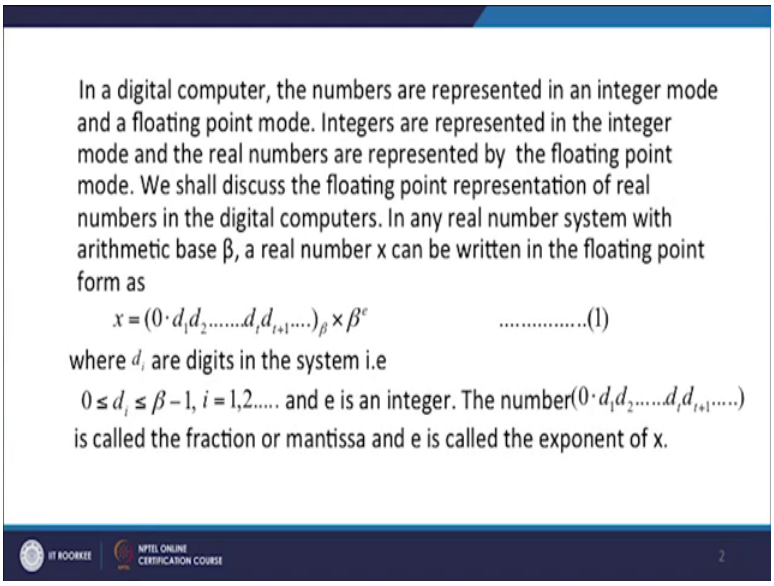
Numerical Linear Algebra
Dr. P. N. Agrawal
Department of Mathematics
Indian Institute of Technology, Roorkee

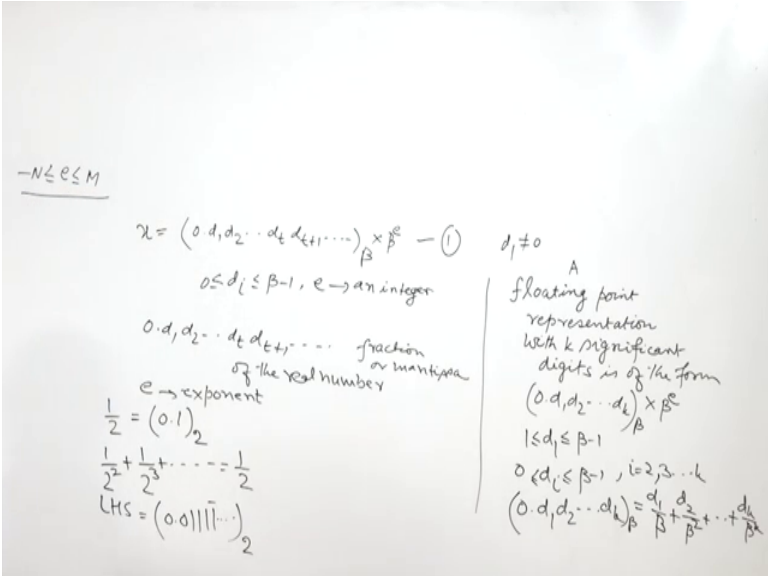
Lecture - 21
Floating point representation

Hello, friends I welcome you to my lecture on computer representations of a representation of numbers 2. So, in this lecture we shall talk about the floating point representation of real numbers, in a computer digital computer. In a digital computer the numbers are represented in an integer mode and the floating point mode, integers are represented in the integer mode, we have already discussed the integer point representation in our previous lecture and the real numbers are represented by the floating point mode. We shall discuss the floating point representation of a real number in the digital computer.

(Refer Slide Time: 01:04)



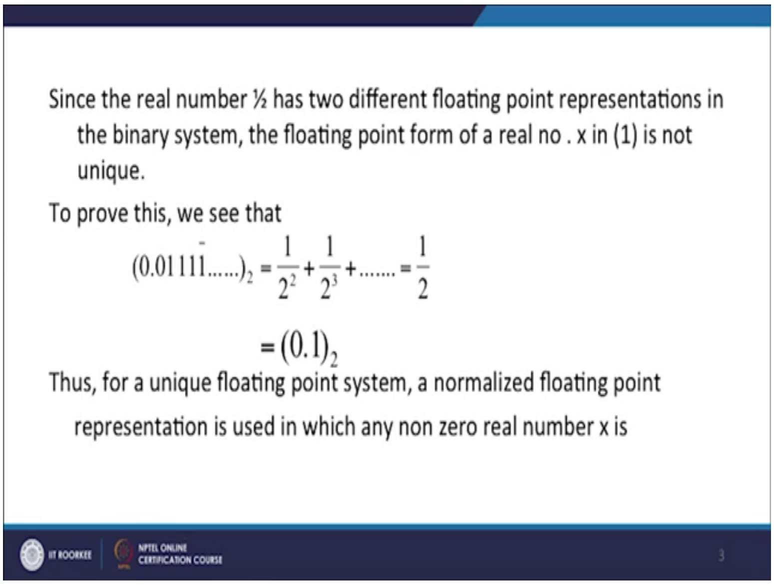
(Refer Slide Time: 01:17)



Now, in any real number system with arithmetic ways with a real number x can be in the floating point form as x equal to 0 point d 1 d 2 d and, so on dt d plus dt plus 1 and, so on beta into beta to the power e x equal to 0 point d 1 d 2 dt, dt plus 1 and so on base beta into beta to the power e. Now, here dis are integer digits in the system that is 0 less than or equal to di, less than r equal to beta minus 1 e is an integer the e is an integer the number 0 point d 1 d 2 and so on dt dt plus 1 and so on is called the fraction or mantissa of the real number of the real number system and the real number and e is called the exponent of x.

