Министерство образования Республики Беларусь

Учреждения образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

По дисциплине ЛОИС

**“Программирование в среде системы MATLAB**”

Выполнил

Студент группы ИИ-21

Романко Н.А.

Проверил

Скарубо А.О.

Брест 2023

**Цель работы:**

Изучить принципы создания М-функций и использования их в среде MATLAB, а также возможностей создания законченных приложений на базе графического интерфейса пользователя системы MATLAB.

**ЗАДАЧА**

***2.1 Вычисление определенных интегралов***

Реализовать М-функцию, вычисляющую интеграл вида

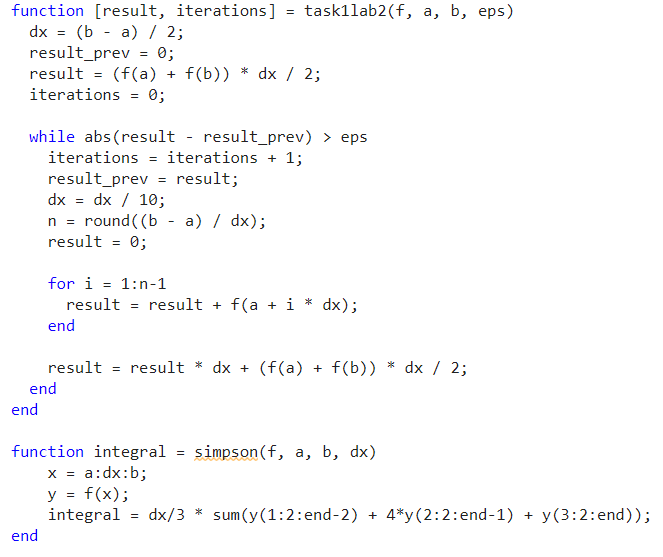
двумя методами интегрирования:

- методом *трапеций*, изменяя шаг интегрирования. Последовательно уменьшая шаги интегрирования в 10 раз (до dx = 0.00001), добиться тем самым стабилизации результата. Контроль достигнутой точности осуществлять путем сравнения последовательных результатов;

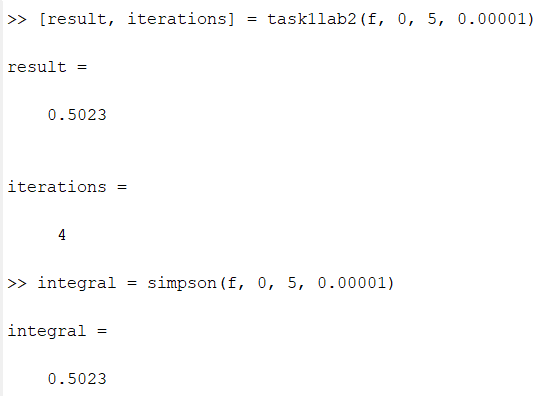
- методом Симпсона.

Программа должна запрашивать ввод метода с клавиатуры. Оформить выполнение метода Симпсона подфункцией (вспомогательной функцией). Определить и сравнить производительность М-функции для каждого метода.

Код программы:



Запуск и вывод программ двумя способами:



***2.2 Решение систем дифференциальных уравнений***

Реализовать сценарий для решения системы дифференциальных уравнений методом Рунге-Кутта 4-5 порядка.

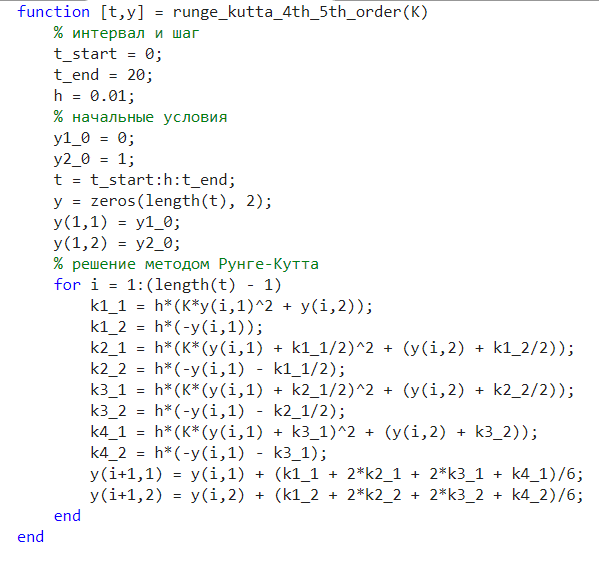


Начальные условия у1(0) = 0; у2(0) = 1. К – коэффициент нелинейности задачи.

