Lecture 3: Integration

Fatih Guvenen
University of Minnesota

November 2023

A Prototype Problem

At grid point (k_i, z_j), we solve:

$$V(k_i, z_j) = \max_{c,k'} [u(c) + \beta \mathbb{E}(V(k', z')|z_j)]$$

$$c + k' = (1 + r)k_i + z_j$$

$$z' = \rho z_j + \eta$$

A Prototype Problem

At grid point (k_i, z_j), we solve:

$$V(k_i, z_j) = \max_{c,k'} \left[u(c) + \beta \int V(k', z') f(z'|z_j) dz' \right]$$

$$c + k' = (1 + r)k_i + z_j$$

$$z' = \rho z_j + \eta$$

A Prototype Problem

At grid point (k_i, z_j), we solve:

$$V(k_i, z_j) = \max_{c,k'} \left[u(c) + \beta \int V(k', z') f(z'|z_j) dz' \right]$$

$$c + k' = (1 + r)k_i + z_j$$

$$z' = \rho z_j + \eta$$

- How to evaluate the conditional expectation for a given z_j?

Integration in DP problems

Integration in DP problems: Two Routes

1. Discretize the State Space:

Integration in DP problems: Two Routes

1. Discretize the State Space:

- Approximate the continuous process for z with one with a discrete state space (often a Markov process)

Integration in DP problems: Two Routes

1. Discretize the State Space:

- Approximate the continuous process for z with one with a discrete state space (often a Markov process)
- Integration becomes summation.

Integration in DP problems: Two Routes

1. Discretize the State Space:

- Approximate the continuous process for z with one with a discrete state space (often a Markov process)
- Integration becomes summation.
- No need to interpolate, and more importantly *extrapolate*, the value function beyond the z grid

Integration in DP problems: Two Routes

1. Discretize the State Space:

- Approximate the continuous process for z with one with a discrete state space (often a Markov process)
- Integration becomes summation.
- No need to interpolate, and more importantly *extrapolate*, the value function beyond the z grid
- Often faster because it doesn't require interpolation in z direction.

Integration in DP problems: Two Routes

1. Discretize the State Space:

- Approximate the continuous process for z with one with a discrete state space (often a Markov process)
- Integration becomes summation.
- No need to interpolate, and more importantly *extrapolate*, the value function beyond the z grid
- Often faster because it doesn't require interpolation in z direction.
- Problematic if the value function is not smooth in z direction (e.g. if DP has a max operator). More on this later.

Discretizing Z

- OK: Tauchen's (1986) method.

Discretizing Z

- OK: Tauchen's (1986) method.
- Better: Tauchen and Hussey's (1991) method.

Discretizing Z

- OK: Tauchen's (1986) method.
- Better: Tauchen and Hussey's (1991) method.
- Arguably best: Rouwenhorst (1995) and variants (see Galindev and Lkhagvasuren (2009): "Discretization of Highly-Persistent Correllated AR(1) Shocks" and Kopecky and Suen (2009): "Finite State Markov-chain Approximations to Highly Persistent Processes")

Discretizing Z

- OK: Tauchen's (1986) method.
- Better: Tauchen and Hussey's (1991) method.
- Arguably best: Rouwenhorst (1995) and variants (see Galindev and Lkhagvasuren (2009): "Discretization of Highly-Persistent Correllated AR(1) Shocks" and Kopecky and Suen (2009): "Finite State Markov-chain Approximations to Highly Persistent Processes")
- Always simulate and compare your discrete approximation (autocorrelation(n), variance, skewness, histogram, etc) to the true process before using it.
  - You may be surprised at how often you get very different statistics. See next example:

Comparing Different Methods (Galindev and Lkhagvasuren):

Approximated AR(1) process

ρ	Tauch.			T-H			T-H-F			A-C			Rouwn.		
	$\frac{\log(1-\hat{\rho})}{\log(1-\rho)}$	$\frac{\hat{\sigma}}{\sigma}$	$\frac{\hat{\kappa}}{\kappa}$	$\frac{\log(1-\hat{\rho})}{\log(1-\rho)}$	$\frac{\hat{\sigma}}{\sigma}$	$\frac{\hat{\kappa}}{\kappa}$	$\frac{\log(1-\hat{\rho})}{\log(1-\rho)}$	$\frac{\hat{\sigma}}{\sigma}$	$\frac{\hat{\kappa}}{\kappa}$	$\frac{\hat{\rho}}{\rho}$	$\frac{\log(1-\hat{\rho})}{\log(1-\rho)}$	$\frac{\hat{\kappa}}{\kappa}$	$\frac{\log(1-\hat{\rho})}{\log(1-\rho)}$	$\frac{\hat{\sigma}}{\sigma}$	$\frac{\hat{\kappa}}{\kappa}$
N=9															
0.5	0.990	1.016	0.943	1.000	1.000	1.000	1.000	1.000	1.000	0.942	0.976	0.773	1.000	1.000	0.916
0.9	0.984	1.066	0.908	0.944	0.928	0.832	0.998	0.994	0.962	0.910	0.976	0.773	1.000	0.999	0.917
0.99	1.273	1.219	0.834	0.622	0.398	0.623	1.220	0.906	0.721	0.798	0.976	0.773	1.001	0.999	0.919
0.999	NA	NA	NA	0.426	0.130	0.601	NA	NA	NA	0.700	0.974	0.775	0.998	0.993	0.925
0.9999	NA	NA	NA	0.321	0.041	0.598	NA	NA	NA	0.650	0.973	0.776	0.988	0.959	0.961
N=19															
0.5	0.990	0.996	0.938	1.000	1.000	1.000	0.999	0.999	0.999	0.977	0.991	0.875	0.998	1.000	0.963
0.9	0.986	0.999	0.921	0.998	0.997	0.979	1.000	0.999	0.997	0.960	0.988	0.878	1.000	1.000	0.961
0.99	0.983	1.099	0.854	0.777	0.585	0.661	1.026	0.998	0.945	0.899	0.993	0.872	1.000	1.001	0.960
0.999	1.487	1.137	0.857	0.543	0.200	0.608	1.817	0.516	1.197	0.789	0.985	0.884	0.994	0.981	0.973
0.9999	NA	NA	NA	0.408	0.063	0.605	NA	NA	NA	0.718	0.993	0.871	1.002	0.994	0.891
N=49															
0.5	0.989	0.991	0.938	0.998	1.000	1.000	1.002	1.000	1.001	0.992	0.997	0.945	1.000	1.000	0.987
0.9	0.987	0.985	0.923	1.000	1.001	1.000	1.000	1.000	1.001	0.987	0.996	0.946	1.000	1.000	0.986
0.99	0.986	0.991	0.900	0.917	0.822	0.753	1.000	1.002	1.000	0.964	0.996	0.947	0.997	0.994	0.982
0.999	0.987	1.101	0.828	0.669	0.315	0.623	1.213	1.008	0.932	0.895	0.995	0.944	0.999	0.997	0.981
0.9999	1.491	0.756	8.004	0.506	0.102	0.610	NA	NA	NA	0.806	1.002	0.939	1.001	1.006	1.079

Table 3: The table compares the results from different methods in approximating an independent AR(1) process. Tauch. is Tauchen's (1986) method, T-H is Tauchen and Hussey's (1991) method, T-H-F is Flodén's alternative of Tauchen and Hussey's (1991) method, A-C is Adda and Cooper's (2003) method and Rouwn. is Rouwenhorst's (1995) method. NA denotes the cases where the corresponding method can not generate any data.

