Distributed Optimization and Machine Learning

Prof. Mayank Baranwal

Computer Science & Engineering, Electrical Engineering, Mathematics

Indian Institute of Technology Bombay

Week-4

Lecture-14: Stability theory

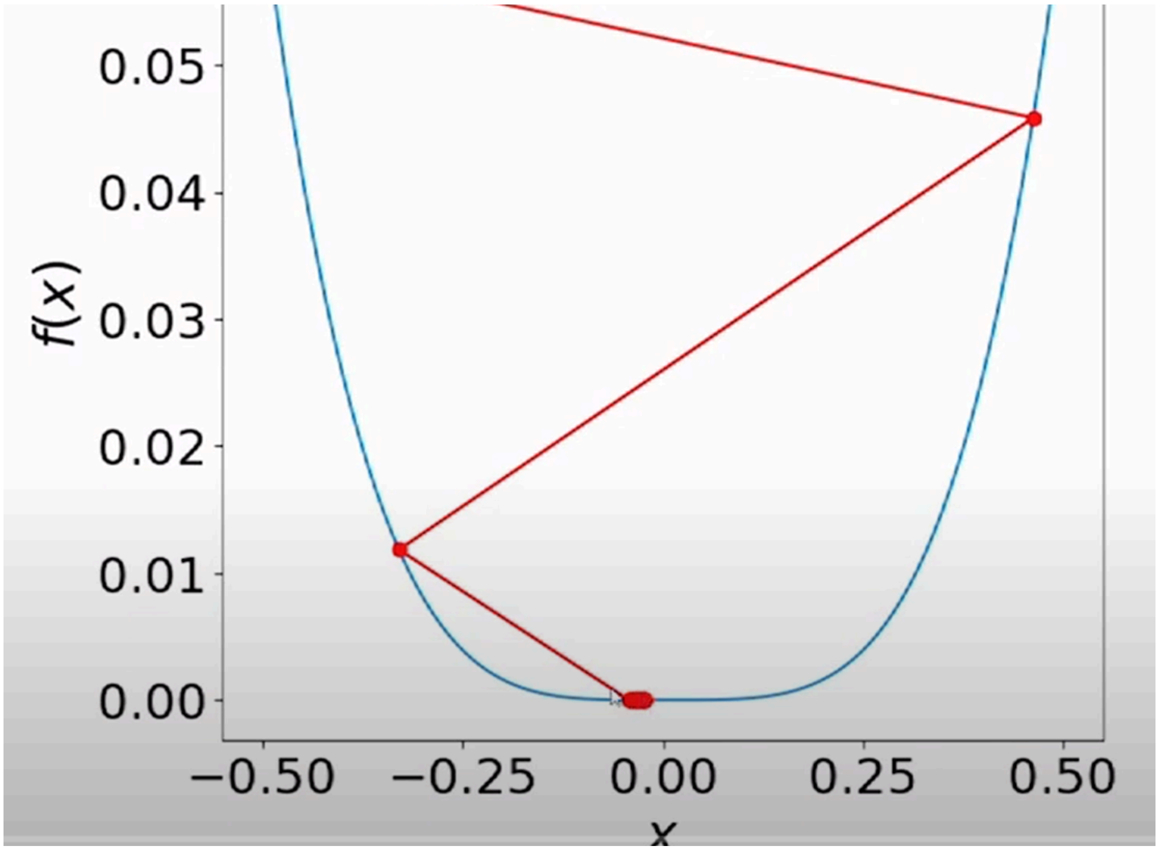
So, in the last class we looked at different optimization algorithms, we looked at gradient descent, heavy ball and the accelerated gradient descent right. So, let us look at an application, a simple application where we are trying to minimize this function f of x which is x to the 4. using the two methods gradient descent and the heavy ball and then we are going to look at the effect of adding momentum to the algorithm and how it sort of translates to different or the accelerated convergent behavior. So, what is the minimum of this function x to the 4? 0 right and compared to x square what is the difference between x to the 4 and x square in terms of convergence to the optimal solution both have 0 being the optimal solution right. So, in x to the 4 close like in the interval minus 1 to 1 you see that the function kind of like decays more slowly than x square right. So, at least in that interval, it may take more effort for the algorithm to converge to the closer to the optimal solution right.

Minimizing $f(x) = x^4$ ¶

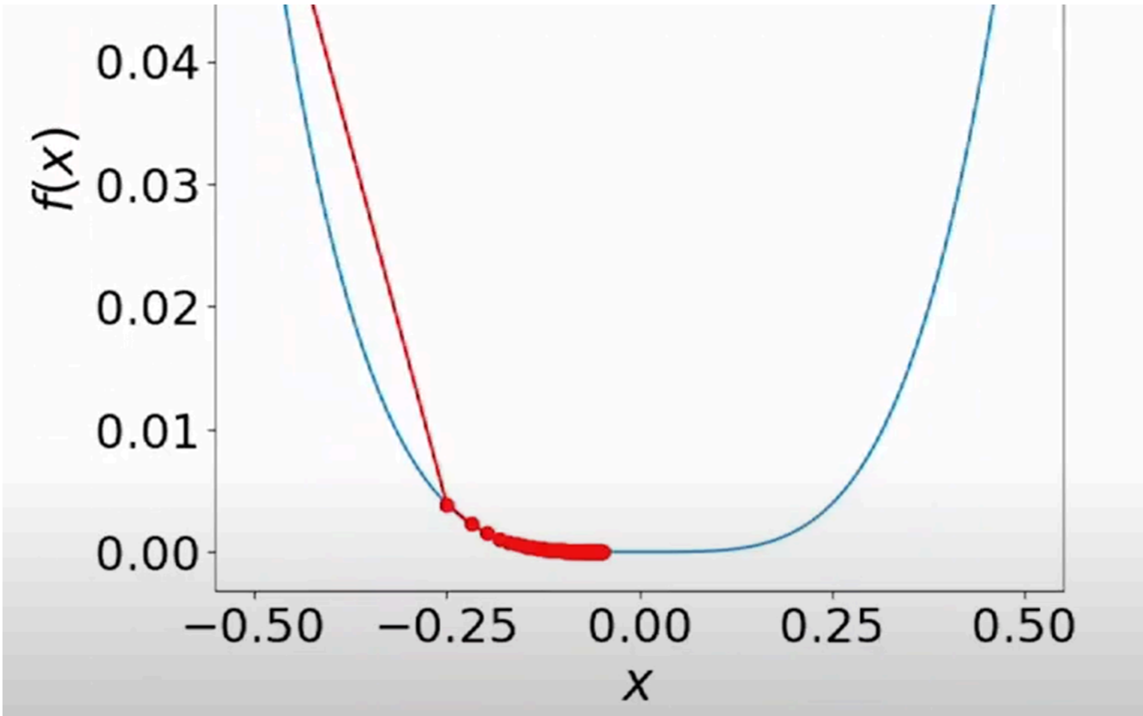
So, let us run this one. So, we are going to be implementing gradient descent first and this is the initial condition. So, x naught basically x naught is negative 0.5.

