

# Foundation of Scientific Computing

Prof. T. K. Sengupta

Department of Aerospace Engineering

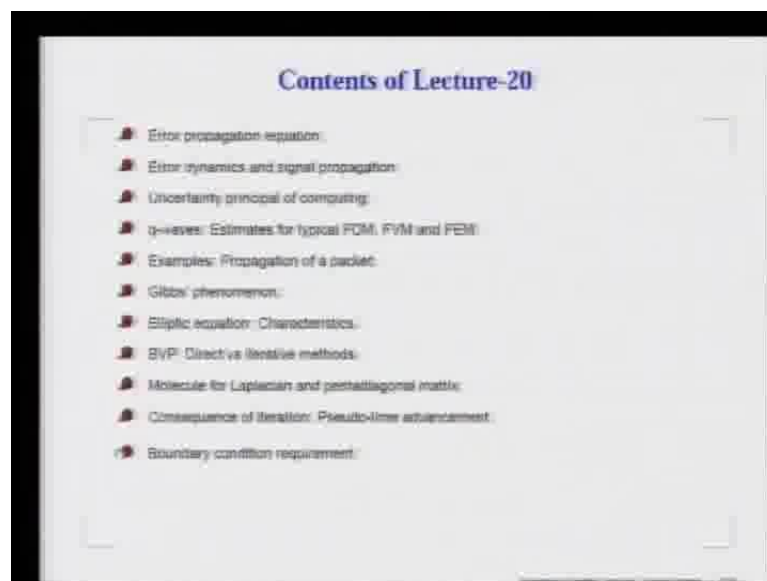
Indian Institute of Technology, Kanpur

Module No. # 01

Lecture No. # 20

In the 20th lecture, today we are going to talk about very important issue of error propagation in computing. This is what we have talked about here as error dynamics and signal propagation. This is central to most of the computing methods and we develop an alternative method or the correct method. That was proposed by Von Neumann.

(Refer Slide Time: 00:15)



In this methodology, we are going to find out that in all computing, because of invoking of numerical dispersion relation, we end up having dispersion errors, phase errors. This leads to what we call as the uncertainty principle of computing. We will show that it is only in the continuum limit we can perform absolutely correct computing; anywhere else, we will have to accept some amount of error. This is the essence of this new spectral theory of error propagation.

Once again we come back to discussion of q waves for various discretization methods. We once again talk about the finite difference method, finite volume and finite element methods. We pick up an example of propagation of a packet and we also show what is known as a Gibbs' phenomenon. This arises whenever we have solution discontinuity.

From the foot or the shoulder of this discontinuity, we see downstream and upstream propagating small scale oscillations and those are known as the Gibbs' phenomenon. This is a major issue about compressible flow calculations which have shock waves present, but, nonetheless, we will not talk about it, but, we should be aware of Gibbs' phenomenon as one of the major elements of scientific computing.

Having completed our discussion to some extent on parabolic equation and on the error propagation equation, we switch over to solving methodologies for elliptic equation. We notice that elliptic equations have characteristics which form conjugate pairs and essentially that require that the problem be treated as a boundary value problem or BVP.

Even in solving this boundary value problem, we have two alternatives: either we solve it directly - that would involve inverting very large dimensional matrix which becomes primitive in most of the cases; that is why we have to resort to the iterative methods. Once we start considering iterative methods, we have some interesting scientific possibilities; that is what we talk about by looking at the very basic equation - the Laplace's equation and identify its molecule.

We will see that this will amount to actually solving a linear algebraic equation with a pentadiagonal matrix which is not bounded together and that would exclude the possibility of having an analytic solution for this pentadiagonal matrix. So, we will have to resort to some iterative methods. These iterative methods will show, it is equivalent to invoking a pseudo time and that has a very interesting consequence because that pseudo time progression is related to the Eigenvalue and Eigen functions of that matrix upon discretization.

Another issue of elliptic equation is basically the requirement of boundary conditions. Elliptic equations have complex conjugate characteristics. So, essentially if we have an elliptic equation of order  $2n$ , we would require  $n$  boundary conditions. So, we are going to discuss quite a bit about this boundary condition requirements of elliptic PDE's.

Let me begin. Today, we wish to basically talk about a related topic that we had started discussing in the last few lectures. This relates again to the spectral analysis and we want to talk about a very important issue which was considered very very significant and it is still considered very significant.

(Refer Slide Time: 04:55)

### Error Propagation Equation

One-dimensional linear wave equation is given by,

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (1)$$

Define the computation error as,

$$e(x, t) = u(x, t) - \bar{u}_N \quad (2)$$

where  $\bar{u}_N$  is a general numerical solution of Eq. (1) and is given by,

$$\bar{u}_N = \int A_0(k) [|G|]^{\frac{t}{\Delta t}} e^{ik(x - c_N t)} dk \quad (3)$$

Numerical dispersion relation is given by,

$$\omega_N = c_N k \text{ and } c_N \neq c$$

It relates to understanding of a phenomenon, as to how error propagates. You see, the whole thing about numerical computation is about how you control error because accuracy is the main concern; we do not want error to hurt our numerical activity. So, we basically start off with the same model equation that we have been looking at. This has got certain features which we are all familiar with.

If this is our basic equation - point of reference to study error propagation, let us first define what constitutes error. Error would be defined as the exact solution minus the numerical solution; the subscript N refers to numerical solution. We have also noted in the last few lectures that the numerical solution for this particular equation would be determined by the initial spectrum as given by  $A_0(k)$ . The method that you choose would have its numerical amplification factor G. So, if I take its modulus and suppose we arrived at time t, after N such steps of  $\Delta t$ , then G would be raised to the power that N, or alternatively, we can write it as  $t / \Delta t$ .

Each step of time evolution gives rise to a phase shift; that we called earlier as the  $\beta_j$ , if you recall. Then, when I sum up all the phase shift that accounts for this numerical

phase speed, that we called as  $c$  of  $N$ . We correctly identified a numerical dispersion relation, which we show by the subscript  $N$ . It shows that numerical circular frequency is equal to the wave number times the numerical phase speed. One of the important finding that we have focused upon is the fact that this numerical phase speed is not equal to  $c$ . Not only that it is not equal to  $c$ , but, also it is not a constant here.

The physical phase speed is a constant;  $c$  is a constant. But  $c_N$  as we have noted that it is going to be a function of  $k$ . This leads to the dependence of  $c$  on  $k$ , and that in turn relates to numerical dispersion and that is why the  $(( ))$  this is a numerical dispersion relation.

(Refer Slide Time: 08:21)

#### Error propagation equation (Cont.)

$$\frac{\partial \bar{u}_N}{\partial x} = \int ik A_o(k) |G| \frac{t}{\Delta t} e^{ik(x-c_N t)} dk \quad (4)$$

$$\begin{aligned} \frac{\partial \bar{u}_N}{\partial t} = & - \int ik c_N A_o(k) |G| \frac{t}{\Delta t} e^{ik(x-c_N t)} dk \\ & + \int \frac{\ln|G|}{\Delta t} A_o(k) |G| \frac{t}{\Delta t} e^{ik(x-c_N t)} dk \end{aligned} \quad (5)$$

Error propagation equation is given by,

$$\begin{aligned} \frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} &= \frac{\partial(u - \bar{u}_N)}{\partial t} + c \frac{\partial(u - \bar{u}_N)}{\partial x} \\ &= \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} - \frac{\partial(\bar{u}_N)}{\partial t} - c \frac{\partial(\bar{u}_N)}{\partial x} \end{aligned} \quad (6)$$

Now, having accepted that expression for numerical solution as a consequence of the adopted numerical method, we can actually look at what this partial derivatives of the numerical solution with respect to  $x$  and  $t$ , amounts to. It is rather simple. Taking  $x$  derivative would be simply multiplying by  $ik$ . So, we are assuming that we are not making any dispersion - I mean any error, in evaluating the derivative. I am not writing  $ik$  equivalent; I am writing the best possible scenario, that is using a spectral method. If I use a spectral method in calculating this derivative, I will just simply take the Fourier amplitude multiply by  $ik$ .

If I start talking about a particular method, then I will replace this  $ik$  by  $ik$  equivalent; that I suppose all of us appreciate. Once I evaluate that derivative, the best possible way

as given here in equation 4. Next, you could do is - evaluate the time derivative of the numerical solution and because time appears in two places: One is of course,, in the phase definition; another one is  $\text{mod } G$  to the power  $t$  by  $\Delta t$ . So, we get two sets of terms.

Now, this is where I think we should allow our self to digress little bit into history. This is related to the Manhattan project. All these big guns of US science were in New Mexico trying to develop the nuclear bomb and one of the showmen was John Von Neumann; he is a Hungarian. He used to go by various names; in America, he was John; to his friend, he was Yuhang. So, John was a guiding spirit behind computing activity at Los Alamos.

