UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA AUTOMATIKU I UPRAVLJANJE SISTEMIMA

Matlab - ODE primeri

Modeliranje i simulacija sistema

Upravljanje, modelovanje i simulacija sistema

Leonhard Euler

Born	15 April 1707 Basel, Switzerland
Died	18 September 1783 (aged 76) [OS: 7 September 1783] Saint Petersburg, Russian Empire
Residence	Kingdom of Prussia, Russian Empire Switzerland
Nationality	Swiss
Fields	Mathematics and physics
Institutions	Imperial Russian Academy of Sciences Berlin Academy
Alma mater	University of Basel
Doctoral advisor	Johann Bernoulli
Doctoral students	Nicolas Fuss Johann Hennert Joseph Louis Lagrange Stepan Rumovsky

Ojlerove (*Euler*) metode

```
t = (0:5)';
y0 = 1;
ya = anaResenje(t,y0);

y1 = euler2a(@difJedn,5,y0,1.0);
y05= euler2a(@difJedn,5,y0,0.5);
[t3,y3]=rkutta3(@difJedn,0,5,y0,1.0);

% Test 1
[t ya y1 y1-ya y05 y05-ya y3 y3-ya]

% Test 2
tol=0.001; hmin=0.01;
[te2, ye2]=euler2b(@difJedn,5,y0,hmin,tol,tol);
plot(t,ya,'o',te2,ye2,'.-',t,y01,'.')
```

```
function [t1,y1c,e] = euler2(f,t0,y0,h)
dy0 = f(t0,y0);          % izvod u tacki (t0,y0)
y1p = y0 + dy0*h;
t1 = t0 + h;
dy1 = f(t1,y1p);         % izvod u tacki (t1,y1p)
e = (dy1-dy0)*h/2.0;     % estimacija greske
y1c = y1p + e;           % korekcija
```

```
function Y=euler2a(f,tk,y0,h)
t0 = 0;
Y = y0;
nsteps = 1/h;
for s=1:tk
    for i=1:nsteps
       [t1,y1,e]=euler2(f,t0,y0,h);
       y0=y1;
       t0=t1;
    end
   Y = [Y; y1];
end
```

```
function dy=difJedn(t,y)
% opis DJ
alpha=1.0; lambda=1.0;
dy(1)=lambda*exp(-alpha*t)*y(1);
```

```
function y=anaResenje(t,y0)
% analitičko rešenje
alpha=1.0; lambda=1.0;
y = y0*exp(lambda/alpha*(1-exp(-alpha*t)));
```

Ojlerove (*Euler*) metode – Test 1

t	ya	ye1	ye1-ya	ye05	ye05-ya	yrk3	yrk3-ya
0.0	1.0000	1.0000	0	1.0000	0	1.0000	0
1.0	1.8816	1.8679	-0.0137	1.8786	-0.0030	1.8732	-0.0083
2.0	2.3742	2.3843	0.0101	2.3786	0.0044	2.3642	-0.0100
3.0	2.5863	2.6131	0.0268	2.5952	0.0090	2.5761	-0.0102
4.0	2.6689	2.7033	0.0343	2.6799	0.0110	2.6588	-0.0101
5.0	2.7000	2.7373	0.0373	2.7118	0.0117	2.6899	-0.0101

Legenda:

