Computational Appendix
“Real Interest Rates, Inflation, and Default”

Sewon Hur, Illenin Kondo, and Fabrizio Perri

November 9, 2018

1 Equilibrium Computation

We solve the recursive problem over a discretized state space. Specifically, we jointly solve for the equilibrium policy functions, value functions and pricing functions that satisfy the Markov Perfect Equilibrium definition using an iterative procedure until convergence is attained. We described below the key aspects of our numerical solution implemented in FORTRAN.

1.1 State Space Discretization

We used a discrete space consisting of $N_S = 225$ points in the space of output and inflation shocks (y, π).[1] These grid points are obtained by a discretization of the inflation-output process based on a variant of Tauchen (1986).[2]

The state space for incoming debt B consists of $N_B = 150$ unevenly spaced points and the state space for outgoing savings policy B' consists of $N_{B'} = 250$ unevenly spaced points. The uneven spacing on the debt grids allocates more points near the ergodic mean of the debt distribution. We then approximate the value functions $\left\{V^o(B,s), V^d(B,s), V^c(B,s)\right\}$, the policy functions $\{B'(B,s), d(B,s)\}$, and the debt pricing functions $\left\{q(s,B'), q^{def}(B,s)\right\}$ using a linear interpolation over the discrete state space.

[1] We use the same grid for both default cost regimes when we have two regimes.

[2] We modify the benchmark Tauchen (1986) algorithm to allow for the same output grid across cyclicality regimes. This modification allows us to more precisely compare policy functions across regimes given the same output realization. The cost of this approach is a slight offset of the inflation variance. The induced discrepancy is symmetric across procyclical and countercyclical regimes with the same variance, however. More importantly, the results are robust to the standard Tauchen (1986) discretization.

1.2 Iterative Solution Algorithm

First, we use the debt pricing under no-default and debt rollover as initial guess for the debt pricing functions $\left\{q_0(s, B'), q_0^{def}(B, s)\right\}$. We then iteratively solve for the Markov Perfect Equilibrium using the loop and updating steps described below.[3]

Debt pricing loop iteration t

1. No-default value function iteration

   Given debt pricing functions $\left\{q_{t-1}(s, B'), q_{t-1}^{def}(B, s)\right\}$ from the previous debt pricing iteration $t-1$,

   - We use a value function iteration to solve for the savings policy functions $\{B_t'(B, s)\}$ and the corresponding no-default value function $\{V_t^c(B, s)\}$ that satisfy the government's problem under no default.
     - We use a convergence criterion for the savings policy function under no default of 10^{-5}:
     $$\frac{1}{N_B \times N_s} \sum_{(B,s)} \left| B_{t-1}'(B, s) - B_t'(B, s) \right| < 10^{-5}$$

2. Implied default policy and value functions

   - We then use the no-default value function $\{V_t^c(B, s)\}$ to derive updated default policies $\{d_t(B, s)\}$ and updated value functions $\left\{V_t^d(B, s), V_t^o(B, s)\right\}$ that solve the government's default decision.

3. Implied default debt prices

   - We finally update the debt pricing during default to $\left\{q_t^{def}(B, s)\right\}$ using the updated policy functions. This is done by iteratively solving the default pricing equation (given $\{q_{t-1}(B, s)\}$ and $\{d_t(B, s)\}$).
     - We use a convergence criterion for the the default debt pricing of 10^{-6}:
     $$\frac{1}{N_B \times N_s} \sum_{(B,s)} \left| q_{t-1}^{def}(B, s) - q_t^{def}(B, s) \right| < 10^{-6}$$

4. Updated debt pricing functions $\left\{q_{t-1}(s, B'), q_{t-1}^{def}(B, s)\right\}$

   - The inner loop for the value function iteration above yields updated policy functions $\{B_t'(B, s), d_t(B, s)\}$ and updated value functions $\left\{V_t^o(B, s), V_t^d(B, s), V_t^c(B, s)\right\}$

[3] Our algorithm is parallelized using OpenMP whenever possible.

- We then update the debt pricing in default to $\{q_t(B,s)\}$ using the updated policy functions. This is done by iteratively solving the debt pricing equation (given $\left\{q_t^{def}(B,s)\right\}$ and $\{d_t(B,s)\}$).
  - We use a convergence criterion for the the debt pricing of 10^{-6}:

$$\frac{1}{N_B \times N_s} \sum_{(B,s)} |q_{t-1}(B,s) - q_t(B,s)| < 10^{-6}$$

5. Convergence and stability

- We allow the number of iterations in the debt pricing loop to be arbitrarily high for convergence but we adopt a penalty approach to guarantee convergence and prevent cycling in the savings policy.
- Specifically, we increase a penalty parameter $\chi_k \geq 0$ whenever the debt pricing loop convergence criterion stops monotonically decreasing.[4]
  - This penalty is implemented as a quadratic adjustment cost to the debt policy function inside the value function iteration step (a):

$$\chi_k \left(B' - B'_{t-1}(B,s)\right)^2$$

- We actually use an outer loop of adjustment costs $\{\chi_k\}$ that nests the debt pricing function loop
  - We verify at convergence that the consumption benefits of this penalty approach are negligible. To do so, we compare the debt policy at convergence with $\chi_k > 0$ to the one-shot relaxation of the penalty using $\chi = 0$.

1.3 Simulations

Once the Markov Perfect Equilibrium is solved, we compute key equilibrium outcomes by performing $N_S = 5,000$ random simulations, each with $T_{\text{simul}} = 20,100$ periods. We reseed the random generator across draws and we discard the first $T_{\text{erg}} = 100$ periods of the simulation to avoid path dependence on the initial state of debt $B_0 = 0$.[5]

[4] With long-term debt pricing, this non-monotonicity can occur when the optimal savings policy starts cycling and induces changes in the default policy, which in turn affect the debt pricing.

[5] We compiled the programs using Intel Fortran 17.0.7. They were executed on 64-core nodes (quad 16 core 2.4 GHz AMD Opteron processors).

References

Tauchen, G., 1986. Finite state markov-chain approximations to univariate and vector autoregressions. Economics letters 20 (2), 177–181.