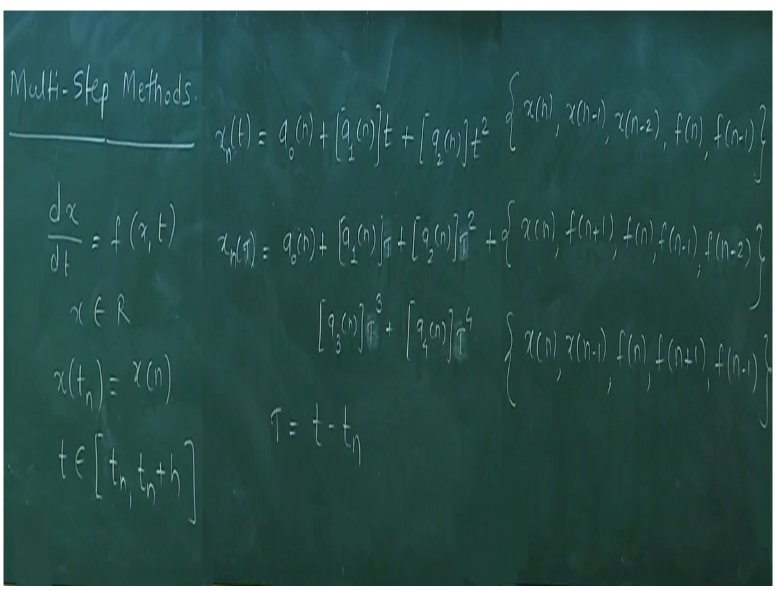
Advanced Numerical Analysis
Prof. Sachin Patwardhan
Department of Chemical Engineering
Indian Institute of Technology - Bombay

Lecture - 43
Solving ODE-IVPs: Generalized Formulation of Multi-Step Methods

So we have been looking at multi-step methods, and yesterday I derived a specific result for fitting a local quadratic interpolation polynomial, and I showed that it could be done in different ways. So continuing our introduction with multi-step methods, I am going to derive today, a general formulation for multi-step methods for fitting any polynomials. But before that I want to derive or point at 1 pattern, I am going to change the notation a little bit.

So I will upload my notes with the changed notation, just to emphasize the fact that we are fitting a polynomial at each time instant I am going to, well I already had that notation I am going to just shift the index n to slightly. So time index is going to be emphasized now with new notation, otherwise there is not much difference between the notations that we have used earlier, it is only the placement of this particular point.

(Refer Slide Time: 01:48)



So we have been looking at this multi-step methods, we have this differential equation dx/dt =f of x, t, where x belongs to R, we are looking at a scalar differential equation. And our problem is to move from, this is my initial condition I want to integrate over time, I want to integrate this initial value problem from tn to tn+ h, where is the fixed integration step size. So we were looking at this polynomial interpolation.

And now let me well yesterday we looked at a specific polynomial interpolation we looked at x t or xn t was a0 n+ I am just moving this subscript n to time index n just to emphasize that these are time varying coefficients, this is not one polynomial at each time point we are fitting or we are having a different interpolation polynomial. So it is a polynomial with the time varying coefficients that is another way of looking at it, we are not fitting 1 polynomial.

Why quadratic? Well I can choose to fit an interpolation polynomial of higher order it is no trouble, I can instead of doing this I can choose xn t=a0 n+a1 n t+a2 n t square+, where I can fit a higher order or I can develop a higher order interpolation polynomial okay, and then either extrapolate or estimate xn+1. Ultimately, what I want to do is move to xn+1 okay, now to do this how many unknowns are there in this polynomial there are 5 unknowns 1 2 3 4 and 5 right.

