Applications of Mathematical Economics

Michael Curran

Trinity College Dublin

Overview

- Introduction.
- Data Preparation – Filters.
- Dynamic Stochastic General Equilibrium Models:
    - Sunspots and Blanchard-Kahn Conditions.
    - Solving and Estimating DSGE Models.
- Other Quantitative Tools (Bayesian Econometrics, Non-Linear Models).
- Computers in Economics (Graphical Programming, Parallel Programming, Comparison of Computer Languages in Economics).

Introduction

- Why should we care about macroeconomics and macroeconometrics?
- Why take the formal approach in Economics?

  'It is not from the benevolence of the butcher, the brewer, or the baker that we expect our dinner, but from their regard to their own interest.' (Wealth of Nations I, ii,2:26-27)

  'Ranges are for cattle. Give me a number.'

Introduction

- Lucas Critique

    'Given that the structure of an econometric model consists of optimal decision rules of economic agents, and that optimal decision rules vary systematically with changes in the structure of series relevant to the decision maker, it follows that any change in policy will systematically alter the structure of econometric models.'
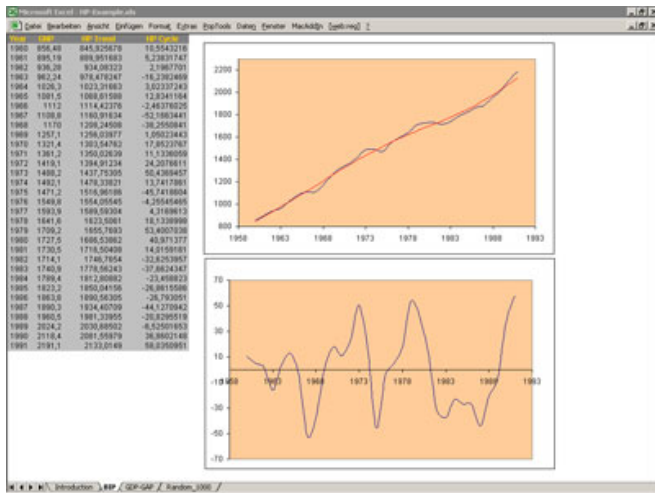
- `http://www.nobelprize.org/mediaplayer/index.php?id=1743`

Data Preparation

- Why should we study the frequency domain?
- Data can be thought of as weighted sum of cosine waves.
- Filters.

Data Preparation

Filters

Figure: Hodrick-Prescott Filter

DSGE Models

- ▶ Dynamic Stochastic General Equilibrium.
- ▶ Stability, Multiplicity and Solutions to Linearised Systems:
    - ▶ Sunspots in Economics.
    - ▶ Blanchard-Kahn Conditions.
    - ▶ What if **A** is not invertible?
    - ▶ Time Iteration.
- ▶ Solving and Estimating a DSGE Model.

DSGE Models

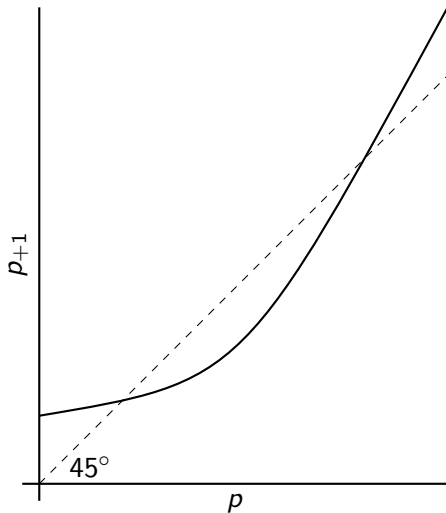Stability, Multiplicity and Solutions to Linearised Systems

Figure: Unique solution and multiple steady states

DSGE Models

Stability, Multiplicity and Solutions to Linearised Systems
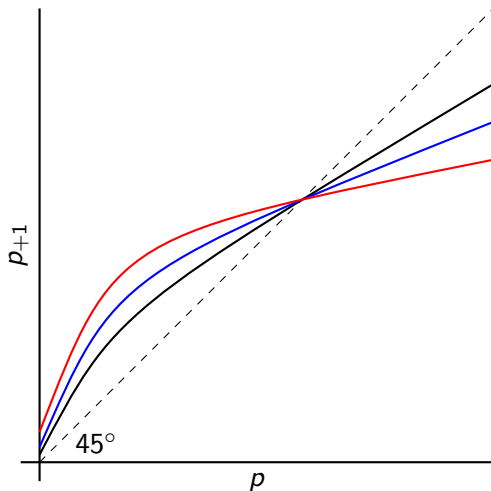

