Macroeconomics III: Computational Methods & Models Review

Exam Preparation Guide

Structured by Algorithms & Methodology

Fall 2025

Contents

1 Method 1: Matrix Operations for Markov Chains

Core tool for HW1, HW2, and the foundation of all heterogeneity models.

1.1 Theoretical Framework

Let $\mathcal{S} = \{1, \ldots, N\}$ be a discrete state space. T is an $N \times N$ transition matrix where $T_{ij} = \Pr(x_{t+1} = j | x_t = i)$.

- **Sparsity:** In economic models, T is sparse. Each row typically has very few non-zero elements (e.g., 2 elements in linear interpolation).
- **Row-Stochastic:** $\sum_j T_{ij} = 1$.

1.2 Key Equations (Do not confuse transpose!)

1. **The Backward Equation (Valuation):** Used to calculate the expected discounted value v (column vector) given a flow utility vector u.

$$v = u + \beta T v \implies (I - \beta T)v = u \implies v = (I - \beta T)^{-1} u \tag{1}$$

   Intuition: $(Tv)_i = \sum_j T_{ij} v_j$ *is the expectation* $\mathbb{E}[v_{t+1} | x_t = i]$. *No transpose needed.*

2. **The Forward Equation (Distribution Dynamics):** Used to evolve the probability mass vector m_t (column vector) over time.

$$m_{t+1} = T^T m_t \tag{2}$$

   Intuition: To find mass at j, *we sum inflows from all sources* i. *This requires summing over columns of* T, *hence* T^T.

1.3 Algorithm: Finding the Stationary Distribution

Goal: Find m^* such that $m^* = T^T m^*$.

Algorithm 1 Computing Stationary Distribution (Direct Linear Algebra Method)

1: Construct the matrix $A = (T^T - I)$.
2: **Issue:** The system $Am = 0$ is singular (rank $N - 1$) because rows sum to 0.
3: **Fix (Normalization):** Replace the last row of A with a row of ones: $[1, 1, \ldots, 1]$.
4: Set the corresponding element in the RHS vector to 1: $b = [0, 0, \ldots, 0, 1]^T$.
5: Solve the linear system $\tilde{A} m = b$ using matrix division (e.g., MATLAB backslash ‘m = A b‘).

1.4 Interpreting Output

- **Ergodicity:** If the distribution converges to a unique m^* regardless of initial m_0, the chain is ergodic (requires irreducibility and aperiodicity).
- **Reducible Chain:** If T has independent blocks (e.g., absorbing states), the stationary distribution depends on initial conditions.
- **Periodic Chain:** If m_t oscillates (e.g., $[1, 0] \to [0, 1] \to [1, 0]$), it has a stationary distribution but does not converge to it.

2 Method 2: Policy Function Iteration (Howard Improvement)

The gold standard for solving Infinite Horizon Dynamic Programming (HW2, HW5).

2.1 Concept

Instead of iterating the value function directly (Standard VFI), we iterate on the *policy.*

- **Policy Evaluation:** Solving a linear system is computationally cheaper than maximizing.
- **Policy Improvement:** Maximizing once per loop improves the policy significantly.
- **Convergence:** Often quadratic (extremely fast).

2.2 Algorithm Steps

Algorithm 2 Howard Improvement Algorithm (PFI)

Initialize: Guess an initial policy $g^0(s,k)$. This can be indices or exact values.
Loop $n = 0, 1, 2, \ldots$ until convergence:
  Step 1: Policy Evaluation (The Matrix Step)
  Construct the induced transition matrix T_{g^n}.
  **If Linear Interpolation: Weights ϕ and $(1-\phi)$ go into columns $j+1$ and j.*
  Solve for value: $v^n = (I - \beta T_{g^n})^{-1} u(g^n)$.
  Step 2: Policy Improvement (The Optimization Step)

for each state (s,k) **do**
    Find k' that maximizes $u(sw + Rk - k') + \beta \mathbb{E}[v^n(s', k')]$.
    **Note: Use v^n from Step 1 as the continuation value.*
    Update policy to obtain $g^{n+1}(s,k)$.

end for
  Step 3: Check Convergence

if $||g^{n+1} - g^n|| < \epsilon$ **then**
    Break

end if
End Loop

2.3 Robustness: Handling Non-Concave Value Functions

In models with discrete choices or specific asset returns (HW4), V may be non-concave.