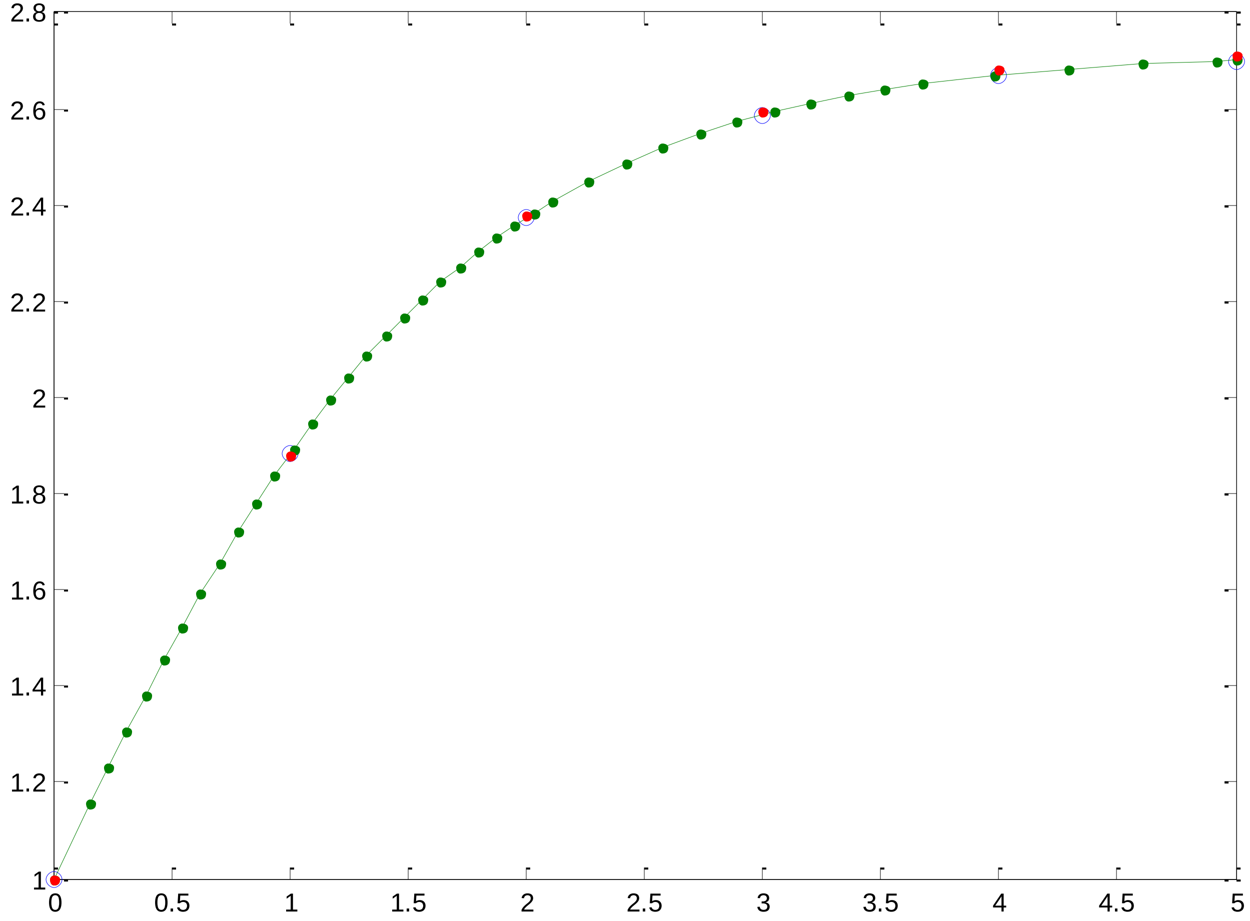
ya – analitičko rešenje

ye1 – Euler 2. reda sa fiksnim korakom h=1

ye05 – Euler 2. reda sa fiksnim korakom h=0.5

ye3 – Euler 3. reda sa fiksnim korakom h=1

Ojlerove (*Euler*) metode – Test 2



<u>Legenda:</u>
Zeleno – Euler 2. reda sa promenljivim korakom
Crveno – Euler 2. reda sa h=0.5
Kružići – tačno rešenje

Carl Runge

Born	30 August 1856 Bremen, German Confederation
Died	3 January 1927 (aged 70) Göttingen, Weimar Republic
Residence	Germany
Citizenship	German
Fields	Mathematics Physics
Institutions	University of Hannover (1886–1904) Georg-August University of Göttingen (1904–1925)
Alma mater	Berlin University
Doctoral advisor	Karl Weierstrass Ernst Kummer
Doctoral students	Max Born
Known for	Runge–Kutta method Runge's phenomenon Laplace–Runge–Lenz vector

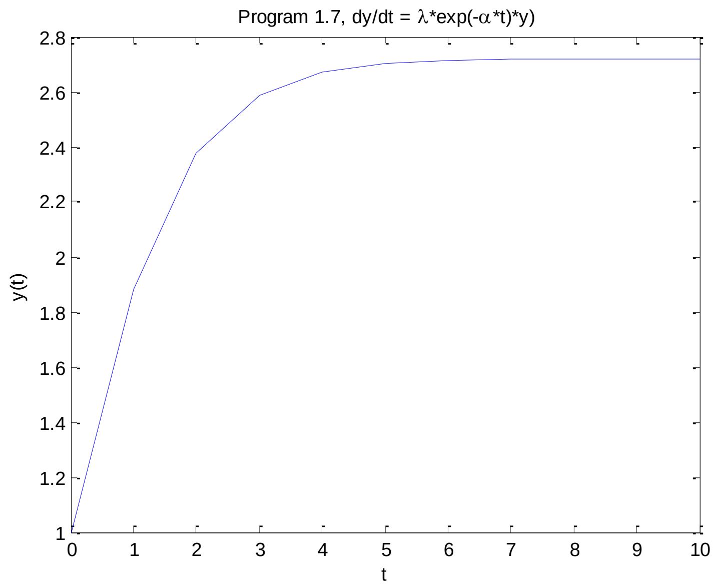
Martin Wilhelm Kutta

Born	November 3, 1867 Pitschen, Upper Silesia
Died	December 25, 1944 (aged 77) Fürstenfeldbruck
Residence	Germany
Nationality	German
Fields	Mathematician
Institutions	University of Stuttgart RWTH Aachen
Alma mater	University of Breslau University of Munich
Doctoral advisor	C. L. Ferdinand Lindemann Gustav A. Bauer
Other academic advisors	Walther Franz Anton von Dyck
Known for	Runge-Kutta method Zhukovsky-Kutta aerofoil Kutta–Joukowski theorem Kutta condition

