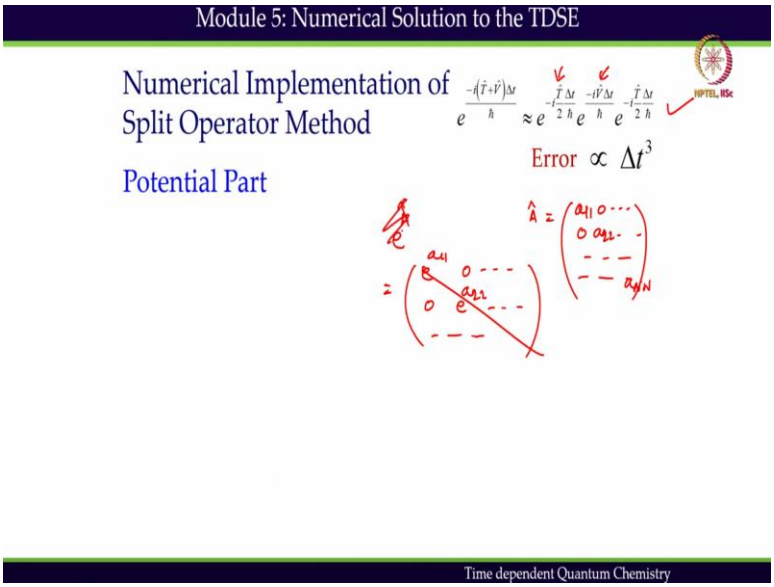
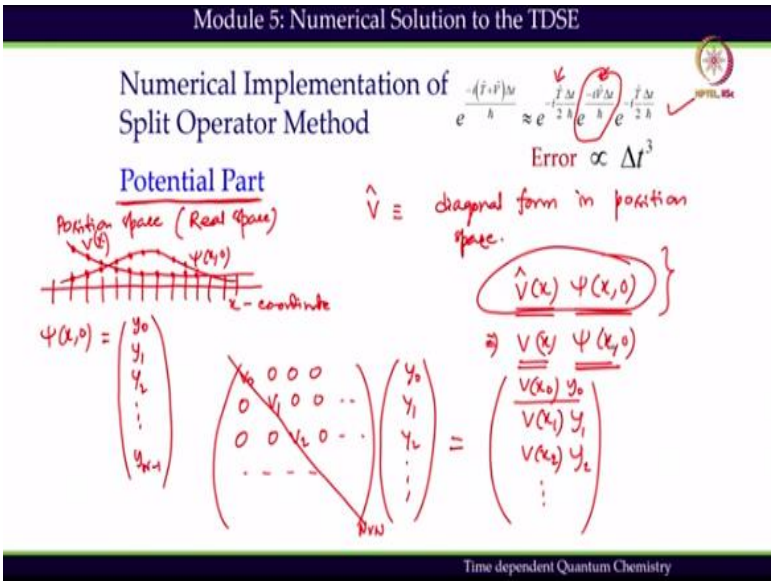
Time Dependent Quant Chemistry
Professor. Atanu Bhattacharya
Department of Inorganic and Physical Chemistry
Indian Institute of Science, Bengaluru
Mod 05 Lecture 34
Numerical Implementation of Split Operator Method

Welcome back to Module 5, where we are continuing a numerical implementation of time evolution operator through which we are going to get the solution of TDSE. And so far we have shown that the symmetrized product, the split operator approach we are going to use and symmetrized product is going to give us the minimum error if we take the time step to be very small. And we will move on.

(Refer Slide Time: 00:53)





So for that implementation, we have to note that the final numerical implementation of symmetrized split operator base time propagator, which we have shown here, requires to present the kinetic and potential energy matrices in the diagonal form, because that is the trick we are going to use. Previously, we said that if a operator is, can be represented in its diagonal form, then I can very easily find out the exponential of the operator by exponentiating each element as this one, like this way.

$$\widehat{A} = \begin{bmatrix} a_{11} & 0 & 0 & . & . & & & \\ 0 & a_{22} & 0 & . & . & & & \\ 0 & 0 & a_{33} & . & . & & & \\ . & . & . & . & . & & & \\ . & . & . & & . & & & \\ & & & & & . & & \\ & & & & & . & . & . \\ & & & & & . & . & a_{NN} \end{bmatrix}$$

$$e^{A} = \begin{bmatrix} e^{a_{11}} & 0 & 0 & . & . & & & \\ 0 & e^{a_{22}} & 0 & . & . & & & \\ 0 & 0 & e^{a_{33}} & . & . & & & \\ . & . & . & . & . & & & \\ . & . & . & & . & & & \\ & & & & & . & & \\ & & & & & . & . & . \\ & & & & & . & . & e^{a_{NN}} \end{bmatrix}$$

So we need to convert now this kinetic energy operator and potential energy operator in the diagonal form, and we have to check what is the general form we get in the grid representation.

