Numerical Ship and Offshore Hydrodynamics
Prof. Ranadev Datta
Department of Ocean Engineering and Naval Architecture
Indian Institute of Technology, Kharagpur

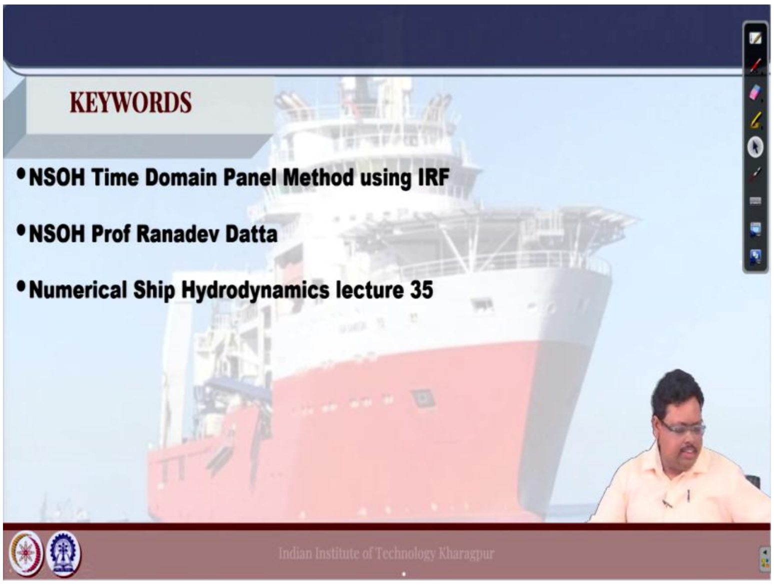
Lecture - 35
Numerical Computation of IRF Based Method (Contd.)

Hello. Welcome to my Numerical Ship and Offshore Hydrodynamics. Today is the lecture 35. So, today is the last lecture of this particular topic. So, today what we are going to show you that MATLAB code that actually I have prepared to show you that how I can get the response of the I mean of the vessel, ok.

(Refer Slide Time: 00:29)

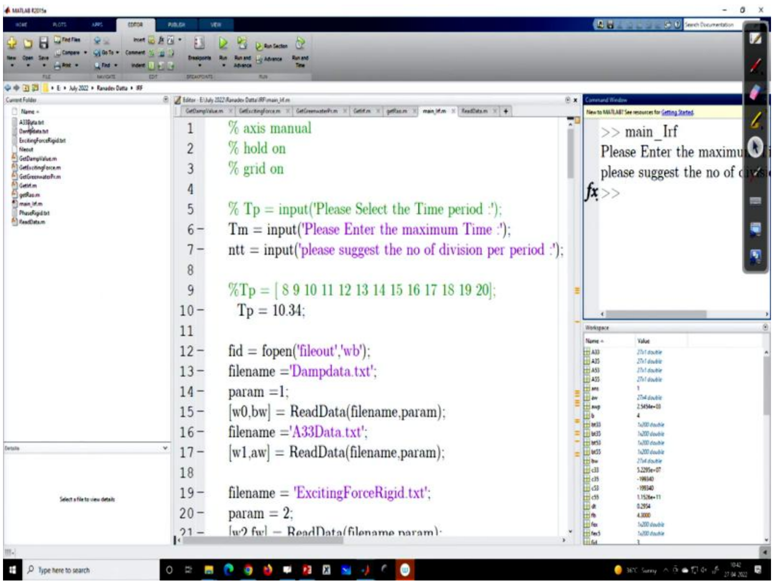
So, actually I got some input, it is a rectangular bar. So, I we have some frequency domain data for that. So, I am taking those data for my input parameters, and then we run the code and will show that how does that the b tau that function should look like, ok and then how the response will get through this. So, all this discussion we are going to have today, ok. And this is the keyword that we have to use to get this lecture, ok.

(Refer Slide Time: 01:15)



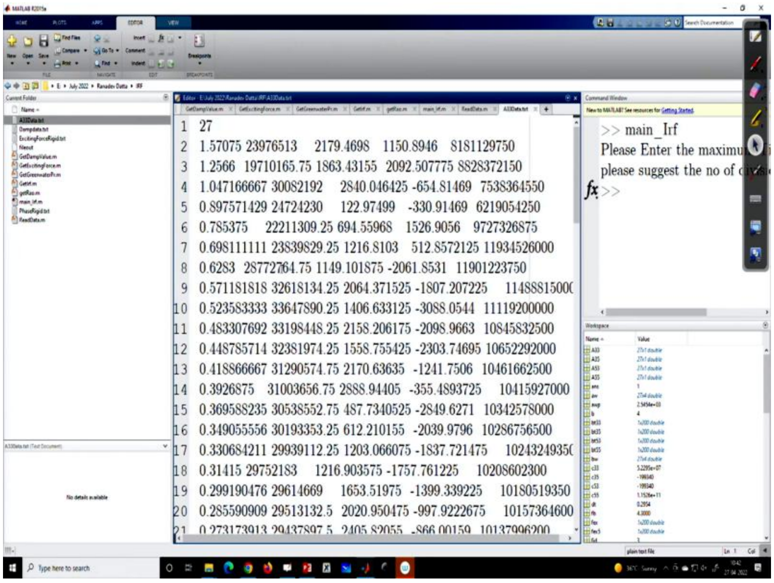
So, further, let us not waste any time and let us just jump into the code.

(Refer Slide Time: 01:28)



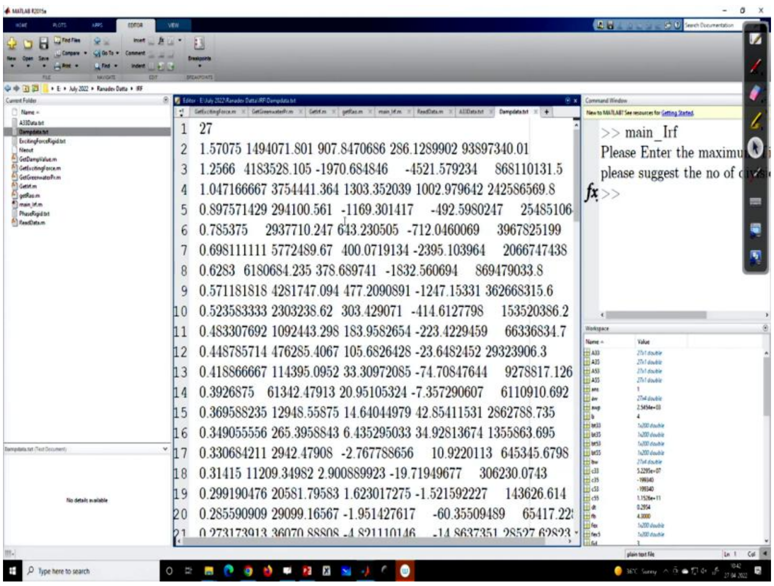
Now, here initially, here you can see that I am using as I said in last class that maximum time step what is the time step that I asked, and also please suggest the number of divisions per period. So, this is the value you need to enter, right when you run the code. Then, I have those data over here. Now, you can see this is the list of this added mass data.

