

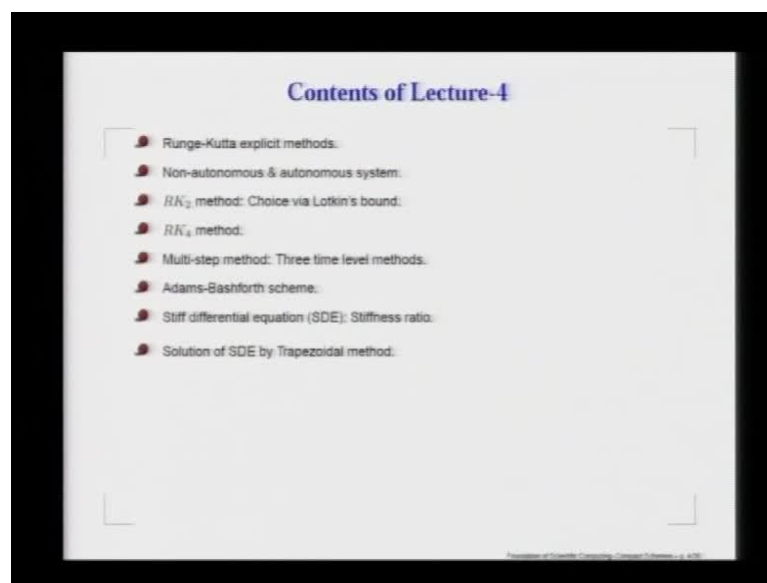
**Foundation of Scientific Computing**  
**Prof. T. K. Sengupta**  
**Department of Aerospace Engineering**  
**Indian Institute of Technology, Kanpur**

**Module No. # 01**

**Lecture No. # 04**

In today's fourth lecture, we are going to talk about only the explicit Runge-Kutta method because that has been found to be one of the most accurate methods.

(Refer Slide Time: 00:16)



As we go along in this course, we will find out why it is preferred, and we will also talk about another classification of differential equations - time dependent differential equations.

This is related to the non-autonomous versus autonomous system, and when we express it as an evolution equation, if the right hand side does not explicitly involve time, we have what is called as the non-autonomous systems. And in this course, we will be mostly focusing our attention to non-autonomous systems because many of the applications will see that time does not appear explicitly. And coming back to various Runge-Kutta methods that we talk about, we begin by two stage Runge-Kutta method which we have shown here as a RK 2 method. This will be followed by RK 3 and RK 4 methods. However, we will be not discussing in greater detail about RK 3 method

because that will be given to you as an assignment that you will be doing as a first assignment for this course.

Having exhausted this discussion on Euler and Runge-Kutta method, we introduce the multi-step method by talking about three time level methods. And in this, we will be talking about Adams-Bashforth scheme and also the leapfrog schemes. I will find out the pluses and minuses, and why we do not want to use it. Finally, talking about this ordinary differential equation solution methods, we need to talk about what are called as stiff differential equations.

If in the solution of this ordinary differential equation, multiple length or time scales are involved, then what we encounter is a specific difficulty which is characterized by this term called stiff differential equation characterized by the parameter stiffness ratio. And then, we talk about the classical methods of solution of stiff differential equation by trapezoidal method. This will be the final topic of today's talk.

(Refer Slide Time: 02:52)

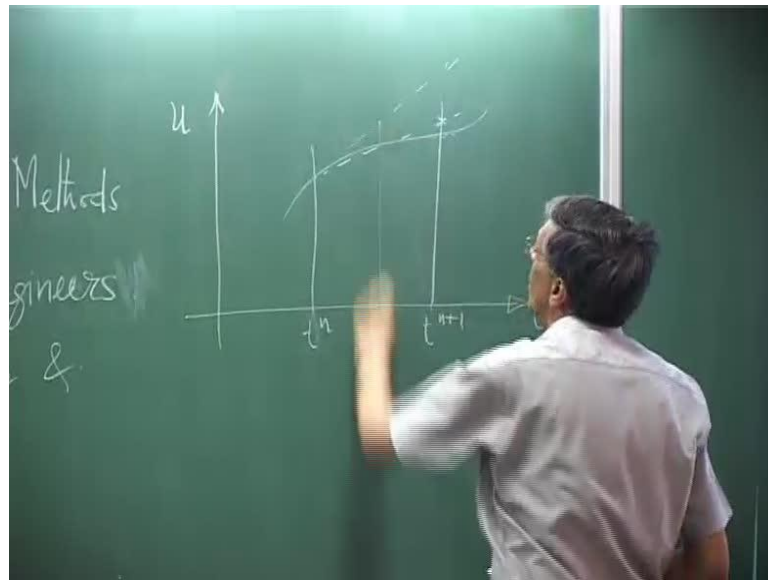
### Runge-Kutta Methods

- In Singlestep two-stage methods we note that for advancing the solution from  $t^n$  to  $t^{n+1}$ , we had to evaluate  $K_1$  and  $K_2$ .
- Runge-Kutta methods are also classified as singlestep multistage methods as the slope for integration is calculated as weighted average of slopes evaluated at multiple intermediate stages between  $t^n$  and  $t^{n+1}$ .

Here is the problem. We noted that you can adopt single step method. From the Physics point of view, that you do not get into numerical or spurious mode, but you could have multi-stage, and what is that, and the first milestone in that road would be a two stage method.

We have talked about some of those; we have talked about Euler-Cauchy method. We have seen that if you are matching the solution from  $t_n$  to  $t_{n+1}$ , you have to evaluate two additional quantities which we called as  $K_1$  and  $K_2$ . So, in that same class, Runge-Kutta method is also formed. They are single stage, multi-stage methods.

(Refer Slide Time: 03:45)



We noted that, if we are trying to solve an equation of this kind and we are going from one time step to the next one, then the Euler method is required that you take a slope here and you extrapolate, and that is where you get, and that could amount to some error, or you could draw this tangent here, which is the parallel to this, and you could get this; that was your backward Euler method; that is what we talked.

So, here, in the multi-stage method, you see that you generate all your information from either of these terminal's points; instead, you also look for some intermediate point from where you keep on getting information and use them, and improve the accuracy of the method.

(Refer Slide Time: 02:52) So, that is the whole idea that you do get information from multiple stages in between these two time levels  $t_n$  and  $t_{n+1}$ . And eventually, you get the solution as a weighted average of all these information that you have culled from these terminal points plus those intermediate stages.

(Refer Slide Time: 05:14)

### Runge-Kutta Methods (cont.)

- We restrict our attention to explicit methods, for which the slopes are evaluated at  $m$  such stages given by,

$$K_1 = hf(t^n, u^n) \quad (45a)$$

$$K_2 = hf(t^n + c_2h, u^n + a_{21}K_1) \quad (45b)$$

$$K_3 = hf(t^n + c_3h, u^n + a_{31}K_1 + a_{32}K_2) \quad (45c)$$

.

$$K_m = hf(t^n + c_mh, u^n + a_{m1}K_1 + a_{m2}K_2 + \dots + a_{mm-1}K_{m-1}) \quad (45m-1)$$

- Each  $K_i$ 's are obtained from the cumulative predecessors, a distinct feature of explicit methods.

- The Runge-Kutta method is then defined by

$$u^{n+1} = u^n + W_1K_1 + W_2K_2 + \dots + W_mK_m \quad (46)$$

Foundations of Scientific Computing-Module 2--p. 3452

So, that gives you a normal leverage and freedom to choose methods from; for example, if you restrict your attention to just simply explicit methods, then you would keep on calculating let us say slopes at  $m$  such points; so,  $m$  could be at your disposal; you decide upon how many stages you would like to. So, we are suggesting that, we choose  $m$  such methods, and we have chosen these stages - intermediate stages in such a way that each  $K$  is could be obtained from the information you have collected up to that point.

So, for example, we have information at  $t_n$ ; so, we could immediately calculate  $K_1$ . Having obtained  $K_1$ , what we could do is we could obtain the solution at a next stage which is in between  $t_n$  and  $t_{n+1}$  so that this  $c_2$  is less than 1. So, you have some intermediate point which is  $c_2h$  from  $t_n$ , and then, there, we also calculate the dependent variable as some kind of a weighting of what we have just now calculated in the previous level  $K_1$ ; that means, in evaluating  $K_2$ , you need  $K_1$ . In evaluating  $K_3$ , you would require  $K_1$  and  $K_2$ , and so on, so forth. So, this is essentially a recipe for explicit methods.

