ENGG1003 - Monday Week 9

Numerical integration: review and applications

Steve Weller

University of Newcastle

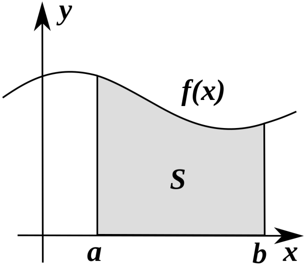
3 May 2021

Last compiled: May 2, 2021 7:38pm +10:00

Lecture overview

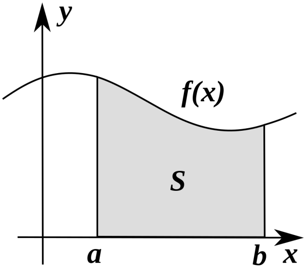
1. Review of integration
2. Applications of integration
   - average value of a function
   - area between curves

1) Review of integration



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

- interested in calculating shaded area S under function $f(x)$ between a and b
  - S is the *definite integral* of f over $[a, b]$



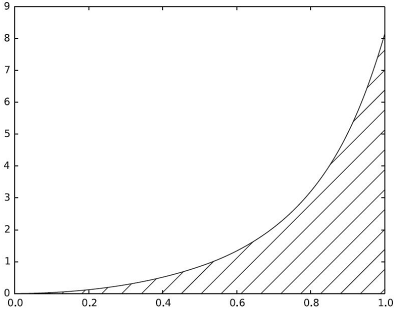
$$\boxed{S = \int_a^b f(x)dx}$$

- for some choices of f, possible to use calculus to compute S exactly
  - eg: MATH1002, MATH1110
  - numerical methods in this lecture apply even when f is hard (or impossible!) to integrate
- assume $f(x) \geq 0$

Integral is area under curve

Example:

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

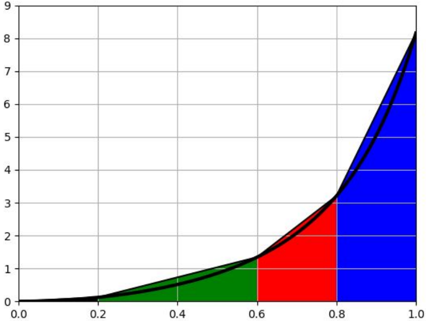


Cross-hatched area:

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

Trapezoidal method

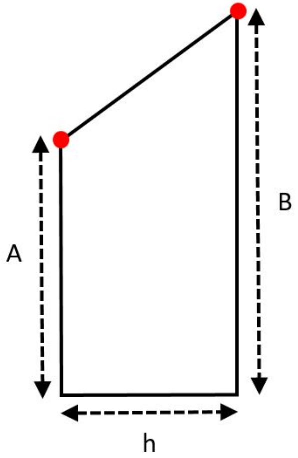
Example:



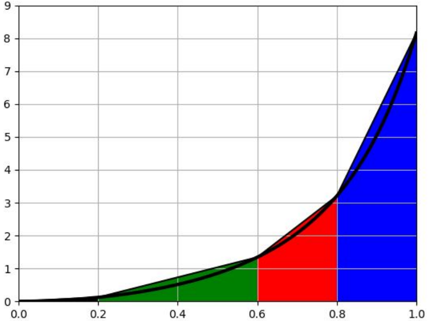
- approximate area under curve by total area of four trapezoids
  - black + green + red + blue
- area of each trapezoid is easy to calculate

Area of trapezoid

- area of trapezoid $= h \cdot \frac{A+B}{2}$

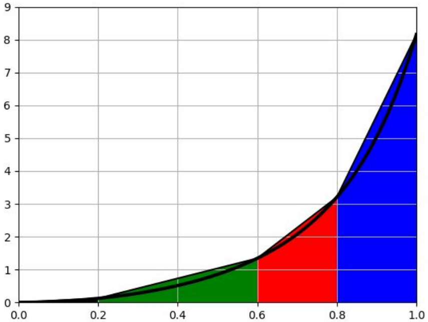


fourPanels.py



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

fourPanels.py

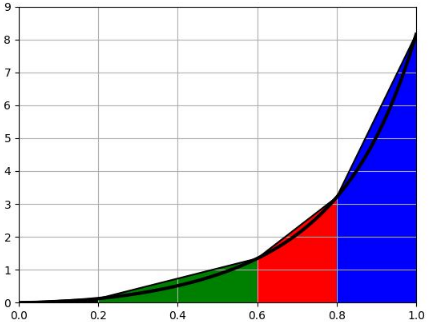


$$\int_0^1 v(t)dt = \int_0^{0.2} v(t)dt + \int_{0.2}^{0.6} v(t)dt + \int_{0.6}^{0.8} v(t)dt + \int_{0.8}^{1} v(t)dt$$

