EXERCISE AND READINGS FROM 4/29 LECTURE

The exercise, due Tuesday 5/4:
Consider the simplest neoclassical optimal growth model, written as a deterministic continuous time problem:

\[
\max_{C,K} E \left[ \int_0^\infty \frac{C_t^{1-\gamma}}{1-\gamma} e^{-\beta t} dt \right] \quad \text{subject to} \quad \dot{K} + C = AK^a - \delta K. (1)
\]

Solve for and plot the time paths of \(C\) and \(K\) starting from an initial condition with \(K = \frac{1}{2} \bar{K}\), where \(\bar{K}\) is the steady-state value of \(K\). Assume \(A = 1\), \(\delta - \beta = .05\), \(a = .3\), \(\gamma = 2\).

Compare this optimal time path for \(y = (K,C)\) with what would emerge from the Solow growth model:

\[
\dot{K} = sAK^a - \delta K \quad \text{(3)}
\]
\[
C = (1 - s)AK^a \quad \text{(4)}
\]

with \(A\), \(\delta\) and \(a\) as above \(s = .15\).

Additional part not mentioned in class: Linearize the optimal growth model around the steady state, find its impulse response to a disturbance in log \(A\), and compare the half-life of deviation from steady state implied by this linearization to the half-life you found in the full non-linear solution. (The half-life is the time from the initial disturbance until the time when the deviation from steady state has been reduced by a factor of .5.)

Remarks: You will want to use Matlab, Octave, or R for this. They all have functions that are wrappers for widely used underlying fortran differential equation solvers. The two-equation system formed from the FOC and the constraint is “stiff” in the terminology of differential equations, so it is hard to solve accurately. Two possible approaches:

(a) Multiple shooting. Guess an initial value for \(C\), solve the equation forward over some span (say for \(t\) in \((0,100)\) or \((0,60)\)), and adjust the initial \(C\) up or down until the solution converges to the steady state over your time interval (without exploding upward or downward). I solved the problem this way in half an hour or so (counting time to code the functions). I used R’s \texttt{vode()}\), which is a wrapper for a fortran program of the same name. Matlab and Octave may have wrappers for the same program. This is an initial-condition solver that works for stiff equation systems. I gave it as a grid of
time points to solve at, $\exp((0.471) \times 0.01)-1$. The idea is that it needs points more closely spaced together near the beginning, where change is rapid. I found that to get convergence to steady state over a full 100-year path I needed to have $C_0$ accurate to around 9 significant figures. I gave the program an analytic Jacobian as well as the analytic form for $\dot{y}$ as a function of $y$.

(b) Use a boundary value, rather than an initial value, solver. In this case, you would just give the program $0.5\bar{K}$ as the initial value of $K$ and $\bar{K}$ as the final value at, say, $t = 100$. This should give an accurate solution, but the boundary value solver I tried could not handle this equation system, at least the way I set it up. I did not try the analytic Jacobian with the boundary value solver, however, and there are a number of different boundary value solvers you could try.

The Solow model is univariate and should solve easily.

References for 5/4 lecture:
Burnside, Eichenbaum, Kleshchelski, and Rebelo (2008)
Weitzman (2007)
Barro (2006)

References