

Simple Quadrature Rules

January 24, 2021

1 Numerical integration

Anne Kværnø, André Massing

Date: Jan 25, 2021

If you want to have a nicer theme for your jupyter notebook, download the [cascade stylesheet file tma4125.css](#) and execute the next cell:

```
[1]: from IPython.core.display import HTML
def css_styling():
    try:
        with open("tma4125.css", "r") as f:
            styles = f.read()
            return HTML(styles)
    except FileNotFoundError:
        pass #Do nothing

# Comment out next line and execute this cell to restore the default notebook
→style
css_styling()
```

```
[1]: <IPython.core.display.HTML object>
```

2 Introduction

Imagine you want to compute the finite integral

$$I[f](a, b) = \int_a^b f(x) dx.$$

The “usual” way is to find a primitive function F (also known as the indefinite integral of f) satisfying $F'(x) = f(x)$ and then to compute

$$\int_a^b f(x) dx = F(b) - F(a).$$

While there are many analytical integration techniques and extensive tables to determine definite integral for many integrands, more often than not it may not be feasible or possible to compute the integral. For instance, what about

$$f(x) = \frac{\log(2 + \sin(1/2 - \sqrt{(x)})^6)}{\log(\pi + \arctan(\sqrt{1 - \exp(-2x - \sin(x))}))}?$$

Finding the corresponding primitive is highly likely a hopeless endeavor. And sometimes there even innocent looking functions like e^{-x^2} for which there is not primitive functions which can be expressed as a composition of standard functions such as \sin, \cos . etc.

A **numerical quadrature** or a **quadrature rule** is a formula for approximating such definite integrals $I[f](a, b)$. Quadrature rules are usually of the form

$$Q[f](a, b) = \sum_{i=0}^n w_i f(x_i),$$

where x_i, w_i for $i = 0, 1, \dots, n$ are respectively the *nodes/points* and the *weights* of the quadrature rule. To emphasize that a quadrature rule is defined by some given quadrature points $\{x_i\}_{i=0}^n$ and weights $\{w_i\}_{i=0}^n$.

We sometimes might write

$$Q[f](\{x_i\}_{i=0}^n, \{w_i\}_{i=0}^n) = \sum_{i=0}^n w_i f(x_i).$$

If the function f is given from the context, we will for simplicity denote the integral and the quadrature simply as $I(a, b)$ and $Q(a, b)$.

2.1 Example 1: Quadrature rules from previous math courses

The [trapezoidal rule](#), the [midpoint rule](#) and [Simpson's rule](#) known from previous courses are all examples of numerical quadratures, and we quickly review them here:

- **Midpoint rule** is the simplest possible quadrature rule defined by

$$Q[f](a, b) := (b - a) f\left(\frac{a + b}{2}\right).$$

The node is given by the midpoint, $x_0 = \frac{a+b}{2}$ with the corresponding weight $w_0 = b - a$.

$$Q[f](a, b) = w_0 f(x_0)$$

- **Trapezoidal rule** is given by

$$Q[f](a, b) := (b - a) \left(\frac{f(a) + f(b)}{2} \right)$$

and thus the nodes are defined by $x_0 = a, x_1 = b$ with corresponding weights $w_0 = w_1 = \frac{b-a}{2}$.

- Finally, **Simpson's rule** which you know from M1, is defined as follows:

$$Q[f](a, b) = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right),$$

which we identify as quadrature rule with 3 points $x_0 = a, x_1 = \frac{a+b}{2}, x_2 = b$ and corresponding weights $w_0 = w_2 = \frac{b-a}{6}$ and $w_1 = \frac{4(b-a)}{6}$.

In this note we will see how quadrature rules can be constructed from integration of interpolation polynomials. We will demonstrate how to do error analysis and how to find error estimates. In the sequel, we will use material from *Preliminaries*, section 3.2, 4 and 5.

As usual, we import the necessary modules before we start computing.

```
[1]: %matplotlib inline

from numpy import *
from matplotlib.pyplot import *
from math import factorial
newparams = {'figure.figsize': (8.0, 4.0), 'axes.grid': True,
             'lines.markersize': 8, 'lines.linewidth': 2,
             'font.size': 14}
rcParams.update(newparams)
```

3 Quadrature based on polynomial interpolation.

This section relies on the content of the note on polynomial interpolation, in particular the section on Lagrange polynomials.

