Zarek McGee  
MTH 410- Optim 2  
HW 4  
03/04/2019

**Parameter Estimation in a Discrete Dynamical System***The Logistic Pop. Growth Model*

Task 1: See attached page.

Task 2:

**Code listing:**

function optimP\_ZM(x0,ym,m,h)

% INPUT:

% x0: initial value of x

% ym: data vector (goal for x(m))

% m: time state of the model

% h: time step

% Returns optimal parameter p\*

[p,fval,exitflag] = fsolve(@(p) costF(x0,p,ym,m,h),0);

% Forward integration to get x using optimal p\*

x = xModel(x0,p,m,h);

% Backward integration to get lam using optimal p\* and x(m)

lam = lambdas(x,p,ym,m,h);

% mu = dF/dp [ F(x0,p) ]

% For j=m:-1:1, mu = mu + h \* lam(j). Equivalently:

mu = h\* sum(lam);

% Compute cost function

f = fcost(x,ym,m);

% Plot results

figure(1)

plot(x); xlabel('model state'); ylabel('x value')

figure(2)

plot(lam); xlabel('model state'); ylabel('Lagrange Multiplier')

% Print results

x, lam, p, f, mu

endfunction

%System of equations given by x(j) = Model\_j-1(x\_j-1)

function [x] = xModel(x0,p,m,h)

% Initialize x vector

x = zeros(m+1,1);

x(1) = x0;

% System of equations of p and x

for j=2:m+1

x(j) = x(j-1) + h\*( x(j-1)\*(1 - x(j-1) + p) );

end

endfunction

function [lam] = lambdas(x,p,ym,m,h)

% Initialize Lagrange Multiplier vector

lam = zeros(m+1,1);

% Backwards integration from lam\_m -> lam\_0

lam(m+1) = x(m+1) - ym;

for j=m:-1:1

lam(j) = lam(j+1)\*(1 + h - 2\*h\*x(j+1));

end

endfunction

% Reduced cost function, F(x0,p), with constant params m & h

% Note that gradient F = [lam(1); mu]

function mu = costF(x0,p,ym,m,h)

x = xModel(x0,p,m,h);

lam = lambdas(x,p,ym,m,h);

mu = h \* sum(lam);

endfunction

% Cost function

function f = fcost(x,ym,m)

f = 0.5\*(x(m+1) - ym)^2;

endfunction

Task 3:

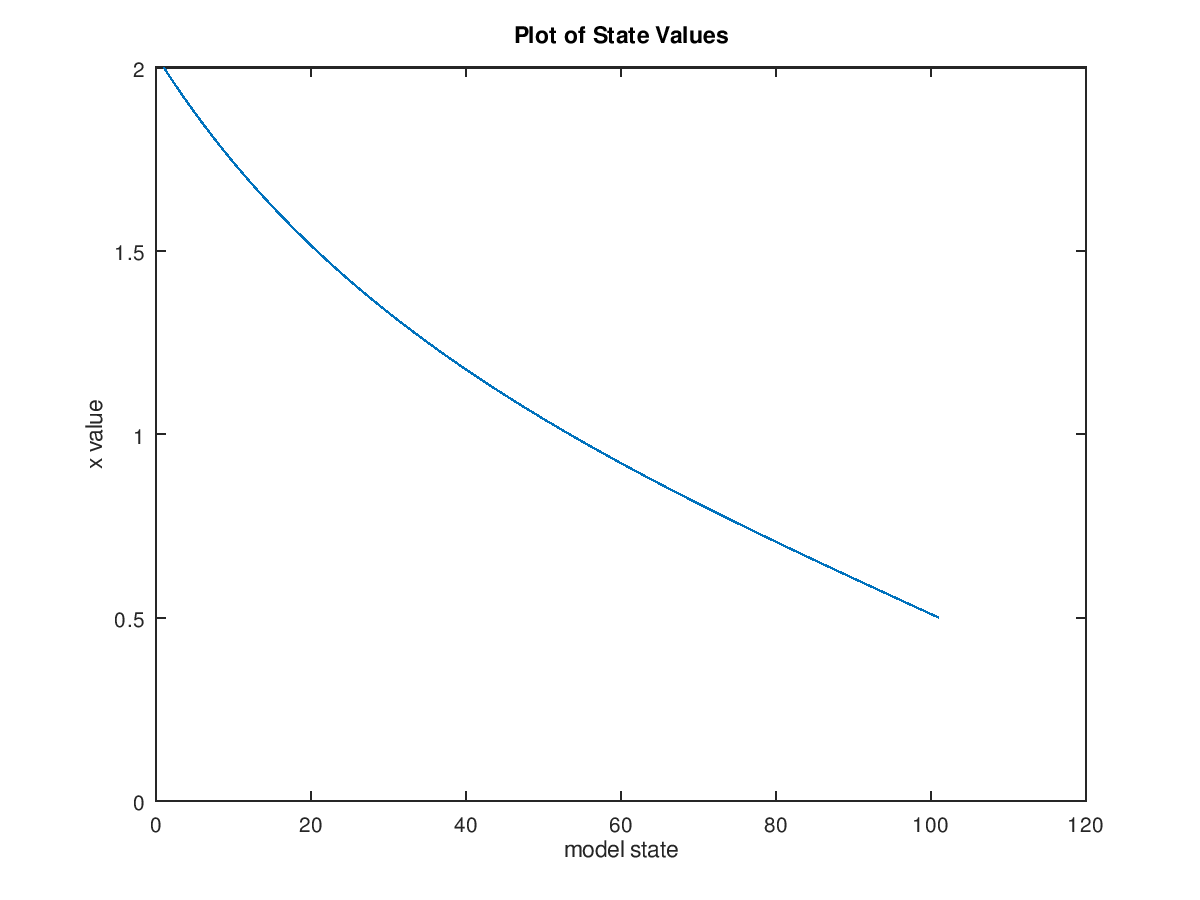
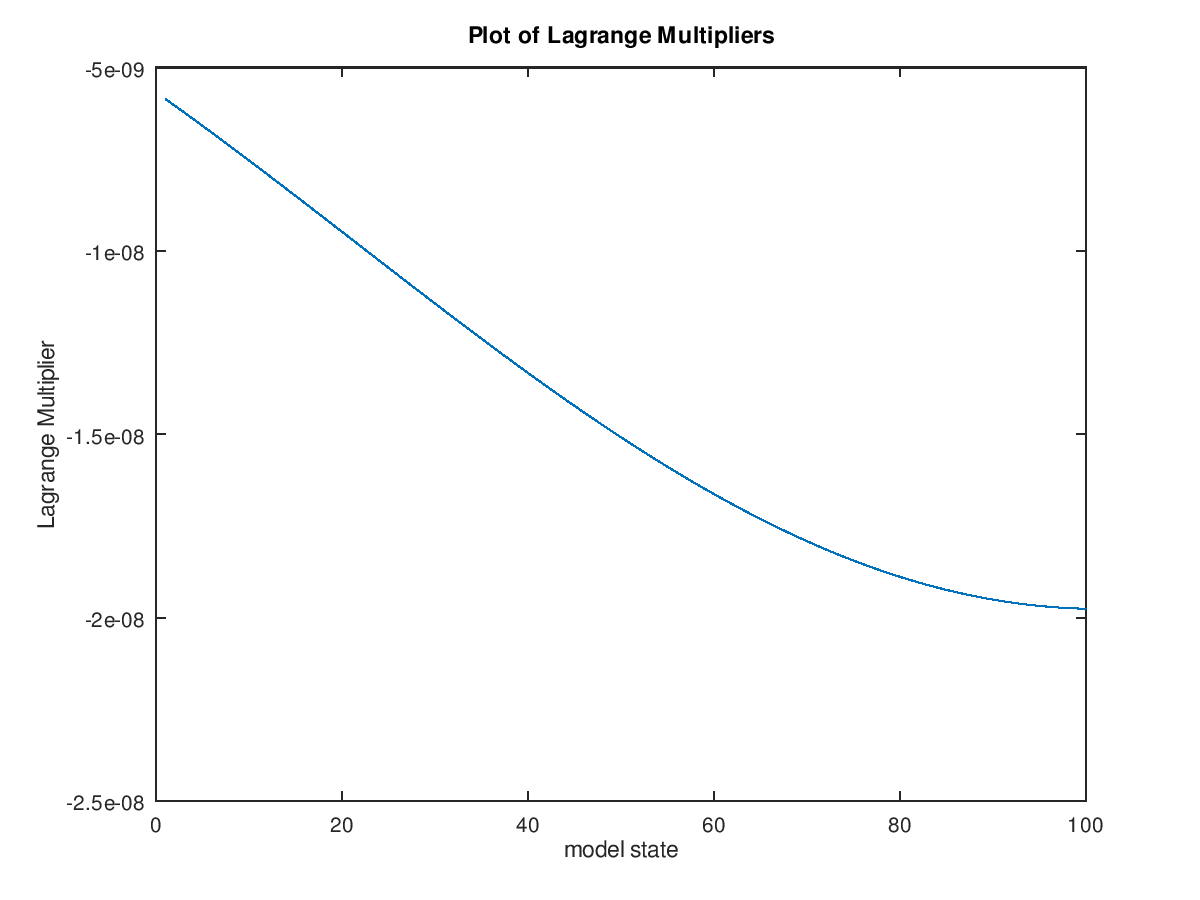
1. ¯*x*0 = 2*,y*¯*m* = 0*.*5*,m* = 100*,h* = 0*.*01

Results:

P\* = -1.2186

f = 1.9518e-16

mu = -0.000000014337

1. ¯*x*0 = 0*.*5*,y*¯*m* = 2*,m* = 100*,h* = 0*.*01

Results:

p = 2.3700

f = 6.8247e-14

mu = -0.00000013333

