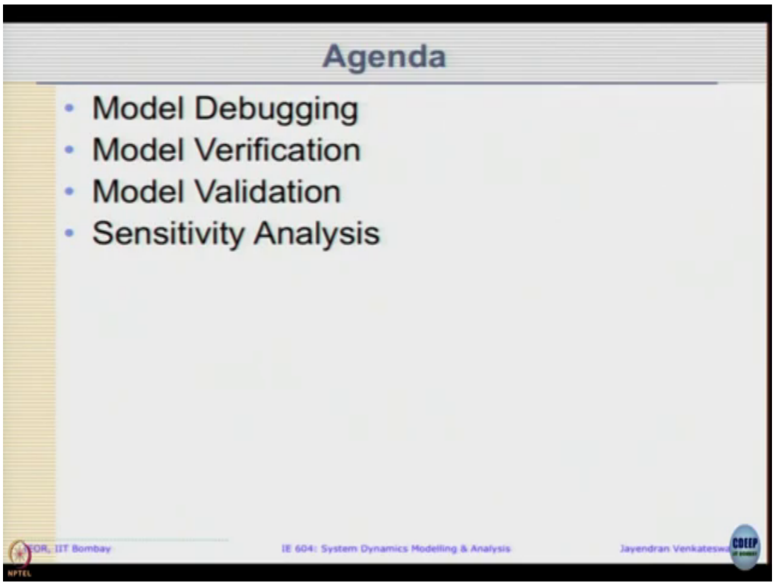
Introduction to System Dynamics Modeling
Prof. Jayendran Venkateswaran
Department of Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

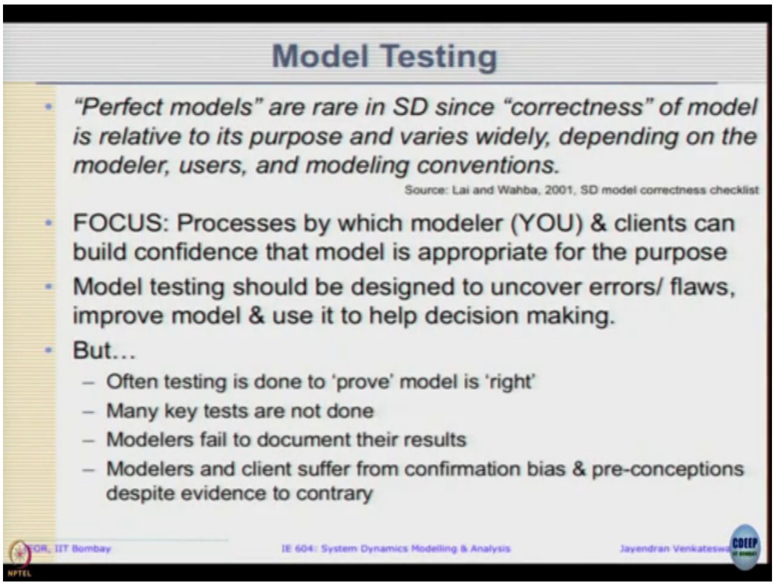
Lecture - 23.1
Testing System Dynamics Models:
Introduction with example 1

(Refer Slide Time: 00:23)



Today, we will briefly look at Testing of SD Models, today's agenda will cover a few topics or other the concept of testing of SD models; you can start with model debugging, model verification, model validation and sensitivity analysis. So, these are broad topics that we look at when you think about model testing.

(Refer Slide Time: 00:55)



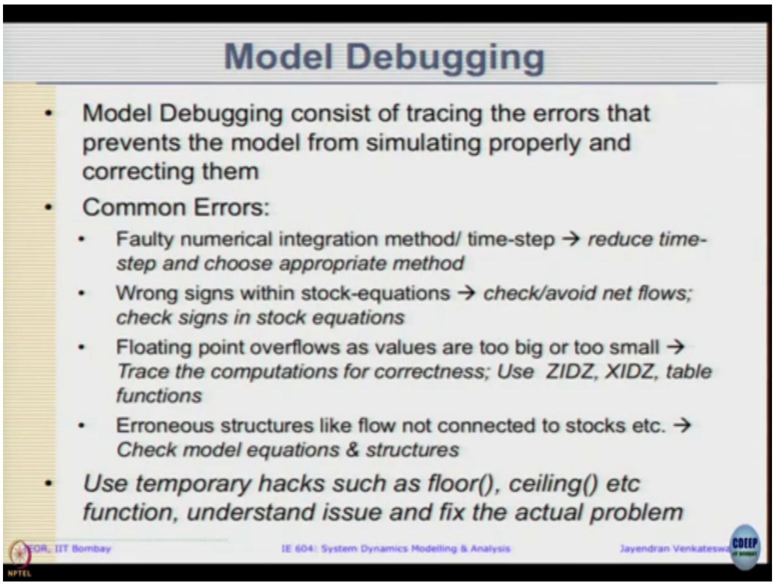
So, why do you want to do model testing? The perfect models are going to be difficult in the case of SD since correctness of model is related to the purpose and varies widely depending on the modular user as well as the modeling conventions.

However, we need to build confidence of the model that we have built so that we can gain useful insight, as well as present our finding appropriate to the purpose on hand. So, this is what we want to do. So, want to test a model so that we build a confidence on a model, so that we can have some belief in the result that is going to give us. In model testing, we should be designed so that we can uncover the various flaws or errors that we do in the modeling. Some are pretty straightforward if it is programming kind of errors, but logical errors a little more difficult and we can use it to improve the model to make it better.

However, we do testing to prove model is right, rather than try to uncover what is a right model; we will come to that later. So, then key test need not be done, modelers fail to document result that is nothing new it is been there for ages. Then OSD started computations, documentation is always be an issue and modulation clients of a confirmation bias and preconceptions despite evidence to the contrary. And then you do not want the results to show what you want to or you not happy with the results it shows; so expect the model to be wrong.

So, we need to address these things systematically. We will start with a few basic steps and go ahead look at how to validate the model etcetera ok.

(Refer Slide Time: 02:41)



So, the first step towards is model debugging; we already had an introduction to it last class some time ago, there we are looked at a model description and try to uncover the various errors within the model; you already had an experience of that.

What you want to do is, debug the model consists of tracing the errors that prevents the models from simulating properly and correcting them; it is pretty basic step. The common errors or some of it are as follows, the faulty numerical integration method time step is being used. So, one way to counter it is reduce the time step and choose an appropriate method. So, what is written above on the right side of this arrow indicates how we have to counter that flow. Like we have to use conjugative method, we used arced method, Euler method or time step is too large; you have to reduce the time step which is now appropriate time step so that we can simulate it properly.

Wrong sings within stock equations; one get avoid is to check and avoid the net flows, many times we end up instead of inflow and outflow explicitly, we can actually moderate as a net inflow. So, we do a net inflow then we are be careful of the signs so that the model stimulate correctly. Floating point overflows as values are too big or too small or we end up dividing by 0; only with avoid check that is to trace the computations, figure out at what point the error occurs, look at the table of values and correct the model to see whether that is did that large value is it realistic or not.

