ENGG1003 - Thursday Week 8

Numerical integration

Steve Weller

University of Newcastle

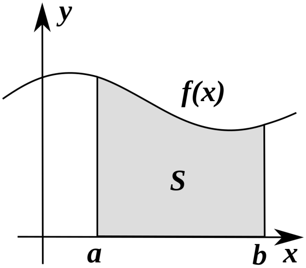
29 April 2021

Last compiled: April 28, 2021 8:43pm +10:00

Lecture overview

1. Basic ideas of numerical integration §6.1
   - engineering applications
   - terminology & notation
   - additivity
2. Trapezoidal method §6.2
3. Simpson's rule

1) Basic ideas of integration



$$S = \int_a^b f(x)dx$$

- area S is area under function $f(x)$ between lower limit a and upper limit b
- assume $f(x) \geq 0$
- calculus, eg: MATH1002, MATH1110

Engineering applications of integration

- 1. Area between curves 2. Distance, Velocity, Acceleration 3. Volume 4. Average value of a function 5. Work 6. Center of Mass 7. Kinetic energy; improper integrals 8. Probability 9. Arc Length 10. Surface Area

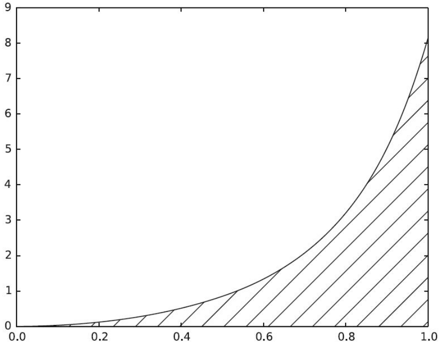
Distance = area under speed-time function

Assume that you speed up your car from rest, on a straight road, and wonder how far you go in T seconds. The displacement is given by the integral

$$\int_0^T v(t)dt$$

where $v(t)$ is the velocity (speed) as a function of time
Example:

$$v(t) = 3t^2 e^{t^3}$$



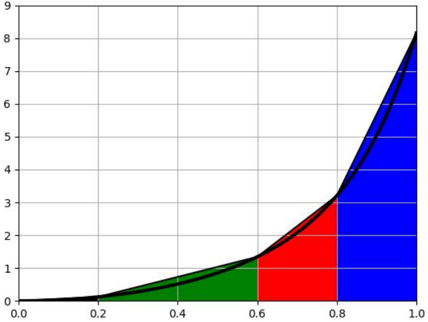
distance traveled in first second is cross-hatched area:

$$\int_0^1 v(t)dt$$

Start at time 0, end at time 1 (these are the lower and upper limits)

2) Trapezoidal method

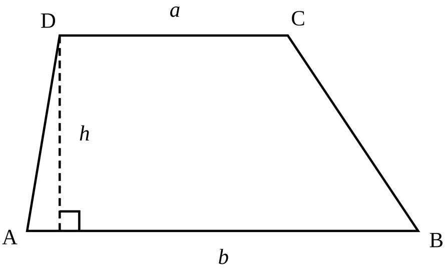
Example:

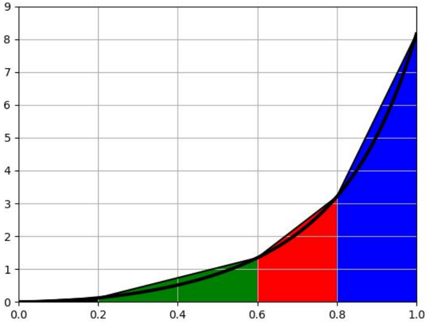


- approximate area under curve by total area of four trapezoids
  - black + green + red + blue
- area of each trapezoid is easy to calculate

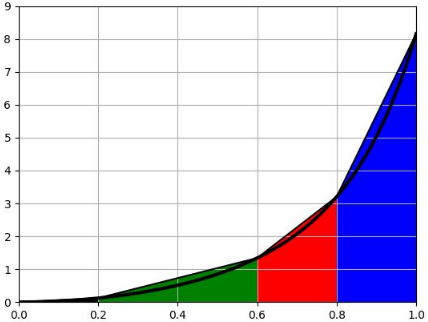
Numerical integration

- *trapezoid* is a "convex quadrilateral with at least one pair of parallel sides"
- area of trapezoid $\frac{a+b}{2}h$
- want "vertical" version – create image in PPT

