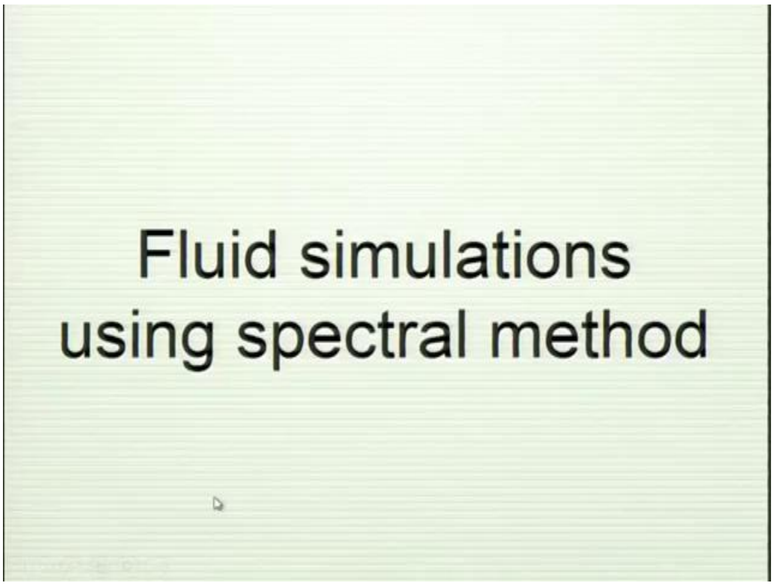
0Physics of Turbulence
Prof. Mahendra K. Verma
Department of Physics
Indian Institute of Technology, Kanpur

Lecture – 25
Energy Transfers
Fluid simulations using spectral method

So far, we have studied various instabilities which takes you to turbulence then I did energy transfers in the last class, now I will discuss how to simulate a hydrodynamic flow and how to compute the energy transfer. I will not describe it in simulation, but how to compute fluxes which are very critical and after that I will go into Kolmogorov theory.

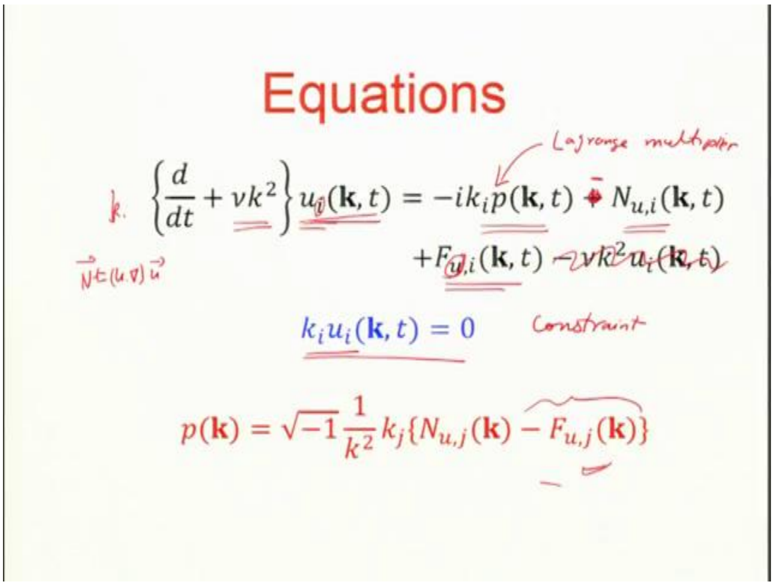
(Refer Slide Time: 00:59)



So let us start. It is simulation using spectral method. There are many methods - finite difference, finite volume, finite elements, vortex method, but in this course we will focus on spectral methods. I like this and it is it has advantages as well as disadvantages. One advantage of this scheme is that it captures scale by scale flows or flows at different scales, large scale and small scale. there are various wavelengths. I am going to basically Fourier transform the flow. they will give me amplitude at different scales like you have already done, it captures amplitude of the flow at different scales which is not possible with finite difference. You will have to work as a post processing after you get the data where every point which is equi-spaced or maybe there is some points are close by but, you find the velocity at different points at a given scale. If somebody ask you what is a flow structure at large scale in finite difference scheme you will have to take the data and then do the transformation; Fourier transforms or something called POD – Proper Orthogonal Decomposition and this PCA – Principle Component Analysis all that is done on top of the data. But spectral method gives you from the output itself because flow is Fourier mode, the amplitude is the large scale structure, and also various transfers in different scales, I described in terms of Fourier.

We also described it in terms of real space which we will do it bit later, but Fourier space gives a very good description of how energy is flowing from one Fourier mode to another Fourier mode, one scale to other scale and that is why it is very useful. The one disadvantage of this method is that it kind of requires very idealized geometry. Fourier transform means periodic box. I can work with free slip box, others things like no sleep or a specially tube or turbines these very difficult to do in spectral. In fact, it is impossible for some flows. So, many engineering flows we do not go for this method.

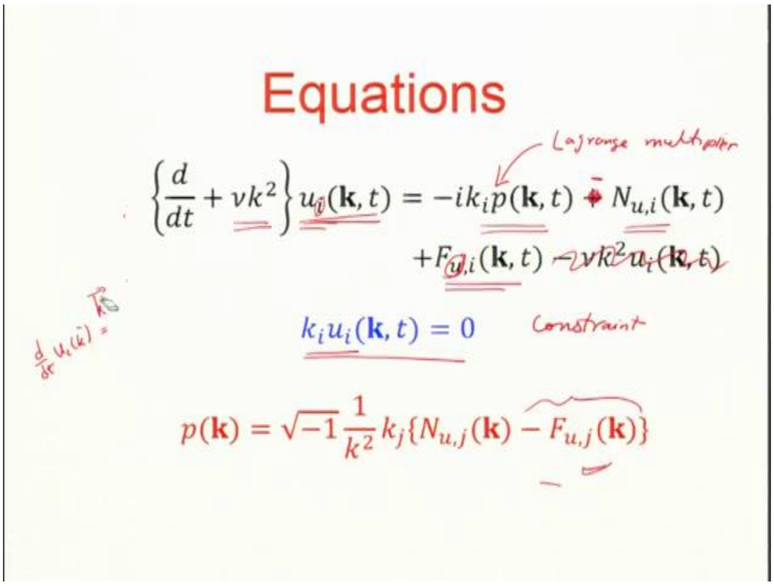
(Refer Slide Time: 03:56)



So, with this background let us start how to solve our equations in spectral space. Let us take only hydro, but I will also to generalize it to a convection or MHD. Let us just work with that hydrodynamic flows. So, this equation (see above figure), I have done it many times. This is in Fourier space ok. We will compute the non-linear term by some trick which I will tell you bit later. We can add external force if you like. Now, a divergence free condition that is $\boldsymbol{k} \cdot \boldsymbol{u}(\boldsymbol{k}) = \mathbf{0}$ which is straight forward. So, this is a constraint, this is not an equation, it does not help you in evolving the velocity field, but it is a constraint on $\boldsymbol{u}$. We will have to satisfy, well basically, pressure will be computed using this condition which I told you before. See the above figure. So, this condition is basically helping us to compute pressure. Pressure, in mathematics language, is Lagrange multiplier.

