Computational Methods for Buera & Shin (2013):

Financial Frictions and the Persistence of History

Replication Notes

January 2025

Abstract

This document provides detailed computational notes for replicating the stationary equilibrium and Figure 2 of [1]. We describe the model setup, the individual optimization problem, value function iteration, computation of the stationary distribution, and the general equilibrium algorithm. Implementation details for Python with Numba acceleration are provided.

Contents

1 Model Environment

1.1 Preferences

There is a continuum of infinitely-lived agents with measure one. Agents have CRRA preferences over consumption:

$$U = \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t), \quad u(c) = \frac{c^{1-\sigma} - 1}{1-\sigma} \tag{1}$$

where $\beta \in (0, 1)$ is the discount factor and $\sigma > 0$ is the coefficient of relative risk aversion. For $\sigma = 1$, we have $u(c) = \log(c)$.

1.2 Technology

Each agent is endowed with entrepreneurial ability $e \geq 1$. An entrepreneur with ability e operates a production technology:

$$y = f(e, k, \ell) = e \cdot (k^{\alpha} \ell^{1-\alpha})^{1-\nu} \tag{2}$$

where:

- k is capital input
- ℓ is labor input
- $\alpha \in (0, 1)$ is the capital share in variable factors
- $\nu \in (0, 1)$ is the entrepreneur's share (span-of-control parameter)

The span-of-control parameter ν implies decreasing returns to scale in (k, ℓ), which bounds firm size even with perfect credit markets.

1.3 Entrepreneurial Ability Process

Ability e follows a Markov process. With probability ψ, ability remains unchanged. With probability $1 - \psi$, a new ability is drawn from the stationary Pareto distribution:

$$\Pr(e'|e) = \psi \cdot \mathbf{1}_{e'=e} + (1 - \psi) \cdot g(e') \tag{3}$$

where the stationary distribution has Pareto density:

$$g(e) = \eta \cdot e^{-(\eta+1)}, \quad e \geq 1 \tag{4}$$

with CDF $G(e) = 1 - e^{-\eta}$ and tail parameter $\eta > 0$.

1.4 Financial Frictions

Agents face a collateral constraint on capital:

$$k \leq \lambda \cdot a \tag{5}$$

where a is the agent's asset holdings and $\lambda \geq 1$ is the financial friction parameter:

- $\lambda = 1$: Financial autarky (no external borrowing)
- $\lambda = \infty$: Perfect credit markets
- $\lambda \in (1, \infty)$: Intermediate financial development

1.5 Markets

Markets are competitive:

- Labor market: wage w
- Capital rental market: rental rate $r + \delta$ where r is the interest rate and δ is depreciation
- Asset market: agents can save at rate r (no borrowing beyond collateral)

2 Individual Optimization Problem

2.1 Occupational Choice

Each period, an agent with state (a, e) chooses to be either:

1. **Worker**: Supplies 1 unit of labor, earns wage w
2. **Entrepreneur**: Operates a firm, earns profit $\pi(a, e; w, r, \lambda)$

The agent chooses the occupation that maximizes income:

$$\text{Occupation} = \begin{cases} \text{Entrepreneur} & \text{if } \pi(a, e; w, r, \lambda) > w \\ \text{Worker} & \text{otherwise} \end{cases} \tag{6}$$

2.2 Entrepreneur's Static Problem

An entrepreneur with assets a and ability e solves:

$$\pi(a, e; w, r, \lambda) = \max_{k,\ell \geq 0} \left\{ e \cdot (k^\alpha \ell^{1-\alpha})^{1-\nu} - w\ell - (r + \delta)k \right\} \tag{7}$$

subject to:

$$k \leq \lambda \cdot a \tag{8}$$

2.2.1 First-Order Conditions

The FOCs for an interior solution are:

$$\frac{\partial f}{\partial k} = (r + \delta) \quad \Rightarrow \quad \alpha(1 - \nu) \cdot e \cdot (k^\alpha \ell^{1-\alpha})^{-\nu} \cdot k^{\alpha-1} \ell^{1-\alpha} = r + \delta \tag{9}$$

$$\frac{\partial f}{\partial \ell} = w \quad \Rightarrow \quad (1 - \alpha)(1 - \nu) \cdot e \cdot (k^\alpha \ell^{1-\alpha})^{-\nu} \cdot k^\alpha \ell^{-\alpha} = w \tag{10}$$

