Physics through Computational Thinking

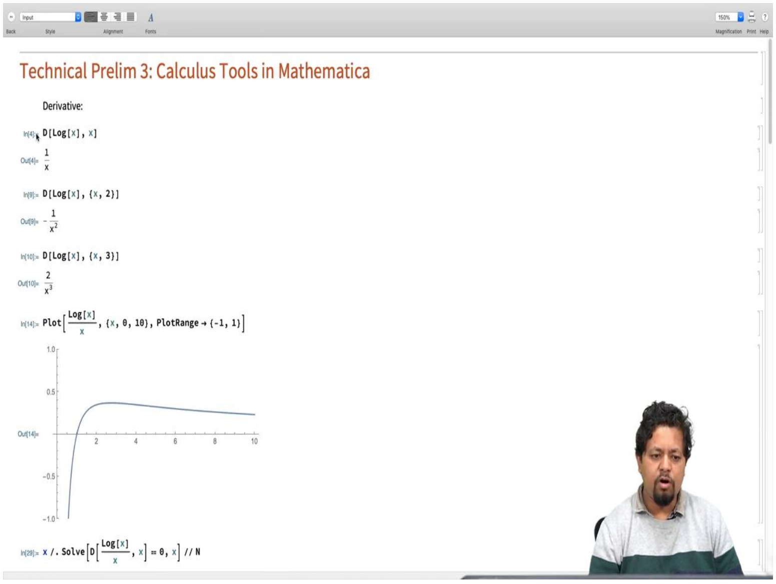
Dr. Auditya Sharma & Dr. Ambar Jain
Departments of Physics
Indian Institute of Science Education and Research, Bhopal

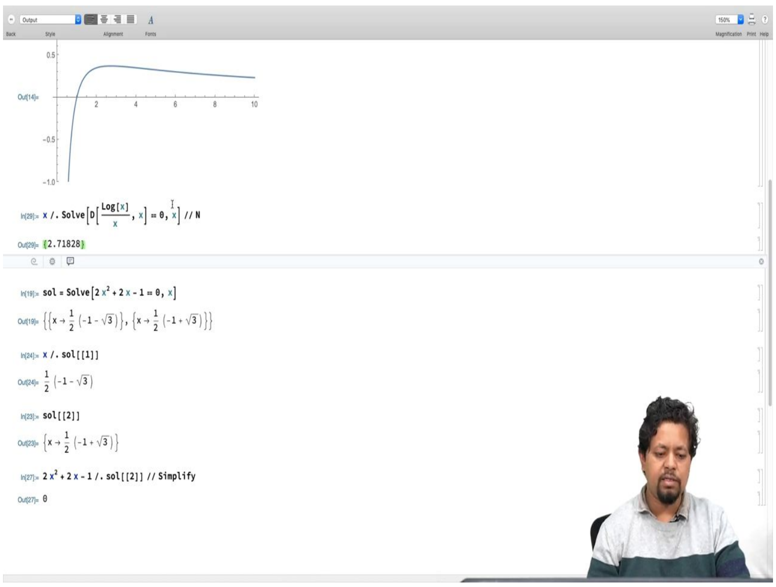
Lecture 20

Technical Prelim 3 – Introduction to Calculus Tools

Welcome back this is Technical Prelim 3. Today we are going to talk about Calculus tools in Mathematica. We are going to talk about how to take differentiation, how to do the integration and how to solve differential equations in Mathematica. So, let us go ahead and get started.

(Refer Slide Time: 0:44)





Let us say we are going to find the derivative of a function and let us say the function is $log(x)/x$, I can simply use the built-in function called D and write down my function which is the $log\,x$. Let us just start with $log\,x$. I want to differentiate this with respect to x, you know the answer, the answer is $1/x$. Let us go and execute this and this is going to give me $1/x$. So, D is the derivative function and you can try it out for other examples.

I can say $x\,log\,x$ and I want to do the first derivative of that. That is going to give me $1 + log\,x$. Let us go back to $log\,x$ for a moment and let us say we want to find, this was the first derivative. Let us say we want to find also its second derivative. For the second derivative, we have to simply add a flower bracket around x and say we want the second derivative and that is going to give me $-1/x^2$.

And if I put back 1 over here, I am going to get $1/x$. By default, if you just provide what variable you want to differentiate with, it is going to give you the first derivative. If you want nth derivative, you just provide the value if n over here. And that is going to give you $2/x^3$. If I put something that is not defined and just write n over there. It is just going to give me, it is going to apply some of the rules.

And, if it can solve it is going to give me still an answer, in this case it says that n>1, then I get this particular object and otherwise it is just going to give me $log\,x$ back. So, you have to use this with a little bit of care. So, let us go ahead and change this back to 2 and we can go ahead and try the third derivative and so on. Let us go back and say we want to find out the first derivative of $log(x)/x$, so let us say D of $log(x)/x$.

This is the first derivative, so I am going to write that and I am going to execute and I get $1 + 1/x^2$- $log(x)/x^2$. Let us say we want to find where does this function $log(x)/x$ has a minima. Just go ahead and plot it first. Let me go ahead and plot $log(x)/x$, x goes from 0 to 4, let us say and we get some function like that. Okay, let us go ahead and make it slightly larger range, maybe 10 and get something like that.

I think it is not printing the negative range so let me put the plot range from -1 to 1 and there we go. So, somewhere this has got maxima or minima. It has got maxima somewhere around here and I want to find out what is that, about what is exactly that number. So to do that, I have to take derivative x and set it to 0, derivative of the function and set it to 0. So, the function is the $log(x)/x$, so I take the derivative of that. I get this quantity and I will set this to 0 and then I have to solve for it.

So, to do that what we will do is, we will set it to 0 by using equal function or double equal to. Double equal to is a comparison. It says that we are going to set it to 0 and that just gives me an equation. What I want to do is: I want to solve for this equation, for that I am going to use the function called Solve. Solve is again a built-in function to solve, for example, polynomial equations or finding zeros or solving for the sort of equations.

