The Econometrics of Uncertainty Shocks

Jesús Fernández-Villaverde

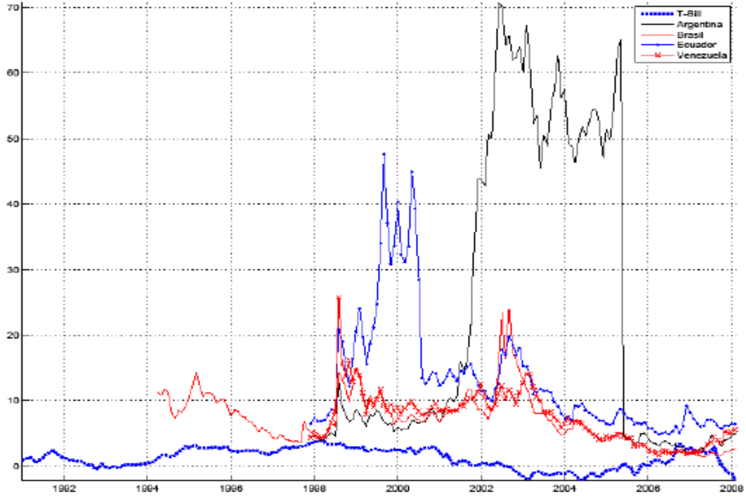
University of Pennsylvania

March 7, 2016

Several challenges

- How do we document the presence of time-varying uncertainty?
- How do we distinguish time-variation in the data and in expectations? Forecaster disagreement?
- How much of the uncertainty is exogenous or endogenous?
- How do we take DSGE models with time-varying uncertainty to the data?
    1. Likelihood function.
    2. Method of moments.
- Because of time limitations, I will focus on the first and last challenges.

Interest rates



A real life example

- Remember our decomposition of interest rates:

$$r_t = \underbrace{r}_{\text{mean}} + \underbrace{\varepsilon_{tb,t}}_{\text{T-Bill shocks}} + \underbrace{\varepsilon_{r,t}}_{\text{Spread shocks}}$$

- $\varepsilon_{tb,t}$ and $\varepsilon_{r,t}$ follow:

$$\varepsilon_{tb,t} = \rho_{tb}\varepsilon_{tb,t-1} + e^{\sigma_{tb,t}} u_{tb,t},\ u_{tb,t} \sim \mathcal{N}(0,1)$$

$$\varepsilon_{r,t} = \rho_{r}\varepsilon_{r,t-1} + e^{\sigma_{r,t}} u_{r,t},\ u_{r,t} \sim \mathcal{N}(0,1)$$

- $\sigma_{tb,t}$ and $\sigma_{r,t}$ follow:

$$\sigma_{tb,t} = \left(1 - \rho_{\sigma_{tb}}\right)\sigma_{tb} + \rho_{\sigma_{tb}}\sigma_{tb,t-1} + \eta_{tb} u_{\sigma_{tb},t},\ u_{\sigma_{tb},t} \sim \mathcal{N}(0,1)$$

$$\sigma_{r,t} = \left(1 - \rho_{\sigma_{r}}\right)\sigma_{r} + \rho_{\sigma_{r}}\sigma_{r,t-1} + \eta_{r} u_{\sigma_{r},t},\ u_{\sigma_{r},t} \sim \mathcal{N}(0,1)$$

Stochastic volatility

- Remember, as well, that we postulated a general process for stochastic volatility:

$$x_t = \rho x_{t-1} + \sigma_t \varepsilon_t,\ \ \varepsilon_t \sim \mathcal{N}(0,1).$$

and

$$\log \sigma_t = (1-\rho_\sigma)\log \sigma + \rho_\sigma \log \sigma_{t-1} + \left(1-\rho_\sigma^2\right)^{\frac{1}{2}} \eta u_t,\ \ u_t \sim \mathcal{N}(0,1).$$

- We discussed this was a concrete example of a richer class of specifications.

- Main point: non-linear structure.

State space representation I

- Define:

$$S_t = \begin{pmatrix} x_{t-1} \\ \sigma_t \end{pmatrix}$$

- Then, we have a transition equation:

$$\begin{pmatrix} x_t \\ \sigma_{t+1} \end{pmatrix} = f\left(\begin{pmatrix} x_t \\ \sigma_t \end{pmatrix}, \begin{pmatrix} \varepsilon_t \\ u_{t+1} \end{pmatrix}; \gamma\right)$$

where the first row of $f(\cdot)$ is:

$$x_t = \rho x_{t-1} + \sigma_t \varepsilon_t$$

and the second is

$$\log \sigma_{t+1} = (1-\rho_\sigma)\log \sigma + \rho_\sigma \log \sigma_t + \left(1-\rho_\sigma^2\right)^{\frac{1}{2}} \eta u_{t+1}$$

- The vector of parameters:

$$\gamma = (\rho, \rho_\sigma, \log \sigma, \eta)$$

State space representation II

- In more compact notation:

$$S_t = f\left(S_{t-1}, W_t; \gamma\right)$$

- We also have a trivial measurement equation:

$$Y_t = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_t \\ \sigma_{t+1} \end{pmatrix}$$

- In more general notation:

$$Y_t = g\left(S_t, V_t; \gamma\right)$$

- Note Markov structure.
- Note also how we can easily accommodate more general cases.

Shocks

- $\{W_t\}$ and $\{V_t\}$ are independent of each other.
- $\{W_t\}$ is known as process noise and $\{V_t\}$ as measurement noise.
- W_t and V_t have zero mean.
- No assumptions on the distribution beyond that.

Conditional densities

- From $S_t = f(S_{t-1}, W_t; \gamma)$, we can compute $p(S_t | S_{t-1}; \gamma)$.
- From $Y_t = g(S_t, V_t; \gamma)$, we can compute $p(Y_t | S_t; \gamma)$.
- From $S_t = f(S_{t-1}, W_t; \gamma)$ and $Y_t = g(S_t, V_t; \gamma)$, we have:

$$Y_t = g(f(S_{t-1}, W_t; \gamma), V_t; \gamma)$$

and hence we can compute $p(Y_t | S_{t-1}; \gamma)$.

Filtering, smoothing, and forecasting

- Filtering: we are concerned with what we have learned up to current observation.
- Smoothing: we are concerned with what we learn with the full sample.
- Forecasting: we are concerned with future realizations.

