Continuous-time quantum Monte Carlo algorithms for impurity problems

Michel Ferrero

Centre de Physique Théorique

Ecole Polytechnique, France

triqs

Quantum Monte Carlo methods at work for novel phases of matter

Trieste, February 2-3, 2012

Outline

- Quantum impurity problems
- Continuous-time quantum Monte Carlo (CT-QMC) methods
- Generic introduction to Monte Carlo
- The interaction-expansion algorithm (CT-INT)
- The hybridization-expansion algorithm (CT-HYB)
- Some applications of CT-QMC solvers
- TRIQS: a Toolbox for Research on Quantum Interacting Systems
- Implementation details for the CT-INT

References

- Quantum impurity problems: *A. C. Hewson, "The Kondo Problem to Heavy Fermions", Cambridge University Press*

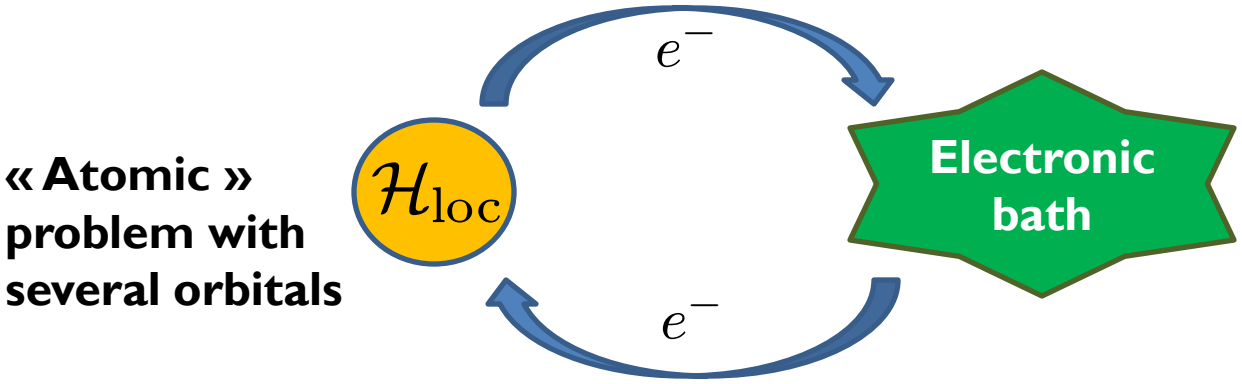
- CT-QMC solvers: *E. Gull et al., RMP (2011)*
  - The interaction-expansion algorithm
    Rubtsov et al., PRB (2005) and Rubtsov and Lichtenstein, JETP Lett. (2004)
  - The hybridization-expansion algorithm
    Werner and Millis, PRB (2006) and Werner et al., PRL (2006)

- Dynamical mean-field theory: *A. Georges et al., RMP (1996)*

- Open source codes with CT-QMC solvers:
  - TRIQS: *http://ipht.cea.fr/triqs*
  - ALPS: *http://alps.comp-phys.org*

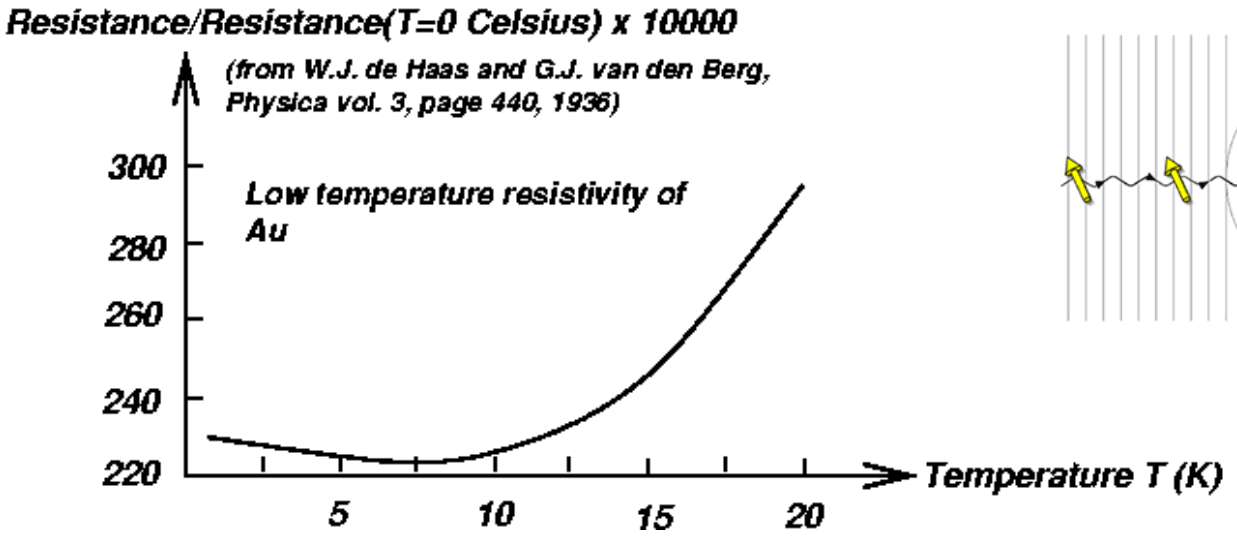
Quantum impurity problems

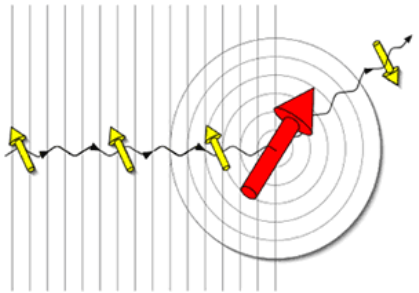
- They generically describe the behavior of a magnetic impurity embedded in an electronic host
- The impurity is a set of "orbitals" carrying local many-body interactions and localized spins. It can exchange electrons with an uncorrelated fermionic bath



The Kondo problem

- Experiments have shown that magnetic impurities in a metallic host (e.g. iron in gold) induce a minimum in the resistivity versus temperature curve

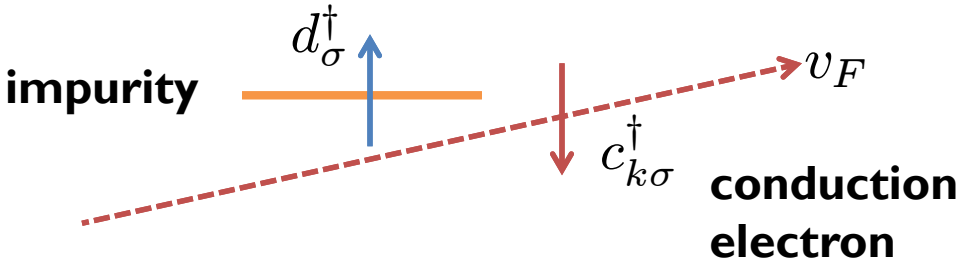




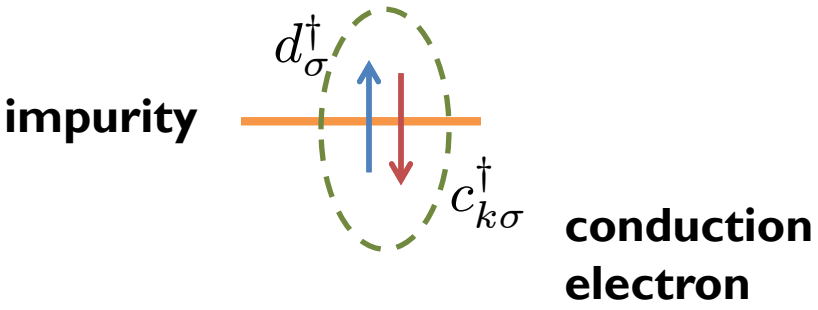
- The reason is that at low temperatures the conduction electrons "screen" the magnetic impurities: this is the Kondo effect

The Kondo effect: general idea

- At high temperature, conduction electrons pass by the impurity magnetic moment and only experience a small antiferromagnetic correlation



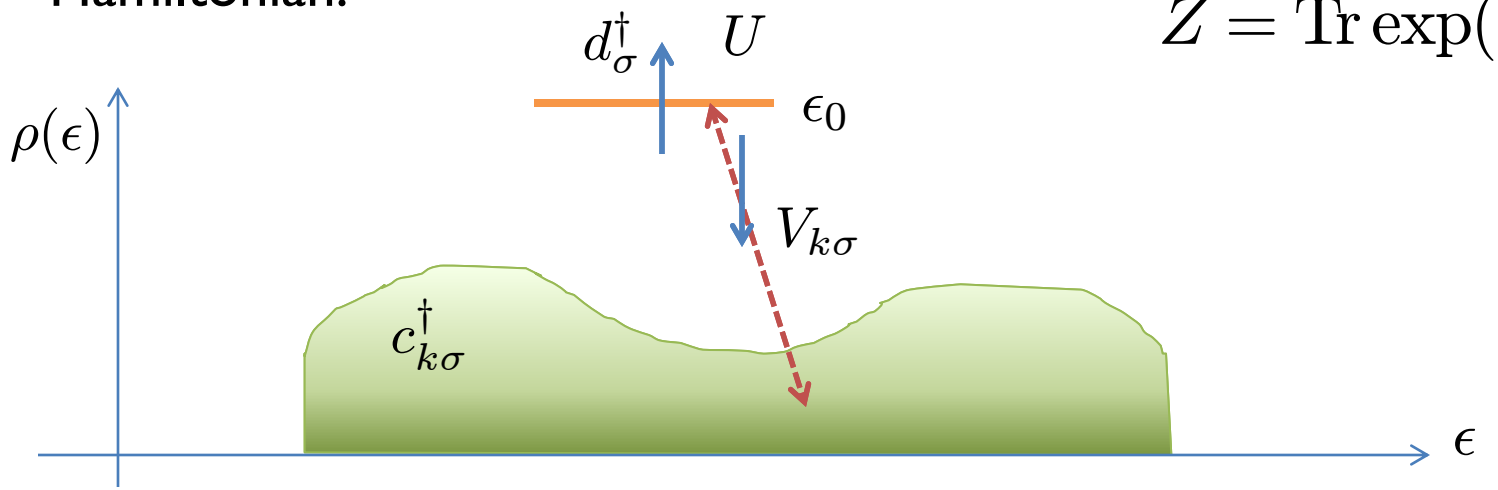
- At low temperature, the impurity magnetic moment and conduction electrons bind to form an overall non-magnetic state



The Anderson model

- A very successful model to understand magnetic impurities in a metallic host is the Anderson model
- Hamiltonian:

$$Z = \mathrm{Tr}\exp(-\beta\mathcal{H})$$



$$\mathcal{H} = \underbrace{\sum_{\sigma=\uparrow,\downarrow} \epsilon_0 d_\sigma^\dagger d_\sigma + U n_{d\uparrow} n_{d\downarrow}}_{\text{local many-body interaction}} + \underbrace{\sum_{k,\sigma} V_{k\sigma} c_{k\sigma}^\dagger d_\sigma + \text{h.c.}}_{\text{Hybridization to the bath}} + \underbrace{\sum_{k,\sigma} \epsilon_{k\sigma} c_{k\sigma}^\dagger c_{k\sigma}}_{\text{free bath states}}$$

Action for the Anderson model

- After integrating out the fermionic bath: $Z = \int \mathcal{D}[d^\dagger, d] e^{-S}$

$$S = -\sum_\sigma \int_0^\beta d\tau d\tau' d_\sigma^\dagger(\tau) G_{0\sigma}^{-1}(\tau - \tau') d_\sigma(\tau') + \int_0^\beta d\tau U n_{d\uparrow}(\tau) n_{d\downarrow}(\tau)$$

$G_{0\sigma}^{-1}(i\omega_n) = i\omega_n - \epsilon_0 - \Delta_\sigma(i\omega_n)$ non-interacting Green's function

$\Delta_\sigma(i\omega_n) = \sum_k \frac{|V_{k\sigma}|^2}{i\omega_n - \epsilon_{k\sigma}}$ hybridization function: describes the transition between the bath and the orbital

Remember: $G_{0\sigma}$ contains the information about the structure of the bath!

- Generic case:

$$S = -\sum_{a,b} \int_0^\beta d\tau d\tau' d_a^\dagger(\tau) G_{0,ab}^{-1}(\tau - \tau') d_b(\tau') + \int_0^\beta d\tau \mathcal{H}_{\text{int}}(\{d_a^\dagger, d_a\})(\tau)$$

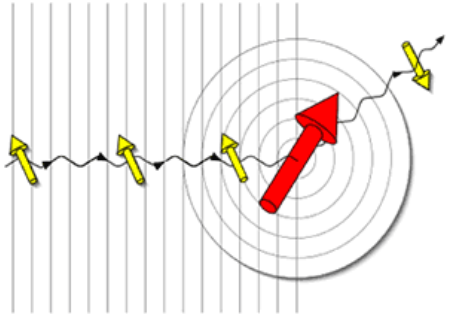
$$G_{0,ab}^{-1}(i\omega_n) = i\omega_n \delta_{ab} + t_{ab} - \Delta_{ab}(i\omega_n)$$

A difficult problem but a rich toolbox of methods

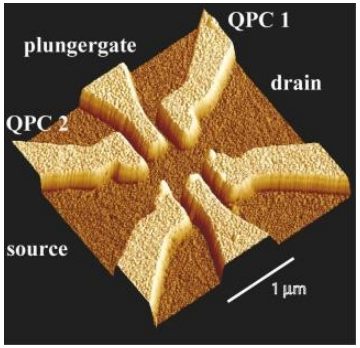
- The Anderson model is a many-body (correlated) problem with an infinite number of degrees of freedom! It has attracted a lot of interest and many techniques have been developed:
- Analytical methods (Bethe Ansatz, BCFT)
- Controllable algorithms:
  - Exact diagonalization
  - Numerical renormalization group
  - Density matrix renormalization group
  - Continuous-time quantum Monte Carlo algorithms
- Approximate solvers (e.g. NCA)
- All have pros and cons!

Impurity model in many contexts

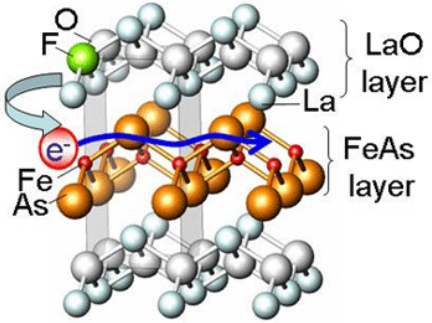
- Quantum impurity problems appear in many different contexts

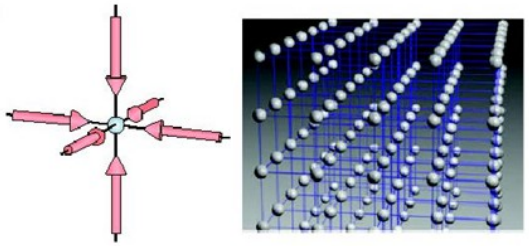


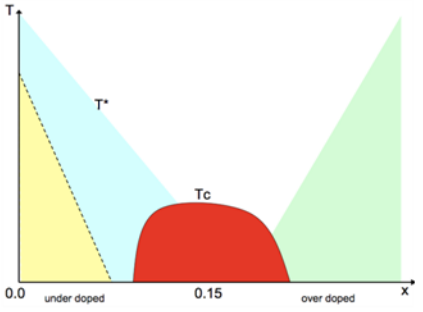
magnetic impurity in metallic host



nanostructures, quantum dots



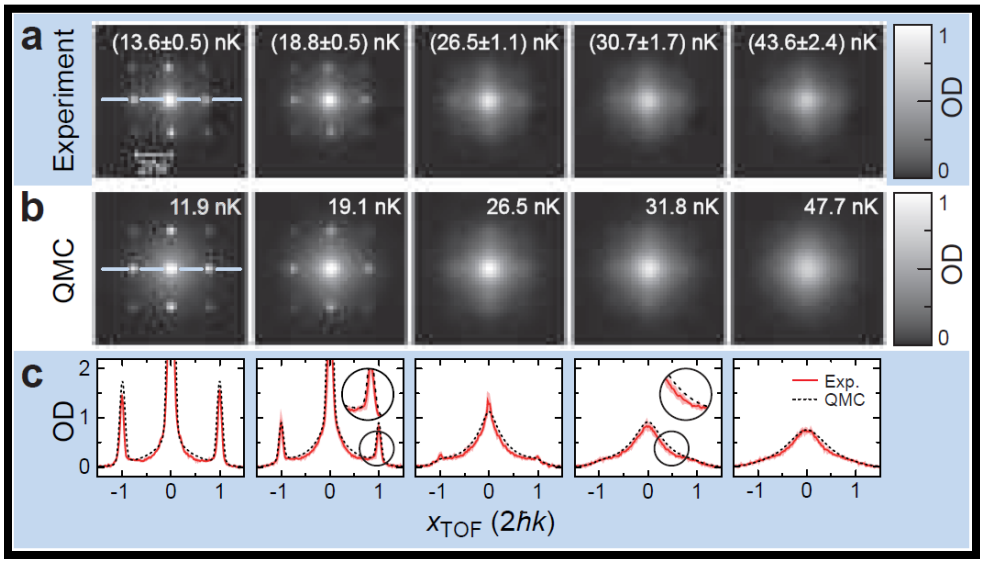




within the DMFT: pnictides, cuprates, cold-atomic systems, …

A renewed interest within DMFT

- There are very efficient Monte Carlo algorithms for bosons



- For fermions, the situation is different and approximation methods are needed $\Rightarrow$ dynamical mean-field theory (DMFT)
- DMFT has brought a renewed interest in quantum impurity models in the late 90's but also emphasized the limitations of existing algorithms and asked for the development of new tools

Dynamical mean-field theory

- The dynamical mean-field theory makes an approximation of a lattice model using an auxiliary quantum impurity problem
- It has been used a lot to find an approximate solution of the Hubbard model:

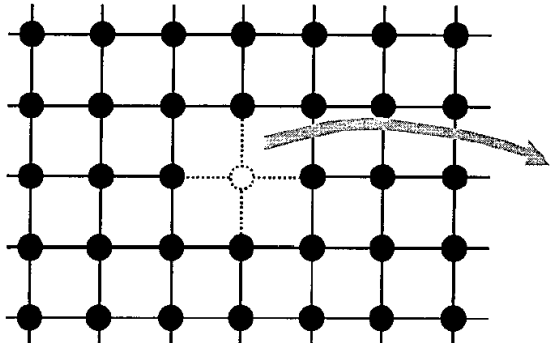
$$\mathcal{H} = \underbrace{\sum_k \epsilon_k c_k^\dagger c_k}_{\text{free electrons on a lattice}} + \underbrace{U \sum_i n_{i\uparrow} n_{i\downarrow}}_{\text{many-body interaction on every lattice site}}$$

- The DMFT has been used in many different contexts: realistic electronic structure calculations, models for superconductivity, cold-atomic systems, etc.

Dynamical mean-field theory cont'd

A. Georges and G. Kotliar, PRB (1992)
A. Georges et al., RMP (1996)

Lattice Hubbard model



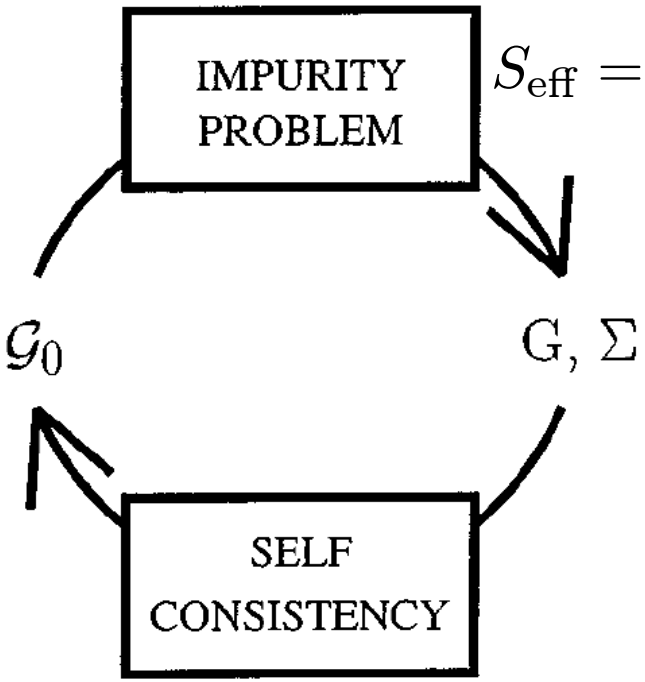
Anderson impurity model

Lattice Hubbard model is mapped onto a single-site model with an effective action:

$$S = -\sum_{\sigma} \int_0^{\beta} d\tau d\tau' d_{\sigma}^{\dagger}(\tau) G_{0\sigma}^{-1}(\tau - \tau') d_{\sigma}(\tau') + \int_0^{\beta} d\tau U n_{d\uparrow}(\tau) n_{d\downarrow}(\tau)$$

The bath has to be set self-consistently

DMFT equations and limitations



$$S_{\text{eff}} = -\int_0^\beta c_\sigma^\dagger(\tau) G_0^{-1}(\tau - \tau') c_\sigma(\tau') + \int_0^\beta U n_\uparrow(\tau) n_\downarrow(\tau)$$

$$\Sigma(i\omega_n) = G_0^{-1}(i\omega_n) - G^{-1}(i\omega_n)$$

$$\Sigma_{\text{lattice}}(i\omega_n, \cancel{k}) \leftarrow \Sigma(i\omega_n)$$

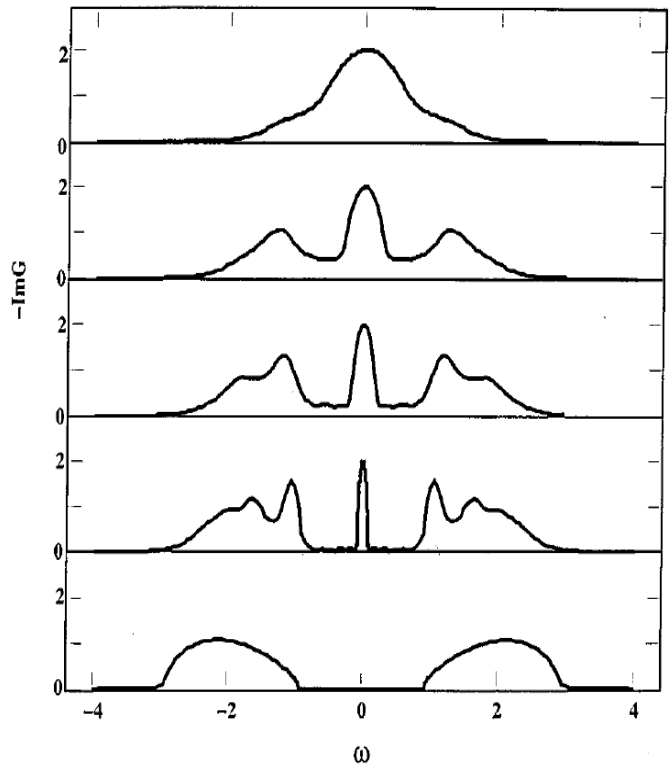
$$G(i\omega_n) = \sum_k G_{\text{lattice}}(i\omega_n, k)$$

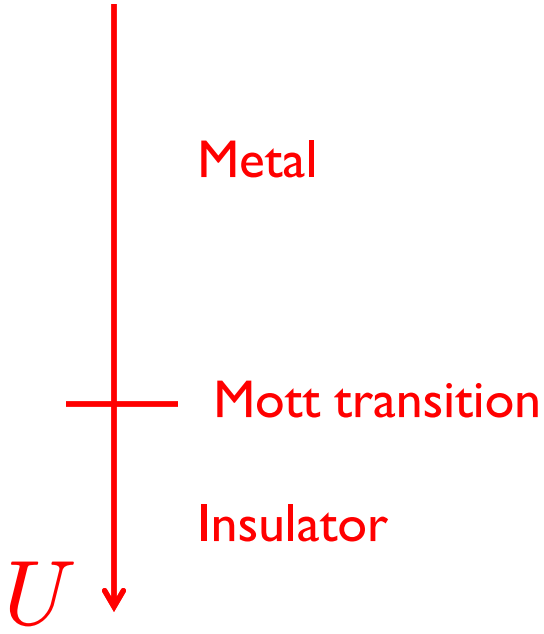
$$G_0^{-1}(i\omega_n) = \left(\sum_k \frac{1}{i\omega_n + \mu - \epsilon_k - \Sigma(i\omega_n)} \right)^{-1} + \Sigma(i\omega_n)$$

In DMFT the self-energy of the lattice is local: Z, m*, coherence temperature, lifetimes are constant along the Fermi surface

Example: the Mott transition

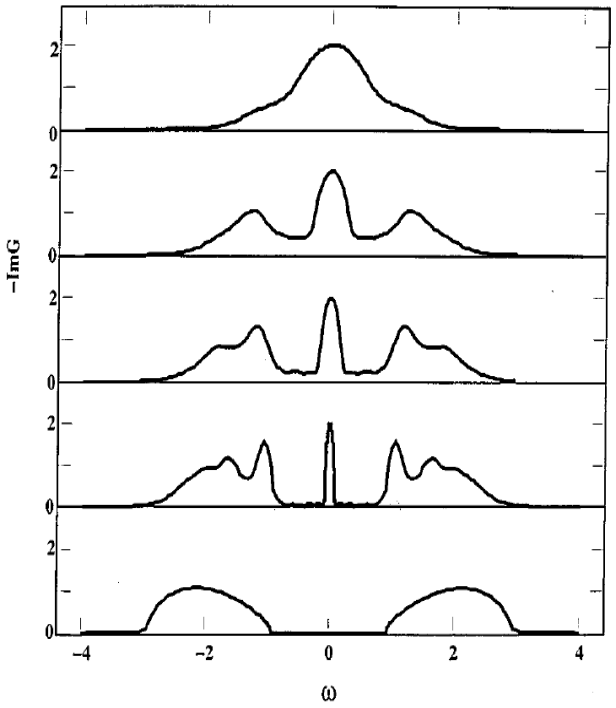
- The DMFT is good at describing transfer of spectral weight due to correlation effects
- Example: the destruction of coherent quasiparticles (small QP coherence scale, short lifetime) near Mott transition





What's the challenge?

