

Министерство образования Республики Беларусь

**Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»**

Кафедра «Промышленная электроника»

Э. М. Виноградов, Ю. В. Крышнев, М. В. Столбов

ОСНОВЫ ТЕОРИИ СИСТЕМ

ПРАКТИКУМ

**по выполнению лабораторных работ для студентов
специальности 1-53 01 07 «Информационные технологии
и управление в технических системах»
дневной формы обучения**

Электронный аналог печатного издания

Гомель 2017

УДК 681(075.8)
ББК 32.81я73
В49

*Рекомендовано к изданию научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 9 от 28.04.2016 г.)*

Рецензент: доц. каф. «Информационные технологии» ГГТУ им. П. О. Сухого
канд. физ.-мат. наук, доц. *А. В. Цитринов*

Виноградов, Э. М.

В49

Основы теории систем : практикум по выполнению лаборатор. работ для студентов специальности 1-53 01 07 «Информационные технологии и управление в технических системах» днев. формы обучения / Э. М. Виноградов, Ю. В. Крышнев, М. В. Столбов. – Гомель : ГГТУ им. П. О. Сухого, 2017. – 68 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

ISBN 978-985-535-343-1.

Содержит пять лабораторных работ, в которых приведены порядок их выполнения, основные теоретические сведения, задания для самостоятельной работы и контрольные вопросы.

Для студентов специальности 1-53 01 07 «Информационные технологии и управление в технических системах» дневной формы обучения.

**УДК 681(075.8)
ББК 32.81я73**

ISBN 978-985-535-343-1

© Виноградов Э. М., Крышнев Ю. В.,
Столбов М. В., 2017

© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2017

ВВЕДЕНИЕ

Понятие «система» является одним из ключевых определений в общеобразовательной программе подготовки инженеров по информационным технологиям и управлению. Рассматривая систему как совокупность взаимодействующих объектов, можно широко использовать преимущества структурного отображения исследуемых явлений и процессов для их математического описания и моделирования на компьютерах.

Современные вычислительные средства позволяют без особого труда и затрат времени решать сложные задачи по исследованию и расчету систем управления не инженерными (адаптированными) методами, а с использованием математических аппаратов любой степени сложности. При этом не требуется помощь программиста для реализации методов и визуализации проводимых исследований.

Примером могут служить программные пакеты MathCad, Mathematica, Maple, Matlab, которые используются студентами, учеными и инженерами во всем мире. Названные коммерческие программы универсальны и рассчитаны на любых пользователей. К сожалению, они имеют существенный недостаток – слишком большая цена за одну версию программы. Однако существуют и свободно распространяемые (некоммерческие) программные пакеты для подобных целей. Примером может служить программный пакет Scilab.

Свободно распространяемый программный пакет Scilab был разработан во Франции, в Национальном исследовательском институте информатики и автоматизации INRIA. В настоящее время он распространяется через Интернет. Последнюю версию пакета всегда можно скачать на официальном сайте программы www.scilab.org.

Scilab позволяет производить численные вычисления с различными функциями, имеет средства для построения графиков. В состав пакета входит инструмент для визуального исследования различных устройств. Scilab имеет свой язык программирования, встроенный редактор текста и средства для отладки программ. Поэтому, используя современную терминологию, можно охарактеризовать Scilab как интегрированную среду разработки для выполнения инженерных расчетов и научных исследований.

Лабораторная работа № 1

НАЧАЛО РАБОТЫ В СРЕДЕ РАЗРАБОТКИ SCILAB

1. Цель работы

Ознакомиться с интерфейсом интегрированной среды разработки Scilab. Исследовать работу Scilab в режиме калькулятора. Изучить и исследовать выполнение математических вычислений со скалярами. Научиться создавать простейшие скрипт-файлы.

2. Порядок выполнения работы

2.1. Создание папки для работы

2.1.1. Для выполнения лабораторных работ Вам необходимо создать на компьютере свою рабочую папку. Сделать это можно следующим образом.

На панели TotCom выберите диск F. Затем выберите папку OTC и раскройте ее. Внутри нее создайте новую папку (с помощью клавиши F7) с именем, соответствующим Вашей фамилии (буквы обязательно латинские), например: Ivanov.

Примечание. При использовании в именах папок символов кириллицы возможна неправильная работа исследуемых программ.

В дальнейшем Вы будете записывать и хранить в Вашей папке все файлы в процессе выполнения лабораторных работ. Полный путь к ней:

F:\OTC\Ivanov

2.1.2. При выполнении лабораторных работ Вы будете создавать много различных файлов. Чтобы было легче ориентироваться в них, желательно для каждой лабораторной работы иметь свою отдельную папку. Допустим, что для выполнения лабораторной работы № 1 мы будем использовать папку с именем Lab1, которую необходимо предварительно создать. С этой целью откройте Вашу рабочую папку (с именем, соответствующим Вашей фамилии) и создайте в ней (с помощью клавиши F7) новую папку с именем Lab1.

В дальнейшем Вы будете записывать и хранить в этой папке все файлы в процессе работы с Matlab. Полный путь к ней:

F:\OTC\Ivanov\Lab1

2.2. Запуск и настройка среды разработки Scilab

2.2.1. Запустить Scilab можно щелчком по кнопке мыши, поместив курсор на ярлык Scilab на рабочем столе компьютера. После запуска программы Вы увидите на экране рабочий стол среды разработки Scilab, который имеет следующие основные окна:

1) **обозреватель файлов**, в котором отображается список файлов и вложенных папок активного в данный момент каталога;

2) **командное окно**, является основным окном, в котором вводятся исполняемые команды и отображаются результаты вычислений;

3) **обозреватель переменных** служит для отображения переменных, которые используются в данном сеансе работы;

4) **журнал команд** используется для отображения содержимого буфера, в котором хранятся выполненные ранее команды.

2.2.2. После запуска Scilab необходимо разместить на экране дисплея слева текст данной лабораторной работы, а справа рабочий стол среды разработки Scilab.

2.2.3. Необходимо установить в качестве активного (текущего) каталог (папку) Lab1. С этой целью подведите курсор мыши к иконке слева от строки со списком папок. Появится подсказка: «Выберите папку». Щелкните кнопкой мыши по иконке. Раскроется окно с заголовком «Выберите папку». Сначала выберите диск F, затем папку с именем ОТС, потом папку с Вашей фамилией (Ivanov) и затем папку Lab1. В строке Folder name должно быть следующее: F:\ОТС\Ivanov\Lab1. Если это так, то щелкните по кнопке Open.

В результате этих действий в строке обозревателя файлов должно появиться:

F:\ОТС\Ivanov\Lab1

2.2.4. При необходимости очистите командное окно и журнал команд, используя пункты меню Правка.

2.3. Исследование работы Scilab в режиме калькулятора

2.3.1. **Задание.** Наберите в командном окне следующие выражения. Для выполнения нажмите клавишу Enter:

1)

--> 7 + 8/2

2)

--> 27 ^ (1/3) + 32 ^ 0.2

3) если вычисляемое выражение слишком длинное, то перед нажатием клавиши Enter следует набрать две или более точек. Это будет означать продолжение в командной строке:

```
--> 1 + 2 + 3 + 4 + ..
```

```
--> 5 + 6 + 7
```

4) текстовый комментарий в Scilab – это строка, начинающаяся с символов // (как в языке Си):

```
--> // This is my comment
```

```
--> // Это мой комментарий
```

5) если символ точки с запятой «;» указан в конце выражения, то результат вычислений не выводится, а активизируется следующая командная строка:

```
--> 1 + 2;
```

```
--> 1 + 2
```

2.3.2. Задание. Требуется подсчитать выражение:

$$\left(3^{-1} \cdot \left(\frac{1}{2} \right)^{-2} - 27^{-\frac{1}{3}} \right) \cdot \left(3 \frac{2}{5} - 0,9 \right)$$

Наберите в командной строке соответствующее выражение (по правилам Scilab) и вычислите его.

2.4. Переменные в Scilab

В среде разработки Scilab любая переменная до использования в формулах и выражениях должна быть определена. Для определения переменной необходимо набрать имя переменной, потом символ «=», и затем значение переменной. Здесь знак равенства – это оператор присваивания, т. е. если в общем виде оператор присваивания записать как

имя_переменной = значение_выражения,

то в переменную, имя которой указано слева, будет записано значение выражения, указанного справа. Выражение в правой части оператора присваивания может быть числом, арифметическим выражением, строкой символов или символьным выражением.

2.4.1. **Задание.** Выполните следующие примеры определения переменных.

1) присваивание значений переменным x и y :

```
--> x = 2.3
```

```
--> y = -34.7
```

Обратите внимание, что имена переменных x и y , а также их значения появились в окне обозревателя переменных на рабочем столе Scilab;

2) для просмотра значения любой переменной из текущего рабочего пространства можно набрать ее имя и нажать клавишу Enter. Проверим, существуют ли в рабочем пространстве в данной сессии (сеансе работы) переменные с именами x , y и z :

```
--> x
```

```
--> y
```

```
--> z
```

Обратите внимание на сообщении об ошибке – переменная z не определена;

3) определим переменные u и v , присвоим им значения, и вычислим значение переменной w :

```
--> u = 1; v = 2;
```

```
--> w = (u + v) - u / v
```

Обратите внимание, что имя переменной w появилось в окне обозревателя переменных.

2.4.2. Когда отпадает необходимость в хранении ряда переменных в текущем сеансе работы, их можно удалить из памяти компьютера. Для очистки рабочего пространства используется команда `clear` в разных формах, например:

`clear x` – уничтожение переменной с именем x ;

`clear a b c` – уничтожение переменных с именами a , b , c ;

`clear` – уничтожение всех переменных.

Задание. Выполните команду `clear` в различных формах. Обратите внимание, как имена переменных исчезают из окна обозревателя переменных:

1) удалить переменную y :

```
--> clear y;
```

2) удалить переменные x и z :

```
--> clear x z;
```

3) очистить текущее рабочее пространство:

```
--> clear;
```

4) проверим, имеются ли в рабочем пространстве переменные с именами:

```
--> x
```

```
--> u
```

Сделайте выводы по результатам выполнения команд.

2.4.3. Среда разработки Scilab имеет несколько predefined переменных и констант, которые называются системными.

Основные системные переменные и константы:

ans – (от англ. answer – ответ) – результат вычисления, имя которого не определено пользователем;

$\%e$ – число e (основание натурального логарифма);

$\%pi$ – число π ;

$\%i$ – мнимая единица ($\sqrt{-1}$);

$\%eps$ – условный нуль, равный 2.220D-16;

$\%inf$ – «бесконечность»;

$\%nan$ – «не число» (от англ. not a number);

$\%t$ или $\%T$ – логическая единица, «истина» (от англ. true);

$\%f$ или $\%F$ – логический нуль, «ложь» (от англ. false).

Задание. Исследуйте системную переменную (константу) $\%pi$:

1) введите в командную строку;

```
--> %pi
```

2) вычислите выражение:

```
--> y = 2 * %pi / 6
```

3) вычислите выражение:

```
--> z = pi / 8
```

Сделайте выводы по использованию системной константы $\%pi$ в выражениях.

2.5. Управление форматом отображения чисел

2.5.1. По умолчанию в Scilab в качестве результата выводится только 10 символов цифр (включая знак числа и десятичную точку).

Задание. Выведите значение системной константы `%pi`:

```
--> %pi
```

Подсчитайте количество значащих цифр.

2.5.2. Можно изменить формат отображения выводимых чисел с помощью команды `format(n)`, где `n` – количество выводимых символов.

Задание. Выполните вывод числа π с 20 значащими цифрами:

```
--> format(20)
```

```
--> %pi
```

2.5.3. Заданный в команде `format(n)` вид отображения чисел сохраняется до окончания текущей сессии (сеанса работы). Для возврата к первоначальному формату необходимо исполнить команду `format(10)`.

Задание. Проверьте формат вывода числа e , а затем вернитесь к формату вывода по умолчанию:

```
--> %e
```

```
--> format(10)
```

```
--> %pi
```

2.6. Выполнение математических вычислений со скалярами

Таблица 1

Основные элементарные математические функции Scilab

Запись в Scilab	Запись в математике
<code>sqrt(x)</code>	\sqrt{x} – корень квадратный из x
<code>abs(x)</code>	$ x $ – абсолютная величина
<code>exp(x)</code>	e^x – экспонента
<code>log(x)</code>	$\ln x$ – логарифм натуральный
<code>log10(x)</code>	$\lg x$ – логарифм десятичный
<code>cos(x), sin(x), tan(x), cot(x)</code> , x – в радианах	$\cos x, \sin x, \operatorname{tg} x, \operatorname{ctg} x$

Запись в Scilab	Запись в математике
cosd(x), sind(x), tand(x), cotd(x), x – в градусах	
acos(x), asin(x), tan(x), acot(x) – x в радианах	arccos x, arcsin x, arctg x, arcctg x
acosd(x), asind(x), atand(x), acotd(x) – x в градусах	
round(x)	Округляет x до ближайшего целого
factorial(x)	x! – выдает факториал числа (x – целое, положительное)

2.6.1. **Задание.** Для вычисления выражения:

$$z = \sin\left(\frac{25\pi}{6}\right) - \cos\left(-\frac{\pi}{3}\right)$$

наберите в командной строке и нажмите Enter:

$$z = \sin(25 * \%pi / 6) - \cos(- \%pi / 3)$$

2.6.2. **Задание.** Вычислите выражение:

$$y = \frac{\cos x^2}{x + \sin^3 x} + e^{-2,1}$$

при значении $x = 1,46$.

