

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

И.С. Надеждин, А.Г. Горюнов

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО КУРСУ
«МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
ФИЗИЧЕСКИХ ПРОЦЕССОВ»**

*Рекомендовано в качестве учебно-методического пособия
Редакционно-издательским советом
Томского политехнического университета*

Издательство
Томского политехнического университета
2023

УДК 53:519.876(076.5)
ББК 22.3в631.0я7
Н17

Надеждин И.С.

Н17 Лабораторный практикум по курсу «Математическое моделирование физических процессов» : учебно-методическое пособие / И.С. Надеждин, А.Г. Горюнов ; Томский политехнический университет. – Томск : Изд-во Томского политехнического университета, 2023. – 111 с.

ISBN 978-5-4387-1146-9

Лабораторный практикум содержит теоретическую информацию об основных методах и подходах, применяемых при математическом моделировании физических процессов, а также ряд примеров программной реализации математического моделирования в пакете MATLAB.

Предназначен для студентов, обучающихся по направлениям 14.05.04 «Электроника и автоматика физических установок» и 14.03.02 «Ядерная физика и технологии», в рамках изучения дисциплин «Математическое моделирование» и «Математическое моделирование физических процессов».

УДК 53:519.876(076.5)
ББК 22.3в631.0я7

Рецензенты

Доктор технических наук, доцент
профессор кафедры компьютерных систем
в управлении и проектировании ТУСУРа
Т.В. Ганджа

Кандидат технических наук, доцент
заведующий кафедрой электроники и автоматики
физических установок СТИ НИЯУ «МИФИ»
К.А. Иванов

ISBN 978-5-4387-1146-9

© ФГАОУ ВО НИ ТПУ, 2023
© Надеждин И.С., Горюнов А.Г., 2023
© Оформление. Издательство Томского
политехнического университета, 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
Лабораторная работа № 1	
ИЗУЧЕНИЕ МЕТОДОВ ВОССТАНОВЛЕНИЯ ФУНКЦИИ	
ОДНОЙ ПЕРЕМЕННОЙ	6
Лабораторная работа № 2	
ИЗУЧЕНИЕ МЕТОДОВ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ	
И ДИФФЕРЕНЦИРОВАНИЯ ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ	24
Лабораторная работа № 3	
РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ,	
ЗАПИСАННЫХ В ФОРМЕ КОШИ, В ПАКЕТЕ MATLAB	53
Лабораторная работа № 4	
ОПРЕДЕЛЕНИЕ МЕТОДОМ КОНЕЧНЫХ РАЗНОСТЕЙ	
СТАЦИОНАРНОГО ПРОФИЛЯ ТЕМПЕРАТУРЫ В ПЛАСТИНЕ	69
Лабораторная работа № 5	
СТАТИСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ	
ПРИ СТОХАСТИЧЕСКОМ ИЗМЕНЕНИИ ИХ ПАРАМЕТРОВ	83
Лабораторная работа № 6	
РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ	
ФИЗИЧЕСКОГО ПРОЦЕССА	91
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	106
ПРИЛОЖЕНИЯ	107

ВВЕДЕНИЕ

Современное развитие науки и техники привело к широкому распространению математического моделирования как для решения научно-производственных задач, так и для реализации современного образовательного процесса. Стоит отметить, что в последние десятилетия данное направление оформилось в отдельную междисциплинарную область знаний с присущими ей подходами и методами исследования. Математическое моделирование имеет большое значение при решении научных и производственных задач в атомной, химической, нефтегазовой и других промышленности. В связи с этим наличие знаний в данной области современных инженеров является ключевым моментом.

В соответствии с учебным планом подготовки студентов по направлениям 14.05.04 «Электроника и автоматика физических установок» и 14.03.02 «Ядерная физика и технологии» дисциплины «Математическое моделирование» и «Математическое моделирование физических процессов» являются обязательными для изучения. Наряду с лекциями для закрепления теоретических знаний и отработки практических навыков предусмотрены лабораторные и практические занятия, а также курсовой проект.

Лабораторный практикум включает в себя изучение основных методов восстановления эмпирических зависимостей (интерполяция и аппроксимация), наиболее распространенных методов численного интегрирования и дифференцирования, а также рассматривает вопросы решения систем дифференциальных уравнений, записанных в форме Коши, расчет стационарного профиля температуры в пластине методом конечных разностей и статистическое моделирование процессов при случайном характере изменения параметров. Кроме того, в рамках шестой лабораторной работы студенты должны реализовать модель физической установки. Прежде чем приступить к лабораторной работе, студенту следует изучить основные теоретические сведения по теме работы и рассмотреть представленные примеры в пакете MATLAB. Лабораторные работы выполняются в пакете MATLAB в соответствии с вариантом индивидуального задания, которое выдается преподавателем.

По результатам лабораторной работы каждый студент должен оформить отчет и защитить его. Требования и рекомендации к оформлению отчета изложены в прил. А–В.

Отчет должен состоять из следующих частей:

- цель работы;
- основные теоретические сведения;
- описание каждого пункта проделанной работы и полученных результатов, а также сформулированные выводы по работе;
- ответы на контрольные вопросы.

Содержание данного лабораторного практикума предусматривает дальнейшее использование полученных знаний во всех специальных технических дисциплинах при последующем обучении, включая курсовое проектирование, а также выполнение учебно-исследовательских и выпускных квалификационных работ.

Лабораторная работа № 1

ИЗУЧЕНИЕ МЕТОДОВ ВОССТАНОВЛЕНИЯ ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ

Цель работы: изучение основных методов восстановления функции одной переменной с использованием специализированного программного обеспечения MATLAB.

Основные теоретические сведения

Интерполяция

Допустим, что в ходе экспериментальных исследований изменялась некая входная величина, например x ($x_0, x_1, x_2, \dots, x_n$). В результате были получены некоторые значения выходной величины y , которую можно описать функцией $y = f(x)$ в виде значений ($y_0, y_1, y_2, \dots, y_n$), представленных в табл. 1 и на рис. 1.

Таблица 1

Табличное представление экспериментальных данных

x_0	x_1	x_2	$\dots$	x_{n-1}	x_n
y_0	y_1	y_2	$\dots$	y_{n-1}	y_n

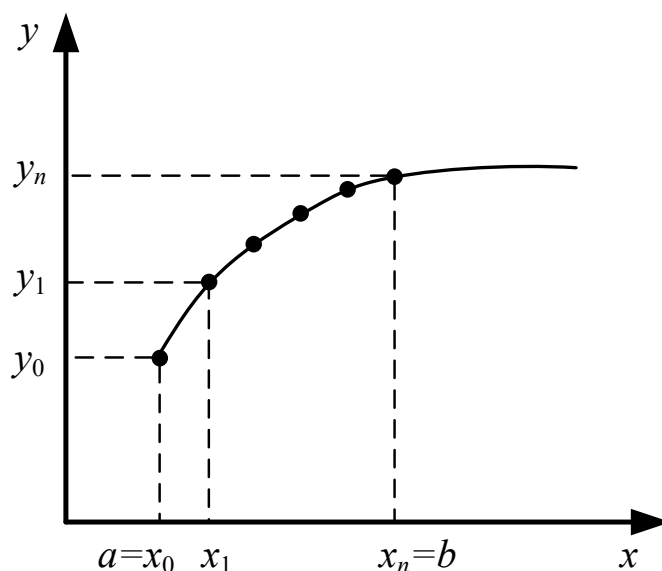


Рис. 1. Графическое представление экспериментальных данных

Интерполяция – способ определения промежуточных значений некой величины по имеющемуся дискретному набору известных значений этой величины.

Следовательно, задача интерполяции заключается в том, чтобы определить некую функцию g , которая бы удовлетворяла следующим условиям:

$$g(x_i) = y_i \quad (i = 0, 1, 2, \dots, n),$$

то есть график функции g будет проходить через заданные точки (x_i, y_i) , которые называются *узлами интерполяции*.

Интерполяционным многочленом степени n для заданных табличных данных (табл. 1) называется многочлен $P_n(x) = \sum_{k=0}^n a_k \cdot x^k$, если он удовлетворяет условиям:

$$P_n(x_i) = y_i \quad (i = 0, 1, 2, \dots, n).$$

Интерполяционный полином Лагранжа

Одна из форм записи интерполяционного многочлена – *многочлен Лагранжа*:

$$L_n(x) = \sum_{j=0}^n y_j \cdot l_{nj}(x), \quad (1)$$

где

$$l_{nj}(x) = \prod_{\substack{k=1 \\ k \neq j}}^n \frac{x - x_k}{x_j - x_k}.$$

В выражении (1) параметр $l_{nj}(x)$ представляет собой многочлен степени n , удовлетворяющий условию:

$$l_{nj}(x_i) = \begin{cases} 1 & \text{при } i = j; \\ 0 & \text{при } i \neq j. \end{cases}$$

Таким образом, степень многочлена L_n равна n и при $x = x_i$ в сумме обращаются в ноль все слагаемые, кроме слагаемого с номером $j = i$, которое соответствует y_i .

В инженерной практике наиболее часто используется интерполяция многочленами 1-й, 2-й и 3-й степени (линейная, квадратичная и кубическая интерполяции). Соответствующие формулы для записи многочлена Лагранжа 1-й и 2-й степени следующие:

$$\begin{aligned} L_1(x) &= y_0 \cdot \frac{x - x_1}{x_0 - x_1} + y_1 \cdot \frac{x - x_0}{x_1 - x_0}; \\ L_2(x) &= y_0 \cdot \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + \\ &+ y_1 \cdot \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \cdot \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \end{aligned}$$

Интерполяция сплайнами

Определение сплайна. Пусть отрезок $[a, b]$ разбит точками $a = x_0 < x_1 < \dots < x_n = b$ на n частичных отрезков $[x_{i-1}, x_i]$. Сплайном степени m называется функция $S_m(x)$, обладающая следующими свойствами:

- 1) функция $S_m(x)$ непрерывна на отрезке $[a, b]$ вместе со всеми своими производными $S_m^{(1)}(x), S_m^{(2)}(x), \dots, S_m^{(p)}(x)$ до некоторого порядка p ;
- 2) на каждом частичном отрезке $[x_{i-1}, x_i]$ функция $S_m(x)$ совпадает с некоторым алгебраическим многочленом $P_{m,i}(x)$ степени m .

Разность $(m - p)$ между степенью сплайна и наивысшим порядком непрерывной на отрезке $[a, b]$ производной называется *дефектом сплайна*.

Наибольшее распространение на практике получили сплайны $S_3(x)$ 3-й степени (кубические сплайны) с дефектом 1 или 2. Такие сплайны на каждом из частичных отрезков $[x_{i-1}, x_i]$ совпадают с кубическим многочленом

$$S_3(x) = P_{3,i}(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3$$

и имеют на отрезке $[a, b]$ по крайней мере одну непрерывную производную $S_3'(x)$.

Пусть функция $y = f(x)$ задана таблицей своих значений (табл. 1). Сплайн $S_m(x)$ называется интерполяционным, если $S_m(x_i) = y_i$ для всех $i = 0, 1, \dots, n$. Значение $s_i = S_m'(x_i)$ называется *наклоном сплайна* в точке x_i . На отрезке $[x_{i-1}, x_i]$ интерполяционный кубический сплайн однозначно определяется заданием значений $y_{i-1}, y_i, s_{i-1}, s_i$ и справедлива следующая формула:

$$\begin{aligned} S_3(x) = P_{3,i}(x) = & \frac{(x - x_i)^2 (2(x - x_{i-1}) + h_i)}{h_i^3} y_{i-1} + \\ & + \frac{(x - x_{i-1})^2 (2(x_i - x) + h_i)}{h_i^3} y_i + \frac{(x - x_i)^2 (x - x_{i-1})}{h_i^2} s_{i-1} + \\ & + \frac{(x - x_{i-1})^2 (x - x_i)}{h_i^2} s_i, \end{aligned} \quad (2)$$

где $h_i = x_i - x_{i-1}$.

Если в точках x_i известны значения производной $y_i' = f'(x_i)$, то можно предположить, что $s_i = y_i'$ для всех $i = 0, 1, \dots, n$. Тогда на каждом частичном отрезке $[x_{i-1}, x_i]$ в соответствии с формулой (2) сплайн однозначно определяется значениями $y_{i-1}, y_i, y_{i-1}', y_i'$ (из-за этого его называют *локальным сплайном*).

Глобальные способы построения сплайнов

Для того чтобы сплайн $S_3(x)$ имел непрерывную на отрезке $[a, b]$ вторую производную $S_3''(x)$, необходимо выбирать наклоны s_i так, чтобы в точках x_i «стыка» многочленов $P_{3,i}$ и $P_{3,i+1}$ совпадали значения их двух производных:

$$P_{3,i}'(x_i) = P_{3,i+1}'(x_i), \quad i = 1, 2, \dots, n-1. \quad (3)$$

Используя выражение (2) и равенство (3) можно получить следующую систему уравнений относительно коэффициентов s_i' :

$$h_i^{-1}s_{i-1} + 2(h_i^{-1} + h_{i+1}^{-1})s_i + h_{i+1}^{-1}s_{i+1} = 3[h_i^{-2}(y_i - y_{i-1}) + h_{i+1}^{-2}(y_{i+1} - y_i)], \quad (4)$$
$$i = 1, 2, \dots, n-1.$$

Система (4) не доопределена, так как число уравнений системы $(n-1)$ меньше числа неизвестных $(n+1)$. Выбор двух оставшихся уравнений обычно связывают с некоторыми дополнительными условиями, накладываемыми на сплайн в граничных точках a, b (*граничными условиями*):

1. Если в граничных точках известны значения первой производной $f'(a)$ и $f'(b)$, то можно допустить:

$$s_0 = f'(a), \quad s_n = f'(b). \quad (5)$$

Дополняя систему (4) уравнениями (5) получим систему уравнений с трёхдиагональной матрицей. Сформированный таким образом сплайн называется *фундаментальным кубическим сплайном*.

2. Если в граничных точках известны значения второй производной $f''(a), f''(b)$, то можно наложить на сплайн граничные условия $S_3''(a) = P_{3,1}''(x_0) = f''(a)$, $S_3''(b) = P_{3,1}''(x_n) = f''(b)$, что приводит к следующим уравнениям:

$$-\frac{4s_0}{h_1} - \frac{2s_1}{h_1} + 6\frac{y_1 - y_0}{h_1^2} = f''(a);$$
$$\frac{2s_{n-1}}{h_n} + \frac{4s_n}{h_n} - 6\frac{y_n - y_{n-1}}{h_n^2} = f''(b). \quad (6)$$

3. Допуская, что в уравнениях (6) $f''(a) = 0$, $f''(b) = 0$, придем к системе уравнений, определяющих так называемый естественный кубический сплайн.

Аппроксимация

При интерполировании функция строго проходит через узловые точки таблицы вследствие того, что количество коэффициентов в интерполирующей функции равно количеству табличных значений.

Аппроксимация – метод приближения, при котором для нахождения дополнительных значений, отличных от табличных данных, приближенная функция проходит не через узлы интерполяции (рис. 2).

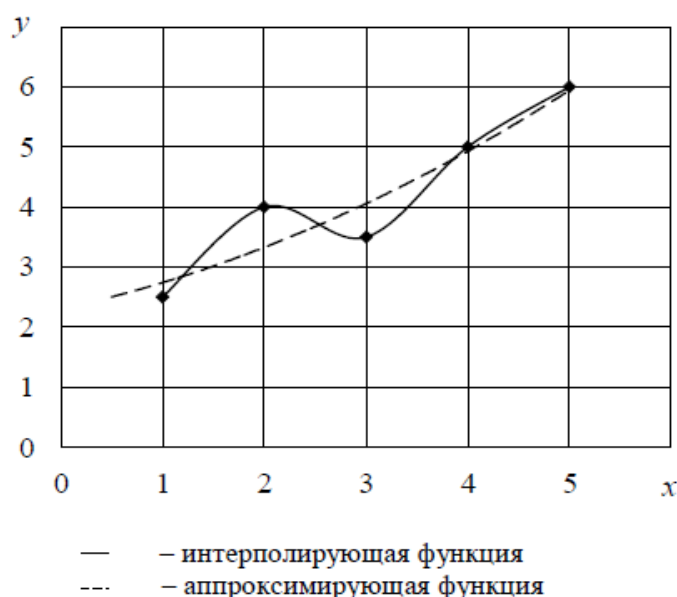


Рис. 2. Аппроксимация и интерполяция экспериментальных данных

Если аналитическое выражение функции, описывающей закон изменения y_i ($i = 1, 2, \dots, n$), неизвестно или весьма сложно, то возникает задача найти такую эмпирическую формулу, значения которой при $x = x_i$ мало отличались бы от экспериментальных данных:

$$f = y(x).$$

Геометрически задача построения функции $f(x)$ по эмпирической формуле состоит в проведении усредненной кривой – кривой, проходящей через середину области значений.

При интерполяции данные описываются более точно, чем при аппроксимации, но в ряде случаев обосновано применение аппроксимации:

- при значительном количестве табличных данных (интерполирующая функция становится громоздкой);
- интерполирующей функцией невозможно описать данные при повторении эксперимента в одних тех же начальных условиях (требуется статистическая обработка;
- для сглаживания погрешностей эксперимента.

Данные x_i и y_i обычно содержат ошибки, поэтому интерполяционная формула повторяет эти ошибки. Из рис. 3 видно, что значения y постоянно и равномерно увеличивается при росте x , а разброс данных относительно аппроксимирующей функции можно объяснить погрешностью эксперимента.

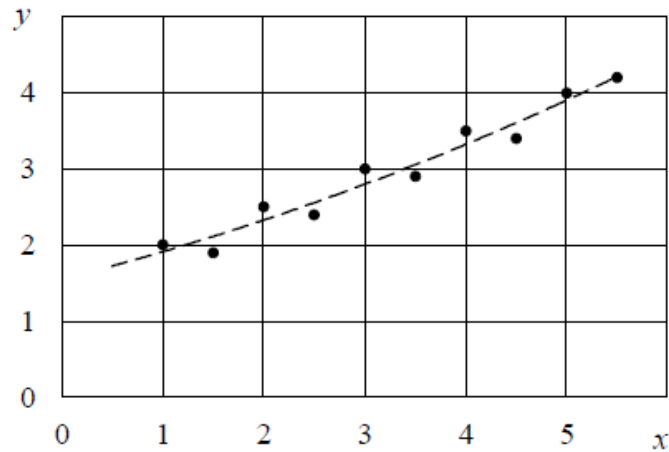


Рис. 3. Аппроксимация экспериментальных данных

При построении аппроксимирующей зависимости определяют:

- аналитический характер эмпирической формулы (предпочтение отдается простым формулам, обладающим хорошей точностью);
- наилучшие параметры эмпирической зависимости.

Метод наименьших квадратов

Суть метода наименьших квадратов (МНК) заключается в нахождении таких значений x_i , при которых сумма квадратов отклонений (ошибок) $e_i = y_i - f_i(x)$ будет стремиться к минимуму:

$$\sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - f(x_i))^2 \rightarrow \min. \quad (7)$$

Поскольку каждое значение x_i в общем случае «сопровождается» соответствующим коэффициентом a_i ($i = 0, 1, 2, \dots, n$), то задача сводится к нахождению данных коэффициентов. Введем обозначение функции:

$$F(a_0, a_1, \dots, a_n) = \sum_{i=1}^n (y_i - f(x_i))^2. \quad (8)$$

Тогда, на основе обращения в точке минимума функции F в нуль ее производных, для определения вышеупомянутых коэффициентов составляется нормальная система уравнений:

$$\begin{cases} \frac{dF}{da_0} = 0; \\ \frac{dF}{da_1} = 0; \\ \dots \\ \frac{dF}{da_n} = 0. \end{cases}$$

Аппроксимация методом наименьших квадратов

В ряде случаев экспериментальные данные распределяются таким образом, что оказывается возможным описать их изменение линейной зависимостью (линейным уравнением) следующего вида:

$$P(x) = a \cdot x + b. \quad (9)$$

Формулы для расчета коэффициентов a и b определяются по методу наименьших квадратов (7). Подставив (9) в (8), получаем:

$$F = \sum_{i=1}^n (y_i - a \cdot x_i - b)^2 \rightarrow \min. \quad (10)$$

Для решения (10) составляется система из двух уравнений с двумя неизвестными:

$$\begin{cases} \frac{dF}{da} = 0; \\ \frac{dF}{db} = 0. \end{cases} \quad (11)$$

В результате подстановки в (10) формулы (11) образуется следующая система:

$$\begin{cases} \frac{dF}{db} = -2 \cdot \sum_{i=1}^n (y_i - a \cdot x_i - b) \cdot 1 = 0; \\ \frac{dF}{da} = -2 \cdot \sum_{i=1}^n (y_i - a \cdot x_i - b) \cdot x_i = 0. \end{cases}$$

Группировка относительно коэффициентов позволяет привести систему к следующему виду:

$$\begin{cases} a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i = \sum_{i=1}^n (x_i \cdot y_i); \\ a \sum_{i=1}^n x_i + n \cdot b \sum_{i=1}^n y_i. \end{cases} \quad (12)$$

Решая систему (12) методом подстановки, получим формулы для нахождения коэффициентов a и b :

$$a = \frac{n \cdot \sum_{i=1}^n (x_i \cdot y_i) - \sum_{i=1}^n x_i \cdot \sum_{i=1}^n y_i}{n \cdot \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2},$$

$$b = \frac{\sum_{i=1}^n y_i \cdot \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right) \sum_{i=1}^n (x_i \cdot y_i)}{n \cdot \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}.$$

Если экспериментальные данные распределяются таким образом, что оказывается невозможным описать их изменение линейной зависимостью, тогда используют зависимости более высокого порядка:

$$P_2(x) = a_0 + a_1x + a_2x^2.$$

Далее аналогичным образом применяется МНК для расчета коэффициентов:

$$F = \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2)^2 \rightarrow \min. \quad (13)$$

Для решения (13) составляется система из трех уравнений с тремя неизвестными:

$$\begin{cases} \frac{dF}{da_0} = -2 \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2) \cdot 1 = 0; \\ \frac{dF}{da_1} = -2 \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2) \cdot x_i = 0; \\ \frac{dF}{da_2} = -2 \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2) \cdot x_i^2 = 0. \end{cases}$$

Выполняем группировку относительно коэффициентов:

$$\begin{cases} a_0 \cdot n + a_1 \cdot \sum_{i=1}^n x_i + a_2 \cdot \sum_{i=1}^n x_i^2 = \sum_{i=1}^n y_i; \\ a_0 \cdot \sum_{i=1}^n x_i + a_1 \cdot \sum_{i=1}^n x_i^2 + a_2 \cdot \sum_{i=1}^n x_i^3 = \sum_{i=1}^n (x_i \cdot y_i); \\ a_0 \cdot \sum_{i=1}^n x_i^2 + a_1 \cdot \sum_{i=1}^n x_i^3 + a_2 \cdot \sum_{i=1}^n x_i^4 = \sum_{i=1}^n (x_i^2 \cdot y_i). \end{cases}$$

Вводим обозначение:

$$\begin{aligned} S_1 &= \sum_{i=1}^n x_i; \quad S_2 = \sum_{i=1}^n x_i^2; \quad S_3 = \sum_{i=1}^n x_i^3; \quad S_4 = \sum_{i=1}^n x_i^4; \\ S_5 &= \sum_{i=1}^n y_i; \quad S_6 = \sum_{i=1}^n (x_i \cdot y_i); \quad S_7 = \sum_{i=1}^n (x_i^2 \cdot y_i). \end{aligned}$$

Получаем систему уравнений для поиска коэффициентов a_0, a_1, a_2 :

$$\begin{cases} a_0 \cdot n + a_1 \cdot S_1 + a_2 \cdot S_2 = S_5; \\ a_0 \cdot S_1 + a_1 \cdot S_2 + a_2 \cdot S_3 = S_6; \\ a_0 \cdot S_2 + a_1 \cdot S_3 + a_2 \cdot S_4 = S_7. \end{cases}$$

Примеры программной реализации методов восстановления функции одной переменной в пакете MATLAB

Пример 1. Генерация, графическое отображение, сохранение и загрузка данных в пакете MATLAB. Для составления исходной таблицы данных в пакете MATLAB может быть использован подход, представленный на рис. 4.

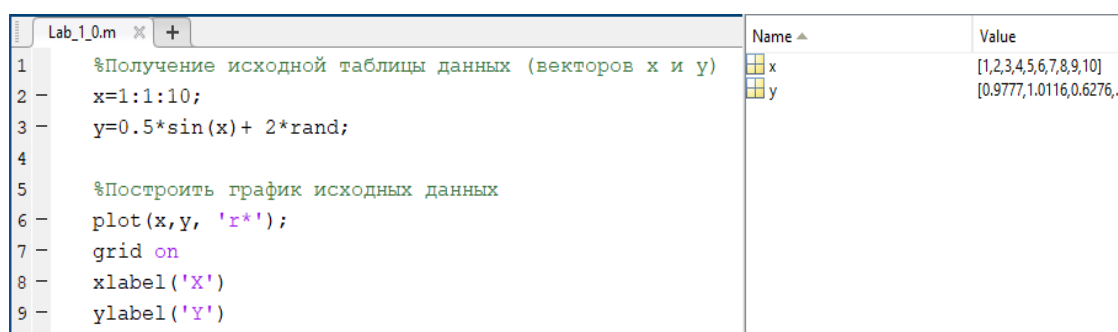


Рис. 4. Получение исходной таблицы данных в пакете MATLAB

1-я строка кода предназначена для определения переменной x от 1 до 10 с постоянным шагом 1. 2-я строка кода – для генерации значений переменной y , которая зависит от переменной x согласно заданному выражению. Команда *rand* позволяет генерировать случайные значения от 0 до 1 в соответствии с нормальным законом распределения.

В правой части экрана, в области Workspace, показаны сгенерированные значения переменных x и y .

Вторая часть кода, строки с 6-й по 9-ю предназначены для графического представления полученных данных. Команда *plot* позволяет построить двумерный график переменных x и y . Команда *grid* используется для отображения сетки. И команды *xlabel* и *ylabel* задают названия осей. Результат выполнения данного кода (рис. 4) представлен на рис. 5. Имеем десять точек, которые по x лежат в диапазоне от 1 до 10, а по y – в диапазоне от 1,3321 до 2,3063.

Данные x и y можно сохранить для удобства дальнейшей обработки. Для этого необходимо в области Workspace выделить ту или иную переменную и, нажав правой кнопкой в появившемся меню, выбрать команду *Save as* (рис. 6).

Далее сохраняем каждую переменную под соответствующим именем *x.mat* и *y.mat*.

В дальнейшем сохраненные данные можно будет использовать, загружая их с помощью команды *load* (рис. 7).

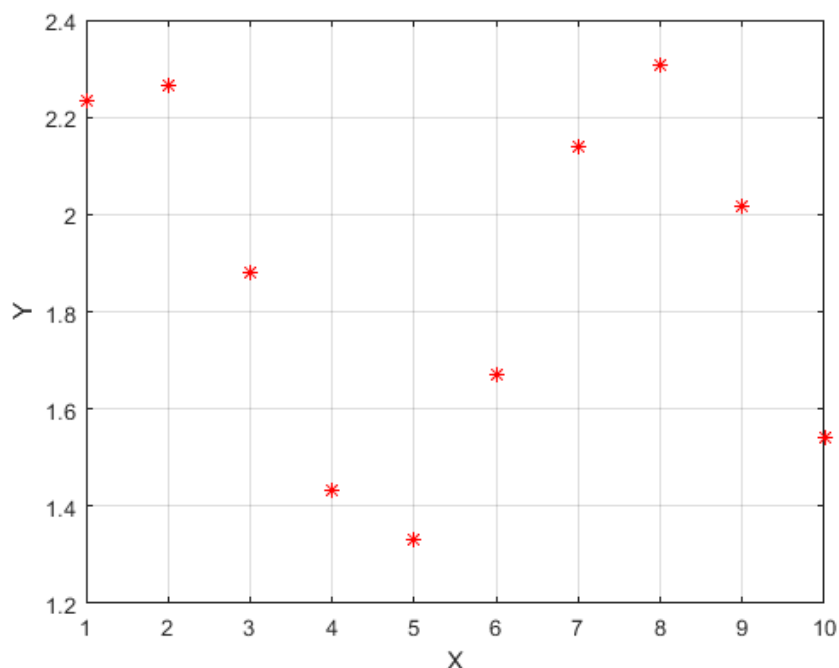


Рис. 5. График исходных данных в пакете MATLAB

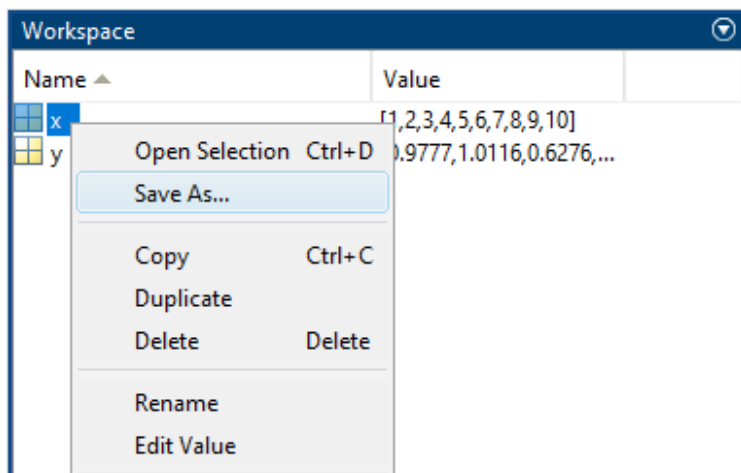


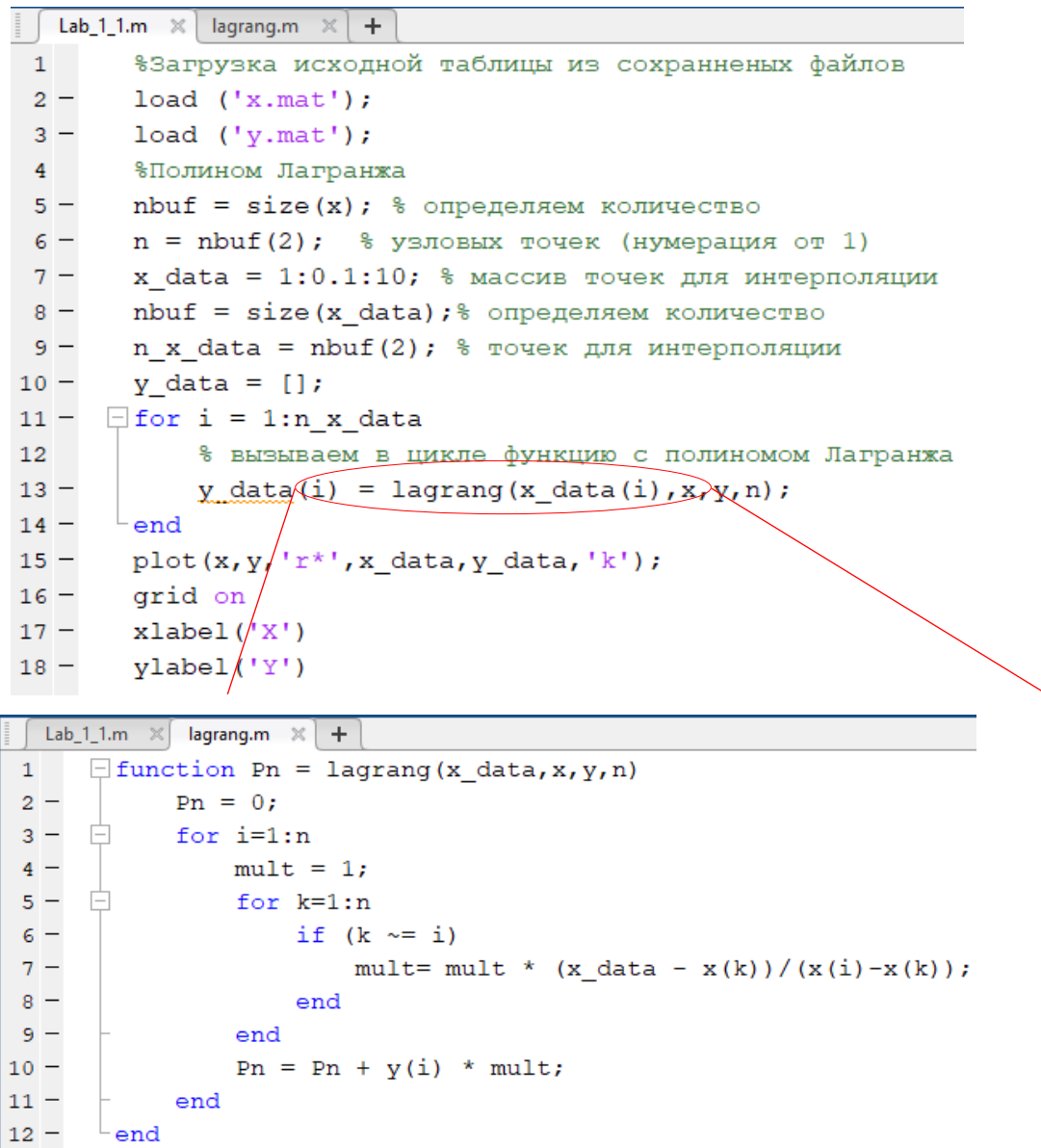
Рис. 6. Сохранение данных в пакете MATLAB

```

1 %Загрузка исходной таблицы из сохраненных файлов
2 load ('x.mat');
3 load ('y.mat');
4 %Построить график исходных данных
5 plot(x,y, 'r*');
6 grid on
7 xlabel('X')
8 ylabel('Y')
```

Рис. 7. Загрузка сохраненных данных в пакете MATLAB

Пример 2. Интерполяция полиномом Лагранжа. Интерполяционный полином Лагранжа, в общем виде описываемый выражением (1), достаточно легко реализовать в виде программного кода в пакете MATLAB. Пример возможной программной реализации показан на рис. 8.



```

1 %Загрузка исходной таблицы из сохраненных файлов
2 load ('x.mat');
3 load ('y.mat');
4 %Полином Лагранжа
5 nbuf = size(x); % определяем количество
6 n = nbuf(2); % узловых точек (нумерация от 1)
7 x_data = 1:0.1:10; % массив точек для интерполяции
8 nbuf = size(x_data); % определяем количество
9 n_x_data = nbuf(2); % точек для интерполяции
10 y_data = [];
11 for i = 1:n_x_data
12     % вызываем в цикле функцию с полиномом Лагранжа
13     y_data(i) = lagrang(x_data(i), x, y, n);
14 end
15 plot(x, y, 'r*', x_data, y_data, 'k');
16 grid on
17 xlabel('X')
18 ylabel('Y')

```

```

1 function Pn = lagrang(x_data, x, y, n)
2     Pn = 0;
3     for i=1:n
4         mult = 1;
5         for k=1:n
6             if (k ~= i)
7                 mult= mult * (x_data - x(k))/(x(i)-x(k));
8             end
9         end
10        Pn = Pn + y(i) * mult;
11    end
12 end

```

Рис. 8. Программная реализация интерполяционного полинома Лагранжа в пакете MATLAB

Как можно заметить (рис. 8), первые две строчки кода реализуют загрузку исходных данных. Строки 5–6 определяют количество узловых точек, а строки 7–9 – количество точек интерполяции. Затем в цикле *for*, строки 11–14, происходит вызов заранее разработанной функции *lagrang* для расчета значений функции.

