

**Dynamic Data Assimilation**  
**Prof. S Lakshmivarahan**  
**Department of Mathematics**  
**Indian Institute of Technology, Madras**

**Lecture - 37**  
**Ensemble Reduced Rank Filter**

In this module 8.4 we are going to be looking at strategies for doing non-linear filtering, approximate non-linear filtering, using the so called ensemble or reduced rank approximations. So, this has come to be called ensemble or reduced rank filters. Let us quickly review where we are.

For the LQGA problem, with the system is linear observations or linear functions of time, when we are using least square criterion, and when all the distributions are Gaussian, we could derive the exact filter equation for the mean and the covariance of both the forecast and the analysis, but when either the system is non-linear or the observations are non-linear are, both are non-linear, we are going to have difficulties. We saw the difficulties in the variety of ways.

The difficulty essentially arises from the fact that, Gaussian all the other distributions, other than Gaussian distributions are not, are cannot be specified simply by the first 2 moments. So, if you get. So, our aim in a non-linear filtering is to work in the infinite dimensional space of distributions.

So, what is the forecast distribution, what is the analysis distribution, how do the forecast distribution analysis distribution interact with each other, how do we update these 2 distributions. We derived sequential updating scheme for updating the forecast filter at, the forecast distribution as well as analysis of the distribution, exactly is in base framework. Much like we did in the linear framework. So, what is the difference? The linear framework they are finite dimensional. There are only 2 variables mean and the covariance.

But in the non-linear filter is infinite dimensional, because we are trying to give an updating scheme for the distribution themselves. So, given the attendant numerical, it is not that we do not know how to solve non-linear problems. We know theoretically how to solve it, we know the exact solution. So, what is the problem? What is the challenge?

It is very difficult to be able to compute, the exact forms of these distributions, they can only be handled numerically given this difficulty, we then move down to computing approximate moment, more approximate moment dynamics.

So, we would derived expressions for the evolution of the mean, and the covariance. In that junction we met with the so called closure problems which are typical of any and every non-linear systems, and as a consequence we saw zeroth order filter, which is depends on linear approximation.

we talked about first order filter, which is which has come to be called the external Kalman filter, and we also talked about second order filter which is a little bit more accurate than the first order filter. While these are meaningful approximations have been used in several walks of life. These approximations are computationally, no less severe for geophysical applications.

Because in geophysical applications, the size of the problem easily of that of million. So, we have to deal with vectors and matrices which are of size million or more, with the ever increasing interest in finer grid. The number of grip points in a typical global model may reach a billion very soon.

And so, the desire to have a more accurate representation using a computational grid with a smaller grid size, is becoming one of the challenges and these challenges cannot be meted out, until I am we have a powerful computers. So, given all this challenges all around us. The reduce ranked filters based on ensemble idea, which is essentially a Monte Carlo idea, has become very popular, it is very simple, you do not need to write on a joint, you do not need to solve any least square problems.

The basic idea is, you have a model code, if you have a multiprocessor, you pick several initial conditions from an appropriate distribution, run the model forward in parallel, and once you spit out the forecast from say 50 models or 50 model runs from 50 model runs of the same model code, but starting from different initial conditions. So, you will get the forecast ensemble, which is generated from an initial ensemble.

Once the forecast ensemble is generated, the forecast mean and the forecast covariance are essentially calculated as a sample mean as a sample covariance of this forecast

ensemble. After having created the forecast ensemble, we then integrate the observation with the forecast ensemble to create an analysis ensemble.

And from the analysis ensemble we again start the model, moving forward creating a forecast ensemble. So, this notion of being able to create a forecast ensemble, from there an analysis ensemble, and starting the model forward again from the given analysis ensemble, is a sequential way of thinking. So, we are still remaining within the sequential framework. The only difference is that, we never look back. We simply keep moving forward. Moving forward not on one simple model run, but per ensemble of model racks. So, that is the basic idea. there does the reduced rank come from. The given size of the problem, let us say million for example, let us fix it. So, in order to be able to ensemble based computation of covariance matrices which are full rank.

I may need to run of the order of million ensemble members. Running a million ensemble members is impractical in today's technology. So, what do they do? They pick a small sized ensemble 50, 100, 200. So, by creating 200 ensemble members of a model run I am going to basing simple statistical principles to be able to estimate, the mean of 50 ensemble members, and the covariance using 50 ensemble members. The basic statistics tells you, if you have the number of samples of the order of 50, if you are trying to compute the covariance matrix. The maximum rank of the covariance matrix can be no more than 50.

So, we are trying to approximate a matrix of rank million by a matrix of rank 50, or 100 or 200; that is what is called the reduced rank approximation. So, if we can capture some of the important modes of behaviour of the model. Maybe we can very carefully approximate the forecast covariance analysis, analysis covariance, analysis mean forecast mean and keep going forward.

So, this is a class of approximation, which is computationally simple, which is scalable. The scalability largely depends on the size of the problem, and the power of the computers. So, because of this flexibility, this class of ensemble based methods have become very popular in many of the operational centres around the world. So, in this module, we are going to provide a very broad overview of the current methodology, that has come to be known as ensemble method or reduced rank filters.

(Refer Slide Time: 08:56)

### A BASIC IDEA

- Basic result:  $f(x)$  density with:  
mean =  $\mu$ , var =  $\sigma^2$
- $x_1, x_2, \dots, x_N$  independent samples from  $f(x)$

$$\bar{x}(N) = \frac{1}{N} \sum_{i=1}^N x_i$$
$$s^2(N) = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x}(N))^2$$

are unbiased estimates of  $\mu$  and  $\sigma^2$  respectively

*Handwritten notes:*  $x \in \mathbb{R}$ ,  $f(x)$  is a box,  $x_i$  is a value.

So, what is basic idea? Let the  $f(x)$  be the density of a random variable  $x$ . So,  $x$  is a vector. Let  $x$  be a vector in  $\mathbb{R}^n$ ,  $f$  of  $x$  is the probability density. Let the mean of the random variable be  $\mu$  and the variance of the random variable. Oh did I say  $f$  of  $x$  is equal to. No we will we will we will simplify this. Let us assume we have only a real random variable to start with. Let us assume I have a real random variable  $x$  and  $f$  of  $x$  be the density function for the real random variable.

With mean  $\mu$  and variance  $\sigma^2$ . Let  $x_1, x_2, \dots, x_N$  be the  $N$  IID samples, what is IID sample independent samples? There are drawn from the same distribution  $f$  of  $x$ . There in the distribution remains the same, I am simply. So, you can think of the distribution as a black box. You ask for a number it will give you, it will give you  $i$ .  $i$  running from 1 to  $N$ .

So, you can draw  $N$  such sample from the same box that represents the distribution  $f$  of  $x$ ; that is called IID samples. So, if I have  $N$  members of the IID sample, I can compute the mean I can compute the sample covariance, and from our class on statistical estimation, we have already know we have already shown that this estimator for the mean and this estimator for the sample variance are unbiased estimates. So, this is a  $\bar{x}(N)$  is an unbiased estimator of  $\mu$ , and  $s^2(N)$  is an unbiased estimate of  $\sigma^2$  very simple basic statistics. This is the first one you have probably learned in any coercion fundamental statistics.

(Refer Slide Time: 10:53)

### A BASIC IDEA PROPERTY

- $$\left. \begin{aligned} \text{Prob}[|\bar{x}(N) - \mu| > \varepsilon] &\longrightarrow 0 \\ \text{Prob}[|s^2(N) - \sigma^2| > \varepsilon] &\longrightarrow 0 \end{aligned} \right\} \text{ as } N \rightarrow \infty$$
- $\bar{x}(N)$  is a r.v. where distribution is centered at  $\mu$
- $$\text{Var}[\bar{x}(N)] = \frac{\sigma^2}{N} \rightarrow \text{std. error} = \frac{\sigma}{\sqrt{N}} \propto \frac{1}{\sqrt{N}}$$
- $$\text{Var}[s^2(N)] = \frac{2\sigma^4}{N-1}$$

What are the basic properties of these estimates? The probability that the estimate of the mean using  $N$  samples, differs from the true mean by a magnitude more than epsilon tends to 0 as  $N$  tends to infinity. So, what is it mean? The sampling distribution of  $\bar{x}_N$ ,  $\bar{x}_N$  is a random variable, it is an estimate.

Estimate is a random variable. The estimate has a distribution. The estimate has a distribution whose mean differs from the original mean  $\mu$  by a smaller and smaller and smaller quantities as  $N$  the number of samples goes to infinity; that is what is called the consistency condition that we have seen earlier.

Likewise if I consider the estimate of the sample covariance, the sample covariance estimated using the formula in the previous page, differs from the true variance by a quantity epsilon. The probability of that event happening goes to 0 as  $m$  goes to infinity. So, what is it mean? Both the sample mean and the sample covariance become closer and closer, ever closer to the true mean and the variance, as the number of samples goes to infinity.

These are very very simple fundamental fact. What is another way of saying this? The variance of the sample mean. So, sample mean is random therefore, it has variance, the sigma square  $N$ . So, sigma square  $N$ , is the square. Sigma square  $N$  is the variance of the sample mean and that goes to zero as  $N$  goes to infinity; that is one of the statements that is captured in the first line. The second line is also captured by this expression for

this time, and the variance of the estimate of the sample covariance, you can see as  $N$  goes to infinity, these 2 goes to 0. The standard error in the estimate is equal to.

I should say the standard error is equal to, the standard error is equal to  $\sigma$  by  $N$  which is the square root of the variance, and that is proportional to. So, I will simply say the following; that is proportional to  $1$  over  $R$ ; that is proportional to  $1$  over  $R$ . So, what is it mean. The standard error which is the square root of the variance is  $\sigma$  over  $N$ ,  $\sigma$  over  $N$  is proportional to  $1$  over  $N$  and  $1$  over  $N$  goes to infinity as  $N$  goes to infinity.

So, that is another way of saying these things. In fact, many of the fundamental ideas of ensemble filters rests on this consistency criterion of basic estimation, of both the mean and the variance. So, what is the only difference, in the context of meteorological system we are going to be concerned with random vectors, instead of random variables, but whatever holds for random variables, also holds the random vectors, and that is what we will exploit.

(Refer Slide Time: 14:15)

### A BASIC IDEA

- Normal vector:  $x \sim N(\mu, \Sigma)$
- 1.  $\Sigma = LL^T$
- 2.  $y \sim N(0, I)$
- 3.  $x = \mu + Ly$

$y = L^{-1}(x - \mu)$

$\Sigma = [\Sigma_{ij}] \quad \Sigma_{ij} = \text{Cov}(x_i, x_j) \neq 0$

$x \in \mathbb{R}^n, \mu \in \mathbb{R}^n, \Sigma \in \mathbb{R}^{n \times n}$

$$E(x) = \mu + LE(y) = \mu$$

$$\text{COV}(x) = E[(x - \mu)(x - \mu)^T]$$

$$= E[Ly(Ly)^T]$$

$$= LE(yy^T)L^T = LL^T = \Sigma$$

$x \rightarrow y$

IS CALLED

WHITENING

TRANSFORMATION

Another tool that we need is the following. Suppose  $x$  is a random variable,  $x$  is a normal random variable with  $\mu$  as the mean and  $\sigma$  as the covariance. I can do a transformation, what is the transformation. First compute the Chomsky factors of  $\sigma$ , which is  $L L$  transpose. Please remember we have already talked about.

Cholesky decomposition the Cholesky factors are also called square roots. So, let us  $\Sigma$  is equal to  $L L^T$ , where  $L$  is called the square root of  $\Sigma$ ,  $L$  is also a lower triangular matrix, we have already given an algorithm, when we were dealing with numerical methods for solving deterministic, static deterministic inverse problems, we had given the algorithm for Cholesky decomposition in great detail. So, given  $\Sigma$  I should be able to find the  $L$  very easily.