Comparing Different Methods: Look Closer

Tauchen (1986)

ρ	N=9					N=19					N=49				
	$\frac{\hat{\rho}}{\rho}$	$\frac{\mathrm{Var}(x)}{\mathrm{Var}(y)}$	$\frac{\mathrm{K}(x)}{\mathrm{K}(y)}$	$\frac{\mathrm{Var}(e)}{\mathrm{Var}(\epsilon)}$	$\frac{\mathrm{K}(e)}{\mathrm{K}(\epsilon)}$	$\frac{\hat{\rho}}{\rho}$	$\frac{\mathrm{Var}(x)}{\mathrm{Var}(y)}$	$\frac{\mathrm{K}(x)}{\mathrm{K}(y)}$	$\frac{\mathrm{Var}(e)}{\mathrm{Var}(\epsilon)}$	$\frac{\mathrm{K}(e)}{\mathrm{K}(\epsilon)}$	$\frac{\hat{\rho}}{\rho}$	$\frac{\mathrm{Var}(x)}{\mathrm{Var}(y)}$	$\frac{\mathrm{K}(x)}{\mathrm{K}(y)}$	$\frac{\mathrm{Var}(e)}{\mathrm{Var}(\epsilon)}$	$\frac{\mathrm{K}(e)}{\mathrm{K}(\epsilon)}$
0.5	0.998	1.057	0.976	1.058	0.984	0.998	1.006	0.974	1.008	0.983	0.998	0.995	0.973	0.997	0.982
0.9	0.998	1.219	0.948	1.238	1.007	0.999	1.033	0.960	1.045	0.997	0.999	0.993	0.962	1.004	0.998
0.99	1.008	1.651	0.876	0.227	41.42	1.000	1.329	0.900	1.330	1.416	1.000	1.037	0.942	1.064	0.999
0.999	NaN	NaN	NaN	NaN	NaN	1.001	1.636	0.842	0.011	1727	1.000	1.374	0.874	1.266	2.060
0.9999	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Rouwenhorst (1995)

ρ	N=9					N=19					N=49				
	$\frac{\hat{\rho}}{\rho}$	$\frac{\mathrm{Var}(x)}{\mathrm{Var}(y)}$	$\frac{\mathrm{K}(x)}{\mathrm{K}(y)}$	$\frac{\mathrm{Var}(e)}{\mathrm{Var}(\epsilon)}$	$\frac{\mathrm{K}(e)}{\mathrm{K}(\epsilon)}$	$\frac{\hat{\rho}}{\rho}$	$\frac{\mathrm{Var}(x)}{\mathrm{Var}(y)}$	$\frac{\mathrm{K}(x)}{\mathrm{K}(y)}$	$\frac{\mathrm{Var}(e)}{\mathrm{Var}(\epsilon)}$	$\frac{\mathrm{K}(e)}{\mathrm{K}(\epsilon)}$	$\frac{\hat{\rho}}{\rho}$	$\frac{\mathrm{Var}(x)}{\mathrm{Var}(y)}$	$\frac{\mathrm{K}(x)}{\mathrm{K}(y)}$	$\frac{\mathrm{Var}(e)}{\mathrm{Var}(\epsilon)}$	$\frac{\mathrm{K}(e)}{\mathrm{K}(\epsilon)}$
0.5	1.000	1.000	0.917	1.000	0.972	1.000	1.000	0.963	1.000	0.988	1.000	1.000	0.986	1.000	0.995
0.9	1.000	1.000	0.917	1.000	1.627	1.000	1.000	0.963	1.000	1.279	1.000	1.000	0.986	1.000	1.105
0.99	1.000	1.000	0.917	1.000	9.125	1.000	1.000	0.963	1.000	4.611	1.000	1.000	0.986	1.000	2.354
0.999	1.000	1.000	0.917	1.000	84.12	1.000	1.000	0.963	1.000	37.94	1.000	1.000	0.986	1.000	14.85
0.9999	1.000	1.000	0.917	1.000	834.1	1.000	1.000	0.963	1.000	371.2	1.000	1.000	0.986	1.000	139.8

Source: Galindev and Lkhagvasuren (RED, 2009)

Integration in DP problems, cont'd

2. Continuous Shocks:

Integration in DP problems, cont'd

2. Continuous Shocks:

1. Solve the DP problem for a finite number of z values in current period.

Integration in DP problems, cont'd

2. Continuous Shocks:

1. Solve the DP problem for a finite number of z values in current period.
2. But treat the conditional density $f(z'|z_j)$ as that of a continuous variable.

Integration in DP problems, cont'd

2. Continuous Shocks:

1. Solve the DP problem for a finite number of z values in current period.
2. But treat the conditional density $f(z'|z_j)$ as that of a continuous variable.
3. Use Romberg integration or Gaussian Quadrature-based methods to evaluate the integral.

Integration in DP problems, cont'd

2. Continuous Shocks:

1. Solve the DP problem for a finite number of z values in current period.
2. But treat the conditional density $f(z'|z_j)$ as that of a continuous variable.
3. Use Romberg integration or Gaussian Quadrature-based methods to evaluate the integral.
4. Sometimes slower, sometimes faster than option 1. Typically more accurate but not always worth the additional trouble.
5. Very useful when V is not smooth in z direction as noted above. Sometimes only feasible choice when accuracy is critical.

Gaussian Quadrature Based Formulas

- We would like to get an accurate approximation of $\mathrm{f(x)}$ over the interval $[\mathrm{a}, \mathrm{b}]$ by using a finite summation:

$$\int_{\mathrm{a}}^{\mathrm{b}} \mathrm{f(x)dx} \approx \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{v_j f(x_j)}$$

Gaussian Quadrature Based Formulas

- We would like to get an accurate approximation of $\mathrm{f(x)}$ over the interval $[\mathrm{a}, \mathrm{b}]$ by using a finite summation:

$$\int_{\mathrm{a}}^{\mathrm{b}} \mathrm{f(x)dx} \approx \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{v_j f(x_j)}$$

- Gaussian quadrature (GQ) provides a way to pick the abscissa points, $\{\mathrm{x_j}\}_{\mathrm{j}=1}^{\mathrm{N}}$, and associated weights, $\{\mathrm{v_j}\}_{\mathrm{j}=1}^{\mathrm{N}}$, to obtain high accuracy with low computational costs.

Gaussian Quadrature Based Formulas

- We would like to get an accurate approximation of $\mathrm{f(x)}$ over the interval $[\mathrm{a}, \mathrm{b}]$ by using a finite summation:

$$\int_{\mathrm{a}}^{\mathrm{b}} \mathrm{f(x)dx} \approx \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{v_j f(x_j)}$$

- Gaussian quadrature (GQ) provides a way to pick the abscissa points, $\{\mathrm{x_j}\}_{\mathrm{j}=1}^{\mathrm{N}}$, and associated weights, $\{\mathrm{v_j}\}_{\mathrm{j}=1}^{\mathrm{N}}$, to obtain high accuracy with low computational costs.
- However, GQ works well only if $\mathrm{f(x)}$ is well approximated by a certain class of polynomial of degree N or less.