Также на рис. 8 показано содержание функции *lagrang*. Непосредственно в функции *lagrang* запрограммированы выражения (1). Для этого используется два вложенных цикла *for* и проверка условия с помощью оператора *if*. В строке 7 записан множитель Лагранжа, который представляет собой произведение дробей, а строка 10 – непосредственную реализацию полинома Лагранжа (1).

Рассчитанные значения функции строятся с помощью уже известной команды *plot*. Результат представлен на рис. 9.

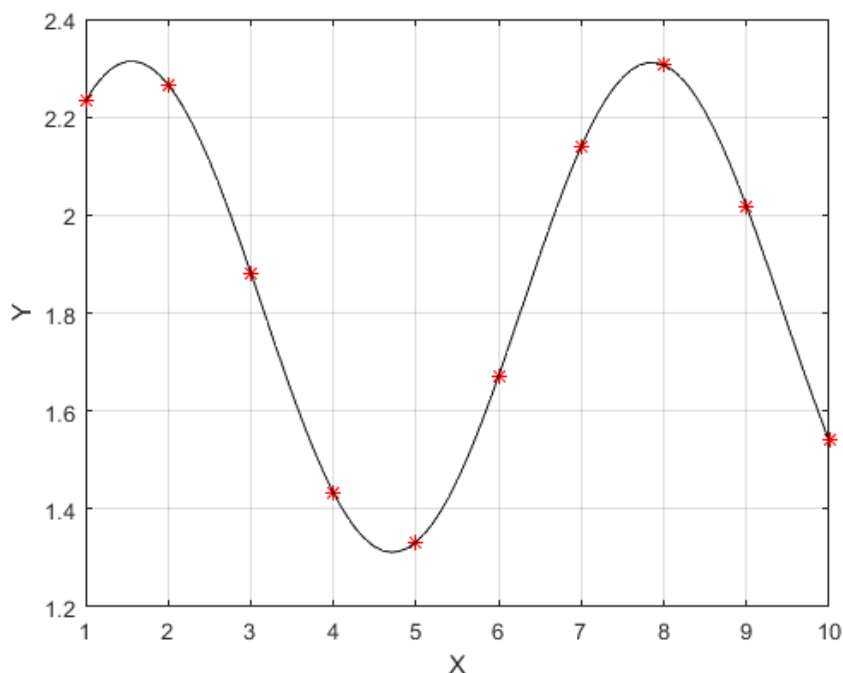


Рис. 9. Интерполяция полиномом Лагранжа в пакете MATLAB

Пример 3. Интерполяция сплайнами. Интерполяцию сплайнами в пакете MATLAB достаточно удобно выполнить с помощью встроенной команды *interp1*. Пример возможной программной реализации представлен на рис. 10.

Как можно заметить (рис. 10), первые две строчки кода реализуют загрузку исходных данных. Строки 5–7 определяют количество точек интерполяции. Затем в цикле *for*, строки 9–11, происходит вызов встроенной функции *interp1* для расчета значений функции. При этом в параметрах функции *interp1* указана линейная интерполяция *linear*.

Полученные значения функции строятся с помощью уже известной команды *plot*. Результат представлен на рис. 11.

Аналогичным образом в пакете MATLAB можно выполнить кубическую интерполяцию сплайнами с помощью встроенной команды *interp1*. Пример возможной программной реализации показан на рис. 12.

```

1      %Загрузка исходной таблицы из сохраненных файлов
2      load ('x.mat');
3      load ('y.mat');
4      %Полином Лагранжа
5      x_data = 1:0.1:10; % массив точек для интерполяции
6      nbuf = size(x_data); % определяем количество
7      n_x_data = nbuf(2); % точек для интерполяции
8      y_data = [];
9      for i=1:n_x_data
10         y_data(i) = interp1(x,y,x_data(i),'linear');
11     end
12     plot(x,y,'r*',x_data,y_data,'k');
13     grid on
14     xlabel('X')
15     ylabel('Y')

```

Рис. 10. Программная реализация интерполяции сплайнами в пакете MATLAB

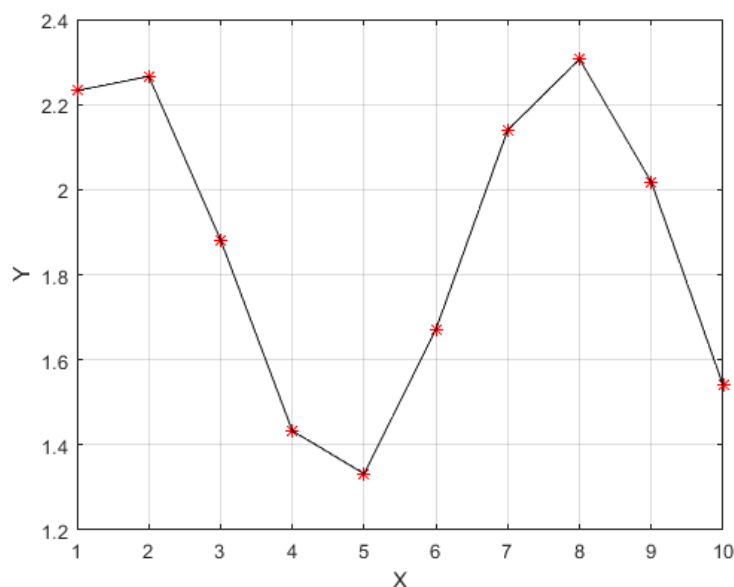


Рис. 11. Интерполяция линейным сплайном в пакете MATLAB

Как можно заметить (рис. 12), в первых двух строчках кода реализуется загрузка исходных данных. Строки 5–7 определяют количество точек интерполяции. Затем в цикле *for*, строки 9–11, происходит вызов встроенной функции *interp1* для расчета значений функции. При этом в параметрах функции *interp1* указана линейная интерполяция *spline*.

Полученные значения функции строятся с помощью уже известной команды *plot*. Результат представлен на рис. 13.

В отличие от линейной интерполяции сплайнами кубическая интерполяция сплайнами обеспечивает более сглаженный вид кривой (рис. 11 и 13).

```

1      %Загрузка исходной таблицы из сохраненных файлов
2      load ('x.mat');
3      load ('y.mat');
4      %Полином Лагранжа
5      x_data = 1:0.1:10; % массив точек для интерполяции
6      nbuf = size(x_data); % определяем количество
7      n_x_data = nbuf(2); % точек для интерполяции
8      y_data = [];
9      for i=1:n_x_data
10         y_data(i) = interp1(x,y,x_data(i),'spline');
11     end
12     plot(x,y,'r*',x_data,y_data,'k');
13     grid on
14     xlabel('X')
15     ylabel('Y')

```

Рис. 12. Программная реализация кубической интерполяции сплайнами в пакете MATLAB

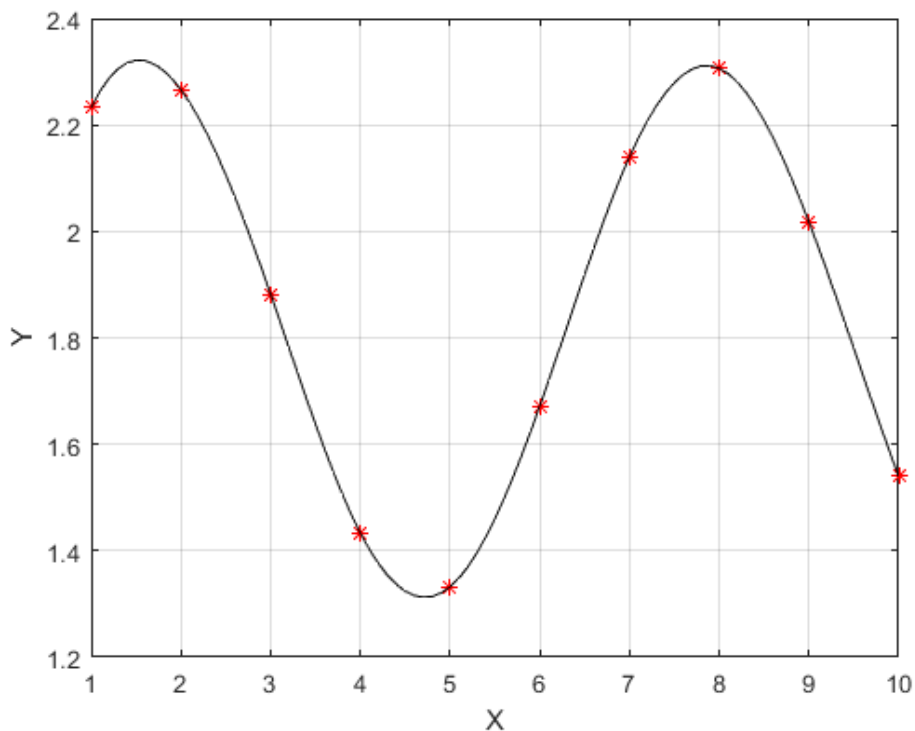


Рис. 13. Кубическая интерполяция сплайнами в пакете MATLAB

Пример 4. Аппроксимация методом наименьших квадратов. Одним из широко распространенных методов аппроксимации исходных данных является метод наименьших квадратов. Для применения данного метода в пакете MATLAB существуют встроенные функции. Пример возможной программной реализации аппроксимации методом наименьших квадратов представлен на рис. 14.

Как можно заметить (рис. 14), в первых двух строчках кода реализуется загрузка исходных данных. Строки 4–6 определяют количество точек аппроксимации. Затем, с помощью встроенной функции *polyfit* в строках 9–12, задаются коэффициенты аппроксимирующих полиномов соответствующей степени. В цикле *for*, строки 13–18, происходит расчет значений аппроксимирующей функции с использованием значений коэффициентов аппроксимирующих полиномов.

```

1      %Загрузка исходной таблицы из сохраненных файлов
2 -   load ('x.mat');
3 -   load ('y.mat');
4 -   x_data = 1:0.1:10; % массив точек для аппроксимации
5 -   nbuf = size(x_data); % определяем количество
6 -   n_x_data = nbuf(2); % точек для аппроксимации
7 -   y_data_2 = [];
8 -   y_data_3 = [];
9 -   coeff_2 = [];
10 -  coeff_2 = polyfit(x,y,2); %определяем коэффициенты полинома 2-ой степени
11 -  coeff_3 = [];
12 -  coeff_3 = polyfit(x,y,3); %определяем коэффициенты полинома 3-ей степени
13 -  for i=1:n_x_data
14 -      y_data_2(i) = coeff_2(1)*x_data(i)^2 + ...
15 -          +coeff_2(2)*x_data(i) + coeff_2(3);
16 -      y_data_3(i) = coeff_3(1)*x_data(i)^3 + ...
17 -          +coeff_3(2)*x_data(i)^2 + coeff_3(3)*x_data(i) + coeff_3(4);
18 -  end
19 -  plot(x,y,'r*',x_data,y_data_2,'k');
20 -  grid on
21 -  legend('Исходные данные', 'Полином 2-ой степени')
22 -  xlabel('X')
23 -  ylabel('Y')
24 -  figure;
25 -  plot(x,y,'r*',x_data,y_data_2,'--b', x_data,y_data_3,'k');
26 -  grid on
27 -  legend('Исходные данные', 'Полином 2-ой степени', 'Полином 3-ей степени')
28 -  xlabel('X')
29 -  ylabel('Y')

```

Рис. 14. Программная реализация аппроксимации методом наименьших квадратов в пакете MATLAB

Полученные значения аппроксимирующих функций представляются графически с помощью уже известной команды *plot*. Результат показан на рис. 15.

Как можно заметить (рис. 15), была выполнена аппроксимация исходных данных с помощью полиномов второй и третьей степеней. При увеличении степени аппроксимирующего полинома точность описания исходных данных значительно возрастает.

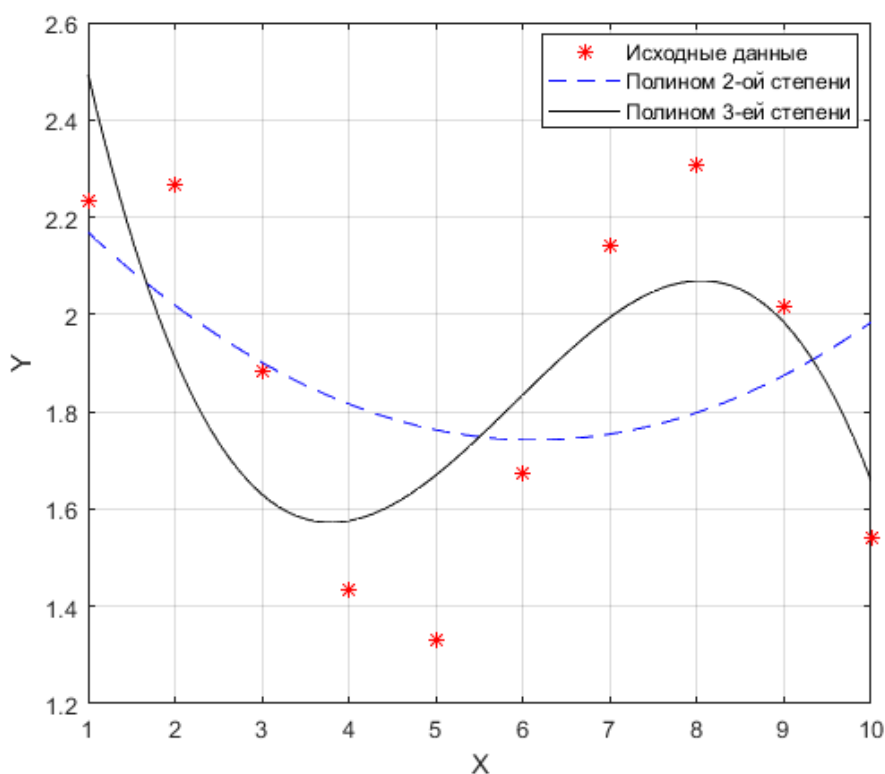


Рис. 15. Аппроксимация методом наименьших квадратов в пакете MATLAB

Индивидуальные задания

1. В соответствии с индивидуальным вариантом сгенерировать таблицу исходных данных (x_i, y_i) в пакете MATLAB.

Номер варианта	Вид функции y и значения аргумента x
1	$y = \log(x) + 0,4 \cdot \text{random}, x = 2 \dots 8, \Delta x = 1$
2	$y = e^x + 0,5 \cdot \text{random}, x = 1 \dots 5, \Delta x = 0,25$
3	$y = 1/x + \text{random}, x = 0,3 \dots 6,3, \Delta x = 1$
4	$y = x^2 + \text{random}, x = 0 \dots 20, \Delta x = 2$
5	$y = x + \text{random}, x = 10 \dots 50, \Delta x = 1$
6	$y = x + x^2 + 0,1 \cdot \text{random}, x = 0 \dots 7, \Delta x = 0,5$
7	$y = \sin(x) + 0,5 \cdot \text{random}, x = 0 \dots 2 \cdot \pi, \Delta x = \pi/5$
8	$y = (\sin(x))^2 + 0,25 \cdot \text{random}, x = \pi \dots 3 \cdot \pi, \Delta x = \pi/5$
9	$y = x^{1/2} + 0,5 \cdot \text{random}, x = 0 \dots 20, \Delta x = 2$
10	$y = \cos(x) + 0,1 \cdot \text{random}, x = 0 \dots \pi, \Delta x = \pi/20$
11	$y = \log(x) + 0,2 \cdot \text{random}, x = 5 \dots 10, \Delta x = 0,2$
12	$y = e^x - 0,2 \cdot \text{random}, x = 0 \dots 5, \Delta x = 0,5$
13	$y = x^{-1} + 0,1 \cdot \text{random}, x = 0,2 \dots 7,2, \Delta x = 0,25$
14	$y = x^2 + 2 \cdot \text{random}, x = 10 \dots 20, \Delta x = 1$
15	$y = x - \text{random}, x = 40 \dots 50, \Delta x = 0,5$
16	$y = x + x^2 - \text{random}, x = 0 \dots 4, \Delta x = 0,25$

Номер варианта	Вид функции y и значения аргумента x
17	$y = \sin(x) + 0,1 \cdot random, x = \pi \dots 3 \cdot \pi, \Delta x = \pi/10$
18	$y = (\sin(x))^2 - 0,6 \cdot random, x = 0 \dots \pi, \Delta x = \pi/10$
19	$y = x^{1/2} - 0,7 \cdot random, x = 2 \dots 10, \Delta x = 0,5$
20	$y = \cos(x) + random, x = 0 \dots \pi, \Delta x = \pi/10$

2. Используя функцию *plot*, вывести график полученных исходных данных.

3. В пакете MATLAB программно реализовать интерполяционный полином Лагранжа:

$$L_n(x) = \sum_{j=0}^n y_j \cdot l_{nj}(x),$$

где

$$l_{nj}(x) = \prod_{\substack{k=1 \\ k \neq j}}^n \frac{x - x_k}{x_j - x_k}.$$

Полученный результат представить графически, используя функцию *plot*.

4. Используя встроенную функцию *interp1*, выполнить интерполяцию исходных данных линейными сплайнами. Полученный результат представить графически с помощью функции *plot*.

5. Используя встроенную функцию *interp1*, выполнить интерполяцию исходных данных естественными кубическими сплайнами. Полученный результат представить графически с помощью функции *plot*.

6. Используя инструменты пакета MATLAB, выполнить аппроксимацию исходных данных методом наименьших квадратов. При этом применить степенные полиномы 1–3-го порядков в качестве базисных функций. Полученные результаты представить графически с помощью функции *plot*, сравнить и сделать выводы.

7. Подготовить отчет в соответствии с требованиями, представленными в прил. А–В. Проанализировать полученные результаты и сделать выводы.

Контрольные вопросы и задания

1. Что такое интерполяция, аппроксимация и экстраполяция? В чем их отличия?
2. Дайте определение понятию «сплайн».
3. Что понимается под термином дефект сплайна?
4. Запишите интерполяционный полином Лагранжа третьей степени.

5. Какие методы аппроксимации Вам известны?
6. В чем суть метода наименьших квадратов?

Список рекомендованной литературы

1. Герасимович А.С. Аппроксимация зависимостей полиномами. Метод накопленной переменной / А.С. Герасимович. – Сергиев Посад: Весь Сергиев Посад, 2013. – 101 с.
2. Половко А.М. Интерполяция. Методы и компьютерные технологии их реализации / А.М. Половко, П.Н. Бутусов. – Санкт-Петербург : БХВ-Петербург, 2004. – 320 с.

Лабораторная работа № 2

ИЗУЧЕНИЕ МЕТОДОВ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ И ДИФФЕРЕНЦИРОВАНИЯ ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ

Цель работы: изучение методов численного интегрирования и дифференцирования функции одной переменной с использованием специализированного программного обеспечения MATLAB.

Основные теоретические сведения

Методы численного интегрирования

Методы численного интегрирования предназначены для вычисления определенного интеграла заданной функции $f(x)$:

$$I = \int_a^b f(x) dx.$$

Как правило, численные методы интегрирования целесообразно использовать в следующих случаях:

- для интегрирования функций, значения которых известны только в некоторых отдельных точках (узлах), например полученных в результате измерений и экспериментальных исследований;
- интегрирования сложных выражений, не имеющих элементарных первообразных либо имеющих слишком громоздкие выражения для них;
- построения методов численного решения уравнений в обыкновенных и частных производных (методы конечных элементов, интегроинтерполяционные методы).

На практике наибольшее распространение получили следующие методы численного интегрирования:

- метод прямоугольников;
- метод трапеций;
- метод Симпсона.

Метод прямоугольников

Одним из наиболее простых методов численного интегрирования является метод прямоугольников, который, в свою очередь, подразделяется еще на три метода:

- метод левых прямоугольников (МЛП);
- метод правых прямоугольников (МПП);
- метод средних (центральных) прямоугольников (МСП).

Как известно, геометрический смысл интегрирования заключается в нахождении площади (S), которая образована подынтегральной функцией ($f(x)$) на интервале от a до b (рис. 16).

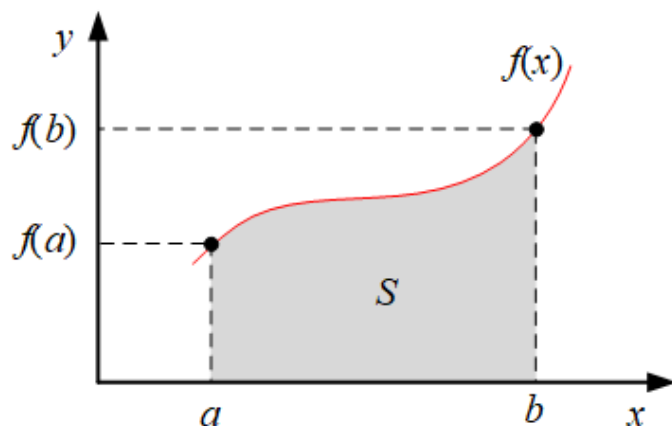


Рис. 16. Геометрический смысл интегрирования

Метод прямоугольников основывается на том, что предполагает нахождение площади (S), которая образована подынтегральной функцией ($f(x)$) на интервале от a до b как площадь прямоугольника. При этом в зависимости от подхода к выделению прямоугольника и формируются методы левых, правых и средних прямоугольников.

Как известно, площадь прямоугольника определяется как произведение его длины и ширины. В данном случае в качестве ширины прямоугольника может быть использована величина $(b-a)$, а длиной прямоугольника является значение функции $f(x)$ в точке a или b .

В случае, если в качестве длины прямоугольника берется значение функции $f(x)$ в точке a , то получаем *метод левых прямоугольников* (рис. 17).

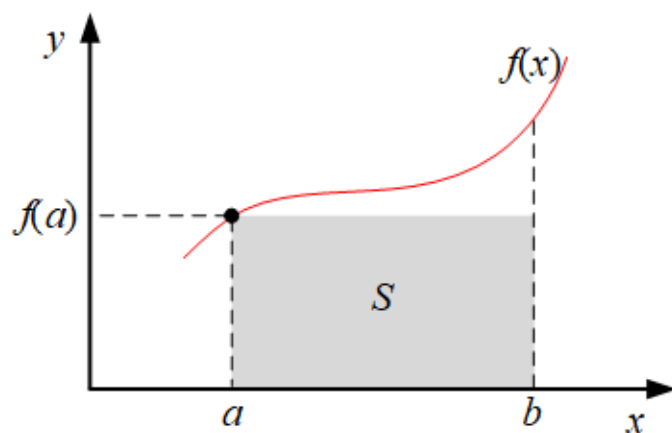


Рис. 17. Метод левых прямоугольников

Формула для МЛП выглядит следующим образом:

$$I = \int_a^b f(x) dx = (b - a) \cdot f(a).$$

В случае, если в качестве длины прямоугольника берется значение функции $f(x)$ в точке b , то получаем *метод правых прямоугольников* (рис. 18).

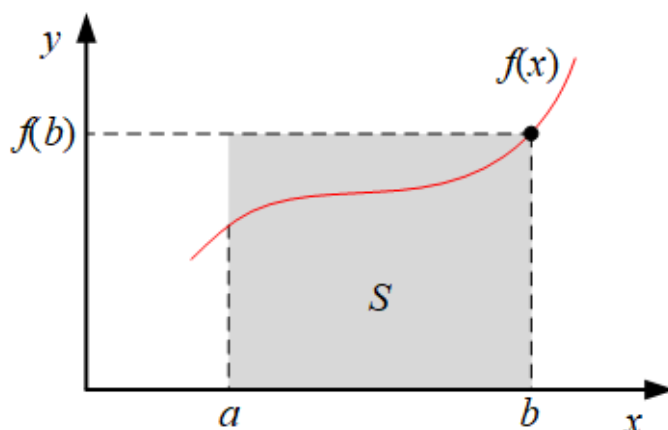


Рис. 18. Метод правых прямоугольников

Формула для МПП выглядит следующим образом:

$$I = \int_a^b f(x) dx = (b - a) \cdot f(b).$$

Нетрудно заметить, что ввиду своих особенностей методы левых и правых прямоугольников не отличаются высокой точностью интегрирования. На примерах (рис. 17, 18) видно, что метод левых прямоугольников существенно занижает результат, а метод правых прямоугольников – завышает.

Большой точностью, по сравнению с представленными методами, обладает *метод средних прямоугольников*. В данном случае в качестве длины прямоугольника берется значение функции $f(x)$ в точке $(a + b)/2$ (рис. 19).

Формула для МСП выглядит следующим образом:

$$I = \int_a^b f(x) dx = (b - a) \cdot f\left(\frac{a + b}{2}\right).$$

Как видно на рис. 19, метод средних прямоугольников более точно описывает площадь подынтегральной функции. Соответственно, данный метод обладает большей точностью интегрирования по сравнению с методами левых и правых прямоугольников.

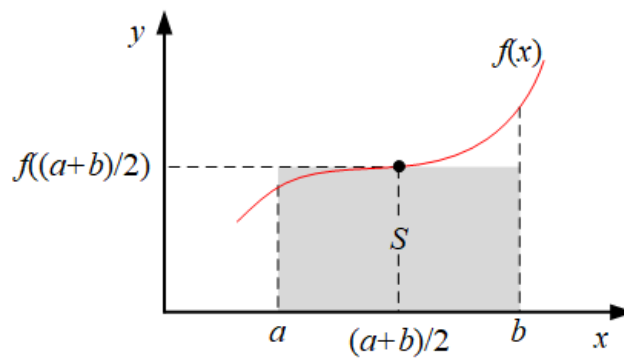


Рис. 19. Метод средних прямоугольников

Для увеличения точности вычислений подынтегральную функцию ($f(x)$) на интервале от a до b разбивают на более мелкие интервалы (рис. 20).

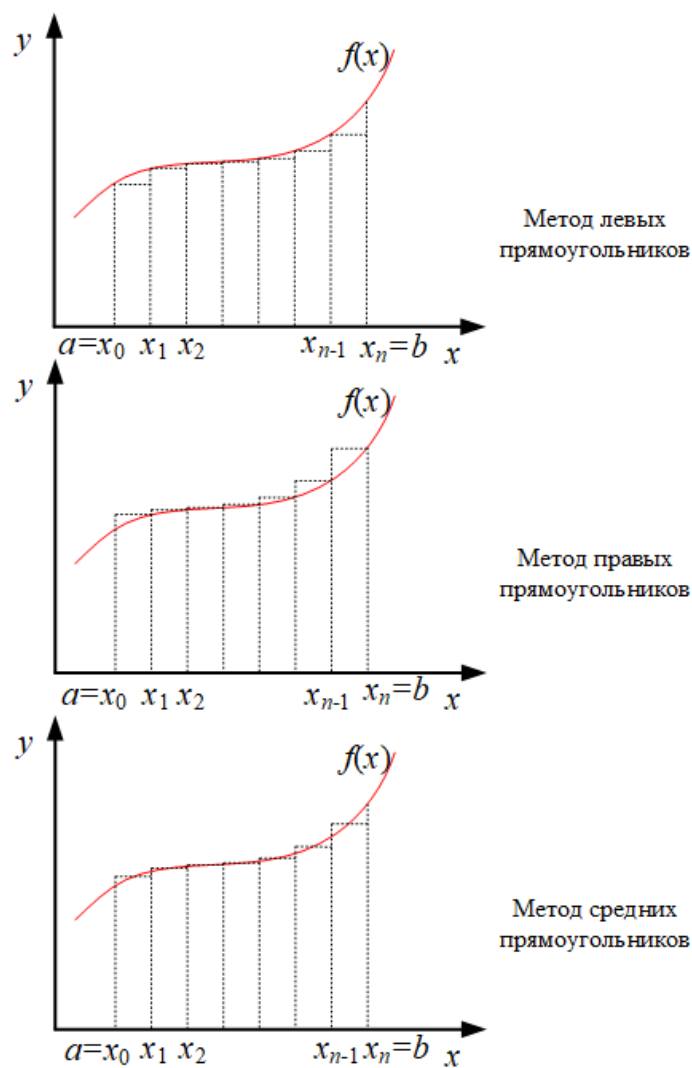


Рис. 20. Метод прямоугольников

Тогда формулы для методов левых, правых и средних прямоугольников принимают следующий вид:

$$\begin{aligned}
 I &= \int_a^b f(x) dx = \sum_{i=0}^{n-1} f(x_i) \cdot (x_{i+1} - x_i); \\
 I &= \int_a^b f(x) dx = \sum_{i=1}^n f(x_i) \cdot (x_i - x_{i-1}); \\
 I &= \int_a^b f(x) dx = \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right) \cdot (x_{i+1} - x_i) = \\
 &= \sum_{i=1}^n f\left(\frac{x_{i-1} + x_i}{2}\right) \cdot (x_i - x_{i-1}).
 \end{aligned} \tag{14}$$

Метод трапеций

Еще одним простым и достаточно эффективным методом численного интегрирования является метод трапеций. В основе данного метода лежит подход нахождения площади (S) (рис. 16), которая образована подынтегральной функцией ($f(x)$) на интервале от a до b как площадь трапеции.

Как известно, площадь трапеции определяется как половина от суммы оснований, умноженная на высоту трапеции. В данном случае в качестве высоты трапеции может быть использована величина $(b - a)$, а длина оснований трапеции соответствует значениям функции $f(x)$ в точках a и b (рис. 21).

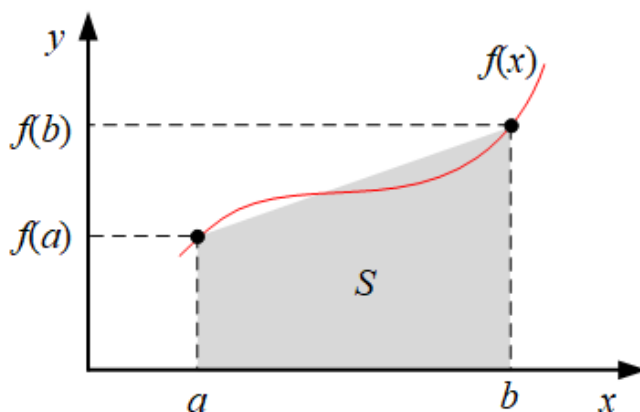


Рис. 21. Метод трапеций

Формула для метода трапеций выглядит следующим образом:

$$I = \int_a^b f(x) dx = (b - a) \cdot \frac{f(a) + f(b)}{2}.$$

В данном случае, для увеличения точности вычислений, подынтегральную функцию ($f(x)$) на интервале от a до b разбивают на более мелкие интервалы (рис. 22).

