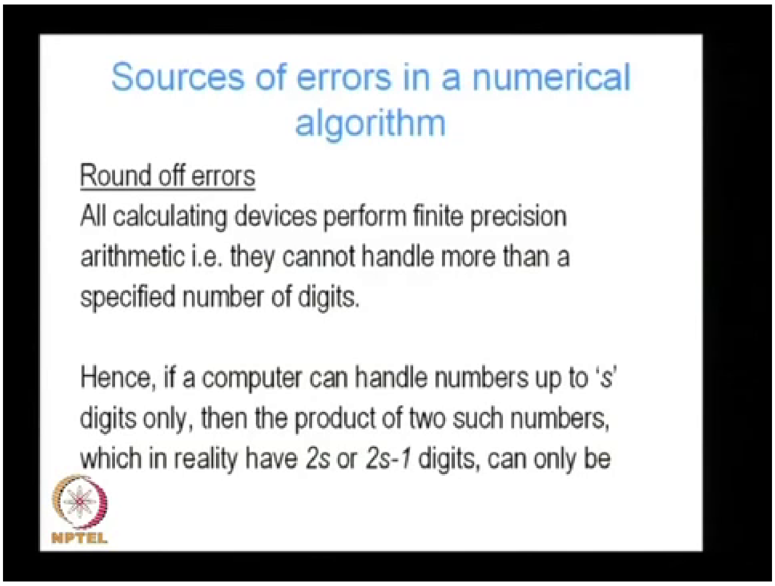
Numerical Methods in Civil Engineering
Prof. Arghya Deb
Department of Civil Engineering
Indian Institute of Technology, Kharagpur

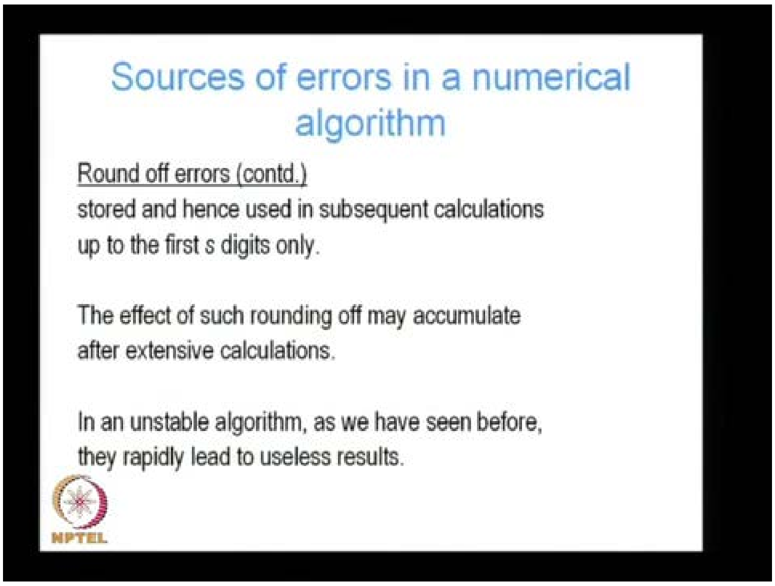
Lecture - 2
Error Analysis

(Refer Slide Time: 00:21)



Since, civil engineering we are going to talk about Error Analysis. First we want to talk about the sources of errors in a numerical algorithm, the first type of error we are going to talk about a round off errors. All calculating devices perform finite precision arithmetic that is, they cannot handle more than a specified number of digits. For instance, if a computer can handle numbers up to s digits only where s is a number then the product of two such numbers which in reality have 2s or 2s minus 1 digits. For instance if I have a number that has two digits for instance 12 if I multiplied by another number which has two digits for instance 12 again I get a number with 144 digits, which has got two times 2 minus 1 that is 2 s minus 1 that is three digits can one.

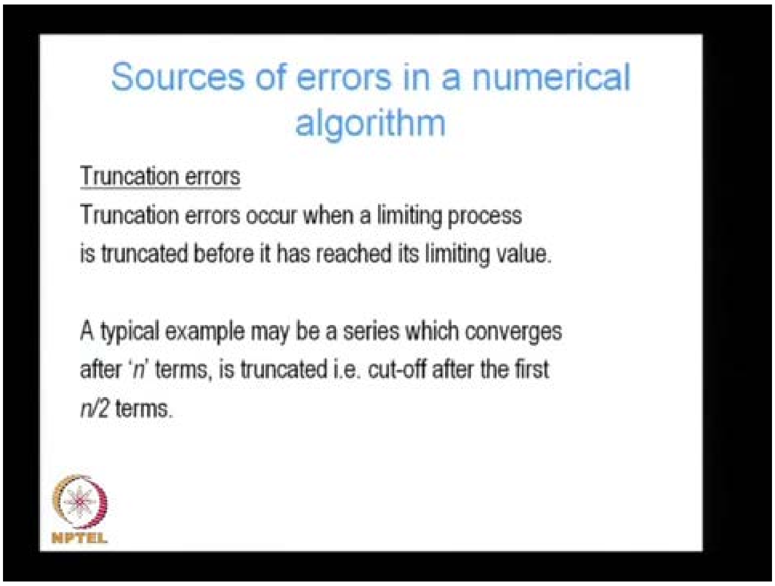
(Refer Slide Time: 01:34)



For a computer however, we will only store it as in s digit number and use it in subsequent calculation using only the first 8 digits of that number. The effect of such rounding off may accumulate after extensive calculations, for instance if we are doing a series of products every time we do a product we instead of storing 2 s digits if each number has s digits instead of storing 2 s digits we are storing s digits.

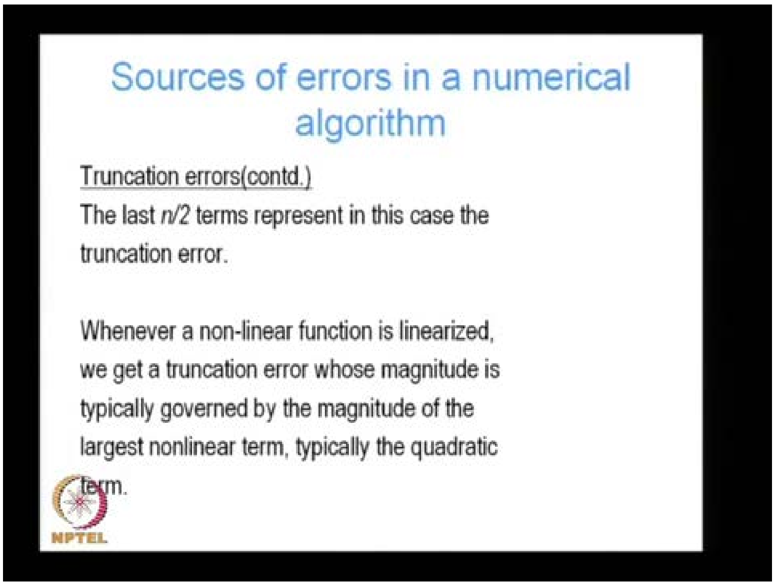
So, each time we are losing a certain accuracy, so each time we are introducing inaccuracy and as this inaccuracy is by law they made these may become significant. In an unstable algorithm as we have seen before in our first lecture, this accumulation of this round off errors lead to useless and or grossly in accurate results.

(Refer Slide Time: 02:33)



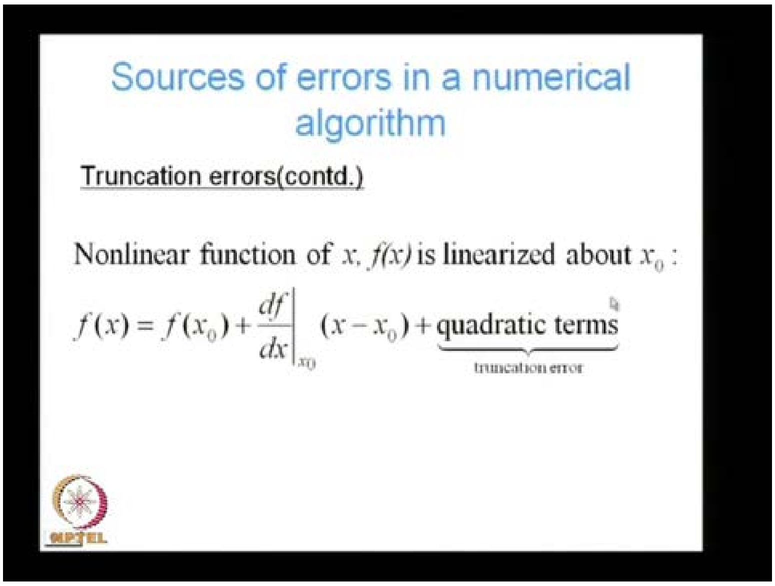
So, first type of errors we talked about was round off errors, the second type of error we are going to talk about are known as truncation errors. Truncation errors occur when a limiting process is truncated before it has reached it is limiting value, a typical example of this may be a series which of suppose converges after n terms which suppose we truncated that is we cut it off after the first n by 2 terms. So, if the series converges after n equal to suppose 10 terms suppose we cut it off after we compute only five terms in the series that is n by 2 terms.

(Refer Slide Time: 03:24)



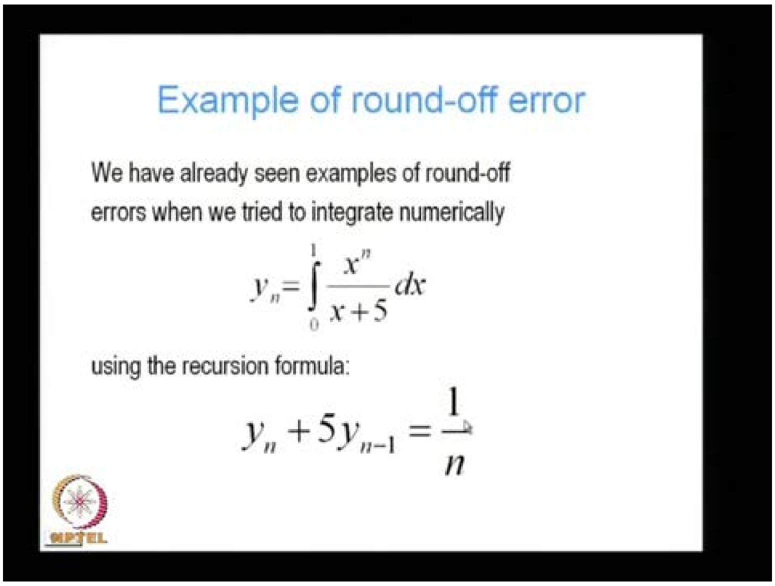
In that case the last n by 2 terms represent the truncation error the last five terms in this series, where n is equal to 10 are not being accounted for, so they represent the truncation error. For instance, whenever a non-linear function is linearize we get a truncation error, whose magnitude is typically governed by the magnitude of the largest non linear term typically the quadratic term. When we linearize the function we ignore all terms which higher than, linear higher than first ordered term. And if the first non linear term is the quadratic term then the error has also dimension are also second order.

(Refer Slide Time: 04:15)



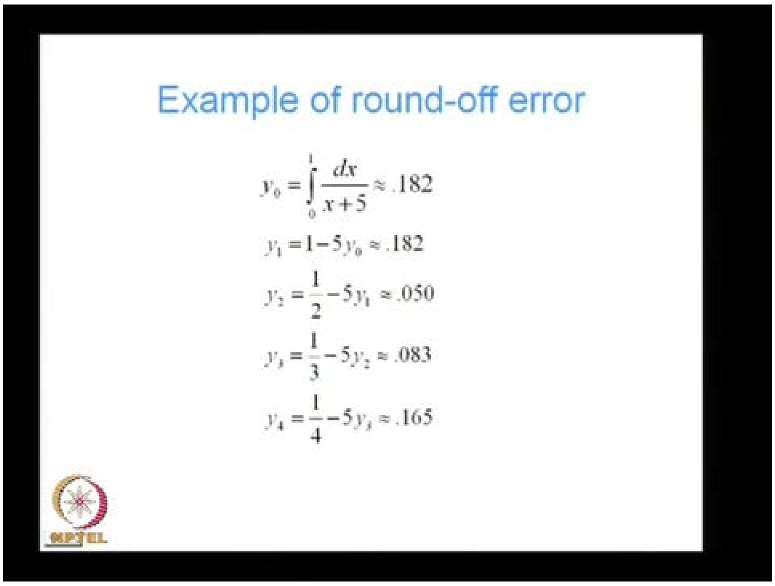
For instance here, if we are considering a non linear function of x f of x which is linearized about the value x 0, we can write f of x is equal to f of x 0 plus the derivative of x f with respect to x evaluated at x 0 times x minus x 0, which is the linear term plus the quadratic terms, which represent the truncation error.