So, this is the learning rate and learning rate I mean in this case you are going in fact this is the theoretical like closer to the theoretical maximum which is the learning rate of 2 beyond which it you will see that it will start diverging. So, we have chosen a learning rate very close to the theoretical upper bound on the learning rate and we are going to run this algorithm for 100 iterations just storing the values. So, simple gradient descent. So, x is equal to x minus or x k plus 1 is x k minus. the step size which is alpha here right times the gradient of the function ok.

Is that clear? And we are just going to be storing the values in this basically in this array. So, let us run this and as you can see. So, it kind of starts at negative 0.5 and this is your function the blue thing is your function x to the 4. So, x to the 4 you see it has a flatter sort of valley near the optimal solution right and it bounces off it kind of converges closer to the optimal solution 0, but then it makes very small progresses over here right ok.

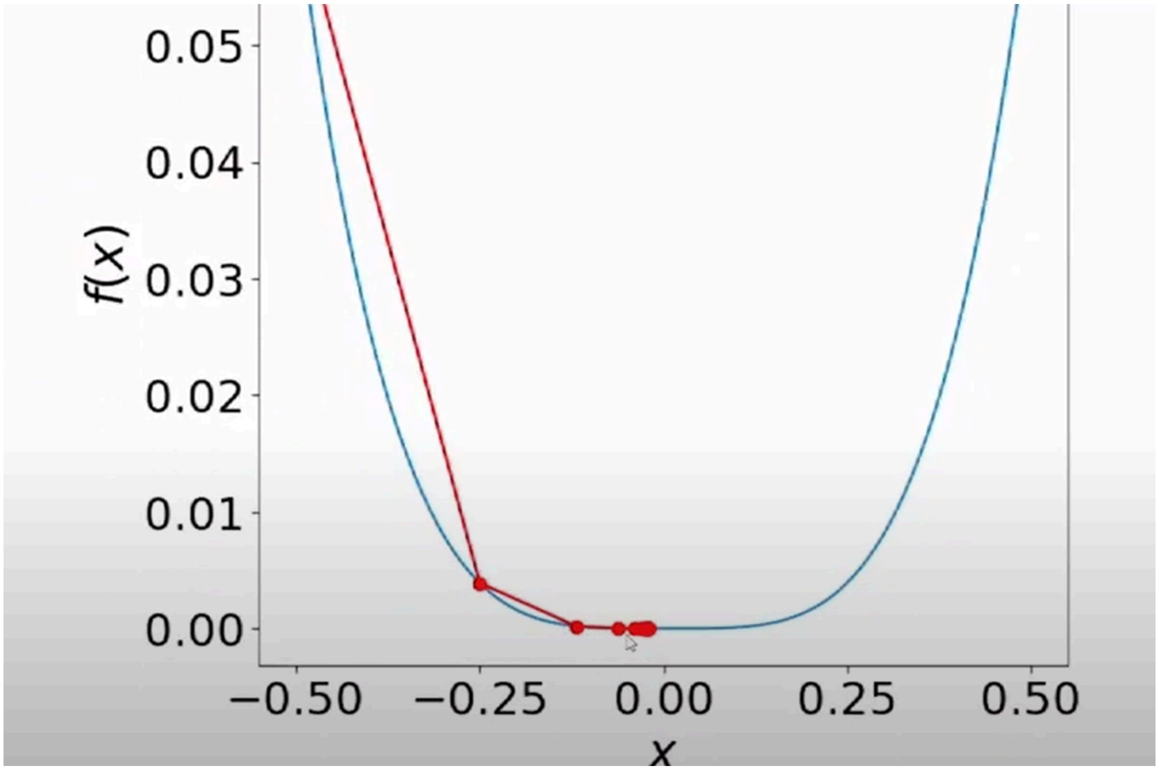


If I increase the learning rate if I decrease the learning rate. So, what do you think would happen if I decrease the learning rate? So, you can see with larger learning rate you kind of have this instability right you jump around a lot in like in the beginning, but once you are closer because the gradients have died down then you sort of start converting more smoothly. if I decrease the learning rate then I may I will be making steady progresses towards but then you can see that because the learning rate is already small it takes many more iterations to get to the same like get to almost a similar point whereas here it may be 10 of after 5 or 10 iterations it gets at least this close to the optimal value right. So that is the difference between If you have smaller learning rate it in the beginning it is going to take lot more time, but then at least the convergence would be much smoother than if you have a larger learning rate right. So, this is the difference.

