Computational Methods and Modelling 3

Lecture 9: Dr. Edward McCarthy

Topic 1: Numerical Integration

THE UNIVERSITY *of* EDINBURGH
School of Engineering

Numerical Integration: What is it?
Why is it used? Don't we have analytical techniques?

1. There are many occasions when it is not possible or efficient to attempt the analytical integration of a known function.

Example: $f(x) = \exp(-x^2)$

Please attempt an analytical integration of this function!

Let's try this:

$$\int_0^1 f(x) = \int_0^1 \exp\left(-x^2\right)dx$$

If we were integrating f(x) = e(x), the resulting integral would be as follows:

$$\int_0^1 \exp(ax) = \Big|_0^1 \frac{1}{a}\exp(ax)$$

However, our function is not of this form and in fact is much more complicated to integrate. The actual result is as follows:

$$\int_0^1 \exp\left(-x^2\right)dx = \frac{\sqrt{\pi}}{2} erf(x)$$

Numerical Integration: What is it?
Why is it used? Don't we have analytical techniques?

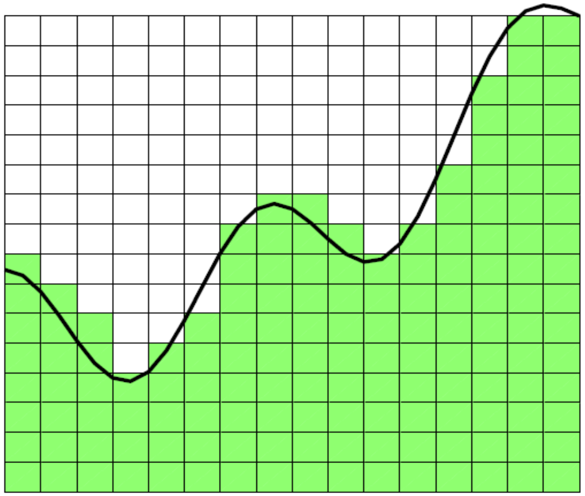
The error function itself is a function of a second variable, t, which complicates the evaluation of the integral considerably.

$$\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2}\, dt$$
$$= \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2}\, dt.$$

To pursue this further, we would need to define an integral interval in both t and x
However, it is possible to evaluate our function numerically using various techniques and algorithms. The two main categories of techniques include:

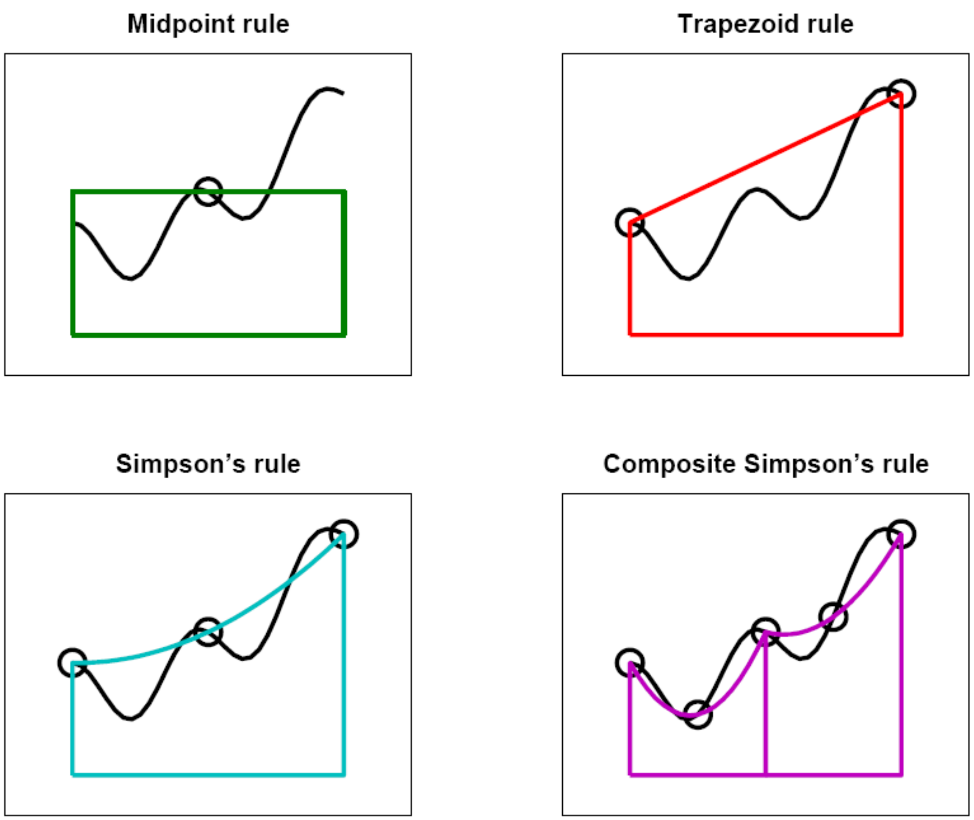
1. Quadrature Techniques
2. Adaptive Algorithms