Choose $n + 1$ distinct nodes $x_i, i = 0, \dots, n$ in the interval $[a, b]$, and let $p_n(x)$ be the interpolation polynomial satisfying the interpolation condition

$$p_n(x_i) = f(x_i), \quad i = 0, 1, \dots, n.$$

We will then use $\int_a^b p_n(x) dx$ as an approximation to $\int_a^b f(x) dx$. By using the Lagrange form of the polynomial

$$p_n(x) = \sum_{i=0}^n f(x_i) \ell_i(x)$$

with the cardinal functions $\ell_i(x)$ given by

$$\ell_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j},$$

the following quadrature formula is obtained

$$\begin{aligned} I[f](a, b) &\approx Q[f](a, b) = \int_a^b p_n(x) \, dx \\ &= \sum_{i=0}^n f(x_i) \int_a^b \ell_i(x) \, dx = \sum_{i=0}^n w_i f(x_i) = Q(a, b), \end{aligned}$$

where the weights in the quadrature are simply the integral of the cardinal functions over the interval

$$w_i = \int_a^b \ell_i(x) \, dx \quad \text{for } i = 0, \dots, n.$$

Let us derive three schemes for integration over the interval $[0, 1]$, which we will finally apply to the integral

$$I(0, 1) = \int_0^1 \cos\left(\frac{\pi}{2}x\right) \, dx = \frac{2}{\pi} = 0.636619\dots$$

3.1 Example 2: The trapezoidal rule revisited

Let $x_0 = 0$ and $x_1 = 1$. The cardinal functions and thus the weights are given by

$$\begin{aligned} \ell_0(x) &= 1 - x, & w_0 &= \int_0^1 (1 - x) \, dx = 1/2 \\ \ell_1(x) &= x, & w_1 &= \int_0^1 x \, dx = 1/2 \end{aligned}$$

and the corresponding quadrature rule is the trapezoidal rule (usually denoted by T) recalled in Example 1: Quadrature rules from previous math courses with $[a, b] = [0, 1]$:

$$T(0, 1) = \frac{1}{2} [f(0) + f(1)].$$

3.2 Example 3: Gauß-Legendre quadrature for $n = 2$

Let $x_0 = 1/2 + \sqrt{3}/6$ and $x_1 = 1/2 - \sqrt{3}/6$. Then

$$\begin{aligned} \ell_0(x) &= -\sqrt{3}x + \frac{1 + \sqrt{3}}{2}, & w_0 &= \int_0^1 \ell_0(x) \, dx = 1/2, \\ \ell_1(x) &= \sqrt{3}x + \frac{1 - \sqrt{3}}{2}, & w_1 &= \int_0^1 \ell_1(x) \, dx = 1/2. \end{aligned}$$

The quadrature rule is

$$Q(0,1) = \frac{1}{2} \left[f\left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right) + f\left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right) \right].$$

3.3 Example 4: Simpson's rule revisited

We construct Simpson's rule on the interval $[0,1]$ by choosing the nodes $x_0 = 0$, $x_1 = 0.5$ and $x_2 = 1$. The corresponding cardinal functions are

$$\ell_0 = 2(x - 0.5)(x - 1) \quad \ell_1(x) = 4x(1 - x) \quad \ell_2(x) = 2x(x - 0.5)$$

which gives the weights

$$w_0 = \int_0^1 \ell_0(x) dx = \frac{1}{6}, \quad w_1 = \int_0^1 \ell_1(x) dx = \frac{4}{6}, \quad w_2 = \int_0^1 \ell_2(x) dx = \frac{1}{6}$$

such that

$$\int_0^1 f(x) dx \approx \int_0^1 p_2(x) dx = \sum_{i=0}^2 w_i f(x_i) = \frac{1}{6} [f(0) + 4f(0.5) + f(1)].$$

3.4 Exercise 1: Accuracy of some quadrature rules

Use the QR function

```
[3]: def QR(f, xq, wq):
    """ Computes an approximation of the integral f
    for a given quadrature rule.

    Input:
    f: integrand
    xq: quadrature nodes
    wq: quadrature weights
    """
    n = len(xq)
    if (n != len(wq)):
        raise RuntimeError("Error: Need same number of quadrature nodes and_
        weights!")
    return np.array(wq)@f(np.array(xq))
```

to compute an approximate value of integral

$$I(0,1) = \int_0^1 \cos\left(\frac{\pi}{2}x\right) dx = \frac{2}{\pi} = 0.636619\dots$$

using the quadrature rules from Example Example 2: The trapezoidal rule revisited-Example 4: Simpson's rule revisited. Tabulate the corresponding quadrature errors $I(0,1) - Q(0,1)$.

```
[4]: # Insert your code here.
```

Hint. You can start with (fill in values for any ...)

```
[ ]: # Define function
def f(x):
    return ...

# Exact integral
int_f = ...

# Trapezoidal rule
xq = [...]
wq = [..., ...]

qr_f = QR(f, xq, wq)
print("Q[f] = {}".format(qr_f))
print("I[f] - Q[f] = {:.10e}".format(int_f - qr_f))
```

3.4.1 Remarks

We observe that with the same number of quadrature points, the Gauß-Legendre quadrature gives a much more accurate answer than the trapezoidal rule. So the choice of nodes clearly matters. Simpson's rule gives very similar results to Gauß-Legendre quadrature, but it uses 3 instead of 2 quadrature nodes.

