

Zespół	Radostaw Smoter Arkadiusz Halat
Numer grupy	LK3
Nazwa ćwiczenia	Rozwiązywanie równań różniczkowych
Numer ćwiczenia	1
Data oddania	25.03.2022
Prowadzący przedmiot	Mgr inż. Denys Gutenko
Ocena	

# Modelowanie Układów Dynamicznych

Spis treści	
Cel ćwiczenia.....	3
Kody źródłowe.....	4
Zadanie 1.....	4
Funkcja wspólna dla wszystkich przykładów.....	4
Przykład 1.....	4
Przykład 2.....	4
Zadanie 2.....	5
Metoda 1. Metoda Eulera.....	5
Metoda 2. Metoda Eulera-Cauchy'ego.....	5
Metoda 3. Metoda Runggego-Kutty RK2.....	6
Metoda 4. Metoda trapezów.....	6
Wyniki.....	7
Opis działania programu.....	10
Metoda Eulera:.....	10
Metoda trapezów:.....	10
Metoda RK2:.....	10
Metoda Eulera-Cauchy'ego:.....	10
Wnioski.....	11
Uwagi.....	12

## Cel ćwiczenia

Rozwiązywanie układów równań różniczkowych na potrzeby modelowania układów dynamicznych, w sytuacjach, gdy „prawdziwe” problemy nie posiadają rozwiązań możliwych do osiągnięcia drogą analityczną.

## Kody źródłowe

### Zadanie 1.

#### Funkcja wspólna dla wszystkich przykładów.

```
function [dy] = funkcja(t,y)
    dy = [y(2); 4*sin(t) + 5*cos(2*t) + y(1)];
end
```

#### Przykład 1.

```
clear; clc; close all;

h = 0.001;           % Step.
time = 10;           % Time.
y = [-1; -2];        % Starting conditions.
t = 0:h:time;        % Full timestamp vector.

% Predeclare yres.
yres = zeros(2, length(t));

for i = 1:length(t)
    yres(:, i) = y + h.*funkcja(t(i), y);
    y = yres(:, i);
end

ydok = -2.*sin(t) - cos(2.*t);

figure(1);
plot( ...
    t, ydok, 'b',           ...
    t, yres(1, :), 'r',     ...
    t, yres(2, :), 'g--');
grid on;
legend('ydok', 'ynum', 'dynam');
```

#### Przykład 2.

```
clear; clc; close all;

h = 0.001;           % Step.
time = 10;           % Time.
y = [-1; -2];        % Starting conditions.
t = 0:h:time;        % Full timestamp vector.

% Predeclare yres.
yres = zeros(2, length(t));

for i = 1:length(t)
    k1 = h.*funkcja(t(i), y);
    k2 = h.*funkcja(t(i) + h, y + k1);
```

```

    yres(:, i) = y + 0.5.*(k1 + k2);
    y = yres(:, i);
end

ydok = -2.*sin(t) - cos(2.*t);
figure(1);
plot(...
    t, ydok, 'b', ...
    t, yres(1, :), 'r--', ...
    t, yres(2, :), 'g--');
grid on;
legend('ydok', 'ynum', 'dynam');

```

## Zadanie 2.

```

function [dy] = funkcja(t, y)
    dy = [-3*y(2)^2 / 3; sin(t)^2 + y(1)];
end

```

## Metoda 1. Metoda Eulera.

```

clear; clc; close all;

h = 0.001;          % Step.
time = 2;           % Time.
y = [4; 5];         % Starting conditions.
t = 0:h:time;       % Full timestamp vector.

% Predeclare yres.
yres = zeros(2, length(t));

for i = 1:length(t)
    yres(:, i) = y + h.*funkcja(t(i), y);
    y = yres(:, i);
end

figure(1)
plot(t, yres(1, :), 'r', t, yres(2, :), 'g--'); grid on;
legend('ynum', 'dynam');

```

## Metoda 2. Metoda Eulera-Cauchy'ego.

```

clear; clc; close all;

h = 0.001;          % Step.
time = 2;           % Time.
y = [4; 5];         % Starting conditions.
t = 0:h:time;       % Full timestamp vector.

% Predeclare yres.
yres = zeros(2, length(t));

for i = 1:length(t)
    k1 = funkcja(t(i), y);
    k2 = funkcja(t(i) + h, y + h*k1);

