Enhancing and testing quantum finite-difference time domain code

Stan Hatko

Dept. of Physics, University of Ottawa

Introduction

The UROP research project that I was working on in the Computational Nanophotonics Group was about the interaction of an atomic cluster with an intense laser pulse. Since the theoretical knowledge about this interaction is limited, a numerical simulation of the evolution of the electron wavefunction under the influence of a high intensity laser pulse was studied and Schrodinger's equation was solved numerically.

Exact solutions of Schrodinger's equation can be obtained for a hydrogen atom (and hydrogen-like atoms), but more complex systems do not have closed-form solutions and must be solved numerically.

The Computational Nanophotonics Group developed a scientific software program called Quantum Finite-Difference Time Domain (Quantum FDTD) for simulations of the clusters exposed to the laser beam.

Methods

Quantum FDTD simulation starts with a known initial state and a configuration file specifying the parameters of a system. The derivatives in Schrodinger's equation are approximated using a finite difference, and using this approximation the next state is determined.

There are three methods implemented in Quantum FDTD:

- the explicit method,
- the implicit method,
- the Crank-Nicolson method.

The explicit method is the simplest method. This method has an OpenCL implementation that runs on a GPU (Graphics Processing Unit). The disadvantage of the explicit method is that a very small step size is required for an accurate simulation.

The implicit method is not used for the simulations because it creates a lot of noise and is inferior to the Crank-Nicolson method which works much better.

The Crank-Nicolson method is a numerically stable hybrid of the explicit and implicit methods that is well suited to imaginary time simulations.

Results

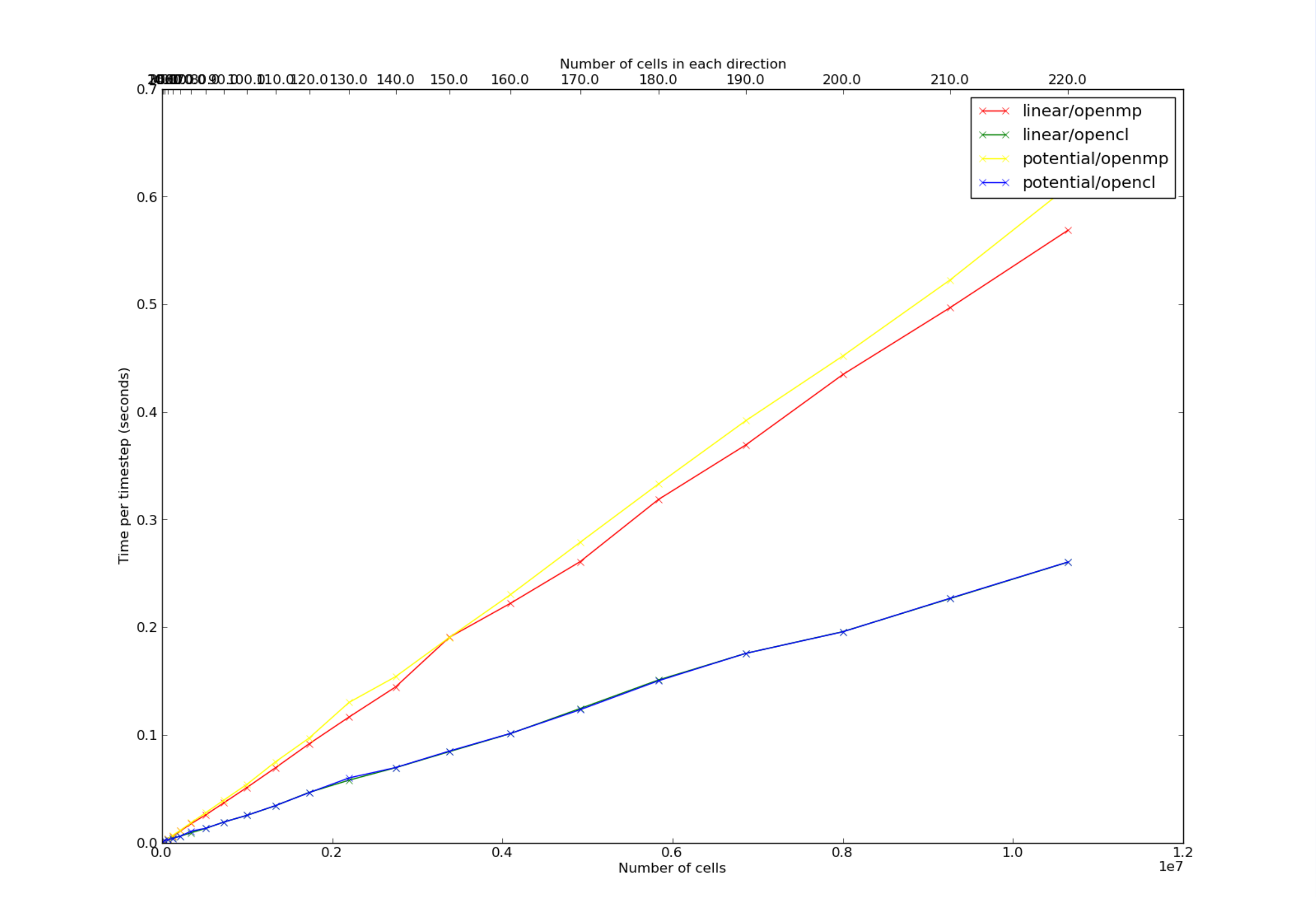
My UROP research contribution was to enhance the Quantum FDTD software by adding new subroutines and verifying the operations of the existing code. I also wrote some post-processing scripts to facilitate data analysis. With my help, an OpenCL implementation of the explicit method has been completed, enabling the code to run on a GPU (Graphics Processing Unit), greatly improving its performance (see Figure 1).