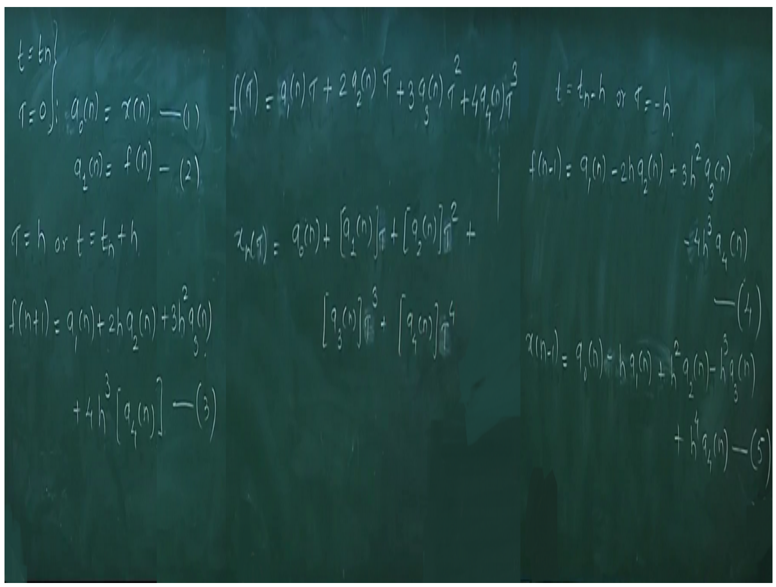
There are 5 unknowns and then I need to get interpolation polynomial coefficients, I need to have 5 equations right, I need to have 5 equations so if I write 5 equations in 5 unknowns then, we made one modification yesterday we just shifted the time scale. So if you define a new time scale tau which is t-tn, if I define the new time scale tau which is t-tn then this becomes the polynomial in the shifted time scale tau, this will be tau, this will be tau square, this will be tau cube and this will be tau to the power 4.

If I use a shifted time scale okay, now to generate 5 equations in 5 unknowns, I can do it in many ways for example I can do this using say x n, x n-1, x n-2, f n okay, and I can decide to do this okay. I can decide to do x n, I can decide to use 4 derivatives I x n okay, it is up to me how to fit, how to generate the interpolation coefficients, I could use derivative terms, I could use okay. I can use x n, x n-1, f n, f n+1, I could use any of these sets okay any one of them will give me a valid way of constructing coefficients of the interpolation polynomial.

If I use this set or if I use this set or if I use this set, I am not going to get identical polynomial interpolation polynomials, polynomials will be different okay. What I want to indicate here is that there is no unique choice by which you developed this okay, it is up to you which way you want to go okay. Suppose, I decide to using this set okay, then well I have 2 equations or I have 5 equations, now we have this we can differentiate this with respect to tau.

And you know that will give us equation which is one more equation okay, and then using f and using x okay, I can write 5 equations in 5 unknowns okay, if I happen to do it for this set, let us see what are those equations and I will just write down the final solution which you get after that okay. Algebraic manipulation are fairly simple and you can do them by just doing elimination of variables.

(Refer Slide Time: 09:23)



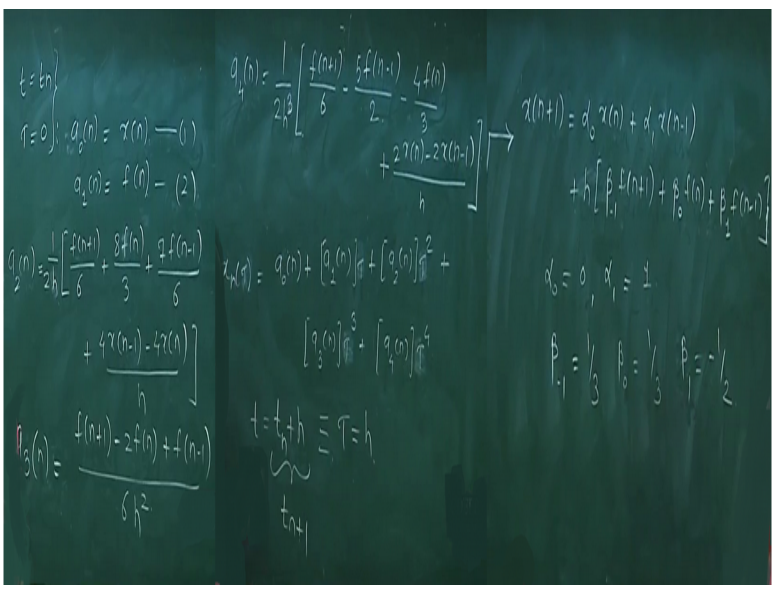
So if I actually go ahead and then at tau =0 setting tau =0, I get this x a0 n comes out to be x n and a1 n comes out to be f n by following the same method which we have used earlier okay, so I substitute for tau. So what I do here is that at tau=0 is nothing but t=tn, at t=tn is same as tau=0 okay, at tau=h or t=tn+ h, we get f n+1= a1 n+2 h a2 n+3 h square a3 n+4 h cube a4 n okay, I am just writing f n+ 1 by setting tau=h which is same as t=tn+ h okay.

