

დიფერენციალური განტოლებები: ეილერის მეთოდი

საწყისი მნიშვნელობის ამოცანა:

$$y' = f(x, y); \quad y(x_0) = y_0.$$

$$y(x_n + h) \approx y(x_n) + hy'(x_n).$$

$$y_{n+1} = y_n + hy'_n.$$

$$y'_n = f(x_n, y_n),$$

$$y_{n+1} = y_n + hf(x_n, y_n).$$

მაგალითი:

$$y' = 0.5y$$

ანალიზური ამონახსნი:

$$y(x) = e^{x/2}.$$

დისკრეტული თარგი (stencil):

$$y_{n+1} = y_n + h \frac{y_n}{2}$$

$$= \left(1 + \frac{h}{2}\right) y_n.$$

x	Euler(x)	Exact(x)
0.00	1.00000	1.00000
0.05	1.02500	1.02532
0.10	1.05063	1.05127
0.15	1.07689	1.07788
0.20	1.10381	1.10517
0.25	1.13141	1.13315
$\vdots$	$\vdots$	$\vdots$
1.00	1.63862	1.64872
2.00	2.68506	2.71828
3.00	4.39979	4.48169
$\vdots$	$\vdots$	$\vdots$
5.00	11.81372	12.18249
10.00	139.56389	148.41316

ეილერის მეთოდი: მეორე (და მაღალი) რიგის დიფ. განტოლებები

$$y'' = f(x, y) ,$$

დამხმარე ცვლადები და ორი პირველი რიგის დიფ. განტოლებების სისტემა:

$$\begin{aligned} Z_1 &\equiv y & Z_2' &= f(x, Z_1) \\ Z_2 &\equiv y' & Z_1' &= Z_2 \end{aligned}$$

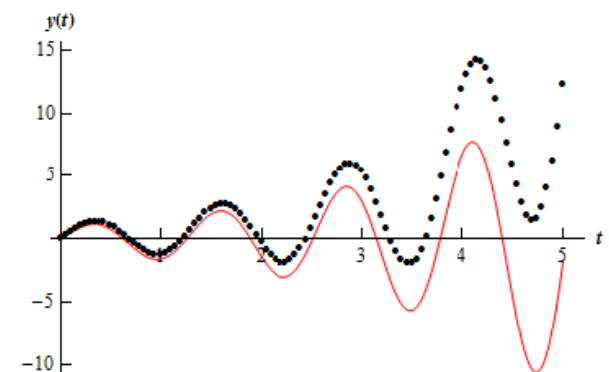
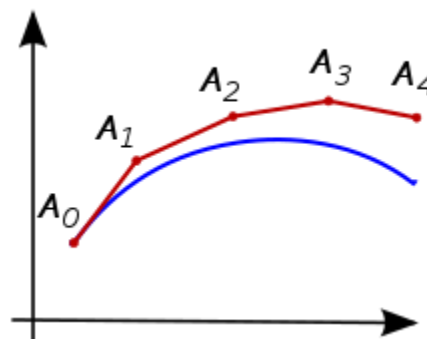
ეილერის დისკრეტიზაცია („მარშირება“):

$$\begin{aligned} Z_2^{n+1} &= Z_2^n + hf(x_n, Z_1^n) \\ Z_1^{n+1} &= Z_1^n + hZ_2^n \end{aligned}$$

N რიგის დიფ. განტოლება: $(N-1)$ დამხმარე ცვლადი და N გადახმული პირველი რიგის დიფ. განტოლება;

ეილერის მეთოდის თვისებები

- + სიმარტივე;
- სიზუსტე;
- შეცდომის დაგროვება;



ჰუნის მეთოდი (Predictor-corrector method)

$$y' = f(t, y), \quad y(t_0) = y_0,$$

1. Predict:

“ამხსნელი ცვლადი”

