
```

clc;
clear;
close all;

h = 0.01;
xspan = 0:h:20;

A0 = 4;
y1_target = 15.366;
dA = 1e-4;

yA0 = solveEuler(A0,h,xspan);

attr = {'Interpreter','latex'};
figure;
plot(xspan,yA0(1,:),xspan,yA0(2,:));
legend("$y_1$","$y_2$","Location","best", attr{:});
title("Metoda Eulera dla $A = 4$", attr{:});
xlabel("$x$", attr{:});
ylabel("$y(x)$", attr{:});

A = A0;
tab = [];

for k = 1:50
    y = solveEuler(A,h,xspan);
    y1A = y(1,end);
    fA = y1A - y1_target;

    y_d = solveEuler(A + dA,h,xspan);
    y1Ad = y_d(1,end);
    fAd = y1Ad - y1_target;

    y2A = y(2,end);
    tab = [tab; k A y1A y2A fA fAd];

    Anew = A - fA * dA / (fAd - fA);

    if abs(Anew - A) < 1e-8
        A = Anew;
        break
    end
    A = Anew;
end

y_final = solveEuler(A,h,xspan);

figure;
plot(xspan,y_final(1,:),xspan,y_final(2,:));
legend("$y_1$","$y_2$","Location","best", attr{:});
title(sprintf("Metoda Eulera dla $A = %.4g$",A), attr{:});

```

```

xlabel("$x$");
ylabel("$y(x)$", attr{::});

disp("Tabela iteracji quasi-Newtona:");
disp(array2table(tab, ...
    'VariableNames',{'iter','A_i','y1_i','y2_i','f(A_i)','f(A_i_plus_dA)' }));

```



```

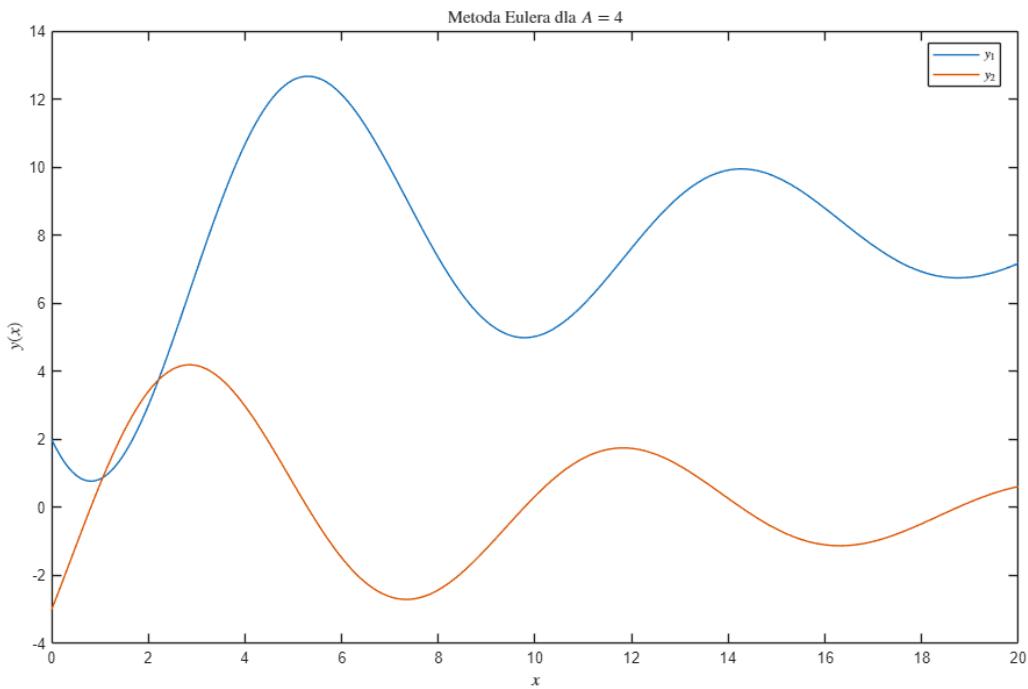
function y = solveEuler(A,h,t)
    f = @(x,y) [y(2); A - 0.5*y(1) - 0.2*y(2)];
    y = zeros(2,length(t));
    y(:,1) = [2; -3];
    for i = 2:length(t)
        y(:,i) = y(:,i-1) + f(t(i-1),y(:,i-1))*h;
    end
end

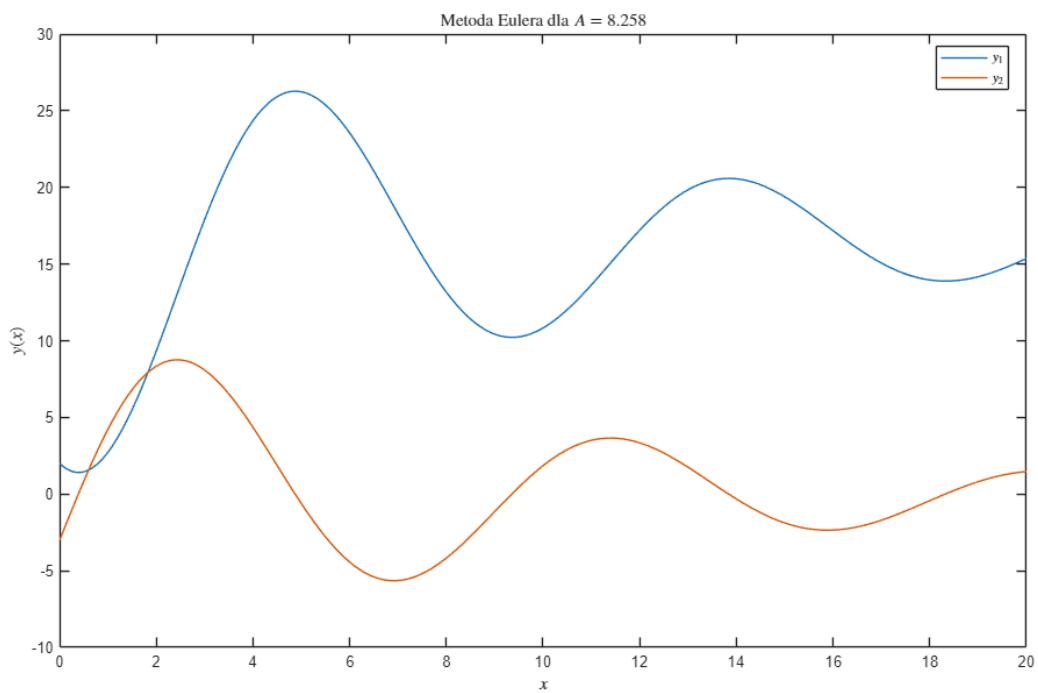
% saveas(gca, "zadanie_14.png");

```

Tabela iteracji quasi-Newtona:

iter	A_i	y1_i	y2_i	f(A_i)	f(A_i_plus_dA)
1	4	7.1704	0.61193	-8.1956	-8.1954
2	8.2579	15.366	1.4691	2.3112e-11	0.00019248





Published with MATLAB® R2025b