A Primer on Local Solution Methods
– Warming Up to Dynare –

Macroeconomics 2
EUI, Spring 2017

Johannes Fleck

February 10, 2017

The aim of this talk

My main objectives today

- Illustrate the type of problem Dynare is used for
- Present two general solution methods
- Discuss some properties of local solution methods

I hope this will help you to

- Understand better what Dynare does
- Build intuition for what can go wrong
- See that Dynare is useful for a large class of Macro problems

---

Some slides borrow material from lectures given at ZICE 2017

The problem to be solved

- Solving most econ models (NK, RBC, OLG) typically requires finding solutions to a system of (non-linear) equations
  - Rootfinding problem:
    Given $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, find vector x s.t. $f(x) = 0$
  - Fixed point problem:
    Given $g : \mathbb{R}^n \mapsto \mathbb{R}^n$, find vector x s.t. $x = g(x)$
- Note these formulations are isomorphic
  - Rootfinding as fixed point problem: Let $g(x) = x - f(x)$
  - Fixed point as rootfinding problem: Let $f(x) = x - g(x)$
- Other than in special cases (i.e. outside of the classroom), these problems cannot be solved analytically

Illustration of a common problem 1/3

Setting

- Consider a deterministic Ramsey model given as

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} u\big(\{c_t\}_{t=0}^{\infty}\big) \tag{1}$$

$$\text{s.t. } 0 \leq c_t \tag{2}$$

$$0 \leq k_{t+1} \tag{3}$$

$$k_{t+1} \leq f(k_t) + (1-\delta)k_t - c_t \quad [= \overline{f}(k_t) - c_t] \tag{4}$$

$$k_0 \text{ and technology } f(k_t) = k_t^{\alpha} \text{ are given} \tag{5}$$

- Standard assumptions on preferences and production imply
  - (2) and (3) never bind
  - (4) always binds: $k_{t+1} = \overline{f}(k_t) - c_t \; \forall t$

Illustration of a common problem 2/3

The key equation

- The associated Lagrangian and its FOCs are fairly tractable

$$\mathcal{L} = \sum_{t=0}^{\infty} \beta^t [u(c_t) + \lambda_t(\overline{f}(k_t) - c_t - k_{t+1}] \tag{6}$$

$$[c_t] \quad \lambda_t = u'(c_t) \tag{7}$$

$$[k_{t+1}] \quad \lambda_t = \beta \lambda_{t+1} \overline{f}'(k_{t+1}) \tag{8}$$

- Our bff the Euler Equation is

$$u'(\overline{f}(k_t) - k_{t+1}) = \beta \overline{f}'(k_{t+1}) u'(\overline{f}(k_{t+1}) - k_{t+2}) \tag{9}$$

Illustration of a common problem 3/3

The Ramsey plan

- Finding the equilibrium consumption function $C(k_t)$ is tricky
  - Need to solve the infinite sequence of the EE
- But we have another option
  - Reduce problem to two periods: today and tomorrow
  - Suppose optimal choice does not depend on t but on k_t
  - Look for recursive equilibrium with k as endogenous state
  - Still no analytical solution but can now re-write EE

$$u'(C(k)) = \beta \overline{f}'(\overline{f}(k) - C(k))u'(C(\overline{f}(k) - C(k))) \tag{10}$$

  - If we specify u and $\overline{f}$, we can re-arrange this into

$$\widetilde{C}(k; u, \overline{f}) = 0 \tag{11}$$

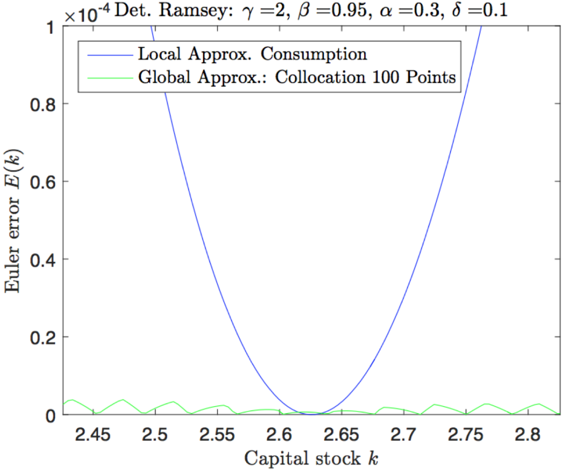
Lack of analytical solution requires numerical methods

- At this point, we have decide how to approximate $\widetilde{C}(k)$
  1. Want *all* $k \in \mathbb{R} \rightarrow$ use VFI, PFI (on a capital grid)
  2. Want $k^* \in \subset \mathbb{R} \rightarrow$ focus on steady state

- Numerical approximation procedures
  - For 1: **'Global Solution Methods'**
    Examples: VFI, PFI, Collocation, Chebyshev Galerkin
  - For 2: **'Local Solution Methods'**
    - Using derivatives: Newton's Method with Taylor (and variants)
    - Derivative-free: Pattern Search, Nelder-Mead (and many more)

Global versus local solution methods

- In general, local solutions methods are faster to compute
- But their accuracy is restricted to a small neighborhood