So, that is what, we said that, we will be keeping our attention focused to explicit methods. We will obtain the slopes at  $m$  such stages. Each  $K_i$ 's will be obtained from the cumulative predecessors. That is a distinct feature of all explicit methods and eventually you get the final accepted solution as the previous time step solution plus this weighting factor times this  $K_1$ .

So, these  $K_i$ 's are nothing but something like your  $\Delta u$ . **right** So, they are the  $\Delta u$ 's and this  $W_1$   $W_2$  and  $W_m$  - they are the weights that you associate with the method.

So, essentially, now if you look back, you have to decide upon what are these coefficients -  $c_2, c_3$ , all the way up to  $c_m$ , and here you have  $a_{21}, a_{31}$ , and  $a_{32}$ , and so on so forth, and finally, these weights are also our unknowns.

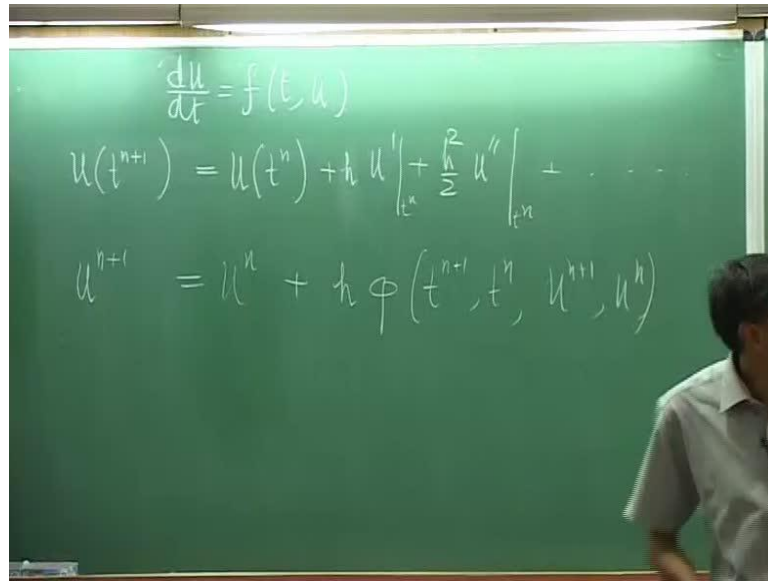
(Refer Slide Time: 07:50)

### Runge-Kutta Methods (cont.)

- The parameters of the Runge-Kutta methods are  $c_j, a_{ij}$  and  $W_j$ 's.
- These parameters are obtained by matching the corresponding expansion of (46) with the Taylor series method.
- The terms in Taylor series those having been matched, defines the order of the method.

So, these are basically the parameters of the Runge-Kutta methods and they have to be obtained. One of the easiest ways is to work out the Taylor series expansion of the function  $u_{n+1}$  in terms of  $u_n$ , and the number of terms that you match from your theoretical Taylor series expansion.

(Refer Slide Time: 08:16)


$$\frac{du}{dt} = f(t, u)$$
$$u(t^{n+1}) = u(t^n) + h \left. u' \right|_{t^n} + \frac{h^2}{2} \left. u'' \right|_{t^n} + \dots$$
$$u^{n+1} = u^n + h \phi(t^{n+1}, t^n, u^{n+1}, u^n)$$

So, what we mean by theoretical Taylor series expansion is, we are writing  $u$  of  $t_n + 1$  – this we are going to write  $u$  of  $t_n$ . Since your governing is, we are looking at  $d$ ; so, if this is what (Refer Slide Time: 08:42) we adopt our governing equation as, then we can keep on adding whatever information that we have;  $\Delta t$  which we will be calling as  $h$ ; So,  $h$  times of  $u$  prime and  $h$  squared  $u$  double prime, and these are all evaluated at  $t_n$ , so on and so forth.

So, this is your analytical expression. Whereas, the numerical solutions we have denoted so far with the simple superscript like this, and we have said that our method would be something like  $h \phi$  of  $t_n + 1, t_n, u_n + 1$ , and of course it will also be a probably the function  $h$  also.

So, what you essentially do is - you obtain two Taylor series - one from the analytical expression; one from the numerical method. You keep matching them, and the number of terms that you match from either side would define your order of the method. So, if I go up to say  $h$  squared, that is a second order method; if I decide to go up to  $h$  cube, that **that** will be our third order method, and so on and so forth. So, this is the essential idea in a nutshell.

(Refer Slide Time: 10:09)

### Explicit Second Order Runge-Kutta Method

- This method is evaluated as (with  $m = 2$ ),  
$$K_1 = hf(t^n, u^n) \quad (47a)$$
$$K_2 = hf(t^n + c_2h, u^n + a_{21}K_1) \quad (47b)$$
- The Second order Runge-Kutta method is then defined by,  
$$u^{n+1} = u^n + W_1K_1 + W_2K_2 \quad (48)$$
- The parameters of the Runge-Kutta methods are  $c_j$ ,  $a_{ij}$  and  $W_j$ 's.
- These parameters are obtained by matching the corresponding expansion of (46) with the Taylor series method.
- The number of terms in Taylor series those having been matched defines the order of the method.

Foundation of Scientific Computing: Module 2 - p. 38/39

So, you can now just work out. One such simplest possible method would be an explicit second order Runge-Kutta method. So, here we are of course fixing  $m$  equal to 2, and we will first evaluate  $K_1$ ; that would be the  $\Delta u$  at the starting  $h$ ;  $h$  times  $f$  evaluated at  $t_n$ ,  $u_n$ , and then, next we will be adding to this path, the independent variable incremented by  $c_2 h$ , and the corresponding dependent variable incremented by  $a_{21} K_1$ , and so that the accepted solution at the  $n$  plus 1th of level would be the predecessor plus the weighting of these two increments  $K_1$  and  $K_2$ .

So, very easily, you can see that 1,2,3,4 - we have four unknowns. So, for this second order method, we will have to work out some ways and means of evaluating these four unknowns.

(Refer Slide Time: 11:18)

## Explicit Second Order Runge-Kutta Method

- Numerically obtained  $u^{n+1}$  should match  $u(t^{n+1})$  as closely as possible. The latter can be expanded by Taylor series as,

$$\begin{aligned} u(t^{n+1}) &= u(t^n) + h u'(t^n) + \frac{h^2}{2!} u''(t^n) + \frac{h^3}{3!} u'''(t^n) + \dots \\ &= u(t^n) + h f(t^n, u(t^n)) + \frac{h^2}{2} [(f_t + f f_u)]_{t^n} + \frac{h^3}{3!} \\ &\quad [f_{tt} + 2f f_{tu} + f^2 f_{uu} + f_u(f_t + f f_u)]_{t^n} + \dots \quad (49) \end{aligned}$$

- We can obtain  $u^{n+1}$  using (47)-(48), by expanding  $K_2/h$  as,

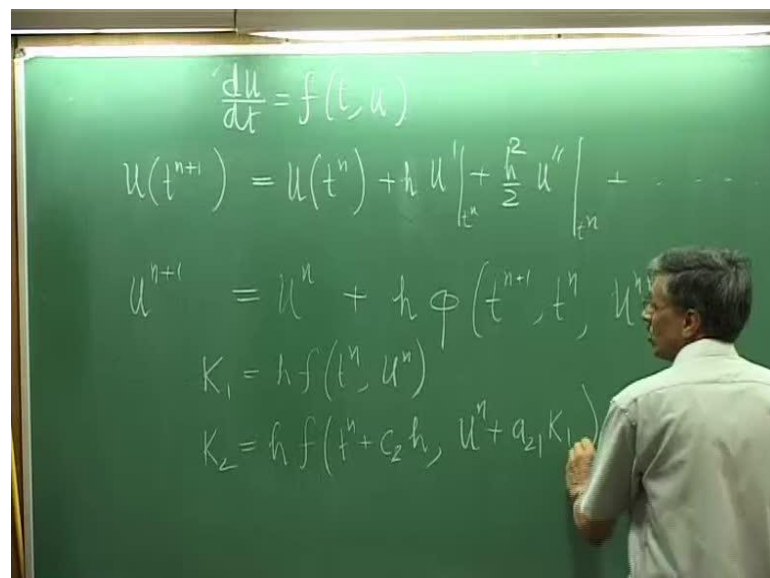
$$\begin{aligned} \frac{K_2}{h} &= f + h c_2 f_t + a_{21} K_1 f_u + \frac{(h c_2)^2}{2!} f_{tt} + 2 h c_2 a_{21} K_1 f_{tu} \\ &\quad + \frac{(a_{21} K_1)^2}{2!} f_{uu} + \frac{(h c_2)^3}{3!} f_{ttt} + \frac{(h c_2)^2}{2} a_{21} K_1 f_{ttu} + \frac{h c_2}{2} (a_{21} K_1)^2 f_{tuu} \\ &\quad + \frac{(a_{21} K_1)^3}{3!} f_{uuu} + \dots \end{aligned}$$

