Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

ITERATIVNE METODE

Vježbe 05 - Numeričke metode za rješavanje ODJ

12. siječnja 2022.

Sastavio: Zvonimir Bujanović

Forward Euler

Proučavamo numeričke metode za rješavanje inicijalnog problema ODJ:

$$\begin{aligned} y'(t) &= f(t, y(t)), \quad t \in \langle a, b \rangle, \\ y(a) &= y_0. \end{aligned}$$

Forward Euler metoda.
Podijelimo interval $[a, b]$ na M jednakih podintervala širine h.

$$y_{j+1} = y_j + h \cdot f(t_j, y_j), \quad j = 0, 1, \ldots, M-1.$$

Eksplicitna, uvjetno stabilna, reda konvergencije 1.

Zadatak 1

1. Forward Euler metodom riješite dif. jednadžbe:
   (a) $\frac{dy}{dt} = 10y(1-y), 0 \leq t \leq 1$
   $y(0) = 0.01$

   (b) $\frac{dy}{dt} = -15y, 0 \leq t \leq 1$
   $y(0) = 1$

2. Usporedite s egzaktnim rješenjem, za različite brojeve podintervala M. Također, nacrtajte log-log grafove pogreške

$$\max_j |y_{egz}(t_j) - y_j|$$

o ovisnosti o M.

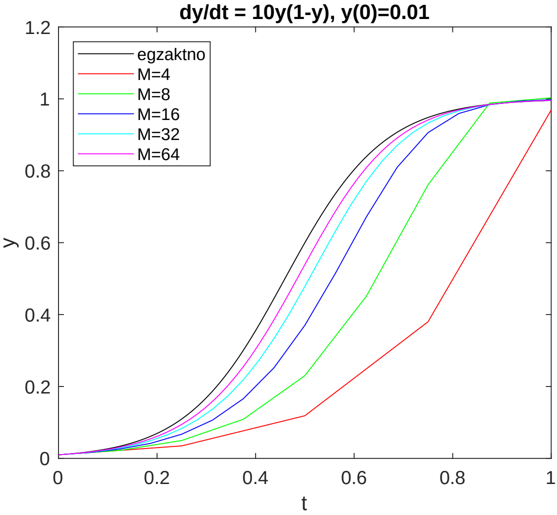
Zadatak 1

Rješenje. U Matlabu možemo funkcijama kao parametre slati druge funkcije.

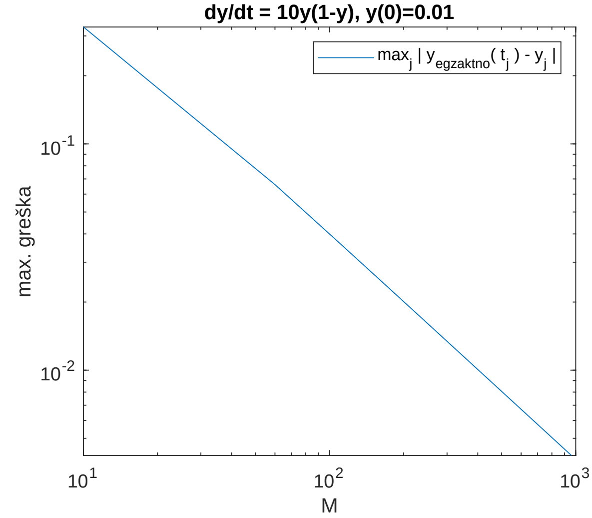
```
[t, y] = forward_euler( 64, 0, 1, 0.01, @logisticka );


function [t, y] = forward_euler( N, a, b, y0, f )
  ...
  % Pozove se prosljeđena funkcija f (ovdje: logisticka)
  y(j+1) = y(j) + h*f( t(j), y(j) );
  ...
end


function ret = logisticka( t, y )
  ret = 10*y*(1-y);
end
```

Zadatak 1

Rješenje. (a)

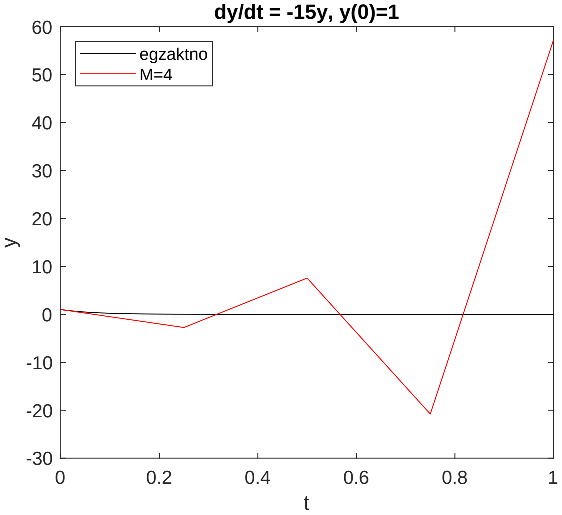


Rješenje. (a)

