Enrique G. Mendoza, Univ. of Pennsylvania & NBER

Lab Session: Using FiPIt to Solve a Model of Macroprudential Policy (application to Bianchi, Liu & Mendoza (JIE, 2016))

This handout describes the application of the FiPIt algorithm to solve the model of macroprudential policy studied in Bianchi, Liu and Mendoza (2016). This model features regime switches in global liquidity and noisy news about the future. Additionally, the handout contains problems that can help you practice and think of applications of this class of models.

1 MPPsolve2new.m

The Matlab code named `MPPsolve2new.m` provides the algorithm for solving the model. The code is divided into seven sections (cells). The code starts with instructions that clear the workspace and start the parallel pool, depending on the Matlab version being used. The latter can be commented out if the parallel toolbox is not going to be used (taking care of also changing all the "parfor" instructions for "for"). Whether using the parallelized code improves performance is code and parameter dependent (e.g., size of the grids, type of nonlinear solver, etc.).

Cell 1. *Values of model and algorithm parameters*: Sets the model parameter values shown in Table 1. In addition, the number of nodes for the grid of bonds is set with NB and the number of nodes for the grids of realizations of the y^T shocks and news shocks is set in NS. The grid of the regime-switching interest rate shocks has two nodes. This cell also sets several important technical parameters for execution of the algorithm: The parameter "uptd" sets the updating coefficient for constructing the new conjectured decision rules for the next iteration. "outfreq" sets the frequency with which the algorithm displays the convergence criterion of the decision rules in the screen. "iter_tol" sets the maximum number of iterations that the loops conducting the decision rule iterations are allowed to run. If it is reached, the algorithm failed to converge in the allowed number of iterations. The convergence tolerance level for the solution of decision rules is set with "tol," as defined in Section 3 below. The tolerance for defining when the collateral constraint is considered to be binding is set with the parameter "tol_EEbind".

Table 1: Baseline Model Parameters

Parameter	Value	Parameter	Value
y^N	1	N_{y^T}	3
$E[y^T]$	1	ρ_{y^T}	0.54
σ_{y^T}	0.059	β	0.91
γ	2	η	0.205
κ_L	0.32	ω	0.32
θ	$\frac{2}{3}$	R^h	1.0145
R^l	0.9672	F_{hh}	0.9333
F_{ll}	0.6		

Cell 2. . *Construction of Markov Chain*:

i) First, discretize y^T shocks using Tauchen and Hussey (1991) method, implemented in function `tauchenhussey.m`. That Matlab routine and other similar methods are available at *http://www2.hhs.se/personal/floden/*. The time-series properties of the y^T process that the method targets are estimates obtained by Bianchi (2011) using data for Argentina, and the corresponding moments are reported in Table 1.

ii) Next, incorporate news shocks according to the formulas in the Section 2.3 of the paper. Recall by Bayes rule:

$$p(y^T_{t+1} = l | s_t = i, y^T_t = j) = \frac{p(s_t = i | y^T_{t+1} = l) p(y^T_{t+1} = l | y^T_t = j)}{\sum_n p(s_t = i | y^T_{t+1} = n) p(y^T_{t+1} = n | y^T_t = j)} \tag{1}$$

Hence we can write:

$$\begin{aligned} \Pi(y^T_{t+1}, s_{t+1}, y^T_t, s_t) &\equiv p(s_{t+1} = k, y^T_{t+1} = l | s_t = i, y^T_t = j) \\ &= p(y^T_{t+1} = l | s_t = i, y^T_t = j) p(s_{t+1} = k | y^T_{t+1} = l) \\ &= p(y^T_{t+1} = l | s_t = i, y^T_t = j) \times \dots \\ &\quad \dots \sum_m \left[p(y^T_{t+2} = m | y^T_{t+1} = l) p(s_{t+1} = k | y^T_{t+2} = m) \right], \end{aligned} \tag{2}$$

iii) Finally we add global liquidity shocks to construct the entire transition matrix, assuming y^T shocks and global liquidity shocks are independent.

