Foundation of Scientific Computing

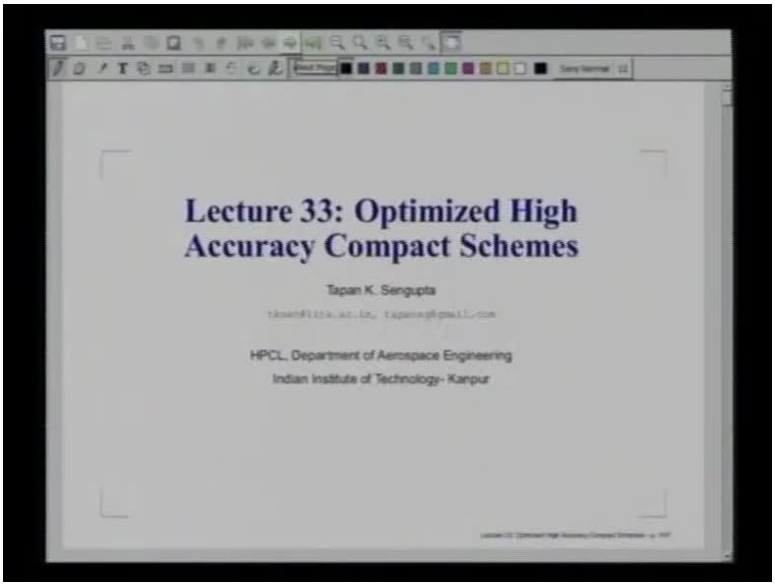
Prof. T. K. Sengupta

Department of Aerospace Engineering

Indian Institute of Technology, Kanpur

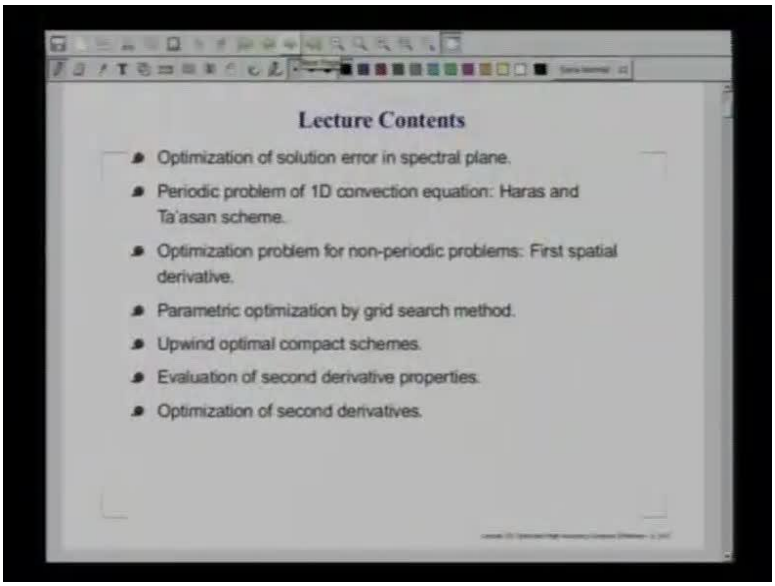
Lecture No. # 33

(Refer Slide Time: 00:19)



This is lecture number 33, and we are going to talk about high accuracy compact schemes, not in the traditional sense, but in a sense that how we optimize them in the theme of this course, that we optimize them in the spectral plane and see how it performs.

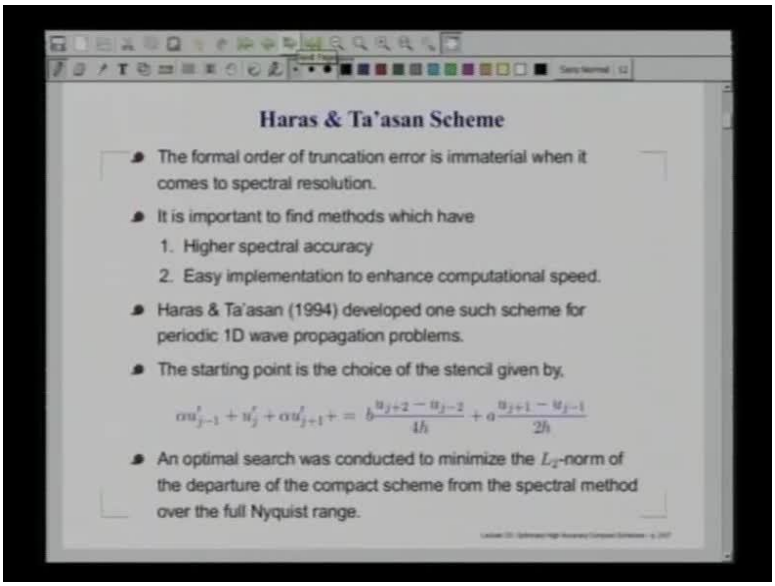
(Refer Slide Time: 00:40)



So, the contents would be basically, once again following the sequence that we will be trying to optimize means, minimize the solution error in the k space; that is your issue number 1. We will start off with a gentle problem where we are talking about a simple periodic problem of 1D convection equation as performed by Haras and Ta'asan; and once we have understood that part, we will be moving on to non-periodic problem involving only first special derivatives.

And we will show you a particular case where parametric optimization would be performed using a grid search method; and we will be subsequently talking about an A class of upwind optimal compact schemes which would be, which has use some of these in an optimization sense. Some of them have been evaluated manually, but all of them uniformly provide very high accuracy and which is probably the state of art at this point in time. Then, we will be moving over to evaluation of second order derivatives second derivatives and we will talk about how we optimize stencils for obtaining second derivatives.

(Refer Slide Time: 02:01)

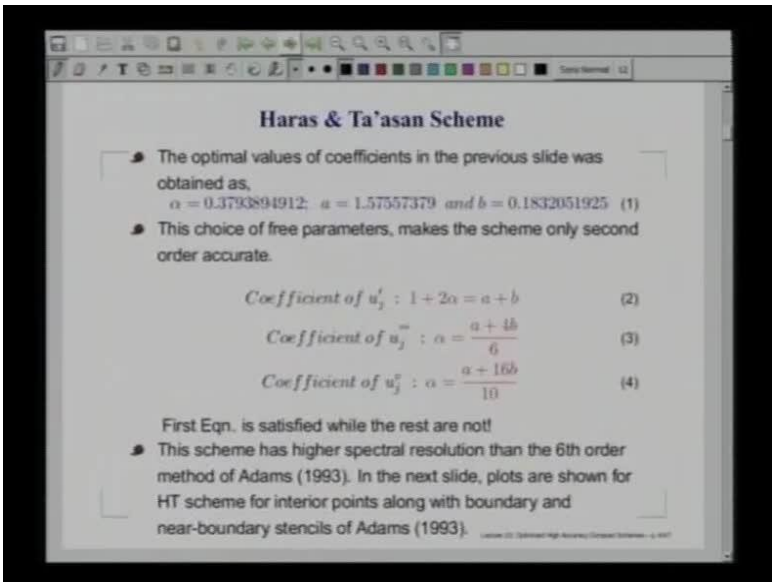


So, first thing is first, we look at what Haras and Ta'asan did for a periodic problem. We have emphasized so for time and again, that we do not worry too much about formal order of the truncation error, instead, what we worry about is the spectral resolution; and we specifically look for methods which provide very high spectral accuracy through this resolution. And, we want to easily implement this so that we also get methods which are very very computationally efficient; they are very fast that is what we want to do.

Haras and ta'asan looked at a 1D wave equation, and in doing so they followed the usual procedure that has been also followed by many other people, where the first derivatives are indicated by primed quantities; like here, in the left hand side, related to the function values, shown here pair-wise in terms of j plus minus 2 and j plus minus 1 point; and the compact scheme of course, requires implicit relationship between the 2, that is why you will see it also the derivatives are obtained simultaneously with j and j plus 1 point also.

So, once you have that, you look at obtain this derivatives and find out what error is being committed. Look at its norm in the mean square, in sense, that is what we mean by L 2- norm, and find out how does it depart from the exact derivative; and try to obtain it over as much of a full range, Nyquist range as possible; that we discussed earlier.

(Refer Slide Time: 04:00)



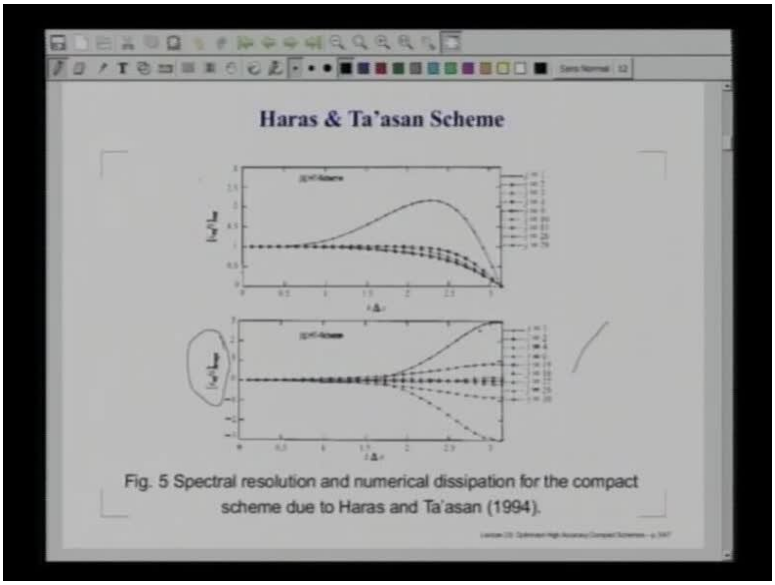
What Haras and Ta'asan found out? That you do not need to really pick out those values of alpha a and b. In the previous slide, as you could see that h is the grid spacing, so that is fixed; so, in this equation, there are three unknowns – alpha, a and b. So, basically the optimization process that one would be going through would try to locate the value of these three constants, so that you get maximum accuracy. What is interesting about the result? That is shown in equation-1. Is that this alpha, a and b? They are very irregular points they are not like the type of term that you get in Taylor's series expansion, matching order terms here. Actually, you will go through that same exercise that we have talked time and again; that you look at the error in the k space and you minimize it with respect to these parameters while satisfying certain conditions.

For example, looking at the previous slide, if I look at this equation and equate the coefficients of various order derivative terms, then the equation-2 corresponds to the coefficients of u j prime. Then equation-3 corresponds to coefficients of the third derivative, and equation-4 corresponds to the fifth derivative term. So, basically you have essentially three equations and three unknowns; so, if you solve it, you are going to get sixth order scheme; but, what Haras and Ta'asan did? They just simply satisfied equation-2 and gave up on 3 and 4, so, what happens? You have formally a second order accurate scheme, right?

However, even though it is second order accurate scheme, as we will see shortly that this choice of values of alpha, a and b provide extraordinary accuracy for periodic problems; and this was a really landmark result of its own time. And, what we can easily demonstrate that this second order scheme is better than a sixth order compact scheme; compact scheme by definition itself is far far superior over explicit schemes.

So, if I have a sixth order accurate compact scheme, that itself is saying a lot. Now, here, what we are talking about is a second order scheme which is far superior than a sixth order compact scheme. So, this scheme, I have just talked here about as a h t scheme; and what we could do is, we can go through this spectral analysis that we have done and we can do it over the full domain. We can look at various nodes together, and what we do is, we take the Haras and Ta'asan scheme, which was developed for a periodic problem to accommodate non-periodic problem; we have taken some additional near boundary closure, that is due to Adams, and we will see what it does.