Goal of filtering I

- Compute conditional densities: $p\left(S_t|y^{t-1};\gamma\right)$ and $p\left(S_t|y^t;\gamma\right)$.

- Why?

  1. It allows probability statements regarding the situation of the system.

  2. Compute conditional moments: mean, $s_{t|t}$ and $s_{t|t-1}$, and variances $P_{t|t}$ and $P_{t|t-1}$.

  3. Other functions of the states. Examples of interest.

- Theoretical point: do the conditional densities exist?

Goals of filtering II

- Evaluate the likelihood function of the observables y^T at parameter values γ:

$$p\left(y^T;\gamma\right)$$

- Given the Markov structure of our state space representation:

$$p\left(y^T;\gamma\right) = p\left(y_1|\gamma\right)\prod_{t=2}^{T} p\left(y_t|y^{t-1};\gamma\right)$$

- Then:

$$p\left(y^T;\gamma\right) = \int p\left(y_1|s_1;\gamma\right) dS_1 \prod_{t=2}^{T}\int p\left(y_t|S_t;\gamma\right) p\left(S_t|y^{t-1};\gamma\right) dS_t$$

- Hence, knowledge of $\left\{p\left(S_t|y^{t-1};\gamma\right)\right\}_{t=1}^{T}$ and $p\left(S_1;\gamma\right)$ allow the evaluation of the likelihood of the model.

Two fundamental tools

1. Chapman-Kolmogorov equation:

$$p\left(S_t|y^{t-1};\gamma\right)=\int p\left(S_t|S_{t-1};\gamma\right)p\left(S_{t-1}|y^{t-1};\gamma\right)dS_{t-1}$$

2. Bayes' theorem:

$$p\left(S_t|y^{t};\gamma\right)=\frac{p\left(y_t|S_t;\gamma\right)p\left(S_t|y^{t-1};\gamma\right)}{p\left(y_t|y^{t-1};\gamma\right)}$$

where:

$$p\left(y_t|y^{t-1};\gamma\right)=\int p\left(y_t|S_t;\gamma\right)p\left(S_t|y^{t-1};\gamma\right)dS_t$$

Interpretation

- All filtering problems have two steps: prediction and update.

  1. Chapman-Kolmogorov equation is one-step ahead predictor.
  2. Bayes' theorem updates the conditional density of states given the new observation.

- We can think of those two equations as operators that map measures into measures.

Recursion for conditional distribution

- Combining the Chapman-Kolmogorov and the Bayes' theorem:

$$p\left(S_t|y^t;\gamma\right) = \frac{\int p\left(S_t|S_{t-1};\gamma\right) p\left(S_{t-1}|y^{t-1};\gamma\right) dS_{t-1}}{\int \left\{\int p\left(S_t|S_{t-1};\gamma\right) p\left(S_{t-1}|y^{t-1};\gamma\right) dS_{t-1}\right\} p\left(y_t|S_t;\gamma\right) dS_t} p\left(y_t|S_t;\gamma\right)$$

- To initiate that recursion, we only need a value for s_0 or $p\left(S_0;\gamma\right)$.

- Applying the Chapman-Kolmogorov equation once more, we get $\left\{p\left(S_t|y^{t-1};\gamma\right)\right\}_{t=1}^{T}$ to evaluate the likelihood function.

Initial conditions

- From previous discussion, we know that we need a value for s_1 or $p\left(S_1;\gamma\right)$.
- Stationary models: ergodic distribution.
- Non-stationary models: more complicated. Importance of transformations.
- Forgetting conditions.
- Non-contraction properties of the Bayes operator.

Smoothing

- We are interested on the distribution of the state conditional on all the observations, on $p\left(S_t|y^T;\gamma\right)$ and $p\left(y_t|y^T;\gamma\right)$.
- We compute:

$$p\left(S_t|y^T;\gamma\right) = p\left(S_t|y^t;\gamma\right)\int \frac{p\left(S_{t+1}|y^T;\gamma\right)p\left(S_{t+1}|S_t;\gamma\right)}{p\left(S_{t+1}|y^t;\gamma\right)}dS_{t+1}$$

a backward recursion that we initialize with $p\left(S_T|y^T;\gamma\right)$, $\left\{p\left(S_t|y^t;\gamma\right)\right\}_{t=1}^T$ and $\left\{p\left(S_t|y^{t-1};\gamma\right)\right\}_{t=1}^T$ we obtained from filtering.

Forecasting

- We apply the Chapman-Kolmogorov equation recursively, we can get $p\left(S_{t+j}|y^t;\gamma\right),\ j\geq 1$.

- Integrating recursively:

$$p\left(y_{l+1}|y^l;\gamma\right)=\int p\left(y_{l+1}|S_{l+1};\gamma\right)p\left(S_{l+1}|y^l;\gamma\right)dS_{l+1}$$

  from $t+1$ to $t+j$, we get $p\left(y_{t+j}|y^T;\gamma\right)$.

- Clearly smoothing and forecasting require to solve the filtering problem first!

Problem of filtering

- We have the recursion

$$p\left(S_t|y^t;\gamma\right) = \frac{\int p\left(S_t|S_{t-1};\gamma\right) p\left(S_{t-1}|y^{t-1};\gamma\right) dS_{t-1}}{\int \left\{\int p\left(S_t|S_{t-1};\gamma\right) p\left(S_{t-1}|y^{t-1};\gamma\right) dS_{t-1}\right\} p\left(y_t|S_t;\gamma\right) dS_t} p\left(y_t|S_t;\gamma\right)$$

- A lot of complicated and high dimensional integrals (plus the one involved in the likelihood).
- In general, we do not have closed form solution for them.
- *Translate*, *spread*, and *deform* (TSD) the conditional densities in ways that impossibilities to fit them within any known parametric family.

Exception

- There is one exception: linear and Gaussian case.
- Why? Because if the system is linear and Gaussian, all the conditional probabilities are also Gaussian.
- Linear and Gaussian state spaces models *translate* and *spread* the conditional distributions, but they do not *deform* them.
- For Gaussian distributions, we only need to track mean and variance (sufficient statistics).
- Kalman filter accomplishes this goal efficiently.

