*Государственное образовательное учреждение   
высшего профессионального образования*

**Московский технический университет связи и информатики**

***Кафедра информатики***

Семенова Т.И., Шакин В.Н.,Загвоздкина А.В.

**Математический пакет Scilab**

**Практикум**

Для студентов по дисциплине   
«Введение в математические пакеты   
прикладных программ»,

обучающихся по направлению 11.03.02 подготовки бакалавров

**Москва, 2019 г.**

***Scilab*** – это математический пакет прикладных программ, предназначенный для численных вычислений и являющийся свободно распространяемым аналогом пакета ***Matlab***. Пакет ***Scilab***, изучению которого посвящен данный практикум, доступен на официальном сайте [***www.scilab.org***](http://www.scilab.org).

Данный практикум может быть использован при изучении различных дисциплин, связанных с численными расчетами при решении прикладных задач. Цель практикума – познакомить студентов с назначением основных окон Scilab, с элементами меню, содержащимися в них командами, с работой по созданию функций пользователя в ***Командном окне*** и в текстовом редакторе, операциями над матрицами, возможностями визуализации результатов вычислений и многими другими возможностями ***Scilab***. При этом большое внимание отводится средствам программирования, без которых не может быть выполнен практически ни один сложный расчет. Кроме того, практикум ориентирован и на решение задач, носящих вычислительный характер, поэтому большая часть предлагаемых работ посвящена решению вычислительных задач средствами Scilab. В Scilab численные методы реализованы в виде множества встроенных функций, доступных пользователю в ходе проведения расчета. Форматы всех используемых при выполнении лабораторных работ функций описаны в учебнике [1], в методических материалах, использующих пакет Scilab [2, 3, 4, 5], а также в справочной системе [6], встроенной в Scilab, обращение к которой может быть произведено с использованием соответствующего элемента меню или из командной строки.

Практикум состоит из двух Разделов. Раздел 1 – «Основы работы с математическим пакетом Scilab» предназначен для проведения 5-ти практических занятий. Выполнение каждого практического задания проиллюстрировано примером его выполнения. Раздел 2 – «Решение вычислительных задач средствами Scilab» содержит 6 практических занятий, где в основу каждого задания положено знание и практическое применение функций, реализующих численные методы [7, 8, 9], соответствующие данной теме. Выполнение заданий базируется на материале, изученном в 1-м разделе практикума.

# **1. Основы работы с математическим пакетом Scilab**

## **1.1. Практическое занятие по теме «Рабочая среда Scilab и простейшие вычисления»**

**Цель работы:** Изучение основных элементов ***Рабочей среды***, объектов и правил вычислений арифметических выражений в ***Командном окне*** системы Scilab.

## **Элементы системы Scilab, подлежащие изучению**

1. Графический интерфейс пользователя и элементы ***Рабочей среды***.
2. Правила работы в ***Командном окне***.
3. Назначение основных окон рабочей среды: ***Обозревателя переменных***, ***Журнала команд*** и ***Обозревателя файлов*** при работе в ***Командном окне***.
4. Установка свойств ***Рабочей среды*** системы Scilab.
5. Основные объекты системы Scilab.
6. Правила записи и вычисления арифметических выражений.
7. Создание и использование в ***Командном окне*** встроенных функций пользователя при вычислении выражений.

## **Общее задание**

1. ***Изучить материал учебника* [1] *(п.п. 1.1-1.2).***
2. ***Выбрать вариант индивидуального*** *задания из* ***табл. 1.1.3-1.***
3. *При выполнении задания* ***вводить*** *перед каждым пунктом задания соответствующие* ***комментарии****.*
4. ***Выполнить команды clear*** *и* ***clc для очистки окна Обозревателя переменных и Командного окна.***
5. ***Задать*** *переменным* **x** *и* **y** *допустимые числовые значения.*
6. ***Проанализировать информацию****, возникшую в окне* ***Обозреватель переменных.***
7. ***Ввести арифметическое выражение*** *для вычисления и* ***получить результат****.*
8. ***Изменить значения*** *исходных данных.*
9. ***Изменить формат*** *вывода результата, выполнив команду, например,* **format(16)***, и* ***произвести перерасчет*** *выражения.*
10. ***Изменить формат*** *вывода данных, например,***format(6)**.
11. ***Описать*** *с помощью оператора* **deff *функцию* fd(x)**, *правая часть которой будет заданным арифметическим выражением, и* ***вычислить*** *её значение при тех же значениях* **х** *и***y***.*
12. ***Описать*** *с помощью оператора***function*функцию* ff(x)**,*правая часть которой будет заданным арифметическим выражением, и* ***вычислить*** *её значение при тех же значениях* **х** *и***y***.*
13. ***Изменить*** *значение переменной* **y**, ***вывести на экран*** *значения описанного (в* **п.7***)арифметического выражения и функции* **ff(x)***.*
14. ***Объяснить****, почему изменение значения* ***y*** *не привело к изменению значения арифметического выражения, но повлияло на значение функции.*
15. ***Задать диапазон изменения*** *аргумента функции с шагом, позволяющим получить таблицу значений функции* **fd(x)** *и* **ff(x)**(*порядка* ***8-10*** *точек),* ***вывести вначале*** *значения функции* **fd(х)** *в строку, а затем значения функции* **ff(х)***в столбец*.
16. ***Cохранить*** *переменные* ***Рабочей области****,* ***обнулить Рабочую область****, а затем* ***восстановить ее****.*
17. ***Выполнить команду* who\_user** *и* ***проанализировать выведенную информацию*** *о данных*.
18. ***Сохранить текст*** *рабочего окна на внешнем носителе в Word.*
19. ***Предоставить результаты*** *работы преподавателю и, ответить на поставленные вопросы.*
20. ***Оформить отчет*** *по выполненной работе*.

## **Варианты индивидуальных заданий**

Таблица 1.1.3-1

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Формулы для  вычислений** | **№** | **Формулы для  вычислений** |
| **1.** |  | **16.** |  |
| **2.** |  | **17.** |  |
| **3.** |  | **18.** |  |
| **4.** |  | **19.** |  |
| **5.** |  | **20.** |  |
| **6.** |  | **21.** |  |
| **7.** |  | **22.** |  |
| **8.** |  | **23.** |  |
| **9.** |  | **24.** |  |
| **10.** |  | **25.** |  |
| **11.** |  | **26.** |  |
| **12.** |  | **27.** |  |
| **13.** |  | **28.** |  |
| **14.** |  | **29.** |  |
| **15.** |  | **30.** |  |

## **Пример выполнения задания**

1. **Пример индивидуального задания:**

Решить задачу вычисления арифметического выражения в соответствии с п.п.3-15 общего задания.

1. **Результат выполнения задания:**

Протокол сессии в ***Командном окне***:

-->//***Практическое занятие 1.1 по теме*** *-->*//***«Рабочая среда Scilab и простейшие вычисления»***

*-->*// *Вариант 0*

*-->*// *Выполнил:Студент группы БИН1801 Иванов И.И.*

*-->*

-->clear // *Очистка окна Обозревателя данных*

-->clc // *Очистка Командного окна*

-->//*Задание начальных значений* x *и* y

-->x=4.55;

-->y=1.5;

-->// *Ввод арифметического выражения*

-->z=(x+sqrt(abs(x\*y)))/(exp(x+y)\*log(x)^2)-cos(y^2)

z =

0.6355303

-->//*Изменение значений* x *и* y

-->x=8.8;y=5.5;

-->//*Изменение формата вывода результата*

-->format(16);

-->//*Перерасчет выражения*

-->z=(x+sqrt(abs(x\*y)))/(exp(x+y)\*log(x)^2)-cos(y^2)

z =

- 0.3938970292202

-->format(6);

-->//*Описание функции* fd *с использованием* **deff**

-->

-->deff('f1 = fd(x)','f1 = (x+sqrt(abs(x.\*y)))./…

(exp(x+y).\*log(x).^2)-cos(y.^2)');

-->

-->//*Вызов функции* fd *и вычисление ее значения*

-->f1=fd(x)

f1 =

- 0.394

-->//*Описание функции* ff *с использованием function*

-->

-->function [f2]=ff(x)

>f2 = (x+sqrt(abs(x.\*y)))./(exp(x+y).\*log(x).^2)-cos(y.^2);

>end

-->

-->//*Вызов функции* ff *и вычисление ее значения*

-->f2=ff(x)

f2 =

- 0.394

-->//*Изменение значения*y

-->y=0.1;

-->//*Вычисление значения выражения при новом значении* y

-->z

z =

- 0.394

-->//*Вычисление значения функции* ff *при новом значении* y

-->f2=ff(x)

f2 =

- 1.000

-->//*Задание диапазона изменения переменной* x

-->x=2:0.2:3.8

x =

2. 2.2 2.4 2.6 2.8 3. 3.23.4 3.6 3.8

-->// *Вычисление функции* fd *для заданного диапазона аргумента и*

*-->*// *вывод полученного результата в строку*

-->fd(x)

ans =

column 1 to 5

-0.376 -0.570 -0.690 -0.771 -0.827

column 6 to 10

-0.868 -0.897 -0.920 -0.937 -0.950

-->// *Вычисление функции* ff *для заданного диапазона аргумента* х*и*

*-->*//*вывод в столбец*

-->ff(x)'

ans =

-0.376

-0.570

-0.690

-0.771

-0.827

-0.868

-0.897

-0.920

-0.937

-0.950

--> // Конец сессии ========================================================

## **Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Протокол работы сессии ***Командного окна***, вначале которого должна быть введена информация в виде комментариев:

* название практического занятия;
* ФИО студента, номер группы;
* вариант индивидуального задания;
* протокол сессии должен быть снабженный подробными комментариями.

## **Контрольные вопросы по теме**

1. Из каких основных элементов состоит **Рабочая среда** Scilab?
2. Каково назначение ***Командного окна***?
3. Каково назначение окна ***Журнал команд***?
4. Каково назначение окна ***Обозревателя переменных***?
5. Каково назначение окна ***Обозреватель файлов***?
6. Как установить текущий каталог в окне ***Обозреватель файлов***?
7. Как установить ***Конфигурацию рабочего окна***?
8. Как называется сеанс работы с системой Scilab в ***Командном окне***?
9. Каким образом перенести командную строку из окна ***Журнал команд*** в окно ***Командное окно***?
10. Каким образом изменить значение переменной в окне ***Обозревателя переменных***?
11. Для чего предназначен редактор ***SciNotes***?
12. Как выбрать тип, размер и стиль шрифта в ***Командном окне***?
13. Для чего используется ***Справочная система*** Scilab?
14. Как вызвать окно ***Справочной системы*** с использованием меню и осуществить поиск нужной команды?
15. Какой командой вызывается справка из командного окна?
16. Что происходит, если, находясь в командной строке, нажать клавишу <**↑**> или <**↓**>?
17. Какой символ предназначен для запрета вывода результата выполнения действия на экран?
18. Какой символ служит для переноса части командной строки на следующую строку?
19. Какой формат имеет оператор описания функции **deff**?
20. Какой формат имеет оператор описания функции **function**?
21. В каких случаях используется оператор **deff**, а в каких **function**?

* 1. **Практическое занятие по теме  
     «Основные объекты системы Scilab»**

**Цель работы:** Изучение создания числовых и логических переменных и выражений, а также их вычисление, создание **sce**-сценариев и вывод результатов вычислений в отформатированном виде.

## **Элементы системы Scilab, подлежащие изучению**

1. Файловая система Scilab и Текущая папка.
2. Встроенный текстовый Редактор SciNotes.
3. Типы программных файлов.
4. **Sce**-сценарии.
5. Сохранение, вызов и выполнение сценария, сохраненного в **sce**-файле.
6. Cохранение и восстановление переменных Рабочей области.
7. Числовые выражения и форматирование их результатов.
8. Логические выражения.
9. Простейшие операторы ввода/вывода данных.

## **Общее задание**

1. ***Изучить материал учебника*** [1] *(п.п. 1.2, 1.5)*.
2. ***Выбрать вариант*** *индивидуального задания из табл. 1.2.3-1.*
3. ***Выполнить команды* clear** *и* **clc** *для очистки окна* ***Обозревателя переменных*** *и* ***Командного окна***.
4. ***Активизировать Редактор SciNotes.***
5. ***Создать* sce*-****сценарий для выполнения индивидуального задания (вычисление выражений) с использованием встроенных в сценарий функций.*
6. ***Сохранить* sce*-****сценарий в* **sce***-файле со смысловым именем.*
7. ***Провести отладку****,* ***исправить ошибки*** *и* ***сохранить*** *сценарий.*
8. ***Загрузить* sce*-****сценарий из* **sce***-файла и* ***выполнить*** его в ***Командном окне****, предварительно* ***присвоив исходным данным*** *допустимые значения.*
9. ***Модернизировать* sce***-сценарий,* ***вставив в него операторы*** *ввода исходных данных и вывода результата в форматированном виде.*
10. ***Сохранить* sce***-сценарий под другим именем,* ***исправить ошибки*** *и* ***выполнить****.*
11. ***Создать сценарий*** *для решения поставленной задачи и выполнить его.*
12. ***Сохранить все результаты для*** *отчета.*
13. ***Предоставить результаты*** *работы преподавателю,* ***ответить*** *на поставленные вопросы.*
14. ***Оформить отчет*** *по выполненной работе*.

## **Варианты индивидуальных заданий**

Таблица 1.2.3-1

|  |  |
| --- | --- |
| **№** | Задание |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 |  |
| 21 |  |
| 22 |  |
| 23 |  |
| 24 |  |
| 25 |  |
| 26 |  |
| 27 |  |
| 28 |  |
| 29 |  |
| 30 |  |

## **Пример выполнения задания**

***Пример выполнения задания***

1. **Решить задачу вычисления арифметических выражений**

 и 

в соответствии с п.п.3-10 общего задания**:**

1. **Результаты выполнения задания (сценарии и протоколы сессии)**

*//СценарийPr1\_2\_1.sce вычисления выражений*

*//Определение функции для вычисления 1-го выражения*

functionz=f1(x, y)

z=(x+sqrt(abs(x.\*y)))./(exp(x+y).\*log(x).^2)-cos(y.^2);

endfunction

*//Определение функции для вычисления2-го выражения*

function z=f2(x, y)

z=2\*x.^2+cos(x.\*y)-x.\*y.^2+5;

endfunction

*//Конец сценарияPr1\_2\_1.sce==============================================*



*-->*//*Протокол сессии Командного окна:*

*-->*// *Практическое занятие 1.2 по теме*

*-->*// *«Основные объекты системы Scilab»*

*-->*// *Вариант 0, задание 1*

*-->*//*Присвоение значений,вычисление выражений и вывод результатов*

*-->*// *Выполнил:Студент группы БИН1801 Иванов И.И.*

*-->*

-->clear // *Очистка окна Обозревателя данных*

-->clc // *Очистка Командного окна*

-->//*Загрузка 1-го сценария*Pr1\_2\_1*и вызов функций, встроенных в сценарий*

-->exec('Pr1\_2\_1.sce', 0);//*Загрузка сценария* Pr1\_2\_1.sce*из текущей папки*

-->

--> //*Задание начальных значений* x *и* y

-->x=2;y=0.1;

--> //*Вызов функций* f1 и f2*, встроенных в сценарий*

--> z=f1(x,y)

z =

-0.3762115

--> f=f2(x,y)

f =

13.960067

*// Конец сессии===========================================================*

*// Сценарий Pr1\_2\_1mod.sce вычисления выражений*

*//Определение функции 1-го выражения*

functionz=f1(x, y)

z =(x+sqrt(abs(x.\*y)))./(exp(x+y).\*log(x).^2)-cos(y.^2);

endfunction

*//Определение функции 2-го выражения*

function z=f2(x, y)

z =2\*x.^2+cos(x.\*y)-x.\*y.^2+5;

endfunction

*//Ввод исходных данных*

х=input("Введите х= ");

y=input("Введите y= ");

*//Вычисление значений функций*

z=f1(x,y);

f=f2(x,y);

*//Форматирование вывода результата*

format(6);

*//Вывод результатов вычисления выражений на экран*

disp(z,"z=");

disp(f,"f=");

*// Конец сценарияPr1\_2\_1mod.sce============================================*



*-->*// *Протокол сессии в Командном окне:*

*-->*// *Практическое занятие 1.2 по теме*

*-->*// *«Основные объекты системы Scilab»*

*-->*// *Вариант 0, задание 1 (модифицированное)*

*-->*//*Ввод исходных данных, вычисление выражений и вывод результатов*

*-->*// *Выполнил: Студент группы БИН1801 Иванов И.И.*

*-->*

--> clear // *Очистка окна Обозревателя данных*

--> clc // *Очистка Командного окна*

-->//*Загрузка 2-го сценария* Pr1\_2\_1mod

--> exec('Pr1\_2\_1mod.sce', 0); // *Загрузка сценария из текущей папки*

Введите х= 2

Введите y= 0.1

z=

-0.376

f=

13.96

*// Конец сессии=============================================================*

## **Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Сценарий, вначале которого должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* вариант индивидуального задания и номер задания.

