Canonical Model of Firm Dynamics

741 Macroeconomics
Topic 2

Masao Fukui

Fall 2024

Structural Model of Firm Dynamics

- The previous lecture took firm dynamics entirely mechanically
  - For example, we took $\underline{n}$ as exogenous
  - Cannot answer questions like how policies affect $\underline{n}$
- Today: write down a structural model of firm dynamics focusing on entry and exit
- Continuous-time version of Hopenhayn-Rogerson (1993)

Ito’s Lemma

Ito's Lemma

If z follows a diffusion with

$$dz = \mu(z)dt + \sigma(z)dZ$$

and v is twice differentiable, then $v(z)$ is also a diffusion with

$$dv(z) = v'(z)\underbrace{\left(\mu(z)dt + \sigma(z)dZ\right)}_{dz} + \frac{1}{2}\sigma(z)^2 v''(z)dt$$

- You may have guessed the expression without the second term, which is a chain rule
- Where does the second term come from?

Ito's Lemma: Proof Sketch

- Consider Taylor expansion:

$$dv(z) \approx v'(z)dz + \frac{1}{2}v''(z)(dz)^2$$

$$= v'(z)\left[\mu(z)dt + \sigma(z)dZ\right] + \frac{1}{2}v''(z)\Big[\underbrace{\mu(z)^2dt^2 + 2\mu(z)dt\sigma(z)dZ}_{\text{order smaller than dt}} + \sigma(z)^2\underbrace{(dZ)^2}_{=dt}\Big]$$

$$\approx v'(z)\left[\mu(z)dt + \sigma(z)dZ\right] + \frac{1}{2}v''(z)\sigma(z)^2dt$$

- **Intuition:**

  If v is convex, the volatility in z imparts upward drift on v through Jensen's inequality

Example

- Let z be geometric Brownian motion:

$$dz = \mu z dt + \sigma z dZ$$

- What is the stochastic process for $v(z) = \log z$? Note $v'(z) = 1/z$, $v''(z) = -1/z^2$.
- Applying Ito's Lemma

$$d \log z = \frac{1}{z}\left(\mu z dt + \sigma z dZ\right) - \frac{1}{2}\frac{1}{z^2}\sigma^2 z^2 dt$$
$$= \left(\mu - \frac{\sigma^2}{2}\right) dt + \sigma dZ$$

Hopenhayn-Rogerson in Continuous Time – Partial Equilibrium

Firm Production Technology

- We assume the firm's production function is

$$f(n, z) = z^{1-\alpha} n^{\alpha}$$

  - z: idiosyncratic productivity, n: employment

- The firm's profit function is

$$\pi(z) = \max_{n} f(n, z) - wn - c_f$$

  - c_f: fixed cost of operation

- Solutions:

$$n(z) = (\alpha/w)^{\frac{1}{1-\alpha}} z, \quad \pi(z) = \alpha^{\frac{\alpha}{1-\alpha}}(1-\alpha) w^{\frac{-\alpha}{1-\alpha}} z - c_f$$

$\Rightarrow$ Firm size n is proportional to firm productivity

- Assume z follows diffusion process

$$dz = \mu(z)dt + \sigma(z)dZ$$

Exit Decision Problem

- Firms can always exit to obtain an (exogenous) value of $\underline{v}$
- Start from a discrete time with time interval dt
- The firm's value function $v(z)$ solves

$$v(z) = \max\{v^*(z), \underline{v}\}$$

$$v^*(z) = \pi(z)dt + e^{-rdt}\mathbb{E}\left[v(z')\right]$$

  - r: discount rate
  - $v^*(z)$: value if firm decides to continue

Value if Continuing

- The value if the firm decides to continue is

$$v^*(z) = \pi(z)dt + \underbrace{e^{-rdt}}_{\approx 1 - rdt} \mathbb{E}\left[v(z')\right]$$

- Add and subtract $(1 - rdt)v(z)$ and defining $dv(z) \equiv v(z') - v(z)$, we have

$$v^*(z) = (1 - rdt)v(z) + \pi(z)dt + (1 - rdt)\mathbb{E}\left[dv(z)\right] \tag{1}$$

- Apply Ito's Lemma to $dv(z)$

$$dv(z) = v'(z)\left(\mu(z)dt + \sigma(z)dZ\right) + \frac{1}{2}\sigma(z)^2 v''(z)dt \tag{2}$$

- Plugging (2) into (1), noting $\mathbb{E}[dZ] = 0$, and dropping dt^2 terms:

$$v^*(z) = v(z) - rv(z)dt + \pi(z)dt + \mu(z)v'(z)dt + \frac{1}{2}\sigma(z)^2 v''(z)dt$$

HJB Variational Inequality

$$v(z) = \max\left\{v^*(z), \underline{v}\right\}$$

$$v^*(z) = v(z) - rv(z)dt + \pi(z)dt + \mu(z)v'(z)dt + \frac{1}{2}\sigma(z)^2 v''(z)dt$$

- Two cases:
  1. Firms continue: $v(z) = v^*(z)$ and $v(z) > \underline{v}$
  2. Firms exit: $v(z) > v^*(z)$ and $v(z) = \underline{v}$

- More compactly,

$$\min\left\{rv(z) - \pi(z) - \mu(z)v'(z) - \frac{1}{2}\sigma(z)^2 v''(z), v(z) - \underline{v}\right\} = 0$$

- This is called "HJB variational inequality"

Analytical Features

- The solution features:
  1. Firms continue if $z > \underline{z}$ and exit at $z = \underline{z}$
  2. The threshold $\underline{z}$ satisfies
     - Value matching: $v(\underline{z}) = \underline{v}$ (firm should be indifferent)
     - Smooth pasting: $v'(\underline{z}) = 0$ (marginal change in $\underline{z}$ shouldn't increase value)

Numerically Solving HJB-VI

Discretization

- We start from the case with $\underline{v} = -\infty$ so firms never exit
- The HJB equation is

$$rv(z) = \pi(z) + \mu(z)v'(z) + \frac{1}{2}\sigma(z)^2 v''(z)$$

- As in KFE, we discretize $z \in [z_1, \ldots, z_J]$ with $\Delta z = z_i - z_{i-1}$ and approximate $v'(z)$ with
  1. Forward difference approximation:

$$v'(z) \approx \frac{v(z_{i+1}) - v(z_i)}{\Delta z}$$

  2. Backward difference approximation:

$$v'(z) \approx \frac{v(z_i) - v(z_{i-1})}{\Delta z}$$

- Use forward when $\mu(z) > 0$ and backward when $\mu(z) < 0$
- The second derivative is

$$v''(z) \approx \frac{v(z_{i+1}) - 2v(z_i) + v(z_{i-1})}{(\Delta z)^2}$$

Discretized HJB

- Let us suppose $\mu(z) < 0$ and let $v_i \equiv v(z_i)$, $\pi_i \equiv \pi(z_i)$, $\mu_i = \mu(z_i)$, $\sigma_i = \sigma(z_i)$.
- For $1 < i < J$:

$$rv_i = \pi_i + \mu_i \frac{v_i - v_{i-1}}{\Delta z} + \frac{\sigma_i^2}{2} \frac{v_{i+1} - 2v_i + v_{i-1}}{(\Delta z)^2}$$

- For $i = 1$, assuming reflecting barrier,

$$rv_i = \pi_i + \mu_i \frac{v_i - v_i}{\Delta z} + \frac{\sigma_i^2}{2} \frac{v_{i+1} - 2v_i + v_i}{(\Delta z)^2}$$

- Likewise, for $i = J$,

$$rv_i = \pi_i + \mu_i \frac{v_i - v_{i-1}}{\Delta z} + \frac{\sigma_i^2}{2} \frac{v_i - 2v_i + v_{i-1}}{(\Delta z)^2}$$