Gaussian Quadrature Based Formulas

- We would like to get an accurate approximation of $f(x)$ over the interval $[a, b]$ by using a finite summation:

$$\int_a^b f(x)dx \approx \sum_{j=1}^{N} v_j f(x_j)$$

- Gaussian quadrature (GQ) provides a way to pick the abscissa points, $\{x_j\}_{j=1}^N$, and associated weights, $\{v_j\}_{j=1}^N$, to obtain high accuracy with low computational costs.
- However, GQ works well only if $f(x)$ is well approximated by a certain class of polynomial of degree N or less.
- This can be often fixed. Let $f(x) \equiv W(x)\frac{f(x)}{W(x)} = W(x)g(x)$ where $g(x)$ is well-approximated by polynomials and $W(x)$ is known.

$$\int_a^b W(x)g(x)dx \approx \sum_{j=1}^{N} w_j g(x_j) \qquad \textbf{where} \quad w_j = v_j/W(x_j).$$

Three Steps

1. For a given $\mathrm{f}(\mathrm{x})$, determine appropriate $\mathrm{W}(\mathrm{x})$ and $\mathrm{g}(\mathrm{x})$ above. ($\mathrm{W}(\mathrm{x}) = 1$ is a possible choice!)

Three Steps

1. For a given $\mathrm{f(x)}$, determine appropriate $\mathrm{W(x)}$ and $\mathrm{g(x)}$ above. ($\mathrm{W(x)} = 1$ is a possible choice!)
2. Contruct the family of orthonormal polynomials (with respect to $\mathrm{W(x)}$) up to degree N.

Three Steps

1. For a given $f(x)$, determine appropriate $W(x)$ and $g(x)$ above. ($W(x) = 1$ is a possible choice!)
2. Contruct the family of orthonormal polynomials (with respect to $W(x)$) up to degree N.
3. Find the zeros of the polynomial of degree N. These are the abscissas, $\{x_j\}_{j=1}^{N}$, that you need.

Three Steps

1. For a given $f(x)$, determine appropriate $W(x)$ and $g(x)$ above. ($W(x) = 1$ is a possible choice!)
2. Contruct the family of orthonormal polynomials (with respect to $W(x)$) up to degree N.
3. Find the zeros of the polynomial of degree N. These are the abscissas, $\{x_j\}_{j=1}^{N}$, that you need.
4. The weights are found by: $w_j = \frac{\langle p_{N-1}|p_{N-1}\rangle}{p_{N-1(x_j)}p'_N(x_j)}$ where $p'_N(x_j)$ is the derivative of the orthogonal polynomial at its root x_j.

Three Steps

1. For a given $f(x)$, determine appropriate $W(x)$ and $g(x)$ above. ($W(x) = 1$ is a possible choice!)
2. Contruct the family of orthonormal polynomials (with respect to $W(x)$) up to degree N.
3. Find the zeros of the polynomial of degree N. These are the abscissas, $\{x_j\}_{j=1}^{N}$, that you need.
4. The weights are found by: $w_j = \frac{\langle p_{N-1}|p_{N-1}\rangle}{p_{N-1(x_j)}p'_N(x_j)}$ where $p'_N(x_j)$ is the derivative of the orthogonal polynomial at its root x_j.

- For an arbitrary $W(x)$ this process is not trivial. Fortunately, for a number of standard weighting functions, we have exact expressions for the polynomial family and good approximations to the roots.

Key Polynomials

1. Gauss-Legendre: $W(x) = 1$ for $-1 < x < 1$
   - $(j+1)P_{j+1} = (2j+1)xP_j - jP_{j-1}$

Key Polynomials

1. Gauss-Legendre: $W(x) = 1$ for $-1 < x < 1$
   - $(j+1)P_{j+1} = (2j+1)xP_j - jP_{j-1}$
2. Gauss-Chebyshev: $W(x) = (1-x^2)^{-0.5} \quad -1 < x < 1$
   - $T_{j+1} = 2xT_j - T_{j-1}$

Key Polynomials

1. Gauss-Legendre: $W(x) = 1$ for $-1 < x < 1$
   - $(j+1)P_{j+1} = (2j+1)xP_j - jP_{j-1}$
2. Gauss-Chebyshev: $W(x) = (1-x^2)^{-0.5} \quad -1 < x < 1$
   - $T_{j+1} = 2xT_j - T_{j-1}$
3. Gauss-Hermite: $W(x) = e^{-x^2} \quad -\infty < x < \infty$
   - $H_{j+1} = 2xH_j - 2jH_{j-1}$

Key Polynomials

1. Gauss-Legendre: $W(x) = 1$ for $-1 < x < 1$
   - $(j+1)P_{j+1} = (2j+1)xP_j - jP_{j-1}$
2. Gauss-Chebyshev: $W(x) = (1-x^2)^{-0.5} \quad -1 < x < 1$
   - $T_{j+1} = 2xT_j - T_{j-1}$
3. Gauss-Hermite: $W(x) = e^{-x^2} \quad -\infty < x < \infty$
   - $H_{j+1} = 2xH_j - 2jH_{j-1}$

- In a DP problem with Gaussian shocks, take $W(x)$ to be the Normal density and $V(x) \equiv g(x)$. If $V(x)$ does not have kinks or poles, we can still apply GQ using Hermite polynomials.

Key Polynomials

1. Gauss-Legendre: $W(x) = 1$ for $-1 < x < 1$
   - $(j+1)P_{j+1} = (2j+1)xP_j - jP_{j-1}$
2. Gauss-Chebyshev: $W(x) = (1-x^2)^{-0.5} \quad -1 < x < 1$
   - $T_{j+1} = 2xT_j - T_{j-1}$
3. Gauss-Hermite: $W(x) = e^{-x^2} \quad -\infty < x < \infty$
   - $H_{j+1} = 2xH_j - 2jH_{j-1}$

- In a DP problem with Gaussian shocks, take $W(x)$ to be the Normal density and $V(x) \equiv g(x)$. If $V(x)$ does not have kinks or poles, we can still apply GQ using Hermite polynomials.
- As you will see in coming slides, Legendre and Chebyshev polynomials are bounded between 0 and 1, whereas Hermite polynomials are unbounded. (So be careful when using the latter).

