7 Lecture #5: Tuesday, March 3rd, 2026

7.1 Introduction to Numerical Analysis

Our main motivation for studying Numerical Analysis in this course comes from a fundamental limitation we have already encountered when working with ODEs: many of them cannot be solved explicitly. In other words, even though a solution may exist, there is often no analytical procedure that allows us to write this solution explicitly. For instance, consider the equation

$$y' = 2x + y^2.$$

This equation does have solutions, but they cannot be expressed explicitly in the same way as the examples we have studied before. The natural question is therefore: what should we do when an analytical solution is not available? This is precisely where Numerical Analysis enters.

Numerical Mathematics is a field that provides practical methods for approximating solutions when exact formulas are not accessible. In many applications, especially in Engineering and Applied Sciences, exact solutions are not strictly necessary. Instead accurate approximations are often sufficient to understand the behavior of a system and make reliable predictions. We have already seen a similar situation in the introduction of these notes. The equation

$$x^5 + x = 13$$

has at least one solution, but there is no algebraic procedure that gives us its exact value explicitly. Nevertheless, numerical methods allow us to approximate this solution as closely as desired. Likewise, even numbers such as $\sqrt[5]{13}$ cannot be handled exactly using elementary operations, but they can be approximated with arbitrary precision. The goal of Numerical Analysis is therefore to develop systematic procedures that produce approximations of solutions, together with an understanding of the accuracy and reliability of these approximations.

Let us consider the following situation. We have a circular table of diameter $d = 1\text{m}$. Suppose that, for some reason, we want to glue a strip of protection all around the table (see Figure 22). What is the length of strip that we really need? As we learn when we are kids, the perimeter of a circle is given by

$$P = d \cdot \pi = \pi\text{m}.$$

However, how can we buy a strip with this exact length? In practice, we need a decimal number, and therefore we need some Numerical Analysis.

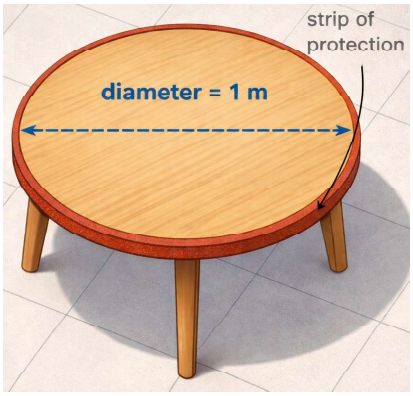

Figure 22: Table of diameter 1m.

The number π has been studied since ancient times, long before the development of modern mathematics. Many civilizations observed that the ratio between the circumference of a circle and its diameter is always the same, regardless of the size of the circle. By performing measurements and approximations, mathematicians in the past concluded that this ratio is close to the rational number

$$\pi \approx \frac{22}{7}.$$

In this course, we will be using the notation "$\approx$" to indicate that two quantities are close to each other, without any precise meaning. Coming back to our example, we cannot buy exactly πm of the strip, and instead we might buy

$$\frac{22}{7}\,\text{m} \approx 3.142857142857\ldots\,\text{m}.$$

By doing so, we have created an error, called a **numerical error**. We will be interested in finding **numerical methods**, that is, procedures for producing approximate answers. This also means that we will need to study the **error of the method**, and how this error behaves. This is a fundamental part of Numerical Analysis. In fact, the key point is to keep the error of the method under control. While working with a numerical method, we must understand how the method reacts to such errors. It turns out that some methods are quite resistant to errors, while others are very sensitive. This property is known as **numerical stability**. Finally, we will also be interested in how much it costs to obtain an answer, in terms of computational time. Some calculations may take seconds, while others might take months or even years.

7.2 Errors in calculations

The first basic concept we will study is the notion of error. What do we mean by an error? Suppose that x is a quantity whose exact value is unknown, and let $\hat{x}$ be an approximation of x. The difference between these two numbers measures how far the approximation is from the true value and is called the **absolute error**, as we formalize below.