Nonlinear filtering

- Different approaches.
- Deterministic filtering:
  1. Kalman family.
  2. Grid-based filtering.
- Simulation filtering:
  1. McMc.
  2. Particle filtering.

Particle filtering I

- Remember,

1. Transition equation:

$$S_t = f\left(S_{t-1}, W_t; \gamma\right)$$

2. Measurement equation:

$$Y_t = g\left(S_t, V_t; \gamma\right)$$

Particle filtering II

- Some Assumptions:

1. We can partition $\{W_t\}$ into two independent sequences $\{W_{1,t}\}$ and $\{W_{2,t}\}$, s.t. $W_t = (W_{1,t}, W_{2,t})$ and $\dim(W_{2,t}) + \dim(V_t) \geq \dim(Y_t)$.

2. We can always evaluate the conditional densities $p\left(y_t | W_1^t, y^{t-1}, S_0; \gamma\right)$.

3. The model assigns positive probability to the data.

Rewriting the likelihood function

- Evaluate the likelihood function of the a sequence of realizations of the observable y^T at a particular parameter value γ:

$$p\left(y^T;\gamma\right)$$

- We factorize it as (careful with initial condition!):

$$p\left(y^T;\gamma\right)=\prod_{t=1}^{T}p\left(y_t|y^{t-1};\gamma\right)$$

$$=\prod_{t=1}^{T}\int\int p\left(y_t|W_1^t,y^{t-1},S_0;\gamma\right)p\left(W_1^t,S_0|y^{t-1};\gamma\right)dW_1^t dS_0$$

A law of large numbers

If $\left\{ \left\{ s_0^{t|t-1,i}, w_1^{t|t-1,i} \right\}_{i=1}^{N} \right\}_{t=1}^{T}$ N i.i.d. draws from $\left\{ p\left(W_1^t, S_0 | y^{t-1}; \gamma \right) \right\}_{t=1}^{T}$, then:

$$p\left(y^T; \gamma \right) \simeq \prod_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} p\left(y_t | w_1^{t|t-1,i}, y^{t-1}, s_0^{t|t-1,i}; \gamma \right)$$

The problem of evaluating the likelihood is equivalent to the problem of drawing from

$$\left\{ p\left(W_1^t, S_0 | y^{t-1}; \gamma \right) \right\}_{t=1}^{T}$$

Introducing particles I

- $\left\{ s_0^{t-1,i}, w_1^{t-1,i} \right\}_{i=1}^{N}$ N i.i.d. draws from $p\left(W_1^{t-1}, S_0 | y^{t-1}; \gamma \right)$.
- Each $s_0^{t-1,i}, w_1^{t-1,i}$ is a *particle* and $\left\{ s_0^{t-1,i}, w_1^{t-1,i} \right\}_{i=1}^{N}$ a *swarm of particles*.
- $\left\{ s_0^{t|t-1,i}, w_1^{t|t-1,i} \right\}_{i=1}^{N}$ N i.i.d. draws from $p\left(W_1^{t}, S_0 | y^{t-1}; \gamma \right)$.

Introducing particles II

- Each $s_0^{t|t-1,i}, w_1^{t|t-1,i}$ is a *proposed particle* and $\left\{ s_0^{t|t-1,i}, w_1^{t|t-1,i} \right\}_{i=1}^{N}$ a *swarm of proposed particles*.
- Weights:

$$q_t^i = \frac{p\left(y_t | w_1^{t|t-1,i}, y^{t-1}, s_0^{t|t-1,i}; \gamma\right)}{\sum_{i=1}^{N} p\left(y_t | w_1^{t|t-1,i}, y^{t-1}, s_0^{t|t-1,i}; \gamma\right)}$$

A proposition

Theorem

Let $\left\{\widetilde{s}_0^i, \widetilde{w}_1^i\right\}_{i=1}^N$ be a draw with replacement from $\left\{s_0^{t|t-1,i}, w_1^{t|t-1,i}\right\}_{i=1}^N$ and probabilities q_t^i. Then $\left\{\widetilde{s}_0^i, \widetilde{w}_1^i\right\}_{i=1}^N$ is a draw from $p\left(W_1^t, S_0|y^t;\gamma\right)$.

- Importance:

  1. It shows how a draw $\left\{s_0^{t|t-1,i}, w_1^{t|t-1,i}\right\}_{i=1}^N$ from $p\left(W_1^t, S_0|y^{t-1};\gamma\right)$ can be used to draw $\left\{s_0^{t,i}, w_1^{t,i}\right\}_{i=1}^N$ from $p\left(W_1^t, S_0|y^t;\gamma\right)$.

  2. With a draw $\left\{s_0^{t,i}, w_1^{t,i}\right\}_{i=1}^N$ from $p\left(W_1^t, S_0|y^t;\gamma\right)$ we can use $p\left(W_{1,t+1};\gamma\right)$ to get a draw $\left\{s_0^{t+1|t,i}, w_1^{t+1|t,i}\right\}_{i=1}^N$ and iterate the procedure.

Algorithm

Step 0, Initialization: Set $t \rightsquigarrow 1$ and set $p\left(W_1^{t-1}, S_0 | y^{t-1}; \gamma\right) = p\left(S_0; \gamma\right)$.

Step 1, Prediction: Sample N values $\left\{ s_0^{t|t-1,i}, w_1^{t|t-1,i} \right\}_{i=1}^{N}$ from the density $p\left(W_1^t, S_0 | y^{t-1}; \gamma\right) = p\left(W_{1,t}; \gamma\right) p\left(W_1^{t-1}, S_0 | y^{t-1}; \gamma\right)$.

Step 2, Weighting: Assign to each draw $s_0^{t|t-1,i}, w_1^{t|t-1,i}$ the weight q_t^i.

Step 3, Sampling: Draw $\left\{ s_0^{t,i}, w_1^{t,i} \right\}_{i=1}^{N}$ with rep. from $\left\{ s_0^{t|t-1,i}, w_1^{t|t-1,i} \right\}_{i=1}^{N}$ with probabilities $\left\{ q_t^i \right\}_{i=1}^{N}$. If $t < T$ set $t \rightsquigarrow t+1$ and go to step 1. Otherwise go to step 4.

