Problem Set #4

MACS 40000, Dr. Evans
Due Thursday, Oct. 27 at 10:30am

1. **Fitting elliptical disutility of labor to CFE function (2 points).** Assume that an individual's time endowment each period is one $\tilde{l} = 1$. Let the period utility of a household be an additively separable function of consumption and leisure,

$$U(c,l) = u(c) + v(l)$$

where the utility of leisure function $v(l)$ is the constant Frisch elasticity (CFE) disutility of labor functional form.

$$v_{cfe}(l) = -\frac{(1-l)^{1+\frac{1}{\theta}}}{1+\frac{1}{\theta}} = -\frac{(n)^{1+\frac{1}{\theta}}}{1+\frac{1}{\theta}}$$

Assume that an approximation to this disutility of labor function is the following elliptical disutility of labor function.

$$v_{elp}(l) = b\left[1-(1-l)^{\mu}\right]^{\frac{1}{\mu}} = b\left[1-(n)^{\mu}\right]^{\frac{1}{\mu}} \quad \text{for} \quad \tilde{l}, b, \mu > 0$$

   (a) The marginal utility of leisure $\frac{\partial v(l)}{\partial l}$ governs the household decision of how much to work. Give the expression for the marginal utility of leisure for the CFE specification $v_{cfe}(l)$ and for the elliptical specification $v_{elp}(l)$.

   (b) Assume that the Frisch elasticity of labor supply in v_{cfe} is $\theta = 0.8$. Using 1,000 evenly spaced points from the support of leisure between 0.05 and 0.95, estimate the elliptical disutility of labor parameters b and μ that minimize the sum of squared deviations between the two marginal utility of leisure functions $v'_{cfe}(l)$ and $v'_{elp}(l)$ from part (i). Plot the two marginal utility of leisure functions.

2. **Checking feasibility in the steady-state (2 points).** Using the calibration of the S-period-lived agent model described in Section 7.7 of Chapter 7 setting $S = 10$, write a Python function named `feasible()` that has the following form,

```
n_low, n_high, c_cnstr, K_cnstr = feasible(f_params, nvec_guess,
    bvec_guess)
```

where the inputs are a tuple `f_params = (`$\tilde{l}$`, A, alpha, delta)`, and a guess for the steady-state labor supply vector `bvec_guess = np.array([`n_1`, `n_2`,...`n_S`])` and savings vector `bvec_guess = np.array([`b_2`, `b_3`,...`b_S`])`. The outputs should be Boolean (`True` or `False`, `1` or `0`) vectors of lengths S and 1, respectively. `K_cnstr` should be a singleton Boolean that equals `True` if $K \leq 0$ for the given `f_params`, `nvec_guess`, and `bvec_guess`. The object `c_cnstr` should be a length-S Boolean vector in which the sth element equals `True` if $c_s \leq 0$ given `f_params`, `nvec_guess`, and `bvec_guess`. The objects `n_low` and `n_high` are length-S Boolean vectors in which the sth element equals `True` if $n_s \leq 0$ for `n_low` or if $n_s \geq \tilde{l}$ for `n_high`.

(a) Which, if any, of the constraints is violated if you choose an initial guess for steady-state savings of `nvec_guess = 0.95 * np.ones(S)` and `bvec_guess = np.ones(S-1)`?

(b) Which, if any, of the constraints is violated if you choose an initial guess for steady-state savings of `nvec_guess = 0.95 * np.ones(S)` and `bvec_guess = np.append([0.0], np.ones(S-2)`?

(c) Which, if any, of the constraints is violated if you choose an initial guess for steady-state savings of `nvec_guess = 0.95 * np.ones(S)` and `bvec_guess = np.append([0.5], np.ones(S-2)`?

(d) Which, if any, of the constraints is violated if you choose the following initial guess for steady-state labor supply and savings?

```
nvec_guess = 0.5 * np.ones(S)
bvec_guess = \
  np.array([-0.01, 0.1, 0.2, 0.23, 0.25, 0.23, 0.2, 0.1,
      -0.01])
```

(e) What is a principle or a rule that might help you in this problem to choose a good initial guess? That is, what properties should a feasible initial guess have? [Hint: What values of n_s and b_{s+1} will likely keep the consumption constraints satisfied $c_s > 0$ for all periods.]