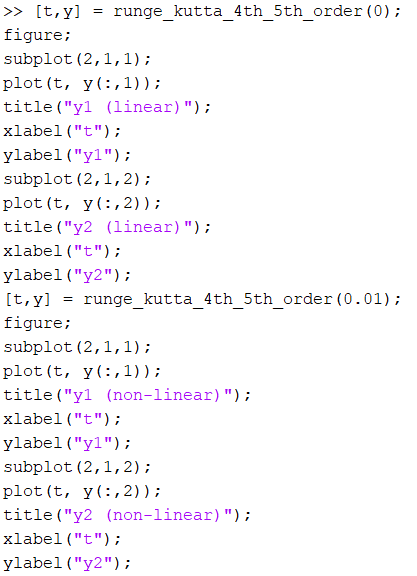
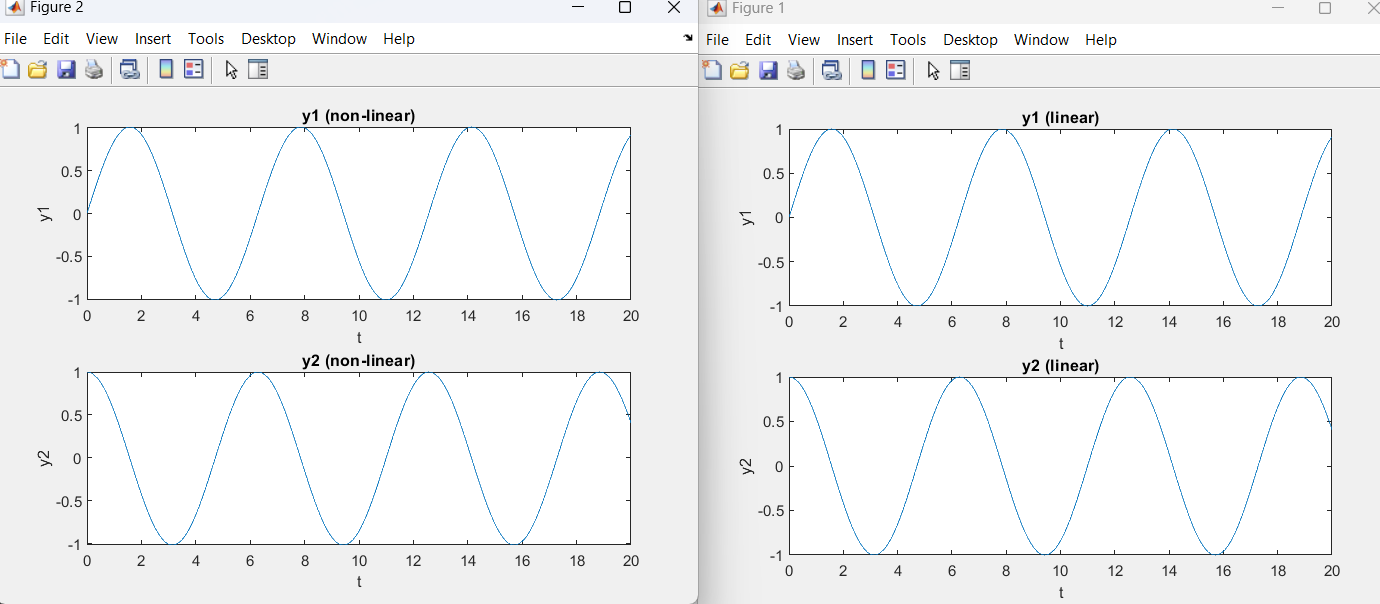
Найти решение системы дифференциальных уравнений на отрезке [0, 20] с заданными начальными условиями при К = 0 (задача становится чисто линейной и описывает гармонические колебания). Повторить решение для нелинейного случая, т.е при К= 0.01.