Recently, a laser has been implemented for Quantum FDTD code and I performed simulations with the laser and without the laser. In these simulations, the starting time, duration, wavelength, and intensity of the laser pulse can be configured. I tested the laser for a hydrogen atom using two methods.

- In the first method, the hydrogen system is started in the ground state (n=1), and before the laser pulse starts, the system is checked to ensure it remains in the ground state. Then a laser pulse is sent, and after the laser pulse ends, the spectrum of the system is found. To validate the laser code, the spectrum should contain higher energy states of hydrogen (n=2, 3, ...), which was confirmed in my testing.

- The second method consists in initially filling the system with random noise and then computing the inner product of each state with the excited states in the system. The expected outcome is that, after the laser pulse is sent, there should be an increase in the inner product with higher energy states. Some initial testing has been performed to validate this result. I also performed a preliminary test of a cluster of atoms exposed to a laser beam.

Figure 1. Performance improvements of Quantum FDTD code running on a system with GPU using OpenCL.



Conclusions

Working on the UROP project gave me an opportunity to devise strategies for implementing and testing numerical solutions to Schrodinger's equation using the Quantum FDTD code and analyzing the electron wavefunction over time. It gave me a deeper understanding of quantum mechanics and laser science, beyond my current coursework.

Developing, implementing and testing new subroutines on the advanced supercomputer platform in the Computational Nanophotonics Lab enhanced my computational algorithm development and programming skills. My work on verification and performance improvements of the Quantum FDTD code give me more insight and understanding of complex systems benchmarking and validation.

Acknowledgements

I'd like to thank my UROP project supervisor, Dr. Lora Ramunno, for her help and guidance. My research was performed in Dr. Ramunno's Computational Nanophotonics Group where I had a great opportunity to interact with, and learn from all the members of the group. I'd like to thank everybody, especially Nicolas Bigaouette, for many technical discussions and collaboration during the project..

Author: Stan Hatko, Physics/Math,
Email: shatk094@uottawa.ca

References

Computational Nanophotonics Group website:
http://mysite.science.uottawa.ca/lramunno/index.html

Figure 2. An example of electron wavefunction generated by Quantum FDTD post-processing scripts.