2.2.2 Unconstrained Optimal Capital

Combining the FOCs, the unconstrained optimal capital k^* satisfies:

$$k^* = \left[\frac{\alpha(1-\nu)}{r+\delta}\right]^{\frac{1-(1-\alpha)(1-\nu)}{\nu}} \cdot \left[\frac{(1-\alpha)(1-\nu)}{w}\right]^{\frac{(1-\alpha)(1-\nu)}{\nu}} \cdot e^{1/\nu} \tag{11}$$

Derivation: Define the span $s \equiv 1-\nu$. From the FOCs:

$$\frac{k}{\ell} = \frac{\alpha}{1-\alpha} \cdot \frac{w}{r+\delta} \tag{12}$$

$$\ell = \left[\frac{(1-\alpha)s \cdot e \cdot k^{\alpha s}}{w}\right]^{\frac{1}{1-(1-\alpha)s}} \tag{13}$$

Substituting into the capital FOC and solving yields the expression above.

2.2.3 Constrained Solution

The optimal capital is:

$$\hat{k}(a,e) = \min\{k^*, \lambda a\} \tag{14}$$

Given $\hat{k}$, the optimal labor is:

$$\hat{\ell}(a,e) = \left[\frac{(1-\alpha)(1-\nu) \cdot e \cdot \hat{k}^{\alpha(1-\nu)}}{w}\right]^{\frac{1}{1-(1-\alpha)(1-\nu)}} \tag{15}$$

And profit is:

$$\pi(a,e) = e \cdot (\hat{k}^{\alpha}\hat{\ell}^{1-\alpha})^{1-\nu} - w\hat{\ell} - (r+\delta)\hat{k} \tag{16}$$

2.3 Dynamic Problem: Bellman Equation

The agent's dynamic optimization problem is:

$$V(a,e) = \max_{a' \geq 0} \left\{u(c) + \beta\mathbb{E}[V(a',e')|e]\right\} \tag{17}$$

subject to the budget constraint:

$$c + a' = y(a,e) + (1+r)a \tag{18}$$

where:

$$y(a,e) = \max\{w, \pi(a,e)\} \tag{19}$$

2.3.1 Expected Continuation Value

Given the ability transition process:

$$\mathbb{E}[V(a',e')|e] = \psi \cdot V(a',e) + (1-\psi)\sum_{e'} g(e')V(a',e') \tag{20}$$

3 Value Function Iteration

3.1 Discretization

3.1.1 Ability Grid

Following the paper (page 235), we discretize ability into $n_e = 40$ grid points $\{e_1, \ldots, e_{40}\}$:

1. Points $j = 1, \ldots, 38$: Equidistant in CDF space

$$G(e_j) = 0.633 + \frac{j-1}{37}(0.998 - 0.633), \quad j = 1, \ldots, 38 \tag{21}$$

2. Points $j = 39, 40$: Capture the tail

$$G(e_{39}) = 0.999, \quad G(e_{40}) = 0.9995 \tag{22}$$

Grid points are computed via the inverse CDF:

$$e_j = (1 - G(e_j))^{-1/\eta} \tag{23}$$

Probability masses:

$$\Pr(e_1) = \frac{G(e_1)}{G(e_{40})}, \quad \Pr(e_j) = \frac{G(e_j) - G(e_{j-1})}{G(e_{40})}, \ j = 2, \ldots, 40 \tag{24}$$

3.1.2 Asset Grid

We use n_a grid points with curvature scaling to concentrate points near zero:

$$a_i = a_{\min} + (a_{\max} - a_{\min}) \left(\frac{i-1}{n_a - 1} \right)^{\gamma}, \quad i = 1, \ldots, n_a \tag{25}$$

where $\gamma = 2$ provides quadratic spacing. Typical values: $n_a = 501$, $a_{\min} = 10^{-6}$, $a_{\max} = 4000$.