Step 4, Likelihood: Use $\left\{ \left\{ s_0^{t|t-1,i}, w_1^{t|t-1,i} \right\}_{i=1}^{N} \right\}_{t=1}^{T}$ to compute:

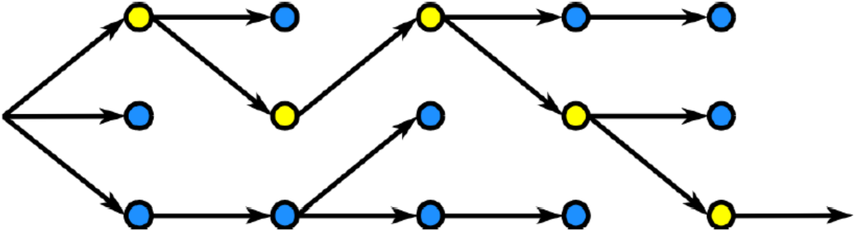
$$p\left(y^T; \gamma\right) \simeq \prod_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} p\left(y_t | w_1^{t|t-1,i}, y^{t-1}, s_0^{t|t-1,i}; \gamma\right)$$

Evaluating a Particle filter

- We just saw a plain vanilla particle filter.
- How well does it work in real life?
- Is it feasible to implement in large models?

Why did we resample?

- We could have not resampled and just used the weights as you would have done in *importance sampling* (this is known as *sequential importance sampling*).
- Most weights go to zero.
- But resampling impoverish the swarm.
- Eventually, this becomes a problem.



Why did we resample?

- Effective Sample Size:

$$ESS_t = \frac{1}{\left[\sum_{i=1}^{N} p\left(y_t \middle| w_1^{t|t-1,i}, y^{t-1}, s_0^{t|t-1,i}; \gamma\right)\right]^2}$$

- Alternatives:

1. Stratified resampling (Kitagawa, 1996): optimal in terms of variance.
2. Adaptive resampling.

Simpler notation

- To simplify notation:

  1. Let me write the conditional distributions in terms of the current state (instead of the innovations and the initial state).
  2. Let me forget about the special notation required for period 1 ($y^{t-1} = \varnothing$).

- Then, the evaluation of the likelihood is just:

$$p\left(y^{T};\gamma\right)=\prod_{t=1}^{T}\int\int p\left(y_{t}|S^{t};\gamma\right)p\left(S_{t}|y^{t-1};\gamma\right)dS_{t}$$

- Thus, we are looking for $\left\{\left\{s^{t|t-1,i}\right\}_{i=1}^{N}\right\}_{t=1}^{T}$ N i.i.d. draws from $\left\{p\left(S_{t}|y^{t-1};\gamma\right)\right\}_{t=1}^{T}$.

Improving our Particle filter I

- Remember what we did:

  1. We draw from
  $$s^{t|t-1,i} \sim p\left(S_t | s^{t-1,i}; \gamma\right)$$
  2. We resample them with:
  $$q_t^i = \frac{p\left(y_t | s^{t|t-1,i}, y^{t-1}; \gamma\right)}{\sum_{i=1}^N p\left(y_t | s^{t|t-1,i}, y^{t-1}; \gamma\right)}$$

- But, what if I can draw instead from $s^{t|t-1,i} \sim q\left(S_t | s^{t-1,i}, y_t; \gamma\right)$?

- Intuition.

Improving our Particle filter II

- New weights:

$$q_t^i = \frac{p\left(y_t | s^{t|t-1,i}, y^{t-1}; \gamma\right) \frac{p\left(S_t | s^{t-1,i}; \gamma\right)}{q\left(S_t | s^{t-1,i}, y_t; \gamma\right)}}{\sum_{i=1}^N p\left(y_t | s^{t|t-1,i}, y^{t-1}; \gamma\right) \frac{p\left(S_t | s^{t-1,i}; \gamma\right)}{q\left(S_t | s^{t-1,i}, y_t; \gamma\right)}}$$

- Clearly, if

$$q\left(S_t | s^{t-1,i}, y_t; \gamma\right) = p\left(S_t | s^{t-1,i}; \gamma\right)$$

  we get back our basic Particle Filter.

- How do we create the proposal $q\left(S_t | s^{t-1,i}, y_t; \gamma\right)$?

  1. Linearized model.
  2. Unscented Kalman filter.
  3. Information from the problem.

Improving our Particle filter III

- Auxiliary Particle Filter: Pitt and Shephard (1999).
- Imagine we can compute either

$$p\left(y_{t+1}|s^{t,i}\right)$$

or

$$\widetilde{p}\left(y_{t+1}|s^{t,i}\right)$$

- Then:

$$q_t^i = \frac{\widetilde{p}\left(y_{t+1}|s^{t,i}\right) p\left(y_t|s^{t|t-1,i}, y^{t-1}; \gamma\right) \frac{p\left(S_t|s^{t-1,i};\gamma\right)}{q\left(S_t|s^{t-1,i},y_t;\gamma\right)}}{\sum_{i=1}^N \widetilde{p}\left(y_{t+1}|s^{t,i}\right) p\left(y_t|s^{t|t-1,i}, y^{t-1}; \gamma\right) \frac{p\left(S_t|s^{t-1,i};\gamma\right)}{q\left(S_t|s^{t-1,i},y_t;\gamma\right)}}$$

- Auxiliary Particle Filter tends to work well when we have fat tail...
- ...but it can temperamental.

Improving our Particle filter IV

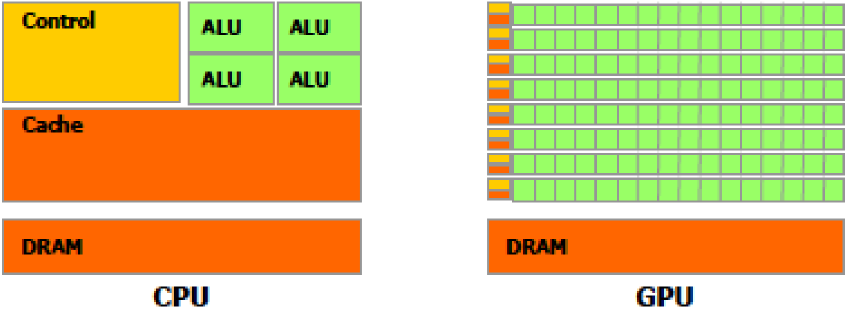
- Resample-Move.
- Blocking.
- Many others.
- A Tutorial on Particle Filtering and Smoothing: Fifteen years later, by Doucet and Johansen (2012)

Nesting it in a McMc

- Fernández-Villaverde and Rubio Ramírez (2007) and Flury and Shepard (2008).
- You nest the Particle filter inside an otherwise standard McMc.
- Two caveats:
  1. Lack of differentiability of the Particle filter.
  2. Random numbers constant to avoid chatter and to be able to swap operators.

