МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра Телевидения и управления (ТУ)

Утверждаю:
Зав. кафедрой ТУ
Д.т.н., с.н.с.
_____Т.Р. Газизов
«___»_____2018 г.

Информационные технологии

Учебно-методическое пособие по выполнению лабораторных
и самостоятельных работ для бакалавров
направления подготовки 11.03.01 «Радиотехника»

Разработчики:

Доцент каф. ТУ, к.т.н.

_____С.П. Куксенко

Ассистент каф. ТУ

_____А.В. Демаков

Томск – 2018

Куксенко С.П., Демаков А.В.

Информационные технологии: учеб. метод. пособие. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2018. – 57 с.

Представлены методические материалы лабораторного практикума, посвященного изучению вычислительных методов, используемых при обработке данных и результатов измерений физических величин, моделировании физических процессов и построении математических моделей радиотехнических средств и систем. Также приведены методические указания по проведению самостоятельных работ.

Предназначено для бакалавров технических вузов, обучающихся по направлению подготовки 11.03.01 «Радиотехника» в рамках дисциплины «Информационные технологии».

Содержание

Введение

Современный инженер радиотехнического профиля в своей работе все чаще сталкивается с задачами, требующими знания методов обработки результатов измерений, планирования эксперимента, а также математических методов моделирования и оптимизации. Это обусловлено широким использованием математического моделирования и сопутствующего информационного обеспечения в виде различных специализированных пакетов и программ, позволяющих значительно сократить как финансовые, так и временные затраты на разработку тех или иных средств и систем.

В общем случае, в основе математического моделирования лежит численный анализ. Одним из этапов моделирования является дискретизация модели, которая заключается в переходе от непрерывных функций к дискретным и от функциональных уравнений к системе линейных алгебраических уравнений (СЛАУ), с помощью численных методов. Таким образом, от выбора наиболее предпочтительного численного метода и знания его программной реализации зависит эффективность всего процесса моделирования.

В инженерной практике также часто приходится оперировать с таблично заданными данными (функциями), что затрудняет восприятие и накладывает ряд ограничений на возможные варианты их обработки. Для аналитического описания подобной информации применяются различные методы аппроксимации, которые, в свою очередь, не являются универсальными для описания всех возможных зависимостей величин. Таким образом, отработка навыков использования и программной реализации методов аппроксимации, численного интегрирования и дифференцирования является необходимой при обучении студентов.

Поэтому освоение широко используемых численных методов, а также получение навыков использования специализированных пакетов прикладных программ является неотъемлемым этапом подготовки специалистов

радиотехнического профиля.

В пособии приведены основные сведения о пакте Octave и примеры работы в нем. Также приведены методические указания по изучению методов решения СЛАУ, численного интегрирования и дифференцирования, а также интерполяции и аппроксимации. По каждой лабораторной работе содержится краткое описание теоретических сведений, общее представление об используемом алгоритме и перечень заданий для самостоятельного выполнения. Также приведены методические указания к самостоятельным работам.

1 Методические указания к лабораторным работам

1.1 Пакет прикладных программ GNU Octave

Цель работы – получение навыков работы и освещение принципов реализации программного кода в пакете прикладных программ GNU Octave.

1.1.1 Краткая справка

GNU Octave (далее по тексту Octave) – это свободно распространяемый язык программирования высокого уровня, ориентированный на проведение численных расчетов, и, по сути, являющийся альтернативой коммерческому пакету MATLAB. Octave обладает богатым инструментарием для решения задач линейной алгебры, нахождения корней нелинейных уравнений, решения дифференциальных уравнений, вычисления интегралов, решения линейных и нелинейных оптимизационных задач, построения графиков и т.д.

После запуска Octave (версия 4.2.1) пользователь видит окно интерпретатора (рисунок 1.1). Перечислим основные инструменты интерфейса пользователя:

1. Область главного меню. Позволяет получить доступ ко всем возможным командам и интерфейсам.

2. Область редактора. Позволяет вносить необходимые изменения в программы, созданные пользователем или импортированные им из внешних источников. Здесь под программой понимается текстовый файл, содержащий команды для среды Octave с расширением .m. Содержащиеся в файлах команды последовательно передаются на исполнение системой через командное окно.

3. *Командное окно*/*Редактор*. В окне командное окно (интерпретатор) пользователь может вводить как отдельные команды языка Octave, так и группы команд. Группы команд удобно объединять в программы (окно *Редактор*). Содержащиеся в файлах команды последовательно передаются на исполнение системой через *Командное окно* (результат выполнения программы также выводится в командном окне).

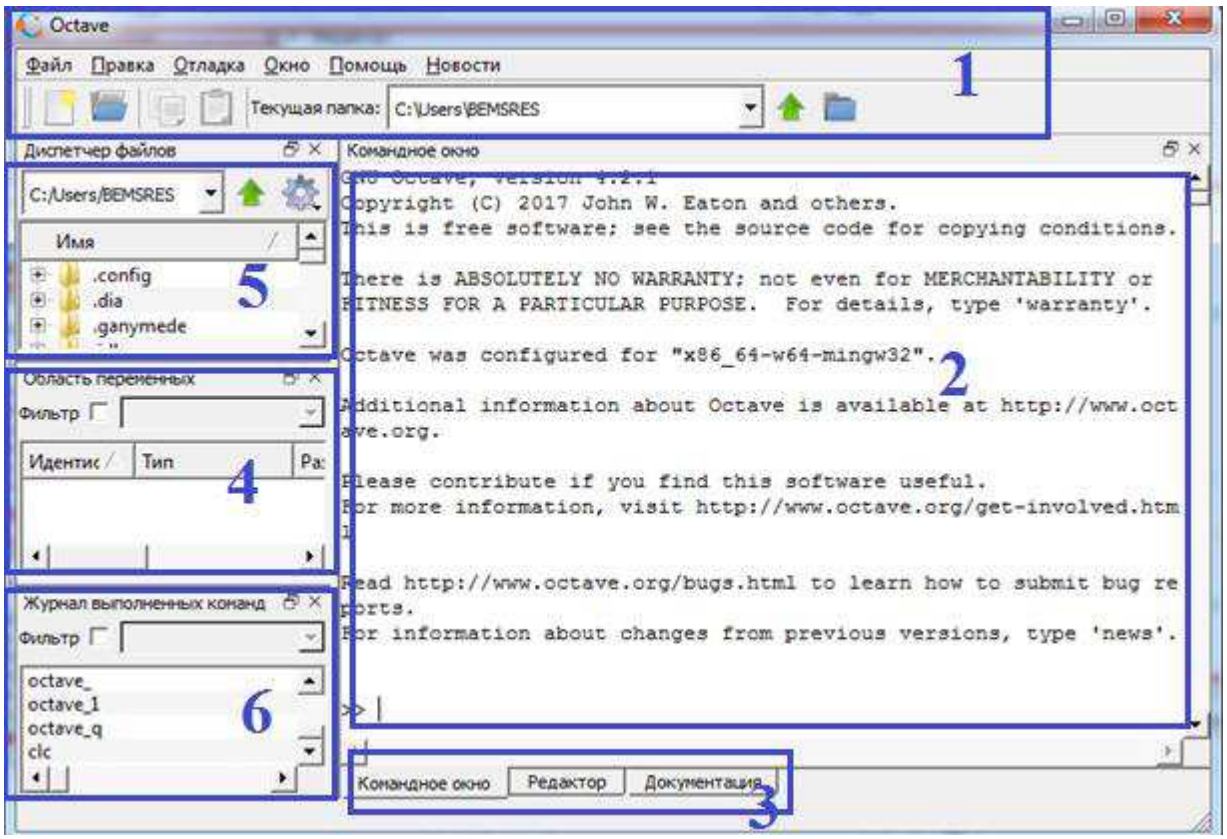

Рисунок 1.1 – Графическое окно Octave

4. Область переменных. В этой области пользователь может следить за значением и типом данных переменных. Такой контроль позволяет контролировать правильность расчетов.

5. Файловый менеджер.

6. Журнал выполненных команд. Содержит список команд и информацию об их выполнении.

При необходимости пользователь может закрывать ненужные ему окна либо открывать новые (см. вкладка *Окно* области главного меню). Возможны два варианта решения любой задачи в Octave:

1. Терминальный режим (вкладка *Командное окно*). В этом режиме в окно интерпретатора последовательно вводятся отдельные команды.

2. Программный режим (вкладка *Редактор*). В этом режиме создаётся текстовый файл с расширением *.m*, в котором хранятся последовательно выполняемые команды Octave. Затем эта программа запускается на выполнение.

Операция присваивания «=» передает значение выражения в левой части равенства переменной, стоящей справа. Так, напишем в командной строке (далее для облегчения восприятия все приводимые примеры будут показаны с помощью курсивного выделения)

a=5;

Результат выполнения команды не будет отражен в командном окне, так как в конце выражения стоит оператор отключения вывода «;». При вводе данного выражения в командную строку переменной *a* будет присвоено значение 5. Тем не менее, *a* с точки зрения интерпретатора является матрицей размером 1×1. Убедиться в этом можно, дав команду (или воспользоваться областью переменных)

>> size(a)
ans =
1 1

Т.е. команда *size*(*a*) предназначена для определения размерность матрицы *a*.

Имя переменной не должно совпадать с именами встроенных процедур, функций и встроенных переменных системы. Система различает большие и малые буквы в именах переменных, т.е. ABC, abc, Abc, aBc и тд. – это имена разных переменных.

Если команда не содержит знака присваивания, то по умолчанию вычисленное значение присваивается специальной системной переменной *ans*. Причем полученное значение можно использовать в последующих вычислениях, но важно помнить, что значение *ans* изменяется после каждого вызова команды без оператора присваивания.

Кроме переменной *ans* в Octave существуют и другие системные переменные, которые можно использовать в математических выражениях:

- i, j – мнимая единица;
- pi – число π;
- *e* –экспонента, 2.71828183;
- realmin – наименьшее число с плавающей точкой (2.2251e-308);
- realmax – наибольшее число с плавающей точкой (1.7977e+308).

- inf – машинный символ бесконечности;
- NaN – неопределенный результат (0/0 или ∝/∝ и тд.).

Как было показано выше, возможно выполнение операции присвоения сразу ко всем элементам матрицы:

```
>> B=[1,2,3]
B =
     1 2 3
```

Команда формирует вектор-строку (матрицу размерности 1×3). При введении данных через «;» будет сформирован вектор-столбец:

```
>> C=[4;5;6]
C =
     4
     5
     6
```

Аналогичным образом, формируются матрицы любой размерности.

Если значения некоторой величины меняются с равным шагом, то оператор присваивания может быть использован совместно с оператором «:», определяющим пределы изменения величины. Так, шаблон команды будет выглядеть следующим образом: ПЕРЕМЕННАЯ=НАЧАЛЬНОЕ ЗНАЧЕНИЕ:ШАГ:КОНЕЧНОЕ ЗНАЧЕНИЕ. Значение шага также задается пользователем, но при его отсутствии он полагается равным 1. Например,

```
>> E=1:4:17
E =
     1 5 9 13 17.
```

Оператор «:» имеет еще одно применение – он может определять совокупность строк или столбцов в матрице. Для демонстрации сначала создадим нулевую матрицу *D* размера 4×4:

```
>> D=zeros(4,4)
D =
     0   0   0   0
     0   0   0   0
     0   0   0   0
     0   0   0   0
```

Затем выполним команду

```
>> D(1,:)=1
```

D =

1 1 1 1
0 0 0 0
0 0 0 0
0 0 0 0

Так, с помощью оператора «:» произошло присвоение значения 1 всем элементам первой строки матрицы *D*. Еще один пример использования данного оператора приведен ниже.

>> D(3:4,2:3)=-1
D =

1 1 1 1
0 0 0 0
0 -1 -1 0
0 -1 -1 0

Для очистки командного окна можно воспользоваться командой *clc*, а для удаления из памяти всех переменных или конкретных переменных командами *clear* и *clear имя_переменной* соответственно. Простейшие арифметические операции в Octave выполняют с помощью следующих операторов:

+ сложение;

– вычитание;

* умножение;

/ деление слева направо;

\ деление справа налево;

^ возведение в степень.

Если вычисляемое выражение слишком длинное, то перед нажатием клавиши ENTER следует набрать три или более точек. Это будет означать продолжение текущей (командной) строки:

>> 1+2+...
4+5
ans = 12

Допускается переназначение переменных не только на уровне значений, но и на уровне типов данных. Например, вещественной переменной можно присвоить символьное значение – при такой операции автоматически произойдет смена ее типа. Интерпретаторы команд отслеживают только

согласование размерности в выражении (умножение символьной строки на матрицу будет вызывать ошибку). Пример: действие оператора присваивания:

```
>> A=[3 4; 7 5];
>> A='line';
>> B=[1 2; 3 4];
>> A*B
error: operator *: nonconformant arguments (op1 is 1x4, op2 is 2x2)
```

При использовании матричного типа данных нужно помнить, что при этом невозможно, например, выполнить сложение или вычитание для матриц разной размерности. Требования к размерности матриц также ограничивают применение операций умножения. Для удобства пользователей некоторые математические операции могут быть также выполнены поэлементно. Указание на поэлементное выполнение дается добавлением знака «.» к оператору. Так, рассмотрим пример.

```
>> A=[1,2;3,4];
>> B=A*A
B =
        7 10
        15 22
>> C=A.*A
C =
        1 4
        9 16
```

В первом случае матрица **A** была умножена сама на себя (выполнено матричное умножение), во втором случае сам на себя был умножен каждый элемент матрицы (поэлементное возведение в квадрат).