(Refer Slide Time: 07:23)

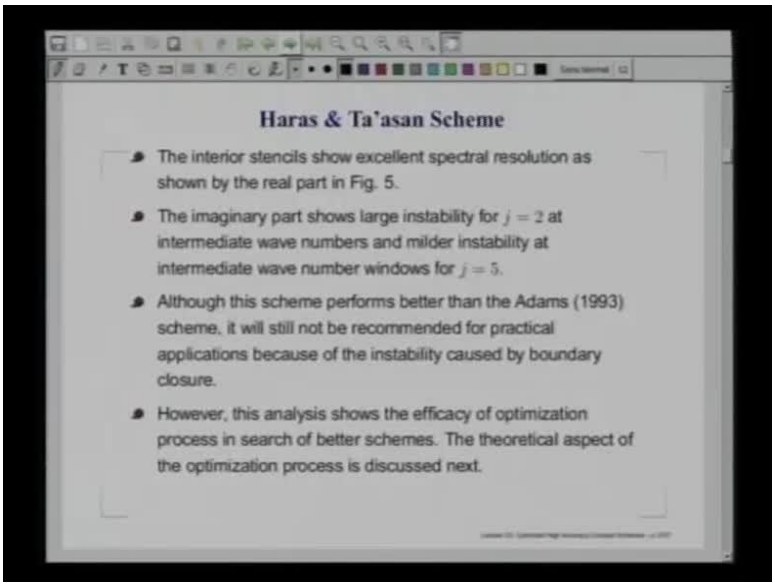


And once again, what we notice, that the performance parameter is k equivalent by k, and it has two parts. The top part is showing the real quantity, that is exactly what we have seen earlier also; we want this to be equal to 1; for as large a range of k delta, x is possible; and what we see for different j values, we have different resolution; this was obtained by using one of the method that we developed in 2003'

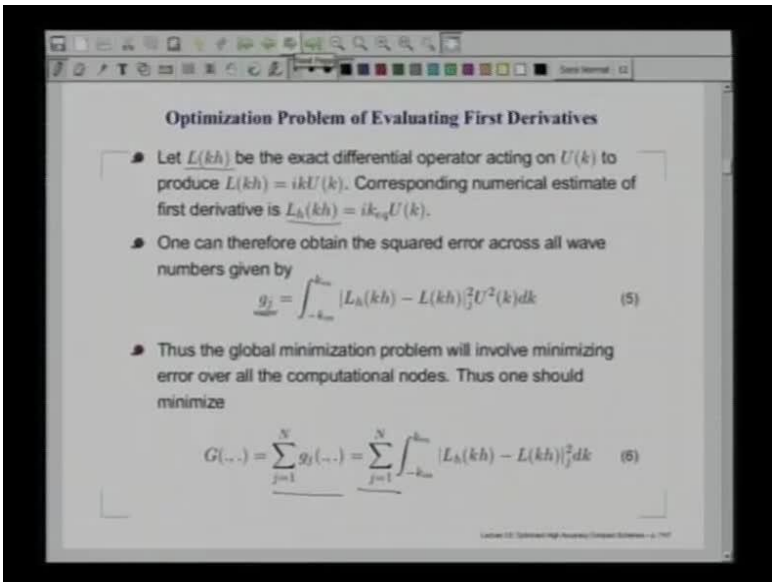
And what we notice, is that, at the first node, you have kind of an overshoot; but the first node is never very important, because first node is the boundary point; that is where you have the boundary conditions; so, do not, I mean, evaluate those derivatives there. Whereas, the rest of the points are pretty much clustered together and they provide quite satisfactory accuracy.

However, when you look at the imaginary part here, this is where we have a serious problem. You have noticed that, what we really want to do is, we want to introduce this imaginary part that should work like diffusion dissipation. We do not want it in any other way; so. that happens when this k equivalent by k, that we have shown here, should have a negative value; and we notice that half the points, half the points are on the negative side, and some of the points, half the points those are shown here are in the positive side. So, these positive values indicate opposite effect of diffusion, that is what we call as anti-diffusion. And we are noticing that this is not going to work, because many points are going to be simultaneously unstable.

(Refer Slide Time: 09:20)



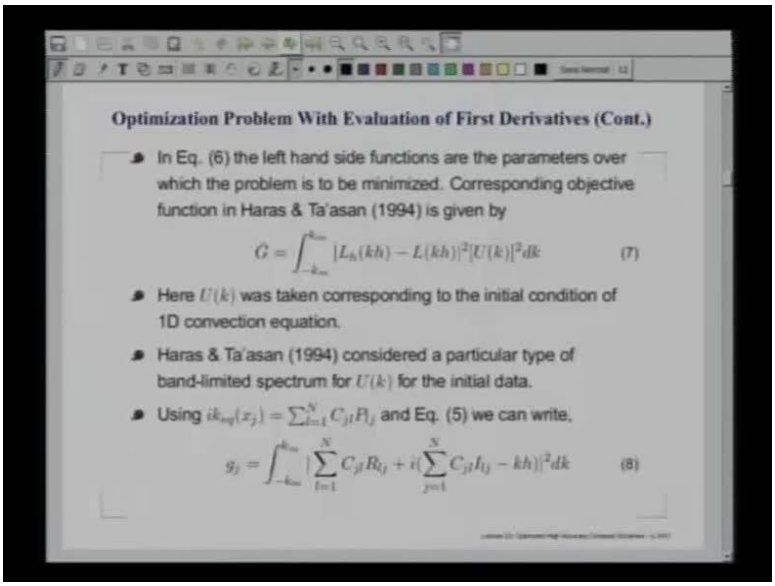
(Refer Slide Time: 09:50)



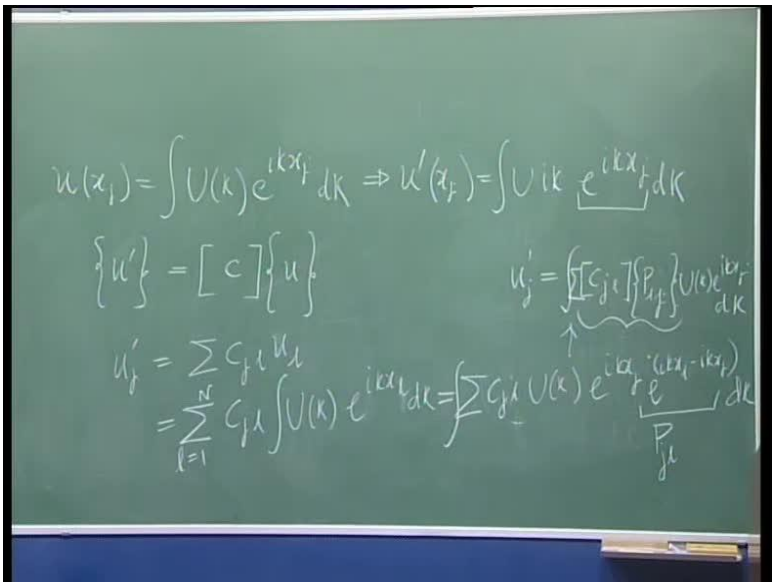
So, just because you have the optimal scheme for a periodic problem, you cannot routinely take it to a non-periodic problem; that is the lesson we learn here. So, we have talked about this, that although the original parent scheme was an optimized scheme, we do not get efficiency. So, what we need to do is, we need to really understand the optimization process itself; and once we have done that, we can really develop scheme of our own for non-periodic problem; because, now we have the ability to analyze the full scheme including those boundary conditions as they are used for the compact scheme.

And, if we talk about l of k h as the exact differential operator, and if we look at the corresponding discrete operator as l of subscript h, then we can define an objective function which I call as g of j, which will be nothing but the departure of these two operators squared; it is operating on the function u of k, and integrate over all possible range of k. And, we would like ideally to take it from minus k m to plus k m, and that is something we do not like to get. All these, we like to get, that, but we will not get it; so, we may actually take a fraction of k m; we will talk about it later. And, if this is for your j th node, we can actually sum up such error for the whole domain and that is what is done here; we have added up for all nodes for j equal to 1 to n.

(Refer Slide Time: 11:09)



(Refer Slide Time: 11:23)



So, if we have this quantity given by equation-6, then we can optimize it, and that is what one can do now. What really happens is that, we have seen that u of k is nothing but the Fourier Laplace amplitude of, let us say, function u of x evaluated at a j th node. So, if I write it like this, this is what I get. Any derivative evaluation that we get, we can eventually write it like this; as the derivatives indicated by prime would be equal to nothing but some constant matrix c times the function, right?

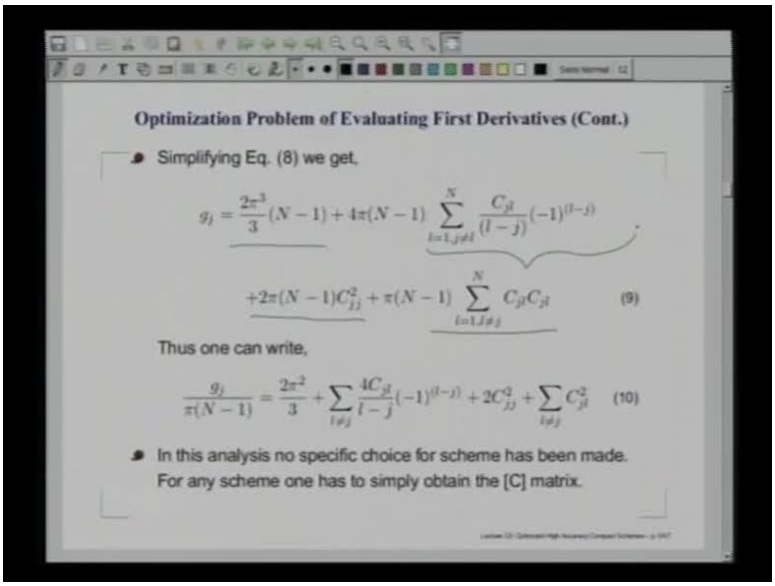
For explicit scheme, what happens is, we have a special nature of c, and for the compact scheme we have a different nature of c. Now, what we are seeing? Suppose I am looking at the j th point, so what we are getting here? u j prime is equal to the j th line multiplying the whole column, so, that is what we are going to get; c j l into u evaluated at the l th node. But usually, what you really want to do when it comes to a derivative evaluation? From here I get u prime of x j, will be nothing but equal to u of k times i k and e to the power i k x j d k, so, what did you notice? The derivative is determined by the phase at the j th node only, whereas numerically, what you are getting here, it is summed over all possible nodes and that is not something you would like to do.

So, what I could do is, since c is a constant matrix, I can use this representation and I can write it like this c j l; and this, I will write it as u of k and e to the power i k x l, right? Because, this is u of l, right? And i will write d k, and this is summed over l equal to 1 to n; but, if I want to represent it in terms of the j th node only, then what I should do is, I should just simply write c j l, and I will keep it as it is. I could take this integral also, write and I could write u of k and here I will write here i k x j.

But, what was it there? It was i k x l, so, what I do is i k x l minus i k x j and d k, I have written just simply, rearrangement; so, this quantity if I call this as some kind of a projection operator that is projecting the l th node to the j th node, then what you are seeing here, is basically, you are getting u prime as something like… here; that you will get c j l, so, if I look at it like this, this will be multiplying by p l j times u of k e to the power i k x j d k, and this is this, right?

So, that is what you are noticing, that this plays the role of what. This, of course, there is a sum over l, right? So, this quantity is nothing, but our i k equivalent. Remember, that is what we are doing.

