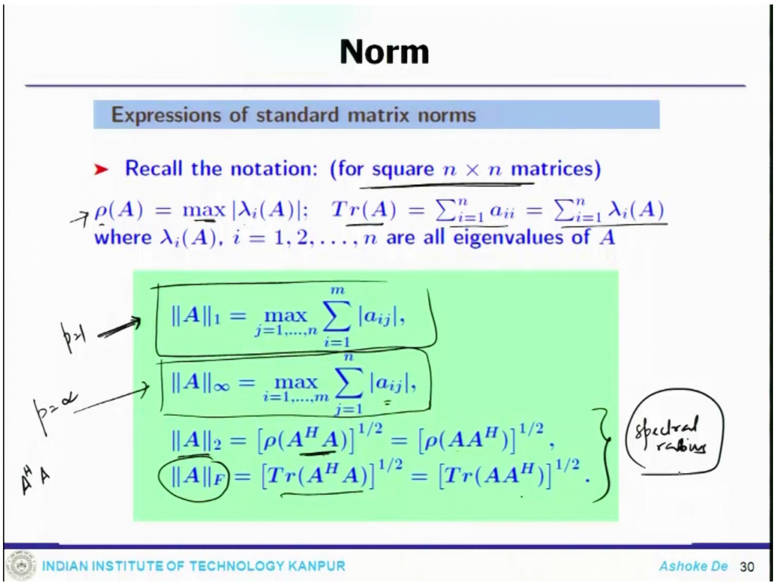
Introduction to Finite Volume Methods -I
Prof. Ashoke De
Department of Aerospace Engineering
Indian Institute of Technology, Kanpur

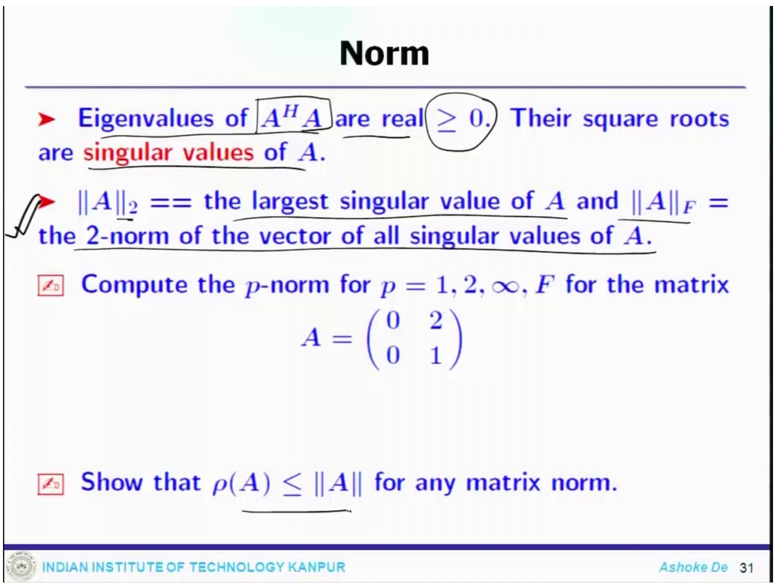
Lecture – 38
Error Analysis-I

(Refer Slide Time: 00:15)



So, welcome to this particular lecture on we will continue our discussion what we have been doing so far.

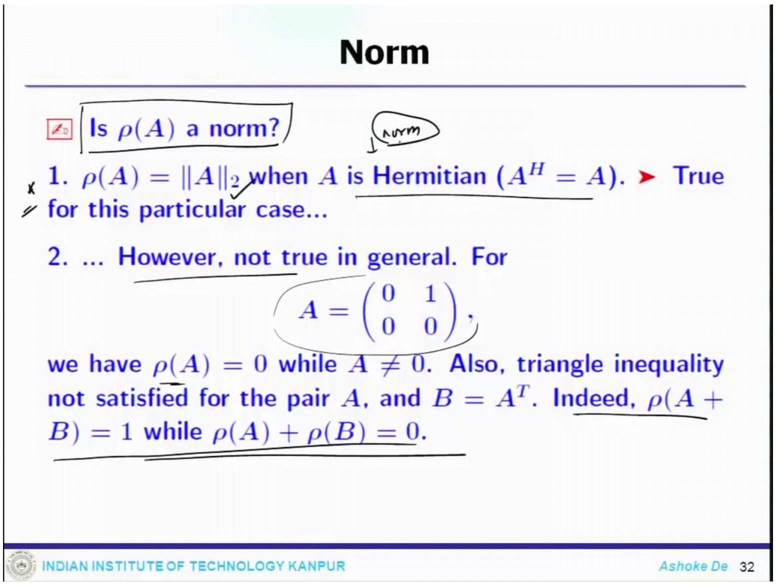
(Refer Slide Time: 00:23)



So, what are the information that one has to have? That information that one has to have is that eigenvalues of A Hermitian A; so, this is a product of the matrices. So, the eigenvalues of A Hermitian A are real; that means, they are all greater than 0. So, that is an important condition that you have. So, that their square roots are singular values of A.