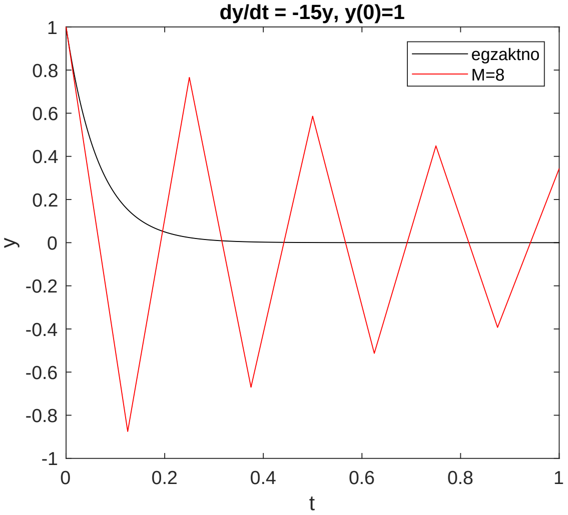

dy/dt = 10y(1-y), y(0)=0.01
max$_j$ | y$_{egzaktno}$(t$_j$) - y$_j$ |
max. greška
10^{-1}
10^{-2}
10^1
10^2
10^3
M

Zadatak 1

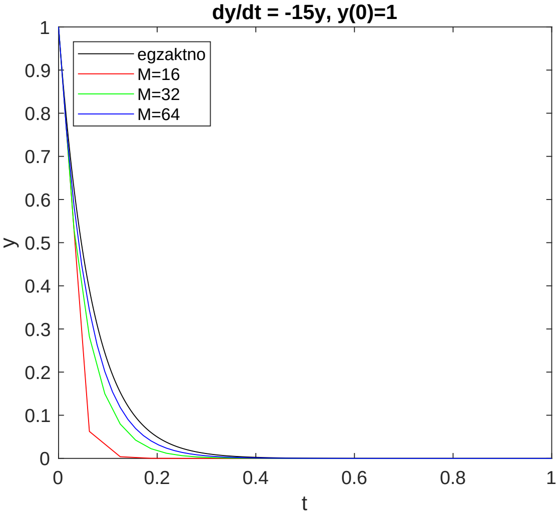
Rješenje. (b)



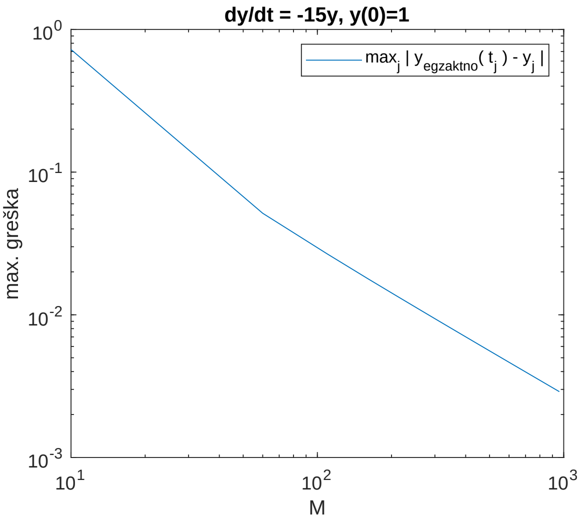
Rješenje. (b)


dy/dt = -15y, y(0)=1
egzaktno
M=8
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
y
0
0.2
0.4
0.6
0.8
1
t

Rješenje. (b)


dy/dt = -15y, y(0)=1
egzaktno
M=16
M=32
M=64
y
t
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1

Rješenje. (b)



dy/dt = -15y, y(0)=1
max. greška
M
10^0
10^{-1}
10^{-2}
10^{-3}
10^1
10^2
10^3
$\max_j | y_{egzaktno}(t_j) - y_j |$

Backward Euler. Trapezna metoda.

