Part IV

Dynamic Linear Economies

CHAPTER

EIGHTEEN

RECURSIVE MODELS OF DYNAMIC LINEAR ECONOMIES

Contents

"Mathematics is the art of giving the same name to different things" – Henri Poincare

"Complete market economies are all alike" – Robert E. Lucas, Jr., (1989)

"Every partial equilibrium model can be reinterpreted as a general equilibrium model." – Anonymous

18.1 A Suite of Models

This lecture presents a class of linear-quadratic-Gaussian models of general economic equilibrium designed by Lars Peter Hansen and Thomas J. Sargent [HS13].

The class of models is implemented in a Python class DLE that is part of quantecon.

Subsequent lectures use the DLE class to implement various instances that have appeared in the economics literature

1. *Growth in Dynamic Linear Economies*
2. *Lucas Asset Pricing using DLE*

18.1.1 Overview of the Models

In saying that "complete markets are all alike", Robert E. Lucas, Jr. was noting that all of them have

- a commodity space.
- a space dual to the commodity space in which prices reside.
- endowments of resources.
- peoples' preferences over goods.
- physical technologies for transforming resources into goods.
- random processes that govern shocks to technologies and preferences and associated information flows.
- a single budget constraint per person.
- the existence of a representative consumer even when there are many people in the model.
- a concept of competitive equilibrium.
- theorems connecting competitive equilibrium allocations to allocations that would be chosen by a benevolent social planner.

The models have **no frictions** such as …

- Enforcement difficulties
- Information asymmetries
- Other forms of transactions costs
- Externalities

The models extensively use the powerful ideas of

- Indexing commodities and their prices by time (John R. Hicks).
- Indexing commodities and their prices by chance (Kenneth Arrow).

Much of the imperialism of complete markets models comes from applying these two tricks.

The Hicks trick of indexing commodities by time is the idea that **dynamics are a special case of statics**.

The Arrow trick of indexing commodities by chance is the idea that **analysis of trade under uncertainty is a special case of the analysis of trade under certainty**.

The [HS13] class of models specify the commodity space, preferences, technologies, stochastic shocks and information flows in ways that allow the models to be analyzed completely using only the tools of linear time series models and linear-quadratic optimal control described in the two lectures Linear State Space Models and Linear Quadratic Control.

There are costs and benefits associated with the simplifications and specializations needed to make a particular model fit within the [HS13] class

- the costs are that linear-quadratic structures are sometimes too confining.

- benefits include computational speed, simplicity, and ability to analyze many model features analytically or nearly analytically.

A variety of superficially different models are all instances of the [HS13] class of models

- Lucas asset pricing model
- Lucas-Prescott model of investment under uncertainty
- Asset pricing models with habit persistence
- Rosen-Topel equilibrium model of housing
- Rosen schooling models
- Rosen-Murphy-Scheinkman model of cattle cycles
- Hansen-Sargent-Tallarini model of robustness and asset pricing
- Many more …

The diversity of these models conceals an essential unity that illustrates the quotation by Robert E. Lucas, Jr., with which we began this lecture.

18.1.2 Forecasting?

A consequence of a single budget constraint per person plus the Hicks-Arrow tricks is that households and firms need not forecast.

But there exist equivalent structures called **recursive competitive equilibria** in which they do appear to need to forecast.

In these structures, to forecast, households and firms use:

- equilibrium pricing functions, and
- knowledge of the Markov structure of the economy's state vector.

18.1.3 Theory and Econometrics

For an application of the [HS13] class of models, the outcome of theorizing is a stochastic process, i.e., a probability distribution over sequences of prices and quantities, indexed by parameters describing preferences, technologies, and information flows.

Another name for that object is a likelihood function, a key object of both frequentist and Bayesian statistics.

There are two important uses of an **equilibrium stochastic process** or **likelihood function**.

The first is to solve the **direct problem**.

The **direct problem** takes as inputs values of the parameters that define preferences, technologies, and information flows and as an output characterizes or simulates random paths of quantities and prices.

The second use of an equilibrium stochastic process or likelihood function is to solve the **inverse problem**.

The **inverse problem** takes as an input a time series sample of observations on a subset of prices and quantities determined by the model and from them makes inferences about the parameters that define the model's preferences, technologies, and information flows.

18.1.4 More Details

A [HS13] economy consists of **lists of matrices** that describe peoples' household technologies, their preferences over consumption services, their production technologies, and their information sets.

There are complete markets in history-contingent commodities.

Competitive equilibrium allocations and prices

- satisfy equations that are easy to write down and solve
- have representations that are convenient econometrically

Different example economies manifest themselves simply as different settings for various matrices.

[HS13] use these tools:

- A theory of recursive dynamic competitive economies
- Linear optimal control theory
- Recursive methods for estimating and interpreting vector autoregressions

The models are flexible enough to express alternative senses of a representative household

- A single 'stand-in' household of the type used to good effect by Edward C. Prescott.
- Heterogeneous households satisfying conditions for Gorman aggregation into a representative household.
- Heterogeneous household technologies that violate conditions for Gorman aggregation but are still susceptible to aggregation into a single representative household via 'non-Gorman' or 'mongrel' aggregation'.

These three alternative types of aggregation have different consequences in terms of how prices and allocations can be computed.

In particular, can prices and an aggregate allocation be computed before the equilibrium allocation to individual heterogeneous households is computed?

- Answers are "Yes" for Gorman aggregation, "No" for non-Gorman aggregation.

In summary, the insights and practical benefits from economics to be introduced in this lecture are

- Deeper understandings that come from recognizing common underlying structures.
- Speed and ease of computation that comes from unleashing a common suite of Python programs.

We'll use the following **mathematical tools**

- Stochastic Difference Equations (Linear).
- Duality: LQ Dynamic Programming and Linear Filtering are the same things mathematically.
- The Spectral Factorization Identity (for understanding vector autoregressions and non-Gorman aggregation).

So here is our roadmap.

We'll describe sets of matrices that pin down

- Information
- Technologies
- Preferences

Then we'll describe

- Equilibrium concept and computation
- Econometric representation and estimation

18.1.5 Stochastic Model of Information Flows and Outcomes

We'll use stochastic linear difference equations to describe information flows **and** equilibrium outcomes.

The sequence $\{w_t : t = 1, 2, ...\}$ is said to be a martingale difference sequence adapted to $\{J_t : t = 0, 1, ...\}$ if $E(w_{t+1}|J_t) = 0$ for $t = 0, 1, ...$.

The sequence $\{w_t : t = 1, 2, ...\}$ is said to be conditionally homoskedastic if $E(w_{t+1}w_{t+1}' \mid J_t) = I$ for $t = 0, 1, ...$.

We assume that the $\{w_t : t = 1, 2, ...\}$ process is conditionally homoskedastic.

Let $\{x_t : t = 1, 2, ...\}$ be a sequence of n-dimensional random vectors, i.e. an n-dimensional stochastic process.

The process $\{x_t : t = 1, 2, ...\}$ is constructed recursively using an initial random vector $x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_0)$ and a time-invariant law of motion:

$$x_{t+1} = Ax_t + Cw_{t+1}$$

for $t = 0, 1, ...$ where A is an n by n matrix and C is an n by N matrix.

Evidently, the distribution of x_{t+1} conditional on x_t is $\mathcal{N}(Ax_t, CC')$.

18.1.6 Information Sets

Let J_0 be generated by x_0 and J_t be generated by $x_0, w_1, \ldots, w_t$, which means that J_t consists of the set of all measurable functions of $\{x_0, w_1, \ldots, w_t\}$.

18.1.7 Prediction Theory

The optimal forecast of x_{t+1} given current information is

$$E(x_{t+1} \mid J_t) = Ax_t$$

and the one-step-ahead forecast error is

$$x_{t+1} - E(x_{t+1} \mid J_t) = Cw_{t+1}$$

The covariance matrix of x_{t+1} conditioned on J_t is

$$E(x_{t+1} - E(x_{t+1} \mid J_t))(x_{t+1} - E(x_{t+1} \mid J_t))' = CC'$$

A nonrecursive expression for x_t as a function of $x_0, w_1, w_2, \ldots, w_t$ is

$$\begin{aligned} x_t &= Ax_{t-1} + Cw_t \\ &= A^2 x_{t-2} + ACw_{t-1} + Cw_t \\ &= \left[\sum_{\tau=0}^{t-1} A^\tau C w_{t-\tau}\right] + A^t x_0 \end{aligned}$$

Shift forward in time:

$$x_{t+j} = \sum_{s=0}^{j-1} A^s C w_{t+j-s} + A^j x_t$$

Projecting on the information set $\{x_0, w_t, w_{t-1}, \ldots, w_1\}$ gives

$$E_t x_{t+j} = A^j x_t$$

where $E_t(\cdot) \equiv E[(\cdot) \mid x_0, w_t, w_{t-1}, \ldots, w_1] = E(\cdot) \mid J_t$, and x_t is in J_t.

It is useful to obtain the covariance matrix of the j-step-ahead prediction error $x_{t+j} - E_t x_{t+j} = \sum_{s=0}^{j-1} A^s C w_{t-s+j}$.

Evidently,

$$E_t(x_{t+j} - E_t x_{t+j})(x_{t+j} - E_t x_{t+j})' = \sum_{k=0}^{j-1} A^k C C' A^{k'} \equiv v_j$$

v_j can be calculated recursively via

$$\begin{aligned} v_1 &= CC' \\ v_j &= CC' + A v_{j-1} A', \quad j \geq 2 \end{aligned}$$

18.1.8 Orthogonal Decomposition

To decompose these covariances into parts attributable to the individual components of w_t, we let i_τ be an N-dimensional column vector of zeroes except in position τ, where there is a one. Define a matrix $\upsilon_{j,\tau}$

$$\upsilon_{j,\tau} = \sum_{k=0}^{j-1} A^k C i_\tau i_\tau' C' A'^k.$$

Note that $\sum_{\tau=1}^N i_\tau i_\tau' = I$, so that we have

$$\sum_{\tau=1}^N \upsilon_{j,\tau} = \upsilon_j$$

Evidently, the matrices $\{\upsilon_{j,\tau}, \tau = 1, \ldots, N\}$ give an orthogonal decomposition of the covariance matrix of j-step-ahead prediction errors into the parts attributable to each of the components $\tau = 1, \ldots, N$.

18.1.9 Taste and Technology Shocks

$E(w_t \mid J_{t-1}) = 0$ and $E(w_t w_t' \mid J_{t-1}) = I$ for $t = 1, 2, \ldots$

$$b_t = U_b z_t \text{ and } d_t = U_d z_t,$$

U_b and U_d are matrices that select entries of z_t. The law of motion for $\{z_t : t = 0, 1, \ldots\}$ is

$$z_{t+1} = A_{22} z_t + C_2 w_{t+1} \text{ for } t = 0, 1, \ldots$$

where z_0 is a given initial condition. The eigenvalues of the matrix A_{22} have absolute values that are less than or equal to one.

Thus, in summary, our model of **information and shocks** is

$$\begin{aligned} z_{t+1} &= A_{22} z_t + C_2 w_{t+1} \\ b_t &= U_b z_t \\ d_t &= U_d z_t. \end{aligned}$$

We can now briefly summarize other components of our economies, in particular

- Production technologies
- Household technologies
- Household preferences

18.1.10 Production Technology

Where c_t is a vector of consumption rates, k_t is a vector of physical capital goods, g_t is a vector intermediate productions goods, d_t is a vector of technology shocks, the production technology is

$$\begin{aligned}\Phi_c c_t + \Phi_g g_t + \Phi_i i_t &= \Gamma k_{t-1} + d_t \\ k_t &= \Delta_k k_{t-1} + \Theta_k i_t \\ g_t \cdot g_t &= \ell_t^2\end{aligned}$$

Here $\Phi_c, \Phi_g, \Phi_i, \Gamma, \Delta_k, \Theta_k$ are all matrices conformable to the vectors they multiply and ℓ_t is a disutility generating resource supplied by the household.

For technical reasons that facilitate computations, we make the following.

Assumption: $[\Phi_c \ \Phi_g]$ is nonsingular.

18.1.11 Household Technology

Households confront a technology that allows them to devote consumption goods to construct a vector h_t of household capital goods and a vector s_t of utility generating house services

$$\begin{aligned}s_t &= \Lambda h_{t-1} + \Pi c_t \\ h_t &= \Delta_h h_{t-1} + \Theta_h c_t\end{aligned}$$

where $\Lambda, \Pi, \Delta_h, \Theta_h$ are matrices that pin down the household technology.

We make the following

Assumption: The absolute values of the eigenvalues of Δ_h are less than or equal to one.

Below, we'll outline further assumptions that we shall occasionally impose.

18.1.12 Preferences

Where b_t is a stochastic process of preference shocks that will play the role of demand shifters, the representative household orders stochastic processes of consumption services s_t according to

$$\left(\frac{1}{2}\right) E \sum_{t=0}^{\infty} \beta^t [(s_t - b_t) \cdot (s_t - b_t) + \ell_t^2] \big| J_0, \ 0 < \beta < 1$$

We now proceed to give examples of production and household technologies that appear in various models that appear in the literature.

First, we give examples of production Technologies

$$\Phi_c c_t + \Phi_g g_t + \Phi_i i_t = \Gamma k_{t-1} + d_t$$

$$\mid g_t \mid \leq \ell_t$$

so we'll be looking for specifications of the matrices $\Phi_c, \Phi_g, \Phi_i, \Gamma, \Delta_k, \Theta_k$ that define them.

18.1.13 Endowment Economy

There is a single consumption good that cannot be stored over time.

In time period t, there is an endowment d_t of this single good.

There is neither a capital stock, nor an intermediate good, nor a rate of investment.

So $c_t = d_t$.

To implement this specification, we can choose A_{22}, C_2, and U_d to make d_t follow any of a variety of stochastic processes.

To satisfy our earlier rank assumption, we set:

$$c_t + i_t = d_{1t}$$

$$g_t = \phi_1 i_t$$

where ϕ_1 is a small positive number.

To implement this version, we set $\Delta_k = \Theta_k = 0$ and

$$\Phi_c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \Phi_i = \begin{bmatrix} 1 \\ \phi_1 \end{bmatrix}, \ \Phi_g = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \ \Gamma = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ d_t = \begin{bmatrix} d_{1t} \\ 0 \end{bmatrix}$$

We can use this specification to create a linear-quadratic version of Lucas's (1978) asset pricing model.

18.1.14 Single-Period Adjustment Costs

There is a single consumption good, a single intermediate good, and a single investment good.

The technology is described by

$$\begin{aligned} c_t &= \gamma k_{t-1} + d_{1t}, \ \ \gamma > 0 \\ \phi_1 i_t &= g_t + d_{2t}, \ \ \phi_1 > 0 \\ \ell_t^2 &= g_t^2 \\ k_t &= \delta_k k_{t-1} + i_t, \ 0 < \delta_k < 1 \end{aligned}$$

Set

$$\Phi_c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ \Phi_g = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \ \Phi_i = \begin{bmatrix} 0 \\ \phi_1 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} \gamma \\ 0 \end{bmatrix}, \ \Delta_k = \delta_k, \ \Theta_k = 1$$

We set A_{22}, C_2 and U_d to make $(d_{1t}, d_{2t})' = d_t$ follow a desired stochastic process.

Now we describe some examples of preferences, which as we have seen are ordered by

$$-\left(\frac{1}{2}\right) E \sum_{t=0}^{\infty} \beta^t \left[(s_t - b_t) \cdot (s_t - b_t) + (\ell_t)^2 \right] \mid J_0 \quad , \ 0 < \beta < 1$$

where household services are produced via the household technology

$$h_t = \Delta_h h_{t-1} + \Theta_h c_t$$

$$s_t = \Lambda h_{t-1} + \Pi c_t$$

and we make

Assumption: The absolute values of the eigenvalues of Δ_h are less than or equal to one.

Later we shall introduce **canonical** household technologies that satisfy an 'invertibility' requirement relating sequences $\{s_t\}$ of services and $\{c_t\}$ of consumption flows.

And we'll describe how to obtain a canonical representation of a household technology from one that is not canonical.

Here are some examples of household preferences.

Time Separable preferences

$$-\frac{1}{2}E\sum_{t=0}^{\infty}\beta^t\left[(c_t-b_t)^2+\ell_t^2\right]\mid J_0 \quad , \, 0<\beta<1$$

Consumer Durables

$$h_t=\delta_h h_{t-1}+c_t \quad , \, 0<\delta_h<1$$

Services at t are related to the stock of durables at the beginning of the period:

$$s_t=\lambda h_{t-1} \, , \; \lambda>0$$

Preferences are ordered by

$$-\frac{1}{2}E\sum_{t=0}^{\infty}\beta^t\left[(\lambda h_{t-1}-b_t)^2+\ell_t^2\right]\mid J_0$$

Set $\Delta_h=\delta_h,\Theta_h=1,\Lambda=\lambda,\Pi=0$.

Habit Persistence

$$-\left(\frac{1}{2}\right)E\sum_{t=0}^{\infty}\beta^t\Big[\big(c_t-\lambda(1-\delta_h)\sum_{j=0}^{\infty}\delta_h^j\,c_{t-j-1}-b_t\big)^2+\ell_t^2\Big]\Big|J_0$$

$$0<\beta<1 \, , \; 0<\delta_h<1 \, , \; \lambda>0$$

Here the effective bliss point $b_t+\lambda(1-\delta_h)\sum_{j=0}^{\infty}\delta_h^j\,c_{t-j-1}$ shifts in response to a moving average of past consumption.

Initial Conditions

Preferences of this form require an initial condition for the geometric sum $\sum_{j=0}^{\infty}\delta_h^j c_{t-j-1}$ that we specify as an initial condition for the 'stock of household durables,' h_{-1}.

Set

$$h_t=\delta_h h_{t-1}+(1-\delta_h)c_t \quad , \, 0<\delta_h<1$$

$$h_t=(1-\delta_h)\sum_{j=0}^{t}\delta_h^j\,c_{t-j}+\delta_h^{t+1}\,h_{-1}$$

$$s_t=-\lambda h_{t-1}+c_t, \; \lambda>0$$

To implement, set $\Lambda=-\lambda, \; \Pi=1, \; \Delta_h=\delta_h, \; \Theta_h=1-\delta_h$.

Seasonal Habit Persistence

$$-\left(\frac{1}{2}\right)E\sum_{t=0}^{\infty}\beta^t\Big[\big(c_t-\lambda(1-\delta_h)\sum_{j=0}^{\infty}\delta_h^j\,c_{t-4j-4}-b_t\big)^2+\ell_t^2\Big]$$

$$0<\beta<1 \, , \; 0<\delta_h<1 \, , \; \lambda>0$$

Here the effective bliss point $b_t + \lambda(1-\delta_h)\sum_{j=0}^{\infty} \delta_h^j\, c_{t-4j-4}$ shifts in response to a moving average of past consumptions of the same quarter.

To implement, set

$$\tilde{h}_t = \delta_h \tilde{h}_{t-4} + (1-\delta_h)c_t \quad , \; 0 < \delta_h < 1$$

This implies that

$$h_t = \begin{bmatrix} \tilde{h}_t \\ \tilde{h}_{t-1} \\ \tilde{h}_{t-2} \\ \tilde{h}_{t-3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \delta_h \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{h}_{t-1} \\ \tilde{h}_{t-2} \\ \tilde{h}_{t-3} \\ \tilde{h}_{t-4} \end{bmatrix} + \begin{bmatrix} (1-\delta_h) \\ 0 \\ 0 \\ 0 \end{bmatrix} c_t$$

with consumption services

$$s_t = -\begin{bmatrix} 0 & 0 & 0 & -\lambda \end{bmatrix} h_{t-1} + c_t \quad , \; \lambda > 0$$

Adjustment Costs.

Recall

$$-\Big(\frac{1}{2}\Big) E \sum_{t=0}^{\infty} \beta^t [(c_t - b_{1t})^2 + \lambda^2 (c_t - c_{t-1})^2 + \ell_t^2] \mid J_0$$

$$0 < \beta < 1 \quad , \; \lambda > 0$$

To capture adjustment costs, set

$$h_t = c_t$$

$$s_t = \begin{bmatrix} 0 \\ -\lambda \end{bmatrix} h_{t-1} + \begin{bmatrix} 1 \\ \lambda \end{bmatrix} c_t$$

so that

$$s_{1t} = c_t$$

$$s_{2t} = \lambda (c_t - c_{t-1})$$

We set the first component b_{1t} of b_t to capture the stochastic bliss process and set the second component identically equal to zero.

Thus, we set $\Delta_h = 0, \Theta_h = 1$

$$\Lambda = \begin{bmatrix} 0 \\ -\lambda \end{bmatrix} \; , \; \Pi = \begin{bmatrix} 1 \\ \lambda \end{bmatrix}$$

Multiple Consumption Goods

$$\Lambda = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \text{ and } \; \Pi = \begin{bmatrix} \pi_1 & 0 \\ \pi_2 & \pi_3 \end{bmatrix}$$

$$-\frac{1}{2}\beta^t (\Pi c_t - b_t)' (\Pi c_t - b_t)$$

$$\mu_t = -\beta^t [\Pi' \Pi \, c_t - \Pi' \, b_t]$$

$$c_t = -(\Pi'\Pi)^{-1} \beta^{-t} \mu_t + (\Pi'\Pi)^{-1} \Pi' b_t$$

This is called the **Frisch demand function** for consumption.

We can think of the vector μ_t as playing the role of prices, up to a common factor, for all dates and states.

The scale factor is determined by the choice of numeraire.

Notions of **substitutes and complements** can be defined in terms of these Frisch demand functions.

Two goods can be said to be **substitutes** if the cross-price effect is positive and to be **complements** if this effect is negative.

Hence this classification is determined by the off-diagonal element of $-(\Pi'\Pi)^{-1}$, which is equal to $\pi_2\pi_3/\det(\Pi'\Pi)$.

If π_2 and π_3 have the same sign, the goods are substitutes.

If they have opposite signs, the goods are complements.

To summarize, our economic structure consists of the matrices that define the following components:

Information and shocks

$$\begin{aligned} z_{t+1} &= A_{22} z_t + C_2 w_{t+1} \\ b_t &= U_b z_t \\ d_t &= U_d z_t \end{aligned}$$

Production Technology

$$\begin{aligned} \Phi_c c_t + \Phi_g g_t + \Phi_i i_t &= \Gamma k_{t-1} + d_t \\ k_t &= \Delta_k k_{t-1} + \Theta_k i_t \\ g_t \cdot g_t &= \ell_t^2 \end{aligned}$$

Household Technology

$$\begin{aligned} s_t &= \Lambda h_{t-1} + \Pi c_t \\ h_t &= \Delta_h h_{t-1} + \Theta_h c_t \end{aligned}$$

Preferences

$$\left(\frac{1}{2}\right) E \sum_{t=0}^{\infty} \beta^t [(s_t - b_t)\cdot(s_t - b_t) + \ell_t^2] \big| J_0, \; 0 < \beta < 1$$

Next steps: we move on to discuss two closely connected concepts

- A Planning Problem or Optimal Resource Allocation Problem
- Competitive Equilibrium

18.1.15 Optimal Resource Allocation

Imagine a planner who chooses sequences $\{c_t, i_t, g_t\}_{t=0}^{\infty}$ to maximize

$$-(1/2) E \sum_{t=0}^{\infty} \beta^t [(s_t - b_t)\cdot(s_t - b_t) + g_t \cdot g_t] \big| J_0$$

subject to the constraints

$$\begin{aligned} \Phi_c c_t + \Phi_g g_t + \Phi_i i_t &= \Gamma k_{t-1} + d_t, \\ k_t &= \Delta_k k_{t-1} + \Theta_k i_t, \\ h_t &= \Delta_h h_{t-1} + \Theta_h c_t, \\ s_t &= \Lambda h_{t-1} + \Pi c_t, \\ z_{t+1} &= A_{22} z_t + C_2 w_{t+1}, \; b_t = U_b z_t, \; \text{ and } \; d_t = U_d z_t \end{aligned}$$

and initial conditions for h_{-1}, k_{-1}, and z_0.

Throughout, we shall impose the following **square summability** conditions

$$E\sum_{t=0}^{\infty}\beta^t h_t \cdot h_t \mid J_0 < \infty \text{ and } E\sum_{t=0}^{\infty}\beta^t k_t \cdot k_t \mid J_0 < \infty$$

Define:

$$L_0^2 = [\{y_t\} : y_t \text{ is a random variable in } J_t \text{ and } E\sum_{t=0}^{\infty}\beta^t y_t^2 \mid J_0 < +\infty]$$

Thus, we require that each component of h_t and each component of k_t belong to L_0^2.

