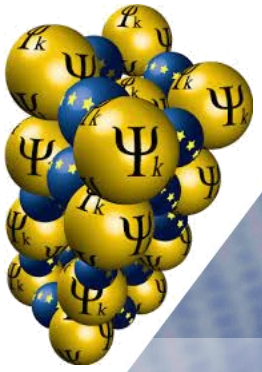
Targeting Real chemical accuracy at the EXascale

SISSA



TREX e-School on Quantum Monte Carlo with TurboRVB

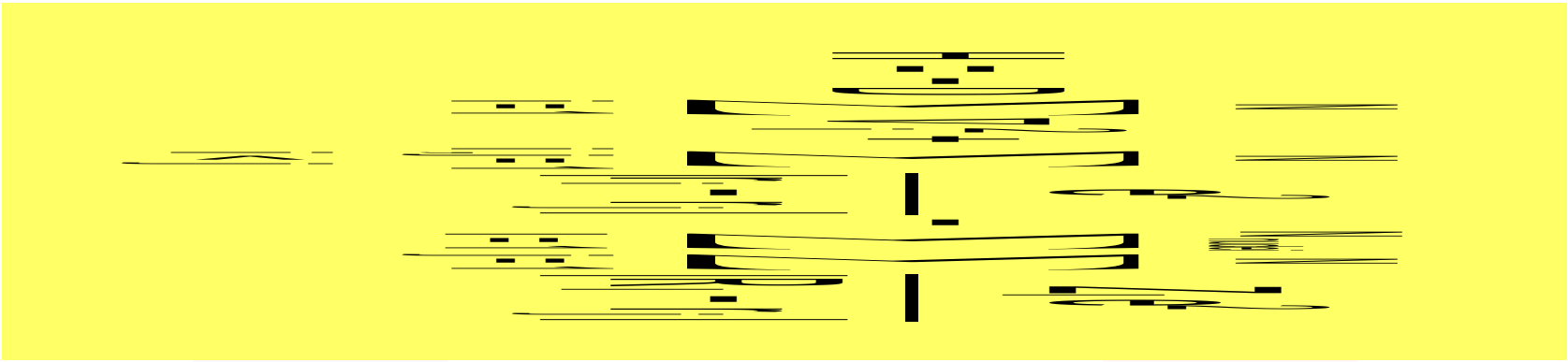
Probability sampling, and variational Monte Carlo

Michele Casula
CNRS and Sorbonne Université, Paris, France

Targeting Real Chemical Accuracy at the Exascale project has received funding from the European Union Horizoon 2020 research and innovation programme under Grant Agreement **No. 952165.**

Outline

- The *ab initio* Hamiltonian
- “Supremacy” of Quantum Monte Carlo
- Stochastic vs deterministic integration
- Sampling a target probability distribution
- Variational quantum Monte Carlo



Coulomb electron-electron, electron-ion interactions
+ quantum kinetic term

$$H\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) = E_{GS}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N)$$

Hard problem to solve,
but there are different (approximated) ways to tackle it…

Possible methods to solve it

- **Density functional theory methods**

Density based methods
Self-consistent solution of an effective mean-field Hamiltonian
Large systems but approximate exchange/correlation
Scaling: $N^2 \log N$ - N^3

- **Post-Hartree-Fock methods (MCSCF, CC, CI,..)**

Wavefunction based methods (Gaussian single-particle basis set)
Expansion in many determinants with slow convergence
Very accurate on small systems
Scaling: N^4 to exponential

- **Quantum Monte Carlo techniques**

Wavefunction based methods (explicitly correlated wave function)
Stochastic solution of the Schrödinger equation
Most accurate benchmarks for medium-large systems
Scaling: N^3-N^4

