Introduction

In this course we will be concerned with algorithms for performing three common Calculus tasks[1]:

1. Computation of integrals (quadrature).

2. Finding zeros of functions (root finding).

3. Solution of initial value problems.

However, Math 110 is not merely a course in computational methods—it is a course in Numerical Analysis! Therefore our main focus will be on the conceptual framework for numerical methods rather than the methods themselves.

By 'conceptual framework' I specifically mean *interpolation theory.* This, roughly speaking, is the theory about approximating complicated functions with simple functions. To see why interpolation is key to Numerical Analysis, consider the problem of one-dimensional quadrature: approximate $\int_a^b f(x)\,dx$. If we can approximate f with a polynomial p on $[a, b]$ then it is easy to approximate the integral of f, namely:

$$\int_a^b f(x)\,dx \approx \int_a^b p(x)\,dx.$$

The integral of a polynomial is, of course, trivial to compute. What is not trivial, however, is understanding how to properly derive the polynomial approximation *that ensures given accuracy*—that is what interpolation theory is largely about.

By the way, after three semesters of Calculus, the preceding paragraph should have rung some bells. Remember approximating functions of one variable with Taylor polynomials? In our new parlance, this was *Taylor interpolation.* If you happen to understand Taylor theory to the extent where you can derive the remainder of Taylor approximation, you may skip the following Calculus review. Otherwise, read on.

[1] If time allows, we will also talk about Numerical Linear Algebra. But no promises!

Rethinking Calculus 2

As a prerequisite for understanding interpolation, which is key to Numerical Analysis, you must be able to solve the following kind of problem from Calculus 2:

> Suppose we approximate $y = e^x$ on the interval $[0, 1]$ using a Taylor polynomial centered at $x = 0$. How large must the degree of the polynomial be in order to ensure that the absolute error does not exceed 10^{-4}?

Unfortunately, most students in Math 110 find this problem impossible in the beginning, regardless of their Math 53 experience. Now you may have your own theory as to why this may be acceptable. Perhaps you never used Taylor polynomials outside of Calculus, so you forgot about them. Or, perhaps, your Calculus 2 experience was rough and you never understood Taylor stuff in the first place. Be this as it may, you must be able to solve the problem and by 'solve' I mean *understand*. Getting a largely meaningless answer through mimicry is not going to cut it.

Forgetting elementary Calculus is a common and utterly depressing problem in advanced math classes. The common cure for it is the dreaded "calculus review" which we are not going to do. I am willing to bet that you have had your share of reviewing. And still you forget things, so reviewing obviously does not work in the long run. What you need to do is to *rethink* Calculus. This means three things: changing priorities, establishing connections, and distilling the essence.

When you studied Calculus you probably did not have a clear motivation for, say, Taylor theory. You did not think it was important and this was further confirmed outside of Calculus. As a result, your mind did not attach much significance to the topic and, consequently, it quickly slipped out of memory. If you continue to undervalue Taylor theory you will never understand Numerical Analysis. So change its priority to the highest possible level. After all, it is one of the most useful pieces of mathematics that you can learn as an undergraduate.

Now let us talk about connections. Human memory is, as you may know, highly associative. We remember things by establishing physical links in our brains. The more connections, the better we remember things. Conversely, it is hard to remember something if that something does not associate with anything else. You cannot and should not memorize Taylor's theorem on its own. It needs to be associated—connected to other theorems.

Finally, let me try to explain what I mean by distilling the essence. In one of the many Buddhist temples in Kyoto there is a famous painting of six persimmons dating to 13th century; it is displayed to the general public only during major holidays. The painting is just a few quick brush strokes, yet it is cherished by Buddhists and art experts alike. Muqi Fachang—the monk who created that painting—somehow managed to capture the *essence* of six persimmons lying on a table—something that countless imitations and high resolution digital photographs fail to do. This is how you should remember mathematics: in its simplest, most elemental form. In particular, in order to remember Taylor's theorem you need to see the essence of Taylor's theorem. Can you paint it with a few brush strokes?

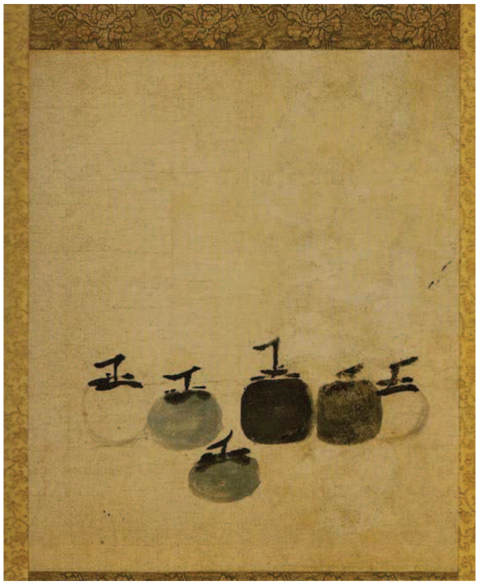

Figure 1: *Six persimmons*, by Muqi Fachang, ink on paper, 14.25 × 15 in, 13-th century

Value theorems

I hope that after reading the introductory paragraph you have reclassified Taylor's theorem as top priority. If so, let us work on establishing connections between Taylor's theorem and the more basic theorems of analysis. One of those theorems is the Fundamental Theorem of Calculus (FTC) which, I trust, you remember. You can use FTC and integration by parts to derive the Taylor remainder in integral form. However, in order to convert the remainder to the more useful Lagrange form, you need a generalization of the Mean Value Theorem (MVT) which in turn requires Intermediate Value Theorem (IVT) and Extreme Value Theorem (EVT). So we should talk about all three "value" theorems starting with IVT and EVT.

Theorem 1 (IVT)**.** *Let $f \in C[a,b]$. For any $\eta \in [f(a), f(b)]$ there exists $\xi \in [a,b]$ such that $f(\xi) = \eta$.*

We are not going to prove IVT at this point. Instead, let us talk about what it means, why it is nontrivial, and how to remember it. By the way, did you find the format in which I presented IVT [slightly] off-putting? I chose the formal presentation intentionally because it gives us an opportunity to practice active learning. Numerical Analysis, and mathematics in general, is replete with statements such as the one above. You need to learn to decipher mathematical code.

Let us start with the first sentence: "Let $f \in C[a,b]$." This is code for: "Let f be a continuous function defined on a closed interval $[a,b]$." Note that the letter C denotes a *class* of functions—continuous functions; the symbol $\in$ means 'belongs to' or 'element of a set.' Notice also how much more concise the formal statement is compared to its expanded version.

The next sentence asserts the existence of a number ξ in $[a,b]$ such that $f(\xi) = \eta$ as long as the number η lies in the interval $[f(a), f(b)]$. What we are trying to say, using strange Greek letters, is that all of the values between $f(a)$ and $f(b)$ are in the range of f—the function assumes all of these values somewhere in $[a,b]$. So, why not say that? There are several reasons which necessitate Greek letters and other obscurities. One reason is precision. Mathematical statements must be absolutely precise. Common language, on the other hand, is often imprecise and ambiguous. Consider how ambiguous the following statement is (from the mathematical point of view): "Six monks painted six persimmons." Did each monk paint six persimmons? Or did they paint one persimmon each? Another reason for formalism—not always the best one—is brevity. Notice again how much shorter the formal version is compared to the "human" version. I will not use brevity as a pretext for speaking in mathematical code and I will try to avoid being completely formal. Yet, on occasion, we will need to confront formal statements. So, practice!

Now why is IVT nontrivial? Does it not state the obvious? *Intuitively* it does. Consider, however, the formal definition of continuity. In Calculus, the function f is defined to be continuous at ξ if the limit equals the value: $\lim_{x \to \xi} f(x) = f(\xi)$. Try to come up with a formal proof of IVT using the Calculus definition of continuity. Not so trivial now! In fact, IVT is not trivial at all. Its essence has to do with the topology of the real line which we may discuss at some point. In the meantime, try to capture the essence of IVT in a few brush strokes. A deep meditation might help.