Now let  $y$  be a standard normal variable, let  $y$  be a standard normal variable. So, in this case what is  $x$ ,  $x$  is a vector,  $\mu$  is a vector,  $\Sigma$  is a matrix,  $L$  is a lower triangular matrix which is a square root of  $\Sigma$   $y$  is a standardized normal variable, which has mean zero and identity as the identity as the covariance. Now I can relate  $x$  and  $y$  through the relation through the relation 3.

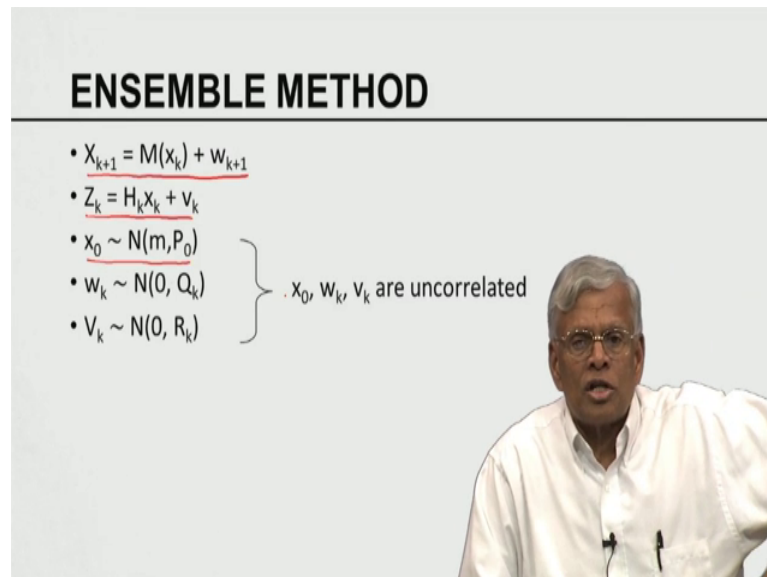
So, what is it say  $y$  is equal to  $L^{-1}(x - \mu)$ , or  $x$  is equal to  $\mu + L y$ . So, this transformation relates a unit normal random vector to a general normal vector with a mean  $\mu$  and covariance  $\Sigma$ , with mean  $\mu$  covariance  $\Sigma$ . And the verification of the third statement is given in the box on the right hand side, you can easily verify that. So, what does this tell you, if somebody gives you a random variable?

How are we going to be using this. We are talked about what is called a whitening filter. What is the whitening filter? In general  $\Sigma$  is a matrix, with element  $\Sigma_{ij}$ ,  $\Sigma_{ij}$  what is the value of that. It is the covariance of  $x_i$  versus  $x_j$ . So, in principle this need not be zero. Therefore, if I have a vector  $x$  whose mean is  $\mu$  and covariance is  $\Sigma$ , the elements of the covariance matrix may be correlated, and the correlation is given by, or may be correlated. The covariance between them is given by  $\Sigma_{ij}$ , but when this is a covariance matrix is  $i$  the diagonal elements are zero.

So,  $y$  is a random, is a Gaussian random vector whose components are uncorrelated,  $x$  is a Gaussian random vector whose components are correlated. So, 3 gives you a transformation, from a correlated random vector to uncorrelated random vector, on uncorrelated another to a correlated random vector. So, the transformation from  $x$  to  $y$  is called whitening transformation,  $x$  to  $y$  is called whitening transformation. Why this is called whitening transformation. White noise is does not have any correlation. So, the  $y$  does not have ah,  $y$  has components with which are uncorrelated. So, considering a given correlated vector how do you convert it with uncorrelated vector.

We have utilized this way turning transformation, in the context of pattern algorithm, where we did the square root version of the, square root version a covariance square root version of the unsolvable term, in a couple of days, in a couple of classes ago, in the same module.

(Refer Slide Time: 18:49)



**ENSEMBLE METHOD**

- $X_{k+1} = M(x_k) + w_{k+1}$
- $Z_k = H_k x_k + v_k$
- $x_0 \sim N(m, P_0)$
- $w_k \sim N(0, Q_k)$
- $v_k \sim N(0, R_k)$

}  $x_0, w_k, v_k$  are uncorrelated

Now, with these two, this is all what you need in principle, to be able to do an ensemble method. So, let us assume I have a model, stochastic model. So, here I am assuming, I have a stochastic model; that means, I have a model core. Somebody has develop the model core. I have an observation;  $x$  naught has the standard 1, you can say  $x$  naught is equal to is normally distributed.

The mean  $m$   $N$  covariance  $P$  naught  $w_k$  is mean 0 and covariance  $Q_k$  and  $v_k$  is mean zero and covariance  $R_k$ . We also assume  $x$  naught  $w_k$  and  $V_k$  are uncorrelated. So, these are the properties of the model. This is also properties of observation, and this is the basis you need to be able to give all this information to do any stochastic data assimilation scheme in particular Kalman filter. So, this corresponds to, the following, the model is non-linear, I have simply assumed the observations of linear functions. I could have assumed the observations also non-linear. Simply I am giving a mix of many things linear with linear, non-linear with non-linear, non-linear with linear.

So, lots of combinations of choices between models and observations.



(Refer Slide Time: 20:08)

### INITIAL ENSEMBLE

- $x_0 \sim N(\overset{m_0}{\mu}, P_0)$ ,  $P_0 = S_0 S_0^T$
- Let  $y_0(i)$ ,  $i = 1, 2, \dots, N$  be  $N$  samples from  $N(0, I)$
- Define initial ensemble
 
$$\hat{\xi}_0(i) = m_0 + S_0 y_0(i) \quad i = 1, 2, \dots, N$$

$$\hat{x}_0(N) = \frac{1}{N} \sum_{i=1}^N \hat{\xi}_0(i)$$

$$= m_0 + S_0 \hat{y}_0(N) \rightarrow m_0$$

$$(\hat{y}_0(N) = \frac{1}{N} \sum_{i=1}^N y_0(i) \rightarrow 0 \text{ as } N \rightarrow \infty)$$

$y_0(i)$  — Ensemble

TIME

$X(i)$

$N(0, I)$

15:15N

$y_0(i)$

So, how does the ensemble method starts. So, I have a model, I have observations I have the underlying properties, I have the basic facts already described. So, what do I need initially I am given the initial distribution  $x$  naught. I am sorry initial distribution of  $x$  naught, which is  $m$  mean and  $P$  naught. So, first do a Chomsky factorization of  $P$  naught which is  $S$  naught,  $S$  naught transpose. Please remember Chomsky we have already seen. Let  $y$  naught  $i$ ,  $i$  running from 1 to  $N$  be the  $N$  samples drawn from the standard normal distribution. Look at this now.

There are 2 indices now coming in here;  $y_0$ . This is the time index. We have to be very careful, and this is  $i$ , this is the ensemble index. Then we will have a representation for the forecast or the analysis. So, you can see I need time, I need analysis of forecast, I also need to keep track of which member the ensemble I am going to be dealing with. So, each of this variable will be loaded, there is a time index, there is an ensemble index, and there will be an index, or a notation for whether I am concerned with forecast or analysis. Once you separate all these things keep in your mind, how a particular notation is formulated.

In our rotation, time is always subscript. Ensemble index I put it in the stomach I wanted to go back. If I have available  $x$  one of the different ways of notating, I can say  $x$  of  $t$ , that stomach. I can put it in the leg that is subscript. I can put it in the head that is forecast. So, instead of  $t$  I will change it to. I am sorry instead of  $t$ , I will change it to  $i$ .

So, this is the  $i$ th member of the ensemble at time  $k$ , but it is a forecast time. So, there are 3 ways of notating to be able to distinguish different quantities. So,  $y_i$   $i$  running from 1 to  $N$ , or  $N$  samples drawn from a box. The box is normally distributed.

So, I am going to get  $y_i$ ,  $y_i$   $i$  running from 1 to  $N$ . So, so I am, I have now supply  $N$ . what is the  $N$ .  $N$  in principle could be million, if the if the vector  $x_i$  is of size million, but even though the vector size may be million, I do not have the luxury of being able to pick large samples.

So, we cannot do a large sample theory, we have to do finite sample theory, small sample theory that is what we are going to be talking about now. So, what did I have I have the initial distribution. I have  $N$  members of the samples driven from a standard normal, and now going to convert the standard normal, ensemble members, to the ensemble members from the initial distribution, using the previous slide the connection between  $x$  and  $y$  and  $x$ .

So, now I am going to introduce another symbol. So, we will use  $x$  for the state. We are going to use  $\psi$  for the ensemble. So,  $\psi_0$   $i$  hat. What is it mean? This is the at time 0  $i$ th the member, the initial ensemble, and also that is the hat represents. it is the initial analysis ensemble. So, that is equal to  $m$  naught. So, what is  $m$  naught. I am sorry, the I am going to say this is  $m$  naught. It should be consistent with this.

So, let that be  $m$  naught,  $m$  naught, the mean of the sample plus  $S$  naught is the Chomsky factor times  $y_i$ . Why it does this make sense. Please go back to the slide here. The equation three, what does it say.  $x$  is equal to  $\mu$  plus  $Ly$ . If  $y$  is from unit normal in order that  $x$  is from normal with mean  $\mu$  and variance  $\sigma^2$ , I need to be able to transform  $y$  into  $L$  using an affine transformation  $\mu$  plus  $L y$ , where  $L$  is the Chomsky factor of  $\sigma$ . I am using the same principle in here,  $S$  naught is the Chomsky factor of  $P$  naught  $y_i$ , is the  $i$ th the member of the ensemble from the standard normal,  $m$  naught is the mean.

So, I have now created. I am sorry I have now created  $N$  samples. Now look at this now, it could be very expensive to be able to create the samples, because I am going to be dealing with a million dimensional system perhaps. So, once I have created sample what is the initial mean. Initial analysis mean, is  $\hat{x}_N$  which is the sample mean of the, which is the sample mean of the initial ensemble. This one will is equal to  $m$  naught

plus  $\hat{y}$  of  $N$ . So, if  $\hat{y}$  of  $N$  is the sample mean of the  $y$  ensemble  $\hat{x}$  of  $N$  at the time zero, will be the sample mean of the initial ensemble. Now what is the basic property,  $y$  is a standard normal, is a mean zero. So, what is the claim?

$\hat{y}$  of  $N$  which is given by this, that tends to zero as  $N$  tends to infinity  $y$ ,  $y$ 's are all the samples created from a standard normal with mean zero. So, these are the properties. So, what are the first thing we require for creating an ensemble. I first need to have a very good random number generator, to generate  $N$  capital and samples of random vectors from the standard normal distribution. Once I have this I can do this. Now what is involved in here? I have to do a matrix vector multiplication, then I have to do a vector vector addition, and this I have repeat  $N$  times, and that relates to the total cost of computing the initial ensemble.

(Refer Slide Time: 26:56)

**SQUARE ROOT FORM**

$$\begin{aligned}
 \bullet \hat{\mathbf{P}}_0(N) &= \frac{1}{N-1} \sum_{i=1}^N [\hat{\xi}_0(i) - \hat{\mathbf{x}}_0(N)][\hat{\xi}_0(i) - \hat{\mathbf{x}}_0(N)]^T \\
 &= \mathbf{S}_0 \left[ \frac{1}{N-1} \sum_{i=1}^N [\mathbf{y}_0(i) - \hat{\mathbf{y}}_0(N)][\mathbf{y}_0(i) - \hat{\mathbf{y}}_0(N)]^T \right] \mathbf{S}_0^T \\
 &\rightarrow \mathbf{S}_0 \mathbf{S}_0^T = \mathbf{P}_0 \quad \text{as } N \rightarrow \infty
 \end{aligned}$$

So, what does this initial ensemble give you. I have an estimate of the initial ensemble covariance, that the analysis covariance is initial time. You can see this is the  $i$ th ensemble remember, that the sample mean,  $i$ th ensemble members are the sample mean. You take the outer product of that the sum over  $N$  divided by  $N$  minus 1.