So, in developing, of course,, he was also associated with the advanced science at Princeton; that big group which were involved in numerical weather prediction; they had multifarious interests; really polymath people - let us all tip our hat to them. They are real geniuses; probably the last bunch of geniuses got together in one place - that must be Los Alamos. There Von Neumann was looking at computing and he came out with this observation.

What seems very very intuitive and acceptable to all of us even today is that if I am looking at a linear system and if I am trying to compute a linear system, then it is natural for us to accept that the error committed also would be given by a linear system; it is common sense thing for us to accept, as the way we are trained in mathematics and physics all the time.

(Refer Slide Time: 11:51)

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$

$$\frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} = 0 \quad \text{X}$$

If that is the case, my governing equation is this. If this is my governing equation, what I would expect? The error should **that is by this right the error should** satisfy this. It took us almost 60 years to figure out that this is not correct.

I sent you the link for the paper that you could download, which actually came out in 2007. How do we come to the conclusion? Let us go through the steps slowly.

(Refer Slide Time: 12:41)

#### Error propagation equation (Cont.)

$$\frac{\partial \bar{u}_N}{\partial x} = \int i k A_a(k) [G] \frac{1}{\Delta t} e^{ik(x-c_N t)} dk \quad (4)$$

$$\begin{aligned} \frac{\partial \bar{u}_N}{\partial t} = & - \int i k c_N A_a(k) [G] \frac{1}{\Delta t} e^{ik(x-c_N t)} dk \\ & + \int \frac{\ln[G]}{\Delta t} A_a(k) [G] \frac{1}{\Delta t} e^{ik(x-c_N t)} dk \end{aligned} \quad (5)$$

Error propagation equation is given by,

$$\frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} = \frac{\partial(u - \bar{u}_N)}{\partial t} + c \frac{\partial(u - \bar{u}_N)}{\partial x} \quad (6)$$

$$= \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} - \frac{\partial \bar{u}_N}{\partial t} - c \frac{\partial \bar{u}_N}{\partial x}$$

So,  $e$  has been defined as departure of the exact solution from the numerical solution. So, I just simply replace  $e$  by  $u$  minus  $u_N$ . Then, of course,, you see it leads us to two sets of

terms. What about the first two terms? By definition, that is 0. So, I find that the error dynamics is not homogeneous. Like what we expected, the right hand side equal to 0, but, it is determined by this quantity. How the numerical solution is going to be? It is such a simple observation.

(Refer Slide Time: 13:26)

#### Error propagation equation (Cont.)

$$\frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} = 0 - \left( \frac{\partial(\bar{u}_N)}{\partial t} + c_N \frac{\partial(\bar{u}_N)}{\partial x} \right) + c_N \frac{\partial(\bar{u}_N)}{\partial x} - c \frac{\partial \bar{u}_N}{\partial x} \quad (7)$$

$$\frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} = - \left\{ \frac{\partial(\bar{u}_N)}{\partial t} + c_N \frac{\partial(\bar{u}_N)}{\partial x} \right\} - \left( 1 - \frac{c_N}{c} \right) c \frac{\partial(\bar{u}_N)}{\partial x} \quad (8)$$

From Eq. (4) and (5),

$$\begin{aligned} \frac{\partial \bar{u}_N}{\partial t} + c_N \frac{\partial \bar{u}_N}{\partial x} = & - \int i k c_N A_o(k) [|G|]^{\frac{1}{\Delta t}} e^{ik(x-c_N t)} dk \\ & + \int \frac{\ln|G|}{\Delta t} A_o(k) [|G|]^{\frac{1}{\Delta t}} e^{ik(x-c_N t)} dk \\ & + c_N \int i k A_o(k) [|G|]^{\frac{1}{\Delta t}} e^{ik(x-c_N t)} dk \end{aligned} \quad (9)$$

Now, if that is so, what I could do is I could write, it is just simple jugglery here, I have added and subtracted this term so that we have  $\frac{\partial u}{\partial t}$  and  $\frac{\partial u}{\partial x}$  minus  $c \frac{\partial u}{\partial t}$  and  $c \frac{\partial u}{\partial x}$ . These two are going to cancel each other.

So, basically, a little bit of reorganization will tell us that we have inhomogeneous equation. The first part relates to what we do numerically. Now, you notice that I have purposely replaced  $c$  by  $c_N$  because that is what we are playing with. Then, in addition we have this term. Since we have already evaluated the temporal and the spatial derivatives, we can put them in and collate. So, we are going to get these three subset terms. It is easy for you to understand that this comes from the  $x$  derivative; the next two terms come from the time derivative. So, we then substitute this, and this is what we get.

Now, what we did in the previous slide, I could see that here  $c_N$  is inside the integral, here the  $c_N$  is outside the integral. So, what I could do is I could integrate this by parts because  $c_N$  and the other thing is also function of  $k$ . So, I do that and then I would be able to cancel out this term. However, in the process of that cancellation what happens is we get this  $dc$  and  $dk$  term.

(Refer Slide Time: 15:16)

### Error propagation equation (Cont.)

$$\begin{aligned} \frac{\partial \bar{u}_N}{\partial t} + c_N \frac{\partial \bar{u}_N}{\partial x} = & \int \frac{\ln|G|}{\Delta t} A_o(k) [|G|]^{\frac{t}{\Delta t}} e^{ik(x-c_N t)} dk \\ & + \int \frac{dc_N}{dk} \left\{ \int ik' A_o(k') [|G|]^{\frac{t}{\Delta t}} e^{ik'(x-c_N t)} dk' \right\} dk \end{aligned} \quad (10)$$

Substituting Eq.(10) in Eq.(8) we obtain,

$$\begin{aligned} \frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} = & - \int \frac{\ln|G|}{\Delta t} A_o(k) [|G|]^{\frac{t}{\Delta t}} e^{ik(x-c_N t)} dk \\ & - \int \frac{dc_N}{dk} \left\{ \int ik' A_o(k') [|G|]^{\frac{t}{\Delta t}} e^{ik'(x-c_N t)} dk' \right\} dk \\ & - \left(1 - \frac{c_N}{c}\right) c \frac{\partial(\bar{u}_N)}{\partial x} \end{aligned} \quad (11)$$

You see this is something we have noticed just a while ago;  $c_N$  is not a constant; it is a function of  $k$ . So, we sourced it as the dispersion error. So, dispersion error directly appears in the set of time. This of course,, comes from the time derivative term. You can now see **what I have been shouting (( )) for last few lectures** that why we must have neutral stability. Now, do you see what I mean by having a neutral stability?

If I have a neutral stable method, then what happens?  $\text{Mod } G$  is 1 and  $G$  is 0. So, you do not accumulate any error if you have a neutrally stable method. So, what happens is you substitute all of these and you get these three sets of terms.



(Refer Slide Time: 16:24)

**Correct governing equation for error**

- Subtracting the governing equation for the numerical solution from the exact, after some manipulation, gives:

Phase error and solution discontinuity

Dispersion error

$$\frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} = -c \left[ 1 - \frac{c_N}{c} \right] \frac{\partial u_N}{\partial x} - \iint \frac{dc_N}{dk} \left[ ik' A_0 |G|^{t-\Delta t} e^{ik'(x-c_N t)} dk' \right] dk$$

$$- \int \frac{Ln|G|}{\Delta t} A_0 |G|^{t-\Delta t} e^{ik(x-c_N t)} dk$$

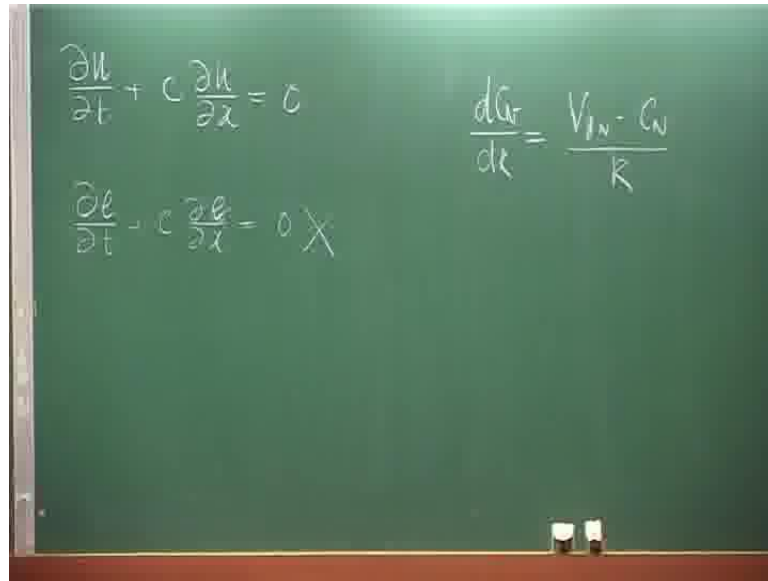
Numerical stability/instability

This is what we get with such a simple common sense observation. So, what we could do is we could club these 3 terms and identify them. I find that the error dynamics is driven by 3 sources of terms; this is something that we are not talking about any specific method being chosen; this is a generic cause; this is the disease of numerical computing. So, whenever we do numerical computing, we are going to suffer by these 3 sets of terms. The first set of term is pretty much obvious because this relates to phase speed; so, we call it as phase error. You also notice that this phase error term actually depends upon the smoothness of your actual solution - numerical solution. With the problem that you are solving, that solution is discontinuous; that means this derivative is large; ~~del~~ x of  $u_N$ . Then, of course,, this is going to hurt you, but, if I have a very smooth function where this quantity itself is small, you may not see its effect very much.