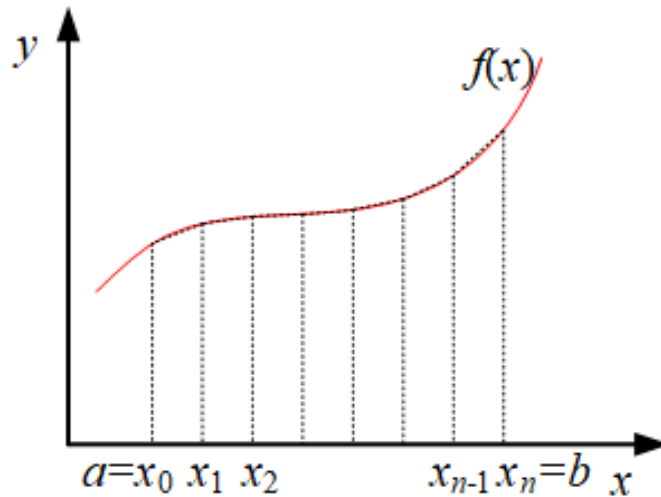


Рис. 22. Метод трапеций

Тогда формула для метода трапеций принимает следующий вид:

$$I = \int_a^b f(x) dx = \sum_{i=0}^{n-1} \frac{f(x_{i+1}) + f(x_i)}{2} \cdot (x_{i+1} - x_i). \quad (15)$$

В большинстве случаев метод трапеций позволяет получить более точный результат интегрирования даже по сравнению с методом средних прямоугольников. Но существуют функции, для которых применение метода прямоугольников является более эффективным.

Метод Симпсона

Наиболее точным и совершенным методом численного интегрирования является метод Симпсона, или метод парабол. В основе данного метода лежит подход нахождения площади (S), которая образована подынтегральной функцией ($f(x)$) на интервале от a до b как площадь параболы (рис. 23).

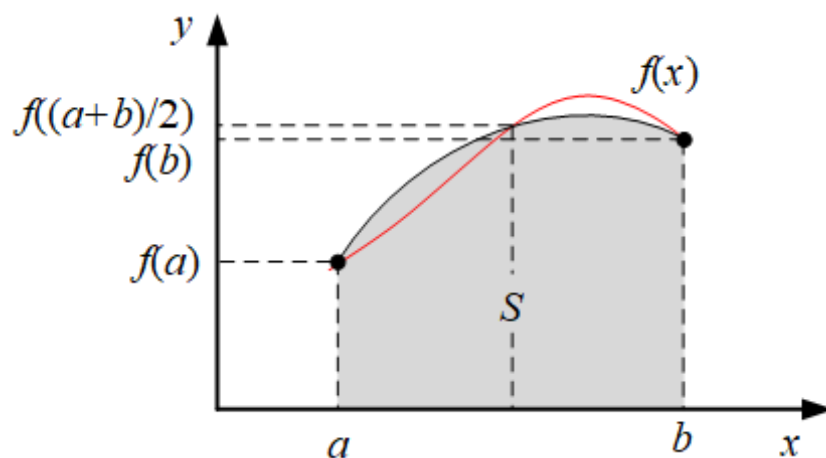


Рис. 23. Метод Симпсона

Формула для метода Симпсона имеет следующий вид:

$$I = \int_a^b f(x) dx = \frac{(b-a)}{6} \cdot \left(f(a) + 4 \cdot f\left(\frac{a+b}{2}\right) + f(b) \right).$$

Метод Симпсона позволяет получить более точный результат интегрирования даже по сравнению с методом трапеций. Однако в данном случае, аналогично методу прямоугольников и трапеций, для увеличения точности вычислений подынтегральную функцию ($f(x)$) на интервале от a до b разбивают на более мелкие интервалы.

Главной особенностью данного метода является то, что количество интервалов (N), на которые разбивается интервал от a до b должно быть четным ($N = 2n$). Тогда формула для метода Симпсона принимает следующий вид:

$$I = \int_a^b f(x) dx = \frac{h}{3} \cdot \left(f(x_0) + 4 \cdot \sum_{i=1}^n f(x_{2i-1}) + 2 \cdot \sum_{i=1}^{n-1} f(x_{2i}) + f(x_{2n}) \right), \quad (16)$$

где $h = \frac{b-a}{N}$.

Методы численного дифференцирования

Согласно определению, производной функции $y = f(x)$ является предел отношения приращения функции (Δy) к приращению аргумента (Δx), при стремлении аргумента к нулю ($\Delta x \rightarrow 0$):

$$\frac{dy}{dx} = y' = f'(x) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x},$$

$$\Delta y = f(x + \Delta x) - f(x).$$

На практике методы численного дифференцирования находят применение в следующих случаях:

- для вычисления производных функций, которые заданы в табличном виде (сеточные функции);
- выполнения аппроксимации производных в случае решения дифференциальных уравнений с помощью конечно-разностных методов.

В инженерной и научной практике достаточно широкое распространение получили следующие методы:

- метод левосторонних конечных разностей;
- метод правосторонних конечных разностей;
- метод центральных конечных разностей.

В случае, если функция представлена в табличном виде (рис. 24), то значениям аргумента x_i соответствуют определенные значения функции y_i . В таком случае разница между x_i и x_{i-1} является величиной постоянной и соответствует шагу дифференцирования $\Delta x = h$.

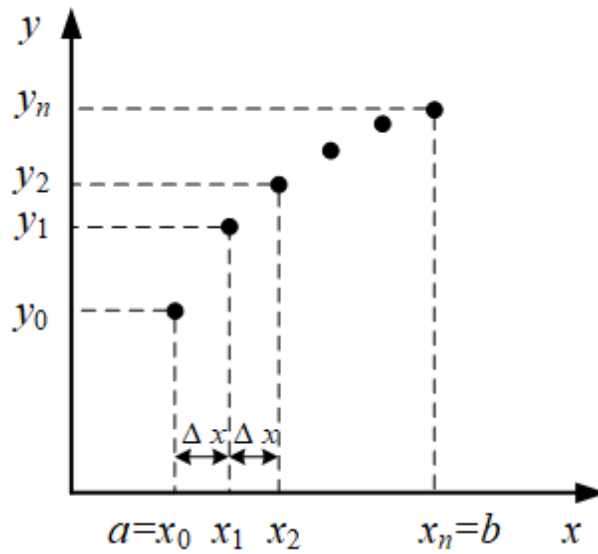


Рис. 24. Сеточная функция

Тогда, с учетом определения производной, можно рассчитать значение производной функции $y = f(x)$ в точке $x = x_1$, используя *метод левосторонних конечных разностей* по следующей формуле:

$$y'_1 = \frac{\Delta y}{\Delta x} \approx \frac{y_1 - y_0}{h} \approx \frac{y_1 - y_0}{x_1 - x_0}; \quad (17)$$

$$y'_1 = \frac{\Delta y}{\Delta x} \approx \frac{y_2 - y_1}{h} \approx \frac{y_2 - y_1}{x_2 - x_1}.$$

Аналогичным образом может быть вычислено значение производной функции $y = f(x)$ в точке $x = x_1$, с помощью *метода правосторонних конечных разностей* по следующей формуле:

$$\Delta y = y_2 - y_1, \quad \Delta x = x_2 - x_1 = h; \quad (18)$$

$$y'_1 = \frac{\Delta y}{\Delta x} \approx \frac{y_2 - y_1}{h}.$$

Представленные формулы (17), (18) соответствуют первому порядку точности. Одним из способов повышения их точности является уменьшение шага дифференцирования $\Delta x = h \rightarrow 0$.

Наибольшей точностью обладают формулы второго порядка точности, к которым относится *метод центральных конечных разностей*, используя который, можно определить значение производной функции $y = f(x)$ в точке $x = x_1$:

$$y'_1 \approx \frac{y_2 - y_0}{2 \cdot h}. \quad (19)$$

В данном случае, при уменьшении шага дифференцирования $\Delta x = h \rightarrow 0$, возрастает точность численного дифференцирования.

Повышать точность формул численного дифференцирования возможно не только с помощью уменьшения шага дифференцирования, но и увеличивая порядок точности. Для выполнения численного дифференцирования возможно получение формул любого порядка точности. Например, формула четвертого порядка точности имеет следующий вид:

$$y'_i \approx \frac{y_{i-2} - 8 \cdot y_{i-1} + 8 \cdot y_{i+1} - y_{i+2}}{12 \cdot h}. \quad (20)$$

Можно заметить, что увеличение порядка точности формул приводит как к росту точности численного дифференцирования, так и к росту количества задействованных значений функции в выражении (20).

Применяя методы численного дифференцирования, можно вычислить значения производных более высоких порядков. При этом также может быть использован как метод левосторонних конечных разностей, так и метод правосторонних конечных разностей и метод центральных конечных разностей. Например, формула для определения второй производной имеет следующий вид:

$$y''_1 = (y'_1)' \approx \frac{y'_2 - y'_1}{h} \approx \frac{\frac{(y_2 - y_1)}{h} - \frac{(y_1 - y_0)}{h}}{h} = \frac{y_2 - 2 \cdot y_1 + y_0}{h^2}.$$

Примеры программной реализации методов численного интегрирования и дифференцирования функции одной переменной в пакете MATLAB

Пример 1. Вычисление определенного интеграла встроенными функциями в пакете MATLAB. Для решения задачи интегрирования в пакете MATLAB имеется ряд инструментов, в частности встроенная функция *int*, которая позволяет выполнить как символьное, так и численное интегрирование заданной функции.

Например, используя функцию *int* в пакете MATLAB, выполним интегрирование следующей функции:

$$y = 0,5 \cdot \sin(x); \quad x = 0 \dots 10, \quad \Delta x = 1. \quad (21)$$

На рис. 25 приведен программный код, реализующий символьное и численное интегрирование заданной функции (21).

```

1 -      syms x
2 -      f=sym(0.5*sin(x));
3 -      FI = int(f,x);
4 -      Is = int(f,x,0,10);
5 -      I = double(Is);

```

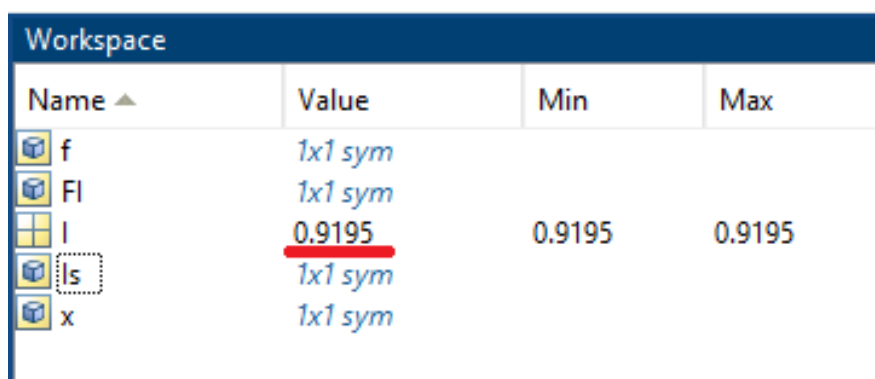
Рис. 25. Символьное и численное интегрирование в пакете MATLAB

Программный код (рис. 25) реализует следующий порядок действий. В 1-й строке объявляется символьная переменная x . Во 2-й строке задается сама исходная функция (21) как символьное выражение. В 3-й строке с помощью встроенной функции *int* выполняется символьное вычисление неопределенного интеграла. Результат представлен на рис. 26.

```
val =  
-cos(x)/2
```

Рис. 26. Результат символьного интегрирования в пакете MATLAB

В 4-й и 5-й строках рассчитывается определенный интеграл на заданном интервале от 0 до 10. Полученный результат можно наблюдать в области Workspace (рис. 27).



Name	Value	Min	Max
f	1x1 sym		
FI	1x1 sym		
I	0.9195	0.9195	0.9195
s	1x1 sym		
x	1x1 sym		

Рис. 27. Результат интегрирования в пакете MATLAB

Таким образом, определенный интеграл заданной функции (21) равен 0,9195.

Пример 2. Численное интегрирование методом прямоугольников в пакете MATLAB. Формулы метода прямоугольников (14) могут быть реализованы в виде программного кода в пакете MATLAB. На рис. 28 представлена программная реализация метода левых прямоугольников.

В 1-й строке (рис. 28) объявляется символьная переменная xs . Во 2-й строке задается сама исходная функция (21) как символьное выражение. В 3-й и 4-й строках вычисляется определенный интеграл на заданном интервале от 0 до 10, с помощью встроенной функции *int*.

В 6-й строке задается массив x (узлы функции) в соответствии с начальными условиями (21). В 7-й строке определяется количество узлов функции. В строках с 9-й по 11-ю реализована формула метода левых

прямоугольников (14). В 13-й строке производится расчет относительной погрешности интегрирования с помощью метода левых прямоугольников. Полученный результат можно наблюдать в области Workspace (рис. 29).

```

1 - syms xs
2 - f=sym(0.5*sin(xs));
3 - Is = int(f,xs,0,10);
4 - I = double(Is);
5
6 - x = 0:1:10;
7 - nbuf = size(x); n = nbuf(2);
8 - Ipr = 0;
9 - for i=1:n-1
10 -     Ipr = Ipr + (x(i+1)-x(i)) * (0.5*sin(x(i)));
11 - end
12
13 - Epr = 100*abs((Ipr - I)/I);

```

Рис. 28. Программная реализация метода левых прямоугольников в пакете MATLAB

Workspace			
Name ▲	Value	Min	Max
Epr	6.3150	6.3150	6.3150
f	1x1 sym		
i	10	10	10
I	0.9195	0.9195	0.9195
Ipr	0.9776	0.9776	0.9776
Is	1x1 sym		
n	11	11	11
nbuf	[1,11]	1	11
x	1x11 double	0	10
xs	1x1 sym		

Рис. 29. Результат численного интегрирования методом левых прямоугольников в пакете MATLAB

Как видно из полученных результатов (рис. 29), определенный интеграл заданной функции (21) равен $I = 0,9195$, а интеграл этой же функции, рассчитанный с помощью метода левых прямоугольников, – $I_{pr} = 0,9776$. Таким образом, относительная погрешность интегрирования с помощью метода левых прямоугольников составляет $E_{pr} = 6,3150\%$.

Аналогичным образом может быть выполнена программная реализация метода правых прямоугольников. Программный код для *метода правых прямоугольников* представлен на рис. 30.

В 1-й строке (рис. 30) объявляется символьная переменная xs . Во 2-й строке задается сама исходная функция (21) как символьное выражение. В 3-й и 4-й строках происходит вычисление определенного интеграла на заданном интервале от 0 до 10, с помощью встроенной функции *int*.

```

1 –      syms xs
2 –      f=sym(0.5*sin(xs));
3 –      Is = int(f,xs,0,10);
4 –      I = double(Is);
5
6 –      x = 0:1:10;
7 –      nbuf = size(x); n = nbuf(2);
8 –      Ipr = 0;
9 –      for i=2:n
10 –          Ipr = Ipr + (x(i)-x(i-1)) * (0.5*sin(x(i)));
11 –      end
12
13 –      Epr = 100*abs((Ipr - I)/I);

```

Рис. 30. Программная реализация метода правых прямоугольников в пакете MATLAB

В 6-й строке формируется массив x (узлы функции) в соответствии с начальными условиями (21). В 7-й строке определяется количество узлов функции. В строках с 9-й по 11-ю реализована формула метода правых прямоугольников (14). В 13-й строке рассчитывается относительная погрешность интегрирования с помощью метода правых прямоугольников.

Полученный результат можно наблюдать в области Workspace (рис. 31).

Как видно (рис. 31), определенный интеграл заданной функции (21) равен $I = 0,9195$, а интеграл этой же функции, рассчитанный с помощью метода правых прямоугольников, – $I_{pr} = 0,7056$. Таким образом, относительная погрешность интегрирования с помощью метода правых прямоугольников $E_{pr} = 23,2663\%$.

Workspace			
Name ▲	Value	Min	Max
Epr	23.2663	23.2663	23.2663
f	1x1 sym		
i	11	11	11
l	0.9195	0.9195	0.9195
lpr	0.7056	0.7056	0.7056
ls	1x1 sym		
n	11	11	11
nbuf	[1,11]	1	11
x	1x11 double	0	10
xs	1x1 sym		

Рис. 31. Результат численного интегрирования методом правых прямоугольников в пакете MATLAB

Аналогичным образом может быть выполнена программная реализация метода средних прямоугольников. Программный код для метода средних прямоугольников представлен на рис. 32.

```

1 - syms xs
2 - f=sym(0.5*sin(xs));
3 - Is = int(f,xs,0,10);
4 - I = double(Is);
5
6 - x = 0:1:10;
7 - nbuf = size(x); n = nbuf(2);
8 - Ipr1 = 0;
9 - Ipr2 = 0;
10 - for i=1:n-1
11 -     Ipr1 = Ipr1 + (x(i+1)-x(i)) * (0.5*sin((x(i)+x(i+1))/2));
12 - end
13
14 - for i=2:n
15 -     Ipr2 = Ipr2 + (x(i)-x(i-1)) * (0.5*sin((x(i-1)+x(i))/2));
16 - end
17
18 - Epr1 = 100*abs((Ipr1 - I)/I);
19 - Epr2 = 100*abs((Ipr2 - I)/I);

```

Рис. 32. Программная реализация метода средних прямоугольников в пакете MATLAB

В 1-й строке (рис. 32) объявляется символьная переменная *xs*. Во 2-й строке задается сама исходная функция (21) как символьное выражение.

В 3-й и 4-й строках происходит вычисление определенного интеграла на заданном интервале от 0 до 10, с помощью встроенной функции *int*.

В 6-й строке формируется массив x (узлы функции) в соответствии с начальными условиями (21). В 7-й строке определяется количество узлов функции. В строках с 10-й по 16-ю реализованы два варианта формул для метода средних прямоугольников (14). В 18-й и 19-й строках производится расчет относительной погрешности интегрирования с помощью метода средних прямоугольников. Полученный результат можно наблюдать в области Workspace (рис. 33).

Workspace			
Name ▲	Value	Min	Max
Epr1	4.2915	4.2915	4.2915
Epr2	4.2915	4.2915	4.2915
f	1x1 sym		
i	11	11	11
I	0.9195	0.9195	0.9195
Ipr1	0.9590	0.9590	0.9590
Ipr2	0.9590	0.9590	0.9590
Is	1x1 sym		
n	11	11	11
nbuf	[1,11]	1	11
x	1x11 double	0	10
xs	1x1 sym		

Рис. 33. Результат численного интегрирования методом средних прямоугольников в пакете MATLAB

Как видно (рис. 33), определенный интеграл заданной функции (21) равен $I = 0,9195$, а интеграл этой же функции, рассчитанный с помощью метода средних прямоугольников, – $Ipr1 = Ipr2 = 0,9590$. Таким образом, относительная погрешность интегрирования с помощью метода средних прямоугольников составляет $Epr1 = Epr2 = 4,2915\%$.

Можно сделать вывод о том, что метод средних прямоугольников оказался более точным по сравнению с методами левых и правых прямоугольников.

Пример 3. Численное интегрирование методом трапеций в пакете MATLAB. Формула метода трапеций (15) может быть реализована в виде программного кода в пакете MATLAB. На рис. 34 представлена программная реализация метода трапеций.

В 1-й строке (рис. 34) объявляется символьная переменная xs . Во 2-й строке задается сама исходная функция (21) как символьное выражение.

В 3-й и 4-й строках происходит вычисление определенного интеграла на заданном интервале от 0 до 10, с помощью встроенной функции *int*.

```

1 - syms xs
2 - f=sym(0.5*sin(xs));
3 - Is = int(f,xs,0,10);
4 - I = double(Is);
5
6 - x = 0:1:10;
7 - nbuf = size(x); n = nbuf(2);
8 - Itr = 0;
9 - for i=1:n-1
10 -     Itr = Itr + (x(i+1)-x(i)) * ((0.5*sin(x(i+1))+0.5*sin(x(i)))/2);
11 - end
12
13 - Etr = 100*abs((Itr - I)/I);

```

Рис. 34. Программная реализация метода трапеций в пакете MATLAB

В 6-й строке формируется массив *x* (узлы функции) в соответствии с начальными условиями (21). В 7-й строке определяется количество узлов функции. В строках с 9-й по 11-ю реализована формула метода трапеций (15). В 13-й строке производится расчет относительной погрешности интегрирования с помощью метода трапеций. Полученный результат можно наблюдать в области Workspace (рис. 35).

Workspace			
Name	Value	Min	Max
Etr	8.4756	8.4756	8.4756
f	1x1 sym		
i	10	10	10
I	0.9195	0.9195	0.9195
Is	1x1 sym		
Itr	0.8416	0.8416	0.8416
n	11	11	11
nbuf	[1,11]	1	11
x	1x11 double	0	10
xs	1x1 sym		

Рис. 35. Результат численного интегрирования методом трапеций в пакете MATLAB

Как видно (рис. 35), определенный интеграл заданной функции (21) равен $I = 0,9195$, а интеграл этой же функции, рассчитанный с помощью метода трапеций, – $I_{tr} = 0,8416$. Таким образом, относительная погрешность интегрирования с помощью метода трапеций составляет $E_{tr} = 8,4756 \%$.

Можно заметить, что в данном случае метод трапеций обеспечил менее точный результат интегрирования по сравнению с методом прямоугольников. Это объясняется видом подынтегральной функции и интервалом интегрирования.

Пример 4. Численное интегрирование методом Симпсона в пакете MATLAB. Формула метода Симпсона (16) может быть реализована в виде программного кода в пакете MATLAB (рис. 36).

```

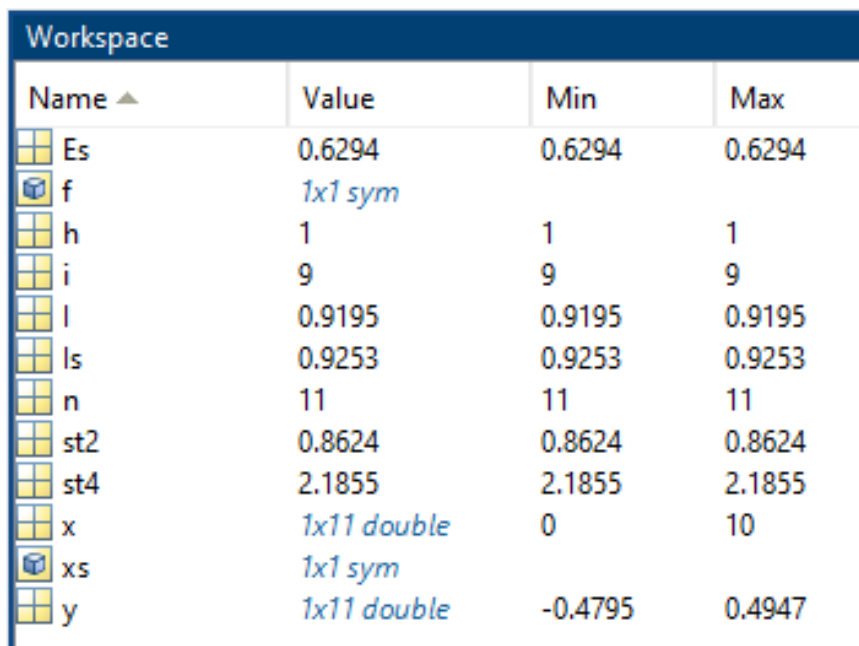
1 –      syms xs
2 –      f=sym(0.5*sin(xs));
3 –      Is = int(f,xs,0,10);
4 –      I = double(Is);
5
6 –      x = 0:1:10;
7 –      y = 0.5*sin(x);
8 –      h = x(2)-x(1);
9 –      n = length(x);
10 –     st4=0;
11 –     st2=0;
12 –     for i=2:2:n
13 –         st4=st4+4*y(i);
14 –     end
15 –     for i=3:2:n-1
16 –         st2=st2+2*y(i);
17 –     end
18 –     Is = h/3*(y(1)+st4+st2+y(n));
19 –     Es = 100*abs((I-Is)/I);

```

Рис. 36. Программная реализация метода Симпсона в пакете MATLAB

В 1-й строке (рис. 36) объявляется символьная переменная xs . Во 2-й строке задается сама исходная функция (21) как символьное выражение. В 3-й и 4-й строках происходит вычисление определенного интеграла на заданном интервале от 0 до 10, с помощью встроенной функции *int*.

В 6-й и 7-й строках формируется исходная матрица (таблица узловых точек сеточной функции) для численного интегрирования в соответствии с начальными условиями (21). В 8-й строке вычисляется шаг. В 9-й строке определяется количество узловых точек заданной функции (21). В строках с 12-й по 18-ю реализована формула метода Симпсона (16). В 19-й строке производится расчет относительной погрешности интегрирования с помощью метода Симпсона. Полученный результат можно наблюдать в области Workspace (рис. 37).



Name ▲	Value	Min	Max
Es	0.6294	0.6294	0.6294
f	1x1 sym		
h	1	1	1
i	9	9	9
l	0.9195	0.9195	0.9195
ls	0.9253	0.9253	0.9253
n	11	11	11
st2	0.8624	0.8624	0.8624
st4	2.1855	2.1855	2.1855
x	1x11 double	0	10
xs	1x1 sym		
y	1x11 double	-0.4795	0.4947

Рис. 37. Результат численного интегрирования методом Симпсона в пакете MATLAB

Как видно (рис. 37), определенный интеграл заданной функции (21) равен $I = 0,9195$, а интеграл этой же функции, рассчитанный с помощью метода Симпсона, – $Is = 0,9253$. Таким образом, относительная погрешность интегрирования с помощью метода Симпсона составляет $Es = 0,6294 \%$.

Стоит отметить, что метод Симпсона обеспечил наиболее точный результат интегрирования по сравнению с остальными методами.

Пример 5. Дифференцирование функции встроенными функциями в пакете MATLAB. Для дифференцирования функций в пакете MATLAB имеется ряд инструментов, в частности встроенная функция *diff*.

Например, используя функцию *diff* в пакете MATLAB, выполним дифференцирование следующей функции:

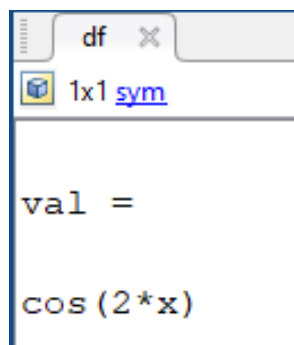
$$y = 0,5 \cdot \sin(2 \cdot x), \quad x = 1 \dots 6, \quad \Delta x = 0,2. \quad (22)$$

На рис. 38 представлен программный код, реализующий символьное дифференцирование заданной функции (22).

```
1 - syms x
2 - f = sym(0.5*sin(2*x));
3 - df = diff(f,x);
4
5 - v = symvar(f);
6 - dFa = @(X) double( subs(df,v,X));
7 - x = 1:0.2:6;
8 - dFa_x = dFa(x);
9 - plot(x,dFa_x,'r*');
10 - grid on
11 - xlabel ('X')
12 - ylabel ('Y')
```

Рис. 38. Символьное дифференцирование в пакете MATLAB

Код (рис. 38) реализует следующий порядок действий. В 1-й строке объявляется символьная переменная x . Во 2-й строке задается сама исходная функция (22) как символьное выражение. В 3-й строке с помощью встроенной функции *diff* происходит символьное вычисление производной. Полученный результат показан на рис. 39.



```
df
1x1 sym

val =

cos(2*x)
```

Рис. 39. Результат символьного дифференцирования в пакете MATLAB

В 5-й строке формируется список переменных из заданного символьного выражения, и в 6-й строке создается функция. В 7-й строке, на основании исходных данных (22), – массив узловых точек переменной x . В 8-й строке происходит расчет значений производной для заданного массива узловых точек переменной x . Полученный результат представляется графически с помощью функции *plot* (рис. 40).

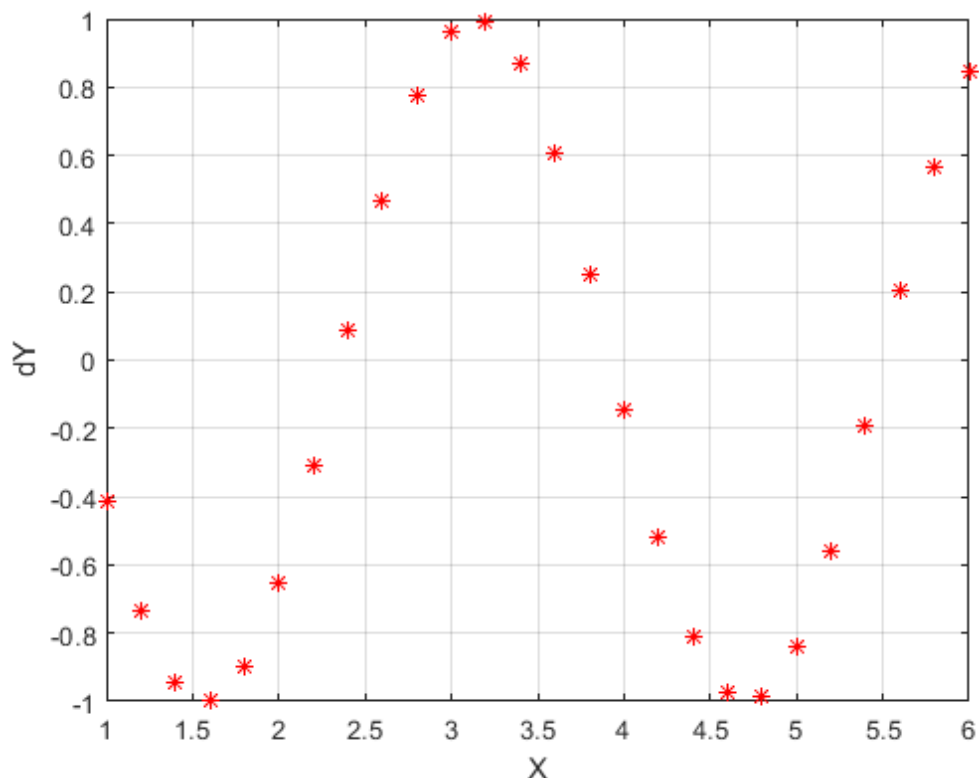


Рис. 40. Результат аналитического дифференцирования в пакете MATLAB

Пример 6. Численное дифференцирование в пакете MATLAB.

Численное дифференцирование в пакете MATLAB можно выполнить с помощью рассмотренных методов конечных разностей (17)–(20), реализовав их программно.

На рис. 41 представлена программная реализация метода левосторонних конечных разностей (17). В 15-й строке на основании исходных данных (22) формируется массив узловых точек переменной x . В 16-й строке – массив y в соответствии с исходными данными (22). В 17-й строке определяется длина массива переменной x . В 18-й строке вычисляется шаг массива переменной x . Строки с 19-й по 23-ю реализуют формулу метода левосторонних конечных разностей (17). В строках с 24-й по 31-ю рассчитывается приведенное среднеквадратичное отклонение результатов численного дифференцирования от результатов, полученных в ходе аналитического дифференцирования в пакете MATLAB.

Результаты дифференцирования представляются графически с помощью функции *plot* (рис. 42).

Как видно из (рис. 43), приведенное среднеквадратичное отклонение результатов численного дифференцирования от результатов, полученных в ходе аналитического дифференцирования, составляет 8,05 %.

```

15 - x = 1:0.2:6;
16 - y=0.5*sin(2*x);
17 - n = length(x);
18 - h = x(2) - x(1);
19 - dF1 = [];
20 - for i=2:n
21 -     dF1(i) = (y(i) - y(i-1))/h;
22 - end
23 - dFa_x(1) = dFa_x(1);
24 - delta = [];
25 - sko = 0;
26 - for i=1:n
27 -     delta(i) = dFa_x(i) - dF1(i);
28 -     sko = sko + (delta(i))^2;
29 - end
30 - sko = sqrt(sko/(n));
31 - sko_l = 100*sko/(max(dFa_x) - min(dFa_x));
32 - plot(x,dFa_x,'r*',x,dF1,'k*');
33 - grid on
34 - legend ('Результаты аналитического дифференцирования', ...
35 -         'Результаты численного дифференцирования')
36 - xlabel ('X')
37 - ylabel ('Y')

```

Рис. 41. Программная реализация метода левосторонних конечных разностей в пакете MATLAB

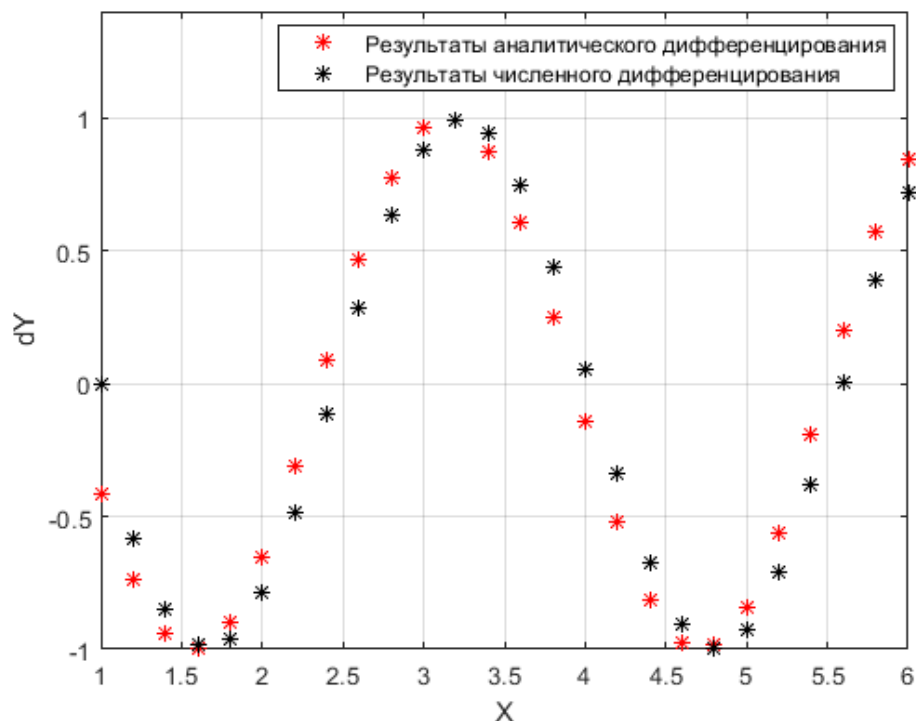


Рис. 42. Результат численного дифференцирования методом левосторонних конечных разностей в пакете MATLAB

Workspace			
Name ▲	Value	Min	Max
delta	1x26 double	-0.4161	0.1989
df	1x1 sym		
dFa	@(X)double(su...		
dFa_x	1x26 double	-0.9983	0.9932
dFl	1x26 double	-0.9930	0.9899
dFp	-0.4161	-0.4161	-0.4161
f	1x1 sym		
h	0.2000	0.2000	0.2000
i	26	26	26
n	26	26	26
sko	0.1604	0.1604	0.1604
sko_l	8.0538	8.0538	8.0538
v	1x1 sym		
x	1x26 double	1	6
y	1x26 double	-0.4981	0.4947

Рис. 43. Оценка точности численного дифференцирования методом левосторонних конечных разностей в пакете MATLAB

Аналогичным образом может быть выполнена программная реализация метода правосторонних конечных разностей (18) (рис. 44). В строке 39 на основании исходных данных (22) формируется массив узловых точек переменной x . В 40-й строке – массив y в соответствии с исходными данными (22). В 41-й строке определяется длина массива переменной x . В 42-й строке вычисляется шаг массива переменной x . Строки с 43-й по 47-ю реализуют формулу метода правосторонних конечных разностей (18). В строках с 48-й по 55-ю рассчитывается приведенное среднеквадратичное отклонение результатов численного дифференцирования от результатов, полученных в ходе аналитического дифференцирования в пакете MATLAB.

