

Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

Е.А.ДЕЙ

**ПРОГРАММИРОВАНИЕ
И РЕШЕНИЕ ФИЗИЧЕСКИХ ЗАДАЧ
С ПРИМЕНЕНИЕМ ЧИСЛЕННЫХ МЕТОДОВ**

ПРАКТИЧЕСКОЕ ПОСОБИЕ
для студентов 2 курса
специальности 1-31 04 01 02 – «Физика»

Гомель 2005

УДК 681.3.06(075)
ББК 32.973я7
Д27

Рецензенты:

Г.Ю.Тюменков, доцент, кандидат физико-математических наук
кафедра теоретической физики учреждения образования
«Гомельский государственный университет
имени Франциска Скорины»

Рекомендовано к изданию научно-методическим советом
учреждения образования «Гомельский государственный университет
имени Франциска Скорины» 26.10.2005 (№ 191)

Д27 Программирование и решение физических задач с
применением численных методов: Практическое
пособие для студентов 2 курса специальности 1-31 04
01 02 – «Физика» / Е.А.Дей: Мин-во обр. РБ. – Гомель:
УО «ГГУ им. Ф. Скорины», 2005. – 84 с.

Практическое пособие ставит своей целью усвоение студентами
основных численных методов и формирование навыков их
применения к решению физических задач и адресовано студентам 2
курса специальности «Физика» для использования на лабораторных
занятиях по курсу «Программирование и математическое
моделирование».

© Е.А.Дей, 2005
© УО «ГГУ им. Ф. Скорины», 2005

СОДЕРЖАНИЕ

Введение	4
Тема 1. Элементы оформления диалоговых программ	5
Тема 2. Создание и использование собственного модуля при разработке прикладных программ	12
Тема 3. Изучение численных методов решения нелинейных уравнений	17
Тема 4. Использование функций и процедур в качестве параметров	23
Тема 5. Применение численных методов решения нелинейных уравнений в физических задачах	30
Тема 6. Изучение и программирование численных методов интегрирования	37
Тема 7. Вычисление определенных интегралов с заданной точностью	46
Тема 8. Изучение и программирование численных методов решения задачи Коши для ОДУ-1	48
Тема 9. Исследование физических систем на основе численного решения ОДУ первого порядка	53
Тема 10. Численное решение тестовой задачи Коши для ОДУ второго порядка	61
Тема 11. Численные методы решения систем линейных алгебраических уравнений	66
Тема 12. Обработка результатов физического эксперимента по методу наименьших квадратов	73
Литература	84

ВВЕДЕНИЕ

Умение разрабатывать собственные программы для решения физических задач с использованием современных средств программирования является необходимым элементом подготовки студента-физика.

Язык программирования Турбо-Паскаль имеет все необходимые для выполнения научных и технических расчетов элементы: удобная система типов данных, структурные операторы управления, логичная структура программы и программных модулей, лаконичные средства работы с экраном и графикой.

Изучение этого языка формирует необходимую методическую основу для последующего перехода к работе в среде визуального программирования Delphi.

Изложению численных методов посвящено немало книг и учебников. Однако большинство этих книг ориентировано на студентов и научных работников математического профиля. Поэтому в настоящее время ощущается потребность в разработке учебного курса по программированию и численным методам, рассчитанного на студентов-физиков и сочетающего достаточную полноту изложения с разумной степенью математической строгости.

Изучение данного курса, кроме того, имеет большое значение для последующего эффективного выполнения студентами курсовых и дипломных работ с применением компьютерных расчетов при решении физических задач.

Практическое пособие имеет целью сформировать у студентов знание математических основ и свойств базовых численных методов и выработать навыки их применения для исследования физических систем. В пособии рассмотрены численные методы интерполяции, численного интегрирования, решения линейных и нелинейных уравнений и систем уравнений, дифференциальных и интегральных уравнений.

По каждой теме приводятся теоретические сведения, описания алгоритмов, примеры программ, а также вопросы для самоконтроля и варианты заданий для самостоятельного выполнения.

Практическое пособие адресовано студентам специальности 1-31 04 01 «Физика» в рамках изучения третьей части курса «Программирование и математическое моделирование».

ТЕМА 1. ЭЛЕМЕНТЫ ОФОРМЛЕНИЯ ДИАЛОГОВЫХ ПРОГРАММ

Цель работы: Изучение элементов оформления прикладных программ, приобретение навыков разработки диалоговых программ для решения физических задач.

1. Общие требования к разработке прикладных программ.

Решение конкретной прикладной задачи подразумевает выбор правильной структуры данных, исследование свойств численного метода, логичную и ясную программную реализацию, подробное представление результатов.

При разработке программ, кроме того, важное значение имеет оформление программы. Следует учитывать, что:

а) программа пишется для других людей, то есть, с программой должно быть удобно работать в плане оформления, последовательности действий при работе с программой, наличия пояснений по ходу работы программы, документации, справочной информации;

б) программа должна быть в определенной степени универсальной, способной решать не одну конкретную задачу, а целый класс задач, различающихся возможными значениями параметров.

2. Элементы оформления диалога.

Диалог проектируется заранее. Важно выделить основные кадры диалога и их назначение: заставка, ввод данных, информация о ходе вычислений, вывод результатов и т.д. В качестве примера рассмотрим вариант организации диалога в программе, реализующей численное решение дифференциального уравнения.

Кадр 1. Заставка (название программы, ее назначение, используемый метод, характеристики, данные разработчика, дата и версия).

Программа предназначена для численного решения дифференциальных уравнений 1-го порядка методом Эйлера
--

Кадр 2. Описание программы (краткая сводка правил работы)

Описание программы

Кадр 3. Ввод исходных данных.

Ввод данных.
Задайте численное значение..

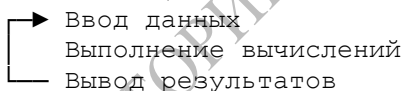
Кадр 4. Индикатор выполнения

Выполняется расчет 34%
■■■■■■■■□□□□□□□□□□□□

Кадр 5. Вывод результатов вычислений

Результаты вычислений:
Координата = 20.56 м
Скорость = 7.83 м/с

Важное свойство прикладной программы – готовность к выполнению новых вычислений, без завершения работы после получения результата. Это достигается организацией глобального цикла, охватывающего все этапы вычислений.



Наиболее распространенный способ организации работы программы – с помощью меню. Это связано с тем, что меню – наиболее быстрый способ выбора режима работы.

Под меню понимается список всех доступных в данный момент возможностей программы. Выбор нужного элемента меню означает команду программе выполнить соответствующие действия.

Типы меню:

- 1) цифровое (символьное) меню.
- 2) Вертикальное (горизонтальное) меню (клавиши ↑↓).

New
Save
Save as . .
Exit

3. Набор процедур для оформления диалога в текстовом режиме. В данном разделе приведен вариант текста процедур, позволяющих оформить диалог пользователя с прикладной программой в текстовом режиме.

Процедура Colors предназначена для установки цвета символов cv1 (8..15) и фона cv2 (1..7). Ее удобно использовать, если на экран выводится несколько элементов с разным цветовым оформлением. При вызове процедуры в качестве параметров следует указать нужные номера цвета символов и цвета фона, например:

```
Colors(14,1); {желтые символы на синем фоне}
```

Для просмотра возможных сочетаний цвета символов и фона в текстовом режиме используйте вспомогательную программу, расположенную в меню <Tools> среды Турбо-Паскаль (дополнительно установлена на компьютере в дисплейном классе).

Процедура Shadow закрашивает прямоугольную область экрана, определяемую координатами верхней левой (x1,y1) и нижней правой (x2,y2) точек заданным цветом фона cv2



и устанавливает цвет символов cv1 для последующего вывода информации в эту область.

Процедура Single ограничивает прямоугольную область экрана (x1,y1)-(x2,y2) рамкой, составленной из символов псевдографики.

Пример программы, содержащей перечисленные процедуры:

```
PROGRAM DIALOG;  
{оформление диалога в текстовом режиме}  
Uses Crt;  
Var a,f : real;  
  
Procedure Colors(cv1,cv2:byte);  
{-----установка цвета символов cv1 и цвета фона cv2}  
begin  
  TextColor(cv1); TextBackGround(cv2);  
end;
```

```

Procedure Shadow(x1,y1,x2,y2,cv1,cv2:byte);
{-----заполнение цветом прямоугольного участка
экрана}
var i:byte;
begin
  Colors(cv1,cv2);
  For i:=y1 to y2 do begin
    GotoXY(x1,i); Write(' ':x2-x1+1);
  end;
end;

Procedure Single(x1,y1,x2,y2,cv1,cv2:byte);
{-----ограничение прямоугольного участка рамкой}
const s1= #218; s2= #196; s3= #191;
      s4= #179; s5= #192; s6= #217;
var i:byte;
begin
  Shadow(x1+1,y1+1,x2+1,y2+1,0,0);      {*-если нужно}
  Shadow(x1,y1,x2,y2,cv1,cv2);
  GotoXY(x1,y1); Write(s1);
  For i:=x1+1 to x2-1 do write(s2);
  write(s3);
  For i:=y1+1 to y2-1 do begin
    GotoXY(x1,i); Write(s4);
    GotoXY(x2,i); Write(s4);
  end;
  GotoXY(x1,y2); Write(s5);
  For i:=x1+1 to x2-1 do write(s2);
  write(s6);
end;

BEGIN                                {-исполняемая часть программы}
  Colors(11,1); ClrScr;                {-фон всего экрана}

  Single(20,6,60,16,10,3);             {-оформление ввода данных}
  GotoXY(29,6); Write('Вычисление синуса числа');
  GotoXY(30,9); Write('Ввод исходных данных:');
  GotoXY(32,12);Write('Аргумент = '); Readln(a);
  Shadow(20,6,61,17,0,1);              {фон для следующего этапа}
  f:=sin(a);                            {-все необходимые вычисления}
  ClrScr;

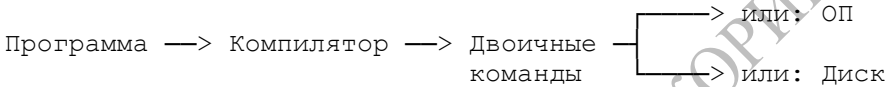
  Single(20,6,60,16,14,5);             {-вывод результатов}
  GotoXY(29,6); Write('Вычисление синуса числа');
  GotoXY(30,9); Write('Результаты вычислений:');

```



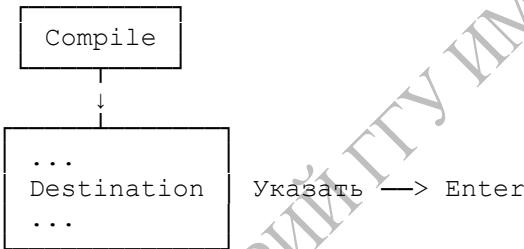
```
GotoXY(32,11);Write('Аргумент = ',a:8:5);
GotoXY(32,13);Write('Синус = ',f:8:5);
Readln;                                     {-завершение работы}
END.
```

4. Создание EXE-файла программы. EXE-файл - это файл, содержащий двоичные команды управления процессором. Команды генерируются компилятором в процессе обработки текста программы на языке Турбо-Паскаль. Файл имеет расширение .EXE (Execute - выполнить).

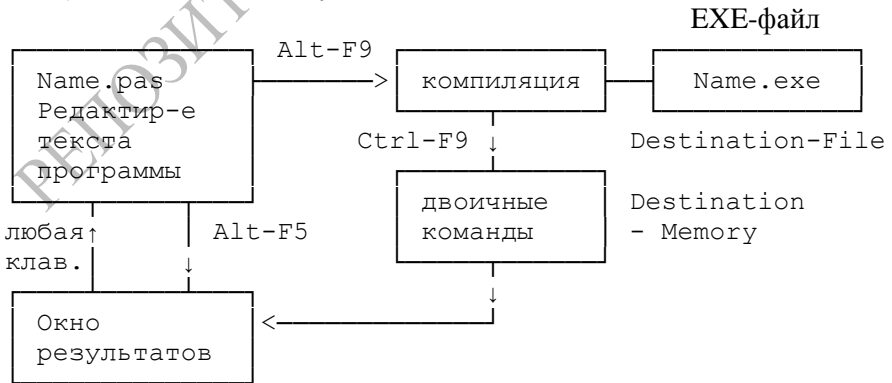


Для создания EXE-файла на диске необходимо:

1) изменить размещение (Destination) результата компиляции в меню Compile:



2) выполнить команду компиляции Alt-F9.



СОДЕРЖАНИЕ ПИСЬМЕННОГО ОТЧЕТА

Название работы, номер варианта, данные исполнителя, результаты вычислений по каждой задаче.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА 1. Выполнить последовательно указанные действия:

1) Набрать и выполнить приведенную программу Dialog, использующую элементы оформления диалога в текстовом режиме. Изменяя параметры ВЫЗОВА процедуры Single (координаты и цвет), посмотреть их влияние на вид экрана. Подобрать наиболее удобные значения параметров.

2) На основе процедуры Single разработать еще одну процедуру с именем Double, рисующую рамку двойными линиями псевдографики (использовать символы псевдографики с кодами s1= #201; s2= #205; s3= #187; s4= #186; s5= #200; s6= #188). Оформить вывод результатов в программе Dialog с помощью процедуры Double.

3) С помощью созданной программы проверить свойство: $\sin(x) \approx x$ при малых значениях x . Для этого вводить в качестве аргумента значения 0.010, 0.012 .. 0.040. Начиная с какого значения синус отличается от аргумента в тысячных долях?

ЗАДАЧА 2. Для решения задачи варианта разработать собственную программу с использованием средств оформления диалога.

Вычисление результата реализовать с помощью подпрограммы-процедуры или подпрограммы-функции. В самой программе выполняется ввод исходных данных, вызов подпрограммы и вывод результатов на экран.

Создать EXE-файл программы. С помощью созданной программы выполнить несколько вычислений для заданных условий. Результаты записать в отчет.

Варианты задания:

1) Определить время падения камня на поверхность земли с высоты H и скорость в момент падения. Сопротивлением воздуха пренебречь. Рассмотреть ситуации: а) $H=300$ м; б) $H=500$ м.

2) Два сопротивления R_1 , R_2 могут быть соединены параллельно или последовательно. Найти общее сопротивление соединения в

обоих случаях. Рассмотреть ситуации: а) $R_1=15$, $R_2=12$; б) $R_1=45$, $R_2=12$;

3) Железная деталь массой m нагревается от температуры t_1 до температуры t_2 (ниже температуры плавления). Определить количество энергии, затраченной на ее нагревание. Удельная теплоемкость железа $c=0,46$ кДж/(кг·°C). Рассмотреть ситуации: а) $m=15,6$ кг, $t_1=40^\circ\text{C}$, $t_2=340^\circ\text{C}$; б) $m=2,5$ кг, $t_1=0^\circ\text{C}$, $t_2=610^\circ\text{C}$;

4) В цепь включены последовательно три конденсатора с емкостью C_1 , C_2 , C_3 . Вычислить общую электрическую емкость цепи. Рассмотреть ситуации: а) $C_1=270$ пф, $C_2=300$ пф, $C_3=560$ пф; б) $C_1=670$ мкф, $C_2=130$ мкф, $C_3=56$ мкф;

5) Реализовать в программе перевод углов из градусной меры в радианную. Произвести вычисления для значений 15° ; 30° ; 45° .

6) Реализовать в программе вычисление вещественной степени положительного числа. Вычислить: 1.2^5 ; 3.5^4 ; 2^{10} .

7) Реализовать в программе вычисление площади треугольника по формуле Герона: $p=(a+b+c)/2$; $S=\text{SQRT}(p(p-a)(p-b)(p-c))$. Вычислить площадь для сторон а) $a=12.4$; $b=23.8$; $c=19.1$; б) $a=2.3$; $b=3.5$; $c=5.1$;

8) Реализовать в программе перевод углов из радианной меры в градусную. Произвести вычисления для значений 0.5 ; 1.0 ; 1.5 .

9) Реализовать в программе вычисление скалярного произведения векторов в 3-мерном пространстве. Вычислить значения всех скалярных произведений для векторов $w=(1.2, 2.5, 12.1)$; $s=(5.8, 11.3, 7.5)$; $q=(6.4, 9.2, 0.8)$.

10) Вычисления модуля вектора в 3-мерном пространстве. Вычислить модуль векторов $w=(1.2, 2.5, 12.1)$; $s=(5.8, 11.3, 7.5)$; $q=(6.4, 9.2, 0.8)$.

11) Вычисление площади круга по заданному радиусу. Произвести вычисления для значений $R=1$; 2 ; 3 ; 4 ; 5 .

12) Вычисление площади трапеции по высоте и основаниям. Произвести вычисления для значений а) $a=1.2$; $b=1$; $h=0.5$; б) $a=2.0$; $b=1.8$; $h=1.5$.

ТЕМА 2. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ СОБСТВЕННОГО МОДУЛЯ ПРИ РАЗРАБОТКЕ ПРИКЛАДНЫХ ПРОГРАММ

Цель работы: изучение структуры модулей в языке Турбо-Паскаль и приобретение навыков создания собственного модуля программиста.

1. Понятие модуля в языке Турбо-Паскаль. В языке программирования Turbo Pascal возможен доступ к большому числу созданных при разработке языка (встроенных) констант, типов данных, переменных, процедур и функций. Количество таких элементов велико, но все сразу они в программах не используются. Поэтому такие элементы собраны в отдельных файлах, называемых модулями.

Модуль - это набор заранее запрограммированных констант, типов данных, переменных, процедур и функций, которые можно вызвать и использовать внутри любой программы.

Язык Турбо-Паскаль предоставляет шесть стандартных модулей: System, Crt, Graph, Dos, Overlay, Printer.

Модуль System содержит все стандартные и встроенные процедуры и функции Turbo Pascal. Любая подпрограмма Turbo Pascal, не являющаяся стандартной и не находящаяся ни в одном другом модуле, находится в System. Этот модуль *автоматически* подключается к каждой программе.

Модуль Crt содержит константы, переменные и программы для работы с экраном в текстовом режиме: управление курсором, управление цветом, ввод информации с клавиатуры, управление звуковым сигналом.

Модуль Graph содержит набор быстродействующих подпрограмм, которые позволяют в полной мере использовать графические возможности компьютера. Для его использования следует в среде Турбо-Паскаль указать его полный путь (в меню Options/Disk/Unit/Directories) или скопировать GRAPH.TPU в текущий каталог.

Модуль Dos определяет многочисленные процедуры и функции Turbo Pascal, обеспечивающие работу с файлами и элементами MS DOS, а также с системными прерываниями.

В учебниках по языку Турбо-Паскаль всегда содержится описание элементов стандартных модулей.

2. Структура модуля. Как и программа, модуль в языке Турбо-Паскаль имеет определенную структуру. Заголовок модуля - слово `Unit`, за которым следует имя модуля.

Следующий элемент - ключевое слово `Interface`. Оно обозначает часть модуля, доступную для всех вызывающих его программ. В интерфейсе модуля можно объявить константы, типы данных, переменные, процедуры и функции. В предложении `Uses` указываются другие модули, которые будет использовать данный модуль. Заголовки процедур и функций, доступных для вызывающих программ, описываются в разделе интерфейса. А их полный текст - в разделе реализации.

Раздел реализации - закрытая, недоступная для внешних программ часть - начинается со слова `Implementation`. Все, что объявлено в части интерфейса, можно использовать в разделе реализации. Кроме того, в разделе реализации могут быть свои собственные дополнительные объявления.

```
Unit <имя модуля>;
Interface           {-открытая часть модуля}
  Uses <список модулей>;
  {-описание доступных извне элементов: Const, Type, Var}
  {-полные заголовки доступных извне функций и процедур}
Implementation
  Uses <список модулей>;
  {-описание внутренних элементов Const, Type, Var, ...}
  {-полные тексты функций и процедур}
Begin
  {команды инициализации}
End.
```

Операторы модуля в области `Begin...End` составляют раздел инициализации (он может и отсутствовать). Если программа использует модуль, то раздел инициализации модуля вызывается перед выполнением самой программы. Если в программе используется несколько модулей, раздел инициализации каждого модуля вызывается (в порядке, указанном в операторе `Uses` программы) до выполнения операторов главной программы.

