МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

КАФЕДРА СИСТЕМ СБОРА И ОБРАБОТКИ ДАННЫХ



**ОТЧЕТ**

по лабораторной работе №2

по теме: «Построение трехмерных графиков. Программирование в MatLAB»

по дисциплине «Теория информационных процессов и систем»

Факультет: АВТ Преподаватель:

Группа: АТ-74 Профессор

Выполнил: Назьмов Александр Рабинович Е.В.

Новосибирск

2020 г.

## **Цель работы:**

Целью лабораторной работы является изучение языка программирования среды MatLAB. Основной задачей ставится изучение типовых конструкций языка, таких как функции, условия и циклы. Также внимание уделяется построению трехмерных графиков.

## **Задание на лабораторную работу:**

1. Написать программу (m-файл), которая строит график сферы или квадратной пирамиды по выбору пользователя. Выбор фигуры и длины запрашиваются у пользователя в процессе работы функции:

* для сферы – это идентификатор и радиус – *‘сфера’*, *R*.
* для пирамиды – это идентификатор, высота, длина стороны основания – *‘пирамида’, h, l.*

Код для Matlab представлен в [приложении А.](#_Приложение_А) На рисунках 1, 2, представлены результаты работы скрипта, который строит сферу, пирамиду соответственно.

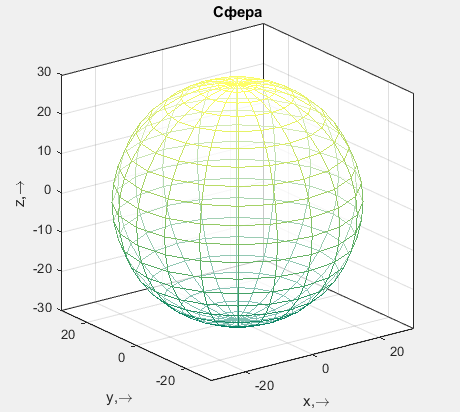
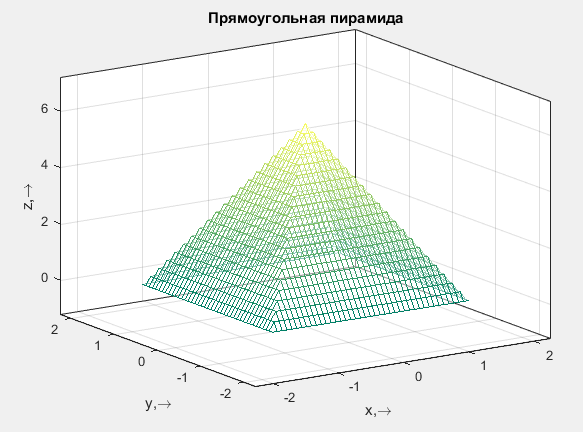
 

Рис.1.Построение сферы радиусом 30 Рис.2. Построение пирамиды.

Высота основания 6. Длина ребра 3

1. Написать программу (m-файл), которая строит график куба или конуса по выбору пользователя. Выбор фигуры задается как входные параметры функции:

* для куба – это идентификатор и длина ребра – *‘куб’, r.*
* для конуса – это идентификатор, высота, радиус основания – *‘конус’, h, r.*

Код для Matlab представлен [в приложении Б.](#_Приложение_Б) На рисунках 3, 4, представлены результаты работы скрипта, который строит куб и конус соответственно.

