Numerical Integration

Numerical Integration: 1-D

✓ **Integrate a real-valued function: f(x)**

- Riemann integral
- Algebraic summation of area under the function f(x)
- Integral is the **limit** of a summation (series)
  - **Riemann sum**

Some number in the Δx_i interval

$$\underbrace{\int_a^b f(x)\,dx}_{\text{Riemann integral}} = \lim_{N\to\infty} \underbrace{\sum_{i=0}^{N} f(x_i)\Delta x_i}_{\text{Riemann sum} \rightarrow \text{series}}$$

- Convergence of the series requires that $\lim_{m\to\infty} f(x_m)\Delta x_m = 0$
  - Therefore:

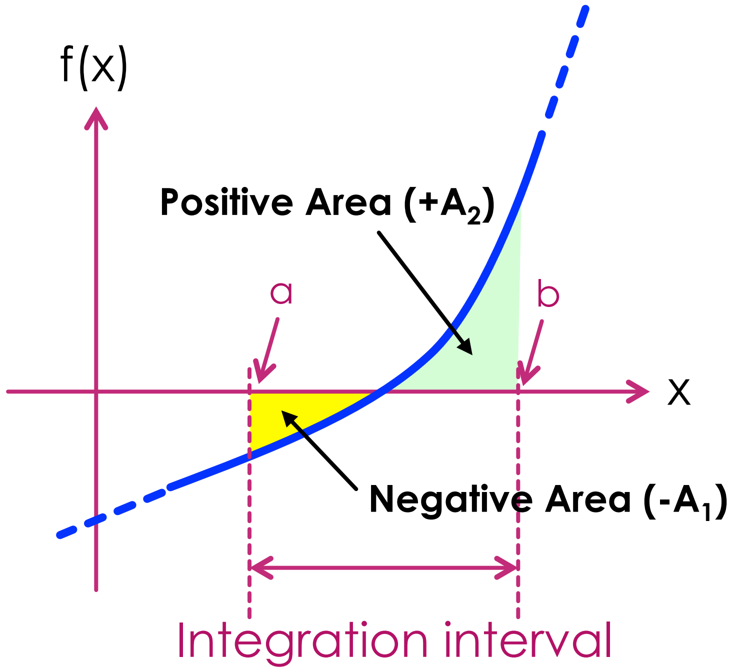
$$\lim_{m\to\infty} \Delta x_m = 0$$

(what is Δx ?)

- How to find the area under a given function?

Visualization

$$\int_a^b f(x)\,dx = A_2 - A_1$$

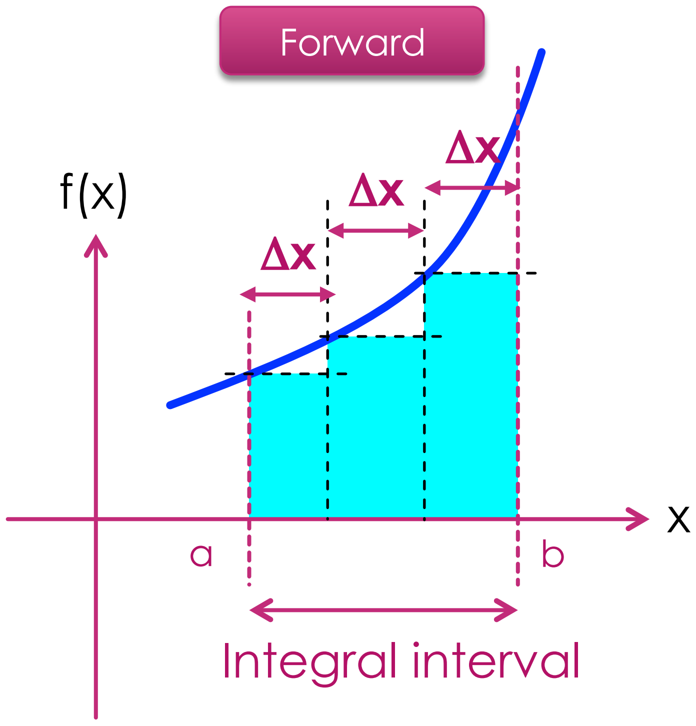


Numerical Integration: 1-D

✓ **Finding the area underneath a function**

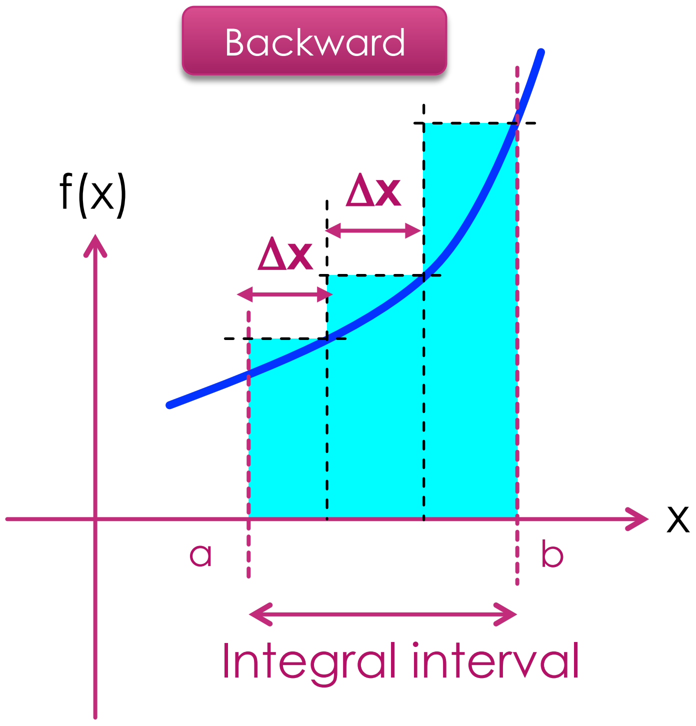
- **Rectangle estimation**
  - Uniform Partitioning

Theorem: Riemann summation will converge to the same result for any arbitrary partitioning.



