Heterogeneous Agent Macroeconomics: Methods and Applications

DIW Masterclass 2023

Ralph Luetticke
University of Tuebingen, UCL, CEPR, CfM, & Stone Centre

July 2023

Heterogeneous agents models with aggregate uncertainty

These models are computational demanding to solve

- The original Krusell and Smith (1997, 1998) algorithm is notoriously slow
- Therefore, many papers study transitions
- or are restricted to relatively simple household decisions

Heterogeneous agents models with aggregate uncertainty

These models are computational demanding to solve

- The original Krusell and Smith (1997, 1998) algorithm is notoriously slow
- Therefore, many papers study transitions
- or are restricted to relatively simple household decisions

- We depart from the Reiter (2002, 2009) perturbation method
- And (try to) provide an accessible algorithm that can deal with high-dimensional heterogeneity

Reiter (2002): Solve by perturbation

- Models can be written as a non-linear difference equation: $\mathbb{E}F(X_t, X_{t+1}, Y_t, Y_{t+1}, \varepsilon_{t+1}) = 0$

The heterogeneous agent model:

- that is function valued and
- needs to be linearized around the stationary equilibrium (StE)

Reiter (2002): Solve by perturbation

- Models can be written as a non-linear difference equation:
  $\mathbb{E}F(X_t, X_{t+1}, Y_t, Y_{t+1}, \varepsilon_{t+1}) = 0$

  The heterogeneous agent model:
- that is function valued and
- needs to be linearized around the stationary equilibrium (StE)

- Functions need to be approximated by finite dimensional objects (e.g. coefficients of polynomials, splines, etc.)
- We show how to do this in a smart efficient way

Course outline

Basics

- Background knowledge
- Standard incomplete markets model
- Perturbation approach

Solving SIM by perturbation

- Reiter method
- Bayer&Luetticke reduction method
- Comparison to MIT shock and Sequence Space solution

HANK models

- Estimation
- Applications

Course outline: Basics

Background Knowledge

- Consumption-Savings Problem
- Basic Numerical Tools:
  - Functional approximation
  - Some useful algebra
  - Root-finding
- Dynamic Economic Problems: Theory
- Dynamic Programming: Value function iteration, Policy Function Iteration, Endogenous Grid Method

Course outline: Basics

Standard incomplete markets model (w/o aggr. shocks)

- Application of Dynamic Programming to SIM
- Markov chains and stochastics in dynamic programs: Markov chains and ergodic distributions,Policy functions as Markov chains
- Stationary heterogeneous agent economies: Partial Equilibrium, General Equilibrium

Perturbation: Theory and Application

- Perturbation methods and solutions to linear difference equations
- Automatic differentiation
- Higher order perturbation solutions
- Here we use the RBC model as example

Course outline: Solving SIM by perturbation

Standard incomplete markets model with aggregate socks

- Reiter method: How to apply perturbation solution to SIM
- Bayer&Luetticke: State space reduction using Sparse-polynomials and Copula functions
- Comparison to MIT shock solution (Boppart et al, Auclert et al)

Here we use the Krusell-Smith model as example.

Course outline: HANK models

Applications

- Estimation of HANK models by full Information or IRF matching. Further model reduction techniques.
- Prototypical HANK model with one or two assets
- Applications: Estimation of drivers of inequality, fiscal policy and debt, small-open-economy HANK model

Resources

- Lecture slides
- Coding exercises
  - I provide templates for their solution in MATLAB and/or Julia.

Heterogeneous agents models with aggregate uncertainty

Available codes:

- **Perturbation with our reduction for estimating HANK models (Julia)**
  `https://github.com/BASEforHANK/`
- Perturbation vs MIT shock for KS model (Matlab)
  `https://github.com/ralphluet/KS_Perturbation_vs_MIT`
- Perturbation with our reduction for KS and HANK models (Matlab)
  `https://github.com/ralphluet/perturbation_codes`
- Perturbation with our reduction for HANK models (Python)
  `https://github.com/econ-ark/BayerLuetticke`

Literature - Academic articles

Perturbation:

- Reiter (2002, 2009), Ahn et al. (2018), Bayer and Luetticke (2020), and Bayer et al. (2019)
- ...

MIT shock:

- Boppart et al. (2018) and Auclert et al. (2019)
- ...

Global:

- Carroll (2006) and Hintermaier and Koeniger (2010)
- ...

Literature - Textbooks

- Heer, B. and A. Maussner (2009), "Dynamic General Equilibrium Modelling", 2nd edition, Springer, Berlin.
  Getting started: Ch. 1, 4, 7, 8
- Adda, J. and R. Cooper (2004): "Dynamic Economics", MIT Press, Cambridge.
  Partial Equilibrium only, Many Applications with Non-Convex Budget Sets.
- Ljungqvist, L. und T. Sargent (2012): "Recursive Macroeconomic Theory", 3rd ed., MIT press, Cambridge.
  Economic Theory Background
- Stockey, N.L. and Lucas, R.E. with E.C. Prescott (1989): "Recursive Methods in Economic Dynamics", Chapters 4 and 9, Havard University Press, Cambridge.
  Mathematical Background to Dynamic Programming

- An excellent quantitative econ source (for Python and Julia though): lectures.quantecon.org by Tom Sargent et al.

Theoretical Foundations:

Consumption-Savings Problem

Self insurance / savings problem

Assume a forward looking, rational consumer

- who wants to smooth consumption in the presence of **income risk**,
- has **no access to insurance markets** at all,
- and can only purchase non-negative amounts of a risk-free asset.

Self insurance

What is self insurance?

- Self insurance is to draw on savings to smooth negative income shocks.

Implications

- Using self insurance:
  - Will consumption converge to a finite positive amount,
  - ot to zero?
  - Or will it diverge or to infinity?

We will find

- Non-stochastic income: Convergence to a finite positive level
- Stochastic income: Divergence to infinity

A model of self insurance

Physical Environment

- Households receive an income stream $\{y_t\}_{t=0}^{\infty}$ and $y_t \in Y$.
- Y is a set of S discrete income levels $\bar{y}_s$ to avoid integration.
- Income is i.i.d. with probabilities $P(y_t = \bar{y}_s) = \pi_s$.

A model of self insurance

Planning Problem

- Households want to maximize

$$E_0 \sum_{t=0}^{\infty} \beta^t u(c_t)$$

where $\beta \in (0,1)$ and $u \in C^2$ is a strictly concave function.

Financial Markets

- There is a single risk-free asset a_t bearing interest $r = \frac{1-\beta}{\beta}$.

A model of self insurance

Reducing the number of state variables

- Define cash-at-hand a (financial wealth + all income), this evolves as:

$$a' = (1+r)(a-c) + y'.$$

- Nonnegative asset holdings is equivalent to $c \leq a$.

Recursive Planning Problem

- Use cash-at-hand to write the problem as a Bellman equation

$$V(a) = \max_{0 \leq c \leq a} \left\{ u(c) + \beta \sum_{s=1}^{S} \pi_s V\left[(1+r)(a-c) + \bar{y}_s\right] \right\}$$

Stochastic Income: Deriving the PIH

A first simple setup

- Before moving to the general case of stochastic income, we consider quadratic preferences without borrowing constraints
- This allows us to derive a form of the permanent income hypothesis

Preferences and financial markets

Quadratic preferences

- Households have a felicity function

$$u(c) = -\frac{1}{2}(c - \gamma)^2$$

And a loose borrowing limit

- We impose a long run no-debt constraint

$$E_0 \lim_{t \to \infty} \beta^t b_t^2 = 0$$

but allow households to have negative consumption.

Implications for borrowing

Budget constraints

- The period budget constraint is

$$c_t + \frac{1}{1+r} b_{t+1} \leq y_t + b_t$$

- Solving forward, we obtain that financial wealth is the present value of the difference between consumption and income

$$b_t = \sum_{j=0}^{\infty} \beta^j \left(c_{t+j} - y_{t+j} \right).$$

What does that mean for the intertemporal allocation?

No constraint on consumption smoothing

- The absence of a period-by-period borrowing constraint implies

$$u_c\left(c_t\right) = E_t\left[u_c\left(c_{t+1}\right)\right].$$

- Of course the consolidated budget constraint binds for the present value of total consumption.

Consumption is a martingale

Optimal policy

- With linearity in marginal utility this implies

$$E_t\left(c_{t+1}\right) = c_t.$$

- In other words: Consumption is a martingale.
- This holds for an arbitrary stationary process for income y_t.

The optimal consumption policy

Iterating the budget constraint forward

- Taking expectations on the budget constraint we obtain

$$b_t =$$

The optimal consumption policy

Iterating the budget constraint forward

- Taking expectations on the budget constraint we obtain

$$b_t = E_t b_t$$

The optimal consumption policy

Iterating the budget constraint forward

- Taking expectations on the budget constraint we obtain

$$b_t = E_t b_t =$$

The optimal consumption policy

Iterating the budget constraint forward

- Taking expectations on the budget constraint we obtain

$$b_t = E_t b_t = E_t \sum_{j=0}^{\infty} \beta^j \left(c_{t+j} - y_{t+j}\right)$$

$$=$$

The optimal consumption policy

Iterating the budget constraint forward

- Taking expectations on the budget constraint we obtain

$$\begin{aligned} b_t &= E_t b_t = E_t \sum_{j=0}^{\infty} \beta^j \left(c_{t+j} - y_{t+j}\right) \\ &= E_t \sum_{j=0}^{\infty} \beta^j c_{t+j} - E_t \sum_{j=0}^{\infty} \beta^j y_{t+j} = \end{aligned}$$

The optimal consumption policy

Iterating the budget constraint forward

- Taking expectations on the budget constraint we obtain

$$\begin{aligned} b_t &= E_t b_t = E_t \sum_{j=0}^{\infty} \beta^j \left(c_{t+j} - y_{t+j}\right) \\ &= E_t \sum_{j=0}^{\infty} \beta^j c_{t+j} - E_t \sum_{j=0}^{\infty} \beta^j y_{t+j} = \frac{1}{1-\beta} c_t - E_t \sum_{j=0}^{\infty} \beta^j y_{t+j}. \end{aligned}$$

The optimal consumption policy

Iterating the budget constraint forward

- Taking expectations on the budget constraint we obtain

$$\begin{aligned} b_t &= E_t b_t = E_t \sum_{j=0}^{\infty} \beta^j \left(c_{t+j} - y_{t+j} \right) \\ &= E_t \sum_{j=0}^{\infty} \beta^j c_{t+j} - E_t \sum_{j=0}^{\infty} \beta^j y_{t+j} = \frac{1}{1-\beta} c_t - E_t \sum_{j=0}^{\infty} \beta^j y_{t+j}. \end{aligned}$$

We obtain the optimal consumption policy

$$c_t = \frac{r}{1+r} \left[b_t + E_t \sum_{j=0}^{\infty} (1+r)^{-j} y_{t+j} \right].$$

The permanent income hypothesis holds

Summary

In the absence of borrowing constraints and with quadratic preferences we obtain Friedman's permanent income hypothesis.

Households consume a constant fraction (the annuity value) of human and non-human wealth.

The evolution of financial wealth

The inter-temporal budget constraint is

$$b_{t+1} = (1+r)\left(y_t - c_t + b_t\right).$$

Plugging in optimal consumption yields

$$b_{t+1} \quad = \quad (1+r)\left(y_t - \frac{r}{1+r}\left[b_t + E_t \sum_{j=0}^{\infty} (1+r)^{-j} y_{t+j}\right] + b_t\right)$$

The evolution of financial wealth

The inter-temporal budget constraint is

$$b_{t+1} = (1+r)\left(y_t - c_t + b_t\right).$$

Plugging in optimal consumption yields

$$b_{t+1} \quad = \quad (1+r)\left(y_t + \frac{1}{1+r}b_t - \frac{r}{1+r}E_t\sum_{j=0}^{\infty}(1+r)^{-j}y_{t+j}\right)$$

The evolution of financial wealth

The inter-temporal budget constraint is

$$b_{t+1} = (1+r)\left(y_t - c_t + b_t\right).$$

Plugging in optimal consumption yields

$$b_{t+1} \quad = \quad b_t + (1+r)y_t - r\sum_{j=0}^{\infty}(1+r)^{-j}E_t y_{t+j}$$

Wealth and thus consumption follow unit root processes!

Consumption and wealth are cointegrated

However, rearranging terms in the consumption formula, we obtain

$$c_t - \frac{r}{1+r} b_t = \frac{r}{1+r} E_t \sum_{j=0}^{\infty} (1+r)^{-j} y_{t+j}.$$

where the rhs is stationary (if y is stationary).

Thus, consumption and wealth are co-integrated unit-root processes.

Taking stock

Summary

- *Both c and b are unit-root processes that are co-integrated.*
- *Consumption only depends on the conditional first-moment of the discounted value of endowment $\leftrightarrow$ certainty equivalence under quadratic dynamic programming.*
- *There is no drift in consumption.*

Why care?

PIH has strong implications

- Timing of (lump-sum) taxes and transfers does not matter.
- (All macroeconomic) policy works through substitution effects.
- Income risk and inequality does not matter for aggregates.

Why important?

- Any first order approximation to the Euler equation is equivalent to quadratic preferences.

General preferences

Next we consider uncertain i.i.d. endowments with general utility functions that satisfy $u' > 0,\ u'' < 0$.

Bellman equation

$$V(a) = \max_{0 \leq c \leq a} u(c) + \sum_{s=1}^{S} \beta \pi_s V\left[(1+r)(a-c) + \bar{y}_s\right]$$

The first order condition reads

$$u_c(c) \geq \sum_{s=1}^{S} \beta (1+r) \pi_s V_a\left[(1+r)(a-c) + \bar{y}_s\right].$$

with equality if $a > c$ (b is positive)

The marginal value of cash is a supermartingale

Now using that $u_c(c) = V_a(a)$ and $\beta(1+r) = 1$ we obtain

$$V_a(a) \geq \sum_{s=1}^{S} \pi_s V_a(a'_s)$$

- where a'_s is the next periods cash-at-hand in case of income $\bar{y}_s$
- and V_a a short-hand notation for $\frac{\partial V}{\partial a}$.

This means that $V_a(a)$ is a supermartingale — the rhs is $E_t[V_a(a_{t+1})]$.

Martingales

Definition

Let the elements of the 3-tuple (Ω, F, P) denote the sample space, a collection of events (Information set) and a probability measure respectively. Let $t \in \mathbb{N}$ denote index time and F_s an increasing sequence of σ-fields of F-sets. Suppose that

1. Z_t is measurable with respect to F_t;
2. $E|Z_t| < +\infty$;
3. $E(Z_t|F_s) = Z_s$ almost surely for all $s < t$; $s, t \in \mathbb{N}$

   Then $\{Z_t, t \in \mathbb{N}\}$ is said to be a **martingale** with respect to F_t. If (3) is replaced by $E(Z_t|F_s) \geq Z_s$ almost surely, then $\{Z_t\}$ is said to be a **submartingale.** If $E(Z_t|F_s) \leq Z_s$ then $\{Z_t\}$ is said to be a **supermartingale.**

Supermartingale Convergence Theorem

Theorem

Let $\{Z_t, F_t\}$ be a non-negative supermartingale. Then there exists a random variable Z, such that $\lim_{t \to \infty} Z_t = Z$ almost surely and $E|Z| < +\infty$. In words: Z_t converges to a finite limit almost surely.

Assets diverge

The marginal value of assets converges to zero

- From the property of V_a being a supermartingale, we can conclude that $\bar{V}_a := \lim_{t\to\infty} V_a(a_t)$ exists, is finite and non-negative.
- This limit can be shown (see next slide) to be zero.

Hence assets diverge to infinity a.s., $a \to_{a.s.} +\infty$.

Assets diverge

- Suppose on the contrary that $\bar{V}_a > 0$.
- Since V is strictly concave (as u is strictly concave), this implies that a_t converges to a positive finite value $\bar{a}$ a.s..

Divergence of assets ... continued

Yet, this cannot be the case:

- For a finite limit for a the Budget constraint implies that for any ϵ there should be a large enough t such that $|(1+r)(\bar{a}-c_t)+y_{t+1}-\bar{a}|<\epsilon$ a.s..
- Yet, this contradicts the stochastic nature of income.

Taking stock

As the assets diverge for $1 = (1+r)\beta$

- the interest rate must be lower than the time preference rate in equilibrium.
- There is supposedly a link between the importance and ability to self-insure and the market rate.

Summary of Consumption Behavior

- Stochastic case: There is an additional motive for saving, the *precautionary motive*, due either to
  - prudence;
  - borrowing constraint.
- Intuitively, conditions under which assets converge will be more stringent.
- Assets dynamics in income fluctuation problem: stochastic case.

	Deterministic Income	Stochastic Income
$\beta(1+r) > 1$	Diverging	Diverging
$\beta(1+r) = 1$	Stationary	Diverging
$\beta(1+r) < 1$	Stationary	Stationary[1]

[1] Under mild conditions.

A particular savings problem

Notation, wages are endogenous

- There are m levels of human capital s_i (e.g. ability, hours of work).
- Example: unemployed $\mathrm{s}_0 = 0$ or employed $\mathrm{s}_1 = 1$.
- An household obtains income ws_t.
- $s_t \in \{\mathrm{s}_0, \ldots \mathrm{s}_n\}$ follows a Markov chain w/ transition probability Π.

Discrete asset choices for simplicity

- The household can hold assets in amounts given by a grid $B = \{\mathrm{b}_1, \ldots, \mathrm{b}_n\}$, where $\mathrm{b}_{j-1} < \mathrm{b}_j,\ 0 \in B$.
- Then assets and income evolve jointly as a discrete Markov Chain.

A particular savings problem

Prices remain constant over time

- because we look at the stationary distribution of assets, i.e. the cross-sectional distribution of s and a is its ergodic distribution.
- W.l.o.g. $\beta(1+r) < 1$.

Planning problem

- The household chooses c_t (and thus b_t) in order to maximize

$$E_0 \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$s.t.\ c_t + \frac{b_{t+1}}{1+r} = b_t + ws_t;\ b_{t+1} \in B$$

A particular savings problem

The Bellman equation:
Choosing next periods wealth

$$V(s,b) = \max_{b' \in B} u\left(b + ws - \frac{b'}{1+r}\right) + \beta E_s V(s', b')$$

Equivalent: Index notation

$$V(i,j) = \max_{k \in \{1 \ldots n\}} u\left(b_j + w\bar{s}_i - \frac{b_k}{1+r}\right) + \beta \sum_{l=1}^{m} \Pi(i,l) V(l,k)$$

A particular savings problem

Policy function

- In "natural" notation:

$$b' = g_{nat}\left(s, b | r, w\right).$$

  map assets and productivity into new assets.

- In index notation (useful for computation):

$$k = g_{ind}\left(i, j | r, w\right).$$

  map asset and productivity indexes into new asset indexes.

- The conditionality reflects that households take prices as given.

Ex-post heterogeneity

Policy is not linear, the wealth and income distribution impacts on prices.

Wealth and income distribution

- Now denote $\mu_t(b,s) = \Pr(b_t = b, s_t = s)$.

and its evolution

- The exogenous Markov chain Π and g induce a law of motion

$$\mu_{t+1}(s', b') = \sum_{s \in S} \sum_{\{b | b' = g(s,b|r,w)\}} \mu_t(s,b) \Pi(s'|s).$$

For given prices, optimal household policy implies

- A law of motion, $\Gamma_{r,w}$, mapping μ to μ'

$$\mu' = \mu\Gamma_{r,w}$$

(Kolmogorov forward / Fokker-Planck equation in discrete time).

Stationary distribution

- A stationary distribution $\bar{\mu}_{r,w}$ is a fixed point of this mapping
- and $\bar{\mu}_{r,w}$ is the unit-eigenvector of $\Gamma_{r,w}$.

Forward and backward equations

A side remark is warranted

- The distribution follows the forward equation

$$\mu' = \mu\Gamma_{r,w}$$

- but the Bellman equation can also be expressed similarly:

$$V = U_{r,w} + \beta\Gamma_{r,w}V'$$

as a forward equation with V and $U = u\left[g_{cons}(i,j|r,w)\right]$ being vectorized value functions and payoffs under the optimal policy.

Standard Incomplete Markets Model (SIM): Setup

Recursive Dynamic Planning Problem

Consider a household problem in presence of aggregate and idiosyncratic risk

- S_t is an (exogenous) aggregate state
- s_{it} is a partly endogenous idiosyncratic state
- μ_t is the distribution over s

- Bellman equation:

$$v(s_{it}, S_t, \mu_t) = \max_{x \in \Gamma(s_{it}, P_t)} u(s_{it}, x) + \beta \mathbb{E} v(s_{it+1}(x, s_{it}), S_{t+1}, \mu_{t+1})$$

Recursive Dynamic Planning Problem

Consider a household problem in presence of aggregate and idiosyncratic risk

- S_t is an (exogenous) aggregate state
- s_{it} is a partly endogenous idiosyncratic state
- μ_t is the distribution over s
- Euler equation:

$$u'[x(s_{it}, S_t, \mu_t)] = \beta R(S_t, \mu_t)\mathbb{E}u'[x(s_{it+1}, S_{t+1}, \mu_{t+1})],$$

No aggregate risk

Recall how to solve for a stationary equlibrium:

- Discretize the state space (vectorized)
- Optimal policy $\bar{h}(s_{it}; P)$ induces flow utility $\bar{u}_{\bar{h}}$ and transition probability matrix $\Pi_{\bar{h}}$

No aggregate risk

- Discretized Bellman equation

$$\bar{v} = \bar{u}_{\bar{h}} + \beta \Pi_{\bar{h}} \bar{v} \tag{1}$$

holds for optimal policy (assuming a linear interpolant for the continuation value)

- and for the law of motion for the distribution (histograms)

$$d\bar{\mu} = d\bar{\mu} \Pi_{\bar{h}} \tag{2}$$

No aggregate risk

Equilibrium requires

- $\bar{h}$ is the optimal policy given P and v (being a linear interpolant)
- $\bar{v}$ and $d\bar{\mu}$ solve (13) and (14)
- Markets clear (some joint requirement on $\bar{h}, \mu, P$, denoted as $\Phi(\bar{h}, \mu, P) = 0$)

This can be solved for efficiently

- $d\bar{\mu}$ is vector corresponding to the unit-eigenvalue of $\Pi_{\bar{h}}$
- Using fast solution techniques for the DP, e.g. EGM
- Using a root-finder to solve for P

Stationary Equilibrium

Definition

A recursive (stationary) competitive equilibrium is an allocation (c, a'), a r^*, and an invariant distribution $\mu^* = \mu(s, a; r^*)$ such that:

1. For given r^*, (c, a') solves the household optimization, $V = TV$.
2. Given μ^*, goods and asset markets clear:

$$\sum_{a \in A} \sum_{s \in S} \left[c(s, a; r^*) - y(s) \right] \mu^*(s, a) = 0 \tag{3}$$

$$\sum_{a \in A} \sum_{s \in S} a'(s, a; r^*) \mu^*(s, a) = 0 \tag{4}$$

3. μ^* is a stationary probability measure consistent with $a' = g(s, a; r^*)$ and $\pi(s'|s)$.

Equilibrium

- Market clearing depends on the specific model

Equilibrium

- Market clearing depends on the specific model
- In a Huggett model, aggregate bond supply is zero and

$$K^S(r) = 0$$

  is the equilibrium condition.

