

Advanced Macroeconomics II

Handout 5 - Integration

Sergio Ocampo

Western University

October 21, 2020

Short recap

Prototypical DP problem:

$$\begin{aligned} V(k, z) &= \max_{\{c, k'\}} u(c) + \beta E \left[V(k', z') | z \right] \\ \text{s.t. } c + k' &= f(k, z) \\ z' &= h(z, \eta); \eta \text{ stochastic} \end{aligned}$$

- ▶ We are looking for functions $\mathbf{V}, \mathbf{g}^c, \mathbf{g}^k$: We cannot solve this.

We need to solve an approximate problem:

- ▶ Approximate continuous function: **Interpolation**
 - ▶ Requires “exact” solution of maximization problem: **Optimization**
 - ▶ Requires computing expectations: **Integration**

Integration - Many options

1. Monte Carlo integration

Integration - Many options

1. Monte Carlo integration
2. Quadrature methods

Integration - Many options

1. Monte Carlo integration
2. Quadrature methods
3. Discretize state space

Integration - Many options

1. Monte Carlo integration
2. Quadrature methods
3. Discretize state space
 - ▶ Tauchen (1986)
 - ▶ Tauchen & Hussey (1991)
 - ▶ Rouwenhorst (2008)
 - ▶ Gaussian mixture (i.a. Civalé, Diez-Catalan & Fazilet, 2017)

Monte Carlo Integration

Monte Carlo integration

- ▶ Idea: Exploit the law of large numbers

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i) = E[f(x)] = \int f(x) dG(x)$$

- ▶ Approximate your integral with a sum

Monte Carlo integration

- ▶ Idea: Exploit the law of large numbers

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i) = E[f(x)] = \int f(x) dG(x)$$

- ▶ Approximate your integral with a sum
- ▶ Key: Where to evaluate it... we need draws from $x \sim G$
 - ▶ Actually, we need a lot of draws
 - ▶ Monte Carlo relies on large numbers to get relative frequencies right
 - ▶ If density of $x \sim G$ at a is higher, there will be more draws x_i close to a

Monte Carlo integration

- ▶ Idea: Exploit the law of large numbers

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i) = E[f(x)] = \int f(x) dG(x)$$

- ▶ Approximate your integral with a sum
- ▶ Key: Where to evaluate it... we need draws from $x \sim G$
 - ▶ Actually, we need a lot of draws
 - ▶ Monte Carlo relies on large numbers to get relative frequencies right
 - ▶ If density of $x \sim G$ at a is higher, there will be more draws x_i close to a
- ▶ Monte Carlo is generally costly, requires too many function evaluations.

Monte Carlo integration - Expectations

Algorithm 1: Expectation by Monte Carlo

input : Number of seeds (N_0) and number of candidates (N^*)

output: $E[V(k', z') | z]$ with $z' = h(z, \eta)$ and $\eta \sim G$

1. Generate N random draws for $\eta \sim G$. Call them $\{\eta_i\}_{i=1}^N$

Note: Do this once at the beginning of the code ;

for $i=1:N$ **do**

2. Evaluate $f_i = V(k', h(z, \eta_i))$

Note: This step requires interpolation of V in the z direction ;

3. Return average: $E[V(k', z') | z] \approx \frac{1}{N} \sum_i f_i$
-

Monte Carlo integration - When to use it?

- ▶ This is rarely the best way to go for expectations
 - ▶ Requires high N to provide good approximation
 - ▶ That is a lot of interpolations for each evaluation of the expectation

Monte Carlo integration - When to use it?

- ▶ This is rarely the best way to go for expectations
 - ▶ Requires high N to provide good approximation
 - ▶ That is a lot of interpolations for each evaluation of the expectation
- ▶ Sometimes evaluations are not that expensive
 - ▶ Then Monte Carlo is cheap and easy

Monte Carlo integration - When to use it?

- ▶ This is rarely the best way to go for expectations
 - ▶ Requires high N to provide good approximation
 - ▶ That is a lot of interpolations for each evaluation of the expectation
- ▶ Sometimes evaluations are not that expensive
 - ▶ Then Monte Carlo is cheap and easy
- ▶ It is a good tool for other integrals
 - ▶ Model simulation (with and without heterogeneity)

Monte Carlo integration - When to use it?

- ▶ This is rarely the best way to go for expectations
 - ▶ Requires high N to provide good approximation
 - ▶ That is a lot of interpolations for each evaluation of the expectation
- ▶ Sometimes evaluations are not that expensive
 - ▶ Then Monte Carlo is cheap and easy
- ▶ It is a good tool for other integrals
 - ▶ Model simulation (with and without heterogeneity)
- ▶ It is the easiest method to parallelize
 - ▶ Depends on your computational resources

Quadrature Methods

Gaussian Quadrature methods

- ▶ Idea: Approximate \int with \sum , like in Monte Carlo, but with less points

$$\int_a^b f(x) dx = \sum_{i=1}^N \omega_i f(x_i)$$

- ▶ Key: Choose where to evaluate $\{x_i\}$ and appropriate weights $\{\omega_i\}$

Gaussian Quadrature methods

- ▶ Idea: Approximate \int with \sum , like in Monte Carlo, but with less points

$$\int_a^b f(x) dx = \sum_{i=1}^N \omega_i f(x_i)$$

- ▶ Key: Choose where to evaluate $\{x_i\}$ and appropriate weights $\{\omega_i\}$
- ▶ Warning: GQ only works well if f is well approximated by an N -degree polynomial

Gaussian Quadrature methods

- ▶ Idea: Approximate \int with \sum , like in Monte Carlo, but with less points

$$\int_a^b f(x) dx = \sum_{i=1}^N \omega_i f(x_i)$$

- ▶ Key: Choose where to evaluate $\{x_i\}$ and appropriate weights $\{\omega_i\}$
- ▶ Warning: GQ only works well if f is well approximated by an N -degree polynomial

Note: For equally spaced grid points and non-smooth functions use Romberg integration (see Numerical Recipes, Sec. 4.3)

Gaussian Quadrature methods

Objective: Get a method with exact results for integrals of the type:

$$\int_a^b f(x) dx = \int_a^b W(x) h(x) dx = \sum_{i=1}^N w_i h(x_i)$$

where $h(\cdot)$ is a polynomial and $W(\cdot)$ is a weighting function

Gaussian Quadrature methods

Objective: Get a method with exact results for integrals of the type:

$$\int_a^b f(x) dx = \int_a^b W(x) h(x) dx = \sum_{i=1}^N w_i h(x_i)$$

where $h(\cdot)$ is a polynomial and $W(\cdot)$ is a weighting function

- ▶ This method will give approximate results for functions (f) that are well approximated by a polynomial (h) times a weighting function (W)
- ▶ We only need that our original function satisfies: $f(x) \approx W(x) h(x)$
- ▶ We can always choose $W(x) = 1$ and then $h(x) = f(x)$

Gaussian Quadrature methods

Objective: Get a method with exact results for integrals of the type:

$$\int_a^b f(x) dx = \int_a^b W(x) h(x) dx = \sum_{i=1}^N w_i h(x_i)$$

where $h(\cdot)$ is a polynomial and $W(\cdot)$ is a weighting function

- ▶ This method will give approximate results for functions (f) that are well approximated by a polynomial (h) times a weighting function (W)
- ▶ We only need that our original function satisfies: $f(x) \approx W(x) h(x)$
- ▶ We can always choose $W(x) = 1$ and then $h(x) = f(x)$
- ▶ We don't actually need to know h . We can define $h(x) = f(x_i)/W(x_i)$:

$$\int_a^b f(x) dx = \int_a^b W(x) h(x) dx \approx \sum_{i=1}^N \omega_i f(x_i) \quad \text{where } \omega_i = \frac{w_i}{W(x_i)}$$

