THE *FiPIt* GLOBAL METHOD FOR SOLVING MODELS WITH INCOMPLETE MARKETS & CREDIT CONSTRAINTS (LAB 1)

Workhorse model again

- Optimization problem:

$$E_0 \left\{ \sum_{t=0}^{\infty} \beta^t u(c_t) \right\} \qquad u(c_t) = \frac{c_t^{1-\sigma}}{1-\sigma}$$

$$c_t = e^{z_t}\bar{y} - A + b_t - qb_{t+1}, \qquad b_{t+1} \geq -\varphi$$

- State variables (b, z), decision rules $b'(b, z), c(b, z)$, A is autonomous spending for calibration (*G*+*I*)

- Optimality conditions in recursive form:

$$c(b,z)^{-\sigma} \geq \beta R \sum_{z'} \pi(z', z) \left[\left(c(b'(b,z), z') \right)^{-\sigma} \right]$$

$$c(b,z) = e^{z}\bar{y} - A + b - qb'$$

Fixed-point iteration (*FiPIt*) method
Mendoza-Villalvazo (2020)

- *FiPIt* is an "Euler equations method" that solves recursive optimality conditions (i.e. focs + market clearing)
  - Useful for economies w. distortions, inefficiencies, policies
  - Avoids nonlinear solvers, uses simple interpolation
  - Requires differentiability and is not guaranteed to converge
- For workhorse model, it finds $b'(b,z), c(b,z)$ that satisfy opt. conditions for a finite, discrete state space given by the set of pairs: $(b,z) \in B \times Z$
- Z is constructed as discrete approx. to time-series process of shocks using quadrature methods (Tauchen-Hussey)
- B is set to include interval that supports the stochastic steady state (the long-run distribution on NFA)

FiPIt algorithm description

1. Start iteration *j* with a conjectured decision rule $\hat{b}'_j(b,z)$
2. Generate consumption dec. rule implied by that conjecture using resource constraint

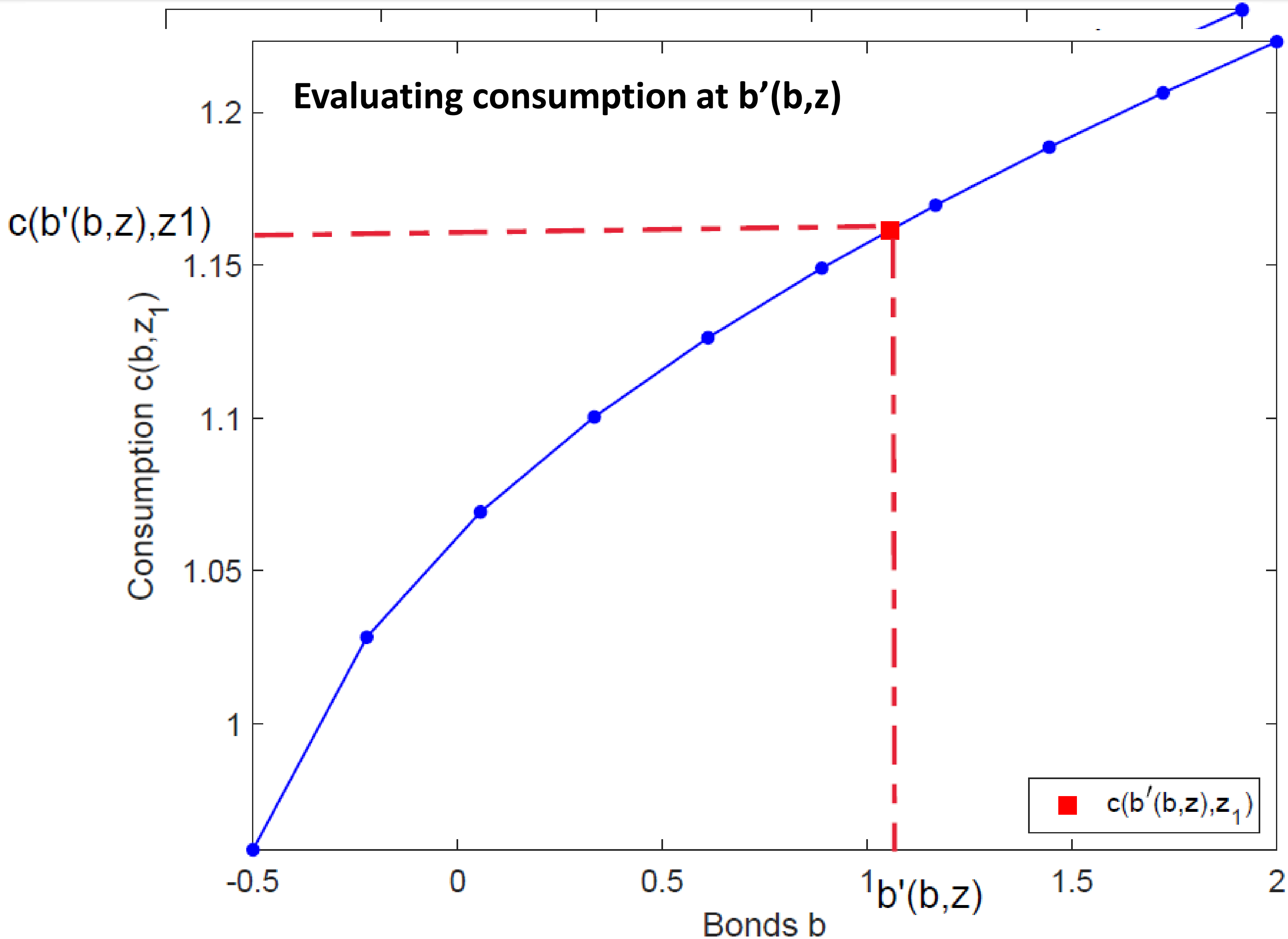
$$c_j(b,z) = e^z \bar{y} - A + b - q\hat{b}'_j(b,z)$$

3. Solve for a new consumption dec. rule "directly" using the Euler eq. (assuming φ is not binding)

$$c_{j+1}(b,z) = \left\{ \beta R \sum_{z'} \pi(z',z) \left[\left(c_j(\hat{b}'_j(b,z),z') \right)^{-\sigma} \right] \right\}^{-\frac{1}{\sigma}}$$

   - In RHS, form c_{t+1} by evaluating the j-th iteration cons. dec. rule using the values of the state variables at t+1
   - Use linear interpolation ($c_j(b,z)$ is only known at grid nodes!)
   - No need for a non-linear solver as with time iteration method

