**Course Name: Turbulence Modelling**

**Professor Name: Dr. Vagesh D. Narasimhamurthy**

**Department Name: Department of Applied Mechanics**

**Institute Name: Indian Institute of Technology, Madras**

**Week - 11**

**Lecture – Lec62**

# 62. Introduction to Direct Numerical Simulations (DNS) – I

So, let us get started. So, today we look into what is called direct numerical simulation. So, this is one of the techniques what we call let us say an eddy resolved technique, eddy resolved method. So, this is resolve means capturing or computing, right? So, you are going to capture these turbulent vortices and eddies. So, you may question what were we doing so far in the RANS, right? You are modeling them.

So, in RANS you modeled all the eddies except the largest one, right? Entire spectrum. So, this is opposite to that where we are going to capture and there are many methods. So, this is one of the method. DNS is one of the eddy-resolved methods, and here, another point to remember is that we do not use any turbulence models here.

No usage of any turbulence models. So, then the question comes then why are we studying this? This is a turbulence modeling course. Then why are we studying a technique which does not use turbulence models? So we will look into this why this is useful. And we have been looking into while modeling we take help from DNS data to actually model. So, the data here is coming from first principles from Navier-Stokes equations where we are not modeling turbulence at all.

We are entirely relying on Navier-Stokes to tell us what this flow looks like. So from that we can, once that flow is computed, you can, of course, apply your averaging techniques. You can decompose the flow, look into its budget, how much turbulence is produced and who is producing it and who is dissipating it, and all these things and that information you can use to improve your model produce new models so the data is useful right it as good as experimental data. Sometimes even better than experimental data. So, we will see like how you can do this, how you can do a direct numerical simulation, and what are the parameters that are required to call it a direct numerical simulation.

So, this I think I have shown much earlier the statement from Peter Bradshaw that basically, when people say they regret that we do not understand turbulence is in his opinion, what it means is we are just unable to integrate, numerically integrate the Navier-Stokes equations in our head. We are not able to calculate this, process it. So, we just need an extremely large supercomputer. In those times Cray was a very big supercomputer company. Today we have many producing big supercomputers.

So basically what he is implying is that the computing power is the limitation. This is the biggest disadvantage of the technique called direct numerical simulation we just don't have a supercomputing power in this world to actually calculate the flows that are of interest or flows that are have industrial you know, relevance that is the problem, so computing power limits the usage of DNS, and it's expensive so we'll see like why it is expensive and why computing power required is so large so then let's define what is direct numerical simulation okay so there are two words here direct numerical simulation so the direct here implies that turbulence is not modelled that I already told you in the beginning no turbulence models are used when we do DNS. So, that means no physical phenomena shall be modelled. For example, if you have turbulent combustion, you know a heat transfer, or if you can have hydrodynamic instabilities, thermal instabilities, you can have turbulent two-phase flows, whatever, whatever physical problem you have defined, it should be calculated using Navier-Stokes equations. It is a system of three equations, conservation of mass, momentum and energy, which is sufficient to solve any of your problems.

You must rely only on them. You should not do anything else other than using these three equations ok, so that's why the name direct came into existence there then the numerical simulation is not like numerically simulating any other equation. It has to be the Navier Stokes equations ok so And the numerical simulation of instantaneous that is unsteady three-dimensional Navier Stokes equations that means full Navier Stokes equations has to be solved numerically without dropping the unsteady term without dropping some direction sometimes you do like two-dimensional flows or one-dimensional flows this is not allowed that's because we already defined that turbulence is three-dimensional unsteady, right, and this also implies that DNS is applicable only for transition and turbulent flow problems that's because for a laminar flow, there is nothing like an indirect method indirect method implies turbulence modelling right suppose if you have a turbulent flow or a transitional flow you can go ahead and use Navier-Stokes to solve this or you can use some RANS models also to solve this whatever the accuracy is, but you can, you have an option. But when you are solving for a laminar flow, there is no indirect method. There is no modeling idea there.