Linear System

- The system of equations is linear in $\boldsymbol{v} \equiv [v_i]_i$

$$[r\boldsymbol{I} - \boldsymbol{A}]\boldsymbol{v} = \boldsymbol{\pi}$$

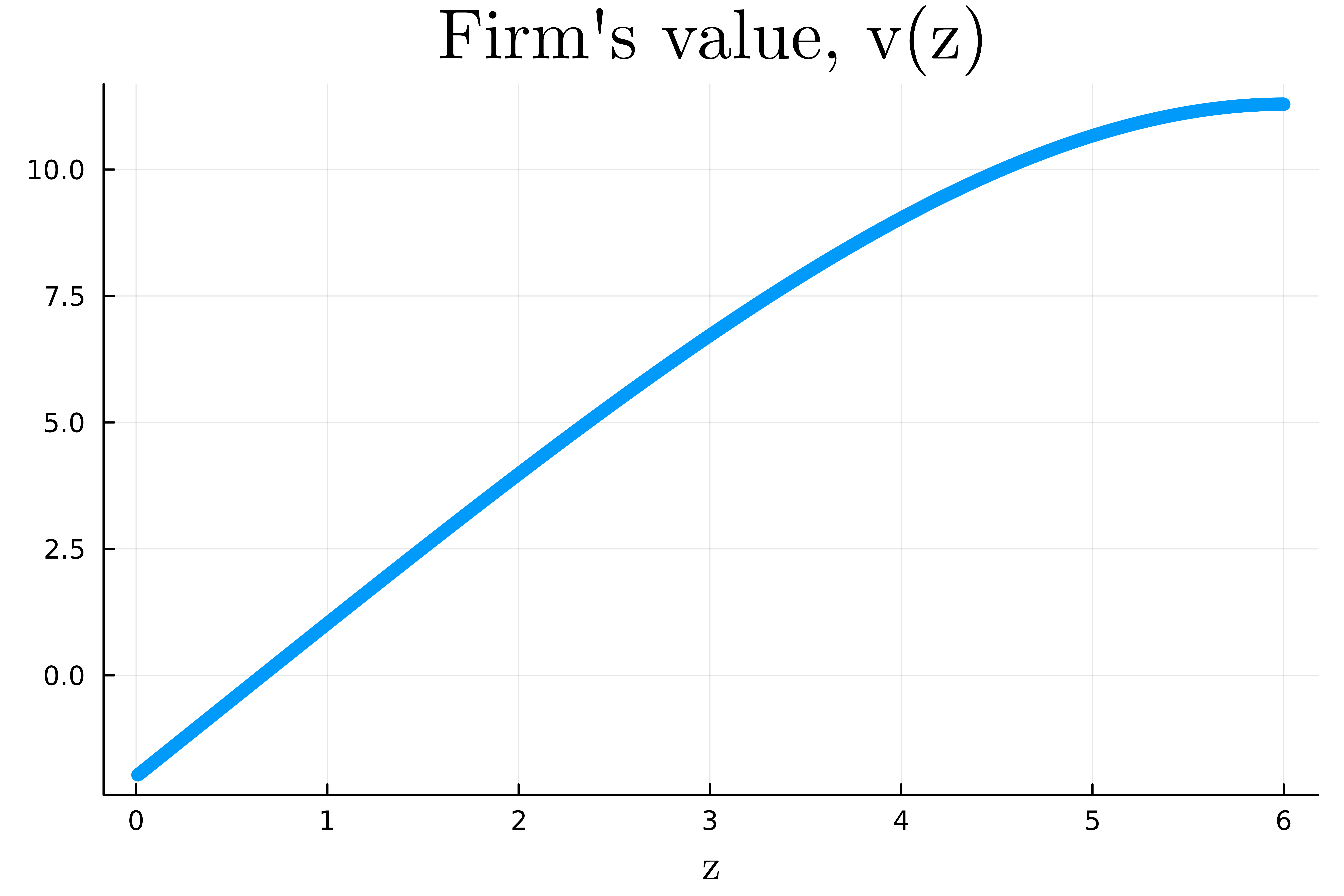
$$\Leftrightarrow \quad \boldsymbol{v} = [r\boldsymbol{I} - \boldsymbol{A}]^{-1}\boldsymbol{\pi}$$

where $\boldsymbol{\pi} \equiv [\pi_i]_i$ and

$$\boldsymbol{A} = \begin{bmatrix} -\frac{(\sigma_1)^2}{2(\Delta z)^2} & \frac{(\sigma_1)^2}{2(\Delta z)^2} & 0 & 0 & \cdots & \cdots & 0 \\ -\frac{\mu_2}{\Delta z} + \frac{(\sigma_2)^2}{2(\Delta z)^2} & \frac{\mu_2}{\Delta z} - \frac{(\sigma_2)^2}{(\Delta z)^2} & \frac{(\sigma_3)^2}{2(\Delta z)^2} & 0 & \cdots & \cdots & 0 \\ 0 & -\frac{\mu_3}{\Delta z} + \frac{(\sigma_3)^2}{2(\Delta z)^2} & \frac{\mu_3}{\Delta z} - \frac{(\sigma_3)^2}{(\Delta z)^2} & \frac{(\sigma_3)^2}{2(\Delta z)^2} & 0 & \cdots & 0 \\ 0 & \cdots & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 & -\frac{\mu_{J-1}}{\Delta z} + \frac{(\sigma_{J-1})^2}{2(\Delta z)^2} & \frac{\mu_{J-1}}{\Delta z} - \frac{(\sigma_{J-1})^2}{(\Delta z)^2} & \frac{(\sigma_{J-1})^2}{2(\Delta z)^2} \\ 0 & \cdots & \cdots & \cdots & 0 & -\frac{\mu_J}{\Delta z} + \frac{(\sigma_J)^2}{2(\Delta z)^2} & \frac{\mu_J}{\Delta z} - \frac{(\sigma_J)^2}{2(\Delta z)^2} \end{bmatrix}$$

```julia
using SparseArrays
using Parameters
using LinearAlgebra
@with_kw mutable struct model
    J = 500
    sig = 0.1
    mu = -0.01
    zg = range(0.001,6,length=J)
    dz = zg[2] - zg[1]
    alph = 0.66
    w = 1
    cf = 0.1
    r = 0.05
    underv = 0
    ng = (alph./w)^(1/(1-alph)).*zg
    pig = zg.^(1-alph).*ng.^alph .- w.*ng .- cf
    max_iter = 1e3
end
function populate_A(param)
    @unpack_model param
    A = spzeros(length(zg),length(zg))
    for (i,z) in enumerate(zg)
        if mu > 0
            A[i,min(i+1,J)] += mu.*z/dz;
            A[i,i] += -mu.*z/dz;
        else
            A[i,i] += mu.*z/dz;
            A[i,max(i-1,1)] += -mu.*z/dz;
        end
        A[i,i] += - (sig*z)^2/dz^2;
        A[i,max(i-1,1)] += 1/2*(sig*z)^2/dz^2;
        A[i,min(i+1,J)] += 1/2*(sig*z)^2/dz^2;
    end
    return A
end
function solve_HJB(param)
    @unpack_model param
    A = populate_A(param)
    v = (r.*I - A)\pig;
    return v
end
param = model()
v = solve_HJB(param)
```

Numerical Solution: $v(z)$



Endogenous Exit

- Now we assume $\underline{v} > -\infty$

$$\min \left\{ rv(z) - \pi(z) - \mu(z)v'(z) - \frac{1}{2}\sigma(z)^2 v''(z), v(z) - \underline{v} \right\} = 0$$

- In a matrix form,

$$\min \left\{ [r\mathbf{I} - \boldsymbol{A}]\boldsymbol{v} - \boldsymbol{\pi}, \boldsymbol{v} - \underline{v}\mathbf{1} \right\} = 0$$

