Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине «Логические основы интеллектуальных систем»

Тема: «Интерактивный режим работы системы MATLAB и его автоматизация с помощью сценариев»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Скарубо А. О.

Брест 2023

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

**Ход работы**

**Задание 1**

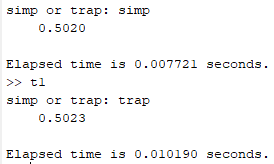
Реализовать М-функцию, вычисляющую интеграл вида двумя методами интегрирования:

- 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.

a = 0;

b = 5;

h = 1;

if findstr(input('simp or trap: ', 's'), 'simp')

tic

disp(simpson(a, h, b))

toc

else

tic

disp(trap(a, h, b))

toc

end

function res = trap(a, h, b)

res = 0;

past\_res = 1;

% Пока разница между текущим значением и предыдущим больше е,

% уменьшать шаг

while abs(abs(res) - abs(past\_res)) > 0.0001

past\_res = res;

res = 0;

for x = a:h:b

res = res + 0.5\*(f(x)+f(x+h))\*h;

end

h = h / 10;

end

end

function res = simpson(a, h, b)

x1 = a+h:h\*2:b-h;

x2 = a+h\*2:h\*2:b-h\*2;

res = h / 3.0 \* (f(a) + f(b) + 4 \* sum(f(x1)) + 2 \* sum(f(x2)));

end

function y = f(x)

y = sin(x).\*exp(-x);

end

**Задание 2**

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

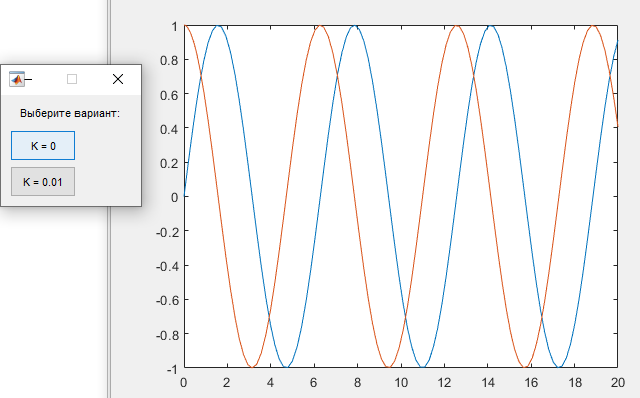


Начальные условия у1(0) = 0; у2(0) = 1. К – коэффициент нелинейности задачи.

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.

m = menu('Выберите вариант:','K = 0','K = 0.01');

if m == 2

disp('K = 0.01')

K = 0.01;

[x, y] = ode45(@f, [0 20], [0;1]);

plot(x, y(:, 1), x, y(:, 2))

else

disp('K = 0')

K = 0;

[x, y] = ode45(@f,[0 20], [0 1]);

plot(x, y(:, 1), x, y(:, 2))

end

function F = f(x, y)

global K;

F = zeros(2, 1);

F(1) = K \* x \* x + y(2);

F(2) = -y(1);

end

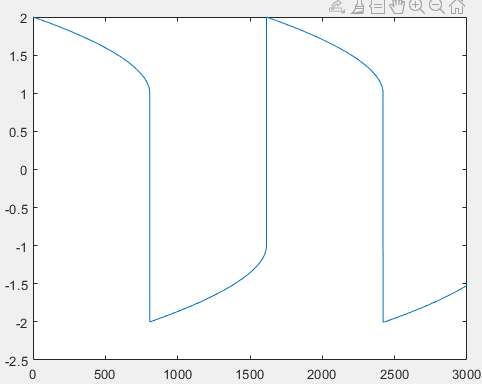
**Задание 3**

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.