Evaluating consumption decision rule



FiPiT algorithm description

4. Generate new bond's decision rule $b'_{j+1}(b,z)$ using the resource constraint. If $b'_{j+1}(b,z) \leq -\varphi$ the debt limit binds and we set $b'_{j+1}(b,z) = -\varphi$

5. Update the initial conjecture for iteration j+1:

$$\hat{b}'_{j+1}(b,z) = (1-\rho)\hat{b}'_j(b,z) + \rho b'_{j+1}(b,z).$$

   – $0 < \rho < 1$ for unstable iterations, $\rho > 1$ for slow convergence

6. Iterate until this convergence criterion holds:

$$\max|b'_{j+1}(b,z) - \hat{b}'_j(b,z)| \leq \epsilon^b, \quad \forall(b,z) \in B \times Z$$

7. Compute ergodic distribution, moments, IRFs etc.

- Mendoza & Villalvazo (2020) provide additional details and de Groot et al. (2020) compare v. local methods

Example from de Groot et al. (2022): Calibration

- E[y] = 1 for simplicity (variables are GDP ratios)
- E[b] = -0.44 Mexico's average NFA/GDP 1985-2004 in Lane & Milesi Ferretti (06)
- E[c] = 69.2 Mexico's average C/GDP 1965-2005
- R = 1.059 Mexico's country real interest rate from Uribe and Yue (06)
- Above values imply: $A = y + b(R-1) - c = 0.282.$
- Discount factor and debt limit: $\varphi = -0.51$ $\beta = 0.94$
  - Set by searching for values of ad-hoc debt limit & discount factor that match E[b]=-0.44 and sd(c)=3.28%

Discrete state space

- Grid of NFA positions:

$$(b, b') \in B = \{b_1 < b_2 < \dots < b_n\} \qquad n=1000$$

  – Spacing=0.001514, b_1 =-0.5123

- Markov income process:

$$\varepsilon \in \mathrm{E} = \{\varepsilon_1 < \varepsilon_2 < \dots < \varepsilon_j\} \qquad \pi(\varepsilon_{t+1} \mid \varepsilon_t)$$

  – Constructed to match to Mexico's detrended GDP

$$y_t = \rho_y y_{t-1} + e_t \qquad \sigma_y = 3.301\% \qquad \rho_y = 0.597$$

$$\sigma_e = \sqrt{\sigma_y^2(1-\rho_y^2)} = 2.648 \text{ percent}$$

  – Discretized using Tauchen-Hussey quadrature method with *j*=5 (yields process with 3.28% s.d. and AR=0.55)

Summary calibration: *FiPIt* & local methods

1. Common parameters

σ	Coefficient of relative risk aversion	2.0
y	Mean endowment income	1.00
A	Absorption constant	0.28
R	Gross world interest rate	1.059
σ_z	Standard deviation of income (percent)	3.27
ρ_z	Autocorrelation of income	0.597

2. Global solution parameters

β	Discount factor	0.940
φ	Ad-hoc debt limit	-0.51

3. Local solution parameters

Common parameters

β	Discount factor	0.944
$\bar{b}$	Deterministic steady state value of NFA	-0.51

Baseline calibration

ψ	Inessential DEIR coefficient	0.001

Targeted calibration

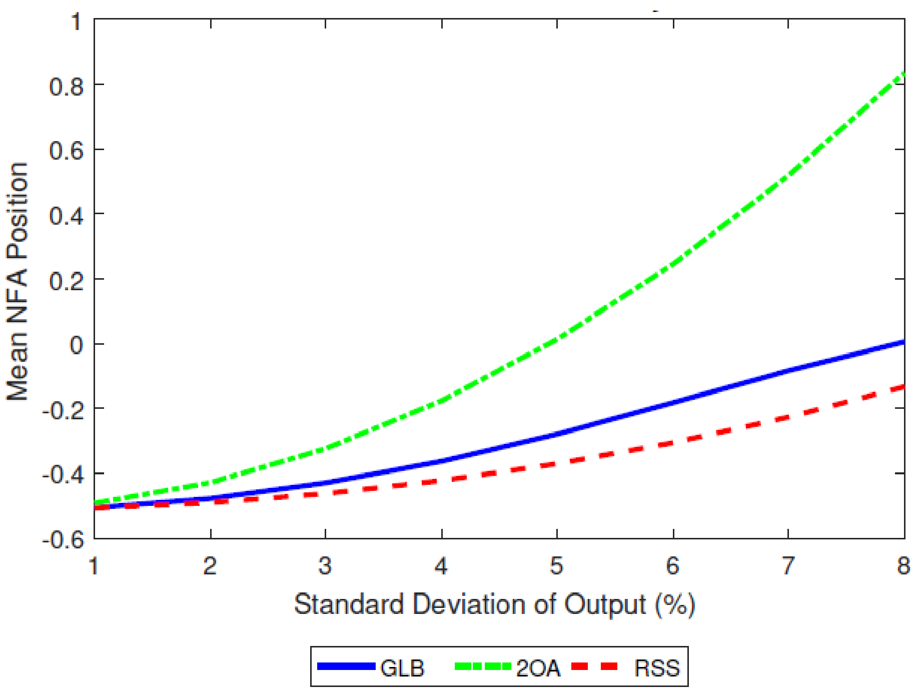
ψ	DEIR coefficeint for 2OA	0.0469
ψ	DEIR coefficient for RSS	0.0469