2.6.3. **Задание.** Вычислите выражение:

$$w = \frac{1,2 - 2x}{\lg(x + 3,2)} - \sqrt{|x - 5,8|}$$

при значении $x = 0,478$.

2.7. Операции с комплексными числами

В Scilab системная переменная %i является мнимой единицей $\sqrt{-1}$.

Комплексное число z можно записать в алгебраической форме:

$$z = \operatorname{Re}(z) + \operatorname{Im}(z)i$$

или в экспоненциальной форме:

$$z = Ze^{i \arg(z)},$$

где $\text{Re}(z)$ – действительная часть числа; $\text{Im}(z)$ – комплексная часть числа; Z – модуль числа; $\text{arg}(z)$ – аргумент числа, который определяется соотношением:

$$\text{arg}(z) = \text{arctg} \frac{\text{Im}(z)}{\text{Re}(z)}.$$

Для записи комплексного числа в алгебраической форме надо ввести в командную строку:

--> $x = 5 + 6 * \%i$

или

--> $y = 10 - \%i * 5$

Таблица 2

Функции комплексного переменного в Scilab

Функция	Описание
$\text{abs}(x)$	Вычисляет модуль комплексного числа x
$\text{conj}(x)$	Возвращает число, комплексно-сопряженное с x
$\text{imag}(x)$	Выдает мнимую часть числа x
$\text{real}(x)$	Выдает действительную часть числа

2.7.1. **Задание.** Выполните следующие примеры работы с комплексными числами:

1) значения переменной i :

--> $\%i$

.....

--> i

.....

2) задайте комплексные числа: $a = 1 + i$, $b = 2 - 3i$

.....

3) вычислите сумму комплексных чисел: $a + b$

.....

4) вычислите произведение комплексных чисел: $a * b$

.....

5) вычислите модуль комплексного числа a :

.....

6) вычислите действительную и мнимую части комплексного числа: $\operatorname{Re}(b)$ и $\operatorname{Im}(b)$:

.....

.....

7) вычислите аргумент комплексного числа b (через \arctg):

.....

8) вычислите число, комплексно сопряженное числу b :

.....

9) вычислите синус комплексного числа a .

.....

2.8. Ведение дневника сессии

В Scilab имеется возможность сохранить в файле всего текста сессии (содержимого командного окна), т. е. сеанса работы в Scilab. Для этого используются специальные команды для ведения дневника сессии:

`diary('file_name')` – ведет запись на диск всех команд в строках ввода и полученных результатов в виде текстового файла с указанием имени. Файл с именем `file_name` – это обычный текстовый файл (не исполняемый). Обычно используется встроенный в Windows редактор типа Notepad (Блокнот). Для этого файл должен иметь расширение `.txt`. Запись файла ведется в папку (директорию), которая в данный момент является текущей;

`diary(0)` – прекращает запись в файл.

2.8.1. Задание. Выполните сохранение нескольких команд сессии в текстовом файле с именем `primer.txt` в папке `Lab1`, которая в данный момент является текущей. Для этого выполните следующие команды:

```
--> diary(primer.txt)
```

```
--> x = 1 + 2
```

```
--> y = x * 5
```

```
diary(0)
```

Обратите внимание, что после выполнения команды `diary('primer.txt')` в командном окне появляется сообщение `ans = 1`. Это значение идентификатора `id`, которое программа присвоила файлу `primer.txt`.

Чтобы прочитать содержимое дневника, щелкните мышью по надписи `Lab1` в окне обозревателя файлов. Должно появиться название файла дневника `primer.txt`. Щелкните дважды по этому имени. Должно раскрыться окно блокнота с текстом – содержимым командного окна Scilab.

Убедитесь, что все заданные команды были записаны.

2.8.2. Иногда требуется записывать в дневник не все команды сессии, а только некоторые из них. Для управления процессом записи в текстовый файл Scilab имеет специальные команды вида:

`diary(id, 'off')` или `diary('file_name', 'off')` – приостанавливают запись в файл;

`diary(id, 'on')` или `diary('file_name', 'on')` – вновь начинает запись в файл.

Здесь параметр `id` – это идентификатор файла `file_name`.

Таким образом, чередуя команды `diary('off')` и `diary('on')`, можно сохранять нужные фрагменты сессии в их формальном виде.

Задание. Выполните сохранение нескольких команд сессии в текстовом файле с именем `primer_2.txt` в папке `Lab1`, которая в данный момент является текущей. Для этого выполните следующие команды:

```
--> diary(primer_2.txt)
```

```
--> x = 4 + 6
```

```
--> diary(1, off)
```

```
--> y = x * 7
```

```
--> diary(1, on)
```

```
--> z = x/25
```

```
--> diary(0)
```

Здесь параметр `1` – это номер идентификатора, который программа присвоила файлу `primer_2.txt` при запуске дневника.

Чтобы прочитать содержимое дневника, щелкните мышью по надписи `Lab1` в окне обозревателя файлов. Должно появиться название файла дневника `primer_2.txt`. Щелкните дважды по этому имени.

Должно раскрыться окно блокнота с текстом – содержимым командного окна Scilab.

Убедитесь, что команда $y = x * 7$ не была записана в дневник.

Полученные текстовые файлы с расширением .txt можно прочитать с помощью текстового редактора Windows (выделив его на панели TotCom, а затем нажав клавишу F3, либо двойным щелчком мыши по его названию). Эти файлы можно распечатать или встраивать в различные документы.

2.9. Создание скрипт-файлов

Скрипт-файл (в буквальном переводе файл-сценарий) – это список команд Scilab, сохраненный на диске. Для сокращения обычно эти файлы называют просто скриптами. Для подготовки, редактирования и отладки скриптов служит специальный редактор среды Scilab, называемый SciNotes (в более ранних версиях он назывался SciPad). Редактора можно вызвать из главного меню рабочего стола Scilab, выполнив команду: Инструменты => Текстовый редактор Scilab. В результате работы этой команды будет создан новый скрипт-файл. По умолчанию он имеет имя «Безымянный документ 1».

Окно редактора скрипт-файлов выглядит стандартно, т. е. имеет заголовок, меню, панель инструментов.

Ввод текста в окно редактора скрипт-файла осуществляется по правилам, принятым для команд Scilab.

2.9.1. Задание. Вызовите редактор SciNotes с помощью команды главного меню среды Scilab: Инструменты => Текстовый редактор Scilab. В окне редактора наберите текст скрипта для решения квадратного уравнения $ax^2 + bx + c = 0$.

Текст скрипта:

```
// Решение уравнения  $ax^2 + bx + c = 0$ 
```

```
a = 1; b = 5; c = 4;
```

```
d = b ^ 2 - 4 * a * c;
```

```
x1 = (-b + sqrt(d)) / (2 * a)
```

```
x2 = (-b - sqrt(d)) / (2 * a)
```

Нетрудно заметить, что точка с запятой «;» ставится после тех команд, которые не требуют вывода значений.

Для сохранения введенной информации необходимо выполнить команду: Файл => Сохранить как... из меню редактора. Открывается

окно с заголовком Сохранить как... . Если открыта папка Lab1, то в строке Filename нужно ввести имя сохраняемого скрипт-файла, например, roots (от англ. корни). Расширение указывать не нужно, оно добавляется автоматически. Щелчок по кнопке Save приведет к сохранению информации, находящейся в окне редактора. Скрипт-файлы сохраняются с расширением .sce.

Проверить, что скрипт-файл с именем roots.sce появился в папке Lab1, можно щелкнув мышью по строке Lab1 в дереве папок в окне обозревателя файлов.

2.9.2. Запуск (исполнение) скрипт-файла может быть выполнен в Scilab двумя основными способами:

1) скрипт-файл может быть запущен непосредственно из окна редактора SciNotes. С этой целью выполните команды из меню редактора: Выполнить => ...файл с отображением команд. В командном окне появятся строки с командами скрипта и результаты вычислений (в данном примере значения корней уравнения x_1 и x_2);

2) скрипт-файл может быть запущен из командного окна. Для этого надо выполнить команды из главного меню: Файл => Выполнить. При этом появляется окно для выбора файла с заголовком Select a file to execute. Нужно выбрать из папки Lab1 файл с именем roots (щелкнув по нему кнопкой мыши), файл появится в строке File name. Затем надо щелкнуть по кнопке Open.

В командном окне появятся значения переменных x_1 и x_2 (результаты вычислений в скрипте). В конце будет выведено сообщение Execution done (исполнение выполнено).

Задание. Запустите скрипт-файл roots.sce двумя способами:

1) из окна редактора SciNotes. После появления результатов в командном окне закройте окно редактора;

2) из командного окна Scilab.

3. Задания для самостоятельной работы

Указание. Отчет по лабораторной работе № 1 выполняется в виде дневника, т.е. содержимого командного окна Scilab в ходе выполнения заданий для самостоятельной работы. Файл дневника создается в папке Lab1, рекомендуемое имя для файла дневника – self_work.txt. Каждое задание должно в начале иметь строку комментария, например, вида:

// Задание 1

Тексты заданий приводить не нужно.

Задание 1. Вычислите следующие выражения:

1) e^{14} и 38280π , каждый результат с точностью до 20 знаков.
Какое из значений больше? (*Подсказка:* предварительно задайте необходимый формат для отображения результата)

2) $\ln 2$

3) $\operatorname{arctg} 0,5$

Примечание. После выполнения заданных вычислений нужно вернуться к формату (по умолчанию) отображения с 10 символами.

Задание 2. Вычислите выражение:

$$\left(\frac{1}{3}\right)^3 \cdot e^{a+b} + \frac{\sqrt{15 - kx^2 - 0,41}}{10^{-2} \cdot |a+b|} + \frac{\ln(a+b)^2}{a+kx^2} - \sqrt[3]{3}$$

при значениях: $a = 2,5$; $b = -5,25$; $x = 1,25$; $k = 4$.

Задание 3. Вычислите выражение:

$$z = \left(\frac{2}{9}\right)^2 + \frac{\sqrt[4]{x^2 + b}}{0,4x} - 10^4 e^{kx} + \cos \sqrt{x^3 + b} + \frac{\sin 3}{(x^2 + b) \cdot n}$$

при значениях: $x = 2,5$; $b = 0,04$; $k = 4$; $n = 5$.

Задание 4. Дано тригонометрическое тождество:

$$\sin^3 x = \frac{1}{4}(3 \sin x - \sin 3x)$$

Проверьте правильность тождества путем вычисления по отдельности левой и правой части при значении $x = 12^\circ$.

Задание 5. Определите две переменные: $\alpha = 5\pi/8$ и $\beta = \pi/8$. Используя эти переменные, покажите, что следующее тригонометрическое тождество справедливо, вычисляя значения левой и правой части:

$$\sin \alpha \cdot \cos \beta = \frac{1}{2}[\sin(\alpha - \beta) + \sin(\alpha + \beta)]$$

Задание 6. Дано тригонометрическое тождество:

$$\operatorname{tg} 4\gamma = \frac{4\operatorname{tg} \gamma - 4\operatorname{tg}^3 \gamma}{1 - 6\operatorname{tg}^2 \gamma + \operatorname{tg}^4 \gamma}$$

Проверьте правильность тождества путем вычисления по отдельности его левой и правой части при $\gamma = 20^\circ$. (Подсказка: вместо греческой буквы γ используйте имя gamma).

Задание 7. Вычислите выражение:

$$\frac{4}{3} n^3 \cdot \sin^2 \frac{\alpha}{2} \cdot \sqrt[3]{\cos \alpha}$$

при $n = 1,7 \cdot 10^3$; $\alpha = 18^\circ$.

Задание 8. Выполните следующие действия с комплексными числами:

1) число z_1 , заданное в алгебраической форме $z_1 = 4 + 3i$, перевести в экспоненциальную $Ze^{i\varphi}$;

2) число z_2 , заданное в экспоненциальной форме $z_2 = 2,71e^{i\pi/12}$, перевести в алгебраическую;

3) вычислить выражение:

$$z_1^2 \cdot z_2 : z_3 + z_4,$$

где

$$z_3 = 1,82e^{-1,2i},$$

$$z_4 = \sqrt{3} - 2i.$$

Задание 9. Объект с начальной температурой T_0 , который помещается в начальное время $t = 0$ внутрь камеры с постоянной температурой T_s , будет изменять свою температуру по закону:

$$T = T_s + (T_0 - T_s) e^{-kt},$$

где T – температура объекта за время t ; k – константа.

