OAK RIDGE National Laboratory

Mixed precision sampling of quantum states of matter

Thomas A. Maier

Oak Ridge National Laboratory

Smoky Mountains Conference, August 2019

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

SciDAC Scientific Discovery through Advanced Computing

OLCF Oak Ridge Leadership Computing Facility

U.S. DEPARTMENT OF ENERGY INCITE LEADERSHIP COMPUTING

U.S. DEPARTMENT OF ENERGY

This work was supported by the Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Sciences, Division of Materials Sciences and Engineering.

Collaborators & Acknowledgements

ETH Zürich

- Thomas C. Schulthess
- Peter Staar
- Giovanni Balduzzi
- Urs Hähner

Oak Ridge

- Jeremy Meredith (now at Google)
- Ed D'Azevedo
- Arghya Chatterjee
- Peter Doak
- Oscar Hernandez
- Wael Elwasif

Los Alamos

- Ying Wai Li

U.S. DEPARTMENT OF ENERGY | Office of Science

SciDAC
Scientific Discovery through Advanced Computing

OLCF
Oak Ridge Leadership Computing Facility

U.S. DEPARTMENT OF ENERGY
INCITE
LEADERSHIP COMPUTING

ORNL LDRD

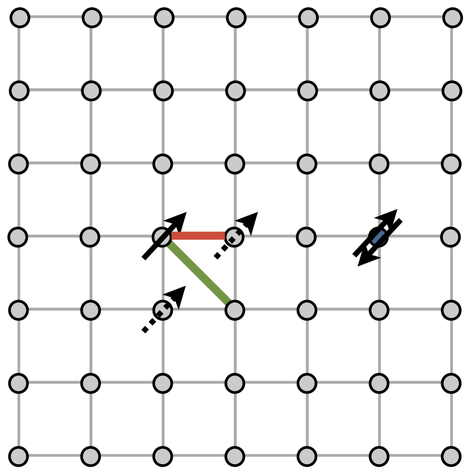
Correlated Quantum Materials

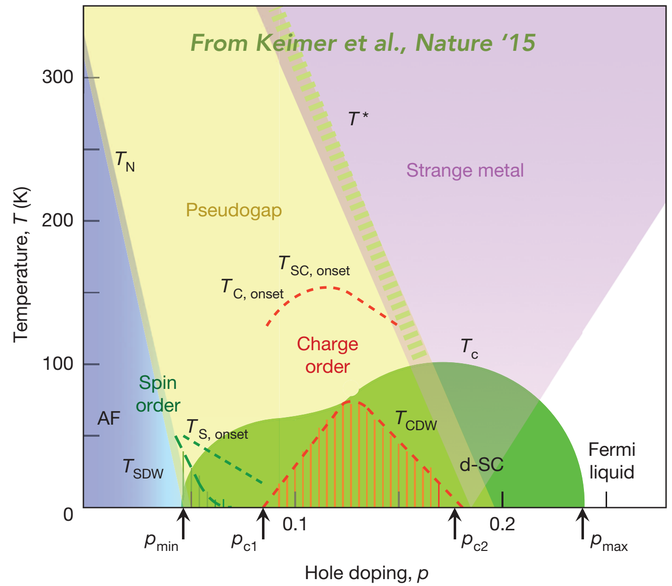
Strong electron-electron correlations

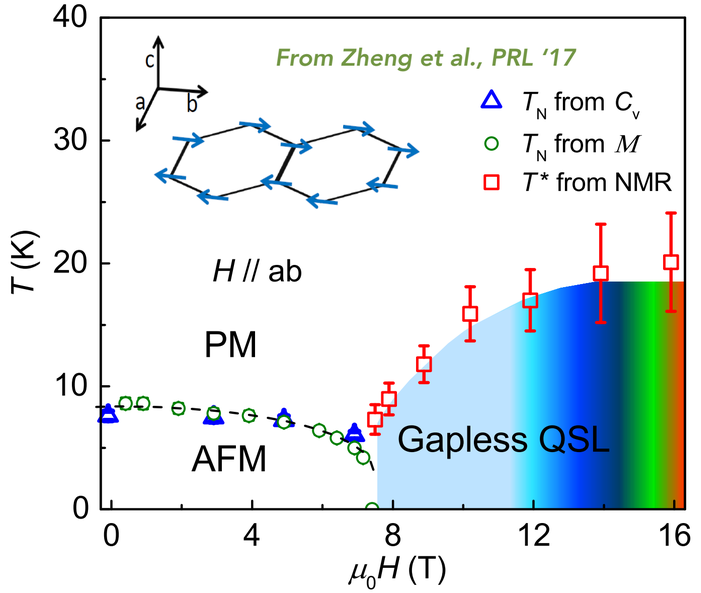
- Kinetic energy ≈ Coulomb repulsion
- **Electrons behave collectively and produce nearly degenerate emergent phases and excitations**
- Magnetism, charge order, nematic states, **superconductivity**, … in metals
- Spin liquid behavior, fractionalized excitations, Majorana fermions, … in geometrically frustrated spin systems

Model description

- Accuracy needed to describe effects of correlations and phase competition requires use of **reduced model Hamiltonians** and **advanced quantum many body methods, in conjunction with high-performance computing**



