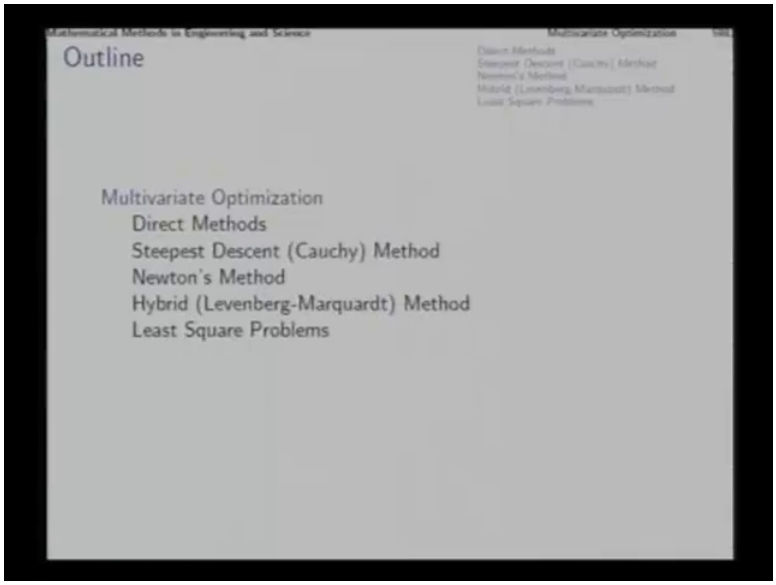
Mathematical Methods in Engineering and Science
Prof. Bhaskar Dasgupta
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur

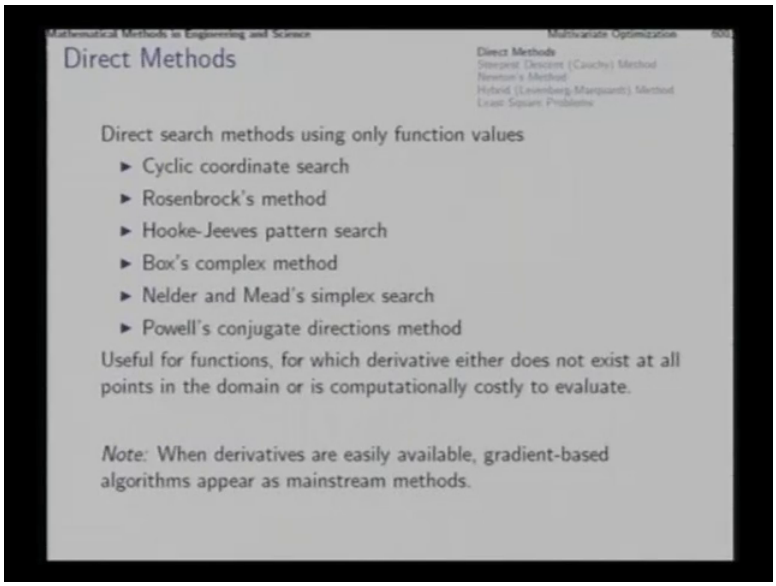
Module - IV
An Introductory Outline of Optimization Techniques
Lecture - 03
Multivariate Optimization

(Refer Slide Time: 00:38)



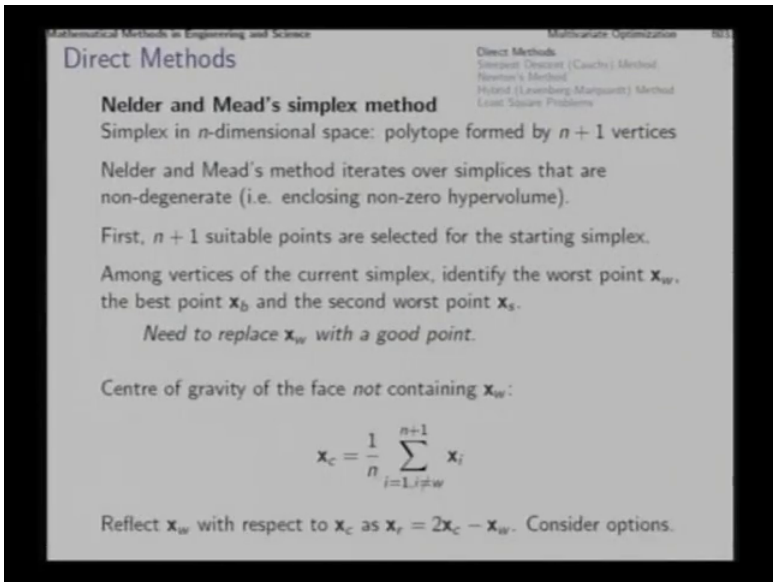
Good morning. In the previous lecture on optimization, we studied the conceptual framework, conceptual background of multivariate optimization. In this lecture, we will be studying some of the methods to solve optimization problems. Now, some of the methods are called direct methods that is because they use only the function values and not derivatives. And one direct method we will study first, and then we will continue into the study of methods based on variance also that is steepest descent, Newton's method, and hybrid method.

(Refer Slide Time: 01:02)



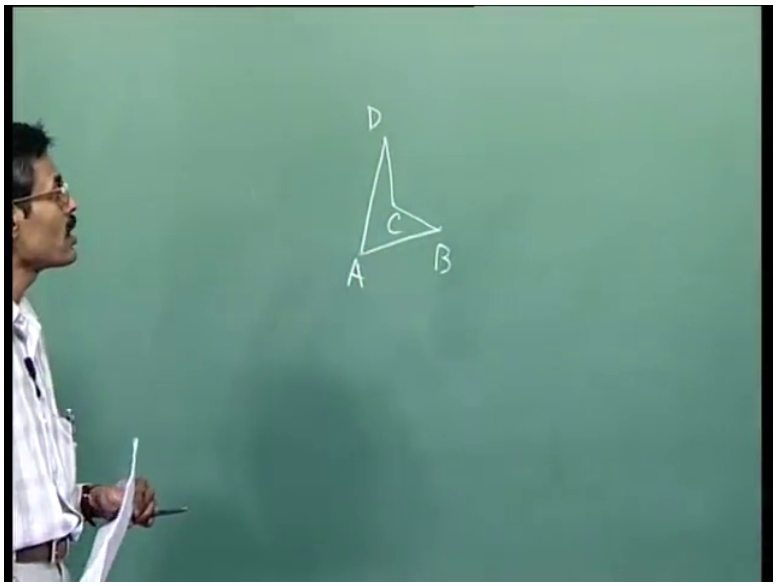
So, first the direct methods, these are some of the direct methods and some of the methods are very simple in operation. And one of these methods I will use here for elaboration that is Nelder and Mead's simplex search method. And all these methods utilize only function values and do not use the variance or derivatives. And therefore, they are of great value for those functions which are not differentiable that is which are not differentiable at several points in the domain. For such functions these are quite important, because derivative based methods will not be appropriate for such functions. Even when derivatives exist, derivatives are defined there are also quite often we find that using this kind of a method which does not use a derivative is helpful if the derivative evaluation is computationally quite costly. However whenever derivatives exist, we use a method which gives us derivative because using derivatives every step of the algorithm can make longer sweeps. So, first we study derivative free method or direct method that is Nelder and Mead's simplex search method.

(Refer Slide Time: 02:24)



Now, those of you who have a background of linear programming, they know that there is a simplex method in the linear programming methods also. Now, this simplex method is different from that and this is called non-linear problems, and that is why to differentiate it from the simplex method for LP problems we call it as Nelder and Mead's simplex search method. Now, first thing what is a simplex? In two-dimensional space, a triangle is the simplex that is a polygon composed of three vertices; in three-dimensional space, a tetrahedron is a simplex. Now, in two-dimensional space, among triangle quadrilateral, pentagon, hexagon and so on what is so special for a triangle, the special property of a triangle which is not shared by any other polygons is that triangle by nature is convex.