Equilibrium

- Market clearing depends on the specific model
- In a Huggett model, aggregate bond supply is zero and

$$K^S(r) = 0$$

  is the equilibrium condition.
- In an Aiyagari model, we require that

$$r + \delta = F_K(K, L), w = F_L(K, L)$$

  where in the most simple case aggregate labor supply is exogenously given. Then, prices are only a function of K and the equilibrium condition is simply

$$K^S(P(K)) = K$$

The Aiyagari (1994) Model

- There is a large number of firms producing output from capital and labor with technology $Y = F(K, L)$. Capital depreciates at rate δ.
- Aggregate labor supply is $E(e)$. Equilibrium in the labor market requires $L = E(e)$.
- For given aggregate capital, firms' optimization implies

$$r = F_k(K, L) - \delta$$
$$w = F_L(K, L).$$

- It is useful to invert the first equation as $K = K(r)$ and to substitute it into the second, $w = w(K(r))$. Both are decreasing functions.

Asset Demand

- Write $A(z;r,w)$ and $G(z;r,w)$ to express the dependence of the policy function and of the invariant distribution on (r,w).
- Aggregate asset demand:

$$\mathcal{A}(r) \equiv \int A(z;r,w(K(r)))dG(z;r,w(K(r))) - \Phi \ .$$

- $\mathcal{A}(r)$ is typically increasing (this is not necessary, however). However, $\mathcal{A}(r) \rightarrow \infty$ when r converges to $\frac{1}{\beta} - 1$ from below.
  - We proved before that if $r = \frac{1}{\beta} - 1$, then assets diverge to $+\infty$ almost surely.

Stationary Equilibrium

A stationary competitive equilibrium is value function $V(z)$, policy function $A(z)$, distribution function $G(z)$, aggregate capital K, an interest rate r and real wage w such that

1. Given w and r, $V(\cdot)$ and $A(\cdot)$ are value and policy functions of the household problem.
2. Given w and r, firms choose labor $L = E(e)$ and capital K optimally: $K = K(r)$ and $w = w(K(r))$.
3. $G(\cdot)$ is an invariant distribution measure consistent with policy function $A(\cdot)$ and with exogenous distribution of shock e.
4. Capital market clearing: $K(r) = \mathcal{A}(r)$.
5. Goods market clearing: $C + \delta K = Y$, where C is aggregate consumption.

Equilibrium: Existence and Uniqueness

- Existence and uniqueness boils down to one equation in one unknown: $K(r) = \mathcal{A}(r)$.
- By Walras' law ignore goods market condition.
- Labor market equilibrium $L = E(e)$ and $\bar{L} \equiv E(e)$ is exogenously given.

Equilibrium: Existence and Uniqueness

- Existence and uniqueness boils down to one equation in one unknown: $K(r) = \mathcal{A}(r)$.
- By Walras' law ignore goods market condition.
- Labor market equilibrium $L = E(e)$ and $\bar{L} \equiv E(e)$ is exogenously given.
- Capital/asset market clearing condition:

$$K(r) = \int A(z;r)dG(z;r) - \Phi \equiv \mathcal{A}(r)$$

- Capital demand of firm $K(r)$ is defined implicitly as

$$r = F_k(K(r), \bar{L}) - \delta$$

- Given assumptions on $F(K, L)$, it follows that $K(r)$ is continuous, strictly decreasing on $r \in (-\delta, \infty)$ with

$$\lim_{r \to -\delta} K(r) = \infty, \quad \lim_{r \to \infty} K(r) = 0.$$

Equilibrium: Existence and Uniqueness, cont'd

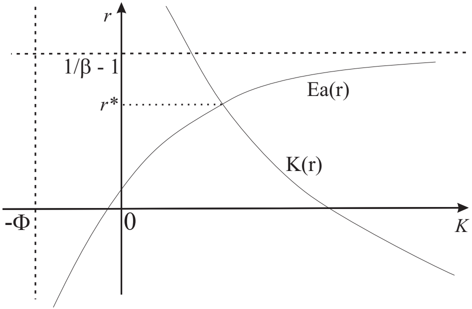
- Now characterization of capital supply (or aggregate savings) $\mathcal{A}(r)$.
- $\mathcal{A}(r) \in [-\Phi, \infty]$ for all $r \in [-\delta, \frac{1}{\beta} - 1]$.
- Under some restrictions, one can prove that the function $\mathcal{A}(r)$ is well-defined on $r \in [-\delta, \frac{1}{\beta} - 1)$. (See previous analysis.)
- Furthermore,
$$\lim_{r \to -\delta} \mathcal{A}(r) < \infty, \quad \lim_{r \to \frac{1}{\beta} - 1} \mathcal{A}(r) = \infty.$$
- Then there exists r^* such that
$$K(r^*) = \mathcal{A}(r^*).$$

Equilibrium: Existence and Uniqueness, cont'd

- Define an excess demand function $ED(r) \equiv K(r) - \mathcal{A}(r)$.
- We proceed in three steps:
  1. There exists $\underline{r} < 0$ such that $ED(\underline{r}) > 0$. Indeed, for r sufficiently low, $K(r) \to \infty$ and $\mathcal{A}(r)$ is finite, hence **capital is in excess demand**.
  2. There exists $\bar{r} > 0$ such that $ED(\bar{r}) < 0$. Indeed, for $r \to \frac{1}{\beta} - 1$ from below, $K(r)$ is finite and $\mathcal{A}(r)$ becomes arbitrarily large, so that **capital is in excess supply**.
  3. Since $ED(\cdot)$ is continuous, by the Intermediate Value Theorem there exists r^* such that $ED(r^*) = 0$.
- Is market clearing r^* unique?

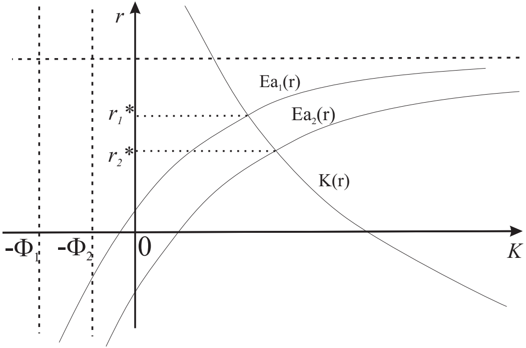
Equilibrium in the Capital Market

The equilibrium has a lower interest rate and a higher capital stock than the model with complete markets. [*Notes:* $Ea(r)$ denotes the capital supply/aggregate savings function $\mathcal{A}(r)$. $K(r)$ is capital demand function. $r^{CM} = \frac{1}{\beta} - 1 > r^*$.]



Comparative Statics

- Tightening the borrowing constraint (reducing Φ), increases $\mathcal{A}(r)$, hence r falls and $K = K(r)$ rises.



Sequential Equilibrium

Definition

Given an initial allocation μ^*, a recursive (along the transition) competitive equilibrium is a sequence of $\{c_t, a'_t\}_{t=0}^{\infty}$, of prices $\{r_t\}_{t=0}^{\infty}$, and of distributions $\{\mu_t\}_{t=0}^{\infty}$ such that:

1. For given r_t, (c_t, a'_t) solves the household optimization, $V = TV$.
2. Given μ_t, goods and asset markets clear:

$$\sum_{a \in A} \sum_{s \in S} [c(s, a; r_t) - y(s)] \mu_t(s, a) = 0 \tag{5}$$

$$\sum_{a \in A} \sum_{s \in S} a'(s, a; r_t) \mu_t(s, a) = 0 \tag{6}$$

3. μ_t is a probability measure consistent with $a'_t = g(s, a; r_t)$ and $\pi(s'|s)$.

Recursive Competitive Equilibrium

Definition

A recursive competitive equilibrium are policy functions $\{c_t, a_t'\}_{t=0}^{\infty}$, pricing functions $\{r_t\}_{t=0}^{\infty}$, and distributions $\{\mu_t\}_{t=0}^{\infty}$ such that:

1. Giving the pricing functions $r_t(Z_t, \mu_t)$ and the law of motion for μ_t, (c_t, a_t') solves the household optimization, $V = TV$.
2. Given μ_t, goods and asset markets clear:

$$\sum_{a \in A} \sum_{s \in S} \left[c\left(s, a; r_t\right) - y\left(s\right)\right] \mu_t\left(s, a\right) = 0 \tag{7}$$

$$\sum_{a \in A} \sum_{s \in S} a'\left(s, a; r_t\right) \mu_t\left(s, a\right) = 0 \tag{8}$$

3. μ_t is a probability measure consistent with $a_t' = g\left(s, a; r_t\right)$ and $\pi\left(s' | s\right)$.

Reminder

Basic Numerical Tools

Function Approximation

- Since computers are discrete machines, it is often necessary to rewrite a problem by an approximation using a finite number of parameters.
- For this purpose, it is often useful to represent a function with a (low-dimenional) vector of parameters.

Interpolation

- Let x_i, $i = 1...N$ be points at which we know $f_i = f(x_i)$. We want to approximate the function f for off-grid points.
- There is two aspects of the problem: First, how to choose the function $\hat{f}$ that represents f. Second, how to choose x_i (if we can).

Global polynomial

- One method to approximate a function is to express it by the coefficients ψ of a polynomial

$$\hat{f}(x) = \sum_{j=1}^{n} \psi_j c_j(x)$$

where $c_i(x)$ are known basis functions such as $c_j(x) = x^j$.

Global polynomial

- One method to approximate a function is to express it by the coefficients ψ of a polynomial

$$\hat{f}(x) = \sum_{j=1}^{n} \psi_j c_j(x)$$

where $c_i(x)$ are known basis functions such as $c_j(x) = x^j$.

- Better than ordinary polynomials are usually Chebyshev polynomials of which the baseline functions are

$$c_j(x) = \cos(j \arccos x)$$

Global polynomial

- One method to approximate a function is to express it by the coefficients ψ of a polynomial

$$\hat{f}(x) = \sum_{j=1}^{n} \psi_j c_j(x)$$

where $c_i(x)$ are known basis functions such as $c_j(x) = x^j$.

- Better than ordinary polynomials are usually Chebyshev polynomials of which the baseline functions are

$$c_j(x) = \cos(j \ \arccos x)$$

- These are orthogonal on [-1,1], i.e.

$$\int_{-1}^{1} c_i(x) c_j(x) \frac{1}{\sqrt{1-x^2}} dx = 0 \forall i \neq j$$

Global polynomial

- Since the evaluation points x_i are known ("grid"), we can compute $\mathbf{C} = [c_j(x_i)]_{i=1...M, j=1...n}$

Global polynomial

- Since the evaluation points x_i are known ("grid"), we can compute $\mathbf{C} = [c_j(x_i)]_{i=1...M, j=1...n}$
- The vector of function values $\hat{\mathbf{f}} = [\hat{f}(x_i)]_{i=1...M}$ is then given by

$$\hat{\mathbf{f}} = \mathbf{C}\psi$$

Global polynomial

- Since the evaluation points x_i are known ("grid"), we can compute $\mathbf{C} = [c_j(x_i)]_{i=1...M, j=1...n}$
- The vector of function values $\hat{\mathbf{f}} = [\hat{f}(x_i)]_{i=1...M}$ is then given by

$$\hat{\mathbf{f}} = \mathbf{C}\psi$$

- Therefore, we can obtain an optimal (minimal MSE) as

$$\psi^* = (\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}'\ \mathbf{f}$$

Global polynomial

- Since the evaluation points x_i are known ("grid"), we can compute $\mathbf{C} = [c_j(x_i)]_{i=1...M,j=1...n}$
- The vector of function values $\hat{\mathbf{f}} = [\hat{f}(x_i)]_{i=1...M}$ is then given by

$$\hat{\mathbf{f}} = \mathbf{C}\psi$$

- Therefore, we can obtain an optimal (minimal MSE) as

$$\psi^* = (\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}'\ \mathbf{f}$$

- The big **advantage** of polynomials is that they can be integrated analytically and that they are differentiable of any order.

Global polynomial: issues

- **Calculation of $(\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}$:** $\mathbf{C}$ depends on the choice of the basis functions and on the choice of the grid. Easily, C can be ill-conditioned (think of the regular polynomial $1, x, x^2 \ldots$). Then the inversion becomes imprecise.

Global polynomial: issues

- **Calculation of $(\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}$: $\mathbf{C}$** depends on the choice of the basis functions and on the choice of the grid. Easily, C can be ill-conditioned (think of the regular polynomial $1, x, x^2 \ldots$). Then the inversion becomes imprecise.
- If the number of grid points is larger than the order of the polynomial, it is not guaranteed that $\hat{\mathbf{f}} = \mathbf{f}$.

Global polynomial: issues

- **Runge's Phenomenon:** Since polynomials tend to infinity as $x \to \infty$ it is not true that the overall fit of a global polynomial gets better, if more grid points and higher order polynomials are used (oscillating behavior).

- Choosing **Chebyshev polynomials** as basis functions and
- grid points as the **roots** $x_i = \cos(\frac{2i-1}{2N})$ for $i = 1 \dots N$ of these polynomials minimizes approximation error.

Discrete Cosine Transforms

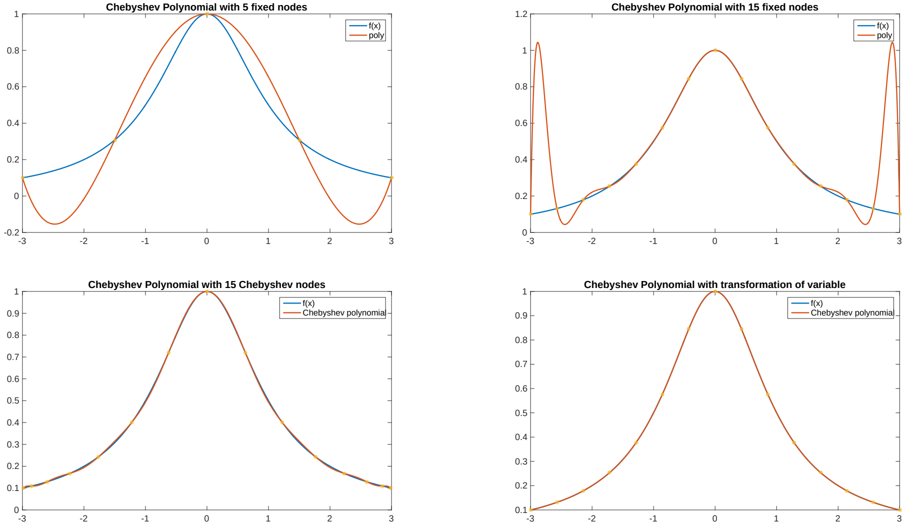
A first observation

- Suppose Chebychev root grid-points are not suitable for our problem.
- Then, we can write $f(x) = f(g(y))$ and
- generate the grid x_i by applying g to the Chebyshev nodes y_i,
- with basis functions $c_j(x) = \cos\left(j \ \arccos g^{-1}(x)\right)$

Discrete Cosine Transform (DCT) and lossy compression

- In particular, if we do not intend to evaluate off-grid, we do not need to know g but just the nodes $y_i = \cos\left(\frac{2i-1}{2N}\pi\right)$ and grid values x_i
- and obtain an equivalent representation of f_i in terms of coefficients.
- Shrinking ≈ 0-coefficients to 0 leaves $\hat{f}_i$ close to unchanged.
- In addition $C'C = I$.

Polynomials in practice



Interpolation

- A computationally easy alternative is to fit a piecewise function to the data.

Interpolation

- A computationally easy alternative is to fit a piecewise function to the data.
- For each point $x_i, f(x_i)$ we require that the interpolation function $\hat{f}_i = f_i$.

Interpolation

- A computationally easy alternative is to fit a piecewise function to the data.
- For each point $x_i, f(x_i)$ we require that the interpolation function $\hat{f}_i = f_i$.
- The interpolation function then generates $x, \hat{f}$ for off-grid points as $f_{i-1} + c_{i-1}(x)$ where x_{i-1} is the next smaller grid point relative to x and $c_i(0) = 0$.

Interpolation

- A computationally easy alternative is to fit a piecewise function to the data.
- For each point $x_i, f(x_i)$ we require that the interpolation function $\hat{f}_i = f_i$.
- The interpolation function then generates $x, \hat{f}$ for off-grid points as $f_{i-1} + c_{i-1}(x)$ where x_{i-1} is the next smaller grid point relative to x and $c_i(0) = 0$.
- In other words, the function $\hat{f}$ is piece-wise defined. and c_i is zero outside the interval $[x_i, x_{i+1}]$.

Interpolation

- A computationally easy alternative is to fit a piecewise function to the data.
- For each point $x_i, f(x_i)$ we require that the interpolation function $\hat{f}_i = f_i$.
- The interpolation function then generates $x, \hat{f}$ for off-grid points as $f_{i-1} + c_{i-1}(x)$ where x_{i-1} is the next smaller grid point relative to x and $c_i(0) = 0$.
- In other words, the function $\hat{f}$ is piece-wise defined. and c_i is zero outside the interval $[x_i, x_{i+1}]$.
- (Particular but relevant case: nearest neighbor interpolation: $c_i = 0$)

Interpolation in MATLAB

- Interpolation in MATLAB is easy.

Interpolation in MATLAB

- Interpolation in MATLAB is easy.
- First define a mesh-grid **M**
  M={GRID.X1,GRID.X2,...,GRID.XN}
  then a function
  FHAT = GRIDDEDINTERPOLANT(M, F, METHOD)
  where **F** is an n-dimensional array of size congruent with the grid and **Method** selects the precise interpolation method and hence the form of the function $c(x)$ to be used to generate function values in between grid points.

Interpolation in MATLAB

- Two interpolation methods are particularly prominent: *linear interpolation* and *cubic splines*.

Interpolation in MATLAB

- Two interpolation methods are particularly prominent: *linear interpolation* and *cubic splines*.
- To obtain off-grid function values, simply execute
  Y = F({z1,z2, . . . , zN})
  to evaluate the function F at z

Linear and Spline Interpolation

- **Linear interpolation** uses linear functions $c_i(x)$ in between grid points with the value matching condition $f_i + c_i(x_{i+1}) = f_{i+1}$ such that $c_i(x) = \frac{f_{i+1} - f_i}{x_{i+1} - x_i}(x - x_i)$

Linear and Spline Interpolation

- **Linear interpolation** uses linear functions $c_i(x)$ in between grid points with the value matching condition $f_i + c_i(x_{i+1}) = f_{i+1}$ such that $c_i(x) = \frac{f_{i+1}-f_i}{x_{i+1}-x_i}(x - x_i)$
- **Cubic Spline interpolation** uses cubic polynomials $c_i(x)$ in between grid points. The three parameters of the polynomials are determined by the value matching condition

$$f_i + c_i(x_{i+1}) = f_{i+1}$$

and smooth-pasting. That means, we require

$$c_i'(x_{i+1}) = c_{i+1}'(x_i) \text{ and } c_i''(x_{i+1}) = c_{i+1}''(x_i).$$

In addition usually $c_1'' = c_n'' = 0$ is required in order to be able to solve for all parameters of the local polynomials.

Linear and Spline Interpolation

- **Linear interpolation** is fast and produces no artifacts due to oscillation. Downside is that it generates functions that are not differentiable. In between grid-points gradient is constant.

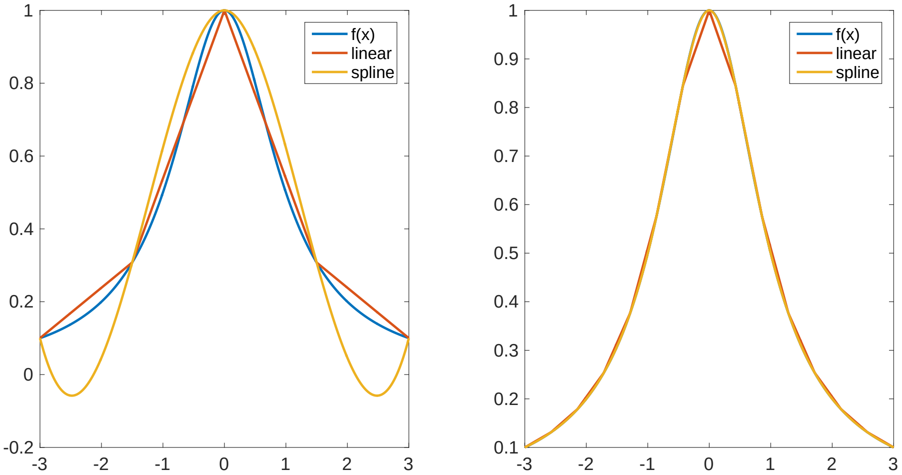
Linear and Spline Interpolation

- **Linear interpolation** is fast and produces no artifacts due to oscillation. Downside is that it generates functions that are not differentiable. In between grid-points gradient is constant.
- **Cubic Spline interpolation** is more tedious because it requires solution of a large system of nonlinear equations. Once this is done, evaluation is fast. It might produce artifacts, if c_i oscillates between grid-points (see Runge's phenomenon).

Linear and Spline Interpolation

- **Linear interpolation** is fast and produces no artifacts due to oscillation. Downside is that it generates functions that are not differentiable. In between grid-points gradient is constant.
- **Cubic Spline interpolation** is more tedious because it requires solution of a large system of nonlinear equations. Once this is done, evaluation is fast. It might produce artifacts, if c_i oscillates between grid-points (see Runge's phenomenon).
- Because of the locality of the approximation. In both cases, the approximation error gets smaller with more grid-points for well behaved functions.

Comparison of interpolation methods



Eigenvalues, Eigenvectors and Eigenvalue Decomposition

- Let $A \in \mathbb{R}^{m \times m}$, then the solutions $\lambda, x \neq 0$ to

$$Ax = \lambda x$$

  are called the Eigenvalue and Eigenvectors, respectively.
- Any matrix $A \in \mathbb{R}^{m \times m}$ has at most m Eigenvectors (of norm 1).
- Suppose A has full rank, then (a) the matrix X of the m Eigenvectors has full rank and (b) we can write $A = X \Lambda X^{-1}$.
- There are relatively fast algorithms to calculate the eigenvectors and eigenvalues of a matrix.
  [X,Λ]=EIG(A)
  calculates the eigenvalue decomposition of $A = X \Lambda X^{-1}$.

Eigenvalues, Eigenvectors and Eigenvalue Decomposition

- For any symmetric matrix of full rank A the Eigenvectors are orthogonal, hence $X'X = XX' = I$.
- The matrix power $A^t = \underbrace{A \times A \times \cdots \times A}_{\text{t times}}$ is given by

$$A^t = X\Lambda^t X^{-1}.$$

- Therefore, A^t converges to a limit different from zero if and only if the largest eigenvalue (in absolute terms) is equal to 1.
- In this case $A^\infty = X \begin{bmatrix} 1 & 0_{1,m-1} \\ 0_{m-1,1} & 0_{m-1,m-1} \end{bmatrix} X^{-1} = [x_1, 0, \ldots, 0]\, X^{-1}$

Rootfinding

- Often we need to solve (systems of) non-linear equations of the form

$$f(x) = 0$$

  or analogously the fixed point problem $g(x) = x$.
- If f is continuous and we have obtained x_1, x_2 such that

$$f(x_1) < 0 < f(x_2)$$

  then we know that a root (a zero of the function) exists.

Bisection Search

- An algorithm that converges for one-dimensional problems $f(x) = 0$ to a given precision in a fixed number of iterations is bi-section search.
- The idea is that in each iteration we have two values $x_1^{(n)}, x_2^{(n)}$ such that $f(x_1^{(n)}) < 0 < f(x_2^{(n)})$ and a candidate value $y = 0.5(x_1^{(n)} + x_2^{(n)})$.