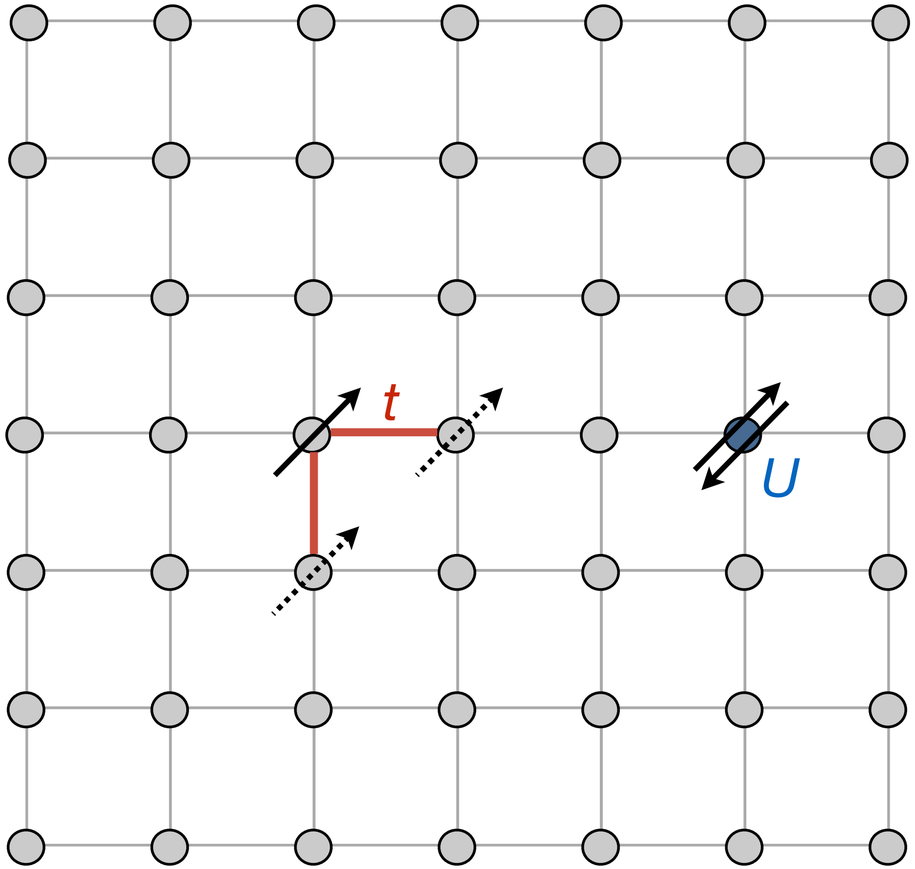




OAK RIDGE National Laboratory

Two-dimensional Hubbard Model

$$\mathscr{H} = -t \sum_{\langle ij \rangle, \sigma} c^{\dagger}_{i\sigma} c_{j\sigma} + U \sum_{i} n_{i\uparrow} n_{i\downarrow}$$



Phenomena

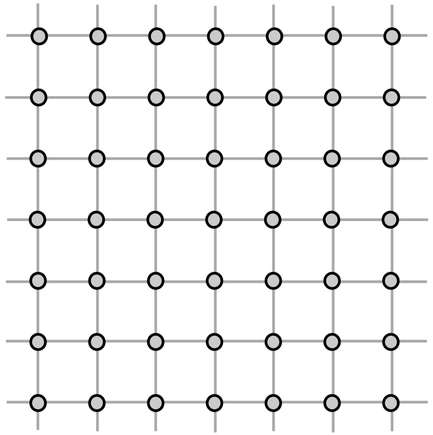
- Superconductivity
- Charge stripes
- Magnetic stripes
- Nematicity
- Strange metal behavior

Complexity

- Partition function: $Z = \mathrm{Tr} e^{-\beta \mathscr{H}}$
- Trace over 4^N states; N: number of sites in lattice
- Further approximations necessary

Dynamic Cluster Quantum Monte Carlo Approximation (DCA)

Infinite size lattice



DCA self-consistently maps infinite size lattice to **effective cluster embedded in dynamic mean-field** that describes the remaining lattice degrees of freedom

(1) Coarse-graining

$$\bar{G}(\mathbf{K}, i\omega_n) = \frac{N_c}{N} \sum_{\mathbf{k}\in\mathscr{P}_\mathbf{K}} G(\mathbf{k}, i\omega_n) = \frac{N_c}{N} \sum_{\mathbf{k}\in\mathscr{P}_\mathbf{K}} \frac{1}{i\omega_n - \varepsilon_\mathbf{k} + \mu - \Sigma_c(\mathbf{K}, i\omega_n)}$$

(2) Cluster exclusion

$$\mathscr{G}_0(\mathbf{K}, i\omega_n) = \left[\bar{G}^{-1}(\mathbf{K}, i\omega_n) + \Sigma_c(\mathbf{K}, i\omega_n)\right]^{-1}$$

$\mathscr{G}_0(\mathbf{K}, i\omega_n)$, U

(3) Quantum Monte Carlo cluster solver

$$S[\phi^*, \phi] = -\int_0^\beta d\tau \int_0^\beta d\tau' \sum_{ij,\sigma} \phi^*_{i\sigma}(\tau)\mathscr{G}_{0,ij,\sigma}(\tau - \tau')\phi_{j\sigma}(\tau) + \int_0^\beta d\tau \sum_i U\phi^*_{i\uparrow}(\tau)\phi_{i\uparrow}\phi^*_{i\downarrow}(\tau)\phi_{i\downarrow}(\tau)$$

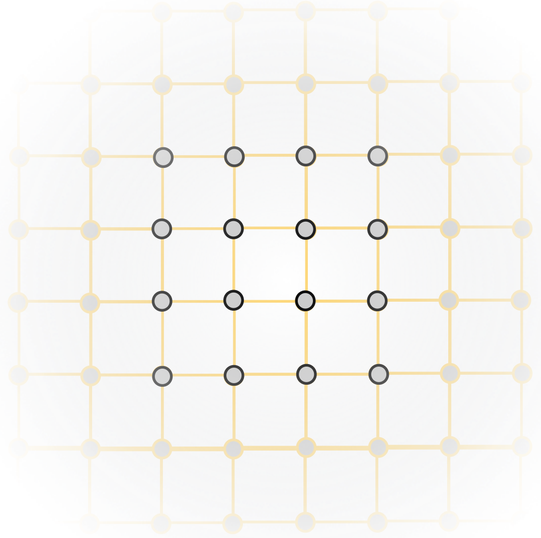
$$G_{c,ij,\sigma}(\tau - \tau') = \frac{1}{Z}\int \mathscr{D}[\phi^*\phi]\phi_{i\sigma}(\tau)\phi^*_{j\sigma}(\tau')e^{-S[\phi^*,\phi]};\ Z = \int \mathscr{D}[\phi^*\phi]e^{-S[\phi^*,\phi]}$$

(4) New self-energy

$$\Sigma_c(\mathbf{K}, i\omega_n) = \mathscr{G}_0^{-1}(\mathbf{K}, i\omega_n) - G_c^{-1}(\mathbf{K}, i\omega_n)$$

$G_c(\mathbf{K}, i\omega_n)$

Cluster embedded in mean-field



> 90% of computation

Numerically exact solution of effective cluster problem with ***quantum Monte Carlo***

Monte Carlo integration

Deterministic integration

$$\int_a^b f(x)dx \approx \sum_{i=1}^{N} f(a+i\Delta x)\Delta x + \mathcal{O}(\Delta x^2)$$

- In *d* dimensions, error is $\mathcal{O}(N^{-\frac{2}{d}})$
- With Simpson rule, error is $\mathcal{O}(N^{-\frac{4}{d}})$

Monte Carlo integration

- Use *N* randomly chosen points $\mathbf{x}_i$

$$\frac{1}{\Omega}\int f(\mathbf{x})d\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) + \mathcal{O}(\sqrt{\frac{\mathrm{Var}f}{N}})$$

- Better: ***Importance sampling***
- Chose $\mathbf{x}_i$ according to normalized probability distribution $p(\mathbf{x})$

$$\frac{1}{\Omega}\int f(\mathbf{x})d\mathbf{x} = \frac{1}{\Omega}\int \frac{f(\mathbf{x})}{p(\mathbf{x})}p(\mathbf{x})d\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} + \mathcal{O}(\sqrt{\frac{\mathrm{Var}f/p}{N}})$$

Markov chain

$$\mathbf{x}_0 \to \mathbf{x}_1 \to \mathbf{x}_2 \to \ldots \to \mathbf{x}_N$$

- Ergodicity, detailed balance
- Probability of accepting a move: min(R,1) (Metropolis); R/(1+R) (heat bath)

$$R \propto \frac{p(\mathbf{x}_{n+1})}{p(\mathbf{x}_n)}$$

- Estimation of observables

$$\langle O \rangle_p \approx \frac{1}{N}\sum_{i=1}^{N} O(\mathbf{x}_i) + \mathcal{O}(1/\sqrt{N})$$

