Comparison of Advanced Quadrature Techniques

William Wang

Rutgers University — Math 373 Spring 2021

Background

Numerical integration, more-or-less synonymous with *quadrature*, is the process of determining the area of a particular figure or region. While there are many different techniques, numerical integration is thought to originate from the ancient Greeks, who understood determination of the area of a figure as the process of geometrically constructing a square having the same area, thus the name *quadrature* for this process[1].

There are many reasons for the utilization of numerical integration techniques, but usually the reason is one of the following:

1. A function f(x) is given or known, but it may be tedious, difficult, computationally expensive, etc. to find an explicit antiderivative.
2. A function f(x) is given or known, but the antiderivative of f(x) itself may be impossible to written in terms of elementary functions.
3. The area between a function f(x) and the abscissa (x-axis) may be of some interest, but the value of f(x) is only known at particular points, possibly obtained through some sampling method.

Hence, numerical integration techniques are very useful, and may be used in various areas of mathematical interest including, but not limited to: geometry and differential equations—as well as physics, engineering, and even the life sciences and humanities[2].

Many numerical integration techniques exist, but I will only be discussing three advanced numerical integration techniques over two dimensions: Gaussian quadrature, Romberg's method, and Tanh-Sinh quadrature. I will also demonstrate the application of Gaussian quadrature and Romberg's method to approximate definite integrals for various given functions, but the application of these techniques will not be discussed over 3-dimensions (cubature) and higher dimensions.

Description of the Methods

Gaussian Quadrature

Starting with the oldest technique of the three, an n-point Gaussian quadrature rule is a numerical integration technique which, in contrast to Newton-Cotes formulas which use values of a function at equally spaced points, utilize points of evaluation chosen in an optimal way. Gaussian quadrature rules are stated as

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

where the coefficients (weights) w_i and the points x_i are chosen to minimize the error in (1).

To minimize the error in (1), we assume that the best choices of these values produce the exact result for polynomials of the highest degree, that is, the choice that gives the greatest degree of precision[3]. We have n coefficients and n points to choose, so in total we have $2n$ parameters to choose.Thus, the Gaussian quadrature rule with the proper choices of coefficients and points will be exact for polynomials of degree $2n-1$ or less.

However, choosing the optimal parameters may be computationally expensive, and may not be feasible when n is large. The example below will demonstrate this.

Example (Adapted from *Burden, Faires. Numerical Analysis. 10E*):
Suppose we want to want to determine w_1, w_2, x_1, and x_2 such that

$$\int_{-1}^{1} f(x)\,dx \approx c_1 f(x_1) + c_2 f(x_2)$$

gives the exact result whenever $f(x)$ is a polynomial of degree $2(2) - 1 = 3$ or less, that is, when

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3,$$

for some constants a_0, a_1, a_2, and a_3. First we can note that

$$\int f(x)\,dx = \int (a_0 + a_1 x + a_2 x^2 + a_3 x^3)\,dx = a_0 \int 1\,dx + a_1 \int x\,dx + a_2 \int x^2\,dx + a_3 \int x^3\,dx$$

Hence, we have a system of four equations:

$$c_1 + c_2 = \int_{-1}^{1} 1\,dx = 2$$

$$c_1 x_1 + c_2 x_2 = \int_{-1}^{1} x\,dx = 0$$

$$c_1 x_1^2 + c_2 x_2^2 = \int_{-1}^{1} x^2\,dx = \frac{2}{3}$$

$$c_1 x_1^3 + c_2 x_2^3 = \int_{-1}^{1} x^3\,dx = 0$$

By solving the system of equations, it turns out that $c_1 = c_2 = 1$, $x_1 = -\frac{\sqrt{3}}{3}$, and $x_2 = \frac{\sqrt{3}}{3}$. This gives the 2-point Gaussian quadrature rule

$$\int_{-1}^{1} f(x)\,dx \approx f(-\frac{\sqrt{3}}{3}) + f(\frac{\sqrt{3}}{3})$$

which is exact for all polynomials of degree 3 or less. As you can see, it may be somewhat difficult to compute the optimal points x_i and weights c_i, and the computational difficulty of calculating them increases as n increases.