So, 0 point d 1 d 2 dt dt plus 1 and so on, this called fraction r mantissa of the real number and e is called as the exponent. Now, let us notice that if you look at the real number half, the real number half has two different floating point representations in the binary system if you look at half.

(Refer Slide Time: 03:05)

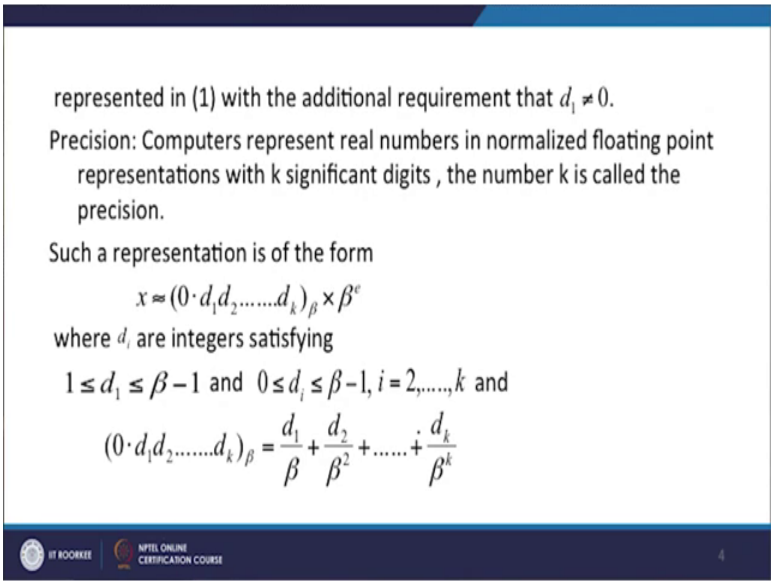


I can write in the floating in the binary system I can write it as 0.1 2 with base 2. So, this is 1 representation of the real number half in the of in the binary system and another 1 is I can write, since 1 by 2 square 1 by 2 to the power 3 and so on. If you look at this infinite series, this infinite series is equal to some of the infinite series is equal to half, we find that the left hand side here left hand side can be written as 0.01111 like this. So, this another representation of the real number half in the binary system.

So, since there are two representations of the real number half in this case of the floating point representation in the binary system, for unique floating point system a normalised floating point representation is used in which any nonzero is used in which any nonzero real number x is represented in 1, in this representation with the additional requirement that d 1 is not equal to 0.

So, in the case of a normalised floating point representation, the representation of x is written with the additional requirement that d 1 is not equal to 0. So, that we have only 1 representation here this representation will then be not considered, we will have half equal to 0.1 in the floating point representation. So, now computers represent real numbers.

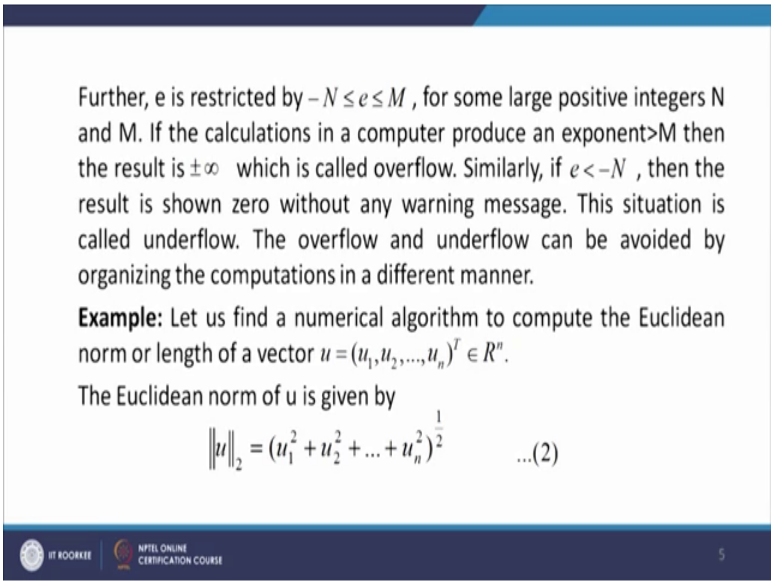
(Refer Slide Time: 05:00)



In normalized floating point representations with say some significant digits let us take k then this number k is equal to called as the precision such a representation is of the form. So, if you have a floating point representation with k significant digits, a floating point representation with k significant digits is of the form 0.d 1 d 2 d 3 and so on ,up to d k with base beta into beta to the power e, and where since it is a normalised floating point representation as we said earlier for a unique floating point representation, we have to use normalised floating point representation in the case of normalised floating point representation d 1 is not assumed equal to 0; that means, that d 1 starts with 1 and goes up to beta minus 1 and d 2 d i, they start with 0 and go up to beta minus 1 for i equal to 2 and 3 and so on up to k.