(Refer Slide Time: 02:05)

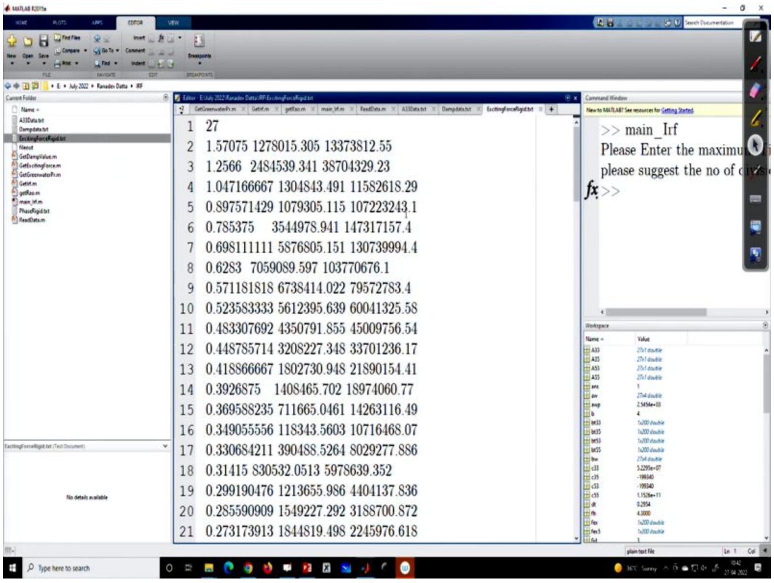


So, here this code is not single degrees of freedom, it is in fact, coupled heave and pitch data. So, we have the list of data for that.

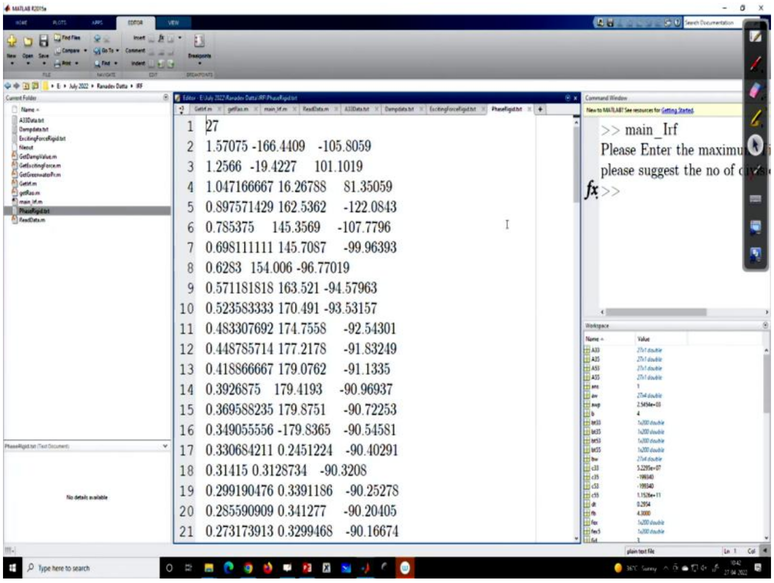
(Refer Slide Time: 02:26)



(Refer Slide Time: 02:32)

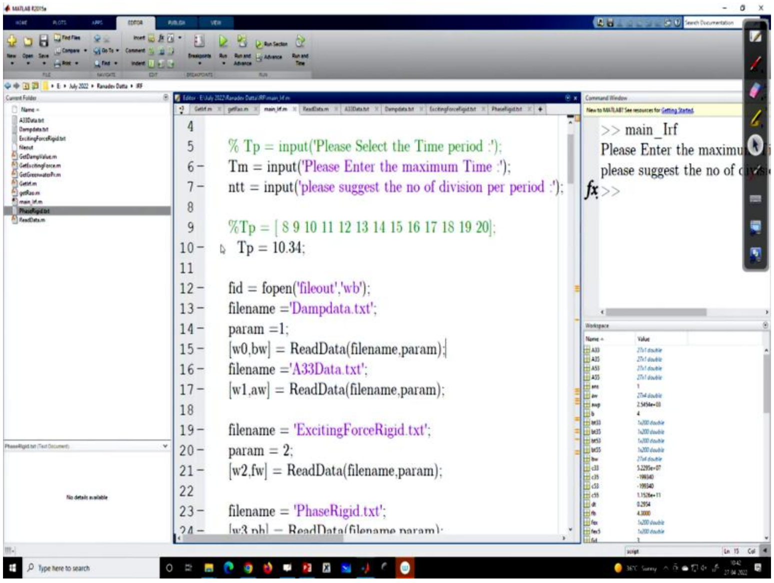


(Refer Slide Time: 02:39)



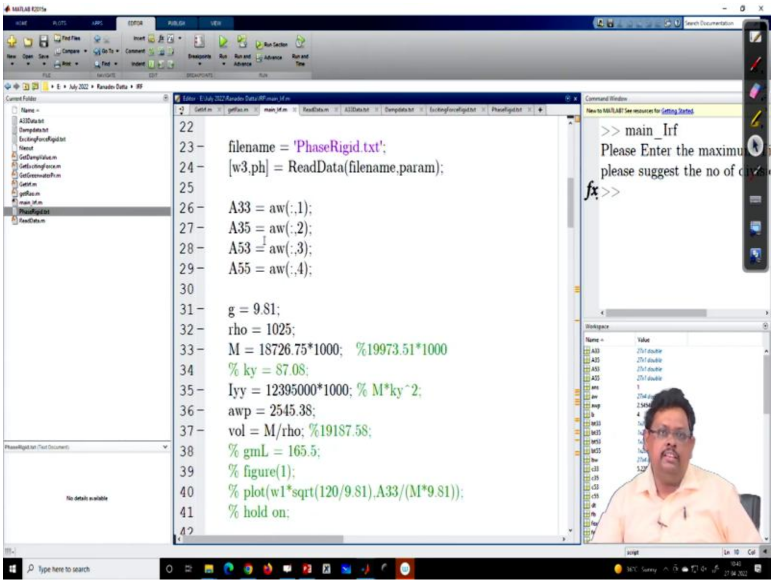
And also, we have this the damping data, ok. And then, you have the exciting force data, the frequency domain data, and also, we have the phase data. So, these are all are my input, right.

(Refer Slide Time: 02:49)



Now, here you can see that, of course, actually this also I should ask from the user, but. Here I am just writing it some Tp = 10.34 second. So, you can see here, it is the time period, I actually it is listed from 8 to 20 because we do not have much time to run all this frequency. If I run all this frequency then at the end we should get the RAO, right.