3.2 VFI Algorithm

Algorithm 1: Value Function Iteration

```
INPUT: Grids (a_i), (e_j), probabilities Pr(e_j), prices (w,r), parameters
INITIALIZE: V^0(a_i, e_j) = u(w + r*a_i)/(1-beta) for all i,j
SET: tolerance eps = 1e-5, iteration n = 0

REPEAT:
  n = n + 1
  FOR j = 1,...,n_e:  [Can parallelize over j]
    Compute expected value:
      EV_j(a') = psi*V^{n-1}(a',e_j) + (1-psi)*SUM_j' Pr(e_j')*V^{n-1}(a',e_j')
    FOR i = 1,...,n_a:
```

```
      Solve entrepreneur: (pi_ij, k_ij, l_ij) = SolveEntrepreneur(a_i, e_j, w, r, lambda)
      Occupation: y_ij = max{w, pi_ij}
      Income: I_ij = y_ij + (1+r)*a_i
      Optimal savings: a'_ij = argmax_{a' < I_ij} {u(I_ij - a') + beta*EV_j(a')}
      Update: V^n(a_i, e_j) = u(I_ij - a'_ij) + beta*EV_j(a'_ij)
   Compute: diff = max_{i,j} |V^n(a_i, e_j) - V^{n-1}(a_i, e_j)|
UNTIL: diff < eps

OUTPUT: Value function V, policy functions a'(a,e), occ(a,e)
```

3.3 Solving the Savings Problem

For each state (a_i, e_j) with income I, we find:

$$a'_* = \arg\max_{k: a_k < I} \{u(I - a_k) + \beta \cdot EV(a_k)\} \tag{26}$$

Grid search: Since u is strictly concave and the grid is finite, we can use:

1. Linear search: Check all feasible a_k (simple but $O(n_a)$)
2. Binary search: Exploit concavity to find the peak ($O(\log n_a)$)

In practice, linear search with early stopping (break when $c \leq 0$) works well since the asset grid is ordered.

4 Stationary Distribution

4.1 Transition Matrix

The state space is $\mathcal{S} = \{1, \ldots, n_a\} \times \{1, \ldots, n_e\}$ with $|\mathcal{S}| = n_a \times n_e$ states.

We index states as $s = (i-1) \cdot n_e + j$ for (a_i, e_j).

The transition matrix $Q \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ has entries:

$$Q_{s',s} = \Pr(\text{state } s' | \text{state } s) = \Pr(a' = a_{i'}, e' = e_{j'} | a = a_i, e = e_j) \tag{27}$$

Given the policy function $a'(a_i, e_j) = a_{i'(i,j)}$:

$$Q_{s',s} = \begin{cases} \psi + (1-\psi) \cdot \Pr(e_{j'}) & \text{if } i' = i'(i,j) \text{ and } j' = j \\ (1-\psi) \cdot \Pr(e_{j'}) & \text{if } i' = i'(i,j) \text{ and } j' \neq j \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

4.2 Computing the Stationary Distribution

The stationary distribution $\mu^* \in \mathbb{R}^{|\mathcal{S}|}$ satisfies:

$$\mu^* = Q\mu^*, \quad \sum_s \mu_s^* = 1 \tag{29}$$

4.2.1 Method 1: Power Iteration

Algorithm 2: Power Iteration for Stationary Distribution

```
INITIALIZE: mu^0 = (1/|S|) * 1  (uniform distribution)

REPEAT:
  mu^{n+1} = Q * mu^n
  mu^{n+1} = mu^{n+1} / ||mu^{n+1}||_1  (normalize)
UNTIL: ||mu^{n+1} - mu^n||_inf < eps

OUTPUT: mu* = mu^{n+1}
```