- The impurity solver must compute the local Green's function
- Bath can have a rich structure, be gapped (insulators, superconductors)
- Structures appear at all scales (transfer of spectral weight in the Mott transition)
- The impurity solver must be able to treat many orbitals (e.g. pnictides)
- The interaction Hamiltonian must be generic (pair-hopping, spin flip terms)
- The model is studied in different temperature regimes
- One would like to be able to have real-frequency spectra

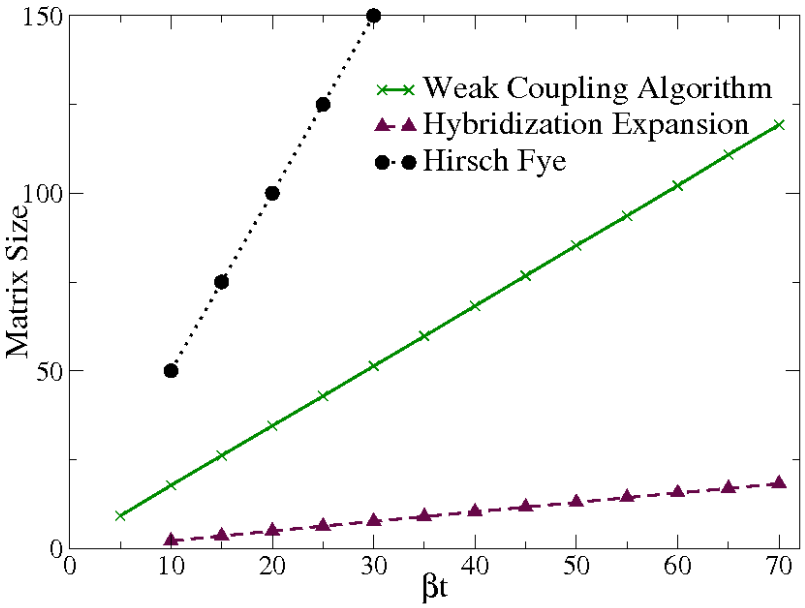


Outline

- Quantum impurity problems
- Continuous-time quantum Monte Carlo (CT-QMC) methods
- Generic introduction to Monte Carlo
- The interaction-expansion algorithm (CT-INT)
- The hybridization-expansion algorithm (CT-HYB)
- Some applications of CT-QMC solvers
- TRIQS: a Toolbox for Research on Quantum Interacting Systems
- Implementation details for the CT-INT

Continuous-time quantum Monte Carlo methods

- They have been a small revolution!
- There are different flavors:
- CT-INT: Interaction expansion
- CT-HYB: Hybridization expansion
- CT-AUX: Auxiliary-field formulation



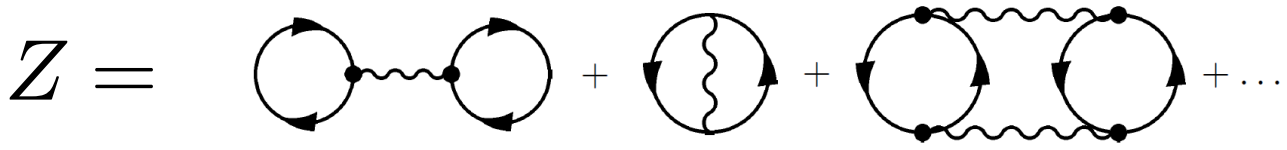
- The underlying principle is the same for all these algorithms
- They stem from earlier work on diagrammatic Monte Carlo
  Prokof'ev and Svistunov, PRL (1998)
  Prokof'ev et al., Sov. Phys. JETP (1998)

Some notable results

- DMFT used together with electronic structure calculations
- Rotationally invariant multi-orbital problems
- Kondo lattice
- Cluster DMFT (e.g. for cuprate superconductors)
- Fermionic cold-atomic systems
- 4-point correlators for extensions of DMFT (dual-fermions)
- First steps in non-equilibrium

Diagrammatic Monte Carlo

- Write the partition function in a perturbation theory



- Generate stochastically diagrams so that the statistical weight of a diagram is the absolute value of its contribution in the partition function expansion (Monte Carlo)
- This is possible even if the diagrams depend on continuous variables
- Use these diagrams to compute the partition function, Green's functions, etc.

Outline

- Quantum impurity problems
- Continuous-time quantum Monte Carlo (CT-QMC) methods
- Generic introduction to Monte Carlo
- The interaction-expansion algorithm (CT-INT)
- The hybridization-expansion algorithm (CT-HYB)
- Some applications of CT-QMC solvers
- TRIQS: a Toolbox for Research on Quantum Interacting Systems
- Implementation details for the CT-INT

Monte Carlo: a quick summary

- The Monte Carlo is a method to do sums:

$$\sum_x p(x) f(x), \qquad \text{where} \quad p(x) > 0, \quad \sum_x p(x) = 1$$

- The idea is to generate stochastically a large set of x_i such that the probability to find a given x_n is $p(x_n)$
- The original sum is replaced by an average over the set $\{x_i\}$

$$\sum_x p(x) f(x) \sim \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

- Note that x should be understood as a set of variables $x = (\tau, \sigma, \ldots)$
- One can think of x as a "configuration" in a phase space

Markov chain

- How do we generate configurations with the correct distribution?
- We don't sample independent configurations, but generate a Markov chain: $x_1 \to x_2 \to x_3 \to x_4 \to \dots$
- The transition probability $W_{x,y}$ to go from configuration x to a configuration y must satisfy:
  - Normalization: $$\sum_y W_{x,y} = 1$$
  - Ergodicity: one must be able to reach any configuration
  - Stationary distribution, balance condition:

$$\frac{d}{dt}p(x) = 0 \quad \Longrightarrow \quad \sum_{y \neq x} p(y) W_{y,x} - \sum_{y \neq x} p(x) W_{x,y} = 0$$

  - Detailed balance (sufficient but not necessary): $$\frac{W_{x,y}}{W_{y,x}} = \frac{p(y)}{p(x)}$$

Metropolis algorithm

- How do we get a transition probability satisfying these criteria?
- Proposal – rejection scheme:
  - Propose a change with a chosen proposal rate $P_{x,y}$
  - Accept this proposal with a probability $A_{x,y}$
  - Otherwise don't change the configuration
  - The total transition rate is $W_{x,y} = A_{x,y} P_{x,y}$
- Metropolis algorithm:

$$A_{x,y} = \min\left[1, \frac{P_{y,x} p(y)}{P_{x,y} p(x)}\right]$$

- It satisfies the detailed balance and therefore the Markov chain will be distributed according to $p(x)$

The fermionic sign problem

- Imagine we want to compute this average:

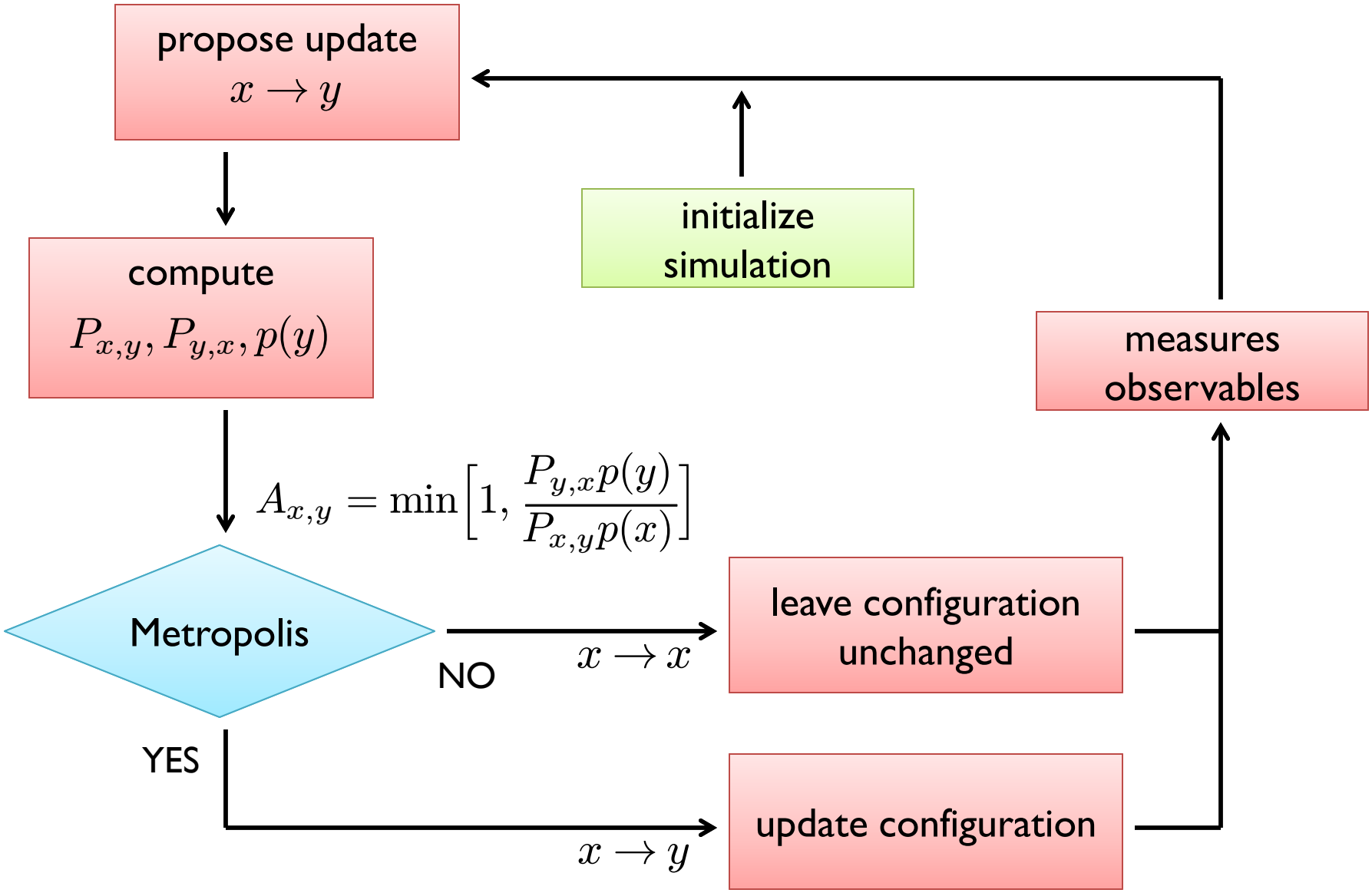
$$\langle f \rangle = \frac{\sum_x w(x) f(x)}{\sum_x w(x)}$$

- We would like to use $w(x)$ as a probability. But what if it can be negative?
- We must use the absolute value instead:

$$\langle f \rangle = \frac{\sum_x |w(x)| f(x) \mathrm{sign}(w(x))}{\sum_x |w(x)| \mathrm{sign}(w(x))} \sim \frac{\sum_{i=1}^N f(x_i) \mathrm{sign}(w(x_i))}{\sum_{i=1}^N \mathrm{sign}(w(x_i))}$$

- If signs alternate the denominator is very small and there is a big variance! Gets worse at low temperatures, big systems…
- Fermionic problems almost always suffer this sign problem!

A Monte Carlo algorithm

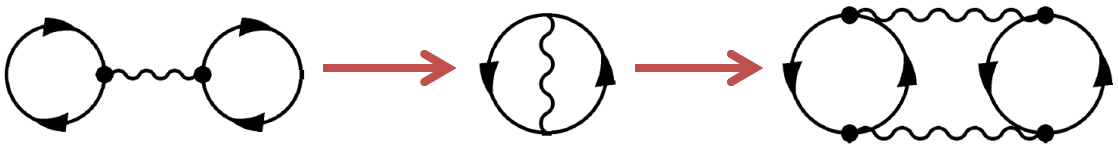


A couple of things to be careful about

- The measurements must start when the Markov chain has reached a stationary distribution. One must make sure to let the system thermalize!
- Successive points in the Markov chain are clearly correlated. These autocorrelation effects must be taken into account when computing the statistical errors:
  - Binning analysis
  - Jackknife
- Ideally one should measure only when configurations have become uncorrelated
- There are tools around to do these analysis!

Diagrammatic Monte Carlo

- Partition function: $$Z = \int_{\mathcal{C}} w(\mathcal{C}) \sim \sum_{\mathcal{C}}^{\mathrm{MC}} \mathrm{sign}(w(\mathcal{C}))$$
- Propose diagrams in a Markov chain:



- Accept these proposals with a rate (Metropolis) such that the diagrams appear with probability density $w(\mathcal{C})$
- From the generated configurations, compute the observables you are interested in $$\langle f \rangle = \frac{1}{Z} \int_{\mathcal{C}} w(\mathcal{C}) f(\mathcal{C}) \sim \frac{1}{Z} \sum_{\mathcal{C}}^{\mathrm{MC}} f(\mathcal{C}) \, \mathrm{sign}(w(\mathcal{C}))$$
- The different versions of continuous-time Monte Carlo solvers correspond to different choices of writing the partition function

Getting a series expansion for the partition function

- Write the action in two parts

$$S = S_A + S_B$$

- Express the partition function as

$$Z = \int \mathcal{D}[d^\dagger, d] e^{-S_A} e^{-S_B} = \int \mathcal{D}[d^\dagger, d] e^{-S_A} \sum_n \frac{(-1)^n}{n!} S_B^n$$

- Using that $\langle X \rangle = \frac{1}{Z} \int \mathcal{D}[d^\dagger, d] e^{-S} X$

we get $Z = Z_A \sum_n \frac{(-1)^n}{n!} \underbrace{\left\langle T_\tau S_B^n \right\rangle_A}$

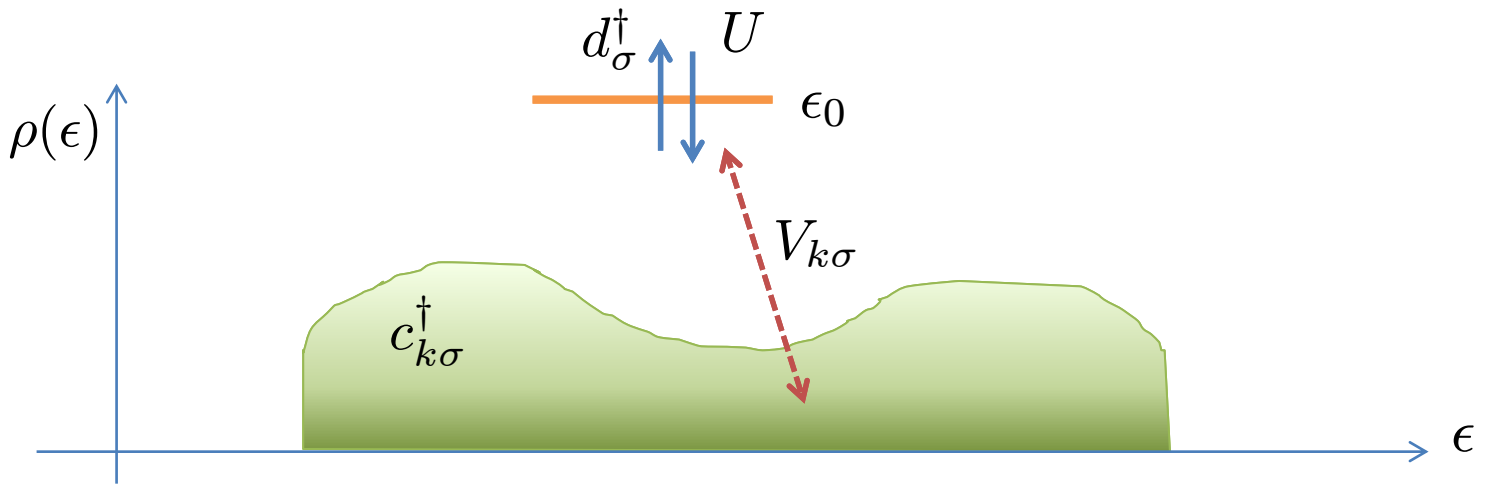
This is an average over the states described by the action A. It generally involves sums and integrals over imaginary time and can have a diagrammatical representation.

Outline

- Quantum impurity problems
- Continuous-time quantum Monte Carlo (CT-QMC) methods
- Generic introduction to Monte Carlo
- The interaction-expansion algorithm (CT-INT)
- The hybridization-expansion algorithm (CT-HYB)
- Some applications of CT-QMC solvers
- TRIQS: a Toolbox for Research on Quantum Interacting Systems
- Implementation details for the CT-INT

Interaction-expansion CT-QMC

- We focus on the simplest Anderson model (can be generalized to density-density multi-orbital problems)
- We want to derive an expansion around the non-interacting limit (expansion in the interaction)



$$\mathcal{H} = \sum_{\sigma=\uparrow,\downarrow} \epsilon_0 d^\dagger_\sigma d_\sigma + U n_{d\uparrow} n_{d\downarrow} + \sum_{k,\sigma} V_{k\sigma} c^\dagger_{k\sigma} d_\sigma + \text{h.c.} + \sum_{k,\sigma} \epsilon_{k\sigma} c^\dagger_{k\sigma} c_{k\sigma}$$

Interaction expansion cont'd

- We work in the imaginary-time formalism

$$Z = \int \mathcal{D}[d^\dagger, d] e^{-S} \quad \langle A \rangle = \frac{1}{Z} \int \mathcal{D}[d^\dagger, d] e^{-S} A$$

- The action for the Anderson model:

$$S = -\sum_\sigma \int_0^\beta d\tau d\tau' d_\sigma^\dagger(\tau) G_{0\sigma}^{-1}(\tau - \tau') d_\sigma(\tau') + \int_0^\beta d\tau U n_{d\uparrow}(\tau) n_{d\downarrow}(\tau)$$

- The partition function can be written as

$$Z = \int \underbrace{\mathcal{D}[d^\dagger, d] e^{-S_0}} \exp\left(\int_0^\beta d\tau (-U) n_{d\uparrow}(\tau) n_{d\downarrow}(\tau) \right)$$

this will produce a time-ordered average over the non-interacting state

Interaction expansion cont'd

- We write a series expansion for the exponential

$$
\begin{aligned}
Z &= Z_0 \left\langle T_\tau \exp\left(\int_0^\beta d\tau (-U) n_{d\uparrow}(\tau) n_{d\downarrow}(\tau) \right) \right\rangle_0 \\
&= Z_0 \left\langle T_\tau \sum_n \frac{(-U)^n}{n!} \left(\int_0^\beta d\tau n_{d\uparrow}(\tau) n_{d\downarrow}(\tau) \right)^n \right\rangle_0 \\
&= Z_0 \sum_n \int_0^\beta d\tau_1 \cdots d\tau_n \frac{(-U)^n}{n!} \left\langle T_\tau n_{d\uparrow}(\tau_1) n_{d\downarrow}(\tau_1) \cdots n_{d\uparrow}(\tau_n) n_{d\downarrow}(\tau_n) \right\rangle_0
\end{aligned}
$$

- At this stage we have a perturbation expansion for the partition function of the type:

$$
Z = \int_{\mathcal{C}} w(\mathcal{C}) \sim \sum_{\mathcal{C}}^{\mathrm{MC}} \mathrm{sign}(w(\mathcal{C}))
$$

- However there is an obvious sign problem!

Interaction expansion cont'd

- The trick is to rewrite the interaction differently

$$Un_\uparrow n_\downarrow = \frac{U}{2}\sum_{s=\uparrow,\downarrow}\Big(n_\uparrow - \alpha_{s\uparrow}\Big)\Big(n_\downarrow - \alpha_{s\downarrow}\Big) + \underbrace{\frac{U}{2}\Big(n_\uparrow + n_\downarrow\Big)} + \cancel{\text{const}}$$

$$\alpha_{s\sigma} = \frac{1}{2} + (2\delta_{s\sigma} - 1)\delta$$

we absorb this term in the chemical potential

- We eventually get (up and down spins decouple)

$$Z = Z_0 \sum_n \frac{(-U)^n}{n!}\int d\tau_i \frac{1}{2^n}\sum_{s_i} \Big\langle T_\tau (n_{d\uparrow}(\tau_1) - \alpha_{s_1\uparrow})\cdots(n_{d\uparrow}(\tau_n) - \alpha_{s_n\uparrow})\Big\rangle_0$$

$$\Big\langle T_\tau (n_{d\downarrow}(\tau_1) - \alpha_{s_1\downarrow})\cdots(n_{d\downarrow}(\tau_n) - \alpha_{s_n\downarrow})\Big\rangle_0$$

- By tuning δ we can improve the sign problem a lot!

Interaction expansion cont'd

- In the end we have

Sum over many (some continuous) variables

$$Z = Z_0 \sum_n \int d\tau_1 \cdots d\tau_n \sum_{s_1} \cdots \sum_{s_n}$$

$$\frac{(-U)^n}{n!2^n} \left\langle T_\tau (n_{d\uparrow}(\tau_1) - \alpha_{s_1\uparrow}) \cdots (n_{d\uparrow}(\tau_n) - \alpha_{s_n\uparrow}) \right\rangle_0$$

Product of two non-interacting averages

$$\left\langle T_\tau (n_{d\downarrow}(\tau_1) - \alpha_{s_1\downarrow}) \cdots (n_{d\downarrow}(\tau_n) - \alpha_{s_n\downarrow}) \right\rangle_0$$

$$Z = \int_{\mathcal{C}} w(\mathcal{C}) \sim \sum_{\mathcal{C}}^{\mathrm{MC}} \mathrm{sign}(w(\mathcal{C})) \quad \langle f \rangle = \frac{1}{Z} \int_{\mathcal{C}} w(\mathcal{C}) f(\mathcal{C}) \sim \frac{1}{Z} \sum_{\mathcal{C}}^{\mathrm{MC}} f(\mathcal{C}) \,\mathrm{sign}(w(\mathcal{C}))$$

- Now we need to find a way to compute the averages

Computing the averages

- The averages are on a non-interacting state. Therefore we can use Wick's theorem
- With this definition of the Green's function

$$G_{0\sigma}(\tau) = -T_\tau \langle c_\sigma(\tau) c_\sigma^\dagger(0) \rangle_0$$

we get

$$\left\langle (n_{d\sigma}(\tau_1) - \alpha_{s_1\sigma}) \cdots (n_{d\sigma}(\tau_n) - \alpha_{s_n\sigma}) \right\rangle_0 = \det D_n^\sigma =$$

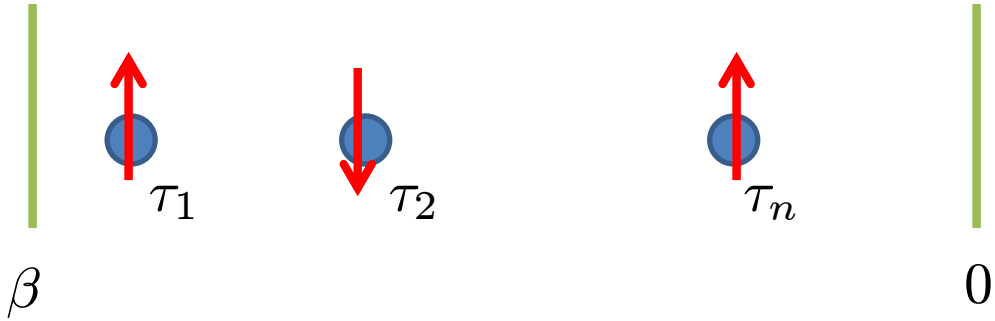
$$\begin{vmatrix} G_{0\sigma}(0^-) - \alpha_{s_1\sigma} & G_{0\sigma}(\tau_1 - \tau_2) & \cdots & \cdots \\ G_{0\sigma}(\tau_2 - \tau_1) & G_{0\sigma}(0^-) - \alpha_{s_2\sigma} & G_{0\sigma}(\tau_2 - \tau_3) & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & G_{0\sigma}(\tau_n - \tau_{n-1}) & G_{0\sigma}(0^-) - \alpha_{s_n\sigma} \end{vmatrix}$$

and $$Z = Z_0 \sum_n \int d\tau_1 \cdots d\tau_n \sum_{s_1} \cdots \sum_{s_n} \frac{(-U)^n}{n! 2^n} \det D_n^\uparrow \det D_n^\downarrow$$

Monte Carlo elements

- MC sum: $$\sum_n \int_{\tau_1 > \cdots > \tau_n} d\tau_1 \cdots d\tau_n \sum_{s_1} \cdots \sum_{s_n}$$

- The configurations are diagrams of the perturbation expansion. They can be seen as a set of interaction vertices at different imaginary times with an auxiliary spin s_i at every vertex.