```

```

    yres(:, i) = y + (0.5*h).*(k1 + k2);
    y = yres(:, i);
end

figure(1)
plot(t,yres(1,:), 'r', t,yres(2,:), 'g--'); grid on;
legend('ynum', 'dynam');

```

### Metoda 3. Metoda Runggego-Kutty RK2.

```

clear; clc; close all;

h = 0.001;          % Step.
time = 2;           % Time.
y = [4; 5];         % Starting conditions.
t = 0:h:time;       % Full timestamp vector.

% Predeclare yres.
yres = zeros(2, length(t));

for i = 1:length(t)
    k1 = h.*funkcja(t(i), y);
    k2 = h.*funkcja(t(i) + h, y + k1);
    yres(:, i) = y + 0.5.*(k1 + k2);
    y = yres(:, i);
end

figure(1)
plot(t,yres(1,:), 'r', t,yres(2,:), 'g--'); grid on;
legend('ynum', 'dynam');

```

### Metoda 4. Metoda trapezów.

```

clear; clc; close all;

h = 0.001;          % Step.
time = 2;           % Time.
y = [4; 5];         % Starting conditions.
t = 0:h:time;       % Full timestamp vector.

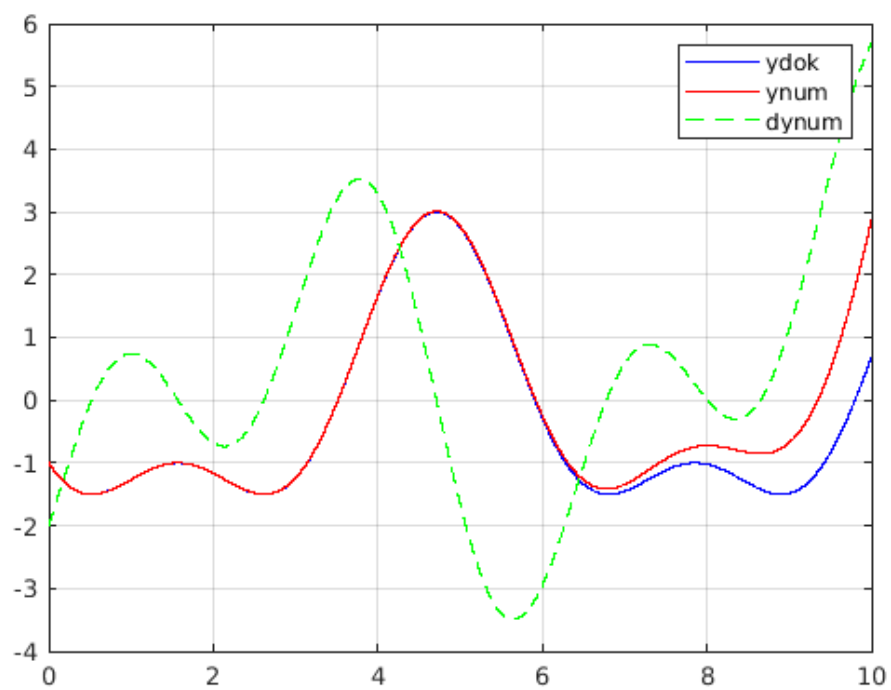
% Predeclare yres.
yres = zeros(2, length(t));

for i = 1:length(t)
    yres(:, i) = y + (h/2).*(funkcja(t(i), y) + funkcja(t(i) + h, y));
    y = yres(:, i);
end

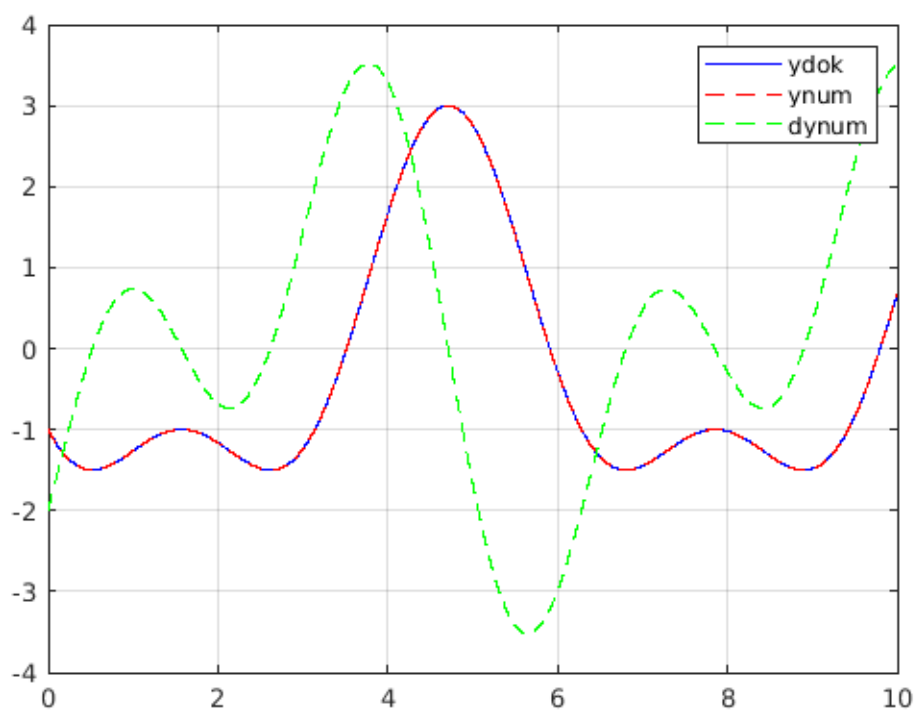
figure(1)
plot(t,yres(1,:), 'r', t,yres(2,:), 'g--'); grid on;
legend('ynum', 'dynam');