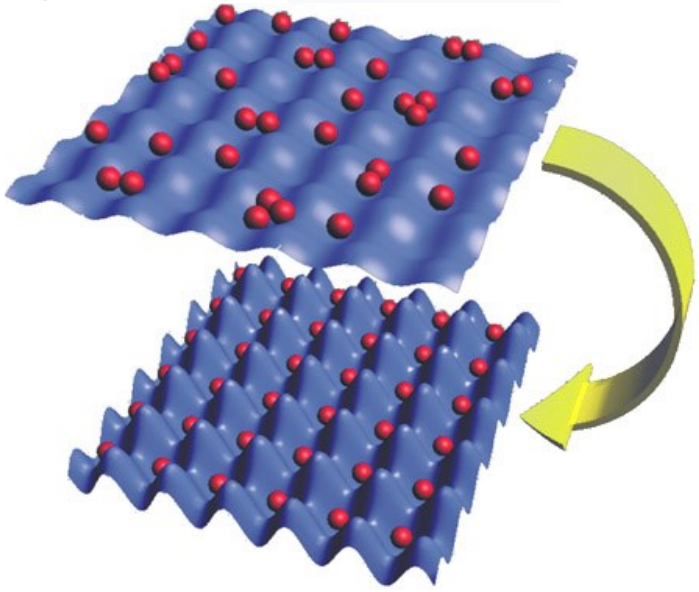
When this choice becomes critical

- In presence of **strong electron correlation**
  - Strong **local Coulomb repulsion** (aka strong correlation for a physicist)
  - Molecular **dissociation limit** (strong correlation for a chemist)
  - Predominance of **charge or spin fluctuations** (Mott phases/magnetic phases)
- When **high accuracy** is required
  - Competing phases
  - Weak dispersive forces
  - Subtle interplay between structural and electronic degrees of freedom

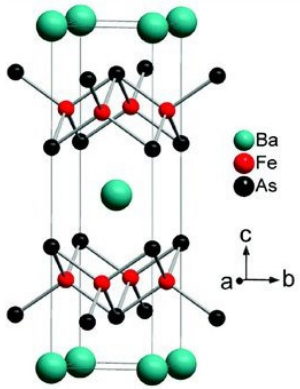
d or f orbitals: very localized on the atomic sites

$$U_{ij} = \int d\mathbf{r} d\mathbf{r}' |\phi_i(\mathbf{r})|^2 \frac{1}{|\mathbf{r} - \mathbf{r}'|} |\phi_j(\mathbf{r}')|^2$$

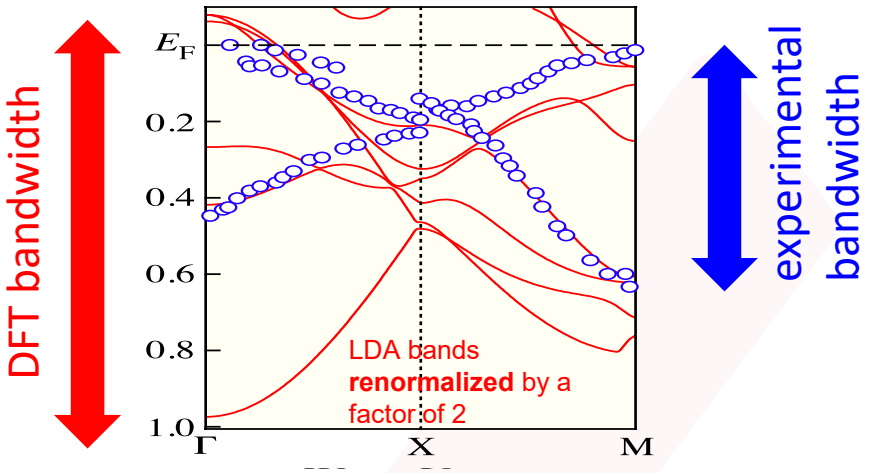
→ Strong local Coulomb repulsion U (large Hubbard U parameters)
"*ATOMIC PHYSICS*" becomes relevant
(breakdown of the local density approximation in DFT functionals)



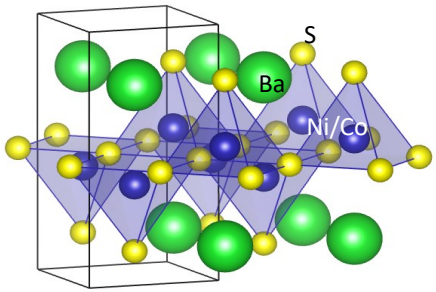
Textbook phenomenon
Mott transition
charge freezing to minimize local U repulsion



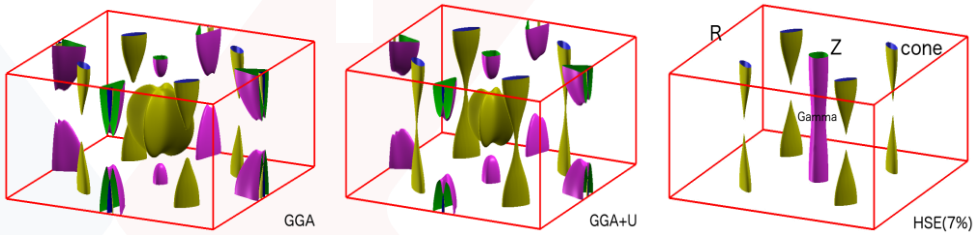
$BaFe_2As_2$ unconventional iron-based superconductor



Reduced bandwidth in experiment!



