Advanced Tools in Macroeconomics

Continuous time models (and methods)

Pontus Rendahl

May 10, 2018

Introduction

- In this lecture we will take a look at models in continuous, as opposed to discrete, time.
- There are some advantages and disadvantages
  - Advantages: Can give closed form solutions even when they do not exist for the discrete time counterpart. Can be very fast to solve. Trendier than sourdough bread, fixed gear bicycles, and skinny jeans combined (so if you do continuous time you need neither).
  - Disadvantages: Intuition is a bit tricky. Contraction mapping theorems / convergence results go out the window (but can be somewhat brought back). Difficult to deal with certain end-conditions (like finite lives etc.). Hard to deal with continuous processes (like AR(1)).

Plan for today

- Continuous time methods and models are not as well documented as the discrete time cases.
- Proceed through examples
  1. The Solow growth model (!)
  2. The (stochastic) Ramsey growth model
- How to solve (turns out to be pretty easy, and we can apply methods we know from earlier parts of the course)

The Solow growth model

- The Solow growth model is characterized by the following equations

$$
\begin{aligned}
Y_t &= K_t^{\alpha}(A_t N_t)^{1-\alpha} \\
K_{t+1} &= I_t + (1-\delta)K_t \\
S_t &= sY_t \\
I_t &= S_t \\
A_{t+1} &= (1+g)A_t \\
N_{t+1} &= (1+\eta)N_t
\end{aligned}
$$

- To solve this model we rewrite it in intensive form

$$
x_t = \frac{X_t}{A_t N_t}, \quad \text{for } X = \{Y, K, S, I\}
$$

The Solow growth model

- Using this and substituting in gives

$$\frac{K_{t+1}}{A_t N_t} = s k_t^{\alpha} + (1-\delta) k_t$$

- We can rewrite as

$$\begin{aligned} \frac{K_{t+1}}{A_{t+1} N_{t+1}} \frac{A_{t+1} N_{t+1}}{A_t N_t} &= s k_t^{\alpha} + (1-\delta) k_t \\ k_{t+1} \frac{A_{t+1} N_{t+1}}{A_t N_t} &= s k_t^{\alpha} + (1-\delta) k_t \\ k_{t+1}(1+g)(1+\eta) &= s k_t^{\alpha} + (1-\delta) k_t \end{aligned}$$

The Solow growth model

- Ta-daa!

$$k_{t+1} = \frac{sk_t^{\alpha}}{(1+g)(1+\eta)} + \frac{(1-\delta)k_t}{(1+g)(1+\eta)}$$

- Balanced growth: $k_{t+1} = k_t = k$

$$k = \left(\frac{g+\eta+g\eta+\delta}{s}\right)^{\frac{1}{\alpha-1}}$$

- This is not textbook stuff. Why? Discrete time. More elegant solution in continuous time.

The Solow growth model

- Continuous time is not a state in itself, but is the effect of a limit. A derivative is a limit, an integral is a limit, the sum to infinity is a limit, and so on.
- Continuous time is the name we use for the behavior of an economy as intervals between time periods approaches zero.

The Solow growth model

- The right approach is therefore to derive this behavior as a limit (much like you probably derived derivatives from its limit definition in high school).
- Eventually you may get so well versed in the limit behavior that you can set it up directly (like you can say that the derivative of $\ln x$ is equal to $1/x$, without calculating $\lim_{\varepsilon \to 0}(\ln(x+\varepsilon) - \ln(x))/\varepsilon$)
- I'm not there yet. I have to do this the complicated way. People like Ben Moll at Princeton is. Take a look at his lecture notes on continuous time stuff. They are great.

The Solow growth model

- Back to the model.
- Suppose that before the length of each time period was one month. Now we want to rewrite the model on a biweekly frequency.
- It seems reasonable to assume that in two weeks we produce half as much as we do in one month: $Y_t = 0.5K_t^{\alpha}(A_t N_t)^{1-\alpha}$.
- It also seems reasonable that capital depreciates slower, i.e. 0.5δ.

The Solow growth model

- Notice that we still have N_t worker and K_t units of capital: Stocks are not affected by the length of time intervals (although the accumulation of them will).
- The propensity to save is the same, but with half of the income saving is halved too (and therefore investment)
- What happens to the exogenous processes for A_t and N_t?

The Solow growth model

- Before

$$A_{t+1} = (1+g)A_t, \quad N_{t+1} = (1+\eta)N_t$$

- Now

$$A_{t+0.5} = (1+0.5g)A_t, \quad N_{t+0.5} = (1+0.5\eta)N_t$$

or

$$A_{t+0.5} = e^{0.5g}A_t, \quad N_{t+0.5} = e^{0.5\eta}N_t \quad ?$$

The Solow growth model

- It turns out that this choice does not matter much for our purpose
- Suppose that the time period is not one month but $\Delta \times$ one month. And suppose that

$$A_{t+\Delta} = (1 + \Delta g) A_t$$

- Rearrange

$$\frac{A_{t+\Delta} - A_t}{\Delta} = g A_t.$$

- And take limit $\Delta \to 0$

$$\dot{A}_t = g A_t$$

The Solow growth model

- Suppose that the time period is not one month but $\Delta \times$ one month. And suppose that

$$A_{t+\Delta} = e^{\Delta g} A_t$$

- Rearrange

$$\frac{A_{t+\Delta} - A_t}{\Delta} = \frac{(e^{\Delta g} - 1)}{\Delta} A_t.$$

- Notice

$$\lim_{\Delta \to 0} \frac{(e^{\Delta g} - 1)}{\Delta} = \lim_{\Delta \to 0} \frac{(g e^{\Delta g})}{1} = g$$