4 Degree of exactness and an estimate of the quadrature error

Motivated by the previous examples, we now take a closer look at how to assess the quality of a method. We start with the following definition.

4.1 Definition 1: The degree of exactness

A numerical quadrature has degree of exactness d if $Q[p](a, b) = I[p](a, b)$ for all $p \in \mathbb{P}_d$ and there is at least one $p \in \mathbb{P}_{d+1}$ such that $Q[p](a, b) \neq I[p](a, b)$.

Since both integrals and quadratures are linear in the integrand f , the degree of exactness is d if

$$\begin{aligned} I[x^j](a, b) &= Q[x^j](a, b), & j = 0, 1, \dots, d, \\ I[x^{d+1}](a, b) &\neq Q[x^{d+1}](a, b). \end{aligned}$$

Observation: All quadratures constructed from Lagrange interpolation polynomials in $n + 1$ distinct nodes will automatically have a degree of exactness of at least n . This follows immediately from the fact that the interpolation polynomial $p_n \in \mathbb{P}_n$ of any polynomial $q \in \mathbb{P}_n$ is just the original polynomial q itself.

4.2 Exercise 2: Degree of exactness for some quadrature rules

- Show that the trapezoidal and midpoint rule from Example Example 1: Quadrature rules from previous math courses is of precision 1
- Show that the Gauß-Legendre quadrature for 2 points from Example Example 3: Gauß-Legendre quadrature for $n=2$ is of precision 3.
- Show that Simpson's rule from Example Example 4: Simpson's rule revisited is also of precision 3.

[]: # Insert your code here.

Hint. You can do this either using pen and paper (boring!) or numerically (more fun!), using the code from Example Exercise 1: Accuracy of some quadrature rules

Solution.

4.3 Error estimates

4.4 Theorem 1: Error estimate for quadrature rule with degree of exactness n

Assume that $f \in C^{n+1}(a, b)$ and let $Q[\cdot](\{x_i\}_{i=0}^n, \{w_i\}_{i=0}^n)$ be a quadrature rule which has degree of exactness n . Then the quadrature error $|I[f] - Q[f]|$ can be estimated by

$$|I[f] - Q[f]| \leq \frac{M}{(n+1)!} \int_a^b \prod_{i=0}^n |x - x_i| dx$$

where $M = \max_{\xi \in [a, b]} |f^{n+1}(\xi)|$.

Proof. Let $p_n \in \mathbb{P}_n$ be the interpolation polynomial satisfying $p_n(x_i) = f(x_i)$ for $i = 0, \dots, n$. Thanks to the error estimate for the interpolation error, we know that

$$f(x) - p_n(x) = \frac{f^{n+1}(\xi(x))}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

for some $\xi(x) \in (a, b)$. Since $Q(a, b)$ has degree of exactness n we have $I[p_n] = Q[p_n] = Q[f]$ and thus

$$\begin{aligned} |I[f] - Q[f]| &= |I[f] - I[p_n]| \leq \int_a^b |f(x) - p_n(x)| dx \\ &= \int_a^b \left| \frac{f^{n+1}(\xi(x))}{(n+1)!} \prod_{k=0}^n (x - x_k) \right| dx \leq \frac{M}{(n+1)!} \int_a^b \prod_{k=0}^n |x - x_k| dx, \end{aligned}$$

which concludes the proof.

The previous theorem often The advantage of the previous theorem is that it is easy to prove. On downside is that the provided estimate can be rather crude, and often sharper estimates can be established. We give two examples here of some sharper estimates (but without proof).

4.5 Theorem 2: Error estimate for the trapezoidal rule

For the trapezoidal rule, there is a $\zeta \in (a, b)$ such that

$$I[f] - Q[f] = \frac{(b-a)^3}{12} f''(\zeta).$$

4.6 Theorem 3: Error estimate for Simpson's rule

For Simpson's rule, there is a $\zeta \in (a, b)$ such that

$$I[f] - Q[f] = -\frac{(b-a)^5}{2880} f^{(4)}(\zeta).$$