Now, so we can say that 0.d 1 d 2 d k can be written as this will be equal to d 1 upon beta d 2 upon beta square and so on, d k upon beta to the power k. Now further the exponent e is restricted minus N is less than or equal to e less than or equal to M..

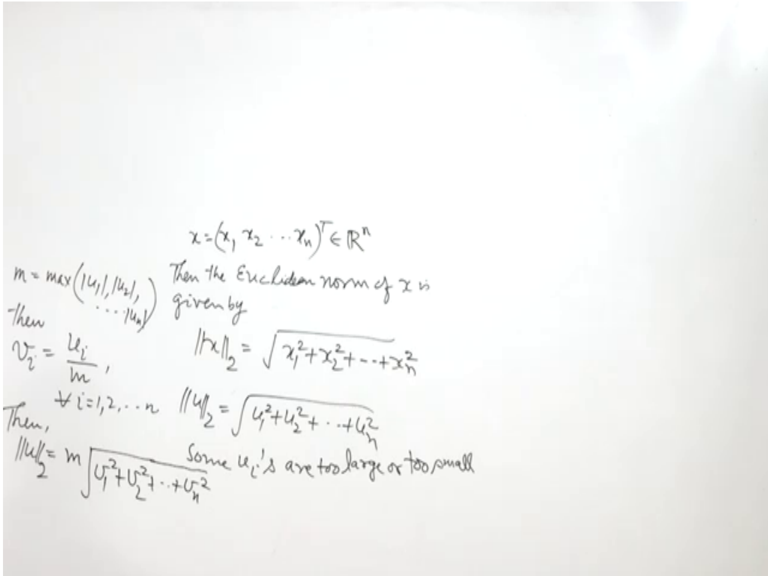
(Refer Slide Time: 07:41)



So, there is a restriction on the value of e it is bounded below by minus N and bounded above by M, for some large positive integers N and M, if the calculations in a computer produce an exponent greater than M, then the result will be plus minus infinity which is called overflow.

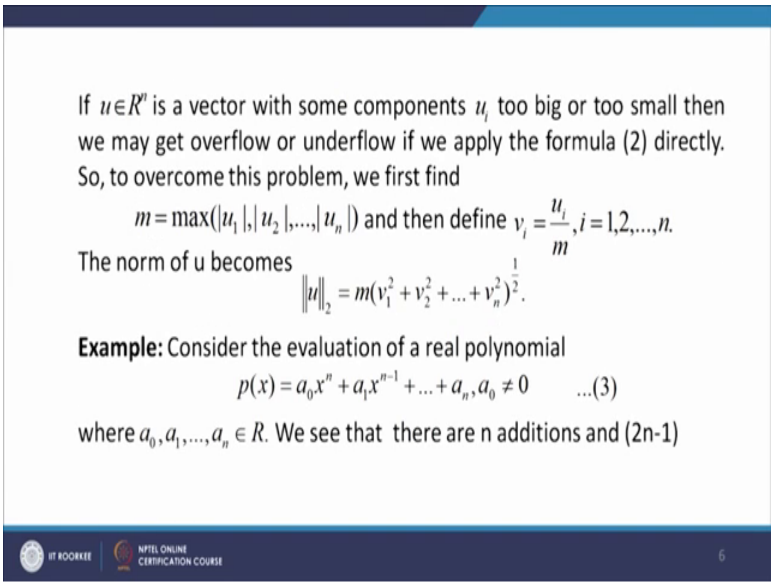
Similarly, if e is less than minus N then the result is shown 0 without any warning message, in this situation we say that there is underflow the overflow and underflow can be avoided by organising the computations in a different manner. Let us take a numerical find a numerical algorithm to compute the Euclidean norm are length of a vector in R n, so let us take a vector in R n x equal to x 1 x 2 x n belonging to a R n this t means transpose.

(Refer Slide Time: 08:46)



So, we are writing it as a column vector, so x 1 x 2 x n then the norm of x the Euclidean norm of x is given by ok. So, suppose we want to find the Euclidean norm of a vector are length of a vector say u, here have taken an example u equal to u 1 u 2 u n transpose belonging to R n, then the Euclidean norm of u will be equal to square root u 1 square plus u 2 square and so on u n square.

(Refer Slide Time: 10:00)



Now, if this vector u belonging to R n is such that some components u i are too big or too small. So, norm of u 2 this equal to square root u 1 square plus u 2 square u n square.