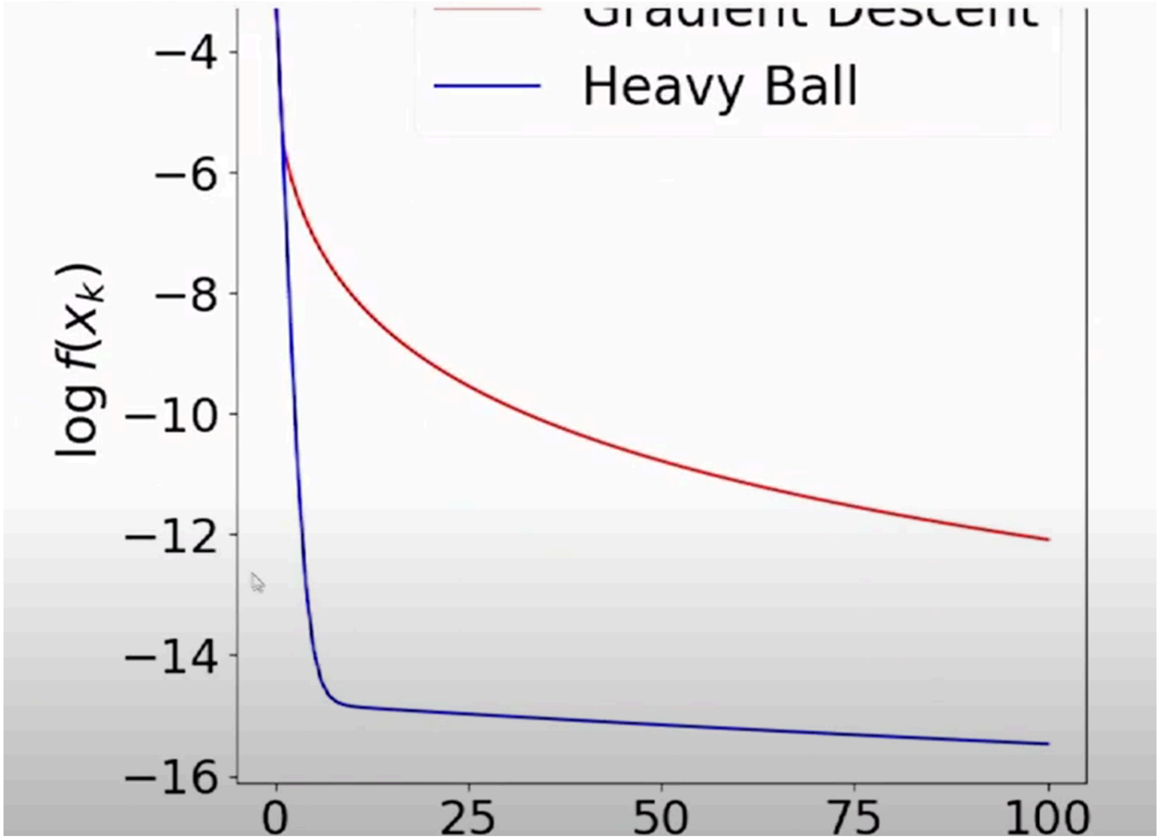


Now in terms of heavy ball. So, what do we do with the heavy ball? So, what does the update look like for the heavy ball and compared to gradient descent? So, we add a beta times x k minus x k minus 1 right. So, that additional momentum term. So, this is this momentum term that we are going to be adding to the the update rule now. So, x is equal to x minus this alpha times the gradient plus beta times this momentum term right and again the learning rate that I am going to be using is 0.

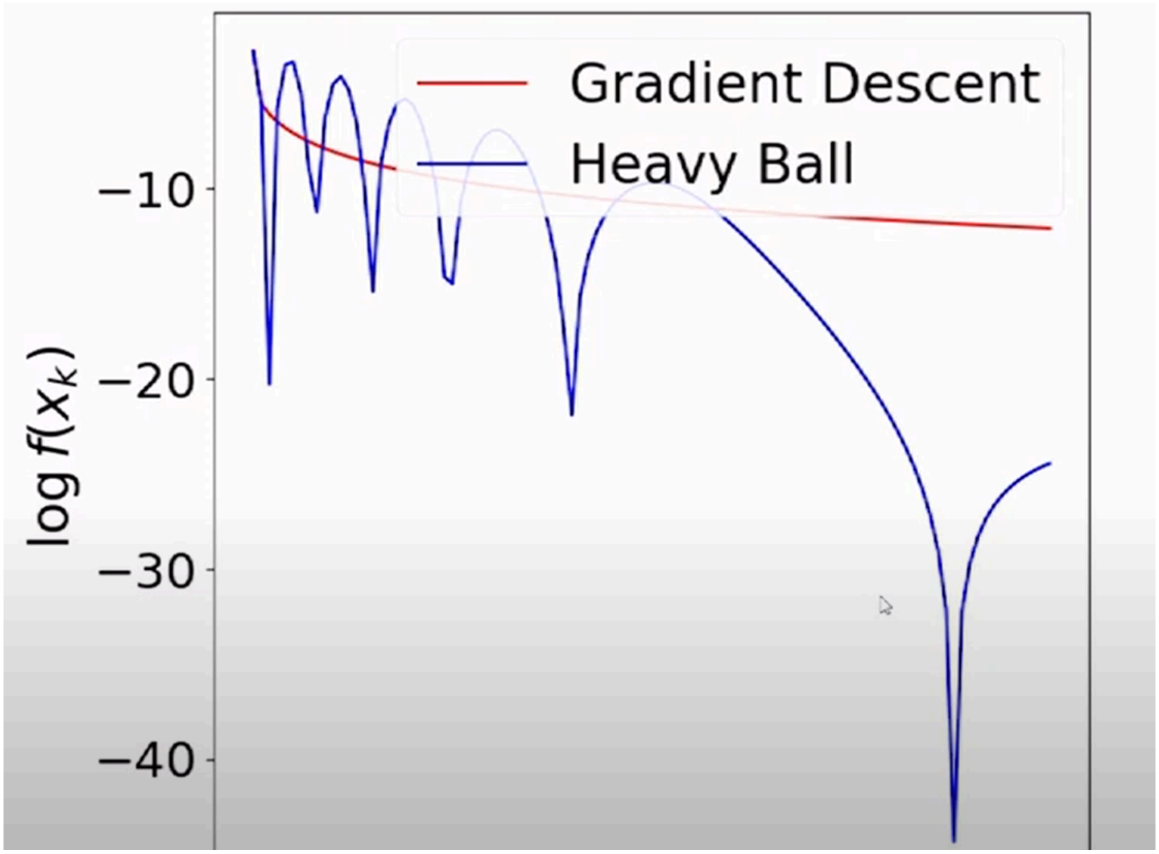
5 and beta like the momentum coefficient I have chosen this to be 0.4 and let us run this one. you can see compared to previous one where alpha equal to 0.5 we converge to this point and then it was making slow progresses towards the optimal solution. Because of this momentum you can see the it is making faster progresses even at the same learning rate right and this also translates.



So, if I plot the log of this particular function because the optimal value we know it is 0. So, the value of f of x k at each iteration k is an indicator how far you are from the optimal solution right. So, if I plot the log of it. So, you can see in almost roughly 10 iterations we are closer to 10 to the negative like in this case it is not log base 10. So, like e to the negative 14 kind of region whereas, compared to.