ode23/ode45 – uticaj tolerancije računanja

```
global lambda alpha ncall;     % Globalne promenljive

lambda=1.0; alpha=1.0;         % Model parameteri

for ncase=1:4
   reltol=1.0e-02^(ncase+1);   % tolerancije racunanje
   abstol=reltol;

   ncall=0;                    % reset brojaca

   t0=0.0; tf=10.0;  tout=[t0:1.0:tf]';

   y0=1.0;                     % Inicijana vrednost

   options=odeset('RelTol',reltol,'AbsTol',abstol);
   [t,y]=ode23('ode1p7',tout,y0,options);

   % Prikaz rezultata
   fprintf('reltol = abstol = %6.2e\n\n',reltol);
   fprintf('   t   ye   y   erry\n');

   ye=y0*exp((lambda/alpha)*(1-exp(-alpha*t)));
   erry=ye-y;
   fprintf('%5.1f%9.4f%9.4f%15.10f\n',[t,ye,y,erry]');

   fprintf('\n ncall = %5d\n',ncall);
end
```

```
function yt=ode1p7(t,y)

% globalne promenljive
global lambda alpha ncall;

% ODE
yt(1)=lambda*exp(-alpha*t)*y(1);

% inkrementiraj brojac
ncall=ncall+1;
```



ode23

```
reltol = abstol = 1.00e-004

   t        ye        y             erry
 0.0    1.0000    1.0000    0.0000000000
 1.0    1.8816    1.8816   -0.0000020034
 2.0    2.3742    2.3743   -0.0000638807
 3.0    2.5863    2.5864   -0.0000933201
 4.0    2.6689    2.6691   -0.0001103656
 5.0    2.7000    2.7002   -0.0001284242
 6.0    2.7116    2.7117   -0.0001434300
 7.0    2.7158    2.7160   -0.0001552112
 8.0    2.7174    2.7175   -0.0001581733
 9.0    2.7179    2.7181   -0.0001592608
10.0    2.7182    2.7183   -0.0001592537
ncall =     73

reltol = abstol = 1.00e-006

   t        ye        y             erry
 0.0    1.0000    1.0000    0.0000000000
 1.0    1.8816    1.8816   -0.0000003221
 2.0    2.3742    2.3742   -0.0000012721
 3.0    2.5863    2.5863   -0.0000016231
 4.0    2.6689    2.6689   -0.0000018847
 5.0    2.7000    2.7000   -0.0000021190
 6.0    2.7116    2.7116   -0.0000023452
 7.0    2.7158    2.7158   -0.0000025176
 8.0    2.7174    2.7174   -0.0000026866
 9.0    2.7179    2.7179   -0.0000028883
10.0    2.7182    2.7182   -0.0000029806
ncall =    268
```

```
reltol = abstol = 1.00e-008

   t        ye        y             erry
 0.0    1.0000    1.0000    0.0000000000
 1.0    1.8816    1.8816   -0.0000000072
 2.0    2.3742    2.3742   -0.0000000188
 3.0    2.5863    2.5863   -0.0000000229
 4.0    2.6689    2.6689   -0.0000000260
 5.0    2.7000    2.7000   -0.0000000285
 6.0    2.7116    2.7116   -0.0000000309
 7.0    2.7158    2.7158   -0.0000000331
 8.0    2.7174    2.7174   -0.0000000354
 9.0    2.7179    2.7179   -0.0000000374
10.0    2.7182    2.7182   -0.0000000394
ncall =   1180

reltol = abstol = 1.00e-010

   t        ye        y             erry
 0.0    1.0000    1.0000    0.0000000000
 1.0    1.8816    1.8816   -0.0000000001
 2.0    2.3742    2.3742   -0.0000000003
 3.0    2.5863    2.5863   -0.0000000003
 4.0    2.6689    2.6689   -0.0000000003
 5.0    2.7000    2.7000   -0.0000000004
 6.0    2.7116    2.7116   -0.0000000004
 7.0    2.7158    2.7158   -0.0000000004
 8.0    2.7174    2.7174   -0.0000000004
 9.0    2.7179    2.7179   -0.0000000005
10.0    2.7182    2.7182   -0.0000000005
ncall =   5392
```