Результаты дифференцирования представляются графически с помощью функции *plot* (рис. 45).

Как видно из (рис. 46), приведенное среднеквадратичное отклонение результатов численного дифференцирования от результатов, полученных в ходе аналитического дифференцирования, составляет 6,98 %.

Аналогичным образом может быть выполнена программная реализация метода центральных конечных разностей (19) (рис. 47). В строке 63 на основании исходных данных (22) формируется массив узловых точек переменной x . В 64-й строке – массив y в соответствии с исходными данными (22). В 65-й строке определяется длина массива переменной x . В 66-й строке вычисляется шаг массива переменной x . Строки с 67-й по 72-ю реализуют формулу метода центральных конечных разностей (19). В строках

с 73-й по 80-ю рассчитывается приведенное среднеквадратичное отклонение результатов численного дифференцирования от результатов, полученных в ходе аналитического дифференцирования в пакете MATLAB.

```

39 - x = 1:0.2:6;
40 - y=0.5*sin(2*x);
41 - n = length(x);
42 - h = x(2) - x(1);
43 - dFp = [];
44 - for i=1:n-1
45 -     dFp(i) = (y(i+1) - y(i))/h;
46 - end
47 - dFp(n) = dFa_x(n);
48 - delta = [];
49 - sko = 0;
50 - for i=1:n
51 -     delta(i) = dFa_x(i) - dFp(i);
52 -     sko = sko + (delta(i))^2;
53 - end
54 - sko = sqrt(sko/(n));
55 - sko_p = 100*sko/(max(dFa_x) - min(dFa_x));
56 - plot(x,dFa_x, 'r*', x,dFp, 'k*');
57 - grid on
58 - legend ('Результаты аналитического дифференцирования', ...
59 -         'Результаты численного дифференцирования')
60 - xlabel ('X')
61 - ylabel ('Y')

```

Рис. 44. Программная реализация метода правосторонних конечных разностей в пакете MATLAB

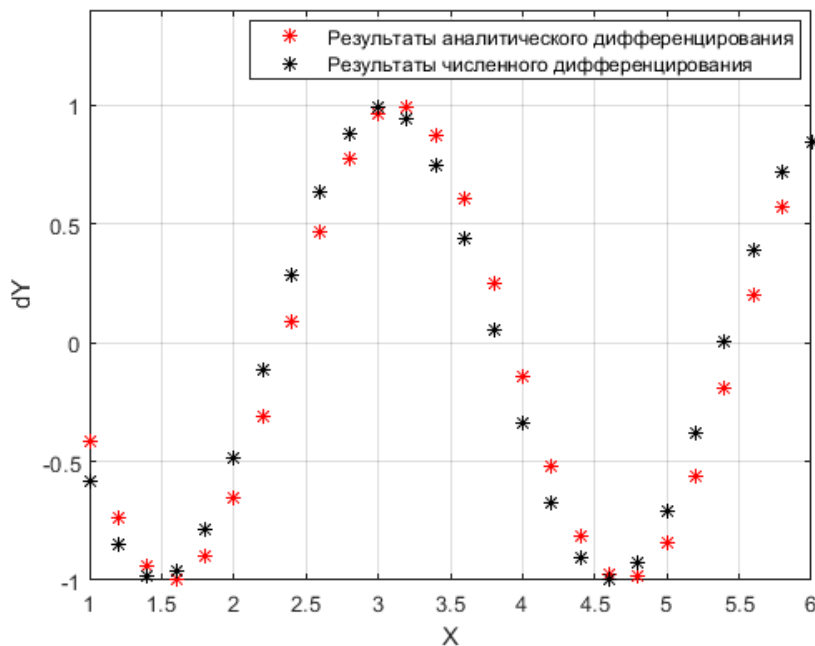


Рис. 45. Результат численного дифференцирования методом правосторонних конечных разностей в пакете MATLAB

Workspace			
Name ▲	Value	Min	Max
delta	1x26 double	-0.1987	0.1977
df	1x1 sym		
dFa	@(X)double(su...		
dFa_x	1x26 double	-0.9983	0.9932
dFI	1x26 double	-0.9930	0.9899
dFp	1x26 double	-0.9930	0.9899
f	1x1 sym		
h	0.2000	0.2000	0.2000
i	26	26	26
n	26	26	26
sko	0.1391	0.1391	0.1391
sko_l	8.0538	8.0538	8.0538
sko_p	6.9841	6.9841	6.9841
v	1x1 sym		
x	1x26 double	1	6
y	1x26 double	-0.4981	0.4947

Рис. 46. Оценка точности численного дифференцирования методом правосторонних конечных разностей в пакете MATLAB

```

63 - x = 1:0.2:6;
64 - y=0.5*sin(2*x);
65 - n = length(x);
66 - h = x(2) - x(1);
67 - dFc = [];
68 - for i=2:n-1
69 -     dFc(i) = (y(i+1) - y(i-1))/(2*h);
70 - end
71 - dFc(1) = dFa_x(1);
72 - dFc(n) = dFa_x(n);
73 - delta = [];
74 - sko = 0;
75 - for i=1:n
76 -     delta(i) = dFa_x(i) - dFc(i);
77 -     sko = sko + (delta(i))^2;
78 - end
79 - sko = sqrt(sko/(n));
80 - sko_c = 100*sko/(max(dFa_x) - min(dFa_x));
81 - plot(x,dFa_x,'r*',x,dFc,'k*');
82 - grid on
83 - legend('Результаты аналитического дифференцирования',...
84 -         'Результаты численного дифференцирования')
85 - xlabel('X')
86 - ylabel('Y')

```

Рис. 47. Программная реализация метода центральных конечных разностей в пакете MATLAB

Результаты дифференцирования представляются графически с помощью функции *plot* (рис. 48).

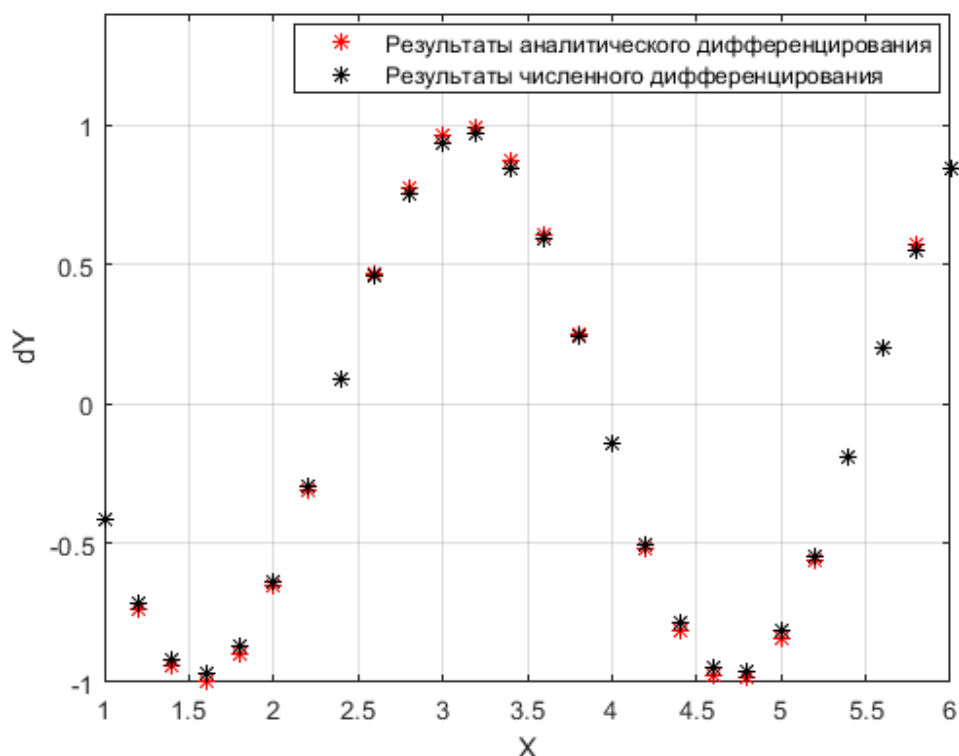


Рис. 48. Результат численного дифференцирования методом центральных конечных разностей в пакете MATLAB

Как видно (рис. 49), приведенное среднеквадратичное отклонение результатов численного дифференцирования от результатов, полученных в ходе аналитического дифференцирования, составляет 0,90 %.

Также может быть выполнена программная реализация формул численного дифференцирования более высокого порядка точности, например четвертого (20) (рис. 50). В строке 91 на основании исходных данных (22) формируется массив узловых точек переменной x . В 92-й строке – массив y в соответствии с исходными данными (22). В 93-й строке определяется длина массива переменной x . В 94-й строке вычисляется шаг массива переменной x . Строки с 95-й по 102-ю реализуют формулу метода центральных конечных разностей (20). В строках со 103-й по 110-ю рассчитывается приведенное среднеквадратичное отклонение результатов численного дифференцирования от результатов, полученных в ходе аналитического дифференцирования в пакете MATLAB.

Результаты дифференцирования представляются графически с помощью функции *plot* (рис. 51).

Workspace			
Name ▲	Value	Min	Max
delta	1x26 double	-0.0264	0.0263
df	1x1 sym		
dFa	@(X)double(su...		
dFa_x	1x26 double	-0.9983	0.9932
dFc	1x26 double	-0.9719	0.9669
dFl	1x26 double	-0.9930	0.9899
dFp	1x26 double	-0.9930	0.9899
f	1x1 sym		
h	0.2000	0.2000	0.2000
i	26	26	26
n	26	26	26
sko	0.0179	0.0179	0.0179
sko_c	0.9013	0.9013	0.9013
sko_l	8.0538	8.0538	8.0538
sko_p	6.9841	6.9841	6.9841
v	1x1 sym		
x	1x26 double	1	6
y	1x26 double	-0.4981	0.4947

Рис. 49. Оценка точности численного дифференцирования методом центральных конечных разностей в пакете MATLAB

```

91 - x = 1:0.2:6;
92 - y=0.5*sin(2*x);
93 - n = length(x);
94 - h = x(2) - x(1);
95 - dF_4 = [];
96 - for i=3:n-2
97 -     dF_4(i) = (y(i-2) - 8*y(i-1) + 8*y(i+1) - y(i+2))/(12*h);
98 - end
99 - dF_4(1) = dFa_x(1);
100 - dF_4(2) = dFa_x(2);
101 - dF_4(n-1) = dFa_x(n-1);
102 - dF_4(n) = dFa_x(n);
103 - delta = [];
104 - sko = 0;
105 - for i=1:n
106 -     delta(i) = dFa_x(i) - dF_4(i);
107 -     sko = sko + (delta(i))^2;
108 - end
109 - sko = sqrt(sko/(n));
110 - sko_4 = 100*sko/(max(dFa_x) - min(dFa_x));
111 - plot(x,dFa_x,'r*',x,dF_4,'k*');
112 - grid on
113 - legend('Результаты аналитического дифференцирования', ...
114 -        'Результаты численного дифференцирования')
115 - xlabel('X')
116 - ylabel('Y')

```

Рис. 50. Программная реализация формулы численного дифференцирования четвертого порядка точности в пакете MATLAB

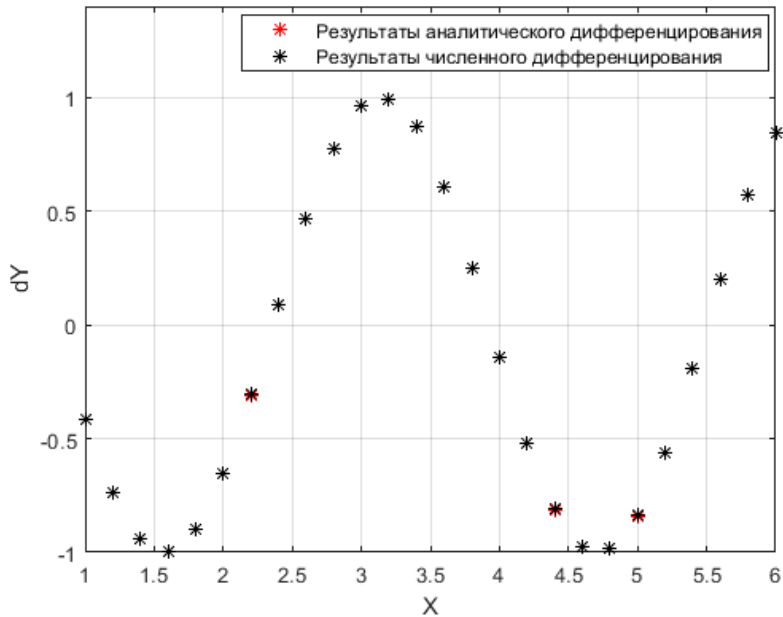


Рис. 51. Результат численного дифференцирования с помощью формулы четвертого порядка точности в пакете MATLAB

Как видно (рис. 52), приведенное среднеквадратичное отклонение результатов численного дифференцирования от результатов, полученных в ходе аналитического дифференцирования, составляет 0,03 %.

Workspace			
Name	Value	Min	Max
delta	1x26 double	-8.3580e-04	8.3152e-04
df	1x1 sym		
dF_4	1x26 double	-0.9975	0.9924
dFa	@(X)double(su...		
dFa_x	1x26 double	-0.9983	0.9932
dFc	1x26 double	-0.9719	0.9669
dFl	1x26 double	-0.9930	0.9899
dFp	1x26 double	-0.9930	0.9899
f	1x1 sym		
h	0.2000	0.2000	0.2000
i	26	26	26
n	26	26	26
sko	5.4711e-04	5.4711e-04	5.4711e-04
sko_4	0.0275	0.0275	0.0275
sko_c	0.9013	0.9013	0.9013
sko_l	8.0538	8.0538	8.0538
sko_p	6.9841	6.9841	6.9841
v	1x1 sym		
x	1x26 double	1	6
y	1x26 double	-0.4981	0.4947

Рис. 52. Оценка точности численного дифференцирования с помощью формулы четвертого порядка точности в пакете MATLAB

Как видно (рис. 52), наиболее точный результат численного дифференцирования получается в случае использования формулы четвертого порядка точности.

Индивидуальные задания

1. В соответствии с индивидуальным вариантом сгенерировать таблицу исходных данных (x_i, y_i) в пакете MATLAB.

Номер варианта	Вид функции y и значения аргумента x
1	$y = \log(x) + 0,4 \cdot \text{random}, x = 2 \dots 8, \Delta x = 1$
2	$y = e^x + 0,5 \cdot \text{random}, x = 1 \dots 5, \Delta x = 0,25$
3	$y = 1/x + \text{random}, x = 0,3 \dots 6,3, \Delta x = 1$
4	$y = x^2 + \text{random}, x = 0 \dots 20, \Delta x = 2$
5	$y = x + \text{random}, x = 10 \dots 50, \Delta x = 1$
6	$y = x + x^2 + 0,1 \cdot \text{random}, x = 0 \dots 7, \Delta x = 0,5$
7	$y = \sin(x) + 0,5 \cdot \text{random}, x = 0 \dots 2 \cdot \pi, \Delta x = \pi/5$
8	$y = (\sin(x))^2 + 0,25 \cdot \text{random}, x = \pi \dots 3 \cdot \pi, \Delta x = \pi/5$
9	$y = x^{1/2} + 0,5 \cdot \text{random}, x = 0 \dots 20, \Delta x = 2$
10	$y = \cos(x) + 0,1 \cdot \text{random}, x = 0 \dots \pi, \Delta x = \pi/20$
11	$y = \log(x) + 0,2 \cdot \text{random}, x = 5 \dots 10, \Delta x = 0,2$
12	$y = e^x - 0,2 \cdot \text{random}, x = 0 \dots 5, \Delta x = 0,5$
13	$y = x^{-1} + 0,1 \cdot \text{random}, x = 0,2 \dots 7,2, \Delta x = 0,25$
14	$y = x^2 + 2 \cdot \text{random}, x = 10 \dots 20, \Delta x = 1$
15	$y = x - \text{random}, x = 40 \dots 50, \Delta x = 0,5$
16	$y = x + x^2 - \text{random}, x = 0 \dots 4, \Delta x = 0,25$
17	$y = \sin(x) + 0,1 \cdot \text{random}, x = \pi \dots 3 \cdot \pi, \Delta x = \pi/10$
18	$y = (\sin(x))^2 - 0,6 \cdot \text{random}, x = 0 \dots \pi, \Delta x = \pi/10$
19	$y = x^{1/2} - 0,7 \cdot \text{random}, x = 2 \dots 10, \Delta x = 0,5$
20	$y = \cos(x) + \text{random}, x = 0 \dots \pi, \Delta x = \pi/10$

2. Используя встроенную функцию *int*, найти аналитическое значение определенного интеграла для функции в соответствии с индивидуальным вариантом п. 1.

3. Выполнить программную реализацию метода левых, правых и средних прямоугольников в пакете MATLAB. Вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованных методов прямоугольников. Оценить среднеквадратические отклонения результатов, полученных с помощью методов численного интегрирования, от истинного значения интеграла, рассчитанного с помощью функции *int*.

4. Увеличить Δx в 2 раза и вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью

реализованных методов прямоугольников. Оценить среднеквадратические отклонения результатов, полученных с помощью методов численного интегрирования, от истинного значения интеграла, рассчитанного с помощью функции *int*.

5. Уменьшить Δx в 2 раза и вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованных методов прямоугольников. Оценить среднеквадратические отклонения результатов, полученных с помощью методов численного интегрирования, от истинного значения интеграла, рассчитанного с помощью функции *int*.

6. Выполнить программную реализацию метода трапеций в пакете MATLAB. Вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованного метода трапеций. Оценить среднеквадратическое отклонение результата, полученного с помощью метода численного интегрирования, от истинного значения интеграла, рассчитанного с помощью функции *int*.

7. Увеличить Δx в 2 раза и вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованного метода трапеций. Оценить среднеквадратическое отклонение результата, полученного с помощью метода численного интегрирования, от истинного значения интеграла, рассчитанного с помощью функции *int*.

8. Уменьшить Δx в 2 раза и вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованного метода трапеций. Оценить среднеквадратическое отклонение результата, полученного с помощью метода численного интегрирования, от истинного значения интеграла, рассчитанного с помощью функции *int*.

9. Выполнить программную реализацию метода Симпсона в пакете MATLAB. Вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованного метода Симпсона. Оценить среднеквадратическое отклонение результата, полученного с помощью метода численного интегрирования, от истинного значения интеграла, рассчитанного с помощью функции *int*.

10. Увеличить Δx в 2 раза и вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованного метода Симпсона. Оценить среднеквадратическое отклонение результата, полученного с помощью метода численного интегрирования, от истинного значения интеграла, рассчитанного с помощью функции *int*.

11. Уменьшить Δx в 2 раза и вычислить значение интеграла для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованного метода Симпсона. Оценить среднеквадратическое отклонение результата, рассчитанного с помощью метода численного интегрирования, от истинного значения интеграла, полученного с помощью функции *int*.

12. С помощью встроенной функции *diff* выполнить символьное дифференцирование функции в соответствии с индивидуальным вариантом п. 1. Используя полученное символьное выражение, вычислить значения производной в узловых точках.

13. Выполнить программную реализацию формул численного дифференцирования первого, второго и четвертого порядков точности в пакете MATLAB. Вычислить значения производной для функции в соответствии с индивидуальным вариантом п. 1 с помощью реализованных формул. Результат представить графически, используя функции *plot*. Оценить среднеквадратические отклонения данных, полученных с помощью методов численного дифференцирования, от истинных значений производной, рассчитанных с помощью функции *diff*.

14. Выполнить п. 13, увеличив Δx в 2 раза. Сравнить полученные результаты.

15. Выполнить п. 13, уменьшив Δx в 2 раза. Сравнить полученные результаты.

16. Подготовить отчет в соответствии с требованиями, представленными в прил. А–В. Проанализировать полученные результаты и сделать выводы.

Контрольные вопросы и задания

1. Запишите формулы метода левых, правых и средних прямоугольников для численного интегрирования. Какой из методов наиболее точен и почему?
2. Запишите формулу метода трапеций для численного интегрирования.
3. Запишите формулу метода Симпсона для численного интегрирования. Как влияет шаг интегрирования на точность метода Симпсона?
4. Запишите формулы первого и второго порядка точности для численного дифференцирования.
5. Какие методы аппроксимации Вам известны?
6. В чем суть метода наименьших квадратов?

Список рекомендованной литературы

1. Зенков А.В. Численные методы : учебное пособие / А.В. Зенков. – Екатеринбург : Изд-во Урал. ун-та, 2016. – 124 с.
2. Мышенков В.И. Численные методы. Часть 1 : учебное пособие / В.И. Мышенков, Е.В. Мышенков. – Москва : МГУЛ, 2001. – 120 с.
3. Самарский А.А. Численные методы: учебное пособие для вузов / А.А. Самарский, А.В. Гулин. – Москва : Наука. Главная редакция физико-математической литературы, 1989. – 432 с.

Лабораторная работа № 3

РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ, ЗАПИСАННЫХ В ФОРМЕ КОШИ, В ПАКЕТЕ MATLAB

Цель работы: изучение основных методов для численного решения дифференциальных уравнений, записанных в форме Коши, с использованием специализированного программного обеспечения MATLAB.

Основные теоретические сведения

Задача Коши

В инженерной и научной практике достаточно широкое применение получили дифференциальные уравнения для описания протекания тех или иных процессов. В силу большой востребованности и распространённости дифференциальных уравнений было сформировано отдельное направление в математике – теория дифференциальных уравнений. Решение дифференциальных уравнений является одной из главных задач данной теории.

Как известно, имеется бесконечно много частных решений для любого дифференциального уравнения вида

$$y' = f(x, y), \quad (23)$$

где $f(x, y)$ – дифференциальное уравнение первого порядка, разрешенное относительно производной.

Для выбора единственного необходимого решения задается условие $y(x_0) = y_0$, которое называется *начальным*.

Таким образом, *задача Коши* заключается в нахождении решения для заданного дифференциального уравнения (23) при определенных начальных условиях $y(x_0) = y_0$.

Существуют разные методы решения дифференциальных уравнений, которые подразделяются на аналитические, графические и численные. В свою очередь, к численным методам относятся:

- метод Эйлера;
- усовершенствованный метод Эйлера;
- метод Рунге–Кутты;
- метод Милна;
- метод Адамса–Бэшфорта;
- метод Адамса–Мултона и др.

Метод Эйлера

Как уже было показано, в общем виде обыкновенное дифференциальное уравнение является разрешенным относительно производной (23), то есть правая часть уравнения не зависит от производной первого порядка:

$$y' = \frac{dy}{dx} = f(x, y). \quad (24)$$

Решение данного уравнения (24) сводится к нахождению значений функции y в определенных точках x , при условии, что заданы начальные условия, то есть $y(x_0) = y_0$.

Выполнив умножение левой и правой частей уравнения (24) на dx , получаем:

$$dy = f(x, y)dx. \quad (25)$$

Внешний вид выражения (25) свидетельствует о необходимости интегрирования правой и левой частей выражения между i -м и $i+1$ -м узлами сетки:

$$\int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x, y)dx;$$
$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y)dx. \quad (26)$$

В результате получаем выражение (26), которое содержит интеграл от неявно заданной функции. Нахождение такого интеграла в аналитическом виде невозможно. В данном случае актуальным становится применение численных методов решения дифференциальных уравнений.

Наиболее простым и распространенным методом является метод Эйлера. Он базируется на применении аппроксимации производной функции $\frac{dy}{dx}$ отношением конечных приращений функции Δy и аргумента Δx . Другими словами, находит свое практическое применение ранее рассмотренный метод конечных разностей для численного дифференцирования. Тогда получаем:

$$y' = \frac{dy}{dx} \approx \frac{\Delta y}{\Delta x} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f(x, y).$$

Учитывая равномерность сетки заданной функции, можно записать выражение для определения искомого значения функции в точке x_{i+1} :

$$y_{i+1} = y_i + (x_{i+1} - x_i) \cdot f(x, y) = y_i + h \cdot f(x, y). \quad (27)$$

Из выражения (27) нетрудно заметить, что для вычисления интеграла функции (26) в методе Эйлера используется метод левых прямоугольников (второе слагаемое в правой части уравнения). Полученная формула (27) является итерационной, при этом y_{i+1} может быть рассчитано, если известны y_i и x_i .

Методы Рунге–Кутты

Еще одним распространенным и более точным методом численного решения обыкновенных дифференциальных уравнений является метод Рунге–Кутты.

Как уже было отмечено в случае с методом Эйлера, для расчета интеграла функции (26) используется метод левых прямоугольников. При этом из предыдущей работы известно, что большую точность вычисления интеграла обеспечивает метод Симпсона. Соответственно, применяя формулу Симпсона в данном случае можно достичь более точного результата при численном решении обыкновенных дифференциальных уравнений.

Как известно, для применения метода Симпсона необходимо иметь значения подынтегральной функции в трех точках. В выражении (26) имеется только две точки, поэтому вводится дополнительная точка в интервале от x_i до x_{i+1} :

$$x_{i+\frac{1}{2}} = x_i + \frac{h}{2}. \quad (28)$$

С учетом (28) выражение (26) принимает следующий вид:

$$y_{i+1} = y_i + \int_{x_{i+\frac{1}{2}} - \frac{h}{2}}^{x_{i+\frac{1}{2}} + \frac{h}{2}} f(x, y) dx;$$

$$y_{i+1} = y_i + \frac{h/2}{3} \cdot [f(x_i, y_i) + 4 \cdot f(x_{i+h/2}, y_{i+h/2}) + f(x_{i+1}, y_{i+1})]. \quad (29)$$

Нетрудно заметить, что выражение (29), которое получилось в результате всех преобразований, является неявным, поскольку в правой его части имеются неизвестные значения функции, такие как $y_{i+h/2}$ и y_{i+1} . Таким образом, для дальнейшего использования формулы (29) необходимо ввести некоторые приближения для вычисления этих значений функции:

$$y_{i+1} = y_i + \frac{h}{6} \cdot [f(x_i, y_i) + 4 \cdot f(x_{i+h/2}, \tilde{y}_{i+h/2}) + f(x_{i+1}, \tilde{y}_{i+1})].$$

Выражения методов Рунге–Кутты разного порядка точности формируются в результате применения разных методов для приближенного определения величин $\tilde{y}_{i+h/2}$ и $\tilde{y}_{i+1}$.

Так, для метода Рунге–Кутты третьего порядка точности (РК-3) справедливы следующие выражения:

$$y_{i+1} = y_i + \frac{1}{6} \cdot (k_0 + 4 \cdot k_1 + k_2), \quad (30)$$

где $k_0 = h \cdot f(x_i, y_i)$; $k_1 = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_0}{2})$; $k_2 = h \cdot f(x_i + h, y_i + 2 \cdot k_1 - k_0)$.

Для данного метода погрешность вычислений составляет порядка h^3 .

Также для метода Рунге–Кутты четвертого порядка точности (РК-4) справедливы следующие выражения:

$$y_{i+1} = y_i + \frac{1}{6} \cdot (k_0 + 2 \cdot k_1 + 2 \cdot k_2 + k_3), \quad (31)$$

где $k_0 = h \cdot f(x_i, y_i)$; $k_1 = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_0}{2})$; $k_2 = h \cdot f(x_i + \frac{h}{2}, y_i + \frac{k_1}{2})$; $k_3 = h \cdot f(x_i + h, y_i + k_2)$.

Для данного метода погрешность вычислений составляет порядка h^4 .

Примеры программной реализации методов решения дифференциальных уравнений в пакете MATLAB

Пример 1. Решение дифференциальных уравнений встроенными функциями в пакете MATLAB. Используя встроенные функции пакета MATLAB, можно найти символьное, аналитическое и численное решение обыкновенного дифференциального уравнения, записанного в форме Коши.

Пусть дано уравнение

$$\frac{dy}{dx} = x^3 - 2 \cdot y, \quad y(0) = 0, \quad x \in [0, 1], \quad \Delta x = 0,1. \quad (32)$$

Программный код, реализующий нахождение символьного и аналитического решений данного дифференциального уравнения (32), представлен на рис. 53.

В 1-й строке кода объявляется символьная переменная x . Во 2-й строке формализуется заданное дифференциальное уравнение (32). В 3-й строке с помощью встроенной команды *dsolve* находится его символьное решение. Результат приведен на рис. 54.

В 5-й строке (рис. 53) задаются начальные условия для нахождения частного решения. В 6-й строке с помощью встроенной команды *dsolve* находится символьное частное решение заданного дифференциального уравнения. Результат представлен на рис. 55.

```

1 - syms y(x);
2 - enq = diff(y,x) == x^3 - 2*y;
3 - S = dsolve(enq);
4
5 - cond = y(0) == 0;
6 - Sc = dsolve(enq,cond);
7
8 - v = symvar(Sc);
9 - dya = @(X) double(subs(Sc,v,X));
10
11 - x = 0:0.1:1;
12 - dya_x = dya(x);
13 - plot(x,dya_x,'r*');
14 - grid on
15 - xlabel('X')
16 - ylabel('Y')

```

Рис. 53. Код реализации символьного и аналитического решений обыкновенного дифференциального уравнения в пакете MATLAB

```

val =
(3*x)/4 - (3*x^2)/4 + x^3/2 + C1*exp(-2*x) - 3/8

```

Рис. 54. Символьное решение обыкновенного дифференциального уравнения в пакете MATLAB

```

val =
(3*x)/4 + (3*exp(-2*x))/8 - (3*x^2)/4 + x^3/2 - 3/8

```

Рис. 55. Символьное частное решение обыкновенного дифференциального уравнения в пакете MATLAB

В 8-й строке (рис. 53) формируется список переменных, а в девятой – создается функция для аналитического решения. После этого в 11-й строке задаются интервал и шаг изменения переменной x . В 12-й строке находится аналитическое решение для заданного дифференциального уравнения. Результат представляется графически с помощью функции *plot* (рис. 56).

Кроме того, используя встроенные решатели семейства *ode* в пакете MATLAB, можно найти численное решение обыкновенного дифференциального уравнения (32) методами Рунге–Кутты 3-го и 4-го порядка точности. Программный код представлен на рис. 57.