(Refer Slide Time: 04:50)



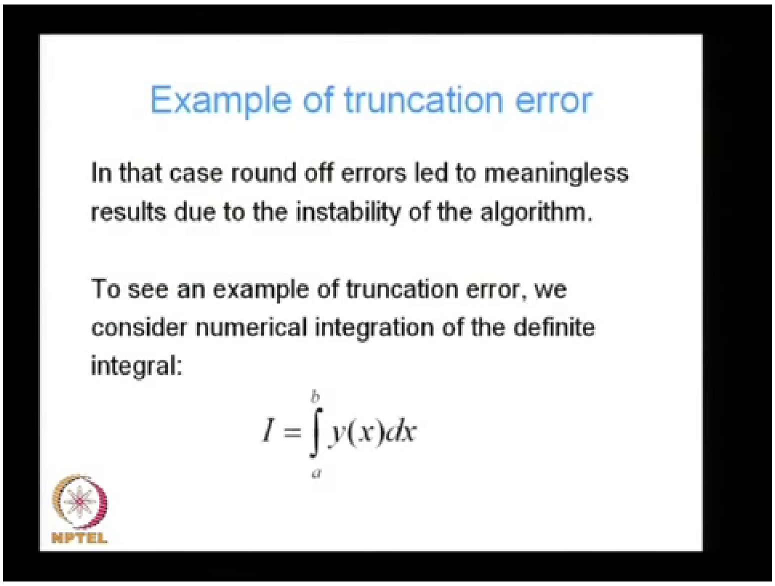
Examples, next we will consider examples of round off and truncation errors we have already seen examples of round off errors, when we tried to integrate numerically y n equal to x n by x plus 5 over the intervals 0 to 1. When we tried to evaluate this integral for various values of n using the following the recursion formula y n plus 5 y n minus 1 is equal to y 1 by n.

(Refer Slide Time: 05:28)



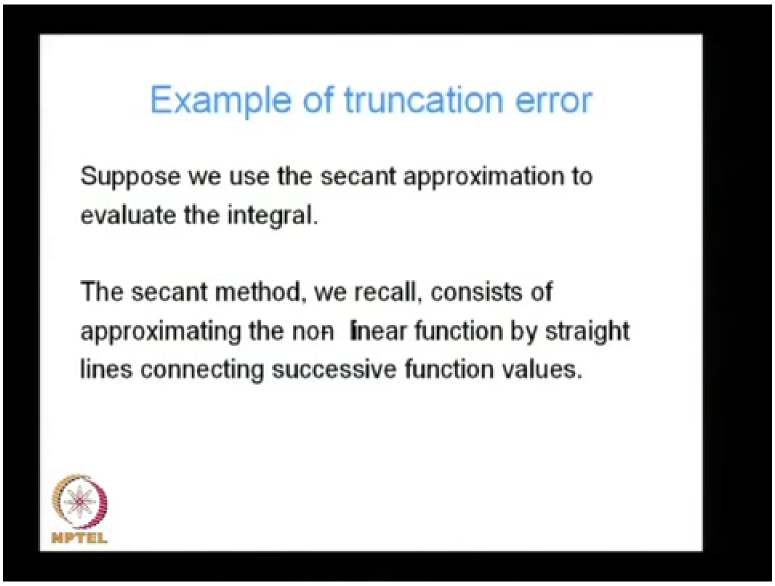
In our first lecture, we found that inherent in stability of this numerical algorithm let to accumulation of the round off errors and very quickly may be even after just five iterations we were getting results which had no relationship with the reality, that the results were totally wrong. We started with y 0 equal to 0.182 and the by the time we reach to y 4 we reached we got a value of 0.165, which was totally different from the true solution and this we found was because of, the inherent instability of the numerical algorithm. We used which allowed these round off errors to pile up and lead to be essentially meaningless results.

(Refer Slide Time: 06:17)



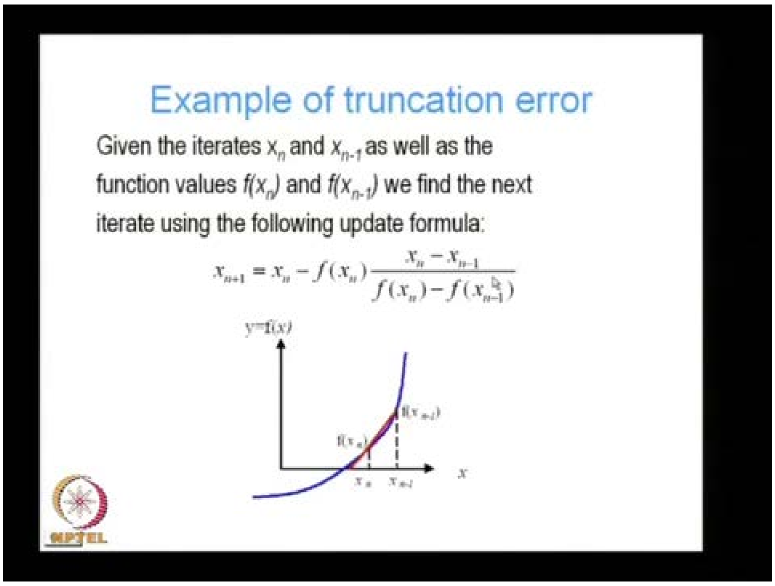
In that case round off errors let to meaningless results due to the instability of the algorithm. To see an example of truncation error, we consider numerical integration of the definite integral y of x integrated between the limits a and b.

(Refer Slide Time: 06:42)



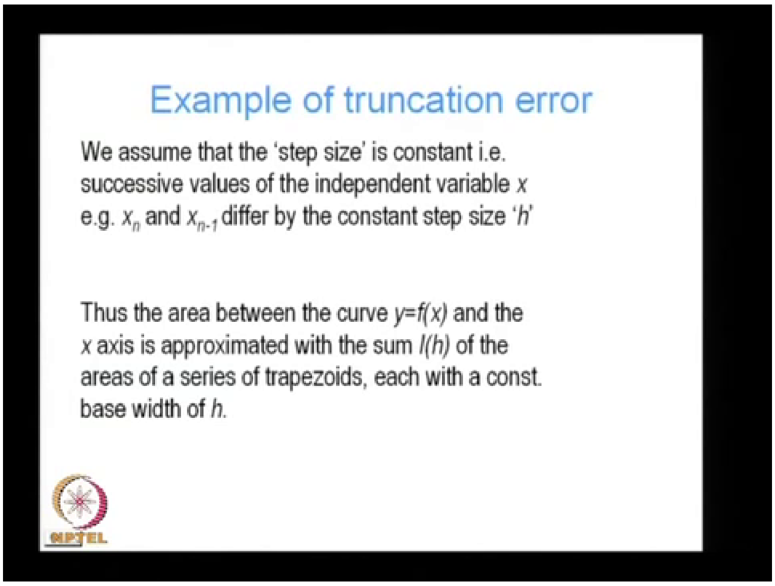
Suppose we use the secant approximation to evaluate this integral. We recall from our last lecture that the secant method, consists of approximating the non-linear function by straight lines connecting successive function values.

(Refer Slide Time: 07:01)



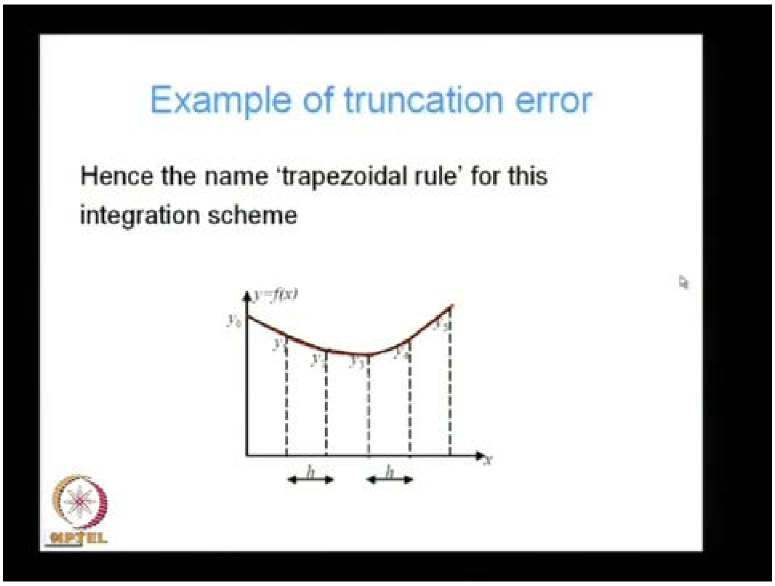
That is given the iterates x n and x n minus one as well as the function values f of x n and f of x n minus 1 we find the next iterate using the following update formula x n plus 1 equal to x n minus f of x n times x n minus x n minus 1 divided by f of x n minus f of x n minus 1. Graphically this means that this non-linear curve in blue is approximated by straight lines joining points on the curve, which represent each iterate an it is value and its function value.

(Refer Slide Time: 07:55)



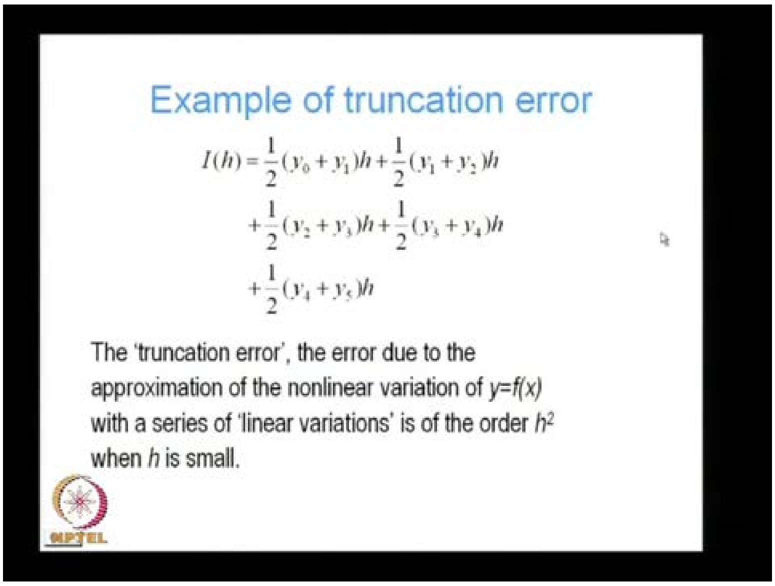
We further assume that in our iteration proceed a using the secant algorithm the steps size we use is constant that is successive values of the independent variable x for example, x n and x n minus 1 differ by a constant step size h. Thus the area between the curve y is equal to f of x and the x axis is approximated with the sum i of h of the areas of a series of trapezoids, each with the constant base width of h.

(Refer Slide Time: 08:39)



That is this area between the curve and the x axis is approximated by these trapezoids, each trapezoid representing an iteration in our secant update algorithm. So, each trapezoid has of course, constant base h since we are we are iterating with a constant step size and since, the total area under the curve is obtained by summing up the area of these trapezoids we use the name trapezoidal rule for this integration scheme.

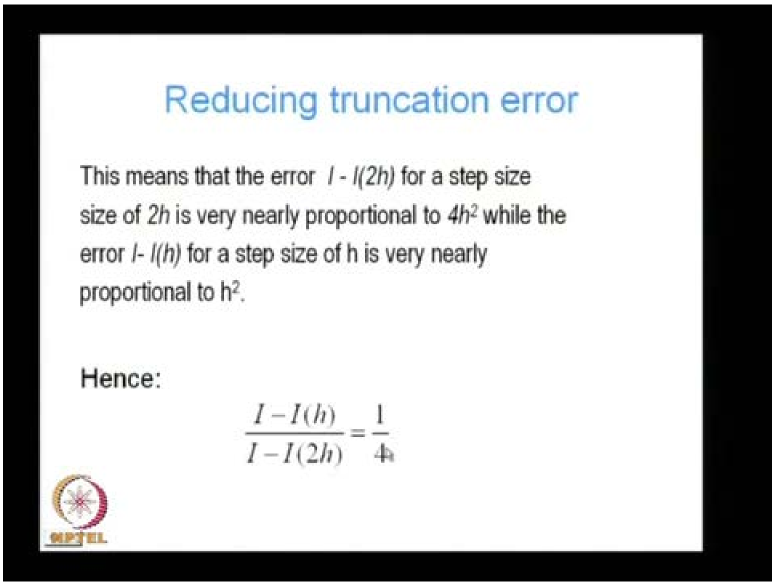
(Refer Slide Time: 09:19)



Let us see what the trapezoidal rule means, if we have five intervals for evaluating our function y is equal to f of x between these two limits. In that case, we can write i of h is equal to sum of the first trapezoid plus the sum of the second trapezoid plus third and the 4th and the fifth trapezoid and as you can see the area of the base in each trapezoid is the same the constants step size h while the sum of the parallel sides is given by y 0 plus y 1.