Gaussian Quadrature methods

Algorithm 2: Gaussian Quadrature

input : Number of points N , integrand f , weighting function W

output: Points $\{x_i\}$, weights $\{\omega_i\}$, integral $\int_a^b f(x)dx \approx \sum_i \omega_i f(x_i)$

1. Choose a weighting function W ;
2. Construct the family of orthonormal polynomials wrt W up to degree N ;
3. Obtain roots of the polynomial of degree N in $[a, b]$

These roots are the points $\{x_i\}$;

4. Evaluate the auxiliary weights $w_i = \frac{\langle p_{N-1} | p_{N-1} \rangle}{p_{N-1}(x_i) p'_N(x_i)}$

The weights we look for are $\omega_i = \frac{w_i}{W(x_i)}$;

5. Evaluate the integral: $\int_a^b f(x)dx \approx \sum_i \omega_i f(x_i)$
-

Gaussian Quadrature methods

We happen to know the solution for a bunch of weighting functions:

Gaussian Quadrature methods

We happen to know the solution for a bunch of weighting functions:

- ▶ Gauss-Legendre: $W(x) = 1$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $(i+1)P_{i+1} = (2i+1)xP_i - iP_{i-1}$

Gaussian Quadrature methods

We happen to know the solution for a bunch of weighting functions:

- ▶ Gauss-Legendre: $W(x) = 1$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $(i+1)P_{i+1} = (2i+1)xP_i - iP_{i-1}$
- ▶ Gauss-Chebyshev: $W(x) = 1/\sqrt{1-x^2}$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $T_{i+1} = 2xT_i - T_{i-1}$

Gaussian Quadrature methods

We happen to know the solution for a bunch of weighting functions:

- ▶ Gauss-Legendre: $W(x) = 1$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $(i+1)P_{i+1} = (2i+1)xP_i - iP_{i-1}$
- ▶ Gauss-Chebyshev: $W(x) = 1/\sqrt{1-x^2}$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $T_{i+1} = 2xT_i - T_{i-1}$
- ▶ Gauss-Hermite: $W(x) = e^{-x^2}$ for $x \in [-\infty, \infty]$
 - ▶ Polynomial recursion: $H_{i+1} = 2xH_i - 2iH_{i-1}$

Gaussian Quadrature methods

We happen to know the solution for a bunch of weighting functions:

- ▶ Gauss-Legendre: $W(x) = 1$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $(i+1)P_{i+1} = (2i+1)xP_i - iP_{i-1}$
- ▶ Gauss-Chebyshev: $W(x) = 1/\sqrt{1-x^2}$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $T_{i+1} = 2xT_i - T_{i-1}$
- ▶ Gauss-Hermite: $W(x) = e^{-x^2}$ for $x \in [-\infty, \infty]$
 - ▶ Polynomial recursion: $H_{i+1} = 2xH_i - 2iH_{i-1}$

See Numerical Recipes for more results (including weights)

Gaussian Quadrature methods

We happen to know the solution for a bunch of weighting functions:

- ▶ Gauss-Legendre: $W(x) = 1$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $(i+1)P_{i+1} = (2i+1)xP_i - iP_{i-1}$
- ▶ Gauss-Chebyshev: $W(x) = 1/\sqrt{1-x^2}$ for $x \in [-1, 1]$
 - ▶ Polynomial recursion: $T_{i+1} = 2xT_i - T_{i-1}$
- ▶ Gauss-Hermite: $W(x) = e^{-x^2}$ for $x \in [-\infty, \infty]$
 - ▶ Polynomial recursion: $H_{i+1} = 2xH_i - 2iH_{i-1}$

See Numerical Recipes for more results (including weights)

Note: Use Gauss-Hermite for integrating Gaussian shocks.

Let $h(x) = \frac{1}{\sqrt{\pi}} V(k', h(z, \sqrt{2}\sigma_\eta x + \mu_\eta))$ and

$$W(x) = e^{-x^2} \propto \Phi(x)$$

Careful with extrapolation... $x \in (-\infty, \infty)$

Gaussian Quadrature - Problems

- ▶ How good is your approximation? God knows!
 - ▶ No practical error estimate...
 - ▶ There are theoretical estimates, they can be very large

Gaussian Quadrature - Problems

- ▶ How good is your approximation? God knows!
 - ▶ No practical error estimate...
 - ▶ There are theoretical estimates, they can be very large
- ▶ From NR: “[$W(x)$ is] ready to give high-order accuracy to integrands of the form polynomials times $W(x)$, and ready to *deny* high order accuracy to integrands that are otherwise perfectly smooth and well behaved.”

Gaussian Quadrature - Problems

- ▶ How good is your approximation? God knows!
 - ▶ No practical error estimate...
 - ▶ There are theoretical estimates, they can be very large
- ▶ From NR: “[$W(x)$ is] ready to give high-order accuracy to integrands of the form polynomials times $W(x)$, and ready to *deny* high order accuracy to integrands that are otherwise perfectly smooth and well behaved.”
- ▶ Methods are not nested: going from N to $N + 1$ changes all $\{x_i, w_i\}$
- ▶ Bad performance when function has kinks, or doesn't look polynomial

Gauss-Kronrod Quadrature

- ▶ This should be your go-to method

Gauss-Kronrod Quadrature

- ▶ This should be your go-to method
- ▶ Uses nested Gaussian quadrature to iteratively evaluate the integral
 - ▶ The nested part helps by reusing old function evaluations

Gauss-Kronrod Quadrature

- ▶ This should be your go-to method
- ▶ Uses nested Gaussian quadrature to iteratively evaluate the integral
 - ▶ The nested part helps by reusing old function evaluations
- ▶ Provides a practical error bound from the change in the integral
- ▶ Better than Gaussian quadrature if function is not polynomial

Discretizing the State Space

General idea

- ▶ Instead of approximating the integral approximate the stochastic process
 - ▶ Discretize z (and $h(z, \eta)$) instead of η
 - ▶ Approximate process for z with a discrete Markov process

General idea

- ▶ Instead of approximating the integral approximate the stochastic process
 - ▶ Discretize z (and $h(z, \eta)$) instead of η
 - ▶ Approximate process for z with a discrete Markov process
- ▶ Markov process characterized by:
 - ▶ Discrete state space: $z \in \{z_1, \dots, z_N\}$
 - ▶ Transition matrix: $\Pi = [\pi_{ij}]$, s.t. $\Pr(z' = z_j | z = z_i) = \pi_{ij}$, $\sum_j \pi_{ij} = 1$

General idea

- ▶ Instead of approximating the integral approximate the stochastic process
 - ▶ Discretize z (and $h(z, \eta)$) instead of η
 - ▶ Approximate process for z with a discrete Markov process
- ▶ Markov process characterized by:
 - ▶ Discrete state space: $z \in \{z_1, \dots, z_N\}$
 - ▶ Transition matrix: $\Pi = [\pi_{ij}]$, s.t. $\Pr(z' = z_j | z = z_i) = \pi_{ij}$, $\sum_j \pi_{ij} = 1$
- ▶ Compute expectation:

$$E \left[V(k', z') | z = z_i \right] = \sum_{j=1}^N \pi_{ij} V(k', z_j)$$

Note: No approximation. No interpolation.

Key problem: How to get transition matrix?

Two approaches:

Key problem: How to get transition matrix?

Two approaches:

1. Full blown estimation

- ▶ Set a grid for z : $\{z_1, \dots, z_N\}$
- ▶ Estimate transition matrix to match desired moments
- ▶ Moments of z or moments of the model
- ▶ $N(N - 1)$ numbers to estimate

Key problem: How to get transition matrix?

