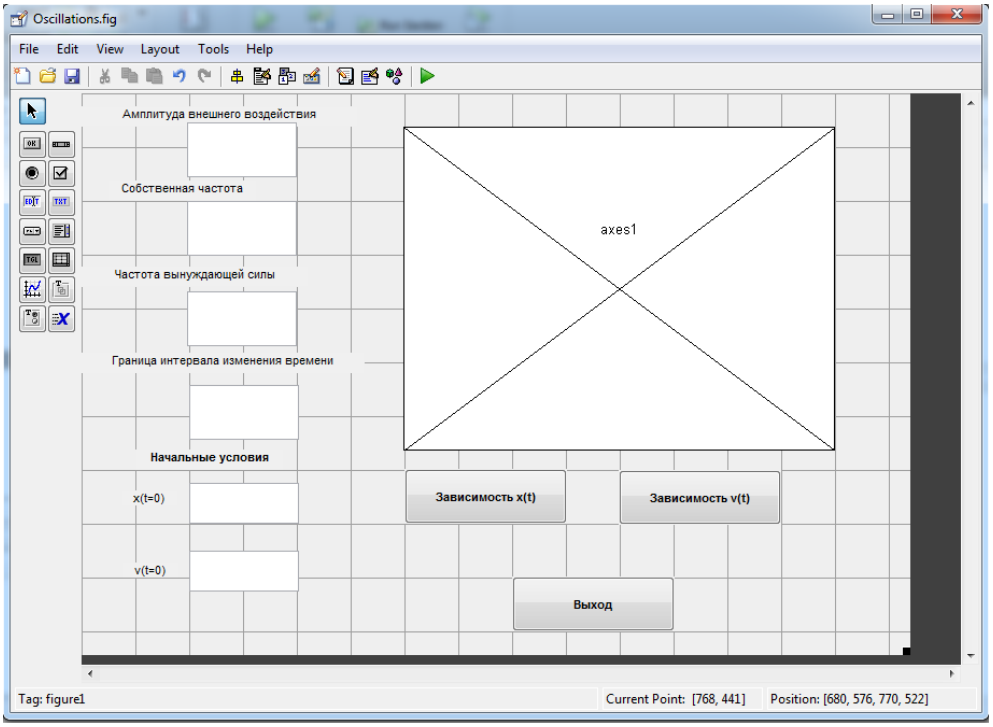


Рис. 7. Расположение элементов интерфейса.

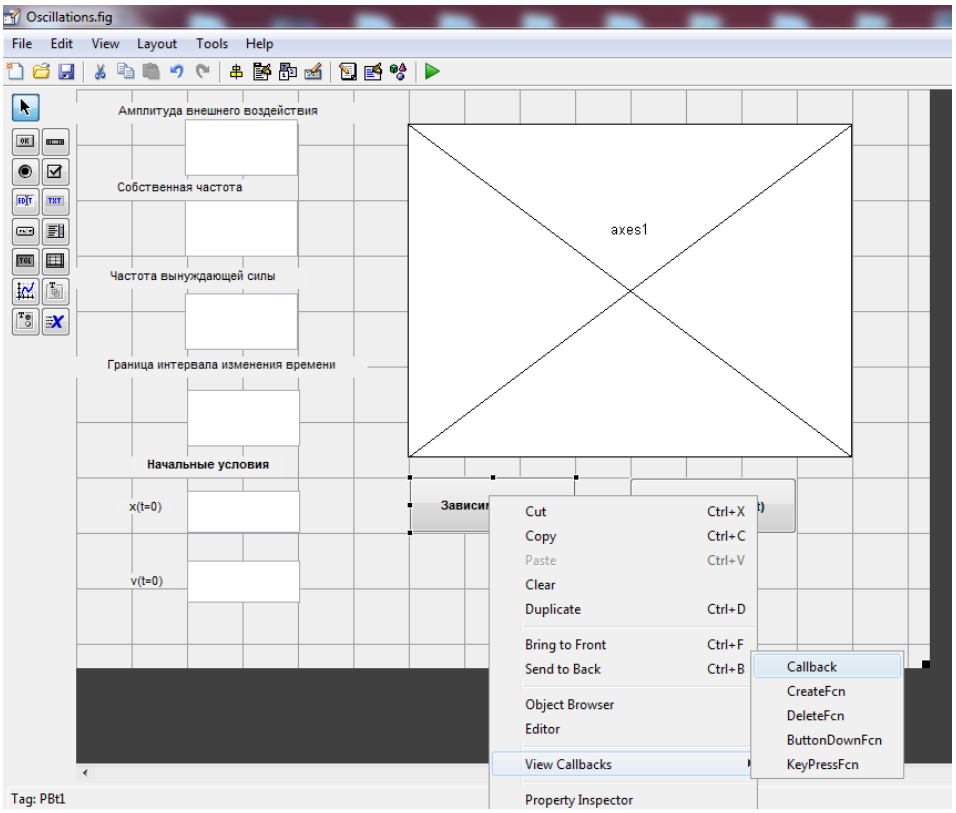
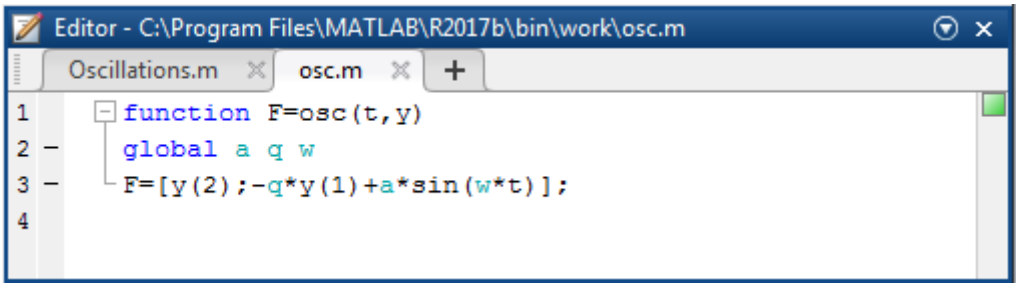


Рис. 8. Обращение к обработчику события - нажатие кнопки.

```
function PBtl_Callback(hObject, eventdata, handles)
% hObject handle to PBtl (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUI-
DATA)

%объявление переменных для вызова вспомогательной функции,
%содержащей вектор правых частей дифференциального уравнения
global a q w b;

%формирование вектора начальных значений, задание констант
%считывание строковых переменных введенных начальных условий

Q1=get(handles.edit5,'String');
Q2=get(handles.edit6,'String');
Q3=get(handles.edit1, 'String');
Q4=get(handles.edit2, 'String');
Q5=get(handles.edit3, 'String');
Q6=get(handles.edit4, 'String');

%и перевод в формат действительного числа

c1=str2double(Q1);
c2=str2double(Q2);
a=str2double(Q3);
q=str2double(Q4);
w=str2double(Q5);
b=str2double(Q6);
Y0=[c1;c2];

%вызов солвера odell3 от функции osc, задание начального и
%конечного момента времени ([0 b] ), вектора начальных
%условий Y0
[T,Y]=odell3('osc',[0 b], Y0);
%вывод графика решения исходного дифуравнения
plot(T,Y(:,2),'r.-')
grid on
```

Для решения дифференциального уравнения требуется задание вектор- функции производных, которую необходимо прописать в отдельном m-файле и сохранить под именем osc:



Кнопка «Зависимость v(t)» с именем BPt2 позволяет строить график зависимости скорости от времени. Запрограммируйте подфункцию BPt2_Callback обработки события Callback

```
global a q w b;
Q1=get(handles.edit5, 'String');
Q2=get(handles.edit6, 'String');
Q3=get(handles.edit1, 'String');
Q4=get(handles.edit2, 'String');
Q5=get(handles.edit3, 'String');
Q6=get(handles.edit4, 'String');
c1=str2double(Q1);
```

```
c2=str2double(O2);
a=str2double(O3);
g=str2double(O4);
w=str2double(O5);
b=str2double(Q6);
Y0=[c1;c2];
[T,Y]=odell3('osc',[0 b], Y0);
plot (T, Y(:,1),'r.-')
grid on
```

Для завершения работы с приложением выполните команду Close в обработчике кнопки «Выход» с именем PBt3.

```
function PBt3_Callback(hObject, eventdata, handles)
% hObject handle to PBt3 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUI-
DATA)
close;
```

Перейдем теперь к запуску приложения **Oscillations**. Нажмите кнопку **Run** на панели инструментов среды GUIDE. Если приложение сохранено в отдельном каталоге, то появляющееся диалоговое окно предлагает сделать этот каталог текущим. Убедитесь, что выбран переключатель Change MATLAB current directory, и нажмите OK. Теперь ваш каталог стал текущим, и при последующих запусках приложения это окно выводиться не будет, при условии, если вы не измените текущий каталог из рабочей среды. Перед запуском может появиться еще одно диалоговое окно, в котором следует подтвердить сохранение файлов с расширениями fig и m, содержащих приложение. После выполнения описанных действий на экране отображается окно приложения **Oscillations**. Задайте параметры модели, начальные условия и правую границу интервала изменения времени в соответствующих окнах ввода текста: в нормированных единицах. Нажатие на кнопку «Зависимость x(t)» приводит к отображению графика функции координаты от времени на осях (рис. 9), нажатие кнопки «Зависимость v(t)» даст временную зависимость скорости.

Запустите приложение и убедитесь, что названия кнопок соответствуют выполняемым действиям.

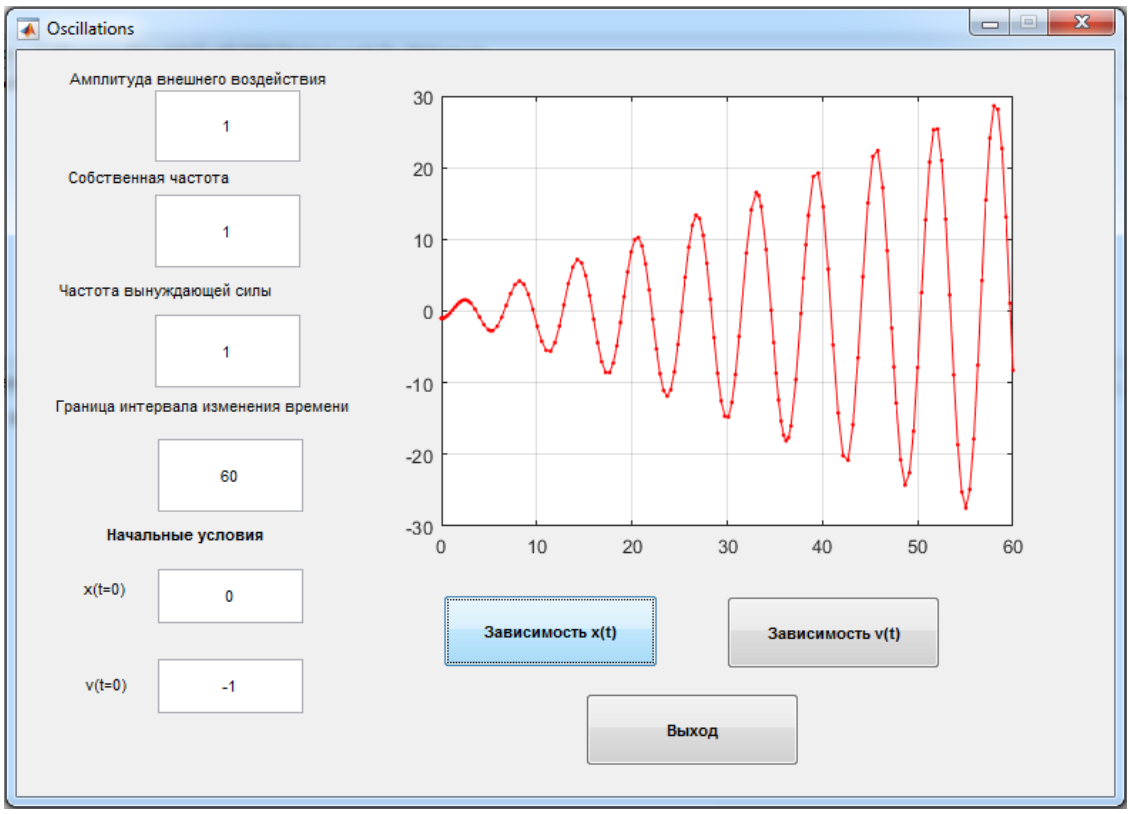


Рис. 9. Вид формы окна после запуска приложения.

Проведем анализ результатов моделирования. Выбрав параметры a=1,ω=1,q=1,y(0)=0,y'(0)=-1,b=60, получим численное решение дифференциального уравнения, совпадающего с аналитическим решением $y(t)=5(\sin t - t\cos t)$. Изменив значение параметра q на 1.5, получим биения (при несовпадении частоты периодической внешней силы с собственной частотой колебаний), график которых показан на следующем рисунке (рис. 10). Физическая причина различия данных графиков заключается в следующем. Если частота внешней силы совпадает с частотой собственных колебаний, как в первом случае, при отсутствии в системе сил трения (сопротивления движению) происходит явление резонанса - неограниченное возрастание амплитуды колебаний, что мы и видим на рис. 9. Естественно, что в реальной системе не может быть полного отсутствия трения, поэтому амплитуда, достигнув максимального значения, далее увеличиваться не будет. Во втором случае происходит сложение двух колебаний с разными частотами, и наблюдаются характерные биения.

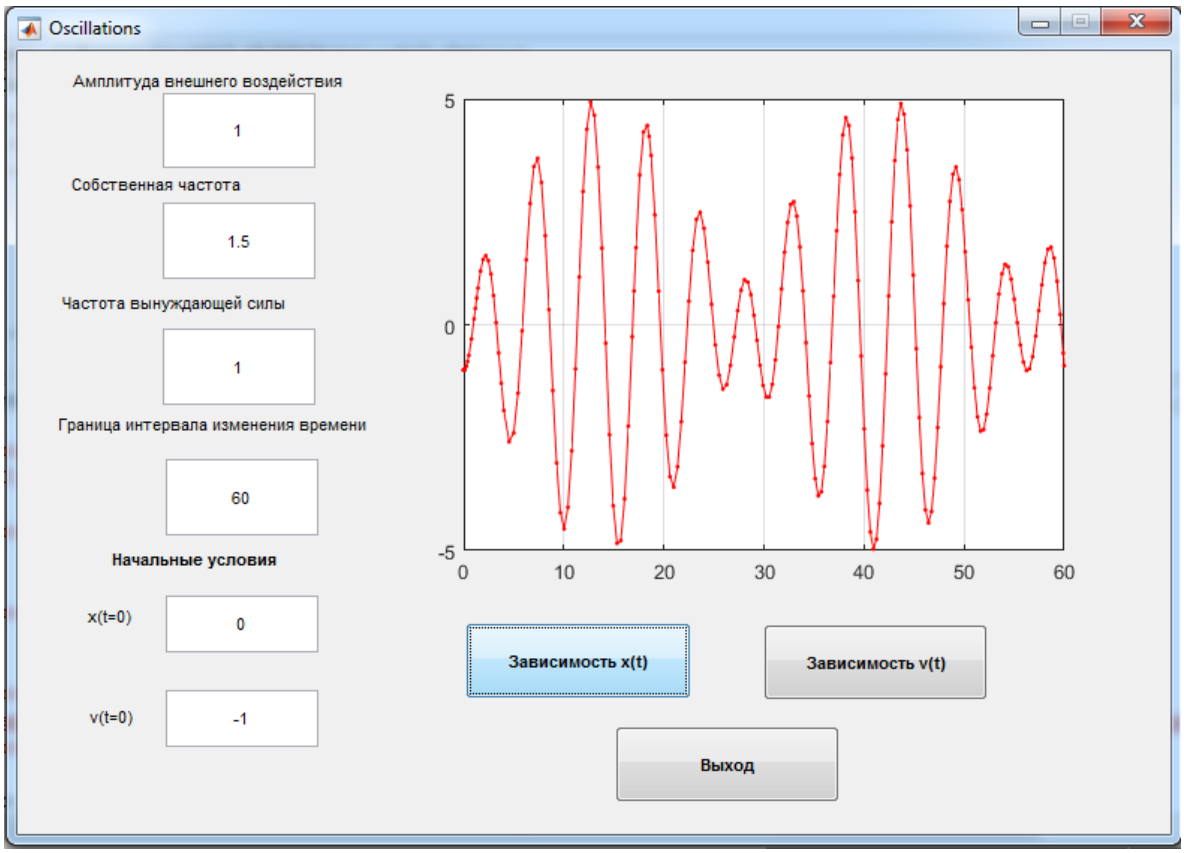


Рис. 10. Вид формы окна после запуска приложения с низменным значением параметра q.

Лабораторная работа № 1. Моделирование физических процессов, описываемых дифференциальными уравнениями 1-го порядка

Краткая теории. Моделирование остывания нагретых тел

Природа переноса тепла от нагретого тела в окружающий его воздух весьма сложна и включает в себя механизмы конвекции, излучения, испарения и теплопроводности. Однако при небольшой разности температур между нагретым телом и окружающей средой для описания процесса остывания можно испытывать простую модель, впервые предложенную Ньютоном. В данной модели температура холодильника (окружающей среды) принимается постоянной, а скорость передачи тепла пропорциональна разности температур между нагретым телом и окружающей средой.

Теплоотдача в окружающую среду осуществляется по закону Ньютона:

$$dQ = \alpha S(T - T_c)dt, \qquad (1)$$

где dQ - количество теплоты, отданное телом за время dt,

T – текущая температура тела;

T_c – температура окружающей среды;

S – площадь поверхности тела;

α – коэффициент теплоотдачи (остывания).

Если в результате теплоотдачи температура тела снижается, формула (1) будет верна только для малого промежутка времени Δt.

Изменение внутренней энергии тела на величину dU, в свою очередь, приводит к изменению температуры тела на величину dT:

$$dT = \frac{1}{C_T}dU,$$

где C_T – теплоемкость тела ($C_T = \frac{dV}{dT}$).