So, we have to remember that we are using the grid representation to represent all these operators in the matrix form. In the position space, first we will deal with the potential part, because it is easy to deal with. So, this part we will deal with first, the middle part.

In the position space, what is position space? Position space, as we have shown, is an x coordinate, this is x coordinate. And in the x coordinate, we have made this grid. So, this is called position space, many times, many occasions, this is also called real space. So, in position space, grid representation, which we have already seen in different, in Python chapter 2, as well as in this module, also we have shown how to prepare the grid in the position space. The potential energy operator can be represented by a diagonal matrix directly. So, that is the good news.

This potential energy operator has a diagonal form in position space. So, it has already a diagonal form. And because it has a diagonal form, one can get the exponential of the potential energy matrix directly by evaluating or exponentiating each diagonal element. So, that is quite straightforward in the method.

So, in the position space grid,

$$\Psi(x,0) = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ . \\ . \\ . \\ . \\ y_{N-1} \end{bmatrix}$$

details of this discretization has been discussed already in Python tutorial 2 and one can review that one quickly.

So, this is the representation, discretized representation of the wavefunction. And if we have this discretized representation of the wavefunction, remember the right hand part of the TDSE is nothing but H**Ψ** and H**Ψ**, it, sorry, not TDSE, but the time evolution. In the time evolution, we have e to the power, I mean, I will show this. So, we have this potential part. So, potential operator, when potential operator is acting on wavefunction, let us say x,0 when it is acting on it,

it is potential, so one can assume that I have a wavefunction like this, so this is your **Ψ**(x,0) and potential let us say I have like this, this is V(x).

So potential, we see that, and if we multiply this function by this function, if we, potential operator is nothing but a multiplication operator. So, this is nothing but

$$V(x)\boldsymbol{\Psi}(x,0)$$

It is a multiplication operator. And if I multiply this potential operator, by this, multiply this wavefunction by this potential operator or potential function, then this is nothing but multiplying individual points on the grid.

So, what we have, we have individual grid discretized potential values and each grid points the values of those individual functions can be multiplied. If we multiply then the, this function will be obtained. So, because the, this part, metrix representation of this part would be simply

$$\begin{bmatrix} V_1\Psi_1 \\ V_2\Psi_2 \\ V_3\Psi_3 \\ . \\ . \\ . \\ . \\ V_{N-1}\Psi_{N-1} \end{bmatrix}$$

like this. It is just multiplying corresponding elements.

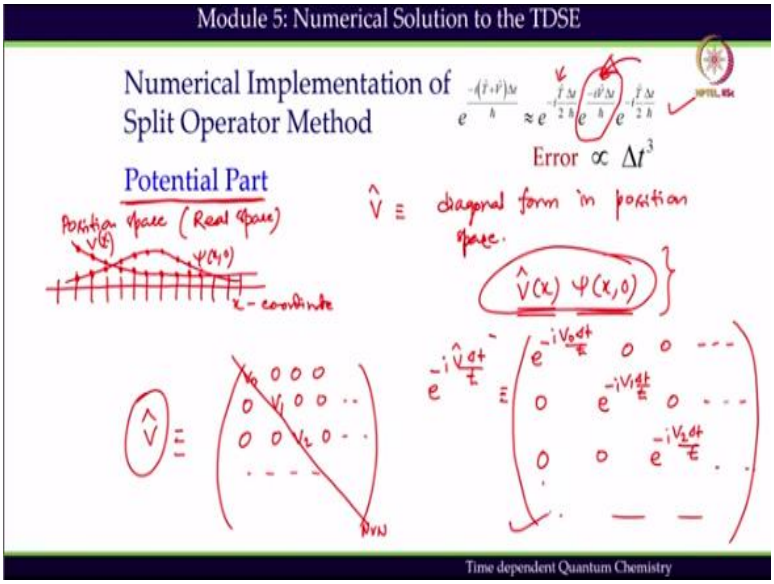
And if it is so, then this can be presented as a diagonal matrix.

