

## 9 Practice #3: Wednesday, March 4th, 2026

### 9.1 Numerical Analysis: error, derivation and integration

In this class we will solve some problems about Numerical Analysis. More specifically, we will be solving problems regarding errors, and numerical derivation and integration.

### 9.2 Problem 1

We would like to approximate the function  $\cos(x)$  near 0. To do this, we use the Taylor expansion. Recall that, using  $O$ -notation, the Taylor expansion of a function  $f$  around  $x = a$  is given by

$$f(a+h) = f(a) + f'(a)h + \frac{1}{2}f''(a)h^2 + \frac{1}{3!}f'''(a)h^3 + \frac{1}{4!}f^{(4)}(a)h^4 + O(h^5).$$

In our case,  $f(x) = \cos(x)$ , and therefore

$$\cos(a+h) = \cos(a) - \sin(a)h - \frac{1}{2!}\cos(a)h^2 + \frac{1}{3!}\sin(a)h^3 + \frac{1}{4!}\cos(a)h^4 + O(h^5).$$

Since we want an approximation near 0, we set  $a = 0$  and obtain

$$\cos(h) = 1 - \frac{1}{2!}h^2 + \frac{1}{4!}h^4 - \frac{1}{6!}h^6 + \frac{1}{8!}h^8 + O(h^{10}). \quad (16)$$

Hence, near  $h = 0$  we may write  $\cos(h) \approx 1$ , or equivalently  $\cos(h) \sim 1$ , which means that  $\cos(h) = 1 + O(h^2)$  as  $h \rightarrow 0$ . This approximation introduces an error given by

$$E_h = -\frac{1}{2}h^2 + \frac{1}{4!}h^4 + O(h^6).$$

Thus, the leading term of the error is  $E_h \sim -\frac{1}{2}h^2$ , meaning that

$$\lim_{h \rightarrow 0} \frac{E_h}{-\frac{1}{2}h^2} = 1.$$

In particular, the error satisfies  $E_h = O(h^2)$ , i.e., there exists a constant  $c > 0$  such that  $|E_h| \leq c \cdot h^2$  for sufficiently small  $h > 0$ .

**Remark 9.1.** Although it is formally correct to write  $E_h = O(h)$ , since

$$E_h = 0 \cdot h - \frac{1}{2}h^2 + \frac{1}{4!}h^4 + O(h^6),$$

this notation loses important information about the true behavior of the error. The statement  $E_h \sim -\frac{1}{2}h^2$  is much more precise, since it captures the dominant term as  $h \rightarrow 0$ .

Suppose now that we want an approximation of  $\cos(x)$  near 0 with error  $O(h^8)$ . Using (16), we obtain

$$\cos(h) \sim 1 - \frac{1}{2}h^2 + \frac{1}{4!}h^4 - \frac{1}{6!}h^6$$

with remainder of order  $O(h^8)$ .

Let us next consider a quadratic approximation of  $\cos(x)$  near 0 (see Figure 40). In this case, as  $h \rightarrow 0$ ,

