Dynamic Data Assimilation
Prof. S Lakshmivarahan
School of Computer Science
Indian Institute of Technology, Madras

Lecture - 19
Minimization algorithms Continued

So far in the context of minimization algorithm we analyzed almost all the properties of gradient based algorithm in the context of quadratic minimization problem. We also indicated how to adapt the gradient algorithm for non quadratic functions, irrespective whether the function is quadratic are not the fundamental principle is I have an operating point, I have the direction of the gradient. For the case of quadratic functions I have an explicit formula for the step length, if the function is more non-linear than quadratic for non quadratic functions in principle that does not exist a formula for optimal value of alpha k, in that case we can only compute approximate values of alpha k the step length parameter at time k and that was done by fitting a quadratic model to the slice of f of x centered at x k in the direction r k.

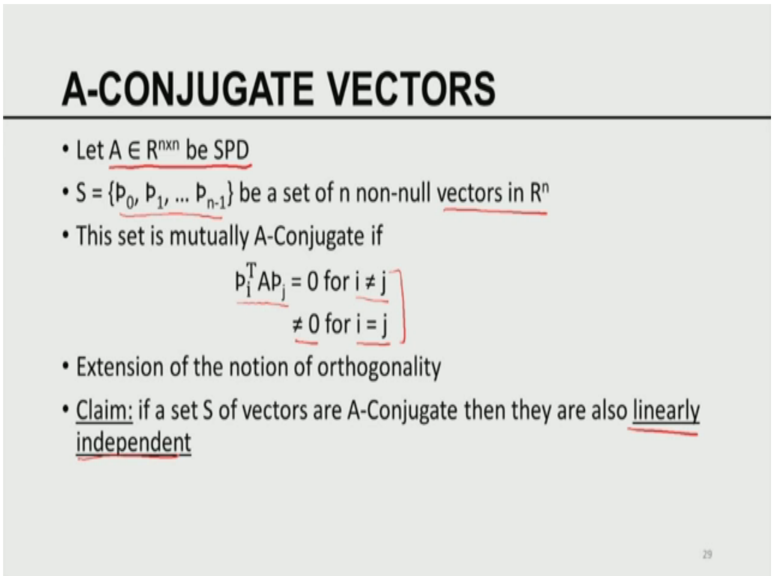
We could further refine that concept by fitting a cubic polynomial or 4th degree polynomial which will be which will give us better and better approximations of the slice of f of x in the direction. So, by first fitting a curve and then we can minimize the fitted curve to fix alpha k. Therefore, all these ideas together cover the general applicability of gradient based algorithm to both quadratic as well as non quadratic. In either case the convergence is asymptotic, oftentimes we may not have the resources needed to wait until convergence more often than not we may want to be able to prematurely cut the iterates, not arbitrary but by measuring how far we are away from the minimum and allowing for certain range of values which are given by epsilon is equal to 10 to the power of minus d, d could be 6, 10, 15 depending on whatever we saw.

So, the best we can have with respect to gradient algorithm is asymptotic convergence that promotes the notion of being able to at least theoretically examine the presence of ideas that can guarantee finite time convergence and that gives raise to the notion of what is called conjugate direction methods your specific class of conjugate direction method is called conjugate gradient method. So, our next topic is to be able to explore the power of conjugacy to be able to produce theoretically a basis for algorithms that can guarantee finite time convergence for quadratic functions.

Please understand quadratic function is a, quadratic functions are essentially model problems in every area we have a notion of a model problem. For example, a differential equation the harmonic oscillator understanding the basic principles of design of harmonic oscillator and analyzing the properties is a model problem in a first course in differential equation. Likewise in optimization theory the model problem is always minimizing a convex function given by positive definite quadratic functions, are positive different quadratic forms. So, if you can guarantee the performance of these algorithms on these model problems then you know you have something to hold on to you have something to, hold on to so that is what the that is, what the basis for concentrating on quadratic functions.

So, we are going to be looking at the derivation of conjugate, the basic principles of conjugate direction methods and the basic principles of conjugate gradient method as our next topic.

(Refer Slide Time: 04:34)



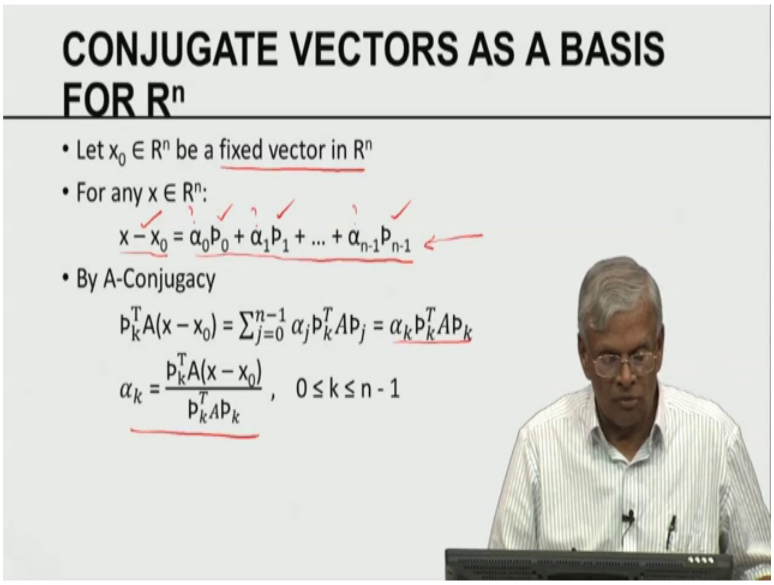
Please recall from our earlier discussion on finite dimension vector space we have already indicated that given a matrix A which is SPD, given a set of directions p 0 to p n minus 1 each of them are vectors in R n. So, I have n vectors please remember instead of labeling from 1 through n we have labels from 0 through n minus 1 nothing is lost, it is one of the standard ways the literature does.

We say a given set of n directions are said to be A conjugate if p i transpose a p j is equal to 0 for i not equal to j is not equal to 0 when i is equal to j. So, that is the fundamental principle of conjugacy. If A is equal to identity matrix conjugacy reduces to orthogonality, so conjugacy is the generalization of the notion of orthogonality. This is a this is a very simple, but a very cover notion of the extension of orthogonality. We are going to state our first result the first result is as follows. If a set of vectors s namely p 0, p 1, p 2, p n minus 1 if they are linearly independent we know they are form a basis. So, linear independence of vectors is the fundamental property, what is important here is that if you have a set of vector that are known to be A conjugate they are immediately linearly independent as well; that means, a conjugacy implies linear independence.

So, if conjugacy implies linear independence, linear independence means a set of n vectors that are linearly independent can be considered as a basis for n dimensional space therefore, by virtue of this property a conjugacy implies leading the independence if I have a bunch of if I have a set of n conjugate directions those being linearly independent one can build the analysis based on these A conjugate vectors as the basis for the n dimensional space in which we are going to perform the computations. We will soon see that doing arithmetic on doing analysis in this conjugate bases simplifies the overall analysis.

So, by the trick is here, by analyzing the problem instead of the original basis, but doing it on the conjugate bases brings out the underlying structure of the problem to the and further simplifies the development of algorithms for minimization that is the basic thought process.

(Refer Slide Time: 07:56)



So, I am now going to verify some of the claims that we made. So, conjugate vectors as a basis for R of n. Because they are linearly independent, if they are linearly independent we should be able to use them as the basis.

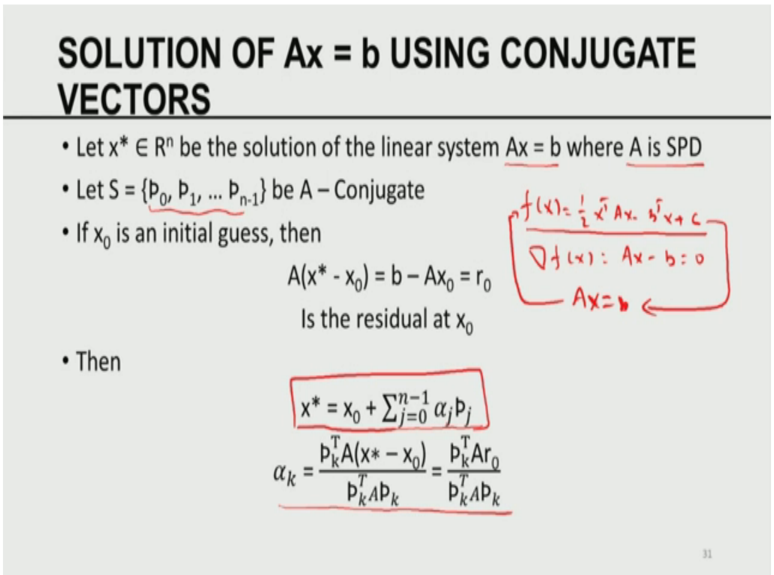
Let x naught be a fixed vector in R n. For any x minus x naught is an arbitrary vector. So, what is the idea here? I pick x naught and consider x naught as the new origin and consider any vector x with respect to x naught. So, x minus x naught is the vector with respect to the origin at x naught. If x naught is 0 x is a vector with respect to the origin itself. So, for any vector x let x minus x naught be expressed as (Refer Time: 09:01) nations of p naught p 1 p n minus 1. We already know if they are linear, I have not proved that they are linearly independent if a conjugacy holds that is a homework problem for you so I am going to build on the result that a conjugacy implies linear independence. If they are linearly independent any arbitrary vector can be expressed as a linear combination of A conjugate vectors and that is what this statement is all about an arbitrary vector is expressed as a linear combination of conjugate vectors.