- Now, we cannot simply invert $\boldsymbol{B} \equiv [r\mathbf{I} - \boldsymbol{A}]$. How do we solve for $\boldsymbol{v}$?
- We will solve using Howard's algorithm (Bokanowski, Maroso & Zidani, 2009)

Howard's Algorithm

1. Guess $\boldsymbol{v}^0$

2. For $k \geq 0$, given $\boldsymbol{v}^k$, set

$$d_i = \begin{cases} 0 & [\boldsymbol{B}\boldsymbol{v}^k - \boldsymbol{\pi}]_i \leq v_i^k - \underline{v} \\ 1 & [\boldsymbol{B}\boldsymbol{v}^k - \boldsymbol{\pi}]_i > v_i^k - \underline{v} \end{cases}$$

3. Set

$$[\tilde{\boldsymbol{B}}]_{ij} = \begin{cases} [\boldsymbol{B}]_{ij} & \text{if } d_i = 0 \\ [\boldsymbol{I}]_{ij} & \text{if } d_i = 1 \end{cases}, \qquad [\boldsymbol{q}]_i = \begin{cases} [\boldsymbol{\pi}]_i & \text{if } d_i = 0 \\ \underline{v} & \text{if } d_i = 1 \end{cases}$$

4. Update v^{k+1} solving

$$\tilde{\boldsymbol{B}}\boldsymbol{v}^{k+1} = \boldsymbol{q} \quad \Leftrightarrow \quad \boldsymbol{v}^{k+1} = \tilde{\boldsymbol{B}}^{-1}\boldsymbol{q}$$

5. If $|v^{k+1} - v^k| < tol$, we are done, otherwise go back to 2 with $k := k + 1$

Howard vs. LCP

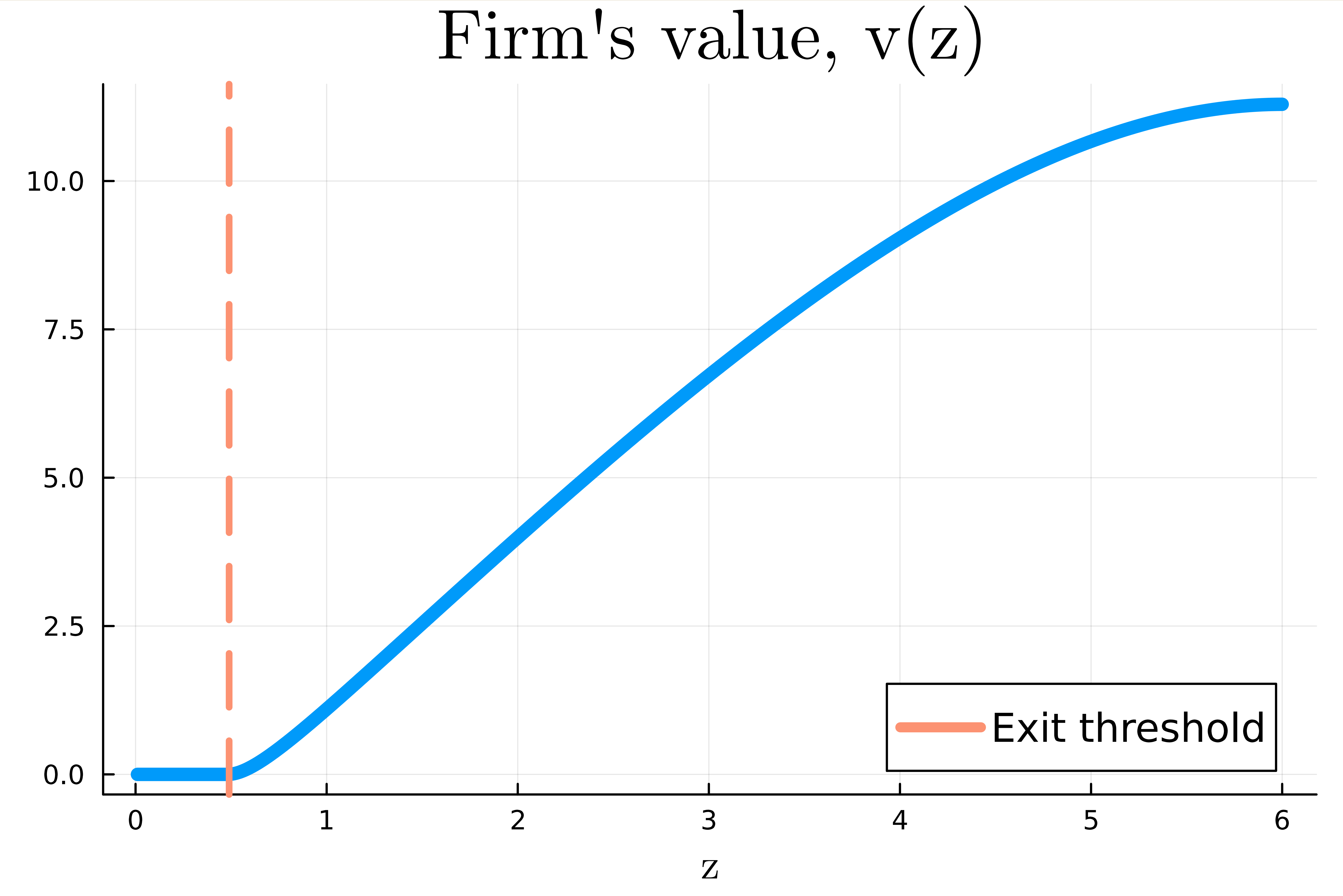
- Bokanowski, Maroso & Zidani (2009) prove this converges in at most J iterations
  - In practice, it converges very quickly
- Many economists solve using LCP (linear complementarity problem) solver
- I found LCP neither efficient nor robust

Julia Code to Solve HJB VI

```
function Howard_Algorithm(param,A)
    @unpack_model param
    B = (r.*I - A);
    iter = 1;
    vold = zeros(length(zg));
    vnew = copy(vold);
    while iter < max_iter
        val_noexit =  (B*vold .- pig);
        val_exit = vold  .- underv
        exit_or_not =val_noexit  .> val_exit;
        Btilde = B.*(1 .-exit_or_not) + I(J) .*(exit_or_not)
        q = pig.*(1 .-exit_or_not) + underv.*(exit_or_not)
        vnew = Btilde\q;
        if norm(vnew - vold) < 1e-6
            break
        end
        vold = copy(vnew)
        iter += 1
    end
    @assert iter < max_iter "Howard Algorithm did not converge"
    return vnew,exit_or_not
end

function solve_HJB_VI(param)
    @unpack_model param
    A = populate_A(param)
    v,exist_or_not = Howard_Algorithm(param,A)
    underz_index = findlast(exist_or_not .>0 )
    if isnothing(underz_index)
        underz_index = J
    end
    underz = zg[underz_index]
    return v,underz
end
v_exit,underz = solve_HJB_VI(param)
```

Numerical Solution: $v(z)$ with Potential Exit



Back to a Mechanical Model

- Further assume when firms exit, they are replaced by the entrants
- At this point, we recover the same structure as the previous lecture
  - in a micro-founded way
- Firm productivity z (which is proportional to n: $n(z) = (\alpha/w)^{\frac{1}{1-\alpha}} z$) evolves

$$dz = \mu(z)dt + \sigma(z)dZ \quad \text{for } z > \underline{z},$$

  firms exit at $z = \underline{z}$, and replaced by the entrants

- Now we would like to move to a general equilibrium by
  1. modeling entry in a less mechanical manner
  2. endogenizing wages, w

