An introduction to Quantum Monte Carlo

Jose Guilherme Vilhena
Laboratoire de Physique de la Matière Condensée et Nanostructures,
Université Claude Bernard Lyon 1

Outline

1) Quantum many body problem;

2) Metropolis Algorithm;

3) Statistical foundations of Monte Carlo methods;

4) Quantum Monte Carlo methods ;
   4.1) Overview;
   4.2) Variational Quantum Monte Carlo;
   4.3) Diffusion Quantum Monte Carlo;
   4.4) Typical QMC calculation;

Outline

1) Quantum many body problem;

2) Metropolis Algorithm;

3) Statistical foundations of Monte Carlo methods;

4) Quantum Monte Carlo methods ;
   4.1) Overview;
   4.2) Variational Quantum Monte Carlo;
   4.3) Diffusion Quantum Monte Carlo;
   4.4) Typical QMC calculation;

1 - Quantum many body problem

- Consider of N electrons. Since $m_e << M_N$, to a good approximation, the electronic dynamics is governed by :

$$\hat{H} = -\frac{1}{2}\sum_i \nabla_i^2 - \sum_i \sum_\alpha \frac{Z_\alpha}{|\boldsymbol{r}_i - \boldsymbol{d}_\alpha|} + \frac{1}{2}\sum_i \sum_{j \neq i} \frac{Z_\alpha}{|\boldsymbol{r}_i - \boldsymbol{r}_j|}$$

which is the Born Oppenheimer Hamiltonian.

- We want to find the eigenvalues and eigenvectos of this hamiltonian, i.e. :

$$\hat{H}\,\Psi(r_{1,}r_{2,}..,r_N) = E\,\Psi(r_{1,}r_{2,}..,r_N)$$

QMC allow us to solve numerically this, thus providing E_0 and Ψ_0 .

1 - Quantum many body problem

- Consider of N electrons. Since $m_e << M_N$, to a good approximation, the elotonic dynamics is governed by :

$$\hat{H} = -\frac{1}{2}\sum_i \nabla_i^2 - \sum_i \sum_\alpha \frac{Z_\alpha}{|\boldsymbol{r}_i - \boldsymbol{d}_\alpha|} + \frac{1}{2}\sum_i \sum_{j \neq i} \frac{Z_\alpha}{|\boldsymbol{r}_i - \boldsymbol{r}_j|}$$

which is the Born Oppenheimer Hamiltonian.

- We want to find the eigenvalues and eigenvectos of this hamiltonian, i.e. :

$$\hat{H}\,\Psi(r_1, r_{2,}.., r_N) = E\,\Psi(r_1, r_{2,}.., r_N)$$

QMC allow us to solve numerically this, thus providing E_0 and Ψ_0 .

1 - Quantum many body problem

- **Varionational principle**
  For any normalized wfn Ψ: $\langle \Psi_T(\boldsymbol{R})|\hat{H}|\Psi_T(\boldsymbol{R})\rangle \geqslant E_0$

Zero Varience Property

The variance of the "local energy" must vanish for $\Psi_T=\Psi_0$,

i.e. $\sigma^2_{E_L}=\int \frac{|\Psi_T|^2}{\int d\boldsymbol{R}|\Psi_T|^2}(E_L-\langle E_L\rangle)^2=0$; *where* $E_L=\frac{H\ \Psi_T(\boldsymbol{R})}{\Psi_T(\boldsymbol{R})}$

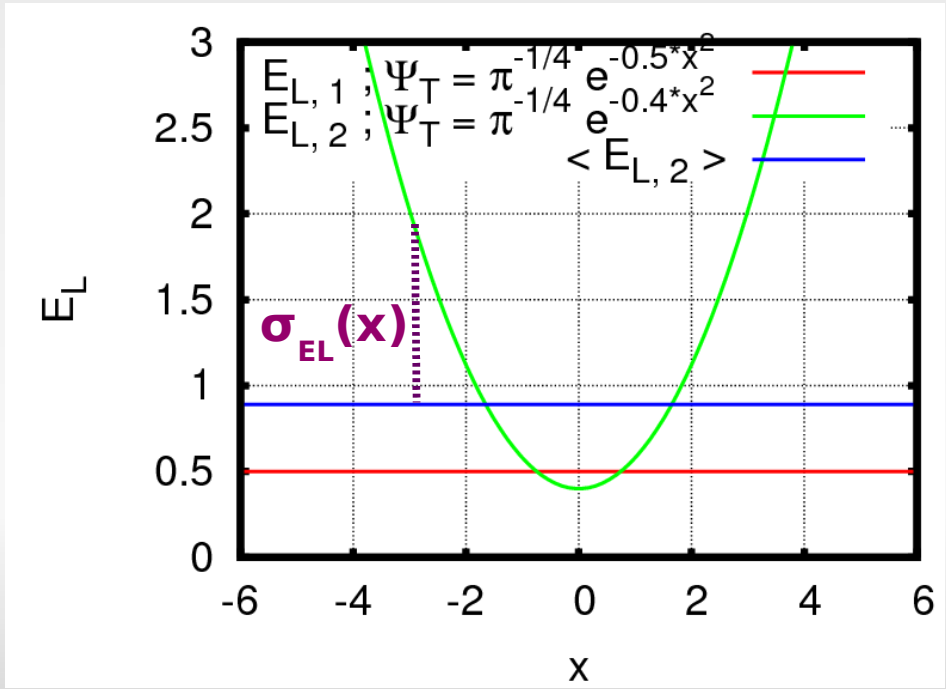
Example :

Harmonic Oscilator

$$E_L(x)=\frac{H\Psi_T(x)}{\Psi_T(x)}$$

then ...

$$\partial_x E_L(x)\begin{cases}=0 & ;if\ \ \Psi_T=\Psi_0\\ \neq 0 & ;if\ \ \Psi_T\neq\Psi_0\end{cases}$$



1 - Quantum many body problem

- **Varionational principle**
  For any normalized wfn Ψ: $\langle \Psi_T(\boldsymbol{R})|\hat{H}|\Psi_T(\boldsymbol{R})\rangle \geqslant E_0$

Zero Varience Property
The variance of the "local energy" must vanish for $\Psi_T=\Psi_0$,

i.e. $\sigma^2_{E_L}=\int \frac{|\Psi_T|^2}{\int d\boldsymbol{R}|\Psi_T|^2}(E_L-\langle E_L\rangle)^2=0$; *where* $E_L=\frac{H\ \Psi_T(\boldsymbol{R})}{\Psi_T(\boldsymbol{R})}$

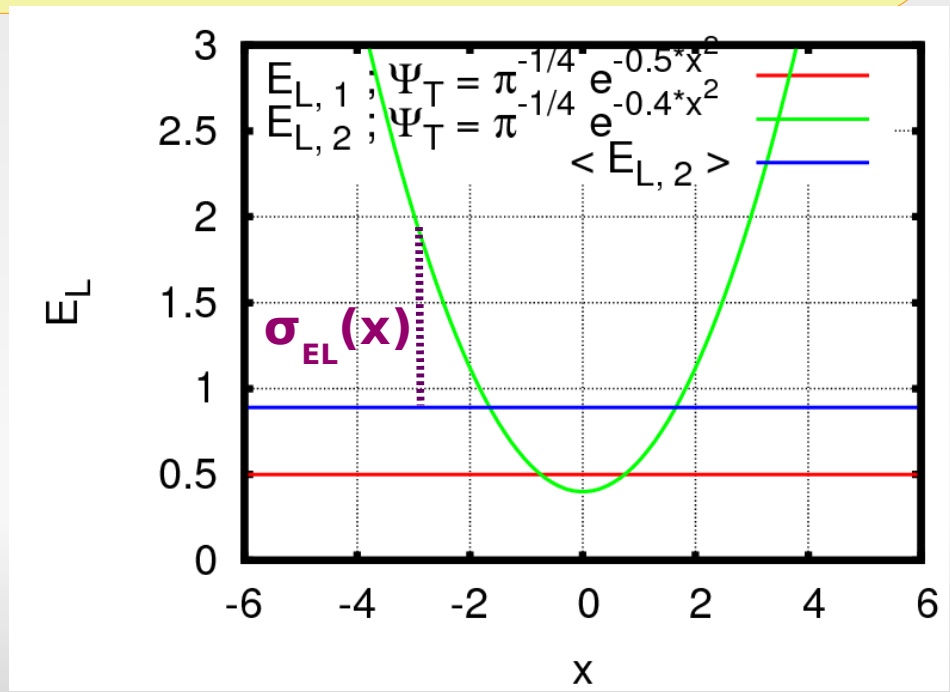
Example :

Harmonic Oscilator

$$E_L(x)=\frac{H\Psi_T(x)}{\Psi_T(x)}$$

then ...

$$\partial_x E_L(x)\begin{cases}=0 & ;if\ \ \Psi_T=\Psi_0\\ \neq 0 & ;if\ \ \Psi_T\neq\Psi_0\end{cases}$$