So, the red curve is the gradient descent you see there is a side like orders of magnitude difference in terms of how quickly it converges just by adding a simple simple momentum term to it right. So, whenever I mean if you choose a suitable value of the momentum term you can sort of make basically make the algorithm converge much faster than the simple gradient descent. So, if I what happens if I increase the momentum coefficient let us say I make it 0.9 So, why not choose larger momentum right yeah. So, it you see I mean it bounces off quite a bit I mean it converges, but it bounces off quite a bit right and if I if I plot it here.



So, you can see I mean it converges to even 10 e to the negative 40, but you can see there are these oscillations, but then you are converging much much closer to the optimal solution compared to your gradient descent ok. So, for like if I hope like let us say for the gradient descent to converge to e to the negative 40 that would take almost forever lot of maybe thousands or lakhs of iterations to get this close with momentum you can still get values which are very close to the optimal solution ok. So, I will be giving a few assignment problems as well that require you to have this kind of coding I will also share this on teams. So, you can also take a look at is that clear. So I haven't implemented accelerated gradient descent method.

Maybe that's something that you can do as part of your homework. Okay. Yeah, if you want to, but then let's say if I'm looking at maybe terminating at 25 iterations, right? I'm worse than the gradient descent. So you also want to have a stable sort of smoother convergence behavior. a lot I mean a very high value may also indicate lot of oscillations around the optimal solution.

So, it is not always recommended to choose a very high value. In fact, you would see that just like you tune the learning rate, you also end up tuning the coefficient for the momentum term. Because again if you choose a very large value it will have lot of oscillation, if I choose a very small value let us say 0.1. So that means, not at not deviating too much from the gradient descent.

So, you can see the plot also looks very similar as with the gradient descent right and if I look at the these two curves there is barely any difference right. So, I mean it is a trade off that for beta being 0.1 yeah very less momentum. I mean still better because I mean you are adding still using momentum, but it is the difference is not so significant. Any questions on this? So, the question is the values of alpha and beta the learning rate and the momentum coefficient does it depend on the function that we are trying to optimize and the answer is yes.

So, it will like for let us say function like x square you would as you would notice I will share this script. So, as you would notice that like if you choose a different function you would have to choose first of all the upper like the maximum value of x alpha and beta that you can work with that or alpha in this case that you can work with that is going to vary with the function. also the suit like the optimal choice of learning rate that is also going to vary and that is going to be dependent. In fact, in certain cases we already saw right like if the function is L-smooth 1 over L is the suitable choice of learning. So, this is something that you are also going to observe ok.

So, you can work with functions like x square L value of L is 2. So, you will see that 0.5 is the suitable choice of learning rate ok. For beta well not so beta is so first so in most cases the momentum coefficient is normalized between 0 and 1.

So, minimum value is 0. So, that means you do not want to use any momentum and technically you can also choose like beta to be more than 1, but it is usually kept routine 0 and 1. So, at least in the context of accelerated gradient method right we saw the way we were adding momentum there. So, all the coefficients were in terms of L and mu. So, in certain cases if you know the L and the mu because for the nonce like for the case when the function is not function is not strongly convex there is no advantage of using heavy ball as we saw in the lectures as well right. So heavy ball is as good as the gradient descent if the function is not strongly convex.

For the L-smooth function, heavy ball has the same kind of, so the role of beta doesn't play that significant a role there, like as in like a suitable choice of beta, but a suitable choice of beta would be important like it would sort of figure in when you start when you also have the function to be strongly convex. So then in that case, I mean, so there's going to be one problem in the homework where you have to optimize on alpha and beta. All right so in the last class we kind of like we concluded that class with the idea of trying with the idea of trying to map optimization algorithms and two dynamical systems and be able to analyze them in continuous time right? So, again just to recap. So, if we have a gradient descent algorithm which looks something like this and by the way that is not the only algorithm that we are going to be analyzing in continuous time, but just to give you and inside what we want to achieve. So, we can write this as we can consider the limiting behavior when the step size eta is very small.

$$* \quad \text{GD:} \quad x_{k+1} = x_k - \eta \nabla F(x_k)$$

$$\lim_{\eta \to 0} \frac{x_{k+1} - x_k}{\eta} = \dot{x}$$

So, this almost looks like x dot and the equivalent dynamical system that we end up getting is x dot is negative of gradient of f of x right and we call this gradient flows. ok and if you have a dynamical system like this. So, an equilibrium of a dynamical system is given by setting. So, what is the characterization of equilibrium? It does not change right if you are at the equilibrium. So, that means setting x dot equal to 0 and that means if I set x dot equal to 0.

$$\lim_{\eta \to 0} \frac{x_{k+1} - x_k}{\eta} = \dot{x} \quad \Big\} \quad \boxed{\dot{x} = -\nabla F(x)}$$

↳ Gradient Flows

So, this would imply that the gradient of f x is also equal to 0 and this is the condition for the optimality right. So, this is the for unconstrained optimization this is the first-order condition for optimality. So, ok. So, in order to be able to analyze these kind of dynamical systems, be able to design dynamical system and I say design dynamical system, maybe I can design something like this, dynamical system which looks something like this. let us say x dot, suppose I design a dynamical system which looks something like this, maybe just in order to for numerical stability just add a little epsilon to it.

$$\dot{x} = -\frac{\nabla f(x)}{\|\nabla f(x)\| + \varepsilon}$$

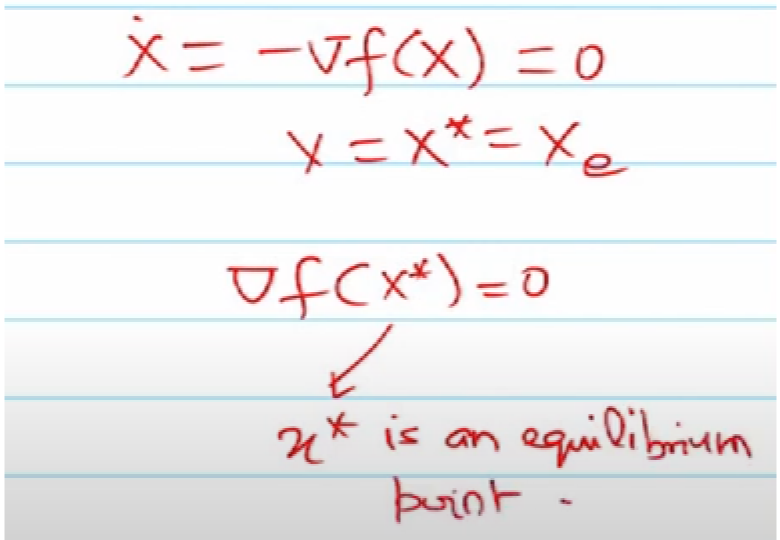
b. $\dot{x} = 0 \Rightarrow \underbrace{\nabla F(X) = 0}_{\text{1st order condition of optimality.}}$

