Numerical Methods

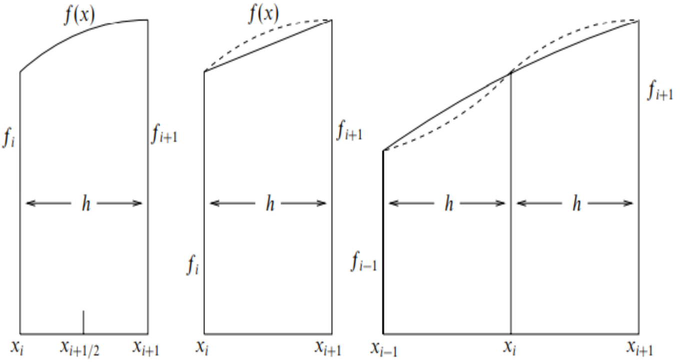
IIIT, Allahabad

Instructor: Srijit Bhattacharjee

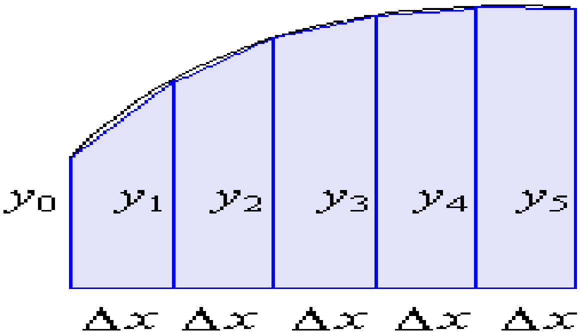
Numerical Integration

- Newton-Cotes method
  equispaced nodes
- Gauss-quadrature method
  non-equispaced nodes
- Monte Carlo method
  nodes are randomly distributed

NI methods



Trapezoidal method



- $I = \int_a^b f(x)dx$
- Let the interval $[a, b]$ be divided into n equal spaced intervals.
- $x_0 = a, x_1 = a + h, \cdots x_n = a + nh = b. \quad h = \frac{b-a}{n}$.
- $x_i = x_0 + ih,\ i = 1, 2, 3, \cdots.\ f_i \equiv y_i$
- $\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx \cdots + \int_{x_{n-1}}^{x_n} f(x)dx$

$$\sim \frac{h}{2}\left[(y_0 + y_1) + (y_1 + y_2) + \cdots + (y_{n-1} + y_n)\right]$$

$$I = \frac{h}{2}\left[y_0 + 2(y_1 + y_2 + \cdots y_{n-1}) + y_n\right]$$

Weights: Counts the number of times a nodal value of the function appears. Here

$$w_i = h(1/2, 1, 1, \cdots, 1/2)$$

Error in Trapezoidal method

- Error (of a cell)=Difference between exact integration of one cell and the area of the trapezium of the same cell.
- Let $\int f(x)dx = F(x);\ F'(x) = f(x)$ in $[x_0, x_1]$.
- Then $\int_{x_0}^{x_1} f(x)dx = F(x_1) - F(x_0)$

$$\int_{x_0}^{x_1} f(x)dx = F(x_0+h) - F(x_0)$$

$$= F(x_0) + hF'(x_0) + \frac{h^2}{2!}F''(x_0) + \frac{h^3}{3!}F'''(x_0) + \cdots - F(x_0)$$

$$= hy_0 + \frac{h^2}{2!}y_0' + \frac{h^3}{3!}y_0'' \cdots = \Delta A_{exct}$$

- Estimated error

$$\Delta A_{est} = \frac{h}{2}(y_0 + y_1)$$

$$= \frac{h}{2}\left[y_0 + y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' \cdots\right]$$

Total Error

- Error for a cell

$$E_i = \Delta A_{exct} - \Delta A_{est} = -\frac{h^3}{12} y_0^{''}(\xi) + O(h^4)$$

- Total error:

$$\sum_i E_i = E = -\frac{h^3}{12}(y_0^{''} + y_1^{''} + \cdots + y_{n-1}^{''}) + O(h^4)$$

$$E \simeq -n\frac{h^3}{12} y^{''}(\xi) = -\frac{(b-a)}{12} h^2 f^{''}(\xi)$$

- where $f^{''}(\xi)$ is the highest among the $f_k^{''}$, $k = 0, 1, 2, \cdots n-1$.