This is the first derivative okay, I am just taking the derivative and substituted, see if I take derivative of this you can write f tau= this is my expression for f tau **“Professor - student conversation starts”** yeah, (()) (11:51) no, it should be f n+1, I am just taking f n+1, I want to use f n+1, f n-1, f n I have already used. **“Professor - student conversation ends.”** See for f n okay f n that is tau 0 okay tau 0 is f n, so this is 0 sorry this should not be tau, this is so this is 0, this is 0, this is 0, only a1 n will remain.

So we will get f n =, tau 0=f n= a1 n, so I have already used f n, I am now using fn+1, so this is my equation number 1 will give me this, 2 is this, my equation 3 is this okay, I need 2 more equations okay. So 2 more equations can be, so I am going to generate 2 more equations that is t=tn-h or tau=-h for this point I get f n-1=a1 n-2, this my equation 4 f n-1 okay. And the last equation I am going to write is x n-1, so this is my fifth equation x n-1, f n-1, f n+1, x n and f n.

I am using 5 equations okay, and I can solve them simultaneously, actually a0 n and a1 n we get directly from the first 2 equations okay, then what remains is finding remaining 3 that is a2, a3 and a4 okay, which you can find out by simple elimination okay. If you do an elimination with these as unknowns okay, and h square, h cube or 3 h square-4 h cube these as coefficients okay, then you can arrive at the solution. I will directly write the solution; the solution looks little more complicated okay.

(Refer Slide Time: 15:25)



So my a0 n, a1 n I already got here okay, my if I do the algebra then my a2 n this turns out to be 1/2 h f n+1/6 okay, this is my a2 n and my a3 n it turns out to be f n. So this is a0, a1, a2, a3 just eliminate, and then you have 5 equations in 5 unknowns out of which 2 are directly the coefficients, so what remains is 3 equations in 3 unknowns, if you do elimination you will get this simple elimination and then find out okay.

And my last coefficient here that is a4 n this turns out to be okay, so finally this is my polynomial a0 n, a1 n, a2 n, a3 n, a4 n and if I substitute that here that is my polynomial okay. I have found out the local interpolation polynomial using 5 equations in 5 unknowns, this is one way of obtaining the coefficients okay, after having found this polynomial see where I want to reach? I want to reach x n+1. So x n+1 is t=tn+ h so which corresponds to tau=h.

If I substitute tau =h I will get value at tn+1, tn+1 this is nothing but tn+1 this is my next point okay. So what I can do now is I can substitute all these coefficients into this polynomial okay, then substitute for tau=h okay, and then I can regroup all the terms okay, they will collapse into a very simple equation, if you look here what all things are appearing in my expression on the right hand side. If I substitute for a0, a1, a2, a3, a4, if I substitute I will get f n+1, f n-1, f n, x n and x n-1 these 5 things I will get okay.

So let me write after substituting and regrouping okay, I can write a very convenient form which is good for computations, this particular form which I have written is not very computation friendly okay. I am going to arrive at a simpler form, so I am going to transform this into a simple formula x n+ 1= alpha 0 x n+ alpha 1 x n-1 okay+ h times beta-1 this is just a notation beta-1 is just a notation, f n+ 1+ beta 0 f n+ beta 1 f n-1, I can rearrange this like this okay.