Числовые результаты могут быть представлены с плавающей (3.2E-9, 6.42E+3), или с фиксированной (4.12, 17.1389) точкой. Числа в формате с плавающей точкой представлены в экспоненциальной форме $mE \pm p$, где m – мантисса (целое или дробное число с десятичной точкой), p – порядок (целое число). Для того чтобы перевести число в экспоненциальной форме к обычному представлению с фиксированной точкой, необходимо мантиссу умножить на десять в степени порядок. Далее приведен пример использование различных

комбинаций задания вещественного числа.

```
>> 5.49e-2
ans =  0.054900
>> 5.49E-2
ans =  0.054900
>> 5.49.*1E-2
ans =  0.054900
>> 5.49.*10^-2
ans =  0.054900
>> 5.49*10^-2
ans =  0.054900
```

Рассмотрим пример ввода числа.

```
>> 0.987654321
ans =  0.98765
```

Видно, что число знаков в дробной части числа в строке ввода больше чем в строке вывода. Происходит это потому, что результат вычислений выводится в виде, который определяется предварительно установленным форматом представления чисел. Команда, с помощью которой можно установить формат числа имеет вид: *format* формат числа. В Octave предусмотрены следующие форматы чисел (приведены используемые в ходе занятий):

- Short – краткая запись, применяется по умолчанию.
- Long – длинная запись.
- Short E (Short e) – краткая запись в формате с плавающей точкой.
- Long E (Long e) – длинная запись в формате с плавающей точкой.

Продемонстрируем использование форматов на примере вывода числа π.

```
>> format short
>> pi
ans =  3.1416
>> format long
>> pi
ans =  3.14159265358979
>> format short E
>> pi
ans =   3.1416E+000
>> format long E
>> pi
ans =   3.14159265358979E+000
```

Имеются также тригонометрические функции и экспоненциальные функции exp(x) – экспонента числа x и log(x) – натуральный алгоритм числа *x*. Рассмотрим пример.

```
>> x=pi/2
>> x =  1.5708
>> sin(x)
ans =  1
>> cos(x)
ans =  6.1230e-017
>> tan(x)
ans =  1.6331e+016
>> exp(x)
ans =  4.8105
>> log(x)
ans =  0.45158
```

Также имеются целочисленные функции:

- fix(x) – округление числа *x* до ближайшего целого в сторону нуля;
- floor(x) – округление числа *x* до ближайшего целого в сторону отрицательной бесконечности;
- ceil(x) – округление числа *x* до ближайшего целого в сторону положительной бесконечности;
- round(x) – округление числа *x* до ближайшего целого;
- rem(x, y) – вычисление остатка от деления *x* на *y*;
- sign(x) – выдает 0, если *x*=0, –1 при $x < 0$ и 1 при $x > 0$;

и другие элементарные функции:

- sqrt(x) – корень квадратный из числа *x*;
- abs(x) – модуль числа *x*;
- log10(x) – десятичный логарифм от числа *x*;
- log2(x) – логарифм по основанию два от числа *x*;
- pow2(x) – возведение двойки в степень *x*;
- gcd(x, y) – наибольший общий делитель чисел *x* и *y*;
- lcm(x, y) – наименьшее общее кратное чисел *x* и *y*;
- rats(x) – представление числа *x* в виде рациональной дроби.

Рассмотрим реализацию комплексных чисел. Как было отмечено выше для обозначения мнимой единицы используются *i* и/или *j*. Ввод комплексного числа производится в следующем формате:

действительная_часть + i*мнимая_часть.

К комплексным числа применимы элементарные арифметические операции: +, –, *, \, /, ^. Функции для работы с комплексными числами:

- real(z) – выдает действительную часть комплексного аргумента *z*;
- imag(z) – выдает мнимую часть комплексного аргумента *z*;
- angle(z) – вычисляет значение аргумента комплексного числа *z* в радианах от –π до π;
- conj(z) – выдает число комплексно сопряженное Z.

Рассмотрим операции отношения, предназначены для выполнения сравнения двух операндов и определяют, истинно выражение или ложно. Результат операции отношения – логическое значение. В качестве логических значений используются 1 («истина») и 0 («ложь»). Операции отношения:

- < – меньше;
- > – больше;
- == – равно;
- ~= – не равно;
- <= – меньше или равно;
- >= – больше или равно.

В Octave существует возможность представления логических выражений в виде логических операторов и логических операций (таблица 1.1).

Таблица 1.1 – Виды логических выражений

Тип выражения	Выражение	Логический оператор	Логическая операция
Логическое «и»	A and B	and(A, B)	A & B
Логическое «или»	A or B	or(A, B)	A | B
Исключающее «или»	A xor B	xor(A,B)	
Отрицание	not A	not (A)	~ A

Оператор *for* предназначен для выполнения заданного числа повторяющихся действий. Самое простое использование оператора *for* осуществляется следующим образом:

for count = start:step:final

%count-переменная цикла,

%start-ее начальное значение

%final-конечное значение

%step-шаг, на который увеличивается count при каждом

%следующем заходе цикла

%тело цикла

end

1.1.2 Порядок выполнения работы

1. Запустить Octave, создать и сохранить новый *m*-файл.

2. В созданном *m*-файле создать переменные: *a*, *b*, *d*, *z*, присвоив им случайные значения в диапазоне от 1 до 100.

3. Выполнить следующие арифметические операции и сохранить их результат в новые переменные:

$$a + \frac{b}{d^z},$$

$$a^3 + b^d \cdot \left(d \cdot b + d^a\right),$$

$$z + \pi \cdot a \cdot b\sqrt{d \cdot z}\,.$$

4. Создать вектор-строки **x** = [1 1.1 1.2] и **y** = [1 2 3]. Транспонировать их. Найти сумму векторов **x** и **y**, разность, скалярное произведение, линейную комбинацию с коэффициентами 2 и 4 соответственно. Полученные результаты свести в таблицу.

5. Увеличить число элементов вектора **x** до пяти. Присвоить четвертому элементу вектора **x** разность первого и второго элемента вектора **y**, пятому элементу присвоить максимальное значение, содержащееся в векторе **x**.

Минимальное значение вектора **y** заменить на сумму четвертого и пятого элемента вектора **x**. Вывести содержимое векторов **x** и **y** в командное окно.

6. Создать матрицы **L** и **M** размерностью 3×3, заполненные нулями и единицами соответственно. Вывести содержимое матриц в командное окно.

7. Переопределить и вывести в командное окно элементы матрицы **L** с помощью цикла *for* согласно выражению:

$$\mathbf{L}_{i,j} = i + 2j\,.$$

8. Для переопределенной матрицы **L** найти ранг, определитель, обратную матрицу. Вывести результаты в командное окно.

9. С помощью условия *if* проверить выполнение неравенства и записать в переменную результат арифметической операции:

$$l_{1,2} + \frac{l_{1,1}}{3} + z\,, \text{ если } z > 25,$$

$$\text{rank}(\mathbf{L}) + 25 \cdot l_{3,1} - \frac{4}{a \cdot b}\,, \text{ если } z \le 25.$$

10. Оформить отчет.

1.1.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

1.2 Метод Гаусса

Цель работы – изучение и программная реализация метода Гаусса.

Для достижения поставленной цели исследуется алгоритм, состоящий из прямого и обратного хода метода, и его программная реализация в среде Octave.

1.2.1 Краткая теоретическая справка

Пусть дана система линейных алгебраических уравнений (СЛАУ), состоящая из n-уравнений с n-неизвестными:

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n = b_1 \\ \cdots \\ a_{n1}x_1 + \ldots + a_{nn}x_n = b_n \end{cases}. \qquad (1.1)$$

В матричном виде СЛАУ вида (1.1) записывается как

$$\mathbf{Ax} = \mathbf{b}, \qquad (1.2)$$

где

$$\mathbf{A} = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nn} \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

Предположим, что для данной системы существует только единственное решение, то есть $\det(\mathbf{A}) \neq 0$. Метод состоит в последовательном обнулении элементов матрицы **A**, расположенных ниже главной диагонали, сначала в первом столбце, затем во втором столбце и т.д., для того чтобы привести её к треугольному виду. Для этого используются линейные комбинации уравнений исходной системы. Поясним суть этих преобразований.

Пусть $a_{11} \neq 0$. Умножим первое уравнение системы (1.1) на величину $m_{i1} = a_{i1} / a_{11}$ и вычтем его из i-го уравнения для $i = 2, 3, \ldots, n$. В результате система (1.1) преобразуется к виду:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \cdots \\ a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n = b_n^{(1)} \end{cases}, \qquad (1.3)$$

где

$$a_{ij}^{(1)} = a_{ij} - m_{i1}a_{1j};\ b_i^{(1)} = b_i - m_{i1}b_1;\ i, j = 2,3,\ldots,n.$$

Пусть $a_{22}^{(1)} \neq 0$. Умножим второе уравнение системы (*1.3*) на величину $m_{i2} = a_{i2}^{(1)} / a_{22}^{(1)}$ и, вычитая его из i-го уравнения для $i = 3, 4, \ldots, n$, обнулим

коэффициенты второго столбца системы (*1.3*), стоящие ниже элемента $a_{22}^{(1)}$. При этом преобразуемые коэффициенты и правая часть вычисляются по формулам:

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{2j}^{(1)};\ b_i^{(2)} = b_i^{(1)} - m_{i2}b_2^{(1)};\ i, j = 3,4,\ldots,n.$$

Продолжая этот процесс, приходим к системе с треугольной матрицей:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \cdots \\ a_{nn}^{(n-1)}x_n = b_n^{(n-1)} \end{cases}, \qquad (1.4)$$

где

$$a_{ij}^{(i-1)} = a_{ij}^{(i-2)} - m_{i,i-1}a_{i-1,j}^{(i-2)},\ j = i, i+1,\ldots,n,$$

$$b_i^{(i-1)} = b_i^{(i-2)} - m_{i,i-1}b_{i-1,j}^{(i-2)},\ i = 3,4,\ldots,n.$$

Стоящие на диагонали коэффициенты этой системы a_{11}, $a_{22}^{(1)}$, … , $a_{nn}^{(n-1)}$, которые по предположению все отличны от нуля, принято называть **ведущими элементами** метода исключения Гаусса. Их произведение дает значение определителя матрицы системы (1.1), которое, в силу использованных преобразований и свойств определителей, совпадает со значением определителя матрицы системы (1.4). Поэтому

$$\det(\mathbf{A}) = a_{11}a_{22}^{(1)}\cdots a_{nn}^{(n-1)}.$$

Решение системы (1.4) может быть получено по рекуррентным формулам:

$$x_n = b_n^{(n-1)} / a_{nn}^{(n-1)},$$

$$x_i = \frac{1}{a_{ii}^{(i-1)}}\left[b_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)}x_j \right],\ i = n-1, n-2,\ldots,1. \qquad (1.5)$$

Приведение системы (1.1) к треугольному виду (1.4) принято называть **прямым ходом** метода исключения Гаусса. Решение системы по рекуррентным формулам (1.5) называют **обратным ходом** этого метода. Далее приведен соответствующий алгоритм.

Алгоритм – Прямой и обратный ходы метода Гаусса

Для $k = 1, 2, \ldots, N - 1$

  Для $i = k + 1, \ldots, N$

    $t_{ik} = a_{ik} / a_{kk}$

    $b_i = b_i - t_{ik}\, b_k$

  Увеличить i

  Для $j = k + 1, \ldots, N$

    $a_{ij} = a_{ij} - t_{ik}\, a_{kj}$

  Увеличить j

Увеличить k

$x_N = b_N / a_{NN}$

Для $k = N - 1, \ldots, 2, 1$

$$x_k = \left(b_k - \sum_{j=k+1}^{N} a_{kj}\, x_j \right) / a_{kk}$$

Увеличить k

Число обусловленности квадратной матрицы **A** (cond(**A**)) показывает, насколько она близка к вырожденной матрице. Если $\mathrm{cond}(\mathbf{A}) \geq 10^3$, то говорят, что матрица **A** плохо обусловлена. Если $1 \leq \mathrm{cond}(\mathbf{A}) \leq 100$, то матрица считается хорошо обусловленной.

Рассмотрим систему вида (1.2). Если матрица **A** вырожденная, то для некоторых **b** решения не существует, а для других **b** оно может быть не единственным. Поэтому, если **A** почти вырожденная, то малые изменения в ней и **b** вызовут большие изменения в **x**. Если же взять в качестве **A** единичную матрицу, то решение системы будет **x** = **b**. Значит, если **A** близка к единичной матрице, то малые изменения в **A** и **b** должны влечь за собой малые изменения в **x**.