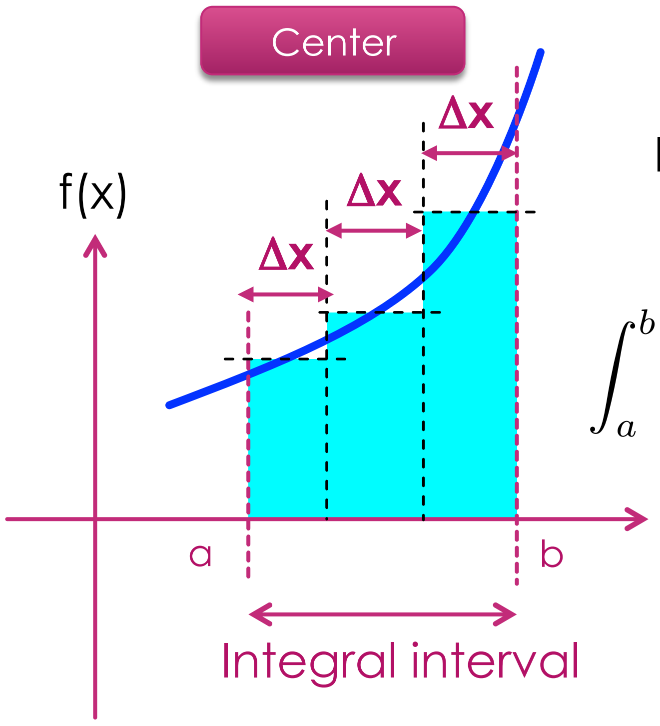
f(x) at the beginning of partition

Area = $f(x_i)\ \Delta x$



f(x) at the end of partition

Area = $f(x_{i+1})\ \Delta x$



f(x) at the center of partition

Area = $f((x_i+x_{i+1})/2)\ \Delta x$

Integral of a constant function?

$$\int_a^b c\,dx = c(b-a)$$

(exact)

Numerical Integration: 1-D

- **Difference summations**
  - Fundamental property

Last index

$$\sum_{k=m}^{n} (a_{k+1} - a_k) = a_{n+1} - a_m$$

First index

  - Useful sequence sums

Constant

$$\sum_{k=m}^{n} c = c \times \underbrace{(n-m+1)}_{\text{Number of terms}} \sim n$$

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6} \sim \frac{n^3}{3}$$

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \sim \frac{n^2}{2}$$

$$\sum_{k=1}^{n} k^3 = \left(\frac{n(n+1)}{2}\right)^2 \sim \frac{n^4}{4}$$

Index can start from 0 as well

Asymptotic approximations for n → ∞

How to prove these?

$$\boxed{\sum_{k=1}^{n} k^p \Rightarrow a_k = k^{p+1}}$$

For p = 1, 2, 3, …

+

Recursion

Numerical Integration: 1-D

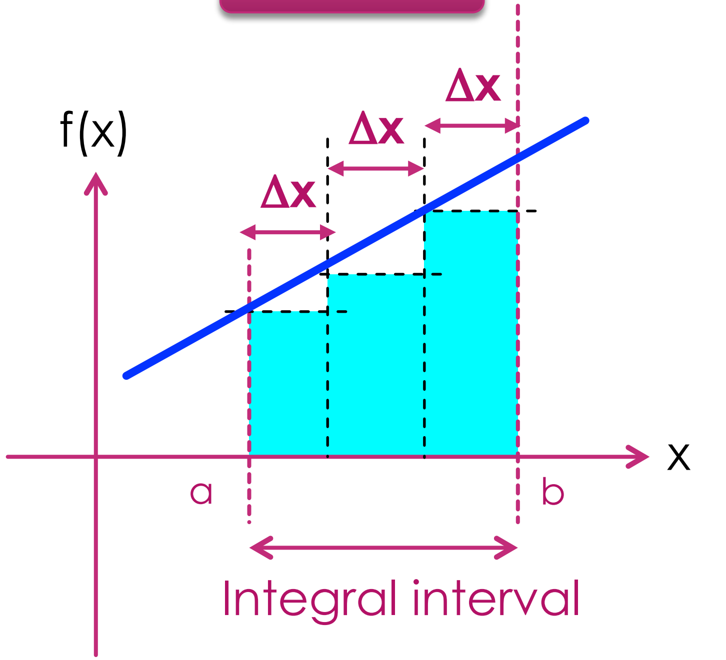
✓ **Finding the area underneath a function**

- **Rectangle estimation**
  - Uniform Partitioning

Break point

$$\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx$$

Forward



f(x) at the beginning of partition

Area = f(x_i) Δx

Example: integral of a linear function f(x) = x → $\Delta x = \dfrac{b-a}{n}$

$$\int_a^b f(x)\,dx = \lim_{n\to\infty} \sum_{i=0}^{n-1} f(x_i)\Delta x \rightarrow \quad x_i = a + i \times \Delta x = a + (b-a)\frac{i}{n} \quad \text{for } i = 0, 1, 2, \ldots$$

$$\sum_{i=0}^{n-1} f(x_i)\Delta x = \sum_{i=0}^{n-1} \left(\frac{a(b-a)}{n} + \frac{(b-a)^2}{n^2} i \right) = a(b-a) + \frac{(b-a)^2}{n^2} \times \frac{(n-1)n}{2} \xrightarrow{\textbf{Limit}}$$

$$\int_a^b f(x)\,dx = a(b-a) + \frac{(b-a)^2}{2} = \frac{b^2-a^2}{2} \longrightarrow \boxed{\int_a^b x\,dx = \left.\frac{x^2}{2}\right|_a^b}$$

Anti-derivative

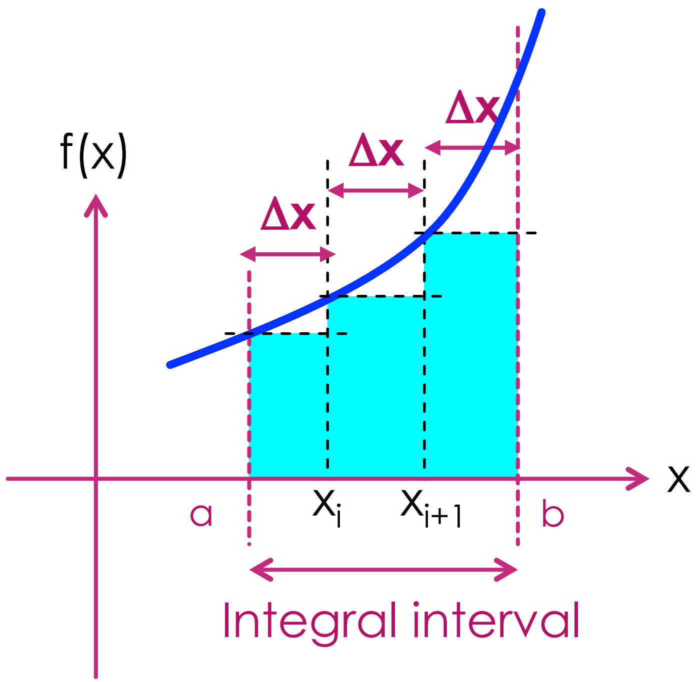
$$\text{Error} = \frac{b-a}{2} \times \Delta$$

Linearity property: $\displaystyle\int_a^b (f(x) + \lambda g(x))\,dx = \int_a^b f(x)\,dx + \lambda \int_a^b g(x)\,dx$

Numerical Integration: 1-D

✓ **Finding the area underneath a function**

- **Rectangle estimation**
  - Uniform Partitioning (N partitions)