Визуализировать полученные результаты. Визуализацию осуществить с помощью функции plot (X, y1) и plot (X, y2), где у1 – первый столбец матрицы Y, а y2 – второй столбец матрицы Y, совместив оба графика в одном графическом окне. Линейность либо нелинейность задачи реализовать с помощью выбора кнопки меню.

Код программы:



Запуск программы и визуализация результатов:

***2.3 Решение системы жестких дифференциальных уравнений***

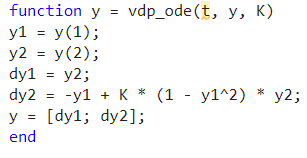
Реализовать сценарий для решения уравнение Ван-дер-Поля, которое описывает нелинейные релаксационные колебания в различных электронных устройствах имеет вид:



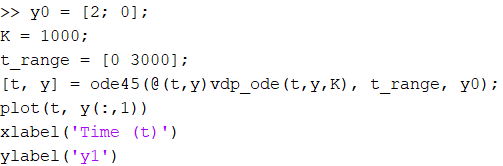
Начальные условия у1(0) = 2; у2(0) = 0; К = 1000.

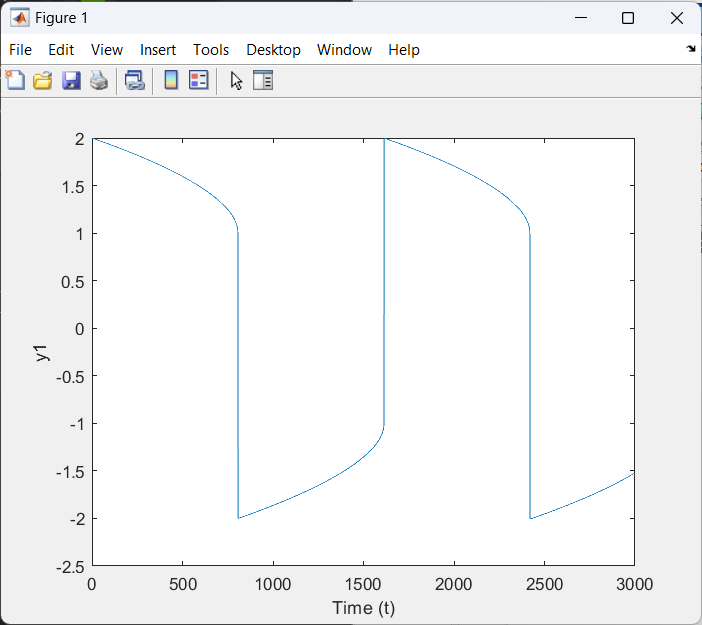
Найти решение данного уравнения на отрезке [0, 3000] с заданными начальными условиями. Визуализировать решение, а именно первый столбец матрицы Y. Оценить производительность решения.

Код программы:



Запуск и результат программы:





***2.4 Использование графических элементов управления***

Реализовать М-файл, создающий в графическом окне один объект axes, три редактируемых поля для ввода числовой информации, три текстовых поля, содержащих подписи редактируемых полей, одну командную кнопку. Использовать полученное графическое окно для построения графиков функций вида Y = f(X). Массив Х задается левой границей диапазона х0, приращением dx и верхней границей диапазона xn, которые вводятся с клавиатуры в редактируемые поля. По нажатию кнопки «Построить» вывести графическое меню выбора функции. В качестве функций использовать стандартные функции (встроенные в MATLAB, такие как sin, cos, tan, exp) и собственные М-функции. По выбору функции должен строиться ее график.