1. Протокол сессии ***Командного окна***, вначале которого должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **Контрольные вопросы по теме**

1. Каким образом открыть новое окно редактора ***SciNotes***?
2. Как в окне ***Обозреватель файлов*** сделать нужный каталог текущим?
3. Каким образом открыть в ***SciNotes*** ранее созданный файл?
4. Что такое сценарий?
5. Имеет ли сценарий параметры?
6. Допускается ли в ***SciNotes*** при вводе текущей строки сценария вернуться к редактированию в предыдущих строках?
7. Какова последовательность действий при сохранении нового сценарий в **sce**-файле?
8. Каким образом сохранить исправленный текст **sce**-сценария?
9. Как в редакторе ***SciNotes*** осуществляется переход между открытыми окнами, содержащими разные сценарии?
10. Можно ли из одного сценария обращаться к другому?
11. Какой командой, файл, созданный в редакторе ***SciNotes*** загрузить в оперативную память?
12. Каким образом в команде **exec** указывается путь к загружаемому файлу со сценарием?
13. Как и где можно установить текущую папку?
14. Что такое глобальные переменные и когда они используются?
15. Если переменная создана в ***Командном окне***, является ли она видимой для загруженного сценария?
16. Какие переменные являются локальными?
17. Формат оператора ввода данных **input**?
18. Формат оператора ввода данных **x\_dialog**?
19. Как с использованием оператора **disp** осуществить вывод значения переменной или текста можно?
20. Как осуществить вывод результатов вычисления в форматированном виде?
21. Можно ли выполнить сценарий из редактора SciNotes?

**1.3 Практическое занятие по теме  
«****Матричные операции создания, доступа,  
извлечения и модификации»**

**Цель работы:** Изучение способов создания векторов и матриц, операций индексирования, поэлементных выражений с векторами и матрицами.

## **1.3.1 Вопросы, подлежащие изучению**

1. Создание векторов и матриц различного типа.
2. Индексация матриц: векторная, матричная и линейная.
3. Доступ к элементам вектора и матрицы.
4. Матричные операции: арифметических, логических и операций отношения.
5. Функции обработки элементов матриц.
6. Матричная и логическая индексация матриц.

## **1.3.2 Общее задание**

1. ***Изучить материал учебника* [1]** *(п.1.3).*
2. ***Выбрать вариант*** *индивидуального задания из* ***табл.1.3-1.*** *Каждое задание содержит две задачи.*
3. ***Выполнить команды* clear** *и* **clc *для очистки окна Обозревателя переменных*** *и* ***Командного окна.***
4. ***Открыть*** *новое окно* ***редактора SciNotes****.*
5. ***Создать*** *для выполнения* ***каждой задачи по два сценария****. Первый сценарий должен содержать функции, предназначенные для выполнений действий над векторами и матрицами с использованием матричных операций. Второй сценарий предназначен для ввода исходных данных, вызова первого сценария и вывода результатов. Здесь для вывода отдельных значений и пояснений следует* ***использовать*** *функцию* **mprintf***, а для вывода числовых массивов – функции* **mprintf** *или* **disp***.*
6. ***Сохранить* sce*-****сценарии в файлах,* ***исправить ошибки****.*
7. ***Решить индивидуальные задания,*** *выполнив созданные сценарии****.***
8. ***Сохранить все результаты*** *для отчета.*
9. ***Предоставить результаты*** *работы преподавателю, ответить на поставленные вопросы.*
10. ***Оформить отчет*** *по выполненной работе.*

## 

## **1.3.3 Варианты индивидуальных заданий**

Таблица 1.3-1

|  |  |  |
| --- | --- | --- |
| **№** | **1-я задача** | **2-я задача** |
| 1 | Ввести 3 вектора разной длины и найти сумму отрицательных элементов каждого из них. | В матрице **А(3х4)**определить произведение элементов, расположенных на нечетных местах в каждой строке. Заменить полученными значениями элементы первого столбца матрицы. |
| 2 | Подсчитать количество «единиц» на четных местах введенного вектора. | В матрице **Х(4x5)**поменять столбец, в котором находится максимальный элемент, с её первым столбцом. |
| 3 | Найти сумму положительных элементов введенного вектора, стоящих на местах, кратных числу **n**. | В матрице **T(3x5)**определить строки, в которых расположено более двух элементов, равных нулю. Заменить все элементы этих строк на **100**. |
| 4 | Найти сумму элементов введенного вектора, стоящих на местах кратных трем и больших заданного числа **x**. | Элементы каждой строки матрицы, значения которых больше среднеарифметического значения данной строки, заменить этим значением. |
| 5 | Найти сумму отрицательных элементов введенного вектора, расположенных на нечетных местах. | Все отрицательные элементы матрицы**С(3х4)** умножить на **х1**, а положительные – разделить на **х2**. Подсчитать количество нулевых элементов. |
| 6 | Найти произведение элементов введенного вектора, больших или равных, введенному значению **х**. | В введенной матрице **В**все элементы, модули которых равны модулям максимального или минимального элемента, заменить значением **y**. |
| 7 | Найти сумму последних **n** элементов введенного вектора, меньших числа **х**. | Найти сумму элементов прямоугольной матрицы **D**, не лежащих в интервале **[a,b]**. Заменить этим значением все элементы, попавшие в этот интервал. |
| 8 | Среди **n** первых элементов введенного вектора найти сумму отрицательных элементов. | Элементы квадратной матрицы ниже главной диагонали уменьшить на **х1,** а элементы выше главной диагонали увеличить на **х2**. |
| 9 | Найти сумму и количество элементов введенного вектора, больших числа **х**. | Найти количество отрицательных элементов в каждой строке квадратной матрицы **F(nxn)**и заменить полученными значениями элементы главной ее диагонали. |
| 10 | Подсчитать количество элементов введенного вектора, принадлежащих интервалу **[a,b].** | В матрице **А(4x5)**определить столбцы, в которых все одинаковые элементы. Заменить элементы этих столбцов случайными целыми числами от **0** до **100.** |
| 11 | Найти первый отрицательный элемент введенного вектора и вывести его номер и значение. | Максимальный элемент каждой строки матрицы **D(4,4)**заменить числом **х***.* |
| 12 | Найти количество элементов введенного вектора, меньших единицы. | Найти среднегеометрическое значение элементов каждого столбца квадратной матрицы и заменить им элементы главной диагонали матрицы **Y**. |
| 13 | Найти количество нулевых элементов введенного вектора среди его последних **n** элементов. | В каждой строке матрицы **К(m×n)** найти количество элементов больше **х1**и заменить их на значение последнего элемента строки. Вывести полученную матрицу по столбцам. |
| 14 | Найти произведение ненулевых элементов вектора. | Минимальный элемент в каждом из столбцов матрицы заменить нулевым значением. |
| 15 | Найти произведение положительных элементов вектора, расположенных на нечетных местах. | В матрице поменять строку, в которой находится минимальный элемент, с последней строкой. |
| 16 | Определить среднее значение всех отрицательных элементов вектора **Р**. | В матрице определить строки, в которых не все элементы одинаковы. Увеличить элементы этих строк на значение **х***.* |
| 17 | Найти сумму элементов вектора, не попавших в интервал **[a,b]**. | В матрице определить сумму отрицательных элементов каждого столбца. Заменить полученными значениями элементы второй строки. |
| 18 | Найти сумму отрицательных элементов вектора на четных местах. | Найти среднее арифметическое значений элементов матрицы **A(nxn)**и заменить этим значением элементы последней строки и последнего столбца. |
| 19 | Определить сумму положительных и произведение отрицательных элементов вектора. | В матрице определить столбцы, в которых расположено более двух нулевых элементов. Заменить элементы в этих столбцах назначение **х**. |
| 20 | Определить количество элементов вектора, кратных **x**. | Элементы каждого столбца матрицы, которые меньше среднеарифметического значения данного столбца, заменить этим значением. |
| 21 | Найти максимальное значение среди элементов вектора между номерами **n1** и **n2**. | В вещественной матрице определить суммы элементов, расположенных на нечетных местах в каждой строке. Заменить полученными значениями элементы последнего столбца матрицы. |
| 21 | Найти общее количество элементов вектора, равных **х1**и **х2**. | Максимальные элементы матрицы (их может быть несколько) увеличить на среднее значение всех элементов, расположенных выше главной диагонали. |
| 22 | Найти предпоследний отрицательный элемент введенного вектора Х. | Все элементы матрицы, кратные **х**, заменить минимальным значением элемента матрицы. |
| 23 | Подсчитать сумму и произведение первых **n** положительных элементов введенного вектора. | Во введенной матрице заменить все элементы, равные числу **х**, на среднее значение элементов матрицы. |
| 24 | Определить номер минимального по модулю элемента вектора-столбца. | Найти количество элементов матрицы**С(5х5)**, лежащих в интервале **[a,b]***,* и заменить этим значением элементы, лежащие на главной диагонали. |
| 25 | Найти наибольший отрицательный элемент вектора. | Найти произведение положительных элементов главной диагонали квадратной матрицы **А**. Заменить полученным значением все отрицательные элементы матрицы. |
| 26 | Подсчитать сумму квадратов четных и сумму квадратов нечетных элементов вектора. | В матрице определить произведения элементов, расположенных на четных местах в каждом столбце. Заменить полученными значениями элементы первой строки матрицы. |
| 27 | Вывести значение и номер последнего нечетного отрицательного элемента вектора. | Элементы матрицы уменьшить на минимальное значение её элементов, кратных **х**. |
| 28 | Определить элемент вектора, наиболее близкий по своему значению к заданному **x**. | В матрице симметрично поменять между собой элементы выше главной диагонали с элементами ниже главной диагонали, а элементы главной диагонали расположить в обратном порядке. |
| 29 | Определить первый максимальный элемент вектора и заменить его нулем. | Для матрицы **А** найти среднее арифметическое сумм элементов, расположенных над и под главной диагональю. |
| 30 | В векторе **Х** поменять местами минимальное и максимальное значения элементов. | Ввести квадратную матрицу и получить из ее элементов другую, каждый элемент которой умножен на значение ее определителя. |

## **1.3.4 Примеры выполнения заданий**

***Пример выполнения задания 1***

1. ***В соответствии с п.п.5-7 общего задания решить задачу 1:*** Среди **n** последних элементов вектора найти количество элементов, равных вводимому

числу **m**.

1. ***Решение 1-й задачи (сценарии и протоколы сессии)***

//*Первый сценарий–функция* poisk\_m\_vect.sce *для обработки вектора*

//*Среди n последних элементов вектора mas функция* poisk\_m\_vect

// *находит количество элементов, равных вводимому числу m*

// *Исходные данные: входные параметры числа m,n и векторmas;*

//*выходные параметры функции* mas2 и kol

function [mas2,kol]=poisk\_m\_vect(mas,m,n)

k=length(mas);//*Кол-во элементов в векторе mas*

//*Поиск числа m среди n последних элементов*

// *Вектор mas2 содержит индексы найденных элементов*

mas2=find(mas(k-n+1:$)==m);

// *Кол-во найденных элементов = длине созданного массива*

Kol=length(mas2);

endfunction

//Конец сценарияpoisk\_m\_vect.sce=========================================

//*Второй сценарий Pr1\_3\_1.sce для решения 1-й задачи*

// *Реализует загрузку сценария* poisk\_m\_vect.sce,*ввод исходных данных,*

// *обращение к функции* poisk\_m\_vect*и вывод результатов*

clear; // *Очистка окна Обозреватель данных*

clc; // *Очистка Командного окна*

//*Загрузка сценария обработки вектора*

exec('poisk\_m\_vect.sce');

// *Создание вектора и вводисходных данных для его обработки*

n=input("Введите число n =");

m=input("Введите искомое число m =");

k=input("Введите кол-во элементов вектора ");

a=input("Введите начальное значение диапазона для вектора ");

b=input("Введите конечное значение диапазона для вектора ");

//*Создание вектора в заданном диапазоне*

mas=grand(1,k,"uin",a,b);

//*Вывод исходного массива на экран*

disp(mas,"Исходный массив:");

//*Обращениек функции для обработки вектора*

[mas2,kol]= poisk\_m\_vect(mas,m,n);

//*Вывод массива mas2 с индексами*

disp(mas2,"Индексы найденных элементов в новом векторе");

// *Вывод результата решения задачи*

mprintf('Среди последних %1d элементов исходного массива \n',n);

mprintf('кол-во чисел =%1d равно %3d',m,kol);

//*Конец сценария Pr1\_3\_1.sce*===============================================



-->// *Сессия Командного окна*:

-->//*Практическое занятие 1.3 по теме*

-->// *«Матричные операции создания, доступа, извлечения*-->// *и модификации»*

*-->*// *Вариант 0, задание 1*

*-->*// *Выполнил: Студент группы БИН1801 Иванов И.И.*

*-->*//*Вызов и выполнение сценария Pr1\_3\_1.sce*

*-->*

-->clear // *Очистка окна Обозревателя данных*

-->clc // *Очистка Командного окна*

--> exec('Pr1\_3\_1.sce', 0);

Введите число n =4

Введите искомое число m =3

Введите кол-во элементов вектора 10

Введите начальное значение диапазона для вектора 2

Введите конечное значение диапазона для вектора 4

Исходный массив:

2. 3. 4. 3. 4. 3. 4. 4. 3. 4.

Индексы найденных элементов в новом массиве

3.

Среди последних 4 элементов исходного массива

кол-во чисел =3 равно 1

*-->*//*Конец сессии 1-го задания* ===========================================

***Пример выполнения задания 2***

1. ***Решить 2-ю задачу:*** Найти среднее арифметическое первого столбца матрицы и заменить этим значением элементы последней строки матрицы.
2. ***Решение 2-й задачи (сценарии и протокол сессии):***

//*Первый сценарий – функция obrab\_matr.sce для решения задачи 2*

//*Функция находит среднее арифметическое первого столбца матрицыA*

//*и заменяет этим значением элементы последней строки матрицы*

function [sred, A]=zamena(A)

sred=mean(A(:,1)); // *Среднее арифметическое первого столбца матрицы*

[Nr,Nc]=size(A); //*Определение числа строк и столбцов матрицы*

A(Nr,:)=sred;//*Замена элементов последней строки*

endfunction

//*Конец сценария obrab\_matr.sce* ==========================================

//*Второй сценарий Pr1\_3\_2.sce для решения 2-й задачи*

// *Реализует загрузку сценария obrab\_matr*.sce, *ввод исходных данных,*  
// *формирование матрицы A, вызов функции zamena и вывод результатов*

clear;// *Очистка окна Обозреватель данных*

clc; // *Очистка Командного окна*

//*Загрузка сценария обработки матрицы*

exec('*obrab\_matr*.sce');

N=input("Введите размер матрицы NxN ");

//*Генерация матрицы размером NxN и значениями от 0 до 100*

A=grand(N,N,"uin",0,10);

disp(A,"Исходная матрица: ");

[sred,A]= zamena(A); //*Вызов функции решения задачи*

//*Вывод результатов на экран*

mprintf('Среднее арифметическое первого столбца равно %f',sred);

disp(A,"Полученная матрица после замены : ");

// *Конец сценарияPr1\_3\_2.sce* ==============================================



-->// *Сессия командного окна:*

-->// *Практическое занятие 1.3 по теме*

-->// *«Матричные операции создания, доступа, извлечения*-->// *и модификации. Векторизация и индексирование»*

*-->*// *Вариант 0, задание 2*

*-->*// *Выполнил:Студент группы БИН1801 Иванов И.И.*

*-->*//*Вызов и выполнение сценарияPr1\_3\_2.sce*

--> exec('Pr1\_3\_2.sce');

Введите размер матрицы NxN 5

Исходная матрица:

10. 0. 1. 5. 4.

7. 3. 3. 10. 6.

4. 10. 10. 1. 8.

10. 10. 1. 8. 7.

2. 5. 0. 10. 3.

Среднее арифметическое первого столбца равно 6.600000

Полученная матрица после замены:

10. 0. 1. 5. 4.

7. 3. 3. 10. 6.

4. 10. 10. 1. 8.

10. 10. 1. 8. 7.

6.6 6.6 6.6 6.6 6.6

*-->*//*Конец сессии 2-го задания* ===========================================

## **1.3.5. Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение каждого сценария;
* номер задания и формулировка решаемой задачи и/или краткое перечисление действий, реализуемых в сценарии.

1. Протокол сессии ***Командного окна***, вначале которого должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **1.3.6. Контрольные вопросы по теме**

1. Способы создания векторов и матриц.
2. С использованием какой функции можно определить длину вектора?
3. С использованием каких функций Scilab можно определить в матрице число строк, число столбцов и общее количество элементов матрицы?
4. С какого числа начинается индексация вектора?
5. Какую матрицу создает операция **ones**?
6. Каким образом в Scilab могут быть объединены две матрицы?
7. Назначение и формат функции **cat**?
8. Каким образом создается вектор с постоянным шагом?
9. Можно ли создать матрицу, элементы строки которой изменяются с постоянным шагом?
10. Каким образом удалить из матрицы определенный столбец?
11. Что такое векторизация?
12. Стандартное индексирование векторов и матриц.
13. Что такое векторное индексирование матриц.
14. Что такое логическое индексирование матриц.
15. Что такое поэлементные действия (операции с точкой) и где они используются?
16. Каким образом можно транспонировать вектор или матрицу?
17. Требуется ли при работе с векторами и матрицами предварительное объявление их размера?
18. Какой символ используются для разделения элементов матрицы в строке, а какой для разделения ее строк?
19. Какие команды предназначены для заполнения матрицы случайными числами, распределенными по равномерному или нормальному закону распределения?
20. Формат команд выбора минимального и максимального значения элемента матрицы.

# **Практическое занятие по теме**

# **«Визуализация результатов вычислений»**

**Цель работы:** Изучение средств визуализации результатов вычислений: графические объекты и их свойства, средства построения графиков (двумерных и трехмерных) и различных геометрических фигур, средства построения графических интерфейсов пользователей.

## **Вопросы, подлежащие изучению**

1. Понятия графических объектов и их свойства.
2. Построение графиков функций одной переменной.
3. Построение графиков функций двух переменных переменной
4. Средства инструментальной панели графических окон.
5. Построение трехмерных изображений с использованием функций   
   Scilab: **mesh**, **plot3**, **surf** и **controur**.
6. Средства построения графических интерфейсов пользователей.