So, this is going to be the sample estimate of the analysis covariance at time 0. analysis covariance at time 0. Now using the transformation I can also rewrite the line 1 by line 2, and the quantity within the parenthesis as  $N$  goes infinity goes to  $i$ ; that is because what

is the quantity in parentheses that is the sample estimate of the covariance of  $y$ . The actual covariance of  $y$  is  $\Sigma$ . So, in time it will go to  $\Sigma$ .

So, the whole thing reduces to for lot sample  $S$  naught times  $S$  naught transpose is equal to  $P$  naught. Therefore, the analysis estimate, the estimate of the analysis covariance at times zero converges to the actual covariance  $P$  naught, as the  $N$  the number of samples got infinity. Why is these limits are important, I keep repeating this, because the quantities calculated using ensemble members, and sample statistics, there is a true statistics, there is always going to be difference. The difference becomes smaller and smaller as the number of samples  $N$  becomes larger and larger. So, if you cut the number of ensembles is to be finite, let us say one 100 200, there is all you, your estimate is going to have an error.

and that is the error you have to deal with  $y$  naught, because I want to, but because of the computational limitations. So, it is not that I like to commit this error, but I do not know how else to do it, because all the other approximations are approximations. Not only they are approximations, they are very expensive. For example, in the extended Kalman filter if you look at the update for the forecast covariance that still requires 2 matrix multiplications. Each matrix multiplication is going to cost to  $NQ$ . So, even though it is approximate, it costs a lot of money to compute that approximation.

So, what is their idea of ensemble? If it is approximate where to spend too much money, can compute in approximation a quick and dirty way? Can I compute approximation in the cheap way, that relates to the notion of ensemble ideas..


(Refer Slide Time: 29:43)

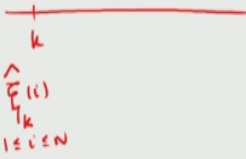
## FORECAST STEP

- Consider  $k$ :  $(\hat{\mathbf{x}}_k(N), \hat{\mathbf{P}}_k(N))$  - Given Analysis
- Let  $\hat{\mathbf{P}}_k = \hat{\mathbf{S}}_k \hat{\mathbf{S}}_k^T$
- Analysis ensemble:
 

$$\hat{\xi}_k(i) = \hat{\mathbf{x}}_k(N) + \hat{\mathbf{S}}_k \mathbf{y}_k(i)$$

where  $\mathbf{y}_k(i) \sim N(0, \mathbf{I})$





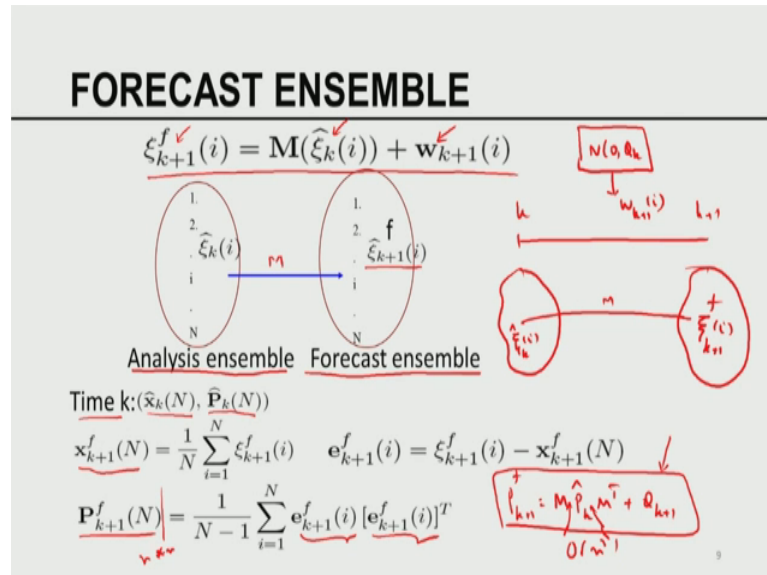
So, If I know how to create the initial ensemble I can fast forward in time, I can fast forward in time. So, what is that now I can assume, I can assume the following. So, let us assume at time  $k$ , I have ensembles forecast ensembles  $k$  i. So, I am sorry this is. these are the analysis ensemble. How it is analysis, hat is analysis superscript  $f$  is forecast. So, let this be the analysis ensemble.

So, what is that what that I have, this is time  $k$ ? So, I have analysis ensemble a  $N$  of them, each point of this is  $\hat{\xi}_k(i)$ . Now so, if  $\hat{\xi}_k(i)$  is the analysis ensemble the analysis ensemble is a related with. the analysis mean through  $\hat{\mathbf{x}}_k(N)$ , and the analysis covariance square root  $\hat{\mathbf{x}}_k$ , through  $\mathbf{y}_k(i)$ . I am sorry  $\mathbf{y}_k(i)$ . So, what is  $\mathbf{y}_k(i)$ . So, let us come back to this now. So, let  $\hat{\mathbf{P}}_k(N)$  be the analysis covariance approximation, as well as this is the analysis mean approximation,  $\hat{\mathbf{P}}_k$  hat can be express with the product of Chomsky factors  $\hat{\mathbf{S}}_k$  hat in  $\hat{\mathbf{S}}_k$  hat transpose. Therefore, I know.

The mean, I know the Chomsky factor. I can also derive; I can also relate from the distribution, I can also relate the samples, the analysis sample to the standard normal distributions. Therefore,  $\mathbf{y}_k(i)$  are belonging to the standard normal distribution, and this formula is an important formula this formula is the same as that one holds good at the initial time except that case, if you put case it will 0 I get what I got. So, if I can do it initial time, I can assume, I can do it at time  $k$ , because I have forwarded in time by induction. So, I have created an initial ensemble. So, this is what is called.

The initial ensemble what is that, its a swarm of points in the N dimensional space. Its a swarm of N points, the capital N points in the little m dimensional space. So, I have an analysis ensemble at time k.

(Refer Slide Time: 32:34)



Now, what do they do. I have to create a forecast ensemble for the next time. So, I am going from time k to k plus 1. I have this analysis ensemble here, I would like to be able to create the forecast ensemble. What do I do? I take a particular analysis ensemble, which is psi had k of i run it through the model, and get the forecast ensemble which is psi k plus 1 i f; that is what is again psi k i is the kth is the ith ensemble member at time k.

By running it through the model m I am going to get the forecast ensemble. So, the forecast ensemble, the ith member the forecast ensemble at time k plus 1, is obtained by taking the ith member the analysis ensemble, running it through the model plus adding a noise. Now let us talk about this noise. Noise you do not add the same noise to every ensemble member use you take. So, at is wk plus 1 I. wk plus 1 i is the ith realization of the model noise. So, what is the model noise generator? This is the model noise generator N 0 Qk, out of that comes w k plus 1 i wk plus 1 i ok.

So, the forecast is random, because of 2 things; one the previous analysis is a random. This is the additional randomness, we are doing. Therefore, this is random; therefore,

that is how I am trying to generate the forecast ensemble. So, from the analysis ensemble at time  $k$ , I am creating the forecast ensemble at time  $k$  plus 1.

In the standard none linear filter or the Kalman filter, there is only one mean, there is only one covariance. I am trying to update the mean, I am going to update the covariance, updating the mean is not expensive; updating the covariance is very expensive. Let me remind you in the classical Kalman filter  $P_{k+1|f}$  is equal to  $m$  times  $P_k$  hat  $m$  transpose plus  $Q_{k+1}$  in Kalman filter application. This is the real killer, why.

I have to do a matrix multiplication. Here I have a matrix multiplication here, each one of them takes the  $N$  cubed time, each one of them takes the  $N$  cube time, each one of them takes  $N$  cubed time o of  $N$  cubed time. And  $N$  is of the order of million we have already seen to do one matrix computation, and a peta flop machine, when  $N$  is a million, it will take about 11 and a half to 12 days. So, this step alone will take close to 24 days. After that we simply cannot, we simply do not have the time, we simply do not have the time; that is why we are looking for all kinds of ways to approximate, which I can do in my lifetime. So, let me summarize the thing as it is given it.

So, at time  $k$ , I am given the analysis mean. I am given the analysis covariance, I am given the assemble itself. Once I have ensemble, I can compute these quantities, I am now have run this analysis ensemble through the model and create the forecast ensemble. Once I have created the forecast ensemble I can compute the forecast mean, I can compute the forecast covariance. This is what I was trying to tell you. So, this is the outer product matrix, I am trying to add  $N$  outer product matrix capital  $N$ , the size of this matrix is  $N$  by  $N$ , but I have only capital  $N$  samples. So, the rank of  $P_{k+1|f}$  at, I am using ensemble is no more than capital  $N$ , capital  $N$  is much smaller.

So, by definition you are dead on arrival. So, what is that, we have the always consider reduced rank maths computationally. These reduce rank matrices are  $L$  conditioned matrices, you cannot invert them, you cannot do many things with them. So, it had to be very covered in trying to do lots of things, and why do we, why do you want to settle for these kinds of crude approximations, because for geophysical problem. These are the only things we can afford to do in today's technology; that is the bottom line. So, I have had analysis of time  $k$ , I had a forecast of time  $k$  plus 1.

(Refer Slide Time: 37:15)

### D. A. STEP

- Build a set of virtual observations

$$z_{k+1}(i) = z_{k+1} + v_{k+1}(i) \quad 1 \leq i \leq N$$

$$\hat{\xi}_{k+1}(i) = \xi_{k+1}^f(i) + K [z_{k+1}(i) - H_{k+1} \xi_{k+1}^f(i)]$$

$x_k \sim H x_{k+1} + v_k$   
 $v_k \sim N(0, R_k)$

$N(0, R_k)$   
 $\downarrow$   
 $v_{k+1}$

$\xi_{k+1}^f(i)$

I have moved things forward. Now what do we do. I had to do a data assimilation. So, there are N forecast members. So, let us go back and understand this now.

So, at time k plus 1, at time k plus 1 I have N forecast ensemble, a typical forecast ensemble is called  $\psi_{k+1}^f$ . I have only one observations  $Z_k$  proof. Remember  $Z_k$  is equal to  $H$  of  $x_k$  plus  $V_k$ ; that is what is given to me, but we already know  $V_k$  is normal with  $R_k$ .

So, what is that? We need to do I have N members, N strands, the forecast I have only one observation, I am not going into the details, if you assimilate the same observation and every strand of the forecast, your analysis covariance will be underestimated. So, to avoid under if the analysis covariance are underestimated that will lead to divergence of the filter computationally, that will be major challenges on problems

So, to avoid that, what do we do? We introduce the notion of what is called virtual observation. You get one observation  $Z_k$ . I am now going to create a set of virtual observations, what is  $Z_{k+1}^i$   $Z_{k+1}^i$  is the  $i$ th version of virtual observation, which is equal to the actual observation given by the satellite radar, whatever it is plus I am going to add to that an  $i$ th version on the observation noise. Please understand  $Z_k$  is already have some noise. I cannot separate  $Z_k$  from the noise, because the noise is unavoidable and inseparable, but I do know the properties of the noise. So, what that we



are trying to do? We are trying to generate another random word generator  $N(0, I)$ , I am going to create a noise vector or  $V_k$ . So, I have  $Z_{k+1}$ .

I am sorry  $V_{k+1}$ , I am going to add  $V_{k+1}$  to get the  $i$ th member of the virtual observation  $Z_k$ . This  $i$  runs from  $i=1$  to  $N$ , then what do I do. I simply apply the Kalman filter, Kalman filter equation. So, the analysis at time  $k+1$  for the  $i$ th ensemble strand is equal to forecast at time  $k+1$ , the  $i$ th member plus  $k$  time  $Z_{k+1}$  minus  $H_{k+1} Z_{k+1}$  here. I would like to enlighten you with a couple of things, I am sure a careful radiogram already noticed it, hey you started with them non-linear model, but you are using a formula that is meant for linear model.