By a conjugacy I can multiply both sides by a, I can multiply both sides by p k transpose so p k transpose a of x minus x naught is equal to alpha j times p k transpose a p j. But p k p j they are they are bound by a conjugacy therefore, p k transpose a p j is 0 if k is not equal to j, that is j is a free variable, k is fixed therefore, the left hand side is equal to alpha k times p k transpose times A p k. Therefore, alpha k is equal to p k transpose a times x minus x naught divided by p k transpose A p k. So, this applies for any k. So, this gives you the values of R k. So, let us see what is that they have accomplished. We are given p k, so we know p naught, p 1, p 2 up to p k, we are given x naught that is given we are given x minus x naught is given. So, I am trying to express x minus x naught a known vector as a linear combinations of ps the ps are known the only things are not known are alphas.

So, the whole question is this if the ps are linearly independent I should be able to express any arbitrary vectors that linear combination. So, the problem reduces to one of finding alphas, the value of alpha that is readed here is given by this. So, alpha k is given by this ratio every quantity on the right hand side are known. So, alpha k can be computed in principle. So, what does this tell you? Given any arbitrary vector x, x minus x naught being another arbitrary vector I can find the coefficient the linear combinations needed to express this arbitrary vector x minus x naught as a linear combination of this conjugate bases. So, that is the take home message from here.

So, what does this say? Any arbitrary vector can be expressed uniquely as the linear combinations as a linear combination of conjugate vectors, that is the thesis that comes out of this.

(Refer Slide Time: 12:02)



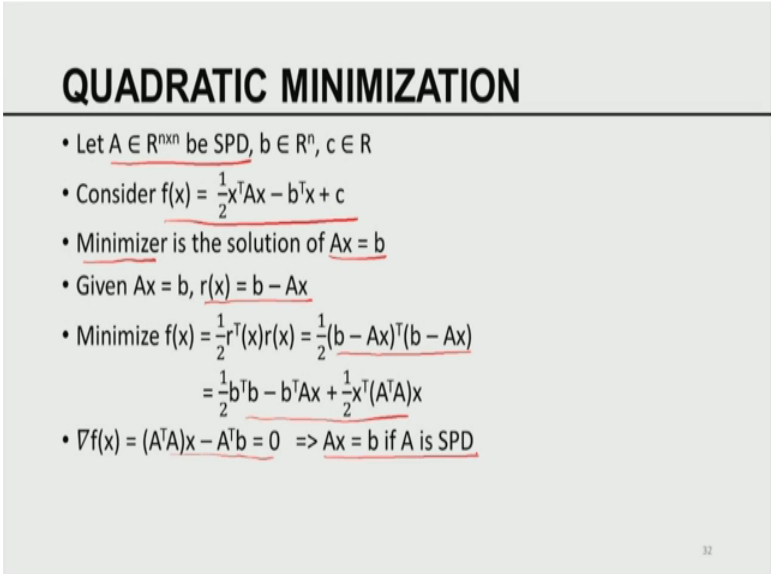
So, with that in mind I am now going to talk about the solution of Ax is equal to b using conjugate vectors. You may wonder I we started minimization now I am considering Ax is equal to b I would like to ask you to recall the following fact. If f of x is equal to 1 half of x transpose Ax minus b transpose x plus c, if I took the gradient of f of x that is equal to Ax minus b and if I sit the gradient to 0 I get Ax is equal to b.

Therefore you can readily see if f of x is a quadratic function at the minimum Ax must be Ax must be equal to b, if Ax is equal to b, Ax is equal to b is the gradient of a quadratic function therefore, minimizing a quadratic function and solving a linear equation are equivalent problems. So, we can pose the conjugate gradient method either as one of solving a linear system or one of solving a minimization of a quadratic form. So, let us assume that we have been given a linear system Ax is equal to b. Let x star be the solution the means x star is equal to A inverse b and I a is symmetric positive definite.

Let p naught to p n minus 1 be the conjugate directions, we have already seen the existence of A conjugate directions. So, let us not be the initial guess for my process of discovering what x star is x star is the solution I am seeking, is also minimize their f of x I want to find. So, Ax star minus A of x star minus x naught is equal to b of Ax star is b so that is equal to b minus Ax naught that is equal to r naught that is equal to initial residual at x naught. Then I can express x naught as at I am sorry x star as, x naught plus the linear combinations of all the conjugate direction vectors that are given.

From the previous analysis we now know the alpha k that enable this expression to be true this expression to be true the alpha ks are given by this expression. Therefore, what does this tell you the minimum x star which is also the solution of Ax is equal to b, can be expressed as x naught plus a linear combinations of the conjugate vector. So, this is the important result, x naught could in principle be 0 or it could be any vector. So, this ability to express the minimum as linear combinations of the conjugate direction method is a very powerful principle that comes out of the analysis that we have already presented that follows from the linear independence of conjugate directions. So, conjugates, a conjugacy, linear independence and the consequence thereof, that is the path.

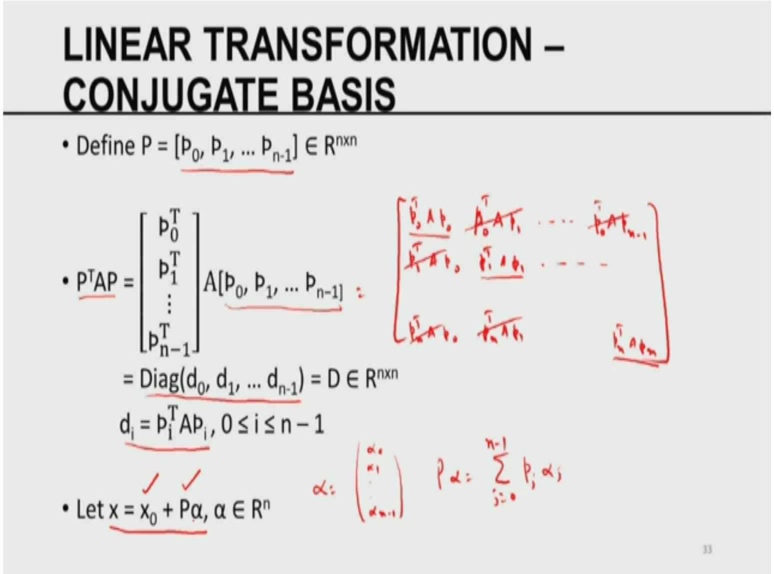
(Refer Slide Time: 15:50)



So, with that property at the back of our mind I am now going to pursue the notion of quadratic minimization let A be an n by m matrix. So, you can see I am still considering the model problem A be SPD f of x is one half of x transpose Ax minus b transpose x plus c, the minimizer is given by Ax is equal to b. Let R of x b the residual which is negative the gradient, which is the negative the gradient.

So, I would like to be able to, I would like to be able to minimize f of x which is one half of R transpose R which is given by this expression which when expanded this given by this expression. If I compute the gradient I get this and that leads to Ax is equal to b, if A is SPD. So, this essentially tells you quadratic minimization problems and solution of linear systems are essentially one of the same are essentially, one of the same. So, minimization of the sum of the squares of the residual is what we are concerned with and its relation to the given quadratic function.

(Refer Slide Time: 17:27)



So, we are now going to look at a representation in the new basis that is constituted by the conjugate directions. So, let p be a matrix that is built out of the conjugate direction vectors is a n by n matrix. So, if I now p transpose A p, p is given by this p is given by this there is A and p transpose; that means, p is given as columns. So, p transpose is given by the rows.

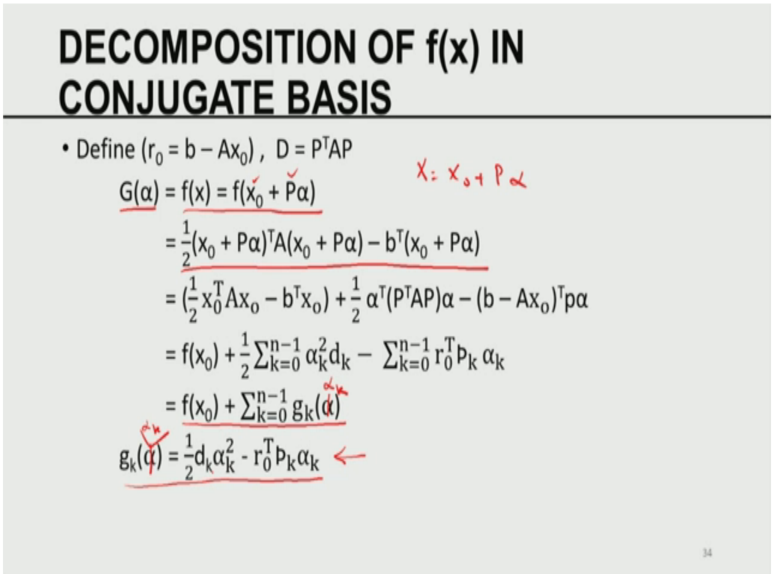
If I multiply this you can essentially see I get a matrix this is equal to a matrix where the first element is p naught transpose A p naught, the second element in here is p naught transpose A p 1, likewise p naught transpose A p n minus 1, here it will be p 1 transpose A p naught, p 1 transpose A p 1 and so on. If you consider the last one this is p n transpose A p naught, p n transpose A p 1, the last element is p n transpose A p n. In view of the conjugacy property all the off diagonal elements are 0, all the off diagonal elements are 0, the elements along the diagonal are not 0. I am now going to call d of i is equal to p i transpose A p i therefore, p transpose A p simply becomes a diagonal element with d i as the diagonal a diagonal matrix, with d as is the diagonal elements.