Euler Error measures the % deviation of consumption relative to the optimum

Local solution methods

- Local solution methods are based on perturbation theory
- The idea
  - We don't know which functional form solves a problem $F(x) = 0$
  - But we know how to make simple functional forms behave similar to the unknown function around a particular point
  - Iteratively, we can make them approximate our function of interest as close as we want
- Prominent choices for the ingredients of local solution methods
  - Taylor Series
  - Steady State
  - Newton's Method

Dynare is a local solution toolbox

- Dynare basic (for this course)
  - Implements a local solution method around the steady state
  - Uses that log-linearization yields a first order Taylor Series
    - Recall: log-linearization around the steady state turns non-linear equations into equations which are linear in terms of the log-deviations of their variables from their steady state values.
  - Applies Newton Method based solvers for approximation
- Dynare advanced
  - Can do a lot more - some examples: `http://www.dynare.org/documentation-and-support/examples`
  - May be very helpful for your own research

Illustration of an iterative approximation: Newton's Method

- Suppose our problem has the form $F(x) = 0$ (cf. example above)
- Let's form a Taylor series approximation around a guess x_k

$$F(x) \approx \nabla F(x_k)(x - x_k) + F(x_k) \tag{12}$$

- Solve for x and iterate

$$x_{k+1} = x_k - \nabla F(x_k)/F(x_k) \tag{13}$$

- Iterative methods á la Newton reduce a non-linear problem to a sequence of linear problems
- Example in the following slides

Illustration of an iterative approximation: Newton's Method

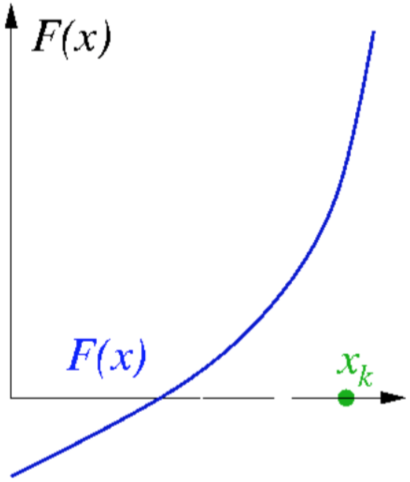


Illustration of an iterative approximation: Newton's Method

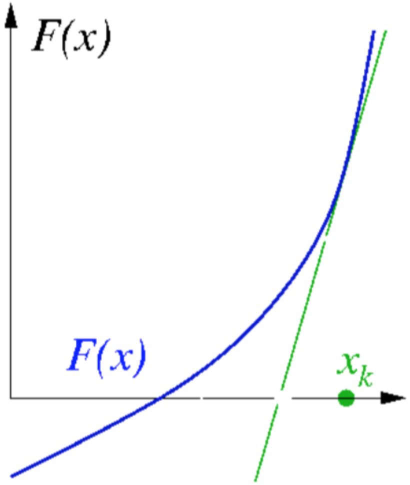


Illustration of an iterative approximation: Newton's Method

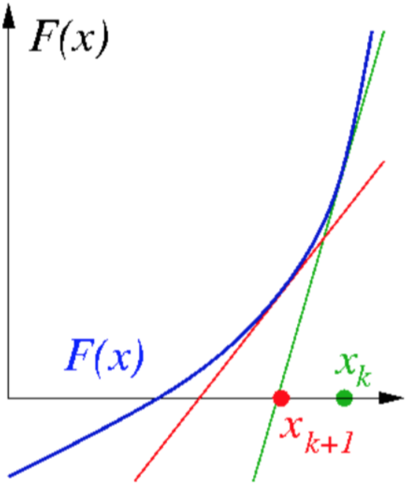
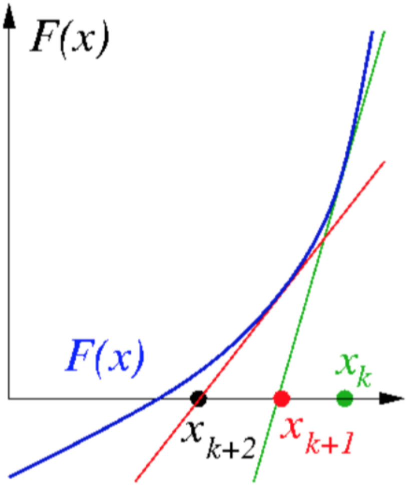


Illustration of an iterative approximation: Newton's Method



return

Iterative methods: some things to keep in mind

- When using iterative procedures, we need to decide when to stop
- Put differently: When are we close (enough) to the solution?
- Not a 'harmless' decision as it hinges on three consequential concepts

1. Stopping criterion
   When is $||F(x_k) - F(x_{k+1})|| - \epsilon = 0$? Choice of ϵ matters!
2. Convergence: asymptotic (limit) behavior
   What happens to $F(x_k)$ as k $\rightarrow \infty$?
3. Again: local versus global
   What type of solution can we find?

1. Stopping criterion

- We stop the iteration once $||F(x_k) - F(x_{k+1})||$ is 'sufficiently' small
- What is sufficiently small in practice?