Quadrature Techniques: Integration of a function by calculation of area under the curve f(x)



- Quadrature is the process of evaluating the integral of a curve (the area under the curve).
- This is approximated as the sum of the areas of many discrete subelements.
- By definition, it is not exact, but approximate.
- Particularly suitable for machine computation.

Quadrature Techniques: Integration of a function by calculation of area under the curve f(x)



- There are four classic variations of quadrature for numerical integration, each increasing in complexity.

THE UNIVERSITY of EDINBURGH
School of Engineering

Quadrature Techniques: Integration of a function by calculation of area under the curve f(x)

- The relative accuracy of the four techniques is traditionally demonstrated by testing the following simple cubic integral.

$$\int_0^1 x^2 dx = \frac{1}{3}.$$

- The Midpoint Rule evaluates this integral as follows with an inaccurate result.

$$M = 1\left(\frac{1}{2}\right)^2 = \frac{1}{4}.$$

- The Trapezoidal Rule overestimates the integral significantly:

$$T = 1\left(\frac{0+1^2}{2}\right) = \frac{1}{2}.$$

- This allows the following rule to be written that reconciles the estimates M and T with the true value of the integral, named S, i.e.

$$S - T = -2(S - M)$$

Derivation of Simpson’s Third Rule

- Rearranging this equation to isolate for S gives the following expression:

$$S = \frac{2}{3}M + \frac{1}{3}T$$

- This is the basis for Simpson’s Third Rule and give an exact answer for a third order integral (i.e. where the highest power of x is 3). It is not exact for powers of 4 and greater, and so is referred to as a fourth order rule.

- Another route to Simpson’s Rule is ***quadratic interpolation*** of the function between limits a, b and their midpoint c = (a + b)/2. Integration of this function gives the following expression for S:

$$S = \frac{h}{6}(f(a) + 4f(c) + f(b)).$$

Derivation of Simpson's Third Rule

First the three quadratic constants are determined from the following analysis where the quadratic function is exactly true at three points for $f_2(x)$:

$$f_2(x) = a_0 + a_1 x + a_2 x^2$$

Choose

$$(a, f(a)),\ \left(\frac{a+b}{2}, f\left(\frac{a+b}{2}\right)\right),\ \text{and}\ (b, f(b))$$

as the three points of the function to evaluate a_0, a_1 and a_2.

$$f(a) = f_2(a) = a_0 + a_1 a + a_2 a^2$$

$$f\left(\frac{a+b}{2}\right) = f_2\left(\frac{a+b}{2}\right) = a_0 + a_1\left(\frac{a+b}{2}\right) + a_2\left(\frac{a+b}{2}\right)^2$$

$$f(b) = f_2(b) = a_0 + a_1 b + a_2 b^2$$

The function is evaluated at three points to enable three equations to solve for a_0, a_1 and a_2.

Solving the above three equations for unknowns, a_0, a_1 and a_2 give

$$a_0 = \frac{a^2 f(b) + abf(b) - 4abf\left(\frac{a+b}{2}\right) + abf(a) + b^2 f(a)}{a^2 - 2ab + b^2}$$

$$a_1 = -\frac{af(a) - 4af\left(\frac{a+b}{2}\right) + 3af(b) + 3bf(a) - 4bf\left(\frac{a+b}{2}\right) + bf(b)}{a^2 - 2ab + b^2}$$

$$a_2 = \frac{2\left(f(a) - 2f\left(\frac{a+b}{2}\right) + f(b)\right)}{a^2 - 2ab + b^2}$$

http://mathforcollege.com/nm/mws/gen/07int/mws_gen_int_txt_simpson13.pdf

Derivation of Simpson's Third Rule

Now Simpson's 1/3rd Rule is derived as follows using these three values:

$$I \approx \int_a^b f_2(x)dx$$

$$= \int_a^b \left(a_0 + a_1 x + a_2 x^2\right)dx$$

$$= \left[a_0 x + a_1 \frac{x^2}{2} + a_2 \frac{x^3}{3}\right]_a^b$$

$$= a_0(b-a) + a_1 \frac{b^2 - a^2}{2} + a_2 \frac{b^3 - a^3}{3}$$

Substituting values of a_0, a_1 and a_2 give

$$\boxed{\int_a^b f_2(x)dx = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]}$$

Basis of derivation is substitution of a quadratic equation to represent f(x).