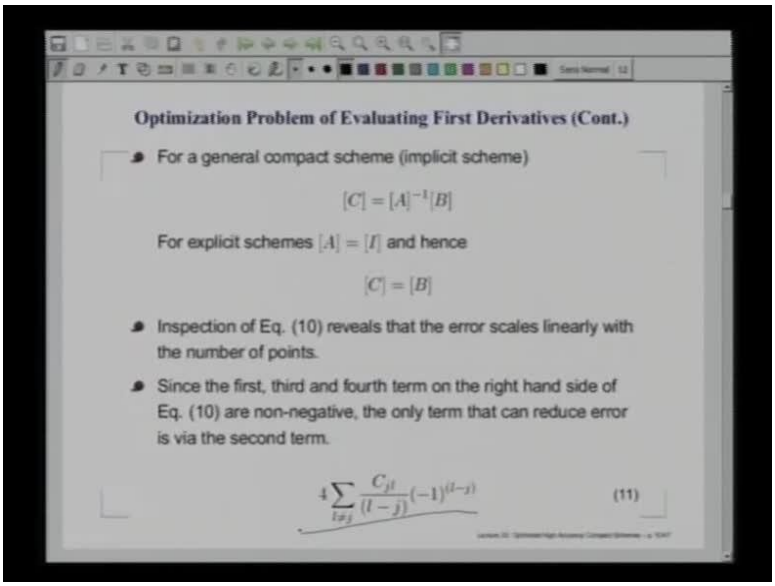
(Refer Slide Time: 16:23)



So, that is what we have written down here, that this i k equivalent x j is nothing but c j l times p l j; and that we can substitute in equation-7. And then, not 7, the equation-5, the previous slide which we have found it for the single node; so, if we go ahead and look at it, this is what we are going to get. So, this is the expression that we are going to get; so, if I decide on a particular algorithm to use c, then I can work out on the c matrix, and this thing is very easily obtained, so that, p l j i could write in terms of its real part and the imaginary part; that is what we have done. And, we can carry through this process and some simplification later; you get equation-9, so, this is the kind of error term that you are going to get. The first part comes from the exact quantity itself; this, you notice that this part comes from the diagonal entry of the c matrix, and this is a squared term, so, what you find is, basically this will be always additive, they will not reduce error; they will never reduce error because it is a c squared term.

So, what happens? You always like to generate schemes which do not have the diagonal term; and we have seen that central difference schemes are that ideal example, where you actually never have those diagonal terms.

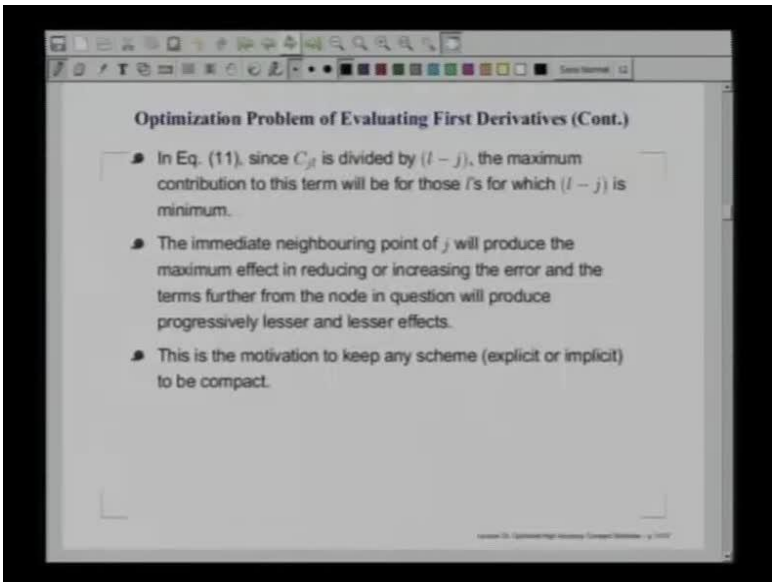
(Refer Slide Time: 17:37)



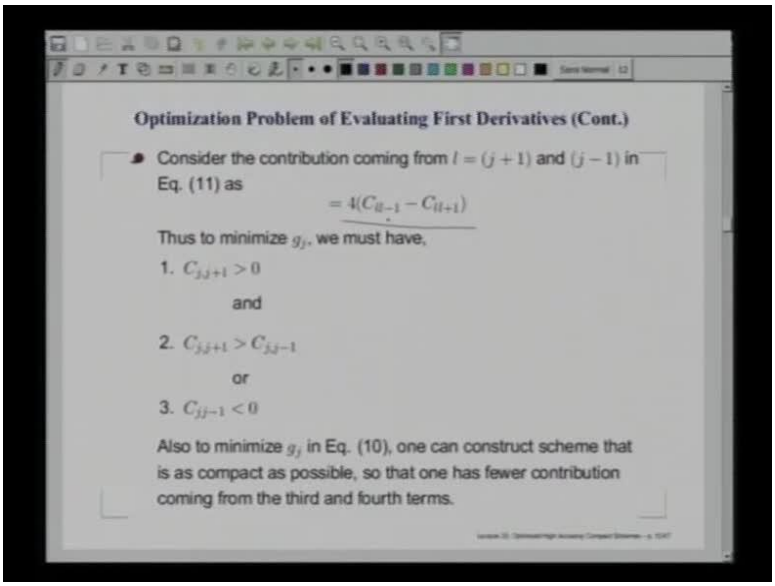
So, we are getting some ideas; whereas this term is also a product term, squared term; this also will not reduce error, this contributed by the off diagonal term; it is only this term that can be manipulated to reduce error, and this is what has been achieved at by us. And what we see is, that such an error term is identified here in the end, and we can choose our c matrix in such a way, we get that.

What you also notice, that this c j l is divided by l minus j, so, you are looking at the j th node, and l is the variable node, so, what you find is, that c j l is scaled by the distance of the j th node from its neighbor; so, the nearest point l is equal to j plus minus 1, would contribute more; because, if you look at plus minus 2 point, that will divide that c j l by that much of amount, so, it is a nearest point that actually plays a greater role for… So, that is also gives you an idea, why you need compactness? You do not want a wide spread stencil that does not do much in terms of reducing error.

(Refer Slide Time: 18:40)

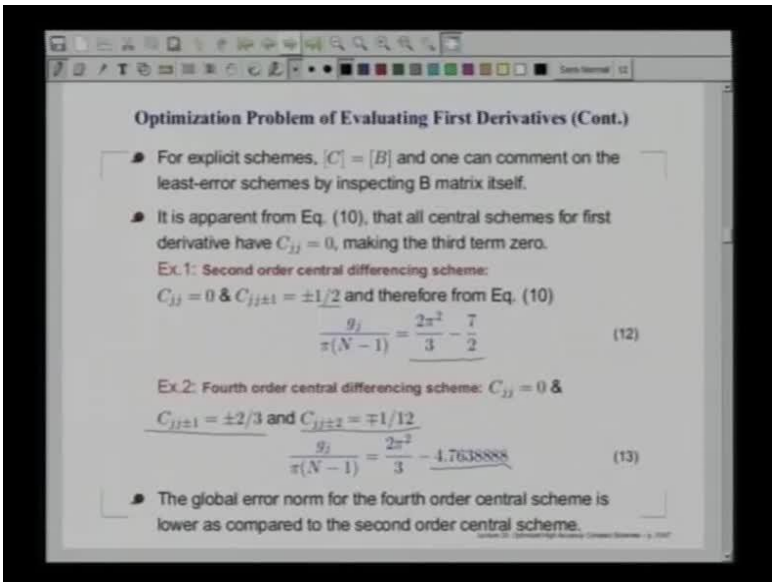


(Refer Slide Time: 18:48)



So, this is something that we can talk about, so we have talked about, now, why we want to have a compact scheme. And then, we can go ahead and estimate this, for example, if I look at the point l is equal to j plus 1 and j minus 1, then the error term comes out like this; so if I want to minimize g j then, what I should have is… Well, you could write it here, j j minus 1 and j j plus 1, I could write this as a positive quantity; because, if this is negative j j plus 1, you can see that, instead of reducing error, it will actually increase error; so, that is what your condition-1 is.

(Refer Slide Time: 19:46)

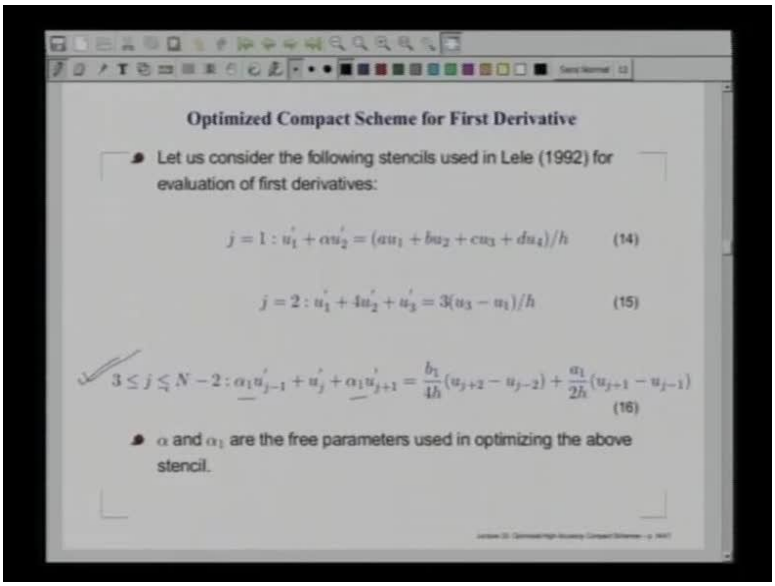


And the second condition is also similarly, the thing that you… If that is true, then you also want its relative magnitude should be more than the magnitude of the other term. And of course, if you have this term as negative, that also adds to reduction of error; so, these are various possibilities by which you can choose the c matrix. Here are some examples of well-known schemes.

For example, the c d 2 scheme that we have talked about, that has a contribution coming from j plus 1 and j minus 1, so, basically, the coefficients are plus minus half; and we can substitute in that expression for g j, and we get this expression. So, you can see the first part is due to the i k exact term and this minus 7 by 2 comes from the choice of your method.

Now, if you move over from c d 2 to fourth order accurate scheme c d 4 scheme, then we also know, these are the coefficients c j j plus minus 1 is plus minus two-thirds and c j j plus minus 2 goes like this; and you substitute there, you get this; and you can see comparison between 12 and 13 is that negative part, is contributed more for the c d 4 scheme. So, of course, c d 4 scheme is more accurate, so, we know it. So, here is a kind of validation of well-known result, that from optimization point of view, also it holds out.

(Refer Slide Time: 21:04)

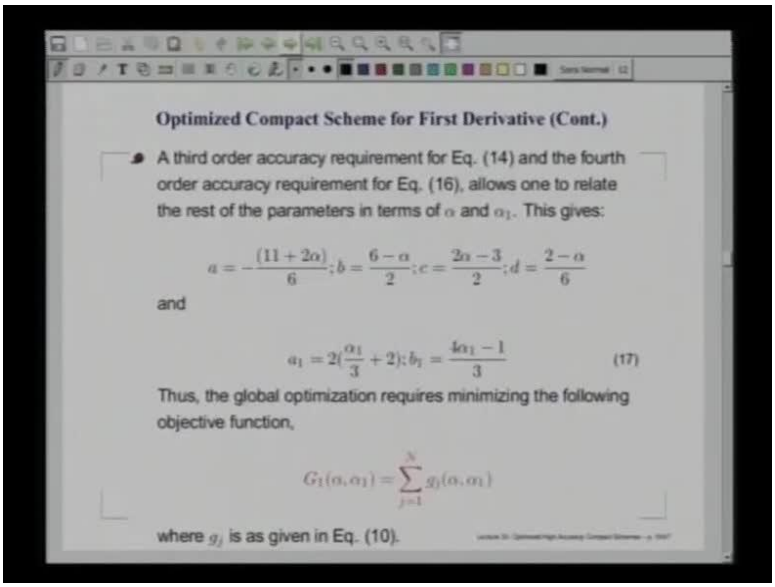


So, what we could do is, if I am trying to, now, optimize a compact scheme, I would basically be taking up something like this. Look at this equation; this is what we have just now shown, that the derivatives are written in terms of these coefficients alpha 1. Those are implicit 3 point stencils for the derivative, whereas the points are 5 points, right? j plus minus 2 to j minus 2, so, with the coefficients b 1 and a 1. But you see, because of this j plus 2 and j minus 2, we cannot use this expression at j equal to 1 and j equal to 2, right? This is what I have been telling you about the closure problem, that this general expression will not work for all the points. You need some additional, I mean, additional schemes for the near boundary points, like what I have shown here; for j equal to 1 and 2, we will have to… also and n minus 1; then, we have a complete scheme.