Theorem 2 (EVT). *On a closed interval a continuous function assumes its maximum and minimum values.*

The statement is devoid of strange symbolism and does not need to be decoded. Yet, again, it somehow seems trite. Do not maxima and minima always exist? In Calculus I they do and way too much attention is devoted to the mechanical process of finding and testing critical points. In order to appreciate EVT, abstract from the Calculus process for finding extrema. EVT tells you that two things can potentially go wrong when you optimize functions: there may be a discontinuity or the domain may not be closed. For instance, consider $y = x^2$ on $(0, 1]$. This function does not have a minimum because the interval is not closed. No matter which value in $(0, 1]$ is chosen, one can move closer to zero which reduces the value of the function.

Theorem 3 (MVT). *Let $f \in C[a, b]$. There exists $\xi \in [a, b]$ such that:*

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

This is one way to state the Mean Value Theorem—in *integral form.* Equivalently, we can use FTC to rewrite the above equation as

$$\frac{F(b) - F(a)}{b - a} = F'(\xi).$$

This is the same MVT but stated in differential form. However, the most useful version of MVT (from the point of view of Numerical Analysis) is the one involving a weight. We will call that Generalized Mean Value Theorem (GMVT).

Theorem 4 (GMVT)**.** *Let $f \in C[a,b]$ and suppose that w is a function that maintains constant sign on $[a,b]$. There exists $\xi \in [a,b]$ such that:*

$$\int_a^b f(x)\,w(x)\,dx = f(\xi)\int_a^b w(x)\,dx.$$

Proof. For definitiveness, assume that w is positive. The case where w is negative is completely analogous.

Since f is continuous, it attains its maximum M and its minimum m on $[a,b]$ by EVT. Therefore we have the double inequality:

$$\int_a^b m\,w(x)\,dx \le \int_a^b f(x)\,w(x)\,dx \le \int_a^b M\,w(x)\,dx.$$

Consequently,

$$m \le \frac{\int_a^b f(x)\,w(x)\,dx}{\int_a^b w(x)\,dx} \le M.$$

Now IVT implies that there exists $\xi \in [a,b]$ such that

$$\frac{\int_a^b f(x)\,w(x)\,dx}{\int_a^b w(x)\,dx} = f(\xi).$$

This proves the theorem. □

As you will see in the next section, GMVT is very useful for converting error estimates from integral form into simpler derivative form. Indeed, we will often use GMVT for that purpose. In contrast, IVT and EVT are rarely used directly. Nevertheless do not discard EVT and IVT. Doing that ruins the integrity of Calculus which leads to disassociation and memory loss!

We conclude this section with a few remarks about the weight in GMVT. We do not mention it in the statement of the theorem, yet it is implicit that w should be *integrable* on $[a,b]$. What that means is a subject of a long conversation. For now, think of integrability as a condition that rules out "bad" singularities like $\frac{1}{x}$. We hasten to add that unlike f the weight w is not required to be continuous. For example, w can have a jump discontinuity as long as it maintains constant sign. Finally, speaking of the constant sign, that assumption can be somewhat relaxed. For instance, it is OK if w is zero somewhere or even everywhere inside $[a,b]$. What we cannot allow is for w to change sign. Think about the things that go wrong in the proof of GMVT if that happens.

Taylor's theorem

We are ready to discuss Taylor's theorem which we now view as a statement about polynomial interpolation. Quite generally, an interpolating polynomial matches—interpolates—the behavior of the function in some way. Taylor polynomials match the values of the first several derivatives of the function at the center of expansion. In the statement of the theorem we use the symbol C^{n+1} to indicate that a function can be differentiated $(n+1)$ times. This symbol also means that the last $(n+1)$-st derivative is continuous.

Theorem 5 (Taylor)**.** *Let $f \in C^{n+1}[a,b]$. Then for any x and $x_0 \in (a,b)$*

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}\,(x-x_0)^k + \frac{f^{(n+1)}(\xi)}{(n+1)!}\,(x-x_0)^{n+1},$$

where ξ lies between x_0 and x.

Proof. Using FTC, we can write

$$f(x) = f(x_0) + \int_{x_0}^{x} f'(t)\,dt.$$

This is a special form of Taylor's theorem valid if $f \in C^1[a,b]$. If $f \in C^2[a,b]$ we can use integration by parts to rewrite the above equation as:

$$f(x) = f(x_0) + f'(x_0)\,(x-x_0) + \int_{x_0}^{x} (x-t)\,f''(t)\,dt.$$

For $f \in C^{n+1}[a,b]$ repeated integration by parts leads to

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}\,(x-x_0)^k + \int_{x_0}^{x} \frac{(x-t)^n}{n!}\,f^{(n+1)}(t)\,dt.$$

This is Taylor's theorem but with remainder in integral form. To finish the proof we apply GMVT. Since $f \in C^{n+1}$ the derivative $f^{(n+1)}$ is continuous. Furthermore, the weight $w(t) = (x-t)^n/n!$ is of constant sign on $[x_0, x]$. Hence

$$\begin{aligned}\int_{x_0}^{x} \frac{(x-t)^n}{n!}\,f^{(n+1)}(t)\,dt &= f^{(n+1)}(\xi)\int_{x_0}^{x} \frac{(x-t)^n}{n!}\,dt \\ &= f^{(n+1)}(\xi)\,\frac{(x-x_0)^{n+1}}{(n+1)!}.\end{aligned}$$

This completes the proof. □

Let us return to the problem about approximating e^x on $[0, 1]$ with Taylor polynomials centered at zero. For the exponential, the Taylor polynomial of degree n at the origin is given by

$$T_n = \sum_{k=0}^{n} \frac{x^k}{k!}.$$

According to Taylor's theorem, the absolute error of n-th degree approximation is

$$|e^x - T_n(x)| = e^{\xi}\,\frac{|x|^{n+1}}{(n+1)!}$$

with $\xi \in [0, x]$. Since e^x is monotonely increasing, the worst possible case is when $x = \xi = 1$. Thus the error is bounded by $1/(n + 1)!$ and we need to choose n so that

$$\frac{1}{(n+1)!} < 10^{-4}.$$

This requires $n \geq 7$.

Lagrange interpolation

A Taylor polynomial matches, or interpolates, the values of several derivatives of a function at a single point. In contrast, the Lagrange polynomial interpolates the values of a function at several distinct points which we will call *nodes*.

Thus the Lagrange polynomial p interpolating f at nodes $x_1, \ldots, x_n$ is the polynomial satisfying

$$p(x_k) = f(x_k), \quad \text{for } k = 1, \ldots, n.$$

The degree of such a polynomial must necessarily be $(n - 1)$. In the special case of two nodes the Lagrange polynomial is just the familiar secant line:

$$p = \frac{f(x_2) - f(x_1)}{x_2 - x_1}\,(x - x_1) + f(x_1)$$

Lagrange polynomials of higher degree are a bit more complicated and will be discussed later. In the meantime, suppose we interpolate f at two distinct nodes. What is the error of interpolation?

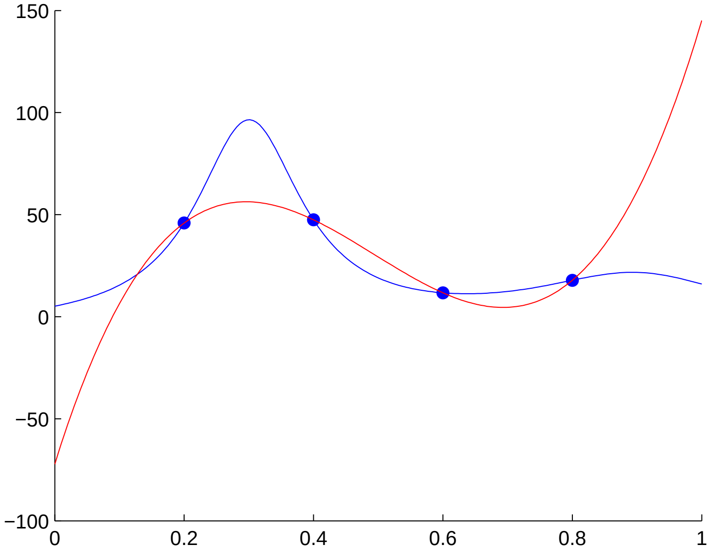

Figure 2: Lagrange cubic (red) interpolating `humps` (blue) at four equispaced nodes in $[0, 1]$

Remainder of linear interpolation

For notational convenience let us label the two nodes a and b ($b > a$) and let

$$R(x) = f(x) - p(x) = f(x) - f(a) - \frac{f(b) - f(a)}{b - a}(x - a).$$

This is the remainder of linear Lagrange interpolation. If we let b approach a we should get the remainder for (linear) Taylor interpolation. This suggests that there should be some similarity with Taylor theory. In particular, we should expect the second derivative to be involved and therefore we should assume that $f \in C^2$.