Simpson's $\frac{1}{3}$rd rule

Instead of linear function (in Trapezium rule), approximate the function or curve in a cell by a quadratic polynomial.
In the neighborhood of x_i for i odd, one can write

$$f(x_i + y) = f_i + ay + by^2$$

Next take $y = \pm h$ and get

$$\begin{aligned} f_{i+1} = f(x_i + h) &= f_i + ha + bh^2 \\ f_{i-1} = f(x_i - h) &= f_i - ah + bh^2 \\ \implies bh^2 = \frac{h}{2}(y_{i+1} - 2y_i + y_{i-1}) \end{aligned}$$

Simpson's $\frac{1}{3}$rd rule

Area (estimated)$= \int_{-h}^{h}(f_i + ay + by^2)dy = 2hf_i + \frac{2}{3}bh^3$

Replacing bh^2 yields: $A_i = \frac{h}{3}(f_{i+1} + f_{i-1} + 4f_i)$

$$A = \frac{h}{3}(f_0 + 4\sum_{i=odd} f_i + 2\sum_{i=even} f_i + f_n)$$

For the first cell the exact area is $\Delta A_{exct} = \int(f(x_0+2h) - f(x_0))dx$
$= 2hf(x_0) + 2h^2f^{'}(x_0) + \frac{4}{3}h^3f^{''}(x_0) + \frac{2}{3}h^4f^{'''}(x_0) + \frac{4}{15}h^5f^{''''}(x_0) + \cdots$

Area estimated: $\Delta A_{est} = \frac{h}{3}(f(x_0) + f(x_0+2h) + 4f(x_0+h))$

This yields:
$\Delta A_{est} = 2hf(x_0)+2h^2f'(x_0)+\frac{4}{3}h^3f^{''}(x_0)+\frac{2}{3}h^4f^{'''}(x_0)+\frac{5}{18}h^5f^{''''}(x_0)+\cdots.$

Errors

$E_i = -\frac{1}{90} h^5 f^{''''}(\xi) + O(h^6).$

The total error: $E = -\frac{1}{90}\frac{n}{2} h^5 f^{''''}(\xi) = -\frac{b-a}{180} h^4 f^{''''}(\xi).$

Simpson 3/8 rule:

$$I = \frac{3h}{8}\left[y_0 + 3\sum_{i\neq 3j}^{n-1} f(x_j) + 2\sum_{i=1}^{n/3-1} f(x_{3i}) + f(x_n)) \right],\ j \in N_0$$

Error for Simpson's 3/8 rule is $E = -\frac{3}{80} n h^5 f^{''''}(\xi)$

Error for Boole's method is $O(h^7)$.

Home work: Do this integral in different methods and compare

$$I = \int_0^1 \frac{1}{1+x^2} dx$$

What we have done?

Newton-Cotes integration formulae: integration points are equally spaced. Weighting factors are same or similar.

We replaced the integrand $f(x)$ by a polynomial

$$I = \int_a^b f(x)dx \simeq \int_a^b f_n(x)dx$$

with, $f_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + f_n(x)x^n$.

Simpson's 1/3rd rule is obtained by replacing a second order Lagrange's interpolating polynomial:

$$I = \int_{x_0}^{x_2} \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_1)(x-x_0)}{(x_2-x_1)(x_2-x_0)} f(x_2)$$

Gaussian quadrature method

Methods of numerical integration, in which the integration points are not equally spaced and the weightings given to the values at each point do not fall into a few simple groups are fall under the class of Gaussian quadrature method.

The node points are unknown

We use Orthonormal Polynomials, whose zeroes are the nodal points of an integration.

Eg: Legendre-Gauss method: Here one uses Legendre polynomials $P_l(x)$.

$\int_a^b f(x)dx \equiv \int_{-1}^{1} \frac{b-a}{2} g(z)dz$, with $z = \frac{2x-b-a}{b-a}$.

x_is are solutions of $P_l(x_i) = 0$.

Legendre-Gaussian quadrature method

Using the properties of $P_l(x)$ and their derivatives one may write

$$\int_{-1}^{1} g(z)dz \simeq \sum_i w_i g(z_i)$$

with

$$w_i = \frac{2}{(1-x_i^2)|P_l'(z_i)|^2}$$

Monte Carlo method

Handy for integrals with complicated boundaries. Overperforms the other methods for multidimensional case.

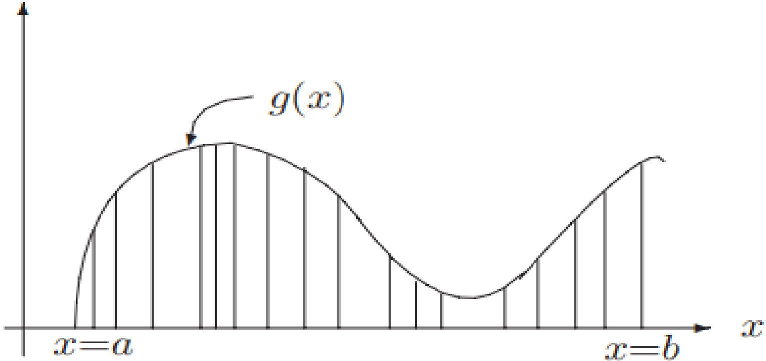
Depends on random numbers. Discretize the interval in unequal lengths generated randomly. Weights also can be unequal.

