Econ3070: Problem Set 4

Install Python! Then let the computation begin. Let me know if you run into any issues.

The due date is March 25th. Please submit your results electronically. This includes both plots and descriptions in an organized document (such as a PDF or a Jupyter notebook) and the underlying code.

1 Ramsey Policies

Consider the standard Ramsey model presented in lecture with population growth, no technological growth ($z = 1$), and log utility. Use the parameter values

$$\rho = 0.05$$
$$n = 0.02$$
$$\alpha = 0.35$$
$$\delta = 0.1$$

For each subpart, compute the time paths of k and c, and plot your results on a phase diagram and as functions of time. Be sure to make use of the "reverse time" trick to improve stability.

(a) Suppose we unexpectedly institute a tax on capital gains $\tau_k = 0.3$, as described in lecture 3.

(b) Now consider the above scenario but where the tax is pre-announced one year in advance.

(c) Suppose instead we have a consumption tax $\tau_c = 0.3$, as in the midterm. We saw that there is no effect when unexpected. So consider the case where it pre-announced one year in advance.

(d) Finally, as in the midterm, consider the case where $\tau_c(t) = \tau_c [1 - \exp(-\beta t)]$ emerges slowly over time, where we use $\tau_c = 0.3$ and $\beta = 1$.

2 Value Function Iteration

Consider a discrete time Ramsey problem where the value function satisfies the following Bellman equation

$$v(k) = \max_{k'} \left\{ u(f(k) + (1-\delta)k - k')) + \beta v(k') \right\}$$

where $f(k) = k^\alpha$ and $u(c) = \log(c)$. Use the parameter values

$$\beta = 0.95$$
$$\alpha = 0.35$$
$$\delta = 0.1$$

(a) Implement a grid solver in pure `numpy` and plot the resulting value function $v(k)$ and the policy function $k'(k)$. Additionally report the log absolute deviation of the Bellman equation above as a function of k.

(b) Implement a solver using `jax` and compare to the previous part in terms of speed and accuracy for a variety of grid sizes and convergence criterion.

(c) Now implement an interpolating solver, where the choice of k' can be continuous. How does this influence the speed-accuracy tradeoff?