So, that division by 0 is not there, but in I mean we only asymptotically converge to 0 and not to never I mean never exactly at 0, but anyway or the equilibrium. But if I design a dynamical system like this and if I am able to analyze its convergence behavior, so this is going to provide me insights onto an equivalent discrete-time algorithm that we can use, right? So for instance, for this particular dynamical system, let's say we are able to analyze how quickly it converges and so on. This can translate to an optimization algorithm of this form, right? something like this.

$$x_{k+1} = x_k - \eta \frac{\nabla f(x_k)}{\|\nabla f(x_k)\| + \varepsilon}$$

While the same convergence guarantees let us say we get an exponential convergence or asymptotic convergence with for one particular algorithm that same convergence guarantees it may be very difficult to obtain for the discrete or discretized algorithm like this. It still provides you insights and you are able to compare a system like this a dynamical system like this versus a dynamical system like this in at least in continuous time and This also helps you in designing new algorithms by designing new dynamical systems.

So, if maybe in continuous time we can design dynamical system that are much much faster, we can try and come up with an equivalent discrete-time implementation and with the hope that it also works out better there. So, the tools that we are going to be studying today, they are going to be useful in first of all analyzing the existing algorithms, but also in designing newer algorithms which have maybe better convergence guarantees. So, when I say x dot is equal to and this is equal to 0 for some x equal to x star or let us call it x e which is the equilibrium point right. So, that means gradient of f of x star or x e is equal to 0.



So, x star is an equilibrium point. it really comes down to how quickly x converges to x star right. So, what we need to analyze is the stability of these equilibrium. So, this is often a misconception among most students. So, stability is not a property of dynamical system, it is a property of the equilibrium. A single dynamical system can have multiple equilibria, some of them can be stable, some of them can be unstable.

So, stability is a property of the equilibrium and so when someone says this particular dynamical system is stable, that is an incorrect statement to say. The correct statement is this particular dynamical system has this equilibrium which is stable or which is not stable and so on. So, stability is a property of the equilibrium. Okay, stability is a property of equilibrium and not of dynamical system.

Is this clear? Yeah. In this one you mean? Yeah. So, that is a very good question. So, the question is, so this would translate to x dot equal to negative of sin of f of x right or gradient of f of x. So, which is the maximum value or the maximum the magnitude of it is always going to be plus or minus 1 or magnitude is always going to be 1. So, in regions where gradient of f of x is very large why do you want to converge at smaller rate because the magnitude is just one right here.

We would actually want to use a larger gradients and converge faster and then maybe start using it and that is something that we are going to do. So, this is just an example. So, just designing in I mean just giving you an idea that you can design dynamical systems with different kind of stability guarantees. for the corresponding equilibria, but I mean as you correctly pointed out this is not the optimal way to or a good way to go about it. So, we are going to be looking at a slightly different form of this ok.

So, let us first talk about a dynamical system. So, we are going to be denoting a dynamical system by. So, let us say x dot is some g of t and x obviously, x is a function of time. So, x dot x dot t is implied there. So, this kind of system is called time varying system or another name for this particular system is non-autonomous system ok.

* Dynamical system:

$$\dot{x} = g(t,x)$$

→ Time-varying system

→ Non-autonomous system

Whereas, if you have a dynamical system of this form x what is g of x, this type of system is called time-invariant. does not matter when you start the future trajectories would be rolled out as if I mean you start at some other time point right? So, it is time invariant or autonomous system. So, both are equivalent nomenclatures. So, when we talk about equilibrium of this particular dynamical system. So, equilibrium would the definition of equilibrium would be let us say x is the equilibrium.

$$\dot{x} = g(x)$$

→ Time-invariant system

→ Autonomous system

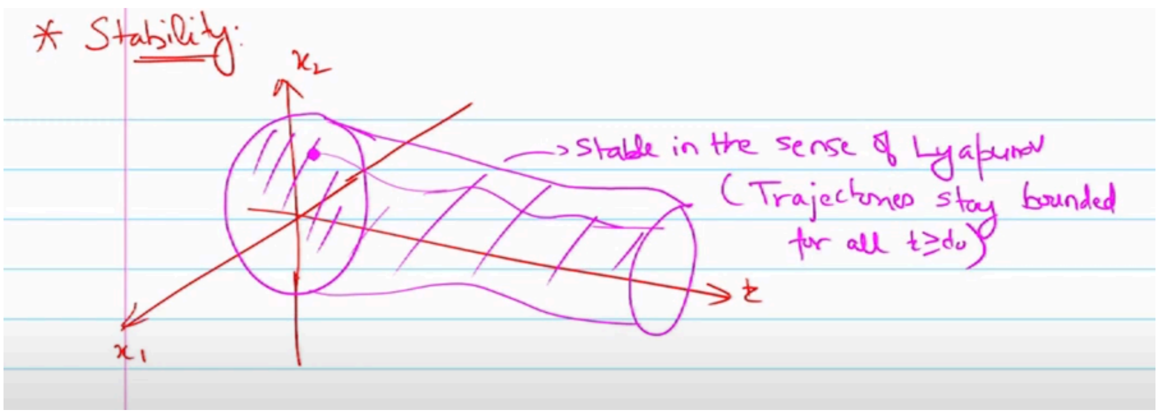
$g(x_e) = 0$ → x_e is the equilibrium

WLOG, we assume $x_e = 0$

So, that means g of x e should be 0 right. So, that is the definition of the equilibrium. You can also have like a set of equilibrium points. Now, you may I mean may not have just one single equilibrium point, you can have a set of equilibrium points right, but then we are saying I mean basically x e is your equilibrium and for the subsequent analysis through this lecture, we are going without loss of generality, we are going to be assuming that this x e is basically equal to 0. That does not mean to say that like if I am minimizing a function x minus 1 whole square, the optimal solution is x equal to 1 right. then so whatever we are going to read for I mean at least study from now on.