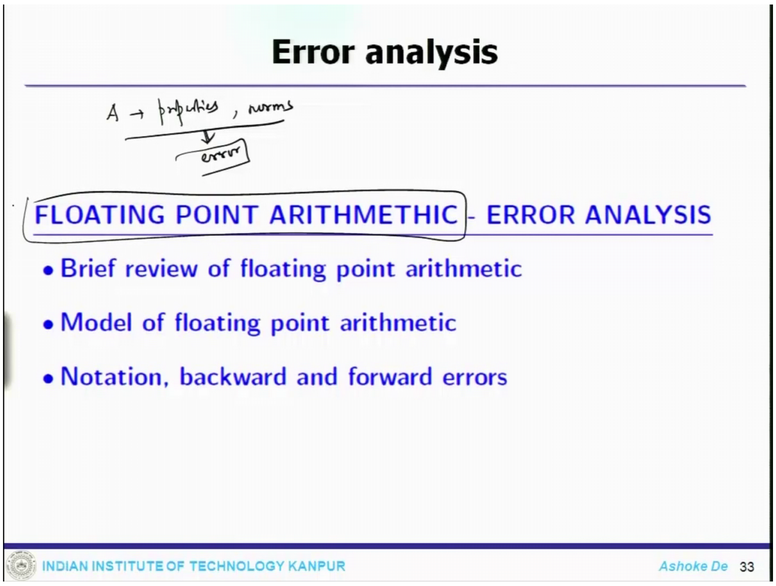
Secondly the A 2 or the magnitude A 2 is equivalent to the largest singular values of A and A F is the l 2 norm of the vector of all singular values of A. So, this is a very very important information that is applicable while getting these different norms for the matrix. Now, one can compute these things and show that the spectral radius is there.

(Refer Slide Time: 01:37)



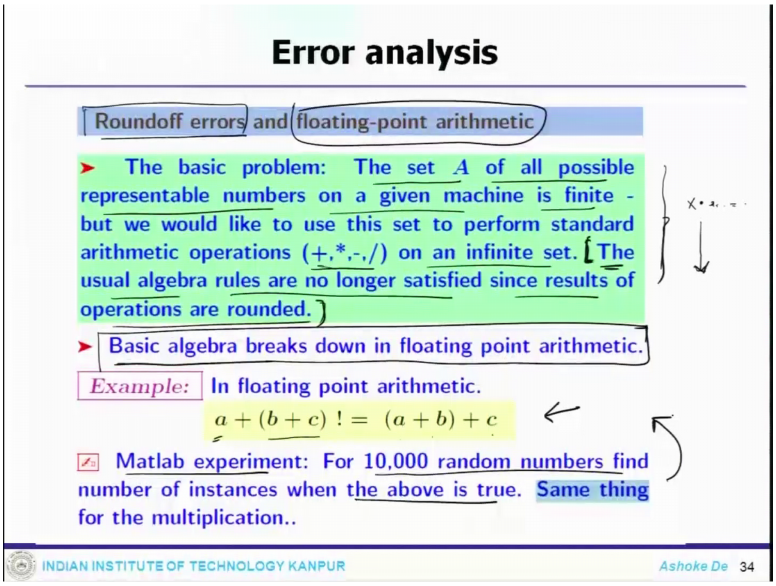
Now, the important question which brings to that is a spectral radius a norm that is an question. Now, this is possibly true for some particular cases, like spectral radius of A would be l 2 norm when A is in Hermitian matrix. So, that is a very very special case, if a is not an Hermitian matrix, this is not valid. So, this is a special case this validity can be obtained. Now, in general this is not true as I said if you have a matrix like that the spectral radius is 0 though a not equals to 0 and the triangular inequality not satisfied. So, this is also there rho A plus B equals to 1 while rho A rho B equals to 0. So, that is why for a special Hermitian matrix or the special case of Hermitian matrix this is possible, this will become a norm otherwise in general they are not.

