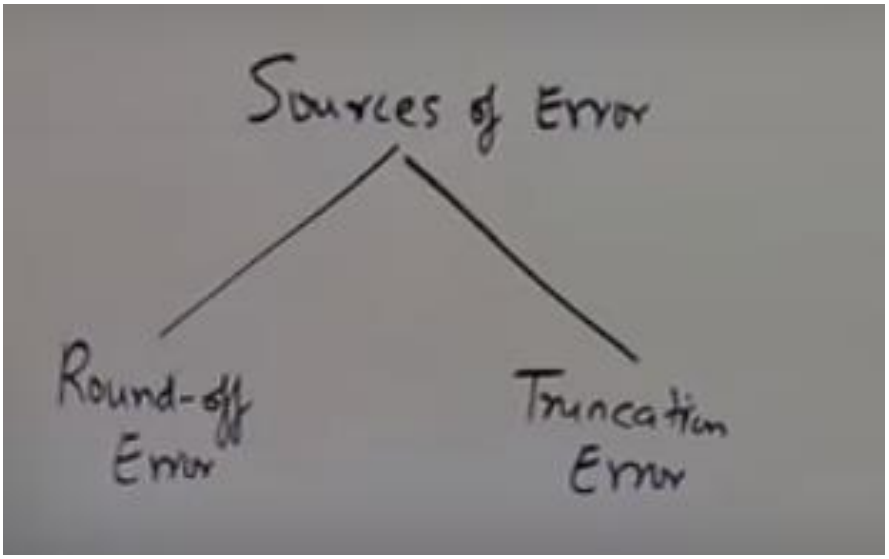
Lecture: Round-off Error: Definition and Examples

Summary: There are two sources of error - one comes from approximating numbers and another from approximating mathematical procedures. In this lecture, the error, round-off error, that is a result of approximating numbers is defined and shown through an example.

Learning Objectives: After this lecture, you will be able to identify and calculate one of the two sources of errors in numerical methods - round-off errors.

SOURCE OF ERROR: Round-off Error

In this segment we're going to talk about round-off errors. **There are several possibilities of error whenever you're going to use numerical methods**, but we want to concentrate here on **just two errors, one is the round-off error and the other is the truncation error.**
So those are the sources of error which we are going to talk about, because those are the ones which are coming from something on which you may or may not have as much control as other errors, like for example if you have made a mistake in programming, or if your logic is wrong, those are not the kind of errors which we are talking about when we talk about numerical methods.
So you're going to have two sources of error, which you are going to have. One is round-off error and the other one is called truncation error. And let's go ahead and concentrate on what round-off error is.



Now **round-off error** is defined as follows, it is basically the error which comes from ***error created due to approximate representation of numbers***. So the round-off error is simply the error created by the approximate representation of numbers, because in a computer you'll be able to only represent a number only so approximately. For example, if you have a number like 1 divided by 3, and you had a

six significant digit computer let's suppose in the decimal notation, then this can be only approximated as 0.333333 a simple rational number like 1 divided by 3 cannot be written exactly in the decimal format. So the amount of round-off error which you are getting here is the difference between the value of 1 divided by 3 and the value of 0.333333. So in this case, this error is 0.0000003333 and so on and so forth.

You're going to get similar round-off errors from other numbers also, like, you may have pi, that also cannot be represented exactly, even in a decimal format, and then square root of 2, things like that. So you're finding out there are many, many numbers, individual numbers, like 1 divided by 3, or pi, or square root of 2, which cannot be represented exactly in a computer.

So that's why this creates the round-off error, the round-off error is the difference between what you want to, what you want to be able to approximate, of what you want to be able to denote, and what you are able to get as its approximation. So that's the, that's what we call as round-off error. So that's the end of this particular segment here.

Truncation Error: Definition

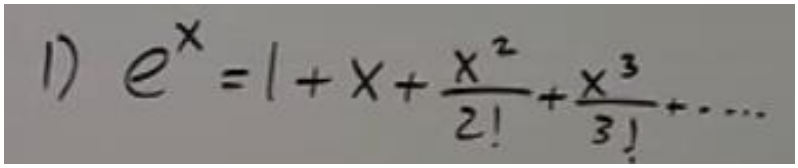
Summary: There are two sources of error - one comes from approximating numbers and another from approximating mathematical procedures. In this lecture, the error, called truncation error, that is a result of approximating mathematical procedures is defined.