So, this phase error is something that you are going to see, what we talked about when we started talking about ODE's. Do you recall? That is part of your first assignment also. You saw that whenever you have large solution gradients, you have to be able to resolve it and this equation and this term tells you that those kinds of gradients actually contribute a lot to the error. So, that is that part.

The second part is pretty much obvious. We call it dispersion error because of this term  $dc$  and  $dk$ .

(Refer Slide Time: 18:27)


$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$
$$\frac{dg}{dk} = \frac{V_{gN} - c_N}{k}$$
$$\frac{\partial \phi}{\partial t} - c \frac{\partial \phi}{\partial x} = 0 \quad \text{X}$$

I have not done it. I will leave it to you to work out show this to be equal to proportional to  $V_{gN} - c_N$  by  $k$ . So, you should be able to show that this term (Refer Slide Time: 18:45) is given by the departure of the numerical group velocity from the numerical phase speed.

If you are looking at the original physical problem, both of them are the same;  $V_g$  equal to  $C$ . So, you do not have it. So, you can see that this term is identically 0 for exact solution because  $V_g$  equal to  $C$ ; it is a non-dispersive system. But in this case, what happens? We found out  $C_N$  is a function of  $k$ ; that is why this is non 0. That is caused by this the departure between the numerical group velocities from the numerical phase speed scaled by the wave number  $k$  itself. So, this is the second source of error that you would be concerned with.

(Refer Slide Time: 19:27)

**Correct governing equation for error**

- Subtracting the governing equation for the numerical solution from the exact, after some manipulation, gives:

Phase error and solution discontinuity

Dispersion error

$$\frac{\partial e}{\partial t} + c \frac{\partial e}{\partial x} = -c \left[ 1 - \frac{c_N}{c} \right] \frac{\partial u_N}{\partial x} - \int \frac{dc_N}{dk} \left[ ik' A_0 |G|^{t/\Delta t} e^{ik'(x-c_N t)} dk' \right] dk$$

$$- \int \frac{Ln|G|}{\Delta t} A_0 |G|^{t/\Delta t} e^{ik(x-c_N t)} dk$$

Numerical stability/instability

The third one is what we have been repeating time and again that if you have anything other than neutral stability, it is going to contribute to that. So, actually you can see that this analysis or this result that is projected in front of you would be colored by the choice of your method. If I focus upon a particular method, I should be able to figure out what these each quantities are. Then, I can figure out what is the actual error for that particular method is, but, this is the most generic description of what we get.

(Refer Slide Time: 20:17)

**Error Dynamics & Signal Propagation**

- The error equation for (1), shows the von Neumann stability and error analysis to be wrong.
- Von Neumann's assumption that error and signal to follow each other seems *natural* for linear dynamical system.
- We have established that the error is driven by the following:
  - i) **Phase error**
  - ii) **Dispersion error**
  - iii) **Error due to stability/ instability – zero for neutrally stable algorithm.**

But can we control phase and dispersion error completely?

**Uncertainty Principle of Computing:** Completely error free computations are possible only in the continuum limit of

$$k \Delta x \rightarrow 0; \text{ and } \omega \Delta t \rightarrow 0$$

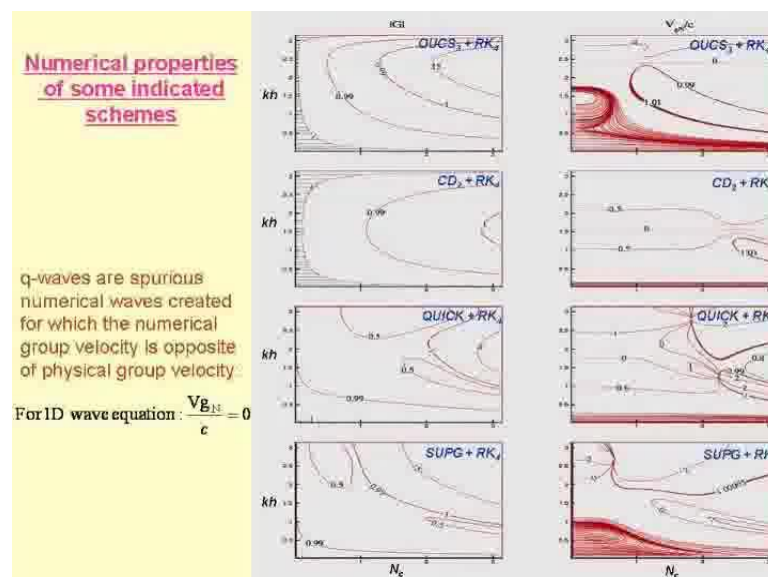
I have put it as a summary. Of course, the Von Neumann stability and error analysis is wrong. Unfortunately, this is what you find in all text books; even including the book that I had written in 2002 or 2003.

We assumed it is so intuitive for us to accept that the linear dynamics should follow that we also fell into the same trap. However, we have shown here that error is driven by the following three sources of error: the phase error, the dispersion error, and the error due to stability or instability. This component is of course, 0 for neutrally stable algorithm. This is the question that we should ask our self - can we really get rid of phase and dispersion error? I have a pessimistic message for all of you – No. That is why we coin this term Uncertainty Principle of Computing.

Whenever you compute, you cannot ever assure that you can take care of these, until and unless you have a perfectly periodic problem and you can adapt a spectral method. So, what happens is most of the practical problems would not let you use spectral method. Then you come to this pessimistic conclusion that we cannot perform completely error free computations.

It is only possible when you go to the origin of that point k (( )) plane where everything is perfectly kosher and nice. So, that would imply that you will have to take delta vanishingly small so that you are going to the continuum limit of k delta x going to 0 and omega delta t going to 0.

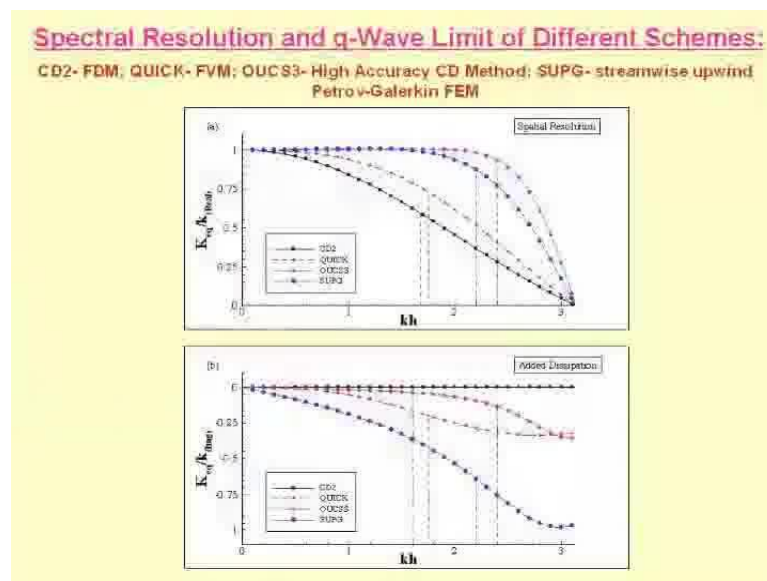
(Refer Slide Time: 22:23)



So, this is what we have seen. Now, I did talk about this part  $g$  and  $V_g$  for these four classes of methods. We picked up from little advanced finite difference calculation; this is a classical finite difference calculation; this is some finite volume calculation and this is a finite element calculation.

I brought to your attention, the existence of this line  $V_g$   $N$  by  $c$  equal to 0, above which you will see  $V_g$   $N$  by  $c$  is negative. These components of error or the solution - we will call them as  $q$ -waves; they are again spurious; they do not belong physically; they are attributes of the numeric. These are those solution components which actually have negative group velocity; they are opposite of the physical group velocity and for 1-D wave equation, 0 is the demarcation line. We can actually catalog what happens to this  $q$ -waves business.

(Refer Slide Time: 23:41)



Now, I have purposely projected this in our  $k$  equivalent by  $k$  solution versus  $kh$  plane; this is the real part and this is the imaginary part. The real part determines the phase of the solution. The imaginary part tells us whether we are adding a numerical dissipation or anti diffusion.