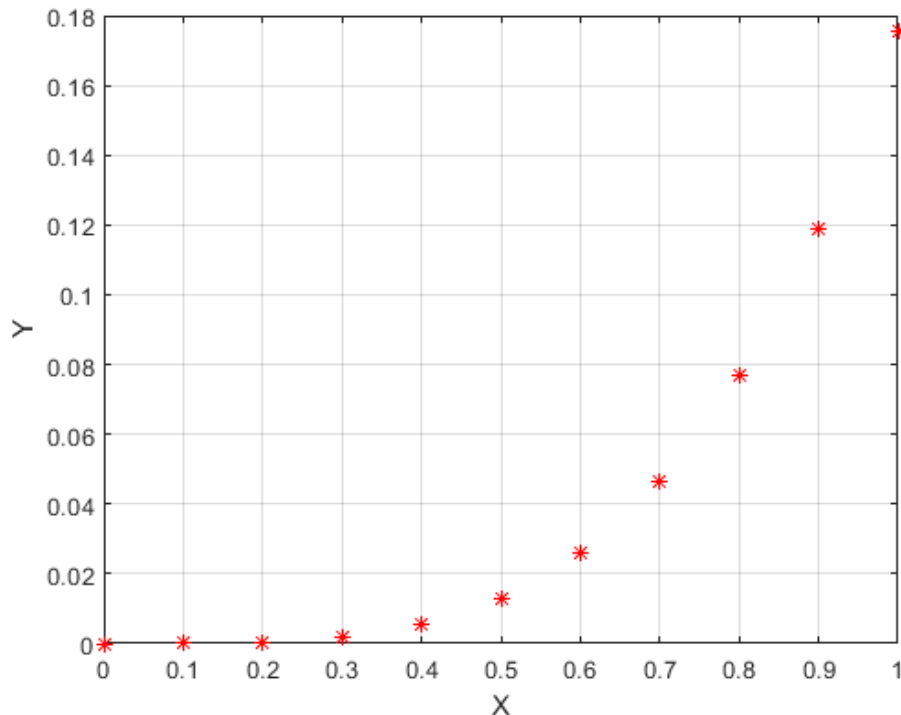


Рис. 56. Аналитическое частное решение обыкновенного дифференциального уравнения в пакете MATLAB

```

15 - x = 0:0.1:1;
16 - [X,Y] = ode23(@ (X,Y) X^3 - 2*Y, x, 0);
17 - hold on
18 - plot(X, Y, 'o')
19 - grid on
20 - xlabel ('X')
21 - ylabel ('Y')
22 - legend ('Аналитическое решение', 'Численный метод')

```

Рис. 57. Код реализации численного решения дифференциального уравнения методом Рунге–Кутты 3-го порядка точности в пакете MATLAB

Как видно (рис. 57), для нахождения численного решения дифференциального уравнения с помощью метода РК-3 требуется всего одна команда – `ode23`. В представленном коде в строке 15 задается интервал и шаг изменения переменной x . В 16-й строке находится частное численное решение заданного дифференциального уравнения (32) с помощью команды `ode23`, которая реализует метод РК-3. Формат вызова решателя `ode` выглядит следующим образом:

$$[X, Y] = \text{Решатель} \begin{pmatrix} @\text{Функция}, \\ [\text{Интервал и шаг изменения переменной}], \\ \text{Начальное условие} \end{pmatrix}.$$

Полученный результат представляется графически с помощью функции *plot* (рис. 58).

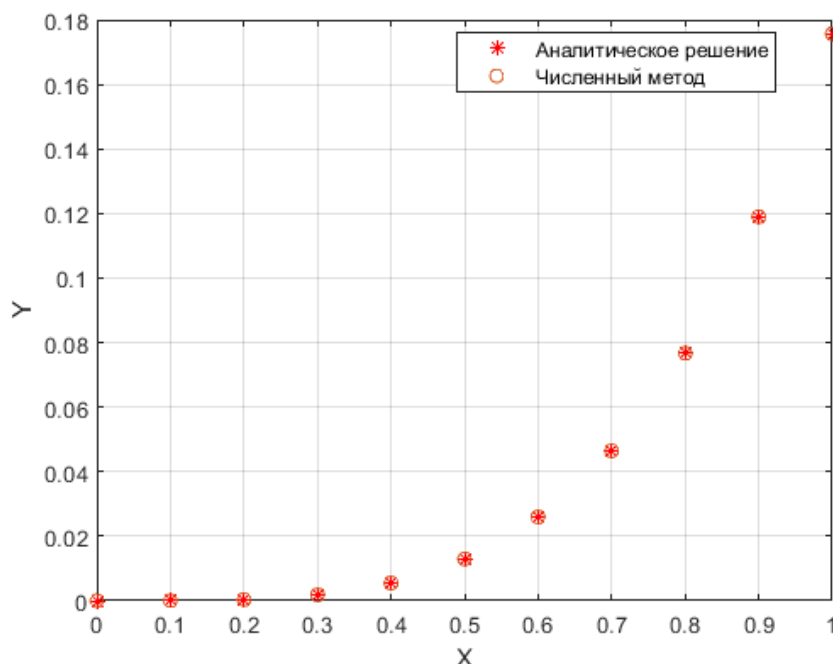


Рис. 58. Аналитическое и численное частное решение обыкновенного дифференциального уравнения методом Рунге–Кутты 3-го порядка точности в пакете MATLAB

Аналогичным образом может быть выполнена программная реализация нахождения численного решения данного дифференциального уравнения (32) с помощью метода РК-4 (рис. 59).

Как видно, код практически идентичен, за исключением того, что изменился решатель, реализующий метод Рунге–Кутты. В данном случае для нахождения численного решения дифференциального уравнения с помощью метода РК-4 используется *ode45*. Формат вызова этой функции аналогичен. Полученный результат представляется графически с помощью функции *plot* (рис. 60).

```

24 - x = 0:0.1:1;
25 - [X,Y] = ode45(@(X,Y) X^3 - 2*Y, x, 0);
26 - hold on
27 - plot(X, Y, 'o')
28 - grid on
29 - xlabel ('X')
30 - ylabel ('Y')
31 - legend ('Аналитическое решение', 'Численный метод')

```

Рис. 59. Код реализации численного решения дифференциального уравнения методом Рунге–Кутты 4-го порядка точности в пакете MATLAB

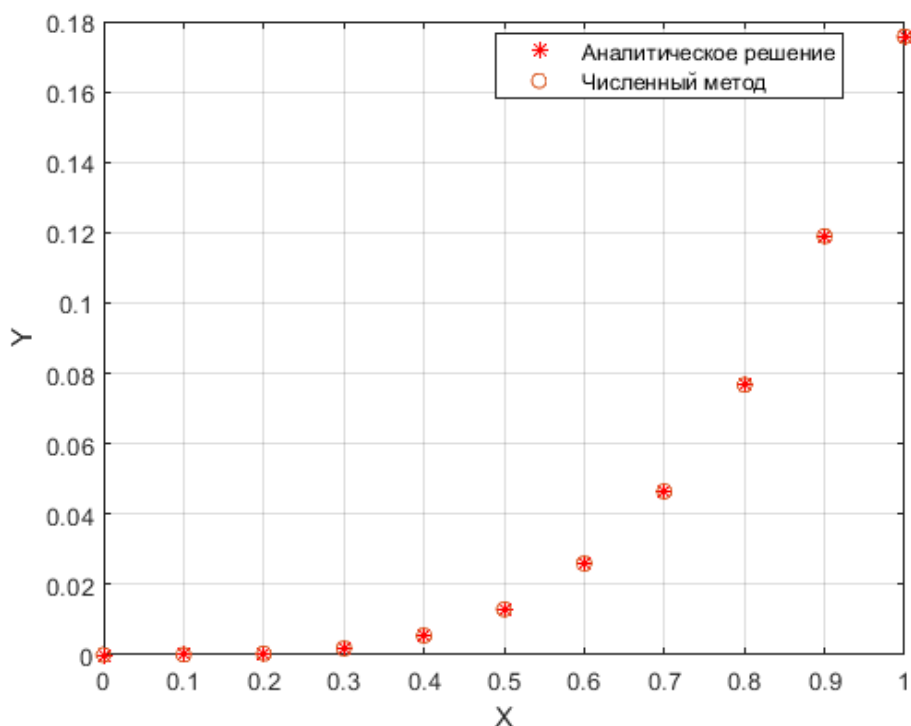


Рис. 60. Аналитическое и численное частное решение обыкновенного дифференциального уравнения методом Рунге–Кутты 4-го порядка точности в пакете MATLAB

Пример 2. Решение дифференциальных уравнений методом Эйлера в пакете MATLAB. Численное решение дифференциальных уравнений в пакете MATLAB можно выполнить с помощью рассмотренных методов, реализовав их программно.

На рис. 61 представлена программная реализация метода Эйлера (27). В строках с 21-й по 23-ю в соответствии с заданием (32) формируется функция, задаются начальные условия, область и шаг изменения переменной x . В 24-й строке определяется длина массива переменной x , а в 25-й – шаг изменения переменной x . Строки с 27-й по 31-ю реализуют формулу метода Эйлера (27). В строках с 33-й по 40-ю рассчитывается приведенное среднеквадратичное отклонение результатов численного решения дифференциального уравнения от результатов, полученных в ходе аналитического решения дифференциального уравнения в пакете MATLAB.

Результаты представляются графически с помощью функции *plot* (рис. 62).

Как видно (рис. 63), приведенное среднеквадратичное отклонение результатов численного решения дифференциального уравнения от результатов, полученных в ходе аналитического решения дифференциального уравнения, составляет 5,41 %.

```

21 - f = @(x1,y1) x1^3 - 2*y1;
22 - y0 = 0;
23 - x = 0:0.1:1;
24 - n = length(x);
25 - h = x(2) - x(1);
26
27 - dye_x = [];
28 - dye_x(1) = y0;
29 - for i=1:n-1
30 -     dye_x(i+1) = dye_x(i) + h*f(x(i),dye_x(i));
31 - end
32
33 - delta = [];
34 - sko = 0;
35 - for i=1:n
36 -     delta(i) = dya_x(i) - dye_x(i);
37 -     sko = sko + (delta(i))^2;
38 - end
39 - sko = sqrt(sko/(n));
40 - sko = 100*sko/(max(dya_x) - min(dya_x));
41
42 - plot(x,dya_x,'r*',x,dye_x,'k*');
43 - grid on
44 - xlabel('X')
45 - ylabel('Y')
46 - legend('Аналитическое решение', 'Метод Эйлера')

```

Рис. 61. Программная реализация метода Эйлера в пакете MATLAB

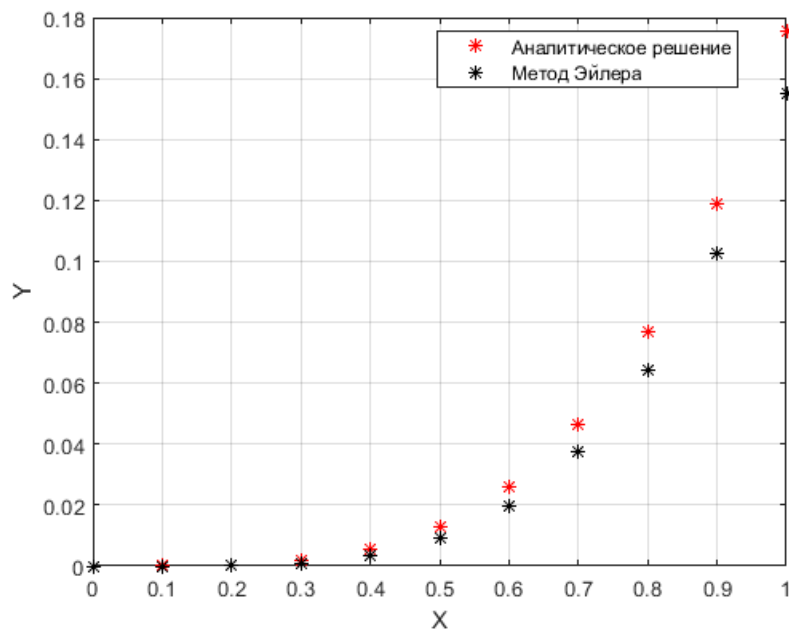


Рис. 62. Результат численного решения дифференциального уравнения с помощью метода Эйлера в пакете MATLAB

Workspace			
Name ▲	Value	Min	Max
cond	1x1 sym		
delta	1x11 double	0	0.0208
dya	@(X)double(su...		
dya_x	1x11 double	0	0.1758
dye_x	1x11 double	0	0.1550
enq	1x1 symfun		
f	@(x1,y1)x1^3-...		
h	0.1000	0.1000	0.1000
i	11	11	11
n	11	11	11
S	1x1 sym		
Sc	1x1 sym		
sko	5.4104	5.4104	5.4104
v	1x1 sym		
x	1x11 double	0	1
y	1x1 symfun		
y0	0	0	0

Рис. 63. Оценка точности численного решения дифференциального уравнения методом Эйлера в пакете MATLAB

Пример 3. Решение дифференциальных уравнений методами Рунге–Кутты в пакете MATLAB. Еще одним достаточно распространённым методом численного решения дифференциальных уравнений является метод Рунге–Кутты.

На рис. 64 представлена программная реализация метода Рунге–Кутты 3-го порядка точности (30).

В строках с 18-й по 20-ю в соответствии с заданием (32) формируется функция, задаются начальные условия, область и шаг изменения переменной x . В 21-й строке определяется длина массива переменной x , а в 22-й – шаг изменения переменной x . Строки с 24-й по 31-ю реализуют формулы метода РК-3 (30). В строках с 33-й по 40-ю рассчитывается приведенное среднеквадратичное отклонение (СКО) результатов численного решения дифференциального уравнения от результатов, полученных в ходе аналитического решения дифференциального уравнения в пакете MATLAB. Результаты представляются графически с помощью функции *plot* (рис. 65).

Как видно (рис. 66), приведенное СКО результатов численного решения дифференциального уравнения от результатов, полученных в ходе аналитического решения дифференциального уравнения, составляет 0,02 %.

```

18 - f = @(x1,y1) x1^3 - 2*y1;
19 - y0 = 0;
20 - x = 0:0.1:1;
21 - n = length(x);
22 - h = x(2) - x(1);
23
24 - dyr_x = [];
25 - dyr_x(1) = y0;
26 - for i=1:n-1
27 -     k0 = h*f(x(i),dyr_x(i));
28 -     k1 = h*f(x(i)+h/2,dyr_x(i)+k0/2);
29 -     k2 = h*f(x(i)+h,dyr_x(i)+2*k1-k0);
30 -     dyr_x(i+1) = dyr_x(i) + (1/6)*(k0 + 4*k1 + k2);
31 - end
32
33 - delta = [];
34 - sko_r = 0;
35 - for i=1:n
36 -     delta(i) = dya_x(i) - dyr_x(i);
37 -     sko_r = sko_r + (delta(i))^2;
38 - end
39 - sko_r = sqrt(sko_r/(n));
40 - sko_r = 100*sko_r/(max(dya_x) - min(dya_x));
41
42 - plot(x,dya_x,'r*',x,dyr_x,'k*');
43 - grid on
44 - xlabel ('X')
45 - ylabel ('Y')
46 - legend ('Аналитическое решение', 'Метод Рунге-Кутты 3-го порядка')

```

Рис. 64. Программная реализация метода Рунге-Кутты 3-го порядка точности в пакете MATLAB

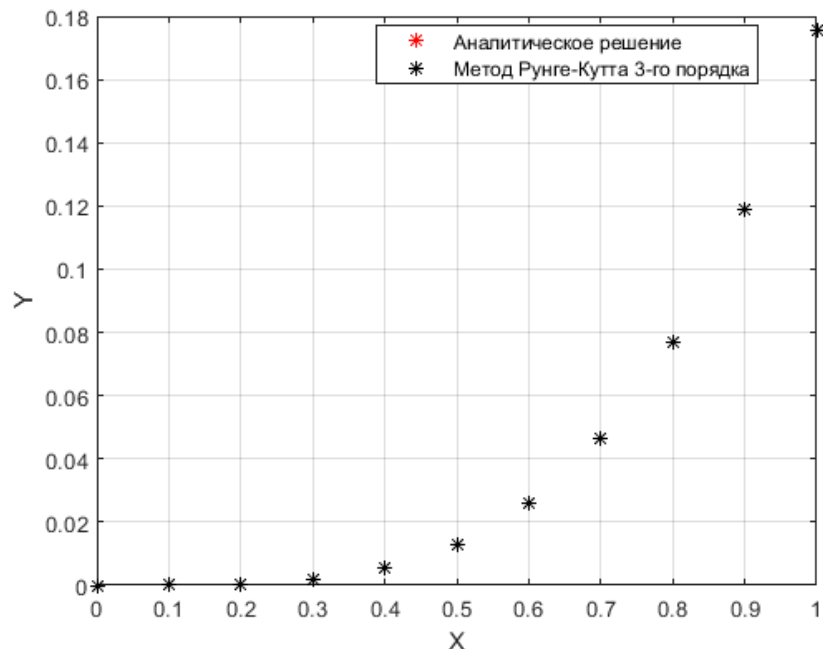


Рис. 65. Результат численного решения дифференциального уравнения с помощью метода Рунге-Кутты 3-го порядка точности в пакете MATLAB

Workspace			
Name ▲	Value	Min	Max
cond	1x1 sym		
delta	1x11 double	-1.3426e-07	6.8700e-05
dya	@(X)double(su...		
dya_x	1x11 double	0	0.1758
dya_r	1x11 double	0	0.1757
enq	1x1 symfun		
f	@(x1,y1)x1^3-...		
h	0.1000	0.1000	0.1000
i	11	11	11
k0	0.0491	0.0491	0.0491
k1	0.0570	0.0570	0.0570
k2	0.0632	0.0632	0.0632
n	11	11	11
S	1x1 sym		
Sc	1x1 sym		
sko_r	0.0176	0.0176	0.0176
v	1x1 sym		
x	1x11 double	0	1
y	1x1 symfun		
y0	0	0	0

Рис. 66. Оценка точности численного решения дифференциального уравнения методом Рунге–Кутты 3-го порядка точности в пакете MATLAB

Аналогичным образом была выполнена программная реализация метода Рунге–Кутты 4-го порядка точности (31) (рис. 67).

Программный код аналогичен. В строках с 18-й по 20-ю в соответствии с заданием (32) формируется функция, задаются начальные условия, область и шаг изменения переменной x . В 21-й строке определяется длина массива переменной x , а в 22-й вычисляется шаг изменения переменной x . Строки с 24-й по 32-ю реализуют формулы метода РК-4 (31). В строках с 34-й по 41-ю реализован расчет приведенного СКО результатов численного решения дифференциального уравнения от результатов, полученных в ходе аналитического решения дифференциального уравнения в пакете MATLAB.

Результаты представляются графически с помощью функции *plot* (рис. 68).

Как видно (рис. 69), приведенное СКО результатов численного решения дифференциального уравнения от результатов, полученных в результате аналитического решения дифференциального уравнения, составляет 0,001 %.

```

18 - f = @(x1,y1) x1^3 - 2*y1;
19 - y0 = 0;
20 - x = 0:0.1:1;
21 - n = length(x);
22 - h = x(2) - x(1);
23
24 - dyr_x = [];
25 - dyr_x(1) = y0;
26 - for i=1:n-1
27 -     k0 = h*f(x(i),dyr_x(i));
28 -     k1 = h*f(x(i)+h/2,dyr_x(i)+k0/2);
29 -     k2 = h*f(x(i)+h/2,dyr_x(i)+k1/2);
30 -     k3 = h*f(x(i)+h,dyr_x(i)+k2);
31 -     dyr_x(i+1) = dyr_x(i) + (1/6)*(k0 + 2*k1 + 2*k2 + k3);
32 - end
33
34 - delta = [];
35 - sko_r = 0;
36 - for i=1:n
37 -     delta(i) = dya_x(i) - dyr_x(i);
38 -     sko_r = sko_r + (delta(i))^2;
39 - end
40 - sko_r = sqrt(sko_r/(n));
41 - sko_r = 100*sko_r/(max(dya_x) - min(dya_x));
42
43 - plot(x,dya_x,'r*',x,dyr_x,'k*');
44 - grid on
45 - xlabel ('X')
46 - ylabel ('Y')
47 - legend ('Аналитическое решение', 'Метод Рунге-Кутты 4-го порядка')

```

Рис. 67. Программная реализация метода Рунге-Кутты 4-го порядка точности в пакете MATLAB

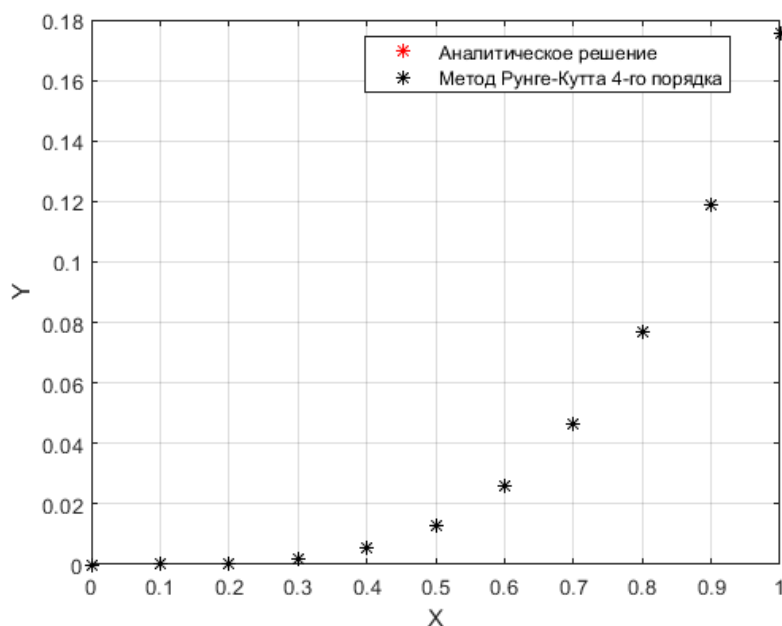


Рис. 68. Результат численного решения дифференциального уравнения с помощью метода Рунге-Кутты 4-го порядка точности в пакете MATLAB

Workspace			
Name ▲	Value	Min	Max
cond	1x1 sym		
delta	1x11 double	-5.0361e-06	0
dya	@(X)double(su...		
dya_x	1x11 double	0	0.1758
dyr_x	1x11 double	0	0.1758
enq	1x1 symfun		
f	@(x1,y1)x1^3-...		
h	0.1000	0.1000	0.1000
i	11	11	11
k0	0.0491	0.0491	0.0491
k1	0.0570	0.0570	0.0570
k2	0.0562	0.0562	0.0562
k3	0.0650	0.0650	0.0650
n	11	11	11
S	1x1 sym		
Sc	1x1 sym		
sko_r	0.0014	0.0014	0.0014
v	1x1 sym		
x	1x11 double	0	1
y	1x1 symfun		
y0	0	0	0

Рис. 69. Оценка точности численного решения дифференциального уравнения методом Рунге–Кутты 4-го порядка точности в пакете MATLAB

Индивидуальные задания

1. В соответствии с индивидуальным вариантом найти символьное и аналитическое решения дифференциального уравнения с заданными начальными условиями и на определенном отрезке, используя встроенные функции пакета MATLAB. Полученное решение представить графически.

Номер варианта	Дифференциальное уравнение в форме Коши
1	$y' + 4 \cdot y = x^2, y(0) = 0, x \in [0, 1]$
2	$y' + 4 \cdot \cos(y) = x^2, y(0) = 0, x \in [0, 1]$
3	$y' + 7 \cdot \sin(y) = x^2, y(0) = 0, x \in [0, 6]$
4	$y' - \sin\left(\frac{y}{2}\right) = \frac{x}{5}, y(0) = 0, x \in [0, 7]$
5	$y' + \exp(2 \cdot y) + x = x^2, y(0) = 0, x \in [0, 4]$
6	$y' - y^2 + y = x^3 - x^2, y(0) = 0, x \in [0, 2]$
7	$y' + y + x^4 = x^3, y(0) = 1, x \in [0, 2]$
8	$y' - y = \cos(2 \cdot x), y(0) = 3, x \in [0, 12]$
9	$y' + y = \sin\left(\frac{x}{2}\right), y(0) = 0, x \in [0, 20]$

Номер варианта	Дифференциальное уравнение в форме Коши
10	$y' + \cos(y) - y^2 = \sin(x), y(0) = 0, x \in [0, 6]$
11	$y' - \cos(x) + y^3 = \sin(x), y(0) = 0, x \in [0, 5]$
12	$y' + y = -x^3 + x^2 - x, y(0) = 0, x \in [0, 1]$
13	$y' + \sin(y) = x^5 + \sqrt{x} - 3 \cdot x, y(0) = 0, x \in [0, 1]$
14	$y' + \sin(y) + y^2 = x^3 - \sqrt{x} + y, y(0) = 1, x \in [0, 4]$
15	$y' + \cos(y) + y^3 = x^3 + x + y, y(0) = 1, x \in [0, 3]$
16	$y' + y - y^3 = x, y(0) = 4, x \in [0, 10]$
17	$y' + y^2 = \frac{x}{5} + \cos(y), y(0) = 3, x \in [0, 6]$
18	$y' + y^2 = \frac{x}{7} + \sin(y), y(0) = 1, x \in [0, 5]$
19	$y' - y = x, y(0) = 5, x \in [0, 1]$
20	$y' - \sin(y) = x + y, y(0) = 5, x \in [0, 1]$

2. Выполнив программную реализацию метода Эйлера, найти численное решение дифференциального уравнения в соответствии с индивидуальным вариантом при $h = 0,1$ и $h = 0,01$. Полученное решение представить графически.

3. Выполнив программную реализацию метода Рунге–Кутты 3-го порядка, найти численное решение дифференциального уравнения в соответствии с индивидуальным вариантом при $h = 0,1$ и $h = 0,01$. Затем, используя стандартный решатель пакета MATLAB (на базе метода Рунге–Кутты 3-го порядка), найти численное решение дифференциального уравнения в соответствии с индивидуальным вариантом при $h = 0,1$ и $h = 0,01$. Полученные результаты сравнить и представить графически.

4. Выполнив программную реализацию метода Рунге–Кутты 4-го порядка, найти численное решение дифференциального уравнения в соответствии с индивидуальным вариантом при $h = 0,1$ и $h = 0,01$. Затем, используя стандартный решатель пакета MATLAB (на базе метода Рунге–Кутты 4-го порядка), найти численное решение дифференциального уравнения в соответствии с индивидуальным вариантом при $h = 0,1$ и $h = 0,01$. Результаты сравнить и представить графически.

5. Для результатов, полученных в пп. 2–4, определить погрешность, используя формулу приведенного среднеквадратичного отклонения.

6. Подготовить отчет в соответствии с требованиями, представленными в прил. А–В. Проанализировать полученные результаты и сделать выводы.

Контрольные вопросы и задания

1. В чем заключается суть задачи Коши?
2. Перечислите наиболее распространенные методы численного решения обыкновенных дифференциальных уравнений.
3. Запишите формулу метода Эйлера для решения обыкновенных дифференциальных уравнений.
4. Перечислите встроенные решатели пакета MATLAB, реализующие методы Рунге–Кутты для решения обыкновенных дифференциальных уравнений.
5. Если сравнить метод Эйлера и метод Рунге–Кутты, то какой метод численного решения обыкновенных дифференциальных уравнений является более точным? Почему?

Список рекомендованной литературы

1. Антоник В.Г. Численные методы: математический анализ и дифференциальные уравнения : учебно-методическое пособие / В.Г. Антоник. – Иркутск : Изд-во Иркутского университета, 2007. – 63 с.
2. Милн В.Э. Численное решение дифференциальных уравнений / В.Э. Милн ; пер. с англ. и под ред. М.Р. Шура-Бура. – Москва : Изд-во иностранной литературы, 1955. – 290 с.
3. Кацман Ю.Я. Прикладная математика. Численные методы : учебное пособие / Ю.Я. Кацман. – Томск : Изд-во ТПУ, 2000. – 68 с.

Лабораторная работа № 4

ОПРЕДЕЛЕНИЕ МЕТОДОМ КОНЕЧНЫХ РАЗНОСТЕЙ СТАЦИОНАРНОГО ПРОФИЛЯ ТЕМПЕРАТУРЫ В ПЛАСТИНЕ

Цель работы: изучение основных методов для численного решения дифференциальных уравнений в частных производных, с использованием специализированного программного обеспечения MATLAB.

Основные теоретические сведения

Для решения задач математического моделирования физических процессов как в инженерной, так и в научной практике нашли широкое применение дифференциальные уравнения в частных производных. Отличительной особенностью уравнений в частных производных является зависимость функции не от одной переменной, а от нескольких, например когда рассматриваемая величина изменяется не только во времени, но и по координатам.

В большинстве случаев для решения дифференциальных уравнений в частных производных применяют численные методы. Наибольшее распространение получили метод конечных элементов (МКЭ) и метод конечных разностей (МКР). Данные методы нашли свое применение практически во всех специализированных программных продуктах, используемых для моделирования и проектирования сложных систем и механизмов.

Отличительными особенностями *метода конечных разностей* являются простота, универсальность и эффективность для решения целого ряда задач. Основная идея МКР заключается в том, что исходная задача с граничными условиями трансформируется в более простую, которая, как правило, заключается в решении системы алгебраических уравнений (САУ).

Выделяют три основных этапа МКР:

- 1) на основании исходных данных (граничные условия) формируется сетка узловых значений определяемой функции с заданным шагом в конкретной области;
- 2) на основании исходных данных, а именно заданного дифференциального уравнения в частных производных, формируется система конечно-разностных уравнений, которая описывает взаимные связи соседних узлов сетки, сформированной на первом этапе;
- 3) используя известные и доступные численные методы, производится решение сформированной системы конечно-разностных уравнений.

Далее стоит рассмотреть более детально перечисленные этапы метода конечных разностей.

Как уже было отмечено, построение сетки узловых значений функции $(f(x, y))$ происходит на основании заданных граничных условий (рис. 70).

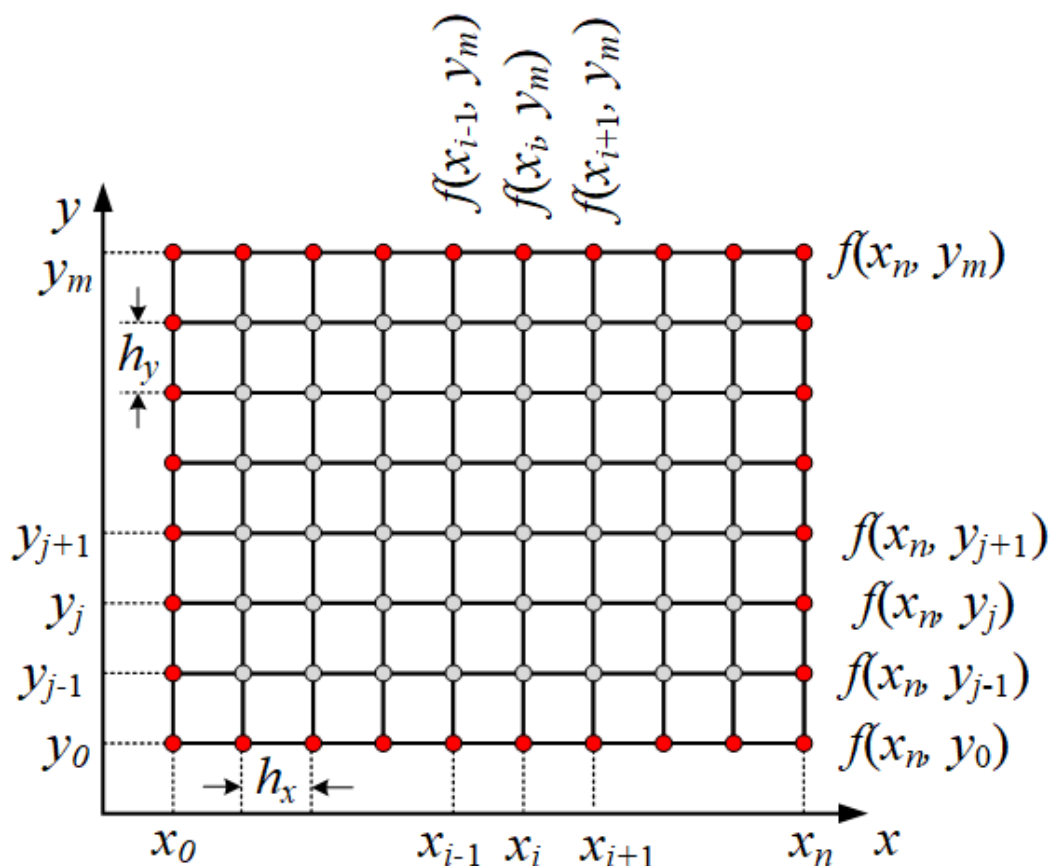


Рис. 70. Пример сетки в заданной двумерной области

Другими словами, изначально задаются массив значений функции $(f(x_i, y_j))$ в граничных узлах сетки и шаг сетки (h_x и h_y). При этом шаг сетки по переменной x (h_x) может быть не равен шагу сетки по переменной y (h_y).

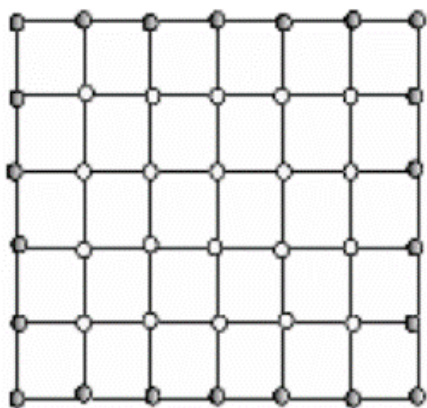
Для вычисления изменений переменных x и y могут быть использованы следующие выражения:

$$x_i = x_0 + i \cdot h_x, \quad i = 0, 1, \dots, n;$$

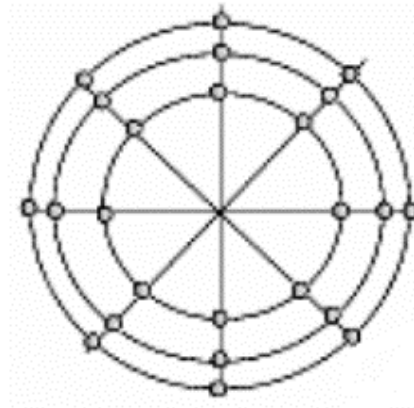
$$y_j = y_0 + j \cdot h_y, \quad j = 0, 1, \dots, m.$$

Представленный пример построения сетки является наиболее распространённым на практике, однако существуют и другие подходы. В зависимости от моделируемого процесса или объекта сетка может

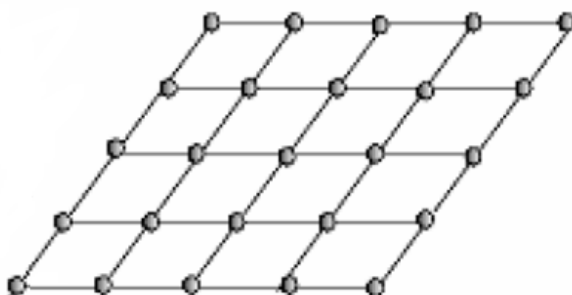
быть построена не в декартовой системе координат, а, например, в полярной, тогда меняется и вид используемой сетки. На сегодняшний день известно порядка 40 разных видов сеток. На рис. 71 представлены наиболее распространенные на практике виды сеток.