Notice that there are several possible ways to compare x and $\hat{x}$. For instance, one could consider the quantities $x - \hat{x}$, $\hat{x} - x$ or even $|x - \hat{x}|$. In practice, working with absolute values is not always convenient in calculations, so, in this course, we define the error as

$$E_x = x - \hat{x}$$

and refer to this quantity as the absolute error. Its magnitude is then measured by $|E_x|$. Let us look at a simple example. Suppose we want to measure the length of a stick using a ruler marked in centimeters (see Figure 23). By visual inspection, we estimate that the length is approximately

$$\hat{\ell} = 13.6\ \text{cm} = 136\ \text{mm}.$$

Of course, this is not the exact length of the stick, so an error has been introduced in the measurement. In the expression $E_x = x - \hat{x}$ the exact value x is unknown, since the true length of the stick is precisely what we do not know. Therefore, instead of computing the exact error, we estimate its size. Taking into account the precision of the ruler, we can conclude that

$$|E_x| \leqslant 0.5\text{mm}.$$

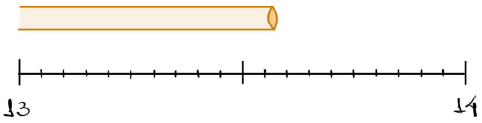

Figure 23: Ruler in centimeters and sticker.

This gives an upper bound for the error and tells us how accurate our measurement is.

More generally, suppose that we know an upper bound for the error. This means that $|E_x| \leqslant e_x$, where e_x is a positive number representing the maximum possible error. Since $E_x = x - \hat{x}$, we can write

$$|x - \hat{x}| = |E_x| \leqslant e_x.$$

This inequality is equivalent to saying that the exact value x lies within a distance e_x from the approximation $\hat{x}$. Therefore,

$$\hat{x} - e_x \leqslant x \leqslant \hat{x} + e_x.$$

There is a convenient shorthand notation for this interval, namely $x = \hat{x} \pm e_x$. In our previous example, this can be written as

$$\ell = 136.0 \pm 0.5\text{mm},$$

meaning that the true length of the stick lies between 135.5mm and 136.5mm.

Let us now address the question of how reliable an approximation is. Consider the following example. Suppose that we know that $|E_x| \leqslant 0.5$mm. Assume first that we are measuring the distance between our house and a bus stop in order to organize our daily routine. Suppose that this distance is approximately $x = 423$m. Expressing this measurement in millimeters, we obtain

$$x = 423000 \pm 0.5\text{mm}.$$

This appears to be an extremely precise measurement, perhaps unrealistically precise. In practice, achieving such accuracy would require very expensive equipment, which suggests that we should carefully check whether the data or units are consistent with the situation. On the other hand, suppose we measure the thickness of a human hair and obtain $x = 0.08$mm. Then the same error estimate would lead to

$$x = 0.08 \pm 0.5\text{mm}.$$

This is clearly a poor measurement, since the error is much larger than the quantity itself. In fact, the interval would even include negative values, which makes no physical sense in this context. These examples show that the absolute error alone is not sufficient to judge the quality of an approximation. In order to evaluate how reliable a measurement is, we must compare the size of the error with the magnitude of the quantity being measured. This gives the concept of the relative error. Let us formalize it in what follows.

Definition 7.1. Let x be a number and $\hat{x}$ its estimate. Then, we define the **absolute error** by

$$E_x = x - \hat{x}.$$

We also define the **relative error** as

$$\varepsilon_x = \frac{|E_x|}{|x|} \quad \text{if} \quad x \neq 0.$$

By an **error estimate** we mean any number e_x satisfying $|E_x| \leqslant e_x$.

Now we would like to get some information by taking a look at the relative error. By definition, the relative error is given by

$$\varepsilon_x = \frac{|E_x|}{|x|},$$

so that

$$|E_x| = \varepsilon_x \cdot |x|.$$

As we have already discussed, this formula is not completely convenient in practice because the exact value x is unknown. There are two common ways to deal with this difficulty. An engineer would typically argue that, since $\hat{x}$ is a good approximation of x, one may replace $|x|$ by $|\hat{x}|$, obtaining

$$|E_x| \approx \varepsilon_x \cdot |\hat{x}| \tag{12}$$

hoping that the additional error introduced by this replacement is negligible. A mathematician, however, prefers to justify this replacement rigorously. Observe that since $x = \hat{x} + E_x$, we have that

$$|E_x| = \varepsilon_x|x| = \varepsilon_x|\hat{x} + E_x|.$$

Applying the triangle inequality, we obtain

$$|E_x| = \varepsilon_x|\hat{x} + E_x| \leqslant \varepsilon_x(|\hat{x}| + |E_x|).$$

Rearranging terms gives

$$|E_x| - \varepsilon_x|E_x| \leqslant \varepsilon_x|\hat{x}|.$$

Factoring the left-hand side,

$$(1 - \varepsilon_x)|E_x| \leqslant \varepsilon_x|\hat{x}|.$$

Assuming $\varepsilon_x < 1$, we conclude that

$$|E_x| \leqslant \frac{\varepsilon_x}{1 - \varepsilon_x}\,|\hat{x}|. \tag{13}$$

This inequality provides a rigorous bound for the absolute error in terms of the approximation $\hat{x}$ and the relative error. Notice that (12) and (13) are remarkably similar.