So, this equation for the update for the data assimilation step. We are trying to do 2 things, we are not doing the actual observation, they are creating virtual observation. We are not doing any non-linear thing, we are only doing a linear thing and that should not be surprising to any one of you, even though the model is linear observations are linear, the estimates for the analysis is always a linear estimate, I want to bring that to focus the models in the non-linear filtering.

What is that we have seen, the model is linear, the observations are could be non-linear, but what is the principle we are applying best a linear estimate build. So, the estimator has still a linear structure that is why I am trying to use. So, what is this. This is the estimate, this is the estimate of the analysis, this is the estimate of the forecast plays a role of a background  $Z_k$  minus this term is the innovation I am multiplying by a Kalman gain.

So, what is the rub here, how do you compute Kalman gain. So, Kalman gain has to be, as you know has it is depended on the forecast covariance. The observation of covariance and the operator  $H$  operator  $H$  is well, very well known the forecast covariance known is only approximately, the observational error is known perfectly. So, some are girt some are not girt. So, we had to combine getting back to create what we want. So, you can readily see  $k$  using the formula that we already know.

That mixes the forecast covariance; the focus covariance is already approximate. So, the  $k$ ; that is going to be using here is also approximate. So, there are another source of error.

(Refer Slide Time: 41:56)

$$\begin{aligned}
 \bullet \text{ Mean: } \hat{\mathbf{x}}_{k+1}(N) &= \frac{1}{N} \sum_{i=1}^N \hat{\xi}_{k+1}(i) \\
 &= \frac{\mathbf{x}_{k+1}^f(N) + \mathbf{K} [\bar{\mathbf{z}}_{k+1}(N) - \mathbf{H}_{k+1} \mathbf{x}_{k+1}^f(N)]}{\quad} \\
 \text{where } \bar{\mathbf{z}}_{k+1}(N) &= \frac{1}{N} \sum_{i=1}^N \mathbf{z}_{k+1}(i) = \mathbf{z}_{k+1} + \frac{1}{N} \sum_{i=1}^N \mathbf{v}_k(i) \\
 &= \mathbf{z}_{k+1} + \bar{\mathbf{v}}_{k+1}(N) \\
 \bullet \text{ COV: } \hat{\mathbf{e}}_{k+1}(i) &= \hat{\xi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}(N) \\
 &= (\mathbf{I} - \mathbf{K} \mathbf{H}_{k+1}) \mathbf{e}_{k+1}^f(i) + \mathbf{K} [\mathbf{v}_{k+1}(i) - \bar{\mathbf{v}}_{k+1}(N)]
 \end{aligned}$$

So, let us try to talk about some of the aspects of analysis. So, I have done the analysis using the expressions in page 10. I have now I am now computing the analysis statistics analysis mean, is going to be given by this formula. If I substitute the expression from the previous the, this is the virtual observation, the virtual observation the mean of this are given by that, the covariance in order to compute the covariance, I have to compute the error.

The error in the analysis is given by this, the error analysis has this structure has this structure. Look at this now  $\mathbf{v}_{k+1}(i)$  is the  $i$ th, the member of the observational error, this is the mean of that. So, this difference is the anomaly that anomaly goes to 0 as  $N$  goes to infinity. So, these are some of the things, one has to remember when tries to one, when tries to manipulate these quantities. These quantities can be very small, but still we need to do those, all these small quantities.

(Refer Slide Time: 43:12)

$$\begin{aligned}
 \hat{\mathbf{P}}_{k+1}(N) &= \frac{1}{N-1} \sum_{i=1}^N \hat{\mathbf{e}}_{k+1}(i) [\hat{\mathbf{e}}_{k+1}(i)]^T \\
 &= (\mathbf{I} - \mathbf{K} \mathbf{H}_{k+1}) \left[ \frac{1}{N-1} \sum_{i=1}^N \mathbf{e}_{k+1}^f(i) [\mathbf{e}_{k+1}^f(i)]^T \right] (\mathbf{I} - \mathbf{K} \mathbf{H}_{k+1})^T \\
 &\quad + \mathbf{K} \left[ \frac{1}{N-1} \sum_{i=1}^N [\mathbf{v}_{k+1}(i) - \bar{\mathbf{v}}_{k+1}(N)] [\mathbf{v}_{k+1}(i) - \bar{\mathbf{v}}_{k+1}(N)]^T \right] \mathbf{K}^T \\
 &\quad + (\mathbf{I} - \mathbf{K} \mathbf{H}) \left\{ \frac{1}{N-1} \sum_{i=1}^N \mathbf{e}_{k+1}^f(i) [\mathbf{v}_{k+1}(i) - \bar{\mathbf{v}}_{k+1}(N)]^T \right\} \mathbf{K}^T \\
 &\quad + \mathbf{K} \left\{ \frac{1}{N-1} \sum_{i=1}^N [\mathbf{v}_{k+1}(i) - \bar{\mathbf{v}}_{k+1}(N)] [\mathbf{e}_{k+1}^f(i)]^T \right\} (\mathbf{I} - \mathbf{K} \mathbf{H})^T \\
 &= (\mathbf{I} - \mathbf{K} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^f(N) (\mathbf{I} - \mathbf{K} \mathbf{H}_{k+1})^T + \mathbf{K} \mathbf{R}_{k+1}(N) \mathbf{K}^T
 \end{aligned}$$

$\mathbf{R}_{k+1} \text{ as } N \rightarrow \infty$

Therefore, I am now going to compute with the analysis covariance. Sorry I am now going to compute the analysis covariance.

The analysis covariance is given by this, I am now substituting the expressions for the analysis error for the  $i$ th ensemble member, which is then given by this expression, these expressions, which when simplified will become this, which will simplify which one simplified will become this. So, in this case, I want you to remember  $\mathbf{R}_{k+1}(N)$  is the estimate of the analysis covariance as  $N$  goes to infinity that tends to  $\mathbf{R}_{k+1}$ . So, the other quantities follow readily from these expressions. I also want you to remember a couple of things. These are all coming from the observation noise, these are all coming from this comes from.

The forecast, this comes from the observation noise, this comes from the observation  $\mathbf{R}$  that is forecast. So, you can see this is the cross covariance between the observation noise and the forecast observation noise. In the forecast these are all finite samples. Now how do we know these finite samples make sense in order to understand, where if this finite sample approximation makes sense, I have to consider the asymptotic versions of these formulas

So, what gives you the right to use the finite sample expressions with the errors, if you can show the errors in the estimate using finite samples go to 0 as  $N$  goes to infinity, that gives you a confidence. Yes I know I am committing error, but I am committing error,

not because I wanted to, and this is a direct result of smaller number of samples. If I have the luxury of being able to go for a larger number of samples, all my errors will vanish, and that is a comfortable thought; that is the reason why we do asymptotic. Not because in practice asymptotic makes sense, but asymptotic analysis gives you a guarantee, that when you do estimates with finite number of samples, you can think of being closed to being close to the truth, and how far you are goes to the truth depends on the number of samples.

So, we have to go between asymptotic analysis versus finite sample analysis, all the time and statisticians do this very cleverly, better than anybody else.

(Refer Slide Time: 45:41)

• (cont.) Where

$$\mathbf{R}_{k+1}(N) = \frac{1}{N-1} \sum_{i=1}^N [\mathbf{v}_{k+1}(i) - \mathbf{v}_{k+1}(N)] [\mathbf{v}_{k+1}(i) - \mathbf{v}_{k+1}(N)]^T$$

and

$$\frac{1}{N-1} \sum_{i=1}^N \underbrace{e_{k+1}^f(i)}_{\text{full rank}} [\mathbf{v}_{k+1}(i) - \mathbf{v}_{k+1}(N)]^T \rightarrow 0$$

• Thus,

$$\mathbf{K} = \underbrace{\mathbf{P}_{k+1}^f(N)}_{\text{full rank}} \underbrace{\mathbf{H}_{k+1}^T}_{\text{low rank}} [\underbrace{\mathbf{H}_{k+1} \mathbf{P}_{k+1}^f(N) \mathbf{H}_{k+1}^T}_{\text{full rank}} + \underbrace{\mathbf{R}_{k+1}}_{\text{full rank}}]^{-1}$$

Handwritten notes on the right side of the slide:

- $\hat{\mathbf{x}}_k(i)$  and  $\hat{\mathbf{x}}_k(N)$  are connected by a red arrow pointing to  $\mathbf{P}_k^f(N)$ .
- $\mathbf{P}_k^f(N)$  and  $\hat{\mathbf{x}}_k(N)$  are connected by a red arrow pointing to  $\hat{\mathbf{x}}_k(N)$ .
- A bracket groups  $\mathbf{P}_k^f(N)$  and  $\hat{\mathbf{x}}_k(N)$  with the label "N-SAMPLE ESTIMATE".
- Below  $\mathbf{P}_k^f(N)$  and  $\hat{\mathbf{x}}_k(N)$  are the terms  $\mathbf{P}_k^f$  and  $\hat{\mathbf{x}}_k$ .

Therefore, and I am now going to do some of the asymptotic analysis, as I talked about this term is the anomaly in the observation noise, and that tends to  $\mathbf{R}_{k+1}$  which I have already alluded to. I also want you to remember you one more, the cross term what is this forecast, the error. I am sorry the cross term, the second term I am sorry first term second term, third term fourth term, the third time and the fourth term in the slide, in slide 12.

This term, and this term involve the cross covariance, it can be shown that cross covariance, this is the cross covariance, what is the cross covariance we may, you may bring this one, this is the forecast error of the  $i$ th ensemble, this is the anomaly of the  $i$ th realization of the observation noise, the cross covariance of that as  $N$  goes to infinity

goes to 0. Therefore, in the previous slide, the third term and the 4th term, they automatically go to 0.

Therefore, the expression for  $k$ , the Kalman gain, which I have been, which I have postponed for, this log is given by the  $N$  sample approximation of  $p_{k+1|f}$ . Now look at this now  $\psi_{k|f} \hat{\psi}_{k|f}$ . These are all  $i$ th member of the forecast and analysis, but when it comes to question of  $p_{k|f}$ . I am going to have an estimate, I am going to put a  $N$ , if I am going to have  $p_{k|N}$ . So, what are these are  $N$  sample estimate.

Estimate based on  $N$  sample  $N$  sample estimate  $N$  sample estimate, because I want to be able to distinguish between. So, if I say  $p_{k|f}$  that is the actual forecast covariance if I say  $p_{k|f}$  that their actual analysis covariance if I attach a  $N$  in the stomach to them. There is an estimate. So, this is the  $N$  sample estimate of the forecast covariance  $h_k$  plus one transpose. This is the same formula as comes from linear analysis, this product again we understand these 2 are the same quantities  $N$  sample approximation, and  $r_{k+1|f}$  plus one  $r_{k+1|f}$  plus one we already know. Look at this. Now I want I would like to talk about a little bit further. Now  $r_{k|f}$  we are assuming is full rank.

hits  $t$  of  $k$  plus one; that is low rank  $h_k$  we are assuming the full rank. So, the product of this is low rank, but I am adding a low rank to a full rank matrix and keep getting the inverse. Thus the inverse exists, yes who guarantees the inverse exists Sherman Morrison Woodbury formula. So, what is that, you can consider this, I am having a low rank perturbation of a full rank matrix. So, this inverse can be shown to exist, and using this I can compute  $k$ . So,  $k$  is an approximation. So,  $k$  is an approximation. So, what is the actuals data assimilation step; that is on page 10. I want to revisit that in the light of these calculations.

So, I have now calculated  $k$ , if I have calculated the  $k$ , I know the  $i$ th virtual observation, I know everything, I can compute the analysis ensemble, if I can compute the analysis ensemble, I can computer analysis mean an analysis forecast. So, let us talk and analysis covariance. So, let us assume the 2 things now. So, what do these centres. Do they have a large code that representing the model they run this model code parallel for  $N$  initial condition  $N$  is 51 100, I do not think the world anybody runs large models, using more than 200, 200 ensembles nobody, it is not, because they do not know its simply, because they do not have the computing power.