There are also other types of Gaussian quadrature rules, but their performance and usage may be constrained to additional or different conditions.

Romberg's Method

Next, Romberg's method—a Newton-Cotes formula— essentially involves approximating a definite integral on an interval $[a, b]$ at equally spaced points by applying Richardson's extrapolation on the Trapezoid or Midpoint rule[3]. For our purposes, we will be using Richardon's extrapolation on the Composite Trapezoidal rule, because the error term of the Composite Trapezoidal rule on some interval $[a, b]$ can be written as

$$K_1 h^2 + K_2 h^4 + K_3 h^6 + ...$$

where h is the step size and each K_i is some constant that only depends on $f^{(2i-1)}(a)$ and $f^{(2i-1)}(b)$. Since Richardson's extrapolation can be performed on any technique where the truncation error is of the form

$$\sum_{i=1}^{n-1} K_i h^{\alpha_i} + O(h^{2n})$$

where each K_i is some constant and when $\alpha_1 < ... < \alpha_n$, the Composite Trapezoidal rule is a suitable choice to perform Richardson's extrapolation on.

To approximate the integral $\int_a^b f(x)\,dx$, we use the Composite Trapezoidal rule with n = 1, 2, 4, 8, ... and denote the resulting approximations, respectively, by $R_{1,1}$, $R_{2,1}$, $R_{3,1}$, and so on. Then, we can apply extrapolation to obtain the approximations $R_{2,2}$, $R_{3,2}$, $R_{4,2}$, and so on by

$$R_{k,2} = R_{k,1} + \frac{1}{3}(R_{k,1} - R_{k-1,1}), \quad k = 2, 3, ...$$

Then, we can apply extrapolation to obtain the approximations $R_{3,3}$, $R_{4,3}$, $R_{5,3}$, and so on by

$$R_{k,3} = R_{k,2} + \frac{1}{15}(R_{k,2} - R_{k-1,2}), \quad k = 3, 4, ...$$

In general, we will iteratively apply extrapolation to get $R_{k,j-1}$ approximations, then use those results to obtain the approximations

$$R_{k,j} = R_{k,j-2} + \frac{1}{4^{j-1}-1}(R_{k,j-1} - R_{k-1,j-1}), \quad k = j, j+1, ...$$

Example (Adapted from *Burden, Faires. Numerical Analysis. 10E*):
Use the Composite Trapezoidal rule to find approximations to $\int_0^\pi \sin x\, dx$ with n = 1, 2, 4, 8, and 16. Then perform Romberg integration on the results.

Composite Trapezoidal Rule Approximations:

$$R_{1,1} = \frac{\pi}{2}[\sin 0 + \sin \pi)] = 0$$

$$R_{2,1} = \frac{\pi}{4}[\sin 0 + 2\sin \frac{\pi}{2} + \sin \pi] = \frac{\pi}{2} \approx 1.5708$$

$$R_{3,1} = \frac{\pi}{8}[\sin 0 + 2(\sin \frac{\pi}{4} + \sin \frac{\pi}{2} + \sin \frac{3\pi}{4}) + \sin \pi] = \frac{\pi}{4}(1+\sqrt{2}) \approx 1.8961$$

$$R_{4,1} = \frac{\pi}{16}[\sin 0 + 2(\sum_{i=1}^{7} \sin \frac{i\pi}{8}) + \sin \pi] \approx 1.9742$$

$$R_{5,1} = \frac{\pi}{32}[\sin 0 + 2(\sum_{i=1}^{15} \sin \frac{i\pi}{16}) + \sin \pi] \approx 1.9936$$