При установлении связи между dU и dQ следует учесть, что в теле могут находиться и др. источники тепла. Рассмотрим упрощенный случай, когда в теле отсутствуют внутренние источники тепла.

В данной модели подразумевается, что все тело имеет одинаковую температуру, что реализуется при достаточно большой теплопроводности, небольших размерах тела и не слишком большой теплоотдаче. Будем считать, изменение внутренней энергии тела полностью происходит за счет отдаваемого тепла:

$dQ=dU,\ dU=C_T dT=cmdT=c\rho VdT.$

где с - удельная теплоемкость тела.

Тогда:

$-\alpha S(T - T_c)dt = c\rho VdT$, откуда имеем:

$$\frac{dT}{dt} = -\frac{\alpha S}{c\rho V}(T - T_c). \qquad (2)$$

Постоянную величину, являющуюся множителем правой части (2), часто называют «коэффициентом остывания» и обозначают весь указанный комплекс одной буквой: $r = \frac{\alpha S}{c\rho V}$ (зависит от механизма передачи тепла, площади нагретого тела и тепловых свойств самого тела).

Математическая постановка задачи и алгоритм решения

В итоге получили математическую запись закона Ньютона в виде дифференциального уравнения первого порядка (2), левая часть которого соответствует скорости изменения температуры.

Для замыкания математической формулировки требуется задание начальных условий:

$$T(t_0) = T_0, \qquad (3)$$

где t_0 - начальный момент времени;

T_0 - начальная температура тела.

Задача (2-3) в математической постановке представляет собой задачу Коши дня обыкновенного дифференциального уравнения.

Таким образом, требуется решить обыкновенное дифференциальное уравнение первого порядка, заданное в общем виде

$$\frac{dy}{dx} = f(x, y(x)). \qquad (4)$$

Уравнение вида (4) может не иметь аналитического решения, выражаемого через элементарные функции. В таких случаях целесообразно применять численные алгоритмы. Для численного решения уравнения (4) с начальными условиями:

$$y(x_0) = y_0 \qquad (5)$$

вместо исходного ОДУ находим решение конечно-разностного уравнения:

$$\frac{dy}{dx} \to \frac{y(x+\Delta x)-y(x)}{\Delta x},$$

$$y(x + \Delta x) \approx y(x) + \Delta x f(x, y(x)).$$

Таким образом, для численного решения необходимо:

1. Задать начальные условия y_0, x_0;
2. Найти первое приближение $y_1 \cong y_0 + \Delta x f(x_0 y_0)$;
3. Повторять процесс с учетом итерационной формулы:

$y_i = y_{i-1} + \Delta x f(x_{i-1} y_{i-1})$, при $i=\overline{2,n}$.

Данный метод носит название «Метод касательных», или «Метод Эйлера». Метод дает хорошее приближение к «истинному значению» функции $y(x)$, если Δx значительно мало.

Аналитическое решение задачи остывания нагретого тела

При теплофизических параметрах охлаждающегося тела, независящих от температуры, задача (2-3) имеет точное решение.

Интегрируя (2) как обыкновенное дифференциальное уравнение 1-го порядка с разделяющими переменными, получим:

$$\int \frac{dT}{(T-T_c)} = -\int \frac{\alpha S}{c\rho V} dt$$

$$In|T - T_c| = -\frac{\alpha S}{c\rho V} t + C.$$

Или (проверьте самостоятельно полученный результат):

$$T = T_c + (T_0 - T_c)e^{-\frac{t}{\tau}},$$

где $\tau = \frac{c\rho V}{\alpha S}$ – параметр, имеющий смысл времени релаксации.

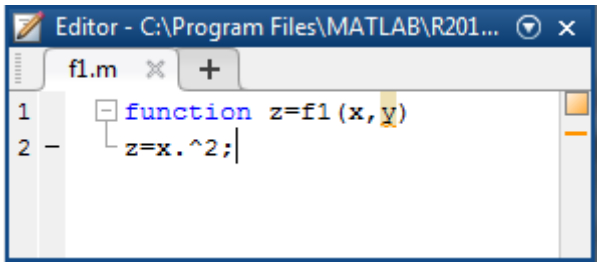
Численный пример. Реализация в пакете Mallab

Решение задачи Коши для дифференциальных уравнений методом Рунге-Кутты 4 порядка реализовано в пакете Matlab в виде функции **ode45()**. Описание других функций, реализующих метод Рунге-Кутты и другие методы, а также решатели систем дифференциальных уравнений можно найти, используя справочную систему Matlab.

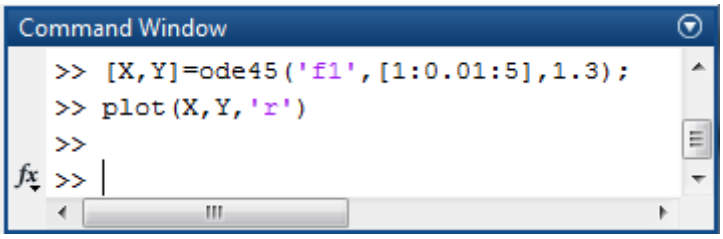
Пример 1. Найти решение задачи Коши для дифференциального уравнения, используя встроенные функции пакета Matlab.

$$\frac{dy}{dx} = x^2,\ y(0) = 1.3$$

1. Создадим функцию fl.m, содержащую значение *f*(*x*,*y*)

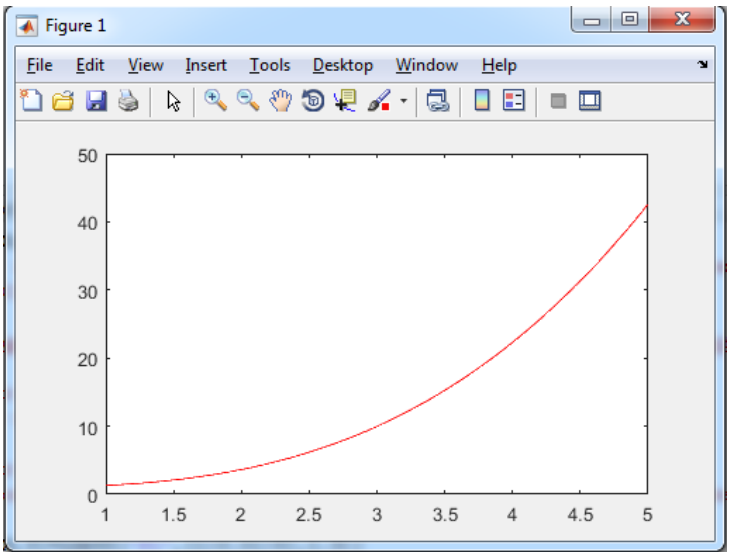


Построим решение в Matlab, используя функцию ode45.



Входными данными для функции ode45 являются: имя файла, содержащего определение функции, стоящей в правой части уравнения (4), вектор, определяющий интервал интегрирования, начальное условие (5).

Результат построения графика функции $y = f(x)$:



Задания к лабораторной работе

Пользуясь изложенной теорией, провести моделирование остывания нагретого тела, выбирая в качестве объекта моделирования «чашку кофе», для которого принимаются следующие параметры моделирования:

промежуток времени остывания: [0,20] (мин.)

время релаксации: 40 (мин.)

начальная температура воды: 90°С,

температура окружающей среды: 22°С.

Задание № 1. Провести моделирование, используя для решения задачи Коши пошаговый метод Эйлера. Найти аналитическое решение задачи.

Задание № 2. Найти решение задачи, используя встроенные функции ППП Matlab. Провести сравнение трех результатов, используя геометрические интерпретации результатов.

Задание № 3. Провести сравнение точного и приближенного решений, варьируя шаг дискретизации времени. Зависит ли разность между численным и точным решением от времени? Построить график разности (ошибки) между точным и приближенным как функцию Δt. Будет ли эта функция убывающей?

Задание № 4. Какой шаг Δt нужно выбрать, чтобы точность 0,1% достигалась в момент времени t=1 мин.? В момент времени 5 мин.?

Задание № 5. Оформить отчет по лабораторной работе.

Лабораторная работа №2. Динамика материальной точки

Введение. Основные понятия

Большинство задач, связанных с определением закона движения тела вблизи земной поверхности, имеют известные аналитические решения. Однако полезными являются решения, полученные с помощью реализации численных алгоритмов, так как на их примере можно продемонстрировать наиболее общие приемы и подходы к решению задач.

Типичным примером движения является падение предметов у земной поверхности. Простейшее описание такого движения не учитывает возможные вращения и внутренние движения тела, т.е. реальное физическое тело заменяется идеализированным объектом, не обладающим внутренней структурой - материальной точкой.

Задача описания закона движения материальной точки традиционно рассматривается в кинематике, где движение материальной точки рассматривается безотносительно причины ее вызывающей, и в динамике, где рассматриваются силы, действующие на тело. В соответствии со II законом Ньютона ускорение $\vec{a}$, с которым движется тело, прямо пропорционально действующей на него силе $\vec{F}$ (или равнодействующей всех сил) и обратно пропорционально его массе *m*:

$$\vec{a} = \frac{\vec{F}}{m}. \qquad (1)$$

В общем случае сила F зависит от положения точки, характеризуемого радиус-вектором $\vec{r}(t)$, скорости $\vec{v}(t)$ и времени: $\vec{F} = \vec{F}(\vec{r},\vec{v},t)$.

Так как по определению:

$$\vec{a} = \frac{d\vec{v}}{dt} = \frac{d^2\vec{r}}{dt^2},$$

то зависимость скорости от времени описывается уравнением

$$\frac{d\vec{v}}{dt} = \frac{\vec{F}}{m}, \qquad (2)$$

а зависимость координаты от времени

$$\frac{d^2\vec{r}}{dt^2} = \frac{\vec{F}}{m}, \qquad (3)$$

то для описания движения материальной точки необходимо решить обыкновенное дифференциальное уравнение второго порядка относительно координаты (3), которое для тела с постоянной массой эквивалентно системе обыкновенных дифференциальных уравнений первого порядка:

$$\begin{cases} \vec{v}(t) = \frac{d\vec{r}}{dt}, \\ \vec{a}(t) = \frac{d\vec{v}}{dt}. \end{cases} \quad (4)$$

Движение тел в гравитационном поле Земли без учета трения

Рассмотрим задачу об одномерном движении материальной точки с постоянной массой *m* в гравитационном поле Земли. По закону всемирного тяготения сила, действующая на тело массой *m*:

$$F = \frac{\gamma M m}{(R+y)^2} = \frac{gm}{\left(1+\frac{y}{R}\right)^2}, \quad (5)$$

где *y* – расстояние от поверхности Земли;

R – радиус Земли;

γ – гравитационная постоянная;

M – масса Земли;

g= $\frac{\gamma M}{R^2}$.

В отсутствие сопротивления воздуха все тела независимо от их массы, размеров и состава на одинаковом расстоянии от земной поверхности имеют одинаковое ускорение. Такое идеализированное движение, при котором сопротивлением воздуха пренебрегают, называют «свободным падением». Ускорение свободно падающего тела вблизи поверхности Земли приблизительно равно g=9.8 м/с^2.

На первом шаге будет считать силу тяжести постоянной, что справедливо, как видно из (5), для *y/R* <<1.

Выберем систему координат, ось *OY* которой направлена вверх, как показано на рис. 1. Тогда подставляя (5) в (3), уравнение движения в системе отсчета:

$$\frac{d^2y}{dt^2} = -g. \quad (6)$$

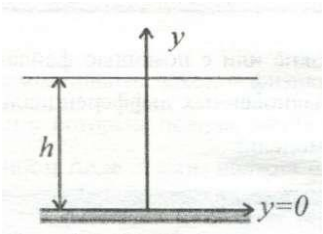


Рис. 1. Система координат, используемая для описания движения тела в гравитационном поле Земли.

Тогда задача Коши дополнится начальными условиями:

$$y(0) = y_0, v(0) = y'(0) = \pm v_0.$$

Аналитическое решение получается последовательным интегрированием (6):

$$v(t) = \pm v_0 - gt, y(t) = y_0 \pm v_0 t - \frac{1}{2} g t^2. \quad (7)$$

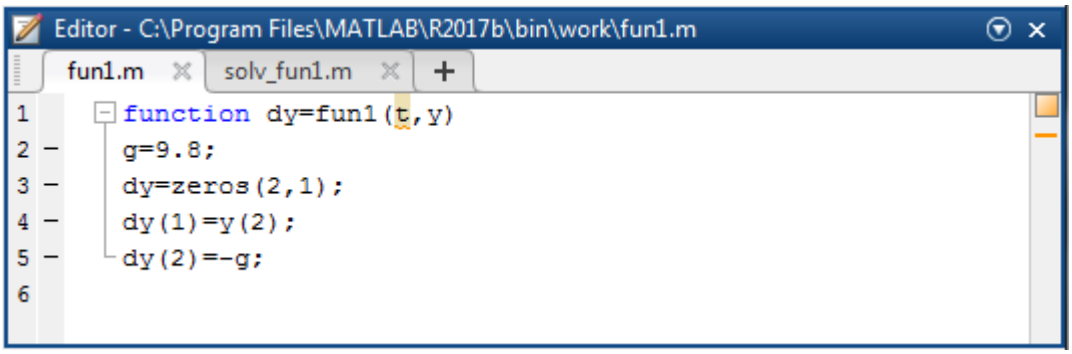
Чтобы реализовать численное решение, представим (6) в виде:

$$\begin{cases} \frac{dy}{dt} = z \\ \frac{dz}{dt} = -g \end{cases}$$ или в векторной форме: $\frac{d}{dt}\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} z \\ -g \end{pmatrix}$

с начальными условиями:

$$\begin{pmatrix} y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ \pm v_0 \end{pmatrix}.$$

Для реализации этой модели в Matlab необходимо создать m-файл, содержащий описание вектор-функции:



Далее в командном окне или с помощью файла-программы необходимо обратиться к решателю обыкновенных дифференциальных уравнений, предварительно задав константы модели:

```
g=9.8;
y0=10;
v0=0.3;
[t,y]=ode45('fun1',[0:0.01:1.5],[y0,v0]);
figure(1);plot(t,y(:,1));
figure(2);plot(t,y(:,2));
```

Результат моделирования зависимости координаты от времени представлен рис. 2.

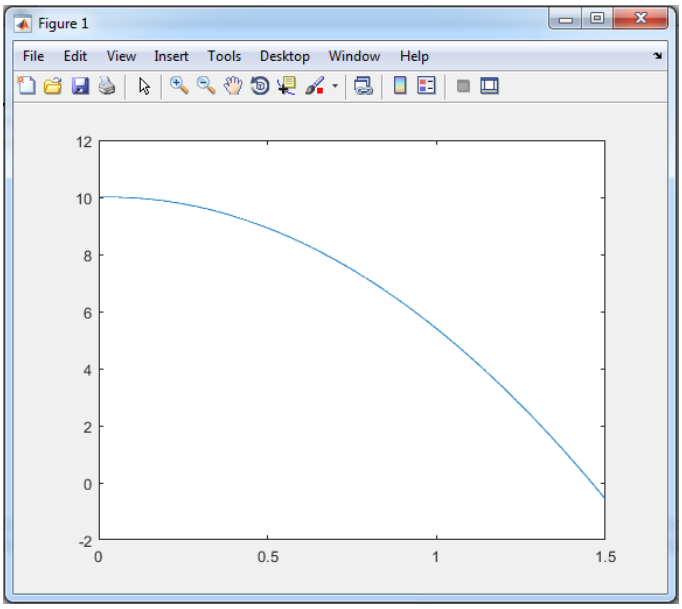


Рис. 2. Зависимость координаты материальной точки от времени – $y(t)$.

Заданием № 1 к лабораторной работе

1. Провести численное моделирование одномерного движения тела в гравитационном поле Земли, используя описанную программную реализацию. Сравнить результаты с аналитическим решением (7).

2. Вывести в графическом представлении зависимости координаты от времени, скорости от времени, скорости от координаты.

3. Построить зависимость кинетической и потенциальной энергий от времени.

4. Одним из фундаментальных законов физики является закон сохранения энергии, в соответствии с которым полная энергия материальной точки, движущейся в гравитационном поле Земли, должна оставаться постоянной. Проверьте полученное численное решение ОДУ на соответствие этому закону. С какой точностью

выполняется закон сохранения энергии и чем можно объяснить полученные расхождения?

Рассмотрим двумерное движение материальной точки в гравитационном поле Земли, описываемое уравнением (3). В системе отсчета, показанной на рис. 3, уравнение (3) эквивалентно следующей системе ОДУ второго порядка:

$$\begin{cases} \dfrac{d^2x}{dt^2} = 0 \\ \dfrac{d^2y}{dt^2} = -g. \end{cases} \tag{8}$$

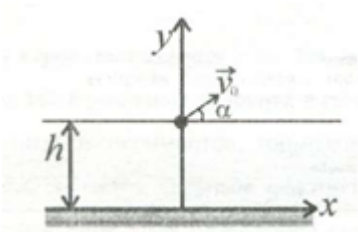
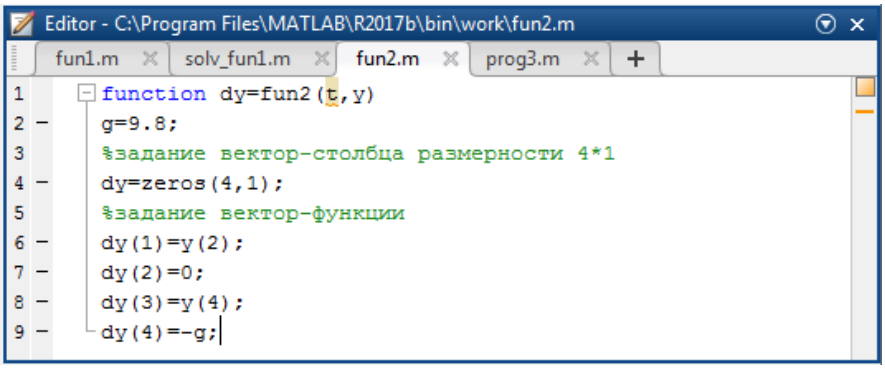


Рис. 3. Система координат, используемая для описания двумерного движения тела в гравитационном поле Земли.