Figure: Multiple solutions and unique (non-zero) steady state

DSGE Models

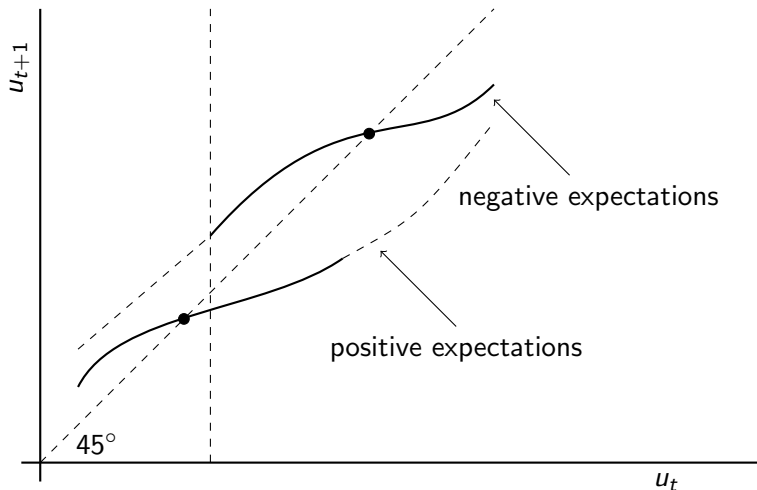Stability, Multiplicity and Solutions to Linearised Systems

Figure: Multiple steady states and sometimes multiple solutions

DSGE Models
Sunspots in Economics

- A solution is a sunspot solution if it depends on a stochastic variable from outside the system.
- https://www.youtube.com/watch?v=UD5VViT08ME

DSGE Models
Sunspots in Economics

Suppose the model is

$$0 = \mathbb{E}\left[H(p_{t+1}, p_t, d_{t+1}, d_t)\right]$$

$$d_t : \text{exogenous random variable}$$

A non-sunspot solution is

$$p_t = f(p_{t-1}, p_{t-2}, \ldots, d_t, d_{t-1}, \ldots)$$

A sunspot solution is

$$p_t : f(p_{t-1}, p_{t-2}, \ldots, d_t, d_{t-1}, \ldots, s_t)$$

$$s_t : \text{random variable with } \mathbb{E}\left[s_{t+1}\right] = 0$$
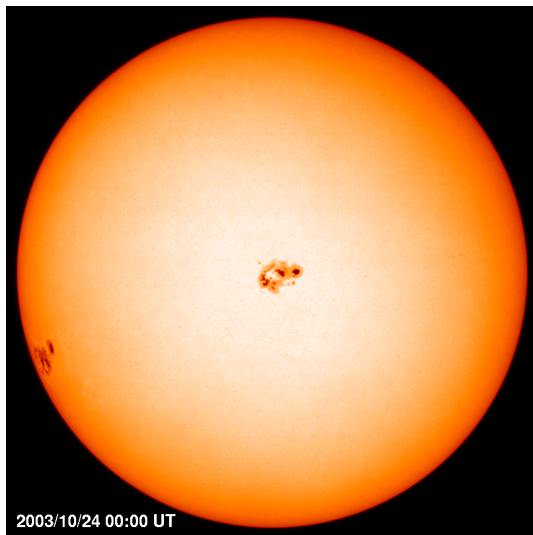
DSGE Models

Sunspots in Economics

Figure: Large sunspots (MDI image of sunspot region 10484).