Then, use Richardson's extrapolation:

$$R_{2,2} = R_{2,1} + \frac{1}{3}(R_{2,1} - R_{1,1}) \approx 1.5708 + \frac{1}{3}(1.5708 - 0) = \frac{4}{3}(1.5708) = 2.0944$$

$$R_{3,2} = R_{3,1} + \frac{1}{3}(R_{3,1} - R_{2,1}) \approx 1.8961 + \frac{1}{3}(1.8961 - 1.5708) \approx 2.0045$$

$$R_{4,2} = R_{4,1} + \frac{1}{3}(R_{4,1} - R_{3,1}) \approx 1.9742 + \frac{1}{3}(1.9742 - 1.8961) \approx 2.0002$$

$$R_{5,2} = R_{5,1} + \frac{1}{3}(R_{5,1} - R_{4,1}) = 1.9936 + \frac{1}{3}(1.9936 - 1.9742) \approx 2.0000$$

$$R_{3,3} = R_{3,2} + \frac{1}{15}(R_{3,2} - R_{2,2}) \approx 2.0045 + \frac{1}{15}(2.0045 - 2.0944) \approx 1.9985$$

$$R_{4,3} = R_{4,2} + \frac{1}{15}(R_{4,2} - R_{3,2}) \approx 2.0002 + \frac{1}{15}(2.0002 - 2.0045) \approx 1.9999$$

$$R_{5,3} = R_{5,2} + \frac{1}{15}(R_{5,2} - R_{4,2}) = 2.0000 + \frac{1}{15}(2.0000 - 2.0002) \approx 2.0000$$

$$R_{4,4} = R_{4,3} + \frac{1}{63}(R_{4,3} - R_{3,3}) = 1.9999 + \frac{1}{63}(1.9999 - 1.9985) \approx 1.9999$$

$$R_{5,4} = R_{5,3} + \frac{1}{63}(R_{5,3} - R_{4,3}) = 2.0000 + \frac{1}{63}(2.0000 - 1.9999) \approx 2.0000$$

$$R_{5,5} = R_{5,4} + \frac{1}{255}(R_{5,4} - R_{4,4}) \approx 2.0000 + \frac{1}{255}(2.0000 - 1.9999) \approx 2.0000$$

Table:

0				
1.5708	2.0944			
1.8961	2.0045	1.9985		
1.9742	2.0002	1.9999	1.9999	
1.9936	2.0000	2.0000	2.0000	2.0000

This particular example is also demonstrated in ex_romberg.m.

Tanh-Sinh Quadrature

Lastly, Tanh-Sinh quadrature, according to Bailey[4], is "*the fastest currently known high-precision quadrature scheme, particularly when one counts for abscissas and weights*". It is also known as the Double-Exponential (DE) formula.

Essentially, it first involves a change of variables $x = \tanh(\frac{1}{2}\sinh(t))$, where $f(x)$ is the integrand, transforming an integral on the interval $x \in [-1, 1]$ to the interval $t \in (-\infty, \infty)$. The quadrature rule is stated as

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=-\infty}^{\infty} w_i f(x_i) \tag{2}$$

where h is the step size, the weight w_i is

$$w_i = \frac{\frac{1}{2}h\pi\cosh(ih)}{\cosh^2(\frac{1}{2}\pi\sinh(ih))} \tag{3}$$

and the node x_i is

$$x_i = \tanh(\frac{1}{2}\pi\sinh(ih)) \tag{4}$$

By performing this change of variables, it conveniently makes the approximation resistant to endpoint behavior and in many cases causes the the approximation to quickly converge[5].

Code Implementation

- **gauss2p.m**: Outputs an approximation of the area under $f(x)$ on the interval $[-1, 1]$ using the 2-point Gaussian quadrature rule.
- **ex_gauss2p.m**: Demonstrates the 2-point Gaussian quadrature rule for the function $f(x) = x^2$ on the interval $[-1, 1]$ using gauss2p.m
- **romberg.m**: Outputs an approximation of the area under $f(x)$ on the interval $[a, b]$ using Romberg's method (Richardson's extrapolation on Composite Trapezoidal rule) with n = $1, 2, 4, ...$
- **ex_romberg.m**: Demonstrates Romberg's method for the function $f(x) = \sin(x)$ on the interval $[0, \pi]$ using romberg.m

The file **main.m** executes **gauss2p.m** and **romberg.m** to display the examples in the next section.