Hopenhayn-Rogerson in Continuous Time – General Equilibrium

Free Entry

- Suppose there is a large mass of potential firms that can create firms with a cost c_e
- Upon entry, firms draw z from density $\psi(z)$
- The free-entry condition is, assuming there is an entry in equilibrium,

$$\int v(z)\psi(z)dz = c_e$$

  - If $\int v(z)\psi(z)dz > c_e$, infinitely many firms enter
  - If $\int v(z)\psi(z)dz < c_e$, no firm enters

- Letting m be the endogenous mass of entrants, the KFE in the steady-state is

$$0 = -\partial_z[\mu(z)g(z)] + \frac{1}{2}\partial_{zz}^2\left[\sigma(z)^2 g(z)\right] + m\psi(z) \quad \text{for } z > \underline{z}$$

Labor Market Clearing

- We have described the labor demand side. What about the supply side?
- Assume households have labor endowment L
- The household's problem is

$$\max_{\{C_t\}} \int_0^\infty e^{-rt} C_t dt$$

$$\text{s.t.} \quad C_t = wL + \int \pi(z; w) g(z) dz - m c_e$$

- Labor market clearing is

$$\int n(z) g(z) dz = L$$

Summarizing Equilibrium System

- Equilibrium consists of $\{v(z), g(z)\}_z, \underline{z}, w, m$ such that

$$\min\left\{rv(z) - \pi(z;w) - \mu(z)v'(z) - \frac{1}{2}\sigma(z)^2 v''(z), v(z) - \underline{v}\right\} = 0$$

$$v(\underline{z}) = \underline{v}$$

$$\int v(z)\psi(z)dz = c_e$$

$$0 = -\partial_z[\mu(z)g(z)] + \frac{1}{2}\partial^2_{zz}\left[\sigma(z)^2 g(z)\right] + m\psi(z) \quad \text{for } z > \underline{z}$$

$$\int n(z;w)g(z)dz = L$$

where $n(z;w) = (\alpha/w)^{\frac{1}{1-\alpha}} z$ and $\pi(z;w) = f(n(z;w), z) - wn(z;w) - c_f$

Numerically Solving General Equilibrium

Block Recursivity: Value Function Block

- Hopenhayn-Rogerson model has a very particular structure – block recursivity
  - Equilibrium value/policy functions are separable from the distribution
- The following system determines $\{\{v(z)\}_z, \underline{z}, w\}$ independent from the rest

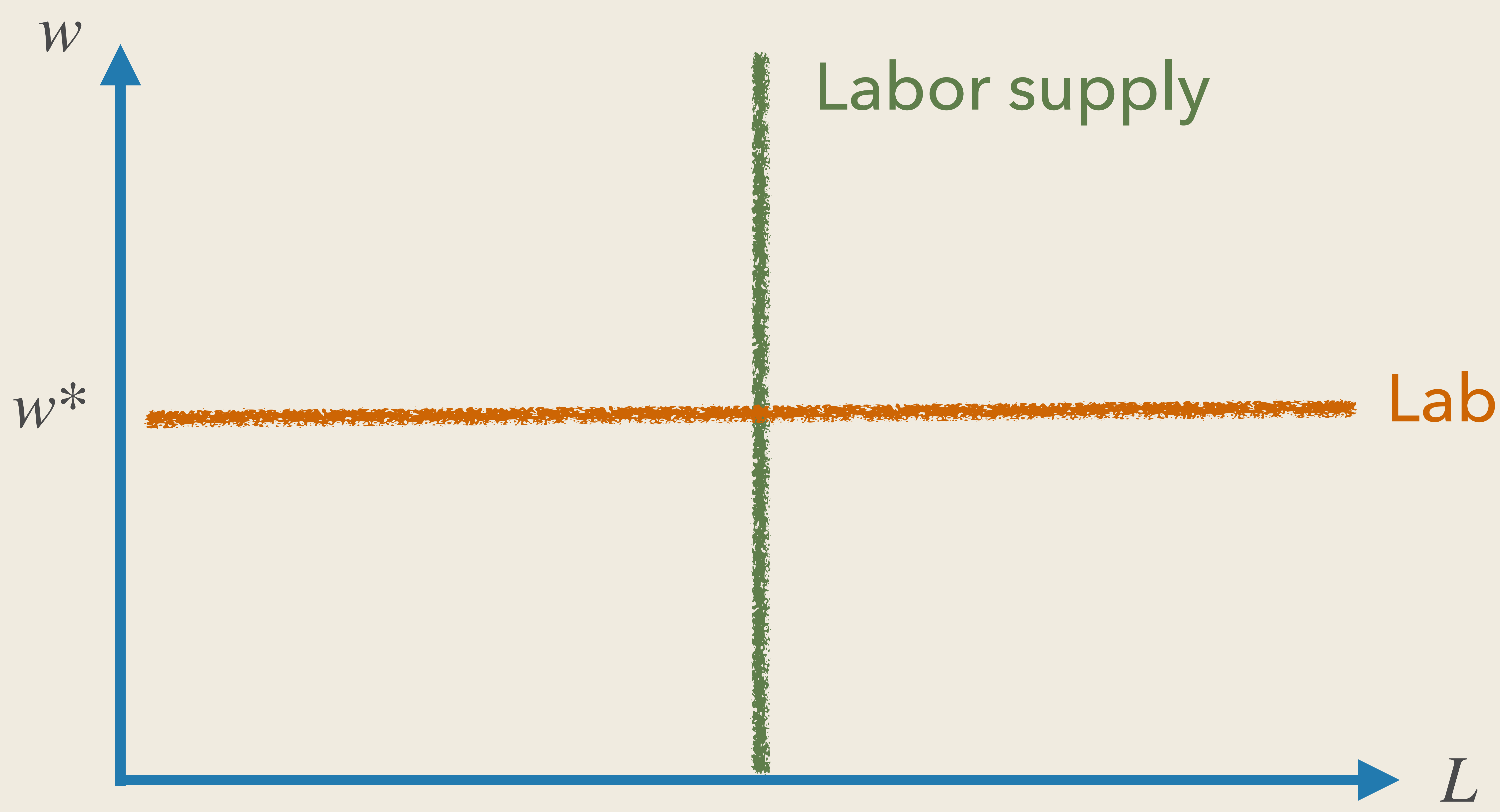
$$\min\left\{rv(z) - \pi(z;w) - \mu(z)v'(z) - \frac{1}{2}\sigma(z)^2v''(z), v(z) - \underline{v}\right\} = 0$$

$$v(\underline{z}) = \underline{v}$$

$$\int v(z)\psi(z)dz = c_e$$

1. Distribution of firms, in itself, is irrelevant to aggregate wage
2. Labor supply is irrelevant to aggregate wage

Horizontal Aggregate Labor Demand



- Labor demand horizontal: $w > w^* \Rightarrow$ infinite demand; $w < w^* \Rightarrow$ no demand.
- For a given wage $w = w^*$, the labor market clears because entry adjusts

Block Recursivity: Distribution Block

- Given $\{w, \underline{z}\}$ obtained in the previous step, $\{g(z)\}$ and m solve

$$0 = -\partial_z[\mu(z)g(z)] + \frac{1}{2}\partial^2_{zz}\left[\sigma(z)^2 g(z)\right] + m\psi(z) \quad \text{for } z > \underline{z}$$

$$\int n(z; w)g(z)dz = L$$

- Defining $\hat{g}(z) \equiv g(z)/m$, we proceed in the following steps:

1. Solve for $\hat{g}(z)$:

$$0 = -\partial_z[\mu(z)\hat{g}(z)] + \frac{1}{2}\partial^2_{zz}\left[\sigma(z)^2 \hat{g}(z)\right] + \psi(z) \quad \text{for } z > \underline{z}$$

2. Obtain m using

$$m\int n(z; w)\hat{g}(z)dz = L$$

Discretized KFE

- Let $\psi_i \equiv \psi(z_i), \mu_i \equiv \mu(z_i), \sigma_i \equiv \sigma(z_i), \boldsymbol{\psi} \equiv [\psi_i]_i$.