(Refer Slide Time: 02:54)



Now, we have looked at some matrix property, then norms including the vectors norm. Now it brings that once you know this properties how do you get the error calculations and where do you get all this error? The error actually arises during the floating point arithmetic calculations. So, what we will do? We will look at some floating point arithmetic and what is the model for that and then we can look at some sort of a notation and forward and backward errors which are associated with this floating point arithmetic.

(Refer Slide Time: 03:52)



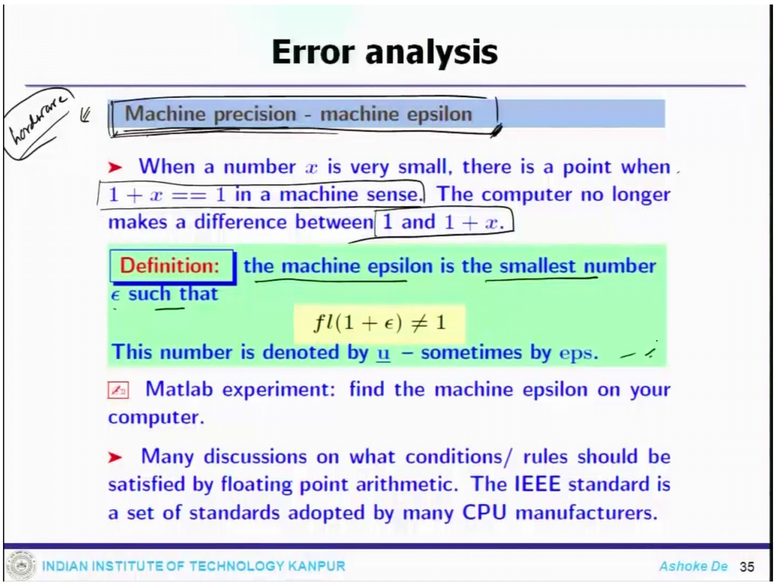
So, again coming back to the story of the same thing, the when you are dealing with any numerical system or numerical methods or approximations one of the crucial component is roundoff error that happens because how the post decimal digits are rounding off to the nearest possible digit and the error can be accumulated for certain problems. And that can lead to a huge error in your final calculation or there are some errors which are associated during floating point calculations.

So, what is the basic problem? The set of A of all possible representable numbers on a given machine is finite, this is a very very important and I would say loaded statement here. Set of A of all possible representable numbers on a given machine is finite, but what we would like to use the set of the set to perform certain operations like addition, subtraction, multiplication, division on an infinite set. So, what brings that the usual algebra rules are no longer satisfied since results of operations are rounded. So, this is where the problem comes.

And while talking about different kind of errors under the discussion of numerical schemes we have also discussed this roundoff error and that time we said the roundoff error appears because of these rounding of the numbers after the decimal points. So, as you rounding off and you go along with your calculation then these are problems because one point of time this can be I mean we concern compounding error and the error in your final solution could be very large. So, it has to do with your, the backbone of that is that the set of arithmetic operation that we are carrying out and this is not an usual algebra rules which are followed that is why the rounding off happens.

So, that tells that basic algebra breaks down in floating point arithmetic. So, that is what the problem is. So, the floating point arithmetic for example, what happens if you have a number a and you do b plus c and then a plus b plus c. So, this is where the problem starts appearing and all your numerical system you are doing some sort of addition, division, gradient calculations and all different calculations which are involved calculations associated with this system. And Matlab one can do some experiment these are some numbers for 1000 random numbers find number instance when the above is true. So, one can find or carry out this exercise by himself and try to check this particular problem.

(Refer Slide Time: 07:50)



Now, once you say that that rounding error that is appearing due to the calculations of the after decibel points or then that has to do with the machine precision. So, now, you see why we are talking about all this important property, because it is not only that you have an governing equations or the partial differential equation. And you use your numerics, discretize them get a linear system and you can straight away use some linear solver and get the solution; that is not as straight forward as it sounds. So, all these when you use your numerics you need to take care of all sort of uncertainties, all possible cases, all sort of exceptional cases, treatment of proper boundary conditions and finally, getting a proper discretize equations.