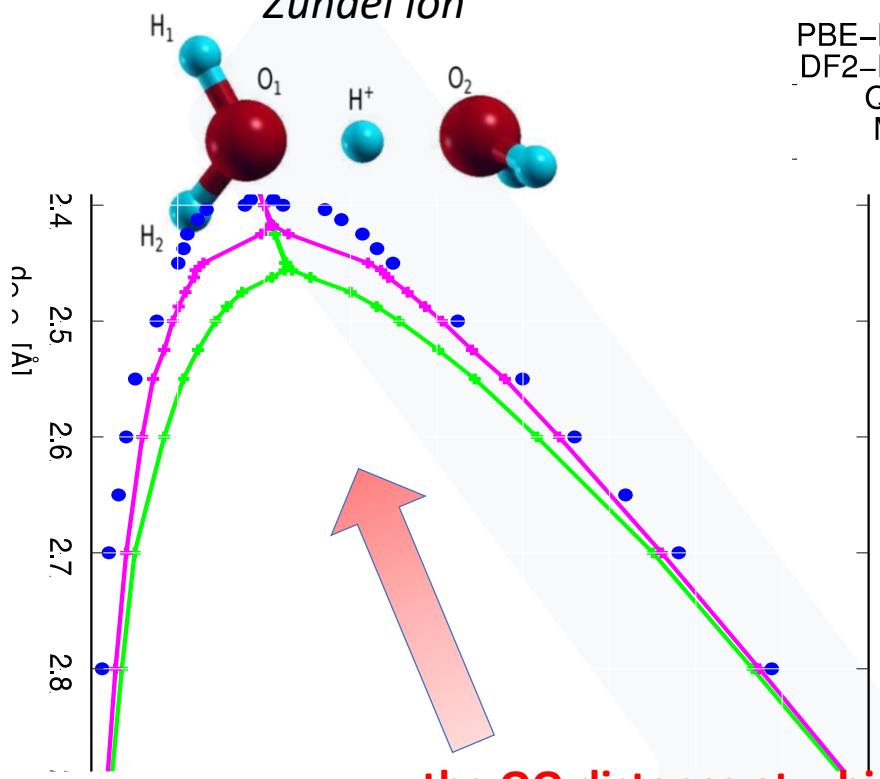
$BaNiS_2$ transition metal (semimetal)