$$\approx h_1 \frac{v(0) + v(0.2)}{2} + h_2 \frac{v(0.2) + v(0.6)}{2}$$

$$+ h_3 \frac{v(0.6) + v(0.8)}{2} + h_4 \frac{v(0.8) + v(1)}{2}$$

fourPanels.py



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

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

General trapezoidal method

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

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

General trapezoidal method

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

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

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

General trapezoidal method

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

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

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

In compact form:

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

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))
```

2) Applications of integration: i) average value of a function

Average (or *mean*) value of function $f(t)$ on interval $[a, b]$ is defined by:

$$\bar{f} = \frac{1}{b-a}\int_a^b f(t)dt$$

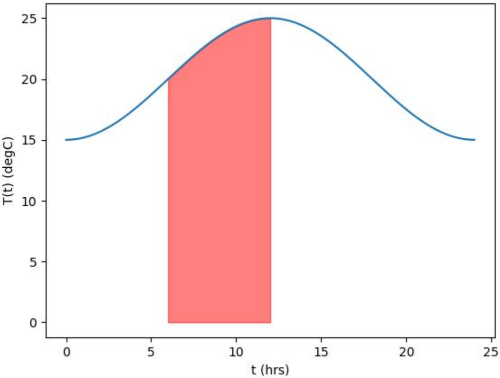
Example: average value of a function

The daily temperature of the outside air is given by the equation

$$T(t) = 20 - 5\cos\left(\frac{\pi t}{12}\right)$$

where t is measured in hours $(0 \le t \le 24)$ and T is measured in degrees Celsius ($^\circ$C)

1. Plot $T(t)$ over one day: $0 \le t \le 24$
2. Use numerical integration to find the average temperature between $a = 6$ and $b = 12$ hours



- blue line: temperature function
- red area: integral $\int_6^{12} T(t)dt$
- average temperature $\bar{T} = \frac{1}{12-6} \int_6^{12} T(t)dt \approx 23.18$ °C

Python code: average via integration

computeaverage.py

```python
# computeaverage
import numpy as np
import matplotlib.pyplot as plt

def T(t):
    return 20 - 5*np.cos(np.pi*t/12)

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))
```

- same as `trapezoidal_method.py` code, Thursday Week 8

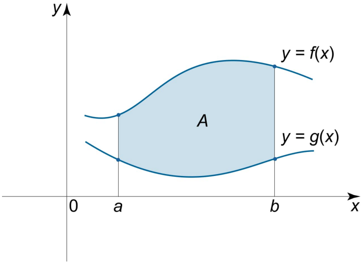
  ...except function to be integrated is now $T(t)$

Python code: average via integration

