Christiano
416, Spring 2008
Homework 1, due Wednesday, April 16.

1. Consider an economy in which household preferences have the following form:

$$\sum_{t=0}^{\infty} \beta^t u(c_t, n_t),\ \beta = .99,\ \text{and } u(c_t, n_t) = \log(c_t) + \psi \log(1 - n_t).$$

The household budget constraint is:

$$c_t + k_{t+1} - (1 - \delta)k_t = r_t k_t + w_t n_t + \pi_t,\ \delta = .028,$$

where r_t, w_t, π_t denote the rental rate on capital, the wage rate, and profits, respectively. Firms operate the following technology:

$$y_t = A_t k_t^{\alpha} n_t^{1-\alpha},\ \alpha = 0.36.$$

Here,

$$A_t = Y_t^{\gamma},$$

where Y_t denotes the economy-wide level of output, in per capita terms. Firms are competitive, and maximize profits, which are zero in equilibrium.

   (a) Derive the household static and dynamic Euler equations. Derive the firm Euler equation.

   (b) In the household Euler equations, substitute out the rental rate on capital and the wage rate for labor, using the firm first order conditions (hint: the firm treats A_t as exogenous). You should have a static and dynamic Euler equation. You should focus on symmetric equilibria, in which the economy-wide average stock of capital and level of employment coincide with the firm's stock of capital and employment. After making this identification, the static and dynamic Euler equations should have the following arguments:

$$\begin{aligned} \text{'static'} &: \quad v_h(n_t, k_t, k_{t+1}) = 0 \\ \text{'dynamic'} &: \quad v_k(k_t, k_{t+1}, k_{t+2}, n_t, n_{t+1}) = 0. \end{aligned}$$

Log-linearize these equations about the steady state values of employment and the firm's stock of capital. In computing the steady states, fix steady state employment at 1/3 and choose the value of ψ that rationalizes that. Initially, fix $\gamma = 0.01$.

Hints on the computation of the steady state: I don't think it is possible to compute the steady state in closed form. A one-dimensional zero-finding algorithm must be used to find the steady state. The algorithm requires a good initial guess. You get a reasonable guess with the steady state associated with $\gamma = 0$, which you can compute in closed form. To compute the steady state for $\gamma > 0$, I suggest you proceed as follows. Set the value of k to the steady state value implied by $\gamma = 0$. Then, compute employment (in closed form) using the steady state intertemporal Euler equation. Then, compute steady state consumption using the steady state resource constraint. Next, evaluate the intratemporal Euler equation for labor. Adjust k until the labor equation is satisfied as a strict equality. The MATLAB routine, fzero, works well for this. You supply it with a function that takes k as input and evaluates v_h above. The routine will quickly adjust k until $v_h = 0$.

(c) Substitute out for labor in the dynamic Euler equation, using the static Euler equation. This will give you one dynamic Euler equation in $\hat{k}_t$, $\hat{k}_{t+1}$, $\hat{k}_{t+2}$. Plot the two roots of this equation against a grid of values, $\tilde{\gamma}$, of γ. Let $\tilde{\gamma}$ be composed of two parts: points 0.001 apart from 0 to 0.42, and points 0.001 apart from 0.473 to 0.57. (That is, $\tilde{\gamma} = [[0 : 0.001 : 0.42]\ [0.473 : 0.001 : 0.57]]$.) Note how both roots are less than unity in the second part of $\tilde{\gamma}$. Note also the interesting behavior of the roots: the one that is less than unity for γ small appears to jump discontinuously from nearly unity in the neighborhood of 0.4722 to nearly minus unity, when it rises monotonically back up towards unity. The root that is greater than unity jumps to plus infinity in the neighborhood of 0.47 and returns back from minus infinity. Finally, note that across the different values of $\tilde{\gamma}$, there are many locally stable solutions, $\hat{k}_{t+1} = a\hat{k}_t$, including cases in which a is negative, and a is very close to unity.

2. Model solving involves a lot of differentiation. Fortunately, MATLAB

can do a lot of this for you. This question will get you started with symbolic algebra in MATLAB.

