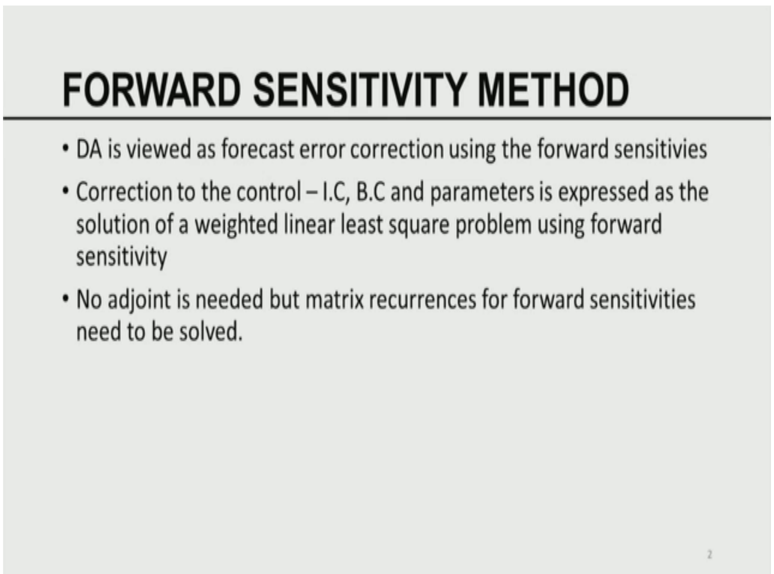
Dynamic Data Assimilation
Prof. S Lakshmivarahan
School of Computer Science
Indian Institute of Technology, Madras

Lecture – 22
Forward sensitivity method

In this module 5, we have been developing methods for assimilating data into deterministic dynamic models. The models are deterministic and we assume the models are perfect. The observations are noisy we are given a finite set of observations we would like to be able to assimilate this observation into the model to be able to determine the initial condition. In this context we developed the 4-D Var method, it is also called first order adjoint method. We developed the details of 4-D Var adjoin methods in the last lecture. We demonstrated the power of these methods using linear models, linear observation, non-linear models, non-linear observation.

In today's talk we are going to be talking about another related approach to assimilating data to assimilating noisy data into deterministic model which we assumed to be perfect. This method has is called the forward sensitivity method it was developed by us around 2010 and it can be shown that this method is in some sense dual of the 4-D Var and we would like to be able to present the details of the forward sensitivity method to correct forecast errors using data.

(Refer Slide Time: 01:50)

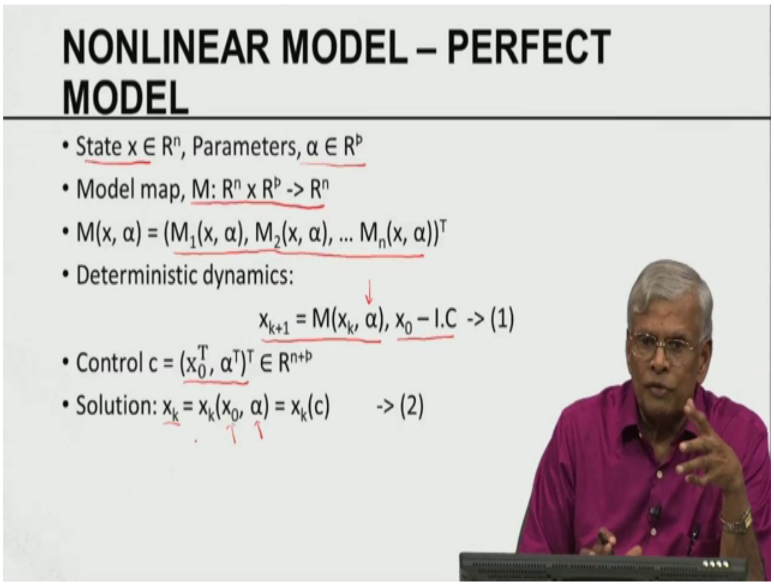


Data assimilation can be viewed as forecast error correction using forward sensitivities that is the view we are going to be taking in this class.

The correction to the control, please remember the solution to a dynamic model depends on the initial condition the boundary condition and parameters. So, initial condition boundary condition and parameters because changing them changes a solution it is customary to call them the control. So, we are assuming that the model is perfect. So, if you start the forecast model with the wrong control namely wrong initial condition or wrong boundary conditions or wrong parameters that would the error in the control will lead to errors in the forecast. We would like to be able to correct the errors in the control by using observations and we would like to be able to formulate this as again a linear a weighted linear least square problem using forward sensitivity.

In this method we do not need any adjoint with, but we need matrix recurrences that compute the forward sensitivities of the solution with respect to the initial condition boundary conditions and parameters. So, adjoint is not needed; that means, the backward integration is not needed everything is forward, but we need to be able to solve a system of system that describes the evolution of the forward sensitivities.

(Refer Slide Time: 03:48)

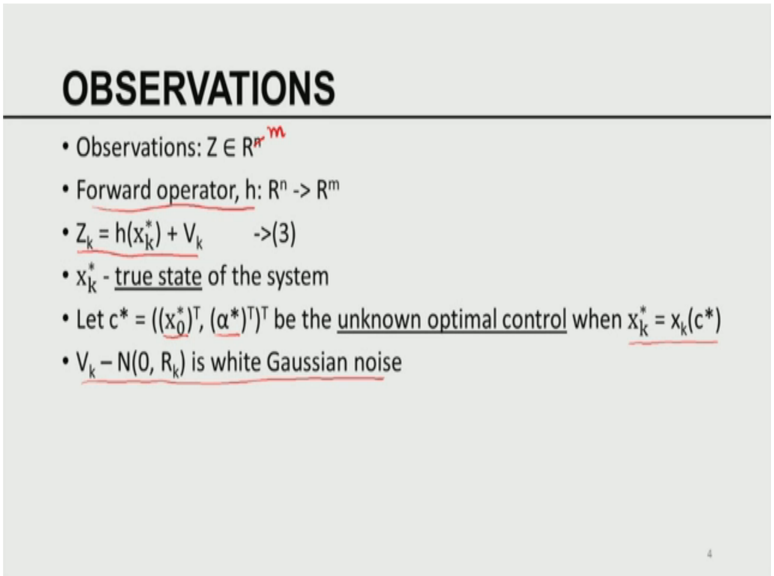


So, we are going to start with a non-linear model the assume model is perfect. Linear case will be a special case of this let x be a state, let x be state alpha be a parameter. One difference between this lecture and the previous lecture is that in the previous lecture we assumed the model is perfect, the parameters define the model because the model is perfect we assume that there is no error in parameters, the only forecast error in the previous talk was essentially due to errors in the initial condition. One way we can think of this as a slight generalization, even though the model is perfect we have not set the values of the parameters when trying to generate forecast to the right value therefore, we assume that that could be errors in the initial condition boundary condition and our parameters.

M is the model map which is the vector valued function, M depends on two vectors one is the state vector x another is the parameter vector alpha alpha is the vector of size p, p is an integer much like N is an integer. So, M of alpha is given by M 1 of alpha, M 1 of x of alpha M 2 of x of alpha M n of x of alpha.

The deterministic dynamical equation is given in the form of a discrete time dynamics x k plus 1 is equal to M of x k and alpha, x naught is the initial condition, alpha is the parameter. I am going to assume the initial value the control c to be x naught and alpha we try to generate the solution using one it can be readily seen that the solution at time k is the function of the initial condition as well as the function of the parameter. Therefore, to indicate this dependence of solution under control we call the solution x k as x k of c when there is no confusion we will simply say x k.

(Refer Slide Time: 06:16)

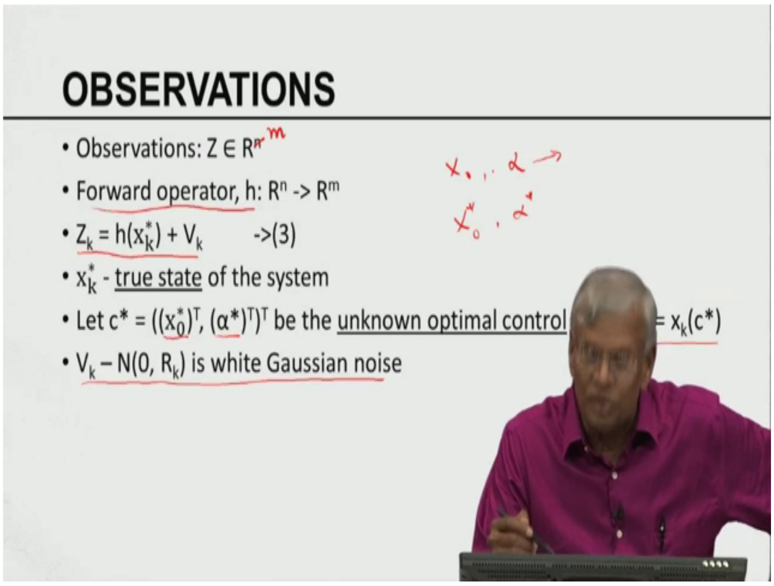


Observations are given, observations are m vectors sorry I am sorry there is there is a typo this must be m vector observations belong to Z observations are m vectors they are denoted by Z and belong to R m; h is the standard forward operator the observation at time k is given by h of x k star plus V k, V k is the true state of the system; c star is be the unknown optimal control that will lead to the true state of the system; V k is the white Gaussian noise.