1 - Quantum many body problem

- **Varionational principle**
  For any normalized wfn Ψ: $\langle \Psi_T(\boldsymbol{R})|\hat{H}|\Psi_T(\boldsymbol{R})\rangle \geqslant E_0$

Zero Varience Property

The variance of the "local energy" must vanish for $\Psi_T=\Psi_0$,

i.e. $\sigma^2_{E_L}=\int \frac{|\Psi_T|^2}{\int d\boldsymbol{R}|\Psi_T|^2}(E_L-\langle E_L\rangle)^2=0$; *where* $E_L=\frac{H\ \Psi_T(\boldsymbol{R})}{\Psi_T(\boldsymbol{R})}$

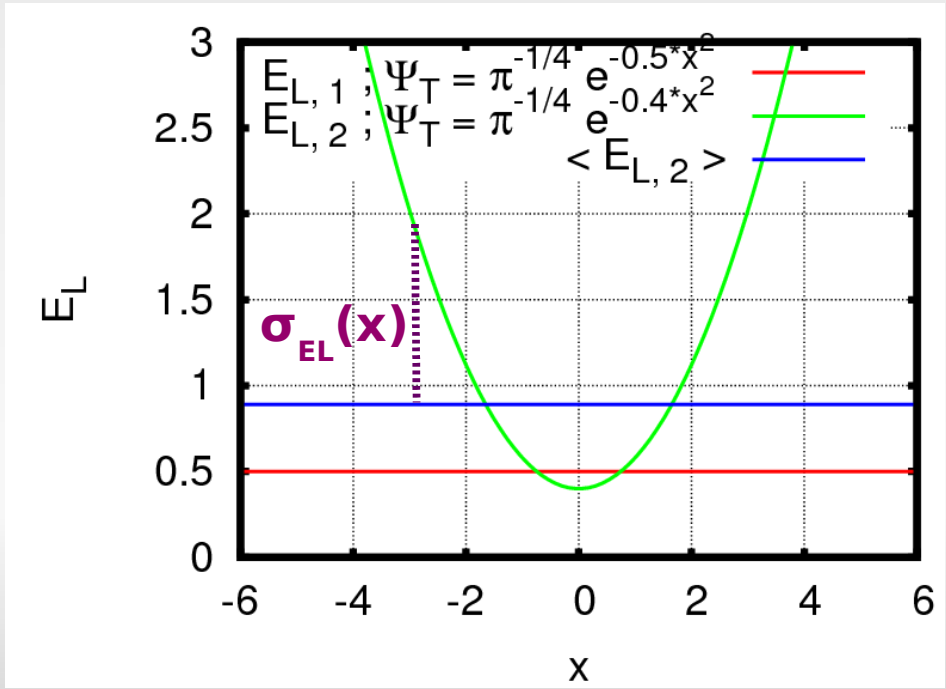
Example :

Harmonic Oscilator

$$E_L(x)=\frac{H\Psi_T(x)}{\Psi_T(x)}$$

then...

$$\partial_x E_L(x)\begin{cases}=0 & ;if\ \ \Psi_T=\Psi_0\\ \neq 0 & ;if\ \ \Psi_T\neq\Psi_0\end{cases}$$



Outline

1) Quantum many body problem;

2) Metropolis Algorithm;

3) Statistical foundations of Monte Carlo methods;

4) Quantum Monte Carlo methods ;
   - 4.1) Overview;
   - 4.2) Variational Quantum Monte Carlo;
   - 4.3) Diffusion Quantum Monte Carlo;
   - 4.4) Typical QMC calculation

2 - Metropolis Algorithm

- **Markov Chain** stocastic process (no memory).

- **Metropolis-Hastings algorith** (Markov chain) randomly samples a “problematic” probability distribuition function;

- Some definitions ...
  - > P(**R**): normalized probability distribuition function
  - > **R**=($\mathbf{r}_1$,..., $\mathbf{r}_N$) : configuration or walker (can be vector with the positions of all electrons)
  - > T(**R'**←**R**): Proposal density (can be 1, a gaussian, ...)
  - > q=rand(0,1): random number in [0,1]

2 - Metropolis Algorithm

- We seek a set $\{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_M\}$ of sample points of P(R).
  How can we get them ?!?
- **Metropolis Algorithm:**
  1) Start in a point $\mathbf{R}^t$;
  2) Propose a point **R'**, with a probability of T(**R'←R**) ;
  3) Calculate the “acceptance rate” : $A(\boldsymbol{R}' \leftarrow \boldsymbol{R}) = \frac{T(\boldsymbol{R}' \leftarrow \boldsymbol{R}) P(\boldsymbol{R})}{T(\boldsymbol{R} \leftarrow \boldsymbol{R}') P(\boldsymbol{R}')}$
  4) if A > 1 then : R^{t+1}=R'
     else if A > q then : R^{t+1}=R'
     else : R^{t+1}=R^t
  5) Repeat all from setp 2;
  6) stop at k>M , and take away the first k-M points.
- The sampled points match P(**R**).
- The mixing is controled by T(**R'←R**).

2 - Metropolis Algorithm

- We seek a set $\{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_M\}$ of sample points of P(R).
  How can we get them ?!?
- **Metropolis Algorithm:**
  1) Start in a point $\mathbf{R}^t$;
  2) Propose a point **R'**, with a probability of T(**R'←R**) ;
  3) Calculate the “acceptance rate” : $A(\boldsymbol{R}' \leftarrow \boldsymbol{R}) = \frac{T(\boldsymbol{R}' \leftarrow \boldsymbol{R}) P(\boldsymbol{R})}{T(\boldsymbol{R} \leftarrow \boldsymbol{R}') P(\boldsymbol{R}')}$
  4) if A > 1 then : R^{t+1}=R'
     else if A > q then : R^{t+1}=R'
     else : $R^{t+1}=R^t$
  5) Repeat all from setp 2;
  6) stop at k>M , and take away the first k-M points.
- The sampled points match P(**R**).
- The mixing is controled by T(**R'←R**).

2 - Metropolis Algorithm

- We seek a set $\{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_M\}$ of sample points of P(R).
  How can we get them ?!?
- **Metropolis Algorithm:**
  1) Start in a point $\mathbf{R}^t$;
  2) Propose a point **R'**, with a probability of T(**R'←R**) ;
  3) Calculate the “acceptance rate” : $A(\boldsymbol{R}' \leftarrow \boldsymbol{R}) = \frac{T(\boldsymbol{R}' \leftarrow \boldsymbol{R}) P(\boldsymbol{R})}{T(\boldsymbol{R} \leftarrow \boldsymbol{R}') P(\boldsymbol{R}')}$
  4) if A > 1 then : R^{t+1}=R'
     else if A > q then : R^{t+1}=R'
     else : R^{t+1}=R^t
  5) Repeat all from setp 2;
  6) stop at k>M , and take away the first k-M points.
- The sampled points match P(**R**).
- The mixing is controled by T(**R'←R**).

2 - Metropolis Algorithm

- We seek a set $\{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_M\}$ of sample points of P(R).
  How can we get them ?!?
- **Metropolis Algorithm:**
  1) Start in a point $\mathbf{R}^t$;
  2) Propose a point **R'**, with a probability of T(**R'←R**) ;
  3) Calculate the “acceptance rate” : $A(\boldsymbol{R}' \leftarrow \boldsymbol{R}) = \frac{T(\boldsymbol{R}' \leftarrow \boldsymbol{R}) P(\boldsymbol{R})}{T(\boldsymbol{R} \leftarrow \boldsymbol{R}') P(\boldsymbol{R}')}$
  4) if A > 1 then : R^{t+1}=R'
     else if A > q then : R^{t+1}=R'
     else : $R^{t+1}=R^t$
  5) Repeat all from setp 2;
  6) stop at k>M , and take away the first k-M points.
- The sampled points match P(**R**).
- The mixing is controled by T(**R'←R**).

2 - Metropolis Algorithm

- We seek a set $\{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_M\}$ of sample points of P(R).
  How can we get them ?!?
- **Metropolis Algorithm:**
  1) Start in a point $\mathbf{R}^t$;
  2) Propose a point **R'**, with a probability of T(**R'←R**) ;
  3) Calculate the “acceptance rate” : $A(\boldsymbol{R}' \leftarrow \boldsymbol{R}) = \frac{T(\boldsymbol{R}' \leftarrow \boldsymbol{R}) P(\boldsymbol{R})}{T(\boldsymbol{R} \leftarrow \boldsymbol{R}') P(\boldsymbol{R}')}$
  4) if A > 1 then : R^{t+1}=R'
     else if A > q then : R^{t+1}=R'
     else : R^{t+1}=R^t
  5) Repeat all from setp 2;
  6) stop at k>M , and take away the first k-M points.
- The sampled points match P(**R**).
- The mixing is controled by T(**R'←R**).