Example: integral of a linear function f(x) = x^2 → $\Delta x = \frac{b-a}{n}$

$$x_i = a + i \times \Delta x = a + (b-a)\frac{i}{n} \quad \text{for } i = 0, 1, 2, \ldots$$

$$\sum_{i=0}^{n-1} f(x_i)\Delta x = \Delta \sum_{i=0}^{n-1} (a^2 + \Delta^2 i^2 + 2\Delta \times a\, i)$$

Note that Δ ~ 1/n

$$n \qquad \frac{(n-1)n(2n-1)}{6} \qquad \frac{(n-1)n}{2}$$

Taking the limit n → ∞

$$\int_a^b x^2\,dx = a^2(b-a) + (b-a)^3\frac{1}{3} + (b-a)^2 a \longrightarrow \boxed{\int_a^b x^2\,dx = \frac{x^3}{3}\bigg|_a^b}$$

$$= \frac{b^3 - a^3}{3}$$

Anti-derivative

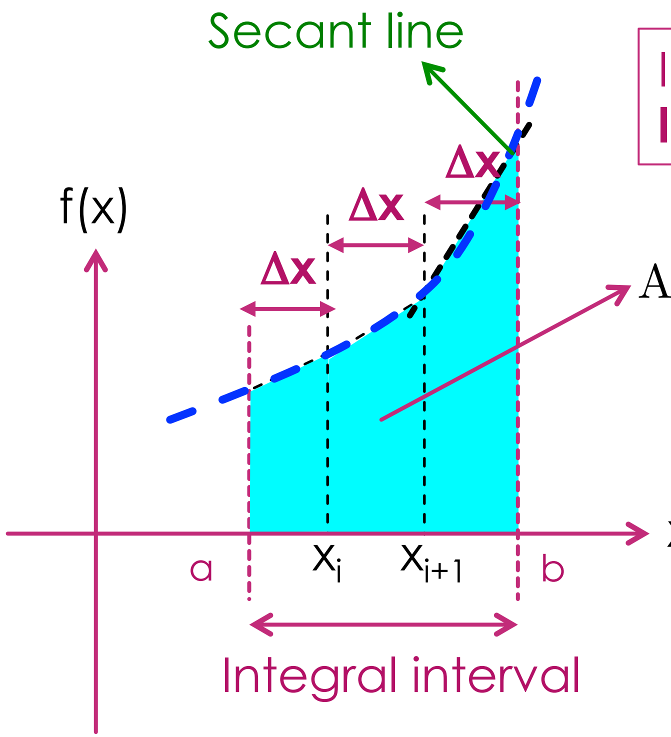
Numerical Integration: 1-D

- ✓ **Finding the area underneath a function: f(x)**
  - **Rectangular estimation**
    - Uniform Partitioning
- ✓ How to implement in Java:
  - **Step 1**: Define an interface to enforce real variable and real function
    - IntegralFunction1D
    - Even better: use the built-in **Function<T_1,T_2>** interface
  - **Step 2**: Define a class that performs summation (Integral1D)
    - Stop criteria
      - Could be number of terms
      - Could be convergence of the series
        - Absolute error
        - Relative error

Numerical Integration: 1-D

✓ **Finding the area underneath a function**

- **Trapezoid's method:** gives <u>exact</u> result for a_1x+a_0 (linear) polynomials
  - Uniform Partitioning (N partitions)



In each partition, considers a **linear** approximation of f(x)

$$\text{Area} = \frac{f(x_i) + f(x_{i+1})}{2} \Delta x$$

f(x) at the beginning and end of partition

Error Estimates

Rectangular: Error ~ Δx $\sim \frac{1}{N}$

Trapezoid: Error ~ Δx^2 $\sim \frac{1}{N^2}$

One order of magnitude

Rectangular → Δx reduced by 10 → Error x 0.1

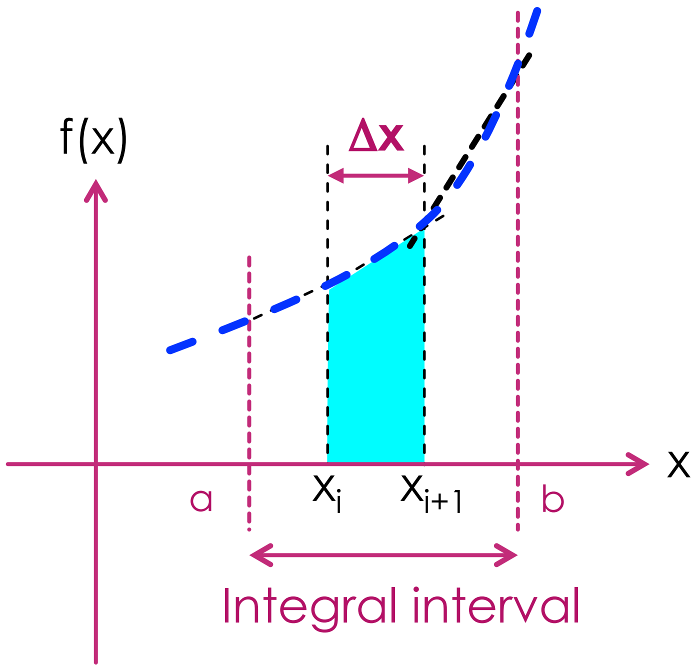
Center → Error ~ Δ^2

Two orders of magnitude

Trapezoid → Δx reduced by 10 → Error x 0.01

Polynomial Integration

- Two approaches
  - **Approach 1:** use standard definition and calculate the limit of sum
  - **Approach 2:** use approximating functions



Approach 1 → $$\int_a^b f(x)\,dx = \lim_{N\to\infty}\sum_{i=0}^{N} f(x_i)\Delta x_i$$

Approach 2 → $$\int_a^b f(x)\,dx = \lim_{N\to\infty}\sum_{i=0}^{N} \underbrace{\int_{x_i}^{x_{i+1}} f_a(x)\,dx}_{\text{Exact integral of the approximating function}}$$