Here I would like to differentiate between two types of state one is the true state which is not known to us the mother nature does not reveal all her secrets. So, x k star is the true state of nature; x k is the state predicted by the model the state predicted by the model will be equal to the true state only when the model control or set to x naught star and alpha star which corresponds to the true state of mother nature, but we do not know what x naught star and alpha star s.

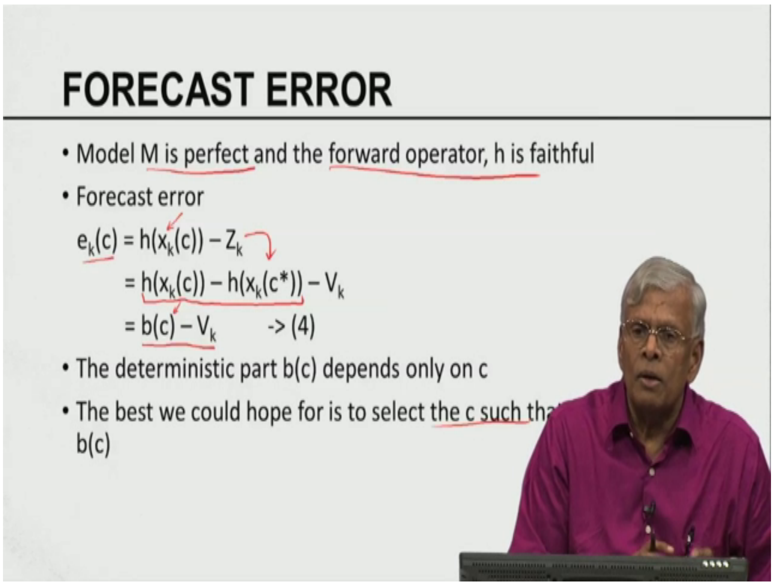
But mother so, but we only know indirectly through observation what the mother nature has selected what is the value of the initial condition that corresponds to the true state which is x naught star and alpha star.

(Refer Slide Time: 08:24)



So, we have we have picked x naught and alpha mother nature has picked x naught star alpha star, if x naught is not equal to x naught star if alpha is not equal to alpha star the solution generated out of this will have errors that corresponds to forecast errors once you know the forecast errors our job is to be able to use these forecast errors to be able to control to be able to alter the control x naught and alpha such that x naught moves closer to x star of x naught star and alpha moves closer to alpha that is the basic idea.

(Refer Slide Time: 09:07)

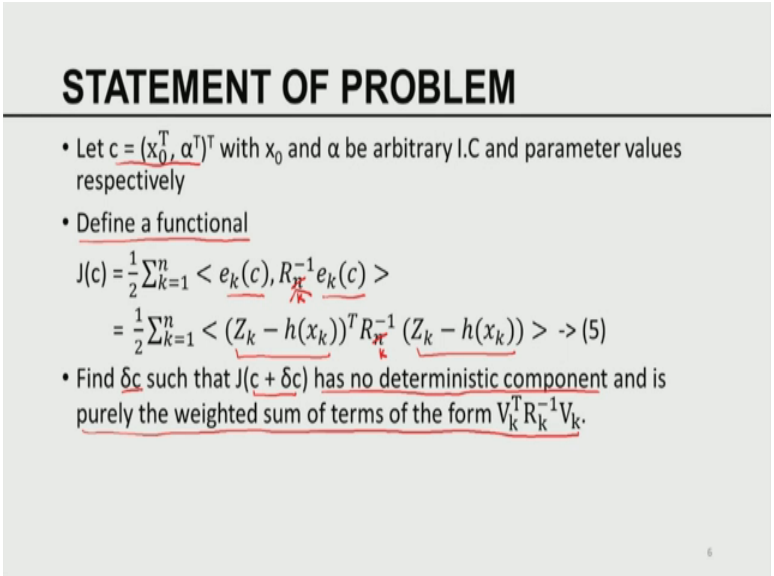


So, where it is useful to think about a classification of forecast errors which we have alluded to in module one itself, but it is worth repeating some of the principles of classifications of forecast errors for our purpose.

The model is perfect, we are assuming that the model is perfect we are assuming the forward operator h is very faithful; that means, I have the physics has been the physic that if that is part of h is very good; that means, the relation between the model state in the observation is pretty accurate. In that case the forecast error e k of c is depended on the actual forecast generated by the model. So, h of x k of c is the model counterpart of the observation Z k is the actual observation, but Z k is equal to h of x k c star minus V k. So, if I substitute for Z k here this I can rewrite it at b of c minus V k where b of c is equal to the difference between the model predicted observation and the true observation if there is no noise. So, this becomes b. So, the determine, b c is called the deterministic part of the forecast error which depends money and see the control. V k is the random part of it we cannot control the random part random part is part of the deal we have to contend with this.

So, the only way we can hope to be able to correct a forecast error is to be able to correct the deterministic part of the forecast error. So, the best way we, the best we could hope for is to select a control c such that it annihilates b c, it reduces b c to 0. If you can reduce this deterministic part to 0 then whatever is left is simply random errors which is beyond our control which is totally uncontrollable. So, this is what is called forecast error correction and this is the aspect of the forecast error that we are concerned with, the model is perfect, the forward operator is perfect the forecast error if any arises because of the errors in control. So, I would like to be able to move the chosen control towards the unknown using the forecast errors in a feedback loop to be able to control, to be able to correct the deterministic part and annihilate it.

(Refer Slide Time: 12:04)

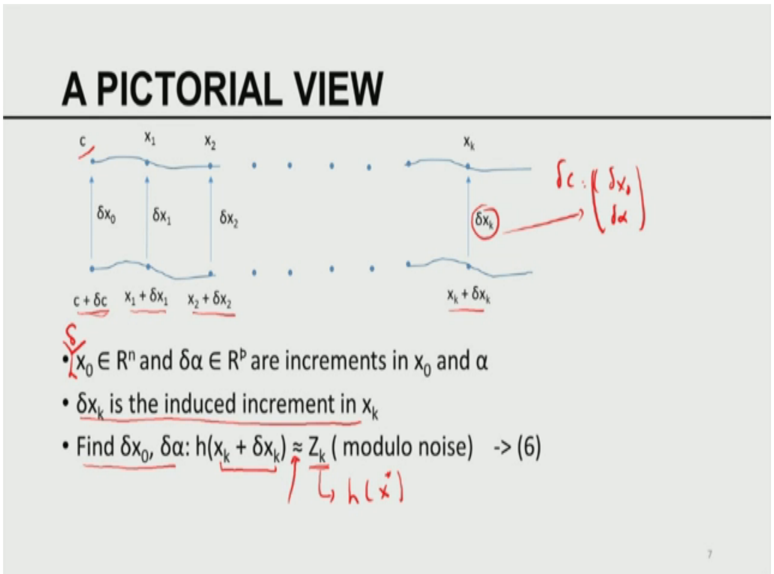


So, with that as a basis and now I am going to state the problem formally. Let c be the chosen control let c be the chosen control where x naught and c are arbitrary. But arbitrary to the extent that general engineers and scientists know they may not know the exact value, but they do know the region where the exact value lies. So, x naught and see while it is not equal to the true state, but they may not be too far from the true state as well, they are arbitrary.

Now, I am going to define a functional again look back in all of the developments in inverse problem there is always a cost functional which is very basic to the solution of the inverse problem the cost functional is essentially is denoted by gfc it is simply the weighted sum of squared errors, e k c we have seen in the previous slide it were given by the equation 4, is the forecast error this must be R k sorry, this must be R k. So, this is the weighted sum of squared of weighted sum of square of forecast errors, if I substitute for e k from the previous slide I get this expression this is the forecast error this is the forecast error transpose and this is a very familiar equation 5 that describes the cast function, is a familiar formula that we have used repeatedly in almost all of our treatment. The reason why this relation occurs again and again because our approach is rooted in least squares the common theme of our course is least squares approach two solving inverse problem, solving data assimilation problem. So, this is the least square cast functional.

So, what is our goal? Our goal is to be able to find a perturbation delta c such that if I change c to c plus delta c, J of c plus delta c has no deterministic component and is purely weighted sum of terms of the kind which are induced by the errors; that means, I would like to be able to add delta c to the control, delta c consists of delta x naught for the initial condition delta c and delta alpha for the parameter. So, how do I add corrections to the initial conditions and the parameters, so as to annihilate the deterministic part of the forecast error leaving behind only the weighted sum of squared errors of Gaussian random variables? So, please remember V k is Gaussian, V k transpose R k inverse V k it is the weighted sum of square is a normalized sum of squares of Gaussian random variables.