- Let $\underline{i}$ such that $z_{\underline{i}} = \underline{z}$. The discretized KFE is (assuming $\mu < 0$)

$$\frac{-\mu_{i+1}\hat{g}_{i+1} + \mu_i \hat{g}_i}{\Delta n} + \frac{1}{2}\frac{\sigma_{i+1}^2 \hat{g}_{i+1} - 2\sigma_i^2 \hat{g}_i + \sigma_{i-1}^2 \hat{g}_{i-1}}{(\Delta n)^2} + \psi_i = 0 \quad \text{for } i = \underline{i}+1, \ldots, J-1$$

$$\frac{-\mu_{i+1}\hat{g}_{i+1} + \mu_i \hat{g}_i}{\Delta n} + \frac{1}{2}\frac{\sigma_{i+1}^2 \hat{g}_{i+1} - 2\sigma_i^2 \hat{g}_i}{(\Delta n)^2} + \psi_i = 0 \quad \text{for } i = \underline{i}$$

$$\frac{+\mu_i \hat{g}_i}{\Delta n} + \frac{1}{2}\frac{\sigma_i^2 \hat{g}_i - 2\sigma_i^2 \hat{g}_i + \sigma_{i-1}^2 \hat{g}_{i-1}}{(\Delta n)^2} + \psi_i = 0 \quad \text{for } i = J$$

$$\hat{g}_i = 0 \quad \text{for } i < \underline{i}$$

KFE in a Matrix Form

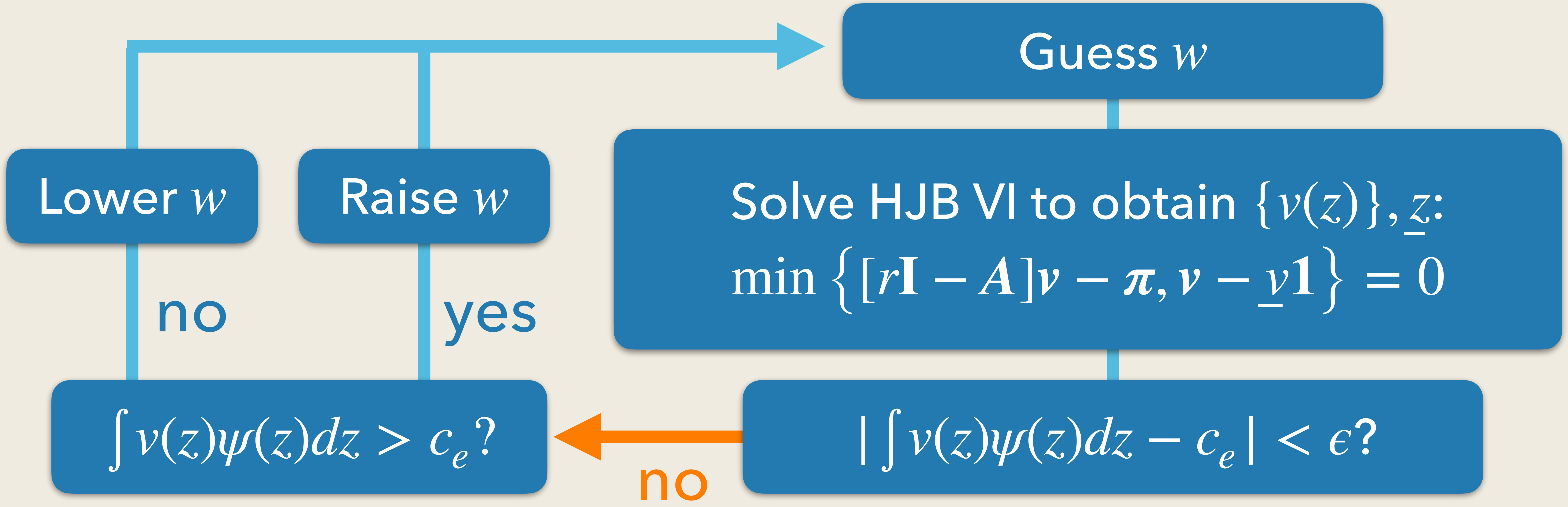
$$\boldsymbol{A}^T \hat{\boldsymbol{g}} + \boldsymbol{\psi} = \mathbf{0} \quad \text{for } i \geq \underline{i}$$

$$\hat{\boldsymbol{g}} = \mathbf{0} \quad \text{for } i < \underline{i}$$

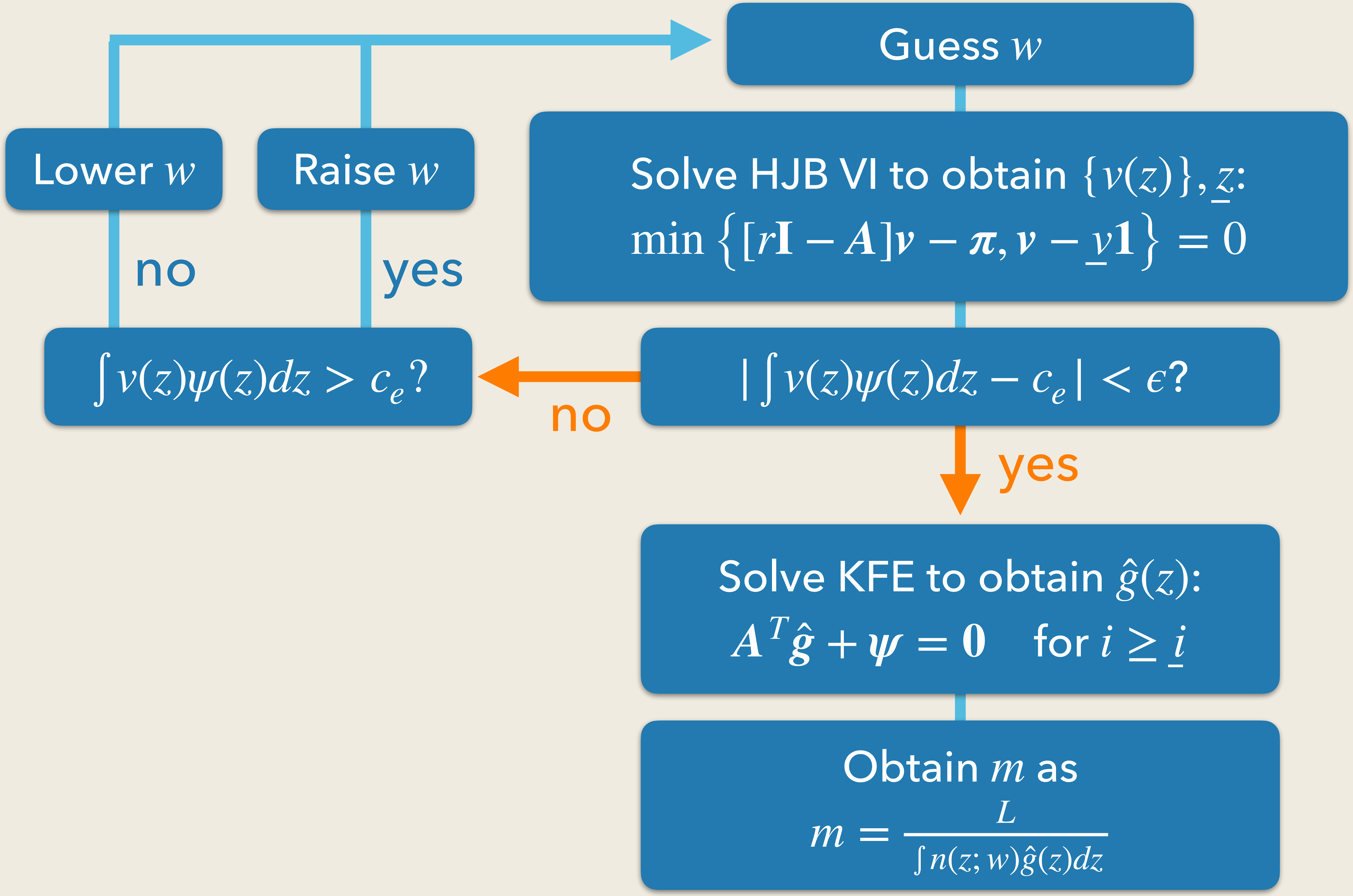
$$\boldsymbol{A} = \begin{bmatrix} -\frac{(\sigma_1)^2}{2(\Delta z)^2} & \frac{(\sigma_1)^2}{2(\Delta z)^2} & 0 & 0 & \cdots & \cdots & 0 \\ -\frac{\mu_2}{\Delta z} + \frac{(\sigma_2)^2}{2(\Delta z)^2} & \frac{\mu_2}{\Delta z} - \frac{(\sigma_2)^2}{(\Delta z)^2} & \frac{(\sigma_3)^2}{2(\Delta z)^2} & 0 & \cdots & \cdots & 0 \\ 0 & -\frac{\mu_3}{\Delta z} + \frac{(\sigma_3)^2}{2(\Delta z)^2} & \frac{\mu_3}{\Delta z} - \frac{(\sigma_3)^2}{(\Delta z)^2} & \frac{(\sigma_3)^2}{2(\Delta z)^2} & 0 & \cdots & 0 \\ 0 & \cdots & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 & -\frac{\mu_{J-1}}{\Delta z} + \frac{(\sigma_{J-1})^2}{2(\Delta z)^2} & \frac{\mu_{J-1}}{\Delta z} - \frac{(\sigma_{J-1})^2}{(\Delta z)^2} & \frac{(\sigma_{J-1})^2}{2(\Delta z)^2} \\ 0 & \cdots & \cdots & \cdots & 0 & -\frac{\mu_J}{\Delta z} + \frac{(\sigma_J)^2}{2(\Delta z)^2} & \frac{\mu_J}{\Delta z} - \frac{(\sigma_J)^2}{2(\Delta z)^2} \end{bmatrix}$$