Moments in FiPIt & local solutions

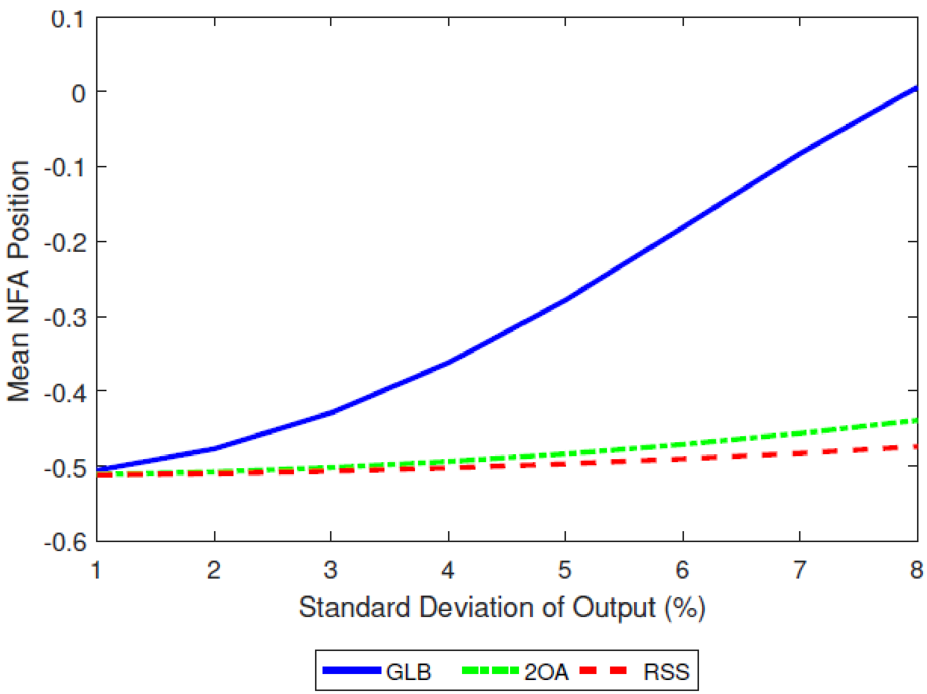
	GLB	Baseline Calibration			Targeted Calibration	
		2OA	RSS		2OA	RSS
		DEIR	$\beta R < 1$	DEIR	DEIR	DEIR
$\psi =$	*na*	0.001	*na*	0.001	0.0469	0.0469
Averages						
$E(c)$	0.694	0.701	0.093	0.692	0.689	0.689
$E(nx/y)$	0.022	0.015	0.625	0.025	0.028	0.028
$E(b/y)$	-0.413	-0.282	-11.210	-0.451	-0.502	-0.506
Standard deviations relative to standard deiation of income						
$\sigma(c)/\sigma(y)$	0.992	1.594	1.161	1.617	1.001	0.997
$\sigma(nx)/\sigma(y)$	0.660	1.327	1.202	1.346	0.730	0.730
$\sigma(nx/y)/\sigma(y)$	0.643	1.311	1.161	1.331	0.709	0.709
$\sigma(b)/\sigma(y)$	7.461	62.327	1.706	40.078	6.647	6.576
$\sigma(b/y)/\sigma(y)$	7.735	61.989	1.892	40.213	7.174	7.118
Income correlations						
$\rho(y,c)$	0.755	0.202	0.188	0.197	0.684	0.684
$\rho(y,nx)$	0.729	0.572	0.312	0.567	0.705	0.708
$\rho(y,nx/y)$	0.704	0.572	0.006	0.567	0.705	0.708
$\rho(y,b)$	0.449	0.128	0.070	0.124	0.489	0.488
$\rho(y,b/y)$	0.549	0.156	0.445	0.149	5.593	0.592
First-order autocorrelations						
ρ_c	0.840	0.995	0.996	0.995	0.929	0.929
ρ_{nx}	0.543	0.819	0.934	0.823	0.583	0.582
$\rho_{nx/y}$	0.551	0.826	0.995	0.830	0.591	0.590
ρ_b	0.977	0.999	0.999	0.999	0.977	0.977
$\rho_{b/y}$	0.961	0.998	0.953	0.998	0.958	0.959
Performance metrics						
Execution time (secs.)	5.9	8.5	n.a.	9.9	8.5	9.8
ratio rel. to GLB	1.0	1.441	n.a.	1.678	1.441	1.661
Max. Abs. Euler eq. errors	9.60E-05	1.10E-03	4.45E-03	1.00E-03	2.60E-03	2.50E-03

Precautionary savings

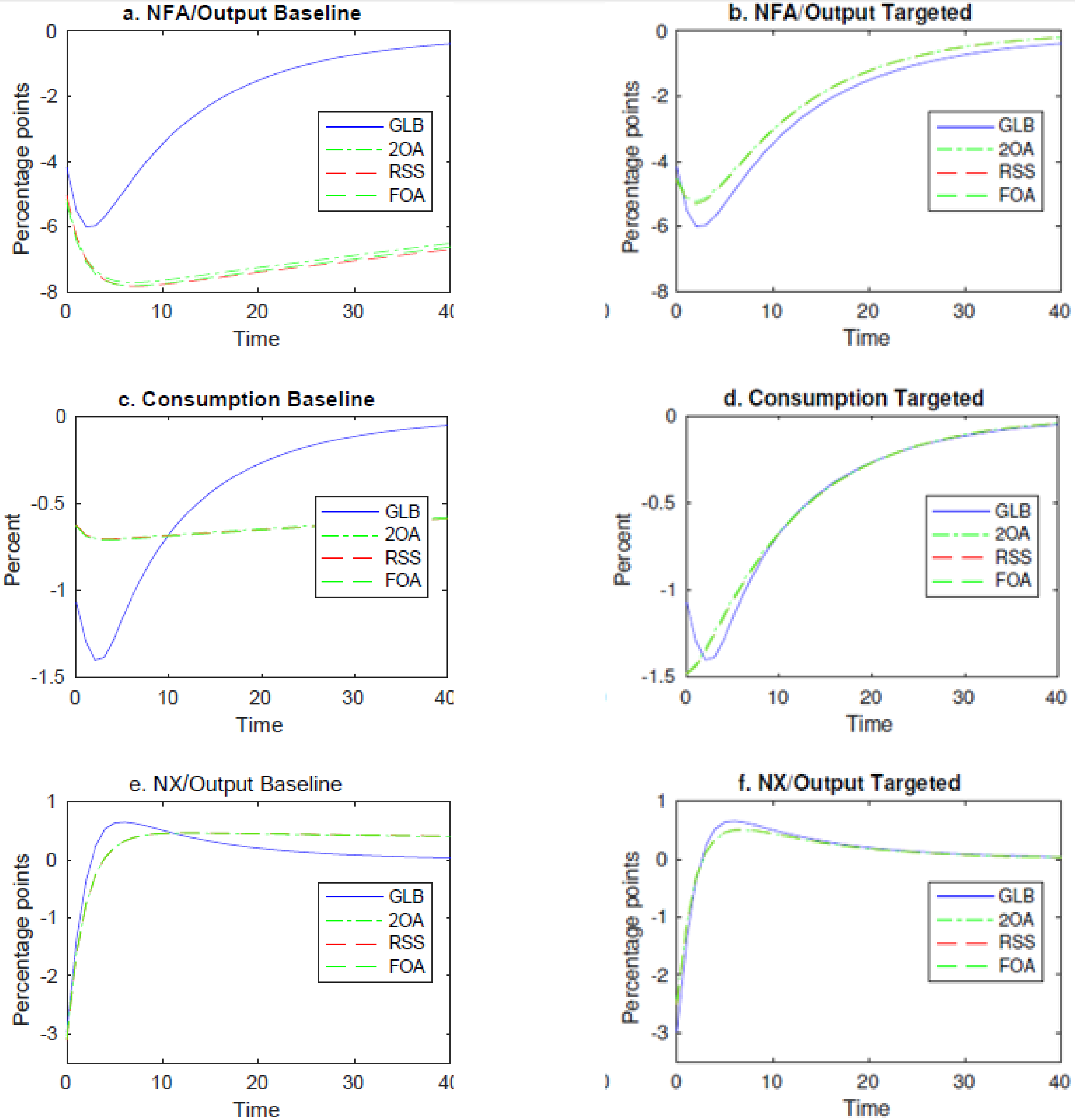
Baseline Calibration ($\psi = 0.001$)