DSGE Models

Sunspots in Economics

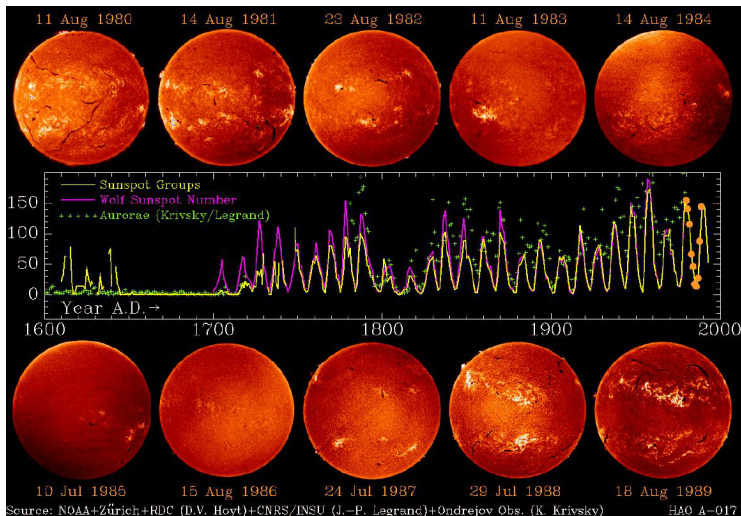Figure: Past sun spot cycles – sun spots had a 'Great Moderation'.
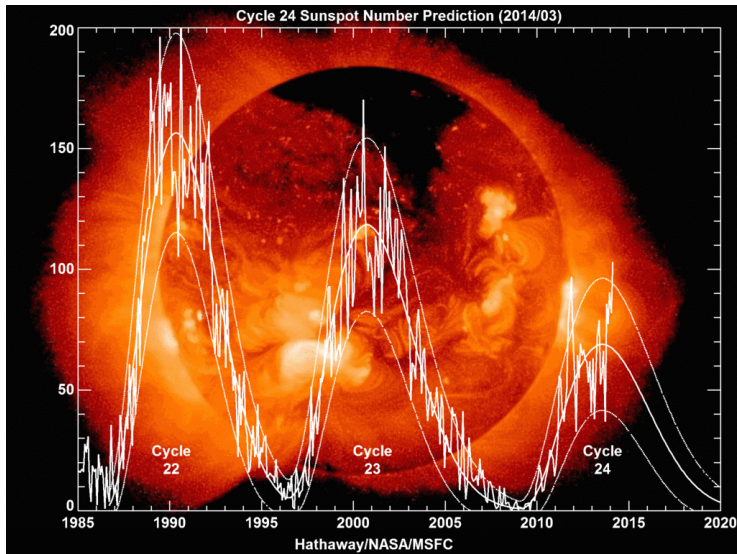
DSGE Models

Sunspots in Economics

Figure: Current cycle (at peak again).

Blanchard-Kahn condition states that the solution of the rational expectations model is unique if the number of unstable eigenvectors of the system is exactly equal to the number of forward-looking (control) variables.

Q: What if **A** is not invertible?

A: Use Schur decomposition (Klein, 2000).

Model: $\Gamma_2 k_{t+1} + \Gamma_1 k_t + \Gamma_0 k_{t-1} = 0$

Impose solution: $k_t = a k_{t-1}$

Start with $a_{[i]}$ and using guess for tomorrow's behaviour, solve for today's behaviour:

$$\Gamma_2 a^2 k_{t-1} + \Gamma_1 a k_{t-1} + \Gamma_0 k_{t-1} = 0 \text{ for all } k_{t-1}$$

$$(\Gamma_2 a_{[i]} + \Gamma_1) k_t + \Gamma_0 k_{t-1} = 0$$
$$k_t = -(\Gamma_2 a_{[i]} + \Gamma_1)^{-1} \Gamma_0 k_{t-1}$$
$$a_{[i+1]} = -(\Gamma_2 a_{[i]} + \Gamma_1)^{-1} \Gamma_0$$

Advantages: simple; nice convergence properties.

