Ch 5. Numerical Integration and Differentiation

Jongchan Lee

Department of Economics
University of California, Santa Cruz

Nov. 10 2014

Overview of Ch. 5

1. Numerical integration or numerical quadrature
   - Approximate a definite integral with a weighted sum of function values

$$\int_I f(x)w(x)dx \approx \sum_{i=0}^{n} w_i f(x_i)$$

   - The methods differ only in how the quadrature weights w_i and the quadrature nodes x_i are chosen.
   - Newton-Cotes methods, Gaussian quadrature methods, and Monte Carlo and quasi-Monte Carlo integration methods
2. Computing finite difference approximations for the derivatives of a real-valued function
   - Solving differential equations and its application to initial value problems

5.1 Newton-Cotes Methods

5.1.1 Trapezoid rule

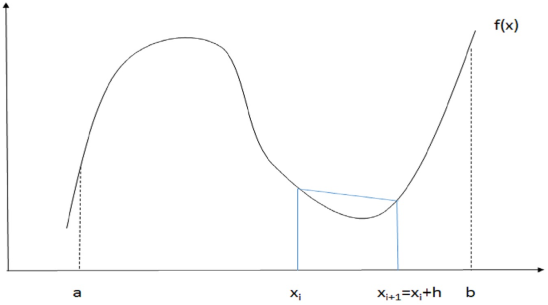
- Piecewise linear approximations to the integrand f.

  $x_i = a + (i - 1)h$ for $i = 1, 2, ..., n$. where $h = (b - a)/(n - 1)$

- The nodes x_i divide the interval $[a, b]$ into $n - 1$ subintervals of equal length h.

$$\int_{x_i}^{x_i+1} f(x)dx \approx \frac{h}{2}[f(x_i) + f(x_{i+1})]$$

5.1.1 Trapezoid rule



5.1 Newton-Cotes Methods

5.1.1 Trapezoid rule

$$\int_{a}^{b} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

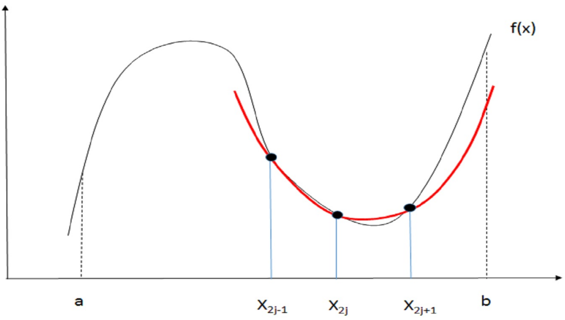
where $w_1 = w_n = h/2$ and $w_i = h$ otherwise.

- If the integrand f is smooth, approximation error is $O(h^2)$, the error shirinks quadratically with the width of the subintervals (h).

5.1.2. Simpson's rule

- Piecewise quadratic approximations to the integrand f.

$x_i = a+(i-1)h$ for $i = 1, 2, ..., n$. where $h = (b-a)/(n-1)$ and n is odd.



5.1.2. Simpson's rule

- The area under the quadratic function provides an estimate of the area under f over the sub interval: (Lagrange polynomial interpolation is used)

$$\int_{x_{2j-1}}^{x_{2j+1}} f(x)dx \approx \frac{h}{3}[f(x_{2j-1}) + 4f(x_{2j}) + f(x_{2j+1})]$$

5.1.2. Simpson's rule

- Summing up the are over [a,b] gives Simpson's rule:

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

where $w_1 = w_n = h/3$ and $w_i = 4h/3$ if i is even and $w_i = 2h/3$ if i is odd.

- Simpson's rule is third order exact.
- If the intergrand is smooth, approximation error $O(h^4)$. Twice more accurate than Trapezoid rule.
- Simpson's rule is not recommended if the integrand shows discontinuites in its first derivative, such as corner solutions case.