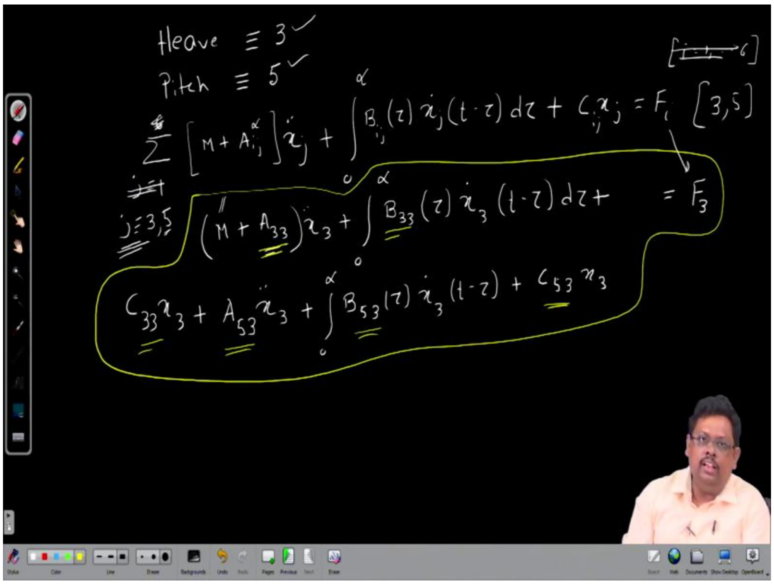
(Refer Slide Time: 03:27)



Now, here I just; so, initially is I am reading all this data that what is this damping data, the added mass data, the exciting force data, the phase, ok. And here I transferred into this $A33$, $A35$, $A53$, $A55$. Actually, you know if you solve this coupled equation of motion, then you need all these 4 data because the heave, we have to, here we are not using the single degree of freedom equation of motion, we are using coupled equation of motions, ok.

So, let us see that when I say the coupled equation of motion that what actually I am talking about.

(Refer Slide Time: 04:12)



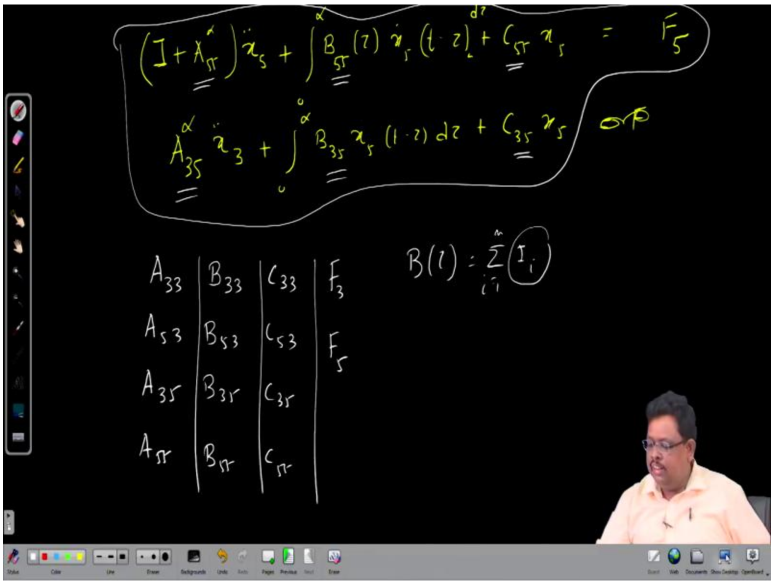
Now, here if you just because it is required to understand the code better way, here we have the coupled heave and pitch. So, we have heave and you can you make index the heave is 5, and then if you have the pitch, you can say the index of the pitch is 5. So, heave is 3 and pitch is 5. So, then how I can set the equation motion.

So, if you remember as I said it is $\sum_{j=1}^{6}\left[M+A_{ij}^{\infty}\right]\ddot{x}_j+\int_0^{\infty}B_{ij}(\tau)\dot{x}_j(t-\tau)+C_{ij}x_j=F_i$, when i also going for 1 to 6. So, this is my coupled equation of motion using IRF.

Now, here I am using only 3 and 5. So, if I put 3 and 5 how I can write it? Now, here in instead of j, this j having only two index which is 3 and 5, ok and then A 5 also having two index which is 3 and 5. So, let us solve for i equal to 3. So, the equation $\left(M+A_{33}\right)\ddot{x}_3+\int_0^{\infty}B_{33}(\tau)\dot{x}_3(t-\tau)d\tau+=F_3$.

Now, I am doing 5. So, 5 actually here I do not write 5. So, it is, but it acts with this here. So, I write plus $A_{53}\ddot{x}_3 + \int_0^\infty B_{53}(\tau)\dot{x}_3(t-\tau) + C_{53}x_3$. Now, this is the, so this actually this equation of motion corresponding to heave. Now, you can see to get this equation you need *A33*, you need *B33*, you need *C33*, *A53*, *B53*, *C53*.

(Refer Slide Time: 07:28)



Similarly, if I do the same thing with respect to pitch, right, so then it should be I, because I indicate the that moment of inertia what this Iyy.

$$(I + A_{55}^{\infty})\ddot{x}_5 + \int_0^\infty B_{55}(\tau)\dot{x}_5(t-\tau)d\tau + C_{55}x_5 = F.$$

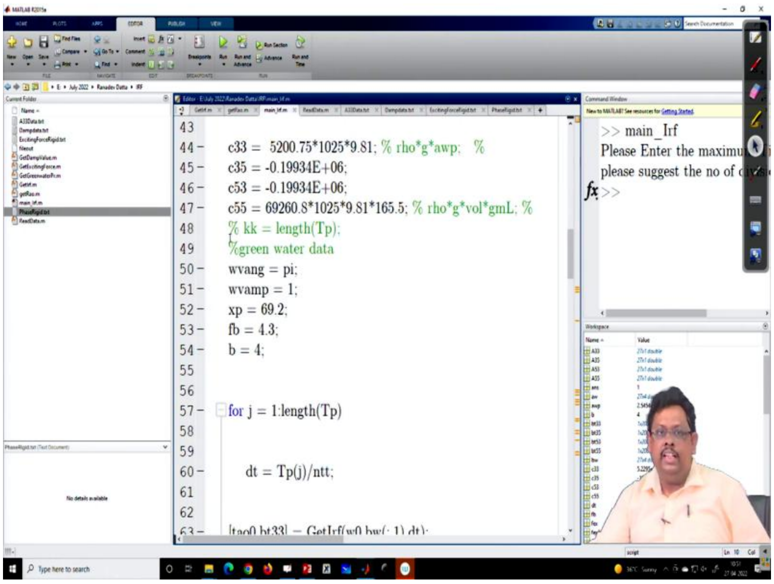
So, first I did the 5, right and then now I am doing the for the 3. So, it should

$$A_{35}^{\infty}\ddot{x}_3 + \int_0^\infty B_{55}(\tau)\dot{x}_5(t-\tau)d\tau + C_{35}x_5 = F_5.$$

Now, you can see if you want to solve this equation, if you want to solve this equation, there you have to have this *A55*, *B55*, *C55*, *A35*, *B35* and *C35*. So, then you have you need 4 data, one is *A33*, then *A53*, then *A35* and *A55*. And similarly for the B also you need *B33*, *B53*, *B35* and *B55*. Similarly, you need *C33*, *C53*, *C35* and *C55*. And similarly, you need F_3 and F_5. So, these are the data that you required from your vomit, right, ok.

Now, let us coming back over here. Now, here, now that is why that is why you can see here I have this 4 data *A33, A35, A53, A55* and similarly I also the B. Now, here I use g = *9.81*, ρ =*1025* kg per meter cube, *M* is the mass, it is $I \times 1000$ because it was earlier it was in tons, so I need to change to kg. So, all this is their water plane area I calculated. So, volume I get $\frac{m}{\rho}$.