The Solow growth model

- So

$$\lim_{\Delta \to 0} \frac{(e^{\Delta g} - 1)}{\Delta} A_t = g A_t$$

  and thus

$$\dot{A}_t = g A_t$$

- Therefore, it doesn't matter if $A_{t+\Delta} = (1 + \Delta g) A_t$ or $A_{t+\Delta} = e^{\Delta g} A_t$. The limits are the same.

The Solow growth model

- Solow growth model in Δ units of time

$$
\begin{aligned}
Y_t &= \Delta K_t^{\alpha}(A_t N_t)^{1-\alpha} \\
K_{t+\Delta} &= I_t + (1-\Delta\delta)K_t \\
S_t &= sY_t \\
I_t &= S_t \\
A_{t+\Delta} &= (1+\Delta g)A_t \\
N_{t+\Delta} &= (1+\Delta\eta)N_t
\end{aligned}
$$

The Solow growth model

- Substitute and rearrange as before

$$\begin{aligned}
\frac{K_{t+\Delta}}{A_{t+\Delta}N_{t+\Delta}}\frac{A_{t+\Delta}N_{t+\Delta}}{A_tN_t} &= s\Delta k_t^\alpha + (1-\Delta\delta)k_t \\
k_{t+\Delta}\frac{A_{t+\Delta}N_{t+\Delta}}{A_tN_t} &= s\Delta k_t^\alpha + (1-\Delta\delta)k_t \\
k_{t+\Delta}(1+\Delta g)(1+\Delta\eta) &= s\Delta k_t^\alpha + (1-\Delta\delta)k_t
\end{aligned}$$

The Solow growth model

- Simplify and rearrange

$$
\begin{aligned}
& k_{t+\Delta}(1+\Delta g)(1+\Delta\eta) = s\Delta k_t^{\alpha} + (1-\Delta\delta)k_t \\
\Rightarrow \quad & k_{t+\Delta} - k_t = s\Delta k_t^{\alpha} - \Delta\delta k_t - \Delta(g+\eta+\Delta g\eta)k_{t+\Delta} \\
\Rightarrow \quad & \frac{k_{t+\Delta} - k_t}{\Delta} = s k_t^{\alpha} - \delta k_t - (g+\eta+\Delta g\eta)k_{t+\Delta}
\end{aligned}
$$

- Take limits $\Delta \to 0$

$$
\dot{k}_t = s k_t^{\alpha} - (g+\eta+\delta)k_t
$$

- With steady state

$$
k = \left(\frac{g+\eta+\delta}{s}\right)^{\frac{1}{\alpha-1}}
$$

The Solow growth model: Solution

- How do you solve this model?
- The equation

$$\dot{k}_t = sk_t^{\alpha} - (g + \eta + \delta)k_t$$

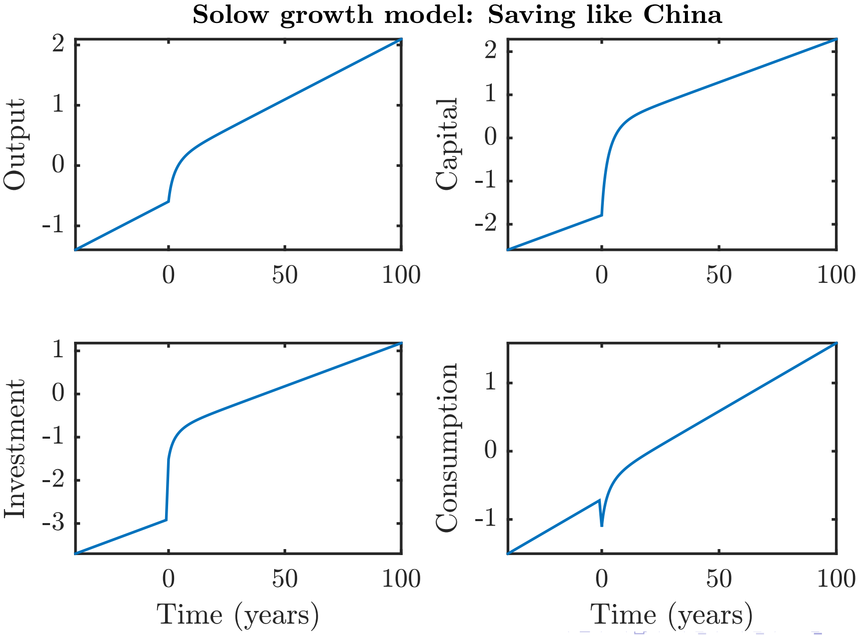
  is an ODE.
- Declare it as a function with respect to time, t, and capital, k, in Matlab as

$$\mathtt{solow} = \mathtt{@(t,k)} \quad \mathtt{sk}^{\alpha} - (\mathtt{g} + \eta + \delta)\mathtt{k};$$