And therefore, DNS, I mean, you cannot, I mean, you can call it, but it is recommended that DNS be used, the terminology for transition and turbulent flows. while you are solving full Navier-Stokes equations and the third criteria is that all scales of turbulence temporal and spatial has to be captured or resolved. that is captured by your time step and grid. This is where the previous lecture will come in handy for you where we discuss what is the smallest scale in a turbulent flow velocity scale, length scale, time scale, right? The Kolmogorov micro scales. So, as long as you are capturing those scales in your flow then this is your simulation will qualify to be called direct numerical simulation.

So, your grid that is $\Delta$ in x, y, z direction must be smaller than or equal to Kolmogorov length scale or an order or of the same order of magnitude also in some cases and your time step $\Delta t = \tau_\eta,$ the time step or the Kolmogorov time scale and this is a Kolmogorov length scale right. The length by time scale, so that means you need to capture all the way from an integral scale to a Kolmogorov scale, and we have seen from the Kolmogorov hypothesis what is the ratio between them suppose if this is your integral length scale $l$ and this is your $\eta$ we have seen the separation right so the $l/\eta$ from Kolmogorov hypothesis is $\left(\frac{1}{Re}\right)^{(3/4)}$ right sorry 1 by what is it $\eta/l.$ So, we have seen the scale separation between the smallest and the largest. So, one can go and estimate in a given problem how small $l$ or how small the $\eta$ will be for a given Reynolds number and your mesh size $\Delta x, \Delta y, \Delta z$ has to be $\eta$ right. So, one can make a simple calculation.

For example, if you take the Reynolds number, let us be let us take a velocity of 1 meter per second and a length scale of 1 meter 1 meter of a pipe flow or anything of that sort. And then let us say it is air, so it is roughly let us say$10^{(-6)}$, then this is easily 1 million Reynolds number. So, to compute a flow where the Reynolds number is 1 million, what would this be? So, $l$ is now 1, what would this $\eta$ be now? Can you calculate and tell me? So, what would this give me $\eta$ to be? The smallest mesh that you have to compute. If you have a calculator, you can, it will be in a few millimetres, micrometres, sorry, 32 micrometres, ok. So, you can see the size for a 1 meter, and this is the smallest mesh that you would need in this flow, and if it is 1 meter, that means this is the $\Delta$.

So, 1 divided by this, you need to mesh like this. Let us say if you do an equidistant mesh, each mesh will have approximately this size up to 1 meter in one direction and in the other two directions. If the length of your pipe is very long, you will have many such grids. So, you can see a lot of grid points are required basically.

Okay. So now other questions that we can ask is, are these direct numerical simulations? For example, can we call steady state Navier-Stokes simulations as DNS? No, because turbulence is always unsteady. You cannot drop the unsteady term in the Navier-Stokes and simulate and say this is DNS. It's not. And two-dimensional Navier-Stokes simulations, that means you drop one of the term. And you know what happens when you drop one of the direction not a term, direction.

What does happen when you drop one direction in a turbulent flow calculation? Redistribution rate, have you forgotten about it? If you have all the three directions are important. Even though you may see that the mean flow is only acting in one or two directions, it doesn't mean that you have to drop the other directions because turbulence is redistributing. also, you need to help the turbulent flow do that right. So, turbulence is always three dimensional. All flows, in my opinion, there can be exceptions.

I said this is a 2D turbulence is something, as I already discussed, is more or less hypothetical. So, most practical flows or flows that we see are three-dimensional. And also instantaneous three-dimensional Navier-Stokes simulation of laminar flow this I already told you for laminar flow, there is no indirect method, right? So, there is no indirect method here and therefore, in strict sense we usually do not call it DNS when we solve laminar flows using full Navier-Stokes equation. And also, if your grid size both temporal and spatial, temporal means $\Delta t,$ the time step here, spatial is $\Delta x, \Delta y, \Delta z$. If that is significantly larger than Kolmogorov micro scales then obviously you are not resolving all the eddies, all the scales of turbulence again you should not call it DNS.