2 - Metropolis Algorithm

- We seek a set {$\mathbf{R}_1$, $\mathbf{R}_2$, ..., $\mathbf{R}_M$} of sample points of P(R).
  How can we get them ?!?
- **Metropolis Algorithm:**
  1) Start in a point $\mathbf{R}^t$;
  2) Propose a point **R'**, with a probability of T(**R'←R**) ;
  3) Calculate the “acceptance rate” : $A(\boldsymbol{R}' \leftarrow \boldsymbol{R}) = \frac{T(\boldsymbol{R}' \leftarrow \boldsymbol{R}) P(\boldsymbol{R})}{T(\boldsymbol{R} \leftarrow \boldsymbol{R}') P(\boldsymbol{R}')}$
  4) if A > 1 then : R^{t+1}=R'
     else if A > q then : R^{t+1}=R'
     else : R^{t+1}=R^t
  5) Repeat all from setp 2;
  6) stop at k>M , and take away the first k-M points.
- The sampled points match P(**R**).
- The mixing is controled by T(**R'←R**).

2 - Metropolis Algorithm

- We seek a set $\{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_M\}$ of sample points of P(R).
  How can we get them ?!?
- **Metropolis Algorithm:**
  1) Start in a point $\mathbf{R}^t$;
  2) Propose a point **R'**, with a probability of T(**R'←R**) ;
  3) Calculate the “acceptance rate” : $A(\boldsymbol{R}' \leftarrow \boldsymbol{R}) = \frac{T(\boldsymbol{R}' \leftarrow \boldsymbol{R}) P(\boldsymbol{R})}{T(\boldsymbol{R} \leftarrow \boldsymbol{R}') P(\boldsymbol{R}')}$
  4) if A > 1 then : R^{t+1}=R'
     else if A > q then : R^{t+1}=R'
     else : $R^{t+1}=R^t$
  5) Repeat all from setp 2;
  6) stop at k>M , and take away the first k-M points.
- The sampled points match P(**R**).
- The mixing is controled by T(**R'←R**).

Outline

1) Quantum many body problem;

2) Metropolis Algorithm;

3) Statistical foundations of Monte Carlo methods;

4) Quantum Monte Carlo methods ;
   - 4.1) Overview;
   - 4.2) Variational Quantum Monte Carlo;
   - 4.3) Diffusion Quantum Monte Carlo;
   - 4.4) Typical QMC calculation

3 – Statistical foundations of MC methods

- Monte Carlo methods: solve problems using random numbers.

- Evaluate integrals:
  - expectation value of a random variable is just the integral over its probability distribution
  - generate a bunch of random numbers and average to get the integral;

- Simulate random processes with random walkers.

- Number of dimensions doesn't matter.
  - Simpsons rule (error ~ O ($M^{-4/d}$))
  - Monte Carlo method (error ~ O ($M^{-1/2}$))

3 – Statistical foundations of MC methods

- Monte Carlo methods: solve problems using random numbers.

- Evaluate integrals:
  - expectation value of a random variable is just the integral over its probability distribution
  - generate a bunch of random numbers and average to get the integral;

- Simulate random processes with random walkers.

- Number of dimensions doesn't matter.
  - Simpsons rule (error ~ O ($M^{-4/d}$))
  - Monte Carlo method (error ~ O ($M^{-1/2}$))

3 – Statistical foundations of MC methods

- Monte Carlo methods: solve problems using random numbers.

- Evaluate integrals:
  - expectation value of a random variable is just the integral over its probability distribution
  - generate a bunch of random numbers and average to get the integral;

- Simulate random processes with random walkers.

- Number of dimensions doesn't matter.
  - Simpsons rule (error ~ O ($M^{-4/d}$))
  - Monte Carlo method (error ~ O ($M^{-1/2}$))

3 – Statistical foundations of MC methods

- Monte Carlo methods: solve problems using random numbers.

- Evaluate integrals:
  - expectation value of a random variable is just the integral over its probability distribution
  - generate a bunch of random numbers and average to get the integral;

- Simulate random processes with random walkers.

- Number of dimensions doesn't matter.
  - Simpsons rule (error ~ O ($M^{-4/d}$))
  - Monte Carlo method (error ~ O ($M^{-1/2}$))

3 – Statistical foundations of MC methods

- Condider a well behaved f(**R**) such that:

$$\mu_f = \int d\,\boldsymbol{R} f(\boldsymbol{R}) P(\boldsymbol{R}) \quad ; \quad \sigma_f = \int d\,\boldsymbol{R} [f(\boldsymbol{R}) - \mu_f]^2 P(\boldsymbol{R})$$

where P(**R**) is any normed **p**robability **d**ensity **f**unction**.**

- **Central limit theorem:**

For any p.d.f. , and a large enough ramdom sample $\{R_1, R_2, \ldots R_M\}$ of it :

$$F(\boldsymbol{R}) = \frac{1}{M} \sum_{k=0}^{M} f(\boldsymbol{R}_k) \approx \mu_f \quad ; \quad \lim_{M \to \infty} F(\boldsymbol{R}) = \int d\,\boldsymbol{R} f(\boldsymbol{R}) P(\boldsymbol{R})$$

$$\sigma_F^2(M) = \langle (F(\boldsymbol{R}; M) - \mu_f)^2 \rangle = \frac{\sigma_f^2}{M}$$

i.e., F(**R**) is normally distributed with $\bar{F} = \mu_f$ and $\sigma_F = \sigma_f M^{-1/2}$

3 – Statistical foundations of MC methods

- Condider a well behaved f(**R**) such that:

$$\mu_f = \int d\,\boldsymbol{R} f(\boldsymbol{R}) P(\boldsymbol{R}) \quad ; \quad \sigma_f = \int d\,\boldsymbol{R} [f(\boldsymbol{R}) - \mu_f]^2 P(\boldsymbol{R})$$

where P(**R**) is any normed **p**robability **d**ensity **f**unction**.**

- **Central limit theorem:**

For any p.d.f. , and a large enough ramdom sample $\{R_1, R_2, \ldots R_M\}$ of it :

$$F(\boldsymbol{R}) = \frac{1}{M} \sum_{k=0}^{M} f(\boldsymbol{R}_k) \approx \mu_f \quad ; \quad \lim_{M \to \infty} F(\boldsymbol{R}) = \int d\,\boldsymbol{R} f(\boldsymbol{R}) P(\boldsymbol{R})$$

$$\sigma_F^2(M) = \langle (F(\boldsymbol{R}; M) - \mu_f)^2 \rangle = \frac{\sigma_f^2}{M}$$

i.e., F(**R**) is normally distributed with $\overline{F} = \mu_f$ and $\sigma_F = \sigma_f M^{-1/2}$

3 – Statistical foundations of MC methods

Consider the integral:

$$I = \int g(\boldsymbol{R}) d\,\boldsymbol{R} = \int P(\boldsymbol{R}) f(\boldsymbol{R}) d\,\boldsymbol{R}$$

P(**R**) is a normed p.d.f. and f(**R**)=g(**R**)/P(**R**)

i) Metropolis algorithm samples P(**R**) to **get** $\{\mathbf{R}_1, \mathbf{R}_2, \ldots \mathbf{R}_M\}$;

ii) Monte Carlo **estimative of I** is $I_{MC} = \frac{1}{M} \sum_{k=1}^{M} f(R_k) \approx I$

with a statistical error of $\sigma_{IMC} = \sigma_f M^{-1/2}$

iii) We can estimate σ_{IMC} using MC again, i.e. :

$$\sigma_f = \frac{1}{M} \sum_{i=1}^{M} \left[f(\boldsymbol{R}_i) - \frac{1}{M} \sum_{j=1}^{M} f(\boldsymbol{R}_j) \right]^2$$

3 – Statistical foundations of MC methods

Consider the integral:

$$I = \int g(\boldsymbol{R}) d\,\boldsymbol{R} = \int P(\boldsymbol{R}) f(\boldsymbol{R}) d\,\boldsymbol{R}$$

P(**R**) is any normed p.d.f. and f(**R**)=g(**R**)/P(**R**)

i) Metropolis algorithm samples P(**R**) to **get {R_1,R_2,...R_M}**;

ii) Monte Carlo **estimative of I** is $I_{MC} = \frac{1}{M}\sum_{k=1}^{M} f(R_k) \approx I$

with a statistical error of $\sigma_{IMC} = \sigma_f M^{-1/2}$

iii) The σ_f can be estimated using MC again, i.e. :

$$\sigma_f = \frac{1}{M}\sum_{i=1}^{M}\left[f(\boldsymbol{R}_i) - \frac{1}{M}\sum_{j=1}^{M} f(\boldsymbol{R}_j)\right]^2$$

3 – Statistical foundations of MC methods

Consider the integral:

$$I = \int g(\boldsymbol{R})\, d\boldsymbol{R} = \int P(\boldsymbol{R}) f(\boldsymbol{R})\, d\boldsymbol{R}$$