5.2 Gaussian Quadrature

- For a specific weighted function w defined on an interval $I \subset R$ of the real line, and for a given order of approximation n, the quadrature nodes $x_1, x_2, ..., x_n$ and quadrature weights $w_1, w_2, ..., w_n$ are chosen so as to satisfy the $2n$ "moment-matching" conditions:

$$\int_I x^k w(x) dx = \sum_{i-1}^{n} w_i x_i^k \text{ for } k = 0, 1, 2, ..., 2n-1$$

5.2 Gaussian Quadrature

- The integral approximation is then computed by forming the prescribed weighted sum of function values at the prescribed nodes:

$$\int_I f(x)w(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

- An n-point Gaussian quadrature rule is order 2n-1 exact.
- Thus, if f can be closely approximated by a polynomial, Gaussian quadrature should provide an accurate approximation to the integral.

5.2 Gaussian Quadrature

- When the weight function w is the known probability density function of some continuous variable $\widetilde{X}$, Gaussian quadrature essentially "discretizes" the continuous random variable $\widetilde{X}$.
- Mass points x_i and probabilities w_i are chosen to satisfy the condition:

$$E[\widetilde{X}^k] = \sum_{i=1}^{n} w_i x_i^k \text{ for } k = 0, 1, 2, ..., 2n - 1$$

- Discrete approximation of the expectation of any function of countinous r.v. $\widetilde{X}$:

$$E[f(\widetilde{X})] = \int_I f(x)w(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

5.2 Gaussian Quadrature

- Example : three point approximatoin (n=3, k=2n-1=5) to the standard normal distribution ($\widetilde{Z}$)

$$E[\widetilde{Z}^k] = \sum_{i=1}^{n} = 3w_i x_i^k \text{ for } k = 0, 1, 2, ..., 5$$

$$E[\widetilde{Z}^0] = 1 = w_1 + w_2 + w_3$$
$$E[\widetilde{Z}^1] = 0 = w_1 x_1 + w_2 x_2 + w_3 x_3$$
$$E[\widetilde{Z}^2] = 1 = w_1 x_1^2 + w_2 x_2^2 + w_3 x_3^2$$
$$E[\widetilde{Z}^3] = 0 = w_1 x_1^3 + w_2 x_2^3 + w_3 x_3^3$$
$$E[\widetilde{Z}^4] = 3 = w_1 x_1^4 + w_2 x_2^4 + w_3 x_3^4$$
$$E[\widetilde{Z}^5] = 0 = w_1 x_1^5 + w_2 x_2^5 + w_3 x_3^5$$

5.2 Gaussian Quadrature

WolframAlpha computational... knowledge engine

$1=w_1+w_2+w_3$, $0=w_1x_1+w_2x_2+w_3x_3$, $1=w_1x_1^2+w_2x_2^2+w_3x_3^2$, $0=w_1x_1^3+w_2x$:

Examples Random

Input:

$$\{1 = w_1 + w_2 + w_3,\ 0 = w_1 x_1 + w_2 x_2 + w_3 x_3,\ 1 = w_1 x_1^2 + w_2 x_2^2 + w_3 x_3^2,\ 0 = w_1 x_1^3 + w_2 x_2^3 + w_3 x_3^3,\ 3 = w_1 x_1^4 + w_2 x_2^4 + w_3 x_3^4,\ 0 = w_1 x_1^5 + w_2 x_2^5 + w_3 x_3^5\}$$

Alternate form:

$$\{w_1 + w_2 + w_3 = 1,\ w_3 x_3 = -w_1 x_1 - w_2 x_2,\ w_1 x_1^2 + w_2 x_2^2 + w_3 x_3^2 = 1,\ w_3 x_3^3 = -w_1 x_1^3 - w_2 x_2^3,\ w_1 x_1^4 + w_2 x_2^4 + w_3 x_3^4 = 3,\ w_3 x_3^5 = -w_1 x_1^5 - w_2 x_2^5\}$$