In order to find an expression for R, we first observe that it satisfies the following boundary value problem:

$$R'' = f'', \quad R(a) = R(b) = 0.$$

Boundary value problems are usually more difficult than initial value problems which you studied in Math 57. However in this particular case the solution can be derived using simple calculus as follows. Integrate the differential equation twice using a as the lower limit. This gives:

$$R = \int_a^x \left[\int_a^s f''(t)\,dt \right] ds + C_1 + C_2\,x.$$

From boundary conditions we infer that

$$\begin{aligned} C_1 + C_2\,a &= 0 \\ C_1 + C_2\,b &= -\int_a^b \left[\int_a^s f''(t)\,dt \right] ds \end{aligned}$$

Therefore

$$R = \int_a^x \left[\int_a^s f''(t)\,dt \right] ds - \frac{(x-a)}{b-a} \int_a^b \left[\int_a^s f''(t)\,dt \right] ds.$$

We now reverse the order of integration (as you did in Math 55) which gives:

$$R = \int_a^x (x-t)\,f''(t)\,dt - \frac{(x-a)}{b-a} \int_a^b (b-t)\,f''(t)\,dt$$

This can be put in the form

$$R = \int_a^b K(x,t)\,f''(t)\,dt$$

where the *kernel* K is given by:

$$K(x,t) = -\frac{1}{b-a} \begin{cases} (t-a)\,(b-x), & a < t \le x, \\ (x-a)\,(b-t), & x < t < b. \end{cases}$$

We have thus succeed in expressing the remainder of Lagrange interpolation as an integral. If you followed the discussion of Taylor's remainder then you know that the next step is to apply GMVT. Since $a \le x \le b$ the kernel K maintains constant (negative) sign on $[a,b]$ (sketch it!). Therefore, by GMVT:

$$R = f''(\xi) \int_a^b K(x,t)\,dt = \frac{1}{2} f''(\xi)(x-a)\,(x-b).$$

Notice that we recover Taylor's remainder if $b \to a$.

Remainder for Lagrange interpolation

Following the discussion of linear interpolation, it is easy to guess the general statement which we prove below.

Theorem 6 (Lagrange)**.** *Let $f \in C^n[a,b]$ and let p denote the interpolating polynomial with nodes $\{x_1, \dots, x_n\}$. The remainder of interpolation is*

$$f(x) - p(x) = \frac{f^{(n)}(\xi)}{n!} (x - x_1)(x - x_2) \dots (x - x_n).$$

Proof. Let

$$w(t) = \prod_{k=1}^{n} (t - x_k)$$

be the *nodal polynomial.* Fix x and introduce the following auxiliary function (of t):

$$g(t) = f(t) - p(t) - (f(x) - p(x)) \frac{w(t)}{w(x)}.$$

The function g is C^n (since f is). Also, since $f(x_k) = p(x_k)$ and $w(x_k) = 0$ we have

$$g(x_k) = 0, \quad k = 1, \dots, n.$$

Moreover, $g(x) = 0$, by construction. Thus g has $n+1$ zeros. It now follows from generalized Rolle's theorem that there is a ξ such that $g^{(n)}(\xi) = 0$. If you do not know [generalized] Rolle's theorem, recall MVT in differential form:

$$\frac{F(b) - F(a)}{b - a} = F'(\xi).$$

If $F(b) = F(a) = 0$ then $F'(\xi) = 0$: the derivative of a function with two zeros must vanish somewhere. If a function has three zeros, its first derivative has at least two zeros, so the second derivative must vanish somewhere. Generalized Rolle's theorem is simply induction of MVT: if a C^n function has $(n+1)$ zeros then its n-th derivative must vanish somewhere. Now the n-th derivative of the cleverly constructed g is:

$$g^{(n)}(\xi) = f^{(n)}(\xi) - (f(x) - p(x)) \frac{n!}{w(x)}.$$

Setting this expression to zero and solving for $f(x) - p(x)$ gives the statement of the theorem.

□

The proof of Theorem 6 is, objectively, sleek and requires seemingly less effort compared to solving a boundary value problem. However it is not constructive. One has to know what the remainder is and, further, have the ingenuity to come up with a useful auxiliary function. A lot of proofs in textbooks are sleek but not constructive. This is why few non-mathematicians bother to read mathematical textbooks.

Exercises

1. Derive the remainder of Lagrange interpolation for three nodes by solving an appropriate boundary value problem. Show that the formula is consistent with Taylor theory.

2. In `Matlab` one can find Lagrange polynomials using `polyfit` command. Write a script that plots Lagrange polynomials through N equispaced nodes $\{x_k = k/(N+1)\}$, $k = 1, ..., N$ for $f(x) = e^x$. What happens as N increases?

3. Repeat the previous exercise with `Matlab`'s `humps` function in place of the exponential. What are your observations?

6. Adaptive Quadrature

In MATLAB the function

$$f(x) = \left((x-3/10)^2 + \frac{1}{100}\right)^{-1} + \left(\left(x - \frac{9}{10}\right)^2 + 1/25\right)^{-1} - 6$$

is implemented as a routine called `humps`. It is often used for testing other routines, such as quadrature. Figure 8 shows the plot of `humps` on the interval $[0, 8]$.

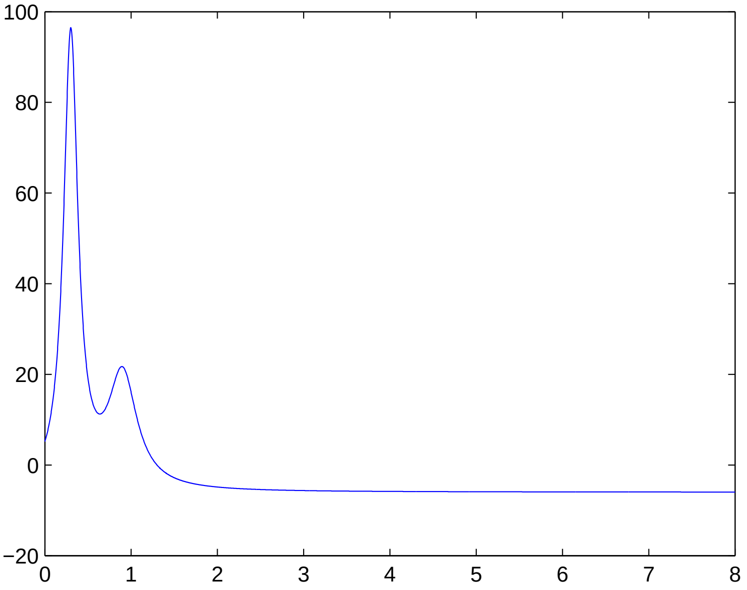

Figure 8. The plot of `humps` on $[0, 8]$

Notice that the function changes rapidly on [0,2] while on the rest of the interval its behavior is quite sedate: for $x > 3$ the function is nearly constant.

Suppose now that we want to use composite trapezoid rule to approximate the integral $\int_0^8 f(x)\,dx$ with a rather large tolerance 10^{-3}. If we use constant step size, we must subdivide the interval $[0, 8]$ into at least 578 subintervals. This already is a lot of functional evaluations for such a large tolerance. However things quickly become worse if we shrink tolerance. To achieve absolute error of 10^{-4} the minimum number of subdivisions must be 1826; whereas for 10^{-5} it is 5774. As the linear portion of Figure 9 implies, the cost of composite trapezoid rule obeys the asymptotic law:

$$\#\text{ of evaluations} \sim \text{tolerance}^{-1/2}, \quad \text{as tolerance} \to 0$$

This suggests that composite trapezoid rule *with even subdivisions* is not computationally effective.