Система уравнений (8) заменой переменных $\frac{dx}{dt} = z_1, \frac{dy}{dt} = z_2$ сводится к системе линейных уравнений первого порядка:

$$\begin{cases} \dfrac{dx}{dt} = z_1, \dfrac{dz_1}{dt} = 0, \\ \dfrac{dy}{dt} = z_2, \dfrac{dz_2}{dt} = -g. \end{cases}$$

Для решения задачи Коши для данной системы следует задать четыре начальных условия $x(0)$, $v_x(0)$, $y(0)$, $v_y(0)$, а также составить m-файл, содержащий определение вектор-функции, стоящей в правой части системы ОДУ.




```
function dy=fun2(t,y)
g=9.8;
%задание вектор-столбца размерности 4*1
dy=zeros(4,1);
%задание вектор-функции
dy(1)=y(2);
dy(2)=0;
dy(3)=y(4);
dy(4)=-g;
```

```
%задание начальных условий
x0=0;
vx0=2;
y0=0;
vy0=5;
%решение ОДУ - вызов солвера
[t,X]=ode45('fun2',[0:0.001:1.019],[x0 vx0 y0 vy0]);
plot(X(:,1),X(:,3))
grid on
```

Результат моделирования зависимости координат представлен на рис.4.

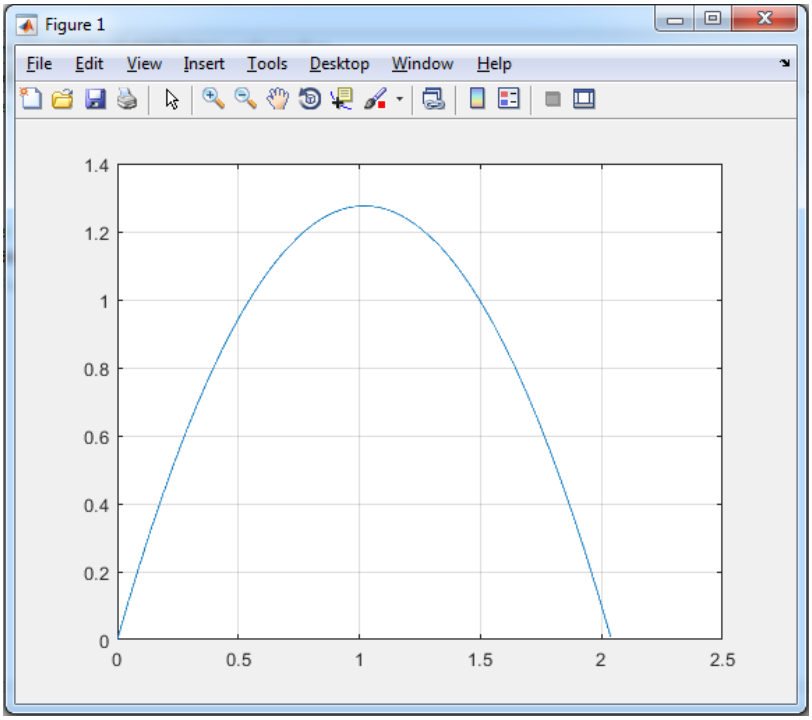


Рис. 4. Зависимость координат материальной точки $y(x)$.

Задание № 2 к лабораторной работе

1. Проведите программную реализацию, постройте зависимости координат и составляющих скорости от времени, координаты y от координаты x.

2. Сравните зависимости, задаваемые аналитическими выражениями, с аналогичными зависимостями, полученными численным интегрированием.

3. Построить зависимость кинетической и потенциальной энергий от времени.

Движение тел в гравитационном поле Земли с учетом силы трения

При движении тел в реальных условиях в газовых или жидких средах, как показывают результаты экспериментов, тормозящая сила, обусловленная сопротивле-

нием среды, является главным фактором, определяющим характер движения тела. Направление этой тормозящей силы должно быть противоположно скорости движения тела. Рассмотрим сначала падение материальной точки. Если пользоваться системой координат, показанной на рис.5., то полную силу, действующую на материальную точку, можно записать в виде:

$$F = F_g - F_d = mg - F_d. \qquad (9)$$

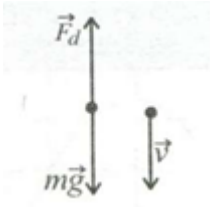


Рис. 5. Силы, действующие на падающее тело, с учетом сопротивления воздуха.

О силе сопротивления известно, что она (за исключением некоторых ситуаций) растет с ростом скорости. В общем случае зависимость F_d от скорости необходимо определять эмпирически, проводя конечную серию наблюдения положения тела. Для функции F_d предполагается какой-то определенный вид зависимости от скорости v, и эта формула используется для нахождения функции $y(t)$. Если вычисленные значения функции $y(t)$ согласуются с экспериментами, то предложенная зависимость считается экспериментально подтвержденной. Наиболее общими зависимостями силы сопротивления от скорости являются:

$$F_C(v) = k_1 v, \qquad (10)$$

$$F_C(v) = k_2 v^2, \qquad (11)$$

где k_1, k_2 - зависят от свойств среды и геометрии тела.

Соотношения (10-11) являются феноменологическими выражениями, приближенно описывающими функцию $F_d(v)$ в некоторых диапазонах скоростей.

Обратимся к случаю двумерных траекторий. Рассмотрим движение материальной точки массой т с начальной скоростью v_0, направленной под углом α к горизонту (рис. 6). На материальную точку действуют сила тяжести и сила сопротивления, направленная противоположно скорости. При относительно малых скоростях движения тела можно считать, что справедливо (11).

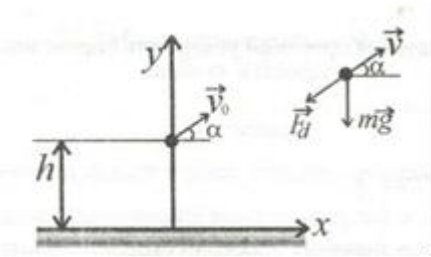


Рис. 6. Силы, действующие на материальную точку.

Тогда запишем уравнения движения Ньютона для компонент *x* и *y*.

$$a_x = \frac{F_{dx}}{m} = \frac{-F\cos\alpha}{m} = \frac{-k_1 v^2 \cos\alpha}{m},$$

$$a_y = \frac{F_{dy}}{m} - g = \frac{\mp F\sin\alpha}{m} = \frac{\mp k_1 v^2 \sin\alpha}{m} - g.$$

Учитывая, что $V_x = V \cdot cos\alpha$, $V_y = \pm V \cdot sin\alpha$, ($V = \sqrt{V_x^2 + V_y^2}$), получим:

$$a_x = -\frac{k_1 \cdot v \cdot v_x}{m},\ a_y = -\frac{k_2 \cdot v \cdot v_y}{m} - g. \qquad (12)$$

Система обыкновенных дифференциальных уравнений второго порядка определит закон движения:

$$\begin{cases} \frac{d^2x}{dt^2} = -\frac{k_1 v v_x}{m} \\ \frac{d^2y}{dt^2} = -\frac{k_2 v v_y}{m} - g \end{cases} \text{или} \begin{cases} \frac{d^2x}{dt^2} = -\frac{k_1 v}{m} \cdot \frac{dx}{dt} \\ \frac{d^2y}{dt^2} = -\frac{k_2 v}{m} \cdot \frac{dy}{dt} - g. \end{cases} \qquad (13)$$

Поскольку аналитически решать уравнения движения (13) с равнодействующей силой, учитывающей сопротивление воздуха, непросто, возможно прибегнуть к использованию численных методов, например, метода Эйлера, обобщенного на случай системы обыкновенных дифференциальных уравнений второго порядка. Для этого необходимо ввести дискретизацию по времени:

$t_i = t_0 + i\Delta t$, $i = \overline{1,n}$,

а также ввести в рассмотрение a_i, v_i, x_i, y_i – мгновенные значения ускорения, скорости и координат на *i*-ом шаге.

Мгновенные значения проекций ускорения определим из (12):

$$\begin{cases} a_{xi} = -\frac{k_1 v_{i-1} v_{xi-1}}{m} \\ a_{yi} = -\frac{k_2 v_{i-1} v_{yi-1}}{m} - g. \end{cases} \qquad (14)$$

Тогда мгновенные значения проекций скорости примут вид:

$$\begin{cases} v_{xi} = v_{xi-1} + a_{xi-1}\Delta t \\ v_{yi} = v_{yi-1} + a_{yi-1}\Delta t. \end{cases} \quad (15)$$

Значения координат в i-ой точке интервала вычисляем через значения производной скорости в начальной точке этого интервала:

$$\begin{cases} x_i = x_{i-1} + v_{xi-1}\Delta t \\ y_i = y_{i-1} + v_{yi-1}\Delta t. \end{cases} \quad (16)$$

Задание № 3 к лабораторной работе

1. Для конкретных параметров и начальных условий провести моделирование траектории движения тела, брошенного под углом к горизонту, используя пошаговое вычисление аналитических зависимостей (14-16). Построить графики зависимостей $x(t)$, $y(t)$, $y(x)$, $v_x(t)$, $v_y(t)$.

2. Провести моделирование, используя для решения системы ОДУ (9) встроенные функции ППП Matlab. Построить графики зависимостей $x(t)$, $y(t)$, $y(x)$, $v_x(t)$, $v_y(t)$. Провести численное сравнение с результатами, полученными в п.1.

3. Построить зависимость перемещения материальной точки и угла между вектором скорости и осью OX от времени.

4. Сравните реализацию той же модели, но без учета сил трения. Можно ли считать реалистичной модель движения тела в гравитационном поле Земли, не учитывающей наличие силы трения?

5. Оформить отчет по лабораторной работе.

Пример набора параметров:

$x_0 = 0$, $y_0 = 10$ *м*, $\alpha = 45°$, $k_1 = 0.16$, $v_0 = 30$ *м/с*, $m = 50$ *кг*, $t \in [0,10]$*с*, $\Delta t = 0.001$*с*.

Лабораторная работа № 3. Задача Кеплера

Введение

Задача о движении планет в поле тяжести небесных светил (является частным случаем задачи о движении в поле центральных сил) известна на протяжении нескольких тысячелетий. Большую часть знаний о движении планет объединили в себе законы Кеплера, полученные на основе данных астрономических наблюдений, которые формулируются следующим образом:

1. Всякая планета движется по эллиптической орбите, в одном из фокусов которого находится Солнце.

2. Скорость планеты возрастает по мере удаления от Солнца таким образом, что прямая, соединяющая Солнце и планету, в равные промежутки времени замещает одинаковую площадь.

3. Для всех планет, вращающихся вокруг Солнца, отношение T^2 / R^3 одинаково (T – период обращения, R – большая полуось эллипса).

Кеплер ввел свои законы на основании тщательного анализа данных наблюдений, систематизированных Тихо Браге. Аналитическое решение задачи Кеплера удается получить только в случаях рассмотрения движения двух тел, взаимодействующих по закону обратных квадратов. Задача Кеплера для трех и более тел аналитического решения не имеет и может быть решена только численно.

Краткая теория. Уравнения движения планет

Движение Солнца и Земли является примером задачи двух тел. Эту задачу можно свести к задаче одного тела двумя методами. В основе первого лежит тот факт, что масса Солнца во много раз больше массы Земли. Следовательно, с хорошей точностью можно считать Солнце неподвижным и связать с ним начало системы координат. Движение двух тел с массами m и M, полная потенциальная энергия которых зависит только от расстояния между ними, можно свести к эквивалентной задаче о движении одного тела приведенной массы μ, определяемой формулой:

$$\mu = \frac{Mm}{M+m}.$$

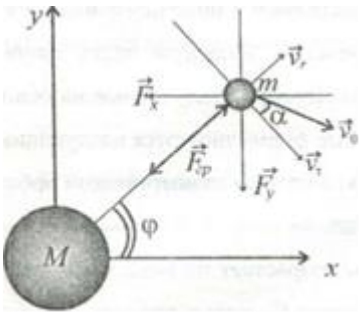


Рис. 1. Схема системы двух планет.

Поскольку масса Земли m = 5.99 10^{24} кг, а масса Солнца M = 1.99 -10^{30} кг, то для большинства практических целей можно считать, что приведенная масса Солнца и Земли равна массе Земли. Поэтому рассмотрим задачу об одной материальной точке массой ш, движущейся вокруг неподвижного силового центра, который примем за начало системы координат.

Закон всемирного тяготения Ньютона утверждает, что частица массой M притягивает другую частицу массой m с силой

$$\vec{F} = -\frac{GMm}{r^3}\vec{r}, \qquad (1)$$

где вектор $\vec{r}$ направлен от тела с массой M к телу с массой m, G - гравитационная постоянная.

У силы тяготения имеется два свойства общего характера: ее величина зависит только от расстояния между телами, а направление совпадает с линией их соединяющей. Такие силы называют центральными. Из данного предположения следует, что орбита Земли лежит в плоскости (x – y), а угловой момент $\vec{L}$ сохраняется и направлен по третьей оси (z). Запишем L_z, используя определение векторного произведения $\vec{L}=\vec{r}\times\vec{p}$, в виде

$$L_z = (\vec{r}\times m\vec{v})_z = m(xv_y - yv_x).$$

Кроме того, движение ограничивается условием сохранения полной энергии E, равной

$$E = \frac{mv^2}{2} - \frac{GMm}{r},$$

Если связать систему координат с телом массой M, то уравнение движения принимает вид:

$$m\frac{d^2\vec{r}}{dt^2} = -\frac{GMm}{r^3}\vec{r}. \qquad (2)$$

Для целей численного моделирования запишем систему (2) в декартовых координатах:

$$\begin{cases} \frac{d^2x}{dt^2} = -\frac{GMx}{r^3} \\ \frac{d^2y}{dt^2} = -\frac{GMy}{r^3} \end{cases}, \qquad (3)$$

где $r^2 = x^2 + y^2$.

Для моделирования часто применяют подход, состоящий в выборе системы единиц, в которой для описания движения, например, Земли, используют в качестве единицы длины большую полуось земной орбиты — астрономическую единицу, которая равна:

1 а.е. =1.496-10^{11} м.

В качестве единицы времени принимается 1 год = $3.15 \cdot 10^{17}$ с. В данных единицах $GM = \frac{4\pi^2 a}{T^2} = 4\pi^2 (a.e.)^3/\text{год}^2$ и траекторию движения планеты определит система (3).

Итак, введем нормированные переменные: $X=x/R$, $Y=y/R$, $\tau=t/T$, где R – радиус орбиты, T – период обращения.

$$\begin{cases} \frac{d^2X}{d\tau^2} = -\frac{GMT^2}{(x^2+y^2)^{\frac{3}{2}} \cdot R^3} X, \\ \frac{d^2Y}{d\tau^2} = -\frac{GMT^2}{(x^2+y^2)^{\frac{3}{2}} \cdot R^3} Y, \end{cases}$$

$$a_{\text{цен}} = \frac{|\vec{v}|^2}{|\vec{R}|}.$$

При движении в гравитационном поле по окружности ускорение $a_{\text{цен}}$ обусловлено гравитационной силой:

$$\frac{m|\vec{v}|^2}{|\vec{R}|} = \frac{GmM}{|\vec{R}|^2}.$$

Следовательно,

$$|\vec{v}| = \left(\frac{GM}{|\vec{R}|}\right)^{1/2} \tag{4}$$

- общее условие для любой круговой орбиты.

Следовательно,

$$T = \frac{2\pi|\vec{R}|}{|\vec{v}|} = 2\pi\sqrt{\frac{|\vec{R}|^3}{GM}}. \tag{5}$$

$$\begin{cases} \frac{d^2X}{d\tau^2} = -\frac{4\pi^2}{(X^2+Y^2)^{\frac{3}{2}}}X, \\ \frac{d^2Y}{d\tau^2} = -\frac{4\pi^2}{(X^2+Y^2)^{\frac{3}{2}}}Y \end{cases} \tag{6}$$

- безразмерные уравнения ни зависят, ни от *R*, ни от *T*.

Так как система (6) является системой с безразмерными переменными, приведем начальные условия к безразмерному виду. Будем считать, что в начальный момент времени тело находилось в точке с радиус-вектором $\vec{r} = (R, 0)$, а скорость тела была направлена вверх $\vec{v}$ = (0,*v*), как показано на рисунке (рис. 2). Выберем замену переменных $\vec{r} = \vec{\rho} \cdot R$, $t = \tau \cdot T$:

$\vec{\rho}$ = (l,0), $\vec{V} = \left(0, v\frac{T}{R}\right)$.

Таким образом, $\vec{V}$ = (0,2π) - дает круговую орбиту.

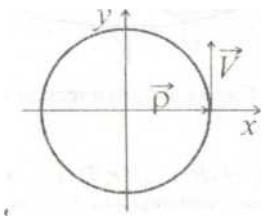


Рис. 2 Схема траектории.

Задание №1 к лабораторной работе

1. Для решения системы обыкновенных дифференциальных уравнений (6) организовать пошаговый метод Эйлера, используя достаточно малое значение шага по времени Δt. Построить зависимости координат от времени, скорости от времени и траекторию движения. Рассмотрите частный случай круговой орбиты. В качестве

примера выберите (в астрономических единицах) $x_0 = 1$, $y_0 = 0$ и $v_x(t = 0) = 0$, $v_y(t = 0) = 2\pi$, $t \in [0:0.0001:2.5]$. Значение шага Δt выберете таким образом, чтобы полная энергия E сохранялась с хорошей точностью.

2. Запустите программу с различными значениями начальных условий, соответствующими условиям круговой орбиты. Для каждой орбиты определит е радиус и период, проверьте правильность третьего закона Кеплера.

