ECON 6130

Problem Set 6

Gabe Sekeres

November 26, 2024

Contents

1 Solutions

1. **Shooting Method**

(a & b) I created the function `pset6_residual.m` in Matlab, and evaluated the 1.0% increase in TFP. I got the same answer as Ryan! The digits are 4 and 5 respectively.

(c) I used fsolve, and got convergence in one iteration

(d) I plotted the linearized approximate form and the exact perfect foresight model. I got:

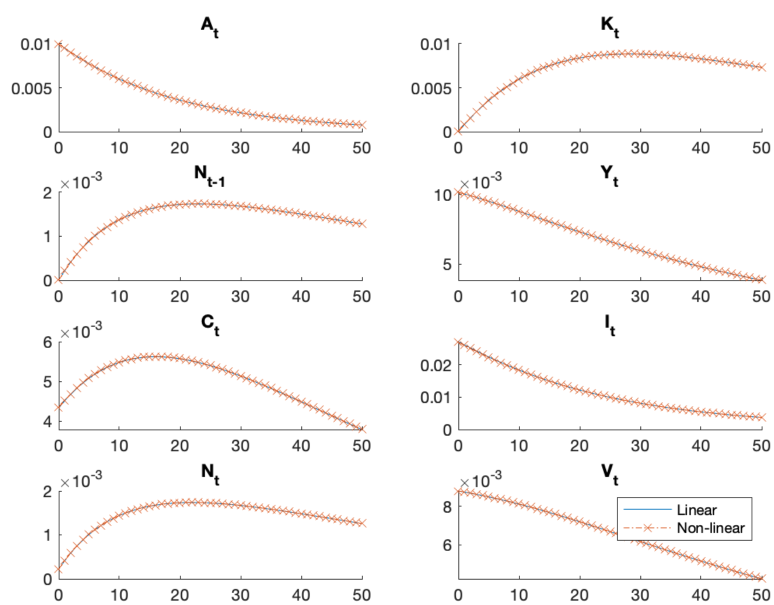

Figure 1: Response to 1% increase in TFP

(e) I did much worse with a 10% change! Convergence took 3 iterations (and significantly longer to run), with an initial residual of 5.1371. Interestingly, the graphs look very similar! There's a difference, but not a big one!

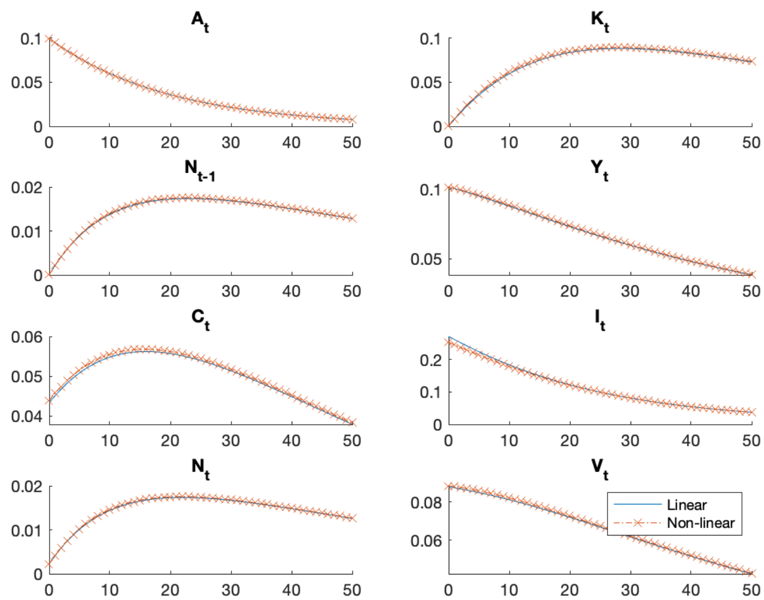

Figure 2: Response to 1% increase in TFP

(f) I decided, against my better judgement, to try this question. I changed one parameter as an exploration, and immediately found such large non-linearities that `fsolve` didn't converge, stopping after 100 iterations and 40,000 functions evaluated. I changed ε to 0.5, reasoning that if the matching function elasticity increased, its derivative (M_V) would have non-linearities in the model equations. I was correct! After playing around a little bit, I found that setting $\varepsilon = 0.3$ actually converged. The initial residual was 368.4282, and it took 7 iterations to converge. Here are the plots:

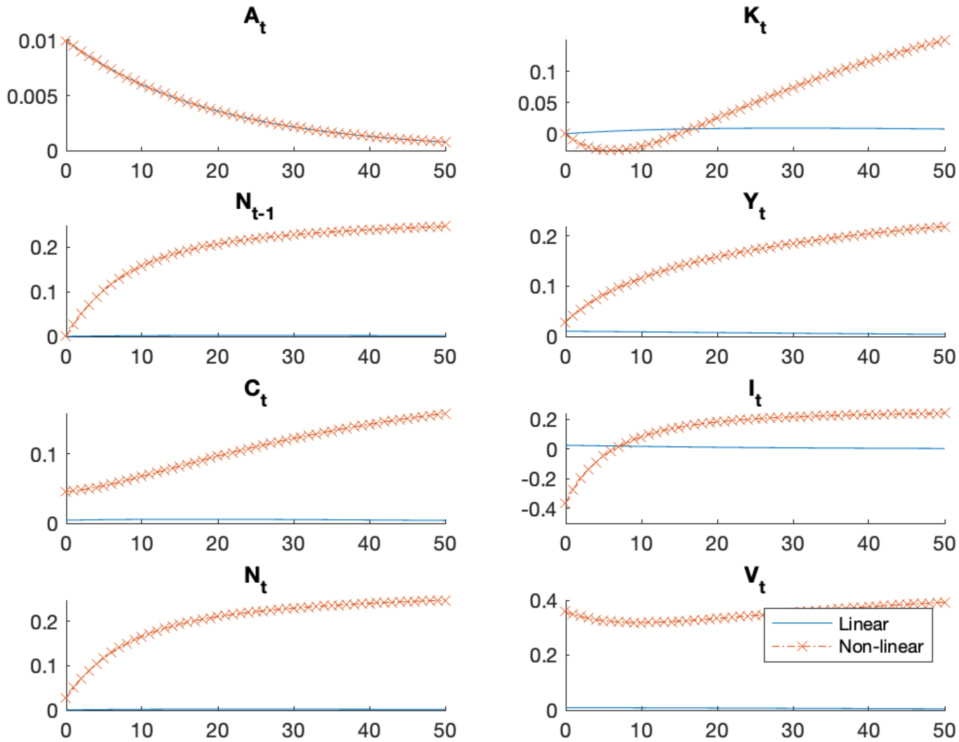

Figure 3: A Large, Nonlinear Response to a 1% shock to TFP

2 Matlab Code

My codebase is the following files:

```
clear;
close all;
tic;

% Set the parameters
param = pset6_parameters;

% Find steady state values
[ss, ~] = pset5_model_ss(param);

% Find linearized policy functions
[fyn, fxn, fypn, fxpn, ~, log_var] = pset6_model(param);
[gx, hx] = pset5_gx_hx(fyn, fxn, fypn, fxpn);
ss(log_var) = log(ss(log_var));

disp('hx')
disp(hx)
disp('gx')
disp(gx)

% Number of periods
N = 500;

% Initialize shock (1% increase in TFP)
eta = zeros(3,3);
eta(1,1) = 0.01;

% Generate impulse responses for initial guess
X = zeros(3, N);
X(:,1) = eta*[1; 0; 0];
for i = 1:N-1
    X(:,i+1) = hx*X(:,i);
end
Y = gx*X;

XY = [X;Y];
var_names = {'A_{t}', 'K_{t}', 'N_{t-1}','Y_{t}', 'C_{t}', 'I_{t}', 'N_{t}','
   V_{t}'};
f = figure;
for ii = 1:8
    s = subplot(4,2,ii); hold on
    p = plot(0:50,XY(ii,1:51));
    title(var_names{ii});
end

% Stack X(t+1) with Y(t) in log deviations
XYv = [X(:,2:end), zeros(3,1); Y(:,1:end)];
```

```matlab
disp('initial residual')
resid0 = pset6_residual(XYv,ss,param, log_var);
disp(num2str(sum(abs(resid0(:)))))

%Solve the equations
options = optimoptions('fsolve');
options.Display = 'iter';
obj = @(x) pset6_residual(x,ss,param,log_var);

XpYshoot = fsolve(obj,XYv,options);
XYshoot = [X(:,1),XpYshoot(1:3,1:end-1); XpYshoot(4:end,:)];

figure(f);
for ii = 1:8
    s = subplot(4,2,ii); hold on
    p = plot(0:50,XYshoot(ii,1:51), '-.x');
end

legend('Linear', 'Non-linear')
saveas(f, '/Users/gabesekeres/Dropbox/Notes/Cornell_Notes/Fall_2024/Macro/
   Matlab/pset6_shoot_01.png')

toc;


 % Find parameterization that makes 1% increase in TFP
 % generate large non-linear effects

param2 = pset6_parameters;
param2.eps = 0.3;

[ss2, ~] = pset5_model_ss(param);

% Find linearized policy functions
[fyn2, fxn2, fypn2, fxpn2, ~, log_var] = pset6_model(param2);
[gx2, hx2] = pset5_gx_hx(fyn2, fxn2, fypn2, fxpn2);
ss2(log_var) = log(ss2(log_var));

disp('hx2')
disp(hx2)
disp('gx2')
disp(gx2)

% Number of periods
N = 500;

% Initialize shock (1% increase in TFP)
eta2 = zeros(3,3);
eta2(1,1) = 0.01;

% Generate impulse responses for initial guess
```

```