Quantum Monte Carlo

Partition function

$$Z = \mathrm{Tr} e^{-\beta \mathscr{H}}$$

- Trace is over 4^N states in Hilbert space of $\mathscr{H}$

Observables

- $\langle O \rangle = \frac{1}{Z} \mathrm{Tr} \left[O e^{-\beta \mathscr{H}} \right]$
- Map *d*-dimensional quantum system onto *d*+1 -dimensional classical system

$$\langle O \rangle = \frac{1}{Z} \mathrm{Tr} \left[O e^{-\beta \mathscr{H}} \right] = \sum_i O(x_i) P(x_i)$$

- $O(x_i)$: value of observable in corresponding (artificial) classical system with weight $P(x_i)$

Fermion negative sign problem

- $P(x_i) < 0$ can arise from Pauli exclusion principle when two fermions are exchanged along Markov chain $\mathbf{x}_0 \to \mathbf{x}_1 \to \mathbf{x}_2 \to \ldots \to \mathbf{x}_N$
- Standard procedure

$$\langle O \rangle = \frac{\sum_i O(x_i) p(x_i)}{\sum_i p(x_i)} = \frac{\sum_i O(x_i) s(x_i) |p(x_i)| / \sum_i |p(x_i)|}{\sum_i s(x_i) |p(x_i)| / \sum_i |p(x_i)|} \equiv \frac{\langle O s \rangle'}{\langle s \rangle'}$$

with $s(x_i) = \mathrm{sign}\, p(x_i)$.

- Problem:

$$\frac{\Delta s}{\langle s \rangle} \sim \frac{e^{\beta N \Delta f}}{\sqrt{N}}$$

- ***Sign problem*** (statistical errors in observables) ***increases exponentially*** with inverse temperature, system size (and interaction parameters).

Monte Carlo methods rely on random sampling and have statistical errors. In principle, Monte Carlo can tolerate reduced precision, if error is within statistical noise.

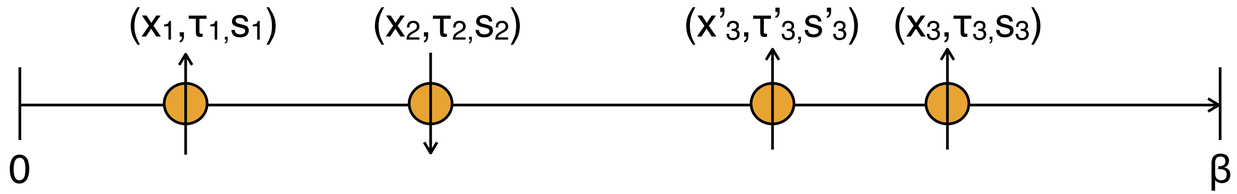
Continuous-time auxiliary field quantum Monte Carlo solver

Gull et al., EPL '08

Partition function

$$Z = \mathrm{Tr}\, e^{-\beta H} \propto \sum_{k=0}^{\infty} \sum_{s_1 \ldots s_k = \pm 1} \int_0^\beta d\tau_1 \ldots \int_{\tau_{k-1}}^\beta d\tau_k \left(\frac{K}{2\beta N_c} \right)^k \prod_\sigma \det N_\sigma^{-1}(\{x, \tau, s\}_k)\,; \;\; G = N G_0$$

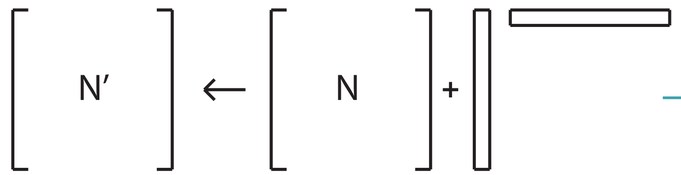
Monte Carlo sampling space, insertion and removal updates



Probability for updating configuration x to x′

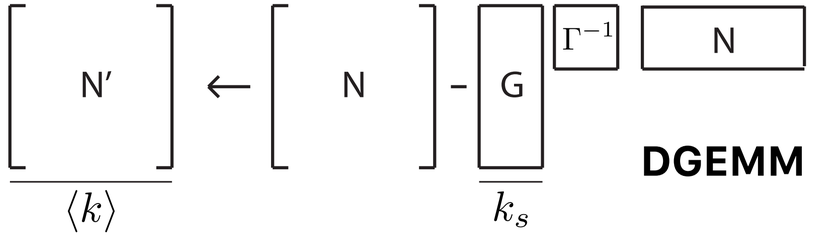
$$R_{x \to x'} = \min(1, R)\,; \;\; R = \frac{K}{k+1} \prod_\sigma \frac{\det \mathbf{N}_\sigma^{-1}(\{x', \tau', s'\})}{\det \mathbf{N}_\sigma^{-1}(\{x, \tau, s\})}$$

If move is accepted, update *N*-matrix, according to rank-1 (rank-k_s) update



Delayed updates

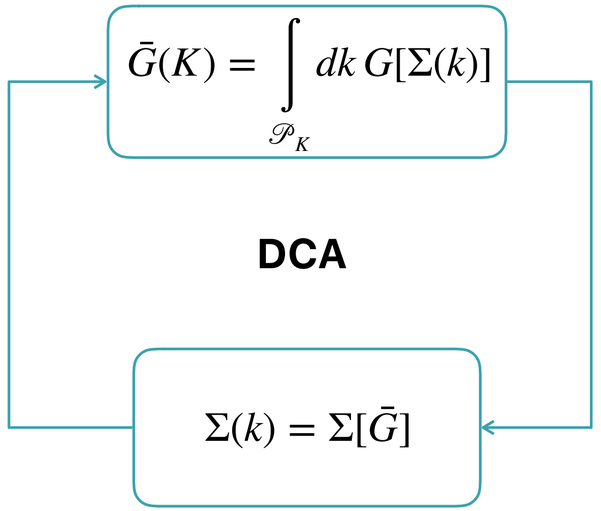
Alvarez et al., SC '08
Nukala et al., PRB '09
Gull et al., PRB '11



Monte Carlo accumulator: Measurements of observables

Single-particle Green's function *G*

- Needed for self-consistency in DCA
- Provides information on **single-particle excitations**
- $G(\mathbf{k},\omega) = \mathcal{G}_0(\mathbf{k},\omega) - \mathcal{G}_0(\mathbf{k},\omega)M(\mathbf{k},\omega)\mathcal{G}_0(\mathbf{k},\omega)$
- Use delayed non-equidistant FFT algorithm to calculate $M(\mathbf{k},\omega)$ from $M(\mathbf{k},\tau_i)$ where τ_i is random time grid