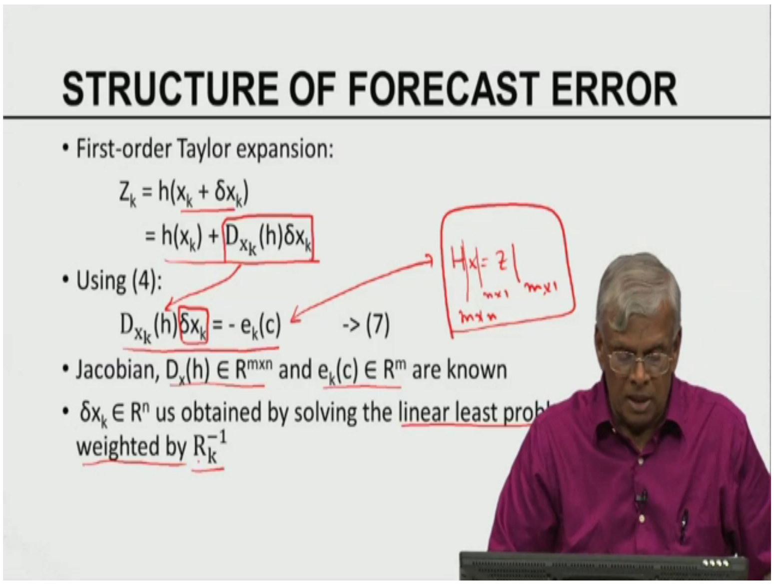
(Refer Slide Time: 15:44)



Pictorially, we are going to start from c this is the forecast x k is the forecast resulting from c. I would like to be able to change your c to c plus alpha; c plus delta c consists of delta x naught and delta alpha. So, this is the change in the initial condition delta c. So, this is the new control from the new control I can get a new state which is x 1 plus delta x 1 delta x 1 is the change in the state at time 1 x 2 delta x 2 delta x to the change in the state at time 2 x k plus delta x k is the change in the state at time k all the delta x k is induced by two changes, one delta x naught the initial condition, two delta alpha and the parameters.

I must say this is delta. So, delta x naught belong to R n delta alpha belongs to R p these are increments in x naught and alpha. Delta x k is the induced increment in x k at time k. So, what is our goal? Our goal is to be able to find delta x naught delta alpha such that when I use this new state x k plus delta x k which is the perturbed stage and applied to the function h it will give the observation model of the noise; that means, it will match the deterministic part of the noise what is the deterministic part of the noise this is h of x star h of x star that is that is essentially the deal here. That means, my model predicted observation will match the actual observation model of the noise that is the equation 6 and that is the goal with which we are going to be, that is the goal we are going to be working towards and this equality is to be interpreted in the least square sense, in the least square sense.

(Refer Slide Time: 18:27)



So, I again we would like to be able to understand the structure of the forecast errors to that end. So, what is that where are we now? We have already assumed I have perturbed the control perturbed control leads to perturb state the perturb state is given by x k plus delta x k. So, Z k must be equal to, we would like to be able to find delta x k we would like to be able to force delta x k such that Z k is equal to h of x k plus delta x k. I can express this in the first order Taylor series living leading to this expression which is h of x k plus the Jacobian of h at x k times delta x k.

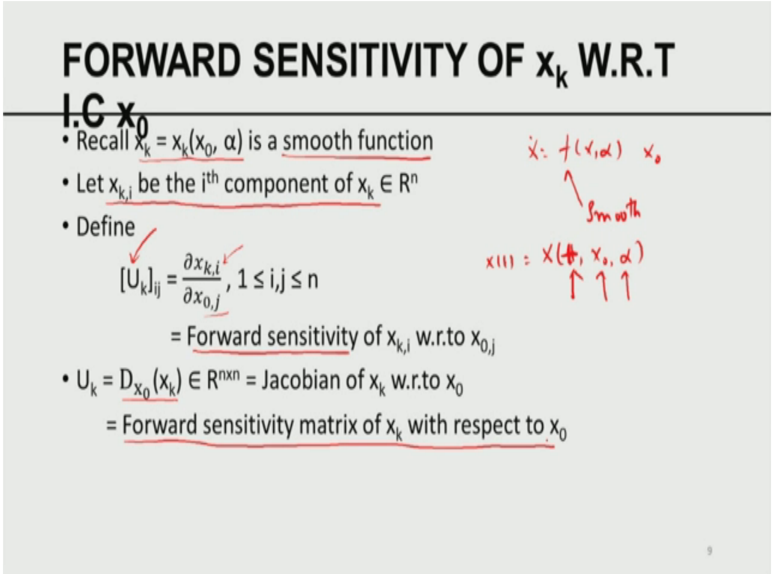
If I using 4 let us say let us go to for a moment now. So, in 4 I have e k of c is equal to h of x k c minus h of x k star c minus V k. So, if I brought that in here, if I brought that in here this part sorry this part which is the increment is given by this equation this is equal to Z minus h of x k Z minus h of x k is essentially minus e k c start up going to equation 4. You can readily see I would like to show it to you once again, equation 4, that is equation 4. So, Z k equation 4 tells you e k is equal to h of x k c minus Z k therefore, Z k minus h of Z is minus e k. So, that is the relation that we have here.

The Jacobian as you know is a matrix of size m by n, e k c is a matrix of is a vector of size m. So, you can think of this like H of x is equal to Z where H is a m by n matrix, x is n by 1 and this is m by 1 that we saw in our lectures on static problems. So, you can see delta x k, x is related replaced by delta x k h is replaced by the Jacobian of H, and Z is replaced by minus e k z. So, once you make this correspondence between the static problem and this problem our job is to be able to find delta x k such that it satisfies this equation. This equation represents an equation of the linear least square type that we have discussed in earlier modules we also know the e k is a random vector it is a its covariance is R k. So, I can determine x k by solving 7 as a weighted linear least square problem, as a weighted linear least square problem. So, let us summarize what we have done.

You run the model forward from an error irenaeus control. I would like to be able to compute or characterized the increment in the state delta x k at time k. So, what is the increment we would need at time k which is delta x k such that it will annihilate the forecast error, such a increment at time k is obtained by solving this linear least square problem as a weighted least square problem because we already know that e k is a random vector whose covariance is a k inverse. So, this is how.

Now, once I characterize this solution of this linearly square problem we already know delta x k is related to delta x naught and delta alpha using forward sensitivities. I am going to now relate delta x k to delta x naught and delta alpha, so that is the next step in our analysis.

(Refer Slide Time: 23:31)



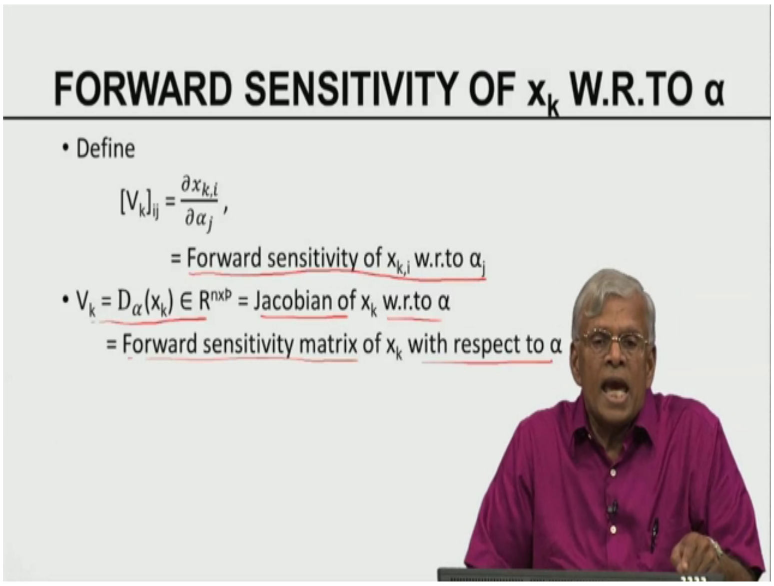
So, recall x k which is the solution at time k is a continuous or a smooth function of x naught and alpha. So, x k depends on x not and alpha smoothly, smoothly means what x k is differentiable with respect to x naught and alpha. Those of you who have taken differential equations you know that under smooth. So, if I have a differential equation x naught is equal to f of x of alpha if x is smooth, smooth in the sense that f has is not only continues, but processes continuous derivative with respect x and alpha of several orders the solution x of t of x naught of alpha this solution is smooth is also smooth it you can you can you can compute the derivative with respect to t you can compute the derivative respect x naught you can compute the derivative with respect to alpha.

That means if f is smooth in a differential equation x naught is equal to f of x of alpha if x naught is the initial condition the solution at any time t is equal to x of t when what we say as x of t is in fact, x of t of x naught an alpha in differential equation theory one can show if f is smooth x is also smooth with respect to x naught and alpha. In the case of discrete time dynamics a similar conclusion also follows if my model map m is smooth the model solution is also smooth with respect to the initial condition and alpha. This smoothness essentially allows us to be able to compute the derivative of x k with respect to x naught the derivative of x k with respect to alpha.

Now, let us denote the ith the component of x ks, x k comma i. I am now going to compute the first order derivative of x k i with respect x naught j. So, this is the jth component of the initial condition this is the ith component of the solution at time k I am going to denote this as U k ij, U k is a matrix i j, i refers to the component at time k j refers the component at time 0 this derivative is called the forward sensitivity of x of x k of i with respect to x 0 of j. So, as you vary i and j in the interval 1 to n you get a matrix that matrix is called U k. So, U k can be thought of is simply as a Jacobian of x k with respect to x naught this Jacobian matrix is called the forward sensitivity matrix of x k with respect to x naught.

So, this is where the forward sensitivity of the solution with respect to x naught comes in. So, what does this say? If you make a small change the initial condition at time 0 what will be the effect of that small change initial condition on the solution at time k that is what this matrix of forward sensitive captures.