Since  $K_1 = h f$ , above can be simplified further to

Foundations of Scientific Computing-Module 2--p. 38/52

So, what we do is, as I told you, we will write down the theoretical estimate in terms of the Taylor series, and as you can see,  $u'$  as given here is  $f$ ; so, that is your  $h$  times  $f$   $u$  double prime. Now, you have to realize that  $f$  is a function of  $u$  and  $t$ . So, if you take  $u$  double prime, that will give you the partial of  $f$  with respect to  $t$  plus  $f$  times  $f_u$ , and the following term - the third order term, would have many more such combinations  $f_{tt}$  plus  $2 f f_{tu}$  plus  $f^2 f_{uu}$  plus this and so on, so forth.

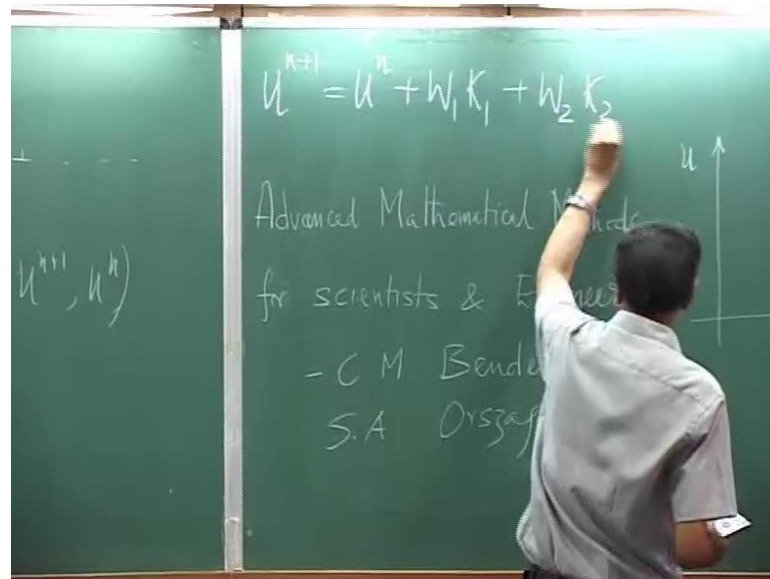
(Refer Slide Time: 12:17)





So, we could keep on writing it all down and the way we have written down  $K_1$ ,  $K_1$  was simply  $h$  of  $f$  evaluated at the previous  $h$ , and  $K_2$  we said would be  $h$  of  $f$  evaluated at  $t_n$  plus  $c_2 h$  and  $u_n$  plus  $a_{21} K_1$ . So, that is the expression for that. So, if I expand this  $K_2$  by  $h$ , that is basically a Taylor series expansion of this about  $t_n$  and  $u_n$ , and that gets you there.

(Refer Slide Time: 13:08)



So, basically, you have this  $K_2$  by  $h$ , and then finally, what you need to do is, you would be writing this  $u_n$  plus  $W_1 K_1$  plus  $W_2 K_2$ .

(Refer Slide Time: 13:25)

### Second Order Runge-Kutta Method (cont.)

$$\begin{aligned} \frac{K_2}{h} = & f + hc_2 f_t + ha_{21} f f_u + \frac{(hc_2)^2}{2!} f_{tt} + 2h^2 c_2 a_{21} f f_{tu} \\ & + \frac{(a_{21} h f)^2}{2!} f_{uu} + \frac{(hc_2)^3}{3!} f_{ttt} + \frac{(hc_2)^2}{2} a_{21} h f f_{ttu} + \frac{hc_2}{2} (a_{21} h f)^2 f_{tuu} \\ & + \frac{(a_{21} h f)^3}{3!} f_{uuu} + \dots \end{aligned}$$

- Plugging this and  $K_1$  in (48) one obtains,

$$u^{n+1} = u^n + W_1 h f + W_2 h \left( f + hc_2 f_t + ha_{21} f f_u + \frac{h^2}{2} f_{tt} + h^2 c_2 a_{21} f f_{tu} + \frac{h^2}{2} a_{21}^2 f^2 f_{uu} + \dots \right) \quad (50)$$

- Equating (49) and (50) up to  $O(h^2)$  terms one gets,

$$O(h) : \quad W_1 + W_2 = 1 \quad (51a)$$

$$O(h^2) : \quad W_2 c_2 f_t + W_2 a_{21} f f_u = \frac{1}{2} [f_t + f f_u] \quad (51b)$$

You have this expression for  $K_1$  and  $K_2$ , and then, you keep using those information in terms of  $u$  and  $f$ , and this is what you get;  $u_{n+1}$  should be  $u_n$  plus  $W_1$  into  $h$  plus  $W_2$  into  $h$ , and these are those expressions that we have written down as Taylor series. Expansion of that, expanded about at  $t_n$  would give you this whole set of terms. So, basically, talking about a second order method, all you would be interested in writing down the coefficients on either side of order  $h$  and order  $h$  squared; order  $h$  you can very clearly see, here, the coefficient **is** would be, analytical expression would be 1 itself. So, that is that;  $h$  itself and coefficient is 1; whereas here you are seeing that it should be  $W_1$  plus  $W_2$  (Refer Slide Time: 14:25). So, that is your order  $h$  condition.

Similarly, order  $h$  square would give you, from here you can see this  $W_2$  into  $h^2$  plus  $f''$  into  $h^3$  over  $2$ , and that is multiplied by  $f'$ , and if you look at the corresponding analytical expression, this is what we have. So, this is your  $u''$ . So, this is basically  $u''$  here; it is nothing but this quantity - half times  $f''$  plus  $f'f'$  (Refer Slide Time: 14:57)

So, what you notice is that  $u''$  is a direct equation that you can work with. What about this? This equation somehow depends on your problem because the problem defines your function  $f$ . So, what happens is - for a general methodology, you should not be talking about any arbitrary function, but if you do, then you note that these coefficients of  $f''$  and the coefficient of  $f'f'$  must also independently match because that would be the most general way of doing it, and if you do, that would actually yield two conditions.

So, basically you have one condition from order  $h$  and two conditions from order  $h$  square; that gives you three relations, but you have four unknowns. So, the easiest way is to solve it parametrically; you choose any one of them as a parameter; let that be  $c_2$ ; then, you can immediately see that, from these equations, you would get  $u''$  equal to  $c_2$ , and  $W_2$  you can work it out from here (Refer Slide Time: 16:20), and  $W_1$  is 1 minus  $W_2$ . So, that will give you these conditions.

So, basically, we have solved it in terms of the parameter  $c_2$ . We still have to decide upon how we go about fixing  $c_2$ , but I may draw your attention at what we had done. We had taken  $t_n$  and then we added a fraction of the time step to go to the intermediate stage.

Whenever you have explicit dependence on  $t$ , you call such systems as non-autonomous; they are called the non-autonomous system. In contrast, you can have autonomous system where the  $t$  does not appear explicitly. So, you would not have this part, if you do not have that part (Refer Slide Time: 17:05), that is what we call as autonomous system. So, that means, we are talking about something like  $c_2 = 0$ , but then, you may say that this looks difficult because  $c_2$  appears in the denominator; it is not that difficult; you can work from the first principle and develop that. We will be actually doing one such example of autonomous system and we will work out for a higher order system, and then you will know what I am talking about.