Backward Euler metoda.

$$y_{j+1} = y_j + h \cdot f(t_{j+1}, y_{j+1})$$

Implicitna, stabilna, reda konvergencije 1.

Trapezna metoda.

$$y_{j+1} = y_j + \frac{h}{2} \cdot \big(f(t_j, y_j) + f(t_{j+1}, y_{j+1})\big)$$

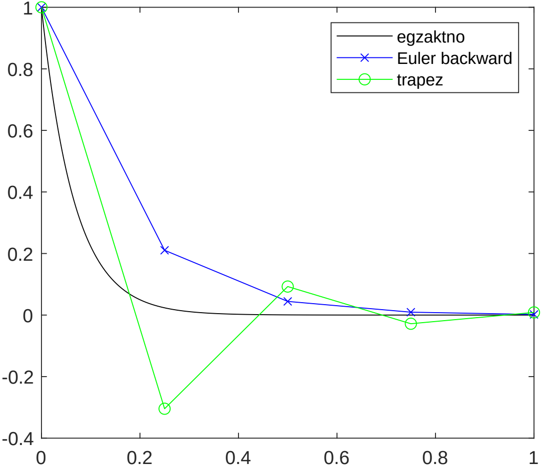
Implicitna, stabilna, reda konvergencije 2.

Zadatak 2

1. Riješite prethodni zadatak i pomoću backward Euler i trapezne metode.
2. Usporedite rješenja s forward Eulerom.

Zadatak 2

Rješenje. (b) Stabilnost (M=4)



Heunova RK2 metoda

Heunova RK2 metoda.

$$\begin{aligned} y_{j+1} &= y_j + \frac{h}{2} \cdot k_1 + \frac{h}{2} \cdot k_2 \\ k_1 &= f(t_j, y_j) \\ k_2 &= f(t_j + h, y_j + h \cdot k_1) \end{aligned}$$

Eksplicitna, uvjetno stabilna, reda konvergencije 2.

Heunova RK2 metoda

Izvod: tražimo a, b, α, β takve da metoda

$$y_{j+1} = y_j + h \cdot (af(t_j, y_j) + bf(t_j + \alpha h, y_j + \beta hf(t_j, y_j))) \qquad (1)$$

bude reda 2.

Neka je y egzaktno rješenje; Taylor:

$$y(t_{j+1}) = y(t_j + h) = y(t_j) + h \cdot y'(t_j) + \frac{h^2}{2} \cdot y''(t_j) + \mathcal{O}(h^3)$$

Uoči: $y'(t_j) = f(t_j, y(t_j))$, pa

$$\begin{aligned} y''(t_j) &= \partial_1 f(t_j, y(t_j)) + \partial_2 f(t_j, y(t_j)) \cdot y'(t_j) \\ &= \partial_1 f(t_j, y(t_j)) + \partial_2 f(t_j, y(t_j)) \cdot f(t_j, y(t_j)) \end{aligned}$$

Oznaka: $y_j^e = y(t_j)$; vratimo nazad u Taylora:

$$\begin{aligned} y_{j+1}^e = y_j^e & \\ &+ h \cdot f(t_j, y_j^e) \\ &+ \frac{h^2}{2} \cdot (\partial_1 f(t_j, y_j^e) + \partial_2 f(t_j, y_j^e) \cdot f(t_j, y_j^e)) \\ &+ \mathcal{O}(h^3) \end{aligned} \tag{2}$$

Heunova RK2 metoda

Sad u Taylorov red razvijemo

$$\begin{aligned} & f(t_j + \alpha h, y_j + \beta h f(t_j, y_j)) \\ & = f(t_j, y_j) + [\alpha h \quad \beta h f(t_j, y_j)] \cdot \nabla f(t_j, y_j) + \mathcal{O}(h^2) \\ & = f(t_j, y_j) + \alpha h \cdot \partial_1 f(t_j, y_j) + \beta h f(t_j, y_j) \cdot \partial_2 f(t_j, y_j) + \mathcal{O}(h^2) \end{aligned}$$

Uvrstimo u (1):

$$\begin{aligned} y_{j+1} = {} & y_j \\ & + h \cdot (a f(t_j, y_j) + b f(t_j, y_j)) \\ & + h^2 (b\alpha \cdot \partial_1 f(t_j, y_j) + b\beta f(t_j, y_j) \cdot \partial_2 f(t_j, y_j)) \\ & + \mathcal{O}(h^3) \end{aligned} \tag{3}$$