Cell 3. *Decentralized Equilibrium*: Solves the decentralized equilibrium using the fixed-point iteration method. Intuitively, this algorithm solves the model by backward recursive-substitution of the model's optimality conditions written in recursive form. In particular, the algorithm solves for the recursive functions $c^T(b,z)$,$P^N(b,z)$ and $B(b,z)$ that satisfy these four conditions:

$$P^N(b,z) = \frac{1-\omega}{\omega}\left(\frac{c^T(b,z)}{y^N}\right)^{1+\eta} \tag{3}$$

$$u_T(c^T(b,z),y^N) \geq \beta R(z) E_z[u_T(c^T(B(b,z),z'),y^N)] \tag{4}$$

$$B(b,z) \geq -\kappa R(z)(P^N(b,z)y^N + y^T(z)) \tag{5}$$

$$c^T(b,z) + q(z)B(b,z) = b + y^T(z) \tag{6}$$

where z is a triple (y^T,q,s) that includes the realizations of the exogenous shocks to y^T, the news signal s, and q (recall that $q=\frac{1}{R}$).

Start the algorithm at an initial iteration $K=1$ and define conjectures for the equilibrium functions for this iteration, denoted $c^T_K(b,z)$, $p^N_K(b,z)$ and $B_K(b,z)$. Then proceed with the following steps:

Step 1. Set initital conjecture for bonds decision rule as $B_{K+1}(b,z) = b$, which is the "stationary" decision rule that assumes the decision rule stays at the same value of b it started from (the 45 degree line). Initial conjectures for tradables consumption and the price of nontradables follow from the budget constraint of tradables and the marginal rate of substitution of tradables and nontradables, respectively. If the constraint binds, $B_{K+1}(b,z) = -\kappa R(z)(P^N_K(b,z)y^N + y^T(z))$ yields the maximum feasible debt (lowest b), and we can calculate $c^T_{K+1}(b,z)$ using equation (6), which yields $c^T_{K+1}(b,z) = b + y^T(z) - (1/R(z))B_{K+1}(b,z)$. For $K=1$, all of these conjectures are just the initialized arrays for the decision rules.

Step 2. Compute marginal utility of c^T, denoted *mup*, and expected marginal utility for $t+1$ as a function of (b,z), $E_z\left[u_T(c^T_K(B_K(b,z),z'),y^N)\right]$, denoted *emu*$(b,z)$. For the latter, note that we need to interpolate the consumption decision rule because, in general, $B_K(b,z)$ will not be strictly on the nodes of the bonds grid. We use linear interpolation.

Step 3. Assuming that the constraint binds in the current period, compute the Euler equation gap. This requires using the binding consumption function "cbind" in the left-hand-side, so that the Euler gap is:

$$EE \equiv u_T(c^T_{K+1}(b,z),y^N) - \beta R(z) E_z\left[u_T(c^T_K(B_K(b,z),z'),y^N)\right]$$

In the RHS, $u_T(c^T_K(B_K(b,z),z'),y^N) = u_T(B_K(b,z),z',y^N)$ requires interpolation over $B_K(b,z)$, which is done when computing *emu*(b,z)

Step 4. If $EE > tol\_EEbind$, the collateral constraint binds and then equations (5) and (6) set bonds decision rule and consumption, respectively

Step 5. If $EE \leq tol\_EEbind$, the collateral constraint does not bind. Solve for $c^T_{K+1}(b,z)$ as the recursive function that satisfies the Euler equation (4) with equality using the "fzero" solver (and interpolating in the RHS as in Step 3). We then compute $B_{K+1}(b,z)$ using equation 6 and we also compute $P^N_{K+1}(b,z)$ using equation 3.

Step 6. The above steps will in general produce a new set of functions $c^T_{K+1}(b,z)$, $p^N_{K+1}(b,z)$ and $B_{K+1}(b,z)$ that will differ from the conjectures $c^T_K(b,z)$, $p^N_K(b,z)$ and $B_K(b,z)$. We thus check the convergence criterion $d2 = \sup|x_{K+1} - x_K|$ for $x = B, c^T$ (we don't need to include p^N because its convergence its implied by convergence of c^T). If $d2 > tol$ (and as long as the maximum number of iterations is not reached, namely $iter < iter\_tol$), the conjectures are updated with a convex combination of the solutions $c^T_{K+1}(\cdot)$, $p^N_{K+1}(\cdot)$ and $B_{K+1}(\cdot)$ and the initial conjectures $c^T_K(\cdot)$, $p^N_K(\cdot)$ and $B_K(\cdot)$ and the procedure returns to step 1 using these new conjectures. If $d2 \leq tol$, the decision rules solved in the last iteration are a solution to the model's decentralized competitive equilibrium in recursive form. The solutions are stored in the file named "MPPpoliciesDE.mat".