$$\cos(h) \sim 1 - \frac{1}{2}h^2.$$

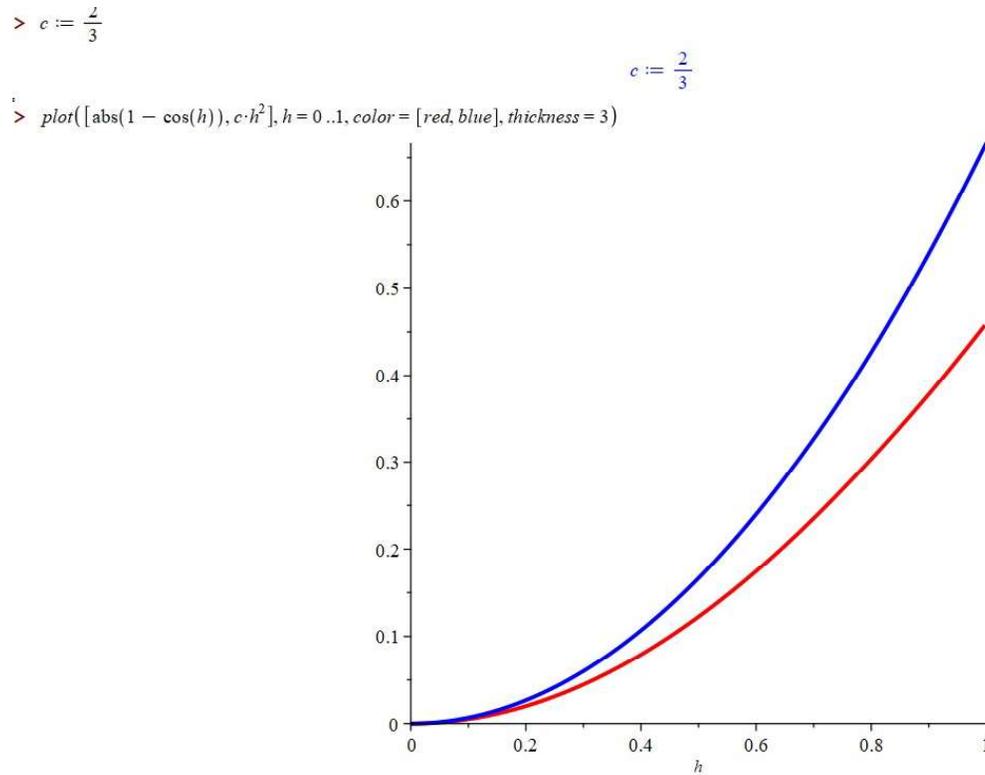


Figure 40: Comparison between the absolute error  $1 - \cos(h)$  and the parabola  $c \cdot h^2$  with  $c = 2/3$ .

How about linear approximations? In this case (see Figure 41), we simply write  $\cos(h) \sim 1$ , which has error  $O(h^2)$ . If we are asked for an approximation with error  $O(h^3)$ , we again use (16): since the quadratic approximation has error  $O(h^4)$ , it also satisfies the weaker requirement  $O(h^3)$ . Hence,

$$\cos(h) \sim 1 - \frac{1}{2}h^2$$

as  $h \rightarrow 0$ .

```
> plot([abs(1 - cos(h)), c·h², 1·h], h = 0..1, color = [red, blue, green], thickness = 3)
```

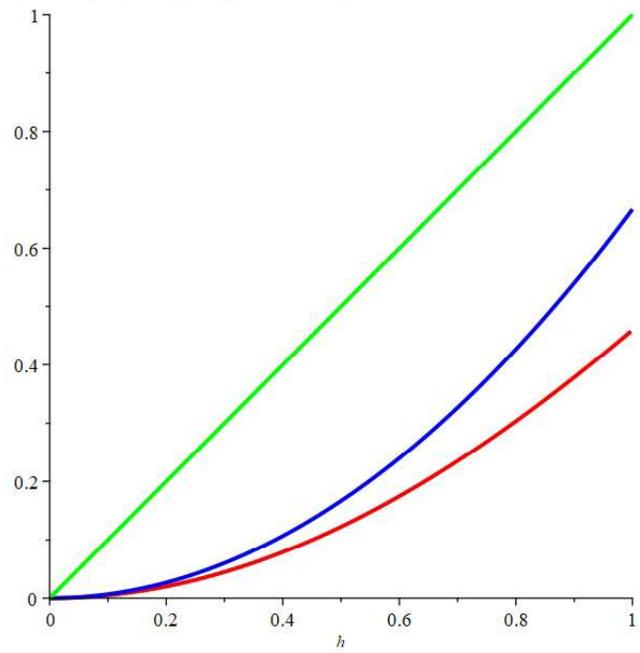


Figure 41: Comparison between the absolute error  $1 - \cos(h)$  (in red), the parabola (in blue), and the linear approximation (in green).

### 9.3 Problem 2

Suppose that  $x \in \mathbb{R}$  is approximated by  $\hat{x}$ . As we know from theory, this introduces an absolute error  $E_x$  and a relative error  $\varepsilon_x$ . We would like to understand how the cosine function reacts to this perturbation. In other words, we want to analyze the errors produced when computing  $\cos(\hat{x})$  instead of  $\cos(x)$ .