So, from the previous slide we now know x is equal to x naught plus linear combinations of p i. The coefficient of the linear combinations are alpha therefore, the coefficient of the linear combination alpha let alpha be a vector in R n where alpha is essentially given by alpha naught, alpha 1 and alpha n minus 1. So, p of alpha is essentially summation p j alpha j, j is equal to 0 to n minus 1 the linear combinations thereof. So, p of alpha is a very succinct way of representing the sum. So, from the previous slide we now know any x can be expressed as x naught plus p of alpha.

So, what is that I now know, x naught is known, p is known. So, if you give me an x there is a corresponding alpha. So, I am transforming x to alpha, the vector x is being transformed to vector alpha and that is the linear transformation that we are talking about. So, if I consider a point in a space x is the coordinates in the ordinary basis alpha will become its coordinate in the conjugate basis. So, I am talking about simple coordinate transformation from the ordinary basis to the conjugate basis, x transforming to alpha. So, instead of working the problem in the x space we can now work the problem in the alpha space. I also want to remind you the alpha here is not the same alpha that we talked about in conjugate, in the gradient methods.

In the gradient method alpha refers to a step length parameter here the same alpha they are the alpha is a scalar here alpha is the vector. Alpha is a new transformed vector that represents points in the with respect to the new basis. So, even though we use alpha indefinite in different places it is imperative that we understand the distinction between the use of these variables their meaning thereof.

(Refer Slide Time: 21:47)

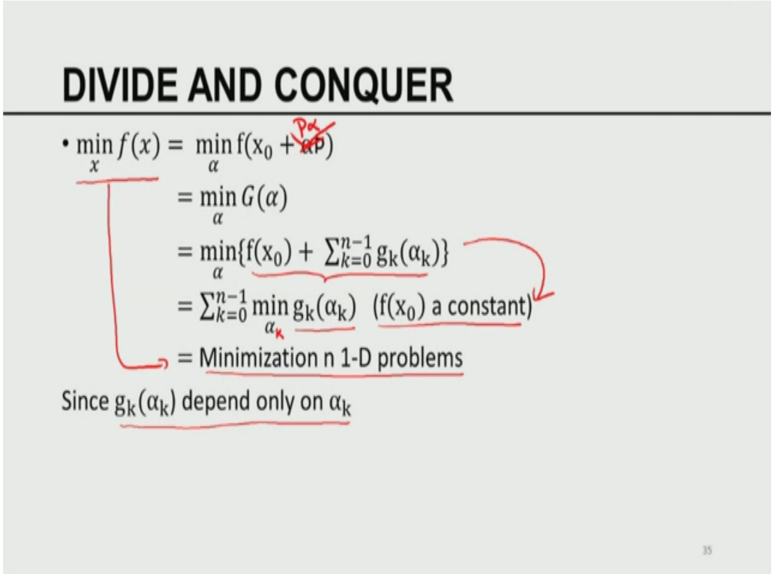


So, what is the basic idea? I have been given f of x, f of x is given by one half of x transpose Ax minus b transpose x plus c. Therefore, if I substitute f of an any x we already know can be represented by x naught plus p of alpha. So, f of x can be replaced by f of x naught plus p of alpha since x naught is known, p is known this is simply a function of alpha I am going to call that function as G of alpha. So, G of alpha is a new name to the same function f of x; f of x denotes the representative function the old basis G of alpha represents the same function in the conjugate basis.

So, I am now going to work the problem not in the ordinary basis, but in the conjugate basis for the sake of this analysis. So, when I substitute x is equal to x naught plus p alpha, alpha in the expression for f of x that takes this form this when simplified using a sequence of matrix vector operations it becomes equal to this f of x naught plus k is equal to 0 to n minus 1 g k of alpha, where g k of alpha is given by this function you can readily see g k is a quadratic function in alpha. That means, I have expressed the f of x I have decomposed it. So, f of x has coupling because the matrix A is symmetric positive definite the off diagonal elements enabled you to couple various variables. The importance of representing in the conjugate basis is that it is decoupled. So, that is the decomposition we are talking about. So, I am trying to express the function in a decoupled form where G of alpha, where G of alpha is represented by G of each component, where G of alpha is represented by each component. So, it is, I probably should say this is alpha k that would be probably better, this also should be alpha k.

So, look at this now. G of alpha is simply sum of g k of alpha k, where g k of alpha k depends only on alpha k. So, that is the decomposition that we have achieved. There is no product terms among alphas each of g of k is a quadratic in only alpha k. So, this is the decoupling or decomposition that we have achieved in going from ordinary basis to the conjugate basis nothing has changed except the representation.

(Refer Slide Time: 24:48)

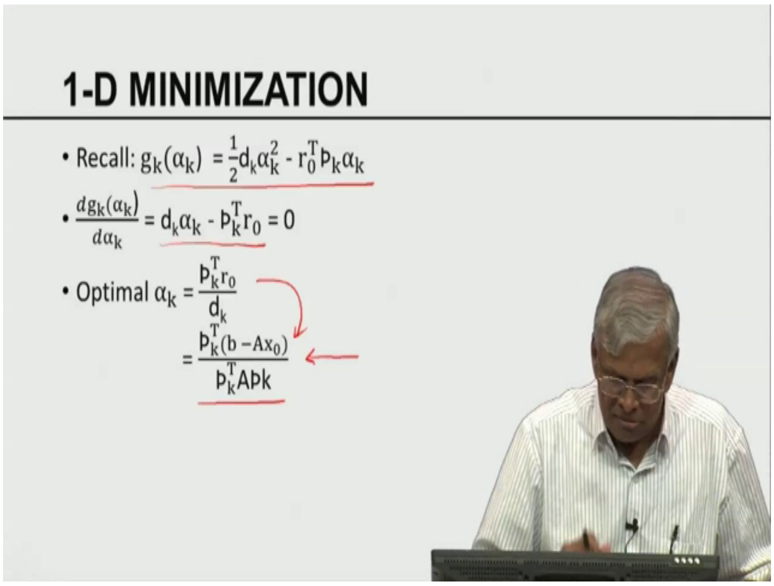


Therefore, minimizing f of x in the x space in the in the original coordinate system is equivalent to minimizing f of x naught plus alpha x naught plus I should have said this is I am sorry this is p of alpha p of alpha, but x naught plus p of alpha is minimizing with respect to G of alpha. But G of alpha from the previous page is given by this, right. The terms in the parentheses are sums of individual alpha ks they depend only on the individual alpha ks. Therefore, the minimum with respect to the vector I can replace it by summation minimum with respect to alpha k for each k, this is alpha k with respect to each k because f of x naught is a constant. So, that does not change the analysis.

So, what did that we have accomplished? Minimization of x in the n dimensional space now reduced to minimization of n one dimensional functions which are called g of k of alpha g k of alpha k. Please remember each g of k depends only on alpha k. So, the transition from here to here is very crucial very critical this transition depends on our ability to decompose because g 1 depends only on alpha 1, g naught depends on only alpha naught, g 3 depends only on alpha 3, g k depends only on alpha k. Therefore, we have made the problem to be 1 of n simultaneous minimization of g of alpha k which are one dimensional, one dimensional problem.

So, to say in other words minimization of one n dimensional problem is converted into the minimization of n 1-D problem that is divide and conquer. So, a hard is decomposed into n small sub problems in 1-D that is the fundamental achievement of going from the original basis to the conjugate basis. I hope you are able to recognize the power of conjugacy in transforming in n dimensional minimization to minimization of n 1 dimensional problems.

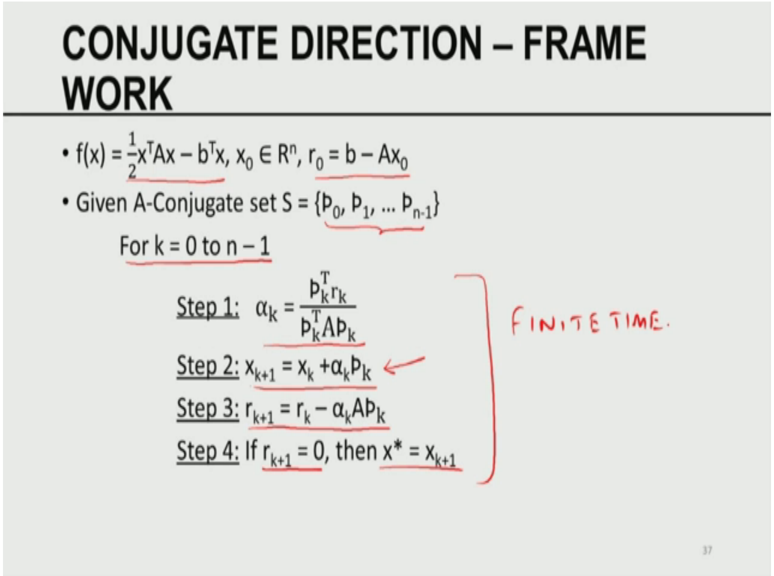
(Refer Slide Time: 27:27)



Now, let us concentrate on 1-D problem let g of alpha k is equal to a quadratic function d k is known r naught is known p k is known. So, it is simply a function of alpha k. So, if I compute the derivative of g of k with respect to alpha k, I get this and if I equate this to 0 I get alpha k to be given by this and that is given by this formula which is very well known and you can readily see this formula is very much related to the formula that we have derived in the early slides.