3. Выберите такие начальные условия, чтобы получилась подходящая эллиптическая орбита. Определите качественно, как изменяется скорость планеты. В каких точках она максимальна (минимальна)?

4. Выберите точки орбиты *A, B, C, D*, как показано на рис. 2. Измерьте площади заштрихованных фигур *OAB* и *OCD*. Измерьте времена t_{AB}, t_{CD} движения планеты из точки *A* в *B*, из *C* в *D*.

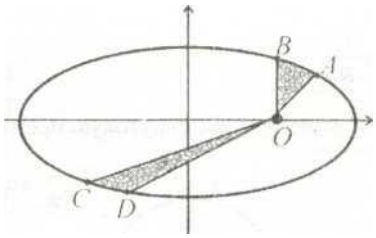


Рис. 2. Схема эллиптической орбиты.

Вычислите отношения OAB / t_{AB}, OCD / t_{CD}. Сравните результаты с теми, которые получаются из второго закона Кеплера.

Выражения, определяющие траекторию движения, можно получить и в полярной системе координат (r, φ):

$$\ddot{r} = \frac{1}{m}\left(\vec{F}_{\text{гр}} + \vec{F}_{\text{цб}}\right)$$

$$\ddot{r} = -G\frac{M}{r^2} + r \cdot \varphi^2$$

$v = \varphi$.

Уравнение для φ определим из закона сохранения момента импульса тела:

$L = mr^2 \cdot \varphi = const$

$$\frac{dL}{dt} = 0$$

$$\frac{dL}{dt} = 2mr \cdot \dot{r} \cdot \dot{\varphi} + mr^2 \cdot \ddot{\varphi} = 0$$

$$\ddot{\varphi} = -\frac{2\dot{\varphi}\cdot\dot{r}}{r}$$

Таким образом, реализация пошагового метода Эйлера для системы (6) эквивалентна реализации этого же метода для системы:

$$\begin{cases} \ddot{\varphi} = -\frac{2\dot{\varphi}\cdot\dot{r}}{r}, \\ \ddot{r} = -G\frac{M}{r^2} + r\cdot\dot{\varphi}^2 \end{cases} \qquad (7)$$

при использовании формул перехода из декартовой системы координат в полярную систему координат:

$$\begin{cases} x = \rho cos\varphi \\ y = \rho sin\varphi. \end{cases}$$

Начальными условиями для построения траектории движения являются: v_0 – начальная скорость; $t^{\times}$ – время полета; r_0 – расстояние до центра гравитации; n – количество шагов.

Задание № 2 к лабораторной работе

1. Для системы (6) организовать пошаговый метод Эйлера для решения дифференциального уравнения, используя достаточно малое значение шага по времени Δt. Графическое представление получить в полярных координатах.

2. Сравнить результаты моделирования с результатами, полученными в задании № 1.

Лабораторная работа № 4

Моделирование колебательных процессов

Линейный гармонический осциллятор

Одним из наиболее распространенных типов механических систем представляют собой малые колебания, которые система совершает вблизи устойчивого положения равновесия. Наиболее простым с механической точки зрения оказывается описание движения систем, имеющих одну степень свободы. Пример такой системы – тело массой *m*, расположенное на гладкой горизонтальной поверхности и прикрепленное к свободному концу пружины жесткостью *k*, *x* – положение тела.

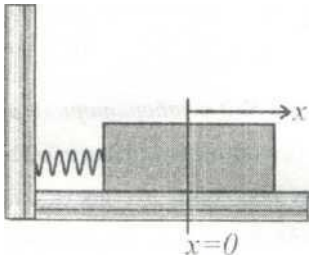


Рис. 1. Схема колебательной системы.

При смещении тела из положения равновесия на небольшое расстояние **x** со стороны пружины будет действовать возвращающая сила: $F = -kx$.

Из 2-го закона Ньютона имеем:

$$m\ddot{x} = -kx, \qquad (1)$$

разделив обе части уравнения на массу *m* и введя обозначение:

$$\omega_0 = \sqrt{\frac{k}{m}} \qquad (2)$$

и уравнение движения линейного гармонического осциллятора запишется в виде:

$$\ddot{x} + \omega_0^2 x = 0, \qquad (3)$$

где ω_0 – циклическая частота.

Полная энергия линейного гармонического осциллятора определяется суммой кинетической и потенциальной энергий:

$$E = \frac{1}{2}mv^2 + \frac{1}{2}kx^2.$$

Аналитическим решением (3) является: $x(t) = A\cos(\omega_0 t + \delta)$, где A – амплитуда, δ – начальная фаза, определяемые из начальных условий для координаты и скорости.

Для определения численного решения ОДУ

$$\frac{d^2x}{dt^2} + \frac{k}{m}x = 0 \text{ с НУ } x_0, y_0 \qquad (4)$$

методом Эйлера необходимо ввести дискретизацию времени с шагом Δt и последовательно определить для каждого временного интервала с номером i,

$$a_i = -\frac{kx_i}{m},$$

$$v_{i+1} = v_i + a_i \cdot \Delta t,$$

$$x_{i+1} = x_i + v_i \cdot \Delta t$$

Задание № 1 к лабораторной работе

1. Построить зависимость координаты, скорости линейного гармонического осциллятора от времени, а также фазовую траекторию осциллятора, используя численное решение задачи (4). Начальные условия и параметры модели выбрать самостоятельно.

2. Построить график зависимости отклонения между численными значениями координаты и скорости и аналитическими.

3. Вычислите полную энергию гармонического осциллятора для тех моментов времени, в которые известны значения координаты и скорости.

Затухающие колебания

В большинстве реальных колебательных систем присутствует трение. Это приводит к тому, что в отсутствие подкачки энергии в колебательную систему амплитуда колебаний уменьшается до ее полной остановки (такие колебания называют затухающими). Пример – движение груза на пружине. При движении с малыми скоростями можно использовать модель, в которой сила трения пропорциональна первой степени скорости движения. В этом случае уравнение движения можно записать в виде:

$$\frac{d^2x}{dt^2} = -\omega_0^2 x - \gamma\frac{dx}{dt}, \qquad (5)$$

γ – коэффициент затухания ($\gamma = \frac{r}{m}$, r - коэффициент вязкости).

Задание № 2 к лабораторной работе

1. При воздействии на линейный гармонический осциллятор внешней переменной силы система будет совершать вынужденные колебания по закону:

$$\frac{d^2x}{dt^2} = -\omega_0^2 x - \gamma\frac{dx}{dt} + \frac{F(t)}{m} \qquad (6)$$

Найдите численное решение для периодически действующей силы $\frac{F(t)}{m} = A_0 cos(\omega t)$. Проведите расчеты и постройте зависимости координаты и скорости от времени для колебательной системы с параметрами ω_0 =3, $x(0)$=1, $v(0)$=0, γ=0.5, A_0=1 ,ω=2.

2. Сравните результаты с результатами, полученными для предыдущего случая. В чем состоит существенное отличие поведения функции $x(t)$ от случая невозмущенного движения?

3. Чему равны период и частота после нескольких колебаний?

4. Исследуйте и опишите особенности движения линейного гармонического осциллятора под влиянием силы $F(t)$:

1) $F(t) = const = F_0$;

2) $F(t) = A\,sin(\beta t)$;

3) $F(t) = F_0\,exp(-\alpha t)$,

считая, что система в начальный момент времени находилась в положении равновесия. Рассмотрите случаи:$\alpha,\beta \ll \omega$,$\alpha,\beta \cong \omega$,$\alpha,\beta \gg \omega$.

Скольжение одного тела по поверхности другого

Результирующая сила, действующая на груз при совершении им колебаний в этом случае:

$$\vec{F} = \vec{F}_{тр}\vec{F}_{упр}$$

(направление $F_{тр}$ зависит от координаты: при растягивании: $F_{тр} \uparrow\downarrow a$, при сжатии: $F_{тр} \uparrow\uparrow a$)

Уравнения движения применительно к данному случаю:

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x - \mu g sign(v), \qquad (7)$$

где μ – коэффициент трения.

Движение тела начнется при условии что: $x_0 > x_{крит}$

При $\overline{Fтр} = \overline{Fупр}$ (a=0 – зона застоя)можно определить $x_{крит}$:

$\frac{k}{m}x_{крит}=\mu g$

$x_{крит}=\frac{\mu g m}{k}$ (-,+)

Задание № 3 к лабораторной работе

Реализовать модель, соответствующую учету сил трения при совершении колебаний, решая ОДУ (7) численными методами и с помощью встроенных функций ППП Matlab, предварительно самостоятельно выбрав константы и параметры моделирования. Оценить $x_{крит}$. Сравнить результаты моделирования с результатами, полученными в задании № 1.

Лабораторная работа № 5. Моделирование статических электрических и магнитных полей

Электрическое поле системы неподвижных зарядов

Электрическое поле, создаваемое неподвижным точечным электрическим зарядом q в вакууме в данной точке пространства, характеризуется скалярным потенциалом:

$$\varphi(\vec{R}) = \frac{1}{4\pi\varepsilon_0}\frac{q}{|\vec{R}-\vec{r}|}, \qquad (1)$$

где $\vec{R}$ – радиус-вектор точки наблюдения;

$\vec{r}$ – радиус-вектор точки, в которой находится электрический заряд;

$\varepsilon_0 = 8.85\cdot10^{-12}$ *Ф/м* – электрическая постоянная.

Векторной характеристикой данного поля является напряженность:

$$\vec{E} = -\vec{\nabla}\varphi(\vec{R}) = \frac{1}{4\pi\varepsilon_0}\frac{q}{|\vec{R}-\vec{r}|^3}(\vec{R}-\vec{r}). \qquad (2)$$

Скалярный потенциал электрической системы, состоящей из N электрических зарядов $q_1, q_2, \ldots q_N$, и напряженность электрического поля удовлетворяют принципу суперпозиции:

$$\varphi(\vec{R}) = \frac{1}{4\pi\varepsilon_0}\sum_{i=1}^{N}\frac{q_i}{|\vec{R}-\vec{r}_i|},$$

$$\vec{E} = \sum_{i=1}^{N}\frac{1}{4\pi\varepsilon_0}\frac{q_i}{|\vec{R}-\vec{r}_i|^3}(\vec{R}-\vec{r}_i),$$

где $\vec{r}_i$ – координата i-го заряда.

При анализе электростатических полей системы произвольно расположенных зарядов, характеризующихся скалярной функцией – потенциалом и векторной функцией – напряженностью, возникает задача наглядного представления этих величин.

Перечислим последовательность действий для представления потенциала электростатического поля:

1) задание функции, которая возвращает значения потенциала, вычисляемые в узлах заданной координатной сетки согласно (3);

2) задание дискретной координатной сетки;

3) вычисление в соответствии с (3) значения $\varphi(\vec{R})$ в каждом узле сетки;

4) построение графика поверхности и карты эквипотенциалей.

Функция Potential возвращает матрицу размерности *Ny***Nx*, содержащую значения потенциала в соответствующих узлах сетки.

Вычислим потенциал, создаваемый системой, состоящей из *N*=6 электрических зарядов, расположенных в точках с координатами $x_i = \left(-5 \cdot R0 + 2 \cdot R0 \cdot (i-1)\right) = -5R0{:}2 * R0{:}5R0, y_i = 0, i = \overline{1,N}$.

Пример расположения подобной цепочки зарядов представлен на рис.1.

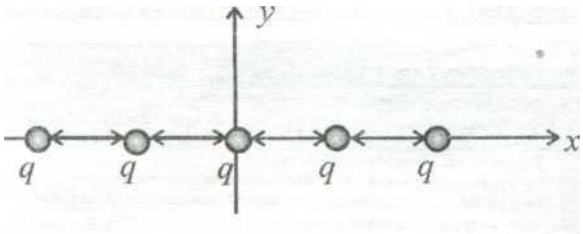


Рис. 1. Схема системы зарядов.

Для проведения вычислений в качестве единиц измерения заряда мож но использовать элементарный заряд $e = 1.6 \cdot 10^{-16}$ *Кл*, единиц измерения длины - $R0 = 10^{-6}$ *м*.

Решение задачи оформим файлом-программой:

Editor - C:\Program Files\MATLAB\R2017b\bin\work\Calculation_Potential.m

Calculation_Potential.m

```
Elem=1.6*10^(-19);
R0=10^(-6);
N=6;
i=1:N;
q(i)=Elem;
x1=-5*R0;
x2=5*R0;
x1q(i)=x1+2*R0*(i-1);
y1q(i)=0;
NN=11;
Xmin=-10*R0;
Ymin=-10*R0;
Xmax=10*R0;
Ymax=10*R0;
h=1:NN;
X(h)=Xmin+2*(h-1)*R0;
s=1:NN;
Y(s)=Ymin+2*(s-1)*R0;
F(h,s)=Potential(q,x1q,y1q,X,Y);
[T,R]=meshgrid(X,Y);
surf(T,R,F)
%shading interp
%contour(T,R,F,20)
```

Editor - C:\Program Files\MATLAB\R2017b\bin\work\Potential.m

Calculation_Potential.m | Potential.m

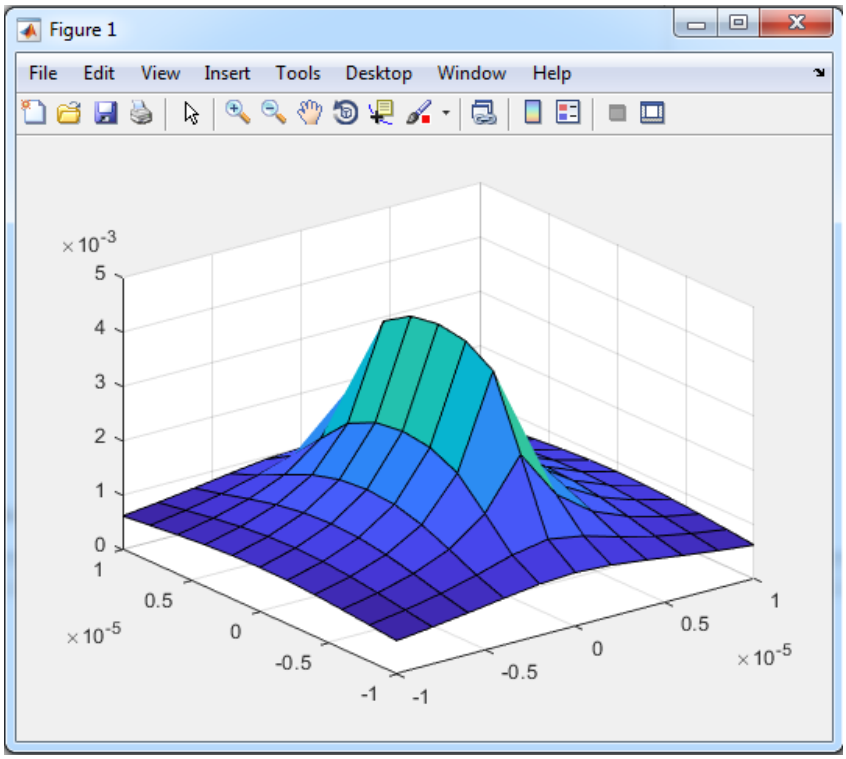
```
function Phi=Potential(q,xq,yq,X,Y)
%q-вектор, содержащий значения электрических зарядов
%xq,yq-векторы, содержащие координаты электрических зарядов
%X,Y-векторы, задающие сетку
Eps=8.85*10^(-12);
Nq=length(q);
Nx=length(X);
Ny=length(Y);
for i=1:Nx
    for j=1:Ny
        s=0;
        for k=1:Nq
            s=s+q(k)/((X(i)-xq(k)).^2+(Y(j)-yq(k)).^2).^0.5;
        end
        M(i,j)=s/(4*pi*Eps);
    end
end
Phi=M;
```

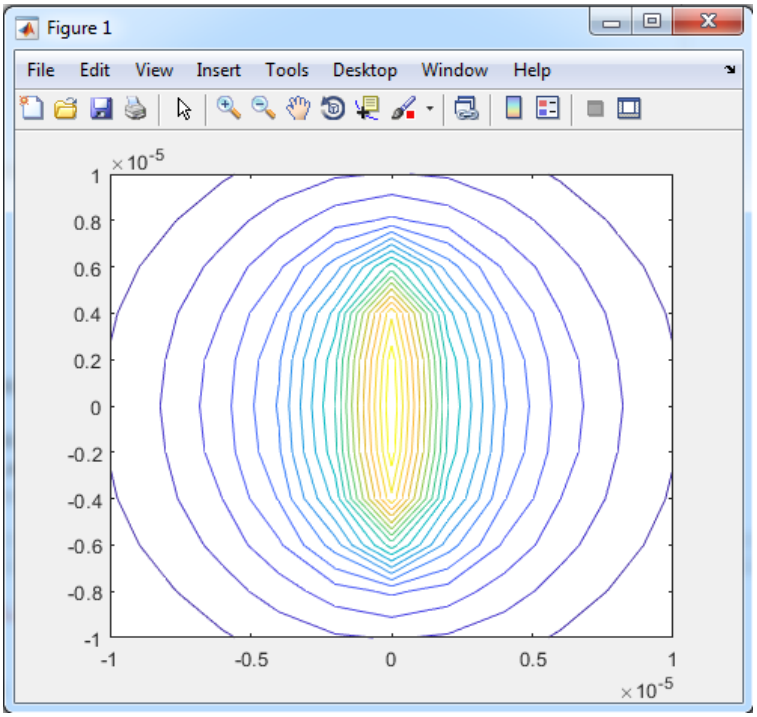
Editor - C:\Program Files\MATLAB\R2017b\bin\work\Calculation_Potential.m

Calculation_Potential.m

```
Elem=1.6*10^(-19);
R0=10^(-6);
N=6;
i=1:N;
q(i)=Elem;
x1=-5*R0;
x2=5*R0;
x1q(i)=x1+2*R0*(i-1);
y1q(i)=0;
NN=11;
Xmin=-10*R0;
Ymin=-10*R0;
Xmax=10*R0;
Ymax=10*R0;
h=1:NN;
X(h)=Xmin+2*(h-1)*R0;
s=1:NN;
Y(s)=Ymin+2*(s-1)*R0;
F(h,s)=Potential(q,x1q,y1q,X,Y);
[T,R]=meshgrid(X,Y);
surf(T,R,F)
%shading interp
%contour(T,R,F,20)
```

Результат выполнения команд представлен на следующих рисунках:





В отличие от функции $\varphi(\vec{R})$ напряженность электростатического поля $\vec{E}(\vec{R})$ является векторной функцией, которая в каждой точке пространства характеризуется величиной поля и направлением. Для описания векторного поля используем силовые линии (линии, касательные к которым в каждой точке па¬раллельны вектору напряженности электрического поля). Для исследования особенностей

напряженности электрического поля, создаваемого произвольной конфигурацией электрических зарядов, необходимо выполнить последовательность действий:

1) задать в пространстве дискретную координатную сетку;

2) вычислить в узлах сетки координаты напряженности электрического поля $\vec{E}(\vec{R}) = (E_x, E_y)$;

3) построить в каждом узле сетки единичные вектора $\vec{n} = \left(\frac{E_x}{|\vec{E}|}, \frac{E_y}{|\vec{E}|}\right)$, касательные к силовой линии электрического поля;

4) построить график поверхности и карту линий уровня функции $|\vec{E}(\vec{R})|$.