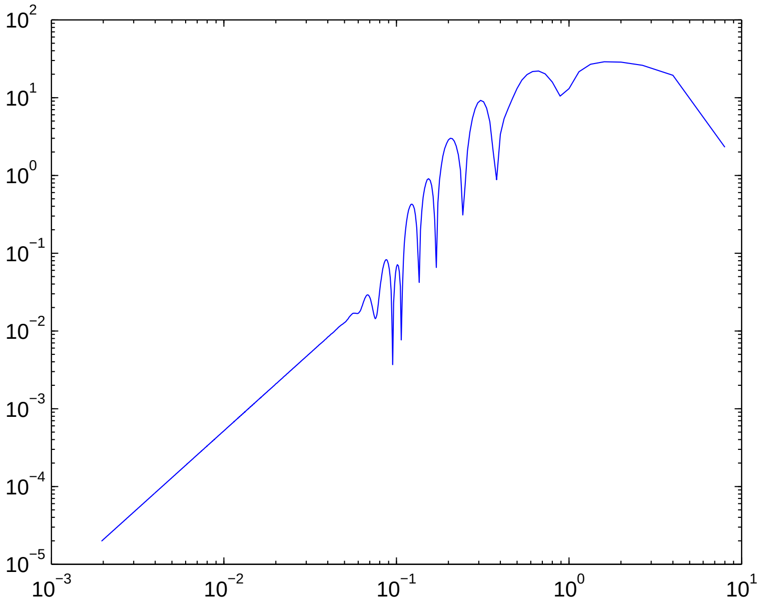

FIGURE 9. Log-log plot of the error of the composite trapezoid rule for $\int_0^8 f(x)\,dx$. Notice that for small subdivisions the plot is linear with slope 2, as predicted by theory.

One way to remedy the situation is to use a higher accuracy quadrature rule, say, the Simpson's rule. Yet, even if we use composite Simpson's rule, the computation will be inefficient for as long as the subdivision is even. The real problem is that bad behavior of the integrand on a small portion of the domain leads to a lot of extra computation on the portions of the domain where the integral is well-behaved. To compute integrals cost effectively we need to subdivide domains in a more flexible manner.

The idea of *adaptive quadrature* is based on additivity of integration:

$$\int_a^b f(x)\,dx = \int_a^c f(x)\,dx + \int_c^b f(x)\,dx.$$

Suppose we have a quadrature rule and a way to estimate error. If the error is too large when we apply the rule on the interval $[a, b]$, we can subdivide the interval into two halves and apply the rule on the smaller subintervals. If the error for either subinterval is too large, that subinterval is subdivided into two halves and the process repeats recursively, until desired tolerance is achieved.

Any quadrature rule can be turned into adaptive quadrature. The key to the enterprize is estimation of error which we discuss next.

6.1. **Error estimation.** Let T_1 be the result of applying the trapezoid rule to f on $[a, b]$:

$$T_1 = \frac{1}{2}\,(f(a) + f(b))\,(b - a).$$

Denote by c the midpoint of $[a, b]$ and let

$$T_2 = \frac{1}{2}\,(f(a) + f(c))\,\frac{b-a}{2} + (f(c) + f(b))\,\frac{b-a}{2}$$

be the estimate of the integral resulting from applying the trapezoid rule to $[a, c]$ and $[c, b]$ and adding the results. It stands to reason that T_2 should be a more accurate approximation to the integral $\int_a^b f(x)\,dx$ than T_1. Now, we know from the previous handout that

$$\int_a^b f(x)\,dx = T_1 - \frac{f''(\xi_1)}{12}(b-a)^3$$

and

$$\int_a^b f(x)\,dx = T_2 - \frac{f''(\xi_2)}{48}(b-a)^3.$$

If we assume that $f''(\xi_1) = f''(\xi_2)$, we get a system of two equations in two unknowns, from which follows that:

$$\int_a^b f(x)\,dx = \frac{1}{3}\,(4\,T_2 - T_1) \tag{33}$$

and

$$\frac{f''(\xi_1)}{12}(b-a)^3 = \frac{4}{3}\,(T_2 - T_1). \tag{34}$$

Equation (33) shows that the best estimate of the integral is not T_2 by itself, as one might expect, but rather a linear combination of T_1 and T_2; the process of combining T_1 and T_2 into an approximation that is more accurate than either T_1 or T_2 is an example of *Richardson's extrapolation.* Meanwhile, Equation (34) gives us a way to gauge the error of the trapezoid rule T_1.

We must bear in mind that we derived both Equation (33) and (34) by making a big assumption about the values of f''. This assumption is not always fulfilled, so our error estimate has to be taken with a grain of salt. Nevertheless, as the next section demonstrates, Equation (34) can be successfully used for error control.

7. Adaptive trapezoid rule in MATLAB

Since MATLAB supports recursive programming, we can implement adaptive quadrature rule using recursion. I prefer the implementation where the quadrature routine has a recursive integrator subroutine, as shown below.

```
function [Q,fcnt] = atrapz(f,a,b,tol)

% ATRAPZ   Numerically evaluate integral, adaptive trapezoid rule.

fa = f(a);
fb = f(b);
fcnt = 2;
Q = .5*(fa + fb)*(b-a);
Q = quadstep(a,b,fa,fb,Q);


function q = quadstep(a,b,fa,fb,T1)
    l = b - a;
    c = a + .5*l;
    fc = f(c);
    fcnt = fcnt + 1;
    Tleft = .25*(fa+fc)*l;
    Tright = .25*(fc+fb)*l;
    T2 = Tleft + Tright;
    E = 4*abs(T2 - T1)/3;
    if (b-a)*E < tol
        q = (4*T2 - T1)/3;
    else
        q = quadstep(a,c,fa,fc,Tleft) + quadstep(c,b,fc,fb,Tright);
    end
end

end
```

Notice that `quadstep` estimates the absolute error of the trapezoid method applied to a subinterval $[a, b]$ using Equation (34). The subdivision criterion, however, is based on the magnitude of

$$\frac{4}{3}\left|T_2 - T_1\right|(b-a).$$

The multiplication by $(b-a)$ improves the efficiency of the method. If the interval $[a, b]$ is very small, it does not need to be subdivided even if the error is relatively large.

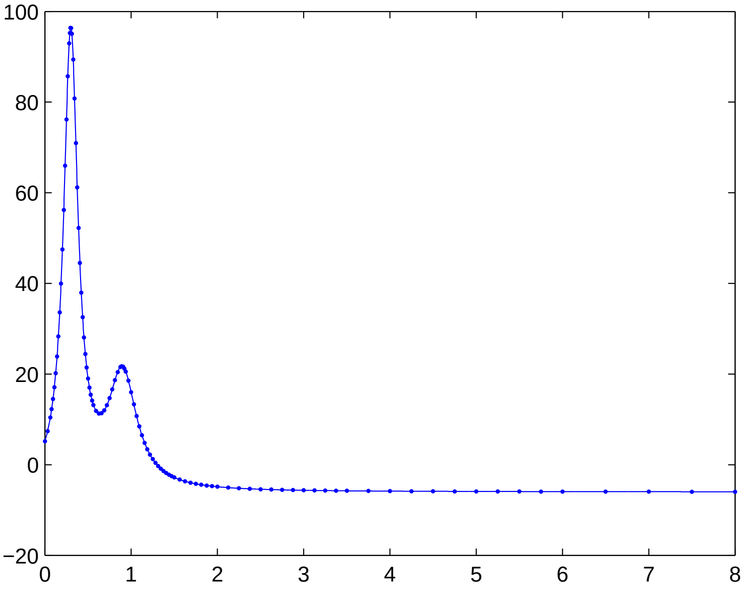

FIGURE 10. Adaptive trapezoid method applied to $\int_0^8 f(x)\,dx$ with tolerance 10^{-3}. The dots mark the resulting subdivision of $[0, 8]$.