Therefore, we have now minimized each of these functions separately with respect to alpha k and the minimizer alpha k is given by this form particular formula. So, we have achieved minimization of n one dimensional function simultaneously where the minimizer is given by this particular formula.

(Refer Slide Time: 28:33)



So, this provides us a framework for what is called conjugate direction method is an idea. So, let me summarize this now. Let f of x be one half of x transpose Ax minus b of x, r naught is b minus A of x naught.

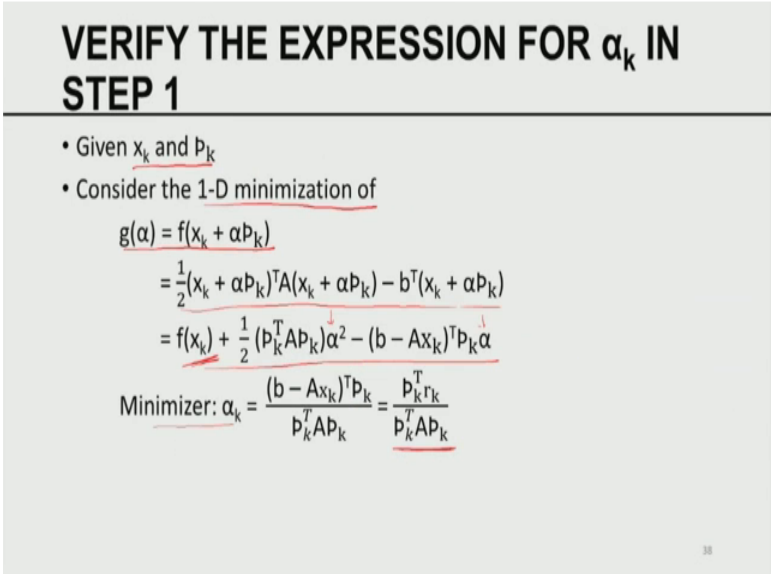
Let us assume I am given a set of n A conjugate direction the whole analysis depends on the existence of the n conjugate directions prespecified given to us. So, if somebody gives me a set of n A conjugate direction, where a is the matrix of the quadratic form where a is the symmetric positive definite matrix then from the above analysis, what do they can do? I can try to minimize each of the g alpha k for k running from 1 to n that is exactly what is being done here for k running from 0 to n minus 1, step one find alpha k the formula that is given in the previous page, compute x k plus 1 is equal to x k plus alpha k p k. That means, I am moving in the direction of the conjugate gradient I would like to now remind you that this is distinct from what we did in the gradient method the gradient method x k plus 1 is equal to x k plus alpha k, r k. So, there we went in the direction of r k which is the negative the gradient here I am going in the direction of the A conjugate direction. So, that is the primary difference between the 2 ideas p k is A conjugate direction.

The residual also can be updated according to step 3. I am now going to test if the residual is 0 if the residual is 0 I get out x then x star is equal to x k plus 1 otherwise you continue. Another fundamental difference between this algorithm and the gradient algorithm is that in the case of gradient algorithm we had a for loop where we said k is equal to 0 1 2 3 up to infinity there is an infinite loop we had an exit condition, here is a finite loop 0 to n minus 1 that essentially tells you I have finite time convergence, the finite time convergence essentially implied by the decomposition that we have produced earlier. So, I can solve one n dimensional problem as n one dimensional problem if I did these n sub problems a sequence I am done. So, the notion of a finite time convergence is inherent in this analysis.

The conjugate direction method, the conjugate direction framework essentially summarizes this idea conditioned on the fact I have been given a set of n A conjugate directions. I still allow the possibility of being able to get out soon if something happens therefore, this provides you a general framework for finite time convergence finite time convergence. This was the idea that was proposed by Hestons in the early 50s this is one of the one of the landmark results in the theory of minimization domain and ever since the conjugate direction based ideas have been exploited and we would like to be able to tell you that this is not, this is not called conjugate gradient method it is simply A conjugate direction framework, it is an idea if you do not have an idea you cannot develop an algorithm.

So, what is the essence of this idea? If I have a quadratic function by choosing a set of n A conjugate direction, I can convert n, I can convert one n dimensional minimization problem to a set of n 1-D minimization problem I can solve these n one dimensional minimization problems in a sequence. So, you know more than n steps I should be able to achieve the minimum the overall minimum of the original function f I am seeking. That is the message of this analysis called, analysis of what is called conjugate direction framework.

(Refer Slide Time: 33:23)



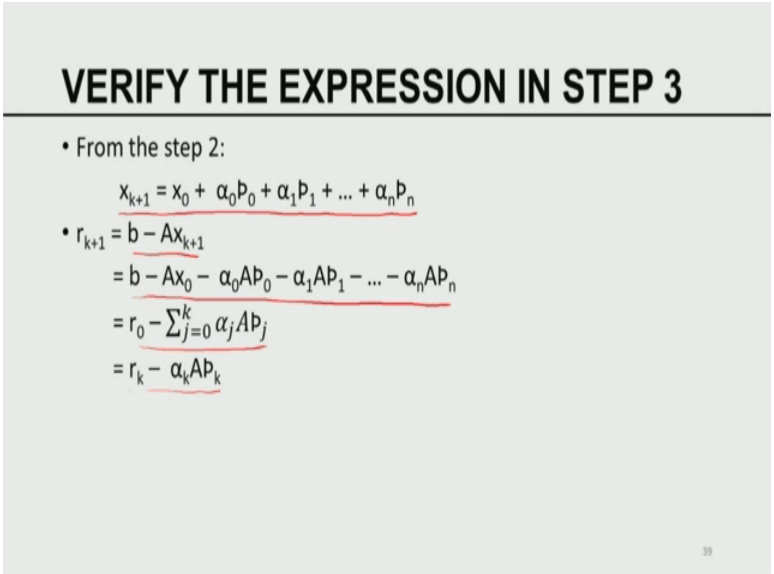
We would like to do some checking to further reinforce the idea of the power of the conjugate direction methodology. So, let us assume given x k and p k what is that; x k is a given operating point p k is the conjugate direction.

Even though we do the analysis in the conjugate basis computations are done in the original basis I want you to remember the thing. So, we do the analysis and the conjugate basis the computations are done the original basis. So, we need to be able to go between these 2 spaces and we would like to be able to reinforce some of the properties of conjugate directions by answering specific by examining specific relations. So, let x k be given, let p k be the conjugate direction along which I am going to move from x k in going to x k plus 1. So, the one dimensional minimization problem in this case becomes g of alpha is equal to f of x k plus alpha times p k if you substitute x k plus alpha p k in my function the function expression takes this form which can be reduced to this, f of x k since x k is given f of x k is known that is a constant.

So, it is the quadratic function in alpha you can readily see I want to minimize this quadratic function I get alpha k is equal to given by this and this is the formula that we had achieved in our earlier analysis. So, this is a further collaboration and verification of the properties looking at the conjugate direction method as one that starts at x naught and minimize as a function along the one dimensional ah direction p k. So, that is the aspect of the verification. So, we are trying to do everything similar to what we did in the gradient direction the only difference being in the gradient method we went along the direction negative the gradient here we are going in the direction of conjugate direction, A conjugate direction p k.

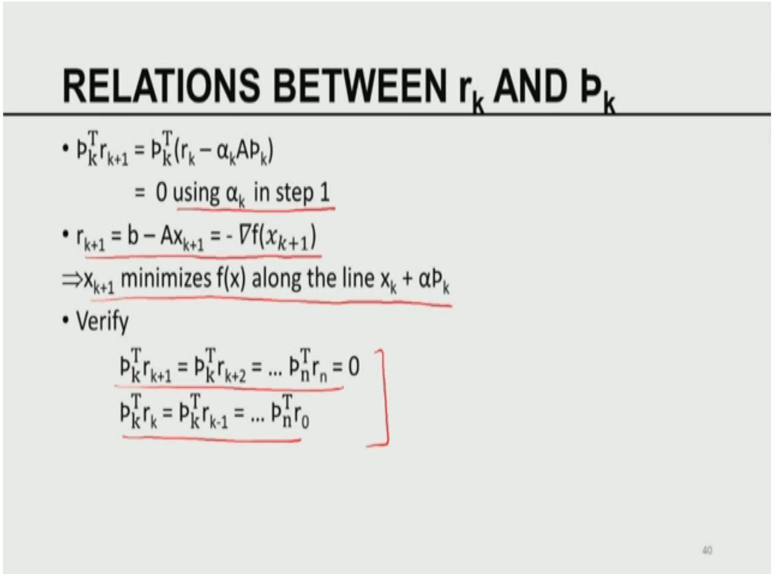
To verify the expression in step 3. So, let us go back, step 3 the conjugate direction method I am going to quickly review.

(Refer Slide Time: 35:58)



From step 2 if you iterate it from x naught x k plus 1 takes the following shape x k plus1. So, r k plus 1 this must be yeah, x k plus 1 is a vector that is given by that and r k plus 1 is given by this and if I substitute x k plus 1 that expression becomes this which can be replaced by this that is exactly equal to r k minus alpha k a P k. So, that is the expression we have got for step 3.

