Tools: Numerical Integration and Stochastic Processes

Emile Alexandre Marin

emarin@ucdavis.edu

October 3, 2022

Numerical Integration / Quadrature

Think back to the trapezoidal rule in high school!

- integrate area under a function by summing up trapeziums, which also exploit the derivatives of the function

For $f : D \in \mathbb{R}^n \to \mathbb{R}$,

* $I \approx \int_D f(x)dx$

Numerical Integration / Quadrature

Think back to the trapezoidal rule in high school!

- integrate area under a function by summing up trapeziums, which also exploit the derivatives of the function

For $f : D \in \mathbb{R}^n \to \mathbb{R}$,

* $I \approx \int_D f(x)dx$

For $f : [a, b] \to \mathbb{R}$:

* $\int_a^b f(x)dx \approx \sum_{i=1}^n \omega_i f(x_i)$,

where ω_i are weights and $x_i \in [a, b]\ \forall i$.

Numerical Integration / Quadrature

Think back to the trapezoidal rule in high school!

- integrate area under a function by summing up trapeziums, which also exploit the derivatives of the function

For $f : D \in \mathbb{R}^n \to \mathbb{R}$,

* $I \approx \int_D f(x)dx$

For $f : [a, b] \to \mathbb{R}$:

* $\int_a^b f(x)dx \approx \sum_{i=1}^n \omega_i f(x_i)$,

where ω_i are weights and $x_i \in [a, b] \; \forall i$.

The challenge is to choose nodes $\{x_1, ..., x_n\}$ and weights $\{\omega_1, ..., \omega_n\}$ to make approximation as accurate as possible.

Numerical Integration / Quadrature

- **Newton-Cotes quadrature:** discretize domain into equidistant points and assign appropriate weights

Numerical Integration / Quadrature

- **Newton-Cotes quadrature:** discretize domain into equidistant points and assign appropriate weights
- **Gaussian quadrature:** discretize domain with efficiently chosen points which work with orthogonal polynomials and *their* weights, e.g. Chebushev or Hermite (normal)

Numerical Integration / Quadrature

- **Newton-Cotes quadrature:** discretize domain into equidistant points and assign appropriate weights
- **Gaussian quadrature:** discretize domain with efficiently chosen points which work with orthogonal polynomials and *their* weights, e.g. Chebushev or Hermite (normal)
- **Monte Carlo Integration:** Law of large numbers + random choices of points on grid

For details see [Judd, 1998, ch. 7]

Numerical Integration / Quadrature

- **Newton-Cotes quadrature:** discretize domain into equidistant points and assign appropriate weights
- **Gaussian quadrature:** discretize domain with efficiently chosen points which work with orthogonal polynomials and *their* weights, e.g. Chebushev or Hermite (normal)
- **Monte Carlo Integration:** Law of large numbers + random choices of points on grid

For details see [Judd, 1998, ch. 7]

Newton-Cotes quadrature

- Simpson's rule: fit piecewise quadratic approximation to interval

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right] - \frac{(b-a)^5}{2880} f^{(4)}(\xi)$$

for some $\xi \in [a, b]$.

- Composite Simpson's rule: break interval into $\frac{b-a}{n}$ and apply piece-wise qudratic spline, then sum up.

Gaussian Quadrature

This is usually our preferred method. It creates a 'good' approximation $\equiv$ exact integration for finite set of functions, e.g. polynomials (strong result!)

Gaussian Quadrature

This is usually our preferred method. It creates a 'good' approximation $\equiv$ exact integration for finite set of functions, e.g. polynomials (strong result!)

General approximation for weighted integral:

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

Why Gaussian quadrature?