Two approaches:

1. Full blown estimation

- ▶ Set a grid for z : $\{z_1, \dots, z_N\}$
- ▶ Estimate transition matrix to match desired moments
- ▶ Moments of z or moments of the model
- ▶ $N(N - 1)$ numbers to estimate

2. Parametrize process for z

- ▶ Typical assumption is AR(1): $z' = h(z, \eta) = \rho z + \eta$
- ▶ Use a method to choose Π to match properties of AR(1)
- ▶ Only have to choose ρ and σ_η

Tauchen (1986)

$$z' = \rho z + \eta \quad \eta \sim N(0, \sigma_\eta)$$

- ▶ Note that $E[z] = 0$ and that shocks are symmetric

Tauchen (1986)

$$z' = \rho z + \eta \quad \eta \sim N(0, \sigma_\eta)$$

- ▶ Note that $E[z] = 0$ and that shocks are symmetric
- ▶ Start from an equally spaced grid centered at 0: $\{z_1, \dots, z_N\}$
 - ▶ Heuristic: Extend grid Ω standard deviations around mean (recall $\sigma_z = \sigma_\eta / \sqrt{1 - \rho^2}$)
 $z_1 = -\Omega\sigma_z, \dots, z_n = z_{n-1} + \Delta_z, \dots, z_N = \Omega\sigma_z$ where: $\Delta_z = \Omega\sigma_z / (N-1)$
- ▶ Usually $\Omega = 3$, but this depends on what you are modeling

Tauchen (1986)

$$z' = \rho z + \eta \quad \eta \sim N(0, \sigma_\eta)$$

- ▶ Note that $E[z] = 0$ and that shocks are symmetric
- ▶ Start from an equally spaced grid centered at 0: $\{z_1, \dots, z_N\}$
 - ▶ Heuristic: Extend grid Ω standard deviations around mean (recall $\sigma_z = \sigma_\eta / \sqrt{1-\rho^2}$)

$$z_1 = -\Omega\sigma_z, \dots, z_n = z_{n-1} + \Delta_z, \dots, z_N = \Omega\sigma_z \text{ where: } \Delta_z = \Omega\sigma_z / (N-1)$$

- ▶ Usually $\Omega = 3$, but this depends on what you are modeling
- ▶ Fill in transition probabilities from normal distribution:

$$\pi_{ij} = \begin{cases} \Phi\left(\frac{z_j - \rho z_i + \Delta_z/2}{\sigma_\eta}\right) & \text{if } j = 1 \\ \Phi\left(\frac{z_j - \rho z_i + \Delta_z/2}{\sigma_\eta}\right) - \Phi\left(\frac{z_j - \rho z_i - \Delta_z/2}{\sigma_\eta}\right) & \text{if } j = 2, \dots, N-1 \\ 1 - \Phi\left(\frac{z_j - \rho z_i - \Delta_z/2}{\sigma_\eta}\right) & \text{if } j = N \end{cases}$$

Tauchen (1986)

Algorithm 3: Tauchen (1986)

input : Number of points N , width of grid Ω , process parameters ρ, σ_η

output: Discrete approximation of $z' = \rho z + \eta$, $\eta \sim N(0, \sigma_\eta)$: z grid and Π

1. Construct grid: $z = \text{range}(-\Omega\sigma_z, \Omega\sigma_z, \text{length} = N)$, $\Delta_z = 2\Omega\sigma_z/N-1$;
where $\sigma_z = \sigma_\eta / \sqrt{1-\rho^2}$;

for $i=1:N, j=1:N$ **do**

2. Fill in π_{ij} as:

$$\pi_{ij} = \begin{cases} \Phi\left(\frac{z_j - \rho z_i + \Delta_z/2}{\sigma_\eta}\right) & \text{if } j = 1 \\ \Phi\left(\frac{z_j - \rho z_i + \Delta_z/2}{\sigma_\eta}\right) - \Phi\left(\frac{z_j - \rho z_i - \Delta_z/2}{\sigma_\eta}\right) & \text{if } j = 2, \dots, N-1 \\ 1 - \Phi\left(\frac{z_j - \rho z_i - \Delta_z/2}{\sigma_\eta}\right) & \text{if } j = N \end{cases}$$

Tauchen & Hussey (1991)

Exploit Gaussian Quadrature

- ▶ Since shocks are normally distributed use Gauss-Hermite

Tauchen & Hussey (1991)

Exploit Gaussian Quadrature

- ▶ Since shocks are normally distributed use Gauss-Hermite
- ▶ Without persistence (say $z' \sim N(\mu, \sigma)$) we get:

$$E \left[V \left(k', z' \right) \right] = \int V \left(k', z' \right) \phi \left(\frac{z' - \mu}{\sigma} \right) dz' \approx \sum_{j=1}^N \frac{w_j}{\sqrt{\pi}} V \left(k', \sqrt{2}\sigma x_j + \mu \right)$$

with $\{x_i\}$ the roots of the Hermite polynomials and $\{w_i\}$ the weights

Tauchen & Hussey (1991)

Exploit Gaussian Quadrature

- ▶ Since shocks are normally distributed use Gauss-Hermite
- ▶ Without persistence (say $z' \sim N(\mu, \sigma)$) we get:

$$E \left[V(k', z') \right] = \int V(k', z') \phi \left(\frac{z' - \mu}{\sigma} \right) dz' \approx \sum_{j=1}^N \frac{w_j}{\sqrt{\pi}} V(k', \sqrt{2}\sigma x_j + \mu)$$

with $\{x_i\}$ the roots of the Hermite polynomials and $\{w_i\}$ the weights

- ▶ With persistence we get: $E \left[V(k', z') | z \right] = \int V(k', z') \phi \left(\frac{z' - \rho z}{\sigma_\eta} \right) dz'$

Tauchen & Hussey (1991)

Exploit Gaussian Quadrature

- ▶ Since shocks are normally distributed use Gauss-Hermite
- ▶ Without persistence (say $z' \sim N(\mu, \sigma)$) we get:

$$E \left[V(k', z') \right] = \int V(k', z') \phi \left(\frac{z' - \mu}{\sigma} \right) dz' \approx \sum_{j=1}^N \frac{w_j}{\sqrt{\pi}} V(k', \sqrt{2}\sigma x_j + \mu)$$

with $\{x_i\}$ the roots of the Hermite polynomials and $\{w_i\}$ the weights

- ▶ With persistence we get: $E \left[V(k', z') | z \right] = \int V(k', z') \phi \left(\frac{z' - \rho z}{\sigma_\eta} \right) dz'$

Issue: Conditional mean of z' varies with z .

- ▶ We would have to evaluate objective for different values with each z

Tauchen & Hussey (1991)

Solution: Express integral wrt unconditional normal, apply formula

$$\begin{aligned} E \left[V \left(k', z' \right) | z \right] &= \int \left(V \left(k', z' \right) \frac{\phi \left(\frac{z' - \rho z}{\sigma_\eta} \right)}{\phi \left(\frac{z'}{\sigma_\eta} \right)} \right) \phi \left(\frac{z'}{\sigma_\eta} \right) dz' \\ &\approx \sum_{j=1}^N \frac{w_j}{\sqrt{\pi}} V \left(k', \sqrt{2} \sigma_\eta x_j \right) \frac{\phi \left(\frac{\sqrt{2} \sigma_\eta x_j - \rho z}{\sigma_\eta} \right)}{\phi \left(\frac{\sqrt{2} \sigma_\eta x_j}{\sigma_\eta} \right)} \end{aligned}$$

Tauchen & Hussey (1991)

