

РГПУ им. А. И. Герцена

К работе допущены _____

Работа выполнена _____

Отчёт сдан _____

Отчет по лабораторной работе №3
«РЕШЕНИЕ ТИПОВЫХ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ В
СИСТЕМЕ MATLAB»

Работу выполнили:

Беспалов Вячеслав

Факультет: ИКНиТО

Направление: ИСТ

Группа: №1

Санкт-Петербург, 2020

Вариант 7

Задача 0

Условие

- написать файл-сценарий (*script*-файл), в котором пользовательский интерфейс оформлен в виде меню. Выбранный пункт меню определяет выполнение файла, соответствующего пункту индивидуального задания. При написании программы для реализации меню использовать *встроенную функцию menu()*, *оператор цикла с предусловием while* и *функцию eval()*;

Программа

```
while 1
    clear()
    choice = menu('Выберите задание', '#1', '#2', '#3', '#4', '#5', 'Выход')
    clc
    switch choice
        case 1
            run("Task1.m")
        case 2
            run("Task2.m")
        case 3
            run("Task3.m")
        case 4
            run("Task4.m")
        case 5
            run("Task5.m")
        case 6
            clear()
            break
    end
end
```

Результаты тестирования



Задача 1

Условие

- для пункта №1 при вычислении корней полинома построить график полинома и отобразить на нем найденные действительные корни. Ввод границ построения графика должен осуществляться с клавиатуры;

1. Вычислить корни полинома

$$x^5 + 2x^2 - x + 11$$

Программа

```
func = @(x) x.^5 + 2 * x.^2 - x + 11;
roots = fzero(func, 0)

start = input('Начало интервала: ');
stop = input('Конец интервала: ');

interval = start:0.05:stop;
figure(1); plot(interval, arrayfun(func, interval), 'color', [0 0.7 0]); title('fn'); hold on;
plot(roots, arrayfun(func, roots), 'r*');
line(get(gca, 'xlim'), [0 0], 'color', [0 0 0]);
line([0 0], get(gca, 'ylim'), 'color', [0 0 0]); grid on;
text(ceil(stop), 0, 'X'); text(0, ceil(func(stop)), 'Y');
```

Результаты тестирования

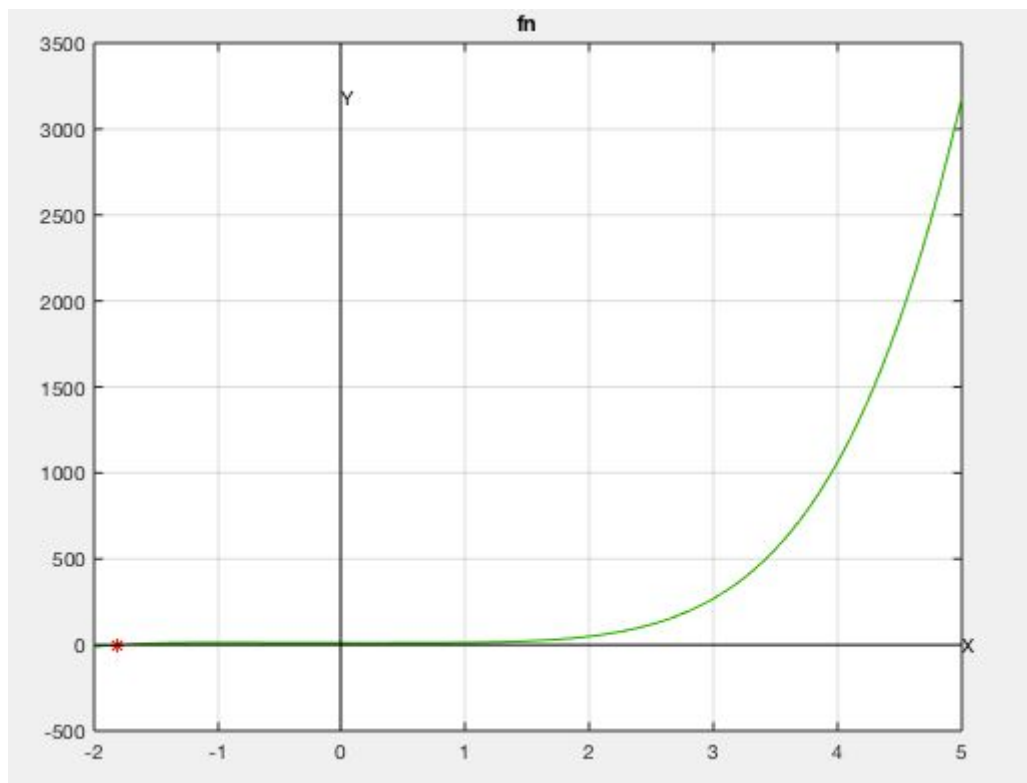
```
>> Task1
```

```
roots =
```

```
-1.8086
```

```
Начало интервала: -2
```

```
Конец интервала: 5
```



Задача 2

Условие

- для пункта №2 при решении системы линейных уравнений осуществить проверку полученного решения;