The truncation error, the error due to the approximation of the non-linear variation of y which is equal to f of x with a series of linear variation in our case these trapezoids is of the order of h square when h is small.

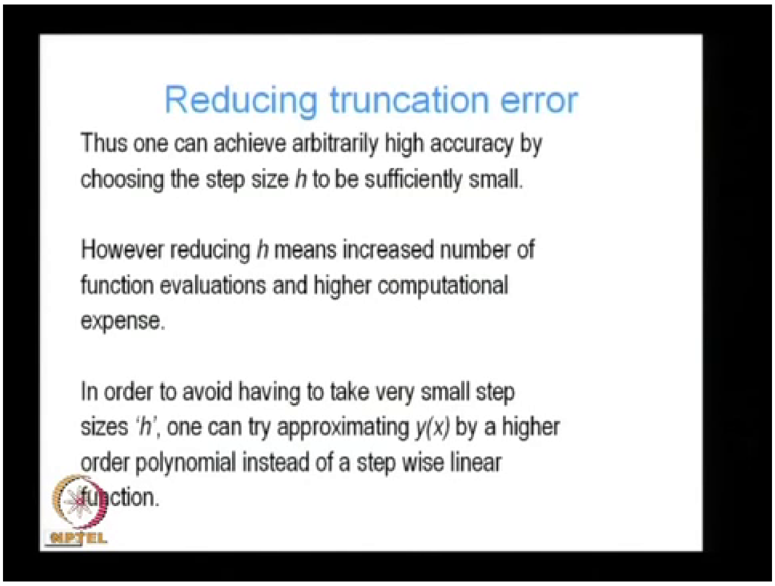
(Refer Slide Time 10:30)



Thus, what it means is that the error between i and i 2 h where i is the exact value of the integral and i within brackets 2 h is the result of our trapezoidal integration scheme where we have used the step s size of 2 h is going to very pro practically with respect to the step size.

So, it is going to be very nearly proportional to 4 h square while, if the step size is h the error between the exact value i and when numerically evaluated value using the trapezoidal rule with the constant step size of h i minus i h is very nearly proportional to h square. Hence, if we take a ratio of the errors i minus i of h divided by i minus i of 2 h we see that this ratio goes as h square by 4 h square h square h square cancels out and we get a ratio of 1 over 4.

(Refer Slide Time: 11:40)

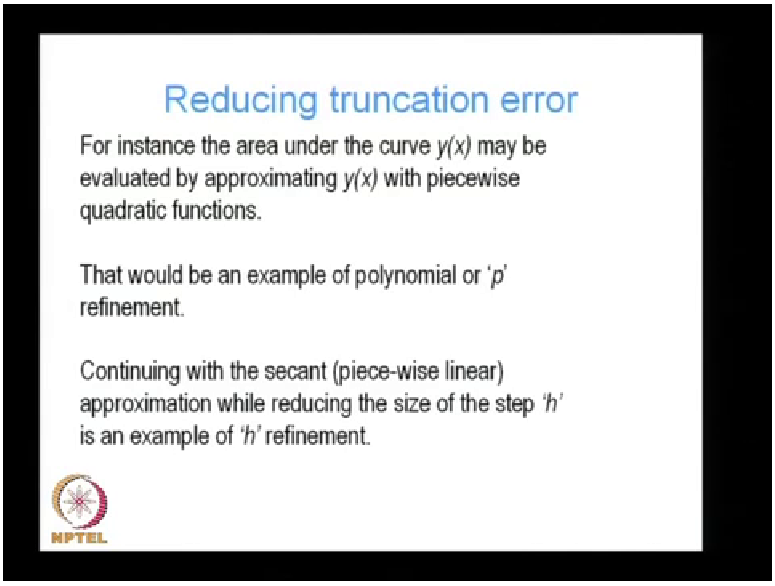


Thus we see that by reducing the step size by half we are getting an error which is just one 4th of the error with the step size with the previous step size. So, we reduce the error reduce the step size from h to 2 h an our error goes down by a factor of 4. We can we can see from this that we can achieve arbitrarily high accuracy by choosing the step size h to be sufficiently small, as we keep on reducing h are error is going to becomes smaller and smaller and the error is going to reduce quadratically that with step size h

However, reducing h means increased number of function evaluations and higher computational expense. This is because, if interval over which we had computing on the integral remains the same as we reduce the step size we have more function evaluation we have to evaluate the function at more points and this begins to greater computational expense.

In order to avoid having to take very small step sizes h, instead of using a linear approximation which we did here one can try approximating y of x by higher order polynomial for instance instead of approximating it y of x by a linear polynomial we can user higher order polynomial for instance a quadratic function.

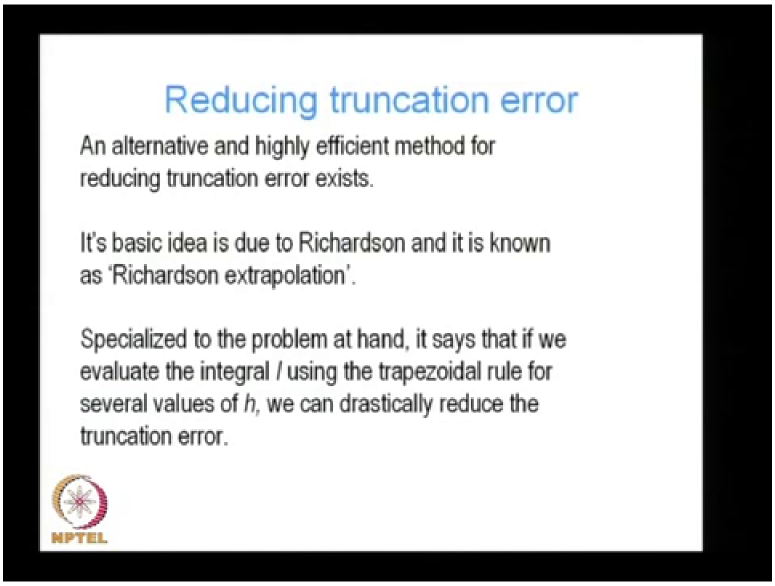
(Refer Slide Time: 13:14)



Quadratic polynomial for instance the area under the curve y of x may be evaluated by approximating y a y x with piece wise quadratic function. That would be an example of polynomial or p refinement, where we have approximated the function y of x by a higher order polynomial 9 in this case a quadratic polynomial similarly, we can use cubic quadratic polynomials to approximate the function y over f x over are intervals over are step sizes.

Alternatively, if we continue with our secant that is our piece wise linear approximation while reducing the size of the step h that would be an example of h refinement, these are very common ideas in numerical analysis polynomial refinement and step size refinement or mesh size refinements. So, these are the two main types of refinements which people use to get more accurate solutions.

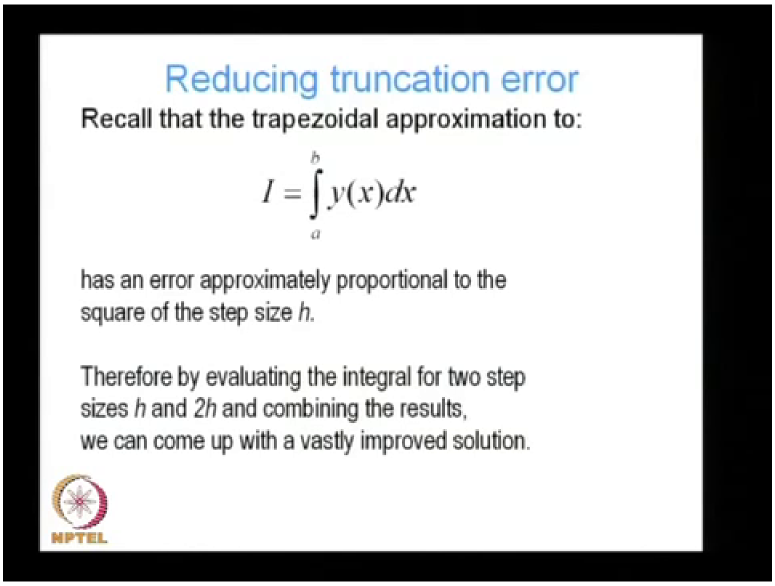
(Refer Slide Time: 14:30)



However instead of using either h or p refinement there is an alternative and highly efficient method for reducing the truncation error, its basic idea is due to richardson and it is known as richardson extrapolation. In our particular problem and where we want to evaluated integral within a certain bounds what it means is that if we use if we evaluate the integral using the trapezoidal rule if instead of reducing h reducing the step size h or increasing the order of our polynomial approximation.

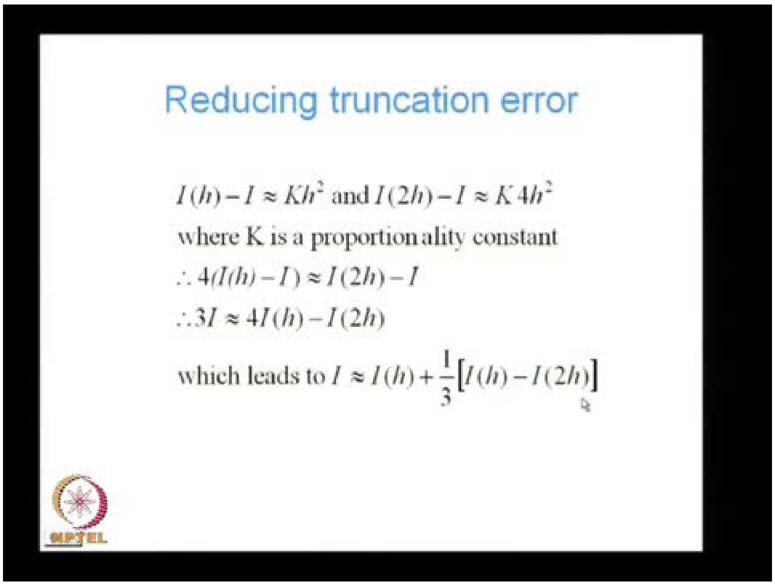
If we cut persist with our linear approximation that is we persist with the trapezoidal rule, but use the trapezoidal rule several times that is for several values of h and we combined the values of the integral obtained using the several values of h, we can drastically reduce the truncation error.

(Refer Slide Time: 15:33)



Let us recall that the trapezoidal approximation to i is equal to integral of y of x evaluated within the bounds a to b has an error approximately proportional to the square of the step size h. Therefore, while evaluating the integral for two step sizes h and 2 h and combining the results, we can come up with a vastly improved solution as we will see in the next slide.

(Refer Slide Time: 16:07)

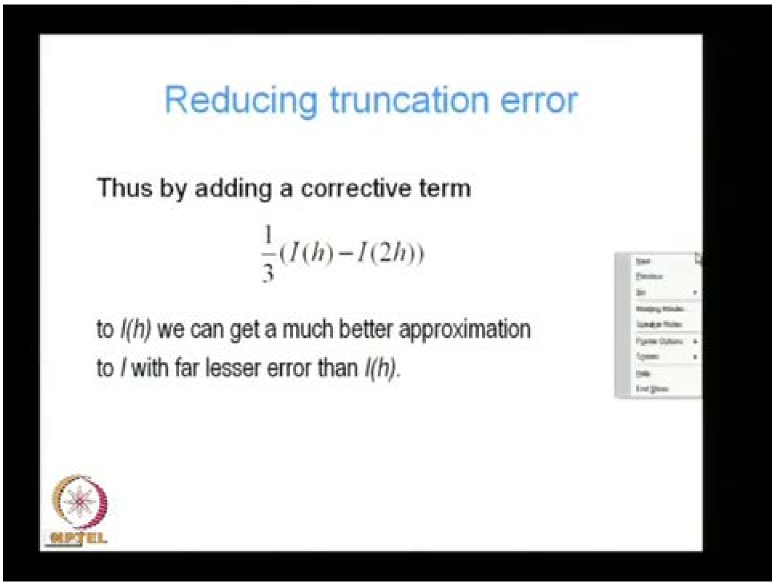


Since the error is of the order of h square we recall that i of h which is i this is the integral which is evaluated using the trapezoidal approximation with step size h, differs from the true solution by h square. So, the error is of can be writ10 as k times h square where k is a proportionality constant since the error is proportional to h square we can write that the error is approximately equal to the proportionality constant k times h square.