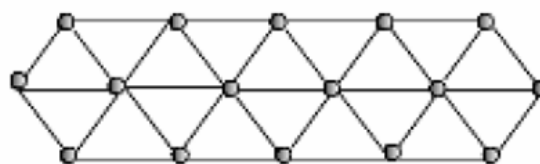
прямоугольная сетка



полярная сетка



скошенная сетка



треугольная сетка

Рис. 71. Существующие виды сеток

Следующим этапом МКР является составление системы конечно-разностных уравнений с учетом заданного дифференциального уравнения в частных производных. Как известно, производную функции можно представить в виде конечных разностей, в том числе и частную производную функции:

$$\frac{\partial f(x, y)}{\partial x} \approx \frac{f(x_i + h_x, y_j) - f(x_i, y_j)}{h_x}.$$

Выделяют левые, правые и центральные конечные разности, в зависимости от того значения функции, в каких узловых точках применяются для расчета производной.

Для вычисления частных производных функции ($f(x, y)$) методом левых конечных разностей могут быть использованы следующие выражения:

$$\begin{aligned}
\frac{\partial f(x, y)}{\partial x} &\approx \frac{f(x_i, y_j) - f(x_i - h_x, y_j)}{h_x}; \\
\frac{\partial f(x, y)}{\partial y} &\approx \frac{f(x_i, y_j) - f(x_i, y_j - h_y)}{h_y}; \\
\frac{\partial^2 f(x, y)}{\partial x^2} &\approx \frac{f(x_i, y_j) - 2 \cdot f(x_i - h_x, y_j) + f(x_i - 2 \cdot h_x, y_j)}{h_x^2}; \\
\frac{\partial^2 f(x, y)}{\partial y^2} &\approx \frac{f(x_i, y_j) - 2 \cdot f(x_i, y_j - h_y) + f(x_i, y_j - 2 \cdot h_y)}{h_y^2}; \\
\frac{\partial^2 f(x, y)}{\partial x \partial y} &\approx \frac{f(x_i, y_j) - f(x_i, y_j - h) - f(x_i - h, y_j) + f(x_i - h, y_j - h)}{h_x \cdot h_y}.
\end{aligned}$$

Для вычисления частных производных функции ($f(x, y)$) методом правых конечных разностей могут быть использованы следующие выражения:

$$\begin{aligned}
\frac{\partial f(x, y)}{\partial x} &\approx \frac{f(x_i + h_x, y_j) - f(x_i, y_j)}{h_x}; \\
\frac{\partial f(x, y)}{\partial y} &\approx \frac{f(x_i, y_j + h_y) - f(x_i, y_j)}{h_y}; \\
\frac{\partial^2 f(x, y)}{\partial x^2} &\approx \frac{f(x_i + 2 \cdot h_x, y_j) - 2 \cdot f(x_i + h_x, y_j) + f(x_i, y_j)}{h_x^2}; \\
\frac{\partial^2 f(x, y)}{\partial y^2} &\approx \frac{f(x_i, y_j + 2 \cdot h_y) - 2 \cdot f(x_i, y_j + h_y) + f(x_i, y_j)}{h_y^2}; \\
\frac{\partial^2 f(x, y)}{\partial x \partial y} &\approx \frac{f(x_i + h_x, y_j + h_y) - f(x_i, y_j + h_y) - f(x_i + h_x, y_j) + f(x_i, y_j)}{h_x \cdot h_y}.
\end{aligned}$$

Для вычисления частных производных функции ($f(x, y)$) методом центральных конечных разностей могут быть использованы следующие выражения:

$$\begin{aligned}
\frac{\partial f(x, y)}{\partial x} &\approx \frac{f(x_i + h_x, y_j) - f(x_i - h_x, y_j)}{2 \cdot h_x}; \\
\frac{\partial f(x, y)}{\partial y} &\approx \frac{f(x_i, y_j + h_y) - f(x_i, y_j - h_y)}{2 \cdot h_y}; \\
\frac{\partial^2 f(x, y)}{\partial x^2} &\approx \frac{f(x_i + h_x, y_j) - 2 \cdot f(x_i, y_j) + f(x_i - h_x, y_j)}{h_x^2};
\end{aligned}$$

$$\frac{\partial^2 f(x, y)}{\partial y^2} \approx \frac{f(x_i, y_j + h_y) - 2 \cdot f(x_i, y_j) + f(x_i, y_j - h_y)}{h_y^2};$$

$$\frac{\partial^2 f(x, y)}{\partial x \partial y} \approx \frac{f(x_i + h_x, y_j + h_y) - f(x_i + h_x, y_j - h_y) - f(x_i - h_x, y_j + h_y) + f(x_i - h_x, y_j - h_y)}{4 \cdot h_x \cdot h_y}.$$

Для преобразования заданного дифференциального уравнения в частных производных в конечно-разностную форму необходимо подставить в него один из рассмотренных типов конечно-разностных уравнений для каждого неизвестного узла сформированной сетки.

На последнем, третьем, этапе метода конечных разностей выполняют решение сформированной системы конечно-разностных уравнений, используя известные и доступные численные методы. В частности, для решения полученной системы алгебраических уравнений могут быть применены метод Ньютона, метод Зейделя, метод простых итераций или др.

Примеры программной реализации методов численного решения дифференциальных уравнений в частных производных в пакете MATLAB

Пример 1. Решение дифференциальных уравнений в частных производных методом конечных разностей в пакете MATLAB. Как правило, дифференциальные уравнения в частных производных используются для описания физического процесса, изменяющегося в пространстве с течением времени. В рамках данной лабораторной работы рассматривается только пространственное изменение зависимой функции. В качестве примера предлагается рассмотреть определение стационарного температурного поля в пластине, используя метод конечных разностей.

Допустим, имеется пластина следующих размеров: ширина (координата – x) равна 4 метрам и длина (координата – y) – 4 м. При этом известна температура ($T(x, y)$) в следующих точках этой пластины:

$$\begin{array}{llll} T(0,0) = 0 \text{ } ^\circ\text{C}; & T(1,0) = 25 \text{ } ^\circ\text{C}; & T(2,0) = 50 \text{ } ^\circ\text{C}; & T(3,0) = 75 \text{ } ^\circ\text{C}; & T(4,0) = 100 \text{ } ^\circ\text{C}; \\ T(0,1) = 0 \text{ } ^\circ\text{C}; & & & & T(4,1) = 100 \text{ } ^\circ\text{C}; \\ T(0,2) = 0 \text{ } ^\circ\text{C}; & & & & T(4,2) = 100 \text{ } ^\circ\text{C}; \\ T(0,3) = 0 \text{ } ^\circ\text{C}; & & & & T(4,3) = 100 \text{ } ^\circ\text{C}; \\ T(0,4) = 0 \text{ } ^\circ\text{C}; & T(1,4) = 6,25 \text{ } ^\circ\text{C}; & T(2,4) = 25 \text{ } ^\circ\text{C}; & T(3,4) = 56,25 \text{ } ^\circ\text{C}; & T(4,4) = 100 \text{ } ^\circ\text{C}. \end{array}$$

Также известно, что описанию распределения температуры в пластине соответствует уравнение Лапласа следующего вида:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0. \quad (33)$$

В соответствии с этапностью метода конечных разностей в первую очередь необходимо построить двумерную сетку с узловыми значениями. Шаг сетки равен 1 м ($h = 1$). Полученная сетка показана на рис. 72.

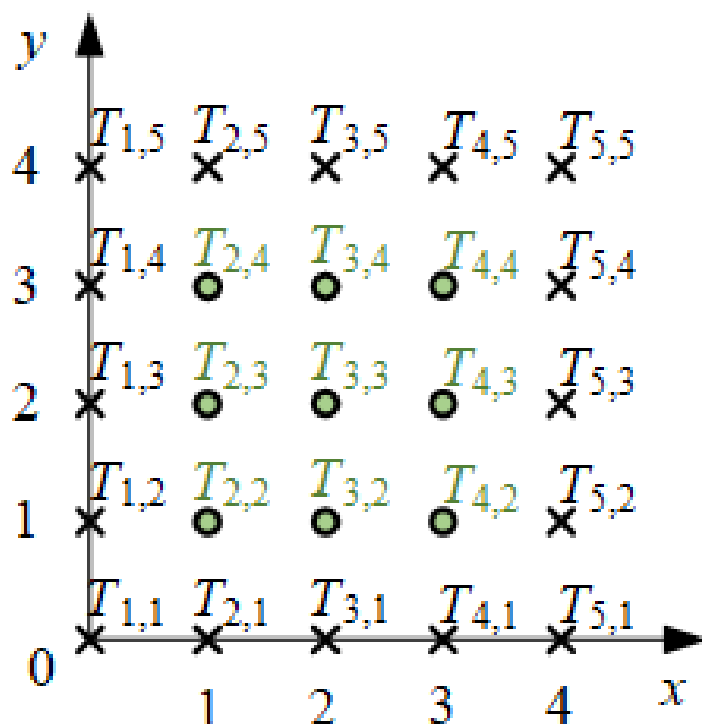


Рис. 72. Двумерная сетка узловых значений в области пластины

В силу того, что в пакете MATLAB индексация переменных начинается с 1, и для удобства дальнейшей работы узловые точки на рис. 72 имеют смещенную на 1 индексацию в отличие от исходных данных. Таким образом, узлы сетки, заданные в качестве граничных условий, имеют следующие значения:

$$\begin{aligned} T_{1,1} &= T_{1,2} = T_{1,3} = T_{1,4} = T_{1,5} = 0; \\ T_{5,1} &= T_{5,2} = T_{5,3} = T_{5,4} = T_{5,5} = 100; \\ T_{2,1} &= 25; \quad T_{3,1} = 50; \quad T_{4,1} = 75; \\ T_{2,5} &= 6,25; \quad T_{3,5} = 25; \quad T_{4,5} = 56,25. \end{aligned}$$

Следовательно, необходимо определить распределение температуры в девяти центральных точках, которые выделены на рис. 72. Для этого уравнение Лапласа (33) представляется в конечно-разностной форме:

$$\left\{ \begin{array}{l} \frac{T_{3,2} - 2T_{2,2} + T_{1,2}}{h^2} + \frac{T_{2,3} - 2T_{2,2} + T_{2,1}}{h^2} = 0; \\ \frac{T_{4,2} - 2T_{3,2} + T_{2,2}}{h^2} + \frac{T_{3,3} - 2T_{3,2} + T_{3,1}}{h^2} = 0; \\ \frac{T_{5,2} - 2T_{4,2} + T_{3,2}}{h^2} + \frac{T_{4,3} - 2T_{4,2} + T_{4,1}}{h^2} = 0; \\ \frac{T_{3,3} - 2T_{2,3} + T_{1,3}}{h^2} + \frac{T_{2,4} - 2T_{2,3} + T_{2,2}}{h^2} = 0; \\ \frac{T_{4,3} - 2T_{3,3} + T_{2,3}}{h^2} + \frac{T_{3,4} - 2T_{3,3} + T_{3,2}}{h^2} = 0; \\ \frac{T_{5,3} - 2T_{4,3} + T_{3,3}}{h^2} + \frac{T_{4,4} - 2T_{4,3} + T_{4,2}}{h^2} = 0; \\ \frac{T_{3,4} - 2T_{2,4} + T_{1,4}}{h^2} + \frac{T_{2,5} - 2T_{2,4} + T_{2,3}}{h^2} = 0; \\ \frac{T_{4,4} - 2T_{3,4} + T_{2,4}}{h^2} + \frac{T_{3,5} - 2T_{3,4} + T_{3,3}}{h^2} = 0; \\ \frac{T_{5,4} - 2T_{4,4} + T_{3,4}}{h^2} + \frac{T_{4,5} - 2T_{4,4} + T_{4,3}}{h^2} = 0. \end{array} \right. \quad (34)$$

Вводится следующее обозначение:

$$lm = \frac{1}{h^2}. \quad (35)$$

Тогда с учетом (35) система (34) принимает следующий вид:

$$\left\{ \begin{array}{l} lm \cdot T_{3,2} - 2 \cdot lm \cdot T_{2,2} + lm \cdot T_{1,2} + lm \cdot T_{2,3} - 2 \cdot lm \cdot T_{2,2} + lm \cdot T_{2,1} = 0; \\ lm \cdot T_{4,2} - 2 \cdot lm \cdot T_{3,2} + lm \cdot T_{2,2} + lm \cdot T_{3,3} - 2 \cdot lm \cdot T_{3,2} + lm \cdot T_{3,1} = 0; \\ lm \cdot T_{5,2} - 2 \cdot lm \cdot T_{4,2} + lm \cdot T_{3,2} + lm \cdot T_{4,3} - 2 \cdot lm \cdot T_{4,2} + lm \cdot T_{4,1} = 0; \\ lm \cdot T_{3,3} - 2 \cdot lm \cdot T_{2,3} + lm \cdot T_{1,3} + lm \cdot T_{2,4} - 2 \cdot lm \cdot T_{2,3} + lm \cdot T_{2,2} = 0; \\ lm \cdot T_{4,3} - 2 \cdot lm \cdot T_{3,3} + lm \cdot T_{2,3} + lm \cdot T_{3,4} - 2 \cdot lm \cdot T_{3,3} + lm \cdot T_{3,2} = 0; \\ lm \cdot T_{5,3} - 2 \cdot lm \cdot T_{4,3} + lm \cdot T_{3,3} + lm \cdot T_{4,4} - 2 \cdot lm \cdot T_{4,3} + lm \cdot T_{4,2} = 0; \\ lm \cdot T_{3,4} - 2 \cdot lm \cdot T_{2,4} + lm \cdot T_{1,4} + lm \cdot T_{2,5} - 2 \cdot lm \cdot T_{2,4} + lm \cdot T_{2,3} = 0; \\ lm \cdot T_{4,4} - 2 \cdot lm \cdot T_{3,4} + lm \cdot T_{2,4} + lm \cdot T_{3,5} - 2 \cdot lm \cdot T_{3,4} + lm \cdot T_{3,3} = 0; \\ lm \cdot T_{5,4} - 2 \cdot lm \cdot T_{4,4} + lm \cdot T_{3,4} + lm \cdot T_{4,5} - 2 \cdot lm \cdot T_{4,4} + lm \cdot T_{4,3} = 0. \end{array} \right. \quad (36)$$

Следует выполнить преобразования системы (36), перенеся неизвестные переменные в левую часть уравнений, а известные – в правую. В результате получается следующая система уравнений:

$$\left\{ \begin{array}{l}
-4 \cdot lm \cdot T_{2,2} + lm \cdot T_{3,2} + lm \cdot T_{2,3} = -lm \cdot T_{1,2} - lm \cdot T_{2,1}; \\
lm \cdot T_{2,2} - 4 \cdot lm \cdot T_{3,2} + lm \cdot T_{4,2} + lm \cdot T_{3,3} = -lm \cdot T_{3,1}; \\
-2 \cdot lm \cdot T_{4,2} + lm \cdot T_{3,2} + lm \cdot T_{4,3} - 2 \cdot lm \cdot T_{4,2} = -lm \cdot T_{5,2} - lm \cdot T_{4,1}; \\
lm \cdot T_{2,2} - 4 \cdot lm \cdot T_{2,3} + lm \cdot T_{3,3} + lm \cdot T_{2,4} = -lm \cdot T_{2,2}; \\
lm \cdot T_{3,2} + lm \cdot T_{2,3} - 4 \cdot lm \cdot T_{3,3} + lm \cdot T_{4,3} + lm \cdot T_{3,4} = 0; \\
lm \cdot T_{3,3} + lm \cdot T_{4,2} - 4 \cdot lm \cdot T_{4,3} + lm \cdot T_{4,4} = -lm \cdot T_{5,3}; \\
lm \cdot T_{2,3} - 4 \cdot lm \cdot T_{2,4} + lm \cdot T_{3,4} = -lm \cdot T_{1,4} - lm \cdot T_{2,5}; \\
lm \cdot T_{3,3} + lm \cdot T_{2,4} - 4 \cdot lm \cdot T_{3,4} + lm \cdot T_{4,4} = -lm \cdot T_{3,5}; \\
lm \cdot T_{4,3} + lm \cdot T_{3,4} - 4 \cdot lm \cdot T_{4,4} = -lm \cdot T_{4,5} - lm \cdot T_{5,4}.
\end{array} \right. \quad (37)$$

Дополнив систему (37) членами с нулевыми коэффициентами, имеем:

$$\left\{ \begin{array}{l}
-4 \cdot lm \cdot T_{2,2} + lm \cdot T_{3,2} + 0 \cdot T_{4,2} + lm \cdot T_{2,3} + 0 \cdot T_{3,3} + 0 \cdot T_{4,3} + 0 \cdot T_{2,4} + \\
+ 0 \cdot T_{3,4} + 0 \cdot T_{4,4} = -lm \cdot T_{1,2} - lm \cdot T_{2,1}; \\
lm \cdot T_{2,2} - 4 \cdot lm \cdot T_{3,2} + lm \cdot T_{4,2} + 0 \cdot T_{2,3} + lm \cdot T_{3,3} + 0 \cdot T_{4,3} + 0 \cdot T_{2,4} + \\
+ 0 \cdot T_{3,4} + 0 \cdot T_{4,4} = -lm \cdot T_{3,1}; \\
0 \cdot T_{2,2} + lm \cdot T_{3,2} - 4 \cdot lm \cdot T_{4,2} + 0 \cdot T_{2,3} + 0 \cdot T_{3,3} + lm \cdot T_{4,3} + 0 \cdot T_{2,4} + \\
+ 0 \cdot T_{3,4} + 0 \cdot T_{4,4} = -lm \cdot T_{5,2} - lm \cdot T_{4,1}; \\
lm \cdot T_{2,2} + 0 \cdot T_{3,2} + 0 \cdot T_{4,2} - 4 \cdot lm \cdot T_{2,3} + lm \cdot T_{3,3} + 0 \cdot T_{4,3} + lm \cdot T_{2,4} + \\
+ 0 \cdot T_{3,4} + 0 \cdot T_{4,4} = -lm \cdot T_{2,2}; \\
0 \cdot T_{2,2} + lm \cdot T_{3,2} + 0 \cdot T_{4,2} + lm \cdot T_{2,3} - 4 \cdot lm \cdot T_{3,3} + lm \cdot T_{4,3} + 0 \cdot T_{2,4} + \\
+ lm \cdot T_{3,4} + 0 \cdot T_{4,4} = 0; \\
0 \cdot T_{2,2} + 0 \cdot T_{3,2} + lm \cdot T_{4,2} + 0 \cdot T_{2,3} + lm \cdot T_{3,3} - 4 \cdot lm \cdot T_{4,3} + 0 \cdot T_{2,4} + \\
+ 0 \cdot T_{3,4} + lm \cdot T_{4,4} = -lm \cdot T_{5,3}; \\
0 \cdot T_{2,2} + 0 \cdot T_{3,2} + 0 \cdot T_{4,2} + lm \cdot T_{2,3} + 0 \cdot T_{3,3} - 0 \cdot T_{4,3} - 4 \cdot lm \cdot T_{2,4} + \\
+ lm \cdot T_{3,4} + 0 \cdot T_{4,4} = -lm \cdot T_{1,4} - lm \cdot T_{2,5}; \\
0 \cdot T_{2,2} + 0 \cdot T_{3,2} + 0 \cdot T_{4,2} + 0 \cdot T_{2,3} + lm \cdot T_{3,3} + 0 \cdot T_{4,3} + lm \cdot T_{2,4} - \\
- 4 \cdot lm \cdot T_{3,4} + lm \cdot T_{4,4} = -lm \cdot T_{3,5}; \\
0 \cdot T_{2,2} + 0 \cdot T_{3,2} + 0 \cdot T_{4,2} + 0 \cdot T_{2,3} + 0 \cdot T_{3,3} + lm \cdot T_{4,3} + 0 \cdot T_{2,4} + lm \cdot T_{3,4} - \\
- 4 \cdot lm \cdot T_{4,4} = -lm \cdot T_{4,5} - lm \cdot T_{5,4}.
\end{array} \right. \quad (38)$$

Далее для решения САУ (38) может быть применён матричный метод. Для этого составляется две матрицы:

$$A = \begin{pmatrix} -4 \cdot lm & lm & 0 & lm & 0 & 0 & 0 & 0 & 0 \\ lm & -4 \cdot lm & lm & 0 & lm & 0 & 0 & 0 & 0 \\ 0 & lm & -4 \cdot lm & 0 & 0 & lm & 0 & 0 & 0 \\ lm & 0 & 0 & -4 \cdot lm & lm & 0 & lm & 0 & 0 \\ 0 & lm & 0 & lm & -4 \cdot lm & lm & 0 & lm & 0 \\ 0 & 0 & lm & 0 & lm & -4 \cdot lm & 0 & 0 & lm \\ 0 & 0 & 0 & lm & 0 & 0 & -4 \cdot lm & lm & 0 \\ 0 & 0 & 0 & 0 & lm & 0 & lm & -4 \cdot lm & lm \\ 0 & 0 & 0 & 0 & 0 & lm & 0 & lm & -4 \cdot lm \end{pmatrix}$$

$$b = \begin{pmatrix} -lm \cdot T_{1,2} - lm \cdot T_{2,1} \\ -lm \cdot T_{3,1} \\ -lm \cdot T_{5,2} - lm \cdot T_{4,1} \\ -lm \cdot T_{2,2} \\ 0 \\ -lm \cdot T_{5,3} \\ -lm \cdot T_{1,4} - lm \cdot T_{2,5} \\ -lm \cdot T_{3,5} \\ -lm \cdot T_{4,5} - lm \cdot T_{5,4} \end{pmatrix}$$

Матрица A является основной и состоит из коэффициентов при неизвестных, а b – матрица свободных членов. С помощью пакета MATLAB выполняется программная реализация матричного метода решения САУ (38). Полученный программный код представлен на рис. 73. В первую очередь формируется массив граничных условий в строках с 1-й по 5-ю. Затем задается шаг сетки – строка 7. В строках с 10-й по 28-ю формируются матрицы для матричного метода решения системы алгебраических уравнений. В строке 30 реализована формула самого матричного метода. В строках с 32-й по 34-ю формируется матрица распределения стационарного температурного поля в пластине. В табличной форме полученный результат показан на рис. 74.

Для графического представления полученного результата используется команда *surf*, которая позволяет строить трехмерные графики (рис. 75).

```

1 - T = zeros(5,5);
2 - T(1,1) = 0; T(1,2) = 0; T(1,3) = 0; T(1,4) = 0; T(1,5) = 0;
3 - T(5,1) = 100; T(5,2) = 100; T(5,3) = 100; T(5,4) = 100; T(5,5) = 100;
4 - T(2,1) = 25; T(3,1) = 50; T(4,1) = 75;
5 - T(2,5) = 6.25; T(3,5) = 25; T(4,5) = 56.25;
6
7 - h = 1;
8 - lm = 1/h;
9
10 - A = [-4*lm lm 0 lm 0 0 0 0 0
11         lm -4*lm lm 0 lm 0 0 0 0
12         0 lm -4*lm 0 0 lm 0 0 0
13         lm 0 0 -4*lm lm 0 lm 0 0
14         0 lm 0 lm -4*lm lm 0 lm 0
15         0 0 lm 0 lm -4*lm 0 0 lm
16         0 0 0 lm 0 0 -4*lm lm 0
17         0 0 0 0 lm 0 lm -4*lm lm
18         0 0 0 0 0 lm 0 lm -4*lm];
19
20 - b = [-lm*T(1,2)-lm*T(2,1)
21         -lm*T(3,1)
22         -lm*T(5,2)-lm*T(4,1)
23         -lm*T(2,2)
24         0
25         -lm*T(5,3)
26         -lm*T(1,4)-lm*T(2,5)
27         -lm*T(3,5)
28         -lm*T(4,5)-lm*T(5,4)];
29
30 - X = (A^-1)*b;
31
32 - T(2,2) = X(1); T(3,2) = X(2); T(4,2) = X(3);
33 - T(2,3) = X(4); T(3,3) = X(5); T(4,3) = X(6);
34 - T(2,4) = X(7); T(3,4) = X(8); T(4,4) = X(9);

```

Рис. 73. Программный код для определения стационарного температурного поля в пластине

	1	2	3	4	5
1	0	0	0	0	0
2	25	23.4933	21.0938	16.3504	6.2500
3	50	47.8795	44.5313	38.0580	25
4	75	73.4933	71.0938	66.3504	56.2500
5	100	100	100	100	100

Рис. 74. Матрица распределения стационарного температурного поля в пластине

```

36 - x=0:1:4;
37 - y=0:1:4;
38 - surf(x, y, T')
39 - xlabel('X, м')
40 - ylabel('Y, м')
41 - zlabel('Z, ^0C')

```

Рис. 75. Код для графического представления полученного результата

На рис. 76 представлен график распределения стационарного температурного поля в пластине.

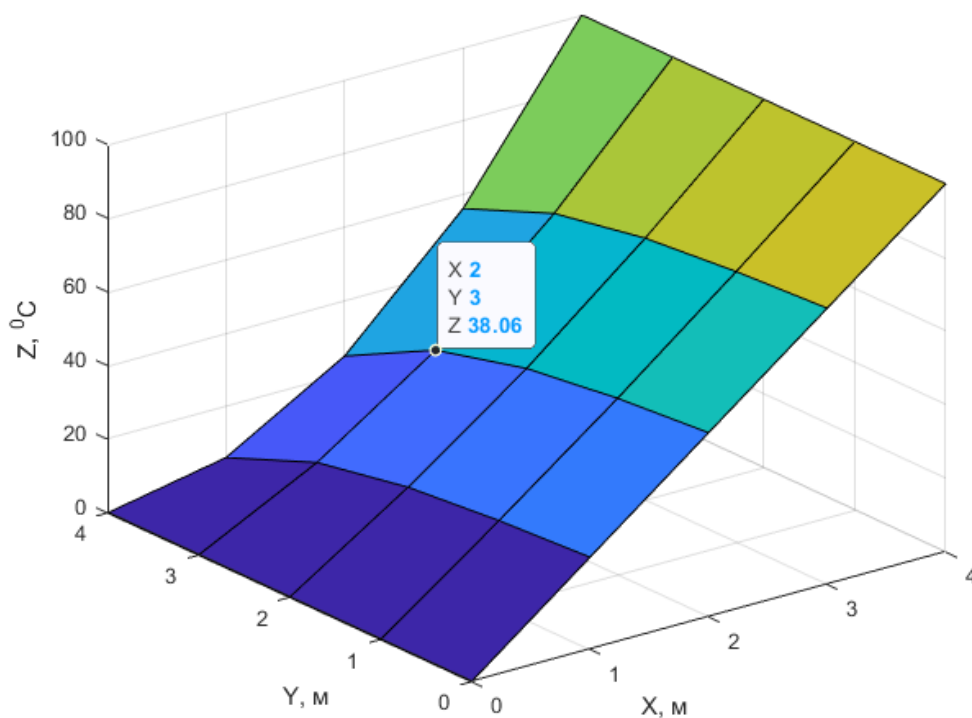


Рис. 76. Распределение стационарного температурного поля в пластине

Индивидуальные задания

1. Для модели процесса теплопередачи, представленной в индивидуальном варианте задания, построить сетку узловых значений в области решения.
2. Сформировать в общем виде конечно-разностное выражение, используя заданное дифференциальное уравнение.
3. Сформировать систему уравнений, позволяющих определить значения функции в неизвестных узловых точках.
4. Для нахождения решения модели с помощью метода конечных разностей разработать схему алгоритма.

5. Выполнить программную реализацию в пакете MATLAB разработанной схемы алгоритма.

6. Полученные результаты представить таблично и графически, используя трехмерный график.

7. Подготовить отчет в соответствии с требованиями, представленными в прил. А–В. Проанализировать полученные результаты и сделать выводы.

Вариант 1

Имеется прямоугольная пластина следующих размеров: ширина (координата – x) равна 4 м, а длина (координата – y) – 3 м. При этом известна температура ($T(x, y)$) в следующих точках этой пластины:

$$T(0,0) = 0\text{ }^{\circ}\text{C}; \quad T(1,0) = 25\text{ }^{\circ}\text{C}; \quad T(2,0) = 50\text{ }^{\circ}\text{C}; \quad T(3,0) = 75\text{ }^{\circ}\text{C}; \quad T(4,0) = 100\text{ }^{\circ}\text{C};$$

$$T(0,1) = 10\text{ }^{\circ}\text{C}; \quad T(4,1) = 130\text{ }^{\circ}\text{C};$$

$$T(0,2) = 50\text{ }^{\circ}\text{C}; \quad T(4,2) = 170\text{ }^{\circ}\text{C};$$

$$T(0,3) = 80\text{ }^{\circ}\text{C}; \quad T(1,3) = 120\text{ }^{\circ}\text{C}; \quad T(2,3) = 140\text{ }^{\circ}\text{C}; \quad T(3,3) = 160\text{ }^{\circ}\text{C}; \quad T(4,3) = 200\text{ }^{\circ}\text{C}.$$

Также известно, что описанию распределения температуры в пластине соответствует уравнение Лапласа следующего вида:

$$\frac{\partial^2 T}{\partial x^2} + \lambda \cdot \frac{\partial^2 T}{\partial y^2} = 0.$$

В представленном уравнении коэффициент λ является коэффициентом теплопроводности и равен 3.

Вариант 2

Имеется прямоугольная пластина следующих размеров: ширина (координата – x) равна 3 м, а длина (координата – y) – 4 м. При этом известна температура ($T(x, y)$) в следующих точках этой пластины:

$$T(0,0) = 0\text{ }^{\circ}\text{C}; \quad T(1,0) = 20\text{ }^{\circ}\text{C}; \quad T(2,0) = 30\text{ }^{\circ}\text{C}; \quad T(3,0) = 40\text{ }^{\circ}\text{C};$$

$$T(0,1) = 20\text{ }^{\circ}\text{C}; \quad T(3,1) = 60\text{ }^{\circ}\text{C};$$

$$T(0,2) = 60\text{ }^{\circ}\text{C}; \quad T(3,2) = 80\text{ }^{\circ}\text{C};$$

$$T(0,3) = 80\text{ }^{\circ}\text{C}; \quad T(3,3) = 100\text{ }^{\circ}\text{C};$$

$$T(0,4) = 100\text{ }^{\circ}\text{C}; \quad T(1,4) = 107\text{ }^{\circ}\text{C}; \quad T(2,4) = 114\text{ }^{\circ}\text{C}; \quad T(3,4) = 120\text{ }^{\circ}\text{C}.$$

Также известно, что описанию распределения температуры в пластине соответствует уравнение Лапласа следующего вида:

$$\frac{\partial^2 T}{\partial x^2} + \lambda \cdot \frac{\partial^2 T}{\partial y^2} = 0.$$

В представленном уравнении коэффициент λ является коэффициентом теплопроводности и равен 2.

Вариант 3

Имеется прямоугольная пластина следующих размеров: ширина (координата – x) равна 4 м, а длина (координата – y) – 5 м. При этом известна температура ($T(x, y)$) в следующих точках этой пластины:

$$\begin{array}{ll} T(0,0) = 0 \text{ }^{\circ}\text{C}; & T(1,0) = 10 \text{ }^{\circ}\text{C}; \quad T(2,0) = 20 \text{ }^{\circ}\text{C}; \quad T(3,0) = 30 \text{ }^{\circ}\text{C}; \quad T(4,0) = 40 \text{ }^{\circ}\text{C}; \\ T(0,1) = 5 \text{ }^{\circ}\text{C}; & T(4,1) = 47 \text{ }^{\circ}\text{C}; \\ T(0,2) = 10 \text{ }^{\circ}\text{C}; & T(4,2) = 54 \text{ }^{\circ}\text{C}; \\ T(0,3) = 15 \text{ }^{\circ}\text{C}; & T(4,3) = 61 \text{ }^{\circ}\text{C}; \\ T(0,4) = 20 \text{ }^{\circ}\text{C}; & T(4,4) = 68 \text{ }^{\circ}\text{C}; \\ T(0,5) = 25 \text{ }^{\circ}\text{C}; & T(1,5) = 30 \text{ }^{\circ}\text{C}; \quad T(2,5) = 40 \text{ }^{\circ}\text{C}; \quad T(3,5) = 55 \text{ }^{\circ}\text{C}; \quad T(4,5) = 75 \text{ }^{\circ}\text{C}. \end{array}$$

Также известно, что описанию распределения температуры в пластине соответствует уравнение Лапласа следующего вида:

$$\frac{\partial^2 T}{\partial x^2} + \lambda^2 \cdot \frac{\partial^2 T}{\partial y^2} = 0.$$

В представленном уравнении коэффициент λ является коэффициентом теплопроводности и равен 2.

