Tutorial III

EABCN Training School

Christian Wolf

June 2018

Tutorial Objectives

1. Fast numerical differentiation: AD
2. Heterogeneous-agent perturbation methods in discrete time
   - Combine perturbation methods with EGP
3. Model reduction techniques (discrete + continuous time)
   - Reducing controls (e.g. value function)
   - Reducing states (e.g. the distribution)
4. Estimation of heterogeneous-agent models
   - VMA-based approach vs. state-space representation
   - Discrete time vs. continuous time

Outline

Outline

1. Automatic Differentiation

2. Perturbation Methods in Discrete Time
   - The Growth Model
   - A Heterogeneous-Agent Version

3. Model Reduction
   - Reducing Controls
   - Reducing Endogenous States

4. Estimation of Heterogeneous-Agent Models
   - SS vs. VMA
   - Likelihood Evaluation in Continuous Time

Differentiation Techniques

- Need to differentiate many times, and accurately
  - For PHACT: $60,000$ equations, $120,000 + x$ derivatives, each derivative requires many uses of the chain rule
- No chance for differentiation by hand or symbolic differentiation
- Passable alternative: finite differences
  - Need to evaluate equilibrium conditions $120,000(\times 2)$ times
  - Problems: pretty slow, often very inaccurate
- Our preferred solution: automatic differentiation
  - See overview and coding package on SeHyoun's webpage
  - How it works: every computer program is just a string of elementary operations with known derivatives, so get full derivative using chain rule
  - Why it's appealing: accurate to machine precision + fast

A Simple Example

1. Specify function of interest and evaluation point

$$\begin{aligned} f(x) &= x^2 \\ x_0 &= 10 \end{aligned}$$

2. Turn x_0 into automatic differentiation object

$$x_0 = \text{myAD}(x_0)$$

3. Apply automatic differentiation package

$$\begin{aligned} f(x_0) &= \text{getvalues}(f(x_0)) = 100 \\ f'(x_0) &= \text{getderivs}(f(x_0)) = 20 \end{aligned}$$

Outline

Outline

1. Automatic Differentiation

2. Perturbation Methods in Discrete Time
   - The Growth Model
   - A Heterogeneous-Agent Version

3. Model Reduction
   - Reducing Controls
   - Reducing Endogenous States

4. Estimation of Heterogeneous-Agent Models
   - SS vs. VMA
   - Likelihood Evaluation in Continuous Time

The Growth Model

- Household block

$$\begin{aligned} C_t^{-\gamma} &= \beta \mathbb{E}_t \left[(1 + r_{t+1}) C_{t+1}^{-\gamma} \right] \\ L_t &= 1 \\ A_t + C_t &= w_t L_t + (1 + r_t) A_{t-1} \end{aligned}$$

- Firm block

$$\begin{aligned} w_t &= (1-\alpha) e^{z_t} K_{t-1}^{\alpha} L_t^{-\alpha} \\ r_t - \delta &= \alpha e^{z_t} K_{t-1}^{\alpha-1} L_t^{1-\alpha} \\ Y_t &= e^{z_t} K_{t-1}^{\alpha} L_t^{1-\alpha} \\ I_t &= K_t - (1-\delta) K_{t-1} \end{aligned}$$

- Aggregation

$$\begin{aligned} A_t &= K_t \\ Y_t &= C_t + I_t \end{aligned}$$

- Exogenous process

$$z_t = \rho_z z_{t-1} + \sigma_z \epsilon_t^z$$

Outline

EGP in a Growth Model

- Firm block, aggregation, exogenous process unaffected
- Useful to split household block into two parts
  - Micro decisions

$$
\begin{aligned}
c_t(s) &= f\left(a(s), w_t, r_t, \text{emuc}_t\right) \\
\text{emuc}_t &= \mathbb{E}_t\left[c_{t+1}(s')\right] \\
\ell_t(s) &= y(s)
\end{aligned}
$$

  - Aggregation

$$
\begin{aligned}
C_t &= \int_{\mathcal{S}} c_t(s) d\lambda_t(s) \\
L_t &= \int_{\mathcal{S}} \ell_t(s) d\lambda_t(s) \\
A_t + C_t &= w_t L_t + (1 + r_t) A_{t-1}
\end{aligned}
$$

The Linearized Model

- Comparison to continuous time
  - Need to keep track of distribution and emuc object (rather than v)
  - Expectational errors only appear in updating of emuc term
- Linearization gives standard gensys form:

$$G_0 \begin{pmatrix} \text{emuc}_t \\ \lambda_t \\ x_t \end{pmatrix} = G_1 \begin{pmatrix} \text{emuc}_{t-1} \\ \lambda_{t-1} \\ x_{t-1} \end{pmatrix} + \Pi \eta_t + \Psi \varepsilon_t$$

  - x_t collects all the other aggregates: C_t, L_t, w_t, r_t, A_t
  - Usual structure: emuc_t and λ_t large-dimensional, x_t small-dimensional
- Let's take a look at some codes . . .