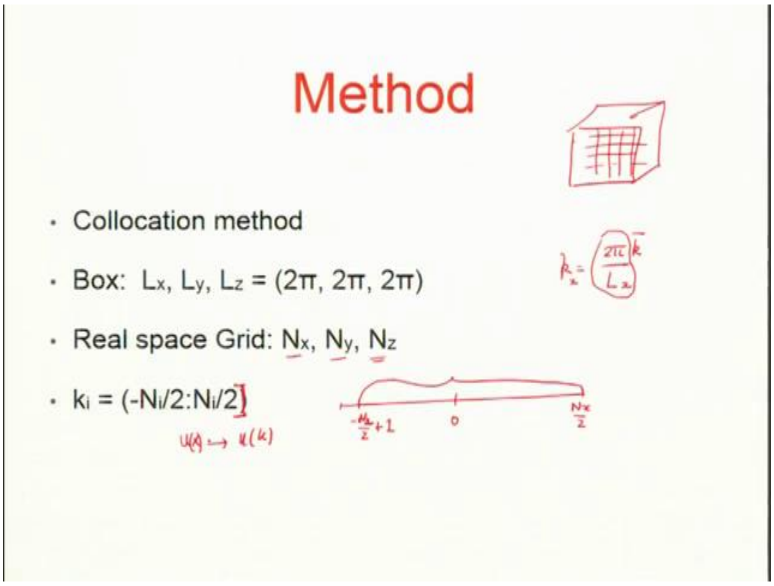
What is the method, it is a very straight forward. So, let us just go back bit once more. This is a set of ordinary differential equations (see below figure), but this is not PDE. I already converted to ODE and how many ODEs I got? Equal to number of Fourier modes or Fourier wave numbers.

(Refer Slide Time: 08:23)



Every $\boldsymbol{k}$ have ODE. It has three components. If you have one triad, great, but if you have more than one triad then you have to do a lot of work with Cartesian, it is not straightforward.

(Refer Slide Time: 09:23)

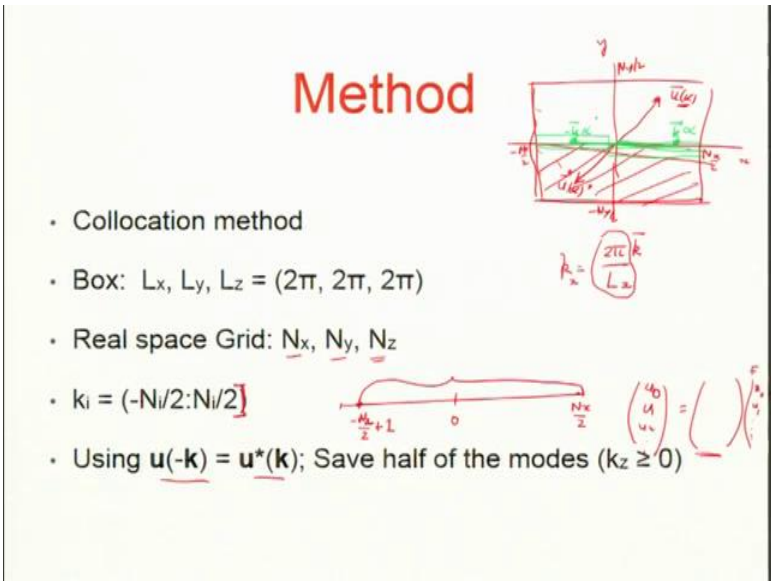


We solve this stuff; the method is called collocation method which I am I will not differentiate among other methods. See the above figure. I have my box. My box is $2\pi \times 2\pi \times 2\pi$. We can do it for arbitrary box, but for simplicity I am choosing this 2π. The reason why it is 2π is my wave number **k, k** is a vector, but let us say I am looking at k_x which would be $2\pi n/L_x$, n is an integer. So, k_x becomes integer, which is convenient, you do not need to multiply wave factor, but in our code, which is the lab code we could put arbitrary L_x. In real space, we have a 3D box and we discretize, like finite different scheme, discretize the whole space into fine mesh. Along x direction is Nx along y is Ny, along z is Nz.

Along each direction since it is Fourier decomposition, is $-N_i/2$ to $N_i/2$, Fourier wave number takes both negative and positive value. Now, we probably need to be slightly more careful, not both ends are not included. So, right end is included not left end. So, what I mean is it 0 is here N x by 2 is here which is included. But N minus N x by 2 is not included. So, the left-hand side is just 2 plus 1.

So, you can see that how many points are included, $-(N_x/2) + 1$ to $-N_x/2$. Fourier transform is the linear transform transformation. So, given u_x in 1D, I get $u(k)$ or vice versa. See the below figure.