P(**R**) is any normed p.d.f. and f(**R**)=g(**R**)/P(**R**)

i) Metropolis algorithm samples P(**R**) to **get {R_1,R_2,...R_M}** ;

ii) Monte Carlo **estimative of I** is $I_{MC} = \frac{1}{M}\sum_{k=1}^{M} f(R_k) \approx I$

with a statistical error of $\sigma_{IMC} = \sigma_f M^{-1/2}$

iii) The σ_f can be estimated using MC again, i.e. :

$$\sigma_f = \frac{1}{M}\sum_{i=1}^{M}\left[f(\boldsymbol{R}_i) - \frac{1}{M}\sum_{j=1}^{M} f(\boldsymbol{R}_j)\right]^2$$

3 – Statistical foundations of MC methods

Consider the integral:

$$I = \int g(\boldsymbol{R})\, d\boldsymbol{R} = \int P(\boldsymbol{R}) f(\boldsymbol{R})\, d\boldsymbol{R}$$

P(**R**) is any normed p.d.f. and f(**R**)=g(**R**)/P(**R**)

i) Metropolis algorithm samples P(**R**) to **get {R_1,R_2,...R_M}** ;

ii) Monte Carlo **estimative of I** is $I_{MC} = \frac{1}{M}\sum_{k=1}^{M} f(R_k) \approx I$

with a statistical error of $\sigma_{IMC} = \sigma_f M^{-1/2}$

iii) The σ_f can be estimated using MC again, i.e. :

$$\sigma_f = \frac{1}{M}\sum_{i=1}^{M}\left[f(\boldsymbol{R}_i) - \frac{1}{M}\sum_{j=1}^{M} f(\boldsymbol{R}_j)\right]^2$$

Outline

1) Quantum many body problem;

2) Metropolis Algorithm;

3) Statistical foundations of Monte Carlo methods;

4) Quantum Monte Carlo methods ;
   - 4.1) Overview;
   - 4.2) Variational Quantum Monte Carlo;
   - 4.3) Diffusion Quantum Monte Carlo;
   - 4.4) Typical QMC calculation

4 – Quantum Monte Carlo methods

4.1 - Overview

- **Q**uantum **M**onte **C**arlo simulate the quantum MB problem; (Variational MC; Difusion MC; Path integral MC; Auxiliary field MC; Reptation MC ; Gaussian quantum MC; etc ...)

- **MC** handles directly the many dimention electronic wave function ($R=[r_1,r_2,\ldots,r_N]$)

- We'll cover two main flavors:
  Variational Monte Carlo (integration over (3N)D space);
  Diffusion Monte Carlo (projector approach)

4 – Quantum Monte Carlo methods

4.1 - Overview

- On 55 molecules, mean absolute deviation of atomization energy is 2.9 kcal/mol (MPPT and CI ~ 2kcal/mol ;$O(N^4\text{-}N^6)$)
- Same accuraccy and better scaling than post Hartree-Fock methods (~ N^3). (Non Fixed node DMC scales as e^N)
- Successfully applied to organic molecules, transition metal oxides, solid state silicon, systems up to ~1000 electrons
- Can calculate accurate atomization energies, phase energy differences, excitation energies, one particle densities, correlation functions, etc..
- No free lunch!! Statistical error ~ (Computational time)-1/2

Outline

1) Quantum many body problem;

2) Metropolis Algorithm;

3) Statistical foundations of Monte Carlo methods;

4) Quantum Monte Carlo methods ;
   - 4.1) Overview;
   - 4.2) Variational Quantum Monte Carlo;
   - 4.3) Diffusion Quantum Monte Carlo;
   - 4.4) Typical QMC calculation

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

- Consider: $\mathbf{R}=(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N)$; $\mathbf{r}_i$ position of the i^{th} $\mathbf{e}^-$; and a trial wfn $\Psi_T(\mathbf{R})$ such that:
  i) $\Psi_T(\mathbf{R})$ and $\nabla\Psi_T(\mathbf{R})$ exist and are continous in all* R;
  ii) $\int |\Psi_T|^2 d\,\boldsymbol{R}$ and $\int \Psi_T^* \hat{H} \Psi_T d\,\boldsymbol{R}$ exist;

- The <H> with respect to $\Psi_T(\mathbf{R})$ is:

$$E_V = \left\langle \Psi_T \middle| \hat{H} \middle| \Psi_T \right\rangle = \int \frac{|\Psi_T|^2}{\int d\,\boldsymbol{R} |\Psi_T|^2} E_L(\boldsymbol{R})\, d\,\boldsymbol{R}$$

  where $E_L=\{H\Psi_T \div \Psi_T\}$ and $P(\mathbf{R})=|\Psi_T|^2 \div \int |\Psi_T|^2 d\mathbf{R}$

- Using metropolis MC method we sample $P(\mathbf{R})$ ($\{R_1,\ldots, R_M\}$) and average the E_L to get:

$$E_V = \langle \hat{H} \rangle = \frac{1}{M} \sum_{k=1}^{M} E_L(\boldsymbol{R}_k)$$

- Optimizing σ_T to get acceptance ~ 50% we improve the mixing of the sample;

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

- Consider: $\mathbf{R}=(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N)$; $\mathbf{r}_i$ position of the i^{th} $\mathbf{e}^-$; and a trial wfn $\Psi_T(\mathbf{R})$ such that:
  i) $\Psi_T(\mathbf{R})$ and $\nabla\Psi_T(\mathbf{R})$ exist and are continous in all* R;
  ii) $\int |\Psi_T|^2 d\boldsymbol{R}$ and $\int \Psi_T^* \hat{H} \Psi_T d\boldsymbol{R}$ exist;

- The <H> with respect to $\Psi_T(\mathbf{R})$ is:

$$E_V = \langle \Psi_T | \hat{H} | \Psi_T \rangle = \int \frac{|\Psi_T|^2}{\int d\boldsymbol{R} |\Psi_T|^2} E_L(\boldsymbol{R}) d\boldsymbol{R}$$

  where $E_L = \{H\Psi_T \div \Psi_T\}$ and $P(\mathbf{R}) = |\Psi_T|^2 \div \int |\Psi_T|^2 d\mathbf{R}$

- Using metropolis MC method we sample $P(\mathbf{R})$ ($\{R_1, \ldots, R_M\}$) and average the E_L to get:

$$E_V = \langle \hat{H} \rangle = \frac{1}{M} \sum_{k=1}^{M} E_L(\boldsymbol{R}_k)$$

- Optimizing σ_T to get acceptance ~ 50% we improve the mixing of the sample;

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

- Consider: $\mathbf{R}=(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N)$; $\mathbf{r}_i$ position of the i^{th} $\mathbf{e}^-$; and a trial wfn $\Psi_T(\mathbf{R})$ such that:
  i) $\Psi_T(\mathbf{R})$ and $\nabla\Psi_T(\mathbf{R})$ exist and are continous in all* R;
  ii) $\int |\Psi_T|^2 d\,\boldsymbol{R}$ and $\int \Psi_T^* \hat{H} \Psi_T d\,\boldsymbol{R}$ exist;

- The <H> with respect to $\Psi_T(\mathbf{R})$ is:

$$E_V = \langle \Psi_T | \hat{H} | \Psi_T \rangle = \int \frac{|\Psi_T|^2}{\int d\,\boldsymbol{R} |\Psi_T|^2} E_L(\boldsymbol{R})\, d\,\boldsymbol{R}$$

  where $E_L = \{H\Psi_T \div \Psi_T\}$ and $P(\mathbf{R}) = |\Psi_T|^2 \div \int |\Psi_T|^2 d\mathbf{R}$

- Using metropolis MC method we sample $P(\mathbf{R})$ ($\{R_1, \ldots, R_M\}$) and average the E_L to get:

$$E_V = \langle \hat{H} \rangle = \frac{1}{M} \sum_{k=1}^{M} E_L(\boldsymbol{R}_k)$$

- Optimizing σ_T to get acceptance ~ 50% we improve the mixing of the sample;

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

- Consider: $\mathbf{R}=(\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_N)$; $\mathbf{r}_i$ position of the i^{th} $\mathbf{e}^-$; and a trial wfn $\Psi_T(\mathbf{R})$ such that:
  i) $\Psi_T(\mathbf{R})$ and $\nabla\Psi_T(\mathbf{R})$ exist and are continous in all* R;
  ii) $\int |\Psi_T|^2 d\,\boldsymbol{R}$ and $\int \Psi_T^* \hat{H} \Psi_T d\,\boldsymbol{R}$ exist;

- The <H> with respect to $\Psi_T(\mathbf{R})$ is:

$$E_V = \langle \Psi_T | \hat{H} | \Psi_T \rangle = \int \frac{|\Psi_T|^2}{\int d\,\boldsymbol{R} |\Psi_T|^2} E_L(\boldsymbol{R})\, d\,\boldsymbol{R}$$

  where $E_L = \{H\Psi_T \div \Psi_T\}$ and $P(\mathbf{R}) = |\Psi_T|^2 \div \int |\Psi_T|^2 d\mathbf{R}$