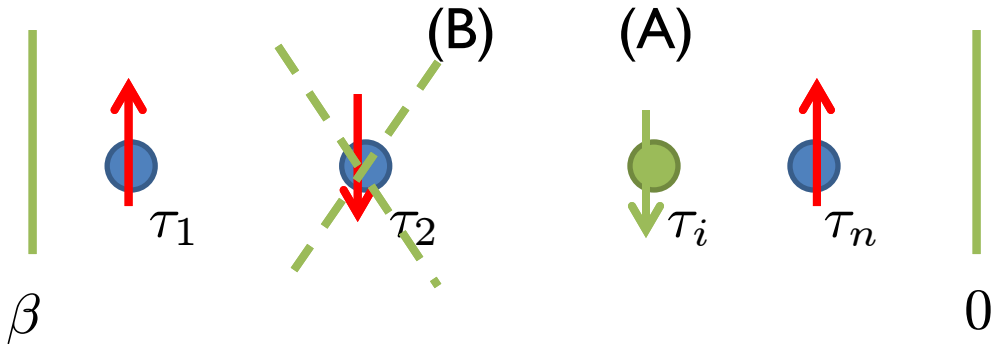


- The weight of every diagram is given by

$$\left(\frac{-U}{2}\right)^n \det D_n^{\uparrow} \det D_n^{\downarrow}$$

Generating diagrams

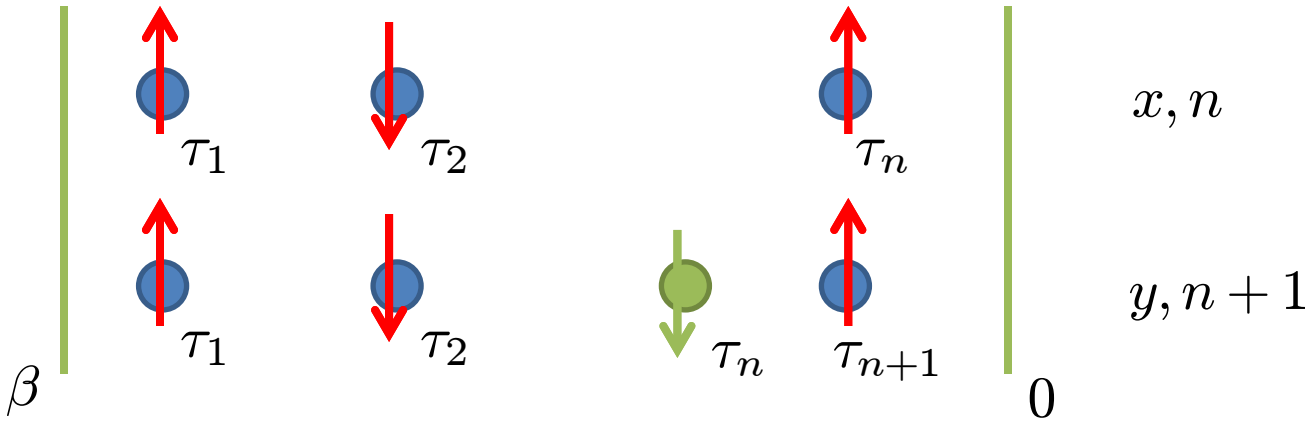
- We need to create a Markov chain of diagrams
- We can propose any changes to go from one diagram to another. A simple solution is to use two "moves":
- An insertion of a vertex: we pick a random imaginary time and insert a vertex with a spin randomly up or down (A)
- A removal of a vertex: pick a random vertex and remove it (B)



Insertion of a vertex

- What is the acceptance rate?

$$A_{x,y} = \min\left[1, \frac{P_{y,x}\rho(y)}{P_{x,y}\rho(x)}\right]$$



$$P_{x,y}\rho(x) = \frac{1}{2} \times \frac{1}{2} \times \frac{d\tau_{n+1}}{\beta} \times \left(\frac{-U}{2}\right)^n \det D_n^{\uparrow} \det D_n^{\downarrow} \prod_{i=1}^{n} d\tau_i$$

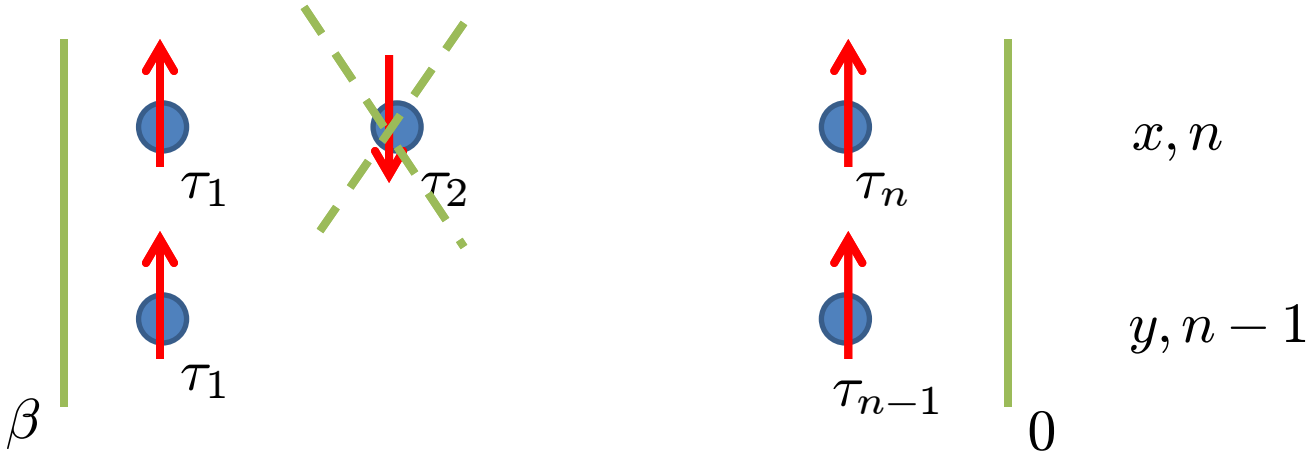
$$P_{y,x}\rho(y) = \frac{1}{2} \times \frac{1}{n+1} \times \left(\frac{-U}{2}\right)^{n+1} \det D_{n+1}^{\uparrow} \det D_{n+1}^{\downarrow} \prod_{i=1}^{n+1} d\tau_i$$

- Accept move with: $A_{x,y} = \min\left[1, \frac{-U\beta}{n+1} \times \frac{\det D_{n+1}^{\uparrow} D_{n+1}^{\downarrow}}{\det D_n^{\uparrow} D_n^{\downarrow}}\right]$

Removal of a vertex

- What is the acceptance rate?

$$A_{x,y} = \min\left[1, \frac{P_{y,x}\rho(y)}{P_{x,y}\rho(x)}\right]$$



$$P_{x,y}\rho(x) = \frac{1}{2} \times \frac{1}{n} \times \left(\frac{-U}{2}\right)^n \det D_n^{\uparrow} \det D_n^{\downarrow} \prod_{i=1}^{n} d\tau_i$$

$$P_{y,x}\rho(y) = \frac{1}{2} \times \frac{1}{2} \times \frac{d\tau_n}{\beta} \times \left(\frac{-U}{2}\right)^{n-1} \det D_{n-1}^{\uparrow} \det D_{n-1}^{\downarrow} \prod_{i=1}^{n-1} d\tau_i d\tau_i'$$

- Accept move with: $A_{x,y} = \min\left[1, \frac{-n}{U\beta} \times \frac{\det D_{n-1}^{\uparrow} D_{n-1}^{\downarrow}}{\det D_n^{\uparrow} D_n^{\downarrow}}\right]$

Getting the Green's function

- We know how to generate a distribution corresponding to the terms in the partition function
- Now we just need to find how to measure the Green's function from this distribution

$$G(\tau) = -T_\tau \langle d(\tau) d^\dagger(0) \rangle = \frac{1}{Z} \int \mathcal{D}[d^\dagger, d] e^{-S} d^\dagger(0) d(\tau)$$

$$S = -\sum_\sigma \int_0^\beta d\tau d\tau' d_\sigma^\dagger(\tau) G_{0\sigma}^{-1}(\tau - \tau') d_\sigma(\tau') + \int_0^\beta d\tau U n_{d\uparrow}(\tau) n_{d\downarrow}(\tau)$$

$$G_{0\sigma}^{-1}(\tau - \tau') = (\partial_{\tau-\tau'} - \epsilon_0)\delta(\tau - \tau') - \Delta_\sigma(\tau - \tau')$$

- We finally see that $G_\sigma(\tau) = -\frac{1}{\beta} \frac{\delta \ln Z}{\delta \Delta_\sigma(-\tau)}$

Getting the Green's function cont'd

- We have $$Z = Z_0 \int_{\mathcal{C}} \left(\frac{-U}{2}\right)^n \det D_n^{\uparrow} \det D_n^{\downarrow}$$

- The functional derivative eventually gives:

$$G_\sigma(i\omega_n) = G_{0\sigma}(i\omega_n) - \frac{1}{\beta} G_{0\sigma}^2(i\omega_n) \times \int_{\mathcal{C}} \sum_{ij} [D_n^\sigma]_{ij}^{-1} e^{i\omega_n(\tau_i - \tau_j)} \left(\frac{-U}{2}\right)^n \det D_n^{\uparrow} \det D_n^{\downarrow}$$

- So we see that we need to compute the following Monte Carlo average to get the Green's function

$$G_\sigma(i\omega_n) \sim G_{0\sigma}(i\omega_n) - \frac{1}{\beta Z} G_{0\sigma}^2(i\omega_n) \sum_{\mathcal{C}}^{\mathrm{MC}} \sum_{ij} [D_n^\sigma]_{ij}^{-1} e^{i\omega_n(\tau_i - \tau_j)} \times \mathrm{sign}(w(\mathcal{C}))$$

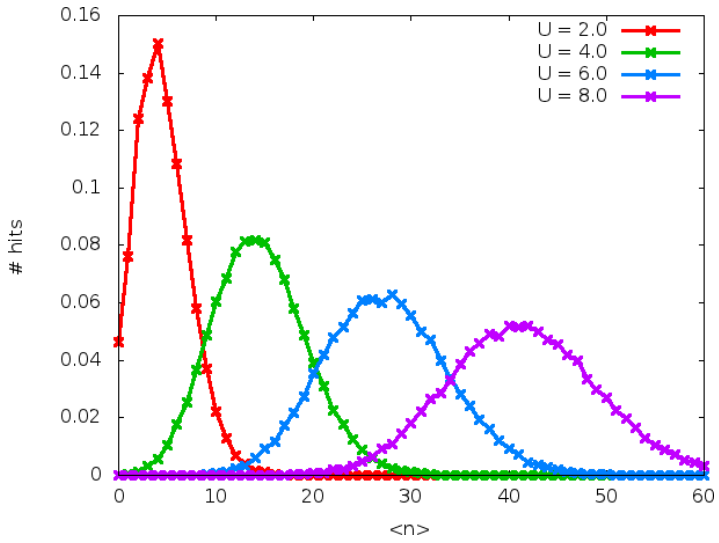
Computational effort

- The effort comes from the calculation of the determinants and of the inverse matrix (needed for the Green's function measure)

$$Z = Z_0 \int_{\mathcal{C}} \left(\frac{-U}{2}\right)^n \det D_n^{\uparrow} \det D_n^{\downarrow}$$

$$G_\sigma(i\omega_n) \sim G_{0\sigma}(i\omega_n) - \frac{1}{\beta Z} G_{0\sigma}^2(i\omega_n) \sum_{\mathcal{C}}^{\mathrm{MC}} \sum_{ij} [D_n^\sigma]_{ij}^{-1} e^{i\omega_n(\tau_i - \tau_j)} \times \mathrm{sign}(w(\mathcal{C}))$$

- It would be very slow to calculate them from scratch at every move
- They can be updated quickly using the Sherman-Morrison formula
- The computational effort grows in $\mathcal{O}(n^3)$

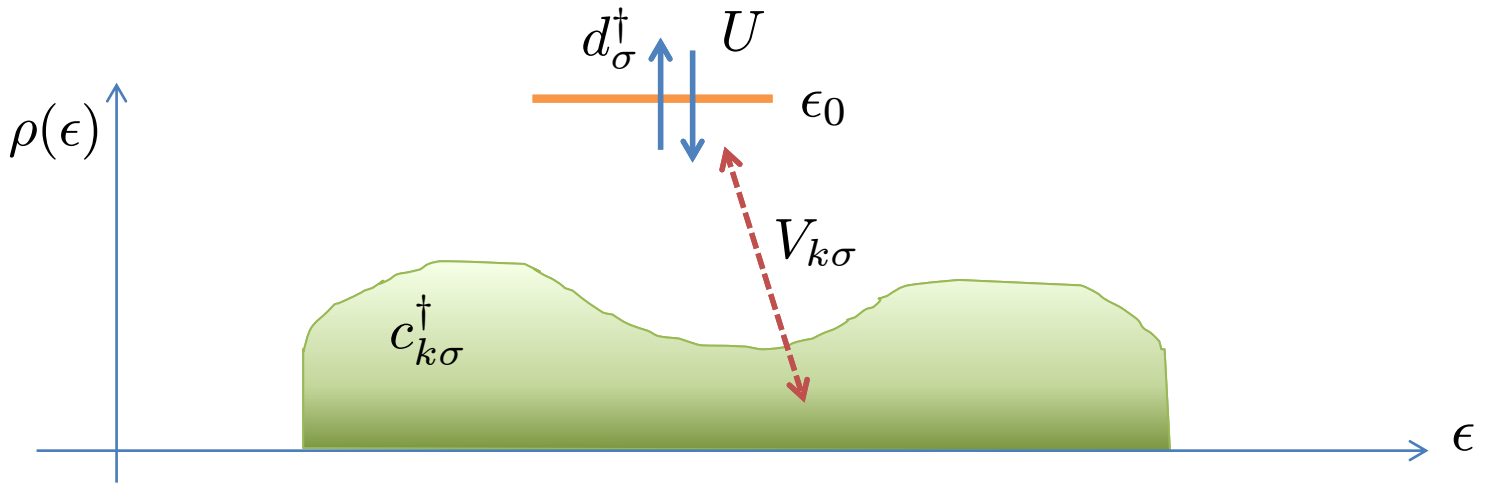


Outline

- Quantum impurity problems
- Continuous-time quantum Monte Carlo (CT-QMC) methods
- Generic introduction to Monte Carlo
- The interaction-expansion algorithm (CT-INT)
- The hybridization-expansion algorithm (CT-HYB)
- Some applications of CT-QMC solvers
- TRIQS: a Toolbox for Research on Quantum Interacting Systems
- Implementation details for the CT-INT

Hybridization-expansion CT-QMC

- We focus on the simplest Anderson model (can easily? be generalized to generic multi-orbital problems)
- We want to derive an expansion around the atomic limit (expansion in the hybridization)



$$\mathcal{H} = \sum_{\sigma=\uparrow,\downarrow} \epsilon_0 d^\dagger_\sigma d_\sigma + U n_{d\uparrow} n_{d\downarrow} + \sum_{k,\sigma} V_{k\sigma} c^\dagger_{k\sigma} d_\sigma + \text{h.c.} + \sum_{k,\sigma} \epsilon_{k\sigma} c^\dagger_{k\sigma} c_{k\sigma}$$

Hybridization expansion cont'd

- We work in the imaginary-time formalism

$$Z = \int \mathcal{D}[d^\dagger, d] e^{-S} \qquad \langle A \rangle = \frac{1}{Z} \int \mathcal{D}[d^\dagger, d] e^{-S} A$$

- The "standard" action for the Anderson model:

$$S = -\sum_\sigma \int_0^\beta d\tau d\tau' d_\sigma^\dagger(\tau) G_{0\sigma}^{-1}(\tau - \tau') d_\sigma(\tau') + \int_0^\beta d\tau U n_{d\uparrow}(\tau) n_{d\downarrow}(\tau)$$

$$G_{0\sigma}^{-1}(i\omega_n) = i\omega_n - \epsilon_0 - \Delta_\sigma(i\omega_n) \qquad \Delta_\sigma(i\omega_n) = \sum_k \frac{|V_{k\sigma}|^2}{i\omega_n - \epsilon_{k\sigma}}$$

- Rewrite the action as the sum of an "atomic" term containing all local terms U, ϵ_0 and a term with the hybridization to the bath:

$$S = S_{\rm loc} + \sum_\sigma \int_0^\beta d\tau d\tau' d_\sigma^\dagger(\tau) \Delta_\sigma(\tau - \tau') d_\sigma(\tau') = S_{\rm loc} + \sum_\sigma S_{\rm hyb}^\sigma$$

Hybridization expansion cont'd

$$Z = \int \mathcal{D}[d^\dagger, d] e^{-S_{\rm loc} - \sum_\sigma S^\sigma_{\rm hyb}} = \int \mathcal{D}[d^\dagger, d] e^{-S_{\rm loc}} \prod_\sigma \Big[\sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \big(S^\sigma_{\rm hyb}\big)^n \Big]$$

$$Z = \sum_{n_\uparrow, n_\downarrow = 0}^{\infty} \Big\langle T_\tau \prod_\sigma \frac{(-1)^{n_\sigma}}{n_\sigma!} \big(\hat{S}^\sigma_{\rm hyb}\big)^{n_\sigma} \Big\rangle_{\rm loc}$$

Sum over many (continuous) variables

$$Z = \sum_{n_\uparrow, n_\downarrow = 0}^{\infty} \int_0^\beta d\tau_1^\uparrow \dots d\tau_{n_\uparrow}'^\uparrow \int_0^\beta d\tau_1^\downarrow \dots d\tau_{n_\downarrow}'^\downarrow$$

Product of hybridization functions

$$\prod_\sigma \frac{(-1)^{n_\sigma}}{n_\sigma!} \prod_{i=1}^{n_\sigma} \Delta_\sigma(\tau_i^\sigma - \tau_i'^\sigma) \times$$

Trace involving both spin up and down operators

$$\mathrm{Tr}\Big[e^{-\beta \mathcal{H}_{\rm loc}} T_\tau \prod_{i=1}^{n_\uparrow} d^\dagger_\uparrow(\tau_i^\uparrow) d_\uparrow(\tau_i'^\uparrow) \prod_{i=1}^{n_\downarrow} d^\dagger_\downarrow(\tau_i^\downarrow) d_\downarrow(\tau_i'^\downarrow) \Big]$$

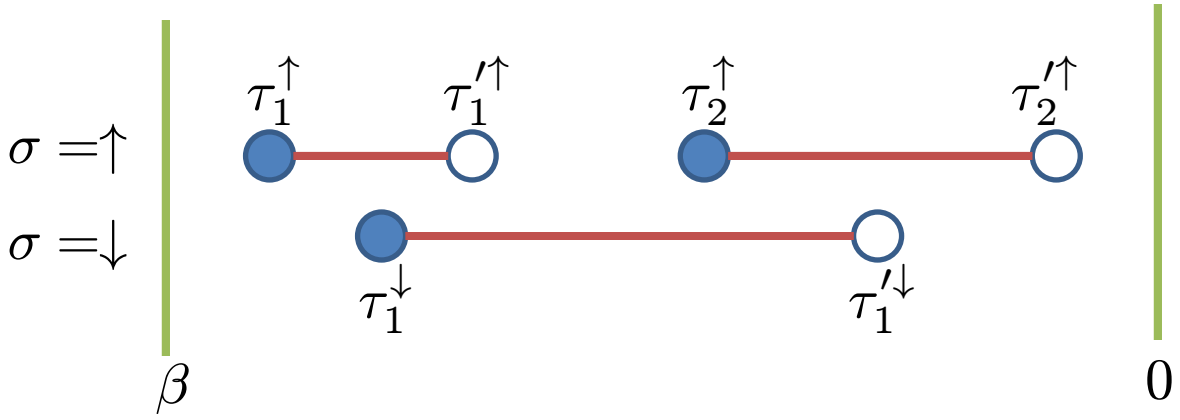
$$Z = \int_{\mathcal{C}} w(\mathcal{C}) \sim \sum_{\mathcal{C}}^{\rm MC} \mathrm{sign}(w(\mathcal{C})) \qquad \langle f \rangle = \frac{1}{Z} \int_{\mathcal{C}} w(\mathcal{C}) f(\mathcal{C}) \sim \frac{1}{Z} \sum_{\mathcal{C}}^{\rm MC} f(\mathcal{C}) \, \mathrm{sign}(w(\mathcal{C}))$$

Hybridization expansion cont'd

- MC sum:

$$\sum_{n_\uparrow, n_\downarrow = 0}^{\infty} \int_0^\beta d\tau_1^\uparrow \dots d\tau_{n_\uparrow}'^\uparrow \int_0^\beta d\tau_1^\downarrow \dots d\tau_{n_\downarrow}'^\downarrow$$

- Diagrams:



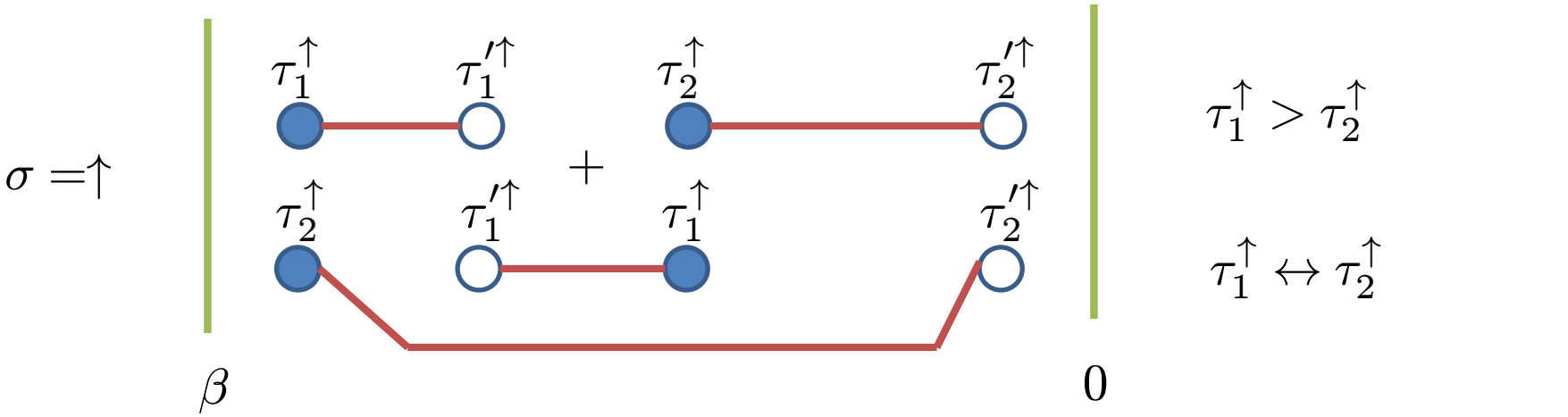
- Weight:

$$\mathrm{Tr}\left[e^{-\beta \mathcal{H}_{\mathrm{loc}}} T_\tau \prod_{i=1}^{n_\uparrow} d_\uparrow^\dagger(\tau_i^\uparrow) d_\uparrow(\tau_i'^\uparrow) \prod_{i=1}^{n_\downarrow} d_\downarrow^\dagger(\tau_i^\downarrow) d_\downarrow(\tau_i'^\downarrow)\right] \times \prod_\sigma \frac{(-1)^{n_\sigma}}{n_\sigma!} \prod_{i=1}^{n_\sigma} \Delta_\sigma(\tau_i^\sigma - \tau_i'^\sigma)$$

- Unfortunately these diagrams have alternating signs $\Rightarrow$ problems!