(Refer Slide Time: 17:39)

### Second Order Runge-Kutta Method (cont.)

- The parameters in (52) defines the simplest possible Runge-Kutta methods of order two given by,

$$u^{n+1} = u^n + (1 - \frac{1}{2c_2})K_1 + \frac{1}{2c_2}K_2 \quad (53)$$

where

$$K_1 = hf(t^n, u^n) \text{ and } K_2 = hf(t^n + c_2h, u^n + c_2K_1) \quad (54)$$

- The leading term omitted in RK2 method is obtained from (50) as the  $O(h^3)$  term given by,

$$\frac{W_2 c_2^2}{8}(f_{tt} + 2f f_{tu} + f^2 f_{uu}).$$

- Hence the local truncation error is given by,

$$\begin{aligned} T^{n+1} &= u(t^{n+1}) - u^{n+1} = \frac{h^3}{3!}[f_{tt} + 2f f_{tu} + f^2 f_{uu} + \\ &\quad f_u(f_t + f f_u)] - \frac{h^3 c_2}{4}(f_{tt} + 2f f_{tu} + f^2 f_{uu}) \\ &= h^3[(\frac{1}{6} - \frac{c_2}{4})(f_{tt} + 2f f_{tu} + f^2 f_{uu}) + \frac{1}{6}\{f_u(f_t + f f_u)\}] \quad (55) \end{aligned}$$

Foundation of Scientific Computing, Module 2, p. 40/62

So, if I now write down everything in terms of  $c_2$ , then  $u^{n+1}$  would be  $u^n$  plus these two weighting functions times the increment  $K_1$  and  $K_2$ . So, one way of fixing  $c_2$  would be to look at what we have omitted while working out this strategy. That would be the next term, that we have omitted, that would be order  $h^3$  term, which is nothing but this (Refer Slide Time: 18:19 to 18:40), and since  $W_2 c_2^2$  is  $h^2 c_2^2 h$ , we can club it in there and then we can see that it is the truncation error term is the leading term or the  $h^3$  term which will work out like this. So, we have the leading truncation error term written in terms of the right hand side of the equation that we are trying to solve plus our choice of  $c_2$ .

(Refer Slide Time: 19:13)

$u^{k+1} = u^k + w_1 k_1 + w_2 k_2$   
 Lofkin's bound:  
 $\left| \frac{\partial^{l+i} f}{\partial t^i \partial u^l} \right| < \frac{L^{l+i}}{M^{j-1}}$   
 $i, l = 0, 1, 2$   

|              |                            |
|--------------|----------------------------|
| $ f  < M$    | $ f_{tt}  < L^2 M$         |
| $ f_u  < L$  | $ f_{tu}  < L^2$           |
| $ f_t  < LM$ | $ f_{uu}  < \frac{L^2}{M}$ |

 $|T^{n+1}| \leq M L^2 h^3 \left\{ \frac{1}{3} \left( \frac{1}{6} - \frac{c_2}{4} \right) + \frac{1}{3} \right\}$

Now, there are various ways of looking at it - how to fix  $c_2$ ? They are something called Lofkin's bound which tells you that you could work out and estimate for the quantities that you see in your truncation error; that would be  $\frac{\partial^i f}{\partial t^i \partial u^j}$ . So, this mixed derivative is bounded by this Lofkin's prescription; that will be  $i$  plus  $j$  minus 1. So, your  $i$  and  $j$  would of course go from 0, 1, 2, etcetera. So, if you do that, you can immediately see what you get. Suppose, I choose  $i$  equal to  $j$  equal to 0, then that would give me an estimate for  $f$ . So,  $\text{mod } f$  would correspond to  $i$  equal to  $j$  equal to 0; if I put that, it would give me this bound, that  $\text{mod } f$  should be bounded by  $m$ .

Now, if I look at this partial derivative, so that would imply what?  $i$  equal to 0,  $j$  equal to 1; And that would be bounded by this quantity.  $L$  and  $M$  which are some finite bounds or finite numbers; so, that we do not have to know in general, but for given specific function, we can work out these quantities.

So,  $f$  of  $t$  would be similarly  $i$  equal to 1,  $j$  equal to 0, and that will give you this. And you could work out, similarly, the other relations:  $f$  of  $t$   $t$  which is  $i$  equal to 2 and  $j$  equal to 0; that would be bounded by  $L$  squared  $M$ , and  $f$   $t$   $u$  would correspond to  $i$  equal to 1,  $j$  equal to 1, and that would be bounded by  $L$  squared, and this will be bounded by  $L$  squared by  $M$ .

So, if I use this information from these bounds, then you could see that  $t^{n+1}$  could be bounded by  $M L$  squared. We are looking at the leading term, as that would be  $3 \frac{1}{6} - \frac{c_2}{4} + \frac{1}{3}$ . So, this is one way of looking at the problem.

So, what? We are saying that this leading truncation error term would be given like this. So, basically, in essence, we have substituted these bounds in this last equation 55; that yields this coefficient as 3 and this one gives you one third (Refer Slide Time: 22:42)

So, **basically**, **then** you can immediately see that this will be minimum, if I choose  $c_2$  in such a way that this quantity here is equal to 0. **right** So, that is **the** one way of doing it; however, we need not have gone through all this. If you are smart enough, you can immediately reason it out here, that for any arbitrary choice of  $f$ , you can see that; this, we do not have any control **right**. So, only thing that you can exercise control in reducing the truncation error would be to put this first set of term equal to 0, and that immediately, such that this coefficient equal to 0. So, without even bothering to work out all these bounds and etcetera, your common sense will immediately tell you that if you put this quantity in the first bracket itself equal to 0, then you have done. So, that is equivalently choosing  $c_2$  equal to two-thirds.

(Refer Slide Time: 23:37)

### Second Order Runge-Kutta Method (cont.)

- It is evident that  $c_2$ -dependent error terms will be minimum for any arbitrary  $f$ , if  $c_2 = 2/3$ .
- This is one possible choice of a RK2 method for which  
 $a_{21} = 2/3, W_1 = 1/4, W_2 = 3/4$  so that,  
 $K_1 = hf(t^n, u^n)$  (56a)  
 $K_2 = hf(t^n + 2h/3, u^n + 2K_1/3)$  (56b)  
and  $u^{n+1} = u^n + \frac{1}{4}(K_1 + 3K_2)$  (56c)
- Another simpler variation is possible with  $c_2 = 1/2$ , so that  
 $u^{n+1} = u^n + K_2$  (57)  
Where  $K_2 = hf(t^n + \frac{h}{2}, u^n + \frac{K_1}{2})$   
This is called the **modified Euler-Cauchy method**.

So, this is one way of developing a Runge-Kutta method. So, you choose  $c_2$  equal to two-thirds; that will immediately give you a  $c_2$  equal to two-thirds and you can work out  $W_1$  and  $W_2$  like this. So, this is essentially the step that you would be doing in writing a program that you could write  $K_1$  and  $K_2$  in terms of this, and eventually  $W_1$  and  $W_2$  will assist you in fixing the solution at the next step. So, that is one way which ensures reduction of error; so, that is the choice

(Refer Slide Time: 24:31)

## Second Order Runge-Kutta Method (cont.)

- If (51b) has to hold for any arbitrary r.h.s. of (23), then one must have the coefficients of  $f_t$  and  $f f_u$  to be same also. Thus,  

$$W_2 c_2 = 1/2 \quad (52a)$$

$$W_2 a_{21} = 1/2 \quad (52b)$$
- We have three relations for four unknowns  $(c_2, a_{21}, W_1, W_2)$ . Solve parametrically in terms of  $c_2$  as,  

$$a_{21} = c_2; \quad W_2 = 1/(2c_2) \quad W_1 = 1 - 1/(2c_2) \quad (52)$$
- We have assumed that  $c_2 \neq 0$  i.e. the r.h.s. of (23) is an explicit function of time. Such systems are called **non-autonomous**.
- What happens to **autonomous system**?

Foundations of Scientific Computing-Module 2 – p. 40/62

Well, you could also see that, here, if I would choose  $c_2$  equal to half, then  $W_1$  becomes equal to 0, **right** and then what I need to obtain is just simply  $W_2$ . And then, I can get this solution in terms of  $K_2$  alone because  $W_1$  equal to 0. So, this is what is called as modified Euler-Cauchy method. So, it is a multi-stage method, but again, what you would be doing is you would be evaluating  $K_1$ , but you do not need to store it; instead, you can directly obtain  $K_2$  and obtain the solution at the next step in terms of  $K_2$  alone. So, this is the way that these methods are worked out.