Approximating function

Polynomial Rule → $$\boxed{\int_a^b x^m\,dx = \frac{1}{m+1}x^{m+1}\Big|_a^b}$$

Simpson's Method

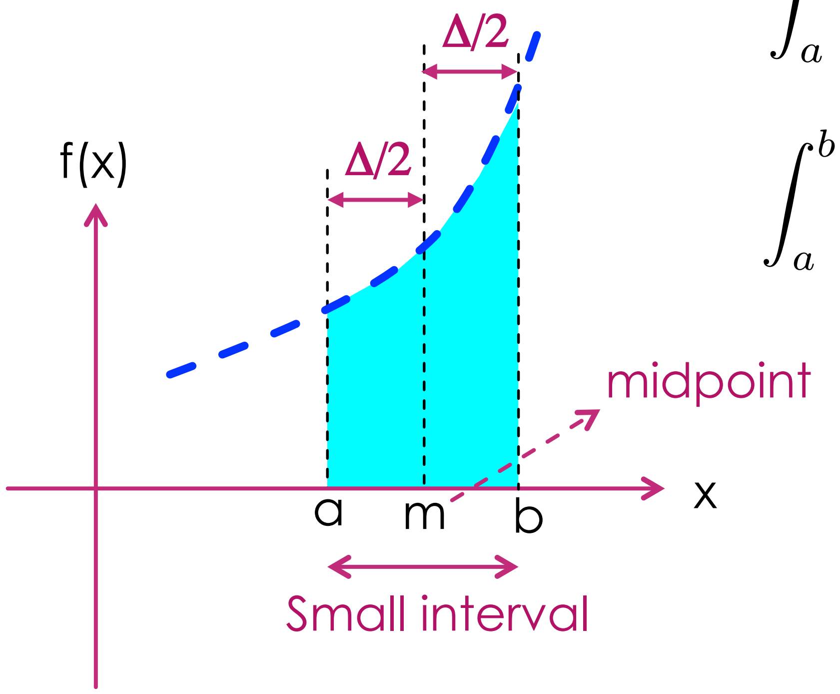
✓ **Finding the area underneath a function**

- For each partition, considers a **quadratic** approximation of f(x)
- Gives exact result for $a_2x^2 + a_1x + a_0$ (quadratic) polynomials
- Smaller error order than trapezoid

Approximating quadratic polynomial

$$\int_a^b f(x)\,dx \approx \int_a^b f_{app}(x)\,dx$$

→ Three points are needed for quadratic fitting: start, end, midpoint



$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left(f(a) + 4f(m) + f(b)\right)$$

Error → Δx^4

Lagrange's interpolating polynomial

$$p(x) = \underbrace{\frac{(x-m)(x-b)}{(a-m)(a-b)}}_{\text{Quadratic polynomial}} f(a) + \frac{(x-a)(x-b)}{(m-a)(m-b)} f(m) + \frac{(x-a)(x-m)}{(b-a)(b-m)} f(b)$$

Recursive Integration

✓ **Finding the area underneath a function**

- We can use recursion
  - Implicit integration
- Make use of the break-point property of integral
  - In each step, divide the interval by 2 or 3 or …
- **Stop condition**
  - What to do when the interval is very small
    - Constant approximation → rectangle method
    - Linear approximation → trapezoid method
    - Quadratic approximation → Simpson method

Break point

$$\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx$$

Recursion: a problem can be divided into two smaller similar problems

Tip: You can use the break-point rule for more than one break points.

Newton-Cotes Rules

- What if we only know the values of the function at certain points (N+1 points)?
- **Newton-Cotes**
  - Interpolate the polynomial with Lagrange's interpolation and calculate the integral
    - N intervals → N+1 points → degree = N
    - Mostly used for equally-spaced points (**uniform** partitioning)

$$f(x) \approx \sum_{i=0}^{n-1} f(x_i) l_i(x) \Rightarrow \int_a^b f(x)\,dx \approx \sum_{i=0}^{n-1} f(x_i) \int_a^b l_i(x) \Rightarrow \int_a^b f(x)\,dx \approx \sum_{i=0}^{n-1} w_i\, f(x_i)$$

Lagrange's term

Newton-Cotes **weights**

Degree n	Step size h	Common name	Formula
1	$b-a$	Trapezoid rule	$\frac{h}{2}(f_0+f_1)$
2	$\frac{b-a}{2}$	Simpson's rule	$\frac{h}{3}(f_0+4f_1+f_2)$
3	$\frac{b-a}{3}$	Simpson's 3/8 rule	$\frac{3h}{8}(f_0+3f_1+3f_2+f_3)$
4	$\frac{b-a}{4}$	Boole's rule	$\frac{2h}{45}(7f_0+32f_1+12f_2+32f_3+7f_4)$

We can apply Newton-Cotes rule to each individual sub-interval

$$\int_a^b f(x)\,dx = \lim_{N\to\infty} \sum_{i=0}^{N} \int_{x_i}^{x_{i+1}} f_a(x)\,dx$$

Gaussian Quadrature Method

- Generalizing numerical approximation of an integral
  - We see a recurring theme in the previous methods
    - Rectangle method: **exact** for polynomial $P(x) = a_0$ → Error ~ Δ
    - Trapezoid method: **exact** for polynomial $P(x) = a_1x + a_0$ → Error ~ Δ^2
    - Simpson method: **exact** for polynomial $P(x) = a_2x^2 + a_1x + a_0$ → Error ~ Δ^4
- **Condition**: approximation MUST be EXACT for polynomials of $x^0, x^1, x^2, \ldots, x^{2n-1}$
  - Find the quadrature points and their weights: **2n** unknown parameters
  - Gauss-Legendre quadrature rule: integration interval is **[-1, 1]**

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

Gauss-Legendre Rule → $$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i\, f(x_i)$$

Linear mapping of intervals [-1,1]→[a,b] $$x \to \frac{b-a}{2}x + \frac{b+a}{2}$$

$$\boxed{\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{b+a}{2}\right)dx}$$

g(x)

Gaussian Quadrature Method

- **Condition**: approximation MUST be EXACT for polynomials of $x^0, x^1, x^2, \ldots, x^{2n-1}$
  - Find the quadrature points and their weights: **2n** unknown parameters
  - Gauss-Legendre quadrature rule: integration interval is **[-1, 1]**

Gauss-Legendre Rule → $\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i\, f(x_i)$

$$\int_{-1}^{1} x^0\, dx = \sum_{i=1}^{n} w_i x_i^0 = \sum_{i=1}^{n} w_i$$

$$\int_{-1}^{1} x^1\, dx = \sum_{i=1}^{n} w_i x_i$$

$$\int_{-1}^{1} x^2\, dx = \sum_{i=1}^{n} w_i x_i^2$$

$$\vdots$$

$$\int_{-1}^{1} x^{2n-1}\, dx = \sum_{i=1}^{n} w_i\, x_i^{2n-1}$$

Must solve this system of equations

$$\boxed{w_i = \frac{2}{(1-x_i^2)[P_n'(x_i)]^2}}$$

Legendre Polynomial

$P_n(1) = 1$

x_i is the i^{th} root of $P_n(x)$

All roots are in [-1,1]

from Wikipedia

Number of points, n	Points, x_i	Approximately, x_i	Weights, w_i	Approximately, w_i
1	0	0	2	2
2	$\pm\frac{1}{\sqrt{3}}$	±0.57735	1	1
3	0	0	$\frac{8}{9}$	0.888889
	$\pm\sqrt{\frac{3}{5}}$	±0.774597	$\frac{5}{9}$	0.555556
4	$\pm\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$	±0.339981	$\frac{18+\sqrt{30}}{36}$	0.652145
	$\pm\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$	±0.861136	$\frac{18-\sqrt{30}}{36}$	0.347855
5	0	0	$\frac{128}{225}$	0.568889
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$	±0.538469	$\frac{322+13\sqrt{70}}{900}$	0.478629
	$\pm\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$	±0.90618	$\frac{322-13\sqrt{70}}{900}$	0.236927