Testing and Results

I will test and compare the performance of Gaussian quadrature and Romberg's method for various functions, in which the absolute error of the approximation versus the actual answer will be measured.

In the error estimate portion of the MATLAB code for each example, the actual answer is represented as **integral(f, a, b)**, where **integral** is the MATLAB built in numerical integration function, f is the function, and a and b are the left- and right-endpoints, respectively. If known, replacing **integral(f, a, b)** with the actual, explicit answer may give a more realistic (better) error estimate, since the implementation details of the **integral** function are not known and may not be completely accurate itself.

In the following examples, I just used

Example 1: $f(x) = \sin x$

For the first example, I will test the performance of the numerical integration techniques in approximating the area under $f(x) = \sin x$ on the interval $[-1, 1]$. The exact answer to $\int_{-1}^{1} \sin(x)\,dx$ is 0.

2-point Gaussian quadrature rule:

```
Approximation of area under f in interval [-1,1] = 0
Approximate error of Gaussian quadrature approximation: 6.245e-17
```

The approximation of the area under $f(x) = \sin x$ on the interval $[-1, 1]$ using the 2-point Gaussian quadrature rule is very accurate, with approximately 16 decimal places of accuracy—Gaussian quadrature seems to perform very well for this example!

Romberg's Method:

```
matrix =

   1.0e-15 *

         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
    0.0278    0.0370    0.0395    0.0401         0         0
   -0.0416   -0.0648   -0.0715   -0.0733   -0.0738         0
    0.0902    0.1342    0.1474    0.1509    0.1518    0.1520

Approximation of area under f in interval [-1,1] = 1.51988e-16
Approximate error of Romberg's method approximation: 2.14438e-16
```

The approximation of the area under $f(x) = \sin x$ on the interval $[-1, 1]$ using Romberg's method also seems to be very accurate, with approximately 15 decimal places of accuracy.

For this example, the performance of the 2-point Gaussian quadrature rule seems to slightly edge out that of Romberg's method, with 1 additional decimal place of accuracy in the approximate error.

Example 2: $f(x) = \cos x$

For the second example, I will test the performance of the numerical integration techniques in approximating the area under $f(x) = \cos x$ on the interval $[-1, 1]$. The exact answer to $\int_{-1}^{1} \cos x \, dx$ is $\sin(1) - \sin(-1) \approx 1.683$.

2-point Gaussian quadrature rule:

```
Approximation of area under f in interval [-1,1] = 1.67582
Approximate error of Gaussian quadrature approximation: 0.00711831
```

The approximation of the area under $f(x) = \cos x$ on the interval $[-1, 1]$ using the 2-point Gaussian quadrature rule seems to perform well for this example, giving us approximately 2 decimal places of accuracy.

Romberg's Method:

```
matrix =

    1.0806         0         0         0         0         0
    1.5403    1.6935         0         0         0         0
    1.6477    1.6835    1.6829         0         0         0
    1.6742    1.6830    1.6829    1.6829         0         0
    1.6808    1.6829    1.6829    1.6829    1.6829         0
    1.6824    1.6829    1.6829    1.6829    1.6829    1.6829

Approximation of area under f in interval [-1,1] = 1.68294
Approximate error of Romberg's method approximation: 4.21885e-15
```

The approximation of the area under $f(x) = \cos x$ on the interval $[-1, 1]$ using Romberg's method seems to be very accurate, with approximately 14 decimal places of accuracy!

For this example, the performance of Romberg's method seems to be much better than that of the 2-point Guassian quadrature rule, with 14 decimal places of accuracy compared to 2.

Example 3: $f(x) = x^2$

For the third example, I will test the performance of the numerical integration techniques in approximating the area under $f(x) = x^2$ on the interval $[-1, 1]$. The exact answer to $\int_{-1}^{1} x^2 \, dx$ is $\frac{2}{3}$.

2-point Gaussian quadrature rule:

```
Approximation of area under f in interval [-1,1] = 0.666667
Approximate error of Gaussian quadrature approximation: 1.11022e-16
```

The 2-point Gaussian quadrature rule approximation of the area under $f(x) = x^3+x^2+x+1$ on the interval $[-1, 1]$ seems to be very good, providing us with about 15 decimal places of accuracy. This is consistent with the theoretical performance described under Gaussian quadrature in the Description of the Methods—the theoretical Gaussian quadrature rule approximation is exact—meaning for this example, the only source of error is from MATLAB's finite precision.