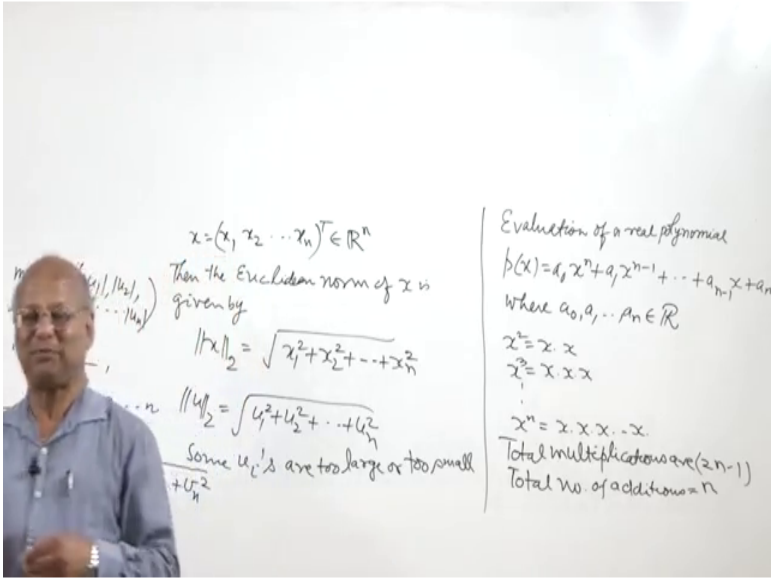
Now, if some u i some u i's are too large or too small, then we can have a situation where we have a overflow or underflow. So, to avoid the case situation of overflow or underflow what we do is we find the maximum value m of mode of u 1 or we have say norm of u 1 norm of u 2 and so on, because they are real numbers we can write modulus of u 1, so we can write m equal to mode of u 1 maximum of.

So, let us find the maximum value of mode of u 1 mode of u 2 u 1 u 2 u n are components of the vector x vector u. So, then let us define v i equal to u i upon m, for i equal to 1 2 and so on, up to n now; obviously, m is not equal to 0 because if m is equal to 0 then u 1 u 2 un all will be 0 norm of u will be 0. So, we can assume that m is equal to not equal to 0.

Now, so let us define v i equal to u i over m then we can find norm of u 2 by the formula m times under root v 1 square plus v 2 square and so on, v n square since we are dividing each component by its modulus the u i's will not be too large or too small. So, in order to avoid overflow or underflow and to calculate the length or the Euclidean norm of the vector u, we use an alternative technique that is we divide each component of u by the maximum value of mode of u 1 mode of u 2 mode of u n.

And then write norm of u 2 equal to this norm of u 2 you can see can be written in alternate way like this. So, m times square root v 1 square plus v 2 square plus v n square by playing an alternative technique, we can find the Euclidean norm or length of the vector u and we can avoid overflow or underflow. Now, let us go to another situation suppose we have to evaluate the real polynomial.

(Refer Slide Time: 13:25)



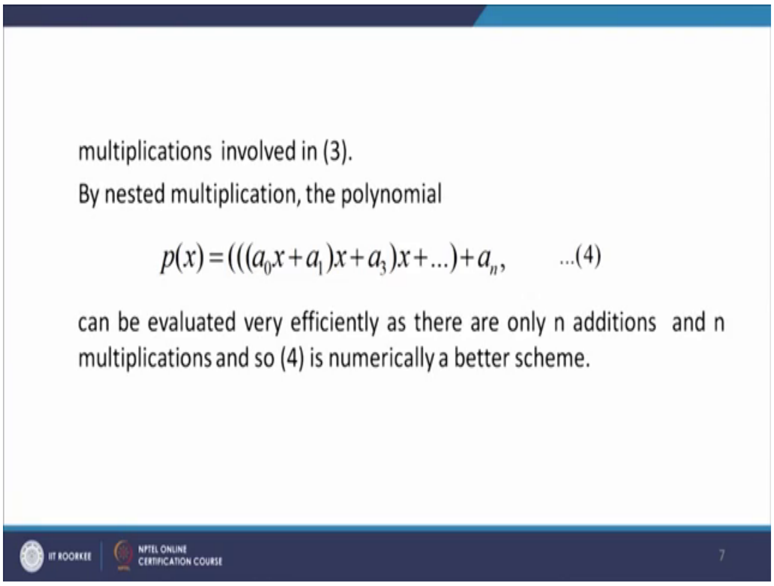
Evaluation of real polynomial, so let us we are given a polynomial p x equal to a naught x to the power n plus a 1 x to the power n minus 1 and so on, a n minus 1 x plus a n we want to evaluate the value of p x for a given value of x and here a naught a 1 a 2 a n minus 1 a n are real numbers. So, to evaluate the value of p x for a given you can see that there are n multiplications see there are n minus 1 multiplications to calculate x square x cube x to the power n x square can be calculated by x into x.

So, there is 1 multiplication x cube can be calculated by x into x into x into x, so there are 2 multiplications and so on, while calculating x to the power n we have to multiply n minus 1 times. So, there are n minus 1 multiplications to evaluate x square x cube and so on, x to the power n and furthermore we have to multiply a naught by x to the power n, that is 1 multiplication a 1 we have to multiply it to x to the power n minus ones there is 1 more multiplications and then we have n minus 1 multiply to x, so there is 1 more multiplications.