There are functions in Vensim like ZIDZ or XIDZ. So, this is when you divide by 0 what is the value to return. So, instead of returning when you divide anything by 0, it will to be indeterminate; to avoid that this XIDZ and ZIDZ can be used; you can look it up in Vensim help. When you divide by 0, then what should it return? It should return 1 or it should return some 0; what should it return that we have to specify and Vensim supports that.

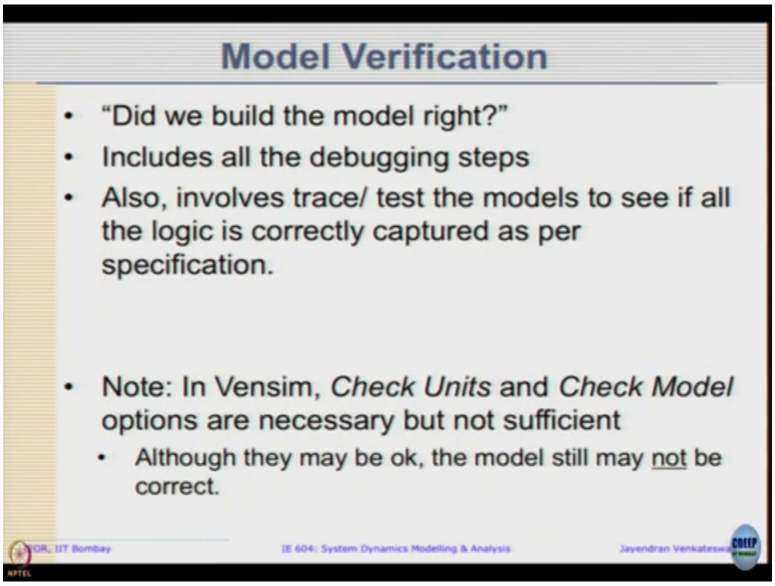
So, maybe that may be required if we did division by 0 occurs or check out the table function, this your is extra politics more than what you desire it to be next especially in extreme; in that case then we need to do the value such that it extrapolates correctly. So two types of errors;

one is a warning that the competition is beyond the table function; that means, extrapolating; some cases it may be ok.

Other is floating per the errors it actually need by dividing by 0 or due to some error in computations it is getting a large value that we need to fix. So, in the simulator and we get an error then there is something you need to fix. And that could be error in the structure itself like flows are not connected to stocks, a drawing was not proper etcetera. So, these you need to check the model equations on the structures directly.

So now, we can use temporary hacks such as floor function or ceiling function or look at a separate part of the sub; sub model of the entire model to see understand issue and fix the actual problem that is actual occurring within the model. So, that is a broad steps about debugging. Closely related debugging is this model verification overall; we what we want answer is this question; have you build the model correctly?

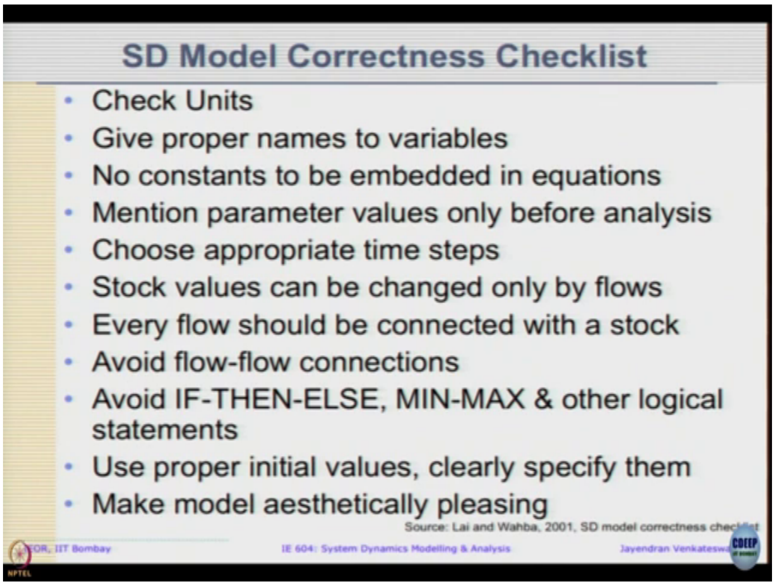
(Refer Slide Time: 06:18)



It includes all the debugging steps, but also goes beyond to trace it and test the model to see of the all the logic is correctly model and captured as per the specification; this is what we want to do; so that in trace what we call this model verification.

So, model verification starts with basic debugging and goes a little beyond that one also. We have few tools in Vensim itself one of it two of it we have already used to check units and check model they are necessary, but not sufficient; that means, even if you give the model and click check units, it says units are fine still there could be errors inside the model; that is in what values are giving or connections you were giving there could still be errors in the model. So, they are necessary, but not sufficient.

(Refer Slide Time: 07:15)



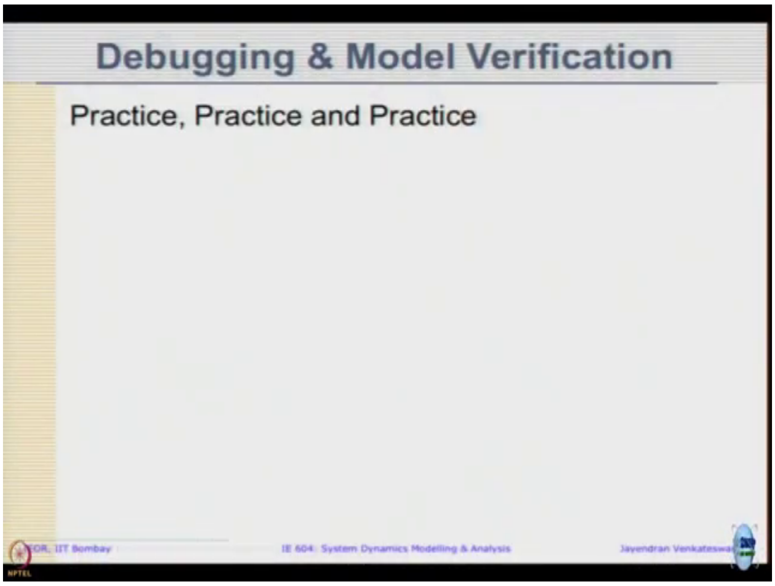
So, we need to actually check for correctness beyond what we see here. Various people are come up with various checklists that we can go through to see if the model is correct. First is to check units and see if proper names are given to the variables or it makes sense to us, no constants are embedded in the equations, they should try to mention the parameter values only before the analysis; choosing appropriate time steps. Stock values can be changed only by flows, every flow should be connected with a stock; we should try to avoid flow connections.

So, try to avoid if then else and min max and other logical statement as much as possible, but problem entails and we need to include it, else we need to use proper initial values and clearly specify them so that model can start at dynamic equilibrium. And lastly popular owners to make model aesthetically pleasing and as we organically grow the model, we will find that it is looking quite complex.

But finally, it has been presented to an external audience how do we make it aesthetically pleasing. So, one easy way is to use curved arrows instead of straight lines because it is a little more pleasing than straight lines and sharp lines. So, this is also quite important. So, some of these things may seem quite obvious that we ought to do, but many times in our case we do not do that because its; its seems to trivial for us to worry about, but unfortunately that is the one which ends up causing the errors or causing the various issues in our actual model.