Trick: resumming diagrams

- The idea is to resum diagrams into a determinant. We start from a diagram where $\tau_1^\uparrow > \ldots > \tau_{n_\uparrow}^\uparrow$ and sum all the permutations of $\{\tau_i^\uparrow\}$



$$\mathrm{Tr}\left[e^{-\beta\mathcal{H}_{\mathrm{loc}}} d_\uparrow^\dagger(\tau_1^\uparrow) d_\uparrow(\tau_1'^\uparrow) d_\uparrow^\dagger(\tau_2^\uparrow) d_\uparrow(\tau_2'^\uparrow)\right] \times \frac{1}{2}\Delta_\uparrow(\tau_1^\uparrow - \tau_1'^\uparrow)\Delta_\uparrow(\tau_2^\uparrow - \tau_2'^\uparrow)$$

$$(-1)\mathrm{Tr}\left[e^{-\beta\mathcal{H}_{\mathrm{loc}}} d_\uparrow^\dagger(\tau_1^\uparrow) d_\uparrow(\tau_1'^\uparrow) d_\uparrow^\dagger(\tau_2^\uparrow) d_\uparrow(\tau_2'^\uparrow)\right] \times \frac{1}{2}\Delta_\uparrow(\tau_2^\uparrow - \tau_1'^\uparrow)\Delta_\uparrow(\tau_1^\uparrow - \tau_2'^\uparrow)$$

$$=$$

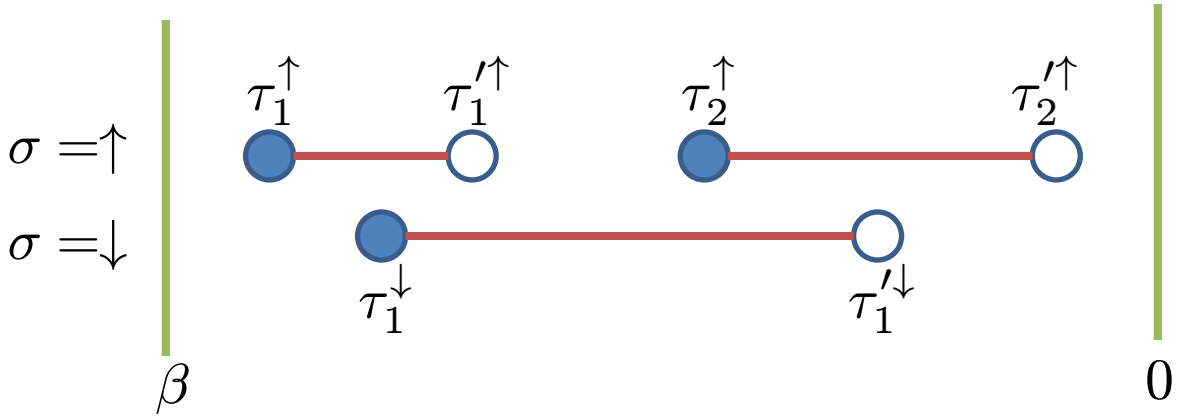
$$\mathrm{Tr}\left[e^{-\beta\mathcal{H}_{\mathrm{loc}}} d_\uparrow^\dagger(\tau_1^\uparrow) d_\uparrow(\tau_1'^\uparrow) d_\uparrow^\dagger(\tau_2^\uparrow) d_\uparrow(\tau_2'^\uparrow)\right] \times \frac{1}{2} \det_{1\leq k,l\leq 2} \Delta_\uparrow(\tau_k^\uparrow - \tau_l'^\uparrow)$$

Hybridization expansion Monte Carlo

- MC sum:

$$\sum_{n_\uparrow,n_\downarrow=0}^{\infty} \int_{\substack{\tau_1^\uparrow>\ldots>\tau_n^\uparrow \\ \tau_1'^\uparrow>\ldots>\tau_n'^\uparrow}} d\tau_1^\uparrow \ldots d\tau_n'^\uparrow \int_{\substack{\tau_1^\downarrow>\ldots>\tau_n^\downarrow \\ \tau_1'^\downarrow>\ldots>\tau_n'^\downarrow}} d\tau_1^\downarrow \ldots d\tau_{n_\downarrow}'^\downarrow$$

- Diagrams:



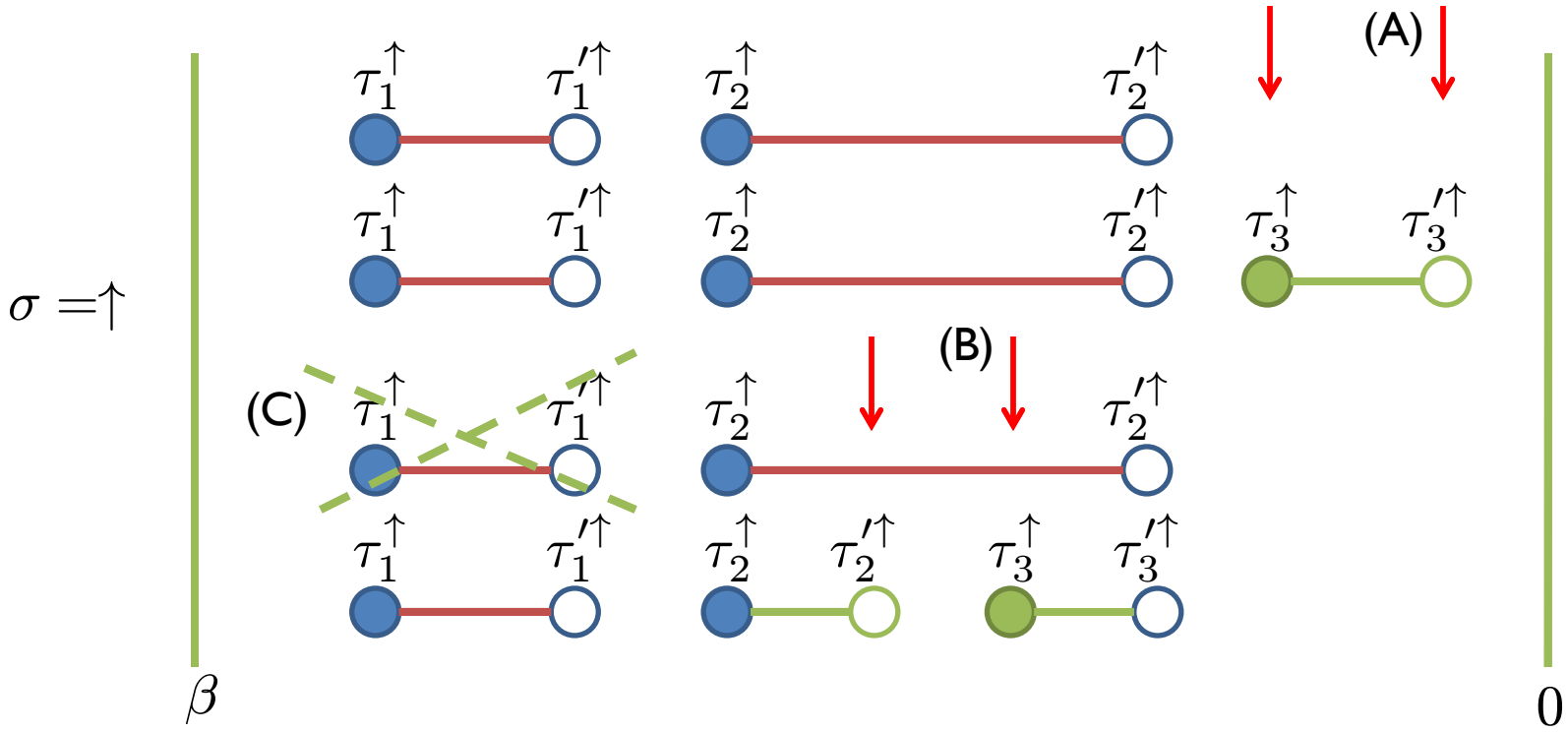
- Weight:

$$\mathrm{Tr}\left[e^{-\beta\mathcal{H}_{\mathrm{loc}}}T_\tau \prod_{i=1}^{n_\uparrow} d_\uparrow^\dagger(\tau_i^\uparrow) d_\uparrow(\tau_i'^\uparrow) \prod_{i=1}^{n_\downarrow} d_\downarrow^\dagger(\tau_i^\downarrow) d_\downarrow(\tau_i'^\downarrow)\right] \times$$

$$(-1)^{n_\uparrow+n_\downarrow} \det_{1\le k,l\le n_\uparrow} \Delta_\uparrow(\tau_k^\uparrow-\tau_l'^\uparrow) \times \det_{1\le k,l\le n_\downarrow} \Delta_\downarrow(\tau_k^\downarrow-\tau_l'^\downarrow)$$

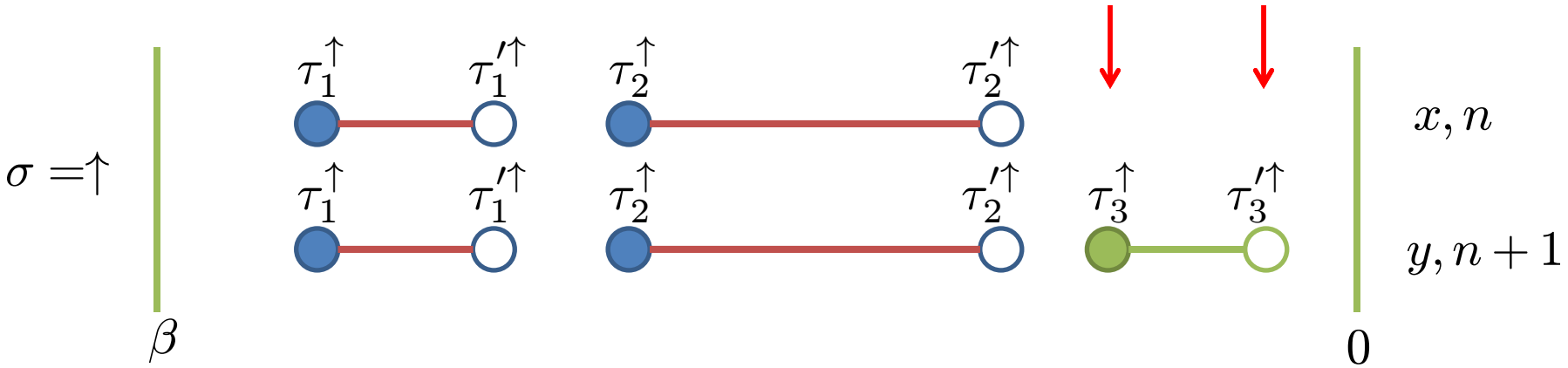
Generating diagrams

- New diagrams are generated with two "moves":
- Insertion of an (anti)-link: chose a spin flavor and pick two random imaginary times such that there is no operator between them. Either construct a link (A) or an anti-link (B)
- Removal of a link: chose a spin flavor and remove a random link (C)



Insertion of an (anti)-link

- What is the acceptance rate for this move? $A_{x,y} = \min\left[1, \frac{P_{y,x}\rho(y)}{P_{x,y}\rho(x)}\right]$



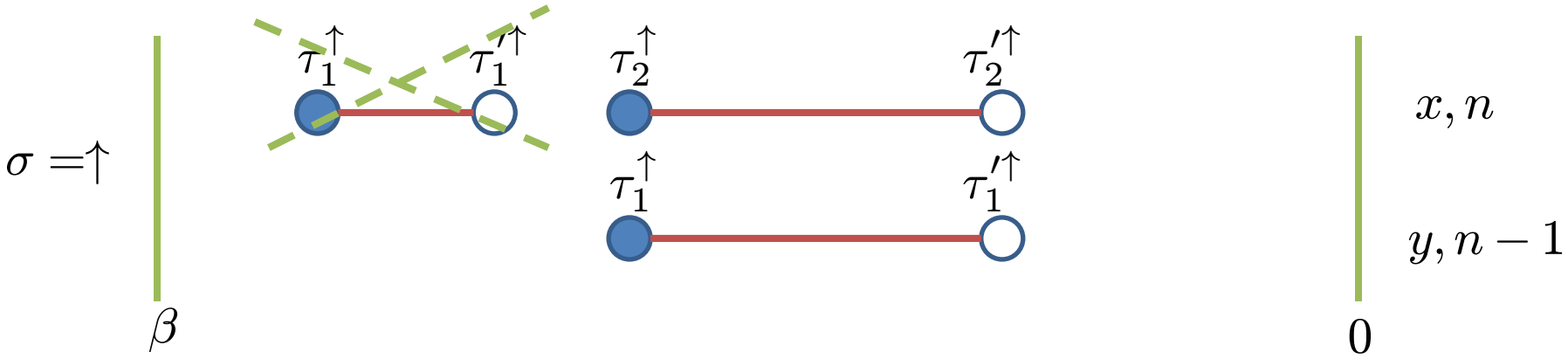
$$P_{x,y}\rho(x) = \frac{1}{2} \times \frac{d\tau_{n+1}}{\beta} \frac{d\tau'_{n+1}}{l_{\max}} \times \mathrm{Tr}\mathcal{C}_x \det \Delta_{\mathcal{C}_x} \prod_{i=1}^{n} d\tau_i d\tau'_i$$

$$P_{y,x}\rho(y) = \frac{1}{2} \times \frac{1}{n+1} \times \mathrm{Tr}\mathcal{C}_y \det \Delta_{\mathcal{C}_y} \prod_{i=1}^{n+1} d\tau_i d\tau'_i$$

- Accept with probability: $A_{x,y} = \min\left[1, \frac{\beta l_{\max}}{n+1} \times \frac{\mathrm{Tr}\mathcal{C}_y \det \Delta_{\mathcal{C}_y}}{\mathrm{Tr}\mathcal{C}_x \det \Delta_{\mathcal{C}_x}}\right]$

Removal of a link

- What is the acceptance rate for this move? $A_{x,y} = \min\left[1, \frac{P_{y,x}\rho(y)}{P_{x,y}\rho(x)}\right]$



$$P_{x,y}\rho(x) = \frac{1}{2} \times \frac{1}{n} \times \mathrm{Tr}\mathcal{C}_x \det \Delta_{\mathcal{C}_x} \prod_{i=1}^{n} d\tau_i d\tau_i'$$

$$P_{y,x}\rho(y) = \frac{1}{2} \times \frac{d\tau_n}{\beta} \frac{d\tau_n'}{l_{\max}} \times \mathrm{Tr}\mathcal{C}_y \det \Delta_{\mathcal{C}_y} \prod_{i=1}^{n-1} d\tau_i d\tau_i'$$

- Accept with probability: $A_{x,y} = \min\left[1, \frac{n}{\beta l_{\max}} \times \frac{\mathrm{Tr}\mathcal{C}_y \det \Delta_{\mathcal{C}_y}}{\mathrm{Tr}\mathcal{C}_x \det \Delta_{\mathcal{C}_x}}\right]$

Measuring the Green's function

- We know how to sample diagrams with weights corresponding to their contribution in the partition function.

$$Z = \int_{\mathcal{C}} (-1)^{n_\uparrow + n_\downarrow} \det \Delta_{\uparrow\mathcal{C}} \det \Delta_{\downarrow\mathcal{C}} \mathrm{Tr}\mathcal{C} = \int_{\mathcal{C}} w(\mathcal{C}) \sim \sum_{\mathcal{C}}^{\mathrm{MC}} \mathrm{sign}(w(\mathcal{C}))$$

- How do we get the Green's function?

$$G_\sigma(\tau) = -\frac{1}{\beta} \frac{\delta \log Z}{\delta \Delta_\sigma(-\tau)}$$

$$G_\uparrow(\tau) = \frac{-1}{Z\beta} \int_{\mathcal{C}} \frac{\delta \det \Delta_{\uparrow\mathcal{C}}}{\delta \Delta_\uparrow(-\tau)} \times (-1)^{n_\uparrow + n_\downarrow} \det \Delta_{\downarrow\mathcal{C}} \mathrm{Tr}\mathcal{C}$$

$$G_\uparrow(\tau) = \frac{-1}{Z\beta} \int_{\mathcal{C}} \sum_{k,l} \delta(\tau_k^\uparrow - \tau_l'^\uparrow + \tau) [\Delta_{\uparrow\mathcal{C}}^{-1}]_{k,l} \times \underbrace{(-1)^{n_\uparrow + n_\downarrow} \det \Delta_{\uparrow\mathcal{C}} \det \Delta_{\downarrow\mathcal{C}} \mathrm{Tr}\mathcal{C}}_{w(\mathcal{C})}$$

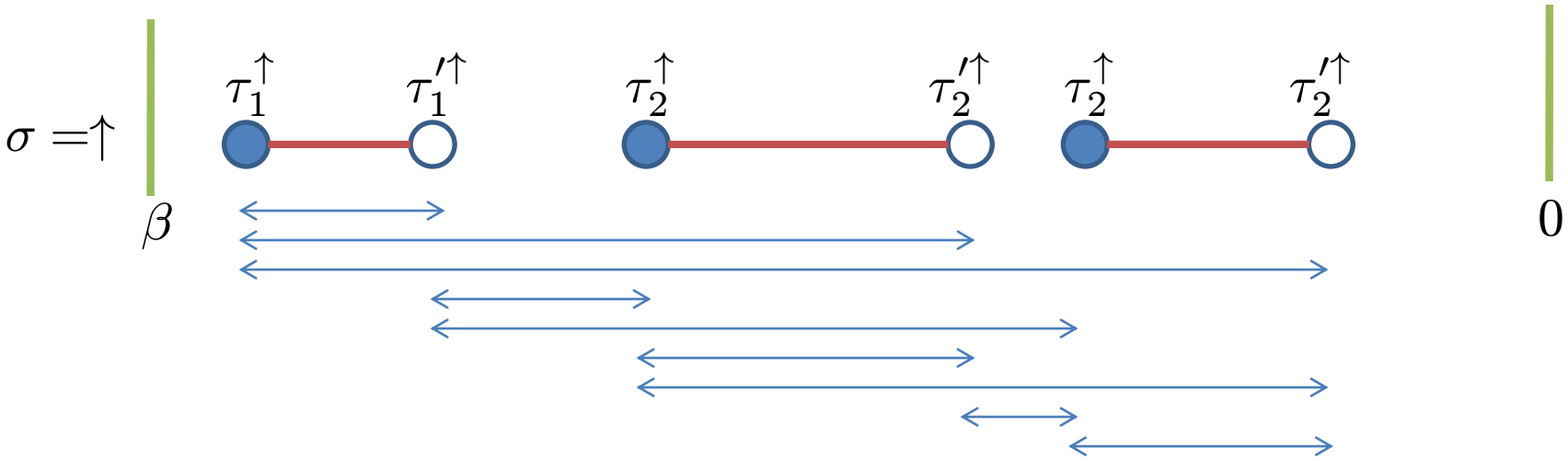
- Measure:

$$G_\sigma(\tau) \sim \frac{-1}{Z\beta} \sum_{\mathcal{C}}^{\mathrm{MC}} \sum_{k,l} \delta(\tau_k^\sigma - \tau_l'^\sigma + \tau) \times [\Delta_{\sigma\mathcal{C}}^{-1}]_{k,l} \times \mathrm{sign}(w(\mathcal{C}))$$

Measuring the Green's function cont'd

- Each configuration give contributions for a discrete set of imaginary times:

$$G_\sigma(\tau) \sim \frac{-1}{Z\beta} \sum_{\mathcal{C}}^{\mathrm{MC}} \sum_{k,l} \delta(\tau_k^\sigma - \tau_l'^\sigma + \tau) \times [\Delta_{\sigma\mathcal{C}}^{-1}]_{k,l} \times \mathrm{sign}(w(\mathcal{C}))$$

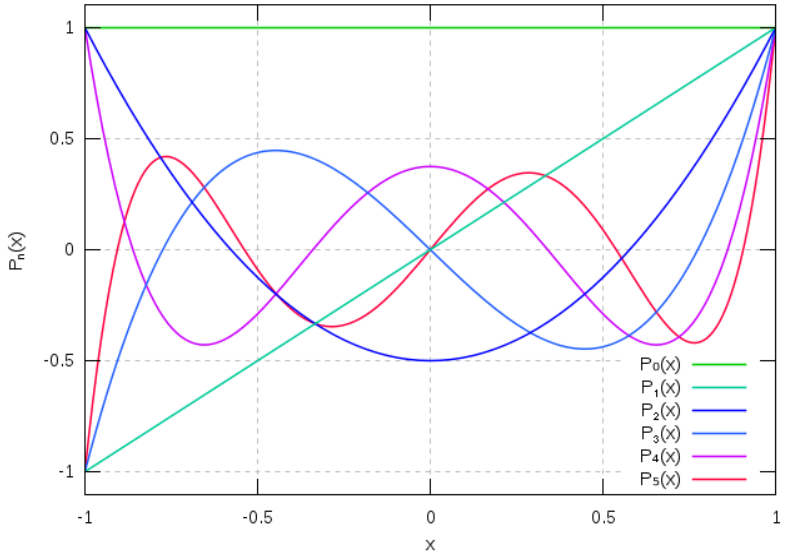


- These contribution can be "binned" on a very fine imaginary-time grid. This induces high frequency noise in Matsubara frequencies

Measuring using Legendre polynomials

L. Boehnke et al., PRB (2011)

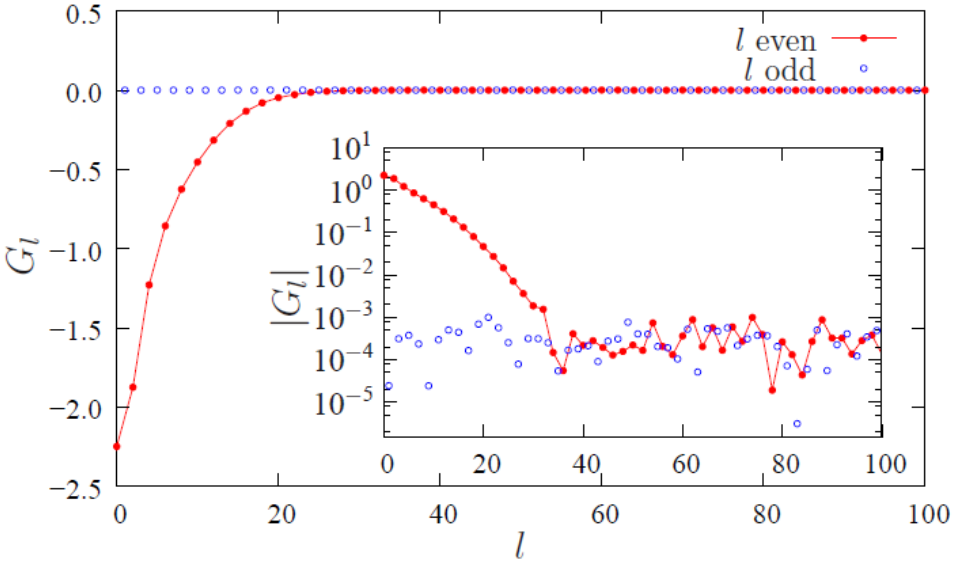
- Legendre polynomials are a basis to express function defined over an interval