Notice that the absolute error is given by  $E_{\cos(x)} = \cos(x) - \cos(\hat{x})$ . As  $E_x = x - \hat{x}$ , we have that  $\hat{x} = x - E_x$ . In this case, we have that

$$E_{\cos(x)} = \cos(x) - \cos(\hat{x}) = \cos(x) - \cos(x - E_x).$$

Now, we can use the Taylor expansion for  $\cos(x - E_x)$  (notice that here we have that  $a = x$  and  $h = -E_x$ ). As

$$\cos(x - E_x) = \cos(x) - \sin(x) \cdot (-E_x) - \frac{1}{2} \cos(x) \cdot (-E_x)^2 + O(E_x^3)$$

we have that

$$\begin{aligned} E_{\cos(x)} &= \cos(x) - \cos(x - E_x) \\ &= \cos(x) - \left[ \cos(x) - \sin(x) \cdot (-E_x) - \frac{1}{2} \cos(x) \cdot (-E_x)^2 + O(E_x^3) \right] \\ &= -\sin(x) \cdot E_x + O(E_x^2). \end{aligned}$$

This means that  $E_{\cos(x)} \approx \sin(x) \cdot E_x$ . As  $|\sin(x)| \leq 1$ , we have that  $|E_{\cos(x)}| \leq |E_x|$ . Now, concerning the relative error we have that

$$\varepsilon_{\cos(x)} = \frac{|E_{\cos(x)}|}{|\cos(x)|} \approx \frac{|-\sin(x)E_x|}{|\cos(x)|} = \frac{|\sin(x)| \cdot |x|}{|\cos(x)|} \cdot \frac{|E_x|}{|x|} = |x| \cdot |\tan(x)| \cdot \varepsilon_x.$$

This means that

$$|\varepsilon_{\cos(x)}| \approx |x| \cdot |\tan(x)| \cdot \varepsilon_x. \tag{17}$$

Let us interpret (17). Suppose we enter a number  $x$  into a computer. Due to rounding, the machine actually uses an approximation  $\hat{x}$ , producing a relative error  $\varepsilon_x$ . The relation above shows that the relative error in computing  $\cos(\hat{x})$  may be amplified by the factor  $|x| \cdot |\tan(x)|$ . In particular, since  $|x|$  can grow without bound and  $\tan(x)$  becomes large near  $\pi/2$ , the relative error in the cosine can become very large (around  $\pi/2$ , for instance), meaning that the computation may be poorly conditioned in those regions.

Let us perform a simple experiment in Maple. Keep in mind that  $\pi/2 \approx 1.5707$ . Suppose that  $x := 1.5$ , and consider two approximations defined by  $xp := 1.01 \cdot x$  and  $xm := 0.99 \cdot x$ , meaning that we perturb  $x$  by 1%. In the next two lines of the code (see Figure 42), we confirm that the relative errors in the input are indeed 1% (corresponding to expressions (5) and (6)). However, expressions (7) and (8) show the reliability of the cosine evaluated at these approximations. We observe that, in both cases, the relative error in the cosine is approximately 21%.

```

=
> #PROBLEM 2
> x := 1.5
x := 1.5 (2)
=
> xp := 1.01 · x
xp := 1.515 (3)
=
> xm := 0.99 · x
xm := 1.485 (4)
=
>  $\frac{\text{abs}(x - xp)}{x}$ 
0.01000000000 (5)
=
>  $\frac{\text{abs}(x - xm)}{x}$ 
0.01000000000 (6)
=
>  $\frac{\text{abs}(\cos(x) - \cos(xp))}{\text{abs}(\cos(x))}$ 
0.2116258651 (7)
=
>  $\frac{\text{abs}(\cos(x) - \cos(xm))}{\text{abs}(\cos(x))}$ 
0.2114008694 (8)

```

Figure 42: Code on Maple which provides the relative error of the cosine.

This experiment clearly shows that small relative errors in the input can be significantly amplified when computing  $\cos(x)$ . We invite the reader to test what happens when we go even closer to  $\pi/2$ . For instance, when  $x := 1.57$ , we get the the relative error in the cosine is approximately 200%! In fact, in this last case,  $xp$  is 1.5857 and, while  $\cos(1.57)$  equals 0.0007963267107, we have that  $\cos(xp)$  is -0.01490312148, a totally different number.