Solution: Express integral wrt unconditional normal, apply formula

$$\begin{aligned} E \left[V(k', z') | z \right] &= \int \left(V(k', z') \frac{\phi\left(\frac{z' - \rho z}{\sigma_\eta}\right)}{\phi\left(\frac{z'}{\sigma_\eta}\right)} \right) \phi\left(\frac{z'}{\sigma_\eta}\right) dz' \\ &\approx \sum_{j=1}^N \frac{w_j}{\sqrt{\pi}} V(k', \sqrt{2}\sigma_\eta x_j) \frac{\phi\left(\frac{\sqrt{2}\sigma_\eta x_j - \rho z}{\sigma_\eta}\right)}{\phi\left(\frac{\sqrt{2}\sigma_\eta x_j}{\sigma_\eta}\right)} \end{aligned}$$

► Fixed grid points: $z_i = \sqrt{2}\sigma_\eta x_i$, where $\{x_i\}$ are Gauss-Hermite nodes

► Define $\omega_i = w_i/\sqrt{\pi}$, where $\{w_i\}$ are Gauss-Hermite weights

► Probabilities: $\pi_{ij} = \frac{\phi\left(\frac{z_j - \rho z_i}{\sigma_\eta}\right)}{\phi\left(\frac{z_j}{\sigma_\eta}\right)} \frac{\omega_i}{s_i}$, where $s_i = \sum_n \frac{\phi\left(\frac{z_n - \rho z_i}{\sigma_\eta}\right)}{\phi\left(\frac{z_n}{\sigma_\eta}\right)} \omega_i$

Tauchen & Hussey (1991)

Algorithm 4: Tauchen & Hussey (1991)

input : Number of points N , process parameters ρ, σ_η

output: Discrete approximation of $z' = \rho z + \eta$, $\eta \sim N(0, \sigma_\eta)$: z grid and Π

1. Obtain Gauss-Hermite nodes and weights $\{x, w\}$;

2. Define grid as $z_i = \sqrt{2}\sigma_\eta x_i$;

for $i=1:N, j=1:N$ **do**

 3. Fill in $\pi_{ij} = \frac{\phi\left(\frac{z_j - \rho z_i}{\sigma_\eta}\right)}{\phi\left(\frac{z_j}{\sigma_\eta}\right)} \frac{\omega_i}{s_i}$ where $\omega_i = w_i/\sqrt{\pi}$ and $s_i = \sum_n \frac{\phi\left(\frac{z_n - \rho z_i}{\sigma_\eta}\right)}{\phi\left(\frac{z_n}{\sigma_\eta}\right)} \omega_i$;

Rouwenhorst (1995)

Objective: Construct a Markov Process that matches moments of z'

- ▶ Instead of starting from a grid and getting transition probabilities

Rouwenhorst (1995)

Objective: Construct a Markov Process that matches moments of z'

- ▶ Instead of starting from a grid and getting transition probabilities

Steps:

1. Construct a particular type of Markov Process
 - ▶ Construction is recursive

Rouwenhorst (1995)

Objective: Construct a Markov Process that matches moments of z'

- ▶ Instead of starting from a grid and getting transition probabilities

Steps:

1. Construct a particular type of Markov Process
 - ▶ Construction is recursive
2. Derive properties of the Markov Process
 - ▶ Find stationary distribution and conditional moments

Rouwenhorst (1995)

Objective: Construct a Markov Process that matches moments of z'

- ▶ Instead of starting from a grid and getting transition probabilities

Steps:

1. Construct a particular type of Markov Process
 - ▶ Construction is recursive
2. Derive properties of the Markov Process
 - ▶ Find stationary distribution and conditional moments
3. Match moments from AR(1) process

Rouwenhorst (1995) - Markov Process

- ▶ Pick size of the grid: $N \geq 2$

Rouwenhorst (1995) - Markov Process

- ▶ Pick size of the grid: $N \geq 2$
- ▶ Let $\{z_1, \dots, z_N\}$ be an equally spaced grid such that $z_1 = -\psi$ and $z_N = \psi$
 - ▶ We need to find ψ . Clearly $\Delta_z = \frac{2\psi}{N-1}$.

Rouwenhorst (1995) - Markov Process

- ▶ Pick size of the grid: $N \geq 2$
- ▶ Let $\{z_1, \dots, z_N\}$ be an equally spaced grid such that $z_1 = -\psi$ and $z_N = \psi$
 - ▶ We need to find ψ . Clearly $\Delta_z = \frac{2\psi}{N-1}$.
- ▶ Construct transition matrix recursively:
$$\Pi_2 = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}$$
$$\Pi_N = p \begin{bmatrix} \Pi_{N-1} & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix} + (1-p) \begin{bmatrix} \vec{0} & \Pi_{N-1} \\ 0 & \vec{0}^T \end{bmatrix} + (1-q) \begin{bmatrix} \vec{0}^T & 0 \\ \Pi_{N-1} & \vec{0} \end{bmatrix} + q \begin{bmatrix} 0 & \vec{0}^T \\ \vec{0} & \Pi_{N-1} \end{bmatrix}$$
where $\vec{0}$ is an $(N-1) \times 1$ zero vector. We need to find p and q .
- ▶ Divide all rows by 2 to ensure they sum to 1 (except top and bottom)

Rouwenhorst (1995) - Moments

Results from Kopecky & Suen (2010)

Conditional Mean	$E[z' z = z_i]$	$(q - p)\psi + (p + q - 1)z_i$
Conditional Var	$V[z' z = z_i]$	$\frac{4\psi^2}{(N-1)^2} [(N - i)(1 - p)p + (i - 1)q(1 - q)]$
Unconditional Mean	$E[z]$	$\frac{q-p}{2-(p+q)}\psi$
Unconditional Var	$V[z] = E[z^2]$	$\psi^2 \left[1 - 4s(1 - s) + \frac{4s(1-s)}{N-1} \right]; \text{ where } s = \frac{1-q}{2-(p+q)}$
Autocovariance	$Cov[z', z]$	$(p + q - 1)V[z]$
Autocorrelation	$Corr[z', z]$	$p + q - 1$

Moreover, the stationary distribution is Binomial $(N - 1, 1 - s)$.

Rouwenhorst (1995) - Matching the AR(1)

$$z' = \rho z + \eta \quad \text{where } \eta \sim N(0, \sigma_\eta)$$

Rouwenhorst (1995) - Matching the AR(1)

$$z' = \rho z + \eta \quad \text{where } \eta \sim N(0, \sigma_\eta)$$

- Conditional moments are immediate:

$$E[z' | z = z_i] = \rho z_i \quad V[z' | z = z_i] = \sigma_\eta^2$$

Also the autocorrelation: $\text{Corr}[z', z] = \rho$

Rouwenhorst (1995) - Matching the AR(1)

$$z' = \rho z + \eta \quad \text{where } \eta \sim N(0, \sigma_\eta)$$

- ▶ Conditional moments are immediate:

$$E[z' | z = z_i] = \rho z_i \quad V[z' | z = z_i] = \sigma_\eta^2$$

Also the autocorrelation: $\text{Corr}[z', z] = \rho$

- ▶ Unconditional moments: $E[z] = 0 \quad V[z] = \frac{\sigma_\eta^2}{1-\rho^2}$

Rouwenhorst (1995) - Matching the AR(1)

$$z' = \rho z + \eta \quad \text{where } \eta \sim N(0, \sigma_\eta)$$

- ▶ Conditional moments are immediate:

$$E[z' | z = z_i] = \rho z_i \quad V[z' | z = z_i] = \sigma_\eta^2$$

Also the autocorrelation: $\text{Corr}[z', z] = \rho$

- ▶ Unconditional moments: $E[z] = 0 \quad V[z] = \frac{\sigma_\eta^2}{1-\rho^2}$
- ▶ Matching moments gives:

$$p = q \quad p + q - 1 = \rho \longrightarrow p = q = \frac{1 + \rho}{2}$$

$$\sigma_\eta^2 = \frac{4\psi^2}{(N-1)^2} [(N-i)(1-p)p + (i-1)q(1-q)] \longrightarrow \psi = \sqrt{N-1} \frac{\sigma_\eta}{\sqrt{1-\rho^2}}$$