So, let us go ahead and do this and I want to, I need to specify what I want to solve for, I want to solve for x over here and when I do that and it gives me a solution that x is e. Notice that this solution is in the form of a replacement rule. The first flower bracket says what is the set of solution and second flower bracket is for more than one solutions.

Let me give you another example of Solve to make this clear and in this case, we can simply take a quadratic equation. Let us take $2x^2+2x-1$ and will say, solve for x. Sorry, you have to specify this as an equation. So, remember you have to put a double equal to, this single equal to will run into some sort of error. So let us go and execute that and then it gives you two solutions.

Each sort of solution is going to give you, one set of will be in one set of flower brackets and the other solution will be in the other set of flower brackets, and all this, both these solutions are put together in the most, outermost flower bracket. If I want to particularly extract a solution, all I have to do is, let me assign this to a variable, let me call that variable sol, execute it and then, when I say sol[[1]], I get the first solution.

Notice that I am using the double square bracket here because I want to read the first element of this list. I want to extract the first element of this list which is this guy and for that, I am going to use double square bracket. Single square brackets are used for arguments, if I do sol[[1]], I will just get solution sol itself and will just up end 1 after that because it does not know what to do this one.

But with double square brackets, it is going to extract me the first element of that list or that array and that is the first solution and I can do the same thing for the second solution. How to extract exactly this piece out? Well you can copy and paste but if you are writing code, you want to extract this out through a code and to do that what we do is, we have already got this part, we simply say $x/.$ sol[[1]] and that will give me the solution.

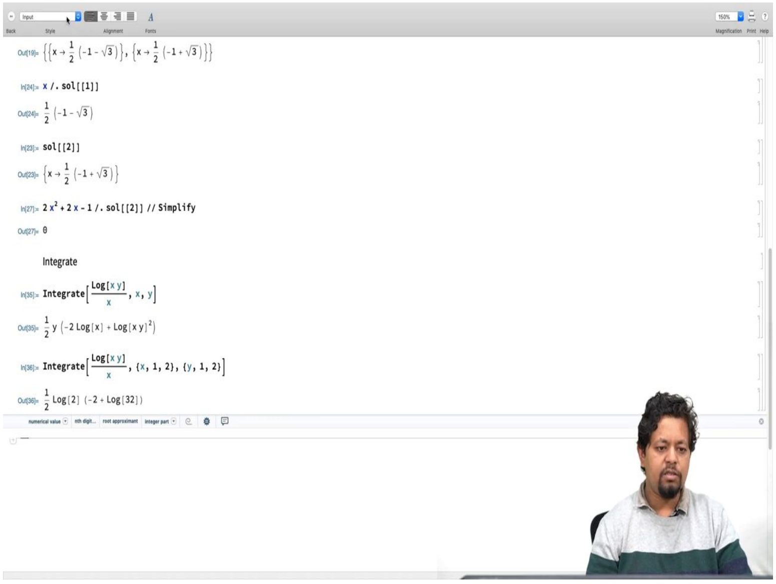
Notice that sol[[1]] or sol[[2]], these are replacement rules and I can apply these replacement rules to any expression that I have. In this case, my expression was simply x and when I apply that it just substituted x by the value that sol[[1]] was providing so I just got that but you may be interested, for example in getting back your equation and you want your back quadratic polynomial.

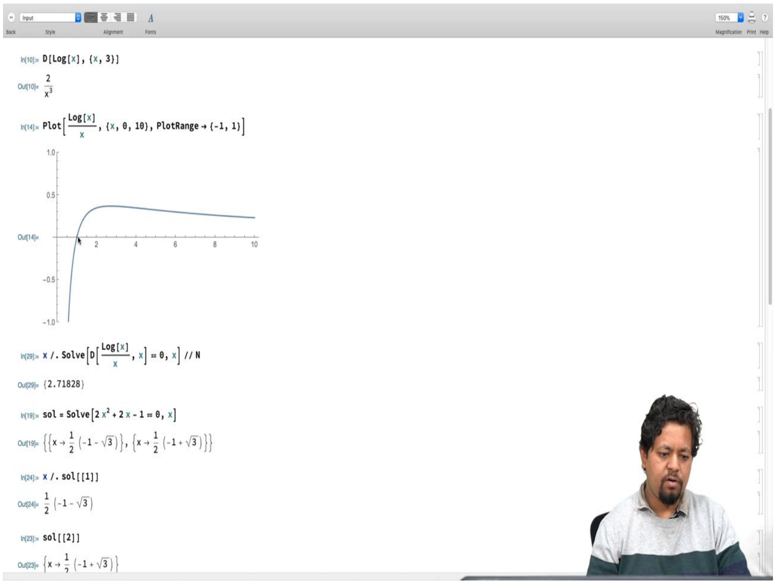
And, you want to say I want to evaluate this polynomial at the first root and what you expect, you expect to get a 0, so I can say I am going to apply this at the first root and it gives me some expression. Let me go and expand it out by doing simplify and that gives me 0. This was expected because this was the root of the equation and I can try the same thing with sol[[2]] and that also gives me 0.

So, this is how you extract the solution, solutions are in the form of replacement rules. I can extract a specific solution; apply that to any expression that I want to check what result they get. Coming back to this equation again, what we will do here first we construct the equation by taking the derivative of $log(x)/x$ equated to 0 and once we have got that we ask to solve for x and that gives us the result x is replaced by e. e is simply the Euler number close to 2.718.

If you want to check it out, I can go ahead and in the same line of code I can simply say x slash dot that I am going to execute that and it is going to give me e, slash and after that I am going to get 2.71828. Slash slash is post-fixed operator, it says that everything that is preceding to this, apply the numerical function and that gives me the number 2.718 which is nothing but the Euler number. So, this was a quick look at finding out derivatives using D and solving, finding out the minima's, maxima's and extrema's using the solve function.