Example 7.2. Suppose that we have obtained the approximation $\hat{x} = 23456789$. Assume that, from some source, we know that the relative error satisfies

$$\varepsilon_x = \frac{1}{100} = 10^{-2}.$$

What does this tell us about the true value x? Using the engineer's approach, we estimate the absolute error by

$$|E_x| \approx \varepsilon_x \cdot |\hat{x}| = 10^{-2} \cdot 23456789 = 234567.89.$$

Therefore, the true value lies approximately in the interval

$$x \approx \hat{x} \pm |E_x| = 23456789.00 \pm 234567.89.$$

This means that only the first 2 digits remain unchanged, and we may write informally

$$x \approx 23 * * * * * * * .$$

Hence, when the relative error is $\varepsilon_x = 10^{-2}$, we can trust roughly the first two significant digits. Assume now that the relative error is given by

$$\varepsilon_x = 10^{-3}.$$

Repeating the same reasoning, the absolute error becomes smaller by one order of magnitude, and we obtain

$$x \approx 234 * * * * * .$$

In this case, we can trust approximately 3 significant digits of the approximation. This example illustrates a general rule: if the relative error is about 10^{-p}, then approximately p significant digits of the approximation are reliable. Notice, however, that this rule should be understood as a guideline and **not** as an absolute statement. For instance, suppose that the relative error is $\varepsilon_x = 10^{-5}$. Using the same reasoning as before, we obtain

$$|E_x| \approx 10^{-5} \cdot 23456789 = 234.56789,$$

and therefore

$$x \approx 23456789 \pm 234.56789.$$

At first sight one could expect that five digits are reliable, but this is not exactly what happens. The uncertainty still affects several of the last digits, and the number may vary within a range that changes more than just the final five positions. This shows that the relation between relative error and the number of reliable digits is only approximate. In practice, one should always check the size of the absolute error to understand how many digits can really be trusted. Thefore, in most of the cases and to stay in the safe side, the general rule should read as follows. If the relative error is about 10^{-p}, then (in most of the cases) approximately $p - 1$ significant digits of the approximation are reliable.

7.3 Floating point representation and error propagation

When working with real numbers on a computer, we immediately face a practical difficulty: many decimal numbers, especially irrational ones, have infinitely many digits and therefore cannot be stored exactly. Since computers have finite memory, only a limited number of digits can be recorded. This means that numbers must be approximated before being stored. To overcome this problem, numerical computation uses a standard way of representing numbers called **floating point representation**. The basic idea is to store a number using a fixed number of significant digits together with an exponent, similarly to scientific notation. In this way, very large and very small numbers can be handled efficiently while keeping the amount of stored information finite. Although this representation introduces a small approximation error, it provides a practical and systematic way to perform numerical calculations.

Example 7.3. Consider the number 13.2567 and suppose that we want to store it in a computer. First, we write it in scientific notation

$$13.2567 = 1.32567 \times 10^1.$$

Assume that the computer can store only $p = 4$ digits in the mantissa[§]. There are two common ways to handle this situation. One possibility is **truncation**, where we simply discard the extra digits. In this case, we store

$$1.325 \times 10^1.$$

[§]As we will see in a moment, when we write a number x in scientific notation $x = m \times 10^e$, m is called the **mantissa** and e is called the **exponent**. The mantissa carries the meaningful digits of the number, while the exponent only indicates the scale (how large or small the number is).

The second possibility is **rounding**, where the last stored digit is adjusted according to the next one. Here, since the following digit is 6, we store

$$1.326 \times 10^1.$$

Let us now modify the example slightly. Suppose that $p = 3$ and consider again the number

$$1.325 \times 10^1.$$

If we round it, we would write

$$1.33 \times 10^1,$$

although 1.325 lies exactly halfway between 1.32 and 1.33. Always rounding in the same direction may introduce a systematic error in computations. For this reason, in practice engineers often use rounding rules that alternate between rounding up and rounding down (or follow a specific convention such as rounding to the nearest even digit) in order to reduce accumulated errors.