And, once I have the complete scheme, means what? I have the c matrix. And once I have the c matrix, I can go through this exercise. So, what we have done is, for j equal to 1, we have written a one sided scheme, because you see, the points are available only in one side. So, if I am looking at j equal to 1, I can only take information from 2, 3 and 3; so, that is your right hand side is. Whereas on the left hand side, I have tried to kept the implicitness of the scheme, so, I have introduced u 1 prime and involved u 2 prime, right? And j equal to 2, I have just taken very deterministic scheme that looks like a symmetric central scheme, because this is the central point, and off diagonal terms are same magnitude plus 1, plus 1; and this is almost like your u 3 minus u 1, like your central difference type of thing.

So, now we have the full scheme here. What are the parameters that we have? Well, we have many parameters; we have alpha 1, alpha on the left hand side; on the right hand side, what we have here? a, b, c, d and a 1 and b 1

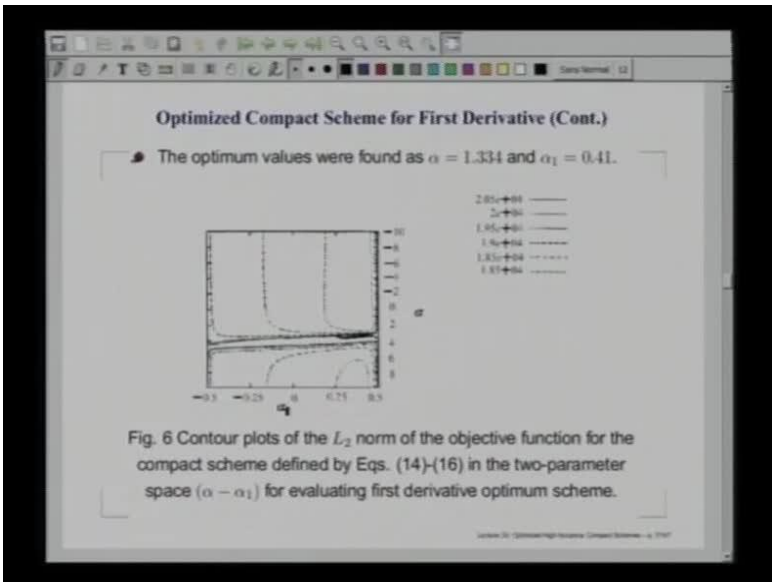
(Refer Slide Time: 24:09)



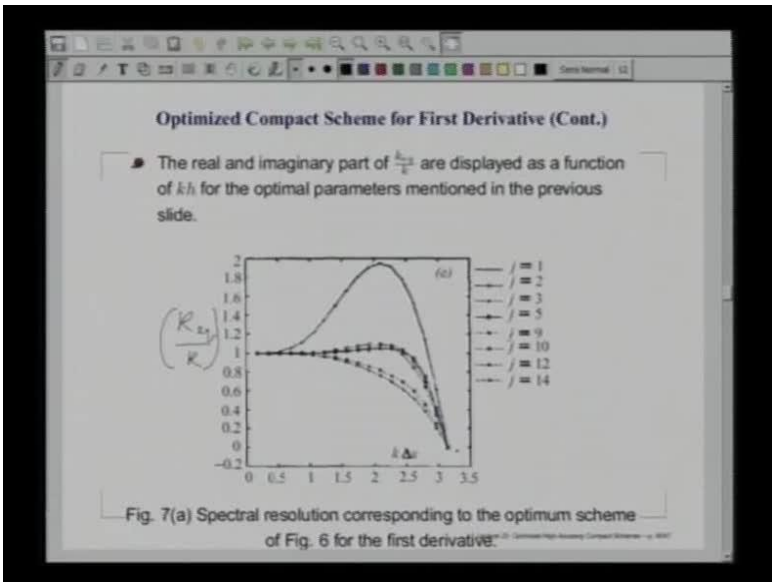
What we could do is, we can expand 14 and demand that we have some kind of a third order scheme, because we want some kind of an upwinding. We do not want instability; so, if we do that, we can get those coefficients in terms of alpha; and that is what is given here. The same way, we can also look at the general stencil and equate the Taylor series, and what we are going to get is this following equation, that is given by relations, given by 17.

So, what happens is essentially, out of all that parameter, we have reduced the optimization issue in terms of only two parameters, alpha and alpha 1. And then, we add it over all possible nodes, j equal to 1 to n; that is our global problem, so, we want to really look for a combination of alpha and alpha 1 for which this g 1 is the minimum; and that is what one can do.

(Refer Slide Time: 24:44)



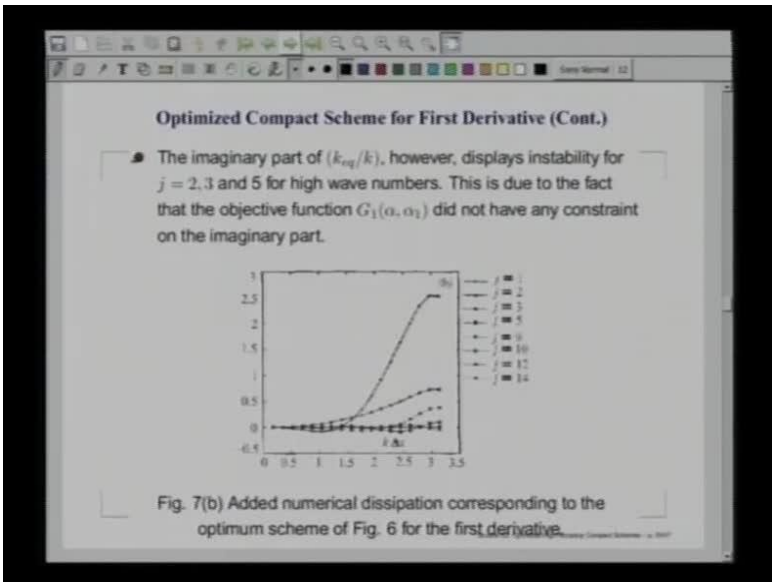
(Refer Slide Time: 25:38)



And the results are as shown here in alpha, alpha 1 plane; and the various contours are plotted. And these contours are like the top value is 2.05 into 10 to the power plus 4, and that keeps reducing to some value, and then again it increases. So, the minimum is somewhere in this neighborhood, somewhere in this neighborhood. Because, if you go, if you increase alpha, I mean, reduce alpha, error increases; you increase alpha, error increases; and we can really find out by from this contour plot, where exactly this g contour is, that we have plotted, attains its minimum value. And once you do that, you get some values of alpha and alpha 1.

Now, you have a scheme which is optimized, and if you do that use those values of alpha and alpha 1, and you work out the full domain analysis; and k equivalent by k is plotted here, so this side is your k equivalent by k and this is your real value. And ideally, you want it to be equal to 1, and that is plotted versus k delta x.

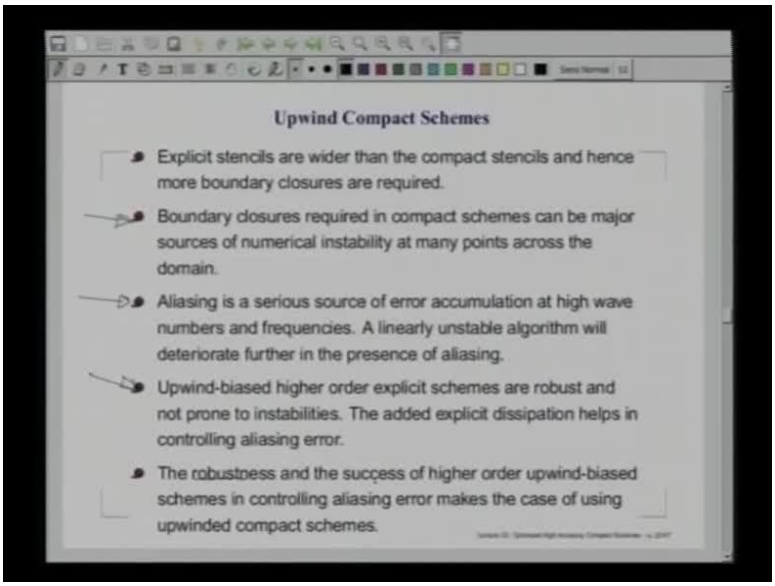
(Refer Slide Time: 26:13)



So, what we can do is, also we can plot the imaginary part of k equivalent by k; and here, you see things are not as good as ought to be, right? What we find here, that for certain nodes, for example, j equal to 1, it remains stable up to some value of k h, but then, it becomes violently unstable; but we have noticed that j equal to 1 is never our main point of concern. However, if you look at j equal to 2, that actually starts off and remains unstable all across. If you look at j equal to 3, it remains stable; then again, it becomes unstable, so, it has windows of stability and instability.

So, what basically is, we are looking at here, that in the process of minimizing error, we have not been able to keep the scheme in such a way that, we get actually a spatial discretization that will lead to stability. Because, you see, what we are looking at here, is only the spatial derivative part; we have not talked about any equation; we have not talked about time discretization; it is just simply the role of… Spatial discretization, itself, can lead to stabilization or destabilization. And here, we are looking at a scenario where the spatial discretization can actually lead to instability.

(Refer Slide Time: 28:14)



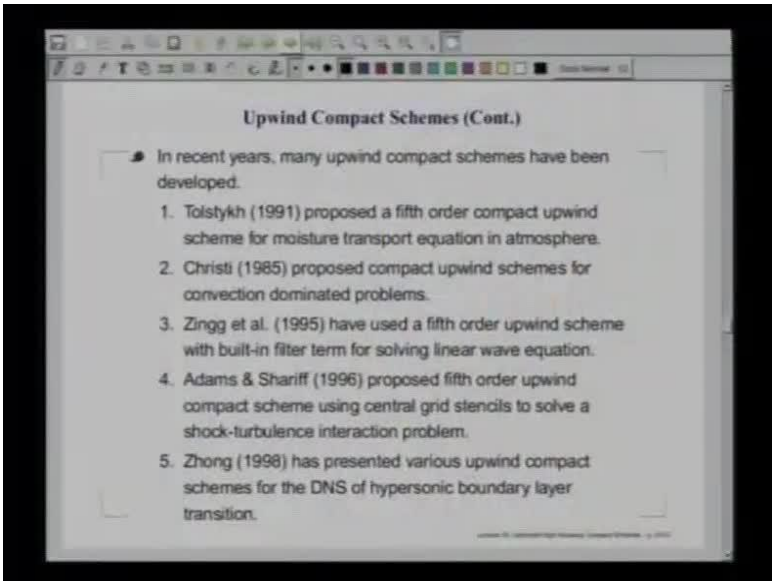
So, this is not something we want, and this happened, because our original scheme was a central scheme. If you look back, if you look back, what we had done here is, we had a central scheme here, the left hand side it was central; so, it shows that central scheme will not function the same way that we have seen for explicit scheme, we needed upwinding. And, that is the reason that we need to resort to even taking upwind compact schemes.