So, we just assume that the optimal solution or the equilibrium and so on these are at the origin, but then the same results hold for basically the case when it is not the origin right. So, just makes things easier to makes it easier to analyze these concepts ok. So, what do we understand by stability of an equilibrium? Double derivative what? Double derivative should be negative, why? Like if I if why do you think it is negative? So, that is ok. I mean the intuition is kind of right, but that is not a good way to define stability. So, we look at the more sort of formal definition of what stability is and then eventually we will try basically figure out ways to characterize stability using something called Lyapunov functions.

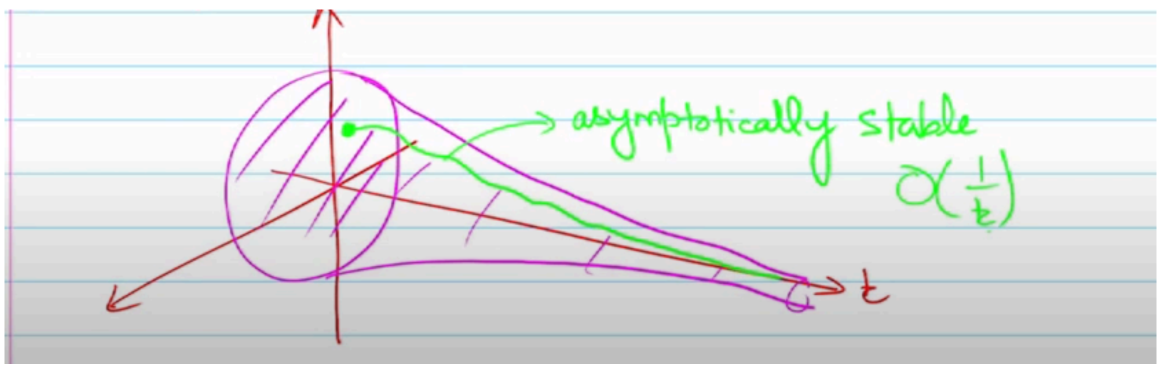
So, let us consider a dynamical system with origin being the equilibrium. So, as I said like we will be just we will just be assuming that origin is the equilibrium and this is your time axis and let us say this dynamical system evolves in 2D space ok, something like this. Now, at t equal to 0, you are going to be starting let us say somewhere in this region consider this point to be somewhere over here, but you are going to be starting in this region. If you can guarantee that for all future times the trajectories. So, when you start and then you evolve the system, the trajectories are going to have some certain form.



So, if you can guarantee that the trajectories for all future times are going to be bounded in a certain tube for all future times not just for over as compact interval, but for all future times they are going to be bounded in this tube. So, this is stable in the sense of lyapunov. So, this is called stable and the idea is trajectories stay bounded for all future times all t greater than equal to t naught. So it doesn't say that the tragic as I said what the equilibrium for this particular dynamical system may be at the origin.

So origin is the equilibria. So ideally I mean a good behavior would have been that it converges to as t goes to infinity it sort of converges to or basically it has a form like this right. That would have been a nicer thing to have. But I mean this by stability I mean think of stability at least at an intuitive level think of stability as boundedness. So, meaning that if I run gradient descent or if I run any optimization algorithm I know that I am not going to go off like it is not going to become divergent it is just that I may also not be converging we do not know ok. So, we run an algorithm we are not going to be going off, but then we it is possible that we may not also converge to the optimal solution.

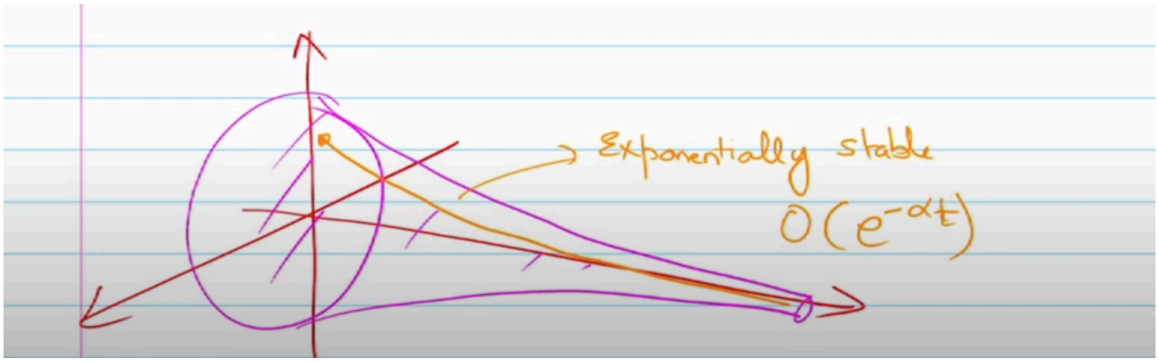
So, that is that is stable in the sense of Lyapunov ok. slightly stronger notion of stability would have been if we can guarantee that as t goes to infinity we also converge to the equilibrium. So, that means if I draw the same picture, if I start anywhere in this disc around the origin or the around the equilibrium as t goes to infinity the size of the tube keeps on shrinking and so on right, and in that case any trajectory which starts somewhere it eventually sort of asymptotically goes to 0 for the equilibrium. So, in that case we say that the equilibrium is asymptotically stable ok. So, what does this mean in the context of optimization algorithm? So, if you run this optimization algorithm as increase the number of iterations you are I mean basically you asymptotically converge to the optimal solution. So, when we talk about rates at which optimization algorithms converge.



So, let us say if I say it converges like order 1 over t. So, this is an example of an asymptotically convergent behavior right. all right. You can have even stronger notion of asymptotic stability.

So, what could that be? Fixed time is one, but. So, one I mean. Exponential. Yeah, exponential right. So, asymptotically saying I mean I can go asymptotically fast with order 1 over t or order 1 over t square t cube and so on right. but then it is it is much better if we converge exponentially fast and not just asymptotically fast, but exponentially fast and that basically relates to exponential stability of the equilibrium.