Figure 10 shows the subdivision of $[0, 8]$ resulting from applying the adaptive trapezoid method to $\int_0^8 f(x)\,dx$ with tolerance 10^{-3}. The required tolerance is achieved using only 103 functional evaluations. For tolerance 10^{-6} adaptive trapezoid rule requires 607 functional evaluations; compare that with 5774 evaluations that composite rule with even subdivision requires to achieve tolerance 10^{-5}.

In order to gauge the efficiency of `atrapz` we can plot the number of functional evaluations against tolerance. Figure 11 shows the log-log plot which is almost a straight line with slope $-.2477$. Similar plots for different functions and intervals all result in lines with slopes close to $-.25$. This suggests that the functional count for adaptive trapezoid rule obeys the power law:

$$\# \text{ of evaluations} \sim \text{tolerance}^{-1/4}.$$

In words, the number of functional evaluations of adaptive trapezoid rule is roughly the square root of the number of functional evaluations of the composite trapezoid rule with even subdivisions.

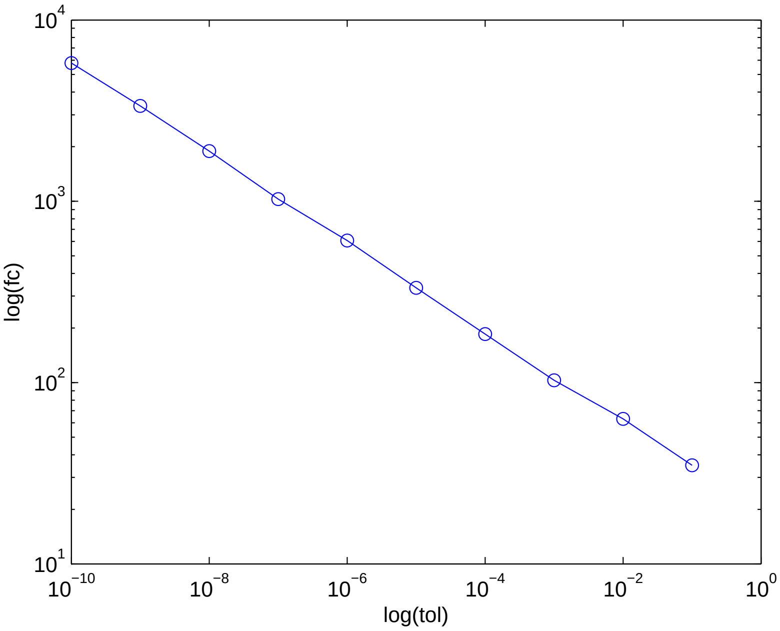

Figure 11. Efficiency of the adaptive trapezoid rule. The log-log plot of the number of functional evaluations against tolerance is nearly a straight line with slope $-.2477$.

Exercises

(1) Implement adaptive Simpson's rule and test it by integrating `humps` on $[0, 8]$ with tolerance ranging from 10^{-1} to 10^{-10}. Present your results in tabular form where for each tolerance level you compute the approximate value of the integral, the absolute error, and the number of functional evaluations. Additionally, make a log-log plot of the number of functional evaluations (cost) against the tolerance, similar to Figure 11. What can you say about the efficiency of the adaptive Simpson's rule?

5. Quadrature

The term *quadrature* has its origin in a simple technique for estimating areas: draw the shape on engineering paper and count the squares. For example, the shaded region in Figure 3 consists of 44 fully and partially filled squares. Therefore

$$\int_0^1 f(x)\,dx \leq 0.44,$$

where f is the function whose graph bounds the region.

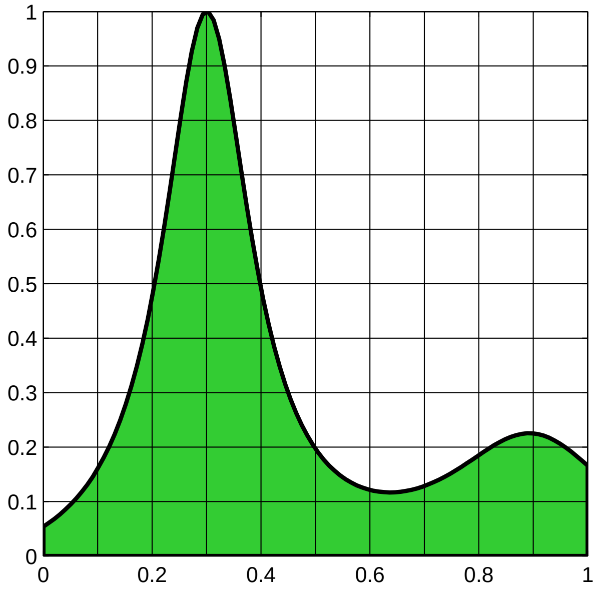

Figure 3. Quadrature in one dimension

As almost every topic in Numerical Analysis, the numerical quadrature is an extensive subject and an area of current research. We will start with quadrature in one dimension where the ideas are easier to bring into focus. Henceforth, our model problem is to approximate the definite integral $\int_a^b f(x)\,dx$ where f is a sufficiently well-behaved function and the limits of integration are finite. Once we understand the principles behind simple one-dimensional quadrature, we will try to extend them to multidimensional integrals and improper integrals[7].

[7] Improper integrals are the ones with infinite limits, integrands having singularities, or both.

5.1. **Quadrature in Calculus.** Figure 4 shows several *composite* quadrature rules that should be familiar from Calculus I.

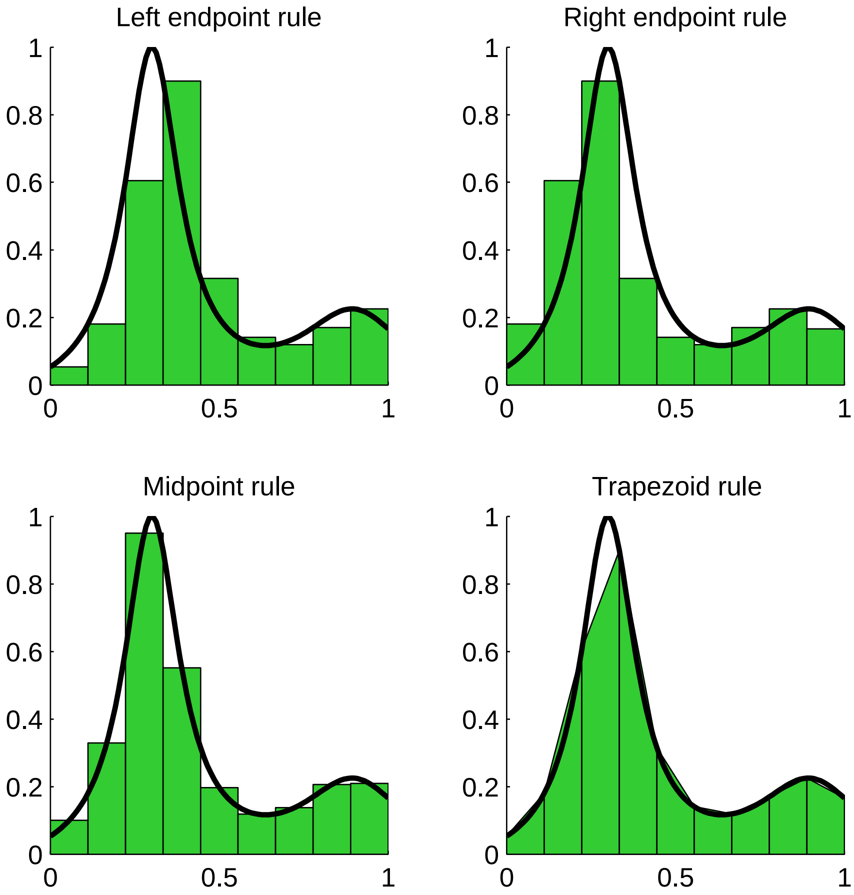

Figure 4. Quadrature in Calculus

In the context of quadrature, the word 'composite' simply means that the rule is applied on small subintervals into which the domain is subdivided; the results are then added up. For instance, the actual left endpoint rule stated for the interval $[a, b]$ is, simply:

$$\int_a^b f(x)\,dx \approx f(a)\,(b-a).$$

Suppose now that the interval $[a, b]$ is subdivided into N subintervals. Denote the left endpoint of the n-th subinterval by x_n and its length by Δx_n. The composite left endpoint rule is then:

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

The analysis of composite rules is straightforward once the rules themselves are understood. With that in mind, we turn our attention to the organizational principle behind quadrature. Can you see what the four methods in Figure 4 have in common?