Why let us talk about this. Now you have to make daily forecasts, you have to, you have. I am sorry you have to make forecasts every day, every day you make forecasts for one day in advance, 2 day in advance, 3 days in advance, 5 days in advance, six in days in advance, seven days in advance, and you keep doing this every day.

So, you have only 24 hours maximum time; that is possible to you of the 24 hours. The observation people have to collect all the observations from radar from satellites, and bring them all to a common platform the modular, have to run the model forward in time 50 ensemble, and 100 ensembles. The module is, give the model output the observation people give the observation then.

The data assimilation person comes to work, the data assimilation person, does not come to work. Until and unless all the observations are made available, until unless all the model runs are available. At that time he takes the both and he tries to create this analysis ensemble.

Once he creates the analysis ensemble, he can create the analysis mean analysis covariance, it is this analysis mean analysis covariance for one day, 2 day, 3 day, 5 day is being announced as a forecast product. So, what must, what will be the time available for modular. Let us say 6 hours, what will be the time available for all the observations, people to get the observation, and more maybe 12 hours. How much time a data assimilation may have 6 hours 8 hours.

So, you have to finish all these things in a fixed number of hours, and one has to depend on the other the data assimilation, person is the last one who comes to work. So, if you do up the 24 hours into these 3 parts, you can see how the operational centres are busy. They do not wait, they do not waste time IOTA of a second, and that is what happens in all the major forecasting centres; such as ends up in Washington D C ECMWF are British meta office Japanese meteorological agency. I am sure Indian meteorological agent agency. They also do that, depending on what kind of a forecasting system data assimilation system, they have assimilated and what kind of computers they have. I hope I have made it.

(Refer Slide Time: 42:59)

## COMMENTS

- (1) When  $N$  is small, cross product terms  
 $\mathbf{e}_{k+1}^f(i) [\mathbf{v}_{k+1}(i) - \bar{\mathbf{v}}(N)]^T$  will not be close to zero  $\Rightarrow$  error is  $\mathbf{P}_{k+1}^f(N)$   
and hence in  $\mathbf{K}_{k+1}$ , the Kalman gain
- (2) Covariance Matrix

$$\mathbf{P}_{k+1}^f = \mathbf{D}_M(\hat{\mathbf{x}}_k) \hat{\mathbf{P}}_k \mathbf{D}_M^T(\hat{\mathbf{x}}_k) + \mathbf{Q}_{k+1}$$

(Approximation)

\*  $\mathbf{D}_M(\hat{\mathbf{x}}_k)$  is non-singular  
 $\mathbf{I}, \mathbf{U} \cdot \mathbf{A} \in \mathbb{R}^{n \times n}$

\*  $\mathbf{P}_{k+1}^f$  is full rank but approximate

(comparison)

$n = 10^6$

$N = 100$

$\text{Rank } \mathbf{P}_{k+1}^f(N) \leq N < n$   
(reduced rank)

Clear up to you, as to how the data assimilation is done within the context of, within the context of ensembles. So, some comments when the  $n$  is small  $n$  is always small, not then  $n$  is small a, the cross product terms you remember where earlier they will not be close to zero; that means, there is, there will be a large error in the estimate of the  $N$  sample estimate of the forecast covariance, is going to be in error forecast covariance errors, directly reflects in to Kalman gain error. So, that is something I wanted to be cognizant to off that error goes to 0, only when number of samples becomes large. I also want to talk about one more, the covariance matrix.

For forecast, forecasts covariance is the bottleneck in in Kalman filtering. So, this is the Jacobian times  $\mathbf{p}_k$  hat Jacobian times  $\mathbf{p}_k$  hat when  $N$  is equal to 1. So, when  $N$  is equal to one million, if I am using 100 samples, the rank of  $\mathbf{p}_k N$  is no more than  $N$ , which is less than  $N$ . Therefore, reduce rank I am trying to reinforce all these things by quantifying some examples. Now please understand this itself is an approximation, even the approximation cost money I have already alluded to this in, I have then the  $\mathbf{D}_m \times \mathbf{k}$  is not available  $\mathbf{D}_m \times \mathbf{k}$  is the million by million matrix. I only know the analysis, I have to evaluate the elements in the million by million matrix.

Along the trajectory. So, that is another hidden cast, that does not come out, that does not come out, but. So, in the case of covariance approximation. I am sorry in the case of first order filter or second order filter. So, the  $\mathbf{p}_k$  full rank in first order, and second order

filter. Therefore, its a full rank, but its still approximate full rank, but not exactly something approximate

So, I would like you to keep all these things on the back of the mind.

(Refer Slide Time: 55:11)

### COMMENTS (CONT'D)

---

- (3) Need for Virtual Obs

If we used actual observation instead of the virtual:

$$\begin{aligned} \hat{\xi}_{k+1}(i) &= \xi_{k+1}^f(i) + \mathbf{K} [\mathbf{z}_{k+1} - \mathbf{H}_{k+1} \xi_{k+1}^f(i)] \\ \Rightarrow \hat{\mathbf{e}}_{k+1}(i) &= \hat{\xi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}(N) \\ &= \hat{\xi}_{k+1}(i) + \mathbf{K} [\mathbf{z}_{k+1} - \mathbf{H}_{k+1} \xi_{k+1}^f(i)] \\ &\quad - \xi_{k+1}^f(N) + \mathbf{K} [\mathbf{z}_{k+1} - \mathbf{H}_{k+1} \xi_{k+1}^f(N)] \\ &= \mathbf{(\mathbf{I} - \mathbf{K} \mathbf{H}_{k+1})} \mathbf{e}_{k+1}^f(i) \end{aligned}$$

15

And I am also not going to argue why you need virtual observations. I have already alluded to that, if you use the actual observations from virtual observation, you would use only one observation for every ensemble. If you compute the analysis error the term for the analysis error becomes this, when you have to compute the analysis error, it becomes this.



(Refer Slide Time: 55:39)

## COMMENTS (CONT'D)

=> There is no term  $K[v_{k+1}(i) - \bar{v}_{k+1}(N)]$  in the error

=>  $\hat{P}_{k+1}(N) = (I - KH_{k+1})\hat{P}_{k+1}(N)(I - KH_{k+1})$

- no  $KRK^T$  terms in it.

=>  $\hat{P}_{k+1}(N)$  is an underestimation

∴ Need for virtual observation to get correct analysis Covariance

G. EVENSEN - (1995)

16

And the analysis covariance matrix computed using this becomes this. Given by this expression we have already discussed all these things very much in detail in our books, but I would like to tell you that the analysis covariance term is given by this, and it does not have the  $K r K^T$  term in it and.

This lack of  $K$  or  $K^T$  term in it leads to underestimation, and if the analysis error as is the analysis covariance underestimated forecasts covariances, also underestimated, because forecast covariant depends of the analysis covariance; that is the reason why we use virtual observation to be able to make up for this deficiency in resulting an underestimation, when ensemble was known as the Monte Carlo method of estimating Kalman filters in non, in for non-linear systems within the control theory literature as early as 1960s, but it was invented by invention in the middle, in in the mid 90s within the context of geophysical applications.

So, within the context of geophysical application, the person who made this ensemble approach popular is Norwegian geophysical scientist Evensen, I think G Evensen around 1995 approximately that off approximately that off. So, ever since the ensemble methods have become popular, most of the centres have gone for this, when originally Evensen published the paper he did not use the virtual observation. People who came after him is quickly realized that if you did not, if you used only the same observation to every ensemble strand.

Your system leads to underestimation, underestimation leads to a numerical difficulties. So, to be able to prevent such numerical difficulties from arising an artificial method was created, and that is what virtual observations all about. So, it is simply a mathematical trick by which I can restore some sanity in the world, where everything is approximate.

(Refer Slide Time: 48:16)

### COMMENTS (CONT'D)

- (4) Modify the Kalman Gain instead of the obs.
  - Let  $\hat{\xi}_{k+1}(i) = \xi_{k+1}^f(i) + \mathbf{W}[z_{k+1} - \mathbf{H}_{k+1}\xi_{k+1}^f(i)]$
  - Mean of the ensemble then becomes
  - $\hat{\mathbf{x}}_{k+1}(N) = \mathbf{x}_{k+1}^f(N) + \mathbf{W}[z_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1}(N)]$
  - $\hat{\mathbf{e}}_{k+1}(i) = (\mathbf{I} - \mathbf{W}\mathbf{H}_{k+1})\mathbf{e}_{k+1}^f(i)$
  - Posterior Cov:
    - $\hat{\mathbf{P}}_{k+1} = (\mathbf{I} - \mathbf{W}\mathbf{H}_{k+1})\mathbf{P}_{k+1}^f(\mathbf{N})(\mathbf{I} - \mathbf{W}\mathbf{H}_{k+1})^T$
    - \*  $\mathbf{P}_{k+1}^f = \mathbf{S}_{k+1}^f(\mathbf{S}_{k+1}^f)^T$  .....Square root factorization
    - $\hat{\mathbf{P}}_{k+1} = (\mathbf{I} - \mathbf{W}\mathbf{H}_{k+1})\mathbf{S}_{k+1}^f(\mathbf{N})(\mathbf{I} - \mathbf{W}\mathbf{H}_{k+1})\mathbf{S}_{k+1}^{fT}$

What is the next idea. Some might say hey instead of trying to modify the observation, why do not we simply change the Kalman gain; such that I do not have to mess with the observations. Why messing with the observation is considered bad in some circle.

Observation is given by Mother Nature. Mother Nature is observed only through observation, and why to corrupt what Mother Nature tells us. So, philosophically some people are opposed to creating virtual observation, because it is not philosophically consistent meddling with what Mother Nature is telling us. Therefore, what is it they were looking for? They were looking for an up, they were looking for an alternate approach, and what is the alternate approach depends on this alternate approach. It depends on the following I have the analysis ith ensemble, I have the forecast ith ensemble, I have the innovation.

Please remember I am using the same observation, while using the same observation can I use another way w; such that I can restore sanity in the analysis of ensemble; that is a very beautiful mathematical question. This question was answered the affirmative. There are several classes of ensemble filters; they came out as a result of this line of

questioning. So, the line of doing the assimilation using virtual observation is called stochastic method. The line where you use the single that the same only one observation, but change the weight that is called a deterministic methods

So, deterministic methods of implementing ensemble filters stochastic methods, for implementing ensemble method stochastic method relies on creating virtual observation. What is the advantage of it. You simply create the virtual observation you are done. You can close the eyes and then say, hey I have decent approximation limited only by the value of here.

In here I have to do a little bit of more calculation, but I do not have to mess with the observation, observations are word of God that is essentially the kind of approach that is taken here. So, that is they approach of how do we modify the Kalman gain instead of the observation. So, this is the. So, if this is the analysis ensemble. This is the mean of the analysis ensemble. If I know the analysis ensemble, and this mean I can complete the analysis the error anomaly. If I had compute the analysis error, I can compute the analysis covariance by this expression.

Please remember I am now talking like a broken record, I am following the same paths, but there is different criterion for mixing the forecast information, and the innovation information. So, w the matrix, I am now going to change the Kalman gain to suit what I want without having to generate virtual observations. Therefore, if the error structure is given by this, the posterior covariance, which is the analysis covariance is given by this expression is given by this expression.

Let us assume the forecast covariance has a square root factorization. Like that has a fair root. So, this is the forecast covariance. So, this is the forecast covariance, let us pretend this forecast covariance has a square root factorization, available in that case by combining these 2 I get by replacing this by the square root, I get the analysis covariance is given by the product of this times that you can readily, you can readily see that. I believe that must be there, must be a parenthesis, then here.

Sorry there must be, there is a parenthesis here, there is must be a parenthesis there, and then there must be a parenthesis here, with that that is correct, because  $I - WH^T S^{-1} W^T$  transpose  $i$  minus (Refer Time: 62:38) times as transpose to the whole thing; that is correct your parentheses missing here. Please add that.