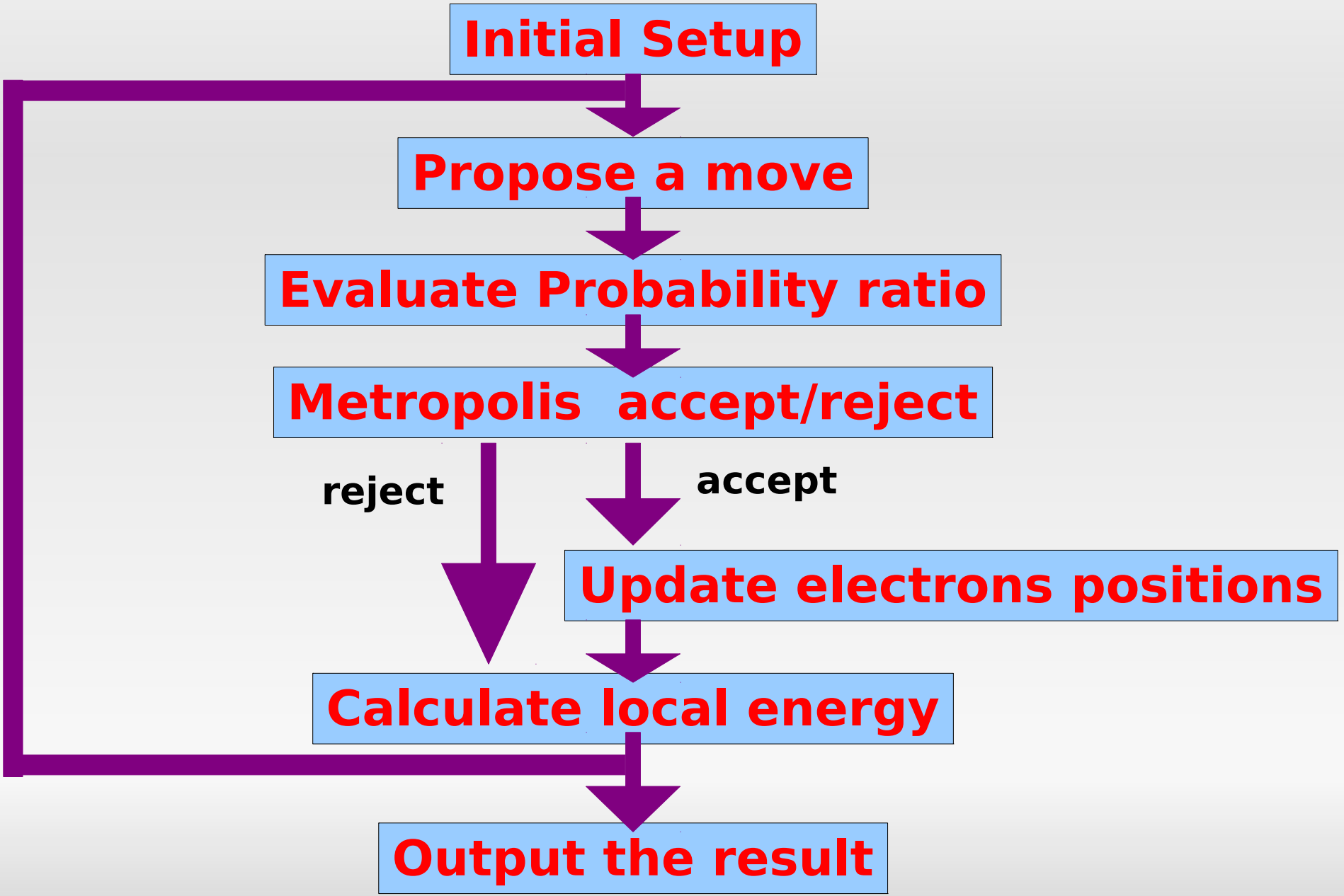
- Using metropolis MC method we sample $P(\mathbf{R})$ ($\{R_1, \ldots, R_M\}$) and average the E_L to get:

$$E_V = \langle \hat{H} \rangle = \frac{1}{M} \sum_{k=1}^{M} E_L(\boldsymbol{R}_k)$$

- Optimizing σ_T to get acceptance ~ 50% we improve the mixing of the sample;

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo



4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

" How many graduate students lives have been lost optimizing wavefuctions ... " D. Ceperley

- Good news !!! Ψ_T can have any functional form !!!

- Most commonly used Ψ_T is a Slater-Jastrow wavefunction:

$$\Psi(\boldsymbol{X}) = e^{J(\boldsymbol{X};\alpha_{1,..},\alpha_i)} \left\{ \sum_{j=1} \beta_j D_j(\boldsymbol{X}) \right\}$$

> $\boldsymbol{X} = (\boldsymbol{x}_{1,} \ldots, \boldsymbol{X}_N)$, with $\boldsymbol{x}_i = \{\boldsymbol{r}_i, \sigma_i\}$

> β_j α_i , coefficients to be optimized ;

> $$D_n(\boldsymbol{X}) = \begin{vmatrix} \phi_1(\boldsymbol{x}_1) & \cdots & \phi_1(\boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ \phi_N(\boldsymbol{x}_1) & \cdots & \phi_N(\boldsymbol{x}_N) \end{vmatrix}$$ $\phi_h(\boldsymbol{x}_g)$ are single particle Slater orbs.

> $e^{J(\boldsymbol{X};\alpha_{1,..},\alpha_i)}$, is the Jastrow correlation function;

> $J = F(d_\gamma, r_i, r_{ij})$, the Jastrow factor: usually sums of Chebyshev polynomials

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

- How do we acctually perform the optimization of the wfn ? (minimize $\sigma_E(\alpha)$ or E_v or a mixing of both)

- The most used is the minimization of $\sigma_E(\alpha)$, because:

  i) We know it's exact value for the g.s.
  ii) Better numerical stability;

- The variance of the energy is given by:

$$\sigma_E^2(\alpha) = \int \frac{|\Psi(\alpha)|^2}{\int |\Psi(\alpha)|^2 d\boldsymbol{R}} \left[E_L(\alpha) - E_V(\alpha) \right]^2 d\boldsymbol{R}$$

  where: α parameters to optimize ; E_v variational energy, E_L local energy.

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

Correlated sampling method

1) Start form a guessed set of paramters $\{\alpha^1_T, ..., \alpha^\kappa_T\}$;
2) Sample the P($\mathbf{R}$,α^κ_T) using MMC method; $P(\boldsymbol{R}) = \frac{|\Psi(\boldsymbol{R},\alpha_T)|^2}{\int |\Psi(\boldsymbol{R},\alpha_T)|^2 d\boldsymbol{R}}$
3) With this sampling minimize $\sigma_E(\alpha)$ like this :

Calculate E_L ; E_V and $\sigma_E(\alpha)$, for a different set of parameters $\{\alpha^1_N, ..., \alpha^\kappa_N\}$ (choosen in a way to minimize $\sigma_E(\alpha)$), like this:

$$E_L(\boldsymbol{R}_i,\alpha_N) = \frac{\hat{H}\Psi(\boldsymbol{R}_i,\alpha_N)}{\Psi(\boldsymbol{R}_i,\alpha_N)} \quad ; \quad \omega(\alpha_N) = \frac{\Psi(\alpha_N)}{\Psi(\alpha_T)}$$

$$E_V = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} E_L(\alpha_N) d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) E_L(R_i,\alpha_N)$$

$$\sigma^2_E(\alpha_N) = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} \left[E_L(\alpha_N) - E_V(\alpha_N)\right]^2 d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) \left[E_L(R_i,\alpha_N) - E_V\right]^2$$

4) Once σ_E reaches a minimum, set $\{\alpha^1_T, ..., \alpha^\kappa_T\} = \{\alpha^1_N, ..., \alpha^\kappa_N\}$
5) Repeat all the steps 2-5 until $\sigma_E(\alpha) \sim 0$

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

Correlated sampling method

1) Start form a guessed set of paramters $\{\alpha^1_T, ..., \alpha^\kappa_T\}$;
2) Sample the $P(\mathbf{R},\alpha^\kappa_T)$ using MMC method; $P(\boldsymbol{R}) = \frac{|\Psi(\boldsymbol{R},\alpha_T)|^2}{\int |\Psi(\boldsymbol{R},\alpha_T)|^2 d\boldsymbol{R}}$
3) With this sampling minimize $\sigma_E(\alpha)$ like this :