Now, on these two frames, we have identified locations - the values of  $kh$ , above which for that particular method actually has  $q$  waves. You can see that the modern the most recent and the most successful finite difference method that some of us use, gives you the highest  $q$  wave limit; that is roughly around 2.4. So, even with the best possible

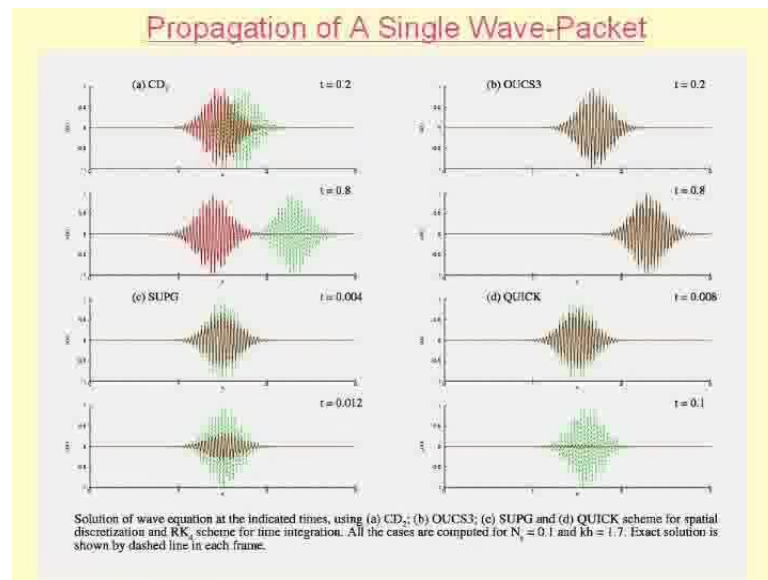
method we are having today, we notice that any  $kh$  component above 2.4 has this unphysical attribute where the solution will propagate. This has got nothing to do with physics; this is totally a numerical error. Whereas, if you come to your CD2 method, which is the lowest possible curve - that happens at  $\pi/2$ . So, at least by going from the classical CD2 to the best possible method that we have, we could actually raise that limit from 1.57 to about 2.4. So, that is something. Now, despite that how come people have not located and reported  $q$  waves before?

Two things you see even at that range are: what happens? What does this real part do? Ideally, anything above this is severely attenuated; even at that point, this attenuation is roughly about 45 percent. So, every time step if I adopt CD2 method, I am actually attenuating the signal by this amount. So, that is about 45 percent. In addition to that the CD2 method does not have any numerical dissipation. So, that is why you have the flat equal to 0. However, this 45 percent attenuation itself will remove that quite effectively.

If some quiescent  $q$  waves are created, for values of  $kh$  above this, they are severely attenuated because of this filtering attribute of this discretization method. Whereas, if you take this good method which actually takes this limit from  $\pi/2$  to 2.4, you can see this part is attenuated by only 3 or 4 percent. So, this is what I was joking that sometimes your strength becomes your liability. You have a better method, but, that is why you get to see the  $q$ -waves. If you choose a bad method, you will not see it and those people will come and commensurate with you saying - bad luck; you should try better methods. So, this is a very very interesting observation. However, I must also point out that some of the other methods - the finite volume, finite element, and better finite difference method. We do add numerical dissipation and you can see what happens.

For example, this finite element method drawn by this blue line is the  $q$  wave limit and that is where we are adding this much of dissipation also; more than 0.5. So, what happens is those people who will be doing finite element method will say - this is a pigment of your imagination because we never see. They do not see it because they add so much of numerical dissipation; they damp out various components unknowingly. If they would have done it knowingly, I would have really acknowledged that. They probably do not even know what is a  $q$  wave; any way, that is different.

(Refer Slide Time: 28:33)



Now, this is the summary sheet for trying to solve the problem. Basically, once you know what the problem is, you can set up critical tests. So, we found out that for the CD2 method, the demarcation feature was  $\pi$  by 2. If I now take a wave packet with a  $kh$  greater than  $\pi$  by 2, then what will happen to CD2 method solution? The whole wave packet will go in the opposite direction. So, that is one thing.

What we have done is - we have chosen a wave packet with  $kh$  equal to 1.7, purposely to test out all these four methods. So, the value of  $kh$  is equal to 1.7 and we have taken a quite a moderate value of  $N$  c. Now, what happens? Let us solve this and show the exact solution by the green lines. This green wave packet is the exact solution which is going to the right because we have taken  $c$  equal to 1. So, the solution is going between these peaks. The difference is about 0.2 here and it keeps simple.

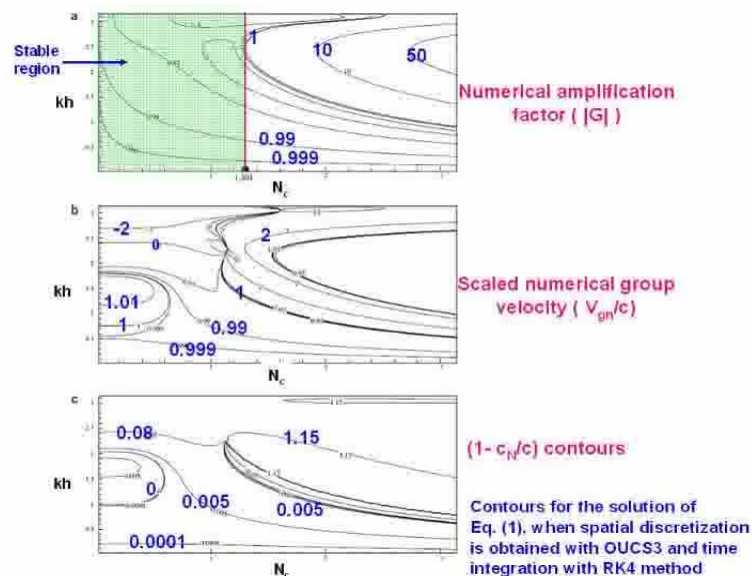
Now, the CD2 method actually sends the whole packets to the wrong direction and it is a non-dissipative method; so, it does not dissipate. So, you retain the identity of the packet, but, it makes it go in the opposite direction. This is the method that we will be talking about in this class. This is what we have developed our self. You cannot distinguish between the exact and the numerical solution. They are going together hand in hand; of course,, our method also will fail, if we choose a value of  $kh$  greater than 2.4.

So, we have to be very very careful in what conclusion we draw. Now, for the same problem, though if I use this finite element method stream wise upwind Petrov-Galerkin

method SUPG and look at the solution that is shown. It is 0.004 and 0.012. You can see the green signal and the red signal. The times are so low that you do not even see the convection, but, you can certainly see the effect of numerical dissipation. See this solution is virtually diminishing there.

This is the corresponding commercial software type of method, like **Fluent**. **Fluent** will do this with their best method. This solution is again shown at 0.008 and this as 0.1. You can see the signal has disappeared; it has all become quiet; that is why I said it is a quiet battle field. So, you want to see it little more graphically and this is what it is you can see an animation (Refer Slide Time: 31:52) - the top one is your exact solution, the middle one what we have carefully crafted, and you can see in this classical CD2 method, the whole packet is going to the left instead of going to the right. I suppose you appreciate what is q wave. Then, q wave is nothing but, an extreme manifestation of dispersion effect because we got it is a property related to  $V_g$ . This  $V_g$  is so bad that it actually takes you in the opposite direction. So, q wave is nothing but, the extreme manifestation of dispersion error.

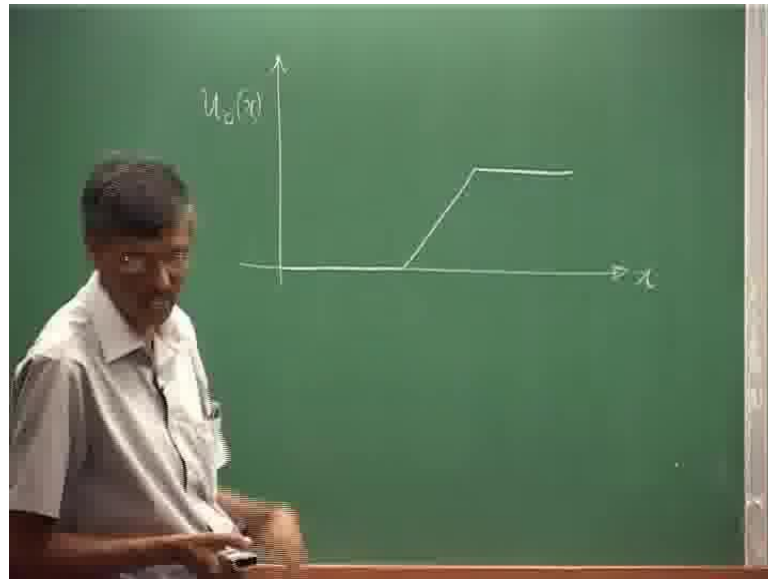
(Refer Slide Time: 32:36)



The next thing that I am trying to show you is, we have just now seen what happens because of dispersion error, how about the phase error or to understand what phase error does, we have again purposely designed a case where  $N_c$  has been chosen in such a way that I have  $g$  equal to 1.