In the simplest of the cases, random numbers within the range $[a, b]$ with Uniform distribution is used.

Let us consider

$$I = \int_a^b g(x)dx$$

MC method: diagram



Recall the Uniform p.d.f.

$$\begin{aligned} f(x) &= \frac{1}{b-a}; \qquad a < x < b \\ &= 0 \qquad \textit{otherwise} \end{aligned}$$

Inserting $f(x)$ into I: $I = (b-a)\int_a^b f(x)g(x)dx$

Of course $I = (b-a)E[g(x)]$

Generate random sample points x_i with the pdf $f(x)$. Evaluate $g(x_i)$ for each x_i and estimate:

$$G/\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

Obviously, for sufficiently large number of points $\bar{G} = \bar{g} = E[g(x)]$.

$$I \simeq (b-a)\bar{G} \simeq \frac{b-a}{N}\sum_{i=1}^{N} g(x_i)$$

More on MC method

The bias in this method can be estimated by computing $E[G] - I$

One needs to keep the variance of the estimate I to be small.

$$\sigma_G^2 = \frac{1}{N}\sum_i^N (g(x_i) - \bar{g})^2 f(x_i) = \frac{1}{N}\sigma_g^2$$

Variance of the sample mean is $1/N$ times of the variance of the population mean. The error committed to estimate the integral I with N sample points is getting reduced by a factor of $\frac{1}{\sqrt{N}}$. (CLT)

Error in MC method

Newton-Cotes methods had an error $\sim O(h^k)$, where h is step size. In terms of number of step size n this is $\sim n^{-k}$, $k \geq 1$

If our integration is multidimensional, lets say a hypercube of length L and dimension d. Number of points $\sim (L/h)^d$. Then N-C methods give an accuracy $N^{-k/d}$, as $N = n^d$ iterations are involved.

MC integration is thus is useful for higher dimensions as Error in MC method always scales as $\frac{1}{\sqrt{N}}$

MC integration is more efficient than an order-k algorithm when $d > 2k$.

Random number generators

- Generating pseudo random numbers: $x_{n+1} = (ax_n) mod\ m$, $a < m$
- x_{n+1} takes values $0, 1, 2, \cdots m - 1$.
- The period or length of random numbers depends on values of a, x_0 and m.

Inverse transform sampling

- One can also sample the nodes using other pdfs.

$$\int_a^b p(x)f(x)dx$$

- This can be acheived by a variable transformation $du = p(x)dx$.

$$\int_a^b p(x)f(x)dx = \int_0^{b-a} f(x(u))du$$

- The unweighted version is restored. Now do the similar procedure introducing Uniform distribution

$$\int_a^b p(x)f(x)dx \simeq \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

Importance sampling

- Consider $p(x)$ is a pdf whose feature resembles that of a function G in $[a, b]$. Therefore
$$\int_a^b p(x)dx = 1$$
- Also $\int_a^b G(x)dx = \int_a^b p(x)\frac{G(x)}{p(x)}dx$.
- Ex: if $p(x)$ is Uniform in $x \in [a, b]$ then $p(x)dx = \frac{dx}{b-a} = p(y)dy$ and
$$y(x) = \int_a^x p(x')dx' \implies y = \frac{x-a}{b-a}$$
- Therefore if y is invertible,
$$\int_a^b p(x)\frac{G(x)}{p(x)}dx = \int_0^{b-a} \frac{G(x(y))}{p(x(y))}dy = \frac{b-a}{N}\sum_{i=1}^{N}\frac{G(x(y_i))}{p(y(x_i))}$$

Solving Ordinary Differential Equations

- An ODE is in general has the following form

$$f((y(x), y'(x), y"(x), \cdots, x) = 0$$

- For first order ODE Euler method provides a numerical or approximate solution.
- Solve $\dfrac{dy}{dx} = f(x, y)$ with initial condition: $y(x_0) = y_0$.
- Taylor expanding y wrt x_0 and keeping only first order terms in step size h:

$$y(x_0 + h) = y(x_1) = y_1 = y_0 + hf(x_0, y_0)$$

- Next

$$y_2 = y_1 + hf(x_1 y_1)$$
$$y_3 = y_2 + hf(x_2, y_2) \cdots$$

Euler Method

- General: $y_{n+1} = y_n + hf(x_n, y_n),\ n = 0, 1, 2, \cdots$
- Problems: Too slow to converge. Error is $O(h^2)$.
  No scope to improve the value of y.
- In modified Euler's method one uses

$$y_1 = y_0 + \int_{x_i}^{x_{i+1}} f(x, y)dx$$

  and the integration is done using Trapezoidal method.