5.2. **The organizational principle.** There are, literally, infinitely many quadrature rules. However, most of the rules—and certainly the most important ones—are based on the simple principle:

> To approximate $\int_a^b f(x)\,dx$, replace f with an interpolating polynomial[8].

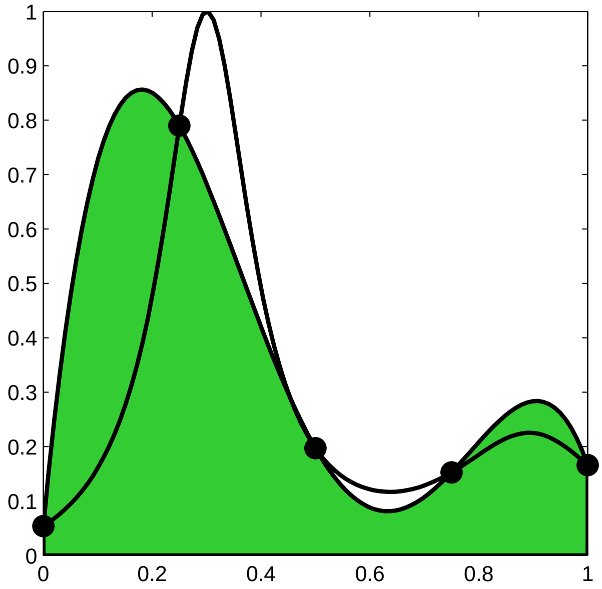

FIGURE 5. Interpolatory quadrature

Since it is often desirable to approximate integrals using only the values of the integrand, we will mostly focus on rules based on Lagrangian interpolation. The latter can be divided into several families differing

[8] Integration of rational functions leads to logarithms which are computationally expensive. Therefore Padé approximation is rarely used in quadrature schemes.

by the choice of nodes. All familiar Calculus rules, as well as the rule illustrated in Figure 5, belong to the class known as Newton-Cotes formulas which we introduce in the next section.

5.3. **Newton-Cotes quadrature rules.** Newton-Cotes rules use equispaced nodes and are divided into two types: open and closed.

Closed rules use the endpoints of the interval as some of the nodes. For example, the closed Newton-Cotes formula with two nodes uses the endpoints of the interval:

$$\begin{aligned} \int_a^b f(x)\,dx &\approx \int_a^b \left(f(a)\,\frac{x-b}{a-b} + f(b)\,\frac{x-a}{b-a} \right) dx \\ &= \frac{1}{2}\left(f(a)+f(b)\right)(b-a). \end{aligned} \tag{22}$$

Notice that Equation (22) is just the trapezoid rule.

The next closed formula is obtained by integrating the quadratic interpolant whose nodes are the endpoints and the midpoint. This rule is known as the Simpson's Rule; it is also sometimes called *parabolic rule* for obvious reasons. Let c denote the midpoint of $[a, b]$. Integration of the quadratic interpolant with nodes at a, b, and c leads to:

$$\begin{aligned} \int_a^b f(x)\,dx \approx \int_a^b \Bigg(& f(a)\,\frac{(x-b)(x-c)}{(a-b)(a-c)} + f(b)\,\frac{(x-a)(x-c)}{(b-a)(b-c)} \\ & + f(c)\,\frac{(x-a)(x-b)}{(c-a)(c-b)} \Bigg) dx \\ = {} & \frac{1}{6}\left(f(a)+4\,f(c)+f(b)\right)(b-a), \quad c = \frac{a+b}{2}. \end{aligned} \tag{23}$$

Here we purposefully omitted the details of algebraic simplification for two reasons. Firstly, they are tedious. Secondly, we would like to encourage the reader to produce Equation (23) independently using a computer algebra system (CAS) such as Maple or Mathematica. The use of CAS will become indispensable when we get to the analysis of errors in quadrature rules.

The closed Newton-Cotes formula based on the cubic interpolant is also attributed to Simpson and is called the Simpson's 3/8 rule. One of the exercises at the end asks you to derive it and explain the name. Higher order Newton-Cotes formulas do not have special names and are, in fact, rarely used in practice.

Open Newton-Cotes formula use only the interior nodes. The simplest example is the midpoint rule:

$$\int_a^b f(x)\,dx \approx \int_a^b f(c)\,dx = f(c)\,(b-a), \quad c = \frac{a+b}{2}.$$

Following that, is the open Newton-Cotes formula with two nodes—it does not have a special name:

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\,(f(x_1)+f(x_2))\,(b{-}a), \quad x_1 = \frac{2\,a+b}{3}, \quad x_2 = \frac{a+2\,b}{3}.$$

Derivation of other open Newton-Cotes formulas is similar and is relegated to exercises. Here we close with the remark that the reason why open formulas are distinguished from closed formulas is because they apply to integrands with singularities at the endpoints of the interval.

Notice that all Newton-Cotes formulas can be written as linear combinations of function values:

$$\int_a^b f(x)\,dx \approx \sum_{n=1}^{N} w_n\, f(x_n). \tag{24}$$

In Equation (24) x_n's are the (equispaced) nodes (interior for open formulas) and w_n are the *weights* which are also called *Cotes numbers.* As this section shows, the Cotes numbers are obtained simply by integrating the appropriate Lagrangian interpolant.

As we have already suggested, in practice one works with Newton-Cotes rules based on a small number of nodes. The main reason for this is the instability of interpolation. For large numbers of nodes the interpolants oscillate wildly and so do the Cotes numbers. Rules where Cotes numbers are of different signs tend to be numerically unstable and are therefore avoided.

5.4. **Error analysis: take one.** Since quadrature rules are in one way or another based on interpolation, we can easily derive an expression for the error by integrating the remainder. For instance, integration of the linear interpolation formula with the remainder

$$f(x) = f(a)\,\frac{x-b}{a-b} + f(b)\,\frac{x-a}{b-a} + \frac{f''(\xi)}{2}\,(x-a)\,(x-b),$$

leads to the error term for the trapezoid rule (22):

$$\int_a^b f(x)\,dx - \frac{1}{2}\,(f(a)+f(b))\,(b-a) = \int_a^b \frac{f''(\xi(x))}{2}\,(x-a)\,(x-b)\,dx.$$

We wrote $\xi = \xi(x)$ inside the integral to emphasize that ξ cannot be treated as a constant. Nevertheless, since $(x-a)\,(x-b) \le 0$ for all $x \in [a,b]$, we can use GMVT to pull out the second derivative. This results in

$$
\begin{aligned}
\int_a^b \frac{f''(\xi(x))}{2}\,(x-a)\,(x-b)\,dx &= \frac{f''(\eta)}{2}\int_a^b (x-a)\,(x-b)\,dx \\
&= -\frac{f''(\eta)}{12}\,(b-a)^3,
\end{aligned} \tag{25}
$$

where η now is a fixed number in $[a,b]$. Equation (25) shows that the trapezoid rule is exact for all polynomials of degree one or less. It further shows that the error of the trapezoid rule depends cubically on the length of the interval. In particular, if the length of the interval is halved, the error decreases by the factor of eight.

Unfortunately, the remainder of the Lagrangian interpolation does not always maintain constant sign on the domain of integration and this limits the use of GMVT. For instance, for the midpoint rule we can certainly write the error as the following integral:

$$\int_a^b f(x)\,dx - f(c)\,(b-a) = \int_a^b f'(\xi(x))\,(x-c)\,dx, \quad c = \frac{a+b}{2}.$$

Yet $(x-c)$ changes sign on $[a,b]$ and GMVT cannot be applied. In order to derive an error term similar to (25), we need a more subtle approach presented in the next section.