4-point two-particle Green's function G_4

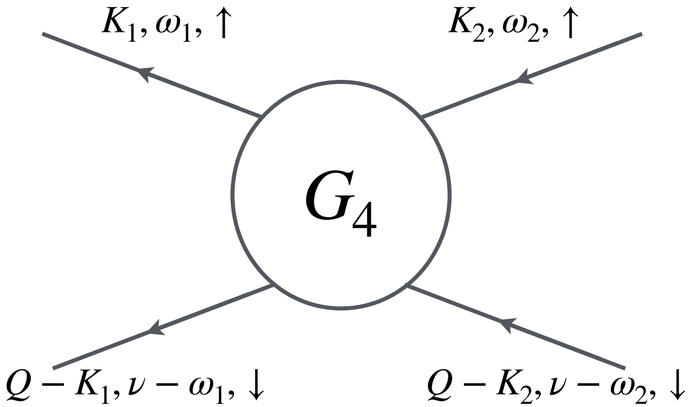
- High-dimensional tensor $G_4(Q,\nu,K_1,\omega_1,K_2,\omega_2)$
- Calculated only in last DCA iteration
- Provides information on **linear response of system to external fields**, and more (e.g. superconducting transition temperature T_c)

$$G_4(Q,K_1,K_2) += \sum_{\sigma} G_{\sigma}(Q-K_1,Q-K_2)G_{\bar{\sigma}}(K_1,K_2)$$

$$G(\omega_1,\omega_2) = G_0(\omega_1) - G_0(\omega_1)M(\omega_1,\omega_2)G_0(\omega_2)$$

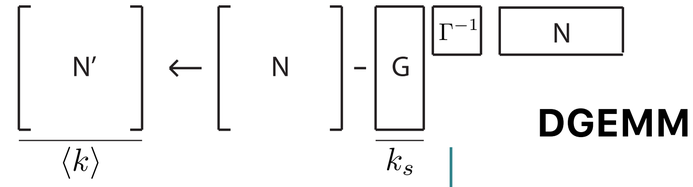
$$M(\omega_1,\omega_2) = \mathrm{FT}_{2D}[M(\tau_1,\tau_2)]$$

- Typical case: ~ 1000 FTs of ~ 70 x 70 matrices (***batched dgemm in MAGMA***)



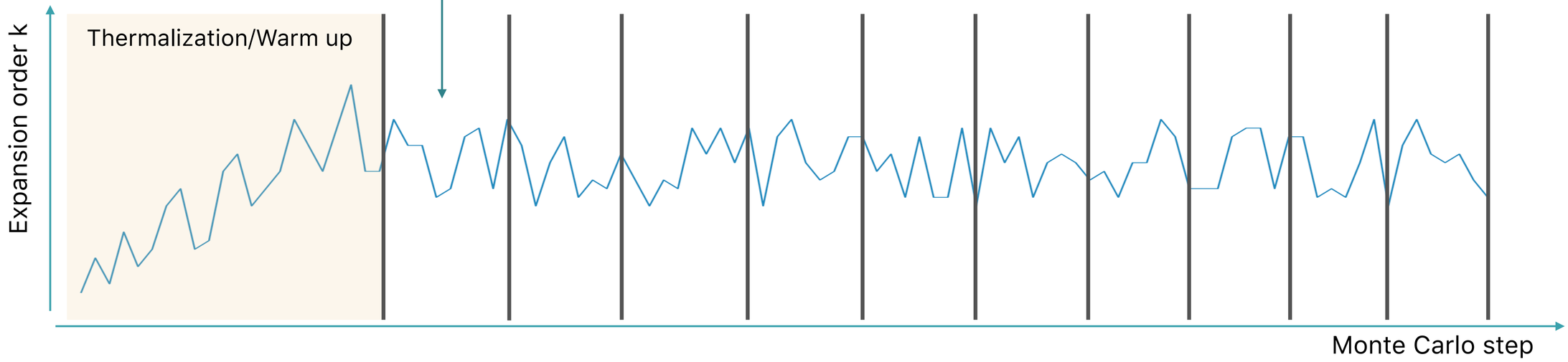
Markov Chain, updates and measurement of observables

Walker: Exploration of phase space



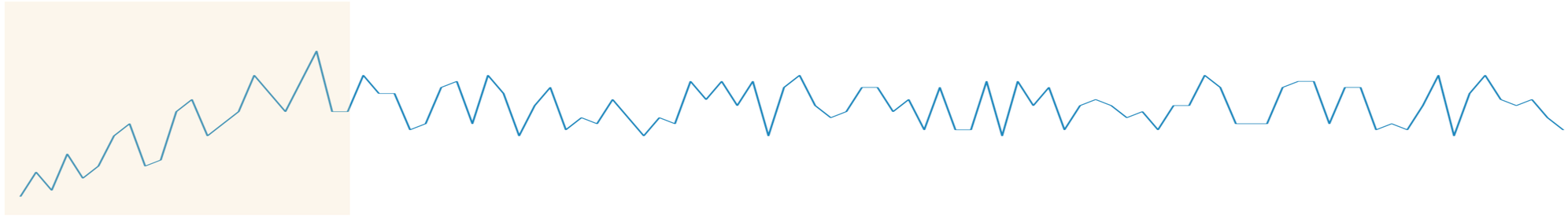
DGEMM

Accumulator: Periodically measure observables (*G* and G_4)