(Refer Slide Time: 27:10)

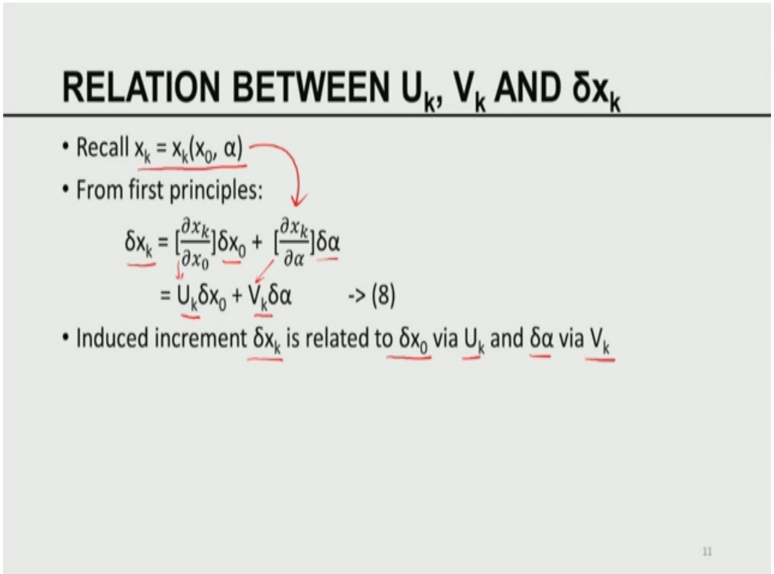


Likewise, I can define the forward sensitivity with respect to alpha. So, V k ij is again the x k i is the ith component of the solution at time k alpha j is the jth component of the control vector. The partial derivative is the ijth element of a matrix V k, so that is called the forward sensitivity of x k i with respect to alpha j. So, you can again think of V k as a

Jacobian of x k with respect to alpha this is called the forward sensitivity matrix of x k with respect to alpha.

So, V k, the matrix V k, the matrix V k is a n by p matrix it represents the forward sensitive to the solution at time k with respect to the p parameters is called parameter sensitivity. U k on the other hand is the forward sensitivity solution with respect to the initial conditions. Both these sensitivities essentially tells you how a small change in the parameter value how your small change the initial condition would be amplified at time k. So, you can think of the sensitivity as a kind of an amplification factor that helps to amplify the initial error.

(Refer Slide Time: 28:32)



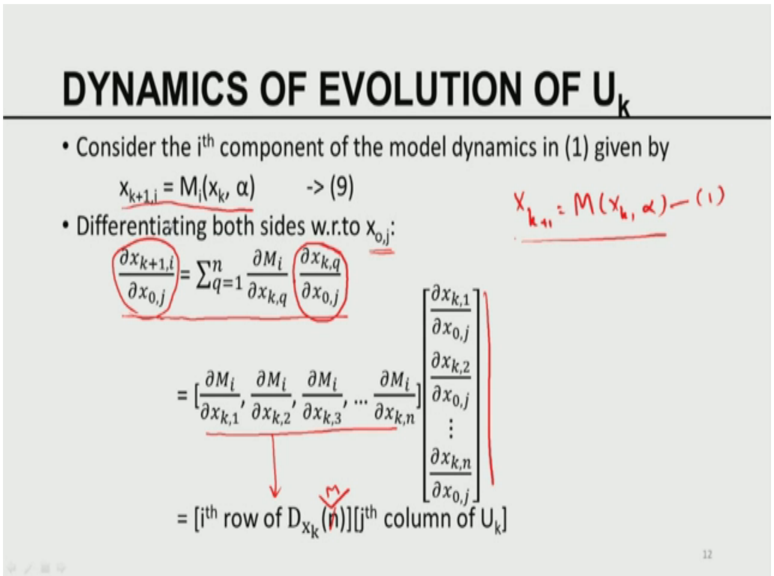
So, having defined the concept of forward sensitivities, now I would like to I would like to use U k and V k to be able to characterize x k. Please remember why are we interested in that we have already said the way to correct the forecast error is to determine delta x k as a solution of a linear least square weighted version of the problem, so that is set aside. Now we have to relate delta x k to delta x naught and delta V k this is where the role of U k and V k comes into being.

So, let x k be the, let x k be the model solution at time k from first principles. The variation, from this equation we can derive this. So, x k is a function of x naught and alpha therefore, the first variation of delta x k induced by delta x naught and delta alpha is essentially given by the product of delta x k the product of the forward sensitivity of x k with respect to x naught times delta x naught plus the product of the forward sensitive x k with respect to alpha times delta alpha is a very basic principle that comes from Taylor series expansion, first variation and the notion of a differentials in calculus. Now we have already seen that this matrix is V k this matrix this U k and this V k. So, this matrix is a n by n matrix this matrix is a n by p matrix therefore, the induced increment is related to delta x naught via U k and delta alpha via V k.

Please understand I need to be able to control, I need to be able to change the control changing the control means it will change the solution. So, everything must be related to time 0. So, this is how we relate the change in the state at time k to the change in the state at time 0 and change in the parameter at time 0.

So, now we have achieved one of the things that we need to be able relate delta x k del naught and delta alpha. But to understand, but to use 8 to characterize delta x k now we need to be able to characterize what is U k and what is V k. U k please remember how the solution at time k varies with respect to initial condition, V k is how the solution at time k varies with respect to the parameters these are called forward sensitivities.

(Refer Slide Time: 31:37)



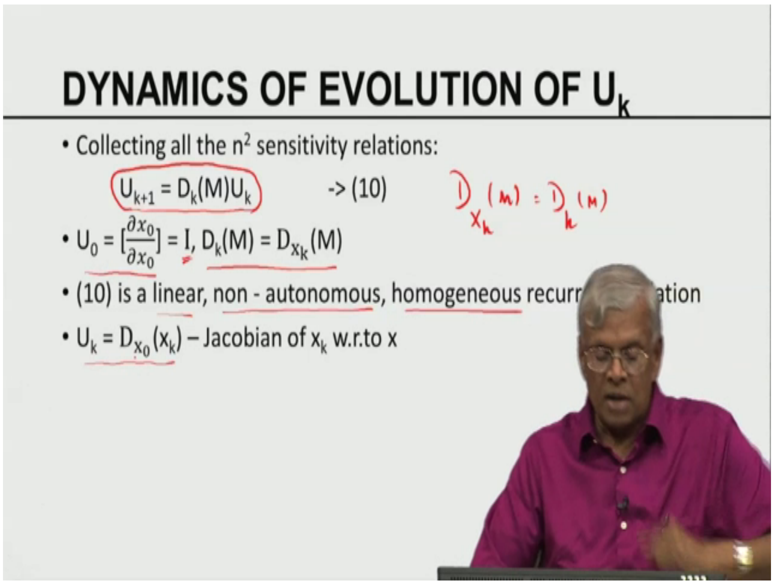
So, we need to be able to explain are we need to be able to find out a way to characterize the evolution of U k and V k please recall these are matrices. So, we have going to collectively talk about the evolution of all these of these two matrices in time. So, the dynamics of evolutional U k is our next topic consider, the ith component of the model dynamics in one. So, please remember the model dynamics is given by x k plus 1 is equal to M of x k of alpha. So, if I consider the ith component on the left hand side I should be able to consider the ith component on the right hand side and that is what equation nine is all about consider the ith component of the equation the model equation in one. So, this is the equation in one which I have written for us to be able to recall.

Now, I would like to be able to differentiate both sides with respect to the jth component to the initial condition. So, the differential coefficient of the ith component at time k plus one with respect to the jth component time 0 that the left hand side, now I can use the chain rule alpha depend does not depend on x naught x k depends on x naught M i depends on x k and x k depends on x naught. So, I have to use the chain rule in standard calculus. So, the left hand side is equal to partial of M i with respect to partial of the qth component of x k and the partial of qth component of x k 2 the partial of jth component of x naught so the product of these two partial derivatives.

Now, q is an arbitrary 1, q can take the value 1 2 3 all the way up to n. So, I have to sum it up over q. So, this is the fundamental relation that relates the dynamics of evolution of the forward sensitivity. Please recall this is the forward sensitivity the ith component with respect to the jth component time k plus 1, time k plus 1 and this is the sensitive of the qth component with respect to the jth component. I can rewrite this in the form of a inner product that is the row, that is the column you can convince yourself very easily. This is the ith row of D k m, this is D k m and this is the jth column of U k that is being, that is a fundamental interest right now, that is a fundamental interest right now.

So, let us go back now. So, one sensitivity component is given by the product of the ith row and the jth column.

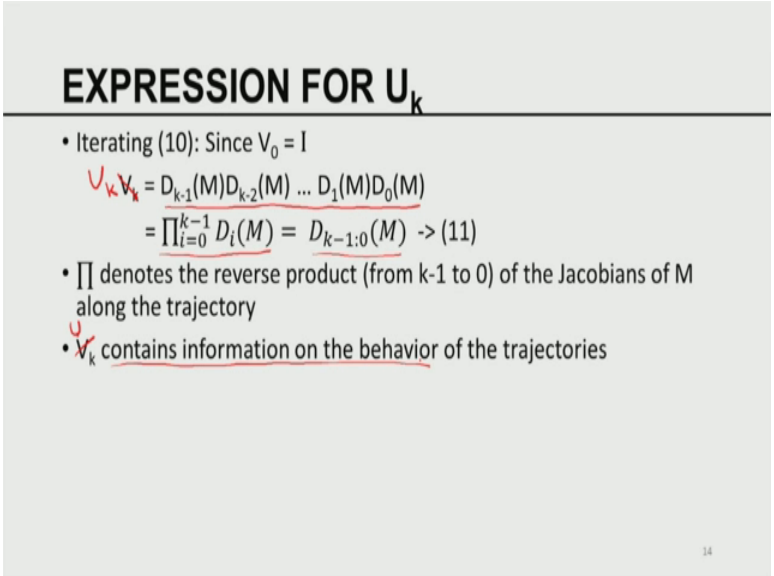
(Refer Slide Time: 34:54)



If I want to consider all the sensitivity matrices simultaneously tthere are n square such sensitivity relations the set of all n square such sensitivity relations is captured in the form of a matrix dynamics. So, U k plus 1 is a sensitivity at time k plus 1, U k is a sensitivity time k U k plus 1, is the matrix U k is a matrix, D k m is the D k m is the sensitivity, I am sorry it is the model Jacobian with respect to x k. So, here I am using a simplified notation I want you to remember that instead of writing d of x k of M we are simply write D k of m, D k of m. So, D k of m is the model Jacobian. So, this is a dynamical equation in discrete time.