We shall compare and utilize two approaches to solving the planning problem

- Lagrangian formulation
- Dynamic programming

18.1.16 Lagrangian Formulation

Form the Lagrangian

$$\begin{aligned}
\mathcal{L} = -E\sum_{t=0}^{\infty}\beta^t\Bigg[&\left(\frac{1}{2}\right)[(s_t-b_t)\cdot(s_t-b_t)+g_t\cdot g_t] \\
&+ M_t^{d\prime}\cdot(\Phi_c c_t+\Phi_g g_t+\Phi_i i_t-\Gamma k_{t-1}-d_t) \\
&+ M_t^{k\prime}\cdot(k_t-\Delta_k k_{t-1}-\Theta_k i_t) \\
&+ M_t^{h\prime}\cdot(h_t-\Delta_h h_{t-1}-\Theta_h c_t) \\
&+ M_t^{s\prime}\cdot(s_t-\Lambda h_{t-1}-\Pi c_t)\Bigg]\Bigg| J_0
\end{aligned}$$

The planner maximizes $\mathcal{L}$ with respect to the quantities $\{c_t, i_t, g_t\}_{t=0}^{\infty}$ and minimizes with respect to the Lagrange multipliers $M_t^d, M_t^k, M_t^h, M_t^s$.

First-order necessary conditions for maximization with respect to c_t, g_t, h_t, i_t, k_t, and s_t, respectively, are:

$$\begin{aligned}
-\Phi_c' M_t^d + \Theta_h' M_t^h + \Pi' M_t^s &= 0, \\
-g_t - \Phi_g' M_t^d &= 0, \\
-M_t^h + \beta E(\Delta_h' M_{t+1}^h + \Lambda' M_{t+1}^s) \mid J_t &= 0, \\
-\Phi_i' M_t^d + \Theta_k' M_t^k &= 0, \\
-M_t^k + \beta E(\Delta_k' M_{t+1}^k + \Gamma' M_{t+1}^d) \mid J_t &= 0, \\
-s_t + b_t - M_t^s &= 0
\end{aligned}$$

for $t = 0, 1, \ldots$.

In addition, we have the complementary slackness conditions (these recover the original transition equations) and also transversality conditions

$$\begin{aligned}
\lim_{t\to\infty}\beta^t E[M_t^{k\prime}k_t] \mid J_0 = 0 \\
\lim_{t\to\infty}\beta^t E[M_t^{h\prime}h_t] \mid J_0 = 0
\end{aligned}$$

The system formed by the FONCs and the transition equations can be handed over to Python.

Python will solve the planning problem for fixed parameter values.

Here are the **Python Ready Equations**

$$
\begin{aligned}
-\Phi_c' M_t^d + \Theta_h' M_t^h + \Pi' M_t^s &= 0, \\
-g_t - \Phi_g' M_t^d &= 0, \\
-M_t^h + \beta E(\Delta_h' M_{t+1}^h + \Lambda' M_{t+1}^s) \mid J_t &= 0, \\
-\Phi_i' M_t^d + \Theta_k' M_t^k &= 0, \\
-M_t^k + \beta E(\Delta_k' M_{t+1}^k + \Gamma' M_{t+1}^d) \mid J_t &= 0, \\
-s_t + b_t - M_t^s &= 0 \\
\Phi_c c_t + \Phi_g g_t + \Phi_i i_t &= \Gamma k_{t-1} + d_t, \\
k_t &= \Delta_k k_{t-1} + \Theta_k i_t, \\
h_t &= \Delta_h h_{t-1} + \Theta_h c_t, \\
s_t &= \Lambda h_{t-1} + \Pi c_t, \\
z_{t+1} &= A_{22} z_t + C_2 w_{t+1}, \; b_t = U_b z_t, \; \text{ and } \; d_t = U_d z_t
\end{aligned}
$$

The Lagrange multipliers or **shadow prices** satisfy

$$
\begin{aligned}
M_t^s &= b_t - s_t \\
M_t^h &= E\left[\sum_{\tau=1}^{\infty} \beta^\tau (\Delta_h')^{\tau-1} \Lambda' M_{t+\tau}^s \mid J_t\right] \\
M_t^d &= \begin{bmatrix} \Phi_c' \\ \Phi_g' \end{bmatrix}^{-1} \begin{bmatrix} \Theta_h' M_t^h + \Pi' M_t^s \\ -g_t \end{bmatrix} \\
M_t^k &= E\left[\sum_{\tau=1}^{\infty} \beta^\tau (\Delta_k')^{\tau-1} \Gamma' M_{t+\tau}^d \mid J_t\right] \\
M_t^i &= \Theta_k' M_t^k
\end{aligned}
$$

Although it is possible to use matrix operator methods to solve the above **Python ready equations**, that is not the approach we'll use.

Instead, we'll use dynamic programming to get recursive representations for both quantities and shadow prices.

18.1.17 Dynamic Programming

Dynamic Programming always starts with the word **let**.

Thus, let $V(x_0)$ be the optimal value function for the planning problem as a function of the initial state vector x_0.

(Thus, in essence, dynamic programming amounts to an application of a **guess and verify** method in which we begin with a guess about the answer to the problem we want to solve. That's why we start with **let** $V(x_0)$ be the (value of the) answer to the problem, then establish and verify a bunch of conditions $V(x_0)$ has to satisfy if indeed it is the answer)

The optimal value function $V(x)$ satisfies the **Bellman equation**

$$
V(x_0) = \max_{c_0, i_0, g_0} [-.5[(s_0 - b_0) \cdot (s_0 - b_0) + g_0 \cdot g_0] + \beta E V(x_1)]
$$

subject to the linear constraints

$$
\begin{aligned}
\Phi_c c_0 + \Phi_g g_0 + \Phi_i i_0 &= \Gamma k_{-1} + d_0, \\
k_0 &= \Delta_k k_{-1} + \Theta_k i_0, \\
h_0 &= \Delta_h h_{-1} + \Theta_h c_0, \\
s_0 &= \Lambda h_{-1} + \Pi c_0, \\
z_1 &= A_{22} z_0 + C_2 w_1, \; b_0 = U_b z_0 \; \text{ and } \; d_0 = U_d z_0
\end{aligned}
$$

Because this is a linear-quadratic dynamic programming problem, it turns out that the value function has the form

$$V(x) = x'Px + \rho$$

Thus, we want to solve an instance of the following linear-quadratic dynamic programming problem:

Choose a contingency plan for $\{x_{t+1}, u_t\}_{t=0}^{\infty}$ to maximize

$$-E\sum_{t=0}^{\infty}\beta^t[x_t'Rx_t + u_t'Qu_t + 2u_t'W'x_t],\ 0 < \beta < 1$$

subject to

$$x_{t+1} = Ax_t + Bu_t + Cw_{t+1},\ t \geq 0$$

where x_0 is given; x_t is an $n \times 1$ vector of state variables, and u_t is a $k \times 1$ vector of control variables.

We assume w_{t+1} is a martingale difference sequence with $Ew_tw_t' = I$, and that C is a matrix conformable to x and w.

The optimal value function $V(x)$ satisfies the Bellman equation

$$V(x_t) = \max_{u_t}\Big\{-(x_t'Rx_t + u_t'Qu_t + 2u_t'Wx_t) + \beta E_tV(x_{t+1})\Big\}$$

where maximization is subject to

$$x_{t+1} = Ax_t + Bu_t + Cw_{t+1},\ t \geq 0$$

$$V(x_t) = -x_t'Px_t - \rho$$

P satisfies

$$P = R + \beta A'PA - (\beta A'PB + W)(Q + \beta B'PB)^{-1}(\beta B'PA + W')$$

This equation in P is called the **algebraic matrix Riccati equation**.

The optimal decision rule is $u_t = -Fx_t$, where

$$F = (Q + \beta B'PB)^{-1}(\beta B'PA + W')$$

The optimum decision rule for u_t is independent of the parameters C, and so of the noise statistics.

Iterating on the Bellman operator leads to

$$V_{j+1}(x_t) = \max_{u_t}\Big\{-(x_t'Rx_t + u_t'Qu_t + 2u_t'Wx_t) + \beta E_tV_j(x_{t+1})\Big\}$$

$$V_j(x_t) = -x_t'P_jx_t - \rho_j$$

where P_j and ρ_j satisfy the equations

$$\begin{aligned}P_{j+1} &= R + \beta A'P_jA - (\beta A'P_jB + W)(Q + \beta B'P_jB)^{-1}(\beta B'P_jA + W')\\ \rho_{j+1} &= \beta\rho_j + \beta \text{ trace } P_jCC'\end{aligned}$$

We can now state the planning problem as a dynamic programming problem

$$\max_{\{u_t, x_{t+1}\}} -E\sum_{t=0}^{\infty}\beta^t[x_t'Rx_t + u_t'Qu_t + 2u_t'W'x_t], \quad 0 < \beta < 1$$

where maximization is subject to

$$x_{t+1} = Ax_t + Bu_t + Cw_{t+1},\ t \geq 0$$

$$x_t = \begin{bmatrix} h_{t-1} \\ k_{t-1} \\ z_t \end{bmatrix}, \qquad u_t = i_t$$

where

$$A = \begin{bmatrix} \Delta_h & \Theta_h U_c [\Phi_c \ \ \Phi_g]^{-1} \Gamma & \Theta_h U_c [\Phi_c \ \ \Phi_g]^{-1} U_d \\ 0 & \Delta_k & 0 \\ 0 & 0 & A_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} -\Theta_h U_c [\Phi_c \ \ \Phi_g]^{-1} \Phi_i \\ \Theta_k \\ 0 \end{bmatrix},\ C = \begin{bmatrix} 0 \\ 0 \\ C_2 \end{bmatrix}$$

$$\begin{bmatrix} x_t \\ u_t \end{bmatrix}' S \begin{bmatrix} x_t \\ u_t \end{bmatrix} = \begin{bmatrix} x_t \\ u_t \end{bmatrix}' \begin{bmatrix} R & W \\ W' & Q \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix}$$

$$S = (G'G + H'H)/2$$

$$H = [\Lambda \ \vdots \ \Pi U_c [\Phi_c \ \ \Phi_g]^{-1} \Gamma \ \vdots \ \Pi U_c [\Phi_c \ \ \Phi_g]^{-1} U_d - U_b \ \vdots \ -\Pi U_c [\Phi_c \ \ \Phi_g]^{-1} \Phi_i]$$

$$G = U_g [\Phi_c \ \ \Phi_g]^{-1} [0 \ \vdots \ \Gamma \ \vdots \ U_d \ \vdots \ -\Phi_i].$$

Lagrange multipliers as gradient of value function

A useful fact is that Lagrange multipliers equal gradients of the planner's value function

$$\begin{aligned} \mathcal{M}_t^k &= M_k x_t \ \text{ and } \ M_t^h = M_h x_t \ \text{ where} \\ M_k &= 2\beta [0 \ I \ 0] P A^o \\ M_h &= 2\beta [I \ 0 \ 0] P A^o \end{aligned}$$

$$\mathcal{M}_t^s = M_s x_t \ \text{ where } \ M_s = (S_b - S_s) \ \text{ and } \ S_b = [0 \ 0 \ U_b]$$

$$\mathcal{M}_t^d = M_d x_t \ \text{ where } \ M_d = \begin{bmatrix} \Phi_c' \\ \Phi_g' \end{bmatrix}^{-1} \begin{bmatrix} \Theta_h' M_h + \Pi' M_s \\ -S_g \end{bmatrix}$$

$$\mathcal{M}_t^c = M_c x_t \ \text{ where } \ M_c = \Theta_h' M_h + \Pi' M_s$$

$$\mathcal{M}_t^i = M_i x_t \ \text{ where } \ M_i = \Theta_k' M_k$$

We will use this fact and these equations to compute competitive equilibrium prices.

18.1.18 Other mathematical infrastructure

Let's start with describing the **commodity space** and **pricing functional** for our competitive equilibrium.

For the **commodity space**, we use

$$L_0^2 = [\{y_t\} : y_t \text{ is a random variable in } J_t \text{ and } E \sum_{t=0}^{\infty} \beta^t y_t^2 \mid J_0 < +\infty]$$

For **pricing functionals**, we express values as inner products

$$\pi(c) = E \sum_{t=0}^{\infty} \beta^t p_t^0 \cdot c_t \mid J_0$$

where p_t^0 belongs to L_0^2.

With these objects in our toolkit, we move on to state the problem of a **Representative Household in a competitive equilibrium**.

18.1.19 Representative Household

The representative household owns endowment process and initial stocks of h and k and chooses stochastic processes for $\{c_t,\, s_t,\, h_t,\, \ell_t\}_{t=0}^\infty$, each element of which is in L_0^2, to maximize

$$-\frac{1}{2}\, E_0 \sum_{t=0}^\infty \beta^t \left[(s_t - b_t) \cdot (s_t - b_t) + \ell_t^2\right]$$

subject to

$$E \sum_{t=0}^\infty \beta^t\, p_t^0 \cdot c_t \mid J_0 = E \sum_{t=0}^\infty \beta^t\, (w_t^0 \ell_t + \alpha_t^0 \cdot d_t) \mid J_0 + v_0 \cdot k_{-1}$$

$$s_t = \Lambda h_{t-1} + \Pi c_t$$

$$h_t = \Delta_h h_{t-1} + \Theta_h c_t, \quad h_{-1}, k_{-1} \text{ given}$$

We now describe the problems faced by two types of firms called type I and type II.

18.1.20 Type I Firm

A type I firm rents capital and labor and endowments and produces c_t, i_t.

It chooses stochastic processes for $\{c_t, i_t, k_t, \ell_t, g_t, d_t\}$, each element of which is in L_0^2, to maximize

$$E_0\, \sum_{t=0}^\infty \beta^t\, (p_t^0 \cdot c_t + q_t^0 \cdot i_t - r_t^0 \cdot k_{t-1} - w_t^0 \ell_t - \alpha_t^0 \cdot d_t)$$

subject to

$$\Phi_c\, c_t + \Phi_g\, g_t + \Phi_i\, i_t = \Gamma k_{t-1} + d_t$$

$$-\,\ell_t^2 + g_t \cdot g_t = 0$$

18.1.21 Type II Firm

A firm of type II acquires capital via investment and then rents stocks of capital to the c, i-producing type I firm.

A type II firm is a price taker facing the vector v_0 and the stochastic processes $\{r_t^0, q_t^0\}$.

The firm chooses k_{-1} and stochastic processes for $\{k_t, i_t\}_{t=0}^\infty$ to maximize

$$E \sum_{t=0}^\infty \beta^t (r_t^0 \cdot k_{t-1} - q_t^0 \cdot i_t) \mid J_0 - v_0 \cdot k_{-1}$$

subject to

$$k_t = \Delta_k k_{t-1} + \Theta_k i_t$$

18.1.22 Competitive Equilibrium: Definition

We can now state the following.

Definition: A competitive equilibrium is a price system $[v_0, \{p_t^0, w_t^0, \alpha_t^0, q_t^0, r_t^0\}_{t=0}^\infty]$ and an allocation $\{c_t, i_t, k_t, h_t, g_t, d_t\}_{t=0}^\infty$ that satisfy the following conditions:

- Each component of the price system and the allocation resides in the space L_0^2.
- Given the price system and given h_{-1}, k_{-1}, the allocation solves the representative household's problem and the problems of the two types of firms.

Versions of the two classical welfare theorems prevail under our assumptions.

We exploit that fact in our algorithm for computing a competitive equilibrium.

Step 1: Solve the planning problem by using dynamic programming.

The allocation (i.e., **quantities**) that solve the planning problem **are** the competitive equilibrium quantities.

Step 2: use the following formulas to compute the **equilibrium price system**

$$p_t^0 = [\Pi' M_t^s + \Theta_h' M_t^h]/\mu_0^w = M_t^c/\mu_0^w$$

$$w_t^0 = | S_g x_t | /\mu_0^w$$

$$r_t^0 = \Gamma' M_t^d/\mu_0^w$$

$$q_t^0 = \Theta_k' M_t^k/\mu_0^w = M_t^i/\mu_0^w$$

$$\alpha_t^0 = M_t^d/\mu_0^w$$

$$v_0 = \Gamma' M_0^d/\mu_0^w + \Delta_k' M_0^k/\mu_0^w$$

Verification: With this price system, values can be assigned to the Lagrange multipliers for each of our three classes of agents that cause all first-order necessary conditions to be satisfied at these prices and at the quantities associated with the optimum of the planning problem.

18.1.23 Asset pricing

An important use of an equilibrium pricing system is to do asset pricing.

Thus, imagine that we are presented a dividend stream: $\{y_t\} \in L_0^2$ and want to compute the value of a perpetual claim to this stream.

To value this asset we simply take **price times quantity** and add to get an asset value: $a_0 = E \sum_{t=0}^\infty \beta^t \, p_t^0 \cdot y_t \mid J_0$.

To compute ao we proceed as follows.

We let

$$y_t = U_a \, x_t$$

$$a_0 = E \sum_{t=0}^\infty \beta^t \, x_t' \, Z_a x_t \mid J_0$$

$$Z_a = U_a' M_c/\mu_0^w$$

We have the following convenient formulas:

$$a_0 = x_0' \, \mu_a \, x_0 + \sigma_a$$

$$\mu_a = \sum_{\tau=0}^{\infty} \beta^\tau \, (A^{o\prime})^\tau \, Z_a \, A^{o\tau}$$
$$\sigma_a = \frac{\beta}{1-\beta} \, \text{trace} \left(Z_a \sum_{\tau=0}^{\infty} \beta^\tau \, (A^o)^\tau \, CC' (A^{o\prime})^\tau \right)$$

18.1.24 Re-Opening Markets

We have assumed that all trading occurs once-and-for-all at time $t = 0$.

If we were to **re-open markets** at some time $t > 0$ at time t wealth levels implicitly defined by time 0 trades, we would obtain the same equilibrium allocation (i.e., quantities) and the following time t price system

$$L_t^2 = [\{y_s\}_{s=t}^{\infty} : \; y_s \; \text{ is a random variable in } \; J_s \; \text{ for } \; s \geq t$$
$$\text{and } E \sum_{s=t}^{\infty} \beta^{s-t} \, y_s^2 \mid J_t < +\infty].$$

$$p_s^t = M_c x_s / [\bar{e}_j M_c x_t], \qquad s \geq t$$
$$w_s^t = \mid S_g x_s | / [\bar{e}_j M_c x_t], \;\; s \geq t$$
$$r_s^t = \Gamma' M_d x_s / [\bar{e}_j M_c x_t], \;\; s \geq t$$
$$q_s^t = M_i x_s / [\bar{e}_j \, M_c x_t], \qquad s \geq t$$
$$\alpha_s^t = M_d x_s / [\bar{e}_j \, M_c x_t], \;\; s \geq t$$
$$v_t = [\Gamma' M_d + \Delta_k' M_k] x_t / \, [\bar{e}_j \, M_c x_t]$$

18.2 Econometrics

Up to now, we have described how to solve the **direct problem** that maps model parameters into an (equilibrium) stochastic process of prices and quantities.

Recall the **inverse problem** of inferring model parameters from a single realization of a time series of some of the prices and quantities.

Another name for the inverse problem is **econometrics**.

An advantage of the [HS13] structure is that it comes with a self-contained theory of econometrics.

It is really just a tale of two state-space representations.

Here they are:

Original State-Space Representation:

$$x_{t+1} = A^o x_t + C w_{t+1}$$
$$y_t = G x_t + v_t$$

where v_t is a martingale difference sequence of measurement errors that satisfies $E v_t v_t' = R, E w_{t+1} v_s' = 0$ for all $t + 1 \geq s$ and

$$x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_0)$$

Innovations Representation:

$$\begin{aligned}\hat{x}_{t+1} &= A^o \hat{x}_t + K_t a_t \\ y_t &= G \hat{x}_t + a_t,\end{aligned}$$

where $a_t = y_t - E[y_t | y^{t-1}]$, $E a_t a_t' \equiv \Omega_t = G \Sigma_t G' + R$.

Compare numbers of shocks in the two representations:

- $n_w + n_y$ versus n_y

Compare spaces spanned

- $H(y^t) \subset H(w^t, v^t)$
- $H(y^t) = H(a^t)$

Kalman Filter:.

Kalman gain:

$$K_t = A^o \Sigma_t G' (G \Sigma_t G' + R)^{-1}$$

Riccati Difference Equation:

$$\begin{aligned}\Sigma_{t+1} = & A^o \Sigma_t A^{o\prime} + CC' \\ & - A^o \Sigma_t G' (G \Sigma_t G' + R)^{-1} G \Sigma_t A^{o\prime}\end{aligned}$$

Innovations Representation as Whitener

Whitening Filter:

$$\begin{aligned}a_t &= y_t - G \hat{x}_t \\ \hat{x}_{t+1} &= A^o \hat{x}_t + K_t a_t\end{aligned}$$

can be used recursively to construct a record of innovations $\{a_t\}_{t=0}^T$ from an $(\hat{x}_0, \Sigma_0)$ and a record of observations $\{y_t\}_{t=0}^T$.

Limiting Time-Invariant Innovations Representation

$$\begin{aligned}\Sigma = & A^o \Sigma A^{o\prime} + CC' \\ & - A^o \Sigma G' (G \Sigma G' + R)^{-1} G \Sigma A^{o\prime} \\ K = & A^o \Sigma_t G' (G \Sigma G' + R)^{-1}\end{aligned}$$

$$\begin{aligned}\hat{x}_{t+1} &= A^o \hat{x}_t + K a_t \\ y_t &= G \hat{x}_t + a_t\end{aligned}$$

where $E a_t a_t' \equiv \Omega = G \Sigma G' + R$.

18.2.1 Factorization of Likelihood Function

Sample of observations $\{y_s\}_{s=0}^T$ on a $(n_y \times 1)$ vector.

$$\begin{aligned}f(y_T, y_{T-1}, \ldots, y_0) &= f_T(y_T | y_{T-1}, \ldots, y_0) f_{T-1}(y_{T-1} | y_{T-2}, \ldots, y_0) \cdots f_1(y_1 | y_0) f_0(y_0) \\ &= g_T(a_T) g_{T-1}(a_{T-1}) \ldots g_1(a_1) f_0(y_0).\end{aligned}$$

Gaussian Log-Likelihood:

$$-.5 \sum_{t=0}^T \left\{ n_y \ln(2\pi) + \ln |\Omega_t| + a_t' \Omega_t^{-1} a_t \right\}$$

18.2.2 Covariance Generating Functions

Autocovariance: $C_x(\tau) = Ex_t x'_{t-\tau}$.

Generating Function: $S_x(z) = \sum_{\tau=-\infty}^{\infty} C_x(\tau) z^\tau, z \in C$.

18.2.3 Spectral Factorization Identity

Original state-space representation has too many shocks and implies:

$$S_y(z) = G(zI - A^o)^{-1}CC'(z^{-1}I - (A^o)')^{-1}G' + R$$

Innovations representation has as many shocks as dimension of y_t and implies

$$S_y(z) = [G(zI - A^o)^{-1}K + I][G\Sigma G' + R][K'(z^{-1}I - A^{o\prime})^{-1}G' + I]$$

Equating these two leads to:

$$\begin{aligned}&G(zI - A^o)^{-1}CC'(z^{-1}I - A^{o\prime})^{-1}G' + R = \\ &[G(zI - A^o)^{-1}K + I][G\Sigma G' + R][K'(z^{-1}I - A^{o\prime})^{-1}G' + I].\end{aligned}$$

Key Insight: The zeros of the polynomial $\det[G(zI - A^o)^{-1}K + I]$ all lie inside the unit circle, which means that a_t lies in the space spanned by square summable linear combinations of y^t.

$$H(a^t) = H(y^t)$$

Key Property: Invertibility

18.2.4 Wold and Vector Autoregressive Representations

Let's start with some lag operator arithmetic.