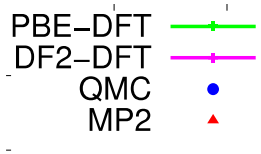


Fermi surface topology strongly functional dependent

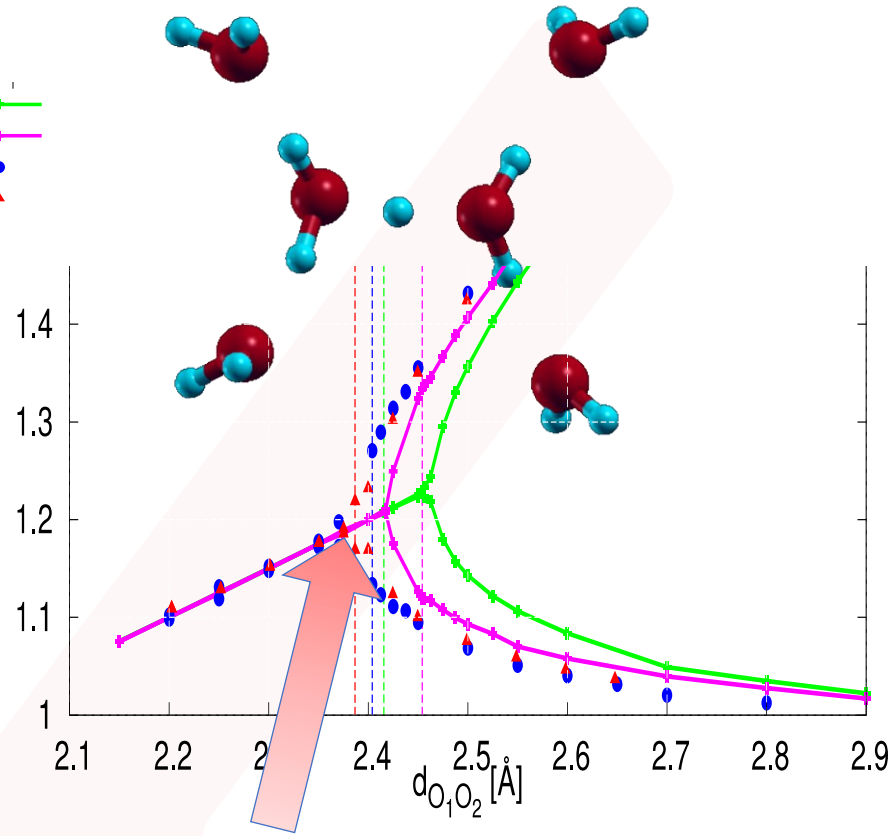
Protonated water clusters

Zundel ion





Protonated water hexamer



the OO distance at which the proton hopping takes place
<u>strongly depends on the level of theory</u>

$$H\Psi(r_1,\ldots,r_N) = E\Psi(r_1,\ldots,r_N)$$

Correlation: beyond Hartree-Fock!

$$\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) = \exp(-J)\sum_k d_k D_k^{\uparrow} D_k^{\downarrow}$$

- Correlation coming from the Jastrow factor (usually called **dynamical correlation**)
- Correlation coming from the antisymmetric part (usually called **static correlation**)

$$H\Psi(r_1,\ldots,r_N) = E\Psi(r_1,\ldots,r_N)$$

Correlation: beyond Hartree-Fock!

$$\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_N) = \exp(-J)\sum_k d_k D_k^{\uparrow} D_k^{\downarrow}$$

- Correlation coming from the Jastrow factor (usually called **dynamical correlation**)
- Correlation coming from the antisymmetric part (usually called **static correlation**)

Drawback: expectation value of the Hamiltonian

$$E = \langle\Psi|H|\Psi\rangle / \langle\Psi|\Psi\rangle$$

becomes much harder to compute than in Hartree-Fock

if a Jastrow-correlated wavefunction is used

Variational energy E: quantum expectation value of the Hamiltonian H

$$E = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$

Deterministic numerical integration (à la Simpson) in 3N variables
At finite mesh M, its error grows exponentially with N

Stochastic numerical integration on M points
Its error decreases as $1/\sqrt{M}$ independently of N!!!

In **3D** already with 3 particles, the Simpson error decreases slower than $1/\sqrt{M}$, with M number of mesh points

For large N, the *stochastic way* of performing the integral is much more efficient!!!

Quadrature (Simpson–like) schemes:

- A regular grid with 10 mesh points per axis would require $10^{d \cdot N}$ evaluations of the integrand, for N particles in d dimensions, i.e., 10^{30} operations for 10 particles in 3 dimensions!
- A simple operation takes say about 10^{-9}s on a present computer. A year is about 3×10^{7}s.
- Integration by quadrature even for 10 particles would take too many years!

Quadrature (Simpson–like) schemes:

- A regular grid with 10 mesh points per axis would require $10^{d \cdot N}$ evaluations of the integrand, for N particles in d dimensions, i.e., 10^{30} operations for 10 particles in 3 dimensions!
- A simple operation takes say about 10^{-9}s on a present computer. A year is about 3×10^{7}s.
- Integration by quadrature even for 10 particles would take too many years! **About 3×10^{13} years!**

PS: age of the universe is about 10^{9} years

- Goal: Computing the integral in a reasonable amount of time **at a fixed target error**
- **How the error scales with the number of points M?**

- If we integrate over a hypercube of side L, with a mesh of size h, the number of grid points is $M = (L/h)^{d \cdot N}$, i.e. $h \propto M^{-1/(d \cdot N)}$.
- Assume that the error $\propto h^l$. Hence

$$\text{error} \propto 1/M^{l/(d \cdot N)}$$

Since l is of order unity, the error decays exceedingly slowly with M.
In fact, the larger is $d \cdot N$ the slower decays the error.