The parameters a_0, a_1 and a_2 are substituted from theory as per previous page.

Composite Simpson's Rule for n subintervals

1. The most basic Simpson's Rule is based on quadratic interpolation at a minimum of three points: a start, middle and end point of the integration interval.
2. However, further subdivisions of the two sub-areas in the intervals ac and cb will make the method more accurate.
3. For instance, halving each of the existing intervals between a and c and between b and c will result in the following **Composites Simpson's Rule for n = 4 intervals.**

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\sum_{j=1}^{n/2}\Big[f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})\Big]$$

$$= \frac{h}{3}\Bigg[f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n)\Bigg]$$

4. The same principle of subdivision will make midpoint and trapezoidal rules more accurate as well.

Computational Methods and Modelling

Lecture 9: Dr. Edward McCarthy

Topic 1: Numerical Integration

THE UNIVERSITY of EDINBURGH
School of Engineering

Quadrature Techniques: Integration of a function by calculation of area under the curve f(x)

In summary, the three main quadrature rules are:

a. Rectangle/Midpoint Rule:

$$\int_a^b f(x)\,dx \approx h\sum_{n=0}^{N-1} f(x_n)$$

where $h = (b-a)/N$ and $x_n = a + nh$.

b. Trapezoidal Rule:

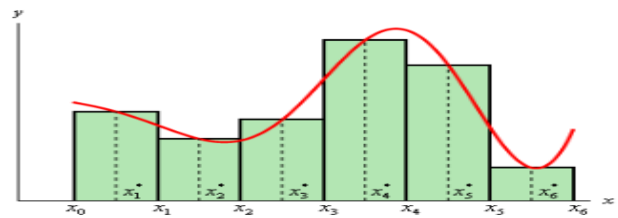
$$\int_a^b f(x)\,dx \approx h\left[\frac{1}{2}f(x_0) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2}f(x_n)\right]$$

c. Composite Simpson's Rule:

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

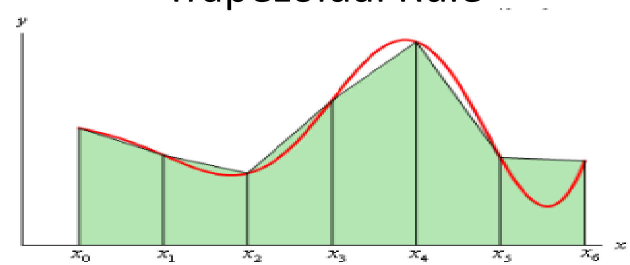
$$= \frac{h}{3}\left[f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n)\right]$$

Rectangle Rule

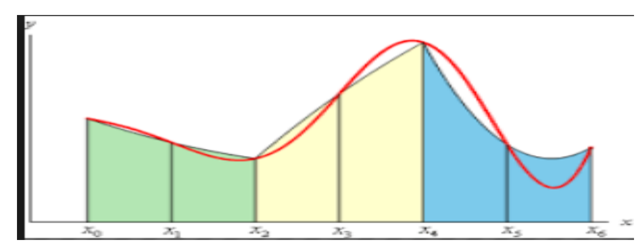


$\int_a^b f(x)\,dx \approx \Delta x\, f(x_1^*) + \Delta x\, f(x_2^*) + \cdots + \Delta x\, f(x_n^*)$

Trapezoidal Rule



Simpson's Rule



Use of the Rectangle Rule: Python Code

```
def calculate_dx (a, b, n):
        return (b-a)/float(n)
```

Calculate the integration step interval dx using the defined bounds a and b and the number of intervals chosen

```
def rect_rule (f, a, b, n):
        total = 0.0
        dx = calculate_dx(a, b, n)
        for k in range (0, n):
                total = total + f((a
+ (k*dx)))
        return dx*total
```

Implements rectangle rule calculation of integral based on calculation of area in each rectangular strip.

```
def f(x):
    return x**2
```

Define the function to be integrated

```
print(rect_rule(f, 0, 10, 100000))
```

Call the rectangle rule function defining the function, a, b and the number of strips used.