(Refer Slide Time: 12:36)

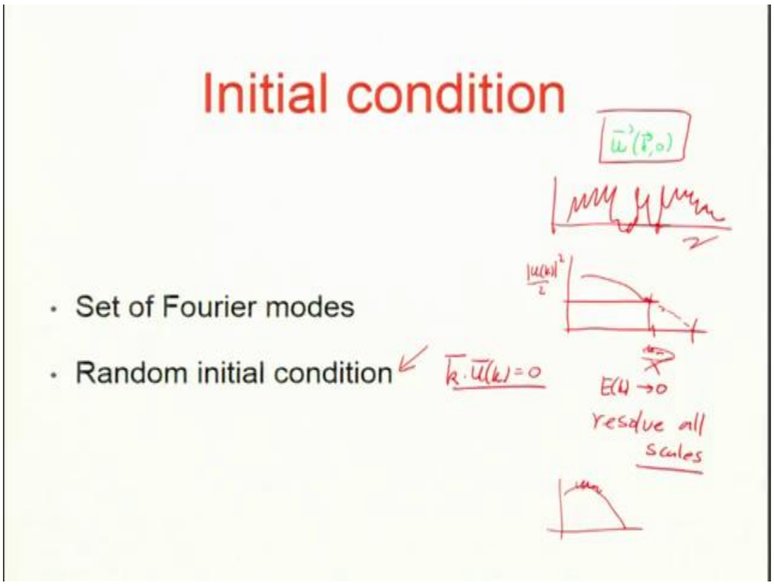


If you notice it if I just write for 1D. So, 1D is u_0, u_1, u_2 .... is equal to matrix Fourier F. See the above figure. And this matrix can compose of e^{ik_x}. A matrix must be $N \times N$ and that is determinant not equal to 0 for unique solution and invertible.

The reality condition says $\boldsymbol{u}(-\boldsymbol{k}) = \boldsymbol{u}^*(\boldsymbol{k})$ to be satisfies because velocity field is real. If we have some wave vector here (kx, ky), then the negative direction will also have a vector. So, if this is vector $\boldsymbol{u}(\boldsymbol{k})$ and this is $\boldsymbol{u}^*(\boldsymbol{k})$, I do not need to save. In fact, I do not want to save that part. So, if there is any errors then reality condition will be violated. I am just taking only to elaborate. So, when I time step this number and this number may not be complex conjugate after time stepping. So, best is to get rid of this. So, you simply do not save this. You save only half of the modes in 3D, kz greater than equal to 0, in 2D only ky is greater than equal to 0. There is only one catch, the catch is the following. I am going to not keep anything below the x axis, but I have to keep the x axis.

Because there are 0, is a very important point, you have to worry about 0, but if you keep this part then what is a complex conjugate or minus k for this wave number? It is on the line itself. And, I want to keep the full line. It is too much level if you keep half the line. Ideally you would like to cut it like that then you do not need to worry about this half, but if you keep the full line which is the convention then you have to make sure that this point here the velocity field is complex conjugate of this. So, this condition has to be satisfied. If this is α, then this must be α*. This reality condition we need to make sure that it is working otherwise your code will have problems in 3D simple generalization. If you are doing it properly then it is quite easy.

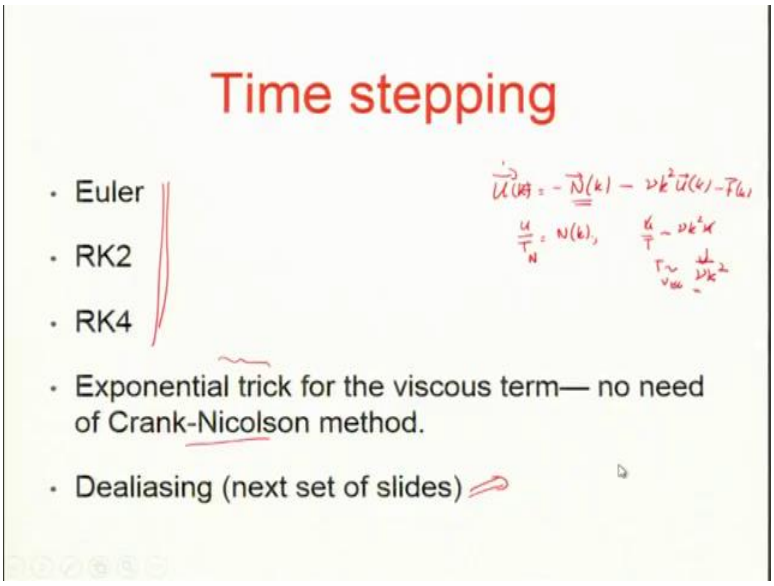
(Refer Slide Time: 16:56)



Initial condition, this is a initial value problem. Given the initial velocity field $\boldsymbol{u}(\boldsymbol{k})$, all wave numbers at time $t = 0$. I am working with the initial condition in spectral space or Fourier space. If you are given in real space, you have convert it into Fourier space. Now, how many points should we choose? How many $\boldsymbol{k}$ you should choose? It is a critical point. Any signal will have fluctuation in different scales. My signal, let us imagine my signal, one component of velocity field is like this random (see above figure).

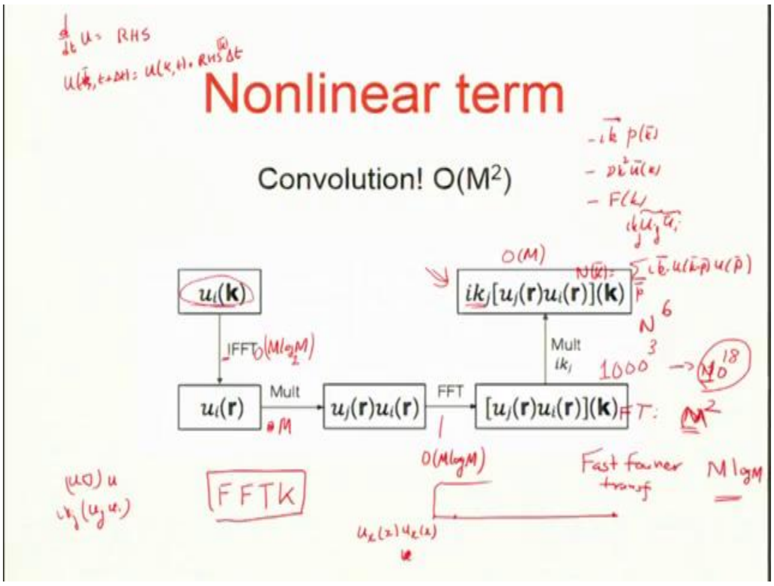
Now, I want to do 1D spectral method. So, what would be my N? what will determine N. Look at the fluctuation at the smallest scale. I need to take care of all the fluctuation which has significant energy. I should in fact, go to smaller scale where this energy becomes 0 or 10^{-8}. If you miss out any signal, then there is a problem. You need to resolve all scales. You should keep N larger than what you are putting in mode, but you can choose the Fourier modes in initial condition or you can also use random initial condition. I mean this is like a random signal, but my phases are random. The waves should this generate random initial signal which preserves divergence free conditions.