Заголовок Procedure или Function в разделе реализации должен быть такой же, как и в разделе интерфейса, или же иметь короткую форму (без указания параметров). Например:

```
Unit Library;
Interface
  Const g=9.81;
  Type Mas10=array [1..10] of byte;
  Procedure SumMas10(a:mas10; var s:integer);
  Function SinH(x:real):real;

Implementation      {-----}

  Procedure SumMas10(a:mas10; var s:integer);
  {суммирование элементов массива типа Mas10}
  var i:integer;
  begin
    s:=0;
    For i:=1 to 10 do s:=s+a[i];
  end; {процедуры SumMas10}

  Function SinH;
  {гиперболический синус}
  begin
    SinH:=(exp(x)-exp(-x))/2;
  end; {функции SinH}

End.{модуля Library}
```

Подпрограммы раздела реализации (не включенные в секцию интерфейса), должны иметь полный заголовок.

3. Использование модулей. Для использования модулей в программе необходимо, чтобы в начале присутствовало предложение Uses, за которым следует список имен всех модулей, разделенных запятыми, например:

```
Program MyProg;
Uses Modul1, Modul2, Modul3;
```

Если предложение Uses отсутствует, Turbo Pascal подсоединяет только стандартный модуль System.

Если имя модуля указано в описании Uses, то все константы, типы данных, переменные, процедуры и функции, объявленные в

интерфейсе этого модуля становятся доступными для Вашей программы.

Например, после включения предложения `Uses Library` в программу появилась возможность ссылаться на все объявления и описания в секции интерфейса этого модуля (`g`, `gz`, `mz`, `mas10` и т.д.). Поэтому в программе, использующей этот модуль, перечисленные элементы уже не нужно описывать.

```
Program TestModul;  
Uses Library;  
Var i,sw: integer;  
    W:mas10;  
    x,sh:real;  
Begin  
  Writeln('Ускорение свободного падения g=',g:0:2,  
    ' м/с^2');  
  Writeln('--Заполните массив из 10 целых чисел: ');  
  For i:=1 to 10 do begin  
    Write(i:2,'-й элемент: '); Readln(w[i]);  
  End;  
  SumMas10(W,sw);  
  Writeln('Сумма элементов массива: ',sw);  
  Writeln('--Вычисление гиперболического синуса');  
  Writeln('Задайте значение аргумента');  
  Readln(x);  
  Sh:=SinH(x);  
  Writeln('SH(X)= ',sh:0:4);  
  Readln;  
End. {программы TestModul}
```

Примечание 1. Чтобы найти модуль, указанный в операторе `Uses`, компилятор вначале просматривает текущий каталог, а затем – каталог, заданный в меню `Options/Disk/Unit/Directories`.

Примечание 2. Если в программе, использующей модуль, некоторые имена совпадают с именами объектов из модуля, то при выполнении программа будет использовать собственные описания, так как они были описаны позже, чем в модуле. Если в такой программе нужно, наряду с собственными объектами, использовать объекты с такими же именами, определенные в модуле, то при использовании в программе перед каждым совпадающим именем

помещается имя модуля с точкой (аналогично указанию поля записи).
Например:

```
P1:=g; P2:=Library.g;
```

4. Создание собственных модулей. Текст модуля набирается в редакторе среды Турбо-Паскаль и сохраняется на диске так же, как и текст программы, и тоже с расширением *.pas. Имя файла, в котором записан текст модуля, должно совпадать с именем модуля. Перед использованием каждый модуль должен быть откомпилирован (так же, как компилируется сама программа).

Таким образом, для создания собственного модуля необходимо выполнить следующие действия:

1) написать подробный текст модуля с соблюдением требований к его структуре;

2) набрать текст модуля при помощи редактора среды Турбо-Паскаль и сохранить на диске под заданным именем (расширение .pas ставится автоматически);

3) для выполнения компиляции в меню Compile указать адрес размещения результата компиляции: пункт Destination – указать и нажать <Enter>. В результате будет установлено значение Destination–Disk);

4) выполнить команду компиляции (меню Compile - пункт Compile), или нажать клавиши Alt-F9. При этом автоматически создается файл с расширением .TPU (Turbo Pascal Unit) и именем модуля.

5) создать текстовый файл, содержащий описание элементов, включенных в модуль.

Перспективная цель – составить библиотеку подпрограмм, реализующих основные численные методы, и использовать ее в дальнейшем для решения физических задач.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА 1. Набрать текст модуля Library, приведенного в п.2. Сохранить его в файле Library.PAS и выполнить компиляцию с сохранением на диске. Результирующий код помещается в файл LIBRARY.TPU.

Набрать текст программы (п.3), использующей модуль Library и выполнить ее (модуль должен находиться в текущем каталоге). С помощью программы вычислить сумму чисел: $120+34+56+30+100+28+75+92+99+255$. Вычислить гиперболический синус для значений аргумента $x=0, 2, 4, 6, 8, 10$.

ЗАДАЧА 2. Создать собственный модуль, содержащий процедуры оформления диалога, разработанные в ходе выполнения предыдущей лабораторной работы. Скомпилировать модуль. Создать текстовый файл, содержащий описание элементов собственного модуля.

Переработать собственную программу из Л.р.-1, использующую эти процедуры, так, чтобы в ней не было текста процедур, а использовался вызов модуля. Скомпилировать программу и выполнить ее посредством запуска EXE-файла.

ТЕМА 3. ИЗУЧЕНИЕ ЧИСЛЕННЫХ МЕТОДОВ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Цель работы: изучение и применение численных методов поиска и уточнения корней нелинейных уравнений и их программной реализации при решении тестовых задач

1. Классификация нелинейных уравнений. Решение нелинейных уравнений является одной из важнейших задач, часто возникающих во многих разделах физики, математики, техники. В общем случае нелинейное уравнение можно записать в виде

$$F(x) = 0, \quad (1)$$

где функция $F(x)$ определена и непрерывна на интервале $[a, b]$. Задача заключается в нахождении действительных корней уравнения (нулей функции F).

Поскольку подавляющее большинство нелинейных уравнений с одной переменной не решается аналитически, на практике их решают численными методами. Численное нахождение корней обычно состоит из трех этапов:

1) отделение корней, т.е., нахождение достаточно малых участков в рассматриваемой области, внутри которых заведомо содержится только один корень;

2) уточнение значения корня до заданной точности;

3) проверка найденного корня подстановкой его в уравнение.

2. Отделение корней. Процесс отделения корней основан на теореме Больцано-Коши: если непрерывная функция $F(x)$ принимает на концах отрезка $[A, B]$ значения разных знаков, т.е. $F(A) \cdot F(B) < 0$, то внутри этого отрезка содержится по крайней мере один корень (график функции пересекает ось OX , и функция меняет знак). Этот корень будет единственным, если производная $F'(x)$ сохраняет знак на $[A, B]$.

Отделение корней во многих случаях можно провести графически. Принимая во внимание, что действительные корни уравнения (1) - это точки пересечения графика функции $F(x)$ с осью абсцисс, достаточно построить график $F(x)$ и выбрать на оси OX отрезки, содержащие по одному корню. Границы отрезка $[x_a; x_b]$ для каждого корня выбираются достаточно произвольно, "на глаз", они служат для ориентации при выполнении действий численного метода и должны лишь указывать, что внутри $[x_a; x_b]$ имеется корень, и притом единственный (и функция монотонна).

Второй способ отделения корней реализуется с помощью пошагового поиска. Для этого вычисляются значения $F(x)$ начиная с точки $x=A$ с некоторым шагом h . Как только обнаружится, что значения функции в соседних точках $F(x)$, $F(x+h)$ имеют разные знаки, соответствующие значения аргумента x можно считать границами отрезка, содержащего корень. Далее поиск продолжается до достижения границы B .

Результатом выполнения пошагового поиска будут выводимые на дисплей или в файл в цикле значения границ отрезков для каждого встреченного на участке $[a; b]$ корня. Очевидно, что шаг h следует выбирать достаточно мелким, так как при большом значении h функция может дважды пересечь ось OX и на границах отрезка иметь одинаковые знаки. Этот алгоритм можно записать следующим образом (псевдокод):

```
| Ввод A, B, h
| xa=A   xb=xl+h   ya=F(xa)
```

```

| Пока  $x_b < B$ 
|    $y_b = F(x_b)$ 
|   Если  $y_a * y_b < 0$ , то Вывод  $x_a, x_b$ 
|    $x_a = x_b$     $x_b = x_a + h$     $y_a = y_b$ 

```

3. Метод половинного деления. Метод половинного деления является самым простым и надежным для нахождения корней нелинейного уравнения.

Пусть уравнение (1) имеет на отрезке $[a, b]$ единственный корень, причем функция $F(x)$ непрерывна. Разделим отрезок $[a, b]$ пополам точкой $c = (a+b)/2$. Если $F(c) < 0$ (что практически наиболее вероятно), то возможны два случая: $F(x)$ меняет знак либо на участке $[a; c]$, либо на участке $[c; b]$. Значит, границы отрезка можно уточнить и изменить значения a или b . Выбирая в каждом случае тот из отрезков, на котором функция меняет знак, и продолжая процесс половинного деления дальше, можно прийти до сколь угодно малого отрезка, содержащего корень уравнения, ширина начального отрезка после N итераций убывает в 2^N раз, так что с увеличением числа итераций метод обеспечивает получение все более точного приближения значения корня.

```

| Ввод  $a, b, delta$ 
| Повторять
|    $c = (a+b) / 2$ 
|   Если  $F(a) * F(c) < 0$ , то
|     |  $b = c$ 
|   иначе
|     |  $a = c$ 
|   Вывод  $c, F(c)$ 
| до выполнения  $abs(b-a) < delta$ 
|  $x = (a+b) / 2$ 
|  $fx = F(x)$ 
| Вывод  $x, fx$ 

```

4. Метод Ньютона (метод касательных). Метод последовательных приближений, разработанный Ньютоном, очень широко используется при построении итерационных алгоритмов. Метод основан на том соображении, что вблизи корня точка пересечения касательной к функции с осью OX расположена ближе к корню, чем точка, в которой касательная построена. Выберем на отрезке

$[x_a, x_b]$, содержащем корень, точку x_0 . На основании разложения функции $F(x)$ в ряд Тейлора в точке $x_1 = x_0 + h$ получаем:

$$F(x_0+h) = F(x_0) + h * F'(x_0) + h^2 * F''(x_0) / 2 + \dots$$

Члены, содержащие h во второй и более высоких степенях, отбрасываются; используется соотношение $h = x_1 - x_0$. Предполагается, что переход от x_0 к x_1 приближает значение функции к нулю, так что $F(x_0+h) = 0$. Тогда

$$x_1 = x_0 - F(x_0) / F'(x_0). \quad (2)$$

Значение x_1 соответствует точке, в которой касательная к кривой в точке x_0 пересекает ось OX . Так как кривая $F(x)$ отлична от прямой, то значение функции $F(x)$ скорее всего не будет в точности равно нулю. Поэтому вся процедура повторяется, причем в качестве x_0 на следующем шаге используется x_1 . Счет прекращается по достижении достаточно малого значения $Abs(x_1 - x_0) < \delta$.

Конечно, в данном методе необходимо знать явный вид производной $F'(x)$. Вычисление функции $F(x)$ и производной $F'(x)$ оформляется посредством отдельных подпрограмм-функций.

Быстрота сходимости в большой мере зависит от удачного выбора исходной точки. Если в процессе итераций тангенс угла наклона касательной $F(x)$ обращается в нуль или очень велик, то применение метода осложняется.

Для выбора исходной точки в методе Ньютона используют следующее правило:

- если функция возрастает и выгнута вверх, или функция убывает и выгнута вниз - то в качестве x_0 выбирается левая граница отрезка $[a, b]$, то есть, $x_0 = a$;

- если функция убывает и выгнута вверх, или функция возрастает и выгнута вниз - то в качестве x_0 выбирается правая граница отрезка $[a, b]$, то есть, $x_0 = b$.

```

Ввод x0, delta
Повторять
  x1 = x0 - F(x0) / F'(x0)
  Вывод x1, F(x1)
  d = x1 - x0      x0 = x1
до выполнения abs(d) < delta
Вывод x1, F(x1)
  
```

3. Запись результатов в текстовый файл. Для записи результатов в текстовый файл в программе следует добавить такие элементы:

```
Program Prim1;
Var   tf:text;           {1-описание файловой переменной}
Begin
. . .
Assign(tf,'Prim1.txt'); {2-значение файловой переменной
                        - имя реального файла}
Rewrite(tf);           {3-Установка режима обработки:
                        создание нового файла}
. . .
Writeln('m= ',massa:0:4); {имеющийся вывод
                          результата на экран}
Writeln(tf, 'm= ',massa:0:4); {4-вывод результата в
                              файл}
. . . .
Close(tf);             {5-закрытие файла}
End.
```

Можно также использовать и один из режимов (действует только один после открытия файла)

```
Reset(tf);           {режим чтения существующего файла}
Append(tf);          {запись в конец существующего файла}
```

СОДЕРЖАНИЕ ОТЧЕТА

1. Условие задачи (конкретный вариант).
2. Математическая формулировка используемого численного метода.
3. Файл программы и результаты вычислений.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА 1. Составить программу, выводящую на экран таблицу значений функции $f(x)$ в заданной области $[a;b]$ с шагом h .

Функцию $f(x)$ запрограммировать как подпрограмму-функцию.

Ввод a , b , h реализовать с использованием процедур оформления диалога.

Проверить работу программы на примере функции, соответствующей уравнению $x^2-3=0$ в области $[-3;3]$ с шагом $h=0,25$. По таблице значений выявить интервалы, содержащие корни уравнения. Результаты записать в отчет.

ЗАДАЧА 2. Дополнить программу операторами, реализующими метод автоматического поиска корней заданной функции на указанном интервале с требуемым шагом h . Все найденные участки, содержащие один корень, вывести на экран.

Проверить работу программы на примере уравнения $x^2-3=0$ в области $[-5;5]$ с шагом $h=0,1$ и $h=0,01$. Результаты записать в отчет.

ЗАДАЧА 3. Выделить корни для заданного уравнения в указанной области с шагом $h=0,1$, изменив подпрограмму-функцию, образующую левую часть нелинейного уравнения. Результаты записать в отчет.

1. $x^5 - 6.7x^4 + 8.9x^3 + 3.6x^2 + 0.3x = 0$; [-1;5]
2. $x^2 \sin(x) - 0.5x \cdot \cos(x) + 4x - 2 = 0$; [0;6]
3. $0.5 \ln(x^2+120) - \sin x \cdot \ln(x^4+1) - \exp(x/3) = 0$; [1;6]
4. $\sin(x-0.5) \ln(x) - \cos(2 \ln(x)) = 0$; [3;12]
5. $e^{\sin x} - (x^2+1) \sin(2x) - x \ln(x+4) = 0$; [1;6]
6. $\cos(2x) - \exp(2-x) + x \cdot \sin 3x = 0$; [0;5]
7. $0,8x^2 - 4x \sin(x/2) - 7 \ln(1+x^2) + 1 = 0$; [0;7]
8. $\cos(x/2) - \sin(x/3) + 0.4x + 4.2 \sin(x) = 0$; [-2;8]
9. $x - x^2 \cos^2(x+1) - 32,4 = 0$; [4;8]
10. $7,3x - \sin(x-1) + 2 \cos(x) - x^2(x+1) = 0$; [-3;3]
11. $e^{-x} \cos(3x+1) - x^3 + x \operatorname{ch}(x) - 1 = 0$; [1;8]
12. $\sin(x+1) \cdot \cos(x-1) - \ln(x) - 2e^{-x} = 0$; [1;4]
13. $x^3 \sin(x/3) - x^2 \cos(x/2) + xe^{-x} - \ln(\pi) = 0$; [-5;5]
14. $x^3 \sin(x) + x^2 \cos(x) - 2 \operatorname{sh}(x) - \ln(x^2+1) = 0$; [-3;3]
15. $\sin(x+1) \cdot \cos(x-1) - \ln(x) - 2e^{-x} = 0$; [1;4]
16. $(x-2)(x-0.5) - 3x \cdot \operatorname{sh}(x-1) + \operatorname{arctg}(x) = 0$; [-2;3]
17. $x^3 \sin(x/3) - x^2 \cos(x/2) + xe^{-x} - \ln(\pi) = 0$; [-5;5]

ЗАДАЧА 4. Составить программу, выполняющую решение нелинейного уравнения указанным методом с точностью ε .

Предусмотреть вывод всех промежуточных и окончательного результатов.

Использовать указанный численный метод:

Варианты: 1, 5, 9, 13, 17, 21 – метод половинного деления;

Варианты: 2, 6, 10, 14, 18, 22 – метод хорд;

Варианты: 3, 7, 11, 15, 19, 23 – метод Ньютона;

Варианты: 4, 8, 12, 16, 20, 24 – метод итераций;

Проверить работу программы на примере уравнения $x^2-3=0$ с точностью $\varepsilon = 10^{-7}$. Сравнить с табличными данными. Результаты записать в отчет.

ЗАДАЧА 5. Вычислить все корни уравнения из Задачи 1 с точностью $\varepsilon = 10^{-6}$. Результаты записать в отчет.

ТЕМА 4. ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ И ПРОЦЕДУР В КАЧЕСТВЕ ПАРАМЕТРОВ

Цель работы: изучение принципов использования процедур и функций в качестве параметров при программировании численных методов.

1. Библиотечные подпрограммы и численные методы.

Подпрограммы, выполняющие решение часто повторяющихся задач, удобно хранить в собственном модуле. Такие подпрограммы называют библиотечными, а набор подпрограмм (модуль) – библиотекой подпрограмм. Значение библиотечных подпрограмм в том, что их можно использовать затем в новых программах, не тратя время на повторное программирование.

При этом важно таким образом оформить библиотечную подпрограмму, чтобы при использовании ее для решения новой задачи не нужно было переделывать сам текст подпрограммы. Настроить подпрограмму на решение новой задачи можно только с помощью нужных значений ее параметров.

Большинство численных методов направлено на обработку заданных функций и получение значений тех функций, которые неизвестны. Это означает, что при решении новой задачи изменяется вид функции, а в программе – имя функции. Возникает необходимость передачи имени функции (или процедуры) в обрабатывающую подпрограмму в качестве параметра.

Например, программа, реализующая вывод на экран таблицы значений функции, может иметь достаточно простой вид

```
PROGRAM TablFun1; {Вывод на экран таблицы значений
функции}
Uses Crt;
Var N,i :integer; a,b,h :real; x,y:real;
Function Fun1(x:real):real;
begin
Fun1:=sin(2*x)*exp(x);
end;
BEGIN
ClrScr;
a:=1; b:=4; h:=0.2;
N:=Round((b-a)/h);
Writeln('Таблица значений функции');
Writeln(' i x y');
Writeln('-----');
For i:=0 to N do begin
x:=a+i*h; y:=Fun1(x);
Writeln(i:2,' ',x:8:4,' ',y:8:4);
end;
Writeln('-----');
Readln;
END.
```

Для того, чтобы выводить таблицы значений функции и в других программах, оформим необходимые действия в виде подпрограммы-процедуры. В качестве параметров процедуры укажем те переменные, которые можно задавать по-разному для одной и той же функции (пределы a,b и шаг h). В этом случае программа будет иметь следующий вид:

```
PROGRAM TablFun2; {Вывод таблицы функции с помощью
подпрограммы}
Uses Crt;
```



```

Var  a,b,h :real;

      Function Fun1(x:real):real;
      begin
          Fun1:=sin(2*x)*exp(x);
      end;

Procedure Tabl(ta,tb,th:real);
Var  N,i:integer;  x,y:real;
begin
    N:=Round((tb-ta)/th);
    Writeln('Таблица значений функции');
    Writeln(' i           x           y');
    Writeln('-----');
    For i:=0 to N do begin
        x:=a+i*th;  y:=Fun1(x);
        Writeln(i:2,' ',x:8:4,' ',y:8:4);
    end;
    Writeln('-----');
    Readln;
end;

BEGIN
    ClrScr;
    a:=1;  b:=4;  h:=0.2;
    Tabl(a,b,h);
    Tabl(0,1,0.1);
    Tabl(5,9,0.4);
END.