(Refer Slide Time: 9:31)





We can also use; we can also do integration using Mathematica and I am going to demonstrate to you now how to do that. There are two ways to do integration, one is analytical way of doing integrations, so for that we have got built-in function called Integrate and if I provide it a function, Let us go and provide it $log(x)/x$ and we will say integrate with respect to x, it is going to give me a result which is what we usually known as the integration.

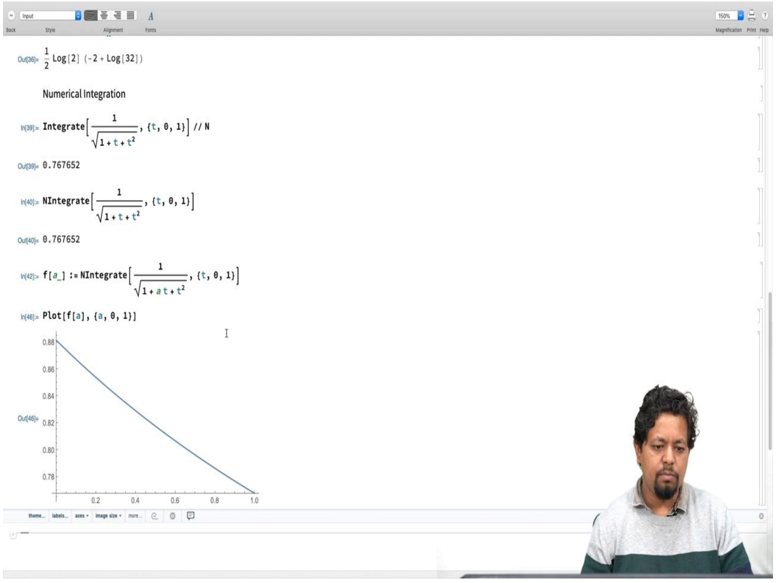
The result is $log(x^2)/2$ and up to a constant. So, this gives me a result up to a constant and if I want, I can do a different integral also by supplying a range for x, let me do this in a separate line. I can supply a range for x by saying as always x goes from; let us say 1 to 2. And the answer I get is will be $log(2^2)/2$. You can validate it to analytical methods that this is actually the right result.

Let us go ahead and do it from 0 to 2, to see what happens. It complains, it says that integral $log(x)/x$ does not converge on 0 to 2 and that is true if you look at the plot over here for $x < 1$. This is negative, and is diverging and if you try to do the integration 0 to 1 range, it is not going to give you anything finite and that is what this integral complains about so let us go back and set this limit from 1 to 2.

So, this is doing analytical integrals. Indefinite analytical integrals, this is definite analytical integrals and you can also do this for something else like y over here. It does not have to be, so the specification that we have provided over here, we are telling integrate that integrate with respect to x, so for that as far as this integral is concerned y is just a constant, I can go ahead and execute this and I am going to get, y is going to be treated as a constant.

And I simply get the ½ $log(x\,y)^2$. If I want to integrate this to y also. You can simply add y as another variable and go ahead and calculate that and you get an indefinite integral for the $log(x\,y)/x$, integral with respect to x and y and you can do this also with definite integrals. So, I can go ahead and put another range for y also. y from 1 to 2 and there we go, it will be a definite integral by substituting the $log(x)$. So, this is integrate, similarly for differentiation also you can provide other terms or other parameters in them and they all will be treated as simply as constants.

(Refer Slide Time: 12:34)



Let us go ahead and look at numerical integration, in cases when you cannot evaluate, cannot evaluate analytical integral. You can do numerical integration. Let us go ahead and try a function where we cannot get a numerical integral. Let us say 1 plus analytical integral, so let us say 1 plus t plus t square, I am not sure if there is an analytical integral for this, but let us find out, if Mathematica has a solution, analytical solution for this, it does.

So let us go ahead and we find another one. Let us just try with this, so this is an analytical integral, let us go ahead and give it a range from t goes from 0 to 1 and this gives me a definite integral and I can actually go ahead and come to its numerical value. To see what I get, that is the numerical result. Let us go ahead and try this with a built-in Mathematica, built-in numerical integration method that is NIntegrate, that is NIntegrate.

And let us provide the same set of arguments to NIntegrate and we are going to simply execute that and wonderfully we get the same result, not surprisingly, of course, this was a simple function, we knew the analytical answer for this result. Thanks to Mathematica's built-in library of functions, it gave us analytical result for integration of this function and Nintegrate also produced the same result.

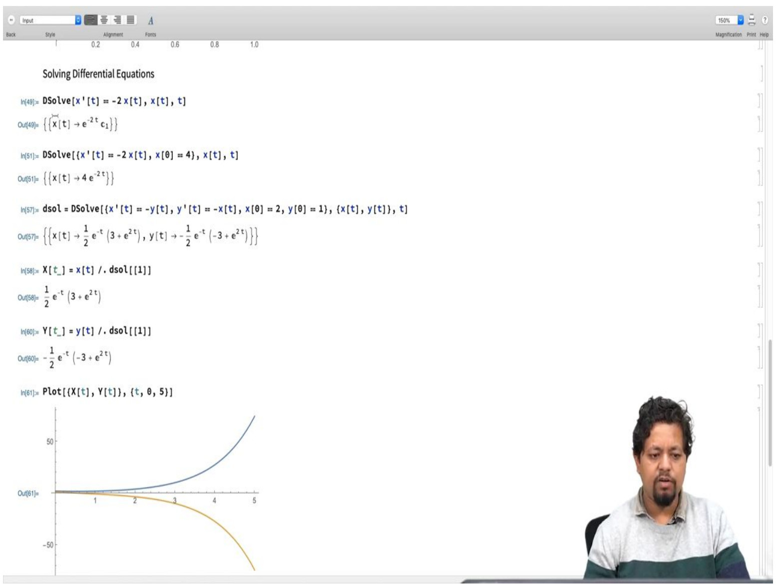
I would recommend that you explore both integrate and NIntegrate, look at the options, look at the documentation, look at in what various ways you can use Nintegrate. You can also do NIntegrate into multiple variables; Let us also explore that really quickly here. I am going to add something at that sort, now actually this is $1 + a\,t + t^2$, if I try to call NIntegrate with that, it is not going to work because NIntegrate does not know what is the value of a and it is not a numerical value, so it cannot apply a numerical integration.