(Refer Slide Time: 21:32)



Now time stepping; once you have the initial condition , it is the initial value problem for every Fourier mode. There are many schemes - Euler, RK2, RK4. Runge-Kutta forth order, there are other schemes which we do not have in our code, but I am now going to describe them anyway. So, there is something like Adam-Bashforth, but sometimes stepping free. And, so, these are there is one important one to keep in mind is that there two there are many time scales; turbulence is multi scale problem. For every k there are two time scales – inertial and viscous time scales. These two time scale can be very different. There could be problem of stiffness. If the two time scales are different then you choose smallest of the two and that can be very expensive. So, one idea is to get rid of the one of the time scale using a trick called exponential trick. We can absorb the viscous term by change of variable. No need of Crank-Nicolson for spectral method for in periodic.

(Refer Slide Time: 24:53)



How do you compute the non-linear term? I compute pressure from my non-linear term and force. We can compute all the terms very easily except the non-linear term.

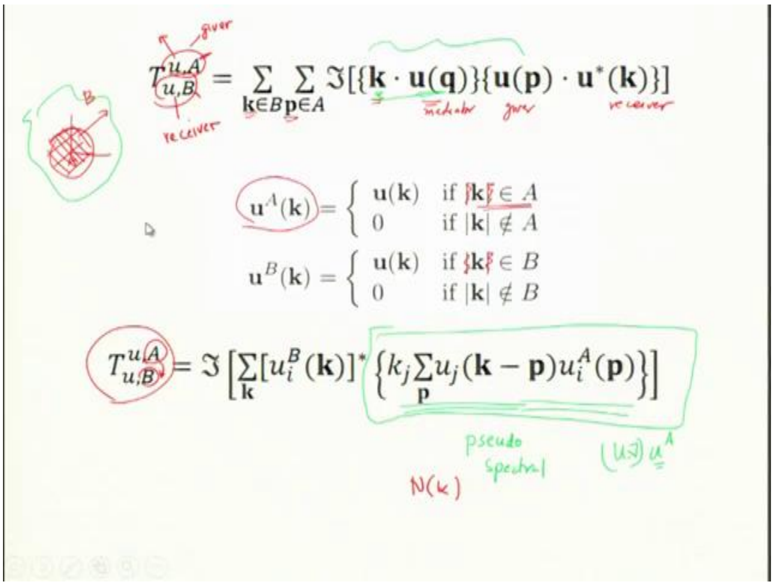
See the above figure. For every k how many products a convolution has? If my grid is N^3, how many products I have? For every **k,** I have N^3 multiplication. Now, how many wave numbers I have got? N^3. So, convolution not possible. There must be well, they are not coming at all.

The brilliant Orszag suggested very clever trick. He says do not do convolution. Now everybody knows convolution is done not directly by multiplication. Using fast Fourier transform. They has the scheme where it could be done in N*logN. There is a big reduction. Now, this N, let me call it $M = N^3$. According to this algorithm fast Fourier transform, it is M log M not M^2. If you have 10^9 points, then it is big reduction. This is why we do not do convolution. In fact, for all unit processing, signal processing, nobody does convolution, is used using fast Fourier transform.

(Refer Slide Time: 35:14)

Now, what about energy transfer? I defined all of it, I defined flux, shall to shall. Energy transfer from any region A to any region B. See the below figure.

(Refer Slide Time: 37:09)



How do you compute? By the way any region A any region B, energy coming out of the modes inside this sphere to the mode outside this sphere. So, this is the region A and outside be the region B ok. We have convolution here as well p is a k equal to p plus q, but remember p is a giver lies in region A and k is a receiver lies in region B. So, now, giver is not all the wave numbers; giver is only belongs to region A. So, I construct a new velocity field.

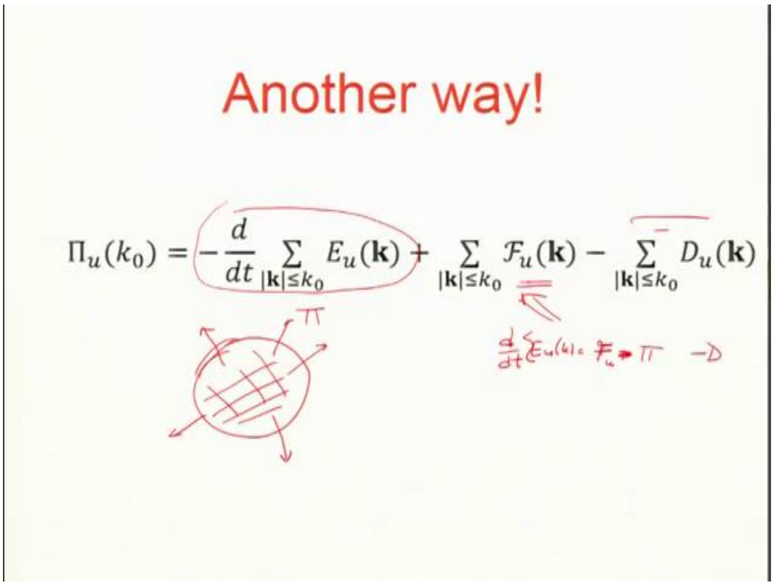
New energy field u^A which is $u(k)$ if k belongs to A. In real space it corresponds to large scale flow. See the above figure. Now u^B is other region. So, we do the same thing. I keep the velocity field non-zero only in one region. Now, this is again a convolution. We use idea fast Fourier transform again and the idea is the following. I rewrite the above equation like this (see above figure).

But it is not full non-linear term constructed by full velocity field, non-linear term constructed by u^A is the same. I can use the same old algorithm which I did in the last slide and just use that function. I go to Fourier space, multiply Fourier transform and then multiply by k_j. This is called pseudo spectral method. Why is it pseudo spectral?

Because I am going to real space, it is not fully spectral, you are cheating in between here, you are not solving everything for your Fourier space, you are going to real space, doing something and coming back and that is why it is called pseudo spectral. Once I compute the Non-linear term, then I multiply by $u(k)$ and sum over k's. I sum over only the B part of velocity field and once I do this and I get this T which is a derived. Finally, this is that energy transfer from A to B.