Calculate E_L ; E_V and $\sigma_E(\alpha)$, for a different set of parameters $\{\alpha^1_N, ..., \alpha^\kappa_N\}$ (choosen in a way to minimize $\sigma_E(\alpha)$), like this:

$$E_L(\boldsymbol{R}_i,\alpha_N) = \frac{\hat{H}\Psi(\boldsymbol{R}_i,\alpha_N)}{\Psi(\boldsymbol{R}_i,\alpha_N)} \quad ; \quad \omega(\alpha_N) = \frac{\Psi(\alpha_N)}{\Psi(\alpha_T)}$$

$$E_V = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} E_L(\alpha_N) d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) E_L(R_i,\alpha_N)$$

$$\sigma^2_E(\alpha_N) = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} \left[E_L(\alpha_N) - E_V(\alpha_N)\right]^2 d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) \left[E_L(R_i,\alpha_N) - E_V\right]^2$$

4) Once σ_E reaches a minimum, set $\{\alpha^1_T, ..., \alpha^\kappa_T\} = \{\alpha^1_N, ..., \alpha^\kappa_N\}$
5) Repeat all the steps 2-5 until $\sigma_E(\alpha) \sim 0$

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

Correlated sampling method

1) Start form a guessed set of paramters $\{\alpha^1_T, ..., \alpha^\kappa_T\}$;
2) Sample the P($\mathbf{R}$,α^κ_T) using MMC method; $P(\boldsymbol{R}) = \frac{|\Psi(\boldsymbol{R},\alpha_T)|^2}{\int |\Psi(\boldsymbol{R},\alpha_T)|^2 d\boldsymbol{R}}$
3) With this sampling minimize $\sigma_E(\alpha)$ like this :

Calculate E_L ; E_V and $\sigma_E(\alpha)$, for a different set of parameters $\{\alpha^1_N, ..., \alpha^\kappa_N\}$ (choosen in a way to minimize $\sigma_E(\alpha)$), like this:

$$E_L(\boldsymbol{R}_i,\alpha_N) = \frac{\hat{H}\Psi(\boldsymbol{R}_i,\alpha_N)}{\Psi(\boldsymbol{R}_i,\alpha_N)} \quad ; \quad \omega(\alpha_N) = \frac{\Psi(\alpha_N)}{\Psi(\alpha_T)}$$

$$E_V = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} E_L(\alpha_N) d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) E_L(R_i,\alpha_N)$$

$$\sigma^2_E(\alpha_N) = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} \left[E_L(\alpha_N) - E_V(\alpha_N)\right]^2 d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) \left[E_L(R_i,\alpha_N) - E_V\right]^2$$

4) Once σ_E reaches a minimum, set $\{\alpha^1_T, ..., \alpha^\kappa_T\} = \{\alpha^1_N, ..., \alpha^\kappa_N\}$
5) Repeat all the steps 2-5 until $\sigma_E(\alpha) \sim 0$

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

Correlated sampling method

1) Start form a guessed set of paramters $\{\alpha^1_T, ..., \alpha^\kappa_T\}$;

2) Sample the P($\mathbf{R}$,α^κ_T) using MMC method; $P(\boldsymbol{R}) = \frac{|\Psi(\boldsymbol{R},\alpha_T)|^2}{\int |\Psi(\boldsymbol{R},\alpha_T)|^2 d\boldsymbol{R}}$

3) With this sampling minimize $\sigma_E(\alpha)$ like this :

Calculate E_L ; E_V and $\sigma_E(\alpha)$, for a different set of parameters $\{\alpha^1_N, ..., \alpha^\kappa_N\}$ (choosen in a way to minimize $\sigma_E(\alpha)$), like this:

$$E_L(\boldsymbol{R}_i,\alpha_N) = \frac{\hat{H}\Psi(\boldsymbol{R}_i,\alpha_N)}{\Psi(\boldsymbol{R}_i,\alpha_N)} \quad ; \quad \omega(\alpha_N) = \frac{\Psi(\alpha_N)}{\Psi(\alpha_T)}$$

$$E_V = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} E_L(\alpha_N) d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) E_L(R_i,\alpha_N)$$

$$\sigma^2_E(\alpha_N) = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} \left[E_L(\alpha_N) - E_V(\alpha_N)\right]^2 d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) \left[E_L(R_i,\alpha_N) - E_V\right]^2$$

4) Once σ_E reaches a minimum, set $\{\alpha^1_T, ..., \alpha^\kappa_T\} = \{\alpha^1_N, ..., \alpha^\kappa_N\}$

5) Repeat all the steps 2-5 until $\sigma_E(\alpha) \sim 0$

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

Correlated sampling method

1) Start form a guessed set of paramters $\{\alpha^1_T, ..., \alpha^\kappa_T\}$;
2) Sample the $P(\mathbf{R},\alpha^\kappa_T)$ using MMC method; $P(\boldsymbol{R}) = \frac{|\Psi(\boldsymbol{R},\alpha_T)|^2}{\int |\Psi(\boldsymbol{R},\alpha_T)|^2 d\boldsymbol{R}}$
3) With this sampling minimize $\sigma_E(\alpha)$ like this :

Calculate E_L ; E_V and $\sigma_E(\alpha)$, for a different set of parameters $\{\alpha^1_N, ..., \alpha^\kappa_N\}$ (choosen in a way to minimize $\sigma_E(\alpha)$), like this:

$$E_L(\boldsymbol{R}_i,\alpha_N) = \frac{\hat{H}\Psi(\boldsymbol{R}_i,\alpha_N)}{\Psi(\boldsymbol{R}_i,\alpha_N)} \quad ; \quad \omega(\alpha_N) = \frac{\Psi(\alpha_N)}{\Psi(\alpha_T)}$$

$$E_V = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} E_L(\alpha_N) d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) E_L(R_i,\alpha_N)$$

$$\sigma^2_E(\alpha_N) = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} \left[E_L(\alpha_N) - E_V(\alpha_N)\right]^2 d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) \left[E_L(R_i,\alpha_N) - E_V\right]^2$$

4) Once σ_E reaches a minimum, set $\{\alpha^1_T, ..., \alpha^\kappa_T\} = \{\alpha^1_N, ..., \alpha^\kappa_N\}$
5) Repeat all the steps 2-5 until $\sigma_E(\alpha) \sim 0$

4 – Quantum Monte Carlo methods

4.2 – Variational Quantum Monte Carlo

Correlated sampling method

1) Start form a guessed set of paramters $\{\alpha^1_T, ..., \alpha^\kappa_T\}$;
2) Sample the P($\mathbf{R}$,α^κ_T) using MMC method; $P(\boldsymbol{R}) = \frac{|\Psi(\boldsymbol{R},\alpha_T)|^2}{\int |\Psi(\boldsymbol{R},\alpha_T)|^2 d\boldsymbol{R}}$
3) With this sampling minimize $\sigma_E(\alpha)$ like this :

Calculate E_L ; E_V and $\sigma_E(\alpha)$, for a different set of parameters $\{\alpha^1_N, ..., \alpha^\kappa_N\}$ (choosen in a way to minimize $\sigma_E(\alpha)$), like this:

$$E_L(\boldsymbol{R}_i,\alpha_N) = \frac{\hat{H}\Psi(\boldsymbol{R}_i,\alpha_N)}{\Psi(\boldsymbol{R}_i,\alpha_N)} \quad ; \quad \omega(\alpha_N) = \frac{\Psi(\alpha_N)}{\Psi(\alpha_T)}$$

$$E_V = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} E_L(\alpha_N) d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) E_L(R_i,\alpha_N)$$

$$\sigma^2_E(\alpha_N) = \int \frac{|\Psi(\alpha_T)|^2 \omega(\alpha_N)}{\int |\Psi(\alpha_T)|^2 \omega(\alpha_N) d\boldsymbol{R}} \left[E_L(\alpha_N) - E_V(\alpha_N)\right]^2 d\boldsymbol{R} \approx \frac{1}{\sum_{k=1}^{M} \omega(R_k,\alpha_N)} \sum_{i=1}^{M} \omega(R_i,\alpha_N) \left[E_L(R_i,\alpha_N) - E_V\right]^2$$

4) Once σ_E reaches a minimum, set $\{\alpha^1_T, ..., \alpha^\kappa_T\} = \{\alpha^1_N, ..., \alpha^\kappa_N\}$
5) Repeat all the steps 2-5 until $\sigma_E(\alpha) \sim 0$

Outline

1) Quantum many body problem;

2) Metropolis Algorithm;

3) Statistical foundations of Monte Carlo methods;

4) Quantum Monte Carlo methods ;
   - 4.1) Overview;
   - 4.2) Variational Quantum Monte Carlo;
   - 4.3) Diffusion Quantum Monte Carlo;
   - 4.4) Typical QMC calculation;

4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

- General strategy: stochastically simulate a differential equation that converges to the eigenstate