(Refer Slide Time: 25:25)

## Runge-Kutta Method for Autonomous System

- RK<sub>4</sub> method for the autonomous system:  $\frac{\partial u}{\partial t} = f(u)$   
 evaluates four slopes,  

$$K_1 = hf(u^n) \quad (58a)$$

$$K_2 = hf(u^n + a_{21}K_1) \quad (58b)$$

$$K_3 = hf(u^n + a_{31}K_1 + a_{32}K_2) \quad (58c)$$

$$K_4 = hf(u^n + a_{41}K_1 + a_{42}K_2 + a_{43}K_3) \quad (58d)$$

$$\text{and } u^{n+1} = u^n + W_1K_1 + W_2K_2 + W_3K_3 + W_4K_4 \quad (58e)$$
- These slopes are expanded next as,  

$$A = \frac{K_2}{h} = f(u^n) + a_{21}K_1f_u + \frac{(a_{21}K_1)^2}{2!}f_{uu} + \frac{(a_{21}K_1)^3}{3!}f_{uuu} \quad (59)$$

$$K_2 = hf + h^2a_{21}ff_u + \frac{h^3}{2!}a_{21}^2f^2f_{uu} + \frac{h^4}{3!}a_{21}^3f^3f_{uuu}$$

$$K_3 = hf + h^2(a_{31}+a_{32})ff_u + h^3[a_{21}a_{32}ff_u^2 + (a_{31}+a_{32})^2f^2f_{uu}/2]$$

$$h^4[\frac{a_{32}a_{21}^2}{2}f^2f_u f_{uu} + (a_{31}+a_{32})a_{32}a_{21}f^2f_u f_{uu} + \frac{(a_{31}+a_{32})^3}{6}f^3f_{uuu}]$$

Foundations of Scientific Computing-Module 2 – p. 43/62



So, if I want to go little higher, for example, we are looking at a fourth order method; so, that I will call as R K 4 method; Runge-Kutta four stage method, and let us make our job simpler by considering an autonomous system. Although it may appear to be sort of a big restriction, but most of the practical problems that you would encounter would be of this class. So, most of the problems that we come across in computing belongs to autonomous class, so that giving up the dependence on  $t$  is not a very serious one. So, the methods that we are going to talk about are going to be quite useful.

So, since we are talking about four stage method, we would be evaluating all these four increment functions:  $K_1$ ,  $K_2$ ,  $K_3$ , and  $K_4$ , and you can see the pattern that you do not have dependence on  $t$ , but you have dependence on  $u$  alone. So, the  $K_2$  will be incremented by a  $2 \times 1$  times,  $K_1$   $K_2$  will be incremented by a  $3 \times 1$   $K_1$  plus a  $3 \times 2$   $K_2$ , and  $K_4$  would be incremented by previous three increments  $K_1$ ,  $K_2$ ,  $K_3$  here itself, and final solution would be the weighting of all these increments.

So, all that needs to be done is essentially the same step that we have worked out just now. We will be working those expressions out for the Taylor series, if I look at  $K_2$ , so, define  $a$  as equal to  $K_2$  by  $h$ ; then, we can expand it, and this is how it is noticed that  $K_1$  itself is  $h$  times  $f$ . So, you can plug the main. And what you are going to see? That  $K_2$  will be  $h$  times  $f$  and this gives you a  $h$ , and there is another  $h$ . So, **it will** they have the second term  $h$  squared a  $2 \times 1$   $f f u$ .

The next term would come from here. That will be  $h$  cube by factorial 2 and if there is a  $2 \times 1$  square and we have  $f$  square coming from here and this  $f u u$  remains as it is and the next term would give you this. By now, you have realized that, having chosen the order of the method, you just need to keep terms up to that order only. So, that is what we are doing. We are just retaining terms of  $2 \times h^4$  because it is a four stage method. So, that is your expansion for  $K_2$ . Similarly,  $K_3$  would work out like this and you can notice that you have to use all this information on  $K_1$  and  $K_2$  in  $K_3$ . It is a little involved and you get this whole set of terms when you retain terms up to  $h^4$  alone.

You realize that, if I were to be doing exercise for a non-autonomous system, it would be lot more tougher because there would be  $f t$  and  $f$ ; mixed derivatives would come; that is why I try to keep it controlled. To look at the way these methods are developed for this autonomous system, non autonomous system is little more involved.

(Refer Slide Time: 29:14)

### RK Method for Autonomous System (cont.)

$$\frac{K_4}{h} = f + (a_{41}K_1 + a_{42}K_2 + a_{43}K_3)f_u + \frac{h^2}{2}(a_{41}f + a_{42}A + a_{43}B)^2 f_{uu} + \frac{h^3}{3!}(a_{41}f + a_{42}A + a_{43}B)^3 f_{uuu}$$

• Where,

$$a_{41}K_1 + a_{42}K_2 + a_{43}K_3 = (a_{41} + a_{42} + a_{43})fh + h^2[a_{42}a_{21}ff_u + a_{43}(a_{31} + a_{32})ff_u] + h^3[\frac{1}{2}a_{42}a_{21}^2f^2f_{uu} + a_{43}a_{32}a_{21}ff_u^2 + \frac{1}{2}a_{43}(a_{31} + a_{32})^2f^2f_{uu}]$$

Foundations of Scientific Computing-Module 2 – p. 44/52

So, the last thing that you need to do is find out this  $K_4$  quantity;  $K_4$  by  $h$ ; that would give you this and you can notice that what you have here is the  $a_{41}K_1$ ,  $a_{42}K_2$ , and  $a_{43}K_3$ , which itself was a kind of a series of terms. And, if I keep in that series of terms only have to  $h$  cube because there is a  $h$  downstairs here. That would make this overall series on the right hand side as  $h^4$ , and that is what we need to do.

(Refer Slide Time: 29:53)

### RK Method for Autonomous System (cont.)

• After rearrangement one obtains,

$$K_4 = fh + h^2ff_u(a_{41} + a_{42} + a_{43}) + h^3\{ff_u^2[a_{42}a_{21} + a_{43}(a_{31} + a_{32})] + (a_{41} + a_{42} + a_{43})^2\frac{f^2f_{uu}}{2}\} + h^4[f^2f_{uu}\frac{a_{42}a_{21}^2}{2} + a_{43}a_{32}a_{21}ff_u^3 + \frac{a_{43}(a_{31} + a_{32})^2}{2}f^2f_{uu} + (a_{41} + a_{42} + a_{43})\{a_{42}a_{21} + a_{43}(a_{31} + a_{32})\}f^2f_{uu} + (a_{41} + a_{42} + a_{43})^3\frac{f^3f_{uuu}}{6}]$$

• These expressions for  $K_1$ ,  $K_2$ ,  $K_3$  and  $K_4$  are plugged in (58e) to obtain  $u^{n+1}$  involving unknown constants  $W_j$ 's and  $a'_{ij}$ 's, that can be equated with the Taylor series expansion of  $u(t^{n+1})$  in terms of  $t^n$  as given below:

$$u(t^{n+1}) = u(t^n) + hu'(t^n) + \frac{h^2}{2!}u''(t^n) + \frac{h^3}{3!}u'''(t^n) + \dots \\ = u(t^n) + hf + \frac{h^2}{2}ff_u + \frac{h^3}{3!}(f^2f_{uu} + ff_u^2) + \frac{h^4}{4!}(f^3f_{uuu} + 4f^2f_{uu}f_u + ff_u^3) + \dots$$

Foundations of Scientific Computing-Module 2 – p. 44/52

So, that is what we do. We do a little bit of algebra and this is what appears as the intermediate solution for this increment  $K_4$ . Having obtained these expressions for  $K_1$ ,



K 2, K 3, and K 4, you go back and plug it in that expression where we have written down  $u_{n+1}$  in terms of  $u_n$ , and this increments K 1 to K 4.

(Refer Slide Time: 30:58)

### RK Method for Autonomous System (cont.)

- Equating the coefficients of  $O(h)$  terms, one obtains:  

$$W_1 + W_2 + W_3 + W_4 = 1 \quad (60)$$
- Similarly, equating the coefficients of  $O(h^2)$  terms one gets,  

$$W_2 a_{21} + W_3(a_{31} + a_{32}) + W_4(a_{41} + a_{42} + a_{43}) = 1/2 \quad (61)$$
- The  $O(h^3)$  terms are with  $f f_u^2$  and  $f^2 f_{uu}$  as coefficients and they are treated as discussed before,  

$$f f_u^2 : W_3 a_{32} a_{21} + W_4[a_{42} a_{21} + a_{43}(a_{31} + a_{32})] = 1/6 \quad (62)$$

$$f^2 f_{uu} : W_2 a_{21}^2 + W_3(a_{31} + a_{32})^2 + W_4(a_{41} + a_{42} + a_{43})^2 = 1/6 \quad (63)$$
- The  $O(h^4)$  terms are with  $f^3 f_{uuu}$ ,  $f^2 f_u f_{uu}$  and  $f f_u^3$  as coefficients. Equating these will yield three more equations.