## **Общее задание**

1. ***Изучить материал учебника*** [1]*(п. 1.4),* [5] *.*
2. ***Выбрать вариант*** *индивидуального задания из* ***табл. 1.4-1****.*
3. ***Создать сценарий для работы*** *с функциями***f1(x)***и***f2(x)** *.*
4. ***Выполнить команды* clear** *и* **clc** *для очистки окна* ***Обозревателя переменных*** *и* ***Командного окна****.*
5. ***Задать диапазон*** *изменения аргумента функции***f1(x*)****,* ***вычислить*** *ее значения и* ***построить*** *ее график в графическом окне с номером 1 с использованием функции* **plot***.*
6. ***Описать функцию* f2(x)**.
7. ***Задать вектор*** *аргумента функции* **f2(x)***.Обратить внимание на совместимость диапазонов аргументов для функций* **f1(x)** *и* **f2(x)***.*
8. ***Построить*** *в том же окне с номером 1 график функции* **f2(x) *другим типом линии***. *Провести оформление графиков,* ***изменив толщину линий*** *и* ***величину шрифта****.*
9. ***Дополнить графики*** *необходимыми пояснениями:* ***заголовок***, ***имена осей***, ***координатная сетка*** *и* ***легенда****.*
10. ***Задать*** *в* ***Командном окне диапазоны*** *изменения значений* ***x*** *и* ***y*** *для вычисления значений и построения графиков функции* **f3(x,y)***.*
11. ***Описать функцию* f3(x,y)** *и* ***получить таблицу*** *ее значений.*
12. ***Получить графики функции* f3(x,y)** *с использованием* **mesh**, **plot3**, **surf** *и* **controur** *в разных графических окнах*.
13. ***Сохранить тексты сценариев*** *и* ***содержимое Командного окна*** *на внешнем носителе.*
14. ***Предоставить результаты*** *работы преподавателю, ответить на поставленные вопросы.*
15. ***Оформить отчет*** *по выполненной работе.*

## **Варианты индивидуальных заданий**

Таблица 1.4.3-1

|  |  |
| --- | --- |
| **№** | **Задания для задач 1 и 2** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 |  |
| 21 |  |
| 22 |  |
| 23 |  |
| 24 |  |
| 25 |  |
| 26 |  |
| 27 |  |
| 28 |  |
| 29 |  |
| 30 |  |

## **Пример выполнения задания**

***Пример выполнения задания 1(одномерные функции)***

1. **В соответствии с п.п. 3-9 общего задания выполнить задание:** Построить графики одномерных функций

****.

1. **Решение задания 1:** построение графиков функций от одной переменной с использованием сценария и протокола:

// *Сценарий Pr1\_4\_1.sce задания 1*

//*Определение функций f1 иf2*

// *и построение графиков двух одномерных функций*

clear; *//Очистка окна Обозревателя данных*

clc; *//Очистка Командного окна*

// *Диапазон изменения аргумента для функции f1(x)*

x=-0.5:0.3:6;

// *Вычисление вектора значений функции f1(x)*

f1=x.\*log(1./abs(x+1));

scf(1); // *Графическое окно 1*

plot(x,f1);// *График функции f1(x)*

// *Описание функции f2(x)*

function y=f2(x)

y=1+10\*sin(x)./x;

endfunction

// *Определение вектора значений аргумента для функции f2(x)*

t=[0.2:0.2:5];

// *Построение графика f2(x)и оформление линий и маркеров*

plot(t,f2(t),'LineStyle','-','Color','green',...

'Thickness',3,'Marker','o','MarkerEdgeColor','r',...

'MarkerFaceColor','k','MarkerSize',10);

// *Оформление линий и маркеров и новое построение графика f1(x)*

plot(x,f1,'LineStyle','--','Color','red',...

'Thickness',3,'Marker','s','MarkerEdgeColor','b',...

'MarkerFaceColor','y','MarkerSize',10);

// *Оформление подписей графиков, осей и сетки*

xtitle('Графики функций f1(x) и f2(x)','X','Y')

legend('Функция f1(x)','Функция f2(x)',1)

xgrid

//*Конец сценария Pr1\_4\_1.sce*==========================================



-->//*Сессия командного окна для выполнения задания 1:*

-->// *Практическое занятие 1.4 по теме*

*-->*// *«Визуализация результатов вычислений»*

*-->*// *Вариант 0, задание 1*

*-->*// *Выполнил: Студент группы БИН1801 Иванов И.И.*

*-->*// *Вызов сценария Pr1\_4\_1.sce*

*-->*// *и построение графиков функции одной переменной*

*-->*

-->exec('*Pr1\_4\_1.sce*');

### лр1_4_1 график1.jpg

***Пример выполнения задания 2 (функция от двух переменных)***

1. **В соответствии с п.п.10-12 общего задания выполнить задание:** Построить график функции f3(x,y)=x2+2y2.
2. **Решение задания 2:** построение графиков различного вида для функции от двух переменных (протокол сессии).

|  |
| --- |
| *-->*// *Сессия Командного окна задания 2:*  *-->*// *Практическое занятие 1.4 по теме*  *-->*// *«Визуализация результатов вычислений»*  *-->*// *Вариант 0, задание 2*  *-->*// *Выполнил: Студент группы БИН1801 Иванов И.И.*  *-->*// *Построение различных видов графиков функции двух переменных*  *-->*  -->// zM = xM.^2 + 2 \* yM.^2  -->  -->// *Формирование двух матриц* xM*и* yM  --> [xM, yM] = meshgrid(-4 : 4, -4 : 4)  xM =  0. 1. 2. 3.  0. 1. 2. 3.  0. 1. 2. 3.  0. 1. 2. 3.  -->  yM =  -3. -3. -3. -3.  -2. -2. -2. -2.  -1. -1. -1. -1.  0. 0. 0. 0.  -->  --> // *Формирование матрицы* zM  --> deff('k = f(x, y)', 'k = x.^2 + 2 \* y.^2');  --> zM = f(xM, yM)  zM =  48. 41. 36. 33. 32. 33. 36. 41. 48.  34. 27. 22. 19. 18. 19. 22. 27. 34.  24. 17. 12. 9. 8. 9. 12. 17. 24.  18. 11. 6. 3. 2. 3. 6. 11. 18.  16. 9. 4. 1. 0. 1. 4. 9. 16.  18. 11. 6. 3. 2. 3. 6. 11. 18.  24. 17. 12. 9. 8. 9. 12. 17. 24.  34. 27. 22. 19. 18. 19. 22. 27. 34.  48. 41. 36. 33. 32. 33. 36. 41. 48.  -->  --> // *Формирование векторов* xV*и* yV  --> x2 = -4 : 4; y2 = -4 : 4;  -->  --> // *Построение двумерных графиков*  --> scf(1); mesh(xM, yM, zM) // *Сетчатый график (окно 1)*  *-->* scf(2); plot3d(xV, yV, zM***)*** // *График точек, соединенных отрезками*  --> // *прямых (окно 2)*  --> scf(3); surf(xM, yM, zM) // *График сплошной поверхности (окно 3*)  --> scf(4); contour(xV, yV, zM, 7) // *График контурных линий (окно 4)* |

|  |  |
| --- | --- |
|  |  |
|  |  |

## **Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* номер задания и формулировка решаемой задачи и/или краткое перечисление действий, реализуемых в сценарии.

1. Протокол сессии ***Командного окна***, вначале которого должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **Контрольные вопросы по теме**

1. Каково назначение функции **scf(n)**?
2. Какой номер у первого графического окна?
3. Можно ли создать графическое окно с определенным номером?
4. Каким образом можно закрыть графическое окно?
5. Назначение элементов меню графического окна: ***Файл***, ***Инструменты***, ***Правка***, ***Справка***.
6. Каким образом можно закрыть графическое окно?
7. Назначение функции **xdel(id*)***.
8. Назначение команды **winsid**.
9. Как выполнить настройку элементов графика (оси, толщина линий, шрифты и т.д.) с использованием элементов меню графического окна?
10. Можно ли с использованием одного оператора **plot**построить несколько графиков?
11. Можно ли в одном графическом окне построить несколько графиков одного типа?
12. Можно ли в одном графическом окне построить несколько графиков разных типов?
13. Каким образом в операторе указывается цвет и символ отображения графика?
14. Какой пояснительной информацией может быть снабжен график, построенный в графическом окне?
15. Назначение и формат функции **xtitle**?
16. Для чего используется функция **legend**?
17. Каково назначение функции **mtlb\_hold**?
18. Каково назначение функции **meshgrid** при построении трехмерных изображений?
19. Какие типы графиков позволяет построить Scilab?
20. Какие типы графиков позволяют строить встроенные функции: **plot**, **contour**, **surf** и **plot3**?
21. Что такое контурные линии, и на каком графике их можно увидеть?

# **Практическое занятие по теме**

# **«Средства программирования в Scilab»**

**Цель работы:** Изучение средств программирования системы Scilab и получение практических навыков разработки, написания и отладки программных приложений в Scilab.

## **Вопросы, подлежащие изучению**

1. Виды **sce-**файлов.
2. Особенности сценариев и функций.
3. Структура сценариев и функций.
4. Запуск на выполнение сценария из текстового редактора.
5. Запуск на выполнение сценария из ***Командного окна***
6. Обращения к файлам и функциям.
7. Средства языка программирования в системе Scilab.
8. Основные операторы языка Scilab, назначение и форматы**.**
9. Создание и использование библиотеки функций.
10. Графический интерфейс приложений пользователя.

## **1.5.2.1 Общее задание №1**

## **«Программирование алгоритмов**

## **итеративных циклических структур»**

1. ***Изучить материал учебника*** [1](п. 1.5).
2. ***Выбрать вариант задания*** *из таблицы 1.5.2-1 по заданию преподавателя.*
3. ***Провести формализацию*** *поставленной задачи*.
4. ***Составить схему алгоритма*** *решения поставленной задачи средствами* ***Visio*** [8].
5. ***Написать программный код сценария-функции*** *с параметрами в соответствии со схемой алгоритма без использования глобальных переменных, предусмотрев страховку от возможного «зацикливания».*
6. ***Написать программный код*** *сценарий выполнения задания, содержащий ввод исходных данных, вызов функции п.4 и вывод результатов.*
7. ***По требованию преподавателя разработать графический интерфейс*** *пользователя. В интерфейсе предусмотреть отображение на форме промежуточных результатов каждой итерации. Написать программный код сценария создания интерфейса пользователя.*
8. ***Выполнить*** *созданное приложение и* ***получить*** *решение*.
9. ***Обосновать правильность полученных результатов*** *на заранее разработанных тестах.*
10. ***Представить отчет*** *преподавателю****.***

### 1.5.2.2 Варианты индивидуальных заданий

Таблица 1.5.2-1

|  |  |
| --- | --- |
| **№** | **Задание** |
| **1)** | Вычислить с точностью ε = 0.00001 константу Эйлера (основание натурального логарифма), воспользовавшись разложением в ряд:    Сравнить результат со значением, полученным с помощью  соответствующей встроенной функции. |
| **2)** | Вычислить и вывести те члены последовательности,  значения которых больше ε = 0.001при x = 0.2. |
| **3)** | Вычислить arctg(x) с точностью ε = 0.0001, воспользовавшись разложением в ряд:  Сравнить результат со значением, полученным с помощью соответствующей встроенной функции при x=1.5. |
| **4)** | Вычислить с точностью ε = 0.00001 значение функции  при **x = 2**, воспользовавшись рекуррентной формулой:  .  Сравнить результат со значением, полученным с помощью соответствующей встроенной функции. |
| **5)** | Вычислить константу с точностью до ε = 0.00001, воспользовавшись разложением в ряд:    Сравнить результат со значением, полученным с помощью соответствующей встроенной функции. |
| **6)** | Вычислить с точностью ε = 0.00001 значение функции  при **x = 2**, воспользовавшись формулой:    Сравнить результат со значением, полученным с помощью соответствующей встроенной функции |
| **7)** | Вычислить **sin 0.5** с точностью ε = 0.0001, воспользовавшись  разложением в ряд:    Сравнить результат со значением, полученным с помощью соответствующей встроенной функции. |
| **8)** | Вычислить с точностью ε = 0.00001, воспользовавшись разложением в ряд:  Сравнить результат со значением, полученным с помощью соответствующей встроенной функции. |
| **9** | Вычислить cos 0.6 с точностью ε = 0.00001, воспользовавшись разложением в ряд:  Сравнить результат со значением, полученным с помощью  соответствующей встроенной функции. |
| **10** | Вычислить с точностью ε = 0.0001 корень уравнения **x - cos(x)=0**, воспользовавшись формулой:.  Проверить правильность решения подстановкой найденного корня в уравнение. |
| **11** | Вычислить и вывести те члены последовательности,    значения которых по модулю больше ε = 0.001 при x = 0.5. |
| **12** | Вычислить  при |x|<1 с точностью до ε = 0.0001, воспользовавшись разложением в ряд:    Сравнить результат со значением, полученным с помощью соответствующей встроенной функции. |
| **13** | Вычислить корень уравнения с точностью ε=0.0001, воспользовавшись итерационной формулой    Проверить правильность решения подстановкой найденного корня в уравнение. |
| **14** | Вычислить значение с точностью ε = 0.00001, воспользовавшись представлением в виде в виде цепной дроби:    Значение дроби равно пределу числовой последовательности, члены которой вычисляются по рекуррентной формуле до достижения заданной точности  .  Сравнить результат со значением, полученным с помощью соответствующей встроенной функции. |
| **15** | Вычислить и вывести те члены последовательности,    значения которых по модулю больше ε = 0.001 при x = 0.3. |
| **16** | Вычислить**ln(x)** с точностью ε = 0.0001, воспользовавшись разложением в ряд:  Сравнить результат со значением, полученным с помощью  соответствующей встроенной функции при x=1.5. |
| **17** | Вычислить sh 0.3 с точностью до ε= 0.00005, воспользовавшись разложением в ряд:  Сравнить результат со значением, полученным с помощью встроенной функции для вычисления **ex**, используя соотношение: |
| **18** | Вычислить корень уравнения **x-0.5(sinx2-1)=0** с точностью ε = 0.001, воспользовавшись итерационной формулой:  Проверить правильность решения подстановкой найденного корня в уравнение. |
| **19** | Вычислить ln(2) с точностью ε = 0.001, воспользовавшись представлением в виде ряда:  Сравнить результат со значением, полученным с помощью соответствующей встроенной функции. |
| **20** | Вычислить с точностью ε = 0.00001 корень уравнения  воспользовавшись итерационной формулой  Проверить правильность решения подстановкой. |
| **21** | Вычислить ch 0.7 с точностью до ε = 0.00005, воспользовавшись разложением в ряд:  Сравнить результат со значением, полученным с помощью встроенной функции, используя соотношение: |
| **22** | Вычислить приближенное значение бесконечной суммы с точностью ε=0,0001 (справа от суммы дается выражение для проверки полученного результата):  (для |x|<1 сумма равна  ) |
| **23** | Вычислить  при |x|>1 с точностью до ε = 0.0001, воспользовавшись разложением в ряд:  Сравнить результат со значением, полученным с помощью соответствующей встроенной функции. |
| **24** | Вычислить ln(x+1)с точностью ε = 0.0001, воспользовавшись разложением в ряд:  Сравнить результат со значением, полученным с помощью  соответствующей встроенной функции при x=0.5. |
| **25** | Вычислить и вывести те члены последовательности,  значения, которых больше ε = 0.01**,** при x = 0.6. |
| **26** | Найти наименьшее целое положительное **n**, при котором: |
| **27** | Пусть  Дано действительное число **ε**>0. Найти первый член , для которого выполнено условие . |
| **28** | Вычислить приближенное значение бесконечной суммы с точностью ε=0,0001 Сравнить полученный результат c точным значением, равным π2/6-1:. |
| **29** | Вычислить приближенное значение бесконечной суммы с точностью ε=0,0001. Сравнить полученный результат c точным значением, равным π/4: |
| **30** | Вычислить приближенное значение бесконечной суммы с точностью ε=0,0001 Сравнить полученный результат c точным значением, равным 3/4: |

### 1.5.2.3 Пример выполнения задания

1. ***Задание на разработку приложения***

Создать приложение для вычисления и отображения на разработанной форме тех членов последовательности

,

значения которых при x=0.5по модулю больше ε=0.0001.

1. ***Формализация и уточнение задания***

Для решения поставленной задачи необходимо вывести рекуррентную формулу вычисления члена последовательности.

Очевидно, что выражение для n-го члена заданной последовательности имеет вид:

.

Тогда формула для (n+1) члена последовательности имеет вид:



Имея в виду, что (n+1)!=n! ∙ (n+1), получим



В результате получаем следующую рекуррентную формулу:

 -начальный член последовательности при n=1.