$$\begin{bmatrix} V_0 & 0 & 0 & . & . & & & \\ 0 & V_1 & 0 & . & . & & & \\ 0 & 0 & V_2 & & & & & \\ . & . & . & . & & & & \\ . & . & . & . & & & & \\ & & & & & & . & . \\ & & & & & & . & V_{N-1} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ .. \\ .. \\ .. \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} V_0 y_0 \\ V_1 y_1 \\ V_2 y_2 \\ . \\ . \\ . \\ . \\ V_{N-1} y_{N-1} \end{bmatrix}$$

And we can multiply this. Why we can multiply this, is because matrix multiplication, rule of matrix multiplication is that we multiply this and this, then we add this and this. But because every elements are 0, we get back only this value. This comes, this is direct result of the matrix multiplication. And because of this matrix multiplication, nature of matrix multiplication, what we get is N*N matrix we get in a diagonal form.

So, potential operator is a, is a diagonal matrix. And because it is a diagonal matrix, one can directly write down the exponential of the potential metrics as follows.

(Refer Slide Time: 08:24)



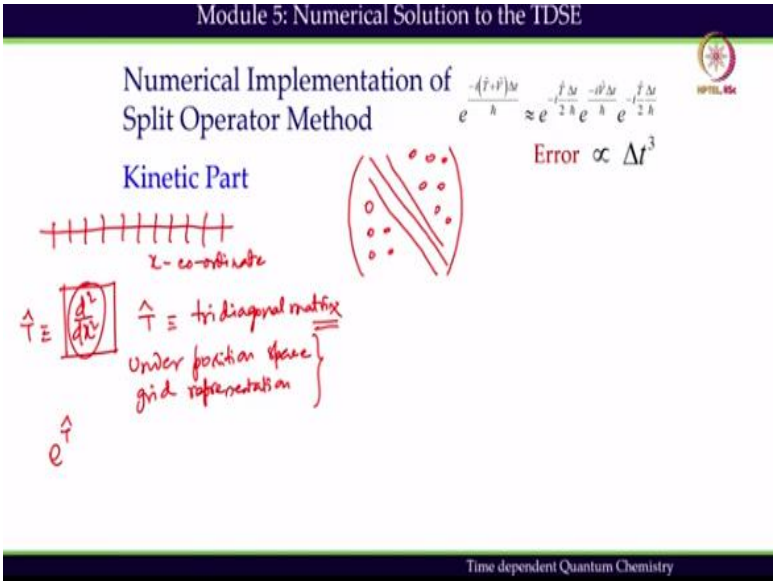
I have this potential matrix V which is in a diagonal form already. So, I can write down propagator, potential part of the propagator at

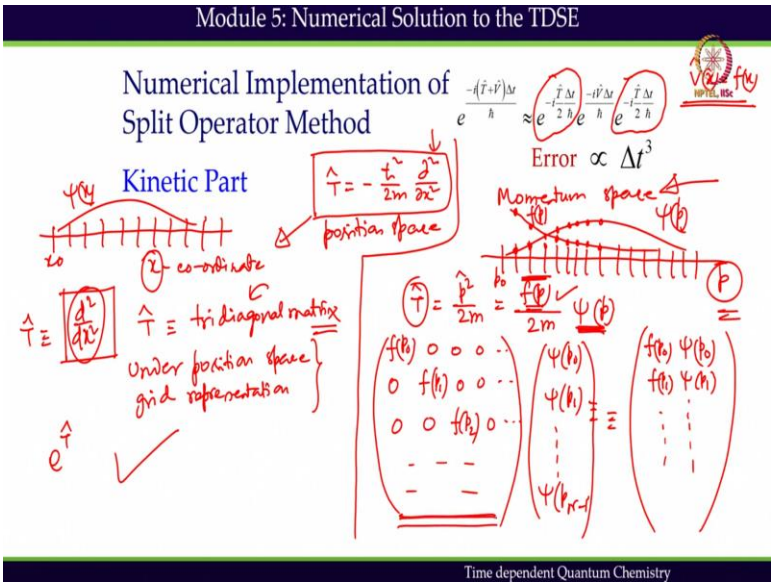
$$e^{-i\frac{2\pi V\Delta t}{h}} = \begin{bmatrix} e^{-i\frac{2\pi V_0\Delta t}{h}} & 0 & 0 & . & . & & \\ 0 & e^{-i\frac{2\pi V_1\Delta t}{h}} & 0 & . & . & & \\ 0 & 0 & e^{-i\frac{2\pi V_2\Delta t}{h}} & . & . & & \\ . & . & . & . & . & & \\ . & . & . & & . & & \\ & & & & . & & \\ & & & & . & . & . \\ & & & & . & . & e^{-i\frac{2\pi V_{N-1}\Delta t}{h}} \end{bmatrix}$$

it is going to be a diagonal matrix.