"It depends"

- To some extent, your computing environment matters
  - Symbolic, e.g. Mathematica: stores numbers exactly (even irrationals)
  - Numeric, e.g. Matlab: stores numbers with finite precision arithmetic
- Finite precision arithmetic restricts the smallest number
  - Single precision: $\epsilon = 5.96 \times 10^{-8}$
  - Double precision: $\epsilon = 1.11 \times 10^{-16}$

$\Rightarrow$ Setting a stopping criterion 'too small' can be a bad idea...

2. Asymptotic (limit) behavior

- In general, three things can happen
  1. Convergence: $\lim_{k\to\infty} x^k = x^*$ Example
  2. Iterate divergence: $\lim_{k\to\infty} ||x^k|| = \infty$ Example
  3. Sequence cycles: Example
     - Multiple convergent subsequences (limit points)
     - Limit points are not solutions

(Partly depends on choice of ϵ)

3. Local versus global

A *really* important announcement on terminology

Given f(x) with $x \in \Omega$ a numerical approximation is said to be

- 'globally convergent'
  Converge to a solution (stationary point) from <u>anywhere</u> in Ω
- 'locally convergent'
  Converge to a solution (stationary point) from <u>close</u> to it
- 'convergent to a global minimizer (maximizer)'
  Converge to x^* with $\mathbf{f}(\mathbf{x}^*) \leq (\geq)\ \mathbf{f}(\mathbf{x})\ \forall \mathbf{x} \in \Omega$

- Newton's Method is locally convergent
  - Some of its variants are globally convergent
- Neither global nor local methods definitely converge to THE solution (assuming it exists...)
  - This depends on the shape of the function (if convex: local = global)

Summing up

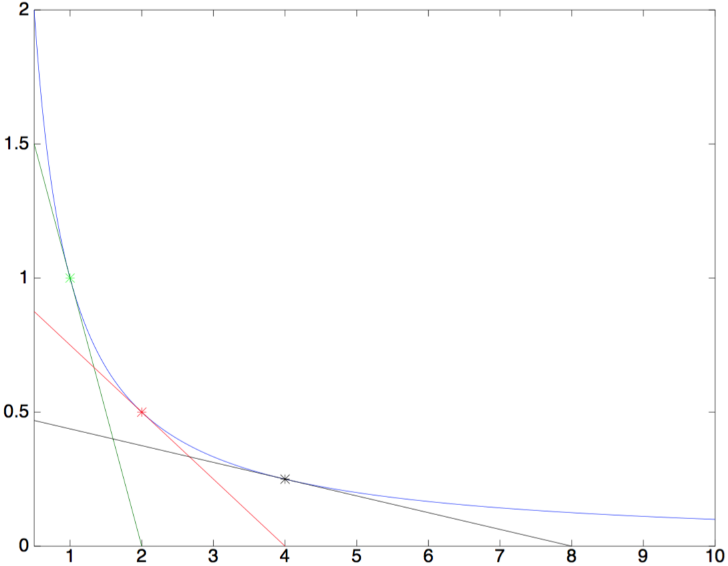
- Dynare can be used to solve a wide range of economic models
  - This includes Aiyagari type models you have seen in macro 1 (see Le Grand and Ragot [2017], den Haan [2017] and others)
- Unless Dynare starts out from close to the steady state solution, you should not expect it to get there

  Impossible to find the steady state. Either the model doesn't have a steady state, there are an infinity of steady states, or the guess values are too far from the solution
  - Trying out different convergence criteria and number of iterations for the steady state solver can be a good idea
  - Model features such as multiple steady states can create serious problems for Dynare
- If you face problems...
  - Keep calm and carry on
  - GIYF and RTFM

Thanks!

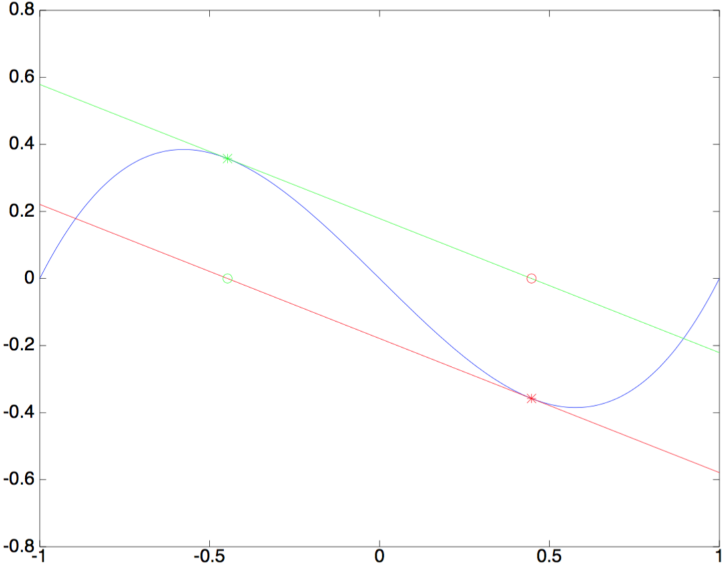
Your Questions?

Illustration of an iterative approximation: Newton's Method



return

Illustration of an iterative approximation: Newton's Method



return