ode45

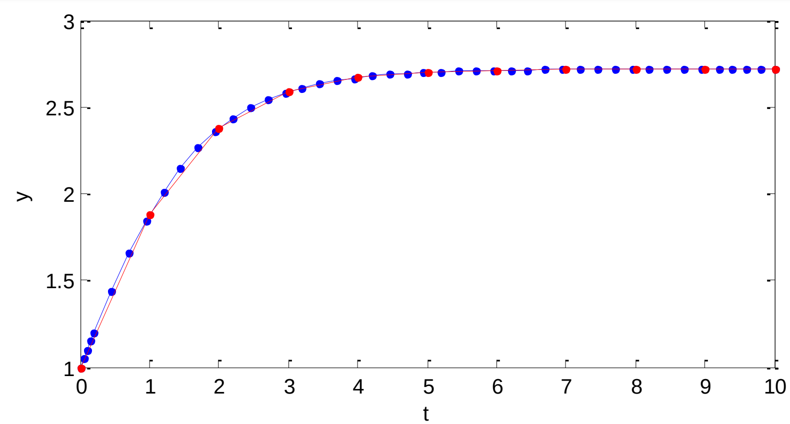
```
reltol = abstol = 1.00e-004

   t        ye        y             erry
  0.0    1.0000    1.0000    0.0000000000
  1.0    1.8816    1.8815    0.0000785756
  2.0    2.3742    2.3742    0.0000230677
  3.0    2.5863    2.5862    0.0000132883
  4.0    2.6689    2.6689    0.0000149230
  5.0    2.7000    2.7000    0.0000165095
  6.0    2.7116    2.7115    0.0000172423
  7.0    2.7158    2.7158    0.0000175329
  8.0    2.7174    2.7174    0.0000176427
  9.0    2.7179    2.7179    0.0000176835
 10.0    2.7182    2.7181    0.0000177068
ncall =     73

reltol = abstol = 1.00e-006

   t        ye        y             erry
  0.0    1.0000    1.0000    0.0000000000
  1.0    1.8816    1.8816    0.0000010053
  2.0    2.3742    2.3742    0.0000010986
  3.0    2.5863    2.5863    0.0000011560
  4.0    2.6689    2.6689    0.0000010643
  5.0    2.7000    2.7000    0.0000010083
  6.0    2.7116    2.7116    0.0000009776
  7.0    2.7158    2.7158    0.0000009652
  8.0    2.7174    2.7174    0.0000009604
  9.0    2.7179    2.7179    0.0000009586
 10.0    2.7182    2.7182    0.0000009579
ncall =     85
```

```
reltol = abstol = 1.00e-008

   t        ye        y             erry
  0.0    1.0000    1.0000    0.0000000000
  1.0    1.8816    1.8816    0.0000000237
  2.0    2.3742    2.3742    0.0000000012
  3.0    2.5863    2.5863    0.0000000128
  4.0    2.6689    2.6689    0.0000000112
  5.0    2.7000    2.7000    0.0000000428
  6.0    2.7116    2.7116    0.0000000092
  7.0    2.7158    2.7158    0.0000000014
  8.0    2.7174    2.7174   -0.0000000231
  9.0    2.7179    2.7179   -0.0000000081
 10.0    2.7182    2.7182    0.0000000077
ncall =    163

reltol = abstol = 1.00e-010

   t        ye        y             erry
  0.0    1.0000    1.0000    0.0000000000
  1.0    1.8816    1.8816    0.0000000001
  2.0    2.3742    2.3742   -0.0000000001
  3.0    2.5863    2.5863   -0.0000000001
  4.0    2.6689    2.6689    0.0000000003
  5.0    2.7000    2.7000    0.0000000004
  6.0    2.7116    2.7116   -0.0000000004
  7.0    2.7158    2.7158    0.0000000003
  8.0    2.7174    2.7174    0.0000000002
  9.0    2.7179    2.7179   -0.0000000003
 10.0    2.7182    2.7182    0.0000000001
ncall =    385
```

ode45 – Diferencijalna jednačina 1. reda

```
>> global lambda alpha ncall;
>> lambda=1; alpha=1; ncall=0;
>> tout = [0 10];
>> y0 = 1;
>> [t,y]=ode45('ode1p7',tout,y0);
>> ncall              % broj poziva f-je
ncall =
    67

>> [t y]
ans =
         0    1.0000
    0.0502    1.0502
    0.1005    1.1003
    0.1507    1.1502
    0.2010    1.1997
    0.4510    1.4375
    0.7010    1.6551
    0.9510    1.8472
    1.2010    2.0119
    1.4510    2.1503
    1.7010    2.2648
    1.9510    2.3581
    2.2010    2.4334
    2.4510    2.4937
           ...
    9.8002    2.7181
   10.0000    2.7181

>> size(t,1)          % broj vrsta u t
ans =
    45
```

```
>> ncall=0;
>> tout = [0:1:10];
>> [t,y]=ode45('ode1p7',tout,y0);

>> ncall
ncall =
    67

>> [t y]
ans =
         0    1.0000
    1.0000    1.8816
    2.0000    2.3742
    3.0000    2.5862
    4.0000    2.6689
    5.0000    2.7000
    6.0000    2.7115
    7.0000    2.7158
    8.0000    2.7173
    9.0000    2.7179
   10.0000    2.7181
```