Код программы:

function untitled

% Create the figure window

f = figure('Position', [300 300 600 400]);

% Create the axes object

ax = axes('Parent', f, 'Position', [0.1 0.1 0.8 0.8]);

% Create the editable fields for entering numerical information

x0\_field = uicontrol('Parent', f, 'Style', 'edit', 'Position', [500 360 60 20], 'String', '0');

dx\_field = uicontrol('Parent', f, 'Style', 'edit', 'Position', [500 330 60 20], 'String', '0.1');

xn\_field = uicontrol('Parent', f, 'Style', 'edit', 'Position', [500 300 60 20], 'String', '1');

% Create the text fields containing editable field labels

x0\_label = uicontrol('Parent', f, 'Style', 'text', 'Position', [450 360 50 20], 'String', 'x0:');

dx\_label = uicontrol('Parent', f, 'Style', 'text', 'Position', [450 330 50 20], 'String', 'dx:');

xn\_label = uicontrol('Parent', f, 'Style', 'text', 'Position', [450 300 50 20], 'String', 'xn:');

% Create the command button

f = uicontrol('Style','edit','String','','Position',[20,50,200,25]);

uicontrol('Style','pushbutton','String','Build','Position',[20,20,70,25],'Callback',@build\_callback);

function build\_callback(hObject, eventdata)

func\_str = get(f,'String');

x = linspace(0,10,100);

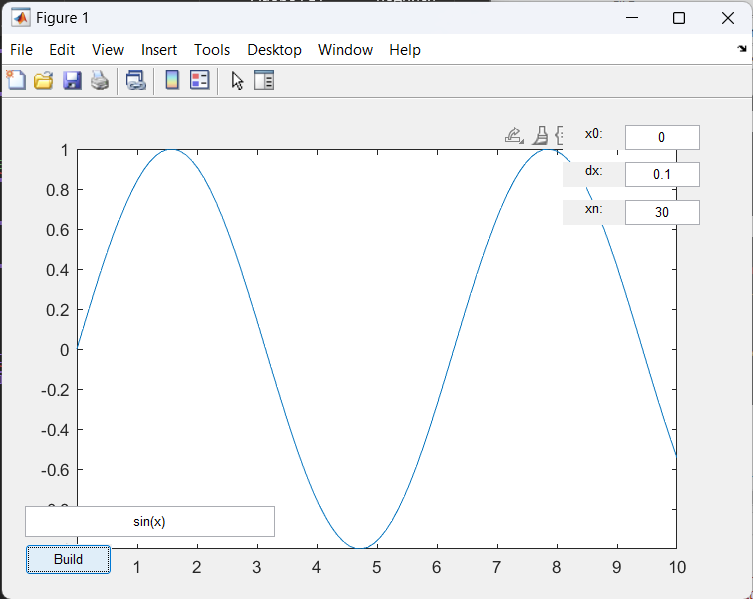
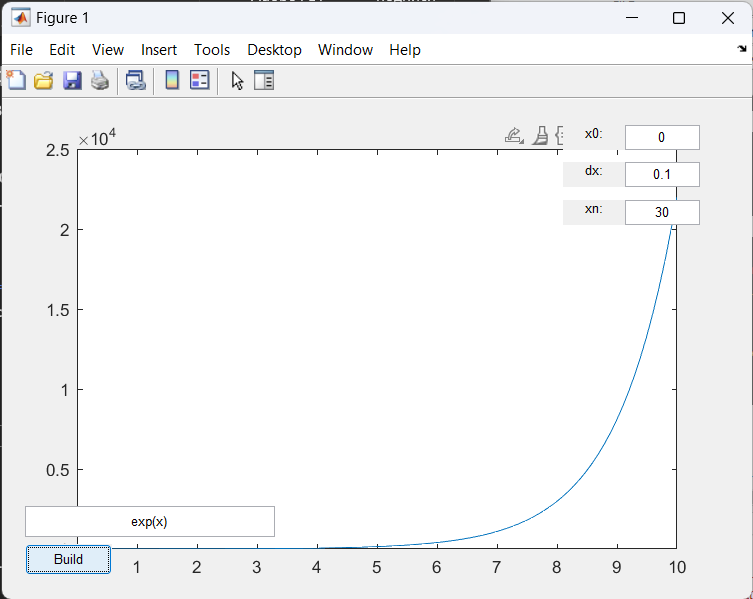
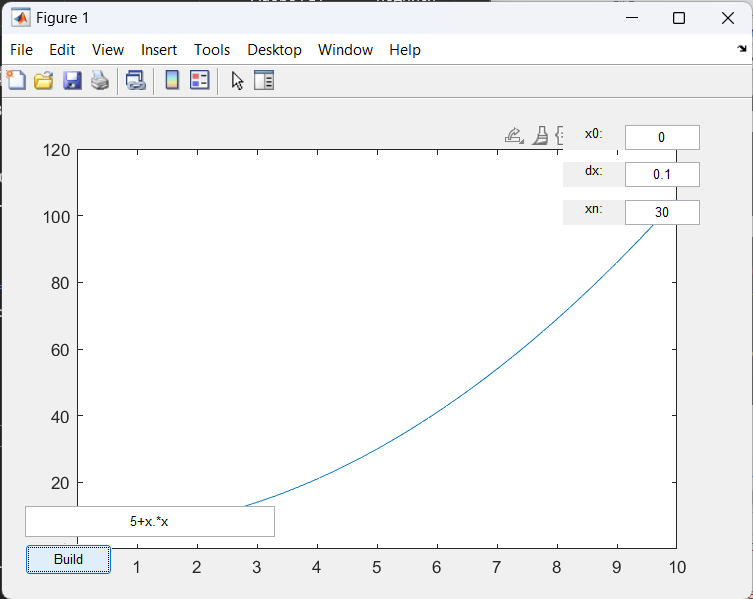
y = eval(func\_str);