(Refer Slide Time: 10:30)



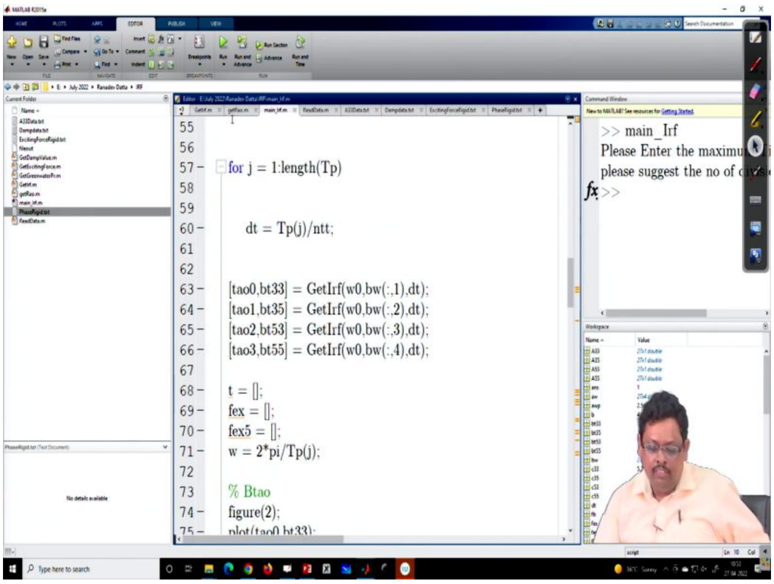
And then we have this the *C33*, and then this we have the *C35* as I said, we have *C53* as I said, and then we have *C55*. These 4 already it is my input. So, you can see here, and of course, is *Iyy* also it is also input, ok and water plane area, this volume, all these are the input values, right.

So, wave angle, heading angle, that I missed that time. So, it is π, so it is a head waves and also, I am using this wave amplitude is equal to 1. Now, this actually xp, fb, b, right now it is not required because this code is actually telling you that what is the advantage of writing this time domain code. That definitely we are going to discuss after this, end of this class here. This is required to you know taking the effect of the green water into the IRF code. So, that is basically the advantage.

In frequency domain code, everything is assumed to be harmonic. But you know the green water loading or the slamming, these are not the harmonic phenomena, right. So, you have to solve you have to have a time domain code where actually you can incorporate this green water effect or this slamming effect.

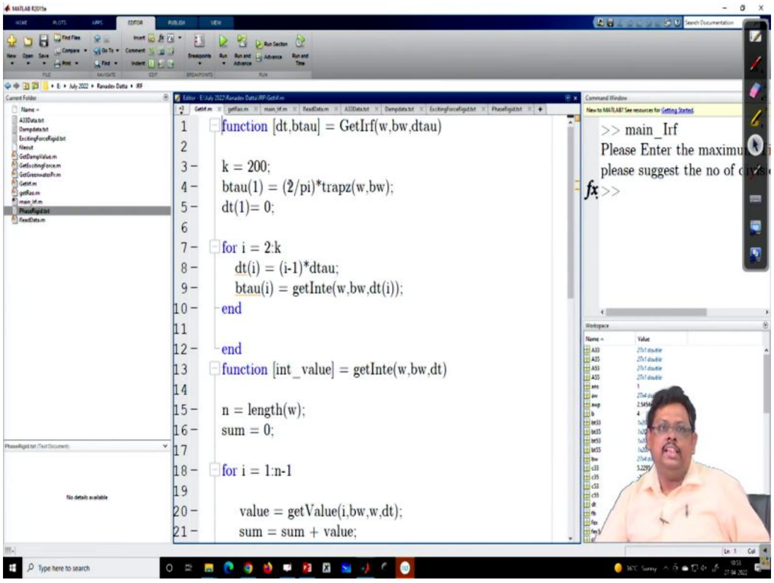
These things actually discuss at the end of this course. So, I am I do not want to discuss right now. But you know these 3 parameters required to you know estimate that also. So, here this code is written that including the motion and taking care the green water effect on the motion, ok.

(Refer Slide Time: 12:15)



Now, here you can see that this entity that I ask as the input, so I just select this dt as that. Now, here this is how I can calculate that, IRF, ok. So, I just called, it is a function. So, it gives you know this $B(\omega)$ data, I am giving, for each mode I am giving this Δt, so because that is required, right. And then I just calculate this, this $B(\tau)$, so $B(\tau)$ 33, $B(\tau)$ 35. So, these are my output.

(Refer Slide Time: 12:53)

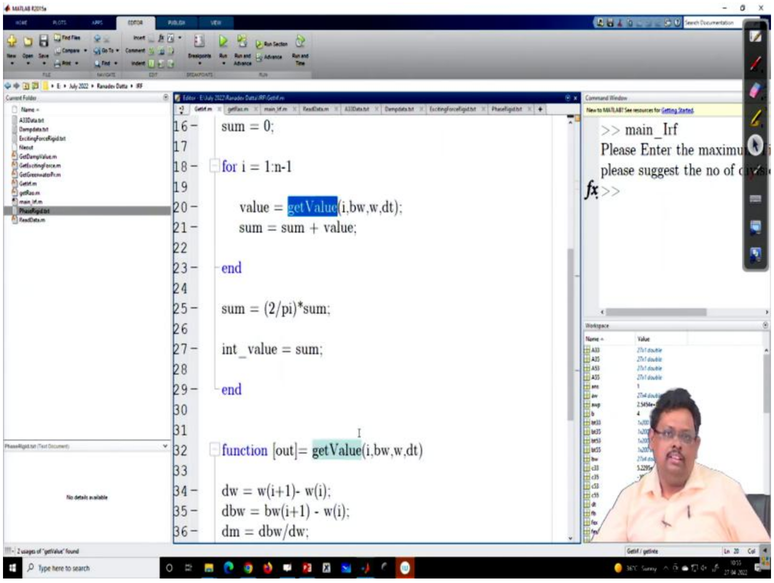


So, now if I look into this GetIrf this code, you can see at t equal to 0, I am using the trapezoidal rule. And then, here as I mentioned the previously when you write the code, this code should be written very neatly. I mean we do not need to write because if some unknown person will see your code, he should able to understand.

So, I really did not use any MATLAB feature as such here. I am writing everything from the scratch, so that you can understand better way because I hope that not MATLAB if you want to do in industry level writing. So, you have to write the code in C, C++, C# sharp in this language. So, better write in little bit structured way and neatly.

