

Zespół	Radostaw Smoter Pan Areczek
Nazwa ćwiczenia	Rozwiązywanie równań różniczkowych.
Numer ćwiczenia	1
Data oddania	6.03.2022
Prowadzący przedmiot	mgr inż. Denys Gutenko
Ocena	

Modelowanie układów dynamicznych

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);
end
```

```

    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.

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.
```

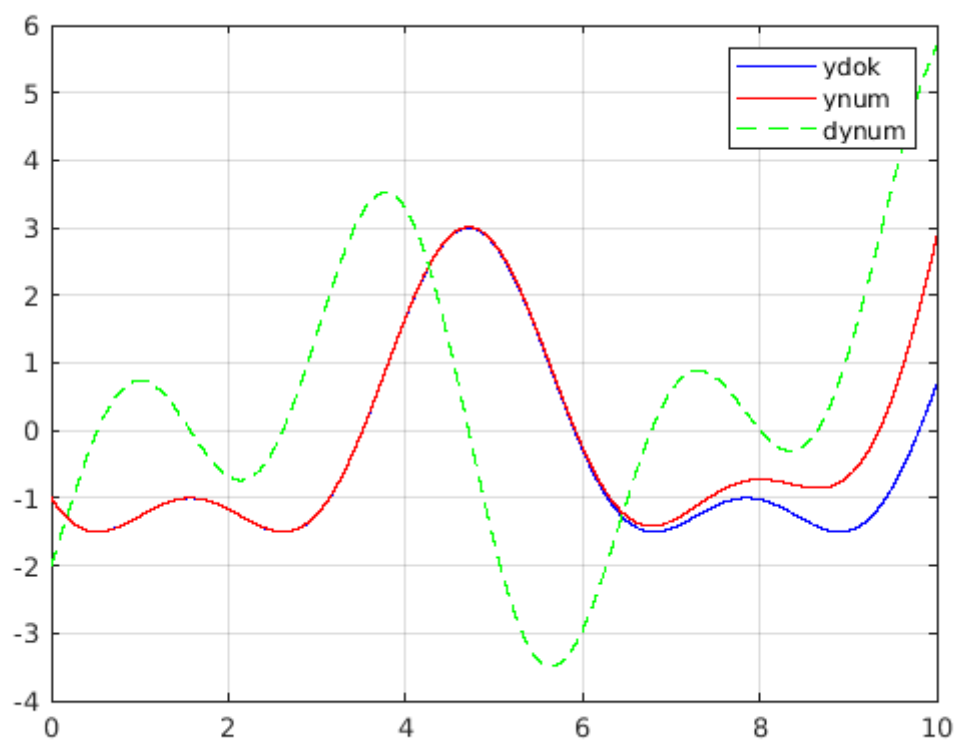
```
% Predeclare yres.
yres = zeros(2, length(t));
```

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

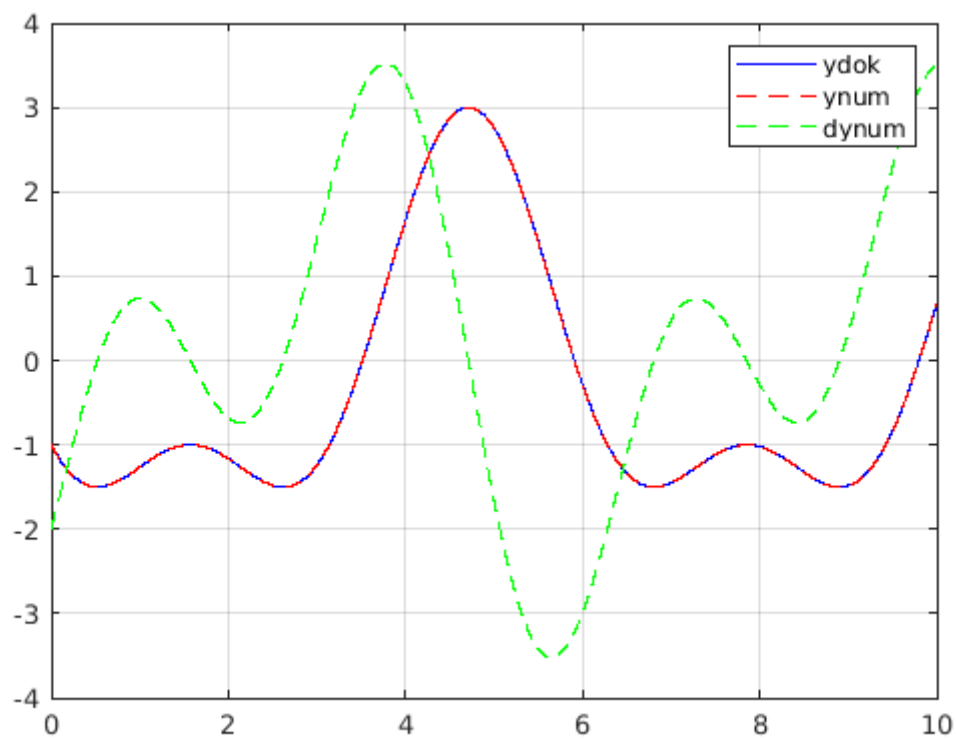
```
figure(1)
plot(t, yres(1,:), 'r', t, yres(2,:), 'g--'); grid on;
legend('ynum', 'dynam');
```

Wyniki

Zadanie 1.

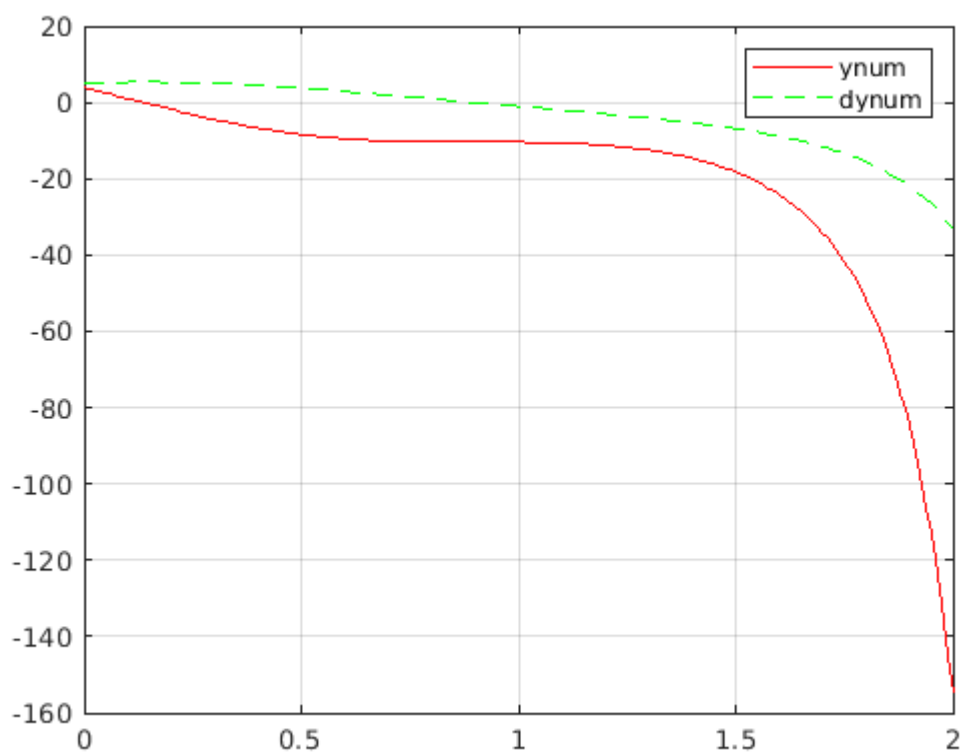


Wykres 1: Zadanie 1. Przykład 1.