Legendre Polynomials

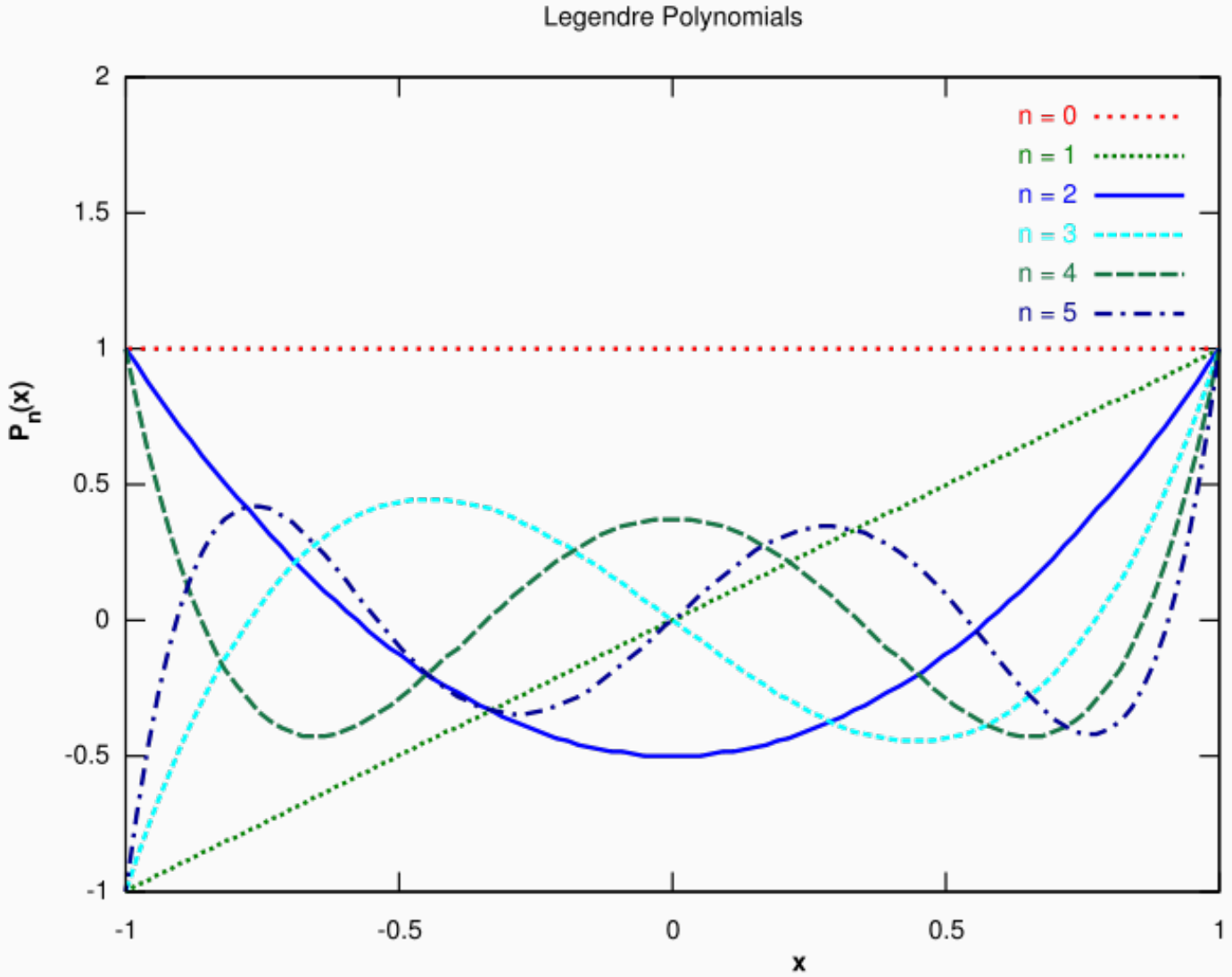
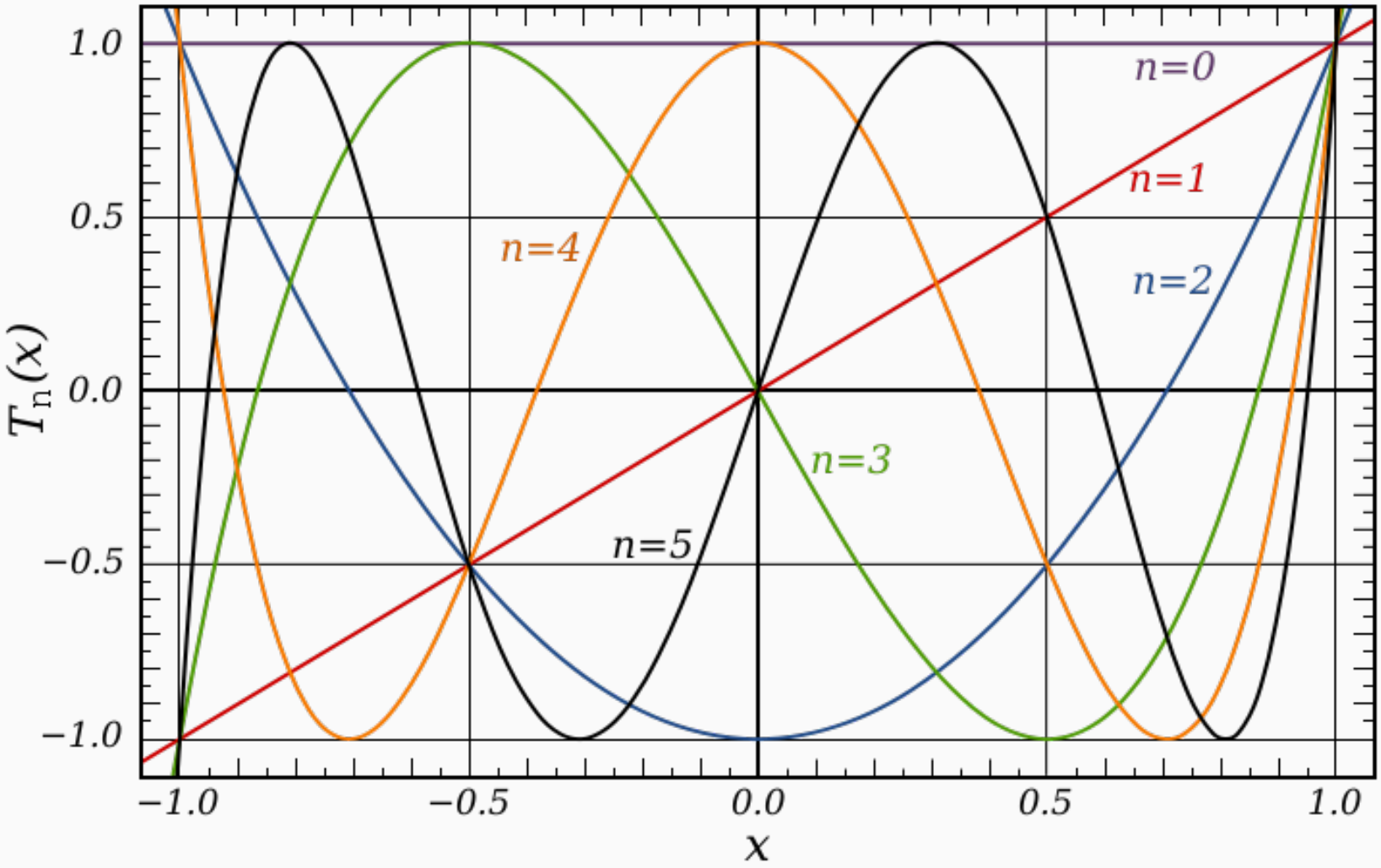