- *for a given family of orthogonal polynomials* $\{\phi_j\}_{j=1}^{\infty}$ *with roots* x_i *and and weight function w(x), there exist weights* ω_i *so that the numerical integral approximation is exact for polynomials of order* $2n-1$.
- we can use these weights for approximation of other functions $f \in C^{2n}[a,b]$:

$$\int_a^b f(x)w(x)dx = \sum_{i=1}^{n} \omega_i f(x_i) + \frac{f^{(2n)}(\xi)}{\theta_n^2 (2n)!}$$

where θ_j is $j-th$ coefficient in polynomial $\phi_j(x)$ and

$$\omega_i = -\frac{\theta_{n+1}/\theta_n}{\phi_n'(x_i)\phi_{n+1}(x_i)} > 0$$

Why Gaussian quadrature?

- *for a given family of orthogonal polynomials $\{\phi_j\}_{j=1}^{\infty}$ with roots x_i and and weight function w(x), there exist weights ω_i so that the numerical integral approximation is exact for polynomials of order $2n-1$.*

- we can use these weights for approximation of other functions $f \in C^{2n}[a,b]$:

$$\int_a^b f(x)w(x)dx = \sum_{i=1}^{n} \omega_i f(x_i) + \frac{f^{(2n)}(\xi)}{\theta_n^2 (2n)!}$$

  where θ_j is $j-th$ coefficient in polynomial $\phi_j(x)$ and

$$\omega_i = -\frac{\theta_{n+1}/\theta_n}{\phi_n'(x_i)\phi_{n+1}(x_i)} > 0$$

We are going to use specific quadratures, e.g. Chebyshev

Gauss-Chebyshev

Approximate intervals of the following form:

$$\int_{-1}^{1} f(z)(1-z^2)^{-1/2} dz,$$

$$w(z) = (1-z^2)^{-1/2}$$

which, if you remember, is the weight that makes Chebyshev polynomials orthogonal.

Gauss-Chebyshev

Approximate intervals of the following form:

$$\int_{-1}^{1} f(z)(1-z^2)^{-1/2}dz,$$

$$w(z) = (1-z^2)^{-1/2}$$

which, if you remember, is the weight that makes Chebyshev polynomials orthogonal.

Roots of the Chebyshev (order n) used as grid points:

$$z_i = cos\left(\frac{2i-1}{2n}\pi\right), \quad i = 1, ...n$$

and quadrature weights $\omega_i = \frac{\pi}{n}$.

Gauss-Chebyshev

Approximate intervals of the following form:

$$\int_{-1}^{1} f(z)(1-z^2)^{-1/2} dz,$$
$$w(z) = (1-z^2)^{-1/2}$$

which, if you remember, is the weight that makes Chebyshev polynomials orthogonal.
Roots of the Chebyshev (order n) used as grid points:

$$z_i = cos\left(\frac{2i-1}{2n}\pi\right), \quad i = 1, ...n$$

and quadrature weights $\omega_i = \dfrac{\pi}{n}$.
For $\xi \in [-1,1]$:

$$\int_{-1}^{1} f(z)(1-z^2)^{-(1/2)} dz = \frac{\pi}{n}\sum_{i=1}^{n} f(z_i) + \frac{\pi}{2^{2n-1}}\frac{f^{(2n)}(\xi)}{(2n)!}$$

Gauss-Chebyshev quadrature (cookbook)

We want to approximate $I = \int_a^b f(x)dx$.

Gauss-Chebyshev quadrature (cookbook)

We want to approximate $I = \int_a^b f(x)dx$.

1. Transform grid point from $[-1, 1]$ to $\mathbb{R}$:

$$x = \frac{(z+1)(b-a)}{2} + a$$

so:

$$I = \frac{b-1}{2} \int_a^b f\left(\frac{(z+1)(b-a)}{2} + a\right) dz$$

Gauss-Chebyshev quadrature (cookbook)

We want to approximate $I = \int_a^b f(x)dx$.

1. Transform grid point from $[-1, 1]$ to $\mathbb{R}$:

$$x = \frac{(z+1)(b-a)}{2} + a$$

so:

$$I = \frac{b-1}{2} \int_a^b f\left(\frac{(z+1)(b-a)}{2} + a\right) dz$$

2. Next multiple and divide by $(1-z^2)^{(-1/2)}$:

$$\int_a^b f\left(\frac{(z+1)(b-a)}{2} + a\right) \frac{(1-z^2)^{-(1/2)}}{(1-z^2)^{-(1/2)}} dz$$

Gauss-Chebyshev quadrature (cookbook)

We want to approximate $I = \int_a^b f(x)dx$.