- For $N = 10$, $d = 3$, $l = 4$ (Simpson rule), halving the error of an evaluation with M points requires going to

$$2^{d \cdot N/l} \cdot M = 2^{3 \cdot 10/4} \cdot M \approx 180 \cdot M$$

points; to reduce it by a factor 4 requires $10^5 \cdot M$ points, and so on!

- Monte Carlo Integration is the only choice: (R is a d N dimensional vector)

$$\int dR\, \pi(R)\mathcal{O}(R) \simeq \frac{1}{M}\sum_{i=1}^{M}\mathcal{O}(R_i), \quad M \text{ large}$$

with an

$$\text{error} \propto 1/\sqrt{M},$$

provided that the configurations or walkers R_i are distributed with the probability $\pi(R)$ (in the case of classical Monte Carlo $\pi(R) = \rho(R)$).

- To halve the error only $4 \cdot M$ points are required; $16 \cdot M$ points are sufficient to reduce the error by a factor 4; and so on. Also, there is no dependence on the dimensionality of the configuration space.

d N multidimensional integral

Deterministic intergation

$$M' = k^{\frac{dN}{4}} M$$

Exponential behavior with N

Stochastic intergation

$$M' = k^2 M$$

Independent of N

d N multidimensional integral

Deterministic intergation

$$M' = k^{\frac{dN}{4}} M$$

Exponential behavior with N

Stochastic intergation

$$M' = k^2 M$$

Independent of N

How to generate configurations R_i distributed according to $\Pi(R_i)$?

How to generate configurations R_i distributed according to $\Pi(R_i)$?

By using random numbers!

- Random sequence of numbers drawn from an assigned probability density, say $u(x)$:

$u(x)dx =$ probability that x falls between x and $x+dx$, $\int u(x)dx = 1$

- Uniform variates:

$$u(x) = 1/(\beta-\alpha), \quad \alpha < x < \beta, \quad \text{or}$$
$$u(x) = 1, \quad 0 < x < 1.$$

- Other variates

$$u(y) = e^{-y}, \quad 0 < y < \infty,$$
$$u(y) = e^{-y^2}/\sqrt{\pi}, \quad -\infty < y < \infty.$$

- Let us concentrate on the uniform distribution ($0 < x < 1$):
  - a proper generator will produce values of x placed at random in the given interval;
  - for large generation numbers the x values will be uniformly distributed in $0 < x < 1$;
  - for large generation numbers both the average and variance calculated on the generated values will reproduce those of the assigned uniform distribution.
- In practice pseudo random numbers are generated on computers with deterministic rules [sequences are perfectly reproducible!].
- In the following we shall assume that a good generator of uniform variates is provided, disregarding the issue of how to device it.

How to generate random numbers distributed according to non-uniform variates starting from pseudorandom numbers uniformely distributed?

How we can generate non-uniform variates from uniform ones, $\pi(y)$ from $u(x) = 1,\ 0 < x < 1$? Let us consider the inversion method.

- Look for $y = f(x)$ such that if x's are distributed according to $u(x)$, then $y = f(x)$ are distributed according to $\pi(y)$.
- Start from

$$u(x)dx = \pi(y)dy$$

and use $u(x) = 1$ to get

$$x = \int^{y} dy' \pi(y') = W(y).$$

If $W(y)$ and its inverse are known, $y = f(x) = W^{-1}(x)$

Sampling non-uniform variates: the rejection method

When the inverse of $W(y) = \int_{-\infty}^{y} dy' \pi(y')$ is not known

[$\pi(y)$ non-uniform], one may resort to the rejection methods

- Look for $f(y) \geq \pi(y)$. Here we choose $f(y) = \pi_{max}$.
- Generate a uniform random number y in the domain of π
- Generate a second uniform random number ξ in $[0, 1[$

$$\pi(y)/\pi_{max} \geq \xi, \quad \text{accept } y$$
$$\pi(y)/\pi_{max} < \xi, \quad \text{reject } y$$

It is easily seen that y is distributed according to $\pi(y)$.

- Note: the normalization of $\pi(y)$ is not necessary!
- Metropolis method is a particular rejection method.