Figure 1: Legendre Polynomials: $W(x) = 1 \quad -1 < x < 1$

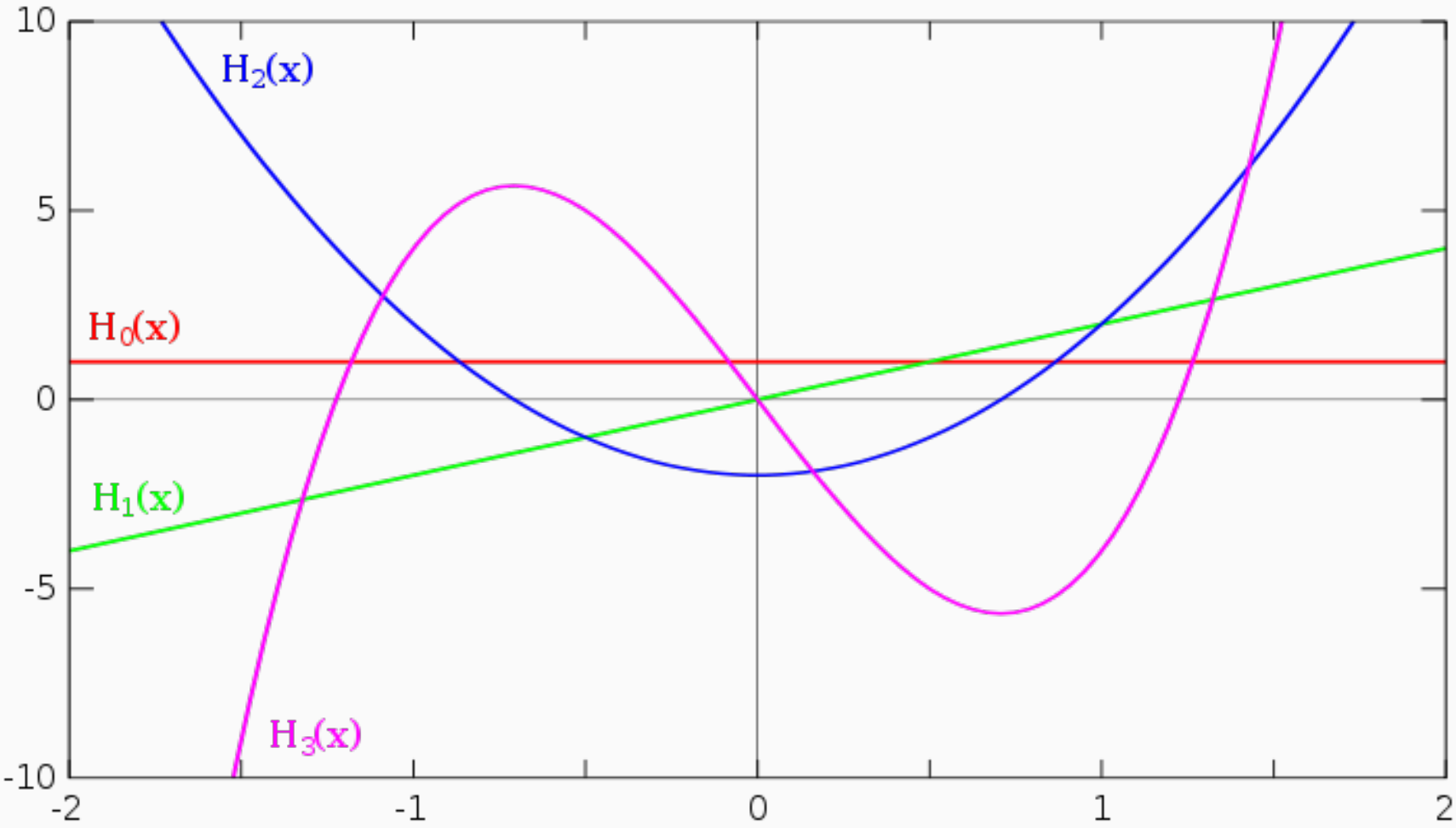
Chebyshev Polynomials

Figure 2: Chebyshev Polynomials: $W(x) = (1 - x^2)^{-0.5} \quad -1 < x < 1$



Hermite Polynomials

Figure 3: Hermite Polynomials: $W(x) = e^{-x^2} \quad -\infty < x < \infty$



Drawbacks of GQ

- Major problem: No practical error estimate! You have no idea how good or bad the approximation to the true integral is.

Drawbacks of GQ

- Major problem: No practical error estimate! You have no idea how good or bad the approximation to the true integral is.
- Theoretical error estimate is:

$$E_n(f) = \frac{f^{2n}(\xi)}{(2n)!k_n^2} \qquad a < \xi < b,$$

and for some relevant functions this is explosive (e.g., $f = x^{-1}$ close to zero).

Drawbacks of GQ

- Major problem: No practical error estimate! You have no idea how good or bad the approximation to the true integral is.
- Theoretical error estimate is:

$$E_n(f) = \frac{f^{2n}(\xi)}{(2n)!k_n^2} \qquad a < \xi < b,$$

  and for some relevant functions this is explosive (e.g., $f = x^{-1}$ close to zero).
- Performance degrades quickly when integrand has kinks and non-polynomial properties.

Drawbacks of GQ

- Major problem: No practical error estimate! You have no idea how good or bad the approximation to the true integral is.
- Theoretical error estimate is:

$$E_n(f) = \frac{f^{2n}(\xi)}{(2n)!k_n^2} \qquad a < \xi < b,$$

  and for some relevant functions this is explosive (e.g., $f = x^{-1}$ close to zero).
- Performance degrades quickly when integrand has kinks and non-polynomial properties.
- It is not nested.

Drawbacks of GQ

- Major problem: No practical error estimate! You have no idea how good or bad the approximation to the true integral is.
- Theoretical error estimate is:

$$E_n(f) = \frac{f^{2n}(\xi)}{(2n)!k_n^2} \qquad a < \xi < b,$$

  and for some relevant functions this is explosive (e.g., $f = x^{-1}$ close to zero).
- Performance degrades quickly when integrand has kinks and non-polynomial properties.
- It is not nested.
- Modern integrators fix most of these problems.

Modern Integrators

1. A quadrature rule is said to be nested if for an n_1-point rule, there is an n_2-point rule (with $n_2 > n_1$) that reuses the original n_1 abscissas and the associated computations performed for the n_1-point rule.

Modern Integrators

1. A quadrature rule is said to be nested if for an n_1-point rule, there is an n_2-point rule (with $n_2 > n_1$) that reuses the original n_1 abscissas and the associated computations performed for the n_1-point rule.

2. A composite quadrature rule subdivides the original interval $[a, b]$ into subintervals and applies a fixed quadrature rule within each subinterval.

Modern Integrators

1. A quadrature rule is said to be nested if for an n_1-point rule, there is an n_2-point rule (with $n_2 > n_1$) that reuses the original n_1 abscissas and the associated computations performed for the n_1-point rule.

2. A composite quadrature rule subdivides the original interval $[a, b]$ into subintervals and applies a fixed quadrature rule within each subinterval.

3. An automatic rule is one that adjusts the number of function evaluations to achieve a certain accuracy for the integral.

Modern Integrators

1. A quadrature rule is said to be nested if for an n_1-point rule, there is an n_2-point rule (with $n_2 > n_1$) that reuses the original n_1 abscissas and the associated computations performed for the n_1-point rule.

2. A composite quadrature rule subdivides the original interval $[a, b]$ into subintervals and applies a fixed quadrature rule within each subinterval.

3. An automatic rule is one that adjusts the number of function evaluations to achieve a certain accuracy for the integral.

4. An adaptive rule chooses the points in which the integrand is evaluated depending on the nature of the integrand—so the rule is adapted to the integrand.

Gauss-Kronrod: Major Improvement

- Unless you have a very good understanding of the shape of your integrand, GQ should never be your first choice.

Gauss-Kronrod: Major Improvement

- Unless you have a very good understanding of the shape of your integrand, GQ should never be your first choice.
- Gauss-Kronrod works in two steps. First constructs an n_1 point GQ integral.
- Then given these n_1 nodes, it inserts one new node in between each pair of existing nodes—in an optimal fashion (for polynomial-like functions)

Gauss-Kronrod: Major Improvement

- Unless you have a very good understanding of the shape of your integrand, GQ should never be your first choice.
- Gauss-Kronrod works in two steps. First constructs an n_1 point GQ integral.
- Then given these n_1 nodes, it inserts one new node in between each pair of existing nodes—in an optimal fashion (for polynomial-like functions)
- Key Advantage: comparing the n_1 point integral and the $2\mathrm{n}_1 + 1$ point one provides an error estimate.

Gauss-Kronrod: Major Improvement

- Unless you have a very good understanding of the shape of your integrand, GQ should never be your first choice.
- Gauss-Kronrod works in two steps. First constructs an n_1 point GQ integral.
- Then given these n_1 nodes, it inserts one new node in between each pair of existing nodes—in an optimal fashion (for polynomial-like functions)
- Key Advantage: comparing the n_1 point integral and the $2n_1 + 1$ point one provides an error estimate.
- Slight loss relative to GQ in polynomial accuracy but typically outperforms in non-polynomial integrands.