Targeted Calibration ($\psi = 0.0469$)



IRFS to negative income shocks



FiPIt for RBC & Sudden Stops models

- Representative firm-household problem (Mendoza AER, 2010)

$$\max_{c_t, L_t, i_t, b_{t+1}} E_0 \left[\sum_{t=0}^{\infty} \beta^t \frac{(c_t - L_t^{\omega}/\omega)^{1-\sigma}}{1-\sigma} \right]$$

s.t.

$$c_t(1+\tau) + k_{t+1} - (1-\delta)k_t + \frac{a}{2}\frac{(k_{t+1}-k_t)^2}{k_t} =$$

$$A_t F(k_t, L_t, v_t) - p_t v_t - \phi(R_t - 1)(w_t L_t + p_t v_t) - q_t^b b_{t+1} + b_t$$

$$q_t^b b_{t+1} - \phi R_t (w_t L_t + p_t v_t) \geq -\kappa q_t k_{t+1}$$

  - Standard RBC-SOE model if credit constraint never binds
  - Endogenous states (b, k), exogenous states (shocks): $s \equiv (A, R, p)$

Algortihm structure & Matlab codes

(Mendoza & Villalvazo, RED, 2020)

- Paper includes step-by-step user's guide (Appendix) and zip file with all the necessary script files
- Logic of the solution is the same as in endowment model
- ***mainFiPItNew.m***: main code that solves the model (both RBC and SS versions, for RBC set κ to high value)
- ***mainFiPItNew.m*** program structure
  - Cell 1: Initialization of parameters & state space
  - Cell 2: Sets initial conjectures, arrays and options
  - Cell 3: Solves model by applying FiPIt method
  - Cell 4: Computes Euler equation errors
  - Cell 5: Computes ergodic distribution

Recursive equilibrium conditions

$$\left(c(b,k,s) - \frac{L(b,k,s)^{\omega}}{\omega}\right)^{-\sigma} = \lambda(b,k,s)(1+\tau) \tag{1}$$

$$\alpha A k^{\gamma} L(b,k,s)^{\alpha-1} v(b,k,s)^{\eta} = w(b,k,s)\left(1+\phi(R-1)+\frac{\mu(b,k,s)}{\lambda(b,k,s)}\phi R\right) \tag{2}$$

$$\eta A k^{\gamma} L(b,k,s)^{\alpha} v(b,k,s)^{\eta-1} = p\left(1+\phi(R-1)+\frac{\mu(b,k,s)}{\lambda(b,k,s)}\phi R\right) \tag{3}$$

$$\lambda(b,k,s) = R\beta E[\lambda(b'(b,k,s),k'(b,k,s),s')] + \mu(b,k,s) \tag{4}$$

$$\begin{aligned}\lambda(b,k,s) = \frac{1}{q(b,k,s)}\beta E\big[&\lambda(b'(b,k,s),k'(b,k,s),s')(d(b'(b,k,s),k'(b,k,s),s') \\ &+ q'(b'(b,k,s),k'(b,k,s),s'))\big] + \mu(b,k,s)\kappa\end{aligned} \tag{5}$$

Recursive equilibrium conditions

$$d(b,k,s) = \gamma A k^{\gamma-1} L(b,k,s)^{\alpha} v(b,k,s)^{\eta} - \delta + \frac{a}{2}\frac{(k'(b,k,s)-k)^2}{k^2} \tag{6}$$

$$q(b,k,s) = 1 + a\left(\frac{k'(b,k,s)-k}{k}\right) \tag{7}$$

$$w(b,k,s) = L(b,k,s)^{\omega-1}(1+\tau) \tag{8}$$

$$\begin{aligned} c(b,k,s)(1+\tau) + k'(b,k,s) - (1-\delta)k + \frac{a}{2}\frac{(k'(b,k,s)-k)^2}{k} &= Ak^{\gamma}L(b,k,s)^{\alpha}v(b,k,s)^{\eta} \\ -pv(b,k,s) - \phi(R-1)(L(b,k,s)^{\omega}(1+\tau) + pv(b,k,s)) - R^{-1}b'(b,k,s) + b & \end{aligned} \tag{9}$$

Cell 1: Initialize parameters & state space

- Model parameters (from Mendoza AER 2010)

σ	coefficient of relative risk aversion	2.0
ω	labor elasticity coefficient	1.8461
β	discount factor	0.92
a	capital adjustment costs coefficient	2.75
ϕ	fraction of input costs requiring working capital	0.2579
δ	depreciation rate	0.088
α	labor share in gross output	0.59
η	imported inputs share in gross output	0.10
γ	capital share in gross output	0.31
τ	tax on consumption	0.17
A	average TFP	6.982

- Algorithm parameters

ρ^b	Updating weight for bonds decision rule	1.00
ρ^μ	Updating weight for multiplier ratio	1.00
ρ^q	Updating weight for price of capital	0.30
ε^f	Function convergence criterion	10e-4

Cell 1: Initialize parameters & state space

- Choose RBC or SS model by setting κ high enough ($pKappa \geq 1$ solves RBC)
- Markov process of $s \equiv (A, R, p)$ from Mendoza (2010) (8 triples)
- Grid of bonds $b \in \mathbf{B}$ has 80 (RBC) or 72 (SS) equidistant nodes

$$b \in \mathbf{B} = \{b_1 < b_2 < \ldots < b_n\}, \qquad n = 72, 80$$

- Grid of capital $k \in \mathbf{K}$ has 30 equidistant nodes

$$k \in \mathbf{K} = \{k_1 < k_2 < \ldots < b_m\}, \qquad m = 30$$

- State space: $(b, k, s) \in \mathbf{B} \otimes \mathbf{K} \otimes \mathbf{S}$ with $80 \times 30 \times 8$ elements