We have the following definition.

Definition 7.4. By a **floating point representation** of a number x with respect to a base β, with precision of p significant digits we mean the best approximation $fl(x)$ of x that can be written as

$$fl(x) = d_1.d_2d_3 \cdots d_p \times \beta^e$$

where $d_1\{1, \ldots, \beta - 1\}$ and $d_2, \ldots, d_p \in \{0, 1, \ldots, \beta - 1\}$. The number e is called the **exponent**, the part $d_1 \cdot d_2 \cdots d_p$ is called the **significand** or the **mantissa**.

Let us have a small observation about the last definition although we might sound repetitive. When we use floating point representation, numbers cannot usually be stored exactly because the computer keeps only a limited number of digits in the mantissa. As a consequence, the stored value is only an approximation of the real number. This means that every time we represent a number in floating point form, a small error is automatically introduced. This error comes from truncation or rounding and is unavoidable in numerical computations.

We now ask how these small errors can influence the final result when we perform calculations on a computer or calculator.

Example 7.5. Let us start with a simple example. Suppose we want to compute

$$\frac{4}{9} + \frac{4}{9} = \frac{8}{9}.$$

Assume that the machine has very limited memory and can store only $p = 2$ digits. In this case,

$$\frac{4}{9} \approx 0.44$$

while

$$\frac{8}{9} \approx 0.89.$$

If we perform the computation with the stored values, we obtain

$$0.44 + 0.44 = 0.88,$$

which is different from the stored value 0.89 for 8/9. We already observe an inconsistency produced by rounding. The error does not come from the arithmetic itself, but from the approximations introduced when the numbers are stored.

This simple example illustrates how small representation errors may propagate and affect the final result. We have the following result.

Fact 7.6. Consider real numbers x, y and their estimates $\hat{x}, \hat{y}$. Then the following is true.

- $|E_{x+y}| \leqslant |E_x| + |E_y|$ and $\varepsilon_{x+y} \leqslant \max(\varepsilon_x, \varepsilon_y)$ for $x, y > 0$.
- $|E_{x-y}| \leqslant |E_x| + |E_y|$ and $\varepsilon_{x-y} \leqslant \dfrac{|x| + |y|}{|x - y|} \max(\varepsilon_x, \varepsilon_y)$.
- $|E_{x\cdot y}| \leqslant |y| \cdot |E_x| + |\hat{x}| \cdot |E_y|$ and $\varepsilon_{x\cdot y} \leqslant \varepsilon_x + (1 + \varepsilon_x)\varepsilon_y$.
- $|E_{x/y}| \leqslant \dfrac{1}{|y|} \left(|E_x| + |E_y| \cdot \dfrac{|\hat{x}|}{|\hat{y}|} \right)$ and $\varepsilon_{x/y} \leqslant \varepsilon_x + \varepsilon_y \cdot \dfrac{1 + \varepsilon_x}{1 - \varepsilon_y}$ for $\varepsilon_y < 1$.
- $|E_{1/x}| \leqslant \dfrac{|x|}{|\hat{x}|} |E_x|$ and $\varepsilon_{1/x} \leqslant \dfrac{1}{1 - \varepsilon_x} \varepsilon_x$ for $\varepsilon_x < 1$.

The first formula has a very natural interpretation. The errors E_x and E_y measure how far the approximations $\hat{x}$ and $\hat{y}$ are from the true values x and y. When we add two approximate numbers, the total error in the result comes from both individual errors acting together. Intuitively, the worst possible situation occurs when both errors act in the same direction: for example, when both approximations are larger than the true values or both are smaller. In that case, the errors accumulate, and the total error can be as large as the sum of their sizes. On the other hand, the errors might partially cancel each other, producing a smaller final error. Therefore, the formula above should be understood as a safe upper bound: the error in the sum cannot be larger than the sum of the individual absolute errors, although in practice it may be smaller.

The third formula for the multiplication describes how errors behave when we multiply two approximate numbers. The key idea is that, unlike addition, the error of a product depends not only on the errors themselves but also on the size of the numbers involved.

The following rules, on the other hand, give an intuitive picture of how relative errors propagate through basic arithmetic operations.