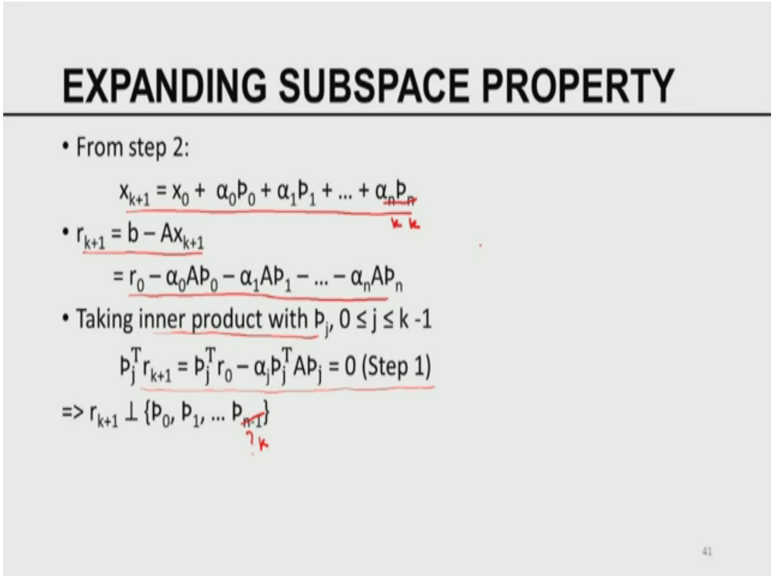
(Refer Slide Time: 36:52)



Relation between r k and p k p k transpose r k plus 1 is 0 using alpha k in step one. I would like you to verify this, these are all important properties one should verify that that; that means, r k plus 1 and p k are orthogonal to each other. Please remember in the case of conjugate gradient method r k plus 1 and r k are orthogonal. Here r k plus 1 is the gradient of the function at x k plus 1 that implies x k plus 1 minimizes f of x along x k plus alpha p k.

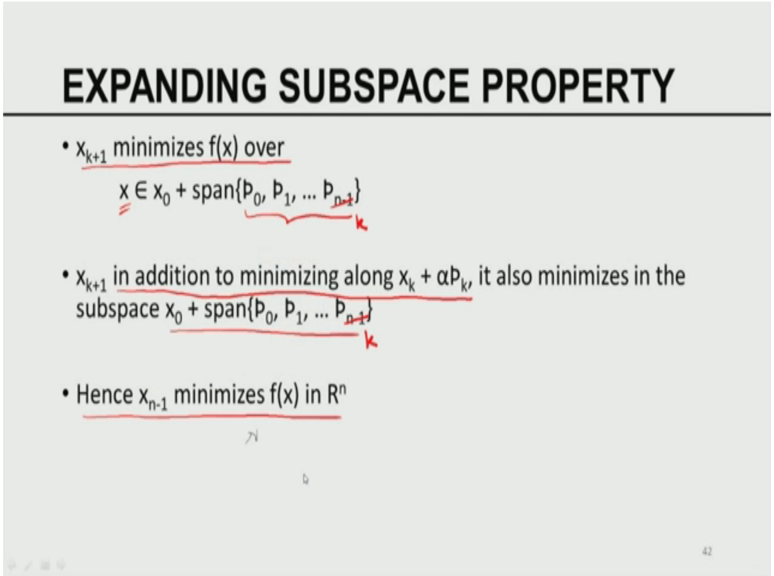
So, from here you can readily verify the following sequence of relations p k transpose r k plus 1 is equal to p k transpose r k plus 2 p k transpose r n is 0. Likewise p k transpose r k p k transpose r k minus 1 is equal to p n transpose r naught there is an inequality that is the orthogonality. These two properties essentially follow from the analysis we have given these are important properties that one should examine one should understand. This essentially tells you the intrinsic relation that exists between conjugate directions and the gradient directions which are the residual armature negative the gradient.

(Refer Slide Time: 38:09)



Another thing is called the expanding subspace property is another interesting aspect of the conjugate direction method x k plus 1 can be expressed as x k plus 1 can be expressed as this. So, r k plus 1 is given by this which we have already seen used that is equal to r naught plus this. Taking the inner product of both sides with respect to p j, I can you can readily verify that this is the result that one gets. So, what does this mean? r k plus 1 let us go back to the previous one, r k plus 1 is orthogonal to p k, p k. In here what is that we have seen r k plus 1 is perpendicular to, is perpendicular to this set; that means, that is what is called the expanding property the expanding property I believe this must be, I want to check this. I want to check this property, I will probably correct it later.

(Refer Slide Time: 39:30)

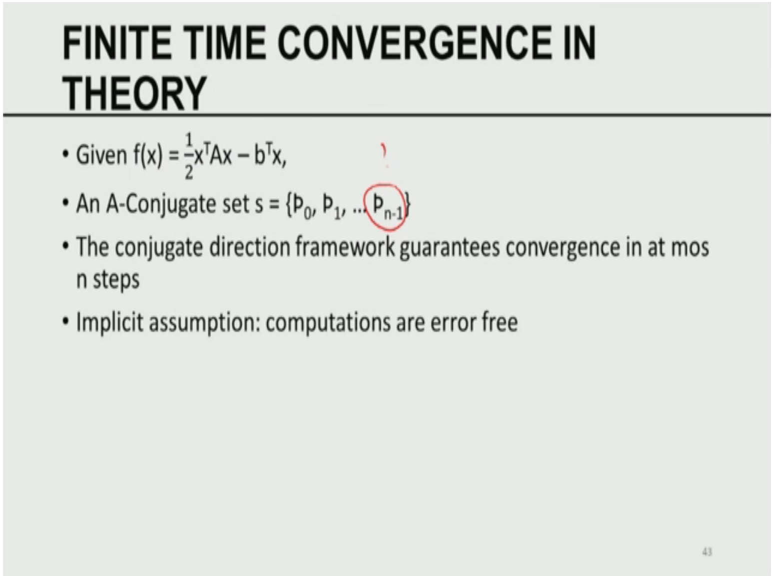


So, that essentially tells you x k plus n minimizes over the span of, over the span of p k plus 1 this must be I am not very clear about. I will have to check the correctness of this I will come back to that. So, what is the basic idea? The basic idea is I am sorry the basic idea is x k plus 1 minimizes over a subspace and the subspace is expanding. So, x naught minimizes over p naught, x 1 minimizes over the span of p naught and p 1, x 3 minimizes over p naught p 1 p 2, x k plus 1 minimizes over the span of p naught, p 1, p 2, p k.

So, in this way when I consider all the vectors p naught through p n minus 1 and span of it and x naught plus this span of it if x belongs to that that minimizes that minimizes f of x. So, that is the fundamental relation that comes out of this expanding subspace property. So, in addition, so what is the basic idea? In addition to minimizing x k plus 1 in addition to minimizing x k plus alpha p k it also minimizes over the subspace therefore, x n minus 1 minimizes f of x over r n. So, I believe this must be k, I believe this must be k that is the correct way to look at it.

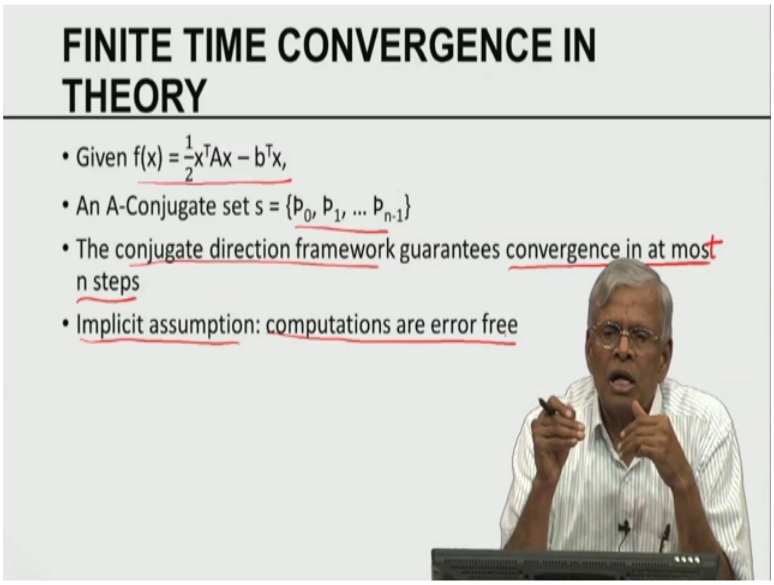
I also believe this must be this must be sorry, I also believe this must be k that is the foundation for the, that is the foundation for the, this is k, this is also k. I can, I will correct this I will send the corrected version of these therefore, you can see what is the summary of this the summary of this is each iterate not only tries to minimize in the direction chosen it also minimizes in the subspace spanned by all the previous conjugate directions. So, when I come to x n minus 1 x n minus 1 minimizes f of x over the span of p naught, p 1, p 2, up to p n minus 1 since p naught p 1 p 2 p n minus 1 span the whole space is the basis I have minimized it over the entire r n. So, that is the fundamental idea of this expanding subspace property, expanding subspace property.