The lag operator L and the inverse lag operator L^{-1} each map an infinite sequence into an infinite sequence according to the transformation rules

$$Lx_t \equiv x_{t-1}$$

$$L^{-1}x_t \equiv x_{t+1}$$

A **Wold moving average representation** for $\{y_t\}$ is

$$y_t = [G(I - A^oL)^{-1}KL + I]a_t$$

Applying the inverse of the operator on the right side and using

$$[G(I - A^oL)^{-1}KL + I]^{-1} = I - G[I - (A^o - KG)L]^{-1}KL$$

gives the **vector autoregressive representation**

$$y_t = \sum_{j=1}^{\infty} G(A^o - KG)^{j-1}Ky_{t-j} + a_t$$

18.3 Dynamic Demand Curves and Canonical Household Technologies

18.3.1 Canonical Household Technologies

$$
\begin{aligned}
h_t &= \Delta_h h_{t-1} + \Theta_h c_t \\
s_t &= \Lambda h_{t-1} + \Pi c_t \\
b_t &= U_b z_t
\end{aligned}
$$

Definition: A household service technology $(\Delta_h, \Theta_h, \Pi, \Lambda, U_b)$ is said to be **canonical** if

- Π is nonsingular, and
- the absolute values of the eigenvalues of $(\Delta_h - \Theta_h \Pi^{-1} \Lambda)$ are strictly less than $1/\sqrt{\beta}$.

Key invertibility property: A canonical household service technology maps a service process $\{s_t\}$ in L_0^2 into a corresponding consumption process $\{c_t\}$ for which the implied household capital stock process $\{h_t\}$ is also in L_0^2.

An inverse household technology:

$$
\begin{aligned}
c_t &= -\Pi^{-1} \Lambda h_{t-1} + \Pi^{-1} s_t \\
h_t &= (\Delta_h - \Theta_h \Pi^{-1} \Lambda) h_{t-1} + \Theta_h \Pi^{-1} s_t
\end{aligned}
$$

The restriction on the eigenvalues of the matrix $(\Delta_h - \Theta_h \Pi^{-1} \Lambda)$ keeps the household capital stock $\{h_t\}$ in L_0^2.

18.3.2 Dynamic Demand Functions

$$
\rho_t^0 \equiv \Pi^{-1\prime} \Big[p_t^0 - \Theta_h' E_t \sum_{\tau=1}^\infty \beta^\tau (\Delta_h' - \Lambda' \Pi^{-1\prime} \Theta_h')^{\tau-1} \Lambda' \Pi^{-1\prime} p_{t+\tau}^0 \Big]
$$

$$
\begin{aligned}
s_{i,t} &= \Lambda h_{i,t-1} \\
h_{i,t} &= \Delta_h h_{i,t-1}
\end{aligned}
$$

where $h_{i,-1} = h_{-1}$.

$$
W_0 = E_0 \sum_{t=0}^\infty \beta^t (w_t^0 \ell_t + \alpha_t^0 \cdot d_t) + v_0 \cdot k_{-1}
$$

$$
\mu_0^w = \frac{E_0 \sum_{t=0}^\infty \beta^t \rho_t^0 \cdot (b_t - s_{i,t}) - W_0}{E_0 \sum_{t=0}^\infty \beta^t \rho_t^0 \cdot \rho_t^0}
$$

$$
\begin{aligned}
c_t &= -\Pi^{-1} \Lambda h_{t-1} + \Pi^{-1} b_t - \Pi^{-1} \mu_0^w E_t \{ \Pi'^{\,-1} - \Pi'^{\,-1} \Theta_h' \\
&\qquad [I - (\Delta_h' - \Lambda' \Pi'^{\,-1} \Theta_h') \beta L^{-1}]^{-1} \Lambda' \Pi'^{-1} \beta L^{-1} \} p_t^0 \\
h_t &= \Delta_h h_{t-1} + \Theta_h c_t
\end{aligned}
$$

This system expresses consumption demands at date t as functions of: (i) time-t conditional expectations of future scaled Arrow-Debreu prices $\{p_{t+s}^0\}_{s=0}^\infty$; (ii) the stochastic process for the household's endowment $\{d_t\}$ and preference shock $\{b_t\}$, as mediated through the multiplier μ_0^w and wealth W_0; and (iii) past values of consumption, as mediated through the state variable h_{t-1}.

18.4 Gorman Aggregation and Engel Curves

We shall explore how the dynamic demand schedule for consumption goods opens up the possibility of satisfying Gorman's (1953) conditions for aggregation in a heterogeneous consumer model.

The first equation of our demand system is an Engel curve for consumption that is linear in the marginal utility μ_0^2 of individual wealth with a coefficient on μ_0^w that depends only on prices.

The multiplier μ_0^w depends on wealth in an affine relationship, so that consumption is linear in wealth.

In a model with multiple consumers who have the same household technologies $(\Delta_h, \Theta_h, \Lambda, \Pi)$ but possibly different preference shock processes and initial values of household capital stocks, the coefficient on the marginal utility of wealth is the same for all consumers.

Gorman showed that when Engel curves satisfy this property, there exists a unique community or aggregate preference ordering over aggregate consumption that is independent of the distribution of wealth.

18.4.1 Re-Opened Markets

$$
\begin{aligned}
\rho_t^t \equiv \Pi^{-1\prime}\Big[p_t^t - \Theta_h' E_t \sum_{\tau=1}^{\infty} \beta^\tau (\Delta_h' - \Lambda'\Pi^{-1\prime}\Theta_h')^{\tau-1}\Lambda'\Pi^{-1\prime}p_{t+\tau}^t\Big] \\
s_{i,t} = \Lambda h_{i,t-1} \\
h_{i,t} = \Delta_h h_{i,t-1},
\end{aligned}
$$

where now $h_{i,t-1} = h_{t-1}$. Define time t wealth W_t

$$
W_t = E_t \sum_{j=0}^{\infty} \beta^j (w_{t+j}^t \ell_{t+j} + \alpha_{t+j}^t \cdot d_{t+j}) + v_t \cdot k_{t-1}
$$

$$
\mu_t^w = \frac{E_t \sum_{j=0}^{\infty} \beta^j \rho_{t+j}^t \cdot (b_{t+j} - s_{i,t+j}) - W_t}{E_t \sum_{t=0}^{\infty} \beta^j \rho_{t+j}^t \cdot \rho_{t+j}^t}
$$

$$
\begin{aligned}
c_t &= -\Pi^{-1}\Lambda h_{t-1} + \Pi^{-1} b_t - \Pi^{-1}\mu_t^w E_t \{\Pi'^{\,-1} - \Pi'^{\,-1}\Theta_h' \\
&\quad [I - (\Delta_h' - \Lambda'\Pi'^{\,-1}\Theta_h')\beta L^{-1}]^{-1}\Lambda'\Pi'^{-1}\beta L^{-1}\} p_t^t \\
h_t &= \Delta_h h_{t-1} + \Theta_h c_t
\end{aligned}
$$

18.4.2 Dynamic Demand

Define a time t continuation of a sequence $\{z_t\}_{t=0}^{\infty}$ as the sequence $\{z_\tau\}_{\tau=t}^{\infty}$. The demand system indicates that the time t vector of demands for c_t is influenced by:

Through the multiplier μ_t^w, the time t continuation of the preference shock process $\{b_t\}$ and the time t continuation of $\{s_{i,t}\}$.

The time $t-1$ level of household durables h_{t-1}.

Everything that affects the household's time t wealth, including its stock of physical capital k_{t-1} and its value v_t, the time t continuation of the factor prices $\{w_t, \alpha_t\}$, the household's continuation endowment process, and the household's continuation plan for $\{\ell_t\}$.

The time t continuation of the vector of prices $\{p_t^t\}$.

18.4.3 Attaining a Canonical Household Technology

Apply the following version of a factorization identity:

$$\begin{aligned}[\Pi + \beta^{1/2} L^{-1} \Lambda (I - \beta^{1/2} L^{-1} \Delta_h)^{-1} \Theta_h]' [\Pi + \beta^{1/2} L \Lambda (I - \beta^{1/2} L \Delta_h)^{-1} \Theta_h] \\ = [\hat\Pi + \beta^{1/2} L^{-1} \hat\Lambda (I - \beta^{1/2} L^{-1} \Delta_h)^{-1} \Theta_h]' [\hat\Pi + \beta^{1/2} L \hat\Lambda (I - \beta^{1/2} L \Delta_h)^{-1} \Theta_h]\end{aligned}$$

The factorization identity guarantees that the $[\hat\Lambda, \hat\Pi]$ representation satisfies both requirements for a canonical representation.

18.5 Partial Equilibrium

Now we'll provide quick overviews of examples of economies that fit within our framework

We provide details for a number of these examples in subsequent lectures

1. *Growth in Dynamic Linear Economies*
2. *Lucas Asset Pricing using DLE*
3. *IRFs in Hall Model*
4. *Permanent Income Using the DLE class*
5. *Rosen schooling model*
6. *Cattle cycles*
7. *Shock Non Invertibility*

We'll start with an example of a **partial equilibrium** in which we posit demand and supply curves

Suppose that we want to capture the dynamic demand curve:

$$\begin{aligned}c_t &= -\Pi^{-1} \Lambda h_{t-1} + \Pi^{-1} b_t - \Pi^{-1} \mu_0^w E_t \{ \Pi'^{-1} - \Pi'^{-1} \Theta_h' \\ &\quad [I - (\Delta_h' - \Lambda' \Pi'^{-1} \Theta_h') \beta L^{-1}]^{-1} \Lambda' \Pi'^{-1} \beta L^{-1} \} p_t \\ h_t &= \Delta_h h_{t-1} + \Theta_h c_t\end{aligned}$$

From material described earlier in this lecture, we know how to reverse engineer preferences that generate this demand system

- note how the demand equations are cast in terms of the matrices in our standard preference representation

Now let's turn to supply.

A representative firm takes as given and beyond its control the stochastic process $\{p_t\}_{t=0}^\infty$.

The firm sells its output c_t in a competitive market each period.

Only spot markets convene at each date $t \geq 0$.

The firm also faces an exogenous process of cost disturbances d_t.

The firm chooses stochastic processes $\{c_t, g_t, i_t, k_t\}_{t=0}^\infty$ to maximize

$$E_0 \sum_{t=0}^\infty \beta^t \{ p_t \cdot c_t - g_t \cdot g_t / 2 \}$$

subject to given k_{-1} and

$$\begin{aligned}\Phi_c c_t + \Phi_i i_t + \Phi_g g_t = \Gamma k_{t-1} + d_t \\ k_t = \Delta_k k_{t-1} + \Theta_k i_t.\end{aligned}$$

18.6 Equilibrium Investment Under Uncertainty

A representative firm maximizes

$$E\sum_{t=0}^{\infty}\beta^t\{p_tc_t - g_t^2/2\}$$

subject to the technology

$$\begin{aligned} c_t &= \gamma k_{t-1} \\ k_t &= \delta_k k_{t-1} + i_t \\ g_t &= f_1 i_t + f_2 d_t \end{aligned}$$

where d_t is a cost shifter, $\gamma > 0$, and $f_1 > 0$ is a cost parameter and $f_2 = 1$. Demand is governed by

$$p_t = \alpha_0 - \alpha_1 c_t + u_t$$

where u_t is a demand shifter with mean zero and α_0, α_1 are positive parameters.

Assume that u_t, d_t are uncorrelated first-order autoregressive processes.

18.7 A Rosen-Topel Housing Model

$$\begin{aligned} R_t &= b_t + \alpha h_t \\ p_t &= E_t\sum_{\tau=0}^{\infty}(\beta\delta_h)^\tau R_{t+\tau} \end{aligned}$$

where h_t is the stock of housing at time t R_t is the rental rate for housing, p_t is the price of new houses, and b_t is a demand shifter; $\alpha < 0$ is a demand parameter, and δ_h is a depreciation factor for houses.

We cast this demand specification within our class of models by letting the stock of houses h_t evolve according to

$$h_t = \delta_h h_{t-1} + c_t, \quad \delta_h \in (0,1)$$

where c_t is the rate of production of new houses.

Houses produce services s_t according to $s_t = \bar{\lambda} h_t$ or $s_t = \lambda h_{t-1} + \pi c_t$, where $\lambda = \bar{\lambda}\delta_h, \pi = \bar{\lambda}$.

We can take $\bar{\lambda}\rho_t^0 = R_t$ as the rental rate on housing at time t, measured in units of time t consumption (housing).

Demand for housing services is

$$s_t = b_t - \mu_0\rho_t^0$$

where the price of new houses p_t is related to ρ_t^0 by $\rho_t^0 = \pi^{-1}[p_t - \beta\delta_h E_t p_{t+1}]$.

18.8 Cattle Cycles

Rosen, Murphy, and Scheinkman (1994). Let p_t be the price of freshly slaughtered beef, m_t the feeding cost of preparing an animal for slaughter, $\tilde{h}_t$ the one-period holding cost for a mature animal, $\gamma_1\tilde{h}_t$ the one-period holding cost for a yearling, and $\gamma_0\tilde{h}_t$ the one-period holding cost for a calf.

The cost processes $\{\tilde{h}_t, m_t\}_{t=0}^{\infty}$ are exogenous, while the stochastic process $\{p_t\}_{t=0}^{\infty}$ is determined by a rational expectations equilibrium. Let $\tilde{x}_t$ be the breeding stock, and $\tilde{y}_t$ be the total stock of animals.

The law of motion for cattle stocks is

$$\tilde{x}_t = (1-\delta)\tilde{x}_{t-1} + g\tilde{x}_{t-3} - c_t$$

where c_t is a rate of slaughtering. The total head-count of cattle

$$\tilde{y}_t = \tilde{x}_t + g\tilde{x}_{t-1} + g\tilde{x}_{t-2}$$

is the sum of adults, calves, and yearlings, respectively.

A representative farmer chooses $\{c_t, \tilde{x}_t\}$ to maximize

$$\begin{aligned} E_0 \sum_{t=0}^{\infty} \beta^t \{p_t c_t - \tilde{h}_t \tilde{x}_t - (\gamma_0 \tilde{h}_t)(g\tilde{x}_{t-1}) - (\gamma_1 \tilde{h}_t)(g\tilde{x}_{t-2}) - m_t c_t \\ - \Psi(\tilde{x}_t, \tilde{x}_{t-1}, \tilde{x}_{t-2}, c_t)\} \end{aligned}$$

where

$$\Psi = \frac{\psi_1}{2}\tilde{x}_t^2 + \frac{\psi_2}{2}\tilde{x}_{t-1}^2 + \frac{\psi_3}{2}\tilde{x}_{t-2}^2 + \frac{\psi_4}{2}c_t^2$$

Demand is governed by

$$c_t = \alpha_0 - \alpha_1 p_t + \tilde{d}_t$$

where $\alpha_0 > 0$, $\alpha_1 > 0$, and $\{\tilde{d}_t\}_{t=0}^{\infty}$ is a stochastic process with mean zero representing a demand shifter.

For more details see *Cattle cycles*

18.9 Models of Occupational Choice and Pay

We'll describe the following pair of schooling models that view education as a time-to-build process:

- Rosen schooling model for engineers
- Two-occupation model

18.9.1 Market for Engineers

Ryoo and Rosen's (2004) [RR04] model consists of the following equations:

first, a demand curve for engineers

$$w_t = -\alpha_d N_t + \epsilon_{1t} \ , \ \alpha_d > 0$$

second, a time-to-build structure of the education process

$$N_{t+k} = \delta_N N_{t+k-1} + n_t \ , \ 0 < \delta_N < 1$$

third, a definition of the discounted present value of each new engineering student

$$v_t = \beta^k E_t \sum_{j=0}^{\infty} (\beta\delta_N)^j w_{t+k+j};$$

and fourth, a supply curve of new students driven by v_t

$$n_t = \alpha_s v_t + \epsilon_{2t} \ , \ \alpha_s > 0$$

Here $\{\epsilon_{1t}, \epsilon_{2t}\}$ are stochastic processes of labor demand and supply shocks.

Definition: A partial equilibrium is a stochastic process $\{w_t, N_t, v_t, n_t\}_{t=0}^{\infty}$ satisfying these four equations, and initial conditions $N_{-1}, n_{-s}, s = 1, \ldots, -k$.

We sweep the time-to-build structure and the demand for engineers into the household technology and putting the supply of new engineers into the technology for producing goods.

$$s_t = [\lambda_1 \; 0 \; \ldots \; 0] \begin{bmatrix} h_{1t-1} \\ h_{2t-1} \\ \vdots \\ h_{k+1,t-1} \end{bmatrix} + 0 \cdot c_t$$

$$\begin{bmatrix} h_{1t} \\ h_{2t} \\ \vdots \\ h_{k,t} \\ h_{k+1,t} \end{bmatrix} = \begin{bmatrix} \delta_N & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} h_{1t-1} \\ h_{2t-1} \\ \vdots \\ h_{k,t-1} \\ h_{k+1,t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} c_t$$

This specification sets Rosen's $N_t = h_{1t-1}, n_t = c_t, h_{\tau+1,t-1} = n_{t-\tau}, \tau = 1, \ldots, k$, and uses the home-produced service to capture the demand for labor. Here λ_1 embodies Rosen's demand parameter α_d.

- The supply of new workers becomes our consumption.
- The dynamic demand curve becomes Rosen's dynamic supply curve for new workers.

Remark: This has an Imai-Keane flavor.

For more details and Python code see *Rosen schooling model*.

18.9.2 Skilled and Unskilled Workers

First, a demand curve for labor

$$\begin{bmatrix} w_{ut} \\ w_{st} \end{bmatrix} = \alpha_d \begin{bmatrix} N_{ut} \\ N_{st} \end{bmatrix} + \epsilon_{1t}$$

where α_d is a (2×2) matrix of demand parameters and ϵ_{1t} is a vector of demand shifters second, time-to-train specifications for skilled and unskilled labor, respectively:

$$\begin{aligned} N_{st+k} &= \delta_N N_{st+k-1} + n_{st} \\ N_{ut} &= \delta_N N_{ut-1} + n_{ut}; \end{aligned}$$

where N_{st}, N_{ut} are stocks of the two types of labor, and n_{st}, n_{ut} are entry rates into the two occupations.

third, definitions of discounted present values of new entrants to the skilled and unskilled occupations, respectively:

$$v_{st} = E_t \beta^k \sum_{j=0}^{\infty} (\beta \delta_N)^j w_{st+k+j}$$

$$v_{ut} = E_t \sum_{j=0}^{\infty} (\beta \delta_N)^j w_{ut+j}$$

where w_{ut}, w_{st} are wage rates for the two occupations; and fourth, supply curves for new entrants:

$$\begin{bmatrix} n_{st} \\ n_{ut} \end{bmatrix} = \alpha_s \begin{bmatrix} v_{ut} \\ v_{st} \end{bmatrix} + \epsilon_{2t}$$

Short Cut

As an alternative, Siow simply used the **equalizing differences** condition

$$v_{ut} = v_{st}$$

18.10 Permanent Income Models

We'll describe a class of permanent income models that feature

- Many consumption goods and services
- A single capital good with $R\beta = 1$
- The physical production technology

$$\begin{aligned} \phi_c \cdot c_t + i_t &= \gamma k_{t-1} + e_t \\ k_t &= k_{t-1} + i_t \end{aligned}$$

$$\phi_i i_t - g_t = 0$$

Implication One:

Equality of Present Values of Moving Average Coefficients of c and e

$$k_{t-1} = \beta \sum_{j=0}^{\infty} \beta^j (\phi_c \cdot c_{t+j} - e_{t+j}) \quad \Rightarrow$$

$$k_{t-1} = \beta \sum_{j=0}^{\infty} \beta^j E(\phi_c \cdot c_{t+j} - e_{t+j}) | J_t \quad \Rightarrow$$

$$\sum_{j=0}^{\infty} \beta^j (\phi_c)' \chi_j = \sum_{j=0}^{\infty} \beta^j \epsilon_j$$

where $\chi_j w_t$ is the response of c_{t+j} to w_t and $\epsilon_j w_t$ is the response of endowment e_{t+j} to w_t:

Implication Two:

Martingales

$$\begin{aligned} \mathcal{M}_t^k &= E(\mathcal{M}_{t+1}^k | J_t) \\ \mathcal{M}_t^e &= E(\mathcal{M}_{t+1}^e | J_t) \end{aligned}$$

and

$$\mathcal{M}_t^c = (\Phi_c)' \mathcal{M}_t^d = \phi_c M_t^e$$

For more details see *Permanent Income Using the DLE class*

Testing Permanent Income Models:

We have two types of implications of permanent income models:

- Equality of present values of moving average coefficients.
- Martingale $\mathcal{M}_t^k$.

These have been tested in work by Hansen, Sargent, and Roberts (1991) [SHR91] and by Attanasio and Pavoni (2011) [AP11].

18.11 Gorman Heterogeneous Households

We now assume that there is a finite number of households, each with its own household technology and preferences over consumption services.

Household j orders preferences over consumption processes according to

$$-\left(\frac{1}{2}\right)\ E\sum_{t=0}^{\infty}\beta^t\left[(s_{jt}-b_{jt})\cdot(s_{jt}-b_{jt})+\ell_{jt}^2\right]\mid J_0$$

$$s_{jt}=\Lambda\, h_{j,t-1}+\Pi\, c_{jt}$$

$$h_{jt}=\Delta_h\, h_{j,t-1}+\Theta_h\, c_{jt}$$

and $h_{j,-1}$ is given

$$b_{jt}=U_{bj}z_t$$

$$E\sum_{t=0}^{\infty}\beta^t\, p_t^0\ \cdot c_{jt}\mid J_0=E\sum_{t=0}^{\infty}\beta^t\,(w_t^0\,\ell_{jt}+\alpha_t^0\ \cdot d_{jt})\mid J_0+v_0\ \cdot k_{j,-1},$$

where $k_{j,-1}$ is given. The j^{th} consumer owns an endowment process d_{jt}, governed by the stochastic process $d_{jt}=U_{dj}\,z_t$.

We refer to this as a setting with Gorman heterogeneous households.

This specification confines heterogeneity among consumers to:

- differences in the preference processes $\{b_{jt}\}$, represented by different selections of U_{bj}
- differences in the endowment processes $\{d_{jt}\}$, represented by different selections of U_{dj}
- differences in $h_{j,-1}$ and
- differences in $k_{j,-1}$

The matrices Λ, Π, Δ_h, Θ_h do not depend on j.

This makes everybody's demand system have the form described earlier, with different μ_{j0}^w's (reflecting different wealth levels) and different b_{jt} preference shock processes and initial conditions for household capital stocks.

Punchline: there exists a representative consumer.

We can use the representative consumer to compute a competitive equilibrium **aggregate** allocation and price system.

With the equilibrium aggregate allocation and price system in hand, we can then compute allocations to each household.

Computing Allocations to Individuals:

Set

$$\ell_{jt}=(\mu_{0j}^w/\mu_{0a}^w)\ell_{at}$$

Then solve the following equation for μ_{0j}^w:

$$\mu_{0j}^w E_0\sum_{t=0}^{\infty}\beta^t\{\rho_t^0\cdot\rho_t^0+(w_t^0/\mu_{0a}^w)\ell_{at}\}=E_0\sum_{t=0}^{\infty}\beta^t\{\rho_t^0\cdot(b_{jt}-s_{jt}^i)-\alpha_t^0\cdot d_{jt}\}-v_0k_{j,-1}$$

$$s_{jt}-b_{jt}=\mu_{0j}^w\rho_t^0$$

$$\begin{aligned}c_{jt}&=-\Pi^{-1}\Lambda h_{j,t-1}+\Pi^{-1}s_{jt}\\ h_{jt}&=(\Delta_h-\Theta_h\Pi^{-1}\Lambda)h_{j,t-1}+\Pi^{-1}\Theta_h s_{jt}\end{aligned}$$

Here $h_{j,-1}$ given.

18.12 Non-Gorman Heterogeneous Households

We now describe a less tractable type of heterogeneity across households that we dub **Non-Gorman heterogeneity**.

Here is the specification:

Preferences and Household Technologies:

$$-\frac{1}{2}E\sum_{t=0}^{\infty}\beta^t\left[(s_{it}-b_{it})\cdot(s_{it}-b_{it})+\ell_{it}^2\right]\mid J_0$$

$$\begin{aligned}s_{it}&=\Lambda_i h_{it-1}+\Pi_i\,c_{it}\\ h_{it}&=\Delta_{h_i}\,h_{it-1}+\Theta_{h_i}c_{it}\;,\;i=1,2.\end{aligned}$$

$$b_{it}=U_{bi}z_t$$

$$z_{t+1}=A_{22}z_t+C_2w_{t+1}$$

Production Technology

$$\Phi_c(c_{1t}+c_{2t})+\Phi_g g_t+\Phi_i i_t=\Gamma k_{t-1}+d_{1t}+d_{2t}$$

$$k_t=\Delta_k k_{t-1}+\Theta_k i_t$$

$$g_t\cdot g_t=\ell_t^2,\qquad \ell_t=\ell_{1t}+\ell_{2t}$$

$$d_{it}=U_{d_i}z_t,\qquad i=1,2$$

Pareto Problem:

$$\begin{aligned}&-\frac{1}{2}\,\lambda E_0\sum_{t=0}^{\infty}\beta^t[(s_{1t}-b_{1t})\cdot(s_{1t}-b_{1t})+\ell_{1t}^2]\\ &-\frac{1}{2}\,(1-\lambda)E_0\sum_{t=0}^{\infty}\beta^t[(s_{2t}-b_{2t})\cdot(s_{2t}-b_{2t})+\ell_{2t}^2]\end{aligned}$$

Mongrel Aggregation: Static

There is what we call a kind of **mongrel aggregation** in this setting.