(Refer Slide Time: 03:34)



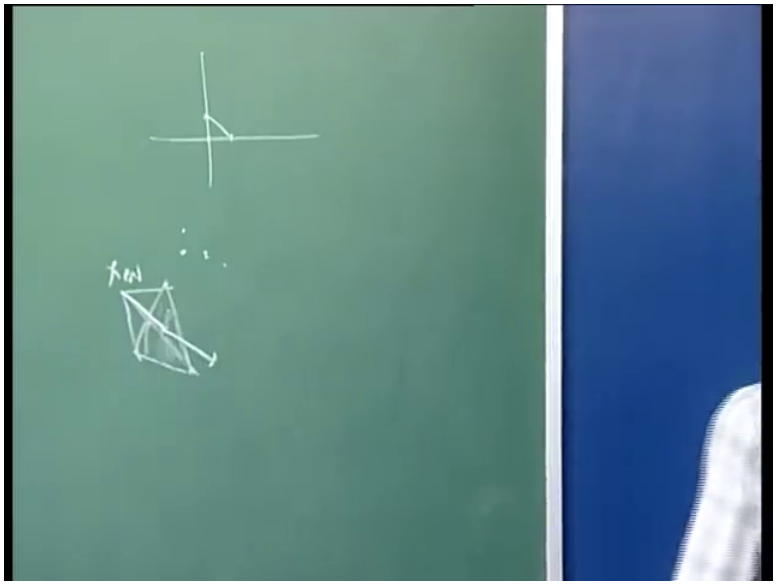
For example if I give you four points in sequence A, B, C D, the quadrilateral that you form out of it in that sequence does not have to be convex, see this is not convex. On the other hand, if I give you only three points and ask you to frame a triangle you cannot frame a triangle without the triangle being convex. So, by nature by definition itself a triangle is a convex region that is an advantage. Similarly, in three-dimensional space tetrahedron as long as it is non-degenerate that is all the four points are not in the single plane in that case a tetrahedron cannot be formed.

So, a tetrahedron is by nature convex to begin with similarly in an n-dimensional space a similar geometric entity a similar geometric figure a polytope composed of n plus 1 vertices is s simplex. So, a triangle is a simplex in two-dimensional space that is a plane; a tetrahedron is a simplex in the three-dimensional space; and in n-dimensional space a polytope formed with n plus 1 vertices is a simplex.

Now Nelder and Mead's method iterates over simplexes that are non-degenerate. To begin with we must give it a simplex which is non-degenerate that is all the n plus vertices do not fall in a single hyper plane that kind of a simplex we have to give in the beginning. And then the methods step the typical iterative step of the simplex method we are ensured that at one step one vertex of the simplex method will be replaced by a new vertex and like that the simplex will keep on changing iteration by iteration travelling

towards a minimum point of the function. Now, framing initial n plus 1 vertices which form a non-degenerate simplex is actually not very difficult.

(Refer Slide Time: 05:53)



For example, if you take one point in n-dimensional space then finding additional n points in order to form a non-degenerate simplex is easy because from this point if you move towards x 1 direction by little amount x 2 direction by a little amount and so on. So, like that n directions among the coordinate directions itself you will get which will give you additional n points and this original point n plus 1 points, so that will be a simplex which will be non-degenerate.

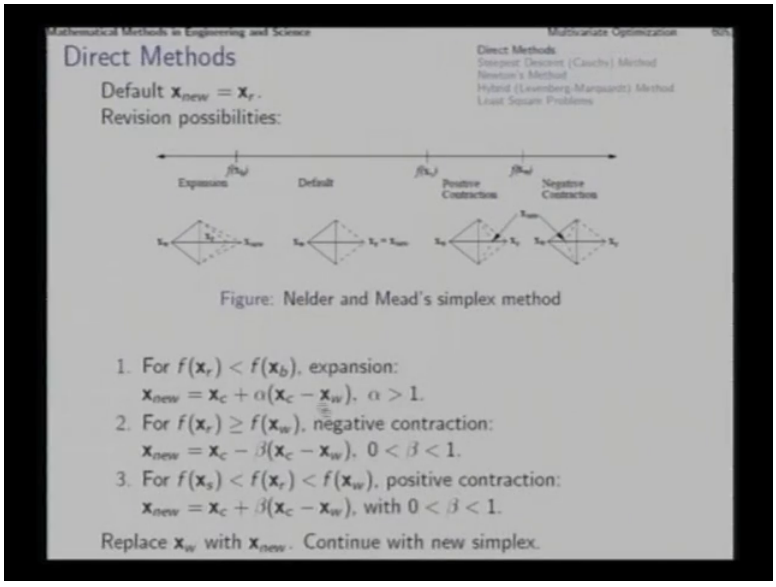
So, in 2D plane the corresponding thing is that corresponding situation is that whatever point you take from that move in the x direction a little and y direction a little and then you get two further points and this is a valid triangle, no chance of all the three points falling on the same line. So, like that developing n plus 1 points which form a non-degenerate simplex is actually easy, you develop such a simplex and start the iteration.

Now, in the typical iteration, beforehand we evaluate the function at these n plus points and after developing the function values after evaluating the function at these points we identify three of the n plus 1 vertices. The point x w the vertex x w which is the worst point, where the function value is the worst that is highest for a minimization problem. The best point x b, where among these n plus 1 vertices the function value is lowest. And the second worst point x s.

Now, in one iteration of the simplex method, this worst point x w will be replaced with a good point how to do that. So, we try to find out the centre of gravity of the face not containing x w. So, in this simplex of n plus 1 vertices every collection of n vertices define a face. Now, that face which contains all the vertices except the worst point the centre of gravity of that face is found by simply adding the position vectors of their vertices its vertices and dividing by n out of the n plus 1 vertices n vertices are included here, that is excepting the worst point. So, therefore, divide by n. So, this x c is the centre of gravity of that face, which does not contain the worst point.

Now, for example, suppose this tetrahedron is the simplex and this is the worst point. Now, the centre of gravity of the face not containing the worst point will be the centre gravity of this triangle. So, suppose this is here. Now, from x w to x c, if we draw a line and extend it further behind then this is a point, which is the reflected point, reflection not by this plane, but against this point. Why we do this, because among the vertices of the simplex this is the worst point then from this plane from this hyper plane this side is a bad side. So, we try to go on the other side. So, we come from x w to x c and further go behind by equal amount and that is the reflected point x r, so that is x r. Now, this x r is the default replacement for x w. Now, other than this default replacement we can consider several other options.

(Refer Slide Time: 09:42)



See default option is here where this reflected point this is x w, this is reflected point x r and this line segment this line shows the face not containing the worst point x w. So, this is x r which is a default new point to be used for replacing x w in the simplex. There can be some other possibilities. For example, if it happens that we find that the function value at x r, f of x r f at x r turns out to be better compared to even the best point that we have right now. Among the current vertices of the simplex whatever is the best point compared to that also if the reflected point is better that will mean that it is a very good direction to go forward.

So, in that case we may decide not to stop at this point itself, but to go further so that means, we will expand the simplex not keeping it of the same size, but we will expand the simplex and go here, this is the (Refer Time: 10:51). So, if the function value at x r turns out to be lower than function value at x best, then we consider an expansion of the simplex. On the other extreme, if the function value at x r turns out to be worse than the current worst point; that means, that staying on this side of the plane itself is better because on that side it is even worse. So, then we consider a negative contraction, contraction on the old side itself not on the new side at all.