Предположим, что бутылка напитка с начальной температурой 30°C помещается в холодильник, где температура 4°C . Определите с точностью до ближайшего целого числа температуру, которую будет иметь напиток через 3 ч. Константа $k = 0,45$. Сначала определите все переменные, а затем вычислите температуру.

Разработайте скрипт и сохраните его в файле с именем `zadanie9.sce` в Вашей папке `Lab1`. Запустите скрипт на исполнение из окна редактора с выводом текста скрипта и убедитесь в его работоспособности.

Задание 10. Снаряд запускается под углом θ и начальной скоростью V . Время полета снаряда до приземления обозначим как t_{travel} , максимальное расстояние до точки падения обозначим как x_{max} , и максимальная высота подъема как h_{max} . Эти величины определяются соотношениями:

$$t_{travel} = 2 \frac{V}{g} \sin \theta;$$

$$x_{max} = 2 \frac{V^2}{g} \sin \theta \cdot \cos \theta;$$

$$h_{max} = 2 \frac{V^2}{g} \sin^2 \theta,$$

где $g = 9,81 \text{ m/s}^2$ – ускорение свободного падения.

Рассмотрите случай, когда $V = 200 \text{ m/s}$ и $\theta = 54^\circ$. Определите V и θ как переменные (для греческой буквы θ используйте слово theta) и вычислите значения t_{travel} , x_{max} , h_{max} .

Разработайте скрипт и сохраните его в файле с именем `zadanie10.sce` в Вашей папке `Lab1`. Запустите скрипт на исполнение из окна редактора с выводом текста скрипта и убедитесь в его работоспособности.

Задание 11. Треугольник имеет стороны $a = 16 \text{ см}$; $b = 18 \text{ см}$; $c = 15 \text{ см}$ и углы α , β , γ .

Определите a , b , c как переменные и затем:

1) вычислите углы (в градусах) с использованием закона косинусов для треугольника:

$$c^2 + b^2 - 2 \cdot a \cdot b \cdot \cos \gamma;$$

$$a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos \alpha;$$

$$b^2 = a^2 + c^2 - 2 \cdot a \cdot c \cdot \cos \beta;$$

2) проверьте, что сумма углов равна 180° .

3) разработайте скрипт для решения задачи и сохраните его в файле с именем `zadanie11.sce` в папке `Lab1`. Запустите скрипт на исполнение с выводом текста скрипта и убедитесь в его работоспособности.

4. Содержание отчета

Отчет по лабораторной работе № 1 выполняется в виде дневника, т. е. содержимого командного окна Scilab в ходе выполнения заданий для самостоятельной работы.

Разработанный дневник необходимо продемонстрировать преподавателю для оценки объема выполненной работы и анализа качества выполнения заданий.

Контрольные вопросы

1. Какие системные переменные и константы существуют в Scilab?
2. Как приводятся текстовые комментарии?
3. Какие имеются основные тригонометрические функции в Scilab?
4. Как сообщается об ошибках в Scilab?
5. Как задается формат отображения чисел?
6. Что такое скрипт?
7. Как можно создать скрипт-файл?

Лабораторная работа № 2

СОЗДАНИЕ МАССИВОВ

1. Цель работы

Изучить способы создания массивов – векторов и матриц. Освоить методы индексации массивов. Исследовать встроенные функции для работы с массивами.

2. Порядок выполнения работы

2.1. Создание папки для работы

При выполнении лабораторной работы № 2 Вы будете использовать папку с именем Lab2, которую необходимо предварительно создать.

С этой целью откройте Вашу рабочую папку (с Вашей фамилией) и создайте в ней (с помощью клавиши F7) новую папку с именем Lab2.

В дальнейшем Вы будете записывать и хранить в этой папке все файлы при выполнении лабораторной работы № 2. Полный путь к этой папке будет такой:

F:\ОТС\Ivanov\Lab2

2.2. Запуск и настройка среды разработки Scilab

2.2.1. После запуска Scilab необходимо разместить на экране дисплея слева текст данной лабораторной работы, а справа – рабочий стол среды разработки Scilab.

2.2.2. Нужно установить в качестве активной (текущей) папку Lab2. С этой целью подведите курсор мыши к иконке слева от строки со списком папок. Появится подсказка «Выберите папку». Щелкните кнопкой мыши по иконке. Раскроется окно с заголовком «Выберите папку». Сначала выберите диск F, затем папку с именем ОТС, затем папку с Вашей фамилией (условно Ivanov) и затем папку Lab2. В строке Folder name должно быть следующее: F:\ОТС\Ivanov\Lab2. Если это так, то щелкните по кнопке Open.

В результате этих действий в строке обозревателя файлов должно появиться:

F:\ОТС\Ivanov\Lab2

2.2.3. При необходимости очистите командное окно и журнал команд, используя пункты меню Правка.

2.3. Создание векторов методом поэлементного ввода

2.3.1. Для создания вектора-строки нужно ввести его элементы с пробелами или запятыми между элементами внутри квадратных скобок.

Задание. Создайте вектор-строку следующими способами:

1) используя пробелы между элементами:

--> v1 = [1 2 3]

2) используя запятые между элементами:

--> v2 = [2, 1, -1]

2.3.2. Для создания вектора-столбца нужно ввести элементы с разделением их символом «;» или нажатием клавиши Enter после ввода каждого элемента.

Задание. Создайте вектор-столбец следующими способами:

1) используя символ «;» после ввода каждого элемента:

--> u1 = [2; 4; 5]

2) используя Enter после ввода каждого элемента:

```
--> u2 = [4
```

```
8
```

```
10]
```

В Scilab возможен ввод элементов векторов в виде арифметических выражений, содержащих любые доступные системе функции. В этом случае лучше для зрительного разделения элементов использовать символы «,» или «;».

Задание. Создайте вектор-строку вида:

```
--> v = [2 + 2/(3 + 4) exp(5) sqrt(10)]
```

2.4. Создание вектора с помощью оператора «:» – двоеточие

Для создания векторов с элементами, представляющими собой упорядоченную числовую последовательность с постоянным промежутком (разностью между значениями соседних элементов), в Scilab используется оператор «:» (двоеточие) в виде:

Начальное_значение : Шаг : Конечное_значение

Если шаг не задан, то по умолчанию он считается равным 1. Если шаг – величина положительная, то последовательность чисел возрастающая, если же шаг – величина отрицательная, то последовательность чисел – уменьшающаяся.

При создании векторов с помощью оператора «:» квадратные скобки при определении вектора можно не использовать.

Задание. Выполните примеры использования оператора «:» для векторов:

1)

```
--> 1:5
```

2)

```
--> i = 0:2:10
```

3)

```
--> j = [10:-2:2]
```

4)

```
--> w = 0 : %pi/2 : 2 * %pi
```

5)

```
--> y = 0 : 8
```

```
--> cos(y)
```

2.5. Создание матриц

Матрица – это двумерный массив размером $m \times n$, где m – количество строк, а n – количество столбцов. Обычно матрица создается путем ввода элементов строка за строкой внутри квадратных скобок. Элементами матриц могут быть числа или математические выражения, включающие числа, определенные ранее переменные и функции. Все строки должны иметь одно и то же количество элементов.

2.5.1. Задание. Создайте матрицу A размером 3×3 с разделением строк символом «;»

```
--> A = [5 35 43; 4 76 81; 21 32 40]
```

2.5.2. Задание. Создайте матрицу B размером 3×5 с разделением строк вводом Enter

```
--> B = [7 2 76 33 8
```

```
1 98 6 25 6
```

```
5 54 68 9 0]
```

2.5.3. Задание. Создайте матрицу C размером 2×3 . Для зрительного разделения элементов используйте символ «,»:

```
--> cd = 6; e = 3; h = 4;
```

```
--> C = [e, cd*h, cos(%pi/3); h^2, sqrt(h*e/cd), 14]
```

2.5.4. Задание. Создайте матрицу D размером 3×6 , в которой некоторые строки вводятся как векторы с элементами в виде числовых значений с постоянным шагом изменения:

```
--> D = [1 : 2 : 12; 0 : 5 : 25; 67 2 43 68 4 13]
```

2.6. Операция конкатенации массивов

Матрицы и векторы можно формировать, составляя их из ранее заданных матриц и векторов. Эта операция называется конкатенацией (объединением).

Задание. Исследуйте операции конкатенации векторов и матриц для заданных векторов:

$$\rightarrow v1 = [1 \ 2 \ 3]; \ v2 = [4 \ 5 \ 6]; \ v3 = [7 \ 8 \ 9];$$

1) горизонтальная конкатенация векторов-строк:

$$\rightarrow V = [v1 \ v2 \ v3]$$

2) вертикальная конкатенация векторов-строк:

$$\rightarrow V = [v1; \ v2; \ v3]$$

3) горизонтальная конкатенация матриц:

$$\rightarrow M = [V \ V \ V]$$

4) вертикальная конкатенация матриц:

$$\rightarrow M = [V; \ V]$$

2.7. Операция транспонирования

Операция транспонирования обозначается знаком «'» – апостроф. Она меняет строки матрицы с ее столбцами. Для квадратных матриц операция транспонирования имеет наглядную геометрическую интерпретацию: на своих местах остаются элементы главной диагонали, а остальные «отражаются симметрично» относительно этой диагонали. Вектор-строки операцией транспонирования преобразуются в вектор-столбцы и наоборот.

Задание. Исследуйте операцию транспонирования:

1) для матрицы A размером 2×3 :

$$\rightarrow A = [1 \ 2 \ 3; \ 4 \ 5 \ 6]$$

$$\rightarrow B = A'$$

2) для квадратной матрицы:

$$\rightarrow C = [1 \ 1 \ 1; \ 2 \ 2 \ 2; \ 3 \ 3 \ 3]$$

$$\rightarrow D = C'$$

3) для вектора-строки:

--> $v = [1 \ 2 \ 3]$

--> $u = v'$

4) для вектора-столбца:

--> $w = [1; 2; 3]$

--> $z = w'$

2.8. Адресация элементов массивов

Элементы в массиве (в векторе или в матрице) могут адресоваться индивидуально или в группах.

2.8.1. Адрес элемента вектора – это его позиция в строке (или в столбце), указанная в круглых скобках.

Задание. Исследуйте способы адресации для вектора:

--> $v = [35 \ 46 \ 78 \ 23 \ 5 \ 14 \ 81 \ 3 \ 55]$

1) значение 4-го элемента вектора:

--> $v(4)$

2) значение 7-го элемента вектора:

--> $v(7)$

3) использование элементов вектора в математическом выражении:

--> $v(2) * v(5) + \text{sqrt}(v(7))$

4) присвоение элементу вектора нового значения:

--> $v(1) = 77; v(5) = 99;$

--> v

2.8.2. Адрес элемента матрицы – это позиция, определяемая номером строки и номером столбца, где он расположен, указанная в круглых скобках.

Задание. Исследуйте способы адресации для матрицы:

--> $M = [3 \ 11 \ 6 \ 5; \ 4 \ 7 \ 10 \ 2; \ 13 \ 9 \ 0 \ 8]$

1) значения определенных элементов:

--> M(1,2)

--> M(3,4)

2) использование элементов матрицы в математическом выражении:

--> M(2,4) – M(1,2)

3) присвоение элементам матрицы новых значений:

--> M(1,2) = 35; M(2,4) = 66;

--> M

2.9. Применение оператора «:» – двоеточие для адресации массивов

Символ «:» (двоеточие) – может использоваться для адресации диапазона элементов в векторе или матрице.

Обычно при обращении к элементам матрицы отсчет индексов ведется от первой строки и первого столбца. Оператор “\$” позволяет адресовать элементы матрицы, отсчитывая от последней строки или столбца в зависимости от контекста, например:

A(k, \$) – элемент на пересечении k-й строки и последнего столбца;

A(\$, j) – элемент на пересечении последней строки и j-го столбца.

Задание. Исследуйте различные варианты адресации для доступа к группе элементов массивов:

1) создайте вектор:

--> v = [4 15 8 12 34 2 50 32 11]

и выделите из него элементы с 3-го по 7-й:

--> u = v(3:7)

2) создайте матрицу:

--> X = [3, 4, 8, 12; 2, 5, 7, 11; 1, 6, 9, 10]

3) для доступа к первой строке введите команду:

--> X(1,:)

4) для доступа к последней строке введите:

--> X(\$,:)

5) для доступа к последним двум строкам введите команду:

--> X(\$-1:\$, :)

6) для доступа к 1-й и 3-й строкам введите команду:

--> X([1,3], :)

7) для доступа к 1-му столбцу введите:

--> X(:,1)

8) для доступа к последнему столбцу введите:

--> X(:, \$)

9) для доступа к 1-му и 3-му столбцам введите команду:

--> X(:, [1,3])

10) для доступа к элементам 2-й строки X(2, 2), X(2, 3), X(2,4):

--> X(2, [2, 3, 4])

11) для доступа к элементам из нескольких строк и столбцов:

--> X([2,3], [2,3,4])

2.10. Добавление элементов к массиву

2.10.1. **Задание.** Элементы могут быть добавлены к вектору путем присвоения значений новым элементам:

1) создайте вектор-строку из 4-х элементов:

--> w = 1:4

добавьте к нему еще 6 элементов:

--> w(5:10) = 10:5:35

2) создайте вектор-строку:

--> z = [5 7 2]

добавьте к нему 8-й элемент:

--> z(8) = 4

2.10.2. **Задание.** Строки и/или столбцы могут быть добавлены к существующей матрице путем присваивания значений новым строкам или столбцам. Однако следует помнить, что размеры добавляемых строк и столбцов должны соответствовать существующей матрице.