3. **Solve for the steady-state equilibrium (4 points).** Use the calibration from Section 7.7 and the steady-state equilibrium Definition 7.1. Write a function named `get_SS()` that has the following form,

```
ss_output = get_SS(params, nvec_guess, bvec_guess, SS_graphs)
```

where the inputs are a tuple of the parameters for the model `params = (beta, sigma, l_tilde, chi_n_vec, b_ellip, upsilon, A, alpha, delta, SS_tol, EulDiff)`, an initial guess of the steady-state labor supply `nvec_guess` and savings `bvec_guess`, and a Boolean `SS_graphs` that generates a figure of the steady-state distribution of consumption and savings and a figure of the distribution of labor supply if it is set to `True`.

The output object `ss_output` is a Python dictionary with the steady-state solution values for the following endogenous objects.

```
ss_output = {
  'n_ss': n_ss, 'b_ss': b_ss, 'c_ss': c_ss, 'w_ss': w_ss,
    'r_ss': r_ss, 'K_ss': K_ss, 'L_ss': L_ss, 'Y_ss': Y_ss,
```

```
'C_ss': C_ss, 'n_err_ss': n_err_ss, 'b_err_ss': b_err_ss,
'RCerr_ss': RCerr_ss, 'ss_time': ss_time}
```

Let `ss_time` be the number of seconds it takes to run your steady-state program. You can time your program by importing the time library. And let the objects `n_err_ss` and `b_err_ss` be the respective steady-state vectors of Euler errors for the labor supply and savings decisions given in ratio form $\frac{\beta(1+\bar{r})u_1(\bar{c}_{s+1},\bar{n}_{s+1})}{u_1(\bar{c}_s,\bar{n}_s)} - 1$ or difference form $\beta(1+\bar{r})u_1(\bar{c}_{s+1},\bar{n}_{s+1}) - u_1(\bar{c}_s,\bar{n}_s)$. The object `RCerr_ss` is a resource constraint error which should be close to zero. It is given by $\bar{Y} - \bar{C} - \delta\bar{K}$.

(a) Solve numerically for the steady-state equilibrium values of $\{\bar{n}_s\}_{s=1}^S$, $\{\bar{c}_s\}_{s=1}^S$, $\{\bar{b}_s\}_{s=2}^S$, $\bar{w}$, $\bar{r}$, $\bar{K}$, $\bar{L}$, $\bar{Y}$, $\bar{C}$, the $2S-1$ Euler errors and the resource constraint error. List those values. Time your function. How long did it take to compute the steady-state?

(b) Generate a figure that shows the steady-state distribution of consumption and savings by age $\{\bar{c}_s\}_{s=1}^S$ and $\{\bar{b}_s\}_{s=2}^S$. Generate another figure that shows the steady-state distribution of labor supply by age $\{\bar{n}_s\}_{s=1}^S$.

(c) What happens to the steady-state distributions of labor supply $\{\bar{n}_s\}_{s=1}^S$ and wealth $\{\bar{b}_s\}_{s=2}^S$ and the wage $\bar{w}$ and interest rate $\bar{r}$ if labor becomes more valuable in the production function as evidenced by a decrease in α to $\alpha = 0.25$?

4. **Solve for the non-steady-state equilibrium time path (4 points).** Use time path iteration (TPI) to solve for the non-steady state equilibrium transition path of the economy. Let the initial state of the economy be given by the following distribution of savings,

$$\{b_{s,1}\}_{s=2}^S = \{x(s)\bar{b}_s\}_{s=2}^S \quad \text{where} \quad x(s) = \frac{(1.5 - 0.87)}{78}(s-2) + 0.87$$

where the function of age $x(s)$ is simply a linear function of age s that equals 0.87 for $s = 2$ and equals 1.5 for $s = S = 10$. This gives an initial distribution where there is more inequality than in the steady state. The young have less than their steady-state values and the old have more than their steady-state values. You'll have to choose a guess for T and a time path updating parameter $\xi \in (0,1)$, but I can assure you that $T < 50$. Use an L^2 norm for your distance measure (sum of squared percent deviations), and use a convergence parameter of $\varepsilon = 10^{-9}$. Use a linear initial guess for the time path of the aggregate capital stock from the initial state K_1^1 to the steady state K_T^1 at time T.

(a) Plot the equilibrium time paths of the aggregate capital stock $\{K_t\}_{t=1}^{T+5}$, wage $\{w_t\}_{t=1}^{T+5}$, and interest rate $\{r_t\}_{t=1}^{T+5}$.

(b) Also plot the equilibrium time path for labor supply $\{n_{3,t}\}$ and savings $\{b_{3,t}\}$ of every person age $s = 3$ in every period.