Perturbation

Judd Chapter 13

David Childers (thanks to Y. Kryukov, K. Judd, and U. Doraszelski)

CMU, Tepper School of Business

April 19, 2023

Perturbation method

- Some models in Macro only need policy function near the steady state
- This suggests Taylor series expansion around the steady state
- Relies on Implicit Function Theorem
- Another derivation-intensive method (formerly)
- Now, with automatic differentiation, extremely easy

Implicit Function Theorem

- We have $H(x, y) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}^n$; derivatives H_x, H_y
- We want $h(x) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$, but we only know that:

$$H(x, h(x)) = 0$$

- This implies that $\frac{d}{dx} H(x, h(x)) = 0$:

$$H_x\left(x, h(x)\right) + H_y\left(x, h(x)\right) h_x(x) = 0$$

- Imagine we know x_0 and $y_0 = h\left(x_0\right)$. Then:

$$h_x(x_0) = -\left[H_y\left(x_0, y_0\right)\right]^{-1} H_x\left(x_0, y_0\right)$$

- Linear approximation:

$$h^L(x) = h\left(x_0\right) + h_x\left(x_0\right)\left(x - x_0\right)$$

- Quality check: $E = H(x, h^L(x))$

Higher-order implicit function

- $\frac{d^2}{dxx} H(x, h(x)) = 0$:

$$\begin{aligned} \frac{d}{dx}\left[H_x\left(x, h(x)\right) + H_y\left(x, h(x)\right) h_x(x)\right] &= 0 \\ H_{xx}[.,.] + 2H_{xy}[., h_x[.]] + H_{yy}[h_x[.], h_x[.]] + H_y[h_{xx}[.,.]] &= 0 \end{aligned}$$

  Note that H_{ab} is a 3-dimensional array ($n \times d_a \times d_b$ tensor)
- Let $H^0_{\ldots} = H_{\ldots}(x_0, y_0)$, $h^0_{\ldots} = h_{\ldots}(x_0)$
- Solve for h_{xx} (using h_x known from first order):

$$h^0_{xx}[,] = -\left[H^0_y\right]^{-1}\left(H^0_{xx}[.,.] + 2H^0_{xy}[., h^0_x[.]] + H^0_{yy}[h^0_x[.], h^0_x[.]]\right)$$

- Solve by flattening tensor to vector and solving linear system, or by iterative methods
- Quadratic approximation:

$$h^Q(x) = h^0 + h^0_x(x - x_0) + \frac{1}{2}(x - x_0)^\top h^0_{xx}(x - x_0)$$

Perturbation strategy

- Formulate the problem as an implicit function.
- Pick x^0 to make $y^0 = h\left(x^0\right)$ easy to compute
  - Deterministic dynamic models: steady state
  - Dynamic stochastic model: steady state becomes stable/stationary distribution
  - Assume zero shocks $\Rightarrow$ deterministic problem; steady state will be close to stationary mode at low noise.
- Construct approximation
- Check solution over full range of values

- Taylor approximation is limited by radius of convergence – distance to nearest singular point (any derivative unbounded)

Recursive Models

- Dynamic models often expressed in recursive form
- Especially models w/ dynamic optimization
- Perturbation approach requires setting up as differentiable system of equations
    - Use Euler method or FOC approach to describe conditions
- Combine many equations
    - Multiple agent decisions (firms, consumers, gov)
    - Equilibrium constraints: market clearing, dynamic budget
    - Exogenous dynamics of shocks, policies, etc

Recursive Models

- Discrete time recursive representation

$$E_x F(x, y, x', y') = 0$$

- y are "jump" variables: determined endogenously by x
- $x = (x_1, x_2)$ are "predetermined" variables
- x_2 evolves exogenously via $x_2' = h_2(x_2) + \sigma\epsilon'$
- ϵ is mean 0, (only) source of stochastic shocks
- May need to add variables to get model to fit format
- *Recursive* equilibrium is functions $g(x), h(x) = (h_1(x), h_2(x_2))$ s.t. $\forall x$

$$E_x F(x, g(x), h(x) + \sigma\epsilon', g(h(x) + \sigma\epsilon')) = 0$$

- Nested structure calls for modified approach

Perturbation for Recursive Models

- Expand around *nonstochastic* steady state
- Set $\sigma = 0$ and find x^*, y^* s.t. $x_2^* = h_2(x_2^*)$ and

$$F(x^*, y^*, x^*, y^*) = 0$$

- Find by nonlinear equation solver
- Want Taylor expansion of h, g in x & σ

$$g(x,\sigma) = y^* + g_x(x - x^*) + g_\sigma \sigma + \dots$$

$$h(x,\sigma) = x^* + h_x(x - x^*) + h_\sigma \sigma + \dots$$

- Derivative wrt x, σ at $x^*, 0$ must be 0: Apply chain rule

$$F_x + F_y g_x + F_{x'} h_x + F_{y'} g_x h_x = 0 \tag{1}$$

$$E_x[F_y g_\sigma + F_{x'}(h_\sigma + \epsilon') + F_{y'}(g_x(h_\sigma + \epsilon') + g_\sigma)] = 0 \tag{2}$$