X2 = zeros(3, N);
X2(:,1) = eta2*[1; 0; 0];
for i = 1:N-1
    X2(:,i+1) = hx2*X2(:,i);
end
Y2 = gx2*X2;

XY2 = [X2;Y2];
var_names = {'A_{t}', 'K_{t}', 'N_{t-1}','Y_{t}', 'C_{t}', 'I_{t}', 'N_{t}','
   V_{t}'};
f = figure;
for ii = 1:8
    s = subplot(4,2,ii); hold on
    p = plot(0:50,XY2(ii,1:51));
    title(var_names{ii});
end

% Stack X(t+1) with Y(t) in log deviations
XYv2 = [X2(:,2:end), zeros(3,1); Y2(:,1:end)];

disp('initial residual')
resid02 = pset6_residual(XYv2,ss2,param2, log_var);
disp(num2str(sum(abs(resid02(:)))))

%Solve the equations
options = optimoptions('fsolve');
options.Display = 'iter';
obj = @(x) pset6_residual(x,ss2,param2,log_var);

XpYshoot2 = fsolve(obj,XYv2,options);
XYshoot2 = [X2(:,1),XpYshoot2(1:3,1:end-1); XpYshoot2(4:end,:)];

figure(f);
for ii = 1:8
    s = subplot(4,2,ii); hold on
    p = plot(0:50,XYshoot2(ii,1:51), '-.x');
end

legend('Linear', 'Non-linear')
saveas(f, '/Users/gabesekeres/Dropbox/Notes/Cornell_Notes/Fall_2024/Macro/
   Matlab/pset6_shoot_01_large.png')

clear;
close all;
tic;

% Set the parameters
param = pset6_parameters;

param.siga = 0.1;

% Find steady state values
```

```
[ss, ~] = pset5_model_ss(param);

% Find linearized policy functions
[fyn, fxn, fypn, fxpn, ~, log_var] = pset6_model(param);
[gx, hx] = pset5_gx_hx(fyn, fxn, fypn, fxpn);
ss(log_var) = log(ss(log_var));

disp('hx')
disp(hx)
disp('gx')
disp(gx)

% Number of periods
N = 500;

% Initialize shock (1% increase in TFP)
eta = zeros(3,3);
eta(1,1) = 0.1;

% Generate impulse responses for initial guess
X = zeros(3, N);
X(:,1) = eta*[1; 0; 0];
for i = 1:N-1
    X(:,i+1) = hx*X(:,i);
end
Y = gx*X;

XY = [X;Y];
var_names = {'A_{t}', 'K_{t}', 'N_{t-1}','Y_{t}', 'C_{t}', 'I_{t}', 'N_{t}','
   V_{t}'};
f = figure;
for ii = 1:8
    s = subplot(4,2,ii); hold on
    p = plot(0:50,XY(ii,1:51));
    title(var_names{ii});
end

% Stack X(t+1) with Y(t) in log deviations
XYv = [X(:,2:end), zeros(3,1); Y(:,1:end)];

disp('initial residual')
resid0 = pset6_residual(XYv,ss,param, log_var);
disp(num2str(sum(abs(resid0(:)))))

%Solve the equations
options = optimoptions('fsolve');
options.Display = 'iter';
obj = @(x) pset6_residual(x,ss,param,log_var);

XpYshoot = fsolve(obj,XYv,options);