Вариант 4

Имеется прямоугольная пластина следующих размеров: ширина (координата – x) равна 3 м, а длина (координата – y) – 3 м. При этом известна температура ($T(x, y)$) в следующих точках этой пластины:

$$\begin{array}{llll} T(0,0) = 0 \text{ }^{\circ}\text{C}; & T(1,0) = 20 \text{ }^{\circ}\text{C}; & T(2,0) = 40 \text{ }^{\circ}\text{C}; & T(3,0) = 60 \text{ }^{\circ}\text{C}; \\ T(0,1) = 20 \text{ }^{\circ}\text{C}; & & & T(3,1) = 70 \text{ }^{\circ}\text{C}; \\ T(0,2) = 40 \text{ }^{\circ}\text{C}; & & & T(3,2) = 80 \text{ }^{\circ}\text{C}; \\ T(0,3) = 60 \text{ }^{\circ}\text{C}; & T(1,3) = 70 \text{ }^{\circ}\text{C}; & T(2,3) = 80 \text{ }^{\circ}\text{C}; & T(3,3) = 90 \text{ }^{\circ}\text{C}. \end{array}$$

Также известно, что описанию распределения температуры в пластине соответствует уравнение Лапласа следующего вида:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\lambda}{4} \cdot \frac{\partial^2 T}{\partial y^2} = 0.$$

В представленном уравнении коэффициент λ является коэффициентом теплопроводности и равен 2.

Вариант 5

Имеется прямоугольная пластина следующих размеров: ширина (координата – x) равна 3 м, а длина (координата – y) – 5 м. При этом известна температура ($T(x, y)$) в следующих точках этой пластины:

$$\begin{array}{llll} T(0,0) = 10 \text{ }^{\circ}\text{C}; & T(1,0) = 30 \text{ }^{\circ}\text{C}; & T(2,0) = 50 \text{ }^{\circ}\text{C}; & T(3,0) = 70 \text{ }^{\circ}\text{C}; \\ T(0,1) = 30 \text{ }^{\circ}\text{C}; & & & T(3,1) = 100 \text{ }^{\circ}\text{C}; \\ T(0,2) = 50 \text{ }^{\circ}\text{C}; & & & T(3,2) = 130 \text{ }^{\circ}\text{C}; \end{array}$$

$$\begin{aligned}
T(0,3) &= 70 \text{ }^{\circ}\text{C}; & T(3,3) &= 160 \text{ }^{\circ}\text{C}; \\
T(0,4) &= 90 \text{ }^{\circ}\text{C}; & T(3,4) &= 190 \text{ }^{\circ}\text{C}; \\
T(0,5) &= 110 \text{ }^{\circ}\text{C}; & T(1,5) &= 140 \text{ }^{\circ}\text{C}; & T(2,5) &= 180 \text{ }^{\circ}\text{C}; & T(3,5) &= 220 \text{ }^{\circ}\text{C}.
\end{aligned}$$

Также известно, что описанию распределения температуры в пластине соответствует уравнение Лапласа следующего вида:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0.$$

Контрольные вопросы и задания

1. Каковы область применения и суть метода конечных разностей?
2. В чем суть метода конечных разностей?
3. Перечислите существующие виды/типы метода конечных разностей.
4. Поясните суть задачи с граничными условиями.
5. Запишите выражение для приближенного вычисления частной производной функции следующего вида: $\frac{\partial^2 f(x, y)}{\partial x^2}$.
6. Запишите выражение для приближенного вычисления частной производной функции следующего вида: $\frac{\partial^2 f(x, y)}{\partial x \cdot \partial y}$.

Список рекомендованной литературы

1. Кольцова Э.М. Численные методы решения уравнений математической физики и химии : учебное пособие для академического бакалавриата. – 2-е изд., испр. и доп. / Э.М. Кольцова, А.С. Скичко, А.В. Женса. – Москва : Юрайт, 2018. – 220 с.
2. Амосов А.А. Вычислительные методы для инженеров : учебное пособие / А.А. Амосов, В.А. Дубинский, Н.В. Копченкова. – Москва : Высшая школа, 1994. – 544 с.

Лабораторная работа № 5 СТАТИСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ПРИ СТОХАСТИЧЕСКОМ ИЗМЕНЕНИИ ИХ ПАРАМЕТРОВ

Цель работы: изучение методов статистического моделирования для исследования физических процессов при стохастическом изменении их параметров, с использованием специализированного программного обеспечения MATLAB.

Основные теоретические сведения

Физическим процессам, окружающим человека, присуща случайная (стохастическая) составляющая. Для учета стохастической составляющей физических процессов при математическом моделировании используют статистическое моделирование – подвид имитационного моделирования. Статистическое моделирование направлено на то, чтобы создаваемая математическая модель физического процесса принимала во внимание случайный характер изменения некоторых параметров моделируемого процесса и позволяла выполнять статистические эксперименты. Под статистическим экспериментом понимается проведение ряда испытаний, то есть математических расчетов или прогонов программы, реализующей математическую модель. Результатом каждого испытания будет являться случайное значение наблюдаемой величины, так как параметры модели задаются случайным образом. Проведение серии таких экспериментов приводит к формированию выборки, состоящей из множества случайных значений наблюдаемой величины. Для дальнейшей обработки такой выборки прибегают к применению методов математической статистики для определения необходимых оценок, таких как среднее, среднеквадратичное отклонение и др. Достоверность получаемого результата тем выше, чем больше число выполненных экспериментов ($N \rightarrow \infty$).

В представленном алгоритме (рис. 77) можно выделить три основных этапа. Первый этап заключается в формировании случайных значений параметров математической модели с помощью генератора псевдослучайных чисел. При этом необходимо учесть, чтобы распределение случайных значений параметров модели соответствовало необходимому закону. Как правило в качестве закона распределения используют нормальный закон распределения (распределение Гаусса). Результатом выполнения первого этапа алгоритма является массив случайных значений параметров модели.

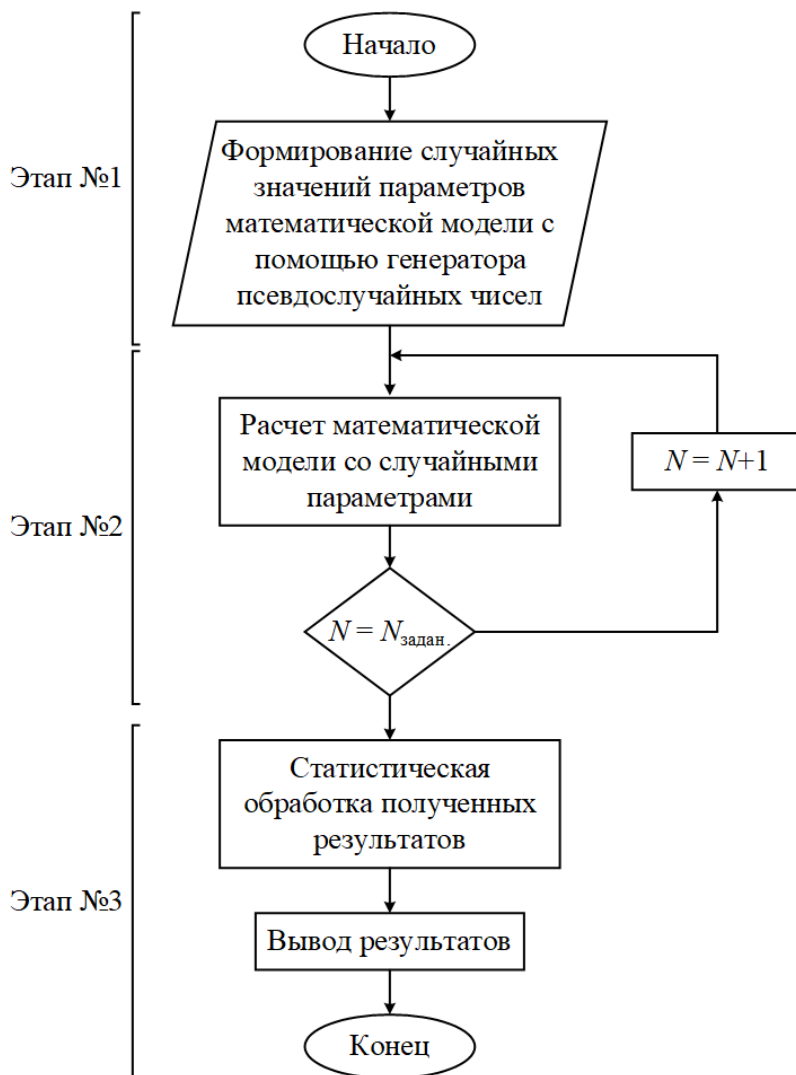


Рис. 77. Алгоритм статистических экспериментов

На втором этапе производится расчет математической модели с использованием случайных значений параметров из массива, сформированного на первом этапе алгоритма. Количество вычислений (численных экспериментов) будет меньше или равно числу случайных значений каждого параметра в сформированном массиве. Результатом выполнения второго этапа алгоритма является множество случайных значений наблюдаемой величины.

На третьем этапе осуществляется статистическая обработка полученного множества случайных значений наблюдаемой величины. Для этого применяются различные методы математической статистики. Как правило, на данном этапе вычисляются среднее значение наблюдаемой величины и ее среднеквадратичное отклонение. Например, если моделирование процесса выполняется в динамике, то есть во времени, то для вычисления среднего значения наблюдаемой величины может быть использована следующая формула:

$$\bar{y}(t_j) = \frac{\sum_{i=1}^N y_i(t_j)}{N},$$

где y_i – значение наблюдаемой величины в момент времени t_j ; N – количество численных экспериментов.

Одним из общеизвестных и широко используемых методов статистического моделирования процессов является метод Монте-Карло, который полностью соответствует представленному алгоритму (рис. 77). Изначально метод Монте-Карло предназначался для решения задач нейтронной физики, но впоследствии он нашел широкое применение во многих областях, где требуется статистическое моделирование. В основе этого метода лежит генератор псевдослучайных чисел. На сегодняшний день метод используют как для нахождения значений определенного интеграла, так и для решения более сложных физических задач, таких как расчет стационарного профиля температуры в пластине, вычисление траекторий распространения нейтронов в веществе и др.

Примеры программной реализации методов статистического моделирования процессов в пакете MATLAB

Пример 1. Математическое моделирование прохождения нейтронов сквозь пластину в пакете MATLAB. В качестве примера предлагается рассмотреть процесс преодоления нейтронами пластины. Имеется пластина толщиной h (рис. 78). На пластину поступают нейтроны, обладающие энергией E_0 . При этом угол падения нейтронов составляет 90° . Возможно три варианта движения нейтронов после попадания на пластину. Вариант a – нейтрон проходит сквозь пластину, вариант b – нейтрон поглощается пластиной, вариант c – нейтрон отражается пластиной. В данном случае делается допущение о том, что в результате рассеяния нейтронов их энергия не меняется.

Задача заключается в необходимости определения вероятностей возникновения одного из возможных вариантов движения нейтрона:

- вариант a – p^+ – нейтрон прошел сквозь пластину;
- вариант b – p^0 – нейтрон поглотился пластиной;
- вариант c – p^- – нейтрон отразился пластиной.

Как известно, процесс взаимодействия нейтронов с веществом характеризуется сечением поглощения (Σ_c) и сечением рассеяния (Σ_s), а их сумма является полным сечением взаимодействия:

$$\Sigma = \Sigma_s + \Sigma_c. \quad (39)$$

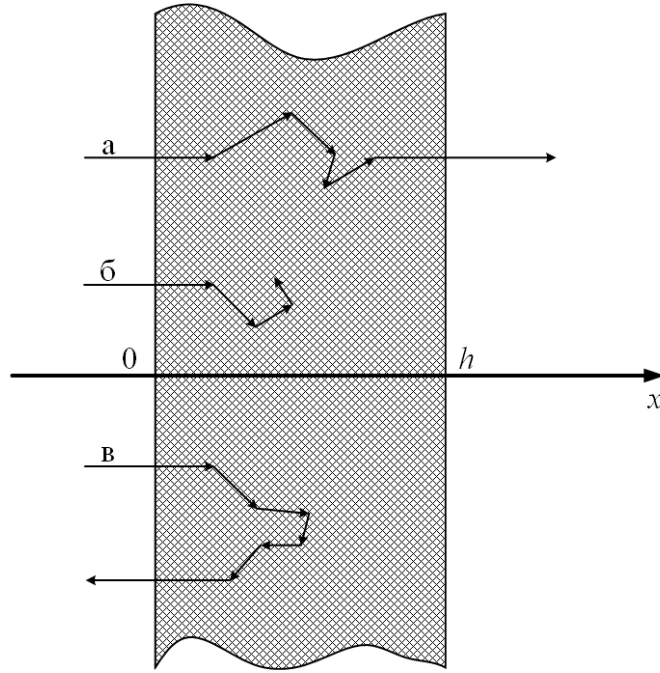


Рис. 78. Схема распространения нейтронов в пластине

Тогда из (39) имеем, что вероятность поглощения определяется по следующей формуле:

$$\frac{\Sigma_c}{\Sigma}.$$

Вероятность рассеяния

$$\frac{\Sigma_s}{\Sigma}.$$

Для нахождения длины свободного пробега нейтрона может быть использована следующая формула:

$$\lambda = -\frac{1}{\Sigma} \cdot \ln \gamma,$$

где γ – является случайной величиной с равномерным распределением.

Далее необходимо рассчитать траекторию нейтрона после рассеяния. Согласно схеме, представленной на рис. 78, пластина является бесконечной и нейтроны попадают на пластину под углом 90° , тогда направление будет зависеть от угла (φ) между осью OX и скоростью. Принимая во внимание требование о том, что любое направление движения нейтрона равновероятное, можно сформулировать равносильное требование, что косинус угла ($\mu = \cos(\varphi)$) должен быть равномерно распределен в интервале от -1 до 1 . Правило розыгрыша в этом случае на указанном интервале $[-1, 1]$ имеет вид

$$\gamma = \frac{\mu - (-1)}{1 - (-1)}.$$

тогда

$$\mu = 2 \cdot \gamma - 1.$$

С учетом всего вышеизложенного можно составить следующую схему алгоритма расчета.

Допустим, что один из нейтронов претерпел k -е рассеяние в пластине. Точка рассеяния имеет координату по оси абсцисс x_k . После этого движение нейтрона продолжилось в направлении μ_k . Тогда длина свободного пробега разыгрывается по следующей формуле:

$$\lambda_k = -\frac{1}{\Sigma} \cdot \ln \gamma.$$

Для определения следующей точки столкновения может быть использована формула

$$x_{k+1} = x_k + \lambda_k \cdot \mu_k.$$

Затем целесообразно проверить условие прохождения нейтрона сквозь пластину: $x_{k+1} > h$. В случае выполнения данного условия переменная-счетчик прошедших нейтронов (N^+) увеличивается на единицу. Если указанное условие не выполняется, то проверяется условие отражения нейтрона: $0 > x_{k+1}$. При корректности неравенства ($0 > x_{k+1}$) переменная-счетчик отраженных нейтронов (N^-) увеличивается на единицу. В противном случае, то есть при $0 \leq x_{k+1} \leq h$, нейтрон испытывает $(k+1)$ -е столкновение в пластине и необходимо определить его дальнейшую судьбу. Производится проверка условия поглощения:

$$\gamma < \frac{\Sigma_c}{\Sigma}. \quad (40)$$

Переменная-счетчик поглощенных нейтронов (N^0) увеличивается на единицу, если выражение (40) справедливо. В противном случае осуществляется розыгрыш процесса рассеяния $\mu_{k+1} = 2 \cdot \gamma - 1$ и повторение цикла, но с другими значениями γ . Начальные условия для каждой траектории имеют следующий вид: $x_0 = 0$, $\mu_0 = 1$.

Для определения вероятностей возникновения одного из возможных вариантов движения нейтрона могут быть использованы следующие выражения:

$$p^+ \approx \frac{N^+}{N};$$

$$p^0 \approx \frac{N^0}{N};$$

$$p^- \approx \frac{N^-}{N};$$

$$N = N^+ + N^0 + N^-.$$

Программный код, реализующий расписанный алгоритм, представлен на рис. 79.

```

1 - sigma = 15; sigma_c = 5;
2 - h = 0.2;
3 - N = 1000;
4 - Nplus = 0; Nnull = 0; Nminus = 0;
5 - x = []; m = [];
6 - j = 1;
7 - while (j <= N)
8 -     k = 1; x(1) = 0; m(1) = 1; cycle = 1;
9 -     while(cycle == 1)
10 -         lambda = (-1/sigma)*log(rand(1));
11 -         x(k+1) = x(k) + lambda * m(k);
12 -         if(x(k+1) > h)
13 -             Nplus = Nplus + 1;
14 -             cycle = 0;
15 -         else
16 -             if(x(k+1) < 0)
17 -                 Nminus = Nminus + 1;
18 -                 cycle = 0;
19 -             else
20 -                 if (rand(1) < (sigma_c/sigma))
21 -                     Nnull = Nnull + 1;
22 -                     cycle = 0;
23 -                 else
24 -                     m(k+1) = 2*rand(1) - 1;
25 -                     k = k + 1;
26 -                 end
27 -             end
28 -         end
29 -     end
30 -     j = j + 1;
31 - end
32 - p_plus = Nplus / N;
33 - p_minus = Nminus / N;
34 - p_null = Nnull / N;
35 - p = p_plus + p_minus + p_null;
36 - labels = {'Прошел', 'Отразился', 'Поглотился'};
37 - pie([p_plus p_minus p_null], '%.2f%%')
38 - legend(labels)

```

Рис. 79. Программный код для определения вероятностей возникновения одного из возможных вариантов движения нейтрона в пластине в пакете MATLAB

Полученные результаты моделирования представлены на рис. 80.

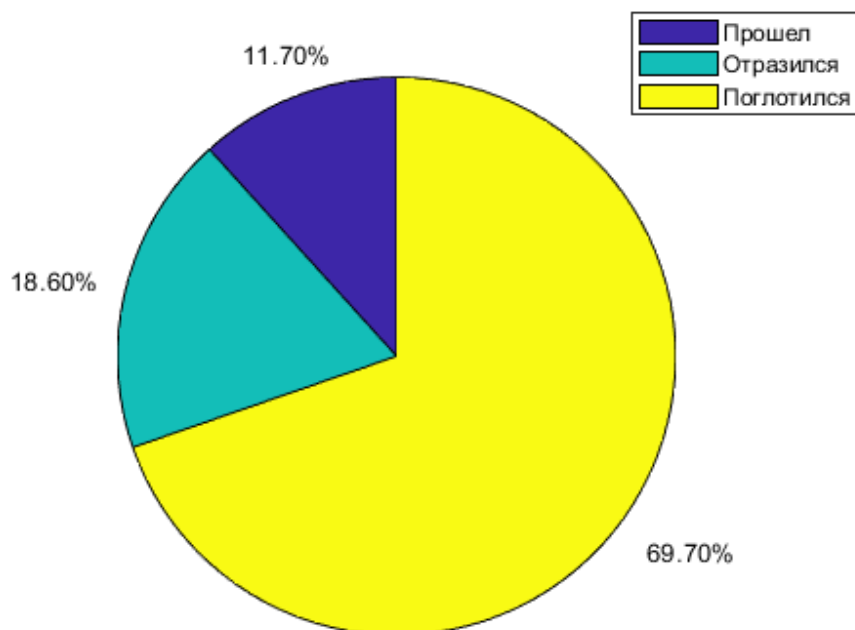


Рис. 80. Вероятности возникновения одного из возможных вариантов движения нейтрона в пластине

Индивидуальные задания

1. В соответствии с индивидуальным вариантом произвести математическое моделирование процесса прохождения нейтронов сквозь пластину в пакете MATLAB. Для этого использовать метод Монте-Карло ($N = 100$ – количество итераций). В итоге должны быть рассчитаны вероятности прохождения (вариант $a - p^+$), поглощения (вариант $b - p^0$) и отражения (вариант $c - p^-$) нейтрона пластиной (рис. 78). Для полученных данных определить среднее значение вероятностей возникновения одного из возможных вариантов движения нейтрона в пластине, а также среднее квадратичное отклонение. Результаты вычислений представить в форме таблицы.

Номер варианта	Параметры процесса
1	$h = 0,01; \Sigma_c = 5; \Sigma_s = 15$
2	$h = 0,02; \Sigma_c = 4; \Sigma_s = 15$
3	$h = 0,03; \Sigma_c = 3; \Sigma_s = 15$
4	$h = 0,03; \Sigma_c = 4; \Sigma_s = 15$
5	$h = 0,02; \Sigma_c = 5; \Sigma_s = 15$
6	$h = 0,02; \Sigma_c = 3; \Sigma_s = 15$
7	$h = 0,01; \Sigma_c = 4; \Sigma_s = 15$
8	$h = 0,03; \Sigma_c = 5; \Sigma_s = 15$

Номер варианта	Параметры процесса
9	$h = 0,01; \Sigma_c = 3; \Sigma_s = 15$
10	$h = 0,04; \Sigma_c = 3; \Sigma_s = 15$
11	$h = 0,04; \Sigma_c = 2; \Sigma_s = 15$
12	$h = 0,05; \Sigma_c = 2; \Sigma_s = 15$
13	$h = 0,05; \Sigma_c = 4; \Sigma_s = 15$
14	$h = 0,04; \Sigma_c = 4; \Sigma_s = 15$
15	$h = 0,05; \Sigma_c = 3; \Sigma_s = 15$

2. Повторить выполнение п. 1 при $N = 1000$. Для полученных данных определить среднее значение вероятностей возникновения одного из возможных вариантов движения нейтрона в пластине, а также средне-квадратичное отклонение. Результаты вычислений представить в форме таблицы.

3. Повторить выполнение п. 1 при $N = 10000$. Для полученных данных определить среднее значение вероятностей возникновения одного из возможных вариантов движения нейтрона в пластине, а также средне-квадратичное отклонение. Результаты вычислений представить в форме таблицы.

4. Подготовить отчет в соответствии с требованиями, представленными в прил. А–В. Проанализировать полученные результаты и сделать выводы.

Контрольные вопросы и задания

1. Объясните суть метода статистического моделирования.
2. Какие задачи могут быть решены с помощью метода статистического моделирования?
3. Перечислите основные этапы метода статистического моделирования.
4. Чем определяется точность данных, полученных в результате статистического моделирования?

Список рекомендованной литературы

1. Михайлов Г.А. Статистическое моделирование. Методы Монте-Карло : учебное пособие для бакалавриата и магистратуры / Г.А. Михайлов, А.В. Войтишек. – Москва : Юрайт, 2019. – 371 с.
2. Соболев М.И. Численные методы Монте-Карло / М.И. Соболев. – Москва : Наука, 1973. – 307 с.

Лабораторная работа № 6

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ ФИЗИЧЕСКОГО ПРОЦЕССА

Цель работы: изучение методов и подходов построения и проверки работоспособности математических моделей физических процессов в разных режимах работы с использованием специализированного программного обеспечения MATLAB.

Основные теоретические сведения

Моделирование является важным инструментом как для исследования окружающего мира, в том числе и технологических процессов, так и для проверки и отработки различных теорий, устройств, алгоритмов и т. д. На сегодняшний день выделяют различные виды моделей: натурные, физические, математические и др.

С развитием ЭВМ и технологий популярность получили математические модели в силу множества достоинств. В литературе существует большое количество определений термина «математическая модель». Обобщая их, можно сформулировать следующее: математическая модель – это описание объектов и процессов разного класса, а также их поведения при разных условиях с помощью математических зависимостей разного вида (алгебраические уравнения, дифференциальные и т. д.).

В рамках математического моделирования можно выделить следующие задачи:

- замещение реального физического объекта-оригинала (аппарата, технологического процесса и т. д.) с помощью его математической модели;
- математическая модель объекта-оригинала повторяет и воспроизводит с определенной точностью его свойства, характеристики и поведение, которые имеют принципиальное значение для решения поставленных задач моделирования;
- проведение численных экспериментов с помощью математической модели объекта-оригинала позволяет исследовать свойства и поведение объекта-оригинала в разных условиях.

На практике можно выделить пять основных этапов разработки математической модели (рис. 81).

Первый этап моделирования заключается в постановке цели и задач. Важность данного этапа заключается в том, что он является определяющим. С учетом этого, для каких целей создается математическая модель, будет

зависеть ее структура и сложность математического аппарата, на котором она базируется.



Рис. 81. Основные этапы разработки математической модели

Второй этап заключается непосредственно в разработке математического описания. В свою очередь, в рамках данного этапа можно выделить ряд других задач.

Во-первых, на основании сформулированной цели моделирования анализируется моделируемый объект или процесс. В результате составляется информационная схема объекта моделирования, которая отражает взаимосвязь входных и выходных переменных (рис. 82).

Выделенные входные и выходные переменные разделяют:

- на переменные – зависят от времени и могут изменяться в процессе численного эксперимента;
- константы – физические и химические константы, которые являются неизменными для данных условий.

Как правило, на информационной схеме (рис. 82) вектор $\bar{X} = \{x_1, x_2, x_3, \dots, x_n\}$ представляет собой значения входных переменных, вектор $\bar{Y} = \{y_1, y_2, y_3, \dots, y_n\}$ – значения выходных переменных и вектор $\bar{Z} = \{z_1, z_2, z_3, \dots, z_n\}$ – значения параметров и констант.

Во-вторых, на основании сформулированных цели и задач моделирования, а также в результате проведенного анализа объекта формируется

перечень допущений и требований к разрабатываемой математической модели.

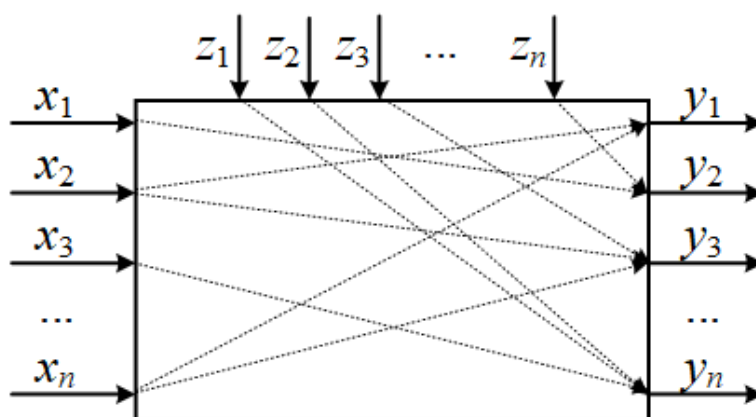


Рис. 82. Пример информационной схемы объекта моделирования

В-третьих, на основании разработанной информационной схемы (рис. 82) сформулированных допущений и требований, формируется система уравнений, описывающих взаимосвязь входных и выходных переменных.

На практике широкое распространение получили такие типовые модели, как модель идеального перемешивания и модель идеального вытеснения.

В основе модели идеального перемешивания лежит идея о том, что поток, поступающий в аппарат, мгновенно и равномерно распределяется по нему и вследствие этого переменная, характеризующая вещество в аппарате (концентрация, температура и др.), одинакова во всех точках аппарата и в выходном потоке (рис. 83).

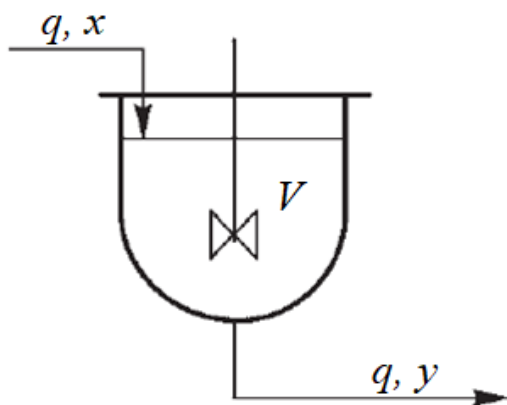


Рис. 83. Схема модели идеального перемешивания

Формула для модели идеального перемешивания имеет следующий вид:

$$\frac{dy}{dt} = \frac{q}{V} \cdot (x - y),$$

где x – переменная, характеризующая входной поток (температура, концентрация и др.); y – переменная, характеризующая выходной поток (температура, концентрация и др.) и вещество в любой точке аппарата; V – объем аппарата; q – скорость входного и выходного потоков.

Модели данного типа еще принято называть моделями с сосредоточенными параметрами, так как отсутствуют изменения переменных в пространстве.

В основе модели идеального вытеснения лежит идея о том, что вещество распределено в движущемся продольно потоке и переменная, характеризующая вещество (концентрация, температура и др.) отличается в каждой точке потока по направлению движения (рис. 84).

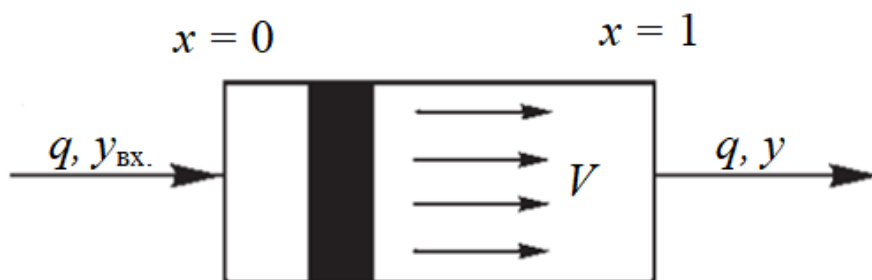


Рис. 84. Схема модели идеального вытеснения

Формула для модели идеального вытеснения имеет следующий вид:

$$\frac{\partial y}{\partial t} = -U \cdot \frac{\partial y}{\partial x};$$

$$U = \frac{q}{S},$$

где U – скорость потока; S – площадь поперечного сечения.

Модели данного типа еще принято называть моделями с распределенными параметрами, так как имеется зависимость переменной, характеризующей вещество в аппарате (концентрация, температура и др.), не только от времени, но и от координаты.

Кроме того, существуют и другие типовые модели (например, диффузионная модель), которые также находят применение на практике для различных объектов.

Стоит отметить что разрабатываемая математическая модель может базироваться как на типовых моделях, так и на моделях другого типа,

например на базе эмпирических полиномов. Также модель может быть выполнена как комбинированная, например в виде системы алгебро-дифференциальных уравнений, которая включает в себя как модель идеального перемешивания, так и эмпирические полиномы.

Третий этап моделирования связан с алгоритмизацией и программной реализацией разработанной математической модели объекта. Предварительно выбираются численный метод и алгоритм решения разработанной математической модели. Согласно разработанному алгоритму выполняется программирование математической модели.

Численный метод решения разработанной математической модели выбирается в зависимости от вида уравнений, на которых она базируется. Реализация численного метода может быть выполнена как самим разработчиком, так и могут быть использованы встроенные функции и команды, реализующие такие методы. Например, в пакете MATLAB имеется ряд функций (*ode45*, *ode23* и другие), которые реализуют численные методы (например, методы Рунге–Кутты), для решения системы дифференциальных уравнений и в частности модели идеального перемешивания.

Четвертый этап математического моделирования связан с проверкой адекватности разработанной математической модели. Для этого существует несколько подходов. При наличии экспериментальных данных возможны их сопоставление с расчетными, полученными на разработанной математической модели, и оценка погрешности их расхождения, например с помощью вычисления среднеквадратичного отклонения. При отсутствии экспериментальных данных производится проверка выполнения физико-химических законов, например закона сохранения энергии, закона сохранения массы и т. д.

В случае если разработанная математическая модель не адекватна или с недостаточной точностью описывает поведение объекта оригинала, возвращаются к первому этапу моделирования и повторяют всю процедуру заново.

Если в ходе проверки доказано, что разработанная математическая модель адекватна, то наступает пятый этап математического моделирования, который связан с эксплуатацией. На данном этапе математическую модель используют для решения задач, под которые она создавалась. Возможен авторский надзор с целью корректировки и дополнения разработанной математической модели.

На сегодняшний день одним из ведущих инструментов для математического моделирования технологических систем, процессов и объектов является программный комплекс MATLAB.

Примеры разработки математической модели физического процесса в пакете MATLAB

Пример 1. Математическое моделирование адиабатического теплообменника-смесителя переменного объема в пакете MATLAB. Схема адиабатического теплообменника-смесителя переменного объема, поясняющая движение входных и выходных потоков, представлена на рис. 85.

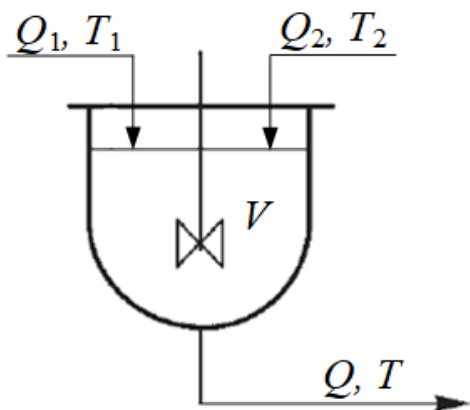


Рис. 85. Схема адиабатического теплообменника-смесителя переменного объема

Согласно условию задания, известно, что в теплообменник поступает два входных потока с объемными расходами Q_1 и Q_2 и с соответствующими температурами T_1 и T_2 . Из теплообменника отбирается один выходной поток с объемным расходом Q . Необходимо определить изменение объема теплоносителя в аппарате с течением времени $V(t)$ и изменение температуры теплоносителя на выходе аппарата с течением времени $T(t)$.