I need an initial condition u naught is given by partial of delta x naught with respect to delta x naught that is an identity matrix. And I have already talked about the change in slight change in notation for the better. Now please remember ten is a linear equation ten is a non autonomous equation, it changes along the trajectory it is a homogeneous recurrence relation, it is a homogeneous recurrence relation. Please recall U k is the sensitivity of x k with refer to x naught therefore, equation 10 tells you how the forward sensitivity of the solution with respect to the initial condition evolves starting from the initial value of i. So, I have to compute all these along the trajectory you may recall in the context of in the context of non-linear version of the 4-D Var we talked about the tangent linear system you can think, I am going to challenge the reader to be able to compare this with the tangent linear system and verify this is very similar to the tangent linear system that we talked about in the context of 4-D Var in the context of 4-D Var. So, 10 is a forward dynamics, it is called the forward sensitivity.

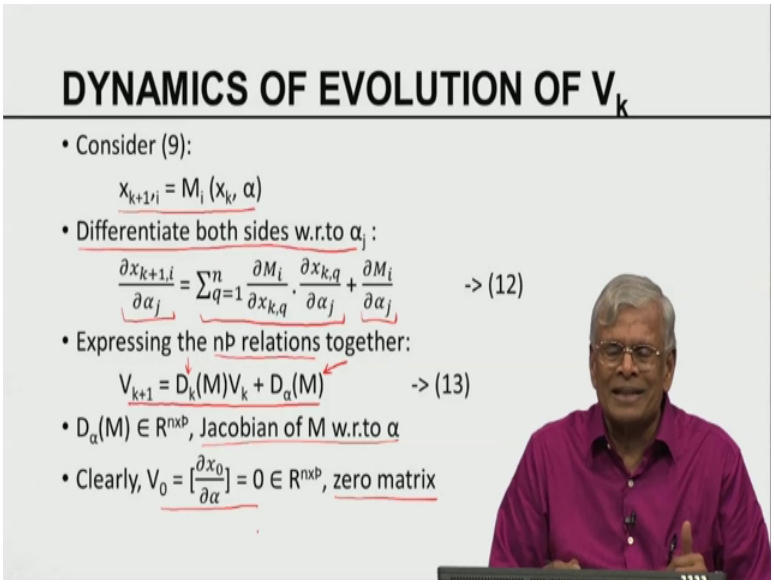
(Refer Slide Time: 37:20)



Now, I can iterate 10 starting from I therefore, V k I can iterate sorry I can iterate 10 sorry this is U k, I can iterate 10 and so U k is the is the given by the product of the Jacobian along the path this can be written as the product succinctly like this which can again using the notation that we have already used is the product of the Jacobian along the trajectory. pi denotes the reverse product from k minus one to 0 of the Jacobian of V m along the trajectory please realize this must be sorry we will correct that this must be U. Therefore U, contains information about the behavior of the trajectories.

In what way does this contain the information with the trajectory? If I perturb the initial condition by a small amount how sensitive the solution is going to be at a future time that is why it is called forward sensitivity. So, we not only derived the forward sensitivity equation 10, but also have solved for the forwards m, found an expression for the forward sensitivity it is simply the product of the Jacobian of the model map along the trajectory.

(Refer Slide Time: 38:54)



Now, we are going to concentrate on deriving a similar dynamics for V k. Please remember V k is the sensitivity the solution with respect to the parameters. Again I am going to consider the ith component of the model equation that was also the starting point previously. Earlier we found the derivative of this with respect to the jth component of the initial condition here we are going to differentiate this with the jth component of alpha.

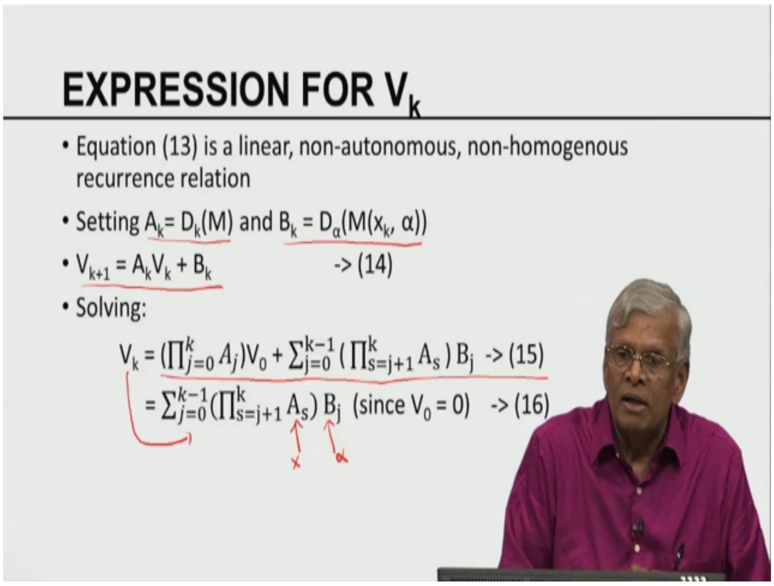
Therefore the left hand side is simply the derivative of this scalar with respect to alpha, the right hand side it is slightly different from the previous derivation x k depends on alpha. So, so M i depends on alpha in two ways M i depends on alpha directly M i also depends on alpha indirectly through x k. So, one is the direct term another is to be obtained by chain rule. This is the term that comes from the chain rule to accommodate to accommodate the implicit dependence, this is the explicit dependence. So, the sensitivity of x k plus 1 the ith component of it with respect to the jth component of alpha is essentially the sum of two terms, the first one is an implicit dependence second one is an explicit dependence.

Again expressing, there i varies from 1 to n, j varies from 1 to p. So, there are n p such relations by collecting all these n p such relations and arranging them in the form of a matrix we get a matrix recurrence. This is the model Jacobian, this is the model Jacobian with respect to parameter with respect to the parameter. So, this is the, D M alpha is the Jacobian of M with respect to alpha.

Now, what is the initial condition from this? Initial condition is dealt is the partial of x naught with respect to alpha. The state or the initial term is x naught alpha is the initial settings of the parameters the initial condition has no bearing on the parameter value and the parameters has no bearings on the initial condition therefore, the initial condition is identically 0, it is a 0 matrix. So, this is the starting initial condition. So, 13 is a recurrence relation that starts from a 0 matrix whereas, in the previous one started from identity as the initial condition.

So, we have derived recurrence relation for the evolution of both the forward sensitivity with respect to initial condition and parameters.

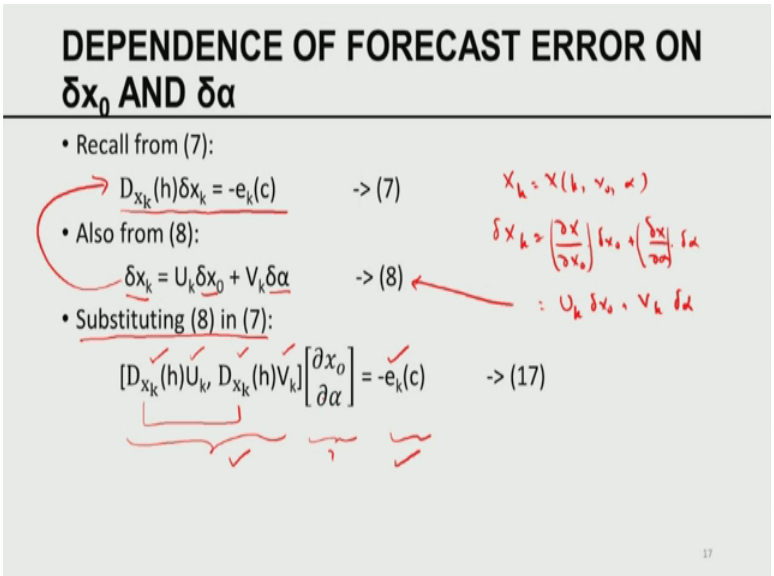
(Refer Slide Time: 41:45)



Equation 13 is a sin is again a linear non autonomous non homogenous recurrence relation. The previous one was linear non autonomous it is homogeneous, now I have a forcing term D M of alpha is the forcing term. We are now going to simplify the solution of this. We are going to change notation let A k be D k M, let B k be that in this case the previous relation 13 can be rewritten as 14 V k plus 1. So, V k plus 1 is equal to A k times V k plus B k.

I can substitute back that is a very good interesting exercise I am going to ask you to do this as an exercise by solving 14 one can verify that the solution V k is expressed as 15 where V naught is the initial condition, but please recall V naught is a 0 matrix therefore, the first term vanishes, the second term is the solution therefore, V k is given by this sums of products of matrices as and bs. As or the model Jacobian with respect to with respect to the state bs are model Jacobian with respect to this is with respect to the state this is with respect to the alpha. So, it is the products of Jacobians model, Jacobians both respect to the state and the parameter the sum that off is a complicated expression much more complicated than the U k, but however, we have an explicit expression for the sensitivity of the solution with respect to the parameters.