I can rearrange that polynomial after substituting for tau=h, I can rearrange it into this nice form okay, if I actually do substitutions do all the rearrangement take coefficients of the similar terms together okay, then it collapses to a very simple formula alpha 0=0, alpha 1=1, beta-1=1/3, beta 0=1/3 and beta 1=-1/2. If I substitute and rearrange okay, I get this I would call this computable form, is everyone with me on this what I am doing.

See I am fitting local interpolation polynomial, local interpolation polynomial has been constructed using these values x n, x n-1, f n, f n-1, f n+1 okay, I found the polynomial I substitute it tau=h which gives me x n+1 okay. And I am just rearranging and writing in a form which is more convenient for programming okay, there is more convenient for programming there is nothing wrong if you maintain this complex form okay nothing wrong.

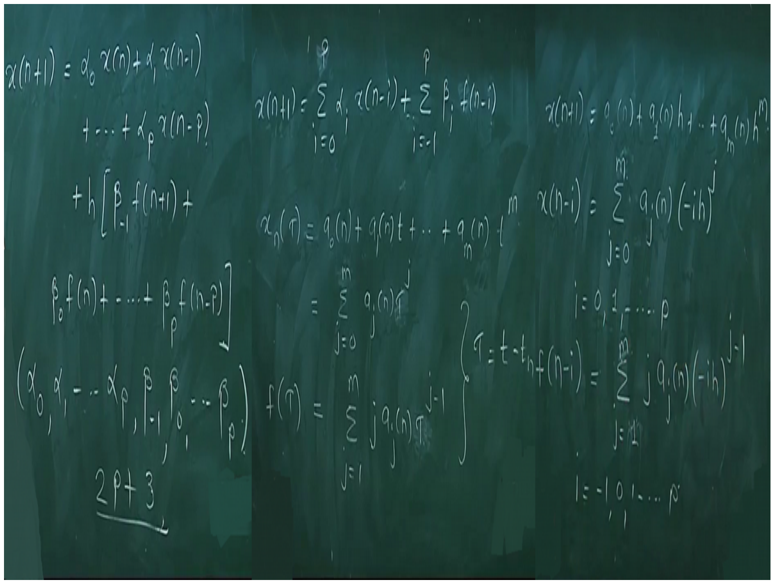
Fundamentally, nothing wrong if you maintain this complex form and every time substitute and find out, what you are going to get ultimately is same as this okay. In fact, all these time varying coefficients seems to do not show up when we write it like this, it appears that x n+1 is some linear combination of x n and past x values, f n+1, f n and past derivative values, all that is appearing is current past and future derivative value, and current and past x values okay.

So my final form is going to always whichever you know if I fit a 7th order polynomial, 8th order polynomial, 10th order polynomial my final form will look something like this x n, x n-1 maybe I have to take x n-2, x n-3, f n-1, f n-2 because I need to estimate all those coefficients. So I need so many other equations, so my final form is going to look always like this. When I work with this form directly instead of doing this complex interpolation polynomial derivations okay.

That simplifies the derivation unfortunately when you do it that way it sometimes you know the polynomial form does not become visible, you just see this but when you derive the formula what you are actually doing is fitting an interpolation formula locally using either derivatives or x values okay. Now let us derive generic expression for fitting any mth order polynomial okay, and then we will look at, what is this just before we go ahead.

What is this particular equation? Is this an explicit or implicit formulation, it is an implicit formulation because f n+1 appears on the right hand side which is function of x n+1, so to solve this you have to do iterations and solve it iteratively okay. Let us move on to more general formulations okay, so I can generalize what we have got I can say that whenever I am going to derive multi-step method. In fact, if you write it in this form the word multi-step becomes more clear why we are calling it multi-step method.