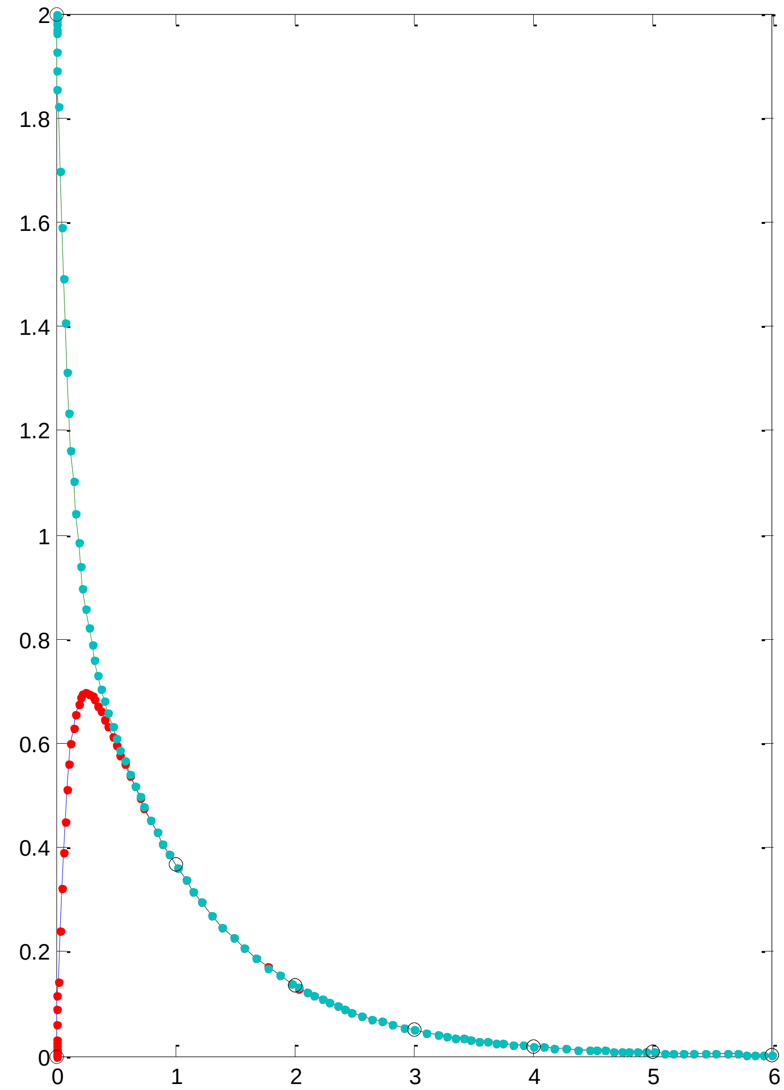


ode45 – Dve diferencijalne jednačine 1. reda = model 2. reda

```
function yt=ode1p8(t,y)
% parametri modela
a=5.5;  b=4.5;
% jednacine
yt(1)=-a*y(1)+b*y(2);
yt(2)= b*y(1)-a*y(2);
yt=yt';
```

```
t0=0.0;  tf=6.0;  tout=[t0 tf];
a=5.5;  b=4.5;
y10=0.0;  y20=2.0;  y0=[y10 y20]';

[t,y]=ode45('ode1p8',tout,y0);

lambda1=-(a-b);  lambda2=-(a+b);
exp1=exp(lambda1*t);  exp2=exp(lambda2*t);
y1e=(y10+y20)/2.0*exp1-(y20-y10)/2.0*exp2;
y2e=(y10+y20)/2.0*exp1+(y20-y10)/2.0*exp2;

[t1,y1]=ode45('ode1p8',t0:tf,y0);

plot(t,[y1e y2e],t,y,'.',t1,y1,'ok');
xlabel('t'), ylabel('y1(t),y2(t)')
```

```
>> size(t)
ans =
   117     1

>> size(y)
ans =
   117     2
```



Lotka–Volterra jednačina

- Model dinamičkog biološkog sistema gde dve vrste imaju odnos: grabljivica-plen

$$\frac{dx}{dt} = x(\alpha - \beta y)$$
$$\frac{dy}{dt} = -y(\gamma - \delta x)$$

Deterministički
Kontinualan
Nelinearan
model 2. reda

x – broj jedinki plena (npr. zečevi)
y – broj jedinki grabljivica (npr. lisice)
izvodi predstavljaju porast broja populacije, *t* vreme, a parametri $\alpha,\beta,\delta,\gamma$ određuju njihovu interakciju.