Use of the Trapezoidal Rule: Python Code

```
def trapz(f,a,b,N=50):
    x = np.linspace(a,b,N+1) # N+1 points make N subintervals
    y = f(x)
    y_right = y[1:] # right endpoints
    y_left = y[:-1] # left endpoints
    dx = (b - a)/N
    T = (dx/2) * np.sum(y_right + y_left)
    return T
```

Define the core Trapezoid rule function defining the function, a, b and the number of strips, N, used. T is the integral in [a,b].

```
def f(x):
    return np.exp(-x**2)
```

Define the function separately that is to be integrated.

```
a=float(input("Please enter a value for the lower bound:\n"))
b=float(input("Please enter a value for the upper bound:\n"))
n=int(input("Please enter a value for the number of intervals:\n"))
```

Ask the user to define the integral boundary and the number of subintervals.

```
print(trapz(f,a,b,n))
```

Then call the function trapz using the print function.

Note that the order of operations is not what you might expect. The trapezoidal operation is defined first followed by the function and parameters. Then the function is called to operate in the last print command

Graphing the Trapezoidal Rule: Python Code

You can also derive the integral by using a geometric approach to calculate the actual area of the trapezoid and visualise this in a plot as follows:

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

x = np.linspace(-1.5,1.5,100)
y = np.exp(x**2)
plt.plot(x,y)

x0 = 0; x1 = 1;
y0 = np.exp(x0**2); y1 = np.exp(x1**2);
plt.fill_between([x0,x1],[y0,y1])

plt.xlim([-1.5,1.5]); plt.ylim([0,10]);
plt.show()
A = 0.5*(y1 + y0)*(x1 - x0)
print("Trapezoid area:", A)
```

Different programme structure used. First import numpy elements.

Now define the x domain and the function y. Plot the curve.

These commands fill a defined region of the graph to form a trapezoid area. Then it calculates the area, A, shown within the trapezoid geometrically. This A is the approximation of the integral of the function.

Use of Simpson's Rule: Python Code

```
def simps(f,a,b,N=50):

    if N % 2 == 1:
        raise ValueError("N must be an even integer.")
    dx = (b-a)/N
    x = np.linspace(a,b,N+1)
    y = f(x)
    S = dx/3 * np.sum(y[0:-1:2] + 4*y[1::2] + y[2::2])
    print(S)
    return S