Bisection Search

- If $f(y) = 0$ we have found a solution.
- If $f(y) < 0$ then we update $x_1^{(n+1)} = y, x_2^{(n+1)} = x_2^{(n)}$.
- If $f(y) > 0$ then we update $x_2^{(n+1)} = y, x_1^{(n+1)} = x_1^{(n)}$.
- We iterate until $|x_2^{(n)} - x_1^{(n)}| < \epsilon$.
- Since the distance between x_1 and x_2 halves every iteration, the distance after n iterations $d^{(n)} = |x_2^{(0)} - x_1^{(0)}|/2^n$. In each iteration the function is evaluated once.
- Thus the number of iterations and function evaluations is given by $n^* = \frac{\log d^{(0)} - \log \epsilon}{\log 2}$.

Newton Algorithm

- Can we do better for differentiable functions?
- The idea is to exploit a Taylor expansion

$$f(x) \approx f(x^*) + f'(x^*)(x - x^*).$$

- Suppose, we know $f(x^{(n)}), f'(x^{(n)})$ then we can generate a new candidate for $f(x) = 0$ by solving

$$0 = f(x^{(n)}) + f'(x^{(n)})(x^{(n+1)} - x^{(n)}).$$

- That is, we obtain $x^{(n+1)} = x^{(n)} - \frac{f(x^{(n)})}{f'(x^{(n)})}$

Newton Algorithm

- Convergence is not guaranteed and worst case performance is hence very bad.
- For well behaved, monotone functions, Newton's Algorithm converges quickly.
- That is, close to the true solution the algorithm is good.

Quasi-Newton Algorithms

- An additional downside is that we need to calculate the derivative in each iteration.
- Quasi-Newton methods replace the derivative calculation by finite differences based on function evaluations in past iterations.
- Brent's method combines a Quasi Newton approach with bisection search and obtains worst case performance of bisections, but converges in practice very fast.
- The method is implemented in MATLAB's
  $\text{XSTAR}=\text{FZERO}(f,[x_0,x_1])$
  .

Quasi-Newton Algorithms

- (Quasi)-Newton methods are also particularly useful for systems of non-linear equations.
- Here, bi-section search methods would need to find solutions conditional on other dimensions.
- Broyden's method is a particularly powerful Quasi-Newton approach - but it is a medium scale algorithm that needs to store large matrices.
- An alternative large scale algorithm is implemented in MATLAB's XSTAR=FSOLVE(f,x_0).

Broyden's method

- The goal is to solve $F(x^*) = 0$.
- A newton algorithm uses the approximation $0 = F(x^*) \approxeq F(x) + J(x)(x^* - x)$, such that

$$x^{(n+1)} = x^{(n)} - J(x)^{-1}F(x^n)$$

- For a Quasi-Newton algorithm, we replace the inverse of the Jacobian by an estimate updated between iterations $B^{(n)}$.

Broyden's method

```
function [xstar, fval, iter] = broyden(f,x0,critF,critX,maxiter)
  distF   =   9999;
  distX   =   9999;
  iter    =   0;
  xnow    =   x0(:); % x needs to be a column vector
  Fnow    =   f(xnow); Fnow=Fnow(:); % F needs to be a column vector
  Bnow    =   eye(length(xnow));
```

Broyden's method continued

```
while distF > critF & distX > critX & iter < maxiter
    iter  = iter+1; % count number of iterations
    Fold  = Fnow; % Store last function values
    xold  = xnow; % Store last x guess
    xnow  = xnow - Bnow*Fnow; % Update x guess
    Fnow  = f(xnow);
    Fnow  = Fnow(:);
    Dx    = xnow - xold; % Change in x
    Dy    = Fnow - Fold; % Change in F(x)
    % update inverse Jacobian
    Bnow  = Bnow + (Dx - Bnow*Dy)*(Dx'*Bnow)/(Dx'*Bnow*Dy);
    distF = max(abs(Fnow));
    distX = max(abs(Dx));
end
fval=Fnow; xstar=xnow;
```

Minimization routines

- Similar to rootfinding, there are several routines to minimize a function f.
- Some are derivative free and hence work for non-continuous functions, others work on first order conditions.

Minimization routines: Golden-Section

- Derivative free and in one-dimension: Golden-section search is like bi-section. Here we work with three evaluation points x_1, x_2 and $x_1 < x_m < x_2$ where $f(x_m) < f(x_1)$ and $f(x_m) < f(x_2)$.
- In golden section search we now select an evaluation candidate x_c such that the distance between $x_2 - x_1$ shrinks at constant rate.
- If $f(x_c) < f(x_m)$, then $x_1^{(n+1)} = x_m^{(n)}, x_m^{(n+1)} = x_c, x_2^{(n+1)} = x_2^{(n)}$.
- If $f(x_c) > f(x_m)$, then $x_2^{(n+1)} = x_c^{(n)}, x_m^{(n+1)} = x_m^{(n)}, x_1^{(n+1)} = x_1^{(n)}$.
- Therefore, we chose the candidate x_c such that $x_c - x_1 = x_2 - x_m$. The initial point x_m is chosen such that $\frac{x_2 - x_m}{x_1 - x_m}$ is the golden ratio.

Minimization routines: More than one Dimension

- Derivative free and in multi-dimension is the Nelder-Mead Simplex Method.
- This is implemented in MATLAB's [XMIN,FMIN]=FMINSEARCH(F,X0).

Minimization routines: More than one Dimension, Quasi-Newton

- Quasi-Newton methods, such as BFGS, calculate Derivatives and try to find the minimum based on finding a root to the Jacobian.
- This is implemented in MATLAB's [XMIN,FMIN]=FMINUNC(F,X0).

Dynamic Programming:
Some Theory & Application

General formulation, finite time horizon

Consider a dynamic problem of a generic form:

$$\max \sum_{t=0}^{T} \beta^t u\left(x_t, x_{t+1}\right), \text{ s.t. } x_{t+1} \in \Gamma_t\left(x_t\right), \; x_0 \text{ given.}$$

where $u\left(x_t, x_{t+1}\right)$ is a payoff function that depends on the current state x_t as well as on the future state x_{t+1} chosen by the decision maker.

General formulation, finite time horizon

Denoting the indirect utility obtained as $V(x_t, t)$, we can restate the Problem as a so called **Bellman equation**

$$V(x_t, t) = \max_{x_{t+1}} u(x_t, x_{t+1}) + \beta V(x_{t+1}, t+1), \text{ s.t. } x_{t+1} \in \Gamma_t(x_t).$$

where V exists if Γ is compact valued (Theorem of the Maximum).

General formulation, finite time horizon

Denoting the indirect utility obtained as $V(x_t, t)$, we can restate the Problem as a so called **Bellman equation**

$$V(x_t, t) = \max_{x_{t+1}} u(x_t, x_{t+1}) + \beta V(x_{t+1}, t+1), \text{ s.t. } x_{t+1} \in \Gamma_t(x_t).$$

where V exists if Γ is compact valued (Theorem of the Maximum).

The time-index t reflects the fact that it matters **how many remaining periods** there are.

General formulation, finite time horizon

Denoting the indirect utility obtained as $V(x_t, t)$, we can restate the Problem as a so called **Bellman equation**

$$V(x_t, t) = \max_{x_{t+1}} u(x_t, x_{t+1}) + \beta V(x_{t+1}, t+1), \text{ s.t. } x_{t+1} \in \Gamma_t(x_t).$$

where V exists if Γ is compact valued (Theorem of the Maximum).

The time-index t reflects the fact that it matters **how many remaining periods** there are.

With finite horizon the optimization problem is necessarily **non-stationary,** i.e. changes with time t.

A cake eating example

To fix ideas consider the usage of a depletable resource (cake-eating)

$$\max_{\{W_t\}_{t=0\ldots\infty}} \sum_{t=0}^{T} \beta^t u\left(c_t\right), \text{ s.t. } W_{t+1} = W_t - c_t,\ c_t \geq 0,\ W_0 \text{ given.}$$

To put this in the general form, expressing the problem only in terms of **state variables $\mathbf{W}_t$** we replace $c_t = W_t - W_{t+1}$

$$\max_{\{W_t\}_{t=0\ldots\infty}} \sum_{t=0}^{T} \beta^t u\left(W_t - W_{t+1}\right), \text{ s.t. } W_{t+1} \leq W_t.$$

A cake eating example

As formulation in terms of the Bellman equation, we obtain

$$V\left(W_t, t\right) = \max_{W_{t+1}} u\left(W_t - W_{t+1}\right) + \beta V\left(W_{t+1}, t+1\right), \text{ s.t. } W_{t+1} \leq W_t.$$

Bellman equations and infinite time horizon

While the dynamic programming (Bellman equation) approach generates somewhat more information about the optimization problem in a finite horizon setup than a direct attack at the problem, it is at the same time more burdensome, since we need to determine V for each t.

Bellman equations and infinite time horizon

While the dynamic programming (Bellman equation) approach generates somewhat more information about the optimization problem in a finite horizon setup than a direct attack at the problem, it is at the same time more burdensome, since we need to determine V for each t.

It becomes a powerful approach, once we look at infinite time horizon problems.

Bellman equations and infinite time horizon

While the dynamic programming (Bellman equation) approach generates somewhat more information about the optimization problem in a finite horizon setup than a direct attack at the problem, it is at the same time more burdensome, since we need to determine V for each t.

It becomes a powerful approach, once we look at infinite time horizon problems.

These **can be stationary,** i.e. they do not change in t, as the remaining time until the end of the decision problem remains always ∞.

Stationary dynamic programming

If the problem is stationary (and a solution does exist), we can state the planning problem as

$$V(x) = \max_{y} u(x,y) + \beta V(y) \text{ s.t. } y \in \Gamma(x).$$

Stationary dynamic programming

If the problem is stationary (and a solution does exist), we can state the planning problem as

$$V(x) = \max_{y} u(x, y) + \beta V(y) \text{ s.t. } y \in \Gamma(x).$$

- Note, however, that not all infinite horizon problems are stationary. Sometimes a problem can be reformulated in stationary terms (like in time series econometrics).

Stationary dynamic programming

If the problem is stationary (and a solution does exist), we can state the planning problem as

$$V(x) = \max_{y} u(x, y) + \beta V(y) \text{ s.t. } y \in \Gamma(x).$$

- Note, however, that not all infinite horizon problems are stationary. Sometimes a problem can be reformulated in stationary terms (like in time series econometrics).
- Also note that a solution may not exist.

Stationary dynamic programming

If the problem is stationary (and a solution does exist), we can state the planning problem as

$$V(x) = \max_y u(x, y) + \beta V(y) \ \text{ s.t. } \ y \in \Gamma(x).$$

- Note, however, that not all infinite horizon problems are stationary. Sometimes a problem can be reformulated in stationary terms (like in time series econometrics).
- Also note that a solution may not exist.
- The unknown of the Bellman equation is **the function** $V(x)$

An Example

The neo-classical growth model

A social planner wants to maximize the stream of utility from consumption in an economy, where

$$U = U(C_t)$$

An Example

The neo-classical growth model

A social planner wants to maximize the stream of utility from consumption in an economy, where

$$\begin{aligned} U &= U(C_t) \\ Y_t &= C_t + I_t;\ Y_t = K_t^{\alpha} \end{aligned}$$

An Example

The neo-classical growth model

A social planner wants to maximize the stream of utility from consumption in an economy, where

$$
\begin{aligned}
U &= U(C_t) \\
Y_t &= C_t + I_t;\; Y_t = K_t^{\alpha} \\
K_{t+1} &= (1-\delta) K_t + I_t
\end{aligned}
$$

An Example

The neo-classical growth model

A social planner wants to maximize the stream of utility from consumption in an economy, where

$$\begin{aligned} U &= U(C_t) \\ Y_t &= C_t + I_t;\ Y_t = K_t^{\alpha} \\ K_{t+1} &= (1-\delta)K_t + I_t \\ V(K_0) &= \max_{\{K_t\}_{t=1\ldots\infty}} \sum_{s=0}^{\infty} \beta^s U\left[C(K_s, K_{s+1})\right] \\ s.t.\ C &= K_t^{\alpha} + (1-\delta)K_t - K_{t+1} \\ K_{t+1} &\geq 0 \\ K_{t+1} &\leq K_t^{\alpha} + (1-\delta)K_t \end{aligned}$$

An Example

The neo-classical growth model

We can rewrite this as

$$\begin{aligned}
V(K_0) &= \max_{\{K_t\}_{t=1\ldots\infty}} \sum_{s=0}^{\infty} \beta^s U\left[C\left(K_s, K_{s+1}\right)\right] \\
&= \max_{K_1} \left\{ u\left(K_0, K_1\right) + \max_{\{K_t\}_{t=2\ldots\infty}} \beta \sum_{s=0}^{\infty} \beta^s u\left(K_s, K_{s+1}\right) \right\} \\
&= \max_{K' \in \Gamma(K)} \left\{ u\left(K, K'\right) + \beta V\left(K'\right) \right\} \\
\Gamma(K) &:= \left\{ K' \in \mathbb{R}_+ \middle| K' \leq K^\alpha + (1-\delta) K \right\} \\
u\left(K, K'\right) &:= U\left[C\left(K, K'\right)\right]
\end{aligned}$$

When does a solution exist?

We can formulate the Bellman equation as a mapping

$$U(x) = T(V(x))$$

$$T[V(x)] = \max_{y} u(x, y) + \beta V(y) \text{ s.t. } y \in \Gamma(x) \tag{9}$$

that maps function V to a new function U.

Existence of a Solution to the Bellman equation

The contraction mapping theorem

Theorem

If the Bellman equation (9) *defines* T *to be a* **contraction mapping** *on the* **set of continuous bounded functions**, *then a solution to* (9) *exists.*

Existence

Proof.

1.) note that if (9) has a solution, then it is a fixed point of T and vice versa.

Existence

Proof.

1.) note that if (9) has a solution, then it is a fixed point of T and vice versa.
2.) if the contraction mapping T has a fixed point, then it is unique: Let

$$Tx = x, Ty = y \Rightarrow$$

Existence

Proof.

1.) note that if (9) has a solution, then it is a fixed point of T and vice versa.

2.) if the contraction mapping T has a fixed point, then it is unique: Let

$$\begin{aligned} Tx &= x, Ty = y \Rightarrow \\ d(x,y) &= d(Tx,Ty) \leq \rho d(x,y) \end{aligned}$$

Existence

Proof.

1.) note that if (9) has a solution, then it is a fixed point of T and vice versa.
2.) if the contraction mapping T has a fixed point, then it is unique: Let

$$\begin{aligned} Tx &= x, Ty = y \Rightarrow \\ d(x,y) &= d(Tx,Ty) \leq \rho d(x,y) \\ &\Rightarrow d(x,y) = 0 \end{aligned}$$

Existence

Proof.

1.) note that if (9) has a solution, then it is a fixed point of T and vice versa.
2.) if the contraction mapping T has a fixed point, then it is unique: Let

$$\begin{aligned} Tx &= x, Ty = y \Rightarrow \\ d(x,y) &= d(Tx,Ty) \leq \rho d(x,y) \\ &\Rightarrow d(x,y) = 0 \\ &\Rightarrow x = y \end{aligned}$$

□

Existence

Proof.

3.) T is continuous, i.e. $y_n \to y \Rightarrow Ty_n \to Ty$ as

$$d\left(Ty_n, Ty\right) \quad \leq \quad \rho d\left(y_n, y\right) \overset{n \to \infty}{\to} 0$$

Existence

Proof.

3.) T is continuous, i.e. $y_n \to y \Rightarrow Ty_n \to Ty$ as

$$\begin{aligned} d\left(Ty_n, Ty\right) &\leq \rho d\left(y_n, y\right) \overset{n\to\infty}{\to} 0 \\ &\Rightarrow Ty_n \to Ty \end{aligned}$$

Existence

Proof.

3.) T is continuous, i.e. $y_n \to y \Rightarrow Ty_n \to Ty$ as

$$\begin{aligned} d\left(Ty_n, Ty\right) &\leq \rho d\left(y_n, y\right) \overset{n\to\infty}{\to} 0 \\ &\Rightarrow Ty_n \to Ty \end{aligned}$$

4.) the limit $\lim_{n\to\infty} T^n x =: x^*$ exists because $T^n x$ is Cauchy.

Existence

Proof.

3.) T is continuous, i.e. $y_n \to y \Rightarrow Ty_n \to Ty$ as

$$\begin{aligned} d\left(Ty_n, Ty\right) &\leq \rho d\left(y_n, y\right) \overset{n\to\infty}{\to} 0 \\ &\Rightarrow Ty_n \to Ty \end{aligned}$$

4.) the limit $\lim_{n\to\infty} T^n x =: x^*$ exists because $T^n x$ is Cauchy.

5.) this implies

$$Tx^* = T\left(\lim_{n\to\infty} T^n x\right)$$

Existence

Proof.

3.) T is continuous, i.e. $y_n \to y \Rightarrow Ty_n \to Ty$ as

$$\begin{aligned} d\left(Ty_n, Ty\right) &\leq \rho d\left(y_n, y\right) \overset{n\to\infty}{\to} 0 \\ &\Rightarrow Ty_n \to Ty \end{aligned}$$

4.) the limit $\lim_{n\to\infty} T^n x =: x^*$ exists because $T^n x$ is Cauchy.

5.) this implies

$$\begin{aligned} Tx^* &= T\left(\lim_{n\to\infty} T^n x\right) \\ &= \lim_{n\to\infty} TT^n x = \lim_{n\to\infty} T^{n+1} x = x^*, \end{aligned}$$

so that x^* is the fixed point of T, which concludes the proof.

An algorithm to solve the Bellman equation

The proof to show existence of a solution to the Bellman equation is constructive. It tells us how to find a solution to the Bellman equation:

1. Show that T is a contraction.

An algorithm to solve the Bellman equation

The proof to show existence of a solution to the Bellman equation is constructive. It tells us how to find a solution to the Bellman equation:

1. Show that T is a contraction.
2. Start with some initial guess V_0 and then construct a sequence $V_n = TV_{n-1}$.

An algorithm to solve the Bellman equation

The proof to show existence of a solution to the Bellman equation is constructive. It tells us how to find a solution to the Bellman equation:

1. Show that T is a contraction.
2. Start with some initial guess V_0 and then construct a sequence $V_n = TV_{n-1}$.
3. After a sufficiently large number of iterations V_n will become arbitrarily close to the solution V.

An algorithm to solve the Bellman equation

The proof to show existence of a solution to the Bellman equation is constructive. It tells us how to find a solution to the Bellman equation:

1. Show that T is a contraction.
2. Start with some initial guess V_0 and then construct a sequence $V_n = TV_{n-1}$.
3. After a sufficiently large number of iterations V_n will become arbitrarily close to the solution V.
4. This algorithm is called "Value-Function-Iteration" (VFI).

Blackwell's condition

Theorem

If T fulfills the following conditions, then T is a contraction mapping on the set of bounded and continuous functions:

1. *T preserves boundedness.*
2. *T preserves continuity.*
3. *T is monotonic: $w \geq v \Rightarrow Tw \geq Tv$*
4. *T satisfies discounting, i.e. there is some $0 \leq \beta < 1$, such that for any real valued constant c and any function v we have $T(v + c) \leq Tv + \beta c$.*

A solution exists in the generic case

Theorem (Existence of the value function)

Assume $u(x, x')$ *is* **real-valued, continuous, and bounded**, $0 < \beta < 1$, *and that the constraint set* $\Gamma(s)$ *is a* **non-empty, compact-valued, and continuous** *correspondence. Then there exists a unique continuous value function* $V(s)$ *that solves the Bellman equation* (9).

Properties of the Value and Policy Function

It is useful to know/show properties of the value and policy function in order to exploit them numerically. Examples are

- Continuity$\rightarrow$ allows interpolation

Properties of the Value and Policy Function

It is useful to know/show properties of the value and policy function in order to exploit them numerically. Examples are

- Continuity$\rightarrow$ allows interpolation
- Differentiability & Convexity $\rightarrow$ allows using first order conditions

Properties of the Value and Policy Function

It is useful to know/show properties of the value and policy function in order to exploit them numerically. Examples are

- Continuity→ allows interpolation
- Differentiability & Convexity → allows using first order conditions
- Monotonicity → provides lower bound if values are searched sequentially

Policy function

Theorem (Existence of the policy function)

Assume $u(x, x')$ is real-valued, continuous, strictly **concave** *and bounded, $0 < \beta < 1$, and that the set of potential states is a* **convex** *subset of $\mathbb{R}^k$ and the constraint set $\Gamma(s)$ is a non-empty, compact-valued, continuous,* **and convex** *correspondence. Then the unique value function $V(s)$ is continuous and strictly concave. Moreover the* **optimal policy**

$$\phi(x) := \arg\max_{y \in \Gamma(x)} u(x, y) + \beta V(x)$$

is a continuous (single-valued) function.

A First Algorithm

The first algorithm we want to study is the **Value Function Iteration** outlined before. It focuses on the Bellman equation, computing the value functions by backward iteration from an initial guess.

Putting things to work: non-stochastic case

- Value function Iteration loops over

$$V^n = T\left(V^{n-1}\right) = \max_{s' \in \Gamma(s)} u\left(s, s'\right) + \beta V^{n-1}\left(s'\right)$$

until $\left|V^n - V^{n-1}\right| < crit$

Putting things to work: non-stochastic case

- Value function Iteration loops over

$$V^n = T\left(V^{n-1}\right) = \max_{s' \in \Gamma(s)} u\left(s, s'\right) + \beta V^{n-1}\left(s'\right)$$

  until $\left|V^n - V^{n-1}\right| < crit$
- To implement this, we specify N discrete points in the state space $s_1, s_2, ..., s_N$ and denote $V_i^n = V^n\left(s_i\right)$.

Putting things to work: non-stochastic case

- Value function Iteration loops over

$$V^n = T\left(V^{n-1}\right) = \max_{s' \in \Gamma(s)} u\left(s, s'\right) + \beta V^{n-1}\left(s'\right)$$

  until $\left|V^n - V^{n-1}\right| < crit$

- To implement this, we specify N discrete points in the state space $s_1, s_2, ..., s_N$ and denote $V_i^n = V^n\left(s_i\right)$.
- Note: In the stochastic case, V_i^n includes the expectations operator.

Putting things to work: non-stochastic case

- Value function Iteration loops over

$$V^n = T\left(V^{n-1}\right) = \max_{s' \in \Gamma(s)} u\left(s, s'\right) + \beta V^{n-1}\left(s'\right)$$

  until $\left|V^n - V^{n-1}\right| < crit$
- To implement this, we specify N discrete points in the state space $s_1, s_2, ..., s_N$ and denote $V_i^n = V^n\left(s_i\right)$.
- Note: In the stochastic case, V_i^n includes the expectations operator.
- Let $U\left(i\right) = \left(u(i,1), u\left(i,2\right), ..., u\left(i,N\right)\right)$ be the vector of all possible instantaneous utility levels obtained by going from state i to j (if impossible, set to $-\infty$).

Putting things to work

- Optimizing starting from state i

$$V_i^n = \max \left\{ U\left(i\right) + \beta \mathbf{V}^{n-1} \right\}$$

Putting things to work

- Optimizing starting from state i

$$V_i^n = \max\left\{U\left(i\right) + \beta \mathbf{V}^{n-1}\right\}$$

- Denote $\iota_N' = \left(1, ..., 1\right)$, and stacking the above expression, we obtain

$$\mathbf{V}^n = \max\left\{\mathbf{U} + \beta \iota_N \mathbf{V}^{n-1}\right\}$$

where bold typeset indicates linewise stacked variables. Maximum is line-by-line.