- We can express the imaginary-time Green's function in this basis

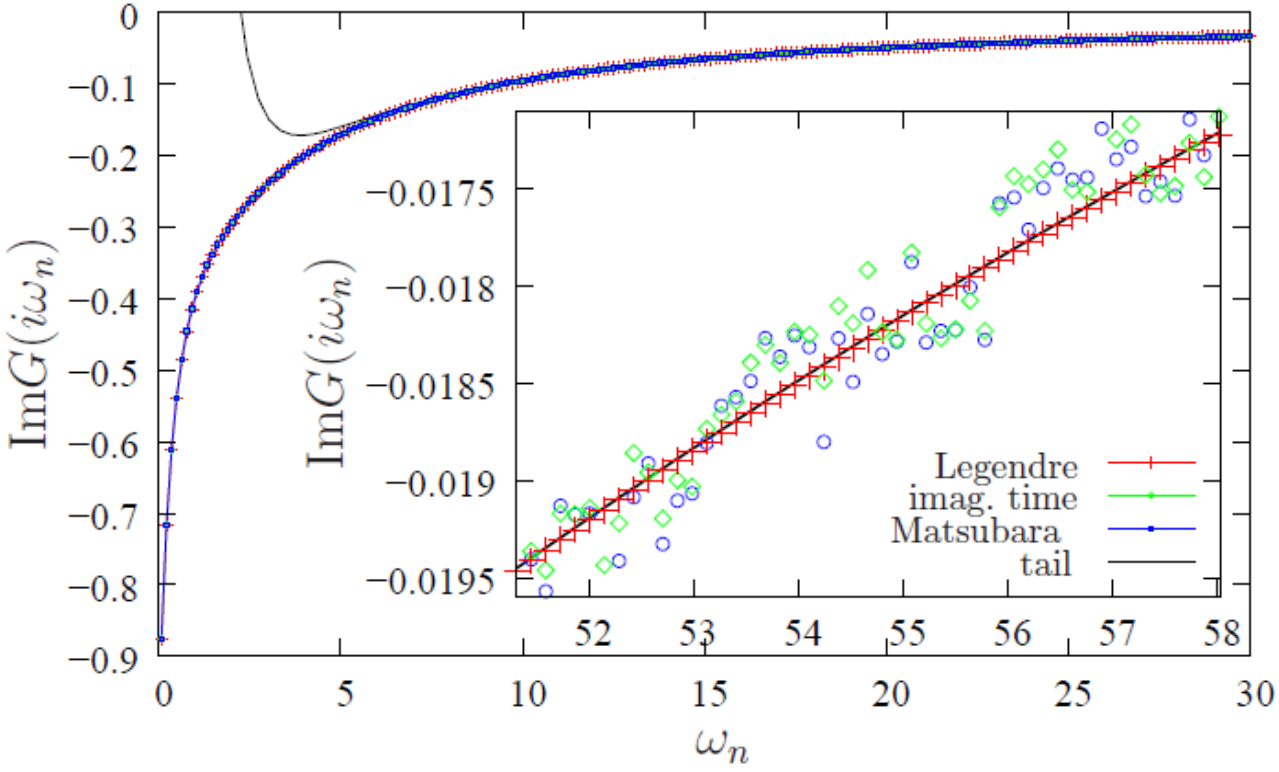
$$G(\tau) = \sum_{l \geq 0} \frac{\sqrt{2l+1}}{\beta} P_l[x(\tau)] G_l$$

- The coefficients G_l in this basis decay very quickly



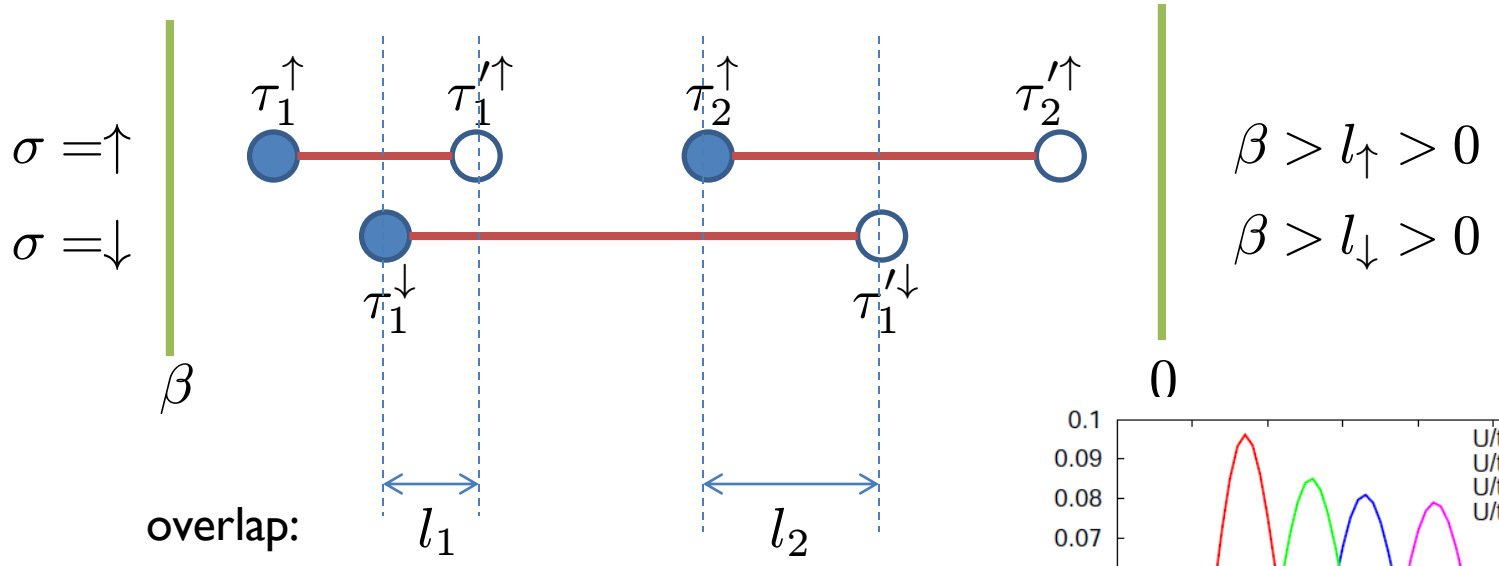
Legendre basis acting as a noise filter

- The noise in the Matsubara frequencies can be reduced by truncating the Legendre coefficients that are zero within their error bars
- A typical outcome of this procedure:



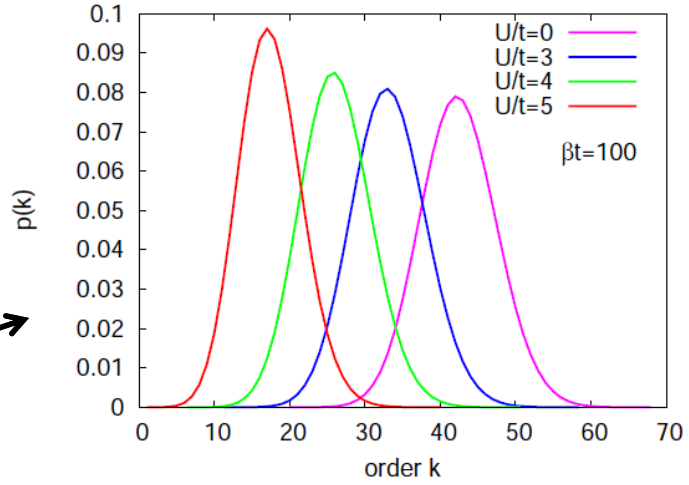
Computational effort

- Can the contribution of a diagram be computed quickly?
- Determinants can be updated quickly (Sherman-Morrison)
- For simple Hamiltonians, the trace is very easy



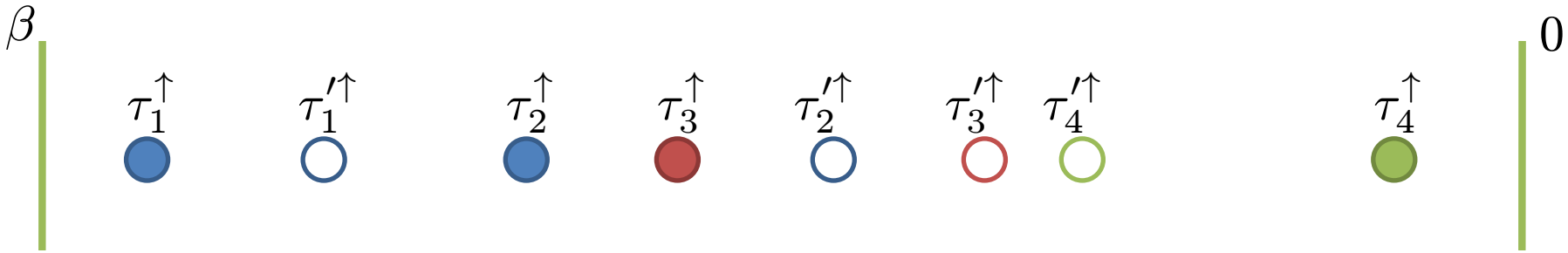
$$\mathrm{Tr}\,\mathcal{C} = e^{(l_\uparrow + l_\downarrow)\mu - U(l_1 + l_2)}$$

- Computational effort grows in $\mathcal{O}(n^3)$



What about non density-density Hamiltonians?

- The hybridization expansion algorithm can be modified for generic Hamiltonians
- Configurations are a set of creation / destruction operators of different flavor on a single imaginary-time line



- The main drawback is that there is no longer a quick way to compute the trace
- Operators are matrices that must be multiplied and traced over all atomic states
- The number of these atomic states quickly becomes huge with several orbitals

CT-INT versus CT-HYB

- CT-INT & CT-AUX: series in the interaction
  - Many orbitals, weak coupling, high temperatures
  - density-density Hamiltonians
  - Average perturbation order $\sim \beta U$

- CT-HYB: series in the hybridization function
  - Good at low temperatures, strong coupling
  - Can treat generic Hamiltonians
  - Hard to treat many orbitals
  - Average perturbation order is the kinetic energy

Pros and cons of the CT-QMC algorithms

- Pros:
  - Faster than before
  - Monte Carlo $\Rightarrow$ can easily be parallelized
  - Flexible Hamiltonians (not CT-INT)
  - Good scaling with number of orbitals if density-density
- Cons:
  - Many orbitals difficult with generic Hamiltonian
  - They are mainly in imaginary time, so one needs to do analytical continuation, and this is a very delicate procedure!
  - Note: real-time algorithms have been developed. However the sign problem is a lot more present there.
  - Sign problem

Outline

- Quantum impurity problems
- Continuous-time quantum Monte Carlo (CT-QMC) methods
- Generic introduction to Monte Carlo
- The interaction-expansion algorithm (CT-INT)
- The hybridization-expansion algorithm (CT-HYB)
- Some applications of CT-QMC solvers
- TRIQS: a Toolbox for Research on Quantum Interacting Systems
- Implementation details for the CT-INT

What did the CT-QMC algorithms change?

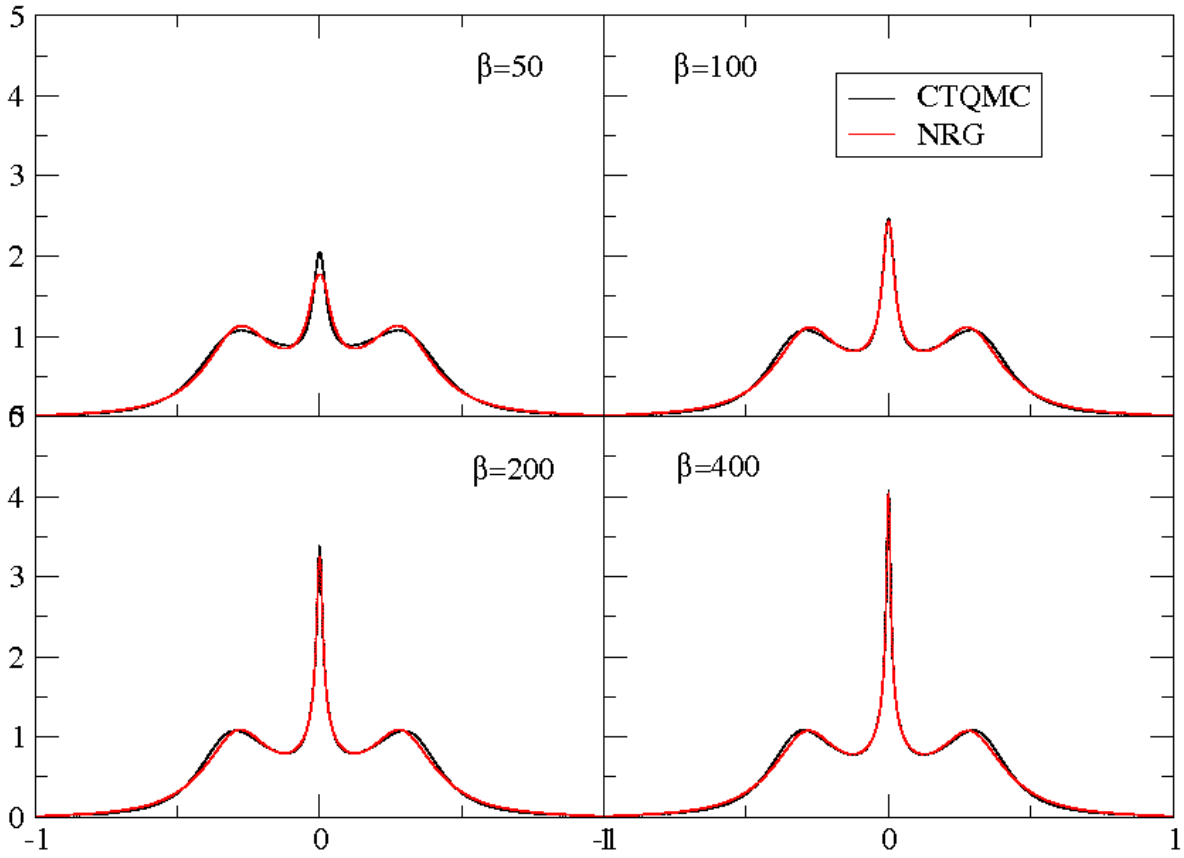
- In "simple" model studies, CT-QMC algorithms allow to:
  - Reach much lower temperatures
  - Acquire high-quality imaginary-time data and do more reliable analytical continuations

- In the study of realistic materials, they allow to treat the full SU(2) exchange in multi-orbital problems (e.g. pnictides, nickelates, ruthenates, …)

- More involved models can be studied
  - Cluster extensions of DMFT
  - Dual-fermion approaches
  - Holstein-Hubbard model, out-of-equilibrium

State of the art: NRG versus CT-QMC

- Direct comparison between NRG and CT-QMC for a half-filled Anderson impurity model (flat "Wilson" bath)
- Parameters are: $W = 1, U = 0.5, U/\pi\Gamma = 2.5$

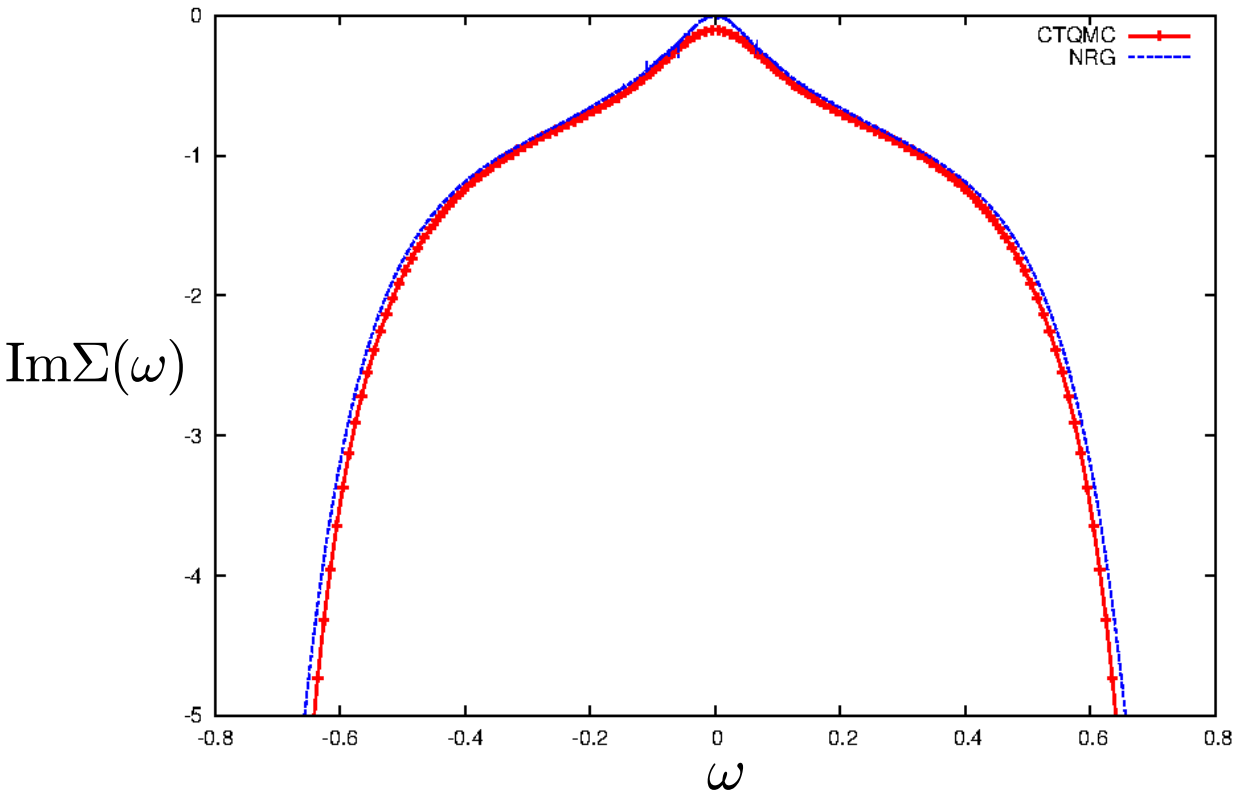


Analytical continuation of the CT-QMC self-energy using Padé approximants

NRG: Rok Zitko
CT-QMC: Jernej Mravlje

Detailed structures of self-energy

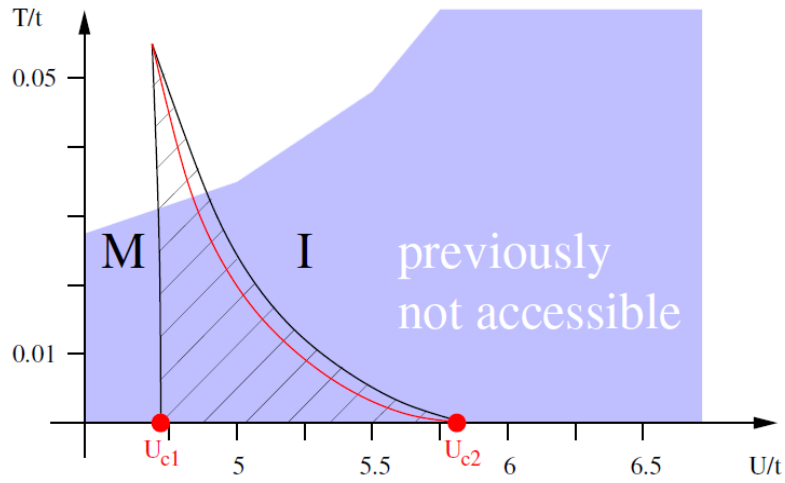
- Even fine details of the self-energy are captured correctly!
- Single-band Bethe lattice DMFT calculation show very good agreement, where only small temperature effect induce a difference between NRG and CT-QMC

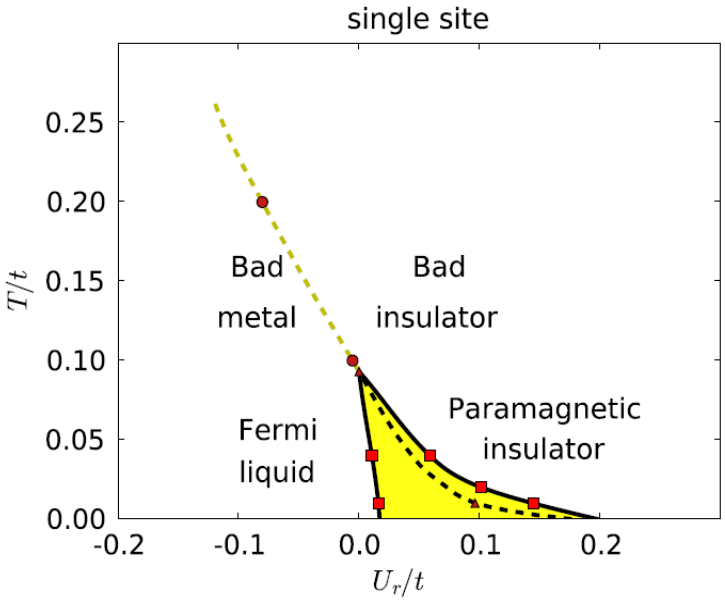


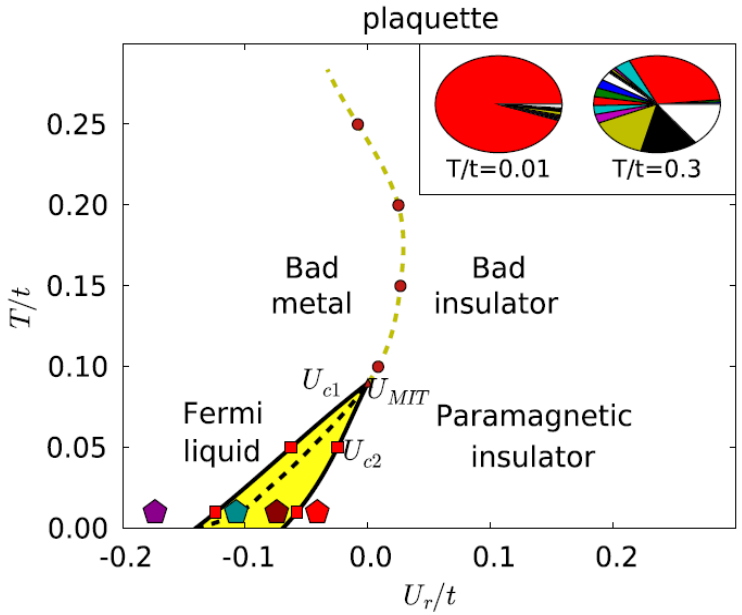
Mott transition in the Hubbard model

Werner et al., PRB (2007)

- Single-site DMFT studies have shown that the transition is first order at finite temperature, but second order at zero temperature
- DMFT cluster extensions have been considered (more later)







Not only single-site DMFT for the Hubbard model

- Holstein-Hubbard model

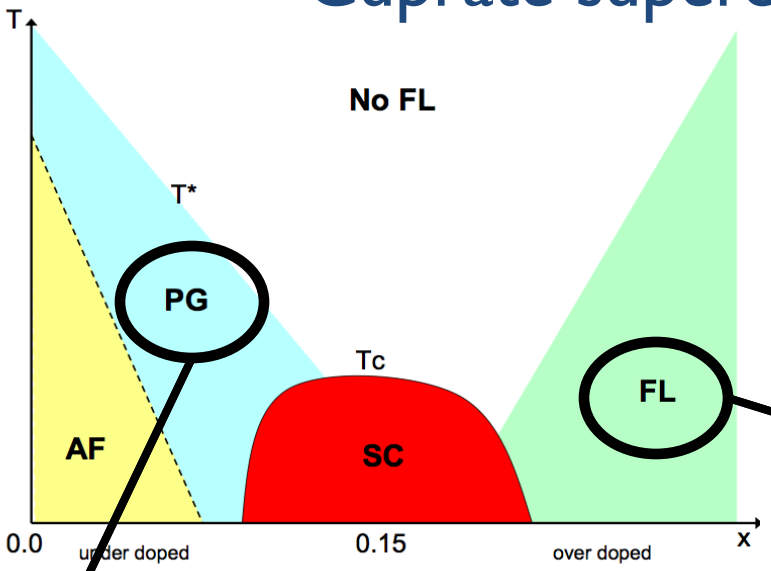
$$\begin{aligned}\mathcal{H} &= \sum_{k\sigma} \epsilon_k c^\dagger_{k\sigma} c_{k\sigma} + \sum_i U n_{i\uparrow} n_{i\downarrow} \\ &+ \lambda \sum_i (b^\dagger_i + b_i)(n_i - 1) + \omega_0 \sum_i b^\dagger_i b_i\end{aligned}$$