Outline

The Basic Idea

- We have the generic system

$$\begin{pmatrix} \dot{v}_t \\ \dot{\lambda}_t \\ \dot{x}_t \end{pmatrix} = G_1 \begin{pmatrix} v_t \\ \lambda_t \\ x_t \end{pmatrix} + \Pi \eta_t + \Psi \varepsilon_t \quad \text{or} \quad \begin{pmatrix} v_t \\ \lambda_t \\ x_t \end{pmatrix} = G_1 \begin{pmatrix} v_{t-1} \\ \lambda_{t-1} \\ x_{t-1} \end{pmatrix} + \Pi \eta_t + \Psi \varepsilon_t$$

  - This is slightly less general than canonical gensys: $G_0 = I$
  - We need this assumption: Schur decomposition vs. QZ decomposition, distribution reduction
  - Not a huge loss of generality (just need to solve out some linear relationships)

- Problem: v_t and λ_t are large-dimensional
- Idea: let's force them to live in subspaces of $\mathbb{R}^{n_v}$ and $\mathbb{R}^{n_\lambda}$, respectively
  - Project on small-dimensional subspaces with semi-orthogonal bases X_v, X_λ
  - This gives

$$\begin{aligned} v_t &= X_v(X_v' X_v)^{-1} X_v' v_t + \text{residual} = X_v \tilde{v}_t + \text{residual} \\ \lambda_t &= X_\lambda(X_\lambda' X_\lambda)^{-1} X_\lambda' \lambda_t + \text{residual} = X_\lambda \tilde{\lambda}_t + \text{residual} \end{aligned}$$

Applying Reduction

- Ignoring the projection errors this gives

$$v_t = X_v \tilde{v}_t, \qquad \tilde{v} = X_v' v_t$$
$$\lambda_t = X_\lambda \tilde{\lambda}_t, \qquad \tilde{v} = X_\lambda' \lambda_t$$

- Applying the reduction:

$$\begin{pmatrix} \dot{\tilde{v}}_t \\ \dot{\tilde{\lambda}}_t \\ \dot{x}_t \end{pmatrix} = X' G_1 X \begin{pmatrix} \tilde{v}_t \\ \tilde{\lambda}_t \\ x_t \end{pmatrix} + X' \Pi \eta_t + X' \Psi \varepsilon_t \text{ or } \begin{pmatrix} \tilde{v}_t \\ \tilde{\lambda}_t \\ x_t \end{pmatrix} = X' G_1 X \begin{pmatrix} \tilde{v}_{t-1} \\ \tilde{\lambda}_{t-1} \\ x_{t-1} \end{pmatrix} + X' \Pi \eta_t + X' \Psi \varepsilon_t$$

where

$$X = \begin{pmatrix} X_v & 0 & 0 \\ 0 & X_\lambda & 0 \\ 0 & 0 & I \end{pmatrix}$$

  - Side note: strictly speaking should also reduce expectational errors with $\tilde{\eta}_t^v = X_v \eta_t^v$, but irrelevant because only subspace $X' \Pi$ matters anyway
  - Reduction before differentiation reduces differentiation requirements

- Resulting system is just $n_{\tilde{v}} + n_{\tilde{\lambda}} + n_x$-dimensional

- Remaining question: how do we find X_v, X_λ?