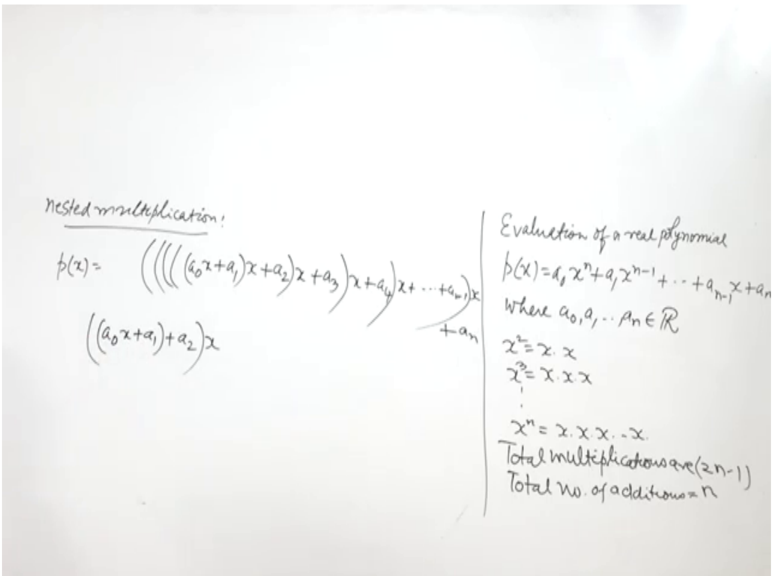
So, we have total 2 n minus 1 here and multiplications are there and here, n minus 1 multiplication are there. So, we have total multiplications involved 2 n minus 1, and there are n additions, 1 addition here, 1 addition here and so on the addition there, so total number of additions. So, to calculate v x in a digital computer for a given value of x it has to carry out n additions and 2 n minus 1 multiplication.

Now, we all we are going to see that by applying alternative technique we can evaluate p x by just n multiplications and n additions. So, we will apply the nested multiplication technique.

(Refer Slide Time: 16:18)



(Refer Slide Time: 16:33)

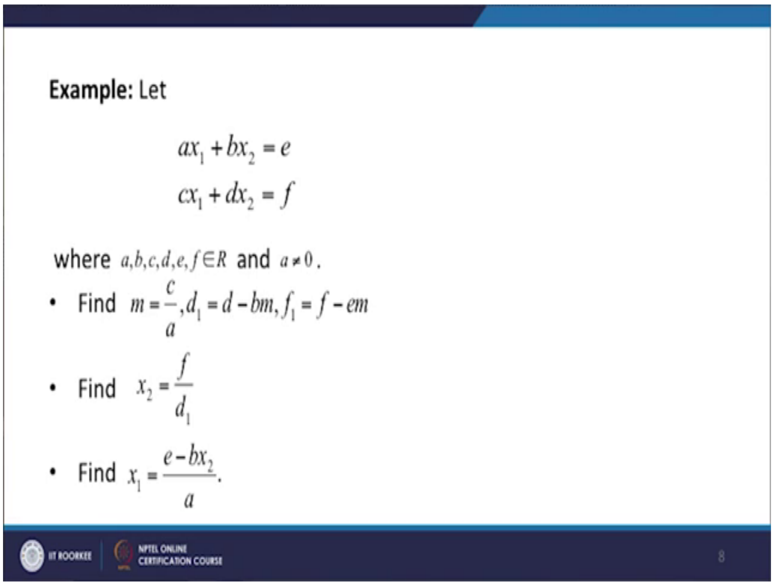


So, let us evaluate p x by nested multiplication, p x can be written as you can see a naught x plus a 1 multiplied by x plus a 2. So, that we get a naught x square plus a 1 x plus a 2 and then again we can multiply by x and add a 3, and then we can multiply by x and add a 4 and so on ok.

So, let us say I we are write like this into x plus we have a n minus 1 into x plus a n, now here you can see we have 1 addition here, 1 addition here 1, addition here 1, here, 1 here, 1 here, we have total n additions and how many multiplications are there you can see just by example I can show you a naught x plus a 1 plus a 2 into x, see if you want to calculate only a naught x plus a 1 we have 1 addition 1 multiplication a naught into x 1 multiplication and a naught x plus a 1 1 addition.

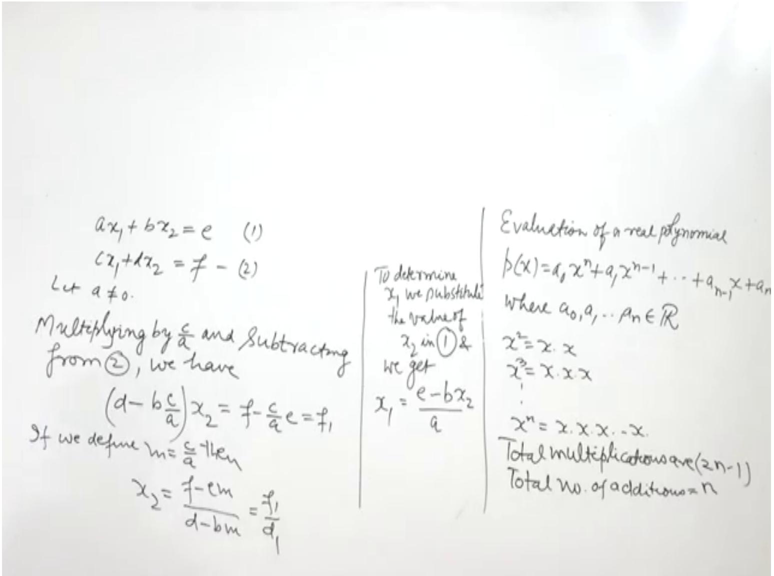
If you want to calculate a naught x square plus a 1 x plus a 2 then you have here, 1 multiplication, 1 addition, then 1 addition, and 1 multiplication. So, we have 2 multiplications and 2 additions, and similarly if you want to calculate this p x 4 n equal to 3 you have 3 additions 3 multiplications in general we have n additions n multiplications. So, we can imply this nested multiplication technique to determine the value of p x where there will be only n addition n multiplications. So, numerically we can say that this nested multiplication is a better numerical scheme.