Gaussian Quadrature Method

- Legendre Polynomial are orthogonal polynomials. Only defined over **[-1,1]** interval.

$$\int_{-1}^{1} p_m(x)\, p_n(x)\, dx = 0 \quad \text{if } n \neq m$$

- Definition via generating function

$$\frac{1}{\sqrt{1-2xt+t^2}} = \sum_{n=0}^{\infty} p_n(x)\, t^n$$

- Definition via Bonnet's recursion formula

$$(n+1)p_{n+1}(x) = (2n+1)x\, p_n(x) - n\, p_{n-1}(x) \quad \text{for} \quad p_0(x) = 1 \quad , \quad p_1(x) = x$$

- Definition via Rodrigues' formula

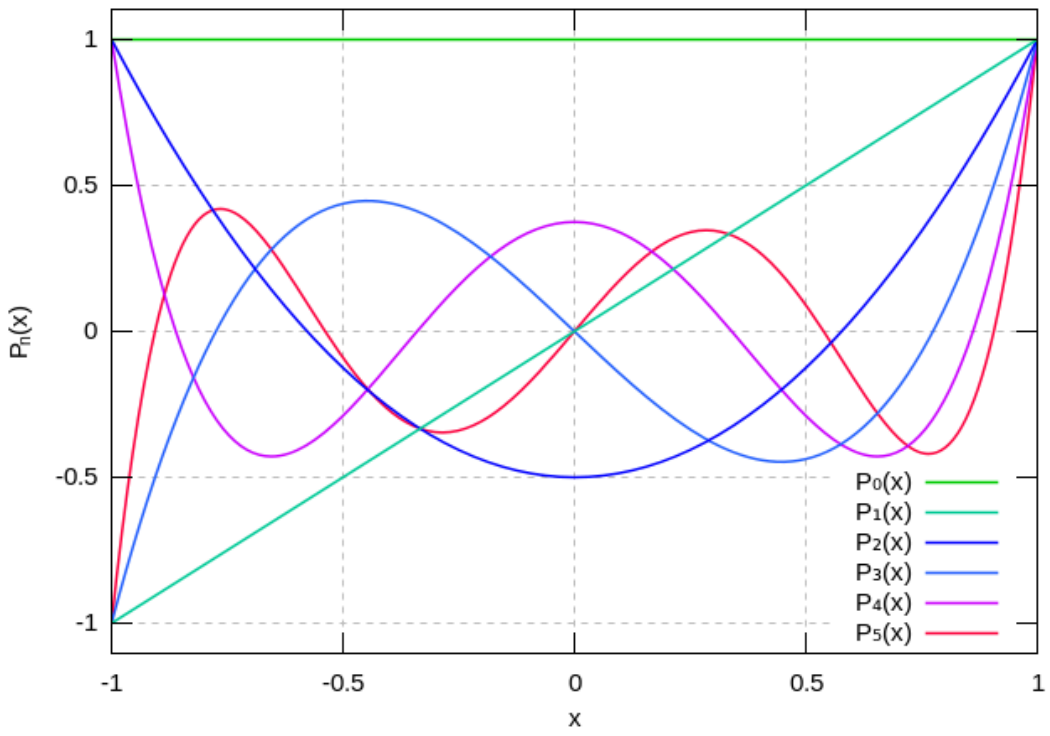
$$p_n(x) = \frac{1}{2^n\, n!} \frac{d^n}{dx^n} (x^2-1)^n$$

$$\boxed{p_n(-x) = (-1)^n\, p_n(x)}$$

Legendre Polynomials

- Definition via Rodrigues' formula

$$p_n(x) = \frac{1}{2^n\, n!} \frac{d^n}{dx^n} (x^2 - 1)^n$$



n	$P_n(x)$
0	1
1	x
2	$\frac{1}{2}(3x^2-1)$
3	$\frac{1}{2}(5x^3-3x)$
4	$\frac{1}{8}(35x^4-30x^2+3)$
5	$\frac{1}{8}(63x^5-70x^3+15x)$
6	$\frac{1}{16}(231x^6-315x^4+105x^2-5)$
7	$\frac{1}{16}(429x^7-693x^5+315x^3-35x)$
8	$\frac{1}{128}(6435x^8-12012x^6+6930x^4-1260x^2+35)$
9	$\frac{1}{128}(12155x^9-25740x^7+18018x^5-4620x^3+315x)$
10	$\frac{1}{256}(46189x^{10}-109395x^8+90090x^6-30030x^4+3465x^2-63)$

Generalized Quadrature Method

- What if f(x) has singularities at x=a or x=b?
- What if a = -∞ or b = ∞?
- Generalized quadrature problem → weighted integral

$$\int_a^b f(x)\,dx \Rightarrow \int_a^b w(x)\,f(x)dx$$

→ Calculate this weighted integral using quadrature method, instead.