So, potential form, sorry, the potential part of this propagator is very simple, under grid presentation, it gets a diagonal form, the potential operator gets a diagonal form. So, I have this exponential part very easily evaluated.

(Refer Slide Time: 09:32)





What about the kinetic energy part? Kinetic energy part has, we have to imply certain trick here. And what is the issue? Issue is that in position space grid. So, we write down the position space grid like this, where we have x coordinate. We have seen that on the positions grid, kinetic energy, we have represented before actually, kinetic energy has this derivative, kinetic energy is a derivative operator. And for the derivative operator, for the evaluation of the derivative operator, we have used in the, in chapter, in module 4, we have used finite difference method to represent this derivative operator.

And in the end, this, because it is a second derivative, the T operator has adopted a tridiagonal matrix, tridiagonal matrix. This has been presented already in module 4, in the last module. And what does it mean by tridiagonal matrix? This potential operator looks like I have a diagonal value, I have upper diagonal and lower diagonal. All these values are present remaining part are 0. That is why it is called tridiagonal, remaining parts are 0. So, this has adopted a tridiagonal matrix form under position space grid representation, under position space grid representation.

And because it is a triadiagonal matrix, e^T is not a straightforward way of evaluation, it is not possible, we cannot use that trick, which we have used for the diagonal matrix. So, what we can do is that instead of this position space, we will look at the exact form of the, this is nothing but

$$\hat{T} = -\frac{h^2}{8\pi^2 m}\frac{\partial^2}{\partial x^2}$$

This is for the position space. This is the form of position space.

If I express the kinetic energy operator in that position space, then I get this form, immediately I get this second derivative and second derivative only will give me, using finite difference method, only give me this tridiagonal matrix form. But instead of position space, if I use the momentum space, what is momentum space?

Similarly, I have this momentum p as a function of momentum, here is a function of x I have now as a function of p, I have this grid representation, same, idea is the same, I have grid representation. In the momentum space, if I try to represent T, what is the form of the kinetic energy operator in the momentum space?

$$\hat{T} = \frac{\hat{P}^2}{2m} = \frac{f(p)}{2m}$$

It is just like the V in the position space, it is a function of x in position space. So, potential energy is a function, simple function of position space. Similarly, kinetic energy operator is nothing but a simple function of position, momentum space. And because it is a simple function of moment space, when I employ that on a wavefunction, what is going to happen? The

wavefunction now has to be here, I have represented the wavefunction in the position space. Now I have to represent the wavefunction in the momentum space.

If I represent the wavefunction in the momentum space, then it is nothing but the multiplication of two functions, one function is coming from the operator, one function coming from the wavefunction. So, here also, if I have this kinetic energy operator represented in the moment space, then multiplication of these two functions is nothing but the element-wise multiplication of each point on the grid, but now the grid is momentum grid, it is not a position grid anymore.

So, element wise multiplication. So, which is nothing, but I have now $\boldsymbol{\Psi}$, but the remember psi previously, we have represented in the position space, now we are going to the represent, we have to represent in the momentum space, that is

$$\begin{bmatrix} \Psi(p_0) \\ \Psi(p_1) \\ \Psi(p_2) \\ \Psi(p_3) \\ .. \\ .. \\ .. \\ \Psi(p_{N-1}) \end{bmatrix}$$

This is a column matrix again in the momentum space. And then this is nothing but a multiplication so if it is multiplication, then finally I get this value.

$$\begin{bmatrix} f(p_0)\Psi(p_0) \\ f(p_1)\Psi(p_1) \\ f(p_2)\Psi(p_2) \\ f(p_3)\Psi(p_3) \\ .. \\ .. \\ .. \\ f(p_{N-1})\Psi(p_{N-1}) \end{bmatrix}$$

It is just multiplication.