```

Отметим, что параметры процедуры имеют свои имена, которые не обязаны совпадать с именами переменных в программе. Но переменные программы, используемые для указания значений параметров при вызове, должны иметь тот же тип, что и параметры подпрограммы.

Процедура `Tabl(ta,tb,th)` позволяет получать на экране таблицы для любых приемлемых интервалов изменения аргумента с любым шагом. Однако, ее нельзя считать библиотечной процедурой и включать в собственный модуль, потому что она обрабатывает только одну фиксированную функцию. Для обработки другой функции придется изменять строку `y:=Fun1(x)`, что недопустимо для текста подпрограммы, находящейся в модуле.

2. Функциональные типы. Язык программирования Турбо-Паскаль позволяет интерпретировать процедуры и функции как элементы, значение которых можно присваивать переменным и передавать в качестве параметров другим подпрограммам. Это особенно важно при программировании алгоритмов обработки функций и численных методов.

При описании параметров необходимо указывать их тип. Так как функциональные типы не являются стандартными, их необходимо описать в программе в разделе Type.

Описание функционального типа во многом совпадает с записью заголовка процедуры или функции, используемой как параметр, но при этом:

- имя процедуры или функции не ставится;
- перечисляются типы параметров;
- имена параметров могут быть произвольными;

Приведем некоторые примеры описаний функционального типа:

```
Type
  RealFunctionType = function(a:real):real;
  Proc2Param = procedure(X,Y:integer);
  NumericalProc = procedure(N:byte; var x,y:mas);
  WorkFun = function(A,B:real;F:MathFunc);
```

Имена параметров в описании процедурного типа играют чисто формальную роль и могут быть выбраны любыми.

После определения процедурного типа появляется возможность описывать переменные и параметры этого типа. Такие переменные называют процедурными переменными.

3. Оформление процедур, передаваемых в качестве параметров. После объявления процедурного типа его можно использовать в описаниях параметров подпрограмм. При этом в программе должны быть запрограммированы и те реальные процедуры и функции, которые будут передаваться как параметры. Их оформление имеет некоторые особенности.

а) Процедуры или функции, которые будут использованы в качестве параметров, должны компилироваться в специальном режиме. Этот режим включается размещением перед текстом подпрограмм директивы компилятора {\$F+}, а после текста подпрограмм выключается директивой {\$F-}:

```
{ $F+ }
```

```
Function SE(tt:real):real;
begin
  se:=Sin(tt)*Exp(tt);
end;
{. . другие подпрограммы, используемые в качестве
параметров}
{$F-}
```

б) нельзя использовать в качестве параметров встроенные функции. При необходимости использования следует составить свою подпрограмму-функцию, переопределяющую встроенную

```
Function MySin(x:real):real;
Begin
  Mysin:=sin(x);
End;
```

С учетом сказанного, дополним текст подпрограммы Tabl параметром функционального типа. В тексте программы используем три различные функции и для каждой построим таблицу значений. Конкретная реализация каждой таблицы определяется указанием параметров при вызове подпрограммы Tabl

```
PROGRAM TablFun3;
{-Вывод на экран таблицы значений функции:
  - с помощью подпрограммы с функциональным параметром
  - использование подпрограммы для нескольких функций}
Uses Crt;
Type RealFun1= Function(x:real):real;
Var a,b,h :real;

{$F+}
Function Alfa(x:real):real;
begin
  Alfa:=sin(2*x+1)*Cos(x/2);
end;
  Function NewFun(x:real):real;
  begin
    NewFun:=Exp(-x)*(x*x*x-3*x*x+5*x-1);
  end;
Function Fun1(x:real):real;
begin
```

```

    Fun1:=sin(2*x)*exp(x);
end;
{$F-}

Procedure Tabl(a,b,h:real; F:RealFun1);
Var N,i:integer; x,y:real;
begin
    N:=Round((b-a)/h);
    Writeln('Таблица значений функции');
    Writeln(' i           x           y');
    Writeln('-----');
    For i:=0 to N do begin
        x:=a+i*h; y:=F(x);
        Writeln(i:2,' ',x:8:4,' ',y:8:4);
    end;
    Writeln('-----');
    Readln;
end;

BEGIN
    ClrScr;
    a:=1; b:=4; h:=0.2;
    Tabl(a,b,h,Fun1);
    Tabl(a,b,h,Alfa);
    Tabl(a,b,h,NewFun);
END.

```

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА 1. Набрать и выполнить текст программы TablFun1. Убедиться в ее работоспособности.

Выполнить необходимые изменения и дополнения текста программы в соответствии с примером TablFun2. Убедиться в ее работоспособности.

Изменить программу TablFun2 таким образом, чтобы параметры таблицы вводились с клавиатуры с соответствующими пояснениями. Получить на экране таблицу функции для значений a, b, h, выбранных самостоятельно.

ЗАДАЧА 2. Создать новый программный файл, содержащий текст программы TablFun3 (используя предыдущие программы в качестве основы). Убедиться в работоспособности программы.

Включить процедуру Tab1 в собственный модуль. Изменить текст программы так, чтобы использовалась процедура Tab1, находящаяся в модуле.

ЗАДАЧА 3. На основе материала предыдущей лабораторной работы составить подпрограмму-процедуру, реализующую автоматический поиск корней нелинейного уравнения, причем параметры a, b, h и имя функции должны быть параметрами процедуры поиска корней. Включить текст процедуры в программу и убедиться в ее работоспособности.

ЗАДАЧА 4. Включить процедуру автоматического поиска корней нелинейного уравнения в собственный модуль. Изменить текст программы так, чтобы использовалась процедура, находящаяся в модуле. Использовать процедуру для поиска корней трех различных нелинейных уравнений (явный вид уравнений получить у преподавателя). Интервалы, содержащие корни, записать в отчет. Учесть, что подпрограммы-функции для всех трех уравнений должны присутствовать в программе. Интервалы, содержащие корни, записать в отчет.

ЗАДАЧА 5. На основе материала предыдущей лабораторной работы составить подпрограмму-функцию, реализующую вычисление корня нелинейного уравнения с заданной точностью. Границы интервала, содержащего корень, требуемая точность результата и имя функции, определяющей левую часть нелинейного уравнения, являются параметрами функции, вычисляющей корень.

ЗАДАЧА 6. Включить функцию, реализующую вычисление корня нелинейного уравнения с заданной точностью, в собственный модуль. Изменить текст программы так, чтобы использовалась функция, находящаяся в модуле. Использовать процедуру для вычисления корней трех различных нелинейных уравнений (явный вид уравнений получить у преподавателя). Результаты записать в отчет.

ТЕМА 5. ПРИМЕНЕНИЕ ЧИСЛЕННЫХ МЕТОДОВ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ В ФИЗИЧЕСКИХ ЗАДАЧАХ

Цель работы: 1) применение процедур, реализующих численное решение нелинейных уравнений, для решения физических задач; 2) совершенствование навыков подготовки физических задач для решения на ЭВМ; 3) совершенствование навыков использования подпрограмм.

1. Компьютерные модели и вычислительный эксперимент. Моделью называется искусственный объект, изучаемые свойства которого аналогичны свойствам природного объекта. Математическая модель – это система математических соотношений (уравнений, неравенств, начальных и граничных условий), сформулированных на основе физических законов. В некоторых случаях эта система соотношений может быть решена аналитически, и ответ может быть получен в замкнутой форме (в виде формулы), но в большинстве случаев для получения решения требуется применение численных методов.

Численными называются методы получения результата путем выполнения арифметических действий над числами.

Компьютерной моделью называется программа, реализующая численные методы решения системы математических соотношений (математической модели), с использованием элементов диалога для ввода данных и установки параметров, а также с использованием графических средств отображения результатов, предназначенная для исследования физических процессов и систем. Компьютерная модель используется для проведения большого количества расчетов, для получения подробной информации о физической системе. В этом отношении использование компьютерной модели соответствует проведению реального физического эксперимента. Поэтому научные направления, связанные с разработкой компьютерных моделей, называется компьютерным моделированием, компьютерным экспериментом, или компьютерной физикой.

2. Этапы разработки компьютерной модели. При разработке компьютерной модели можно выделить следующие этапы (первые – без компьютера):

1) постановка задачи (словесная формулировка). Задание диапазона изменения физических параметров, выделение наиболее существенных физических величин;

2) разработка математической модели - формулировка системы математических соотношений;

3) качественное исследование математической модели. Получение приближенных аналитических решений в частных или предельных случаях, оценка вклада различных величин, формулировка тестовых примеров;

4) выбор и формулировка численного метода. Оценка точности численного метода, оценка его устойчивости и сходимости;

5) составление и отладка программы, реализующей компьютерную модель. Выявление синтаксических и логических ошибок, тестирование программы (решение задач с известным ответом);

6) проведение расчетов в ходе выполнения компьютерного эксперимента. Расчеты проводятся многократно при регулярном изменении параметров;

7) анализ результатов, возможная модификация модели (возврат к началу);

8) техническая реализация. Оптимизация.

9) написание отчета.

3. Пример качественного исследования математической модели. Рассмотрим задачу. Два одинаковых металлических шарика массой m подвешены в одной точке на невесомых нерастяжимых непроводящих нитях длины L . Каждому из них сообщён электрический заряд q . Найти зависимость угла отклонения α от вертикали от величины заряда $\alpha(q)$.

Для получения расчетных соотношений запишем условие равновесия одного из шариков (в силу симметрии их отклонения от вертикальной оси будут одинаковы) в проекциях на оси координат.

$$OX : \quad T \sin \alpha = \frac{q^2}{4\pi\epsilon_0\epsilon(2L \sin \alpha)^2};$$

$$OY : \quad T \cos \alpha = mg;$$

Разделив первое равенство на второе, получаем

$$\frac{\sin \alpha}{\cos \alpha} = \frac{q^2}{(4\pi\epsilon_0\epsilon 4L^2 \sin^2 \alpha) mg},$$

откуда следует нелинейное уравнение для нахождения угла α при заданных параметрах q, l, m, ϵ

$$\sin^3 \alpha = \frac{q^2}{4\pi\epsilon_0\epsilon 4L^2 mg} \cos \alpha; \quad \alpha(q) \text{ -- ?}$$

Полученное нелинейное уравнение в общем случае можно решить только численными методами. Однако, с помощью качественного исследования можно получить определенные сведения о решении. Введем обозначение

$$b = \frac{q^2}{4\pi\epsilon_0\epsilon(2L)^2 mg} = \frac{F_{кул, max}}{F_{тяж}}$$

Важно отметить, что величина b является безразмерной величиной, это просто число (выражается через отношение двух сил). Уравнение принимает вид

$$\sin^3 \alpha = b \cos \alpha.$$

Исследуем теперь, как зависит величина угла α от параметра b , то есть, рассмотрим функцию $\alpha(b)$.

1) Рассмотрим случай очень малых зарядов, $b \ll 1$, когда угол отклонения α - малая величина порядка ϵ , $\alpha \sim \epsilon$. В этом случае $\sin \alpha \sim \alpha$, $\cos \alpha \sim 1$, и уравнение (2) принимает вид:

$$\alpha^3 = b$$

Значит, для малых значений α мы можем получить аналитическое решение уравнения

$$\alpha \approx \sqrt[3]{b}$$

2) Рассмотрим теперь другой предельный случай – очень больших зарядов $b \gg 1$ и, соответственно, очень больших отклонений:

$\alpha \sim \frac{\pi}{2}$. Так как α не является малой величиной, перейдем к малому параметру

$$\alpha \sim \frac{\pi}{2} - \beta; \quad \beta \sim \varepsilon;$$

При этом тригонометрические функции, входящие в уравнение, можно приближенно заменить

$$\sin \alpha = \sin\left(\frac{\pi}{2} - \beta\right) = \cos \beta \approx 1; \quad \cos \alpha = \cos\left(\frac{\pi}{2} - \beta\right) = \sin \beta \approx \beta$$

Так что уравнение (1) запишется в виде

$$1 = b \cdot \beta.$$

И в этом случае легко получаем аналитическое решение уравнения:

$$\beta \sim \frac{1}{b}, \text{ то есть, } \alpha \approx \frac{\pi}{2} - \frac{1}{b}.$$

Таким образом, переход к безразмерным величинам – один из методов качественного решения физических задач.

Уже на приведенном примере видно, что введение безразмерного параметра дает следующие преимущества:

- вид уравнения существенно упростился;
- количество независимых параметров уменьшилось с 5 до одного;
- облегчается переход к численному решению, так как в уравнении $\sin^3 \alpha = b \cos \alpha$ все величины – числа.

СОДЕРЖАНИЕ ЗАДАНИЯ

Сформулировать в соответствии с условием варианта нелинейное уравнение $f(x)=0$ и преобразовать его к безразмерным переменным.

Вывод уравнения изложить в отчете.

Используя программу из предыдущей Л.р. в качестве основы, составить новую программу, в которой выполнить:

- построить таблицу значений функции на заданном интервале с достаточно мелким шагом;
- выбрать по таблице значений начальное приближение корня

- вычислить корень с точностью 10^{-6} и значение функции в найденной точке.

Записать в отчет все промежуточные значения корня, окончательный результат и значение функции в найденной точке.

Варианты: 1, 5, 9, 13 – метод половинного деления;

Варианты: 2, 6, 10, 14 – метод хорд;

Варианты: 3, 7, 11, 15 – метод Ньютона;

Варианты: 4, 8, 12, 16 – метод итераций;

1) Плотность тока насыщения при термоэлектронной эмиссии из горячего катода в электровакуумном приборе меняется по закону:

$$j(T) = bT^2 \exp\left(-\frac{A}{kT}\right),$$

где энергия активации $A=4,23$ эВ; коэффициент $b=5,3 \cdot 10^5$ А/(м²К²); $k=1,38 \cdot 10^{-23}$ Дж/К – постоянная Больцмана. Найти температуру, при которой плотность тока равна 120 мА/мм².

2) Два положительных электрических заряда величины $Q_1=4,2 \cdot 10^{-7}$ Кл и $Q_2=8,3 \cdot 10^{-7}$ Кл расположены на оси ОУ в точках (0,L) и (0,-L), где $L=0,25$ м. Определить, на каком расстоянии R на оси ОХ величина модуля напряженности поля, создаваемого двумя зарядами, будет равна $E=1000$ В/м.

3) Баскетболист бросает мяч с начальной скоростью $V_0=10,2$ м/с. Под каким углом он должен совершить бросок, чтобы попасть в кольцо с расстояния $L=8$ м по навесной траектории. Высота закрепления кольца $H=2,8$ м, высота баскетболиста $h=1,95$ м. Ответ выразить в радианах и в градусах. Указание: описать движение в проекциях на координатные оси $x(t)$, $y(t)$ и, исключив время t , найти уравнение траектории движения $y(x)$.

4) Последовательно соединены резистор из металлической проволоки и полупроводниковый резистор. Температурные зависимости их сопротивлений соответственно определяются формулами

$$R_1(T) = R_a \alpha T; R_2(T) = R_b \cdot \exp\left(\frac{W}{kT}\right),$$
 где $R_a=125$ Ом; $R_b=5,86 \cdot 10^{-2}$

Ом; $\alpha=4,18 \cdot 10^{-3}$ 1/К; $W=4,8 \cdot 10^{-20}$ Дж; $k=1,38 \cdot 10^{-23}$ Дж/К. Определить, при какой температуре общее сопротивление будет минимальным. Указание: положение минимума можно определить из равенства нулю

первой производной, причем она в окрестности этой точки должна возрастать (вторая производная >0).

5) Для резисторов, описанных в предыдущей задаче, найти температуру, при которой их параллельное соединение будет иметь максимальное сопротивление. Указание: положение максимума для $r(T)$ можно определить из равенства нулю первой производной, причем она в окрестности этой точки должна убывать.

6) Электрон влетает в однородное электрическое поле плоского конденсатора с напряженностью $E=880$ В/м, имея начальную скорость $V_0=7.2 \cdot 10^5$ м/с, направленную под углом φ к пластинам, длина которых $L=10$ см. Начальная высота электрона над нижней пластиной, заряженной положительно, равна $h_0=0.6$ см, а при вылете из поля - равна $h=0.4$ см. Вычислить угол φ . Ответ выразить в радианах и в градусах. Указание: описать движение электрона в проекциях на координатные оси $x(t)$, $y(t)$ и, исключив время t , найти уравнение траектории движения $y(x)$.

7) Снаряд вылетает из пушки под углом $\alpha=38^\circ$ к горизонту с начальной скоростью 345 м/с. Через какой промежуток времени его расстояние (не проекция!) от точки выстрела будет равно 2,5 км. Сопротивление воздуха не учитывать.

8) Электрон, имея скорость 10^4 м/с, влетел в однородное магнитное поле с индукцией $B=30$ мТл под углом α к направлению линий индукции. Чему равен угол α , если через время $t=0.0001$ сек расстояние электрона от начальной точки составило 1,2 мм.

9) Наклонная плоскость, образующая угол α с плоскостью горизонта, имеет длину $L=2$ м. Тело, двигаясь равноускоренно соскользнуло с этой плоскости за время $t=2$ с. Найти угол α , если коэффициент трения $k_{тр}=0,28$.

10) Снаряд, выпущенный из зенитного орудия вертикально вверх со скоростью $V_0=850$ м/с, поднялся до высшей точки траектории за время $t=38,76$ с. Считая силу сопротивления воздуха пропорциональной скорости, определить массу снаряда. Указание: сформулировать и решить аналитически уравнение движения снаряда.

11) Моторная лодка массой $m=420$ кг, начинает двигаться по озеру и через время $t=18,5$ с приобретает скорость $V=7,2$ м/с. Сила тяги F мотора равна 0,2 кН. Считая силу сопротивления $F_{сопр}$ движению пропорциональной скорости, определить коэффициент сопротивления. Указание: сформулировать и решить аналитически уравнение движения лодки.

12) Большой из двух соударяющихся абсолютно упругих шаров с отношением масс $k=M/m=2,5$ покоится. Какую часть μ своей кинетической энергии потерял меньший шар в результате прямого удара.

13) При наличии в колебательном контуре активного сопротивления r свободные колебания контура являются затухающими, так что ток в контуре изменяется следующим образом:

$$I(t) = I_0 \exp\left(-\frac{r}{2L}t\right) \sin(\omega t), \text{ где } I_0 - \text{ начальная амплитуда, } L -$$

индуктивность контура, ω - частота собственных колебаний. Найти, через какое время после начала свободных колебаний ток в контуре будет иметь значение $0.45 I_0$ при $C=20 \text{ МкФ}$, $L=0,1 \text{ Гн}$, $r=1,2 \text{ Ом}$.

14) На какую глубину H погружен центр плавающего в воде деревянного (плотность $\rho=0.65 \text{ г/см}^3$) шара, имеющего радиус $R=12 \text{ см}$. Объем шарового сегмента высоты h равен $V=(\pi/3)h^2(3R-h)$.

15) Угол α отклонения луча в треугольной призме при симметричном расположении входящего и выходящего лучей зависит от преломляющего угла θ в вершине призмы и показателя преломления материала призмы n : $n \cdot \sin(\theta/2) = \sin((\theta + \alpha)/2)$. Какой угол θ должна иметь призма, чтобы при $n=1.9$ угол отклонения луча составил $\alpha = 27,5$ градусов.