- The simulate it for, say 100 units of time, with initial condition k_0 as

$$[\mathtt{time}, \mathtt{capital}] = \mathtt{ode45}(\mathtt{solow}, [0 \;\; 100], \mathtt{k}_0);$$

The Solow growth model: Solution



The Solow growth model: Solution

A few pointers

- Once you got the solution of a deterministic continuous time model, the solution will always be of the form $\dot{x}_t = f(x_t)$, whether or not x_t is a vector.
- The matlab function ode45 (or other versions) can then simulate a transition (such as an impulse response).
- You could also simulate on your own through the approximation

$$\dot{x}_t \approx \frac{x_{t+\Delta} - x_t}{\Delta}$$

  and thus find your solution as $x_{t+\Delta} = x_t + \Delta f(x_t)$.
- For this to be accurate, Δ must be small if there are a lot of nonlinearities.
- The ODE function in matlab uses so-called Runge Kutta methods to vary the step-size Δ in an optimal way.

The Ramsey growth model

- Now consider the Ramsey growth model (without growth)

$$
\begin{aligned}
v(k_t) = \max_{c_t, k_{t+1}} & \{u(c_t) + (1-\rho)v(k_{t+1})\} \\
\text{s.t.} \quad & c_t + k_{t+1} = k_t^\alpha + (1-\delta)k_t
\end{aligned}
$$

- In Δ units of time

$$
\begin{aligned}
v(k_t) = \max_{c_t, k_{t+\Delta}} & \{\Delta u(c_t) + (1-\Delta\rho)v(k_{t+\Delta})\} \\
\text{s.t.} \quad & \Delta c_t + k_{t+\Delta} = \Delta k_t^\alpha + (1-\Delta\delta)k_t
\end{aligned}
$$

The Ramsey growth model

- Notice that all flows change when the length of the time period on which they are defined changes. Stocks, k, are the same.
- I discount the future with $1 - \Delta\rho$ instead of $(1 - \rho)$ (or with $e^{-\Delta\rho}$ instead of e^{ρ}, but these are, in the limit, equivalent).
- One funny thing: Consumption, c, is still "monthly" consumption, but it now only cost Δ as much, and I only get a Δ fraction of the utility!
- These assumptions are for technical reasons, and it will (hopefully) soon be clear why they are made.

The Ramsey growth model

- Bellman equation

$$v(k_t) = \max_{c_t, k_{t+\Delta}} \{\Delta u(c_t) + (1 - \Delta\rho) v(k_{t+\Delta})\}$$
$$\text{s.t.} \quad \Delta c_t + k_{t+\Delta} = \Delta k_t^{\alpha} + (1 - \Delta\delta) k_t$$

- Subtract $v(k_t)$ from both sides and insert the budget constraint into $v(k_{t+\Delta})$

$$\begin{aligned} 0 = \max_{c_t} \{ & \Delta u(c_t) + v(k_t + \Delta(k_t^{\alpha} - \delta k_t - c_t)) - v(k_t) \\ & - \Delta\rho v(k_t + \Delta(k_t^{\alpha} - \delta k_t - c_t))\} \end{aligned}$$

The Ramsey growth model

- From before

$$0 = \max_{c_t}\{\Delta u(c_t) + v(k_t + \Delta(k_t^\alpha - \delta k_t - c_t)) - v(k_t) \\ - \Delta\rho v(k_t + \Delta(k_t^\alpha - \delta k_t - c_t))\}$$

- Divide by Δ

$$0 = \max_{c_t}\{u(c_t) + \frac{v(k_t + \Delta(k_t^\alpha - \delta k_t - c_t)) - v(k_t)}{\Delta} \\ - \rho v(k_t + \Delta(k_t^\alpha - \delta k_t - c_t))\}$$

The Ramsey growth model

- From before

$$0 = \max_{c_t}\{u(c_t) + \frac{v(k_t + \Delta(k_t^\alpha - \delta k_t - c_t)) - v(k_t)}{\Delta} - \rho v(k_t + \Delta(k_t^\alpha - \delta k_t - c_t))\}$$

- Take limit $\Delta \to 0$ and rearrange

$$\rho v(k_t) = \max_{c_t}\{u(c_t) + v'(k_t)(k_t^\alpha - \delta k_t - c_t)\}$$

- This is know as the Hamilton-Jacobi-Bellman (HJB) equation.

The Ramsey growth model: Solving

- Dropping time notation we have

$$\rho v(k) = \max_c \{u(c) + v'(k)(k^\alpha - \delta k - c)\}$$

- This is simple to solve and (can be) blazing fast!

The Ramsey growth model: Solving

- Dropping time notation we have

$$\rho v(k) = \max_c \{u(c) + v'(k)(k^\alpha - \delta k - c)\}$$

- This is simple to solve and (can be) blazing fast!
- Why fast? Maximization is trivial: First order condition

$$u'(c) = v'(k)$$

- So if we know $v'(k)$ we know optimal c without searching for it!