- Kondo lattice

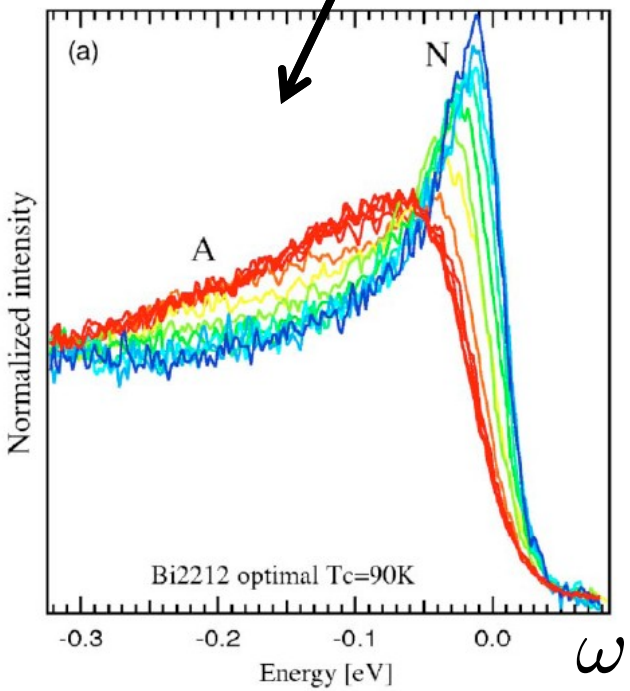
$$\mathcal{H} = \sum_{k\sigma} \epsilon_k c^\dagger_{k\sigma} c_{k\sigma} + J \sum_i \vec{S}_i \cdot \vec{\sigma}_i$$

- Extensions of DMFT to include spatial correlations
  - Cluster extensions that ask for multi-impurity models
  - Dual-fermion approaches that require for higher correlators

Cuprate superconductors

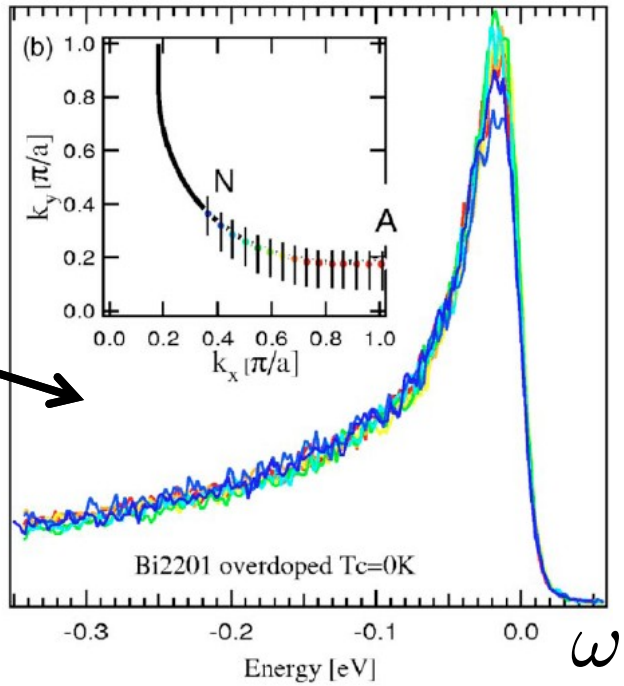


$A(k_i, \omega)$

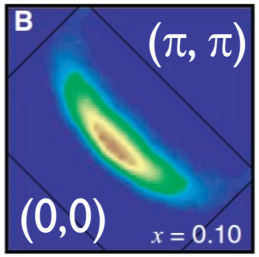


Kaminski et al., PRB (2005)

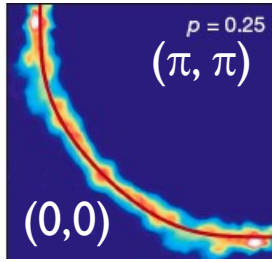
$A(k_i, \omega)$ ARPES



$A(k, \omega = 0)$

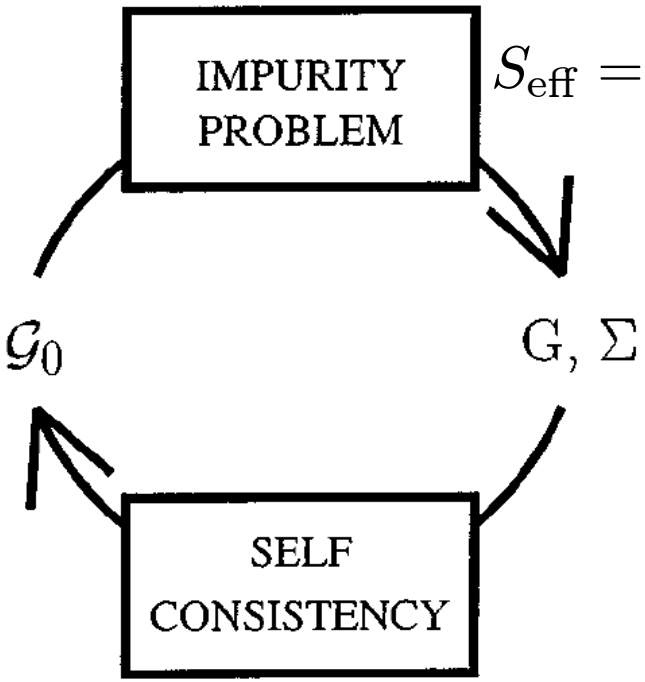


Spectral properties are isotropic along the Fermi surface



Fermi arcs ⇒ Strong variations over the Brillouin zone!

k-space variations are a problem for single-site DMFT



$$S_{\text{eff}} = -\int_0^\beta c_\sigma^\dagger(\tau) G_0^{-1}(\tau - \tau') c_\sigma(\tau') + \int_0^\beta U n_\uparrow(\tau) n_\downarrow(\tau)$$

$$\Sigma(i\omega_n) = G_0^{-1}(i\omega_n) - G^{-1}(i\omega_n)$$

$$\Sigma_{\text{lattice}}(i\omega_n, k) \leftarrow \Sigma(i\omega_n)$$

$$G(i\omega_n) = \sum_k G_{\text{lattice}}(i\omega_n, k)$$

$$G_0^{-1}(i\omega_n) = \left(\sum_k \frac{1}{i\omega_n + \mu - \epsilon_k - \Sigma(i\omega_n)} \right)^{-1} + \Sigma(i\omega_n)$$

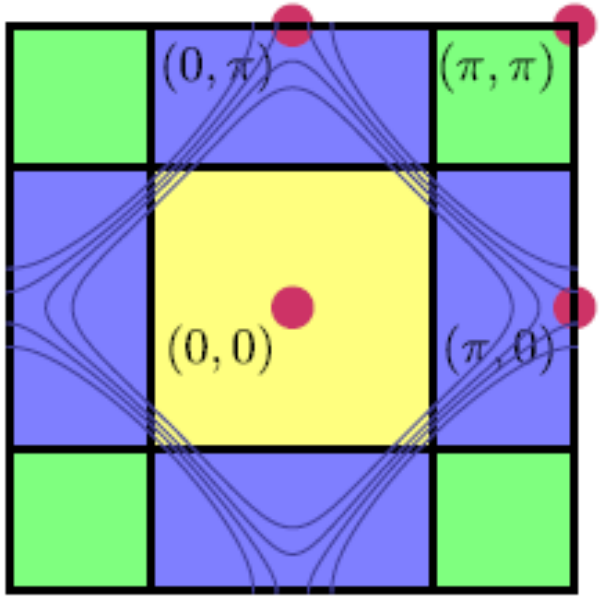
In DMFT the self-energy of the lattice is local: Z, m*, coherence temperature, lifetimes are constant along the Fermi surface

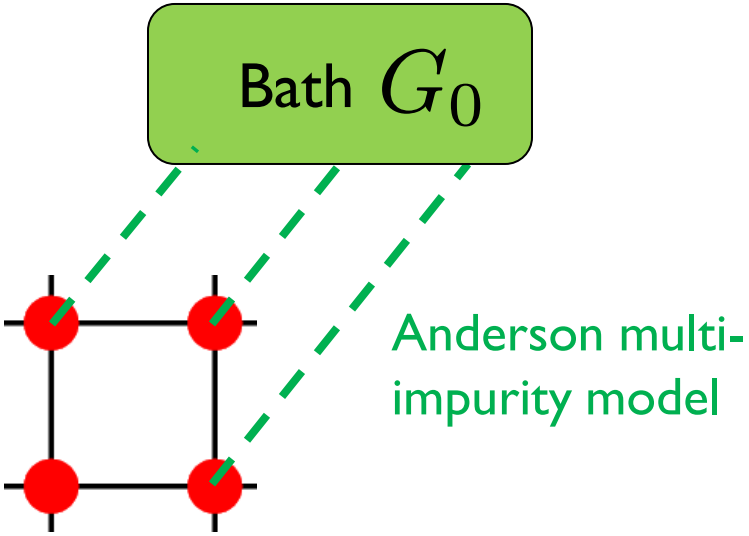
Including a k-dependence

Dynamical cluster approximation (DCA)

M. H. Hettler et al., PRB (1998)

T. Maier et al., RMP (2005)





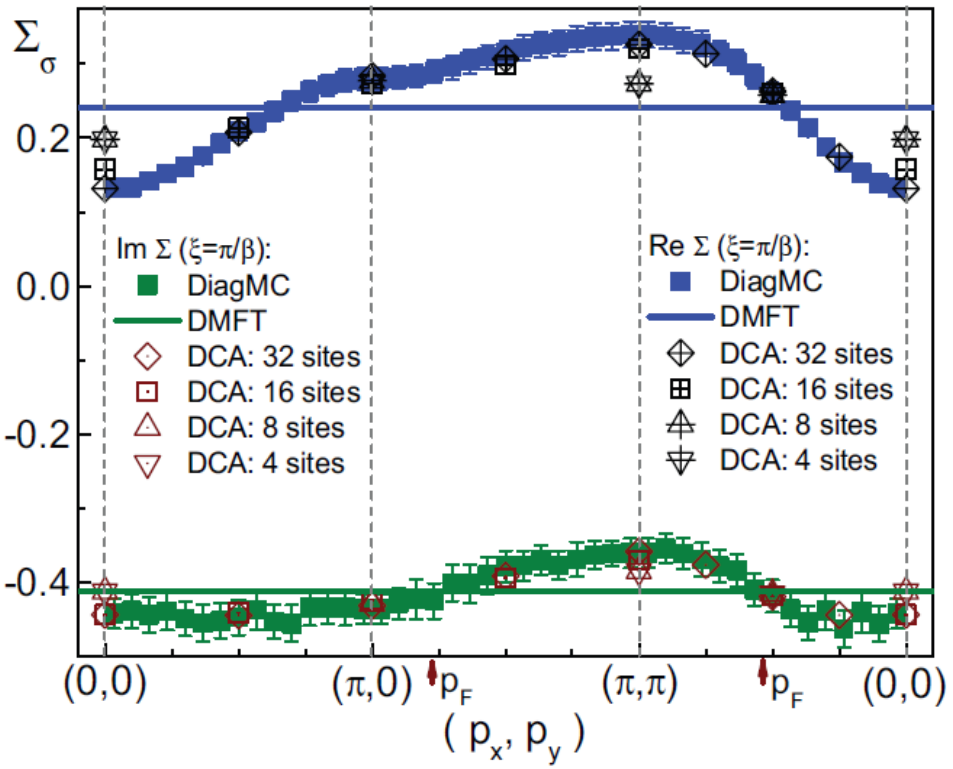
Cluster self-energies for every cluster momenta: $\Sigma(K)$

There are ways to improve DMFT, but then you need to solve a multi-impurity Anderson model!

k-dependence of the self-energy

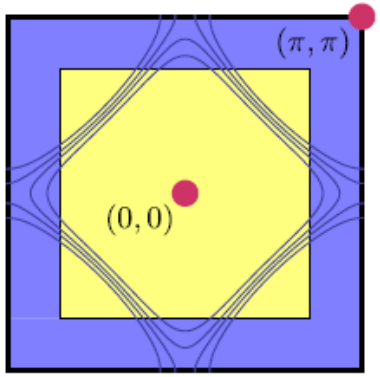
Kozik et al., EPL (2010)

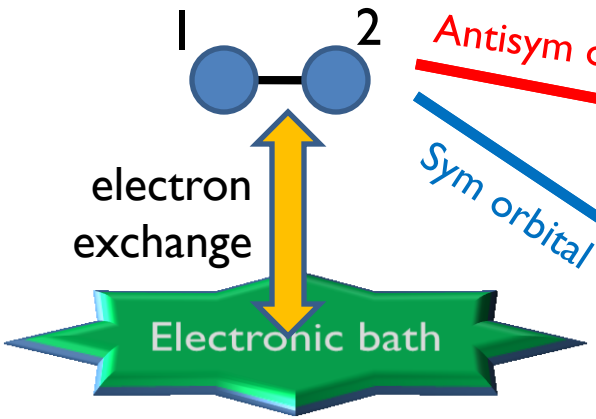
- The DCA approach has been tested against a direct Monte Carlo evaluation of the Hubbard lattice problem
- The solution scales nicely with the number of sites and has converged for 32 sites



Back to cuprates with a 2-site extension

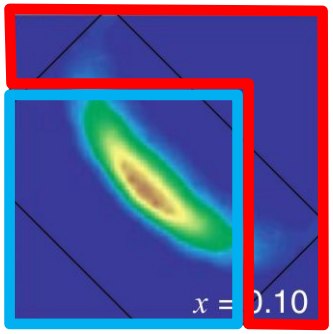
M. Ferrero et al., PRB (2009)

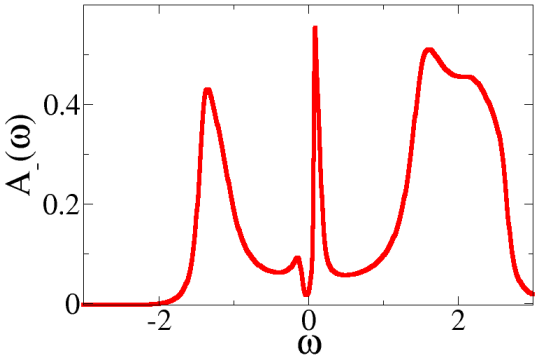




Antisym orbital 1-2

Sym orbital 1+2





Ordinary Fermi liquid metal

pseudogap in antinodal orbital

metallic, protected nodal orbital

0%

≈16%

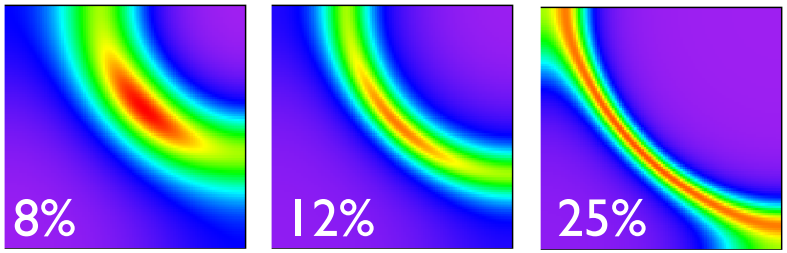
Hole doping

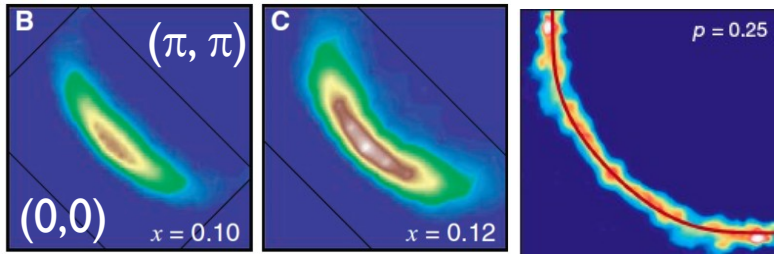
Singlet state dominates

Selective Mott transition

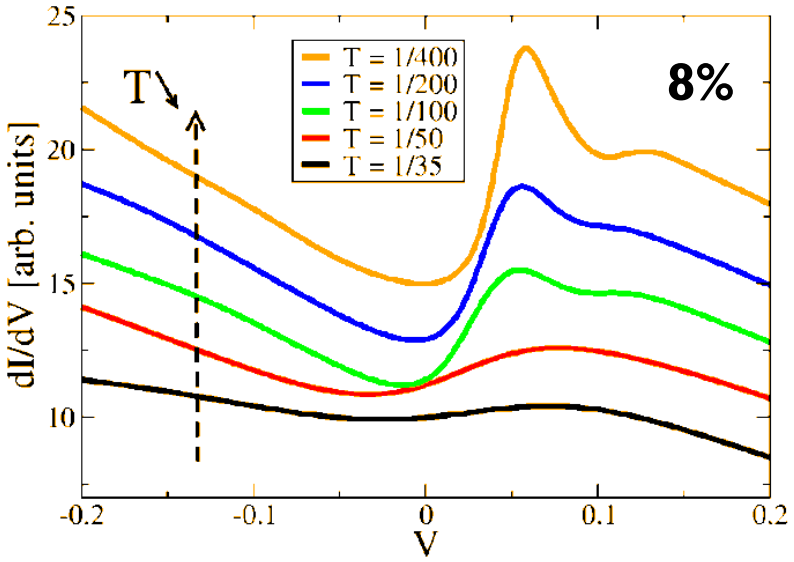
Can one describe the phenomenology of the cuprates?

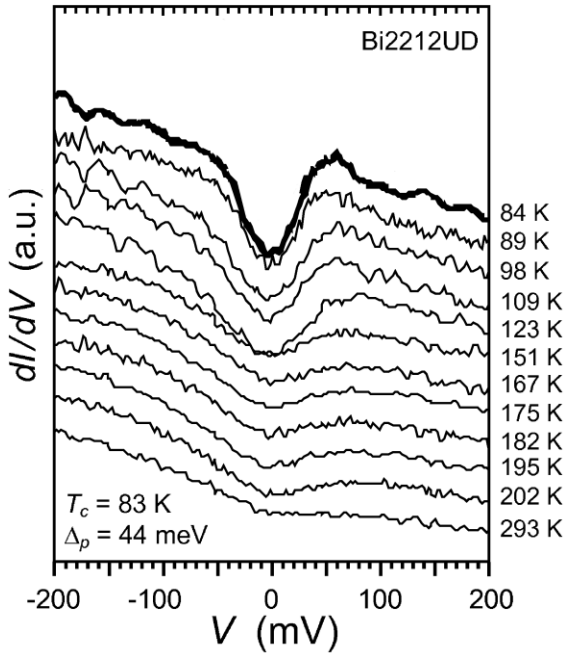
- ARPES intensity maps





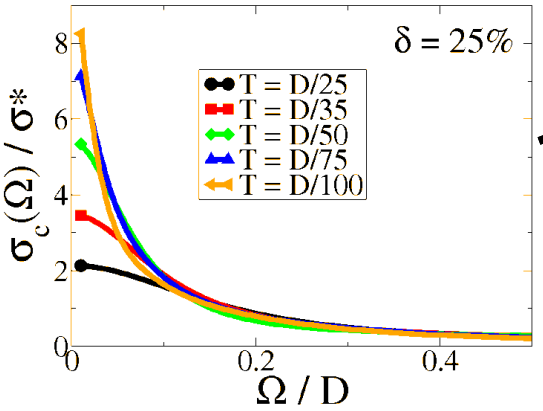
- Tunneling conductance (STM)

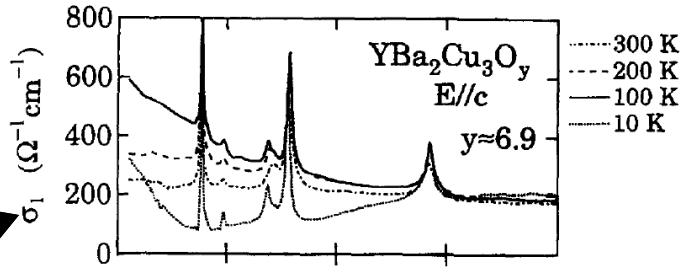


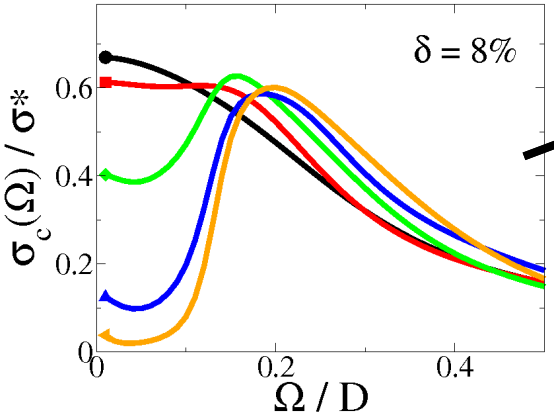


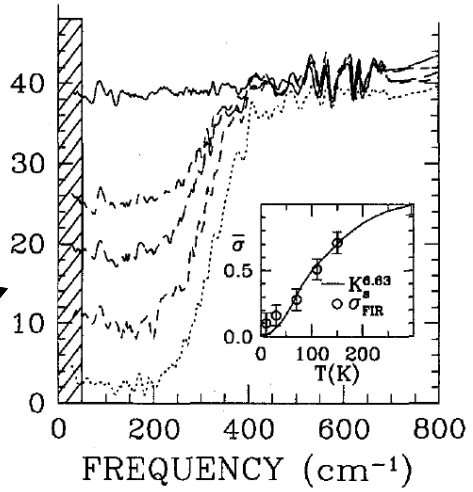
Interplane charge dynamics

- Optical conductivity

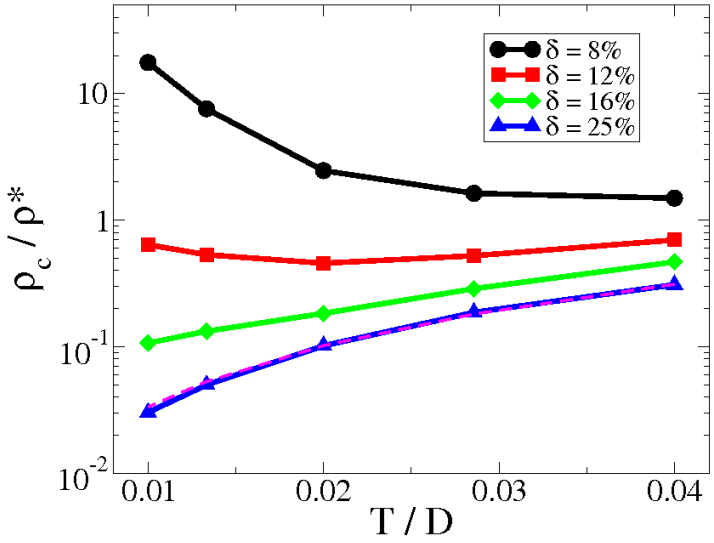


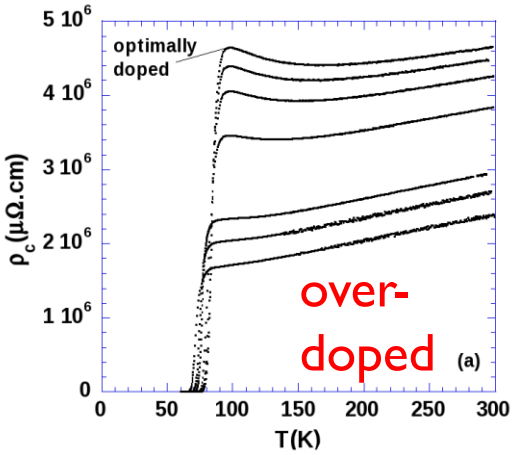


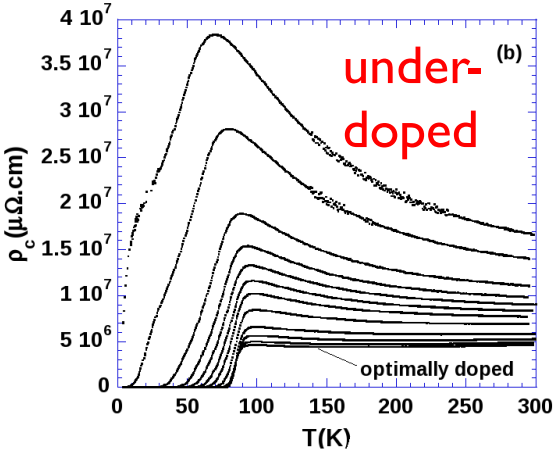




- Resistivity



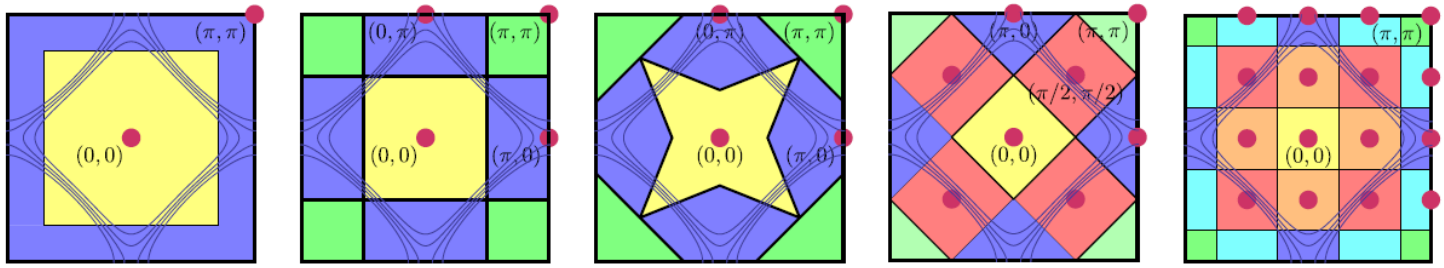




Confirmation of the results on larger clusters

E. Gull et al., PRB (2010)

- Computations on clusters up to 16 sites have been performed

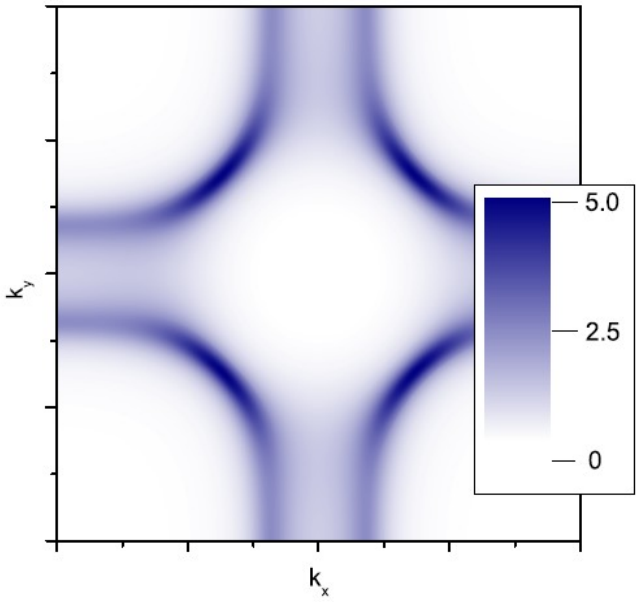


- Results show that the main features of the above results are robust
- The approach to the Mott insulator is characterized by successive Mott transitions in reciprocal space
- The regions close to the antinode get insulating first, while regions close to the node remain metallic down to the Mott insulating state

