# What is Happening With HF

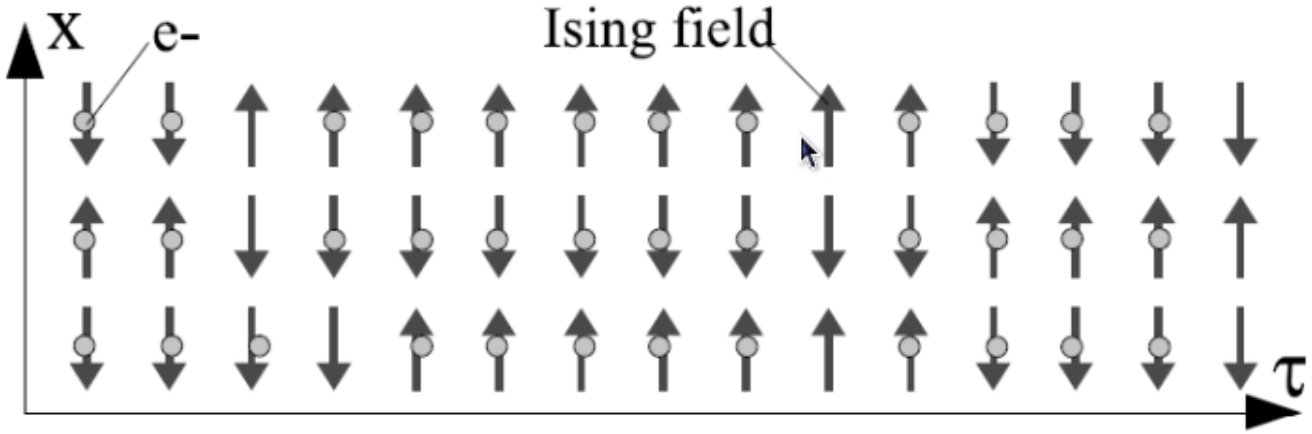
Conrad Moore
Sameer AbuAsal

# Quick Hirsch-Fye Monte Carlo Review

- The weight of an "Ising" decoupling field in HFQMC is

$$\prod_{\sigma=\pm 1} \det G'^{-1}_{\sigma}(\mathbf{s}).$$

where $G_\sigma(\mathbf{s})$ is the Green function for spin $\sigma$ and Ising configuration $\mathbf{s}$.




- Calculates Green functions for the measurement of most one and two-particle quantities.

[1] J.E. Hirsch and R.M. Fye, Phys. Rev. Lett. 56, 2521 (1986)

# Optimizing the Update Equations for GPU

- The updated Green Function $G'$ follows from:

$$G' = G + A \otimes B$$

- Our code implements a delayed (or Ed) update[1]:

$$G_{ij}^{(M)} = G_{ij}^{(1)} + \sum_{N=1}^{M} A_i^{(N)} B_j^{(N)}$$

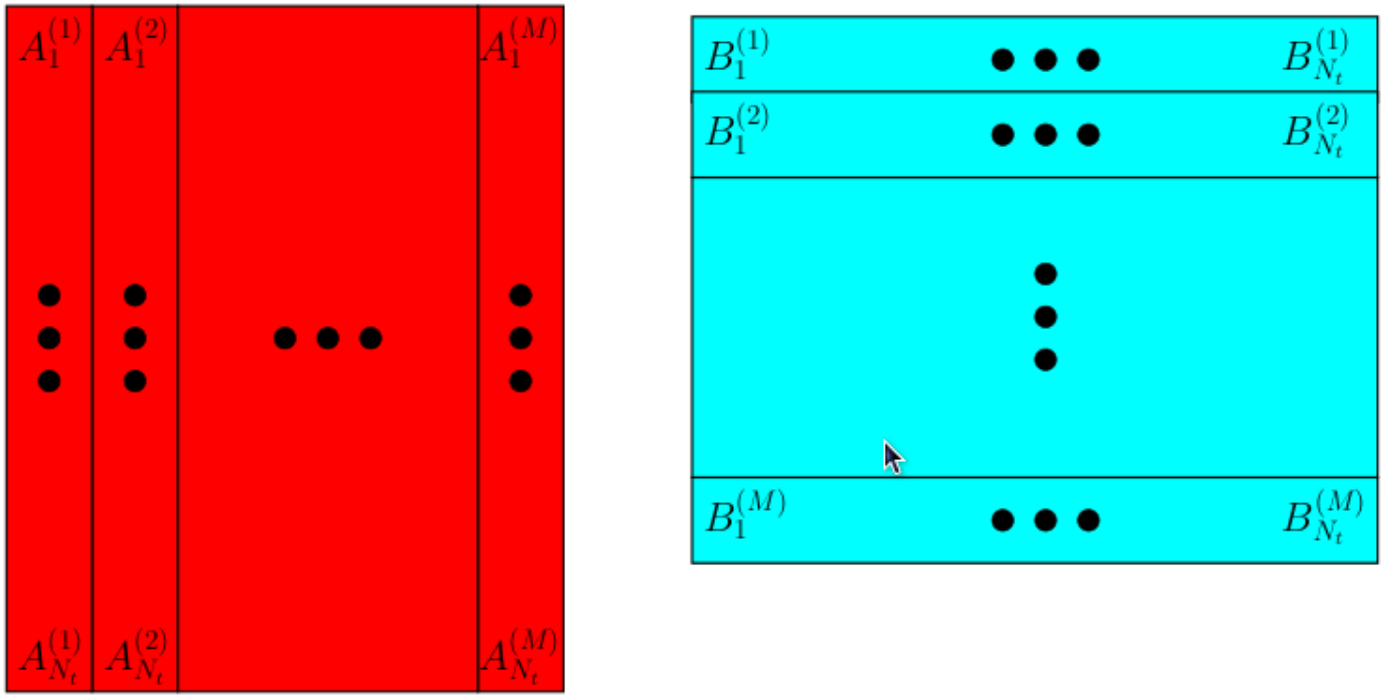
where the sum of outer products becomes a matrix multiplication. To Calculate the $A$'s and $B$'s.

$$B_j^{(N)} = row_j^{(N)} = row_j + \sum_{i=1}^{N-1} A^{(N)} B_j^{(N)} \qquad (1)$$

[1] Gonzalo Alverez *et. al.* Proc. of ACM/IEEE Conference on High Performance Computing, SC, 2008

# The delayed update

The sum $\sum_{N=1}^{M} A_i^{(N)} B_j^{(N)}$ becomes the panel multiplication




Requires $O(M * (N_c * L)^2)$ operations; increases the FLOPs per memory access.

CPU implementation: $M = 32$ provided optimal performance[1].