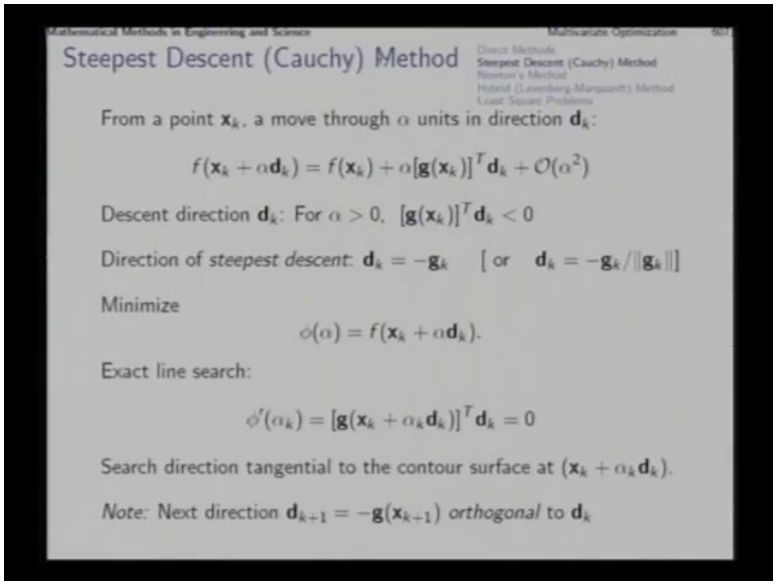
On the other hand, if we find that the function value is between x s and x w that is it is better than the worst point, but worse than the current second worst point even that means, that it is not a great idea to go all the way to x r. Because the moment we accept this and frame the new simplex this new comer will be ready for expansion, because after the x w point is replaced with this then this will become the worst point because it worse than the second worst point. So, that is why going x r may not be a good idea though this direction is good. So, in that case we consider a positive contraction that is here.

So, all these special measures expansion, contraction of either kind can be affected, so this kind of measures, that if the reflected point is better than the current best then we expand that is x c plus alpha into x c minus x w, alpha equal to 1 would mean taking a side itself. So, alpha is greater than 1 that is that will bring us here the simplex will be expanded. If x r is worse than the current worst point that is here then negative contraction that is x c minus beta into x c minus x w. So, beta is between 0 and 1, so that will give us this kind of a point in between that way the simplex will get reduced in size.

In this case, where x r is better than the worst, but worst than the second worst that is in between the two worst points currently then we consider a positive contraction, this in place of this minus sign, we have a plus sign the rest of it is same. So, the simplex is nevertheless reduced in size, but it is brought to this side. And if the function value at x r the reflected point turns out to be worse than the current best point, but better then the two worst points at present, then we take the default value default point which is the reflected point itself. This is a typical iteration. And as this situation goes on then finding good directions the simplex is expanded in order to explore the such space and when good directions do not come forthcoming then the simplex is reduced to such negative contractions and positive contractions, and slowly the size of the simplex goes down and that way squeezing the minimum point.

So, finally, the termination condition part term aside is when the vertices of the simplex come extremely close to each other approaching that tolerance or accuracy required by the problem. So, this is one very good method which uses only function values and it operates remarkably well for most of the problems. However, if variant is available then that is if the function is differentiable, and the derivatives can be developed without too much of computational cost then most of the time we use derivative based methods because they are relatively faster more efficient.

(Refer Slide Time: 14:43)



So, the most straight forward method conceptually simplest idea is that of steepest descent method or Cauchy's method. This is typically a line searched based method in which from a point x k initially x 0 the guest point given and in intermediate steps the current point the current iterate. So, from a current point x k, a move through alpha units in a direction d k results in this kind of a situation f of x k plus alpha into d k which will be up to first order approximation will be alpha into variant transpose d k. Now, this is a first order up to first. So, if you omit the high order terms then you find that this is the change in the function value.

Now, if along the direction d k, if the function value decreases at least in the local neighborhood, then you call that direction d k as a descent direction that is for positive alpha if this is negative then a direction is called a descent direction. It is along the direction an infinite decimal step will tend to decrease the function. And since our problem is that of minimization typically we would like to operate along a descent direction.

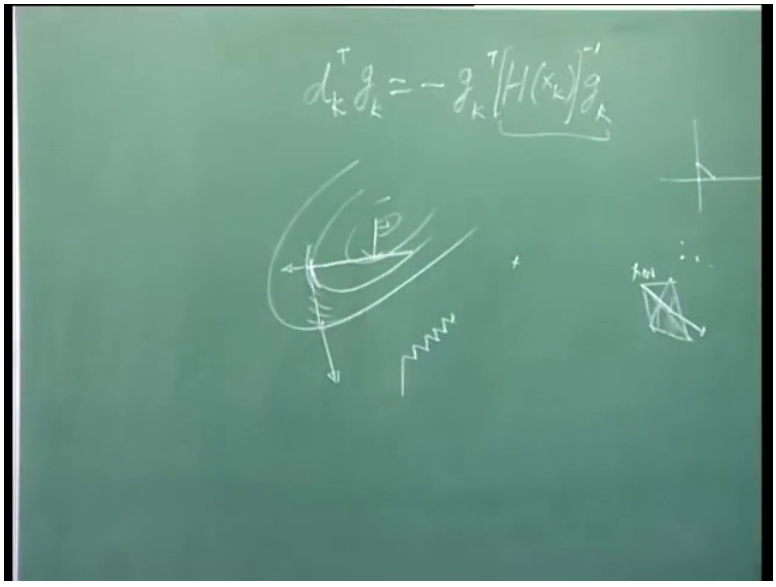
Now, if we are going to operate on a descent direction, then y naught pick up that direction along which the descent is fastest or steepest descent and that will be in the direction of negative gradient, because gradient of a function is gives you that direction along which the function increases fastest. So, its negative direction will give you a direction along which the function will decrease fastest. So, if you take that direction which is the direction of steepest descent that is fastest decrease that is a negative gradient. You can take minus g k negative of the gradient factor or the unit factor along that direction it does not matter. So, after selecting that if you select that direction then that corresponding method is called the method of steepest descent or Cauchy's method.

Then you say that after deciding the direction we decide we try to pose the problem to minimize the function along that direction that is how far to go in that direction first we have decided which way to go and that choice of that direction like this has characterized the method of steepest descent. And then along that direction we want to now decide how far to go that is the line search sub problem. So, if you want to conduct a line search along that direction then we say that how far what is alpha how far to go, so that along that line the function is minimized.

So, after decision of the direction d k has been made, this problem is a single variable problem because we are trying to find out how far what is the distance alpha that we have to move. So, f of x k plus alpha k we want to minimize with respect to alpha. So, this function pi of alpha is a single variable function. So, if we try to exactly minimize the function along this line then the process is called exact line search. On the other hand, sometimes we conduct an inexact line sets that is decreased sufficiently and then from there we try to work out a new direction that is also in practice. In fact, more professional algorithms use inexact line set, but for a time being to keep things simple we talk of exact line sets only.

So, for exact line sets, we will terminate at that point where along that line the function while reducing, reducing, reducing, stops reducing and then starts increasing again. That means, we are looking for that point where the function stops changing along this direction that is phi prime is 0 at which value of alpha the phi, phi prime the derivative of phi with respect to be alpha becomes 0. If I differentiate this with the help of the chain rule then we find gradient of this function at this point transpose derivative of this with respect to alpha that is d k, so this should be 0. So, we are looking for that point that alpha k where this will be 0.