Static and dynamic Monte Carlo (MC) methods

- The rejection method is a static MC method: it is inefficient.
- Points are initially generated uniformly in the domain of π, treating on the same footing regions of low and high probability: this becomes very inefficient as one moves to higher dimensionality.
- We need a smarter way in proposing the moves, other than uniform random numbers!
- We have to invent an alternative manner to generate sequences of states (configurations or sample) $\{s_1, s_2, \ldots, s_M\}$, which at equilibrium are distributed with the chosen $\pi(s)$.
- Using a law whereby s_n is determined only by s_{n-1}, through a probability matrix $p(s_{n-1}, s_n)$, naturally leads to the concept of Markov chains

A Markov chain is fully specified by the initial distribution, say $\pi_0(s)$ and by the transition probability $p(s, s')$. Markov chains provide a convenient way to sample multidimensional probability distributions.

The state (or configuration) s of the system is changed randomly according to the transition probability $p(s, s') = p(s \to s')$ satisfying

$$\sum_{s'} p(s, s') = 1 \quad \text{and} \quad p(s, s') \geq 0,$$

thus generating a random walk (or sample) $(s_0, s_1, s_2, \dots)$.

If $p(s, s')$ is ergodic there exists a (unique) probability measure $\pi(s)$ satisfying at equilibrium the stationarity condition:

$$\sum_{s} \pi(s) p(s, s') = \pi(s').$$

How to obtain the desired distribution π as the stationary one?

A sufficient condition to obtain $\pi(s)$ as stationary distribution is to chose the transition probability to satisfy

$$\pi(s)p(s,s') = \pi(s')p(s',s). \quad \textit{Detailed balance condition}$$

In fact summing the above over s one gets

$$\sum_s \pi(s)p(s,s') = \pi(s')\sum_s p(s',s) = \pi(s').$$

The transition probability may be conveniently decomposed into the product of an irreducible proposal or sampling matrix $T(s, s')$ and an acceptance matrix $A(s, s')$

$$p(s, s') = T(s, s')A(s, s').$$

Imposing the detailed balance yields

$$\frac{A(s, s')}{A(s', s)} = \frac{\pi(s')T(s', s)}{\pi(s)T(s, s')} \equiv q(s, s'),$$

which can be satisfied quite generally by choosing

$$A(s, s') = F[q(s, s')],$$

where the function $F : [0, \infty] \to [0, 1]$ satisfies

$$\frac{F[z]}{F[1/z]} = z, \quad \text{for all } z.$$

- Metropolis choice:

$$F[z] = min[1, z]$$

- An alternative choice could be:

$$F[z] = \frac{z}{1 + z}$$

Implementation of Metropolis algorithm

Given a probability $\pi(s)$ to sample (with s a state of the system):

- Choose the proposal matrix $T(s, s')$
  (usually a homogeneous distribution for ΔR_i in the $[-\delta, \delta[$ interval);
- Initialize the system in the state s_0;
- To advance from s_n to s_{n+1}:
  - sample s' from $T(s_n, s')$
    (in the case of the homogeous move, $s' = s_n + (2r_n - 1)\delta$
    with uniform random number $r_n \in [0, 1[$),
  - calculate
    $$q(s_n, s') = \frac{\pi(s')T(s', s_n)}{\pi(s_n)T(s_n, s')},$$
  - generate a random number ξ uniformly distributed with $0 < \xi < 1$
    and compare it with $F = min[q(s_n, s'), 1]$:
    - if $F > \xi$: $s_{n+1} = s'$
    - else $s_{n+1} = s_n$.

- If the proposal matrix T is chosen to be symmetric,

$$T(\mathbf{R}',\mathbf{R}_i) = \begin{cases} 1 & \text{if } |\mathbf{R}' - \mathbf{R}_i| \le \Delta \\ 0 & \text{elsewhere} \end{cases} \qquad A(\mathbf{R}_f \,|\, \mathbf{R}_i) = \min\left[1, \frac{\Pi(\mathbf{R}_f)}{\Pi(\mathbf{R}_i)}\right]$$

the algorithm is called **simple Metropolis**.

- The proposal matrix can be non-symmetric to decrease the correlation time between two configurations (autocorrelation time); in this case the algorithm is called **generalized Metropolis**.

- Throw away the first k *states* as being out of equilibrium;
- Collect averages using the configurations with $n > k$ and block them to calculate error bars.
- The normalization of the probability, $\int ds\, \pi(s)$, is never needed and in fact cannot be calculated (... easily).
- Particles can be moved one at time (single-particle move);
- For the generalized algorithm ($T(s, s')$ is not a constant) one has to sample both forward and reverse transition;
- An optimal acceptance is

$$\mathcal{A} = \frac{\text{moves accepted}}{\text{total moves}} \simeq 1/2.$$

In fact the overall efficiency may dictate different choices (see, e.g., DMC).