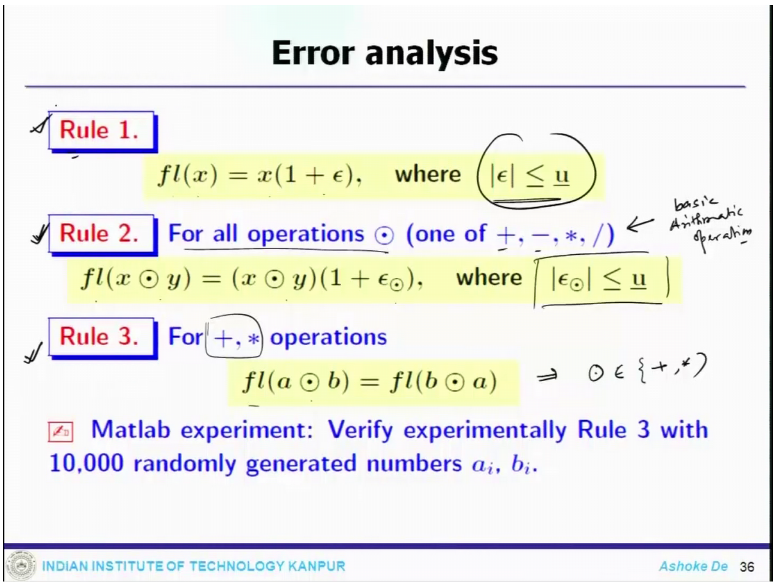
Now, when move down to the linear algebra or solution of the linear algebra these are the issues associated with that. How to get an efficient linear solver which would be cost effective which can give you accurate solution, but at the same time fast enough stable. So, it just like you have everything at the same time in your basket and you need to some sort of a maintain and balance to get an optimum solution for your system. And that is where it is always challenging for the CFD code how good your linear solver is how what is your order of accuracy of the numerics; that means, the discretize system is. And they are connected with some sort of an calculations which is associated with your hardware machine precision means these are hardware.

So, you may have a high end CFD code or the numerical method which is higher order accurate everything is taken care of very nicely, but then you do not have proper hardware or the proper machine precision then again you are going to get lot of errors. So, it is it has to have some sort of a handshaking between both the hardware and your set of mathematical instruction that you provide to the hardware through your coding and that is where the precision of the machine becomes very important.

Now, when a number x is very small; so, there is a point when 1 plus x equal to a in a machine sense. So, this I mean to say in machine sense it is not that. So, the computer no longer makes a difference between 1 and 1 plus x that is where the machine precision has been taken care of and there is no error associated with this precision. So, in a nutshell if I put them in a mathematical expression the machine epsilon is the smallest number epsilon such that floating of 1 plus epsilon not equals to 1. So, this number is denoted by u over sometime epsilon ok.

So, many discussions are going on and rules and all these things for this kind of calculations how to get a precised machine in error.

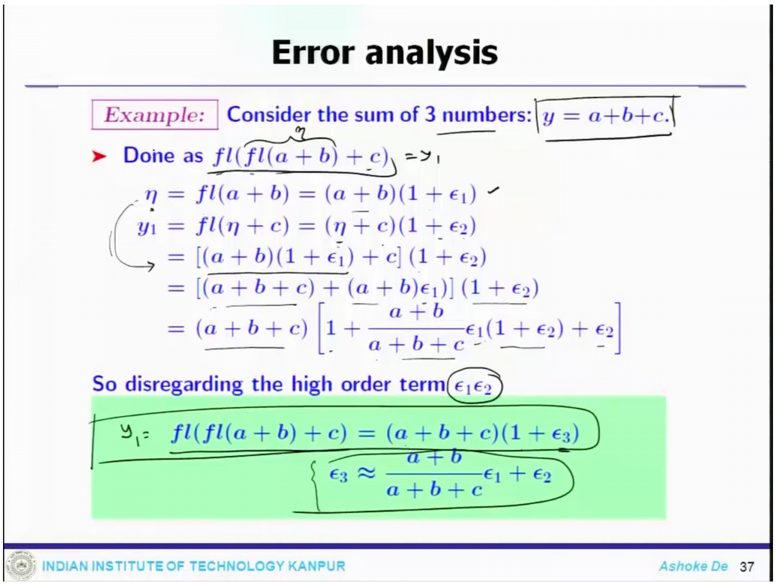
(Refer Slide Time: 11:48)



So, once you talk about the errors which are associated with the machine precision you have certain rules to follow and rules number 1, which says f l x equals to x into 1 plus epsilon where epsilon is going to be very small, here u sometimes epsilon this is the number ok. So, the epsilon satisfy this; rule number 2 for all operation, like any arithmetic operation addition, subtraction, multiplication, division f l x dot y must be x dot y one plus epsilon dot where epsilon dot is again less than equals to u. So, this is again an important rule which takes care the basic arithmetic operation. So, that takes care of that thing.

Rule number 3; which is very very specific to addition and multiplication operation. So, the floating point a dot b equals to floating point b dot a. So, this is only true for this dot belong the set of plus and multiplication only. So, these are 3 rules which are very important for handling the precision of the machine.