So, if we can. So, in this case for instance, I mean this may be converging like 1 over t or 1 over t square, but definitely not like or may not converge in the exponentially fast. So, I mean even stronger notion of stability is when you converge when this that size of the tube sort of decreases or the width of the tube decreases exponentially fast and any trajectory starting here would have an exponential kind of behave convergence behavior towards the optimal solution. So, this is exponentially fast or exponentially stable. So, that means, convergence is order let us say e to the negative alpha t kind ok. Why is exponential better than polynomial convergence rates like these? No, so if I expand the c to the negative alpha d right you will get anyway get those terms.



So, at there will be a point beyond which you basically it will take over the one in the polynomial rate right. So, exponential stability or exponential convergence is preferred over a simple asymptotic stability ok. There is a new notion of stability and then we will get more into it in the due course of time which is called fixed time stability which is even faster notion of stability. And it largely makes sense for continuous time dynamical system, but there are few results in discrete domain as well. So, if you have, so if you have a similar disk and regardless of where you start this tube basically shrinks to a zero size in a finite time.

capital T and then the size of the tube sort of stays or the width of the tube stays 0 for all future times. So, the meaning that if I if what regardless of whatever trajectory I start with or whatever initial condition I start with there exists some capital time T after which the solution is just going to stay converged. First of all it converges to the optimal solution or the equilibrium and then it stays converged for all future times. So, this is something called finite time or fixed time stability and I will basically come to different basically we will look at the differentiation between finite and fixed time later, but the idea is there exists a fixed time or there is a. So, T less than infinity right such that for in this case for all t greater than equal to T you have x t equal to 0.

So, this would be the definition of or the notion of finite or fixed time stability. So, the difference between finite and fixed time here is if this capital T is dependent on the initial condition x naught So, then it becomes finite time stability if it is independent of where you start x naught there I mean no matter where you start you are guaranteed to converge in a fixed amount of time which is upper bounded by this capital T then it becomes fixed time stability and we are going to look at ways design algorithms which are actually finite time or fixed time stable and so on. So, you can also mathematically capture these notions of stability. So, let us look at these one by one.

So, the first thing is simple stability or Lyapunov stable. So, in this case we say origin is stable in the sense of Lyapunov if for every epsilon greater than 0 there exists delta. which is a function of t naught when you start and epsilon such that whenever x t naught is less than delta this implies that x t is going to be less than epsilon for all t greater than equal to t naught ok. So, what does this statement mean? So, think of delta as the size of this initial this ball right. So, if you start anywhere within this delta you are guaranteed to stay bounded within some epsilon ball and this delta depending on what epsilon you want like how closer you want to be to the origin or the to the equilibrium for every delta for every such epsilon there must exist a delta and we call it uniformly stable we call origin to be uniformly stable. So, this is the definition of stability of the equilibrium of stability of origin, but it is going to be uniformly stable if delta is a function of just epsilon.

$\dot{x} = g(x)$ s.t $g(0)=0$

* Lyapunov stable: Origin is stable in the sense of Lyapunov,

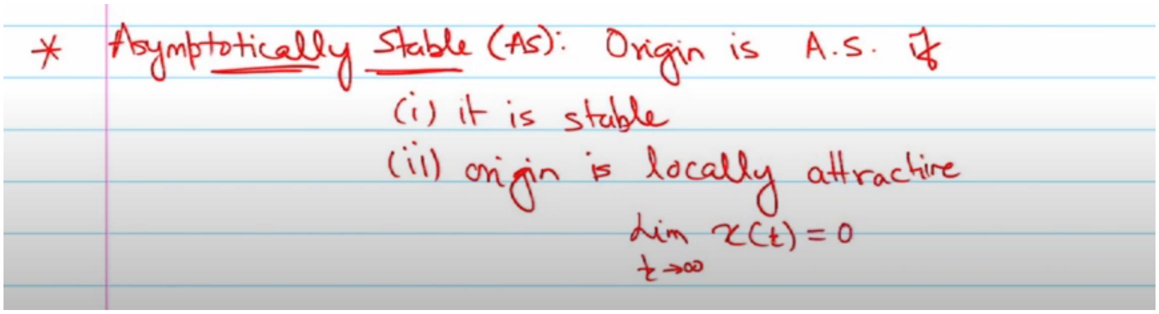
if $\forall \varepsilon>0 \; \exists \, \delta(t_0,\varepsilon)$ s.t

$$\|x(t_0)\|<\delta \implies \|x(t)\|<\varepsilon \quad \forall t\geq t_0$$

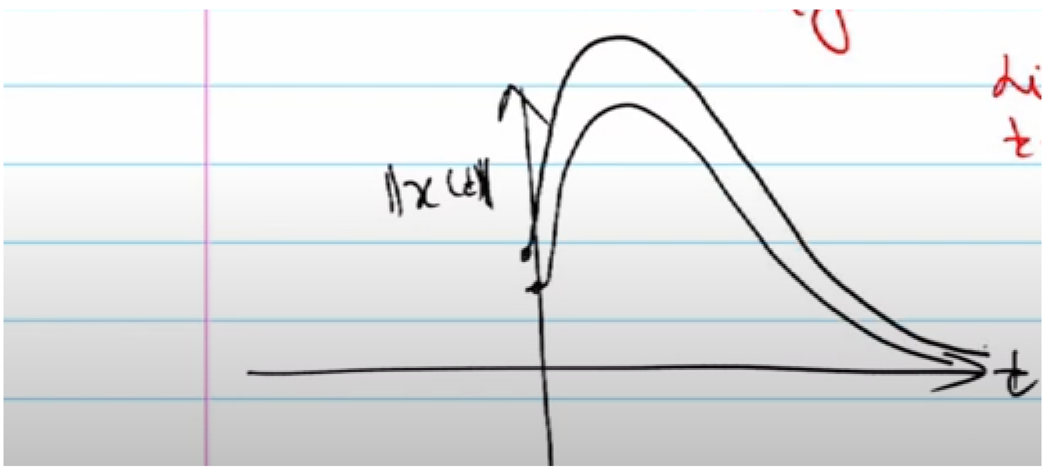
Uniformally stable if $\delta(\varepsilon)$.

So, it is independent of T naught then it is uniformly stable. So, by the way for all of this analysis we are going to be considering a dynamical system like this. x dot is g t of x or g of x, you can consider autonomous. Let us just for the sake of simplicity, let us just consider it an autonomous dynamical system such that origin is the equilibrium. Is this definition clear? So, the next notion of stability was asymptotic stability. So, when is the equilibrium asymptotically stable? If first of all it is stable and second is origin is or let us say you talk about local stability.