Variational Monte Carlo energy

$$E_{VMC} = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{\int d\mathbf{R}\, \Psi^*(\mathbf{R}) H \Psi(\mathbf{R})}{\int d\mathbf{R}\, \Psi^*(\mathbf{R}) \Psi(\mathbf{R})} = \int d\mathbf{R}\, \frac{H\Psi(\mathbf{R})}{\Psi(\mathbf{R})} \frac{|\Psi(\mathbf{R})|^2}{\int d\mathbf{R}\, |\Psi(\mathbf{R})|^2}$$

One would like to evaluate the *true* mean

$$\langle \mathcal{O} \rangle = \int ds\, \pi(s) \mathcal{O}(s),$$

whereas MC yield a sample $(s_1, s_2, \ldots, s_M)$ of length $\simeq M$ of states distributed according to $\pi(s)$. Evidently, one can define a sample mean

$$\overline{\mathcal{O}} = \frac{1}{M} \sum_{i=1}^{M} \mathcal{O}_i, \quad \text{with } \mathcal{O}_i \equiv \mathcal{O}(s_i).$$

The sample mean is an unbiased estimator of the true mean, i.e., $\langle \overline{\mathcal{O}} \rangle = \langle \mathcal{O} \rangle$ independently of M. Also, it is possible to prove:

- the law of large numbers, $\lim_{M \to \infty} \overline{\mathcal{O}} = \langle \mathcal{O} \rangle$;

- The central limit theorem, which states that $\overline{\mathcal{O}}$ is *normally* distributed around $\langle \mathcal{O} \rangle$.

Central Limit Theorem

de Moivre (1733), Laplace (1812), Lyapunov (1901), Pólya (1920)

Let $s_1, s_2, s_3, \ldots, s_M$ be a sequence of M independent random variables sampled from a probability density function with a finite expectation value, μ, and variance σ^2. The central limit theorem states that as the sample size M increases, the probability density of the sample average of these random variables approaches the normal distribution, $\frac{1}{\sqrt{2\pi}\overline{\sigma}} \exp^{-(s-\mu)^2/(2\overline{\sigma}^2)}$, with a mean μ, and variance σ^2/M irrespective of the original probability density function.

Therefore we need to evaluate the variance

$$\sigma^2(\overline{\mathcal{O}}) = \langle (\overline{\mathcal{O}} - \langle \mathcal{O} \rangle)^2 \rangle,$$

whose root we may interpret as statistical error on $\overline{\mathcal{O}}$.

Using $\overline{\mathcal{O}} = (1/M)\sum_{i=1}^{M}\mathcal{O}_i$, one obtains for the variance

$$\sigma^2(\overline{\mathcal{O}}) = \frac{1}{M^2}\sum_{i,j=1}^{M} C(|i-j|) \approx \frac{1}{M^2}\sum_{i=1}^{M}\sum_{t=-\infty}^{\infty} C(|t|) = \frac{\tau}{M}\sigma^2(\mathcal{O}).$$

Here

$$C(t) = \langle \mathcal{O}_s \mathcal{O}_{s+t}\rangle - \langle \mathcal{O}\rangle^2$$

is the normalized *time* autocorrelation function, which evidently reduces to the variance of $\mathcal{O}$ at time 0, $C(0) = \sigma^2(\mathcal{O})$, and the integrated correlation time

$$\tau = 1 + 2\sum_{t=1}^{\infty}\frac{C(t)}{C(0)},$$

$$\tau = 1 + 2\sum_{t=1}^{\infty} \frac{C(t)}{C(0)},$$

accounts for the correlation existing between walkers in the Markov chain.

In general $\tau > 1$.

A sample estimate of $C(t)$, with a bias of order $1/M$ is given by

$$\tilde{C}(t) = \frac{1}{M - |t|} \sum_{i=1}^{M-|t|} (\mathcal{O}_i - \overline{\mathcal{O}})(\mathcal{O}_{i+|t|} - \overline{\mathcal{O}}).$$

Thus one has an estimate for $\sigma^2(\mathcal{O}) = C(0) \approx \tilde{C}(0)$,

$$\tilde{\sigma}^2(\mathcal{O}) = \frac{1}{M} \sum_{i=1}^{M} (\mathcal{O}_i - \overline{\mathcal{O}})^2,$$

and the correlation time can also be calculated from $\tilde{C}(t)$.