So, the one way or rather the only way to ensure you are doing proper debugging and proper verification is to practice.

(Refer Slide Time: 09:13)

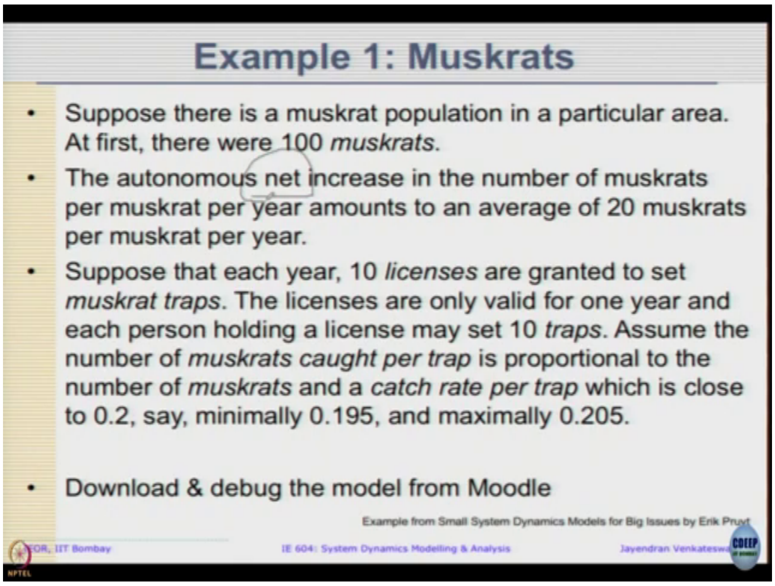


So, that is what we are going to do today; we are going to take up a few models and try to practice the debugging. In fact, this time what we are going to do is; I am going to lead the debugging that is I am going to read the description, point you to different things so that you can observe and follow; even if you are not able to catch up in the model my suggestion is you go back and look at this video again to see how I am impressing through the model so that we can also follow the similar steps.

So, typically how I go about checking the model is what I am going to go through. So, let us and for; so we have three examples we are going to do todays class. Reach how example that the model with errors is already online, you can download them from Moodle. And when each scenario comes you can try to reopen the model and see whether it confirms with our understanding that ok. I hope I will remember mention all the points that I am going to check, but let us see.

Again these are only some of the cases that you are going to see. So, it is important for you to practice with other kinds of models actually make some mistakes at least some say for example, from exam point of view; we may not get exact same errors. So, there may be any other form of combinations, I try to give as many examples as possible.

(Refer Slide Time: 10:42)



So, let us see; the first is let us this understand the model no need to look at the Vensim model yet, we have time for it. So, very simple model that is something called as muskrats it is like a big large road in; section native of North America and invasive speries in large parts of Europe and North Asia that is regions near Russia and stuff.

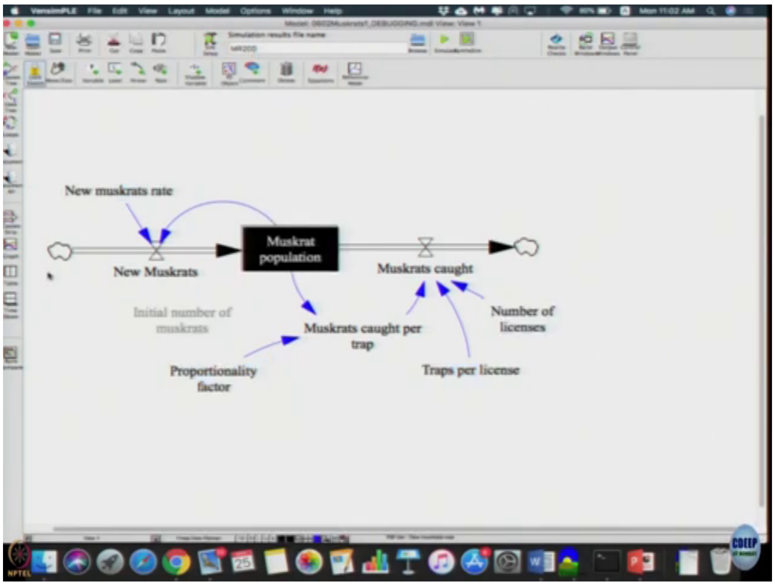
So let us go with this; suppose there is a muskrat's population area initially were 100 muskrats. So, as and when we read the description; we needed visually what visually what can I say, I need to imagine how the model is going to look like; as soon we use the term like population muskrats population area some 100 muskrats. So, terms that are you could be a stop; so look at the population, so maybe it is a stock; so let us picture that autonomous net increase in the number of muskrats per muskrat per year amongst an average of 20 muskrats per muskrat per year.

So, let me just talking about some net increase in the muskrat population. So, this must; this should be a flow that is going into the muskrat stock ok. Then suppose it each year then license are granted to set muskrat traps, these licenses are valid only for 1 year and each person holding a license may set 10 traps, assuming the number of muskrats caught per trap is proportional to the number of muskrats and a catch; catch rate per trap which is close to 0.2. So, minimally 0.195 maximally 0.205; so this is only description given.

So, thissecond part since we already talking about net flow, the second part may be something referring with how much is removed from the population. So, this must be the out flow within the model; explicitly as you can see there is nothing about birth and death that is given. But if you carefully observe the description, there is; there is a word net given this could mean ok; there is something about birth and death that is happening, in spite of that an average of 20 muskrat per muskrat is increasing. So, that is what the description that is given.

So, now you need to download and debug this model sometimes when you open the model it talks about scaling and preferred as 100 percent; so you get the exact same size.

(Refer Slide Time: 13:14)

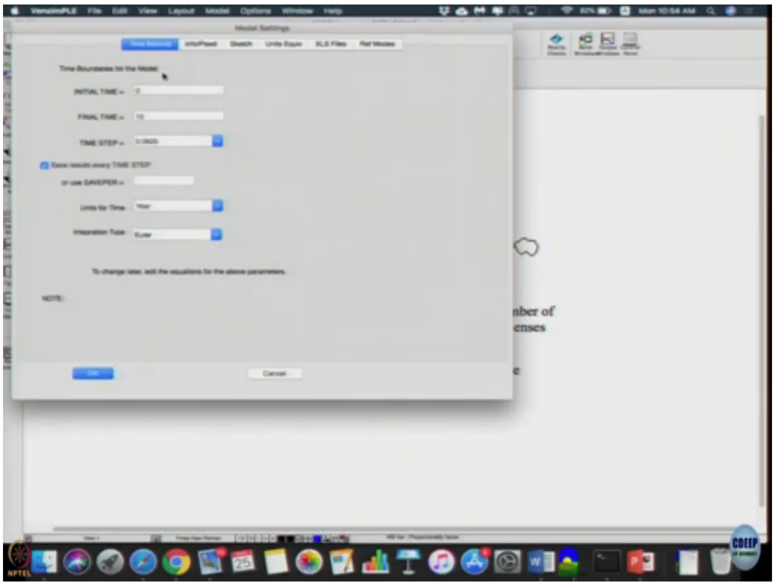


This is a model that you will see; first is the first step is you look at the structure right. So, we imagine there is some muskrat population some new muskrats come in and some muskrats is getting caught; so they removed from the population.