Dual-fermion calculations for the Hubbard model

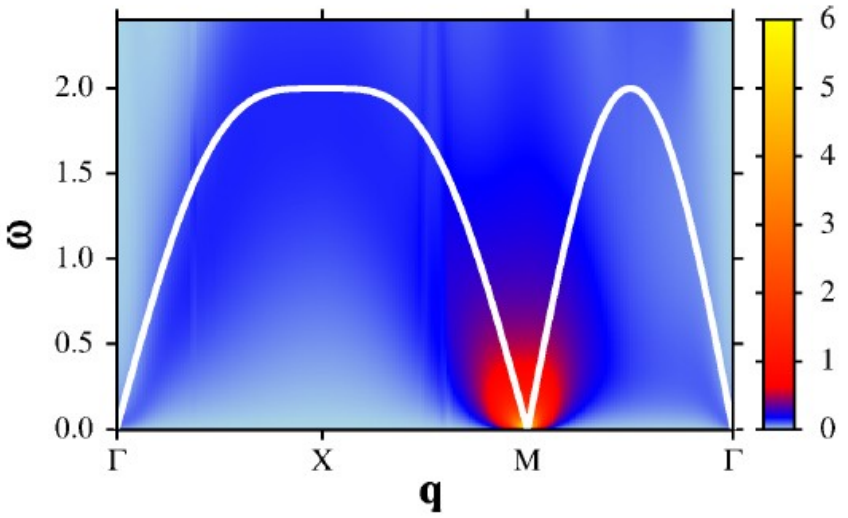
- Dual-fermion approaches are another way to include spatial correlations. They require the knowledge of 4-point vertex functions
- These methods have benefitted a lot from the new CT-QMC

Spectral intensity map of the Hubbard model at 14% doping



Rubtsov et al., PRB, (2009)

Dynamical susceptibility at half-filling with a visible magnon spectrum in the paramagnetic state

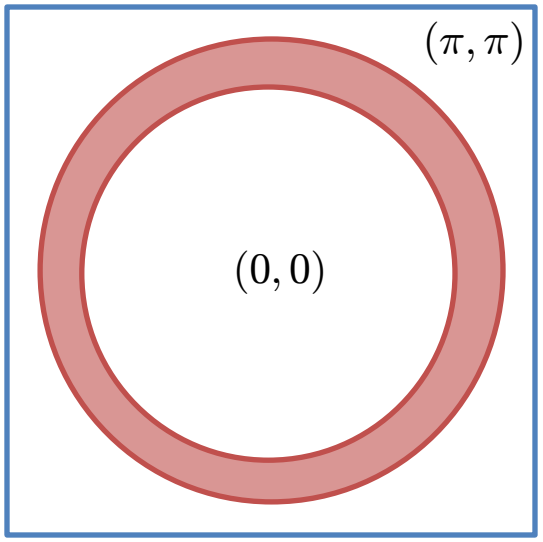


Hafermann et al. , PRL, (2009)

Stabilization of a Sarma phase in cold-atomic systems

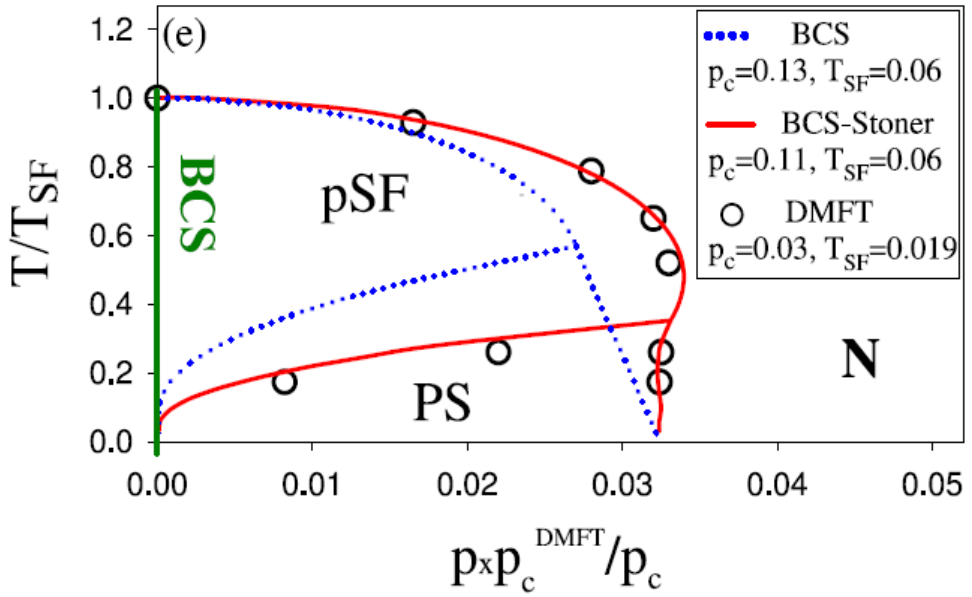
Dao et al., PRL (2008)

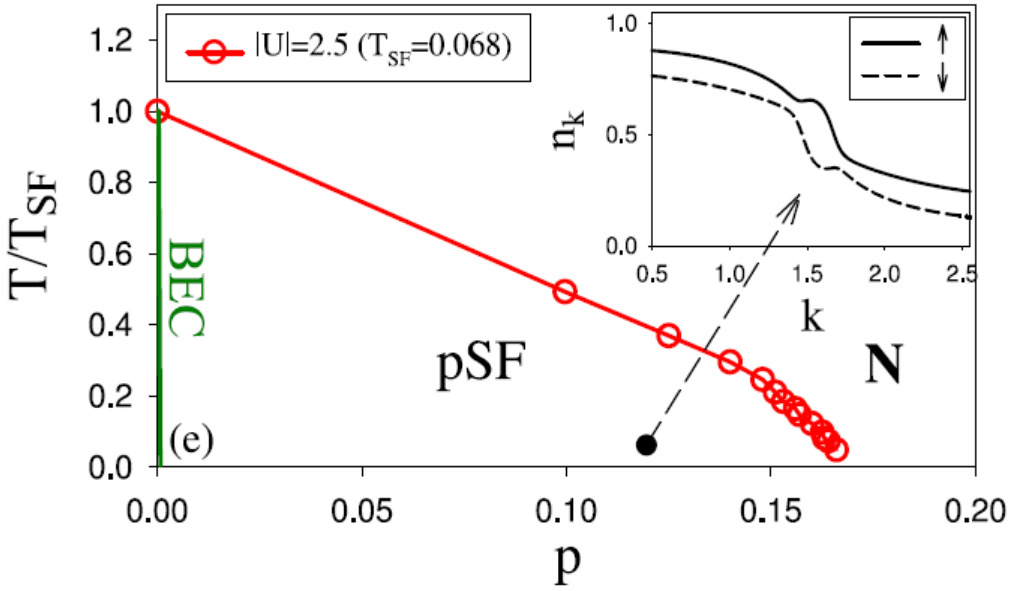
- Cold-atomic systems allow to study superfluid phases with great precision
- The situation is rather clear when there is an attraction between two balanced populations (BCS – BEC crossover)
- But what happens as we introduce a population imbalance and the Fermi surface mismatch?
- Could we obtain a Sarma phase which is both superfluid and polarized?
- Sarma has shown that it is not stable at weak coupling
- But what about strong coupling?



Sarma phase cont'd

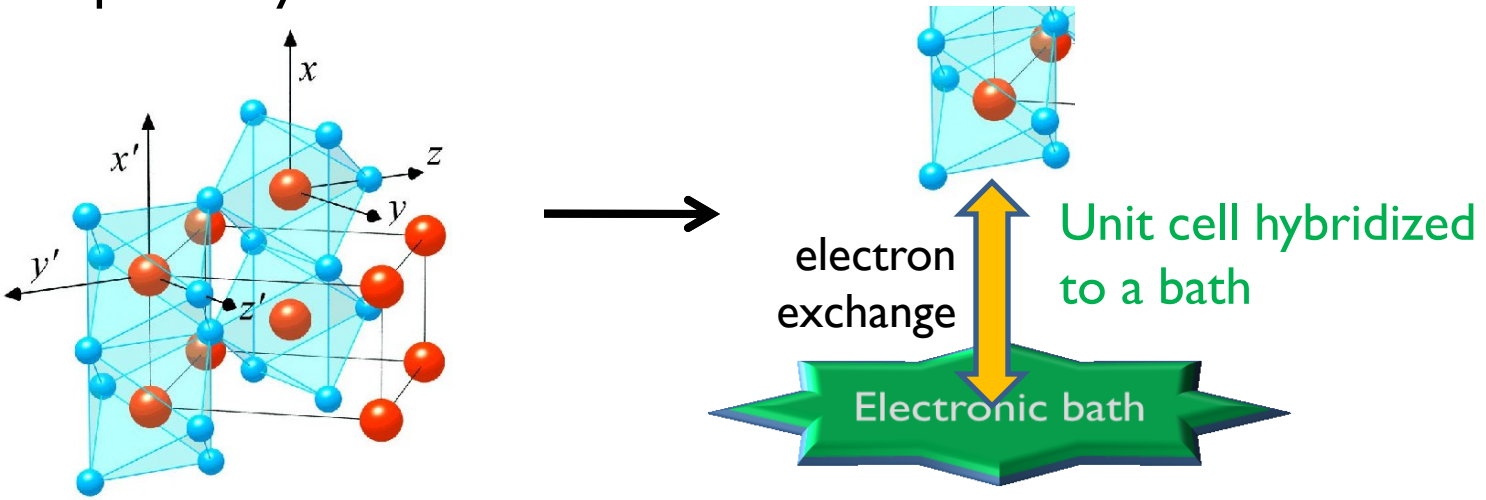
- A study of a negative-U Hubbard model shows that:
- At weak coupling a phase separated state is stabilized at low temperature
- At larger coupling a polarized superfluid is stabilized down to very low temperatures
- The presence of two Fermi surfaces could be observed in cold-atomic experiments





Electronic structure calculations

- There has been a large effort to treat fully many-body effects within density-functional-based (DFT) electronic structure calculations
- One of the approaches is DFT+DMFT where the many-body effects are computed by DMFT

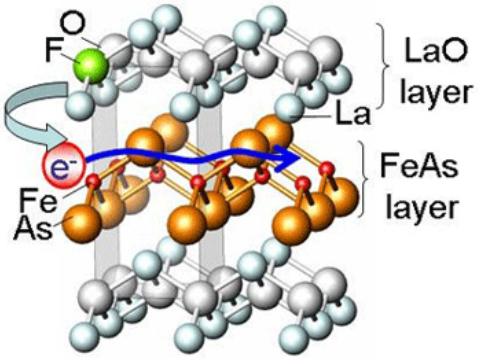


- Because realistic systems often involve many bands and because it is sometimes important to consider a full SU(2) Hamiltonian, new solvers have been extremely useful

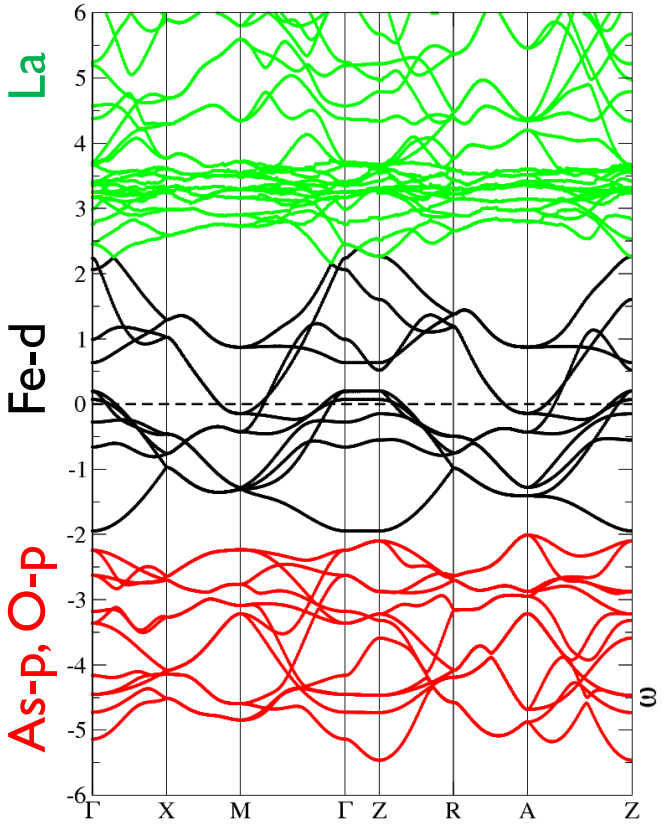
Correlations in pnictides superconductors

M. Aichhorn et al., PRB (2009)

LaFeAsO

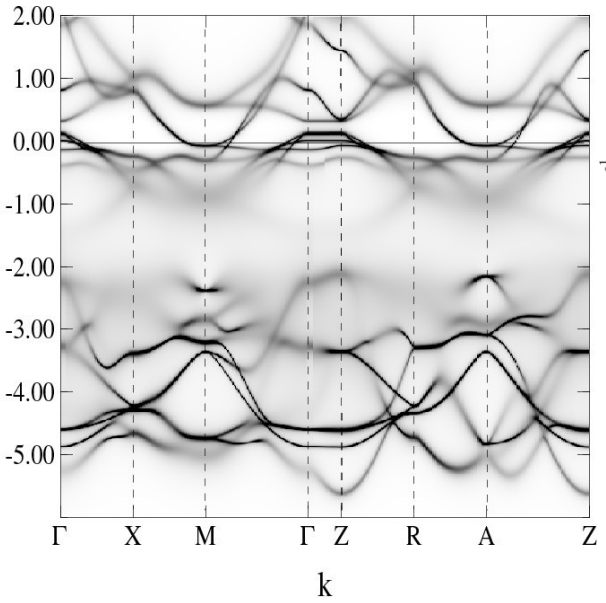


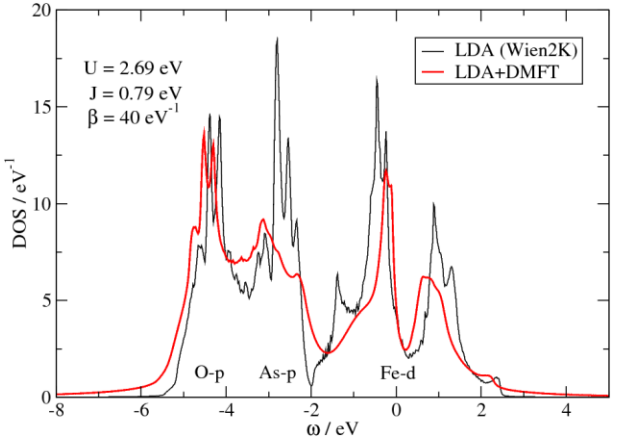
- moderate correlations
- $m^* \approx 1.6$
- lifetime effects
- shifted bands



LDA bands

$A(k,\omega)$





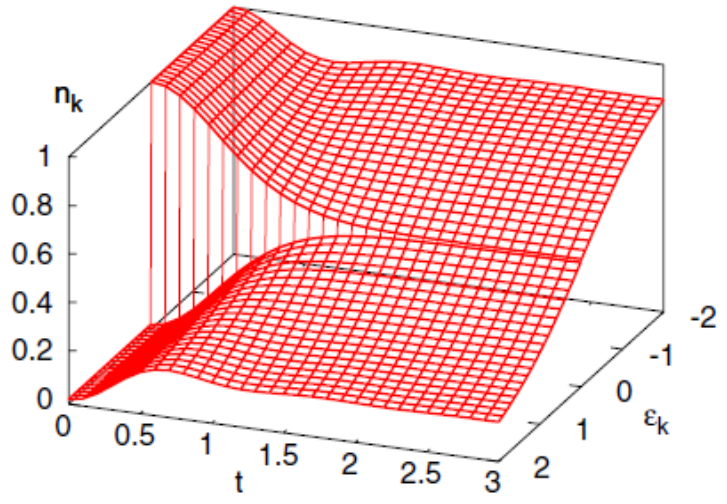
Density of states

Correlation effects seen by LDA+DMFT

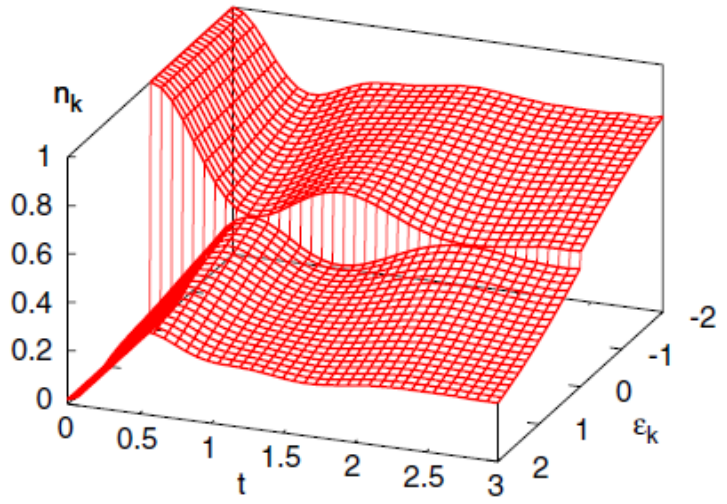
Interaction quench in the Hubbard model

Eckstein et al., PRB (2010)

- What happens it we suddenly turn on an interaction in the Hubbard model? Does it reach a steady state? Is this steady state a thermal state?
- A real-time CT-QMC algorithm has been used to study this problem within non-equilibrium DMFT



$0 < U < 3.2V$



$3.2V < U$

Summary

- Continuous-time quantum Monte Carlo algorithms have allowed for progress in computing the properties of strongly-correlated systems
  - Lower temperatures
  - Generic Hamiltonians, new models
  - Larger number of orbitals / sites
- The idea of the algorithms is to sample stochastically the diagrams of a series expansion of the partition function
- According to one's need, different expansions can be used
- There are still limitations (sign problem, speed, …) and more work has to be done!

Outline

- Quantum impurity problems
- Continuous-time quantum Monte Carlo (CT-QMC) methods
- Generic introduction to Monte Carlo
- The interaction-expansion algorithm (CT-INT)
- The hybridization-expansion algorithm (CT-HYB)
- Some applications of CT-QMC solvers
- TRIQS: a Toolbox for Research on Quantum Interacting Systems
- Implementation details for the CT-INT

A need for tools to do many-body calculations

- Solving a quantum impurity problem is usually just one step of a larger calculation
- For example, a typical DMFT calculation might require to
  - Manipulate Green's functions
  - Solve a quantum impurity problem
  - Sum quantities over the Brillouin zone of a lattice
  - Analyze, archive, plot data
- One might also have to use codes from different origins and "glue" results together
- It is therefore useful to have a set of flexible tools to work with the objects of a many-body calculations

Illustration: a DFT+DMFT calculation

Converged?

$$\rho_{\mathrm{LDA+DMFT}} \rightarrow V_{\mathrm{KS}}$$

DFT Step

$$V_{\mathrm{KS}} \rightarrow \epsilon_{k\nu}, \Psi_{k\nu}(r)$$

Wien2k

Wien2k

New density

$$N^k_{\nu\nu'} = \sum G_{\nu\nu'}(k, i\omega_n) e^{i\omega_n 0^+}$$

$$\rho_{\mathrm{DMFT}}(r) = \sum_{k,\nu\nu'}^{n} \langle r|\Psi_{k\nu}\rangle N^k_{\nu\nu'} \langle \Psi_{k\nu'}|r\rangle$$

dmftproj

Projectors

$$P_{m\nu}(k) = \langle \omega_{km}|\Psi_{k\nu}\rangle$$

$$\mathcal{G}^0_{mm'} = \left(\Sigma_{mm'} + G^{\mathrm{loc}^{-1}}_{mm'}\right)^{-1}$$

Green's functions manipulations

Lattice Green's function

$$G^{-1}_{\nu\nu'}(k, i\omega_n) = (i\omega_n + \mu - \epsilon_{k\nu})\delta_{\nu\nu'} - \Sigma_{\nu\nu'}(k, i\omega_n)$$

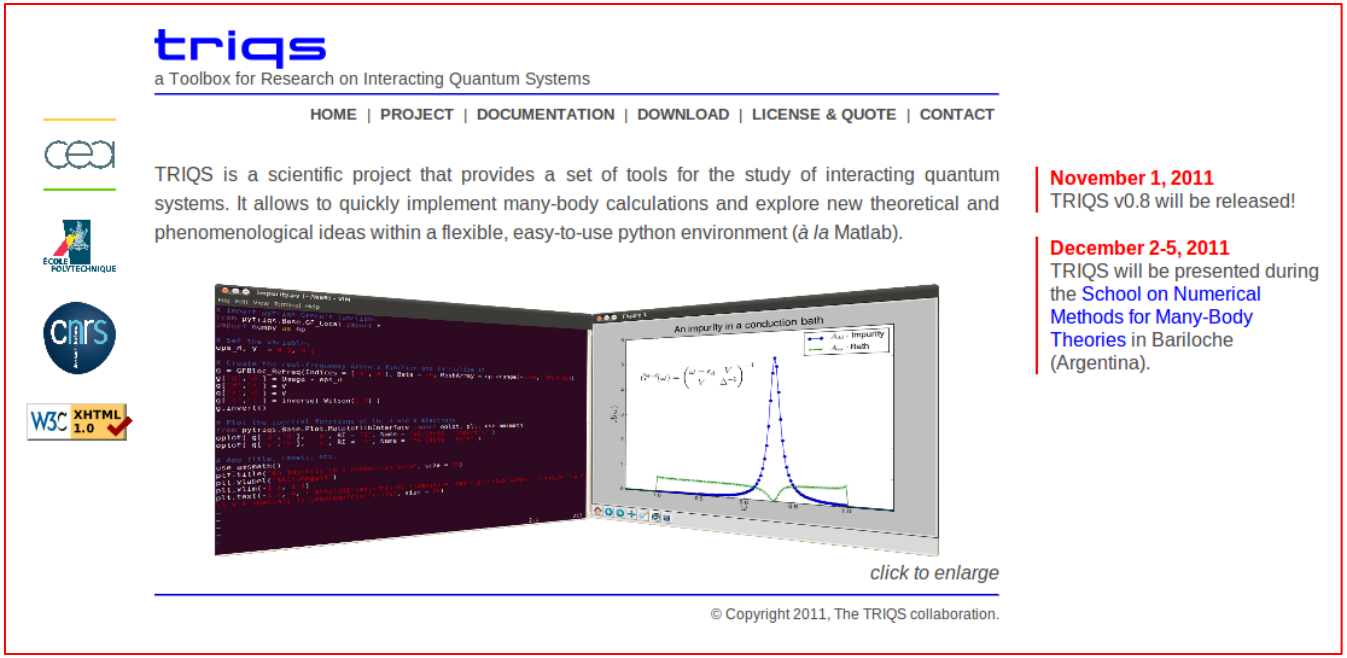
$$G^{\mathrm{loc}}_{mm'}(i\omega_n) = \sum_{k\nu\nu'} P_{m\nu}(k) G_{\nu\nu'}(k, i\omega_n) P^*_{\nu' m'}(k)$$