- Pretpostavke:
  - Plen uvek ima dovoljno hrane
  - Zalihe hrane za grabljivice zavise od populacije plena
  - Stopa promene populacije je proporcionalna njenoj veličini
  - Okruženje se ne menja u korist jedne vrste i nema genetske adaptacije

Promena populacije Plen-Grabljivica

- Promena populacije plena
  - Raste srazmerno broju jedinki zbor reprodukcije
  - Oprada srazmerno učestanosti susreta sa grabljivicama

$$\frac{dx}{dt} = \alpha x - \beta xy$$

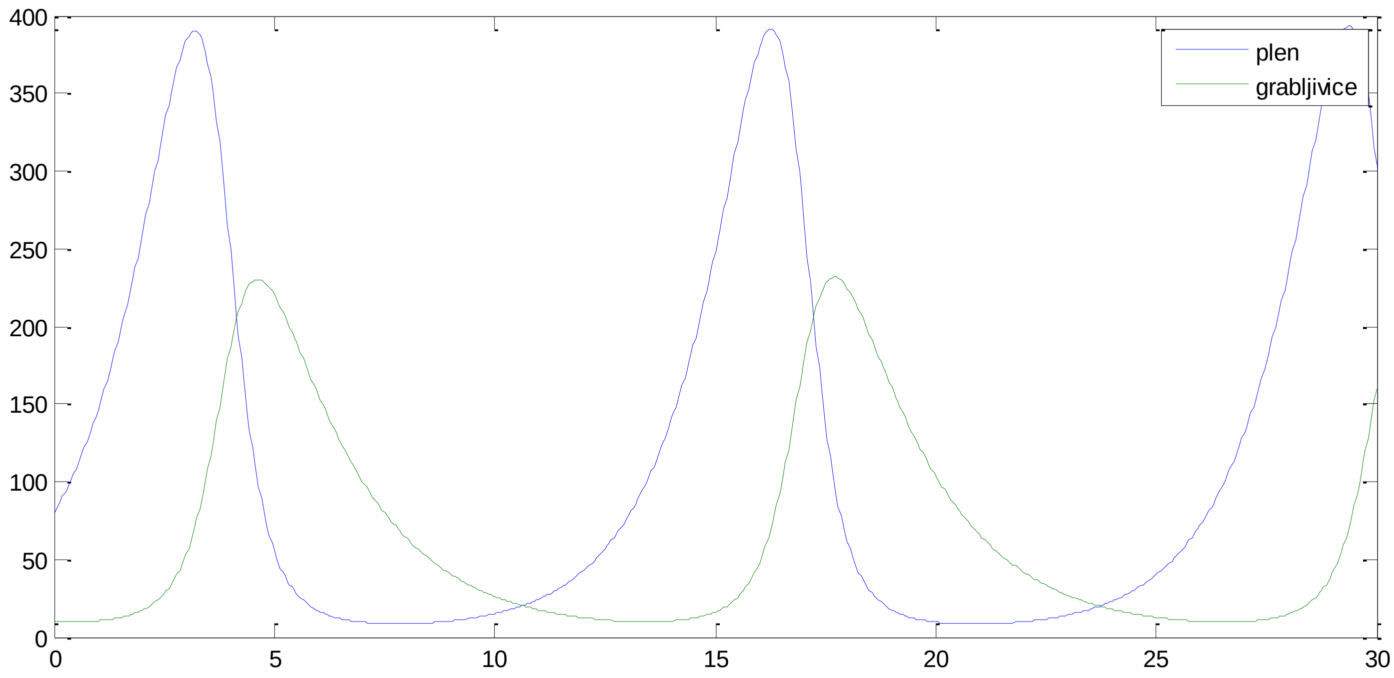
- Promena populacije grabljivica
  - Raste srazmerno ishrani
  - Opada zbog prirodne smrti