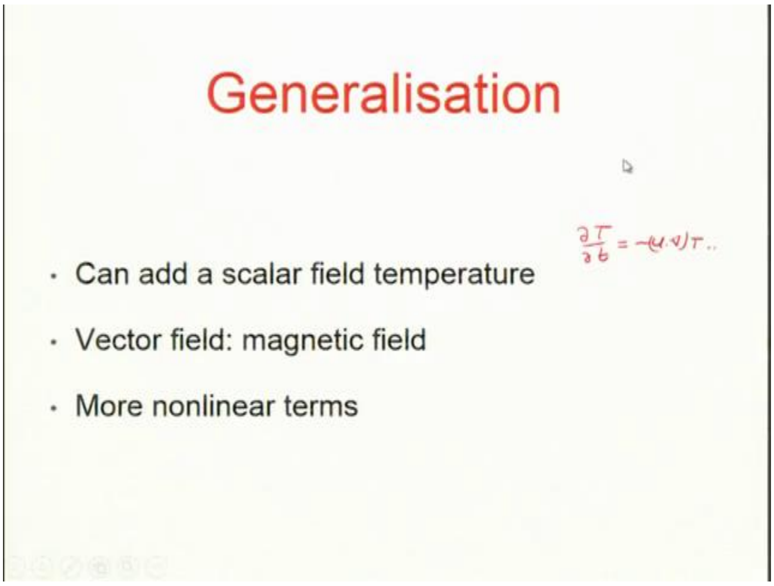
Top is the giver and bottom is the receiver and the first letter is field. Now, I am only using velocity field, but sometime later we will also use magnetic field. So, this may change to b. So, to compute flux I just had to put region A is inside this sphere and B is outside this sphere and they will give me the flux. This is how we compute flux in our simulations.

(Refer Slide Time: 43:08)



There is another way which is also used to calculate flux. See the above figure. What is the flux? So, the energy is coming out by non-linear interaction to these flux. Now, the question is how energy can change inside this sphere? Energy can change inside this sphere by several ways, one thing is I am losing by viscosity and the second, I lose by the non-linear interaction by exchange. This can be positive or negative and third is I get from some external force. If flux is positive then I am losing. I can compute the time rate of energy of this sphere at two different times. All the three terms in the right hand side can be computed. To compute the time derivative I need it to different times, but all three terms can be computed. Once I compute all three terms I can just add subtract I can get flux. So, this is another way to compute Kolmogorov flux. So, you should keep dt small. So,. So, basically, we can solve our Navier-Stokes equation, we also compute various energy transfers.

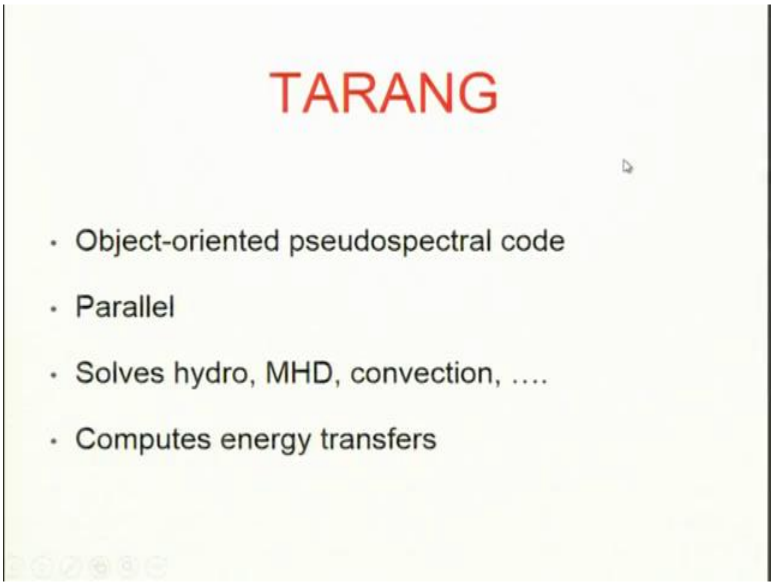
(Refer Slide Time: 46:08)



Now, you can generalize it to more complex fields like convection, we have an additional field, temperature. So, we have another PDE partial differential equation. See above figure. We can solve exactly the same way, compute the non-linear term by pseudo spectral method, viscous term will be a diffusion term, follow the same scheme.

For MHD, there is one more equation for the magnetic. If you have magnetic convection then we have three fields velocity, magnetic and temperature, but no problem we can just solve this using same method. So, having more equation, method is the same, but just you have to take care of more variables, more non-linear terms and so on.

(Refer Slide Time: 47:05)



Now, I just want to tell you, there is a code TARANG developed in our lab, it is object oriented. It is object oriented because we have one code to solve, fluid, MHD, convection, rotating flows. This it is a general PDE solver, PDE spectral solver. Since it is general, I can plug in more variables plug in more non-linear terms and we do it. And It is parallel, means I can use many computers to solve this equation, it is faster otherwise this will take a long time impossible in fact, if you do not use parallel. This is advantage of this solver. We compute the energy transfers and it gives you a lot of insights. I think this is about my discussion on a spectral solver except aliasing, and FFT.

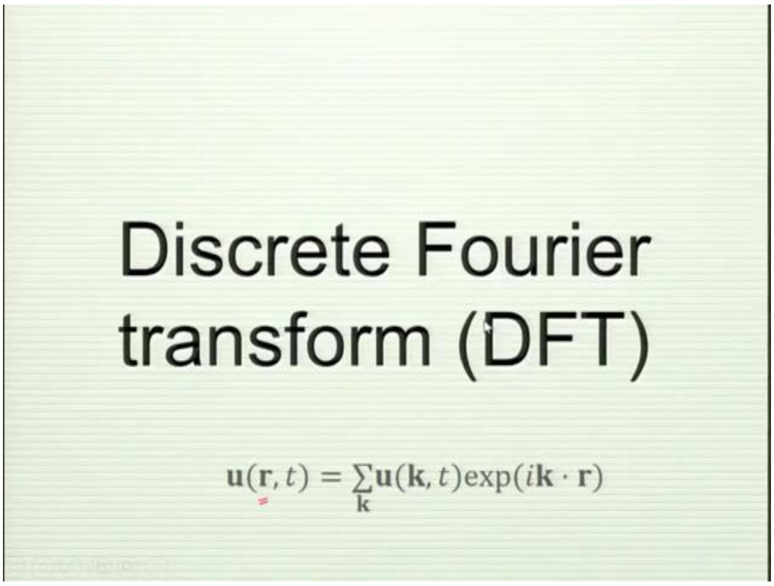
Thank you.