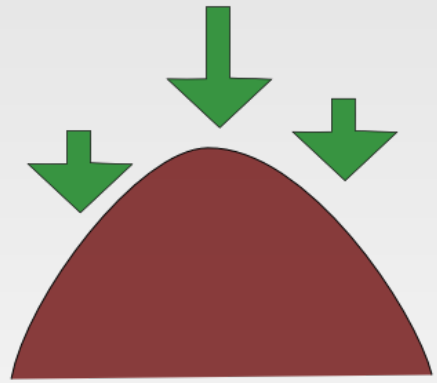
- Equation: $-\frac{d\Psi(\boldsymbol{R},t)}{d\mathrm{t}}=(\hat{H}-E)\Psi(\boldsymbol{R},t)$

- Must propagate an entire function forward in time <=> distribution of walkers

4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

- We want to find a wave function so $H\Psi = E\Psi$
- Our differential equation $-\frac{d\Psi(\boldsymbol{R},t)}{d\,\mathrm{t}} = (\hat{H}-E)\Psi(\boldsymbol{R},t)$
- Suppose that $\left(-\frac{1}{2}\nabla^2 + V(R)\right)\Psi > E\Psi$
- $|\Psi|$ decreases
- Kinetic energy (curvature) decreases, potential energy stays the same
- Time derivative is zero when

$$H\Psi = E\Psi$$



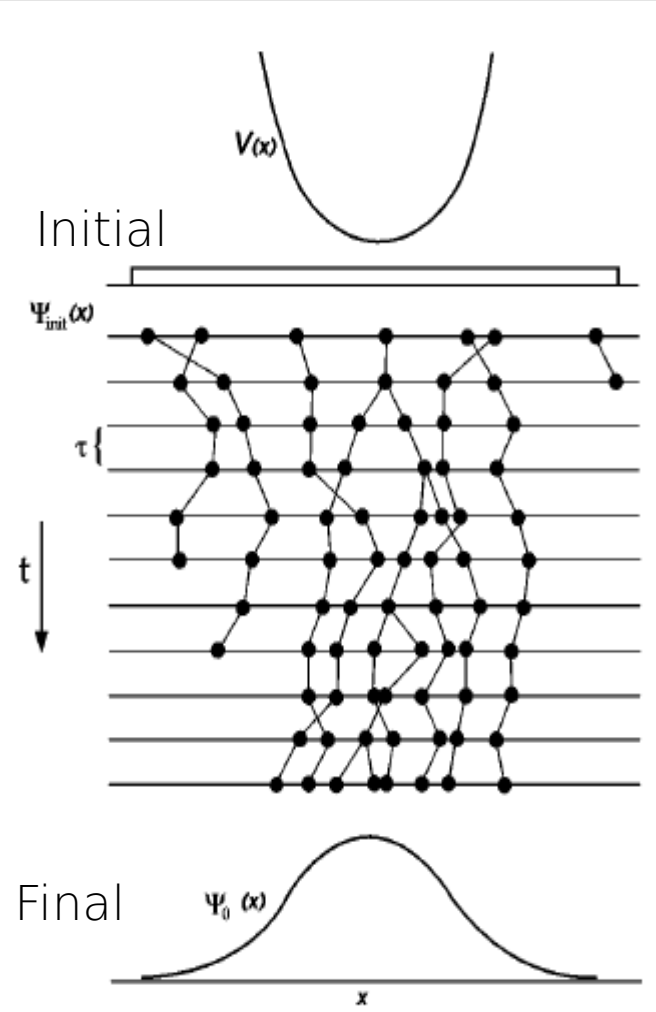
t=0

t=infinity

4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

$$-\frac{d\Psi(\boldsymbol{R},t)}{dt} = \underbrace{-\frac{1}{2}\nabla^2\Psi(\boldsymbol{R})}_{\text{Diffusion}} + \underbrace{(V(\boldsymbol{R})-E)\Psi(\boldsymbol{R},t)}_{\text{Birth/death}}$$

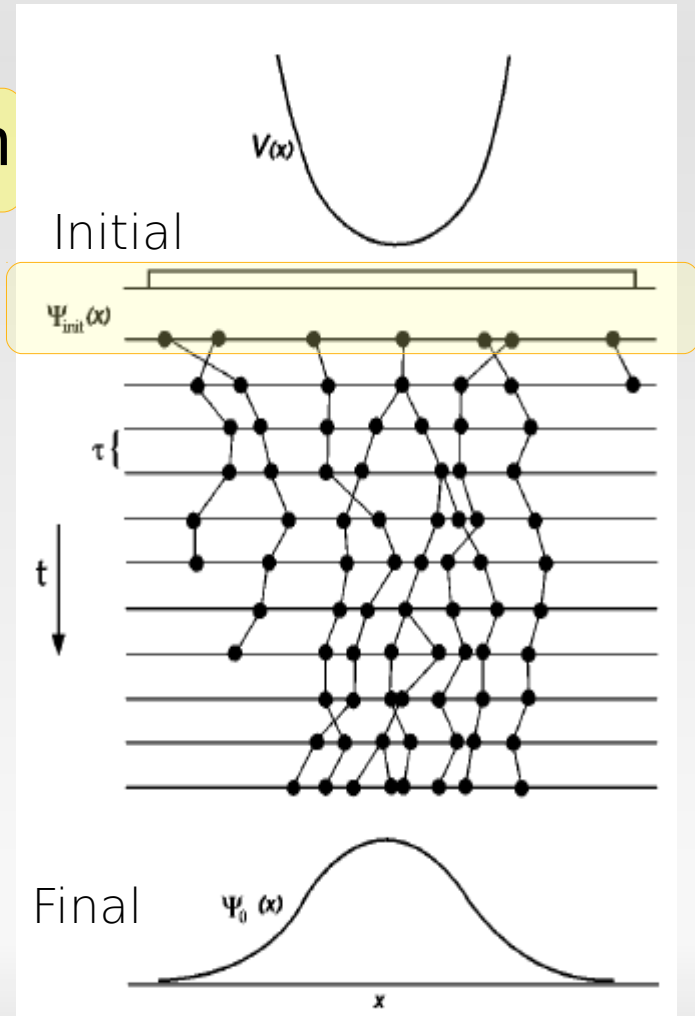
- Generate walkers with a guess distribution
- Each time step:
  - Take a random step (diffuse)
  - A walker can either die, give birth, or just keep going
- Keep following rules, and we find the ground state!
- Works in an arbitrary number of dimensions



4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

$$-\frac{d\Psi(\boldsymbol{R},t)}{dt} = \underbrace{-\frac{1}{2}\nabla^2\Psi(\boldsymbol{R})}_{\text{Diffusion}} + \underbrace{(V(\boldsymbol{R})-E)\Psi(\boldsymbol{R},t)}_{\text{Birth/death}}$$

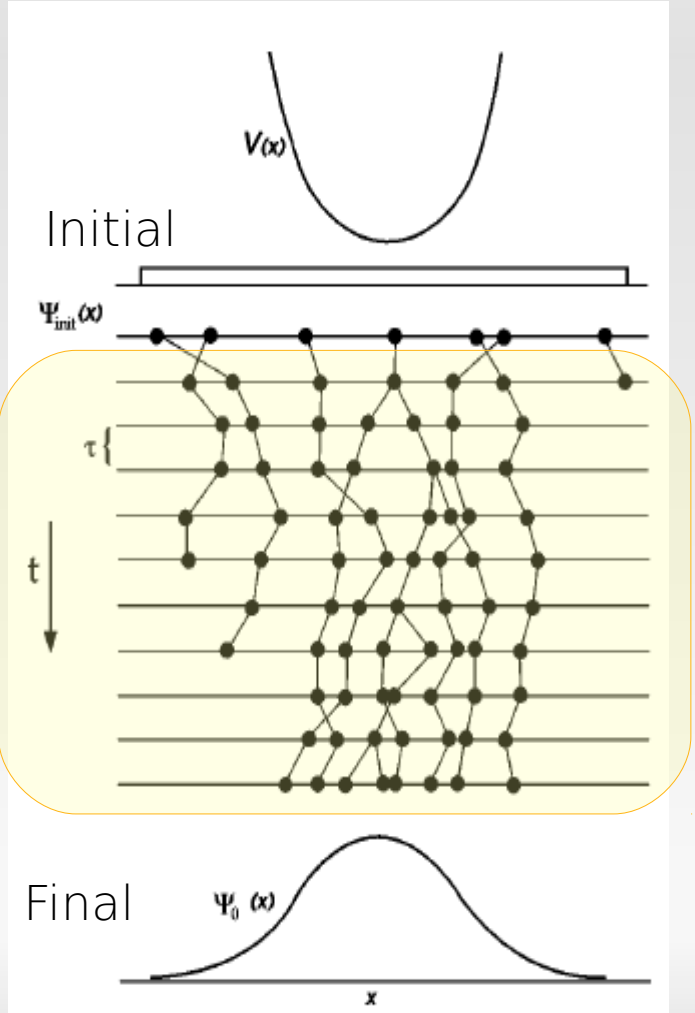
- Generate walkers with a guess distribution
- Each time step:
  - Take a random step (diffuse)
  - A walker can either die, give birth, or just keep going
- Keep following rules, and we find the ground state!
- Works in an arbitrary number of dimensions