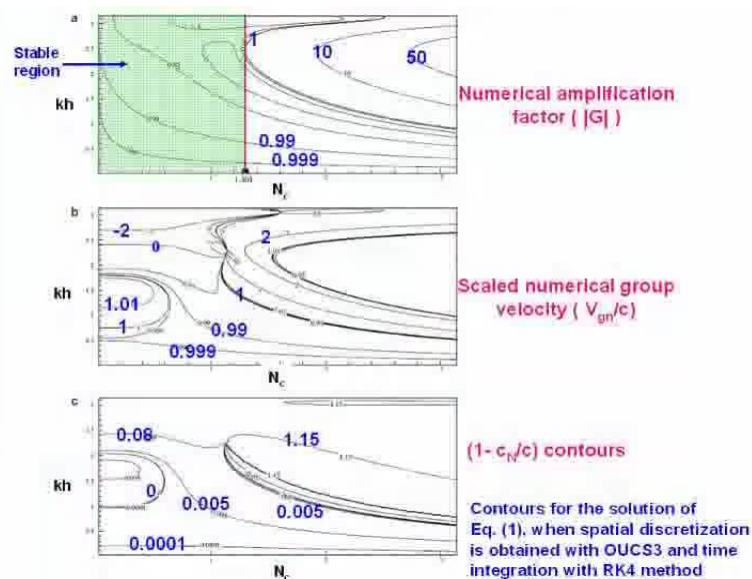


(Refer Slide Time: 33:26)



I have chosen a method here, for which this  $G$  is almost equal to 1. So, what happens here? We are trying to study the solution of this equation for a function whose initial condition is given like this. So, it has a ramp like this. So, there is a slope discontinuity here and there. Recall that phase error depends on  $\Delta x$  and  $u/N$ . So, if I set up a problem where I purposely have a slope discontinuity, I should be able to see its effect.

(Refer Slide Time: 33:50)

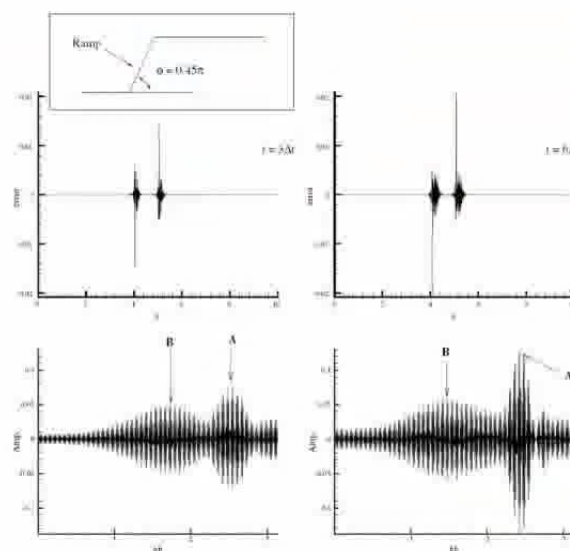


Now, these are the corresponding properties of the super accurate method that we have developed our self - that optimum scheme; these are some of the data which has been

highlighted by Yogesh here, for you to view that If I choose  $kh$  equal to 1.7 and  $N_c$  equal to 1.3. So, we are purposely doing a very large time step calculation -  $N_c$  equal to 1.3. Now, what happens when I have a solution like this? This is not solving a wave or a wave packet; this is not really monochromatic event; this is a real true polychromatic phenomenon because I have this  $u$  versus  $x$ , I can do the Fourier transform and I will see the whole range of  $k$  is excited. So, there is nothing that you can synthetically pick out one value of  $kh$ . So, this whole range of  $kh$  is coming into play here.

So, you have to remember that this is not like the previous case, wherein, for the wave packet we chose  $kh$  is equal to 1.7. For this problem, you do not have that luxury; the whole range of  $kh$  has been excited.

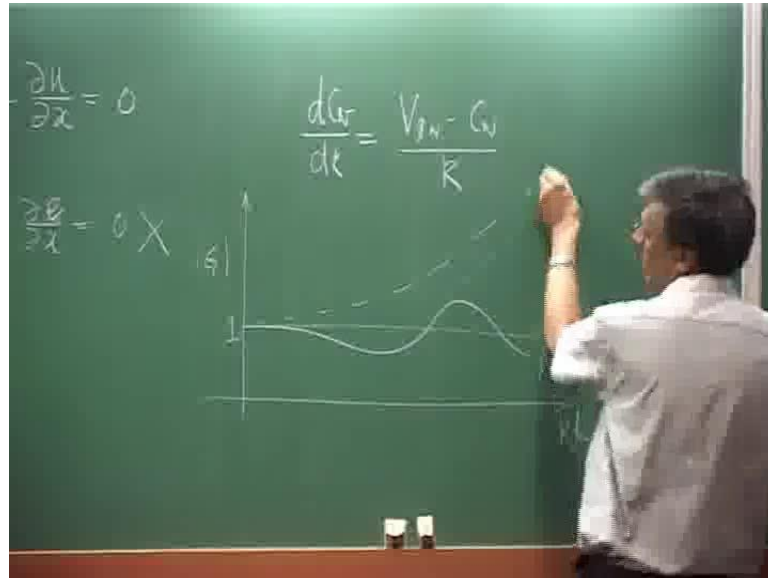
(Refer Slide Time: 35:17)



Now, if you solve this, the ramp actually propagates to the right. I have taken the ramp angle as less than 90 degree; so, it is about 0.45 pi. What happens is - at the foot and at the shoulder, because of the slope discontinuity, I pick up error; that error has been plotted here. They have been plotted at a very early time. This is about 3  $\Delta t$ ; this is at 6  $\Delta t$ ; so the story has just begun. I have this error; I can do a Fourier transform and I can plot it. This is what I get - the Fourier transform, Fourier amplitude versus  $kh$  has been plotted and I get this type of footprint. Now, what do you find? The error seems to have 2 maximum: one corresponds to a little higher wave number which I have marked by a, and one corresponds to little lower value.

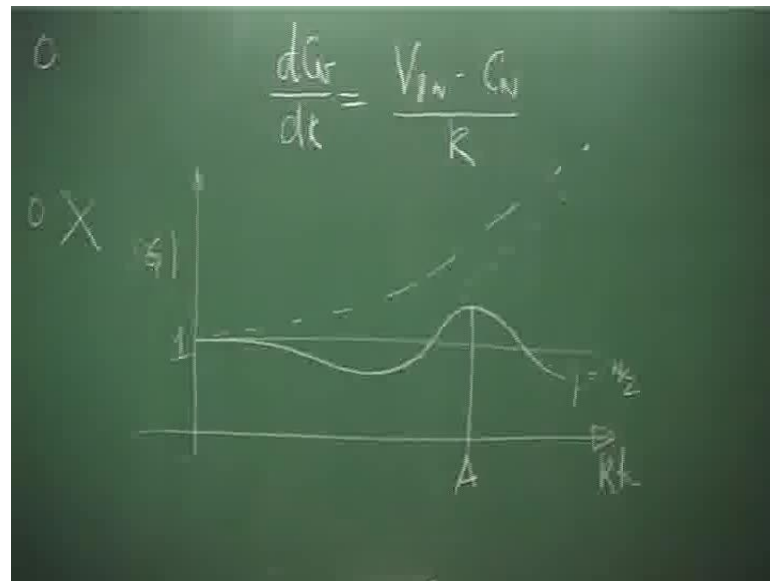
As I have identified to you, you can download the paper; if you do, you will notice what really happens and why we are seeing this; what are those two points a and b referred to as the source of those errors.

(Refer Slide Time: 37:00)



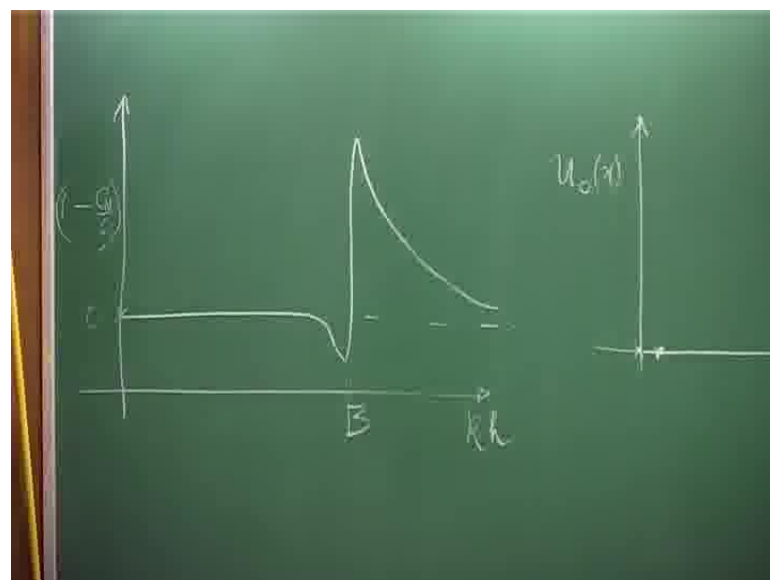
What we could do is - for the chosen value of  $N_c$ , I could plot mod  $G$ . Let us say, I have some value of mod  $G$  equal to 1. Then, we find that we are going to get the different values of  $G$  for different points. So, we are going to see something like this (Refer Slide Time: 37:28 to 37:53); some cases the error would even be going higher. So, this could be  $j$  equal to ... the middle point I am talking about. So, you can get this. In fact, if I have a first point here and if I look at the second point, I find that this part  $G$  actually goes like this and you know why? Because, the second point information has been brought in from ...).

