

დიფერენციალური განტოლებები: ეილერის მეთოდი

საწყისი/სასაზღვრო მნიშვნელობის ამოცანა:

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

$$y(x_n + h) = y(x_n) + hy'(x_n) + h^2 \frac{y''(X)}{2},$$

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

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

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

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

მაგალითი:

განტოლება: $y' = 0.5y$

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

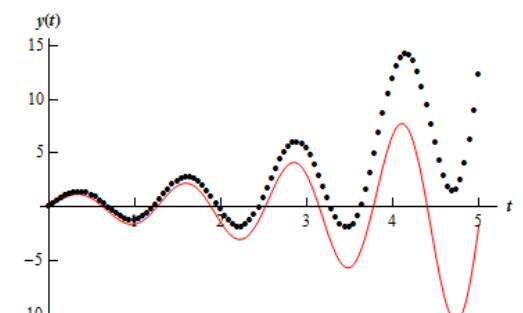
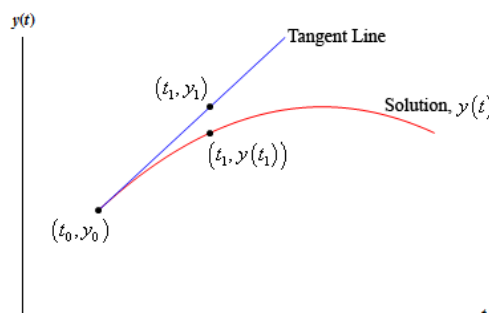
დისკრეტული შაბლონი:

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

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

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

- + მარტივი
- სიზუსტე
- შეცდომის დაგროვება



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

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

დამხმარე ცვლადები:

$$Z_1 \equiv y$$

$$Z_2 \equiv y'$$

პირველი რიგის დიფ განტოლებების სისტემა:

$$Z_2' = f(x, Z_1)$$

$$Z_1' = Z_2$$

ეილერის ფორმულირება („მარშირება“):

$$Z_2^{n+1} = Z_2^n + hf(x_n, Z_1^n)$$

$$Z_1^{n+1} = Z_1^n + hZ_2^n$$

N რიგის დიფ. განტოლება:

N დამხმარე ცვლადი და N ჩაბმული პირველი რიგის დიფ. განტოლება.

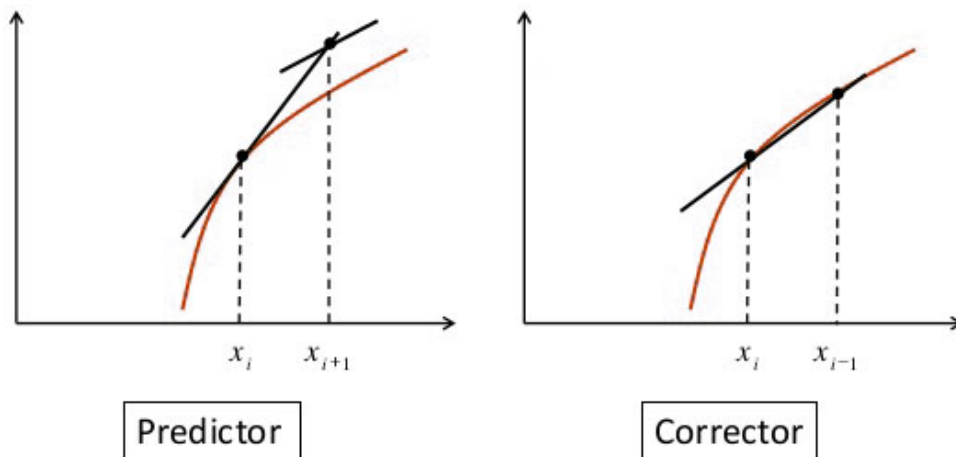
Predictor-Corrector მეთოდი

(ჰუნის მეთოდი)

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

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

წარმოებულის წინასწარმეტყველება და „შესწორება“: დახრის კუთხის გასაშუალოება.



მეორე რიგის predictor-corrector მეთოდი.

დახრის კუთხის შესაფასებლად შესაძლებელია მეტი მეზობელი წერტილის გამოყენება. ზოგადად n წერტილის გამოყენებისას ვამბობთ, რომ დისკრეტული სქემა n რიგისაა.

მაგალითად: რუნგე კუტას კლასიკური სქემა (the Runge-Kutta method “RK4”):

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

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

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

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

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

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

განზოგადოებული რუნგე-კუტა სქემები

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

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

სადაც:

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

c_i - პრეკიდტორების რაოდენობა (და ბიჯი) h ინტერვალის შიგნით.

a_{ij} , b , c პარამეტრებს განსაზღვრავს ბაჩერის ცხრილი (Butcher tableau):

ამ ცხრილის მეშვეობით შესაძლებელია რუნგე-კუტა მეთოდის მოკლე რეკურენტული ჩაწერა.

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

ბაჩერის ცხრილის გამოყენებით შეგვიძლია მოკლედ ჩავწეროთ უკვე განხილული დისკრეტული სქემები:

პირველი რიგის (ეილერის მეთოდი):

0	0
	1

ზოგადი მეორე რიგის მეთოდი:

0	0	0
x	x	0
	$1 - \frac{1}{2x}$	$\frac{1}{2x}$

რუნგე-კუტას მესამე რიგის მეთოდი:

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

კლასიკური რუნგე-კუტას მეოთხე რიგის მეთოდი:

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

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 სქემის ფიქსირებული სიზუსტე: ადაპტური ბიჯი

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

ცდომილება:

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

ცდომილების მოცემული სიდიდე: tol

$e_i > tol$: ბიჯის განახევრება//რიგის მომატება: RK4->RK5
 $e_i < tol$: შემდეგი ბიჯი;

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: function_name.m

tspan: time vector

y0: initial conditions

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:

$$\begin{aligned} \frac{d}{dt} y(t) &= \sin(t) - y(t); \\ y(0) &= 1; \end{aligned}$$

ode_eqs.m

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

main.m

```
t0 = 0;  
tfinal=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');
```

მაღალი რიგის ჩვეულებრივ წარმოებულნი განტოლება:

$$y''(t) = -2\pi^2 \sin^2(t) y(t)$$

$$\begin{aligned} y_1 &= y \\ y_2 &= y' \end{aligned}$$

$$\begin{aligned} y_1' &= y_2; \\ y_2'(t) &= -2\pi^2 \sin(t)^2 y_1; \end{aligned}$$

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);
```

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');
```

დავალება:

1. ამოხსენით 4-ე რიგის დიფ. განტოლება დროით ინტერვალში (0-100) შემდეგი საწყისი პირობებით:

$$5z'''' + 4z''' + 3z'' + 2z' + z = \sin(t)$$

$$z(0) = 0$$

$$z'(0) = 10$$

$$z''(0) = 20$$

$$z'''(0) = 30$$

2. იპოვეთ ლორენცის ატრაქტორის ამონახსნის ტრაექტორია 0-50 დროით ინტერვალში სხვადასხვა (შემთხვევითი) საწყისი პირობებისათვის:

$$x'(t) = 10(y-x)$$

$$y'(t) = x(28-z) - y$$

$$z'(t) = xy - 2.7z$$

საწყისი პირობები:

$$x(0) = 5 * \text{rand};$$

$$y(0) = 5 * \text{rand};$$

$$z(0) = 5 * \text{rand};$$

3. ამოხსენით ჰელმჰოლცის განტოლება სასაზღვრო პირობით $B_z(0)=1$:

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

და

$$1) \quad \lambda^2(r) = \frac{\lambda_0^2}{1+r^2}$$

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