Solutions: Approximate forms More solutions

$$w_1 = \frac{1}{6},\quad w_2 = \frac{1}{6},\quad w_3 = \frac{2}{3},\quad x_1 = -\sqrt{3},\quad x_2 = \sqrt{3},\quad x_3 = 0$$

$$w_1 = \frac{1}{6},\quad w_2 = \frac{1}{6},\quad w_3 = \frac{2}{3},\quad x_1 = \sqrt{3},\quad x_2 = -\sqrt{3},\quad x_3 = 0$$

$$w_1 = \frac{1}{6},\quad w_2 = \frac{2}{3},\quad w_3 = \frac{1}{6},\quad x_1 = -\sqrt{3},\quad x_2 = 0,\quad x_3 = \sqrt{3}$$

$$w_1 = \frac{1}{6},\quad w_2 = \frac{2}{3},\quad w_3 = \frac{1}{6},\quad x_1 = \sqrt{3},\quad x_2 = 0,\quad x_3 = -\sqrt{3}$$

$$w_1 = \frac{2}{3},\quad w_2 = \frac{1}{6},\quad w_3 = \frac{1}{6},\quad x_1 = 0,\quad x_2 = -\sqrt{3},\quad x_3 = \sqrt{3}$$

5.2 Gaussian Quadrature

- Computing n-degree Gaussian nodes and weights are not easy as we have to solve 2n non-linear equations for w_i and x_i.
- But efficient and specialized numerical routines for well known functions (such as normal, gamma, exponential, chi-square, beta distribution) are available.
- Generalizatoin of Gaussian quadrature rules to higher-dimensional integration
- Discretizing lognorlam random variable

5.3 Monte Carlo Integration

- Motivated by Law of Large numbers
  - LLM : x_1, x_2,... x_n are independent realizations of a random variable $\widetilde{X}$ and f is a continuous function, then

$$\lim_{x \to \infty} \frac{1}{n} \sum_{n=1}^{n} f(x_i) = E[f(\widetilde{X})]$$

- Monte Carlo integration
  - A random sample x_1, x_2,...x_n is drawn from the distribution $\widetilde{X}$, then

$$E[f(\widetilde{X})] \approx \frac{1}{n} \sum_{n=1}^{n} f(x_i)$$

5.3 Monte Carlo Integration

- Usefulness of pseudorandom variables from uniform distribution on (0,1)
  - Most numerical software provides a routine for it.
  - Useful for generating random samples from other distributions.
  - Suppose $\widetilde{X}$ has a cumulative distribution function

$$F(x) = \Pr(\widetilde{X} \leq x)$$

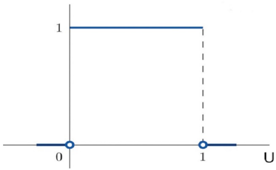
  - If $\widetilde{U}$ is uniformly distributed on (0,1), then $F^{-1}(\widetilde{U})$ has same distribution as $\widetilde{X}$.
  - Random sample x_1, x_2,...x_n are generated by

$$x_i = F^{-1}(u_i)$$

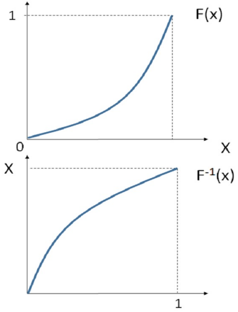
    where random sample u_1, u_2,...u_n are from uniform distribution

5.3 Monte Carlo Integration

- Usefulness of pseudorandom variables from uniform distribution on (0,1)



Uniform distribution



5.3 Monte Carlo Integration

- Pseudorandom sequences of lognormal and multivariate normal variables
  - Most numerical software have intrinsic routine that generate peudorandom standard normal variables.
  - x_j of lognormal (μ, σ^2) variates : $x_j = \exp(\mu + \sigma^2 z_j)$ where z_j is a sequence of pseudorandom standard normal variates.
  - (x_{1j}, x_{2j}) of bivariate normal random vectors with meam μ and variance matrix Σ:

$$x_i j = \mu_i + R_1 z_{1j} + R_1 z_{2j}$$

for i=1,2 where R is the Cholesky square root of Σ.