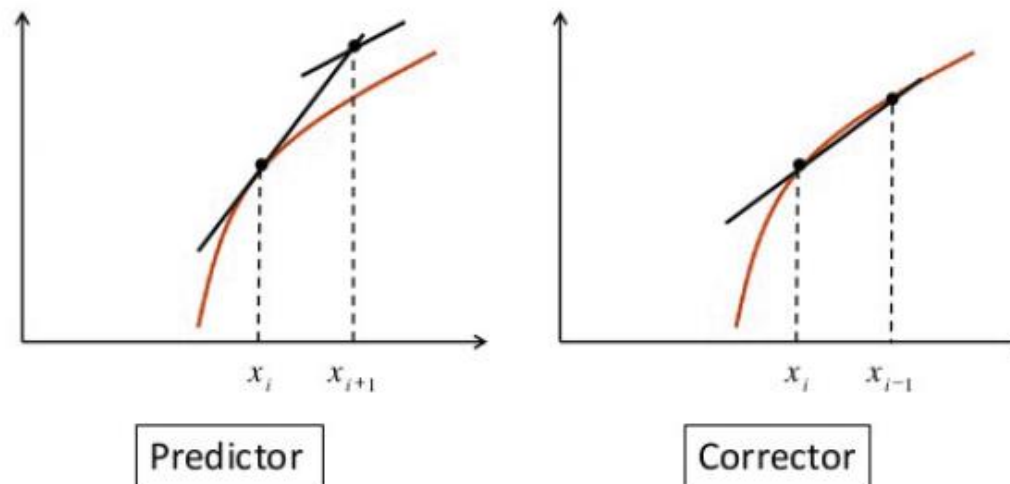
$$\tilde{y}_{i+1} = y_i + hf(t_i, y_i).$$

2. Correct:

„კორექტორი“

$$y_{i+1} = y_i + \frac{1}{2}h(f(t_i, y_i) + f(t_{i+1}, \tilde{y}_{i+1})).$$

წარმოებულით დახრის კუთხის წინასწარმეტყველება და შესწორება: დახრის კუთხის გასაშუალოება;



დახრის კუთხის შესაფასებლად ვიყენებთ ორ (საწყის და საბოლოო) წერტილს: მეორე რიგის სქემა;
N წერტილის გამოყენებით შეგვიძლია გამოვიყვანოთ N-ე რიგის დისკრეტული სქემა;

განზოგადოებული რუნგე-კუტა სქემები

$$\frac{dy}{dt} = f(t, y)$$

დისკრეტული თარგი:

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i$$

სადაც:

$$k_i = f \left(t_n + c_i h, y_n + h \sum_{j=1}^s a_{ij} k_j \right).$$

დისკრეტული სქემის პარამეტრები: a_{ij} , b_i , c_i

c_i - პრედიქტორების რაოდენობა;

სქემის ვიზუალიზაცია: ბაჩერის ცხრილი (Batcher tableau)

c_1	a_{11}	a_{12}	$\dots$	a_{1s}
c_2	a_{21}	a_{22}	$\dots$	a_{2s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	$\dots$	a_{ss}
	b_1	b_2	$\dots$	b_s

დისკრეტული მარშირების სქემის ვიზუალიზაცია ცხრილით;

მაგალითი:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

ეილერის მეთოდის წარმოდგენა ბაჩერის ცხრილით:

ზოგიერთი დისკრეტული მეთოდი

რუნგე-კუტას მესამე რიგის მეთოდი RK3:

0	0	0	0
1/2	1/2	0	0
1	-1	2	0
	1/6	2/3	1/6

რუნგე-კუტას კლასიკური მეოთხე რიგის მეთოდი RK4:

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
	1/6	1/3	1/3	1/6

შესაბამისი დისკრეტული თარგი:

$$y_{n+1} = y_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

$$t_{n+1} = t_n + h$$

$$k_1 = f(t_n, y_n),$$

$$k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1),$$

$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2),$$

$$k_4 = f(t_n + h, y_n + hk_3).$$

რუნგე-კუტას 3/8 რიგის მეთოდი (Kutta 1901):

0	0	0	0	0
1/3	1/3	0	0	0
2/3	-1/3	1	0	0
1	1	-1	1	0
	1/8	3/8	3/8	1/8

დისკრეტული სქემა ფიქსირებული სიზუსტით: ადაპტური ბიჯი

დისკრეტული თარგი:

$$y_{n+1}^* = y_n + h \sum_{i=1}^s b_i^* k_i,$$

ცდომილება:

$$e_{n+1} = y_{n+1} - y_{n+1}^* = h \sum_{i=1}^s (b_i - b_i^*) k_i,$$

ცდომილების წინასწარ მოცემული სიდიდე: *tolerance*

$e_n < tolerance$: შემდეგი ბიჯი

$e_n > tolerance$: ბიჯის განახევრება - RK სქემის რიგის მომატება, RK4→RK5

c_1	a_{11}	a_{12}	$\dots$	a_{1s}
c_2	a_{21}	a_{22}	$\dots$	a_{2s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	a_{s2}	$\dots$	a_{ss}
	b_1	b_2	$\dots$	b_s
	b_1^*	b_2^*	$\dots$	b_s^*

დორმანდ-პრინცის მეთოდი (MATLAB **ode45**):