Fact 7.7 (Engineering fact)**.** Assume that there is a rounding error $\varepsilon > 0$ on input, then for $x, y > 0$ we (almost) have

- $\varepsilon_{ax+y} \leqslant \varepsilon$.
- $\varepsilon_{x-y} \leqslant \dfrac{x + y}{|x - y|} \varepsilon$.
- $\varepsilon_{x\cdot y} \leqslant 2\varepsilon$.
- $\varepsilon_{x/y} \leqslant 2\varepsilon$.
- $\varepsilon_{1/x} = \varepsilon$.

They show that addition is usually safe, since the relative error does not grow significantly. Multiplication and division are also well behaved, because the relative error roughly doubles but remains controlled. Taking the reciprocal preserves the same level of relative error. The critical case is subtraction: when two numbers are close to each other, the denominator $|x - y|$ becomes very small and the relative error can become very large. This phenomenon is known as loss of significance or cancellation, and it explains why subtracting nearly equal numbers may lead to a serious loss of accuracy in numerical computations.

Example 7.8. Let us illustrate how subtraction can lead to a dramatic loss of accuracy. Suppose that, due to rounding, we only know the numbers

$$x = 1.23456 \quad \text{and} \quad y = 1.23455$$

with an error of about $\pm 10^{-5}$. The exact difference is

$$x - y = 0.00001.$$

However, the errors in x and y are of the same order as the result itself. For instance, if the stored values are slightly changed by rounding, we might have

$$\hat{x} = 1.23457 \quad \text{and} \quad \hat{y} = 1.23454,$$

and then

$$\hat{x} - \hat{y} = 0.00003.$$

Although the original numbers changed only in the fifth decimal digit, the final result changed by a factor of three. This happens because subtraction cancels the leading digits, leaving only a very small difference where the relative error becomes huge.

7.4 The Taylor approximation and its error - the O-notation

Now we introduce two key tools that will be used throughout this text. Let us begin with the following question: how can we approximate $\sqrt{1.25}$? There is no simple direct procedure for calculating this value exactly, so we must approximate it. To do this, we study the function $f(x) = \sqrt{x}$ for $x \geqslant 0$. We will use the Taylor expansion, as introduced in Calculus. Recall that the Taylor expansion of a function f about a point a is given by

$$f(x) = f(a) + f'(a)(x - a) + \frac{1}{2!}f''(a)(x - a)^2 + \frac{1}{3!}f'''(a)(x - a)^3 + \cdots .$$

A more convenient way to write this expansion is to introduce the substitution $h = x - a$. Then $x = a + h$ and the Taylor expansion becomes

$$f(a + h) = f(a) + f'(a)h + \frac{1}{2!}f''(a)h^2 + \frac{1}{3!}f'''(a)h^3 + \cdots . \tag{14}$$

In Numerical Analysis, we usually work with the form in (14). We will use it with $a = 1$ for the function $f(x) = \sqrt{x}$. First compute the derivatives:

$$\begin{aligned} f'(x) &= \frac{1}{2}x^{-1/2}, \\ f''(x) &= -\frac{1}{4}x^{-3/2}, \\ f'''(x) &= \frac{3}{8}x^{-5/2}, \\ f^{(4)}(x) &= -\frac{15}{16}x^{-7/2}. \end{aligned}$$

Evaluating at $a = 1$, we obtain

$$\begin{aligned}\sqrt{1+h} &= 1 + \frac{1}{2}h - \frac{1}{2}\cdot\frac{1}{4}h^2 + \frac{1}{6}\cdot\frac{3}{8}h^3 - \frac{1}{24}\cdot\frac{15}{16}h^4 + \cdots \\ &= 1 + \frac{1}{2}h - \frac{1}{8}h^2 + \frac{1}{16}h^3 - \frac{5}{128}h^4 + \cdots .\end{aligned}$$

Since this is an infinite series, in practice we truncate it. For example, two possible approximations are

$$\sqrt{1+h} \approx 1 + \frac{1}{2}h \quad \text{and} \quad \sqrt{1+h} \approx 1 + \frac{1}{2}h - \frac{1}{8}h^2.$$