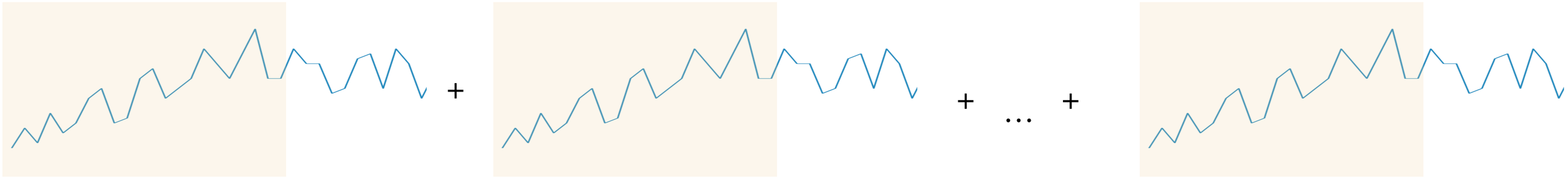


Overall scaling: $\mathcal{O}(k^3) \sim \mathcal{O}((N_c U/T)^3)$

Monte Carlo parallelism: Concurrent independent Markov chains



=

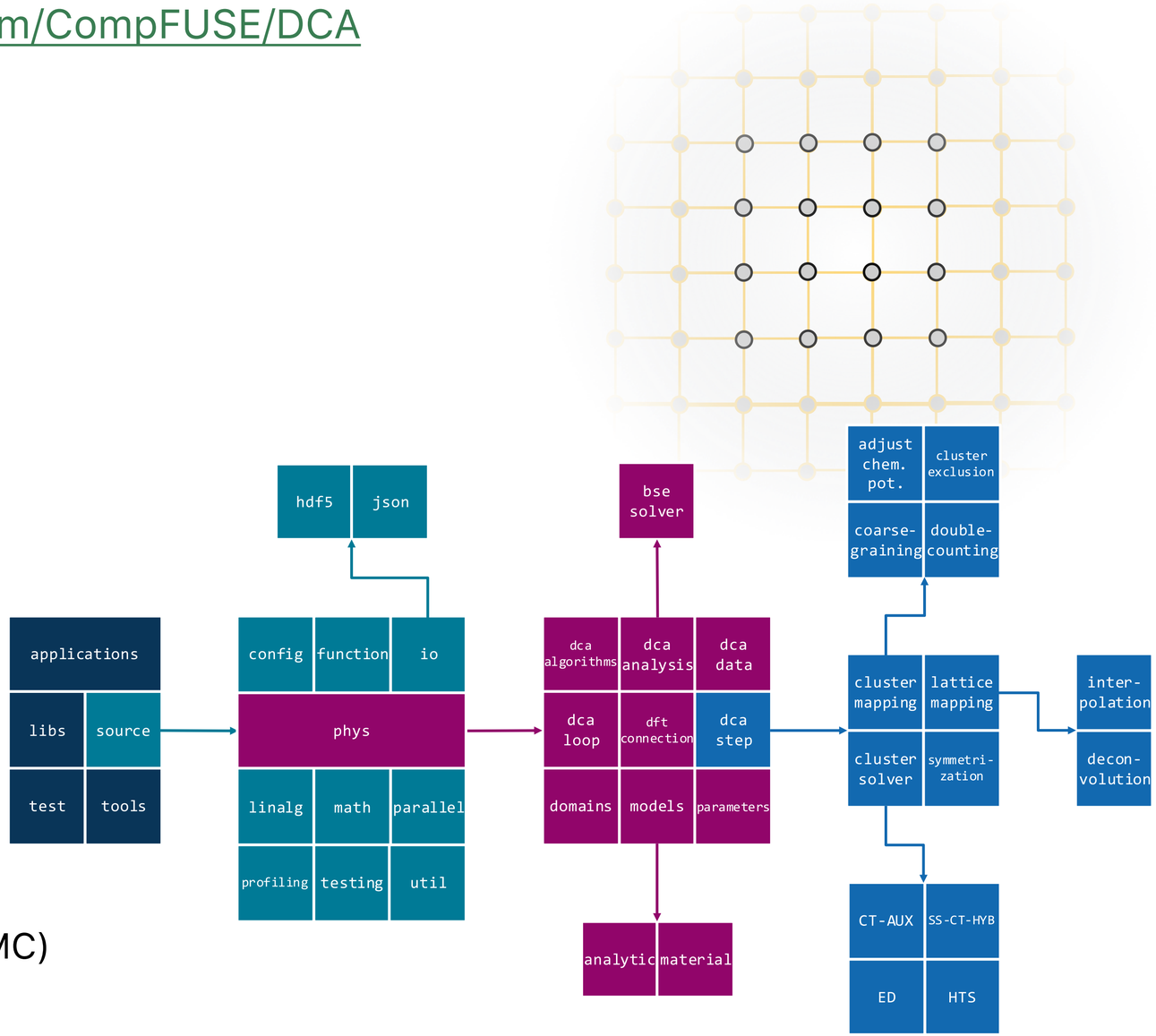


Split single Markov chain into many smaller chains.
Fixed thermalization steps do not scale.

DCA++ code

https://github.com/CompFUSE/DCA

- ***Implements dynamic cluster quantum Monte Carlo approximation with QMC solvers***
- ***Co-developed at ORNL and ETH Zürich***
- ***2008 Gordon Bell winner (peak performance), finalist in 2013***
- Recently re-written in C++14, ***heavily templated***
- MPI, C++ std:threads, CUDA
- BLAS, LAPACK, MAGMA for dense linear algebra operations
- FFTW3 for fast Fourier transforms on CPU, MAGMA on GPU
- Different QMC algorithms implemented (Hirsch-Fye QMC —> continuous-time auxiliary field, interaction representation, hybridization expansion QMC)
- SPEC MPI ACCEL benchmark suite code (2020)
- **~ 75 PFlops on Summit (FP64)**



Hähner et al., CPC '18

Reduced precision test

Matrix multiply test

- 1024 x 1024 and 2048 x 2048 random matrices
- Baseline: CPU long doubles
- Ten runs of same matrices on CPU and GPU (NVIDIA G80) computed in single-precision
- Mean relative error due to single-precision ~ 10^{-7} similar on CPU and GPU

Performance (GFlops)

	Single precision	Double precision
NVIDIA G80	120	n/a
NVIDIA GT200	220	45

Meredith et al., Parallel Computing '09

Parallel Computing 35 (2009) 151–163

Contents lists available at ScienceDirect

Parallel Computing

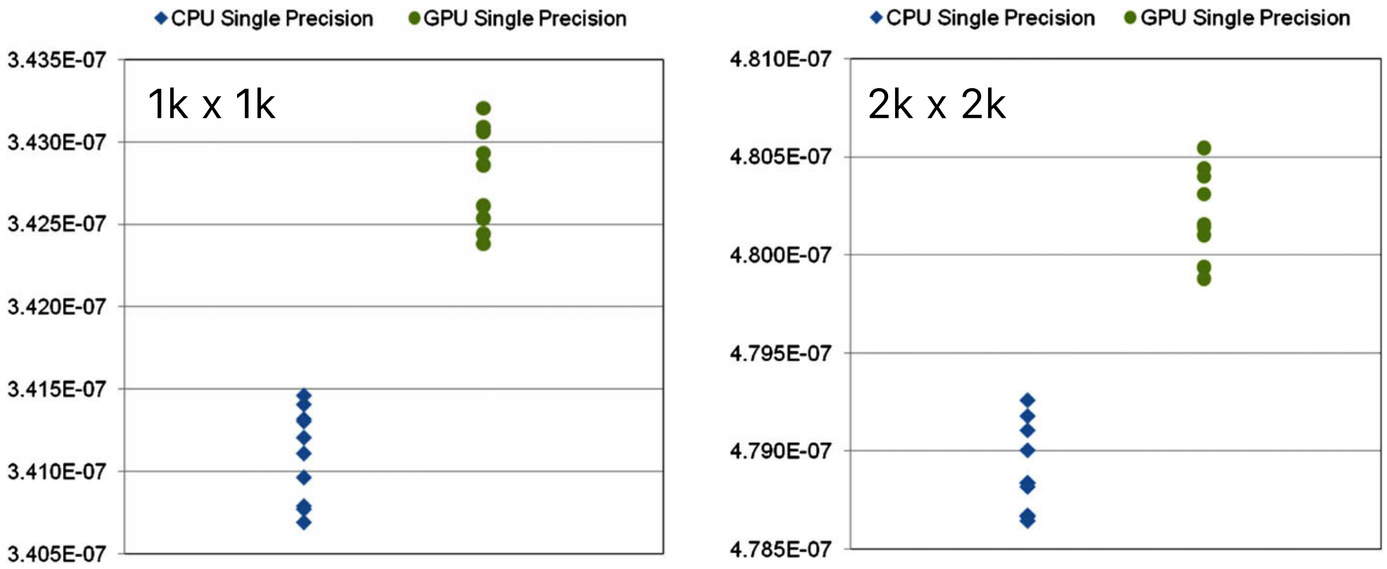
journal homepage: www.elsevier.com/locate/parco

