```
%% PART 1: 5 NON-STIFF ODEs – for Convergence Analysis
%% (Use analytical solution or high-accuracy ode45 as reference)


% Problem 1.1: Linear, Gaussian decay
f1 = @(t, y) -2 * t * y;
y0_1 = 1;
tspan_1 = [0, 3];
y_exact_1 = @(t) exp(-t.^2);  % Analytical

% Problem 1.2: Linear nonhomogeneous
f2 = @(t, y) y - t^2 + 1;
y0_2 = 0.5;
tspan_2 = [0, 2];
y_exact_2 = @(t) (t+1).^2 - 0.5 * exp(t);  % Analytical

% Problem 1.3: Forced linear oscillator
f3 = @(t, y) cos(t) - y;
y0_3 = 0;
tspan_3 = [0, 5];
y_exact_3 = @(t) 0.5 * (sin(t) - cos(t) + exp(-t));  % Analytical

% Problem 1.4: Nonlinear Riccati-type
f4 = @(t, y) y.^2 - t;
y0_4 = 1;
tspan_4 = [0, 1.5];
% No analytical solution → use ode45 with tight tolerances
[t_ref4, y_ref4] = ode45(f4, tspan_4, y0_4,
odeset('RelTol',1e-12,'AbsTol',1e-14));
y_ref_4 = @(t) interp1(t_ref4, y_ref4, t, 'spline');

% Problem 1.5: Damped forced oscillator
f5 = @(t, y) -y + 2 * exp(-t) .* cos(2*t);
y0_5 = 0;
tspan_5 = [0, 4];
y_exact_5 = @(t) exp(-t) .* sin(2*t);  % Analytical


% Problem 1.6
fA1 = @(t, y) y .* (1 - y);
y0_A1 = 0.1;
tspan_A1 = [0, 5];
y_exact_A1 = @(t) 1 ./ (1 + 9 * exp(-t));


% Problem 1.7
fA2 = @(t, y) -y + exp(-2*t);
y0_A2 = 1;
tspan_A2 = [0, 3];
```

```
% Решение: y(t) = (1/3)*exp(-2*t) + (2/3)*exp(-t)
y_exact_A2 = @(t) (1/3)*exp(-2*t) + (2/3)*exp(-t);


% Problem 1.8
fA3 = @(t, y) y - y.^3;
y0_A3 = 0.5;
tspan_A3 = [0, 4];
y_exact_A3 = @(t) 1 ./ sqrt(1 + 3 * exp(-2*t));


% Problem 1.9
fA4 = @(t, y) sin(y);
y0_A4 = 1;
tspan_A4 = [0, 3];
y_exact_A4 = @(t) 2 * atan( tan(0.5) * exp(t) );


% Problem 1.10
fA5 = @(t, y) -y + t.^3;
y0_A5 = 0;
tspan_A5 = [0, 2];
y_exact_A5 = @(t) t.^3 - 3*t.^2 + 6*t - 6 + 6*exp(-t);


%% PART 2: 5 STIFF SCALAR ODEs — for Stability Comparison
%% (Use ode15s with tight tolerances as reference)


% Problem 2.1

lambda_B1 = 1500;
fB1 = @(t, y) -lambda_B1 * y + exp(-t);
y0_B1 = 1;
tspan_B1 = [0, 1];


% Problem 2.2
lambda2 = 1000;
f7 = @(t, y) -lambda2 * y + sin(t);
y0_7 = 1;
tspan_7 = [0, 2];

% Problem 2.3
lambda3 = 500;
f8 = @(t, y) -lambda3 * (y - cos(t));
y0_8 = 0;
tspan_8 = [0, 5];
```

```
% Problem 2.4
lambda4 = 2000;
f9 = @(t, y) -lambda4 * y + t;
y0_9 = 0;
tspan_9 = [0, 1];


% Problem 2.5
lambda5 = 800;
f10 = @(t, y) -lambda5 * (y - exp(-t));
y0_10 = 0;
tspan_10 = [0, 3];

% Problem 2.6
lambda_B2 = 1000;
fB2 = @(t, y) -lambda_B2 * (y - sin(0.1*t));
y0_B2 = 0;
tspan_B2 = [0, 5];


% Problem 2.7
lambda_B3 = 2000;
fB3 = @(t, y) -lambda_B3 * y + (t < 0.5);  % Хевисайда: 1 при t<0.5, 0 иначе
y0_B3 = 0;
tspan_B3 = [0, 1];


% Problem 2.8
lambda_B4 = 800;
fB4 = @(t, y) -lambda_B4 * (y.^2 - 1);  % Два устойчивых состояния ±1
y0_B4 = 0.5;  % Начинаем между ними → быстрая динамика
tspan_B4 = [0, 0.1];  % Достаточно короткий интервал



% Problem 2.9
lambda_B5 = 2500;
fB5 = @(t, y) -lambda_B5 * y + 10 * cos(5*t);
y0_B5 = 0;
tspan_B5 = [0, 0.5];


% Problem 2.10
lambda_B6 = 3000;
fB6 = @(t, y) -lambda_B6 * y + lambda_B6 * exp(-10*t);
y0_B6 = 0;
tspan_B6 = [0, 0.2];


% For all stiff problems, use ode15s as reference
```

```
ref_solver = @(f, tspan, y0) ode15s(f, tspan, y0,
odeset('RelTol',1e-12,'AbsTol',1e-14));

% Example: get reference for problem 6
[t_ref6, y_ref6] = ref_solver(f6, tspan_6, y0_6);
y_ref_6 = @(t) interp1(t_ref6, y_ref6, t, 'spline');
```