$$y_1 = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, y(x_1))]$$

- $y(x_1)$ is obtained using Euler's method. $y_1 = y_0 + hf(x_0, y_0)$, and the first approximation of y_1 becomes

$$y_1^{(1)} = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, y_1^{(0)}))]$$

Euler Method

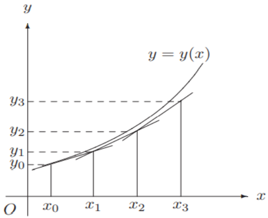
- In a generic step

$$y_{i+1}^{(k+1)} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_i, y_i^{(k)}))], k = 0, 1, 2, \cdots i = 0, 1, 2, \cdots$$

- The integration $\int_{x_i}^{x_{i+1}} f(x, y)$ can also be done by Monte Carlo method when f is independent of y.
- One can obtain the iteration formula using MC estimator

$$y(x_i) = y(x_{i-1}) + \frac{x_i - x_{i-1}}{N} \sum_{j=1}^{N} f(x_j)$$

EM and Modified EM



E = Euler Method
M E = Modifed Euler Method

x	E h = 0.1	E h = 0.05	M E h = 0.1	M E h = 0.05	Exact
0.000	1.000000000	1.000000000	1.000000000	1.000000000	1.000000000
0.100	0.800000000	0.810113105	0.820409365	0.819331760	0.819003663
0.200	0.640818731	0.657393464	0.674403812	0.672642391	0.672107566
0.300	0.515336265	0.535590690	0.556553290	0.554402342	0.553750941
0.400	0.417208317	0.439092731	0.461944052	0.459617490	0.458914649
0.500	0.340955917	0.363016799	0.386268321	0.383916533	0.383207751
0.600	0.281961719	0.303216926	0.325840313	0.323564517	0.322880195
0.700	0.236412367	0.256242290	0.277567879	0.275432049	0.274791217
0.800	0.201213145	0.219269254	0.238899650	0.236940201	0.236353524
0.900	0.173891893	0.190022158	0.207760869	0.205994363	0.205466518
1.000	0.152502724	0.166692376	0.182486361	0.180915522	0.180447044

Examples

- Find the solution of the equation for $x = 0.5$,

$$y' = x^2 - y, \; y(0) = 1$$

- ODEs are very common in Engineering and Science. Example bacterial growth equation:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right), \; N(0) = N_0$$

System of ODEs

- Evaluation of values of dependent variables y, for more than one ODEs can also be obtained by the methods discussed.
- Enzyme reactions:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} SE \xrightarrow{k_2} P + E$$

$$\frac{ds}{dt} = -k_1 es + k_{-1} c, \quad \frac{de}{dt} = -k_1 es + (k_{-1} + k_2)c$$
$$\frac{dc}{dt} = k_1 es - (k_{-1} + k_2)c, \quad \frac{dp}{dt} = k_2 c.$$

- One can write the system of equations symbolically as

$$y_i' = f_i(t, y_1, y_2, \cdots)$$

- The Euler's method can be written as

$$y_{i,j+1} = y_{i,j} + hf_i(t, y_{1j}, y_{2j}, \cdots y_{mj}); m = 1, 2, \cdots 4; j = 0, 1, 2, \cdots N$$

Predictor-Corrector Method

- Equation: $y' = f(x,y), y(0) = y_0$
- Value of y for forward difference

$$y_{i+1} = y_i + hf(x_i, y_i),\ \ \textit{Prediction}$$

.

- If one considers a parabolic curve that describes $y(x)$ between x_{i-1} and x_{i+1}, we have

$$y' = f(x,y) \simeq a + b(x - x_i)$$

$a = f_i,\ \ b = \frac{f_i - f_{i-1}}{h}$

- This yields: $y_i - y_{i-1} \sim \int_{x_i}^{x_{i+h}} \left[f_i + \frac{f_i - f_{i-1}}{h}(x - x_i) dx \right]$
- $\implies y_{i=1} = y_i + hf_i + \frac{h}{2}(f_i - f_{i-1})$ correction

Predictor-Corrector Method

Steps for P-C method

- First predict the value of y by$y_{i+1} = y_i + hf_i$
- Calculate $f_i + 1$ using the above relation
- Next, correct the value using $y_{i+1} = y_i + h(f_i + f_{i+1})/2$.