plot(x,y);

end

end

Результат программы:

***2.5 Динамическая перестройка элементов управления***

Реализовать проект, в котором на этапе построения графика функции объект axes расширяется и перекрывает элементы управления для ввода начальных значений и запуска вычислений, но при этом появляется новая кнопка, нажатие на которую возвращает все в исходное положение. Реализовать проект для решения жестких дифференциальных уравнений, реализуемых с помощью функции ode15s (п. 2.3).

Код программы

function create\_gui()

% Define the ODE

function dydt = rigid\_ode(t, y)

dydt = [-50\*y(1) + 49\*y(2); 50\*y(1) - 51\*y(2) + 2\*y(3); y(1) + y(2) + y(3)];

end

% Create the figure and controls

fig = figure('Position', [200 200 800 600], 'ToolBar', 'none');

ax = axes('Units', 'normalized', 'Position', [0.2 0.2 0.6 0.6]);

iv\_label = uicontrol('Style', 'text', 'String', 'Initial value:', ...

'Position', [50 50 100 30], 'HorizontalAlignment', 'right');

iv\_edit = uicontrol('Style', 'edit', 'Tag', 'iv', 'Position', [160 50 100 30]);

tend\_label = uicontrol('Style', 'text', 'String', 'End time:', ...

'Position', [270 50 100 30], 'HorizontalAlignment', 'right');

tend\_edit = uicontrol('Style', 'edit', 'Tag', 'tend', 'Position', [380 50 100 30]);

solve\_button = uicontrol('Style', 'pushbutton', 'String', 'Solve', ...

'Position', [490 50 100 30], 'Callback', @solve\_callback);

reset\_button = uicontrol('Style', 'pushbutton', 'String', 'Reset', ...

'Position', [600 50 100 30], 'Callback', @reset\_callback);

% Callback function for the Solve button

function solve\_callback(~, ~)

% Hide the input fields

set([iv\_label iv\_edit tend\_label tend\_edit], 'Visible', 'off');

% Get the initial value and end time from the edit boxes

y0 = str2double(get(iv\_edit, 'String'));

tend = str2double(get(tend\_edit, 'String'));

% Solve the ODE

[t, y] = ode15s(@rigid\_ode, [0 tend], [y0; y0; y0]);

% Plot the solution

plot(ax, t, y(:, 1), t, y(:, 2), t, y(:, 3));

legend(ax, 'y1', 'y2', 'y3');

end

% Callback function for the Reset button

function reset\_callback(~, ~)

% Reset the figure and controls to their original positions

set(ax, 'Position', [0.2 0.2 0.6 0.6]);

set([iv\_label iv\_edit], 'Position', [50 50 100 30], 'Visible', 'on', 'String', '');

set([tend\_label tend\_edit], 'Position', [150 50 100 30], 'Visible', 'on', 'String', '');

set(findobj('String', 'Solve'), 'Position', [490 50 100 30], 'Enable', 'on');