Parallel programming

- Why?
- Divide and conquer.
- Shared and distributed memory.
- Main approaches:
  1. OpenMP.
  2. MPI.
  3. GPU programming: CUDA and OpenCL.


Control
ALU
ALU
ALU
ALU
Cache
DRAM
CPU
DRAM
GPU

Tools

- In `Matlab`: `parallel toolboox`.
- In `R`: `package parallel`.
- In `Julia`: `built-in procedures`.
- In `Mathematica`: `parallel computing tools`.
- GPUs: `ArrayFire`.

Parallel Particle filter

- Simplest strategy: generating and evaluating draws.
- A temptation: multiple swarms.
- How to nest with a McMc?

A problem

- Basic Random Walk Metropolis Hastings is difficult to parallelize.
- Why? Proposal draw θ_{i+1}^{*} depends on θ_i.
- Inherently serial procedure.
- Assume, instead, that we have N processors.
- Possible solutions:
  1. Run parallel chains.
  2. Independence sampling.
  3. Pre-fetching.

Multiple chains

- We run N chains, one in each processor.
- We merge them at the end.
- It goes against the principle of one, large chain.
- But it may works well when the burn-in period is small.
- If the burn-in is large or the chain has subtle convergence issues, it results in waste of time and bad performance.

Independence sampling

- We generate N proposals $\widetilde{\theta}^{j}_{i+1}$ from an independent distribution.
- We evaluate the posterior from each proposal in a different processor.
- We do N Metropolis steps with each proposal.
- Advantage: extremely simple to code, nearly linear speed up.
- Disadvantage: independence sampling is very inefficient.
  Solution⇒design a better proposal density.

Prefetching I

- Proposed by Brockwell (2006).
- Idea: we can compute the relevant posteriors several periods in advance.
- Set superindex 1 for rejection and 2 for acceptance.
- Advantage: if we reject a draw, we have already evaluated the next step.
- Disadvantage: wasteful. More generally, you can show that the speed up will converge only to $\log_2 N$.

Prefetching II

1. Assume we are at θ_i.

2. We draw 2 paths for iteration $i+1$, $\left\{\widetilde{\theta}_{i+1}^{1} = \theta_i, \widetilde{\theta}_{i+1}^{2} \sim g\left(\theta_i\right)\right\}$.

3. We draw 4 paths for iteration $i+2$
$\left\{\widetilde{\theta}_{i+2}^{11} = \widetilde{\theta}_{i+1}^{1}, \widetilde{\theta}_{i+2}^{12} \sim g\left(\widetilde{\theta}_{i+1}^{1}\right), \widetilde{\theta}_{i+2}^{21} = \widetilde{\theta}_{i+1}^{2}, \widetilde{\theta}_{i+2}^{22} \sim g\left(\widetilde{\theta}_{i+1}^{2}\right)\right\}$.

4. We iterate h steps, until we have $N = 2^h$ possible sequences.

5. We evaluate each of the posteriors $p\left(\widetilde{\theta}_{i+h}^{1,\dots,1}\right), \dots, p\left(\widetilde{\theta}_{i+h}^{2,\dots,2}\right)$ in each of the N processors.

6. We do a MH in each step of the path using the previous posteriors (note that any intermediate posterior is the same as the corresponding final draw where all the following children are "rejections").

A simpler prefetching algorithm I

- Algorithm for N processors, where N is small.
- Given θ_i:
  1. We draw N $\left\{\widetilde{\theta}_{i+1,1}, ..., \widetilde{\theta}_{i+1,N}\right\}$ and we evaluate the posteriors $p\left(\widetilde{\theta}_{i+1,1}\right), ..., p\left(\widetilde{\theta}_{i+1,N}\right)$.
  2. We evaluate the first proposal, $\widetilde{\theta}_{i+1,1}$:
     1. If accepted, we disregard $\theta_{i+1,2}, ..., \theta_{i+1,2}$.
     2. If rejected, we make $\theta_{i+1} = \theta_i$ and $\widetilde{\theta}_{i+2,1} = \widetilde{\theta}_{i+1,2}$ is our new proposal for $i+2$.
     3. We continue down the list of N proposals until we accept one.
- Advantage: if we reject a draw, we have already evaluated the next step.

A simpler prefetching algorithm II

- Imagine we have a PC with $N = 4$ processors.
- Well gauged acceptance for a normal SV model$\approx 20\% - 25\%$. Let me fix it to 25% for simplicity.
- Then:
  1. $P(accepting\ 1^{st}\ draw) = 0.25$. We advance 1 step.
  2. $P(accepting\ 2^{nd}\ draw) = 0.75 * 0.25$. We advance 2 steps.
  3. $P(accepting\ 3^{tr}\ draw) = 0.75^2 * 0.25$. We advance 3 steps.
  4. $P(accepting\ 4^{th}\ draw) = 0.75^3 * 0.25$. We advance 4 steps.
  5. $P(not\ accepting\ any\ draw) = 0.75^{*4}$. We advance 4 steps.
- Therefore, expected numbers of steps advanced in the chain:

$$1 * 0.25 + 2 * 0.75 * 0.25 + 3 * 0.75^{*2} * 0.25 + 4 * 0.75^3 = 2.7344$$

First-order approximation

- Remember that the first-order approximation of a canonical RBC model without persistence in productivity shocks:

$$\widehat{k}_{t+1} = a_1 \widehat{k}_t + a_2 \varepsilon_t, \ \ \varepsilon_t \sim \mathcal{N}(0,1)$$

- Then:

$$\begin{aligned} \widehat{k}_{t+1} &= a_1 \left(a_1 \widehat{k}_{t-1} + a_2 \varepsilon_{t-1} \right) + a_2 \varepsilon_t \\ &= a_1^2 \widehat{k}_{t-1} + a_1 a_2 \varepsilon_{t-1} + a_2 \varepsilon_t \end{aligned}$$

- Since $a_1 < 1$ and assuming $\widehat{k}_0 = 0$

$$\widehat{k}_{t+1} = a_2 \sum_{j=0}^{t} a_1^j \varepsilon_{t-j}$$

which is a well-understood MA system.