Putting things to work

- Optimizing starting from state i

$$V_i^n = \max\left\{U(i) + \beta \mathbf{V}^{n-1}\right\}$$

- Denote $\iota_N' = (1, ..., 1)$, and stacking the above expression, we obtain

$$\mathbf{V}^n = \max\left\{\mathbf{U} + \beta \iota_N \mathbf{V}^{n-1}\right\}$$

where bold typeset indicates linewise stacked variables. Maximum is line-by-line.

- You can take the max of an array along dimension j (here $j = 2$) by [VALUE, INDEX]=MAX(X,[],J)

Speed of Convergence

- Value Function Iteration is notoriously slow.
- Let V^* be the true solution to the Bellman equation.
- Suppose, we guess $\hat{V} = V^* + c$, i.e. an exact guess up to a constant.

Speed of Convergence

- Then the value function update yields

$$T(\hat{V}) = \max_{s'}\{u(s,s') + \beta(V^* + c)\} = \max_{s'}\{u(s,s') + \beta V^*\} + \beta c$$

- Since V^* is the solution to the Bellman equation, we obtain

$$T(\hat{V}) = \max_{s'}\{u(s,s') + \beta(V^* + c)\} = V^* + \beta c,$$

i.e., the distance to the true solution is $(1-\beta)^n c$ after n iterations.

Exercise 1

Exercise

Solve the consumption-savings problem with two income states, z. *Consider* $z \in \{0.9, 1.1\}$ *where the transition probabilities are given by* $\begin{bmatrix} 0.875 & 0.125 \\ 0.125 & 0.875 \end{bmatrix}$, *i.e., with probability 87.5% the economy remains in one productivity state.*

1. *Write a code that solves the model on a grid. Plot consumption as a function of* (K, z).

Off-grid search

- Alternatively to discretizing the decision problem, we can approximate the value function parametrically.

Off-grid search

- Alternatively to discretizing the decision problem, we can approximate the value function parametrically.
- Again, we want to solve the continuous state-space problem

$$V(s) = \max_{y \in \Gamma(s)} u(s, s') + \beta V(s').$$

Off-grid search

- Alternatively to discretizing the decision problem, we can approximate the value function parametrically.
- Again, we want to solve the continuous state-space problem

$$V(s) = \max_{y \in \Gamma(s)} u(s, s') + \beta V(s').$$

- So far, we used a fine grid $\{s_1, \ldots, s_N\}$ for s and then solving by iterating over the approximated, discretized problem.

$$\mathbf{V}^n = \max \left\{ \mathbf{U} + \iota_N \mathbf{V}^{n-1} \right\}.$$

Off-grid search: Setup

Alternatively, we can define a function

$$\hat{V}\left(s|\,\theta_n\right)$$

for a given parametric form and solve for θ such that the Bellmann equation holds.

Off-grid search: Value Function update

Again, for value function iteration, we start with a guess for θ and update the value function on a grid of points s_i

$$V_i^{n+1} = \max_{s' \in \Gamma(s_i)} u\left(s_i, s'\right) + \beta \hat{V}\left(s' | \theta_n\right)$$

The maximization now does no longer restrict s' to be on a grid.

Off-grid search

- As a next step, we need to update parameters.
- How we do this? Particularly useful parametric families have a one-to-one mapping from $V^{(n)} = [V_i]_{i=1...m}$ to $\theta^{(n+1)}$.
- Two examples: Chebyshev polynomials of order m or interpolations.
- For interpolation we obtain
  $\hat{V}^{(n+1)} = \textsc{griddedInterpolant}(\textsc{grid.s}, V^{(n)})$
  where grid.s is the grid for the state variables.
- For Chebyshev polynomials, we obtain
  $\theta^{(n+1)} = \textsc{inv}(\text{P'*P})\text{*P'}\ V^{(n)}$
  where P is the matrix of the m Chebyshev polynomials evaluated at the m grid points in grid.s.

Exercise 1 b

Exercise

Use the consumption-savings problem as in Exercise 1.

Implement spline interpolation to solve for the value function by VFI, starting with $V = 0$ *as initial guess, save running times.*

Towards more efficient solution methods

- Looking at the Bellman equation through the lens of a parameterized problem makes the following extension to the solution method straightforward:
- The Bellman equation now defines a mapping in $\mathbb{R}^m$, $T : \theta \rightarrow \theta'$ and we are still looking for a fixed point of this mapping.
- We can rewrite the fixed point problem in the form of a root-finding problem $f(\theta) = \theta - T(\theta) = 0$ and apply a (Quasi)-Newton method, e.g. Broyden's method, to the problem.
- Broyden's method is particularly well-suited as the first update step is exactly

$$x^{(1)} = x^{(0)} - f(x^{(0)}) = x^{(0)} - x^{(0)} + T(x^{(0)}) = T(x^{(0)})$$

- Policy Function Iteration is similar in that it can be represented as a Newton Method, too.

Exercise 1 c

Exercise

Redo Exercise 1b but use Broyden's rootfinding procedure instead of VFI - compare running times and numbers of iterations.

Need for Speed

- Value Function Iteration is a simple and robust algorithm.

Need for Speed

- Value Function Iteration is a simple and robust algorithm.
- because it comes straight from proof of existence of a solution.

Need for Speed

- Value Function Iteration is a simple and robust algorithm.
- because it comes straight from proof of existence of a solution.
- Yet it is slow!

Need for Speed

- Value Function Iteration is a simple and robust algorithm.
- because it comes straight from proof of existence of a solution.

- Yet it is slow!
- Terribly slow! The convergence rate declines to $1 - \beta$ at some point.

Need for Speed

- Value Function Iteration is a simple and robust algorithm.
- because it comes straight from proof of existence of a solution.

- Yet it is slow!
- Terribly slow! The convergence rate declines to $1 - \beta$ at some point.

- $\implies$ need alternative (global) solution methods to dynamic decision problems.

Solving directly for policy functions

- VFI relies on calculating of the value function to obtain policies.

Sometime we do not need to know the VF

Solving directly for policy functions

- VFI relies on calculating of the value function to obtain policies.
- Only the latter is (in most applications) the object of interest.

Sometime we do not need to know the VF

Solving directly for policy functions

- VFI relies on calculating of the value function to obtain policies.
- Only the latter is (in most applications) the object of interest.

Sometime we do not need to know the VF

- **Projection Methods** (PM) and

Solving directly for policy functions

- VFI relies on calculating of the value function to obtain policies.
- Only the latter is (in most applications) the object of interest.

Sometime we do not need to know the VF

- **Projection Methods** (PM) and
- **Endogenous Grid Methods** (EGM)

Solving directly for policy functions

- VFI relies on calculating of the value function to obtain policies.
- Only the latter is (in most applications) the object of interest.

Sometime we do not need to know the VF

- **Projection Methods** (PM) and
- **Endogenous Grid Methods** (EGM)
- both directly solve for the policy function.

Solving directly for policy functions

- VFI relies on calculating of the value function to obtain policies.
- Only the latter is (in most applications) the object of interest.

Sometime we do not need to know the VF

- **Projection Methods** (PM) and
- **Endogenous Grid Methods** (EGM)
- both directly solve for the policy function.

- Often substantially faster, but–at least in “pure” form–require convex problems characterized by sufficient first order conditions.

Euler equation

- Consider our stochastic dynamic programming problem

$$V(s, \xi) = \max_{s'} u(s, s', \xi) + \beta E_{\xi} V(s', \xi')$$

Euler equation

- Consider our stochastic dynamic programming problem

$$V(s,\xi) = \max_{s'} u(s, s', \xi) + \beta E_{\xi} V(s', \xi')$$

- The first order condition is

$$\frac{\partial u}{\partial s'}(s, s', \xi) + \beta E_{\xi} \frac{\partial V}{\partial s}(s', \xi') = 0$$

Euler equation

- Consider our stochastic dynamic programming problem

$$V(s,\xi) = \max_{s'} u(s,s',\xi) + \beta E_{\xi} V(s',\xi')$$

- The first order condition is

$$\frac{\partial u}{\partial s'}(s,s',\xi) + \beta E_{\xi} \frac{\partial V}{\partial s}(s',\xi') = 0$$

- Optimality (envelope theorem) implies

$$\frac{\partial V}{\partial s}(s) = \frac{\partial u}{\partial s}(s,s',\xi)$$

Euler equation

- Therefore first order condition yields the **Euler equation**

$$\frac{\partial u}{\partial s'}(s, s', \xi) = -\beta E_{\xi} \frac{\partial u}{\partial s}(s', s'', \xi')$$

where $\frac{\partial u}{\partial s}$ is the derivative w.r.t. the first and $\frac{\partial u}{\partial s'}$ the derivative w.r.t. the second argument.

Endogenous Grid Method

- All the solution methods discussed so far involve root-finding.

Endogenous Grid Method

- All the solution methods discussed so far involve root-finding.
- This is numerically intense.

Endogenous Grid Method

- All the solution methods discussed so far involve root-finding.
- This is numerically intense.
- Caroll (2005) proposes a method to solve dynamic optimization problems without relying on root-solving.

Endogenous Grid Method

- All the solution methods discussed so far involve root-finding.
- This is numerically intense.

- Caroll (2005) proposes a method to solve dynamic optimization problems without relying on root-solving.
- This method makes the grid and not the policy “endogenous”.

Endogenous Grid Method

- The idea is akin to forward induction: "Given the state I am in now, what must have been the state I was in before, if I behaved rational".

Endogenous Grid Method

- The idea is akin to forward induction: "Given the state I am in now, what must have been the state I was in before, if I behaved rational".
- The method deals with occasionally binding constraints, but requires

Endogenous Grid Method

- The idea is akin to forward induction: "Given the state I am in now, what must have been the state I was in before, if I behaved rational".
- The method deals with occasionally binding constraints, but requires
  - differentiability of the value function everywhere,

Endogenous Grid Method

- The idea is akin to forward induction: "Given the state I am in now, what must have been the state I was in before, if I behaved rational".
- The method deals with occasionally binding constraints, but requires
  - differentiability of the value function everywhere,
  - (in its basic form) first order conditions that are sufficient, and

Endogenous Grid Method

- The idea is akin to forward induction: "Given the state I am in now, what must have been the state I was in before, if I behaved rational".
- The method deals with occasionally binding constraints, but requires
  - differentiability of the value function everywhere,
  - (in its basic form) first order conditions that are sufficient, and
  - monotone policy function (isomorphisms)

Endogenous Grid Method

- Consider the dynamic programming problem

$$V(s, \xi) = \max_{s' \geq b} u\left((1+r)s - s' + \xi\right) + \beta E V\left(s', \xi'\right).$$

Endogenous Grid Method

- Consider the dynamic programming problem

$$V(s,\xi) = \max_{s' \geq b} u\left((1+r)s - s' + \xi\right) + \beta E V\left(s', \xi'\right).$$

- The first-order condition

$$u'\left((1+r)s - s' + \xi\right) = (1+r)\beta E u'((1+r)s' - s'' + \xi') + \lambda \tag{10}$$

characterizes the optimal solution. Where $\lambda = 0$ if $s' > b$

Endogenous Grid Method

As in projection method, we solve for the policy function directly
Yet, the algorithm works iteratively.

1. Start with a policy guess (for simplicity for consumption).

Endogenous Grid Method

As in projection method, we solve for the policy function directly
Yet, the algorithm works iteratively.

1. Start with a policy guess (for simplicity for consumption).
2. Thereby, obtain values for the RHS of (10) for each grid point.

Endogenous Grid Method

As in projection method, we solve for the policy function directly
Yet, the algorithm works iteratively.

1. Start with a policy guess (for simplicity for consumption).
2. Thereby, obtain values for the RHS of (10) for each grid point.
3. Then solve for the current state s such that (10) holds for each s'.

Endogenous Grid Method

As in projection method, we solve for the policy function directly
Yet, the algorithm works iteratively.

1. Start with a policy guess (for simplicity for consumption).
2. Thereby, obtain values for the RHS of (10) for each grid point.
3. Then solve for the current state s such that (10) holds for each s'.
4. This yields a set of inner solution pairs $\{s', (s, \xi)\}$ to be inverted.

Endogenous Grid Method

As in projection method, we solve for the policy function directly
Yet, the algorithm works iteratively.

1. Start with a policy guess (for simplicity for consumption).
2. Thereby, obtain values for the RHS of (10) for each grid point.
3. Then solve for the current state s such that (10) holds for each s'.
4. This yields a set of inner solution pairs $\{s', (s, \xi)\}$ to be inverted.
5. It also identifies the current states s for which the constraint binds.

Endogenous Grid Method

As in projection method, we solve for the policy function directly
Yet, the algorithm works iteratively.

1. Start with a policy guess (for simplicity for consumption).
2. Thereby, obtain values for the RHS of (10) for each grid point.
3. Then solve for the current state s such that (10) holds for each s'.
4. This yields a set of inner solution pairs $\{s', (s, \xi)\}$ to be inverted.
5. It also identifies the current states s for which the constraint binds.
6. Go back to 2. and iterate until convergence.

Endogenous Grid Method

- Say $c^{(n)}$ is the policy function in iteration n. Then, we can be calculate the necessary assets s as

$$c^*(s', \xi) = u'^{-1} \left\{ (1+r)\beta E_{\xi} u' \left[c^{(n)}(s', \xi') \right] \right\}$$
$$(1+r)s^*(s', \xi) = s' - \xi + c^*(s', \xi)$$

Endogenous Grid Method

- Say $c^{(n)}$ is the policy function in iteration n. Then, we can be calculate the necessary assets s as

$$c^*(s', \xi) = u'^{-1}\left\{(1+r)\beta E_\xi u'\left[c^{(n)}(s', \xi')\right]\right\}$$
$$(1+r)s^*(s', \xi) = s' - \xi + c^*(s', \xi)$$

- For a given grid of points s', s^* is typically off-grid!

Endogenous Grid Method

- Say $c^{(n)}$ is the policy function in iteration n. Then, we can be calculate the necessary assets s as

$$c^*(s',\xi) = u'^{-1}\left\{(1+r)\beta E_\xi u'\left[c^{(n)}(s',\xi')\right]\right\}$$
$$(1+r)s^*(s',\xi) = s' - \xi + c^*(s',\xi)$$

- For a given grid of points s', s^* is typically off-grid!
- However, we have solved a policy function for some asset levels:

$$(s^*,\xi) \rightarrow c^*(s',\xi)$$

Policy functions on a fixed grid

- For well behaved felicity functions u, s^* and c^* are monotone in s'. This allows us to move to the next iteration making use of the following two implications:

Policy functions on a fixed grid

- For well behaved felicity functions u, s^* and c^* are monotone in s'. This allows us to move to the next iteration making use of the following two implications:
  - For a given ξ we can obtain consumption policies $c^{(n+1)}(s,\xi)$ by interpolating c^* at $s^*(s'_1 = b, \xi) \leq s \leq s^*(s'_n, \xi)$.

Policy functions on a fixed grid

- For well behaved felicity functions u, s^* and c^* are monotone in s'. This allows us to move to the next iteration making use of the following two implications:
  - For a given ξ we can obtain consumption policies $c^{(n+1)}(s,\xi)$ by interpolating c^* at $s^*(s'_1 = b,\xi) \leq s \leq s^*(s'_n,\xi)$.
  - For any $s < s^*(s'_1,\xi)$ households would like to choose assets *lower* than the borrowing limit. Of course they cannot. Hence, we know their asset policy: $s'(s,\xi) = b$ and their consumption is $c^{(n+1)}(s,\xi) = (1+r)s + \xi - b$.

Policy functions on a fixed grid

- For well behaved felicity functions u, s^* and c^* are monotone in s'. This allows us to move to the next iteration making use of the following two implications:
  - For a given ξ we can obtain consumption policies $c^{(n+1)}(s, \xi)$ by interpolating c^* at $s^*(s'_1 = b, \xi) \leq s \leq s^*(s'_n, \xi)$.
  - For any $s < s^*(s'_1, \xi)$ households would like to choose assets *lower* than the borrowing limit. Of course they cannot. Hence, we know their asset policy: $s'(s, \xi) = b$ and their consumption is $c^{(n+1)}(s, \xi) = (1 + r)s + \xi - b$.
- We then iterate $c^{(n)}$ until convergence.

EGM: Further issues

- One important issue is how to choose starting guesses for $c^{(0)}$. Here it is useful to recall that the infite horizon planning problem can be viewed as the limit of a finite horizon problem. Hence start with $c^{(0)} = (1+r)s + \xi$ (if possible).

EGM: Further issues

- One important issue is how to choose starting guesses for $c^{(0)}$. Here it is useful to recall that the infite horizon planning problem can be viewed as the limit of a finite horizon problem. Hence start with $c^{(0)} = (1+r)s + \xi$ (if possible).
- There are extensions on

EGM: Further issues

- One important issue is how to choose starting guesses for $c^{(0)}$. Here it is useful to recall that the infite horizon planning problem can be viewed as the limit of a finite horizon problem. Hence start with $c^{(0)} = (1+r)s + \xi$ (if possible).
- There are extensions on
  - **How to deal with multiple assets** (Hintermaier and Koeniger, 2010): Find the asset combinations in $t+1$ that can be optimal using FOCs, then map back from only these optimal points.

EGM: Further issues

- One important issue is how to choose starting guesses for $c^{(0)}$. Here it is useful to recall that the infite horizon planning problem can be viewed as the limit of a finite horizon problem. Hence start with $c^{(0)} = (1+r)s + \xi$ (if possible).
- There are extensions on
  - **How to deal with multiple assets** (Hintermaier and Koeniger, 2010): Find the asset combinations in $t+1$ that can be optimal using FOCs, then map back from only these optimal points.
  - **How to deal with non-convex setups** (Fella, 2015): Use the fact that FOCs are still necessary and compare potential solutions.

Exercise 2

Exercise

Solve the consumption-savings problem from before using EGM.

Stochastic Fluctuations and Markov Chains

A discrete, finite **Markov chain** is given by

- a finite grid of states ξ_i, $i = 1...N$,
- a transition probability matrix $P = (p_{ij})$, and
- a sequence of stochastic variables X_t

such that:

- X_t takes only values on the grid
- the probability to go from state i to state j is given by p_{ij}

Since probabilities depend only on last state

- Markov chains have finite memory, and
- $\pi_t = \pi_0 P^t$ gives the unconditional distribution of X_t, for the distribution X_0 is given by π_0.

Markov Chains and stationary distributions

- The **stationary distribution** of a Markov chain fulfills the equation

$$\pi = \pi P$$

- We can therefore calculate a stationary distribution of P as a **left unit-eigenvector** of P.

Markov Chains and stationary distributions

- A Markov chain is said to be **irreducible** and aperiodic if the probability to go from any state to any other state is strictly positive after sufficiently many iterations. That is $P^n > 0$.
- Any irreducible and aperiodic Markov chain has a unique stationary distribution, which is also equal to the limit of the sequence of probability distributions $\pi^* = \lim_{n\to\infty} \pi P^n$.

Why Markov Chains in Dynamic Programming?

- If the planning problem is stationary and Markovian, i.e. the decisions will depend only on a finite history, then we can write it in recursive form.
- Therefore, to be able to write the planning problem recursively, we need to require the stochastic process to be Markovian as well.

Easy conditional expectations

- In addition, if the stochastic process is approximated by a DMC, we can express conditional expectations fairly simple.

Easy conditional expectations

- In addition, if the stochastic process is approximated by a DMC, we can express conditional expectations fairly simple.
- Suppose s are the endogenous states, written in a way such that we can separate them from the stochastic states ξ.

Easy conditional expectations

- In addition, if the stochastic process is approximated by a DMC, we can express conditional expectations fairly simple.
- Suppose s are the endogenous states, written in a way such that we can separate them from the stochastic states ξ.
- Then we can write the vector valued value function $V(s) = [v(s, \xi_i)]_{i=1...N} \in \mathbb{R}^{1\times N}$.

Easy conditional expectations

- In addition, if the stochastic process is approximated by a DMC, we can express conditional expectations fairly simple.
- Suppose s are the endogenous states, written in a way such that we can separate them from the stochastic states ξ.
- Then we can write the vector valued value function $V(s) = [v(s, \xi_i)]_{i=1\ldots N} \in \mathbb{R}^{1\times N}$.
- The expected value functions conditional on ξ then simply read

$$EV(s'|\xi) = V(s')P' = [\sum_j p_{ij} v(s', \xi_j)]_{i=1\ldots N}.$$

Easy conditional expectations

- In addition, if the stochastic process is approximated by a DMC, we can express conditional expectations fairly simple.
- Suppose s are the endogenous states, written in a way such that we can separate them from the stochastic states ξ.
- Then we can write the vector valued value function $V(s) = [v(s, \xi_i)]_{i=1...N} \in \mathbb{R}^{1\times N}$.
- The expected value functions conditional on ξ then simply read

$$EV(s'|\xi) = V(s')P' = [\sum_j p_{ij} v(s', \xi_j)]_{i=1...N}.$$

- Recall matrix multiplication is fast (due to optimized BLAS)!

Markov Chains, Two examples

Example

The Markov chain characterized by

$$P = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

has an ergodic distribution, since

$$P^2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

is positive everywhere.

Markov Chains, Two examples

Example

The Markov chain characterised by

$$P = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$$

has no ergodic distribution, since the upper and the lower block of states do not "connect".

How to simulate a Markov chain

1. If we want to simulate a Markov chain with transition matrix P, we need to start with some initial state $s_0 = i$.

How to simulate a Markov chain

1. If we want to simulate a Markov chain with transition matrix P, we need to start with some initial state $s_0 = i$.
2. We calculate $\Pi = \left(\pi_{i,j}\right)_{i,j=1\ldots N} = \left(\sum_{k=1}^{j} p_{i,k}\right)_{i,j=1\ldots N}$.

How to simulate a Markov chain

1. If we want to simulate a Markov chain with transition matrix P, we need to start with some initial state $s_0 = i$.
2. We calculate $\Pi = \left(\pi_{i,j}\right)_{i,j=1\ldots N} = \left(\sum_{k=1}^{j} p_{i,k}\right)_{i,j=1\ldots N}$.
3. We start with $t = 0$.

How to simulate a Markov chain

1. If we want to simulate a Markov chain with transition matrix P, we need to start with some initial state $s_0 = i$.
2. We calculate $\Pi = (\pi_{i,j})_{i,j=1...N} = \left(\sum_{k=1}^{j} p_{i,k}\right)_{i,j=1...N}$.
3. We start with $t = 0$.
4. Then we draw a uniformly $(0, 1)$ distributed variable u_t.

How to simulate a Markov chain

1. If we want to simulate a Markov chain with transition matrix P, we need to start with some initial state $s_0 = i$.
2. We calculate $\Pi = \left(\pi_{i,j}\right)_{i,j=1\ldots N} = \left(\sum_{k=1}^{j} p_{i,k}\right)_{i,j=1\ldots N}$.
3. We start with $t = 0$.
4. Then we draw a uniformly $(0,1)$ distributed variable u_t.
5. and find j such that $\pi_{s_t,j-1} < u_t \leq \pi_{s_t,j}$. This is state $s_{t+1} = j$.

How to simulate a Markov chain

1. If we want to simulate a Markov chain with transition matrix P, we need to start with some initial state $s_0 = i$.
2. We calculate $\Pi = \left(\pi_{i,j}\right)_{i,j=1...N} = \left(\sum_{k=1}^{j} p_{i,k}\right)_{i,j=1...N}$.
3. We start with $t = 0$.
4. Then we draw a uniformly $(0,1)$ distributed variable u_t.
5. and find j such that $\pi_{s_t,j-1} < u_t \leq \pi_{s_t,j}$. This is state $s_{t+1} = j$.
6. We repeat steps 4 and 5 until $t = T$.