So, description is not fully complete, but given the description you can assume that the muskrat is caught and terminated and not just caught and released back into the population. So, there are some reasonable assumptions to make; there are some variables like how much muskrat is caught per trap, how many traps per license, how many number of licenses; it seems kind of structurally ok.

The next we can try to do is look at the model and so check units; units seems to me ok; model check model, model also seems to be. Then we can quickly trace through the equations; so new muskrat was about 20 muskrat per muskrat per year right.

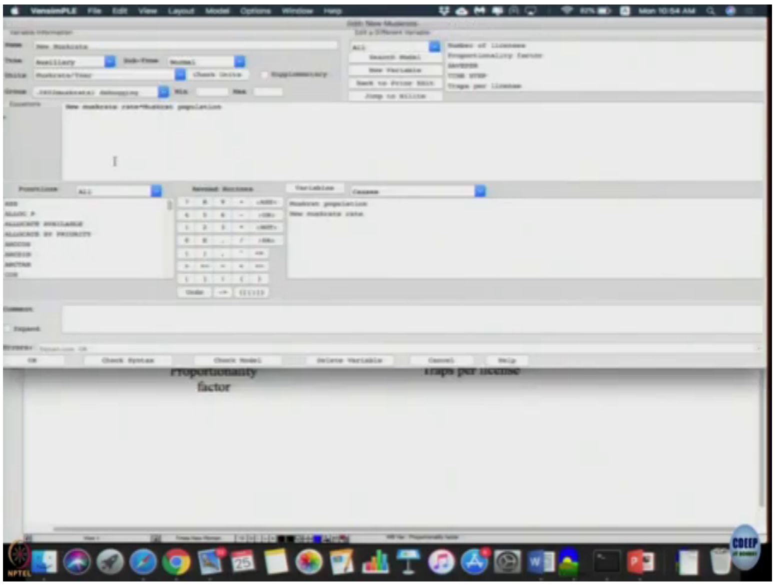
(Refer Slide Time: 14:33)



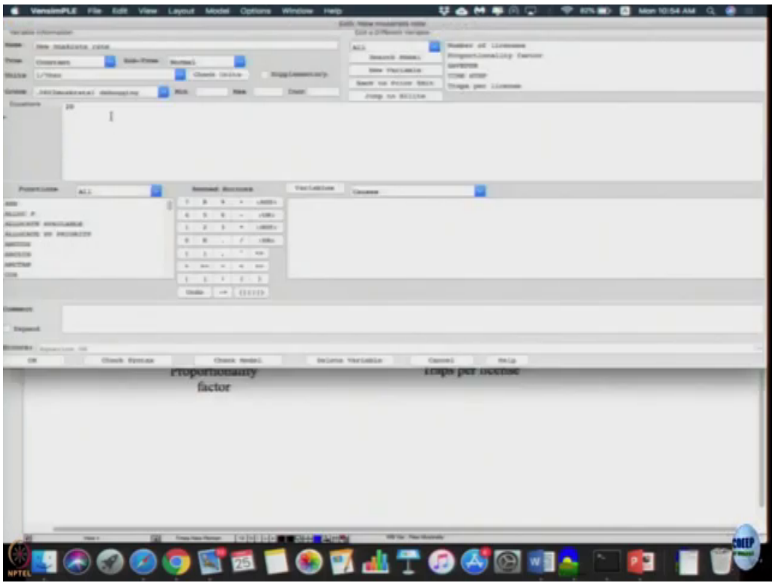
Ah. So, before we going to that; we can actually look at model settings to look at what is a time units; first try to you see that ok. All the description where in nears and we have time units also in nears and let us look at this side time step; 0.0625 Euler's method. So, is fine is reasonable quite low, but let us see we do not know since it have an impact.

So, let us look at the equations; you just move the mouse, what are the units pops up below that; you can see. Units seems actually be fine to open it which says new muskrat in to muskrat population.

(Refer Slide Time: 15:08)



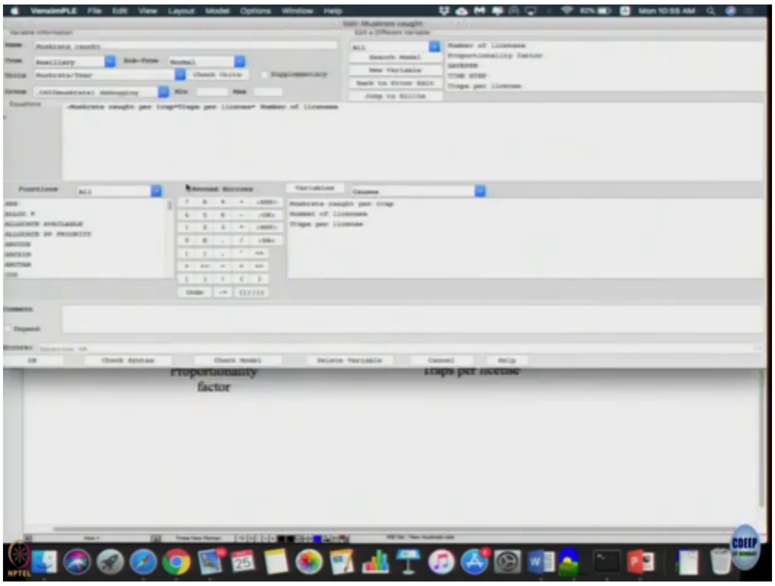
(Refer Slide Time: 15:13)



A new muskrat rate is about 20; it is consistent with what numbers we have. See any change in constants is not that big of issue; it is need to ensure that it is fairly accurate, units that is a bigger problem.

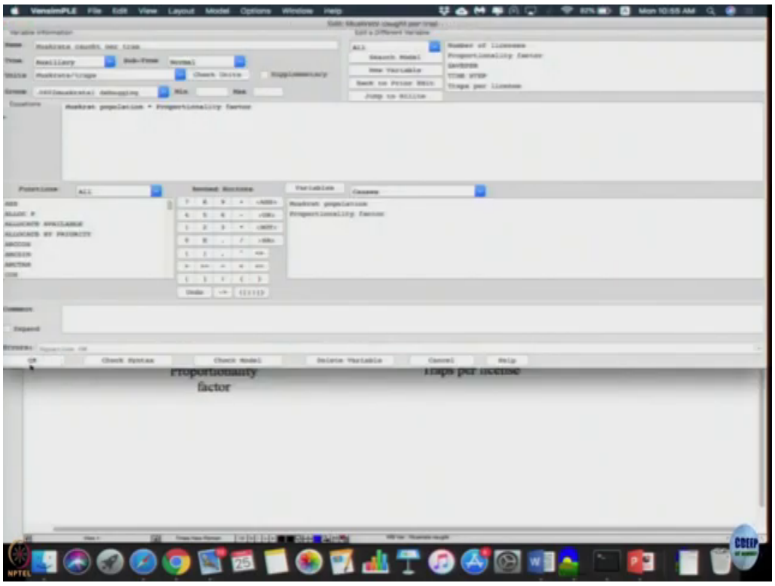
So, muskrat population should be roughly the muskrat caught per trap into the proportionality factor as per the description. And number caught should be number of licenses into number of tasks per license multiplied by number of muskrat per trap that should logically give us the number of muskrats caught.