- **Failure of FOC:** First-order conditions $u'(c) = \beta V'$ may locate a local minimum or a lesser local maximum.
- **Solution:** Do not rely on monotonicity of the derivative. Instead, perform a **Global Search** on the grid. Specifically, check the objective function value at **all** candidate peaks (boundaries of grid intervals and valid interior solutions) and pick the global max.

3 Method 3: Taste Shocks (Extreme Value Shocks)

smoothing discrete choices to aid convergence (HW5, Sovereign Default).

3.1 Mathematical Derivation

Agents receive an i.i.d. shock $\epsilon_i \sim$ Type I Extreme Value (Gumbel) for each discrete choice i. The value function is:

$$V(s) = \mathbb{E}_\epsilon \left[\max_i \{v_i + \sigma\epsilon_i\} \right]$$

where v_i is the fundamental value of choice i.

1. Expected Value (Log-Sum-Exp Formula): The expected value *before* the shock realizes has a closed form:

$$V(s) = \sigma \ln \left(\sum_i \exp\left(\frac{v_i}{\sigma}\right) \right) \tag{3}$$

2. Choice Probabilities (Softmax/Logit): The probability of choosing option j is:

$$P(j) = \frac{\exp(v_j/\sigma)}{\sum_i \exp(v_i/\sigma)} \tag{4}$$

3.2 Interpretation of Output

- **Smoothing Effect:** The "kinks" in the max operator are smoothed out. The value function becomes differentiable everywhere.
- **Dispersion vs. Concentration:**
  - **High σ (Large shocks):** Choices are driven by randomness. Probability distribution over assets is dispersed (flat).
  - **Low σ (Small shocks):** Choices are driven by fundamental value v_i. Probability mass concentrates on the optimal asset (converges to standard max).

4 Method 4: General Equilibrium (Aiyagari)

Closing the model: finding prices where Demand = Supply (HW4).

4.1 Objective

Find the equilibrium interest rate r^* such that Aggregate Capital Supply $A(r)$ equals Aggregate Capital Demand $K(r)$.

4.2 Algorithm: The Bisection Method

Algorithm 3 Solving Aiyagari Equilibrium

Initialize: Set bounds $\underline{r}$ (Excess Demand region) and $\overline{r}$ (Excess Supply region). Note: $\overline{r} < 1/\beta - 1$.
Loop until $|\overline{r} - \underline{r}| < \epsilon$:
  $r_{try} = (\underline{r} + \overline{r})/2$.
  1. Firm Side: Compute $w(r_{try})$ using FOC: $w = (1-\alpha)(\frac{r_{try}+\delta}{\alpha})^{\frac{\alpha}{\alpha-1}}$.
  2. Household Side:
    Solve Infinite Horizon Bellman for $g(s,k)$ given (r_{try}, w).
    Compute Stationary Distribution μ^* using $g(s,k)$.
    Aggregate Assets: $A(r_{try}) = \sum k \cdot \mu^*$.
  3. Market Clearing:
    Compute Demand $K(r_{try}) = (\frac{r_{try}+\delta}{\alpha})^{\frac{1}{\alpha-1}}$.
    Calculate Excess Demand $ED = K(r_{try}) - A(r_{try})$.
  4. Update Bounds:
if $ED > 0$ **then** $\underline{r} = r_{try}$ ▷ Need higher r to incentivize saving
else $\overline{r} = r_{try}$
end if
End Loop
return $r_{try}, K(r_{try})$

4.3 Graph Interpretation

- **Supply Curve $A(r)$:** Upward sloping. Why? Substitution effect (higher return) and precautionary savings motive usually outweigh income effect.
- **Demand Curve $K(r)$:** Downward sloping due to diminishing marginal product of capital ($F_{KK} < 0$).
- **Precautionary Savings:** The equilibrium r^* is lower than the complete markets rate $(1/\beta - 1)$ because agents overs-save to self-insure against risk.