(Refer Slide Time: 25:49)



I can write a general formula for a multi-step method, which is alpha 0 x n+ alpha 1 x n-1 alpha p x n-p+ h times beta-1 f n+1+ beta 0 f n, I can write a general formula for multi-step method okay which is linear combination of x n, x n-1, x n-2, x n-3 whatever up to p steps in the past, and f n+1, f n-1 up to p derivatives in the past okay. So we are finding out the new value okay not just using local derivative we are basing our information based on past derivatives, we are basing on information past x values okay.

So the new value is constructed using some kind of local past information okay, how much you want to go in past? How many steps in the past okay, we will decide well but the accuracy of the method and so on okay. Now how do you come up with a formula that will, how do you estimate these coefficients? What are the unknowns here? Let us list the unknowns, unknowns are alpha 0, alpha 1 to alpha p, beta-1, beta 0, beta p, how many unknowns are there? 2 p+3 unknowns are there right.

So we need to solve this we somehow at to construct 2 p+3 equations and okay. Now in a more compact form I am going to write x n+1=summation i going from 0 to p alpha i x n-i+ summation I going from -1 to p beta i f n-i, in fact I write it like this this -1 beta-1 unusual notation will become clear why I am using beta-1, this is a compact way of expressing the same thing okay. Now what I need to do is of course I am going to fit local polynomial okay.

My polynomial is going to be xn t, let us work with shifted time scale since that tau is 0 and the same idea which have done earlier okay. So my local polynomial is going to be a0 n+, let us consider mth order polynomial in general okay, let me consider mth order polynomial and p step method okay. I want to generalize this which I can write as summation j going from 0 to m aj n tau to the power j right, this is aj n, n is only for the local coefficient and this is my okay.

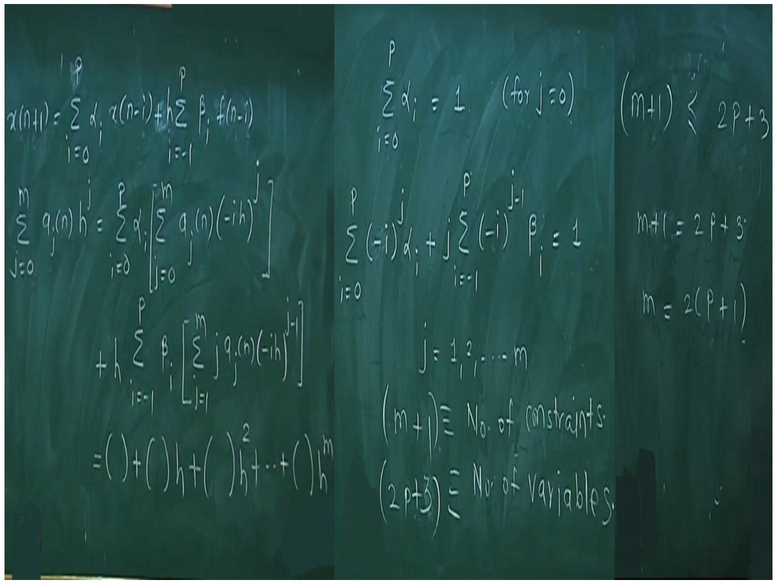
What is f tau? f tau will be summation j going from 1 because initial term will be 0 d/dt of this will be 0, so j going from 1 to m okay, it will be j times aj n tau to the power j-1 right, is everyone with me on this, tau yes it should be tau, it should be the shifted time scale tau. So where our tau is just to remind you tau is t-tn okay, so this is my f tau, this is my x tau okay. Now using our notation okay let us write general expression.

What is x n+1? x n=1 this will be a0 n+ a1 n times h up to am n h to the power m, correct this is x n+1. What is x n-i? i times in the past, how will you write this? It will be i h time will be i h rather -i h, time will be -i h okay. So I am going to write this as j going from 0 to n okay aj n -i h raise to j fine, I am just substituting for ith instant in the past, j going to 0 to m not to n okay. Now this I can do for i=0, 1 up to p, I can write this equation for each you know x n, x n-1, x n-2 x n-3 up to x n-p right.