We can now approximate $\sqrt{1.25}$ by taking $h = 0.25 = 1/4$. Using the first approximation, we get

$$\sqrt{1.25} \approx 1 + \frac{1}{2}\cdot\frac{1}{4} = 1 + \frac{1}{8} = 1.125$$

and, using the second approximation, we get

$$\sqrt{1.25} \approx 1 + \frac{1}{8} - \frac{1}{8\cdot 16} = 1.1171875.$$

The key question is: how good are these approximations? In numerical methods, we study the approximation error, denoted by E_h. For the first approximation, $\sqrt{1+h} \approx 1 + \frac{1}{2}h$, the error is

$$E_h = -\frac{1}{8}h^2 + \frac{1}{16}h^3 - \cdots .$$

For the second approximation,

$$E_h = \frac{1}{16}h^3 - \frac{5}{128}h^4 + \cdots .$$

A natural question is which approximation is better. Intuitively, the second approximation should perform better since it includes more terms of the Taylor expansion. Before answering this question formally, let us consider the following.

Recall from Calculus that, for example, the polynomial $x^3 + 13x^2$ behaves like x^3 as $x \to \infty$, since the term $13x^2$ becomes negligible compared to x^3. More generally, let f and g be nonzero functions near a point $a \in \mathbb{R}$. We say that f is **asymptotically equivalent** to g at a, and we write

$$f(x) \sim g(x) \quad \text{as } x \to a,$$

if

$$\lim_{x\to a} \frac{f(x)}{g(x)} = 1.$$

Around 0, a similar idea applies, but now higher powers become negligible compared with lower powers. For instance,

$$h^2 + 13h^3 \sim h^2 \qquad \text{as } h \to 0,$$

since

$$\lim_{h\to 0} \frac{h^2 + 13h^3}{h^2} = \lim_{h\to 0}(1 + 13h) = 1.$$

Equivalently, one may observe that

$$\frac{h^3}{h^2} = h \longrightarrow 0 \quad \text{as } h \to 0,$$

which shows that h^3 is negligible compared to h^2 near the origin.

Returning to our error terms, for the linear approximation we have

$$E_h = -\frac{1}{8}h^2 + \frac{1}{16}h^3 - \cdots \sim -\frac{1}{8}h^2 \qquad \text{as } h \to 0.$$

For the quadratic approximation,

$$E_h = \frac{1}{16}h^3 - \frac{5}{128}h^4 + \cdots \sim \frac{1}{16}h^3 \qquad \text{as } h \to 0.$$

This shows that the second approximation is more accurate than the first one, since near 0 we have $|h^3| \leqslant |h^2|$. Therefore, the leading error term of the quadratic approximation is significantly smaller than that of the linear approximation. Observe Figure 24. There we can compare $\sqrt{1+h} - \left(1 + \frac{h}{2}\right)$ and $-\frac{h^2}{8}$ near $h = 0$. Observe that both expressions are very similar when h is close to 0, reflecting their asymptotic equivalence. However, as h approaches 1, the two expressions differ significantly.