16) Металлическую заготовку, имеющую форму полушара радиуса R , необходимо разрезать на две части равной массы. На каком расстоянии от основания должна проходить плоскость разреза. Объем шарового сегмента высоты h равен $V=(\pi/3)h^2(3R-h)$. Составить таблицу результатов при изменении R от 1 до 10 с шагом 1.

17) В архитектуре и изобразительном искусстве широко используется принцип "золотого сечения", определяющий пропорцию при делении отрезков. При этом сечение отрезка AC на две части выполняется так, чтобы большая его часть AB относилась к меньшей части BC так, как весь отрезок AC относится к AB . Вычислить значение этого отношения. Определить, с какой точностью приближают его дроби $5/3$; $8/5$; $13/8$.

18) Два шарика массой $m=0,12 \text{ г}$ каждый подвешены в одной точке на нитях длиной $L=18 \text{ см}$ каждая. Получив одинаковый электрический заряд $Q=46.3 \text{ нКл}$, шарики разошлись так, что нити образовали между собой угол α . Найти величину этого угла. Выразить ответ в радианах и градусах.

ТЕМА 6. ИЗУЧЕНИЕ И ПРОГРАММИРОВАНИЕ ЧИСЛЕННЫХ МЕТОДОВ ИНТЕГРИРОВАНИЯ

Цель работы: изучение и программирование численных методов интегрирования и получения результата с заданной точностью на основе теоретических оценок погрешности.

1. Квадратурные формулы. Рассмотрим практическую задачу. Какой путь пролетел этот объект по прямолинейной траектории, если результаты измерения его скорости следующие:

t_i	v_i
0	1.2
0.1	1.8
0.2	2.6
0.3	2.2
0.4	1.7

$$S = \int_{t_1}^{t_2} v(t) dt$$

Таким образом, задача свелась к вычислению определенного интеграла.

Пусть требуется вычислить определенный интеграл $\int_a^b y(x) dx$, где

$y(x)$ - некоторая заданная на отрезке $[a,b]$ функция. Известно, что аналитическими методами вычисляется лишь довольно узкий класс интегралов, поэтому в большинстве практически важных задач возникает необходимость приближенного вычисления определенного интеграла с помощью ЭВМ, с заранее заданной точностью ε .

Интеграл для неотрицательных функций $y(x)$ геометрически представляет собой площадь фигуры, сверху ограниченной графиком кривой $y(x)$.

Во многих случаях выражение для первообразной не существует, или же подинтегральная функция может быть задана не в виде формулы, а в табличном виде.

В этих случаях подинтегральную функцию заменяют на некоторую интерполирующую функцию, интеграл от которой легко вычисляется в явном виде.

Приближенное вычисление интеграла основано на том, что на малых участках подинтегральная функция заменяется на элементарную функцию с известным поведением, соответствующим

поведению интегрируемой функции, и считается площадь, ограниченная сверху графиком этой новой функции. Если отклонения от исходной функции невелики, то невелика и погрешность такого расчета. Формулы приближенного интегрирования различаются между собой выбором приближений.

$$y(x) = P^L(x) + r(x); \quad P^L(x) = \sum_i y_i l_i(x)$$

$$\int_a^b y(x) dx = \int_a^b P^L(x) dx + \int_a^b r(x) dx$$

$$\int_a^b y(x) dx = \sum_i y_i \underbrace{\int_a^b l_i(x) dx}_{A_i} + \underbrace{\int_a^b r(x) dx}_R$$

получаем общий вид квадратурной формулы

$$\int_a^b y(x) dx = \sum_i A_i y(x_i) + R$$

где A_i – весовые коэффициенты, x_i – узлы квадратурной формулы, R – погрешность квадратурной формулы.

Квадратурные формулы, полученные на основе интерполяционных полиномов Лагранжа, называются формулами Ньютона-Котеса.

Численное интегрирование применяется, если:

- 1) подынтегральная функция задана таблицей значений;
- 2) определенный интеграл не вычисляется в элементарных функциях;
- 3) подынтегральная функция является громоздкой, сложной для аналитического интегрирования.

2. Методы прямоугольников. В методе прямоугольников подынтегральную функцию приближенно описывают с помощью полинома нулевого порядка – константы.

$$P_0(x) = a = \text{const. Система узлов } h = \frac{b-a}{N}.$$

Метод левых прямоугольников

$$\int_a^b y(x) dx \approx I_{np} = h \sum_{i=0}^{N-1} y(x_i) + R_{III}$$

Метод правых прямоугольников

$$\int_a^b y(x) dx \approx I_{np} = h \sum_{i=1}^N y(x_i) + R_{III}$$

Метод центральных прямоугольников

$$\int_a^b f(x) dx \approx I_{np} = h \sum_{i=1}^N y\left(x_i + \frac{h}{2}\right) + R_{III} = h \sum_{i=1}^N y_{i+1/2} + R_{III}$$

Погрешность методов правых и левых прямоугольников $\sim h$, поэтому эти методы практически не используются в научных и технических расчетах.

Теоретическая оценка погрешности для формулы центральный прямоугольников имеет вид

$$\left| R_{III,N} \right| = \left| I_{np} - \int_a^b y(x) dx \right| \leq \frac{(b-a)}{24} h^2 M_2,$$

где M_2 – максимальное значение модуля второй производной подинтегральной функции в области интегрирования

$$M_2 = \max_{[a,b]} |y''(x)|$$

Таким образом, верхний предел погрешности

$$\left| R_{III,max} \right| = \frac{(b-a)}{24} h^2 M_2.$$

3. Квадратурная формула трапеций. Разобьем интервал $[a,b]$ на N (в общем случае неравных) частей точками $x_0, x_1, x_2, \dots, x_i, \dots, x_N$, причем $x_0=a, x_N=b$. Обозначим значения подинтегральной функции $y=f(x)$ в этих точках через $y_0, y_1, y_2, \dots, y_i, \dots, y_N$, то есть, $y_i=f(x_i)$. Отметим, что длина интервала разбиения $h=(b-a)/N$ тем меньше, чем больше N .

Рассмотрим малый участок $[x_{i-1}, x_i]$, $i=1, \dots, N$. Применим к подинтегральной функции линейную интерполяцию, то есть, заменим график функции $y=f(x)$ на отрезке $[x_{i-1}, x_i]$ отрезком прямой,

проходящей через точки (x_{i-1}, y_{i-1}) и (x_i, y_i) . Площадь получившейся прямоугольной трапеции будет близка к интегралу от заданной функции

$$\int_{x_{i-1}}^{x_i} y(x) dx \approx h \frac{y_{i-1} + y_i}{2}$$

Просуммируем все такие приближенные площади

$$\begin{aligned} \int_a^b y(x) dx &= \int_{x_0}^{x_1} y(x) dx + \int_{x_1}^{x_2} y(x) dx + \dots + \int_{x_{N-1}}^{x_N} y(x) dx \approx \\ &\approx \frac{h}{2} [(y_0 + y_1) + (y_1 + y_2) + \dots + (y_{N-1} + y_N)] \end{aligned}$$

и учтем, что в сумме все значения функции, кроме крайних точек, встречаются дважды. В результате получаем формулу трапеций для расчета определенного интеграла:

$$\int_a^b y(x) dx \approx I_{mp} = h \left(\frac{y_0 + y_N}{2} + \sum_{i=1}^{N-1} y_i \right)$$

Теоретическая оценка погрешности для метода трапеций имеет вид

$$|R_{TP,N}| = \left| I_{mp} - \int_a^b y(x) dx \right| \leq \frac{(b-a)}{12} h^2 M_2.$$

Таким образом, верхний предел погрешности

$$|R_{TP,max}| = \frac{(b-a)}{12} h^2 M_2.$$

По теоретическим оценкам погрешности метод центральных прямоугольников вдвое точнее метода трапеций.

Приведем алгоритм расчета по методу трапеций:

Описать функцию $y(x)$

Задать значения a, b, N

Вычислить $h = (b-a) / 2$

$Sum = (y(a) + y(b)) / 2$

Для i от 1 до $N-1$ выполнять

| $X = a + i * h$
| $Sum = sum + y(x)$

$I = sum * h$

Вывод результата I

4. Метод Симпсона (парабол). В этом методе также используется разбиение отрезка интегрирования $[a,b]$ на N равных частей, причем N должно быть четным, так как приближенная формула применяется к каждой паре элементарных отрезков.

Рассмотрим отрезок $[x_i, x_{i+2}]$ длины $2h$ и применим к подинтегральной функции на этом отрезке квадратичную интерполяцию.

Проведем через три точки (x_i, y_i) , (x_{i+1}, y_{i+1}) , (x_{i+2}, y_{i+2}) параболу с осью, параллельной оси OY .

Уравнение этой параболы можно представить в виде (используем интерполяционный полином Лагранжа второго порядка):

$$y(x) \approx P_2^L(x) = ax^2 + bx + c$$

где коэффициенты a , b и c находятся из условия, что парабола проходит через три заданные точки. На основании условий интерполяции

$$y(x_{i-1}) = y_{i-1}; \quad y(x_i) = y_i; \quad y(x_{i+1}) = y_{i+1};$$

получаем систему трех уравнений для нахождения коэффициентов a , b , c :

$$\begin{cases} ax_{i-1}^2 + bx_{i-1} + c = y_{i-1}; \\ ax_i^2 + bx_i + c = y_i; \\ ax_{i+1}^2 + bx_{i+1} + c = y_{i+1}; \end{cases}$$

Интегрируя затем непосредственно квадратичную функцию, получим, что площадь под параболой на отрезке $[x_i, x_{i+2}]$ равна

$$\int_{x_{i-1}}^{x_{i+1}} y(x) dx \approx \int_{x_{i-1}}^{x_{i+1}} (ax^2 + bx + c) dx = \frac{h}{3} (y_{i-1} + 4y_i + y_{i+1})$$

Пусть $m=N/2$ - число сдвоенных интервалов длины $2h$ (N - четное). Просуммируем все такие приближенные площади:

$$\begin{aligned} \int_a^b y(x) dx &= \int_a^{x_2} y(x) dx + \int_{x_2}^{x_4} y(x) dx + \dots + \int_{x_{2m-2}}^{x_{2m}} y(x) dx \approx \\ &\approx \frac{h}{3} [(y_0 + 4y_1 + y_2) + (y_2 + 4y_3 + y_4) + \dots + (y_{2m-2} + 4y_{2m-1} + y_{2m})] \end{aligned}$$

и учтем, что в сумме все значения функции, относящиеся к четным точкам (кроме крайних точек), встречаются дважды. В результате получаем формулу парабол (Симпсона) для расчета определенного интеграла:

$$\int_a^b y(x)dx = \frac{h}{3} \left(y_0 + y_{2m} + 4 \sum_{i=0}^{m-1} y_{2i+1} + 2 \sum_{i=1}^{m-1} y_{2i} \right) + R_{II}$$

Еще один вариант записи формулы парабол:

$$\int_a^b y(x)dx = \frac{h}{3} \left(y_0 + y_N + \sum_{i=1}^{N-1} [3 + (-1)^{i+1}] y_i \right) + R_{II}$$

Теоретическая оценка погрешности для метода парабол составляет

$$|R_{II}| \leq \frac{(b-a)}{180} h^4 M_4,$$

где M_4 – максимальное значение модуля четвертой производной подинтегральной функции. Таким образом, верхний предел погрешности

$$|R_{II,max}| = \frac{(b-a)}{180} h^4 M_4; \quad M_4 = \max_{[a,b]} |y^{(IV)}|$$

5. Вычисление интегралов с заданной точностью на основании теоретических оценок погрешности. Используя теоретическую оценку максимальной погрешности интегрирования, можно получить, какое число разбиений следует выбрать, чтобы получить результат с требуемой точностью.

$$\int_a^b y(x)dx = \sum_i A_i y(x_i) + R$$

Например, для метода центральных прямоугольников теоретическая оценка максимальной погрешности интегрирования определяется формулой:

$$|R_{ЦП,max}| = \frac{(b-a)h^2}{24} M_2$$

Считая, что эта оценка не должна превышать требуемую точность вычислений, получаем формулы для нахождения максимально возможного шага h (или минимально возможного числа разбиений N)

$$R_{\text{цп},\max} < \varepsilon, \rightarrow \frac{(b-a)}{24} h^2 M_2 < \varepsilon,$$

откуда следует

$$h < \sqrt{\frac{24\varepsilon}{(b-a)M_2}}.$$

Можно выбрать любое значение h , удовлетворяющее этому условию. Однако следует помнить, что h должно определять *согласованную* сетку, то есть, на заданной области $[a;b]$ должно помещаться **целое число шагов**.

С учетом взаимосвязи $h=(b-a)/N$ можно получить формулу для выбора числа шагов

$$\frac{(b-a)^3}{24N^2} M_2 < \varepsilon \rightarrow N > \sqrt{\frac{(b-a)^3 M_2}{24\varepsilon}},$$

причем значение N следует округлить до ближайшего целого в большую сторону.

Аналогичные соотношения можно получить и для метода трапеций:

$$h < \sqrt{\frac{12\varepsilon}{(b-a)M_2}} \text{ или } N > \sqrt{\frac{(b-a)^3 M_2}{12\varepsilon}}$$

и метода парабол

$$h < \sqrt[4]{\frac{180\varepsilon}{(b-a)M_4}} \text{ или } N > \sqrt[4]{\frac{(b-a)^5 M_4}{180\varepsilon}}.$$

Пример. Вычислить интеграл $\int_0^1 x \sin x dx$ методом центральных прямоугольников с точностью $\varepsilon=10^{-6}$. Найдем производные подинтегральной функции

$$y(x) = x \sin x; \quad y'(x) = \sin x + x \cos x; \quad y''(x) = 2 \cos x - x \sin x;$$

$$|y''| < |x \sin x| + 2|\cos x|; \quad \max_{[0;1]} |y''| = 3$$

$$N_{\text{цп}} > \sqrt{\frac{1^3 \cdot 3}{24 \cdot 10^{-6}}} = \sqrt{\frac{10^6}{8}} = 340.$$

Значит, нужно применить квадратурную формулу центральных прямоугольников при $N=340$, чтобы получить результат с точностью до одной миллионной.

Примечание. Теоретические оценки дают величину максимально возможной погрешности. На практике реальная погрешность численного интегрирования может быть существенно меньше.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА 1. Составить программу, содержащую:

- вычисление подинтегральной функции $y(x)$ в виде подпрограммы-функции;
- вычисление первообразной функции $Y(x)$ в виде подпрограммы-функции;
- подпрограмму-функцию вычисления определенного интеграла указанным в варианте численным методом. Параметры подпрограммы: пределы интегрирования $[a;b]$ и количество разбиений N ;
- ввод пределов интегрирования и числа разбиений N с клавиатуры;
- численный расчет заданного определенного интеграла посредством вызова соответствующей подпрограммы-функции;
- вычисление точного значения определенного интеграла обращением к первообразной функции (контроль точности в тестовом примере);
- вычисление относительной погрешности численного интегрирования

$$\delta = \left| \frac{I_{\text{точн.}} - I_{\text{числен.}}}{I_{\text{точн.}}} \right| * 100\%.$$

Путем многократного выполнения программы получить:

а) значения относительной погрешности при $N = 2, 4, 6, \dots, 20$.

Построить в отчете таблицу зависимости $\delta(N)$.

б) Оценить, при каком значении N относительная погрешность численного результата станет меньше 10^{-5} .

Результаты записать в отчет.

Метод центральных прямоугольников	Метод Трапеций	Метод Симпсона
1) $\int_0^{\pi} \frac{dx}{(x+2)(x+3)}$	2) $\int_{2\pi}^{3\pi} (0,12x^4 - 2,6x^5) dx$	3) $\int_{0.2}^{2.8} (2x^4 + \sin x) dx;$
4) $\int_{0.3}^{\pi/2} \frac{dx}{x^2 + 2}$	5) $\int_{1.3}^{4.5} (x^6 + 0,2x^3) dx$	6) $\int_{1.3}^{7.4} \cos(x - 0.4) dx$
7) $\int_{0.3}^{7.5} \sin^2 x dx$	8) $\int_{\pi/8}^{7\pi/8} (2\sqrt{x} + \sin 2x) dx$	9) $\int_{1.0}^{8.4} (\sin x + \cos x) dx$
10) $\int_{0.1}^{6.5} (x^5 - 1,2x^3) dx$	11) $\int_{0.2}^{9.8} (3x^4 + 4,6x^3) dx$	12) $\int_{0.2}^{1.9} \cos(2x + 0,1) dx$
13) $\int_0^{\pi/4} \frac{dx}{x^2 + 2}$	14) $\int_{2.0}^{9.5} \exp(0,2x + 0,3) dx$	15) $\int_{1.5}^{3.7} \frac{dx}{x + 0.2}$

ЗАДАЧА 2. Используя теоретическую оценку погрешности численного метода, найти значение N (и, соответственно, h), обеспечивающее получение результата с точностью $\varepsilon=10^{-6}$. Получить результат и сравнить его с точным. Решение задачи изложить в отчете.

ЗАДАЧА 3. Преобразовать подпрограмму, вычисляющую определенный интеграл, так, чтобы имя подинтегральной функции являлось параметром подпрограммы. Включить подпрограмму в собственный модуль и проверить ее работу.

ЗАДАЧА 4. Преобразовать подпрограмму, вычисляющую определенный интеграл с заданной точностью, так, чтобы имя подинтегральной функции являлось параметром подпрограммы. Включить подпрограмму в собственный модуль и проверить ее работу.

ТЕМА 7. ВЫЧИСЛЕНИЕ ОПРЕДЕЛЕННЫХ ИНТЕГРАЛОВ С ЗАДАННОЙ ТОЧНОСТЬЮ

Цель работы: изучение и программирование метода удвоения числа разбиений для вычисления определенных интегралов с заданной точностью.

Вычисление интегралов с заданной точностью методом удвоения числа разбиений. Этот метод позволяет автоматически получить заданную точность результата без использования теоретических оценок погрешности.

Так как оценить старшие производные удается далеко не всегда, то на практике для вычисления интегралов используется метод автоматического достижения нужного числа разбиений.

Процесс основан на выполнении свойства сходимости

$$\lim_{N \rightarrow \infty} \left| \int_a^b y(x) dx - \sum_i A_i y_i \right| = 0.$$

Он заключается в следующем: задают некоторое начальное значение N , (или h) и выполняют расчет при этом N , используя составленную заранее подпрограмму-функцию.

$$N=2 \\ V = \text{Integral}(a, b, N)$$

Затем вдвое увеличивают N , (уменьшают вдвое h), и снова считают по той же формуле новую сумму W .

$$N=N*2 \\ W = \text{Integral}(a, b, N)$$

Затем сравнивают между собой V и W . Если разница между ними велика, то, на следующем шаге значение W принимают в качестве исходного (переобозначая $V:=W$) и снова увеличивают вдвое число шагов разбиения N . Процесс этот продолжается до тех пор, пока две последовательно вычисленные суммы не будут отличаться одна от другой на величину, меньшую заданной точности. С помощью такого процесса двойного пересчета происходит автоматический выбор числа шагов N и, соответственно, шага h . Можно выделить следующие элементы такого расчета (псевдокод):

```

Подпрограмма-функция y(x)
Подпрограмма-функция Integral(a,b,N)
Ввод a,b,ε
N=2;
V=Integral(a,b,N)
Повторять
  N=2*N
  W=Integral(a,b,N)
  {текущий вывод N,W}
  r=|W-V| {Если |W|>1, то r=|(W-V)/W| иначе r=|W-V|}
  V=W
До получения результата r<ε
Вывод результата N,W

```

Алгоритм двойного пересчета удобно оформить в виде подпрограммы.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА. Используя программу - решение предыдущей Л.р. в качестве основы, составить новую программу, в которой тот же интеграл вычислялся бы с задаваемой с клавиатуры точностью ϵ .

Для этого вычисление интеграла при произвольном N оформить как подпрограмму-функцию, а в главной программе реализовать алгоритм двойного пересчета. Начальное значение $N=2$. Для контроля за вычислениями вывести на экран и в текстовый файл точное значение интеграла, все промежуточные значения N , и $I(N)$ и окончательный численный результат.