****[x,y]=ode45(@f,[0 3000],[2 0]);

plot(x,y(:,1))

function F = f(x, y)

F = zeros(2, 1);

F(1) = y(2);

F(2) = -y(1) + 1000 \* (1-y(1)^2) \* y(2);

end

**Задание 4**

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.

hFig= figure("Position",[550 400 600 500]);

hAxes = axes('Units', 'points', 'Position', [20 40 250 300], 'FontSize',6);

hT1 = uicontrol('Style','text','Position', [400 430 100 20],'String','Левая граница');

hE1 = uicontrol('Style','edit','Position', [400 410 100 20]);

hT2 = uicontrol('Style','text','Position', [400 390 100 20],'String','Приращение');

hE2 = uicontrol('Style','edit','Position', [400 370 100 20]);

hT3 = uicontrol('Style','text','Position', [400 350 100 20],'String','Правая граница');

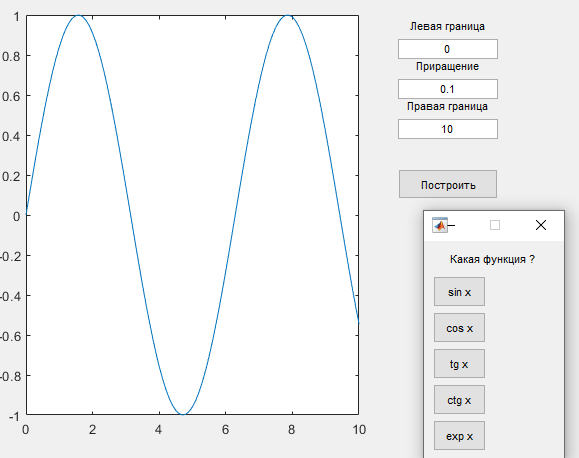
hE3 = uicontrol('Style','edit','Position', [400 330 100 20]);

hB = uicontrol('Style','pushbutton','Position', [400 270 100 30],'String','Построить','Callback',@build);

function build(d,c)

m = menu('Какая функция ?','sin x','cos x','tg x','ctg x','exp x');

global hAxes hE1 hE2 hE3;



cla(hAxes)

x = str2double(get(hE1,'String')) : str2double(get(hE2,'String')) : str2double(get(hE3,'String'));

switch m

case 1

plot(hAxes, x, sin(x))

case 2

plot(hAxes, x, cos(x))

case 3

plot(hAxes, x, tan(x))

case 4

plot(hAxes, x, cotd(x))

case 5

plot(hAxes, x , exp(x))

otherwise

b = 0 : 0.25 : 10;

plot(b, cosh(b))

end

end

**Задание 5**

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).

hFig= figure("Position",[550 400 600 500]);

hAxes = axes('Units', 'points', 'Position', [20 40 250 300], 'FontSize',6);

hT1 = uicontrol('Style','text','Position', [400 430 100 20],'String','Левая граница');

hE1 = uicontrol('Style','edit','Position', [400 410 100 20]);

hT2 = uicontrol('Style','text','Position', [400 390 100 20],'String','Приращение');

hE2 = uicontrol('Style','edit','Position', [400 370 100 20]);

hT3 = uicontrol('Style','text','Position', [400 350 100 20],'String','Правая граница');

hE3 = uicontrol('Style','edit','Position', [400 330 100 20]);

hB = uicontrol('Style','pushbutton','Position', [400 270 100 30],'String','Построить','Callback',@build);

function build(src,event)

m = menu('Какая функция ?','sin x','cos x','tg x','ctg x','exp x');

global hAxes hE1 hE2 hE3 hT1 hT2 hT3 hB hB2;

hB2 = uicontrol("Style","pushbutton","String",'Новая функция','Position',[200 0 100 30],'Callback',@rebuild);

set(hAxes,'Position', [40 40 350 300]);

set(hB2,'Visible','on');

set(hB,'Visible','off');

set(hE1,'Visible','off');

set(hT1,'Visible','off');

set(hE2,'Visible','off');

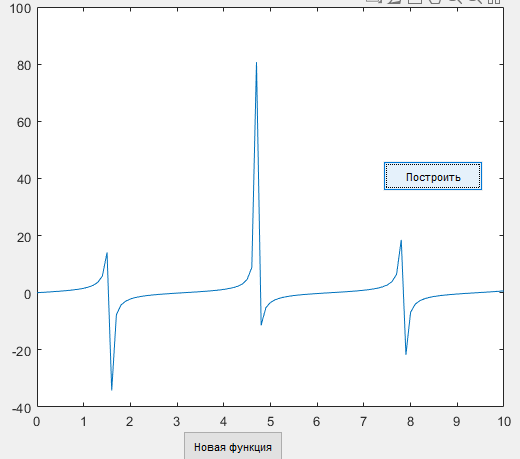
set(hT2,'Visible','off');

set(hE3,'Visible','off');