Рассмотрим пример:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 2 & 4.01 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} 3 \\ 6 \end{pmatrix}.$$

Решением данной СЛАУ $\mathbf{x} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$. Если изменить вектор свободных членов на $\mathbf{b}_1 = \begin{pmatrix} 3 \\ 6.01 \end{pmatrix}$. Тогда решение системы $\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Видно, что даже малое изменение вектора свободных членов **b** приводит к изменению решения СЛАУ.

Заметим, что при последовательном исключении в методе Гаусса вычисления возможны, если ведущие элементы системы не равны нулю. Добиться выполнения этого условия можно, переставляя элементы строк и столбцов матрицы. Но среди ведущих элементов могут оказаться малые по абсолютной величине значения. При делении на такие ведущие элементы формируется погрешность округления (вычислительная погрешность). Чтобы избежать сильного влияния вычислительной погрешности на решение, применяется метод Гаусса с выбором ведущего элемента. Рассмотрим это на следующем примере.

Дано:

$$\mathbf{A} = \begin{pmatrix} 3 & 4 & -9 & 5 \\ -15 & -12 & 50 & -16 \\ -27 & -36 & 73 & 8 \\ 9 & 12 & -10 & -16 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} -14 \\ 44 \\ 142 \\ -76 \end{pmatrix}.$$

Прямой ход, 1 шаг. Максимальный по модулю элемент первого столбца $a_{33} = -27$. Переставив первое и третье уравнения местами, получим:

$$\mathbf{A} = \begin{pmatrix} -27 & -36 & 73 & 8 \\ -15 & -12 & 50 & -16 \\ 3 & 4 & -9 & 5 \\ 9 & 12 & -10 & -16 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} 142 \\ 44 \\ -14 \\ -76 \end{pmatrix}.$$

Вычислим масштабирующие множители для первого шага:

$$\mu_{21} = \frac{-15}{-27} = \frac{5}{9}; \mu_{31} = \frac{3}{-27} = -\frac{1}{9}; \mu_{41} = \frac{9}{-27} = -\frac{1}{3},$$

и выполним преобразование матрицы **A** и вектора свободных членов **b**:

$$\mathbf{A}=\begin{pmatrix} -27 & -36 & 73 & 8 \\ 0 & 8 & \frac{85}{9} & -\frac{184}{9} \\ 0 & 0 & -\frac{8}{9} & \frac{53}{9} \\ 0 & 0 & \frac{43}{3} & -\frac{40}{3} \end{pmatrix},\ \mathbf{b}=\begin{pmatrix} 142 \\ -\frac{314}{9} \\ \frac{16}{9} \\ -\frac{86}{3} \end{pmatrix}.$$

Второй шаг прямого хода не изменяет **A** и **b**.

Прямой ход, 3 шаг. Максимальный по модулю элемент третьего столбца: $a_{43} = 43/3$. Переставим третье и четвертое уравнение местами:

$$\mathbf{A}=\begin{pmatrix} -27 & -36 & 73 & 8 \\ 0 & 8 & \frac{85}{9} & -\frac{184}{9} \\ 0 & 0 & \frac{43}{3} & -\frac{40}{3} \\ 0 & 0 & -\frac{8}{9} & \frac{53}{9} \end{pmatrix},\ \mathbf{b}=\begin{pmatrix} 142 \\ -\frac{314}{9} \\ -\frac{86}{3} \\ \frac{16}{9} \end{pmatrix},$$

и вычислим масштабирующие множители для третьего шага:

$$\mu_{43}=-\frac{8\cdot 3}{9\cdot 43}=-\frac{8}{129}.$$

Далее выполним преобразование матрицы **A** и вектора свободных членов **b**:

$$\mathbf{A}=\begin{pmatrix} -27 & -36 & 73 & 8 \\ 0 & 8 & \frac{85}{9} & -\frac{184}{9} \\ 0 & 0 & \frac{43}{3} & -\frac{40}{3} \\ 0 & 0 & 0 & \frac{653}{129} \end{pmatrix},\ \mathbf{b}=\begin{pmatrix} 142 \\ -\frac{314}{9} \\ -\frac{86}{3} \\ 0 \end{pmatrix}.$$

Обратный ход. Из последнего уравнения находим, что $x_4 = 0$. Из третьего уравнения системы следует, что $x_3 = –2$. Согласно второго уравнения $x_2 = –2$. И из первого уравнения получим $x_1 = –8$.

1.2.2 Порядок выполнения работы

1. В соответствии с выданным преподавателем вариантом задания решить СЛАУ методом Гаусса.

Вариант 1

$$\begin{cases} x_1 + x_2 + x_3 - x_4 = 2{,}0005 \\ x_1 - x_2 - x_3 + x_4 = 0 \\ 2x_1 + x_2 - x_3 + 2x_4 = 9{,}12301 \\ 3x_1 + x_2 + 2x_3 - x_4 = 7 \end{cases}$$

Вариант 2

$$\begin{cases} -12{,}73x_1 + 37{,}45x_2 - 26{,}05x_3 - 0{,}52x_4 = 0{,}09 \\ -65{,}26x_1 - 104{,}13x_2 + 11{,}69x_3 + 32{,}57x_4 = 0{,}16 \\ -5{,}18x_1 + 67{,}52x_2 - 53{,}30x_3 - 19{,}30x_4 = 0{,}356231 \\ -31{,}82x_1 + 35{,}24x_2 + 15{,}50x_3 + 23{,}36x_4 = 2{,}23 \end{cases}$$

Вариант 3

$$\begin{cases} 0{,}85x_1 + 0{,}50x_2 + 1{,}41x_3 + 0{,}43x_4 = 1{,}05 \\ 1{,}15x_1 + 0{,}88x_2 + 0{,}66x_3 + 1{,}07x_4 = 0{,}10 \\ 1{,}43x_1 + 1{,}50x_2 + 0{,}46x_3 + 1{,}05x_4 = 0{,}00003 \\ 0{,}13x_1 + 1{,}20x_2 + 1{,}45x_3 + 1{,}47x_4 = 1{,}38 \end{cases}$$

Вариант 4

$$\begin{cases} -3{,}22x_1 + 11{,}08x_2 - 16{,}29x_3 - 2{,}54x_4 = 1{,}08 \\ -22{,}63x_1 + 50{,}14x_2 + 0{,}15x_3 + 13{,}79x_4 = 2{,}27 \\ -10{,}41x_1 + 59{,}91x_2 - 30{,}05x_3 - 9{,}09x_4 = 0{,}30007 \\ -36{,}64x_1 + 74{,}51x_2 - 18{,}71x_3 + 12{,}21x_4 = 0{,}24 \end{cases}$$

Вариант 5

$$\begin{cases} 1{,}91x_1 + 1{,}83x_2 + 0{,}70x_3 + 0{,}10x_4 = 2{,}14 \\ 1{,}68x_1 + 1{,}15x_2 + 1{,}46x_3 + 1{,}88x_4 = 0{,}861208 \\ 0{,}23x_1 + 0{,}30x_2 + 0{,}26x_3 + 2{,}17x_4 = 2{,}47 \\ 2{,}48x_1 + 2{,}15x_2 + 1{,}35x_3 + 0{,}35x_4 = 2{,}35 \end{cases}$$

Вариант 6

$$\begin{cases} -1{,}38x_1 + 5{,}21x_2 - 7{,}29x_3 + 7{,}77x_4 = 0{,}43 \\ -0{,}30x_1 + 1{,}73x_2 - 3{,}41x_3 + 3{,}65x_4 = 0{,}2540025 \\ -1{,}26x_1 + 3{,}64x_2 - 2{,}08x_3 + 6{,}36x_4 = 1{,}19 \\ -1{,}81x_1 + 4{,}19x_2 - 4{,}58x_3 + 3{,}75x_4 = 0{,}01 \end{cases}$$

Вариант 7

$$\begin{cases} -2{,}11x_1 + 5{,}56x_2 - 4{,}53x_3 + 0{,}32x_4 = 0{,}42 \\ 0{,}57x_1 + 1{,}62x_2 - 3{,}11x_3 - 0{,}77x_4 = 0{,}10047 \\ -9{,}36x_1 + 8{,}61x_2 + 5{,}10x_3 + 7{,}63x_4 = 0{,}47 \\ -5{,}23x_1 + 11{,}75x_2 - 1{,}16x_3 + 5{,}76x_4 = 0{,}40 \end{cases}$$

Вариант 8

$$\begin{cases} 1{,}83x_1 + 4{,}34x_2 - 7{,}49x_3 + 11{,}07x_4 = 1{,}49 \\ -2{,}15x_1 + 4{,}94x_2 - 3{,}89x_3 + 6{,}48x_4 = 2{,}86 \\ -0{,}50x_1 + 1{,}94x_2 - 3{,}32x_3 + 4{,}33x_4 = 0{,}2742 \\ -4{,}27x_1 + 8{,}45x_2 - 7{,}71x_3 + 12{,}30x_4 = 1{,}94 \end{cases}$$

2. Реализовать в Octave вычисления согласно прямого и обратного ходов метода Гаусса и сравнить получено решение, с результатами, полученными в п. 1.

3. Вычислить число обусловленности матрицы **A** с помощью встроенных средств Octave. Произвести малое изменение одного из элементов вектора свободных членов и повторно найти решение СЛАУ. Оценить расхождение полученных результатов.

4. Реализовать в Octave метод Гаусса с выбором ведущего элемента по столбцам. Оценить различия в полученных результатах.

1.2.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "Работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

1.3 LU-разложение

Цель работы – изучение и программная реализация LU-разложения матриц для решения СЛАУ.

Для достижения поставленной цели используется программное обеспечение GNU Octave. В ходе работы необходимо программно реализовать алгоритм LU-разложения квадратной матрицы и на основе разложения найти решение СЛАУ.

1.3.1 Краткая теоретическая справка

LU-разложение – представление матрицы **A** в виде произведения нижнетреугольной матрицы с единичной диагональю **L** и верхнетреугольной матрицы **U**.

Будем использовать следующие обозначения для элементов матриц:

$$\mathbf{L} = (l_{ij}),\ \mathbf{U} = (u_{ij}),\ i, j = 1..n, \tag{1.6}$$

причем диагональные элементы матрицы **L**:

$$l_{ii} = 1,\ i = 1..n.$$

Расчет элементов матриц (1.6) может быть выполнен на основе приведенного ниже алгоритма.

Алгоритм – Вычисление элементов матриц **L** и **U**

1. $u_{1j} = a_{1j},\ j = 1..n$

2. $l_{j1} = \dfrac{a_{j1}}{u_{11}},\ j = 2..n\ \ (u_{11} \neq 0)$

3. $u_{ij} = a_{ij} - \sum_{k=1}^{i} l_{jk} \cdot u_{kj},\ j = i..n\ \ i = 2..n$

4. $l_{ji} = \dfrac{1}{u_{ii}}\left(a_{ji} - \sum_{k=1}^{i} l_{jk} \cdot u_{ki}\right),\ j = i+1..n\ \ i = 2..n$

Таким образом, решение СЛАУ сводится к виду:

$$\begin{cases} \mathbf{Ux} = \mathbf{y} \\ \mathbf{Ly} = \mathbf{b} \end{cases}. \tag{1.7}$$

При вычислениях сначала решается второе уравнение данной системы, а затем первое.

1.3.2 Порядок выполнения работы

1. В соответствии с выданным преподавателем вариантом задания реализовать LU-разложение для матрицы **A**.