- This $\boldsymbol{A}$ is the same $\boldsymbol{A}$ that we used in HJB!

Computational Algorithm



Computational Algorithm



```julia
using SparseArrays
using Parameters
using LinearAlgebra
using Distributions
using Plots
@with_kw mutable struct model
    J = 200
    sig = 0.41
    mu = -0.001
    zg = range(0.001,100,length=J)
    dz = zg[2] - zg[1]
    alph = 0.64
    cf = 1
    r = 0.05
    L = 1
    underv = 0.0
    xi = 10
    psig = entry_dist(xi,zg)
    ce = 0.001
    max_iter = 1e3
end
function entry_dist(xi,zg)
    d = Pareto(xi, 1)
    psig = diff(cdf(d,zg))
    psig = [psig; psig[end]]
    psig = psig./sum(psig)./(zg[2]-zg[1])
    return psig
end
function populate_A(param)
    @unpack_model param
    A = spzeros(length(zg),length(zg))
    for (i,z) in enumerate(zg)
        if mu > 0
            A[i,min(i+1,J)] += mu.*z/dz;
            A[i,i] += -mu.*z/dz;
        else
            A[i,i] += mu.*z/dz;
            A[i,max(i-1,1)] += -mu.*z/dz;
        end
        A[i,i] += - (sig*z)^2/dz^2;
        A[i,max(i-1,1)] += 1/2*(sig*z)^2/dz^2;
        A[i,min(i+1,J)] += 1/2*(sig*z)^2/dz^2;
    end
    return A
end
```

```julia
function Howard_Algorithm(param,B,pig)
    @unpack_model param
    iter = 1;
    vold = zeros(length(zg));
    vnew = copy(vold);
    exit_or_not = []
    while iter < max_iter
        val_noexit =  (B*vold .- pig);
        val_exit = vold  .- underv
        exit_or_not =val_noexit  .> val_exit;
        Btilde = B.*(1 .-exit_or_not) + I(J).*(exit_or_not)
        q = pig.*(1 .-exit_or_not) + underv.*(exit_or_not)
        vnew = Btilde\q;
        if norm(vnew - vold) < 1e-6
            break
        end
        vold = copy(vnew)
    end
    return vnew,exit_or_not
end
function solve_HJB_VI(param,w)
    @unpack_model param
    A = populate_A(param)
    B = (r.*I - A);
    ng = (alph./w)^(1/(1-alph)).*zg
    pig = zg.^(1-alph).*ng.^alph .- w.*ng .- cf
    v,exit_or_not = Howard_Algorithm(param,B,pig)
    underz_index = findlast(exit_or_not .> 0 )
    if isnothing(underz_index)
        underz_index = 1
    end
    return v,underz_index,ng,exit_or_not
end
```

Solving Wage

```julia
function solve_w(param)
    @unpack_model param
    w_ub = 10;
    w_lb = 0;
    w = (w_ub + w_lb)/2
    err_free_entry = 100
    iter = 0
    underz_index = 0
    v = [];
    ng = [];
    exit_or_not = [];
    while iter < 1000 && abs(err_free_entry) > 1e-6
        w = (w_ub + w_lb)/2
        v, underz_index,ng,exit_or_not = solve_HJB_VI(param,w)
        err_free_entry = sum(v.*psig.*dz) - ce
        if err_free_entry > 0
            w_lb = w
        else
            w_ub = w
        end
        println("iter: ",iter," w: ",w," err_free_entry: ",err_free_entry)
        iter += 1
    end
    @assert iter < 1000

    return (w = w, v = v, underz_index  = underz_index,ng=ng,exit_or_not=exit_or_not)
end
```