1. Transform grid point from $[-1, 1]$ to $\mathbb{R}$:

$$x = \frac{(z+1)(b-a)}{2} + a$$

so:

$$I = \frac{b-1}{2} \int_a^b f\left(\frac{(z+1)(b-a)}{2} + a\right) dz$$

2. Next multiple and divide by $(1-z^2)^{(-1/2)}$:

$$\int_a^b f\left(\frac{(z+1)(b-a)}{2} + a\right) \frac{(1-z^2)^{-(1/2)}}{(1-z^2)^{-(1/2)}} dz$$

3. The approximation formula (on z_i Chebyshev nodes) is :

$$\int_a^b f(x)dx \approx \frac{\pi(b-a)}{2n} \sum_{i=1}^{n} f\left(\frac{(z_i+1)(b-a)}{2} + a\right) (1-z_i^2)^{1/2}$$

Gauss-Hermite quadrature

A Hermite polynomial of order n, denoted $H_n : \mathbb{R} \to \mathbb{R}$ is given by:

$$H_n(z) = (-1)^n exp(z^2) \frac{d^n exp(-z^2)}{dz^n}$$

where $d^n(\cdot)/dz^n$ is the n^{th} derivative of f.

Gauss-Hermite quadrature

A Hermite polynomial of order n, denoted $H_n : \mathbb{R} \to \mathbb{R}$ is given by:

$$H_n(z) = (-1)^n exp(z^2) \frac{d^n exp(-z^2)}{dz^n}$$

where $d^n(\cdot)/dz^n$ is the n^{th} derivative of f.

The polynomial can also be conveniently defined recursively:

$$H_0(z) = 1, \quad H_1(z) = 2z,$$
$$H_{n+1}(z) = 2zH_n(z) - 2nH_{n-1}(z)$$

The weight function that makes Hermite polynomials orthogonal:

$$w(x) = exp(-z^2)$$

The roots z_i and weights ω_i are tabulated for polynomials up to $n = 10$ (see e.g. Greenwood and Miller, 1948)

Gauss-Hermite quadrature contd.

$$\int_{-\infty}^{\infty} f(z) exp(-z^2) dz = \sum_{i=1}^{n} \omega_i f(z_i) + \frac{n!\sqrt{\pi}}{2^n} \frac{f^{2n}(\xi)}{2n!}$$

with $\xi \in (-\infty, \infty)$.

Gauss-Hermite quadrature contd.

$$\int_{-\infty}^{\infty} f(z)exp(-z^2)dz = \sum_{i=1}^{n} \omega_i f(z_i) + \frac{n!\sqrt{\pi}}{2^n}\frac{f^{2n}(\xi)}{2n!}$$

with $\xi \in (-\infty, \infty)$.

Hermite quadratures are useful when we want to calculate moments of functions of normal variables (think normal pdf)

Gauss-Hermite quadrature contd.

$$\int_{-\infty}^{\infty} f(z)exp(-z^2)dz = \sum_{i=1}^{n} \omega_i f(z_i) + \frac{n!\sqrt{\pi}}{2^n}\frac{f^{2n}(\xi)}{2n!}$$

with $\xi \in (-\infty, \infty)$.

Hermite quadratures are useful when we want to calculate moments of functions of normal variables (think normal pdf)

To calculate $E(h(X)), X \sim N(\mu, \sigma^2)$, denoting f the Normal pdf and ϕ the standard normal:

$$I = \int_{-\infty}^{\infty} h(x)f(x)dx = \frac{1}{\sigma}\int_{-\infty}^{\infty} h(x)\phi(\frac{x-\mu}{\sigma})dx =$$

$$\frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} h(x)exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]dx$$

Gauss-Hermite quadrature (cookbook)