Какое число разбиений понадобится для получения результата с точностью $\epsilon = 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$?

Включить в отчет блок-схему вычисления интеграла, текст программы и полученные результаты.

ТЕМА 8. ИЗУЧЕНИЕ И ПРОГРАММИРОВАНИЕ ЧИСЛЕННЫХ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ КОШИ ДЛЯ ОДУ-1

Цель работы: Изучение простейших численных методов решения задачи Коши для обыкновенных дифференциальных уравнений первого порядка.

1. Задача Коши для ОДУ-1. Поведение разнообразных физических систем зачастую определяется тем, как скорость изменения одной или нескольких величин зависит от других величин. Такие ситуации описываются дифференциальными соотношениями (скорость - это производная) и приводят к дифференциальным уравнениям или системам дифференциальных уравнений.

Многие дифференциальные уравнения, возникающие в прикладных задачах, не решаются аналитически. Поэтому универсальными, а в большинстве случаев и единственными методами решения ОДУ являются численные методы. Настоящая лабораторная работа посвящена изучению и применению простейших одношаговых численных методов решения задачи Коши для обыкновенных дифференциальных уравнений 1-го порядка.

Простейший случай - дифференциальное уравнение первого порядка

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

На плоскости (x, y) оно имеет бесконечное множество решений $y = Y(x) + C$, называемых интегральными кривыми.

Для дальнейшего очень важно учитывать геометрический смысл функции $f(x, y)$. В общем случае она определена для любых значений x и y на плоскости и в КАЖДОЙ точке (x, y) определяет наклон касательной к одной из интегральных кривых $tga = dy/dx = f(x, y)$.

Для нахождения конкретного решения дифференциального уравнения на отрезке $[a, b]$ необходимо задать дополнительные условия. Возможны два варианта:

а) условия задаются в начальной точке интервала (задача с начальными условиями - задача Коши);

б) условия задаются на обеих границах интервала (граничная, или краевая задача).

Для дифференциального уравнения первого порядка задача Коши заключается в нахождении функции $y(x)$ по заданному дифференциальному уравнению и начальному условию:

$$\frac{dy}{dx} = f(x, y); \quad y(x_0) = y_0; \quad a \leq x \leq b; \quad y(x) = ?$$

Для решения задачи на интервале $[a;b]$ с достаточно малым шагом h строится система равноотстоящих точек

$$x_i = a + i \cdot h; \quad i = 0, 1, \dots, N; \quad x_N = b; \quad h = \frac{b-a}{N}.$$

В одношаговых численных методах значения приближенного решения вычисляются последовательно от точки к точке, начиная с x_0 , причем для вычисления решения на i -м шаге используется значение решения, найденного на предыдущем $(i-1)$ -м шаге.

2. Метод Эйлера и его модификации. Наиболее простым и наглядным методом решения задачи (1) является метод Эйлера.

По определению, производная есть предел отношения приращения функции к приращению аргумента при стремлении последнего к нулю:

$$\frac{dy}{dx} = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x}.$$

Чем ближе $\Delta x = x_{\text{кон}} - x_{\text{нач}}$ к нулю, тем точнее отношение $\Delta y / \Delta x$ дает значение реальной производной. Метод Эйлера основан на неполном выполнении предельного перехода, в нем считается, что Δx - очень малая, но конечная величина, так что

$$\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x} = \frac{y_{\text{кон}} - y_{\text{нач}}}{x_{\text{кон}} - x_{\text{нач}}}.$$

Если задача Коши решается на большом отрезке $[a;b]$, то приращение аргумента $(b-a)$ велико, его нельзя брать в качестве Δx . Выход заключается в разбиении интервала на большое число N отрезков с малой длиной $h=(b-a)/N$. Точки разбиения пронумеруем:

$$x_0 = a; \quad x_1 = a + h; \quad x_2 = a + 2h; \dots \quad x_N = b;$$

$$x_i = x_0 + i \cdot h; \quad i = 0..N.$$

Параметр h задает постоянное расстояние между двумя соседними точками

$$x_i - x_{i-1} = h .$$

Такой набор дискретных точек называется сеткой (при $h=\text{const}$ получаем равномерную сетку). Значение функции в точке x_i обычно обозначается $y(t_i)=y_i$.

В соответствии с уравнением, в каждой точке x_i получаем приближенное соотношение

$$\left. \frac{dy}{dx} \right|_{x_i} \approx \frac{\Delta y_i}{\Delta x_i} = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} = \frac{y_i - y_{i-1}}{h} = f(x_{i-1}, y_{i-1}),$$

откуда следует расчетная формула метода Эйлера:

$$y_i = y_{i-1} + hf(x_{i-1}, y_{i-1}) .$$

Используя начальные значения (x_0, y_0) , можно найти приближенные значения функции $y(x)$ в узловых точках x_i . Вычисления ведутся последовательно от точки к точке, т.е, формула метода Эйлера используется для $i=1, 2, \dots, n$.

$$y_1 = y_0 + hf(x_0, y_0)$$

$$y_2 = y_1 + hf(x_1, y_1)$$

.....

$$y_N = y_{N-1} + hf(x_{N-1}, y_{N-1}) .$$

Метод Эйлера обладает исключительной простотой, однако имеет малую точность, что резко ограничивает возможности его применения. Вместе с тем, метод Эйлера служит основой для разработки более точных методов и используется в них как отдельный этап вычислений.

Погрешность расчета по методу Эйлера на каждом шаге связана с тем, что численный результат определяется наклоном касательной, а сама касательная строится в *левой* точке отрезка. Реальная функция обычно отличается от прямой, и на протяжении отрезка это отличие становится заметным.

3. Метод Эйлера-Коши. В методе Эйлера-Коши на каждом шаге расчет по формуле Эйлера рассматривается как предварительный результат

$$\tilde{y}_i = y_{i-1} + hf(x_{i-1}, y_{i-1}); \quad x_i = x_{i-1} + h;$$

который на этом же шаге затем уточняется

$$y_i = y_{i-1} + \frac{h}{2}(f(x_{i-1}, y_{i-1}) + f(x_i, \tilde{y}_i)).$$

4. Улучшенный метод Эйлера. В этом методе на каждом шаге сначала вычисляют наклон касательной к интегральной кривой в середине интервала

$$x_{i+1/2} = x_{i-1} + \frac{h}{2}; \quad y_{i+1/2} = y_{i-1} + \frac{h}{2} f(x_{i-1}, y_{i-1}).$$

Проведение прямой с таким наклоном из исходной точки x_{i-1} дает более точный результат

$$y_i = y_{i-1} + hf(x_{i+1/2}, y_{i+1/2}).$$

По сравнению с обычным методом Эйлера его модификации обладают значительно более высокой точностью.

5. Программная реализация метода Эйлера. Расчет по методу Эйлера заключается в выполнении цикла по всем точкам сетки и заполнении массивов $\{x_i\}$, $\{y_i\}$. В тестовых задачах, когда известно точное решение, для каждой точки x_i вычисляется также точный результат $u_{i,\text{точн}}$ и абсолютная погрешность Δ_i .

В соответствии с требованиями языка Турбо-Паскаль структура используемых массивов должна быть описана в описании Type, а их имена - в описании Var.

```
Program Name;  
- описание типа массивов;  
- описание переменных;  
- подпрограмма-функция вычисления правой части ДУ  
F(x, y);  
- подпрограмма-функция вычисления точного решения  
Ftochn(x); (для тестовых расчетов)  
Begin  
  - ввод исходных данных a, b, N, x0, y0;  
  h:=(b-a)/N;  
  - вычисление элементов массивов;  
  x[0]:=a; y[0]:=y0; Delta[0]:=0;  
  For l:=1 to N do begin  
    x[i]:=x[i-1]+h;
```

```

y[i]:=y[i-1]+h*F(x[i-1],y[i-1]);
ytochn[i]:=Ftochn(x[i]);
Delta[i]:=Abs(ytochn[i]-y[i]);

```

end;

- вывод результатов в виде таблицы;
 - вывод результатов в текстовый файл для печати
 - построение графика решения
- End.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА 1. Для заданного дифференциального уравнения 1-го порядка (условие получить у преподавателя) решить задачу Коши: а) аналитически (проверить решение подстановкой); б) численно методом Эйлера.

Для выполнения численного решения составить программу, которая содержит:

- описание массивов X и Y для хранения значений аргумента и функции;
- процедуру вывода на экран заставки с подробным описанием программы;
- процедуру ввода параметров задачи: границ интервала [a,b], начального условия y_0 и числа шагов N;
- процедуру вычисления в точках x_i на интервале [a,b] точного и численного значений решения и относительной погрешности численного решения $\delta_{i\%}$ (в процентах), а также максимальной погрешности на интервале;
- процедуру вывода на экран и в текстовый файл (для последующего вывода на принтер) результатов вычислений в виде таблицы [i; x_i ; $y_{i, \text{точн}}$; $y_{i, \text{числ}}$; $\delta_{i\%}$].

При выводе информации на экран и вводе исходных данных использовать процедуры оформления диалога.

Выполнить отладку программы с помощью тестовой задачи:
 $y' = x^2/y$; $y(0) = 1$; $y_{\text{точн}}(x) = (2x^3/3)^{1/2} + 1$.

ЗАДАЧА 2. Дополнить программу таким образом, чтобы решение выполнялось и методом Эйлера-Коши.

ЗАДАЧА 3. Дополнить программу таким образом, чтобы решение выполнялось и улучшенным методом Эйлера.

ТЕМА 9. ИССЛЕДОВАНИЕ ФИЗИЧЕСКИХ СИСТЕМ НА ОСНОВЕ ЧИСЛЕННОГО РЕШЕНИЯ ОДУ ПЕРВОГО ПОРЯДКА

Цель работы: Отработка навыков решения физических задач на основе численного решения ОДУ.

СОДЕРЖАНИЕ ЗАДАНИЯ

Сформулировать соответствующую физической задаче задачу Коши. Решить ее аналитически. Для численного решения составить программу, отвечающую требованиям предыдущих лабораторных работ, с использованием улучшенного метода Эйлера. Результаты вывести на экран в виде таблицы точного и численного решений постранично по 20 строк и записать в файл. Напечатать график численного решения. Создать EXE-файл программы и в ходе многократных расчетов (изменяя при необходимости исходные параметры) исследовать характерные особенности поведения физической системы. Напечатать графики исследуемых величин для наиболее интересных или типичных случаев.

Провести графическое исследование решения и ответить на вопросы к задаче.

1. Ракета с начальной массой $M=5100$ кг движется в поле силы тяжести вертикально вверх под действием постоянной реактивной силы $F=3 \cdot 10^5$ Н, при этом масса ее уменьшается на $\Delta m=200$ кг за секунду. Приняв, что ракета стартовала в момент времени $t=0$, исследовать поведение физической системы:

- вычислить ее скорость через $T=(M-m_{\text{ост}})/\Delta m$ сек после старта, если остаточная масса $m_{\text{ост}}=200$ кг.
- на сколько процентов изменится максимальная скорость подъема, если $\Delta m=180$ кг/с;
- какой величины должна быть постоянная реактивная сила F , чтобы конечная скорость полета была 8 км/с;
- как изменится величина скорости через T сек после старта, если учитывать сопротивление атмосферы $f_{\text{сопр}}=-0.001 v^2$;
- чему должна быть равна остаточная масса $m_{\text{ост}}$, чтобы конечная скорость ракеты была равна 8000 км/с.

2. Начальная температура котла равна температуре окружающей среды $T = 20^\circ \text{C}$. Котел получает тепло от нагревательного прибора. Скорость подачи тепла $\Phi(t) = 0,29t^2$ Дж/с. Теплоемкость котла $C = 0,88 \cdot 10^3$ Дж/град. Кроме того, котел отдает тепло окружающей среде (скорость охлаждения пропорциональна разности между температурами котла и среды с коэффициентом $\lambda_0 = 2,7 \cdot 10^2$ Дж/(сек·град)). Исследовать поведение физической системы:

- найти температуру через 200 сек от начала нагревания;
- каким должен быть коэффициент в функции $\Phi(t)$, чтобы конечная температура была равна 80 градусов;
- найти время, через которое конечная температура котла станет равной 120 градусов;
- выполнить расчеты для $T_0 = 0^\circ \text{C}$;
- как изменится время нагревания до температуры $T = 100^\circ \text{C}$, если коэффициент λ будет линейно зависеть от температуры $\lambda = \lambda_0 (1 + \alpha T)$, где $\alpha = 1/273$ 1/град..

3. Модель ракеты движется в поле силы тяжести вертикально вверх, выбрасывая непрерывную струю газов с постоянной относительной скоростью $V = 800$ м/с. Расход газа $\mu = 0,4$ кг/с, начальная масса модели ракеты $m_0 = 3,2$ кг. Начальная скорость равна 0 . Исследовать поведение физической системы:

- какую скорость относительно Земли приобретает ракета через время $T = (m_0 - m_1) / \mu$ сек после начала движения ($m_1 = 0,2$ кг);
- какой должна быть относительная скорость газовой струи, чтобы конечная скорость модели была равна 600 м/с;
- во сколько раз изменится конечная скорость, если расход газа увеличится в $1,4$ раза;
- как изменится величина скорости через T сек после старта, если учитывать сопротивление атмосферы $f_c = -0,001V^2$.

4. Парашютист опускается на парашюте, имеющем форму полусферы радиуса $R = 4$ м. Его масса вместе с массой парашюта $m = 115$ кг. Сила сопротивления воздуха $F = -\gamma S V^2$, где S - площадь наибольшего сечения, перпендикулярного к направлению движения, V - скорость движения, $\gamma = 0,81$ 1/м³. Считая, что парашют раскрылся при начальной скорости $V_0 = 60$ м/с, исследовать поведение физической системы:

- найти скорость V парашютиста через $T = 1,5$ сек после раскрытия парашюта;

- через какое время скорость падения установится практически постоянной (изменяется менее чем на 5% за 1 сек);

- как изменится время установления стационарной скорости падения, если радиус парашюта будет равен 5,6 метра.

5. Скорость охлаждения радиатора пропорциональна разности между его температурой и температурой окружающей среды. Коэффициент пропорциональности в некоторых случаях линейно зависит от времени $K = K_0(1+\alpha t)$. Начальная температура $T(0) = 150^\circ\text{C}$, а температура окружающей среды $T_1=0^\circ\text{C}$. Для расчета принять $\alpha = 1/273$ 1/с; $K_0=1,85 \cdot 10^{-2}$ Дж/(с·град). Исследовать при этом предположении поведение физической системы:

- найти температуру радиатора T в момент времени $t = 60$ с;

- через какое время температура радиатора уменьшится в 3 раза;

- какой должна быть начальная температура радиатора, чтобы через 2 минуты он остыл до 45 градусов.

6. Катушка индуктивности подключена к внешнему источнику напряжения. Коэффициент самоиндукции $L = 5$ Гн, сопротивление обмотки $R = 1,5$ Ом. Электродвижущая сила меняется по закону $E = E_0 \sin(\omega t)$. Исследовать поведение тока в зависимости от времени $I(t)$, если начальный ток $I(0)=0$, $E_0 = 4,2$ В, $\omega=2\pi/5$ 1/с:

- построить график тока в катушке за время 12 сек;

- как изменится поведение силы тока в цепи, если $L=0,5$ Гн;

- определить силу тока в момент $t=5$ сек, если частоту ω увеличить вдвое;

- найти величину тока в катушке в момент $t=0,8$ сек;

- чему равно амплитудное значение силы тока;

- как изменится сила тока в момент $t=2$ сек, если частоту ЭДС уменьшить вдвое.

7. Деревянный шар радиуса r падает с высоты H в воду. Учитывая силу вязкого трения (силу Стокса) и принимая $r=3$ см, $H=500$ см, вычислить:

- скорость шара через 0,4 сек после падения в воду;

- время, через которое шар достигнет максимальной глубины погружения;

- как изменится время достижения максимальной глубины погружения, если радиус шара увеличить в 2 раза;
- как изменится время достижения максимальной глубины погружения, если H уменьшить в 2 раза;
- время, через которое шар, всплывая, достигнет поверхности воды (для ответа на этот вопрос переформулировать уравнение относительно dV/dh);
- как изменится время всплытия, если радиус шара увеличить в 1,5 раза;

8). К источнику с ЭДС $E=10$ В через сопротивление $R=2000$ Ом подключают конденсатор емкостью $C=10^{-6}$ Ф. Исследовать изменение заряда на конденсаторе с течением времени до момента $t=0.05$ сек.

- каков характер изменения $Q(t)$?
- через какое время заряд на конденсаторе практически перестанет изменяться;
- как изменится время насыщения, если $R=3700$ Ом;
- как изменится время насыщения, если $C=10^{-5}$ Ф;
- задавая значение ЭДС 20; 30; 40 В, выяснить, как влияет на решение величина E ;
- задавая значение емкости $C=10^{-3}$; 10^{-4} ; 10^{-5} ; 10^{-6} Ф, выяснить, как влияет на решение величина C .

9). Футбольный мяч, имеющий массу 380 г, брошен вертикально вверх с начальной скоростью $v_0=18$ м/сек. Сопротивление воздуха пропорционально квадрату скорости и составляет 4,8 Н при скорости 1 м/с. Сформулируйте дифференциальное уравнение, выражающее скорость движения через высоту подъема над поверхностью Земли.

- найдите время и наибольшую высоту подъема мяча без учета сопротивления воздуха. Убедитесь в том, что в этом случае время подъема и время падения мяча одинаковы, а общее время движения до падения на поверхность Земли составляет v_0/g ;
- повторите вычисления с учетом силы сопротивления $F_c = -k_c v|v|$, $k_c=0.003$. Как изменились значения времени подъема, времени падения и максимальной высоты подъема;
- какой должна быть начальная скорость, чтобы время подъема с учетом сопротивления было равно 0.8 сек;
- чему должен быть равен коэффициент сопротивления, чтобы время подъема было в 1,2 раза больше, чем время падения.

10. Найти количество теплоты, необходимое для нагрева 1 кг железа от 20°C до 200°C . Удельная теплоемкость железа выражается зависимостью $C(t)=0,1053+0,000142\cdot T$, где T - температура. Результат выразить в Дж и калориях. Выяснить также:

- как увеличится необходимое количество теплоты при нагревании от 20 до 600 градусов;
- как относятся результаты для интервалов нагревания 20-200 и 200-400 градусов;
- на сколько увеличится результат, если масса железа будет равна 1,5 кг.

11. Ракета запущена вертикально вверх с начальной скоростью $v_0=100$ м/с. Соппротивление воздуха замедляет ее движение с силой $F_c=-kv^2$. Определить время достижения ракетой наивысшего положения, если $k=0,004$ кг/м. Выяснить также:

- как изменится время подъема, если $v_0=140$ м/с;
- какое значение будет иметь скорость ракеты через 35 с;
- сколько времени потребуется для снижения скорости ракеты до 30 м/с.

12. Капля дождя, имеющая начальную массу $m_0=0.01$ г, свободно падает в воздухе с большой высоты с начальной скоростью $v_0=0$. При падении она равномерно испаряется со скоростью $d=0,00001$ г/с. Сила сопротивления воздуха пропорциональна скорости движения капли и квадрату ее радиуса $F_c=-\alpha R^2 v^2$, причем $\alpha=1,2\cdot 10^{-6}$ г/см³ и $\alpha \neq d$. Найти скорость движения капли через 10 с после начала движения. Определить также:

- как изменится скорость падения капли, если коэффициент α уменьшится вдвое;
- чему будет равно время движения капли до полного испарения, считая высоту достаточно большой;
- во сколько раз изменятся результаты, если пренебречь действием силы тяжести.