The Ramsey growth model: Solving

- How do we find $v'(k)$?
- Suppose we have hypothetical values of $v(k)$ on a uniformly spaced grid of k, $\mathcal{K} = \{k_0, k_1, \ldots, k_N\}$ with stepsize Δk.
- We can then approximate $v'(k)$ at gridpoint k_i ($i \neq 1, N$) as

$$
\begin{aligned}
v'(k_i) = 0.5(v(k_{i+1}) - v(k_i))/\Delta k \\
+ 0.5(v(k_i) - v(k_{i-1}))/\Delta k
\end{aligned}
$$

or

$$
v'(k_i) = \frac{v(k_{i+1}) - v(k_{i-1})}{2\Delta k}
$$

The Ramsey growth model: Solving

- and for k_1 and k_N

$$v'(k_1) = (v(k_2) - v(k_1))/\Delta k$$

and

$$v'(k_N) = (v(k_N) - v(k_{N-1}))/\Delta k$$

- There are many ways of doing this. If you have a vector of $v(k)$ values – call it `V` – then `dV=gradient(V)/dk`.

The Ramsey growth model: Solving

- I prefer an alternative method.
- Construct the matrix D as

$$D = \begin{pmatrix} -1/dk & 1/dk & 0 & 0 & \dots & 0 \\ -0.5/dk & 0 & 0.5/dk & 0 & \dots & 0 \\ 0 & -0.5/dk & 0 & 0.5/dk & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1/dk & 1/dk \end{pmatrix}$$

- Then

$$v'(k) \approx D \times v(k)$$

The Ramsey growth model: Solving

Algorithm

1. Construct a grid for k.
2. For each point on the grid, guess for a value of $\texttt{V}_0$.
3. Calculate the derivative as $\texttt{dV}_0\texttt{=D*V}_0$.
4. Find $\texttt{V}_1$ from

$$\rho V_1 = u(c_0) + dV_0(k^\alpha - \delta k - c_0),$$
$$\text{with} \quad u'(c_0) = dV_0$$

5. Back to step 3 with V_1 replacing V_0. Repeat until convergence.

The Ramsey growth model: Solving

Algorithm

1. Construct a grid for k.
2. For each point on the grid, guess for a value of `V0`.
3. Calculate the derivative as `dV0=D*V0`.
4. Find `V1` from

$$\rho V_1 = u(c_0) + dV_0(k^\alpha - \delta k - c_0),$$
$$\text{with} \quad u'(c_0) = dV_0$$

5. Back to step 3 with V_1 replacing V_0. Repeat until convergence.

Beware: The contraction mapping theorem does not work, so convergence is an issue. Solution: update slowly. That is, $V_1 = \gamma V_1 + (1-\gamma)V_0$, for a low value of γ.

The Ramsey growth model: Euler equation

- Let's go back to the HJB equation.

$$\rho v(k) = u(c) + v'(k)(k^\alpha - \delta k - c)$$
$$\text{with} \quad u'(c) = v'(k)$$

- Thus

$$\rho v'(k) = v''(k)(k^\alpha - \delta k - c) + v'(k)(\alpha k^{\alpha-1} - \delta)$$

- And

$$v''(k) = u''(c)c'(k)$$

The Ramsey growth model: Euler equation

- Using

$$\rho v'(k) = v''(k)(k^\alpha - \delta k - c) + v'(k)(\alpha k^{\alpha-1} - \delta)$$

Together with $v'(k) = u'(c)$ and $v''(k) = u''(c)c'(k)$ gives

$$\rho u'(c) = u''(c)c'(k)(k^\alpha - \delta k - c) + u'(c)(\alpha k^{\alpha-1} - \delta)$$

or

$$-u''(c)c'(k)(k^\alpha - \delta k - c) = u'(c)(\alpha k^{\alpha-1} - \delta - \rho)$$

- Suppose CRRA utility, such that $\frac{u''(c)c}{u'(c)} = -\gamma$

The Ramsey growth model: Euler equation

- Then the last equation

$$-u''(c)c'(k)(k^{\alpha} - \delta k - c) = u'(c)(\alpha k^{\alpha-1} - \delta - \rho)$$

is equal to

$$\gamma \frac{c'(k)}{c}(k^{\alpha} - \delta k - c) = (\alpha k^{\alpha-1} - \delta - \rho)$$

- This is the Euler equation in continuous time.

The Ramsey growth model: Euler equation

Before we attempt to solve the Euler equation, recall that we had

$$k_{t+\Delta} + \Delta c_t = \Delta k_t^\alpha + (1 - \Delta\delta)k_t$$

rearrange

$$k_{t+\Delta} - k_t = \Delta(k_t^\alpha - \delta k_t - c_t)$$

Divide with Δ and take limit $\Delta \to 0$ to get

$$\dot{k}_t = k_t^\alpha - \delta k_t - c_t$$

Or dropping time notation

$$\dot{k} = k^\alpha - \delta k - c$$

The Ramsey growth model: Euler equation

- Our Euler equation is

$$\frac{c'(k)}{c}(k^\alpha - \delta k - c) = \frac{1}{\gamma}(\alpha k^{\alpha-1} - \delta - \rho)$$

or now

$$\gamma \frac{c'(k)}{c}\dot{k} = (\alpha k^{\alpha-1} - \delta - \rho)$$

- What is $c'(k)\dot{k}$? Recall chain rule

$$\dot{c} = \frac{\partial c_t}{\partial t} = \frac{\partial c_t}{\partial k}\frac{\partial k}{\partial t} = c'(k)\dot{k}$$