Rouwenhorst (1995)

Algorithm 5: Rouwenhorst (1995): Discretize AR(1)

Function Rouwenhorst(N, ρ, σ_η):

1. Define $p = 1 + \rho/2$ and $\psi = \sigma_\eta \sqrt{N-1/1-\rho^2}$
2. Construct grid: $z = \text{range}(-\psi, \psi, \text{length} = N)$, $\Delta_z = 2\psi/N-1$
- if $N==2$ then
 - 3.1. Define $\Pi_2 = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}$
- else
 - 3.2.1. $\Pi_{N-1} = \text{Rouwenhorst}(N-1, \rho, \sigma_\eta)$
 - 3.2.2. $\Pi_N = p \begin{bmatrix} \Pi_{N-1} & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix} + (1-p) \begin{bmatrix} \vec{0} & \Pi_{N-1} \\ 0 & \vec{0}^T \end{bmatrix} + (1-q) \begin{bmatrix} \vec{0}^T & 0 \\ \Pi_{N-1} & \vec{0} \end{bmatrix} + q \begin{bmatrix} 0 & \vec{0}^T \\ \vec{0} & \Pi_{N-1} \end{bmatrix}$
 - 3.2.3. Adjust intermediate rows to sum to 1
4. Bonus: Return stationary distribution $G = \text{Binomial}(N-1, 1/2)$

Gaussian Mixtures

Issue: Our AR(1) processes do not match higher order moments of data

Gaussian Mixtures

Issue: Our AR(1) processes do not match higher order moments of data

- ▶ Papers by Guvenen and Song show earnings data is far from “Normal”
- ▶ Income process presents left skewness and high kurtosis
- ▶ Bloom, Guvenen & Salgado show importance of skewness for firms

Gaussian Mixtures

Issue: Our AR(1) processes do not match higher order moments of data

- ▶ Papers by Guvenen and Song show earnings data is far from “Normal”
- ▶ Income process presents left skewness and high kurtosis
- ▶ Bloom, Guvenen & Salgado show importance of skewness for firms

Solution: Use Gaussian Mixture Models

Gaussian Mixtures

Issue: Our AR(1) processes do not match higher order moments of data

- ▶ Papers by Guvenen and Song show earnings data is far from “Normal”
- ▶ Income process presents left skewness and high kurtosis
- ▶ Bloom, Guvenen & Salgado show importance of skewness for firms

Solution: Use Gaussian Mixture Models

- ▶ Shocks come from a mixture of gaussian sources
- ▶ More sources of variation allow us to capture higher order moments

Matching higher moments

Results from Civalé, Diez-Catalan & Fazilet (2015)

$$z' = \rho z + \eta \quad \text{where } \eta \sim \begin{cases} N(\mu_1, \sigma_1) & \text{with prob. } p \\ N(\mu_2, \sigma_2) & \text{with prob. } 1 - p \end{cases}$$

Matching higher moments

Results from Civalé, Diez-Catalan & Fazilet (2015)

$$z' = \rho z + \eta \quad \text{where } \eta \sim \begin{cases} N(\mu_1, \sigma_1) & \text{with prob. } p \\ N(\mu_2, \sigma_2) & \text{with prob. } 1 - p \end{cases}$$

- ▶ This process is flexible enough to generate skewness and kurtosis in η
 - ▶ These properties are inherited by z

Matching higher moments

Results from Civalé, Diez-Catalan & Fazilet (2015)

$$z' = \rho z + \eta \quad \text{where } \eta \sim \begin{cases} N(\mu_1, \sigma_1) & \text{with prob. } p \\ N(\mu_2, \sigma_2) & \text{with prob. } 1 - p \end{cases}$$

- ▶ This process is flexible enough to generate skewness and kurtosis in η
 - ▶ These properties are inherited by z
- ▶ The process imposes constraints
 - ▶ Parameter ρ is key
 - ▶ Given ρ and moments of z we get moments of $\Delta_k z$
 - ▶ Moments of $\Delta_k z$ are often the target in the data

How to use Gaussian mixtures

Good news: Tauchen's method applies!

How to use Gaussian mixtures

Good news: Tauchen's method applies!

- ▶ Let F_η be the CDF of the Gaussian mixture η
- ▶ Let $\{z_1, \dots, z_N\}$ be a grid for z
- ▶ All formulas apply using the new distribution

How to use Gaussian mixtures

Good news: Tauchen's method applies!

- ▶ Let F_η be the CDF of the Gaussian mixture η
- ▶ Let $\{z_1, \dots, z_N\}$ be a grid for z
- ▶ All formulas apply using the new distribution

Bad news: Results are sensitive to choice of state space grid

How to use Gaussian mixtures

Good news: Tauchen's method applies!

- ▶ Let F_η be the CDF of the Gaussian mixture η
- ▶ Let $\{z_1, \dots, z_N\}$ be a grid for z
- ▶ All formulas apply using the new distribution

Bad news: Results are sensitive to choice of state space grid

- ▶ Civale, Diez-Catalan & Fazilet (2015) propose optimizing over grid
 - ▶ They use the method of moments
- ▶ Process is incompatible with persistence of Skewness and Kurtosis
 - ▶ See Appendix B.2. where they propose a method to address this

Gaussian mixtures - Very popular

Gaussian mixtures used widely, mostly for income fluctuation problems

- ▶ Housing Wealth Effects: The Long View (Guren, McKay, Nakamura, Steinsson, 2020)
- ▶ Time-Varying Idiosyncratic Risk and Aggregate Consumption Dynamics (McKay, 2017)
- ▶ Countercyclical Labor Income Risk and Portfolio Choices over the Life-Cycle (Catherine, 2020)
- ▶ Nonlinear household earnings dynamics, self-insurance, and welfare (DeNardi, Fella, Paz-Pardo, 2020)
- ▶ Monetary policy according to HANK (Kaplan, Moll, Violante, 2018)
 - ▶ Continuous time methods mixing a jump process for kurtosis

Final Words on Methods

Literature's take: Use Rouwenhorst

Literature's take: Use Rouwenhorst

- ▶ Three separate papers get to the same conclusion
 - ▶ Kopecky & Suen (2010)
 - ▶ Galindev & Lkhagvasuren (2010)
 - ▶ Fella, Gallipoli & Pan (2019)

Literature's take: Use Rouwenhorst

- ▶ Three separate papers get to the same conclusion
 - ▶ Kopecky & Suen (2010)
 - ▶ Galindev & Lkhagvasuren (2010)
 - ▶ Fella, Gallipoli & Pan (2019)
- ▶ Rouwenhorst's method outperforms all competition
 - “A 5 point grid Rouwenhurst approximation is generally as accurate as a 25 grid point approximation with other methods”

Literature's take: Use Rouwenhorst

- ▶ Three separate papers get to the same conclusion
 - ▶ Kopecky & Suen (2010)
 - ▶ Galindev & Lkhagvasuren (2010)
 - ▶ Fella, Gallipoli & Pan (2019)
- ▶ Rouwenhorst's method outperforms all competition

“A 5 point grid Rouwenhurst approximation is generally as accurate as a 25 grid point approximation with other methods”

 - ▶ Other methods suffer when $\rho \rightarrow 1$. Rouwenhorst suffers much less.
 - ▶ Rouwenhorst's design to match moments makes it better
 - ▶ Rouwenhorst does not target higher order moments, but still outperforms other methods at low grid sizes.