We first describe the idea within a simple static setting in which there is a single consumer static inverse demand with implied preferences:

$$c_t=\Pi^{-1}b_t-\mu_0\Pi^{-1}\Pi^{-1\prime}p_t$$

An inverse demand curve is

$$p_t=\mu_0^{-1}\Pi' b_t-\mu_0^{-1}\Pi'\Pi c_t$$

Integrating the marginal utility vector shows that preferences can be taken to be

$$(-2\mu_0)^{-1}(\Pi c_t-b_t)\cdot(\Pi c_t-b_t)$$

Key Insight: Factor the inverse of a 'covariance matrix'.

Now assume that there are two consumers, $i=1,2$, with demand curves

$$c_{it}=\Pi_i^{-1}b_{it}-\mu_{0i}\Pi_i^{-1}\Pi_i^{-1\prime}p_t$$

$$c_{1t} + c_{2t} = (\Pi_1^{-1} b_{1t} + \Pi_2^{-1} b_{2t}) - (\mu_{01} \Pi_1^{-1} \Pi_1^{-1\prime} + \mu_{02} \Pi_2 \Pi_2^{-1\prime}) p_t$$

Setting $c_{1t} + c_{2t} = c_t$ and solving for p_t gives

$$\begin{aligned} p_t &= (\mu_{01} \Pi_1^{-1} \Pi_1^{-1\prime} + \mu_{02} \Pi_2^{-1} \Pi_2^{-1\prime})^{-1} (\Pi_1^{-1} b_{1t} + \Pi_2^{-1} b_{2t}) \\ &\quad - (\mu_{01} \Pi_1^{-1} \Pi_1^{-1\prime} + \mu_{02} \Pi_2^{-1} \Pi_2^{-1\prime})^{-1} c_t \end{aligned}$$

Punchline: choose Π associated with the aggregate ordering to satisfy

$$\mu_0^{-1} \Pi' \Pi = (\mu_{01} \Pi_1^{-1} \Pi_2^{-1\prime} + \mu_{02} \Pi_2^{-1} \Pi_2^{-1\prime})^{-1}$$

Dynamic Analogue:

We now describe how to extend mongrel aggregation to a dynamic setting.

The key comparison is

- Static: factor a covariance matrix-like object
- Dynamic: factor a spectral-density matrix-like object

Programming Problem for Dynamic Mongrel Aggregation:

Our strategy for deducing the mongrel preference ordering over $c_t = c_{1t} + c_{2t}$ is to solve the programming problem: choose $\{c_{1t}, c_{2t}\}$ to maximize the criterion

$$\sum_{t=0}^{\infty} \beta^t [\lambda (s_{1t} - b_{1t}) \cdot (s_{1t} - b_{1t}) + (1 - \lambda)(s_{2t} - b_{2t}) \cdot (s_{2t} - b_{2t})]$$

subject to

$$\begin{aligned} h_{jt} &= \Delta_{hj} \, h_{jt-1} + \Theta_{hj} \, c_{jt}, j = 1, 2 \\ s_{jt} &= \Delta_j h_{jt-1} + \Pi_j c_{jt} \, , j = 1, 2 \\ c_{1t} + c_{2t} &= c_t \end{aligned}$$

subject to $(h_{1,-1}, \, h_{2,-1})$ given and $\{b_{1t}\}$, $\{b_{2t}\}$, $\{c_t\}$ being known and fixed sequences.

Substituting the $\{c_{1t}, \, c_{2t}\}$ sequences that solve this problem as functions of $\{b_{1t}, \, b_{2t}, \, c_t\}$ into the objective determines a mongrel preference ordering over $\{c_t\} = \{c_{1t} + c_{2t}\}$.

In solving this problem, it is convenient to proceed by using Fourier transforms. For details, please see [HS13] where they deploy a

Secret Weapon: Another application of the spectral factorization identity.

Concluding remark: The [HS13] class of models described in this lecture are all complete markets models. We have exploited the fact that complete market models **are all alike** to allow us to define a class that **gives the same name to different things** in the spirit of Henri Poincare.

Could we create such a class for **incomplete markets** models?

That would be nice, but before trying it would be wise to contemplate the remainder of a statement by Robert E. Lucas, Jr., with which we began this lecture.

> "Complete market economies are all alike but each incomplete market economy is incomplete in its own individual way." Robert E. Lucas, Jr., (1989)

CHAPTER

NINETEEN

GROWTH IN DYNAMIC LINEAR ECONOMIES

Contents

This is another member of a suite of lectures that use the quantecon DLE class to instantiate models within the [HS13] class of models described in detail in *Recursive Models of Dynamic Linear Economies*.

In addition to what's included in Anaconda, this lecture uses the quantecon library.

```
!pip install --upgrade quantecon
```

This lecture describes several complete market economies having a common linear-quadratic-Gaussian structure.

Three examples of such economies show how the DLE class can be used to compute equilibria of such economies in Python and to illustrate how different versions of these economies can or cannot generate sustained growth.

We require the following imports

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from quantecon import LQ, DLE
```

19.1 Common Structure

Our example economies have the following features

- Information flows are governed by an exogenous stochastic process z_t that follows

$$z_{t+1} = A_{22} z_t + C_2 w_{t+1}$$

  where w_{t+1} is a martingale difference sequence.

- Preference shocks b_t and technology shocks d_t are linear functions of z_t

$$b_t = U_b z_t$$

$$d_t = U_d z_t$$

- Consumption and physical investment goods are produced using the following technology

$$\Phi_c c_t + \Phi_g g_t + \Phi_i i_t = \Gamma k_{t-1} + d_t$$

$$k_t = \Delta_k k_{t-1} + \Theta_k i_t$$

$$g_t \cdot g_t = l_t^2$$

  where c_t is a vector of consumption goods, g_t is a vector of intermediate goods, i_t is a vector of investment goods, k_t is a vector of physical capital goods, and l_t is the amount of labor supplied by the representative household.

- Preferences of a representative household are described by

$$-\frac{1}{2}\mathbb{E}\sum_{t=0}^{\infty}\beta^t[(s_t - b_t)\cdot(s_t - b_t) + l_t^2], 0 < \beta < 1$$

$$s_t = \Lambda h_{t-1} + \Pi c_t$$

$$h_t = \Delta_h h_{t-1} + \Theta_h c_t$$

where s_t is a vector of consumption services, and h_t is a vector of household capital stocks.

Thus, an instance of this class of economies is described by the matrices

$$\{A_{22}, C_2, U_b, U_d, \Phi_c, \Phi_g, \Phi_i, \Gamma, \Delta_k, \Theta_k, \Lambda, \Pi, \Delta_h, \Theta_h\}$$

and the scalar β.

19.2 A Planning Problem

The first welfare theorem asserts that a competitive equilibrium allocation solves the following planning problem.

Choose $\{c_t, s_t, i_t, h_t, k_t, g_t\}_{t=0}^{\infty}$ to maximize

$$-\frac{1}{2}\mathbb{E}\sum_{t=0}^{\infty}\beta^t[(s_t - b_t)\cdot(s_t - b_t) + g_t \cdot g_t]$$

subject to the linear constraints

$$\Phi_c c_t + \Phi_g g_t + \Phi_i i_t = \Gamma k_{t-1} + d_t$$

$$k_t = \Delta_k k_{t-1} + \Theta_k i_t$$

$$h_t = \Delta_h h_{t-1} + \Theta_h c_t$$

$$s_t = \Lambda h_{t-1} + \Pi c_t$$

and

$$z_{t+1} = A_{22} z_t + C_2 w_{t+1}$$

$$b_t = U_b z_t$$

$$d_t = U_d z_t$$

The DLE class in Python maps this planning problem into a linear-quadratic dynamic programming problem and then solves it by using QuantEcon's LQ class.

(See Section 5.5 of Hansen & Sargent (2013) [HS13] for a full description of how to map these economies into an LQ setting, and how to use the solution to the LQ problem to construct the output matrices in order to simulate the economies)

The state for the LQ problem is

$$x_t = \begin{bmatrix} h_{t-1} \\ k_{t-1} \\ z_t \end{bmatrix}$$

and the control variable is $u_t = i_t$.

Once the LQ problem has been solved, the law of motion for the state is

$$x_{t+1} = (A - BF)x_t + Cw_{t+1}$$

where the optimal control law is $u_t = -Fx_t$.

Letting $A^o = A - BF$ we write this law of motion as

$$x_{t+1} = A^o x_t + Cw_{t+1}$$

19.3 Example Economies

Each of the example economies shown here will share a number of components. In particular, for each we will consider preferences of the form

$$-\frac{1}{2}\mathbb{E}\sum_{t=0}^{\infty}\beta^t[(s_t - b_t)^2 + l_t^2], 0 < \beta < 1$$

$$s_t = \lambda h_{t-1} + \pi c_t$$

$$h_t = \delta_h h_{t-1} + \theta_h c_t$$

$$b_t = U_b z_t$$

Technology of the form

$$c_t + i_t = \gamma_1 k_{t-1} + d_{1t}$$

$$k_t = \delta_k k_{t-1} + i_t$$

$$g_t = \phi_1 i_t \, , \phi_1 > 0$$

$$\begin{bmatrix} d_{1t} \\ 0 \end{bmatrix} = U_d z_t$$

And information of the form

$$z_{t+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} z_t + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} w_{t+1}$$

$$U_b = \begin{bmatrix} 30 & 0 & 0 \end{bmatrix}$$

$$U_d = \begin{bmatrix} 5 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

We shall vary $\{\lambda, \pi, \delta_h, \theta_h, \gamma_1, \delta_k, \phi_1\}$ and the initial state x_0 across the three economies.

19.3.1 Example 1: Hall (1978)

First, we set parameters such that consumption follows a random walk. In particular, we set

$$\lambda = 0, \pi = 1, \gamma_1 = 0.1, \phi_1 = 0.00001, \delta_k = 0.95, \beta = \frac{1}{1.05}$$

(In this economy δ_h and θ_h are arbitrary as household capital does not enter the equation for consumption services We set them to values that will become useful in Example 3)

It is worth noting that this choice of parameter values ensures that $\beta(\gamma_1 + \delta_k) = 1$.

For simulations of this economy, we choose an initial condition of

$$x_0 = \begin{bmatrix} 5 & 150 & 1 & 0 & 0 \end{bmatrix}'$$

```
# Parameter Matrices
γ_1 = 0.1
ϕ_1 = 1e-5

ϕ_c, ϕ_g, ϕ_i, γ, δ_k, θ_k = (np.array([[1], [0]]),
                              np.array([[0], [1]]),
                              np.array([[1], [-ϕ_1]]),
                              np.array([[γ_1], [0]]),
                              np.array([[.95]]),
                              np.array([[1]]))

β, l_λ, π_h, δ_h, θ_h = (np.array([[1 / 1.05]]),
                         np.array([[0]]),
                         np.array([[1]]),
                         np.array([[.9]]),
                         np.array([[1]]) - np.array([[.9]]))

a22, c2, ub, ud = (np.array([[1,   0,   0],
                             [0, 0.8,   0],
                             [0,   0, 0.5]]),
                   np.array([[0, 0],
                             [1, 0],
                             [0, 1]]),
                   np.array([[30, 0, 0]]),
                   np.array([[5, 1, 0],
                             [0, 0, 0]]))

# Initial condition
x0 = np.array([[5], [150], [1], [0], [0]])

info1 = (a22, c2, ub, ud)
tech1 = (ϕ_c, ϕ_g, ϕ_i, γ, δ_k, θ_k)
pref1 = (β, l_λ, π_h, δ_h, θ_h)
```

These parameter values are used to define an economy of the DLE class.

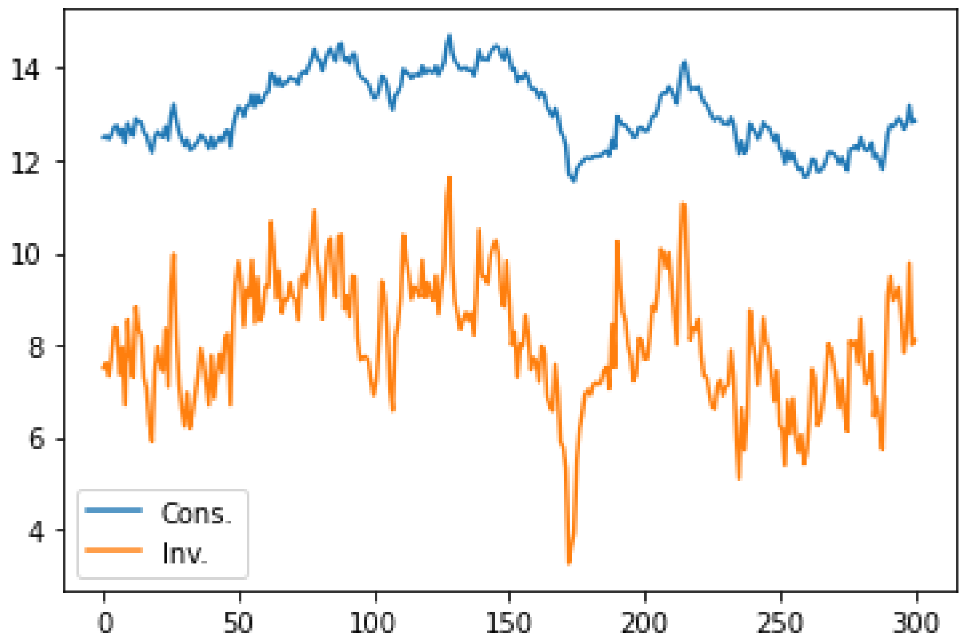
```
econ1 = DLE(info1, tech1, pref1)
```

We can then simulate the economy for a chosen length of time, from our initial state vector x_0

```
econ1.compute_sequence(x0, ts_length=300)
```

The economy stores the simulated values for each variable. Below we plot consumption and investment

```
# This is the right panel of Fig 5.7.1 from p.105 of HS2013
plt.plot(econ1.c[0], label='Cons.')
plt.plot(econ1.i[0], label='Inv.')
plt.legend()
plt.show()
```



Inspection of the plot shows that the sample paths of consumption and investment drift in ways that suggest that each has or nearly has a **random walk** or **unit root** component.

This is confirmed by checking the eigenvalues of A^o

```
econ1.endo, econ1.exo
```

```
(array([0.9, 1. ]), array([1. , 0.8, 0.5]))
```

The endogenous eigenvalue that appears to be unity reflects the random walk character of consumption in Hall's model.

- Actually, the largest endogenous eigenvalue is very slightly below 1.
- This outcome comes from the small adjustment cost ϕ_1.

```
econ1.endo[1]
```

```
0.9999999999904767
```

The fact that the largest endogenous eigenvalue is strictly less than unity in modulus means that it is possible to compute the non-stochastic steady state of consumption, investment and capital.

```
econ1.compute_steadystate()
np.set_printoptions(precision=3, suppress=True)
print(econ1.css, econ1.iss, econ1.kss)
```

```
[[4.999]] [[-0.001]] [[-0.021]]
```

However, the near-unity endogenous eigenvalue means that these steady state values are of little relevance.

19.3.2 Example 2: Altered Growth Condition

We generate our next economy by making two alterations to the parameters of Example 1.

- First, we raise ϕ_1 from 0.00001 to 1.
  - This will lower the endogenous eigenvalue that is close to 1, causing the economy to head more quickly to the vicinity of its non-stochastic steady-state.
- Second, we raise γ_1 from 0.1 to 0.15.
  - This has the effect of raising the optimal steady-state value of capital.

We also start the economy off from an initial condition with a lower capital stock

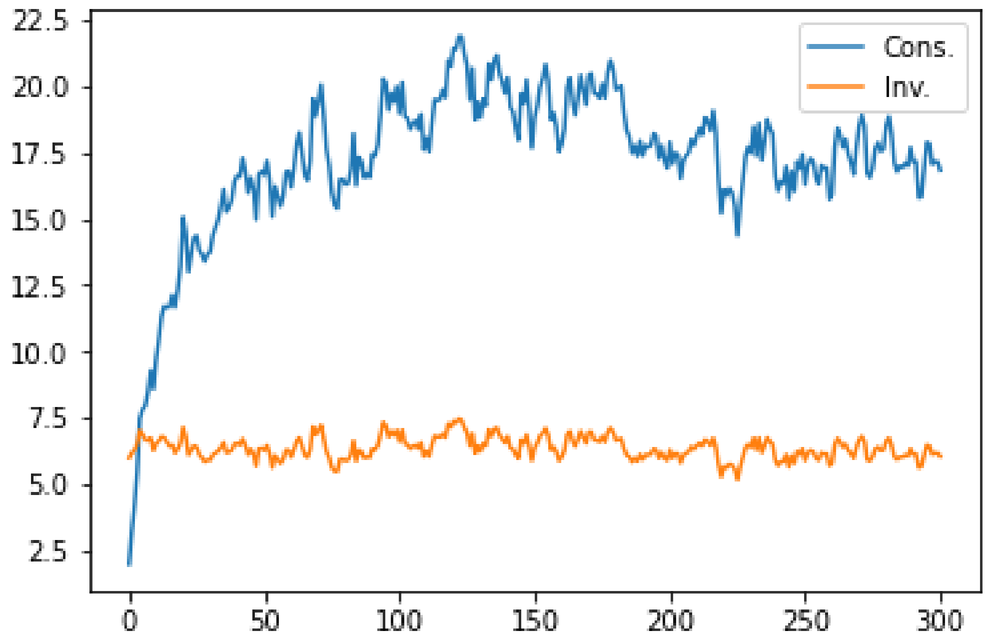
$$x_0 = \begin{bmatrix} 5 & 20 & 1 & 0 & 0 \end{bmatrix}'$$

Therefore, we need to define the following new parameters

```
γ2 = 0.15
γ22 = np.array([[γ2], [0]])

ϕ_12 = 1
ϕ_i2 = np.array([[1], [-ϕ_12]])

tech2 = (ϕ_c, ϕ_g, ϕ_i2, γ22, δ_k, θ_k)

x02 = np.array([[5], [20], [1], [0], [0]])
```

Creating the DLE class and then simulating gives the following plot for consumption and investment

```
econ2 = DLE(info1, tech2, pref1)

econ2.compute_sequence(x02, ts_length=300)

plt.plot(econ2.c[0], label='Cons.')
plt.plot(econ2.i[0], label='Inv.')
plt.legend()
plt.show()
```



Simulating our new economy shows that consumption grows quickly in the early stages of the sample.

However, it then settles down around the new non-stochastic steady-state level of consumption of 17.5, which we find as follows

```
econ2.compute_steadystate()
print(econ2.css, econ2.iss, econ2.kss)
```

```
[[17.5]] [[6.25]] [[125.]]
```

The economy converges faster to this level than in Example 1 because the largest endogenous eigenvalue of A^o is now significantly lower than 1.

```
econ2.endo, econ2.exo
```

```
(array([0.9  , 0.952]), array([1. , 0.8, 0.5]))
```

19.3.3 Example 3: A Jones-Manuelli (1990) Economy

For our third economy, we choose parameter values with the aim of generating *sustained* growth in consumption, investment and capital.

To do this, we set parameters so that Jones and Manuelli's "growth condition" is just satisfied.

In our notation, just satisfying the growth condition is actually equivalent to setting $\beta(\gamma_1 + \delta_k) = 1$, the condition that was necessary for consumption to be a random walk in Hall's model.

Thus, we lower γ_1 back to 0.1.

In our model, this is a necessary but not sufficient condition for growth.

To generate growth we set preference parameters to reflect habit persistence.

In particular, we set $\lambda = -1$, $\delta_h = 0.9$ and $\theta_h = 1 - \delta_h = 0.1$.

This makes preferences assume the form

$$-\frac{1}{2}\mathbb{E}\sum_{t=0}^{\infty}\beta^t[(c_t - b_t - (1-\delta_h)\sum_{j=0}^{\infty}\delta_h^j c_{t-j-1})^2 + l_t^2]$$

These preferences reflect habit persistence

- the effective "bliss point" $b_t + (1-\delta_h)\sum_{j=0}^{\infty}\delta_h^j c_{t-j-1}$ now shifts in response to a moving average of past consumption

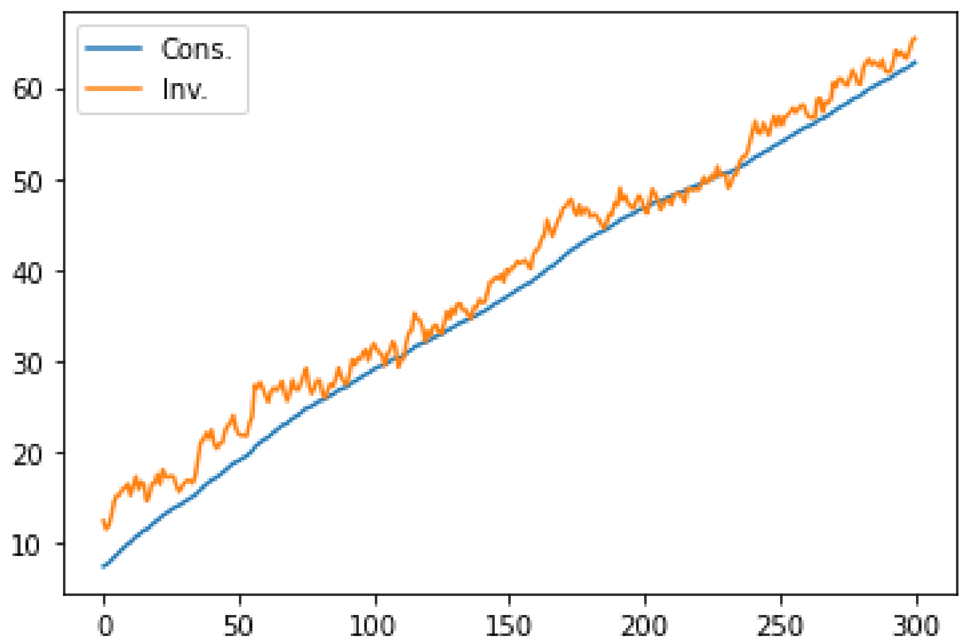
Since δ_h and θ_h were defined earlier, the only change we need to make from the parameters of Example 1 is to define the new value of λ.