- Thus

$$\frac{\dot{c}}{c} = \frac{1}{\gamma}(\alpha k^{\alpha-1} - \delta - \rho)$$

The Ramsey growth model: Exploring

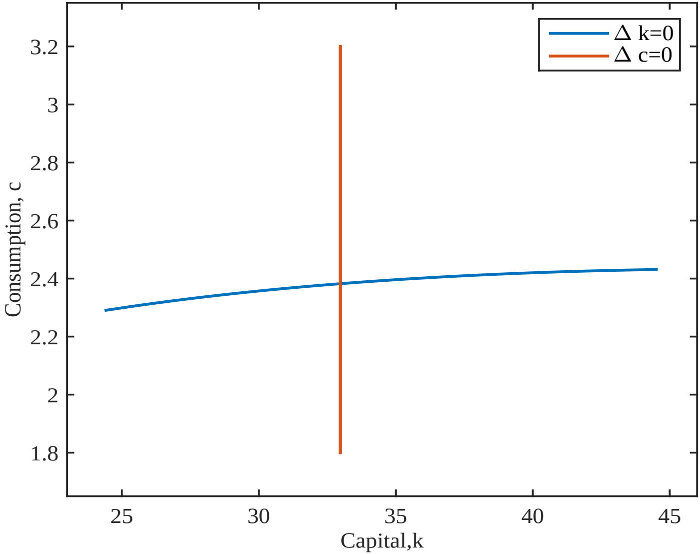
- Two equations

$$\dot{c} = \frac{c}{\gamma}(\alpha k^{\alpha-1} - \delta - \rho)$$
$$\dot{k} = k^{\alpha} - \delta k - c$$

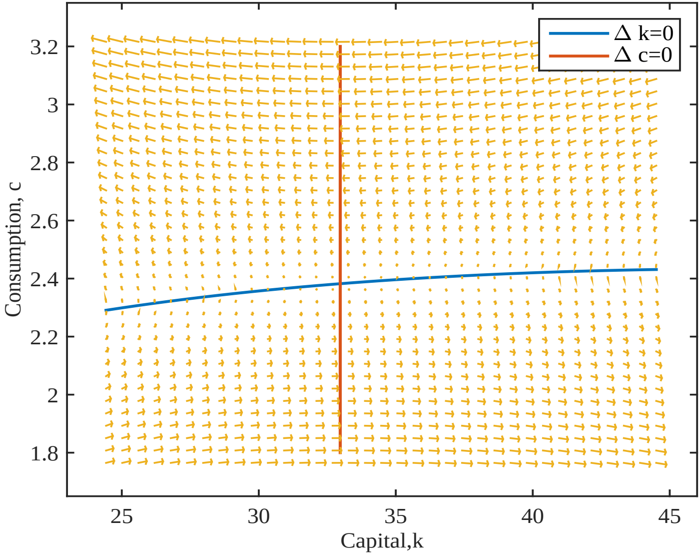
- Nullclines

$$0 = \alpha k^{\alpha-1} - \delta - \rho$$
$$0 = k^{\alpha} - \delta k - c$$

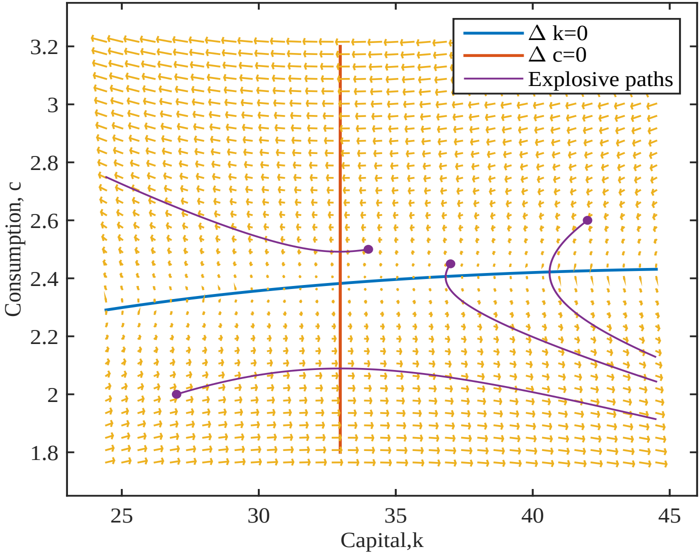
The Ramsey growth model: Exploring



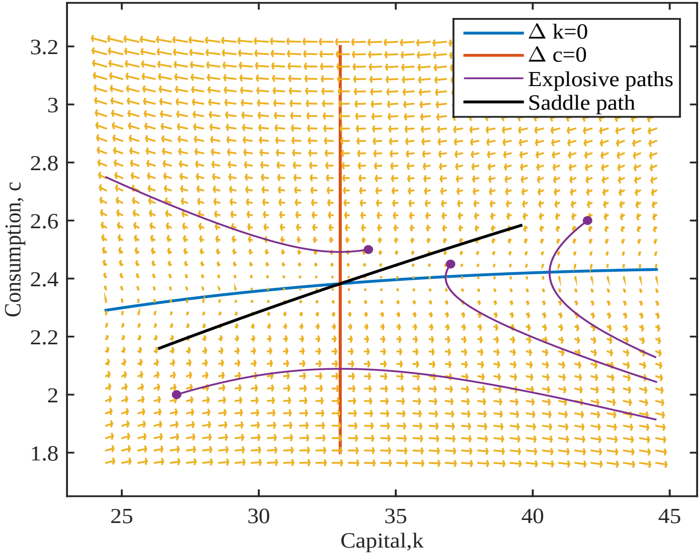
The Ramsey growth model: Exploring



The Ramsey growth model: Exploring



The Ramsey growth model: Exploring



The Ramsey growth model: Exploring

- How did I do that?
- I created a grid for k and c, and found $\dot{c}$ and $\dot{k}$ through