Similarly, if we reduce the step size if we use a step size of 2 h then in that case the difference between i of 2 h where i is the error with the step size of 2 h and i 2 h is the is the integral with the step size of 2 h and i is the exact value the difference between these 2 values is given by k times 4 of h square.

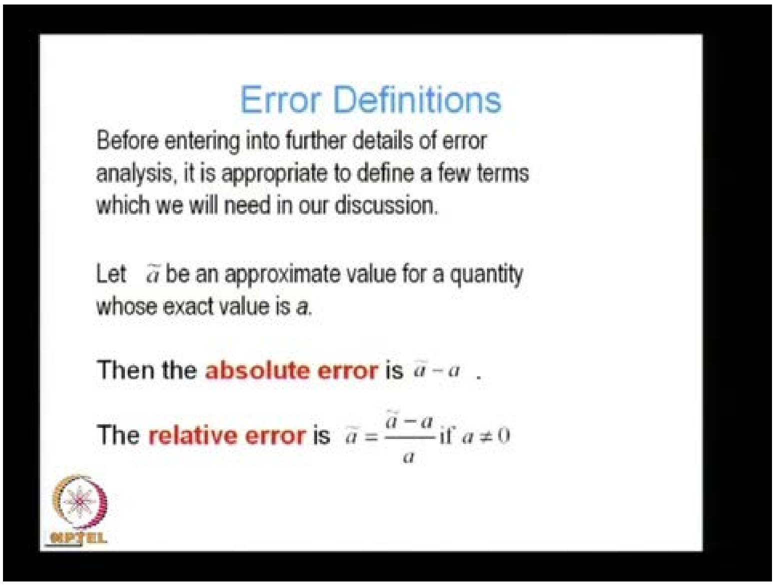
Subtracting these 2 we see that 4 times i of h minus i is approximately equal to i of 2 h minus i 4 times i of h minus i that is equal to 4 times k h square i of 2 h minus i is again equal to k times 4 h square, so these 2 are equal. So, this gives me an equation for the exact value of the integral which is i if we solve this equation for i we get three i is approximately equal to 4 times i of h, the integral evaluated with the step size of h minus i of 2 h that is the integral evaluated with the step size 2 h. This leads me to the expression that the exact value of the integral is approximately equal to i of h plus one third of i of h minus i of 2 h.

(Refer Slide Time: 18:16)



Thus by adding the corrective term one third i of h minus i of 2 h 2 i of h we get a much better approximation to i with far lesser error than i of h that is, by adding this additional term which is one third times i of h minus i of 2 h 2 i of h we get an error we get a solution we get a value of the integral which is much closer to the true solution than i of h itself.

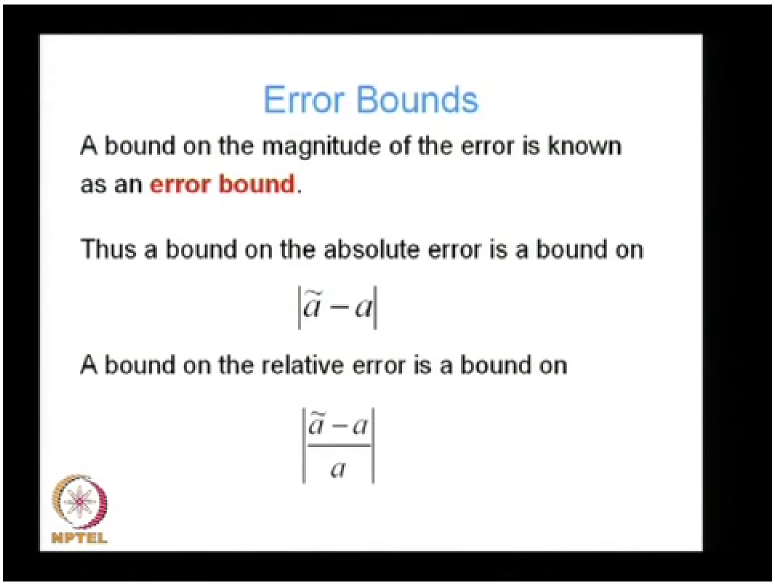
(Refer Slide Time: 18:55)



Before entering into further details of error analysis, we would like to define a few terms which will need in our discussion. First terms that we want to define we let we let two types of errors if we denote a tilde to be an approximate value for a, quantity whose exact value is a then we define the absolute error in a is given by the approximate value minus to the true value that is the absolute error is a tilde minus a, the relative error on the other hand is given by a tilde is equal to a tilde minus a divided by a.

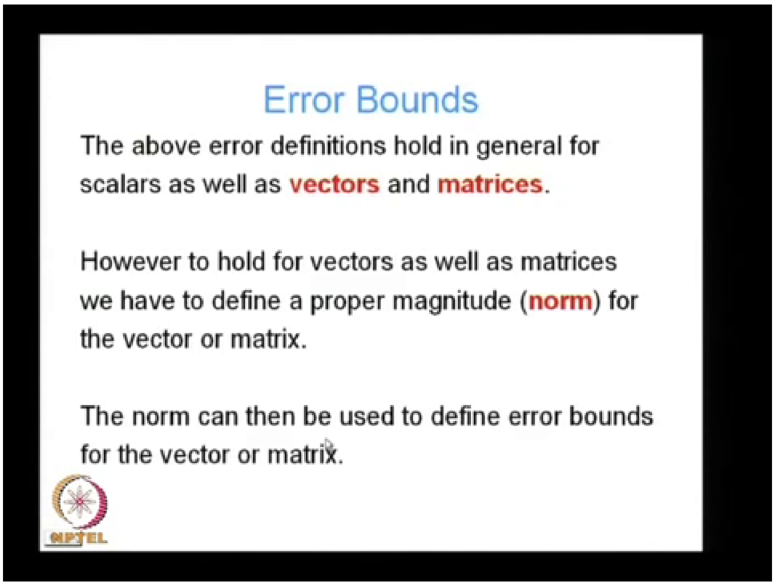
This of course, assumes that a is not equal to 0 if a is equal to 0 we cannot define the relative error. So, the absolute error is just the numerical solution a tilde minus the true solution a where the relative error is the numerical solution a tilde minus the true solution a divided by the true solution a assuming of course, that the true solution is not equal to 0.

(Refer Slide Time: 20:21)



A bound on the magnitude of the error is known as an error bound. Thus a bound on the absolute error is a bound on a tilde minus a, it is a bound on the absolute value of a tilde minus a, a bound on the relative error on the other hand is a bound on the absolute value of a tilde minus a divided by a.

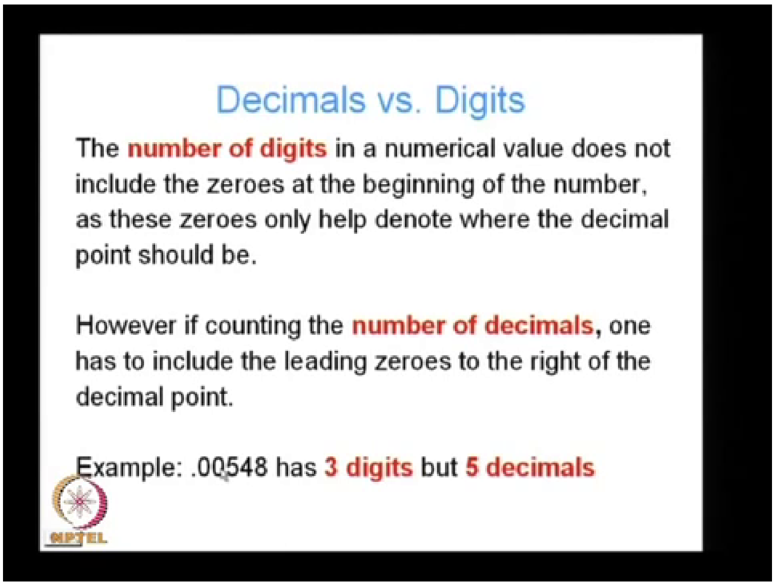
(Refer Slide Time: 20:52)



The above error definitions hold in general for scalars as well as vectors and matrices. In our definition of absolute error and relative error we looked at scalars a is assume to be a scalar, but we now, saying that a can as well b a vector or a matrix provided we can calculate a value for the magnitude of the vector or the matrix or we can calculate a value of the norm of the vector or the matrix.

Once we can calculate the value of the magnitude of the norm of the vector or matrix we can use that norm to define error bounds for the vector or the matrix.

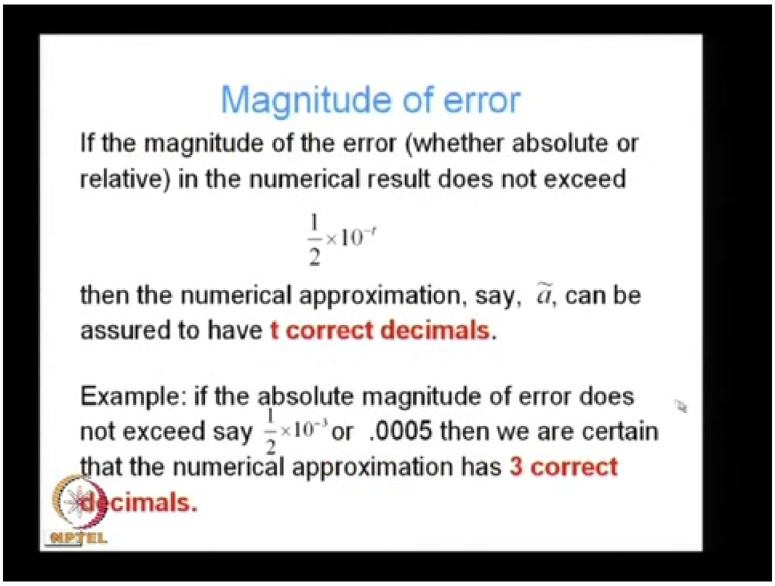
(Refer Slide Time: 21:46)



That was about errors about relative errors an absolute errors, next we how do we measure errors how do we decide how much is the error to decide that, we have to talk about the measures of accuracy in order to talk about to measure is of accuracy we want to talk about decimals the errors interns of decimals an errors interns of digits, the number of digits in a numerical value does not include the zeroes at the beginning of the number as these zeroes only help denote where the decimal point should be.

However, if we are counting the number of decimals one has to include the leading 0es to the right of the decimal point. These becomes clear if we considered an examples for instance the number 0.00548 has only three digits because, we ignore the two zeroes which occur at the beginning of the number however, it has got five decimals since the number of decimals include the leading zeroes to the right of the decimal point.

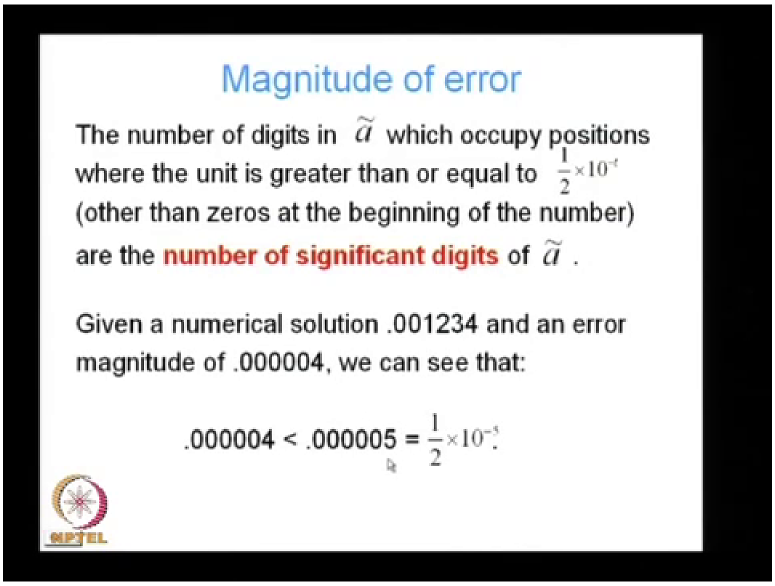
(Refer Slide Time: 23:00)



If the magnitude of the error whether it be the absolute error or relative error in the numerical result does not exceed half times 10 to the power minus t, then we say by definition that the numerical approximation, a tilde has t correct decimals. That is if the error is lesser than 0.5 times 10 to the power minus t the error is set to have t correct decimals.