Pause: Open-Source Software Libraries

- Important tip: Learn how to use wrappers. You can call very fast Fortran or C libraries from Python, Julia, Matlab, etc.
- GNU Scientific library: Written in C but can use wrappers.
- NetLib (netlib.org): lots of very useful routines in Fortran.
- Quadpack: The automatic integrators we talk about next can be found here: www.netlib.org/quadpack/
- If you are using Julia or another suitable language, you can call both libraries in C and Fortran.

Automatic Integrators

- You specify the accuracy and they do the rest.
- For general integrands that may include kinks or poles, one of the best choices is QAGS (part of QUADPACK).
- Available for free online. Also part of NAG and IMSL.
- If integrand has no singularity QAG is as good, and maybe faster.

Romberg Integration

- This is a simple and elegant method with some desirable properties.
- It works well even when the function has kinks and other non-polynomial features.
- It is often a good first choice when you're not sure about the shape of the value function and accuracy trumps speed.

Romberg Integration

- This is a simple and elegant method with some desirable properties.
- It works well even when the function has kinks and other non-polynomial features.
- It is often a good first choice when you're not sure about the shape of the value function and accuracy trumps speed.
- Basic idea: Use composite trapezoidal rule iteratively, subdividing into finer regions and quickly improving accuracy.

Romberg Integration

- This is a simple and elegant method with some desirable properties.
- It works well even when the function has kinks and other non-polynomial features.
- It is often a good first choice when you're not sure about the shape of the value function and accuracy trumps speed.
- Basic idea: Use composite trapezoidal rule iteratively, subdividing into finer regions and quickly improving accuracy.
- The *Euler-McLaurin summation formula* for integration error:

$$\mathrm{e_k(f)} = \mathrm{c_1 h^2 + c_2 h^4 + ... + c_N h^{2N} + \mathcal{O}(h^{2N+2})}, \tag{1}$$

with $\mathrm{N} = 2^{\mathrm{k-1}}$ subintervals, where h is interval width, and $\mathrm{c_i}$, $\mathrm{i} = 1, 2, ..., \mathrm{n}$ are coefficients that don't depend on h.

Romberg Integration

- Notice: Error formula only has even powered polynomials. (Which makes trapezoidal rule a good starting point).

Romberg Integration

- Notice: Error formula only has even powered polynomials. (Which makes trapezoidal rule a good starting point).
- Romberg's idea is to combine estimates with step sizes $\mathrm{h}_1, \mathrm{h}_2, ...\mathrm{h}_\mathrm{k}$ in a particular way

$$\mathrm{T}_1(\mathrm{f}) = \frac{(\mathrm{b}-\mathrm{a})}{2}\left(\mathrm{f}(\mathrm{a})+\mathrm{f}(\mathrm{b})\right)$$
$$\mathrm{T}_2(\mathrm{f}) = \frac{(\mathrm{b}-\mathrm{a})}{4}\left(\mathrm{f}(\mathrm{a})+2\mathrm{f}(\frac{\mathrm{a}+\mathrm{b}}{2})+\mathrm{f}(\mathrm{b})\right).$$

Romberg Integration

- Notice: Error formula only has even powered polynomials. (Which makes trapezoidal rule a good starting point).
- Romberg's idea is to combine estimates with step sizes $h_1, h_2, ...h_k$ in a particular way

$$T_1(f) = \frac{(b-a)}{2}(f(a) + f(b))$$
$$T_2(f) = \frac{(b-a)}{4}\left(f(a) + 2f(\frac{a+b}{2}) + f(b)\right).$$

- The Euler-McLaurin formula says:

$$T_1(f) = I(f) + c_1 h^2 + \mathcal{O}(h^4) \tag{2}$$
$$T_2(f) = I(f) + c_1\left(\frac{h}{2}\right)^2 + \mathcal{O}(h^4). \tag{3}$$

Romberg Integration

- Using these two equations, we can get:

$$T_1(f) - 4T_2(f) = -3I(f) + \mathcal{O}(h^4) \Rightarrow I(f) = \underbrace{\frac{4T_2(f) - T_1(f)}{3}}_{R_{2,2}} + \mathcal{O}(h^4).$$

Romberg Integration

- Using these two equations, we can get:

$$\mathrm{T}_1(\mathrm{f}) - 4\mathrm{T}_2(\mathrm{f}) = -3\mathrm{I}(\mathrm{f}) + \mathcal{O}(\mathrm{h}^4) \Rightarrow \mathrm{I}(\mathrm{f}) = \underbrace{\frac{4\mathrm{T}_2(\mathrm{f}) - \mathrm{T}_1(\mathrm{f})}{3}}_{\mathrm{R}_{2,2}} + \mathcal{O}(\mathrm{h}^4).$$

- Error improved to h^4! This step is called extrapolation.
- Keep subdividing and extrapolating to quickly improve accuracy quadratically in every step.

Romberg Integration

- Using these two equations, we can get:

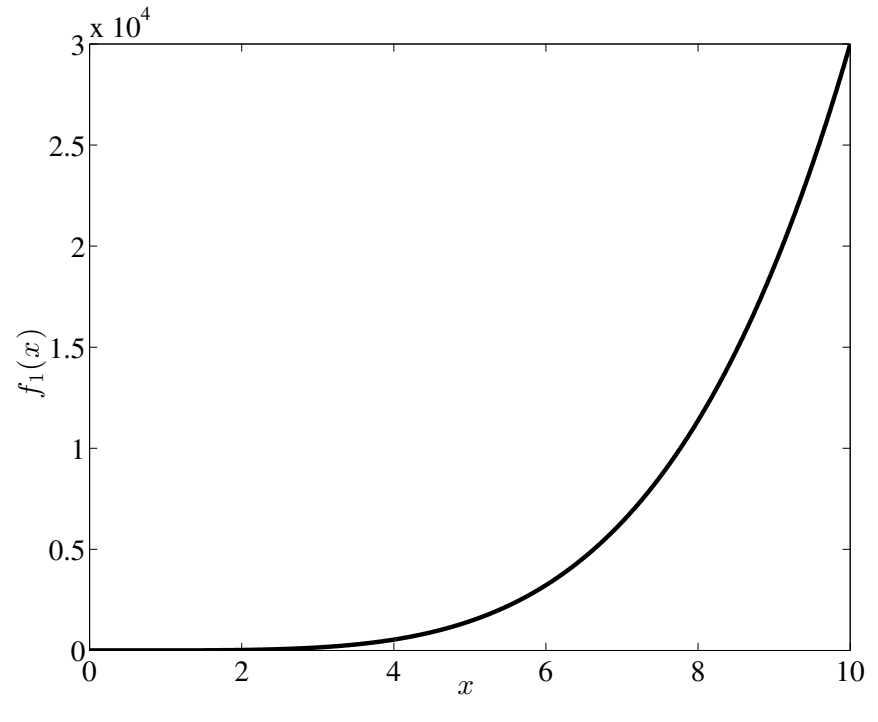
$$\mathrm{T}_1(\mathrm{f}) - 4\mathrm{T}_2(\mathrm{f}) = -3\mathrm{I}(\mathrm{f}) + \mathcal{O}(\mathrm{h}^4) \Rightarrow \mathrm{I}(\mathrm{f}) = \underbrace{\frac{4\mathrm{T}_2(\mathrm{f}) - \mathrm{T}_1(\mathrm{f})}{3}}_{\mathrm{R}_{2,2}} + \mathcal{O}(\mathrm{h}^4).$$