Accuracy and performance of graphics processors: A Quantum Monte Carlo application case study

Jeremy S. Meredith *, Gonzalo Alvarez, Thomas A. Maier, Thomas C. Schulthess, Jeffrey S. Vetter

Oak Ridge National Laboratory, 1 Bethel Valley Road, MS 6173 Oak Ridge, TN 37831, USA

Single precision mean error (relative to long doubles)



Mixed precision DCA on CPU

DCA with mixed single/double precision

- Coarse-graining (mapping of lattice to cluster) in double precision
- QMC in single-precision (walkers and accumulators)

Leading eigenvalue of Bethe-Salpeter equation

$$\mathbf{G}_4^Q = \mathbf{GG} + \mathbf{GG}\,\boldsymbol{\Gamma}_{pp}^Q\,\mathbf{G}_4^Q$$

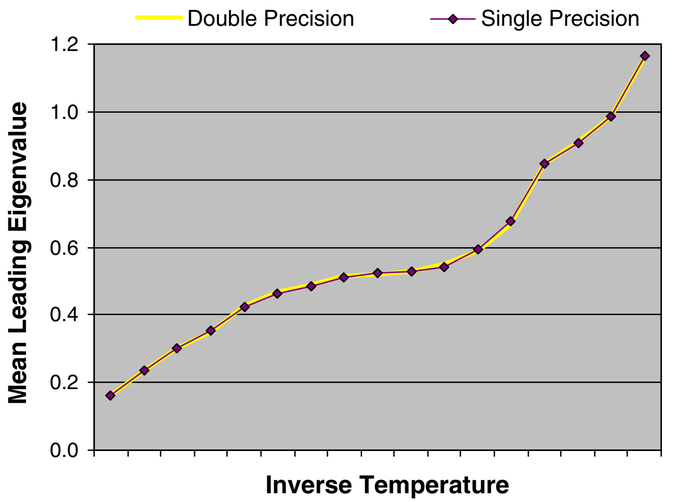
$$\boldsymbol{\Gamma}_{pp}^Q\,\mathbf{GG}\,\phi_\alpha = \lambda_\alpha \phi_\alpha$$

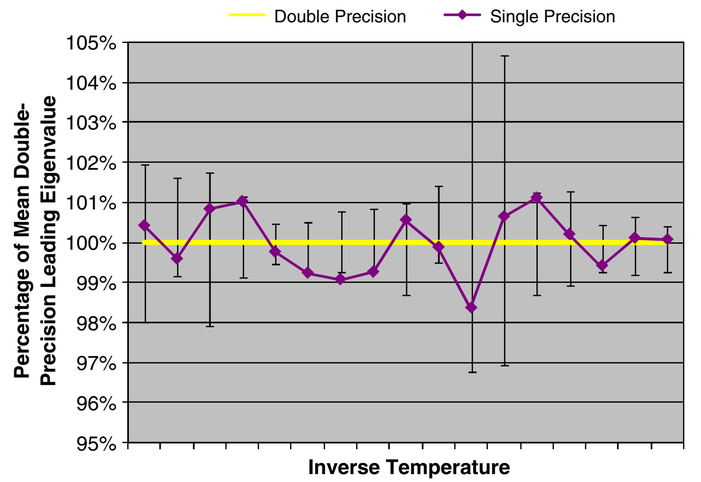
- Superconducting transition when leading eigenvalue $\lambda_\alpha = 1$

Single vs. double precision

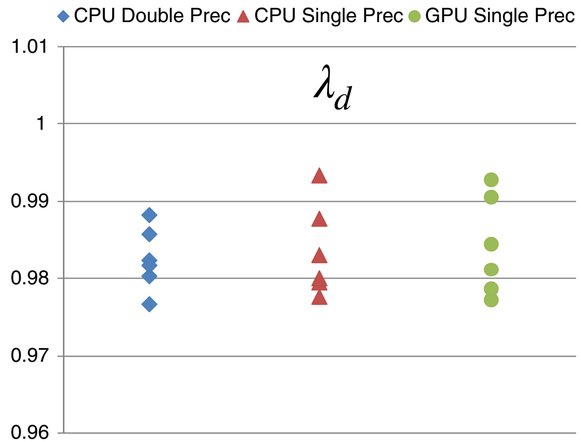
- Single precision results have less than 1% deviation from double precision results
- Variation within double precision runs greater than discrepancy between single and double precision runs

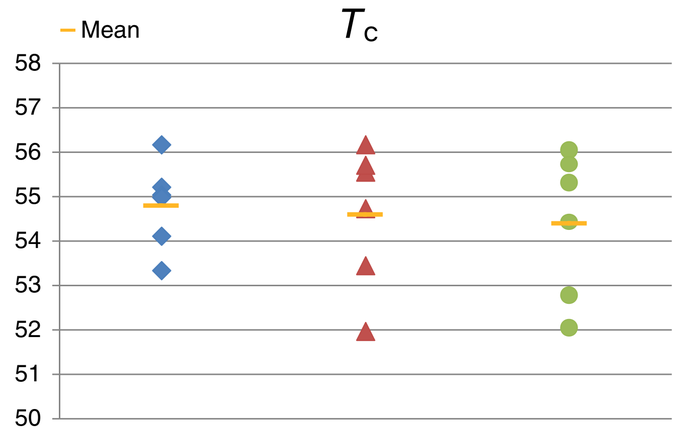
CPU-based single precision





GPU vs. CPU





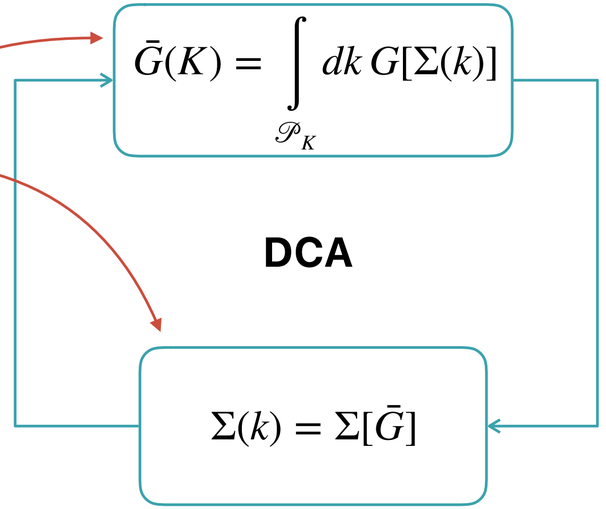
Single-precision has sufficient accuracy