Physics of Turbulence
Prof. Mahendra K. Verma
Department of Physics
Indian Institute of Technology, Kanpur

Lecture - 25
Energy Transfers
Fluid Simulations
Dealiasing

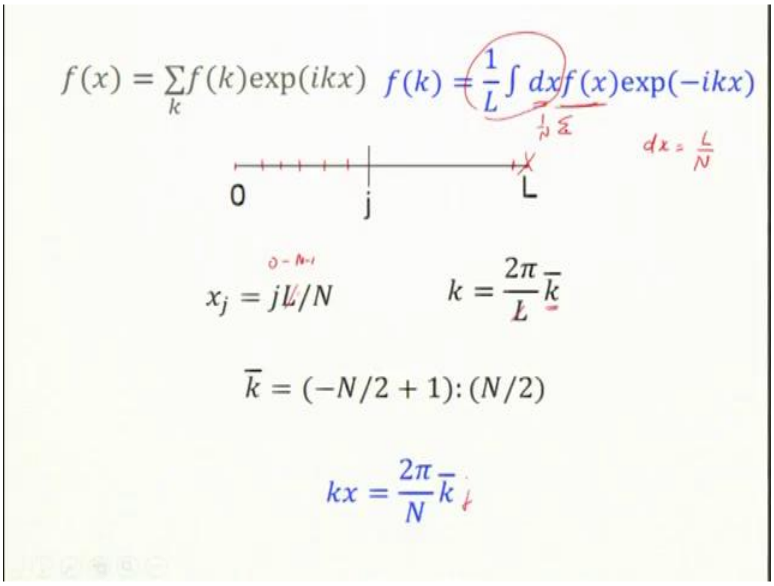
In this lecture, I will give a brief intro on Fourier transfer, FFT. We will discuss how it is done. I think is it is interesting to discuss dealliasing. I will just briefly explain that. Now we will do discrete Fourier transform, both real space and Fourier space.

(Refer Slide Time: 00:34)



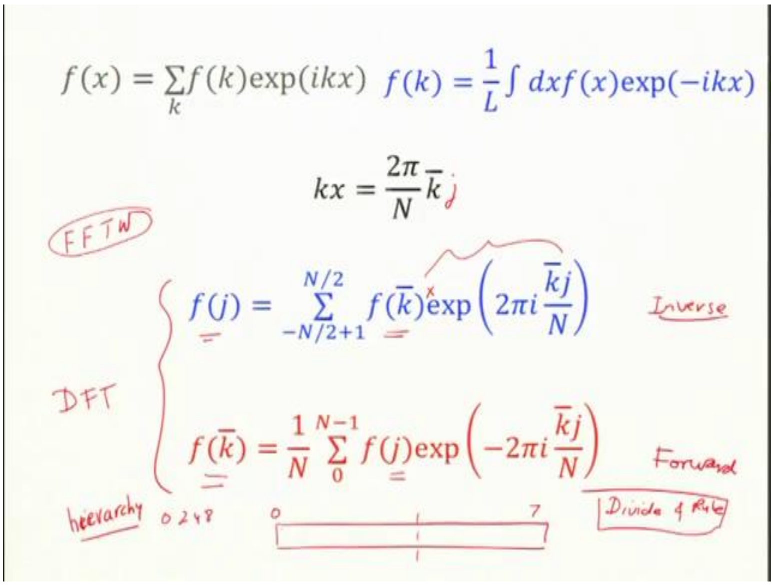
Now, if you recall you know let us say x was a real, but k was discrete, but in simulation both x and k are discrete in computer, d for discrete and r for real. I will illustrate using 1D all these discussion.

(Refer Slide Time: 01:11)



For Fourier transformation see above figure. We have this 1D line, 0 to L a periodic. You do not save both the points to satisfy the periodic boundary condition. So, I have N points. See above figure for indices and discretization.

(Refer Slide Time: 03:27)



Finally we get Fourier transform of $f(x)$ as $f(k)$. See above figure. This is called DFT Discrete Fourier Transform. This is also called forward transform and the opposite is called inverse Transform. There is a reasonably good manual called FFTW, Fastest Fourier Transform in West. If you look at their website you will find some more details.

As I said it will take given N point it will take N squared if I just do my ordinaries. This is very expensive as this is sum as well as multiplication. So, the multiplication here of this exponential and this of course, computing exponentially also it is done live it is not saved in a data saved in array, exponentially is computed live and so, this is multiplication component exponential and there is a sum.

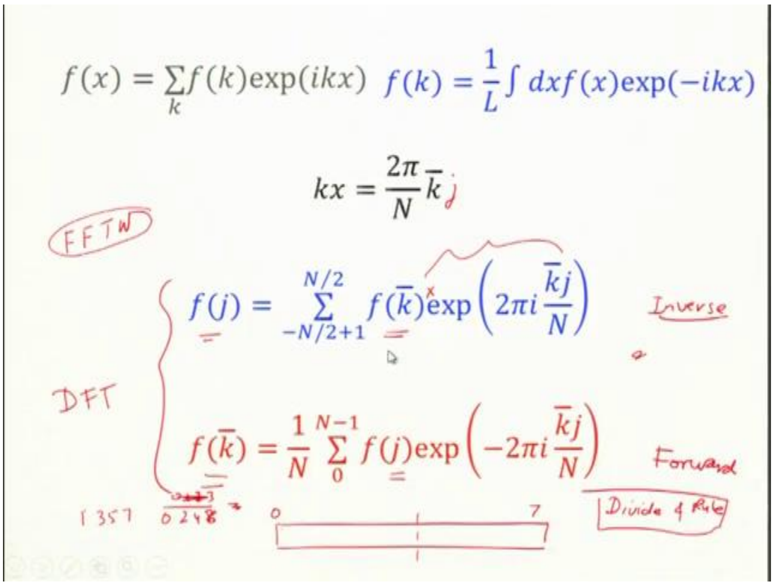
So, how many sums and how many multiplications are there? So, for every j I need to compute N times and there are N js. So, it becomes N squared in 1D and in 3D this is easily generalizable. So, it is M, M is number of points.