5 Method 5: Firm Dynamics (Hopenhayn)

Endogenous Entry and Exit (HW6). The only model with production heterogeneity.

5.1 Core Equations

- **Entry Value:** $W^e(w) = \int W(s,w)d\nu(s) - c_e$.
- **Free Entry Condition (FE):** $W^e(w^*) = 0$.
- **Labor Market Clearing (LMC):** $L^s(w^*) = M^* \times \hat{L}^d$.

5.2 Algorithm: The Three-Step Method

Key insight: We can find prices (w^*) before finding quantities (M^*).

Algorithm 4 Hopenhayn Stationary Equilibrium

Step 1: Find Equilibrium Wage w^*
Define function ‘CheckEntry(w)‘:
  Solve Incumbent Firm Bellman $W(s,w)$.
  Identify Exit Rule: $X(s) = 1$ if $W(s,w) < 0$.
  Calculate Entry Value $W^e(w) = \mathbb{E}_\nu[W(s,w)] - c_e$.
Use Bisection on ‘CheckEntry(w) == 0‘ to find w^*.
Step 2: Find Stationary Distribution $\hat{\mu}$
Fix entry mass $M = 1$ (Normalization).
Iterate law of motion: $\mu' = \mu T(1-X) + M\nu$.
Converge to $\hat{\mu}$. Calculate aggregate labor demand $\hat{L}^d$ for this normalized economy.
Step 3: Find Entry Mass M^*
Calculate Household Labor Supply $L^s(w^*)$ (from Household FOC).
Scale the economy: $M^* = L^s(w^*)/\hat{L}^d$.
Actual Distribution: $\mu^* = M^* \times \hat{\mu}$.

5.3 Interpretation

- **Entry Value Plot $W^e(w)$:** Strictly decreasing in w. The intersection with 0 determines the unique equilibrium wage.
- **Selection Effect:** The stationary distribution μ^* is right-skewed compared to the entrant distribution ν. Low productivity firms exit; surviving firms grow.

6 Method 6: Life-Cycle Modeling

Finite Horizon Dynamics (HW3).

6.1 Algorithm: Backward & Forward

Unlike Infinite Horizon, the Value Function and Transition Matrix depend on age t.

Algorithm 5 Life-Cycle Solution

Phase 1: Backward Induction (Solving Decisions)
Set terminal value $V_{T_{end}+1}(s, a) = 0$.
for $t = T_{end}$ **down to** 1 **do**
    Solve Bellman: $V_t(s, a) = \max\{u(c) + \beta \mathbb{E}[V_{t+1}(s', a')]\}$.
    Crucial: Use V_{t+1} *computed in the previous loop step.*
    Store optimal policy g_t and induced matrix T_t.
end for
Phase 2: Forward Iteration (Simulating Demographics)
Initialize newborn distribution f_1 (e.g., all mass at $a = 0$).
for $t = 1$ **to** $T_{end} - 1$ **do**
    Evolve distribution: $f_{t+1} = T_t^T f_t$.
    Crucial: Use the age-specific matrix T_t *and Transpose.*
end for

6.2 Interpretation

- **Life-Cycle Hypothesis:** Assets follow a "hump shape": accumulation during working years, decumulation (dissaving) during retirement.
- **Horizon Effect:** Near T_{end}, agents consume everything (marginal propensity to consume $\to 1$). In Infinite Horizon, this never happens.