13. Температура вынутого из печи хлеба в течение 20 мин падает от 100°C до 60°C . Температура окружающего воздуха $t_0=25^{\circ}\text{C}$. Через какое время от момента начала охлаждения температура хлеба понизится до 30°C . Выяснить также:

- как изменится время охлаждения, если $t_0=18^\circ\text{C}$;
- какое значение будет иметь температура хлеба через 50 мин;
- сколько времени потребуется для охлаждения до температуры 23°C .

14. Электрическая цепь состоит из последовательно соединенных источника напряжения $E=300\text{ В}$, сопротивления $R=150\text{ Ом}$ и катушки индуктивности с коэффициентом самоиндукции $L=3,08\text{ Гн}$. За какое время с момента замыкания цепи возникающий в ней ток установится практически постоянным. Выяснить также:

- за какое время ток увеличится до значения $0,75\text{ А}$;
- при каком значении L время установления тока будет равно $0,1\text{ сек}$.

15. Ракета запущена вертикально вверх с начальной скоростью $V_0=100\text{ м/с}$. Сопротивление воздуха замедляет ее движение с силой $F_c=-kV^2$. Рассматривая зависимость скорости от высоты подъема, определить максимальную высоту подъема, если $k=0,004\text{ кг/м}$.

Выяснить также:

- как изменится максимальная высота подъема, если $V_0=140\text{ м/с}$;
- чему будет равна скорость ракеты на высоте 2000 м ;
- на какой высоте скорость ракеты снизится до 30 м/с .

16. Разность потенциалов на зажимах катушки индуктивности равномерно уменьшается от величины внешней электродвижущей силы $E_0=2\text{ В}$ до значения $E_1=1\text{ В}$ в течение 10 секунд . Сопротивление катушки $R=0,12\text{ Ом}$, коэффициент самоиндукции $L=0,1\text{ Гн}$. Найти силу тока в конечный момент времени, если в начале опыта она составляла $I_0=16,2\text{ А}$. Выяснить также:

- время, через которое сила тока будет равна 10 А ;
- силу тока в момент $t=1\text{ с}$;
- какое значение должен иметь параметр L , чтобы через 8 сек после начала опыта сила тока была равна $0,8\text{ А}$;
- выбирая значения параметра R большие и меньшие указанного, исследовать качественно характер влияния этого параметра на изменение силы тока;
- аналогичным образом выяснить влияние параметра L .

17. Снаряд массой $m=20$ кг выпущен из зенитного орудия вертикально вверх со скоростью $v_0=800$ м/с. Считая силу сопротивления воздуха пропорциональной скорости с коэффициентом $k=0,24$ кг/с, определить время T подъема снаряда до высшей точки. Выяснить также:

- как изменится результат, если сопротивлением воздуха пренебречь;
- на какую высоту поднимется снаряд за 3 сек;

18. В двух чашках находится по 0,2 л кипяченой воды при температуре $t_0=90^\circ\text{C}$. Через 3 минуты в первую чашку добавляют 50 мл воды, имеющей температуру $t_1=20^\circ\text{C}$. Во вторую чашку таким же образом добавляют воду через 6 минут. В какой чашке температура воды через 12 минут после начала опыта будет больше. Температура окружающей среды $T_c=20^\circ\text{C}$, коэффициент теплообмена $\eta=0,62$ Вт/(м·град). Выяснить также:

- температуру в момент $t=3$ мин в обоих случаях;
- какое значение должен иметь параметр η , чтобы через 8 мин после начала опыта температура была равна 34°C ;
- выбирая значения параметра η большие и меньшие указанного, исследовать качественно характер влияния этого параметра на изменение температуры;

19. Изолированному проводнику сообщен заряд $Q=1000$ Кл. Вследствие несовершенства изоляции проводник постепенно теряет свой заряд. Скорость потери заряда в текущий момент времени пропорциональна величине оставшегося заряда проводника. Какой заряд останется на проводнике по истечении времени $t=10$ мин, если за первую минуту потеряно $d=80$ Кл. Выяснить также:

- время, через которое заряд будет равен 400 Кл;
- заряд в момент $t=7$ мин;
- какое значение должен иметь параметр d , чтобы через 8 мин после начала опыта заряд принял значение 700 Кл;
- выбирая значения параметра d большие и меньшие указанного, исследовать качественно характер влияния этого параметра на изменение заряда изолированного проводника.

20. Начальная скорость пули $u_0=800$ м/с. При движении в воздухе за время $t=0,8$ с ее скорость уменьшилась до $u=400$ м/с. Масса пули

$m=10$ г. Считая силу сопротивления воздуха пропорциональной u^α , $\alpha=17/7$, определить коэффициент сопротивления k . Действием силы тяжести пренебречь. Выяснить также:

- время, через которое скорость будет равна 180 м/с;
- скорость в момент $t=1,8$ с;
- какое значение должен иметь параметр k , чтобы через 2 сек скорость была равна 160 м/с;
- выбирая значения параметра α большие и меньшие указанного, исследовать качественно характер влияния этого параметра на изменение скорости пули;
- аналогичным образом выяснить влияние параметра k .

21. С вертолета, неподвижно висящего на некоторой высоте над поверхностью Земли, сброшен груз массой $m=100$ кг. Считая, что сила сопротивления воздуха изменяется пропорционально скорости, определить, через какой промежуток времени Δt скорость груза будет равна 5 м/с. Коэффициент сопротивления $k=200$ Н/с. Выяснить также:

- скорость в момент $t=2$ с;
- какое значение должен иметь параметр k , чтобы через 0,4 с после начала движения скорость приняла значение 8 м/сек;
- выбирая значения параметра m большие и меньшие указанного, исследовать качественно характер влияния этого параметра на изменение скорости.

22. Трубопровод тепловой магистрали (диаметр 20 см) защищен изоляцией толщиной $L=10$ см, причем величина коэффициента теплопроводности изоляции $k=0,00017$. Температура поверхности трубы 160°C , наружная температура изоляции 30°C . Найти распределение температуры внутри изоляции. Выяснить также:

- радиус, на котором температура уменьшается вдвое;
- выбирая значения параметра k большие и меньшие указанного, исследовать качественно характер влияния этого параметра на распределение температуры.

Справочные сведения. Теплоемкость тела определяется как $C=dQ/dt$. Удельная теплоемкость определяет эту величину, приходящуюся на единицу массы

$$C(t) = \frac{1}{m} \frac{dQ}{dt}.$$

Закон теплопроводности Фурье: $q = -\lambda(t) \frac{dt}{dx}$, где q – удельный поток теплоты; λ - коэффициент теплопроводности; dt/dx - градиент температуры.

Закон Ньютона для внешнего теплообмена: скорость охлаждения тела пропорциональна разности температур тела и окружающей среды $\frac{d\theta}{d\tau} = -k(\theta - \theta_0)$, где τ - время, θ - температура поверхности тела, θ_0 - температура окружающей среды.

При исследовании зависимости $v(x)$ используется замена

$$\frac{d^2 x}{dt^2} = \frac{dv}{dt} = \frac{dv}{dx} \cdot \frac{dx}{dt} = v \frac{dv}{dx}.$$

ТЕМА 10. ЧИСЛЕННОЕ РЕШЕНИЕ ТЕСТОВОЙ ЗАДАЧИ КОШИ ДЛЯ ОДУ ВТОРОГО ПОРЯДКА

Цель работы: Изучение метода численного решения задачи Коши для обыкновенных дифференциальных уравнений 2-го порядка.

1. Обобщение расчетных формул на случай системы ОДУ 1-го порядка. Ранее были рассмотрены методы численного решения задачи Коши для одного ОДУ первого порядка

$$\frac{dx}{dt} = f(t, x); \quad x(t_0) = x_0$$

Физический смысл такой задачи - нахождение координаты частицы при одномерном движении (вдоль оси OX) при заданной скорости $F(t,x)$.

При движении частицы в пространстве аналогичное уравнение получаем для радиус-вектора

$$\frac{d\vec{r}}{dt} = f(t, \vec{r}); \quad \vec{r}(t_0) = \vec{r}_0.$$

В проекциях на оси координат оно эквивалентно системе трех дифференциальных уравнений

$$\frac{dx_1}{dt} = f_1(t, x_1, x_2, x_3); \quad x_1(t_0) = x_{10};$$

$$\frac{dx_2}{dt} = f_2(t, x_1, x_2, x_3); \quad x_2(t_0) = x_{20};$$

$$\frac{dx_3}{dt} = f_3(t, x_1, x_2, x_3); \quad x_3(t_0) = x_{30}.$$

Формы записи эквивалентны, - всегда можно перейти от одной к другой и наоборот.

Для получения расчетных формул, например, метода Эйлера, следует заменить непрерывную производную вектора $d\vec{r}/dt$ конечно-разностной

$$\frac{d\vec{r}}{dt} \approx \frac{\Delta\vec{r}}{\Delta t} = \frac{\vec{r}(t+h) - \vec{r}(t)}{h}.$$

Очевидно, все формулы будут иметь ту же форму, если в соотношениях заменить переменную x на вектор \vec{r} .

2. Формулировка численных методов для решения задачи Коши для ОДУ 2-го порядка. Любой из описанных ранее численных методов решения ОДУ первого порядка можно использовать для решения дифференциальных уравнений более высоких порядков и систем дифференциальных уравнений.

При решении задачи Коши (задачи с начальными условиями) для дифференциального уравнения N -го порядка с помощью $N-1$ вспомогательных функций уравнение сводится к системе N уравнений 1-го порядка.

Например, в дифференциальном уравнении 2-го порядка

$$\frac{d^2x}{dt^2} = f(t, x, \frac{dx}{dt}); \quad x(t_0) = x_0; \quad \frac{dx}{dt}(t_0) = v_0;$$

можно ввести вспомогательную функцию

$$z = \frac{dx}{dt}.$$

Тогда

$$\frac{d^2x}{dt^2} = \frac{d}{dt} \frac{dx}{dt} = \frac{dz}{dt},$$

и мы приходим к задаче Коши для системы двух уравнений 1-го порядка

$$\frac{dx}{dt} = g(t, x, y) = z; \quad \frac{dz}{dt} = f(t, x, y); \quad x(t_0) = x_0; \quad z(t_0) = v_0;$$

3. Пример преобразования уравнения движения нелинейного осциллятора. Одномерное движение тела массы m в поле нелинейного периодического потенциала в среде с сопротивлением имеет вид

$$m \frac{d^2x}{dt^2} + a \frac{dx}{dt} + cx - \sigma x^{5/2} \sin t = 0; \quad x(0) = 2; \quad \frac{dx}{dt}(0) = 1;$$

Здесь a , c , σ - коэффициенты сопротивления среды, упругой силы и интенсивности внешней силы.

Введем вспомогательную функцию $z(t) = dx/dt$. Тогда исходная задача преобразуется к системе уравнений 1-го порядка:

$$\frac{dx}{dt} = z; \quad \frac{dz}{dt} = -\frac{a}{m} z - \frac{c}{m} x + \frac{\sigma}{m} x^{5/2} \sin t; \quad x(0) = 2; \quad z(0) = 1;$$

Полученная запись соответствует стандартной форме системы уравнений в которой функции f , g имеют вид

$$f(t, x, z) = z; \quad g(t, x, z) = -\frac{a}{m} z - \frac{c}{m} x + \frac{\sigma}{m} x^{5/2} \sin t.$$

4. Решение ОДУ высших порядков. Аналогичная процедура позволяет привести уравнение произвольного N -го порядка к системе N уравнений 1-го порядка. Например, задача Коши с уравнением N -го порядка

$$\frac{d^N x}{dt^N} = f\left(t, x, \frac{dx}{dt}, \frac{d^2 x}{dt^2}, \dots, \frac{d^{N-1} x}{dt^{N-1}}\right);$$

$$x(t_0) = a_0; \quad \frac{dx}{dt}(t_0) = a_1; \dots; \frac{d^{N-1} x}{dt^{N-1}}(t_0) = a_{N-1};$$

сводится заменой производных искомой функции на новые неизвестные функции

$$\frac{dx}{dt} = y_1(t); \quad \frac{d^2 x}{dt^2} = \frac{dy_1}{dt} = y_2(t); \quad \frac{d^3 x}{dt^3} = \frac{dy_2}{dt} = y_3(t); \quad \dots$$

к системе уравнений

$$\frac{dx}{dt} = y_1; \quad \frac{dy_1}{dt} = y_2; \quad \frac{dy_2}{dt} = y_3; \quad \dots \quad \frac{dy_{N-1}}{dt} = f(t, x, y_1, y_2, \dots, y_{N-1});$$

$$x(t_0) = a_0; \quad y_1(t_0) = a_1; \quad y_2(t_0) = a_2; \quad \dots \quad y_{N-1}(t_0) = a_{N-1};$$

Для полученной системы строится расчетная схема в соответствии с выбранным численным методом.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА. Преобразовать задачу Коши для ОДУ-2 к системе двух уравнений 1-го порядка. Составить программу для решения полученной системы с использованием численного метода Эйлера-Коши (четные варианты) и улучшенного метода Эйлера (нечетные варианты). Численную схему преобразовать для решения полученной системы уравнений 1-го порядка по аналогии с описанной выше. Предусмотреть в программе вычисление точного решения (проверить его подстановкой). Построить таблицу значений численного решения, точного решения и относительной погрешности численных результатов для заданного уравнения на указанном отрезке $[a, b]$ с шагом h . Построить график искомой функции и ее производной. Построить график погрешности решения в зависимости от значений аргумента и сопоставить его с поведением функции.

С помощью программы исследовать свойства численного метода при решении конкретной задачи Коши для ОДУ-2:

- найти максимальную погрешность на $[a, b]$ для значений $h=0.0125, 0.025, 0.05, 0.125, 0.25, 0.5$;
- найти порядок сходимости метода;

- установить, при каком значении h максимальная погрешность не превышает $0,01$.

Результаты исследования изложить в отчете. Сделать вывод.

№	Уравнение	Начальные условия		Точное решение
1	$Y''' + 2Y'' + 2Y' = 2e^{-x} \cos x$	$Y(0) = 1$ $Y'(0) = 0$	[0;3]	$e^{-x}(\cos x + \sin x + x \sin x)$
2	$X^2 Y''' - 4x Y'' + 6Y' = 2$	$Y(1) = 1, 1 + 1/3$ $Y'(1) = 2, 3$	[1;4]	$x^2 + 0, 1x^3 + 1/3$
3	$Y''' - Y = \sin x + \cos 2x$	$Y(0) = 1, 8$ $Y'(0) = -0, 5$	[0;4]	$e^x + e^{-x} - 0, 5 \sin x - 0, 2 \cos 2x$
4	$Y''' + 4Y = \cos 3x$	$Y(0) = 0, 8$ $Y'(0) = 2$	[0;6]	$\cos 2x + \sin 2x - 0, 2 \cos 3x$
5	$Y''' + Y'' - 6Y' = 3x^2 - x - 1$	$Y(0) = -0, 9$ $Y'(0) = 3, 2$	[0;2,5]	$0, 1e^{-2x} - e^{-3x} - 0, 5x^2$
6	$Y''' - Y = e^{2x}(x-1)$	$Y(0) = 11/9$ $Y'(0) = -11/9$	[0;3]	$2\operatorname{ch}x + e^{2x}(x/3 - 7/9)$
7	$Y''' + (1/x)Y'' - (1/x^2)Y' = 8x$	$Y(1) = 4$ $Y'(1) = 4$	[1;6]	$2x + 1/x + x^3$
8	$X^2 Y''' + 2, 5x Y'' - Y = 0$	$Y(1) = 2$ $Y'(1) = 3, 5$	[1;4]	$3x^{1/2} - 1/x^2$
9	$Y''' - 3y'' + 2y' = \cos 2x$	$Y(0) = 1, 95$ $Y'(0) = 2, 7$	[0;2]	$e^x + e^{2x} - (3\sin 2x + \cos 2x)/20$
10	$Y''' + 4Y'' + 4Y' = 2x - 3$	$Y(0) = -1/4$ $Y'(0) = -1/2$	[0;3]	$(1+x)e^{-2x} + x/2 - 5/4$
11	$Y''' + Y = 1/\cos x$	$Y(0) = 1$ $Y'(0) = 0$	[0;1.2]	$\cos x + x \sin x + \cos x * \ln(\cos x)$
12	$(1+x^2)Y''' + (Y')^2 + 1 = 0$	$Y(0) = 1$ $Y'(0) = 1$	[0;2]	$1 - x + 2\ln(1+x)$
13	$Y''' + 4Y' = \sin x + \sin 2x$	$Y(0) = 1$ $Y'(0) = -23/12$		$\cos 2x - \sin 2x + (1/3)\sin x - (1/4)x \cos 2x$
14	$Y''' - 0, 5Y'' - 0, 5Y' = 3e^{x/2}$	$Y(0) = -4$ $Y'(0) = -2, 5$	[0;4]	$e^x + e^{-x/2} - 6e^{x/2}$
15	$4xY''' + 2Y'' + Y = 0$	$Y(1) = 1, 38177$ $Y'(1) = -0, 15058$	[1;11]	$\sin(x^{1/2}) + \cos(x^{1/2})$

СОДЕРЖАНИЕ ОТЧЕТА:

1. Постановка задачи (конкретный вариант).
2. Расчетные формулы для решения задачи.
3. Текст программы.
4. Таблица результатов вычислений.
5. График решения (использовать процедуру Grafik1 из модуля Grafik_D).
6. Результаты исследования и выводы.

ТЕМА 11. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Цель работы: программная реализация и практическое применение метода Гаусса и его модификаций для решения систем линейных уравнений.

1. О методах решения систем линейных уравнений. Решение систем линейных алгебраических уравнений:

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2; \\ &\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n;\end{aligned}$$

является одной из задач, наиболее часто встречающихся на практике как окончательный или как промежуточный этап расчета.

Наиболее распространенными среди прямых методов являются метод исключения Гаусса и его модификации. Для упрощения записи отдельных действий метода Гаусса обычно используют т.н. расширенную матрицу, i -я строка которой содержит коэффициенты и правые части i -го уравнения ($a_{n,n+1} = b_n$).

$$R = \left[\begin{array}{cccccc} a_{11} & a_{12} & \dots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \dots & a_{2n} & a_{2,n+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nn} & a_{n,n+1} \end{array} \right]$$

Тогда действия над элементами строки расширенной матрицы соответствуют обработке отдельного уравнения в целом, с учетом правой и левой частей.

2. Алгоритм метода Гаусса. Рассмотрим применение метода Гаусса для системы уравнений

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1; \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2; \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3; \end{aligned} \quad R = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$$

где $a_{14}=b_1$; $a_{24}=b_2$; $a_{34}=b_3$.

Метод Гаусса основан на приведении расширенной матрицы системы к треугольному виду, в котором все поддиагональные элементы равны 0, а диагональные равны 1 (т.н. схема единственного деления). Это соответствует последовательному исключению неизвестных из уравнений системы.

$$\begin{aligned} x_1 + a_{12}'x_2 + a_{13}'x_3 &= b_1'; \\ x_2 + a_{23}'x_3 &= b_2'; \\ x_3 &= b_3'; \end{aligned} \quad R = \begin{bmatrix} 1 & a_{12}' & a_{13}' & a_{14}' \\ 0 & 1 & a_{23}' & a_{24}' \\ 0 & 0 & 1 & a_{34}' \end{bmatrix}$$

Сначала первая строка нормируется (все элементы строки делятся на a_{11} , и на диагонали появляется 1), а затем с помощью первой строки получаются 0 в элементах первого столбца (при этом первая строка называется ведущей). Затем нормируется вторая строка, и формируются нулевые элементы во втором столбце (ведущая - вторая строка), и так далее. Этот процесс, называемый *прямым ходом метода Гаусса*.

Обратный ход метода Гаусса начинается с решения последнего уравнения

$$x_3 = a_{34}'.$$

Используя это значение, можно найти x_2 из второго уравнения, а затем x_1 из первого:

$$\begin{aligned} x_2 &= (a_{24}' - a_{23}'x_3); \\ x_1 &= (a_{14}' - a_{12}'x_2 - a_{13}'x_3). \end{aligned}$$

Аналогично строится вычислительный алгоритм для линейной системы с произвольным числом уравнений.

