GPU-Accelerated Variational Monte Carlo

B. Tasseff1,4, N. Setty2, N. Sengupta3, Z. Yun1, S. Abu Asal2,4, Y. Fang2,4, S. Pathak3, J. Moreno3,4, J. Ramanujam3,4

1Department of Physics, University of Northern Iowa 2Department of Electrical & Computer Engineering, Louisiana State University 3Department of Physics & Astronomy, Louisiana State University 4Center for Computation & Technology, Louisiana State University

How can we model and describe the behavior of strongly-correlated electronic systems (e.g., high-temperature superconductors) in an accurate, computationally-efficient manner?

The Variational Method

In quantum mechanics, the variational method is a strategy used to find and study the ground state of a system, \( \Psi_G \). The procedure is relatively straightforward:

- Start with a random initial configuration and state \( \Psi(\alpha) \).
- Choose an electron at random and move it, producing \( \Psi'(\alpha) \).
- If \( \langle \Psi'(\alpha) | H | \Psi'(\alpha) \rangle \) is more than \( \langle \Psi(\alpha) | H | \Psi(\alpha) \rangle \), compute and store the new energy in the configuration. Otherwise, keep the old configuration and repeat 2.
- Repeat 2 and 3 until reaching an equilibrium energy \( E_\alpha(\alpha) \).

This process is then repeated for many \( \alpha_i \). The parameter value which best minimizes \( E_\alpha \) is then used to build \( \Psi_G \).

Variational Monte Carlo

An electronic system may be modeled as a lattice consisting of atomic sites housing spin-up and spin-down electrons. The configuration of electrons within the lattice implicitly defines the energy of the system.

- The Connection: For every value of \( \alpha \), there exists one configuration with minimal energy, \( E_\alpha(\alpha) \). The value of \( \alpha \) that best minimizes this energy is used to obtain \( \Psi_G \).
- The Problem: A 100-site lattice with 50 spin-up and 50 spin-down electrons will have nearly \( 10^{68} \) possible configurations. It would be difficult to find one configuration with minimal energy \( E_\alpha(\alpha) \).
- The Solution: Randomly sample configurations near each minimal energy. Average these results to predict each minimal energy. Repeat while varying \( \alpha \) to obtain \( \alpha_G \). \( E_G \), and \( \Psi_G \).

Validation

After building the GPU implementation, we first ensured it worked comparably to the CPU version. We also observed the increase in data quality obtained when using multiple Markov processes. Any Markov process (CPU) and 110 Markov processes (GPU).

Results

Finally, to adequately benchmark the GPU implementation, we compared the speeds of CPU and GPU versions of the code.

Discussion

Upon inspection of figure 2, there is a clear increase in data quality when executing multiple Markov processes. However, as shown in figure 3, the current GPU code does not provide a speedup over the CPU code. Nevertheless, the CPU code scales at a much quicker rate than the GPU counterpart.

Conclusions and Outlook

Although we find benefit in executing multiple Markov processes, the current GPU code does not provide improved performance over its CPU counterpart. However, many improvements remain to be made to the foundational GPU code; such optimization may eventually permit a significant speedup over the serial CPU code.

References


Acknowledgements

This material is based upon work supported by the Louisiana Board of Regents contract number LEQSF(2007-12)-ENH-PRSF-P8-01 with additional support from the Center for Computation & Technology at Louisiana State University.