(Refer Slide Time: 43:37)



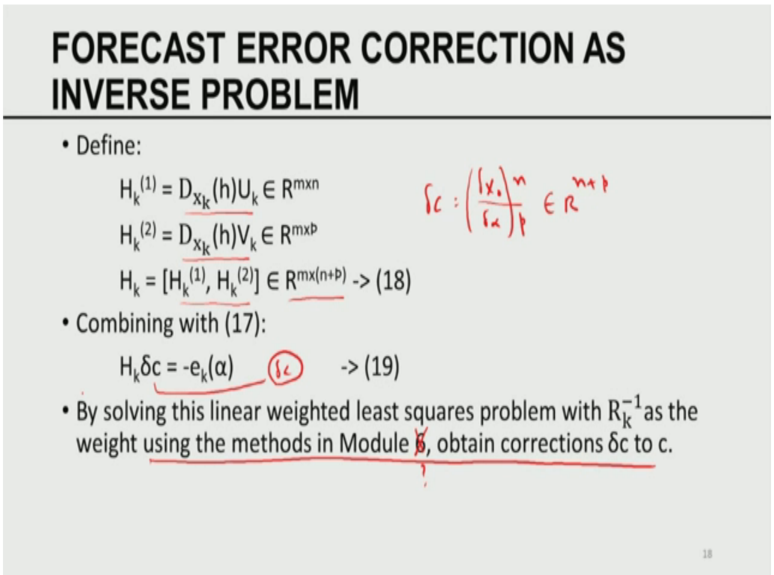
I think we have we are almost there to state the final version of the problem. So, please recall from 7 we know the delta x k that will help to inoculate the deterministic part of the error is given by 7, I have quoted the 7 back again here.

From 8 we know, now let us let us talk about the change. So, x k is equal to x of k of x naught and alpha therefore, delta x k from basic differential equation our basic theory of differentials is that delta x by delta x naught times delta x naught plus delta x by delta alpha times delta alpha. So, this is the matrix, this is the matrix and this matrix is U of k of delta x naught plus this is V of k plus delta alpha and that is what equation 8 is all about. Therefore, we are able to express the x k in terms of delta x naught and delta alpha.

Now, I can substitute this delta x k from here to the equation 7. If I substituted this and simplify a little bit I get D of x k h times U k come on D of x k h x V k times delta x k delta alpha must be equal to minus e k minus e k of c that is equation 17. Now let us look at the left hand side, I know the Jacobian of h I know the forward sensitivity because I have already solved the recurrence relation again these two are the same. So, I know this, this is again computed from solving the forward sensitivity e k is known.

So, now, we can see the left hand side is a matrix that is the vector that is the vector we know this, we know this, we do not know this part, we can essentially see I got a classical linear least square problem. Why this is linear? The unknowns are or the unknowns what are the unknowns delta x k delta alpha they occur to their first degree. Therefore, we have converted the forecast correction problem to one of being able to compute the increments in the initial condition delta x naught and delta alpha and the these increments are going to be decided by the solution of this linear least square problem and we will formulate it as a weighted linearly square problem because in the right hand side is stochastic and I know its variances r k.

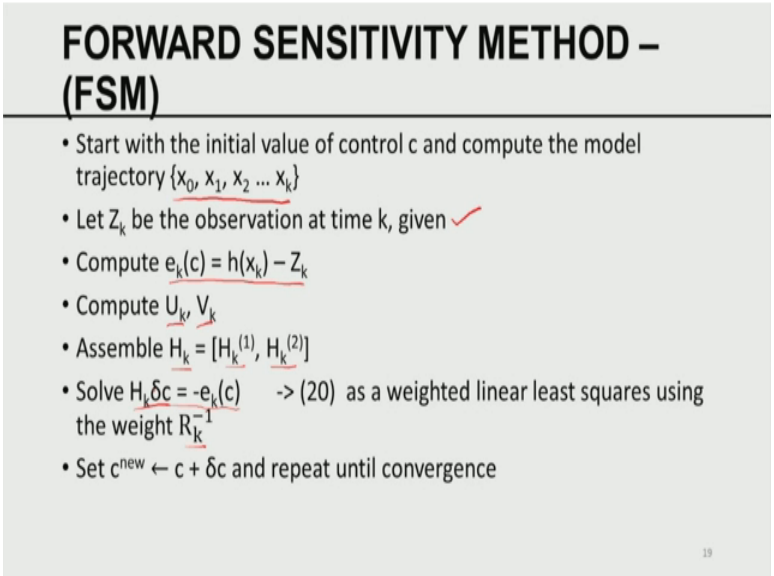
(Refer Slide Time: 46:38)



So, to further simplify the notation, the forecast error correction as an inverse problem, I have already alluded to this and that is what we are going to be emphasizing again. So, I am going to define H k of 1 as the product of these two matrices, H k of 2 as the product of these two matrices. I am going to define a new matrix which is H k which is the two matrices put together side by side. So, this matrix H k is a m by n plus p matrix. So, combining this with 17 via please recall delta x naught and delta alpha is the vector have delta c this is a size n, this is a size p. So, the whole thing belongs to R n plus p, therefore, by solving this linear least weighted least square problem with R k inverse as the weight using the methods that we have already discussed, using the methods I do not think the module 6 is correct I will correct that.

Obtain the corrections to correction delta t to c and I can then compute delta c by the solution that is the whole idea of the forward sensitivity method. Now you can see how the forward sensitivity method gets into the matrix H k on the left hand side of 19.

(Refer Slide Time: 48:13)



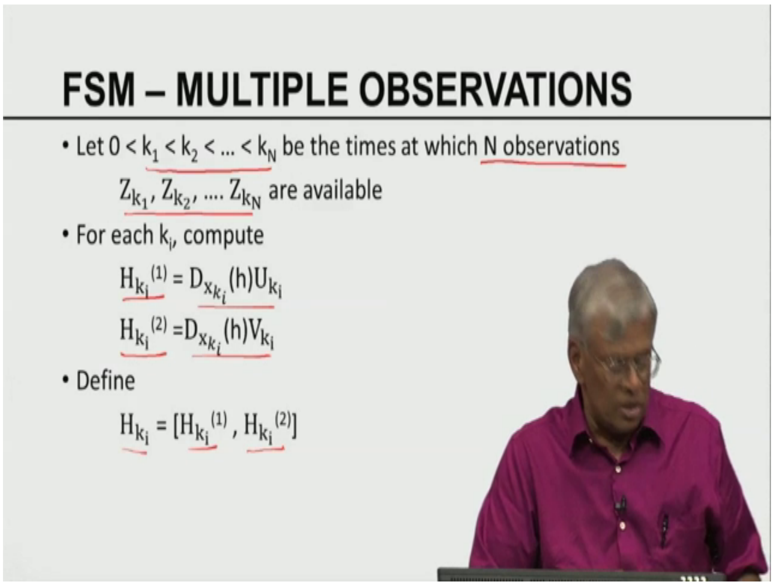
So, I am going to talk about an algorithm which is called the forward sensitivity method FSM is an acronym. Start with the initial value of the control c start with the initial value of the control c, compute the model trajectory x naught through x k let Z k be the observation at time k that is given to you that is that is Z k is given to you compute the error which is h of x k. So, you know h you know x k. So, you know h of x k, h of x k minus Z is the error.

Compute U k V k by solving the forward sensitivity dynamics the matrix recurrence relation assemble the matrix H k 1, H k 2 that gives you the matrix H. Then solve the resulting least square problem as a weighted linear least square problem with the weight R k inverse.

Once you compute delta c you can add delta t to c I get a new control. Since we are have used the first order method well the new control will be better than the old control in the annihilating error you can expect the error will not be annihilated in its entirety because we have used only first order recurrence of first order Taylor series expansion in our theory. Therefore, what is a way to further reduce the forecast error you repeat the whole procedure again starting with c nu and do whatever you did with c. So, if you iterate this several times each time you are going to keep adding increments to the control that continuously keep moving the forecast error towards purely random part it annihilates continuously the deterministic part that is the idea, that is the idea.

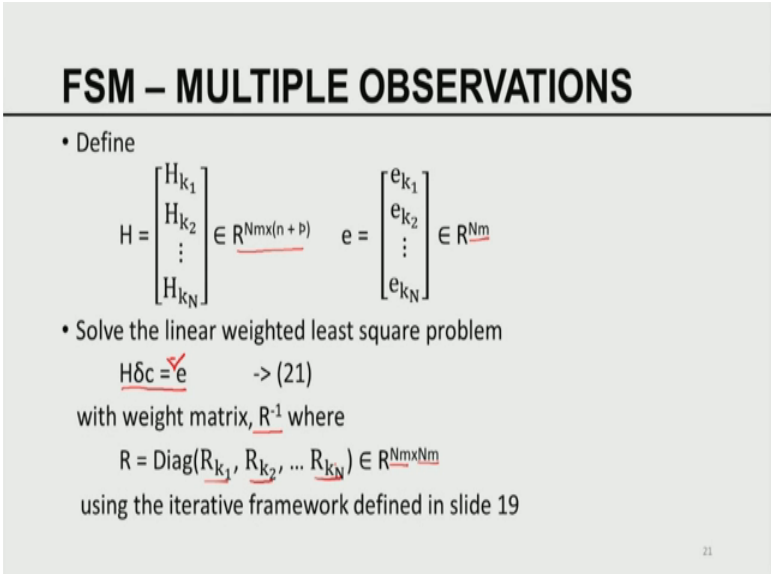
So, in all these discussions we essentially said if I had only one observation how I can use one observation to be able to compute the control.