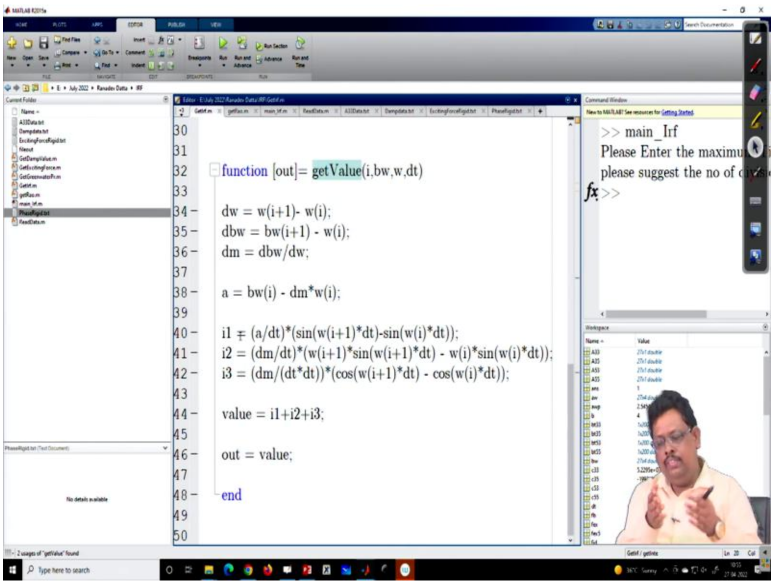
Now, here I actually if you look at this code, I really do not use very big code. I just used very small cluster. So, here you can see I call another function, this get integral, right. So, this function get integral is calculating this integral value at each time. Now, you see that this is get integral is nothing, but as I mentioned in your in our lecture which is just tell you like we have solved this $B(\tau)=\sum_{i=1}^{n} I_i$. So, this I_i is nothing but that function which is great, that is written over there, ok.

(Refer Slide Time: 14:41)



So, now let us see that what is written here. Now, you can see here also I again the value again I just passed another function, right. So, this value, this entire thing will do the process and finally, I multiply by that $2/pi$, right.

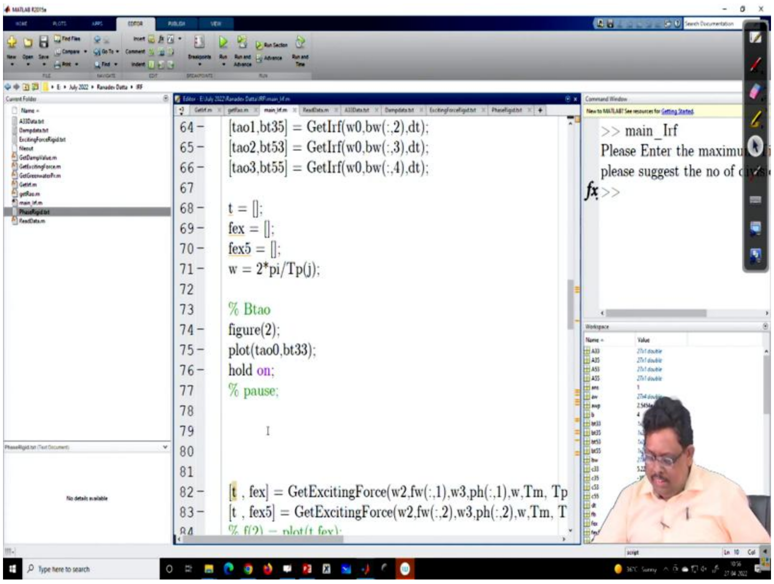
(Refer Slide Time: 14:57)



Now, here you can see this value I calculate here, right. So, you can see that $d(\omega)$ I calculate, then this $db(\omega)$ is nothing but the db and then this is the slope dm and this is the constant a. And then I have, you remember in my last class if you, last to last class I think I there you can have 3 integrals. So, here I just calculate 3 integrals, I just add it and just returned it, right.

So, you can see that writing this code, if you write in small part and then it is really if some unknown people looking at your code, really he will understand what actually you have done in your code. So, here I call all these $B(\tau)$ values and then this ω that is my that is my input because I just input here the time period, right. If you remember, my input is not frequency, my input was the time period here. So, I need to change that to frequency because it is a $\cos(\omega t)$ is the thing, right.

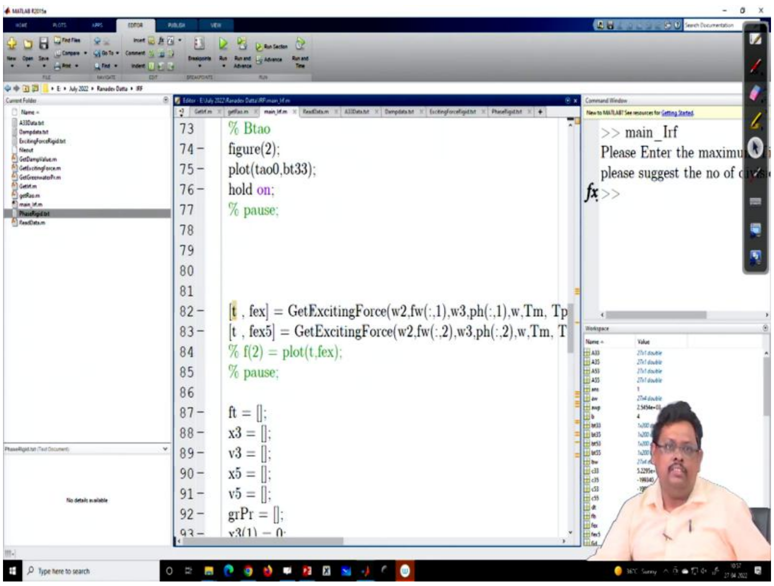
(Refer Slide Time: 16:18)



(Refer Slide Time: 16:28)

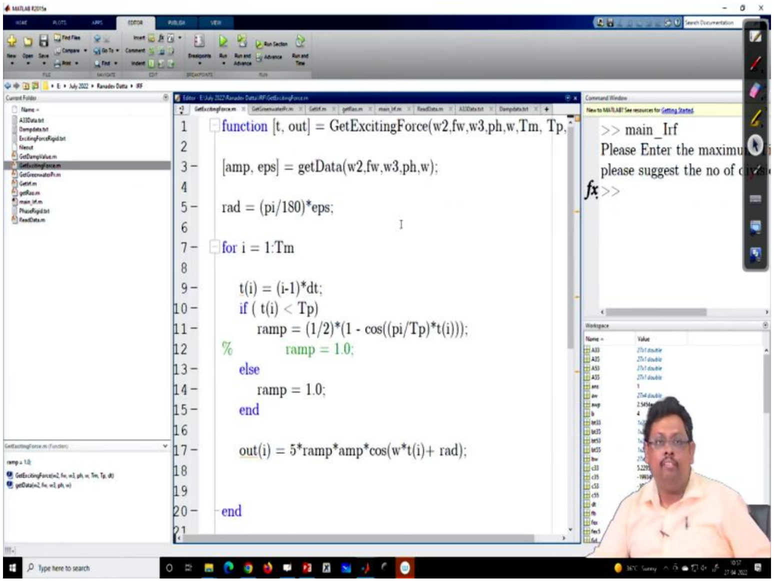
So, now here after, ok after making this omega, I just the, here I just give a plot of *B33*. So, here already I have plotted this value of *B33* if you look at it. Now, you can see that this plot it going to 0, right. Like as the time progress, it started with a very high values and then it starts oscillating and then it is actually going to 0 almost, right.