How to simulate a Markov chain

1. We can speed this up somewhat by drawing a vector u at the beginning of the simulation.

How to simulate a Markov chain

1. We can speed this up somewhat by drawing a vector u at the beginning of the simulation.
2. We can use this algorithm also in vectorized form to simulate many Markov Chains in parallel.

Policy functions inducing a Markov chain

Suppose we have solved a dynamic programming problem

$$V(s,\xi) = \max_{s'} u(s,s') + \beta E_{\xi'} V(s',\xi')$$

at nodes $(s,\xi) \in S \times \Xi$ by **linearly** interpolating V off nodes, where S and Ξ are indexed sets $S = \{s_1,\ldots,s_n\}$, $\Xi = \{\xi_i,\ldots,\xi_m\}$.

(For exposition we assume that S is one dimensional, but everything extends to higher dimensional S.)

Policy functions and indexes

Then, we have obtained a policy function

$$s^*(s,\xi) = \arg\max_{s'} u(s,s') + \beta E_{\xi'} V(s',\xi').$$

Now let $i^*(s,\xi)$ be the index of the next smallest element in S relative to s^*, i.e. $s^* \in [s_i, s_{i+1})$.

Policy functions: Linear interpolation weights

Define weights $w(s,\xi) = \frac{s^* - s_{i^*}}{s_{i^*+1} - s_{i^*}}$.

- then the linearly interpolated value function is given by

$$V(s,\xi) = u\left[s, s^*(s,\xi)\right] + \left[1 - w(s,\xi)\right] E_\xi V(s_{i^*}, \xi') + w(s,\xi) E_\xi V(s_{i^*+1}, \xi')$$

- Observe that V' is now only on grid in the s dimension!

Policy functions: Linear interpolation weights

Now assume the evolution of ξ is given by a DMC.

- then the linearly interpolated value function is given by

$$\begin{aligned} V(s_i,\xi_j) &= u\left[s_h, s^*(s_i,\xi_j)\right] \\ &+ \sum_{j'} p_{jj'}\left\{ \left[1-w(s_i,\xi_j)\right] V(s_{i^*},\xi_{j'}) + w(s_i,\xi_j)V(s_{i^*+1},\xi_{j'})\right\} \end{aligned} \quad (11)$$

- Observe that V' fully on grid!

Policy functions: Linear interpolation weights

Moreover, we can reinterpret the weights

- Define

$$\gamma_{(i,j)\to(i',j')} = \begin{cases} p_{jj'}\left[1 - w(s_i, \xi_j)\right] & \text{if } i' = i^*(i,j) \\ p_{jj'} w(s_i, \xi_j) & \text{if } i' = i^*(i,j) + 1 \\ 0 & \text{else} \end{cases} \tag{12}$$

- Then $\Gamma := \left[\gamma_{(i,j)\to(i',j')}\right]_{(i,j)}^{(i',j')}$ is a stochastic (transition) matrix
- of a DMC on the vectorized state space.

Policy functions: Transition probability matrix

Why is this useful?

- We can reinterpret the linear interpolant:
  The decision maker chooses only (fair) lotteries over on-grid points.

$$V_t = \max_{s'} u(s,s') + \beta\Gamma_{s'}V_{t+1}$$

- This we can use to obtain the ergodic distribution of states the planning problem induces **without simulation** and therefore fast.
- If an ergodic distribution exists, it is given by the left unit eigenvector of $\Gamma = [\gamma(i,j)]_{i=1\ldots N}^{j=1\ldots N}$, as

$$\mu_{t+1} = \mu_t\Gamma.$$

Standard Incomplete Markets Model (SIM): Setup

Recursive Dynamic Planning Problem

Consider a household problem in presence of aggregate and idiosyncratic risk

- S_t is an (exogenous) aggregate state
- s_{it} is a partly endogenous idiosyncratic state
- μ_t is the distribution over s

- Bellman equation:

$$v(s_{it}, S_t, \mu_t) = \max_{x \in \Gamma(s_{it}, P_t)} u(s_{it}, x) + \beta \mathbb{E} v(s_{it+1}(x, s_{it}), S_{t+1}, \mu_{t+1})$$

Recursive Dynamic Planning Problem

Consider a household problem in presence of aggregate and idiosyncratic risk

- S_t is an (exogenous) aggregate state
- s_{it} is a partly endogenous idiosyncratic state
- μ_t is the distribution over s

- Euler equation:

$$u'[x(s_{it}, S_t, \mu_t)] = \beta R(S_t, \mu_t) \mathbb{E} u'[x(s_{it+1}, S_{t+1}, \mu_{t+1})],$$

No aggregate risk

Recall how to solve for a stationary equlibrium:

- Discretize the state space (vectorized)
- Optimal policy $\bar{h}(s_{it}; P)$ induces flow utility $\bar{u}_{\bar{h}}$ and transition probability matrix $\Pi_{\bar{h}}$

No aggregate risk

- Discretized Bellman equation

$$\bar{v} = \bar{u}_{\bar{h}} + \beta \Pi_{\bar{h}} \bar{v} \tag{13}$$

holds for optimal policy (assuming a linear interpolant for the continuation value)

- and for the law of motion for the distribution (histograms)

$$d\bar{\mu} = d\bar{\mu} \Pi_{\bar{h}} \tag{14}$$

No aggregate risk

Equilibrium requires

- $\bar{h}$ is the optimal policy given P and v (being a linear interpolant)
- $\bar{v}$ and $d\bar{\mu}$ solve (13) and (14)
- Markets clear (some joint requirement on $\bar{h}, \mu, P$, denoted as $\Phi(\bar{h}, \mu, P) = 0$)

This can be solved for efficiently

- $d\bar{\mu}$ is vector corresponding to the unit-eigenvalue of $\Pi_{\bar{h}}$
- Using fast solution techniques for the DP, e.g. EGM
- Using a root-finder to solve for P

Equilibrium

- Market clearing depends on the specific model

Equilibrium

- Market clearing depends on the specific model
- In an Aiyagari model, we require that

$$r + \delta = F_K(K, L), w = F_L(K, L)$$

where in the most simple case aggregate labor supply is exogenously given. Then, prices are only a function of K and the equilibrium condition is simply

$$K^S(P(K)) = K$$

Equilibrium

- Market clearing depends on the specific model
- In an Aiyagari model, we require that

$$r + \delta = F_K(K, L), w = F_L(K, L)$$

where in the most simple case aggregate labor supply is exogenously given. Then, prices are only a function of K and the equilibrium condition is simply

$$K^S(P(K)) = K$$

- In a Huggett model, aggregate bond supply is zero and

$$K^S(r) = 0$$

is the equilibrium condition.

Exercise 3

Bewley model

Exercise

Solve the consumption savings problem by EGM . Obtain the stationary distribution of households using Young's method. Calculate average savings.

Exercise 4

Huggett model

Exercise

Use the code from Exercise 3 and find the equilibrium interest rate such that net aggregate savings are zero (households can only trade IOUs).

Exercise 5

Aiyagari model

Exercise

Solve the Aiyagari model, again using the EGM and Young's method. Here household can save in capital. The production function is

$$F(K, N) = K^{\alpha} N^{1-\alpha},$$

where $N = n_i$ is exogenous.

Transition Path or MIT shock or Sequence Space

See Kurt's slides.

See Adrien's slides.

Introducing aggregate risk

With aggregate risk

- Prices and distributions change over time

Yet, for the household

- Only prices and continuation values matter
- Distributions do not influence the decisions directly

Redefining equilibrium (Reiter, 2002)

A **sequential equilibrium with recursive individual planning**

- A sequence of discretized Bellman equation, such that

$$v_t = \bar{u}_{P_t} + \beta \Pi_{h_t} v_{t+1} \tag{15}$$

  holds for optimal policy, h_t (which results from v_{t+1} and P_t)
- and a sequence of histograms, such that

$$d\mu_{t+1} = d\mu_t \Pi_{h_t} \tag{16}$$

  holds given the optimal policy
- (Policy functions, h_t, that are optimal given P_t, v_{t+1})
- Prices, distributions and policies lead to market clearing

Compact notation (Schmitt-Grohé and Uribe, 2004)

The equilibrium conditions as a non-linear difference equation

- Controls: $Y_t = \begin{bmatrix} v_t & P_t & Z_t^Y \end{bmatrix}$ and
- States: $X_t = \begin{bmatrix} \mu_t & S_t & Z_t^X \end{bmatrix}$
  where Z_t are purely aggregate states/controls
- Define

$$F\left(d\mu_t, S_t, d\mu_{t+1}, S_{t+1}, v_t, P_t, v_{t+1}, P_{t+1}, \varepsilon_{t+1}\right) \tag{17}$$
$$= \begin{bmatrix} d\mu_{t+1} - d\mu_t \Pi_{h_t} \\ v_t - \left(\bar{u}_{h_t} + \beta \Pi_{h_t} v_{t+1}\right) \\ S_{t+1} - H(S_t, d\mu_t, \varepsilon_{t+1}) \\ \Phi(h_t, d\mu_t, P_t, S_t) \\ \varepsilon_{t+1} \end{bmatrix}$$

s.t.

$$h_t(s) = \arg \max_{x \in \Gamma(s, P_t)} u(s, x) + \beta \mathbb{E} v_{t+1}(s') \tag{18}$$

Compact notation (Schmitt-Grohé and Uribe, 2004)

In words

- First set of equations: Difference of one forward iteration of the distribution to assumed value.
- Second set of equations: Difference of one backward iteration of the value function (or policy functions in EGM) to assumed value.
- Last two sets of equations: Macro model.

Compact notation (Schmitt-Grohé and Uribe, 2004)

The equilibrium conditions as a non-linear difference equation

- Function-valued difference equation $\mathbb{E}F(X_t, X_{t+1}, Y_t, Y_{t+1}, \varepsilon_{t+1}) = 0$
- turns real-valued when we replace the functions by their discretized counterparts
- Standard techniques to solve by perturbation (Dynare etc)

Perturbation:
Some Theory and Applications

Perturbation References: General

- General:
  1. A First Look at Perturbation Theory by James G. Simmonds and James E. Mann Jr.
  2. Advanced Mathematical Methods for Scientists and Engineers: Asymptotic Methods and Perturbation Theory by Carl M. Bender, Steven A. Orszag.

- This lecture:
  1. "Perturbation Methods for General Dynamic Stochastic Models" by Hehui Jin and Kenneth Judd.
  2. "Computational Methods for Economists" by Jesus Fernandez-Villaverde.
  3. "Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function" by Martin Uribe and Stephanie Schmitt-Grohe.

Non-linear difference equation

- A large class of economic models can be written as a set of non-linear difference equations of the form

$$E_t f(s_{t+1}, s_t, c_{t+1}, c_t) = 0$$

where s are all state and c are all control variables now.

Perturbation methods

- More generally, functional equations of the form:

$$\mathcal{H}(d) = 0$$

  for an unknown decision rule d.
- Perturbation solves the problem by specifying:

$$d^n(x, \theta) = \sum_{i=0}^{n} \theta_i (x - x_0)^i$$

- We use implicit-function theorems to find coefficients θ_i's
- Inherently local approximation.

Motivation

- Many complicated mathematical problems have:
  - either a particular case
  - or a related problem

  that is easy to solve.

- Often, we can use the solution of the simpler problem as a building block of the general solution.

- Sometimes perturbation is known as asymptotic methods.

A simple example

- Imagine we want to compute $\sqrt{26}$ by hand
- Note that:

$$\sqrt{26} = \sqrt{25 * 1.04} = 5 * \sqrt{1.04} \approx 5 * 1.02 = 5.1$$

- Exact solution: $\sqrt{26} = 5.09902$
- More generally:

$$\sqrt{x} = \sqrt{y^2 * (1 + \epsilon)} = y * \sqrt{(1 + \epsilon)} \approx y * (1 + \epsilon)$$

- Accuracy depends on how big ϵ is

Applications in economics

- Judd and Guu (1993) showed how to apply it to economic problems
- Recently, perturbation methods have been gaining much popularity
- In particular, second- and third-order approximations are easy to compute and notably improve accuracy
- Perturbation theory is the generalization of the well-known linearization strategy
- Hence, we can use much of what we already know about linearization

Regular versus singular perturbations

- Regular perturbation: a small change in the problem induces a small change in the solution.
- Singular perturbation: a small change in the problem induces a large change in the solution.

- Example: excess demand function.
- Most problems in economics involve regular perturbations.
- Sometimes, however, we can have singularities.
  Example: introducing a new asset in an incomplete market model.

Perturbation methods

- Back to our economic model cast in the following form:

$$E_t f(s_{t+1}, s_t, c_{t+1}, c_t) = 0$$

  where s are state and c are control variables.

- Rewrite f introducing a parameter for uncertainty :

$$E_t f(s_{t+1}, s_t, c_{t+1}, c_t; \sigma) = 0$$

  .

Evolution of states

- Dynamic stochastic general equilibrium models in addition have a structure where a subset of state variables s_t^1 are predetermined and endogenous while the remainder are exogenously driven as

$$s_{t+1}^2 = H_2(s_t^2, \sigma) + \sigma \Sigma \epsilon_{t+1}$$

  where ϵ_{t+1} are i.i.d. with zero mean, unit covariance and bounded support.
- Stacking all state variables, we can write

$$s_{t+1} = [H_1(s_t, \sigma); H_2(s_t, \sigma)] + \sigma \eta \epsilon_{t+1}$$

- with $\eta = [0; \Sigma]$

Evolution of controls

- That c_t is a control means that there is a function $c_t = G(s_t, \sigma)$
- The goal is to solve for the unknown H_1 and G.

Local approximation

- Take a Taylor series approximation of G and H:

$$\begin{aligned} G(s,\sigma) &= G(s^*,\sigma^*) + G_s(s^*,\sigma^*)(s-s^*) + G_\sigma(s^*,\sigma^*)(\sigma-\sigma^*) \\ &+ 1/2G_{ss}(s^*,\sigma^*)(s-s^*)^2 + G_{s\sigma}(s^*,\sigma^*)(s-s^*)(\sigma-\sigma^*) \\ &+ 1/2G_{\sigma\sigma}(s^*,\sigma^*)(\sigma-\sigma^*)^2 + ... \end{aligned}$$

$$\begin{aligned} H(s,\sigma) &= H(s^*,\sigma^*) + H_s(s^*,\sigma^*)(s-s^*) + H_\sigma(s^*,\sigma^*)(\sigma-\sigma^*) \\ &+ 1/2H_{ss}(s^*,\sigma^*)(s-s^*)^2 + H_{s\sigma}(s^*,\sigma^*)(s-s^*)(\sigma-\sigma^*) \\ &+ 1/2H_{\sigma\sigma}(s^*,\sigma^*)(\sigma-\sigma^*)^2 + ... \end{aligned}$$

Local approximation

- Replace s and c in F:

$$\begin{aligned} F(s,\sigma) \equiv & E_t f\left(H(s_t,\sigma) + \sigma\eta\epsilon_{t+1}, s_t, G[H(s_t,\sigma) + \sigma\eta\epsilon_{t+1},\sigma], G(s_t,\sigma)\right) \\ = & \, 0 \end{aligned}$$

- The goal is to solve for the unknown H_1 and G.
- Local approximation means that we solve for H_1, G by taking a Taylor expansion of F around the non-stochastic steady state s^*, for which $\sigma^* = 0$.

Local approximation

- Define non-stochastic steady state as vectors (s^*, c^*) :

$$f(s^*, s^*, c^*, c^*) = 0$$

- $c^* = G(s^*, 0)$ and $s^* = H(s^*, 0)$
- Note that if $\sigma = 0$, then $E_t f = f$

Local approximation

- Approximation of G and H around the point $(s,\sigma) = (s^*,0)$

$$G(s,0) = G(s^*,0) + G_s(s^*,0)(s - s^*) + G_\sigma(s^*,0)\sigma$$
$$H(s,0) = H(s^*,0) + H_s(s^*,0)(s - s^*) + H_\sigma(s^*,0)\sigma$$

- $G(s^*,0), H(s^*,0)$ identified by steady state values
- Remaining coefficients are identified by:

$$F_s(s^*,0) = 0$$
$$F_\sigma(s^*,0) = 0$$

Local approximation

- Take derivative of F w.r.t. uncertainty :

$$\begin{aligned} F_\sigma(s^*,0) &= E_t f_{s'}(H_\sigma + \eta\epsilon') + f_{c'}[G_s(H_\sigma + \eta\epsilon') + G_\sigma] + f_c G_\sigma \\ &= f_{s'} H_\sigma + f_{c'}[G_s H_\sigma + G_\sigma] + f_c G_\sigma \\ &= 0 \end{aligned}$$

- This is a system of n equations:

$$\begin{pmatrix} f_{s'} + f_{c'} G_s & f_{c'} + f_c \end{pmatrix} \begin{pmatrix} H_\sigma \\ G_\sigma \end{pmatrix} = 0$$

- This equation is linear and homogeneous in H_σ, G_σ. Thus we have that $H_\sigma = 0$ and $G_\sigma = 0$.

Local approximation

Important theoretical result:

- In words, up to first order, we do not need to adjust the steady state solution when changing aggregate risk σ.
- Expected values of s_t and c_t are equal to their non-stochastic steady-state values.
- In a first order approximation the certainty equivalence principle holds, i.e., the policy function is independent of the variance-covariance matrix of ϵ.
- Interpretation: no precautionary behavior.

Local approximation

- Differentiation w.r.t s yields:

$$F_s(s^*, 0) = f_{s'} H_s + f_s + f_{c'} G_s H_s + f_c G_s = 0$$

- In matrix form:

$$\begin{pmatrix} f_{s'} & f_{c'} \end{pmatrix} \begin{pmatrix} I \\ G_s \end{pmatrix} H_s = -\begin{pmatrix} f_s & f_c \end{pmatrix} \begin{pmatrix} I \\ G_s \end{pmatrix}$$

Local approximation

- Let $A = [f_{s'} \quad f_{c'}]$ and $B = [f_s \quad f_c]$
- Let $\hat{s}_t \equiv s_t - s^*$, then postmultiply:

$$A \begin{pmatrix} I \\ G_s \end{pmatrix} H_s \hat{s}_t = -B \begin{pmatrix} I \\ G_s \end{pmatrix} \hat{s}_t$$

- Consider a perfect foresight equilibrium. In this case, $H_s \hat{s}_t = \hat{s}_{t+1}$

$$A \begin{pmatrix} I \\ G_s \end{pmatrix} \hat{s}_{t+1} = -B \begin{pmatrix} I \\ G_s \end{pmatrix} \hat{s}_t$$

Local approximation

- This leaves us with a system of quadratic equations that we need to solve for H_s, G_s.
- Procedures to solve rational expectations models:
  1. Blanchard and Kahn (1980).
  2. Uhlig (1999).
  3. Sims (2000).
  4. Klein (2000).

Local properties of the solution I

- Perturbation is a local method.
- It approximates the solution around the deterministic steady state of the problem.
- It is valid within a radius of convergence.

Local properties of the solution II

- What is the radius of convergence of a power series around x? An $r \in \mathcal{R}_{+}^{\infty}$ such that $\forall x', |x' - z'| < r$, the power series of x will converge.
- A Remarkable Result from Complex Analysis:
  The radius of convergence is always equal to the distance from the center to the nearest point where the decision rule has a (non-removable) singularity. If no such point exists then the radius of convergence is infinite.
- Singularity here refers to poles, fractional powers, and other branch powers or discontinuities of the functional or its derivatives.

Solution

- Using an eigenvalue decomposition of $H_s = P\Lambda P^{-1}$ we obtain,

$$AZ\Lambda = BZ \quad Z = [I; G_s]P$$
$$A = [f_{s'}, f_{c'}] \quad B = -[f_s, f_c]$$

which implies that the solution corresponds to a subset of the solutions to the generalized eigenvalue problem

$$AXD = BX$$

- But which?
- If we have a stable system, then $\lim_{t\to\infty} H_s^t = 0$. Therefore, we are searching for the exactly n_s eigenvalues smaller than unity.

Solution

- Splitting all solutions to the eigenvalue problem above and below eigenvalues of 1, we obtain

$$A\begin{bmatrix} X_{11} & X_{21} \\ X_{12} & X_{22} \end{bmatrix}\begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} = B\begin{bmatrix} X_{11} & X_{21} \\ X_{12} & X_{22} \end{bmatrix}$$

and therefore

$$H_s = X_{11}D_1X_{11}^{-1}$$

and

$$G_s = X_{12}X_{11}^{-1}$$

Alternative Solution: Time Iteration

- Rendahl (2018) extends linear time iteration
- Intuitive, robust, and easy to implement algorithm. Now x is vector of all controls and states. F_n is transition matrix of states and controls.

- Rewrite difference equation in "end of last period" notation

$$Ax_{t-1} + Bx_t + Cx_{t+1} = 0$$

- Let F_n be a candidate solution:

$$Ax_{t-1} + Bx_t + CF_n x_t = 0$$
$$x_t = -(B + CF_n)^{-1} Ax_{t-1}$$

- Thus update guess F_{n+1} as

$$F_{n+1} = -(B + CF_n)^{-1} A$$

Higher order approximations

- Obtaining higher-order approximations to the solution of the non-linear system is a sequential procedure.
- The coefficients of the ith term of the jth-order approximation are given by the coefficients of the ith term of the ith order approximation, for $j > 1$ and $i < j$.
- More importantly, obtaining the coefficients of the ith order terms of the approximate solution given all lower-order coefficients involves solving a linear system of equations.

Notation: Tensors

- General trick from physics.
- An n^{th} -rank tensor in a m-dimensional space is an operator that has n indices and m^n components and obeys certain transformation rules.
- $[F_y]^i_\alpha$ is the (i, α) element of the derivative of F with respect to y:
  1. The derivative of F with respect to y is an n x n_y matrix.
  2. Thus, $[F_y]^i_\alpha$ is the element of this matrix located at the intersection of the i-th row and α-th column.
  3. Thus, $[F_y]^i_\alpha [G_x]^\alpha_\beta [H_x]^\beta_j = \sum_{\alpha=1}^{n_y} \sum_{\beta=1}^{n_x} \frac{\partial F^i}{\partial y^\alpha} \frac{\partial G^\alpha}{\partial x^\beta} \frac{\partial H^\beta}{\partial x^j}$
- $[F_{yy}]^i_{\alpha\gamma}$
  1. F^i_{yy} is a three dimensional array with n rows, n_y columns, and n_y pages.
  2. $[F_{yy}]^i_{\alpha\gamma}$ denotes the element at the intersection of row i, column α, and page γ

Second order approximation

- Derivatives of $F(s,\sigma)$:

$$[F_{ss}(s^*,0)]^i_{jk} = 0$$
$$[F_{\sigma\sigma}(s^*,0)]^i = 0$$
$$[F_{s\sigma}(s^*,0)]^i_j = 0$$

Second order approximation

- Cross derivatives are equal to zero when evaluated at $(s^*, 0)$:

$$[F_{\sigma s}(s^*,0)]^i_j = [F_{s'}]^i_\beta [H_{\sigma s}]^\beta_j + [F_{c'}]^i_\alpha [G_s]^\alpha_\beta [H_{\sigma s}]^\beta_j + [F_{c'}]^i_\alpha [G_{\sigma s}]^\alpha_\gamma [H_s]^\gamma_j + [F_c]^i_\alpha [G_{\sigma s}]^\alpha_j = 0$$

- This is a system of n x n_s equations in the n x n_s unknowns given by the elements of $G_{\sigma s}$ and $H_{\sigma s}$.
- The system is homogeneous in the unknowns. Thus, if a unique solution exists, it is given by $G_{\sigma s} = 0$ and $H_{\sigma s} = 0$.