An example

- ▶ Discretize $z' = \rho z + \eta$, where $\eta \sim N(\mu_\eta, \sigma_\eta^2)$
- ▶ Choose $\rho = 0.95$, $\mu_\eta = 0$ and $\sigma_\eta = 0.2$
- ▶ Simulate 10.000 periods of the Markov Chain

An example

- ▶ Discretize $z' = \rho z + \eta$, where $\eta \sim N(\mu_\eta, \sigma_\eta^2)$
- ▶ Choose $\rho = 0.95$, $\mu_\eta = 0$ and $\sigma_\eta = 0.2$
- ▶ Simulate 10.000 periods of the Markov Chain

	Exact		Tauchen			Rouwenhorst		
	Moments		N=5	N=11	N=21	N=5	N=11	N=21
$E[z]$	$\frac{\mu_\eta}{1-\rho}$	0	0.05	-0.03	0.03	-0.03	0.00	0.01
$\sqrt{V[z]}$	$\frac{\sigma_\eta}{\sqrt{1-\rho^2}}$	0.64	0.87	0.73	0.67	0.65	0.63	0.63
$corr(z, z')$	ρ	0.95	0.99	0.95	0.95	0.95	0.95	0.95

Relevant Extensions

- ▶ Correlated AR(1) process (like a VAR)
 - ▶ Galindev & Lkhagvasuren (2010)
 - ▶ Method reduces to decomposing covariance matrix to get independent shocks
 - ▶ Then apply a modified version of Rouwenhorst

Relevant Extensions

- ▶ Correlated AR(1) process (like a VAR)
 - ▶ Galindev & Lkhagvasuren (2010)
 - ▶ Method reduces to decomposing covariance matrix to get independent shocks
 - ▶ Then apply a modified version of Rouwenhorst
- ▶ Non-stationary process - Designed for life cycle models
 - ▶ Fella, Gallipoli & Pan (2019)
 - ▶ Methods are extensions of Tauchen or Rouwenhorst

Relevant Extensions

- ▶ Correlated AR(1) process (like a VAR)
 - ▶ Galindev & Lkhagvasuren (2010)
 - ▶ Method reduces to decomposing covariance matrix to get independent shocks
 - ▶ Then apply a modified version of Rouwenhorst
- ▶ Non-stationary process - Designed for life cycle models
 - ▶ Fella, Gallipoli & Pan (2019)
 - ▶ Methods are extensions of Tauchen or Rouwenhorst
- ▶ Many other methods...Read the papers!

Application:

GE capitalist/union economy

The economy: Two agents

Capitalists:

- ▶ Infinitively lived derive utility from consumption: $u(c) = c^{1-\gamma}/1-\gamma$
- ▶ Produce output with capital and labor (CRS technology): $y = zk^\alpha \ell^{1-\alpha}$
- ▶ Use their own capital (no borrowing), hire labor in market at wage w

The economy: Two agents

Capitalists:

- ▶ Infinitively lived derive utility from consumption: $u(c) = c^{1-\gamma}/1-\gamma$
- ▶ Produce output with capital and labor (CRS technology): $y = zk^\alpha \ell^{1-\alpha}$
- ▶ Use their own capital (no borrowing), hire labor in market at wage w

Union:

- ▶ Infinitively lived derive utility from consumption: $u(c) = c^{1-\gamma}/1-\gamma$
- ▶ Weird union preferences: Demand constant wage
- ▶ Union internalizes effect on wages and controls labor to adjust price
- ▶ Hand-to-mouth (no borrowing or savings)

Capitalists

$$V(z, k; w) = \max_{\{c, k'\}} u(c) + \beta E \left[V(z', k'; w') \mid z \right]$$

$$\text{s.t. } c + k' \leq \pi(z, k; w)$$

$$\pi(z, k; w) = \max_{\ell} z k^{\alpha} \ell^{1-\alpha} - w\ell + (1 - \delta) k$$

$$\log z' = \rho \log z + \eta; \quad \eta \sim N(0, \sigma_{\eta}^2)$$

Law of motion for wages (more on this later)

Capitalists

$$V(z, k; w) = \max_{\{c, k'\}} u(c) + \beta E \left[V(z', k'; w') \mid z \right]$$

$$\text{s.t. } c + k' \leq \pi(z, k; w)$$

$$\pi(z, k; w) = \max_{\ell} z k^{\alpha} \ell^{1-\alpha} - w \ell + (1 - \delta) k$$

$$\log z' = \rho \log z + \eta; \quad \eta \sim N(0, \sigma_{\eta}^2)$$

Law of motion for wages (more on this later)

- ▶ Note that capitalists have to take w as given
 - ▶ We will talk how to deal with this explicitly later
 - ▶ For now: leap of faith

Capitalists - Profits

$$\pi(z, k; w) = \max_{\ell} z k^{\alpha} \ell^{1-\alpha} - w\ell + (1 - \delta) k$$

Optimal labor choice:

$$\ell^* = \left(\frac{1 - \alpha}{w} z \right)^{\frac{1}{\alpha}} k$$

Optimal profits:

$$\pi(z, k; w) = \left[\underbrace{\alpha \left(\frac{1 - \alpha}{w} \right)^{\frac{1-\alpha}{\alpha}} z^{\frac{1}{\alpha}} + (1 - \delta)}_{\Gamma(z; w)} \right] k$$

Capitalists - Homothetic-Homogeneous DP

We are in luck!

- ▶ Capitalist problem is that of maximizing a homothetic objective subject to homogenous constraint.

Capitalists - Homothetic-Homogeneous DP

We are in luck!

- ▶ Capitalist problem is that of maximizing a homothetic objective subject to homogenous constraint.
- ▶ Particularly constraint is homogenous of degree 1

$$V(z, k; w) = \max_{\{c, k'\}} u(c) + \beta E \left[V(z', k'; w') \mid z \right]$$
$$\text{s.t. } c + k' \leq \Gamma(z; w) k$$

Capitalists - Homothetic-Homogeneous DP

We are in luck!

- ▶ Capitalist problem is that of maximizing a homothetic objective subject to homogenous constraint.
- ▶ Particularly constraint is homogenous of degree 1

$$\begin{aligned} V(z, k; w) &= \max_{\{c, k'\}} u(c) + \beta E \left[V(z', k'; w') \mid z \right] \\ \text{s.t. } c + k' &\leq \Gamma(z; w) k \end{aligned}$$

- ▶ We can guess that the problem is separable in z and k

$$V(z, k; w) = v(z; w) u(k)$$

Capitalists - Guess and verify

$$V(z, k; w) = \max_{k'} \frac{(\Gamma(z; w) k - k')^{1-\gamma}}{1-\gamma} + \underbrace{\beta E \left[v(z'; w') | z \right]}_{\Upsilon(z)} \frac{(k')^{1-\gamma}}{1-\gamma}$$

First order condition:

$$\begin{aligned} (\Gamma(z; w) k - k')^{-\gamma} &= \Upsilon(z) (k')^{-\gamma} \\ \Gamma(z; w) k &= \left(1 + (\Upsilon(z))^{\frac{-1}{\gamma}} \right) k' \end{aligned}$$

Policy function: Save a fraction of income

$$k' = \underbrace{\frac{\Upsilon(z)^{\frac{1}{\gamma}}}{1 + \Upsilon(z)^{\frac{1}{\gamma}}}}_{s(z; w)} \underbrace{\Gamma(z; w) k}_{\pi(z, k; w)}$$