(Refer Slide Time: 19:20)



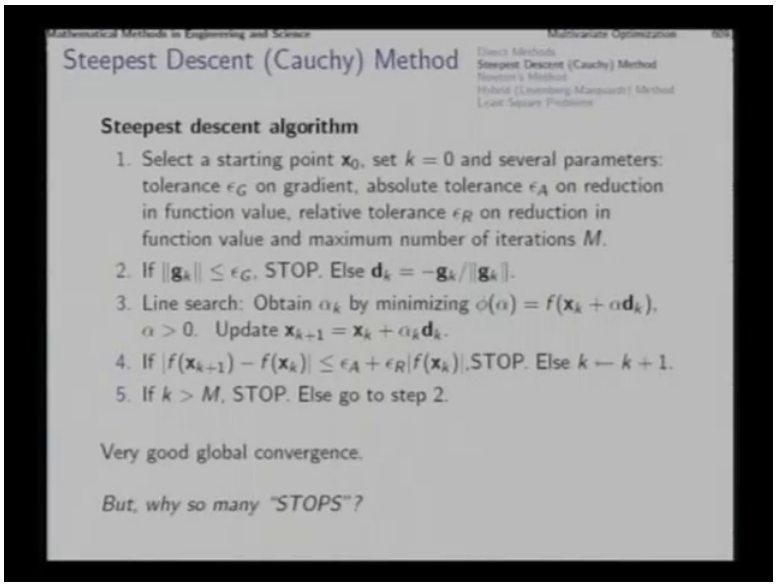
What happens is that if the function contours are like this, here is a minimum point. And if I have started somewhere at this point, then at this point the gradient is this way

orthogonal to the contour and then negative gradient is this direction, and the steepest descent direction is this. So, in the steepest descent method, this is the direction along which the line search is conducted. And as we proceed along this direction on the way contours are cut like this. And finally, we approach a point where a contour is tangential to the direction that is a point where the line sets ends, there is a point where an exact line set would end. So, that is as you cut the contour inward, the function value goes on decreasing, decreasing, decreasing and at the tangential point it does not decrease anymore; beyond that it would you would end up cutting the same contours outward, so you stop here at this point.

So, this is the point where the gradient is in this direction and it is orthogonal to the current search direction. Initially we started searching along the direction, which was exact negative to the gradient, but finally we arrive at a point where the current direction is at right angle to the gradient direction, so that is the end of one iteration. From there the fresh gradient is evaluated from this way and the search in a negative gradient would go like this which would be tangential to the another contour here, and this is the way in which we will proceed. So, this is the method of steepest descent.

Now if you conduct exact line search in any method then you will find that the direction along which the line search is made at the end of the exact line search, the gradient at the final point turns out to be orthogonal to this search direction. In the case of the steepest descent method, its negative will be the next search direction orthogonal to d k. So, this is the way the steepest descent method works.

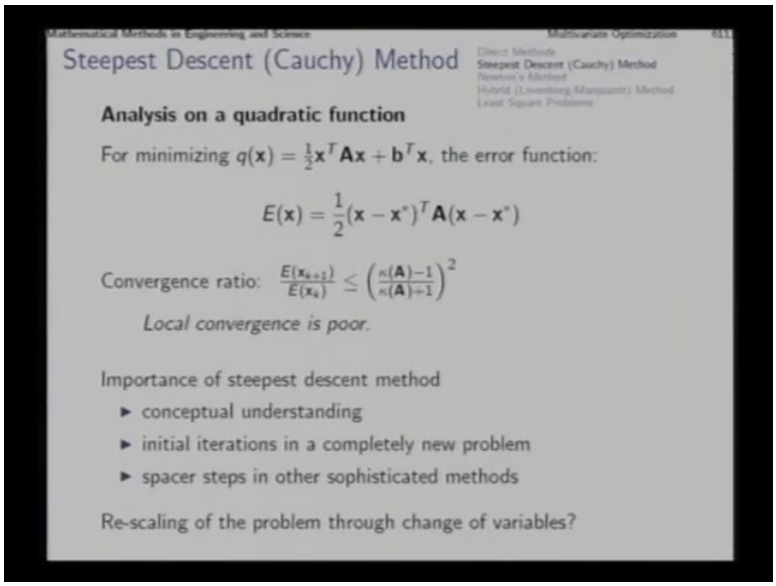
(Refer Slide Time: 21:44)



And if you try to work out an algorithm out of it then this is how it will look like selecting a starting point x 0 and several termination parameters tolerance values etcetera and maximum number of iterations. The termination condition for the steepest descent algorithm will be the vanishing of the gradient. If a gradient at a point is found to have very small magnitude that is almost 0, then we stop else we evaluate the direction. And then in that direction conduct a line search by minimizing this and accordingly update the point x that is x k plus alpha k d k, the result of line search that gives the next point.

And then we can check whether there has been significant change in the function value in terms of absolute tolerance and relative tolerance. If not if not much change has taken place then we can stop; otherwise if the number of iteration exceeds, then also we can stop which means that we are losing patience, we do not expect to have further improvement. And otherwise if the number of iterations is reasonable not approaching the maximum allowed number, then we go to the step two again evaluate the check this gradient condition and continue into finding the next direction and so on. So, this is the typical go. Now one good quality one great merit of this method is that it has excellent global convergence that is started anywhere at every step, it is assured that will make significant, it will make a descent step and it will approach the minimum point. But why we put so many stops because in spite of having excellent global convergence the method of steepest descent has a very poor local convergence.

(Refer Slide Time: 23:47)



The reason for that can be analyzed; if you consider the benchmark problem of minimizing a quadratic function like this. Now, minimizing this function and this error function is actually equivalent, because if x star is the minimum point of this function then it is a minimum point of this function also, and the difference between these two functions is actually a constant. Now, therefore while analyzing the steepest descent method on a quadric function, we typically analyze it over this function. And through a long direction, you can prove that the ratio the convergence ratio first of all it has linear convergence rate, and then the convergence ratio of that linear convergence process that is error at the next point divided by a error at the previous point is limited by this number.

And this number is this number can be very large depending upon what is the condition number of A. For example, A is a matrix - the Hessian matrix. And if you find that the largest Eigen value by least Eigen value of that Hessian matrix is something like 9 then you will find that this will be 8 and this will be 10, 8 by 10. And the square of that will be 64 by 100 that means, 64 percent which will mean that 64 percent of the error at the previous step is likely to remain in the next step, so that shows that the convergence is quite slow. And this is why in badly scaled problems in which the Hessian is badly conditioned the condition number of Hessian is large in such situations you will find that the convergence ratio is poor and the algorithm does not operate quite well.