Second order approximation

Important theoretical result:

- The coefficients of the policy function on the terms that are linear in the state vector do not depend on the size of the variance of the underlying shocks
- Uncertainty only affects the constant term in the policy function

Second order approximation

- Approximation of G and H around the point $(s,\sigma) = (s^*, 0)$

$$
\begin{aligned}
[G(s,\sigma)]^i = & [G(s^*,0)]^i + [G_s(s^*,0)]^i_a [(s-s^*)]_a \\
& + \frac{1}{2}[G_{ss}(s^*,0)]^i_{ab}[(s-s^*)]_a[(s-s^*)]_b \\
& + \frac{1}{2}[G_{\sigma\sigma}(s^*,0)]^i[\sigma^2]
\end{aligned}
$$

$$
\begin{aligned}
[H(s,\sigma)]^j = & [H(s^*,0)]^j + [H_s(s^*,0)]^j_a [(s-s^*)]_a \\
& + \frac{1}{2}[H_{ss}(s^*,0)]^j_{ab}[(s-s^*)]_a[(s-s^*)]_b \\
& + \frac{1}{2}[H_{\sigma\sigma}(s^*,0)]^j[\sigma^2]
\end{aligned}
$$

Second order approximation

- Approximation of G and H around the point $(s,\sigma) = (s^*,0)$ ctd
- The unknowns of this expansion are $[G_{ss}(s^*,0)]^i$, $[G_{\sigma\sigma}(s^*,0)]^i$, $[H_{ss}(s^*,0)]^j$, and $[H_{\sigma\sigma}(s^*,0)]^j$
- Derivatives of $F(s,\sigma)$ yield as many equations as we have unknowns. Perfectly identified linear system!

Second order approximation

$$
\begin{aligned}
[F_{xx}(\bar{x},0)]^i_{jk} = & ([f_{y'y'}]^i_{\alpha\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{y'y}]^i_{\alpha\gamma}[g_x]^\gamma_k \\
& + [f_{y'x'}]^i_{\alpha\delta}[h_x]^\delta_k + [f_{y'x}]^i_{\alpha k})[g_x]^\alpha_\beta[h_x]^\beta_j \\
& + [f_{y'}]^i_\alpha[g_{xx}]^\alpha_{\beta\delta}[h_x]^\delta_k[h_x]^\beta_j \\
& + [f_{y'}]^i_\alpha[g_x]^\alpha_\beta[h_{xx}]^\beta_{jk} \\
& + ([f_{yy'}]^i_{\alpha\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{yy}]^i_{\alpha\gamma}[g_x]^\gamma_k + [f_{yx'}]^i_{\alpha\delta}[h_x]^\delta_k + [f_{yx}]^i_{\alpha k})[g_x]^\alpha_j \\
& + [f_y]^i_\alpha[g_{xx}]^\alpha_{jk} \\
& + ([f_{x'y'}]^i_{\beta\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{x'y}]^i_{\beta\gamma}[g_x]^\gamma_k + [f_{x'x'}]^i_{\beta\delta}[h_x]^\delta_k + [f_{x'x}]^i_{\beta k})[h_x]^\beta_j \\
& + [f_{x'}]^i_\beta[h_{xx}]^\beta_{jk} \\
& + [f_{xy'}]^i_{j\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{xy}]^i_{j\gamma}[g_x]^\gamma_k + [f_{xx'}]^i_{j\delta}[h_x]^\delta_k + [f_{xx}]^i_{jk} \\
= & 0; \quad i = 1,\ldots,n, \quad j,k,\beta,\delta = 1,\ldots,n_x; \quad \alpha,\gamma = 1,\ldots,n_y.
\end{aligned}
$$

- System of $n x n_s x n_s$ linear equations in the $n x n_s x n_s$ unknowns given by the elements of G_{ss} and H_{ss}.

Higher order approximations

- We can iterate this procedure as many times as we want.
- We can obtain n-th order approximations.
- Levintal (2017) uses tensor-unfolding to work with higher-order derivatives

- Problems:
  1. Existence of higher order derivatives.
  2. Numerical instabilities.
  3. Computational costs.

Example: Simple RBC

Stochastic neoclassical growth model

$$
\begin{aligned}
&max \quad E_0 \sum_{t=0}^{\infty} \beta^t \log c_t \\
\text{s.t.} \quad & c_t + k_{t+1} = e^{z_t} k_t^{\alpha} \\
& z_t = \rho z_{t-1} + \sigma \epsilon_t, \quad \epsilon \sim \mathcal{N}(0,1)
\end{aligned}
$$

- Note: full depreciation.
- Equilibrium conditions:

$$
\begin{aligned}
\frac{1}{c_t} &= \beta E_t \frac{1}{c_{t+1}} \alpha e^{z_{t+1}} k_{t+1}^{\alpha-1} \\
c_t + k_{t+1} &= e^{z_t} k_t^{\alpha} \\
z_t &= \rho z_{t-1} + \sigma \epsilon_t
\end{aligned}
$$

Solution and steady state

- Exact solution (found by "guess and verify"):

$$c_t = (1-\alpha\beta)e^{z_t}k_t^{\alpha}$$
$$k_t = (\alpha\beta)e^{z_t}k_t^{\alpha}$$

- Steady state is also easy to find:

$$k = (\alpha\beta)^{\frac{1}{1-\alpha}}$$
$$c = (\alpha\beta)^{\frac{\alpha}{1-\alpha}} - (\alpha\beta)^{\frac{1}{1-\alpha}}$$
$$z = 0$$

The goal

- We are searching for decision rules:

$$\begin{aligned} c_t &= c(k_t, z_t) \\ k_{t+1} &= k(k_t, z_t) \end{aligned}$$

- Then we have:

$$\begin{aligned} \frac{1}{c(k_t, z_t)} &= \beta E_t \frac{1}{c(k(k_t, z_t), z_{t+1})} \alpha e^{z_{t+1}} k(k_t, z_t)^{\alpha - 1} \\ c(k_t, z_t) + k(k_t, z_t) &= e^{z_t} k_t^{\alpha} \\ z_t &= \rho z_{t-1} + \sigma \epsilon_t \end{aligned}$$

- This is a system of functional equations (after substituting z_t)

A perturbation solution

- Add perturbation parameter σ
  - When $\sigma = 0$ deterministic case (with $z_0 = 0$ and $e^{z_t} = 1$)
  - When $\sigma > 0$ stochastic case
- Now we are searching for decision rules:

$$
\begin{aligned}
c_t =& c(k_t, z_t; \sigma) \\
k_{t+1} =& k(k_t, z_t; \sigma)
\end{aligned}
$$

Taylor's theorem

- We will build a local approximation around $(k^*, 0; 0)$
- Given equilibrium conditions:

$$\frac{1}{c(k_t, z_t; \sigma)} = \beta E_t \frac{1}{c(k(k_t, z_t; \sigma), \rho z_t + \sigma \epsilon_{t+1}; \sigma)} \alpha e^{\rho z_t + \sigma \epsilon_{t+1}} k(k_t, z_t; \sigma)^{\alpha - 1}$$
$$c(k_t, z_t; \sigma) + k(k_t, z_t; \sigma) = e^{\rho z_{t-1} + \sigma \epsilon_t} k_t^{\alpha}$$

- Take derivatives w.r.t. k_t, z_t, σ and evaluate them around $(k^*, 0; 0)$

Compact Notation

$$F(k_t, z_t, \sigma) = E_t \begin{pmatrix} \frac{1}{c(k_t,z_t;\sigma)} - \beta E_t \frac{\alpha e^{\rho z_t + \sigma \epsilon_{t+1}} k(k_t,z_t;\sigma)^{\alpha-1}}{c(k(k_t,z_t;\sigma),\rho z_t + \sigma \epsilon_{t+1};\sigma)} \\ c(k_t, z_t; \sigma) + k(k_t, z_t; \sigma) - e^{\rho z_{t-1} + \sigma \epsilon_t} k_t^{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- Note that:

$$\begin{aligned} F(k_t, z_t, \sigma) &= \mathcal{H}(c_t, c_{t+1}, k_t, k_{t+1}, z_t; \sigma) \\ &= \mathcal{H}(c(k_t, z_t; \sigma), c(k(k_t, z_t; \sigma), z_t; \sigma), k_t, k(k_t, z_t; \sigma), z_t; \sigma) \end{aligned}$$

- Because $F(k_t, z_t, \sigma)$ must be equal to zero for any possible values of k , z , and σ, the derivatives of any order of F must also be zero.

First-order approximation

- Take first-order derivatives of $F(k_t, z_t, \sigma)$ around $(k^*, 0; 0)$

$$F_k(k, 0; 0) = 0$$
$$F_z(k, 0; 0) = 0$$
$$F_\sigma(k, 0; 0) = 0$$

Second-order approximation

- Take second-order derivatives of $F(k_t, z_t, \sigma)$ around $(k^*, 0; 0)$

$$F_{kk}(k, 0; 0) = 0$$
$$F_{kz}(k, 0; 0) = 0$$
$$F_{k\sigma}(k, 0; 0) = 0$$
$$F_{zz}(k, 0; 0) = 0$$
$$F_{z\sigma}(k, 0; 0) = 0$$
$$F_{\sigma\sigma}(k, 0; 0) = 0$$

- We substitute the coefficients that we already know.
- A linear system of 12 equations on 12 unknowns.
- Cross-terms on $k\sigma$ and $z\sigma$ are zero.
- More general result: all the terms in odd derivatives of σ are zero.

Correction for risk

- We have the term $1/2c_{\sigma\sigma}(k,0;0)$
- Captures precautionary behavior.
- We do not have certainty equivalence any more!
- Important advantage of second order approximation.
- Changes ergodic distribution of states.

Higher-order terms

- We can continue the iteration for as long as we want.
- Great advantage of procedure: it is recursive!
- Often, a few iterations will be enough.
- The level of accuracy depends on the goal of the exercise: e.g. Welfare analysis: Kim and Kim (2001).

Exercise 6

Exercise

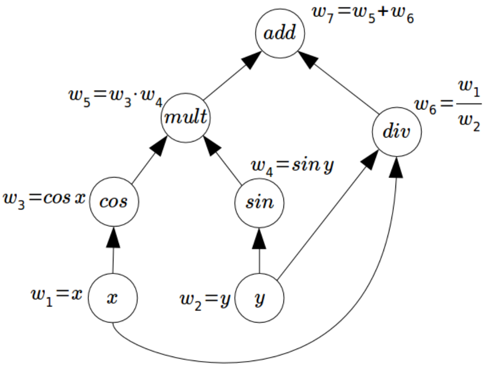
Solve the simple stochastic growth model using perturbation methods. For this purpose, first write a function that calculates the Euler equation errors, errors from capital accumulation, and the law of motion for productivity. Define consumption as control and capital and productivity as states. Compare the first and second order perturbation of $c(k, z, \sigma)$ *to the true solution.*

Excursus: Automatic differentiation

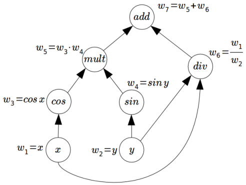
- Modern computer languages like Julia offer easy to implement automatic differentiation libraries.
- Automatic differentiation is neither:
  - Symbolic differentiation
    - Inefficient code
  - nor Numerical differentiation (the method of finite differences)
    - If you make h too small, then your accuracy gets killed by floating point roundoff
    - If h is too big, then approximation errors start ballooning
- AD avoids these problems: it calculates exact derivatives, so your accuracy is only limited by floating point error.

Excursus: Automatic differentiation

- AD applies the chain rule to your function
- Any complicated function f can be rewritten as the composition of a sequence of primitive functions
- Let $f(x, y) = \cos x \sin y + \frac{x}{y}$



Excursus: Automatic differentiation



$$
\begin{aligned}
Df &= Dw_7 = D(w_5 + w_6) = Dw_5 + Dw_6 \\
Dw_6 &= D\frac{w_1}{w_2} = \frac{w_1 Dw_2 - w_2 Dw_1}{w_2^2} \\
Dw_5 &= Dw_3 w_4 = w_3 Dw_4 + w_4 Dw_3 \\
Dw_4 &= D \sin w_2 = \cos w_2 \cdot Dw_2 \\
Dw_3 &= D \cos w_1 = -\sin w_1 \cdot Dw_1 \\
Dw_2 &= Dy \\
Dw_1 &= Dx
\end{aligned}
$$

Excursus: Automatic differentiation

- AD is implemented by a nonstandard interpretation of the program in which real numbers are replaced by dual numbers and the numeric primitives are lifted to operate on dual numbers.
- Dual numbers: Replace every number x with the number $x + x'\varepsilon$, where x' is a real number, but ε is an abstract number with the property $\varepsilon^2 = 0$

- Julia does this for you!
- ForwardDiff Package: `www.juliadiff.org/`
- Examples: `www.juliadiff.org/ForwardDiff.jl/stable/user/advanced.html`

Excursus: Julia

- Julia combines three key features for highly intensive computing tasks as perhaps no other contemporary programming language does: it is fast, easy to learn and use, and open source.

- Introduction by Fernandez-VillaVerde:
  `www.sas.upenn.edu/~jesusfv/Chapter_HPC_8_Julia.pdf`
- Introduction by QuantEcon:
  `https://lectures.quantecon.org/jl/`

Back to heterogeneous agent model

The equilibrium conditions as a non-linear difference equation

- Controls: $Y_t = \begin{bmatrix} v_t & P_t & Z_t^Y \end{bmatrix}$ and
- States: $X_t = \begin{bmatrix} \mu_t & S_t & Z_t^X \end{bmatrix}$
  where Z_t are purely aggregate states/controls
- Define

$$F\left(d\mu_t, S_t, d\mu_{t+1}, S_{t+1}, v_t, P_t, v_{t+1}, P_{t+1}, \varepsilon_{t+1}\right) \tag{19}$$

$$= \begin{bmatrix} d\mu_{t+1} - d\mu_t \Pi_{h_t} \\ v_t - \left(\bar{u}_{h_t} + \beta \Pi_{h_t} v_{t+1}\right) \\ S_{t+1} - H(S_t, d\mu_t, \varepsilon_{t+1}) \\ \Phi(h_t, d\mu_t, P_t, S_t) \\ \varepsilon_{t+1} \end{bmatrix}$$

s.t.

$$h_t(s) = \arg \max_{x \in \Gamma(s, P_t)} u(s, x) + \beta \mathbb{E} v_{t+1}(s') \tag{20}$$

So, is all solved?

The dimensionality of the system F is still an issue

- With high dimensional idiosyncratic states, discretized value functions and distributions become large objects
- For example:
  4 income states (grid points)
  $\times$ 100 illiquid asset states
  $\times$ 100 liquid asset states
  $\implies \geq 40{,}000$ control variables in F
- Same number of state variables

Perturbing SIM:
Reduction Methods

Reiter (2009): Reduce dimensionality ex ante

Problem:

- Dimensionality of the difference equation is large

Proposal:

- Reduce dimensionality ex ante (before solving the StE)
- e.g. (sparse) splines to represent policy functions
- Then linearize

Winberry (2016) extends this to the distribution functions

What we do

Proposal:

- Reduce dimensionality after StE, but before linearization
- Extract from the StE the *important* basis functions to represent individual policies (akin to image compression)
- Perturb only those basis functions but use the StE as “reference frame” for the policies (akin to video compression)
- Similarly for distributions (details later)

Our idea

1.) Apply compression techniques as in video encoding

- Apply a discrete cosine transformation to all value/policy functions (Chebycheff polynomials on roots grid)
- Define as reference "frame": the StE value/policy function
- Write fluctuations as differences from this reference frame
- Assume all coefficients of the DCT from the StE close to zero do not change after shock

Our idea

2.) Transform joint-distribution μ into copula and marginals

- Calculate the Copula, $\bar{\Xi}$ of μ in the StE
- Perturb the marginal distributions
- Approximate changes in the Copula (via DCT) or use fixed Copula to calculate an approximate joint distribution from marginals

- Idea follows Krusell and Smith (1998) in that some moments of the distribution do not matter for aggregate dynamics

Copula

A distribution of probabilities

A *Copula* is a joint distribution function of univariate marginal probabilities for a multivariate stochastic variable. It maps $[0,1]^n \rightarrow [0,1]$

Sklar's theorem

Every distribution function F can be represented by the marginal distribution functions F_i and a *Copula*, Ξ, with $F(x_1, \ldots, x_n) = \Xi\left[F_1(x_1), \ldots, F_n(x_n)\right]$.

Details

1.) Apply compression techniques as in video encoding

- DCT yields the coefficients of the fitted (multi-dimensional) Chebyshev polynomial, where the polynomial is constructed such that the tensor grid for s is mapped to the Chebyshev knots.
  See Ahmed et al. (1974) for the seminal contribution.
- Importantly, the absolute value of the coefficients has an interpretation in terms of the R^2 contribution of the corresponding polynomial in fitting the data.
- This allows us to order and select the polynomial terms based on their importance.

Excursus: Global polynomial

- Express a function by the coefficients ψ of a polynomial

$$\hat{f}(x) = \sum_{j=1}^{n} \psi_j c_j(x)$$

  where $c_i(x)$ are known basis functions such as $c_j(x) = x^j$.
- Better than ordinary polynomials are usually Chebyshev polynomials of which the baseline functions are

$$c_j(x) = \cos(j \ \arccos x)$$

- These are orthogonal on [-1,1], i.e.

$$\int_{-1}^{1} c_i(x) c_j(x) \frac{1}{\sqrt{1-x^2}} dx = 0 \forall i \neq j$$

Excursus: Global polynomial

- Since the evaluation points x_i are known ("grid"), we can compute $\mathbf{C} = [c_j(x_i)]_{i=1...M, j=1...n}$
- The vector of function values $\hat{\mathbf{f}} = [\hat{f}(x_i)]_{i=1...M}$ is then given by

$$\hat{\mathbf{f}} = \mathbf{C}\psi$$

- Therefore, we can obtain an optimal (minimal MSE) as

$$\psi^* = (\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}'\ \mathbf{f}$$

- The big **advantage** of polynomials is that they can be integrated analytically and that they are differentiable of any order.

Excursus: Global polynomial: issues

- **Runge's Phenomenon:** Since polynomials tend to infinity as $x \to \infty$ it is not true that the overall fit of a global polynomial gets better, if more grid points and higher order polynomials are used (oscillating behavior).
- Choosing **Chebyshev polynomials** as basis functions and
- grid points as the **roots** $x_i = \cos(\frac{2i-1}{2N})$ for $i = 1 \ldots N$ of these polynomials minimizes approximation error.

Excursus: Discrete Cosine Transforms

A first observation

- Suppose Chebychev root grid-points are not suitable for our problem.
- Then, we can write $f(x) = f(g(y))$ and
- generate the grid x_i by applying g to the Chebyshev nodes y_i,
- with basis functions $c_j(x) = \cos\left(j \ \arccos g^{-1}(x)\right)$

Discrete Cosine Transform (DCT) and lossy compression

- In particular, if we do not intend to evaluate off-grid, we do not need to know g but just the nodes $y_i = \cos\left(\frac{2i-1}{2N}\pi\right)$ and grid values x_i
- and obtain an equivalent representation of f_i in terms of coefficients.
- Shrinking ≈ 0-coefficients to 0 leaves $\hat{f}_i$ close to unchanged.
- In addition $C'C = I$.

Details

1.) Apply compression techniques as in video encoding

- Let $\bar{\Theta} = dct(\bar{v})$ be the coefficients obtained from the DCT of the value function in StE

- A DCT expresses a finite sequence of data points in terms of sum of cosine functions at different frequencies
- Linear, invertible function $f = \Re^N - > \Re^N$ (equivalently: an invertible NxN matrix)
- x_n is transformed to X_k according to:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\pi/N(n + 1/2)k\right], k = 0, ..., N-1$$

Details

1.) Apply compression techniques as in video encoding

- Define an index set $\mathcal{I}$ that contains the x percent largest (i.e. most important) elements from $\bar{\Theta}$
- Let θ be a sparse vector with non-zero entries only for elements $i \in \mathcal{I}$
- Define $\tilde{\Theta}(\theta_t) = \begin{cases} \bar{\Theta}(i) + \theta_t(i) & i \in \mathcal{I} \\ \bar{\Theta}(i) & else \end{cases}$

Details

Decoding

- Now we reconstruct $v_t = v(\theta_t) = idct(\tilde{\Theta}(\theta_t))$
- This means that in the StE the reduction step adds no additional approximation error as $v(0) = \bar{v}$ by construction
- Yet, it allows to reduce the number of derivatives that need to be calculated from the outset

Details

2) Analogously for the distribution function

- Define μ_t as $\Xi_t(\bar{\mu}_t^1, \ldots, \bar{\mu}_t^n)$ for n being the dimensionality of the idiosyncratic states
- The StE distribution is obtained when $\mu = \bar{\Xi}(\bar{\mu}^1, \ldots, \bar{\mu}^n)$

- We can treat the copula as an interpolant defined on the grid of steady-state marginal distributions, and also approximate Ξ_t as a sparse expansion around the steady-state copula $\bar{\Xi}$.
- The most extreme variant of this is to treat the copula as time fixed.

Details

2) Analogously for the distribution function

- Typically prices are only influenced through the marginal distributions
- The approach ensures that changes in the mass of one dimension, say wealth, are distributed in a sensible way across the other dimensions
- The implied distributions look “similar” to the StE one

Obtaining the copula function of the StE

To obtain an estimate of the Copula of the StE:

1. Accumulate the histogram along every dimension to obtain CDF estimate, M.
2. Add a leading zero to the CDF matrix, M, along every dimension.
3. Calculate marginal distributions, m_i, from the CDF (summing out other dims)
4. Obtain the Copula estimate as an interpolant of M on $\{m_1, \dots, m_n\}$

$$\bar{\Xi} = \textsc{griddedInterpolant}(\{m_1, \dots, m_n\}, M)$$

.

Details

Too many equations

- The system

$$F\Big(\{d\mu_t^1,\dots,d\mu_t^n\},S_t,\{d\mu_{t+1}^1,\dots,d\mu_{t+1}^n\},S_{t+1},\\
\theta_t,P_t,\theta_{t+1},P_{t+1}\Big)=\\
\begin{bmatrix} d\bar{\Xi}(\bar{\mu}_t^1,\dots,\bar{\mu}_t^n)-d\bar{\Xi}(\bar{\mu}_t^1,\dots,\bar{\mu}_t^n)\Pi_{h_t} \\ dct\left[idct(\tilde{\Theta}(\theta_t))-\left(\bar{u}_{h_t}+\beta\Pi_{h_t}idct(\tilde{\Theta}(\theta_{t+1}))\right)\right] \\ S_{t+1}-H(S_t,d\mu_t) \\ \Phi(h_t,d\mu_t,P_t,S_t)\end{bmatrix} \tag{21}$$

  has too many equations
- Use only difference in marginals and the differences on $\mathcal{I}$

Quality of approximation

- David Childers (2018), "Automated Solution of Heterogeneous Agent Models":
- Under some regularity conditions the solution algorithm is guaranteed to converge to the first derivative of the true infinite dimensional solution as the discretization is refined.
- Convergence rates for the approximation are provided as well, depending on the choices of interpolation method including polynomials, splines, histograms, and wavelets.