1. Transform variables from x to z:

$$-z^2 = -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \implies x = (\sigma\sqrt{2})z + \mu,$$

$$dx = (\sigma\sqrt{2})dz$$

2. $I = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} h((\sigma\sqrt{2})z + \mu) exp(-z^2) dz$

3. The Hermite quadrature approximation is:

$$I = E(h(X)) \approx \int_{-\infty}^{\infty} h(x) f(x) dx = \frac{1}{\sqrt{\pi}} \sum_{i=1}^{n} \omega_i h((\sigma\sqrt{2})z_i + \mu)$$

where z_i are G-H notes and weights over $(-\inf, \inf)$ and $f(x)$ is the $N(\mu, \sigma^2)$ pdf.

Gauss-Hermite example

hermquad.m is a script which generates appropriate abscissae and weights. Typing *[X,W]=hermquad(5)* yields:

$$X = \begin{bmatrix} 2.0202 \\ 0.9586 \\ 0 \\ -0.9586 \\ -2.0202 \end{bmatrix}, \quad W = \begin{bmatrix} 0.0200 \\ 0.3936 \\ 0.9453 \\ 0.3936 \\ 0.0200 \end{bmatrix}$$

These need to be normalized: $\hat{X} = X\sqrt{2}\sigma$, $\hat{W} = \pi^{-1/2}W$.

Gauss-Hermite example contd.

Lets use $\sigma = 0.01$ for the normalization.

Gauss-Hermite example contd.

Lets use $\sigma = 0.01$ for the normalization.

- $h(x) = x^2$
- use $n = 5$
- $\sum_{i=1}^{5} h(x_i) w_i = 0.0001$

Gauss-Hermite example contd.

Lets use $\sigma = 0.01$ for the normalization.

- $h(x) = x^2$
- use $n = 5$
- $\sum_{i=1}^{5} h(x_i) w_i = 0.0001$

Note that $\sqrt{0.0001} = 0.01$ which is the σ we provided

Gauss-Hermite example contd.

Lets use $\sigma = 0.01$ for the normalization.

- $h(x) = x^2$
- use $n = 5$
- $\sum_{i=1}^{5} h(x_i) w_i = 0.0001$

Note that $\sqrt{0.0001} = 0.01$ which is the σ we provided $\implies N = 5$ is enough with N shocks and GH!

Monte Carlo Integration

Different approach to quadratures based on law of large numbers (LLN):

Monte Carlo Integration

Different approach to quadratures based on law of large numbers (LLN):

- Nodes x_i randomly drawn, quadrature weights ω_i *can* be equal
- Expected value of collection of $r.v.$ X with density f:

$$\mathbb{E}[x] = \int_S x f(x) fx$$

- If X_i are iid random in $[0, 1]$:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} X_i = \int_0^1 x f(x) dx \quad a.s.$$

with $var(\frac{1}{n}\sum_{i=1}^{n} X_i) = \sigma_X^2 / n$

Monte Carlo Integration example

Example: If $X \sim U(0,1)$:

$$I_h = \int_0^1 h(x)dx = \mathbb{E}[h(x)]$$

Monte Carlo Integration example

Example: If $X \sim U(0,1)$:

$$I_h = \int_0^1 h(x)dx = \mathbb{E}[h(x)]$$

Then:

$$\tilde{I}_h = \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$

Monte Carlo Integration example

Example: If $X \sim U(0,1)$:

$$I_h = \int_0^1 h(x)dx = \mathbb{E}[h(x)]$$

Then:

$$\tilde{I}_h = \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$

so, like general integration, but weights are $1/n$ and x_i randomly drawn.

see 8.6 in Judd for portfolio optimization with MC integration example.

Monte Carlo Integration contd.

However, different from before, $\tilde{I}_h$ is itself a random variable with a variance:

$$\sigma^2_{\tilde{I}_h} = \frac{1}{n}\sigma^2_h = \frac{1}{n}\int_0^1 (h(x))^2 dx - \left(\int_0^1 h(x)dx\right)^2$$

which is unknown.

Monte Carlo Integration contd.