The precise estimate of the error bar requires the calculation of time correlation functions, which one would rather avoid.

An alternative is provided by the blocking procedure. The sample is broken in a number of blocks $M = N_b n_b$, with N_B the number of blocks and and n_b the length of each block. New variable are constructed as block averages

$$\mathcal{O}_{b,I} = \frac{1}{n_b} \sum_{i=1}^{n_b} \mathcal{O}_{(I-1)n_b+i},$$

and clearly have a mean equal to the run mean $\overline{\mathcal{O}}$. Intuitively, if $n_b \gg \tau$, this new variables should become statistically independent and therefore

variance around their mean $\overline{\mathcal{O}}$ given by

$$\sigma^2(\mathcal{O}_b) = \frac{1}{N_b(N_b-1)} \sum_{I=1}^{N_b} (\mathcal{O}_{b,I} - \overline{\mathcal{O}})^2$$

One can indeed show that provided $n_b \gg \tau$ and yet $n_b \ll M$ or equivalently N_b large

$$\sigma^2(\mathcal{O}_b) \simeq \sigma^2(\overline{\mathcal{O}}).$$

A plot of $\sigma^2(\mathcal{O}_b)$ versus n_b will reveal a plateau, where in fact the above relation holds, and therefore it also yields an estimate of the correlation time.

$$E_{VMC} = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{\int d\mathbf{R}\, \Psi^*(\mathbf{R}) H \Psi(\mathbf{R})}{\int d\mathbf{R}\, \Psi^*(\mathbf{R}) \Psi(\mathbf{R})} = \int d\mathbf{R}\, \frac{H\Psi(\mathbf{R})}{\Psi(\mathbf{R})} \frac{|\Psi(\mathbf{R})|^2}{\int d\mathbf{R}\, |\Psi(\mathbf{R})|^2}$$

$$= \int d\mathbf{R}\, E_L(\mathbf{R}) \Pi(\mathbf{R}) = \langle E_L \rangle_\Psi$$

Local energy $E_L(\mathbf{R}) = \frac{H\Psi(\mathbf{R})}{\Psi(\mathbf{R})}$

Probability distribution $\Pi(\mathbf{R}) = \frac{|\Psi(\mathbf{R})|^2}{\int d\mathbf{R}\, |\Psi(\mathbf{R})|^2}$

Generate a sample of M points $\boldsymbol{R}_i$ distributed according to Π and average E_L over this sample:

$$\langle E_L \rangle_\Psi \approx \frac{1}{M} \sum_{i=1}^{M} E_L(\mathbf{R}_i)$$

With the aim to obtain a set of $\{\mathbf{R}_1,\ldots,\mathbf{R}_M\}$ distributed as Π

1. Pick a starting $\boldsymbol{R}_i$ and initialize the configuration
2. Advance the configuration from $\boldsymbol{R}_i$ to $\boldsymbol{R}_f$
   a) Sample $\boldsymbol{R'}$ from $T(\mathbf{R}'|\mathbf{R}_i)$
   b) Calculate the ratio $q=\frac{T(\mathbf{R}_i|\mathbf{R}')\Pi(\mathbf{R}')}{T(\mathbf{R}'|\mathbf{R}_i)\Pi(\mathbf{R}_i)}$
   c) Accept or reject with probability F=min[q,1]
      pick a uniformly distributed random number $p \in [0,1[$
      if p < F, move accepted ➡ set $\boldsymbol{R}_f=\boldsymbol{R'}$
      if p ⌊ F, move rejected ➡ set $\boldsymbol{R}_f=\boldsymbol{R}_i$
3. Throw away first *k* configurations of equilibration time
4. Collect the averages and block them to obtain the error bars

Within VMC, we can use any “computable” wave function, if it is

- Continuous, normalizable, proper symmetry
- Finite variance

$$\sigma^2 = \frac{\langle \Psi | (H - E_{VMC})^2 | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \langle (E_L - E_{VMC})^2 \rangle_\Psi$$

since the Monte Carlo error goes as $\sigma \propto \sqrt{\frac{\sigma^2}{M}}$

Zero variance principle

If the wave function Ψ is an exact eigenstate of H, the local energy E_L is a constant, and the variance is zero.

Therefore, the closer the wave function to an eigenstate, the faster the convergence of Monte Carlo error with the number of samples M.