(a) For an introduction to symbolic algebra in MATLAB, type 'doc' at the MATLAB command prompt. Then, select the 'search' tab, and type the word, symbolic, in the 'search for' window. Then, just read through the various titles in the Symbolic Math Toolbox product. For example, 'Symbolic Objects', 'Arithmetic Operations'. After reading through several of these files, and you start to feel comfortable with symbolic math in MATLAB, select the 'contents' tab and go to 'symbolic math toolbox', and select 'using the symbolic toolbox' and look in particular at 'differentiation' and most specifically, Jacobian. Now, you're in a position to do this homework question, using symbolic algebra in MATLAB.

(b) Consider the following problem:

$$\max_{c,n} u\left(c,n\right), \;\; u\left(c,n\right) = \left[\alpha c^{\frac{1}{\rho}} + \left(1-\alpha\right)\left(1-n\right)^{\frac{1}{\rho}}\right]^{\rho} \tag{1}$$

subject to

$$c \leq wn.$$

Note that the marginal rate of substitution is:

$$\frac{-u_l}{u_c} = \frac{1-\alpha}{\alpha}\left(\frac{1-n}{c}\right)^{\frac{1-\rho}{\rho}} - w = 0, \tag{2}$$

or,

$$\left(\frac{1-\alpha}{a}\right)\left(\frac{1-n}{wn}\right)^{\frac{1-\rho}{\rho}} - w = 0. \tag{3}$$

after substituting out the budget constraint evaluated at equality.

i. type in the formula, (1), into MATLAB, declaring ρ, c, and n to be symbolic variables and α, w to parameters.

syms n c rho

alpha=0.75;

w = 1;

$$u = (alpha * (c)\hat{}(1/rho) + (1 - alpha) * (1 - n)\hat{}(1/rho))\hat{}rho;$$

Now, the object, u, is defined as the above function in which $alpha$ is a constant. Type u at the MATLAB command line and note how it is a function of numbers (the value of α) and the variables, c and n.

ii. Define the vector, z, of endogenous variables

$$z = \left[\begin{array}{cc} c & n \end{array} \right],$$

and obtain the derivative of u with respect to z as follows:

$$up = jacobian\,(u, z)\,.$$

This operation produces the 1×2 vector, up, containing the derivative of u with respect to the first element of z and the second, respectively. You should verify that the formulas in up are correct. Now, define the Euler equation by computing the ratio:

$$eul = -up(2)/up(1) - w,$$

which is a function of the variables n, c, and ρ. Note how eul is also a function of the variables, n, c, and rho. Now, evaluate eul on a grid of values of n, keeping $c = wn$. Do this for $rho = 3$ and 10. Here is some MATLAB code to accomplish the job:

$$\begin{array}{l}
rho = 3; \\
nn = 0 : .001 : .7; \\
for\ ii = 1 : length(nn) \\
n = nn(ii); \\
c = w * n; \\
ff3(ii) = eval(eul); \\
end \\
rho = 20; \\
nn = 0 : .001 : .7; \\
for\ ii = 1 : length(nn) \\
n = nn(ii);
\end{array}$$

$$c = w * n;$$
$$ff10(ii) = eval(eul);$$
$$end$$

Finally, graph the results using

$$plot(nn, ff3, nn, ff10)$$
$$legend(\text{`}\backslash rho = 3\text{'}, \text{`}\backslash rho = 10\text{'})$$
$$axis\ tight$$

Confirm that the Euler equation passes through zero at the value of employment that you can compute analytically for this example.

(c) Repeat question 1 using the symbolic toolbox, exploiting what you learned in the previous parts of this question. Use the canonical form presented in class (see example 2 in the notes on log-linear solutions). Set up the two Euler equations in terms of the vectors, $zp, z0$, $zm1$ (where zp means z_{t+1}, $z0$ means z_t and $zm1$ means z_{t-1}). Use the Jacobian command to differentiate the two equations:

$$\begin{aligned} alpha\_0 &= jacobian(ff, zp) \\ alpha\_1 &= jacobian\,(ff, z0) \\ alpha\_2 &= jacobian(ff, zm1), \end{aligned}$$

where $z_{t+i} = (k_{t+1+i}, n_{t+i})\,,\ i = 1, 0, -1$. Then, combine the above alpha's to obtain the coefficients in the single Euler equation involving capital alone. These coefficients should be symbolic equations in the variables. Then, graph the roots, and verify that all your answers correspond to what you found in question 1.