4.2.2 Method 2: Eigenvalue Method

Find the eigenvector corresponding to eigenvalue 1:

$$Qv = v \quad \Rightarrow \quad \mu^* = \frac{v}{\|v\|_1} \tag{30}$$

In practice, use sparse matrix methods since Q has at most n_e non-zero entries per column.

4.3 Sparse Matrix Implementation

The transition matrix Q is highly sparse. For each state $s = (i, j)$:

- There is exactly one destination asset index $i' = i'(i, j)$
- But n_e possible destination ability indices $j' \in \{1, \dots, n_e\}$

Total non-zero entries: $n_a \times n_e \times n_e$.
Store in Compressed Sparse Row (CSR) format for efficient matrix-vector multiplication.

5 Aggregate Variables

Given the stationary distribution $\mu(a, e)$ (reshaped to $n_a \times n_e$), we compute:

5.1 Aggregate Capital

$$K = \sum_{i,j} \mu(a_i, e_j) \cdot \hat{k}(a_i, e_j) \cdot \mathbf{1}_{\text{entrepreneur}}(a_i, e_j) \tag{31}$$

5.2 Aggregate Labor Demand

$$L^d = \sum_{i,j} \mu(a_i, e_j) \cdot \hat{\ell}(a_i, e_j) \cdot \mathbf{1}_{\text{entrepreneur}}(a_i, e_j) \tag{32}$$

5.3 Aggregate Output

$$Y = \sum_{i,j} \mu(a_i, e_j) \cdot f(e_j, \hat{k}(a_i, e_j), \hat{\ell}(a_i, e_j)) \cdot \mathbf{1}_{\text{entrepreneur}}(a_i, e_j) \tag{33}$$

5.4 Aggregate Assets

$$A = \sum_{i,j} \mu(a_i, e_j) \cdot a_i \tag{34}$$

5.5 Share of Entrepreneurs

$$\text{share}_e = \sum_{i,j} \mu(a_i, e_j) \cdot \mathbf{1}_{\text{entrepreneur}}(a_i, e_j) \tag{35}$$

5.6 External Finance

$$\text{ExtFin} = \sum_{i,j} \mu(a_i, e_j) \cdot \max\{0, \hat{k}(a_i, e_j) - a_i\} \cdot \mathbf{1}_{\text{entrepreneur}}(a_i, e_j) \tag{36}$$

5.7 Total Factor Productivity

Using standard capital share of 1/3:

$$\text{TFP} = \frac{Y}{K^{1/3} L^{2/3}} \tag{37}$$

6 General Equilibrium

6.1 Market Clearing Conditions

6.1.1 Labor Market

Labor supply equals labor demand:

$$L^s = 1 - \text{share}_e = L^d \tag{38}$$

Workers supply 1 unit each, so total labor supply is the fraction of workers.

6.1.2 Capital Market

Capital demand equals total assets:

$$K = A \tag{39}$$

In equilibrium, entrepreneurs' capital is funded by all agents' savings.

6.2 Equilibrium Algorithm

Algorithm 3: General Equilibrium Computation

```
INPUT: Initial guesses (w^0, r^0), tolerance tau, max iterations N
INITIALIZE: best_error = inf

FOR n = 1,...,N:
  Solve VFI with (w^{n-1}, r^{n-1}) -> policies (a'(.), occ(.))
  Compute stationary distribution mu
  Compute aggregates: K, L^d, A, share_e

  Excess demands:
    ExcL = L^d - (1 - share_e)
    ExcK = K - A

  IF |ExcL| < tau AND |ExcK| < tau:
    CONVERGED

  Update prices:
    w_new = w^{n-1} * (1 + gamma_w * ExcL)
    r_new = r^{n-1} + gamma_r * ExcK

  Damped update:
    w^n = theta * w^{n-1} + (1-theta) * w_new
    r^n = theta * r^{n-1} + (1-theta) * r_new

  Apply bounds: w^n in [0.01, 2.0], r^n in [-0.06, 0.12]

OUTPUT: Equilibrium (w*, r*) and aggregates
```