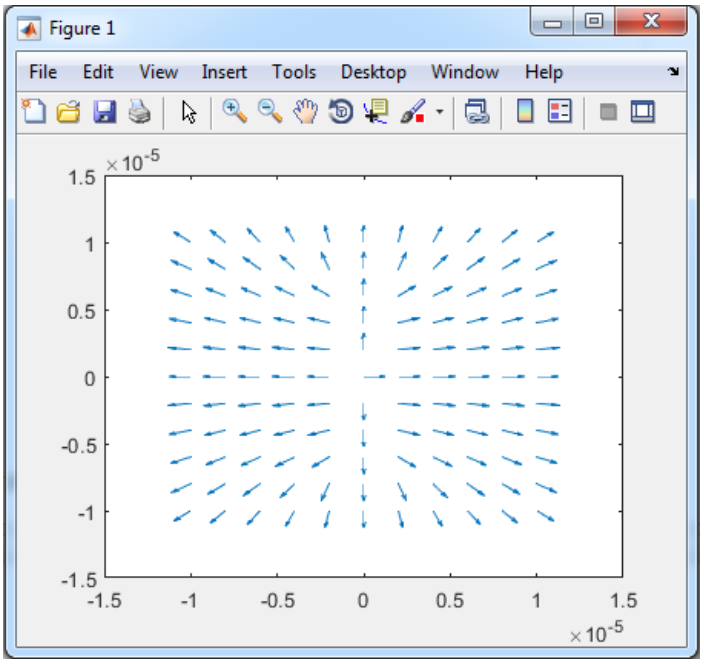
Данный алгоритм реализуем в Matlab в виде следующей подпрограммы (начало полностью дублирует вид подпрограммы Calculation_Potential):

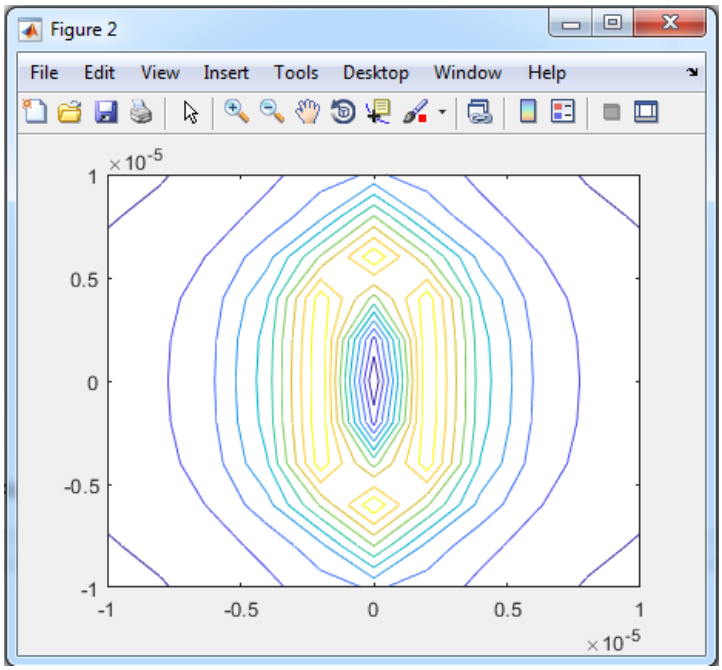
Editor - C:\Program Files\MATLAB\R2017b\bin\work\Electrostatic_Intensity.m*

Calculation_Potential.m | Potential.m | Electrostatic_Intensity.m*

```
h=1:NN;
X(h)=Xmin+2*(h-1)*R0;
s=1:NN;
Y(s)=Ymin+2*(s-1)*R0;
F(h,s)=Potential(q,x1q,y1q,X,Y);
[T,R]=meshgrid(X,Y);
%вычисление градиента функции Phi(R)
[px,py]=gradient(-F,0.1,0.1);
%вычисление координат единичных векторов,
%касательных к силовой линии
px1=px./(px.^2+py.^2).^0.5;
py1=py./(px.^2+py.^2).^0.5;
%построение векторного поля единичных векторов,
%касательных к силовой линии в узлах сетки
figure(1);
quiver(T,R,px1,py1,0.5);
%вычисление абсолютных знаний вектора напряженности
%в узлах сетки
mp=(px.^2+py.^2).^0.5;
figure(2);
contour(T,R,mp,10);
```

Результаты моделирования представлены на следующих рисунках:





Задание № 1 к лабораторной работе

1. Создайте программу, позволяющую строить потенциал и напряженность электрического поля протяженного диполя, т.е. диполя, состоящего из N (>10) зарядов, знаки которых последовательно чередуются.

2. Сравнить электрическое поле при четном и нечетном числе зарядов, составляющих диполь, в ближней и дальней зонах.

3. Сравнить электрическое поле с электрическим полем точечного заряда.

4. Исследуйте зависимость электрического поля протяженного диполя в ближней и дальней зонах от расстояния между зарядами.

Магнитное поле витка с постоянным током

Изучение электрических полей систем неподвижных электрических зарядов основано на законе Кулона и принципе суперпозиции. Аналогом закона Кулона в электростатике в магнитостатике является закон Био-Савара-Лапласа, в соответствии с которым вектор магнитной индукции магнитного поля, создаваемого элементом тока $Id\vec{l}$, имеет вид:

$$d\vec{B} = \frac{\mu_0}{4\pi} I \frac{[d\vec{l}\times(\vec{R}-\vec{r})]}{|(\vec{R}-\vec{r})|^3}, \qquad (3)$$

где $\vec{R}$ – радиус-вектор точки наблюдения;

$\vec{r}$ – радиус-вектор точки нахождения проводника;

$\vec{B}$ – индукция магнитного поля;

$\mu_0 = 4\pi \cdot 10^{-7}$ Тл·м/А – магнитная проницаемость вакуума.

Для вычисления основной силовой характеристики магнитного поля вектора $\vec{B}$, создаваемого произвольной системой постоянных токов, согласно принципу суперпозиции:

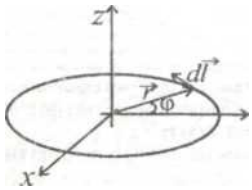
$$d\vec{B} = \frac{\mu_0}{4\pi} I \sum_{i=1}^{N} \frac{[d\vec{l}_i\times(\vec{R}-\vec{r}_i)]}{|(\vec{R}-\vec{r}_i)|^3}, \qquad (4)$$

где N – число элементов;

$\vec{r}_i$ – радиус-вектор соответствующего элемента.

Вычислить магнитное поле аналитически, используя (3-4), оказывается возможным только для структур, имеющих достаточно высокую степень симметрии (прямой провод, провод кольца и т.п.). Для большинства конфигураций расчет магнитного поля может быть проведен только численно. Наиболее важные токовые конфигурации структур: прямой провод, петля, соленоид, тороидальная обмотка. В качестве примера рассмотрим модель петли с током.

Для расчета индукции магнитного поля петли выберем СК, в которой плоскость XOY совпадает с плоскостью петли.



В выбранной СК положение *i*-го элемента тока:

$d\vec{l}_i = a\begin{bmatrix} -sin\varphi_i \\ cos\varphi_i \\ 0 \end{bmatrix} d\varphi$, где $d\varphi = \frac{2\pi}{N}$, $\varphi_i = id\varphi$, i=0,1,…N–1, $\vec{r}_i = \begin{bmatrix} cos\varphi_i \\ sin\varphi_i \\ 0 \end{bmatrix}$ – радиус-вектор, описывающий *i*-ый элемент тока, *a* - радиус кольца.

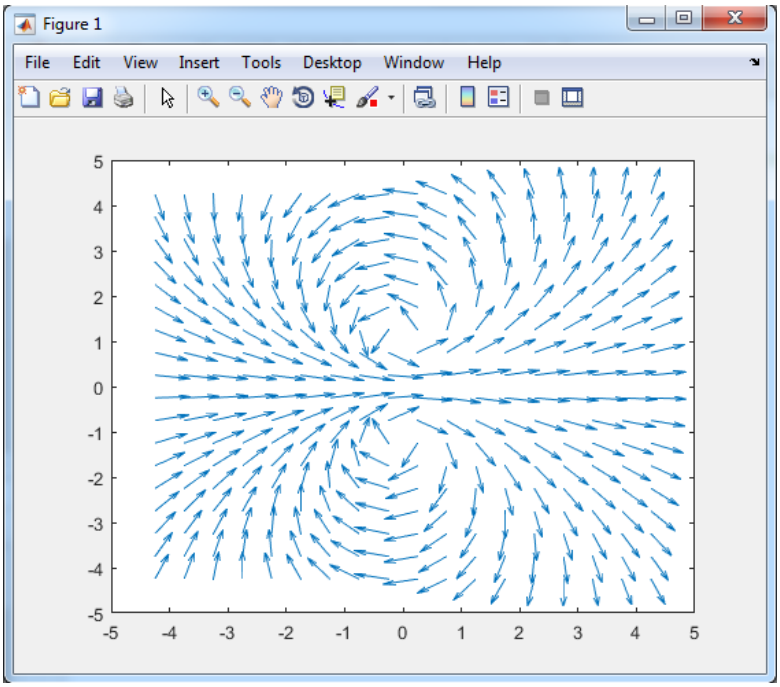
Для расчета составляющих вектора $\vec{B}$ магнитного поля воспользуемся не аналитическими преобразованиями, а численным вычислением на основе применения встроенных функций Matlab. В рассматриваемой плоскости (z=0) составляющая магнитной индукции равна нулю, поэтому будем проводить вычисления составляющих в одной из перпендикулярных плоскостей, например, в плоскости XOZ. Поскольку интересна форма силовых линий, а не абсолютное значение магнитной индукции, множитель $\frac{\mu_0}{4\pi} I$ в (4) нормируем и примем его значение равным единице. При вычисления шаг сетки следует выбирать таким образом, чтобы знаменатель (4) не обращался в ноль.

Создадим файл MagForce.m, возвращающий значения составляющих вектора магнитной индукции:

```
Calculation_MagForce.m   MagForce.m
1   function [Bx,Bz]=MagForce(a,X,Z)
2   %X0Z - плоскость наблюдения (расчета) магнитного поля
3   %X0Y - плоскость положения кольца
4   Nstep=100;%кол-во элементов, аппроксимирующих кольцо
5   Y=0;%координаты для плоскости наблюдения - X0Z
6   Nx=length(X);%число узлов координатных сеток
7   Nz=length(Z);
8   deltaphi=2*pi/Nstep;%шаг по углу
9   n=1:Nstep+1;
10  phi(n)=deltaphi*(n-1);%значение текущего угла в рад.
11  %вычисление составляющих вектора магнитной индукции
12  for i=1:Nx
13      for j=1:Nz
14          s=[0 0 0];
15          for h=1:Nstep+1
16              dL=[-a*sin(phi(h))*deltaphi a*cos(phi(h))*deltaphi 0];
17              r=[a*cos(phi(h)) a*sin(phi(h)) 0];
18              R=[X(i) Y Z(j)];
19              s=s+(cross(dL,(R-r)))./(((R-r)*(R-r)').^(3/2));
20          end
21          bx(i,j)=s(1);
22          bz(i,j)=s(3);
23      end
24  end
25  Bx=bx;
26  Bz=bz;
```

Расчет и визуализацию векторного поля оформим файлом-программой:

```
Editor - C:\Program Files\MATLAB\R2017b\bin\work\Calculation_MagForce.m
Calculation_MagForce.m   MagForce.m
1   NN=18;%число узлов сетки
2   i=1:NN;
3   %задаем координаты сеток по 0X и 0Z, плоскость наблюдения - X0Z
4   XX(i)=-4.25+0.5*(i-1);
5   ZZ(i)=-4.25+0.5*(i-1);
6   a=1;%радиус кольца
7   [Bx,Bz]=MagForce(a,XX,ZZ);
8   %вычисление значения модуля вектора магнитной индукции
9   %в узлах сетки
10  mp=(Bx.^2+Bz.^2).^0.5;
11  %вычисление координат единичных векторовб касательных
12  %к силовой линии магнитного поля, в узлах сетки
13  bx1=Bx./mp;
14  bz1=Bz./mp;
15  %визуализация векторного поля
16  quiver(ZZ,XX,bz1,bx1);
```



Задание № 2 к лабораторной работе

Провести вычисления магнитного поля витка с постоянным током, имеющего форму эллипса с полуосями a = 10 *см*, b = 12 *см*, по которому проходит ток $I = 10^{-5}$ *А*. Контур располагается в плоскости $x = 0$ – YOZ. Плоскость для вычисления компонент вектора магнитной индукции выбрать самостоятельно. Моделирование провести в размерных физических единицах. Оформить отчет по лабораторной работе.

Лабораторная работа № 6. Моделирование случайных величин и случайных событий

Важным классом реализуемых на ЭВМ моделей являются стохастические (или недетерминированные) модели, в которых реализация рассматриваемого процесса зависит от случайных параметров. Выходные значения для таких моделей при заданном наборе входных данных можно предсказать только в вероятностном смысле.

Основу моделирования в этом случае составляет использование равномерно распределенных случайных чисел. Равномерно распределенные случайные числа заключены в интервале от 0 до 1 и выбираются случайным образом в соответствии с функцией распределения

$F(x) = \Pr\{X < x\} = x,\ 0 < x < 1.$

Здесь $\Pr\{X < x\}$ — вероятность того, что случайная величина X примет значение меньше x.

При равномерном распределении одинаково правдоподобно появление любых значений случайной величины в интервале (0, 1). Основным методом получения случайных чисел является их генерация по модулю. Таким: образом, мы имеем дело с «псевдослучайными числами». Пусть m, a, c, x_0 — целые числа, такие, что $m > x_0$, $a, c, x_0 > 0$. Псевдослучайное число x_i, из последовательности $\{x_i\}$ получается с помощью рекуррентного соотношения:

$$x_i = ax_{i-1} + c(mod\ m).$$

Стохастические характеристики генерируемых чисел решающим образом зависят от выбора m, a, c. Неудачный выбор этих параметров приводит к ошибочным результатам, например, при моделировании методом Монте-Карло. Этим термином обозначается большая группа методов, в которых используются генераторы случайных чисел. Метод Монте-Карло используется для решения различных математических задач: интерполяции, вычисления интегралов, решения дифференциальных и интегральных уравнений, поиска экстремума.

Поскольку генерация случайных чисел осуществляется по формуле вида $\gamma_{k+1} = F(\gamma_k)$, а в коде программы можно записать лишь конечное число различных чисел, начиная с некоторого номера члены последовательности будут повторяться. Для

численного моделирования часто требуется большое количество случайных чисел, поэтому период последовательности генерируемых случайных чисел, после которого последовательность начинает повторяться, должен быть достаточно большим. Он должен быть существенно больше требуемого для моделирования количества случайных чисел, иначе получаемые результаты будут искажены.

Программные оболочки, как правило, содержат генератор случайных чисел. Однако большинство статистических тестов показывает коррелированность между получаемыми случайными числами.

В Matlab для генерации псевдослучайных чисел, равномерно распределенных в интервале от 2^{-53} до $1-2^{-53}$ (примерно (0,1)), используют функцию rand. Теоретически функция rand позволяет без повторений сгенерировать 2^{1492}, rand(n) возвращает квадратную матрицу n-го порядка, элементами которой являются случайные числа. Для перехода от дробных чисел к числам из интервала [*a*,*b*] достаточно воспользоваться преобразованием *a*+(*b*-*a*)*rand. Функция randn позволяет сгенерировать случайные числа, распределенные по нормальному закону со следующими параметрами:

1) среднее значение случайной величины равно 0; $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$;

2) дисперсия $\sigma^2 = 1$; $\sigma^2 = \frac{1}{N}\left(\sum_{i=1}^{N} x_i^2 - \bar{x}^2\right)$;

3) средняя квадратичная ошибка $\sigma = 1$.

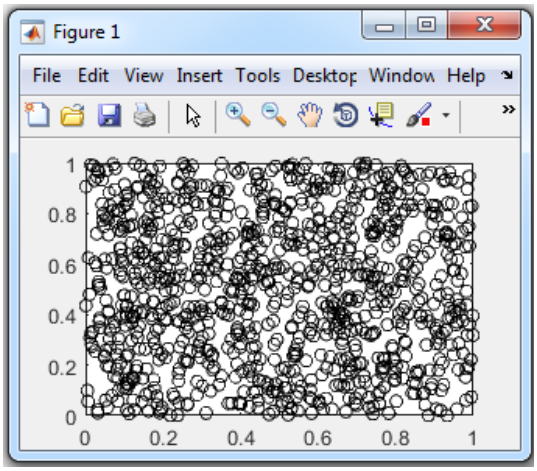
В математической статистике используют различные числовые характеристики случайных величин. Различают генеральные и выборочные характеристики. Наиболее важными характеристиками являются дисперсия и среднее значение.