$$\dot{c} = \frac{c}{\gamma}(\alpha k^{\alpha-1} - \delta - \rho)$$
$$\dot{k} = k^{\alpha} - \delta k - c$$

- Then I used Matlab's command `quiver(k,c,`$\dot{k}$`,`$\dot{c}$`)`
  - This creates the swarm of arrows
- I then used Matlab's command `streamline(k,c,`$\dot{k}$`,`$\dot{c}$`)` at various starting values to get the explosive paths.
- Lastly I solved for the saddle path and plotted it.

The Ramsey growth model: Euler equation solution

- Back to the "recursive" Euler

$$\frac{c'(k)}{c}(k^{\alpha} - \delta k - c) = \frac{1}{\gamma}(\alpha k^{\alpha-1} - \delta - \rho)$$

- Solve for c

$$c = \frac{c'(k)(k^{\alpha} - \delta k)}{\frac{1}{\gamma}(\alpha k^{\alpha-1} - \delta - \rho) + c'(k)}$$

The Ramsey growth model: Euler equation solution

Algorithm

1. Construct a grid for k.
2. For each point on the grid, guess for a value of c_0.
3. Calculate the derivative as `dc0=D*c0`.
4. Find c_1 from

$$c_1 = \frac{dc_0(k^\alpha - \delta k)}{\frac{1}{\gamma}(\alpha k^{\alpha-1} - \delta - \rho) + dc_0}$$

5. Back to step 3 with c_1 replacing c_0. Repeat until convergence.

The Ramsey growth model: Euler equation solution

Algorithm

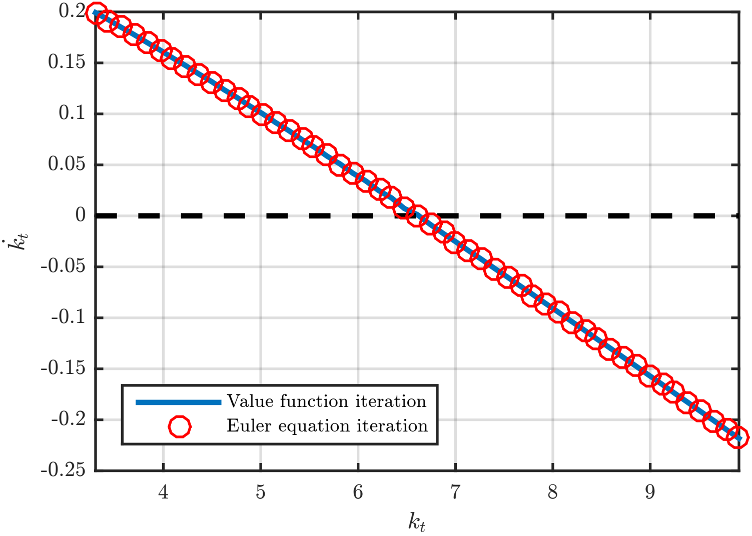
1. Construct a grid for k.
2. For each point on the grid, guess for a value of c_0.
3. Calculate the derivative as `dc0=D*c0`.
4. Find c_1 from

$$c_1 = \frac{dc_0(k^\alpha - \delta k)}{\frac{1}{\gamma}(\alpha k^{\alpha-1} - \delta - \rho) + dc_0}$$

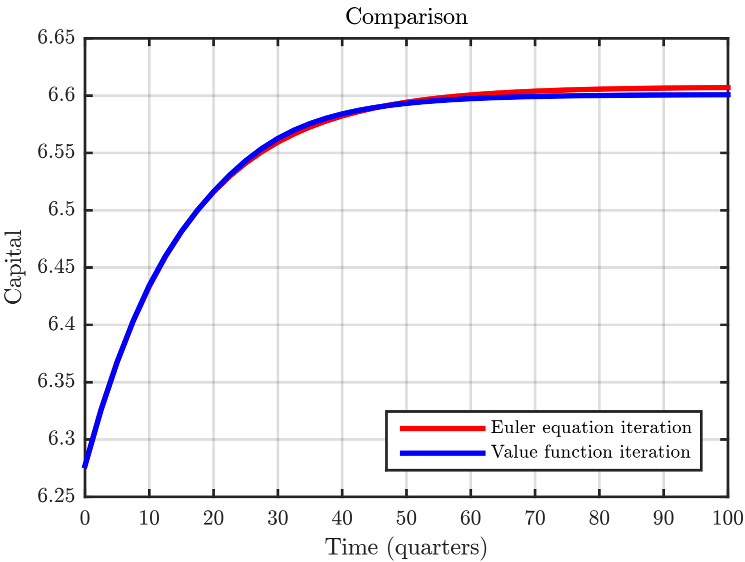
5. Back to step 3 with c_1 replacing c_0. Repeat until convergence.