Higher-order approximations

- Second-order approximation:

$$\widehat{k}_{t+1} = a_0 + a_1 \widehat{k}_t + a_2 \varepsilon_t + a_3 \widehat{k}_t^2 + a_4 \varepsilon_t^2 + a_5 \widehat{k}_t \varepsilon_t, \ \ \varepsilon_t \sim \mathcal{N}(0,1)$$

- Then:

$$\begin{aligned} \widehat{k}_{t+1} \quad = \quad & a_0 + a_1 \left(a_0 + a_1 \widehat{k}_t + a_2 \varepsilon_t + a_3 \widehat{k}_t^2 + a_4 \varepsilon_t^2 + a_5 \widehat{k}_t \varepsilon_t \right) + a_2 \varepsilon_t \\ & + a_3 \left(a_0 + a_1 \widehat{k}_t + a_2 \varepsilon_t + a_3 \widehat{k}_t^2 + a_4 \varepsilon_t^2 + a_5 \widehat{k}_t \varepsilon_t \right)^2 + a_4 \varepsilon_t^2 \\ & + a_5 \left(a_0 + a_1 \widehat{k}_t + a_2 \varepsilon_t + a_3 \widehat{k}_t^2 + a_4 \varepsilon_t^2 + a_5 \widehat{k}_t \varepsilon_t \right) \varepsilon_t \end{aligned}$$

- We have terms in $\widehat{k}_t^3$ and $\widehat{k}_t^4$.

Problem

- For a large realization of ε_t, the terms in $\widehat{k}_t^3$ and $\widehat{k}_t^4$ make the system explode.

- This will happen as soon as we have a large simulation⇒ no unconditional moments would exist based on this approximation.

- This is true even when the corresponding linear approximation is stable.

- Then:

  1. How do you calibrate? (translation, spread, and deformation).
  2. How do you implement GMM or SMM?
  3. Asymptotics?

A solution

- For second-order approximations, Kim *et al.* (2008): pruning.
- Idea:

$$\begin{aligned}\widehat{k}_{t+1} &= a_0 + a_1\left(a_0 + a_1\widehat{k}_t + a_2\varepsilon_t + a_3\widehat{k}_t^2 + a_4\varepsilon_t^2 + a_5\widehat{k}_t\varepsilon_t\right) + a_2\varepsilon_t \\ &+ a_3\left(a_0 + a_1\widehat{k}_t + a_2\varepsilon_t + a_3\widehat{k}_t^2 + a_4\varepsilon_t^2 + a_5\widehat{k}_t\varepsilon_t\right)^2 + a_4\varepsilon_t^2 \\ &+ a_5\left(a_0 + a_1\widehat{k}_t + a_2\varepsilon_t + a_3\widehat{k}_t^2 + a_4\varepsilon_t^2 + a_5\widehat{k}_t\varepsilon_t\right)\varepsilon_t\end{aligned}$$

- We omit terms raised to powers higher than 2.
- Pruned approximation does not explode.

What do we do?

- Build a pruned state-space system.
- Apply pruning to an approximation of any arbitrary order.
- Prove that first and second unconditional moments exist.
- Closed-form expressions for first and second unconditional moments and IRFs.
- Conditions for the existence of some higher unconditional moments, such as skewness and kurtosis.
- Apply to a New Keynesian model with EZ preferences.
- Software available for distribution.

Practical consequences

1. GMM and IRF-matching can be implemented without simulation.

2. First and second unconditional moments or IRFs can be computed in a trivial amount of time for medium-sized DSGE models approximated up to third-order.

3. Use the unconditional moment conditions in optimal GMM estimation to build a limited information likelihood function for Bayesian inference (Kim, 2002).

4. Foundation for indirect inference as in Smith (1993) and SMM as in Duffie and Singleton (1993).

5. Calibration.

Dynamic models and state-space representations

- Dynamic model:

$$\mathbf{x}_{t+1} = \mathbf{h}\left(\mathbf{x}_t, \sigma\right) + \sigma \boldsymbol{\eta} \boldsymbol{\epsilon}_{t+1},\ \boldsymbol{\epsilon}_{t+1} \sim IID\left(\mathbf{0}, \mathbf{I}\right)$$
$$\mathbf{y}_t = \mathbf{g}\left(\mathbf{x}_t, \sigma\right)$$

- Comparison with our previous structure.

- Again, general framework (augmented state vector).

The state-space system I

- Perturbation methods approximate $\mathbf{h}\left(\mathbf{x}_t, \sigma\right)$ and $\mathbf{g}\left(\mathbf{x}_t, \sigma\right)$ with Taylor-series expansions around $\mathbf{x}_{ss} = \sigma = 0$.

- A first-order approximated state-space system replaces $\mathbf{g}\left(\mathbf{x}_t, \sigma\right)$ and $\mathbf{h}\left(\mathbf{x}_t, \sigma\right)$ with $\mathbf{g}_{\mathbf{x}}\mathbf{x}_t$ and $\mathbf{h}_{\mathbf{x}}\mathbf{x}_t$.

- If $\forall \bmod \left(eig\left(\mathbf{h}_{\mathbf{x}}\right)\right) < 1$, the approximation fluctuates around the steady state (also its mean value).

- Thus, easy to calibrate the model based on first and second moments or to estimate it using Bayesian methods, MLE, GMM, SMM, etc.

The state-space system II

- We can replace $\mathbf{g}\left(\mathbf{x}_t, \sigma\right)$ and $\mathbf{h}\left(\mathbf{x}_t, \sigma\right)$ with their higher-order Taylor-series expansions.
- However, the approximated state-space system cannot, in general, be shown to have any finite moments.
- Also, it often displays explosive dynamics.
- This occurs even with simple versions of the New Keynesian model.
- Hence, it is difficult to use the approximated state-space system to calibrate or to estimate the parameters of the model.