1) создайте матрицу:

```
--> D = [1 2 3 4; 5 6 7 8]
```

2) добавьте элементы 3-й строки:

```
--> D(3,:) = [10:4:22]
```

3) добавьте элемент:

```
--> D(4,5) = 17
```

2.11. Удаление элементов массивов

Для удаления элемента или группы элементов массива используется символ «[]» – пустые квадратные скобки.

2.11.1. **Задание.** Исследуйте способы удаления элементов вектора:

1) создайте вектор:

```
--> v = [2 8 40 65 3 55 23 154 75 80]
```

2) удалите 6-й элемент:

```
--> v(6) = [ ]
```

3) удалите элементы с 3-го по 6-й:

```
--> v(3:6) = [ ]
```

2.11.2. Исследуйте способы удаления строк и столбцов матрицы:

1) создайте матрицу:

```
--> N = [1 2 3; 4 5 6; 7 8 9]
```

2) удалите 2-й столбец матрицы:

```
--> N(:,2) = [ ]
```

3) удалите 2-ю строку матрицы:

```
--> N(2,:) = [ ]
```

2.12. Специальные матричные функции

Для работы с матрицами и векторами в Scilab существуют специальные функции. Рассмотрим некоторые функции, которые могут формировать матрицы и определять их основные параметры.

Функция `matrix(A [, n, m])` – преобразует матрицу A в матрицу другого размера.

Задание. Исследуйте работу функции `matrix()`:

1) создание матрицы D размером 3×2 :

```
--> D = [1 2; 3 4; 5 6]
```

2) создание матрицы размером 2×3 :

```
--> matrix(D,2,3)
```

3) создание матрицы размером 1×6 (вектора-строки):

```
--> matrix(D,1,6)
```

4) создание матрицы размером 6×1 (вектора-столбца):

```
--> matrix(D,6,1)
```

Функция `length(X)` – определяет количество элементов массива X ; если X – вектор, то его длину; если X – матрица, то вычисляет общее число ее элементов.

Задание. Исследуйте работу функции `length()`:

1) создайте вектор-строку:

```
--> v = [-1 0 3 -2 1 -1 1]
```

2) определите его длину:

```
--> length(v)
```

3) создайте матрицу:

```
--> M = [1 2 3; 4 5 6]
```

4) определите количество элементов матрицы:

```
--> length(M)
```

Функция `size(X [, fl])` – определяет размер массива X ; если X – матрица, то `size(X, 1)` или `size(X, 'r')` определяет число строк матрицы X , а `size(X, 2)` или `size(X, 'c')` – число столбцов.

Задание. Исследуйте работу функции `size()`:

1) создайте матрицу:

```
--> M = [1 2; 3 4; 5 6; 7 8]
```

2) определите количество строк `m` и столбцов `n` матрицы:

```
--> [m,n] = size(M)
```

3) определите количество строк:

```
--> size(M,1)
```

4) определите количество столбцов:

```
--> size(M,2)
```

Функция `ones(m, n)` – создает матрицу из единиц, размером `m` строк и `n` столбцов.

Задание. Исследуйте работу функции `ones()`:

1) формирование квадратной матрицы размером 2×2 :

```
--> ones(2,2)
```

2) формирование вектора-строки:

```
--> ones(1,3)
```

3) формирование матрицы размера $m \times n$:

```
--> m = 3; n = 2;
```

```
--> X = ones(m,n)
```

4) формирование матрицы `Y` из единиц с размерами матрицы `M`:

```
--> M = [1 2 3; 4 5 6]
```

```
--> Y = ones(M)
```

Функция `zeros(m, n)` – создает матрицу из нулей, размером `m` строк и `n` столбцов.

Задание. Исследуйте работу функции `zeros()`:

1) формирование матрицы размером 3×2 :

```
--> zeros(3,2)
```

2) формирование нулевой матрицы Z с размером вектора v :
--> $v = [1 \ 2 \ 3 \ 4 \ 5]$

--> $Z = \text{zeros}(v)$

Функция $\text{eye}(m, n)$ – формирует единичную матрицу из m строк и n столбцов.

Задание. Исследуйте работу функции $\text{eye}()$:

1) создание квадратной матрицы:

--> $\text{eye}(3,3)$

2) создание вектора-столбца:

--> $\text{eye}(5,1)$

3) формирование единичной матрицы E с размерами матрицы M :

--> $M = [0 \ 1; \ 2 \ 3]$

--> $E = \text{eye}(M)$

3. Задания для самостоятельной работы

Указание. Отчет по лабораторной работе № 2 выполняется в виде дневника, т. е. содержимого командного окна Scilab в ходе выполнения заданий для самостоятельной работы. Файл дневника создается в папке Lab2, рекомендуемое имя для файла дневника – `self_work.txt`. Каждое задание должно в начале иметь строку комментария, например, вида:

// Задание 1

Тексты заданий приводить не нужно.

Задание 1. Используйте одну команду для создания вектора-строки (с присвоением его значений переменной с именем w) с 19 элементами:

$w = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 9 \ 8 \ 7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1]$

Не вводите вектор поэлементно.

Задание 2. Для заданной матрицы:

$M = [2 \ 4 \ 1; \ 6 \ 7 \ 2; \ 3 \ 5 \ 9]$

введите команды, которые выполняют следующее:

- а) присваивают первую строку матрицы M вектору с именем $x1$;
- б) присваивает последние 2 строки матрицы M новой матрице с именем Y .

Задание 3. Для заданной матрицы:

$$A = [2, 7, 9, 7; 3, 1, 5, 6; 8, 1, 2, 5]$$

напишите команды, которые выполняют следующее:

- а) присваивают четные столбцы матрицы A новой матрице с именем B ;
- б) присваивают нечетные строки матрицы A новой матрице с именем C ;
- в) преобразуют матрицу A в матрицу размером 4×3 ;
- г) извлекают корень квадратный из всех элементов матрицы A .

4. Содержание отчета

Отчет по лабораторной работе № 2 выполняется в виде дневника, т. е. содержимого командного окна Scilab в ходе выполнения заданий для самостоятельной работы.

Разработанный дневник необходимо продемонстрировать преподавателю для оценки объема выполненной работы и анализа качества выполнения заданий.

Контрольные вопросы

1. Какими способами можно создать вектор-строку?
2. Какими способами можно создать вектор-столбец?
3. Какими способами можно создать матрицу?
4. Как можно использовать оператор «:» (двоеточие) для создания вектора?
5. Объясните термин «конкатенация массивов».
6. Объясните термин «транспонирование массивов».
7. Как можно транспонировать вектор-строку и вектор-столбец?
8. Как выполняется адресация для векторов?
9. Как выполняется адресация для матриц?

Лабораторная работа № 3

МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ С МАССИВАМИ

1. Цель работы

Изучить и исследовать методы математических операций с векторами и матрицами. Исследовать встроенные функции для операций с массивами.

2. Порядок выполнения работы

2.1. Создание папки для работы

При выполнении лабораторной работы № 3 Вы будете использовать папку с именем Lab3, которую необходимо предварительно создать.

С этой целью откройте Вашу рабочую папку (с Вашей фамилией) и создайте в ней (с помощью клавиши F7) новую папку с именем Lab3.

В дальнейшем Вы будете записывать и хранить в этой папке все файлы при выполнении лабораторной работы № 3. Полный путь к этой папке будет такой:

F:\ОТС\Ivanov\Lab3

2.2. Запуск и настройка среды разработки Scilab

2.2.1. После запуска Scilab необходимо разместить на экране дисплея слева текст данной лабораторной работы, а справа – рабочий стол среды разработки Scilab.

2.2.2. Нужно установить в качестве активной (текущей) папку Lab3. С этой целью подведите курсор мыши к иконке слева от строки со списком папок. Появится подсказка «Выберите папку». Щелкните кнопкой мыши по иконке. Раскроется окно с заголовком «Выберите папку». Сначала выберите диск F, затем папку с именем ОТС, затем папку с вашей фамилией (условно Ivanov) и затем папку Lab3. В строке Folder name должно быть следующее: F:\ОТС\Ivanov\Lab3. Если это так, то щелкните по кнопке Open.

В результате этих действий в строке обозревателя файлов должно появиться:

F:\ОТС\Ivanov\Lab3

2.2.3. При необходимости очистите командное окно и журнал команд, используя пункты меню Правка.

2.3. Исследование матричных математических операций

2.3.1. **Задание.** Исследуйте выполнение операций сложения и вычитания с векторами и матрицами:

1) создайте векторы $\text{vect}A = [8 \ 5 \ 4]$ и $\text{vect}B = [10 \ 2 \ 7]$. Выполните сложение векторов:

$$\text{vect}C = \text{vect}A + \text{vect}B,$$

а затем вычитание:

$$\text{vect}D = \text{vect}A - \text{vect}B$$

2) создайте две матрицы:

$$A = [5 \ -3 \ 8; \ 9 \ 2 \ 10] \text{ и } B = [10 \ 7 \ 4; \ -11 \ 15 \ 1]$$

Выполните операции сложения матриц $A + B$ и вычитания $A - B$. Затем выполните операцию сложения вектора $\text{vect}A$ с матрицей A :

$$\text{vect}A + A$$

Объясните полученные результаты.

2.3.2. **Задание.** Исследуйте операции сложения и вычитания скаляров с векторами и матрицами:

1) создайте вектор $\text{vect}A = [1 \ 5 \ 8 \ -10 \ 2]$ и затем выполните операцию:

$$\text{vect}A + 4$$

2) создайте матрицу $A = [6 \ 21 \ -15, \ 0 \ -4 \ 8]$ и затем выполните:

$$A - 5$$

Объясните полученные результаты.

2.3.3. **Задание.** Выполните операции матричного умножения:

1) создайте две матрицы:

$$A = [1 \ 4 \ 2, \ 5 \ 7 \ 3, \ 9 \ 1 \ 6, \ 4 \ 2 \ 8]$$

$$B = [6 \ 1, \ 2 \ 5, \ 7 \ 3]$$

и выполните следующие операции:

$$C = A * B$$

$$D = B * A$$

Объясните полученные результаты.

2) создайте две квадратные матрицы:

$$F = [1 \ 3, \ 5 \ 7]$$

$$G = [4 \ 2, \ 1 \ 6]$$

и выполните следующие операции:

$$F * G$$

$$G * F$$

Объясните полученные результаты.

3) создайте два вектора:

$$v = [2 \ 5 \ 1]$$

$$u = [3, \ 1, \ 4]$$

и выполните следующие операции:

$$v * u$$

$$u * v$$

Объясните полученные результаты.

4) создайте матрицу:

$$A = [2 \ 5 \ 7 \ 0; \ 10 \ 1 \ 3 \ 4; \ 6 \ 2 \ 11 \ 5]$$

и выполните операции умножения ее на скалярную величину:

$$3 * A$$

$$A * 3$$

Объясните полученные результаты.

2.3.4. Операция деления массивов не определена в линейной алгебре. В Scilab операция деления массивов используется для решения матричных уравнений. Используется два варианта операции деления: « \ » – левое деление и « / » – правое деление.

Операция левого деления используется для решения матричного уравнения:

$$A \cdot X = B,$$

где X и B являются векторами-столбцами.

Решением этого уравнения будет:

$$X = A \setminus B$$

Операция правого деления используется для решения матричного уравнения

$$X \cdot C = D,$$

где X и D являются векторами-строками.

Решением этого уравнения будет:

$$X = D/C$$

Задание. Используйте матричные операции для решения следующей системы линейных алгебраических уравнений (СЛАУ):

$$\begin{aligned} 4x - 2y + 6z &= 8; \\ 2x + 8y + 2z &= 4; \\ x + 10y + 3z &= 0. \end{aligned} \quad (1)$$

Эту систему можно записать в матричной форме как:

$$A \cdot X = B, \quad (2)$$

где

$$A = \begin{bmatrix} 4 & -3 & 6 \\ 2 & 8 & 2 \\ 6 & 10 & 3 \end{bmatrix};$$

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix};$$

$$B = \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix}.$$

Выполните решение уравнения (2) несколькими способами:

1) с использованием операции левого деления:

$$\rightarrow A = [4 \ -2 \ 6; 2 \ 8 \ 2; 6 \ 10 \ 3]$$

$$\rightarrow B = [8; 4; 0]$$

$$\rightarrow X = A \setminus B$$

2) с использованием функции вычисления обратной матрицы:

$$\rightarrow Xa = \text{inv}(A) * B$$

3) с использованием символа инверсии матрицы A^{-1} :

$$\rightarrow Xb = A^{(-1)} * B$$

Сделайте выводы по полученным результатам;

4) запишите уравнение (1) в форме:

$$X \cdot C = D, \quad (3)$$

где X – вектор-строка неизвестных; $C = A'$ – матрица коэффициентов системы; $D = B'$ – вектор-строка свободных членов.