0							
1/5	1/5						
3/10	3/40	9/40					
4/5	44/45	-56/15	32/9				
8/9	19372/6561	-25360/2187	64448/6561	-212/729			
1	9017/3168	-355/33	46732/5247	49/176	-5103/18656		
1	35/384	0	500/1113	125/192	-2187/6784	11/84	
	5179/57600	0	7571/16695	393/640	-92097/339200	187/2100	1/40
	35/384	0	500/1113	125/192	-2187/6784	11/84	0

MATLAB ODE

```
[T,Y] = solver(odefun,tspan,y0,options)
```

solvers: ode23, **ode45**, ode113, ode15s, ode23s, ode23t

odefun ფუნქციის სახელი

tspan დროის (არგუმენტის) ვექტორი

y0 საწყისი მნიშვნელობები

options: relative tolerance, absolute tolerance;

მატლაბის სტანდარტული სოლვერები:

Solver	Problem Type	Order of Accuracy	When to Use
ode45	Nonstiff	Medium	Most of the time. This should be the first solver you try.
ode23	Nonstiff	Low	For problems with crude error tolerances or for solving moderately stiff problems.
ode113	Nonstiff	Low to high	For problems with stringent error tolerances or for solving computationally intensive problems.
ode15s	Stiff	Low to medium	If ode45 is slow because the problem is stiff.
ode23s	Stiff	Low	If using crude error tolerances to solve stiff systems and the mass matrix is constant.
ode23t	Moderately Stiff	Low	For moderately stiff problems if you need a solution without numerical damping.
ode23tb	Stiff	Low	If using crude error tolerances to solve stiff systems.

მაგალითი:

$$\frac{d}{dt}y(t) = \sin(t) - y(t)$$

$$y(0) = 1$$

ode_eqs.m

```
function dy = ode_eqs(t,y);  
  
dy = sin(t) - y;
```

main.m

```
t0 = 0;  
tfin=15;  
  
NSTEPS = 100;  
  
for j=1:NSTEPS  
    tspan(j)= t0 + (j-1)*(tfinal-t0)/(NSTEPS-1);  
end  
  
options = odeset('RelTol',1e-5, 'AbsTol',1e-7);  
  
y0 = 1;  
  
[t,y] = ode45(@ode_eqs,tspan,y0,options);  
  
figure(1);  
plot(t,y);  
xlabel('t');  
ylabel('y');
```


მეორე რიგის ჩვეულებრივ წარმოებულნი დიფ. განტოლება:

$$\frac{d^2}{dt^2}y(t) = -2\pi^2 \sin(t)^2 y(t)$$

ახალი ცვლადები:

$$\begin{aligned} y_1 &\equiv y \\ y_2 &\equiv \frac{dy}{dt} \end{aligned}$$

წრფივი სისტემა:

$$\frac{d}{dt}y_1(t) = y_2(t)$$

$$\frac{d}{dt}y_2(t) = -2\pi^2 \sin(t)^2 y_1(t)$$

main.m

```
t0 = 0; tfinal=2;
NSTEPS = 100;

y1 = 0; y2 = 1;

for j=1:NSTEPS
    tspan(j)= t0 + (j-1)*(tfinal-t0)/(NSTEPS-1);
end

options = odeset('RelTol',1e-5, 'AbsTol',1e-7);
init = [y1,y2];

[t,y] = ode45(@oden_eqs,tspan,init,options);
A1 = y(:,1);
A2 = y(:,2);

figure(1);
plot(t,A1);
xlabel('t');
ylabel('y');
```

oden_eqs.m

```
function dy = oden_eqs(t,y);

dy = ones(2,1);

dy(1) = y(2);
dy(2) = 2*pi^2*sin(t)^2*y(1);
```

დავალება 8.1.

იპოვეთ ლორენცის ატრაქტორის ამონახსნი (0-500) დროით ინტერვალში სხვადასხვა შემთხვევითი საწყისი პირობებისთვის:

$$\begin{aligned}x'(t) &= 10(y - x) & x(0) &= 5rand; \\y'(t) &= x(28 - z) - y & y(0) &= 5rand; \\z'(t) &= xy - 2.7z & z(0) &= 5rand;\end{aligned}$$

დავალება 8.2.

ამოხსენით ჰელმჰოლცის განტოლება

$$-\frac{1}{r} \frac{d}{dr} \left(r \frac{dB_z}{dr} \right) = \lambda^2 B_z$$

სასაზღვრო პირობით $B_z(0)=0$, $dB_z(0)/dr=1$ შემდეგი პარამეტრებისათვის:

ა) $\lambda^2(r) = \frac{1}{1+r^2}$

ბ*) $\lambda(r) \log(\lambda(r)) = 1 + r^2$