(Refer Slide Time: 38:07)



That gives rise to a large source of error. So, this peak where  $G$  is greater than 1 and this point corresponds to  $A$  because that is where you get more than one.

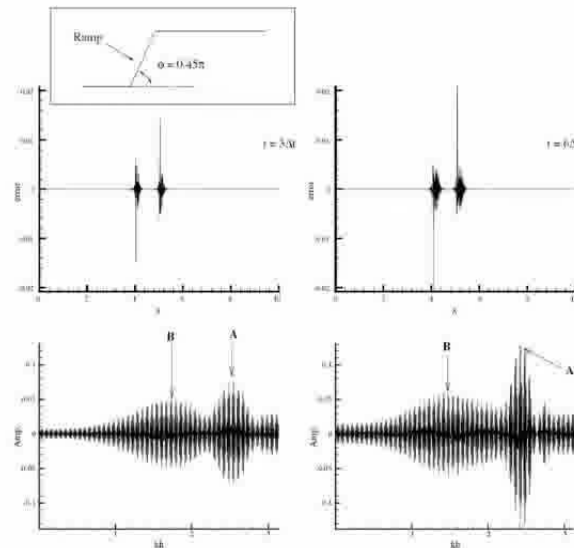
(Refer Slide Time: 38:38)



So, with time this peak in  $A$  is actually progressively increasing; that is one thing that you notice. In addition, if you have noticed that error revolution equation, we have a component which is proportional to  $1 - C_N$  by  $C$  and that I could plot versus  $kh$ . What is found? If this value is 0 here (Refer Slide Time: 38:57), for most of the points this remains 0, but, then at one point it actually comes down and burst upon  $u$  like this.

This point corresponds to B. So, what happens? This quantity remains benign for the whole range of  $kh$ , but, for a particular value of  $kh$ , it actually has a very sharp maximum and that maximum actually gives rise to this second peak and that also keeps growing.

(Refer Slide Time: 39:40)



If you look at this kind of a feature of the solution near discontinuity where solutions have overshoot and undershoot, this is what is called as Gibbs' phenomenon.

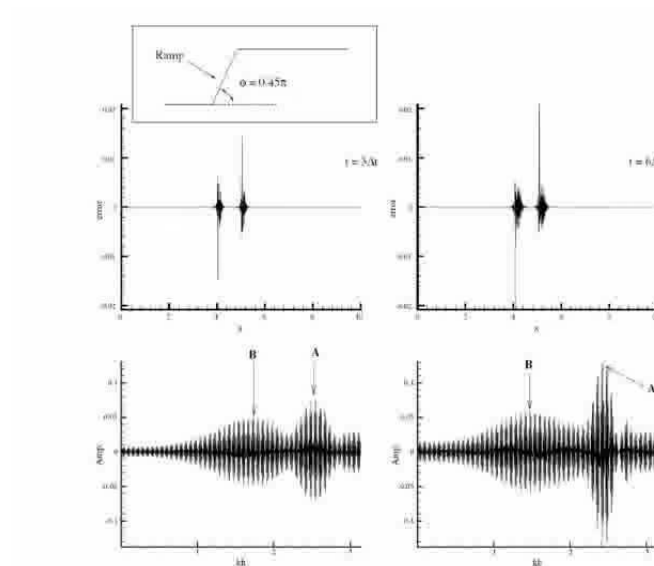
There are lots of myths in the literature about Gibbs' phenomenon. This is the same Gibbs which you may have come across in chemistry, thermodynamics. He was a very very prolific gentleman. Do you know he was the first professor of Engineering at Yale with the salary of 1 dollar; it was more an honour than a salary. So, Gibbs actually noted this and this is called as a Gibbs' phenomenon.

(Refer Slide Time: 40:47)



There are lots of myths, but, now you have an explanation - what causes Gibbs' phenomenon. That comes out from this term. So, you can understand whether it is numerical or not. This is the driver term for Gibbs' phenomenon.

(Refer Slide Time: 41:06)



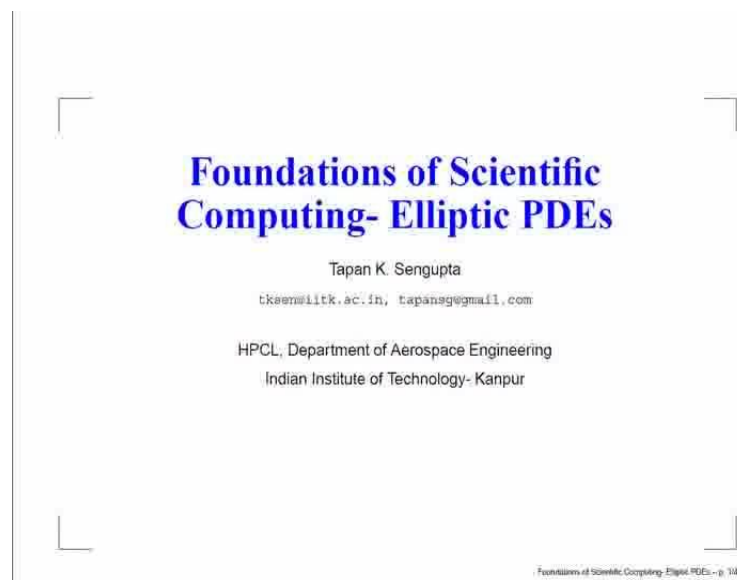
Of course, you will have to find out this undershoot and overshoot looks like a wave. When they do look like a wave, they also behave like a wave. They move and you got to find out what is the direction of propagation of these waves, by looking at the corresponding V-g plots. So, you might see, sometimes it will progressively only go in

one direction and not in the other direction. In this case, you can look at the paper and you will see that if I increase this ramp angle, I will make this thing worst. So, that also tells you the role of this term. So, if this term is made more adverse, you are going to see more error. This problem is not academic because this truly a problem of engineering dimension. Because if you are an aerospace engineer, you know that you do get shock waves. There, the solution actually jumps like the pressure, the velocity jump like this.

If you are designing a traffic system, you can come across traffic bottle neck; that is where also you get solution discontinued or hydraulic jump. We talked about other things of sonic boom etcetera; they all suffer from this issue of Gibbs' phenomenon. So, I wanted you to know about this; what this is; may be it is time that we should do a non-numerical home assignment. So, I would ask you to go back and write out the error revolution equation for the heat equation. We have done it here for 1D convection equation; so, you do it for the heat equation.

I will send all the email so that those who are not here, do not have an excuse for not submitting. So, everybody will get a chance to submit and get appropriate credit.

(Refer Slide Time: 43:24)



Now, we will go somewhere else. I get the feeling that we are going slightly slowly than what I would have, but, at the same time, I want to be with the whole class. So, please do interrupt me if you do not follow any of these things that I am saying here.

These are some of the things you will not find in any book. These are not some of the things which you have (( )) of not written completely. So, as we go along, I keep narrating things to you. So, feel free to interrupt me any point in time.

(Refer Slide Time: 44:31)

## Introduction – Theoretical Considerations

- In classifying quasi-linear PDE's, we have noted that elliptic PDE's have complex characteristics.
- Lack of real characteristics imply information to propagate at a point from its neighbours. Those neighbours in turn, depend upon their neighbours. This process goes on till the boundary points are reached.
- Thus, any interior point eventually depends upon boundary conditions. Thus, elliptic PDE's constitute the boundary value problem (BVP).
- This implies that theoretically an elliptic PDE has to be solved over the complete computational domain.
- After discretization of an elliptic problem, if we end up with a linear algebraic equation:

$$[A] \{X\} = \{b\} \quad (1)$$

Foundations of Scientific Computing: Elliptic PDEs – p. 381

Let us go back and see. This is a parallel track of development of CFD. Computing people have been solving heat equation, parabolic equation; people have been solving elliptic equations. These are the two major preoccupations of people, may be 100 years ago. Lots of the things belong to the realm of theory and you understand what their ramifications are.

