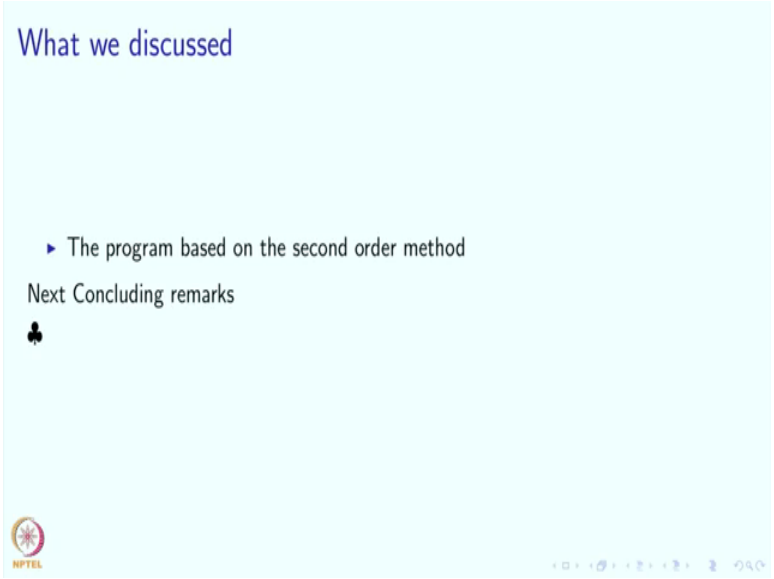
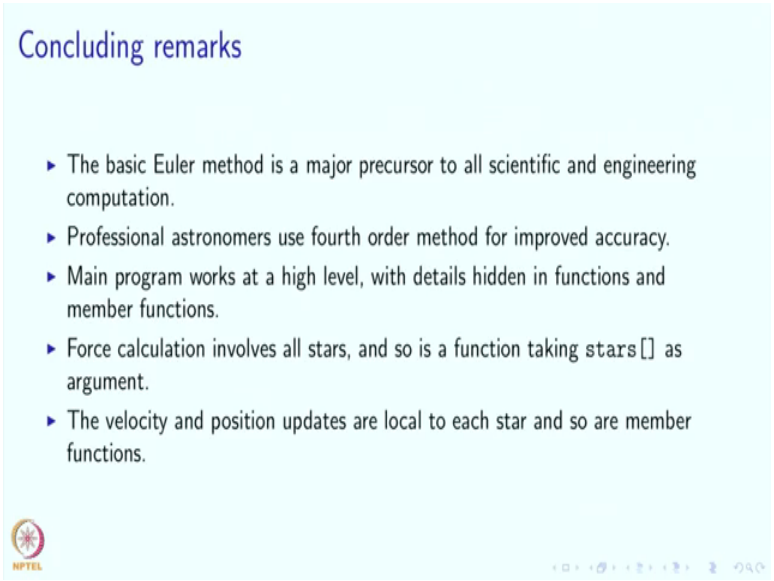
An Introduction to Programming through C++
Professor Abhiram G. Ranade
Department of Science and Engineering
Indian Institute of Technology Bombay
Lecture No 27
Cosmological Simulation
Part - 4
Concluding Remarks

(Refer Slide Time: 0:19)



Welcome back, in the previous segment, we saw a program based on the Second-Order Method and we also saw a demo of it. In this segment, we are going to conclude this lecture.

(Refer Slide Time: 0:39)