Weight function

- Orthogonal polynomials

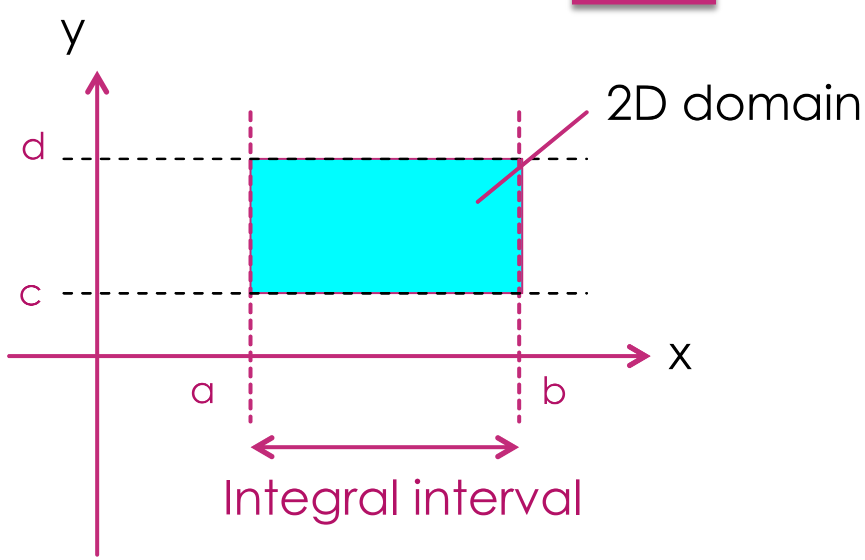
$$\int_a^b w(x)\,x^k\,p_n(x)\,dx = 0$$

For all k = 0, 1, 2, …, n-1

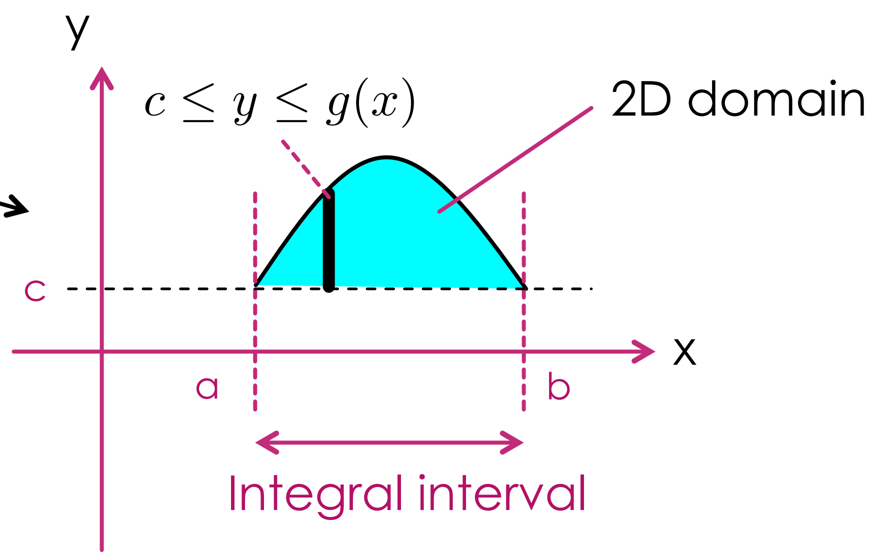
Interval	$\omega(x)$	Orthogonal polynomials
[−1, 1]	1	Legendre polynomials
(−1, 1)	$(1-x)^\alpha(1+x)^\beta, \quad \alpha, \beta > -1$	Jacobi polynomials
(−1, 1)	$\frac{1}{\sqrt{1-x^2}}$	Chebyshev polynomials (first kind)
[−1, 1]	$\sqrt{1-x^2}$	Chebyshev polynomials (second kind)
[0, ∞)	e^{-x}	Laguerre polynomials
[0, ∞)	$x^\alpha e^{-x}, \quad \alpha > -1$	Generalized Laguerre polynomials
(−∞, ∞)	e^{-x^2}	Hermite polynomials

2D & 3D Integration

- Requires the definition of domains
  - Use interface for implementing 2D and 3D domains
  - Surface element: **dS = dx dy = r dr dϕ**
  - Volume element: **dV = dx dy dz = r² sinθ dr dθ dϕ**
- Variables can be connected:
  - Via the function f(x,y) or f(x,y,z)
  - Via the domain bounds
- **Separation Rule**
  - Independent domain bounds
  - Separable functions



$$\text{Area} = \int_a^b \int_c^{g(x)} dydx$$



$$\underbrace{\int_{y=c}^{d} \int_{x=a}^{b}}_{\text{Independent bounds}} \underbrace{f(x)g(y)}_{\text{Separable functions}} \, dxdy = \int_{x=a}^{b} f(x)\, dx \times \int_{y=c}^{d} g(y)\, dy$$

2D & 3D Integration

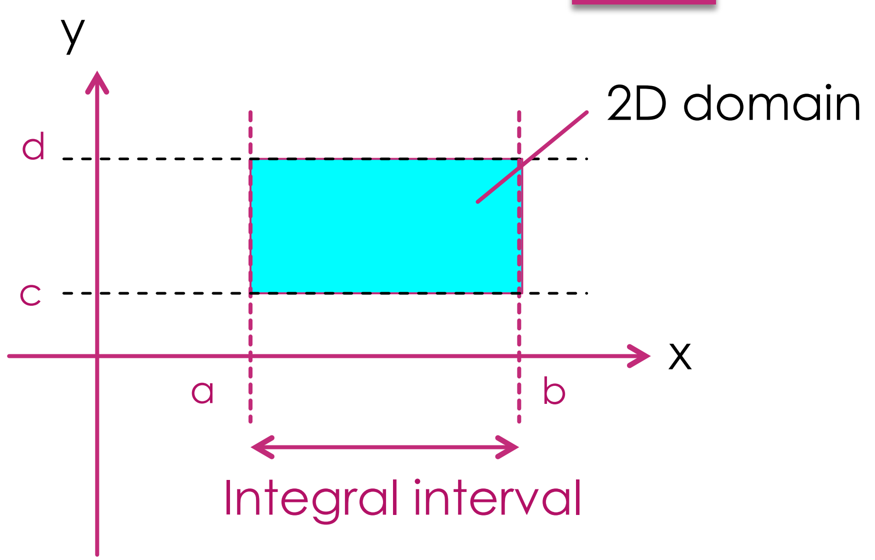
- Implementing Domain in java
- **2D domain**

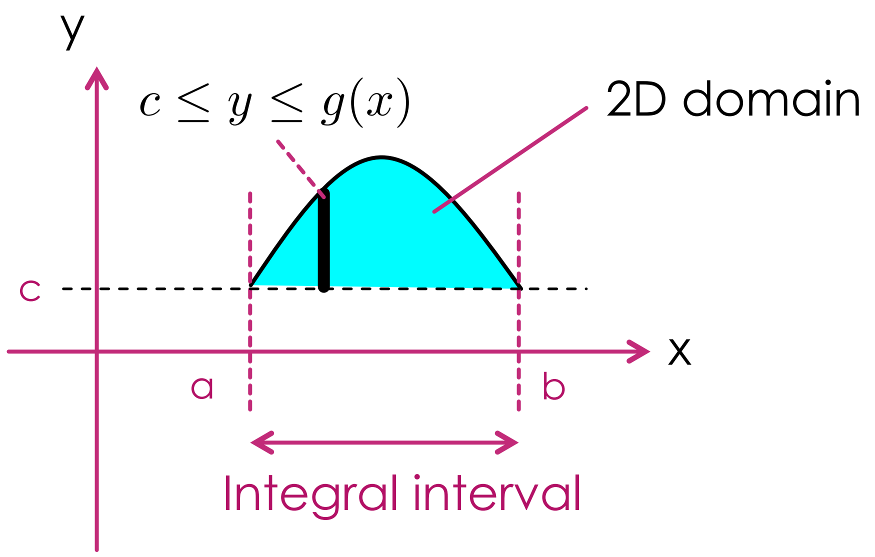
```
public interface IntegralDomain2D {
    double getVar1Min() ;
    double getVar1Max() ;
    double getVar2Min(double var1) ;
    double getVar2Max(double var1) ;
}
```