```
l_λ2 = np.array([[-1]])
pref2 = (β, l_λ2, π_h, δ_h, θ_h)
```

```
econ3 = DLE(info1, tech1, pref2)
```

We simulate this economy from the original state vector

```
econ3.compute_sequence(x0, ts_length=300)

# This is the right panel of Fig 5.10.1 from p.110 of HS2013
plt.plot(econ3.c[0], label='Cons.')
plt.plot(econ3.i[0], label='Inv.')
plt.legend()
plt.show()
```



Thus, adding habit persistence to the Hall model of Example 1 is enough to generate sustained growth in our economy.

The eigenvalues of A^o in this new economy are

```
econ3.endo, econ3.exo
```

```
(array([1.+0.j, 1.-0.j]), array([1. , 0.8, 0.5]))
```

We now have two unit endogenous eigenvalues. One stems from satisfying the growth condition (as in Example 1).

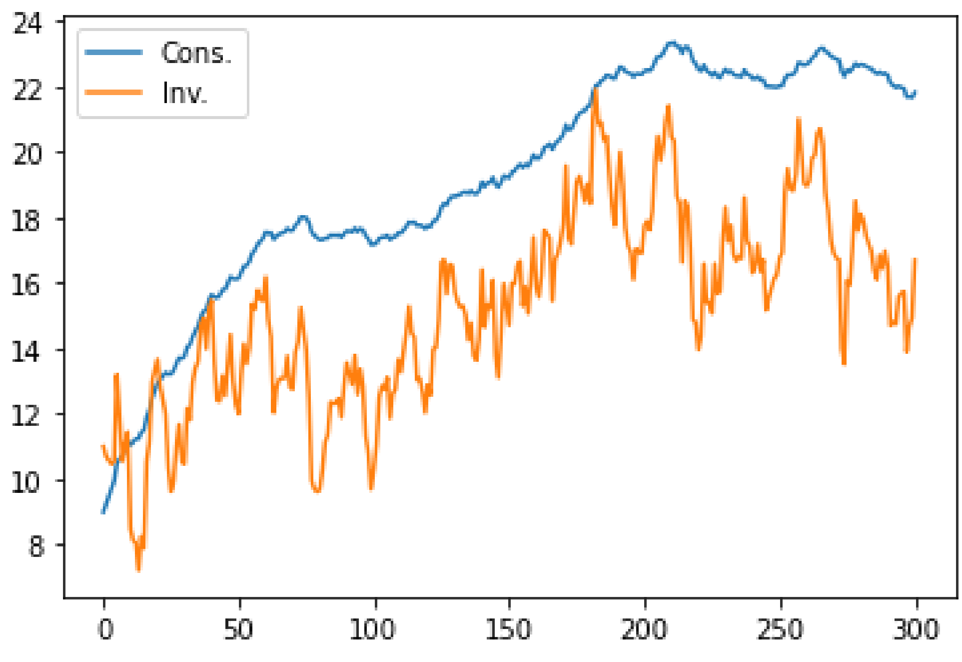
The other unit eigenvalue results from setting $\lambda = -1$.

To show the importance of both of these for generating growth, we consider the following experiments.

19.3.4 Example 3.1: Varying Sensitivity

Next we raise λ to -0.7

```
l_λ3 = np.array([[-0.7]])
pref3 = (β, l_λ3, π_h, δ_h, θ_h)

econ4 = DLE(info1, tech1, pref3)

econ4.compute_sequence(x0, ts_length=300)

plt.plot(econ4.c[0], label='Cons.')
plt.plot(econ4.i[0], label='Inv.')
plt.legend()
plt.show()
```



We no longer achieve sustained growth if λ is raised from -1 to -0.7.

This is related to the fact that one of the endogenous eigenvalues is now less than 1.

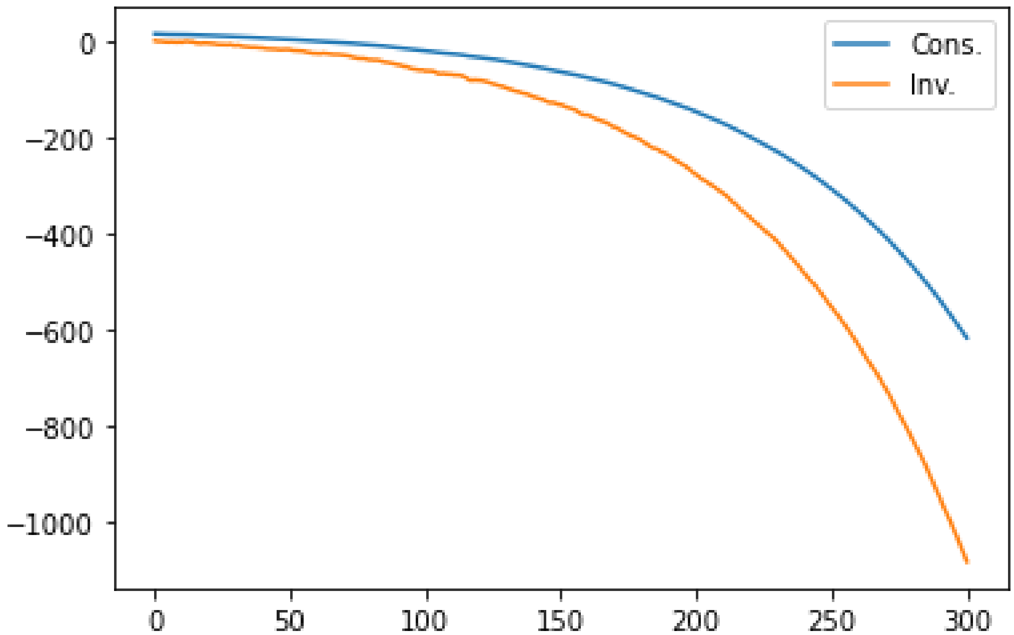
```
econ4.endo, econ4.exo
```

```
(array([0.97, 1.  ]), array([1. , 0.8, 0.5]))
```

19.3.5 Example 3.2: More Impatience

Next let's lower β to 0.94

```
β_2 = np.array([[0.94]])
pref4 = (β_2, l_λ, π_h, δ_h, θ_h)

econ5 = DLE(info1, tech1, pref4)

econ5.compute_sequence(x0, ts_length=300)

plt.plot(econ5.c[0], label='Cons.')
plt.plot(econ5.i[0], label='Inv.')
plt.legend()
plt.show()
```



Growth also fails if we lower β, since we now have $\beta(\gamma_1 + \delta_k) < 1$.

Consumption and investment explode downwards, as a lower value of β causes the representative consumer to front-load consumption.

This explosive path shows up in the second endogenous eigenvalue now being larger than one.

```
econ5.endo, econ5.exo
```

```
(array([0.9  , 1.013]), array([1. , 0.8, 0.5]))
```

CHAPTER TWENTY

LUCAS ASSET PRICING USING DLE

Contents

This is one of a suite of lectures that use the quantecon DLE class to instantiate models within the [HS13] class of models described in detail in *Recursive Models of Dynamic Linear Economies*.

In addition to what's in Anaconda, this lecture uses the quantecon library

```
!pip install --upgrade quantecon
```

This lecture uses the DLE class to price payout streams that are linear functions of the economy's state vector, as well as risk-free assets that pay out one unit of the first consumption good with certainty.

We assume basic knowledge of the class of economic environments that fall within the domain of the DLE class.

Many details about the basic environment are contained in the lecture *Growth in Dynamic Linear Economies*.

We'll also need the following imports

```
import numpy as np
import matplotlib.pyplot as plt
from quantecon import LQ
from quantecon import DLE
%matplotlib inline
```

We use a linear-quadratic version of an economy that Lucas (1978) [Luc78] used to develop an equilibrium theory of asset prices:

Preferences

$$-\frac{1}{2}\mathbb{E}\sum_{t=0}^{\infty}\beta^t[(c_t-b_t)^2+l_t^2]|J_0$$

$$s_t = c_t$$

$$b_t = U_b z_t$$

Technology

$$c_t = d_{1t}$$

$$k_t = \delta_k k_{t-1} + i_t$$

$$g_t = \phi_1 i_t \,, \phi_1 > 0$$

$$\begin{bmatrix} d_{1t} \\ 0 \end{bmatrix} = U_d z_t$$

Information

$$z_{t+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} z_t + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} w_{t+1}$$

$$U_b = \begin{bmatrix} 30 & 0 & 0 \end{bmatrix}$$

$$U_d = \begin{bmatrix} 5 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$x_0 = \begin{bmatrix} 5 & 150 & 1 & 0 & 0 \end{bmatrix}'$$

20.1 Asset Pricing Equations

[HS13] show that the time t value of a permanent claim to a stream $y_s = U_a x_s \,, s \geq t$ is:

$$a_t = (x_t' \mu_a x_t + \sigma_a)/(\bar{e}_1 M_c x_t)$$

with

$$\mu_a = \sum_{\tau=0}^{\infty} \beta^\tau (A^{o'})^\tau Z_a A^{o\tau}$$

$$\sigma_a = \frac{\beta}{1-\beta} \text{trace}(Z_a \sum_{\tau=0}^{\infty} \beta^\tau (A^o)^\tau C C' (A^{o'})^\tau)$$

where

$$Z_a = U_a' M_c$$

The use of $\bar{e}_1$ indicates that the first consumption good is the numeraire.

20.2 Asset Pricing Simulations

```
gam = 0
γ = np.array([[gam], [0]])
ϕ_c = np.array([[1], [0]])
ϕ_g = np.array([[0], [1]])
ϕ_1 = 1e-4
ϕ_i = np.array([[0], [-ϕ_1]])
δ_k = np.array([[.95]])
θ_k = np.array([[1]])
β = np.array([[1 / 1.05]])
ud = np.array([[5, 1, 0],
               [0, 0, 0]])
a22 = np.array([[1,   0,    0],
```

(continues on next page)

(continued from previous page)

```
                 [0, 0.8,    0],
                 [0,   0, 0.5]])
c2 = np.array([[0, 1, 0],
               [0, 0, 1]]).T
l_λ = np.array([[0]])
π_h = np.array([[1]])
δ_h = np.array([[.9]])
θ_h = np.array([[1]]) - δ_h
ub = np.array([[30, 0, 0]])
x0 = np.array([[5, 150, 1, 0, 0]]).T

info1 = (a22, c2, ub, ud)
tech1 = (ϕ_c, ϕ_g, ϕ_i, γ, δ_k, θ_k)
pref1 = (β, l_λ, π_h, δ_h, θ_h)
```

```
econ1 = DLE(info1, tech1, pref1)
```

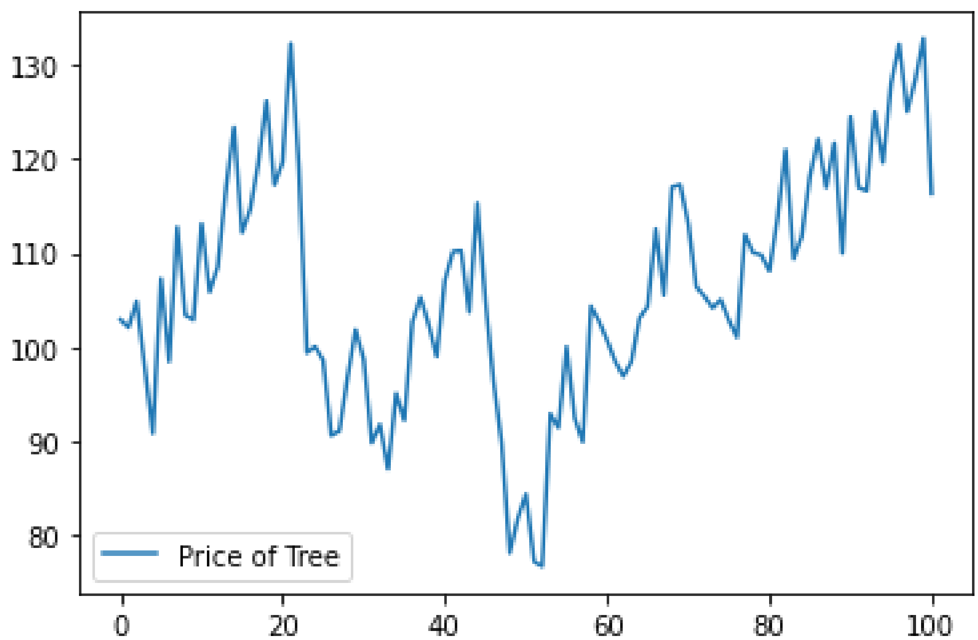
After specifying a "Pay" matrix, we simulate the economy.

The particular choice of "Pay" used below means that we are pricing a perpetual claim on the endowment process d_{1t}

```
econ1.compute_sequence(x0, ts_length=100, Pay=np.array([econ1.Sd[0, :]]))
```

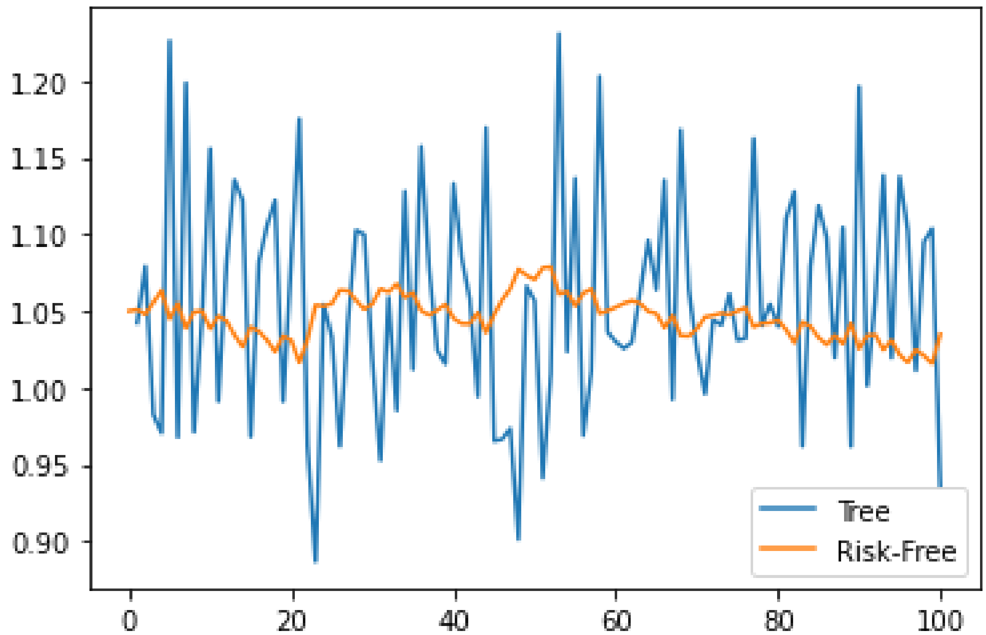
The graph below plots the price of this claim over time:

```
### Fig 7.12.1 from p.147 of HS2013
plt.plot(econ1.Pay_Price, label='Price of Tree')
plt.legend()
plt.show()
```



The next plot displays the realized gross rate of return on this "Lucas tree" as well as on a risk-free one-period bond:

```
### Left panel of Fig 7.12.2 from p.148 of HS2013
plt.plot(econ1.Pay_Gross, label='Tree')
plt.plot(econ1.R1_Gross, label='Risk-Free')
plt.legend()
plt.show()
```



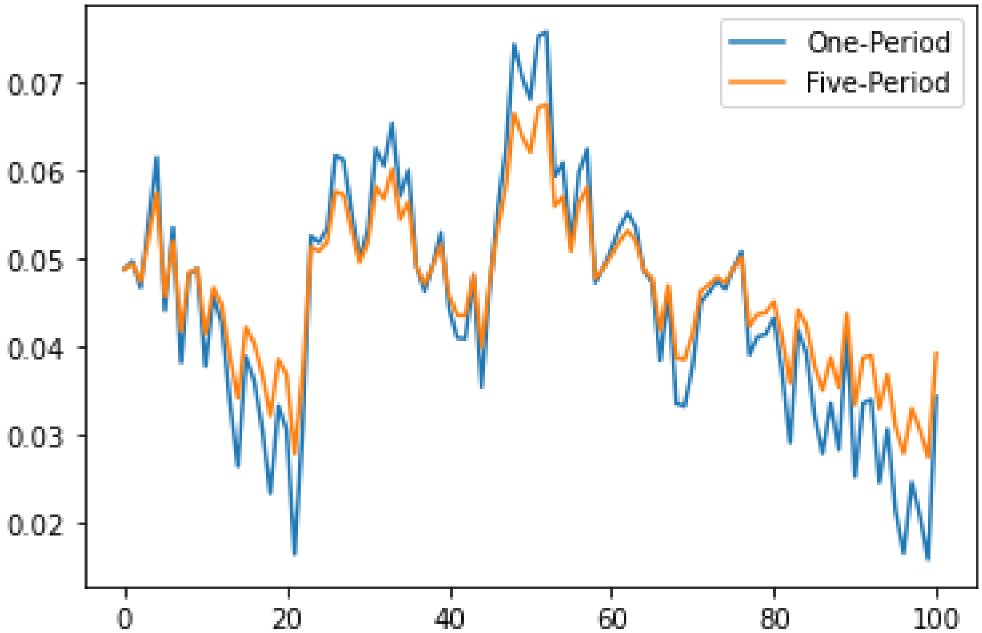

```
np.corrcoef(econ1.Pay_Gross[1:, 0], econ1.R1_Gross[1:, 0])
```

```
array([[ 1.        , -0.43550535],
       [-0.43550535,  1.        ]])
```

Above we have also calculated the correlation coefficient between these two returns.

To give an idea of how the term structure of interest rates moves in this economy, the next plot displays the *net* rates of return on one-period and five-period risk-free bonds:

```
### Right panel of Fig 7.12.2 from p.148 of HS2013
plt.plot(econ1.R1_Net, label='One-Period')
plt.plot(econ1.R5_Net, label='Five-Period')
plt.legend()
plt.show()
```



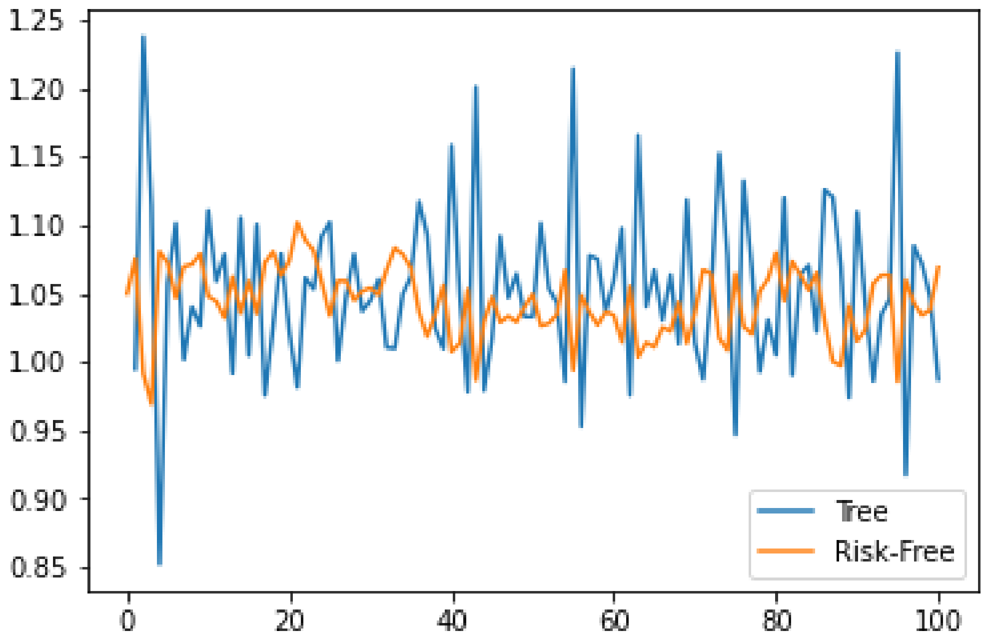
From the above plot, we can see the tendency of the term structure to slope up when rates are low and to slope down when rates are high.

Comparing it to the previous plot of the price of the "Lucas tree", we can also see that net rates of return are low when the price of the tree is high, and vice versa.

We now plot the realized gross rate of return on a "Lucas tree" as well as on a risk-free one-period bond when the autoregressive parameter for the endowment process is reduced to 0.4:

```
a22_2 = np.array([[1,   0,   0],
                  [0, 0.4,   0],
                  [0,   0, 0.5]])
info2 = (a22_2, c2, ub, ud)

econ2 = DLE(info2, tech1, pref1)
econ2.compute_sequence(x0, ts_length=100, Pay=np.array([econ2.Sd[0, :]]))
```

```
### Left panel of Fig 7.12.3 from p.148 of HS2013
plt.plot(econ2.Pay_Gross, label='Tree')
plt.plot(econ2.R1_Gross, label='Risk-Free')
plt.legend()
plt.show()
```



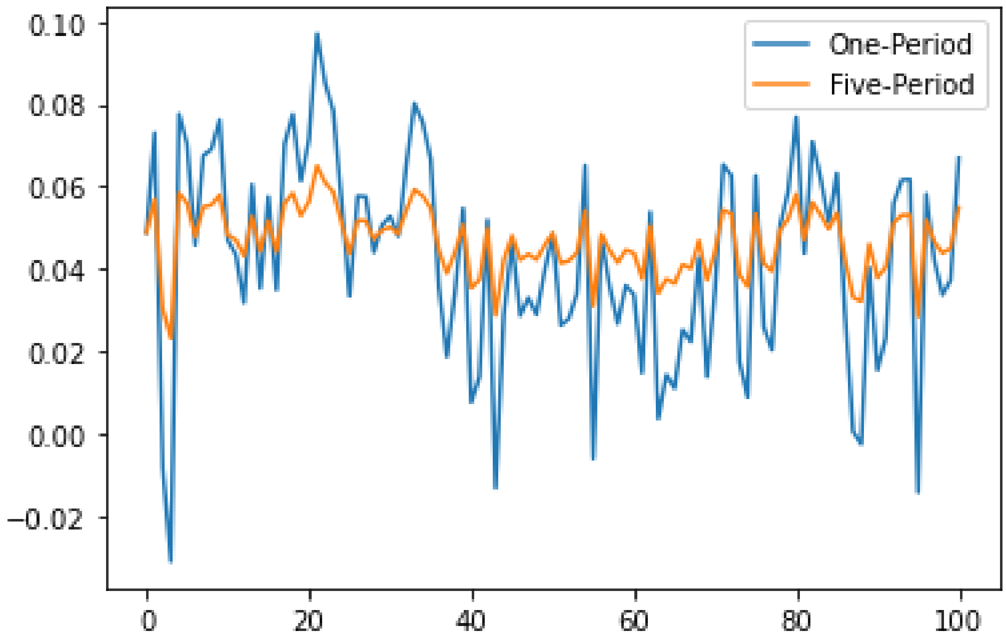

```
np.corrcoef(econ2.Pay_Gross[1:, 0], econ2.R1_Gross[1:, 0])
```

```
array([[ 1.        , -0.67932033],
       [-0.67932033,  1.        ]])
```

The correlation between these two gross rates is now more negative.

Next, we again plot the *net* rates of return on one-period and five-period risk-free bonds:

```
### Right panel of Fig 7.12.3 from p.148 of HS2013
plt.plot(econ2.R1_Net, label='One-Period')
plt.plot(econ2.R5_Net, label='Five-Period')
plt.legend()
plt.show()
```



We can see the tendency of the term structure to slope up when rates are low (and down when rates are high) has been accentuated relative to the first instance of our economy.

CHAPTER

TWENTYONE

IRFS IN HALL MODELS

Contents

This is another member of a suite of lectures that use the quantecon DLE class to instantiate models within the [HS13] class of models described in detail in *Recursive Models of Dynamic Linear Economies*.

In addition to what's in Anaconda, this lecture uses the quantecon library.

```
!pip install --upgrade quantecon
```

We'll make these imports:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from quantecon import LQ
from quantecon import DLE
```

This lecture shows how the DLE class can be used to create impulse response functions for three related economies, starting from Hall (1978) [Hal78].

Knowledge of the basic economic environment is assumed.

See the lecture "Growth in Dynamic Linear Economies" for more details.

21.1 Example 1: Hall (1978)

First, we set parameters to make consumption (almost) follow a random walk.

We set

$$\lambda = 0, \pi = 1, \gamma_1 = 0.1, \phi_1 = 0.00001, \delta_k = 0.95, \beta = \frac{1}{1.05}$$

(In this example δ_h and θ_h are arbitrary as household capital does not enter the equation for consumption services.

We set them to values that will become useful in Example 3)

It is worth noting that this choice of parameter values ensures that $\beta(\gamma_1 + \delta_k) = 1$.