Solving SS Distribution

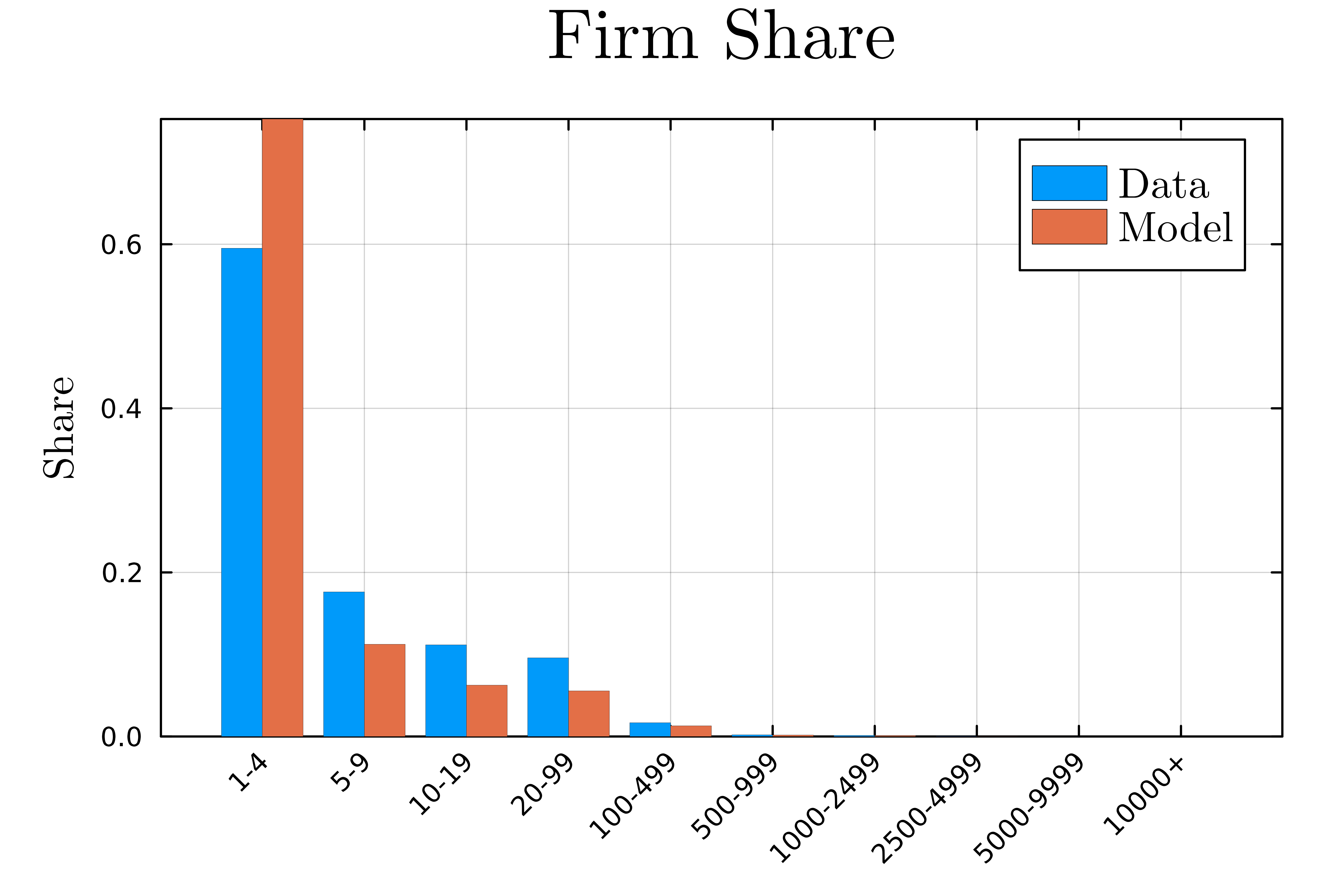
```
function solve_stationary_distribution(param,HJB_result)
    @unpack_model param
    @unpack exit_or_not = HJB_result
    D = spdiagm(0 => exit_or_not)
    I_D = I-D;
    A = populate_A(param)
    tildeA = A*I_D + D;
    B = I_D*psig;
    hatg = (tildeA')\B;
    m = L/sum(hatg.*ng.*dz)
    g = hatg.*m
    return g
end
```

Calibration

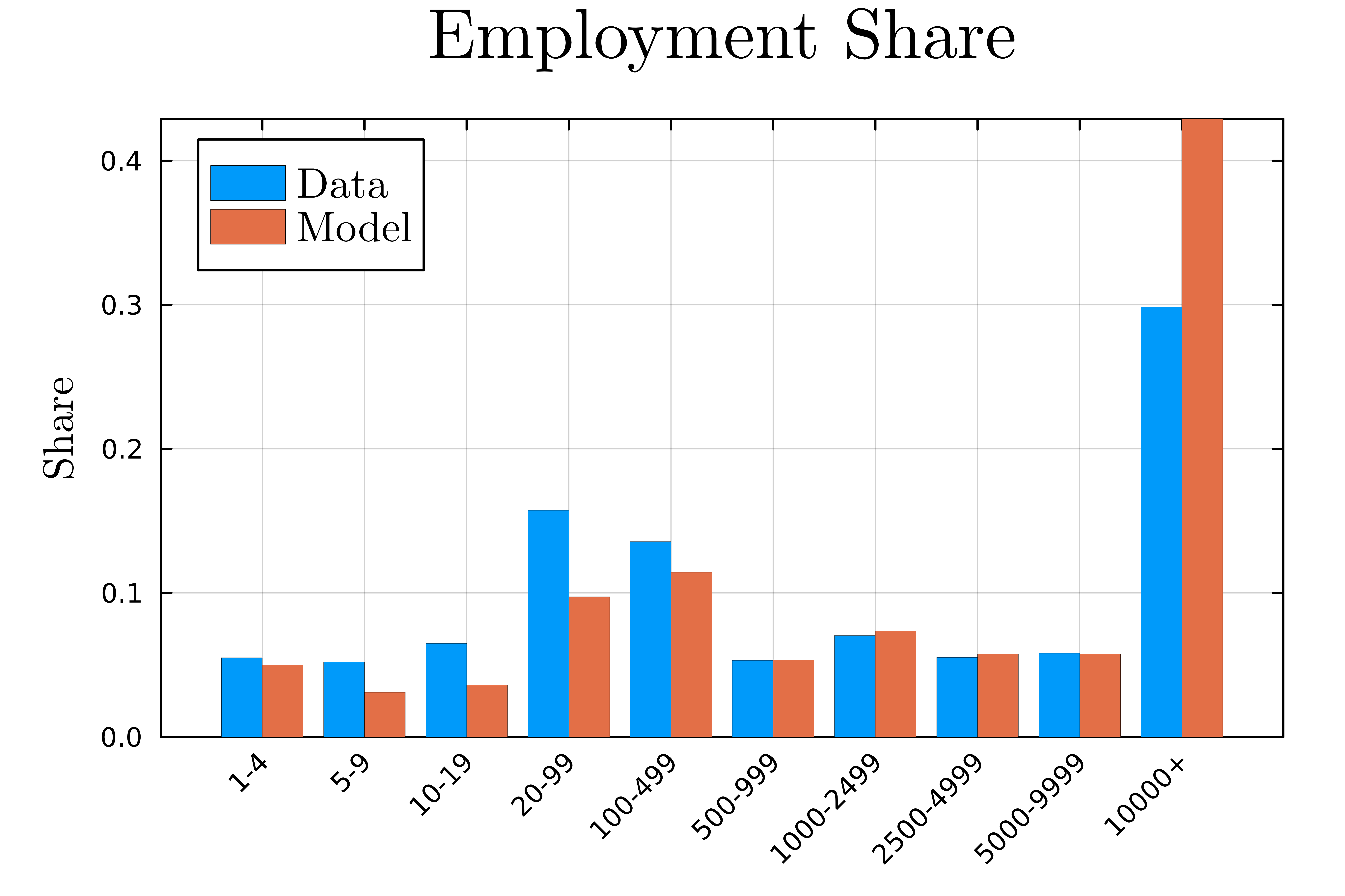
Calibration

- $r = 0.05$ so that the annual interest rate is 5%
- $\alpha = 0.64$ to match labor share
- Assume geometric Brownian motion ($\mu(z) = \mu z, \sigma(z) = \sigma z$):
  - $\sigma = 0.41$ to match std$(\Delta \ln n) = 0.41$ documented in Elsby & Micheales (2013)
  - $\mu = -0.002$ so that the theoretical tail of size distribution is $\zeta = 1.05$
- Assume Pareto distribution for entrants: $\psi(z) = \zeta_e \frac{1}{z}(z/z_{min})^{-\zeta_e}$
- $\zeta_e = 1.1$ to match the entry/exit rate of 9% (in 2021)
- $c_e = 4.9$ so that the average firm size is 23 (in 2021)
- Normalize $\underline{v} = 0$ and $c_f = 0.1$