Romberg's Method:

```
matrix =

    2.0000         0         0         0         0         0
    1.0000    0.6667         0         0         0         0
    0.7500    0.6667    0.6667         0         0         0
    0.6875    0.6667    0.6667    0.6667         0         0
    0.6719    0.6667    0.6667    0.6667    0.6667         0
    0.6680    0.6667    0.6667    0.6667    0.6667    0.6667

Approximation of area under f in interval [-1,1] = 0.666667
Approximate error of Romberg's method approximation: 1.11022e-16
```

The approximation using Romberg's method for the area under $f(x) = x^2$ on the interval $[-1, 1]$ also seems to be very good, with 15 decimal places of accuracy.

Interestingly for this example, the approximations using Romberg's method and the 2-point Gaussian quadrature rule, and the approximate errors of both techniques, seems to be exactly the same.

Example 4: $f(x) = x^3 + x^2 + x + 1$

For the third example, I will test the performance of the numerical integration techniques in approximating the area under $f(x) = x^3+x^2+x+1$ on the interval $[-1, 1]$. The exact answer to $\int_{-1}^{1} x^3+x^2+x+1\,dx$ is $\frac{8}{3}$.

2-point Gaussian quadrature rule:

```
Approximation of area under f in interval [-1,1] = 2.66667
Approximate error of Gaussian quadrature approximation: 4.44089e-16
```

The 2-point Gaussian quadrature rule approximation of the area under $f(x) = x^3 + x^2 + x + 1$ on the interval $[-1, 1]$ seems to very good, providing us with about 15 decimal places of accuracy. Likewise as in Example 3, the theoretical error of the 2-point Gaussian quadrature rule for this function is 0, so the only source of error for this approximation should be from MATLAB's finite precision.

Romberg's Method:

```
matrix =

    4.0000         0         0         0         0         0
    3.0000    2.6667         0         0         0         0
    2.7500    2.6667    2.6667         0         0         0
    2.6875    2.6667    2.6667    2.6667         0         0
    2.6719    2.6667    2.6667    2.6667    2.6667         0
    2.6680    2.6667    2.6667    2.6667    2.6667    2.6667

Approximation of area under f in interval [-1,1] = 2.66667
Approximate error of Romberg's method approximation: 0
```

The approximation using Romberg's method for the area under $f(x) = x^3 + x^2 + x + 1$ on the interval $[-1, 1]$ seems to be a perfect estimate, with an approximate error of 0.

Comparing the approximate errors of 0 and 4.44089×10^{-16} for Romberg's method and the 2-point Gaussian quadrature rule respectively, Romberg's method seems to perform better for this example, although the 2-point Gaussian quadrature rule still performs extremely well.

Discussion

Comparing the two techniques accross all four examples, it can be seen that for certain functions, the 2-point Gaussian quadrature rule may perform better than Romberg's method, for certain functions the opposite, and for others that the performance of the two are equal. Neither techniques are completely accurate for all functions, but in our examples, both seemed to be very accurate.

References

[1] *Numerical integration* (2021). Retrieved March 28, 2021 from `https://en.wikipedia.org/wiki/Numerical_integration`

[2] *Numerical analysis* (2021). Retrieved April 7, 2021 from `https://en.wikipedia.org/wiki/Numerical_analysis`

[3] R. Burden, D. Faires, and A. Burden. *Numerical Analysis 10ed.* (2014). Cengage Learning

[4] D. Bailey. *Tanh-Sinh High Precision Quadrature* (2006). Retrieved March 27, 2021 from `https://www.davidhbailey.com/dhbpapers/dhb-tanh-sinh.pdf`

[5] *Tanh-sinh quadrature* (2021). Retrieved March 28, 2021 from `https://en.wikipedia.org/wiki/Tanh-sinh_quadrature`