Beware: No guaranteed convergence. Update slowly. Fewer gridpoints appears to provide some stability.

The Ramsey growth model: Solution



The Ramsey growth model: Solution



More on continuous time Euler equations

- We derived the Euler equation in a slightly roundabout way
  1. Discrete time Bellman equation
  2. To continuous time HJB equation
  3. To continuous time Euler equation using the envelope condition.
- This can be done more directly from the discrete time Euler equation.

More on continuous time Euler equations

- The discrete time Euler equation is given by

$$u'(c_t) = (1-\rho)(1+\alpha k_{t+1}^{\alpha-1} - \delta)u'(c_{t+1})$$

- In Δ units of time

$$u'(c_t) = (1-\Delta\rho)(1+\Delta(\alpha k_{t+\Delta}^{\alpha-1} - \delta))u'(c_{t+\Delta})$$

- Use the approximation $x_{t+\Delta} \approx x_t + \dot{x}_t\Delta$ to get

$$u'(c_t) = (1-\Delta\rho)(1+\Delta(\alpha k_{t+\Delta}^{\alpha-1} - \delta))u'(c_t + \dot{c}_t\Delta)$$

More on continuous time Euler equations

$$u'(c_t) = (1 - \Delta\rho)(1 + \Delta(\alpha k_{t+\Delta}^{\alpha-1} - \delta))u'(c_t + \dot{c}_t\Delta)$$

- Move the $u'(c_t + \dot{c}_t\Delta)$ term to the left-hand side and expand

$$\begin{aligned} &u'(c_t) - u'(c_t + \dot{c}_t\Delta) \\ &\quad = \Delta[\alpha k_{t+\Delta}^{\alpha-1} - \delta - \rho - \rho\Delta(\alpha k_{t+\Delta}^{\alpha-1} - \delta)]u'(c_t + \dot{c}_t\Delta) \end{aligned}$$

- Divide by Δ and take limits $\Delta \to 0$

$$-u''(c_t)\dot{c}_t = [\alpha k_t^{\alpha-1} - \delta - \rho]u'(c_t)$$

More on continuous time Euler equations

$$-u''(c_t)\dot{c}_t = [\alpha k_t^{\alpha-1} - \delta - \rho]u'(c_t)$$

- Lastly, use the CRRA property to get

$$\frac{\dot{c}_t}{c_t} = \frac{1}{\gamma}[\alpha k_t^{\alpha-1} - \delta - \rho]$$

Let's take a step back

- The HJB equation is given by

$$\rho v(k) = u(c) + v'(k)(k^\alpha - \delta k - c)$$

- I mentioned that the iteration

$$\rho v_{n+1}(k) = u(c) + v_n'(k)(k^\alpha - \delta k - c)$$

  is not a contraction mapping.

Why is the contraction property lost?

- Consider the deterministic Ramsey growth model again

$$v(k) = \max_c \{u(c) + (1-\rho)v(f(k) + (1-\delta)k - c)\}.$$

- In discrete time we iterate as

$$v_{n+1}(k) = \max_c \{u(c) + (1-\rho)v_n(f(k) + (1-\delta)k - c)\},$$

- This is a contraction mapping and we know that $v_n \to v$.

Why is the contraction property lost?

- Let's, heuristically, convert this into continuous time

$$v_{n+1}(k) = \max_c \{\Delta u(c) + (1 - \Delta\rho) v_n(k + \Delta(f(k) - \delta k - c))\}.$$

$$0 = \max_c \{u(c) + \frac{v_n(k + \Delta(f(k) - \delta k - c)) - v_{n+1}(k)}{\Delta} - \rho v_n(k + \Delta(f(k) - \delta k - c))\}.$$

Taking limits and rearranging

$$\rho v_n(k) = \max_c \{u(c) + \lim_{\Delta \to 0} \frac{v_n(k + \Delta(f(k) - \delta k - c)) - v_{n+1}(k)}{\Delta}\}.$$

Why is the contraction property lost?

- **Problem 1:**

$$\lim_{\Delta \to 0} \frac{v_n(k + \Delta(f(k) - \delta k - c)) - v_{n+1}(k)}{\Delta}$$
$$\neq v_n'(k)(f(k) - \delta - c)$$

  The right hand side of the HJB equation contains v_{n+1}.

- **Problem 2:** The left hand side of the HJB equation is v_n.