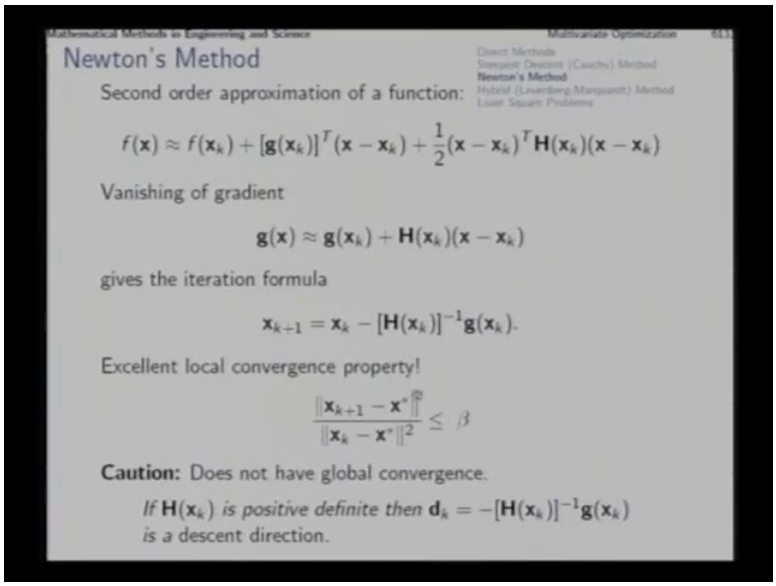
However, steepest descent method has its own advantages. One great advantage is that its conceptual understanding is direct, conceptually that is the simplest method that one could think of. Second is that in a completely new problem, it is advantageous to start the process with a steepest descent based method, because that has excellent global convergence. And this global convergence property also helps in the utility of steepest descent steps into other professional algorithms which then generate directions based on more sophisticated considerations.

The more sophisticated considerations in more professionally sensible methods like conjugate direction method or Quasi-Newton methods may develop directions which most of the time operate better, but there are situations where even those directions turn out to be weak or poor. In such situations typically one-step of steepest descent method intersperse between steps of other method helps in regenerating the progress in the process of improvement of the function in a good manner, so that is considered spacer steps. Spacer steps in other sophisticated methods are quite often used with the help of developed with the help of steepest descent method.

Now, the concept of selecting direction and conducting a line search in that direction in this manner based on gradient is inherited also in the more sophisticated method of conjugate directions or conjugate gradient method. But what conjugate gradient method does over and above steepest descent method is at the first step it takes along the negative gradient, and then subsequent steps it takes in such a well orchestrated manner not necessary in the negative gradient direction that the work done in the previous steps are taken advantage of in the later steps.

For example, this kind of in the narrow contour case, the steepest descent method quite often does a zigzag motion like this for approaching to a minimum point like this, that kind of demerit is remedied in conjugate gradient method which is based on conjugate directions. So, that is a little more advance method, which we will not be discussing in this course though it is there in the book in chapter 23, but we will be omitting that chapter in our study here.

(Refer Slide Time: 28:52)



Now, other than steepest descent or Cauchy's method there is one more method which is called a basic method and that is Newton's method that relies on a second order approximation of the function based on a truncated Taylor series. So, for a function f x at x k in the neigh immediate neighborhood of x k the current iterate, the second order truncated Taylor series looks like this. This is the value of the function at the current point plus first order change plus second order change with the higher order changes neglected. Now, for the minimum point in the neighborhood, if we try to consider the condition for vanishing gradient that is the first order necessary condition, then differentiating this we will get this relationship or the first order truncated Taylor series called the gradient itself which will be g x roughly equal to g x k plus Hessian into delta x.

Now, we say that in the neighborhood of x k we try to look for that point, where the gradient vanishes that means, this is 0. If this is zero then we can find out x minus x k which will be negative of this pre-multiplied with the Hessian inverse that is this. So, that is x minus x k. So, for finding x we have to add x k to that and that is this. So, this is the typical iteration which is very much like the equation solving process because this is also essentially equation solving process the equation to be solved is gradient equal to 0. So, this is the typical Newton's iteration formula for minimization. The great merit of this method is that it has got excellent local convergence that is local convergence is quadratic. See the error in the next step divided by error in the previous step into error

individual steps squared that is finite that means at every step you will be expecting two orders of decrease in the error value.

However, the point to caution is that this method does not have global convergence, something, which is a bare minimum necessary for any optimization method that it must have global convergence. The idea of global convergence is that at every step the function should decrease or there should be an approach towards the minimum point. So, Newton's method does not guarantee that. However, if started sufficiently close it has excellent local convergence in the sense that it approaches the solution at a faster rate, if it does at all. On the other hand, if stared far away from the optimal point, it may not approach the optimum point at all it may go somewhere far away. Because there is no guarantee as such that an x k plus 1 generated through this formula will be a point where the function value is lower than the function value at x k. That means, it does not have the property of global convergence.

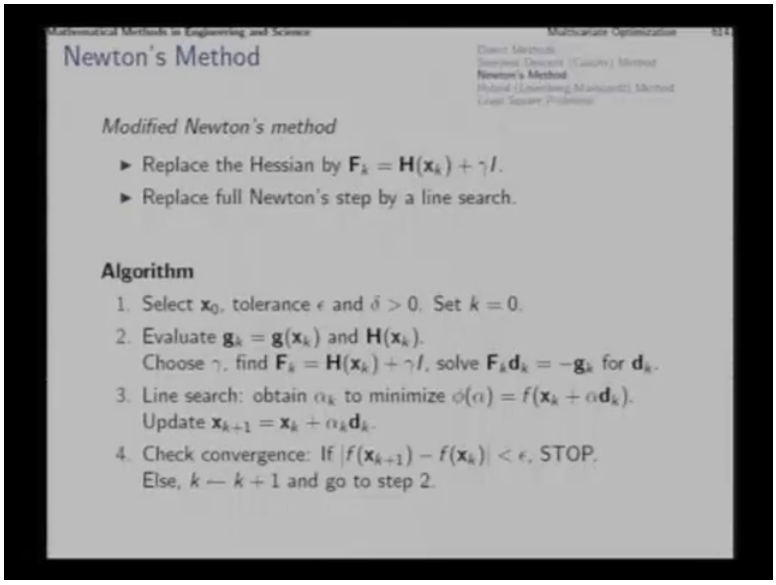
In the special case, where the Hessian matrix is positive definite in that kind of a situation also all that we can say is that direction suggested by Newton's method is a descent direction. That we can say because if H x k there is a Hessian matrix as the current point is positive definite then x inverse is also positive definite. And in that case d k with this formula will give you d k transpose g k as minus g k transpose H x k g k, it is minus this is the direction d k with that you take the inner product of g k that is g k transpose added here put here multiplied here. So, you get this

So, if the Hessian is positive definite then this is positive for all g k and that means, with this negative sign this is negative and that means, the direction suggested by Newton's method that is minus H into g k is a descent direction. Even that does not mean that the entire complete step of Newton's method will be a descent step. Because if the direction maybe a descent direction along the direction the function value might start decreasing, but in the entire complete step in between it might start increasing again. So, it may be a descent direction only if the Hessian matrix this is inverse.

So, there are two points here one is that if the Hessian is not positive definite then it may not be a descent step the function value may increase and it may not be even a descent direction. If the Hessian matrix is positive definite then it is guaranteed that the direction will be a descent direction though nothing can be said about the complete step. So, based

on this observation, we may think of the modification needed in the Newton's method for its developing into a worthwhile optimization method having global convergence property has then two aspects. One is the necessity of this Hessian matrix being positive definite which as it is cannot be guaranteed at any point because we cannot guarantee the positive definiteness of the Hessian matrix at any point. And the second is not to take the complete step suggested by the Newton's method though we may like to accept the direction, in case this is positive definite. So, these two aspects are addressed in what is called the modified Newton's method.

(Refer Slide Time: 35:12)



In modified Newton's method we replace the Hessian by Hessian plus gamma into identity such that this resulting matrix is positive definite. And that makes sense because by adding gamma into identity we are basically enriching the diagonal entries of the Hessian matrix that is trying to make it diagonally dominant which will ensure that the matrix is positive definite. And the second measure that we take is that from the Newton's method we take only the direction that is direction is minus f inverse g, but then we do not take the full Newton's step, but between 0 and 1 we conduct a line search. So, we replace the full Newton's step by a line search.