1. ***Разработка схемы алгоритма функции Pos*** (рис.1.5.4-1)



|  |
| --- |
|  |

Рис. 1.5.4-1 Схема алгоритма решения задачи

1. ***Программный код*** решения задачи

*// Сценарий-функция Pr1\_5\_1\_fun.sce*

*// Практическое занятие 1.5по теме:*

*// «Программирование алгоритмов*

*// итеративных циклических структур»*

// *Вариант 0*

*// Входные параметры: аргумент х и точность eps*

*// Выходные параметры: вектор y содержит значения*

*// членов ряда, вектор k содержит номера выполненных итераций*

function [**y**, **k**]=Pos(**x**, **eps**)

a=**x**-1; n=1;

while abs(a)>**eps**&& n<100

**k**(n)=n;

**y**(n)=a;

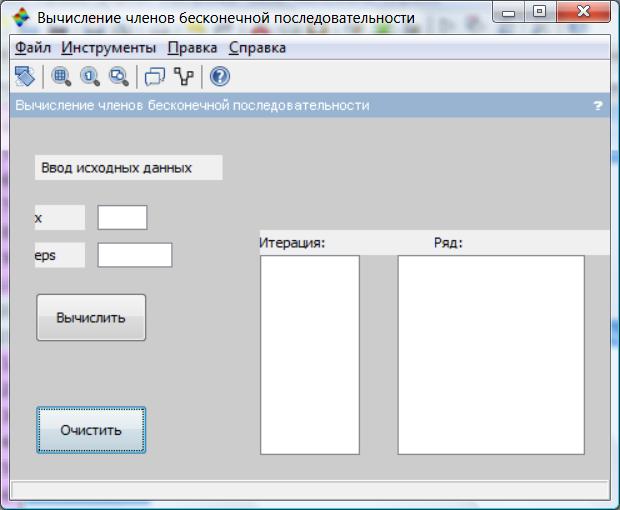
a=a.\*(**x**-1)./(n+1);

n = n + 1;

end

endfunction

1. ***Графический интерфейс пользователя:***



1. ***Программный код*** сценария для создания графического интерфейса и сценария для выполнения задания, где последний содержит ввод исходных данных, вызов функции п.4 и вывод результатов.

*// Сценарий Pr1\_5\_1GUI.sce*

*// Создание интерфейса пользователя*

*// Создание графического окна*

f1=figure('position',[20,40,480,290],'figure\_name',...

'Вычисление членов бесконечной последовательности');

*// Определение текстовых полей для ввода исходных данных*

*//Текстовое поле заголовка Ввод исходных данных*

text\_vvod=uicontrol(f1,'style','text','string','',…

'position',[20,240,150,20],…

'String',' Ввод исходных данных');

*// Поле ввода x*

edit\_x=uicontrol(f1,'style','edit','string','0.5',...

'position',[70, 200, 40, 20]);

textx=uicontrol(f1,'style','text','string','x',...

'position',[20,200,40,20]);

*// Поле ввода eps*

edit\_eps=uicontrol(f1,'style','edit','string','0.0001',...

'position',[70, 170, 60, 20]);

text\_eps=uicontrol(f1,'style','text','string','eps',...

'position',[20,170,40,20]);

*// Определение текстовых полей для вывода результатов итераций*

list1 = uicontrol(f1,'style','listbox','position',[200,20,80,160]);

list2 = uicontrol(f1,'style','listbox','position', [310, 20, 150, 160]);

*// Заголовки полей вывода результатов*

TextRez1 = uicontrol(f1, 'style', 'text', 'string', '',

'position',[200,180,150,20],'string', 'Итерация:');

TextRez2 = uicontrol(f1, 'style', 'text', 'string', '',…

'position',[340,180,150,20],'string', 'Ряд:');

*// Определение кнопки Вычислить*

bt1 = uicontrol(f1, 'style', 'pushbutton', 'string', 'Вычислить',…

'position', [20,110,90,40], 'CallBack','YY');

*// Определение кнопки Очистить*

bt2 = uicontrol(f1, 'style', 'pushbutton', 'string', 'Очистить',…

'position', [20,20,90,40], 'CallBack','C');

// *Конец сценария Pr1\_5\_1GUI.sce =========================================*

*// Сценарий выполнения задания Pr1\_5\_1.sce*

*// Подключение сценария создания интерфейса*

exec('Pr1\_5\_1GUI.sce');

*// Подключение сценария-функции решения задачи*

exec('Pr1\_5\_1fun.sce');

*// Функция YY() содержит инструкции ввода исходных данных,*

*// вызов функции решения задачи(вычисления членов ряда)*

*// из сценария Pr1\_5\_1\_fun.sce и инструкции вывода результатов*

*// Выполнение функции YY() инициируется нажатием кнопки "Выполнить"*

function YY()

sx1=get(edit\_x,'string');*// считывание x и преобразование в строку*

x=strtod(sx1);*// преобразование sx1 к типу double*

seps=get(edit\_eps,'string');*//считывание точности в строку*

e=strtod(seps);*// преобразование seps к типу double*

[y,k]=Pos(x,e);*// вызов функции решения задачи*

*// вывод результатов в соответствующие текстовые поля*

set(list1,'string',sprintf("%d\n",k));

set(list2,'string',sprintf("%1.5f\n",y));

endfunction

*// Функция С() очистки текстовых полей*

*// Выполнение функции инициируется нажатием кнопки "Очистить"*

function C()

set(list1,'string',"");

set(list2,'string',"");

set(edit\_x,'string',"");

set(edit\_eps,'string',"");

endfunction

//*Конец сценарияPr1\_5\_1.sce*==============================================

-->// *Сессия командного окна задания 1*:

*-->*// *Практическое занятие 1.5 по теме:*

*-->*// «*Программирование алгоритмов итеративных циклических структур»*

*-->*// *Вариант 0, задание 1*

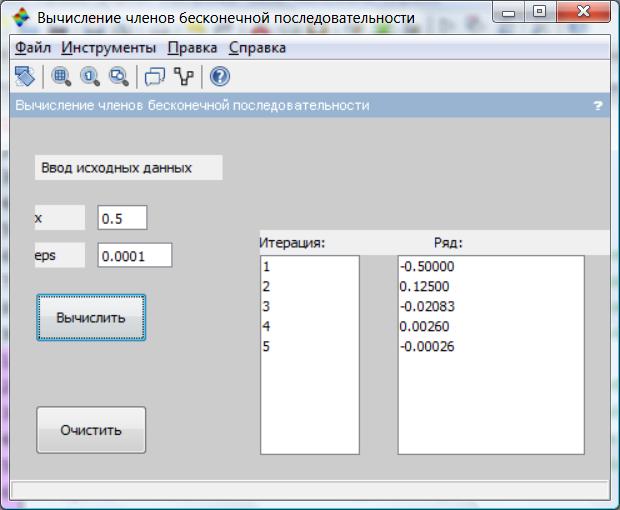
*-->*// *Выполнил: Студент группы БИН1801 Иванов И.И.*

-->

-->exec('*Pr1\_5\_1.sce*');

*-->*//*Конец сессии* ========================================================

1. ***Результат работы приложения***

****

1. ***Доказательство правильности работы приложения***

Все выведенные значения последовательности больше заданной точности

**eps=0.0001***.*

## **Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Результаты формализации задания
5. Графический интерфейс пользователя (по требованию преподавателя).
6. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* номер задания и формулировка решаемой задачи и/или краткое перечисление действий, реализуемых в сценарии.

1. Протокол сессии ***Командного окна***, вначале которого должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **1.5.3.1 Общее задание №2**.

## **«Программирование алгоритмов**

## **регулярных циклических структур»**

1. ***Изучить материал учебника*** [1](п. 1.5).
2. ***Выбрать индивидуальное задание*** *из* ***табл. 1.5.3-1****.*
3. *В соответствии с индивидуальным заданием* ***создать 1-й сценарий с функцией решения задачи.*** *Эта функция может быть разбита на отдельные вспомогательные функции для реализации базовых алгоритмов, например, вычисления конечных сумм, разветвлений, поиска минимума и максимума в последовательности данных и т.п.*
4. ***Написать программный код 2-го*** *сценария выполнения задания, который должен содержать ввод исходных данных, вызов функции 1-го сценария и вывод результатов.*
5. ***По требованию преподавателя разработать графический интерфейс*** *пользователя. В интерфейсе предусмотреть отображение на форме исходных данных и результатов*. *Написать программный код* *сценария создания интерфейса пользователя.*
6. ***Выполнить*** *созданное приложение и* ***получить*** *решение.*
7. ***Оформить отчет*** *по выполненной работе****.***

## **Варианты индивидуальных заданий**

Таблица. 1.5.3-1

|  |  |
| --- | --- |
| **№** | **Задание** |
| 1 | Ввести натуральное число ***n****и вектор действительных чисел* ***y1,y2,…yn***Найти: ***max****(****|z1|,|z2|,…|zn|****),* где |
| 2 | Вычислить  где |
| 3 | Ввести натуральное число n и массив ***a1,a2,…an***. Сформировать массив ***w1,w2,…,wn*** по правилу: ***wi= ai2***, если ***ai*** кратно 3; ***wi= ai*** /3, если ***ai*** кратно 5; а в остальных случаях ***wi=Cos(ai)***. |
| 4 | Ввести натуральное число n и вектор действительных чисел. Вычислить произведение, где |
| 5 | Ввести натуральное число n и действительное число ***х***. Вычислить и вывести на экран искомую сумму и каждое слагаемое суммы: |
| 6 | Ввести натуральное число ***n***. Найти наибольшее среди значений , где ***k=1, 2,…,n***, а также сумму всех полученных значений. |
| 7 | Ввести натуральное число n и вычислить сумму значений , где (i=1,2,…n). Вывести индексы и полученные значения в виде таблицы. Определить минимальное значение элемента массива *а*. |
| 8 | Ввести натуральное число n и вектор действительных чисел. Определить каких положительных или отрицательных чисел в векторе больше, а также наибольшее из отрицательных и наименьшее из положительных значение . |
| 9 | Ввести матрицу ***B(5,7)*** и сформировать из первых наибольших элементов строк массива ***B*** вектор ***С(5)***. Вывести его элементы в строку и столбец. |
| 10 | Сформировать вектор по правилу:  , где k=2,3,…7, если  Найти сумму квадратов тех чисел, модуль которых не превосходят 2. |
| 11 | Ввести натуральное число n и вектор действительных чисел. Найти количество двух соседних положительных чисел и двух соседних чисел разного знака. |
| 12 | Ввести квадратную матрицу ***А(4,4)*** и, сформировав из максимальных элементов ее столбцов вектор ***X***, и вывести его элементы на экран в прямой и обратной последовательности. |
| 13 | Ввести вектор целых чисел . Преобразовать его таким образом, чтобы сначала располагались нули, а затем все остальные элементы. Определить сумму и количество элементов, значения которых кратны 5. |
| 14 | Ввести вектор вещественных чисел . Создать из его элементов массив ***x***, каждый элемент которого максимальный из 3-х элементов, идущих подряд в массиве ***z***. |
| 15 | Сформировать матрицу ***А(4,4)*** по правилу:    Найти и вывести значения и индексы двух одинаковых элементов. Если таковых не окажется, вывести сообщение. |
| 16 | Сформировать матрицу ***D(3,2)*** по правилу:    Создать и вывести на экран вектор, состоящий из отрицательных элементов полученной матрицы. |
| 17 | Задать натуральное число ***n***. Посчитать, какая из матриц размером **n**на **n** содержит большее количество положительных элементов, если ее элементы формируются по правилу:    Соответствующее сообщение и сформированные матрицы вывести на экран. |
| 18 | Ввести квадратную матриц ***А(4,4)*** из вещественных чисел. Найти сумму значений наибольших элементов ее строк. Сформировать и вывести на экран новую матрицу ***В(4,4)***, каждый элемент которой получен путем умножения его значения на найденную сумму. |
| 19 | Ввести матрицу вещественных чисел ***А(4,7)***. Сформировать и вывести на экран вектор С(4), элементами которого являются:   * наибольший из элементов в 1-й строке; * наименьший из элементов во 2-й строке; * среднее арифметическое элементов 3-й строки; * сумма элементов 4-й строки. |
| 20 | Ввести натуральное число n и матрицу вещественных чисел ***С(n,n)***. Найти среднее арифметическое наибольшего и наименьшего значений ее элементов, и, заменив этим значением диагональные элементы матрицы ***С***, вывести полученную матрицу на экран. |
| 21 | Ввести натуральные числа ***k1***, ***k2*** и действительную матрицу размера ***8х4***. Поменять в матрице местами элементы ***k1*** и ***k2*** строк и вывести на экран модифицированную матрицу. |
| 22 | Ввести натуральные числа ***m,n*** и матрицу целых чисел ***С(m,n)***. Сформировать и вывести на экран вектор, состоящий из сумм четных элементов каждого столбца матрицы ***С(m,n)***. |
| 23 | Ввести вектора действительных чисел ***x(5)***,***y(6)***,***z(7)***. Вывести значение величины t, вычисляемой по следующему алгоритму: |
| 24 | Ввести натуральное число ***n*** и массив . Получить и вывести на экран матрицу ***с(2,n),*** первая строка которой – массив ***b***, упорядоченный по возрастанию, а вторая – массив ***b***, упорядоченный по убыванию. |
| 25 | Ввести вектор действительных чисел ***x(10)***. Получить из него другой массив p(10), элементы которого упорядочены по возрастанию. |
| 26 | Ввести матрицу вещественных чисел ***А(3,4)***. Заменить элементы строки матрицы с максимальной суммой значений элементов – единицами, с минимальной - 2, а остальные элементы матрицы положить равными нулю. |
| 27 | Сформировать матрицу ***А(4,4)*** по правилу Удалить из него столбцы, содержащие элементы, меньшие 10. Вывести на экран сформированную и модифицированную матрицы. |
| 28 | Сформировать матрицу ***В(9,3)*** по правилу  Определить наименьший элемент в каждой строке матрицы и записать его в соответствующий элемент вектора ***С***. Вывести на экран сформированную матрицу и полученный вектор ***С***. |
| 29 | Ввести матрицу вещественных чисел ***А(3,4)***, все элементы которой различны. Найти в каждой строке матрицы наибольшее и наименьшее значение, и записать в соответствующий элемент вектора ***С(3)*** сумму значений индексов столбцов, в которых они расположены. Вывести на экран полученный вектор ***С***. |
| 30 | Ввести матрицу вещественных чисел ***А(4,4)***. Сформировать и вывести на экран:   * ***B(4)*** – вектор полученный из элементов главной диагонали; * ***C(4)*** - вектор полученный из элементов побочной диагонали |

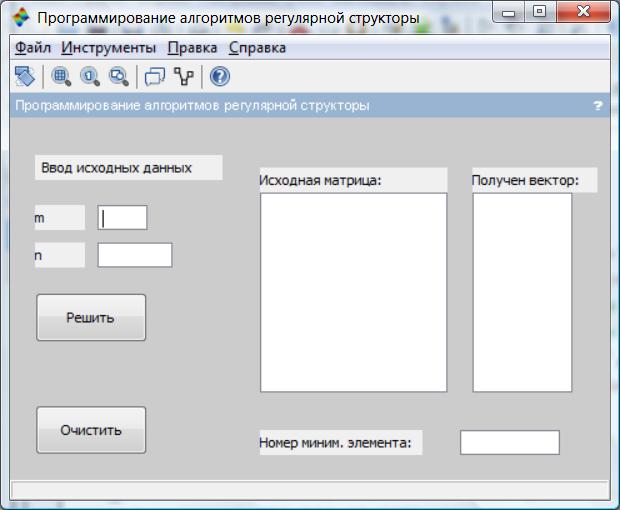
## **Пример выполнения задания**

1. ***Задание:*** Ввести натуральные числа ***m,n*** и сформировать матрицу целых чисел ***С(m,n)***. Найти среднее арифметическое каждой строки матрицы и сформировать из этих значений вектор. Найти в полученном векторе номер минимального элемента
2. ***Программный код*** 1-го сценария:

|  |
| --- |
| *// Сценарий-функция zadacha1\_5\_2.sce решения задачи обработки матрицы*  *// Найти среднее арифметическое каждой строки матрицы*  *// сформировать из этих значений вектор,*  *// и найти в векторе номер минимального элемента*  *// Входные параметры: матрица A, число строк m и число столбцов n*  *// Выходные параметры: вектор B значений среднего арифметического каждой  // строки матрицы, nom - номер минимального элемента*  function [B, nom]=zadacha1\_5\_2(A, m, n)  *// вычисление среднего арифметического каждой строки*  *//и формирование вектора*  for i=1:m  s=0;  for j=1:n  s=s+A(i,j);  end;  B(i)=s/n;  еnd  *//Определение в векторе минимального элемента и номера*  Minim=B(1); nom=1;  for i =2:m  if B(i)<minim  minim=B(i);  nom=i;  end  end  endfunction  *// Конец сценария zadacha1\_5\_2.sce =================================* |

1. ***Графический интерфейс пользователя***

Форма графического интерфейса пользователя приведена на рисунке:



1. ***Программный код*** создания графического интерфейса и выполнения задания, который содержит ввод исходных данных, вызов функции решения задачи и вывод результатов:

*// Сценарий Pr1\_5\_2\_GUI.sce задания 2*

*// Создание графического интерфейса пользователя*

*// Создание графического окна*

f1=figure('position',[20,40,480,290],…

'figure\_name','Программирование алгоритмов регулярной структуры');

*// Определение текстовых полей для ввода исходных данных*

*// Текстовое поле заголовка Ввод исходных данных*

text\_vvod=uicontrol(f1,'style','text','string','',…

'position', [20,240,150,20],'String',' Ввод исходных данных');

*// Поле ввода m*

edit\_m=uicontrol(f1,'style','edit','string','6',...

'position',[70, 200, 40, 20]);

text\_m=uicontrol(f1,'style','text','string','m',...

'position',[20,200,40,20]);

*// Поле ввода n*

edit\_n=uicontrol(f1,'style','edit','string','4',...

'position',[70, 170, 60, 20]);

text\_n=uicontrol(f1,'style','text','string','n',...

'position',[20,170,40,20]);

*// Определение текстовых полей для вывода результатов итераций*

list1=uicontrol(f1,'style','listbox','position',[200,70,150,160]);

list2 = uicontrol(f1,'style','listbox','position', [370, 70, 80, 160]);

editRez = uicontrol(f1,'style','edit','position', [360, 20, 80, 20]);

*// Заголовки полей вывода результатов*

TextRez1 = uicontrol(f1, 'style', 'text', 'string', '',…

'position', [200,230,150,20],…

'string', 'Исходная матрица:');

TextRez2 = uicontrol(f1, 'style', 'text', 'string', '',…

'position', [370,230,100,20],…

'string', 'Получен вектор:');

TextRez3 = uicontrol(f1,'style','text', 'string', '',…

'position', [200,20,130,20],…

'string', 'Номер минимального элемента:');

*// Определение кнопки Решить*

bt1 = uicontrol(f1, 'style', 'pushbutton', 'string', 'Решить',...