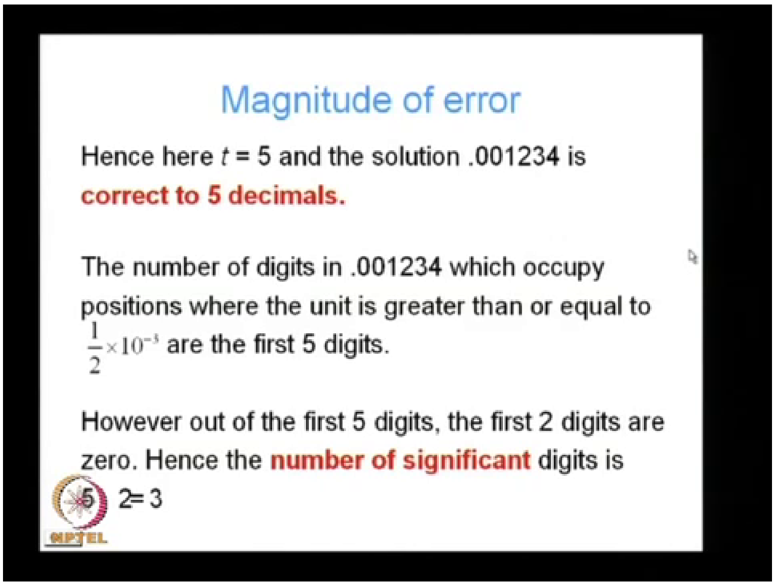
Again we considered an example if the absolute magnitude of error does not exceeds a half into 10 to the power minus three or 0.0005 then we are certain that the numerical approximation has three correct decimals this is evident because, the error is only appearing in the 4th decimal place and the a magnitude of the error is less than 0.0005. So, the first three decimals in a tilde in the numerical approximation must be correct, so the number has three correct decimals a numerical approximation has three correct decimals.

(Refer Slide Time: 24:28)



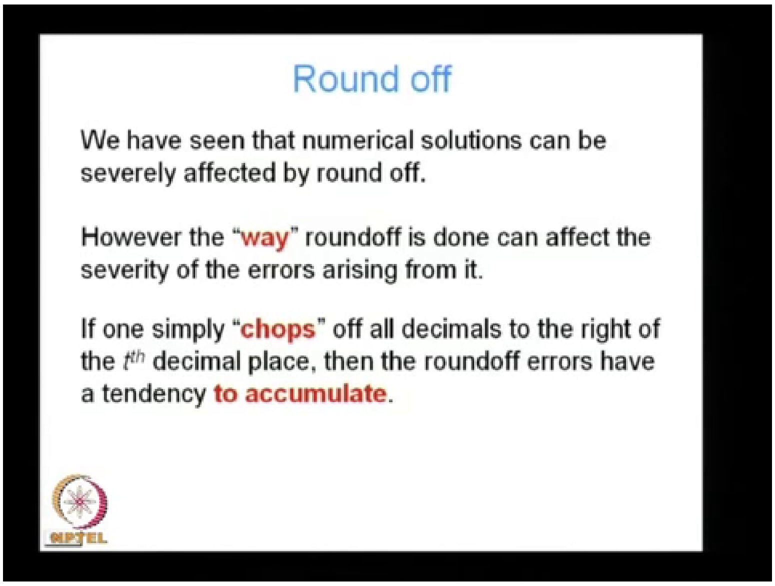
The number of digits in a tilde which occupy positions where the unit is greater than or equal to, half into 10 to the power minus t of course, we ignore the zeroes at the beginning of the number other number of significant digits of a tilde. Suppose, we have a numerical solution 0.001234 and we know that, the numerical solution has an error of magnitude 0.000004 we can see that 0.000004 is of course, less than 0.000005 which is equal to half times 10 to the power minus 5, thus we can see there are error is in this 6 decimal place because the error is in the 6 decimal place.

(Refer Slide Time: 25:33)



So, we can see that the we have five correct decimals, that is here in this solution t is equal to five and the solution 0.001234 is correct to 5 decimals the number of digits in 0.001234, which occupy positions where the unit is greater than or equal to, half in to 10 to the power minus three are the first five digits this should actually be half in to 10 to the power minus five I apologized for the type, but the number of digits 0.001234 which occupy positions where the unit is greater than or equal to half in to 10 to power minus 5 are the first five digits. However, out of the first five digits the first 2 digits are 0 the first 2 digits are 0 hence the number of significant digits is five minus 2 is equal to three. So, we have three significant digits in our numerical solution

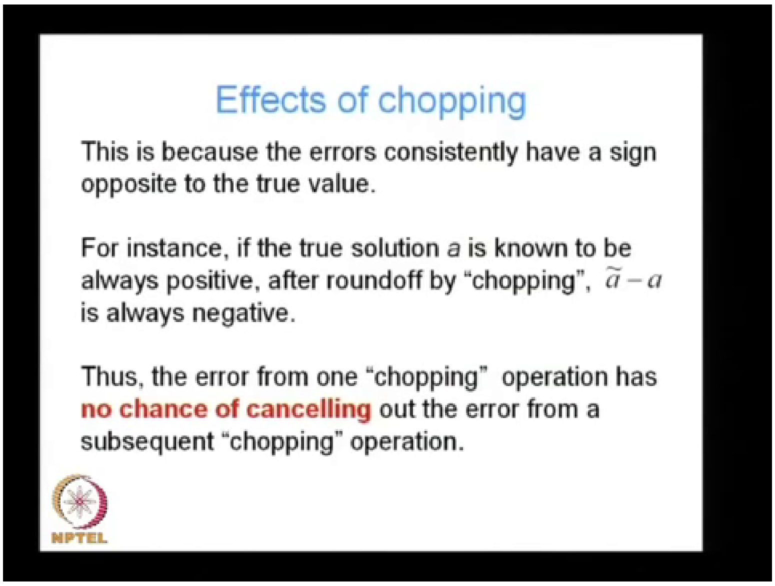
(Refer Slide Time: 26:53)



We have seen previously that numerical solutions are severely affected by round off. However, the way we do the round off is critical an f can affect the severity of the errors arising from round off if, one simply chops of all the decimals to the right of the t th decimal place then the round off errors have a 10dency to accumulate, suppose our solution our number has s digits.

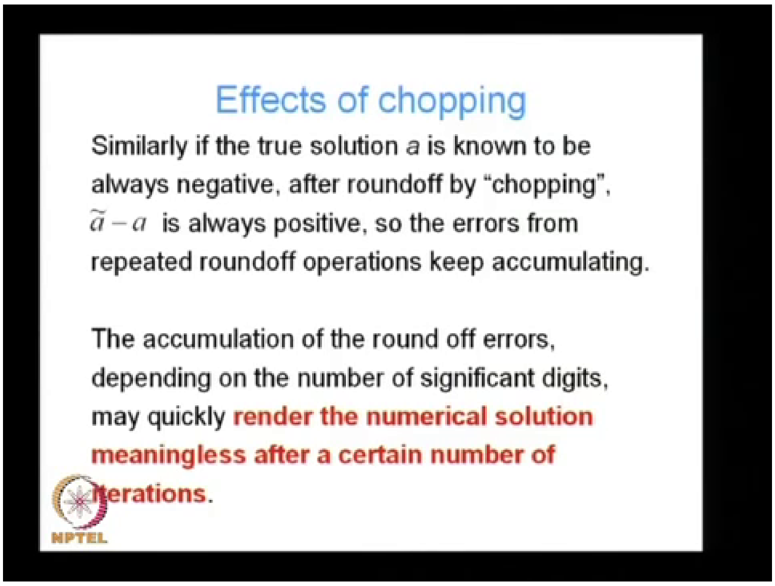
but if we suppose if we chop it of after the t th decimal place then we all the numbers which followed the t th decimal place get removed from a numerical solution, but if we do this chopping arbitrarily if we do this round off arbitrarily then we will see that these round off errors have a 10dency to accumulate.

(Refer Slide Time: 28:00)



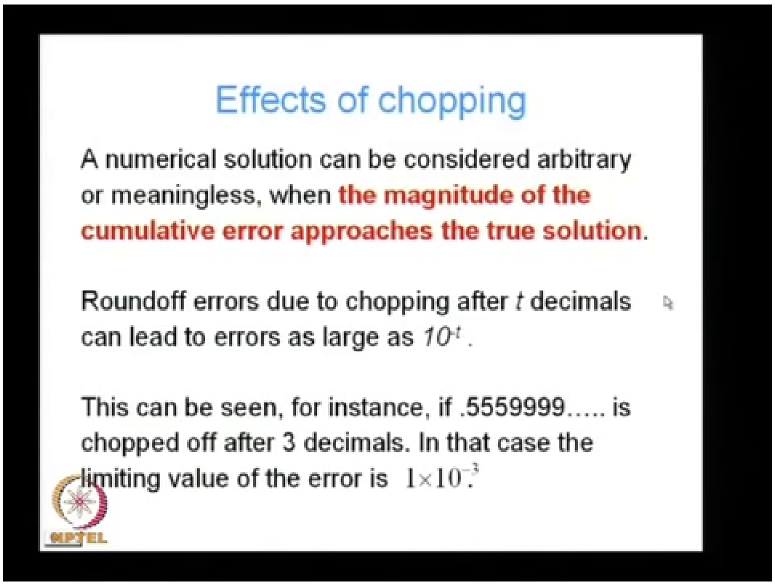
This is because the errors consis10tly have a sign opposite to the true value, for instance if the true solution a is known to be always positive, after round off by chopping a tilde minus a is always going to be negative. Because after chopping a tilde has to be less than a because we have got rid of the additional digits which follow the t th digits, so a tilde has got to be less than a and a tilde minus a will always be negative, thus the error from chop one chopping operation has no chance of canceling out the error from a subsequent chopping operation.

(Refer Slide Time: 28:46)



Similarly, if the true solution a is known to be always negative, after round off by chopping a tilde minus a is always going to be positive because, a tilde is going to be less negative than a, so a tilde minus a is going to be always positive, so the errors from repeated round off operations will keep accumulating. The accumulation of the round off errors depending on the number of significant digits may quickly render the numerical solution meaningless after a certain number of iterations.

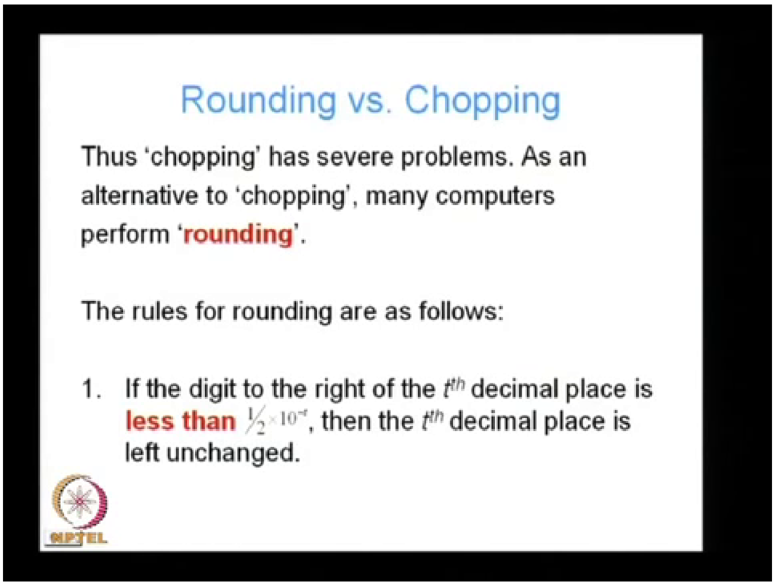
(Refer Slide Time: 29:35)



A numerical solution can be considered arbitrary or meaningless, when the magnitude of the cumulative error approaches the true solution, round off errors due to chopping after t decimals can lead to errors as large as 10 to the power minus t. This can be seen for instance if we consider the number 0.5559999 up to infinity, then if we chop this number after three decimals that is we store only 0.0555 in our computer.