Learning Objectives: After this lecture, you will be able to identify and calculate one of the two sources of errors in numerical methods - truncation errors.

In this segment we're going to talk about truncation error. I want to say that we have sources of error in numerical methods. And we're not talking about the errors which are created by writing the wrong program, so far as logic or syntax is concerned, but the errors which are inherent when you are using numerical methods, and one is called the round-off error, and the other one is called the truncation error. So in this segment we're going to talk about, what does it mean when we say that, hey, we are having a truncation error? So let's go ahead and write down the definite of truncation error.

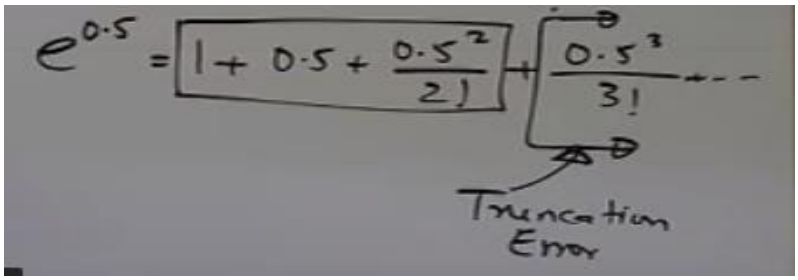
Truncation error is defined as the error created by truncating a mathematical procedure

Now, some people don't like the word truncating in the definition of truncation error itself, because they say that it doesn't mean much. So I'm going to cross it off there, and I'm going to say, hey, approximating a mathematical procedure. So if you're going to approximate a mathematical procedure, it is going to create some error, and that error is associated with truncation error. **Please don't think that truncation error is something which is associated with rounding off numbers**. It is, truncation error is related to the error which is created by approximating, not numbers, but a mathematical procedure. Examples of truncation error as follows, so let's look at some examples. In this segment I'm just going to enumerate the examples, and then we will have three more segments, which will show each individual example with some numbers.



One of the examples is, let's suppose you are using Maclaurin series. The Maclaurin series for e to the power x is 1 plus x plus x squared by factorial 2 plus x cubed by factorial 3, and plus so on and so forth.

So you have infinite number of terms in this particular series for e to the power x. So if you want to calculate e to the power x at some value of x, let's suppose. And let's suppose if somebody says, hey, calculate e to the power 0.5, so I would say 1 plus 0.5 plus 0.5 squared divided by 2 factorial plus 0.5 cubed, factorial 3, and so on and so forth. Now you can realize that since there are infinite terms in this Maclaurin series to calculate e to the power 0.5, I don't have the privilege or the luxury to use all the terms, all the infinite number of terms which I have in that particular series. If somebody were to say, hey, I'm going to use only the first three terms of the series to calculate my value of e to the power 0.5.



So what's happening is that you are not accounting for these other infinite terms after the fourth term, you're not accounting for those terms at all in your calculation e to the power 0.5, and whatever is leftover is your truncation error. Because what you did was, the original mathematical procedure required you to use infinite number of terms, but you are using only three terms, so whatever is leftover is truncation error, because you have basically truncated a procedure, a mathematical procedure requiring you to use infinite number of terms, and you're using only a few terms out of that . . . out of that series there. Now what happens is that, in the past, I used to give only this as an example of truncation error, and many students would think that truncation error is something which is only related to series. But there are other examples where you will see how a mathematical procedure gets truncated. So let's look at that.

For both types, the relationship between the exact, or true, result and the approximation can be formulated as

$$\text{True value} = \text{approximation} + \text{error} \tag{3.1}$$

By rearranging Eq. (3.1), we find that the numerical error is equal to the discrepancy between the truth and the approximation, as in

$$E_t = \text{true value} - \text{approximation} \tag{3.2}$$

$$\text{True fractional relative error} = \frac{\text{true error}}{\text{true value}}$$

where, as specified by Eq. (3.2), error = true value − approximation. The relative error can also be multiplied by 100 percent to express it as

$$\varepsilon_t = \frac{\text{true error}}{\text{true value}} 100\% \tag{3.3}$$

where ε_t designates the true percent relative error.