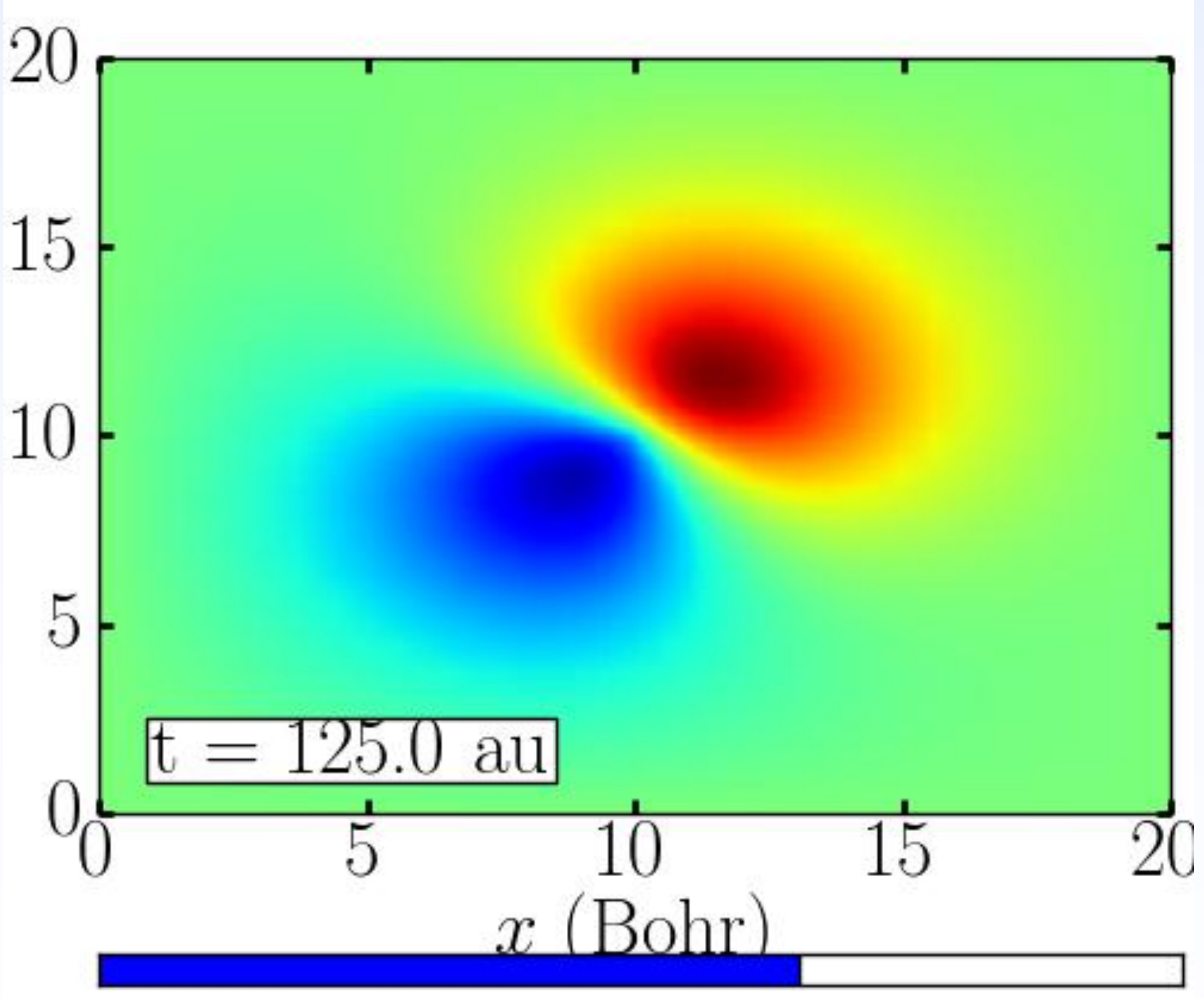


Figure A1. Wavefunction of hydrogen atom, t = 0.0 au

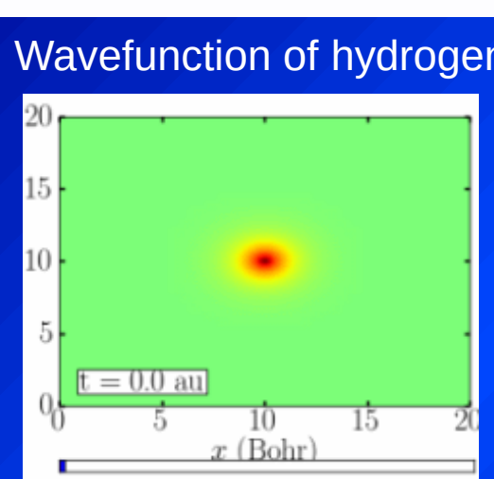


Figure A2. Wavefunction of hydrogen atom, t = 55.0 au

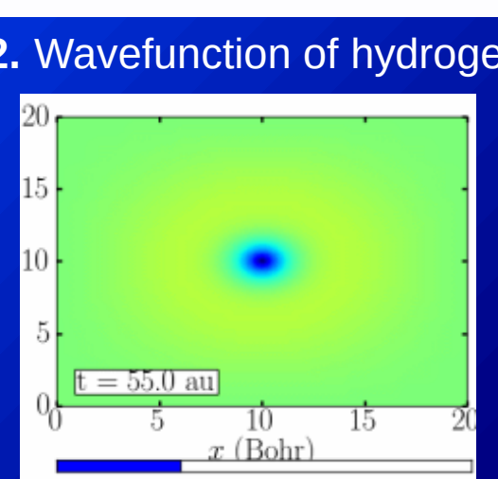


Figure A3. Wavefunction of hydrogen atom, t = 60.0 au

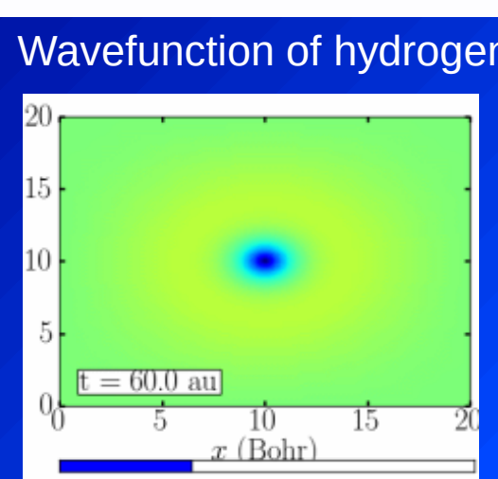