```
1  n = 1000
2  a = 6
3  b = 12
4  Tavg = trapezoidal(T, a, b, n)/(b-a)
5  print('Average temp over [{:},{:}] hours is {:.2f} degC'.format(a,b,Tavg))
6  
7  t = np.linspace(0,24,1000)
8  t612 = np.linspace(6,12,1000)
9  plt.plot(t,T(t))
10 plt.fill_between(t612,T(t612),color='r',alpha=0.5) #alpha=transparency
11 plt.xlabel('t (hrs)')
12 plt.ylabel('T(t) (degC)')
13 plt.show()
```

- lines 1–4: approximate $\bar{T}$ using trapezoidal method
  - divide $[6, 12]$ into 1000 sub-intervals
- lines 8 & 10: `fill_between` function plots region of integration over $[6, 12]$

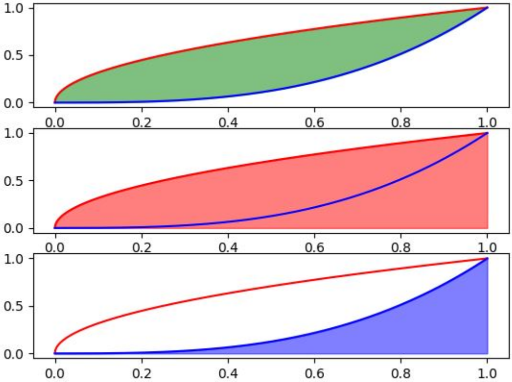
2) Applications of integration: ii) area between curves



- $f(x) \geq g(x)$ on interval $[a, b]$
- area between $f(x)$ and $g(x)$ on this interval:

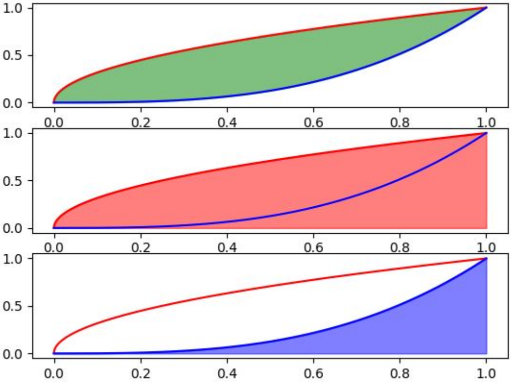
$$A = \int_a^b [f(x) - g(x)]\, dx = \int_a^b f(x)dx - \int_a^b g(x)dx$$

Example: area between curves



red curve: $f(x) = \sqrt{x}$
blue curve: $g(x) = x^3$
curves intersect at $(0, 0)$ and $(1, 1)$

Example: area between curves



red region: area under $f(x) = \sqrt{x}$

blue region: area under $g(x) = x^3$

green region: area between curves $f(x)$ and $g(x)$

Example: area between curves

Using `areabetweencurves.py`:

- trapezoidal method
- 1000 sub-intervals on $[0, 1]$

$$
\begin{aligned}
A &= \int_0^1 \sqrt{x}dx - \int_0^1 x^3 dx \\
&\approx 0.666660 - 0.25 \\
&= 0.416660
\end{aligned}
$$

- exact answer: $A = \frac{5}{12} = 0.416667$

https://tutorial.math.lamar.edu/classes/calcii/centerofmass.aspx

Python code: area between curves

areabetweencurves.py

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return np.sqrt(x)

def g(x):
    return x**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 = 1000
x = np.linspace(0,1,1000)
underf = trapezoidal(f, 0, 1, n)
underg = trapezoidal(g, 0, 1, n)
A = underf - underg
```

Python code commentary

- lines 4–8: want area between $f(x)$ and $g(x)$
  - $f(x) = \sqrt{x}$ and $g(x) = x^3$
- lines 10–16: trapezoidal approximation to integral
- lines 20–21: approximate areas under f and g
- line 22: area between f and g is area under f less area under g

Python code: area between curves

areabetweencurves.py—continued

```
print('Trapezoidal, {} sub-intervals'.format(n))
print('Area under f: {:.6f}'.format(underf))
print('Area under g: {:.6f}'.format(underg))
print('Area between f and g: {:.6f}'.format(A))

plt.subplot(3,1,1)
plt.plot(x, f(x), color='r')
plt.plot(x, g(x), color='b')
plt.fill_between(x, f(x), g(x), color='g', alpha=0.5)  # alpha=
    transparency
plt.subplot(3,1,2)
plt.plot(x,f(x),color='r')
plt.plot(x,g(x),color='b')
plt.fill_between(x,f(x),color='r',alpha=0.5) #alpha=transparency
plt.subplot(3,1,3)
plt.plot(x,f(x),color='r')
plt.plot(x,g(x),color='b')
plt.fill_between(x,g(x),color='b',alpha=0.5) #alpha=transparency

plt.show()
```

Python code commentary

- lines 1–4: display numerical results on console
- lines 6, 10, 14: use `subplot` to display three plots in one window
    - first & second arguments represent number of rows & columns
    - third argument is index of current plot
    - Example: `subplot(3,1,2)` means figure has 3 rows 1 column and this is the second plot ie: middle plot of a stack of three plots in a column
- lines 9, 13, 17: `fill_between` plots green, red and blue regions

Lecture summary

- Review of integration
- Applications of integration
- Next lecture: use numerical integration to compute probabilities
- **Reminder:** assessed lab 2 this week, in face-face lab sessions