(Refer Slide Time: 15:48)



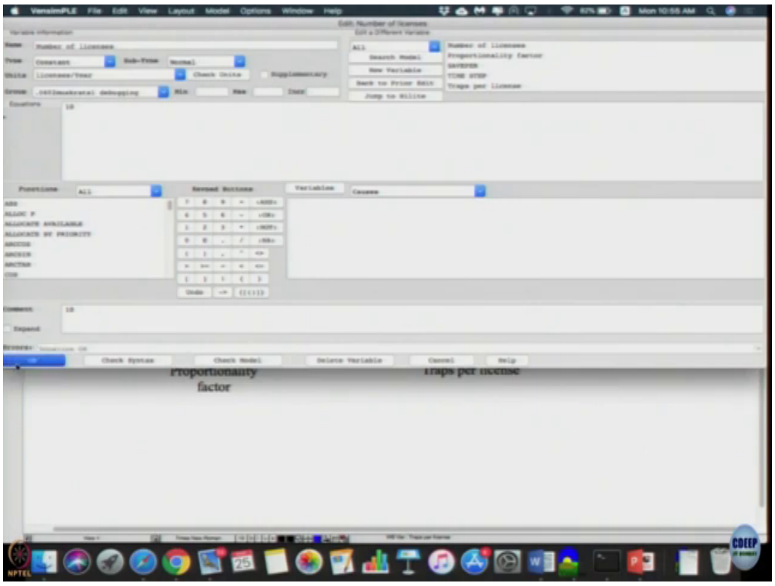
So, which is a product of all three.

(Refer Slide Time: 15:53)



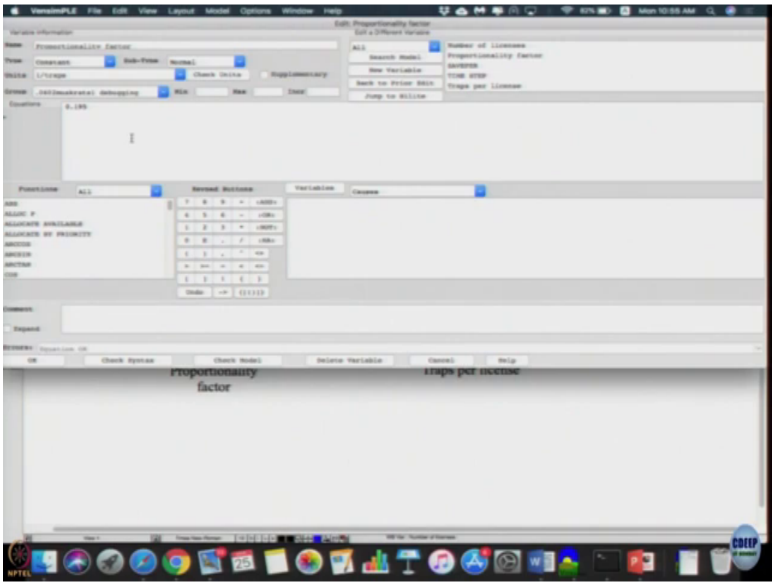
Muskrat caught is into proportional to factor.

(Refer Slide Time: 15:58)

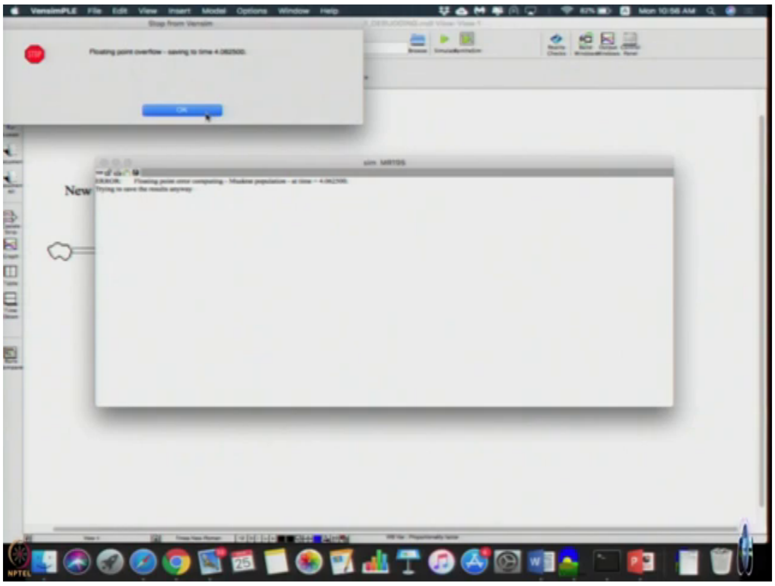


Traps per license is 10, number of license is 10 which is roughly the what the description as said 10 on 10 traps, proportionality constant is 0.195 on this taken here 0.195; so minimally it is here right.

(Refer Slide Time: 16:16)



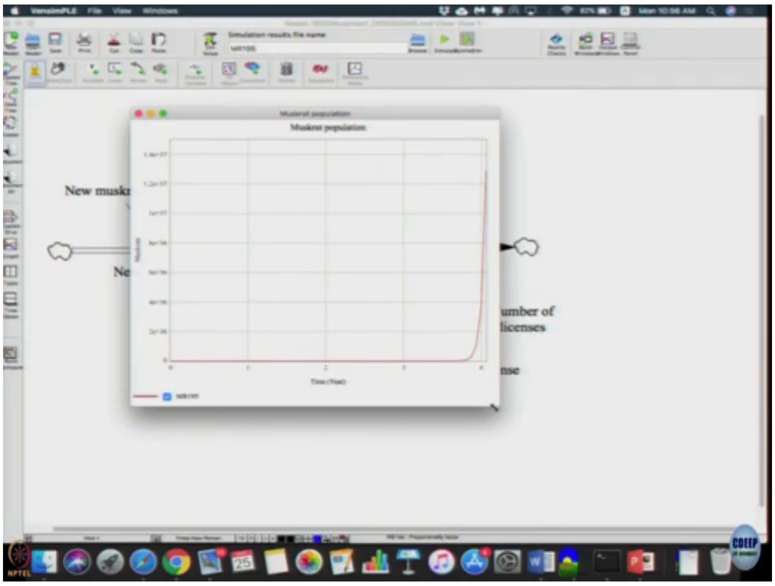
(Refer Slide Time: 16:28)



So, seems let us just stimulate it; that means, let us see if there is errors over write yes ok.

To get this errors floating point overflow; there is an error and error says floating point error computing muskrat population at time 4.06; a simulation running length of 10 years at around 4.06 itself is getting a floating point error, trying to say results anyway. So, let us click ok, close it.