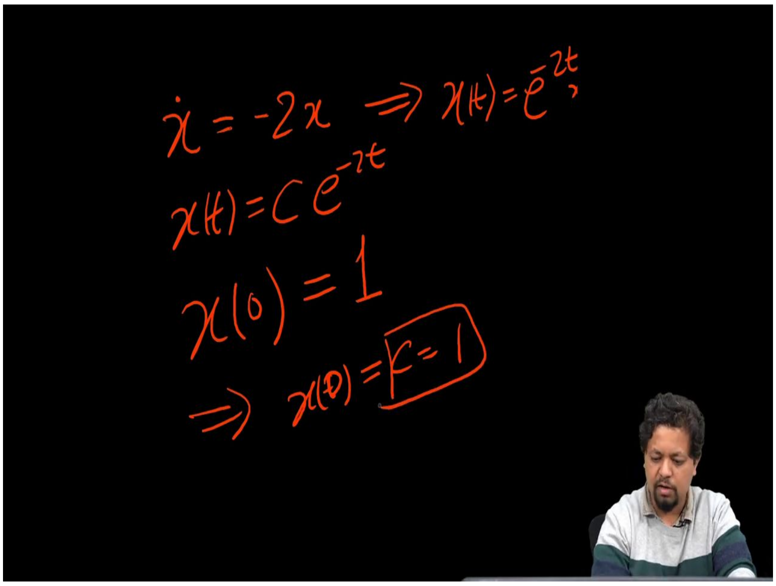
So, it is going to complain, the integrand given by this expression has evaluated to non-numerical values for sampling points in the region with boundaries 0, 1. So, it says that in your specified range of variable t wherein 0, 1, this integrand does not evaluate to any numerical value. So, therefore it cannot actually do numerical integration. So, therefore we have to provide a value for a.

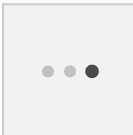
If I want to do something on that sort, the best way to do that is to define a function $f[a]$ with a colon equal to which is the set delayed operator. It says that I am defining this function but I am not going to evaluate it right now. I will evaluate it at the later stage, so I am going to go ahead and define that, now I can call f with some value, Let us say $f[1]$ and the answer is 0.767, now you can go ahead and try out $f[2]$ and $f[3]$ and so on.

If I wanted to make a plot of f of a with respect to, with a going from really small values 0 to 1. You can make the plot and what you see here is that it has the function that decreases, so this numerical integration of the function of a drops down. Very good, so this is numerical integration, let us go ahead and move on to solving differential equations for that I am going to solve the differential equations.

(Refer Slide Time: 17:00)





I am going to use one more inbuilt function called DSolve. DSolve stands for solving differential equations. D is for differential and solve was used for solving polynomial equations or linear equations or functions. DSolve is for solving differential equations, so let us see how this works. Let us say we are solving the equation . Let us go ahead and solve that equation, so to do that, it is understood that $\dot{x} - 2x$, derivative with respect to some variable. Let us say it is t in this case or its time.

So, I am going to say $x'(t)$ which means first derivative of x with respect to t and I am going to set that equal to, I will again use double equal to symbol and I will set that equal to $-2\,x$ and I am going to say give me a solution with respect to x. Oh, this is incorrect, the reason for this is incorrect is that I have not told that what is this x, is this x a function or is this x a symbol.

What exactly it is? I am saying the complaint, let me show you that, it says that DSolve called with 2 arguments, 3 or more are expected. So what went wrong? Well, we should have said, we should solve this differential equation with respect to what variable. So, that variable is time t, so let us go and execute that now, this is still going to be wrong because it says the function x appears with no arguments.

So, here is x appearing with the function, it depicts that x is a function because I have used $x'(t)$. So, it does not like that, you have to provide the argument t saying that x is actually a function of t. Let us go ahead and execute that and there we go again, we get the solution. The solution is again in the form of a set of replacement rules. This is one replacement rule and since there is only one solution. I only get 1 replacement rule which is $x(t) = e^{-2t}$ times, unknown constant c1.

We did not provide the boundary conditions, so it figured out there was some integration constant and that integration constant is that c1. Now this is an exponential decaying solution, we know that this is the exponential decaying solution and this is indeed the correct solution. You can go ahead and check it out analytically yourself. What we will do now is, we will go ahead and make it a boundary value problem.

To do that what we will do is, rather than providing 1 equation, we will provide a set of equations. Now, ask it to solve all together and boundary value conditions we are going to add as again equations. So, let us fix the boundary condition here. Let us say this is $\dot{x} = -2x$ and exponential decaying solution is x as a function of t is e^{-2t}, times a constant.

And I want $x(t)$ is some constant times e^{-2t}, in order to find the constant I need some boundary conditions, let us say at $x(t = 0) = 1$ and then we can immediately follow from this that $x(t)$, $x(0) = 1$ and $x(0) = c$ which is 1 and therefore we establish that $c = 1$. Let us go ahead and do this automatically with DSolve. So, we are going to go ahead and say $x(0)$.

Remember that we are providing a boundary condition, so we say $x(0) = 1$ and it gives me, e^{-2t}, it has found out that c has to be 1. You can try by changing it, by make it 4, is going to give me the constant as 4. So, DSolve is quite capable of solving quite a few differential equations. In fact, it can solve a large first order differential equations. So, let us go ahead and experiment a little bit more with this.

We will use this for, let us see if we can, let us use this for solving a set of coupled differential equations, let us go ahead and use $\dot{x}(t)$ is $-y(t)$ and we will say $y'(t) = -x(t)$ and just remove this boundary condition for now. Let us go ahead and say that we want to solve for $x(t)$ and $y(t)$ with respect to t, there we go, we execute it and we get a solution and there is again 1 solution but only 1 solution is there.