```

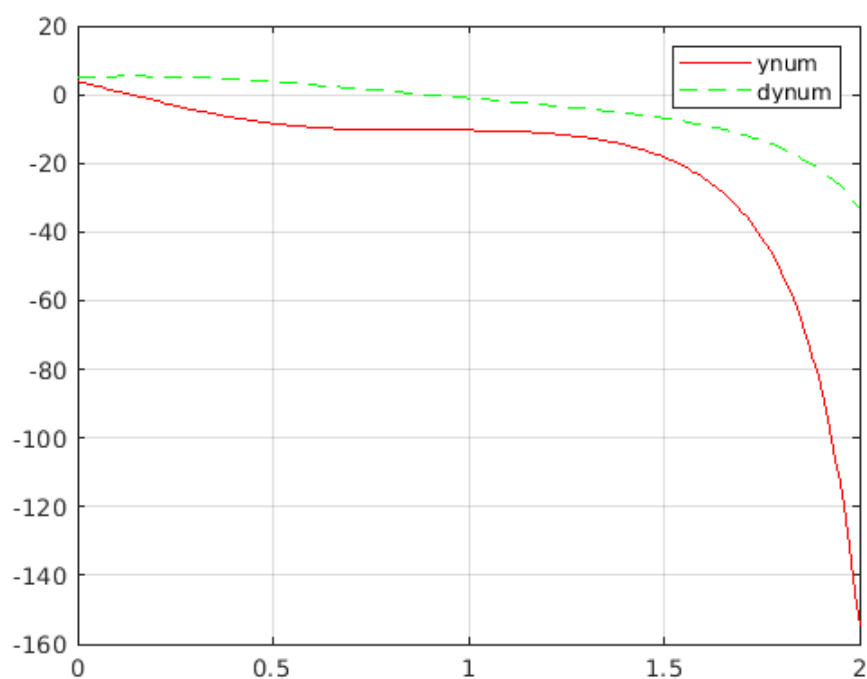
## Wyniki



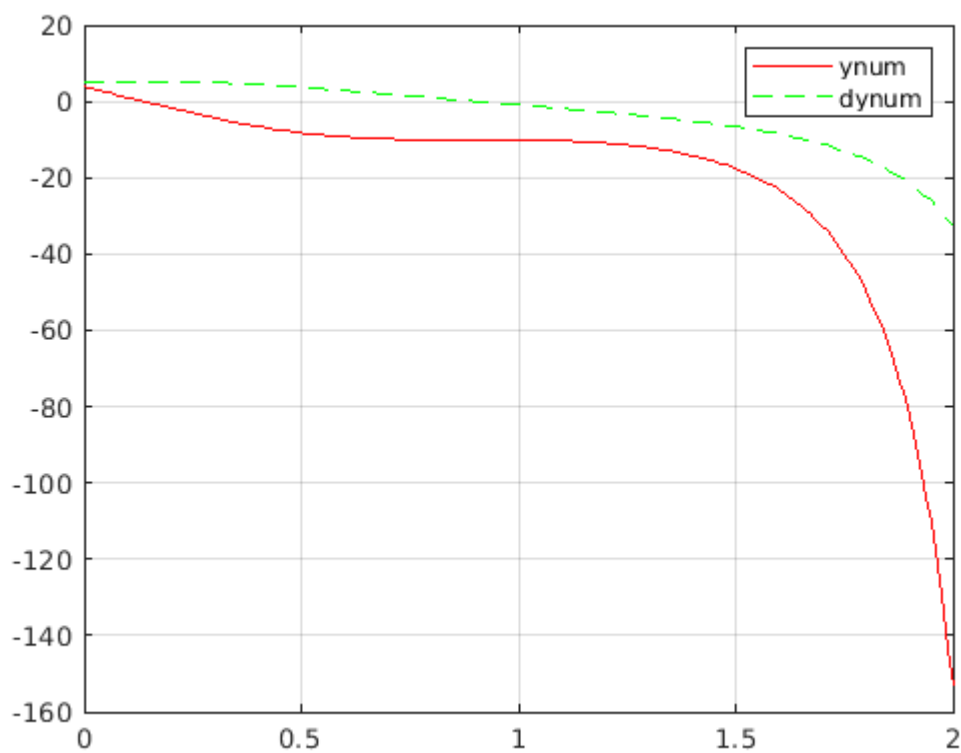
Wykres 1: Zadanie 1. Przykład 1.