Cell 4. *Social Planner*: Solves the social planner's problem. The algorithm is a fixed-point iteration code very similar to the one that solves the decentralized equilibrium (DE). The SP's solution uses the DE solutions as the conjectures for the planner's decision rules in the first iteration. The main difference with the cell that solves the DE is that for the SP, the marginal benefit of consuming an additional unit of c^T at date t is not just the marginal utility of c^T_t but it includes also the externality term $\mu^{SP}_t \psi_t$, where μ^{SP} is the multiplier on the collateral constraint for the planner ($\mu^{SP} \geq 0$, with strict inequality if the collateral constraint (5) binds) and ψ is the externality term given by:

$$\kappa(\eta+1)\left(\frac{1-\omega}{\omega}\right)\left(\frac{c^T}{y^N}\right)^{\eta} \tag{7}$$

It is important to note that, because of this difference in the marginal benefit of c^T_t of the planner, the Euler equation gap of the planner becomes:

$$EESP \equiv \frac{u_T(c^T_{K+1}(b,z), y^N) - \beta R(z) E_z\left[u_T(c^T_K(B_K(b,z),z'), y^N)\right]}{1-\psi(b,z)}$$

The rest of the algorithm that solves the SP decision rules is essentially the same as for the DE case. The updating coefficient for forming the new decision rules as convex combinations of the current iteration's conjectures and solutions puts significantly more way on the conjectures, which means the decision rules are updating at a much slower pace. This was found to be useful for speeding up the algorithm's execution time.

Cell 5. *Welfare Calculation*: Uses the decision rules of the DE and SP solutions and computes the corresponding value functions that measure expected lifetime utility for each coordinate in the state space. This is done by iterating to convergence on the recursive sums that take the subjective present value of the utility flows in each solution. We then calculate the welfare gain as in Bianchi (2011).

Cell 6. *Optimal Tax*: Calculates the optimal macro-prudential debt tax that recreates the SP solution as a DE solution of an economy with a debt tax. With taxes on debt τ_t, the first-order condition of the Decentralized Equilibrium is:

$$u_T(t) - \mu_t^{DE} = (1+\tau_t)\frac{\beta}{q_t}\mathbb{E}_t\left[u_T(t+1)\right] \tag{8}$$

The optimal tax must recreate the planner's allocations, and the SP's allocations satisfy the planner's Euler Equation, which is:

$$u_T(t) + \mu_t^{SP}(\psi_t + 1) = \frac{\beta}{q_t}\mathbb{E}_t\left[u_T(t+1) + \mu_{t+1}^{SP}\psi_{t+1}\right]. \tag{9}$$

The optimal tax is constructed in two stages. First, the optimal tax when the constraint is not binding at present ($\mu_t^{DE} = 0$).[1] In this case, we impose the planner's allocations on the DE Euler equation and solve for the value of the optimal macroprudential debt tax as follows:

$$1 + \tau_t = \frac{q_t}{\beta}\frac{u_T(t)}{\mathbb{E}_t\left[u_T(t+1)\right]} = 1 + \frac{\mathbb{E}_t\left[\mu_{t+1}\psi_{t+1}\right]}{\mathbb{E}_t\left[u_T(t+1)\right]} \tag{10}$$

where the first equality is obtained by solving from (8) and the second replaces $u_T(t)$ by equation (9). This is equation (15) in the paper. Second, when the collateral constraint binds at t, we verify if the constraint binds with a zero debt tax, and if it does we set the tax to zero. The allocations are independent of the tax when the collateral constraint binds, since feasible consumption for a given state (b, z) is the identical in the DE and SP solutions if the constraint binds. The results from the SP problem and the optimal taxes are stored in the file named "MPPpolicies.mat".

Cell 7. This cell makes figures of the optimal macro-prudential tax across different scenarios. Compare to Figures 6 and 7 in the paper.