So, it is locally attractive. So, attractive. So, then it is locally asymptotically stable. So, what do we mean by attractivity? Yeah. So, that means as t goes to infinity. So, that is the definition of attractivity.



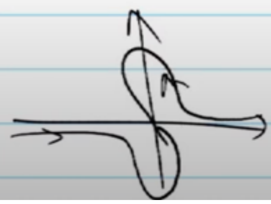
Why cannot we just have second as the criterion and not include one. So that is right and so for instance a dynamical system whose behavior looks something like this for instance if I run this let us say I run on this y axis I am plotting the norm of x t and on time and x axis is time so it is possible that maybe you start somewhere over here. So, instead of so, you basically you do something like this and if I start somewhere over here I it goes something like this, something like this and so on right. So, uniform stability is if delta is not a function of t naught. So, it is uniformly for all time t ok. So, you can have a system like this and this would be an example of a system which is.



attractive, but not stable. So, this is an example of an equilibrium which is attractive, but not stable ok. So, if you have heard of something called Vinograd system. So why is let us say why is this kind of behavior not desired in general if eventually t goes to infinity that if everything converges, why is this behavior not desired? Yeah, so if you let us say if you just run it for 10 iterations or 50 iterations or 100 iterations right, you would want to be closer to the value like optimal solution then be farther away from it right. So you want to I mean ideally you would not want to design a dynamical system.

equilibrium is just locally attractive or attractive, but it is not stable ok. So, Vinograd system let me just write this down. So, Vinograd system exhibits this kind of behavior. So, it is essentially the trajectories are if you look at the phase portrait of Vinograd system, it is something that you can try it out. Let me just write x 1 That is the dynamics of x 1 and x 2 dot in the same term in the denominator. So, you can so you can show that the equilibrium or the 0 0 or the origin in this case it is locally or it is attractive, but it is not stable ok.

Ex: Vinograd system



$$\begin{cases} \dot{x}_1 = \dfrac{x_1^2(x_2-x_1)+x_2^5}{(x_1^2+x_2^2)(1+(x_1^2+x_2^2)^2)} \\ \dot{x}_2 = \dfrac{x_2^2(x_2-2x_1)}{(x_1^2+x_2^2)(1+(x_1^2+x_2^2)^2)} \end{cases}$$

So, I advise you to just look at the plot the phase portrait of it and you will get a sense. So, the phase portrait looks something like this. So, kind of looks so it converges, but then it is like some it looks something like this So, the next notion of stability was exponentially stable and the origin is exponentially stable if you can find coefficients or m and alpha such that both greater than 0. such that this holds for every t ok.

* Exponentially stable: If $\|x(t)\| \le m e^{-\alpha(t-t_0)} \quad \forall t \ge t_0$

$m, \alpha > 0$

So, this would be the definition of exponential stability of the equilibrium ok. So, now given a dynamical system first of all we know how to find the equilibrium points of it right by just equating the vector field to 0 or the right-hand side to 0 you would be able to obtain the set of all equilibrium points. But then how do we figure out whether a particular equilibrium point is stable or asymptotically stable or exponentially stable? Is there an approach for this? Well eigenvalues would make sense if you let us say linearize it and then talk about the locally, I mean so that linearization test is not that conclusive. if it is stable then stable, if it is unstable it is unstable, but if it is marginally stable then you cannot say and so there are some issues with that. So, you can use something called Lyapunov's direct method using something called Lyapunov functions.

So, this particular Lyapunov method for determining stability of equilibria. I think that was part of his master's thesis. So, if any of you here is a master student you are in too late. So, this has been a seminal contribution in the field of stability theory where Lyapunov I mean we basically came up with a way to characterize notions of characterizing different notions of stability using something called Lyapunov function. So, for folks who know what Lyapunov functions are intuitively what do they capture? energy of the system right.

So, let us say you have a spring mass damper system or a simple pendulum system right. So, if you have a pendulum which is which basically hangs this I mean in a upside down. So, we know that this particular position is going to be its equilibrium position. So, theta equal to 0 that would be like if this is if this angle is your theta. So, theta equal to 0 is the equilibrium position. And from our I mean observations we know that this kind of like equilibrium is stable right, because if you give it a tap eventually it sort of basically arrives at its equilibrium position.

Is there a way to sort of I mean intuitively argue how we can arrive at this, arrive at this behavior. So, when the pendulum is swinging right, it has some energy kinetic energy or the potential energy and that kind of eventually decays and because of the decay of energy or no growth in the energy like further no growth in the energy. it kind of arrives at the equilibrium. If let us say the energy does not decay then at least and if it does not increase then you know that it is just going to be oscillating all the time, but it is not going to be in a resonating kind of mode where you. So, resonance happens when you like tap the system with the same frequency as the natural frequency of the system right.

So, if you are not providing any external force it will retain its energy it will never be I mean decay, but then it would not also go unbounded. So, that would be stability. if it decays eventually that would be asymptotic stability or exponential stability of the equilibrium. But then the idea is if I can somehow capture the energy of the system, I should be and I can talk about the rate at which those in that energy sort of dissipates, I should be able to get a sense of whether it is a stable equilibrium or asymptotically stable equilibrium or exponentially stable equilibrium and so on right.

and that particular notion of energy is what Lyapunov functions sort of capture. So, Lyapunov functions are like proxies of the energy of a system. It may be equal to the physical energy. So, like for instance spring mass tamper case half kx square plus half like mx dot square So, that can be like that is the physical energy of the system. So, sometimes you can choose those to be Lyapunov functions and you show that it decays and therefore it stays stable or sometimes you have to engineer those Lyapunov functions and for the larger part of his PhD work Lyapunov tried to come up with a way.

So, where he could like we are given let us say I know I give you a dynamical system. and let us say it is exponentially like I mean the equilibrium is exponentially stable. How to come up with the dynamical like how to come up with the Lyapunov function through which we can show that it is exponentially stable or asymptotically stable and things like that. So, is there a converse way to come up with the Lyapunov function given the equilibrium stability of the equilibrium. he at least I mean try to come up with a way through which you can construct these Lyapunov functions and unfortunately it is still an open problem. So, we do not know if I mean if I given the stability of the equilibria can we come up with a can we always come up with a Lyapunov function analytically that would guarantee that. Thank you very much.