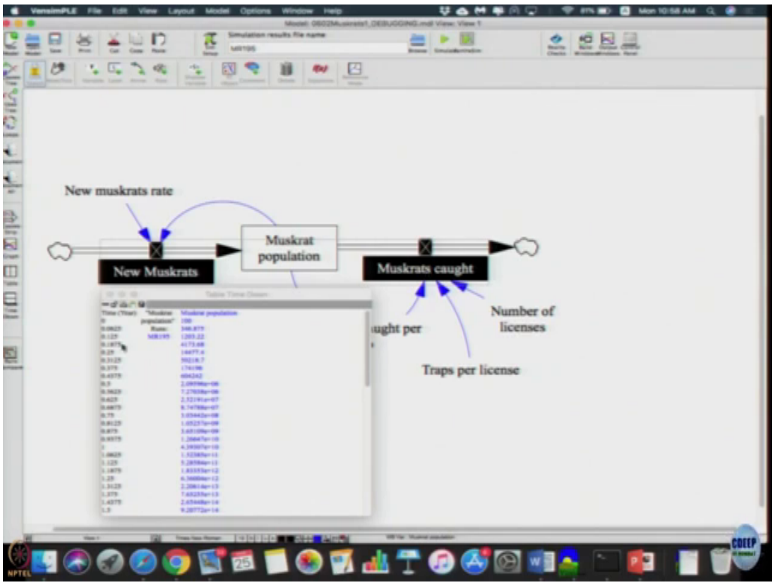
(Refer Slide Time: 16:53)



Let us; see the graph, the graph is growing; how is a graph growing? No, seems to be growing hyper exponential difference; exponential means it has to have doubling time which is constant; here it is growing hyper exponentially much more than what it has to be right. So, there must be some error that is one and two; actually if you look at it, there is a positive feedback system it has keep increasing, but this is the negative feedback system; so it has to decrease right

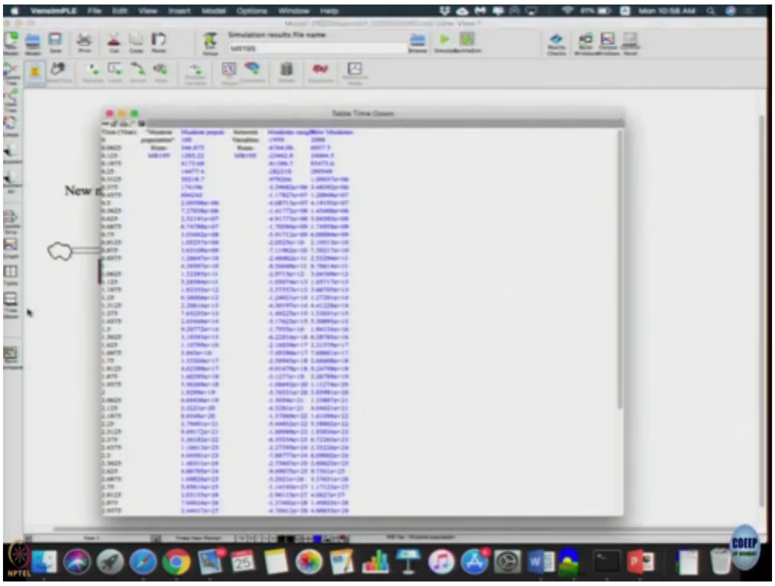
So only descriptions it can have its exponential growth, exponential decay or equilibrium inflow and out flow is equal right. This system cannot have an exponential; hyper exponential growth hyper exponential growth your proportionality constant or a new muskrat rate or something has to keep reducing or keep increasing; number of new muskrats should increase a proportional rate should keep decreasing. So, none of the proportionality constants are changing; here the proportionality constant is 0.195, 10; 10 right. So, none of the constants are changing correct. So hyper exponential growth should not occur; so just try to follow my steps.

(Refer Slide Time: 18:26)



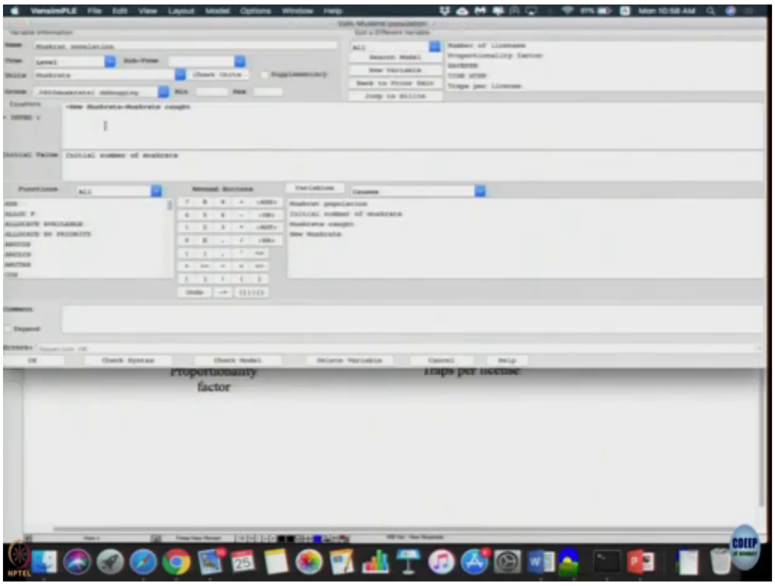
So, but still muskrat population is changing; if you look at that population is just increasing all the way and what affects the population? Population affected by both these flows. So, let us include that also in the table.

(Refer Slide Time: 18:38)



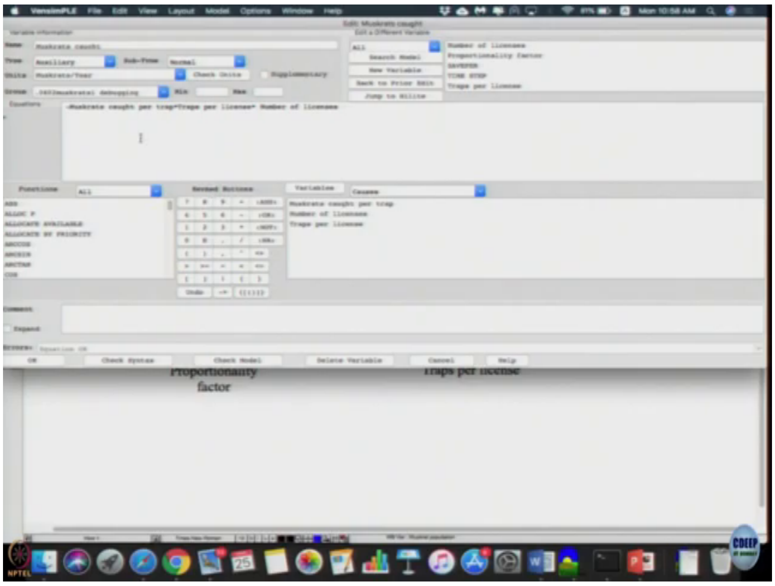
I click the time down table; here muskrat population and then suddenly you find that new muskrats are 2000, new muskrats first period is 2000 which make sense; 20 into 100, 2000; number caught is 0.195 into 10 into 10, it should be 1950 which is fine, but it is minus 1950, let us look at it what is happening there.