(Refer Slide Time: 42:43)



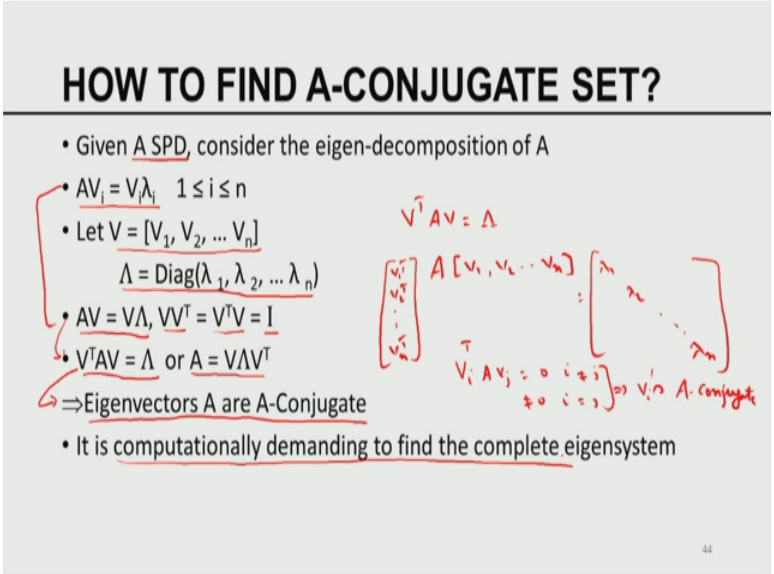
So, that essentially gives you the notion of, that essentially gives you the notion of finite time convergence that essentially gives you the notion of finite time convergence which we are now going to state explicitly. So, given f of x is equal to, given f of x is equal to x transpose Ax minus b transpose x, given a set of conjugate directions.

(Refer Slide Time: 43:06)



The conjugate direction framework guarantees convergence in at most n steps, in at most n steps; that means, finite time convergence, but what is the tacit assumption that we have making the implicit assumption in this is that computations are error free. That means, I have a hypothetical machine which has infinite position. So, if I have a computer with the infinite position there is no computational error I can check for a conjugacy perfectly. So, if I have the ability to examine the a conjugacy perfectly this framework essentially provides you finite term convergence of minimization algorithm provided we do the searches along the conjugate directions. That is the principle conclusion that comes out of this.

(Refer Slide Time: 44:18)



The question here is that we have assumed the presence of n conjugate direction, the whole question is this we did not show or we did not verify such A conjugate direction exists. Sure to prove the existence of conjugate direction now I am going to look at eigen decomposition of A and show the eigenvectors of A are in principle could be used as conjugate directions. So, if I can show that we already know at least one set of conjugate directions exist if there is one set of conjugate direction exist the framework for conjugate direction as we have developed make sense.

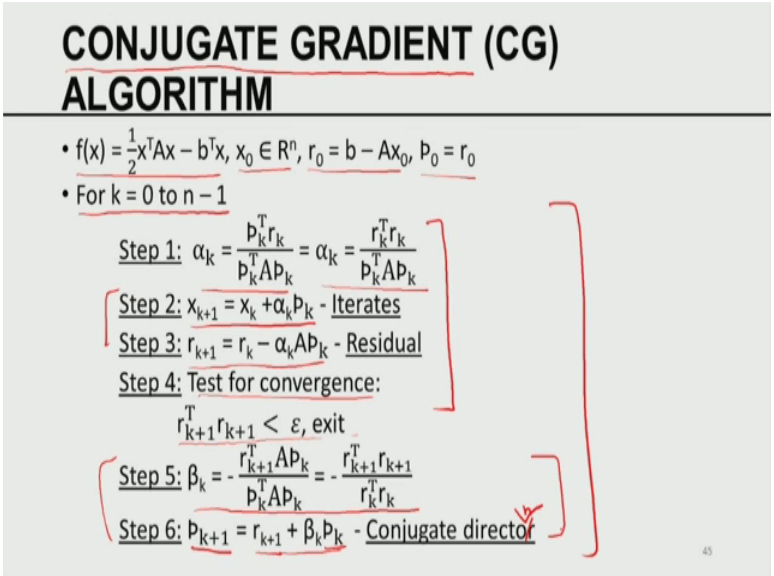
So, in order to show such A conjugate direction exists I am now going to start with a given matrix A which is SPD, a given matrix A which is SPD. Consider the eigen decomposition AV i is equal to V i lambda i. I am going to consider the matrix of eigenvectors V. I am going to consider lambda of diagonal elements this relation can be expressed as a matrix relation a V is equal to V lambda VV transpose V transpose V is i, V transpose therefore, from this relation now we now have either V transpose AV is lambda or a is equal to V lambda V transpose.

So, what does this tell you? V is equal V transpose AV is equal to lambda essentially tells you if you. So, let us consider that V transpose AV is equal to lambda what does it tell you it tells you the following A V 1, V 2 V n here I am going to have V 1 transpose V 2 transpose V n transpose if you consider this, this is going to be a diagonal matrix lambda 1, lambda 2, lambda n. So, this essentially tells you V i transpose a V j is equal to 0 if i not equal to j is not equal to 0 if i is equal to j this essentially tells you the Vs the V is are A conjugate. The V is are essentially are essentially A conjugate V is are essentially V is are A conjugate, A conjugate.

So, that essentially proves that I have at least one system of A conjugate direction for a given matrix and A conjugate directions are essentially the eigenvectors of A. Even though this proves the existence of conjugate direction it is computationally extremely demanding to find the complete eigen system that can be used as A conjugate directions because you have to spend lot of money in trying to find the eigenvectors. So, you spend lot of money to find the eigenvectors and then you have to perform the minimization the n one dimensional minimization as dictated. Therefore, this idea of using eigenvectors of A as conjugate direction while in principle feasible is computationally inexpensive is computationally demanding therefore, we should look for an alternate method for defining conjugate direction which are much less expensive.

This idea of trying to incorporate the method of finding the conjugate direction along with the search as we go on is the principle that is embodied in conjugate gradient method. So, what is the difference between conjugate direction method and conjugate gradient method? The conjugate direction method is not an algorithm is a framework it essentially tells you if you give me a set of n A conjugate directions I can do the analysis I can prove I can converge in n steps. So, that is simply the framework it does not confine, it does not consider how do you how does one deliver the n A conjugate direction therefore, in order to make this framework a reality we must integrate the process of defining the conjugate direction along with the search along with the one dimensional search combined in a very nice way that will guarantee not only a conjugacy, but also finite time convergence. These two ideas melding together gives rise to a new class of algorithm they are called, that is called conjugate gradient algorithm. So, that is the difference between conjugate direction and conjugate gradient.

(Refer Slide Time: 49:23)



So, the basic principle the conjugate gradient algorithm we already saw. Now I am going to describe the various steps involved. Given the function f of x let x naught be initial point are not as the initial residual. I am going to choose the initial conjugate direction to be the same as the initial residual. Please understand I need a set of n conjugate direction the first direction could be anything.

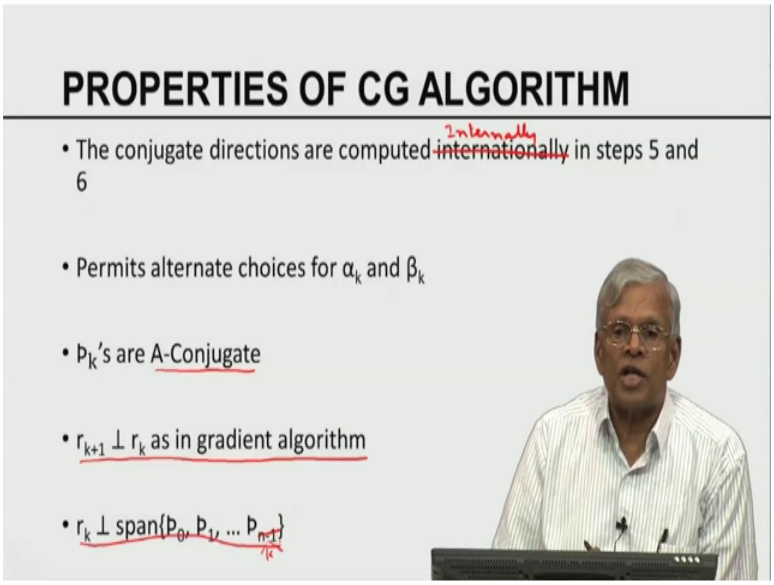
Here we are going to pick the first conjugate direction to be the negative of the residual at x naught. So, p naught is r naught. So, for k running from 0 to m minus 1, I compute alpha k by this formula I can also compute alpha k by another formula these two formulas essentially the same. We are not going to indulge into the proof of the equivalence between 2 expressions like this many books and many papers written on conjugate gradient method essentially gives you the ability to compute these in two different ways both are equivalent.

Now I am going to iterate, I am now going to update the residual we are going to test for convergence, that test for convergence essentially rests on the smallness of the magnitude of the residual. If the residual is not small we need to continue, we need to first define A conjugate direction, conjugate direction is not directable, conjugate direction. So, steps 5 and 6 together help you to define the conjugate direction, p k plus 1 is the new conjugate direction; p k is the old conjugate direction r k plus 1 is the new residual I have already computed. So, using the new residual and the old conjugate direction I am taking a linear combination to get the new conjugate direction the coefficient of the linear combination is beta k and beta k is again given by two ways of computation one by this formula another by the other formula. So, 5 and 6 together define the conjugate direction.