XYshoot = [X(:,1),XpYshoot(1:3,1:end-1); XpYshoot(4:end,:)];
```

```
figure(f);
for ii = 1:8
    s = subplot(4,2,ii); hold on
    p = plot(0:50,XYshoot(ii,1:51), '-.x');
end

legend('Linear', 'Non-linear')
saveas(f, '/Users/gabesekeres/Dropbox/Notes/Cornell_Notes/Fall_2024/Macro/
   Matlab/pset6_shoot_10.png')

toc;

function resid = pset6_residual(XYv, XYss, param, log_var)
    % Declare parameters
    bet = param.bet;
    sig = param.sig;
    alpha = param.alpha;
    deltak = param.deltak;
    deltan = param.deltan;
    phin = param.phin;
    chi = param.chi;
    eps = param.eps;
    rho = param.rho;
    siga = param.siga;

    % Take input argument, add back steady-state
    XYv = reshape(XYv, [8, numel(XYv)/8]) + XYss(:);

    % Put variables back into levels
    XYv(log_var,:) = exp(XYv(log_var,:));
    XYss(log_var) = exp(XYss(log_var));

    % Combine the (fixed) X0 with the guessed paths for X1, X2, ...
    A = [exp(log(XYss(1)) + siga), XYv(1,:)];
    K = [XYss(2), XYv(2,:)];
    N_m = [XYss(3), XYv(3,:)];

    % Combine the guessed paths for Y0, Y1, ... with (fixed) YT
    Yt = [XYv(4,:), XYss(4)];
    C = [XYv(5,:), XYss(5)];
    I = [XYv(6,:), XYss(6)];
    N = [XYv(7,:), XYss(7)];
    V = [XYv(8,:), XYss(8)];

    % Get the t+1 values
    A_p = A(2:end);
    K_p = K(2:end);
    N_m_p = N_m(2:end);
    C_p = C(2:end);
    N_p = N(2:end);
    V_p = V(2:end);
```

```
    % Get the t values
    A = A(1:end-1);
    K = K(1:end-1);
    N_m = N_m(1:end-1);
    Yt = Yt(1:end-1);
    C = C(1:end-1);
    I = I(1:end-1);
    N = N(1:end-1);
    V = V(1:end-1);

    % Model Equations
    resid = zeros(8, 500);  % Should be 500 periods
    resid(1,:) = 1 - bet .* (C_p ./ C).^(-sig) .* (A_p .* alpha .* (K_p ./ N_p
       ).^(alpha - 1) + 1 - deltak);
    resid(2,:) = phin ./ (eps .* chi .* V.^(eps - 1)) - A .* (1 - alpha) .* (K
        ./ N).^alpha - bet .* ((C_p ./ C).^(-sig) .* phin ./ (eps .* chi .*
       V_p.^(eps - 1))) .* (1 - deltan);
    resid(3,:) = Yt - A .* K.^alpha .* N.^(1 - alpha);
    resid(4,:) = Yt - C - I - phin .* V;
    resid(5,:) = K_p - (1 - deltak) .* K - I;
    resid(6,:) = N - (1 - deltan) .* N_m - chi .* V.^eps;
    resid(7,:) = log(A_p) - rho .* log(A);
    resid(8,:) = N_m_p - N;
end
```