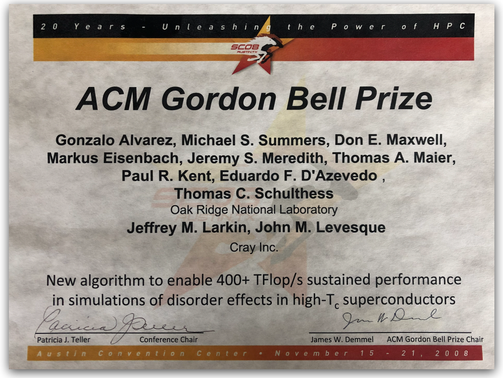
Meredith et al., Parallel Computing '09

Mixed precision DCA on ORNL's Jaguar

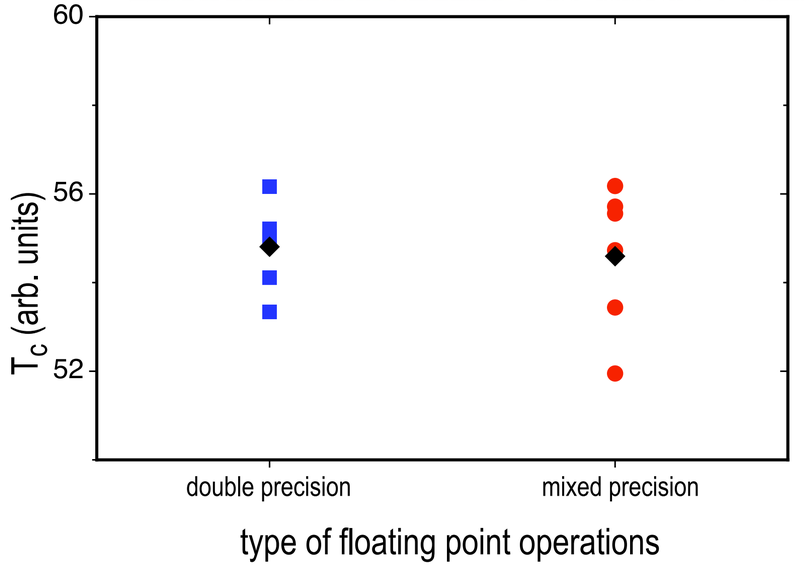
Simulation of disorder effects in high-T_c superconductors

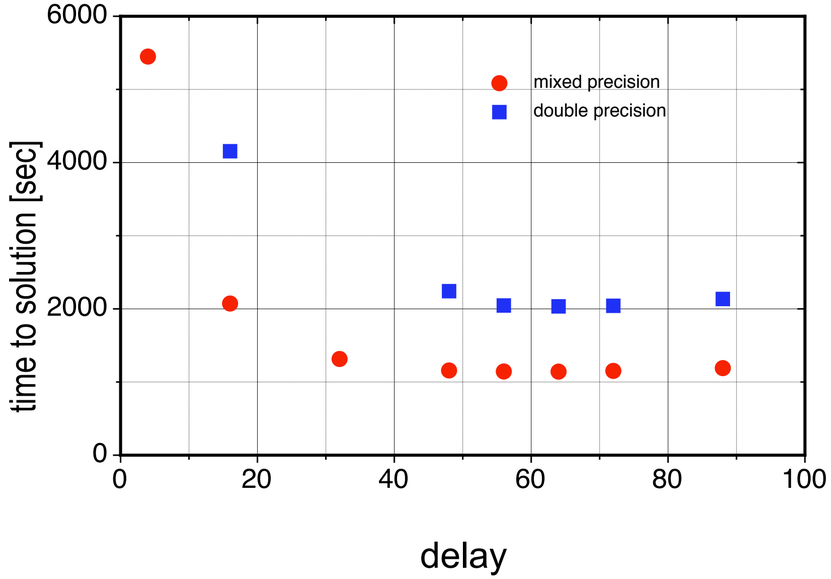
- Double-precision in cluster mapping
- Single-precision in QMC solver
- No loss of precision in single-precision runs
- Speedup of 1.5x - 2x
- BLAS SGEMM factor 2.02 faster than DGEMM
- ~ 400 TFlop/s on Cray XT4, later ~1 PFlop/s on Cray XT5





Alvarez et al., SC '08

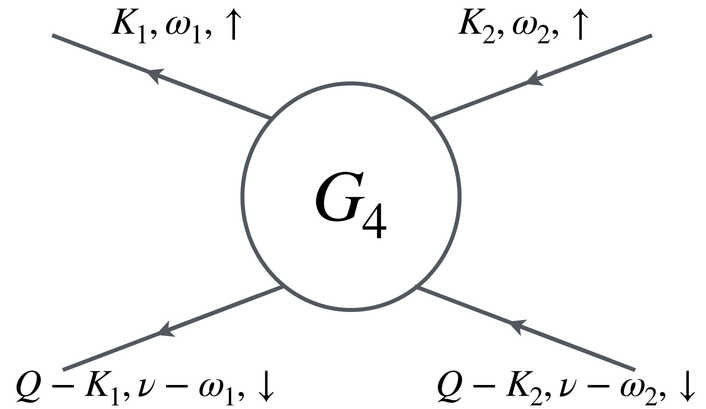




Memory improvements due to mixed precision

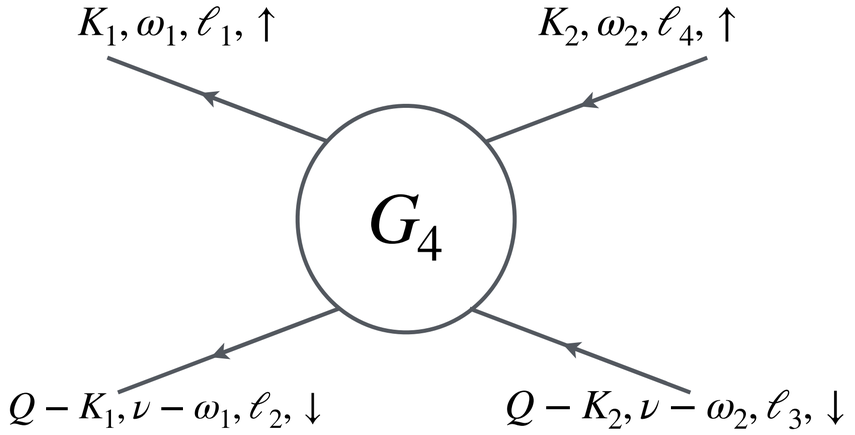
Most memory intensive part in DCA++

- **4-point Green's function** $G_4(Q, \nu, K_1, \omega_1, K_2, \omega_2)$
- Typical production runs (single-band Hubbard):
  # Q, $K1$, $K2$: 32
  # ν, ω_1, ω_2 : 128
  Total: $32^3 \times 128^3 \approx 7 \times 10^{10}$ complex numbers
- Storage requirement FP64: ~ 1 TB



More complex problems

- Multi-orbital models
- **4-point Green's function** $G_{4,\ell_1\ell_2\ell_3\ell_4}(Q, \nu, K_1, \omega_1, K_2, \omega_2)$
- E.g. 3-orbital model: another factor of 3^4 —> ~ 100 TB (FP64)



DCA++ and Tensor Cores: Half-precision arithmetics?