(Refer Slide Time: 18:49)



Now, let us say suppose we want solve a system of linear equations by Gaussian elimination method.

(Refer Slide Time: 19:03)



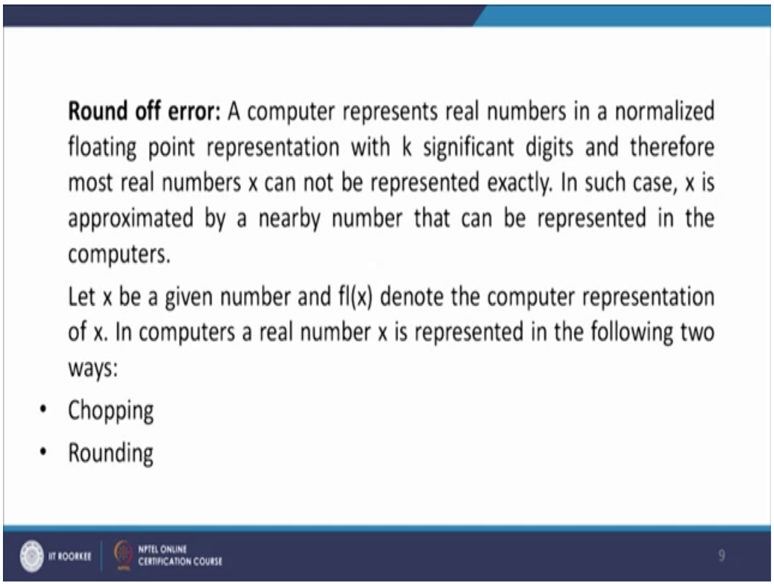
So, in the Gaussian elimination method the equations are let us say a x 1 plus b x 2 equal to c a b a x 1 plus b x 2 equal to e and then c x 1 plus d x 2 equal to f, then what we will do let us assume that a is not equal to 0. So, what we will do we will multiply this is equation let us say 1 this is equation 2 multiplying 1 by c by a and adding and subtracting from the second equation subtracting from 2 ok.

We shall have c by a we are multiplying to the equation 1, so the coefficient of x 1 will become c x 1 which when subtracted from c x 1 gives 0 into x 1 and then the coefficient of x 2 will become d minus b in to c by a equal to f minus c by a into e. So, if I define if we define m equal to c by a we shall have x 2 equal to f minus em divided by d minus bm.

So, we can calculate this and I can call this as f minus em I can call as f 1, this is f 1 and this I can call as d 1 d minus bm I can call as d 1. So, then f x 2 equal to f 1 by d 1 once we have evaluated the value of x 2 we can go to the first equation substitute the value of x 2 to determine x 1, we substitute the value of x 2 in equation 1 and we get x 2 equal to e minus bx 2 divided by a. So, we can put the value of x 2 equal to f 1 by d 1 here and determine the value of x 1.

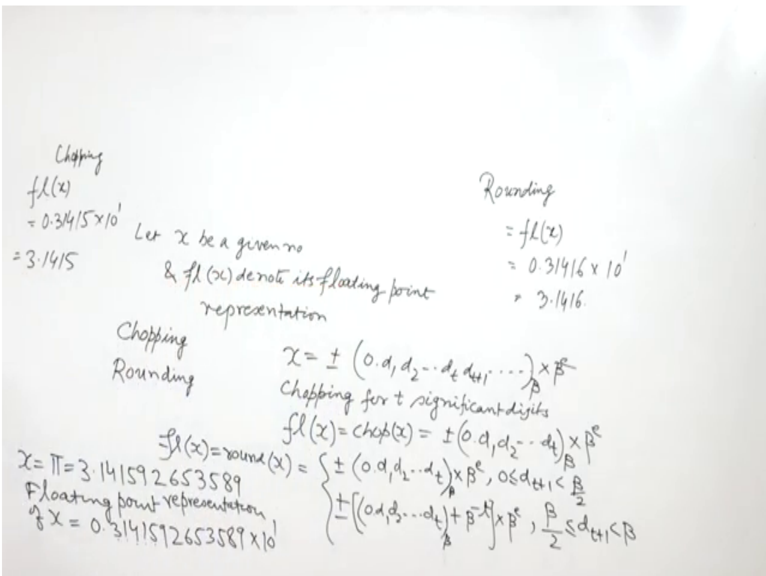
So, we can use this scheme of Gaussian elimination to solve the system of linear equations in the 2 known x 1 and x 2. Now let us see a how we can I mean discuss what do we mean by round off error, a computer represents real numbers we have seen in a normalized floating point representation with a certain number of significant digits which we can take as k.