f = lambda x: x**3
solution = simps(f,1,2,24)
print(solution)
```

Define the function, limits, and number of evaluation strips, N.

Evaluate the modulus (%) of N by 2 (i.e. if there is a remainder of 1 after N is divided by 2, N is an odd number and can't be used.

Define x increment, dx, an define a linear space in x with N+1 points.

Calculate the area under curve for the region [a,b] using Simpson's (One Third) Rule

Define the function f, call simps and allocate the solution to 'solution'. Print this value.

https://www.math.ubc.ca/~pwalls/math-python/integration/simpsons-rule/

Three Eighths Simpson's Rule

1. Up to now we have used a version of the Simpson's Rule that is based on quadratic interpolation at three points: a start, middle and end point of the integration interval.
2. A more sophisticated version of this model is built by interpolating a function f(x) using a cubic rather than a quadratic function with arbitrary polynomial coefficients that have to be fitted at four points along f(x).
3. The derivation follows the same path as we saw for the One-Third Simpson's Rule.
4. The resulting expression is:

$$\int_a^b f(x)\,dx = \frac{(b-a)}{8}\left[f(a) + 3f\left(\frac{2a+b}{3}\right) + 3f\left(\frac{a+2b}{3}\right) + f(b)\right]$$

5. So how do all these techniques compare in the case of our original challenging integral?
6. Firstly, write a code to execute this integral.

Comparison of Outputs of Rules

Let us take the example of the following function and evaluate it using the four rules implemented by the three Python codes shown above for 1000 intervals (N = 1000):

$$\int_0^1 f(x) = \int_0^1 \exp\left(-x^2\right) dx$$

Exact Integral = 0.746824

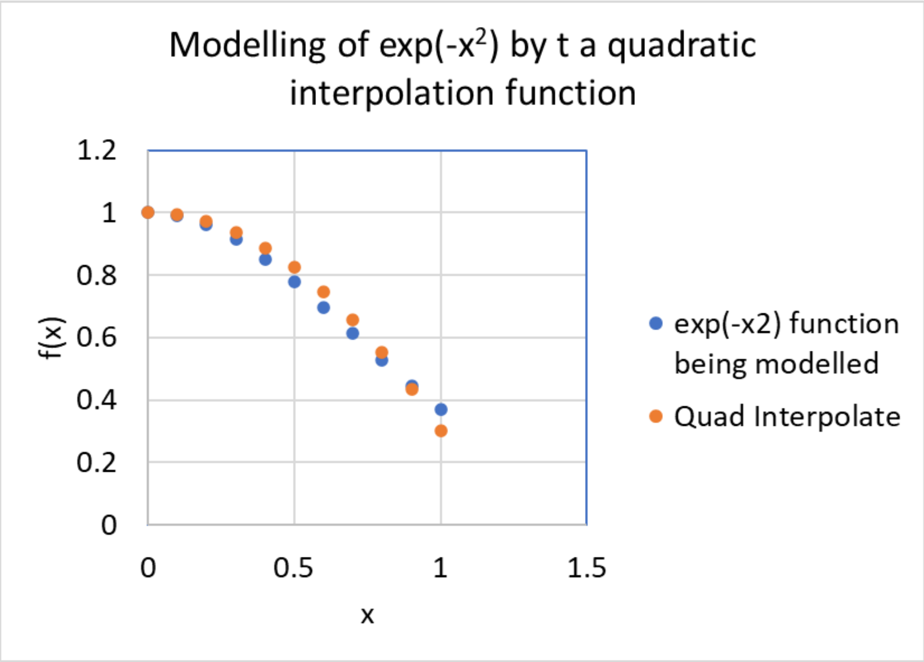
Rule (Composite Impl.)	Estimated Integral	% Error on Exact
Rectangle	0.747140	-4.23E-04
Trapezoid	0.683939	8.42E-02
Simpson's Third	0.746824	1.78E-07
Simpson's Three Eighths	0.746732	1.23E-04

In this case, due to the curvature of the function above, Simpson's Third Rule (quadratic interpolation) is more accurate than the Three Eighths version, which uses a cubic interpolate.

Comparison of Outputs of Rules

$$\int_0^1 f(x) = \int_0^1 \exp\left(-x^2\right)dx$$ Exact Integral = 0.746824

Plotting a quadratic interpolation of the function over the range of interest, one can see the close fit achieved over this range. The flat curvature of the function above does not require the extra complexity of a cubic function.



THE UNIVERSITY *of* EDINBURGH
School of Engineering

Adaptive Algorithm Numerical Methods

1. We just saw that ***quadrature techniques work well for certain functions and less well for others.*** Another approach to numerical integration is the use of adaptive algorithms.

2. Adaptive algorithms allow intervals ***to be subdivided as necessary during the numerical integration if sufficient accuracy is not achieved after n steps.*** (i.e., the interval for the n+1th step is subdivided where accuracy for the nth step is inadequate).

3. As such ***the adaptive algorithm approach can be applied dynamically within any of the three quadrature techniques*** listed above.

4. This has the advantage of initially saving computation time by adapting a course step size, which is only reduced if the error of the developing solution remains beneath a pre-set value.

5. The first step is adaptation of Simpson's Rule to enable in-code adjustment of step size.

Adaptive Algorithm Numerical Methods

6. The 'basic' equation is as follows:

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

7. We can refine the estimate returned by halving the step size to give an equation of the following form:

$$\int_a^b f(x) \approx \frac{b-a}{12}\left[f(a)+4f(d)+2f(c)+4f(e)+f(b)\right]$$

8. Here, **c = (a+b)/2; d = (a+c)/2; e = (c+b)/2.** **h_1 = (b-a)/6; h_2 = (b-a)/12.**

9. The reduction in error between the two equations can be calculated as the difference in the integral estimates, i.e.

$$E = I(h_2) - I(h_1)$$

10. This subdivision approach can be applied forever, theoretically, but we must establish a reasonable estimate using minimal computational power.

Exercise: Write a code to implement this algorithm

Adaptive Algorithm Numerical Methods

1. Let's examine how this would improve the calculation of the numerical integral for the following complex function:

Exercise for this Week.

$$\int_0^1 f(x) = \int_0^1 \exp\left(-x^2\right) dx$$

2. Run the code you have written for the Adaptive Simpson's Technique (previous slide).
3. Evaluate the integral above and compare the value obtained with that from previous techniques in this Lecture.