Cell 2: Initial conjectures & options

- Each iteration j starts with conjectured functions for the bonds decision rule $(\hat{B}_j(b,k,s) \equiv \hat{b}_j(b,k,s))$, price of capital $\hat{q}_j(b,k,s)$, and multiplier ratio $\hat{\tilde{\mu}}_j(b,k,s)$

- ***Step 1***: Initial conjectures for $j = 0$

$$\hat{B}_0(b,k,s) = b, \quad \forall (b,k,s) \in \mathbf{B} \otimes \mathbf{K} \otimes \mathbf{S}$$

$$\hat{q}_0(b,k,s) = 1, \quad \forall (b,k,s) \in \mathbf{B} \otimes \mathbf{K} \otimes \mathbf{S}$$

$$\hat{\tilde{\mu}}_0(b,k,s) = 0, \quad \forall (b,k,s) \in \mathbf{B} \otimes \mathbf{K} \otimes \mathbf{S}$$

  - Initialization of arrays for other endogenous variables
  - Options for *fzero* (or *fsolve*) non-linear solver used to solve allocations when the occasionally binding constraint binds

Cell 3: *FiPIt* solution iterating dec. rules

- Cell 3 solves the model using FiPIt on $\hat{B}, \hat{q}, \hat{\tilde{\mu}}$
- Recursive equilibrium conditions in the Appendix
- ***Step 2***: Use $\hat{B}_j(\cdot), \hat{q}_j(\cdot), \hat{\tilde{\mu}}_j(\cdot)$ to get **iteration *j* values** for $K_j(\cdot)$, $\tilde{i}_j(\cdot)$, $v_j(\cdot)$, $L_j(\cdot)$, $y_j(\cdot)$ and $c_j(\cdot)$ for all (b, k, s):

$$K_j(b,k,s) = \frac{k}{a}[\hat{q}_j(b,k,s) - 1 + a]$$

$$\tilde{i}_j(b,k,s) = (K_j(b,k,s) - k)\left[1 + \frac{a}{2}\left(\frac{K_j(b,k,s) - k}{k}\right)\right] - \delta k$$

$$\ldots$$

- $\hat{\tilde{\mu}}_j(\cdot)$ depends on choice of RBC or SS model
  - RBC: $\hat{\tilde{\mu}}_j(\cdot) = 0$ in all iterations
  - SS: $\hat{\tilde{\mu}}_j(\cdot) > 0$ endogenously in states in which constraint binds

Rest of equations for Cell 3, Step 2

$$v_j(b,k,s) = \left\{ \frac{Ak^\gamma \eta^{\frac{\omega-\alpha}{\omega}} \frac{\alpha}{1+\tau}^{\frac{\alpha}{\omega}}}{p^{\frac{\omega-\alpha}{\omega}} [1+\phi(R-1)+\hat{\tilde{\mu}}_j(b,k,s)\phi R]} \right\}^{\frac{\omega}{\omega(1-\eta)-\alpha}}$$

$$L_j(b,k,s) = \left\{ \frac{\alpha}{\eta(1+\tau)} p v_j((b,k,s) \right\}^{\frac{1}{\omega}}$$

$$y_j(b,k,s) = Ak^\gamma L_j(b,k,s)^\alpha v_j(b,k,s)^\eta$$

$$\begin{aligned}(1+\tau)c_j(b,k,s) = y_j(b,k,s) - pv_j(b,k,s) \\ - \phi(R-1)\left[(1+\tau)L_j(b,k,s)^\omega + pv_j(b,k,s)\right] \\ - \tilde{i}_j(b,k,s) - \frac{\hat{b}'_j(b,k,s)}{R} + b\end{aligned}$$

Cell 3: *j+1* solutions if constraint does not bind

- ***Step 3.1***: Assume credit constraint does not bind ($\tilde{\mu}_{j+1}(\cdot) = 0$) and solve for $v_{j+1}(\cdot)$, $L_{j+1}(\cdot)$, $y_{j+1}(\cdot)$ (they depend on k, s only!)

$$v_{j+1}(b,k,s) = \left\{ \frac{Ak^{\gamma}\eta^{\frac{\omega-\alpha}{\omega}} \frac{\alpha}{1+\tau}^{\frac{\alpha}{\omega}}}{p^{\frac{\omega-\alpha}{\omega}}[1+\phi(R-1)]} \right\}^{\frac{\omega}{\omega(1-\eta)-\alpha}}$$

$$L_{j+1}(b,k,s) = \left\{ \frac{\alpha}{\eta(1+\tau)} p v_{j+1}(b,k,s) \right\}^{\frac{1}{\omega}}$$

$$y_{j+1}(b,k,s) = Ak^{\gamma} L_{j+1}(b,k,s)^{\alpha} v_{j+1}(b,k,s)^{\eta}$$

Since these depend only on (k,s), these are the same in RBC version where constraint never binds or in SS model with credit constraint when it does not bind

Cell 3: *j+1* solutions if constraint does not bind

- ***Step 3.2***: *fFiPIt_Cons.m* solves for $c_{j+1}(\cdot)$ using bonds Euler eq.

$$c_{j+1}(b,k,s) = \left\{\beta R E\left(c_j(\hat{B}_j(b,k,s), K_j(b,k,s), s') - \frac{L_j(\hat{B}_j(b,k,s), K_j(b,k,s), s')^{\omega}}{\omega}\right)^{-\sigma}\right\}^{-\frac{1}{\sigma}} + \frac{L_{j+1}(b,k,s)^{\omega}}{\omega}$$

  - *t+1* variables are based on iteration j conjectures
  - Bi-linear interpolation on $c_j(\cdot)$, $L_j(\cdot)$ over their first twc arguments (using *fBiLinearInterpolation.m*)

- ***Step 3.3***: Solve for $B_{j+1}(\cdot)$ using resource constraint

Cell 3: *j+1* solutions if constraint binds

- ***Step 3.4***: Check if credit constraint binds:

$$\frac{B_{j+1}(b,k,s)}{R} - \phi R[(1+\tau)L_{j+1}(b,k,s)^{\omega} + pv_{j+1}(b,k,s)]$$
$$+ \kappa \hat{q}_j(b,k,s)K_j(b,k,s) \geq 0$$