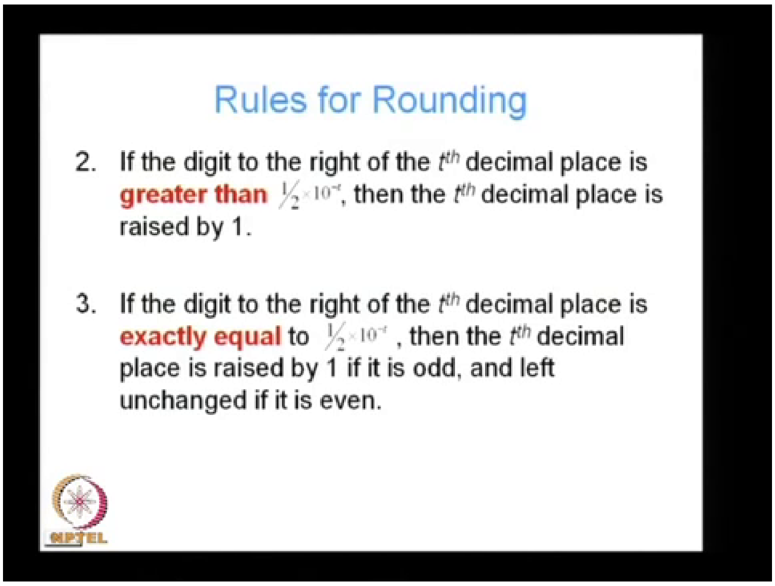
In that case the error has error is of magnitude 0.00099999 up to infinity and the limiting value of the error, we can see is one in to 10 to the power minus 3, that is round off errors to due to chopping after three decimals can lead to errors as large as 10 to the power minus 3.

(Refer Slide Time: 30:40)



Thus chopping is seen to have severe problems as an alternative to chopping many computers perform what is known as rounding. The rules for rounding are as follows there are three rules, the first rules says if the digit to the right of the t th decimal place is less than half in to 10 to the power minus t then the t th decimal place is left unchanged, for instance if we are considering t equal to three if the digit to the right of the third decimal place is less than 0.0005 that is half in-to 10 to the power minus three which is 0.0005 then the third decimal place is going to be left unchanged.

(Refer Slide Time: 31:36)

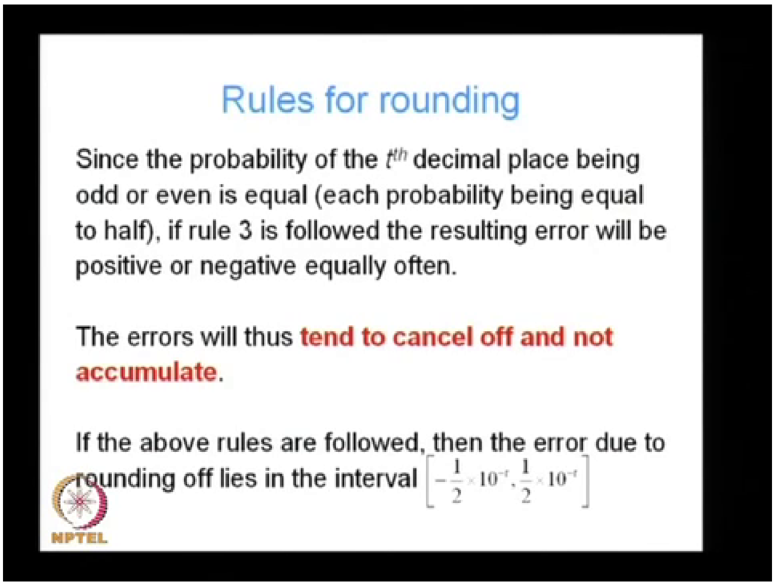


If the digit to the right of the t th decimal place on the other hand is greater than half in to 10 to the power minus t, then the t th decimal place is raised by one, for instance in our example if our number is 0.3254 then the first rule would apply the first rule would tell me I have to approximate 0.3 to 54 as 0.3 to 5. Because, the digit to the right of the t th decimal place is less than 5 in to 10 the power minus 4 however, if my number is for instance 0.3256, then we are going to we are going to store 0.3 to 56 as 0.3 to 0.6 because, the number to the right of the t th.

In this case the third decimal place is 0.0006 which is greater than 0.0005 which is half in to 10 to the power minus 3 then because of, that we have raised that t th decimal place we have change 0.325 to 0.326, if the digit to the right of the t th decimal place is exactly equal to half in to 10 to the power minus t then the t th decimal place is raised by one if it is odd and left unchanged if it is even. So, these are our three rules for rounding.

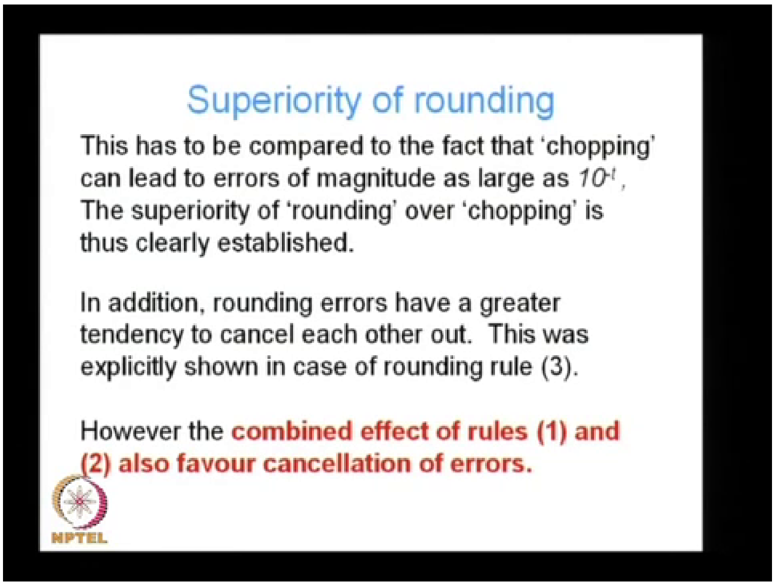
We can we will see that these rules result in lower error magnitudes for instance, let us consider rule three the rule three says that if the let us go back and look at rule three again which says that if the digit to the right of the t th decimal place is exactly equal to half in to the 10 to the power minus t. Then the t th decimal place is raised by one if it is odd and left unchanged if it is even.

(Refer Slide Time: 33:45)



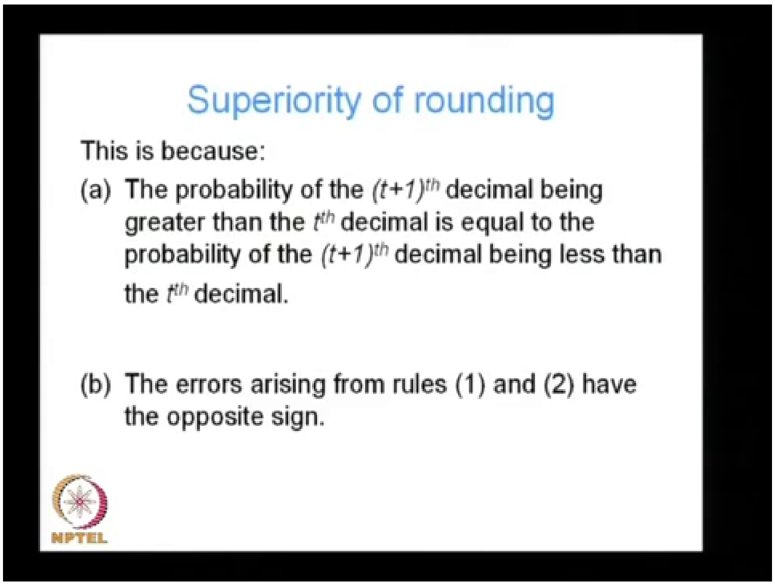
Since the probability of the t th decimal place being odd or even is equal each probability being equal to half, if rule three is followed the resulting error will be positive or negative equally of10 because, the possibility of the t th decimal place being odd or even is equal. So, the resulting error we get is also going to be positive or negative equally of10 the errors will thus 10d to cancel off and not accumulate, if the above rules are followed then the error due to rounding off lies in the interval minus half in to 10 to the power minus t to half in to 10 to the power minus t.

(Refer Slide Time: 34:39)



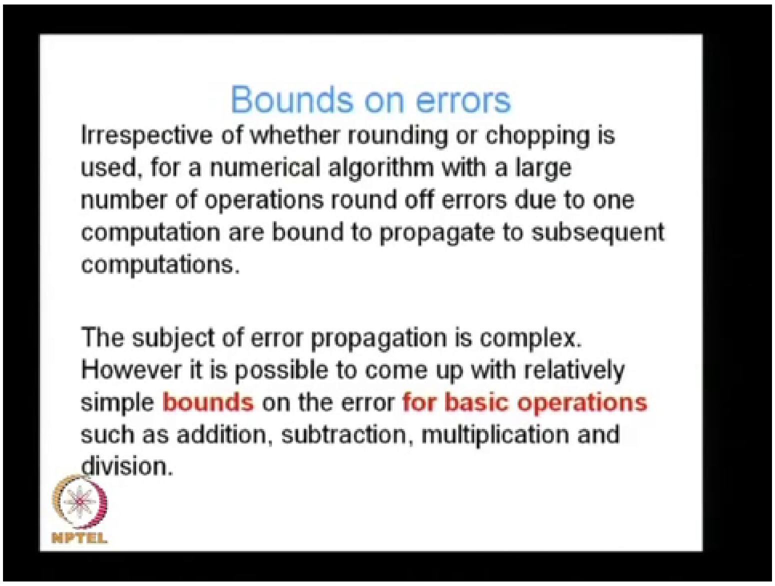
This is to be compared with the fact that if we do chopping instead of using around of algorithm which we just described we can get errors as large as 10 to power minus t, instead of 10 to the power minus t. Now, are errors lie in the range minus half 10 to the power minus t to half 10 to the power minus t, the superiority of rounding over chopping is thus clearly established in addition rounding errors have a greater tendency to cancel each other out. We should this explicitly in case of rounding rule three however, the combined effect of rules one and 2 also favor the cancellation of errors.

(Refer Slide Time: 35:29)



This is because, the probability of the t plus 1 th decimal being greater than the t th decimal is equal to the probability of the t plus 1 th decimal being less than the t th decimal. The errors arising from rules one and 2 therefore, have opposite signs and therefore, have a tendency to cancel each other out.

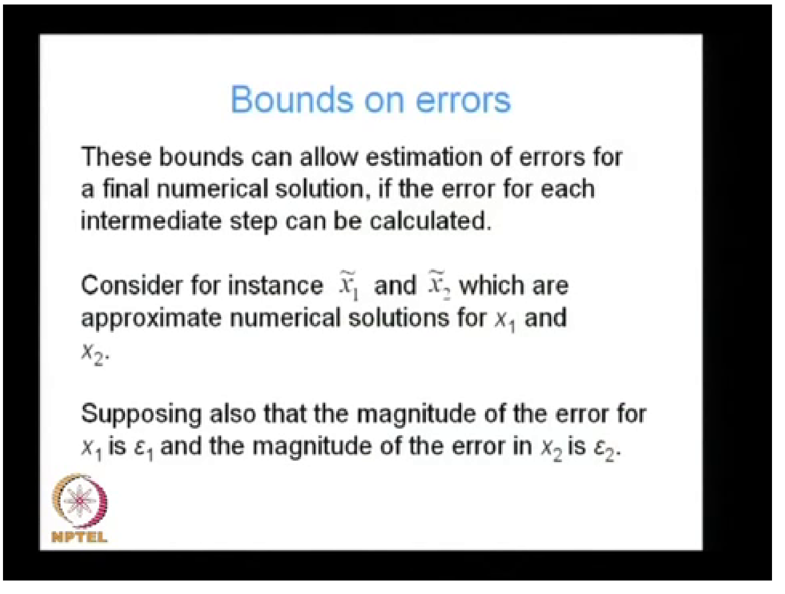
(Refer Slide Time: 36:00)



However, irrespective of whether rounding or chopping is used, for a numerical algorithm with a large number of operations round off errors due to one computation are bound to propagate to subsequent computations. The errors in one computation are going to affect the results of the subsequent computation this is known as propagation of errors or error propagation the subject of error propagation is complex

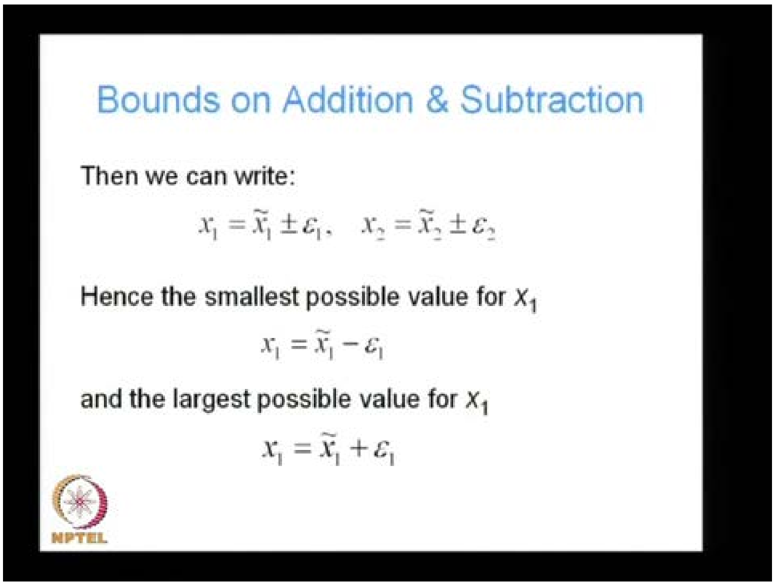
However, it is possible to come up with relatively simple bounds on the error for basic operations such as addition subtraction multiplication and division. The idea being that if we can come up with simple bounds for the basic operations and since all numerical methods, are basically a combination of the simple operations of addition subtraction multiplication and division. if we know the errors due to each of these individual operations we can come up with an error for the entire numerical algorithm.

