Macro, Money and (International) Finance – Problem Set 2

September 16, 2020

Problem set prepared by Sebastian Merkel (smerkel@princeton.edu). Please let me know, if any tasks are unclear or you find mistakes in the problem descriptions.

The submission deadline is Thursday, September 24 (end of day Princeton time). Please submit your group's solution via email to Fernando Mendo (fmendolopez@gmail.com).

1 PDE Review

Read the remaining sections in the differential equations notes.

2 The Kolmogorov Forward Equation for an Ornstein-Uhlenbeck Process

In this problem you are asked to write a solution algorithm for a relatively simple PDE, a (time-varying) Kolmogorov forward equation for a simple process. The purpose of this problem is to familiarize yourself with the solution methods you read about in Problem 1. For this reason, you should write your own PDE solution code and not rely on a solver written by someone else.

The generic form of the Kolmogorov forward equation for diffusion processes (Ito processes) is

$$\frac{\partial p}{\partial t}(x,t) = -\frac{\partial}{\partial x}\left(\mu(x,t)p(x,t)\right) + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(\sigma^2(x,t)p(x,t)\right). \tag{1}$$

If p_0 is the probability density of the initial value X_0 of a process X that follows the evolution

$$dX_t = \mu(X_t,t)dt + \sigma(X_t,t)dZ_t,$$

then $p(\cdot,t)$ is the density of the random variable X_t, if the function p solves equation (1) and satisfies the initial condition $p(x,0) = p_0(x)$ for all x. In this problem, let X follow an Ornstein-Uhlenbeck process,

$$dX_t = \theta(\bar{x} - X_t)dt + \sigma dZ_t.$$

The Kolmogorov forward equation is then

$$\frac{\partial p}{\partial t}(x,t) = \theta(x-\bar{x})\frac{\partial}{\partial x}p(x,t) + \theta p(x,t) + \frac{\sigma^2}{2}\frac{\partial^2}{\partial x^2}p(x,t). \tag{2}$$

Suppose the initial condition p_0 is a normal density with mean m_0 and variance v_0. This process is the continuous-time analog of a discrete-time AR(1) process with normal noise and we know that for a discrete-time AR(1) process x, x_t is normally distributed for all $t \geq 0$, if the initial value x_0 is normally distributed (potentially degenerate normal with zero variance). Indeed, the same is true for continuous time. The solution function is given by

$$p(x,t) = \frac{1}{\sqrt{v(t)}}\phi\left(\frac{x-m(t)}{\sqrt{v(t)}}\right) \tag{3}$$

with

$$\begin{aligned} v(t) &= v_0 e^{-2\theta t} + \left(1 - e^{-2\theta t}\right)\frac{\sigma^2}{2\theta} \\ m(t) &= m_0 e^{-\theta t} + \left(1 - e^{-\theta t}\right)\bar{x} \end{aligned}$$

where $\phi(z) = \frac{1}{\sqrt{2\pi}}e^{-z^2/2}$ is the standard normal density.[1] Use this PDE problem as a test problem to practice numerical solutions of PDEs. For your convenience, I have written a matlab script `problem2.m` that performs most of the auxiliary tasks required for problems 2-5.

1. Write a generic Kolmogorov forward equation solver for the Ornstein-Uhlenbeck process. The function should take the following inputs:

   (a) equation paramters θ, σ and $\bar{x}$;

   (b) generic grid vectors along the x and t dimensions (not necessarily uniformly spaced);

   (c) a vector of initial density values (p_0) on the specified x grid (not necessarily a normal density);

   (d) vectors of (artificial) boundary conditions $p(\underline{x},\cdot)$, $p(\overline{x},\cdot)$ on the specified t grid, where $\underline{x}$ and $\overline{x}$ are the left and right end points of the x grid;

   (e) an option specifying the solution method (explicit or implicit Euler);

   (f) an option specifying the computation of first derivatives (central differences, left differences, right differences or upwind)

   The function is supposed to return a matrix that contains the solution values $p(x,t)$ for all elements (x,t) on the specified grids.

   I have distributed a matlab template file `solveKfeOu.m` with this problem set that you can use as a starting point.

2. Using your solver function, solve equation (2) for the simple case $\bar{x} = 0$, $\theta = 0$, $\sigma = 1$ (then $dX = dZ$). Let $\underline{x} := -5$, $\overline{x} := 5$, $T := 2$ solve the equation on $[\underline{x}, \overline{x}]$ in the time interval $[0,T]$ taking as an initial value a normal density with variance $v_0 = 0.1$ and mean $m_0 = 0$, and a uniform space and time discretization with $\Delta x = 0.1$ and $\Delta t = 0.005$. Use central differences for the computation of first derivatives and compute a numerical solution using both the explicit and implicit method. For both numerical solutions $\tilde{p}$ plot the absolute error $\log_{10}|\tilde{p}-p|$ and the relative error $\log_{10}\frac{|\tilde{p}-p|}{|p|}$ as a two-dimensional plot, where p is the closed-form solution (3). Also plot the maximum relative error along the space dimension over time for both methods.