We also need to worry about boundary closure, because the boundary closure was the major source of error. Because, we saw the major problem was coming from j equal to 1 and 2, so, that is a major issue that we must really pay attention in boundary closure. Then, of course, error does accumulate at high wave number due to aliasing problem and we need to really be cautious about aliasing. What we need to do is, then, we need to add upwinding; and when we do the upwinding, we notice that they are more, which should be more effective at high wave number; that will also take care of the aliasing problem that we talked about, that its essentially high wave number operation.

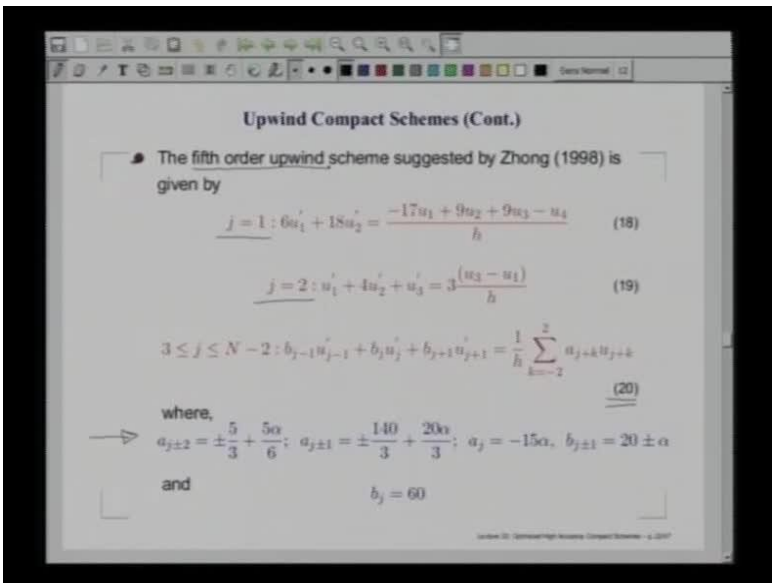
So upwind schemes are desirable for many points; however, what we need is that, they should be robust, and that will lead to really give us two possible positive attributes. Number-1 is, they will add to numerical stability, and they will also reduce your numerical issue of aliasing. Aliasing is a kind of a non-linear instability, whereas, numerical stability analysis that we talked about so far, it is all relates to the linear

mechanism; so, aliasing leads to non-linear instability. So, upwinding is needed to control both these linear instability as well as non-linear instability.

(Refer Slide Time: 30:02)



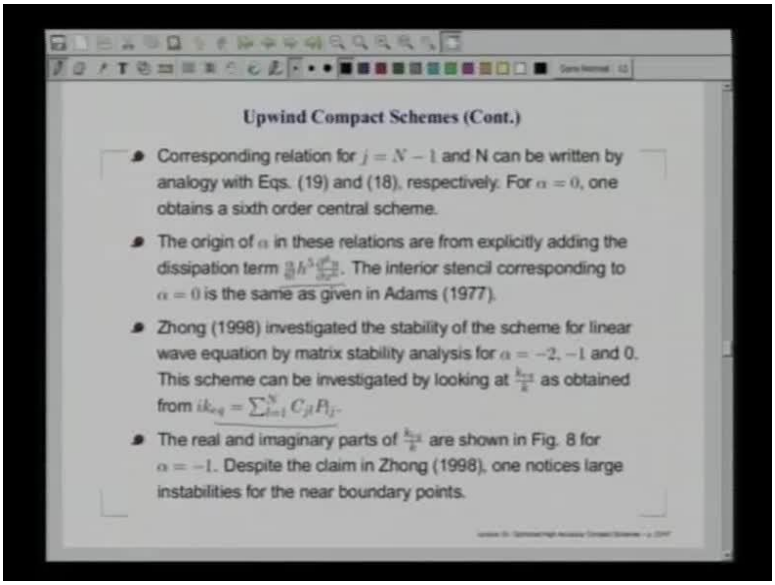
(Refer Slide Time: 30:10)



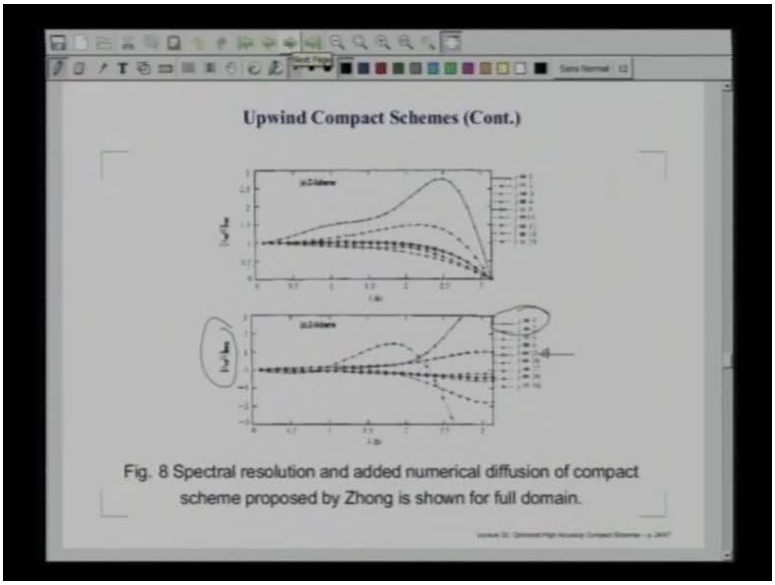
There were many people who have really contributed to upwind compact schemes; some of them are noted here. And, we will just simply look at one of such schemes which was proposed by Zhong, and he used this to simulate hypersonic flow problem. And, as you can see, the main stencil is given by equation-20; and as you notice, that here it is also a 5-point stencil, on the right hand side. So, we need to have boundary closure at j equal to

1 and j equal to 2, as well as at j equal to n and n minus 1. I have only shown you here j equal to 1 and 2, and these were the stencils used by him; and various coefficients that were given by him, actually, are written here in the last line; and choice of these parameters were governed by seeking a fifth order upwind scheme, so, fixing the fifth order, helps us in choosing these coefficients given in the last line.

(Refer Slide Time: 31:08)



(Refer Slide Time: 31:32)



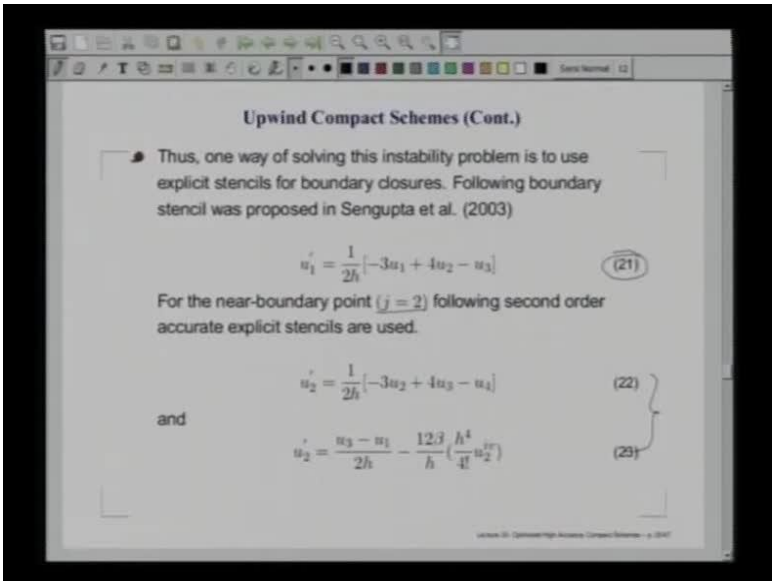
So, what we have done here, is basically added a sixth derivative term; this is like our fifth order upwind scheme, right? We would add a sixth derivative term, so, that is

precisely what has been done. And, since we know what i k equivalent is, we can plot k equivalent by k of this particular scheme, and we are going to get a picture like this. Despite the publication and the claim that they have solved hypersonic problem, you can notice from the imaginary part of the plot, that there are many many points which are violently unstable.

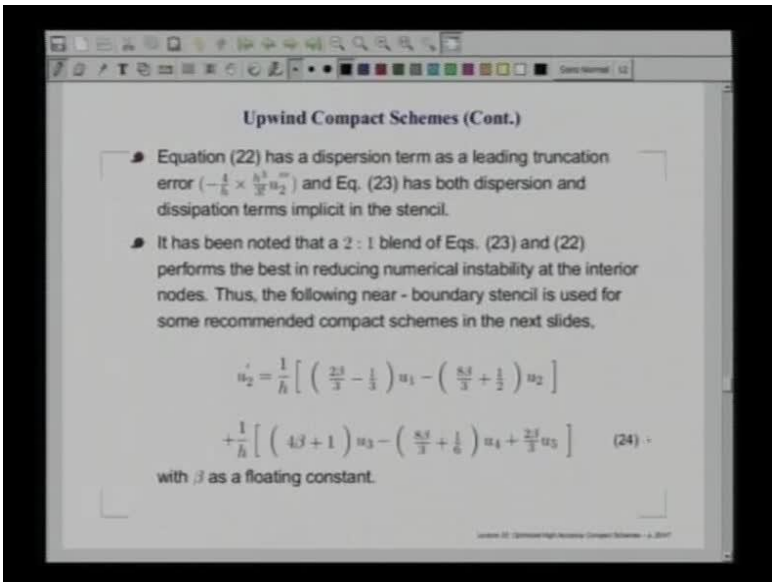
Now, why does it work? It works because, if you are looking at a convection problem, this unstable nature of the problem is at the inflow of the domain; that is why, it is unstable. But then, what has happening? Those disturbances are propagating inboard, inside; once they go inside, they are no more under the effect of instability; they are probably getting into the region where they are stable.

So, what happens? Here, this is a very typical case numerical instability, excites the flow at the inflow, and those disturbances convict downstream and becomes quiet; and then, we claim we have done it, direct simulation of the flow. It is not quite right, but we will not go into that for the time being; but, what we are noticing here, is that this scheme has a spurious excitation at the inflow; that may not be physical problem.

(Refer Slide Time: 32:50)

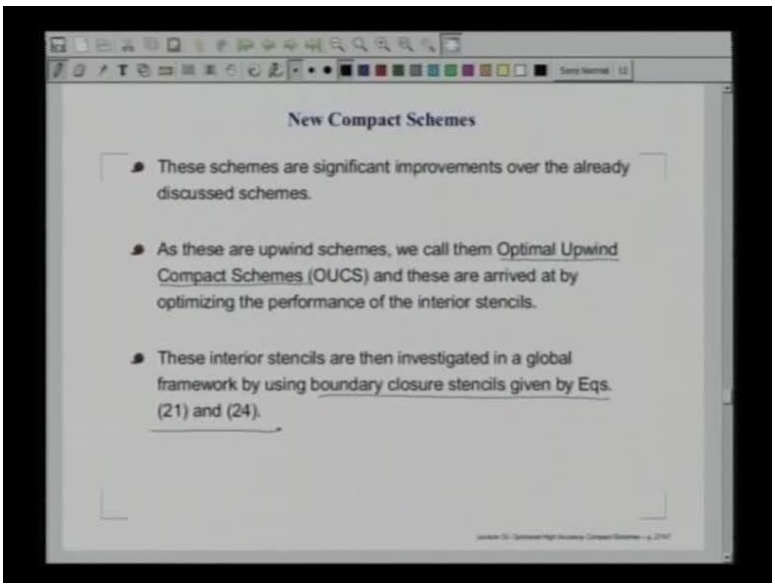


(Refer Slide Time: 33:26)



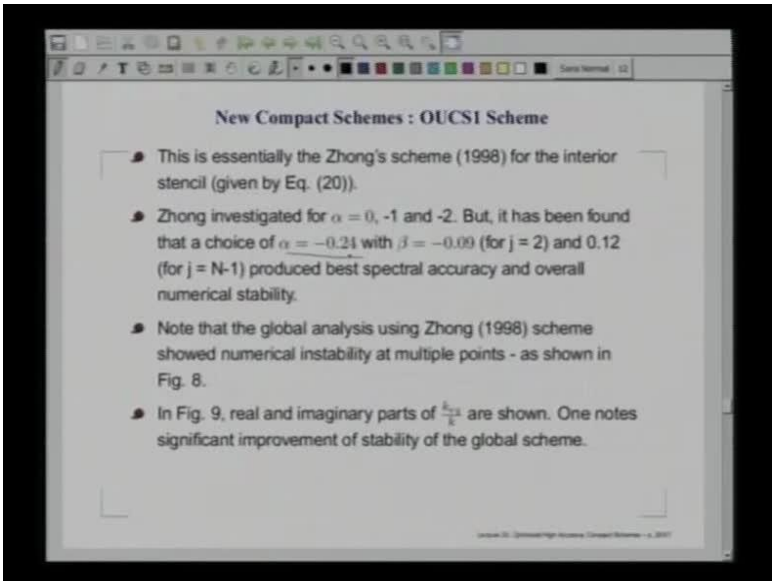
So, having identified that most of the source of the problem appears near the boundary, and those are due to the boundary closure, we decided to propose some explicit scheme at the boundary; and this is what you are seeing here. Equation-21 is proposed for the first point; and for the second point, j equal 2, what we have done? We have taken two stencils, given by 22 and 23, and from these two stencils, we have constructed this stencil. We have constructed this stencil; and this stencil actually ensures that globally, we will have a stable system and that should solve our problem.