However, different from before, $\tilde{I}_h$ is itself a random variable with a variance:

$$\sigma^2_{\tilde{I}_h} = \frac{1}{n}\sigma^2_h = \frac{1}{n}\int_0^1 (h(x))^2 dx - \left(\int_0^1 h(x)dx\right)^2$$

which is unknown.

This variance can be estimated as follows:

$$\sigma^2_{\tilde{I}_h} = \frac{1}{n-1}\sum_{i=1}^{n}(h(x_i) - \tilde{I}_h)^2$$

but the variance tends to be very high, especially when n not very large.

- Judd 8.2 explains several ways to reduce variance: stratified sampling, antithetic variates, control variates, importance sampling.
- MC integration often used for multivariate cases (curse of dimensionality with standard quadratures) : see Judd 9

Adaptive Quadrature

As $n \to \inf$, approximation will tend to true integral

Adaptive Quadrature

As $n \to \inf$, approximation will tend to true integral

Use following tactic:

1. decide on type of quadrature/ MCI
2. take a small n and calculate $\tilde{I}^{(n)}$
3. calculate $\tilde{I}^{(n)}$ for larger n
4. stop when integrals don't change much (keeping tolerance constant)

Approximating Stochastic Processes

Consider a normal iid process $\eta_t \sim N(\mu, \sigma^2)$

Approximating Stochastic Processes

Consider a normal iid process $\eta_t \sim N(\mu, \sigma^2)$

Objective: Discretize the range (or state space) of η_t into points $\zeta_1, ..., \zeta_n$ and give each point an approximate probability of occurring p_i

Strategy:

1. Discretize
   - equal distance on $\mathbb{R}$
   - equal probability within each interval
2. using the fact that η is normal:
   - given ζ_i calculate p_i
   - given p_i calculate ζ_i

Equispaced grid (1/2)

- Let $\zeta_1, ..., \zeta_n$ be the discretized space for η_t
- $\zeta_1 = \mu - r\sigma$, $\zeta_N = \mu + r\sigma$
- $\zeta_2, ...\zeta_{N-1}$ are defined by an equispaced grid, i.e.:

$$d = \frac{\zeta_N - \zeta_1}{N-1} = \frac{2r\sigma}{N-1},$$
$$\zeta_i = \zeta_1 + (i-1)d$$

- Define the borders of the interval
  $m_i = (\zeta_i + 1 + \zeta_i)/2 = \zeta_i + d/2$

Equispaced grid (2/2)

- $$\zeta_i = \begin{cases} (-\infty, m_1], & i = 1 \\ (m_{i-1}, m_i], & 1 < i < N \\ (m_{N-1}, \infty), & i = N \end{cases}$$

Equipspaced grid (2/2)

- $$\zeta_i = \begin{cases} (-\infty, m_1], & i = 1 \\ (m_{i-1}, m_i], & 1 < i < N \\ (m_{N-1}, \infty), & i = N \end{cases}$$

- $$p_i = Pr(\zeta_i) \approx \begin{cases} Pr(\eta_t \leq m_1), & i = 1 \\ Pr(m_{i-1} < \eta_t \leq m_i), & 1 < i < N, \\ 1 - Pr(\eta_t \leq m_N), & 1 < i = N \end{cases}$$

$$p_i = \begin{cases} \Phi\left(\frac{m_1 - \mu}{\sigma}\right), & i = 1 \\ \Phi\left(\frac{m_i - \mu}{\sigma}\right) - \Phi\left(\frac{m_{i-1} - \mu}{\sigma}\right), & 1 < i < N, \\ 1 - \Phi\left(\frac{m_N - \mu}{\sigma}\right), & 1 < i = N \end{cases}$$

Equiprobable grid (1/2)

Here, we define border points m_i such that:

$$p_1 = Pr(\eta_t \leq m_1) = \Phi\left(\frac{m_1 - \mu}{\sigma}\right) = \frac{1}{N},$$

$$\cdots,$$

$$p_i = Pr(m_{i-1} < \eta_t \leq m_i) = \Phi\left(\frac{m_{i-1} - \mu}{\sigma} - \frac{1}{N}\right) = \frac{1}{N}$$