Application: Krusell-Smith model

A simple KS economy

Incomplete Markets and TFP

- Household productivity can be high or low
- No contingent claims
- Households save in capital goods (which they rent out)
- Households supply labor (disutility) and consume (utility)
- Aggregate productivity (TFP) follows a log AR-1 process
- Cobb-Douglas production function

A simple KS economy

Numerical setup

- Asset grid has 100 points ($\implies$ a total grid size of 200)
- Policies solved by EGM (instead of VFI)

Different levels of “compression”

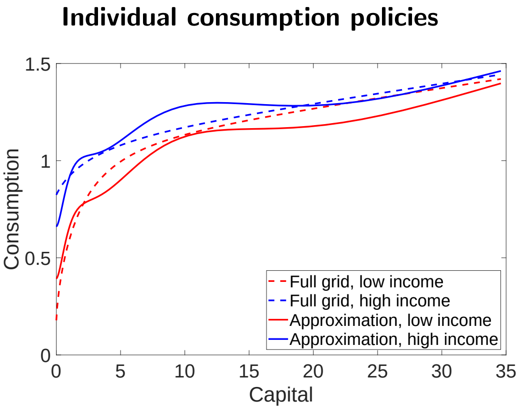


Figure: Policy and **10** most important basis functions

Different levels of “compression”

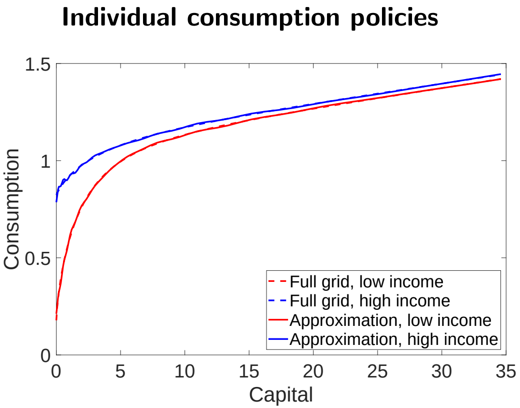


Figure: Policy and **50** most important basis functions

Different levels of “compression”

Individual policy response to a 20%TFP shock

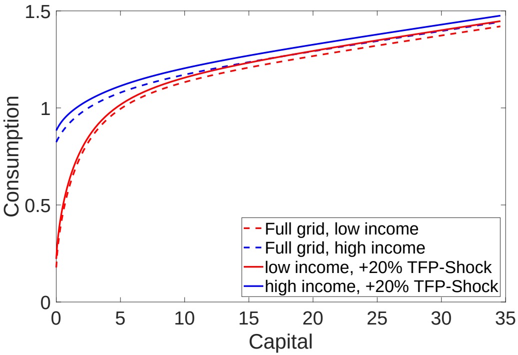


Figure: Perturbing **10** most important basis functions

Different levels of “compression”

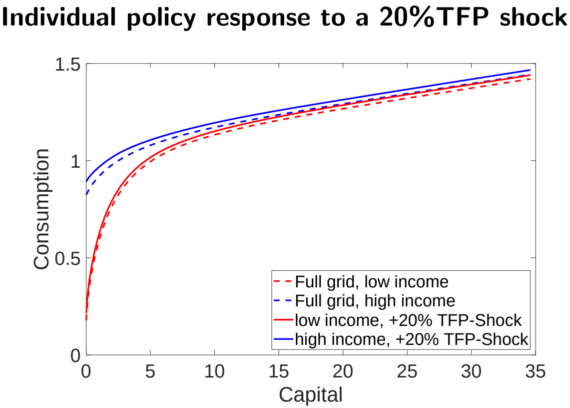


Figure: Perturbing all **200** basis functions

Different levels of “compression”

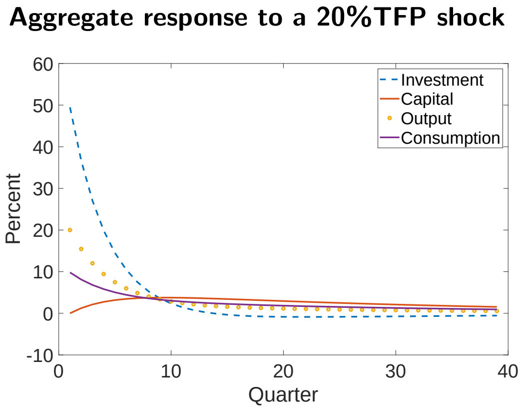


Figure: Perturbing **10** most important basis functions

Different levels of "compression"

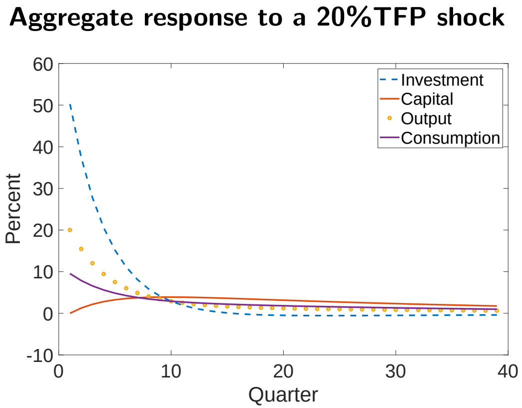
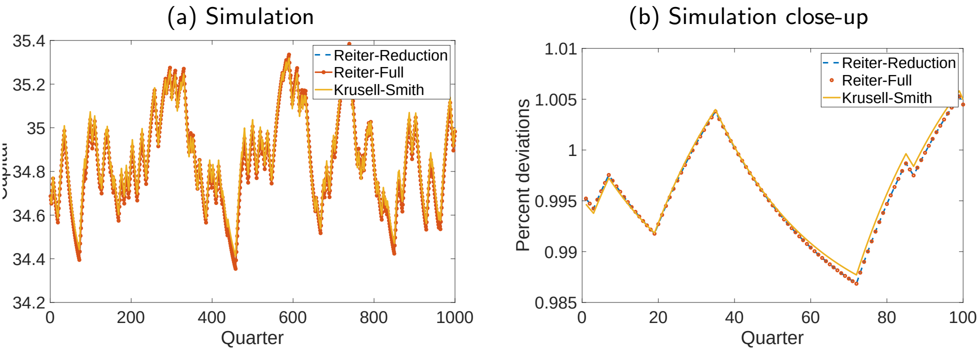


Figure: Perturbing all **200** basis functions

Taking stock

- When looking only at the StE policy function one concludes that roughly 50% of the information is needed to reconstruct the policies well
- This is roughly level of state reduction Reiter (2009) approach would achieve
- Using the StE as reference one can achieve much higher reduction
- For the aggregate dynamics maintaining only 3-6% of the basis functions suffices

Simulation performance



Notes: Both panels show simulations of the Krusell & Smith (1998) model with TFP shocks solved with (1) the Reiter method with our proposed state-space reduction, (2) the original Reiter method without state-space reduction, (3) the original Krusell & Smith algorithm

Error Statistics

Table: Den Haan errors

	Absolute error (in %) for capital K_t		
	Reiter-Reduction	Reiter-Full	K-S
Mean	0.0119	0.0119	0.1237
Max	0.0152	0.0152	0.3491

Notes: Differences in percent between the simulation of the linearized solutions of the model and simulations in which we solve for the intratemporal equilibrium prices in every period and track the full histogram over time for $t = \{1, ..., 1000\}$; see Den Haan (2010)

Computing time

Table: Run time for Krusell & Smith model

	StE	K & S	Reiter-Reduction	Reiter-Full
in seconds	6.28	49.85	0.43	0.91

Notes: Run time in seconds on a Dell laptop with an Intel i7-7500U CPU @ 2.70GHz x 4. Model calibration and number of grid points as in Den Haan et al. (2010). Code in Matlab.

MIT shock solution

- See Kurt Mitman's slides.

Computer Exercise 7

Exercise

Solve the Krusell-Smith model using first order perturbation. For this purpose, first solve the steady state by either EGM or VFI. Then write a function that calculates the Euler equation errors, errors from capital accumulation, and the law of motion for productivity.

Computer Exercise 8

Exercise

Solve the Krusell-Smith model using MIT shock solution approach.

Computer Exercise 9

Exercise

Solve the Krusell-Smith model using first order perturbation and dimensionality reduction proposed by Bayer and Luetticke (2018). For this purpose, split the joint-distribution into Copula and marginals and define the marginals as state. Apply the DCT transformation to the policy function and keep only the most important basis functions as controls. Write the corresponding non-linear difference equations as a function Fsys.

Application: Estimating HANK models

Bayer, Born, Luetticke (2020): Shocks, Frictions, and Inequality in US Business Cycles

What we do

- Fuse two-asset HANK model with a Smets-Wouters-type medium scale DSGE model
- Estimate the model using (Bayesian) full-information approach
- IRF analysis and variance decompositions

- Research Question:
  What shocks and frictions drive the US business cycle and US inequality?

Introducing more macro structure

Linearization techniques easily allow for more structure

- Say, we want to add price stickiness, monetary, and fiscal policy.
- This requires additional extra state variables.
- This is numerical cheap when linearizing.

Recap: Reiter's method(s)

The starting point is the following observation:

- For the household, current prices and a sequence of value functions suffices to describe the decision problem. In discretized form this is

$$v_t = \bar{u}_{h_t} + \beta\Gamma_{h_t} v_{t+1} \tag{22}$$

- h_t is the optimal policy given prices (or other aggregate controls) P_t and continuation values v_t
- This induces payoffs $\bar{u}_{h_t}$ and a transition matrix Γ_{h_t}
- and this transition matrix also induces the law of motion

$$\mu_{t+1} = \mu_t \Gamma_{h_t} \tag{23}$$

- We can view (22) and (23) as the equation describing the idiosyncratic part of a sequential equilibrium with recursive individual planning.

Recap: Compact notation (Schmitt-Grohé and Uribe, 2004)

Allows to write equilibrium as non-linear difference equation

- Add P_t and S_t, purely aggregate controls and states, respectively.
- Define "market-clearing" conditions $\Phi(h_t, \mu_t, P_t, S_t)$
- and a mapping $\Xi(S_t, P_t, \sigma\Sigma\varepsilon_{t+1})$ of controls to $t+1$ states
- Define

$$F\left(\mu_t, S_t, \mu_{t+1}, S_{t+1}, v_t, P_t, v_{t+1}, P_{t+1}, \varepsilon_{t+1}\right) \tag{24}$$

$$= \begin{bmatrix} \mu_{t+1} - \mu_t \Gamma_{h_t} \\ S_{t+1} - \Xi(S_t, P_t, \sigma\Sigma\varepsilon_{t+1}) \\ v_t - \left(\bar{u}_{h_t} + \beta \Gamma_{h_t} v_{t+1}\right) \\ \Phi(h_t, \mu_t, P_t, S_t) \\ \varepsilon_{t+1} \end{bmatrix}$$

s.t.

$$h_t(s) = \arg \max_{x \in \Gamma(s, P_t)} u(s, x) + \beta \mathbb{E} v_{t+1}(s') \tag{25}$$

Estimation

Estimation: Overview

- i.e., method linearizes the resulting non-linear difference equation
- Write as $Ax_t = Bx_{t+1}$ and solve using standard methods
- State-space representation of the model solution

- Use Kalman filter to evaluate likelihood
- Maximize posterior likelihood
- Draw from posterior

Estimation: Solve linear state space model

- First-order perturbation of the non-linear difference equation $EF(x_t, x_{t+1}, \epsilon_t) = 0$ around the stationary equilibrium to obtain a local approximation to the solution

  Need for speed:

1) Again approximate the policy functions as sparse polynomials around their stationary equilibrium values and approximate the distribution functions by histograms of their marginals and a **time-varying Copula**
2) **The resulting system can be reduced even further**
3) **Be smart about parameter updates**

3) Parameter Updates

Changing the aggregate macro structure is easy

- As long as a change in the model does not affect what income is composed of and which choices households can make given prices and incomes, but only how prices are formed, we can change the aggregate part of the model without touching the micro part.

- Modular: Micro and Macro block: $F(\dots) = [F_1, F_2]'$

$$F_1 = \begin{bmatrix} d\bar{\Xi}(\bar{\mu}_t^1, \dots, \bar{\mu}_t^n) - d\bar{\Xi}(\bar{\mu}_t^1, \dots, \bar{\mu}_t^n)\Gamma_t \\ dct\left[idct(\tilde{\Theta}(\theta_t)) - \left(\bar{u}_{h_t} + \beta\Gamma_t idct(\tilde{\Theta}(\theta_{t+1}))\right)\right] \end{bmatrix}$$

$$F_2 = \begin{bmatrix} S_{t+1} - \Xi(S_t, P_t) \\ \Phi(h_t, \mu_t, P_t, S_t) \end{bmatrix}$$

3) Parameter Updates

- We linearize the difference equation describing the model, writing it as

$$\underbrace{\begin{bmatrix} A_{ff} & A_{fX} \\ A_{Xf} & A_{XX} \end{bmatrix}}_{=A} \begin{bmatrix} f_t \\ X_t \end{bmatrix} = -\underbrace{\begin{bmatrix} B_{ff} & B_{fX} \\ B_{Xf} & B_{XX} \end{bmatrix}}_{=B} \begin{bmatrix} f_{t+1} \\ X_{t+1} \end{bmatrix}, \tag{26}$$

where we ordered the "idiosyncratic" equations (Bellman and distributional law of motions) first and all other equations last.

3) Parameter Updates

$$\underbrace{\begin{bmatrix} A_{ff} & A_{fX} \\ A_{Xf} & A_{XX} \end{bmatrix}}_{=A} \begin{bmatrix} f_t \\ X_t \end{bmatrix} = -\underbrace{\begin{bmatrix} B_{ff} & B_{fX} \\ B_{Xf} & B_{XX} \end{bmatrix}}_{=B} \begin{bmatrix} f_{t+1} \\ X_{t+1} \end{bmatrix}, \tag{27}$$

- Only few parameters, a subset of those parameters that affect the stationary equilibrium, affect the first row of blocks, $A_{ff}, B_{ff}, A_{fX}, B_{fX}$, because they directly enter the household problem
- The blocks A_{Xf} and B_{Xf} are parameter free as well because the value functions themselves do not show up in any aggregate equation of the model and the distributions only through summary variables (e.g. their means)
- only A_{XX} and B_{XX} have to be updated after a parameter change during the estimation procedure.

2) Further Model Reduction

- Only few derivatives need updating after a parameter change, still the linearized difference equation, $Ax_t = -Bx_{t+1}$, is still typically a very large system

- In a sequence space solution (Auclert et al., 2019), after some T periods any shock has (or is assumed to have) negligible effects on agents' behavior, on aggregates, and prices:
  - Shock-induced equilibrium changes in the functionals f_t, f_{t+1} have at most as many degrees of freedom as the dimensionality of the sequence of relevant aggregate prices and quantities, $\{Q_s\}_{s=t-T}^{t+T}$
  - The sequence $\{Q_s\}$typically follows some VAR(n) of much smaller order than T

- There must be a factorization of f_t that has at most $J \times T$ factors, but contains the same information as in the sequence space solution.
- See Bayer, Born, Luetticke (2023) appendix for a proof.

2) Further Model Reduction

- Find an orthonormal basis $\mathcal{P} \in \mathbb{R}^{n \times m}$ with $m << n$ such that we can write $f_t = \mathcal{P} Y_t$ and replace the original system by a system with factors Y_t:

$$\underbrace{\begin{bmatrix} \mathcal{P}' A_{ff} \mathcal{P} & \mathcal{P}' A_{fX} \\ A_{Xf} \mathcal{P} & A_{XX} \end{bmatrix}}_{=A'} \begin{bmatrix} Y_t \\ X_t \end{bmatrix} = - \underbrace{\begin{bmatrix} \mathcal{P}' B_{ff} P & \mathcal{P}' B_{fX} \\ B_{Xf} P & B_{XX} \end{bmatrix}}_{=B'} \begin{bmatrix} Y_{t+1} \\ X_{t+1} \end{bmatrix}. \tag{28}$$

- The difficulty here is finding the appropriate basis.
- Ahn et al. (2018) discuss some possible approaches based on an approximation of the IRFs.
- Here we proceed slightly differently, leveraging the strengths of the Bayesian approach.

2) Further Model Reduction

- Calculate for (the DCT representation of) each class of functionals, f, (marginal value of bonds, marginal value of capital, copula) the variance covariance matrix Σ_f.
- This variance-covariance describes how much each DCT-coefficient contributes to the variation of the functional f over time.
- Based on an eigenvalue decomposition of this variance-covariance matrix

$$\Sigma_f = \begin{bmatrix} \mathcal{P}'_{1f} & \mathcal{P}'_{2f} \end{bmatrix} \begin{bmatrix} \lambda_{1f} & 0 \\ 0 & \lambda_{2f} \end{bmatrix} \begin{bmatrix} \mathcal{P}_{1f} \\ \mathcal{P}_{2f} \end{bmatrix} \tag{29}$$

- Determine factors behind the time-series variation of f by splitting the eigenvalues $\lambda_f = \begin{bmatrix} \lambda_{1f} & \lambda_{2f} \end{bmatrix}$ of Σ_f based on being above or below a critical value (typically machine precision). The matrix $\mathcal{P}_{1f}$ that corresponds to the largest eigenvalues, λ_{1f}, then builds a good basis to be used in the model reduction (28). Concretely basis $\mathcal{P}$ is then the block diagonal of each $\mathcal{P}_{1f}$ basis matrix for each respective functional f.

2) Further Model Reduction

- Dependence on shock processes means that it is useful to update the basis used in the model reduction during the estimation process
- This can be done infrequently
- We suggest to generate it once based on the model's priors and then a second time after finding a tentative mode of the parameter distributions (before running the Markov-Chain Monte-Carlo algorithm)

Further Model Reduction - Accuracy

- One can check the precision of our second-step model reduction by comparing likelihood functions and impulse responses to the model solution that only runs the first step model reduction (based on DCTs and the stationary equilibrium)
- An alternative quality check is against a completely different solution technique such as the sequence-space method proposed by Auclert et al. (2019).

1) Time varying Copula

- The copula itself, we write in form of a DCT transformation which allows us, through parameter restrictions, to make sure that any perturbation of the copula is itself a copula.
- Concretely, we write the copula $C_t(\mu_b, \mu_k, \mu_h)$ at time t as the sum of the linear interpolant generated from the steady-state copula $\bar{C}(\mu_b, \mu_k, \mu_h)$ and a perturbation term, which again is a linear interpolant, $c_t(\mu_b, \mu_k, \mu_h)$.

1) Time varying Copula

We allow the two components to have different nodal grids:

- The nodal values of c_t are generated by a DCT with N points in each dimension, $\sum_{l,m,n=1}^{N} \gamma^t_{l,m,n} T_l(\mu_b) T_m(\mu_k) T_n(\mu_h)$, where T are Chebycheff polynomials and where we constrain coefficients $\gamma^t_{l,m,n}$ for which $l+m=2$ or $l+n=2$ or $m+n=2$ to zero.
- This restriction ensures that $\int dc_t = 0$.
- The nodes themselves are chosen to represent an equal fraction of the respective aggregate.
- This way the number of grid points used in the perturbation is delinked from the number of grid-points used in calculating the steady state.

Solving the linear state space model

- Write as $Ax_t = Bx_{t+1}$ and solve using Klein's method to obtain G and H.
- Uses generalized Schur decomposition (computational efficiency)
- Algorithm cost $\mathcal{O}(n^3)$ floating point operations

- We also experimented with the Anderson and Moore (1985) algorithm. While it is more than twice as fast as Klein's method for the HANK model with two assets in many cases, it appears to produce less numerically stable results in a setting such as ours, where the Jacobians are not very sparse.

Solving the linear state space model

Alternative: Speed up Linear Time Iteration (Papp&Reiter, 2020)

- $F_{n+1} = -(B + CF_n)^{-1}A$
- F need not be initialized to zero if an estimate of F is available from an earlier calculation with similar parameter values.
- The linear equation system can be solved making use of a variant of the Sherman-Morrison-Woodbury formula (blockwise matrix inversion).
- Computational complexity only depends on number of states (and not controls)!

Estimation: Kalman filter

Similar to An and Schorfheide (2007) and Fernández-Villaverde (2010)

1. Kalman filter to obtain the likelihood from the state-space representation of the model solution.
2. Advantage of State-Space: Deal with mixed frequency and missing observations.
3. Roughly one evaluation of the Kalman filter every other second.
4. Maximize posterior likelihood

Estimation: Kalman filter

- For a one-frequency data set without missing values, one can speed up the estimation by employing so-called "Chandrasekhar recursions" for evaluating the likelihood.
- These recursions replace the costly updating of the state variance matrix by multiplications involving matrices of much lower dimension (see Herbst, 2014, for details).
- This is especially relevant for the two-asset HANK model as the speed of evaluating the likelihood is dominated by the updating of the state variance matrix, which involves the multiplication of matrices that are quadratic in the number of states.

Estimation: Posterior

- Random Walk Metropolis Hastings algorithm to draw from posterior
- Standard to draw 200k times to recover posterior distribution

Speed up:

- Run multiple chains
- Go sequential Monte Carlo (NY Fed has implemented this for our perturbation approach)

Estimation: Numerical details

For each new draw of the parameter vector (ca 300ms):

1. Update of Jacobian of $F[.]$
2. Solve linear state space model
   - State&Control vector has roughly 150 entries
   - Klein's method via schur decomposition
3. Run Kalman Filter to obtain log-likelihood

Gaussian State Space

- Solution to linearized model takes state space form, which can be written as

$$x_{t+1} = Gx_t + w_{t+1},\ w_{t+1} \overset{iid}{\sim} \mathcal{N}(0, Q) \tag{30}$$

$$y_t = Hx_t + v_t,\ v_t \overset{iid}{\sim} \mathcal{N}(0, R) \tag{31}$$

- G, H, Q, R are functions of the model parameters θ
- x_t is an $n_x \times 1$ vector of states
- y_t is an $n_y \times 1$ vector of observables
- w_t is a $p \times 1$ vector of structural errors
- v_t a vector of measurement errors
- Assumption: w_t and v_t are orthogonal

$$E_t(w_{t+1} v_s) = 0 \quad \forall\ t+1 \text{ and } s \geq 0$$

Fundamental Problem: Unobserved States

- This implies that

$$y_t = H(Gx_{t-1} + w_t) + v_t \tag{32}$$

- Thus, y_t is normally distributed:

$$y_t \sim \mathcal{N}(HGx_{t-1}, HQH + R) \tag{33}$$

- If all states were observed, we could directly construct the likelihood $f(y_T, \ldots, y_1|\theta)$
- We could then run optimizer over our estimated parameter set $\tilde{\theta} \subseteq \theta$ to get ML estimate of $\tilde{\theta}$
- Problem: we have unobserved states and cannot use equation (33)
- Solution: turn to Kalman filter to back out states from the observed data $\rightarrow$ Filtering problem

Kalman Filter: Summary

At time t, given $\hat{x}_{t|t-1}, \Sigma_{t|t-1}$ and observing y_t

1. Compute the forecast error in the observations using

$$a_t = y_t - H\hat{x}_{t|t-1} \tag{34}$$

2. Compute the Kalman Gain K_t using

$$K_t = G\Sigma_{t|t-1}H'\left(H\Sigma_{t|t-1}H' + R\right)^{-1} \tag{35}$$

3. Compute the state forecast for next period given today's information

$$\hat{x}_{t+1|t} = G\hat{x}_{t|t-1} + K_t\left(y_t - H\hat{x}_{t|t-1}\right) = G\hat{x}_{t|t-1} + K_t a_t \tag{36}$$

4. Update the covariance matrix

$$\Sigma_{t+1|t} = (G - K_t H)\,\Sigma_{t|t-1}\,(G - K_t H)' + Q + K_t R K_t' \tag{37}$$

Kalman Filter: Initialization

- How to initialize filter at $t = 0$ where no observations are available?