So how many such equations are there p+1 equations are there okay. Then what about f n-i, we just use that expression, summation j going from -1 to m okay j times j= sorry j=1 to m, p=-1 to sorry i goes from -1 to m, i goes from-1, -1 means f n+ 1 okay, if I put i=-1 I get f n+ 1, I put i= 0 I will get f n, I will get i=1 I will get f n-1 okay. Because we are taking derivative at current point, next point and previous point and so on.

So how many of these equations are there? p+2 okay, I have p+2 equations and okay. So what I am going to do now is I am just going to substitute these expressions x n-i, f n-i these summation expressions okay, I am going to substitute them here okay, and then I am going to substitute this expression here, first one on the right side this will be on the right side, these expressions will be substituted here x n-i, f n-i okay.

(Refer Slide Time: 37:45)



There is one small correction, this expression here has h times, h which I forgot to write when I wrote the generic expression there is an iteration interval h will come here, so this is h times summation i going from -1 to p beta i okay. I have just substituting this here, after that what I am going to do is on the right hand side, we just now this looks very complex equation right, but the way I want to simplify this is like this.

See look on the left hand side you have h, h square, h cube up to h to the power m okay, on the right hand side also you will get if you look carefully you will get maximum power of h will be h to the power m okay, you will have h to the power 0, h to the power 1, h to the power 2, h to the power 3 okay. So what I am going to do now okay I am going to arrange this rearrange this, in between step I am just leaving it for you to do the algebra, I will just explain this qualitatively.

I am going to explain I am going to rewrite this complex right hand side as something+ something * h+ something * h square up to okay, I can do this because there are terms on the left hand side and right hand side, which do not exceed h to power m okay. So I can take all the terms together okay and then if I want to get this equation for any h, what should happen? I should match the coefficients okay.

So the first 0th coefficient should be matched with the 0th efficient here and so on okay, if I do that business of matching the coefficients okay, I get following set of equations, if I just equate the coefficients okay one by one I get this, how many equations I am expected to get? m+1 equations, because how many see we have put mth order polynomial okay, so I have m+1. How many equations I get here? These are m+1 equations right how many variables I have 2 p+3 okay.

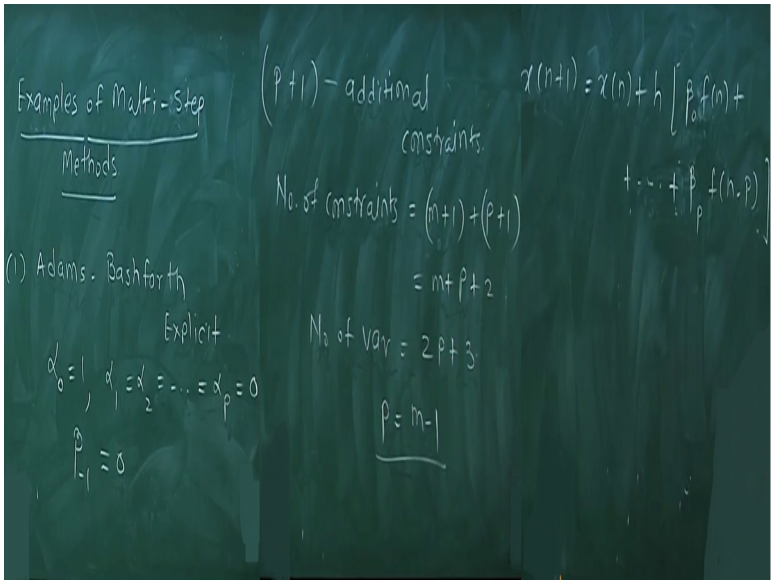
Number of constraints I got is m+1 okay, number of constraints is m+1 okay, and 2 p+3 corresponds to the number of total number of variables, what are the variables alpha 0, alpha 1, alpha p, and beta-1, beta 0, beta 1, to beta p, these are the variables. And these are the equations that relate them okay, by taking general multi-step methods these are the equations that relate them. So these are called as exactness constraints.

