Macroeconomics B Notes

Nicholas Umashev *

2019

*Please see google notes

Dynamic Optimisation in Continuous Time

Continuous vs Discrete Time: in discrete time the intervals between periods are $\Delta > 0$ (e.g. the interval between t and $t+1$ is 1), in continuous time the intervals between periods are $\Delta \to 0$

- Continuous Variables Issue: while state variables are often naturally continuous, when using computational methods we can only evaluation the value function on a finite number of points
  - Solution: we can address this problem via using (1) discretization of the state space, or, (2) polynomial approximation to the value function

Discretization Method: consider a problem with continuous state and control variables $\mathfrak{x} \in \mathbb{R}$, discretization just replaces $\mathfrak{x}$ and $\mathfrak{u}$ by the finite grids $\widehat{\mathfrak{x}} = \{x^1, \ldots, x^n\}$ and $\widehat{\mathfrak{u}} = \{u^1, \ldots, u^n\}$

- Value Function Impact: now the value function becomes a finite list of numbers, $V = [V^1, \ldots, V^n]^T$
- Advantage: the maximization step is much simpler than under the original bellman equation, which is a key advantage of discretization methods
- Disadvantage: there is a "curse of dimensionality" in muiltidimensional state spaces where we must decides whether to have N points for a one-dimensional state space vs N^k points for a k-dimensional state space
  - Grid Decision: requires some a-priori information about the state space, which is sometimes difficult to obtain (e.g. upper and lower bounds)
- Optimal Growth Illustration: suppose $V(k) = \max_{k'} \{\ln(Ak^\alpha - k') + \beta V(k')\}$ where in this case $\mathfrak{x} = \mathfrak{u}$
  - Discretization: if we discretize $\mathfrak{x}$ then the Bellman equation becomes $V_i = \max_j \{\ln(Ak_i^\alpha - k_j) + \beta V_j\}$ for all $i = 1, \ldots, n$
  - Value Function Iteration Solution: suppose we applied value function iteration and therefore iterate on the mapping $V_i^s = \max\left\{\ln(Ak_i^\alpha - k_1) + \beta V_1^{s-1}, \ldots, \ln(Ak_i^\alpha - k_n) + \beta V_n^{s-1}\right\}$ for all $i = 1, \ldots, n$ where s indexes the iteration step

VF ITERATION

(1) Start w/ a guess:

$$\underset{n\times 1}{V^0} = \begin{bmatrix} V_1^0 \\ V_2^0 \\ \vdots \\ V_n^0 \end{bmatrix}, \text{ where } V_i^0 \equiv V^0(k_i)$$

(2) Find V^1 by solving:

$$V_i^1 = \max_j \left\{ \ln\left(Ak_i^\alpha - k_j\right) + \beta V_j^0 \right\}$$

expanding

*

$$V_i^1 = \max \begin{bmatrix} \ln(Ak_i^\alpha - k_1) + \beta V_1^0 \\ \ln(Ak_i^\alpha - k_2) + \beta V_2^0 \\ \vdots \\ \ln(Ak_i^\alpha - k_n) + \beta V_n^0 \end{bmatrix} \qquad \forall i$$

so I end up with:

$$\underset{n \times 1}{V^1} = \begin{bmatrix} V_1^1 \\ V_2^1 \\ \vdots \\ V_n^1 \end{bmatrix}$$

*

(3) Check for convergence:

$$\|V^1 - V^0\| \le \varepsilon \longrightarrow \text{tolerance level}$$

$$\| \| \qquad \sqrt{\sum_{i=1}^{n} \left(V_i^1 - V_i^0\right)^2} \longrightarrow \text{Euclidean Norm.}$$

(4) Compute V^s by solving:

$$V_i^s = \max \left\{ \ln(Ak_i^\alpha - k_j) + \beta V_j^{s-1} \right\} \qquad \forall i$$