  - If true, constraint binds at point (b,k,s) move to *Step 4*
  - Otherwise, save *j+1* function values for (b,k,s) and jump to *Step 5*

- ***Step 4***: Use non-linear solver to solve for $\tilde{\mu}_{j+1}(b,k,s)$
  - Reduce system to one non-linear equation in $\tilde{\mu}_{j+1}(\cdot)$
  - *fFiPIt_MuHat.m* solves using either fzero or fsolve Matlab nonlinear solvers
  - Non-linear solver not needed if $\tilde{\mu}_{j+1}(\cdot)$ and allocations are block recursive (e.g. without working capital in constraint or if $b_{t+1} \geq \varphi$)

Cell 3: Nonlinear system when constraint binds

$$v(\tilde{\mu}) = \left\{ \frac{Ak^{\gamma} \eta^{\frac{\omega-\alpha}{\omega}} \frac{\alpha}{1+\tau}^{\frac{\alpha}{\omega}}}{p^{\frac{\omega-\alpha}{\omega}} [1 + \phi(R-1) + \tilde{\mu}\phi R]} \right\}^{\frac{\omega}{\omega(1-\eta)-\alpha}}$$

$$L(\tilde{\mu}) = \left\{ \frac{\alpha}{\eta(1+\tau)} pv(\tilde{\mu}) \right\}^{\frac{1}{\omega}}$$

$$\frac{B(\tilde{\mu})}{R} = -\kappa \hat{q}_j K_j + \phi R p v(\tilde{\mu}) \left[1 + \frac{\alpha}{\eta} \right]$$

$$(1+\tau)c(\tilde{\mu}) = y(\tilde{\mu}) - pv(\tilde{\mu}) - \phi(R-1)pv(\tilde{\mu}) \left[1 + \frac{\alpha}{\eta} \right] - \tilde{i}_j - \frac{B(\tilde{\mu})}{R} + b$$

$$y(\tilde{\mu}) \;=\; Ak^{\gamma} L(\tilde{\mu})^{\alpha} v(\tilde{\mu})^{\eta}$$

$$\tilde{\mu}_{j+1}(b,k,s) = 1 - \frac{\beta R E\left[\left(c_j(\hat{B}_j(b,k,s), K_j(b,k,s), s') - \frac{L_j(\hat{B}_j(b,k,s), K_j(b,k,s), s')^{\omega}}{\omega} \right)^{-\sigma} \right]}{\left(c(\tilde{\mu}_{j+1}(b,k,s)) - \frac{L(\tilde{\mu}_{j+1}(b,k,s))^{\omega}}{\omega} \right)^{-\sigma}}$$

Cell 3: *j+1* capital price & convergence

- ***Step 5***: Repeat Steps *2.-4.* for all (b, k, s)
- ***Step 6***: Compute $q_{j+1}(b, k, s)$ using capital Euler eq.:

$$q_{j+1}(b,k,s) = \beta E_t \left[\frac{\left(c_{j+1}(B_{j+1}(\cdot), K_j(\cdot), s') - \frac{L_{j+1}(B_{j+1}(\cdot), K_j(\cdot), s')^{\omega}}{\omega}\right)^{-\sigma}}{\left(c_{j+1}(\cdot) - \frac{L_{j+1}(\cdot)^{\omega}}{\omega}\right)^{-\sigma}(1 - \kappa\tilde{\mu}_{j+1}(\cdot))} \times [d'(\cdot) + \hat{q}_j(B_{j+1}(\cdot), K_j(\cdot), s')] \right]$$

  - fFiPIt_PriceK.m computes this using bi-linear interpolation
  - *Step 6.2* is slower option that iterates on capital pricing function

- ***Step 7***: Check convergence of $q, B, \tilde{\mu}$. If it fails, update conjectures and return to *Step 2*:

$$\hat{x}_{j+1}(b,k,s) = (1 - \rho^x)\hat{x}_j(b,k,s) + \rho^x x_{j+1}(b,k,s), \quad \text{for } x = [q, B, \tilde{\mu}]$$

Cell 4: Euler equation errors

- Recursive equilibrium solution is done when *Step 7* is completed
- Cell 4 uses decision rules to compute errors of the Euler equations of bonds and capital using fFiPIt_EulerError.m
- For SS model the errors are:

	FiPIt-large k grid	FiPIt
Bonds Euler equation		
Max Log10 abs. Euler error	-1.56	-1.56
At grid points (b, k, s)	(1, 11, 3)	(1, 6, 3)
Mean Log10 abs. Euler error	-6.27	-6.27
Capital Euler equation		
Max Log10 abs. Euler error	-6.68	-6.68
At grid points (b, k, s)	(72, 1, 7)	(72, 1, 7)
Mean Log10 abs. Euler error	-7.04	-7.05
Grid size $(\#b, \#k)$	(72, 60)	(72, 30)
Seconds elapsed	1985	810
Relative to FiPIt	2.5	1.0
Number of iterations	196	196

Cell 5: Ergodic distribution

- Iterate to convergence on law of motion of conditional prob. Λ_t

$$\Lambda_{t+1}(b', k', s') = \sum_{s} \sum_{\{b:b'=B(b,k,s)\}} \sum_{\{k:k'=K(b,k,s)\}} Pr[s'|s]\Lambda_t(b, k, s)$$

- Since (b', k') are not generally on the grid, the law of motion needs adjustments akin to interpolation
    1. Find closest grid points $b_L \leq B(b, k, s) \leq b_U$, $k_L \leq K(b, k, s) \leq k_U$
    2. Iterate to convergence on four prob. distributions using distances to the closest grids as weights. For example, for b_L, k_L:

$$\Lambda_{t+1}(b_L, k_L, s') = \sum_{s} Pr[s'|s]\Lambda_t(b, k, s)\left(\frac{b_U - B(\cdot)}{b_U - b_L}\right)\left(\frac{k_U - K(\cdot)}{k_U - k_L}\right)$$