Выполните решение уравнения (3) несколькими способами:

1) с использованием операции правого деления:

$$\rightarrow C = A'$$

$$\rightarrow D = B'$$

$$\rightarrow Xc = D / C$$

2) с использованием функции вычисления обратной матрицы:

$$\rightarrow Xd = D * \text{inv}(C)$$

3) с использованием символа инверсии матрицы C^{-1} :

$$\rightarrow Xe = D * C^{(-1)}$$

Сделайте выводы по полученным результатам.

2.4. Исследование поэлементных математических операций с массивами

2.4.1. **Задание.** Исследуйте поэлементные операции умножения, деления и возведения в степень матриц:

1) создайте матрицы:

$$\rightarrow A = [2 \ 6 \ 3; 5 \ 8 \ 4]$$

$$\rightarrow B = [1 \ 4 \ 10; 3 \ 2 \ 7]$$

2) выполните операции:

$$\rightarrow A .* B$$

$$\rightarrow C = A ./ B$$

--> $V.^3$

--> $A * V$

Сделайте выводы по полученным результатам.

2.4.2. Задание. Исследуйте поэлементные операции с векторами:

1) создайте вектор-строку:

--> $x = [1:8]$

и вычислите выражение.

В Scilab для этого нужно ввести:

--> $y = x.^2 - 4 * x$

2) создайте вектор-строку:

--> $t = 1:5$

3) вычислите функцию:

--> $y = \cos(t)$

4) вычислите функцию:

--> $z = y / t$

Результатом будет единственное число, а не функция $z(t) = \cos(t)/t$ от нескольких значений аргумента t ;

5) исправьте вычисление функции z , используя поэлементную операцию деления:

--> $z = y ./ t$

Сделайте выводы по полученным результатам.

2.5. Встроенные функции для операций с массивами

Для работы с матрицами и векторами в Scilab существуют специальные функции. Рассмотрим некоторые функции, которые могут определять основные параметры массивов:

$\text{mean}(A)$ – если A – вектор, то функция возвращает среднее (арифметическое) значение элементов;

$\text{max}(A)$ – если A – вектор, то функция возвращает наибольший элемент;

$\text{min}(A)$ – если A – вектор, то функция возвращает наименьший элемент;

`sum(A)` – если A – вектор, то функция возвращает сумму элементов;

`length(A)` – если A – вектор, то функция возвращает длину вектора (количество элементов).

Подробную информацию о функциях можно получить из справочной системы Scilab, выполнив команду:

`help function_name`

3. Задания для самостоятельной работы

Указание. Отчет по лабораторной работе № 3 выполняется в виде дневника, т.е. содержимого командного окна Scilab в ходе выполнения заданий для самостоятельной работы. Файл дневника создается в папке Lab3, рекомендуемое имя для файла дневника – `self_work.txt`. Каждое задание должно в начале иметь строку комментария, например, вида:

// Задание 1

Тексты заданий приводить не нужно.

Задание 1. Выполните следующие команды и объясните, почему Scilab генерирует ошибку:

1)

`x = [1, 2, 3];`

`y = [4, 5, 6];`

`z = x * y`

2)

`x = [1, 2, 3];`

`y = [4, 5, 6];`

`z = y * x'`

3)

`x = [1, 2, 3];`

`y = [4, 5, 6];`

`z = y' * x`

Задание 2. Пусть известны округленные среднесуточные значения температуры воздуха в течение недели, °C. Эти данные заданы в массиве $temp = [-1 \ 0 \ 2 \ 3 \ -5 \ -7 \ -4]$. Требуется определить:

- 1) минимальную температуру;
- 2) максимальную температуру;
- 3) среднюю температуру.

Указание. Для вычисления значений необходимо использовать встроенные функции Scilab для работы с массивами.

Задание 3. Создайте три матрицы:

$$A = \begin{bmatrix} 2 & 4 & -1 \\ 3 & 1 & -5 \\ 0 & 1 & 4 \end{bmatrix}; \quad B = \begin{bmatrix} -2 & 5 & 0 \\ -3 & 2 & 7 \\ -1 & 6 & 9 \end{bmatrix}; \quad C = \begin{bmatrix} 0 & 3 & 5 \\ 2 & 1 & 0 \\ 4 & 6 & -3 \end{bmatrix}$$

и вычислите:

1) выражения $A + B$ и $B + A$, чтобы показать, что сложение матриц подчиняется переместительному закону;

2) выражения $A * (B + C)$ и $A * B + A * C$, чтобы показать, что умножение матриц подчиняется распределительному закону.

Задание 4. Используйте матрицы A , B и C из задания 4, чтобы ответить на следующие вопросы:

- 1) справедливо ли равенство: $A * B = B * A$?
- 2) справедливо ли равенство: $A * (B * C) = (A * B) * C$?
- 3) справедливо ли равенство: $(A * B)' = B' * A'$?
- 4) справедливо ли равенство: $(A + B)' = A' + B'$?

Задание 5. Определите вектор $x = [1 : 10]$ и напишите команды для выполнения следующих операций:

1) прибавить число 10 к каждому элементу вектора x . Присвоить результат новому вектору m ;

2) прибавить число 3 только к нечетным элементам вектора x . Присвоить результат новому вектору g ;

3) вычислить квадратный корень из каждого элемента вектора x . Присвоить результат новому вектору s ;

4) возвести в квадрат каждый элемент вектора x . Присвоить результат новому вектору t .

Задание 6. Определите два вектора-столбца:

$$x = [1; 2; 3; 4] \text{ и } y = [5; 6; 7; 8].$$

Напишите команды, которые выполняют следующие операции:

1) прибавить сумму элементов вектора x к элементам вектора y (используйте для суммирования встроенную функцию $\text{sum}(\)$). Присвоить полученный результат новому вектору m ;

2) возвести каждый элемент вектора x в степень, соответствующую элементу вектора y . Присвоить результат новому вектору s ;

3) разделить каждый элемент вектора y на соответствующий элемент вектора x . Присвоить результат новому вектору z ;

4) умножить каждый элемент вектора x на соответствующий элемент вектора y . Присвоить результат новому вектору r ;

5) найти сумму элементов вектора x и присвоить результат переменной s именем t ;

6) вычислить выражение: $x' * y - t$

Задание 7. Создайте вектор $t = 1:0.2:2$, затем напишите и выполните следующие математические выражения:

1) $\ln(2 + t + t^2)$;

2) $\lg(2 + t + t^2)$;

3) $e^{1 + \cos 3t}$;

4) $\cos^2 t + \sin^2 t$;

5) $\text{arctg } t$;

6) $\text{ctg } t$.

Задание 8. Докажите, что $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$. Сделайте это путем создания вектора x с элементами: 1.5, 1.0, 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001. Затем создайте новый вектор y , в котором каждый элемент определяется из элементов x с помощью вычисления выражения $\frac{\sin x}{x}$.

Сравните полученные значения элементов из y со значением 1 (используйте формат для отображения чисел с 20 символами).

Задание 9. Используйте Scilab для доказательства, что сумма членов бесконечного ряда:

$$\sum_{n=1}^{\infty} \frac{1}{2^n} = \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \dots$$

стремится к 1. Сделайте это путем вычисления суммы для

1) $n = 10$;

2) $n = 20$;

3) $n = 30$;

4) $n = 40$.

Для каждого варианта создайте вектор с именем n , в котором 1-й элемент будет 1, шаг равен 1, а последний элемент, соответственно, 10, 20, 30 или 40. Затем используйте поэлементные операции для создания вектора, в котором элементы равны $\frac{1}{2^n}$. И наконец, используйте встроенную функцию `sum()` для сложения членов ряда. Сравните значения, полученные в вариантах 1), 2), 3) и 4) со значением 1 (не забудьте поставить в конце команд символ «;», чтобы не выводить на дисплей множество элементов векторов).

Задание 10. Решите следующую СЛАУ:

$$3x - 2y + 2z = 7.5;$$

$$-4.5x + 2y + 3z = 5.5;$$

$$5x + y - 2.5z = 4.5.$$

Проверьте правильность вычисления неизвестных x , y , z .

Задание 11. Решите следующую СЛАУ:

$$3u + 1.5v + w + 0.5x + 4y = -11.75;$$

$$-2u + v + 4w - 3.5x + 2y = 19;$$

$$6u - 3v + 2w + 2.5x + y = -23;$$

$$u + 4v - 3w + 0.5 - 2y = -1.5;$$

$$3u + 2v - w + 1.5x - 2y = -3.5.$$

Проверьте правильность вычисления неизвестных.

4. Содержание отчета

Отчет по лабораторной работе № 3 выполняется в виде дневника, т. е. содержимого командного окна Scilab в ходе выполнения заданий для самостоятельной работы.

Разработанный дневник необходимо продемонстрировать преподавателю для оценки объема выполненной работы и анализа качества выполнения заданий.

Контрольные вопросы

1. Как выполняется матричное сложение и вычитание?
2. Как выполняется матричное умножение?
3. Для каких целей в Scilab введены операции левого деления “ \ ” и правого деления “ / “ ?
4. Как выполняются операции поэлементного умножения и деления с массивами?
5. Как выполняется операция поэлементного возведения в степень с массивом?
6. Какие основные встроенные функции для операций с массивами имеются в Scilab?

Лабораторная работа № 4

УПРАВЛЕНИЕ ВВОДОМ И ВЫВОДОМ ДАННЫХ В СКРИПТ-ФАЙЛАХ

1. Цель работы

Изучить и исследовать функции ввода `input()` и вывода `disp()` информации в скрипт-файлах. Исследовать функцию форматированного вывода данных `mprintf()`.

2. Порядок выполнения работы

2.1. Создание папки для работы

При выполнении лабораторной работы № 4 Вы будете использовать папку с именем Lab4, которую необходимо предварительно создать.

С этой целью откройте Вашу рабочую папку (с Вашей фамилией) и создайте в ней (с помощью клавиши F7) новую папку с именем Lab4.

В дальнейшем Вы будете записывать и хранить в этой папке все файлы при выполнении лабораторной работы № 4. Полный путь к этой папке будет такой:

F:\ОТС\Ivanov\Lab4

2.2. Запуск и настройка среды разработки Scilab

2.2.1. После запуска Scilab необходимо разместить на экране дисплея слева текст данной лабораторной работы, а справа – рабочий стол среды разработки Scilab.

2.2.2. Нужно установить в качестве активной (текущей) папку Lab4. С этой целью подведите курсор мыши к иконке слева от строки со списком папок. Появится подсказка «Выберите папку». Щелкните кнопкой мыши по иконке. Раскроется окно с заголовком «Выберите папку». Сначала выберите диск F, затем папку с именем ОТС, затем папку с Вашей фамилией (условно Ivanov) и затем папку Lab4. В строке Folder name должно быть следующее: F:\ОТС\Ivanov\Lab4. Если это так, то щелкните по кнопке Open.

В результате этих действий в строке обозревателя файлов должно появиться:

F:\ОТС\Ivanov\Lab4

2.2.3. При необходимости очистите командное окно и журнал команд, используя пункты меню Правка.

2.3. Исследование метода задания данных для скрипта из командного окна

2.3.1. Переменные из скрипт-файлов и переменные из командного окна находятся в едином рабочем пространстве Scilab, т. е. используют одну и ту же область памяти. Поэтому можно определять переменные и присваивать им различные значения непосредственно в командном окне перед исполнением скрипта, не редактируя сам скрипт-файл.

Задание. Откройте редактор SciNotes и создайте следующий скрипт:

```
// Скрипт вычисляет среднее значение трех переменных x1, x2, x3
```

```
// Переменные определяются в командном окне
```

```
average = (x1 + x2 + x3) / 3
```

Сохраните скрипт-файл под именем, например, ave.sce в папке Lab4.

Закройте окно редактора SciNotes и затем:

1) введите в командном окне данные для скрипта:

```
--> x1 = 10;
```

```
--> x2 = 20;
```

```
--> x3 = 30;
```

Запустите скрипт с помощью команды:

```
--> exec('ave.sce')
```

Будет вычислено среднее значение:

```
average =
```

```
20.
```

2) выполните скрипт-файл при новых исходных данных, например, $x_1 = 15$, $x_2 = 25$, $x_3 = 35$.

2.4. Использование команды input для ввода данных в скрипт-файл

2.4.1. В этом случае переменная определяется в скрипте, а когда файл исполняется, пользователю предлагается присвоить значение переменной.

Формат команды input:

```
variable_name = input('string-message')
```

Когда идет прогон скрипта, выполняется команда input и строка-сообщение отображается в командном окне. Обычно эта строка сообщает, что пользователь должен ввести значение, которое присваивается переменной. Строка-сообщение должна быть заключена в одинарные «'» «'» или двойные «"» «"» кавычки. Текст сообщения может быть на английском или русском языках.