So, these are the pointers when we call it DNS or at least if you are doing DNS you must be aware of this, that you must make sure that you are resolving all the scales use the full Navier-Stokes equation and to prove that it is DNS. Because this data will be used by others, especially those who are doing modelling, to validate right, they are going to say if their model is good or bad relying on your data, so you must do it correctly. So once you have the equation, then of course it's obviously at the end of the day it's a CFD technique so there are many numerical recipes there is no nothing like you have to use one particular technique, so all you have is this full Navier-Stokes equations and you can discretize using whatever technique this is not like I am recommending a particular technique here where you can one can use pseudo-spectral solution where you have Fourier ChebyChev polynomials finite volume is popular finite difference finite element any method that you feel is good you can go ahead to discretize and people are familiar with those who have some basic CFD knowledge are familiar with these three properties of numerical schemes that you choose that is scheme should be conservative, bounded and transporting ok. For example, finite volume schemes all the finite volume schemes are conservative by definition ok. We take a conservative form of governing equations there to solve.

So, finite volume schemes are by default conservative and the central differencing scheme is unique here because I am going to tell why I am mentioning only that. This is actually not a bounded scheme in the CFD; you would have learnt in your CFD course that CD schemes introduce oscillations and numerical oscillations, and they are unbounded. So, it is generally not recommended to use CD schemes. That is what you would have learnt in the CFD course. And CD schemes are also not transportive.

They do not reflect the flow of information, the directional dependency. Whether it is a convection driven, diffusion driven, it does not recognize that. So, for these two reasons, we generally say do not use CD scheme. But when you solve eddy-resolved techniques, direct numerical simulation, large eddy simulation and so on, it is recommended to use a central differencing scheme in DNS, LES, and other eddy-resolved techniques. In RANS techniques, you may not use it.

So why are we recommending CD scheme here despite it is giving numerical oscillation is first thing is, as I said, DNS grids are so tiny that numerical error becomes negligible, right? Your mesh is going like in you just saw like some 30 microns or so the mesh is so small. So, the error in a CFD calculation reduces with the mesh size smaller the mesh lesser the error irrespective of your numerical discretization scheme so this does not mean that the oscillation is becoming zero but the oscillation we want this oscillation to be there but minimal. That is the second point DNS requires numerical perturbations analogous to physical perturbations in nature. So, it's very easy to start turbulence in laboratory or in nature because the background flow has perturbations oscillations. It's not a clean environment in lab or in nature.

There is always some perturbation which is aiding the turbulence to kickstart. But when you do a CFD calculation, the environment is so clean, right? You are giving an initial condition where everything is quiescent, everything is uniform. Then how do you bring this chaos or randomness inside? So, of course the Navier-Stokes equations help, the nonlinear terms help, but you need to help the equation also. There is also a term, which is the viscous term, trying to get rid of these oscillations. So, an unbounded scheme like the CD scheme will be of use, and it is generally recommended to use it because it gives you small numerical perturbations; it is small because your mesh itself is very tiny.

And again, you can use any type of mesh. This is completely your free choice. Staggered grid, collocated grids, boundary fitted, body fitted grids or overset grids, whatever, cell centered, node centered, all the techniques, whatever is your wish list, that is not a problem here. Only the mesh size is the important factor. Not the methods and the schemes that you are using here, and this Poisson equation becomes critical when we do DNS because most of the computing power goes into calculating the pressure field.

And I'm talking about incompressible flows, so you need very strong iterative solvers or accelerators like this multi-grid or using this sip or successive over-relaxation. All these techniques, you need accelerators to make the flow converge, and this will be the most critical part in the computational perspective. What is the computing power required? The boundary conditions are easier here you are only solving for three velocity components one pressure, right? So, pressure is set by Neumann, velocities its Dirichlet conditions like you have no slip and kinematic boundary condition you do not have to worry about boundary condition is straightforward. And since the mesh size is very large, typically it goes into tens or hundreds of millions or even a billion. If the mesh size is so large, you would need lots of memory, probably several hundred GBs or even a terabyte of memory is required.

So with that kind of memory requirement, A single computer cannot do this. The memory on a single computer typically goes like maybe maximum 100 GB or so. You would need like terabytes of RAM memory, then obviously you need to parallelize the code, your solution. And MPI is the default useful technique, message passing interface.

MP can be done for small problems. Because MPI is required if you are, if you are having a very large problem ah parallelization is straight forward or not straight forward, I would say it is ah only viable technique is MPI for very large problems. Small problems which can fit in a one particular node with multi core architecture, you can do MP parallelization also. This is the important point, the grid resolution. So, what I already told you, you need to resolve all the scales of turbulence, temporal and spatial. So, if I am looking into a spatial grid here, let us call it delta.