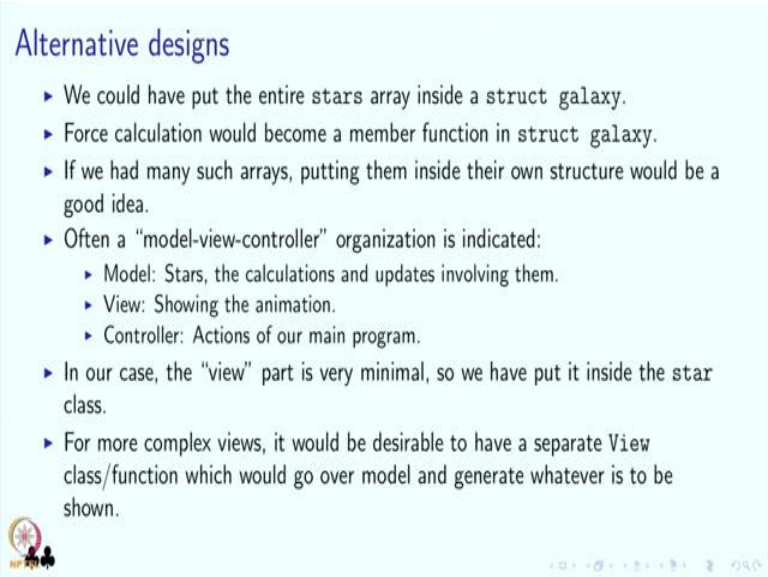


The main thing that we saw in this is a way of evolving the state of a system and we began by looking at the basic Euler method. In some sense, the basic Euler method is a precursor to all Scientific and Engineering calculations. So the idea that I can, I can find the state at the next step by taking the derivative is a very fundamental very important idea. And then the Second-Order Method and the other methods that people use today which are more sophisticated are sort of are developments on this basic insight.

So the insight is that somehow knowing what happens at time t, we can use derivative like information to predict what happens at t+delta. And this, this is sort of at, at the base, you can say that this is at the base of everything that goes on almost. Yeah, so we showed you a code based on the Second-Order Method. Now, this is better than the First-Order Method, but professional astronomers use the Fourth-Order Method. So effectively that means keeping 4 terms from the Taylor series. And the reason for that is that with a Fourth-Order Method, you can take bigger time steps. So while the calculation of a single time step becomes more complicated the savings that result because you have to take fewer time steps are much more. Now, let me comment a little bit on the program structure for this program. And let us see, let me first observe that our main program does work at a high-level and the details are hidden in functions and member functions. So this is what we wanted and we have accomplished it to a certain extent. Now, the force involves all stars, and so it is a function taking stars the array stars as an argument. And the velocity and the position updates are local to each star, and so the

updates are member functions. So in some sense, once you decide where you are going to place the data, then the positions of the functions may also often be naturally indicated.

(Refer Slide Time: 3:37)



Now I should point out that designing a program is an, is an art, not a perfect science. So there can be other ways of writing the same program. So, let me, let me indicate what the other designs might have been and of course, these are only a few of the other possibilities. So one possibility is that perhaps we could have put the entire stars array into a struct we could which we could fancifully call Galaxy. Because after all, that entire collection also is an important entity. And if we had done that, , the force calculation would become a member function inside our struct Galaxy. And we have not done this right now, but this would definitely be a thing worth doing if we had many entities like galaxies. So if we had if we had several kinds of arrays, then instead of exposing all those arrays in the main program, it would have been better to keep it to only have a struct, a container for each array and let the main program see the container. And the details of what is inside will be hidden in member functions of that container. And in fact, in some of our earlier programs, we have done that. Then the second alternative design that might be possible is based on an idea which is, which has been described as a model-view controller design. So what does this mean? So this says that the system that you are simulating or evolving over time contains a mathematical model, a contains a mathematical description which indicates how the system actually changes.

And then there is some code, or some data needed to show all that is how the system is changing on your screen and that is the view part. And then there is the controller, the controller is the one which commands the model to change and it commands the view to show whatever is needed to be shown on the screen. You should have these 3 parts nicely separated. And I should comment that we have not quite done it. Let us see what we have done? So, we do have a model part, the stars and the calculations and updates involving them sort of constitute our model. And what is our view? The view is showing the animation. So our view code and our model code are mixed up, and I guess purists would frown on it. But the point over here is that the animation, the showing the animation is really tiny. So every time we update a star, that update changes the star to change its position. And the and the change in in the view is not very complicated. And, therefore, we have just put that view update as a part of our model update. But if you have a very complicated model and a complicated view that you need to generate, then certainly writing the code for the view in one class and writing the code for the model in another class would be a good idea. And the controller in this case is our main program itself. So in some sense this this model view controller organization, even if it is not there, you can see how it could be there in our main program, in in our program.

Yeah, so as I said in our case, the view part is very minimal, so we have put it inside the star class itself. The star class does the work of the model as well as the view. Yeah, for more complex views, it would be desirable to have a separate view class or function, which would go over the model and generate whatever is to be shown. So that concludes our discussion of this cosmological simulation. It is discussed at the book in chapter 19, and there are some problems after the chapter as well. So I will invite you to solve the problems. And as I said simulation is a very important part, taking, taking the system and evolving it is an is a very important part of the whole range of computations that people perform and what we have tried to do over here is give you a glimpse of it. So that is it and that is the end of this lecture. Thank you.