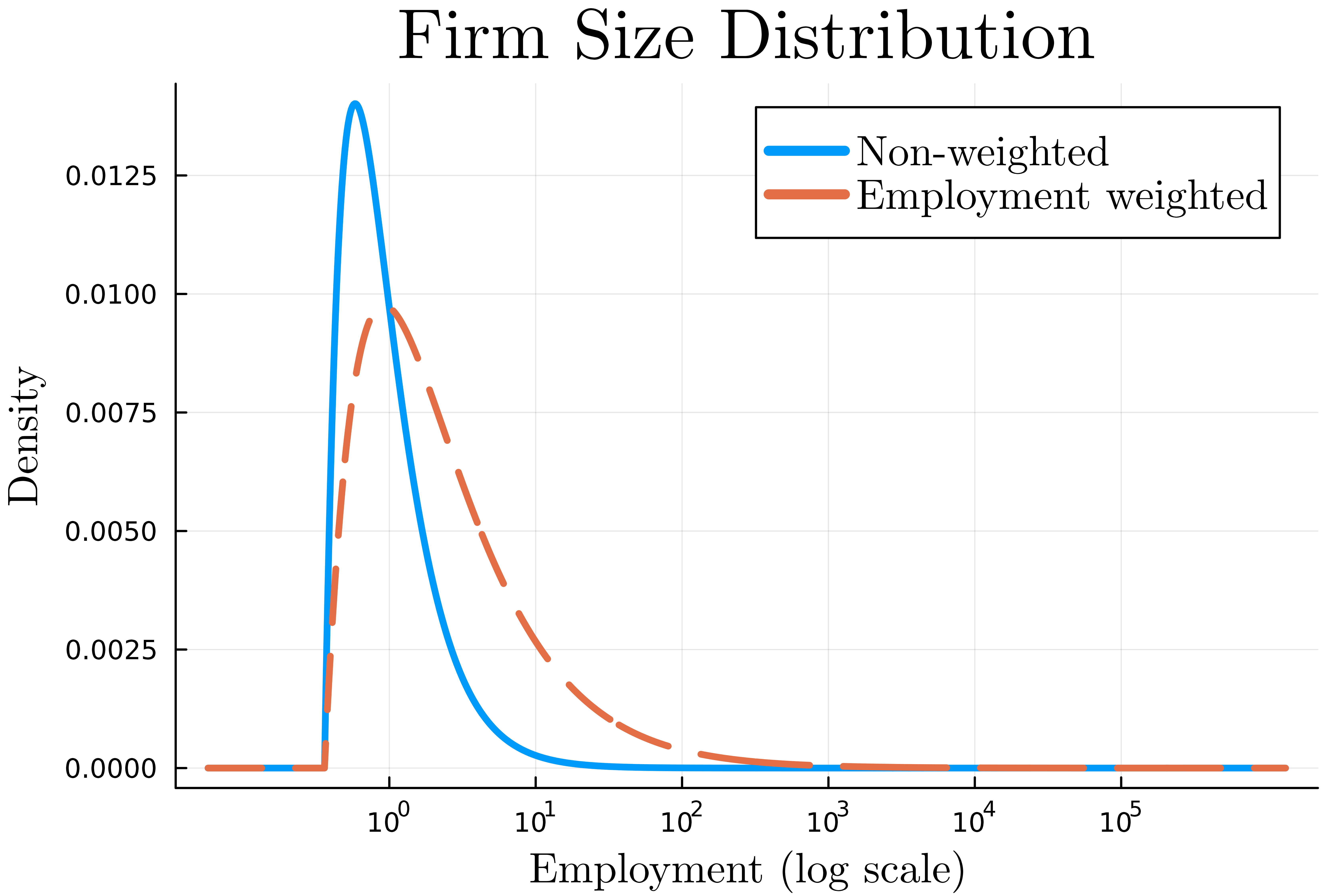
Firm Size Distribution: Data vs. Model



Employment Weighted Size Distribution



Density Plot



Power Law: Model vs. Data

Further Questions

Further Questions

- What is z?
  - Many argue z relates to customer base (e.g., Einav, Klenow, Levin & Murciano-Goroff, 2022; Foster, Haltiwanger, Syverson, 2015; Argente, Fitzgerald, Moreira & Priolo, 2021)
- Does the model get the age distribution right?
  - The average age of Walmart/Amazon size class in the model is 100 years
  - Walmart is 60 years old, and Amazon is 30 years old
- In the model, large firms are large just by luck (ex-ante homogenous). Are they?
  - Hurst & Pugsley (2011) and Pugsley, Sedláček & Sterk (2020) argue not

Appendix:
Discrete vs. Continuous

Transition Matrix vs. Infinitesimal Generator

- With discretized state space, we had

$$\partial_t \boldsymbol{g}_t = \boldsymbol{A}^T \boldsymbol{g}$$

  and $[A]_{ij}$ had a clear interpretation of the transition rate from state i to j

- KFE is just an exact analog with continuous state space

- **Definition**:
  an infinitesimal generator $\mathscr{A}$ for any function $v(n)$ is an operator defined by

$$\mathscr{A}v(n) \equiv \mu(n)\partial_n v(n) + \frac{1}{2}\sigma(n)^2\partial_{nn}^2 v(n) \qquad \text{(A1)}$$

  with a boundary condition $\partial_n v(\underline{n}) = 0$

- If you discretize $\mathscr{A}$, you will obtain A

Transpose vs. Adjoint Operator

- **Definition**: the **inner product** of two functions $v(n)$ and $g(n)$ is

$$\langle v, g \rangle = \int_{\underline{n}}^{\infty} v(n)g(n)dx$$

- **Definition**: the **adjoint** of an operator $\mathscr{A}$ is the operator $\mathscr{A}^*$ that satisfies

$$\langle \mathscr{A}v, g \rangle = \langle v, \mathscr{A}^* g \rangle$$

- Rewrite the KFE using the operator $\mathscr{A}^*$:

$$\partial_t g_t(n) = -\partial_n[\mu(n)g_t(n)] + \frac{1}{2}\partial^2_{nn}\left[\sigma(n)^2 g_t(n)\right] \equiv \mathscr{A}^* g_t(n)$$

with a boundary condition $\mu(\underline{n})g(\underline{n}) + \frac{1}{2}\partial_n[\sigma^2(n)g(n)] = 0$

- **Result**: $\mathscr{A}^*$ is the adjoint of $\mathscr{A}$ as defined in (A1)

- Adjoint is just an exact analog of transpose. Consider two vectors, $\boldsymbol{v}$ & $\boldsymbol{g}$:

$$\langle \boldsymbol{v}, \boldsymbol{g} \rangle \equiv \boldsymbol{v}^T \boldsymbol{g}, \quad \langle \boldsymbol{A}\boldsymbol{v}, \boldsymbol{g} \rangle = \langle \boldsymbol{v}, \boldsymbol{A}^* \boldsymbol{g} \rangle \quad \Leftrightarrow \quad A^* = A^T$$

Proof

$$
\begin{aligned}
\langle v, \mathscr{A}^* g\rangle &= \int_{\underline{n}}^{\infty} v(n)\left(-\partial_x[\mu(n)g(n)] + \frac{1}{2}\partial_{xx}[\sigma(x)^2 g(x)]\right) \\
&= \left[v(x)\left(-\mu(x)g(x) + \frac{1}{2}\partial_x[\sigma(x)^2 g(x)]\right)\right]_{\underline{n}}^{\infty} - \int_{\underline{n}}^{\infty} v'(x)\left(-\mu(x)g(x) + \frac{1}{2}\partial_x[\sigma(x)^2 g(x)]\right) dx \\
&= \left[v(x)\left(-\mu(x)g(x)+\frac{1}{2}\partial_x[\sigma(x)^2 g(x)]\right) - v'(x)\frac{1}{2}\sigma(x)^2 g(x)\right]_{\underline{n}}^{\infty} \\
&\qquad + \int_{\underline{n}}^{\infty} \mu(x)v'(x)g(x)dx + \int_{\underline{n}}^{\infty} \frac{1}{2}\sigma(x)^2 v''(x)g(x)dx \\
&= \int_{\underline{n}}^{\infty} \left[\mu(x)v'(x)g(x)dx + \frac{1}{2}\sigma(x)^2 v''(x)g(x)\right] dx \\
&= \langle \mathscr{A} v, g\rangle
\end{aligned}
$$

where we used integration parts twice and boundary conditions