(Refer Slide Time: 19:12)



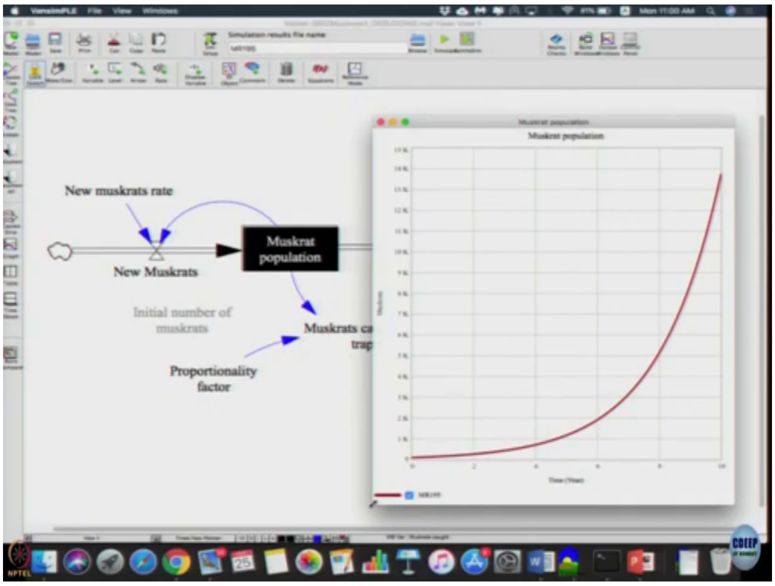
Let us looks at the population here this new muskrat minus muskrat caught ok; inflow minus outflow this is fine.

(Refer Slide Time: 19:21)



Just look at muskrat caught; it is minus muskrat caught per trap into this. So, this minus should not be there because its minus and then we already in compute in the stocks if minus of minus; so it became plus. So, stock value started increasing more than it odd 2. So, let us remove this minus sign; let us click ok, let us run the model, we did not get any errors when we run it this time.

(Refer Slide Time: 19:50)



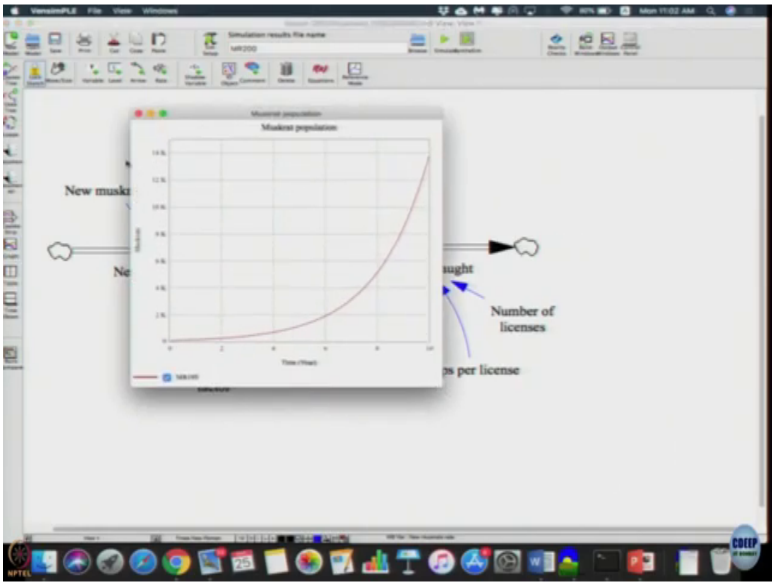
Still we are getting exponential growth this is fine, but at least doubling time is you can check it; doubling time should remain constant.

So, this looks reasonable for this kind of errors; it does not seem to be any other errors using the model, but these kind of small bugs also can weep into the model. So, even though when you run it; it did use some results, it did not make logical sense. Expectation is we actually look at the result and then we try to see how we can understand the dynamics and see whether the model inherently can cause it right. But if we move to sensitivity analysis in other concepts also we will end up using these; these models itself. So, save the what can I say this version of the model ok.

Now, you may not still be done; it says catch rate per trap is 0.2, minimum 0.195 to maximal 0.205. We can logically see for example, muskrat population is 100 and this is 20, new muskrats is 100 into 20, 2000. So, instead of proportionality factor in 0.195, if it is 0.2; then we get 0.2 into 10 into 10 into 100 which will also be equal to 2000 correct. So, in that case we need to get a straight line. So, we can check if the model is actually producing all the behaviors as far a expectation.

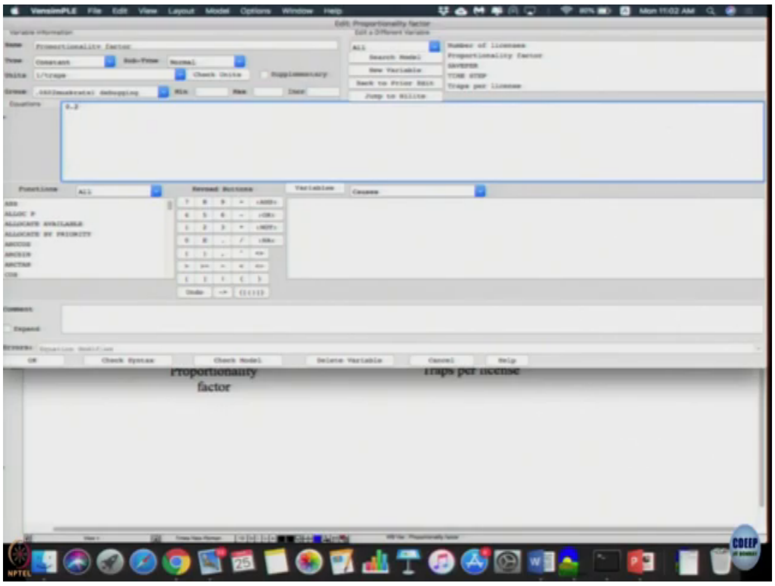
So if it is anything beyond 0.2; then what should it be? It should be have exponential decay. So, we can check with the model actually producing all the behaviors; only then it is a completely verified model. So, one way to do is without see changing, the model as let us instead of MR 195; I am going to write 200, click sim setup, click proportionality factor and write 0.2; it is ok, let us stop it.

(Refer Slide Time: 22:36)



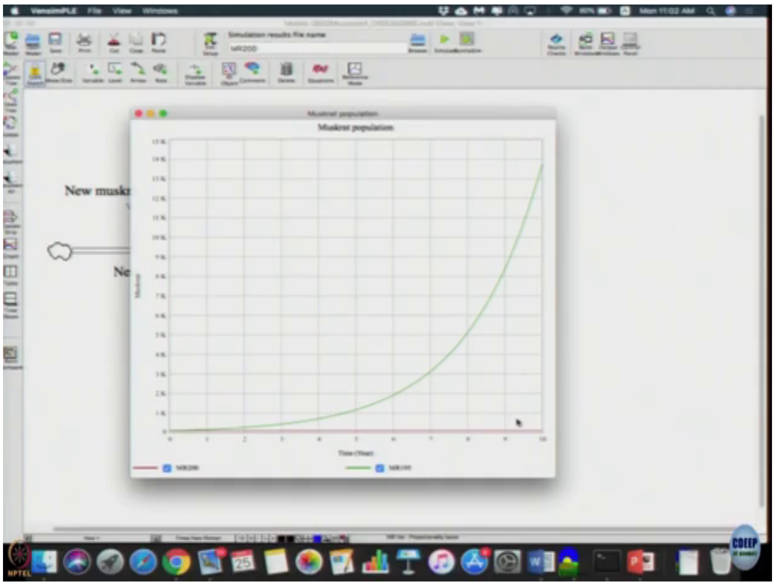
Now, let us do muskrat population; I just stopped here fine (Refer Time: 22:42) setup now we will do it. So, MR 200 let us change it.