(Refer Slide Time: 62:44)

### COMMENTS (CONT'D)

- Question: How to choose  $W$ ?
 

COV

$$\hat{P}_{k+1} = P_{k+1}^f - P_{k+1}^f H_{k+1} [H_{k+1} P_{k+1}^f H_{k+1}^T + R_{k+1}]^{-1} H_{k+1} P_{k+1}^f$$
 (30.1.27) <= Factorize this!
 

Andrews(1968)
- Simply the \_\_\_\_\_ tion:
 

Let  $P^f = S^f (S^f)^T$     $A = (HS^f)^T$

$$\hat{P} = S^f (S^f)^T + S^f (S^f)^T H_{k+1} [H_{k+1} S^f (S^f)^T H_{k+1}^T + R_{k+1}]^{-1} H_{k+1} S^f (S^f)^T$$

$$= S^f [I - (S^f)^T H_{k+1} [H_{k+1} S^f (S^f)^T H_{k+1}^T + R_{k+1}]^{-1} H_{k+1} S^f] (S^f)^T$$

$$= S^f [I - A(A^T A + R)^{-1} A^T] (S^f)^T$$

$\underline{S^f}$ 
 $\underline{\hat{S}}$ 
 $\underline{S}$

COV SQUARE ROOT VERSION
- Now factorize this!

So, with that the question is, how to choose  $W$ . I am going to quickly talk about the method for choosing  $W$ , this idea an electrical engineering systems literature, as Andrews 1968, it was reinvented in the meteorological community in the late 90s, after Evensen put forth his ideas, it is very hard to tell how much they knew of the previous literature, but anyway they have been reinvented. So, this is the analysis covariance, the forecast covariance minus this. If they remember this, from their classical Kalman filter equations.

Though. So, what is the basic idea, I would like to be able to simply factorize, and why do I, why do I want to talk about factorization. There is a close connection between square covariance, square root implementations, and reduced rank and ensemble implementation. They are nearly related and.

So, if I have a cholesky factorization, I am now going to call  $A$  is equal to  $H$  times  $S^f$  transpose. So, I am now going to factor the forecast covariance. I am going to define a new quantity. So, this is the forecast covariance factorization, this is the new quantity  $A$  I am going to introduce.

With these changes in notation this  $\hat{P}$ . So, what is that I am now trying to do? I am trying to drop the time index, drop the time index. Why drop the subscripts. I do not want to cover my expressions, because this is all happening at a given time. So, I do not have to remind myself and build those all the subscripts together.

So, by drop the subscripts from the covariance matrices. Therefore, P hat is sub product of the square roots plus this expression. So, these 2 expressions are essentially one of the same. This is the covariance version. This is the square root version, covariance square root version, covariance square root version, square root version. And with this square root version, the expression becomes this, and that can be written as this expression after simple matrix algebra.

Now, what do I want to do? I would like to be able to factorize the one; that is within the square bracket in here, and that is what the factorization is to be done I would like to remind you now the the pk plus one hat becomes this and now that has been expressed this way I would like to be able to. So, if the forecast covariance can be expressed as the square root form analysis covariance also must be expressed within those square root forms. If I express the analysis covered in the square root form, I can identify the W, the quantity that I need very easily.

(Refer Slide Time: 65:55)

### COMMENTS (CONT'D)

- Let  $(A^T A + R) = SS^T$ ,  $R = FF^T$   
 $\Rightarrow (A^T A + R)^{-1} = (SS^T)^{-1} = S^{-T} S^{-1}$
- Then  
 $A(A^T A + R)^{-1} A^T$   
 $= A S^{-T} S^{-1} A^T$   
 $= A S^{-1} [S + F]^{-1} (S + F) (S + F)^T (S + F)^{-T} S^{-1} A^T$   
 $= A S^{-T} (S + F)^{-1} [(S + F) (S + F)^T] (S + F)^{-T} S^{-1} A^T$   
 $= A S^{-T} (S + F)^{-1} [S(S + F)^T + (S + F) S^T - A^T A] (S + F)^{-T} S^{-1} A^T$   
(see following slides)

$$a = \frac{(a + b - b)}{a \times \frac{b}{b}}$$

So, let A transpose, they let this be the square root of. Look at this, now I am using several different devils, several different symbols. There are lots of quantities involved in here. Here is S hat S. This is the square root of the forecast covariance. This is square root of the analysis covariance, S is the square root of A is a generic square root. I am going to use the generic square root in order to be able to define the square root of the quantity this quantity. Sorry I am going to define the square root of that quantity, this

quantity. So, the matrix  $A$  is a matrix  $A$  transpose  $A$  is a germium  $A$  transpose  $A$  plus  $R$  is a symmetric matrix. Let this be the square root of that; that is generic. Let  $R$  be  $ff$  transpose. Therefore, the inverse of, this is the inverse of this, which is given by the inverse of that.

And I am now going to consider a matrix of this type. I already know the square root of  $A$  transpose  $A$  plus  $R$  given by that. I am going to substitute this using this. now I am going to multiply and divide. So, what is that, we can do if I am  $A$  is equal to  $A$  plus  $B$  minus  $B$ . I can add and subtract  $A$  is equal to  $A$  times  $B$  divided  $B$ . I can multiply and divide. These are standard tricks in mathematics, to create newer expressions.

So, what is that? I am now trying to do. I am trying to multiply, I am trying to multiply the inverse, and the matrix the transpose and its inverse. So, I am simply trying to do multiply, and divide. If I multiply and divide, I get an expression that expression when simplified becomes this. See the following slides for the details.

(Refer Slide Time: 67:49)

### COMMENTS (CONT'D)

---

- $(S+F)(S+F)^T = (S+F)(S^T+F^T) = SS^T + SF^T + FS^T + FF^T$
- $S(S+F)^T + (S+F)S^T - A^TA$
- $= SS^T + SF^T + SS^T + FS^T - A^TA$
- $= SS^T + SF^T + FS^T + FF^T$
- $(SS^T - A^TA = A^TA + R - A^TA = FF^T)$

20

So, the simplifications are given in here, using this simplification.

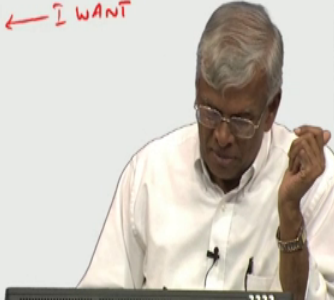
(Refer Slide Time: 67:55)

### COMMENTS (CONT'D)

- Substitute and verify
 
$$\hat{P} = S^f [I - A(A^T A + R)^{-1} A^T] (S^f)^T$$

$$= S^f [I - A S^{-T} (S + F)^{-1} A^T] [I - A S^{-T} (S + F)^{-1} A^T]^T (S^f)^T \quad \text{K.F.}$$
- Compare it with
 
$$\hat{P} = [(I - W H_{k+1}) S_{k+1}]^f [(I - W H_{k+1}) S_{k+1}]^T \quad \leftarrow \text{I WANT}$$

$$\Rightarrow W = S^f A S^{-T} (S + F)^{-1} = P^f H^T S^{-T} (S + F)^{-1}$$
- Whitaker and Hamill (2002)
- Tippet et al. (2003)



Now I can substitute and verify, but my analysis covariance our time. Yes, I know I am going a little fast, but I want you to be patiently look through this. These are grant this, this is, this all grant to work from matrix algebra, and if you want to be thorough in understanding data assimilation method, we have to be ready for this grant work making you ready for the grant work as part of the aims of these lectures. Therefore, this is the original expression by substituting, the factorize the expression from the previous one, I get this.

Now, what do I want. So, this is the expression I get from Kalman filter. This is the expression I get, I want these to be equal, if I equate these 2 i everybody, but W are known. So, I am kind, I am computing the same quantity in 2 different ways. If you compute the same quantity in 2 different ways, they are equal. So, by equating these two. So, this expression. And this expressions are equal one expression has a W. Other expression does not have a W. The W expression is a new way of deriving the one, without expression is the classical way of deriving by. So, what is the, our aim. I want to change the Kalman gain, but I do not want to change the result.

So, how do I use W, and still enforce the same result as I would have garden, had I use the standard Kalman gain, without having to use the virtual observation. So, that gave rise to an expression for W, which is given by this, which is given by this. Therefore, if you use this W as a new Kalman gain in our analysis step. Let me go back in our analysis

step in page 17, I will get the same result as Kalman, but without having to worry about, without having to worry about the virtual observations.

So, this corresponds to the deterministic approach, to deterministic approach to, Kalman ensemble based ideas ensemble filters. Some people call it ensemble filters, some people call ensemble Kalman filters. I think it is misnomer to called ensemble Kalman filter, because classically Kalman filter refers to LQG. Here nothing is LQ, nothing is L, everything could be in general linear. So, within the context of meteorology, we use my preferred way of telling, or explaining this is. It is simply an ensemble filter or a reduce rank filter ok.

Now, use ensemble filter equal to reduce rank filters. Now I, there are million ways in which one can create or reduce rank filter. Ensemble method is one version of trying to create a registrant filters. So, in my view reduce rank filters is a larger class of filtering mechanisms, where we deal with approximations of different degrees ensemble methods is simply one way of being able to create such approximations meaningfully, by simply running the model forward in time. So, what is the basic thing in here. If I have the muscle power to be able to run the model in parallel thousand times, and better off, but there is no brain cover involved there are simple statistics.

So, by trading muscle power from the brain power, ensemble method is rests on making quick simple estimates. Of course, there are nuances do I do virtual observation, do I do other things so, but in principle the totality the amount of computation that one has to make in ensemble methods, is simply dominated by the time required for model rats; that is where parallel computers come in to be do you make 200 runs of the same model, starting from different initial condition on the same computer. No the technology has provided very powerful supercomputers, which are, which consists of several individual units.

So, what is that you do in the world. They have large parallel computers, they can copy the model code into many of these computers, and let all the computers run at near synchronous speed, starting from different initial conditions by. So, by exploiting the parallel computing technology, by combining it with the ensemble methodology in modern days, we can create very meaningful forecasts, based on data assimilation techniques and this has. This is becoming one of the major workhorse in the forecasting



industry, especially in weather forecasting all around the world, all around the world, the British metaphase spearheaded this, ends up is following. This meet you a friends Japanese meteorological agency, and many of these places, they have a research unit as well as a prediction unit, production unit, they all work together in conjunction.

While they run the old version. They are also trying to bring in the new codes for some time, they run both the systems in parallel to be able to test it, once the ensemble methods are well validated, then they slowly provide a sunset class for the old methods, and the new methods take hold in trying to generate daily forecasts; that is the present situation. If you want to know more details about this, you can refer to the papers by with Whitaker and Hamill. You can also refer to the paper by tippet reference to all these. papers are available in our textbook, we have a rather extensive annotations to the literature.

(Refer Slide Time: 73:47)

### RRSQRT FILTER

- Let  $1 \leq p \leq n$ , we seek rank  $p$  sq. root filter.
  - SQRT to improve condition number
  - RR to reduce computation cost
- Initialization:
  - $x_0: E(x_0) = m_0, \text{Cov}(x_0) = P_0$
  - $P_0 V = V \Lambda$   $V V^T = V^T V = I$
  - $P_0 = V \Lambda V^T = V \Lambda^{1/2} \Lambda^{1/2} V^T = (V \Lambda^{1/2}) (V \Lambda^{1/2})^T = S_0 S_0^T$
  - $S_0 = (V \Lambda^{1/2})$

$P$

$t_V(P) = \text{TOTAL VAR}$

$= \sum_{i=1}^n \lambda_i$

$P = \frac{\sum_{i=1}^p \lambda_i}{\sum_{i=1}^n \lambda_i} I$

Let  $P$  be a number in between one and one, and an  $N$  is the dimensionality of the state space. So, if  $P$  is less than  $N$ , I am going to be talking about  $P$ s are the reduced length implementation of square root versions of the covariance filters mouth is fill. So, we are seeking a rank  $P$  squared filters. What is the use of square roots. Square root filter generally improved condition number. Condition number improves the quality of the numerical computation, reduce rank in. generally reduces the computational cost why the computational cost you can readily. See that if the dimension is large the cost is large.