In general we may have dependent bounds

- **3D domain**

```
public interface IntegralDomain3D {
    double getVar1Min() ;
    double getVar1Max() ;
    double getVar2Min(double var1) ;
    double getVar2Max(double var1) ;
    double getVar3Min(double var1, double var2) ;
    double getVar3Max(double var1, double var2) ;
}
```





2D & 3D Integration

- **2D Integral**
  - Reduce the problem into **two** 1D integration
  - One variable is a free running variable
  - Second variable can be a dependent variable
    - **MUST integrate over the dependent variable first**

Integral2D

$$\boxed{\int_{x=a}^{b}\int_{y=g(x)}^{h(x)} f(x,y)\,dydx}$$

IntegralFunction2D

Step 1

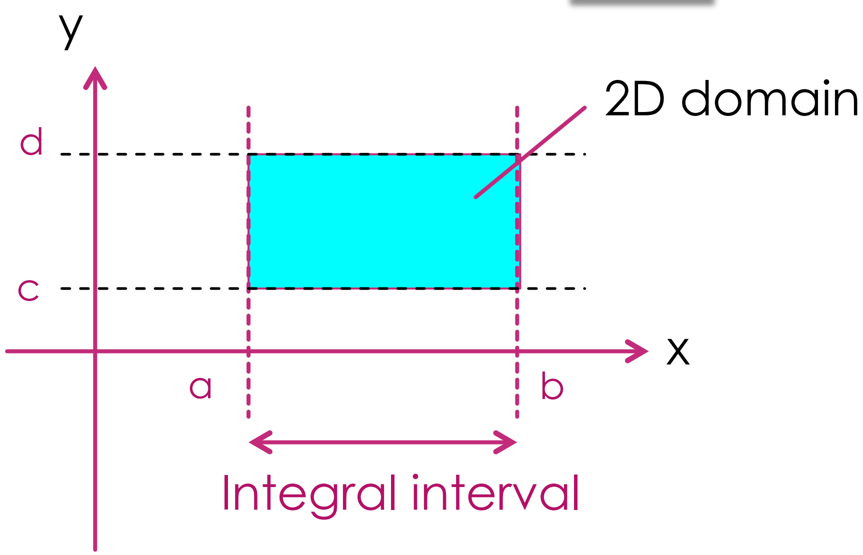
Step 2

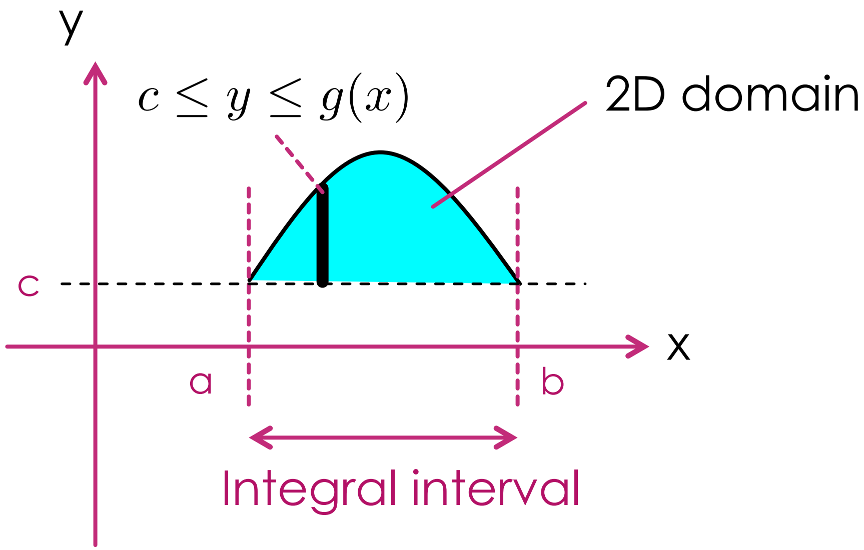
Integral1D

$$J(x) = \int_{y=g(x)}^{h(x)} f(x,y)\,dy \quad \Rightarrow \quad \int_{a}^{b} J(x)\,dx$$

To IntegralFunction1D

IntegralFunction1D





2D & 3D Integration

- **3D Integral**
  - Reduce the problem into **three** 1D integration
  - One variable is a free running variable
  - Second and third variables can be dependent variables

Note that
- Free runner → var1
- var2 → only on var1
- var3 → on var1, var2

This is my convention

$$\int_{x=a}^{b}\int_{y=g(x)}^{h(x)}\int_{z=K(x,y)}^{L(x,y)} f(x,y,z)\,dzdydx$$

Step 1 → Step 2 → Step 3

$$J(x,y) = \int_{z=K(x,y)}^{L(x,y)} f(x,y,z)\,dz \;\Rightarrow\; Q(x) = \int_{y=g(x)}^{h(x)} J(x,y)\,dy \;\Rightarrow\; \int_{x=a}^{b} Q(x)\,dx$$

Var 3 Var 2 Var 1

Tip: Result of each step is a 1D integral function for the next step.

Memoization of Integration

- Consider the following integral function

$$g(t) = \int_a^t f(x)\,dx$$ ➡ **Fundamental lemma of calculus :** $$\frac{d}{dt}g = f(t)$$

- How to find **g(t)** over an entire interval [a, t_1]?

Have to increase the number of subintervals

  - Option 1: simply integrate over [a, t] for each t → What if the interval is very large: t >> a?
  - Option 2: solve the differential equation with the initial condition **g(a) = 0**
  - Option 3: use "**memoized**" integration

Could be costly for good accuracy

$$\int_a^{t_2} = \int_a^{t_1} + \int_{t_1}^{t_2}$$

Memoization From Wikipedia

From Wikipedia, the free encyclopedia

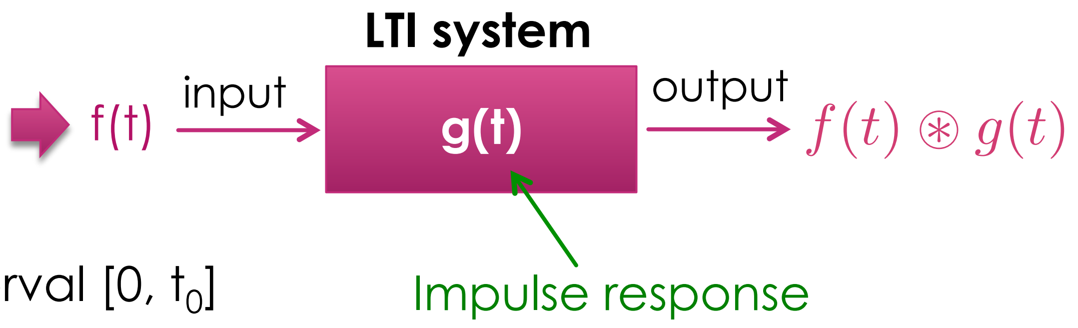
In computing, memoization is an optimization technique used primarily to speed up computer programs by storing the results of expensive function calls and returning the cached result when the same inputs occur again.