Вариант 1

$$\begin{cases} x_1 + x_2 + x_3 - x_4 = 2{,}0005 \\ x_1 - x_2 - x_3 + x_4 = 0 \\ 2x_1 + x_2 - x_3 + 2x_4 = 9{,}12301 \\ 3x_1 + x_2 + 2x_3 - x_4 = 7 \end{cases}$$

Вариант 2

$$\begin{cases} -12{,}73x_1 + 37{,}45x_2 - 26{,}05x_3 - 0{,}52x_4 = 0{,}09 \\ -65{,}26x_1 - 104{,}13x_2 + 11{,}69x_3 + 32{,}57x_4 = 0{,}16 \\ -5{,}18x_1 + 67{,}52x_2 - 53{,}30x_3 - 19{,}30x_4 = 0{,}356231 \\ -31{,}82x_1 + 35{,}24x_2 + 15{,}50x_3 + 23{,}36x_4 = 2{,}23 \end{cases}$$

Вариант 3

$$\begin{cases} 0{,}85x_1 + 0{,}50x_2 + 1{,}41x_3 + 0{,}43x_4 = 1{,}05 \\ 1{,}15x_1 + 0{,}88x_2 + 0{,}66x_3 + 1{,}07x_4 = 0{,}10 \\ 1{,}43x_1 + 1{,}50x_2 + 0{,}46x_3 + 1{,}05x_4 = 0{,}00003 \\ 0{,}13x_1 + 1{,}20x_2 + 1{,}45x_3 + 1{,}47x_4 = 1{,}38 \end{cases}$$

Вариант 4

$$\begin{cases} -3{,}22x_1 + 11{,}08x_2 - 16{,}29x_3 - 2{,}54x_4 = 1{,}08 \\ -22{,}63x_1 + 50{,}14x_2 + 0{,}15x_3 + 13{,}79x_4 = 2{,}27 \\ -10{,}41x_1 + 59{,}91x_2 - 30{,}05x_3 - 9{,}09x_4 = 0{,}30007 \\ -36{,}64x_1 + 74{,}51x_2 - 18{,}71x_3 + 12{,}21x_4 = 0{,}24 \end{cases}$$

Вариант 5

$$\begin{cases} 1{,}91x_1 + 1{,}83x_2 + 0{,}70x_3 + 0{,}10x_4 = 2{,}14 \\ 1{,}68x_1 + 1{,}15x_2 + 1{,}46x_3 + 1{,}88x_4 = 0{,}861208 \\ 0{,}23x_1 + 0{,}30x_2 + 0{,}26x_3 + 2{,}17x_4 = 2{,}47 \\ 2{,}48x_1 + 2{,}15x_2 + 1{,}35x_3 + 0{,}35x_4 = 2{,}35 \end{cases}$$

Вариант 6

$$\begin{cases} -1{,}38x_1 + 5{,}21x_2 - 7{,}29x_3 + 7{,}77x_4 = 0{,}43 \\ -0{,}30x_1 + 1{,}73x_2 - 3{,}41x_3 + 3{,}65x_4 = 0{,}2540025 \\ -1{,}26x_1 + 3{,}64x_2 - 2{,}08x_3 + 6{,}36x_4 = 1{,}19 \\ -1{,}81x_1 + 4{,}19x_2 - 4{,}58x_3 + 3{,}75x_4 = 0{,}01 \end{cases}$$

Вариант 7

$$\begin{cases} -2{,}11x_1 + 5{,}56x_2 - 4{,}53x_3 + 0{,}32x_4 = 0{,}42 \\ 0{,}57x_1 + 1{,}62x_2 - 3{,}11x_3 - 0{,}77x_4 = 0{,}10047 \\ -9{,}36x_1 + 8{,}61x_2 + 5{,}10x_3 + 7{,}63x_4 = 0{,}47 \\ -5{,}23x_1 + 11{,}75x_2 - 1{,}16x_3 + 5{,}76x_4 = 0{,}40 \end{cases}$$

Вариант 8

$$\begin{cases} 1{,}83x_1 + 4{,}34x_2 - 7{,}49x_3 + 11{,}07x_4 = 1{,}49 \\ -2{,}15x_1 + 4{,}94x_2 - 3{,}89x_3 + 6{,}48x_4 = 2{,}86 \\ -0{,}50x_1 + 1{,}94x_2 - 3{,}32x_3 + 4{,}33x_4 = 0{,}2742 \\ -4{,}27x_1 + 8{,}45x_2 - 7{,}71x_3 + 12{,}30x_4 = 1{,}94 \end{cases}$$

2. Реализовать в Octave алгоритм нахождения матриц **L** и **U** и последующее решение СЛАУ согласно (*1.7*). Для проверки корректности реализованного алгоритма применить результат предыдущего пункта.

3. Сравнить результаты решения СЛАУ, полученные методом Гаусса (из предыдущей работы) и методом LU-разложения.

4. Реализовать обращение матрицы с помощью полученного LU-разложения.

1.3.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

1.4 QR-разложение

Цель работы – изучение и программная реализация QR-разложения матриц.

Для достижения поставленной цели используется программное обеспечение GNU Octave. В ходе работы необходимо программно реализовать алгоритм QR-разложения матрицы и сравнить разложение с результатами, полученными с помощью встроенных функций Octave.

1.4.1 Краткая теоретическая справка

QR-разложение является основой одного из методов решения СЛАУ и поиска собственных значений (чисел) и векторов матрицы. Пусть **A** – квадратная невырожденная матрица, тогда существует единственное QR-разложение, если наложить дополнительное условие, что элементы на диагонали матрицы **R** являются положительными вещественными числами.

Пусть **A** – матрица размерности 3×3, т.е.

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix}, \tag{1.8}$$

где $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$ – вектор-столбцы, причем векторы $\mathbf{a}_1$, $\mathbf{a}_2$ и $\mathbf{a}_3$ линейно независимы. С помощью процесса ортогонализации построим ортонормированный базис $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$ в пространстве столбцов матрицы **A**. Тогда матрица **Q** будет ортогональной. Процесс ортогонализации приводит к следующим соотношениям:

$$\mathbf{b}_1 = \mathbf{a}_1, \tag{1.9}$$

$$\mathbf{b}_2 = \mathbf{a}_2 + c \cdot \mathbf{b}_1, \tag{1.10}$$

$$\mathbf{b}_3 = \mathbf{a}_3 + d_1 \cdot \mathbf{b}_1 + d_2 \cdot \mathbf{b}_2, \tag{1.11}$$

где числа c, d_1 и d_2 выбираются из условия ортогональности векторов $\mathbf{b}_1$, $\mathbf{b}_2$ и $\mathbf{b}_3$:

$$c = \frac{-\mathbf{b}_1^{T'} \cdot \mathbf{a}_2}{\mathbf{b}_1^T \cdot \mathbf{b}_1}, \tag{1.12}$$

$$d_1 = \frac{-\left(\mathbf{b}_1^T \cdot \mathbf{a}_3\right)}{\mathbf{b}_1^T \cdot \mathbf{b}_1}, \tag{1.13}$$

$$d_2 = \frac{-\left(\mathbf{b}_2^T \cdot \mathbf{a}_3\right)}{\mathbf{b}_2^T \cdot \mathbf{b}_2}. \tag{1.14}$$

Пронормировав векторы $\mathbf{b}_1$, $\mathbf{b}_2$ и $\mathbf{b}_3$, получим:

$$\mathbf{q}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|},\ \mathbf{q}_2 = \frac{\mathbf{b}_2}{\|\mathbf{b}_2\|},\ \mathbf{q}_3 = \frac{\mathbf{b}_3}{\|\mathbf{b}_3\|}. \tag{1.15}$$

Запишем процесс ортогонализации в матричном виде. Получим, что $\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}$, где

$$\mathbf{R} = \begin{pmatrix} \|\mathbf{b}_1\| & \dfrac{\mathbf{a}_2^T \cdot \mathbf{b}_1}{\|\mathbf{b}_1\|} & \dfrac{\mathbf{a}_3^T \cdot \mathbf{b}_1}{\|\mathbf{b}_1\|} \\ 0 & \|\mathbf{b}_2\| & \dfrac{\mathbf{a}_3^T \cdot \mathbf{b}_2}{\|\mathbf{b}_2\|} \\ 0 & 0 & \|\mathbf{b}_3\| \end{pmatrix}. \tag{1.16}$$

1.4.2 Порядок выполнения работы

1. Сформировать матрицу **A** размерностью 3×3 таким образом, чтобы её столбцы были линейно независимыми.

2. Выполнить разложение матрицы **A** согласно (1.8)–(1.16) для получения QR-разложения. Выполнить проверку полученных матриц **Q** и **R**.

3. Найти QR-разложение матрицы **A** с помощью встроенных средств Octave. Сравнить полученное разложение с результатом предыдущего пункта. Сделать выводы.

4. Создать копию матрицы **A** и удалить в ней последний столбец. Пусть **B** – полученная в результате матрица. Найти QR-разложение для

матрицы **B** с помощью встроенной функции Octave. Объяснить принцип QR-разложения прямоугольных матриц.

1.4.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

1.5 Метод Якоби

Цель работы – изучение и программная реализация итерационного метода Якоби для решения СЛАУ.

Для достижения поставленной цели используется программное обеспечение Octave. В ходе работы необходимо программно реализовать итерационный метод и оценить его сходимости для заданной точности решения.

1.5.1 Краткая теоретическая справка

Метод Гаусса, как и методы, основанные на LU- и QR-разложении, относятся к прямым методам решения СЛАУ, т.е. если отбросить ошибку округления, решение, полученное посредством данных методов, является точным. Другую группу составляют итерационные методы, при помощи которых решение находится путем последовательных приближений. Если итерационный процесс сходится, то каждое последующее приближение уточняет предыдущее. Представим матрицу системы (1.1) в виде:

$$\mathbf{A} = \mathbf{L} + \mathbf{P} + \mathbf{D}, \tag{1.17}$$

где

$$\mathbf{D}=\begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 & 0 \\ 0 & a_{22} & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix}, \mathbf{L}=\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ a_{21} & 0 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{n,n-1} & a_{nn} \end{pmatrix},$$

$$\mathbf{P}=\begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1,n-1} & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2,n-1} & a_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Предполагаем, что $a_{ii} \neq 0$, $i = 1, 2, \ldots, n$. Тогда систему (1.1) с учетом (1.17) можно записать в виде:

$$\mathbf{Lx}+\mathbf{Dx}+\mathbf{Px}=\mathbf{b},$$

или

$$\mathbf{x}=-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{P})\mathbf{x}+\mathbf{D}^{-1}\mathbf{b}. \qquad (1.18)$$

Сравнивая (1.1) и (1.18), видно, что

$$\mathbf{B}=-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{P}),\ \mathbf{C}=\mathbf{D}^{-1}.$$

На основе формулы (1.18) записывается итерационный процесс метода Якоби:

$$\mathbf{x}_{k+1}=-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{P})\mathbf{x}_k+\mathbf{D}^{-1}\mathbf{b}. \qquad (1.19)$$

Пусть $\mathbf{x}_k=\left(x_1^{(k)},x_2^{(k)},\ldots,x_n^{(k)}\right)$. Тогда матричная формула (1.19) с учетом вида матриц **L**, **P** и $\mathbf{D}^{-1}$ записывается покомпонентно в виде:

$$\begin{cases} x_1^{(k+1)}=\dfrac{b_1}{a_{11}}-\dfrac{a_{12}}{a_{11}}x_2^{(k)}-\dfrac{a_{13}}{a_{11}}x_3^{(k)}-\ldots-\dfrac{a_{1n}}{a_{11}}x_n^{(k)} \\ x_2^{(k+1)}=\dfrac{b_2}{a_{22}}-\dfrac{a_{21}}{a_{22}}x_2^{(k)}-\dfrac{a_{23}}{a_{22}}x_3^{(k)}-\ldots-\dfrac{a_{2n}}{a_{22}}x_n^{(k)} \\ \cdots \\ x_n^{(k+1)}=\dfrac{b_n}{a_{nn}}-\dfrac{a_{n1}}{a_{nn}}x_2^{(k)}-\dfrac{a_{n3}}{a_{nn}}x_3^{(k)}-\ldots-\dfrac{a_{n,n-1}}{a_{nn}}x_n^{(k)} \end{cases},$$

или

$$x_i^{(k+1)}=\frac{b_i}{a_{ii}}-\sum_{j=1}^{i-1}\frac{a_{ij}}{a_{ii}}x_j^{(k)}-\sum_{j=i+1}^{n}\frac{a_{ij}}{a_{ii}}x_j^{(k)},\ i=1,2,\ldots,n. \qquad (1.20)$$

Тогда матрица **B** имеет вид:

$$\mathbf{B}=\begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \dots & -\frac{a_{1,n-1}}{a_{11}} & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \dots & -\frac{a_{2,n-1}}{a_{22}} & -\frac{a_{2n}}{a_{22}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \dots & -\frac{a_{n,n-1}}{a_{nn}} & 0 \end{pmatrix}.$$

Для сходимости метода Якоби достаточно, чтобы $\|\mathbf{B}\| < 1$. Достаточном условием сходимости метода Якоби является:

$$\sum_{\substack{j=1 \\ i\neq j}}^{n} |a_{ij}| < |a_{ii}|,\ i=1,2,\dots,n. \qquad (1.21)$$

Неравенство (1.21) означает диагональное преобладание в исходной матрице **A**, которого следует добиться до вычислений согласно формуле (1.20).