Equiprobable grid (1/2)

Here, we define border points m_i such that:

$$p_1 = Pr(\eta_t \leq m_1) = \Phi\left(\frac{m_1 - \mu}{\sigma}\right) = \frac{1}{N},$$
$$\dots,$$
$$p_i = Pr(m_{i-1} < \eta_t \leq m_i) = \Phi\left(\frac{m_{i-1} - \mu}{\sigma} - \frac{1}{N}\right) = \frac{1}{N}$$

We recover m_i from:

$$\Phi\left(\frac{m_i - \mu}{\sigma}\right) = \frac{i}{N} \implies m_i = \Phi^{-1}\left(\frac{i}{N}\right)\sigma + \mu$$

Equiprobable grid (1/2)

Here, we define border points m_i such that:

$$p_1 = Pr(\eta_t \leq m_1) = \Phi\left(\frac{m_1 - \mu}{\sigma}\right) = \frac{1}{N},$$
$$\cdots,$$
$$p_i = Pr(m_{i-1} < \eta_t \leq m_i) = \Phi\left(\frac{m_{i-1} - \mu}{\sigma} - \frac{1}{N}\right) = \frac{1}{N}$$

We recover m_i from:

$$\Phi\left(\frac{m_i - \mu}{\sigma}\right) = \frac{i}{N} \implies m_i = \Phi^{-1}\left(\frac{i}{N}\right)\sigma + \mu$$

Therefore:

$$\zeta_i = \begin{cases} (-\infty, m_1], & i = 1 \\ (m_{i-1}, m_i], & 1 < i < N \\ (m_{N-1}, \infty), & i = N \end{cases}$$

Equiprobable grid (2/2)

In particular ζ_i should be placed at the expected value of η_t, given that η_t is between m_{i-1}, m_i.

* If a r.v. $X \sim N$, conditional on $a < X \leq B$:

$$\mathbb{E}[X|a < X \leq b] = \mu - \sigma \frac{\phi\left(\frac{b-\mu}{\sigma}\right) - \phi\left(\frac{a-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}$$

So:

$$\zeta_i = \begin{cases} \mu - \sigma N \phi\left(\frac{m_1-\mu}{\sigma}\right), & i = 1 \\ \mu - \sigma N \left[\phi\left(\frac{m_i-\mu}{\sigma}\right) - \phi\left(\frac{m_{i-1}-\mu}{\sigma}\right)\right], & 1 < i < N \\ \mu + \sigma N \phi\left(\frac{m_{N-1}-\mu}{\sigma}\right), & i = N \end{cases}$$

Tauchen Method

Approximating AR(1) with a Markov Chain

- Consider AR(1) process:

$$\zeta_t = \mu(1-\rho) + \rho z_{t-1} + \eta_t$$

  where $\eta_t \sim N(0, \sigma^2)$, where the range is $\mathbb{R}$ and $\sigma = \frac{\sigma}{\sqrt{1-\rho^2}}$

- Discretize range (=state space) into $\zeta_i, i = 1, ..., N$ and assign probabilities a point ζ_j of π_{ij} **given previous state was** ζ_i
- $\Pi = [\pi_{ij}]$ is the transition matrix

Here, we proceed with equispaced grid, but you can use equiprobable as an alternative.

Tauchen Method cookbook (1/3)

- Let $\zeta_1 < \zeta_2 < ...\zeta_N$ be the discretized space for z_t
- Define upper and lower bounds:

$$\zeta_1 = \mu - r\sqrt{\left(\frac{\sigma^2}{1-\rho^2}\right)}, \quad \zeta_N = \mu + r\sqrt{\left(\frac{\sigma^2}{1-\rho^2}\right)}$$

- and $\zeta_2, ..., \zeta_{N-1}$ using an equispaced grid:

$$d = \frac{\zeta_N - \zeta_1}{N-1} = \frac{2r\sigma_z}{N-1},$$
$$\zeta_i = \zeta_1 + (i-1)d$$