(Refer Slide Time: 16:48)



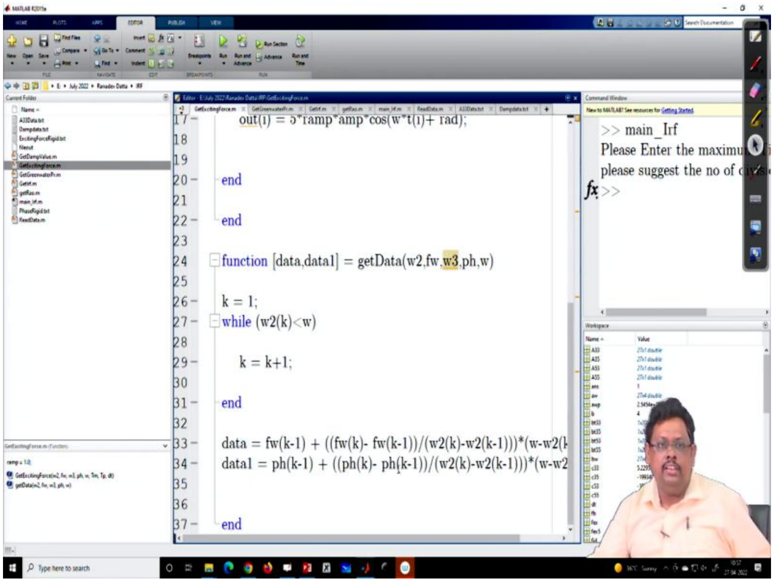
And then, here I just this use the external force, I just called another function get external force. So, then I have this code get, yes. So, let me show you that get exciting force.

(Refer Slide Time: 17:06)



So, this is the function. So, I just changes this the radian from the pi and then I have this data. And here you see that I am using the ramp function, right, is it not. So, and then I generate this ramp function over here, ok.

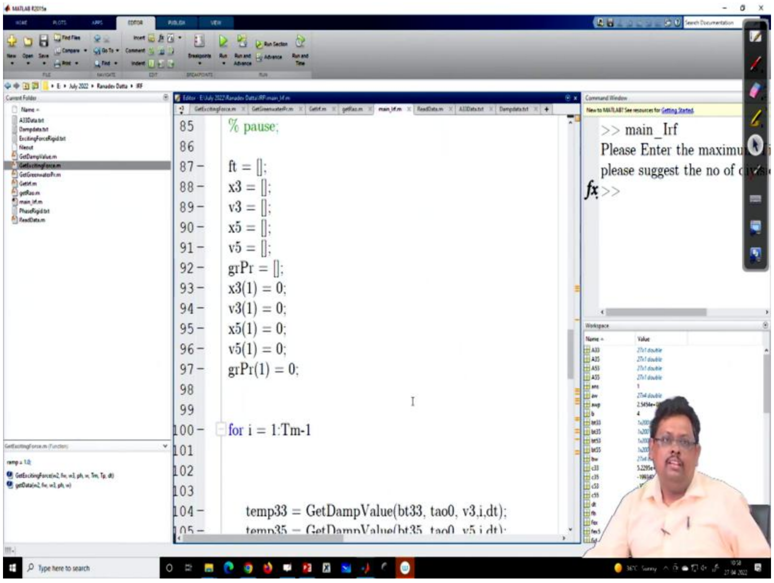
(Refer Slide Time: 17:46)



And then, this is the function, this gets, ok. So, I understand this get data that function, it will be actually try to find out that force, that the linear interpolation I am talking about, right. So, this is the linear interpolation, ok. So, this is how actually I give the exciting force.

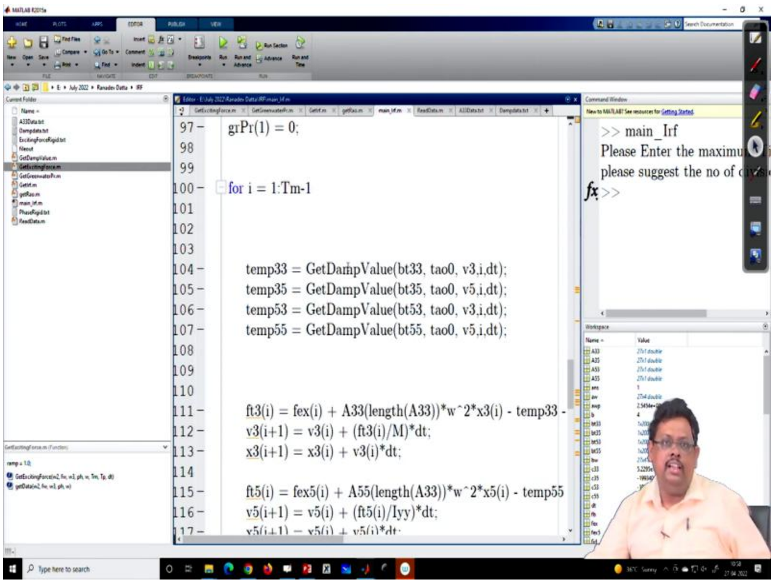
Now, again coming back to the main code. So, if you write the small function like this, so then people will understand actually what actually, how you are logically approached to the solution, ok.

(Refer Slide Time: 18:13)



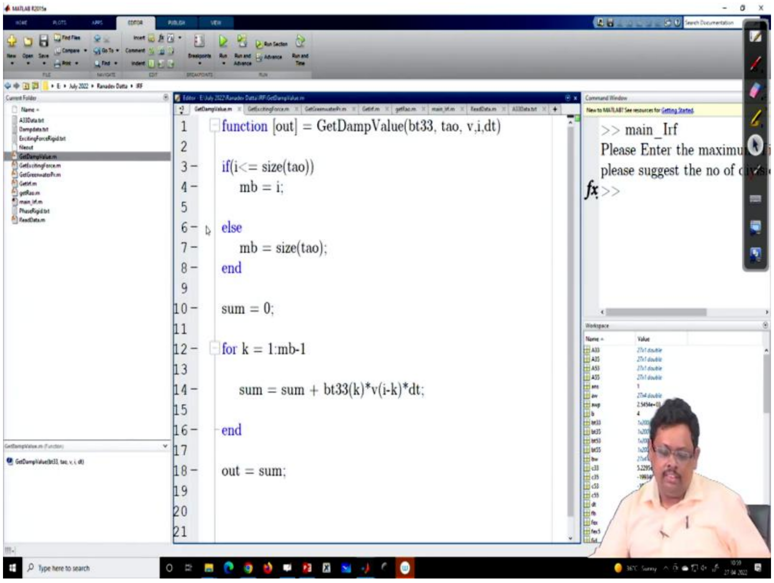
Now, these are something that the parameter I define because my ultimate aim to get x3, v3, x5 and v5, right.

(Refer Slide Time: 18:25)



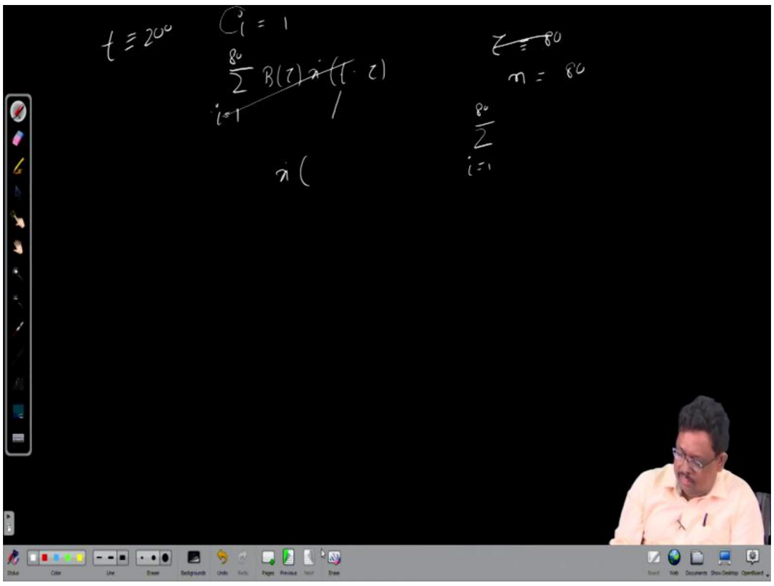
So, now here the time marching algorithm starts. And then, you can see here, I am asking for the get damp data. Get damp data is nothing but as I said the function. So, here I am as asking this b3 $b(\tau)$ values and then this v3 the velocity, and then present time say i, and also dt.