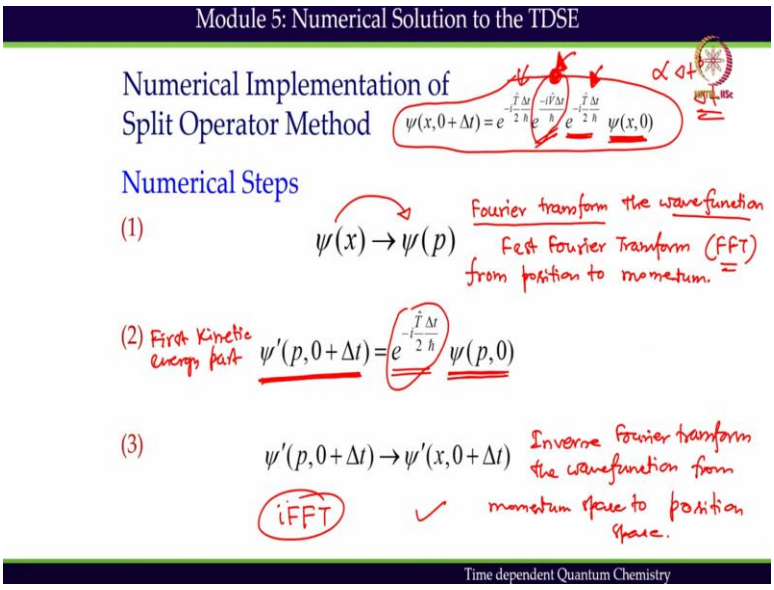
And if it is multiplication, then matrix form would be like this. I have a diagonal matrix.

$$\begin{bmatrix} f(p_0) & 0 & 0 & . & . & & \\ 0 & f(p_1) & 0 & . & . & & \\ 0 & 0 & f(p_2) & & & & \\ . & . & . & . & & & \\ . & . & . & . & & & \\ & & & & . & . & \\ & & & & . & f(p_{N-1}) \end{bmatrix} \begin{bmatrix} \Psi(p_0) \\ \Psi(p_1) \\ \Psi(p_2) \\ \Psi(p_3) \\ .. \\ .. \\ .. \\ \Psi(p_{N-1}) \end{bmatrix} = \begin{bmatrix} f(p_0)\Psi(p_0) \\ f(p_1)\Psi(p_1) \\ f(p_2)\Psi(p_2) \\ f(p_3)\Psi(p_3) \\ .. \\ .. \\ .. \\ f(p_{N-1})\Psi(p_{N-1}) \end{bmatrix}$$

So, we get the diagonal form.

So, what is the basic idea of what I have shown here is that if I want to use this part and this part in my calculation, I cannot do it in the position grid. I have to do it in the momentum space. In the momentum space, so the bottom line here is the momentum space, kinetic energy gets a diagonal form, and because kinetic energy gets a diagonal form, we have to convert it to the momentum space.

(Refer Slide Time: 17:24)



So, the overall scheme is following now, we are going to write down the entire scheme. So, this is the mathematical or analytical form, approximate form of the split operator method,

$$e^{-i\frac{2\pi H\Delta t}{h}} \approx e^{-i\frac{2\pi T\Delta t}{h}} e^{-i\frac{2\pi V\Delta t}{h}} e^{-i\frac{2\pi T\Delta t}{2h}}$$

where we have used a symmetrized product, symmetrized product, it is symmetrically arranged, here is the kinetic energy part, here is a kinetic energy part, and remaining part is a potential energy part here. And note this half, because there are -- is divided by 2, kinetic energy is somehow divided by 2 and then they are distributed symmetrically.

Now, this is the analytical form of the split operator method, which we are going to use. For this, we know that error is going to be 10 to the, delta t to the part 3, which means that error would be very small if I take the delta t to be very small. So, the first cycle of this numerical step would be, as you can see that this operator has to act on this wavefunction. So, first step will take to be wave function has to be converted to the momentum space.

$$\boldsymbol{\Psi(x) \rightarrow \Psi(p)}$$

So, that is going to be the first step, which means that Fourier transform, and that can be done using Fourier transform the wavefunction positioned to momentum conversion of a wavefunction, position presentation to momentum representation of the wavefunction can be done using Fourier transform and that we have already encountered in Python tutorial 3 and how to do this kind of Fourier transform.