Solving the system

- By 0 mean, can show 2 implies $h_\sigma, g_\sigma = 0$
- Rewrite 1 in matrix form $A = [F_x, F_y]$, $B = -\left[F_{x'}, F_{y'}\right]$

$$A \begin{bmatrix} I \\ g_x \end{bmatrix} = B \begin{bmatrix} I \\ g_x \end{bmatrix} \begin{bmatrix} h_x & 0 \\ 0 & h_x \end{bmatrix} \tag{3}$$

- Matrix equation has many solutions: need additional conditions
- Usually, model is BVP: require transversality
- Sufficient condition: $h^n(x_0) \to x^*$ return to steady state
- In linear case, means $h_x^n(x - x^*) \to 0$
  - Holds if $\max |\mathrm{eig}(h_x)| < 1$
- Together, these conditions select a solution

Matrix Decomposition

- To impose stability, break system into stable and unstable parts
  - Align coordinates along stable vs unstable manifold
- For linear system $x' = Mx$, use diagonalization: Jordan decomposition
- $M = U^{-1}DU$, D (block) diagonal, eigenvalues along diagonal
- $\tilde{x} = Ux$ split into $1-d$ linear maps: solve in closed form
- If A invertible, can apply to $A^{-1}B$ (Blanchard Kahn method)
- In general, A noninvertible & decomposition numerically ill-posed
- Solve both by *Generalized Schur* (QZ) Decomposition $(A, B) = (QSU, QTU)$
- Q, U Unitary S, T block upper triangular
- U unitary means $U^* = U^{-1}$, $UU^* = I$, $U^*U = I$

Applying Matrix Decomposition

- QZ allows decomposing into blocks

$$(S, T) = \left(\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \right)$$

- Ordered so that $\max \left| \mathrm{eig}(S_{11}^{-1} T_{11}) \right| < 1$
- Applying Generalized Schur to 3, obtain

$$QSU \begin{bmatrix} I \\ g_x \end{bmatrix} = QTU \begin{bmatrix} I \\ g_x \end{bmatrix} \begin{bmatrix} h_x & 0 \\ 0 & h_x \end{bmatrix}$$

- Can remove Q from both sides and decompose U conformably

$$\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} I \\ g_x \end{bmatrix} =$$

$$\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} I \\ g_x \end{bmatrix} \begin{bmatrix} h_x & 0 \\ 0 & h_x \end{bmatrix}$$

Solution by Matrix Decomposition

- Simplify

$$\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \begin{bmatrix} U_{11} + U_{12} g_x \\ U_{21} + U_{22} g_x \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} (U_{11} + U_{12} g_x) h_x \\ (U_{21} + U_{22} g_x) h_x \end{bmatrix}$$

- Jump variables move so that system always on stable manifold
- This is ensured if $g_x = -U_{22}^{-1} U_{21}$
- Then second line is 0, and first line implies

$$h_x = (U_{11} - U_{12} U_{22}^{-1} U_{21})^{-1} S_{11}^{-1} T_{11} (U_{11} - U_{12} U_{22}^{-1} U_{21})$$

- By construction, h_x satisfies stability
- To do this, need U_{22}^{-1} to exist: Blanchard-Kahn conditions
  - Map from unstable manifold to jump variables
  - # of jump variables equals # of eigenvalues ≥ 1
  - If too many large eigenvalues, no non-explosive solutions
  - If too few first order info can't pin down unique solution

Properties of linearized solution

- Using above solutions, have linear approx to dynamics
- Can calculate impulse responses, simulate forward, or estimate
- Linear model allows VAR (if all observed) or linear state space model (if not) representation of system
  - Gaussian shocks: construct likelihood by Kalman filter
  - Likelihood-based estimation orders of magnitude faster if linear
- Cost of matrix calculations is $O(n^3)$ in number n of equations
- Compare: nonlinear methods can scale exponentially in n
  - Curse of dimensionality reduced since only ask for local info
- Derivatives can be automatic or symbolic, canned routines exist
  - Dynare, Schmitt-Grohe-Uribe, DSGE.jl, Gensys, even Stata