Существует быстрый тест, с помощью которого необходимо проверять каждый генератор. Качество генератора случайных чисел можно продемонстрировать, заполняя полностью (*d*-мерную решетку (например, двух- и трехмерную, т. е. квадрат или куб). Хороший генератор должен заполнить все пространство такой ограниченной области (гиперкуба).

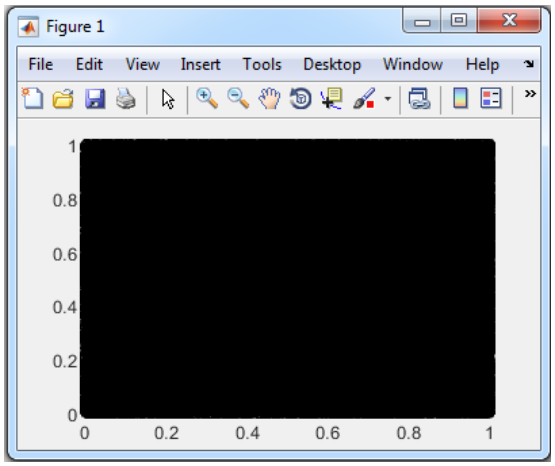
Результаты подобного теста в Matlab:

```
>> n=100000;
>> X=rand(1,n);
```

```
>> Y=rand(1,n);
>> plot(X,Y,'ko');
```



а (при *n*=1000)



б (при *n*=100000)

Следует отметить хорошее качество данного генератора. Для более объективной оценки качества подобных генераторов существует множество критериев.

Пример 1.

В качестве примера применения метода Монте-Карло рассмотрим вычисление интегралов (наиболее эффективно применение данного метода для вычисления многомерных интегралов, когда прямое интегрирование обычными численными методами невозможно). Основная идея метода заключается в том, чтобы вычислять подынтегральную функцию не в каждом узле интегрирования, поскольку число таких узлов может быть очень велико, а только в точках, являющихся репрезентативной случайной выборкой. Согласно теореме математического анализа, интеграл определяется средним значением подынтегральной функции на отрезке интегрирования. Таким образом, оценка интеграла методом «выборочного среднего»:

$$I = (b-a)\langle f\rangle = (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i),$$

где x_i – случайные числа, равномерно распределенные на отрезке [a,b];

N – количество испытаний.

Предположим, что нам нужно вычислить интеграл в области

$\Omega = \{(x, y)\colon 0 \le x \le 1, 1 \le y \le 2\}\ dxdy$

и сравним результат с точным значением, равным 8/3.

Реализация в Matlab

Editor - C:\Program Files\MATLAB\R2017b\bin\work\fMK.m

fMK.m | MK1.m

```
function z=fMK(x,y)
z=x.^2+y.^2;
```

Editor - C:\Program Files\MATLAB\R2017b\bin\work\MK1.m

fMK.m | MK1.m

```
n=5000;
X=0+(1-0)*rand(1,n);
Y=1+(2-1)*rand(1,n);
S=0;
J=0;
for i=1:n
    S=S+fMK(X(i),Y(i));
end
J=(1/n)*S;
disp(J);
eps=J-8/3;
disp(eps);
```

Результат работы программы:

```
>> MK1
    2.6738
    0.0071
```

Пример 2.

Интересным случайным процессом является броуновское движение. В 1827 г. английский ботаник Броун наблюдал под микроскопом хаотическое движение мелких частиц в капле жидкости. Математическая теория этого процесса была построена гораздо позже А. Эйнштейном, а затем Н. Винером. Если броуновская частица является достаточно мелкой, то количество столкновений молекул воды с данной частицей не будет одним и тем же по всей площади поверхности частицы, что и приводит к ее хаотичному движению.

Рассмотрим двумерное броуновское движение. Пусть $\xi_1(t)$ и $\xi_2(t)$ - координаты частицы в момент времени t. Очевидно, что $\xi_1(t)$ и $\xi_2(t)$ являются случайными величинами. Можно допустить, что эти величины распределены по нормальному закону с нулевым математическим ожиданием и дисперсией $\sigma^2(t) = \sigma^2 t$, где константа σ^2 называется коэффициентом диффузии.

Моделирование броуновского движения в пакете MathCad показано на рисунке. Разность координат частицы при каждом последующем шаге задается функцией

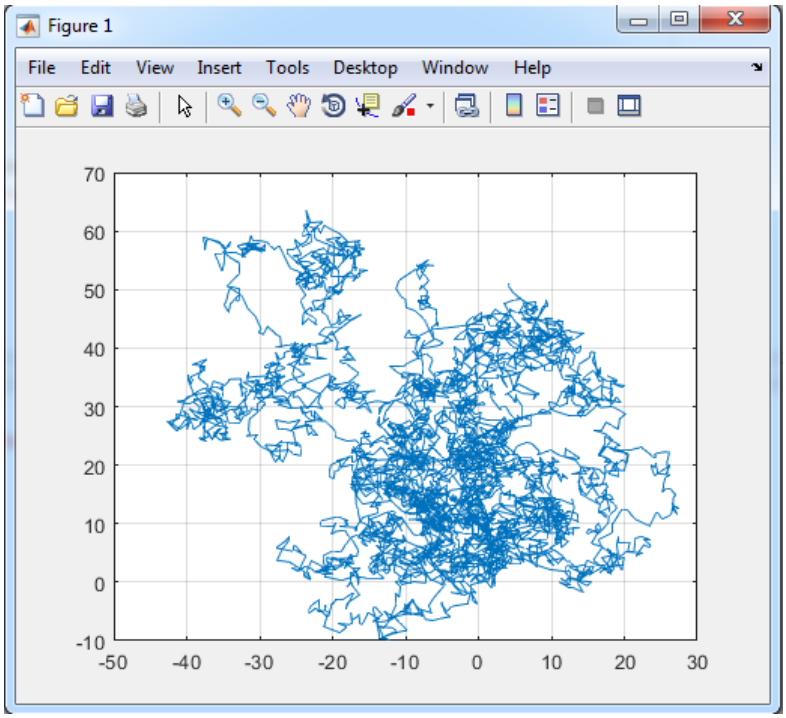
нормального распределения **rnorm(*n*,*a*,σ)**, где *n* — число координат случайного вектора (в нашем случае – 2, поскольку мы рассматриваем двумерное движение), а ***a***,**σ** – среднее и дисперсия нормального распределения.

Реализация задачи в Matlab:

Editor - C:\Program Files\MATLAB\R2017b\bin\work\MK2.m

fMK.m | MK1.m | MK2.m

```
n=5000;
X=rand(1,n);
Y=rand(1,n);
X(1)=0;
Y(1)=0;
for i=2:n
    X(i)=X(i-1)+randn;
    Y(i)=Y(i-1)+randn;
end
plot(X,Y);
grid on;
```



Задание к лабораторной работе

Реализовать двумерную и трехмерную модели броуновского движения, используя возможности эффекта анимации в Matlab.

Лабораторная работа № 7. Реализация математических моделей физических процессов, сводимых к решению нелинейных уравнений

Одна из простейших задач, часто возникающая при математическом моделировании различных систем, – поиск приближенных значений корней нелинейных уравнений.

Рассмотрим задачу приближенного нахождения корней уравнения вида:

$$f(x)=0, \qquad (1)$$

где $f : R_1, \longmapsto R_1$ – алгебраическая или трансцендентная функция. Аналитические методы решений скалярных уравнений (1) существуют только в отдельных классах уравнений. В общем случае можно говорить лишь о приближенном вычислении корней уравнения (1), т.е. таких значений аргумента x=ξ, при которых равенство $f(\xi) = 0$ истинно. При этом под близостью приближенного значения x к корню уравнения (1) понимают выполнение неравенства:

$$|\xi - \bar{x}| < \varepsilon. \qquad (2)$$

При малых $\varepsilon > 0$ часто важно контролировать не абсолютную погрешность, а относительную: $\frac{|\xi-\bar{x}|}{|\bar{x}|}$.

Локализация корней

Нелинейная функция $f(x)$ в своей области определения $D(f) \subseteq R_1$ может иметь конечное или бесконечное количество нулей или не иметь их вообще. Большинство же методов нахождения корней требует знания промежутков, где имеется и притом единственный нуль функции. Таким образом, ставится подзадача существования и единственности, нахождения границ и локализации (изоляции, отделения) корней. Для функций общего вида нет универсальных способов решения поставленных подзадач. На практике используют следующий подход.

Графический способ локализации корней. Если по геометрической интерпретации функции $f(x)$ можно однозначно представить ситуацию, касающуюся количе-

ства и расположения нулей, то локализацию производят, отделяя те промежутки на оси абсцисс, где график функции $y = f(x)$ пересекает ось Ox.

Если построение графика функции $y = f(x)$ вызывает затруднение, но исходное уравнение (1) очевидным образом представляется в виде $f_1(x) = f_2(x)$ и функции $f_1(x)$ и $f_2(x)$ таковы, что построение их графиков возможно, задача определения корней и областей их единственности решается отслеживанием точек пересечения этих графиков и выделением на оси абсцисс тех промежутков, которым принадлежат проекции данных точек.

Пример 1. Найти промежутки локализации корней уравнения
$x^2 - \sin x - 1 = 0$.

Представив уравнение в виде $x^2 - 1 = \sin x$, строим графики функций $y_1 = x^2 - 1$ и $y_2 = \sin x$, как показано на рис. 1. Совместное рассмотрение графиков позволяет сделать заключение, что данное уравнение имеет два корня:
$\xi_1 \in [-1,0]$ и $\xi_2 \in [1,\pi]$.

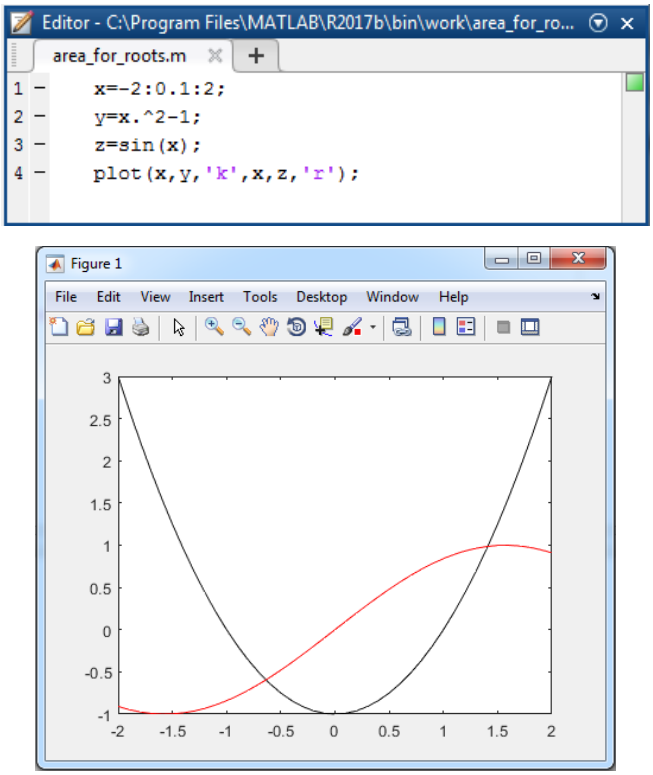


Рис. 1. Графическая локализация корней уравнения $x^2 - \sin x - 1 = 0$.

Пример математической модели, сводимой к решению нелинейного уравнения

В качестве более сложного примера рассмотрим применение стандартных функций нахождения корня нелинейного уравнения в задаче квазиклассического квантования колебаний двухатомной молекулы, если потенциальная энергия определяется потенциалом $v(x)$, в частности, потенциалом Ленарда-Джонса. Из законов квантовой механики следует, что в этом случае энергию колебаний молекулы можно найти из уравнения

$$s(\varepsilon_n) = \gamma \int_{x_{in}}^{x_{out}} [\varepsilon_n - v(x)]^{1/2} dx = \left(n + \frac{1}{2}\right)\pi.$$

Данное уравнение записано в безразмерных переменных

$$\varepsilon = \frac{E}{V_0}, x = \frac{r}{a}, \gamma = \left(\frac{8\pi^2 m a^2 V_0}{h^2}\right)^{1/2},$$

где E — энергия связанного (дискретного) состояния; а, V_0 – параметры потенциала Ленарда-Джонса; r – расстояние между ядрами молекул; m – приведенная масса двух ядер; h – постоянная Планка. Величина s называется действием. Потенциал Ленарда-Джонса в безразмерных переменных имеет вид

$$v(x) = 4\left(\frac{1}{x^{12}} - \frac{1}{x^6}\right).$$

Параметры x_{in} и x_{out} соответствуют точкам, в которых подынтегральное выражение обращается в ноль.

Число связанных состояний в молекуле водорода H_2 можно определить, задавая очень малую энергию ε и определяя, какому n она соответствует. Для этого необходимо задать ошибку в определении энергии tol$E$ (например, =0,005) и найти точки поворота, приняв ε = –tolE и решая уравнение $\varepsilon - v(x) = 0$. Найдя точки поворота x_{in} и x_{out}, с помощью решения нелинейного уравнения, определяем величину действия $s(\varepsilon)$. После этого определяем число связанных состояний N по формуле

$$N = \frac{s}{\pi} - \frac{1}{2} + 1.$$

Примем γ = 21,27, что характерно для молекулы водорода H_2. По результатам расчета этой модели число связанных состояний равно 6, что плохо согласуется с экспериментальными данными, согласно которым таких состояний в молекуле водорода 15. Один из способов повышения адекватности данной модели – замена потенциала Ленарда-Джонса более реалистичным потенциалом Морса:

$$V(x) = \left(1 - e^{-\beta(x-1)}\right)^2 - 1.$$

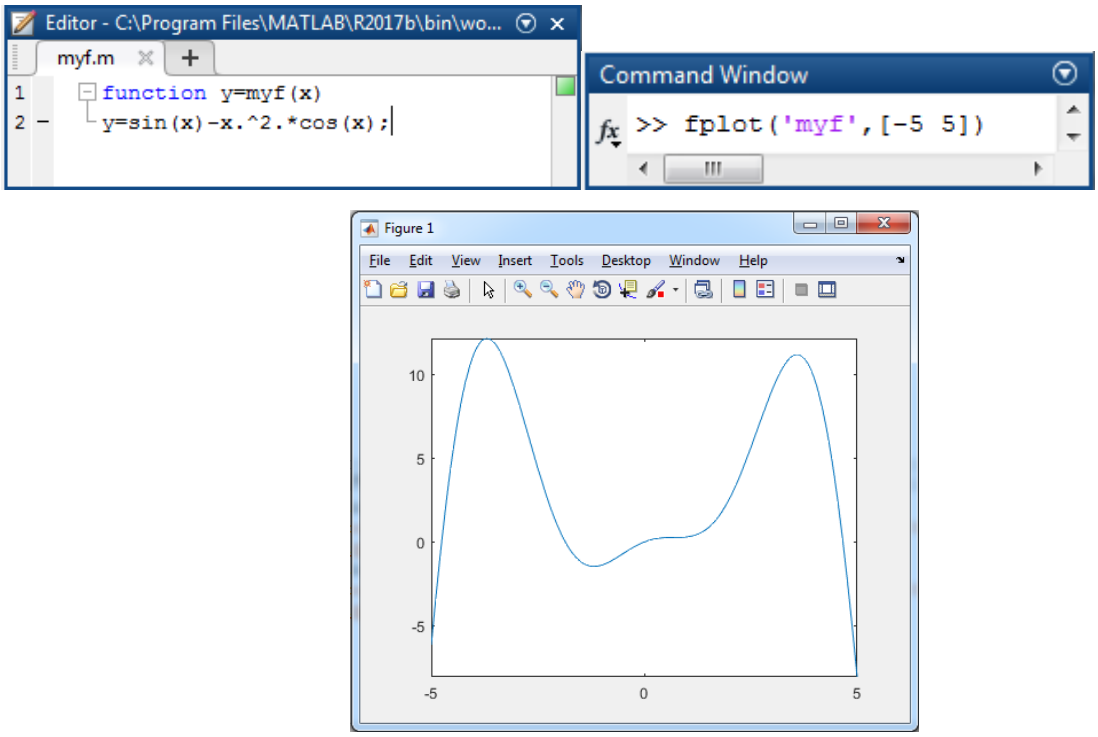
При подходящем выборе параметра β (=1.5) аналогичный расчет дает «правильное» значение – 15.

Численные примеры. Реализация в пакете Matlab

Функция **fzero** позволяет приближенно вычислить корень уравнения на некотором интервале или ближайший к заданному начальному приближению. В простейшем варианте **fzero** вызывается с двумя входными и одним выходным аргументом **x=fzero('func_name',x0)**, где **func_name** – имя файл-функции, вычисляющей левую часть уравнения, **x0** – начальное приближение к корню, **x** – найденное приближенное значение корня.

Пример 2. Решить на отрезке [-5,5] уравнение $\sin x - x^2 \cos x = 0$.

Выполним локализацию корня.



Найдем значение корня вблизи x= –5.

Для того чтобы увидеть больше значащих цифр, установите формат **long**, алгоритм функции **fzero** находит корень уравнения с точностью **eps**.

```
Command Window
>> x1=fzero('myf',-5)

x1 =

   -4.7566

>> format long
>> x1

x1 =

  -4.756559405702904

>> eps

ans =

     2.220446049250313e-16

>>
```

Важной особенностью функции **fzero** является то, то она вычисляет только те корни, в которых функция меняет свой знак.