(Refer Slide Time: 50:37)



In principle you may have multiple observations. So, I am going to provide a quick extension of this let there be N times, let there be N observations the observation that these N times be given by Z k 1, Z k 2, Z k N. For each k I now I simply need to repeat what I did for one observation to each of these observations and then collide them all together that is all the idea there is nothing more because each observation is going to give me give me information about the increment. So, I have to repeat as many times as I had. So, I need to be able to compute H k i 1, earlier I had H k 1 here I have H k i 1, H k i 2 which is given by this which is given by this U k i, V k i are the sensitivities of the solution at time k, x k i h, x k i h are the Jacobian the forward operator at time k I therefore, H k i is the concatenation of these two matrices side by side.

(Refer Slide Time: 51:53)

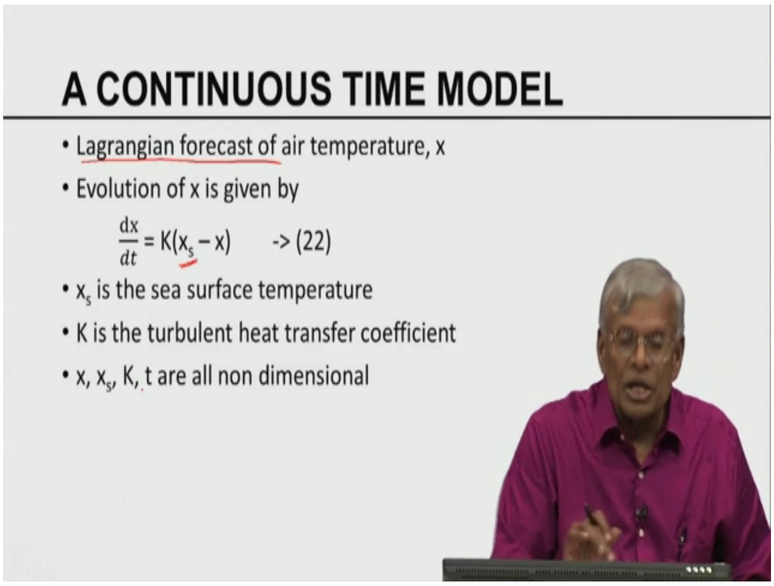


Now, for each time k i, I have this matrix, now I have N such times now I have to prepare another gigantic matrix. So, create a newer metric h where I stack H k 1, H k 2, H k N, this matrix is of size or Nm cross n by p likewise I also stack all the errors forecast errors at time k 1, k 2 k N and that is going to be giving you a large vector of size Nm. So, if you pull them altogether I have now a new linear least square problem please remember delta c has not changed only the matrix h has changed only the vector e has changed, in this case I have I think it should be minus e minus e.

I can now solve this as a weighted linear least square problem the weight matrixes R inverse. R inverse is simply a diagonal matrix, R inverse is a matrix of size [Noise], cross Nm. So, each of the yeah each of the N matrices are put along the diagonal of a matrix R this R will be a collection of all the covariances of observations at each time, so is a gigantic matrix. And you can formulate this as a linearly square problem since we have solved the static linearly square problem weighted unweighted versions, in detail in previous modules we are not going to describe that part. But one can see that if you know the linear least square static problem you can solve this dynamic problem simply an application of those concepts here.

So, what is the interesting thing here is that even though it is a dynamic data assimilation problem we convert the dynamic data assimilation problem to a static deterministic inverse problem and that is the essence of forward sensitivity method.

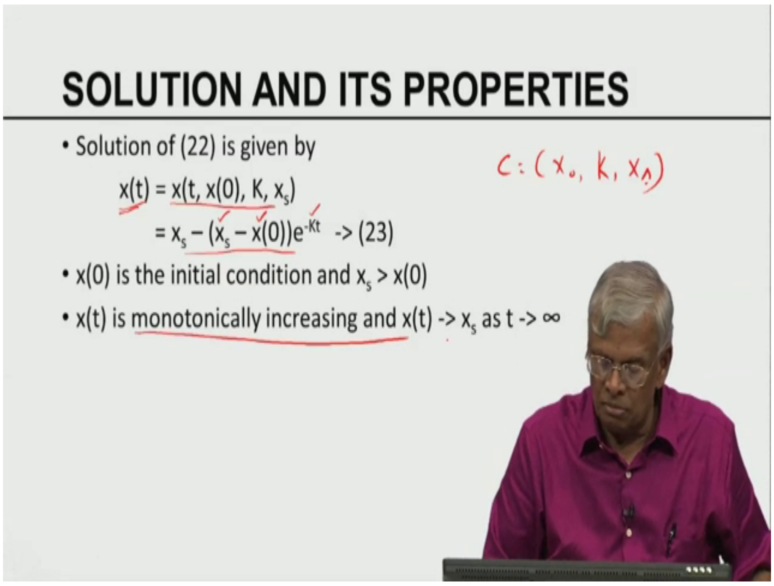
(Refer Slide Time: 54:09)



We can also talk about a continuous time versions we simply for the sake of simplicity used the discrete time version. Now I am going to present a quick example of the forward sensitivity method using a continuous time model. This is a model that describes a motion of cold air over a hot ocean. So, this is called the Lagrangian forecast in meteorological circles. The air temperature is x, x s is the temperature of the sea surface x s is greater than. So, in United States we have the gulf of Mexico during wintertime cold air sweeps from the north the waters of the gulf are very warm in October November, it holds lot of the heat from the summer. So, when the cold air moves over the warm gulf there is a heat transfer from the warm gulf waters to the cold air. So, the air becomes warmer and warmer.

So, I am now the equation 22 gives you the dynamics of evolution of the temperature x of the air column k. So, x is the sea surface temperature k is the turbulent mixing are a turbulent heat transfer coefficient and we are assuming x k k t are all non dimensional in other words originally I had a dimensionalized version assume that I have non-dimensionalized it we have already done that. So, this is the non dimensional version of the model equation.

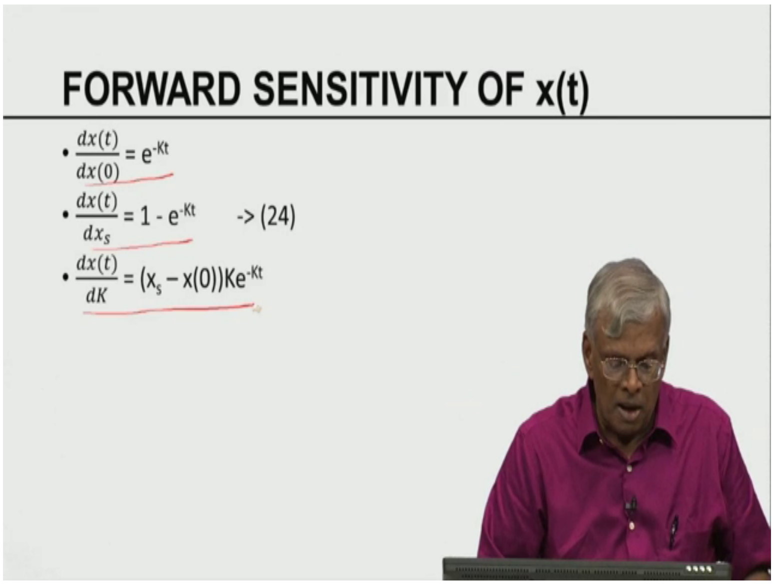
(Refer Slide Time: 55:59)



The solution of this model can be given explicitly because simply a first order differential equation we can readily solve from standard experience in calculus. So, this is the solution this is the form of the solution x naught is the, you can see the solution depends on x naught the solution depends on x s which is called the boundary condition and the solution also depends on the parameter K. So, the control here c consists of x naught, K and x s.

So, the control has three components one is the state of the system one is the initial state of the system another is the parameter, another is the sea surface temperature. The solution is monotonically increasing and x t tends to x s as t goes to infinity; that means, as the water are as the air moves over the water the heat transfer takes place the air temperature rises the heat transfer stops when the air temperature is equal to the sea surface temperature that is this very simple dynamics.

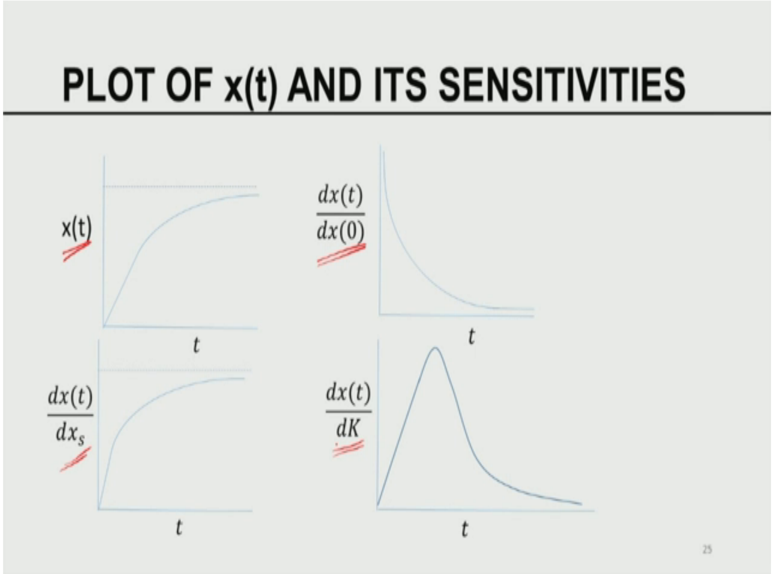
(Refer Slide Time: 57:09)



So, because the solution is known I can compute the sensitivities rather explicitly, I do not have to use the differential equation to compute the evolution of the sensitivity. So, that is the initial condition sensitivity, boundary condition sensitivity, parameter sensitivity.