(Refer Slide Time: 34:09)

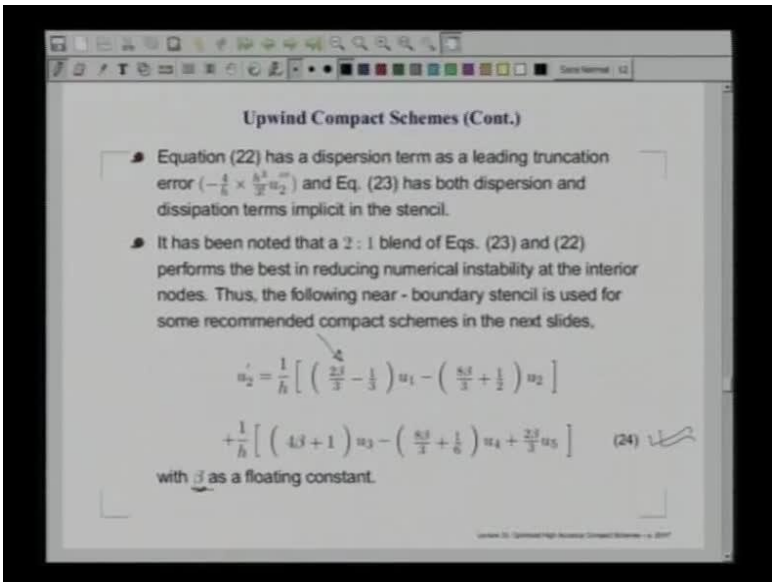


So, this was essentially the improvement of the traditional, various order schemes which we do not need to do optimization. We look at their problem, source of problem, and then we rectify those problems by changing the boundary closure; and thereby, we actually ended up getting a set of compact schemes which we have called as optimal upwind compact schemes, or OUCS schemes. Here, in our HPCL, we have actually developed four such schemes, OUCS 1 to OUCS 4; and you recall that, in one of the earlier lectures, I have shown you the properties of OUCS 4 scheme. It was accurate all the way up to k h equal to 2.7 or so; and that was a significant achievement, probably the most accurate scheme that is available so far in the published literature.

(Refer Slide Time: 35:03)

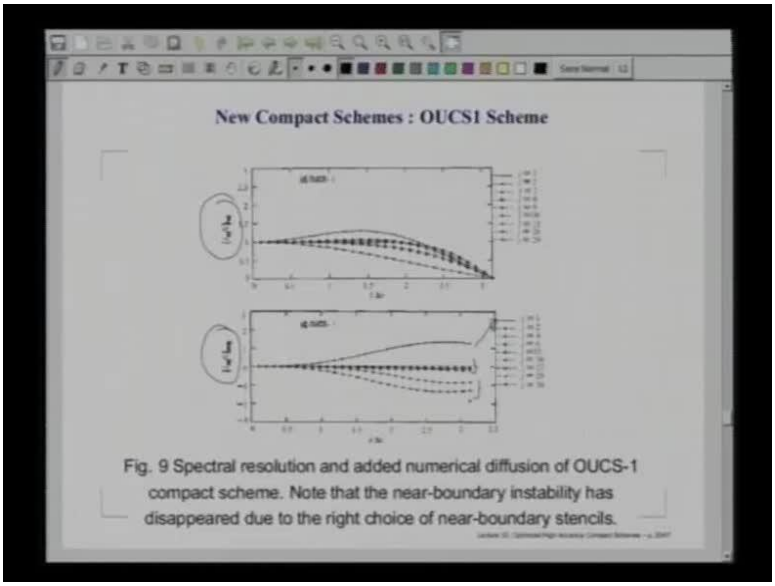


(Refer Slide Time: 35:51)



Now, what we can do is, we can look at those interior stencils in conjunction with this boundary stencils, given by equation-21 to 24 and come out; these new schemes to do; not what we did. We looked at first, the Zhong scheme itself, which we have been critical about, and we fixed its problem. What we did was, we fixed the value of alpha, and if you recall, that for the point 2, we have a floating parameter here, beta. So, we can choose the value of beta to tune; and that is what we did in coming or fixing this Zhong scheme. We found out, that we need to take value of beta equal to minus 0.009 for j equal to 2, and plus 0.12 for j is equal to n minus 1.

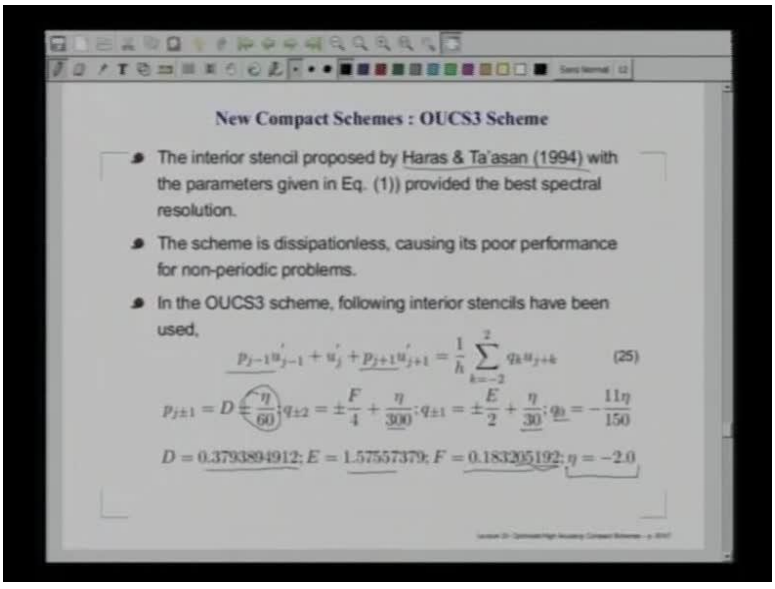
(Refer Slide Time: 35:58)



And, you notice that for our first point, we had explicit scheme; so, we do not have to have any fix there. Now, what we can do is, we can plot the property of this scheme; and you see what has happened? This is what we get for the real part, and we get a fairly decent property as good as, or even better than Zhong's scheme, better than Zhong's scheme. But look at the imaginary part; the imaginary part has spectacularly improved. You see, barring j equal to 1 point, which is unstable, rest of the points are all stable. And what is you notice is, most of the points are clustered near 0 value, so, we are adding very minute trace amount of numerical dissipation.

These large values that you are getting, is near the outflow boundary; and at the outflow boundary, what you want most of the time? In computing, you are getting the propagation of disturbances, and if you do not set the outflow boundary condition properly, they actually reflect from the outflow boundary and distort the solution. So, we want to avoid that. So, we will see that this property of excessive attenuation near the outflow boundary, actually helps in dissipating those disturbances, so that, the reflection becomes weaker; and sometimes, it actually removes those reflections altogether. So, this attribute of this scheme is actually positive thing.

(Refer Slide Time: 37:34)

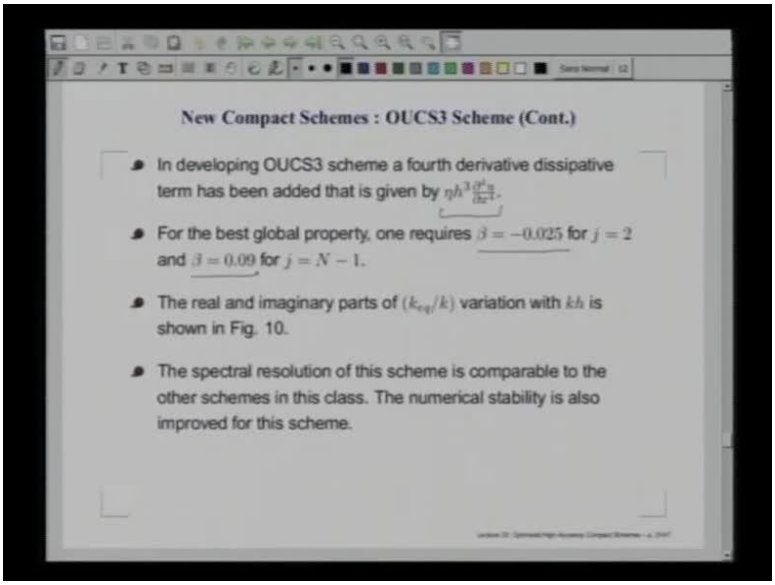


So, in the scheme of new compact schemes, we just showed you another such scheme OUCS 3 scheme. And this is again borrowing that Haras and Ta'asan scheme that we talked about, which was spectacular for a periodic problem. Now, we converted it for a

non-periodic problem; and what we did was, we implicitly added the dissipation term and that is shown here in this coefficient, p j minus 1 and p j plus 1; that you notice, that there are these terms, plus minus eta by 60, though. So, eta is some kind of an upwind parameter in Haras and Ta'asan scheme; this eta was 0.

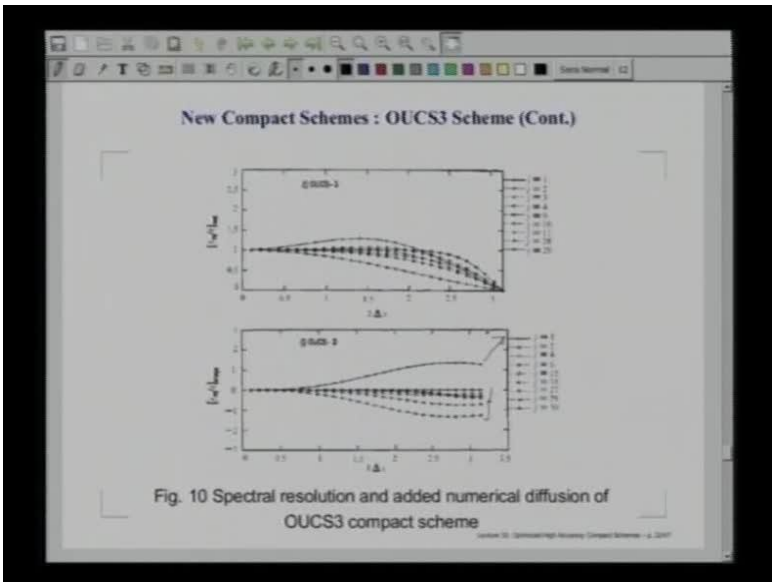
So, we have purposely added that eta term to control; so,,eta is additional degree of freedom for us to choose the scheme; and on the right hand side, the coefficients are all given. They also involve those upwind coefficients that you can see; and you also notice, since we are doing a upwinding, so, we also need to have the point itself, and that coefficient is non-zero for an upwind scheme. And if we take eta equal to 0, q naught will be 0; and then, we will be end up with a central scheme, right? So, this is what we happen to see; and then, we take the coefficients d, e and f, which we called before as alpha, a and b; are essentially nothing, but those d, a, e and f. So, they remain the same; and we have shown here, a scheme for eta equal to 2 minus 2.