5.3 Monte Carlo Integration

- Fundamental problem
  - It is almost impossible to generate a truly random sample sequences for any distribution.
  - Computing random sample routines employ iteration rules that generate a purely diterministic sequence. (If the generator is repeated initiated at the same point, it will return same sequence.) $\triangleright$ Pseudurandom
  - Approximation will vary from one integration to the next, unless initiated at the same point. (Problematic when applied to dynamic probramming or MLE)

5.3 Monte Carlo Integration

- When do we use it?
  - Monte Carlo integration is preferred over Gaussian quadrature if the routine for computing Gaussian mass points and probabilies are not readily available or if the integration is over many dimensions.

5.4 Quasi-Monte Carlo Integration

- Quasi-Monte Carlo methods rely on sequences x_j with the property that

$$\lim_{x \to \infty} \frac{b-a}{n} \sum_{n=1}^{\infty} f(x_i) = \int_a^b f(x)dx$$

without regard to whether the sequence x_j passes standard tests of randomness.

- It can be shown that if sequences are deterministic, but attempt to fill in space in a regular manner, they can often provide more accurate approximations than random sequences do.

5.4 Quasi-Monte Carlo Integration

- Equidistributed sequences: Faure sequences
  - Let r be any prime number ($\geq$2). Any integer n has a unique expansion in terms of base r.
  - We can generate anumber in the interval [0,1) by reflecting the expansion in base r about the decimal point.
  - An example is r=3 and n=7.

$$7 = 2 \times 3^1 + 1 \times 3^0 = 21_{(3)}$$

  - When we reflect $21_{(3)}$ about the decimal point we obtain

$$\phi_3(7) = \frac{2}{3^2} + \frac{1}{3^1} = \frac{5}{9} \in [0,1)$$

$$\because \phi_3(\infty) = \frac{2}{3} + \frac{2}{3^2} + \frac{2}{3^2} ... = 1$$

$$8 = 22_3, \phi_3 8 = \frac{2}{3^2} + \frac{2}{3^1} = \frac{8}{9} \in [0,1)$$

5.4 Quasi-Monte Carlo Integration

- Equidistributed sequences: Faure sequences
  - The first 9 number (1,2,..,9) in the sequences are

$$\frac{9}{27}, \frac{18}{27}, \frac{3}{27}, \frac{12}{27}, \frac{21}{27}, \frac{6}{27}, \frac{15}{27}, \frac{24}{27}, \frac{1}{27}.$$

  - The new numbers that are added tend to fill in the gaps in the existing sequence.
  - The general expression for n in terms of the base r is

$$\mathrm{n} = \sum_{\mathrm{j}=0}^{\mathrm{m}} \mathrm{a_j(n)r^j}$$

  - The corresponding quasi-random number according to this procedure is

$$\phi_{\mathrm{r}}(\mathrm{n}) = \sum_{\mathrm{j}=0}^{\mathrm{m}} \mathrm{a_j(n)r^{-j-1}}$$

5.4 Quasi-Monte Carlo Integration

- Equidistributed sequences : Neiderreiter, Weyl, and Haber sequences
  - x_{ij} denote the jth coordinate of the ith vector in a sequence of equidistributed vectors on the d-dimentional unit hypercube.

    $$x_{ij} = \mathrm{frac}(2^{q_{ij}})$$

    where for the Neiderreiter

    $$q_{ij} = ij/(d+1)$$

    for the Weyl

    $$q_{ij} = ip_j$$

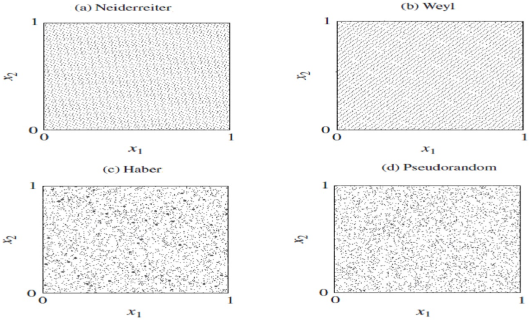
    and for Haber

    $$q_{ij} = i(i+1)p_j/2$$

  - Here, p_j is jth positive prime number, and frac(x) is x minus the greatest integer less than or equal to x.