But again this is a set of 2 equations, one for, or two replacements rules one for x and one for y. Let me go ahead and give this a name DSolve equal to this, so let us go ahead and see this is $x(t)$, solution for $x(t)$ is that the two integration constants c1 and c2 because there were 2 sets of first order differential equations. Again, there are constant c1 and c2 also appear for $y(t)$. There is the solution for c1 and c2.

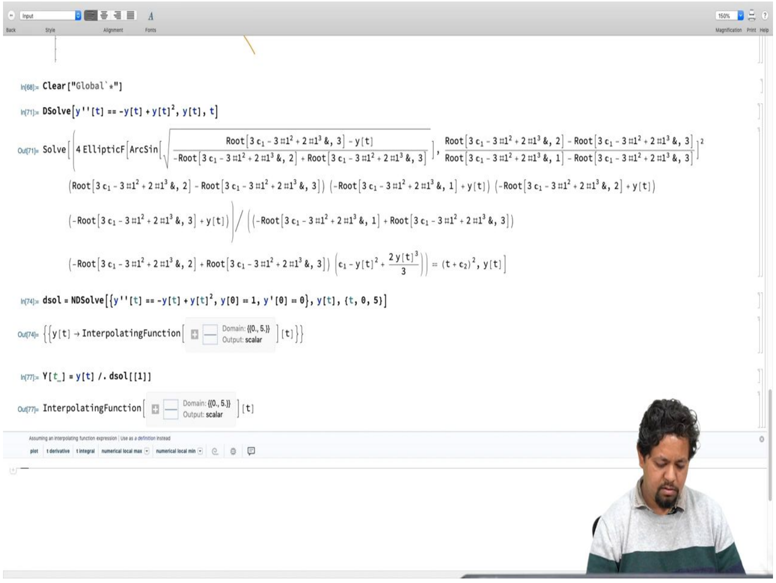
And if you want to extract these solutions, we have to again say dsol and we have to say first, give me the first item which is just the first set of solution that we have and then, we can say $x(t)$/. dsol and that gives me this. Further, if I want I can go ahead and say, you know $X(t)$ is equal to that and that will make a definition as $X(t)$ that. Let us go ahead and supply a boundary condition.

So you can supply a boundary condition here by saying $x(0) = 2$ and $y(0) = 1$. Let us go ahead and see what happens with that. There we go and that is my dsol. I will extract $x(t)$ from that and this time I am going to get $x(t)$ is that, I got it over here. I have defined as this $X(t)$. Now, going to plot $X(t)$ and to do that plot, I simply mentioned t goes from, we solve it from 0 to 5 and let us see what the plot looks like.

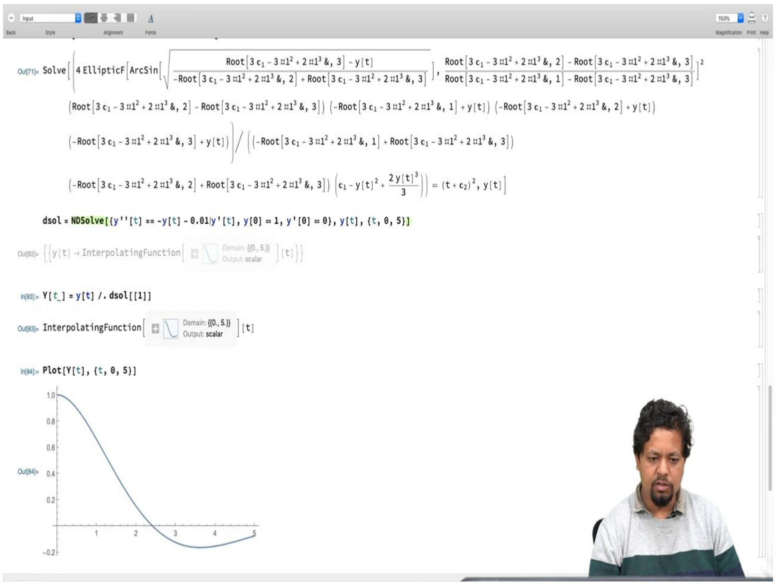
Similarly, we can also go ahead and do for $y(t)$, you can say $y(t)$ is nothing but $y(t)$/., slash dot is for replacement rule, this is the replacement rule that I am applying, I execute that and I get, this is $y(t)$, so let me go and put $x(t)$ and $y(t)$ in the same plot to compare these two branches of the solution, these two into different directions that is $x(t)$ and $y(t)$.

So, that was Mathematica's built-in function for solving differential equations using DSolve. Now this is an analytical approach, Mathematica gives you an analytical solution for a set of differential equations and the boundary conditions. But, there are examples where we can actually not use DSolve and we have to resort to some numerical method.

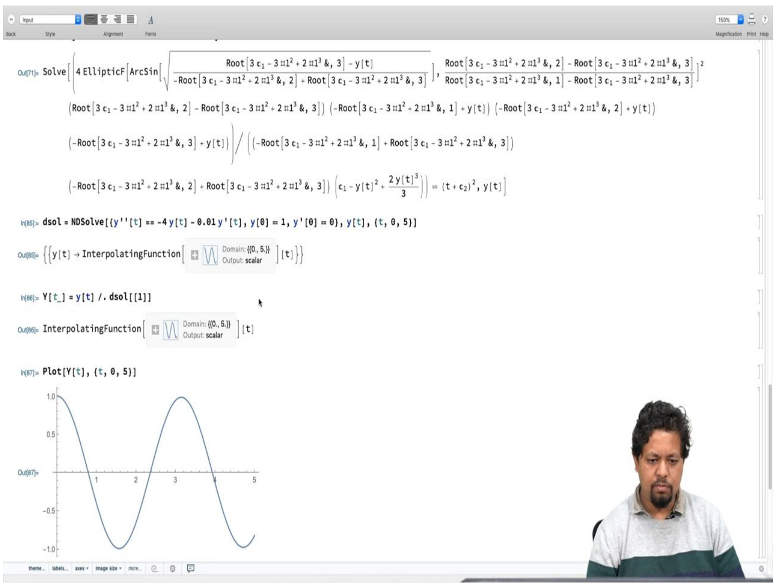
(Refer Slide Time: 26:00)