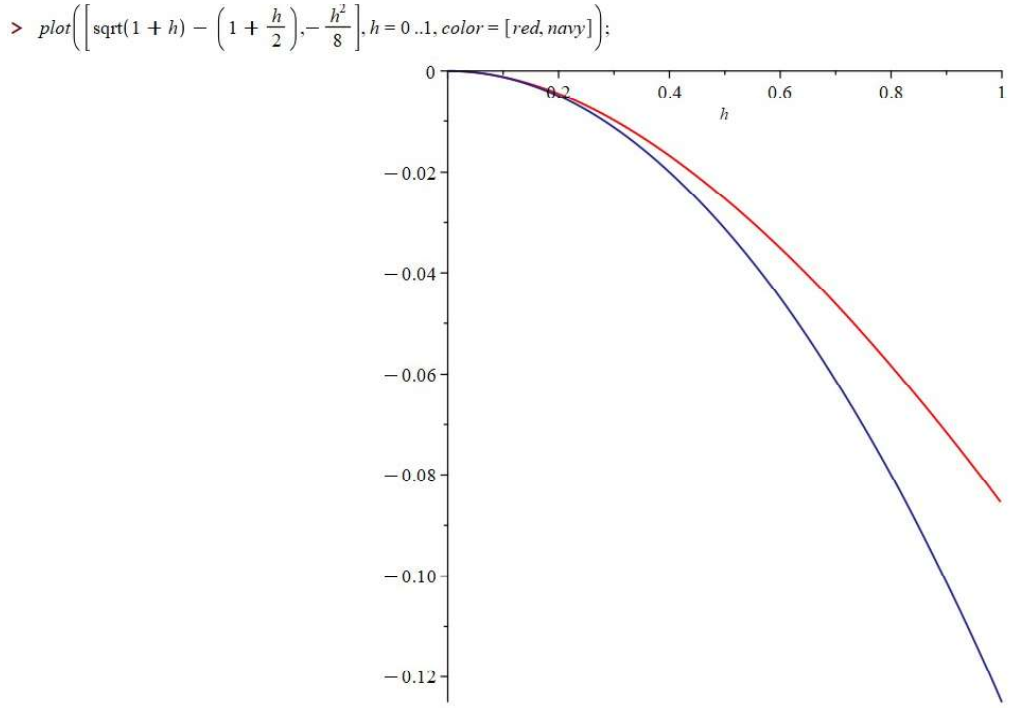

Figure 24: Comparison between $E_h = \sqrt{1+h} - (1 + \frac{h}{2})$ and $-\frac{h^2}{8}$ around 0.

On the other hand, we can compare the error $E_h = \sqrt{1+h} - (1 + \frac{h}{2} - \frac{h^2}{8})$ with $\frac{h^3}{16}$ around 0 as in Figure 25 and convince ourselves that they are asymptotically equivalent at 0.

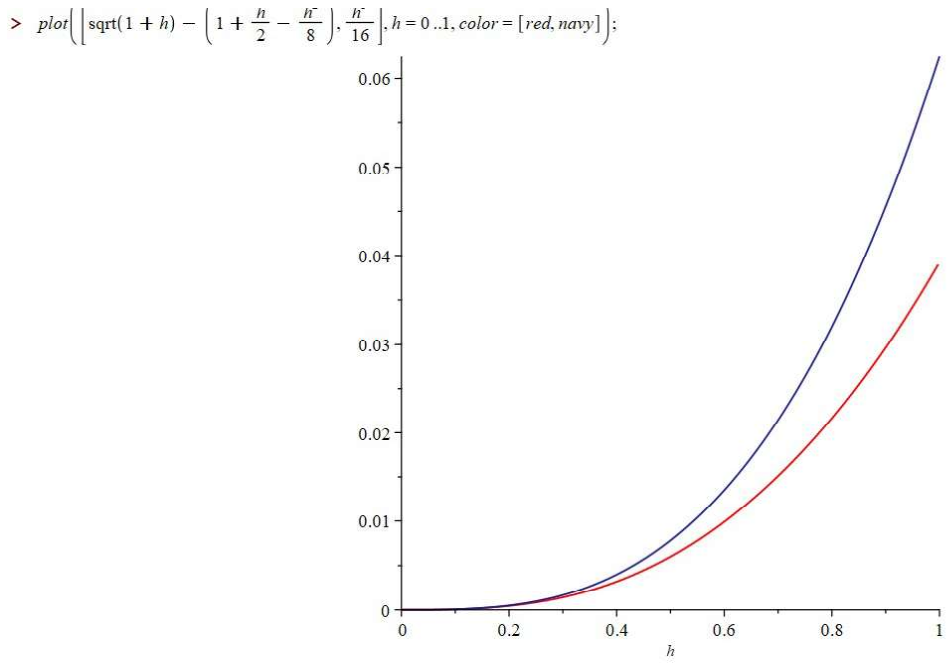

Figure 25: Comparison between $E_h = \sqrt{1+h} - (1 + \frac{h}{2} - \frac{h^2}{8})$ and $\frac{h^3}{16}$ around 0.

We can compare both errors directly and observe their behavior near 0, as shown in Figure 26. Notice that the red curve, which represents the error of the quadratic approximation, approaches zero much faster than the navy curve, which corresponds to the linear approximation. Once again, this illustrates that the quadratic approximation provides a significantly better fit near the expansion point.