Foundations of Scientific Computing-Module 2 - p. 46/52

Then, you would have obtained a series where we have retained terms up to  $h^4$  and then what you do is you compare that series with the Taylor series expansion of  $u(t)$  at  $n+1$  and go on till  $h$  to the power 4; that is, your last line of that slide tells you, and equate term by term once you start doing that. Since you have access to this material, I am just not bothering you with all this derivation on the board. You can convince yourself; they are the ones that will appear. So, the lowest order term will tell you that this  $W_1$  to  $W_4$  adds up to 1, and that is what you would expect because we are talking about weightages. So, all the weights should add up to 1 and that is what you are seeing here.

Look at order  $h^2$  term; that will give you this. There is no problem there; however, if you look at order  $h^3$  term, you would once again see, there would be terms which are coefficients with  $f f_u^2$ , and there would be terms which are  $f^2 f_{uu}$  terms multiplied to that. So, if we equate the Taylor series with this algorithm, and then equate the coefficient of these two sets of terms as we did before, then order  $h^3$  will generate two equations, and that is what we have written down. 62 and 63 are those terms corresponding to  $f f_u^2$  and  $f^2 f_{uu}$ . You look at order  $h^4$  and there you would have terms which are multipliers of  $f^3 f_{uuu}$ ,  $f^2 f_u f_{uu}$ , and  $f f_u^3$ , and

f f u cube. So, if we match these three sets of terms, that will give us three more equations.

(Refer Slide Time: 32:56)

### RK Method for Autonomous System (cont.)

- These relations are:

$$f^3 f_{uuu} : W_2 a_{21}^3 + W_3 (a_{31} + a_{32})^3 + W_4 (a_{41} + a_{42} + a_{43})^3 = 1/4 \quad (64)$$

$$f f_u^3 : W_4 a_{43} a_{32} a_{21} = 1/24 \quad (65)$$

$$f^2 f_u f_{uu} : W_3 a_{32} a_{21}^2 + 2W_3 (a_{31} + a_{32}) a_{32} a_{21} + W_4 a_{42} a_{21}^2 + W_4 a_{43} (a_{31} + a_{32})^2 + W_4 (a_{41} + a_{42} + a_{43}) (a_{42} a_{21} + a_{43} (a_{31} + a_{32})) = 1/3 \quad (66)$$

- These are seven equations for ten unknowns.
- We can fix three of the unknowns & (65) tells that  $a_{43}$ ,  $a_{32}$  and  $a_{21}$  should not be arbitrary.
- Thus, we choose  $a_{31}$ ,  $a_{41}$  and  $a_{42}$  as zero.
- For non-autonomous system, classical RK4 method uses the following weights:  $W_1 = 1/6$ ,  $W_2 = 1/3$ ,  $W_3 = 1/3$ ,  $W_4 = 1/6$

So, what do we have now? We have from order h 1 equation, from order h square 1 equation; order h cube, we have two equations; order h 4 - three equations. So, all together, we have seven equations. How many unknowns do we have? We have those four weightages  $W_1$  to  $W_4$ , and then we have  $a_{21}$ ,  $a_{31}$ ,  $a_{32}$ , and  $a_{41}$ ,  $a_{42}$ ,  $a_{43}$ . So, there are ten unknowns now. So, that gives you some bit of room to maneuver. What you could do is, you could fix any three of them according to your convenience, and convenience means you would like to put some of them equal to 0; that would mean lesser number of computation. **right** If I set some coefficients equal to 0, that gives me the liberty of not evaluating those sets of terms.

So, we look for such opportunities, but if you look at this equation, that should send you a warning bell by saying – look, be careful; do not try to put  $a_{43}$ ,  $a_{32}$ ,  $a_{21}$  equal to 0; then you would be violating the condition that would not be; that is why, I have written it down that you should really not choose them as equal to 0. What we could instead do? We could pick up the other three:  $a_{31}$ ,  $a_{41}$ ,  $a_{42}$  equal to 0 because that is what is involved in those increment functions. So, if I trace these three to 0, then we can decide to have some kind of advantage of computing less. And I wanted to bring one fact r to your attention that, I have looked at some textbooks, very old ones and good ones, Ralston, and there

are some books by Indian authors too. Unfortunately, I could not reconcile to the fact that they talk about eight equations and what we are seeing here that we get seven equations. So, what happens is - working with 8 equations for 10 unknowns; they give you some kind of solutions.

(Refer Slide Time: 35:44)

### RK Method for Autonomous System (cont.)

- From (61) and (64), we obtain:  $a_{32} = 1 \text{ or } 1/2$
- For  $a_{32} = 1/2$ , from (61) and (65), we get:  $a_{43} = 1$
- For  $a_{32} = 1/2$ , using (63) and (65):  $a_{43} = 1 \text{ or } 1/\sqrt{2}$
- For  $a_{32} = 1/2$ , using (62) and (65):  $a_{43} = 1$
- Using  $a_{32} = 1/2$  in (63) and (65), we get, either:  $a_{43}^3 = 1$   
or:  $a_{43}^3 = 1/4$
- Most of the combinations do not satisfy (66), except the values obtained as  $a_{21} = 1/2$ ,  $a_{32} = 1/2$  and  $a_{43} = 1$ .
- Better approach in choosing the coefficients would be to minimize the next higher order truncation error terms.

Foundation of Scientific Computing, Module 2 - p. 47/52

So, we get these weightages as one sixth, one third; one third and one sixth, and accordingly some set of conditions are given. So, I asked one of my PG student from Maths to really see what is going on here, if we are making some mistake; the good news is, we are not making some mistake; apparently, those books are not reporting something consistent because as we can see, we cannot have more than seven equations, if I am going up to fourth order.

So, we decided to check out their solutions. So, we will adopt all those weights that they have chosen, and then work out for the rest. This is what we get - that two of the equations gives a  $3 \times 2$  either as 1 or one half; so, we decided to take one half because you do not want to go in an intermediate step, all the way to the terminal point so that 1 would be a non-admissible solution. **right** So, that is what. We decided that we should take a  $3 \times 2$  as half; if we do that, then we will get a  $4 \times 3$  as 1, and from two sets of equations that we will get a  $4 \times 3$  equal to 1, and if we take another sets, then we get two possibilities - 1 or 1 by root 2, and if I take another two equations, then I get a  $4 \times 3$  equal to 1.

So, all these tell you, somehow, that this  $1/\sqrt{2}$  is odd man here. So, that should not be a valid solution; instead, if we choose  $4/3$  equal to 1, they seem to satisfy 61, 62, 63 and 65; 64 is already satisfied. So, all you need to see if that equation 66 is satisfied most of the combinations that we get do not satisfy the last equation except this condition  $2/1$  is half,  $3/2$  is half and  $4/3$  is one. This is a big relief because what happened is - this is exactly the solution reported in those books; I do not know; it is a bit of mystery at this point that they have done something inconsistent; they come to some such solutions and that solution seems to be quite ok. And even if you correct those mistakes, you still get a set of solutions which are the same.

Well. This looks somewhat very inelegant. We have been looking at some kind of a solution which was obtained by someone. We are just checking this. The best thing would be to choose a method where we could minimize the next higher order truncation error term like what we talked about in RK 2 method. Just now, we looked at the truncation error term, order  $h^3$  term, and that we minimized, and that is how we fixed  $c_2$ .

So, we should be able to do similar such exercise; looks like a bit of a work, and that kind of a possibility always tell you, with all apologies to PG students, that PG students have bit of a time to work those details out. So, I have given it to my young friend there to work it out. So, that is the way.