$$\frac{dy}{dt} = -\gamma y + \delta xy$$

Simulacija

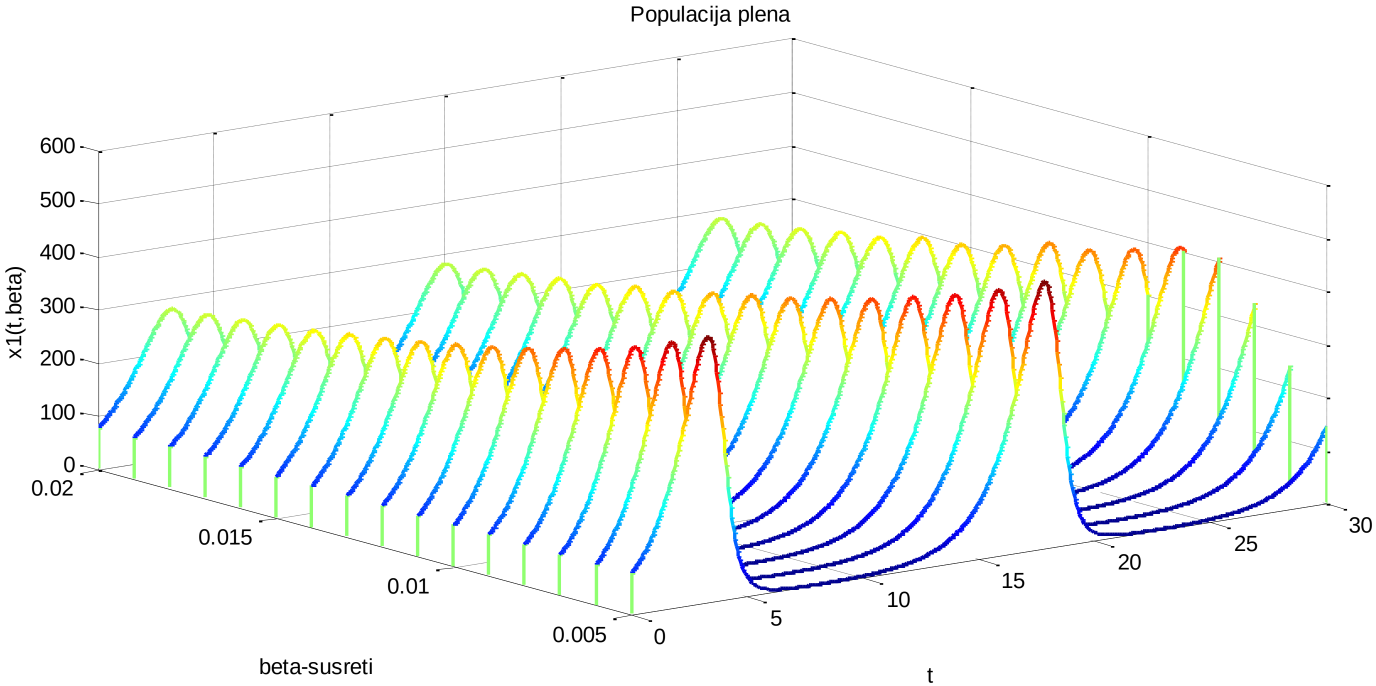
```
function zp = lotka7(t,z,alpha,beta,gama,delta)
% Lotka-Volterra model
x = z(1); y = z(2);
zp = [ x*(alpha - beta*y)
      -y*(gama - delta*x) ];
```

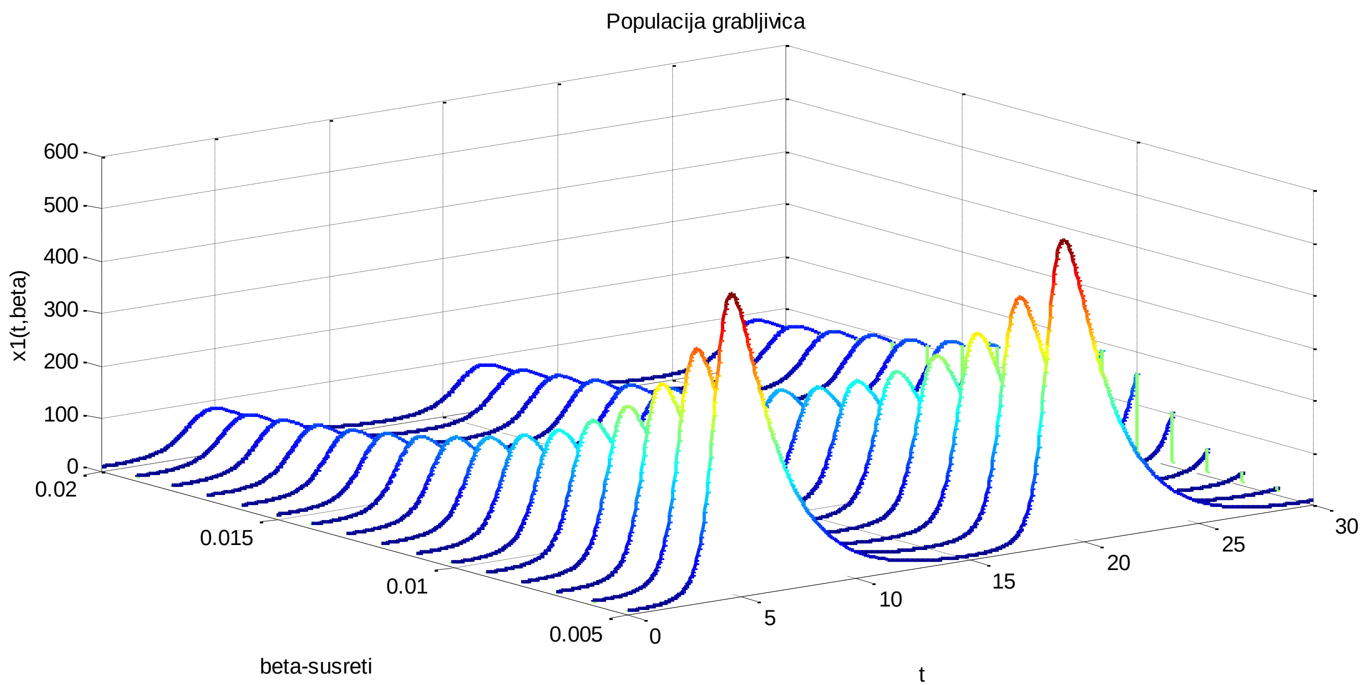
```
al = 0.7; be = 0.01; ga = 0.8; de = 0.05;
tout = 0:0.05:30;
[t,x] = ode45(@lotka7,tout,[40; 80],[], al, be, ga, de);
plot(t,x)
```