So, we will skip those techniques here. We will just mention that Fourier transform is a technique by which one can transform the wavefunction from position space to the momentum space. And one can make use of Fast Fourier Transform algorithm for this. This is called Fast Fourier Transform, it is called FFT algorithm to transform this one. Why you need to transform this one? Because I have to use this momentum operator in the Fourier domain, in the momentum domain, and that is why.

So, now, if I use this first kinetic energy part, this is the first kinetic energy part. The first kinetic part of the propagator that will be employed here as you can see.

$$\boldsymbol{\Psi'(p, 0 + \Delta t) = e^{-i\frac{2\pi T \Delta t}{2h}} \Psi(p, 0)}$$

But remember, here we have, we are using this in the momentum space and we are getting propagating the wavefunction, that is easy to do, because this part is a diagonal form now. Once we have got this propagation due to the first part of the wavefunction, remember, next operator is

going to be this part which is potential. And our potential has to be done, potential operator, this has to act in the real space.

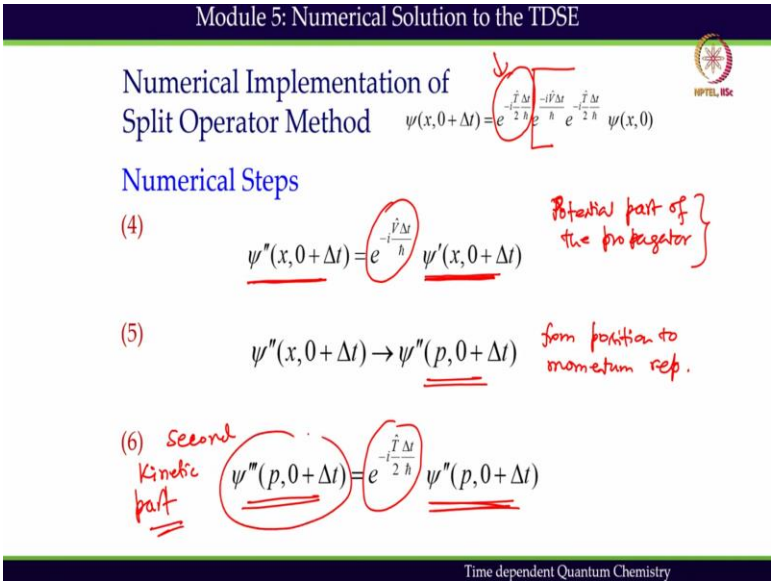
Because it has to act on the real space, all we have to do is that we have to use inverse Fourier transform, inverse Fourier transform the wavefunction from momentum space to position space again. Because this operator will act in the position space. That is why wavefunction has to be converted.

$$\Psi'(p, 0 + \Delta t) = \Psi'(x, 0 + \Delta t)$$

Here we have converted from, the first step we have converted from position our real space to momentum space.

But here, we are doing the inverse of Fourier Transform. And inverse Fourier transform can also be done with the help of this, the same Fast Fourier Transform algorithm, which is called iFFT inverse Fourier Transform algorithm can be used.

(Refer Slide Time: 22:29)



Once we have got the propagated wavefunctions in the position space, we will employ this kinetic. Sorry, this is potential part. So, this is the potential part, part of the propagator, that is acting on the position space representation of the wavefunction and we get another time evolution for the position, for the potential part.

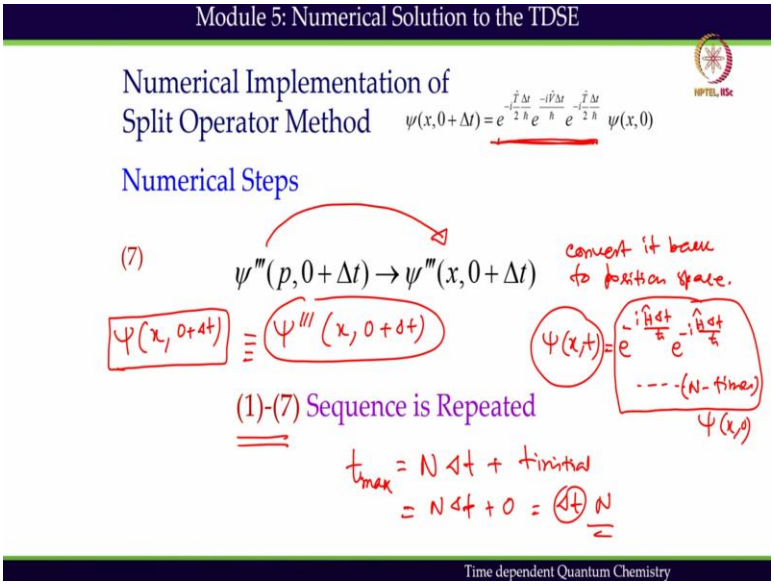
$$\Psi''(x, 0 + \Delta t) = e^{-i\frac{2\pi V \Delta t}{h}} \Psi'(x, 0 + \Delta t)$$