[1] The multipliers for the two problems μ_t^{DE} and μ_t^{SP} do not need to be equal. However if for some allocation the constraint is not binding in the Decentralized Equilibrium, then it will not bind in the planners problem either.

2 MPPsimulation.m

The file `MPPsimulation.m` loads the results from `MPPsolve2new.m` and simulates the economy to obtain moments and to perform an event analysis. This is a standard procedure and we describe it briefly:

Cell 1. Allocates vectors and initial period values for simulations. 200,000 periods will be simulated and the first 1,000 will be discarded. Hence, vectors of size 201,000 for all exogenous and endogenous variables are created, as well as values for debt and the exogenous shock in the first period of the simulation. This cell also calls the function `markov.m` which simulates a Markov chain given a transition matrix and a initial state. Here the transition matrix built in cell 2 of `MPPsolve2new.m` is used and the evolution of the exogenous states is obtained.

Cell 2. Makes the recursive loop given the random simulations of exogenous states and the policy functions that generate values for the next period debt. After all the simulation, the first 1000 periods are discarded to get rid of potential dependence on the arbitrary first state.

Cell 3. Calculates aggregate moments of this economy and prints results. Compare with Table 2 in the paper. Also defines Sudden Stops as periods when the borrowing constraint binds and the current account is more than two standard deviations above its mean. Sudden Stops events are identified in the vector `SS` such that $SS(t) = 1$ if and only if there was a Sudden Stop in the simulation period t.

Cell 4. Calculates moments conditional on a Sudden Stop and prints results. This is basically calculating means, variances and covariances taking into account only those periods with $SS(t) = 1$. Compare with baseline model (3) of Table 3 in the paper.

Cell 5. Identifies Sudden Stop events for analysis. Variable `nbd=3` indicates the window goes from 3 periods before to 3 periods after the sudden stop. Given that, two periods with a Sudden Stop are considered the same event if they are less than 3 periods apart. Variable `nE` has the number of Sudden Stop events. Once identified, the cell makes matrices of size 7×`nE` for each variable to store its evolution and calculate average evolution paths around Sudden Stop events.

Cell 6. Makes figures for event analysis. Compare to Figures 2, 3 and 4 in the paper.

3 Problems

1. Existing empirical studies for developing countries show that the elasticity of substitution $\left(\frac{1}{1+\eta}\right)$ is less than unitary, ranging between 0.4 and 0.83. In the lab session we considered the upper bound value. How do the results (long-run moments, crisis frequencies, crisis moments & dynamics) differ when you consider the lower bound? Provide intuition for your answer.

2. Write an additional section for the code where you compute welfare gains of the constrained problem with respect to the unconstrained problem.[2] Is it evident that values of κ such that the collateral constraint binds generate cases where we observe welfare losses with respect to the unconstrained case? Explain your results.

3. (Hard) We saw in the lab that the optimal macro-prudential tax was computed according to equation (11). This optimal tax can be hard to implement as it may need to be varied frequently over time. A simpler rule would be to consider a constant tax (this is, the same tax rate for every pair (b, z)).

   (a) Fix a grid for a constant tax rate τ. For each value of the grid solve the problem with constant taxes and compute the welfare gains with respect to the decentralized solution. What is the value of τ that maximizes the welfare gains with respect to the decentralized case? Are the welfare gains larger than the ones obtained with the optimal macro-prudential tax?

   (b) Once you have identified the constant tax from the previous question, compute the long-run and crisis moments and compare them with those obtained when using the optimal macro-prudential tax. When are crisis more frequent? Explain your findings.

   (c) Using the welfare-maximizing constant tax identified above compute the crisis dynamics and compare them with those of the decentralized case. Are crisis more or less severe with the constant tax than with the optimal macro-prudential tax? Why?

[2] There are many ways of doing so. One alternative is to produce a new section that starts by calling the results of the unconstrained solution, and then solve for the constrained model. With the results of both specifications you can simulate the economy and compute welfare gains.

References

Bianchi, J. (2011). "Overborrowing and Systemic Externalities in the Business Cycle", *American Economic Review*, Vol. 101, No. 7, pp. 3400-3426.

Tauchen, G., and R. Hussey (1991). "Quadrature-based Methods for Obtaining Approximate Solutions to Nonlinear Asset Pricing Models", *Econometrica*, Vol. 59, No. 2, pp. 371-396.