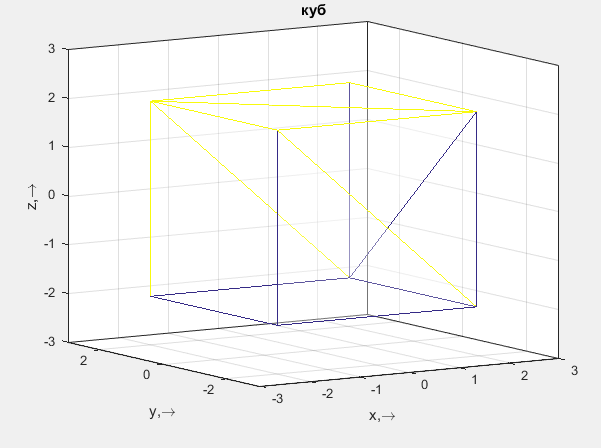
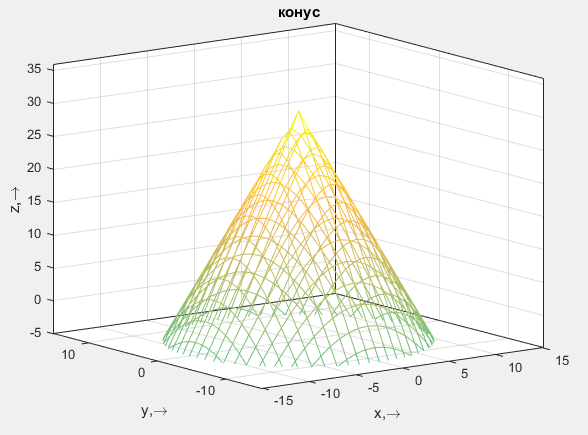
 

Рис.3.Построение куба. Длина ребра 4 Рис.4. Построение конуса. Высота 30.

Радиус основания 10

1. Написать программу (m-файл), которая создает трехмерную матрицу и «раскрашивает» её значениями в шахматном порядке по всему объему. Значение раскрашивания зависят от входного параметра идентификации *id*:

* *id* = *‘digit’* – матрица заполняется числами *1* и *0;*
* *id* = *‘char’* – матрица заполняется буквами *‘1’* и *‘0’.*

Входные параметры – это размерность матрицы по трём координатам, идентификатор.

Выходной – сама матрица.

Код для Matlab представлен в [приложении В](#_Приложение_В). На рисунках 5, 6, представлены результаты работы скрипта для матриц с разными параметрами идентификации id.

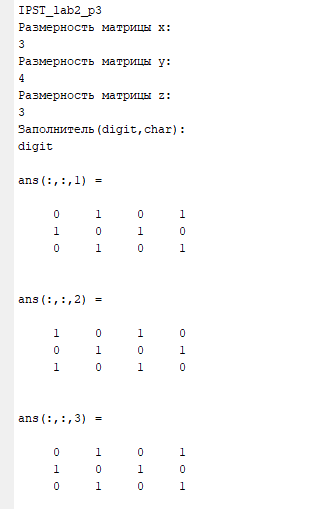
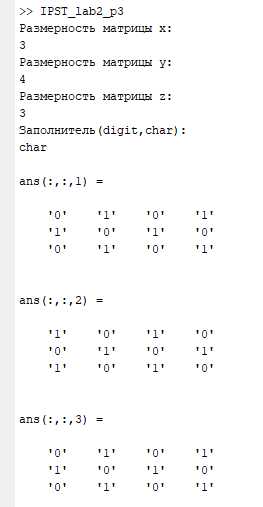
 

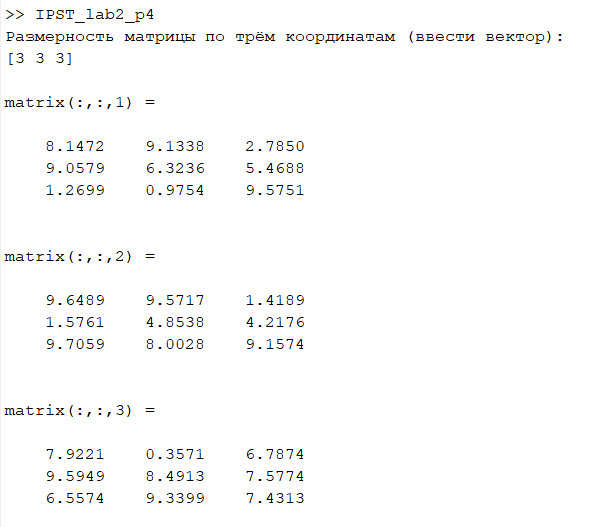
Рис.5.Трехмерная матрица с *id* = *‘digit’* Рис.6. Трехмерная матрица с *id = ‘char’*