Рассмотрим пример. Пусть дана система уравнений:

$$\begin{cases} 100x_1 + 30x_2 - 70x_3 = 60 \\ 15x_1 - 50x_2 - 5x_3 = -40 \\ 6x_1 + 2x_2 + 20x_3 = 28 \end{cases},$$

где

$$\mathbf{A}=\begin{pmatrix} 100 & 30 & -70 \\ 15 & -50 & -5 \\ 6 & 2 & 20 \end{pmatrix}, \mathbf{b}=\begin{pmatrix} 60 \\ -40 \\ 28 \end{pmatrix}.$$

В исходной матрице **A** диагонального преобладания нет. Поэтому до основных вычислений выполним преобразования. Сложим первые два уравнения:

$$\begin{cases} 115x_1 - 20x_2 - 75x_3 = 20 \\ 15x_1 - 50x_2 - 5x_3 = -40 \\ 6x_1 + 2x_2 + 20x_3 = 28 \end{cases}.$$

В результате матрица СЛАУ преобразуется к матрице с диагональным преобладанием. Тогда итерационный процесс можно записать следующим образом:

$$\begin{cases} x_1^{(k+1)} = \dfrac{20}{115} + \dfrac{20}{115} x_2^{(k)} + \dfrac{75}{115} x_3^{k} \\ x_2^{(k+1)} = \dfrac{40}{50} + \dfrac{15}{50} x_1^{(k)} - \dfrac{5}{50} x_3^{(k)} \\ x_3^{(k+1)} = \dfrac{28}{20} - \dfrac{6}{20} x_1^{(k)} - \dfrac{2}{20} x_2^{(k)} \end{cases}.$$

При этом матрица **B** примет вид:

$$\mathbf{B} = \begin{pmatrix} 0 & 0{,}17391 & 0{,}65217 \\ 0{,}3 & 0 & -0{,}1 \\ -0{,}3 & -0{,}1 & 0 \end{pmatrix}.$$

Нормы матрицы $\|\mathbf{B}\|_1 = 0{,}826 < 1$, $\|\mathbf{B}\|_2 = 0{,}75217 < 1$, что говорит о выполнении условий сходимости метода Якоби. В качестве начального приближения возьмем:

$$\mathbf{x}_0 = \mathbf{C} \cdot \mathbf{b} = \begin{pmatrix} 0{,}17391 \\ 0{,}8 \\ 1{,}4 \end{pmatrix}.$$

Результат вычисления 6 итераций приведен ниже:

$$\begin{cases} x_1^{(1)} = 1{,}2260 \\ x_2^{(1)} = 0{,}7122 \\ x_3^{(1)} = 1{,}2678 \end{cases}, \begin{cases} x_1^{(2)} = 1{,}2246 \\ x_2^{(2)} = 1{,}041 \\ x_3^{(2)} = 0{,}961 \end{cases}, \begin{cases} x_1^{(3)} = 0{,}9817 \\ x_2^{(3)} = 1{,}0413 \\ x_3^{(3)} = 1{,}2829 \end{cases},$$

$$\begin{cases} x_1^{(4)} = 1{,}1917 \\ x_2^{(4)} = 0{,}9662 \\ x_3^{(4)} = 1{,}0014 \end{cases}, \begin{cases} x_1^{(5)} = 0{,}9950 \\ x_2^{(5)} = 1{,}0574 \\ x_3^{(5)} = 0{,}9459 \end{cases}, \begin{cases} x_1^{(6)} = 0{,}9747 \\ x_2^{(6)} = 1{,}0039 \\ x_3^{(6)} = 0{,}9958 \end{cases}.$$

Из полученных данных хорошо видна сходимость от итерации к итерации.

1.5.2 Порядок выполнения работы

1. В соответствии с выданным преподавателем вариантом выполнить преобразования над матрицей **A** для удовлетворения условия сходимости метода Якоби.

Вариант 1
$$\begin{cases} 10x_1 + x_2 - x_3 = 11 \\ x_1 + 10x_2 - x_3 = 10 \\ -x_1 + x_2 + 10x_3 = 10 \end{cases}$$

Вариант 2
$$\begin{cases} 20{,}9x_1 + 1{,}2x_2 + 2{,}1x_3 = 31 \\ 1{,}2x_1 + 21{,}2x_2 + 1{,}5x_3 = 10 \\ 2{,}1x_1 + 1{,}5x_2 + 19{,}8x_3 = 14 \end{cases}$$

Вариант 3
$$\begin{cases} 14x_1 + 3{,}5x_2 + 7{,}1x_3 = 7.14 \\ 5{,}1x_1 + 17x_2 + 1{,}5x_3 = 2 \\ 8{,}1x_1 + 0{,}1x_2 + 11x_3 = 3 \end{cases}$$

Вариант 4
$$\begin{cases} 3x_1 + 0{,}89x_2 + 0{,}64x_3 = 2 \\ 1{,}11x_1 + 4x_2 + 1{,}2x_3 = 2.2 \\ 0{,}14x_1 + 0{,}15x_2 + 9x_3 = 22 \end{cases}$$

Вариант 5
$$\begin{cases} 5{,}2x_1 + 1{,}23x_2 + 2{,}11x_3 = 15 \\ 0{,}01x_1 + 7{,}41x_2 + 1{,}95x_3 = 3 \\ 0{,}02x_1 + 0{,}03x_2 + 4{,}45x_3 = 0{,}64 \end{cases}$$

Вариант 6
$$\begin{cases} 75{,}2x_1 + 8{,}23x_2 + 2{,}54x_3 = 0{,}14 \\ 2{,}51x_1 + 24{,}8x_2 + 5{,}41x_3 = 8 \\ 6{,}36x_1 + 2{,}25x_2 + 64{,}31x_3 = 3{,}15 \end{cases}$$

Вариант 7
$$\begin{cases} 14x_1 + 2x_2 + 3x_3 = 5 \\ x_1 + 7x_2 + 4x_3 = 0{,}12 \\ 4x_1 + 3x_2 + 74x_3 = 14 \end{cases}$$

Вариант 8
$$\begin{cases} 3x_1 + x_2 + x_3 = 15 \\ 2x_1 + 4x_2 + 0{,}54x_3 = 0{,}012 \\ 1{,}12x_1 + 0{,}078x_2 + 9x_3 = 0{,}14 \end{cases}$$

2. Записать итерационный процесс и решить СЛАУ методом Якоби до шестой итерации включительно.

3. Реализовать в Octave метод Якоби для решения СЛАУ с точностью $\varepsilon = 0{,}001$. Перед нахождением решения в программе выполнить проверку на соответствие исходных данных требованиям метода. Сравнить решения, полученные вручную до шестой итерации, с результатом решения реализованной программы в Octave. При нахождении решения для *i*-ой итерации построить вектор $\mathbf{V}_i = [x_1, x_2, \ldots, x_k]$, где x_k – получаемое *k*-е решение системы уравнений. По достижении заданной точности построить график

зависимости $\mathbf{V}_i$ от номера итерации i. Оценить по построенному графику скорость сходимости метода.

1.5.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

1.6 Метод релаксации

Цель работы – изучение и программная реализация итерационного метода релаксации для решения СЛАУ.

Для достижения поставленной цели используется программное обеспечение GNU Octave. В ходе работы необходимо программно реализовать метод релаксации и оценить скорость сходимости итерационного процесса для нахождения решения СЛАУ с заданной точностью.

1.6.1 Краткая теоретическая справка

Представив матрицу $\mathbf{A}$ в виде (1.17) запишем итерационный процесс в виде:

$$\frac{(\mathbf{D}+\omega\cdot\mathbf{L})(\mathbf{x}_{s+1}-\mathbf{x}_s)}{\omega}+\mathbf{A}\mathbf{x}_s=\mathbf{b}, \qquad (1.22)$$

где $\mathbf{x}_s$ – приближение, полученное на итерации с номером s, $\mathbf{x}_{s+1}$ – следующее приближение, $\omega > 0$ – числовой параметр, задаваемый пользователем и определяющий скорость сходимости метода.

Необходимым условием сходимости метода релаксации при произвольном начальном приближении $\mathbf{x}_0$ к точному решению $\mathbf{x}$ является выполнение условия $\omega \in (0, 2)$. Если же матрица $\mathbf{A}$ симметрична и положительно определена, то выполнение данного условия является также достаточным. При этом если

$\omega \in (0, 1)$, то говорят о методе нижней релаксации, а при $\omega \in (1, 2)$ – о методе верхней релаксации. При $\omega = 1$ метод релаксации совпадает с методом Гаусса-Зейделя.

Из выражения (1.22) получим формулы для отыскания $\mathbf{x}^{S+1}$ по предыдущему приближению $\mathbf{x}^S$ в явном виде:

$$(\mathbf{D} + \omega \cdot \mathbf{L})(\mathbf{x}_{s+1} - \mathbf{x}_s) + \omega \mathbf{A}\mathbf{x}_s = \omega \mathbf{b} \Rightarrow \mathbf{D}\mathbf{x}_{s+1} + \omega \mathbf{L}\mathbf{x}_{s+1} - \mathbf{D}\mathbf{x}_s - \omega \mathbf{L}\mathbf{x}_s + \omega \mathbf{A}\mathbf{x}_s = \omega \mathbf{b} \Rightarrow$$

$$\mathbf{D}\mathbf{x}_{s+1} = -\omega \mathbf{L}\mathbf{x}_{s+1} + \mathbf{D}\mathbf{x}_s - \omega(\mathbf{A} - \mathbf{L})\mathbf{D}\mathbf{x}_s + \omega \mathbf{b}.$$

Тогда с учетом того, что $\mathbf{A} - \mathbf{L} = \mathbf{P} + \mathbf{D}$, получим:

$$\mathbf{D}\mathbf{x}_{s+1} = -\omega \mathbf{L}\mathbf{x}_{s+1} + (1 - \omega)\mathbf{D}\mathbf{x}_s - \omega \mathbf{P}\mathbf{x}_s + \omega \mathbf{b}. \qquad (1.23)$$

В начале задается начальное приближение $\mathbf{x}_0$, далее на каждом шаге итерационного процесса (1.23) необходимо обратить в ноль максимальную невязку:

$$\mathbf{N}_{s+1} = \mathbf{x}_{s+1} - \mathbf{x}_s.$$

Алгоритм завершается, когда

$$|\mathbf{N}_s| < \varepsilon,$$

где ε – задаваемая пользователем точность вычислений.

1.6.2 Порядок выполнения работы

1. В соответствии с выданным преподавателем вариантом выбрать СЛАУ и реализовать метод релаксации для нахождения её решения.

Вариант 1
$$\begin{cases} 10x_1 + x_2 - x_3 = 11 \\ x_1 + 10x_2 - x_3 = 10 \\ -x_1 + x_2 + 10x_3 = 10 \end{cases}$$

Вариант 2
$$\begin{cases} 20{,}9x_1 + 1{,}2x_2 + 2{,}1x_3 = 31 \\ 1{,}2x_1 + 21{,}2x_2 + 1{,}5x_3 = 10 \\ 2{,}1x_1 + 1{,}5x_2 + 19{,}8x_3 = 14 \end{cases}$$

Вариант 3
$$\begin{cases} 14x_1 + 3{,}5x_2 + 7{,}1x_3 = 7.14 \\ 5{,}1x_1 + 17x_2 + 1{,}5x_3 = 2 \\ 8{,}1x_1 + 0{,}1x_2 + 11x_3 = 3 \end{cases}$$

Вариант 4
$$\begin{cases} 3x_1 + 0{,}89x_2 + 0{,}64x_3 = 2 \\ 1{,}11x_1 + 4x_2 + 1{,}2x_3 = 2.2 \\ 0{,}14x_1 + 0{,}15x_2 + 9x_3 = 22 \end{cases}$$

Вариант 5
$$\begin{cases} 5{,}2x_1 + 1{,}23x_2 + 2{,}11x_3 = 15 \\ 0{,}01x_1 + 7{,}41x_2 + 1{,}95x_3 = 3 \\ 0{,}02x_1 + 0{,}03x_2 + 4{,}45x_3 = 0{,}64 \end{cases}$$

Вариант 6
$$\begin{cases} 75{,}2x_1 + 8{,}23x_2 + 2{,}54x_3 = 0{,}14 \\ 2{,}51x_1 + 24{,}8x_2 + 5{,}41x_3 = 8 \\ 6{,}36x_1 + 2{,}25x_2 + 64{,}31x_3 = 3{,}15 \end{cases}$$

Вариант 7
$$\begin{cases} 14x_1 + 2x_2 + 3x_3 = 5 \\ x_1 + 7x_2 + 4x_3 = 0{,}12 \\ 4x_1 + 3x_2 + 74x_3 = 14 \end{cases}$$

Вариант 8
$$\begin{cases} 3x_1 + x_2 + x_3 = 15 \\ 2x_1 + 4x_2 + 0{,}54x_3 = 0{,}012 \\ 1{,}12x_1 + 0{,}078x_2 + 9x_3 = 0{,}14 \end{cases}$$

2. Выполнить оценку сходимости метода для заданной СЛАУ при $\omega = 0{,}4$; 1,3; 1,95 и $\varepsilon = 0{,}0001$. Сравнить по точности и скорости сходимости полученные решения с решением, полученным методом Якоби. Сделать выводы.

3. Удалить последнюю строку, последний столбец из заданной матрицы **A** и последнею строку из вектора **b**. Повторно выполнить пункт 2 для измененной матрицы **A** при $\varepsilon_1 = 0{,}0001$ ($\omega = 0{,}4$; 1,3; 1,95) и $\varepsilon_2 = 0{,}00001$ ($\omega = 0{,}4$; 1,3; 1,95). Сравнить по скорости сходимости решение для матрицы **A** порядка 2 и 3.

1.6.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

1.7 Численное интегрирование

Цель работы – изучение и программная реализация методов численного

интегрирования.

Для достижения поставленной цели используется программное обеспечение GNU Octave. В ходе работы необходимо программно реализовать методы левых, правых, средних прямоугольников, метод Симпсона и оценить погрешность вычисления значений интеграла для заданных функций.

1.7.1 Краткая теоретическая справка

К численному интегрированию прибегают в случаях, когда невозможно аналитически получить первообразную, или, когда такая первообразная имеет слишком сложный для вычисления вид. Задача численного интегрирования заключается в вычислении интеграла вида (здесь и далее рассматриваются только определенные интегралы):

$$I = \int_a^b f(x)dx, \tag{1.24}$$

где a и b – нижний и верхний пределы интегрирования, $f(x)$ – непрерывная функция на отрезке $[a, b]$.