(Refer Slide Time: 22:52)



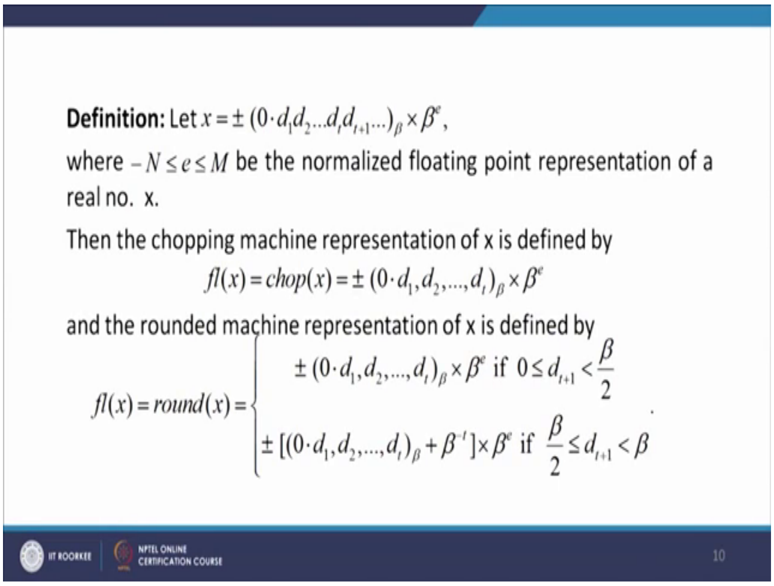
And therefore, most real numbers x cannot be represented in an exact manner. So, in such a case x is approximated by a nearby by number that can be represented in the computer. So, let us say let x be a given number and, fl x denote its floating point representation, so let us say x be a given number.

(Refer Slide Time: 23:31)



And fl x denote its floating point representation, then real number can be x represented in the following 2 ways chopping, rounding because it is representing the real number x up to and certain number of significant digits, let us say take k then we have to chop that number or we have to round of round that number to k significant digits. So, while writing fl x we will apply 2 things either chopping or rounding and what will happen in the case of chopping and in the case of rounding it is given in this definition.

(Refer Slide Time: 24:52)



So, let us say x equal to plus minus 0 point d 1 d 2 d t d t plus 1 and so on, with base beta into beta to the power e where minus n is less than or equal to e less than or equal to m let this be the normalised floating point representation of a real number x ok.

So, let us again recall in the normalised floating point representation d 1 is assumed to be nonzero; that means, d 1 is greater than or equal to 1, but less than or equal to beta minus 1 while the other d i is from i equal 2 onwards they vary from 0 to beta minus 1, they are greater than or equal to 0, but less than or equal to beta minus 1. So, let us say x equal to plus minus 0 point d 1 d 2 d t d t plus 1 and so on, beta into beta to the power e this is the floating point representation normalises floating point of a real number x.

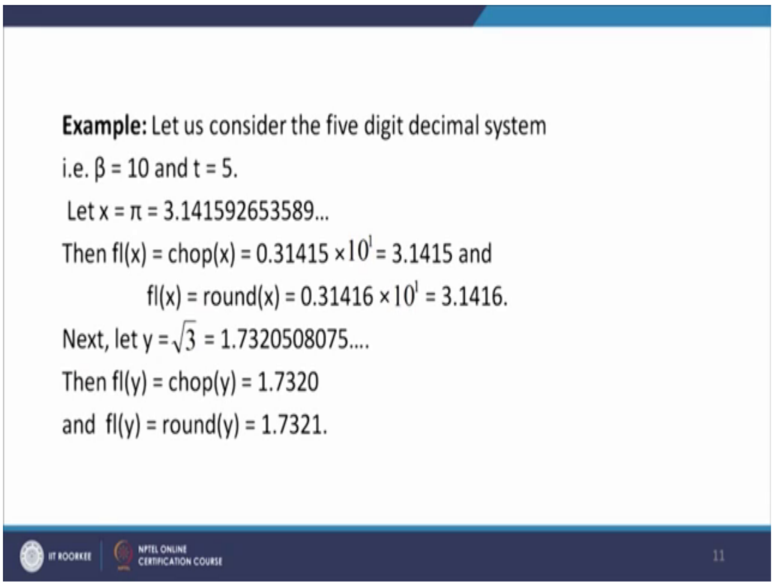
Now, if we chop these 2 say t significant digit here we are taking k equal to t. So, then the chopping machine representation of x to t significant digits, chopping for t significant digits will give us chop x equals to fl x equal to we simply ignore all the digits that appear after d t in the case of chopping. So, chopping for t significant digits will give us the floating fl x equal to chop x equal to plus minus 0 point d 1 d 2 and so on, d t with base beta into beta to the power e.

And in the case of rounded machine representation of x what we will have fl x equal to round x this will be equal to plus minus 0 point d 1 d 2 and so on d t again up to t significant digits, provided 0 is less than or equal to d t plus 1, less than beta by 2 and plus minus 0 point d 1 d 2 d t plus beta to the power minus t 0, we have multiplied here into beta to the power e we have to multiply.

So, what we will have in the case of this in the case where 0 is less than r equal to d t plus 1 less than beta by 2, we simply write 0 point d 1 d 2 d t there is no change in d t we simply ignore the remaining digits into beta to the power e, but in the case of beta by 2 being less than or equal to d t plus 1 or you can say d t plus 1 greater than or equal to beta by 2 and less than beta. We add beta to the power minus t beta to the power minus t now beta to the power minus t means 1 upon beta to the power t we add to the t th; t th place the t significant digits; that means, that the t th significant digits is added by 1, so 1 over beta to the power t.