We have noted - if we are looking at quasi-linear partial differential equations and if they happen to be elliptic PDEs, we have complex characteristics that render them as a boundary value problem, and the reason is that you do not have real characteristics. So, information does not necessarily have directionality, that it is approaching that point from one particular direction. It means that it can appear from everywhere. So, this sort of feature that you do not have a specific preferred direction makes it omnidirectional. So, every point is affected by every other point in this neighborhood. Those points are eventually influenced by their neighborhood. In the end, you hit the boundary. That is why these are called boundary value problem because everything is determined by the boundary conditions and nothing else, in the end. That is what we have written here.



Now, that also actually puts in a damper in all our activities; it means that if I am trying to solve elliptic PDE, unlike parabolic PDEs where we knew that information propagates along  $P$  equal to constant like in the heat equation. I could solve one timeline at a time and I can go forward right. So, that makes the job much simpler, but, because this is a boundary value problem, we will have to take the whole domain together and it makes the job little more involved. You have to solve the whole problem over the complete computational domain, and after you have done that you end up with the same thing -  $Ax$  equal to  $b$ .

Depending on whether you are choosing a very structured discretization or unstructured discretization, this  $A$  can have all kinds of features; for example, when we adopted that implicit method for heat equation. We saw that  $A$  happened to be a tridiagonal matrix. If it was a periodic problem, we had periodic tridiagonal matrix.

(Refer Slide Time: 47:13)

### Introduction – Theoretical Consideration

- Direct solution of (1) requires calculations of the order of  $N^3$ , where  $N$  is the rank of the matrix  $[A]$ . This is prohibitively expensive.
- For this reason, linear algebraic equations are often solved by iterative methods.
- The moot question remains: What happens to the equivalent governing differential equation for the iterative methods? Do they still keep the original elliptic nature of the governing equation?

Foundations of Scientific Computing: Elliptic PDEs - (1/34)

What happens to elliptic equations? We can have different natures of  $A$ . We will talk about few very simple cases.

We also know that to solve  $Ax$  equal to  $b$  by direct inversion, we would require  $N$  cube calculations and that is prohibitively expensive; very very bad thing to encounter. So, what we do is we circumvent the problem. Historically, this has been done ever since the time of Jacobi that you try to use iterative methods. So, you make some kind of initial

guess and then you keep on increasing the guess based on some algorithm or methods. Now, if we do that the main question - that remains.

We have started solving a problem which has elliptic nature. Now, the moment I choose on an algorithm, I had picked the problem; I have chased the problem. So, I could work out what is the equivalent equation. That is what you did in your mid-sem and you found out that you solved something else. So, this should alert us that we should not panic and give up, but, we should find out what it does.

(Refer Slide Time: 48:36)

### Iterative method- An Analysis

- Consider the solution  $u(x, y)$ , to be obtained by solving Laplace equation:

$$\nabla^2 u = 0 \quad (2)$$

- To solve the equation in a Cartesian grid with uniform spacing,  $\Delta x = \Delta y = h$ , one gets the difference equation as:

$$u_{ij} = \frac{1}{4} [u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}] \quad (3)$$

- For the ease of discussion, let the boundary conditions be given by Dirichlet condition on all the edges of the domain. Thus, Eq. (3) can be written down for all the interior nodes,  $i = 2$  to  $(N - 1)$  and  $j = 2$  to  $(M - 1)$ .
- This will constitute the linear algebraic equation given in Eq. (1), with  $[A]$  having the structure shown next.

Foundations of Scientific Computing: Elliptic PDEs - p. 641

For example, solving this Laplace's equation in a Cartesian grid with uniform spacing in both directions; **this amounts to this.**

This is a very interesting observation that  $u_{ij}$  is nothing but, the average of its four neighbors. What does it indicate? If I am trying to solve the Laplace's equation in a domain where would I get the maximum value of the solution, can it be in the middle? It has to be only at the boundaries; why because interior points are nothing but, averages. Average cannot be more than the constituents. It is a simple observation. So, that is the maximum principle of Laplace's equation.

Please do not confuse. I have seen places where people keep saying that this maximum principle is true for all elliptic equation; or even for Poisson equation, it is not true. Poisson equation is where you have a right hand side. If you have something on the right

hand side, this logic goes out through the window. So, do not try to do this. At the same time, if you are solving a Laplace's equation, and in the process of your solution, if you see somewhere in the middle of the domain, that the solution is becoming larger, you should stop here and go back to the drawing board, and see whether your code is right.

Now, let us also make our life easier by considering boundary conditions, given as Dirichlet conditions. Then, we will have to be solving problems for  $i$  equal to 2 to  $N$  minus 1 and  $j$  equal to 2 to  $N$  minus 1.

(Refer Slide Time: 50:51)

**Iterative Method- An Analysis**

Consider the linear algebraic equation (3), the matrix  $[A]$  has the following form:

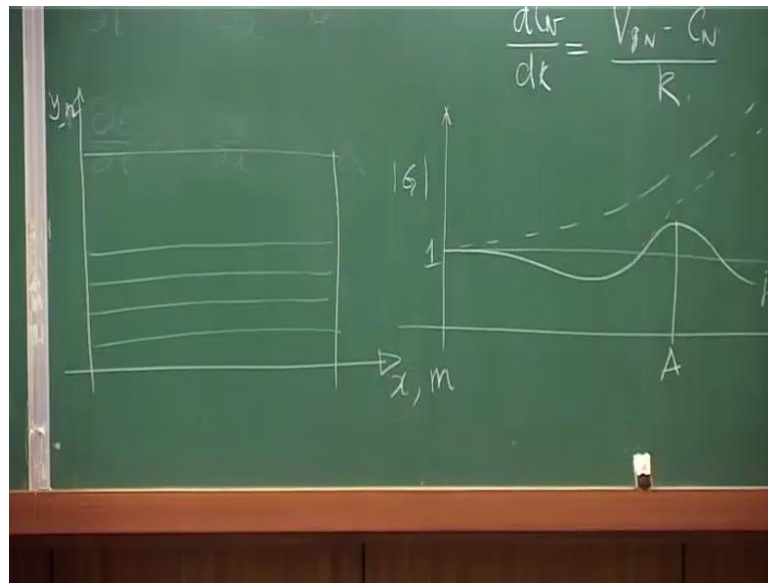
$$\begin{bmatrix} -4 & 1 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 1 & -4 & 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & -4 & 1 & 0 & \dots & 0 & 1 & \dots & 0 \\ 0 & 0 & 1 & -4 & 1 & \dots & 0 & 0 & 1 & 0 \\ \dots & \dots & \dots & 1 & -4 & 1 & \dots & 0 & \dots & 1 \\ 1 & \dots & \dots & \dots & 1 & -4 & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & \dots & \dots & 1 & -4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & \dots & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 1 & \dots & \dots & 0 & 1 & -4 \end{bmatrix}$$

This is a penta-diagonal matrix. This sparse matrix is not directly invertible- Unlike tridiagonal matrix.

Foundations of Scientific Computing: Elliptic PDEs - p. 541

Now, for this problem, as we have written in 3, we get this structure of the matrix. The  $A$  matrix looks like this. The diagonal term comes from that  $4 u_{ij}$ , the minus sign; then you have 1 point to the left, 1 point to the right coming from the  $x$  derivatives.

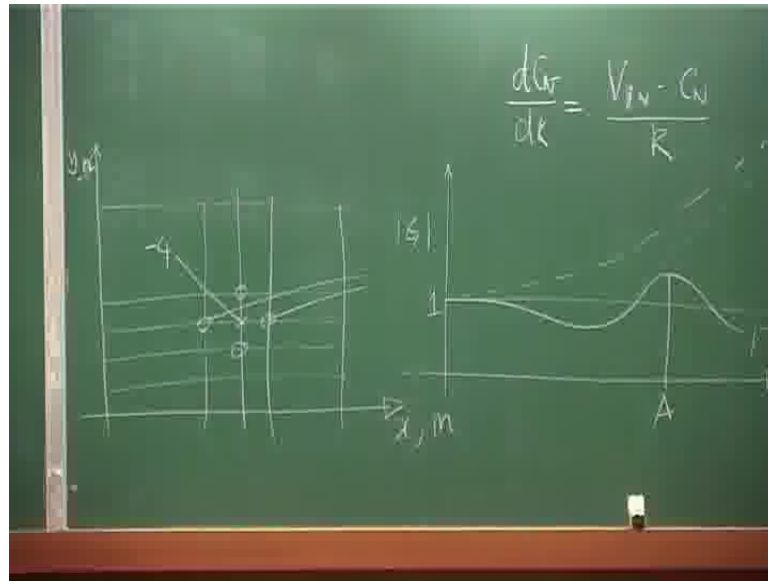
(Refer Slide Time: 52:23)



Please be aware of the fact that if I do not say anything specifically, we are always going through this. So, if I talk about this  $x$  or let us say  $I$  or  $m$  index and if I talk about  $y$  or  $n$  index, we will be identifying a domain and then, we will be following this sequence from left to right and from bottom to top. This is what is called as a lexicographic pattern. What is lexicographic pattern? That is what you do in reading books. Different languages have different way, but, they still do have the structure. So, either you start from left to right, or from bottom to top; that is how we read books.