So, by ensuring the positive definiteness of the effective Hessian we ensured that the direction suggested by Newton step is at descent direction. And then rather than taking the full step the descent of which is not guaranteed, we conduct a line search along that

descent direction and the line search process is bound to terminate at a point through a descent step. So, with these two modifications what we get is modified Newton's method in which the algorithm will proceed like this. After selecting the point x 0, we evaluate the gradient and Hessian and choose gamma in order to make it positive definite. If it is already positive definite in that case gamma can be chosen as 0; otherwise we select an appropriate gamma to make it positive definite, and then in place of Hessian we use this F, F k and then solve this to get a direction not a full step. In a pure Newton's method, that would be taken as the full step, but in the modified Newton's method from here we take only the direction and then along that direction we conduct a line search as usual and then update and go for the next evaluation.

The typical termination condition is this that is if no function improvement has taken place in the previous iteration then we stop. So, this is modified Newton's method which addresses the two most important objections of Newton's method. Yet one disadvantage of Newton's method remains that is the task of evaluating the Hessian which may be costly, because Hessian will require n square second derivatives. Evaluating a second derivative is costly and evaluating n square of them at least n square by 2 you can say because half of them you may not have to evaluate. So, even half of them evaluating such a large matrix of second derivatives is going to be computationally costly.

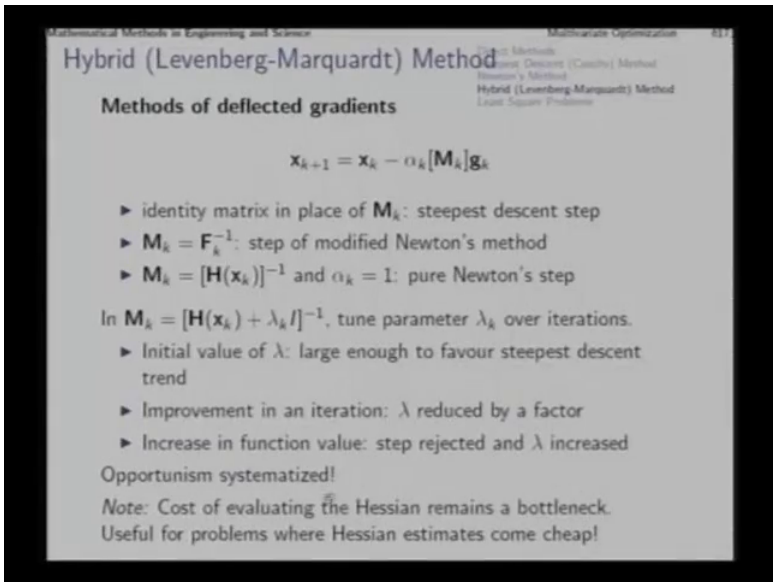
Now, how to handle this problem this problem is handled in two different ways there is a family of methods called Quasi-Newton methods that is Newton like methods that these are some quite sophisticated methods with a deep theory behind them, which we will be omitting in this course. But the theme of Quasi-Newton methods is the development of a Hessian through steps that is if we evaluate only gradients, and take steps accordingly then the step that we took along that step what was the change in the gradient. So, change in the gradient through a step that gives us a little bit of information about the Hessian. Why, because Hessian into the step Hessian into delta x is suppose to be delta g gradient k, change in gradient should be Hessian into change in x.

So, through every step we generate one bit of information regarding the Hessian and through updates over iterations if we try to construct the Hessian or rather the inverse Hessian to be used while solving this then such methods are called Quasi-Newton methods they try to get most of the advantages of Newton's method. But they do not work with explicit and actual Hessian all the time they try to develop the approximate

estimate of the Hessian on the way through iterations that is the family of Quasi-Newton methods. As it is another kind of situation may arise in which we may use Newton based method and that is in those problems where Hessian is cheaply available.

Not only if the second derivative expressions are easy and cheap in calculation, but also situations where a good Hessian estimate can be developed based on first derivatives only. Such situation arise in problems where you have a least square minimization kind of problem or equation solving kind of problem. And one such problem, one such method which utilizes that fact is Levenberg-Marquardt method which has a few other interesting features also.

(Refer Slide Time: 40:29)



To see those interesting features consider this typical iteration formula which is called the method of deflected gradients. So, in this single formula actually a large number of methods are embedded consider this formula in which x k minus alpha k M k g k is the new point. Now, in place of M k if we put identity matrix and alpha k is determined by line search then we get what is the steepest descent step. On the other side, in place of M k, if we put f k inverse and determine alpha k by line search, we get modified Newton's method, which we discussed just now. In place of M k, if we put actual Hessian inverse and alpha k we put as one then we get the pure Newton's method. So, all these methods are actually embedded in this formula that tells us that all the three methods that we

considered till now steepest descent, Newton and modified Newton all these are somehow related to each other.

And therefore, it should not be impossible to move from one method to another through some small adjustments and that maybe of great significance, because in this family we have one method which is steepest descent method which is very good in global convergence and very poor in local convergence. On the other extreme, we have Newton's method which is very good in local convergence, but very poor in global convergence what about combining both of them through a formula of this kind this is what is done in a hybrid method called Levenberg-Marquardt method.

How? We consider M k to be Hessian plus lambda into identity inverse. Now, we note that if lambda is kept very large then with respect to lambda k I with respect to lambda I the Hessian will turn out to be insignificant and then it will approach the steepest descent step. On the other hand, if lambda is kept extremely small then lambda k I will be insignificant compared to the actual Hessian and it will approach the pure Newton step. And then we notice that we can tune this parameter lambda over iterations in order to favor a step which is Newton like or a step which is steepest descent like or Quasi like.

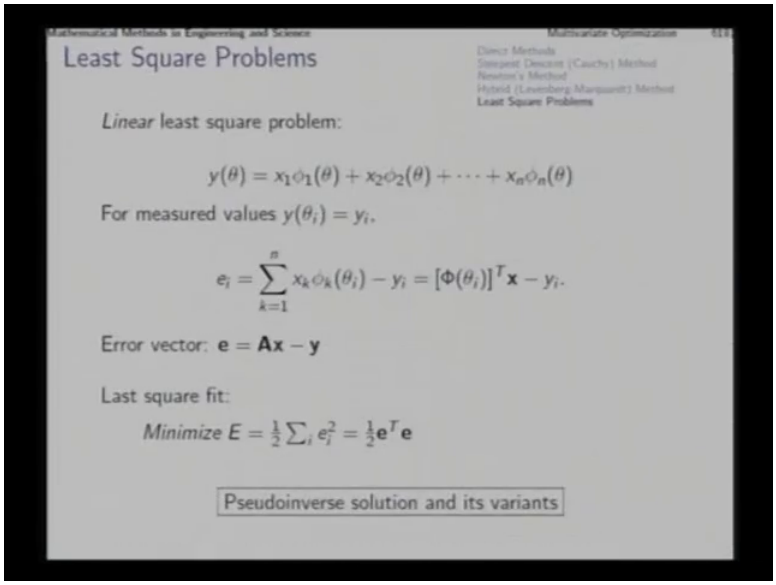
So, since the initial iterations should be more on the steepest descent side, so initially we keep a large value of lambda and take some initial steps. And after every step if we find that there has been a improvement in the function value then we decrease the value of lambda. So, improvement in an iteration will lead to a reduction of lambda by a factor. On the other hand, if we find that the function value tends to increase in a step then we reject that step we do not move the point and we increase the lambda. That means, whenever we find that improvements are being made good improvements are taking place that means, we are approaching the solution, we are going close to the solution where Newton's method is likely to perform better. So, we reduce lambda in order to favor a Newton like step.