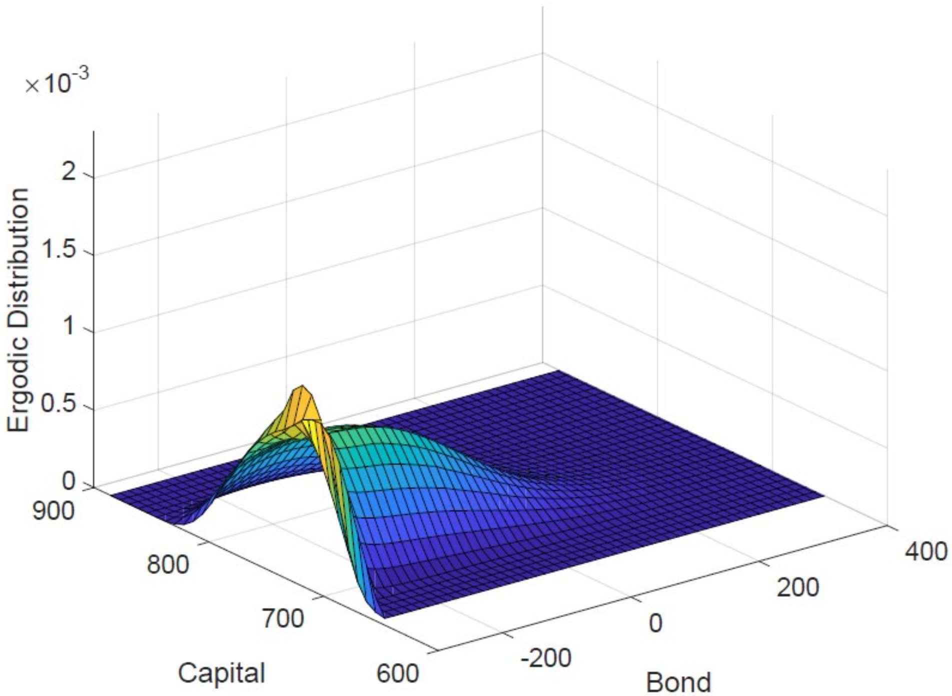
Codes for generating results

- Ergodic distributions ($mainFiPIt.m$)
- Long-run moments ($script1\_moments.m$)
- Decision rules ($script2\_PolicyPlot.m$)
- Sudden stops event windows ($script3\_Simulation.m$)
- Amplification and asymmetry during sudden stop events ($script4\_TableDiffAmpl.m$)

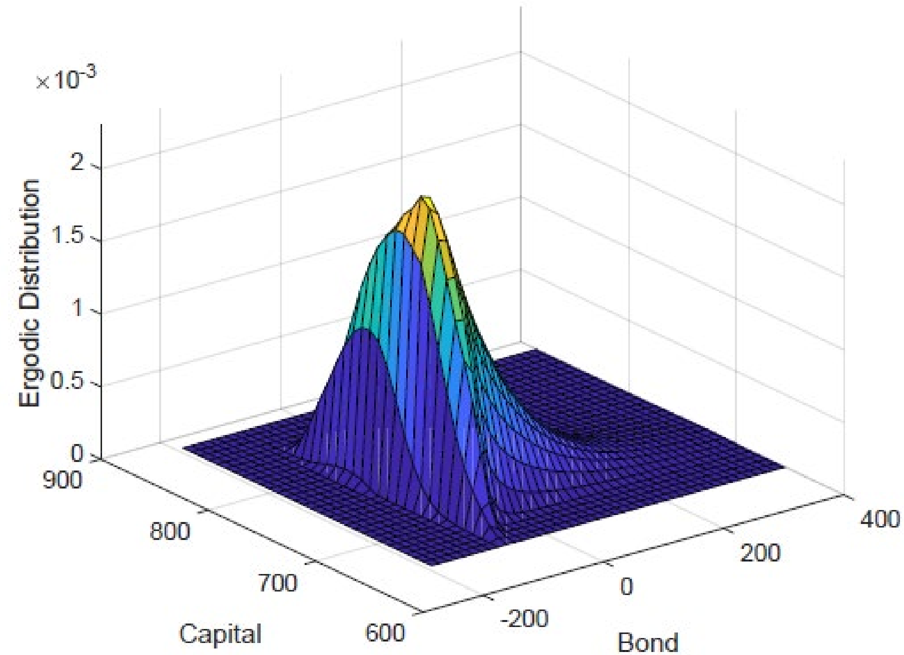
FiPIt advantages

- Euler equations solved directly without nonlinear solver
  - Similar to endogenous grids method
  - Standard time iteration solves Euler eqn's. as nonlinear system
- Standard linear interpolation of rectangular grids
  - Similar to time iteration method
  - Endogenous grids method has irregular grids, needs complex interpolation (Delaunay tringulation)
- Needs nonlinear solver only when $\tilde{\mu}_{j+1}(\cdot) > 0$ *and* if $\tilde{\mu}_{j+1}(\cdot)$ and allocations are not block recursive (i.e. need to be solved jointly)
- Mendoza and Villalvazo (2019) show FiPIt is much faster than, and just as accurate as, time iteration for RBC & SS models
- de Groot et al. (2019) show FiPIt is more accurate and of similar speed as DynareOBC/OccBin for SS model