(Refer Slide Time: 13:42)



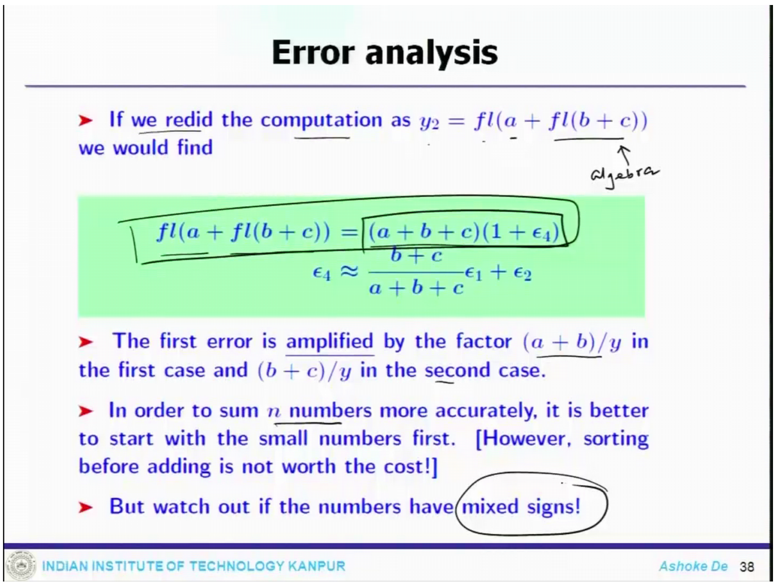
Now, one can take an example and quickly glance through the things you can consider the sum of 3 numbers which are very simple, I mean these are some sort of an school kid they do y equals to a plus b plus c you given a 3 numbers you just get the addition ok.

So, what you do done as floating point of floating point a plus b plus c. So, your eta equals to floating point a plus b which is this component, this is my eta which is equivalents to a plus b 1 plus epsilon; now y 1 is floating point eta plus c. So, that is the total component this is your y 1. So, it is eta plus c 1 plus epsilon 2, now what is eta? Eta is a plus b 1 plus epsilon. So, I am putting back the values of eta here. So, eta is a plus b 1 plus epsilon plus c into multiplied with 1 plus epsilon 2.

So, if you just rearrange this stuff I mean essentially some sort of an algebra it turns out a plus b plus c plus a plus b into epsilon with the whole thing the bracket gets multiplied with 1 plus epsilon. And if you take out that a plus b plus c out 1 plus a plus b divided by a plus b plus c epsilon 1 1 plus epsilon 2 and epsilon 2. So, they are terms which are associated with epsilon 1 epsilon 2. And then if you disregard those higher order term; that means, the product of epsilon 1 and epsilon 2, if epsilon happens to be small. So, this would not contribute too much, then the whole calculation of that which is essentially your y 1 is a plus b plus c 1 plus some epsilon 3.

So, epsilon 3 is nothing, but a plus b by these epsilon and epsilon 2 and we have actually discarded the values for higher order terms epsilon 1. So, these send you back the property of this. So, that is what it satisfies.

(Refer Slide Time: 16:51)

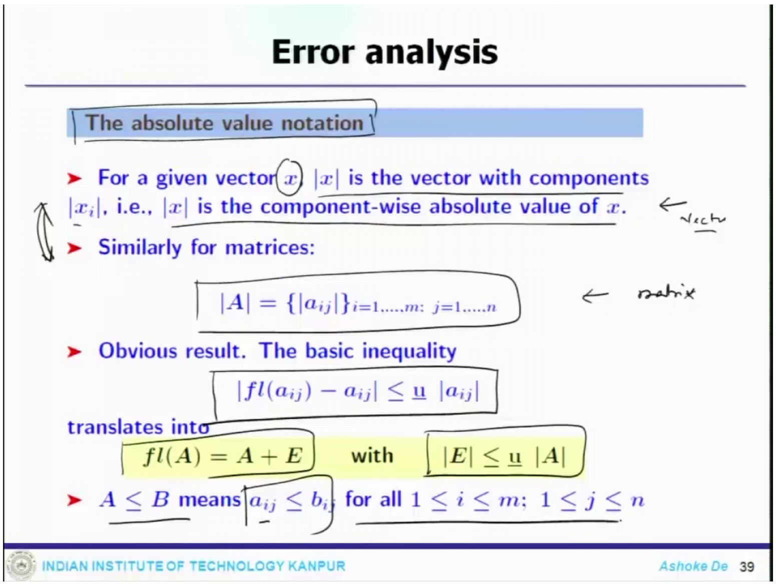


So now, if you redo the calculation or the computation as y 2 equals to floating point of a plus floating point b plus c. So, you need to do that algebra here.