1. Написать программу (m-файл), которая создает трёхмерную матрицу, заполненную случайными целыми числами (от 0 до 10) и производит арифметические поэлементные действия между двумерными массивами этой матрицы (по любой координате внутри матрицы). Значения операции зависит от идентификатора:

* *id = ‘+’*
* *id = ‘–’*
* *id = ‘\*’* (поэлементное умножение).

Размерность матрицы по трём координатам и идентификатор запрашивается у пользователя в процессе работы функции. Выходной параметр – полученная матрица.

Код для Matlab представлен в [приложении Г](#_Приложение_Г). Итоговая матрица является результатом применения выбранной операции (суммирования) вдоль строк исходной матрицы. На рисунках 7, 8 представлены результаты работы скрипта.



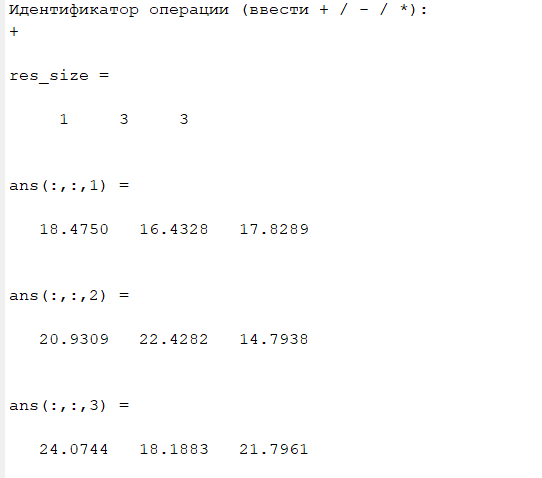


Рис.7. Исходная матрица

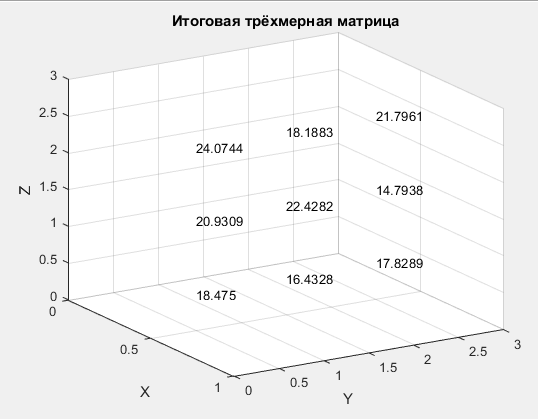


Рис.8. Итоговый график трехмерной матрицы при id = ‘+’

1. Написать программу (m-файл), которая раскрашивает черно-белый рисунок красным, зеленым или синим цветом. Цвет зависит от входного параметра:

* *id = 'красный';*
* *id = 'зеленый';*
* *id = 'синий'.*

Входной параметр – раскрашиваемый цвет.

Код для Matlab представлен в [приложении Д](#_Приложение_Д). На рисунках 10, 11, 12, 13 представлены результаты работы скрипта.

Рис.9. Исходное изображение Рис.10.Результат при *id = 'зеленый'*

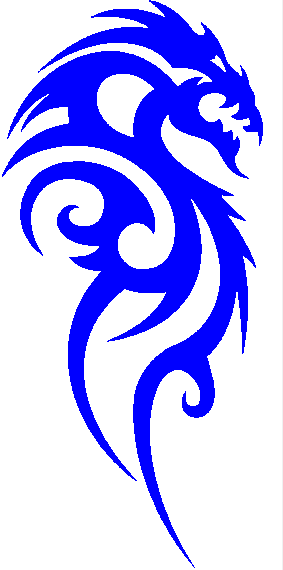


Рис.11.Результат при *id = 'красный'* Рис.12.Результат при *id = 'синий'*

1. Написать программу (m-файл), которая инвертировала цвета черно-белого изображения (черное в белое, белое в черное) и поворачивала рисунок на угол кратный 90° вокруг своей оси. Угол поворота зависит от входного параметра:

* id = 90 – поворот на 90°;
* id = 180 – поворот на 180°;
* id = 270 – поворот на 270°;
* id = 0 – нет поворота;

Код для Matlab представлен в [приложении Е.](#_Приложение_Е) На рисунках 14, 15, 16, 17 представлены результаты работы скрипта инвертирования цвета и поворота черно-белого изображений .

Рис.13. Исходное изображение Рис.14. Угол поворота 0° Рис.15. Угол поворота 180°



Рис.16. Угол поворота 90° Рис.17. Угол поворота 270°

## Вывод:

Ознакомились с основами работы в среде MatLAB на базовом уровне: изучили типовые конструкции языка, такие как: цикл с известным количеством повторений for, операторы проверки условия if, elseif, которые позволяют организовать разветвление исполнения программы. Так как в MatLAB оптимизированный матричные операции, они выполняются значительно быстрее, чем равносильные вычисления в цикле, поэтому циклами следует пользоваться в действительно нужных ситуациях.

Один из видов представления строк в MatLAB - это вектор с элементами-кодами символов. Поэтому для таких строк доступны все матричные операции и дополнительные функции для работы со строками. Чтобы хранить в массиве несколько строк и не происходила их автоматическая конкатенация можно использовать ячеистый (cell) массив.

Параметры, требуемые для выполнения пользовательской функции, можно передавать в качестве входных аргументов функции или интерактивно - в консоль во время работы функции при запросе.

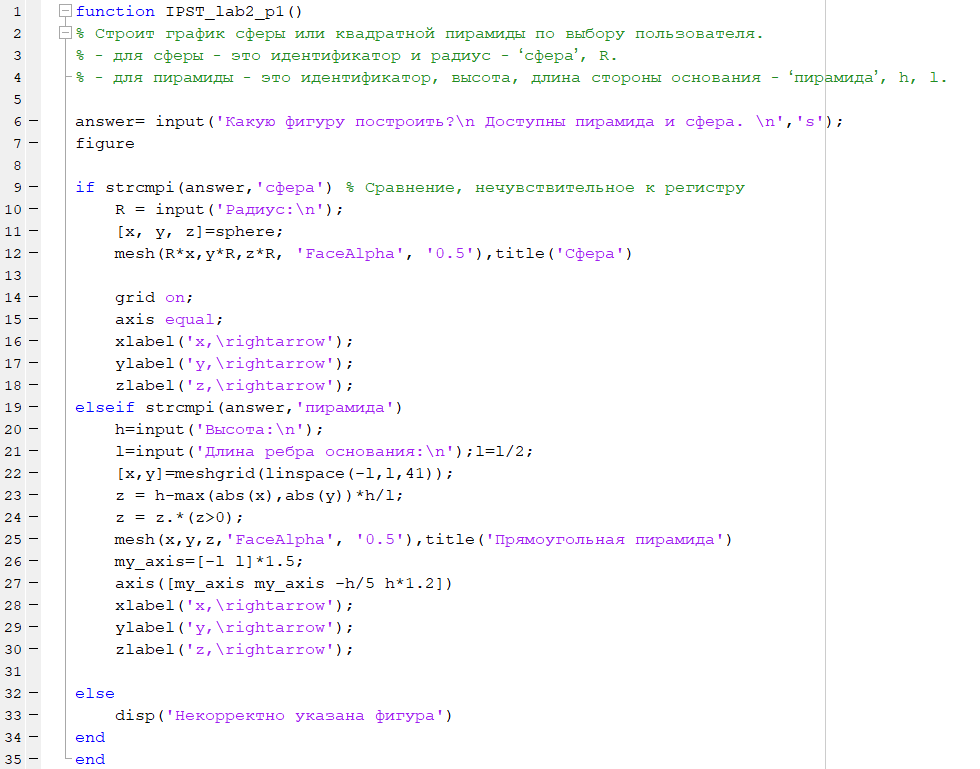
К элементам матриц можно обращаться на основе условий (тогда формируется соразмерный тензор-маска из значений типа logical) или последовательности индексов, таким образом производя быструю выборку.