5.4 Quasi-Monte Carlo Integration

- Equidistributed sequences : Neiderreiter, Weyl, and Haber sequences
  - Two dimentional examples - each of the plots are 4,000 values



5.4 Quasi-Monte Carlo Integration

- Equidistributed sequences : Neiderreiter, Weyl, and Haber sequences
- Example

$$\int_{-1}^{1}\int_{-1}^{1} \exp(-x_1)\cos(x_2^2)dx_1dx_2 = 4.580997$$

Table 5.2
Approximation Errors for Alternative Quasi-Monte Carlo Methods

n	Neiderreiter	Weyl	Haber	Random
1,000	0.00291	0.00210	0.05000	0.10786
10,000	0.00190	0.00030	0.01569	0.01118
100,000	0.00031	0.00009	0.00380	0.01224
1,000,000	0.00002	0.00001	0.00169	0.00197

Practical summary - Numerical integration

	Pros	Cons
Newton-Cotes	Simple and robust	How many partition points are needed?
Gaussian Quadrature	Direct application to probability density function	1. Not efficient for non-smooth integrands. 2. Computation is hard except that well-known probability function is applied.
Monte Carlo	Simple scheme (LLN)	Pseudo random number issue - accuracy problem
Quasi-Monte Carlo	More accurate than Monte Carlo	

5.6. Numerical Differentiation

$$f'(x) = \lim_{h \to 0}$$

- Taylor expansoin

  $$f(x+h) = f(x) + f'(x)h + O(h^2)$$

  where $O(h^2)$ means that other terms are expressible in terms of second or higher powers of h.

  $$f'(x) = \mathrm{fracf}(x+h) - f(x)h + O(h)$$

- (Since $O(h^2)/h = O(h)$), so the approximation for the derivative f'(x) has an $O(h)$ error.

5.6. Numerical Differentiation

- A more accurate finite difference approximation to the f'(x)?
  - Two second-order Taylor expansions

$$f(x+h) = f(x) + f'(x)h + f''(x)\frac{h^2}{2} + O(h^3)$$

$$f(x-h) = f(x) - f'(x)h + f''(x)\frac{h^2}{2} + O(h^3)$$

  Subtract the second expression from the first, and rearrange,

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

  - The centered finite difference approximation to the f'(x)
  - One order more accurate than the previous one-sided finite difference approximation.

5.6. Numerical Differentiation

- Three-point approximation

$$f'(x) \approx af(x) + bf(x+h) + cf(x+\lambda h)$$

  - Taylor expanstion up to second order and a simple trick (see Miranda and Fackler in detail) gives

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \frac{1}{h\lambda(1-\lambda)} \begin{pmatrix} \lambda^2 - 1 \\ -\lambda^2 \\ 1 \end{pmatrix}$$

    and results in

$$af(x) + bf(x+h) + cf(x+\lambda h) = f'(x) + O(h^2).$$

  - When λ=-1, it returns to approximation in the centered finite difference approximation.

5.6. Numerical Differentiation

- Three-point approximation
  - When λ=2, it is useful when a derivative is needed at a boundary of a domain.

$$f'(x) = \frac{1}{2h}[-3f(x) + 4f(x+h) - f(x+2h)] + O(h^2)$$

  - Use h>0 for a lower bound and h<0 for an upper bound
- Derivation of second derivative approximation uses similar methods. (See book for detail)