```
clear
syms x y z
sigma   = 16;
b       = 5;
r       = 330;

%J(x,y,z) = jacobian([sigma*(y-x); r*x - y - x*z; x*y - b*z], [x y z])

f = @(t,x)[sigma*(x(2) - x(1)); r*x(1) - x(2) - x(1)*x(3); x(1)*x(2) -
 b*x(3)];

t_to_discard = 20;
start = [1,1,1];


[~,xData] = ode45(@(t,x) f(t,x), [0 t_to_discard], start);
x0 = xData(end, :)';

dt = 1e-3;%1e-7;
T = 1e3;
%T = t_to_discard + dt;
timeVector = t_to_discard:dt:T;
figureFactor = 10000;

Q = eye(3);
lambda = zeros(3,1);
tic
lambdaData = zeros(ceil(length(timeVector)/figureFactor)-1,3);
timeData = zeros(ceil(length(timeVector)/figureFactor)-1,1);

for j = 1:length(timeVector)
    if mod(j, round(length(timeVector)/10000)) == 0
        fprintf('%.2f %% done.\n', 100*j/length(timeVector))
    end
    x0 = x0 + dt*[sigma*(x0(2) - x0(1)); r*x0(1) - x0(2) -
 x0(1)*x0(3); x0(1)*x0(2) - b* x0(3)];

%     [t,xData] = ode45(@(t,x) f(t,x), [0 dt], x0);
%     x0 = xData(end, :);

    currentJacobian = [-sigma, sigma,0; r - x0(3), -1,   -x0(1);
 x0(2),  x0(1), -b];
    M = eye(3) + currentJacobian*dt;
    [Q, R] = qr(M*Q);
    lambda = lambda + log(abs(diag(R)));
    if mod(j, figureFactor) == 0
        lambdaData(j/figureFactor, :) = lambda/(timeVector(j)-
t_to_discard);
        timeData(j/figureFactor) = timeVector(j);
    end
end
disp(lambda/(T-t_to_discard))
```

```
toc

save('task3c.mat')

for j = 1:3
    figure(j)
    subplot(1,2,1)
%     plot(timeVector(round(end/2):end),
 lambdaData(round(end/2):end,i))
    plot(timeData, lambdaData(:,j))

    subplot(1,2,2)
    semilogx(timeData, lambdaData(:,j))
end
```

Published with MATLAB® R2020b