So, you can think of it as a kind of a reduction, largely induced by the desire to reduce the computational cost, as well as advantage to improve the condition number. So, what is that, we do let us talk about the initialization stage. There is the mean; that is a covariance. Let  $V$  be the matrix.

I am sorry let  $V$  with Eigen vectors of  $P$  naught corresponding to the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_N$ . So,  $\lambda_i, v_i$  are the pair. I can call it all the eigenvectors into a matrix  $V$ . I can call it, all the eigenvalues along the diagonal and call it a matrix  $\Lambda$ . So, this is the classical eigenvalue decomposition which can be written like this. Since we already know  $V V^T$  is equal to  $V^T V$ ,  $V$  is equal to  $I$ . I do not know how many times we have seen this eigenvalue decomposition is very fundamental.

We have also seen there are methods for finding square root one is using cholesky. Another is using eigenvalue decomposition. So, here I am interested reduced rank. I am using eigenvalue decomposition. So, I am using eigenvalue decomposition as a tool to implement reduced rank approximations. So, that is the pathway.

So, I can express  $\Lambda$  is equal to a  $\Lambda$  square root of  $\Lambda$  times square root of  $\Lambda$ . Then I can associate this like this. I am going to call  $S$  naught  $S$  naught transpose that  $S$  naught is given by this. Therefore,  $P$  naught is equal to  $S$  naught  $S$  naught transpose is the full square root factorization, where  $S$  naught is one factor,  $S$  naught transpose is the another factor, these 2 factorization in even the looks at the Cholesky factorization I obtained  $S$  naught not as a cholesky factor, but as a product of  $V$  and  $\Lambda$  to the power half. I want you to recognize this is a different way of doing factorization. So, far no approximation. Now I would like to be able to come to the approximation.

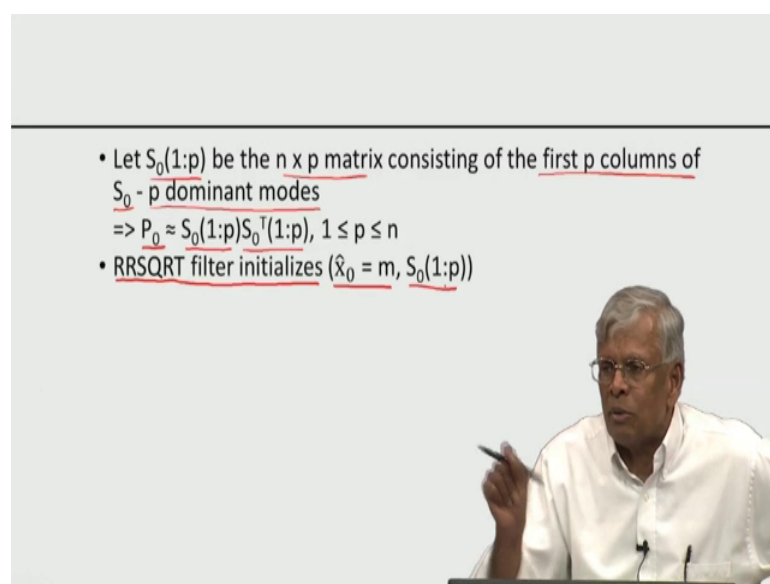
We have ordered the eigenvalues the decreasing in order. So, let us pick the top  $P$  values of the eigenvalues, and corresponding top  $P$  eigenvectors  $V_1$  to  $V_P$ ,  $\lambda_1$  to  $\lambda_P$ , why is that we have already alluded to this. I will say it again the trace of your  $P$  is a covariance of trace of  $P$  is equal to total variance in all the components, and the total variance is the summation  $\lambda_i$   $i$  is equal to 1 to  $N$  if the  $\lambda$ s are distributed as fast decreasing. See what are the various ways in which  $\Lambda$  can occur in for some matrices  $\Lambda$  can be flat like that, for some matrices  $\Lambda$  can decrease like that. So, 1 2 3  $n$ .

So, this is the case they are interested in the bottom one, if the lambda has decreased very fast I can pick A P such that  $\sum_{i=1}^P \lambda_i$  is equal to 1 to P divided by  $\sum_{i=1}^N \lambda_i$  is greater than 90 percent. So, what does it mean the first P components together can explain 90 percent of the overall total variance; that is hidden in the system. So, is there 90 percent accurate approximation, it could be 80 percent approximation, it will be 99.9 percent approximation.

So, you decide the level of approximation, you are willing to contend with given that level 90 80 99. I can compute A P that corresponds to this inequality. The sum of the first P eigenvalues to the total sum of all the eigen values is greater than or equal to A. Previously accepted threshold of 80 percent 90 percent 99 percent. So, that is how I am going to pick A P.

Now, if the, and the other hand the eigenvalues grows a decay, very slowly what does it mean. You may have to have the entire eigenvalue system, the P may be very close to N. So, in this case 1, this case is 2. I am considering the case 1, in the case 2. You may not be able to reduce the rank by simply chopping off the last one, because even the last eigenvalue may be very large. So, it all depends on the intrinsic properties of the Eigen system of the associated covariance matrices. So, I am now going to assume my associated system has a picture, which is very similar to the curve one.

(Refer Slide Time: 79:37)



The slide contains the following text:

- Let  $S_0(1:p)$  be the  $n \times p$  matrix consisting of the first  $p$  columns of  $S_0$  -  $p$  dominant modes
- $\Rightarrow P_0 \approx S_0(1:p)S_0^T(1:p)$ ,  $1 \leq p \leq n$
- RRSQRT filter initializes ( $\hat{x}_0 = m$ ,  $S_0(1:p)$ )

In the bottom right corner of the slide, there is a video inset showing a man with grey hair and glasses, wearing a white shirt, speaking and gesturing with his right hand.

So, if I do that  $\Sigma$  please go back to in in the previous one what is  $\Sigma$   $\Sigma$   $\Sigma$  is a  $N$  by  $N$  matrix is the  $N$  by  $N$  matrix I am going to keep the first  $p$  columns of it and drop the last  $N$  minus  $p$  columns. So, the matrix sorry. So, the matrix that is built out of the first  $p$  columns of  $\Sigma$  is now going to be called just not one column  $p$ . So, the matrix  $N$  by  $p$  matrix consisted in the first  $p$  columns of  $\Sigma$  of the  $p$  dominant modes what do you mean by dominant modes the eigenvalues are the larger the one that I am dropping together the total sum is less than 10 percent is less than 5 percent is less than one percent. So, I am trying to divide the eigen system into dominant modes and not that dominant a mode I am simply collecting all the dominant modes together.

In that case I can take this  $\Sigma$  one column  $p$  and its not transverse your one column  $p$  for some  $p$  in the range one to  $p$  I can approximate by  $\Sigma$  by that. So, what is this this is that rank  $p$  approximation of  $\Sigma$ . So, what is that they are now going to we are going to initialize yeah reduce rank square root filter with the medium instead of the full covariance I am going to be talking about into the full initial covariance square root I am going to be talking about the first predominant modes of the covariance square I hope I hope you see you see you see the difference this is a very beautiful idea.

(Refer Slide Time: 81:20)

• Forecast:

$$\underline{x_{k+1}^f} = M_k \hat{x}_k$$

• What is  $\underline{S_{k+1}^f(1:p)}$ ?

• Recall:  $\underline{P_{k+1}^f} = M_k \hat{P}_k M_k^T + Q_{k+1}$

$$= M_k \hat{S}_k \hat{S}_k^T M_k^T + \underline{S_{k+1}^Q (S_{k+1}^Q)^T}$$

$$= \underline{S_{k+1}^f (S_{k+1}^f)^T}$$

where  $\underline{S_{k+1}^f} = [M \hat{S}_k, S_{k+1}^Q] \in \mathcal{R}^{n \times 2n}$

24

So, now I am going to create a forecast. So, what is the forecast covariance one to  $p$  the forecast covariance one to  $p$  I am now going to talk about the your expression for this this is the full rank forecast covariance this  $P_k$  has a square root implementation like this

and this as a square root implementation like this this I can rewrite it  $\mathbf{S}_k$  plus one  $\mathbf{f}^T \mathbf{S}_k$  plus one  $\mathbf{f}$  transpose where  $\mathbf{S}_k$  plus one  $\mathbf{f}$  is given by this and that is our  $N$  by  $2N$  we talked about already the expansion part in the last lecture when we talked about covariance square root implementation.

So, recall we do not know  $\bar{\mathbf{S}}_k$  we do not know  $\bar{\mathbf{S}}_k$  because we do not know the full.

(Refer Slide Time: 82:18)

• Recall we do not know  $\hat{\mathbf{S}}_k$  but only its rank approximation  $\hat{\mathbf{S}}_k(1:p)$   
 • Also we only know  $\mathbf{S}_{k+1}^q(1:q)$  for some  $q$   
 • Let  $\mathbf{S}_{k+1} = [\mathbf{M} \hat{\mathbf{S}}_k(1:p), \mathbf{S}_{k+1}^q(1:q)] \in \mathcal{R}^{n \times (p+q)}$   
     Overall rank is  $p+q$   
     Adding this increases the rank of L.H.S.  
 • Question is how to obtain  $\hat{\mathbf{S}}_k(1:p)$ ?

But only the rank  $p$  approximation of  $\bar{\mathbf{S}}_k$ . So, that is important thing we have been satisfied with the rank  $p$  approximation. So, that is that is the basic aspect of it we also know. So, one more this is the I am sorry this is the square root of the model noise ah covariance we may only know the rank  $q$  rank  $q$  of this. So, this may be. So, I may only know the rank  $q$  of this I may only rank  $p$  of this. So, I have to deal with I have to deal with a rank  $p$  approximation here I have to deal with a rank  $p$  approximation here and the rank  $q$  approximation in here. So, totally this matrix  $\mathbf{S}_k$  plus one it becomes a matrix of size  $N$  by  $p$  plus  $q$  the reduced rank approximation.

But our calculations are always rank  $p$ . So, how do you reduce a matrix of size  $N$  by  $N$  plus  $p$  to  $N$  by  $p$  matrix that is what is called the reduction this reduction can be easily implemented by.

(Refer Slide Time: 83:43)

• Rank reduction process

A:  $p + q < n$

B:  $p + q > n$

• Case A:  $p + q < n$ , SVD

• Step 1:  $V \in \mathbb{R}^{(p+q) \times (p+q)}$  be an ortho. Matrix of eigenvectors of  $S_{k+1}^T |_{(p+q) \times n} S_{k+1} |_{n \times (p+q)} \in \mathbb{R}^{(p+q) \times (p+q)}$

$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{p+q} > 0$

$S_{k+1}^T S_{k+1} = V \Lambda V^T$

GRAMMIAN

So, this is what is called the rank reduction process. Sorry this is what is called the rank reduction process. The rank reduction process depends on whether  $P$  plus  $Q$  is less than  $N$  or  $P$  plus  $Q$  is greater than  $N$ .  $P$  is the rank of the analysis covariance matrix,  $Q$  is the rank of the square root of the analysis process noise.  $P$  plus  $Q$  are 2 integers.  $P$  and  $Q$  are 2 integers. So,  $P$  plus  $Q$  could be less than  $N$  or greater than  $N$ . I am going to consider case a, where  $P$  plus  $Q$  is less than  $N$ . So, what is that we are going to do. We are going to fall back on singular value decomposition.

Now, you know why we did all the singular value decomposition at great length, why we did cholesky decomposition at great length. We did eigenvalue decomposition at great length, why if you understand these matrix methodology, very well the underlying basic principle sub matrix algorithm. They are used repeatedly in any advanced methods for dynamic data assimilation, and data assimilation in a stochastic context are in general advanced methods, and that is why we use repeatedly several of the algorithms from matrix theory.