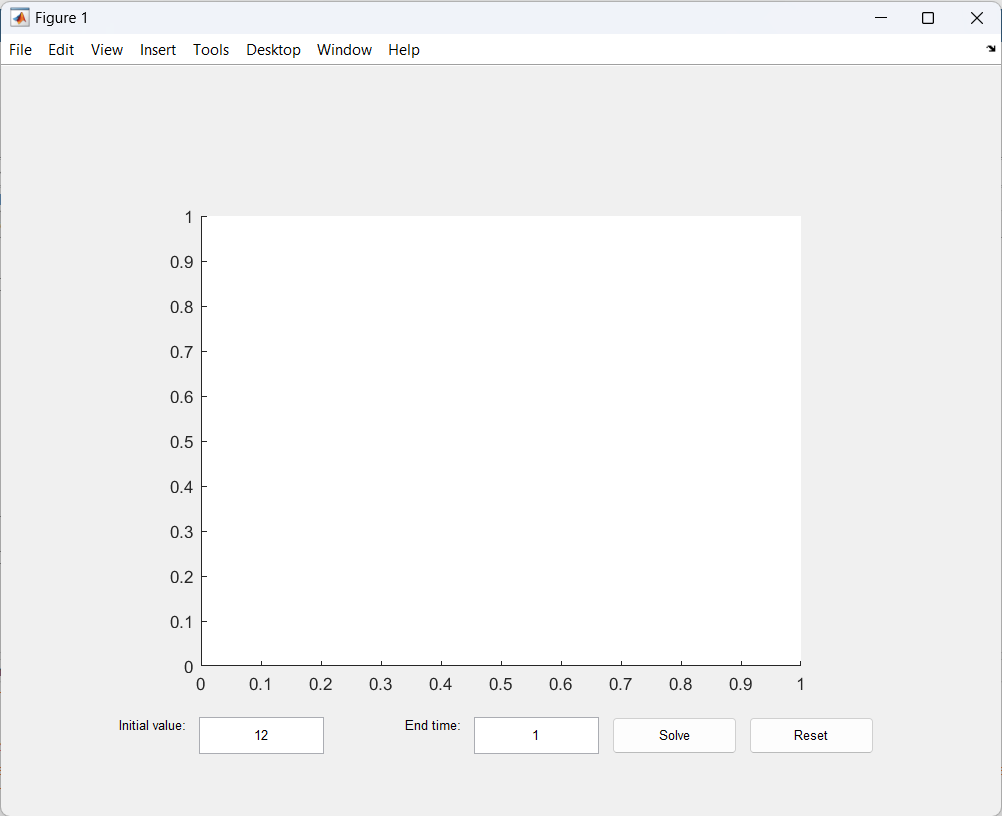
set(findobj('String', 'Reset'), 'Position', [600 50 100 30], 'Enable', 'off');