DSGE Models
Solving and Estimating DSGE Models

- ▶ Cast in linear/log-linear form or non-linear model representation.
- ▶ Linear solution techniques: Blanchard / Sims / Klein / Method of Undetermined Coefficients.
- ▶ Non-linear solution techniques: global / iteration / local.
- ▶ Calibration and Moment Matching.
- ▶ Estimation: Moment Matching (Generalised Method of Moments / Simulated Method of Moments / indirect inference) / Maximum Likelihood; Bayesian methods (Sequential Monte Carlo and Markov Chain Monte Carlo).
- ▶ Dynare (runs in MATLAB): `www.dynare.org` and `http://www.dynare.org/documentation-and-support/user-guide/Dynare-UserGuide-WebBeta.pdf`.
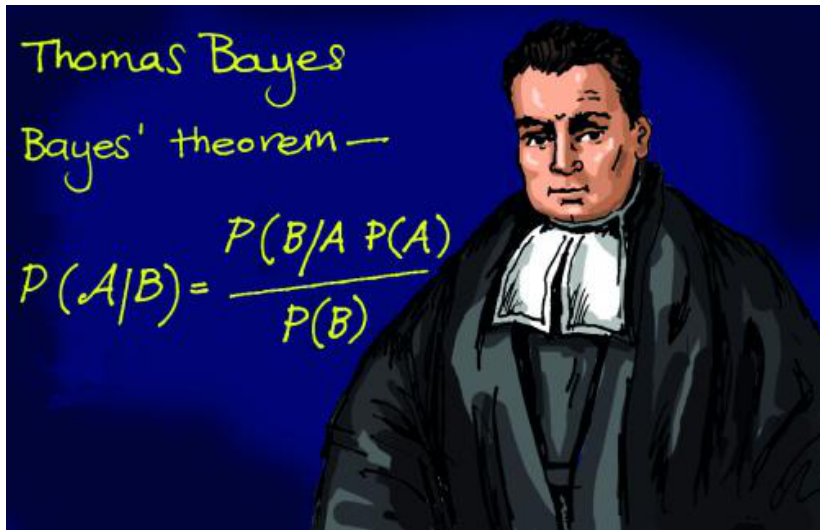
Other Quantitative Tools

Bayesian Econometrics

Figure: Thomas Bayes – Bayes' Rule

Bayesian Econometrics and Non-Linear / Non-Gaussian DSGE Models

- Why Econometrics Should Always and Everywhere be Bayesian: `http://sims.princeton.edu/yftp/EmetSoc607/AppliedBayes.pdf`.
- Why Non-Linear / Non-Gaussian DSGE Models? `http://www.nber.org/econometrics_minicourse_2011/`.

Computers

```
http://www.youtube.com/watch?v=Fg85ggZSHMw
www.top500.org
```

A brief history of computing

- 3000BC abacus (1 FLOP)
- 1613 ('computer': a person who computes)
- 1642 Pascal's adding machine
- 1832 Babbage's difference engine
- 1904 Diodes
- 1936 'On Computable Numbers' (Turing)
- 1943 Colossus (code breaking)
- 1945 ENIAC
- 1947 Transistors
- 1949 MONIAC
- 1957 Fortran
- 1962 Atlas
- 1969 Arpanet
- 1970 Unix and C

- 1976 Cray I (100 MFLOPS)
- 1983 SX-1 (570 MFLOPS)
- ETA-10P (750 MFLOPS) [Mendoza: '87/'91: RBC in SOE]
- 1994 MPI
- 1997 OpenMP
- 2001 Earth Simulator (40 TFLOPS)
- 2008 Roadrunner (1 PFLOPS)
- 2009 TCHPC [Trinity College]: Lonsdale (11 TFLOPS)
- 2013 Tianhe-2 (34 PFLOPS) / Ireland: Fionn (140 TFLOPS)
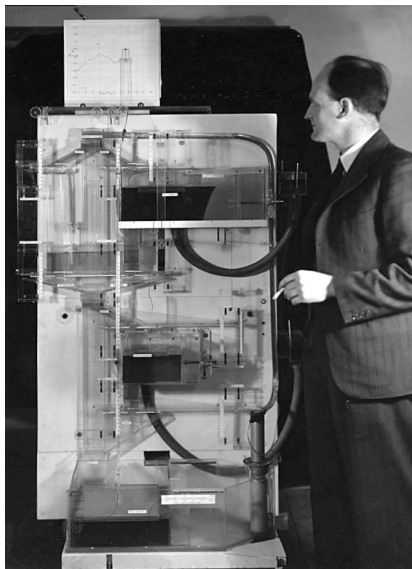- 2014 NVIDIA Geforce GTX Titan Z (8 TFLOPS per card 'under your desk')
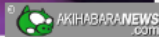
Figure: MONIAC.

Figure: MONIAC and Phillips.