Как и для любой переменной в Scilab, переменная и присвоенное ей значение может быть отображены в командном окне, если не напечатан символ «;» в конце команды input.

Задание. Откройте редактор SciNotes и создайте скрипт:

```
// Скрипт вычисляет среднее значение трех переменных x1, x2, x3
```

```
// Переменные объявляются в скрипте, но значения им присваиваются
```

```
// с помощью команды input при исполнении скрипта в командном окне
```

```
x1 = ('Введите значение переменной x1: ');
```

```
x2 = ('Введите значение переменной x2: ');
```

```
x3 = ('Введите значение переменной x3: ');
```

```
average = (x1 + x2 + x3) / 3
```

Сохраните набранный скрипт-файл в папке Lab4 под именем ave2.sce. Закройте редактор SciNotes и затем:

1) запустите скрипт из командного окна с параметром 2, чтобы не отображались команды файла:

```
--> exec('ave2.sce', 2)
```

Введите последовательно запрашиваемые значения переменных, например, $x_1 = 10$, $x_2 = 20$, $x_3 = 30$;

2) повторно запустите скрипт ave2.sce и введите значения переменных $x_1 = 15$, $x_2 = 25$, $x_3 = 35$.

2.4.2. В рассмотренном примере присваивались значения скалярным переменным. Однако с помощью команды input можно присваивать значения векторам и матрицам (вводом левой квадратной скобки, затем – строка за строкой, в заключение – правая квадратная скобка).

Задание. Откройте блокнот SciNotes и создайте следующий скрипт:

```
// Скрипт вычисляет среднее значение элементов вектора-строки
```

```
X = input ('Введите произвольное количество чисел как  
элементов вектора-строки X = ');
```

```
// Количество элементов вектора
```

```
lenX = length(X)
```

```
// Сумма элементов вектора
```

```
sumX = sum(X)
```

```
// Среднее значение элементов вектора
```

```
averageX = sumX / lenX
```

Примечание. Обратите внимание, каким образом в команде input выполнен ввод длинной строки.

Сохраните скрипт-файл в папке Lab1 под именем ave3.sce. Закройте редактор SciNotes.

Запустите скрипт на исполнение с помощью команды

```
--> exec(ave3.sce, 2)
```

Введите вектор-строку с произвольным количеством чисел (5–10). Убедитесь в правильности вычисленных данных.

2.5. Команда вывода на дисплей *disp*

2.5.1. Команда `disp` используется для отображения элементов переменной без отображения ее имени, или для отображения текста.

Формат команды:

```
disp (variable_name)
```

или

```
disp('text')
```

Всякий раз, когда исполняется команда `disp`, выводимое сообщение появляется с новой строки.

Задание. Выполните несколько примеров вывода при помощи команды `disp`:

1) вывод значения скаляра:

```
--> x1 = 10,
```

```
--> disp(x1)
```

2) вывод элементов матрицы:

```
--> A = [5 9 1; 7 2 4];
```

```
--> disp(A)
```

3) вывод строки

```
--> disp ('The problem has no solution')
```

```
--> disp ('Задача не имеет решения')
```

2.5.2. Команда `disp` может использоваться в скрипт-файлах для вывода данных.

Задание. Откройте редактор SciNotes и запишите в него следующий скрипт:

```
// Скрипт вычисляет среднее значение трех переменных x1, x2, x3
```

```
x1 = input ('Введите значение переменной x1 = ');
```

```
x2 = input ('Введите значение переменной x2 = ');
```

```
x3 = input ('Введите значение переменной x3 = ');
```

```
average = (x1 + x2 + x3) / 3,
```

```
disp(''); // вывод пустой строки
disp ('Среднее значение трех переменных равно: ')
disp(average)
```

Сохраните набранный скрипт-файл в папке Lab4 под именем ave4.sce. Закройте редактор SciNotes.

Запустите скрипт из командного окна с параметром 2, чтобы не отображались команды файла:

```
--> exec(ave4.sce, 2)
```

Введите последовательно запрашиваемые значения переменных, например, $x_1 = 10$, $x_2 = 20$, $x_3 = 30$.

Убедитесь в правильности выполнения программы.

2.5.3. Во многих случаях возникает необходимость выводить данные в виде таблицы. Так как с помощью команды `disp` может быть выведена только одна переменная, то необходимо выводимые в виде таблицы переменные определить как столбцы матрицы, которая будет являться аргументом в команде `disp`.

Задание. Наберите в командном окне программу, которая выводит на дисплей таблицу из двух столбцов: в левом столбце будут значения целых чисел, а в правом столбце – их квадраты:

```
--> x = 1 : 6;      // создать вектор-строку из шести чисел
--> y = x.^2;      // вычислить квадраты этих чисел
--> table(:,1) = x'; // сформировать матрицу, у которой 1-й
// столбец будет иметь значения элементов вектора x
--> table(:,2) = y'; // сформировать матрицу, у которой 2-й
//столбец будет иметь значения элементов вектора y
disp (' Числа:   Их квадраты:')
disp(' ')
disp(table)
```

Объясните выведенные на дисплей данные.

2.6. Команда форматированного вывода данных *mprintf*

Команда `mprintf` может быть использована для отображения вывода числовых данных или текста на дисплее. С помощью этой команды (в отличие от команды `disp`) вывод может быть отформатирован. Например, текст и числовые значения данных могут быть произвольно перемешаны и отображаться в одной строке. В дополнение формат отображения чисел может быть управляем.

2.6.1. Для вывода текста формат команды `mprintf` имеет вид:

```
mprintf('text as string')
```

Задание. Выполните вывод в командное окно текста:

```
--> mprintf('The problem has no solution')
```

```
--> mprintf('Задача не имеет решения')
```

Если выводимая строка очень длинная, то можно разбить ее на несколько строк при выводе вставкой символа “ \n “ перед символом в строке, который должен начинать новую строку вывода.

Задание. Выполните вывод «длинной» строки:

```
--> mprintf('The problem has no solution.\n Please check the input data.')
```

2.6.2. Для отображения смеси текста и числа (значения переменной) команда `mprintf` имеет формат:

```
mprintf('text %-5.2f additional text\n', variable_name)
```

Здесь символ % указывает место, где число должно быть вставлено в текст, а `-5.2f` – это форматизирующие элементы.

Подробную информацию о форматизирующих элементах можно получить из справочной системы Scilab по команде:

```
help mprintf
```

Так, например, символ `f` означает вывод числа в формате с плавающей точкой; символ `d` – вывод числа в формате целого; `.2f` – вывод числа в формате с плавающей точкой и двумя дробными цифрами.

Задание. Выполните вывод числа π в разных форматах:

```
--> mprintf('Вывод целого числа: %d\n', %pi)
```

```
--> mprintf('Вывод дробного числа: %f\n', %pi)
```

```
--> mprintf('Вывод дробного числа: %.2f\n', %pi)
```

Объясните полученные результаты вывода.

2.6.3. С помощью команды `mprintf` возможно вставить более одного числа (значения переменной) в текст. Это делается с помощью наборов знаков `%` с последующими элементами форматирования в тех местах текста, где числа должны быть вставлены.

В общем случае команда `mprintf` выглядит так:

```
mprintf('text ..... %d ..... %f .....%f ...text\n', var1, var2, var3)
```

Задание. Введите в командное окно следующую программу:

```
--> x1 = 10; x2 = 20.2; x3 = 30.3333; x4 = 40.44444;  
--> mprintf('x1 = %d, x2 = %d, x3 = %f, x4 = %.2f\n', x1, x2, x3, x4)
```

Объясните полученные результаты вывода данных.

2.6.4. Во многих случаях возникает необходимость выводить данные в виде таблицы. Для этой цели переменные для выводимых данных нужно представить в виде векторов-столбцов.

Задание. Введите в командное окно программу, которая выводит таблицу значений целых чисел и их квадратов:

```
--> x = 1:6; // создание вектора-строки из целых чисел  
--> y = x.^2; // вычисление квадратов чисел  
  
// вывод таблицы чисел и их квадратов как элементов векторов-  
// столбцов  
--> mprintf('Число равно %d, его квадрат равен %d\n', x', y')
```

Объясните полученные результаты вывода данных.

2.6.5. Команда `mprintf` часто используется для вывода данных в скрипт-файлах.

Задание. Необходимо создать скрипт-файл, в котором задать путем присваивания необходимые исходные данные. Записать выражения для решения задачи. Вывести результаты расчета и исходные данные в командное окно Scilab с помощью функции `mprintf`. Вывод осуществить так, как указано в задании. Вывести все заданные и рассчитанные числовые значения с двумя знаками после десятичной точки. Вывести рядом с числовыми значениями название единиц измерения.

Условие задачи. Задать мощности P_1, P_2, P_3 трех потребителей электроэнергии на участке промышленного предприятия и их координаты $x_1, y_1, x_2, y_2, x_3, y_3$. Рассчитать координаты x_0, y_0 центра нагрузок по формулам:

$$x_0 = \frac{P_1 x_1 + P_2 x_2 + P_3 x_3}{P_1 + P_2 + P_3};$$

$$y_0 = \frac{P_1 y_1 + P_2 y_2 + P_3 y_3}{P_1 + P_2 + P_3}.$$

Вывести рассчитанные и исходные значения в виде:

КООРДИНАТЫ ЦЕНТРА ЭЛЕКТРИЧЕСКИХ НАГРУЗОК:

$$x_0 = 21,02 \text{ м} \quad y_0 = 34,03 \text{ м}$$

ИСХОДНЫЕ ДАННЫЕ:

Номер	X, м	Y, м	P, кВт
1	13.50	45.00	55.80
2	20.00	32.00	17.40
3	35.00	15.60	31.30

Откройте редактор SciNotes и наберите следующий текст:

```
// Задаем исходные данные
x1=13.5; y1=45; P1=55.8;
x2=20; y2=32; P2=17.4;
x3=35; y3=15.6; P3=31.3;

// Выполняем расчет
x0=(P1*x1+P2*x2+P3*x3)/(P1+P2+P3);;
y0=(P1*y1+P2*y2+P3*y3)/(P1+P2+P3);

// Выводим в командное окно
mprintf('\n')
mprintf('КООРДИНАТЫ ЦЕНТРА ЭЛЕКТРИЧЕСКИХ НАГРУ-
ЗОК:\n')
mprintf('-----\n')
mprintf('x0 = %.2f м y0 = %.2f м\n', x0, y0)
```

```

mprintf('\n')
mprintf('ИСХОДНЫЕ ДАННЫЕ:\n')
mprintf('-----\n')
mprintf('Номер      X, м      Y, м      P, кВт\n')
mprintf('  1         %.2f    %.2f    %.2f\n', x1,y1,P1)
mprintf('  2         %.2f    %.2f    %.2f\n', x2,y2,P2)
mprintf('  3         %.2f    %.2f    %.2f\n', x3,y3,P3)
mprintf('\n')

```

Сохраните скрипт-файл под именем koord.sce. Выполните скрипт из окна редактора без отображения команд. Сравните выведенный текст с заданным. Если будут существенные отличия, то проведите корректировку скрипта.

3. Задания для самостоятельной работы

Указание. Отчет по лабораторной работе № 4 выполняется в виде дневника, т. е. содержимого командного окна Scilab в ходе выполнения заданий для самостоятельной работы. Файл дневника создается в папке Lab4, рекомендуемое имя для файла дневника – self_work.txt. Каждое задание должно в начале иметь строку комментария, например, вида:

// Задание 1

Тексты заданий приводить не нужно.

Задание 1. Создайте скрипт-файл для решения следующей задачи.

Условие задачи. Для динамической колебательной системы, содержащей груз с массой m , пружину с коэффициентом жесткости k и демпфер с коэффициентом демпфирования c , нужно вычислить резонансную частоту системы по формуле

$$f_r = f_n \sqrt{1 - 2s^2},$$

где коэффициент затухания системы s вычисляется по формуле:

$$s = \frac{c}{2\sqrt{km}}.$$

Собственная частота незатухающих колебаний системы рассчитывается по формуле

$$f_n = \frac{1}{2\pi} \sqrt{\frac{k}{m}}.$$

Подсказка. Исходными данными для задачи являются:

m – масса системы;

k – коэффициент жесткости пружины;

c – коэффициент демпфирования.

Эти параметры вводятся с помощью трех команд `input`, в которых должен быть текст вида: “Задайте массу системы: “.

Промежуточными являются параметры:

s – коэффициент затухания системы;

f_n – собственная частота незатухающих колебаний.

Результаты вычисления этих параметров не отображаются на дисплее.