Сущность большинства методов вычисления определенных интегралов вида (1.24) состоит в замене подынтегральной функции $f(x)$ аппроксимирующей функцией $\phi(x)$, для которой можно легко записать первообразную в элементарных функциях, т.е.

$$I = \int_a^b f(x)dx \approx \int_a^b \phi(x)dx + R,$$

где R – погрешность вычисления.

Наиболее широко на практике для вычисления определенных интегралов используются квадратурные формулы – приближенные равенства вида:

$$\int_a^b f(x)dx = \sum_{i=0}^{n} \int_{x_{i-2}}^{x_i} f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i), \tag{1.25}$$

что соответствует разбиению полного интеграла на сумму интегралов от функции по элементарным отрезкам. Здесь x_i – некоторые точки (узлы) из отрезка $[a, b]$, разбитого на n элементарных отрезков $[x_{i-1}, x_i]$, при этом $x_0 = a$, а

$x_n = b$, а a_i – некоторые коэффициенты. Сумма, стоящая в правой части выражения, называется **квадратурной суммой**. Погрешность вычисления по этой формуле определяется выражением:

$$R = \int_a^b f(x)dx - \sum_{i=0}^{n} a_i f(x_i).$$

Заметим, что интеграл вида (1.25) определяет площадь фигуры, ограниченной прямыми $y = 0$, $x = a$, $x = b$ и графиком функции $y = f(x)$. Для вычисления интеграла можно разбить площадь под кривой на простые фигуры. На этом и основаны простейшие методы вычисления определенного интеграла.

Формулы прямоугольников. Рассмотрим разбиение отрезка [a, b] на n элементарных отрезков [x_{i-1}, x_i]. Пусть $h = x_i - x_{i-1}$, тогда каждому значению x_i можно поставить в соответствие значение функции $f(x_i)$. Рассматривая $f(x_i)$ как высоту прямоугольника с основанием h можно получить две формулы: формулу левых прямоугольников (рисунок *1.2а*):

$$I \approx h \sum_{i=1}^{n} f(x_{i-1}), \qquad (1.26)$$

и формулу правых прямоугольников:

$$I \approx h \sum_{i=1}^{n} f(x_i). \qquad (1.27)$$

Аналогично за высоту прямоугольника можно взять его середину. В этом случае отрезок обычно разбивают на $2n$ отрезков и рассматривают только нечетные узлы, что эквивалентно рассмотрению прямоугольников высотой, равной значению функции в середине элементарного отрезка. Квадратурная формула называется формулой средних прямоугольников (рисунок *1.2б*) и имеет вид:

$$I \approx 2h \sum_{i=0}^{2n} f(x_{2i+1}). \qquad (1.28)$$

Необходимо отметить, что формулы (1.26) и (1.27) имеют высокую погрешность. Это в равной степени относится и к формуле (1.28).

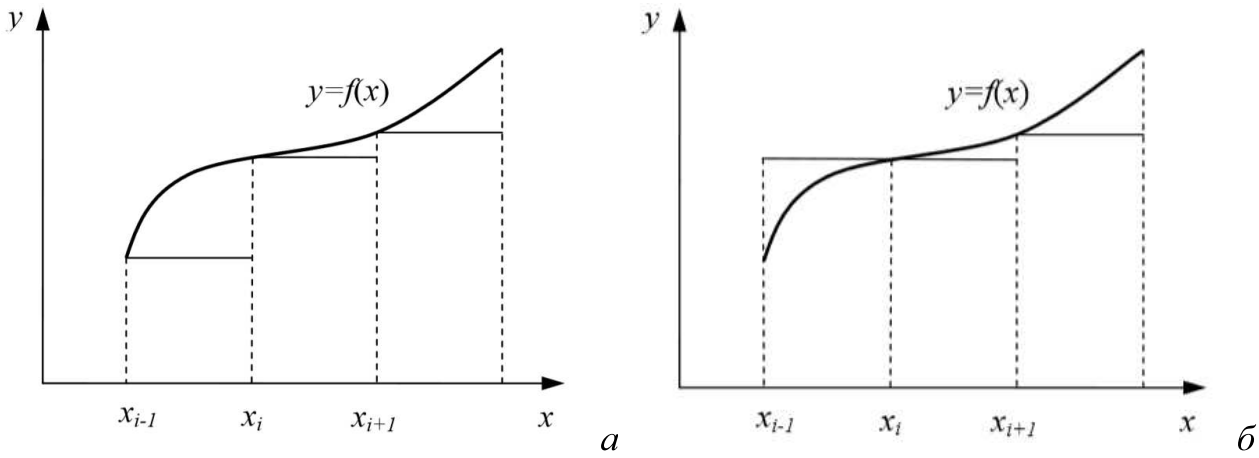

Рисунок 1.2 – Интегрирование по формуле левых (*а*)
и средних (*б*) прямоугольников

Формула трапеций. Рассмотрим в качестве приближения функции не прямоугольник, а трапецию с высотами $f(x_{i-1})$ и $f(x_i)$ (рисунок *1.3*). В этом случае площадь *i*-ой элементарной трапеции с основанием *h* будет иметь вид:

$$I_i^{mp} = \frac{h}{2}\left(f\left(x_{i-1}\right) + f\left(x_i\right)\right).$$

Приближенное значение интеграла определяется как:

$$I \approx h\left(\frac{f\left(x_0\right) + f\left(x_n\right)}{2} + \sum_{i=1}^{n-1} f\left(x_i\right)\right). \qquad (1.29)$$

Формула (1.29) соответствует приближенной замене площади исходной криволинейной трапеции площадью фигуры, ограниченной ломаной линией.

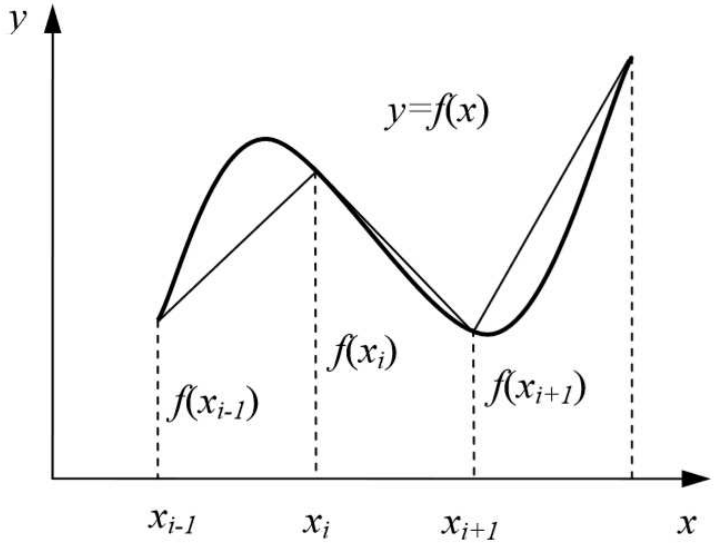

Рисунок 1.3 – Интегрирование по формуле трапеций

Метод Симпсона. Очевидно, что возможна аппроксимация функции $f(x)$

полиномом некоторой степени. Воспользуемся интерполяционным полиномом Ньютона второй степени $P_2(x)$ для решения данной задачи в следующем виде:

$$P_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \qquad (1.30)$$

Таким образом, на отрезке $[x_0, x_1]$ интеграл от $f(x)$ заменяется на интеграл от $P_2(x)$. Чтобы определить неизвестные коэффициенты этого полинома необходимо знать значения $f(x)$ в трех точках, чем и определяется длина интервала. Определим неизвестные коэффициенты a_0, a_1 и a_2. Для интерполяционных полиномов их значения в узлах совпадают со значением интерполируемой функции. Получаем:

$$a_0 = f(x_0),$$
$$a_1 = \frac{f(x_1) - f(x_0)}{h},$$
$$a_2 = \frac{f(x_0) - 2f(x_1) + f(x_2)}{h^2},$$

где $h = x_i - x_{i\text{-}1}$.

Введем новую переменную $z = x - x_0$, тогда полином (1.30) примет вид:

$$P_2(z) = a_0 + (a_1 - a_2)z + a_2 z^2. \qquad (1.31)$$

Вычислим интеграл от полинома (1.31):

$$\int_{x_0}^{x_2} P_2(x)dx = \int_0^{2h} P_2(z)dz = \frac{h}{3}\left(f(x_0) + 4f(x_1) + f(x_2)\right). \qquad (1.32)$$

Соотношение (1.32) представляет собой метод Симпсона (рисунок 1.4) или метод парабол. Рассмотренные ранее простейшие методы вычисления интегралов (методы правых, левых прямоугольников и трапеций) представляют собой методы Ньютона-Котеса нулевого и первого порядка, а метод Симпсона – второго порядка. Формулы Ньютона-Котеса точны для многочленов степени n, которые определяют количество интервалов разбиения интегрируемого отрезка.

Формулы вычисления погрешности R для описанных методов приведены в таблице 1.2.

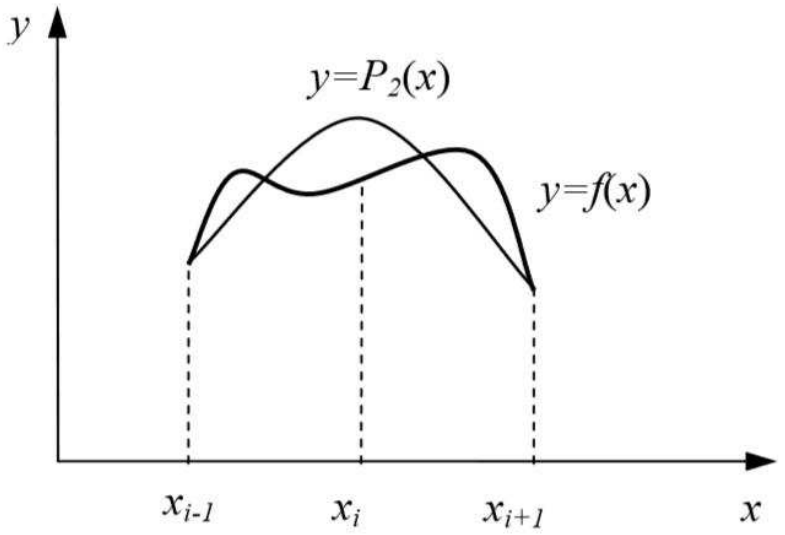

Рисунок 1.4 – Метод Ньютона-Котеса второго порядка (метод Симпсона)

Таблица 1.2. – Формулы вычисления погрешности

Название метода	Выражение
Метод левых и правых прямоугольников	$\lvert R\rvert \le \frac{b-a}{2} Mh,\ M = \max\limits_{x\in[a,b]} \lvert f'(x)\rvert$
Метод средних прямоугольников	$\lvert R\rvert \le \frac{b-a}{3} Mh^2,\ M = \max\limits_{x\in[a,b]} \lvert f''(x)\rvert$
Метод трапеций	$\lvert R\rvert \le \frac{b-a}{12} Mh^2,\ M = \max\limits_{x\in[a,b]} \lvert f''(x)\rvert$
Формула Симпсона	$\lvert R\rvert \le \frac{b-a}{180} Mh^4,\ M = \max\limits_{x\in[a,b]} \lvert f^{(4)}(x)\rvert$

1.7.2 Порядок выполнения работы

1. В соответствии с выданным преподавателем вариантом реализовать программу в Octave для расчета интеграла функции по формулам левых и правых прямоугольников и формуле трапеций. Сравнить результаты между собой. Шаг интегрирования взять равным $|b-a|/15,\ x\in[0,1]$.

Вариант 1 $f(x)=\sin 5x+2x$

Вариант 2 $f(x)=\cos 8x+3$

Вариант 3 $f(x)=10x\exp(-5x)$

Вариант 4 $f(x) = \left(2x^2 - 3\right) / (2x - 3)$

Вариант 5 $f(x) = \cos 7x + 3x$

Вариант 6 $f(x) = 1 / \sqrt{1 + x^2}$

Вариант 7 $f(x) = \sinh x + \sin 5x$

Вариант 8 $f(x) = \cosh x + \cos 4x$

2. В соответствии с выданным преподавателем вариантом реализовать программу в Octave для расчета интеграла функции по формулам средних прямоугольников и Симпсона. Сравнить результаты между собой. Шаг интегрирования взять равным $|b - a| / 15, x \in [0,1]$.

Вариант 1 $f(x) = |\cos 2x|$

Вариант 2 $f(x) = \sin 4x \cos 2x + 1$

Вариант 3 $f(x) = \cos^2 2x + x$

Вариант 4 $f(x) = \left|3x^2 + x - 1\right|$

Вариант 5 $f(x) = x \cosh(3x - 2)$

Вариант 6 $f(x) = x^2 \sin 3x$

Вариант 7 $f(x) = x \sinh(2x - 3) + 3$

Вариант 8 $f(x) = 1 / \sqrt{1 - x^2}$

3. Оценить погрешности вычисления интегралов. Для вычисления производных воспользоваться средствами символьных вычислений Octave.