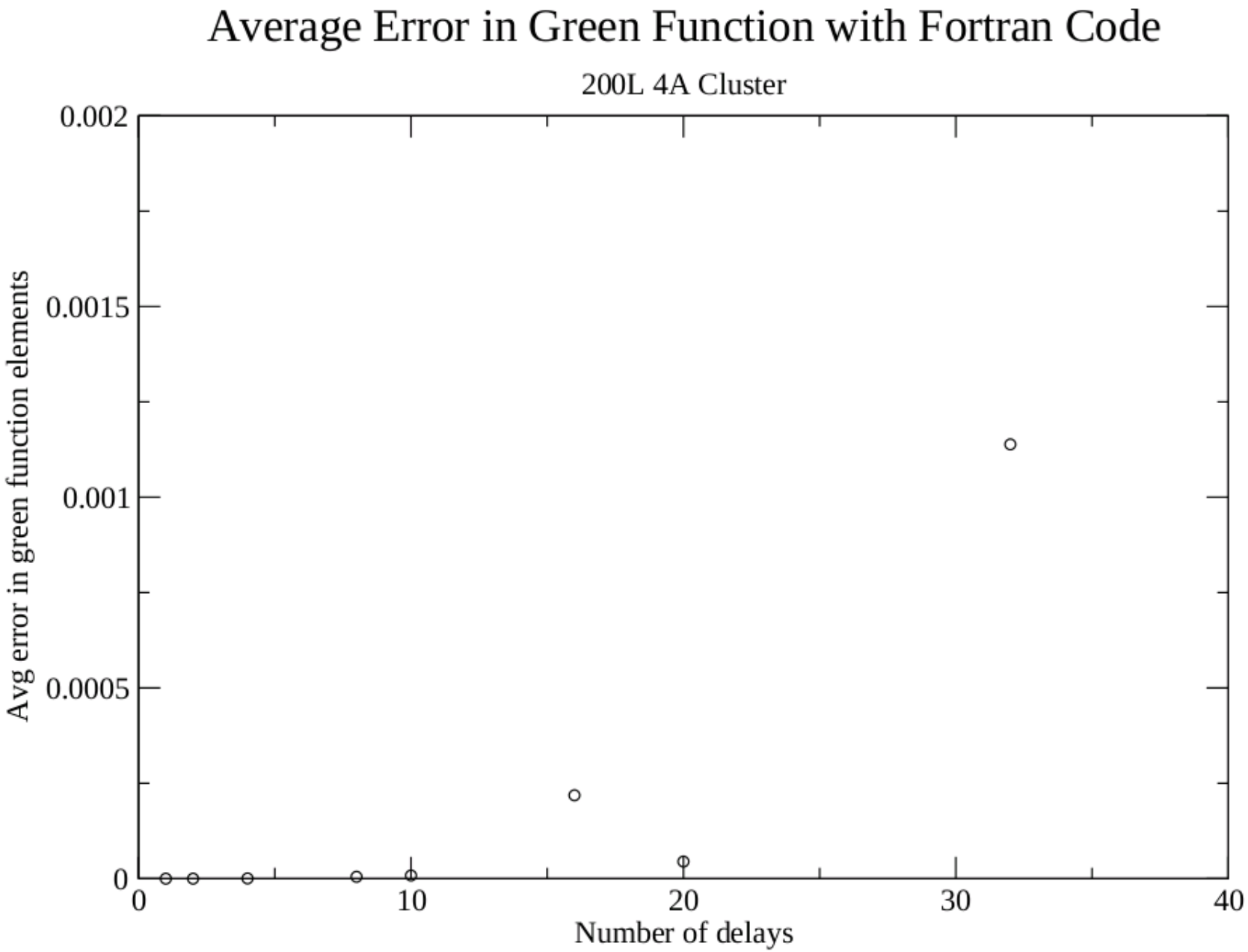
[1] M. Jarrell, A. Macridin, K. Tomko and E. D'Azevedo Optimizing Quantum Monte Carlo Algorithms for the Quantum Cluster Problem

# what is accomplished till now

- Starting from February we started with a fresh serial version.
- Developing a new serial version helped us understand the problem more. Which helps us to parallelize it more efficiently.
- MPI Fortran version was hard to understand and trace.
- Serial version now employs delayed updating , outputs match these from the Fortran code ( first time ever checked !! )

# What are we studying now ?

- How far can we go with delayed updating in single precision .
- What is the "Sweet spot" that gives you acceptable results and fast performance.
- Results next slide shows delays VS numerical error for serial version.


Average Error in Green Function with Fortran Code
200L 4A Cluster
Avg error in green function elements
0.002
0.0015
0.001
0.0005
0
0
10
20
30
40
Number of delays

# on the to-do list

- we need a better way to move the inputs of the green function. maybe a standard HDF5 to make sure we are marinating full possible precision . Maybe move the generation of input green  function to C.
- Use accelerator for delayed updating.
- Write a GPU kernel for the delayed updating.

# Lesson(s) learned.

- What appears to be working?
  Start coding with both CS/E and Physics folks present. The same thing worked for parallel tempering.

Thank you