'position', [20,110,90,40],'CallBack','YY');

*// Определение кнопки Очистить*

bt2 = uicontrol(f1, 'style', 'pushbutton', 'string', 'Очистить',...

'position', [20,20,90,40],'CallBack','C');

*// Конец сценария Pr1\_5\_2\_GUI.sce ==========================================*

*// Сценарий выполнения задания Pr1\_5\_2.sce*

*// Подключение сценария создания интерфейса*

exec(' *Pr1\_5\_2\_GUI*.sce');

*// Подключение сценария-функции решения задачи*

exec('zadacha1\_5\_2.sce');

*// Функция создания строковой матрицы для вывода числовой матрицы*

*// Входной параметр - выводимая матрица*

*// Выходной параметр - матрица строк*

function vA=vivodMatr(A)

[m, n] = size(A);

vA=['']; // *Создание пустой строковой матрицы*

*//Изменение ee размера под размер выводимой матрицы*

vA=resize\_matrix(vA,m,n);

for i =1:m

for j=1:n

vA(i)=vA(i)+sprintf("%7s", string(A(i,j)));

end

end

endfunction

*// Функция YY содержит инструкции ввода исходных данных,*

*// генерации матрицы случайными числами, и ее вывод*

*// вызов функции решения задачи*

*// из сценария zadacha1\_5\_2.sce и инструкции вывода результата*

*// Выполнение функции YY инициируется нажатием кнопки "Выполнить"*

function YY()

*// считывание числа строк матрицы и преобразование в число*

m =eval(get(edit\_m,'string')) ;

*// считывание числа столбцов матрицы и преобразование в число*

n = eval(get(edit\_n,'string'));

*//Генерация матрицы размером m×n и значениями от 0 до 10*

A=grand(m,n,"uin",0,10);

*//Вызов функции создания строковой матрицы из исходной числовой*

vA=vivodMatr(A);

*// Вывод в соответствующее текстовое поле list1*

set(list1, 'string', vA);

*//Вызов функции решения задачи*

[B,nom]=zadacha1\_5\_2(A,m,n);

*// Вывод полученного вектора в соотв. текс. поле list2*

set(list2, 'string', sprintf("%1.5f\n\n\n", B));

*// Вывод найденного номера в соотв. текс. поле editRez*

set(editRez, 'string', sprintf("%d", nom));

endfunction

*// Функция С очистки текстовых полей*

*// Выполнение функции выполняется нажатием кнопки "Очистить"*

function C()

set(edit\_m, 'string', "");

set(edit\_n, 'string', "");

set(editRez, 'string', "");

set(list1, 'string', "");

set(list2, 'string', "");

endfunction

//Конец сценария *Pr1\_5\_2.sce*=============================================

-->// Сессия командного окна задания 2

*-->*// *Практическое занятие 1.5 по теме:*

*-->*//*«Программирование алгоритмов регулярной структуры»*

*-->*// *Вариант 0, задание 2*

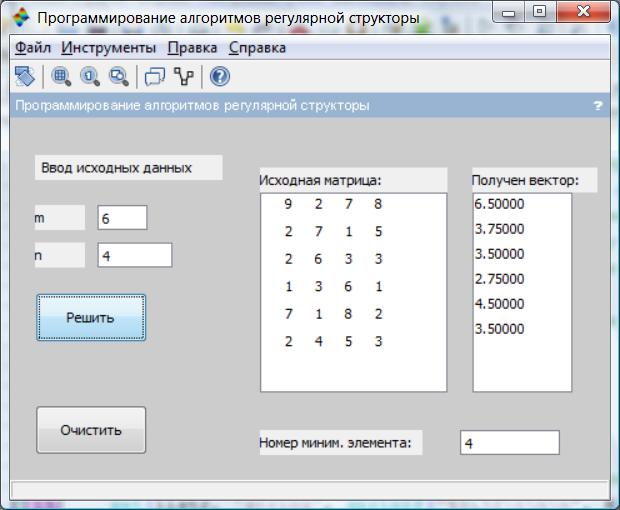
*-->*// *Выполнил: Студент группы БИН1801 Иванов И.И.*

*-->*// Загрузка сценария '*Pr1\_5\_2.sce*'

-->exec('*Pr1\_5\_2.sce*');

*-->*// Конец сессии ========================================================

1. ***Результат выполнения приложения***

****

### 1.5.3.4 Содержание отчета

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя (по требованию преподавателя).
5. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* номер задания и формулировка решаемой задачи и/или краткое перечисление действий, реализуемых в сценарии.

1. Протокол сессии ***Командного окна***, вначале которого должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **1.5.4.1 Общее задание №3**

## **«Программирование алгоритмов обработки матриц»**

1. ***Изучить материал учебника*** [1] (п. 1.5)*.*
2. ***Выбрать индивидуальное задание из табл. 1.5.4-1.***
3. *В соответствии с индивидуальным заданием* ***создать 1-й сценарий с необходимыми функциями*** *для реализации базовых алгоритмов (например, вычисления конечных сумм, разветвлений, поиска минимума и максимума в последовательности данных и т.п.) и с* ***функцией решения задачи***
4. ***Написать программный код*** *2-го сценария, который содержит ввод исходных данных, вызов функции из 1-го сценария решения задачи и вывод результатов.*
5. ***По требованию преподавателя разработать графический интерфейс*** *пользователя. В интерфейсе предусмотреть отображение на форме исходных данных и результатов.* ***Написать*** *сценарий создания интерфейса пользователя.*
6. ***Получить*** *решение.*
7. ***Оформить отчет*** *по выполненной работе****.***

## **Варианты индивидуальных заданий**

Таблица 1.5.4-1

|  |  |
| --- | --- |
| **№** | **Задание** |
| 1 | Составить программу, которая получает матрицу **Х** размером **m\*n** и вектор **A**, размером **m**. Строку с номером **i** назовем отмеченной, если соответствующий **i** элемент вектора **A** больше **0**, и неотмеченной – в противном случае. Программа формирует новую матрицу по следующему правилу: все отрицательные элементы в отмеченных строках исходной матрицы в формируемой матрице программа заменяет отрицательной единицей, положительные заменяет положительной единицей, а нулевые остаются без изменений. Неотмеченные строки исходной матрицы копируются в новую матрицу в обратном порядке. |
| 2 | Составить программу, которая получает матрицу **Х** размером **m\*n** и если в исходной матрице есть строки и столбцы, все элементы которой равны **0**, то программа уплотняет исходную матрицу влево и вверх, а затем формирует новую матрицу, поменяв местами строки и столбцы. Программа выдает на экран исходную и сформированную матрицу, сообщив, уплотнялась ли исходная матрица. |
| 3 | Составить программу, которая получает матрицу **Х** размером **m\*n** и вектор **A**, размером **к**. Программа формирует новую матрицу, в которую включены только те строки, которые содержат хотя бы одно число из вектора **А**. |
| 4 | Составить программу, которая получает матрицу **В** размером **m\*n** и формирует новую матрицу, удалив из исходной столбец и строку, на пересечении которых находится наибольший по модулю элемент. |
| 5 | Составить программу, которая получает матрицу **Х** размером **m\*n** и вектор **А**, размером m. Каждому элементу вектора поставлена в соответствие строка с тем же номером. Программа сортирует вектор А в порядке возрастания и формирует новую матрицу, расположив строки исходной матрицы в том же порядке, что и элементы вектора после сортировки. |
| 6 | Составить программу, которая получает матрицу **Х** размером **m\*n**. Программа в каждой строке определяет наименьший элемент и его индексы и формирует из этих чисел матрицу, расположив в первом ее столбце наименьшие элементы строк, а во втором и третьем столбцах их индексы. |
| 7 | Составить программу, которая получает матрицу **В** размером **m\*n** и формирует новую матрицу, включив в нее только те строки, элементы которых не образуют арифметической прогрессии. Если таких строк не найдено, то новая матрица не формируется и об этом выдается сообщение. |
| 8 | Составить программу, которая получает матрицу **А** размером **m\*n** и формирует из нее матрицу В, поменяв местами строки и столбцы. Для каждой из матриц функция вычисляет числа, где **m**- количество строк, а **n**- количество столбцов, и выдает сообщение об их равенстве или неравенстве. |
| 9 | Составить программу, которая получает матрицу **Х** размером **m\*n** и число **К**. Если программа находит число **К** среди элементов исходной матрицы, то она удаляет из исходной матрицы столбцы с элементом, равным **К**, в противном случае программа формирует новую матрицу, записав в нее только четные строки. Программа выдает на экран исходную матрицу и новую, если ее удалось создать. |
| 10 | Составить программу, которая, получив матрицу, размером **m\*n**, создает новую матрицу. В новой матрице наибольший элемент каждой строки меняется местами с наименьшим элементом этой же строки. |
| 11 | Составить программу, которая получает матрицу **А** размером **m\*n** и формирует новую матрицу, поместив в нее только те строки из исходной матрицы, которые образуют монотонную последовательность. |
| 12 | Составить программу, которая получает матрицу **В** размером n\*n и проверяет свойство  где **В**- исходная матрица, **Т** означает транспонирование  Составьте функции транспонирования матрицы и сравнения двух матриц на равенство. |
| 13 | Составить программу, которая, получив матрицу, размером m\*n, создает новую матрицу, в которой наибольший элемент каждой строки меняется местами с элементом главной диагонали этой строки. |
| 14 | Составить программу, которая получает матрицу **А** размером **m\*n**. Из положительных элементов матрицы **A** программа формирует матрицу **B**, размером **m\*r**. Количество элементов r в строке матрицы **B** определяется как максимальное число положительных элементов среди всех строк матрицы **A**. В каждой строке матрицы **B** расположены подряд положительные элементы соответствующей строки матрицы **A**. На место недостающих положительных значений в матрице **В** программа дописывает нули. |
| 15 | Составить программу, которая получает матрицу **А** размером **m\*n** и формирует новую матрицу, все строки которой упорядочены по возрастанию. |
| 16 | Составить программу, которая получает матрицу **Х** размером **m\*n** и матрицу **Y** размером **m\*n**. Программа формирует матрицу  **Z** размером m\*n, где элемент **(i,j)**матрицы **Z** равен сумме элементов **i-**й строки матрицы **X**, которые отсутствуют в **j**–м столбце матрицы **Y**. |
| 17 | Составить программу, которая получает матрицу **Х** размером **n\*n** и матрицу **Y** размером **n\*n**. Программа из двух исходных квадратных матриц **Х** и **Y** формирует результирующую матрицу **Z** путем суммирования каждого элемента каждого столбца матрицы **А** с произведением соответствующей строки матрицы **В**. |
| 18 | Составить программу, которая получает матрицу **А** размером **m\*n** и формирует новую матрицу, строки которой расположены в порядке возрастания сумм элементов этих строк. |
| 19 | Составить программу, которая получает матрицу **А** размером **m\*n** . Программа создает новую матрицу, удалив из каждого столбца матрицы **А** минимальный и максимальные элементы. |
| 20 | Составить программу, которая получает матрицу **Х** размером **n\*n**. Программа определяет количество положительных, нулевых и отрицательных элементов в каждом столбце матрицы X и формирует из них матрицу **Y**, размером **3\*n**. |
| 21 | Составить программу, которая получает матрицу **Х** размером **m\*n** и формирует новую матрицу, получив ее из исходной перестановкой строк: первой с последней, второй с предпоследней и т.д. |
| 22 | Составить программу, которая получает матрицу **C** размером **m\*n**. Программа находит наибольший элемент матрицы **С** и его индексы и получает новую матрицу, разделив все элементы исходной матрицы (кроме найденного наибольшего) на сумму индексов самого элемента и индексов найденного максимального элемента. |
| 23 | Составить программу, которая получает матрицу **Х** размером **m\*n** и находит минимальный и максимальный элементы матрицы. Если оба элемента находятся под главной диагональю, то формируется новая матрица, из которой удалены строки и столбцы, на пересечении которых находятся эти элементы, в противном случае новая матрица не формируется и на экран выдается сообщение об этом. |
| 24 | Составить программу, которая получает матрицу **В** размером **m\*n**. Программа формирует новую матрицу, меняя местами элементы строки с номером i на элементы строки с номером **j** исходной матрицы **В**. |
| 25 | Составить программу, которая получает матрицу **Х** размером **m\*n** и формирует новую матрицу, удалив из исходной матрицы те строки или столбцы, которые оказались равными между собой. Программа выдает на экран исходную и сформированную матрицу с указанием, сколько строк и столбцов удалось удалить в исходной матрице. |
| 26 | Составить программу, которая получает матрицу **А** размером **m\*n** и формирует из нее матрицу **В**, получаемую из исходной перестановкой в каждой строке наименьшего по абсолютной величине элемента с ее диагональным. |
| 27 | Составить программу, которая получает матрицу **Х** размером **m\*n** и вектор **А**, размером **k**. Программа формирует новую матрицу, в которой элементы исходной матрицы, которые совпали с одним из элементов вектора **А**, заменяются **0**. Если в новой матрице есть строки или столбцы, все элементы которых оказались равны **0**, то об этом выдается сообщение. |
| 28 | Составить программу, которая получает матрицу **Х** размером **m\*n** и целое число **К**. Программа формирует новую матрицу, осуществив циклический сдвиг всех строк исходной матрицы вверх **К** раз. |
| 29 | Составить программу, которая получает матрицу **А**  размером **m\*n** и, вычислив сумму и количество элементов матрицы **А**, находящихся под главной диагональю и удовлетворяющих условию, и формирует новую матрицу, удалив из исходной матрицы те строки, которые не содержат элементов отвечающих указанному условию. |
| 30 | Составить программу, которая получает матрицу **А** размером **m\*n**. Программа ищет в каждой строке максимальный элемент и в новой матрице ставит его в начало строки, а минимальный элемент ставит в конце строки, все остальные элементы исходной строки переписывает в строку новой матрицы в обратном порядке. |

## **Пример выполнения задания**

1. ***Задание:*** Составить программу, которая получает матрицу **А** размером **m\*n**. Добавить к исходной матрице новую последнюю строку, состоящую из максимальных значений каждого столбца. Затем добавить к полученной матрице справа столбец произведений положительных элементов каждой строки. Если положительных элементов в строке матрицы нет, то принять произведение равным единице.
2. ***Программный код 1-го сценария***

*// Сценарий zadacha1\_5\_3.sce для решения задачи обработки матрицы*

*// Добавить к исходной матрице новую последнюю строку, состоящую из*

*// максимальных значений каждого столбца. Затем добавить справа*

*// к полученной матрице столбец произведений положительных*

*// элементов каждой строки*

*// Сценарий содержит 3 функции, которые разработаны для решения задачи*

*// Функция поиска максимального элемента каждого столбца*

*// и формирование из них вектора*

*// Входные параметры: матрица A*

*// Выходной параметр: вектор B содержащий значения*

*// максимальных элементов каждого столбца матрицы*

function B=poisk\_max(A)

*//Определение числа строк и столбцов матрицы А*

[m, n] = size(A);

for j=1:n

maxel=A(1,j);

for i=2:m

if A(i,j)>maxel maxel=A(i,j); end

end;

B(j)=maxel;

end

endfunction

*// Функция вычисления произведения положительных элементов*

*// в каждой строке матрицы*

*// Входные параметры: матрица A*

*// Выходной параметр: вектор РР, содержащий значения*

*// произведений положительных элементов каждой строки матрицы*

function PP=Proiz(A)

[m, n] = size(A);

for i=1:m

p=1;

for j=1:n

if A(i,j)>0

p=p\*A(i,j);

end

end

PP(i)=p;

end

endfunction

*// Функция преобразования исходной матрицы*

*// Входные параметры: исходная матрица A,*

*// число строк m и число столбцов n*

*// Выходной параметр: преобразованная матрица A*

function [A]=zadacha1\_5\_3(A, m, n)

*// Вызов функции поиска максимумов в каждом столбце*

B=poisk\_max(A);

A(m+1,:)=B;*// Добавление последней строки*

*// Вызов функции вычисления произведений в каждой строке*

PP=Proiz(A);

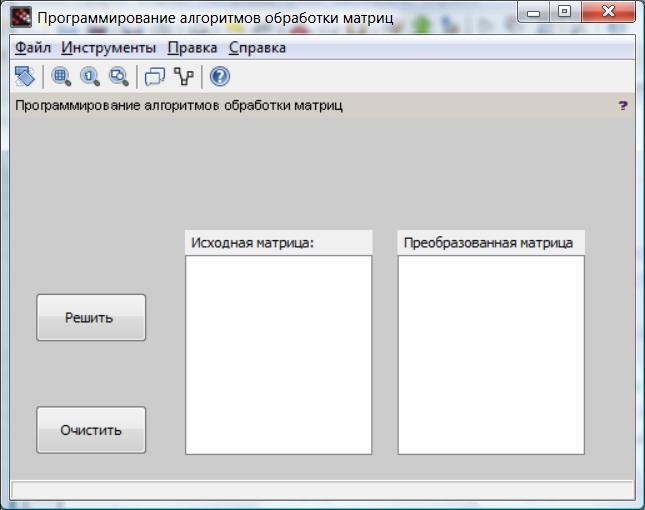
A(:,n+1)=PP; *// Добавление последнего столбца*

еndfunction

// Конец сценария *zadacha1\_5\_3.sce =======================================*

1. ***Графический интерфейс пользователя***

Форма интерфейса пользователя приведена на следующем рисунке:



1. ***Программный код 2-го сценария*** выполнения задания, который содержит ввод исходных данных, вызов функции решения задачи и вывод результатов.

|  |
| --- |
| *// Сценарий выполнения задания Pr1\_5\_3.sce*  *//*  *// Подключение сценария создания интерфейса*  exec('Pr1\_5\_3 \_GUI.sce');  *// Подключение сценария с функциями решения задачи*  exec('zadacha1\_5\_3.sce');  *//*  *// Функция* vivodMatr *создания строковой матрицы для вывода числовой матрицы*  *// Входной параметр - выводимая матрица*  *// Выходной параметр - матрица строк*  function vA=vivodMatr(**A**)  [m, n] = size(**A**);  vA=['']; *// Создание пустой строковой матрицы*  //*Изменение ee размера под размер выводимой матрицы*  vA=resize\_matrix(vA,m,n);  for i =1:m  for j=1:n  vA(i)=vA(i)+sprintf("%7s", string(A(i,j)));  end  end  endfunction  *// Функция YY содержит инструкции ввода исходной матрицы,*  *// вывода ее в соответствующее текс. поле list1,*  *// вызов функции решения задачи (преобразование матрицы)*  *// из сценария zadacha1\_5\_3.sce и инструкции вывода результата*  *// Выполнение функции YY инициируется нажатием кнопки "Выполнить"*  function YY()  *//Ввод матрицы через диалог окно*  A = evstr(x\_dialog('МатрицаA =', ' '));  *//Определение числа строк и столбцов матрицы А*  [m, n] = size(A);  *//Вызов функции создания строковой матрицы из исходной числовой*  vA=vivodMatr(A);  *// Вывод в текстовое поле list1*  set(list1, 'string', vA);  *//Вызов функции решения задачи*  [A]=zadacha1\_5\_3(A,m,n);  *// вывод измененной матрицы в текстовое поле list2*  vA=vivodMatr(A);  set(list2, 'string', vA);  endfunction  *// Функция С очистки текстовых полей*  *// Выполнение функции инициируется нажатием кнопки "Очистить"*  function C()  set(list1, 'string', "");  set(list2, 'string', "");  endfunction  // *Конец сценария Pr1\_5\_3.sce =============================================* |

1. ***Программный код*** создания графического интерфейса

|  |
| --- |
| *// Сценарий Pr1\_5\_3\_GUI.sce создания интерфейса пользователя*  *//*  *// Создание графического окна*  f=figure('position',[20,40,500,290],…  'figure\_name','Програм-ние алгоритмов обработки матриц');  *//*  *// Создание listbox для вывода матриц*  list1 = uicontrol(f,'style','listbox','position', [140 20 150 160]);  list2=uicontrol(f,'style','listbox','position', [310 20 150 160]);  *//*  *//Создание текстового поля, для вывода исходной матрицы*  text1=uicontrol(f,'style','text','string','',…  'position',[140,180,150,20],'string',' Исходная матрица:');  *//*  *//Создание текстового поля,для вывода преобразованной матрицы*  text2=uicontrol(f,'style','text','string','',…  'position', [310,180,150,20],…  'string','Преобразованная матрица');  *// Создание кнопки и установление ихсвойств*  bt1=uicontrol('style','pushbutton','string','Решить ',...  'position',[20,110,90,40],'CallBack','YY');  bt2=uicontrol('style','pushbutton','string','Очистить',...  'position',[20,20,90,40],'CallBack','C');  *// Конец сценария Pr1\_5\_3\_GUI.sce ===================================* |

-->// Сессия Командного окна для выполнения задания 3

*-->*// *Практическое занятие 1.5 по теме:*

*-->*// *«Программирование алгоритмов обработки матриц»*

*-->*// *Вариант 0, задание 3*

*-->*// *Выполнил: Студент группы БИН1801 Иванов И.И.*

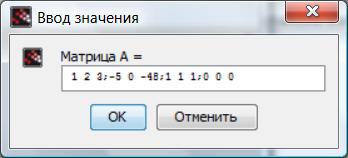
*-->*

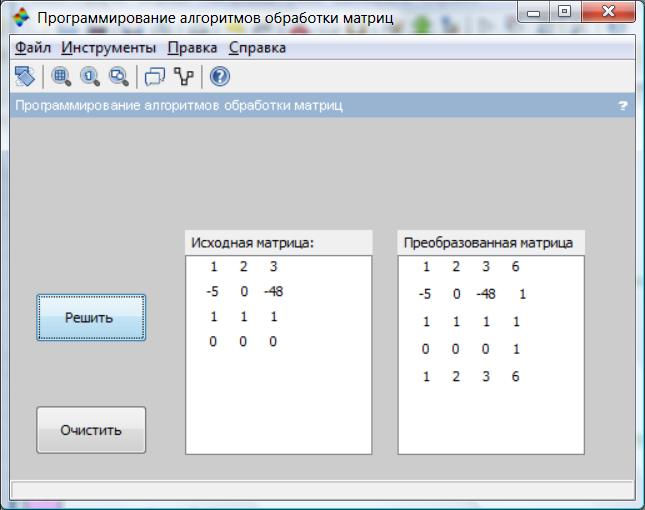
*-->*// Загрузка и выполнение сценария *Pr1\_5\_3.sce*

-->exec('*Pr1\_5\_3.sce*');

*-->*// Конец сессии ========================================================

1. ***Результат выполнения приложения***

******

****

## **,Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя (по требованию преподавателя).
5. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* номер задания и формулировка решаемой задачи и/или краткое перечисление действий, реализуемых в сценарии.

1. Протокол сессии ***Командного окна***, вначале которого должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **Контрольные вопросы по теме**

1. Что такое **sci**-функция и каковы её отличия от сценария?
2. Формат **sci**-функции.
3. Каким образом **sci**-функция загружается в оперативную память и запускается на выполнение?
4. Можно ли в одном сценарии описать несколько функций?
5. В какой последовательности должны быть расположены **sci**-функции в одном сценарии, если одна из них является главной, которая вызывает остальные функции?
6. Может ли ***sci***-функция участвовать подобно библиотечным функциям в записи арифметического выражения?
7. Что такое формальные и фактические параметры функции?
8. Можно ли переменные, используемые в функции, созданной в сценарии, после его загрузки увидеть в окне ***Обозреватель переменных***?
9. Может ли функция иметь несколько выходных параметров?
10. Может ли функция не иметь выходных параметров?
11. Может ли функция не иметь входных параметров?
12. Может ли функция не иметь ни входных, ни выходных параметров?
13. Может ли передаваться в качестве входного (или выходного) параметра матрица?
14. Может ли входным параметром функции являться число?
15. Формат оператора **input**.
16. Как с использованием оператора **if…end** реализовать стандартное, усеченное и вложенное разветвление?
17. Формат оператора множественного разветвления **selec*t***.
18. Формат оператора регулярного цикла **for…end**, особенности задания значений переменной цикла.
19. Оператор итеративного цикла **while…end** и его структура.
20. Назначение операторов **continue** и **brek.**
21. Что представляет собой итеративный цикл?
22. В чем отличие организации регулярных и итеративных циклов?

Раздел 2. Решение вычислительных задач средствами Scilab

# **2.1. Практическое занятие по теме**

# **«Использование полиномов»**

**Цель работы:** Изучение способов создания полиномов в Scilab, операций и функций над полиномами, нахождения корней, производных от полиномов.

## **2.1.1. Вопросы, подлежащие изучению**

1. Представление полиномов в Scilab.
2. Создание полиномов с использованием функции **pol*y***.
3. Оценка значений полиномов.
4. Операции и функции над данными типа полином.
5. Нахождение корней и производных от полиномов.

## **2.1.2. Общее задание**

1. ***Изучить материал учебника*** [1] (п. 2.1)***.***
2. ***Выбрать индивидуальное задание*** *из* ***табл. 2.1-1****.*
3. ***Создать сценарий*** *для выполнения практического задания.*
4. ***Создать вектор-строку V1*** *из коэффициентов полинома.*
5. ***Создать полином* P1(х)** *с помощью функции* ***poly*,** *в которой в качестве параметра используется имя вектора* ***V1****.*
6. ***Создать произвольный полином* P2(х)** *с помощью функции* ***poly****, параметром которой является не имя вектора, а сам числовой вектор произвольных коэффициентов.*
7. ***Оценить значения полинома* P1(х)** *в нескольких точках.*
8. ***Вычислить корни*** *полученного полинома* **P1(х)***, используя функцию* **roots***.*
9. ***Создать*** *новый полином* **P3(х)** *по полученным корням полинома***P1** *с использованием соответствующего формата функции* **poly***. Сравнить полиномы* **P1(х)** *и* **P3(х)***.*
10. ***Вычислить*** *корни полинома* **P1(х)** *с использованием другого формата функции* ***roots****, позволяющего вычислить корни полинома по вектору его коэффициентов.*
11. ***Вычислить производную*** *от созданного полинома* **P3(х)**.
12. ***Преобразовать полиномы*** *в символьную строку.*
13. ***Найти*** *сумму, разность, произведение и частное полиномов* **P1(х)** *и* **P2(х)**.
14. ***Спроектировать приложение:*** *«Использование полиномов» для ввода данных и отображения результатов (по требованию преподавателя)*
15. ***Предоставить*** *результаты работы преподавателю и* ***ответить*** *на поставленные вопросы.*
16. ***Оформить отчет*** *по выполненной работе*.

## **2.1.3. Варианты заданий**

Таблица 2.1-1

|  |  |  |  |
| --- | --- | --- | --- |
| **№ варианта** | **Полином  *P1*(*x*)** | **№ варианта** | **Полином *P1*(*x*)** |
| 1. | *x*4 +16*x*3 – 35*x*2 +18*x* | 16. | *х3* – 9*x*2 + 24*х* –20 |
| 2. | *x*4 – 12.5*x*3 + 49*x*2 – 60*x* | 17. | *x*3– 16*x*2 +85*x*–150 |
| 3. | *x*3 +7*x*2– 28*x* +20 | 18. | *x*3– 12*x2 +*45*х*−54 |
| 4. | *x*3 – 9.5*x*2 + 29.5*x*-30 | 19. | *x*3 – 3.9*x2 +*2.88*х*– 0,54 |
| 5. | *x*4 −12*x*3 +47*x*2 − 60*x* | 20. | *х3*– 2.9*x*2 +1.98*x*–0,36 |
| 6. | *x*3 - 15*x*2 +71*x*− 105 | 21. | *x*3 – 5.6*x*2+ 9*x*– 3,6 |
| 7. | *x*4 − 21*x*3 + 151*x*2 − 411*x*+ 280 | 22. | *x*3 – 9*x*2 + 26*х* – 24 |
| 8. | *x*3 − 14*x*2 + 53*x* - 40 | 23. | *x*3 – 12*x2*+ 48*х* − 64 |
| 9. | *x*3 – 23*x*2 + 174*x*− 432 | 24. | *x*3 – 10.5*x2* + 36*x* – 40.5 |
| 10. | *x*4  − 21*x*3 + 147*x*2 − 343*x* | 25. | *x*3 – 6*x*2 + 8.75*x* |
| 11. | *x*3 − 14*x*2+49*x* | 26. | *x*3 – 5.5*x2*+ 9.75*х* – 5.625 |
| 12. | *x*4 − 25*x*3 + 220*x*2− 780*x*+ 864 | 27. | х3 – 10.5*x*2 +13.5*х* |
| 13. | *x*4− 22*x*3 + 168*x*2 − 490*x +*343 | 28. | *x*3 –17.5*x*2 + 84*x*– 90 |
| 14. | *x*4 − 6*x*3 + 13*x*2 − 12*x +*4 | 29. | *x*4 –7.5 *x*3 + 12.5*x*2 + 7.5*x*– 13.5 |
| 15. | *x*3 − 5*x*2*+8x*− 4 | 30. | *x*3 – 15.5*x2*+54*х* – 49.5 |

## **2.1.4. Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя
5. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* вариант индивидуального задания.

1. Протокол сессии ***Командного окна***, вначале которой должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **2.1.5. Контрольные вопросы по теме**

1. Является ли полином типом данных?
2. Каким образом задать полином в виде вектора степеней?
3. Каким образом задать полином с использованием системной константы **%z.**
4. Какой формат имеет функция **poly**?
5. Как задать матрицу полиномов?
6. Как оценить значение полинома с использованием функции **horner**?
7. Каков формат вычисления корней полинома с использованием функции **roots**?
8. Можно ли создать полином по его корням?
9. Можно ли с полиномами осуществлять операции, такие как: сложение, вычитание, умножение и деление?
10. Каков формат функции **derivat** для вычисления производной от полинома?
11. Как преобразовать полином в символьную строку?

# **2.2. Практическое занятие по теме**

# **«Аппроксимация и интерполяция функций»**

**Цель работы:** Изучение технологий аппроксимации и интерполяции с использованием функций Scilab, получение интерполяционных многочленов в явном виде и построение их графиков.

## **2.2.1. Вопросы, подлежащие изучению**

1. Постановки задачи аппроксимации и интерполяции функции [7].
2. Технология аппроксимации табличной функции с использованием функции **datafit**.
3. Технология линейной, кубической сплайн-интерполяции таблично заданной функции**.**
4. Получение интерполяционных многочленов в явном виде.
5. Построение графиков аппроксимирующих и интерполирующих функций.

## **2.2.2. Общее задание**

1. ***Изучить материал учебника*** [1] (*п. 2.2).*
2. ***Выбрать индивидуальное задание****: номера узлов и номер аппроксимируемой функции из* ***табл. 2.2-1****; узлы аппроксимации и значения функции в узлах из* ***табл. 2.2-2****.*
3. ***Создать сценарий*** *для выполнения практического задания.*
4. ***Задать*** *в виде векторов значения узлов и соответствующие им значения функции.*
5. ***Вычислить коэффициенты*** *аппроксимирующих функций для линейной, квадратичной и кубической аппроксимации с использованием* **datafit** *и* ***получить*** *соответствующие аппроксимирующие функции в явном виде.*
6. ***Вычислить*** *с использованием полученных аппроксимирующих функций значения в произвольной точке, принадлежащей отрезку, но не совпадающей с узлами аппроксимации, и* ***сравнить полученные*** *результаты.*
7. ***Построить графики*** *табличной и трех аппроксимирующих функций в одном графическом окне, снабдив их легендой* [5]*.*
8. ***Провести*** *линейную и кубическую интерполяцию функции, заданной таблично, с использованием* **interpln** *и* **interp***.* ***Получить*** *значения интерполирующих функций в точке, не совпадающей с узлами интерполяции, и* ***проанализировать*** *полученные результаты.*
9. ***Построить*** *графики табличной и двух интерполирующих функций в одном графическом окне, снабдив их легендой* [5].
10. ***Спроектировать и реализовать приложение****:* «*Аппроксимации и интерполяции функций» для ввода данных и отображения результатов (по требованию преподавателя).*
11. ***Предоставить*** *результаты работы преподавателю и* ***ответить*** *на поставленные вопросы.*
12. ***Оформить отчет*** *по выполненной работе.*

## **2.2.3. Варианты заданий**

## Таблица 2.2-1

|  |  |  |
| --- | --- | --- |
| **№** | **Номера узлов xi** | **Номер функции** |
| 1 | 1 3 5 7 9 10 13 |  |
| 2 | 1 2 4 5 7 10 12 |  |
| 3 | 1 3 6 7 10 11 13 |  |
| 4 | 1 2 4 7 9 11 13 |  |
| 5 | 3 6 7 9 10 11 12 |  |
| 6 | 2 3 6 8 9 10 13 |  |
| 7 | 1 4 5 7 9 11 12 |  |
| 8 | 1 2 4 7 9 12 13 |  |
| 9 | 2 3 5 7 8 11 12 |  |
| 10 | 1 3 6 7 9 10 13 |  |
| 11 | 1 3 7 8 10 11 13 |  |
| 12 | 1 2 5 6 7 10 12 |  |
| 13 | 1 4 5 8 10 12 13 |  |
| 14 | 1 3 5 7 9 10 13 |  |
| 15 | 1 3 6 7 8 10 13 |  |
| 16 | 1 4 5 7 9 11 12 |  |
| 17 | 2 4 5 6 8 12 13 |  |
| 18 | 1 4 5 7 9 11 12 |  |
| 19 | 1 4 5 8 10 11 12 |  |
| 20 | 2 4 5 6 8 12 13 |  |
| 21 | 1 4 5 810 12 13 |  |
| 22 | 2 3 6 8 9 10 13 |  |
| 23 | 1 3 5 8 10 12 13 |  |
| 24 | 1 4 5 7 9 11 12 |  |
| 25 | 2 4 5 6 8 12 13 |  |
| 26 | 3 4 5 7 8 9 12 |  |
| 27 | 3 5 8 10 11 12 13 |  |
| 28 | 2 4 7 9 10 11 13 |  |
| 29 | 2 4 5 7 8 10 12 |  |
| 30 | 1 4 5 7 9 11 13 |  |

Таблица 2.2-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **i** | **xi** |  |  |  |
| 1 | -5 | 1.38 | 2.44 | 1.676 |
| 2 | -4.5 | 1.221 | 2.359 | 2.025 |
| 3 | -4 | 1.511 | 1.751 | 1.736 |
| 4 | -3.5 | 1.501 | 2.13 | 1.203 |
| 5 | -3 | 1 | 1.455 | 1.511 |
| 6 | -2.5 | 0.728 | 1.482 | 1.362 |
| 7 | -2 | 0.976 | 1.437 | 0.75 |
| 8 | -1.5 | 1.065 | 0.803 | 0.976 |
| 9 | -1 | 0.599 | 1.175 | 0.957 |
| 10 | -0.5 | 0.192 | 0.49 | 0.272 |
| 11 | 0 | 0.3 | 0.375 | 0.3 |
| 12 | 0.5 | 0.319 | -6.51\*10-3 | 0.165 |
| 13 | 1 | -0.405 | -1.965 | -1.185 |

## **2.2.4. Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя (по требованию преподавателя).
5. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* вариант индивидуального задания.