So, if you need to do the algebra, then what you get the floating point a plus floating point b it would be a plus b plus c into epsilon 4. So, this also satisfied. So, the first error is amplified by the factor a plus b by y in the first case and b plus c by y in the second case. And in order of to sum n numbers more accurately it is better to start with the small numbers first ok, but one has to be careful if the numbers have mixed sign. So, that is also an important condition to be noticed.

But while we say that it you start with the smaller number first then the sorting also takes some sort of an computing cost. So, everything comes with some sort of an computing overhead and all these computing overhead is associated with your final output. So, given a option you need to have a very good code or numerical methods which should be efficient and should be fast enough.

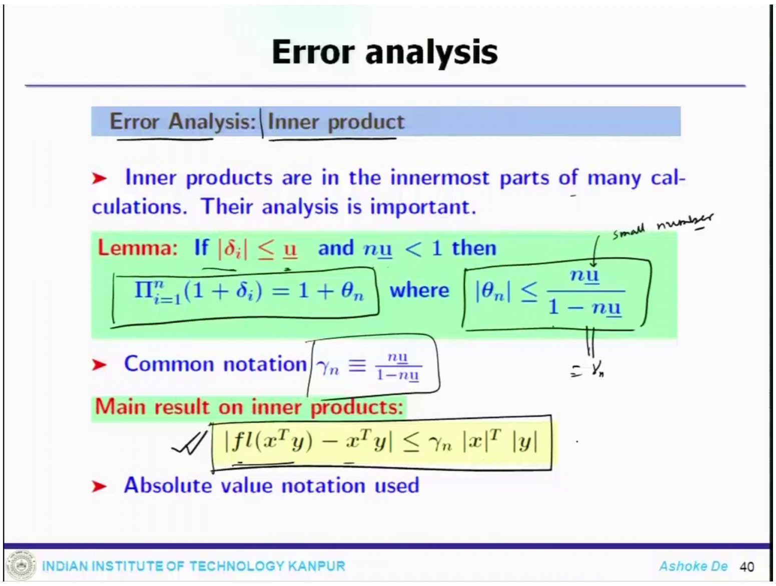
(Refer Slide Time: 18:44)



Now, moving ahead with this discussion of norms and others; now you can find out the absolute value notation. How one would define? You have certain given vectors x and mod x x and mod x is the vector with components x i. So, you have two different things that is x is the component wise absolute value of x. So, if you look at the similarity property of the matrices. So, mod of A equivalent to a i j mod of that. So, these are the two things one is applicable for vector other is for matrix and most of the time these properties are using the properties of vector. So, the obvious inequality that turns out to be floating point of a i j minus a i must be less than of u bar alpha i j which translate into floating point of A plus E with mod e is this.

So, the arithmetic operations the errors which are associated with the vector that we have seen can be extended now for matrix. And there could be also similar sort of errors and which means a less than equals to b means a i j less than equals to b i j for all i and j ok.

(Refer Slide Time: 20:46)

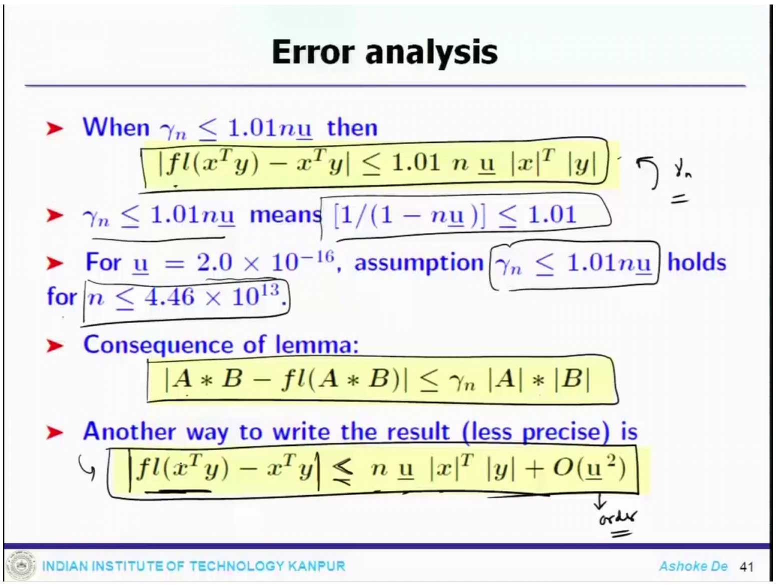


So, again one would like to look at the error for the inner product. So, we have defined the inner product earlier. Now, one would like to see the error product which is associated with these inner products.

