Unemployment Insurance and Capital Accumulation: Computational Appendix

Eric R. Young*
Department of Economics
University of Virginia

July 12, 2006

Abstract

This paper discusses in detail the computational procedure used in Young (2004).

1. Computational Details

This appendix details the computation of the equilibrium in Young (2004). The equilibrium for this economy is a stationary distribution over the states of the world, denoted $\Gamma(k, e)$, which satisfies the conditions given in Definition 1. Computing this object involves obtaining 3 value functions, $v(k, 0)$, $v(k, 1)$, and $v(k, 2)$, in the baseline case. These value functions are computed as follows. The details for other versions of the model are similar; when important differences arise, I specifically address them.

I first choose a grid in the k direction. As the value function has much more curvature near the borrowing constraint, the grid is not chosen in a uniform fashion; rather, the number of grid points is much larger for values close to the borrowing constraint. For the current model, I chose 150 grid points with 40 lying between 0 and 1 and the majority of the rest between 1 and 25. I chose an upper bound for the grid of $k = 1250$; the number of agents in the baseline case who have this level of assets is zero to 10 decimal places. However, households are free to choose levels of capital above this point – they simply choose not to do so. For each point on the grid $K \times \{0, 1, 2\}$

* Department of Economics, University of Virginia, Charlottesville, VA 22904, ph: 434-924-3811, fax: 850-924-9208, ey2d@virginia.edu.

I solve the consumer's problem in the following fashion. My results are not sensitive to increases in the number of grid points.

I first partition the Bellman equation as follows. Let $\varepsilon = 1$; that is, the consumer has been offered an employment opportunity this period. Next period's wealth must then solve the problem

$$\widetilde{v}_1(k) = \max_{k_1'} \left\{ \frac{1}{1-\sigma} [(r+1-\delta)k + (1-\tau)wh - k_1']^{1-\sigma} + \beta v(k_1', 0) \right\} \tag{B1}$$

subject to the borrowing constraint. Note that employed agents' savings decisions do not depend on their past employment status. Now let $\varepsilon = 0$; no employment opportunity has been offered. Next period's wealth must now solve the problem

$$\widetilde{v}_0(k, e) = \max_{k_0'} \left\{ \frac{1}{1-\sigma} [(r+1-\delta)k + B_e wh - k_0']^{1-\sigma} + \beta v(k_0', \min\{e+1, \hat{\jmath}\}) \right\} \tag{B2}$$

subject to the borrowing constraint. Finally, I solve for the optimal search effort by solving

$$v(k, e) = \max_{a} \{ [1 - \exp(-\gamma_e a)] \widetilde{v}_1(k) + \exp(-\gamma_e a) \widetilde{v}_0(k, e) - a^{\chi} \} \tag{B3}$$

subject to $a \geq 0$. Partitioning is useful because solving multivariate constrained optimization problems is much more difficult than solving one nonlinear equation. This three-part procedure defines a recursive algorithm which can be iterated upon, from an initial guess, to obtain the value function as the fixed point of the Bellman operator. All conditions needed to ensure the existence of this object are satisfied as well as the conditions that the value function be continuous, differentiable, and concave. To increase speed, I implement a form of Howard's improvement algorithm, in which computed policy functions are used repeatedly to update value functions without reoptimizing at each step. This approach can be shown to be a contraction operator as well, with a modulus of contraction strictly smaller than β.[1]

I solve the above problems with a two-part method. For values near the borrowing constraint – where it might bind – I use a bisection method to find the optimal level of next period's capital: I zero the first-order condition with the added constraint that k' cannot fall below k_b. For values of k that are not likely to lead to binding borrowing constraints, I use a fast Newton-Raphson routine

[1] The program contains a variable **hfix** that rescales aggregate output; it has no effect on the results but is important because it interacts with the grid and determines the range over which the value function has significant curvature. It is set to 0.3271 which puts output close to 1.

to maximize the function.[2] Both of these methods require that the derivative be continuous and the Newton-Raphson routine requires a continuous second derivative as well. For these reasons I use cubic spline interpolation for evaluating the value function at points off the grid. This method has advantages over completely grid-bound methods; namely, it does not introduce discreteness and is much faster as it requires fewer grid points for a given level of accuracy. It also has a speed advantage over methods based on approximating the value function by polynomials – solving for the coefficients in the cubic spline involves nothing more than inverting a tridiagonal matrix, rather than some minimization routine. The solution for a is located using a Newton-Raphson routine after checking that the condition $\widetilde{v}_1 > \widetilde{v}_0$ holds; if that condition does not hold, then $a = 0$.