Wykres 2: Zadanie 2. Przykład 2.

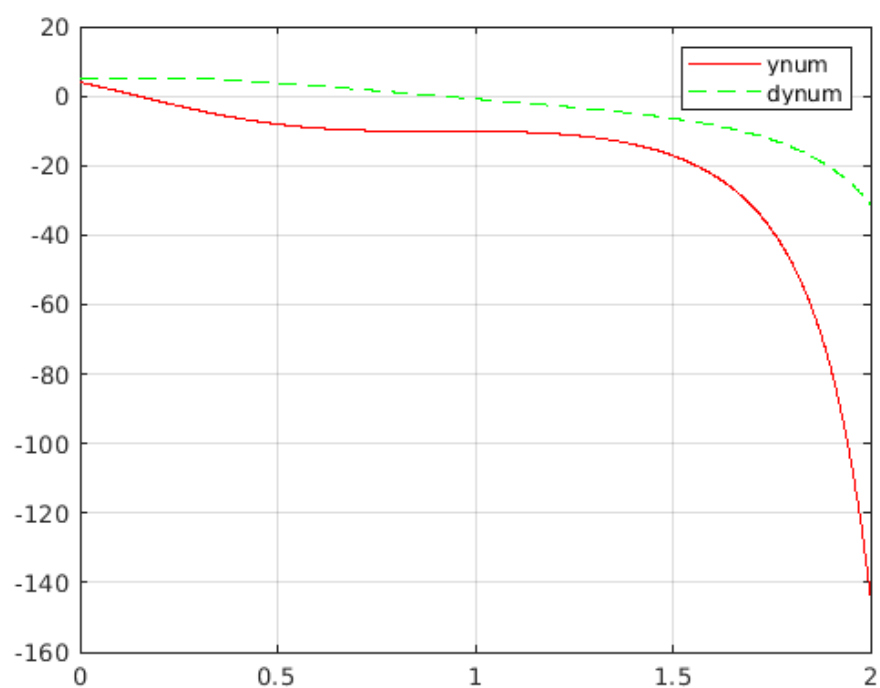


Wykres 3: Zadanie 2. Metoda Eulera.

