

---

```

clc;
clear;
close all;

n = 200;
lambdas = linspace(6.7, 6.8, n);
y_t_f = zeros(length(lambdas), 1);

t_0 = 0;
t_f = 1;
y_0 = [0; 1];

% Maximum number of rows for Richardson extrapolation
max_rows = 17;
tol = 1e-15;

for k = 1:length(lambdas)
    lambda = lambdas(k);

    % Richardson extrapolation matrix
    A = zeros(max_rows, max_rows);

    h = (t_f - t_0);
    y = trapezoid(@odefun, t_0, t_f, lambda, h, y_0);

    A(1, 1) = y(1);

    for i = 1:(max_rows - 1)
        h = h/2;
        y = trapezoid(@odefun, t_0, t_f, lambda, h, y_0);
        A(i + 1, 1) = y(1);

        for j = 1:i
            A(i + 1, j + 1) = ((4^j)*A(i + 1, j) - A(i, j))/((4^j) - 1);
        end

        if abs(A(i + 1, i + 1) - A(i, i)) < tol
            break;
        end
    end

    y_t_f(k) = abs(A(max_rows, max_rows));
end

% Find the value of lambda that makes y(1) closest to 0
optimal_lambda_idx = find(y_t_f == min(y_t_f));
fprintf("Optimal lambda: %0.16f\n", lambdas(optimal_lambda_idx));

function [f] = odefun(t, y, lambda)
    f = [y(2); ...
        (1/(1+t))*y(2) - (1+t)*lambda*y(1)];
end

```

---

---

```
function [y] = trapezoid(odefun, t_0, t_f, lambda, h, y_0)
    y = y_0;

    while t_0 <= t_f
        E_y = y + (h/2)*odefun(t_0, y, lambda);
        I_m = [1, -h/2;
                (h/2)*(1 + (t_0 + h))*lambda, 1 - (h/2)*(1/(1 + (t_0 + h)))];
        I_y = I_m\E_y;

        y = E_y + (h/2)*odefun(t_0 + h, I_y, lambda);
        t_0 = t_0 + h;
    end
end
```

*Optimal lambda: 6.7738693467336679*

*Published with MATLAB® R2022a*