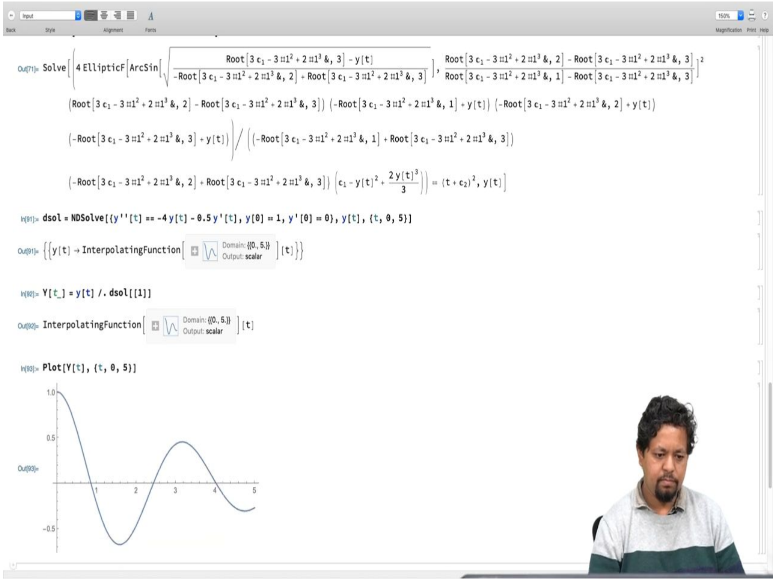
In[68]:= Clear["Global`*"]
In[71]:= DSolve[y''[t] == -y[t] + y[t]^2, y[t], t]
Out[71]= Solve[...]
In[74]:= dsol = NDSolve[{y''[t] == -y[t] + y[t]^2, y[0] == 1, y'[0] == 0}, y[t], {t, 0, 5}]
Out[74]= {{y[t] → InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]}}
In[77]:= Y[t_] = y[t] /. dsol[[1]]
Out[77]= InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]

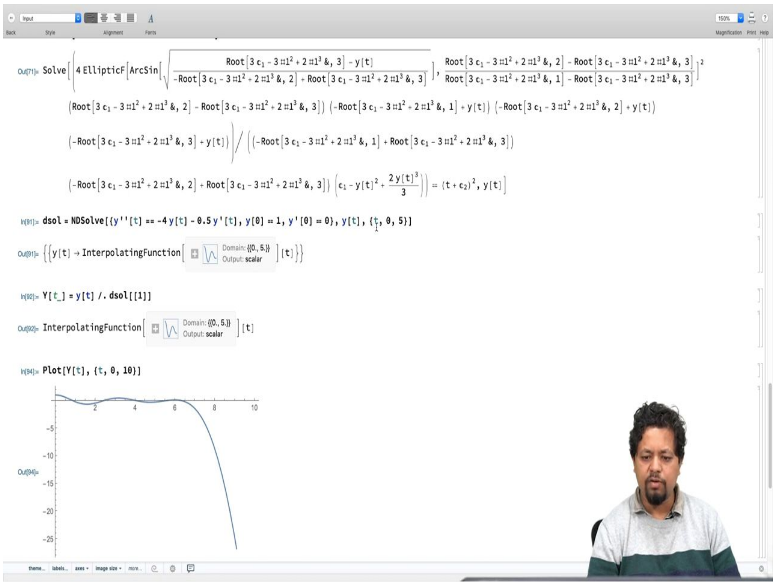


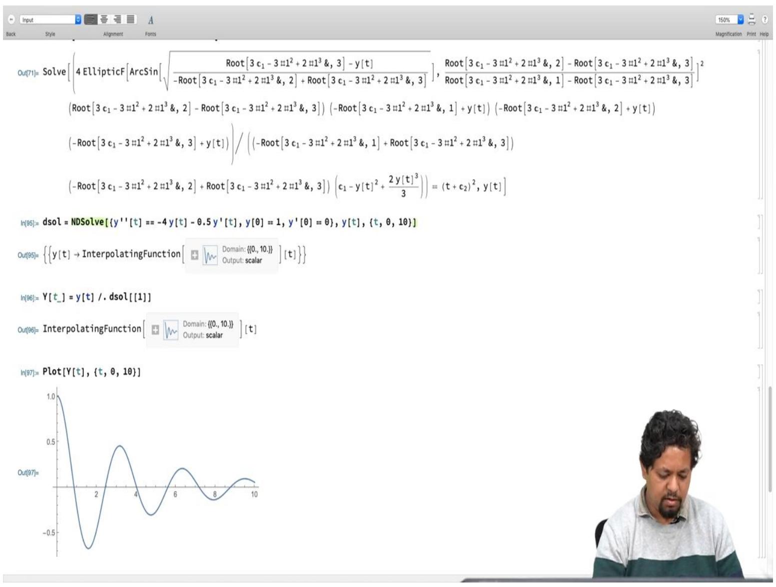
Out[71]= Solve[...]
dsol = NDSolve[{y''[t] == -y[t] - 0.01 y'[t], y[0] == 1, y'[0] == 0}, y[t], {t, 0, 5}]
Out[82]= {{y[t] → InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]}}
In[83]:= Y[t_] = y[t] /. dsol[[1]]
Out[83]= InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]
In[84]:= Plot[Y[t], {t, 0, 5}]
Out[84]=