## 9.4 Problem 3

Suppose that  $f(x) = \frac{1}{x^2}$  and let  $a = 1$ . We know that  $(x^{-2})' = -2x^{-3}$ , and therefore  $f'(1) = -2$ . Let us now pretend that we do not know this derivative and instead use the forward difference method to approximate it. We have that

$$f'(1) \approx \frac{f(1+h) - f(1)}{h} = \frac{\frac{1}{(1+h)^2} - \frac{1}{1^2}}{h}.$$

Let us take  $h = 0.01$ , which should provide a reasonable approximation of the derivative. In this case,

$$f'(1) \approx \frac{\frac{1}{(1.01)^2} - \frac{1}{1^2}}{0.01}.$$

Usually, we do not know the exact value of the derivative and must rely on error estimates to judge the quality of the approximation. In our case, however, we can compute the error explicitly as follows

$$E_{0.01} = -2 - \frac{\frac{1}{(1.01)^2} - \frac{1}{1^2}}{0.01} = -2 - \left( \frac{100}{(1.01)^2} - 100 \right).$$

The relative error is then given by

$$\varepsilon_{0.01} = \frac{|E_{0.01}|}{|-2|}.$$

Let us check Figure 43.

```

=> #PROBLEM 3
=> x := 'x' #it assigns the name x to itself
                                     x := x
=
=> f := 1/x^2
                                     f := 1/x^2
=
=> a := 1
                                     a := 1
=
=> d := eval(diff(f, x), x = a)
                                     d := -2
=
=> h := 0.01
                                     h := 0.01
=
=> (eval(f, x = a + h) - eval(f, x = a)) / h
                                     -1.970395060
=
=> E := d - %
                                     E := -0.029604940
=
=> eps := abs(E) / abs(d)
                                     eps := 0.01480247000
=

```

Figure 43: Calculations for the absolute and relative errors using the forward difference method.

Notice, however, that if we use the central difference method, the relative error is significantly smaller, as shown in Figure 44. Since the relative error is approximately 0.0002, this suggests that we can trust about three digits of the approximation, which is confirmed by the results displayed in the same code.

```

> h := 0.01 :
  (eval(f, x = a + h) - eval(f, x = a - h)) /
  2 * h
E := d - %;
eps := abs(E) /
      abs(d);
                                     -2.000400080
E := 0.000400080
eps := 0.0002000400000

```

Figure 44: Absolute and relative errors using the central difference method.

## 9.5 Problem 4

We would like to approximate the integral

$$I = \int_1^7 e^x dx.$$

Once again, let us assume that we do not know how to compute it exactly. We will approximate it using a partition with  $n = 3$  subintervals and compare several numerical methods.

Let us begin with the left-rectangle method. The step size, that is, the length of each subinterval, is

$$h = \frac{7 - 1}{3} = 2.$$

This divides the interval  $[1, 7]$  into three subintervals. Using the left endpoints, we obtain the approximation

$$I \approx 2e^1 + 2e^3 + 2e^5 = 2 \cdot (e + e^3 + e^5).$$

Recall that the order of this method is 1, that is,  $E_h = O(h)$ , which is the same as writing  $|E_h| \leq c \cdot h$  for some  $c > 0$  and small values of  $h$ .

```
> IntNumeric(exp(x), x=1..7, method=left, numsteps=3, output=graph);  
Using step size 2.000000.
```

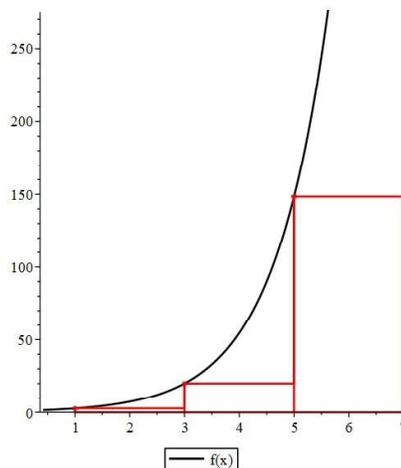


Figure 45: Left-rectangles method.

Let us now apply the right-rectangle method. In this case, we obtain

$$I \approx 2e^3 + 2e^5 + 2e^7 = 2 \cdot (e^3 + e^5 + e^7).$$

As discussed earlier, this method has the same order of accuracy as the left-rectangle method.

```
> IntNumeric(exp(x), x=1..7, method=right, numsteps=3, output=graph);
Using step size 2.000000.
```

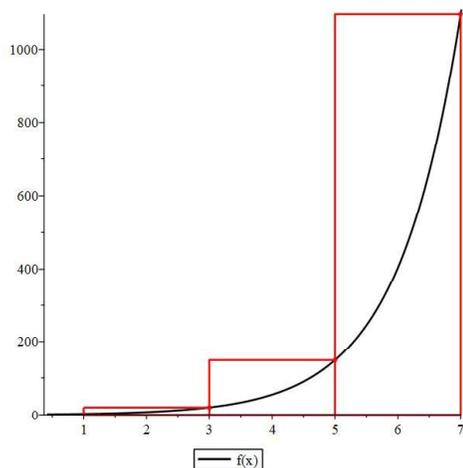


Figure 46: Right-rectangles method.