(5) Check convergence

$$\|V^s - V^{s-1}\| \le \varepsilon \longrightarrow$$

*

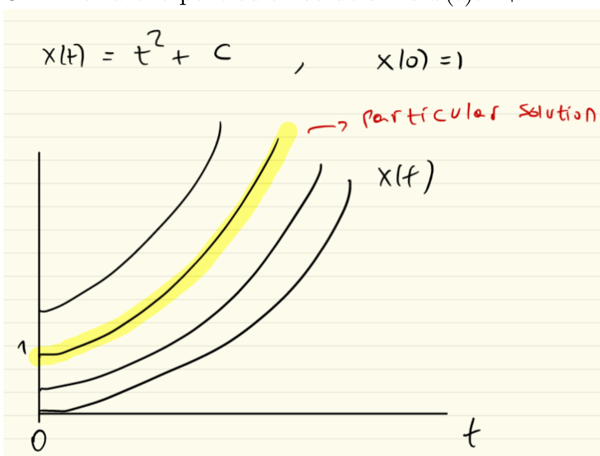
- If $\|V^s - V^{s-1}\| \le \varepsilon$ holds $\longrightarrow$ $\underline{\underline{\text{STOP}}}$
- If $\|V^s - V^{s-1}\| > \varepsilon$ does not hold

$\Rightarrow$ Go Back to step (4)

*

<u>Ordinary Differential Equations (ODEs)</u>: a "differential" equation" is one where the unknown is a function (instead of a variable) and the equation includes one or more of the derivatives of the function, an "ordinary differential equation" equation is one for which the unknown is a function of only one variable (typically time)

- <u>Partial ODEs</u>: where the unknown is a function of more than one variable
- <u>First-Order ODE Form</u>: $\dot{x}(t) = F(t, x(t))$, where $\dot{x}(t) \equiv dx(t)/dt$ and $t \in [t_a, t_b]$

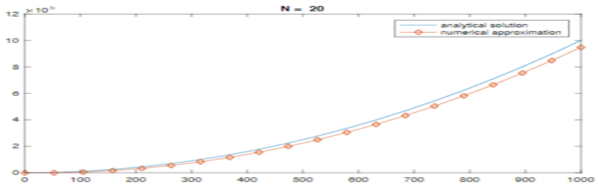
- Unknown: here the unknown is a function $x(t)$ with $x : [t_a, t_b] \to \mathbb{R}$
- Uniqueness: the solution is not unique with the form having infinitely many solutions indexed by an integrating constant C. However, generally the constant C can be uniquely determined by requiring the solution to pass through a given point on the tx-plane
- Example: $\dot{x}(t) = 2t$ with $t \in [0, \infty)$
  * Solution without Initial Condition: the general solution is $x(t) = t^2 + C$
  * Solution with Initial Condition: suppose we impose the initial condition $x(0) = 1$, therefore $C = 1$ and the particular solution is $x(t)t^2 + 1$
  * 

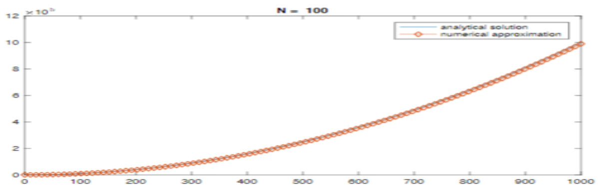
• Finite-Difference Methods for Solution: here we approximate derivates using finite-differences to approximate the solution to the ODE. This involves finding $\dot{x}(t) = F(t, x(t))$, $x(t_a) = x_a$, and $t \in [t_a, t_b]$. There are a wide range of methods depending on how the derivatives are approximated
- Euler's Method
  * Step 1: specify a grid for t, i.e. $t_0 = t_a < t_1 < t_2 < \cdots < t_N = t_b$
  * Step 2: approximate the ODE using the difference equation

$$\frac{x(t_{i+1}) - x(t_i)}{t_{i+1} - t_i} = F(t_i, x(t_i))$$

over $i = 0, \ldots, N-1$, where $x(t_0) = x_a$ is fixed by the initial condition

$$\dot{x}(t) = 2t, \qquad x(0) = 1, \qquad t \in [0, 1000]$$





*

Euler's Method

① Discretize $[t_a, t_b]$

② Iterate forward on:

$$x(t_{i+1}) = x(t_i) + (t_{i+1} - t_i)\, F(t_i, x(t_i)) \quad \forall i$$

$x(t_a) = x_a$

$\underline{i=0}$:

$$x(t_1) = x(t_0) + (t_1 - t_0)\, F(t_0, x(t_0))$$

$\underline{i=1}$: --- (etc.)

*