Going to Higher Order

- Can extend to second and higher order analogously to above

$$h(x,\sigma) \approx h(x^*,0) + h_x(x-x^*) + h_\sigma \sigma + \frac{1}{2} h_{xx}\left[x-x^*, x-x^*\right]$$

$$+ h_{x\sigma}\left[x-x^*,\sigma\right] + \frac{1}{2} h_{\sigma\sigma}\sigma^2 + \dots$$

- Order (k) derivatives found by taking derivatives of $E[F(x, g(x,\sigma), h(x,\sigma)+\sigma\epsilon, g(h(x,\sigma)+\sigma\epsilon,\sigma)] = 0$
- Equations in second derivatives based on second derivatives of system, first derivatives calculated by 1st order method
- Solve this system for second derivatives, then use these in system to find third, etc
- 2^{nd} & higher order methods result in *linear* matrix equations

Higher order perturbation

- Simple to solve and no additional existence/uniqueness issues
- But sytem dimension $O(n^{(k)})$ grows exponentially in order
  - Have to stop at small to moderate order
- Each order increases set of moments of shock that matter
  - Linear solution *certainty equivalent*
  - $\rightarrow$ Same coefs as if no shock at all
  - Variance starts to affect mean at order 2, Skewness at 3, etc
  - Need even higher orders for effect of moment to vary with x
- Higher order perturbation good compromise if system dimension moderate but nonlinearities/interactions matter
- May be unwieldy for estimation or simulation
  - Likelihood methods need particle filter: slow
  - Simulations may fail to be stationary when true solution is
  - May need to "prune" higher order terms for reasonable behavior

When is perturbation feasible?

- Need to be able to write system so differentiable
  - Often just algebra: Take FOCs to get rid of max
  - Can fail with non-smooth model
  - Kinks, discontinuities, discrete variables all cause issues
  - "Mixed" procedures exist for some of these: not always reliable
- Need nonstochastic steady state to exist
- Existence usually not an issue: *uniqueness* is
- None of above improved by higher order solutions

What if steady state nonunique?

- If multiple isolated solutions, can choose 1, but may not be stable
  - Nonlinear solution may involve moving between
  - Even if BK conditions hold, only correct if true shocks bounded, keep system near 1 point
- May have continuum of steady states
  - Some variables not determined with no shocks
  - E.g.: portfolio choice nonunique if all assets have 0 risk
  - Can apply modified approach: bifurcation methods
  - Hack: add small penalty function which differentiates choices
  - Picks out unique choice, but maybe not same as in full nonlinear model

Perturbation Approach: Pros and Cons

- Way faster than projection for moderate order Taylor expansion (say ≤ 5)
- Also way easier to code, nearly completely automated
- Especially suitable for high-dimensional models
    - Method of choice for large DSGE models with many variables
    - In low-moderate dimensions, projection feasible and may be more accurate
- Con: accuracy low far from steady state
- Major con: if no steady state, or many, completely misrepresents dynamic properties
- Works poorly for models with kinks, jumps, multimodality, discreteness, etc
    - Zero lower bound, borrowing constraint, regime switches, etc
- Even if you plan to use global method, often good idea to start local to find initial guesses

Example: Stochastic Growth Model

- Perturbation example and software implementation in Dynare
  - From Collard, Juillard, Villemot (2009)
- Representative consumer chooses consumption and labor to optimize

$$\max_{(c_t,h_t)_{t=0}^{\infty}} E_t \sum_{t=0}^{\infty} \beta^t \log(c_t) - \theta \frac{h_t^{1+\psi}}{1+\psi}$$

$$\text{s.t. } k_{t+1} = \exp(b_t)(y_t - c_t) + (1-\delta)k_t$$

$$y_t = \exp(a_t) k_t^{\alpha} h_t^{1-\alpha}$$

- with exogeneous shocks

$$\begin{bmatrix} a_t \\ b_t \end{bmatrix} = \begin{bmatrix} \rho & \tau \\ \tau & \rho \end{bmatrix} \begin{bmatrix} a_{t-1} \\ b_{t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_t \\ \nu_t \end{bmatrix}$$

$$\text{Cov} \begin{bmatrix} \epsilon_t \\ \nu_t \end{bmatrix} = \begin{bmatrix} \sigma_\epsilon & \phi\sigma_\epsilon\sigma_\nu \\ \phi\sigma_\epsilon\sigma_\nu & \sigma_\nu \end{bmatrix}$$