Tip: use a data structure such as List or Map for memoization.

Convolution Integral

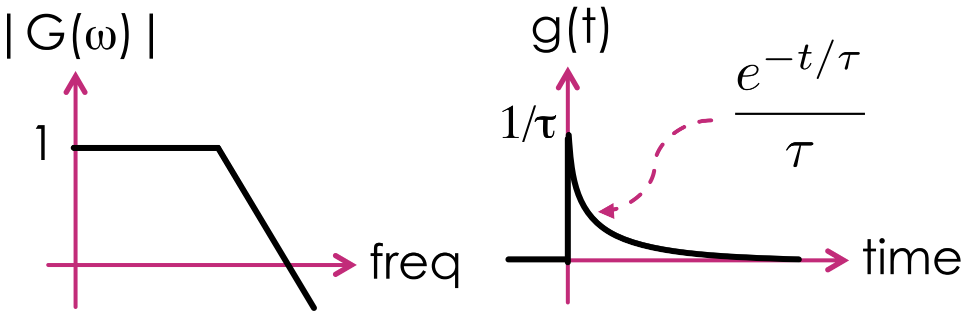
- **Convolution** appears a lot for linear time-invariant (LTI) systems (control theory)

$$f(t) \circledast g(t) = \int_{-\infty}^{\infty} f(y)\, g(t-y)\, dy = \int_{-\infty}^{\infty} f(t-y)\, g(y)\, dy$$

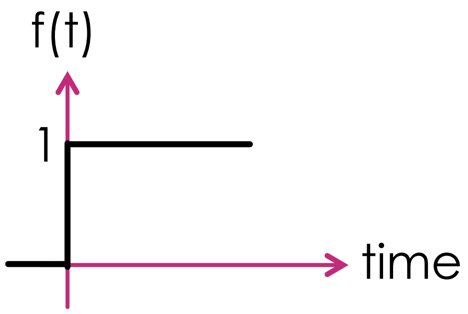


- We want to find the value of the output over a time interval [0, t_0]
  - Naïve approach: do the integration for each point in time
  - Professional: use memoization technique for fast evaluation
- Example: **Low-pass filter →** Calculating **Step Response →**

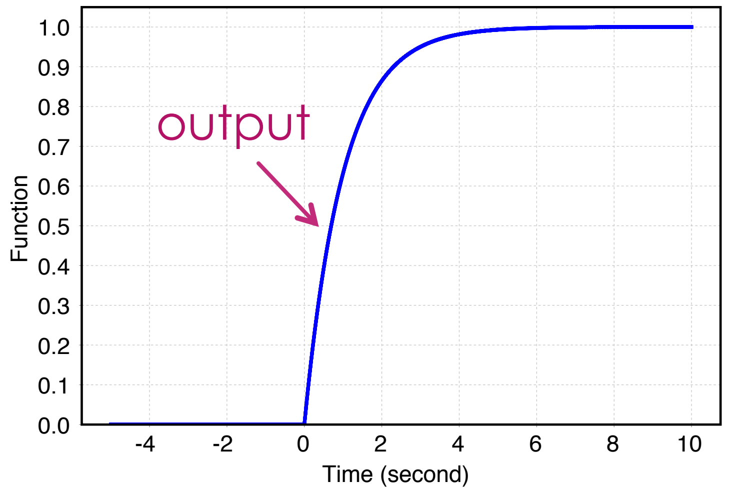
$$\text{response} = \int_0^t g(y)\, dy$$



Impulse response of low-pass filter



Input signal



Other Integral Libraries

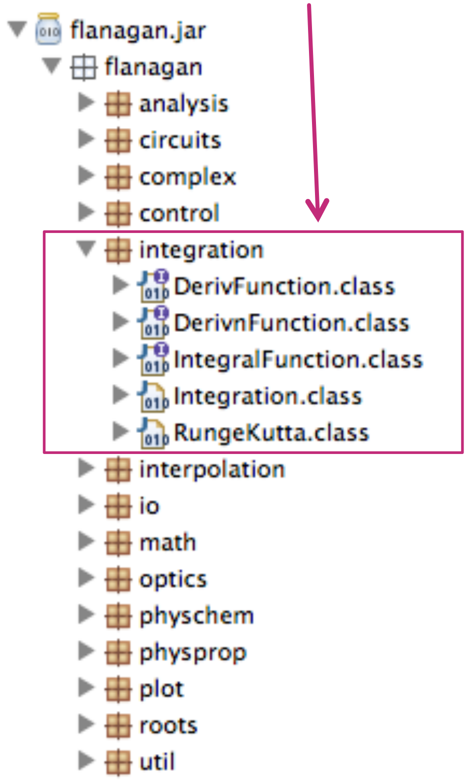
- Michael Flanagan's java scientific library
  - https://www.ee.ucl.ac.uk/~mflanaga/java/
- Has a package called "integration"
  - **Part 1:**
    - Integral Function
    - Integration

    An interface for defining function and a class for performing various integrations
  - **Part 2:**
    - DerivFunction
    - Deriv**n**Function
    - RungeKutta

    Interfaces and a class for solving systems of first order differential equations using Runge-Kutta method

$$y'(x) = \underbrace{f(x, y)}_{\text{DerivFunction}}$$

Integration package



Romberg Integration

- Trapezoid integration formula on steroid using Richardson acceleration
- We can apply Richardson method to any numerical integration approach
  - Rectangle
  - Trapezoid
  - Simpson
- **From Wikipedia:**

"Memoization"

Recursive expression for Romberg integration

$$\boxed{R(n,m) = \frac{1}{4^m - 1}\left(4^m\, R(n,m-1) - R(n-1,m-1)\right)}$$

$$R(0,0) = (b-a) \times \frac{f(a)+f(b)}{2}$$ → Trapezoid formula

Romberg's method

From Wikipedia, the free encyclopedia

In numerical analysis, Romberg's method (Romberg 1955) is used to estimate the definite integral by applying Richardson extrapolation (Richardson 1911) repeatedly on the trapezium rule or the rectangle rule (midpoint rule). The estimates generate a triangular array. Romberg's method is a Newton–Cotes formula – it evaluates the integrand at equally spaced points. The integrand must have continuous derivatives, though fairly good results may be obtained if only a few derivatives exist. If it is possible to evaluate the integrand at unequally spaced points, then other methods such as Gaussian quadrature and Clenshaw–Curtis quadrature are generally more accurate.