Here, we are going to follow this term left to right and bottom to top. Imagine if the books are to be written in this fashion, all of us would take some time getting used to reading from the last line and going to the top of the page.

(Refer Slide Time: 52:46)



Anyway, in computing that is what we do. Then, if I am looking at one particular point, here then I need four neighbors; so, this point gives me that minus 4 - the diagonal entry; this point and this point (Refer Slide Time: 53:04) because we are going from left to right; so, they have contiguity. They follow each other; those are the ones that you are seeing here - plus one here and a plus one there. Now, this is really the neighboring points.

What about this? This point has happened a pitch before. So, these are those entries which are corresponding to  $j$  minus 1;  $j$  plus 1 would be given by these diagonal entries. So, these are once again called penta diagonal matrix because you have 5 diagonals; but, they are not packed together.

(Refer Slide Time: 53:52)

## Iterative Method- An Analysis

- Consider the linear algebraic equation (3), the matrix  $[A]$  has the following form:

$$\begin{bmatrix} -4 & 1 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 1 & -4 & 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & -4 & 1 & 0 & \dots & 0 & 1 & \dots & 0 \\ 0 & 0 & 1 & -4 & 1 & \dots & 0 & 0 & 1 & 0 \\ \dots & \dots & \dots & 1 & -4 & 1 & \dots & 0 & \dots & 1 \\ 1 & \dots & \dots & \dots & 1 & -4 & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & \dots & \dots & 1 & -4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & \dots & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 1 & \dots & \dots & 0 & 1 & -4 \end{bmatrix}$$

- This is a penta-diagonal matrix. This sparse matrix is not directly invertible- Unlike tridiagonal matrix.

Foundations of Scientific Computing: Elliptic PDEs - p. 544

If these two lines - this diagonal and that diagonal were sitting next to those, other three lives would have been a lot simpler because like Thomas algorithm, we should be able to write out an exact solution procedure. But unfortunately, because of this intervening 0s, this is one of the vexing problems - that has baffled applied mathematicians for long time. So, we cannot directly invert this matrix, although it is sparse. So, this is something we should keep in mind that this direct inversion is not possible

(Refer Slide Time: 54:37)

## Iterative Methods- An Analysis

- Solution of (3) takes very large computing time- If the size of matrix is large. In actual time-dependent calculations, one requires to solve (3) at each time step and direct methods are prohibitive.
- For this reason, one would solve (3) in its discrete form by various approximate means.
- One of the oldest and classical method is due to Jacobi (1844). This is also referred to as Richardson's method.
- Here, a point-by-point iterative method is used:

$$\frac{[u_{i+1,j}^{(n)} - 2u_{i,j}^{(n+1)} + u_{i-1,j}^{(n)}]}{h^2} + \frac{[u_{i,j+1}^{(n)} - 2u_{i,j}^{(n+1)} + u_{i,j-1}^{(n)}]}{k^2} = 0 \quad (4)$$

where  $h = \Delta x$  and  $k = \Delta y$  and  $n$  is an iteration index. If one ascribes  $n$  with time-like variation, then the above equation can be viewed as a time-dependent equation.

Foundations of Scientific Computing: Elliptic PDEs - p. 644

If the size of the matrix is large and if you are trying to do a direct inversion, we are going to take very large computing time, at every time we have to solve that equation. In actual physical problem, it may so happen that for a time dependent problem, we will have to solve a elliptic equation at every time step. So, you can imagine that any time saved for each time step accumulates and saves you lots and lots of computing time. So, there is an incentive for us to look for methods which will allow us to solve this equation in an efficient manner.

Now, that was the motivation behind developing various approximate means and let me start with the oldest and the classical method due to Jacobi. Later on, Richardson also used the same method; maybe he popularized. So, it is also called the Richardson method.

What we do here? We are solving that Laplace's equation with unequal delta x and delta y, given by h and k. This is a discrete equation. What you do here in the Jacobi method is - you try to evaluate the solution by pegging the diagonal entry at the current level and everything else at the previous level. So, this actually was practiced with the hope that if I take a very very large number of steps, when n goes to infinity, the solution would not know what the difference between n and n plus 1 is.

(Refer Slide Time: 56:53)

### Iterative Methods - An Analysis (cont.)

- Associating  $n$  with time, one can write:

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} + \Delta t \frac{\partial u}{\partial t} + O(\Delta t^2) \quad (5)$$

- Where  $\Delta t$  is a pseudo-time step. Substitution of (5) in (4) provides the following time-dependent equation,

$$\alpha \frac{\partial u}{\partial t} + \nabla^2 u = 0 \quad (6)$$

where  $\alpha = 2 \Delta t \left[ \frac{1}{h^2} + \frac{1}{k^2} \right]$

- We classify Eqn. (6) as a PDE. In the framework of (6), the solution  $u(x, y, t)$  is such that the auxiliary equations are obtained as,

$$du = u_t dt + u_x dx + u_y dy \quad (7)$$

$$du_x = u_{xx} dx + u_{xy} dy \quad (8)$$

$$du_y = u_{xy} dx + u_{yy} dy \quad (9)$$

- Equations (6) to (9) provides the following linear algebraic equation:

So, initially all got stopped there, but, then subsequently people went through the analysis and tried to justify why and how, this method would work. You have seen it

yourself in your exam paper that you actually are doing something like this, because when I am shifting this index superscript by one level - that is equivalent to doing this. I am introducing some kind of this pseudo time stepping.

If I substitute this in the prior equation, with non-uniform delta x, I mean they are uniform in x direction, they are uniform in the y direction, but, delta x is not equal to delta y. Then, we get this equation 6. This alpha multiplicative constant depends on the time step times the space steps that we take. We have done this. So, I do not think I need to explain it to you to that extent. Only thing is - you note that in 8 and 9, we do not write  $u_x t$  and  $u_y t$ . The reason is obvious; we do not want to increase the order of the system because  $u t$  itself is Laplacian.

(Refer Slide Time: 58:14)

### Iterative Methods - An Analysis (Cont.)

$$\begin{bmatrix} \alpha & 1 & 0 & 1 \\ dt & 0 & 0 & 0 \\ 0 & dx & dy & 0 \\ 0 & 0 & dx & dy \end{bmatrix} \begin{bmatrix} u_t \\ u_{xx} \\ u_{xy} \\ u_{yy} \end{bmatrix} = \begin{bmatrix} 0 \\ du - u_x dx - u_y dy \\ du_x \\ du_y \end{bmatrix}$$

- Note that  $u_{xt}$  and  $u_{yt}$  have not been considered. Why?
- The characteristics of the above are obtained by equating the determinant of the matrix to zero.
- This provides,  $dt [dx^2 + dy^2] = 0$  (10)
- Thus, the system is parabolic in time ( $t = \text{const.}$ ) and elliptic in space ( $dy/dx = \pm i$ ).
- As an assignment, work out the error propagation equation for Eq. (6).

Foundations of Scientific Computing: Elliptic PDEs - p. 845

If I take  $u_{xt}$ , that would be actually increasing the order of the equation. So, anyway, we figured it out; we got the solutions. These auxiliary equations collated with the differential equation and gave us the characteristic determinant in terms of equation 10.

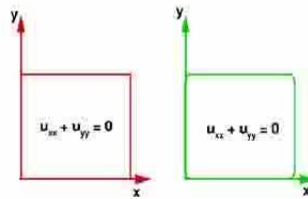
You can see - one of the characteristics is of course,  $t$  equal to constant; that tells you that you are basically marching in time. So, that is your parabolic direction and in the space direction, we have seen  $dy$  by  $dx$  as it appear on this.



(Refer Slide Time: 59:14)

## Elliptic PDE- Revisited

- The elliptic PDE's are of order ' $2n$ '.
- The number of boundary conditions are  $n$  in number.
- Do not be taken in by separation of variables in deciding the number of boundary conditions.



How many b.c.s for the problem on the left & on the right?

Foundations of Scientific Computing-Elliptic PDEs - p. 344

Please do understand that all elliptic equations have to be of even order because if you are going to get complex characteristic, they have to be conjugates. So, always elliptic equations will be of order  $2n$ . That is what I noted down on top that elliptic PDEs are of order  $2n$ .

Now, this is a very interesting observation. The next line, the number of boundary conditions - this is something I find very baffling. So, I just drew two diagrams; think about it; one is a perfect rectangle; another is a rectangle, but, with corners rounded off. You come back and tell me how many boundary conditions we need in the next class.

We will stop there.