So this delta can be $\Delta x, \Delta y\ or\ \Delta z$ in the three direction, the spatial grid and I am looking into an order of magnitude. So, I have two pointers here. To make sure that the mesh is good. One is, of course, the Kolmogorov length scale $\eta$, right? So, when you make a calculation, let us say you have chosen a mesh which is $100 \times 100 \times 100$ grid points in three directions, then you can compute the dissipation rate after the simulation is done. Using that, you can compute your Kolmogorov length scale in that simulation.

Then you can compare your grid $\Delta x, \Delta y, \Delta z$ with $\eta$. If it is not good, then you refine it to make a better grid. Maybe you do 1000 grid points in each direction, you become a 1000 cube grid, then you make a simulation from the simulation data, you compute dissipation rate $\varepsilon$, then you compute $\eta$, and then again you look into the ratio of your grid $\Delta/\eta$ to see whether it is $\Delta$ is smaller or equal to $\eta$. So, this is the direction you can go in when you do the calculation and you can use the Kolmogorov hypothesis to make your first guess of the mesh because you do not know what is that first simulation mesh that you would need right. For that, you have this ratio of the integral length scale to the Kolmogorov is coming from the theory you can use it, and if you use this $l$ by eta that we just discussed scales like $Re^{(3/4)}$ using the Kolmogorov hypothesis right.

So, this is your Kolmogorov hypothesis here using that. So, $l$ is, let us say, in one direction, $x$; what about $y$ and $z$? So, in each direction, if it is scaling like $Re^{(3/4)}$, it becomes $Re^{(9/4)}$. So, $N^3$ is the mesh in all three directions if you have a box, ok? Let us say your this is your numerical domain. So, where this is your all the three direction, let us say $Nx$, $Ny$ and let us call this $Nz$. So, this $N^3$ is nothing, but your $Nx, Ny, and\ Nz$ is your N cube here.

the total number of grid points that you would need not the length I am not talking about the length here n is the number of grid points along each direction. So, the $N^3$ will go like this according to the Kolmogorov hypothesis: very large for the same problem. Can you see what it is? The Reynolds number is 1 million, and what will be $N^3$? I mean, we just discussed. Let us say 1 million Reynolds number where u is 1 meters per second, l is 1 meter, air takes the air as the fluid. What is it coming? $N^3$, example $Re$ equal to $10^6$ will give.

What will be the $N^3$? Even if you square it is $10^{12}$, right, $10^{17}$. You can see the mesh 10 rise to 9 is billion a billion degrees of freedom ok, so you can compute here how many trillion, not even billion degrees of freedom is required and whether you can do that kind of a mesh. No, I have not seen any DNS calculation where Reynolds number is 1 million even for a flow where one meters per second and one meter pipe which is usually common people would have seen right in that kind of a scenario this kind of a flow there is no supercomputer on earth which can handle $10^{17}$ mesh points degrees of freedom so that is a limitation that we talked about in the beginning no supercomputer on earth is available to do high Reynolds number turbulent flows ok. So, Reynolds number is the limitation here. We can do only low Reynolds number turbulence using DNS, and some of the studies like this are again literature.

Some note that the actual dissipation it does not occur at the Kolmogorov length scale eta, but something like $15\eta$, but these are literature. These are of course special flows. So, in general it is better to look at delta compared to Kolmogorov length scale $\eta$. This is only result of a particular problem where you see that at $15\eta$ is where the dissipation is occurring.

So, that means $15\eta$ mesh is fine. But in general, I would use always $\eta$ to be less than oh, sorry, the delta to be less than $\eta$. Not $\eta$, $\Delta$ to be comparable to $\eta$, same order of magnitude. That's what I would use. Slightly larger is also fine, but not greater than the order of magnitude. It's better to have the mesh size and Kolmogorov length scale, same orders of magnitude.

Equal is good. The other pointer that one can use is viscous length scale that we discussed. Your $\nu/u^*$ that is used to make your $y^+$ and all these things. So, you are using this to get your inner scaling coordinate system or coordinates. So, you can also compare your mesh size with the viscous length scale $\nu/u^*$, provided you are working with wall turbulence. If you have jets, wakes, mixing layer like open flows then better to compare with the Kolmogorov length scale.