On the other hand, the moment we find that lambda has been decreased too much that is it has become too small and the Newton's method is not going to give a good next point then we reject that step and increase lambda in order to go into the relative safety of the Cauchy step or steepest descent step. So, this opportunism gives us a method which is the Levenberg-Marquardt method where this tuning parameter lambda is adjusted

iteration over iteration and we take advantage of the global convergence of steepest descent method and the local convergence of Newton's method.

Now a particular way of implementing Levenberg-Marquardt method is found to be highly successful in non-linear least square problems and equation solving problems in which a cheap estimate computationally cheap estimate of the Hessian can be developed based on first derivatives only. And that removes the last bottleneck of evaluating the Hessian that is for that kind of problems.

(Refer Slide Time: 45:12)

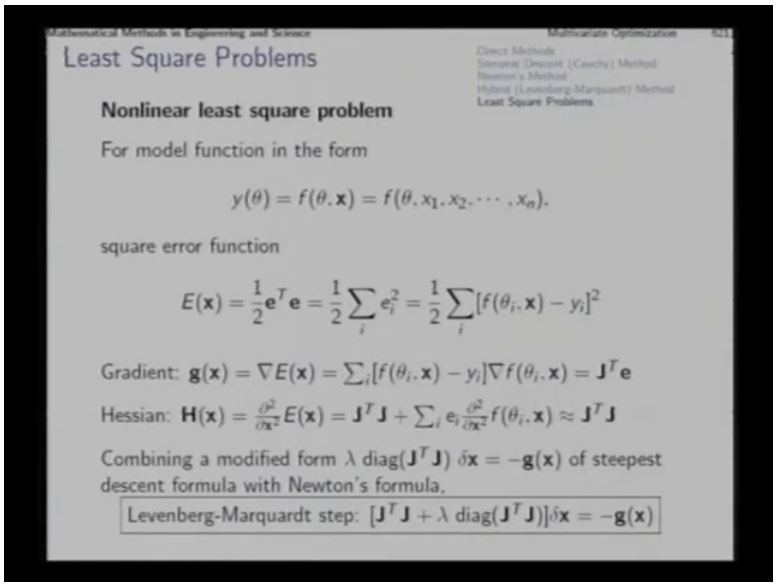


So, suppose we try to see what kind of a least square problem, what least square problem look like. A linear least square problem is like this, in which we trying to model a function y of theta which is available in this manner phi 1, phi 2 etcetera are known functions of theta and x 1, x 2. x 3 etcetera are the unknown coefficients which we want to determine. Now, for that for a lot of measured values of y against theta we try to find out the values of x 1, x 2 which will make the error minimum in the least square cells. That is error in the measured data we take and square the errors and consider the sum of those squared errors and minimize that sum.

If we try to do that then error is this expression minus y measured y. Then this xs are the unknowns. And when we try to find out the minimum value of the sum of error squares then we get a problem which we actually solved earlier in chapter 7 and chapter 14 in earlier linear algebra lectures, where the least square problem was found to be the pseudo

inverse solution of this A x minus y equal to 0. So, that is the pseudo inverse solution that we have already seen. This is a linear least square problem. Now, if the unknown coefficients unknown parameter x 1, x 2 do not appear in a linear fashion like this, but in a general non-linear fashion.

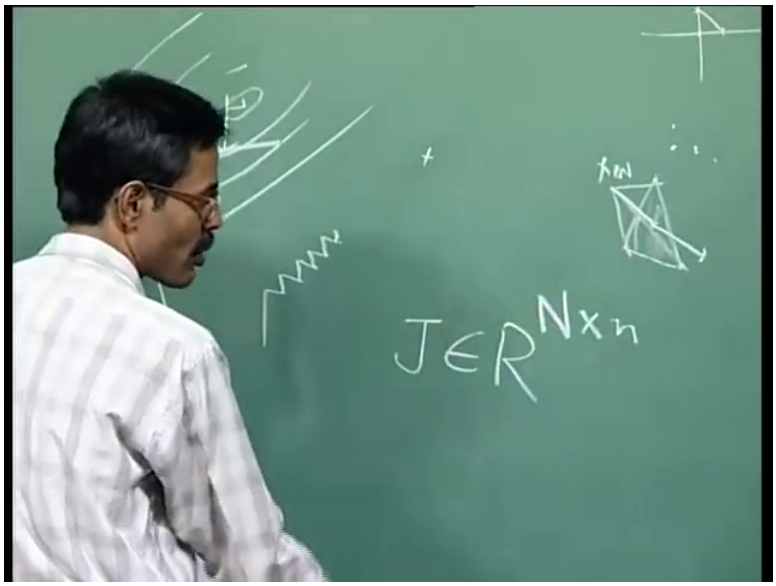
(Refer Slide Time: 46:57)



Then the typical symbolic representation will be like this y of f and theta y of theta is f of theta and x in which the unknown parameters x 1 and x 2 can appear in any manner. The square error function we can still define in the same manner and that will be this theta i x y i are measured values for a large number of data points. We want that value of x for which this least square error is least square error is least square error is minimized, this is why it is called a non-linear least square problem. Non-linear because x 1, x 2, x 3, x 4 affect the function in a non-linear manner not in the linear sense.

Now, if we try to find out the derivatives of this then we find that the gradient of this function turns out to be half remains outside; from this sum we get twice which will cancel this half, this stuff which is e; and then that then the derivative of this with respect to x that is variant of f. So, this is the error into gradient of f. Now, gradient of f at every data point will fill up rows and rows and rows and we will get the complete Jacobian. So, J transpose e will be the gradient of this function.

(Refer Slide Time: 48:37)



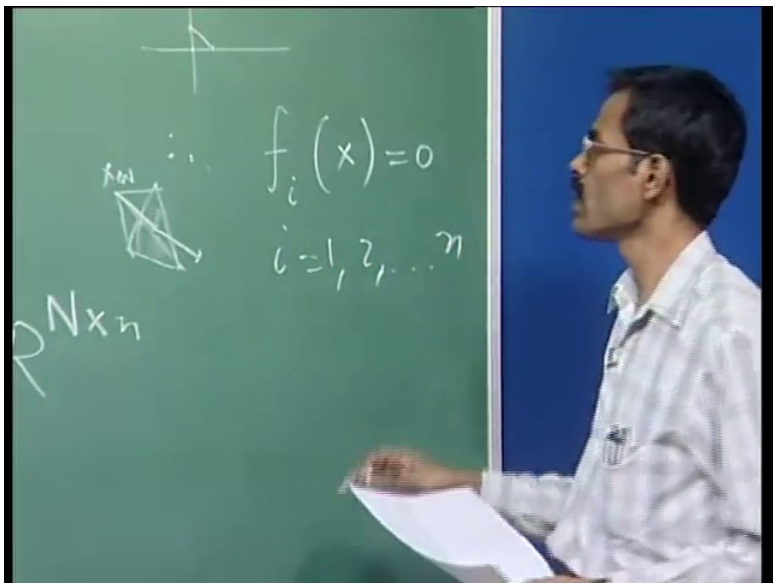
Where J is the N by n matrix n is larger where capital N is the number of data points and small n is a number of parameters x 1, x 2, x 3 up to x ns that we want to determine. So, J transpose e turns out to be the gradient of this error function and the Hessian of that when we try to evaluate then we will have two parts in the Hessian that is Hessian is the derivative of this. So, in the two parts in one of the part, this will be differentiated keeping this as constant that will give us J transpose J. And in the other part of the Hessian, this will be kept constant and this will be differentiated that is the actually second order terms error into the second order terms that will be sum of all these.