Out[71]= Solve[...]
In[85]:= dsol = NDSolve[{y''[t] == -4 y[t] - 0.01 y'[t], y[0] == 1, y'[0] == 0}, y[t], {t, 0, 5}]
Out[85]= {{y[t] → InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]}}
In[86]:= Y[t_] = y[t] /. dsol[[1]]
Out[86]= InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]
In[87]:= Plot[Y[t], {t, 0, 5}]
Out[87]=


dsol = NDSolve[{y''[t] == -4 y[t] - 0.5 y'[t], y[0] == 1, y'[0] == 0}, y[t], {t, 0, 5}]
{{y[t] → InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]}}
Y[t_] = y[t] /. dsol[[1]]
InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]
Plot[Y[t], {t, 0, 5}]


dsol = NDSolve[{y''[t] == -4 y[t] - 0.5 y'[t], y[0] == 1, y'[0] == 0}, y[t], {t, 0, 5}]
{{y[t] → InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]}}
Y[t_] = y[t] /. dsol[[1]]
InterpolatingFunction[Domain: {{0., 5.}} Output: scalar][t]
Plot[Y[t], {t, 0, 10}]


dsol = NDSolve[{y''[t] == -4 y[t] - 0.5 y'[t], y[0] == 1, y'[0] == 0}, y[t], {t, 0, 10}]
{{y[t] → InterpolatingFunction[Domain: {{0., 10.}} Output: scalar][t]}}
Y[t_] = y[t] /. dsol[[1]]
InterpolatingFunction[Domain: {{0., 10.}} Output: scalar][t]
Plot[Y[t], {t, 0, 10}]

Let me give an example, Let us try DSolve for some equation, let us say $x''(t)$, so $x(t)$ is let us say $-x$. We know this is the equation for, harmonic oscillator and we know the solutions of this are $cos(t)$ and $sin(t)$. Let us go ahead and check it out, we will just say $x(t)$ and we say that I want to solve for $x(t)$ as a function of t. There we go, I made a mistake, it is to be equal to equal to and we got to clear x.

I have to actually go and clear, $x'(t)$ I think. When that happens the only way is to clear everything, so think over, start and clear everything from the context. I am going to try now, it still does not work. Let me go ahead and make this, change the variable here. There we go, so that happened because I accidently used equal to and double equal to, and something goes in the context that gets defined and Mathematica does not like it.

If it happens with you, do not panic. Just clear up the memory, close the file, and close the Mathematica and reload the file or change the variable just like the way I did, if that happens to you. So now when I solve this I get this solution $c1\ cos(t) + c2\ sin(t)$, you all know that this is the general solution for a simple harmonic oscillator. Or let me go ahead and add a term to this.

Let us say I am going to add $y(t)$ to the power square as to $y(t)^2$ and go ahead and try to solve this. It can actually, it appears it is solving but this solution is not very useful because see this solution is packed inside the solve function itself. So, it says that it has got some idea of the solution, it says that you know the solution is given by all of this nasty stuff, the solution is given by solving this equation and you have to, if you can solve this set of equations you have the solution for $y(t)$.

Now that is not very helpful. But, otherwise, this is not making much sense and in this kind of case we will probably want to get a numerical solution. In that case, we will resolve to rather than DSolve, we will resolve to NDSolve, which is a numerical method for solving differential equations. Now NDSolve can solve a whole lot of differential equations but maybe not everything.

But in any case what we will do is we will go ahead and explore NDSolve, for this particular problem so the function is simply NDSolve like I have shown you over here. But, NDSolve does not work if I do not give a boundary condition; I have to give initial value conditions or boundary conditions for NDSolve to work. Just like Nintegrate does not work for doing indefinite integrals, NDSolve does not work to solving indefinite differential equations without any boundary conditions.

In order to provide boundary conditions, I must put this, provide some more equations for that I am going to go ahead and say $y(t)$, $y(0)$ that is $t = 0$ is, let us say it is some amplitude 1 and since this is a second order of differential equation, I have to provide 1 more boundary condition, this time it will be $y'(0)$, that is the first derivative of $y(0)$. Let me go ahead and call it 0, so essentially what I am doing here is I am, this is an oscillator, this is an anharmonic oscillator.

I am saying that you take my harmonic oscillator set it at $t = 0$, set it to some amplitude 1 and speed 0 or y dot $= 0$. That is like setting the pendulum or your harmonic oscillator the maximum amplitude and releasing it. So, let us go ahead and try that, we have to also say for what time range. Let us go ahead and try, just like that it should complain. It says that the range specification, t is not a form of the form some range for t. so we want to provide some range for t.

Let us go ahead and say t goes from 0 to 5 and let us go ahead and execute that, there we go, it gives me some expression, it is unclear what that result is, it is in some sort of interpolating function, but if you see this do not panic, the important thing to note here is that this is still a replacement rule and $y(t)$ is some in-built numerical function that Mathematica is providing, we will learn how to extract this out and plot it.

So, here is again 1 single solution, it has found and let us go ahead and extract this out. So, let me call this is as dsol as my solution. We need to go ahead and execute that and I am going to say dsol[[1]] which is simply going to give me the replacement rule, now say $y(t)$ apply to dsol gives me this function, again I want to do what I did some time back. I wanted to define $Y(t)$ as a function. I would say that this is equal to that.

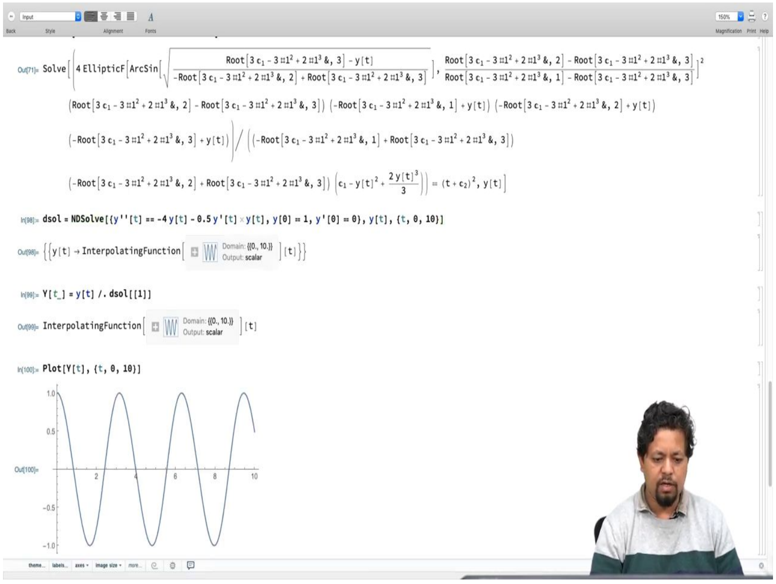
Let us execute it, now $Y(t)$ is an ordinary function for me, which I can go ahead and use like ordinary function and plot it. So, t goes from 0 to 5 and that gives me just a constant line. So, that did not give me anything interesting, let me go back and change something about it. Let me go ahead and make it cube and try this out. Again, it seems to give a flat line, okay, let us try something else.