(Refer Slide Time: 37:35)



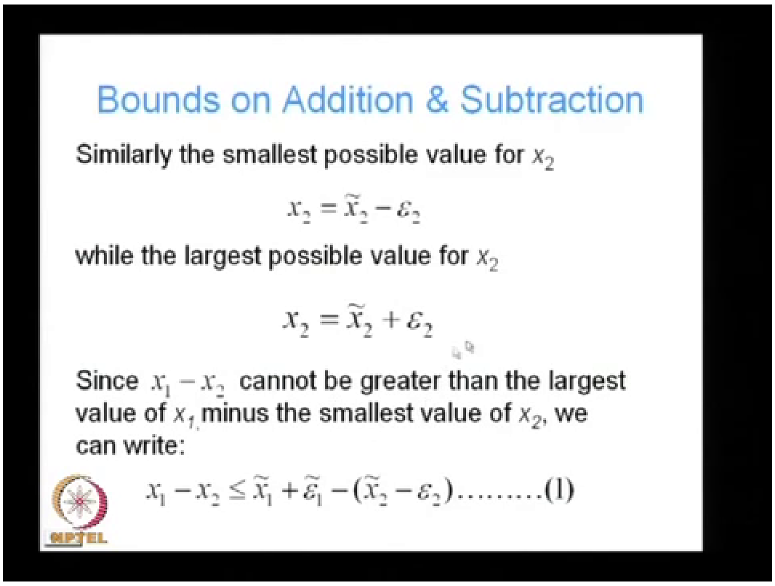
So, if we can again going back if we can bound these errors for the simple operations we can establish bounds for our final numerical solution, in case we compute the error for each intermediate step if we compute the error for each intermediate step and we keep adding we keep track of those errors. If we do some book keeping and if keep track of the those errors then we can get the final error in our numerical solution consider for instance x1 tilde and x 2 tilde which are approximate numerical solutions for x 1 and x 2. Supposing also that the magnitude of the error for x 1 is epsilon 1 and the magnitude of the error in x 2 is epsilon 2. This we know that the magnitude of the error in x 1 in is in epsilon 1 and the magnitude of the error in x 2 is epsilon 2 now, suppose knowing those two values knowing the magnitudes of the error in x 1 and x 2.

(Refer Slide Time: 38:51)



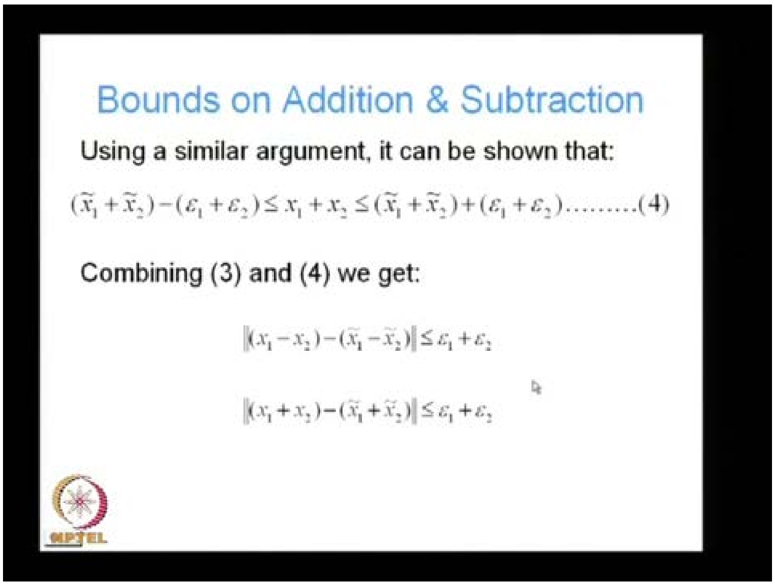
We can write x 1 is equal to x 1 tilde plus minus epsilon 1 and x 2 is equal to, x 2 tilde plus minus epsilon 2 that is x 1 can be can be as low as x 1 tilde minus epsilon 1 and can be as high as x 1 tilde plus epsilon 1 similarly, x 2 can be as low as x 2 tilde minus epsilon 2 and can be as high as x 2 tilde plus epsilon 2. Hence the smallest possible value for x 1 is equal to x 1 tilde minus epsilon 1 and the largest possible value for x 1 is equal to x 1 tilde plus epsilon 1.

(Refer Slide Time: 39:58)



Similarly, the smallest possible value for x 2 is also x 2 tilde minus epsilon 2, where epsilon 2 recall is the error in x 2 and the largest possible value for x 2 is x 2 tilde plus epsilon 2. Since x 1 minus x 2 cannot be greater than the largest value of x 1 minus the smallest value of x 2 the magnitude of x 1 minus x 2 cannot exceed the largest value of x 1 minus the smallest value of x 2. We can write x 1 minus x 2 is bounded on the right that is bounded has an upper bound x 1 tilde plus epsilon tilde minus x 2 tilde minus epsilon 2 x sorry this should not be epsilon one tilde it is actually epsilon one, so x 1 tilde plus epsilon one minus x 2 tilde minus epsilon 2.

(Refer Slide Time: 41:17)

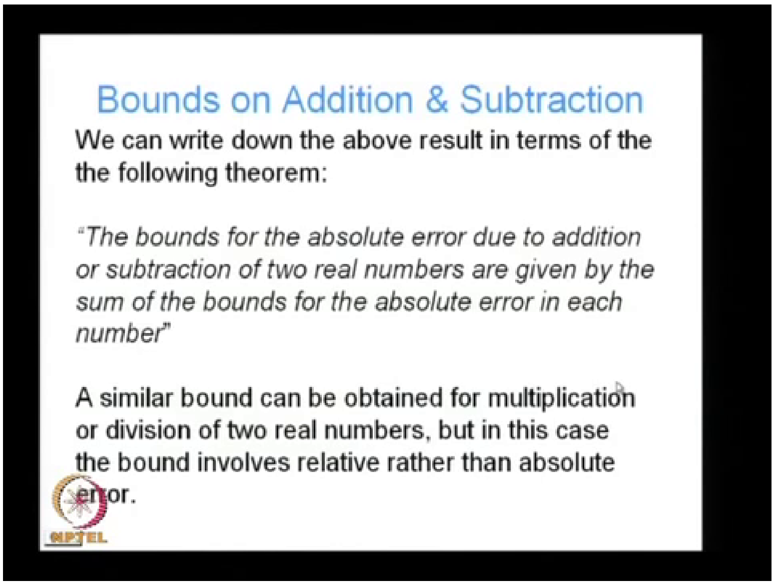


Similarly, since x 1 minus x 2 cannot be less than the smallest value of x 1 minus the largest values of x 2, we can write x 1 tilde minus epsilon 1 minus x 2 tilde plus epsilon 2 must be lesser than or equal to x one 1 x 2 or x 1 minus x 2 is bounded from below bounded. On the left by x 1 tilde minus epsilon 1 minus x 2 tilde plus epsilon 2 because, x 1 tilde minus epsilon 1 is the smallest value of x one smallest possible value of x 1 and x 2 tilde plus epsilon 2 is the largest possible value of x 2.

Combining equations 1 and 2 combining this equation with this equation we can get a, bound or x 1 minus x 2 which gives me a lower bound as well as an upper bound which tells me x 1 minus x 2 is bounded from below by x 1 tilde minus x 2 tilde minus epsilon 1 plus epsilon 2, while it is bounded from above by x 1 tilde minus x 2 tilde plus epsilon one plus epsilon 2 using a similar argument now, instead of considering subtraction, which we considered previously instead of considering x 1 minus x 2, if we try to get bounds for addition that is we try to get bounds on x 1 plus x 2. By using a very similarly, argument we can show that x 1 plus x 2 is also bounded from above and below by this by x 1 tilde plus x 2 tilde minus epsilon 1 plus epsilon 2 from below and x 1 tilde plus x 2 tilde plus epsilon 1 plus epsilon 2 from above.

Combining bounds 3 and 4 we get norm of that is absolute value of x 1 minus x 2 minus x 1 tilde minus x 2 tilde, must always be less than epsilon 1 plus epsilon 2 and x 1 the bound on x 1 plus x 2 minus x 1 tilde plus x 2 tilde is always lesser than or equal to epsilon 1 plus epsilon 2, thus the error due to subtraction is bounded by, epsilon plus epsilon 1 plus epsilon 2. Similarly the error due to addition is also bounded by epsilon one plus epsilon 2.

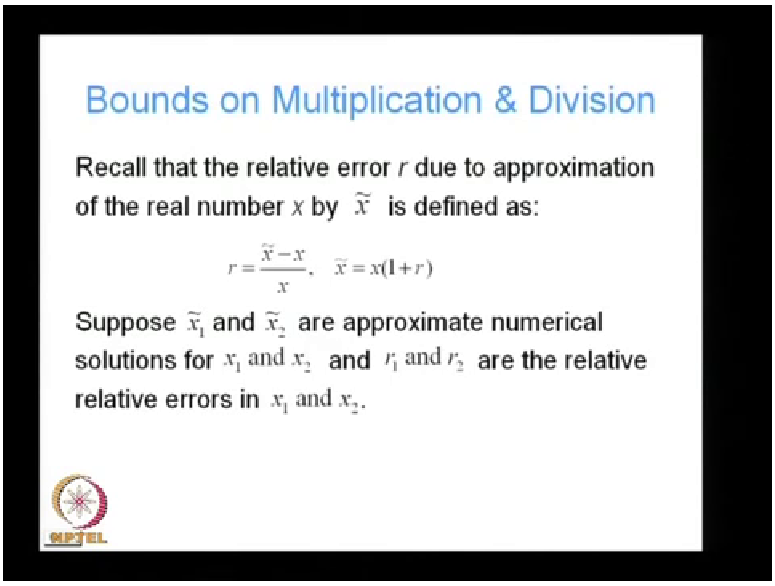
(Refer Slide Time: 42:48)



We can write down the above result in terms of the following theorem, which states that the bounds for the absolute error due to addition or subtraction of 2 real numbers in our case x one and x 2 are given by the sum of the bounds for the absolute error in each number, the sum of the bounds on the on the absolute error in each number which is epsilon 1 which is the absolute error in x 1 and epsilon 2 which is the absolute error in x 2.