(Refer Slide Time: 18:55)



So, let us see that the get damp value here. Now, you see I just really do nothing. I just simply using the trapezoidal here, right. I did not do anything. But here is the catch is that size because you see here in your, if you able to you know just only one thing you need to remember with this code that you have to be little bit careful because your summation is here, yours.

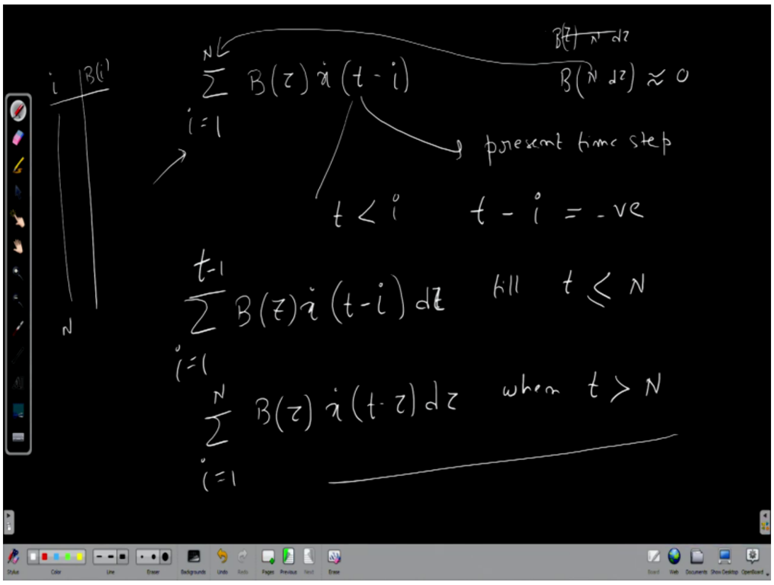
(Refer Slide Time: 19:21)



If you do the trapezoidal your summation is $B(\tau)\dot{x}(t-\tau)$. Now, τ is having let us say some 80-step let us take, I mean that n, n; that means, you know n is let us say your 80 second because your summation then runs from i =1 to 80. Because remember that $B(\tau)$ that time step your final time step it catches n = 80, let us see.

But now your t is let us say initially it is although, your total time frame t maybe you are doing for the 200 steps, but initially when your time marching algorithm starting from 1 in case of a MATLAB, it will 1. So, then x dot, if you now if you make here some i = 1 to 80 that is how to do the trapezoidal, ok. Let us do one thing, these two indices make it different, ok.

(Refer Slide Time: 20:40)



So, let us do one thing because this will confuse you. So, I am here in this particular function what I need to do? I do very simple thing I just add some i = 1 to some N. And this N, how I compute? This $N.\Delta t$ at this you know the $B(\tau)$, at you know B at that time $N.\Delta\tau$, at that time it is really approaches to 0.

So, in that way I can extract the value of N. So, I know that, that I need that this is my i versus B(i) and this actually I have this N data, so i =1 to N, ok. Now, I need to sum this, $B(\tau)\dot{x}$. Now, I just write t minus i, just make the two-index different. So, I am summing over i and this is my present time step. Now, if this present time step t < i, so then after some time (t – i) become negative, and then your program will collapse, right.

So, we have to take care this part that, so that actually you should do $\sum_{i=1}^{t-1} B(\tau)\dot{x}(t-i)dt$. This you need to do, right; till your $t \leq N$, right. And then once t>N that time actually you can do $\sum_{i=1}^{N} B(\tau)\dot{x}(t-\tau)d\tau$ when $t > N$. So, this is the only catch. This is the only catch nothing more, ok.

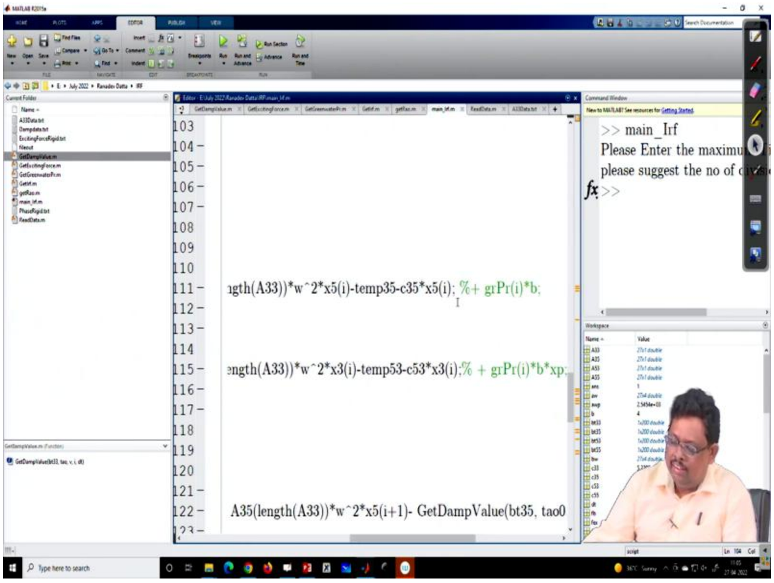
So, that is why if I look at here then you can see that; I have taken care of these things. So, mb is the size, this is the size of tau I find out, right. So, here I just do the sum. And if you look at it; this is let me, check actually that check may be before that. Yes, here if i is which is the present time step with the less than that size of size of tau, so mb becomes i. So, that means, here I just taken care that fact that what I told you.

This is what I told you here that it should be here, I just say that when it is less than N, so this should be 1 to $(t-1)$. So, that is what actually here also in this code, the same thing I am I have done. So, if $i \leq N$, then mb = i, otherwise mb equal to the size of the τ.

And then, I simply do this trapezoidal. It is called the one-point trapezoid rule. I did not do much. I did not use this conventional way. I just simply use this value b3 and then $(v_i - k)\Delta t$, ok. Very simple, very simple. I make it very simple, ok.

So, now again, let us go back now here, the main code here. Now here so, I have this damp data 33, 35, 53, 55, everything is there and after that I calculate the total f(t) as I said, right, ok.