set(hT3,'Visible','off');

cla(hAxes);

x = str2double(get(hE1,'String')):str2double(get(hE2,'String')):str2double(get(hE3,'String'));

 switch m

case 1

plot(hAxes, x, sin(x));

case 2

plot(hAxes, x, cos(x));

case 3

plot(hAxes, x, tan(x));

case 4

plot(hAxes, x, cotd(x));

case 5

plot(hAxes, x, exp(x));

otherwise

b = 0 : 0.25 : 10;

plot(b, cosh(b))

end

end

function rebuild(src, event)

global hAxes hE1 hE2 hE3 hT1 hT2 hT3 hB hB2;

cla(hAxes);

set(hAxes,'Position', [20 40 250 300]);

set(hB2,'Visible','off');

set(hE1,'Visible','on');

set(hT1,'Visible','on');

set(hE2,'Visible','on');

set(hT2,'Visible','on');

set(hE3,'Visible','on');

set(hT3,'Visible','on');

set(hB,'Visible','on');

end

hFig= figure("Position",[550 400 600 500]);

global hAxes hE1 hE2 hE3 hE4 hE5

hT1=uicontrol('Style','text','Position', [20 430 150 20],'String','Левая граница отрезка х');

hE1=uicontrol('Style','edit','Position', [50 410 100 20]);

hT2=uicontrol('Style','text','Position', [20 390 150 20],'String','Правая граница отрезка х');

hE2=uicontrol('Style','edit','Position', [50 370 100 20]);

hT3=uicontrol('Style','text','Position', [50 350 100 20],'String','Значение y1(0)');

hE3=uicontrol('Style','edit','Position', [50 330 100 20]);

hT4=uicontrol('Style','text','Position', [50 310 100 20],'String','Значение y2(0)');

hE4=uicontrol('Style','edit','Position', [50 290 100 20]);

hT5=uicontrol('Style','text','Position', [50 270 100 20],'String','Значение K');

hE5=uicontrol('Style','edit','Position', [50 250 100 20]);

hB=uicontrol('Style','pushbutton','String','Вычислить','Position',[50 230 100 20],'Callback',@DiffBegin);

hAxes = axes('Units', 'points', 'Position', [170 40 250 300], 'FontSize',6);

function DiffBegin(src,entry)

global hAxes hE1 hE2 hE3 hE4 hE5;

x1 = str2double(get(hE1,'String'));

x2 = str2double(get(hE2,'String'));

y1 = str2double(get(hE3,'String'));

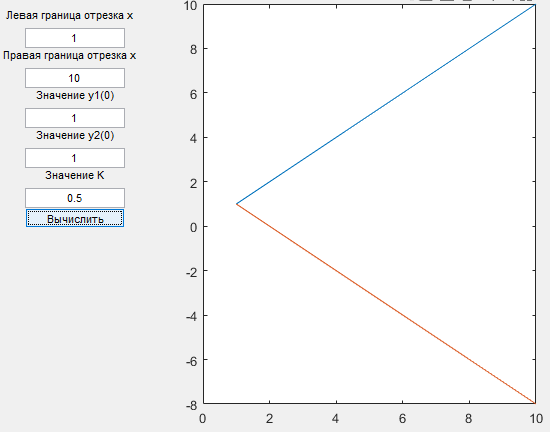
y2 = str2double(get(hE4,'String'));

k = str2double(get(hE5,'String'));

cla(hAxes);

[x,y]=ode15s(@(x,y)[y2;-y1+k\*(1-y1^2)\*y2],[x1 x2],[y1,y2]);

disp(y(:,1))

 disp(x)

plot(hAxes,x,y(:,1))

hold on

plot(hAxes,x,y(:,2))

hold off

end

function dydt=diffunc(x,y)

global k;

dydt=[k\*x^2+y(2);-y(1)];

end

function dydt = diffunc2(x,y)

dydt=[0.01\*x^2+y(2);-y(1)];

end

function dydt = diffunc23(x,y)

dydt=[y(2);-y(1)+1000\*(1-y(1)^2)\*y(2)];

end

**Вывод:** в ходе лабораторной работы я изучил принципы создания М-функций и использования их в среде MATLAB, а также возможностей создания законченных приложений на базе графического интерфейса пользователя системы MATLAB.