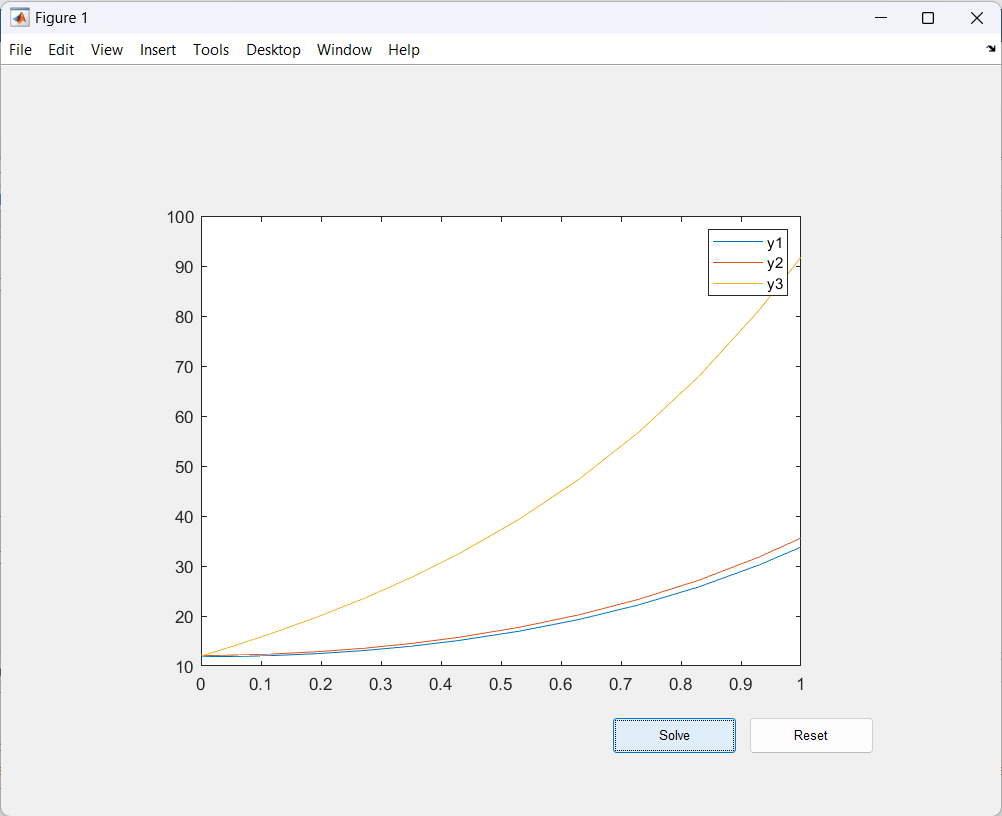
cla(ax); % Clear the plot

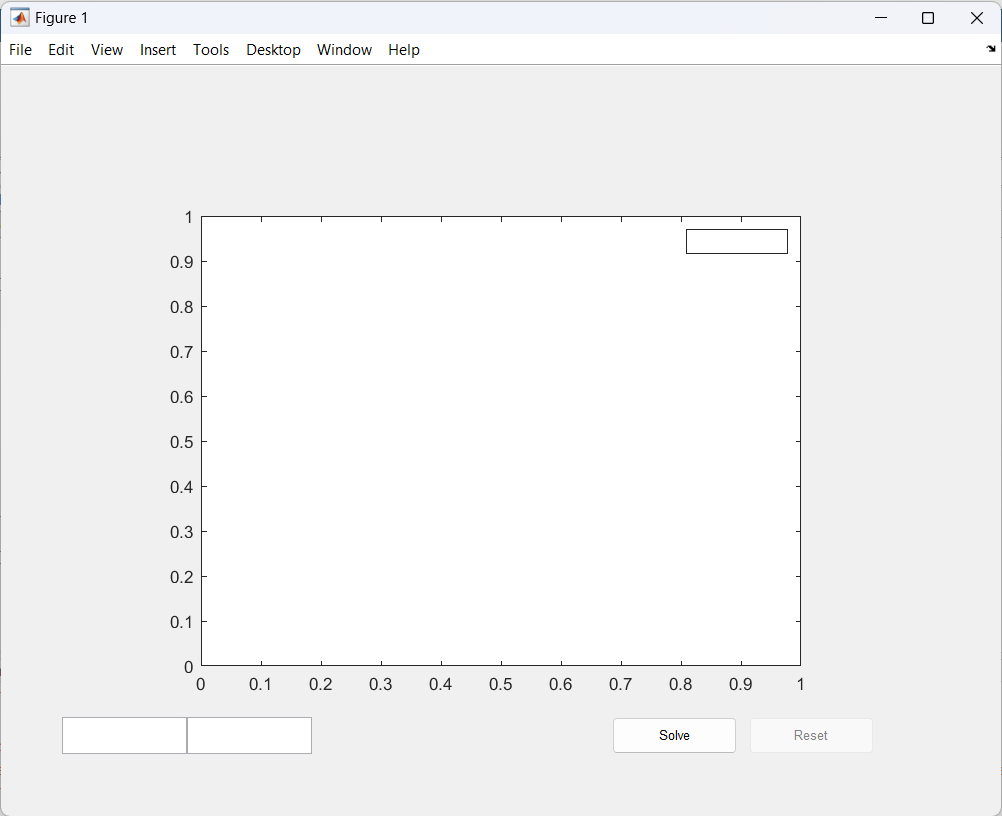
end

end

Вывод программы:







**Вывод:** Изучил принципы создания М-функций и использования их в среде MATLAB, а также возможности создания законченных приложений на базе графического интерфейса пользователя системы MATLAB.