As a robustness check, I also compute the Euler equation errors for arbitrary points off the capital grid (for the regions where the borrowing constraint does not bind). The Euler equation associated with an employed agent's savings decision is given by

$$\frac{-1}{c_1(k)^{\sigma}} + \beta \left\{ \left[1 - \exp\left(-\gamma_0 a\left(k_1'\right)\right)\right] \frac{r+1-\delta}{c_1\left(k_1'\right)^{\sigma}} + \exp\left(-\gamma_0 a\left(k_1', 0\right)\right) \frac{r+1-\delta}{c_0\left(k_1', 0\right)^{\sigma}} \right\} \leq 0 \qquad \text{(B4)}$$

where primes denote next period values. Similarly, the unemployed agent has an Euler equation of the form

$$\frac{-1}{c_0(k,e)^{\sigma}} + \beta \left\{ \left[1 - \exp\left(-\gamma_{e^*} a\left(k_0', e^*\right)\right)\right] \frac{r+1-\delta}{c_1\left(k_0'\right)^{\sigma}} + \exp\left(-\gamma_{e^*} a\left(k_0', e^*\right)\right) \frac{r+1-\delta}{c_0\left(k_0', e^*\right)^{\sigma}} \right\} \leq 0 \quad \text{(B5)}$$

where $e^* = \min\{e+1, \hat{\jmath}\}$. Both inequalities are equalities if $k' > k_b$. These errors are never larger than 10^{-5} even for the regions of the state space where the grid points are extremely sparse (of course, I only compute these for values at which the Euler equations hold with equality). Values for each of the decision rules are computed using cubic splines.

Having solved the consumer's problem, I am now left with computing prices. I begin by guessing values for r and τ; w can be obtained from the firm's first-order conditions. Next I solve the consumer's problem as detailed above. Then, having obtained the value function, I resolve it on a much finer uniform grid; this grid has 1050 evenly spaced points in the k direction. A uniform grid is required here, so the number of grid points must be substantially increased. Using this new grid, I iterate on the consumers' decision rules until a stationary distribution emerges. This iteration proceeds as follows.

[2] It is in general not safe to use Newton-Raphson close to the constraint because extrapolation of cubic splines is not recommended and the routine is likely to attempt negative values when the solution is close to zero.

Take a point (k,e) in the state space. Locate $k_1'(k)$ and $k_0'(k,e)$ for this point using a search routine – an efficient one can be found in Press *et.al* (1993). Also compute $\Pr(\varepsilon=1|e)$. Then allocate the mass located at (k,e) to the two grid points bracketing each of the four new asset holdings as follows. Let ω_1 be the weight attached to the lower grid point, denoted k_{1L}, for an agent who obtains a job. Then, I have

$$\begin{aligned}\Gamma^{n+1}(k_{1L},0) &= \Pr(\varepsilon=1|e)*\omega_1*\Gamma^n(k,e)\\ \Gamma^{n+1}(k_{1H},0) &= \Pr(\varepsilon=1|e)*(1-\omega_1)*\Gamma^n(k,e).\end{aligned}$$

where Γ^n is the distribution obtained in the nth iteration of this procedure and

$$\omega_1 = 1-\frac{k_1'-k_{1L}}{k_{1H}-k_{1L}}.$$

Similarly, I have

$$\begin{aligned}\Gamma^{n+1}(k_{0eL,}e^*) &= [1-\Pr(\varepsilon=1|e)]*\omega_{0e}*\Gamma^n(k,e)\\ \Gamma^{n+1}(k_{0eH},e^*) &= [1-\Pr(\varepsilon=1|e)]*(1-\omega_{0e})*\Gamma^n(k,e).\end{aligned}$$

I terminate the loop when the changes in the distribution are small. This method of computing the cross-sectional distribution of wealth is better than the simulation methods typically used in the literature because it does not involve sampling error. A uniform grid is required; otherwise, there will be a discrete upward jump at any point where the distance between grid points increases.

