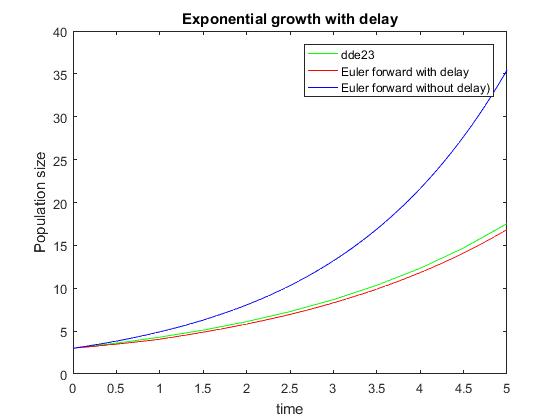
Timothy och Mika

1.14 Time delay



clc

clear all

close all

% Set parameters

tMax = 5; % Max time

dt = 0.05; % Time step

timeVector = 0:dt:tMax; % Time stamps

nIterations = length(timeVector); % Number of iterations in Euler forward

growthRate = 0.5; % Exponential growth factor

y0 = 3; % Initial condition

timeSpan = [0 tMax]; % Time span (used in MATLAB ODE solver)

% Set dy/dt function

exp\_growth = @(t,y,Z) growthRate.\*Z;

tau=1;

r2=0.3;

historyfunc=@(t) y0\*exp(r2\*t);

% Matlab inbuilt ODE-solver

sol = dde23(exp\_growth, tau, historyfunc, timeSpan);

t=sol(1).x;

y=sol(1).y;

% Plot the solution

figure(1);

plot(t,y,'g');

title('Exponential growth with delay');

xlabel('time');

ylabel('Population size');

hold on

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%Euler with delay

y=1;

t=0;

for i=1:(tau/dt)

y(i)=y0\*exp(r2\*t);

t=t+dt;

end

for i=(tau/dt):length(timeVector)-1

y(i+1)=y(i)+dt\*growthRate\*y(i-(-1+tau/dt));

end

plot(timeVector,y,'r');

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

i=1;

y=3;

for i=1:length(timeVector)-1

y(i+1)=y(i)+dt\*growthRate\*y(i);

end

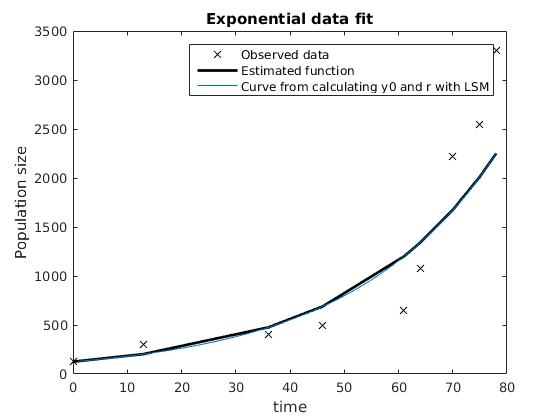
plot(timeVector,y,'b');

legend('dde23','Euler forward with delay', 'Euler forward without delay)');

Difference between delay and no delay: Growth slower with maturation delay.

Difference between Euler and dde23: dde23 is more accurate, but Euler is a good estimation if a low steplength is used.

1.2 Linear regression



close all

clear all

load population\_data\_swedish\_bears.mat;

t = t - t(1); % Shift the time so that it starts from t = 0

% Plot observed data

figure(3);

plot(t,data,'kx');

title('Exponential data fit');

xlabel('time');

ylabel('Population size');

% Make data linear

lnData = log(data);

tData = [ones(size(t)),t];

% Estimate parameters using MATLAB row reduction

parameters = tData\lnData;

y0 = exp(parameters(1)); % y0 = exp(lny0);

growthFactor = parameters(2);

% Generate data using the exponential model with estimated parameters

data\_fit = y0.\*exp(growthFactor.\*t);

% Plot resulting data

figure(3);

hold on

plot(t,data\_fit,'k-','linewidth',2);

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%y=y0.\*exp(r.\*t);

%log(y)=log(y0) + r\*t

%Y=b+m\*t

tid=1:78;

c=polyfit(t,log(data),1);

r=c(1);

y0=exp(c(2));

yy=y0.\*exp(r.\*tid);

plot(tid,yy);

legend('Observed data','Estimated function','Curve from calculating y0 and r with LSM');

derp=polyval(c,tid);

figure(2)

plot(tid,derp);

%Från boken (det andra är mer effektivt)

% sumy=sum(log(data));

% sumx=sum(t);

% t2=t.\*t;

% sumx2=sum(t2);

% tdata=t.\*log(data);

% sumxy=sum(tdata);

% m=(9\*sumxy-sumx\*sumy)/(9\*sumx2-sumx^2);

% b=(sumx2\*sumy-sumx\*sumxy)/(9\*sumx2-sumx^2);

%

% tid=0;

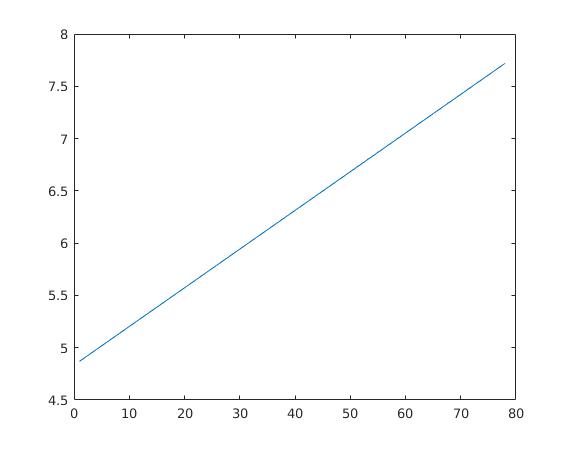
% Y=0;

% for i=1:78

% Y(i)=m\*tid+b;

% tid=tid+1;

% end



Uträkningarna ger att r = 0.037 och att y0 = 125.5619

(där y0 fås av att e^m)