So, once I know the sensitivity I am ready to compute the matrix h, but before that I need to be able to compute the observations or have the observations. So, using the observations starting from the initial condition make the forecast compute, the errors compute the forward sensitivity, you know the error using the forward sensitivity I can compute the matrix h we got the standard linearly square problem.

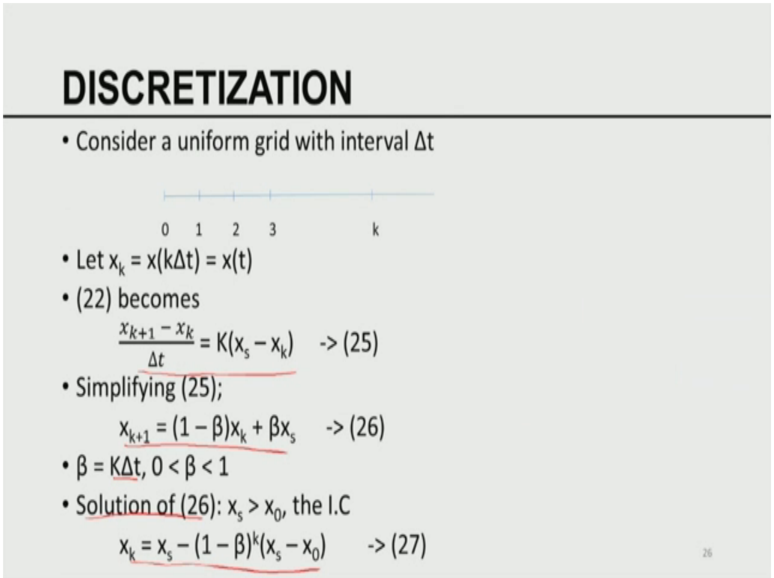
(Refer Slide Time: 57:52)



Here are the examples for this model the model solutions this is the model solution x of t, this is the model solution this is the sensitivity the model solution with respect to x naught, this is the sensitivity of the model solution with respect to the boundary condition, this is the sensitivity of the model solutions with respect to parameters.

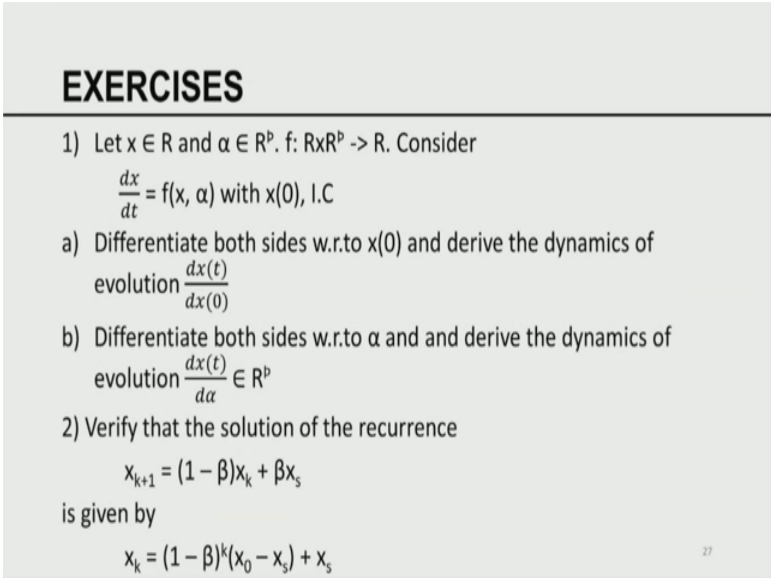
So, you can see solution and three sensitivities are graphically given here.

(Refer Slide Time: 58:21)



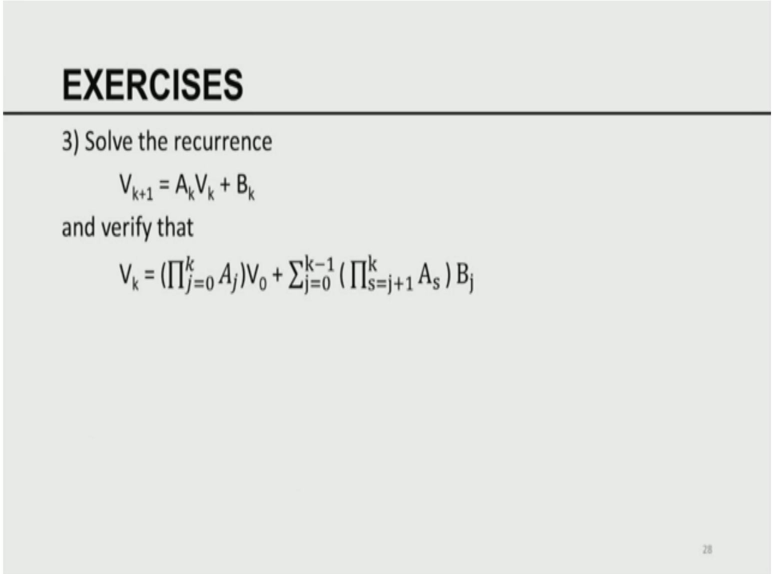
I am now going to discretize the model equation using standard Euler scheme this is the standard Euler scheme. So, if I simplify this 25 I get this equation. This is again a linear equation because we started with a linear model beta is k times delta t, delta t is the time interval for time discretization. I can solve this model equation explicitly and this is the discrete version of the model solution that is given in 27 we have already given the continuous time version of the model solution earlier.

(Refer Slide Time: 59:10)



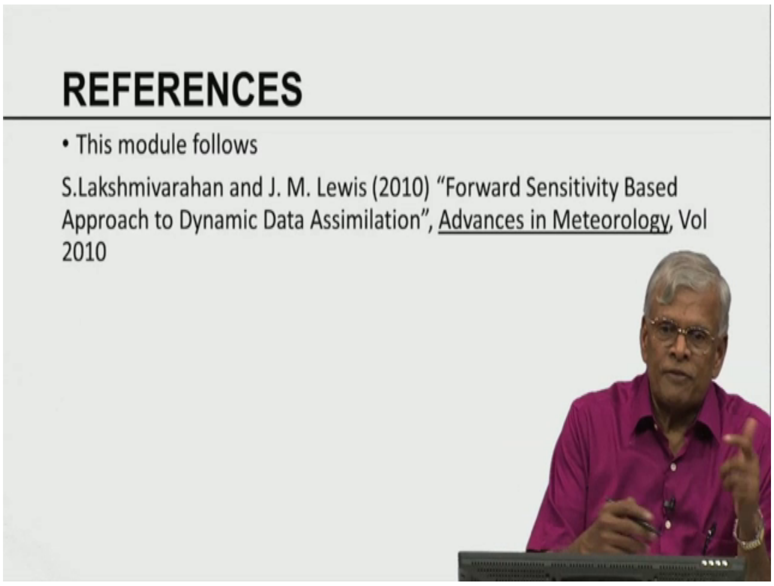
So, what is that we are going to ask you to do? We are going to ask you to be able to with that is what problem two is all about, verify that the solution of the recurrence is given by this. We also want you to be able to solve the recurrence relation that are given in the previous slides when we did that.

(Refer Slide Time: 59:22)



I also want you to be able to derive for the continuous time dynamics that is problem number 1. I would like you to be able to differentiate the continuum dynamics and derive the dynamics of evolution for the forward sensitivities both with respect to the parameter and the initial conditions.

(Refer Slide Time: 59:49)



This FSM method was developed by as it was published in a paper in 2010 entitled Forward Sensitivity Based Approach to Dynamic Data Assimilation which appeared in, Advances in Meteorology, volume in 2010. In this paper we describe both continuous time as well as a discrete time. The example that I talked about we discuss in complete detail, we would like to refer the reader to this paper and I would also very strongly encourage you to be able to reproduce the results in this paper to make sure that you have a total and a thorough understanding of this methodology.

With this we conclude our discussion of data simulation using FSM forward sensitivity method. You can really see this is another way to do this. In our next lecture we will talk about the relation between 4-D Var and FSM. I strongly encourage you to be able to perform a data assimilation using it using the forward sensitivity. How do we do that? You assume certain values of the parameters, you generate the solution, you generate observation and in change the initial condition create erroneous forecast then use the generated observation to be able to control the erroneous forecast and that is what is called a twin experiment which we have been doing all along. Why? What is the best way to be able to benchmark your algorithm is to be able to generate data artificially using the model and if the model, if the method is capable of being able to recover the true initial condition then you have shown the proof of concept using these artificially designed observations that essentially gives you a clue that it should also work in real situations. So, that is the basic idea of the whole approach.

With this we have completed the discussion of FSM. We will in the next lecture talk about the relation between FSM and 4-D Var.

Thank you.