Tuning parameters:

- Step sizes: $\gamma_w = 0.3$, $\gamma_r = 0.01$
- Damping: $\theta = 0.5$
- Tolerance: $\tau = 10^{-3}$

6.3 Intuition for Price Adjustments

- **Excess labor demand** ($L^d > L^s$): Raise wage w to reduce labor demand and encourage more agents to become workers
- **Excess capital demand** ($K > A$): Raise interest rate r to reduce capital demand (higher rental cost) and encourage more savings

7 Implementation Details

7.1 Numba JIT Compilation

Key functions are compiled with `@njit(cache=True)` for C-level performance:

- `solve_entrepreneur`: Static profit maximization
- `utility`: CRRA utility evaluation
- `bellman_operator`: Main VFI iteration
- `compute_aggregates_on_grid`: Aggregate computation

7.2 Parallelization

The Bellman operator is parallelized over ability states using `@njit(parallel=True)` with `prange`:

```
@njit(cache=True, parallel=True)
def bellman_operator(...):
    for i_z in prange(n_z):  # Parallel loop
        for i_a in range(n_a):
            # Solve individual problem
```

This provides near-linear speedup with the number of CPU cores.

7.3 Memory Efficiency

- Use sparse matrices for the transition matrix Q
- Pre-compute expected values $EV(a', e)$ before the inner loops
- Store policy functions as integer indices rather than values

7.4 Numerical Stability

- Bound rental rate: $r + \delta > 10^{-8}$
- Bound wage: $w > 10^{-8}$
- Bound consumption: $c > 10^{-10}$ before computing utility
- Use -10^{10} as utility for infeasible consumption

8 Calibration

9 Computational Performance

10 Results Summary

Key findings:

Parameter	Value	Target/Source
σ	1.5	Standard value
β	0.904	Interest rate $\approx 4.5\%$
α	0.33	Capital share
ν	0.21	Span-of-control
δ	0.06	Depreciation rate
η	4.15	Top 10% employment share
ψ	0.894	Exit rate of establishments

Table 1: Baseline calibration from [1]

Component	Time per λ	Notes
VFI (per iteration)	~ 2s	501×40 states
VFI convergence	$\sim 50 - 200$ iter	Depends on initial guess
Stationary distribution	~ 0.5s	Power iteration
GE iteration	$\sim 10 - 50$ iter	Price adjustment
Total per λ	$\sim 20 - 100$s	Varies with convergence

Table 2: Approximate computation times (single core, 3.5GHz)

- Financial autarky ($\lambda = 1$) reduces GDP by $\sim 32\%$ relative to perfect credit
- TFP loss from misallocation: $\sim 22\%$
- Interest rates decline with tighter frictions (excess savings due to precautionary motives)

References

[1] Buera, Francisco J., and Yongseok Shin. *Financial frictions and the persistence of history: A quantitative exploration.* Journal of Political Economy 121.2 (2013): 221-272.

λ	ExtFin/GDP	GDP (rel.)	TFP (rel.)	r
∞	1.69	1.00	1.00	4.6%
2.00	1.26	0.83	0.87	−2.0%
1.75	1.06	0.81	0.86	−3.7%
1.50	0.75	0.78	0.84	−4.0%
1.25	0.44	0.73	0.81	−4.5%
1.00	0.00	0.68	0.78	−6.0%

Table 3: Replication of Figure 2: Long-run effect of financial frictions