Now the important issue concept that is in this completes Hessian expression, this term will have a very good reason to be small in magnitude. Why, because the second derivatives are multiplied with errors which are going to become small as the convergence process proceeds, as the convergence process progresses, so that is why neglecting this part which involves the computation of second order derivatives. We can make an estimate of the Hessian based on this only J T J. And J - Jacobian is evaluated based on first derivative as on. So, with the help of first derivative as on sitting in the matrix J, we work out a Hessian estimate which is quite accurate at least in the later iterations.

So, a respectable estimate of the Hessian is evaluated based on calculations of first derivatives as on the calculations of this second derivative part we will omit. With this

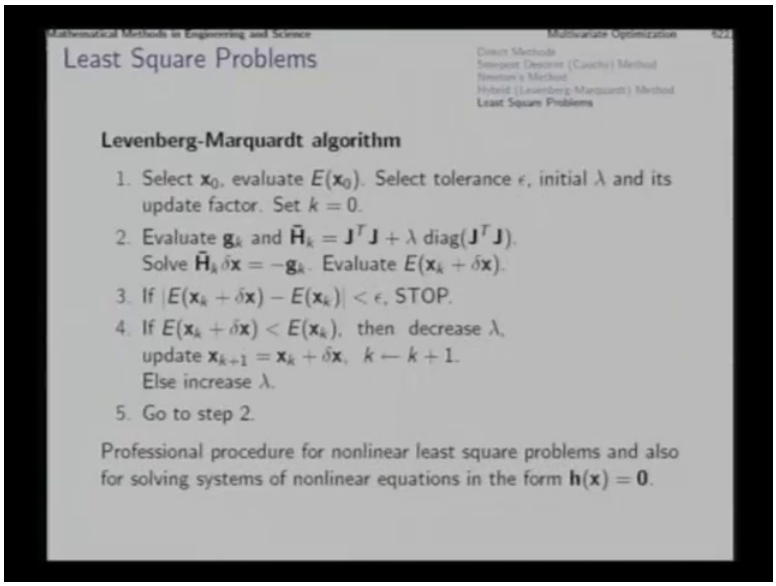
Hessian estimate, we combine a modified form of the steepest descent and get the typical Levenberg-Marquardt step, which goes like this. So, this part is the representative of the Newton's step of Hessian matrix and this part is actually a reformulated or a modified form of steepest descent consideration. So, based on the combination of the two, we try to work out the step delta x for a particular iteration. And this tuning parameter lambda we keep on tuning iteration by iteration in order to favor the Newton step or the steepest descent me step as the situation demands that is whether enough progress is being made or progress is not being made. So, this is the typical Levenberg-Marquardt step used for non-linear least square problems. And the same can be used when we have an equation solving problem.

(Refer Slide Time: 51:46)



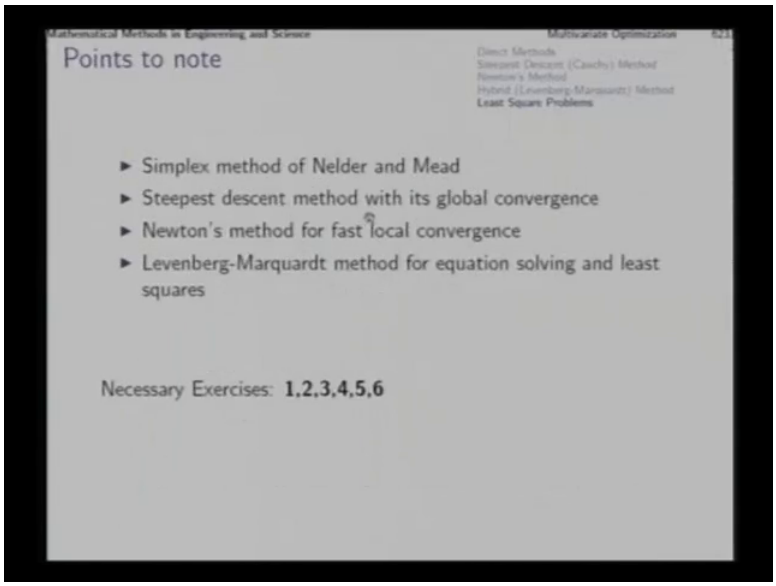
That is if we have a large number of equations to solve like this, then we formulate the problem as f 1 square plus f 2 square plus f 3 square as the function to be minimized. So, that also actually boils down to the minimization of the sum of f u squares right in the same manner. So, the solution of a non-linear system of equations also can be framed in this same form and the same method can be utilized. So, Levenberg-Marquardt method is found to be very useful in the solution of non-linear least square problems and non-linear equation solving problems. Though in ordinary optimization problems, it has a disadvantage that the Hessian calculation is costly. In the equation solving and least square problems, it has the advantage that a good Hessian estimate can be computed based on the first derivative first order derivative as on.

(Refer Slide Time: 52:54)



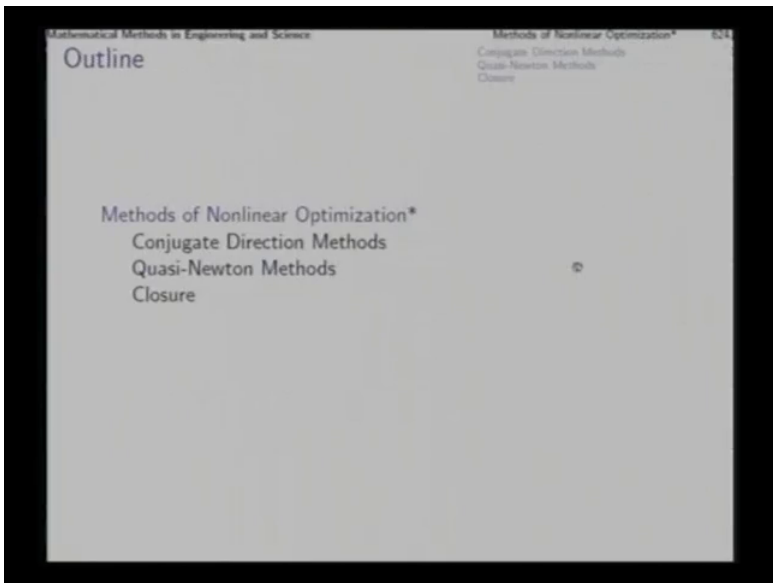
So, the algorithmic steps of the Levenberg-Marquardt algorithm is given here. So, starting from an initial point we evaluate the error, select tolerance, initial lambda quite high and the update factor which depends on their choice. And then with the gradient and Hessian estimate based on this we workout delta x error matrix. If the convergence has taken place then we stop; otherwise if the step offers an advantage then decrease lambda if the and update; if the step offers a not an advantage, but it leads to a disadvantage then we do not update and increase lambda and continue. So, this is a typical Levenberg-Marquardt algorithm. So, professional procedures professional implementations or sub routines for non-linear least square problems and also for systems for systems of non-linear equations in the form typically use this kind of a method.

(Refer Slide Time: 53:56)



So, the important points to note from this particular lesson is the direct methods one of which we discussed Nelder and mead simplex method steepest descent method which is global convergence. Newton's method for fast local convergence and also for the risks of Newton method which you need to safeguard against. And Levenberg-Marquardt method for equation solving and least square problems.

(Refer Slide Time: 54:26)



The next chapter of the book which has conjugate direction and Quasi-Newton method we will omit. And in the next lecture, we will go to the discussion of constrain optimization problems.

Thank you.