Only if you have a wall bounded flow a boundary layer flow or wherever wall turbulence is present then you can use this. viscous length scale $\nu/u^*$. I am just showing here some example. So, this is the plane Couette flow that I have been talking about plane Couette flow turbulent plane Couette flow. So, here I am looking into on the left side for this left side figure here, here I have what I am looking into here is that the y axis is representing max of $\Delta x, \Delta y, \Delta z$ this is the some of the best practice guidelines I am giving for DNS.

So, what is plotted here on the y axis is I am looking into the max of the size everywhere. So, you look at it is a 3D calculation, right? So, you look at max of this $\Delta x, \Delta y, \Delta z$ at every point and divide by the local Kolmogorov length scale at that particular point. Let us say you have 100 million mesh, then at each mesh point take the max of this. Of course, if you have equidistant grid in all three, then it is fine, but sometimes you stretch the mesh. So, maybe $\Delta x$ is larger than $\Delta y$ and smaller than $\Delta z$ and non-uniform mesh.

Then take the max of this because you are preparing for the worst case scenario. Take the max of this divided by $\eta$ and then here this is the wall normal direction basically. So I am just plotting here. So now you can see that it is not equal to $\eta$. The mesh size here the max of this it is going like 3 and a half 4 or something.

So this is also acceptable. So that means this is that means your $\Delta$ here. So this is the delta by $\eta$ right $\Delta/\eta$ is now of the same order of magnitude right. So, delta comparable, same order of magnitude, which is acceptable or there is another criteria called Grotzbach criterion here. This is also used.

So, where as you see here it is the cube root of this. So, here, $\Delta$ is defined differently y-axis $\Delta$ is essentially $(\Delta x, \Delta y, \Delta z)^{1/3}/\pi\eta$. So, this criterion is also popular to use, and when I plot that for the same problem again, this is the wall-normal direction. This is y by h, of course; this is in the y plus coordinate.

same problem. So I can see now it is less than 1. So this is good. So now the criterion has to be less than 1. Criterion value should be whatever you are computing here should be less than 1. So, which is good here? So it qualifies. If it is not then you refine the mesh and redo the calculation again.

That's what it is. So, this is when you are comparing with Kolmogorov length scale. For the same case, you can also do compare with viscous length scale because this is a wall bounded turbulent flow, right? It's a plane turbulent Couette flow. So, here I am comparing a large table where I am comparing here. Our own data with some reference articles and focus on these things. The last three columns $\Delta x^{+}, \Delta y^{+}, \Delta z^{+}$ that means your mesh in viscous scaling that is $\Delta x$ by your viscous scale $\nu/u^{*}$.

That is what is done here. So, $y^{+}$ is essentially your $\nu/u^{*}$. So, that y was the length that is replaced with the $\Delta x$. So, this has to be look that at every grid point it is not like at one location you see and it is qualifying as DNS. It is a 3D volume everywhere it should qualify the criterion. So, as you see this is our data there are reference data as well sometimes you see that the mesh can be much larger here usually it should be of the same order of magnitude like this.

So, it is less than 10, right? So, our $\Delta x$ is now the same order of magnitude as viscous length scale. So, it should not be 10 times larger. This is in the x direction, and this is in the $\Delta y$ is actually the spanwise, and this is the wall normal. So, it is the other way.

So, I let us call this y in our terminology and this is z ok. So, wall normal is you see it is much tinier, 0.41. So, this is the $y^{+}$ the first grid is set at 0.41. I have told you to put $y^{+}$ 1 at least right in RANS when you used a low Reynolds number formulation.

Here, I am using more I am having more grids because I need to compute the gradients also for the wall shear stress, right? So, I am having more points so that my wall shear stress data is better. I need to compute du dy on the first grid point. So, generally, we put many grids inside y plus 1 in DNS, maybe 5 grids, 10 grids as much as you can afford, and the largest grid along y is 1.71 of the viscous length scale at the centre of the channel.

So, it is very little stretching, slow stretching from 0.41 to 1.71 at the centre of the channel. All right, so,