Heunova RK2 metoda

Oduzmemo (3) i (2).
Želimo da se pokrate članovi uz h i h^2, pa dobivamo ove uvjete:

$$\begin{aligned} a + b &= 1 \\ b\alpha &= \frac{1}{2} \\ b\beta &= \frac{1}{2} \end{aligned}$$

Heunovu metodu dobijemo tako da stavimo

$$a = b = \frac{1}{2}, \quad \alpha = \beta = 1.$$

“Klasična” RK4 metoda.

$$
\begin{aligned}
y_{j+1} &= y_j + \frac{h}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 &= f(t_j, y_j) \\
k_2 &= f(t_j + \frac{h}{2}, y_j + \frac{h}{2} \cdot k_1) \\
k_3 &= f(t_j + \frac{h}{2}, y_j + \frac{h}{2} \cdot k_2) \\
k_4 &= f(t_j + h, y_j + h \cdot k_3)
\end{aligned}
$$

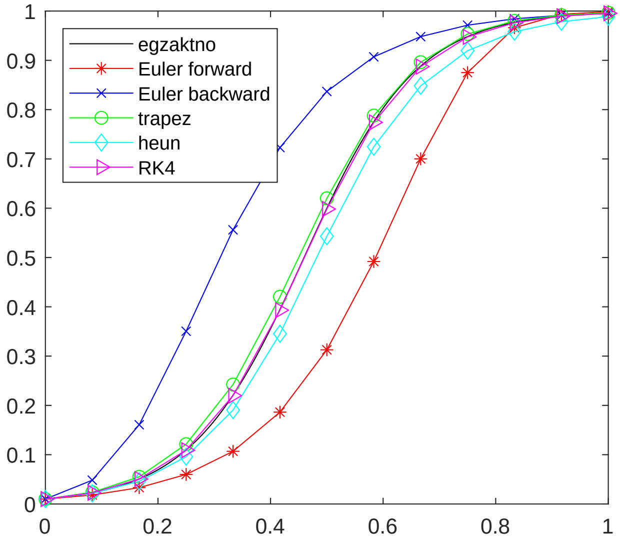
Eksplicitna, uvjetno stabilna, reda konvergencije 4.

Zadatak 3

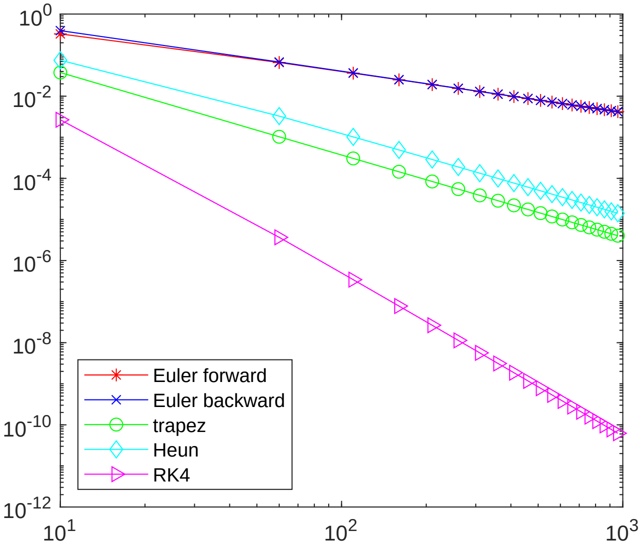
Riješite prethodni zadatak i pomoću Heunove i klasične RK4 metode.
Usporedite rješenja svih metoda.