For simulations of this economy, we choose an initial condition of:

$$x_0 = \begin{bmatrix} 5 & 150 & 1 & 0 & 0 \end{bmatrix}'$$

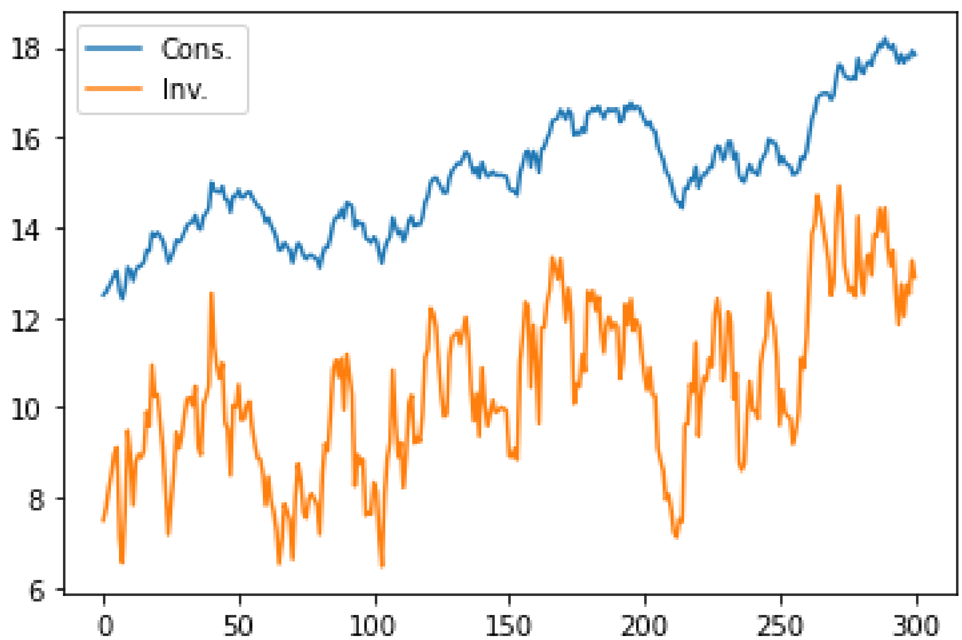
```
γ_1 = 0.1
γ = np.array([[γ_1], [0]])
ϕ_c = np.array([[1], [0]])
ϕ_g = np.array([[0], [1]])
ϕ_1 = 1e-5
ϕ_i = np.array([[1], [-ϕ_1]])
δ_k = np.array([[.95]])
θ_k = np.array([[1]])
β = np.array([[1 / 1.05]])
l_λ = np.array([[0]])
π_h = np.array([[1]])
δ_h = np.array([[.9]])
θ_h = np.array([[1]])
a22 = np.array([[1,   0,   0],
                [0, 0.8,   0],
                [0,   0, 0.5]])
c2 = np.array([[0, 0],
               [1, 0],
               [0, 1]])
ud = np.array([[5, 1, 0],
               [0, 0, 0]])
ub = np.array([[30, 0, 0]])
x0 = np.array([[5], [150], [1], [0], [0]])

info1 = (a22, c2, ub, ud)
tech1 = (ϕ_c, ϕ_g, ϕ_i, γ, δ_k, θ_k)
pref1 = (β, l_λ, π_h, δ_h, θ_h)
```

These parameter values are used to define an economy of the DLE class.

We can then simulate the economy for a chosen length of time, from our initial state vector x_0.

The economy stores the simulated values for each variable. Below we plot consumption and investment:

```
econ1 = DLE(info1, tech1, pref1)
econ1.compute_sequence(x0, ts_length=300)

# This is the right panel of Fig 5.7.1 from p.105 of HS2013
plt.plot(econ1.c[0], label='Cons.')
plt.plot(econ1.i[0], label='Inv.')
plt.legend()
plt.show()
```

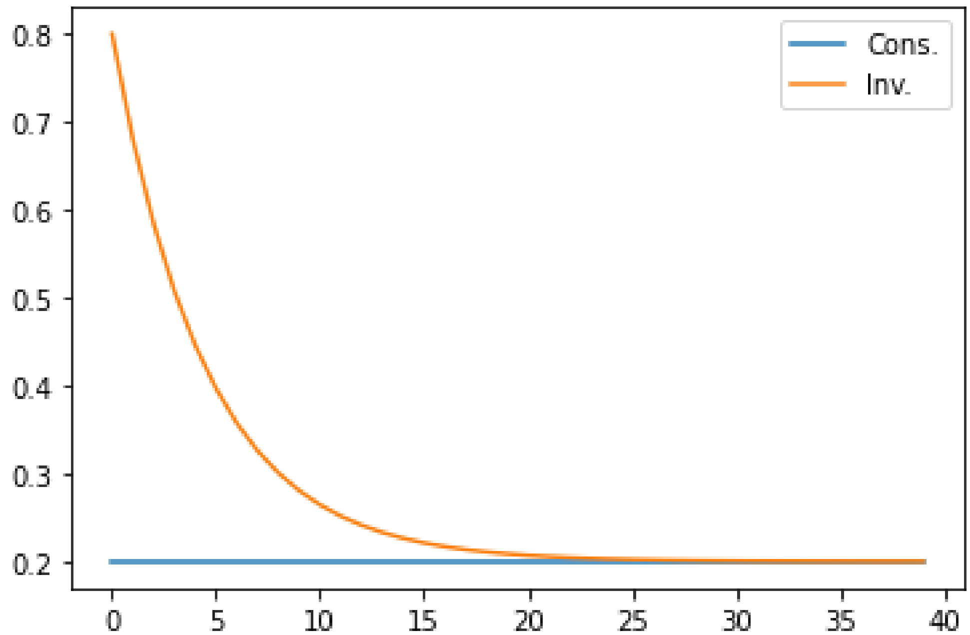


The DLE class can be used to create impulse response functions for each of the endogenous variables: $\{c_t, s_t, h_t, i_t, k_t, g_t\}$.

If no selector vector for the shock is specified, the default choice is to give IRFs to the first shock in w_{t+1}.

Below we plot the impulse response functions of investment and consumption to an endowment innovation (the first shock) in the Hall model:

```
econ1.irf(ts_length=40, shock=None)
# This is the left panel of Fig 5.7.1 from p.105 of HS2013
plt.plot(econ1.c_irf, label='Cons.')
plt.plot(econ1.i_irf, label='Inv.')
plt.legend()
plt.show()
```



It can be seen that the endowment shock has permanent effects on the level of both consumption and investment, consistent with the endogenous unit eigenvalue in this economy.

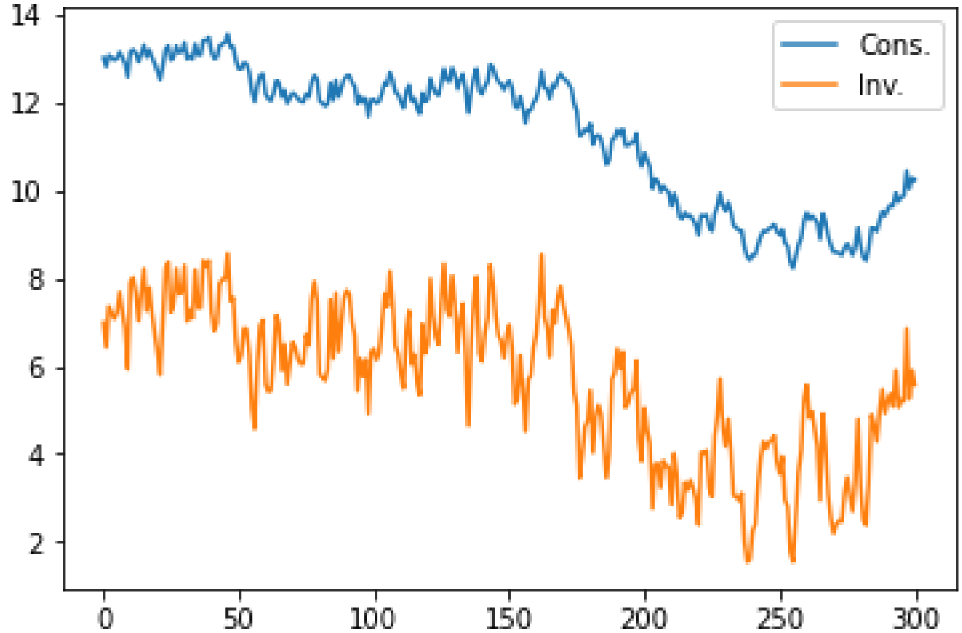
Investment is much more responsive to the endowment shock at shorter time horizons.

21.2 Example 2: Higher Adjustment Costs

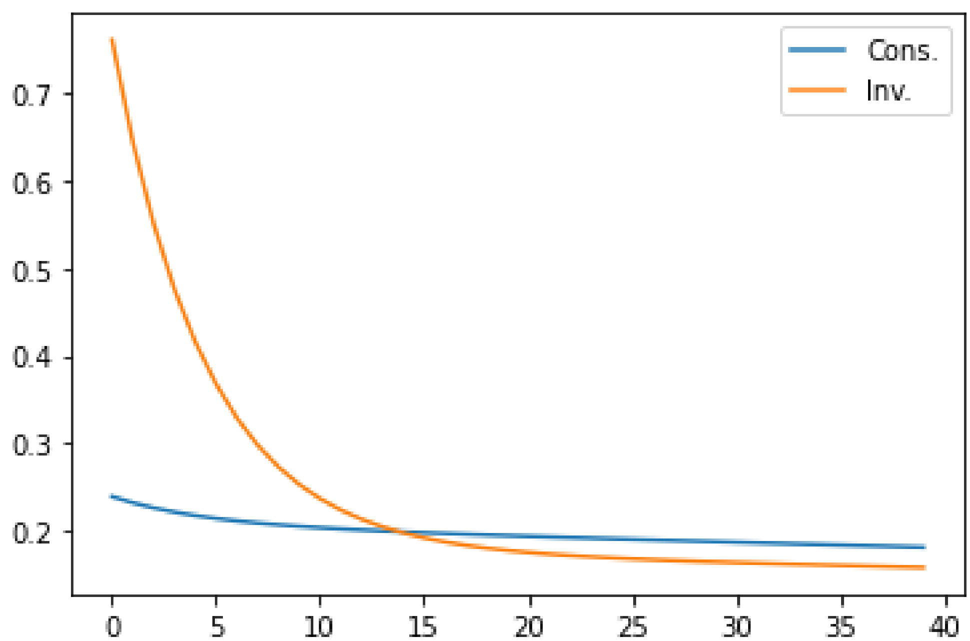
We generate our next economy by making only one change to the parameters of Example 1: we raise the parameter associated with the cost of adjusting capital,ϕ_1, from 0.00001 to 0.2.

This will lower the endogenous eigenvalue that is unity in Example 1 to a value slightly below 1.

```
ϕ_12 = 0.2
ϕ_i2 = np.array([[1], [-ϕ_12]])
tech2 = (ϕ_c, ϕ_g, ϕ_i2, γ, δ_k, θ_k)

econ2 = DLE(info1, tech2, pref1)
econ2.compute_sequence(x0, ts_length = 300)

# This is the right panel of Fig 5.8.1 from p.106 of HS2013
plt.plot(econ2.c[0], label='Cons.')
plt.plot(econ2.i[0], label='Inv.')
plt.legend()
plt.show()
```




```
econ2.irf(ts_length=40,shock=None)
# This is the left panel of Fig 5.8.1 from p.106 of HS2013
plt.plot(econ2.c_irf,label='Cons.')
plt.plot(econ2.i_irf,label='Inv.')
plt.legend()
plt.show()
```




```
econ2.endo
```

```
array([0.9        , 0.99657126])
```

```
econ2.compute_steadystate()
print(econ2.css, econ2.iss, econ2.kss)
```

```
[[5.]] [[2.12173041e-12]] [[4.2434517e-11]]
```

The first graph shows that there seems to be a downward trend in both consumption and investment.

This is a consequence of the decrease in the largest endogenous eigenvalue from unity in the earlier economy, caused by the higher adjustment cost.

The present economy has a nonstochastic steady state value of 5 for consumption and 0 for both capital and investment.

Because the largest endogenous eigenvalue is still close to 1, the economy heads only slowly towards these mean values.

The impulse response functions now show that an endowment shock does not have a permanent effect on the levels of either consumption or investment.

21.3 Example 3: Durable Consumption Goods

We generate our third economy by raising ϕ_1 further, to 1.0. We also raise the production function parameter from 0.1 to 0.15 (which raises the non-stochastic steady state value of capital above zero).

We also change the specification of preferences to make the consumption good *durable*.

Specifically, we allow for a single durable household good obeying:

$$h_t = \delta_h h_{t-1} + c_t \, , 0 < \delta_h < 1$$

Services are related to the stock of durables at the beginning of the period:

$$s_t = \lambda h_{t-1} \, , \lambda > 0$$

And preferences are ordered by:

$$-\frac{1}{2} \mathbb{E} \sum_{t=0}^{\infty} \beta^t [(\lambda h_{t-1} - b_t)^2 + l_t^2] | J_0$$

To implement this, we set $\lambda = 0.1$ and $\pi = 0$ (we have already set $\theta_h = 1$ and $\delta_h = 0.9$).

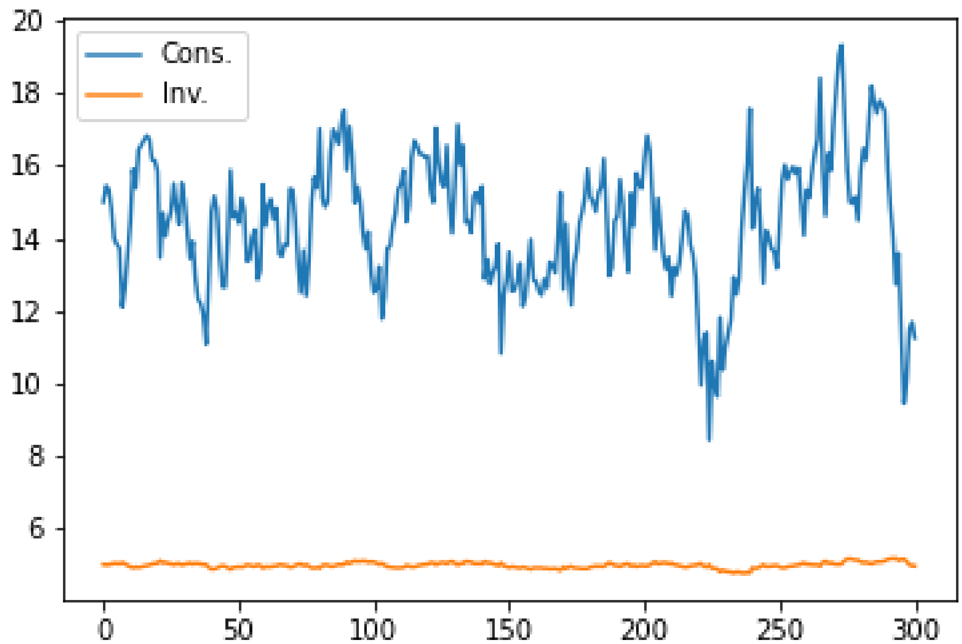
We start from an initial condition that makes consumption begin near around its non-stochastic steady state.

```
ϕ_13 = 1
ϕ_i3 = np.array([[1], [-ϕ_13]])

γ_12 = 0.15
γ_2 = np.array([[γ_12], [0]])

l_λ2 = np.array([[0.1]])
π_h2 = np.array([[0]])

x01 = np.array([[150], [100], [1], [0], [0]])

tech3 = (ϕ_c, ϕ_g, ϕ_i3, γ_2, δ_k, θ_k)
pref2 = (β, l_λ2, π_h2, δ_h, θ_h)

econ3 = DLE(info1, tech3, pref2)
```

(continues on next page)

(continued from previous page)

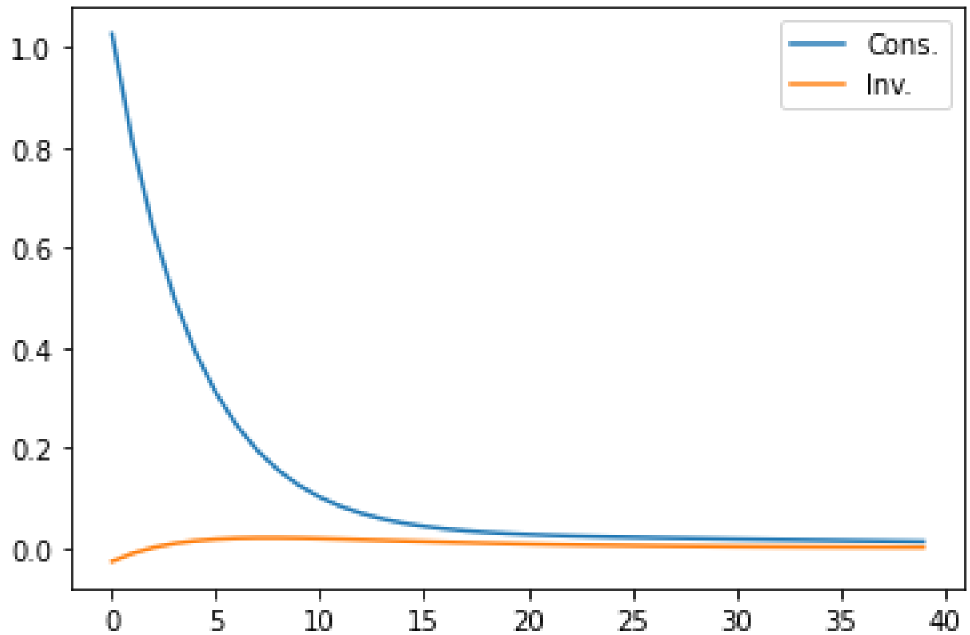
```
econ3.compute_sequence(x01, ts_length=300)

# This is the right panel of Fig 5.11.1 from p.111 of HS2013
plt.plot(econ3.c[0], label='Cons.')
plt.plot(econ3.i[0], label='Inv.')
plt.legend()
plt.show()
```



In contrast to Hall's original model of Example 1, it is now investment that is much smoother than consumption.

This illustrates how making consumption goods durable tends to undo the strong consumption smoothing result that Hall obtained.

```
econ3.irf(ts_length=40, shock=None)
# This is the left panel of Fig 5.11.1 from p.111 of HS2013
plt.plot(econ3.c_irf, label='Cons.')
plt.plot(econ3.i_irf, label='Inv.')
plt.legend()
plt.show()
```



The impulse response functions confirm that consumption is now much more responsive to an endowment shock (and investment less so) than in Example 1.

As in Example 2, the endowment shock has permanent effects on neither variable.

CHAPTER

TWENTYTWO

PERMANENT INCOME MODEL USING THE DLE CLASS

Contents

This lecture is part of a suite of lectures that use the quantecon DLE class to instantiate models within the [HS13] class of models described in detail in *Recursive Models of Dynamic Linear Economies*.

In addition to what's included in Anaconda, this lecture uses the quantecon library.

```
!pip install --upgrade quantecon
```

This lecture adds a third solution method for the linear-quadratic-Gaussian permanent income model with $\beta R = 1$, complementing the other two solution methods described in Optimal Savings I: The Permanent Income Model and Optimal Savings II: LQ Techniques and this Jupyter notebook.

The additional solution method uses the **DLE** class.

In this way, we map the permanent income model into the framework of Hansen & Sargent (2013) "Recursive Models of Dynamic Linear Economies" [HS13].

We'll also require the following imports

```
import quantecon as qe
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt
%matplotlib inline
from quantecon import DLE

np.set_printoptions(suppress=True, precision=4)
```

22.1 The Permanent Income Model

The LQ permanent income model is an example of a **savings problem**.

A consumer has preferences over consumption streams that are ordered by the utility functional

$$E_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \tag{1}$$

where E_t is the mathematical expectation conditioned on the consumer's time t information, c_t is time t consumption, $u(c)$ is a strictly concave one-period utility function, and $\beta \in (0,1)$ is a discount factor.

The LQ model gets its name partly from assuming that the utility function u is quadratic:

$$u(c) = -.5(c - \gamma)^2$$

where $\gamma > 0$ is a bliss level of consumption.

The consumer maximizes the utility functional (1) by choosing a consumption, borrowing plan $\{c_t, b_{t+1}\}_{t=0}^{\infty}$ subject to the sequence of budget constraints

$$c_t + b_t = R^{-1} b_{t+1} + y_t, t \geq 0 \tag{2}$$

where y_t is an exogenous stationary endowment process, R is a constant gross risk-free interest rate, b_t is one-period risk-free debt maturing at t, and b_0 is a given initial condition.

We shall assume that $R^{-1} = \beta$.

Equation (2) is linear.

We use another set of linear equations to model the endowment process.

In particular, we assume that the endowment process has the state-space representation

$$\begin{aligned} z_{t+1} &= A_{22} z_t + C_2 w_{t+1} \\ y_t &= U_y z_t \end{aligned} \tag{3}$$

where w_{t+1} is an IID process with mean zero and identity contemporaneous covariance matrix, A_{22} is a stable matrix, its eigenvalues being strictly below unity in modulus, and U_y is a selection vector that identifies y with a particular linear combination of the z_t.

We impose the following condition on the consumption, borrowing plan:

$$E_0 \sum_{t=0}^{\infty} \beta^t b_t^2 < +\infty \tag{4}$$

This condition suffices to rule out Ponzi schemes.

(We impose this condition to rule out a borrow-more-and-more plan that would allow the household to enjoy bliss consumption forever)

The state vector confronting the household at t is

$$x_t = \begin{bmatrix} z_t \\ b_t \end{bmatrix}$$

where b_t is its one-period debt falling due at the beginning of period t and z_t contains all variables useful for forecasting its future endowment.

We assume that $\{y_t\}$ follows a second order univariate autoregressive process:

$$y_{t+1} = \alpha + \rho_1 y_t + \rho_2 y_{t-1} + \sigma w_{t+1}$$

22.1.1 Solution with the DLE Class

One way of solving this model is to map the problem into the framework outlined in Section 4.8 of [HS13] by setting up our technology, information and preference matrices as follows:

Technology: $\phi_c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\phi_g = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\phi_i = \begin{bmatrix} -1 \\ -0.00001 \end{bmatrix}$, $\Gamma = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$, $\Delta_k = 0,\ \Theta_k = R$.

Information: $A_{22} = \begin{bmatrix} 1 & 0 & 0 \\ \alpha & \rho_1 & \rho_2 \\ 0 & 1 & 0 \end{bmatrix}$, $C_2 = \begin{bmatrix} 0 \\ \sigma \\ 0 \end{bmatrix}$, $U_b = \begin{bmatrix} \gamma & 0 & 0 \end{bmatrix}$, $U_d = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

Preferences: $\Lambda = 0, \Pi = 1, \Delta_h = 0, \Theta_h = 0$.

We set parameters

$\alpha = 10, \beta = 0.95, \rho_1 = 0.9, \rho_2 = 0, \sigma = 1$

(The value of γ does not affect the optimal decision rule)

The chosen matrices mean that the household's technology is:

$$c_t + k_{t-1} = i_t + y_t$$

$$\frac{k_t}{R} = i_t$$

$$l_t^2 = (0.00001)^2 i_t$$

Combining the first two of these gives the budget constraint of the permanent income model, where $k_t = b_{t+1}$.

The third equation is a very small penalty on debt-accumulation to rule out Ponzi schemes.

We set up this instance of the DLE class below:

```
α, β, ρ_1, ρ_2, σ = 10, 0.95, 0.9, 0, 1

γ = np.array([[-1], [0]])
ϕ_c = np.array([[1], [0]])
ϕ_g = np.array([[0], [1]])
ϕ_1 = 1e-5
ϕ_i = np.array([[-1], [-ϕ_1]])
δ_k = np.array([[0]])
θ_k = np.array([[1 / β]])
β = np.array([[β]])
l_λ = np.array([[0]])
π_h = np.array([[1]])
δ_h = np.array([[0]])
θ_h = np.array([[0]])

a22 = np.array([[1,   0,   0],
                [α, ρ_1, ρ_2],
                [0, 1, 0]])

c2 = np.array([[0], [σ], [0]])
ud = np.array([[0, 1, 0],
               [0, 0, 0]])
ub = np.array([[100, 0, 0]])

x0 = np.array([[0], [0], [1], [0], [0]])
```

(continues on next page)

(continued from previous page)

```
info1 = (a22, c2, ub, ud)
tech1 = (ϕ_c, ϕ_g, ϕ_i, γ, δ_k, θ_k)
pref1 = (β, l_λ, π_h, δ_h, θ_h)
econ1 = DLE(info1, tech1, pref1)
```

To check the solution of this model with that from the **LQ** problem, we select the S_c matrix from the DLE class.