Stochastic steady states

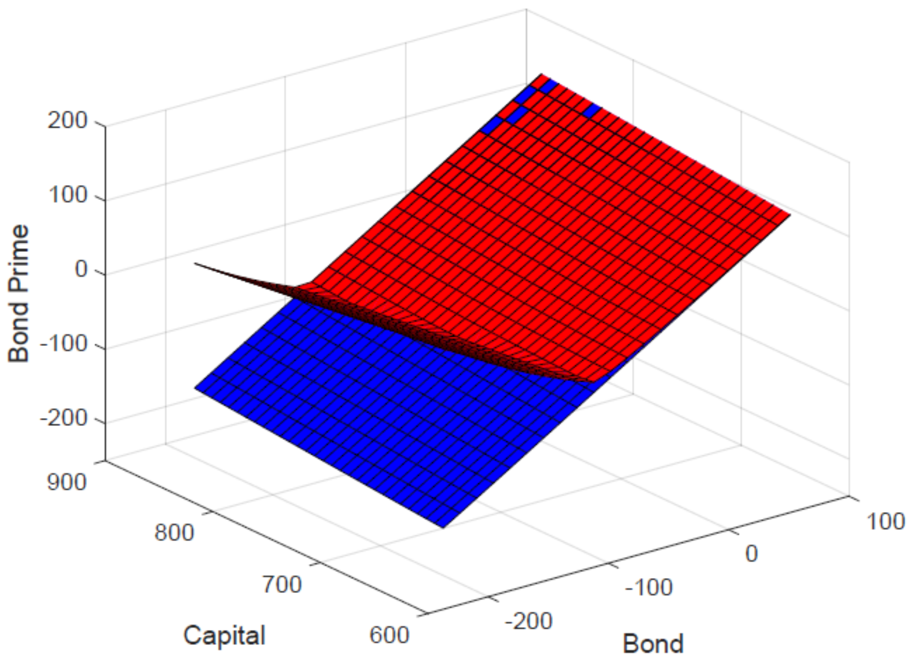


RBC model

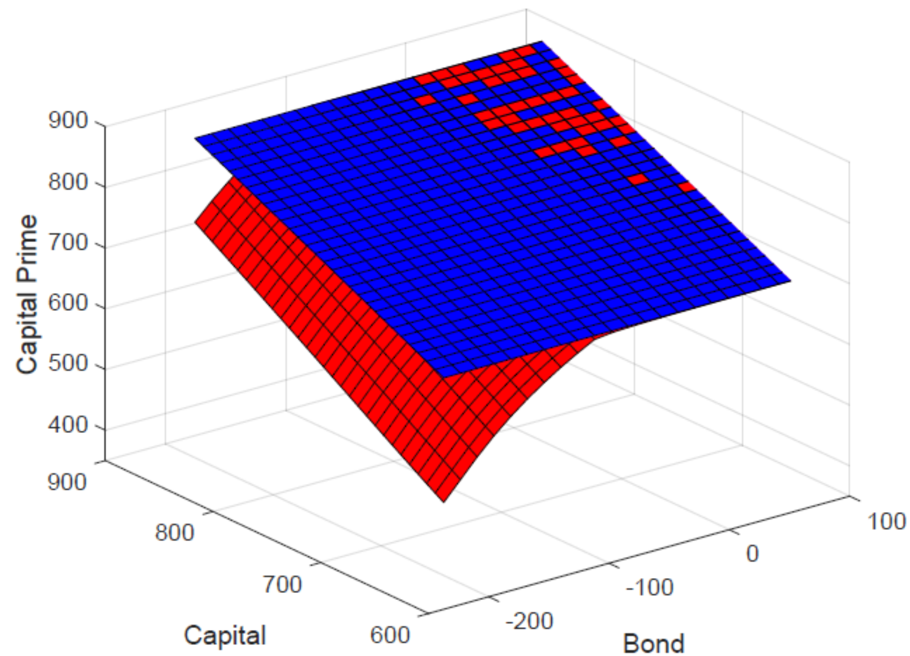


Sudden Stops model

Equilibrium decision rules



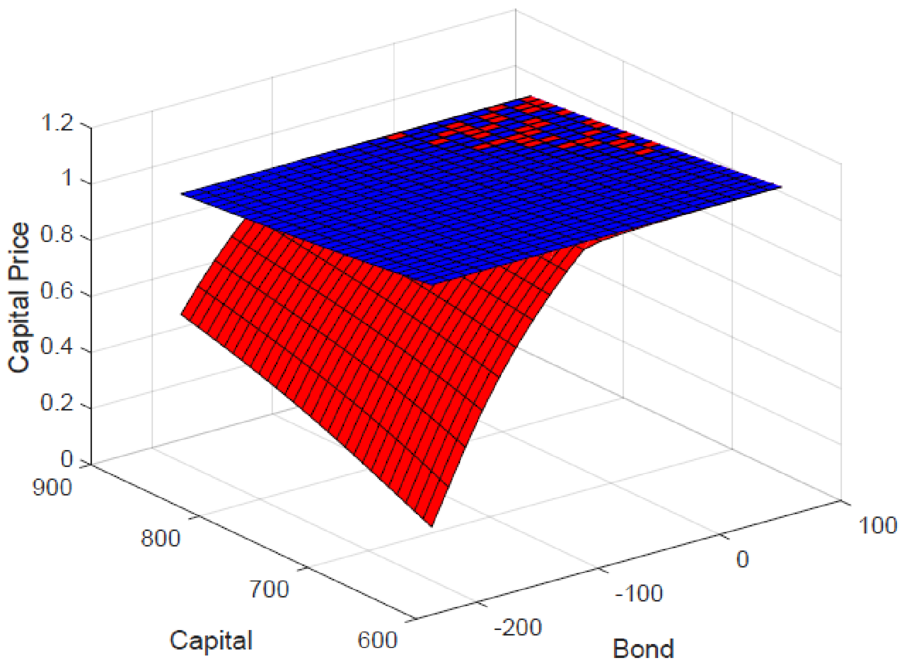
(a) Bonds decision rule



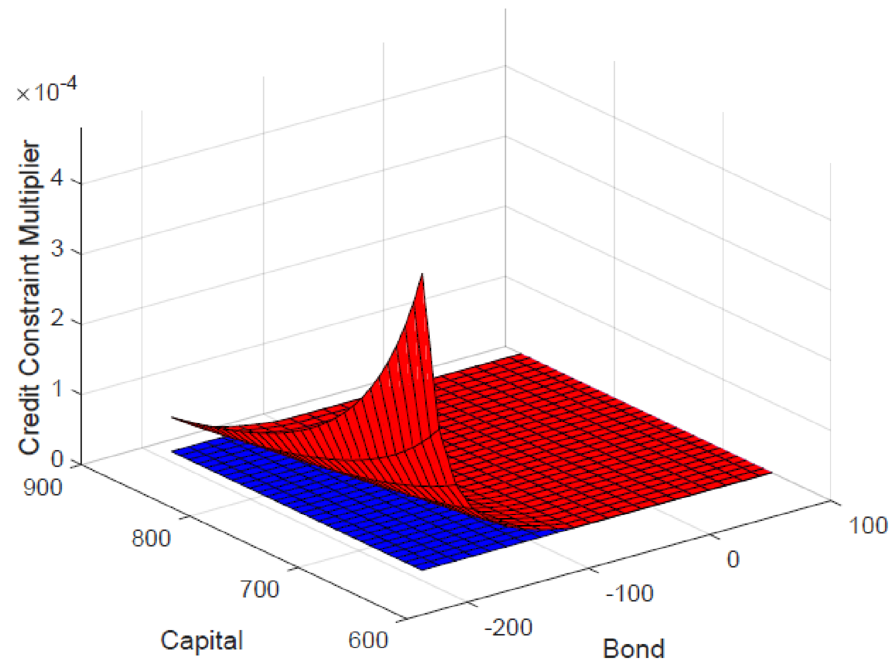
(b) Capital decision rule

RBC model, SS model

Equilibrium prices & multipliers



(c) Price of capital



(d) Credit Constaint Multiplier

RBC model, SS model

Sudden Stops