Anderson Impurity Model

$$\mathcal{G}^0_{mm'}(i\omega_n) \rightarrow \Sigma_{mm'}(i\omega_n)$$

CT-QMC

Toolbox for Research on Interacting Quantum Systems



http://ipht.cea.fr/triqs

- Was born from the need to have a versatile environment to study interacting quantum systems in different contexts
- Started at Ecole Polytechnique and IPhT Saclay (with O. Parcollet, M. Aichhorn, L. Pourovskii, V. Vildosola, C. Martins, L. Boehnke, …)
- It is an open-source (GPL license) library

What is in the package?

triqs

http://ipht.cea.fr/triqs

Python libraries:

Lattice tools

Data analysis tools

Plotting interface

Green's functions

Quantum impurity solvers

Ready-to-use applications:

Wien2TRIQS

C++ libraries for developers:

Monte Carlo

Array library

Determinant manipulations

Example: Bethe lattice DMFT

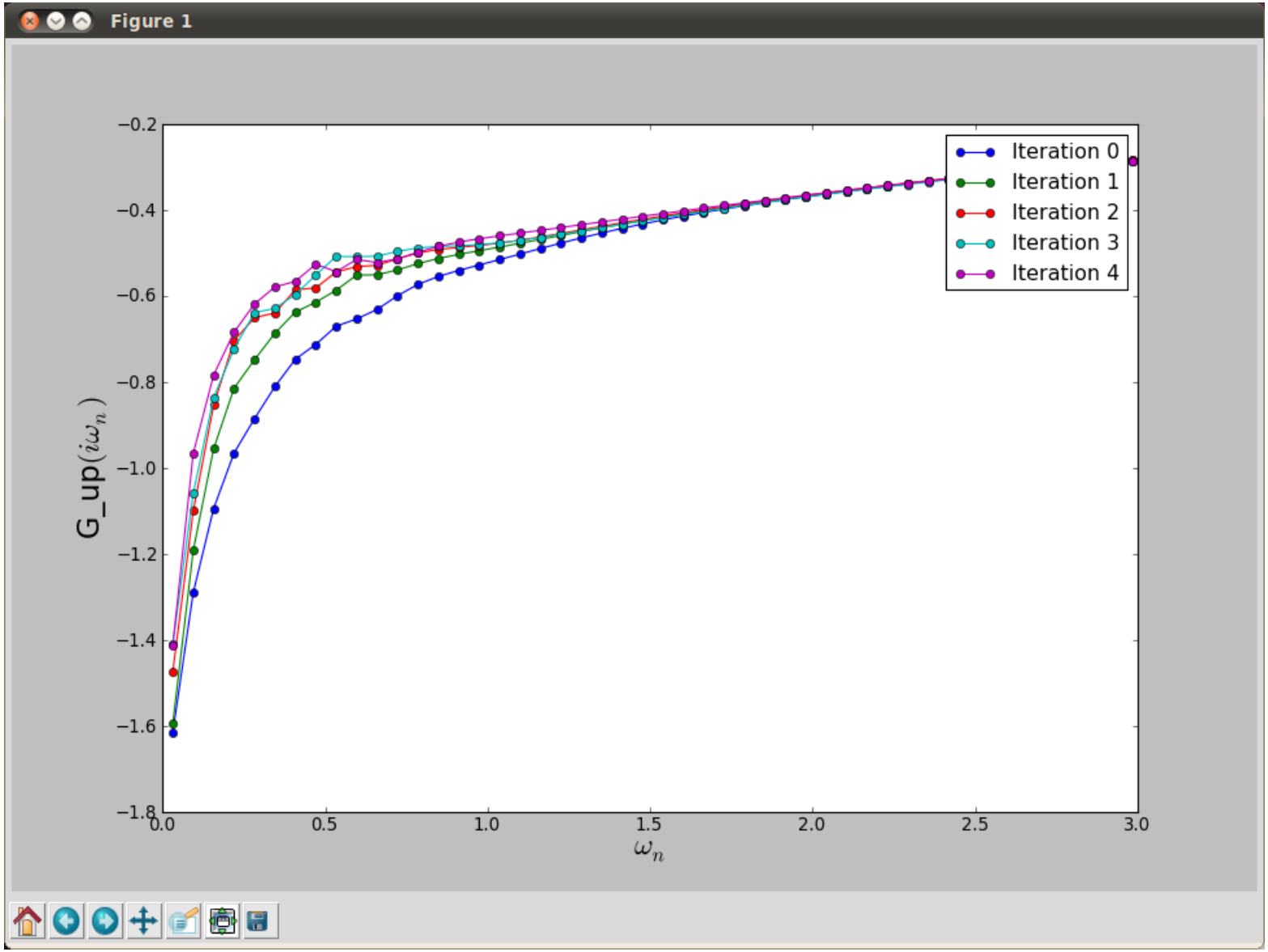
mferrero@mferrero-laptop: ~/TEST

File Edit View Terminal Help

```
from pytriqs.Base.GF_Local import *
from pytriqs.Base.Archive import HDF_Archive
from pytriqs.Base.Plot.MatplotlibInterface import oplot

# Set up a few parameters
Half_Bandwidth = 1.0
U = 2.5
Chemical_Potential = U/2.0
Beta = 100
N_loops = 5

# Construct a CTQMC solver
from pytriqs.Solvers.Operators import *
from pytriqs.Solvers.HybridizationExpansion import Solver
S = Solver(Beta = Beta,
           GFstruct = [ ('up',[1]), ('down',[1]) ],
           H_Local = U * N('up',1) * N('down',1),
           Quantum_Numbers = { 'Nup' : N('up',1), 'Ndown' : N('down',1) },
           N_Cycles = 5000,
           N_Warmup_Cycles = 1000,
           N_Legendre_Coeffs = 30,
           Use_Segment_Picture = True)

# Initalize the Green's function to a semi circular
S.G <<= SemiCircular(Half_Bandwidth)

# Now do the DMFT loop
for IterationNumber in range(N_loops):

    # Compute S.G0 with the self-consistency condition while imposing paramagnetism
    g = 0.5 * ( S.G['up'] + S.G['down'] )
    for name, g0block in S.G0:
        g0block <<= inverse( iOmega_n + Chemical_Potential - (Half_Bandwidth/2.0)**2  * g )

    # Run the solver
    S.Solve()

    # Some intermediate saves
    oplot(S.G['up'].imag, '-o', x_window=(0,3), Name="Iteration %i"%IterationNumber)
```

1,1 All

Example: Bethe lattice DMFT



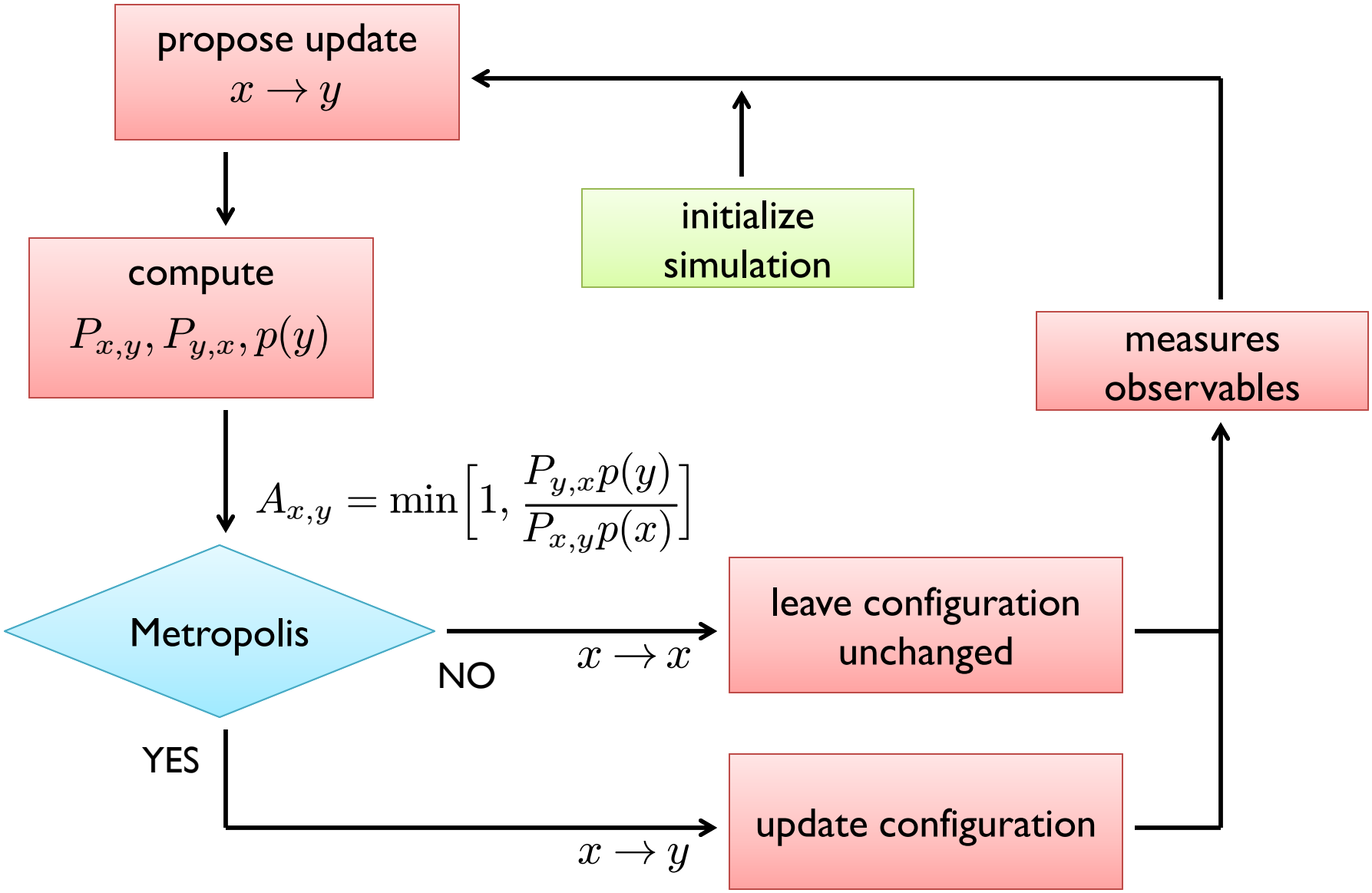
Making codes open source

- Probably the best way to learn about Monte Carlo methods is to implement one's own algorithm
- In the hands-on we try to code an interaction-expansion continuous-time algorithm
- Note however that in general it is not always a good idea to reinvent the wheel!
- There are good open-source tools around and one should never hesitate to use them
- Making one's codes available for the rest of the scientific community is a very good idea!

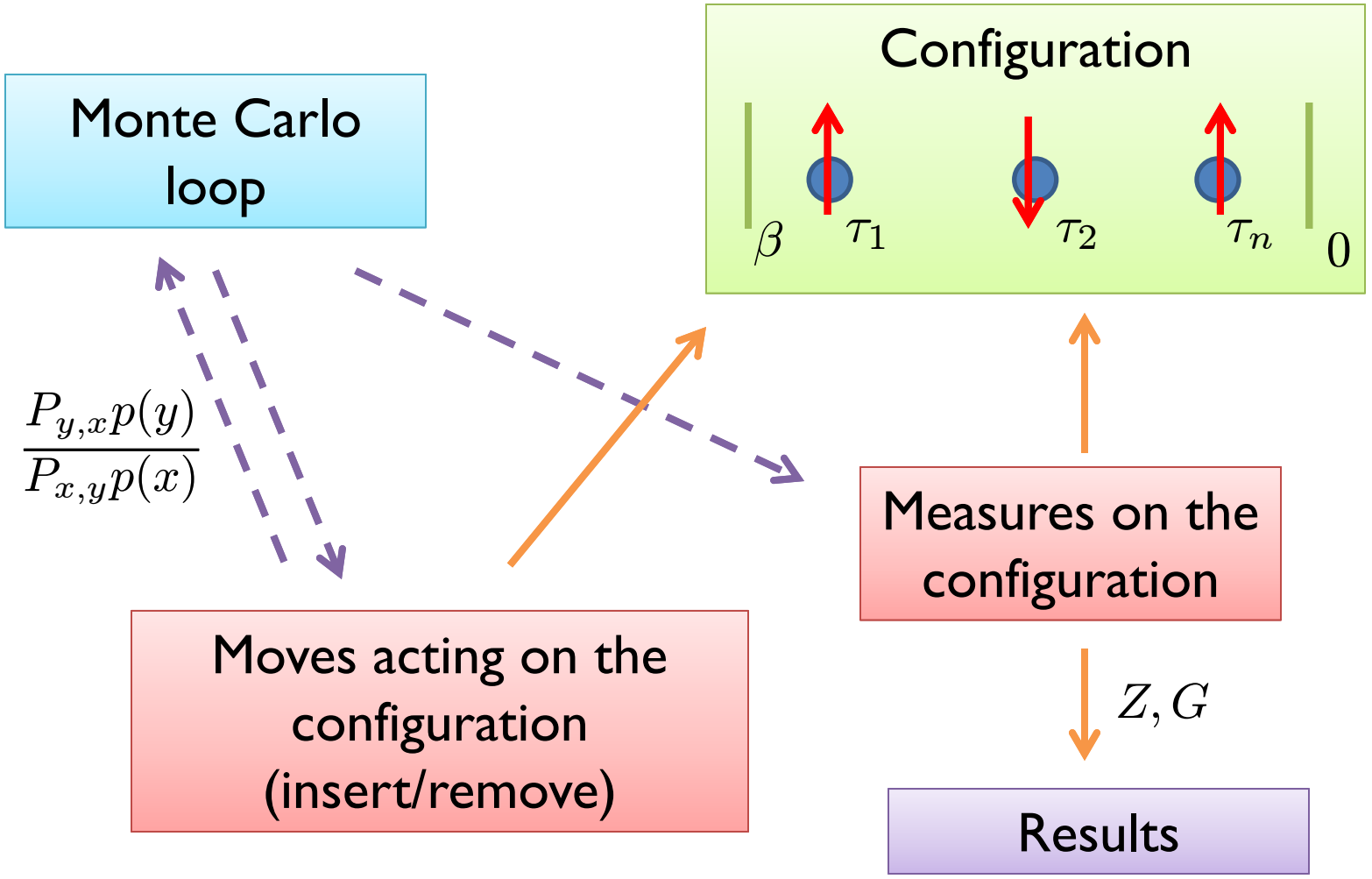
Outline

- Quantum impurity problems
- Continuous-time quantum Monte Carlo (CT-QMC) methods
- Generic introduction to Monte Carlo
- The interaction-expansion algorithm (CT-INT)
- The hybridization-expansion algorithm (CT-HYB)
- Some applications of CT-QMC solvers
- TRIQS: a Toolbox for Research on Quantum Interacting Systems
- Implementation details for the CT-INT

Reminder: A Monte Carlo algorithm

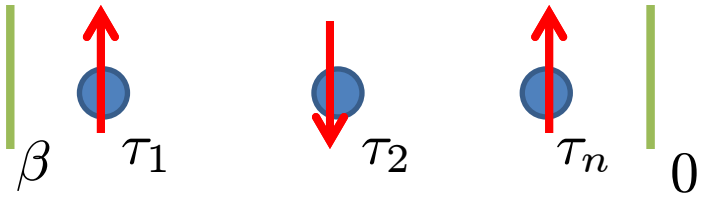


Implementing the CT-INT algorithm in practice



Implementation cont'd

- Good practice to try to separate the algorithm in pieces
- The Monte Carlo loop does not need to know about the system. It just calls moves and expects back a Metropolis ratio to decide if the move is accepted or rejected
- What we need to do is implement insertions and removals in the configuration



and compute the ratio

$$\frac{P_{y,x}p(y)}{P_{x,y}p(x)}$$

- This ratio involves a determinant! This is the tricky part!

Reminder

- The weight of a Monte Carlo configuration is

$$p(\mathcal{C}_n) = \left| \left(\frac{-U}{2} \right)^n \det D_n^{\uparrow} \det D_n^{\downarrow} \right|$$

- And the matrix in the determinant has this definition

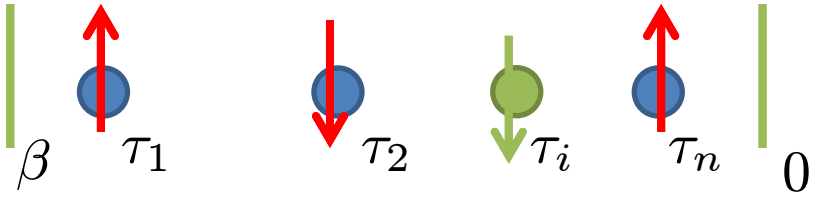
$$\det D_n^{\sigma} = \begin{vmatrix} G_{0\sigma}(0^-) - \alpha_{s_1\sigma} & G_{0\sigma}(\tau_1 - \tau_2) & \cdots & \cdots \\ G_{0\sigma}(\tau_2 - \tau_1) & G_{0\sigma}(0^-) - \alpha_{s_2\sigma} & G_{0\sigma}(\tau_2 - \tau_3) & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & G_{0\sigma}(\tau_n - \tau_{n-1}) & G_{0\sigma}(0^-) - \alpha_{s_n\sigma} \end{vmatrix}$$

with

$$\alpha_{s\sigma} = \frac{1}{2} + (2\delta_{s\sigma} - 1)\delta$$

Inserting a new vertex

- Inserting a vertex adds a line and column to the matrix



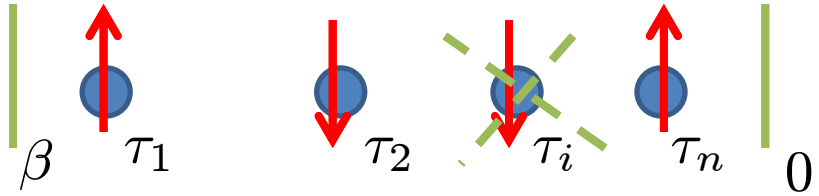
$$A_{x,y} = \min\left[1, \frac{-U\beta}{n+1} \times \frac{\det D^{\uparrow}_{n+1} D^{\downarrow}_{n+1}}{\det D^{\uparrow}_{n} D^{\downarrow}_{n}}\right]$$

$$\det D^{\sigma}_{n} = \begin{vmatrix} G_{0\sigma}(0^-) - \alpha_{s_1\sigma} & G_{0\sigma}(\tau_1 - \tau_2) & G_{0\sigma}(\tau_1 - \tau_3) \\ G_{0\sigma}(\tau_2 - \tau_1) & G_{0\sigma}(0^-) - \alpha_{s_2\sigma} & G_{0\sigma}(\tau_2 - \tau_3) \\ G_{0\sigma}(\tau_3 - \tau_1) & G_{0\sigma}(\tau_3 - \tau_2) & G_{0\sigma}(0^-) - \alpha_{s_3\sigma} \end{vmatrix}$$

$$\det D^{\sigma}_{n+1} = \begin{vmatrix} G_{0\sigma}(0^-) - \alpha_{s_1\sigma} & G_{0\sigma}(\tau_1 - \tau_2) & G_{0\sigma}(\tau_1 - \tau_3) & G_{0\sigma}(\tau_1 - \tau_4) \\ G_{0\sigma}(\tau_2 - \tau_1) & G_{0\sigma}(0^-) - \alpha_{s_2\sigma} & G_{0\sigma}(\tau_2 - \tau_3) & G_{0\sigma}(\tau_2 - \tau_4) \\ G_{0\sigma}(\tau_3 - \tau_1) & G_{0\sigma}(\tau_3 - \tau_2) & G_{0\sigma}(0^-) - \alpha_{s_3\sigma} & G_{0\sigma}(\tau_3 - \tau_4) \\ G_{0\sigma}(\tau_4 - \tau_1) & G_{0\sigma}(\tau_4 - \tau_2) & G_{0\sigma}(\tau_4 - \tau_3) & G_{0\sigma}(0^-) - \alpha_{s_4\sigma} \end{vmatrix}$$

Removing a vertex

- Removing a vertex removes a line and column in the matrix



$$\begin{vmatrix} G_{0\sigma}(0^-) - \alpha_{s_1\sigma} & G_{0\sigma}(\tau_1 - \tau_2) & G_{0\sigma}(\tau_1 - \tau_3) & G_{0\sigma}(\tau_1 - \tau_4) \\ G_{0\sigma}(\tau_2 - \tau_1) & G_{0\sigma}(0^-) - \alpha_{s_2\sigma} & G_{0\sigma}(\tau_2 - \tau_3) & G_{0\sigma}(\tau_2 - \tau_4) \\ G_{0\sigma}(\tau_3 - \tau_1) & G_{0\sigma}(\tau_3 - \tau_2) & G_{0\sigma}(0^-) - \alpha_{s_3\sigma} & G_{0\sigma}(\tau_3 - \tau_4) \\ G_{0\sigma}(\tau_4 - \tau_1) & G_{0\sigma}(\tau_4 - \tau_2) & G_{0\sigma}(\tau_4 - \tau_3) & G_{0\sigma}(0^-) - \alpha_{s_4\sigma} \end{vmatrix}$$

$$A_{x,y} = \min\left[1, \frac{-n}{U\beta} \times \frac{\det D^{\uparrow}_{n-1} D^{\downarrow}_{n-1}}{\det D^{\uparrow}_{n} D^{\downarrow}_{n}}\right]$$

No need to recompute a full determinant

- We see that we need to compute ratios of determinant of matrices that have an added / removed line and column
- This can be done faster that by computing the full determinant of the matrix before and after the move
- The trick is to keep a copy of both the matrix and its inverse and to update them
- Indeed:

$$D^{-1}_{ij} = (-1)^{i+j} \frac{1}{\det D} \det D[j, i]$$

Means the matrix with line j and column i removed

Fast updates for determinant and inverse matrix

- This is a special case of the Sherman-Morrison formula

$$\begin{pmatrix} A & B \\ C & d \end{pmatrix} \times \begin{pmatrix} U & V \\ W & x \end{pmatrix} = \begin{pmatrix} AU + BW & AV + xB \\ CU + dW & CV + xd \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- Insertion

$$V = -xA^{-1}B \qquad W = -xCA^{-1}$$

$$x^{-1} = d - CA^{-1}B \qquad U = A^{-1} + x^{-1}VW$$

$$\frac{\det \begin{pmatrix} A & B \\ C & d \end{pmatrix}}{\det (A)} = x^{-1}$$

- Removal

$$A^{-1} = U - x^{-1}VW \qquad \frac{\det (A)}{\det \begin{pmatrix} A & B \\ C & d \end{pmatrix}} = x$$

Fast updates cont'd

- One can find the determinant ratios and the new inverse matrix in operations of order n^2
- Here we have shown insertions and removals happening at the very end of the matrix. If the insertions or removals are somewhere in the middle, one must add a correction $(-1)^{i+j}$
- An important thing to note is that the determinant ratio can be obtained before actually computing the inverse matrix. We can therefore decide to accept or reject the move and update the inverse matrix only if the move is accepted!
- Another good thing about have the inverse matrix is that we need it anyway to compute the Green's function
- These are all the ingredients to write an efficient algorithm!