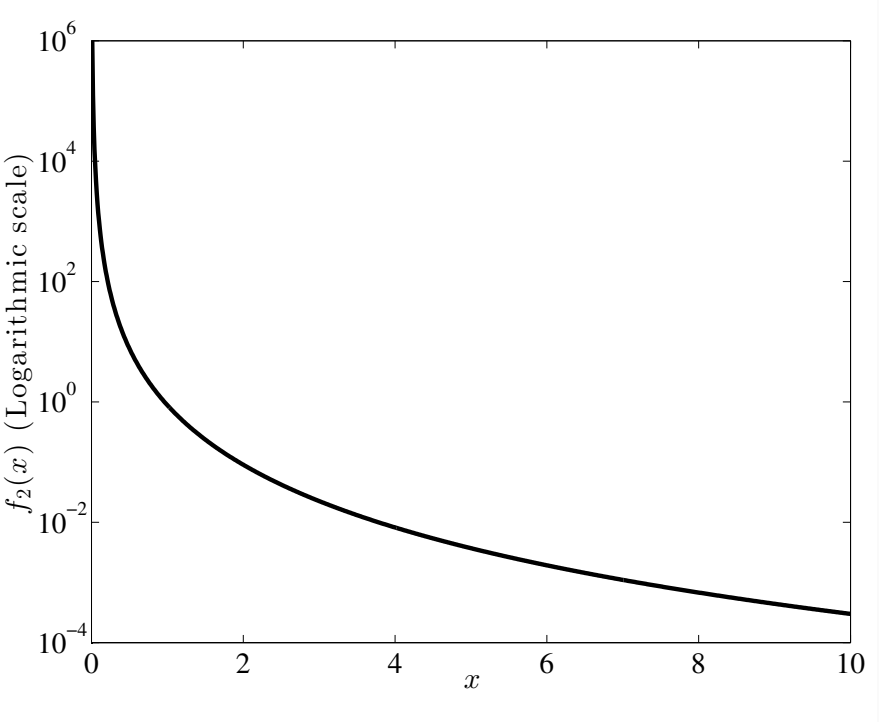
- Error improved to h^4! This step is called extrapolation.
- Keep subdividing and extrapolating to quickly improve accuracy quadratically in every step.
- But each subsequent step has double the intervals!

Benchmarking Integrators

Benchmarking Integrators

Figure 4: Smooth Integrands Without (Left) and With (Right) A Singularity





Five Test Functions

$$f_1 = x^4 \times \mathsf{log}(x + \sqrt{x^2 + 1})$$

$$f_2 = x^{-4} \times \mathsf{log}(x + \sqrt{x^2 + 1})$$

$$f_3 = x^{-5} \times \mathsf{exp}(-x^2)$$

$$f_4 = \begin{cases} \sqrt{x} & \text{if } x < 5 \\ \sqrt{5} + \sqrt{x - 5} & \text{if } x \geq 5 \end{cases}$$

$$f_5 = \sqrt{x} + \begin{cases} 0.1\sqrt{x} & \text{if } x \in \{[1, 2], [3, 4]\} \\ -0.1\sqrt{x - 1} & \text{if } x \in \{[2, 3], [4, 5]\} \end{cases}$$

Benchmarks

	$f_1\ (k = 4)$			$f_2(k = -4)$		
Method	Error	Time	f calls	Error	Time	f calls
Trapezoid	$1(-7)$	1.4898	4097	$4(-6)$	15.0195	32769
Romberg	$-5(-8)$	0.0107	17	$2(-6)$	0.4713	1025
GL (10)	$9(-13)$	0.0059	10	$-4(-1)$	0.0064	10
	$-1(-14)$	0.0479	100	$4(-8)$	0.0600	100
	$-1(-14)$	0.2035	500	$4(-8)$	0.2386	500
GK (`QAGS`)	$3(-10)$	0.0098	21	$4(-8)$	0.1309	231
GK (`QAG`)	$-3(-8)$	0.0074	15	$2(-8)$	0.1150	195
C-Curtis	$4(-7)$	0.0029	7	1.9	0.0035	7
	$-8(-9)$	0.0112	19	$-7(-6)$	0.0447	55
	$-8(-9)$	0.0118	19	$3(-7)$	0.4430	487

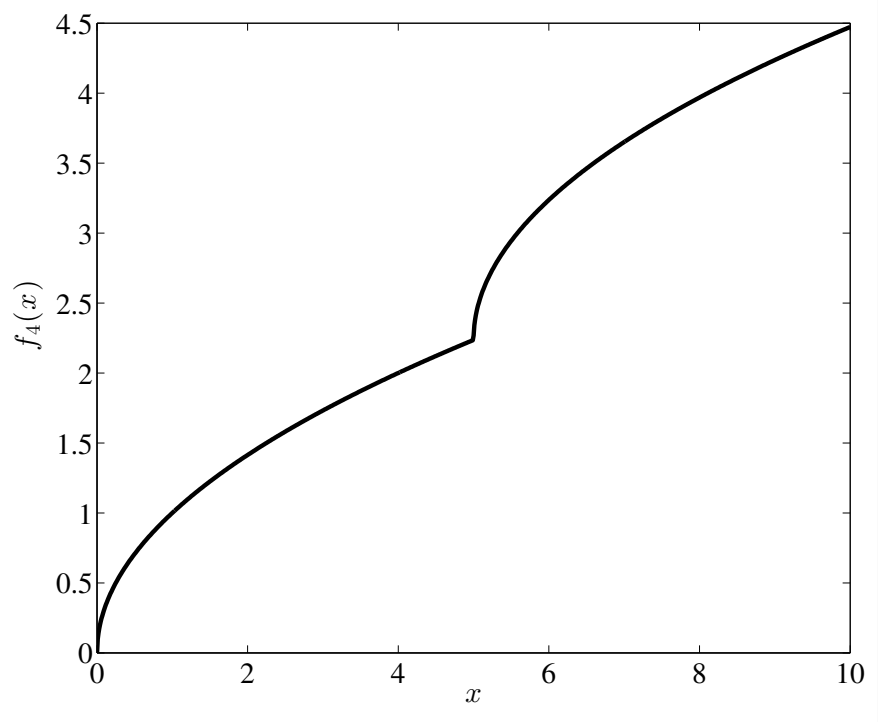
Benchmarks

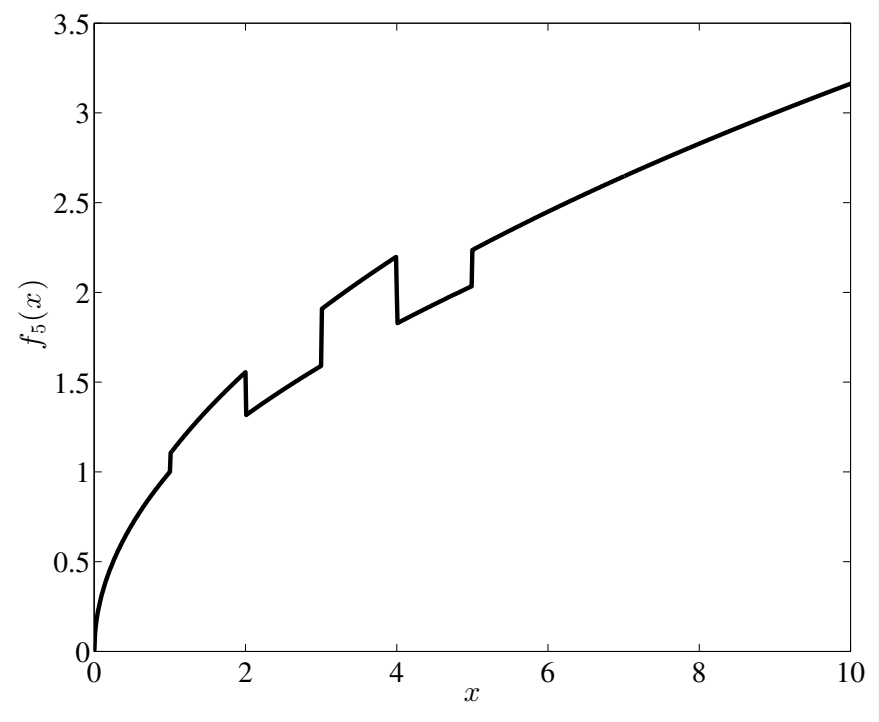
Table 1: Conditional Expectation With Respect to Standard Normal Density