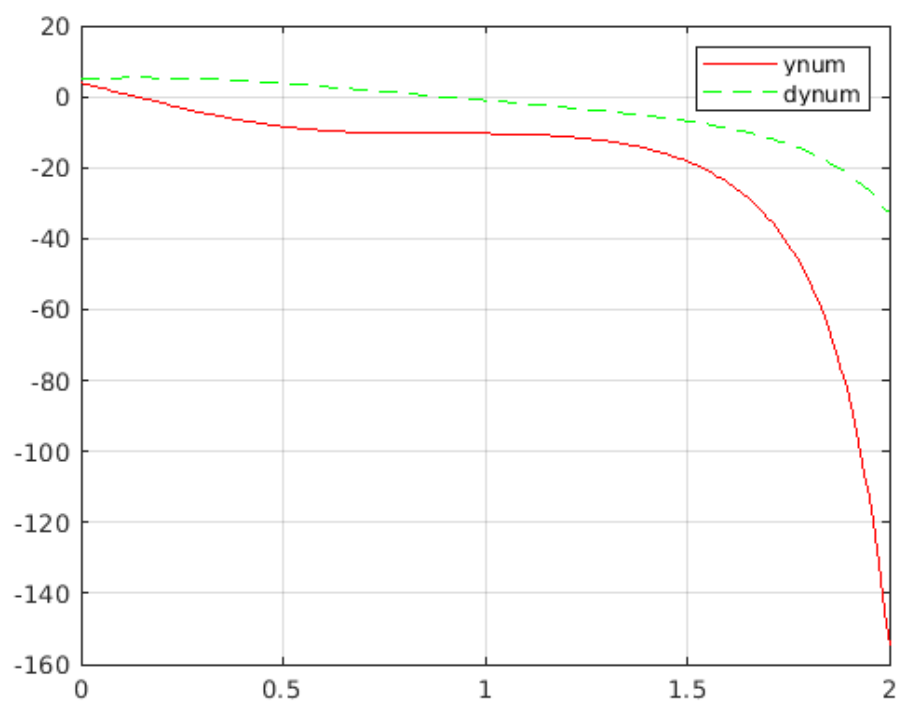


Wykres 4: Zadanie 2. Metoda Eulera-Cauchy'ego.





Wykres 5: Zadanie 2. Metoda RK2.



Wykres 6: Zadanie 2. Metoda trapezów.

## Opis działania programu

Metoda rozwiązywania równań różniczkowych metodami numerycznymi jest podobna dla wszystkich sposobów. W pliku **funkcja.m** znajduje się reprezentacja zadanego równania różniczkowego w postaci pionowego wektora. W każdym kolejnym elemencie tego wektora znajduje się kolejna, niższa, całka. Jako argumenty, ta funkcja przyjmuje czas **t** oraz **y**, co jest wynikiem poprzedniej iteracji tejże funkcji (a przy pierwszej iteracji – warunkami początkowymi), zwracanymi przez **[dy]**.

Następnie każdy program zawiera warunki początkowe, tj. **y**; oraz krok iteracji **h**, czas **time**, wektor czasu **t**.

Dla szybkości wykonywania obliczeń predeklarowany jest wektor **yres**, który stanowi rozwiązanie równania różniczkowego, które zawarte jest w pętli, odbywającej się dla całego czasu **t**. W niej zawarte jest rozwiązanie z użyciem odpowiednich wzorów (Eulera, Eulera-Cauchy'ego, RK2, trapezów).

### Metoda Eulera:

$$x_{k+1} = x_k + h \cdot f(x_k, t_k)$$

### Metoda trapezów:

$$x_{k+1} = x_k + \frac{h}{2} \cdot [f(x_k, t_k) + f(x_k, t_{k+1})]$$

### Metoda RK2:

$$x_{k+1} = x_k + \frac{1}{2}(k_1 + k_2)$$

$$k_1 = h \cdot f(t_k, x_k)$$

$$k_2 = h \cdot f(t_k + h, x_k + k_1)$$

### Metoda Eulera-Cauchy'ego:

$$x_{k+1} = x_k + k_2$$

$$k_1 = h \cdot f(t_k, x_k)$$

$$k_2 = h \cdot f(t_k + \frac{1}{2}h, x_k + \frac{1}{2}k_1)$$

## Wnioski

Istnieje wiele metod rozwiązywania równań różniczkowych. Znajdują one swoje zastosowanie w sytuacjach, gdy dokładne rozwiązywanie równań jest niemożliwe. Wiele z tych metod osiąga podobne wyniki, gdy skala jest mała, jednak różnią się nieznacznie nie tylko między sobą, ale też rozwiązaniem dokładnym. Stąd, znajomość i umiejętność zastosowania odpowiedniej metody, której złożoność obliczeniowa jest adekwatna do zadanego problemu oraz uzyskuje przybliżenie o akceptowalnej niedokładności, jest istotne w pracy inżyniera.

## **Uwagi**

Brak uwag własnych.