Результатом выполнения программы должен быть параметр:

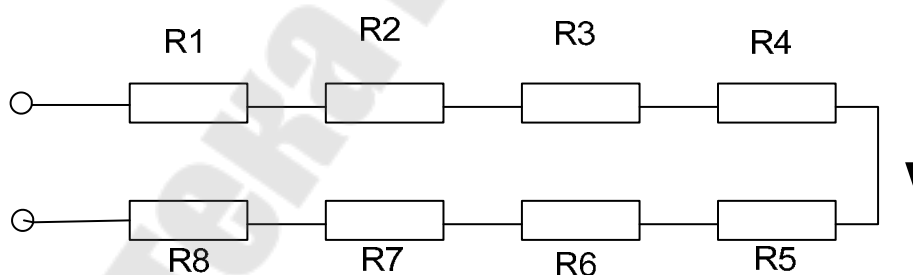
f_r – резонансная частота системы.

Этот параметр должен выводиться на дисплей посредством команды `disp`. Перед выводом значения f_r необходимо вывести на дисплей текст вида: “Резонансная частота системы равна: “.

Разработанный скрипт-файл сохраните в папке Lab4 под именем, например, `zadanie1.sce`. Затем закройте редактор SciNotes.

Запустите скрипт из командного окна без отображения строк программы. Введите для тестирования параметры: $m = 10$; $k = 10$; $c = 0.6$. Должно получиться $f_r = 0.1590$.

Задание 2. Имеется электрическая цепь с 8-ю последовательно соединенными резисторами:



Падение напряжения на отдельном резисторе:

$$U_n = R_n I = \frac{R_n}{R_{eq}} U_s,$$

где U_s – напряжение источника питания; $I = U_s / R_{eq}$ – ток в цепи; $R_{eq} = \sum R_n = R1 + R2 + R3 + R4 + R5 + R6 + R7 + R8$ – эквивалентное (общее) сопротивление цепи.

Мощность, рассеиваемая в каждом резисторе, равна:

$$P_n = I^2 R_n = \frac{R_n}{R_{eq}} U^2.$$

Требуется написать программу (скрипт), которая вычисляет падение напряжения на каждом резисторе и мощность, рассеиваемую в нем. Когда скрипт начинает исполняться, выдается запрос пользователю сначала ввести величину напряжения источника, а затем ввести сопротивления резисторов как элементов вектора-строки.

Программа выводит таблицу со значениями сопротивлений резисторов в первом столбце; падением напряжения на резисторах – во втором столбце; мощностью, рассеиваемую на резисторах, – в третьем столбце. Вслед за таблицей программа выводит величину тока в цепи и общую рассеиваемую мощность.

Исходные данные для расчета:

$U_s = 24 \text{ В}; R_1 = 20 \text{ Ом}; R_2 = 14 \text{ Ом}; R_3 = 12 \text{ Ом}; R_4 = 18 \text{ Ом}; R_5 = 8 \text{ Ом}; R_6 = 15 \text{ Ом}; R_7 = 10 \text{ Ом}; R_8 = 16 \text{ Ом}.$

Разработанный скрипт-файл сохраните в папке Lab4 под именем, например, zadanie2.sce. Затем закройте редактор SciNotes.

Запустите скрипт из командного окна без отображения строк программы.

Задание 3. Зная величины r_{12} , r_{23} , r_{31} треугольника сопротивлений, требуется рассчитать значения сопротивлений r_1 , r_2 , r_3 эквивалентной звезды:

$$r_1 = \frac{r_{12}r_{31}}{r_{12} + r_{23} + r_{31}}; \quad r_2 = \frac{r_{23}r_{12}}{r_{12} + r_{23} + r_{31}}; \quad r_3 = \frac{r_{23}r_{31}}{r_{12} + r_{23} + r_{31}}.$$

Вывести исходные и расчетные значения в виде:

СОПРОТИВЛЕНИЯ ТРЕУГОЛЬНИКА:

$$r_{12} = 3.40 \text{ Ом}$$

$$r_{23} = 5.50 \text{ Ом}$$

$$r_{31} = 2.90 \text{ Ом}$$

СОПРОТИВЛЕНИЯ ЭКВИВАЛЕНТНОЙ ЗВЕЗДЫ:

$$r_1 = 0.84 \text{ Ом}$$

$$r_2 = 1.58 \text{ Ом}$$

$$r_3 = 1.35 \text{ Ом}$$

Решение задачи оформите в виде скрипт-файла с именем, например, `zadanie3.sce`. Выполните скрипт из окна редактора без отображения команд. Сравните выведенный текст с заданным. Если будут существенные отличия, то проведите корректировку скрипта.

Задание 4. Задать высоту H , внешний R и внутренний r радиусы цилиндрического кольца. Вычислить его объем V и площадь S всей поверхности по формулам:

$$S = 2\pi H(R + r) + 2\pi(R^2 - r^2);$$

$$V = \pi H(R^2 - r^2).$$

Вывести заданные и рассчитанные значения в виде:

ОТВЕТ ЗАДАЧИ:

$$V = 54.924 \text{ куб. см}$$

$$S = 219.733 \text{ кв. см}$$

ЗАДАЧА РЕШЕНА

ИСХОДНЫЕ ДАННЫЕ: $H = 5.49 \text{ см}$

$$R = 3.17 \text{ см}$$

$$r = 2.62 \text{ см}$$

Решение задачи оформите в виде скрипт-файла с именем, например, `zadanie4.sce`. Выполните скрипт из окна редактора без отображения команд. Сравните выведенный текст с заданным. Если будут существенные отличия, то проведите корректировку скрипта.

4. Содержание отчета

Отчет по лабораторной работе № 4 выполняется в виде списка скрипт-файлов, разработанных в ходе выполнения заданий для самостоятельной работы.

Разработанный дневник необходимо продемонстрировать преподавателю для оценки объема выполненной работы и анализа качества выполнения заданий.

Контрольные вопросы

1. Какими способами можно загрузить скрипт-файл в рабочее пространство Scilab?
2. Для каких целей применяется команда `input()`?
3. Для каких целей используется команда `disp()`?
4. Каким образом выполняется форматизируемый вывод данных в Scilab?
5. Как можно вывести на дисплей «длинную» строку?
6. Как можно вывести на дисплей целое число?
7. Как можно вывести на дисплей дробное число с двумя цифрами после точки?

Лабораторная работа № 5

ПОСТРОЕНИЕ И ОФОРМЛЕНИЕ ГРАФИКОВ ФУНКЦИЙ

1. Цель работы

Изучить основные функции интегрированной среды Scilab для построения двумерных графиков. Изучить и исследовать возможности оформления графиков и графических окон. Приобрести навыки построения графиков в Scilab.

2. Порядок выполнения работы

2.1. Создание папки для работы

При выполнении лабораторной работы № 5 Вы будете использовать папку с именем Lab5, которую необходимо предварительно создать.

С этой целью откройте Вашу рабочую папку (с Вашей фамилией) и создайте в ней (с помощью клавиши F7) новую папку с именем Lab5.

В дальнейшем Вы будете записывать и хранить в этой папке все файлы при выполнении лабораторной работы № 5. Полный путь к этой папке будет такой:

F:\ОТС\Ivanov\Lab5

2.2. Запуск и настройка среды разработки Scilab

2.2.1. После запуска Scilab необходимо разместить на экране дисплея слева текст данной лабораторной работы, а справа – рабочий стол среды разработки Scilab.

2.2.2. Нужно установить в качестве активной (текущей) папку Lab5. С этой целью подведите курсор мыши к иконке слева от строки со списком папок. Появится подсказка «Выберите папку». Щелкните кнопкой мыши по иконке. Раскроется окно с заголовком «Выберите папку». Сначала выберите диск F, затем папку с именем ОТС, затем папку с Вашей фамилией (условно Ivanov) и затем папку Lab5. В строке Folder name должно быть следующее: F:\ОТС\Ivanov\Lab5. Если это так, то щелкните по кнопке Open.

В результате этих действий в строке обозревателя файлов должно появиться:

F:\ОТС\Ivanov\Lab5

2.2.3. При необходимости очистите командное окно и журнал команд, используя пункты меню Правка.

2.3. Исследование функции plot

2.3.1. Для построения двумерных графиков функции одной переменной вида $y = f(x)$ в Scilab существует команда plot, обращение к которой в простейшем случае выглядит следующим образом:

`plot(x, y)`

Здесь x – список значений независимой переменной, а y – список значений функции $f(x)$ в этих точках. Следовательно, для того чтобы построить график в Scilab, необходимо создать два списка: значений переменной и значений функции в этих точках.

При этом следует учесть, что график, построенный при помощи функции plot, представляет собой ломаную линию, соединяющую значения функции, вычисленные в заданных точках (x, y) . Значит, чтобы получить гладкий график, необходимо вычислить значение функции в большом количестве точек.

Задание. Для построения графика функции $y = e^{\cos x}$ на промежутке $[-2\pi; 2\pi]$ с шагом 0.1 наберите в командном окне Scilab следующее:

```
--> x = -2 * %pi : 0.1 : 2 * %pi;
```

```
--> y = exp(cos(x));
```

```
--> plot(x, y)
```

2.3.2. Для построения нескольких графиков в одном окне можно использовать обращение к функции `plot` следующего вида:

$$\text{plot}(x_1, y_1, x_2, y_2, x_3, y_3, \dots)$$

Здесь x_1, y_1 – список значений переменной и первой функции, x_2, y_2 – второй, x_3, y_3 – третьей и т. д. Таким образом строятся графики практически какого угодно количества функций.

Задание. Для построения графиков функций $y = e^{\cos x}$ и $z = \sin x$ на промежутке $[-2\pi; 2\pi]$ с шагом 0.1 введите в командное окно Scilab:

```
--> x = -2 * %pi : 0.1 : 2 * %pi;
```

```
--> y = exp(cos(x));
```

```
--> z = sin(x);
```

```
--> plot(x, y, x, z)
```

2.4. Видоизменение графиков при помощи функции plot

Вид графика можно изменять, добавив при обращении к функции `plot`, помимо основных, еще один аргумент – строку, состоящую из трех символов, определяющих цвет линии, тип маркера, которым будет нарисован график, и тип линии. Обращение к функции `plot` в этом случае будет выглядеть так:

$$\text{plot}(x_1, y_1, \text{string}_1, x_2, y_2, \text{string}_2, \dots)$$

Строка `string` выглядит следующим образом:

‘параметр1параметр2параметр3’

Параметры пишутся один за другим без разделителей.

Параметр 1 определяет цвет графика (табл. 3).

Таблица 3

Цвет графика

Символ	Описание
y	Желтый
m	Розовый
c	Голубой
r	Красный
g	Зеленый
b	Синий
w	Белый
k	Черный

Параметр 2 задает маркер для рисования графика (табл. 4).

Таблица 4

Маркер для рисования графика

Символ	Описание маркера
.	Точка
o	Кружок
x	Крестик
+	Знак «плюс»
*	Звездочка
s	Квадрат
d	Ромб
v	Треугольник вершиной вниз
^	Треугольник вершиной вверх
<	Треугольник вершиной влево
>	Треугольник вершиной вправо
p	Пятиконечная звезда
h	Шестиконечная звезда

Параметр 3 устанавливает тип линии графика (табл. 5).

Таблица 5

Тип линии графика

Символ	Описание
-	Сплошная
:	Штрих-пунктирная
--	Штриховая

Если один из символов не указан, его значение выбирается по умолчанию (как правило, синяя сплошная линия). При неуказанном типе символа – значение будет отсутствовать.

Задание. Выполните построение трех графиков:

- 1) $f_1(x) = |3x|$ на интервале $[-3; 3]$ – сплошная синяя линия;
- 2) $f_2(x) = 8\sin x$ на интервале $[-\pi; \pi]$ – пунктирная красная линия;
- 3) $f_3(x) = e^x / 10$ на интервале $[-1; 4]$ – график, нарисованный зелеными кружочками.

Шаг изменения аргументов функций примите 0.2.

Наберите в командном окне следующее:

```
--> x1 = -3:0.2:3; f1 = abs(3 * x1);  
--> x2 = -%pi:0.2:%pi; f2 = 8 * sin(x2);  
--> x3 = -1:0.2:4; f3 = exp(x3) / 10;  
--> plot(x1, f1, 'b-', x2, f2, 'r-', x3, f3, 'go')
```

Убедитесь в правильности работы программы.

2.5. Оформление графиков

Оформление графиков выполняется следующими средствами.

Для вывода названия графика и осей используется команда:

```
xtitle('plot_name', 'x_name', 'y_name'),
```

где `plot_name` – название (заголовок) графика; `x_name` – название оси X; `y_name` – название оси Y.

Чтобы отобразить сетку на графике, следует воспользоваться оператором:

```
xgrid(n)
```

Здесь `n` – целое число, отвечающее за цвет сетки. По умолчанию, если параметр `n` не указан, то сетка имеет черный цвет.

Описание линий (так называемая легенда) осуществляется командой:

```
legend(line1, line2, ..., linen, place, frame),
```

где `line1` и другие – названия графиков; `place` – местоположение описания: 1 – верхний правый угол графического окна; 2 – верхний левый угол; 3 – нижний левый угол; 4 – нижний правый угол; 5 – определяется

пользователем после изображения графика; `frame` – задает рамку для описания: `%t` – описание заключено в рамку; `%f` – рамки нет.