Figure: China's Tianhe-2 (MilkyWay-2).
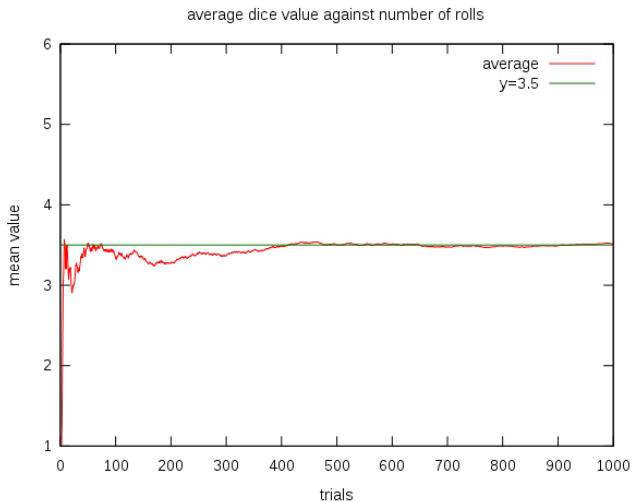
Simulation

Laws of Large Numbers

Figure: Illustration of law of large numbers.
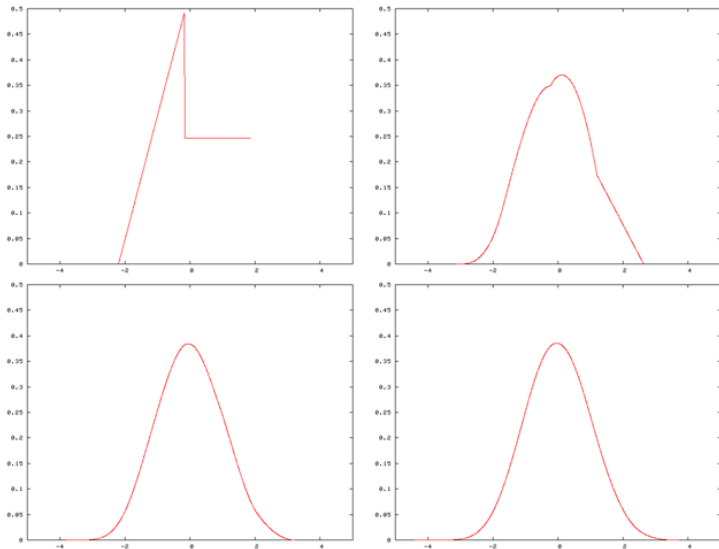
Simulation

Central Limit Theorems

Figure: Illustration of central limit theorem.

Graphical Programming

- 'Tapping the supercomputer under your desk: Solving dynamic equilibrium models with graphics processors' by Aldrich, Fernández-Villaverde, Gallant and Rubio-Ramírez (2011).
- `http://www.youtube.com/watch?v=2JjxgJcXVE0`.

Parallel Programming

- Limits on serial code (memory, CPU time).
- Split up task into parts that can be parallelised and parts that cannot.
- Example: 95% of task can be executed in parallel; even with an infinite number of processes on the parallel part, we still need 5% of the original time to execute the serial part.
- Amdahl's law: maximum speed up is given by

$$\frac{1}{S + \frac{1-S}{N}}$$

where S is proportion of code to be executed in serial and N is the number of processes in the parallel part.

Amdahl's Law
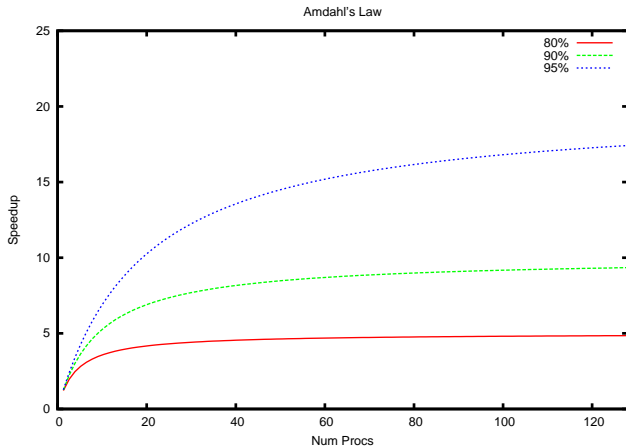
Parallel Programming

- This suggests no point in writing code for more than 8-16 processes. . . not true! Running on larger problems, often the serial part scales linearly but the parallel part scales with n^2 or n^3.
- By tackling larger problems, 95% parallel problem can become a 99% parallel problem and eventually a 99.9% parallel problem.
- $1 - S = 99.9\%$, $N = 1024$: Amdahl $\implies$ 506 speedup!
- Overhead.
- Low level parallelism [processor] (compiler) vs high level parallelism [interconnect] (programmer).
- OpenMP for shared memory (PC) vs MPI for distributed memory (cluster): work for C/C++/Fortran.

Comparing Programming Languages in Economics

```
http://economics.sas.upenn.edu/~jesusfv/comparison_
languages.pdf
```