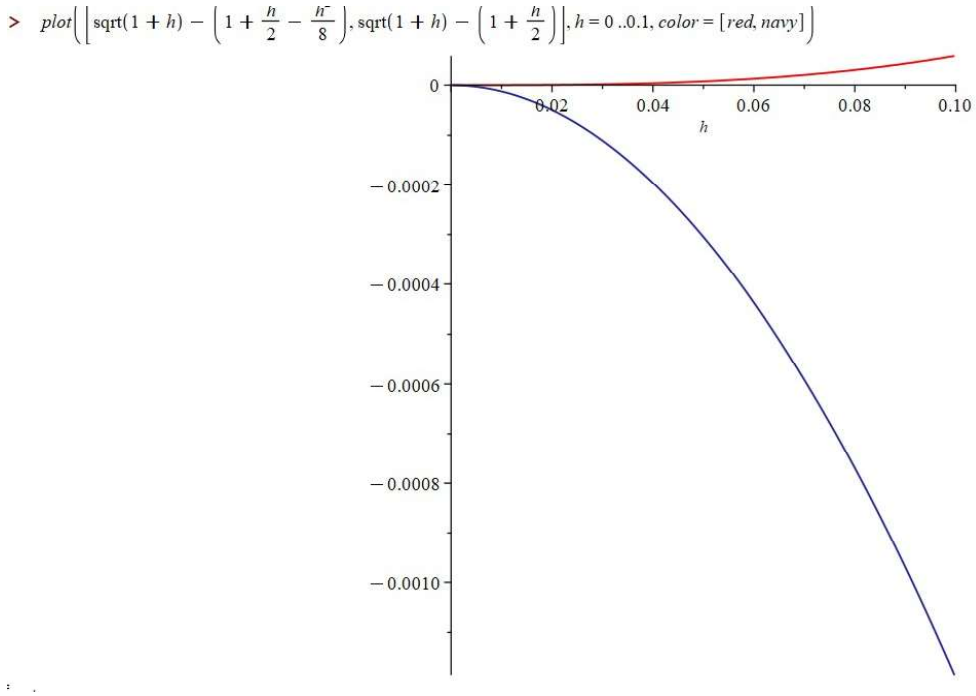

Figure 26: Comparison between both errors.

There is a very useful notation that greatly simplifies asymptotic calculations. Let us introduce the following definition.

Definition 7.9 (O-notation)**.** Let $a \in \mathbb{R}$ or $a = \pm\infty$, and let f and g be functions defined on a reduced neighborhood of a. We write

$$f = O(g) \quad \text{as } x \to a$$

if there exist a constant $C > 0$ and a reduced neighborhood P of a such that

$$|f(x)| \leqslant C\,|g(x)| \qquad \text{for all } x \in P.$$

In the previous example, we found that $E_h \sim -\frac{1}{8}h^2$ as $h \to 0$. Using the O-notation, we can write more succinctly $E_h = O(h^2)$ as $h \to 0$. Similarly, for the quadratic approximation we have $E_h = O(h^3)$ as $h \to 0$. Therefore, we may write

$$\sqrt{1+h} = 1 + \frac{1}{2}h + O(h^2)$$

for the linear approximation, and

$$\sqrt{1+h} = 1 + \frac{1}{2}h - \frac{1}{8}h^2 + O(h^3)$$

for the quadratic approximation. Notice that the O-notation allows us to perform calculations that are formally precise while avoiding unnecessary detail about higher-order terms. In fact, there are some manipulations one needs to deal with.

Fact 7.10. If $b \geqslant a \geqslant 0$, then

$$h^b = O(h^a) \text{ as } h \to 0.$$

Moreover, we have the following result.

Fact 7.11. We the following operations.

(i) For $b \geqslant a \geqslant 0$ and $\alpha, \beta \in \mathbb{R}$, $\alpha O(h^a) \pm \beta O(h^b) = O(h^a)$ as $h \to 0$.

(ii) For $a, b \geqslant 0$, $O(h^b) \cdot O(h^a) = O(h^{a+b})$ as $h \to 0$.

(iii) For $b \geqslant a \geqslant 0$, $\frac{1}{h^a}O(h^b) = O(h^{b-a})$ as $h \to 0$.

Intuitively, the statement

$$h^b = O(h^a) \quad \text{as } h \to 0 \qquad (b \geq a \geq 0)$$

expresses the idea that higher powers of h become smaller faster near 0. In other words, powers with larger exponents are negligible compared to lower powers near the origin, which is why they can be absorbed into an $O(h^a)$ term. The algebraic rules for O-notation simply reflect the way dominant terms behave under basic operations. In sums, the smallest power of h dominates, so adding a higher-order term does not change the overall order. In products, powers add, so multiplying $O(h^a)$ and $O(h^b)$ naturally gives $O(h^{a+b})$. Finally, dividing by h^a reduces the power by a, which explains why $\frac{1}{h^a}O(h^b) = O(h^{b-a})$. Altogether, these rules allow us to manipulate asymptotic expressions just like ordinary powers while keeping track only of the leading order of magnitude.