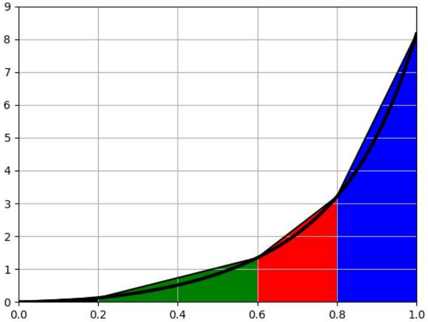




$$\int_0^1 v(t)dt \;=\; \int_0^{0.2} v(t)dt + \int_{0.2}^{0.6} v(t)dt + \int_{0.6}^{0.8} v(t)dt + \int_{0.8}^{1} v(t)dt$$



$$
\begin{aligned}
\int_0^1 v(t)dt &= \int_0^{0.2} v(t)dt + \int_{0.2}^{0.6} v(t)dt + \int_{0.6}^{0.8} v(t)dt + \int_{0.8}^{1} v(t)dt \\
&\approx h_1 \frac{v(0)+v(0.2)}{2} + h_2 \frac{v(0.2)+v(0.6)}{2} + \\
&\quad + h_3 \frac{v(0.6)+v(0.8)}{2} + h_4 \frac{v(0.8)+v(1)}{2}
\end{aligned}
$$



$$
\begin{aligned}
\int_0^1 v(t)dt &= \int_0^{0.2} v(t)dt + \int_{0.2}^{0.6} v(t)dt + \int_{0.6}^{0.8} v(t)dt + \int_{0.8}^{1} v(t)dt \\
&\approx h_1 \frac{v(0)+v(0.2)}{2} + h_2 \frac{v(0.2)+v(0.6)}{2} + \\
&\quad + h_3 \frac{v(0.6)+v(0.8)}{2} + h_4 \frac{v(0.8)+v(1)}{2}
\end{aligned}
$$

$$h_1 = 0.2, \quad h_2 = 0.4, \quad h_3 = 0.2, \quad h_4 = 0.2$$

General trapezoidal method

- want to approximate integral $\int_a^b f(x)dx$ by n trapezoids *of equal width*
  - total of n intervals: $[x_0, x_1]$, $[x_1, x_2]$, $\ldots [x_{n-1}, x_n]$

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)dx$$

General trapezoidal method

- want to approximate integral $\int_a^b f(x)dx$ by n trapezoids *of equal width*
  - total of n intervals: $[x_0, x_1]$, $[x_1, x_2]$, $\ldots [x_{n-1}, x_n]$

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)dx$$

$$\approx h\frac{f(x_0) + f(x_1)}{2} + h\frac{f(x_1) + f(x_2)}{2} + \cdots + h\frac{f(x_{n-1}) + f(x_n)}{2}$$

General trapezoidal method

- want to approximate integral $\int_a^b f(x)dx$ by n trapezoids *of equal width*
  - total of n intervals: $[x_0, x_1]$, $[x_1, x_2]$, $\ldots [x_{n-1}, x_n]$

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)dx$$

$$\approx h\frac{f(x_0) + f(x_1)}{2} + h\frac{f(x_1) + f(x_2)}{2} + \cdots + h\frac{f(x_{n-1}) + f(x_n)}{2}$$

In compact form:

$$\int_a^b f(x)dx \approx h\left[\frac{1}{2}f(x_0) + \{f(x_1) + \cdots + f(x_{n-1})\} + \frac{1}{2}f(x_n)\right]$$

Python code for trapezoidal method

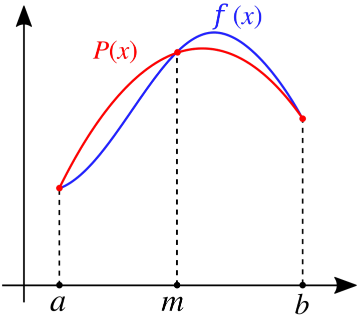
trapezoidal_method.py

```
import numpy as np

def v(t):
    return 3*t**2*np.exp(t**3)

def trapezoidal(f, a, b, n):
    h = (b-a)/n
    f_sum = 0
    for i in range(1, n, 1):
        x = a + i*h
        f_sum = f_sum + f(x)
    return h*(0.5*f(a) + f_sum + 0.5*f(b))

n = 4
trap = trapezoidal(v, 0, 1, n)
exact = np.exp(1) - 1

print('Trapezoidal, {} sub-intervals: {:.8f}'.format(n,trap))
print('Exact answer: {:.8f}'.format(exact))
```

Trapezoidal method: simulation results

- lines 3–4: function to be integrated
- lines 6–12: function to approximate integral using n trapezoids of equal width h
  - lines 8–11: compute $f(x_1) + \cdots f(x_{n-1})$
- line 16: exact result $\int_0^1 3t^2 e^{t^3} dt = e - 1$
- live demo, try $n = 4$, $n = 100$ and $n = 1000$ sub-intervals

3) Simpson's rule



- approximate $f(x)$ with parabola $P(x)$
- parabola $P(x)$ takes same values as $f(x)$ at end-points a and b, and midpoint $m = (a + b)/2$

Simpson's rule

- area under parabola $P(x)$ between a and b is:

$$\int_a^b P(x)dx$$

  ...which can be calculated *exactly* (proof omitted):

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

- as for trapezoidal rule, apply Simpson's rule on each "panel" of width h

Python code for Simpson's rule

- write as a function
- live demo, experiment with number of panels

Lecture summary

1. Basic ideas of integration
2. Trapezoidal method §6.2
3. Simpson's rule