4. С помощью стандартных средств Octave вычислить интеграл с относительной погрешностью 10^{-3}.

5. Сравнить полученные результаты п. 1–4.

1.7.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

1.8 Интерполяция и аппроксимация

Цель работы – изучение и программная реализация методов аппроксимации и интерполяции таблично-заданных функций.

Для достижения поставленной цели используется программное обеспечение GNU Octave. В ходе работы необходимо программно реализовать метод наименьших квадратов для нахождения коэффициентов аппроксимирующей функции и метод нахождения интерполяционного полинома Ньютона с расчетом значения функции вне узловых точек.

1.8.1 Краткая теоретическая справка

Основу математических моделей многих процессов и явлений во всех областях инженерии и науки составляют уравнения различного вида. Для решения подобных уравнений необходимо иметь возможность вычислять значения функций, входящих в описание математической модели рассматриваемого процесса или явления для произвольного значения аргумента. Для сложных моделей подобные вычисления могут быть трудоемкими даже при использовании компьютера.

Используемые в математических моделях функции могут быть заданы как аналитическим способом, так и в табличном виде, при котором функция известна при определенных дискретных значениях аргумента. На практике могут быть необходимы значения функции и в других точках, отличных от заданных. Таким образом, требуется вычислить приближенные значения

функции при любом значении аргумента на основе табличных данных. Эта задача решается путем приближенной замены функции $f(x)$ более простой функцией $g(x)$, которая вычисляется при любом значении аргумента в заданном интервале. Введенную функцию можно использовать не только для приближенного определения численных значений $f(x)$, но и для проведения аналитических расчетов при теоретическом исследовании модели.

Приближение функции $f(x)$ более простой функцией $g(x)$ называется аппроксимацией. Аппроксимирующую функцию $g(x)$ строят таким образом, чтобы отклонения $g(x)$ от $f(x)$ в заданной области было наименьшим. Широко используемым методом аппроксимации является метод наименьших квадратов. Идея метода заключается в том, что аппроксимирующую функцию $f(x, a_1, a_2, \dots, a_k)$ необходимо подобрать таким образом, чтобы сумма квадратов отклонений измеренных значений y_i от расчетных $Y_i = f(x_i, a_1, a_2, \dots, a_k)$ была наименьшей:

$$S(a_1, a_2, \dots, a_k) = \sum_{i=1}^{n} (y_i - Y_i)^2 = \\ = \sum_{i=1}^{n} (y_i - f(x_i, a_1, a_2, \dots, a_k))^2 \to \min. \tag{1.33}$$

Задача состоит из двух этапов:

1. По результатам эксперимента определить внешний вид подбираемой зависимости.

2. Подобрать коэффициенты зависимости $Y = f(x, a_1, a_2, \dots, a_k)$.

Математически задача подбора коэффициентов зависимости сводится к определению коэффициентов a_i из условия (1.33). В Octave её можно решить несколькими способами:

1. Поиском минимума функции многих переменных без ограничений с использованием функции *sqp*.

2. С помощью функции *polyfit*(x, y, n).

3. С помощью аппарат высшей математики: составить и решить СЛАУ для определения коэффициентов a_i.

Стоит отметить, что метод наименьших квадратов не может дать ответа на вопрос о том, какого вида функция лучше всего аппроксимирует экспериментальные данные. Вид функции должен быть задан на основании каких-либо имеющихся физических соображений.

Рассмотрим пример определения коэффициентов линейной функции $Y = a_1 + a_2 \cdot x$. Для этого составим функцию:

$$S(a_1, a_2) = \sum_{i=1}^{n} (y_i - a_1 - a_2 x_i)^2 \to \min .$$

Продифференцировав функцию S по a_1 и a_2, получим систему уравнений:

$$\begin{cases} 2\sum_{i=1}^{n} (y_i - a_1 - a_2 x_i)(-1) = 0 \\ 2\sum_{i=1}^{n} (y_i - a_1 - a_2 x_i)(-x_i) = 0 \end{cases} \Rightarrow \begin{cases} a_1 n + a_2 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \\ a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i \end{cases},$$

решив которую, определим требуемые коэффициенты:

$$\begin{cases} a_1 = \dfrac{1}{n}\left(\sum_{i=1}^{n} y_i - a_2 \sum_{i=1}^{n} x_i \right) \\ a_2 = \dfrac{n\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i}{n\sum_{i=1}^{n} x_i^{\,2} - \left(\sum_{i=1}^{n} x_i \right)^2} \end{cases}.$$

Далее рассмотрим задачу интерполирования с помощью полинома в форме Ньютона. Так, пусть на отрезке $[a, b]$ заданы $n + 1$ точек $x_0, x_1, x_2, \ldots, x_n$ ($a = x_0$, $b = x_n$), называемых узлами интерполяции, и значения неизвестной функции $f(x)$ в этих точках:

$$f(x_0) = y_0,\ f(x_1) = y_1,\ f(x_2) = y_2, \ldots,\ f(x_n) = y_n . \qquad (1.34)$$

Требуется построить интерполирующую функцию $F(x)$, которая в узлах интерполяции принимает те же значения, что и $f(x)$:

$$F(x_0) = y_0,\ F(x_1) = y_1,\ F(x_2) = y_2, \ldots,\ F(x_n) = y_n .$$

Полученную интерполяционную формулу $y = F(x)$ зачастую используют для приближённого нахождения значений данной функции $f(x)$ в точках x, отличных от узлов интерполирования.

Рассмотрим интерполяцию полиномами Ньютона и понятие разделенных разностей. Пусть задана функция $y = f(x)$ на отрезке $[x_0, x_n]$, который разбит на n одинаковых отрезков (случай равноотстоящих значений аргумента), $\Delta x = h = const$. Для каждого узла x_0, $x_1 = x_0 + h$, ..., $x_n = x_0 + n \cdot h$ значения функции определены в виде (1.34). Введем понятия разделенных разностей. Так, разности первого порядка:

$$\begin{cases} \Delta y_0 = y_1 - y_0 \\ \Delta y_1 = y_2 - y_1 \\ \cdots \\ \Delta y_{n-1} = y_n - y_{n-1} \end{cases}.$$

Разделенные разности второго порядка:

$$\begin{cases} \Delta^2 y_0 = \Delta y_1 - \Delta y_0 \\ \Delta^2 y_1 = \Delta y_2 - \Delta y_1 \\ \cdots \\ \Delta^2 y_{n-2} = \Delta y_{n-1} - \Delta y_{n-2} \end{cases},$$

и разности высших порядков:

$$\begin{cases} \Delta^k y_0 = \Delta^{k-1} y_1 - \Delta^{k-1} y_0 \\ \Delta^k y_1 = \Delta^{k-1} y_2 - \Delta^{k-1} y_1 \\ \cdots \\ \Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i, i = 0,1,\ldots,n-k \end{cases}.$$

Данные разности удобно располагать в виде таблиц, которые могут быть диагональными (таблица 1.3) или горизонтальными (таблица 1.4).

Пусть для функции $y = f(x)$ заданы значения $y_i = f(x_i)$ для равностоящих значений независимых переменных:

$$x_n = x_0 + nh,$$

где h – шаг интерполяции. Необходимо найти полином $P_n(x)$ степени не выше n, принимающий в точках (узлах) x_i соответствующие значения функции:

$$P_n(x_i) = y_i,\ i = 0, \ldots, n.$$

Тогда интерполирующий полином ищется в виде:

$$P_{\text{n}}(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots$$
$$\ldots + a_n(x - x_0)\ldots(x - x_{n-1}). \qquad (1.35)$$

Таблица 1.3. – Диагональная таблица разделенных разностей

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$
x_0	y_0					
		Δy_0				
x_1	y_1		$\Delta^2 y_0$			
		Δy_1		$\Delta^3 y_0$		
x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$	
		Δy_2		$\Delta^3 y_1$		$\Delta^5 y_0$
x_3	y_3		$\Delta^2 y_2$		$\Delta^4 y_1$	
		Δy_3		$\Delta^3 y_2$		
x_4	y_4		$\Delta^2 y_3$			
		Δy_4				
x_5	y_5					

Таблица 1.4 – Горизонтальная таблица разделенных разностей

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_{\text{i}}$	$\Delta^4 y_i$	$\Delta^5 y_i$
x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$	$\Delta^5 y_0$
x_1	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	$\Delta^4 y_1$	
x_2	y_2	Δy_2	$\Delta^2 y_2$	$\Delta^3 y_2$		
x_3	y_3	Δy_3	$\Delta^2 y_3$			
x_4	y_4	Δy_4				
x_5	y_5					

Таким образом, задача построения многочлена сводится к определению коэффициентов a_i из условий:

$$\begin{cases} P_n(x_0) = y_0 \\ P_n(x_1) = y_1 \\ \quad \cdots \\ P_n(x_n) = y_n \end{cases}.$$

Полагая $x = x_0$, найдем a_0:

$$P_{\text{n}}(x_0) = a_0 = y_0.$$

Полагая $x = x_1$, найдем a_1:

$$P_{\text{n}}(x_1) = y_1 = y_0 + a_1(x_1 - x_0) \Rightarrow$$

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\Delta y_0}{h}.$$

Для определения a_2 составим разделенную разность второго порядка. Тогда при $x = x_2$ получим:

$$P_n(x_2) = y_2 = y_0 + \frac{\Delta y_0}{h}(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = y_0 + 2\Delta y_0 + a_2 2h^2,$$

$$a_2 = \frac{y_2 - y_0 - 2\Delta y_0}{2h^2} = \frac{y_2 - y_0 - 2y_1 + 2y_0}{2h^2} = \frac{y_2 - 2y_1 + y_0}{2h^2} =$$

$$= \frac{(y_2 - y_1) - (y_1 - y_0)}{2h^2} = \frac{\Delta y_1 - \Delta y_0}{2h^2} = \frac{\Delta^2 y_0}{2!h^2}.$$

Аналогично можно найти и другие коэффициенты. Тогда общая формула имеет вид:

$$a_k = \frac{\Delta^k y_0}{k!h^k}, k = 1 \ldots n. \qquad (1.36)$$

Подставляя выражение (1.36) в (1.35), получим:

$$P_n(X) = y_0 + \frac{\Delta y_0}{1!\cdot h}(x - x_0) + \frac{\Delta^2 y_0}{2!\cdot h^2}(x - x_0)(x - x_1) + \ldots$$

$$\ldots + \frac{\Delta^n y_0}{n!\cdot h^n}(x - x_0)\ldots(x - x_{n-1}),$$

где x_i, y_i – узлы интерполяции, x – текущая переменная, h – разность между двумя узлами интерполяции.

Для наглядности рассмотрим следующий пример. Пусть дана таблица значений теплоёмкости вещества в зависимости от температуры $C_p = f(T)$.

x(T)	300	400	500	600
Y(C_p)	52,88	65,61	78,07	99,24

Требуется построить интерполяционный полином Ньютона для заданных значений функции при $n = 3$, $h = 100$. Для этого составим таблицу разделенных разностей функции. С помощью описанного выше подхода получим:

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
300	52,88	12,73	-0,27	8,98
400	65,61	12,46	8,71	
500	78,07	21,17		
600	99,24			

Тогда

$$P_3(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1) +$$
$$+ \frac{\Delta^3 y_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2),$$

или

$$P_3(x) = 52.88 + \frac{12.73}{100}(x - 300) - \frac{0.27}{2!100^2}(x - 300)(x - 400) +$$
$$+ \frac{8.98}{3!100^3}(x - 300)(x - 400)(x - 500).$$

После выполнения преобразований получим интерполяционный многочлен Ньютона вида:

$$P_3(x) = 1.5 \cdot 10^{-6} x^3 - 0.00181x^2 + 0.842x - 76.94.$$

Данный полином имеет третью степень и дает возможность вычисления значений y для неизвестных значений x.

1.8.2 Порядок выполнения работы

1. В соответствии с выданным преподавателем вариантом определить тип зависимости таблично заданной функции, рассчитать коэффициенты аппроксимирующей функции с помощью метода наименьших квадратов. Вывести график таблично заданной функции и аппроксимирующей функции. Оценить максимальную погрешность аппроксимации.

Вариант 1

x	0,23	0,57	0,94	1,48	2,03
y	1,45	1,65	1,89	2,34	3,02

Вариант 2

x	0,23	0,57	0,94	1,48	2,03
y	1,45	0,99	0,65	0,73	1,3

Вариант 3

x	1	2	3	4	5
y	5,3	6,3	4,8	3,8	3,3

Вариант 4

x	1	2	3	4	5
y	–5,3	–6,3	–4,8	–3,8	–3,3

Вариант 5

x	0,23	0,57	0,94	1,48	2,03
y	–1,45	–1,65	–1,89	–2,34	–3,02

Вариант 6

x	1	2	3	4	5
y	5,3	6,3	7,3	8,6	9,2

Вариант 7

x	1	2	3	4	5
y	2,1	2,4	2,6	2,8	3,0

Вариант 8

x	1	2	3	4	5
y	–100	–90	–75	–52	–12

2. В соответствии с выданным преподавателем вариантом выбрать таблично заданную функцию. Реализовать программу в Octave для расчета интерполяционного полинома по формуле Ньютона и вычислить численное значение полинома в заданных точках ($x_{(1)}$ и $x_{(2)}$) . Вывести график таблично заданной функции и интерполирующего полинома.