Capitalists - Guess and verify

$$v(z; w) \frac{k^{1-\gamma}}{1-\gamma} = ((1-s(z; w))\Gamma(z; w))^{1-\gamma} \frac{k^{1-\gamma}}{1-\gamma} + \Upsilon(z)(s(z; w)\Gamma(z; w))^{1-\gamma} \frac{k^{1-\gamma}}{1-\gamma}$$

$$v(z; w) = ((1-s(z; w))\Gamma(z; w))^{1-\gamma} + \Upsilon(z)(s(z; w)\Gamma(z; w))^{1-\gamma}$$

$$v(z; w) = \left[(1-s(z; w))^{1-\gamma} + \Upsilon(z)s(z; w)^{1-\gamma} \right] \Gamma(z; w)^{1-\gamma}$$

$$v(z; w) = \left[1 + \Upsilon(z)^{\frac{1}{\gamma}} \right] \left(\frac{\Gamma(z; w)}{1 + \Upsilon(z)^{\frac{1}{\gamma}}} \right)^{1-\gamma}$$

$$v(z; w) = \left[1 + \Upsilon(z)^{\frac{1}{\gamma}} \right]^{\gamma} (\Gamma(z; w))^{1-\gamma}$$

$$v(z; w) = \left[1 + \left(\beta E \left[v(z'; w') | z \right] \right)^{\frac{1}{\gamma}} \right]^{\gamma} (\Gamma(z; w))^{1-\gamma}$$

Workers' Union

$$W(z, K) = u(\bar{w}\ell^s(z, K)) + \beta E \left[W(z', K'(z, K)) | z \right]$$

Not much to do here... sorry

- ▶ Union dictates labor supply of workers according to some rule ℓ^s
 - ▶ Rule is set to maintain a constant wage of \bar{w}
- ▶ Agents need to know the equilibrium law of motion for aggregate capital K
 - ▶ We know we will get it from the capitalists

General equilibrium

Market clearing:

- ▶ Union sets $\ell^s(z, k)$ such that:

$$w^* = (1 - \alpha) z \left(\frac{k}{\ell^s(z, k)} \right)^\alpha = \bar{w}$$

- ▶ Labor depends on z and k , but not wages

General equilibrium

Market clearing:

- ▶ Union sets $\ell^s(z, k)$ such that:

$$w^* = (1 - \alpha) z \left(\frac{k}{\ell^s(z, k)} \right)^\alpha = \bar{w}$$

- ▶ Labor depends on z and k , but not wages
- ▶ Capitalists do not take into account their effect on aggregate prices

General equilibrium

Market clearing:

- ▶ Union sets $\ell^s(z, k)$ such that:

$$w^* = (1 - \alpha) z \left(\frac{k}{\ell^s(z, k)} \right)^\alpha = \bar{w}$$

- ▶ Labor depends on z and k , but not wages
- ▶ Capitalists do not take into account their effect on aggregate prices
- ▶ However, **we know** how capital and productivity affect prices

Dynamic Programming: Final

$$v(z) = \left[1 + \left(\beta E \left[v(z') | z \right] \right)^{\frac{1}{\gamma}} \right]^{\gamma} (\Gamma(z))^{1-\gamma}$$

where

$$\Gamma(z) = \alpha \left(\frac{1-\alpha}{\bar{w}} \right)^{\frac{1-\alpha}{\alpha}} z^{\frac{1}{\alpha}} + (1-\delta)$$

- Looks the same... but we dropped w as it is constant in equilibrium

Dynamic Programming: Final

$$v(z) = \left[1 + \left(\beta E \left[v(z') | z \right] \right)^{\frac{1}{\gamma}} \right]^{\gamma} (\Gamma(z))^{1-\gamma}$$

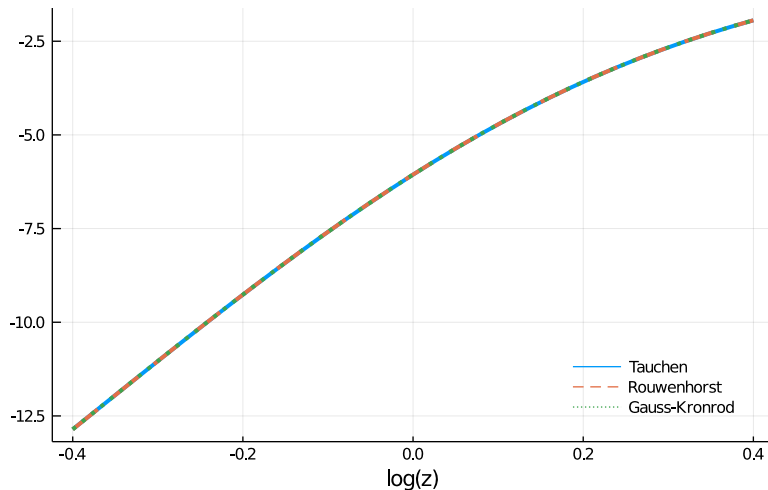
where

$$\Gamma(z) = \alpha \left(\frac{1-\alpha}{\bar{w}} \right)^{\frac{1-\alpha}{\alpha}} z^{\frac{1}{\alpha}} + (1-\delta)$$

- ▶ Looks the same... but we dropped w as it is constant in equilibrium
- ▶ To solve the dynamic programming problem we need to integrate
 - ▶ No max involved, only integrals