So, inner products are in the innermost parts of many calculations, because if you are dealing with vector or if you are dealing with rather matrix comprises with some row and column vectors essentially or whether directly or indirectly you deal with lot of vectors calculations. So, that is where the inner product becomes an important key component, what is the lemma associated with that if delta i mod of delta i less than that small value u bar n into u bar less than 1. Then product of 1 plus delta i equals to 1 plus theta n where theta n must be less than equals to n multiplied with this u bar divided by 1 minus n u subscript.

This is the small number associated with machine precision. So, the common notation is that you can define gamma n which is essentially this particular term can be equivalent to gamma n. So, the result of this inner product is that floating point x transpose y minus x transpose y the magnitude of that satisfies less than equals to gamma n x transpose mod y. So, this is an very very important theory that is required for this computing of this matrix system.

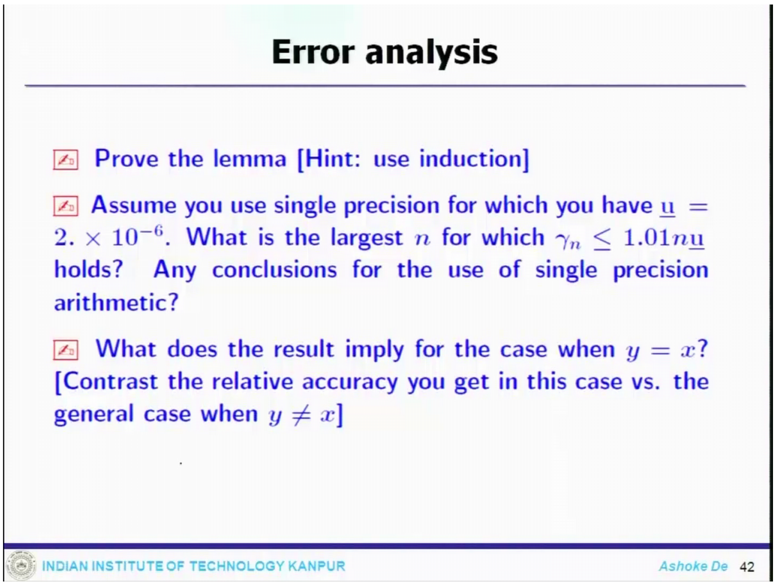
(Refer Slide Time: 22:51)



Now, one can look at this using some sort of an numbers. Let us say when your gamma n less than equals to 1.01 n u then floating point x transpose y minus x transpose y would be less than that. So, whatever we got here, it just uses that value and gamma n less than equals to 1 point means 1 by 1 minus n u less than equals to 1.01 or 1. So, every term which are associated with all these calculations gamma n and all these they have certain significance or physical significance.

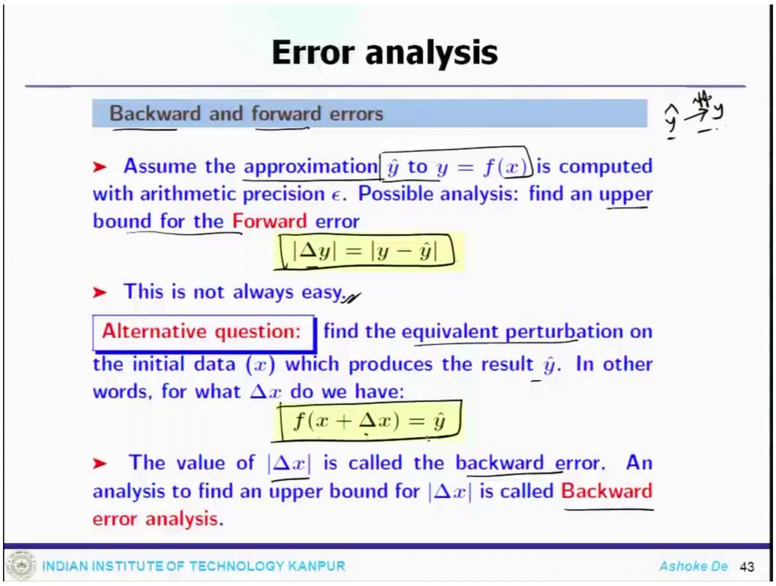
Now, for the smallest number being 2 into 10 to the power minus 16 the assumption gamma n less than equals to 1.01 holds for n less than equals to 4.46. So, one can always do some sort of an back calculation of all this. So, what is the consequence of this particular lemma is that A star B A multiplied with B minus floating point of A multiplied with B which must be less than equals to gamma n mod A multiplied with mod B. So, that is what the so, one can write in a other way (Refer Time: 24:37) floating point x transpose y minus x transpose y the magnitude of that less than equals to n the u subscript x transpose y bar which is order of small u square. So, u subscript square so that is the order. So, one can obtain the value separately.