Calculation of Errors

Problem Statement. Suppose that you have the task of measuring the lengths of a bridge and a rivet and come up with 9999 and 9 cm, respectively. If the true values are 10,000 and 10 cm, respectively, compute (*a*) the true error and (*b*) the true percent relative error for each case.

Solution.

(a) The error for measuring the bridge is [Eq. (3.2)]

$$E_t = 10{,}000 - 9999 = 1 \text{ cm}$$

and for the rivet it is

$$E_t = 10 - 9 = 1 \text{ cm}$$

(b) The percent relative error for the bridge is [Eq. (3.3)]

$$\varepsilon_t = \frac{1}{10{,}000} 100\% = 0.01\%$$

and for the rivet it is

$$\varepsilon_t = \frac{1}{10} 100\% = 10\%$$

Notice that for Eqs. (3.2) and (3.3), E and e are subscripted with a t to signify that the error is normalized to the true value. In Example 3.1, we were provided with this value. However, in actual situations such information is rarely available. For numerical methods, the true value will be known only when we deal with functions that can be solved analytically. Such will typically be the case when we investigate the theoretical behavior of a particular technique for simple systems. However, in real-world applications, we will obviously not know the true answer a priori. For these situations, an alternative is to normalize the error using the best available estimate of the true value, that is, to the approximation itself, as in

$$\varepsilon_a = \frac{\text{approximate error}}{\text{approximation}} 100\% \tag{3.4}$$

where the subscript a signifi es that the error is normalized to an approximate value. Note also that for real-world applications, Eq. (3.2) cannot be used to calculate the error term for Eq. (3.4). One of the challenges of numerical methods is to determine error estimates in the absence of knowledge regarding the true value. For example, certain numerical methods use an iterative approach to compute answers. In such an approach, a present approximation is made on the basis of a previous approximation. This process is performed repeatedly, or iteratively, to successively compute (we hope) better and better approximations. For such cases, the error is often estimated as the difference between previous and current approximations. Thus, percent relative error is determined according to

$$\varepsilon_a = \frac{\text{current approximation} - \text{previous approximation}}{\text{current approximation}} 100\% \tag{3.5}$$

Error Estimates for Iterative Methods

Problem Statement. In mathematics, functions can often be represented by infinite series. For example, the exponential function can be computed using

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} \tag{E3.2.1}$$

Thus, as more terms are added in sequence, the approximation becomes a better and better estimate of the true value of e^x. Equation (E3.2.1) is called a *Maclaurin series expansion.*

Starting with the simplest version, $e^x = 1$, add terms one at a time to estimate $e^{0.5}$. After each new term is added, compute the true and approximate percent relative errors with Eqs. (3.3) and (3.5), respectively. Note that the true value is $e^{0.5} = 1.648721\ldots$. Add terms until the absolute value of the approximate error estimate ε_a falls below a prespecified error criterion ε_s conforming to three significant figures.

Solution. First, Eq. (3.7) can be employed to determine the error criterion that ensures a result is correct to at least three significant figures:

$$\varepsilon_s = (0.5 \times 10^{2-3})\% = 0.05\%$$

Thus, we will add terms to the series until ε_a falls below this level.

The first estimate is simply equal to Eq. (E3.2.1) with a single term. Thus, the first estimate is equal to 1. The second estimate is then generated by adding the second term, as in

$$e^x = 1 + x$$

or for $x = 0.5$,

$$e^{0.5} = 1 + 0.5 = 1.5$$

This represents a true percent relative error of [Eq. (3.3)]

$$\varepsilon_t = \frac{1.648721 - 1.5}{1.648721} 100\% = 9.02\%$$

Equation (3.5) can be used to determine an approximate estimate of the error, as in

$$\varepsilon_a = \frac{1.5 - 1}{1.5} 100\% = 33.3\%$$

Because ε_a is not less than the required value of ε_s, we would continue the computation by adding another term, $x^2/2!$, and repeating the error calculations. The process is continued until $\varepsilon_a < \varepsilon_s$. The entire computation can be summarized as

Terms	Result	ε_t (%)	ε_a (%)
1	1	39.3	
2	1.5	9.02	33.3
3	1.625	1.44	7.69
4	1.645833333	0.175	1.27
5	1.648437500	0.0172	0.158
6	1.648697917	0.00142	0.0158