(Refer Slide Time: 22:48)



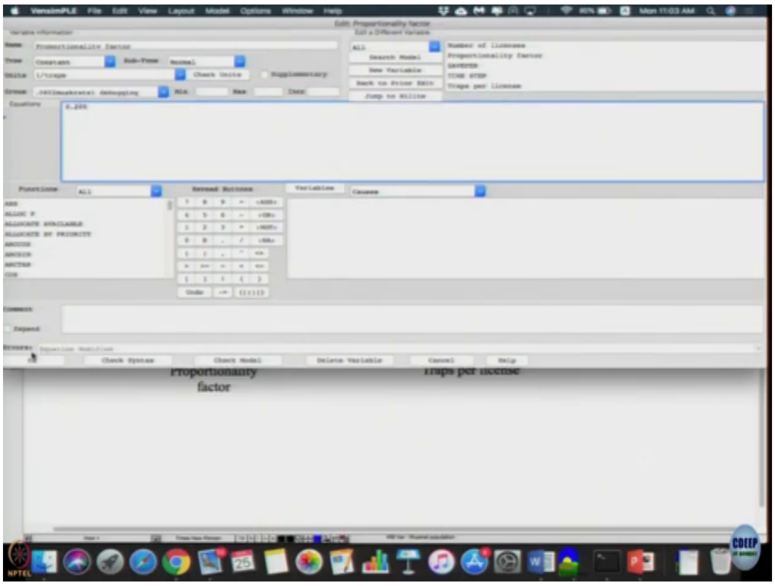
Let us go to the proportionality factor; I just make it 0.2 to play.

(Refer Slide Time: 22:57)



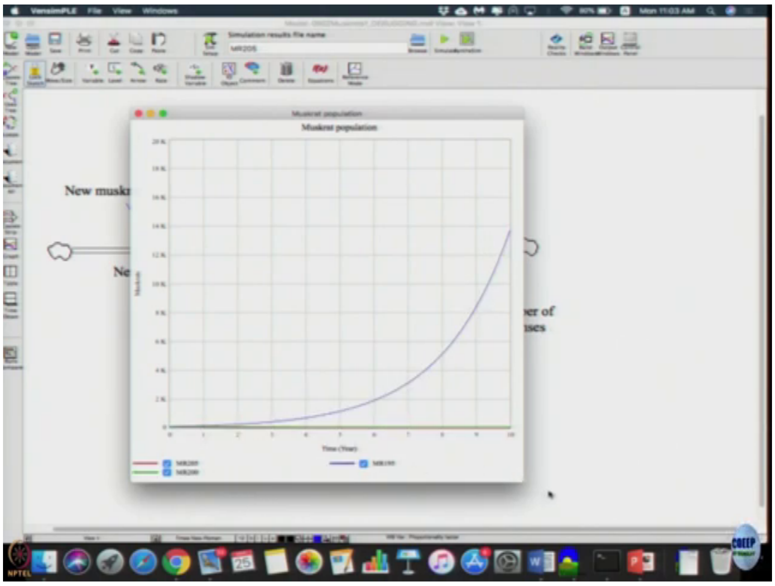
Now, actually we get two graphs; one is a small red line in the bottom is actually being constant and the other is increasing the exponential, you can change it to 205 as a simulation file result name.

(Refer Slide Time: 23:20)



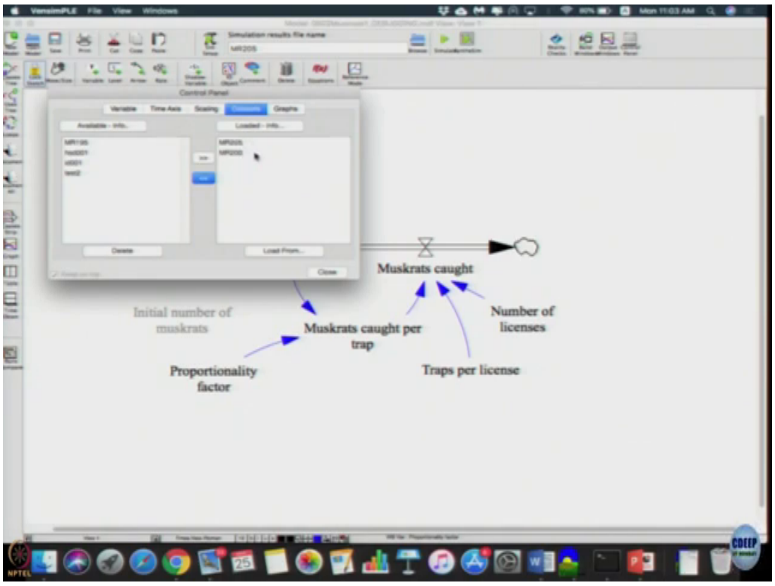
And change the proportionality factor to 0.205; click ok, click run.

(Refer Slide Time: 23:27)



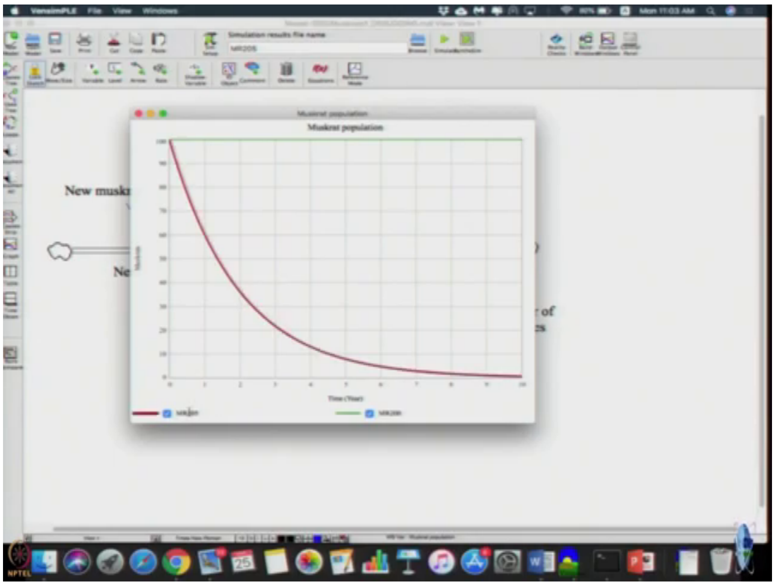
Now, if you do the caught the population. So difficult to discern here; I have exponential growth and other only supposed to be exponential decay, but I can make it out from here.

(Refer Slide Time: 23:40)



So, in that case; you can just go to our control panel; data sets, remove the 195 data set 200 and 205 is what we have, click ok.

(Refer Slide Time: 23:51)



Look the graph; now you can get a constant line corresponding to proportionality constant of 0.2 an exponent decay when if it corresponds to 0.205; it is exponential decay, it is able to produce the results that we can expect from a first order system. So, model is verified which can use for further description and analysis. Just to quickly go over it; so we have record how to go about it. So, we revise it; it is quite easy for you.