Idea is to divide this in even and odd. This called divide and conquer, this is a British rule recorded we divide and rule. So, divide this data into even and odd. So, not divide like left right ok; so, even and odd and then again further keep dividing. So, you make a like a tree. So, instead of doing this thing with equal, then you say you compute even odd even odd. So, basically it is dividing literary. So, instead of doing it in one ways, then you just sort of make a tree structure and solve in hierarchically.

So, this one idea which I am drawing myself in some different fields, let me just make a remark. So, this is solving a problem by hierarchy scaling hierarchy. So, you do not solve a problem in it at equal level. So, the thing is I like to say if you want to solve a country problem countries problem you do not go to every house and solve the problem it will solve at a in multi-scale level.

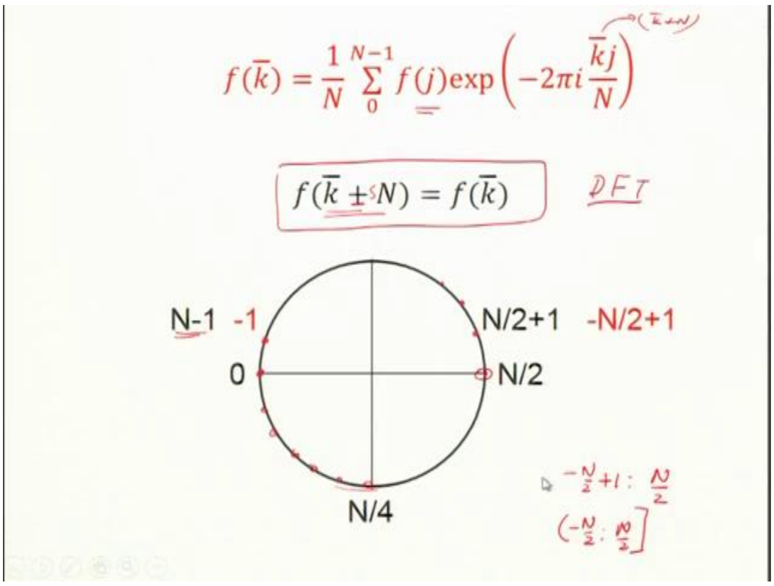
So, you have to make a policy at large at the country level, then district level, then state well village level. So, you have to solve in a multi-step way. So, it is like divide and conquer you do not do all of it in one shot, it just too bulky. So, divide even odd even odd and it turns out it is done in N log N and this is the very powerful idea. So, this is was even in mathematics.

(Refer Slide Time: 07:44)



That is how it is called dividing and once the data becomes small then you can do the Fourier transform on small data and then you go up. So, this is if you look at the we keep here find the detail. So, anyway we will not get into this, but this is done in N log N now FFT.

(Refer Slide Time: 08:52)

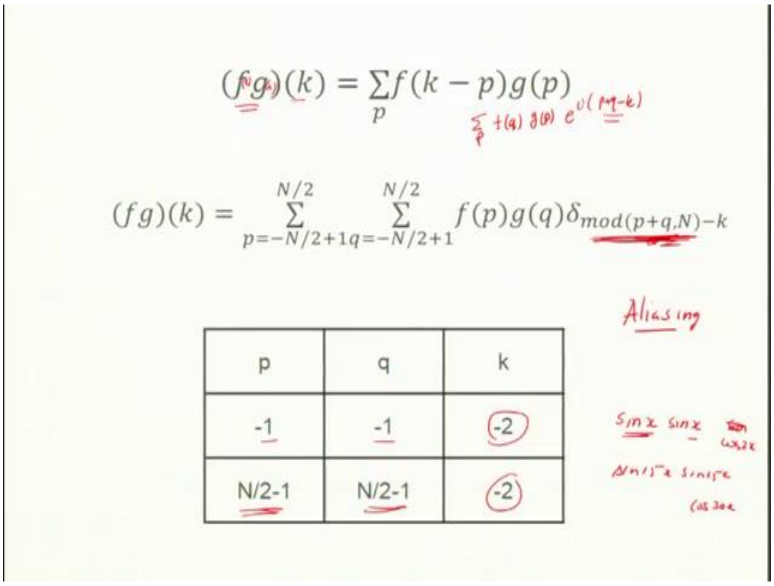


So, this what we need to compute $f(k)$. So, these are another property of Fourier transform, this property of discrete Fourier transform DFT. See above figure. So, this is really important property. So, it is because of this properties nice to represent you know circle this is a followed in many textbooks. So, my data is starting from here and I just put them is points on the circle. So, these N by 2 then this N by 2 plus 1, but by using this property N by 2 plus 1 is I just subtract N add or subtract know.

So, what if I subtract and what will I get? Minus N by 2 plus 1 so, these are points that N points what is this N minus 1. So, what is that N minus 1? Subtract means minus 1. So, all the negative ones were here all the positive ones are below if you like till N by 2, this point is N by 2. So, positive ones from 0 to N by 2 lower, lower, lower half and above part is negative is that clear?

But in for solving this problem it is important it is sometimes useful to think 0 to N minus 1 do not think negative positive. But for physics it is important to keep in mind that it is like that or we can write like this, this include you know square bracket means included left bracket means not included is that clear; so, this notation. Now, make life bit difficult for you. So, this called dealliasing problem. So, what is the problem first?

(Refer Slide Time: 11:45)



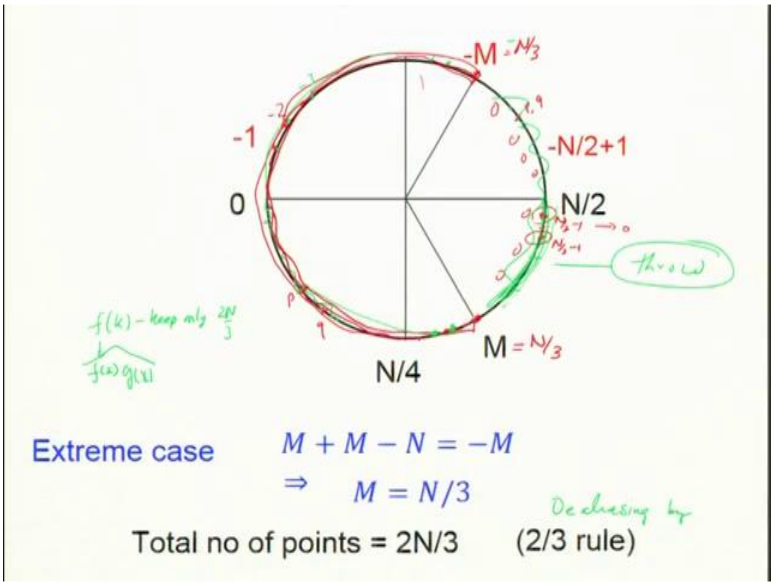
The two functions f and g I multiply these in real space, see above figure, then do the Fourier transform. Now it is a convolution in 1D, there is no vector.