2. Найти решение системы линейных уравнений

$$\begin{cases} x_1 + x_2 + 2x_3 = -1 \\ 2x_1 - x_2 + 2x_3 = -4 \\ 4x_1 + x_2 + 4x_3 = -2 \end{cases}$$

Программа

```
syms x y z
eq1 = x + y + 2 * z == -1;
eq2 = 2 * x - y + 2 * z == -4;
eq3 = 4 * x + y + 4 * z == -2;

sol = solve([eq1, eq2, eq3], [x, y, z]);
fprintf('x = %0.5f; y = %0.5f; z = %0.5f\n', sol.x, sol.y, sol.z)
verif = isAlways(subs([eq1, eq2, eq3], sol));
assert(all(verif), 'Решение неверно')
```

Результаты тестирования

```
>> Task2
x = 1.00000; y = 2.00000; z = -2.00000
```

Задача 3

Условие

- уравнения и функции для выполнения пунктов № 3–5 описать в виде файлов-функций;
- для пункта №3 при поиске максимума и минимума функции построить график заданной функции в заданных границах и отобразить на нем полученные экстремумы маркерами разного цвета;

3. Найти значение локального минимума и максимума функции

$$f(x) = 2 \sin x^2 - x$$

$$1 \leq x \leq 3$$

Программа

```
function y = Function3(x)
    y = 2 * sin(x).^2 - x;
end
```

```

x = 1:0.05:3;
A = arrayfun(@Function3, x);
TF1 = islocalmax(A);
TF2 = islocalmin(A);

if all(TF2(:) == 0)
    [M,I] = min(A);
    TF2(I) = 1;
end

if all(TF1(:) == 0)
    [M,I] = max(A);
    TF2(I) = 1;
end
|
fprintf('Минимум: %0.5f; Максимум: %0.5f\n', A(TF2), A(TF1))

figure(1); plot(x, A, 'color', [0 0.7 0]); title('fn'); hold on;
    plot(x(TF1), A(TF1), 'm*');
    plot(x(TF2), A(TF2), 'r*');

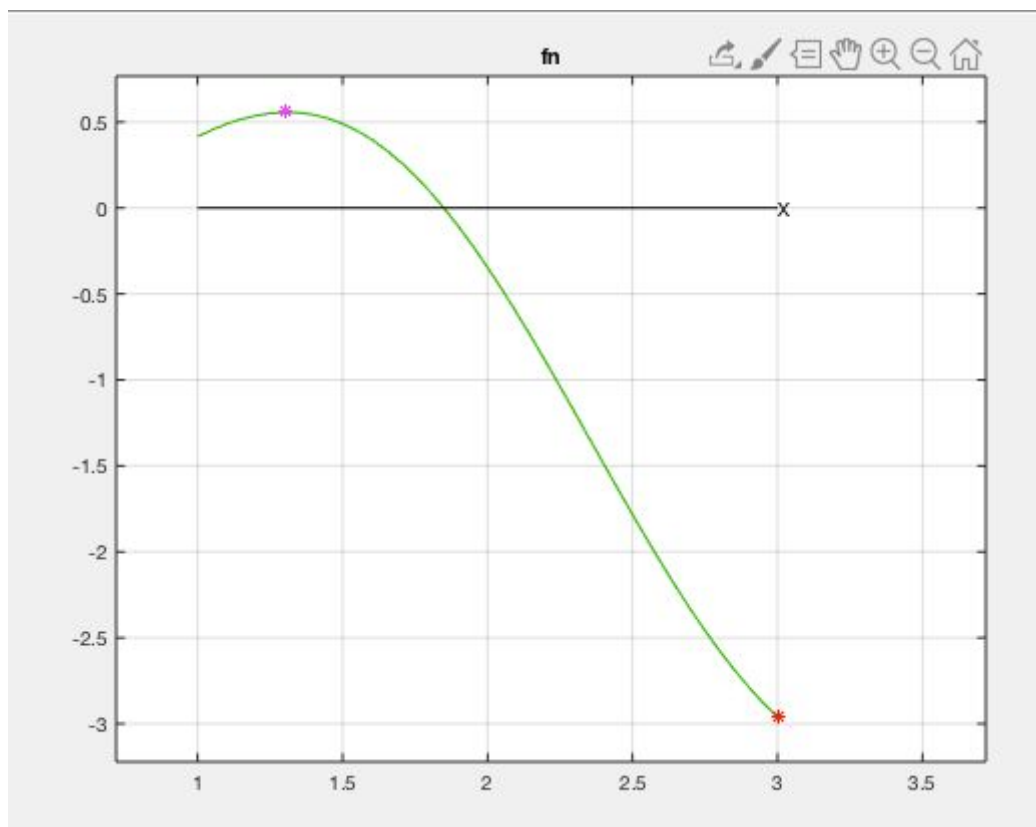
```

Результаты тестирования

```

>> Task3
Минимум: -2.96017; Максимум: 0.55689

```



Задача 4

Условие

- уравнения и функции для выполнения пунктов № 3–5 описать в виде файлов-функций;
- для пункта №4 при вычислении интеграла построить график подынтегральной функции, границы графика вводить с клавиатуры и закрасить площадь, ограниченную функцией на заданном отрезке;