Zadatak 3

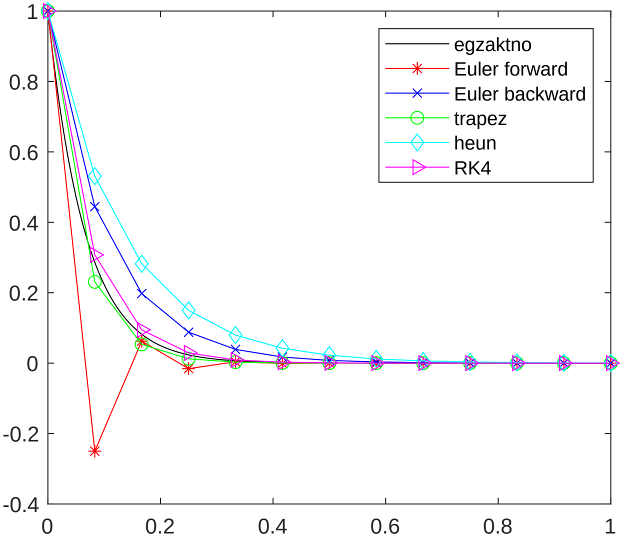
Rješenje. (a) $M = 12$



Rješenje. (a) $\max_j |y_j^{egz} - y_j|$, za $M = 10 \ldots 1000$

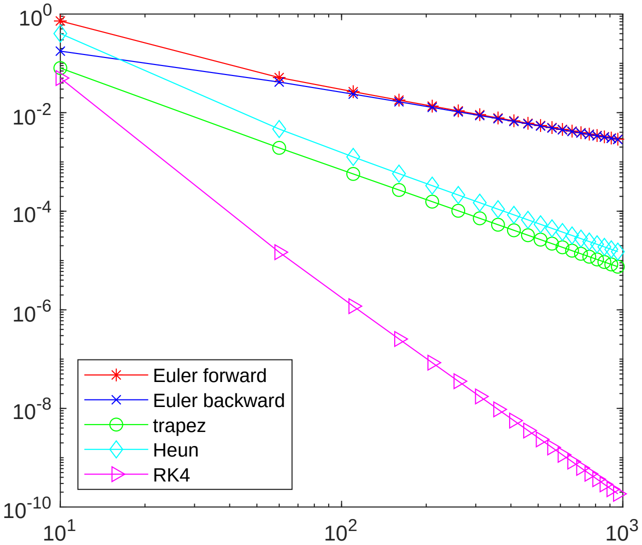


Rješenje. (b) $M = 12$



Zadatak 3

Rješenje. (b) $\max_j |y_j^{egz} - y_j|$, za $M = 10 \ldots 1000$



Zadatak 4

1. U ravnini je smješteno n tijela zadanih masa i početnih koordinata i vektora brzina. Tijela djeluju jedno na drugo gravitacijskom silom. Izvedite diferencijalne jednadžbe kojima je moguće odrediti položaje tijela kroz vrijeme.

2. Riješite numerički dobivene jednadžbe korištenjem svih metoda iz ove prezentacije. Prikažite animaciju za svaku pojedinu metodu i komentirajte rezultat. Možete koristiti sljedeće podatke (Sunce, Zemlja, Mjesec):

$$\begin{aligned}
m_1 &= 1.989 \cdot 10^{30}, & x_1 &= [0;0], & v_1 &= [0,0];\\
m_2 &= 5.9722 \cdot 10^{24}, & x_2 &= [152.1 \cdot 10^6; 0], & v_2 &= [0;29.29]\\
m_3 &= 7.3477 \cdot 10^{22}, & x_3 &= [152.5054 \cdot 10^6; 0], & v_3 &= [0;30.312].
\end{aligned}$$

Gravitacijska konstanta je $G = 6.6743015 \cdot 10^{-20} \frac{km^3}{kg \cdot s^2}$.
Koliko puta se Mjesec okrene oko Zemlje u jednoj godini?

Zadatak 4

Primjer animacije u Matlabu:

```
x = -2*pi:0.2:2*pi; y = sin(x);

% Nacrtaj neku početnu sliku tako da se fiksiraju osi.
plot( x, y, ...
      'Color', [1, 0, 0], 'Marker', '.', 'MarkerSize', 16 );

xlim( [-max(x)*1.1, max(x)*1.1] );
ylim( [-max(y)*1.1, max(y)*1.1] );

ax = gca; ax.NextPlot = 'replaceChildren';

% Video ćemo spremiti u avi datoteku.
v = VideoWriter( 'matlab_ani.avi' );
open(v);

...
```

Zadatak 4

Primjer animacije u Matlabu:

```
loops = 100; % Broj sličica u animaciji.
for j = 1:loops
    % Obriši prethodnu sličicu.
    cla;

    % Nacrtaj i-tu sličicu animacije (ovdje dva grafa)
    plot( x, sin(x + pi*j/loops), ...
          'LineStyle', 'none', ...
          'Color', [1, 0, 0], 'Marker', '.', 'MarkerSize', 16 );
    hold on;
    plot( x, cos(x + pi*j/loops), ...
          'Color', [0, 0, 1], 'Marker', '*', 'MarkerSize', 6 );

    % Spremi u filmić.
    drawnow;
    F = getframe(gcf);
    writeVideo( v, F );
end
close( v ); % Zatvori datoteku s animacijom.
```