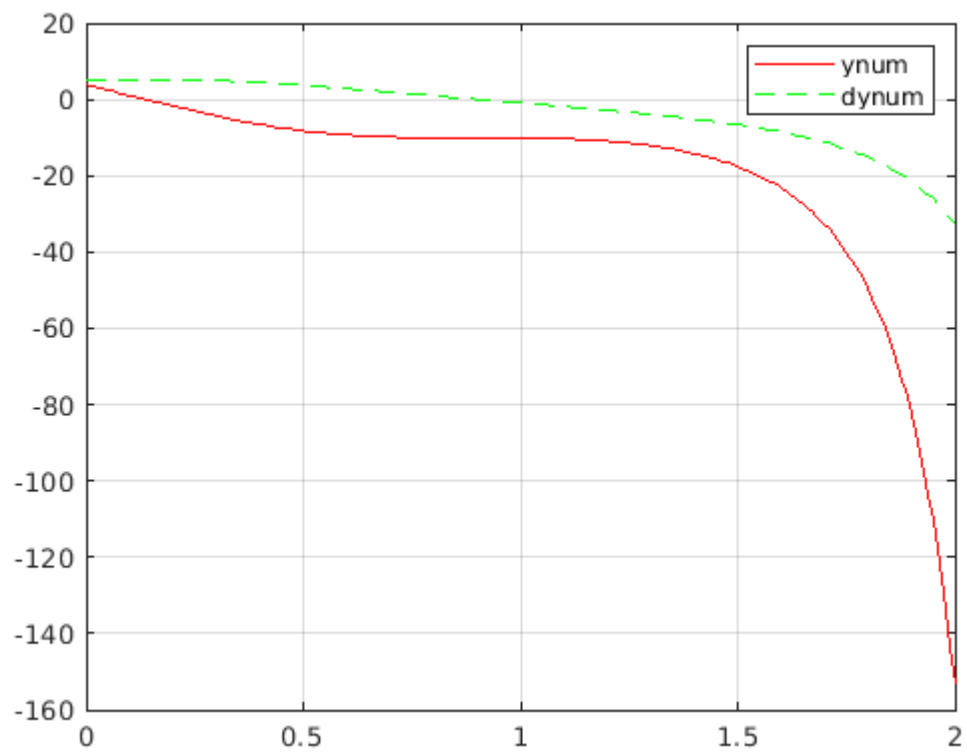


Wykres 2: Zadanie 1. Przykład 2.

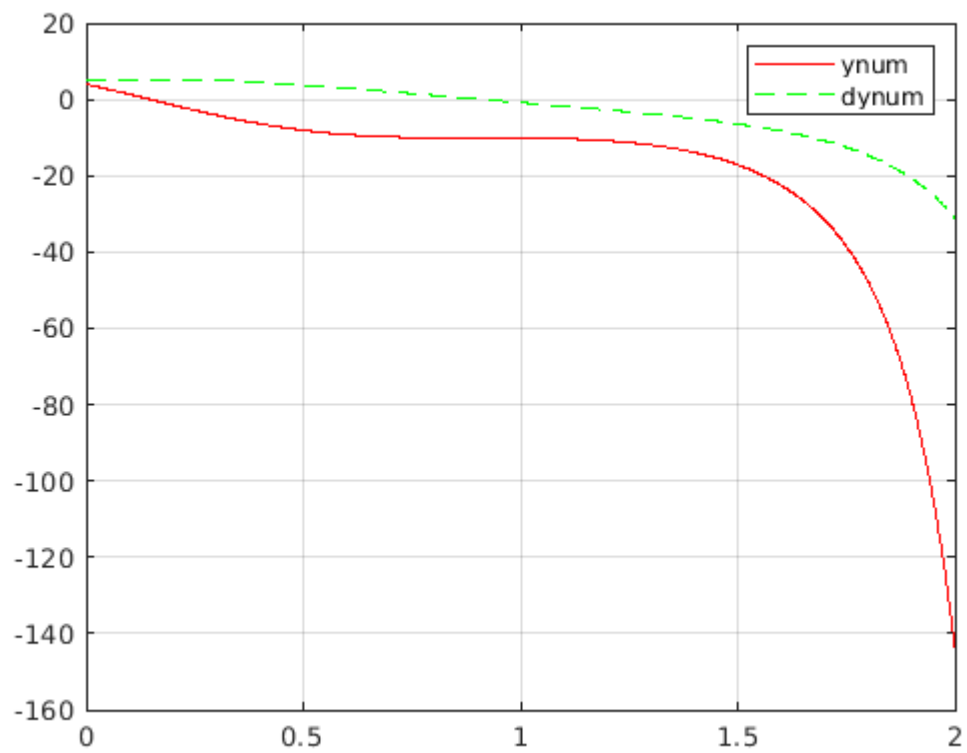
Zadanie 2.



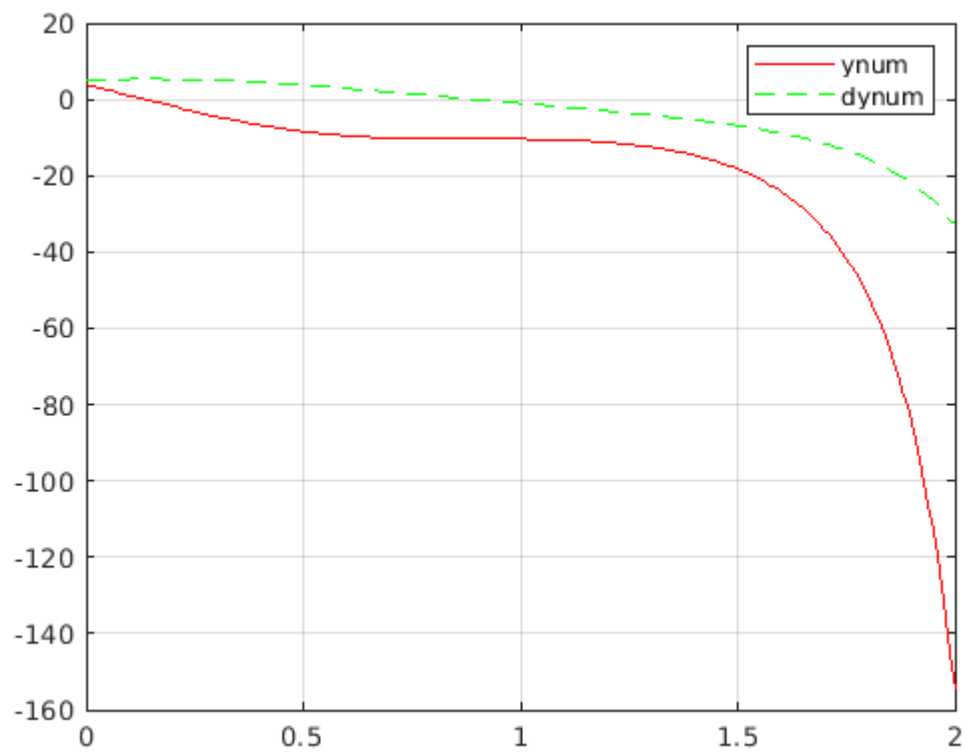
Wykres 3: Zadanie 2. Metoda Eulera.



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



Wykres 5: Zadanie 2. Metoda RK2.



Wykres 6: Zadanie 2. Metoda trapezów.

Opis działania programu.

Wnioski

Uwagi

Brak uwag własnych.