The solution to the DLE economy has:

$$c_t = S_c x_t$$

```
econ1.Sc
```

```
array([[ 0.     , -0.05   , 65.5172,  0.3448,  0.     ]])
```

The state vector in the DLE class is:

$$x_t = \begin{bmatrix} h_{t-1} \\ k_{t-1} \\ z_t \end{bmatrix}$$

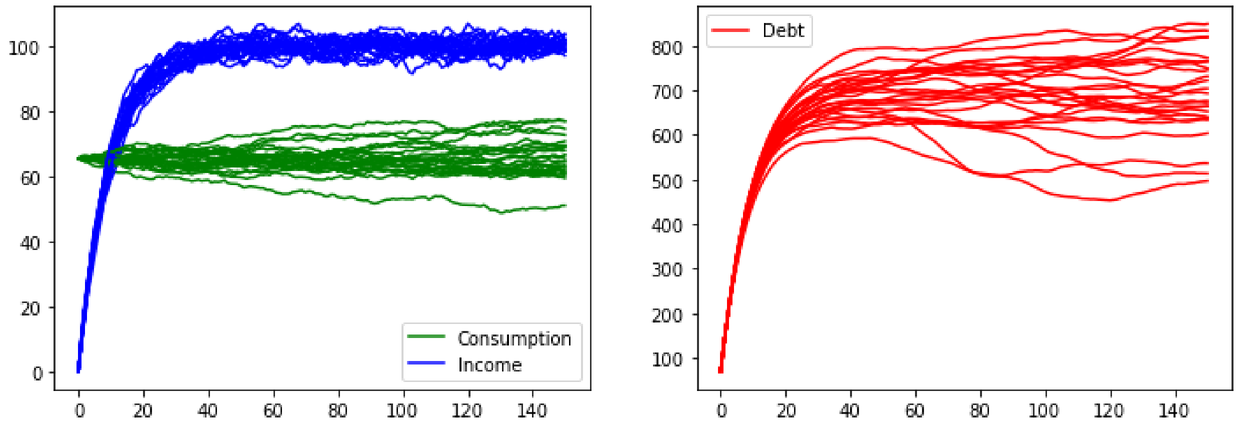
where $k_{t-1} = b_t$ is set up to be b_t in the permanent income model.

The state vector in the LQ problem is $\begin{bmatrix} z_t \\ b_t \end{bmatrix}$.

Consequently, the relevant elements of econ1.Sc are the same as in $-F$ occur when we apply other approaches to the same model in the lecture Optimal Savings II: LQ Techniques and this Jupyter notebook.

The plot below quickly replicates the first two figures of that lecture and that notebook to confirm that the solutions are the same

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))

for i in range(25):
    econ1.compute_sequence(x0, ts_length=150)
    ax1.plot(econ1.c[0], c='g')
    ax1.plot(econ1.d[0], c='b')
ax1.plot(econ1.c[0], label='Consumption', c='g')
ax1.plot(econ1.d[0], label='Income', c='b')
ax1.legend()

for i in range(25):
    econ1.compute_sequence(x0, ts_length=150)
    ax2.plot(econ1.k[0], color='r')
ax2.plot(econ1.k[0], label='Debt', c='r')
ax2.legend()
plt.show()
```


Consumption
Income
Debt

CHAPTER

TWENTYTHREE

ROSEN SCHOOLING MODEL

Contents

This lecture is yet another part of a suite of lectures that use the quantecon DLE class to instantiate models within the [HS13] class of models described in detail in *Recursive Models of Dynamic Linear Economies*.

In addition to what's included in Anaconda, this lecture uses the quantecon library

```
!pip install --upgrade quantecon
```

We'll also need the following imports:

```
import numpy as np
import matplotlib.pyplot as plt
from quantecon import LQ
from collections import namedtuple
from quantecon import DLE
from math import sqrt
%matplotlib inline
```

23.1 A One-Occupation Model

Ryoo and Rosen's (2004) [RR04] partial equilibrium model determines

- a stock of "Engineers" N_t
- a number of new entrants in engineering school, n_t
- the wage rate of engineers, w_t

It takes k periods of schooling to become an engineer.

The model consists of the following equations:

- a demand curve for engineers:

$$w_t = -\alpha_d N_t + \epsilon_{dt}$$

- a time-to-build structure of the education process:

$$N_{t+k} = \delta_N N_{t+k-1} + n_t$$

- a definition of the discounted present value of each new engineering student:

$$v_t = \beta_k \mathbb{E} \sum_{j=0}^{\infty} (\beta \delta_N)^j w_{t+k+j}$$

- a supply curve of new students driven by present value v_t:

$$n_t = \alpha_s v_t + \epsilon_{st}$$

23.2 Mapping into HS2013 Framework

We represent this model in the [HS13] framework by

- sweeping the time-to-build structure and the demand for engineers into the household technology, and
- putting the supply of engineers into the technology for producing goods

23.2.1 Preferences

$$\Pi = 0, \Lambda = \begin{bmatrix} \alpha_d & 0 & \cdots & 0 \end{bmatrix}, \Delta_h = \begin{bmatrix} \delta_N & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \Theta_h = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

where Λ is a k+1 x 1 matrix, Δ_h is a k_1 x k+1 matrix, and Θ_h is a k+1 x 1 matrix.

This specification sets $N_t = h_{1t-1}$, $n_t = c_t$, $h_{\tau+1,t-1} = n_{t-(k-\tau)}$ for $\tau = 1, ..., k$.

Below we set things up so that the number of years of education, k, can be varied.

23.2.2 Technology

To capture Ryoo and Rosen's [RR04] supply curve, we use the physical technology:

$$c_t = i_t + d_{1t}$$

$$\psi_1 i_t = g_t$$

where ψ_1 is inversely proportional to α_s.

23.2.3 Information

Because we want $b_t = \epsilon_{dt}$ and $d_{1t} = \epsilon_{st}$, we set

$$A_{22} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \rho_s & 0 \\ 0 & 0 & \rho_d \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, U_b = \begin{bmatrix} 30 & 0 & 1 \end{bmatrix}, U_d = \begin{bmatrix} 10 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where ρ_s and ρ_d describe the persistence of the supply and demand shocks

```
Information = namedtuple('Information', ['a22', 'c2','ub','ud'])
Technology = namedtuple('Technology', ['ϕ_c', 'ϕ_g', 'ϕ_i', 'γ', 'δ_k', 'θ_k'])
Preferences = namedtuple('Preferences', ['β', 'l_λ', 'π_h', 'δ_h', 'θ_h'])
```

23.2.4 Effects of Changes in Education Technology and Demand

We now study how changing

- the number of years of education required to become an engineer and
- the slope of the demand curve

affects responses to demand shocks.

To begin, we set $k = 4$ and $\alpha_d = 0.1$

```
k = 4  # Number of periods of schooling required to become an engineer

β = np.array([[1 / 1.05]])
α_d = np.array([[0.1]])
α_s = 1
ε_1 = 1e-7
λ_1 = np.full((1, k), ε_1)
# Use of ε_1 is trick to aquire detectability, see HS2013 p. 228 footnote 4
l_λ = np.hstack((α_d, λ_1))
π_h = np.array([[0]])

δ_n = np.array([[0.95]])
d1 = np.vstack((δ_n, np.zeros((k - 1, 1))))
d2 = np.hstack((d1, np.eye(k)))
δ_h = np.vstack((d2, np.zeros((1, k + 1))))

θ_h = np.vstack((np.zeros((k, 1)),
                 np.ones((1, 1))))

ψ_1 = 1 / α_s

ϕ_c = np.array([[1], [0]])
ϕ_g = np.array([[0], [-1]])
ϕ_i = np.array([[-1], [ψ_1]])
γ = np.array([[0], [0]])

δ_k = np.array([[0]])
θ_k = np.array([[0]])

ρ_s = 0.8
ρ_d = 0.8

a22 = np.array([[1,   0,   0],
                [0, ρ_s,   0],
                [0,   0, ρ_d]])

c2 = np.array([[0, 0], [10, 0], [0, 10]])
ub = np.array([[30, 0, 1]])
ud = np.array([[10, 1, 0], [0, 0, 0]])

info1 = Information(a22, c2, ub, ud)
```

(continues on next page)

(continued from previous page)

```
tech1 = Technology(ϕ_c, ϕ_g, ϕ_i, γ, δ_k, θ_k)
pref1 = Preferences(β, l_λ, π_h, δ_h, θ_h)

econ1 = DLE(info1, tech1, pref1)
```

We create three other instances by:

1. Raising α_d to 2
2. Raising k to 7
3. Raising k to 10

```
α_d = np.array([[2]])
l_λ = np.hstack((α_d, λ_1))
pref2 = Preferences(β, l_λ, π_h, δ_h, θ_h)
econ2 = DLE(info1, tech1, pref2)

α_d = np.array([[0.1]])

k = 7
λ_1 = np.full((1, k), ε_1)
l_λ = np.hstack((α_d, λ_1))
d1 = np.vstack((δ_n, np.zeros((k - 1, 1))))
d2 = np.hstack((d1, np.eye(k)))
δ_h = np.vstack((d2, np.zeros((1, k+1))))
θ_h = np.vstack((np.zeros((k, 1)),
                 np.ones((1, 1))))

Pref3 = Preferences(β, l_λ, π_h, δ_h, θ_h)
econ3 = DLE(info1, tech1, Pref3)

k = 10
λ_1 = np.full((1, k), ε_1)
l_λ = np.hstack((α_d, λ_1))
d1 = np.vstack((δ_n, np.zeros((k - 1, 1))))
d2 = np.hstack((d1, np.eye(k)))
δ_h = np.vstack((d2, np.zeros((1, k + 1))))
θ_h = np.vstack((np.zeros((k, 1)),
                 np.ones((1, 1))))

pref4 = Preferences(β, l_λ, π_h, δ_h, θ_h)
econ4 = DLE(info1, tech1, pref4)

shock_demand = np.array([[0], [1]])

econ1.irf(ts_length=25, shock=shock_demand)
econ2.irf(ts_length=25, shock=shock_demand)
econ3.irf(ts_length=25, shock=shock_demand)
econ4.irf(ts_length=25, shock=shock_demand)
```

The first figure plots the impulse response of n_t (on the left) and N_t (on the right) to a positive demand shock, for $\alpha_d = 0.1$ and $\alpha_d = 2$.

When $\alpha_d = 2$, the number of new students n_t rises initially, but the response then turns negative.

A positive demand shock raises wages, drawing new students into the profession.

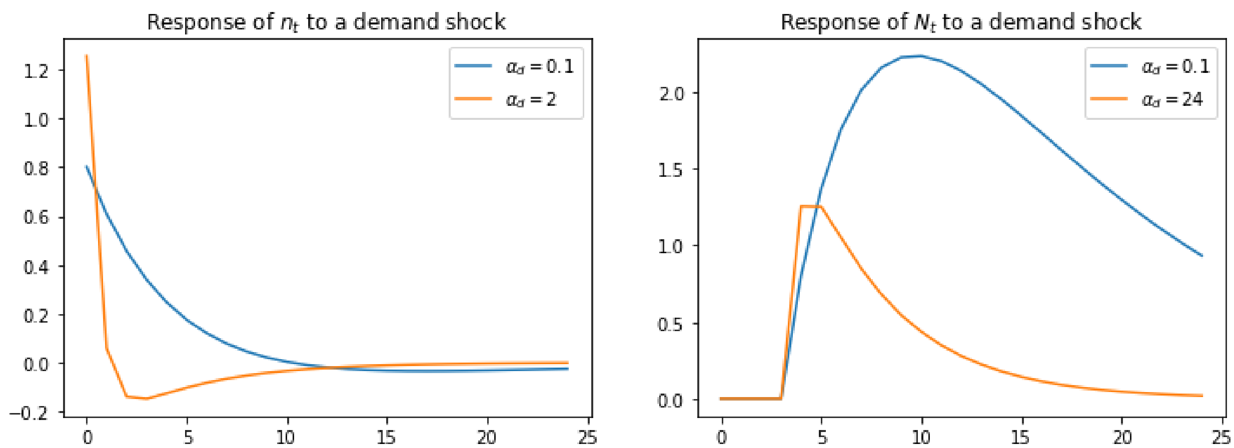
However, these new students raise N_t.

The higher is α_d, the larger the effect of this rise in N_t on wages.

This counteracts the demand shock's positive effect on wages, reducing the number of new students in subsequent periods.

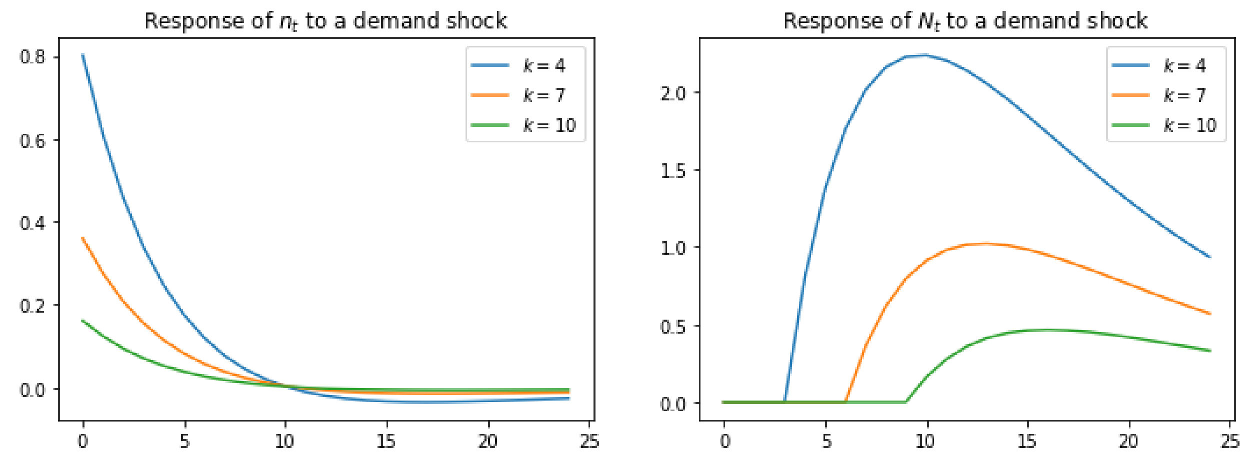
Consequently, when α_d is lower, the effect of a demand shock on N_t is larger

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(econ1.c_irf,label='$\\alpha_d = 0.1$')
ax1.plot(econ2.c_irf,label='$\\alpha_d = 2$')
ax1.legend()
ax1.set_title('Response of $n_t$ to a demand shock')

ax2.plot(econ1.h_irf[:, 0], label='$\\alpha_d = 0.1$')
ax2.plot(econ2.h_irf[:, 0], label='$\\alpha_d = 24$')
ax2.legend()
ax2.set_title('Response of $N_t$ to a demand shock')
plt.show()
```



The next figure plots the impulse response of n_t (on the left) and N_t (on the right) to a positive demand shock, for $k = 4$, $k = 7$ and $k = 10$ (with $\alpha_d = 0.1$)

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(econ1.c_irf, label='$k=4$')
ax1.plot(econ3.c_irf, label='$k=7$')
ax1.plot(econ4.c_irf, label='$k=10$')
ax1.legend()
ax1.set_title('Response of $n_t$ to a demand shock')

ax2.plot(econ1.h_irf[:,0], label='$k=4$')
ax2.plot(econ3.h_irf[:,0], label='$k=7$')
ax2.plot(econ4.h_irf[:,0], label='$k=10$')
ax2.legend()
ax2.set_title('Response of $N_t$ to a demand shock')
plt.show()
```



Both panels in the above figure show that raising k lowers the effect of a positive demand shock on entry into the engineering profession.

Increasing the number of periods of schooling lowers the number of new students in response to a demand shock.

This occurs because with longer required schooling, new students ultimately benefit less from the impact of that shock on wages.

CHAPTER

TWENTYFOUR

CATTLE CYCLES

Contents

This is another member of a suite of lectures that use the quantecon DLE class to instantiate models within the [HS13] class of models described in detail in *Recursive Models of Dynamic Linear Economies*.

In addition to what's in Anaconda, this lecture uses the quantecon library.

```
!pip install --upgrade quantecon
```

This lecture uses the DLE class to construct instances of the "Cattle Cycles" model of Rosen, Murphy and Scheinkman (1994) [RMS94].

That paper constructs a rational expectations equilibrium model to understand sources of recurrent cycles in US cattle stocks and prices.

We make the following imports:

```
import numpy as np
import matplotlib.pyplot as plt
from quantecon import LQ
from collections import namedtuple
from quantecon import DLE
from math import sqrt
%matplotlib inline
```

24.1 The Model

The model features a static linear demand curve and a "time-to-grow" structure for cattle.

Let p_t be the price of slaughtered beef, m_t the cost of preparing an animal for slaughter, h_t the holding cost for a mature animal, $\gamma_1 h_t$ the holding cost for a yearling, and $\gamma_0 h_t$ the holding cost for a calf.

The cost processes $\{h_t, m_t\}_{t=0}^{\infty}$ are exogenous, while the price process $\{p_t\}_{t=0}^{\infty}$ is determined within a rational expectations equilibrium.

Let x_t be the breeding stock, and y_t be the total stock of cattle.

The law of motion for the breeding stock is

$$x_t = (1-\delta)x_{t-1} + gx_{t-3} - c_t$$

where $g < 1$ is the number of calves that each member of the breeding stock has each year, and c_t is the number of cattle slaughtered.

The total headcount of cattle is

$$y_t = x_t + gx_{t-1} + gx_{t-2}$$

This equation states that the total number of cattle equals the sum of adults, calves and yearlings, respectively.

A representative farmer chooses $\{c_t, x_t\}$ to maximize:

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \{p_t c_t - h_t x_t - \gamma_0 h_t (gx_{t-1}) - \gamma_1 h_t (gx_{t-2}) - m_t c_t - \frac{\psi_1}{2} x_t^2 - \frac{\psi_2}{2} x_{t-1}^2 - \frac{\psi_3}{2} x_{t-3}^2 - \frac{\psi_4}{2} c_t^2\}$$

subject to the law of motion for x_t, taking as given the stochastic laws of motion for the exogenous processes, the equilibrium price process, and the initial state $[x_{-1}, x_{-2}, x_{-3}]$.

Remark The ψ_j parameters are very small quadratic costs that are included for technical reasons to make well posed and well behaved the linear quadratic dynamic programming problem solved by the fictitious planner who in effect chooses equilibrium quantities and shadow prices.

Demand for beef is government by $c_t = a_0 - a_1 p_t + \tilde{d}_t$ where $\tilde{d}_t$ is a stochastic process with mean zero, representing a demand shifter.

24.2 Mapping into HS2013 Framework

24.2.1 Preferences

We set $\Lambda = 0, \Delta_h = 0, \Theta_h = 0, \Pi = \alpha_1^{-\frac{1}{2}}$ and $b_t = \Pi \tilde{d}_t + \Pi \alpha_0$.

With these settings, the FOC for the household's problem becomes the demand curve of the "Cattle Cycles" model.

24.2.2 Technology

To capture the law of motion for cattle, we set

$$\Delta_k = \begin{bmatrix} (1-\delta) & 0 & g \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \Theta_k = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

(where $i_t = -c_t$).

To capture the production of cattle, we set

$$\Phi_c = \begin{bmatrix} 1 \\ f_1 \\ 0 \\ 0 \\ -f_7 \end{bmatrix}, \Phi_g = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Phi_i = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \Gamma = \begin{bmatrix} 0 & 0 & 0 \\ f_1(1-\delta) & 0 & gf_1 \\ f_3 & 0 & 0 \\ 0 & f_5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

24.2.3 Information

We set

$$A_{22} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \rho_1 & 0 & 0 \\ 0 & 0 & \rho_2 & 0 \\ 0 & 0 & 0 & \rho_3 \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 15 \end{bmatrix}, U_b = \begin{bmatrix} \Pi\alpha_0 & 0 & 0 & \Pi \end{bmatrix}, U_d = \begin{bmatrix} 0 \\ f_2 U_h \\ f_4 U_h \\ f_6 U_h \\ f_8 U_h \end{bmatrix}$$

To map this into our class, we set $f_1^2 = \frac{\Psi_1}{2}$, $f_2^2 = \frac{\Psi_2}{2}$, $f_3^2 = \frac{\Psi_3}{2}$, $2f_1 f_2 = 1$, $2f_3 f_4 = \gamma_0 g$, $2f_5 f_6 = \gamma_1 g$.

```
# We define namedtuples in this way as it allows us to check, for example,
# what matrices are associated with a particular technology.

Information = namedtuple('Information', ['a22', 'c2', 'ub', 'ud'])
Technology = namedtuple('Technology', ['ϕ_c', 'ϕ_g', 'ϕ_i', 'γ', 'δ_k', 'θ_k'])
Preferences = namedtuple('Preferences', ['β', 'l_λ', 'π_h', 'δ_h', 'θ_h'])
```

We set parameters to those used by [RMS94]

```
β = np.array([[0.909]])
lλ = np.array([[0]])

a1 = 0.5
πh = np.array([[1 / (sqrt(a1))]])
δh = np.array([[0]])
θh = np.array([[0]])

δ = 0.1
g = 0.85
f1 = 0.001
f3 = 0.001
f5 = 0.001
f7 = 0.001

ϕc = np.array([[1], [f1], [0], [0], [-f7]])

ϕg = np.array([[0, 0, 0, 0],
               [1, 0, 0, 0],
               [0, 1, 0, 0],
               [0, 0, 1,0],
               [0, 0, 0, 1]])

ϕi = np.array([[1], [0], [0], [0], [0]])

γ = np.array([[           0,  0,      0],
              [f1 * (1 - δ),  0, g * f1],
              [          f3,  0,      0],
              [           0, f5,      0],
              [           0,  0,      0]])

δk = np.array([[1 - δ, 0, g],
               [    1, 0, 0],
               [    0, 1, 0]])

θk = np.array([[1], [0], [0]])
```

(continues on next page)

(continued from previous page)

```
ρ1 = 0
ρ2 = 0
ρ3 = 0.6
a0 = 500
γ0 = 0.4
γ1 = 0.7
f2 = 1 / (2 * f1)
f4 = γ0 * g / (2 * f3)
f6 = γ1 * g / (2 * f5)
f8 = 1 / (2 * f7)

a22 = np.array([[1, 0, 0, 0],
                [0, ρ1, 0, 0],
                [0, 0, ρ2, 0],
                [0, 0, 0, ρ3]])

c2 = np.array([[0, 0,  0],
               [1, 0,  0],
               [0, 1,  0],
               [0, 0, 15]])

ub = np.array([[πh * a0, 0, 0, πh]])
uh = np.array([[50, 1, 0, 0]])
um = np.array([[100, 0, 1, 0]])
ud = np.vstack(([0, 0, 0, 0],
                f2 * uh, f4 * uh, f6 * uh, f8 * um))
```

```
<ipython-input-4-568c6d4eae41>:59: VisibleDeprecationWarning: Creating an ndarray␣
↪from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or␣
↪ndarrays with different lengths or shapes) is deprecated. If you meant to do this,␣
↪you must specify 'dtype=object' when creating the ndarray
  ub = np.array([[πh * a0, 0, 0, πh]])
```

Notice that we have set $\rho_1 = \rho_2 = 0$, so h_t and m_t consist of a constant and a white noise component.

We set up the economy using tuples for information, technology and preference matrices below.

We also construct two extra information matrices, corresponding to cases when $\rho_3 = 1$ and $\rho_3 = 0$ (as opposed to the baseline case of $\rho_3 = 0.6$).