Outline

Reducing Controls

- Controls are very amenable to simple spline-based reduction
    - Example: $v(a, z)$/emuc(a, z) are smooth in a given z and z given a
    - Spline reduction (e.g. cubic) using a coarse basis thus promising
- What we normally do
    - Choose $n_{\tilde{a}}$ spline points to reduce $n_a \times 1$ asset grid a
    - For each z apply spline reduction
    - Provides overall reduction from $n_a \times n_z$ to $n_{\tilde{a}} \times n_z$
- Could probably do better exploiting smoothness in z direction
- Bottom line: generally not a big deal

Outline

1. Automatic Differentiation

2. Perturbation Methods in Discrete Time
   - The Growth Model
   - A Heterogeneous-Agent Version

3. Model Reduction
   - Reducing Controls
   - Reducing Endogenous States

4. Estimation of Heterogeneous-Agent Models
   - SS vs. VMA
   - Likelihood Evaluation in Continuous Time

The Basic Idea

- Once solved the system will look like this:

$$\dot{s}_t = As_t + B\varepsilon_t, \qquad s_t = As_{t-1} + B\varepsilon_t$$
$$y_t = Cs_t + D\varepsilon_t, \qquad y_t = Cs_{t-1} + D\varepsilon_t$$

- Why there's scope for reduction
  - We are interested in the mapping from ε_t (small-dimensional) to y_t (small-dimensional) through s_t (large-dimensional)
  - The mapping from inputs to outputs is thus small-dimensional
  - Why should we need to carry around all the information in s_t? a subspace MUST be enough to characterize input-output linkage

- Logic of what I'll say works equally well for continuous and discrete time
  - Bayer et al. (2017): "In discrete time, there is no obvious basis for the state-space reduction"
  - To show that this is wrong I'll do discrete time today

Input-Output Linkage and Reduction

- In the full model we have

$$h_t = \begin{cases} D & \text{if } t = 0 \\ CA^{t-1}B & \text{if } t > 0 \end{cases}$$

- With reduced state vector $\tilde{s}_t = X_s' s_t$ we instead get

$$\tilde{h}_t = \begin{cases} D & \text{if } t = 0 \\ CX_sX_s'A^{t-1}X_sX_s'B & \text{if } t > 0 \end{cases}$$

- Result: if we choose X_s to be a semi-orthogonal basis of

$$\mathcal{O}_k(C, A) = \begin{pmatrix} C' \\ A'C' \\ (A')^2C' \\ \dots \\ (A')^{k-1}C' \end{pmatrix}$$

then $\tilde{h}_t = h_t$ for $t = 0, 1, \dots, k$ (proof in paper)

Making this practical . . .

- Problem is that we don't know the solved-out dynamics . . .

- First simple idea: just focus on λ-part of the system

$$\begin{aligned} \lambda_t &= G_1^{\lambda\lambda}\lambda_{t-1} + \Psi^{\lambda}\varepsilon_t + \text{rest} \\ x_t &= G_1^{x\lambda}\lambda_{t-1} + \Psi^{x}\varepsilon_t + \text{rest} \end{aligned}$$

  - Do reduction based on $A = G_1^{\lambda\lambda}$, $B = \Psi^{\lambda}$, $C = G_1^{x\lambda}$, $D = \Psi^{x}$
  - Not efficient, but will attain correct solution for large (but finite) k

- For efficiency gains

  1. Solve once using some starting guess
  2. Do correct reduction based on current solution
  3. Iterate until convergence

Outline

1. Automatic Differentiation

2. Perturbation Methods in Discrete Time
   - The Growth Model
   - A Heterogeneous-Agent Version

3. Model Reduction
   - Reducing Controls
   - Reducing Endogenous States

4. Estimation of Heterogeneous-Agent Models
   - SS vs. VMA
   - Likelihood Evaluation in Continuous Time

A Quick Review of Estimation

- Macro models are routinely estimated using a likelihood approach
  - Let's write the likelihood as
  $$p(\mathcal{Y} \mid \theta)$$
  where $\mathcal{Y}$ is data and θ is the parameter vector
  - Estimation then proceeds using ML or in some Bayesian fashion
- Likelihood evaluation proceeds in two steps
  1. Map parameters θ into an econometric model
     - Requires model solution (e.g. perturbation, MIT shock)
     - This gives state-space representation
     $$\begin{aligned} s_t &= A s_{t-1} + B\varepsilon_t \\ y_t &= C s_t \end{aligned}$$
     or vector moving-average representation
     $$y_t = \sum_{\ell=0}^{T_{\max}} \Theta_\ell \varepsilon_{t-\ell}$$
  2. Need to evaluate likelihood of model given data $\mathcal{Y}$

Challenges of Heterogeneous-Agent Modeling

- Solving big models
    - This was the point of the previous lectures/tutorials
    - Conclusion: toolkit should have perturbation and MIT shocks in both continuous and discrete time
- Likelihood evaluation with rich panel dimension
    - Raises special econometric difficulties
    - No time; exciting work coming up by Mikkel Plagborg-Moller
- Likelihood evaluation beyond small recursive discrete-time models
    - How to evaluate the likelihood in a VMA model?
    - How to evaluate the likelihood in SS/VMA in continuous time?
    - When should we prefer SS, when VMA?