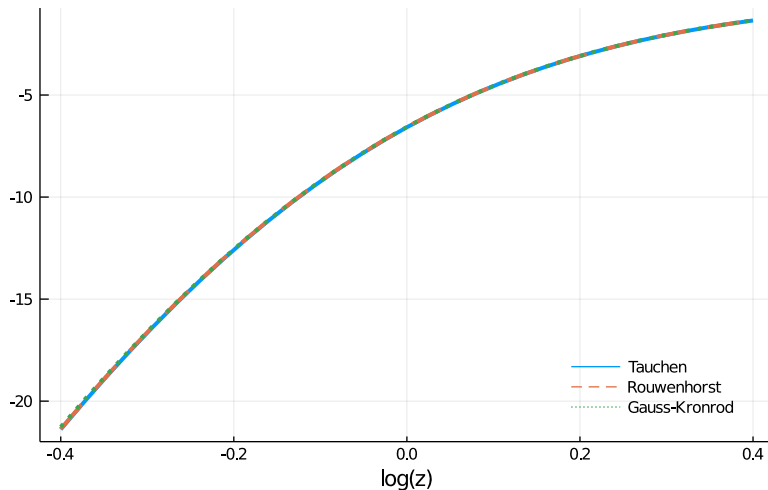
Dynamic Programming: Result

Value Function: $v(z)/(1-\gamma)$ - $\rho=0.5$, $N=31$



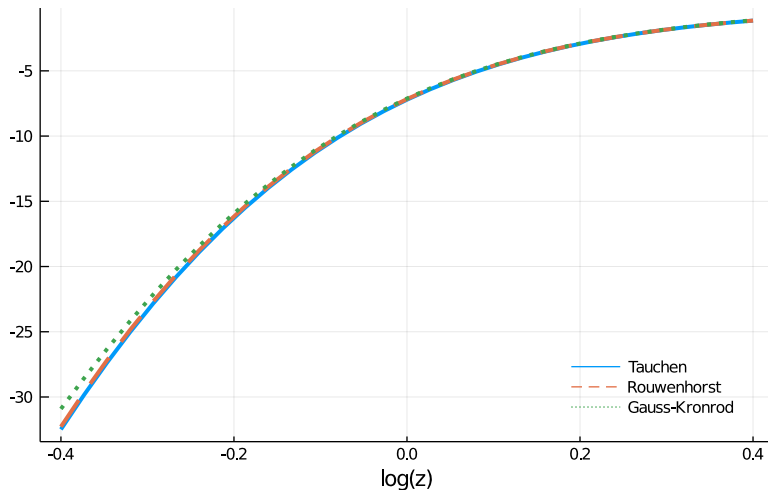
Dynamic Programming: Result

Value Function: $v(z)/(1-\gamma)$ - $\rho=0.8$, $N=31$



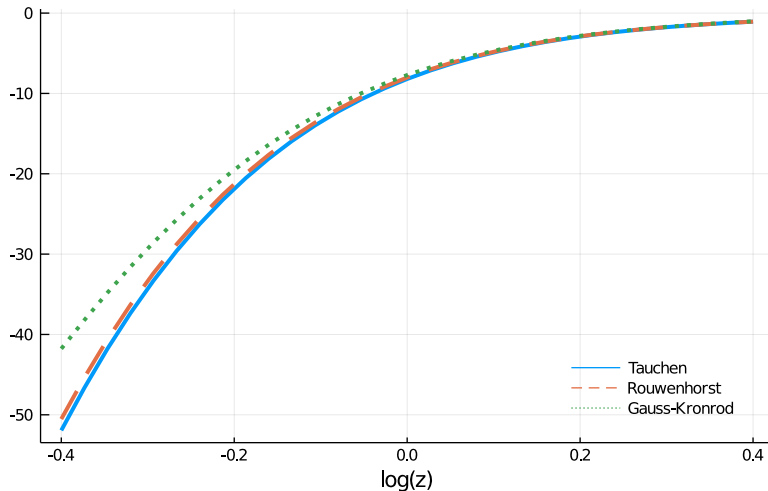
Dynamic Programming: Result

Value Function: $v(z)/(1-\gamma)$ - $\rho=0.9$, $N=31$



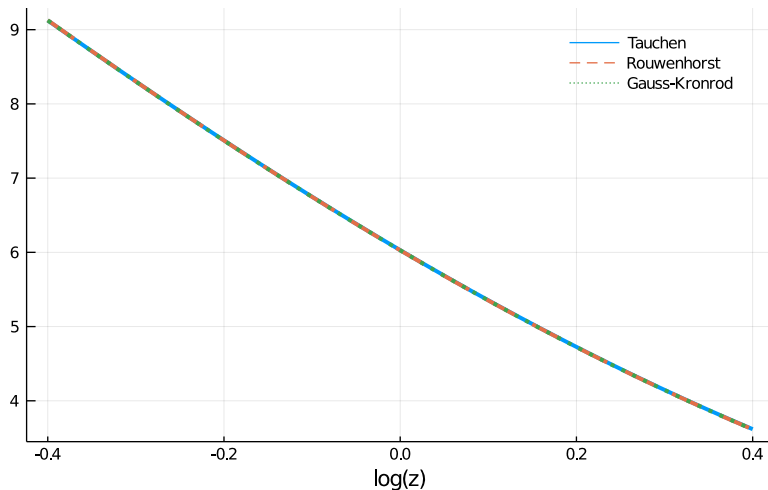
Dynamic Programming: Result

Value Function: $v(z)/(1-\gamma) - \rho=0.95, N=31$



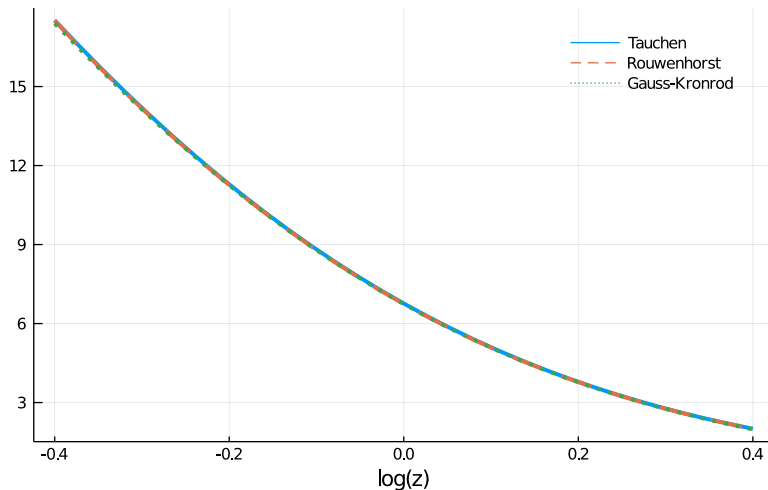
Differences in expectations

Expectation: $\beta E[v(z')|z] - \rho=0.5, N=31$



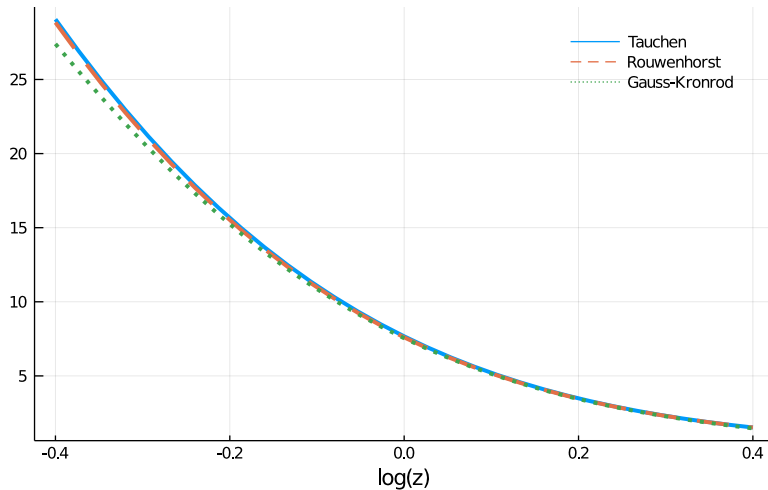
Differences in expectations

Expectation: $\beta E[v(z')|z] - \rho=0.8, N=31$



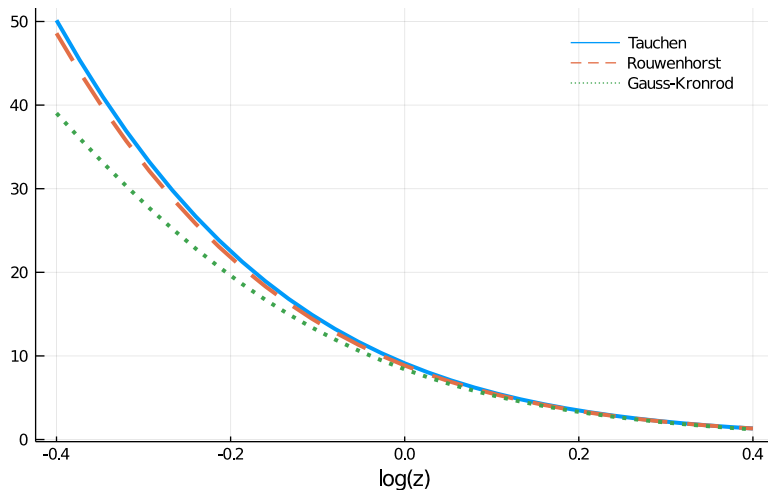
Differences in expectations

Expectation: $\beta E[v(z')|z] - \rho=0.9, N=31$



Differences in expectations

Expectation: $\beta E[u(z')|z] - \rho=0.95, N=31$



What to make of these graphs?

- ▶ We do not have a closed form solution... so hard to claim success

What to make of these graphs?

- ▶ We do not have a closed form solution... so hard to claim success
- ▶ Tauchen and Rouwenhorst give similar results:
 - ▶ But we were using 31 nodes... kind of cheating

What to make of these graphs?

- ▶ We do not have a closed form solution... so hard to claim success
- ▶ Tauchen and Rouwenhorst give similar results:
 - ▶ But we were using 31 nodes... kind of cheating
- ▶ Rouwenhorst vs Gauss-Kronrod
 - ▶ Hard to tell because GK uses a lot of extrapolation

What to make of these graphs?

- ▶ We do not have a closed form solution... so hard to claim success
- ▶ Tauchen and Rouwenhorst give similar results:
 - ▶ But we were using 31 nodes... kind of cheating
- ▶ Rouwenhorst vs Gauss-Kronrod
 - ▶ Hard to tell because GK uses a lot of extrapolation
- ▶ Rouwenhorst seems like the best option
 - ▶ Computationally feasible
 - ▶ Reliable for high persistence values