As a robustness check, I solved the model by simulation; the results are not sensitive here except that the budget balancing tax rate is very sensitive to the simulation length (it depends on the fractions of agents in each point in the employment distribution, and these values converge very slowly to their unconditional means because they cannot contain more significant digits than digits in the simulation length). This requires a relatively long simulation length. The other advantage for my method is in the calibration procedure – it ensures that small changes in parameter values cause small changes in endogenous variables; with simulation, many changes in the parameters – especially the γ_e's – will not change the unemployment rates due to the finiteness of the simulation length. Such a result makes solving the calibration equations impossible. Finally, I have directly imposed the law of large numbers – this is important in obtaining the correct equilibrium unemployment rate.

Once the stationary distribution is obtained, I can use the firm's first-order condition to obtain an implied value for r and the government budget constraint to find an implied value for τ. I then iterate on these values until they converge. In practice, I use a nested version of Brent's method to ensure a well-behaved convergence – the problem is extremely sensitive to small changes in r and it is convenient to have a solution method which brackets the zero for r. It can be proven that r must satisfy

$$r \leq \frac{1}{\beta} - 1 + \delta \tag{B6}$$

which provides an upper bound on r. A lower bound is 0, although this is obviously not an efficient choice. The algorithm guesses a value for τ, then computes the equilibrium value for r. For this value of r the government budget surplus is computed. τ is then iterated on, solving at each step for the equilibrium interest rate, until the budget is balanced. For the calibration procedure, I use Brent's method to find equilibrium r for given values of $\{\tau, \gamma_0, \gamma_1, \gamma_2\}$ then update that vector using a Newton-Raphson multidimensional routine. Since Laffer curve considerations are important here I make sure to select the lowest tax rate consistent with a particular steady state – there will also exist a high tax rate steady state where aggregate activity is smaller.

The computation of the transition path involves backward induction along the path. I start by computing the steady state equilibria for both ends of the transition. I then choose the length of the transition, T and assume that the economy is in the new steady state in period $T+1$. The value function of households in period $T+1$ is then given by the value function in the new steady state. Next I choose a sequence of interest rates and tax rates $\{r_t, \tau_t\}_{t=0}^{T}$ that end with the values in the new steady state (because the unemployment rate can adjust immediately, the values in the first period will not be those of the old steady state). Given this sequence, I iterate backward on the Bellman equation, using the value function obtained in the previous step as the value of saving in the current period. I continue until I reach period 0. At this point, I iterate forward on the initial distribution of wealth and employment status using the decision rules (which are dependent on t) until the end of the transition, computing the market clearing values for r and τ at every step. If the new values obtained are close to the ones I took as given, I stop. If not, I update by adding 10 percent of the difference to each and repeat. Once the sequences have converged, I check to see that the length T was adequate; if there are significant jumps from the final period to the new steady state (for example, larger than any other change in the interest rate), I increase T and repeat the process.

In practice, the transition must be handled slightly differently due to the extreme sensitivity of the model to changes in r. As a result, I modify the above algorithm slightly. I choose a sequence of interest and tax rates that actually converge to their new values in $T - S$ periods. I then iterate backward as before, but never update the prices in the last S periods; they remain constant. In addition, during the simulated transition I use the stationary decision rules in periods $S + 1 - T$. Once the sequences have converged, I let S go slowly to zero, recomputing the entire transition at each step. This algorithm controls the sensitivity of the model and converges to the true transition path.

References

[1] Young, Eric R. (2004), "Unemployment Insurance and Capital Accumulation," *Journal of Monetary Economics* **51(8)**, pp. 1683-710.