Step 2 and 3 define the update of the iterate and the update of the residual. The step one essentially gives you the update of the coefficient which is used the step length parameter. So, step one gives you the step length parameter step 5 gives you the step length parameter needed to define the conjugate direction. Step 2 and 3 define the iterate and the update of the residual vectors. Step 4 essentially gives you a convert a convergence test. The overall convergence is repeated no more than n times 0 to n minus 1. So, if the computer is such that either is no roundoff errors this gives you a framework to be able to minimize in n steps.

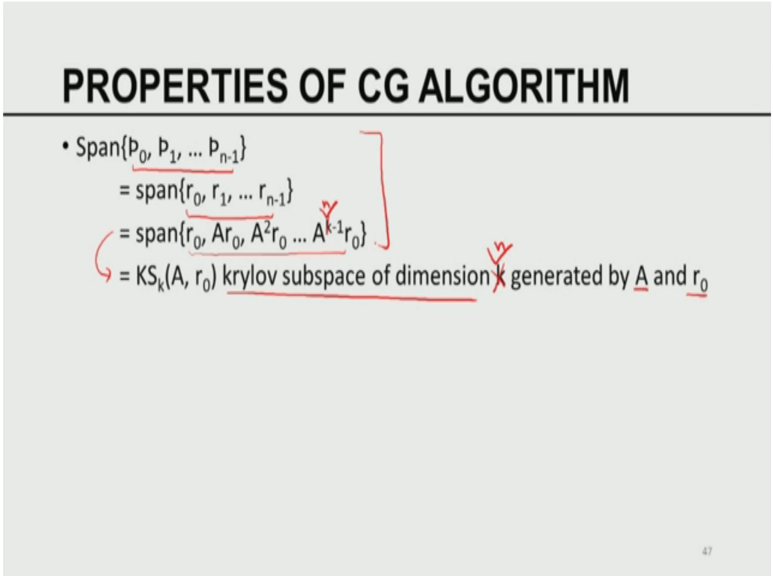
The advantage of this framework is that you are not, you need not be given a prior, a set of conjugate direction I can build the conjugate direction iteratively as I proceed. So, this ability to integrate the search and the building of the conjugate direction together simultaneously in this process is the power of the idea behind the conjugate gradient algorithm, conjugate gradient algorithm has been a very powerful workhorse in the industry. So, we understand the properties of steps 1 through 4. So, what is the only thing that one need to understand we need to be able to show that the p k defined by steps 5 and 6 are indeed A conjugate and here is a summary of the properties of the conjugate gradient algorithm CG.

(Refer Slide Time: 53:26)



The conjugate directions are computed is not internationally I am sorry, there is an error there is a typo. Internally, it is computed internally in steps 5 and 6 is not given a prior i; alternate choices of alpha k and b k are given in the respective steps they are equivalent p ks are A conjugate I am not going to indulge in the proof of that that will take a little bit longer time, but the proof that p k is so generated or A conjugate are contained in several sources. r k plus 1 is perpendicular to r k as it happens the gradient algorithm. So, the residuals preserve the same property as the gradient algorithm and the r k is also perpendicular to the span I think this must be k, I am sorry, I think this must be k I have to carefully check some of these things I will do that.

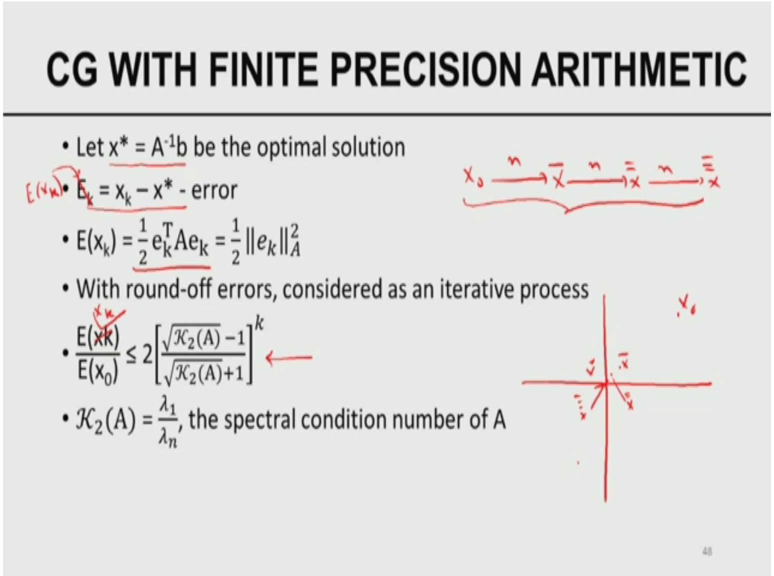
(Refer Slide Time: 54:54)



Another property is span of p, p naught, span of p naught to p n minus 1 is the same as span of r naught to r n minus 1 which is also equal to the span of r naught, Ar naught, A square r naught, A to the power k minus 1 r naught this must be A to the power of n minus 1 r naught and this space that is generated like this is called the Krylov subspace. Krylov subspace, this Krylov subspace of dimension n is generated by A and r naught. The Krylov subspace generated by n and A and r naught; that means, given A vector r naught and A matrix A by successively multiplying r naught by A, A square, A to the power of n minus 1 I create different vectors the span of these vectors is called the Krylov subspace.

You can readily see the same space has different representation span of p, span of r, span of the vectors r naught, Ar naught, A square r naught and so on. It is this property of equivalence representation from the same space using different basis is the ultimate power of the conjugate gradient algorithm.

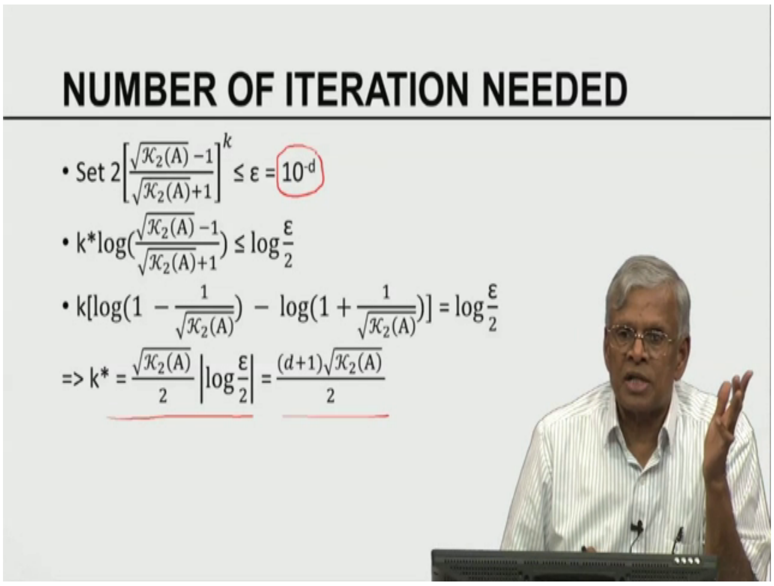
(Refer Slide Time: 56:25)



So, conjugate gradient algorithm with finite precision arithmetic. Until now we talked about conjugate gradient with infinite position. When on real; when used on real computers because of finite precision our infinite precision arithmetic we cannot check a conjugacy perfectly. So, what we think as A conjugate direction are not precisely A conjugate they are only approximately A conjugate direction therefore, if x star is the optimum solution if E k is the error it can be shown this must be E of x k. The E of x k is given by this quadratic function much like the E of x k that we used in gradient method. So, when there is a round of errors what happens you started x naught you perform n steps you come to x star, this x star because the finite precision arithmetic will not be the I should not say x star let me change the notation a little bit.

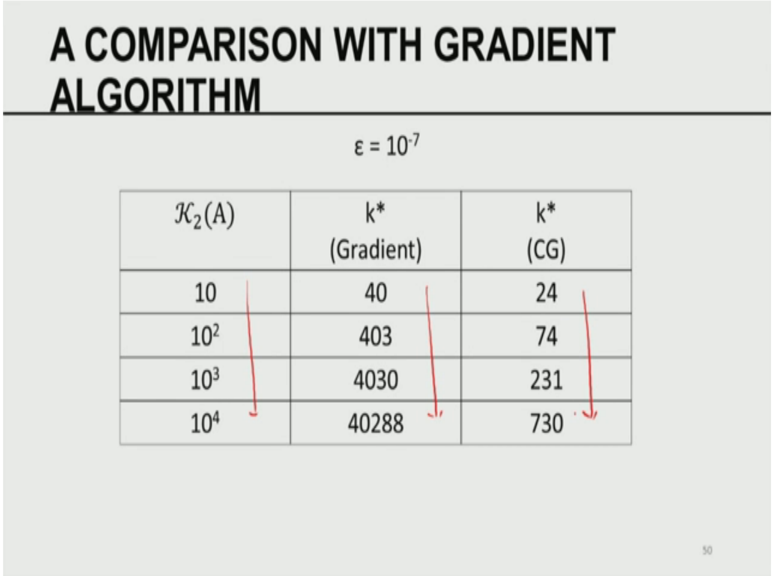
This is let us assume x bar, this x bar will not be equal to x star the minimum, but close to that. Then what do we do? You start from here you do one more n steps you go to x double bar then you start from here you do n steps you x go to x triple bar. So, it becomes an iterative process. It turns out if you consider this as an iterative process one can find out the ratio of E of x k, this is x k x sub k, E of x k by x naught is given by 2 time is less than or equal to 2 times a function that depends on kappa much like gradient algorithm where kappa is lambda 1 by lambda n is a spectral condition number. So, you can readily see in the case of iterative in the case of finite precision arithmetic you cannot achieve finite time convergence it looks as though is not it is an infinite reprocess in this case the convergence rate is given by this expression, convergence rate is given by this expression.