(Refer Slide Time: 39:18)



So, if I do that, I basically notice, that in choosing this eta, what we have done is, we have added a fourth derivative term; and the essential idea is that, we do not want to add second derivative, because most of the physical problem has physic[al]- second derivative as a part of the physics itself. So, if we want to add some dissipation, it should be higher order, so, it should not interfere with the physical nature of the problem; that is why we added this fourth derivative term.

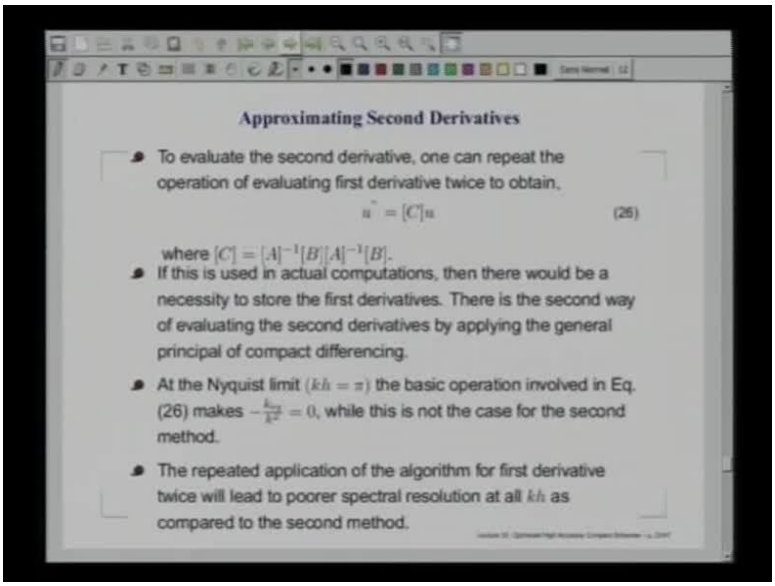
(Refer Slide Time: 40:11)



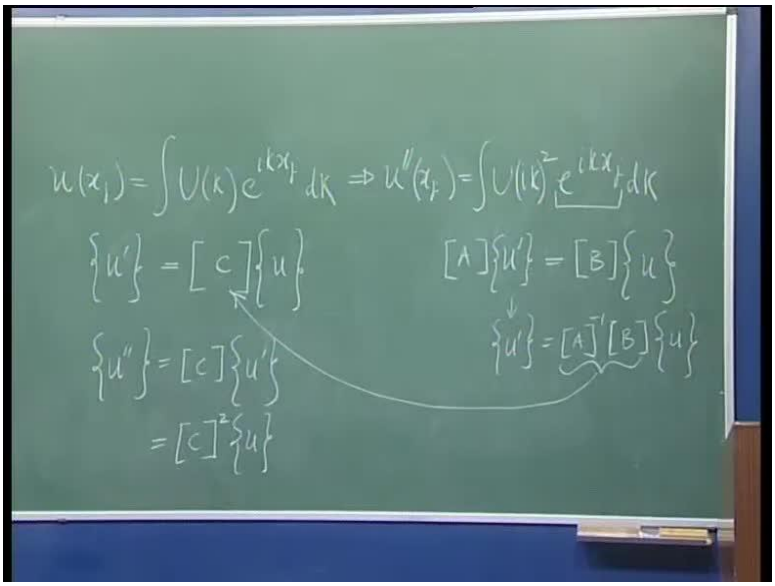
And for best global properties, remember, that closure scheme for j equal 2 and j equal to n minus 1, we had. So, here, we figured out, if we take beta equal to minus 0.025 for j equal to 2; and beta equal to plus plus 09.09 for j equal to n minus 1, we get very good scheme, and the real and imaginary part is shown here.

And once again you can see, that the most of the time, you see, look at this value of the real part, it remains flat, all the way up to 1, up to about 2.2, 2.3; so, we have a scheme which actually gives you a very very flat performance like what you would have gotten using a Fourier spectral method. However, the advantage of compact scheme, as we have noted by now, that we can circumvent all the drawbacks of spectral method; so, we get a near spectral accuracy work on a non-uniform grid; and we get very high quality result, and they are all stable. The point j equal to 1 is of no concern to us; it is a Dirichlet point. So, we should not worry about it.

(Refer Slide Time: 41:03)



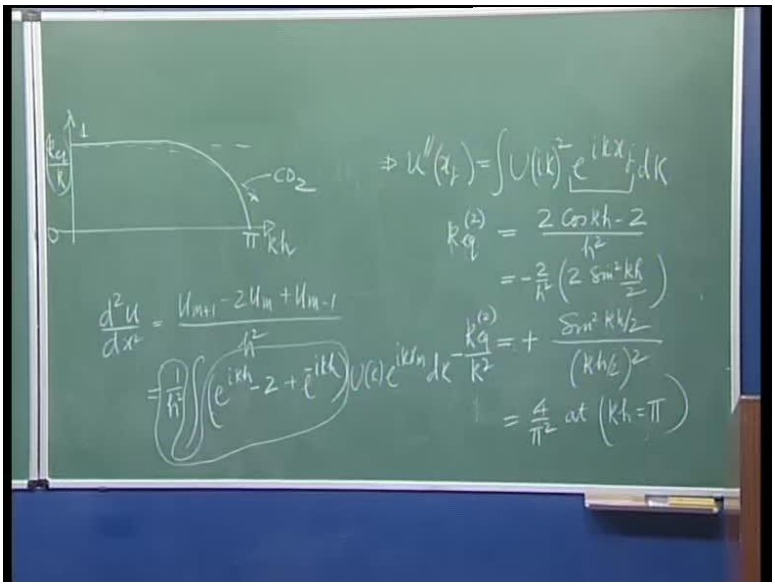
(Refer Slide Time: 41:15)



The next thing is about the second derivative. As we explained about the first derivative here, so, what we could do is, we could also relate the second derivative with respect to the function values. And now, what you can expect here is that, if you look at the exact quantity, the second derivative, what you will get? You will get here i k square, right? So, that is what you would like to get. And, what happens is, we have now talked about evaluating the first derivative once. So, if I have a scheme for evaluating the first derivative given by a linear algebraic equation of this kind, then you see, what you can get, I could write this as a u prime is equal to A inverse B operating on…

So, this is what we called it as c matrix; you see the connection. So, if if that was for a first derivative, so, what I could do is, if I have a first derivative like this, I could similarly write a second derivative relating it with the first derivative the same way; and then, this itself is c times u, so I could write it as c square.

(Refer Slide Time: 43:22)



So, that is what we have said here, that in equation-22; that I will choose C as A inverse B multiplied twice; it is like c square, right? So, that is what we could do; one thing we notice, that in most of the time, when we plotted the properties for the first derivative, what we noticed is that, k equivalent by k, which determines the performance parameter plotted against k h. If I plot k equivalent by k real part, then ideally I should get equal to 1. And the schemes that we have noticed, they all started off with 1 and then they fell off to 0 at pi; so, this is 0, right? However, you notice, that what happens is, suppose I evaluate the second derivative by explicit scheme, so if I am trying to do it like this; if I do it like this, and if I use the Fourier Laplace transform, what do i get? I will get 1 over h square integral, here I will get e to the power i k h shifted, so, I get e to the power i k h, here I will get minus two; here I will get e to the power minus i k h, and this thing is multiplied by u of m, u of m is nothing but u of k e to the power i k x m d k.

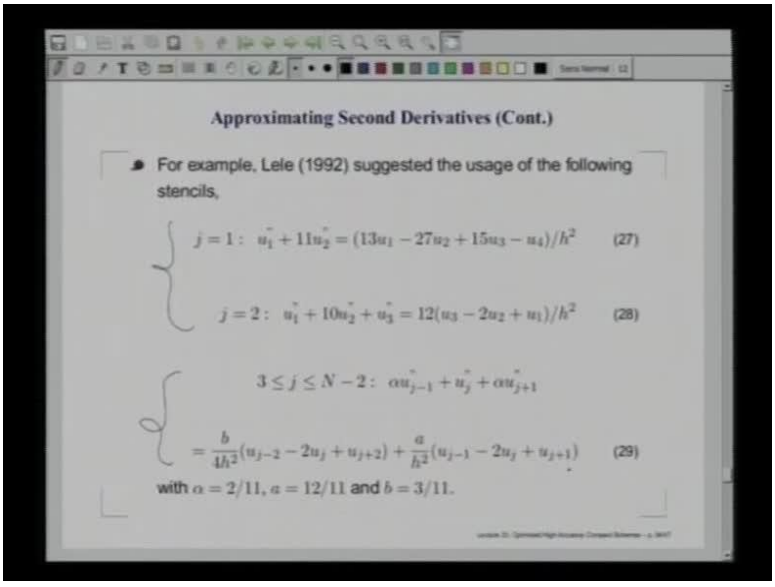
So, what you are getting here, is actually this whole thing taken together plays the same role as what we have here. So, this is also, I could call it something like my k equivalent for the second derivative; so, what I would do is, basically I would see what it is. So, if I

call that as k equivalent for the second derivative, not square; that is why I have put it in the bracket, that will be nothing but these two will give me 2 cos k h minus 2 by h square, right?

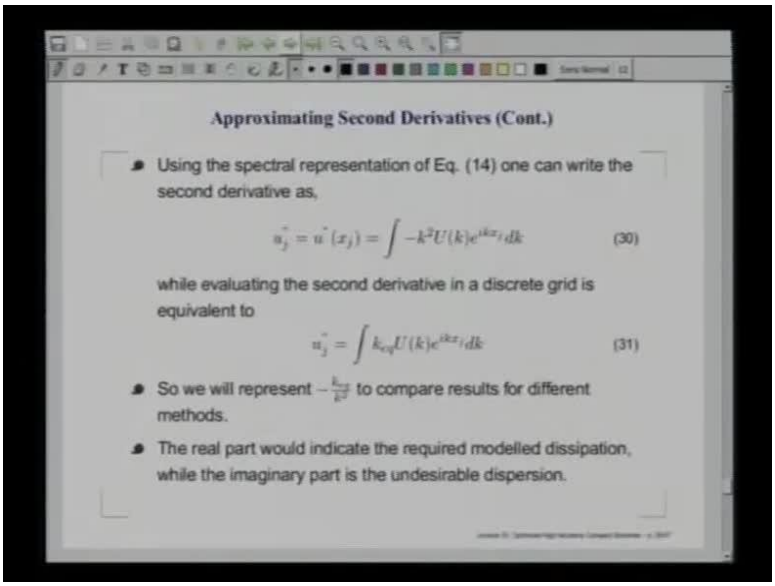
So, what I could do is, I could write it as 2 by h square, and this will be 1 minus cos k h, 1 minus cos k h, right? So, what I am going to get, is basically nothing but sin square k h by 2 divided by k h by 2 whole square. If I decide to divide this by k square, and I would do it, because there is a minus sign here, I will put a minus sign here; and make this as plus.