1. Протокол сессии ***Командного окна***, вначале которой должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **2.2.5 Контрольные вопросы по теме**

1. Что такое аппроксимация функции, и в каких случаях она используется?
2. В чем отличие аппроксимации от интерполяции?
3. В чем заключается основное условие интерполяции?
4. Какой метод аппроксимации реализован в функции **datafit**?
5. Что служит результатом выполнения функции **datafit**?
6. Какой параметр функции **datafit** определяет погрешность аппроксимации?
7. При использовании какого количества узлов функция **datafit** решает задачу интерполяции и почему?
8. Назначение и формат функции **interpln**?

# **2.3. Практическое занятие по теме**

# **«Интегрирование и дифференцирование в среде Scilab»**

**Цель работы:** Изучение возможностей вычисления значений определенных интегралов, а также значений производной в точке с использованием функций Scilab.

## **2.3.1. Вопросы, подлежащие изучению**

1. Постановка задачи численного интегрирования [7].
2. Вычисление значения определенного интеграла с использованием функций **inttrap** и **integrate**.
3. Вычисление значения производной в точке с использованием функций Scilab **numderivative**.
4. Методы численного интегрирования: метод трапеций и метод Симпсона.

## **2.3.2. Общее задание**

1. ***Изучить материал учебника*** [1](*п. 2.3).*
2. ***Выбрать*** *из* ***табл. 2.3-1*** *вариант индивидуального задания.*
3. ***Создать сценарии*** *для выполнения практического задания.*
4. ***Описать*** *подынтегральную функцию***f(x)***.*
5. ***Вычислить*** *определенный интеграл с использованием функции* **inttrap(x,y)***, реализующей формулу трапеций, получив предварительно таблицу значений подынтегральной функции с шагом* **h***, и задав значения аргумента и функции в виде векторов.*
6. ***Вычислить*** *значение определенного интеграла с различной точностью, используя функцию* **integrate('f',a,b)***, реализующей формулу Симпсона, где* **f** *– имя подынтегральной функции, взятое в одинарные кавычки.*
7. ***Вычислить*** *производную от функции* **y(x)***в нескольких точках, задавая их значения как вектор аргументов. Затем вычислить производную в этих же точках по отдельности. Сравнить результаты.*
8. ***Спроектировать и реализовать приложение****:* ***«Интегрирование и дифференцирование в среде Scilab»*** *для ввода данных и отображения результатов (по требованию преподавателя).*
9. ***Предоставить*** *результаты работы преподавателю и* ***ответить*** *на поставленные вопросы.*
10. ***Оформить отчет*** *по выполненной работе****.***

## **2.3.3. Варианты индивидуальных заданий**

Таблица 2.3-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Интеграл** | **y(x)** | **№** | **Интеграл** | **y(x)** |
| 1 |  |  | 16 |  |  |
| 2 |  |  | 17 |  |  |
| 3 |  |  | 18 |  |  |
| 4 |  |  | 19 |  |  |
| 5 |  |  | 20 |  |  |
| 6 |  |  | 21 |  |  |
| 7 |  |  | 22 |  |  |
| 8 |  |  | 23 |  |  |
| 9 |  |  | 24 |  |  |
| 10 |  |  | 25 |  |  |
| 11 |  |  | 26 |  |  |
| 12 |  |  | 27 |  |  |
| 13 |  |  | 28 |  |  |
| 14 |  |  | 29 |  |  |
| 15 |  |  | 30 |  |  |

**2.3.4. Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя
5. Сценарии, вначале которых должна быть также введена информация в виде комментариев:

* имя и назначение сценария;
* вариант индивидуального задания и номер задания.

1. Протокол сессии ***Командного окна***, вначале которой должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **2.3.5. Контрольные вопросы по теме**

1. Что такое определенный интеграл?
2. Каково назначение функции **inttrap**?
3. Что возвращает функция **inttrap,** если **y(x)** – матрица?
4. Можно ли использовать функцию **inttrap,** если узлы по оси **х** – не равноотстоящие?
5. Каким образом вычислить интеграл, если подынтегральная функция задана таблицей с не равноотстоящими узлами?
6. Какая функция Scilab позволяет вычислить определенный интеграл с заданной точностью?
7. Какой численный метод заложен в функции **integrate**?
8. Какие существуют способы задания подынтегральной функции при вычислении определенного интеграла с использованием функции **integrate**?
9. Как задать допустимую погрешность вычисления определенного интеграла?
10. С какой точностью производится вычисление определенного интеграла, если погрешность не задана?
11. Какой встроенной функцией системы Scilab вычисляется производная в заданной точке?
12. Какое назначение и формат имеет функции **numderivative**?

# **2.4. Практическое занятие по теме**

# **«Решение нелинейных уравнений»**

**Цель работы:** Изучение средств решения нелинейных уравнений в Scilab, реализации этапов отделения и уточнения корней нелинейных уравнений.

## **2.4.1. Вопросы, подлежащие изучению**

1. Постановка задачи решения нелинейных уравнений [7].
2. Этапы решения нелинейных уравнений: отделение и уточнение корня.
3. Графическое и аналитическое отделение корней нелинейных уравнений.
4. Построение графиков функций нелинейных уравнений средствами Scilab.
5. Получение таблиц значений функций и производных в заданных границах изменения аргумента.
6. Решение нелинейных уравнений с использованием встроенных функций пакета Scilab: **fsolve**,**poly**и **roots**.

## **2.4.2. Общее задание**

1. ***Изучить материал учебника*** [1] *(п. 2.4)*.
2. ***Выбрать индивидуальный вариант*** *задания из* ***табл. 2.4-1****.*
3. ***Создать сценарий*** *для выполнения практического задания.*
4. ***Отделить корень*** *нелинейного уравнения* **f(x)=0** *с использованием средств пакета Scilab, для чего необходимо:*

* ***построить графики*** *функции* **f(x)***и ее первой производной;*
* ***выбрать отрезок****, на котором существует один корень;*
* ***получить таблицы*** *значений аргумента, функции* **f(x)***и ее первой производной на выбранном отрезке пересечения графика с осью* **ОХ***, причем для вычисления производной воспользуйтесь функцией* **numderivative***;*
* ***проверить условие*** *существования единственного корня на выбранном отрезке.*

1. ***Решить 1-е нелинейное уравнение*** *с использованием* **fsolve***, получив значение корня и значение функции в точке корня.*
2. ***Задать*** *для* ***2-го уравнения*** *вектор коэффициентов.*
3. ***Сформировать*** *с использованием функции* **poly** *полином с коэффициентами, хранящимися в векторе.*
4. ***Вычислить*** *корни полученного полинома, используя функцию* **roots***.*
5. ***Спроектировать и реализовать приложение****:* ***«Решение нелинейных уравнений»*** *для ввода данных и отображения результатов (по требованию преподавателя).*
6. ***Предоставить*** *результаты работы преподавателю и* ***ответить*** *на поставленные вопросы.*
7. ***Оформить отчет*** *по выполненной работе.*

## **2.4.3. Варианты индивидуальных заданий**

Таблица 2.4 -1

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Уравнение** | **№** | **Уравнение** |
| 1 |  | 16 |  |
| 2 | =0 | 17 |  |
| 3 |  | 18 |  |
| 4 | =0 | 19 |  |
| 5 |  | 20 |  |
| 6 |  | 21 |  |
| 7 |  | 22 |  |
| 8 | =0 | 23 |  |
| 9 |  | 24 |  |
| 10 |  | 25 |  |
| 11 |  | 26 | =0 |
| 12 |  | 27 |  |
| 13 |  | 28 |  |
| 14 |  | 29 |  |

## **2.4.4. Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя
5. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* вариант индивидуального задания и номер задания.

1. Протокол сессии Командного окна, вначале которой должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **2.4.5. Контрольные вопросы по теме**

1. Что называется нелинейным уравнением?
2. Из каких этапов состоит решение нелинейного уравнения численным методом?
3. В чем состоит графический метод отделения корней нелинейного уравнения и как он реализуется средствами Scilab?
4. В чем состоит аналитический метод отделения корней нелинейного уравнения и как он реализуется средствами Scilab?
5. Как задается функция нелинейного уравнения при его решении с использованием функции **fsolve**?
6. Какая функция Scilab позволяет решить нелинейное уравнение с заданной точностью?
7. В чем назначение и каков формат функции **fsolve**?
8. Для каких нелинейных уравнений в ходе их решения используется функция **poly**?
9. Какие нелинейные уравнения решаются с использованием функции **roots**?
10. Какая функция Scilab позволяет найти как действительные, так и мнимые корни нелинейного уравнения?
11. Какие выходные параметры могут иметь функции **fsolve** и **roots**?
12. Как проверить правильность полученных корней нелинейного уравнения?

# **2.5. Практическое занятие по теме**

# **«Решение обыкновенных**

# **дифференциальных уравнений»**

**Цель работы:** Изучение средств Scilab для решения обыкновенных дифференциальных уравнений (ОДУ)1-го и 2-го порядка, а также систем ОДУ, получение решения ОДУ в виде таблицы и графика.

## **2.5.1**. **Вопросы, подлежащие изучению**

1. Постановка задачи решения ОДУ [7].
2. Функция **ode**, предназначенная для решения ОДУ.
3. Вывод результатов решения ОДУ в виде таблицы.
4. Получение графического решения ОДУ.
5. Решение ОДУ 2-го порядка с использованием функции **ode**.

## **2.5.2. Общее задание**

1. ***Изучить материал учебника*** [1] (*п. 2.5).*
2. ***Выбрать*** *индивидуальный вариант задания из* ***табл. 2.5-1****.*
3. ***Создать сценарии*** *для выполнения практического задания.*
4. ***Найти*** *решение ОДУ на отрезке* **[x0;b]** *с шагом* **h** *с использованием функции* **ode**.
5. ***Создать*** *матрицу решений, записав в первый столбец аргумент, а во второй - решение, полученное с использованием функций* **ode***.*
6. ***Вывести*** *полученную таблицу по столбцам.*
7. ***Построить*** *график полученного решения ОДУ.*
8. ***Выбрать*** *индивидуальный вариант задания из* ***табл. 2.5-2****.*
9. ***Создать сценарий*** *для решения системы ОДУ или решения ОДУ 2-го порядка согласно вашему варианту.*
10. ***Найти*** *решение системы ОДУ, вывести матрицу решения и построить график решения системы ОДУ.*
11. ***Спроектировать и реализовать приложение****:* ***«Решение обыкновенных дифференциальных уравнений»*** *для ввода данных и отображения результатов (по требованию преподавателя).*
12. ***Предоставить*** *результаты работы преподавателю и* ***ответить*** *на поставленные вопросы.*
13. ***Оформить отчет*** *по выполненной работе.*

## **2.5.3. Варианты индивидуальных заданий**

Таблица 2.5-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **ОДУ** | **Начальные условия** | **b** | **h** |
| 1 |  |  | 6 | 0,25 |
| 2 |  |  | 7 |  |
| 3 |  |  | 1.2 | 0.1 |
| 4 |  |  | 4 | 0,1 |
| 5 |  |  | 6 | 0,25 |
| 6 |  |  | 4 | 0.1 |
| 7 |  |  | 1.4 | 0,2 |
| 8 |  |  | 1.5 | 0,2 |
| 9 |  |  | 5 |  |
| 10 |  |  | 6 | 0,25 |
| 11 |  |  | 4 | 0,1 |
| 12 |  |  | 6 | 0,25 |
| 13 |  |  | 2,5 | 0.25 |
| 14 |  |  | 2 | 0,1 |
| 15 |  |  | 4 | 0,25 |
| 16 |  |  | 3 | 0,25 |
| 17 |  |  | 5 | 0,25 |
| 18 |  |  | 5 | 0,25 |
| 19 |  |  | 6 | 0,5 |
| 20 |  |  | 4 | 0,25 |
| 21 |  |  | 2 | 0.1 |
| 22 |  |  | 8 | 0,25 |
| 23 |  |  | 2,5 | 0,25 |
| 24 |  |  | 6.5 | 0.25 |
| 25 |  |  | 5.5 | 0,25 |
| 26 |  |  | 4 | 0,125 |
| 27 |  |  | 5 | 0,5 |
| 28 |  |  | 4,4 | 0,2 |
| 29 |  |  | 5 | 0,2 |
| 30 |  |  | 7 | 0.5 |

Таблица 2.5-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***№*** | ***Система ОДУ или ОДУ 2-го порядка*** | ***Начальные условия*** | ***b*** | ***h*** |
| *1* |  | *y(0)=2, z(0)=1* | *2* | *0,2* |
| *2* |  | *y(0)=1, z(0)=5* | *1* | *0,1* |
| *3* |  | *y(0)=1, z(0)=1/4* | *1* | *0,1* |
| *4* |  | *y(π)=0, z(π)=3* | *2π* | *π/6* |
| *5* |  | *y(0)=0, z(0)=0* | *1* | *0,1* |
| *6* |  | *y(0)=0, z(0)=0* | *1* | *0,1* |
| *7* |  | *y(0)=2, z(0)=1* | *2* | *0,2* |
| *8* |  | *y(1)=1, z(1)=0* | *2* | *0,1* |
| *9* |  | *y(0)=2, z(0)=-1* | *2* | *0,2* |
| *10* |  | *y(0)=1, z(0)=0* | *2* | *0,2* |
| *11* |  | *y(0)=1, z(0)=0* | *1* | *0,1* |
| *12* |  | *y(0)=1, z(0)=7* | *2* | *0,2* |
| *13* |  | *y(0)=2, z(0)=3* | *π* | *π/6* |
| *14* |  | *y(0)=0, z(0)=-1/20* | *2* | *0,2* |
| *15* |  | *y(0)=0, z(0)=1* | *2* | *0,2* |
| *16* | *y’’– 2y’+ y=ex(x2+1)* | *y(0)=1, y’(0)=2* | *2* | *0,2* |
| *17* | *y’’+ 3y’+2y=2cos(3x)+4sin(3x)* | *y(0)=1, y’(0)=2* | *4* | *0,4* |
| *18* | *y’’+ y’=x sin(x)* | *y(0)=1, y’(0)=4* | *2* | *0,2* |
| *19* | *y’’– 2y’+y = 2ex* | *y(0)=2, y’(0)=5* | *1* | *0,1* |
| *20* | *y’’– 2y’– 8y = e2x+5* | *y(0)=3, y’(0)=1* | *1* | *0,1* |
| *21* | *y’’+ y’ = 5x+2ex* | *y(0)=1, y’(0)=3* | *2* | *0,2* |
| *22* | *y’’– y’ = 2x – 1 – 3e-x* | *y(0)=1, y’(0)=4* | *2* | *0,2* |
| *23* | *y’’ – y’ = 2x sin(x)* | *y(0)=2, y’(0)=2* | *2* | *0,2* |
| *24* | *y’’+ y’ = cos(3x)* | *y(π/2)=4, y’(π/2)=1* | *2π* | *π/4* |
| *25* | *y’’+2y’+y = e-x* | *y(0)=1, y’(0)=2* | *π* | *π/6* |
| *26* | *y’’ – 4 y’+4y = x2* | *y(0)=1, y’(0)=2* | *1* | *0,1* |
| *27* | *y’’– 2y =ex* | *y(0)=1, y’(0)=4* | *2* | *0,2* |
| *28* | *y’’+ 4y =sin(x)* | *y(0)=1, y’(0)=1* | *π* | *π/6* |
| *29* | *y’’– 2y’+5y=ex +cos(2x)* | *y(0)=1, y’(0)=1* | *π* | *π/6* |
| *30* | *y’’ – 8y’ +7y=14* | *y(0)=1, y’(0)=2* | *1* | *0,1* |

## **2.5.4. Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя
5. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* вариант индивидуального задания и номер задания.

1. Протокол сессии ***Командного окна***, вначале которой должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **2.5.5. Контрольные вопросы по теме**

1. Какие начальные условия должны быть заданы в соответствии с задачей Коши при решении ОДУ средствами системы Scilab?
2. В чем отличие аналитического решения ОДУ от численного?
3. Какие численные методы реализованы в функции **ode**?
4. Какие параметры являются в функции **ode** обязательными?
5. В какой форме должно быть записано ОДУ при использовании **ode**?
6. Какими средствами решается система ОДУ?
7. Что такое порядок ОДУ?
8. Что представляет собой решение ОДУ 1-го и 2-го порядка при использовании функции **ode?**

# **2.6. Практическое занятие по теме**

# **«Решение систем линейных уравнений»**

**Цель работы:** Изучение способов решения систем линейных уравнений с использованием встроенной функции Scilab, а также сценариев, реализующих соответствующие численные методы.

## **2.6.1. Вопросы, подлежащие изучению**

1. Постановка задачи решения системы линейных уравнений (СЛУ) [7].
2. Вычисление значений корней СЛУ с использованием функции Scilab **linsolve**.
3. Вычисление значений корней СЛУ с использованием ***метода обратной матрицы***.
4. Методы численного решения СЛУ: метод *Крамера* и метод *Гаусса*.
5. Вычисление значений корней СЛУ в Scilab по методу Крамера.
6. Вычисление значений корней СЛУ в Scilab по методу Гаусса.