```
info1 = Information(a22, c2, ub, ud)
tech1 = Technology(ϕc, ϕg, ϕi, γ, δk, θk)
pref1 = Preferences(β, lλ, πh, δh, θh)

ρ3_2 = 1
a22_2 = np.array([[1,  0,  0,    0],
                  [0, ρ1,  0,    0],
                  [0,  0, ρ2,    0],
                  [0,  0,  0, ρ3_2]])

info2 = Information(a22_2, c2, ub, ud)

ρ3_3 = 0
a22_3 = np.array([[1,  0,  0,    0],
                  [0, ρ1,  0,    0],
                  [0,  0, ρ2,    0],
                  [0,  0,  0, ρ3_3]])
```

(continues on next page)

(continued from previous page)

```
info3 = Information(a22_3, c2, ub, ud)

# Example of how we can look at the matrices associated with a given namedtuple
info1.a22
```

```
array([[1. , 0. , 0. , 0. ],
       [0. , 0. , 0. , 0. ],
       [0. , 0. , 0. , 0. ],
       [0. , 0. , 0. , 0.6]])
```

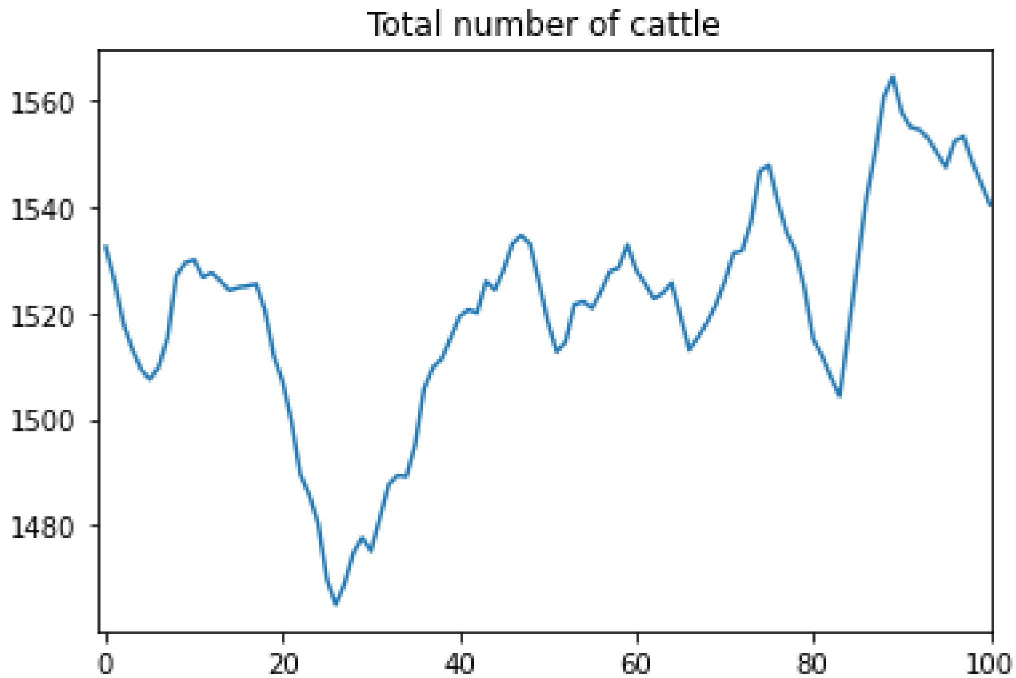
```
# Use tuples to define DLE class
econ1 = DLE(info1, tech1, pref1)
econ2 = DLE(info2, tech1, pref1)
econ3 = DLE(info3, tech1, pref1)

# Calculate steady-state in baseline case and use to set the initial condition
econ1.compute_steadystate(nnc=4)
x0 = econ1.zz
```

```
econ1.compute_sequence(x0, ts_length=100)
```

[RMS94] use the model to understand the sources of recurrent cycles in total cattle stocks.

Plotting y_t for a simulation of their model shows its ability to generate cycles in quantities

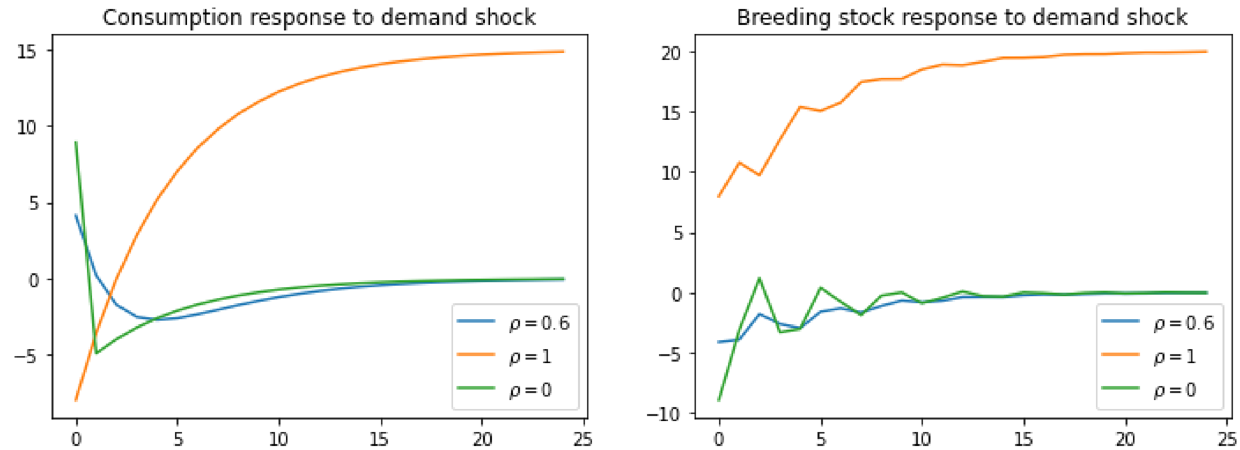
```
# Calculation of y_t
totalstock = econ1.k[0] + g * econ1.k[1] + g * econ1.k[2]
fig, ax = plt.subplots()
ax.plot(totalstock)
ax.set_xlim((-1, 100))
ax.set_title('Total number of cattle')
plt.show()
```



In their Figure 3, [RMS94] plot the impulse response functions of consumption and the breeding stock of cattle to the demand shock, $\tilde{d}_t$, under the three different values of ρ_3.

We replicate their Figure 3 below

```
shock_demand = np.array([[0], [0], [1]])

econ1.irf(ts_length=25, shock=shock_demand)
econ2.irf(ts_length=25, shock=shock_demand)
econ3.irf(ts_length=25, shock=shock_demand)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(econ1.c_irf, label=r'$\rho=0.6$')
ax1.plot(econ2.c_irf, label=r'$\rho=1$')
ax1.plot(econ3.c_irf, label=r'$\rho=0$')
ax1.set_title('Consumption response to demand shock')
ax1.legend()

ax2.plot(econ1.k_irf[:, 0], label=r'$\rho=0.6$')
ax2.plot(econ2.k_irf[:, 0], label=r'$\rho=1$')
ax2.plot(econ3.k_irf[:, 0], label=r'$\rho=0$')
ax2.set_title('Breeding stock response to demand shock')
ax2.legend()
plt.show()
```



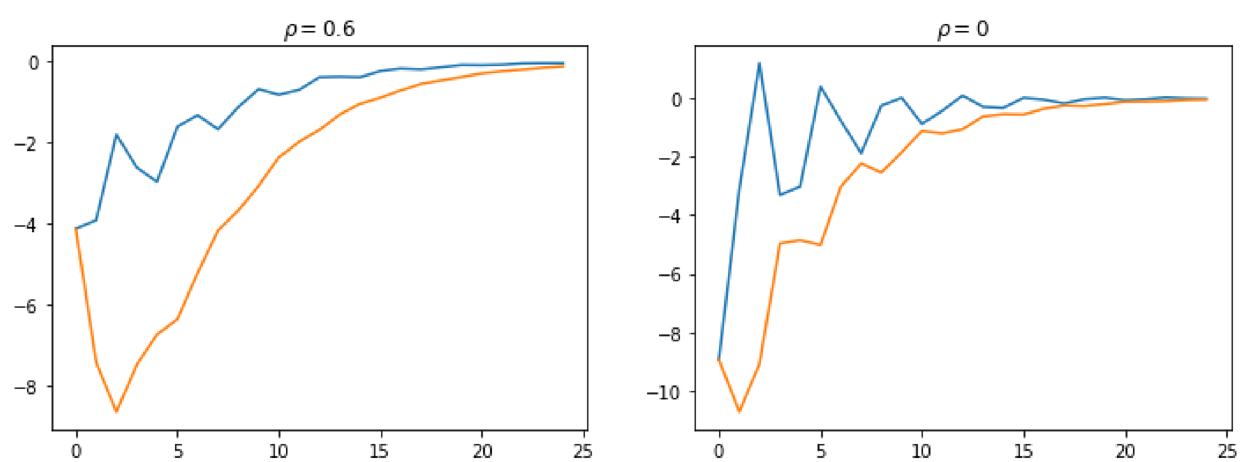
The above figures show how consumption patterns differ markedly, depending on the persistence of the demand shock:

- If it is purely transitory ($\rho_3 = 0$) then consumption rises immediately but is later reduced to build stocks up again.
- If it is permanent ($\rho_3 = 1$), then consumption falls immediately, in order to build up stocks to satisfy the permanent rise in future demand.

In Figure 4 of their paper, [RMS94] plot the response to a demand shock of the breeding stock *and* the total stock, for $\rho_3 = 0$ and $\rho_3 = 0.6$.

We replicate their Figure 4 below

```
total1_irf = econ1.k_irf[:, 0] + g * econ1.k_irf[:, 1] + g * econ1.k_irf[:, 2]
total3_irf = econ3.k_irf[:, 0] + g * econ3.k_irf[:, 1] + g * econ3.k_irf[:, 2]

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(econ1.k_irf[:, 0], label='Breeding Stock')
ax1.plot(total1_irf, label='Total Stock')
ax1.set_title(r'$\rho=0.6$')

ax2.plot(econ3.k_irf[:, 0], label='Breeding Stock')
ax2.plot(total3_irf, label='Total Stock')
ax2.set_title(r'$\rho=0$')
plt.show()
```



The fact that y_t is a weighted moving average of x_t creates a humped shape response of the total stock in response to demand shocks, contributing to the cyclicality seen in the first graph of this lecture.

CHAPTER

TWENTYFIVE

SHOCK NON INVERTIBILITY

Contents

25.1 Overview

This is another member of a suite of lectures that use the quantecon DLE class to instantiate models within the [HS13] class of models described in detail in *Recursive Models of Dynamic Linear Economies*.

In addition to what's in Anaconda, this lecture uses the quantecon library.

```
!pip install --upgrade quantecon
```

We'll make these imports:

```
import numpy as np
import quantecon as qe
import matplotlib.pyplot as plt
from quantecon import LQ
from quantecon import DLE
from math import sqrt
%matplotlib inline
```

This lecture can be viewed as introducing an early contribution to what is now often called a **news and noise** issue.

In particular, it analyzes a **shock-invertibility** issue that is endemic within a class of permanent income models.

Technically, the invertibility problem indicates a situation in which histories of the shocks in an econometrician's autoregressive or Wold moving average representation span a smaller information space than do the shocks that are seen by the agents inside the econometrician's model.

This situation sets the stage for an econometrician who is unaware of the problem and consequently misinterprets shocks and likely responses to them.

A shock-invertibility that is technically close to the one studied here is discussed by Eric Leeper, Todd Walker, and Susan Yang [LWY13] in their analysis of **fiscal foresight**.

A distinct shock-invertibility issue is present in the special LQ consumption smoothing model in *quantecon lecture*.

25.2 Model

We consider the following modification of Robert Hall's (1978) model [Hal78] in which the endowment process is the sum of two orthogonal autoregressive processes:

Preferences

$$-\frac{1}{2}\mathbb{E}\sum_{t=0}^{\infty}\beta^t[(c_t-b_t)^2+l_t^2]|J_0$$

$$s_t = c_t$$

$$b_t = U_b z_t$$

Technology

$$c_t + i_t = \gamma k_{t-1} + d_t$$

$$k_t = \delta_k k_{t-1} + i_t$$

$$g_t = \phi_1 i_t \, , \phi_1 > 0$$

$$g_t \cdot g_t = l_t^2$$

Information

$$z_{t+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} z_t + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} w_{t+1}$$

$$U_b = \begin{bmatrix} 30 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U_d = \begin{bmatrix} 5 & 1 & 1 & 0.8 & 0.6 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The preference shock is constant at 30, while the endowment process is the sum of a constant and two orthogonal processes.

Specifically:

$$d_t = 5 + d_{1t} + d_{2t}$$

$$d_{1t} = 0.9 d_{1t-1} + w_{1t}$$

$$d_{2t} = 4w_{2t} + 0.8(4w_{2t-1}) + 0.6(4w_{2t-2}) + 0.4(4w_{2t-3})$$

d_{1t} is a first-order AR process, while d_{2t} is a third-order pure moving average process.

```
γ_1 = 0.05
γ = np.array([[γ_1], [0]])
ϕ_c = np.array([[1], [0]])
ϕ_g = np.array([[0], [1]])
ϕ_1 = 0.00001
ϕ_i = np.array([[1], [-ϕ_1]])
δ_k = np.array([[1]])
θ_k = np.array([[1]])
β = np.array([[1 / 1.05]])
l_λ = np.array([[0]])
π_h = np.array([[1]])
δ_h = np.array([[.9]])
θ_h = np.array([[1]]) - δ_h
ud = np.array([[5, 1, 1, 0.8, 0.6, 0.4],
               [0, 0, 0,   0,   0,   0]])
a22 = np.zeros((6, 6))
# Chase's great trick
a22[[0, 1, 3, 4, 5], [0, 1, 2, 3, 4]] = np.array([1.0, 0.9, 1.0, 1.0, 1.0])
c2 = np.zeros((6, 2))
c2[[1, 2], [0, 1]] = np.array([1.0, 4.0])
ub = np.array([[30, 0, 0, 0, 0, 0]])
x0 = np.array([[5], [150], [1], [0], [0], [0], [0], [0]])

info1 = (a22, c2, ub, ud)
tech1 = (ϕ_c, ϕ_g, ϕ_i, γ, δ_k, θ_k)
pref1 = (β, l_λ, π_h, δ_h, θ_h)

econ1 = DLE(info1, tech1, pref1)
```

We define the household's net of interest deficit as $c_t - d_t$.

Hall's model imposes "expected present-value budget balance" in the sense that

$$\mathbb{E} \sum_{j=0}^{\infty} \beta^j (c_{t+j} - d_{t+j}) | J_t = \beta^{-1} k_{t-1} \, \forall t$$

If we define the moving average representation of $(c_t, c_t - d_t)$ in terms of the w_ts to be:

$$\begin{bmatrix} c_t \\ c_t - d_t \end{bmatrix} = \begin{bmatrix} \sigma_1(L) \\ \sigma_2(L) \end{bmatrix} w_t$$

then Hall's model imposes the restriction $\sigma_2(\beta) = [0 \;\; 0]$.

The agent inside this model sees histories of both components of the endowment process d_{1t} and d_{2t}.

The econometrician has data on the history of the pair $[c_t, d_t]$, but not directly on the history of w_t.

The econometrician obtains a Wold representation for the process $[c_t, c_t - d_t]$:

$$\begin{bmatrix} c_t \\ c_t - d_t \end{bmatrix} = \begin{bmatrix} \sigma_1^*(L) \\ \sigma_2^*(L) \end{bmatrix} u_t$$

The Appendix of chapter 8 of [HS13] explains why the impulse response functions in the Wold representation estimated by the econometrician do not resemble the impulse response functions that depict the response of consumption and the deficit to innovations to agents' information.

Technically, $\sigma_2(\beta) = [0 \;\; 0]$ implies that the history of u_ts spans a *smaller* linear space than does the history of w_ts.

This means that u_t will typically be a distributed lag of w_t that is not concentrated at zero lag:

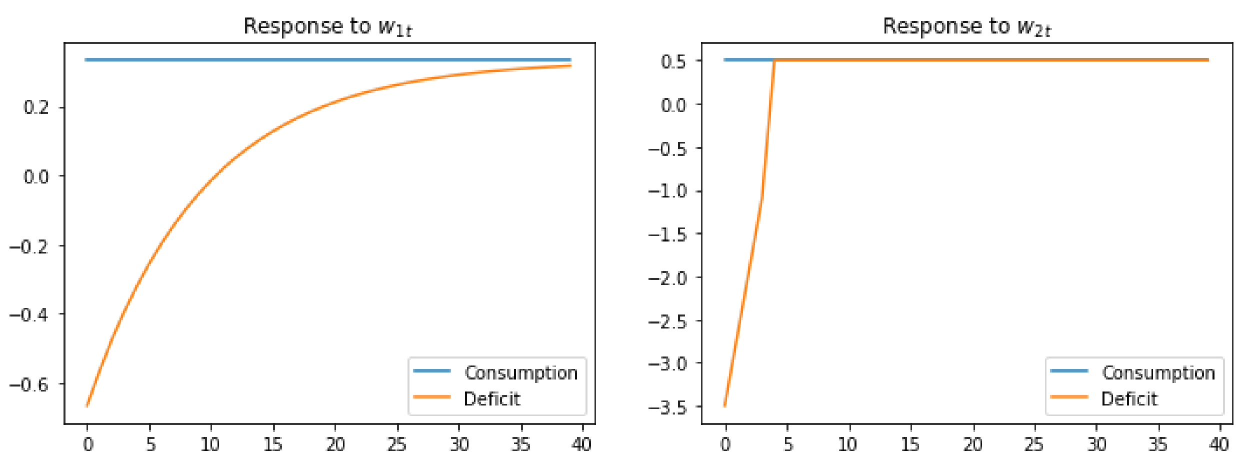
$$u_t = \sum_{j=0}^{\infty} \alpha_j w_{t-j}$$

Thus, the econometrician's news u_t potentially responds belatedly to agents' news w_t.

25.3 Code

We will construct Figures from Chapter 8 Appendix E of [HS13] to illustrate these ideas:

```
# This is Fig 8.E.1 from p.188 of HS2013

econ1.irf(ts_length=40, shock=None)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(econ1.c_irf, label='Consumption')
ax1.plot(econ1.c_irf - econ1.d_irf[:,0].reshape(40,1), label='Deficit')
ax1.legend()
ax1.set_title('Response to $w_{1t}$')

shock2 = np.array([[0], [1]])
econ1.irf(ts_length=40, shock=shock2)

ax2.plot(econ1.c_irf, label='Consumption')
ax2.plot(econ1.c_irf - econ1.d_irf[:,0].reshape(40, 1), label='Deficit')
ax2.legend()
ax2.set_title('Response to $w_{2t}$')
plt.show()
```



The above figure displays the impulse response of consumption and the deficit to the endowment innovations.

Consumption displays the characteristic "random walk" response with respect to each innovation.

Each endowment innovation leads to a temporary surplus followed by a permanent net-of-interest deficit.

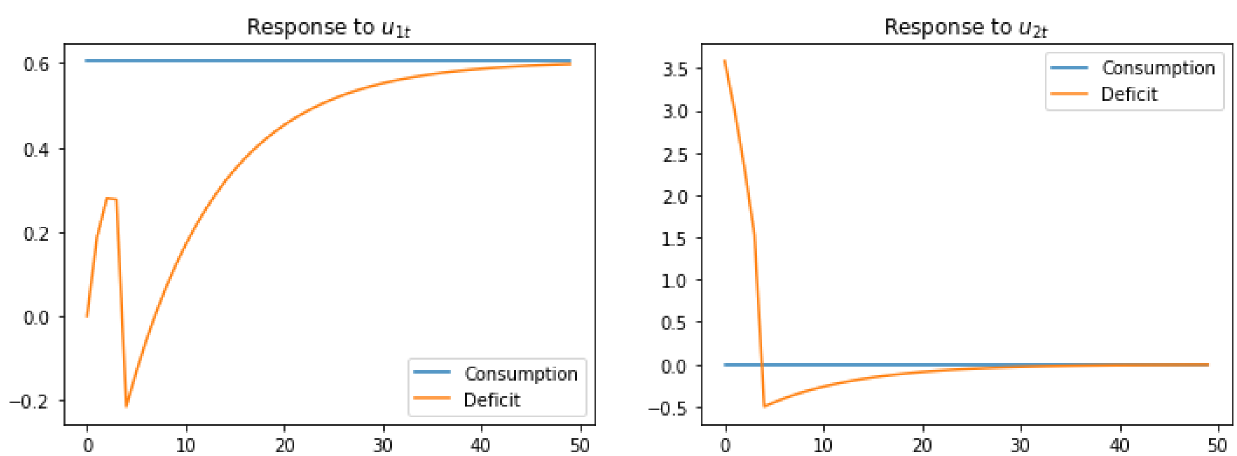
The temporary surplus just offsets the permanent deficit in terms of expected present value.

```
G_HS = np.vstack([econ1.Sc, econ1.Sc-econ1.Sd[0, :].reshape(1, 8)])
H_HS = 1e-8 * np.eye(2)  # Set very small so there is no measurement error
lss_hs = qe.LinearStateSpace(econ1.A0, econ1.C, G_HS, H_HS)

hs_kal = qe.Kalman(lss_hs)
w_lss = hs_kal.whitener_lss()
ma_coefs = hs_kal.stationary_coefficients(50, 'ma')
```

(continues on next page)

(continued from previous page)

```
# This is Fig 8.E.2 from p.189 of HS2013

ma_coefs = ma_coefs
jj = 50
y1_w1 = np.empty(jj)
y2_w1 = np.empty(jj)
y1_w2 = np.empty(jj)
y2_w2 = np.empty(jj)

for t in range(jj):
    y1_w1[t] = ma_coefs[t][0, 0]
    y1_w2[t] = ma_coefs[t][0, 1]
    y2_w1[t] = ma_coefs[t][1, 0]
    y2_w2[t] = ma_coefs[t][1, 1]

# This scales the impulse responses to match those in the book
y1_w1 = sqrt(hs_kal.stationary_innovation_covar()[0, 0]) * y1_w1
y2_w1 = sqrt(hs_kal.stationary_innovation_covar()[0, 0]) * y2_w1
y1_w2 = sqrt(hs_kal.stationary_innovation_covar()[1, 1]) * y1_w2
y2_w2 = sqrt(hs_kal.stationary_innovation_covar()[1, 1]) * y2_w2

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(y1_w1, label='Consumption')
ax1.plot(y2_w1, label='Deficit')
ax1.legend()
ax1.set_title('Response to $u_{1t}$')

ax2.plot(y1_w2, label='Consumption')
ax2.plot(y2_w2, label='Deficit')
ax2.legend()
ax2.set_title('Response to $u_{2t}$')
plt.show()
```



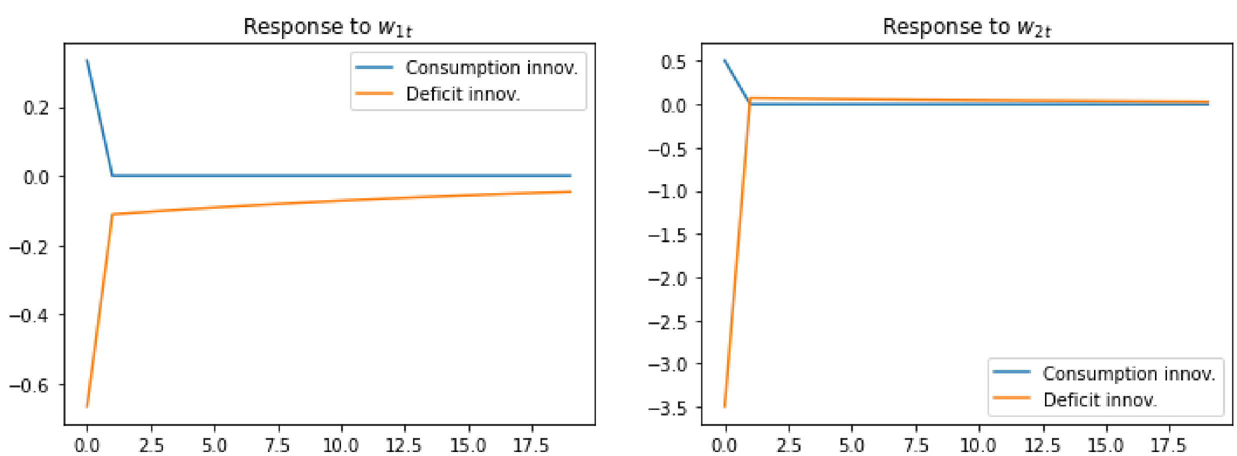
The above figure displays the impulse response of consumption and the deficit to the innovations in the econometrician's Wold representation

- this is the object that would be recovered from a high order vector autoregression on the econometrician's observations.

Consumption responds only to the first innovation

- this is indicative of the Granger causality imposed on the $[c_t, c_t - d_t]$ process by Hall's model: consumption Granger causes $c_t - d_t$, with no reverse causality.

```
# This is Fig 8.E.3 from p.189 of HS2013

jj = 20
irf_wlss = w_lss.impulse_response(jj)
ycoefs = irf_wlss[1]
# Pull out the shocks
a1_w1 = np.empty(jj)
a1_w2 = np.empty(jj)
a2_w1 = np.empty(jj)
a2_w2 = np.empty(jj)

for t in range(jj):
    a1_w1[t] = ycoefs[t][0, 0]
    a1_w2[t] = ycoefs[t][0, 1]
    a2_w1[t] = ycoefs[t][1, 0]
    a2_w2[t] = ycoefs[t][1, 1]

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(a1_w1, label='Consumption innov.')
ax1.plot(a2_w1, label='Deficit innov.')
ax1.set_title('Response to $w_{1t}$')
ax1.legend()
ax2.plot(a1_w2, label='Consumption innov.')
ax2.plot(a2_w2, label='Deficit innov.')
ax2.legend()
ax2.set_title('Response to $w_{2t}$')
plt.show()
```



The above figure displays the impulse responses of u_t to w_t, as depicted in:

$$u_t = \sum_{j=0}^{\infty} \alpha_j w_{t-j}$$

While the responses of the innovations to consumption are concentrated at lag zero for both components of w_t, the responses of the innovations to $(c_t - d_t)$ are spread over time (especially in response to w_{1t}).

Thus, the innovations to $(c_t - d_t)$ as revealed by the vector autoregression depend on what the economic agent views as "old news".