  $\rightarrow$ start with unconditional mean $E(x)$ and Variance Σ

- Given covariance stationarity, the unconditional mean is

$$E(x) = Ex_{t+1} = E(Gx_t + w_{t+1}) = GE(x) \Rightarrow (I - G)E(x) = 0$$

  hence, $E(x) = 0$

- For the covariance matrix, we have

$$\begin{aligned} \Sigma &= E\left[(Gx_t + w_t)(Gx_t + w_t)'\right] \\ &= E\left[Gx_t x_t{}' G' + w_t w_t{}'\right] \\ &= G\Sigma G' + Q \end{aligned} \tag{38}$$

  $\rightarrow$ so-called Lyapunov-equation

Metropolis Hastings-Algorithm

- Start with a vector θ_0
- Repeat for $j = 1, \ldots, N$
  - Generate $\tilde{\theta}$ from $q(\theta_{j-1}, \cdot)$ and u from $\mathcal{U}(0,1)$
  - If $\tilde{\theta}$ is valid parameter draw (steady state exists, Blanchard-Kahn conditions satisfied etc.) and $u < \alpha(\theta^{j-1}, \theta^j)$ set $\theta_j = \tilde{\theta}$
  - Otherwise, set $\theta_j = \theta_{j-1}$ (implies setting $\pi(\tilde{\theta}) = 0$ if draw invalid)
- Return the values $\{\theta_0, \ldots, \theta_N\}$
- After the chain has passed the transient stage and the effect of the starting values has subsided, the subsequent draws can be considered draws from the posterior

$\Rightarrow$ burnin required that assures remaining chain has converged

The Random-Walk Metropolis Hastings Algorithm

- As long as the regularity conditions are satisfied, any proposal density will ultimately lead to convergence to the invariant distribution
- However: speed of convergence may differ significantly
- In practice, people often use the Random-Walk Metropolis Hastings algorithm where

$$q\left(\theta,\tilde{\theta}\right) = q_{RW}\left(\tilde{\theta}-\theta\right) \tag{39}$$

  and q_{RW} is a multivariate density
- The candidate $\tilde{\theta}$ is thus given by the old value θ plus a random variable increment

$$\tilde{\theta} = \theta + z, z \sim q_{RW} \tag{40}$$

Estimation: Two-step procedure

- First, we calibrate or fix all parameters that affect the steady state of the model.
- Second, we estimate by full-information methods all parameters that only matter for the dynamics of the model, i.e., the aggregate shocks, frictions, and policy rules.
- We set the priors for shocks, frictions, and policy rules to standard values from the representative agent literature

Fusing incomplete markets and price stickiness

A HANK model

- Introduce price stickiness (NKPC),
- introduce a central bank (Taylor rule),
- introduce a fiscal authority (Expenditure rule),
- introduce capital goods producers (quadratic adjustment costs).

Issues:

- We need to deal with assigning monopoly rents.
- We need to deal with the portfolio problem (Bonds vs. Capital).
- We need to make labor supply endogenous

Overview of the model

Workers		Production Sector	Government
Trade Assets	**Obtain Income**	**Produce and Differentiate Consumption Goods**	**Monetary Authority, Fiscal Authority**
Bonds, b>B; and Illiquid Assets (trading friction)	Wages set by unions s.t. Rotemberg wage adjustment costs (Idiosyncratic Income Risk) Interest Dividends Profits	**Intermediate goods producers** Rent capital & labor Competitive Market for Intermediate Goods **Entrepreneurs** Monopolistic resellers s.t. Rotemberg price adjustment costs	Policy Rules: • Monetary authority sets nominal interest rate -> Taylor rule • Fiscal authority supplies government debt, consumes goods, taxes labor income and profits -> Spending rule

A HANK model

See Bayer et al., 2019

Extending the household sector

1. Assume GHH preferences (for business cycles reasonable)

$$u(c,n) = \frac{\left(c - h\frac{n^{1+\gamma}}{1+\gamma}\right)^{1-\xi}}{1-\xi}$$

   Scaling with productivity h allows for easy aggregation w.l.o.g. if taxes are linear.
2. Assign profits to either to (a) a group of households, (b) the government, or (c) a profit-asset.

A HANK model

Modeling portfolio choice: easy version

- All households hold the same bonds-to-capital ratio.
- All assets can be traded without any friction.
- Choice is over total wealth.
- For first order approximation: Returns must equal in expectations, i.e. define a safe return on bonds R_t, prices of capital goods q_t and rental rates of capital r_t, then

$$\mathbb{E}_t \frac{r_{t+1} + q_{t+1}}{q_t} = R_{t+1}$$

Equilibrium conditions (idiosyncratic part)

This leaves us with the following equilibrium conditions:
(A) idiosyncratic part, using linear interpolations in micro problem:

1. Recursive planning. For the vectors of marginal utilities $\mathbf{u}_{c,t}$:

$$\mathbf{u}_{c,t} = \underbrace{\beta R_{t+1} \Gamma_t (\mathbf{u}_{c,t+1} + \lambda_{t+1})}_{\text{one EGM backwards step}}$$

   with Γ_t induced by optimal policies,
   - given future marginal utils $\mathbf{u}_{c,t+1}$, and expected returns R_{t+1}
   - and current incomes determined through wages w_t, dividends r_t, profits π_t, and capital prices q_t.
2. Law of motion for distribution of capital

$$\mu_{t+1} = \mu_t \Gamma_t$$

Equilibrium conditions (summary variables)

(B) summary variables, model free:

1. It is useful to introduce an aggregate control that summarize μ_t: $K_t := \sum_j k_j \mu_t^j$ where k_j is the capital grid.
2. Let $\phi_t := \frac{B_t}{K_t}$ be the bonds-to-capital ratio entering period t.
3. For any unit of capital households hold, they have $r_t + q_t + \phi_t R_t$ resources for consumption.
4. Every unit of capital for next period sells at $q_t + \phi_{t+1}$

Equilibrium conditions (macro model)

(C) prices:

1. Factor prices as controls from FOCs of firms

$$w_t = (1-\alpha) mc_t Z_t \left(\frac{K_t}{N_t}\right)^{\alpha}, \quad r_t = \alpha mc_t Z_t \left(\frac{N_t}{K_t}\right)^{1-\alpha} - \delta,$$

prices of undifferentiated goods, mc_t, and total profits accordingly

$$\hat{\pi}_t = \beta \mathbb{E}_t \hat{\pi}_{t+1} + \kappa \left(mc_t^{-1} - \bar{\mu}\right)$$

$$\Pi_t = (1 - mc_t) Y_t - \text{adjustment costs/profits}$$

2. Returns on government bonds from Taylor rule (state variable)

$$R_{t+1} = R_t^{\rho_R} \hat{\pi}_t^{(1-\rho_R)\theta_\pi} \hat{Y}_t^{(1-\rho_R)\theta_Y}$$

Observe that adjustment costs are zero up to first order around stationary equilibrium.

Equilibrium conditions (macro model)

(D) aggregate quantities:

1. Labor supply
$$(1-\tau)w_t = N_t^{\gamma}$$
2. Production of capital goods (ignore externality)
$$q_t = 1 + \phi \frac{K_{t+1} - K_t}{K_t}$$
3. Total output and components
$$Y_t = Z_t K_t^{\alpha} N_t^{1-\alpha}, \quad C_t = Y_t - G_t - K_{t+1} + (1-\delta)K_t$$
4. A fiscal rule (spending adjusts, B_t is a state, G_t a control)
$$\hat{G}_t = (\hat{B}_t \hat{R}_t / \hat{\pi}_t)^{\rho_B} \hat{\pi}_t^{\gamma_\pi} \hat{Y}_t^{\gamma_Y}, \quad B_{t+1} = G_t + B_t R_t / \pi_t - \tau w_t N_t$$
5. Goods market clearing is residual.

2 asset models

HANK models have more action with more assets

- The literature has highlighted the role of wealthy hand-to-mouth consumers (Kaplan et al., 2014).
- HANK models with more assets feature asset substitution as in the older Keynesian literature (see e.g. Tobin, 1969), which is supported by the data (see Bayer et al., 2019; Luetticke, 2020).

A tractable structure

- For many applications it suffices to assume that capital can only be traded from time to time randomly (Calvo shock).

2 asset models

2 marginal values of assets

- Marginal value of liquid assets results from usual envelop condition

$$V_b(h,b,k) = Ru_c$$

- Marginal value of illiquid assets results from usual envelop condition

$$V_k(h,b,k|adjust) = (q+r)u_c^a$$

when trade is possible and from the marginal value of the dividend payment plus discounted marginal value if no trade is possible

$$V_k(h,b,k|not) = ru_c^n + \beta \mathbb{E} V_k'(h',b',k)$$

- Thus, $V_k(h,b,k) = \lambda(q+r)u_c^a + (1-\lambda)(ru_c^n + \beta \mathbb{E} V_k'(h',b',k))$
- Optimal asset choices require $q\mathbb{E}V_b(h',b',k') = \mathbb{E}V_k(h',b',k')$ which allows us to trace out potentially optimal $(b',k')(h)$ pairs

Computational aspects of HANK-2

- requires both assets (100 points each)
- 22 income states
- uses both value functions and consumption policies as controls
- Full set would have $> 600,000$ variables
- Reduction to 100 distribution states and ca 300 controls for value functions and policies

Exercise 10: Krusell-Smith-model with nominal rigidity

Exercise

Take the setup from last exercise: *and add a government that runs a central bank, a fiscal authority and owns all profit incomes. Households have GHH preferences over labor and consumption, but still unemployment shocks.*

1. *Solve for the steady state without aggregate risk.*
2. *Solve using Bayer and Luetticke's refinement.*

Assume the central bank only reacts to inflation and past interest rates $\rho_R = 0.95$ *and* $\theta_\pi = 1.25$. *The fiscal side only reacts to the level of debt* $\rho_B = -0.1$. *Assume steady state profits are* 10% *and the Phillips Curve reflects price adjustment of roughly once a year if it was from Calvo. Assume steady state labor taxes are 25%.*

Sources of Fluctuations

Standard in complete markets model

- total factor productivity
- gov. bond spread (a.k.a. "risk premium")
- price markup
- wage markup
- monetary policy
- government spending

Sources of Fluctuations

Standard in complete markets model

- total factor productivity
- gov. bond spread (a.k.a. "risk premium")
- price markup
- wage markup
- monetary policy
- government spending

New in the incomplete markets model

- idiosyncratic income risk
- tax progressivity

Observables

Quarterly US data from 1954Q1 – 2019Q4

In first-differences

- GDP, Consumption, Investment
- the real wage

In log-levels

- GDP deflator based inflation rates
- Hours worked per capita
- the (shadow) federal funds rate

All demeaned and without measurement error.

Observables

Further data non-quarterly availability

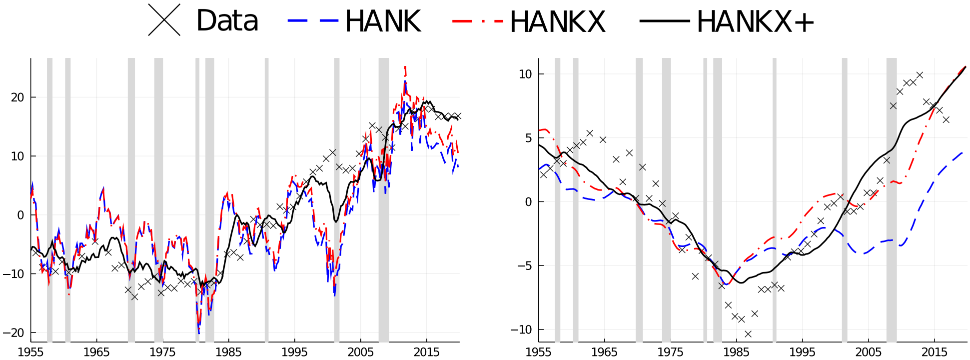
- Measures of inequality:
  - Wealth share of the top 10% (Piketty-Saez WID) (1954 – 2019)
  - Income share of the top 10% (Piketty-Saez WID) (1954 – 2019)

All in log-levels, demeaned and with measurement error.

Estimated model variants

Shocks \ Data	Aggregate Data	+ Cross-sectional Data
Aggregate Shocks	HANK (vs RANK)	HANKX
+ Cross-sectional Shocks		HANKX+

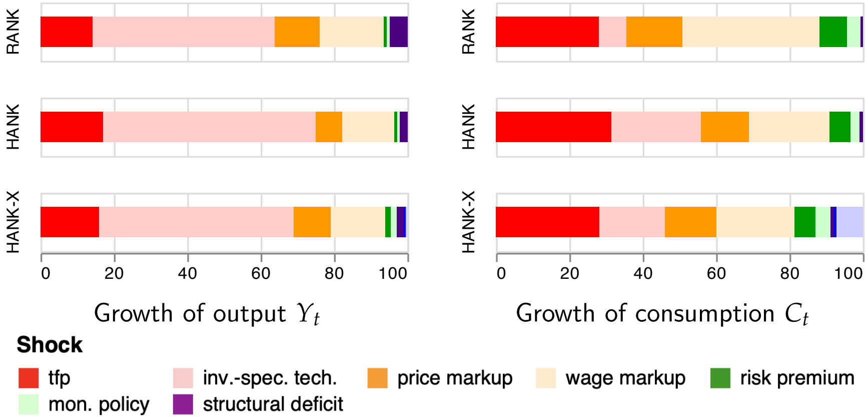
Wealth Inequality in the US



(a) Top 10% income share (b) Top 10% wealth share

Shocks and Frictions in US Business Cycles

Variance decomposition: GDP and components



US business cycles: Summary

HANK and RANK models give only a somewhat different view

Estimation results

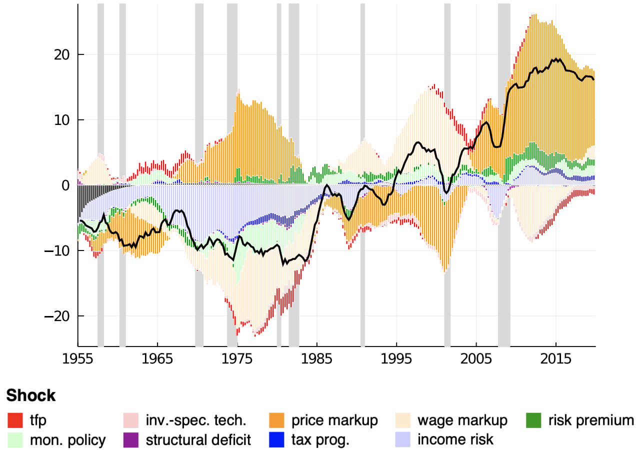
- Key is the estimation that makes the dynamics of both models more similar
- Estimated HANK model features less nominal and real frictions than RANK

Decomposition results

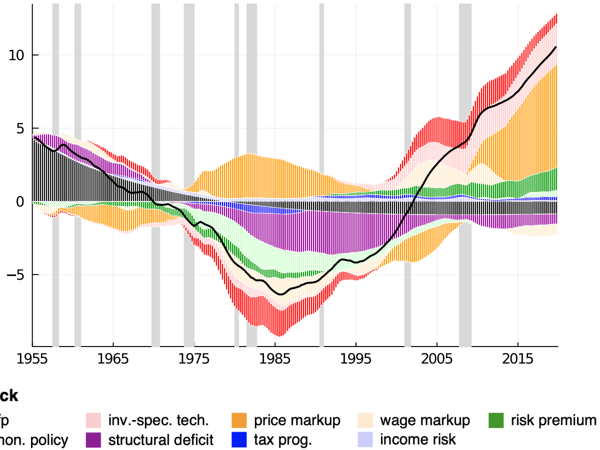
- Investment-specific technology becomes less important because it induces wealth effects on consumption via asset prices
- Risk premium, monetary, and wage markup shocks become more important
- Income risk shocks can partly replace risk premium shocks

Shocks and Frictions in US Inequality

Shock decomposition: Income share of top 10%



Shock decomposition: Wealth share of top 10%



Contribution of shocks to US inequality 1985-2019

Shock	Top 10% Income	Top 10% Wealth
TFP, ϵ^Z	-0.38	2.63
Inv.-spec. tech., ϵ^Ψ	-0.17	3.26
Price markup, $\epsilon^{\mu Y}$	11.69	4.3
Wage markup, $\epsilon^{\mu W}$	5.82	0.87
Risk premium, ϵ^A	-0.62	2.07
Income risk, ϵ^σ	2.57	-0.14
Monetary policy, ϵ^R	1.30	1.98
Structural deficit, ϵ^G	-0.05	1.60
Tax progressivity, τ^P	1.54	0.67
Sum of shocks	21.55	16.79

US inequality: Summary

Business cycle shocks are important drivers of inequality dynamics

Income inequality

- Price and wage markups explain two-third of the increase since 1985
- Rising income risk and falling tax progressivity explain the remaining one-third

Wealth inequality

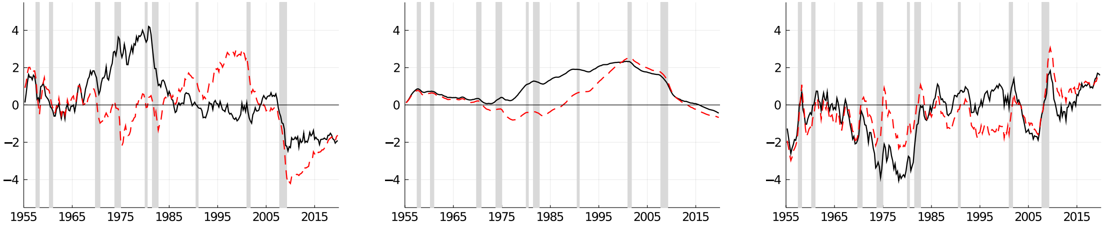
- Technology shocks via their effect on asset prices explain most of the increase since 1985
- The two markup shocks explain only one-third of this increase
- Monetary policy and fiscal deficit shocks are important as well

Policy Counterfactuals

Policy counterfactual: Inequality

- How important are the estimated policy coefficients for the evolution of inequality?
- Run estimated shock sequence with counterfactually set policy parameters
  - Hawkish monetary policy (double inflation response, θ_π)
  - Dovish monetary policy (double output response, θ_Y)

Counterfactual evolution of inequality: Monetary policy



Top 10% income share   Top 10% wealth share   Output

Log point deviations from baseline. Black: Hawkish; Red: Dovish

Policy counterfactual: Summary

Effect of monetary policy depends on supply vs demand shocks

Hawkish monetary policy (triple θ_π)

- Higher inequality in the 70s as markup (cost-push) shocks are important

Dovish monetary policy (triple θ_Y)

- Lower inequality in the 70s and aftermath of the Great Recession

Very persistent effect on wealth inequality.

Summary: Bayer et al. (2020)

Our HANK model can jointly explain the US business cycle and inequality

US business cycle

- Not a radically different view on the US business cycle
- HANK models stress the importance of portfolio choice for the transmission of aggregate shocks

US inequality

- Business cycles are important to understand the evolution of US inequality.
- Business cycle shocks and policy responses can account for most of the increase in US inequality since the 1980s.

Conclusion

No excuse!

- Even when heterogeneity is high dimensional,
- our algorithm is an easy approach to these models
- It is a fast and simple to code

Conclusion

No excuse!

- It requires knowledge of only two standard tools of macro:
  1. Solving a recursive het. agent model for a StE
  2. Linearizing a rep. agent model
  3. (and a little twist in between)
- The fixed design for dimensionality reduction allows to employ the method to estimate models with standard techniques

Bibliography I

- Ahmed, N., T. Natarajan, and K. R. Rao (1974). "Discrete cosine transform". *IEEE Transactions on Computers C-23* (1), 90–93.
- Ahn, SeHyoun, Greg Kaplan, Benjamin Moll, Thomas Winberry, and Christian Wolf (2018). "When inequality matters for macro and macro matters for inequality". *NBER Macroeconomics Annual 32* (1), 1–75.
- An, Sungbae and Frank Schorfheide (2007). "Bayesian analysis of DSGE models". *Econometric Reviews 26* (2-4), 113–172.
- Anderson, Gary and George Moore (1985). "A linear algebraic procedure for solving linear perfect foresight models". *Economics Letters 17* (3), 247–252.
- Auclert, Adrien, Bence Bardóczy, Matthew Rognlie, and Ludwig Straub (2019). "Using the sequence-space Jacobian to solve and estimate heterogeneous-agent models". NBER Working Paper 26123.
- Bayer, Christian, Benjamin Born, and Ralph Luetticke (2020). "Shocks, Frictions, and Inequality in US Business Cycles". CEPR Discussion Paper 14364.

Bibliography II

Bayer, Christian and Ralph Luetticke (2020). "Solving heterogeneous agent models in discrete time with many idiosyncratic states by perturbation methods". *Quantitative Economics forthcoming*.

Bayer, Christian, Ralph Luetticke, Lien Pham-Dao, and Volker Tjaden (2019). "Precautionary savings, illiquid assets, and the aggregate consequences of shocks to household income risk". *Econometrica 87* (1), 255–290.

Boppart, Timo, Per Krusell, and Kurt Mitman (2018). "Exploiting mit shocks in heterogeneous-agent economies: the impulse response as a numerical derivative". *Journal of Economic Dynamics and Control 89*, 68–92.

Carroll, C.D. (2006). "The method of endogenous gridpoints for solving dynamic stochastic optimization problems". *Economics Letters 91* (3), 312–320.

Den Haan, Wouter J (2010). "Assessing the accuracy of the aggregate law of motion in models with heterogeneous agents". *Journal of Economic Dynamics and Control 34* (1), 79–99.

Bibliography III

- Den Haan, Wouter J, Kenneth L Judd, and Michel Juillard (2010). "Computational suite of models with heterogeneous agents: incomplete markets and aggregate uncertainty". *Journal of Economic Dynamics and Control 34* (1), 1–3.
- Fernández-Villaverde, Jesús (2010). "The econometrics of DSGE models". *SERIEs 1* (1), 3–49.
- Herbst, Edward (2014). "Using the "Chandrasekhar Recursions" for likelihood evaluation of DSGE models". *Computational Economics 45* (4), 693–705.
- Hintermaier, T. and W. Koeniger (2010). "The method of endogenous gridpoints with occasionally binding constraints among endogenous variables". *Journal of Economic Dynamics and Control 34* (10), 2074–2088.
- Kaplan, Greg, Giovanni L Violante, and Justin Weidner (2014). "The wealthy hand-to-mouth". *NBER Working Paper No. 20073.*
- Krusell, Per and Anthony A. Smith (1998). "Income and wealth heterogeneity in the macroeconomy". *Journal of Political Economy 106* (5), 867–896.

Bibliography IV

Krusell, Per and Anthony A. Smith (1997). "Income and wealth heterogeneity, portfolio choice, and equilibrium asset returns". *Macroeconomic Dynamics 1* (02), 387–422.

Luetticke, Ralph (2020). "Transmission of monetary policy with heterogeneity in household portfolios". *American Economic Journal: Macroeconomics forthcoming*.

Reiter, Michael (2002). "Recursive computation of heterogeneous agent models". *mimeo, Universitat Pompeu Fabra*.

——— (2009). "Solving heterogeneous-agent models by projection and perturbation". *Journal of Economic Dynamics and Control 33* (3), 649–665.

Schmitt-Grohé, Stephanie and Martın Uribe (2004). "Solving dynamic general equilibrium models using a second-order approximation to the policy function". *Journal of Economic Dynamics and Control 28* (4), 755–775.

Tobin, James (1969). "A general equilibrium approach to monetary theory". *Journal of Money, Credit and Banking 1* (1), 15–29.

Bibliography V

Winberry, Thomas (2016). "A toolbox for solving and estimating heterogeneous agent macro models". *mimeo, Chicago Booth.*