Заметим, что в процессе нормировки строк приходится выполнять операцию (единственного) деления на диагональные коэффициенты a_{11} , a_{22} и т. д. Поэтому они должны быть отличными от нуля; в противном случае соответственным образом переставить уравнения системы. Перестановка уравнений должна быть

предусмотрена в вычислительном алгоритме при его реализации на ЭВМ.

Алгоритм простого метода Гаусса можно отобразить следующим образом:

Запрос и ввод количества уравнений N

Заполнение массива $\{a_{11} \dots a_{n,n+1}\}$

Для $k=1$ до N выполнить (----прямой ход)

```

| norm=akk
| Для j=k до N+1 выполнить akj=akj/norm
| Для i=k+1 до N выполнить
|   | z=aik
|   | Для j=k до N+1 выполнить aij=aij-akj*z
|   | Вывод элементов матрицы на экран

```

$x_N=a_{N,N+1}$

(----обратный ход)

Для $i=N-1$ до 1 шаг (-1) выполнить

```

| s=0
| Для j=i+1 до N выполнить s=s+aij*xj
| xi= ai,N+1 - s

```

Вывод массива $\{x_i\}$

3. Метод Гаусса, схема с выбором главного элемента. Он состоит в том, что требование неравенства нулю диагональных элементов a_{kk} , на которые происходит деление в процессе нормирования строк, заменяется более жестким: из всех оставшихся в k -м столбце элементов нужно выбрать наибольший по модулю и переставить уравнения так, чтобы этот элемент оказался диагональным. Это позволяет уменьшить вычислительные погрешности при выполнении операции деления.

4. Метод Гаусса-Жордана. Если с помощью ведущей строки получать 0 не только в поддиагональных, но и в наддиагональных элементах расширенной матрицы, то обратный ход становится особенно простым: $x_i=a_{i,n+1}$.

5. Вычисление определителя матрицы. Непосредственное нахождение определителя требует большого объема вычислений. Вместе с тем легко вычисляется определитель треугольной матрицы: он равен произведению ее диагональных элементов. Для приведения матрицы к треугольному виду может быть использован *метод*

исключения, т. е. прямой ход метода Гаусса. В процессе исключения элементов величина определителя не меняется. В схеме единственного деления диагональные элементы $z=a_{kk}$ нормируются на 1. Поэтому их произведение будет соответствовать определителю матрицы. Знак определителя меняется на противоположный при перестановке его столбцов или строк. Следовательно, значение определителя после приведения матрицы A к треугольному виду вычисляется как произведение диагональных элементов преобразованной матрицы

$$\det A = \pm a_{11}' a_{22}' \dots a_{nn}'$$

Знак зависит от того, четной или нечетной была суммарная перестановка строк (или столбцов) матрицы при ее приведении к треугольному виду (для получения ненулевого или максимального по модулю ведущего элемента на каждом этапе исключения). Благодаря методу исключения можно вычислять определители до 100-го более порядков.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА 1. Составить программу, в которой выполняются следующие действия:

а) описан двумерный массив вещественных чисел для хранения элементов расширенной матрицы системы уравнений (до 10 уравнений);

б) организован ввод с клавиатуры числа уравнений и элементов расширенной матрицы по строкам, с соответствующими пояснениями;

в) реализован прямой и обратный ход метода Гаусса и вычисление решения системы линейных уравнений;

г) выполняется численная проверка решения с выдачей сообщения о результате проверки;

д) организован вывод (с соответствующими пояснениями) исходной информации и результатов решения системы уравнений на экран в формате 3 цифры после запятой;

е) организован вывод (с соответствующими пояснениями) исходной информации и результатов решения в текстовый файл.

ЗАДАЧА 2. Для тестирования составленной программы решить все приведенные тестовые примеры и сравнить полученные ответы с

известными. В случае отличия в ответах проверить текст программы и выявить ошибки.

- а) $2x_1+7x_2=-22$; б) $2x_1+x_2-x_3=5$; в) $x_1+3x_2+2x_3-x_4=9$;
 $9x_1-5x_2=47$; $x_1-x_2+x_3=10$; $2x_1-6x_2-x_3+2x_4=-5$;
 (Ответ: 3 -4) $-2x_1+3x_2+x_3=3$; $3x_1-2x_2+x_3-x_4=-2$;
 (Ответ: 5 2 7) $x_1+5x_2-3x_3+x_4=6$;
 (Ответ: 1 2 3 4)

ЗАДАЧА 3. Модернизировать составленную программу таким образом, чтобы численное решение системы выполнялось в процедуре, входными параметрами которой являются массив расширенной матрицы и число уравнений, а выходной параметр – одномерный массив, содержащий вычисленные значения неизвестных.

Проверить работу процедуры на тестовых примерах.

ЗАДАЧА 4. Реализовать в процедуре метод Гаусса-Жордана. Проверить работу процедуры на тестовых примерах.

ЗАДАЧА 5. Реализовать в процедуре выбор главного элемента. Проверить работу процедуры на тестовых примерах.

ЗАДАЧА 6. Перенести процедуру в собственный модуль. Проверить работу процедуры на тестовых примерах при подключении ее из модуля.

ЗАДАЧА 7. С помощью составленной и протестированной программы решить систему трех линейных уравнений. Результат вывести с точностью 0,001:

№ 1.
 $2,23x_1-0,71x_2+0,63x_3=1,28$;
 $-0,71x_1+1,45x_2-1,34x_3=0,64$;
 $0,63x_1-1,34x_2+0,77x_3=-0,87$.

№ 3.
 $0,78x_1+1,08x_2-1,35x_3=0,57$;
 $1,08x_1-1,28x_2+0,37x_3=1,27$;
 $-1,35x_1+0,37x_2+2,86x_3=0,47$.

№ 5.
 $2,74x_1-1,18x_2+1,23x_3=0,16$,

№ 2.
 $1,63x_1+1,27x_2-0,84x_3=1,51$
 $1,27x_1+0,65x_2+1,27x_3=-0,63$
 $-0,84x_1+1,27x_2-1,21x_3=2,15$

№ 4.
 $0,83x_1+2,18x_2-1,73x_3=0,28$
 $2,18x_1-1,41x_2+1,03x_3=-1,18$
 $-1,73x_1+1,03x_2+2,27x_3=0,72$

№ 6.
 $1,35x_1-0,72x_2+1,38x_3=0,88$

$$\begin{aligned} -1, 18x_1+1, 71x_2-0, 52x_3=1, 81, \\ 1, 23x_1-0, 52x_2+0, 62x_3=-1, 25. \end{aligned}$$

№ 7.

$$\begin{aligned} 1, 48x_1+0, 75x_2-1, 23x_3=0, 83; \\ 0, 75x_1-0, 96x_2+1, 64x_3=-1, 12; \\ -1, 23x_1+1, 64x_2-0, 55x_3=0, 47. \end{aligned}$$

№ 9.

$$\begin{aligned} 0, 63x_1-1, 72x_2+3, 37x_3=-0, 75, \\ -1, 72x_1-2, 27x_2+1, 62x_3=1, 27; \\ 3, 27x_1+1, 62x_2-0, 43x_3=2, 74. \end{aligned}$$

№ 11.

$$\begin{aligned} 2, 32x_1+1, 17x_2-0, 28x_3=1, 43; \\ 1, 17x_1-1, 43x_2+0, 88x_3=-0, 47; \\ -0, 28x_1+0, 88x_2-1, 45x_3=1, 09. \end{aligned}$$

№ 13.

$$\begin{aligned} 1, 18x_1+2, 32x_2-0, 67x_3=1, 83; \\ 2, 32x_1+1, 87x_2+1, 35x_3=-0, 73 \\ -0, 67x_1+1, 35x_2-0, 88x_3=0, 68. \end{aligned}$$

№ 15.

$$\begin{aligned} 1, 17x_1-0, 65x_2+1, 54x_3=-1, 43 \\ -0, 65x_1+1, 16x_2-1, 73x_3=0, 68; \\ 1, 54x_1-1, 73x_2+2, 15x_3=1, 87. \end{aligned}$$

$$\begin{aligned} -0, 72x_1+1, 45x_2-2, 18x_3=1, 72 \\ 1, 38x_1-2, 18x_2+0, 93x_3=-0, 72 \end{aligned}$$

№ 8.

$$\begin{aligned} 2, 16x_1-3, 18x_2+1, 26x_3=1, 83 \\ -3, 18x_1+0, 63x_2-2, 73x_3=0, 54 \\ 1, 26x_1-2, 73x_2+3, 15x_3=1, 72 \end{aligned}$$

№ 10.

$$\begin{aligned} 1, 36x_1+0, 92x_2-1, 87x_3=2, 15 \\ 0, 92x_1-2, 24x_2+0, 77x_3=-2, 06 \\ -1, 87x_1+0, 77x_2-1, 16x_3=0, 17 \end{aligned}$$

№ 12.

$$\begin{aligned} 0, 75x_1-1, 24x_2+1, 56x_3=0, 49 \\ -1, 24x_1+0, 18x_2-1, 72x_3=-0, 57 \\ 1, 56x_1-1, 72x_2+0, 79x_3=1, 03 \end{aligned}$$

№ 14.

$$\begin{aligned} 0, 78x_1+1, 13x_2+1, 87x_3=0, 83 \\ 1, 13x_1-0, 68x_2+2, 16x_3=-0, 27 \\ 1, 87x_1+2, 16x_2-2, 63x_3=1, 37 \end{aligned}$$

№ 16.

$$\begin{aligned} 0, 87x_1+1, 35x_2-0, 44x_3=1, 51 \\ 1, 35x_1-1, 22x_2+2, 32x_3=0, 71 \\ -0, 44x_1+2, 32x_2-3, 73x_3=0, 53 \end{aligned}$$

ЗАДАЧА 8. С помощью составленной и протестированной программы решить систему четырех линейных уравнений. Результат вывести с точностью 0,0001:

№ 1

$$\begin{aligned} 1, 42x_1+2, 34x_2-0, 88x_3+0, 53x_4 &= 0, 72; \\ 0, 71x_1-1, 15x_2+0, 53x_3-0, 67x_4 &=-0, 18; \\ 0, 55x_1-0, 93x_2-1, 42x_3+1, 32x_4 &= 0, 68; \\ 0, 44x_1-0, 25x_2+1, 92x_3-1, 08x_4 &= 0, 43. \end{aligned}$$

№ 2

$$\begin{aligned} 3, 51x_1+0, 17x_2+3, 75x_3-0, 28x_4 &= 0, 75; \\ 4, 52x_1+2, 11x_2-0, 11x_3-0, 12x_4 &= 1, 11; \\ -2, 11x_1+3, 17x_2+0, 12x_3-0, 15x_4 &= 0, 21; \\ 3, 17x_1+1, 81x_2-3, 17x_3+0, 22x_4 &= 0, 05. \end{aligned}$$

№ 3

$$\begin{aligned} 0, 17x_1+0, 75x_2-0, 18x_3+0, 21x_4 &= 0, 11; \\ 0, 75x_1+0, 13x_2+0, 11x_3+1, 00x_4 &= 2, 00; \\ -0, 33x_1+0, 11x_2+3, 01x_3-2, 01x_4 &= 0, 11; \\ 0, 11x_1+1, 12x_2+1, 11x_3-1, 31x_4 &= 0, 13. \end{aligned}$$

№ 4

$$\begin{aligned}
 -1,00x1+0,13x2-2,00x3-0,14x4 &= 0,15; \\
 0,75x1+0,18x2-0,21x3-0,77x4 &= 0,11; \\
 0,28x1-0,17x2+0,39x3+0,48x4 &= 0,12; \\
 1,00x1+3,14x2-0,21x3-1,00x4 &=-0,11.
 \end{aligned}$$

№ 5

$$\begin{aligned}
 3,01x1-0,14x2+1,00x3-0,15x4 &= 1,00; \\
 -1,75x1+1,11x2+0,13x3-0,75x4 &= 0,13; \\
 0,17x1-2,11x2+0,71x3-1,71x4 &= 1,00; \\
 0,21x1+0,21x2+0,35x3+0,33x4 &= 0,17.
 \end{aligned}$$

№ 6

$$\begin{aligned}
 1,15x1+0,62x2-0,83x3+0,92x4 &= 2,15; \\
 0,82x1-0,54x2+0,43x3-0,25x4 &= 0,62; \\
 0,24x1+1,15x2-0,33x3+1,42x4 &=-0,62; \\
 0,73x1-0,81x2+1,27x3-0,67x4 &= 0,88.
 \end{aligned}$$

№ 7

$$\begin{aligned}
 2,20x1-3,17x2+1,24x3-0,87x4 &= 0,46; \\
 1,50x1+2,11x2-0,45x3+1,44x4 &= 1,50; \\
 0,86x1-1,44x2+0,62x3+0,28x4 &=-0,12; \\
 0,48x1+1,25x2-0,63x3-0,97x4 &= 0,35.
 \end{aligned}$$

№ 8

$$\begin{aligned}
 0,64x1+0,72x2-0,83x3+4,20x4 &= 2,23; \\
 0,58x1-0,83x2+1,43x3-0,62x4 &= 1,71; \\
 0,86x1+0,77x2-1,83x3+0,88x4 &=-0,54; \\
 1,32x1-0,52x2-0,65x3+1,22x4 &= 0,65.
 \end{aligned}$$

№ 9

$$\begin{aligned}
 1,42x1+0,32x2-0,42x3+0,85x4 &= 1,32; \\
 0,63x1-0,43x2+1,27x3-0,58x4 &=-0,44; \\
 0,84x1-2,23x2-0,52x3+0,47x4 &= 0,64; \\
 0,27x1+1,37x2+0,64x3-1,27x4 &= 0,85.
 \end{aligned}$$

№ 10

$$\begin{aligned}
 0,73x1+1,24x2-0,38x3-1,43x4 &= 0,58; \\
 1,07x1-0,77x2+1,25x3+0,66x4 &=-0,66; \\
 1,56x1+0,66x2+1,44x3-0,87x4 &= 1,24; \\
 0,75x1+1,22x2-0,83x3+0,37x4 &= 0,92.
 \end{aligned}$$

№ 11

$$\begin{aligned}
 1,32x1-0,83x2-0,44x3+0,62x4 &= 0,68; \\
 0,83x1+0,42x2-0,56x3+0,77x4 &= 1,24; \\
 0,58x1-0,37x2+1,24x3-0,62x4 &= 0,87; \\
 0,35x1+0,66x2-1,38x3-0,93x4 &=-1,08.
 \end{aligned}$$

№ 12

$$\begin{aligned}
 0,11x1-0,17x2+0,72x3-0,34x4 &= 0,17; \\
 0,81x1+0,12x2-0,91x3+0,17x4 &= 1,00; \\
 0,17x1-0,18x2+1,00x3+0,23x4 &= 0,21; \\
 0,13x1+0,17x2-0,99x3+0,35x4 &= 2,71.
 \end{aligned}$$

№ 13

$$\begin{aligned}0,18 \times 1 + 2,11 \times 2 + 0,13 \times 3 - 0,22 \times 4 &= 0,22; \\0,33 \times 1 - 0,22 \times 2 - 1,00 \times 3 + 0,17 \times 4 &= 0,11; \\-1,00 \times 1 + 0,11 \times 2 + 2,00 \times 3 - 0,45 \times 4 &= 1,00; \\7,00 \times 1 - 0,17 \times 2 - 0,22 \times 3 + 0,33 \times 4 &= 0,21.\end{aligned}$$

№ 14

$$\begin{aligned}2,00 \times 1 + 0,05 \times 2 - 3,01 \times 3 - 0,11 \times 4 &= 0,21; \\1,00 \times 1 - 2,00 \times 2 + 3,02 \times 3 + 0,05 \times 4 &= 0,18; \\0,17 \times 1 + 0,99 \times 2 - 2,00 \times 3 - 0,17 \times 4 &= 0,17; \\0,33 \times 1 - 0,07 \times 2 + 0,33 \times 3 + 2,00 \times 4 &= 0,17.\end{aligned}$$

№ 15

$$\begin{aligned}0,17 \times 1 - 0,13 \times 2 - 0,11 \times 3 - 0,12 \times 4 &= 0,22; \\1,00 \times 1 - 1,00 \times 2 - 0,13 \times 3 + 0,13 \times 4 &= 0,11; \\0,35 \times 1 + 0,33 \times 2 + 0,12 \times 3 + 0,13 \times 4 &= 0,12; \\0,13 \times 1 + 0,11 \times 2 - 0,13 \times 3 - 0,11 \times 4 &= 1,00.\end{aligned}$$

ТЕМА 12. ОБРАБОТКА РЕЗУЛЬТАТОВ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА ПО МЕТОДУ НАИМЕНЬШИХ КВАДРАТОВ

Цель работы: Изучение методов аппроксимации результатов эксперимента или компьютерного расчета. Отработка навыков исследования физических систем на основе аппроксимации данных.

1. Постановка задачи аппроксимации. Умение решать системы линейных уравнений открывает возможности изучения и применения новых численных методов, в которых решение систем является необходимым этапом. В данной лекции мы рассмотрим методы аппроксимации данных, полученных на эксперименте или в ходе вычислений.

Пусть зависимость между переменными величинами x и y выражается в виде экспериментально полученной таблицы

X: $x_1 \ x_2 \ x_3 \ \dots \ x_{n-1} \ x_n$

Y: $y_1 \ y_2 \ y_3 \ \dots \ y_{n-1} \ y_n$

Если аналитическое выражение зависимости $y=F(x)$ неизвестно или очень сложно, то возникает *задача аппроксимации (приближения)*, т.е. задача нахождения такой достаточно простой функции $y=f(x)$, значения которой на отрезке $[a;b]$, содержащем точки $x_i, i=1..n$, как можно меньше отличаются от значений искомой функции $F(x)$.

Принцип решения: так как точки получены с погрешностью, то кривая может проходить не через все точки.

В данном случае мы имеем задачу аппроксимации поиска функции, проходящей как можно ближе к заданным точкам.

Это требование близости можно записать математически как поиск минимума квадрата отклонений $\delta_i = F(x_i) - y_i$

$$\sum_{i=1}^N \delta_i^2 = \sum_{i=1}^N (F(x_i) - y_i)^2 = \min$$

Метод, основанный на этом требовании, называется методом наименьших квадратов.

Так как явный вид функции неизвестен, то его выбирают в нескольких вариантах и используют далее тот вариант, для которого S меньше.

При выборе функции предполагают определенный класс функций с соответствующими параметрами.