Разработка математической модели адиабатического теплообменника-смесителя переменного объема выполняется в соответствии с рассмотренными ранее этапами моделирования.

Цель моделирования вытекает из задания и заключается в необходимости определения изменения объема теплоносителя в аппарате с течением времени $V(t)$ и изменения температуры теплоносителя на выходе аппарата с течением времени $T(t)$.

Далее приступают к анализу процесса. Информационная схема адиабатического теплообменника-смесителя переменного объема как объекта моделирования представлен на рис. 86.

Согласно рисунку, выходом разрабатываемой модели будут являться объем теплоносителя в аппарате (V) и температура теплоносителя на выходе аппарата (T). В свою очередь, выходные переменные модели зависят от ряда входных, классификация которых представлена в табл. 2.

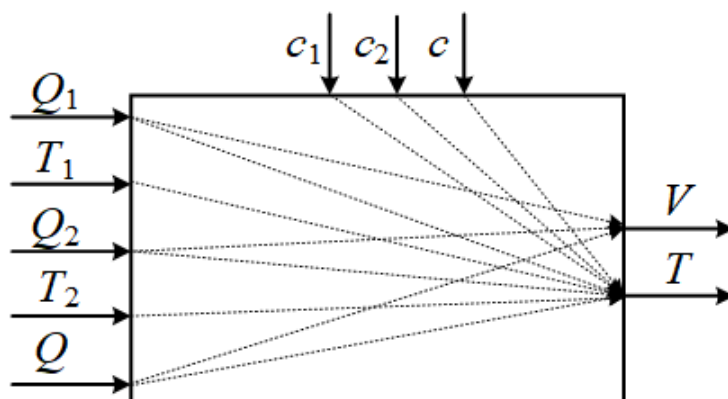


Рис. 86. Информационная схема адиабатического теплообменника-смесителя переменного объема как объекта моделирования

Таблица 2

Классификация переменных модели адиабатического теплообменника-смесителя переменного объема

№	Обозначение	Наименование	Тип	Единицы измерения
1	Q_1	Объемный расход первого входного потока	Входная переменная	м ³ /ч
2	Q_2	Объемный расход второго входного потока	Входная переменная	м ³ /ч
3	T_1	Температура первого входного потока	Входная переменная	°C
4	T_2	Температура второго входного потока	Входная переменная	°C
5	Q	Объемный расход выходного потока	Входная переменная	м ³ /ч
6	V	Объем теплоносителя в аппарате	Выходная переменная	м ³
7	T	Температура выходного потока	Выходная переменная	°C
8	c_1	Теплоемкость первого входного потока	Константа	Дж/°C
9	c_2	Теплоемкость второго входного потока	Константа	Дж/°C
10	c	Теплоемкость выходного потока	Константа	Дж/°C

На основании представленной схемы объекта моделирования (рис. 85) и выполненного анализа можно сделать следующие допущения:

- распределение поступающих в аппарат входных потоков происходит равномерно и мгновенно за счет имеющейся в аппарате мешалки;
- температура теплоносителя одинакова во всех точках аппарата и на выходе из него;
- теплоемкости входных потоков и выходного являются константами и равны между собой ($c_1 = c_2 = c$), так как используется один тип теплоносителя.

Разрабатываемая математическая модель адиабатического теплообменника-смесителя переменного объема должна отвечать следующим требованиям:

- адекватно описывать взаимосвязь входных и выходных переменных модели в динамическом режиме;
- позволять изменять значения входных переменных в процессе моделирования.

Далее математически описывается зависимость выходных переменных от входных. На основании сформулированных допущений может быть применена модель идеального перемешивания.

Согласно уравнению материального баланса, изменение объема теплоносителя в аппарате с течением времени $V(t)$ можно вычислить по следующей формуле

$$\frac{dV}{dt} = Q_1 + Q_2 - Q.$$

Согласно уравнению теплового баланса, изменение температуры теплоносителя в аппарате и на выходе из него с течением времени $T(t)$ определяется выражением вида:

$$c \cdot \frac{d(V \cdot T)}{dt} = Q_1 \cdot T_1 \cdot c_1 + Q_2 \cdot T_2 \cdot c_2 - Q \cdot T \cdot c. \quad (41)$$

В силу того, что $c_1 = c_2 = c$, формула (41) принимает следующий вид:

$$\begin{aligned} \frac{d(V \cdot T)}{dt} &= Q_1 \cdot T_1 \cdot \frac{c_1}{c} + Q_2 \cdot T_2 \cdot \frac{c_2}{c} - Q \cdot T \cdot \frac{c}{c}; \\ V \cdot \frac{dT}{dt} + T \cdot \frac{dV}{dt} &= Q_1 \cdot T_1 + Q_2 \cdot T_2 - Q \cdot T; \\ V \cdot \frac{dT}{dt} &= Q_1 \cdot T_1 + Q_2 \cdot T_2 - Q \cdot T - T \cdot (Q_1 + Q_2 - Q); \\ \frac{dT}{dt} &= \frac{Q_1 \cdot T_1 + Q_2 \cdot T_2 - T \cdot (Q_1 + Q_2)}{V}. \end{aligned}$$

В результате математическая модель адиабатического теплообменника-смесителя переменного объема может быть описана следующей системой дифференциальных уравнений:

$$\begin{cases} \frac{dV}{dt} = Q_1 + Q_2 - Q; \\ \frac{dT}{dt} = \frac{Q_1 \cdot T_1 + Q_2 \cdot T_2 - T \cdot (Q_1 + Q_2)}{V}. \end{cases} \quad (42)$$

Разработанная математическая модель (42) может быть решена с помощью метода Рунге–Кутты 4-го порядка точности в пакете MATLAB. Для этого можно использовать встроенный решатель *ode45*.

На рис. 87 представлен программный код, реализующий математическую модель адиабатического теплообменника-смесителя переменного объема.

```

1 - step=1;
2 - t_mod=4*3600;
3
4 - V = zeros(1,t_mod);
5 - V(1)=10;
6
7 - Q_in1 = zeros(1,t_mod);
8 - Q_in1(3600:7200)=1;
9 - Q_in1(7201:10800)=0;
10 - Q_in1(10801:t_mod)=3;
11 - T_in1 = zeros(1,t_mod);
12 - T_in1(1:7200)=20;
13 - T_in1(7201:t_mod)=80;
14
15 - Q_in2 = zeros(1,t_mod);
16 - Q_in2(3600:7200)=2;
17 - Q_in2(7201:10800)=0;
18 - Q_in2(10801:t_mod)=4;
19 - T_in2 = zeros(1,t_mod);
20 - T_in2(1:7200)=20;
21 - T_in2(7201:t_mod)=70;
22
23 - Q_out = zeros(1,t_mod);
24 - Q_out(7201:t_mod)=4;
25 - T_out = zeros(1,t_mod);
26 - T_out(1) = 50;
27
28 - j=2;
29 - for i=1:step:t_mod
30 -     options = odeset('InitialStep', step, 'MaxStep',step);
31 -     [T,Y] = ode45(@(t,y) sys_eq(t, y, Q_in1(i), Q_in2(i), Q_out(i), T_in1(i), T_in2(i)), ...
32 -         [i-1 i], [V(j-1) T_out(j-1)], options);
33 -     V(i)=Y(end, 1);
34 -     T_out(i)=Y(end, 2);
35 -     t(i)=T(end, 1);
36 -     j=i+1;
37 - end

```

```

1 function dy = sys_eq(t,y, Q_in1, Q_in2, Q_out, T_in1, T_in2)
2 dy = zeros(2,1); % a column vector
3
4 dy(1) = (Q_in1 + Q_in2 - Q_out)/3600;
5 dy(2) = (Q_in1*T_in1 + Q_in2*T_in2 - y(2)*(Q_in1+Q_in2))/y(1)/3600;

```

Рис. 87. Программный код, реализующий математическую модель адиабатического теплообменника-смесителя переменного объема

В строках с 1-й по 26-ю (рис. 87) задаются время и шаг моделирования, изменения входных потоков на протяжении всего времени

моделирования, а также начальные условия. В строках с 28-й по 37-ю выполняется решение разработанной математической модели (42) с помощью решателя *ode45*, который реализует метод Рунге–Кутты 4-го порядка точности. Разработанная математическая модель адиабатического теплообменника-смесителя переменного объема (42) реализована в виде специальной функции *sys_eq*.

Для графического представления полученных результатов моделирования используется команда *plot* (рис. 88).

```

39 - figure
40 - subplot(2, 2, 1)
41 - plot(t/3600, Q_in1)
42 - title('Расход теплоносителя на входе аппарата (поток №1)', 'FontName', 'Time')
43 - ylabel('Q_в_х_1, м^3/ч', 'FontName', 'Time')
44 - ylim([0.9*min(Q_in1) 1.1*max(Q_in1)])
45 - xlabel('t, ч', 'FontName', 'Time')
46 - xlim([0 max(t/3600)])
47 - grid on
48 - subplot(2, 2, 2)
49 - plot(t/3600, Q_in2)
50 - title('Расход теплоносителя на входе аппарата (поток №2)', 'FontName', 'Time')
51 - ylabel('Q_в_х_2, м^3/ч', 'FontName', 'Time')
52 - ylim([0.9*min(Q_in2) 1.1*max(Q_in2)])
53 - xlabel('t, ч', 'FontName', 'Time')
54 - xlim([0 max(t/3600)])
55 - grid on
56
57 - subplot(2, 1, 2)
58 - plot(t/3600, Q_out)
59 - title('Расход теплоносителя на выходе аппарата', 'FontName', 'Time')
60 - ylabel('Q_в_ы_х, м^3/ч', 'FontName', 'Time')
61 - ylim([0.9*min(Q_out) 1.1*max(Q_out)])
62 - xlabel('t, ч', 'FontName', 'Time')
63 - xlim([0 max(t/3600)])
64 - grid on
65
66 - figure
67 - plot(t/3600, V)
68 - title('Объем теплоносителя в аппарате', 'FontName', 'Time')
69 - ylabel('V, м^3', 'FontName', 'Time')
70 - ylim([0.98*min(V) 1.02*max(V)])
71 - xlabel('t, ч', 'FontName', 'Time')
72 - xlim([0 max(t/3600)])
73 - grid on

```

Рис. 88. Программный код для графического представления результатов моделирования

На рис. 89–91 представлены результаты моделирования адиабатического теплообменника-смесителя переменного объема. Они описывают взаимосвязь входных и выходных переменных модели в динамическом режиме. При этом имеется возможность изменять значения входных переменных в процессе моделирования.

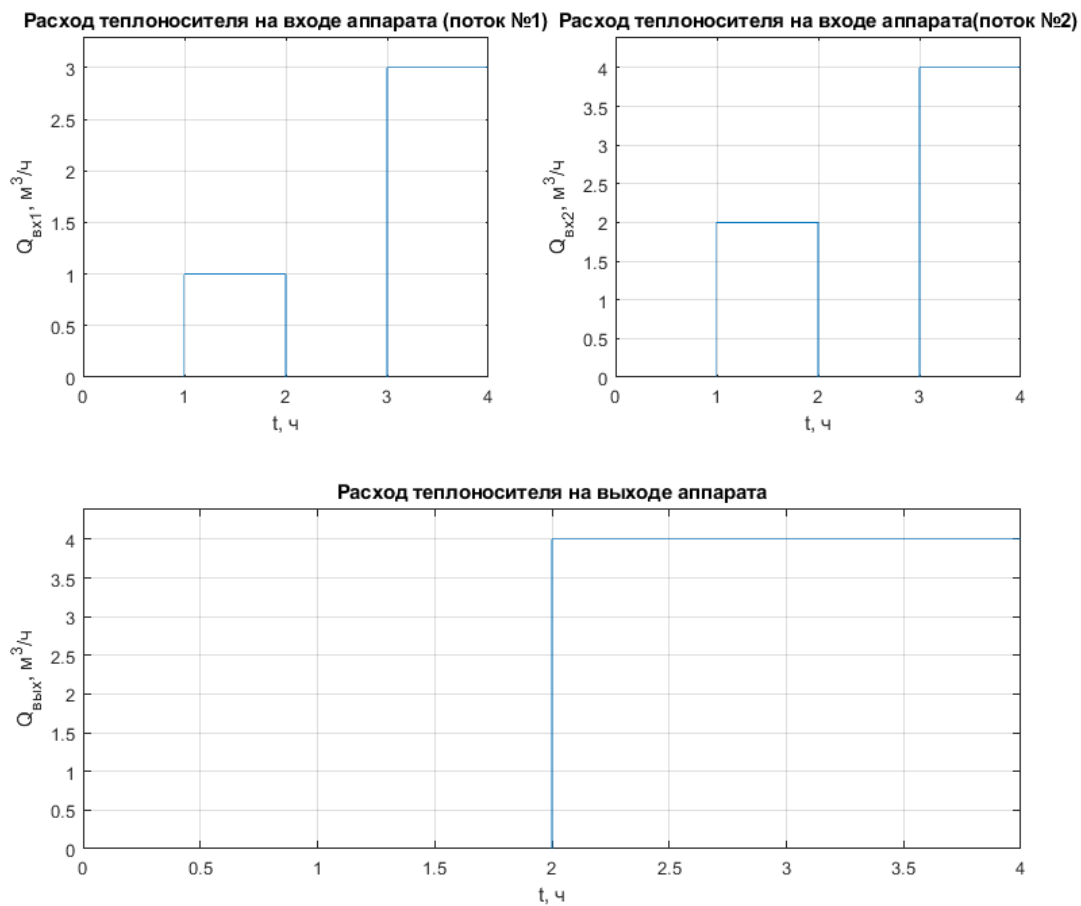


Рис. 89. Изменения объемных расходов входных и выходного потоков с течением времени

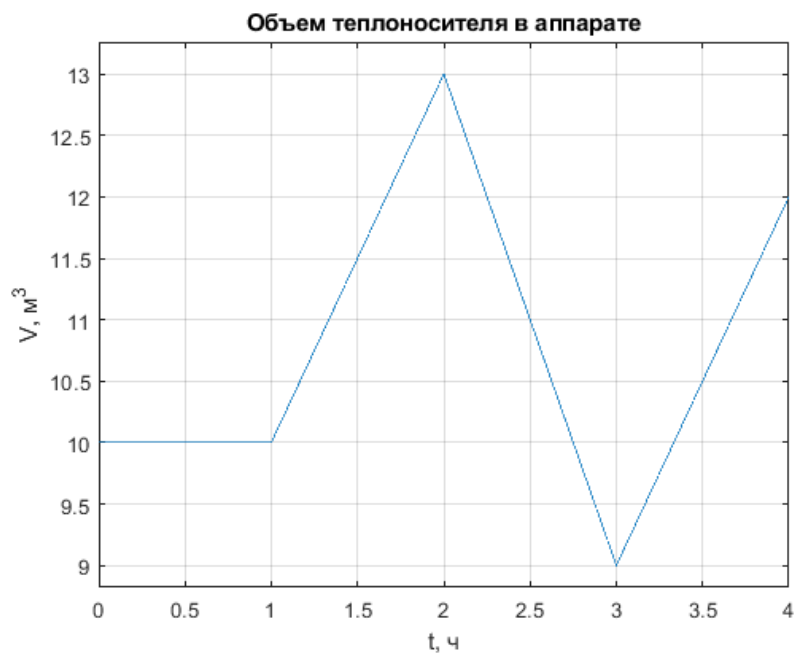


Рис. 90. Изменения объема теплоносителя в аппарате с течением времени

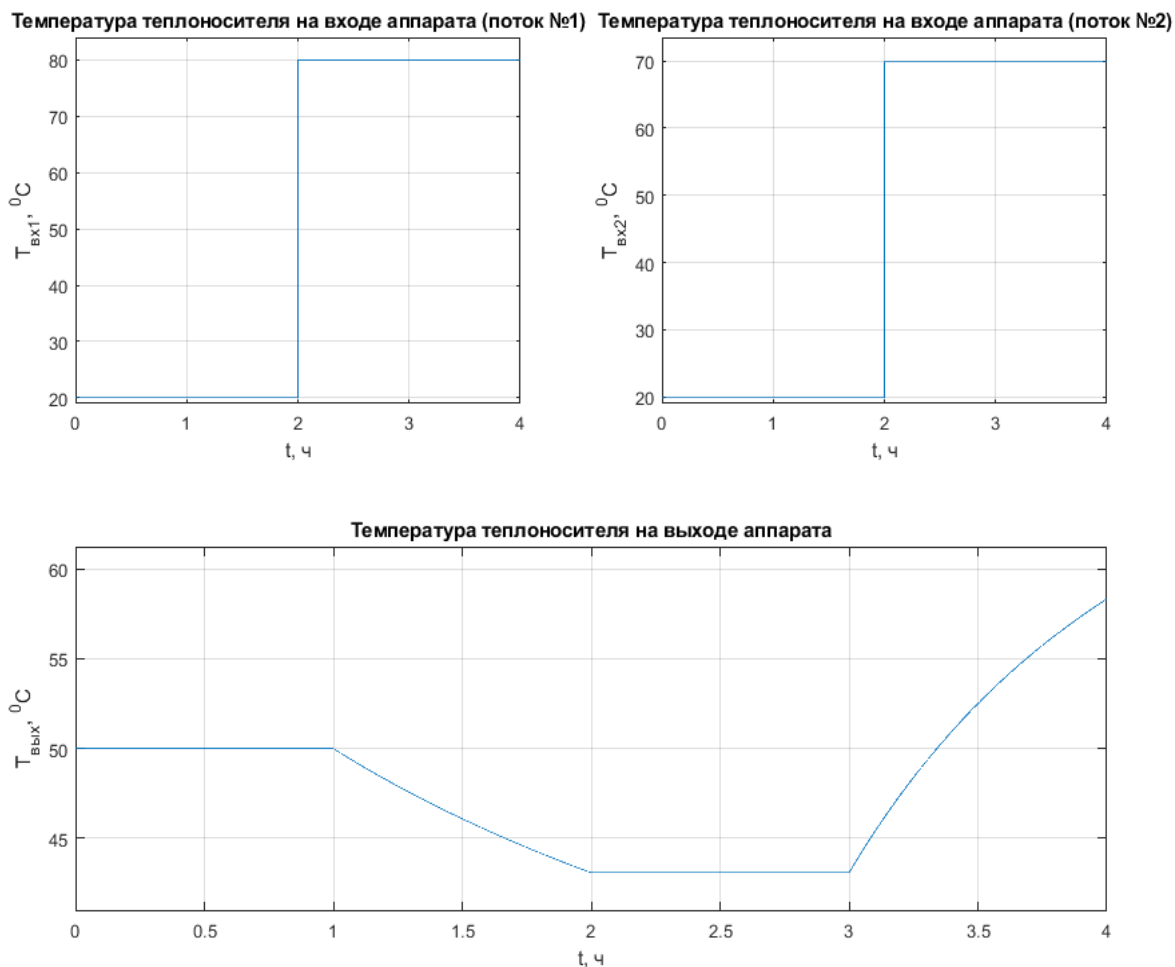


Рис. 91. Изменение температуры во входных и выходном потоках с течением времени

Для проверки адекватности разработанной математической модели адиабатического теплообменника-смесителя переменного объема оценивается соблюдение материального и теплового балансов. На рис. 89 показано, что в начальный момент времени $t = 0$ расходы теплоносителя на входе и выходе аппарата равны нулю, при этом не изменяется объем теплоносителя в аппарате (рис. 90) и его температура (рис. 91). Затем в момент времени $t = 1$ объемный расход первого потока теплоносителя на входе аппарата соответствует $1 \text{ м}^3/\text{ч}$, второго – $2 \text{ м}^3/\text{ч}$, при этом расход теплоносителя на выходе аппарата равен $0 \text{ м}^3/\text{ч}$ (рис. 89). В результате в течение часа (с $t = 1$ до $t = 2$) увеличивается объем теплоносителя в аппарате с 10 до 13 м^3 (рис. 90). С учетом того, что температура поступающего теплоносителя ниже температуры теплоносителя в аппарате, происходит снижение температуры теплоносителя в аппарате (рис. 91). Далее в момент времени $t = 2$ расходы теплоносителя на входе аппарата становятся равными нулю, а расход теплоносителя на выходе аппарата

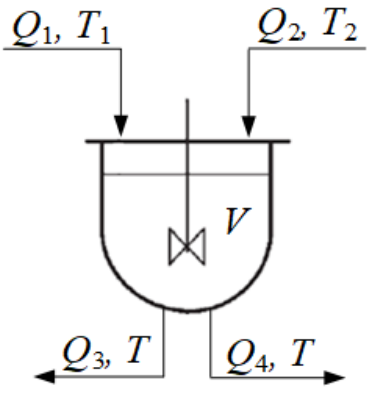
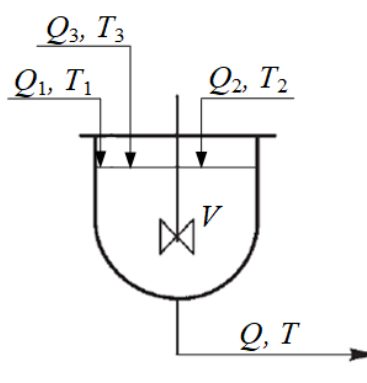
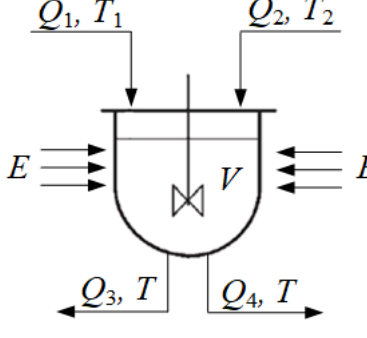
равен $4 \text{ м}^3/\text{ч}$ (рис. 89). В результате в течение часа (с $t = 2$ до $t = 3$) уменьшается объем теплоносителя в аппарате с 13 до 9 м^3 (рис. 90). С учетом того, что теплоноситель не поступает в аппарат, температура теплоносителя в аппарате и на выходе из него не изменяется (рис. 91). Затем в момент времени $t = 3$ объемный расход первого потока теплоносителя на входе аппарата становится равным $3 \text{ м}^3/\text{ч}$, второго – $4 \text{ м}^3/\text{ч}$, при этом расход теплоносителя на выходе аппарата равен $4 \text{ м}^3/\text{ч}$ (рис. 89). В результате в течение часа (с $t = 3$ до $t = 4$) увеличивается объем теплоносителя в аппарате с 9 до 12 м^3 (рис. 90). С учетом того, что температура поступающего теплоносителя выше температуры теплоносителя в аппарате, увеличивается температура теплоносителя в аппарате (рис. 91). Таким образом, материальный и тепловой балансы соблюдаются. Соответственно, можно говорить об адекватности разработанной математической модели адиабатического теплообменника-смесителя переменного объема.

Таким образом, разработанная математическая модель готова для дальнейшего использования в пакете MATLAB. В случае необходимости она может быть реализована на любом другом языке программирования.

Индивидуальные задания

1. В соответствии с индивидуальным вариантом произвести математическое моделирование теплообменника-смесителя переменного объема в пакете MATLAB.

Номер варианта	Описание процесса
1	В теплообменник поступает два входных потока с объемными расходами Q_1 и Q_2 и с соответствующими температурами T_1 и T_2. Из теплообменника отбирается один выходной поток с объемным расходом Q. Извне к теплообменнику-смесителю подводится тепловой поток с энергией E. Необходимо определить изменение объема теплоносителя в аппарате с течением времени $V(t)$ и изменение температуры теплоносителя на выходе аппарата с течением времени $T(t)$

Номер варианта	Описание процесса
2	 В адиабатический теплообменник поступает два входных потока с объемными расходами Q_1 и Q_2 и с соответствующими температурами T_1 и T_2. Из теплообменника отбирается два выходных потока с объемными расходами Q_3 и Q_4. Необходимо определить изменение объема теплоносителя в аппарате с течением времени $V(t)$ и изменение температуры теплоносителя на выходе аппарата с течением времени $T(t)$
3	 В адиабатический теплообменник поступает три входных потока с объемными расходами Q_1, Q_2 и Q_3 и с соответствующими температурами T_1, T_2 и T_3. Из теплообменника отбирается один выходной поток с объемным расходом Q. Необходимо определить изменение объема теплоносителя в аппарате с течением времени $V(t)$ и изменение температуры теплоносителя на выходе аппарата с течением времени $T(t)$
4	 В теплообменник поступает два входных потока с объемными расходами Q_1 и Q_2 и с соответствующими температурами T_1 и T_2. Из теплообменника отбирается два выходных потока с объемными расходами Q_3 и Q_4. Извне к теплообменнику-смесителю подводится тепловой поток с энергией E. Необходимо определить изменение объема теплоносителя в аппарате с течением времени $V(t)$ и изменение температуры теплоносителя на выходе аппарата с течением времени $T(t)$

2. При разработке математической модели выполнить и описать все основные этапы процесса разработки в соответствии со схемой (рис. 81).

3. Подготовить отчет в соответствии с требованиями, представленными в прил. А–В. Проанализировать полученные результаты и сделать выводы.

Контрольные вопросы и задания

1. Перечислите основные этапы разработки математической модели.
2. Дайте определения моделям с распределенными и сосредоточенными параметрами.
3. В чем заключается ключевое отличие модели с распределенными параметрами от модели с сосредоточенными параметрами?
4. Дайте определение модели идеального перемешивания.
5. Запишите формулу модели идеального перемешивания.
6. Дайте определение модели идеального вытеснения.
7. Запишите формулу модели идеального вытеснения.
8. В чем заключается проверка математической модели на адекватность?
9. Какие существуют способы проверки математической модели на адекватность?

Список рекомендованной литературы

1. Гартман Т.Н. Моделирование химико-технологических процессов. Принципы применения пакетов компьютерной математики : учебное пособие / Т.Н. Гартман, Д.В. Клушин. – Санкт-Петербург : Лань, 2020. – 404 с.
2. Луценко В.А. Аналоговые вычислительные машины в химии и химической технологии. – 2-е изд., перераб. и доп. / В.А. Луценко, Л.Н. Финякин. – Москва : Химия, 1979. – 248 с.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

L_n – интерполяционный полином Лагранжа.

I – определенный интеграл заданной функции.

$\frac{df}{dx}$ – производная функции одной переменной $f(x)$ по переменной x .

$\frac{\partial f}{\partial x}$ – частная производная функции нескольких переменных $f(x, y)$

по переменной x .

МНК – метод наименьших квадратов.

МЛП – метод левых прямоугольников.

МПП – метод правых прямоугольников.

МСП – метод средних прямоугольников.

Метод РК-3 – метод Рунге–Кутты третьего порядка точности.

Метод РК-4 – метод Рунге–Кутты четвертого порядка точности.

СКО – среднеквадратичное отклонение.

МКЭ – метод конечных элементов.

МКР – метод конечных разностей.

САУ – система алгебраических уравнений.

ЭВМ – электронная вычислительная машина.

ПРИЛОЖЕНИЯ

Приложение А

Требования и рекомендации по оформлению отчета по лабораторной работе

Отчет по лабораторной работе выполняется на бумаге формата А4 (210 × 297 мм). Текст отчета размещается на одной стороне листа.

К оформлению текста предъявляются следующие требования:

- междустрочный интервал – 1,5 пт;
- шрифт – Times New Roman;
- размер шрифта – 12–14 пт;
- цвет шрифта – черный;
- выравнивание текста – по ширине;
- абзац – отступ 1,25 см;
- размеры полей – левое – 30 мм, правое – 15 мм, верхнее и нижнее – 20 мм.

В случае необходимости допускается размещение таблиц и графиков на листах с альбомной ориентацией.

Для оформления формул использовать редактор формул MathType.

Дополнительные требования и рекомендации по оформлению отчета по лабораторной работе представлены в стандарте организации СТО ТПУ 2.5.01.

Примеры оформления рисунков, формул и таблиц в отчете по лабораторной работе

При наличии в тексте рисунка на него обязательно должна быть ссылка до его размещения. Рисунок выравнивается по центру страницы и подписывается (рис. 92).

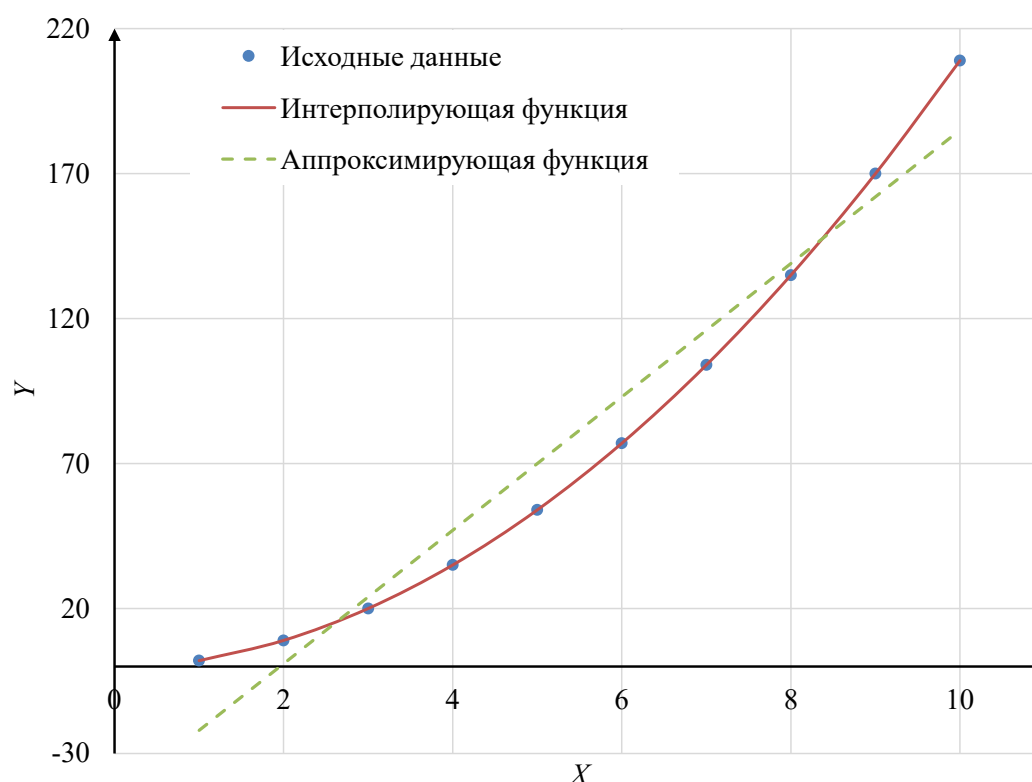


Рис. 92. Пример оформления рисунков

Оси на рисунке должны быть подписаны и иметь шкалу. Если на графике представлено несколько кривых, то должен быть поясняющий текст, однозначно определяющий каждую кривую. Также для разграничения нескольких кривых на одном графике необходимо использовать разные цвета либо штрихпунктирные линии.

Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Если уравнение не уместится в одну строку, оно должно быть перенесено после знака равенства или после знаков плюс, минус, умножения, деления или других математических знаков. На новой строке знак повторяется.

Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности,

зв которой они представлены в формуле. Значение каждого символа и числового коэффициента необходимо приводить с новой строки. Первую строку пояснения начинают со слова «где» без двоеточия с абзаца.

Формулы нумеруются арабскими цифрами сквозной нумерацией. В формулах прямой шрифт используется для цифр, букв русского алфавита, обозначений химических элементов, математических функций и международных обозначений единиц измерения физических величин. Для всего остального (буквы латинского и греческого алфавита) применяется наклонный шрифт.

Пример оформления формул представлен выражением (43):

$$m(t) = V \cdot \rho(t), \quad (43)$$

где V – объем вещества, м³; $\rho(t)$ – плотность вещества, кг/м³.

Ссылки в отчете на порядковые номера формул приводятся в скобках: в формуле (43).

При наличии в тексте таблицы на нее обязательно должна быть ссылка до ее размещения. Таблица выравнивается по центру страницы и подписывается (табл. 3).

Таблица 3

Пример оформления таблицы

№	Заголовок 1	Заголовок 2	Заголовок 3
1			
2			

Приложение В

Пример оформления титульного листа отчета по лабораторной работе

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Школа – Инженерная школа ядерных технологий
Отделение – Отделение ядерно-топливного цикла
Специальность – Электроника и автоматика физических установок

Отчет по лабораторной работе № 1
«Изучение методов восстановления функции одной переменной»
по дисциплине
«Математическое моделирование физических процессов»

Выполнил студент гр. 0772

«__» ____ 20__

Иванов И.И.

Проверил доцент ОЯТЦ

«__» ____ 20__

Петров П.П.

Томск 2022

Учебное издание

НАДЕЖДИН Игорь Сергеевич
ГОРЮНОВ Алексей Германович

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО КУРСУ
«МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
ФИЗИЧЕСКИХ ПРОЦЕССОВ»**

Учебно-методическое пособие

Корректурa *Е.Л. Тен*
Компьютерная верстка *К.С. Чечельницкая*
Дизайн обложки *Т.В. Буланова*

Подписано к печати 09.06.2023. Формат 60×84/16. Бумага «Снегурочка».
Печать CANON. Усл. печ. л. 6,46. Уч.-изд. л. 5,84.
Заказ 188-23. Тираж 100 экз.



ИЗДАТЕЛЬСТВО
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