The pruning method: second-order approximation I

- Partition states:

$$\left[\begin{array}{cc} \left(\mathbf{x}_t^f\right)' & \left(\mathbf{x}_t^s\right)' \end{array}\right]$$

- Original state-space representation:

$$\mathbf{x}_{t+1}^{(2)} = \mathbf{h}_{\mathbf{x}}\left(\mathbf{x}_t^f + \mathbf{x}_t^s\right) + \frac{1}{2}\mathbf{H}_{\mathbf{xx}}\left(\left(\mathbf{x}_t^f + \mathbf{x}_t^s\right) \otimes \left(\mathbf{x}_t^f + \mathbf{x}_t^s\right)\right) + \frac{1}{2}\mathbf{h}_{\sigma\sigma}\sigma^2 + \sigma\boldsymbol{\eta}\boldsymbol{\epsilon}_{t+1}$$

$$\mathbf{y}_t^{(2)} = \mathbf{g}_{\mathbf{x}}\mathbf{x}_t^{(2)} + \frac{1}{2}\mathbf{G}_{\mathbf{xx}}\left(\mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(2)}\right) + \frac{1}{2}\mathbf{g}_{\sigma\sigma}\sigma^2$$

The pruning method: second-order approximation II

- New state-space representation:

$$\begin{aligned}\mathbf{x}_{t+1}^{f} &= \mathbf{h}_{\mathbf{x}}\mathbf{x}_{t}^{f} + \sigma\boldsymbol{\eta}\boldsymbol{\epsilon}_{t+1}\\ \mathbf{x}_{t+1}^{s} &= \mathbf{h}_{\mathbf{x}}\mathbf{x}_{t}^{s} + \frac{1}{2}\mathbf{H}_{\mathbf{xx}}\left(\mathbf{x}_{t}^{f}\otimes\mathbf{x}_{t}^{f}\right) + \frac{1}{2}\mathbf{h}_{\sigma\sigma}\sigma^{2}\\ \mathbf{y}_{t}^{f} &= \mathbf{g}_{\mathbf{x}}\mathbf{x}_{t}^{f}\\ \mathbf{y}_{t}^{s} &= \mathbf{g}_{\mathbf{x}}\left(\mathbf{x}_{t}^{f}+\mathbf{x}_{t}^{s}\right) + \frac{1}{2}\mathbf{G}_{\mathbf{xx}}\left(\mathbf{x}_{t}^{f}\otimes\mathbf{x}_{t}^{f}\right) + \frac{1}{2}\mathbf{g}_{\sigma\sigma}\sigma^{2}\end{aligned}$$

- All variables are second-order polynomials of the innovations.

The pruning method: third-order approximation I

- Partition states:

$$\left[\begin{array}{ccc} \left(\mathbf{x}_t^f\right)' & \left(\mathbf{x}_t^s\right)' & \left(\mathbf{x}_t^{rd}\right)' \end{array} \right]$$

- Original state-space representation:

$$\begin{aligned} \mathbf{x}_{t+1}^{(3)} &= \mathbf{h}_{\mathbf{x}}\mathbf{x}_t^{(3)} + \frac{1}{2}\mathbf{H}_{\mathbf{xx}}\left(\mathbf{x}_t^{(3)} \otimes \mathbf{x}_t^{(3)}\right) + \frac{1}{6}\mathbf{H}_{\mathbf{xxx}}\left(\mathbf{x}_t^{(3)} \otimes \mathbf{x}_t^{(3)} \otimes \mathbf{x}_t^{(3)}\right) \\ &+ \frac{1}{2}\mathbf{h}_{\sigma\sigma}\sigma^2 + \frac{3}{6}\mathbf{h}_{\sigma\sigma\mathbf{x}}\sigma^2\mathbf{x}_t^{(3)} + \frac{1}{6}\mathbf{h}_{\sigma\sigma\sigma}\sigma^3 + \sigma\boldsymbol{\eta}\boldsymbol{\epsilon}_{t+1} \\ \mathbf{y}_t^{(3)} &= \mathbf{g}_{\mathbf{x}}\mathbf{x}_t^{(3)} + \frac{1}{2}\mathbf{G}_{\mathbf{xx}}\left(\mathbf{x}_t^{(3)} \otimes \mathbf{x}_t^{(3)}\right) + \frac{1}{6}\mathbf{G}_{\mathbf{xxx}}\left(\mathbf{x}_t^{(3)} \otimes \mathbf{x}_t^{(3)} \otimes \mathbf{x}_t^{(3)}\right) \\ &+ \frac{1}{2}\mathbf{g}_{\sigma\sigma}\sigma^2 + \frac{3}{6}\mathbf{g}_{\sigma\sigma\mathbf{x}}\sigma^2\mathbf{x}_t^{(3)} + \frac{1}{6}\mathbf{g}_{\sigma\sigma\sigma}\sigma^3 \end{aligned}$$

The pruning method: third-order approximation II

- New state-space representation:

$$
\begin{aligned}
& \text{Second-order pruned state-space representation}+ \\
\mathbf{x}_{t+1}^{rd} &= \mathbf{h}_{\mathbf{x}}\mathbf{x}_t^{rd} + \mathbf{H}_{\mathbf{xx}}\left(\mathbf{x}_t^f \otimes \mathbf{x}_t^s\right) + \frac{1}{6}\mathbf{H}_{\mathbf{xxx}}\left(\mathbf{x}_t^f \otimes \mathbf{x}_t^f \otimes \mathbf{x}^f\right) \\
&\quad + \frac{3}{6}\mathbf{h}_{\sigma\sigma\mathbf{x}}\sigma^2\mathbf{x}_t^f + \frac{1}{6}\mathbf{h}_{\sigma\sigma\sigma}\sigma^3 \\
\mathbf{y}_t^{rd} &= \mathbf{g}_{\mathbf{x}}\left(\mathbf{x}_t^f + \mathbf{x}_t^s + \mathbf{x}_t^{rd}\right) + \frac{1}{2}\mathbf{G}_{\mathbf{xx}}\left(\left(\mathbf{x}_t^f \otimes \mathbf{x}_t^f\right) + 2\left(\mathbf{x}_t^f \otimes \mathbf{x}_t^s\right)\right) \\
&\quad + \frac{1}{6}\mathbf{G}_{\mathbf{xxx}}\left(\mathbf{x}_t^f \otimes \mathbf{x}_t^f \otimes \mathbf{x}_t^f\right) + \frac{1}{2}\mathbf{g}_{\sigma\sigma}\sigma^2 + \frac{3}{6}\mathbf{g}_{\sigma\sigma\mathbf{x}}\sigma^2\mathbf{x}_t^f + \frac{1}{6}\mathbf{g}_{\sigma\sigma\sigma}\sigma^3
\end{aligned}
$$

- All variables are third-order polynomials of the innovations.