$r = 3$, $N = 9$ *tend to work well for AR(1)*

Tauchen Method cookbook (2/3)

- Create borders $[\zeta_i, \zeta_{i+1}]$

$$m_i = \frac{\zeta_{i+1} + \zeta_i}{2} = \zeta_1 + (2i-1)\frac{d}{2},$$

$$\zeta_i = \begin{cases} (-\infty, m_1], & i = 1 \\ (m_{i-1}, m_i], & 1 < i < N \\ (m_{N-1}, \infty), & i = N \end{cases}$$

- For each row i of the transition matrix, for $j = 2, ..., N-1$:

$$\begin{aligned} \pi_{ij} = Pr(z_{t+1} = \zeta_j | z_t = \zeta_i) &= Pr(\mu(1-\rho) + \rho\zeta_i + \eta_{t+1} = \zeta_j) \\ &\approx Pr(m_{j-1} \leq \mu(1-\rho) + \rho\zeta_i + \eta_{t+1} \leq m_j) \\ = \Phi\left(\frac{m_j - \rho\zeta_i - \mu(1-\rho)}{\sigma}\right) &- \Phi\left(\frac{m_{j-1} - \rho\zeta_i - \mu(1-\rho)}{\sigma}\right) \end{aligned}$$

Tauchen Method cookbook (3/3)

- for $j = 1$

$$\begin{aligned}\pi_{i1} = Pr(z_{t+1} = \zeta_1 | \zeta_t = \zeta_i) = \\ Pr(\mu(1-\rho) + \rho z_t + \eta_{t+1} = \zeta_1 | \zeta_t = \zeta_i) = \\ Pr(\mu(1-\rho) + \rho z_t + \eta_{t+1} = \zeta_1) \\ \approx Pr(\mu(1-\rho) + \rho z_t + \eta_{t+1} < m_1) \\ = \Phi\left(\frac{m_1 - \rho\zeta_i - \mu(1-\rho)}{\sigma}\right)\end{aligned}$$

- for j=N

$$\begin{aligned}\pi_{iN} = Pr(z_{t+1} = \zeta_N | \zeta_t = \zeta_i) \\ 1 - \Phi\left(\frac{m_{N-1} - \rho\zeta_i - \mu(1-\rho)}{\sigma}\right)\end{aligned}$$

Simulate a Markov Chain

- Generate a series y_t $t = 1, ..., T$ from Markov Chain
- Consider 2 states with a given transition matrix
- Determine length of realizations T and use transition matrix Π (2x2) estimated via Tauchen

Algorithm: *Assign an initial value for the realization, say $S(1) = 1$.*
Create vectors of cumulative distribution of each state
$Q_j = [p_{j1}, p_{j1} + p_{j2}]$

Simulate a Markov Chain

- Generate a series y_t $t = 1, ..., T$ from Markov Chain
- Consider 2 states with a given transition matrix
- Determine length of realizations T and use transition matrix Π (2x2) estimated via Tauchen

Algorithm: *Assign an initial value for the realization, say $S(1) = 1$.*
Create vectors of cumulative distribution of each state
$Q_j = [p_{j1}, p_{j1} + p_{j2}]$
GENERATE THE CHAIN:
For $i = 2 : T$
generate a number x from URNG
select the cum. dis. vector that corresponds to$S(i-1)$
if x is smaller than the first element, then $S(i) = 1$
if x is smaller than the largest element only, then $S(i) = 2$

Key references:

- *Tauchen, 1986, Finite state markov-chain approximations to univariate and vector autoregressions, Economics LettersJudd, 1998. Chapter 7.*
- *Adda and Cooper, 2003, Dynamic Economics, section 3.A.2 (for Tauchen algorithms)*
- *Greenwoord and Miller, 1948. Zeros of the Hermite polynomials and weights for Gauss' mechanical quadrature formula, Bull. Amer. Math. Soc. 54, 765-769. Available at: http://www.ams.org/journals/bull/1948-54-08/S0002-9904-1948-09075-9/S0002-9904-1948-09075-9.pdf*