4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

$$-\frac{d\Psi(\boldsymbol{R},t)}{dt} = \underbrace{-\frac{1}{2}\nabla^2\Psi(\boldsymbol{R})}_{\text{Diffusion}} + \underbrace{(V(\boldsymbol{R})-E)\Psi(\boldsymbol{R},t)}_{\text{Birth/death}}$$

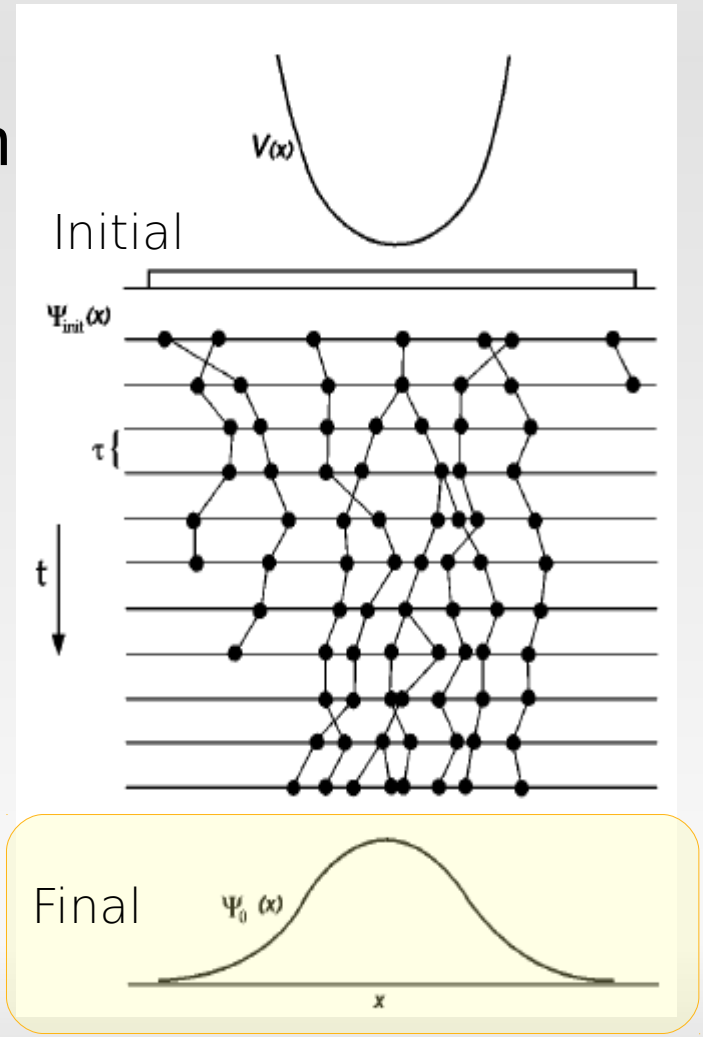
- Generate walkers with a guess distribution
- Each time step:
  - Take a random step (diffuse)
  - A walker can either die, give birth, or just keep going
- Keep following rules, and we find the ground state!
- Works in an arbitrary number of dimensions



4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

$$-\frac{d\Psi(\boldsymbol{R},t)}{dt} = \underbrace{-\frac{1}{2}\nabla^2\Psi(\boldsymbol{R})}_{\text{Diffusion}} + \underbrace{(V(\boldsymbol{R})-E)\Psi(\boldsymbol{R},t)}_{\text{Birth/death}}$$

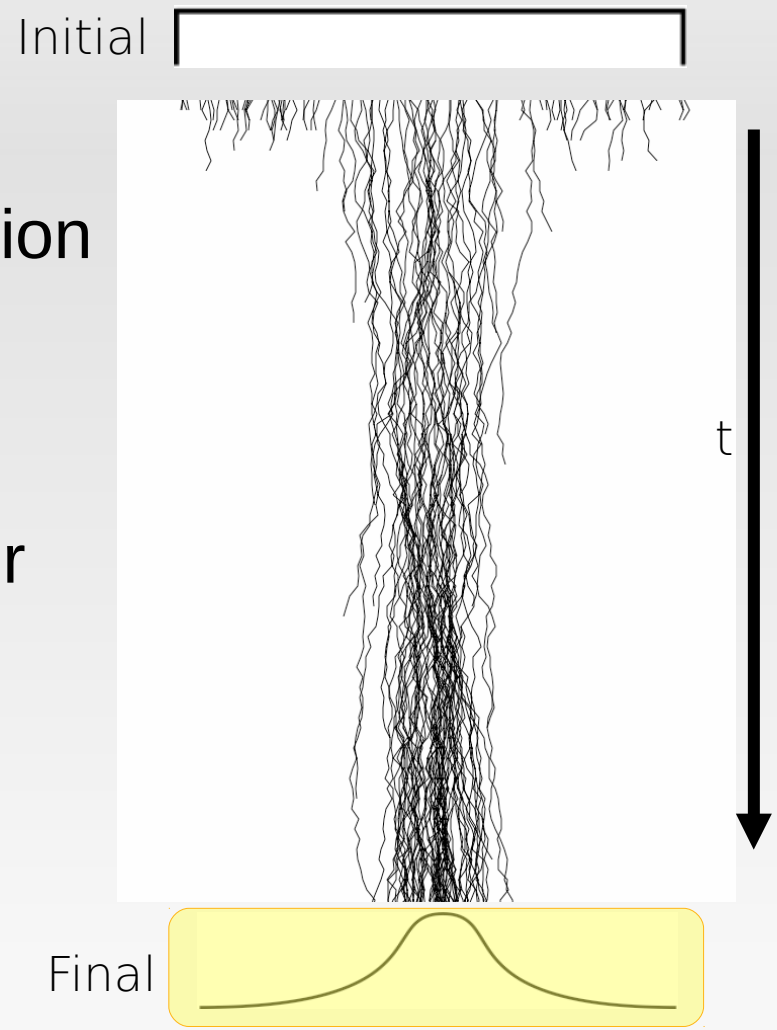
- Generate walkers with a guess distribution
- Each time step:
  - Take a random step (diffuse)
  - A walker can either die, give birth, or just keep going
- Keep following rules, and we find the ground state!
- Works in an arbitrary number of dimensions



4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

$$-\frac{d\Psi(\boldsymbol{R},t)}{dt} = \underbrace{-\frac{1}{2}\nabla^2\Psi(\boldsymbol{R})}_{\text{Diffusion}} + \underbrace{(V(\boldsymbol{R})-E)\Psi(\boldsymbol{R},t)}_{\text{Birth/death}}$$

- Generate walkers with a guess distribution
- Each time step:
  - Take a random step (diffuse)
  - A walker can either die, give birth, or just keep going
- Keep following rules, and we find the ground state!
- Works in an arbitrary number of dimensions



4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

- Importance sampling: multiply the differential equation by a trial wave function
  - Converges to $\Psi_T \Phi_0$ instead of Φ_0
  - The better the trial function, the faster DMC is-- feed it a wave function from VMC

- Fixed node approximation: for fermions, ground state has negative and positive parts
  - Not a pdf--can't sample it
  - Approximation: $\Psi_T \Phi_0 > 0$

4 – Quantum Monte Carlo methods
4.3 – Diffusion Quantum Monte Carlo

- Importance sampling: multiply the differential equation by a trial wave function
  - Converges to $\Psi_T \Phi_0$ instead of Φ_0
  - The better the trial function, the faster DMC is-- feed it a wave function from VMC

- Fixed node approximation: for fermions, ground state has negative and positive parts
  - Not a pdf--can't sample it
  - Approximation: $\Psi_T \Phi_0 > 0$

Outline

1) Quantum many body problem;

2) Metropolis Algorithm;

3) Statistical foundations of Monte Carlo methods;

4) Quantum Monte Carlo methods ;
   - 4.1) Overview;
   - 4.2) Variational Quantum Monte Carlo;
   - 4.3) Diffusion Quantum Monte Carlo;
   - 4.4) Typical QMC calculation;

4 – Quantum Monte Carlo methods
4.4 – Typical QMC calculation

- Choose system and get one-particle orbitals
- Optimize wave function using VMC, evaluate energy and properties of wave function
- Use optimized wave function in DMC for most accurate, lowest energy calculations

4 – Quantum Monte Carlo methods
4.4 – Typical QMC calculation

- Choose system and get one-particle orbitals
- Optimize wave function using VMC, evaluate energy and properties of wave function
- Use optimized wave function in DMC for most accurate, lowest energy calculations

4 – Quantum Monte Carlo methods
4.4 – Typical QMC calculation

- Choose system and get one-particle orbitals
- Optimize wave function using VMC, evaluate energy and properties of wave function
- Use optimized wave function in DMC for most accurate, lowest energy calculations