How to get it back

- Back to discrete time

$$v_{n+1}(k) = \max_c \{u(c) + (1-\rho)v_n(f(k) + (1-\delta)k - c)\},$$

- Call the optimal choice c_n (it's really a function of k but I'm saving some space)
- Howard's Improvement Algorithm says that we can then iterate on

$$v_{n+1}^{h+1}(k) = u(c_n) + (1-\rho)v_{n+1}^{h}(f(k) + (1-\delta)k - c_n)\},$$

  with $v_{n+1}^0 = v_n$.
- Until $v_{n+1}^{h+1} \approx v_{n+1}^h$. This can speed things up considerably, and preserves the contraction property

How to get it back

- Suppose that it holds exactly $v_{n+1}^{h+1} = v_{n+1}^{h}$, and let's just call this function v_{n+1}. Then it must satisfy

$$v_{n+1}(k) = u(c_n) + (1-\rho)v_{n+1}(f(k) + (1-\delta)k - c_n),$$

- In Δ units of time

$$v_{n+1}(k) = \Delta u(c_n) + (1-\Delta\rho)v_{n+1}(k + \Delta(f(k) - \delta k - c_n)).$$

How to get it back

Rearrange

$$0 = u(c_n) + \frac{v_{n+1}(k + \Delta(f(k) - \delta k - c_n)) - v_{n+1}(k)}{\Delta} \\ - \rho v_{n+1}(k + \Delta(f(k) - \delta k - c_n))\}.$$

and take limits

$$\rho v_{n+1}(k) = u(c_n) + v'_{n+1}(k)(f(k) - \delta k - c_n),$$

How to get it back

$$\rho v_{n+1}(k) = u(c_n) + v'_{n+1}(k)(f(k) - \delta k - c_n),$$

- Now the awkward discrepancy between v_{n+1} and v_n is gone!
- But the problem looks a bit hard to solve!
- Turns out it is not!
- This is where the "implicit method" comes in.

The implicit method

1. Start with a grid for capital $\mathbf{k} = [k_1, k_2, \ldots, k_N]$.
2. For each grid point for capital you have a guess for $v_0(k_i)$, $\forall k_i \in \mathbf{k}$
3. So you have a vector of N values of v_0. Call this $\mathbf{v}_0$
4. You should also have a difference operator (an $N \times N$ matrix) $\mathbf{D}$ such that

$$\mathbf{D}\mathbf{v} \approx \mathbf{v}'(k), \quad \forall k_i \in \mathbf{K}$$

The implicit method

5. Optimal consumption choice given by FOC

$$u'(\mathbf{c}_0) = \mathbf{D}\mathbf{v}_0$$

   reasonable to call this $c(\mathbf{v}_0)$ – an $N \times 1$ vector
6. This implies another $N \times 1$ vector of savings

$$\mathbf{s}_0 = (f(\mathbf{k}) - \delta\mathbf{k} - c(\mathbf{v}_0))$$

   (This vector can be used to improve on $\mathbf{D}$ – more on that in a second).
7. Create the $N \times N$ matrix $\mathbf{S}_0 = diag(\mathbf{s_0})$

The implicit method

That is

$$\mathbf{S} = \begin{pmatrix} s_1 & 0 & \dots & 0 \\ 0 & s_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & s_N \end{pmatrix},$$

The implicit method

8. Then our HJB equation can now be written as

$$\rho \mathbf{v}_1 = u(c(\mathbf{v}_0)) + \mathbf{S}_0 \mathbf{D} \mathbf{v}_1$$

9. Manipulate

$$(\rho \mathbf{I} - \mathbf{S}_0 \mathbf{D}) \mathbf{v}_1 = u(c(\mathbf{v}_0))$$

10. Lastly

$$\mathbf{v}_1 = (\rho \mathbf{I} - \mathbf{S}_0 \mathbf{D})^{-1} u(c(\mathbf{v}_0))$$

11. Generally

$$\mathbf{v}_{n+1} = (\rho \mathbf{I} - \mathbf{S}_n \mathbf{D})^{-1} u(c(\mathbf{v}_n))$$

The implicit method

Or even more generally

$$\mathbf{v}_{n+1} = ((\rho + 1/\Gamma)\mathbf{I} - \mathbf{S}_n\mathbf{D})^{-1}[u(c(\mathbf{v}_n)) + \mathbf{v}_n/\Gamma]$$

for Γ very large (my experience: $\Gamma = \infty$ is fastest, but set lower if convergence issues arise)

- In matlab always use backslash operator to calculate $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. I.e. $\mathbf{x} = \mathbf{A}\backslash\mathbf{b}$

We are talking very substantial speed/robustness gains here. Perhaps by a factor of 1,000.

The implicit method: Improvement trick I

- We created the matrix $\mathbf{D}$ as central differences
- We can do better. In particular, $\mathbf{s}_n$ tells us where the economy is drifting for each $k_i \in \mathbf{k}$
- So trick one is to use forward differences for all

$$\{k_i \in \mathbf{k} : s_i > 0\}$$

- and backward differences for all

$$\{k_i \in \mathbf{k} : s_i < 0\}$$

- This leads to

$$\mathbf{v}_{n+1} = ((\rho + 1/\Gamma)\mathbf{I} - \mathbf{S}_n\mathbf{D}_n)^{-1}[u(c(\mathbf{v}_n)) + \mathbf{v}_n/\Gamma]$$

The implicit method: Improvement trick II

- Inspect the matrix

$$((\rho + 1/\Gamma)\mathbf{I} - \mathbf{S}_n\mathbf{D}_n),$$

  and notice that all matrices are super sparse!
- So declaring them as sparse will free up a lot of memory and give you enormous speed gains too (this is particularly true for problems with $N > 200$ or so. Below that it doesn't really matter).
- **Never** declare any of these matrices as anything else than sparse! Use commands as `speye` and `spdiags`
- **Don't** be too concerned about loops. That doesn't seem to be what can clog these systems.