$I(y) = \int_{0.1}^{10} \left(x^{-5} \exp(-x^2)\right) dx$			
Method	**Error**	Time	**f calls**
Trapezoid	$9(-6)$	12.7978	32769
Romberg	$5(-8)$	0.7604	2049
Gauss-Legendre	$-6(-1)$	0.0057	10
	$-4(-3)$	0.0179	30
	$-3(-14)$	0.2004	500
GK-qags	$-2(-9)$	0.1282	273
GK-qag	$-2(-8)$	0.1081	225
Clenshaw-Curtis	$3.3(0)$	0.0028	7
	$-3(-5)$	0.0382	55
	$3(-7)$	0.3799	487

Kinks and Jumps

Figure 5: Integrand With A Kink (Left) and With Jumps (Right)





Kinks and Jumps

Table 2: Integrand f_5: Square-Root Function With Jumps

Function	Error$\times 10^6$	Time	f calls
Trapezoid	$3(-5)$	3.4456	32769
Romberg	$-4(-5)$	1.7091	16385
Gauss-Legendre	$-1(-1)$	0.0065	10
	$4(-3)$	0.0948	100
	$1(-3)$	0.0891	500
GK-gags	$2(-8)$	0.3934	1575
GK-qag	$3(-9)$	0.6339	2535
Clenshaw-Curtis	$-1(-1)$	0.0014	7
	$1(-2)$	0.0299	55
	$4(-5)$	2.5734	4375

Another Critical Issue: Extrapolating While Integrating

- With an autocorrelated process for z (e.g. AR(1)), $W(x)$ needs to be the density of the shock tomorrow conditional on its value today.

Another Critical Issue: Extrapolating While Integrating

- With an autocorrelated process for z (e.g. AR(1)), $W(x)$ needs to be the density of the shock tomorrow conditional on its value today.
- Now consider solving the DP problem for z_1 or z_N (lowest or highest grid point). We need to evaluate:

$$\int V(w', z') f(z'|z_N) dz'$$

where $f(z'|z_N)$ is Gaussian with mean ρz_N and variance σ_ϵ^2.

Another Critical Issue: Extrapolating While Integrating

- With an autocorrelated process for z (e.g. AR(1)), $W(x)$ needs to be the density of the shock tomorrow conditional on its value today.
- Now consider solving the DP problem for z_1 or z_N (lowest or highest grid point). We need to evaluate:

$$\int V(w', z') f(z'|z_N) dz'$$

  where $f(z'|z_N)$ is Gaussian with mean ρz_N and variance σ_ϵ^2.
- If $\rho z_N + 2 \times \sigma_\epsilon > z_N$ you will be extrapolating $V(w, z)$ beyond the upper end of the grid.

Another Critical Issue: Extrapolating While Integrating

- With an autocorrelated process for z (e.g. AR(1)), $W(x)$ needs to be the density of the shock tomorrow conditional on its value today.
- Now consider solving the DP problem for z_1 or z_N (lowest or highest grid point). We need to evaluate:

$$\int V(w', z') f(z'|z_N) dz'$$

  where $f(z'|z_N)$ is Gaussian with mean ρz_N and variance σ_ϵ^2.
- If $\rho z_N + 2 \times \sigma_\epsilon > z_N$ you will be extrapolating $V(w, z)$ beyond the upper end of the grid.
- If the extrapolation is sufficiently inaccurate (which may very well be!) and the probability weight is non-negligible you'll have a serious problem!

Another Critical Issue: Extrapolating While Integrating

- With an autocorrelated process for z (e.g. AR(1)), $W(x)$ needs to be the density of the shock tomorrow conditional on its value today.
- Now consider solving the DP problem for z_1 or z_N (lowest or highest grid point). We need to evaluate:

$$\int V(w', z') f(z'|z_N) dz'$$

  where $f(z'|z_N)$ is Gaussian with mean ρz_N and variance σ_ϵ^2.
- If $\rho z_N + 2 \times \sigma_\epsilon > z_N$ you will be extrapolating $V(w, z)$ beyond the upper end of the grid.
- If the extrapolation is sufficiently inaccurate (which may very well be!) and the probability weight is non-negligible you'll have a serious problem!
- Especially problematic if ρ and/or σ_ϵ is high.

Integration with Kinks

- A particularly tricky issue arises when the value function is not smooth, e.g., has a kink in z direction.

Integration with Kinks

- A particularly tricky issue arises when the value function is not smooth, e.g., has a kink in z direction.
- Discretizing the z grid almost never works well in these cases!

Integration with Kinks

- A particularly tricky issue arises when the value function is not smooth, e.g., has a kink in z direction.
- Discretizing the z grid almost never works well in these cases!
- If you can locate the kink, the best approach is to integrate the two parts separately:

$$\int_{-\infty}^{\overline{z}} V_1(w', z') f(z'|z) dz' + \int_{\overline{z}}^{\infty} V_2(w', z') f(z'|z_N) dz'$$

Integration with Kinks

- A particularly tricky issue arises when the value function is not smooth, e.g., has a kink in z direction.
- Discretizing the z grid almost never works well in these cases!
- If you can locate the kink, the best approach is to integrate the two parts separately:

$$\int_{-\infty}^{\overline{z}} V_1(w', z') f(z'|z) dz' + \int_{\overline{z}}^{\infty} V_2(w', z') f(z'|z_N) dz'$$

- Notice that the densities are truncated normals. So GQ is a bit tricky to implement.

Integration with Kinks

- A particularly tricky issue arises when the value function is not smooth, e.g., has a kink in z direction.
- Discretizing the z grid almost never works well in these cases!
- If you can locate the kink, the best approach is to integrate the two parts separately:

$$\int_{-\infty}^{\overline{z}} V_1(w', z') f(z'|z) dz' + \int_{\overline{z}}^{\infty} V_2(w', z') f(z'|z_N) dz'$$

- Notice that the densities are truncated normals. So GQ is a bit tricky to implement.
- I use Romberg or QAGS in this case.

Integration with Kinks

- A particularly tricky issue arises when the value function is not smooth, e.g., has a kink in z direction.
- Discretizing the z grid almost never works well in these cases!
- If you can locate the kink, the best approach is to integrate the two parts separately:

$$\int_{-\infty}^{\overline{z}} V_1(w', z') f(z'|z) dz' + \int_{\overline{z}}^{\infty} V_2(w', z') f(z'|z_N) dz'$$

- Notice that the densities are truncated normals. So GQ is a bit tricky to implement.
- I use Romberg or QAGS in this case.
- If you cannot locate the kink, you can still use Romberg but is likely to be less accurate and dependable.