Analiza uticaja susreta grabljivica i plena

```
al = 0.7; ga = 0.5; de = 0.005;
beta = 0.005:0.001:0.02;         % parametar susretanja
tout = 0:0.05:30;

xx = []; yy = [];
for be = beta
   [t,x] = ode45(@lotka7, tout, [80; 10], [], al, be, ga, de );
   xx = [xx x(:,1)];
   yy = [yy x(:,2)];
end

[X,Y]=meshgrid(t,beta);
g=waterfall(X,Y,xx');
xlabel('t'), ylabel('beta-susreti'), zlabel('x1(t,beta)')
set(g,'linewidth',2), title('Populacija plena')
pause
g=waterfall(X,Y,yy');
xlabel('t'), ylabel('beta-susreti'), zlabel('x1(t,beta)')
set(g,'linewidth',2), title('Populacija grabljivica')
```


Populacija plena
x1(t,beta)
600
500
400
300
200
100
0
0.02
0.015
0.01
0.005
beta-susreti
0
5
10
15
20
25
30
t


Populacija grabljivica
x1(t,beta)
600
500
400
300
200
100
0
0.02
0.015
0.01
0.005
beta-susreti
0
5
10
15
20
25
30
t

Van der Pol oscilator

- Van der Pol oscilator je nekonzervativni oscilator sa nelinearnim prigušenjem.
- Opisuje se diferencijalnom jednačinom 2. reda:

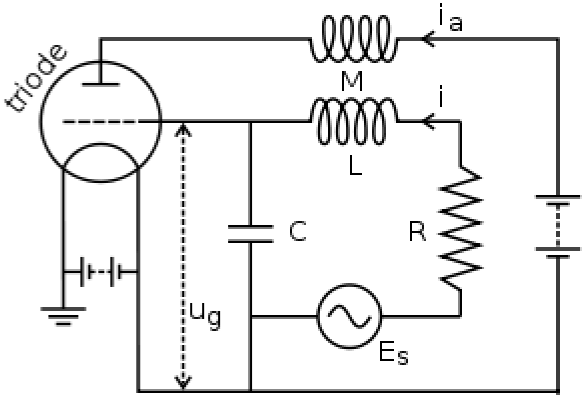
$$\frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x = 0$$

gde je *x* promenljiva zavisna od vremena *t*, a *μ* je parametar (mera nelinearnosti i jačina prigušenja)

- Ovaj oscilator je 1920. godine osmislio holandski el. inženjer i fizičar Balthasar van der Pol dok je radio za Philips.

Balthasar van der Pol

Born	January 27, 1889 Utrecht
Died	October 6, 1959 (aged 70) Wassenaar
Nationality	Dutch
Fields	Physics
Notable awards	IEEE Medal of Honor



Van der Pol jednačina

- Diferencijalna jednačina 2. reda: $\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0$
- Uvodimo smenu: $y = \frac{dx}{dt}$
- Nakon diferenciranja dobijamo: $\frac{dy}{dt} = \frac{d^2x}{dt^2} = \mu(1 - x^2)\frac{dx}{dt} - x$

- Konačno model od dve diferencijalne jednačine 1. reda (ekvivalentan polaznoj jednačini)

$$\frac{dx}{dt} = y$$

$$\frac{dy}{dt} = \mu(1 - x^2)y - x$$

```
function zprim = vdpol(t, z, mi)
% Opis Van der Pol-ove dif. jednacine
x = z(1);
y = z(2);
xprim = y;
yprim = mi*(1-x^2)*y – x;
zprim = [xprim; yprim];
```

Van der Pol simulacija

```
tp = 0;  tk = 100;  x0 = [0.5 0];

mi = 0.2;
[t, x] = ode45( @vdpol, [tp tk], x0, [], mi);
subplot(3,1,1), plot( t, x(:,1) ),
title('mi=0.2'), axis([0 100 -3 3])

mi = 1;
[t, x] = ode45( @vdpol, [tp tk], x0, [], mi);
subplot(3,1,2), plot( t, x(:,1) ),
title('mi=1'), axis([0 100 -3 3])

mi = 10;
[t, x] = ode45( @vdpol, [tp tk], x0, [], mi);
subplot(3,1,3), plot( t, x(:,1) ),
title('mi=10'), axis([0 100 -3 3])
```