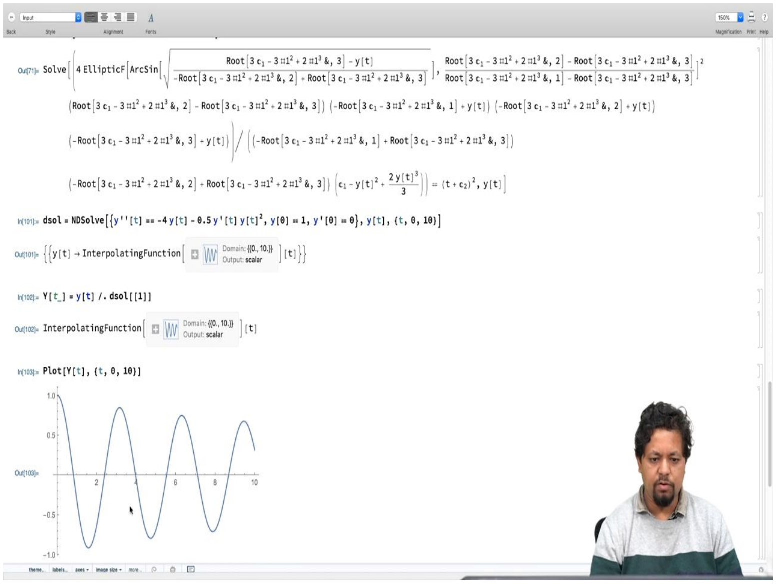
Let us try a damped harmonic oscillator, so let me put this as $-y'(t)$ and there we go. That looks something interesting and there we do, this is a damped harmonic oscillator, if we make the damping term really small, you should be able to see many oscillations happening. And let us go ahead and increase the frequency, let us make it 4 and there we go and now the damping is probably too slow, so let us increase it slightly. So, let me see a nice damped oscillator, you may get it with more, there we go, and we have a nice oscillator.

We can try and go ahead and say what happens, see we solved it up to equal to 5. But, Let us go ahead and extend it out to 10. You see when we do that up to 5, it gives a reasonable solution. But, then it diverges at 10, because we asked dsol to solve only between t from 0 to 5. It only gave me an accurate solution between t equal to 0 and 5.

If I increase it to 10, you will see how the result will change and there we go and that's the solution of a damped harmonic oscillator using a numerical method of doing NDSolve, this equation is analytically solvable and we will do this in one of the lectures. But, let us take an example where this is not analytically solvable and that happens when, that happens when, when I put a frictional term.

(Refer Slide Time: 34:06)

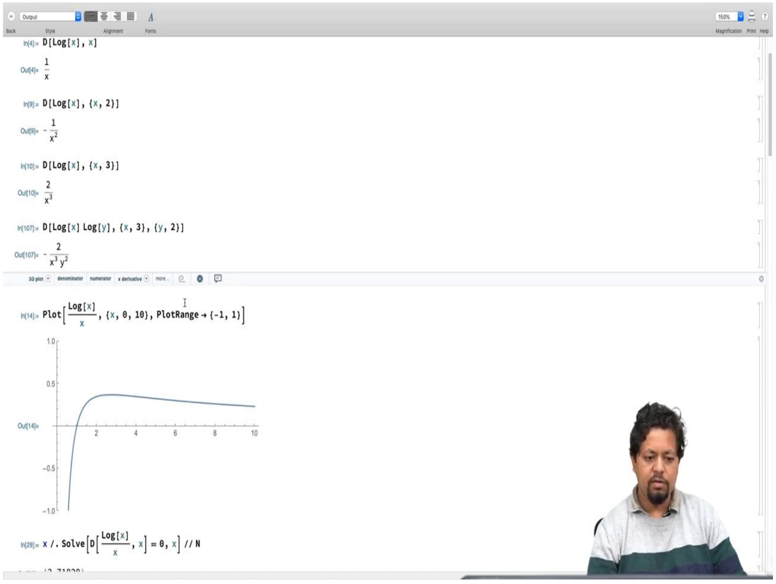




For example, if I put in some other terms let me go ahead and put in $y(t)$ as well and that is probably not solvable. But, Let us go and see what happens over there, there is some strange behaviour nevertheless. Let us make it square, it is probably doing some restoration. There we go there is some damping. So, this is some sort of damping term where damping is opposite to velocity but with the y^2 down.

We can increase the damping and that should make the amplitude go down faster, so this was an introduction to calculus tools built into Mathematica. Just do a quick overview, we talked B about derivatives, we talked about higher-order derivatives. We can also do derivatives with respect to multiple variables, for example, Let us go ahead and check out this 1 more example.

(Refer Slide Time: 35:00)



Let us say this is a $log(x)$ and $log(y)$ and I want to find derivative respect to x and y, I can simply say third derivative x and second derivative respect to y and that will give me the corresponding result. You can cross-check analytically that this is actually correct. So, we talked about derivatives, we talked about integral, definite and indefinite integral.

Then we talked about the numerical integration method and solving differential equations both analytically using DSolve. And numerically using NDSolve. We will use some of these functions as we go ahead and progress in this course. We will see you in the next video.