Optimization Toolbox позволяет решить нелинейные уравнения с помощью функции **fsolve**. В общем случае вызов этой функции со всеми принятыми по умолчанию настройками имеет вид **x=fsolve(‘func_name’,x0, optimset(‘fsolve’))**

Пример 4. Решить уравнение из примера 3.

```
Command Window
>> x1=fsolve('myf',-5,optimset('fsolve'))

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the selected value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>


x1 =

  -4.756559405702904

>>
```

Для нахождения корней полинома в пакете Matlab предусмотрена соответствующая функция **roots()**, возвращающая вектор-столбец, компоненты которого являются корнями полинома.

Пример 3. Найти корень уравнения $x^3 + 8x^2 - 14x - 20 = 0$, используя функцию **roots()**.

```
Command Window
>> c=[1 8 -14 -20];
>> roots(c)

ans =

   -9.2767
    2.2395
   -0.9627

>>
```

Задание к лабораторной работе

Определить значение критической силы для стержня один конец которого защемлен, а другой может перемещаться в вертикальном направлении. Критическая сила в этом случае определяется трансцендентным уравнением:

$tg\sqrt{\frac{P}{EI}}L = \sqrt{\frac{P}{EI}}L$, где P – критическая сила, при которой стержень теряет устойчивость, EI – изгибная жесткость стержня, L — его длина. (Принять в нормировке EI=1 и L=1, где i – номер варианта.)

1. Используя геометрический метод, найти область существования решения уравнения. Найти корень уравнения $f(x) = 0$ в найденной области, используя возможности пакета Matlab.

2. Найти корень уравнения $f(x) = 0$ в найденной области, используя возможности пакета Matlab.

Лабораторная работа № 8. Обработка экспериментальных данных. Интерполирование функций

Пусть функция $f(x)$ задана множеством своих значений для дискретного набора точек:

x_0	x_1	…	x_n
f_0	f_1	…	f_n

здесь $f_i = f(x_i)$.

Требуется найти интерполяционный многочлен $P(x) = P_n(x)$ степени не выше п, значения которого в узлах интерполяции x, совпадают со значениями данной функции $P(x_i) = f_i$.

Интерполяционный полином Лагранжа

Для функции $f(x)$, заданной таблицей, интерполяционный *многочлен Лагранжа* имеет вид

$$L_n(x) = \sum_{i=1}^{n} \frac{(x-x_0)(x-x_1)\cdot\ldots\cdot(x-x_{i-1})(x-x_{i+1})\cdot\ldots\cdot(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdot\ldots\cdot(x_i-x_{i-1})(x_i-x_{i+1})\cdot\ldots\cdot(x_i-x_n)}.$$

Сплайн-интерполяция

При большом количестве узлов интерполяции приходится использовать полиномы высокой степени. Можно этого избежать, разбив отрезок интерполяции на несколько частей и построив на каждой части свой интерполяционный многочлен. Существенный недостаток такого интерполирования состоит в том, что в точках сшивки разных интерполяционных полиномов их первая производная будет разрывной, поэтому для решения задачи кусочно-линейной интерполяции используют особый вид кусочно-полиномиальной интерполяции – сплайн-интерполяцию.

Сплайн – это функция, которая на каждом частичном отрезке интерполирования является алгебраическим многочленом, а на заданном отрезке непрерывна вместе с несколькими своими производными.

Численные примеры. Реализация в пакете Matlab

Пример 1. Решить задачу интерполяции для функции $f(x)=\sin(x)$, заданной таблично на интервале $[0, 2\pi]$ при $n = 8$ средствами пакета Matlab.

Для решения задачи одномерной интерполяции в пакете Matlab используется функция **interp1(x,y,xi,'interp_type')**. Для использования этой функции необходимо задать точеные массивы таблично-заданной функции, координаты абсцисс промежуточных точек и способ интерполирования.

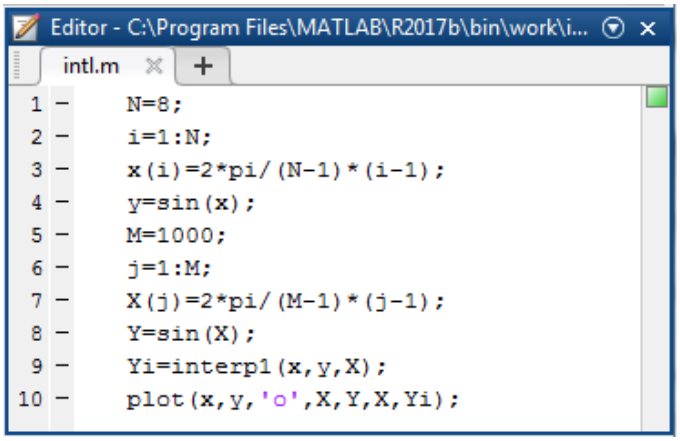
'nearest' – приближение по соседним элементам,

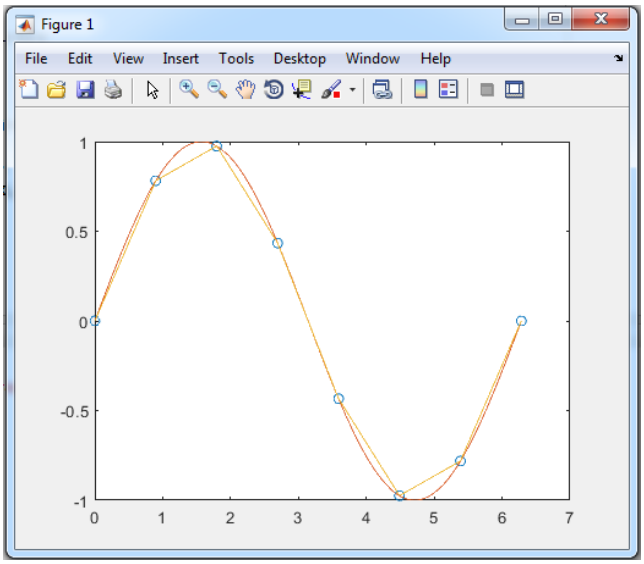
'linear' – линейная интерполяция,

'spline' – интерполяция кубическими сплайнами,

'pchip' – интерполяция кубическими эрмитовыми сплайнами.

Исходные данные и результат линейной интерполяции (задается по умолчанию) представлены ниже.

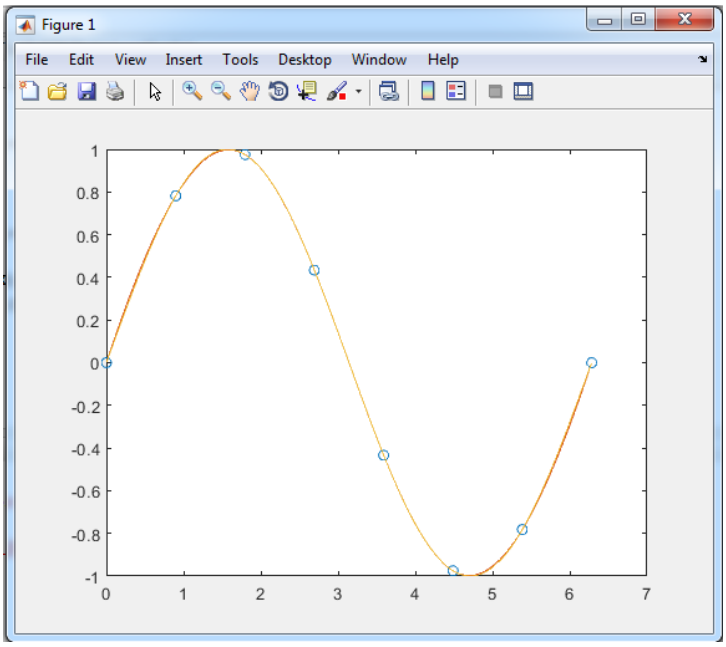




Покажем решение задачи из примера с помощью кубических сплайнов, используя функцию пакета Matlab **spline()**.

```
N=8;
i=1:N;
x(i)=2*pi/(N-1)*(i-1);
y=sin(x);
M=1000;
j=1:M;
X(j)=2*pi/(M-1)*(j-1);
Y=sin(X);
yy=spline(x,y,X);
plot(x,y,'o',X,yy,X,Y);
```

Визуализация исходных данных и результатов кубической сплайн-интерполяции показана на графике.



В Matlab можно также осуществить интерполяцию многомерных данных. Для интерполирования двумерных данных следует задать промежуточные узлы командой **meshgrid** и воспользоваться функцией **interp2**, которая реализует один из способов интерполирования:

‘nearest’ – интерполяция по соседним элементам,

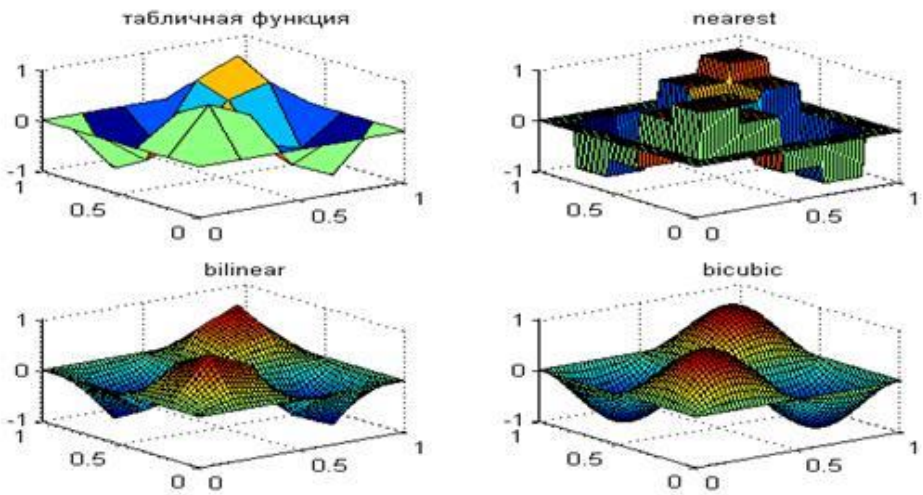
‘bilinear’ или ‘linear’ – билинейная интерполяция,

‘bicubic’ или ‘cubic’ – интерполяция бикубическими сплайнами,

‘spline’ – интерполяция кубическими сплайнами.

Пример использования функций, реализующих двумерную интерполяцию, показан на следующем листинге.

```
>> [X,Y]=meshgrid(0:0.2:1);
>> Z=sin(2*pi*X).*sin(2*pi*Y);
>> subplot(2,2,1)
>> surf(X,Y,Z)
>> [Xi,Yi]=meshgrid(0:0.02:1);
>> Z1=interp2(X,Y,Z,Xi,Yi,'nearest');
>> Z2=interp2(X,Y,Z,Xi,Yi,'bilinear');
>> Z3=interp2(X,Y,Z,Xi,Yi,'bicubic');
>> subplot(2,2,2)
>> surf(Xi,Yi,Z1)
>> subplot(2,2,3)
>> surf(Xi,Yi,Z2)
>> subplot(2,2,4)
>> surf(Xi,Yi,Z3)
```



Задания к лабораторной работе

Вариант 1

Применяя подходящий вид интерполяции, решить следующую прикладную задачу. Найти непрерывный график зависимости изменения температуры в пригороде Лос-Анджелеса по заданным табличным данным и вычислить примерное значение температуры в 14 час. 30 мин.:

t, час.	13-00	14-00	15-00	16-00	17-00	18-00
T, °C	22	20	19	18	17	17

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 2

С помощью подходящего вида интерполяции построить график зависимости значений коэффициентов внутреннего трения (вязкости) для водного раствора сахара (концентрация сахара 20 %) при различных значениях температуры, найти значение коэффициента вязкости при значении температуры 32°С:

T, °С	0	10	20	30	40	50	60	80
η·100, *г*/(*см·с*)	3,804	2,652	1.96	1,504	1.193	0.97	0,808	0,59

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 3

С помощью подходящего вида интерполяции построить график зависимости температуры замерзания морской воды при различных значениях концентрации солей, найти значение температуры при значении концентрации 32 %:

n, %	2	6	10	14	20	24	30	36	40
T, °С	0-0.108	-0.320	-0.534	-0.74860	-1.0745	-1.294	-1.627	-1.9670	-2.196

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 4

С помощью подходящего вида интерполяции построить график зависимости плотности водных растворов серной кислоты при различных значениях температуры (для значения концентрации соли 20%), найти значение плотности при значении температуры 27°С:

T, °С	0	5	10	15	20	25	30	40
ρ, *г/см*^{3}	1,151	1,1481	1,1453	1,1424	1,1394	1,1365	1,1333	1,1275

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 5

С помощью подходящего вида интерполяции построить график зависимости плотности водных растворов серной кислоты при различных значениях концентра-

ции солей (при температуре 20°С), найти значение температуры при значении концентрации 21%:

ρ, *г/см*^{3}	0.9982	1.0317	1.0661	1.1020	1.1394	1.2023	1.2185	1.2515	1.3028
n, %	0	5	10	15	20	28	30	34	40

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 6

С помощью подходящего вида интерполяции построить график зависимости точки кипения воды при различных барометрических давлениях, найти значение температуры при значении давления 726 мм рт. ст.:

H, мм рт.ст	680	697	704	721	739	745	761	777	799
T, °С	96,916	97,596	97,872	98,582	99,218	99,443	100,037	100,62	100,441

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 7

Нормальная термопара Ле-Шателье (Pt-Pt10%Rh) рекомендована международным комитетом как основной прибор при точных измерениях температур. Однако на практике термопара не применяется и служит для контроля показаний рабочих термопар. С помощью подходящего вида интерполяции построить график зависимости значений электродвижущей силы *E* от температуры, найти значение э.д.с. при значении температуры 950°С:

T, °С	0	200	400	600	800	1000	1200	1400	1600
E, мВ	0	1,44	2,26	5,24	7,34	9,61	11,96	14,36	16,73

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 8

С помощью подходящего вида интерполяции построить график зависимости значений коэффициентов внутреннего зрения (вязкости) для водного раствора эти-

лового спирта (концентрация спирта 20%) при различных значениях температуры, найти значение коэффициента вязкости при значении температуры 43°С:

T, °С	0	10	20	30	40	50	60	80
η·100, г/(см·с)	6,319	1,306	1,004	0,801	0,653	0,549	0,47	0,356

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 9

Для анализа покупательской активности в некотором супермаркете зафиксирована численность покупателей в различное время суток. Оценить число покупателей в 15-30, построить непрерывный график зависимости, используя подходящий вид интерполяции.

t	8-00	9-00	10-00	12-00	14-00	16-00	17-30	18-00	20-00
N	33	52	78	126	115	108	136	158	85

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Вариант 10

С помощью подходящего вида интерполяции построить график зависимости точки кипения воды при различных барометрических давлениях, найти значение температуры при значении давления 742 мм рт. ст.:

H, мм рт.ст	680	697	710	721	739	745	750	777	799
T, °С	96.916	97.596	98.108	98.582	99.218	99.443	99.63	100.62	100.441

Привести графики исходных данных, результатов интерполяции и погрешности аппроксимации.

Обработка экспериментальных данных методом наименьших квадратов

Пусть в результате измерений получена таблица некоторой зависимости $f(x)$:

x	x_1	x_2	…	x_n

$F(x)$	y_1	y_2	…	y_n

Требуется найти формулу, выражающую данную зависимость аналитически. Один из подходов состоит в построении интерполяционного многочлена, значения которого в узлах интерполяции x_i совпадают со значениями данной функции $f(x_i)=f$.

Если значения функции $f(x)$ известны с некоторой погрешностью, то требование совпадения значений в узлах интерполяции не оправдано, поскольку оно не означает совпадения характеров исходной и интерполирующей функции. Поэтому поставим задачу следующим образом – найти функцию вида

$y = F(x)$,

которая в точках x_1, x_2, …, x_n принимает значения, близкие к табличным значениям $y_1, y_2, \ldots, y_n$.

Предположим, что приближающая функция $F(x)$ в точках x_1, x_2, …, x_n имеет значения $\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_n$. Тогда нужно найти функцию $F(x)$ определенного вида так, чтобы сумма квадратов

$$(y_1 - \bar{y}_1)^2 + (y_2 - \bar{y}_2)^2 + \cdots + (y_n - \bar{y}_n)^2 \qquad (1)$$

была наименьшей.

Пример 1. Используя табличные значения, найти параметры квадратичной аппроксимирующей функции с помощью встроенной функции Matlab.

x	0	1	2	3	4	5	6	7	8	9	10
y	-0.28	0.42	2.11	4.82	7.75	12.43	12.16	15.41	23.07	31.06	36.68

Для решения задачи обобщенной нелинейной регрессии в пакете Matlab имеется функция **lsqnonlin()**, возвращающая решение задачи нахождения точки минимума функции:

$$\min_x\big(f(x)\big) = f_1^2(x) + f_2^2(x) + \cdots + f_n^2(x) + L,$$

где $f(x)$ – вектор-функция, x – столбец искомых переменных, L – искомая константа.