So, you see what happens at the limit k h equal to pi, what happens? k h equal to pi, this becomes 1, right? This becomes 1; and this is pi by 2, so, I will get this is equal to 4 by pi square at k h equal to pi; that is a remarkable thing. So, second derivative, If I do it by even a finite difference, c d 2 form, then at k h equal to pi it is not 0. So, it is somewhere, some value here, so that could be something like this; so, this is your c d 2 representation. But if you notice that, if we try to evaluate the second derivative by using the stencil of first derivative twice, I am going to get it to 0; and that is a very bad news.

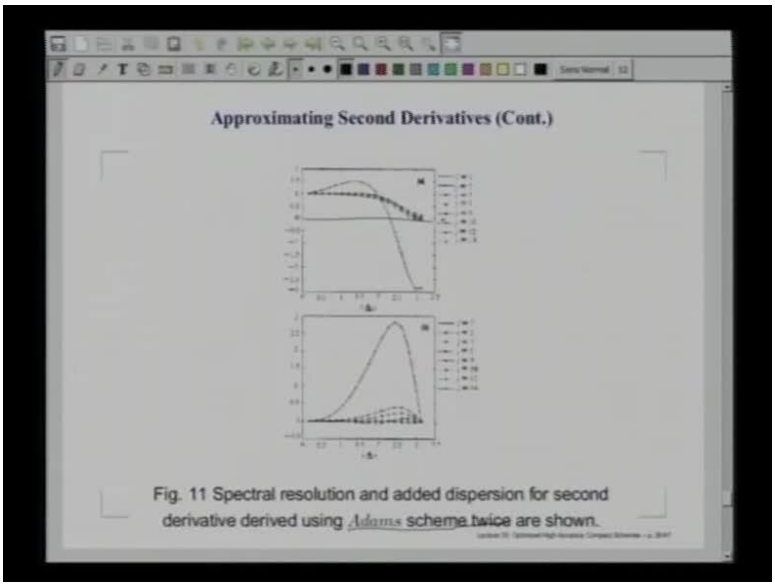
(Refer Slide Time: 47:55)



(Refer Slide Time: 48:15)

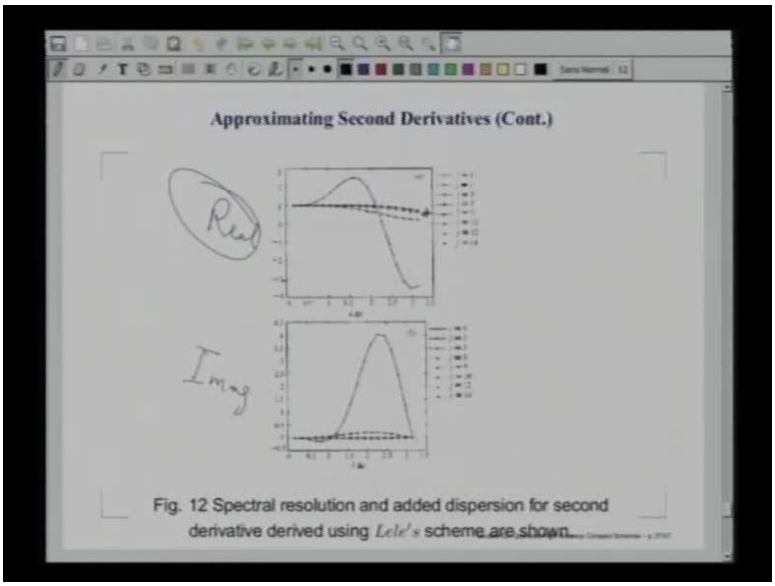


(Refer Slide Time: 48:28)



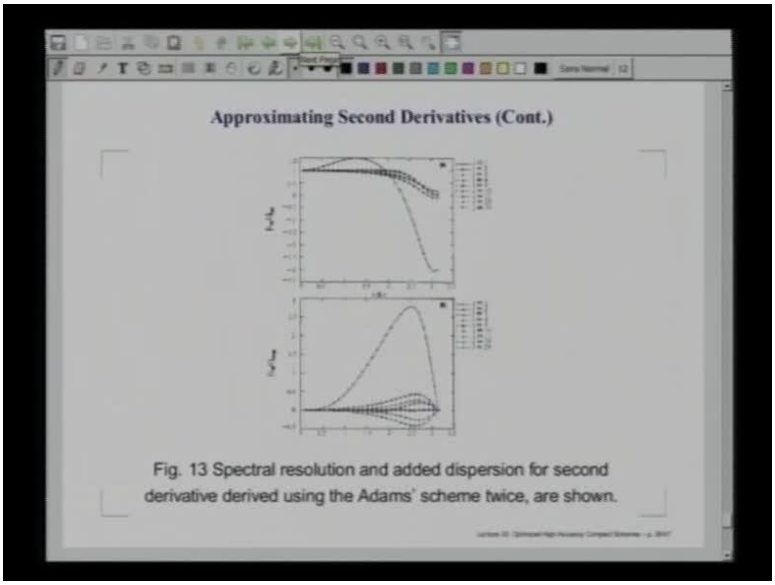
So, this is not something what one should really look at. Repeating first derivative algorithm twice is not a good idea; and that is where Lele actually suggested a host of equations. This is the main stencil, and that is supplemented by those boundary closures for j equal 1 and 2. Notice that the second derivatives are related to the function directly, so, this is directly using compact scheme for the second derivative. You do this, and then you can work out the same way that we talked about just now; that we need to really plot minus of k equivalent by k square, and then we will see its performance parameter; and that is what you get, if you look at this.

(Refer Slide Time: 48:51)



This is the Adams scheme for first derivative applied twice; and you can see that all of them actually come to 0, right? Because, that is the property; in contrast, if you look at the Lele scheme, what you notice is, that most of the points, they are here; and this value is roughly about 0.7, 0.8, what is this value? This is roughly about 0.4, right? pi square is above 10, so, it is about 0.4

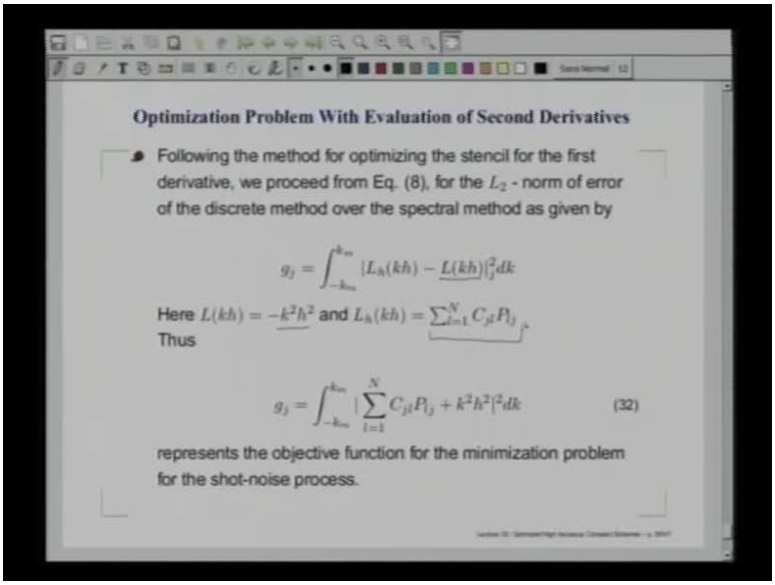
(Refer Slide Time: 49:45)



So, in the Lele scheme, you get much better. So, if you write out a compact scheme directly for the second derivative, you can get it. And you see, it will also a real part that

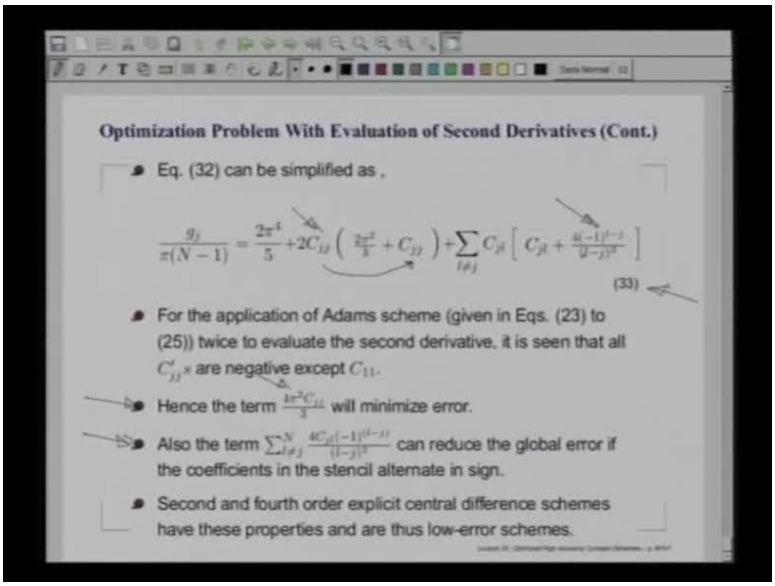
is shown here; so, this is your real part and this is your imaginary part. What does imaginary part do? Imaginary part for a second derivative will be odd derivative, so, it will add dispersion; it will not be dissipation because, this itself is dissipation, real part itself is dissipation; so, we should keep that in mind, what it does. So, this is what we have already talked about Adams scheme.

(Refer Slide Time: 50:14)



Now, what we could do is, we could go through the same optimization issue; the way we have talked about first derivative, we talked about how Haras and Ta'asan obtained their values. So, we can go through that same exercise; all we need to do is, we need to find out the exact operator; exact operator will be what minus k square, right? So, that is what we have done; and most of the time, it is scaled by h square, so, we have to non-dimensionalized it and written it as minus k square h square; that is your exact operator.

(Refer Slide Time: 50:54)



And, as we have discussed for the first derivative, we can for the discrete operation. We can write in terms of some c matrix times those projection operator; we can plug it in and open it up. Go through that exercise that we have done for the first derivative somewhat, and then we come out with an expression given like this.

Once again, we can see the choice of c matrix will determine what kind of quantities that we are going to get; and what we can see is that, this quantity is…; we need to find out which are the quantities that can reduce global error. This could be one because, that could flip signs, right? minus 1 to the power l minus j, so, that that that can do that, and of course, we can try to basically keep the sign of c j j diagonal term; that will tell you say, for example, here the diagonal term is here minus 2, so that is why, such a term actually leads to error reduction; but not this c j j square, will not. It is only the this part, that is what we have noted down here, that that term will reduce error, right?

This term, of course, will reduce error; so, basically we are looking at the second and the fourth term that can reduce error; and we can see that, this is one way of minimizing error. So, I would like to probably stop here itself, and say that we have stated the way - how we can optimize compact schemes. In doing so, what we really need to worry about, is not only the central stencil, but the boundary closures are equally important; and we have seen, like Lele scheme. Here, Lele scheme actually does take care of that; you actually take a central scheme you do not want added dispersion due to second derivative

discretization; and the boundary points are also very nicely taken, and that is why you do get the following property. Although you get some bit of dispersion effect, but this point is for j equal to 1, so, it does not matter. The same way, this point is not to be something we should be worried about; by and large, it provides you very decent second derivative.

We will see in future, how this second derivative is so important, also discretization also, for controlling aliasing error. We will talk about in one of the following lectures, a new type class of schemes called combined compact differencing scheme, where we simultaneously obtain first and second derivative together. What we have talked about today is, we have talked about obtaining the first derivative separately; from obtaining the second derivative separately; but in a future lecture, we will talk about compact one. May be, the next lecture, we will talk about the combined compact scheme, and you will see what wonderful things it can do. We have obtained some results which are quite accurate and revealed lot of new physics.

I think, with this I will stop here and for today's lecture; and we will come back to combined compact scheme in the next class.