5.5. **Take two: Peano Kernel Theorem.**

All interpolatory quadrature schemes are, by design, exact for low order polynomials. Let n be the highest degree a polynomial can have so that the quadrature rule produces an exact answer. This nonnegative integer is called the *order of accuracy* of the rule. For instance, both the trapezoid and midpoint rules are of first order. Peano Kernel Theorem expresses the error of interpolatory quadrature as an integral of the form

$$\int_a^b f^{(n+1)}(t)\,K(t)\,dt,$$

where n is the order of accuracy and $K(t)$—the Peano kernel—is independent of f. Unlike the Lagrangian remainder, the Peano kernel tends to maintain constant sign on $[a,b]$. This allows for the use of GMVT leading to error formulas of the type $k\,f^{(n+1)}(\eta)$ with $k = \int_a^b K(t)\,dt$. In order to state Peano's theorem, we need to introduce some language from linear algebra which we proceed to do.

5.5.1. *Linear functionals.* Think of a functional as a procedure that takes a function as an input and produces a real or complex number as an output. For instance, consider the functional L defined by

$$L(f) = \int_a^b f(x)\,dx.$$

This, clearly, is a familiar object that is of prime interest to us now. However, let us think of the integral differently: not as “the area under the curve” or “the limit of a Riemann sum” but as a “function” operating on functions. The functional L is linear because

$$L(c_1\,f_1 + c_2\,f_2) = c_1\,L(f_1) + c_2\,L(f_2)$$

for any functions f_1, f_2 and constants c_1, c_2. In words, L maps linear combinations of inputs into linear combinations of outputs. Any functional that does not have that property is nonlinear, e.g.:

$$N(f) = \int_a^b f^2(x)\,dx.$$

Linearity is a crucial attribute that is going to be required by our theory. We will therefore only consider linear functionals.

As another example consider the evaluation functional: $f \mapsto f(p)$. This functional simply evaluates a function at some point p and is clearly linear. Any linear combination of linear functionals is itself a linear functional (exercise). In particular, any interpolatory quadrature rule, being a linear combination of evaluation functionals, is a linear functional designed to approximate L.

Let Q be a linear functional that approximates L in some sense. We will call the difference $E = L - Q$ the *error functional* corresponding to Q. For instance, the error functional for the midpoint rule is given explicitly by:

$$E(f) = \int_a^b f(x)\,dx - f\left(\frac{a+b}{2}\right)(b-a). \tag{26}$$

5.5.2. *Peano Kernel Theorem.* Since the midpoint rule is of first order, its error functional E defined by (26) is zero for all linear functions. We will say that E *annihilates* first degree polynomials. Peano observed that something constructive can be said about linear functionals that annihilate polynomials.

Theorem 6 (Peano Kernel Theorem)**.** *Let L be a linear functional acting on smooth functions defined on the interval $[a,b]$. Suppose that L annihilates all polynomials of degree n or less. Then for any $f \in C^\infty([a,b])$:*

$$L(f) = \int_a^b f^{(n+1)}(t)\,K(t)\,dt, \tag{27}$$

where

$$K(t) = \frac{1}{n!}\, L_x\left((x-t)_+^n\right).$$

The subindex in L_x indicates that the functional is applied in the x-variable; the plus sign in $(x-t)_+^n$ is standard notation for the truncated power function:

$$(x-t)_+^n = \begin{cases} (x-t)^n, & t < x, \\ 0, & \textit{otherwise.} \end{cases}$$

Proof. Let $f \in C^\infty([a,b])$. The Taylor expansion of f to n-th order at $x = a$ can be written as follows:

$$\begin{aligned} f(x) &= f(a) + f'(a)\,(x-a) + \frac{f''(a)}{2}\,(x-a)^2 + \dots \\ &+ \frac{f^{(n)}(a)}{n!}\,(x-a)^n + \int_a^x \frac{(x-t)^n}{n!}\, f^{(n+1)}(t)\,dt. \end{aligned} \tag{28}$$

Now apply L to Equation (28). Since L is linear and since L annihilates polynomials of degree n or less, the result is:

$$L(f) = L\left(\int_a^x \frac{(x-t)^n}{n!}\, f^{(n+1)}(t)\,dt\right).$$

To finish the proof, we would like to bring L inside the integral. To do that, introduce the truncated power function and write the integral as follows:

$$\int_a^x \frac{(x-t)^n}{n!}\, f^{(n+1)}(t)\,dt = \int_a^b \frac{(x-t)_+^n}{n!}\, f^{(n+1)}(t)\,dt.$$

Now that the limits are constant, L can be brought inside the integral and, since L acts on the variable x, the result is:

$$L(f) = \int_a^b L_x\left(\frac{(x-t)_+^n}{n!}\right) f^{(n+1)}(t)\,dt = \int_a^b K(t)\, f^{(n+1)}(t)\,dt.$$

Which was to be demonstrated. □

Of particular use to us will be the following consequence of Theorem 6.

Corollary 7. *If Peano kernel maintains constant sign on $[a,b]$ then Equation (27) can be rewritten as:*

$$L(f) = f^{(n+1)}(\xi) \int_a^b K(t)\,dt. \tag{29}$$

The proof of Corollary 7 is a straightforward application of GMVT.

5.5.3. *The use of Peano Kernel Theorem.* As an application of Theorem 6, let us find the error of the midpoint rule. Since the rule is of first order, the Peano kernel is obtained by applying the error functional (26) to $(x-t)_+$:

$$K(t) = \int_a^b (x-t)_+ \, dx - \left(\frac{a+b}{2} - t\right)_+ (b-a).$$

Since $(x-t)_+ = 0$ as long as $x < t$, we can compute the integral as follows:

$$\int_a^b (x-t)_+ \, dx = \int_t^b (x-t) \, dx = \frac{(b-t)^2}{2}.$$

Now, by definition of the truncated power function:

$$\left(\frac{a+b}{2} - t\right)_+ (b-a) = \begin{cases} \left(\frac{a+b}{2} - t\right)(b-a), & a \le t < \frac{a+b}{2}, \\ 0, & \frac{a+b}{2} \le t \le b. \end{cases}$$

Hence

$$\begin{aligned} K(t) &= \frac{(b-t)^2}{2} + \begin{cases} \left(\frac{a+b}{2} - t\right)(b-a), & a \le t < \frac{a+b}{2} \\ 0, & \frac{a+b}{2} \le t \le b \end{cases} \\ &= \begin{cases} \frac{(b-t)^2}{2} + \left(\frac{a+b}{2} - t\right)(b-a), & a \le t < \frac{a+b}{2} \\ \frac{(b-t)^2}{2}, & \frac{a+b}{2} \le t \le b \end{cases} \\ &= \begin{cases} \frac{(a-t)^2}{2}, & a \le t < \frac{a+b}{2}, \\ \frac{(b-t)^2}{2}, & \frac{a+b}{2} \le t \le b. \end{cases} \end{aligned}$$

Evidently, $K(t) \ge 0$ on $[a,b]$. Therefore, by Corollary 7:

$$\begin{aligned} E(f) &= f''(\xi) \int_a^b \left(\begin{cases} \frac{(a-t)^2}{2}, & a \le t < \frac{a+b}{2}, \\ \frac{(b-t)^2}{2}, & \frac{a+b}{2} \le t \le b. \end{cases} \right) dt \\ &= f''(\xi) \left(\int_a^{\frac{a+b}{2}} \frac{(a-t)^2}{2} \, dt + \int_{\frac{a+b}{2}}^b \frac{(b-t)^2}{2} \, dt \right) \\ &= \frac{f''(\xi)}{24} (b-a)^3. \end{aligned}$$

Curiously, the midpoint rule, which uses a single node, is twice as accurate[9] as the trapezoid rule which uses two nodes.

[9] We do not mean to say that in a given situation the error of the midpoint rule will necessarily be half that of the trapezoid rule. The second derivative in the two error terms is evaluated at different ξ. So, it may very well happen that in a particular situation the trapezoid rule outperforms midpoint rule. However, on average, one can expect the midpoint rule to be somewhat more accurate.