So, this is what we do in the case of rounding now let us look at an example and see how we apply this chopping and rounding, let's consider the 5 digits decimal system.

(Refer Slide Time: 30:00)



That means we are taking k equal to 5, here and we are taking decimal system means we are taking beta equal to 10. So, beta equal to 10 is given and t is equal to 5 is given x is given to be pi equal to approximate value of pi we are taking here pi equal to 3.14159265 and then we have 3589.

Now, the floating point representation of this floating point representation will be in the floating point representation of x it will be written as 0.3141592653589 into 10 to the power 1 beta is here 10 exponent e is equal to 1, because we are writing it in this form sign is plus ok.

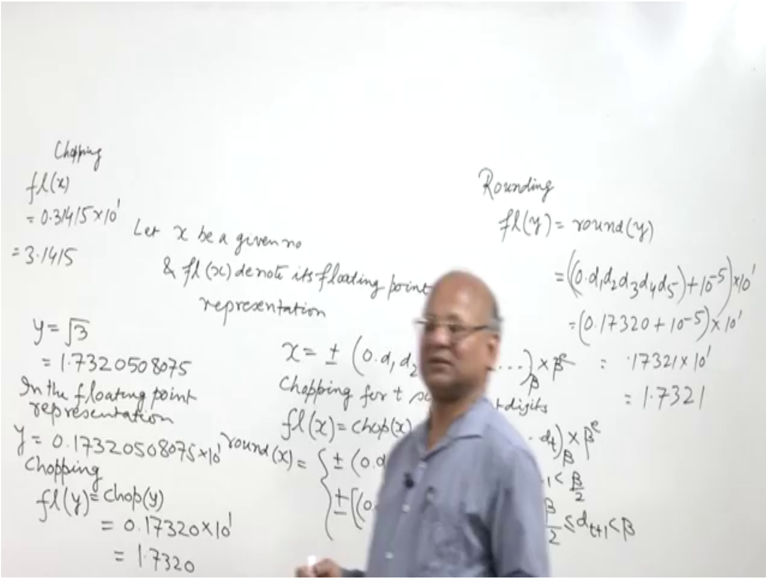
Now, we have to consider t significant k we have to consider 5 significant digits, because it is given that we have to consider 5 digit decimal system 5 digit means 1 2 3 4 5 now this we have to see this 6 digit, if we are doing chopping first let us first chopping we have do ok.

So, fl x in the case of chopping what we do after 5 significant digits, we just I mean ignore the remaining digits this means 0.31415 into 10 to the power 1. So, in the case of chopping floating point representation of x will be point 31415 into 10 to the power 1 which means that 3.1415 ok.

Now in the case of rounding, we have to see the 6 digit d t plus 1 t is equal to 5 here. So, this 6 digit is 9 here and you see that if the d t plus 1 is lying between 0 and beta is equal to 10 here. So, between 0 and 5 0 less than or equal to d 6 less than 5 then we do not change d t, but if d t plus 1 is more than or equal to beta by 2 that is 5, but less than 10 we add 1 by beta to the power t in the t th place ok.

So, what we do here this will be equal to in the floating this fl rounding in the case fl x in this case it will be equal to 3.1415, 5 will become now 6 we have to add 1 at the t th place that is the fifth decimal. So, 3.1416 multiplied by 10 to the power 1 you see here beta to the power minus 10 t is equal to 5 beta is equal to 10. So, 10 to the power minus 5; 10 to the power minus 5 means 1 by 10 to the power 5 1 by 10 to the power 5 when you add to 0.31415 it will become 0.31416, so this into this is equal to 3.1415 so in the case of rounding it will become 3.1416. Now, in the other example let’s say y equal to root 3.

(Refer Slide Time: 34:44)



So, let us say y equal to root 3 and the value of root 3 we and take as 1.7320508075, so we will write in the floating point representation y as 0.17320508075 into 10 to the power 1 ok.

Now, here again we are considering 5 digit decimal system. So, beta we have taken equal to 10 e is equal to 1, so in the case of chopping we will have fl x y equal to chop y now we are asked to t take t equal to 5. So, we take 5 significant digits 17320 and discard the remaining ones.

So, we shall have 0.17320 into 10 to the power 1 or we will get 1.7320 now in the case of rounding fl y equal to round y. So, in the case round y we have there are 2 (Refer Time: 36:36) cases 1 is 0 less than or equal to d t plus 1 less than beta by 2 the other 1 is b by 2 less than or equal to d t plus 1 less than beta t is equal to 5 so; that means, d 6, d 6 means 1 d 1 d 2 d 3 d 4 d 5 d 6 this is d 6 d 6 is 5 here and beta is 10. So, 5 by 2 sorry 10 by 2 means 5 is less than or equal to d 6 less than 10 ok.

So, since d 6 is equal to 5, we have to apply this case it will mean that we have 0 . d 1 d 2 d 3 d 4 d 5 plus beta to the power minus t; that means, 10 to the power minus 5 multiplied by 10 to the power 1, now d 1 d 2 d they are 0.17320 plus 10 to the power minus 5 into 10 to the power 1. So, we have 0.17321 into 10 to the power 1 which gives you 1.7321. So, this how we calculate fly in the case of 5 digit decimal system with this I would like to conclude my lecture.

Thank you very much for your attention.