Once we get that, again, so we are done up to this, but then again, I have to employ this kinetic energy part. And in order to do the kinetic energy part, again, I have to move to the momentum part. So, I have to convert it from position to momentum representation of the wavefunction. Once I present it, then I can employ the remaining part of the kinetic part, the second kinetic part I can employ.

$$\Psi''(x, 0 + \Delta t) \rightarrow \Psi''(p, 0 + \Delta t)$$

And I can confirm, I can get this propagation due to, the, this second kinetic part in the momentum space. And once we do that, then finally I get this wavefunction.

(Refer Slide Time: 24:24)



And this wavefunction needs to be converted to the position space again, back to position space.

$$\Psi'''(p, 0 + \Delta t) \rightarrow \Psi'''(x, 0 + \Delta t)$$

So, what I get is that a

$$\Psi(x, 0 + \Delta t) \rightarrow \Psi'''(x, 0 + \Delta t)$$

And it is quite clear now that the entire step which I have presented right now, 1 to 7 steps, the sequence has to be now repeated far getting my desired time.

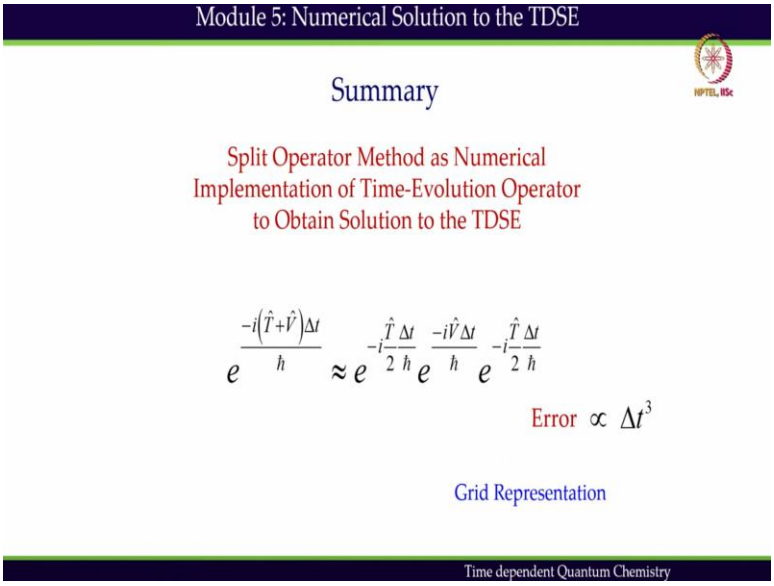
Why I need to repeat, because in the end, I have represented already that this final wavefunction at a particular time t is nothing but product of this small step propagator

$$\Psi(x,t) = e^{-i\frac{2\pi H\Delta t}{h}} e^{-i\frac{2\pi H\Delta t}{h}} e^{-i\frac{2\pi H\Delta t}{h}} \ldots\ldots\ldots\ldots (N\ times) \ldots\ldots\ldots\ldots \Psi(x,0)$$

So, I have to repeat this propagation, Δtshort time propagation, which includes all three terms here, N times I have to repeat so that I can get the desired time.

$$t_{max} = N\Delta t + t_{initial}$$
$$= N\Delta t + 0 = \Delta tN$$

(Refer Slide Time: 27:17)



We have come to the end of this module. In this module, we have presented the numerical implementation, how numerically one can solve TDSE and what is the approach one should use. One should use grid representation. This is not the only way one can get the numerical solution of TDSE, this is one efficient way, very frequently used way of getting numerical solution of TDSE.

And what we have shown is that split operator approach can be very useful, what is the approximation behind split operator approach, we have presented that. And we have seen that the symmetrized product of the split operator is going to be more efficient because it is, its error associated with delta T^3. And we will, we will see how to implement this. In the Python programming, we have to see the implementation of this split operator approach and we will see it in one of the Python module. In fact, in the next Python module, we will see that. We will stop here, we will meet again for the next module.