So well what you can guess is that you should have number of equations at least=number of unknowns, if number of unknowns is more not a problem you can fix some of them arbitrarily okay, if the number of unknowns are more than the number of equations you have to choose some arbitrarily okay. And but at least to get some tangible solution you should have this= this okay, so I can generally give a condition to get a solution.

We should have m+1 should be <=2 p+3, if number of unknowns is more we can fix some of them arbitrarily, you can set something to 0 or whatever okay, otherwise they should be equal, this can be greater or this can be equal. And what is the condition for equality to hold? Okay if you want equality to hold, then m+1 should be 2 p+3 okay or order should be the polynomial order should be chosen okay.

If you want a p step method okay and if you want to get exact number of constraints the polynomial orders should be chosen, suppose I want one step method if p=1 okay, then m should be 4 and so on, so just gives you a way of calculating what is the order of polynomial that you need to fit okay. So this is the generic derivation of multi-step methods, now there are different classes of multi-step methods which are popular in the literature, we will just have a look at them.

(Refer Slide Time: 47:11)



One class of algorithms are called as Adams-Bashforth multi-step algorithms okay, so this is a class of algorithms okay, you choose parameters in a particular way okay, and together with the exactness constraints we arrived at the polynomial coefficients, and then get one method of integration okay. First of all you should realize that what we have derived now is not one integration method okay.

What we have derived is a way of arriving at integration methods choosing different number of different p okay will give you one particular method okay, if you choose 5 steps in the past you will get one particular method, there is no unique way of getting those coefficients okay. So there are enough number of enough degrees of freedom, so that you can have one method by each one can have one method by his own name, if you choose to have that okay.

So some of these are popular methods which are used in the literature, I will just list them but it is not that this is the only way to come up with the multi-step methods. So in this Adams-Bashforth explicit methods, in this methods we only use derivatives we do not use x except one particular x, so we choose alpha 0=1, alpha 1=alpha 2= alpha p=0 okay. In Adams-Bashforth method we are not going to use past x values, we are going to use past only function values okay.

And these are this is an explicit method, which means beta-1=0 okay, I am not going to use the future derivative okay. So if you put these constraints together with the earlier contains how many constraints are here now? p+, this will follow we can right now because the first equation will give you this alpha 0=1, because your summation you know first equation is summation alpha i from 0 to p is 1, so the constraints that you have here are p+1 additional constraints.

So total number of constraints m+1 original constraints we have which I wrote right for j going from 0 to m okay plus these additional p+1 constraints, so this is m+ p+ 2 okay, and total number of variables is okay. So we also impose one more constraint here that p=m-1, this will fall out actually p=m-1, if you put p=m-1 okay you will get total number of equations and total number of constraints. And then you know you can set up those equations for Adams-bashforth. And then finally the algorithm will look like this x n+1= x n+ h times beta 0 f n+ okay.

Finally, you will get beta 0, beta 1, beta 2, beta 3 for the pth order method, if you some of the books will list these coefficients okay or you can derive them by setting up the constraints I have given here the equation, you get a linear algebraic equation okay in 2+ m variables 2 m+1 variables into 2 m+1 unknowns, you can solve them together and then you can get these coefficients, these coefficients should be listed in any one of the books. Likewise, we will talk about implicit methods.

Those implicit methods are called Adams-moulton methods those are implicit methods okay. And then implicit method will be one which requires x n+1 in the future to be estimated. So what you do is you know you tie them up when you are solving the implicit method, you use an explicit method to get the initial guess, and then solve the implicit method iteratively okay, so that is why it is called prediction-correction methods also, you do a prediction using explicit method, use the prediction in the implicit method as a initial guess and do a correction okay.

We will come to this idea next class. There is another class of methods called Gear’s method, Gear’s method is the one in which you do not use past function values, you use past x values, and only one function value, those class of methods are called as gear’s methods. These class of methods are called as Adams-moulton, Adams-bashforth and so on okay. So we will look at these classes which are popular classes, and then we will move on to the next method.