So, this is fine, but when I do discretization, then I do not have minus infinity to plus infinity I have minus N by 2 plus 1 to N by 2 that is where the problem is; where convolution will have error because of discretization. If I had infinite N then no problem convolution will be exact, but because of convolution there is a chance of error and what is the chance of error.

This is a proof which I did in the class, see the figure. I get a delta function Kronecker delta function, δ. Because of this circular property $p+q$ cannot be arbitrary large, k is bounded between $-N/2$ to $+N/2$.

So, is when I do the Fourier transform by discrete Fourier transform these what I will get; In the following picture it becomes clearer both p and q are going from $\frac{N}{2}+1$ to $N/2$.

(Refer Slide Time: 14:19)



Now, the dealiasing is quite easy to see. See the figures. sin x I think multiply by sin x, I get sin 2 x or sin square x cos 2 x know it gives you cos 2 x which is 1 plus 1 gives you 2 sin has plus minus 1. So, please bear with me on that part. But because of this property large wave numbers can also give us small wave number.

So, these are problem, if I have large wave numbers I am adding them, if I have sin 15 x sin 15 x will give you cos 30 x if 30 x there is no place to save then is going to come back by this by this rule is that clear. So, this called aliasing. So, this minus 2 has 2 possibilities. So, I get coming 2 names. So, one thing is coming contribution coming from minus 1 and minus 1 other one comes from this contribution.

So, to get a wave number to I just sin x and sin x i multiply, but somebody tells you that sin 15 x and sin 15 x, multiply gets sin 2 x then something is odd right something wrong I cannot multiply 2 large sin functions, but computer will give you that. There is a problem of DFT because of this rule. This called aliasing problem.

So, I have 2 wave numbers p and q add them I get somewhere here. So, they are small numbers 1 plus 1 gives you 2 no problem, but it becomes large then this going around happens if I take some number here N by 2 let us say N by 2 minus 1 N by 2 minus 1 that is what I was taking you know in the my previous table if I add them I will get N minus 2 which comes somewhere here minus 2.

So, minus 2 can be obtained by minus 1 plus minus 1 in convolution or it can also come from here if these modes are nonzero if they are present now that is the problem having 2 large wave numbers giving you a small wave number is not a desirable thing ok. So, what do I do? I make some of these 0 I demand that in fact, the demand is. So, I am going to make a circle I I could not do it in the computer only these are nonzero.

So, now, if it is 0 then multiply I will get 0. So, to what is the worst scenario? So, look if I take 2 large wave numbers add them I get large wave number.

So, I want to put a condition that I will get if I had them I get somewhere here it is a negative number, but somewhere here I am going to explain this in minute ok. So, these are condition and that gives you M is $N/3$. So, this I want to make is $N/3$.

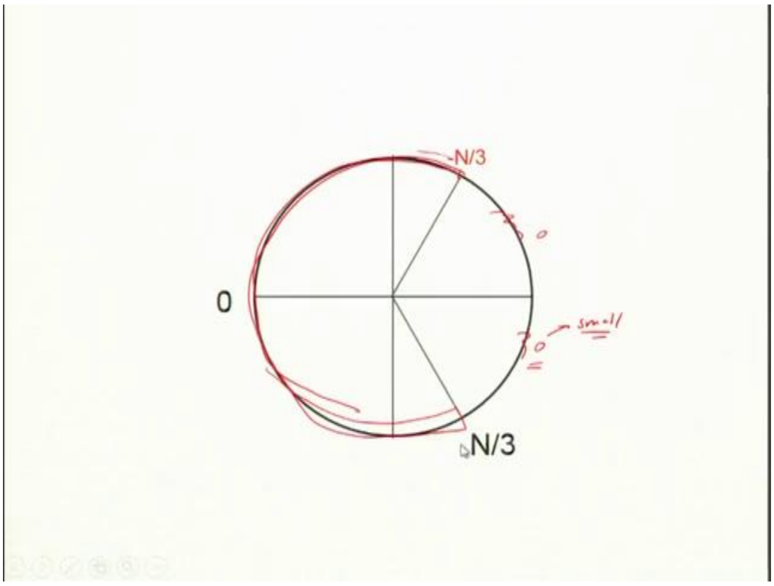
So, let me explain what it means. If I have my wave numbers somewhere here then if I add them I will be if I lie within this band lower band, I am fine because my I am going to keep only this wave numbers. So, when I add my k my result is valid result there is no aliasing problem. If I have somewhere if I my problem is if I have somewhere here then what happens? If the 2 of them lying somewhere here then add them adding them will give you it will come somewhere here, but these I am not keeping I am just throwing it out.

So, no problem this is you throw it out throw garbage your house is safe no problem. So, you throw that is garbage, but anything if we keep let us say p and q are here. So, we are lying inside this band no issues. But if I am here then of course, it will generate a k outside my range, but I will throw it out.

If I am come somewhere here then what happens? In fact, this is big and it comes somewhere here minus becomes big minus and that is like you add plus N to that it becomes positive. So, these again I throw. So, throw this part throw. So, I keep my data from minus N minus N by 3 to N by 3 then I am safe.

So, the idea is the following f full of k set keep only $2N/3$, throw out the other things. Now from these you construct real space from here I construct the $f(x)$ and g(x) multiply then again to the transform, but we keep only the data of 2 minus N by g 3 to N by 3 that is a reliable data. I will get nonzero amplitudes, but I do not keep it.

(Refer Slide Time: 25:22)



So, just to summarize, you keep this data only here and these are the zeros before the before we going to real space, multiply and after the data is done you again throw them out ok. In our code we keep this data row after the transform these numbers are small typically well large wave numbers the amplitudes are small, they are not correct not accurate, but we keep the data for a spectrum calculations it gives a bigger range.

Thank you.