Задание. Нужно построить график функций $f_1(x) = \sin(x)$ и $f_2(x) = \cos(x)$ на промежутке $[-2\pi; 2\pi]$ с шагом 0.2. Оформите его, подписав сам график, оси и отобразив описание линий и сетку.

Для разработки программы удобно создать скрипт-файл:

```
x = -2 * %pi : 0.2 : 2 * %pi;  
f1 = sin(x);  
f2 = cos(x);  
plot(x, f1, x, f2);  
xtitle ('Графики функций f1(x) и f2(x)', 'Ось X', 'Ось Y');  
xgrid();  
legend('f1(x)', 'f2(x)', 3, %f);
```

Сохраните скрипт-файл в папке Lab5 под именем, например, `grafic1.sce`. Запустите скрипт из окна редактора. Убедитесь в правильности работы программы.

2.6. Построение нескольких графиков в одном графическом окне

В Scilab можно выводить несколько графиков в одном окне, не совмещая их в одних координатных осях. Например, если графическое окно должно содержать 4 самостоятельных графика, все окно разбивается на 4 области (подокна), а затем в каждую из них выводится график функции.

Наиболее простым способом изображения нескольких графиков в одном окне является использование функции `subplot`. Обращение к ней имеет вид:

```
subplot(m, n, p)
```

или

```
subplot(mnp)
```

Выполнение этой функции приводит к тому, что графическое окно разбивается на m окон по вертикали и n окон по горизонтали, текущим окном становится окно с номером p . Для формирования графика в каждом окне можно использовать функцию `plot`.

Пример. Построить графики шести функций: $y = \sin(x)$; $z = \cos(x)$; $u = \cos(\sin(x))$; $v = \sin(\cos(x))$; $w = \exp(\sin(x))$; $r = \exp(\cos(x))$ в одном графическом окне, каждый в своей системе координат. Аргумент x изменяется на интервале $[-10; 10]$ с шагом 0.01.

С помощью функции `subplot` разбиваем графическое окно на 6 областей. Примем, что в каждом столбце по вертикали должно быть 3, а по горизонтали – 2 области для вывода графиков. Третье число в записи функции указывает, в какую из областей (счет ведется по порядку: слева направо и сверху вниз) выводится график, формируемый функцией `plot(x, y)`.

Задание. Напишите программу для рассмотренного примера. В программе ожидается много команд, поэтому удобно оформить ее в виде скрипт-файла. С этой целью откройте редактор SciNotes и запишите текст:

```
// Скрипт формирования шести графиков функций
x = [-10:0.01:10];
y = sin(x), z = cos(x);
u = cos(sin(x)), v = sin(cos(x));
w = exp(sin(x)); r = exp(cos(x));
subplot(3, 2, 1);
plot(x, y);
subplot(3, 2, 2);
plot(x, z);
subplot(3, 2, 3);
plot(x, u);
subplot(3, 2, 4);
plot(x, v);
subplot(3, 2, 5);
plot(x, w);
subplot(3, 2, 6);
plot(x, r);
```

Сохраните скрипт-файл в папке Lab5 под именем, например, grafic2.sce.

Затем запустите скрипт из окна редактора. По содержимому графического окна убедитесь в правильности решения поставленной задачи.

2.7. Функция *plot2d*

Функция `plot2d` предоставляет пользователю больше возможностей для оформления графиков. Простейшее обращение к этой функции осуществляется командой:

$$\text{plot2d}(x, y)$$

За внешний вид графика отвечают ключи команды, задаваемые в виде `key = value`:

$$\text{plot2d}(x, y, \text{key1}=\text{value1}, \text{key2}=\text{value2}, \text{key3}=\text{value3}, \dots)$$

Возможные значения ключей:

1) `style` – определяет цвет графика, который задается указанием RGB-кода: `style = color(r, g, b)`. Каждое из значений `r`, `g`, `b` соответствует «количеству» красного, зеленого и синего, и может меняться в пределах от 0 до 255. Есть более простая возможность – указать непосредственно имя цвета: `style = color('имя цвета')`. Наиболее распространенные названия даны в табл. 6;

Таблица 6

Имена цветов

Цвет	Имя	Цвет	Имя
yellow	Желтый	blue	Синий
cyan	Голубой	black	Черный
red	Красный	brown	Коричневый
green	Зеленый	grey	Серый

2) `rect` – устанавливает размер окна вокруг графика: значение этого ключа задается в виде вектора `rect = [xmin, ymin, xmax, ymax]`, где `xmin`, `xmax` – интервал отображения графического окна по оси `x`; `ymin`, `ymax` – по оси `y`;

3) `axesflag` – задает наличие рамки вокруг графика; базисные значения этого параметра: 0 – нет рамки; 1 – изображение рамки, ось слева; 2 – изображение рамки без подписи осей; 3 – изображение

осей, ось справа; 5 – изображение осей в виде перекрестка в начале координат. По умолчанию принято значение 1;

4) `naх` – устанавливает число значений координатных осей; его значение представляет собой массив `naх = [nx, Nx, ny, Ny]`, где `Nx (Ny)` – число основных делений оси `X (Y)`, которые подписаны на графике; `nx (ny)` – число промежуточных делений;

5) `leg` – определяет легенду для каждого графика; задается в виде:

$$\text{leg} = \text{'leg1@leg2@leg3...'},$$

где `leg1` – описание первого графика; `leg2` – второго и т. д.

Пример. Построение графика функции $f(x) = \exp(\sin(x))$ на промежутке $[-2\pi; 2\pi]$ с шагом 0,2 со следующим оформлением: линия графика зеленого цвета; размеры графического окна: от -10 до 10 – по оси `x` и от -1 до 4 – по оси `y`; оси должны быть изображены в виде перекрестия; количество основных меток: 11 – по оси `x` и 6 – по оси `y`; вспомогательных – по 2 для обеих осей; отобразить легенду.

Задание. Наберите в командном окне программу для построения графика из рассмотренного примера:

```
--> x = -2 * %pi : 0.2 : 2 * %pi; f = exp(sin(x));  
--> plot2d(x, f, style = color('green'), rect = [-10, -1, 10, 4], ...  
--> axesflag = 5, naх = [2, 11, 2, 6], leg = 'f(x) = exp(sin(x))'
```

Сравните полученный график с заданным в примере.

2.8. Построение точечных графиков

Многие прикладные задачи (например, обработка и анализ экспериментальных данных) требуют построения точечного графика. Для этой цели можно использовать функцию `plot2d` в следующей форме:

$$\text{plot2d}(x, y, d),$$

где `x, y` – массивы координат точек, которые нужно отобразить на графике; `d` – отрицательное число, определяющее тип маркера (табл. 7).

Таблица 7

Тип маркера

Число	Описание
-0	Точка
-1	Плюс

Число	Описание
-2	Крестик
-3	Плюс, вписанный в окружность
-4	Закрашенный ромб
-5	Незакрашенный ромб
-6	Треугольник вершиной вверх
-7	Треугольник вершиной вниз
-8	Плюс, вписанный в ромб
-9	Кружок
-10	Звездочка
-11	Квадрат
-12	Треугольник вершиной вправо
-13	Треугольник вершиной влево
-14	Пятиконечная звезда

Массивы координат могут быть вычислены или введены непосредственно.

Пример 1. Построить точечный график функции $y = \sin(x)$ с типом маркера «плюс, вписанный в ромб». Диапазон изменения аргумента: $-2\pi \leq x \leq 2\pi$ с шагом 0.25.

Задание. Введите в командное окно программу для построения графика:

```
--> x = -2 * %pi : 0.25 : 2 * %pi;
```

```
--> y = sin(x);
```

```
--> plot2d(x, y, -8)
```

Сравните полученный график с заданным в примере 1.

Пример 2. Требуется построить точечный график функции $y = \sin(x)$ на промежутке $[0; 3]$ с шагом 0.15 крестиками и нанести на этот же график точки с координатами (0.1; 0.3), (0.4; 0.8), (1; 0.6), (2; 0.1), (2.5; 0.4) кружками.

Задание. Наберите в командном окне программу для построения графика из примера 2:

```
--> x = 0 : 0.15 : 3; y = sin(x);
```

```
--> x1 = [0.1, 0.4, 1, 2, 2.5]; y1 = [0.3, 0.8, 0.6, 0.1, 0.4];
```


--> plot2d(x, y, -2)

--> plot2d(x1, y1, -9)

Сравните полученные графики с заданными в примере 2.

3. Задания для самостоятельной работы

Указание. Отчет по лабораторной работе № 5 выполняется в виде списка скрипт-файлов, разработанных в ходе выполнения заданий для самостоятельной работы.

Задание 1. Разработайте скрипт для построения графика функции:

$$y = 3.5^{-0.5x} \cos 6x$$

для диапазона $-2 \leq x \leq 4$ с шагом 0.01. Используйте функцию plot. Скрипт-файл сохраните в папке Lab5 под именем zadanie1.sce. Проверьте работу скрипта, запустив его из окна редактора.

Задание 2. Разработайте скрипт для построения графиков функций:

$$y1(x) = x \ln \frac{1}{|x|}; \quad y2(x) = 1 + 10 \frac{\sin x}{x}$$

на интервале $0.3 \leq x \leq 10$ с шагом 0.1. Линия графика $y1$ – красного цвета, сплошная; линия графика $y2$ – зеленого цвета, пунктирная. Сделайте заголовок графика и заголовки осей. Нанесите сетку черного цвета. Разместите легенду в правом верхнем углу с рамкой.

Разработанный скрипт-файл сохраните в папке Lab5 под именем zadanie2.sce. Проверьте работу скрипта, запустив его из окна редактора.

Задание 3. Разработайте скрипт для построения четырех графиков функций:

$$y = \sin(2x); \quad z = \cos(3x); \quad u = \cos(\sin(2x)); \quad v = \sin(\cos(3x))$$

в одном графическом окне, каждый в своей системе координат. Примите, что x изменяется на интервале $[-10 : 10]$ с шагом 0.01. Скрипт-файл сохраните в папке Lab5 под именем zadanie3.sce. Проверьте работу скрипта, запустив его из окна редактора.

4. Содержание отчета

Отчет по лабораторной работе № 5 выполняется в виде списка скрипт-файлов, разработанных в ходе выполнения заданий для самостоятельной работы.

Разработанные скрипты необходимо продемонстрировать преподавателю для оценки объема выполненной работы и анализа качества выполнения заданий.

Контрольные вопросы

1. Как с помощью функции `plot()` вывести несколько графиков в одном графическом окне?
2. Как можно задать цвет и тип линии графика в функции `plot()`?
3. Каким способом можно задать заголовок графика, выведенного с помощью функции `plot()`?
4. Как сделать надписи на осях графика, выведенного с помощью функции `plot()`?
5. Какие возможности для вывода имеет функция `plot2d()`?
6. Как можно построить график точечной функции в Scilab?

Литература

1. Алексеев, Е. Р. Scilab: решение инженерных и математических задач / Е. Р. Алексеев, О. В. Чеснокова, Е. А. Рудченко. – М. : ALT Linux ; Бином. Лаб. зн., 2008. – Режим доступа: <https://docs.altlinux.org/books/2008/altlibrary-scilab-20090409.pdf>.
2. Панкратов, И. А. Scilab. Первые шаги : учеб. пособие / И. А. Панкратов. – Саратов : СГУ, 2012. – Режим доступа: http://www.sgu.ru/sites/default/files/textdocsfiles/2013/12/10/scilab_first_steps.pdf.
3. Решение инженерных задач в среде Scilab : учеб. пособие / А. Б. Андриевский [и др.]. – СПб. : НИУ ИТМО, 2013. – Режим доступа: <http://www.boors.ifmo.ru/file/pdf/1366.pdf>.

Содержание

Введение.....	3
<i>Лабораторная работа № 1. Начало работы в среде разработки Scilab</i>	4
<i>Лабораторная работа № 2. Создание массивов</i>	19
<i>Лабораторная работа № 3. Математические операции с массивами</i>	32
<i>Лабораторная работа № 4. Управление вводом и выводом данных в скрипт-файлах</i>	42
<i>Лабораторная работа № 5. Построение и оформление графиков функций</i>	55
Литература	67

Учебное электронное издание комбинированного распространения

Учебное издание

Виноградов Эдуард Михайлович
Крышнев Юрий Викторович
Столбов Михаил Викторович

ОСНОВЫ ТЕОРИИ СИСТЕМ

Практикум

**по выполнению лабораторных работ для студентов
специальности 1-53 01 07 «Информационные технологии
и управление в технических системах»
дневной формы обучения**

Электронный аналог печатного издания

Редактор
Компьютерная верстка

Т. Н. Мисюрова
Н. Б. Козловская

Подписано в печать 11.07.17.

Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».

Ризография. Усл. печ. л. 4,18. Уч.-изд. л. 4,45.

Изд. № 80.

<http://www.gstu.by>

Издатель и полиграфическое исполнение
Гомельский государственный
технический университет имени П. О. Сухого.
Свидетельство о гос. регистрации в качестве издателя
печатных изданий за № 1/273 от 04.04.2014 г.
пр. Октября, 48, 246746, г. Гомель