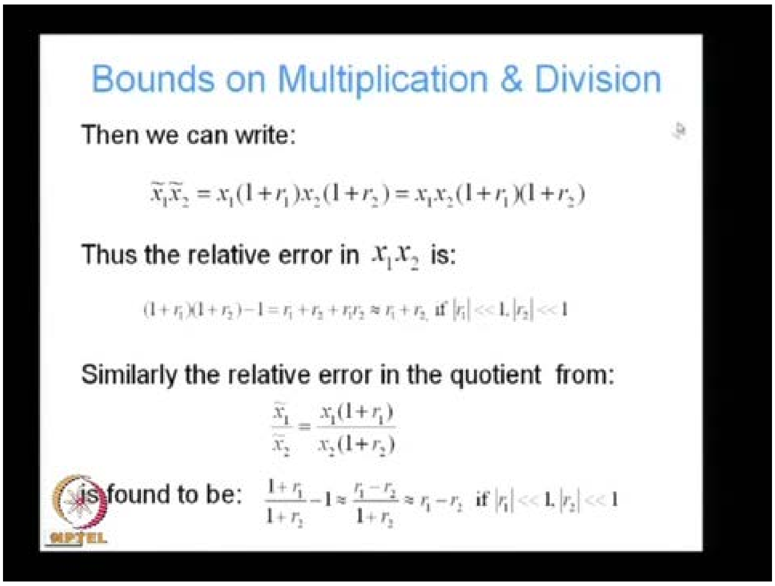
A similar bound can be obtained for multiplication or division of 2 real numbers, but in this case the bound involves relative rather than absolute error. Till now, we have been considering absolute errors in evaluating our bounds, but in order to obtain bounds for multiplication or division of two real numbers we have to consider relative rather than absolute error.

(Refer Slide Time: 45:32)



Again recall form our definition of relative error r due to the approximation of a real number x by x tilde we can define, the relative error as r is equal to x tilde minus x divided by x or x tilde is equal to x times of one plus r. Suppose x 1 tilde and x 2 tilde are approximate numerical solutions for x 1 and x 2 and r 1 and r 2 are the relative errors in x one and x 2. In that case we can write the product x 1 tilde times x 2 tilde as x one times 1 plus r one recall x tilde is equal to x 1 plus r. So, x 1 tilde is equal to x 1 times one plus r 1 times x 2 tilde which is x 2 times one plus r 2 which gathering terms.

(Refer Slide Time: 46:16)

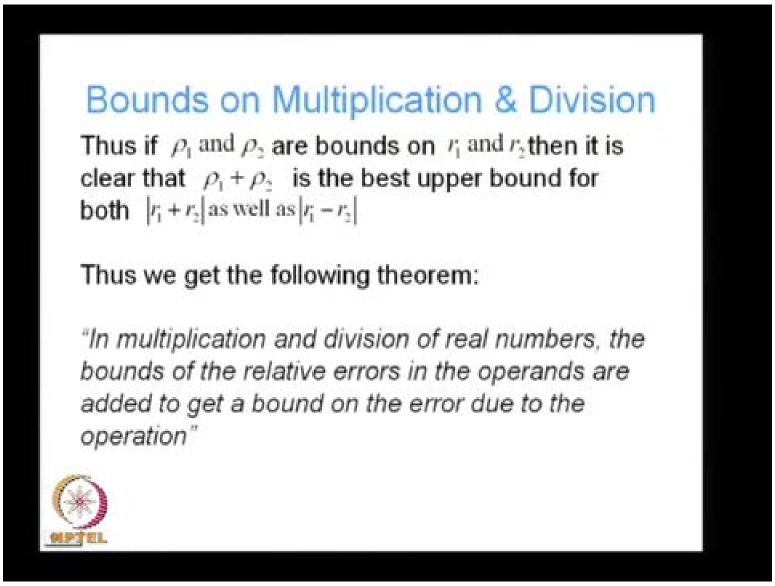


We can write, x 1 times x 2 times 1 plus r 1 into one plus r 2, thus the relative error in x 1 x 2 is given by again from our definition the relative error will be given by x 1 tilde times x 2 tilde minus x one x 2 divided by x one x 2. So, if we take x 1 tilde x 2 tilde we subtract x 1 x 2 from it we get x 1 x 2 times 1 plus r 1 times 1 plus r 2 minus x 1 x 2 dividing the whole thing by x 1 x 2, we get 1 plus r 1 times one plus r 2 minus 1 which is equal to r 1 plus r 2 plus r 1 r 2.

Suppose r relative errors in x 1 and x 2 are the much smaller than one in that case we can write r 1 plus r 2 plus r 1 r 2 to be approximately equal to r 1 plus r 2. Similarly, the relative error in the quotient can be evaluated for instance, x 1 tilde by x 2 tilde which is the quotient of x 1 and x 2 in terms of it is numerical solutions x 1 tilde and x 2 tilde can be written as x 1 times 1 plus r 1 divided by x 2 times 1 plus r 2, the relative error in the quotient will then be equal to x 1 tilde divided by x 2 tilde minus x 1 by x 2 divided by x 1 by x 2.

Which we if perform that operation we get 1 plus r 1 divided by 1 plus r 2 minus 1 which is approximately equal to r 1 minus r 2 divided by 1 plus r 2, which is not approximately which is actually exactly equal to r 1 minus r 2 divided by 1 plus r 2 and which is approximately equal to r 1 minus r 2 if r 2 is much less than one.

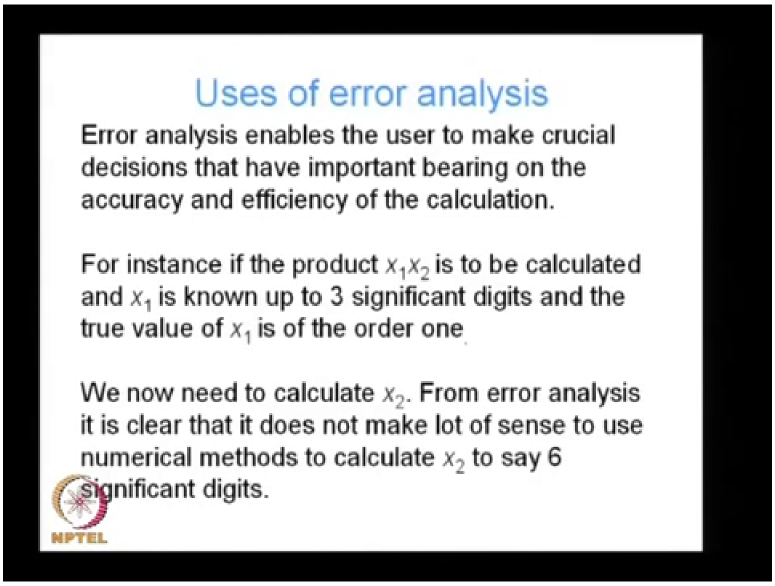
(Refer Slide Time: 49:10)



Thus if rho1 and rho 2 are bounds on the relative errors r 1 and r 2 then it is clear that rho 1 plus rho 2 is the best upper bound for both r 1 plus r 2 as well as r 1 minus r 2. If we want to we want to find bounds on r 1 plus r 2 which is the relative error due to multiplication and r 1 minus r 2 which is the relative error due to division we know that the relative error due to r 1 plus r 2 is bounded by the sum of the errors in r 1 and r 2 rho 1 and rho 2.

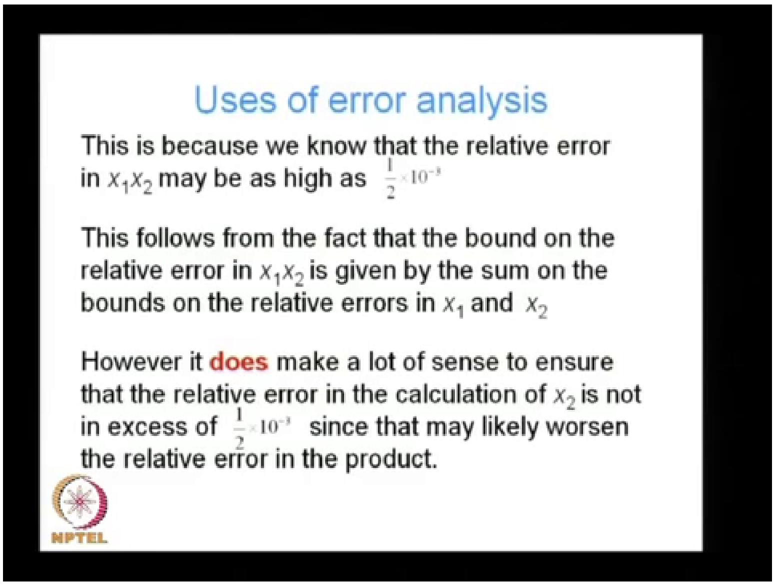
Thus if rho 1 and rho 2 are bounds on r 1 and r 2 then rho 1 plus rho 2 is the best upper bound for both r 1 plus r 2 as well as r one minus r 2. From this we get the following theorem, which states that in multiplication and division of real numbers the bounds of the relative errors in the operations are added to get a bound on the error due to the operation, basically which tells me that the bound of the error due to multiplication division is bounded by the sum of the relative errors in x 1 and x 2.

(Refer Slide Time: 50:50)



Error analysis enables the user to make crucial decisions that have important bearing on the accuracy and efficiency of the calculation. For instance if the product x 1 x 2 is to be calculated and x 1 is known up to three significant digits and the true value of x 1 is of the order 1 and we have to decide to what degree of accuracy we need to calculate x 2 we already know the value of x one and we know that it is accurate up to three significant digits and then we have to calculate the value of x 2 to in order to minimize the error in the product x 1 x 2. We from error analysis it is clear to us that it does not make a lot of sense to use numerical methods to calculate x 2 to say six significant digits.

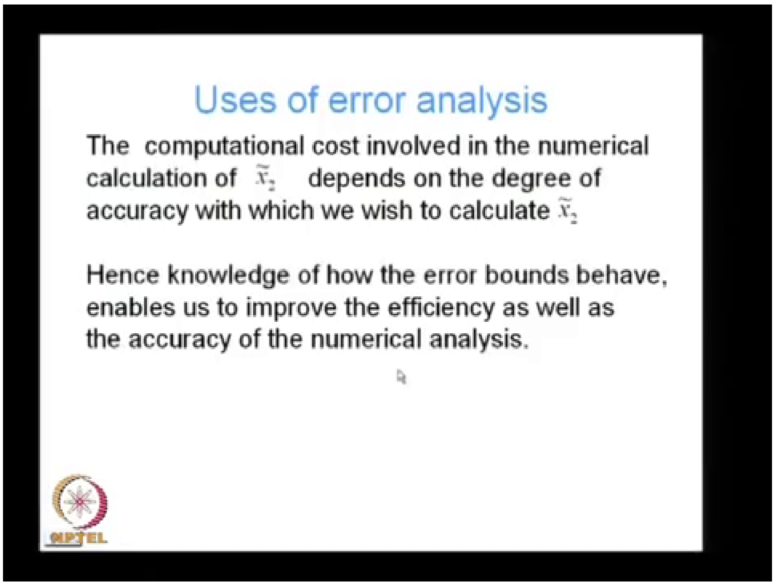
(Refer Slide Time: 51:51)



Why is that this is because we know that the relative error in x 1 x 2 may be as high as half in to 10 to the power minus 3 because, our error in x one is because x 1 known only up to three significant digits. So, the error in x one is bounded by half in to 10 to the power minus 3 and since the error in the product is bounded by the sum of the errors in the operands themselves, in that case the error in x 1 x 2 can be as high as half in to 10 to the power minus 3.

This follows from the fact as we mentioned then the bound on the relative error in x one x 2 is given by the sum of the sum on the bounds on the relative errors in x one and x 2. So, there is no sense in calculating x 2 to an accuracy greater than three significant digits However it does make a lot of sense of ensure that the relative error in the calculation of x 2 is not in excess of half in to the 10 to the power minus 3 that is the error in x 2 should not be more than three significant digits since that is going to worsen the relative error in the product.

(Refer Slide Time: 53:21)



Since the computational cost involved in the numerical calculation of x 2 tilde depends on the degree of accuracy with which we wish to calculate x 2 tilde. Thus error using error analysis we can find out to what degree of accuracy we wish to calculate x 2 tilde this gives us important information important information which helps us improve the efficiency as well as the accuracy of our numerical analysis.

So, at the end of this lecture let us sum up, so we have looked at different types of errors and we have looked at absolute errors we have looked at relative errors, we have looked at how bounds on those errors. We have found out how we can bound errors absolute errors as well as relative errors and we have shown how we can use those bounds to find out the total error estimate for a numerical solution, by bounding individual operations such as addition, multiplication, subtract, addition, multiplication, subtraction and division. We can actually find bounds on the total numerical solution next time we are going to continue our discussion on error analysis and look at how knowing, if we know the errors on individual variables x 1 through x n. And we can find out the error on y which is a function of individual variables x 1 through x n.