4. Вычислить значение определенного интеграла

$$\int_{0,4}^{1,2} \frac{dx}{\sqrt{2 + 0,5x^2}}$$

Программа

```
function y = Function4(x)
    y = 1 ./ sqrt(2 + 0.5 .* x.^2);
end

I = integral(@(x) Function4(x), 0.4, 1.2)

start = input('Начало интервала: ');
stop = input('Конец интервала: ');

plotx = start:0.05:stop;
integralx = 0.4:0.05:1.2;
figure(1); plot(plotx, arrayfun(@Function4, plotx), 'color', [0 0.7 0]); title('fn'); hold on;
area(integralx, arrayfun(@Function4, integralx));
line(get(gca, 'xlim'), [0 0], 'color', [0 0 0]);
line([0 0], get(gca, 'ylim'), 'color', [0 0 0]); grid on;
text(ceil(stop), 0, 'X'); text(0, ceil(Function4(stop)), 'Y');
```

Результаты тестирования

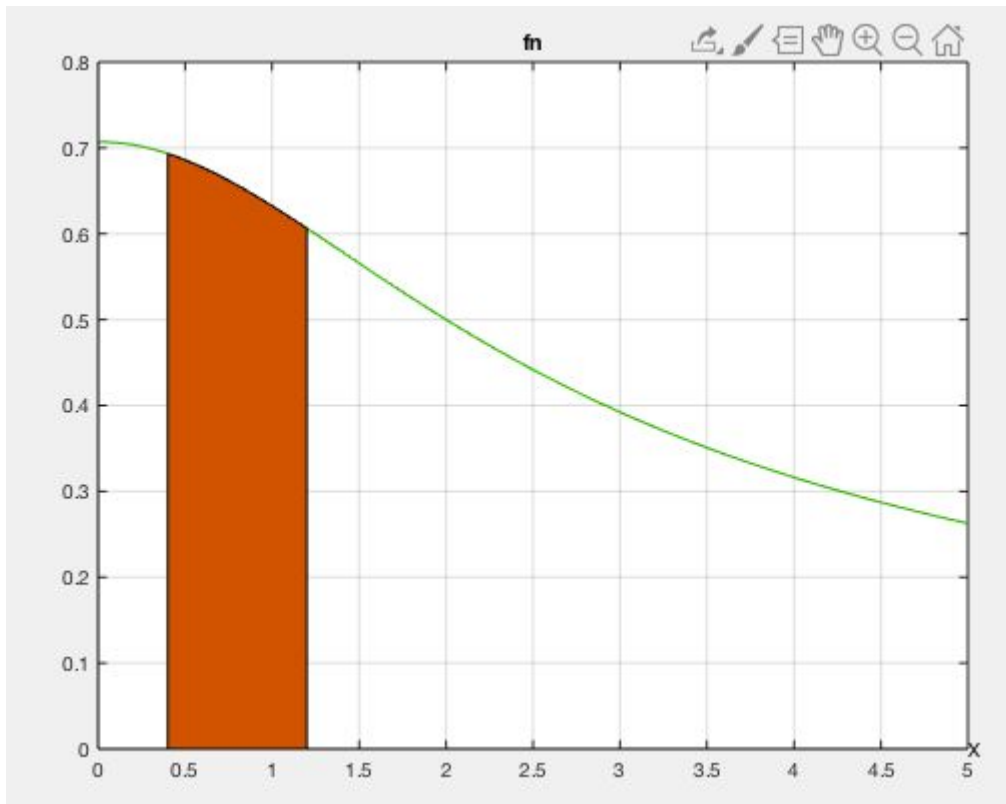
```
>> Task4
```

```
I =
```

```
0.5234
```

```
Начало интервала: 0
```

```
Конец интервала: 5
```

Задача 5

Условие

- уравнения и функции для выполнения пунктов № 3–5 описать в виде файлов-функций;
- для пункта №5 при решении трансцендентного уравнения построить график функции в границах, заданных пользователем в форме диалога, и отобразить на нем значение корня уравнения цветом, отличным от цвета графика;

5. Решить трансцендентное уравнение

$$x + \lg x - 0,5 = 0$$

Программа

```
function y = Function5(x)
    y = x + log10(x) - 0.5;
end
```



```

func = @(x) Function5(x);
roots = double(solve(func))

prompt = {'Начало графика:', 'Конец графика:'};
dlgtitle = 'Ввод';
dims = [1 35];
definput = {'0', '2'};
answer = str2double(inputdlg(prompt, dlgtitle, dims, definput));

interval = answer(1):0.05:answer(2);
figure(1); plot(interval, arrayfun(func, interval), 'color', [0 0.7 0]); title('fn'); hold on;
plot(roots, arrayfun(func, roots), 'r*');
line(get(gca, 'xlim'), [0 0], 'color', [0 0 0]);
line([0 0], get(gca, 'ylim'), 'color', [0 0 0]); grid on;
text(ceil(answer(2)), 0, 'X'); text(0, ceil(func(answer(2))), 'Y');

```

Результаты тестирования

```
>> Task5
```

```
roots =
```

```
0.6724
```