Tensor Cores

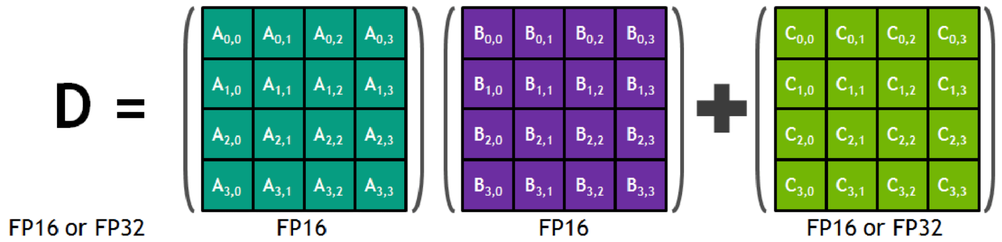
- V100 GPUs have hardware acceleration FP16 arithmetics
- 120 TFlop/s

Matrix multiply on Nvidia V100

FP64 → FP32	~ 2x faster
FP32 → FP16	~ 2x faster
FP32 → FP16 (TC)	~ 8x faster

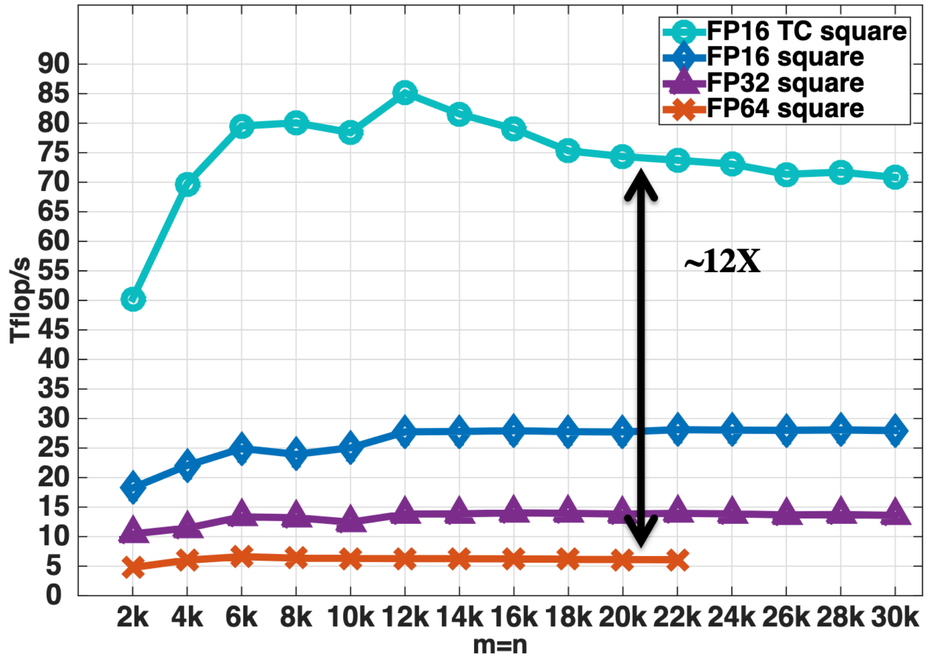
Brute force?

- Error due to half-precision likely too large in most cases



Matrix matrix multiply on Nvidia V100

(Stan Tomov, Presentation at OLCF User Meeting, Oak Ridge, 2019)



Half-precision arithmetics for DCA++ ?

General idea for FP16 arithmetics with FP32 accuracy

(Ed D'Azevedo)

- Approximate FP32 vector by scaled sum of 2 FP16 vectors
- $A = a_1 A_1 + a_2 A_2$; $B = b_1 B_1 + b_2 B_2$
- Combined accuracy is 22 bits in mantissa
- Use tensor cores to perform $C = A \times B$
- Can be evaluated approximately (~ 10^{-7} error) by
  3 matrix-matrix multiplies of FP16 matrices on tensor cores

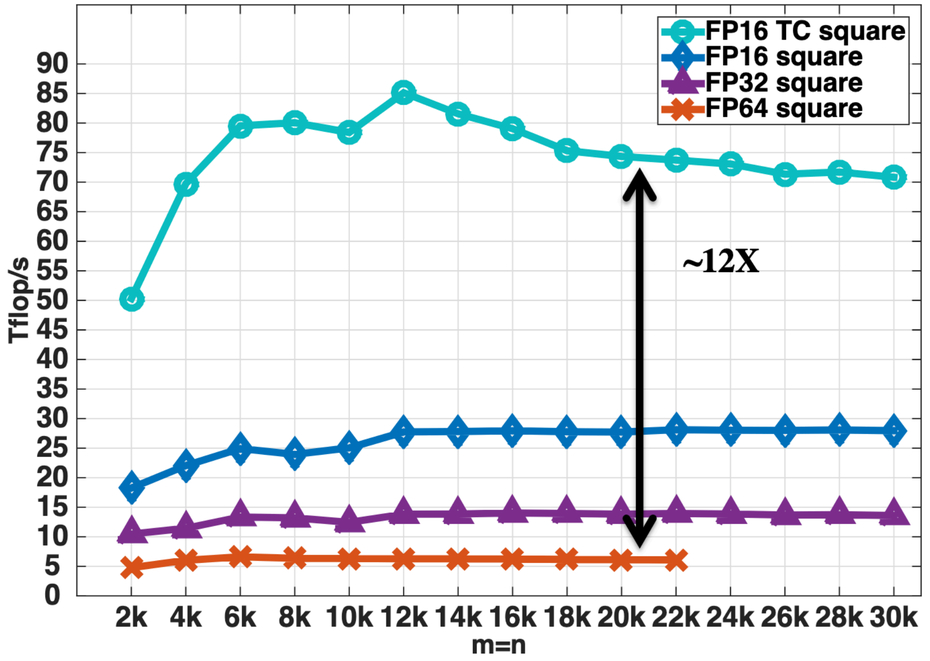
Potential performance gain

- Peak performance for tensor cores ~ 8x faster than FP32
  → 8/3 ~ 2.7x performance gain over FP32
- FP32 ~ 2x faster than FP64
- DGEMM ~ 80% of runtime in DCA++

→ ***Potential for overall performance gain over FP64: 2.9x***

Matrix matrix multiply on Nvidia V100

(Stan Tomov, Presentation at OLCF User Meeting, Oak Ridge, 2019)



Questions to address

Am I guaranteed the stability, accuracy and convergence properties using lower precision?

- (Quantum) Monte Carlo methods can afford reduced precision in most cases due to their statistical nature

What memory and performance improvements can I expect when using lower precision?

- Factor 1.5x - 2x with mixed single/double precision, and potentially more with Tensor Cores
- Significant memory reductions

What implementation challenges exist for application and enabling technologies developers?

- Minimal due to heavily templated code
- Challenge mostly in testing whether reduced precision error is within statistical Monte Carlo error