Higher-order approximations

- We can generalize previous steps:

1. Decompose the state variables into first-, second-, ... , and kth-order effects.

2. Set up laws of motions for the state variables capturing only first-, second-, ... , and kth-order effects.

3. Construct the expression for control variables by preserving only effects up to kth-order.

Statistical properties: second-order approximation I

Theorem

If $\forall \bmod (eig(\mathbf{h_x})) < 1$ *and* ϵ_{t+1} *has finite fourth moments, the pruned state-space system has finite first and second moments.*

Theorem

If $\forall \bmod (eig(\mathbf{h_x})) < 1$ *and* ϵ_{t+1} *has finite sixth and eighth moments, the pruned state-space system has finite third and fourth moments.*

Statistical properties: second-order approximation II

- We introduce the vectors

$$\mathbf{z}_t^{(2)} \equiv \left[\begin{array}{ccc} \left(\mathbf{x}_t^f\right)' & \left(\mathbf{x}_t^s\right)' & \left(\mathbf{x}_t^f \otimes \mathbf{x}_t^f\right)' \end{array} \right]'$$

$$\boldsymbol{\xi}_{t+1}^{(2)} \equiv \left[\begin{array}{c} \boldsymbol{\epsilon}_{t+1} \\ \boldsymbol{\epsilon}_{t+1} \otimes \boldsymbol{\epsilon}_{t+1} - vec\left(\mathbf{I}_{n_e}\right) \\ \boldsymbol{\epsilon}_{t+1} \otimes \mathbf{x}_t^f \\ \mathbf{x}_t^f \otimes \boldsymbol{\epsilon}_{t+1} \end{array} \right]$$

- First moment:

$$\mathbb{E}\left[\mathbf{x}_t^{(2)}\right] = \underbrace{\mathbb{E}\left[\mathbf{x}_t^f\right]}_{=0} + \underbrace{\mathbb{E}\left[\mathbf{x}_t^s\right]}_{\neq 0}$$

$$\mathbb{E}\left[\mathbf{x}_t^s\right] = \left(\mathbf{I} - \mathbf{h}_{\mathbf{x}}\right)^{-1} \left(\frac{1}{2}\mathbf{H}_{\mathbf{xx}} \left(\mathbf{I} - \mathbf{h}_{\mathbf{x}} \otimes \mathbf{h}_{\mathbf{x}}\right)^{-1} \left(\sigma\boldsymbol{\eta} \otimes \sigma\boldsymbol{\eta}\right) vec\left(\mathbf{I}_{n_e}\right) + \frac{1}{2}\mathbf{h}_{\sigma\sigma}\sigma^2 \right)$$

$$\mathbb{E}\left[\mathbf{y}_t^s\right] = \mathbf{C}^{(2)}\mathbb{E}\left[\mathbf{z}_t^{(2)}\right] + \mathbf{d}^{(2)}$$

Statistical properties: second-order approximation III

- Second moment:

$$\mathbb{V}\left(\mathbf{z}_t^{(2)}\right) = \mathbf{A}^{(2)}\mathbb{V}\left(\mathbf{z}_t^{(2)}\right)\left(\mathbf{A}^{(2)}\right)' + \mathbf{B}^{(2)}\mathbb{V}\left(\boldsymbol{\xi}_t^{(2)}\right)\left(\mathbf{B}^{(2)}\right)'$$

$$Cov\left(\mathbf{z}_{t+l}^{(2)}, \mathbf{z}_t^{(2)}\right) = \left(\mathbf{A}^{(2)}\right)^l \mathbb{V}\left(\mathbf{z}_t^{(2)}\right) \quad \text{for } l = 1, 2, 3, \ldots$$

$$\mathbb{V}\left[\mathbf{x}_t^{(2)}\right] = \mathbb{V}\left(\mathbf{x}_t^f\right) + \mathbb{V}\left(\mathbf{x}_t^s\right) + Cov\left(\mathbf{x}_t^f, \mathbf{x}_t^s\right) + Cov\left(\mathbf{x}_t^s, \mathbf{x}_t^f\right)$$

$$\mathbb{V}\left[\mathbf{y}_t^s\right] = \mathbf{C}^{(2)}\mathbb{V}\left[\mathbf{z}_t\right]\left(\mathbf{C}^{(2)}\right)'$$

$$Cov\left(\mathbf{y}_t^s, \mathbf{y}_{t+l}^s\right) = \mathbf{C}^{(2)} Cov\left(\mathbf{z}_{t+l}^{(2)}, \mathbf{z}_t^{(2)}\right)\left(\mathbf{C}^{(2)}\right)' \quad \text{for } l = 1, 2, 3, \ldots$$

where we solve for $\mathbb{V}\left(\mathbf{z}_t^{(2)}\right)$ by standard methods for discrete Lyapunov equations.

Statistical properties: second-order approximation IV

- Generalized impulse response function (GIRF): Koop *et al.* (1996)

$$GIRF_{\mathbf{var}}\left(l, \nu, \mathbf{w}_t\right) = \mathbb{E}\left[\mathbf{var}_{t+l} \middle| \mathbf{w}_t, \epsilon_{t+1} = \nu\right] - \mathbb{E}\left[\mathbf{var}_{t+l} \middle| \mathbf{w}_t\right]$$

- Importance in models with volatility shocks.

Statistical properties: third-order approximation

Theorem

If $\forall \bmod \left(eig \left(\mathbf{h_x} \right) \right) < 1$ *and* ϵ_{t+1} *has finite sixth moments, the pruned state-space system has finite first and second moments.*

Theorem

If $\forall \bmod \left(eig \left(\mathbf{h_x} \right) \right) < 1$ *and* ϵ_{t+1} *has finite ninth and twelfth moments, the pruned state-space system has finite third and fourth moments.*

- Similar (but long!!!!!) formulae for first and second moments and IRFs.