[1] If one wanted to derive this closed-form solution, one could start by guessing that the solution is a normal density, substituting it into the PDE and solving for v and m. Here, you can just take this solution as given.

3. Repeat the exercise of part 2, but this time use the finer space discretization $\Delta x = 0.05$. What happens?

4. Revert back to the values of Δx, Δt of part 2, but now add mean reversion, $\theta = 3$ and reduce the volatility to $\sigma = 0.33$. Also, let $v_0 = 0.33$, $m_0 = -3$ and $T = 1$. Solve the equation with all 8 possible method options (explicit/implicit, central/left/right/upwind). For each method check whether it yields an acceptable result and report in 1-2 sentences why it does or what goes wrong. Add some plots to your answer to illustrate your point (this is not required for all 8 possibilities, 2-5 plots are sufficient).

5. Check that your function written in part 1 also works for nonuniformly spaced grids. Generate a random space grid with $N = 200$ points and the restriction that $x_0 = x_L = -5$, $x_N = x_R = 5$ and solve the problem with parameters $\bar{x} = 0$, $\theta = 0.5$, $\sigma = 0$ and initial conditions $m_0 = 0$, $v_0 = 0.1$ (how exactly you implement the random grid generation is your choice as long as it is random). Run it a number of times and verify that the numerical approximation quality is acceptable.

3 The Lecture 3 Model with Generalized Preferences

In this problem, you will solve an augmented version of the Lecture 3 model numerically. You can use the existing solution code in the lecture notes and change it so that it solves the model presented here.

The physical environment and the set of available contracts is the same as in lecture 3, but we allow for more general preferences. Expected utility of agent type $i \in \{e, h\}$ is

$$\mathbb{E}\left[\int_0^\infty \Theta_t^i u_i(c_t^i)dt\right],$$

where u_i is a CRRA utility function with parameter $\gamma_i > 0$ (you may assume $\gamma_i \neq 1$ as in the lecture) and Θ_t^i is a stochastic time discounting process with the properties

$$\Theta_0^i = 1, \qquad \frac{d\Theta_t^i}{\Theta_t^i} = -\rho^i dt + \sigma^{\Theta,i} dZ_t.$$

The stochastic component of Θ^i can be interpreted as taste shocks.[2] If $\sigma^{\Theta,e} = \sigma^{\Theta,h} = 0$, then we get the familiar time preferences $\Theta_t^i = e^{-\rho^i t}$. If we assume in addition $\gamma^e = \gamma^h = \gamma$, this model reduces to the one solved in Lecture 3.

You can use without proof that the following facts remain true in this generalized setup:

- agents' value functions inherit the curvature of their period utility function, that is $V_t^i = \frac{u_i(\omega_t^i n_t^i)}{\rho_i}$;
- the SDF of each agent is the discounted marginal value of wealth,

$$\xi_t^i = \Theta_t^i \frac{\partial V_t^i}{\partial n_t} = \Theta_t^i \omega_t^i u_i'(\omega_t^i n_t^i)/\rho^i;$$

- the optimal consumption choice equates the marginal utility of consumption with the marginal value of wealth,

$$u_i'(c_t^i) = \frac{\partial V_t^i}{\partial n_t} = \omega_t^i u_i'(\omega_t^i n_t^i)/\rho^i.$$

[2]You have seen these shocks before in a problem at the Princeton Initiative.

1. Explain why solution steps 1 and 2 remain unaffected by this change in assumptions (1-2 sentences are sufficient)

2. Work through step 3 of the solution procedure and adjust it to this augmented preference specification. Derive

   (a) BSDEs (drift expressions) for the descaled value functions v^e and v^h;

   (b) prices of risk ς^e and ς^h;

   (c) consumption-wealth ratios c^e/n^e and c^h/n^h in terms of η, q and the descaled value functions v^e and v^h.

3. Update the code of the Lecture 3 model to incorporate the augmented preference specification. Your code should allow the user to choose arbitrary combinations of the six preference parameters γ_e, γ_h, ρ_e, ρ_h, $\sigma^{\Theta,e}$ and $\sigma^{\Theta,h}$ (currently, it imposes the restrictions $\gamma_e = \gamma_h$ and $\sigma^{\Theta,e} = \sigma^{\Theta,h}$) and run (at least) for economically reasonable choices of these parameters.

4. Set $\sigma^{\Theta,e} = \sigma^{\Theta,h} = 0$ and investigate what happens when we allow for $\gamma^e \neq \gamma^h$. Specifically, take the baseline parameterization from Slide 87 of Lecture 3, reproduce the result and then compute solutions for larger and smaller γ^e keeping all other parameters as in the baseline. For all three specifications, plot q, σ^q, κ, χ, $\mu^\eta\eta$, $\sigma^\eta\eta$ and whatever else you find informative to understand how a change in expert risk aversion changes model outcomes. Explain what happens.

5. Go back to $\gamma^e = \gamma^h$, but add taste shocks for experts. Again, use the same baseline as in part 4 and compare it to a situation with $\sigma^{\Theta,e} > 0$ and $\sigma^{\Theta,e} < 0$ by plotting solution functions. How do taste shocks affect the equilibrium?