Implementation Steps

- To solve by perturbation
  1. Represent in terms of recursive differentiable equations
  2. Solve nonlinear system for nonstochastic steady state
  3. Differentiate to get first derivatives
  4. Apply matrix decomposition to get first order solution
  5. (Optional) Differentiate again and solve linear system to get next higher order
  6. Repeat previous step as many times as desired
- Software takes care of steps (2)-(6), (1) usually manual

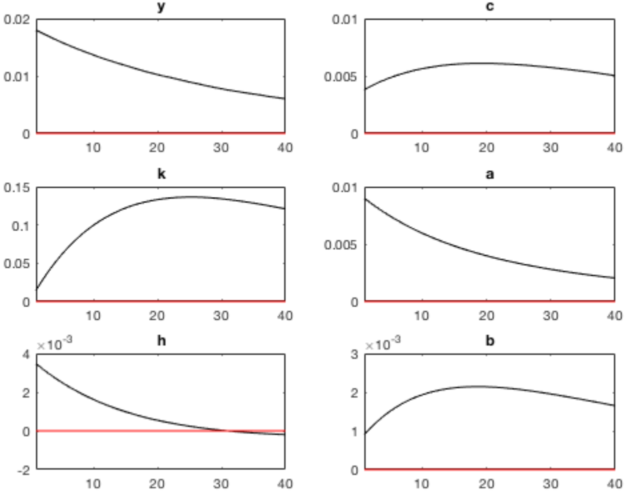
Stochastic Growth Model: Equations

- Step (1) here means deriving Euler equation and FOC for h_t

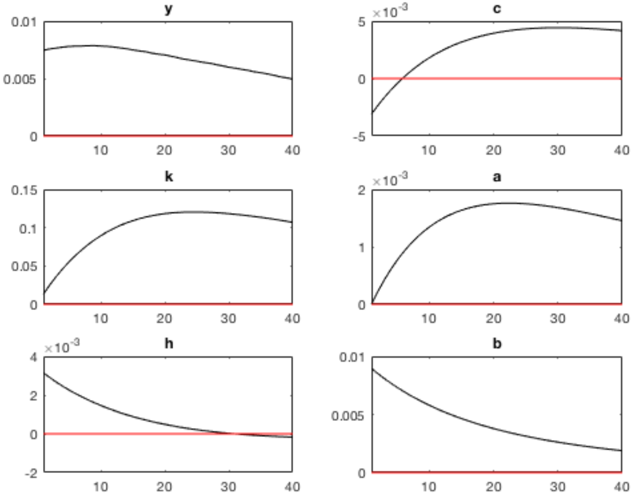
$$
\begin{aligned}
c_t \theta h_t^{1+\psi} &= (1-\alpha) y_t \\
\beta E_t \left[\left(\frac{\exp(b_t) c_t}{\exp(b_{t+1}) c_{t+1}} \right) \left(\exp(b_{t+1} \alpha \frac{y_{t+1}}{k_{t+1}} + 1 + \delta) \right) \right] &= 1 \\
y_t &= \exp(a_t) k_t^{\alpha} h_t^{1-\alpha} \\
k_{t+1} &= \exp(b_t)(y_t - c_t) + (1-\delta) k_t \\
a_t &= \rho a_{t-1} + \tau b_{t-1} + \epsilon_t \\
b_t &= \tau a_{t-1} + \rho b_{t-1} + \nu_t
\end{aligned}
$$

- Code *example1.mod* in Dynare or *RBCPerturbation.ipynb* in *DifferentiableStateSpaceModels.jl*
    - Declares variables, params, starting value for nonlinear solver
    - Model then written as above
    - Declare order and run to simulate, get IRFs and moments

Stochastic Growth Model: IRFs for second order solution: epsilon



Stochastic Growth Model: IRFs for second order solution: nu



Conclusions

- Main advantage of perturbation is speed of coding
- Well-maintained public libraries ensure speed, extensibility, follow-up analyses
- Put together a model for your problem in a day or so
  - Estimate it also in a weekend
- Quickly iterate through models to check implications, compare against data, devise fixes and improvements
- Goal of computation is to answer economic question
  - Value comes from results, not methods
  - Use sophisticated methods when genuinely needed to produce answer

References

- Theory
  - Judd Ch 13, Ferández-Villaverde et al Handbook chapter
- Software
  - Dynare: extremely polished, full-featured. C++, optional Matlab, Julia interfaces
  - DifferentiableStateSpaceModels.jl: Adds autodiff features
  - Variants: Sims gensys, Uhlig Toolkit, Schmitt-Grohe-Uribe, DSGE.jl, etc
- Extensions
  - Continuous time: Sims (2002) "Solving linear rational expectations models"
  - Piecewise linear: Occbin (in Dynare)
  - ∞-dimensions: next class