Let us go now for the trapezoid method. In this case, we have

$$I \approx 2 \cdot \frac{1}{2} \cdot (e^1 + e^3) + 2 \cdot \frac{1}{2} \cdot (e^3 + e^5) + 2 \cdot \frac{1}{2} \cdot (e^5 + e^7) = e^1 + 2e^3 + 2e^5 + e^7.$$

Recall that the order of this method is 2, that is,  $|E_h| \leq c \cdot h^2$  for some  $c > 0$  and small values of  $h$ .

```
> IntNumeric(exp(x), x=1..7, method=trapezoid, numsteps=3, output=graph);
Using step size 2.000000.
```

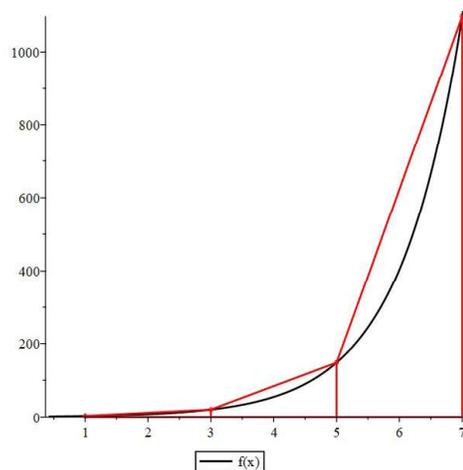


Figure 47: Trapezoid method.

## 9.6 Problem 5

Suppose we have computed, using Maple, the absolute error of the left-rectangle approximation for

$$I = \int_0^1 \cos(x) dx,$$

and obtained  $E_{0.2} \approx 0.0432$  (see Figure 48) with  $n = 5$ , which corresponds to a step size  $h = 0.2$ .

```
> IntNumeric(cos(x), x=0..1, method=left, numsteps=5, output=totalerror);  
Using step size 0.200000.  
  
0.043162994412297579764382824249235291
```

Figure 48: Absolute error for the approximation of the integral  $\int_0^1 \cos(x) dx$ .

Let us consider the following questions.

- (a) What error should we expect if we double the number of subintervals, that is, if  $n = 10$ ? In this case the step size becomes  $h = 0.1$ . Since the left-rectangle method is of order 1, we know that  $|E_h| \leq c \cdot h$  for some constant  $c > 0$  and sufficiently small  $h$ . Therefore, the error behaves approximately like  $E_h \approx c \cdot h$ . It follows that

$$E_{h/2} \approx c \left( \frac{h}{2} \right) = \frac{1}{2} ch = \frac{1}{2} E_h.$$

Hence, halving the step size roughly halves the error. In particular,

$$E_{0.1} \approx 0.0216.$$

- (b) What value of  $h$  should we choose in order to obtain an error of approximately  $E_h = 0.001$ ? Since the method is first order, we have  $E_h \approx c \cdot h$ . Using the known value  $E_{0.2} \approx 0.0432$ , we estimate

$$E_{0.2} \approx c \cdot 0.2,$$

and therefore

$$c \approx \frac{E_{0.2}}{0.2} = \frac{0.0432}{0.2} = 0.216.$$

Hence, the step-size  $\tilde{h}$  required to achieve an error close to 0.001 is obtained from

$$0.001 \approx 0.216 \cdot \tilde{h},$$

that is,

$$\tilde{h} \approx \frac{0.001}{0.216}.$$

The recommended step size for part (b) is therefore  $\tilde{h} \approx 0.0046$ .

There is a second way to solve item (b). As observed earlier, for a first-order method we have

$$E_{h/a} \approx \frac{1}{a} E_h$$

for some positive number  $a$ . Therefore, if we want an error of approximately 0.001, we write

$$0.001 \approx \frac{1}{a} E_{0.2}.$$

Hence,

$$a \approx \frac{E_{0.2}}{0.001}.$$

The new step size  $\tilde{h}$  is then

$$\tilde{h} = \frac{h}{a} = h \cdot \frac{1}{a} = \frac{0.001}{E_{0.2}} \cdot 0.2 \approx 0.0046.$$

Thus, this alternative argument leads to the same recommended step size.