Вариант 1

x	0,1	0,3	0,5	0,7	0,9	1,1	1,3	1,5	1,7	1,9
y	0,9950	0,9553	0,8776	0,7646	0,6216	0,4536	0,2075	0,0707	–0,128	–0,323

$x_{(1)} = 0{,}2$ и $x_{(2)} = 2{,}0$

Вариант 2

x	2,0	2,2	2,4	2,6	2,8	3,0	3,2	3,4	3,6	3,8
y	0,909	0,808	0,6755	0,515	0,335	0,1411	–0,058	–0,255	–0,442	–0,611

$x_{(1)} = 1{,}9$ и $x_{(2)} = 3{,}7$

Вариант 3

x	0,1	0,3	0,5	0,7	0,9	1,1	1,3	1,5	1,7	1,9
y	0,9093	0,8085	0,6755	0,5155	0,3350	0,1411	–0,058	–0,255	–0,442	–0,611

$x_{(1)} = 0{,}17$ и $x_{(2)} = 1{,}91$

Вариант 4

x	2,0	2,2	2,4	2,6	2,8	3,0	3,2	3,4	3,6	3,8
y	–0,416	–0,588	–0,737	–0,859	–0,942	–0,990	–0,966	–0,896	–0,791	–0,670

$x_{(1)} = 1{,}9$ и $x_{(2)} = 3{,}7$

Вариант 5

x	6,3	6,5	6,7	6,9	7,1	7,3	7,5	7,7	7,9	8,1
y	0,0168	0,2151	0,4048	0,5784	0,729	0,8504	0,938	0,9882	0,9989	0,9699

$x_{(1)} = 6{,}2$; $x_{(2)} = 7{,}6$

Вариант 6

x	0,72	0,92	1,12	1,32	1,52	1,72	1,92	2,12	2,32	2,52
y	0,4868	0,3985	0,3269	0,2671	0,2187	0,1791	0,1446	0,1200	0,0983	0,0805

$x_{(1)} = 1{,}6$; $x_{(2)} = 0{,}8$

Вариант 7

x	0,45	0,50	0,55	0,60	0,65	0,70	0,75	0,80	0,85	0,90
y	0,4831	0,5463	0,6131	0,6841	0,7602	0,8423	0,9316	1,0296	1,1382	1,2602

$x_{(1)} = 0{,}48$; $x_{(2)} = 0{,}87$

Вариант 8

x	0,47	0,52	0,57	0,62	0,67	0,72	0,77	0,82	0,87	0,92
y	0,508	0,5726	0,641	0,7139	0,7922	0,877	0,9696	1,0717	1,1853	1,3133

$x_{(1)} = 0{,}48$; $x_{(2)} = 0{,}9$

1.8.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

1.9 Численное дифференцирование

Цель работы – изучение теоретических основ и программная реализация методов численного дифференцирования таблично и аналитически заданных функций.

Для достижения поставленной цели используется программное обеспечение GNU Octave. В ходе работы необходимо программно реализовать алгоритм нахождения первой и второй производной таблично заданной функции с помощью первого интерполяционного многочлена Ньютона, а также реализовать формулы левосторонней, правосторонней и центральной конечно-разностной аппроксимации производных.

1.9.1 Краткая теоретическая справка

При решении инженерно-технических задач часто встает задача нахождения производной определенного порядка от функции $f(x)$, заданной таблично. Кроме того, в силу сложности её аналитического выражения, непосредственное дифференцирование этой функции является затруднительным. В этих случаях обычно используют численное дифференцирование. Одной из простейших формул для вычисления производной функции является формула центральной конечно-разностной аппроксимации:

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_{k-1})}{2h},$$

где $h = x_k - x_{k-1} = x_{k+1} - x_k$ – шаг между значениями таблично заданной функции.

Также применяются формулы левосторонней и правосторонней аппроксимации:

$$f'_л(x_k) = \frac{f(x_k) - f(x_{k-1})}{h},$$

$$f'_п(x_k) = \frac{f(x_{k+1}) - f(x_k)}{h}.$$

Также функцию $f(x)$, определенную на отрезке $[a, b]$, заменяют интерполирующей функцией, полагая:

$$f'(x) = P'(x),$$

при $a \le x \le b$. Аналогичным образом вычисляют производные функции $f(x)$ высших порядков.

Если известна погрешность интерполирующей функции $P(x)$:

$$R(x) = f(x) - P(x),$$

то погрешность производной $P'(x)$ равна:

$$r(x) = R'(x) = f'(x) - P'(x).$$

Таким образом, погрешность производной интерполирующей функции равна производной от погрешности этой функции. Это справедливо и для

производных высших порядков. Численное дифференцирование является менее точной операцией, чем интерполирование функции. Иными словами, близость друг к другу ординат функций $f(x)$ и $P(x)$ на отрезке $[a, b]$ не гарантирует близости на этом отрезке их производных.

Для нахождения первой и второй производных функцию y, заданную в равноотстоящих точках x_i $(i = 0, 1, 2, \ldots, n)$ отрезка $[a, b]$ значениями $y_i = f(x_i)$, приближенно заменяют интерполяционным многочленом Ньютона, построенным для системы узлов x_i:

$$y(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + \frac{q(q-1)(q-2)}{3!}\Delta^3 y_0 + \\ + \frac{q(q-1)(q-2)(q-3)}{4!}\Delta^4 y_0 + \cdots, \tag{1.37}$$

где $q = \dfrac{x - x_0}{h}$.

Раскрыв скобки в (1.37) и учитывая, что

$$\frac{dy}{dx} = \frac{dy}{dq}\cdot\frac{dq}{dx} = \frac{1}{h}\cdot\frac{dy}{dq},$$

получим:

$$y'(x) = \frac{1}{h}\left[\Delta y_0 + \frac{2q-1}{2}\cdot\Delta^2 y_0 + \frac{3q^2 - 6q + 2}{6}\Delta^3 y_0 + \frac{2q^3 - 9q^2 + 11q - 3}{12}\Delta^4 y_0 + \cdots\right].$$

Аналогичным образом рассчитывается вторая производная:

$$y'' = \frac{d(y')}{dx} = \frac{d(y')}{dq}\cdot\frac{dq}{dx} = \frac{1}{h}\cdot\frac{d(y')}{dq},$$

$$y''(x) = \frac{1}{h^2}\left[\Delta^2 y_0 + (q-1)\Delta^3 y_0 + \frac{6q^3 - 18q^2 + 11}{12}\Delta^4 y_0 + \cdots\right].$$

Рассмотрим пример. Требуется найти первую и вторую производную в точке $x = 1$ функции $y(x)$, заданной таблично:

x	0,96	0,98	1,00	1,02	1,04
y	0,7825361	0,7739332	0,7651977	0,7563321	0,7473390

Составим таблицу разделенных разностей до 4 порядка включительно:

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0,96	0.7825361				
		–86029			
0,98	0,7739332		–1326		
		–87355		25	
1,00	0,7651977		–1301		1
		–88656		26	
1,02	0,7563321		–1275		
		–89931			
1,04	0,7473390				

В соответствии с таблицей получим: $h = 0{,}02$, $x = 1$, $x_0 = 1$, следовательно, $q = 0$. Используя рассчитанные значения разностей, получим:

$$y'(1) = \frac{1}{0{,}02}\left[-\frac{87355 + 88656}{2} \cdot 10^{-7} - \frac{1}{6} \cdot \frac{25 + 26}{2} \cdot 10^{-7} + \frac{1}{30} \cdot 0 \cdot 10^{-7} \right] = -0{,}4400485,$$

$$y''(1) = \frac{1}{0{,}02^2}\left[-1301 \cdot 10^{-7} - \frac{1}{12} \cdot 1 \cdot 10^{-7} \right] = -0{,}325250.$$

1.9.2 Порядок выполнения работы

1. В соответствии с выданным преподавателем вариантом выбрать таблично заданную функцию и реализовать в Octave программу для нахождения её первой и второй производной в заданной точке (помечена жирным) с помощью первого интерполяционного полинома Ньютона. Составить таблицу разделенных разностей до пятого порядка включительно.

Вариант 1

x	0,47	0,52	0,57	**0,62**	0,67	0,72	0,77	0,82	0,87	0,92
y	0,508	0,572	0,641	0,713	0,792	0,877	0,969	1,071	1,185	1,313

Вариант 2

x	0,45	0,50	0,55	0,60	0,65	**0,70**	0,75	0,80	0,85	0,90
y	0,483	0,546	0,613	0,684	0,760	0,842	0,931	1,029	1,138	1,260

Вариант 3

x	0,92	1,12	1,32	1,52	1,72	1,92	**2,12**	2,32	2,52	0,72
y	0,398	0,326	0,267	0,218	0,179	0,144	0,120	0,098	0,080	0,486

Вариант 4

x	6,3	6,5	6,7	6,9	7,1	7,3	7,5	**7,7**	7,9	8,1
y	0,016	0,215	0,404	0,578	0,729	0,850	0,938	0,988	0,998	0,969

Вариант 5

x	2,0	2,2	2,4	2,6	2,8	3,0	3,2	3,4	**3,6**	3,8
y	–0,41	–0,58	–0,73	–0,85	–0,94	–0,99	–0,96	–0,89	–0,79	–0,67

Вариант 6

x	0,1	0,3	0,5	0,7	0,9	1,1	1,3	1,5	1,7	**1,9**
y	0,909	0,808	0,675	0,515	0,335	0,141	–0,05	–0,25	–0,44	–0,61

Вариант 7

x	**2,0**	2,2	2,4	2,6	2,8	3,0	3,2	3,4	3,6	3,8
y	0,909	0,808	0,675	0,515	0,335	0,141	–0,05	–0,25	–0,44	–0,61

Вариант 8

x	0,1	**0,3**	0,5	0,7	0,9	1,1	1,3	1,5	1,7	1,9
y	0,995	0,955	0,877	0,764	0,621	0,453	0,207	0,070	–0,12	–0,32

2. В соответствии с выданным преподавателем вариантом выбрать функцию и вычислить её первую производную при шаге дифференцирования: $h = 10^{-1}$, 10^{-3} и 10^{-5}. Сравнить полученные результаты с результатами аналитически вычисленной производной. Построить график зависимости погрешности численного дифференцирования от шага h.

Вариант 1 $f(x) = \ln(x)$

Вариант 2 $f(x) = \exp(x)$

Вариант 3 $f(x) = \log(x)$

Вариант 4 $f(x) = x^2 - 16x^3$

Вариант 5 $f(x) = 3x \cdot e^x$

Вариант 6 $f(x) = \ln\left(\frac{1}{2x}\right)$

Вариант 7 $f(x)=15\sin(2x)$

Вариант 8 $f(x)=\cos(x)\cdot x$

1.9.3 Содержание и требования к оформлению отчета

Отчет должен содержать титульный лист, название работы и цель работы, исходные данные, результаты расчетов, таблицы и графики, анализ результатов и выводы по работе.

Оформление должно соответствовать ОС ТУСУР 01-2013 "работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления".

2 Методические указания к самостоятельным работам

В процессе подготовки к лабораторным работам, студентам необходимо обратить особое внимание на проработку лекционного материала и самостоятельное изучение рекомендованной учебно-методической, а также научной и популярной литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем, что позволяет студентам выявить широкий спектр мнений по изучаемой проблеме.

Для получения больших навыков работы с пакетом GNU Octave рекомендуется рассмотрение тестовых примеров и их усовершенствование, изучение новых функций и прочих особенной программной реализации кода.

Список использованных источников

1. Алексеев Е.Р. GNU Octave для студентов и преподавателей / Е.Р. Алексеев, О.В. Чеснокова // Донецк.: ДонНТУ, Технопарк ДонНТУ УНИТЕХ. –2011. – С. 332.
2. Алексеев Е.Р. Scilab: Решение инженерных и математических задач / Е.Р. Алексеев, О.В. Чеснокова, Е.А. Рудченко. – М.: ALT Linux; БИНОМ. Лаборатория знаний, 2008. – 260 с.
3. Квасов В.И. Численные методы анализа и линейной алгебры. Использование Matlab и Scilab. – СПб.: Издательство «Лань», 2016. – 328 с.
4. Горбаченко В.И. Вычислительные методы линейной алгебры: Лабораторный практикум в системе MATLAB / В.И. Горбаченко, Г.Ф. Убиенных. – Пенза: Изд-во ПГУ, 2010. – 93 с.
5. Луппова Е.П. Вычислительная математика: Учебное пособие / Е.П. Луппова, П.В. Зиновьев. – Владивосток: ДВГТУ, 2005. – 30 с.
6. Куксенко С.П., Газизов Т.Р. Использование методов решения СЛАУ: Учебное методическое пособие. – Томск: кафедра ТУ, ТУСУР, 2012. – 63 с.