Outline

SS vs. VMA

- Pros and cons of the state-space representation
  - Exploits the fact that mapping from ε_t to y_t has special structure going through s_t
  - With n_s small this is very convenient for likelihood evaluation via Kalman filtering $\rightarrow$ ubiquitous use in representative-agent literature
  - Maybe not as convenient in het-agent models where n_s is large
- Pros and cons of the VMA representation
  - System size is always large (e.g. $n_y \times n_\varepsilon \times 250$), but bounded above independent of micro heterogeneity
  - Reasonably fast likelihood evaluation feasible using Whittle likelihood approximation

Outline

Likelihood Evaluation in Continuous Time

- We consider the system

$$\dot{s}_t = As_t + B\varepsilon_t, \quad y_t = Cs_t$$

- Likelihood evaluation proceeds via a DT-CT hybrid Kalman filter
  1. Prediction step:

$$\dot{\hat{s}}_{k|t-1} = -A\hat{s}_{k|t-1} \quad \rightarrow \quad \hat{s}_{t|t-1} = e^{-A}\hat{s}_{t-1|t-1}$$

$$\dot{P}_{k|t-1} = -AP_{k|t-1} - P_{k|t-1}A' + \Omega \quad \rightarrow \quad p_{t|t-1} = e^{-\tilde{A}}p_{t|t-1} + \tilde{A}^{-1}(I - e^{-\tilde{A}})\omega$$

     where $\Omega \equiv BB'$ and $\tilde{A} \equiv (I \otimes A) + (A \otimes I)$

  2. Updating step: from the innovation $v_t = y_t - C\hat{s}_{t|t-1}$ we get the Kalman gain

$$K_t = P_{t|t-1}C'(CP_{t|t-1}C')^{-1}$$

     and so update

$$\begin{aligned} \hat{s}_{t|t} &= \hat{s}_{t|t-1} + K_t v_t \\ P_{t|t} &= P_{t|t-1} - K_t C P_{t|t-1} \end{aligned}$$

Thanks for your attention – Questions?

Appendix: General Stochastic Properties

- For the states we start with conditional distributions: from

$$s_t = e^{-At} s_0 + \int_0^t e^{A(u-t)} B \, dW_u$$

we get

$$\mathbb{E}[s_t | s_0] \equiv \mu_t^s = e^{-At} s_0, \quad \mathrm{Var}[s_t | s_0] \equiv \Sigma_t^s = \int_0^t e^{A(u-t)} \Omega e^{A'(u-t)} du$$

or simplifying a bit

$$\begin{aligned}
\mathrm{vec}(\Sigma_t^s) &= \mathrm{vec}\left(\int_0^t e^{A(u-t)} \Omega e^{A'(u-t)} du \right) \\
&= \int_0^t e^{(A \oplus A)(u-t)} \mathrm{vec}(\Omega) du \\
&= (A \oplus A)^{-1} \left(I - e^{-(A \oplus A)t} \right) \mathrm{vec}(\Omega)
\end{aligned}$$

- The stationary distribution is now easy:

$$\begin{aligned}
\mu^s &= 0 \\
\Sigma^s &= \mathrm{vec}^{-1} \left((A \oplus A)^{-1} \mathrm{vec}(\Omega) \right)
\end{aligned}$$

Appendix: General Stochastic Properties

- For covariance properties:

$$\begin{aligned} \text{Cov}[s_{t_1}, s_{t_2}|s_0] &\equiv \Sigma^s_{t_1,t_2} \\ &= \int_0^{t_1} e^{A(u-t_1)}\Omega e^{A'(u-t_2)} du \\ &= \Sigma^s_{t_1} e^{-A'(t_2-t_1)} \end{aligned}$$

  and so

$$\text{Cov}[s_t, s_{t+q}] \equiv \Sigma^s_q = \Sigma^s e^{A'q}$$

- Properties of y then follow immediately
  - Alternatively they can be computed directly from the VMA representation:

$$y_t = \int_0^{\infty} \Theta_s dW_{t-s}$$

    and so

$$\begin{aligned} \Sigma^y &= \int_0^{\infty} \Theta_s \Theta'_s ds \\ \Sigma^y_q &= \int_0^{\infty} \Theta_s \Theta'_{s+q} ds \end{aligned}$$