(Refer Slide Time: 25:10)



So, this is some.

(Refer Slide Time: 25:13)

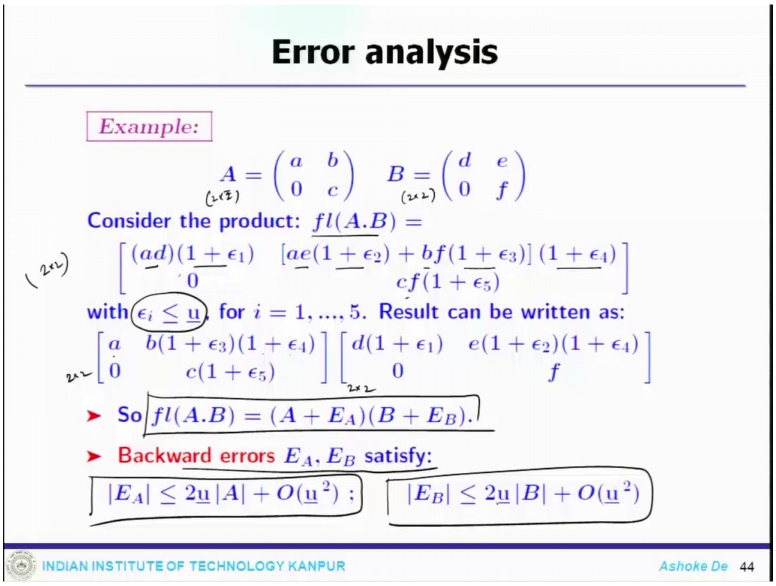


Now, once you move so, you got to know by this time how to calculate the error. So, now, one can say how we define forward error, how we can define backward errors. So, the definition is like these assume the approximation y hat to y, where y is a function and it can be computed with some arithmetic precision epsilon. So, the upper bound for the forward error would be delta y equals to y minus y hat or the magnitude of that.

So, that is the upper bound of for the forward error. So, this is not always easy. So, one has to note that. So, the alternatively one can find some equivalent perturbation on the original data or the initial data which produces the result y hat. So, in other words if I have to write I can write f x plus delta x which will lead to my upper hat ok. So, these this is the problem statement the approximation of y hat to y. So, the what we are doing that approximation to y. So, the initial data if it is perturbed it will lead to that. So, the value of delta x is called the backward error and you can do certain analysis and find that bound of the backward error.

So, when you talk about the forward error; that means, I am going from these to this. So, I am straight away finding what is my final these things and the difference of that is going to get me the forward error and when you are trying to do the backward error initial data is perturbed such that you get to the y hat with less error.

(Refer Slide Time: 27:33)



So, take an again an example which will allow you to understand that thing in a much clear way, let us consider the matrix a having 2 by 2 matrix; obviously, you have a b 0 c B another matrix which is also 2 by 2 you got d e 0 f. So, the similar elements like second row first element are having 0 values. So, you consider the product of the floating point product of a dot b.

So, how what do you get first term you get a d 1 plus epsilon 1 second term you get a e 1 plus epsilon 2 b f 1 plus epsilon 3 and the whole thing multiplied with 1 plus epsilon 4.

This exactly follows the multiplication rule that we have discussed earlier, this term would be 0 because they are both having 0 and this term the last term would be c f and this would be also a 2 by 2 system c f 1 plus epsilon 5.

So, all this epsilons 1, 2, 3, 4, 5 they are basically some small numbers. So, this results one can write you use some property of the matrices and write that A in the first matrix which you divide by 2 by 2 and the second one 2 by 2 you write a b into 1 plus epsilon 3 1 plus epsilon 4 0 c into 1 plus epsilon 5 d into 1 plus epsilon one e into epsilon 2 epsilon 4 0 f. So, what you get floating point A dot B is A plus E A B plus E B. So, the backward errors E A satisfy the criteria less than second orders and this is also less than B. So, this is what you get when you find out the forward and backward error. So, we will stop here and take it up in the next lecture.