So, what is the step 1. The step 1 is let. So, we already have  $A_h$  a matrix, which is  $N$  by  $N$  plus  $P$  plus  $Q$ . let  $V$  be this matrix be an orthonormal matrix of eigenvectors of  $S_{k+1}^T S_{k+1}$ . We are essentially doing the same singular value decomposition. Please remember that I would like to relate some basic ideas.

In our static case, we had  $H$  is equal to  $m$  by  $n$ . So, we had 2 kinds of hessian matrix, grammian matrix  $H^T H$  and  $H H^T$  by doing an eigenvalue decomposition of this. I derived an eigenvalue decomposition of the Do of the one, using  $S V d$ , the same principle is applied here, but  $H$  is replaced by  $H$  is going to be a replaced by the matrix  $S_{k+1}^T S_{k+1}$  is the matrix, that we have defined in the previous page  $S_{k+1}$ , defined in page 25 is this matrix. So, by replacing  $H$  by  $S_{k+1}$ . I can do all the analysis, that I did for  $H$  which are mathematically similar.

So, by doing the eigenvector eigenvalue decomposition for this grammian. Let these be the eigenvalues, let this be the eigenvalue decomposition. So, once I know. So, once I know this, I am sorry. Once I know the eigenvalue decomposition  $H^T H$ , I can derive the eigenvalue decomposition of  $H$ ,  $H^T$  using SVD.

(Refer Slide Time: 86:39)

• **Step 2:**  $S_{k+1} S_{k+1}^T \in \mathbb{R}^{n \times n}$  and  $S_{k+1}^T S_{k+1}$  have the same set of eigenvalues.

Let  $V = S_{k+1} V \Lambda^{1/2}$  - matrix of eigenvectors of  $S_{k+1} S_{k+1}^T$

Then  $S_{k+1} S_{k+1}^T = V \Lambda V^T$

$$= (S_{k+1} V \Lambda^{-1/2}) (\Lambda^{-1/2} V^T S_{k+1})$$

$$= (S_{k+1} V) (S_{k+1} V)^T$$

Let  $V(1:p)$  be the matrix of the first  $p$  columns of  $V$

$\Rightarrow S_{k+1}^f(1:p) = S_{k+1} V(1:p)$  is the required rank  $p$  approximation

*Handwritten red notes:*

- Diagram showing  $S_{k+1}^T S_{k+1}$  and  $S_{k+1} S_{k+1}^T$  with arrows pointing to "SVD".
- Red box around  $S_{k+1}^f(1:p)$  with arrows pointing to  $n \times (p+1)$ .

And that is exactly what I am trying to do. I have already known the eigenvalue decomposition. I have already known the eigenvalue decomposition of  $S_{k+1}^T S_{k+1}$ . So, I am now going to talk about  $S_{k+1} S_{k+1}^T$  from here to here, using the standard rule of SVD.

. So, these 2 matrices by theory have the same eigen, same set of nonzero eigenvalues. I can define a matrix like this matrix of eigenvectors of that. if I multiply these 2 I get this from there. You can readily verify that one. Therefore, the matrix  $V$  which is defined in here. The first  $P$  columns of that matrix are the. So, this is the. So, this is the whole

matrix. This is the first P columns of this matrix. It can be shown that these 2 matrix relations are, is the rank P approximation.

. So, the rank P approximation matrix for the forecast I want, is given by multiplying  $S_k$  plus 1 which I already have, which is a matrix of size N by P plus Q by this. So, this is the required rank P approximation.

(Refer Slide Time: 88:03)

- Case B:  $(p+q) > n$ 
  - $V \in \mathbb{R}^{n \times n}$  and  $\Lambda = \text{Diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$   
 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$   
be the eigen vectors/values of  $S_{k+1} S_{k+1}^T$   
 $S_{k+1} S_{k+1}^T = V \Lambda V^T = (V \Lambda^{1/2}) (V \Lambda^{1/2})^T = (\bar{V}) (\bar{V})^T$   
 $S_{k+1}^f(1:p) = \bar{V}(1:p)$
  - So,  $S_{k+1}^f(1:p) = \Pi_p(S_{k+1})$

Case B is similar, I do not want to go over the detail by reciting the whole thing. I would leave the case 2 as an exercise for you; that means, I can do the same reduction in the rank by this process.



(Refer Slide Time: 88:19)

- D.A. Step: This is very similar to the full rank counterpart in KF-SQRT VERSION
- (1) Compute  $A = (H_{k+1} S_{k+1}^f(1:p))^T \in \mathbb{R}^{p \times m}$
- (2)  $B = (A^T A + R_{k+1})^{-1} A^T \in \mathbb{R}^{m \times p}$
- (3) Find C:  $CC^T = I - AB$
- (4)  $K_{k+1} = S_{k+1}^f(1:p) A (A^T A + R_{k+1})^{-1} = S_{k+1} B^T$
- (5)  $\hat{x}_{k+1} = x_{k+1}^f + K_{k+1} [z_{k+1} - H_{k+1} x_{k+1}^f]$
- (6) The rank p s.q. root of  $\hat{P}_{k+1}$  is given by  
 $\hat{S}_{k+1}(1:p) = S_{k+1}^f(1:p) C \in \mathbb{R}^{n \times p}$   
 $(\hat{x}_{k+1}, \hat{S}_{k+1}(1:p))$  is the result

SEE ALSO  
COV. SQ RT. FILTERS

Now I am going to come back to the data assimilation step in the last step, in this process of a reduced rank implementation.

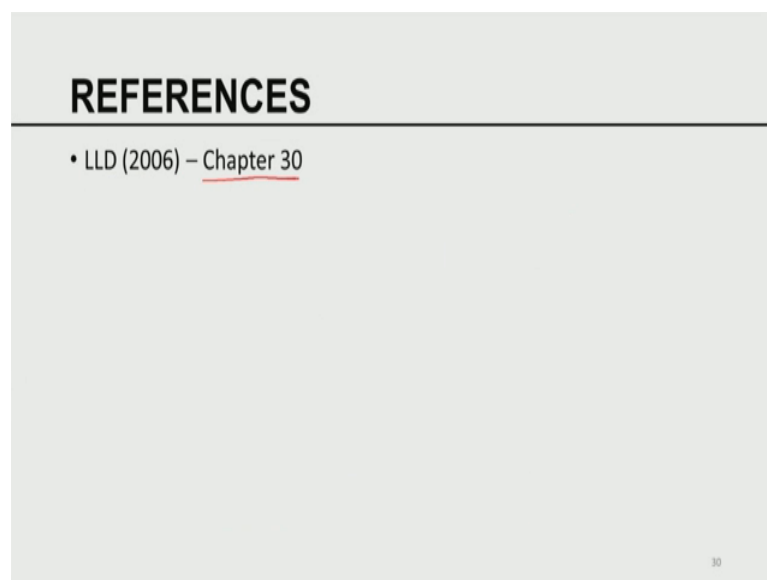
. So, the data assimilation step. This is very similar to the full rank approximation, very similar to the full rank counterpart. I do not have to say in counterpart in Kalman filter square root version. It was correct version. So, I am going to summarize the algorithm compute A H is known S<sub>k</sub> plus 1 f is known; that is a square root rank P square roots. Now if you make P is equal to m, I get that complete full rank filter. If you pick P to be anything in between 1 and 1 and 1, and then I get the reduced rank filter. So, this algorithm is so beautiful you can by bellowing in and out, the value of P you get quite a variety of approximation. So, this algorithm gives you not one approximation, but a family approximation. The family essentially parameterize by the value of P and P lies in the interval 1 to n.

So, depending on how much money you can afford to spend, depending on how much accuracy you want, depending on contingent of the properties of the behaviour. The eigen values you can create a wide family of approximations. So, you can compute this matrix, then you compute B, you compute C. We have already seen these calculations in the context of porters implementation of the square root covariance filter. So, see also covariance square root filters in the similar thing arise, we do exactly the same kind of computations.

So, I can now give an expression for S reduce rank approximation to Kalman gain. If I have a reduced rank approximation, the Kalman gain I have a reduced rank update. Please understand here, I am using only one observation, because there is no ensemble in here. It is simply a deterministic reduced rank improvement as opposed to ensemble methods for reduce rank implementation, which could be either stochastic or deterministic.

So, the rank P square root of  $P_k$  plus 1 is given by this, which is given by C, and this is the end result. So, I have gone from A forecast step to the analysis step. So, that completes the cycle in the computation of the filtering equations.

(Refer Slide Time: 91:13)



This material is derived from chapter 30 of our book, where we discussed many of these things in great detail. Yes, the analysis of case B I did not do it explicitly, but it simply involves some other types of matrix manipulations. The matrix algebra is the key in here. So, with this, what is there we have accomplished. We have accomplished quite a few methods for doing data assimilation in stochastic dynamic models. In a stochastic dynamic model can be linear, non-linear the stochastic dynamic model, may have model errors. The model errors are always represented by stochastic noise; that is the weakest link in this process.

If I know the model error, if I know how to correct the model level I would have corrected the model level. So, the corrected model does not have as much error as then

corrected model. So, when do you consider model, or to be random I have an easy feelings, that I have not incorporated several small scale processes, and when does that small scale processes, you neglect. Suppose you discretize a primitive equation model on a grid. We all know from basic theorem in sampling that, with a given grid length of size  $\Delta x$ . I cannot create samples of frequencies more than whose wave lengths are related to more than. I am sorry less than twice the twice the sample size. I am sorry twice the grid length.

So, sampling theorem essentially tells you when you try to describe as a continuous quantity. You may be missing out on high frequency or lower wavelength terms. So, the primitive equation model converts the whole spectrum of processes, when it discretized it, I truncate, I can resolve only frequencies up to a particular level. We cannot resolve processes involving higher frequencies.

Therefore, we have some idea that the neglected terms are high frequency terms. So, one way to approximate the unknown high frequency term, is to add a random noise. One way to approximate this is add a white noise. So, white noise addition of white noise to compensate the model error, is largely a reflection of our belief, that I have captured all the low frequency processes involved. It is the high frequency things, I am not able to capture, because of the finiteness of the grid size.

So, that is the credence that one can bring to bear for adding a stochastic noise; such as white noise to the model level observations are have always errors the. So, we have quite a variety of data assimilation problems. These problems were solved by Kalman in 1960 Kalman and BC in 1961 Kalman did it in continue discrete time. Kalman BC did it in continuing. I am sorry Kalman did in discrete time; Kalman BC did in continuous time. the discrete time derivation was perfect. The continuous time derivations used lots of heuristic principles of taking limits. It was soon realized the only way to derive mathematically, Kosher derivation for these non-linear filters, is with, is within the framework of what is called stochastic calculus or ETO calculus.

So, there is a parallel development of non-linear filtering within the stochastic calculus at stochastic modeling literature, for reasons that we do not have background in stochastic calculus. We restricted ourselves only to discrete time, we have given you the LQG problem complete solution Kalman filter. We have given a complete solution to the

discrete time, non-linear model non-linear observation, non-linear filter equations. We have given several different types of full rank approximation, using zeroth rank, first rank and second rank filters.

Then we talked about approximate reduce rank filters. We talked about reduce rank filters coming out of ensemble methods, that itself we have 2 versions; one by using random or virtual observation, another one is by changing a deterministic gain, which is different from Kalman filters. We also talked about deterministic reduced rank implementations of covariance square root based filters. So, I am assuming that this gives you the breadth and depth of the literature has come to be called filtering, within the context for data assimilation.

So, the Kalman filter even though the ideas were known as early as 1960 by 1969, 1970. The entire problem of non-linear carbon fill or non-linear filtering in continuous time using stochastic calculus was thoroughly done understood. So, they have put the problem is essentially solved, but implementation of those continuous time ideas still is a big challenge in principle non-linear problems, whether you attack it through continuous time formulation, or discrete time formulation continues to be a challenge in the computational world, with that is we conclude our discussion of data assimilation using non-linear filtering techniques.

Thank you