(Refer Slide Time: 58:51)



So, now I can do whatever I did with respect to gradient algorithm I can set this number which is an upper band equal to epsilon 10 to the power of d, by taking the logarithm I can compute an expression for k star by simplifying the, by simplifying this expression I can readily see that k star is given by this which is equal to which is equal to d plus 1 time square root of k 2 kappa A by 2. So, that is the number of steps. So, k star is the number of steps in the iterative process needed to be able to get closer of the order of 10 to the power of minus d, d is the precision 6 14 and so on. This follows again in the same idea.

(Refer Slide Time: 59:40)



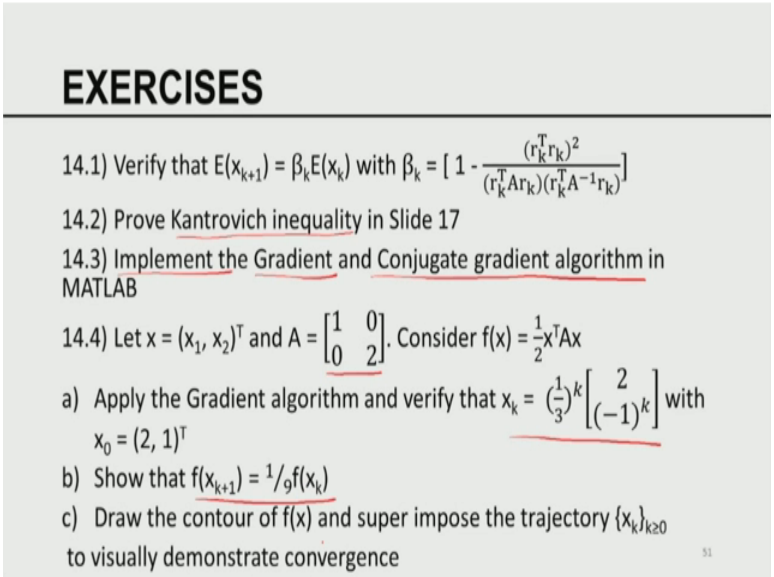
So, now, I would like to be able to compare conjugate gradient with respect to the gradient algorithm. So, here I have given you various examples of kappa various examples of kappa. Here is the number of values or the number of iterations that I have needed in the gradient algorithm here is the number of iterations that are needed the conjugate gradient algorithm.

So, with the presence of finite precision arithmetic conjugate gradient beats the gradient method hands down it can perform absolutely very well that is the power of the conjugate gradient when if there is put to the gradient algorithm, conjugate gradient with respect the gradient algorithm and this difference we can see is very measurable. So, let me summarize this now.

So, how do you how does one utilize the conjugate gradient method? That is given by the following if you start at x naught you go to x bar in n steps, if you start at x bar you go to x double bar in n steps if you start at x double bar you go to x triple bar in n steps. It is said that in most of the time, when you are doing an experiment when we are doing an experiment it is enough to repeat it about 3 times. So, used I will give you another graphical representation you start at x naught you get to x bar here, you get x bar here you start at x bar you go to x double bar closer to r, you then x. So, you get ever closer in other words I will tell you the basic idea here. So, you start at x naught you come to x bar x double bar is closer than that this is x double bar x triple bar is even closer than that. So, that is x triple x triple bar and that is the minimum which is x star.

So, x star is the minimum x double star is closer to the minimum that x bar x triple bar is closer to the minimum then x double bar. So, it is said that if you apply the conjugate gradient method in 3 phases 3 n iterations in principle you should be able to get very close to the optimum and that is the power of the conjugate gradient method.

(Refer Slide Time: 62:20)

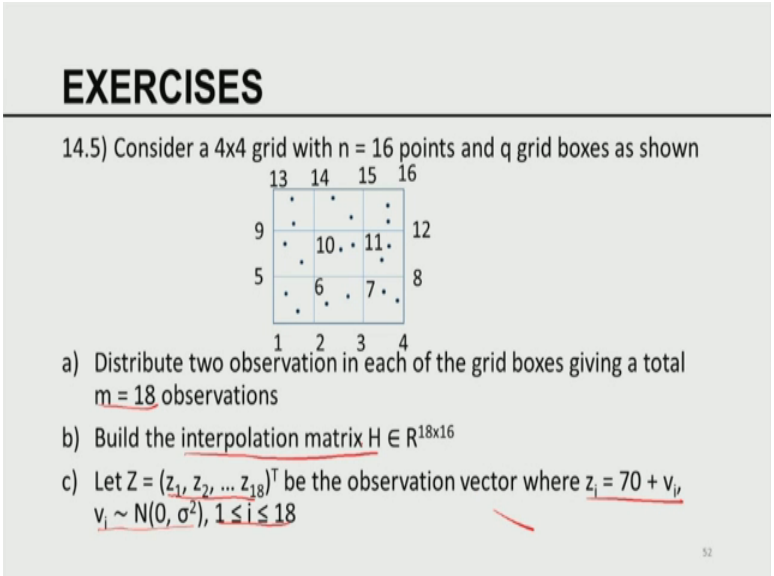


With this we conclude the overall presentation of the minimization algorithm we said there are Gradient algorithm, Conjugate Gradient algorithm and Quasi Newton algorithm. For lack of time you will not indulge in the analysis of quasi newton algorithm.

We have given a several sets of exercises. The exercises relates to verification of very many different properties of gradient and conjugating the algorithm. I would like you to indulge in the prove of Kantrovich inequality. I would like you to implement gradient algorithm and conjugate gradient algorithm on this for the same problem and compare the convergence. I would like you to take a test problem with this A, apply the gradient algorithm and verify that this is the theoretical way in which the iterates proceed starting with the initial condition 2 and 1.

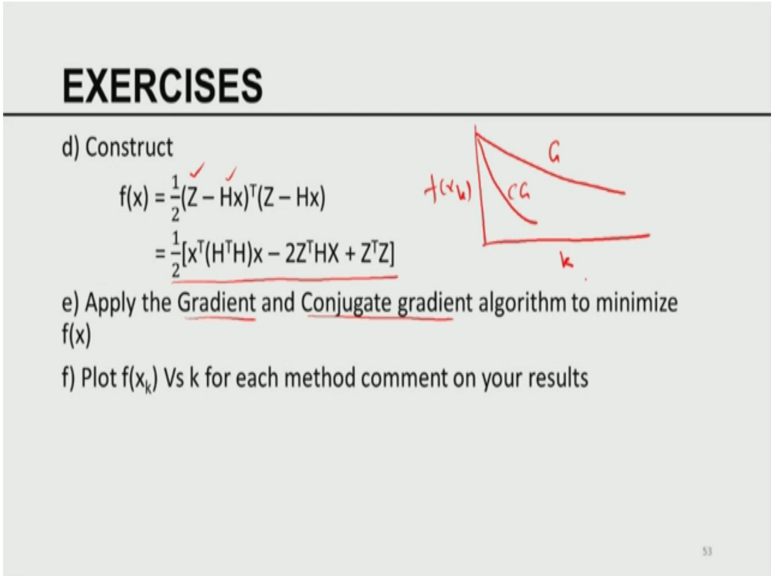
For this problem verify f of x k is equal to one-ninth of f of x k plus 1 is equal to one-ninth of f of x k. I would like you to draw the contours and super impose the trajectory so that you can visually demonstrate the convergence.

(Refer Slide Time: 63:35)



I would like to now combine couple of problems you consider the 4 d grid with 16 point. I am given observations at 2 observations at each locations. So, to distribute 2 observations each of the grid boxes m is equal to 18 observations, m is equal to 18 observations there are 16 points is an over determinant system, build the interpolation matrix H which is 18 by 16, create artificially 18 observations of temperature. Let us assume z i is equal to 70 plus v i ,v i is the random noise for i is equal to 1 to 18.

(Refer Slide Time: 64:29)



So, we have 18 observations you have the matrix H. We can now consider the problem a quadratic minimization problem Z minus H of x transpose Z minus H of x, Z is given 18 observations we have already you have already generated, you already have the matrix. So, it is a function of x. This is given by this, this is the quadratic function to this quadratic function you can apply the gradient algorithm and the conjugate gradient algorithm and compare and what that I would like you to do? I would like you to be able to plot the value of f of x k for the gradient algorithm for the conjugate gradient algorithm, the gradient algorithm conjugate gradient algorithm. You can readily see the value of f of k reduces faster for the conjugate gradient algorithm compared to the gradient algorithm because we have already seen from the table that conjugate gradient algorithm requires much smaller number of iteration compared to the gradient algorithm. So, this will essentially help you to verify the power of the conjugate gradient algorithm means solving problems.

With this we come to the end of the discussion of the optimization algorithms. With this we have also completed some of the fundamental mathematical background needed finite dimensional vector space, matrix properties, properties from multivariate calculus, principles of optimization, matrix based algorithms as well as minimization algorithms. These are the various topics that address the crux of the mathematical tool needed in doing data assimilation. With this behind us from now on we are simply going to be concentrating on saw on solving various types of inverse problems, our next step is to be able to look at dynamic inverse problems leading to the standard 4 d var methods and that is what we will begin in our next lecture.

Thank you. Bye.