Figure A4. Wavefunction of hydrogen atom, t = 125.0 au

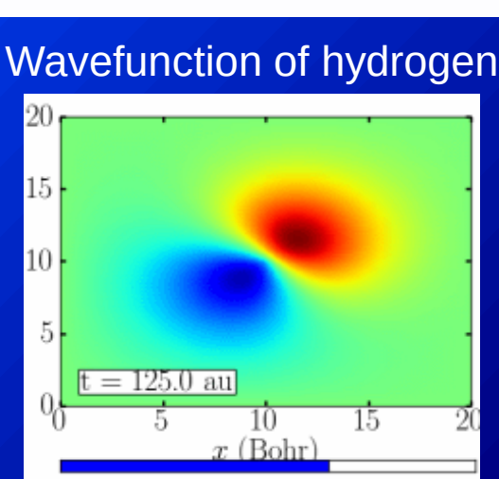


Figure A5. Wavefunction of hydrogen atom, t = 140.0 au

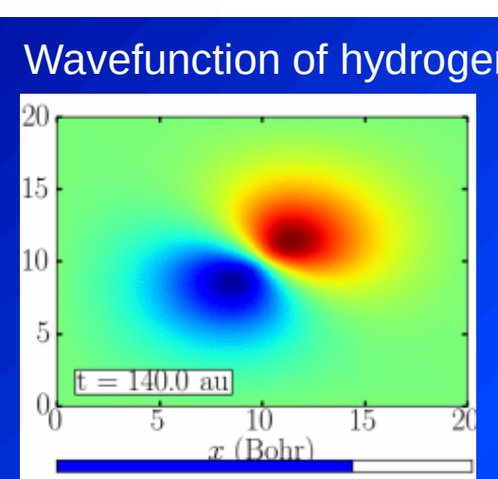


Figure B. General form of time-dependent Schrodinger equation

$$i\hbar\frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$