2. Аппроксимация полиномом. В качестве аппроксимирующей функции обычно выбирают наиболее простой и удобный тип зависимости - полином невысокого (обычно не более 5-го) порядка

$$F(x) = P_m(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m$$

Тогда суммарное квадратичное отклонение S будет зависеть от выбора параметров полинома и запишется в виде

$$\begin{aligned} S(a_0, a_1, a_2, \dots, a_m) &= \sum_i (P_m(x_i) - y_i)^2 = \\ &= \sum_{i=1}^N (a_0 + a_1x_i + a_2x_i^2 + \dots + a_mx_i^m - y_i)^2 \end{aligned}$$

Условие минимума суммы

$$\min S: \frac{\partial S}{\partial a_0} = 0; \quad \frac{\partial S}{\partial a_1} = 0; \quad \dots \quad \frac{\partial S}{\partial a_m} = 0;$$

Используя явный вид S, можно вычислить частные производные. В результате приходим к системе линейных уравнений относительно неизвестных параметров a_0, \dots, a_m .

$$\frac{\partial S}{\partial a_0} = 2 \sum_{k=1}^n (a_0 + a_1 x_k + a_2 x_k^2 + a_3 x_k^3 + \dots + a_m x_k^m - y_k) \cdot 1 = 0;$$

$$\frac{\partial S}{\partial a_1} = 2 \sum_{k=1}^n (a_0 + a_1 x_k + a_2 x_k^2 + a_3 x_k^3 + \dots + a_m x_k^m - y_k) \cdot x_k = 0;$$

$$\frac{\partial S}{\partial a_2} = 2 \sum_{k=1}^n (a_0 + a_1 x_k + a_2 x_k^2 + a_3 x_k^3 + \dots + a_m x_k^m - y_k) \cdot x_k^2 = 0;$$

.....

$$\frac{\partial S}{\partial a_m} = 2 \sum_{k=1}^n (a_0 + a_1 x_k + a_2 x_k^2 + a_3 x_k^3 + \dots + a_m x_k^m - y_k) \cdot x_k^m = 0.$$

Сократив каждое уравнение системы на 2 и раскрыв скобки, получим

$$a_0 \sum_{k=1}^n 1 + a_1 \sum_{k=1}^n x_k + a_2 \sum_{k=1}^n x_k^2 + \dots + a_m \sum_{k=1}^n x_k^m = \sum_{k=1}^n y_k;$$

$$a_0 \sum_{k=1}^n x_k + a_1 \sum_{k=1}^n x_k^2 + a_2 \sum_{k=1}^n x_k^3 + \dots + a_m \sum_{k=1}^n x_k^{m+1} = \sum_{k=1}^n y_k x_k;$$

$$a_0 \sum_{k=1}^n x_k^2 + a_1 \sum_{k=1}^n x_k^3 + a_2 \sum_{k=1}^n x_k^4 + \dots + a_m \sum_{k=1}^n x_k^{m+2} = \sum_{k=1}^n y_k x_k^2;$$

.....

$$a_0 \sum_{k=1}^n x_k^m + a_1 \sum_{k=1}^n x_k^{m+1} + a_2 \sum_{k=1}^n x_k^{m+2} + \dots + a_m \sum_{k=1}^n x_k^{2m} = \sum_{k=1}^n y_k x_k^m;$$

Введем следующие обозначения:

$$t_0 = \frac{1}{n} \sum_{k=1}^n 1 = 1; \quad t_1 = \frac{1}{n} \sum_{k=1}^n x_k; \quad t_2 = \frac{1}{n} \sum_{k=1}^n x_k^2; \quad \dots$$

$$t_m = \frac{1}{n} \sum_{k=1}^n x_k^m;$$

$$A_{ij} = \frac{1}{N} \sum_{k=1}^N x_k^{i+j} = t_{i+j}; \quad b_i = \frac{1}{N} \sum_{k=1}^N y_k x_k^i; \quad i, j = 0, 1, \dots, m$$

Тогда систему (1) можно записать в виде

$$\begin{cases} A_{00}a_0 + A_{01}a_1 + A_{02}a_2 + A_{03}a_3 + \dots + A_{0m}a_m = b_0 \\ A_{10}a_0 + A_{11}a_1 + A_{12}a_2 + A_{13}a_3 + \dots + A_{1m}a_m = b_1 \\ A_{20}a_0 + A_{21}a_1 + A_{22}a_2 + A_{23}a_3 + \dots + A_{2m}a_m = b_2 \\ \dots \\ A_{m0}a_0 + A_{m1}a_1 + A_{m2}a_2 + A_{m3}a_3 + \dots + A_{mm}a_m = b_m \end{cases}$$

В матричной форме эта система принимает вид

$$A\vec{a} = \vec{b}$$

Система (8) является линейной относительно параметров $a_0, a_1, a_2, \dots, a_m$, и определитель этой системы отличен от нуля, т.е., система всегда имеет единственное решение. Матрица системы симметрическая и невырожденная.

Таким образом, для получения коэффициентов аппроксимирующего полинома m -го порядка необходимо сформировать и решить систему $(m+1)$ линейных уравнений. Конечно, для решения полученных систем удобно применить изученный ранее метод Гаусса.

3. Выбор степени аппроксимирующего многочлена. Для выбора степени аппроксимирующего многочлена можно поступить следующим образом:

а) построить график-диаграмму экспериментальных точек (x_i, y_i) , $i=1..n$ и оценить визуально, графику какой функции (прямой, параболы второго порядка, параболы третьего порядка и т.д.) приблизительно соответствует расположение точек;

б) выбрать некоторое m , найти из условия (2) соответствующие коэффициенты α_i , вычислить полученное при этом среднее квадратическое отклонение

$$\delta = \sqrt{\frac{1}{n} \sum_{i=1}^n [P_m(x_i) - y_i]^2} \quad (7)$$

и сравнить его с известной погрешностью эксперимента ε . Если $\delta \gg \varepsilon$, т.е. математическая погрешность аппроксимации много больше физической погрешности эксперимента, значит степень многочлена недостаточно велика для описания функциональной зависимости $y=f(x)$, и надо увеличить m . Если $\delta < \varepsilon$, то старшие коэффициенты аппроксимации физически недостоверны, и надо уменьшить m . Если $\delta \approx \varepsilon$, то число коэффициентов оптимально. Обычно расчет начинают с $m=1$, когда заведомо $\delta \gg \varepsilon$, и увеличивают число коэффициентов до тех пор, пока не выполняется условие $\delta \approx \varepsilon$.

3. Линейная аппроксимация. Простейший случай – линейная аппроксимация. Тем не менее, он часто используется для описания данных, так как линейно пропорциональная зависимость реализуется между многими физическими величинами.

Запишем общие формулы (1)-(3) для случая $m=1$.

$$P_1(x) = a_0 + a_1 x$$

Система уравнений

$$\begin{cases} A_{00}a_0 + A_{01}a_1 = b_0 \\ A_{10}a_0 + A_{11}a_1 = b_1 \end{cases}$$

где коэффициенты имеют вид

$$\begin{cases} a_0 + t_1 a_1 = b_0 \\ t_1 a_0 + t_2 a_1 = b_1 \end{cases}$$

$$t_1 = \frac{1}{N} \sum_{k=1}^N x_k; \quad t_2 = \frac{1}{N} \sum_{k=1}^N x_k^2; \quad b_0 = \frac{1}{N} \sum_{k=1}^N y_k; \quad b_1 = \frac{1}{N} \sum_{k=1}^N y_k x_k$$

Систему двух уравнений несложно решить методом Крамера.

$$\Delta = \begin{vmatrix} 1 & t_1 \\ t_1 & t_2 \end{vmatrix} = t_2 - t_1^2 \quad \Delta_0 = \begin{vmatrix} b_0 & t_1 \\ b_1 & t_2 \end{vmatrix} = b_0 t_2 - b_1 t_1 \quad \Delta_1 = \begin{vmatrix} 1 & b_0 \\ t_1 & b_1 \end{vmatrix} = b_1 - b_0 t_1$$

$$a_0 = \frac{\Delta_0}{\Delta} = \frac{b_0 t_2 - b_1 t_1}{t_2 - t_1^2}; \quad a_1 = \frac{\Delta_1}{\Delta} = \frac{b_1 - b_0 t_1}{t_2 - t_1^2};$$

4. Квадратичная (параболическая) аппроксимация. В этом случае аппроксимирующий полином является параболой

$$F(x) \approx P_2(x) = a_0 + a_1 x + a_2 x^2$$

Для нахождения коэффициентов полинома получаем систему уравнений

$$\begin{cases} a_0 + t_1 a_1 + t_2 a_2 = b_0 \\ t_1 a_0 + t_2 a_1 + t_3 a_2 = b_1 \\ t_2 a_0 + t_3 a_1 + t_4 a_2 = b_2 \end{cases}$$

где коэффициенты системы имеют вид

$$t_1 = \frac{1}{N} \sum_{k=1}^N x_k; \quad t_2 = \frac{1}{N} \sum_{k=1}^N x_k^2; \quad t_3 = \frac{1}{N} \sum_{k=1}^N x_k^3; \quad t_4 = \frac{1}{N} \sum_{k=1}^N x_k^4;$$

$$b_0 = \frac{1}{N} \sum_{k=1}^N y_k; \quad b_1 = \frac{1}{N} \sum_{k=1}^N y_k x_k; \quad b_2 = \frac{1}{N} \sum_{k=1}^N y_k x_k^2;$$

Вычисляя соответствующие суммы, формируем расширенную матрицу системы и применяем метод Гаусса для решения системы уравнений.

$$R = \begin{bmatrix} 1 & t_1 & t_2 & b_0 \\ t_1 & t_2 & t_3 & b_1 \\ t_2 & t_3 & t_4 & b_2 \end{bmatrix}$$

5. Аппроксимация нелинейной двухпараметрической зависимостью. Если экспериментальные точки (x_i, y_i) располагаются вдоль некоторой линии, сходной по форме, например, с графиком гиперболической, показательной, логарифмической или других неполиномиальных функций, то для аппроксимации данных можно использовать и нелинейные функции. Это особенно просто сделать, если нелинейная функция имеет два параметра. С помощью замены переменных такая функция может быть преобразована в линейную зависимость, применение которой подробно описано в п.3.1. Этот

способ очень эффективен и часто используется при получении эмпирических формул.

Например, при необходимости использовать зависимость вида

$$F(x) = ax^2 + b$$

перейдем к новой переменной

$$t = x^2; \quad z = F$$

Тогда в переменных (z, y) имеем дело с линейной зависимостью

$$z(t) = at + b$$

где теперь используются данные

$$t_i = x_i^2; \quad z_i = y_i$$

Обработывая таблицу данных $\{t_i, z_i\}$ методом п.3, получаем значения параметров a, b , которые входят и в выражение (3). Аналогично, для зависимости

$$F(x) = px^k$$

преобразование заключается в логарифмировании обеих частей

$$\ln F = k \ln x + \ln p$$

Вводя новые переменные

$$t = \ln x; \quad z = \ln F; \quad b = \ln p$$

получаем линейную связь

$$z = kt + b$$

для точек данных

$$t_i = \ln x_i; \quad z_i = \ln y_i$$

для которой несложно найти параметры k, b , и далее $p = e^b$.

Аналогичные действия можно реализовать и в других случаях (см. Таблицу 1).

Таблица 1. Линеаризация функции

Вид функциональной зависимости	Формулы преобразования	Линейная зависимость
--------------------------------	------------------------	----------------------

Гиперболическая	$y = a/x + b \quad t = 1/x;$	$z = y \quad z = at + b$
-----------------	------------------------------	--------------------------

Экспоненциальная	$a) y = ae^{kx} + b$	$t = e^{kx};$	$z = y$	$z = at + b$
	$b) y = be^{k/x}$	$t = 1/x;$	$z = \ln y$	$z = kt + p$
Логарифмическая	$y = k \log_b x + c$		$t = \log_b x$	$z = y$
Степенная	$y = px^a$	$t = \ln x;$	$z = \ln y;$	$z = at + b$
			$b = \ln p;$	

Элементы программной реализации вычислений. На лабораторных занятиях необходимо использовать разработанную ранее процедуру решения системы линейных уравнений.

Для хранения данных в программе должны быть описаны массивы X[1..100], Y[1..100]. Количество реально используемых точек следует записать в переменной NT.

Данные, хранящиеся в файле, необходимо занести в массивы. В файле данные хранятся в формате «координаты каждой точки – в отдельной строке»:

```
X1  Y1
X2  Y2
X3  Y3
. . .
```

Соответствующий фрагмент программы может быть таким

```
Var FT:text; NT:integer;
Begin
. . .
Assign(FT,'filename.txt');
Reset(FT);
NT:=0;
While NOT EOF(FT) do begin
  Readln(FT,x[NT], Y[NT]);
  NT:=NT+1;
End;
```

Для вычисления целой степени числа удобно написать отдельную подпрограмму-функцию Stepen(x,m). Для заполнения элементов расширенной матрицы подсчитываются суммы

$$A_{ij} = \frac{1}{N} \sum_{k=1}^N x_k^{i+j} = t_{i+j}; \quad b_i = \frac{1}{N} \sum_{k=1}^N y_k x_k^i;$$

$$i, j = 0, 1, \dots, m$$

. . .


```

S:=0;
For k:=1 to NT do S:=S+Stepen(x[k],i+j);
S:=S/NT;
A[i,j]:=S; A[j,i]:=S;
. . .

```

После расчета коэффициентов аппроксимации необходимо вычислить реальное среднеквадратичное отклонение

$$S = \sqrt{\frac{1}{N} \sum_{k=1}^N (F(x_i) - y_i)^2}$$

При использовании нескольких способов аппроксимации выбрать наилучший по значениям S.

Вопросы для самоконтроля

1. Общая схема метода наименьших квадратов.
2. Аппроксимация линейной зависимостью.
3. Аппроксимация квадратичной зависимостью.
4. Аппроксимация полиномом N-го порядка.
5. Нелинейная аппроксимация данных.

СОДЕРЖАНИЕ ЗАДАНИЯ

ЗАДАЧА. В текстовом файле с именем fileNN.dat, где NN-номер варианта (01, 02, и т.д.) хранятся результаты измерений указанной в условии варианта физической величины в формате:

аргумент1	функция1
аргумент2	функция2
...	...

Составить программу, в которой выполняются следующие действия:

- а) чтение данных из текстового файла в массивы;
- б) аппроксимация данных линейной зависимостью (расчет параметров линейной зависимости выполнить по конечным формулам) и расчет величины среднеквадратичного отклонения δ_1 ;
- в) аппроксимация данных квадратичной зависимостью (расчет параметров квадратичной зависимости выполнить с помощью

процедуры решения системы линейных уравнений) и расчет величины среднеквадратичного отклонения δ_2 ;

г) аппроксимация данных кубической зависимостью (расчет параметров кубической зависимости выполнить с помощью процедуры решения системы линейных уравнений) и расчет величины среднеквадратичного отклонения δ_3 ;

По величине суммарного среднеквадратичного отклонения выбрать лучший способ аппроксимации. Используя его, вычислить исследуемую величину в точках, составляющих 1/7 и 5/7 области определения аргумента.

По результатам аппроксимации определить требуемые параметры системы и погрешность их вычисления.

Варианты объектов измерений:

1. Спектральная характеристика фотоумножителя. Зависимость относительной чувствительности $N(\%)$ серебряно-цезиевого фотокатода от длины волны излучения $l(\text{нм})$.

2. Зависимость давления $P(\text{мм рт. ст.})$ насыщенного водяного пара при различных температурах $t(^{\circ}\text{C})$.

3. Зависимость давления $P(\text{мм рт. ст.})$ атмосферы от высоты $h(\text{км})$ над Землей.

4. Зависимость энергии связи $E(\text{МэВ})$ ядер от массового числа A для тяжелых ядер.

5. Зависимость плотности (г/м^3) насыщенного водяного пара от температуры $t(^{\circ}\text{C})$.

6. Зависимость удельной теплоты парообразования (испарения) $r(\text{кДж/кг})$ от температуры $T(^{\circ}\text{C})$.

7. Основная кривая намагничивания технически чистого железа. Зависимость индукции $B(\text{Тл})$ магнитного поля от напряженности $H(\text{А/м})$ намагничивающего поля.

8. Зависимость линейного коэффициента $\mu(1/\text{см})$ ослабления плотности потока фотонов в свинце от энергии фотонов $E(\text{МэВ})$.

9. Зависимость температуры $T(^{\circ}\text{C})$ атмосферы от высоты $h(\text{км})$ над Землей.

10. Зависимость коэффициента трения k от скорости $V(\text{м/с})$ движения стальной пластины по стальной поверхности.

11. Зависимость вязкости воды $\nu(\text{мкПа}\cdot\text{с})$ от температуры $t(^{\circ}\text{C})$.

12. Зависимость скорости звука в азоте $V(\text{м/с})$ от давления $P(\text{МПа})$ при температуре 20°C в диапазоне частот 200-500 кГц.

13. Зависимость скорости волн $V(\text{см/с})$ на поверхности глубокой воды от длины волны $\lambda(\text{см})$ (дисперсия поверхностных волн).
14. Зависимость диэлектрической проницаемости сегнетовой соли от напряженности поля $E(\text{кВ/см})$ при 20°C .
15. Зависимость плотности $\rho(\text{кг/м}^3)$ атмосферы от высоты $h(\text{км})$ над Землей.
16. Зависимость напряжения на кислотном аккумуляторе $U(\text{В})$ от времени зарядки нормальным током $Q/4 \text{ А}$ от времени $t(\text{час})$. Q - емкость аккумулятора, А/час .
17. Изотермический процесс для 1,2 молей кислорода. Зависимость давления $P(10^4 \text{ Па})$ от объема $V(10^{-3} \text{ м}^3)$. По результатам обработки данных определить температуру газа.
18. Зависимость удельной теплоемкости воды $C(\text{кДж/кг/град})$ от температуры $t(^\circ\text{C})$.
19. Зависимость пробега альфа-частицы в фотоэмульсии $R(\text{мкм})$ от энергии частицы $E(\text{МэВ})$.
20. Зависимость средней скорости $V(\text{м/с})$ движения молекул водорода от температуры $T(^\circ\text{K})$.

ЛИТЕРАТУРА

Алексеев В.Е. и др. Вычислительная техника и программирование. Практикум по программированию. Под ред. А.В.Петрова. – М., 1991. – 400 с.

Бородич Л.И. и др. Справочное пособие по приближенным методам решения задач высшей математики. – Мн.: Выш. шк., 1986. – 189 с.

Воробьева Г.Н., Данилова А.Н. Практикум по вычислительной математике. – М.: Высшая школа, 1990. — 208 с.

Дей Е.А. Методика использования функции Tools при разработке диалоговых программ в среде программирования Турбо-Паскаль // Информатизация образования. – 2001. – №2. – С. 47-54.

Епанешников А.М., Епанешников В.А. Программирование в среде Турбо Паскаль 7.0. – 3-е изд. – М.: Диалог-МИФИ, 1996. – 288 с.

Жаблон К., Симон Ж.-К. Применение ЭВМ для численного моделирования в физике. – М., 1983, – 237 с.

Калиткин Н. Н. Численные методы. – М.: Наука, 1978. – 512 с.

Копченова Н.В., Марон И.А. Вычислительная математика в примерах и задачах. – М.: Наука, 1972. – 367 с.

Мак-Кракен Д., Дорн У. Численные методы и программирование на фортране. – М. Мир, 1977. – 584 с.

Офицеров Д.В., Долгий А.Б., Старых В.А. Программирование на персональных ЭВМ. Практикум. – Мн, Вышэйшая школа, 1993.–256 с.

Пономарев К.К. Составление дифференциальных уравнений. Мн.: Вышэйшая школа, 1973. – 560 с.

Светозарова Г.И. и др. Практикум по программированию на алгоритмических языках. – М.: Наука, 1980. – 320 с.

Слободянюк А.И., Тимофеева И.А. Что значит решить физическую задачу на компьютере // Фізика. Праблемы выкладання. 1998, №1. с.52-69.

Турчак Л.И. Основы численных методов. М.: Наука, 1987. – 320 с.

Фаронов В.В. Турбо Паскаль 7.0. Начальный курс. Уч. пособие. Изд. 7-е. – М.: Изд-во «Нолидж», 2001. – 576 с.

Фаронов В.В. Турбо Паскаль 7.0. Практика программирования. Уч. пособие. Изд. 7-е. – М.: Изд-во «Нолидж», 2001. – 416 с.

Шуп Т. Решение инженерных задач на ЭВМ. – М., 1982. – 238 с.

Дей Евгений Александрович

**ПРОГРАММИРОВАНИЕ
И РЕШЕНИЕ ФИЗИЧЕСКИХ ЗАДАЧ
С ПРИМЕНЕНИЕМ ЧИСЛЕННЫХ МЕТОДОВ**

ПРАКТИЧЕСКОЕ ПОСОБИЕ
для студентов 2 курса
специальности 1-31 04 01 – 02 – «Физика»

В авторской редакции

Лицензия ЛИ № 02330/0133208 от 30.04.03.

Подписано в печать _____. Формат 60x84 1/16. Бумага
писчая №1. Гарнитура «Таймс». Усл.п.л. _____. Уч.-изд.л. _____.
Тираж 50 экз. Заказ № _____.