So, you basically look at order  $h^5$  term and minimize that truncation error term. So, what we are talking about? We are talking about fixing three quantities in that; exercise is arbitrary; then you have 7 equations. And try to look at the next higher order term in that three dimensional space and find out where the minimum is, and that is how you do it. So, this is all about Runge-Kutta methods. Let me assure you that Runge-Kutta methods are one of the most elegant, physically correct, efficient methods. So, that is why I would not ask you to look for anything beyond Runge-Kutta methods, for all your requirements of time advancement for sure, and we will come back to that topic again when we come to discussing PDEs.

(Refer Slide Time: 40:05)

## Multistep Methods

- Here, we introduce only three time-level methods. For Eqn. (23), this is given by
$$u^{n+1} = \alpha_1 u^n + \alpha_2 u^{n-1} + \beta_1 h f(u^n) + \beta_2 h f(u^{n-1}) \quad (67)$$
- If we demand second order accuracy, then we must have
$$\alpha_1 = 1 - \alpha_2; \beta_1 = \frac{1}{2}(\alpha_2 + 3) \text{ and } \beta_2 = \frac{1}{2}(\alpha_2 - 1) \quad (68)$$
with  $\alpha_2$  as the free parameter.
- If we set  $\alpha_2 = 1$  then we obtain the mid-point leap frog method.
$$u^{n+1} = u^{n-1} + 2h f(u^n) \quad (69a)$$
- If  $\alpha_2 = 0$  then we obtain the second order Adams-Bashforth integration scheme.
$$u^{n+1} = u^n + \frac{3}{2}h f(u^n) - \frac{1}{2}h f(u^{n-1}) \quad (69b)$$

Foundations of Scientific Computing-Module 2 - p. 46/52

Now, just for the sake of completeness, I will briefly touch upon multi-step methods. You ought to know because lots of work gets done using multi-step method; say, for example, a generic equation that derives a three time level method is written here in this equation 67. So, you have the solution advance time level in terms of what you have in the current time level plus what you have stored in the previous time level.

So, that is how the three time level gets into the picture, and of course, in solving that equation  $du/dt$  equal to  $f$ . You need to evaluate those, and once again, we are talking here about autonomous system because that is what we get. Now, what you can do is you have to fix this coefficients  $\alpha_1$   $\alpha_2$   $\beta_1$   $\beta_2$ . So, what you do is, you write down the Taylor series expansion.

So, you write  $u^{n+1}$  in terms of  $u^n$ , and this of course, remains as it is. This, you also write in terms of  $u^n$ . So, what you do is you write down the Taylor series of this; also, that we have seen, and equate term by term and you get this equation;  $\alpha_1$  becomes  $1 - \alpha_2$   $\beta_1$  works out like this and  $\beta_2$  works out like this. So, you can once again treat  $\alpha_2$  as a free parameter. So, if you choose  $\alpha_2$  equal to 1, you immediately see that  $\alpha_1$  becomes 0, and  $\beta_2$  becomes 0, and the method that you get is called a Leapfrog Method, and this is what all the weather prediction people have been doing ever since they started their business. They always use this Leapfrog Method.

I will come to discussing about this particular method that after having said that multi-step methods have to be observed, but still it so happens that if you are solving in viscous equation, this method happens to be one of the most accurate methods, and that is for some good reason, atmospheric science people have been using for decades. Now, we will talk about it.

So, this is one of the methods you see. The three level method works out fine because what happens here, you get two modes and both the modes becomes coincident, and that happens to be the physical mode. So, that is the beauty of this method. Unfortunately, though this works only if you are looking at in viscous equation, the moment you add diffusion terms, you would notice that this method will blob on your face.

So, that is a major issue that you would not like to use a priori without knowing pros and cons of the method. It is a warning to you that appropriate analysis must be performed before adopting a method.

This last one, I have purposely added. This is advanced by Adams and Bashforth; these were two, I suppose, scientists in Cambridge in 19th century and they were solving some problem of surface tension, and they worked it out that they could do it by hand and they did come out with this method. What it does is, it advances the solution in terms of the current solution and this right hand side here is evaluated at the current time level, and the information obtained at the previous time level. The bad news about this method is, it is **the** overly glorified; lots and lots of people use it; in fact, you go and use any search engine, you will find that, lots and lots of people are claiming that they are doing most accurate calculation using this Adams-Bashforth method, which we will see that it is not really a good thing to adopt.

I purposely mention that Adams and Bashforth, when they developed, they were using their brain power to work it out by hand. So, they did not do many many time steps and that is why they did not probably realize the pitfall of the method which we did about few years ago, and we did report it in couple of papers and warn people, but looks like the warning has not registered. People still continue to use; so, this is something.

So, having come to a bit of a discussion about ODEs and their solution method, I try to bring in a little bit of advanced topic and that is what is called as a stiff differential equation. However, that may not appear to be such an oddity for people coming from

chemistry and chemical engineering. Stiff differential equations are those which are, of course, very stiff resistant. What happens is, if you try to integrate, then you would see that there are components of solutions which vary completely, differently.

(Refer Slide Time: 46:02)

## Stiff Differential Equations

- A general system of ODEs of the form  $u' = f(t, u_1, u_2, \dots, u_m)$  with  $u(t_0) = u_0$  is said to be stiff if the eigenvalues of the jacobian matrix  $[\partial f / \partial u]$  differ greatly in magnitude.
- The stiffness is quantified by the **Stiffness Ratio (SR)**  
as:  $SR = \frac{\max_i |\text{real part of } \lambda_i|}{\min_i |\text{real part of } \lambda_i|}$
- An example of such a system is the **Stuart-Landau-Eckhaus equation** given by:  

$$\frac{dA_1}{dt} = \alpha_1 A_1 + \beta_{11} A_1 |A_1|^2 + \beta_{12} A_1 |A_2|^2 \quad (70a)$$

$$\frac{dA_2}{dt} = \alpha_2 A_2 + \beta_{21} A_2 |A_1|^2 + \beta_{22} A_2 |A_2|^2 \quad (70b)$$
- This always occurs for **multi-modal systems or higher order differential equations**.

Foundations of Scientific Computing-Module 2 – p. 49/52

So, if I talk about a set of equations, so what we have written here is a sort of a vector equation.  $u$  prime means  $u_1$  prime  $u_2$  prime; I have written it in a columnar fashion and the right hand sides are functions of time plus all those  $u$ 's. And what you have is, you are provided with a solution at the starting gate at  $t_0$ .

So, what you are going to see is, this is a set of coupled equation and you could write  $u$  prime equal to some  $a u$ , and that  $a$  would come from this partial derivatives, what we call as a Jacobean. So, this nothing but  $\partial f / \partial u_1 \partial f / \partial u_2$ , and so on so forth. So, you can fill up a matrix and that matrix will have  $n$  Eigen values. **right** We have an order of the system  $m$ . So, we will have  $m$  Eigen values.

What happens is, you can work out a ratio, which we will call as the stiffness ratio; that is nothing but the quotient of maximum for the real part of  $\lambda_i$  and minimum of the real part. So, that basically tells you that, in that system, something is going with this kind of rate with time. That is what it means **right** real part of time. So, we have generic solution and  $e$  to the power  $\lambda_i t$ . **right**



So, the maximum growth rate would correspond to this part growth or decay because we are talking about modulus. So, it could also be a decay rate **also**, and here, the complementary part which will be just on the other end of the spectrum, if you look at this ratio, if this ratio is too far apart, then you have all kinds of problem.

Why because, your numerical requirement will be tied to these extremes. **right** It is almost like a poor man's blanket; you know, you try to cover up this part, that part will be exposed, and vice versa.