Большое внимание уделили построению трехмерных графиков. Для построения трехмерных графиков, изучили функцию [X,Y]=meshgrid(x,y) которая создает сетку координат на плоскости. А также функции mesh, surf и surfl, позволяющих добиваться большей реалистичности в изображении трехмерных графиков. Команда *axis equal* устанавливает масштаб, который обеспечивает одинаковые расстояния между метками по осям x и y, для лучшего просмотра графиков.

Растровое изображение может быть представлено двумерной матрицей значений яркости, если оно бинарное или полутоновое; тремя такими матрицами, если оно состоит из каналов - красного, синего, зеленого.

Все задания данной лабораторной работы выполнены в файл-функциях , для использования в других скрип-файлах , а также для удобства многократного исполнения кода.

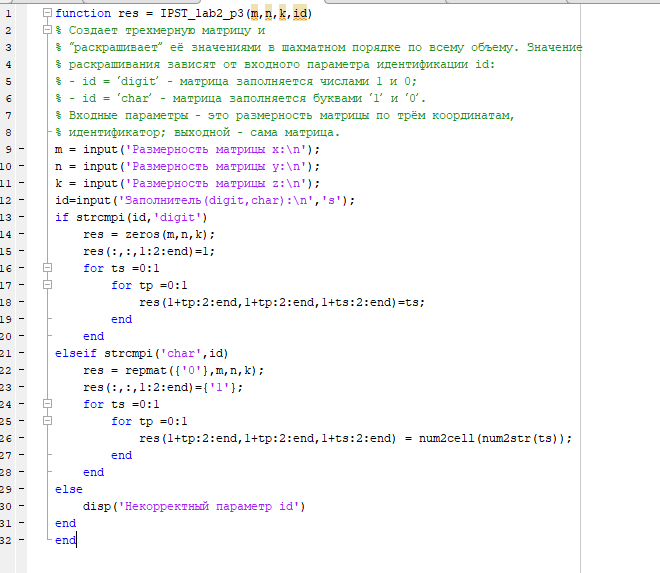
## Приложение А



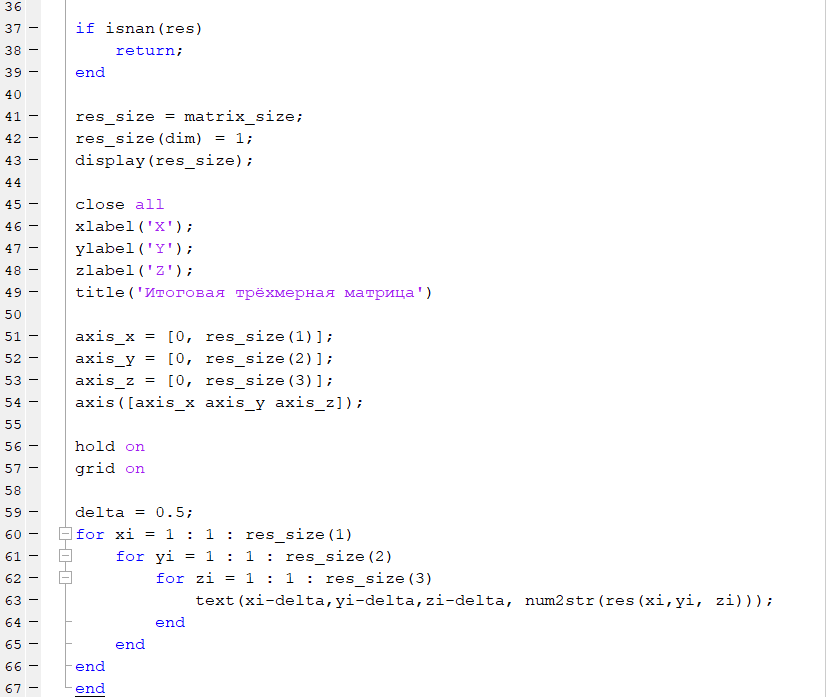
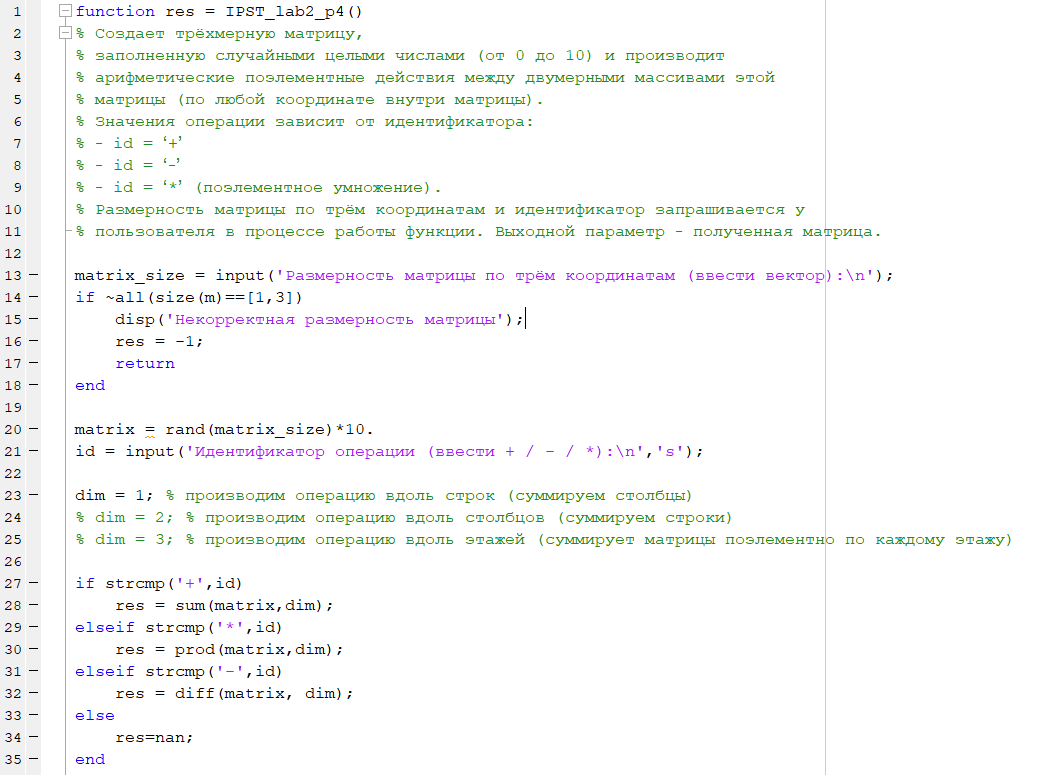
## Приложение Б

## 

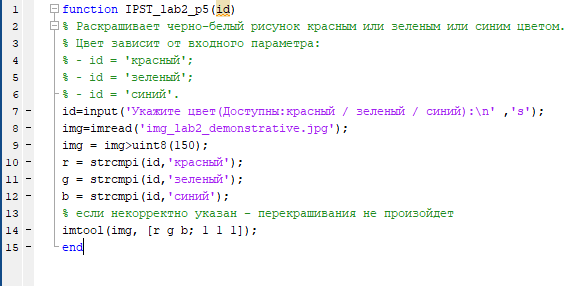
## Приложение В



## Приложение Г



## Приложение Д



## Приложение Е