5.6. **Error of composite rules.** We will illustrate the analysis of composite quadrature using the trapezoid rule; other composite rules are analyzed in the same manner. Recall that 'composite' means that we break up the interval $[a, b]$ into N subintervals, apply the rule to each subinterval, and add up the results. For simplicity, let us assume that the subdivision is even, so all subintervals have length:

$$h = \frac{b-a}{N}.$$

Let x_n, $n = 0, \dots, N$ denote equispaced subdivision points with $x_0 = a$ and $x_N = b$. The formula for the composite trapezoid rule has very simple form

$$\frac{f(x_0) + f(x_N)}{2}\, h + h \sum_{n=1}^{N-1} f(x_n) \tag{30}$$

and is implemented in MATLAB as `trapz`.

The error of the composite trapezoid rule is the sum:

$$\sum_{n=1}^{N} \left(-\frac{f''(\eta_n)}{12}\, h^3 \right) = -\frac{h^3}{12} \sum_{n=1}^{N} f''(\eta_n), \quad x_{n-1} \le \eta_n \le x_n. \tag{31}$$

If the second derivative is continuous, which is our tacit assumption, then by IVT:

$$\sum_{n=1}^{N} f''(\eta_n) = N \times \frac{1}{N} \sum_{n=1}^{N} f''(\eta_n) = N\, f''(\xi), \quad a \le \xi \le b.$$

Furthermore, $h\, N = b - a$. Using that, Equation (31) can be simplified to:

$$-\frac{f''(\xi)}{12}\, h^2\, (b - a). \tag{32}$$

To confirm Equation (32), let us apply it to $\int_0^1 x^2\, dx$. Since the length of the interval is one and the second derivative of x^2 is constant, the absolute error is simply:

$$E = \frac{h^2}{6}.$$

Applying the logarithms, we get

$$\log(E) = 2\, \log(h) + \log\left(\frac{1}{6}\right).$$

This means that the log-log plot of the error should produce a straight line with slope two and intercept log(1/6). This is indeed the case as shown in Figure 6.

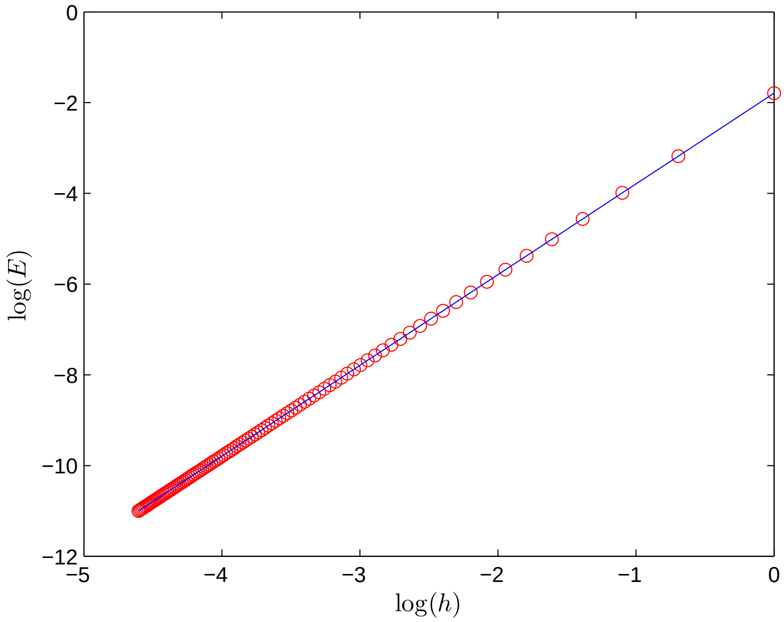

Figure 6. The log-log plot of the error of the trapezoid rule applied to $\int_0^1 x^2\,dx$. Plotted in red circles is the data computed using MATLAB's `trapz` command. The best linear fit (blue line) has slope 2 and intercept log(1/6). The fit is perfect because the second derivative of x^2 is constant.

Figure 6 was produced with the following code:

```
f = @(x) x.^2;
I = quad(f,0,1);
h = 1./(1:100);
E = zeros(size(h));
for n=1:length(R)
    x = linspace(0,1,n+1);
    y = f(x);
    Q = trapz(y)*h(n);
    E(n) = abs(I-Q);
end
u = log(h);
v = log(E);
p = polyfit(u,v,1);
plot(u,v,'ro')
```

```
hold on
plot(u,polyval(p,u))
xlabel('$\log(h)$','Interpreter','latex','FontSize',12)
ylabel('$\log(E)$','Interpreter','latex','FontSize',12)
```

For more complicated functions, the fit cannot be perfect, yet it still suggests quadratic dependence in most cases.

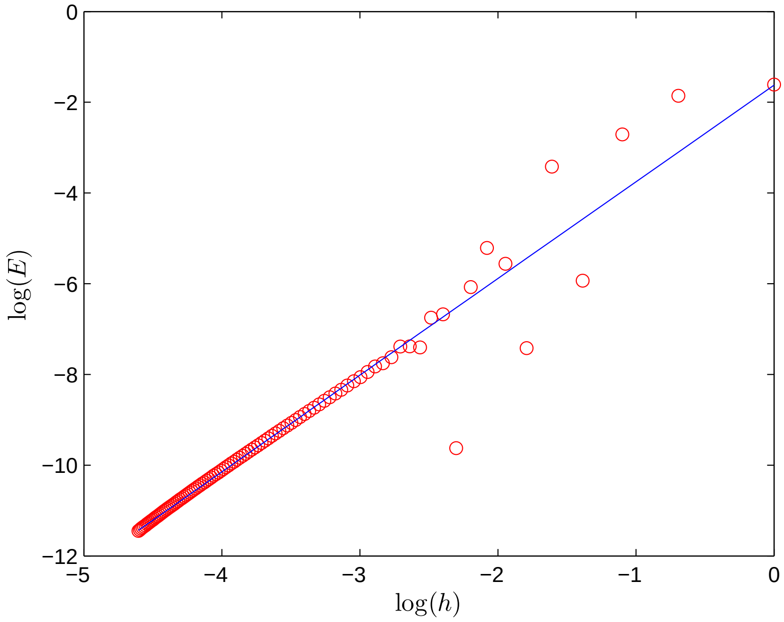

FIGURE 7. The log-log plot of the error of the trapezoid rule applied to $\int_0^1 f(x)\,dx$ where f is the function shown in Figure 3 (MATLAB's `humps` scaled by its maximum). Plotted in red circles is the error of `trapz`; the best linear fit (blue line) now has slope 2.1312 due to noise in the data. The fit still strongly suggests that the error depends quadratically on h.

EXERCISES

(1) Derive the formula for the closed Newton-Cotes rule with four nodes and explain why it is called the Simpson's 3/8 rule. Use CAS if you can.

(2) Use Peano Kernel Theorem to find the error term for the Simpson's Rule. Be sure to provide a clean derivation of the Peano kernel (it will be easier if you use CAS).

(3) Consider the following quadrature rule:
$$\int_{-1}^{+1} f(x)\,dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right).$$
Find the order of accuracy and the error term.

(4) Let
$$D(f) = \frac{f(b) - f(a)}{b - a}$$
be an approximation to $f'(b)$. Find the error of this approximation.

(5) Repeat the previous exercise but regard D as the approximation to $f'(c)$ where $c = \frac{a+b}{2}$.

(6) Let $f \in C^2([a, b])$. Use interpolation to construct a rule for finding the second derivative $f''(b)$ from the function values at five equispaced nodes (CAS is recommended). What is the order of accuracy of the rule? What is the error term?

(7) Derive the error term for composite Simpson's rule. Illustrate it with figures similar to 6 and 7

(8) Confirm numerically that Equation (32) applies to
$$\int_0^{2\pi} \frac{dx}{2 + |\sin(x)|}$$
even though the integrand is not differentiable. How would you explain that?

(9) Perform several numerical experiments where you compare the accuracy of the composite trapezoid and composite midpoint rules. Is it fair to say that the midpoint rule tends to be twice as accurate as trapezoid?

(10) Investigate (numerically) the validity of Equation (32) for the following integral:
$$\int_0^{2\pi} \frac{dx}{2 + \sin(x)}.$$
How fast does the error of the composite trapezoid rule seems to decrease for smooth periodic functions?