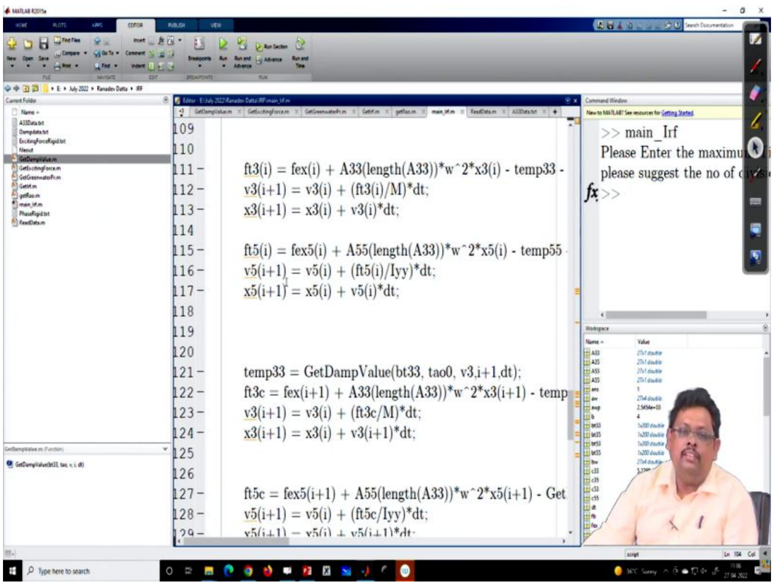
(Refer Slide Time: 25:17)



So, I just writing. So, now, here you can see that right now, actually I am this green water pressure thing that actually, I right now I make it inactive. But in this right-hand side that is the fun part of using this IRF code. Whatever the non-linear force you want to add always you need you can add at the right-hand side of the total force, ok. So, this is how actually one should write.

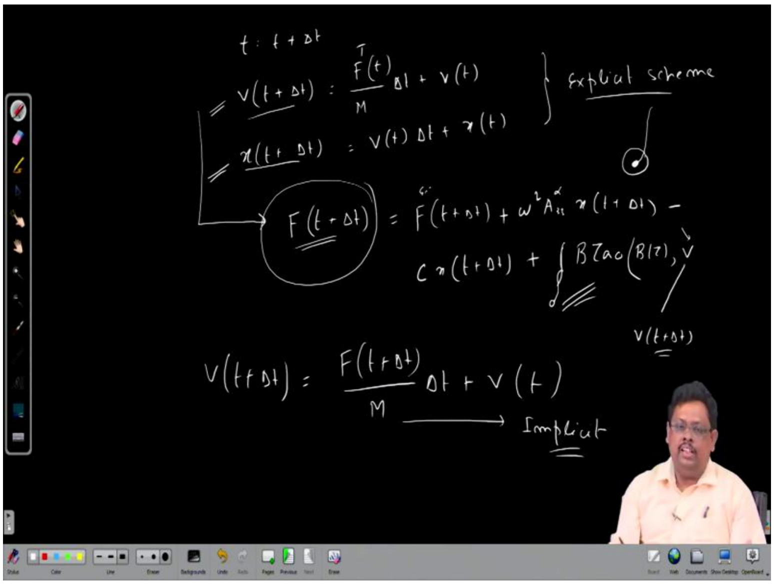
So, this is the main advantage of writing this IRF based code. You can add all the non-linear forces at the right-hand side.

(Refer Slide Time: 25:52)



So, then if it is a total force, then now, you can say next step it is the v3(i) by f by M d t. So, that you know is the Euler scheme. And then I did, I do that for the pitch mode also, right. Now, here what I do this, it is we discuss later on it is, actually it is one can say that predictor corrector scheme, right. So, here, let me explain it here. This is actually to make the code numerically more stable.

(Refer Slide Time: 26:36)



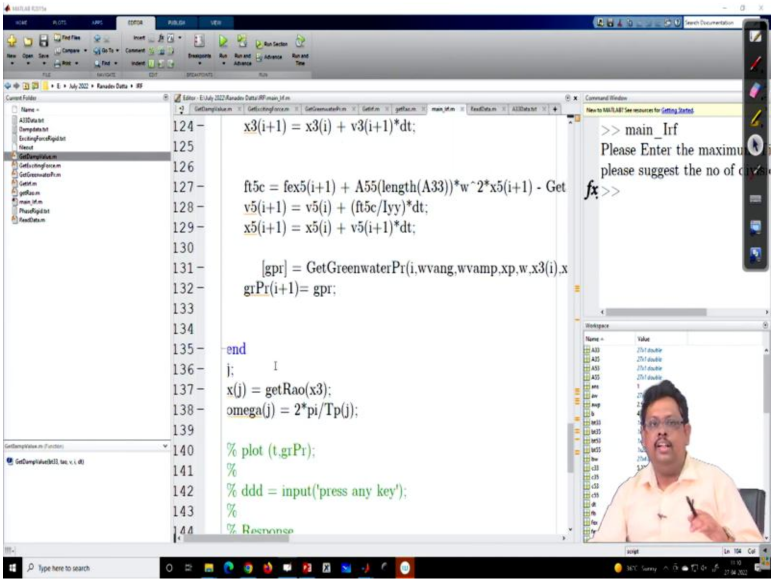
So, what we make that $t = t + \Delta t$ and then we get the $V(t+\Delta t) = \dfrac{F^T(t)}{M}\Delta t + V(t)$, right. And then I get actually $x(t+\Delta t) = V(t)\Delta t + x(t)$. So, this is how I am getting.

Now, this velocity at the next time step and then this displacement also in the next time step. So, with these two, again I recalculate my force at time. So, now, this is called the explicit scheme where actually I using this functional value at this point.

So, now, having getting this acceleration and then the velocity and the displacement in the next time step, I calculate the total force also in the next time step. So, everything I have, right, I have the value for $F(t-\Delta t) = F(t+\Delta t) + \omega^2 A^{\infty} x(t+\Delta t) - Cx(t+\Delta t) + \int_0^{\tau} B(\tau)$, here I am now I am I need to calculate this integration, it is for the next time step where actually everything is same, the only thing that this v now I am putting this v have the information of $V(t+\Delta t)$. So, therefore, this function only needs to change.

So, what I do is I calculate, I estimate the force in the next time step and now I am using $V(t+\Delta t)=\frac{F^T(t)}{M}\Delta t+V(t)$. So, this is now called the implicit scheme. So, then actually with making this small thing your code become little bit more stable, ok so, that is what I do over here.

(Refer Slide Time: 29:40)



if you use this; so, you can get here this ft, the next times I mean that is I calculate these two you can see that this (Refer Time: 29:48), now, again I send, I recalculate that total force in the next time step. And again, I am using the implicit scheme to get the velocity and the displacement, right. And then finally, again actually I again go back. So, here you can see this is how I can write the time marching algorithm, right.

So, again just very quickly let us go through again. Here it starts with (Tm -1) and then I calculate temp, I calculate the total force, the velocity, use the time marching algorithm to get the velocity displacement. Again, I calculate the force, next time steps, and use the implicit scheme to get it, and I calculate the green water because green water, ok let us not discuss this. And then I just plot this graph, ok.

(Refer Slide Time: 30:56)

So, let us see that how this graph looks like and how it comes, so I calculate here also. Now, you can see here. I can give very nice, very nice the response of the vessel. And it is very meaningful, it is you can see that it is very realistic, right, ok.

So, today, we have we have shown you the MATLAB code, and the numerical technique or that we actually I use to find out the response of the vessel. So, we stop this topic today. And from the next class we are going to start the Strip Theory.

Thank you.