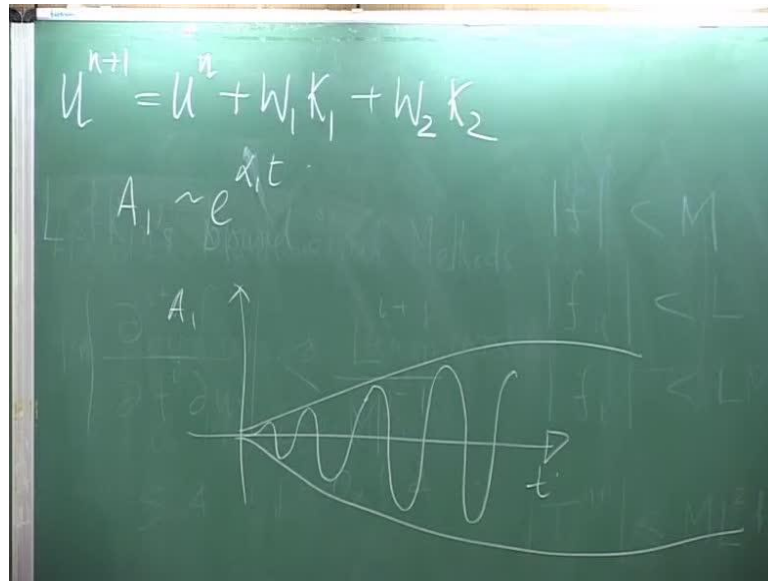
So, what happens is, you have to be really be conscious about it. As I mentioned about this example from chemistry, this could relate to say, simple chemical equation. So, you have a forward part of the reaction and the reverse reaction, and you know the forward part of course goes at a very high rate compared to the reverse reaction in most of the stable reactions, and that gives rise to a regular problem of stiffness issue, and that is what I thought I will explain to you. There was this equation that we have been looking at recently relates to stability of physical systems. This was proposed originally by Landau, some bit of work was done by Stuart, and later, Eckhaus contributed to it.

Eckhaus' contribution comes through this coupling term. So, suppose, let us say, we have a two degree of freedom system given by  $a_1$  and  $a_2$ , and we are looking at its time rate, now what? You notice that this path, the first term, corresponds to the linear growth rate. Suppose I remove the second and third term on the right hand side, then what I get?  $\frac{da_1}{dt} = \alpha_1 a_1$ . So, I can immediately integrate and I can see  $a_1$  goes as  $\lambda_1 t$ ; same way,  $a_2$  goes as  $\lambda_2 t$ .

Now, this term, the second set of term which was a bit of a mystery Landau suggested, but never shown the proof, and he did not completely tell us how he got it, but that seemed to work out very fine for some systems, and you would be surprised to see that people use it across many disciplines; not only for what he intended in fluid mechanics, but it has been used even in quantum mechanics. Landau-Ginzburg equations, people do use it is a variation of this equation.



(Refer Slide Time: 50:48)



So, this second term that you are seeing here, this one, this second term here and the third term here, that gives rise to what is called as a non-linear saturation because this one - the first term, the first term that we have seen, tells us  $A_1$  goes as  $e$  to the power  $\alpha_1 t$ . So, if I plot  $A_1$ , then from a present condition, it will perhaps start growing exponentially. That is what it means. If real part of  $\alpha_1$  is positive, that is what it does. This second term that we are seeing here,  $\beta_1 A_1$ , that has this tendency to saturate that, and this is very often seen in physical system. So, that is a basically non-linear saturation term.

So, that gives you eventually an envelope which goes like this. So, you get a stable system description; in fact, those of you who have seen or attended any course of fluid mechanics, you may have seen that, if you keep a bluff body in a flow, you get this vortex shedding, which comes out in a nice manner. **right** You get alternate from the alternate side; you see that shedding, that is very well captured by this Stuart-Landau equation.

However, for some situations, for some parameters, these do not work very well, and that provoked Eckhaus to add this coupling term which was done by one of our students recently. We did look at it and what we found that whenever you are looking at it, let us say you have predominantly a single degree of freedom system  $A_1$ , but somehow another

mode also comes into play, which you cannot neglect for all time. So, then, you need to worry about this third term here and this second term here.

However, now, what happens? You see, a 1 may go at a higher rate and a 2 contributes very little, but still, its presence has to be there. It is exactly like your forward and reverse reaction. You want to talk about the equilibrium; you will have to keep both; so, here is a similar scenario to maintain the equilibrium. You need to have these cross-coupling terms, and what happens is this coupling is rather weak, and the moment this coupling becomes weak, you are looking at a large value of this stiffness ratio.

(Refer Slide Time: 53:45)

### Solution Methods of SDE

- Consider the system of two ODEs given below

$$\frac{dy_1}{dt} = -1001y_1 + 999y_2 + 2 \quad (71a)$$

$$\frac{dy_2}{dt} = 999y_1 - 1001y_2 + 2 \quad (71b)$$

- with the exact solution as:

$$y_1(t) = e^{-2000t} + e^{-2t} + 1 \quad (72a)$$

$$y_2(t) = -e^{-2000t} + e^{-2t} + 1 \quad (72b)$$

- Note that the solution has dissimilar rates of decay for the two transients  $e^{-2000t}$ ,  $e^{-2t}$  with the former decaying rapidly to zero.
- This can be solved by taking very small  $\Delta t < 0.0014$  and this restriction depends on  $SR$ .
- Trapezoidal method can be used to circumvent this restriction.

Foundations of Scientific Computing-Module 2 – p. 59/62

So, what? You find that, when you are looking at the dynamics of multi-model system is not dominated by single mode, or if you are looking at higher order differential equations, you would often face stiffness of differential equations, and this is not imaginary. This keeps happening. So, to illustrate it, we have chosen a couple of ODEs looks little odd in terms of these coefficients, but you solve it out; it looks like this.

This is a synthetic problem. So, you can see, we have purposely concocted this solution to really work out two fundamental modes. One decays with time as  $e$  to the power minus 2000  $t$ ; the other one is much more gentler in its decay. Its  $e$  to the power minus 2  $t$ . So, one is decaying rapidly as compared to the other.

So, you might as well say why should we even bother; this is anyway going to disappear in a short time. So, we do not really need to worry about  $e$  to the power minus 2000  $t$ , but then, life is not always that simple; you will not guess the solution, if you knew you wouldn't be doing, you would be given, handed down, a couple of coupled equations; you will have to be solving it and if you solve it, you would find that the time step requirement that comes out from the numerical instability point of view, which we will talk about later, puts a very restrictive condition on the amount of time increment that you can choose. So, for this equation, depending on this stiffness ratio, now, you can see this is about 1000 and the time step is restricted to 0.0014.

So, even for such a simple looking linear equations, you would have to spend a lot of effort. You cannot just simply say I will not worry about it. If you say that, you will see to your peril that solution process will break down. You decide to take anything larger than this. This value of  $\Delta t$  essentially tells you, that you are capturing this; without capturing that, you cannot go ahead.

Now, mathematically speaking, this is about the coupling of the equation. We have two modes and when we compute, we also have what we call as the round-off error. You cannot compute without round-off error, and what happens is, round off error would always be there because your set of equations of both the modes, although this mode by itself does not say its importance, but when you are trying to obtain a numerical solution, you have to do the same thing that you do analytically also. You cannot just simply get a solution without both the modes being present. right

So, what you need is, we need to keep solutions where all these modes remain, retain their linear independence. right If one part of the solution becomes dependent on the other, then you have numerical contamination. The solution breaks down and that is what this stability limit is telling us; that, if I do not take small steps, I would not be able to retain both the modes, and I will lose linear independence. So, what happens is the parasitic error which is given by this riding of the small sources of error, like round-off error rides, on what seems to be benign and does not affect the solution, but those errors ride on those unmodeled paths, and eventually solution blows up; that is what it says.

So, classically, what is to happen is people used to use a method called trapezoidal method, to solve at least this class of problem given here.

(Refer Slide Time: 58:07)

### Trapezoidal Method for SDE

- Consider the system of two ODEs given below

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2)$$

$$\frac{dy_2}{dt} = f_2(t, y_1, y_2)$$

- The Trapezoidal method for these equation becomes:

$$y_1^{n+1} = y_1^n + \frac{\Delta t}{2} [f_1(t^n, y_1^n, y_2^n) + f_1(t^{n+1}, y_1^{n+1}, y_2^{n+1})]$$

$$y_2^{n+1} = y_2^n + \frac{\Delta t}{2} [f_2(t^n, y_1^n, y_2^n) + f_2(t^{n+1}, y_1^{n+1}, y_2^{n+1})]$$

- Above SDE's being linear, it is easy to implement the trapezoid method explicitly. It allows increasing time-steps by almost two orders of magnitude.

So, if I talk about the system of two equations here, so,  $\frac{dy_1}{dt}$  equal to  $f_1$  and  $\frac{dy_2}{dt}$  equal to  $f_2$ , then the trapezoidal method would involve evaluating this right hand side as weightage of the function evaluated at the current time level and the function evaluated next time.

So, basically, this is how you go about, and if the equations are linear, it works fine, but if there are non-linearities, we will have to work little harder, and in some cases, even for linear equation, this does not work out. So, this is what we are going to talk about tomorrow's class.