## **2.6.2. Общее задание**

1. ***Изучить материал учебника*** [1] (п. 2.6).
2. ***Выбрать*** *из* ***табл. 2.6-1*** *вариант индивидуального задания.*
3. ***Создать сценарий*** *для выполнения практического задания.*
4. ***Описать*** *матрицу коэффициентов* **A** *и вектор свободных членов***b***.*
5. ***Решить СЛУ*** *с использованием функции* **linsolve(A,-b)***, сохранив решение СЛУ в векторе* **Х1***.*
6. ***Решить СЛУ*** *с использованием метода обратной матрицы в матричной форме* **x2=b\A***.*
7. ***Сравнить полученные решения*** *и* ***проверить*** *их с использованием формул* **b1=x1\*A***и***b2=x2\*A.**
8. ***Найти решение СЛУ*** *по методу Крамера, сохранив корни в векторе* **Х3***. и провести проверку по формуле***P1 = A \* X3 - b***.*
9. ***Найти решение СЛУ*** *по методу Гаусса, сохранив корни в векторе* **Х4***. и провести проверку по формуле* **P2 = A\* X4 - b***. Сравнить результаты.*
10. ***Спроектировать и реализовать приложение****:* ***«Решение систем линейных уравнений»*** *для ввода данных и отображения результатов (по требованию преподавателя).*
11. ***Предоставить*** *результаты работы преподавателю и* ***ответить*** *на поставленные вопросы.*
12. ***Оформить отчет*** *по выполненной работе*

## **2.6.3. Варианты индивидуальных заданий**

***Таблица 2.6-1***

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Система уравнений** | **№** | **Система уравнений** |
| 1 |  | 16 |  |
| 2 |  | 17 |  |
| 3 |  | 18 |  |
| 4 |  | 19 |  |
| 5 |  | 20 |  |
| 6 |  | 21 |  |
| 7 |  | 22 |  |
| 8 |  | 23 |  |
| 9 |  | 24 |  |
| 10 |  | 24 |  |
| 11 |  | 26 |  |
| 12 |  | 27 |  |
| 13 |  | 28 |  |
| 14 |  | 29 |  |
| 15 |  | 30 |  |

## **2.6.4. Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя
5. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* вариант индивидуального задания.

1. Протокол сессии ***Командного окна***, вначале которой должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **2.6.5. Контрольные вопросы по теме**

1. Как формулируется постановка задачи численного решения системы линейных уравнений?
2. Какие методы решения СЛУ известны?
3. Что называют плохо ***обусловленными*** линейными системами?
4. Какой формат имеет функция **linsolve**?
5. Что представляет собой алгоритм решения СЛУ средствами Scilab по *методу Крамера*?
6. Что представляет собой алгоритм решения СЛУ средствами Scilab по *методу Гаусса*?

# **2.7. Практическое занятие по теме**

### «Решение задач одномерной и многомерной

### оптимизации»

**Цель работы:** Изучение средств Scilab для решения задач одномерной и многомерной оптимизации и технологии поиска координат точки минимума с использованием решателей.

## **2.7.1. Вопросы, подлежащие изучению**

1. Постановка задачи одномерной оптимизации [7].
2. Условие унимодальности функции **f(x)**на отрезке.
3. Получение таблиц значений целевой функции и её производных с использованием средств пакета Scilab.
4. Постановка задачи многомерной оптимизации.
5. Построение графиков функции от двух переменных   
   **F(x1,х2)**и ее линий уровня средствами пакета Scilab.
6. Технология поиска координат точки минимума одномерной и многомерной функции **F(x1,х2)** с использованием решателя **optim** и **fminsearch**.

## **2.7.2. Общее задание**

1. ***Изучить материал учебника*** [1] (п.2.7.).
2. ***Выбрать*** *индивидуальное задание из* ***табл. 2.7-1****.*
3. ***Создать сценарий*** *для выполнения практического задания.*
4. ***Построить*** *график функции* **f(x)** *и найти отрезок, содержащий единственный минимум.*
5. ***Проверить*** *на выбранном отрезке условие унимодальности функции, получив таблицу значений первой или второй производной.*
6. ***Выбрать*** *начальную точку поиска минимума* **x0***.*
7. ***Создать*** *вспомогательную функцию* **costf1** *для вычисления значений целевой функции и её производной.*
8. ***Найти*** *координаты точки минимума* **f(x)** *с использованием решателей* **optim** *и* **fminsearch**. *Сравнить полученные результаты.*
9. ***Построить*** *графики функции* **F(x1,х2)** *и её линий уровней, и выбрать координаты начальной точки поиска минимума (***x0, y0***).*
10. ***Создать*** *для использования функции* **optim***вспомогательную функцию* **costf2***, необходимую для вычисления значений целевой функции и её частных производных.*
11. ***Найти*** *координаты точки минимума функции* **F(x1,х2)** *и значение функции в этой точке с использованием решателей* **optim** *и* **fminsearch**. *Сравнить полученные результаты.*
12. ***Спроектировать и реализовать приложение****:* ***«Решение задач одномерной и многомерной оптимизации»*** *для ввода данных и отображения результатов (по требованию преподавателя).*
13. ***Предоставить*** *результаты работы преподавателю и* ***ответить*** *на поставленные вопросы.*
14. ***Оформить отчет*** *по выполненной работе.*

## **2.7.3. Варианты индивидуальных заданий**

Таблица 2.7-1

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Функции** | № | **Функции** |
| 1 |  | 16 |  |
| 2 |  | 17 |  |
| 3 |  | 18 |  |
| 4 |  | 19 |  |
| 5 |  | 20 |  |
| 6 |  | 21 |  |
| 7 |  | 22 |  |
| 8 |  | 23 |  |
| 9 |  | 24 |  |
| 10 |  | 25 |  |
| 11 |  | 26 |  |
| 12 |  | 27 |  |
| 13 |  | 28 |  |
| 14 |  | 29 |  |
| 15 |  | 30 |  |

## **2.7.4. Содержание отчета**

1. Титульный лист
2. Название и цель практического занятия
3. Общее задание
4. Графический интерфейс пользователя
5. Сценарии, вначале которых должна быть введена информация в виде комментариев:

* имя и назначение сценария;
* вариант индивидуального задания и номер задания.

1. Протокол сессии Командного окна, вначале которой должна быть введена информация в виде комментариев:

* название практического занятия;
* вариант индивидуального задания и номер задания;
* ФИО студента, номер группы;
* краткое перечисление действий, выполняемых во время сессии.

## **2.7.5. Контрольные вопросы по теме**

1. Как формулируется постановка численного решения задачи оптимизации нелинейных функций?
2. Что называется локальным и глобальным минимумом функции?
3. Что является безусловной и условной оптимизацией нелинейных функций?
4. Что является необходимыми и достаточными условиями существования минимума функции от одной переменной?
5. Можно ли средствами Scilab вычислить глобальный минимум заданной функции?
6. Какие исследования необходимо произвести перед вычислением значения локального минимума функции от одной переменной?
7. Что называют в Scilab решателем и в чем его отличие от функции?
8. Для чего при использовании решателя **optim** необходимо формирование вспомогательной функции **costf**?
9. Каково назначение функции **numderivative** при решении задачи одномерной оптимизации?
10. Можно ли с использованием функции **optim** вычислить локальный максимум?
11. Что служит результатом выполнения функции **optim**?
12. Что является достаточным условием существования минимума многомерной функции?
13. Какие исследования целевой функции целесообразно произвести для выбора начального приближения к точке минимума многомерной функции?
14. Какие средства Scilab известны для решения задач оптимизации функций?
15. Что является входными и выходными параметрами решателя **optim**?
16. Можно ли использовать решатель **optim** без выходных параметров?
17. Каково назначение функции **costf***,*а также ее входных и выходных параметров при решении задачи многомерной оптимизации?
18. Что представляет собой формат обращения к решателю **fminsearch** при решении задачи многомерной оптимизации?

Список литературы

1. В.Н.Шакин, Т.И.Семенова, В.В.Фриск Базовые средства   
   математического пакета Scilab. Учебник/ -М.: ……, 2019. -315с.
2. Семенова Т.И., Шакин В.Н., Математический пакет Scilab: учебное пособие для бакалавров. -М.: ЭБС МТУСИ, 2017.-127 с. Режим доступа http://www.mtuci.ru/structure/library/catalogue/download.php?book\_id=1834
3. Семенова Т.И., Юскова И.Б., Юсков И.О. Проведение расчетов в среде пакета Scilab: Практикум.-М.:ЭБС МТУСИ, 2018.-35с. Режим доступа: http://www.mtuci.ru/structure/library/catalogue/download.php?book\_id=1963
4. Сборник заданий по работе и программированию в системе SciLab: методические указания к лабораторным работам по дисциплине «Информатика» для бакалавров » / сост. А.Е.Усачев.–Ульяновск: УлГТУ, 2015.–84с.
5. Семенова Т.И., Загвоздкина А.В., Загвоздкин В.А. Графическое представление результатов расчетов в Scilab. 2018 г. В сборнике «Материалы II Международной научно-практической конференции» г.Донецк, Донецкая Народная Республика, 2018г, —4-10 с.
6. Документация Scilab – Режим доступа: https://help.scilab.org/docs/6.0.1/ru\_RU/index.html
7. Семенова Т.И., Кравченко О.М., Шакин В.Н. Вычислительные модели и алгоритмы решения задач численными методами. Учетное пособие / МТУСИ. –М., 2017. - 84с. Режим доступа: http://www.mtuci.ru/structure/library/catalogue/download.php?book\_id=1819
8. Семенова Т.И., Юсков И.О., Юскова И.Б., Алгоритмизация вычислительных задач: Электронное учебное пособие. -М.:ЭБС МТУСИ, 2017.- 64 с. Режим доступа:

http://www.mtuci.ru/structure/library/catalogue/download.php?book\_id=1833

1. Семенова Т.И., Загвоздкина А.В., Загвоздкин В.А. Изучение численных методов с использованием средств пакета Scilab //Экономика и качество систем связи, 2017, №4 (6), С.60-69.

**Оглавление**

[**1. Основы работы с математическим пакетом Scilab** 3](#_Toc9087743)

[**1.1. Практическое занятие по теме «Рабочая среда Scilab и простейшие вычисления»** 3](#_Toc9087744)

[**1.1.1.** **Элементы системы Scilab, подлежащие изучению** 3](#_Toc9087745)

[**1.1.2.** **Общее задание** 3](#_Toc9087746)

[**1.1.3.** **Варианты индивидуальных заданий** 4](#_Toc9087747)

[**1.1.4.** **Пример выполнения задания** 5](#_Toc9087748)

[**1.1.5.** **Содержание отчета** 7](#_Toc9087749)

[**1.1.6.** **Контрольные вопросы по теме** 7](#_Toc9087750)

[**1.2** **Практическое занятие по теме «Основные объекты системы Scilab»** 9](#_Toc9087751)

[**1.2.1.** **Элементы системы Scilab, подлежащие изучению** 9](#_Toc9087752)

[**1.2.2.** **Общее задание** 9](#_Toc9087753)

[**1.2.3.** **Варианты индивидуальных заданий** 10](#_Toc9087754)

[**1.2.4.** **Пример выполнения задания** 12](#_Toc9087755)

[**1.2.5.** **Содержание отчета** 14](#_Toc9087756)

[**1.2.6.** **Контрольные вопросы по теме** 15](#_Toc9087757)

[**1.3 Практическое занятие по теме «Матричные операции создания, доступа, извлечения и модификации»** 16](#_Toc9087758)

[**1.3.1 Вопросы, подлежащие изучению** 16](#_Toc9087759)

[**1.3.2 Общее задание** 16](#_Toc9087760)

[**1.3.3 Варианты индивидуальных заданий** 17](#_Toc9087761)

[**1.3.4 Примеры выполнения заданий** 19](#_Toc9087762)

[**1.3.5. Содержание отчета** 22](#_Toc9087763)

[**1.3.6. Контрольные вопросы по теме** 22](#_Toc9087764)

[**1.4.** **Практическое занятие по теме**](#_Toc9087765)  [**«Визуализация результатов вычислений»** 24](#_Toc9087766)

[**1.4.1.** **Вопросы, подлежащие изучению** 24](#_Toc9087767)

[**1.4.2.** **Общее задание** 24](#_Toc9087768)

[**1.4.3.** **Варианты индивидуальных заданий** 25](#_Toc9087769)

[**1.4.4.** **Пример выполнения задания** 27](#_Toc9087770)

[**1.4.5.** **Содержание отчета** 31](#_Toc9087771)

[**1.4.6.** **Контрольные вопросы по теме** 31](#_Toc9087772)

[**1.5.** **Практическое занятие по теме** 33](#_Toc9087773)

[**«Средства программирования в Scilab»** 33](#_Toc9087774)

[**1.5.1.** **Вопросы, подлежащие изучению** 33](#_Toc9087775)

[**1.5.2.1 Общее задание №1**](#_Toc9087776)  [**«Программирование алгоритмов**](#_Toc9087777)

[**итеративных циклических структур»** 33](#_Toc9087778)

[1.5.2.2 Варианты индивидуальных заданий 34](#_Toc9087779)

[1.5.2.3 Пример выполнения задания 37](#_Toc9087780)

[**1.5.2.4** **Содержание отчета** 41](#_Toc9087781)

[**1.5.3.1 Общее задание №2**](#_Toc9087782)  [**«Программирование алгоритмов** 42](#_Toc9087783)

[**регулярных циклических структур»** 42](#_Toc9087784)

[**1.5.3.2** **Варианты индивидуальных заданий** 43](#_Toc9087785)

[**1.5.3.3** **Пример выполнения задания** 45](#_Toc9087786)

[**1.5.3.4 Содержание отчета** 49](#_Toc9087787)

[**1.5.4.1 Общее задание №3**](#_Toc9087788)  [**«Программирование алгоритмов обработки матриц»** 50](#_Toc9087789)

[**1.5.4.2** **Варианты индивидуальных заданий** 50](#_Toc9087790)

[**1.5.4.3** **Пример выполнения задания** 53](#_Toc9087791)

[**1.5.4.4** **Содержание отчета** 57](#_Toc9087792)

[**1.5.4.5** **Контрольные вопросы по теме** 58](#_Toc9087793)

[Раздел 2. Решение вычислительных задач средствами Scilab 59](#_Toc9087794)

[**2.1. Практическое занятие по теме**](#_Toc9087795)  [**«Использование полиномов»** 59](#_Toc9087796)

[**2.1.1. Вопросы, подлежащие изучению** 59](#_Toc9087797)

[**2.1.2. Общее задание** 59](#_Toc9087798)

[**2.1.3. Варианты заданий** 60](#_Toc9087799)

[**2.1.4. Содержание отчета** 60](#_Toc9087800)

[**2.1.5. Контрольные вопросы по теме** 61](#_Toc9087801)

[**2.2. Практическое занятие по теме**](#_Toc9087802)  [**«Аппроксимация и интерполяция функций»** 62](#_Toc9087803)

[**2.2.1. Вопросы, подлежащие изучению** 62](#_Toc9087804)

[**2.2.2. Общее задание** 62](#_Toc9087805)

[**2.2.3. Варианты заданий** 63](#_Toc9087806)

[**2.2.4. Содержание отчета** 64](#_Toc9087810)

[**2.2.5 Контрольные вопросы по теме** 64](#_Toc9087811)

[**2.3. Практическое занятие по теме**](#_Toc9087812)  [**«Интегрирование и дифференцирование в среде Scilab»** 66](#_Toc9087813)

[**2.3.1. Вопросы, подлежащие изучению** 66](#_Toc9087814)

[**2.3.2. Общее задание** 66](#_Toc9087815)

[**2.3.3. Варианты индивидуальных заданий** 67](#_Toc9087816)

**2.3.4 . Содержание отчета** ……………...…………………………………………….……68

[**2.3.5. Контрольные вопросы по теме**……………………………………………………68](#_Toc9087817)

[**2.4. Практическое занятие по теме**](#_Toc9087818) [**«Решение нелинейных уравнений»** 69](#_Toc9087819)

[**2.4.1. Вопросы, подлежащие изучению** 69](#_Toc9087820)

[**2.4.2. Общее задание** 69](#_Toc9087821)

[**2.4.3. Варианты индивидуальных заданий** 70](#_Toc9087822)

[**2.4.4. Содержание отчета** 71](#_Toc9087823)

[**2.4.5. Контрольные вопросы по теме** 71](#_Toc9087824)

[**2.5. Практическое занятие по теме**](#_Toc9087825)  [**«Решение обыкновенных**](#_Toc9087826)

[**дифференциальных уравнений»** 72](#_Toc9087827)

[**2.5.1**. **Вопросы, подлежащие изучению** 72](#_Toc9087828)

[**2.5.2. Общее задание** 72](#_Toc9087829)

[**2.5.3. Варианты индивидуальных заданий** 73](#_Toc9087830)

[**2.5.4. Содержание отчета** 75](#_Toc9087831)

[**2.5.5. Контрольные вопросы по теме** 76](#_Toc9087832)

[**2.6. Практическое занятие по теме**](#_Toc9087833)  [**«Решение систем линейных уравнений»** 77](#_Toc9087834)

[**2.6.1. Вопросы, подлежащие изучению** 77](#_Toc9087835)

[**2.6.2. Общее задание** 77](#_Toc9087836)

[**2.6.3. Варианты индивидуальных заданий** 78](#_Toc9087837)

[**2.6.4. Содержание отчета** 79](#_Toc9087838)

[**2.6.5. Контрольные вопросы по теме** 79](#_Toc9087839)

[**2.7. Практическое занятие по теме**](#_Toc9087840)  [«Решение задач одномерной и многомерной](#_Toc9087841)

[оптимизации» 80](#_Toc9087842)

[**2.7.1. Вопросы, подлежащие изучению** 80](#_Toc9087843)

[**2.7.2. Общее задание** 80](#_Toc9087844)

[**2.7.3. Варианты индивидуальных заданий** 81](#_Toc9087845)

[**2.7.4. Содержание отчета** 82](#_Toc9087846)

[**2.7.5. Контрольные вопросы по теме** 82](#_Toc9087847)