1. Создадим файл F77.m, содержащий функцию, которая возвращает значение вектор-функции $f(x)$.

```
Editor - C:\Program Files\MATLAB\R2017b\bin\work\F77.m
F77.m   poll.m
1   function z=F77(C,vx,vy)
2 -   k=1:length(vx);
3 -   z(k)=vy(k)-(C(1)*vx.^2+C(2)*vx+C(3));
```

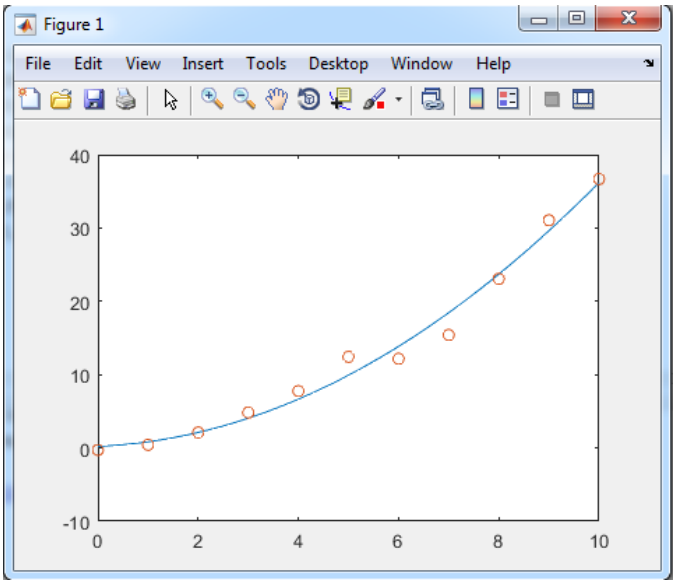
2. Вычислим значения аппроксимирующей функции с помощью функции lsqnonlin() и визуализируем исходные данные и аппроксимирующую функцию.

```
Editor - C:\Program Files\MATLAB\R2017b\bin\work\poll.m
F77.m   poll.m
1 -   N=11;
2 -   i=1:N;
3 -   vx(i)=i-1;
4 -   vy=[-0.28 0.42 2.11 4.82 7.75 12.43 12.16 15.41 23.07 31.06 36.68]
5 -   z=[0 0 0];
6 -   C=lsqnonlin('F77',z,[],[],[],vx,vy)
7 -   F=inline('a*x.^2+b*x+c','a','b','c','x');
8 -   X=vx(1):0.1:vx(length(vx));
9 -   Y=feval(F,C(1),C(2),C(3),X);
10 -  plot(X,Y,vx,vy,'o');
```

Параметры приближающей функции:

```
Command Window
>> poll

vy =

  Columns 1 through 7

   -0.2800    0.4200    2.1100    4.8200    7.7500   12.4300   12.1600

  Columns 8 through 11

   15.4100   23.0700   31.0600   36.6800


Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>


C =

    0.3320    0.2788    0.2264
```

График исходных данных и приближающей функции:



Задание к лабораторной работе

В таблице даны экспериментальные данные для остывания чашки кофе, помещенной в керамический стакан. Температура регистрировалась с точностью 0.1 °С. Температура окружающего воздуха была равна 22.0 °С.

t, мин	0	2	4	6	8	10	12	13	14	15
T, °С	83	75,1	71,1	67,8	64,7	62,1	59,9	58,7	57,8	56,6

Определить параметр остывания *r*, используя аналитическое решение задачи о «чашке кофе» $T = T_C + (T_0 - T_C)e^{-\frac{t}{\tau}}$. Найти аппроксимирующую функ-цию средствами пакета Matlab для исходных данных, представленных в таблице.

Правила выполнения лабораторных работ

Лабораторная работа выполняется строго в соответствии с выданным преподавателем заданием и вариантом. **Лабораторная работа** считается **выполненной** с отметкой «**зачтено**», если: 1) реализация в математическом пакете строго соответствует заданию, 2) студент отвечает на основные теоретические вопросы по соответствующему разделу.

Правила выполнения индивидуальных зачетных работ

Индивидуальной зачетной работой является расширенный вариант лабораторной работы по индивидуальной теме, предложенной преподавателем. Данная работа должна содержать математическую модель физического явления или процесса, теоретическое обоснование, алгоритм решения и программную реализацию в ППП Matlab (собственный программный блок и/или использование встроенных функций пакета), обязательным элементом является графический интерфейс пользователя. Работа должна отвечать требованиям, предъявляемым к оформлению курсовых работ, и содержать разделы 1-4, описанные ниже. Предложенная тематика может быть заменена на другую или модифицирована только по согласованию с преподавателем.

Правила оформления лабораторных и зачетных работ

Оформлять работу следует четко и аккуратно, придерживаясь основных правил оформления отчетных работ:

1. Титульный лист (содержит: наименование вуза, кафедра, ФИО исполнителя, № группы, курс, дисциплина, тема лаб. работы, вариант, выполнил, проверил и т. д.), содержание, введение.

Основная часть.

2. Лист задания (содержит предложенное задание). Раздел, содержащий теоретические основы соответствующего раздела курса (включая подробный алгоритм основного метода).

3. Раздел, содержащий описание программной реализации. Листинг программного блока и описание интерфейса программы (если таковой имеется) можно вынести в приложение. Предлагаемый ППП — Matlab.

4. Заключение, список использованной литературы.

Индивидуальная работа считается выполненной с отметкой «зачтено», если: содержание работы полностью соответствует заданию, студент представил и защитил работу, ответил на дополнительные вопросы, касающиеся реализуемой модели.

Требования к зачету

Курс «Вычислительная физика» выполнен студентом с отметкой «Зачтено», если он не имеет задолженностей по лабораторному практикуму, выполнил и защитил индивидуальную работу по программной реализации модели, а также успешно сдал теоретический и практический минимум зачета.

2 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ

2.1 Общие рекомендации по организации внеаудиторной самостоятельной работы

В высшей школе студент должен прежде всего сформировать потребность в знаниях и научиться учиться, приобрести навыки самостоятельной работы, необходимые для непрерывного самосовершенствования, развития профессиональных и интеллектуальных способностей.

Самостоятельная работа – это процесс активного, целенаправленного приобретения студентом новых для него знаний и умений без непосредственного участия преподавателей.

Для успешной самостоятельной работы студент должен планировать свое время и за основу рекомендуется брать рабочую программу учебной дисциплины.

При организации самостоятельной работы следует взять за правило:

- учиться ежедневно, начиная с первого дня семестра, пропущенные дни будут потеряны безвозвратно;
- чтобы выполнить весь объем самостоятельной работы, необходимо заниматься по 3–5 часов ежедневно;
- начиная работу, надо выбрать что-нибудь среднее по трудности, затем перейти к более трудной работе, и напоследок оставить легкую часть, требующую не столько больших интеллектуальных усилий, сколько определенных моторных действий.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференциальный характер, учитывать специфику специальности, изучаемой дисциплины, индивидуальные особенности студента.

2.2 Работа с учебно-методическим и информационным обеспечением

Важной составляющей самостоятельной внеаудиторной подготовки по всем типам занятий является работа с литературой. Умение работать с литературой озна-

чает научиться осмысленно пользоваться учебно-методическим и другим информационным обеспечение дисциплины.

Для изучения дисциплины вся рекомендуемая литература подразделяется на основную и дополнительную и приводится в п. 10 рабочей программы дисциплины.

К основной литературе относятся источники, необходимые для полного и твердого усвоения учебного материала (учебники и учебные пособия).

Поскольку в учебной литературе (учебниках) зачастую остаются неосвещенными современные проблемы, а также не находят отражение новые документы, события, явления, научные открытия последних лет, то рекомендуется для более углубленного изучения программного материала дополнительная литература.

Прежде чем приступить к чтению, необходимо запомнить или записать выходные данные издания: автор, название, издательство, год издания, название интересующих глав.

Содержание (оглавление) дает представление о системе изложения ключевых положений всей публикации и помогает найти нужные сведения.

Предисловие или введение книги поможет установить, на кого рассчитана данная публикация, какие задачи ставил перед собой автор, содержится краткая информация о содержании глав работы. Иногда полезно после этого посмотреть послесловие или заключение. Это помогает составить представление о степени достоверности или научности данной книги.

Изучение научной учебной и иной литературы требует ведения рабочих записей. Форма записей может быть весьма разнообразной: простой или развернутый план, тезисы, цитаты, конспект. Такие записи удлиняют процесс проработки, изучения книги, но способствуют ее лучшему осмыслению и усвоению, выработке навыков кратко и точно излагать материал. При изучении литературы особое внимание следует обращать на новые термины и понятия. Записи позволяют восстановить в памяти ранее прочитанное без дополнительного обращения к самой книге.

Процесс изучения дисциплины предполагает также активное использование информационных технологий при организации своей познавательной деятельности.

Наличие огромного количества материалов в Сети и специализированных поисковых машин делает Интернет незаменимым средством при поиске информации в процессе обучения.

Однако при использовании интернет-ресурсов следует учитывать следующие рекомендации:

- необходимо критически относиться к информации;
- следует научиться обрабатывать большие объемы информации, представленные в источниках, уметь видеть сильные и слабые стороны, выделять из представленного материала наиболее существенную часть;
- необходимо избегать плагиата, поэтому, если текст источника остается без изменения, необходимо сделать ссылки на автора работы.

2.3 Подготовка к лабораторным и практическим работам

Подготовка к лабораторным работам осуществляется студентом во внеаудиторное время в часы, отведенные на самостоятельную работу. Подготовка требует немалого времени, поэтому целесообразно планировать ее заранее за несколько дней до занятия и целесообразно проводить в следующей последовательности.

1. Внимательно ознакомьтесь с описанием соответствующей лабораторной работы руководствуясь методическими указаниями к выполнению лабораторной работе, уясните цель и задачи эксперимента.

2. Используя конспект лекций и рекомендованную в описании лабораторной работы учебную литературу, изучите теоретические вопросы, относящиеся к лабораторному эксперименту. Выясните теоретические положения, знание которых необходимо для выполнения работы и понимания полученных результатов.

3. Изучите принципиальную схему лабораторной установки, приведенную в описании работы. Ознакомьтесь с применяемым оборудованием, контрольно-измерительными приборами, принципом их действия, правилами эксплуатации.

4. Ознакомьтесь с порядком выполнения работы, усвойте методику измерения физических величин в лабораторном эксперименте, последовательность операций и форму представления полученных результатов.

5. В индивидуальном лабораторном журнале подготовьте «заготовку отчета», который должен быть отражением работы по систематизации приобретенных знаний, опорным планом для проведения эксперимента.

«Заготовку отчета» рекомендуется выполнять по следующей схеме:

- номер, название и цель работы;
- оборудование, приборы и материалы, применяемые в процессе измерений;
- краткий конспект теоретических положений по теме исследования – анализ физических основ метода и описание методики эксперимента, который включает:

— физическое явление, изучаемое в работе, связь между величинами, его описывающими;

— объект исследования, его особенности;

— физическое явление, положенное в основу метода измерений;

— зависимость, которая может быть экспериментально проверена;

— условия, позволяющие осуществить такую проверку;

— и поясняющие теоретический материал рисунки;

- принципиальная схема установки;
- описание метода измерения;
- заготовки таблиц, в которых будут представлены результаты измерений и расчетов (примеры таблиц даются в методическом указании к работе).

6. Проверьте степень подготовленности к лабораторному занятию по контрольным вопросам, приводимым в описании работы.

Окончательное оформление работы, обработка результатов эксперимента и подготовка к защите по контрольным вопросам проводится студентом в часы, отведенные на самостоятельную работу.

К следующему (после выполнения очередной лабораторной работы) занятию каждый студент должен представить окончательный отчет о выполненной лабораторной работе, в который входит «заготовка отчета» дополненная следующими пунктами:

- результаты измерений и вычислений в виде таблиц (или ином виде, согласно методическим рекомендациям к данной лабораторной работе);

- расчетные формулы, по которым производились вычисления с примером вычисления по каждой формуле, что позволяет при необходимости быстро проверить правильность расчета;
- систематизированные результаты эксперимента – схемы, графики, диаграммы и т.п., в соответствии с заданием на лабораторную работу и требованиям, предъявляемым к их оформлению;
- оценка надежности и достоверности результатов (примеры вычислений величин, измеряемых косвенно, и погрешностей для прямых и косвенных измерений);
- основные выводы по результатам работы, вытекающие из экспериментальных данных или на основании сравнения полученных результатов с теоретическими данными, если это возможно с объяснением расхождения.

2.4 Самостоятельное изучение и конспектирование отдельных тем

Для подготовки конспекта рекомендуется использовать основную и дополнительную литературу.

При написании конспекта придерживайтесь следующих рекомендаций.

1. Прежде чем приступить к чтению, необходимо записать выходные данные издания: автор, название, издательство, год издания.

2. Внимательно прочитайте текст.

3. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта.

4. Выделите главное, составьте план.

5. Кратко сформулируйте основные положения текста, отметьте аргументацию автора.

6. Законспектируйте материал, четко следуя пунктам плана.

При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно. Грамотно записывайте цитаты, учитывайте лаконичность, значимость мысли. В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения.

2.5 Подготовка к текущему и промежуточному контролю

Подготовка к промежуточной аттестации. Формой промежуточной аттестации (контроля) является зачет. Зачет может проводиться в виде письменного опроса с последующим собеседованием или с применением тестирования.

Зачет – форма проверки полученных теоретических и практических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умения синтезировать полученные знания.

Основная цель подготовки к зачету – достичь понимания законов и явлений, а не только механически заучить материал.

Подготовка к устной сдаче зачета включает в себя несколько основных этапов:

– просмотр программы учебного курса;

– определение необходимых для подготовки источников (учебников, дополнительной литературы и т.д.) и их изучение;

– использование конспектов лекций;

– консультирование у преподавателя.

Для успешной сдачи зачета рекомендуется соблюдать несколько правил.

1. Подготовка к зачету начинается с первого занятия по дисциплине, на котором студенты получают общую установку преподавателя и перечень основных требований к текущей и промежуточной аттестации. При этом важно с самого начала планомерно осваивать материал, руководствуясь, прежде всего перечнем вопросов к экзамену, конспектировать важные для решения учебных задач источники.

2. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до зачета. В течение этого времени нужно успеть повторить и систематизировать изученный материал.

3. За несколько дней перед зачетом распределите вопросы равномерно на все дни подготовки, возможно, выделив последний день на краткий повтор всего курса.

4. Каждый вопрос следует проработать по конспекту лекций, по учебнику или учебному пособию.

5. После повтора каждого вопроса нужно, закрыв конспект и учебники, самостоятельно воспроизвести иллюстративный материал с последующей самопроверкой.

6. Все трудные и не полностью понятые вопросы следует выписывать на отдельный лист бумаги, с последующим уточнением ответов на них у преподавателя на консультации.

7. При ответе на вопросы билета студент должен продемонстрировать знание теоретического материала и умение применить при анализе качественных и количественных задач. Изложение материала должно быть четким, кратким и аргументированным.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Попов И.С. Вычислительные методы в задачах теоретической физики [Электронный ресурс] : учебно-методическое пособие / И.С. Попов, М.А. Медведева. — Электрон. текстовые данные. — Омск: Омский государственный университет им. Ф.М. Достоевского, 2015. — 52 с. — 978-5-7779-1903-8. — Режим доступа: http://www.iprbookshop.ru/59590.html

2. Васильев А.Н. Matlab [Электронный ресурс] : самоучитель. Практический подход / А.Н. Васильев. — Электрон. текстовые данные. — СПб. : Наука и Техника, 2015. — 448 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/43318.html

3. Кудинов Ю.И. Практическая работа в MATLAB [Электронный ресурс] : учебное пособие / Ю.И. Кудинов. — Электрон. текстовые данные. — Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2013. — 62 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/55606.html

4. Введение в математический пакет Matlab [Электронный ресурс] : учебно-методическое пособие / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 88 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61469.html

5. Дьяконов В.П. MATLAB. Полный самоучитель [Электронный ресурс] / В.П. Дьяконов. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 768 с. — 978-5-4488-0065-8. — Режим доступа: http://www.iprbookshop.ru/63590.html

6. Зализняк В.Е. Основы вычислительной физики. Часть 2. Введение в методы частиц [Электронный ресурс] / В.Е. Зализняк. — Электрон. текстовые данные. — Москва, Ижевск: Регулярная и хаотическая динамика, Ижевский институт компьютерных исследований, 2006. — 156 с. — 5-93972-481-7. — Режим доступа: http://www.iprbookshop.ru/16584.html

7. Зализняк В.Е. Основы вычислительной физики. Часть 1. Введение в конечно-разностные методы [Электронный ресурс] / В.Е. Зализняк. — Электрон. текстовые данные. — Москва-Ижевск: Регулярная и хаотическая динамика, Ижевский ин-

ститут компьютерных исследований, 2004. — 252 с. — 5-93972-326-8. — Режим доступа: http://www.iprbookshop.ru/17647.html.

8. Поршнев, С.В. Компьютерное моделирование физических процессов в пакете MATLAB. + CD [Электронный ресурс] : учеб. пособие / С. В. Поршнев. - СПб.; М. ; Краснодар : Лань, 2011. - 736 с.

9. Гулд, Х. Компьютерное моделирование в физике : в 2-х ч.: Учеб. пособие. Ч. 1 / Х. Гулд, Я. Тобочник; Пер. с англ. А.Н. Полюдова, В.А. Панченко. - М. : Мир, 1998. - 349 с.

10. Власова, Е. А. Приближенные методы математической физики: учеб. : доп. Мин. обр. РФ / Е. А. Власова, В. С. Зарубин, Г. Н. Кувыркин. - М. : Изд-во Моск. гос. техн. ун-та им. Н.Э. Баумана, 2001. - 700 с.

11. Ращиков В.И., Рошаль А.С. Численные методы решения физических задач: Учеб. Пособие. – СПб.: Изд-во «Лань», 2005. – 208 с.

СОДЕРЖАНИЕ

Введение 3
1 Методические рекомендации к практическим и лабораторным работам 3

1.1 Подготовка к выполнению лабораторной работы 3

1.2 Выполнение лабораторной работы в лаборатории 3

1.3 Оформление отчета и подготовка к «защите» лабораторной работы 5

1.4 Предлагаемые к выполнению лабораторные и практические работы и методические указания к ним 5

2 Методические рекомендации к самостоятельной работе 91

2.1 Общие рекомендации по организации внеаудиторной самостоятельной работы 91

2.2 Работа с учебно-методическим и информационным обеспечением 91

2.3 Подготовка к лабораторным и практическим работам 92

2.4 Самостоятельное изучение и конспектирование отдельных тем 95

2.5 Подготовка к текущему и промежуточному контролю 95

Библиографический список 98

Нещименко Виталий Владимирович,

доцент кафедры Физики АмГУ, доктор физ.-мат. наук

Юрина Виктория Юрьевна

ассистент кафедры Физики АмГУ