PHY604 Lecture 23

November 13, 2025

Today’s lecture:
QMC and Machine learning

- Variational and Diffusion Quantum Monte Carlo

- Intro to Machine learning

Quantum Monte Carlo (Pang Sec. 10.5)

- So far, we have studied classical systems
- Monte Carlo algorithm can be generalized to study quantum systems
- Most direct generalization of the Metropolis algorithm: Variational quantum Monte Carlo
- We will just introduce some basic concepts in QMC and show how what we learned on classical systems transfers

General many-body quantum problem:

- We are seeking approximate solutions of the Hamiltonian:

$$H = \sum_{i=1}^{N} \left[-\frac{\hbar^2}{2m_i} \nabla_i^2 + U_{\text{ext}}(\mathbf{r}_i) \right] + \sum_{i>j} V(\mathbf{r}_i, \mathbf{r}_j)$$

Kinetic energy

External potential

Electron-electron interaction

General many-body quantum problem:

- We are seeking approximate solutions of the Hamiltonian:

$$H = \sum_{i=1}^{N} \left[-\frac{\hbar^2}{2m_i} \nabla_i^2 + U_{\text{ext}}(\mathbf{r}_i) \right] + \sum_{i>j} V(\mathbf{r}_i, \mathbf{r}_j)$$

- Time-independent many-body Schrödinger equation

$$H\Psi_n(\mathbf{R}) = E_n \Psi_n(\mathbf{R})$$

- **R**=($\mathbf{r}_1$,$\mathbf{r}_2$,…,$\mathbf{r}_N$) is the positions of all particles

- In general, cannot obtain analytic solution for more than two particles
- Numerically exact solutions are also limited to few particles

The many-body ground state

- Often, we would like to study the ground state of the system
- In that case, we can make use of the variational principle
  - Any other state has higher energy than the ground state
  - Introduce a trial state Φ to approximate the ground state, and minimize with respect to some set of parameters α_i

$$E[\alpha_i] = \frac{\langle \Phi | H | \Phi \rangle}{\langle \Phi | \Phi \rangle} \geq E_0$$

- Minimize by taking parameters in the Euler-Lagrange equation

$$\frac{\delta E[\alpha_i]}{\delta \alpha_i} = 0$$

Variational minimization of many-body ground state

- We can write:

$$E[\alpha_i] = \frac{\int \Phi^\dagger(\mathbf{R}) H \Phi(\mathbf{R}) d\mathbf{R}}{\int |\Phi(\mathbf{R}')|^2 d\mathbf{R}'} \equiv \int \mathcal{W}(\mathbf{R}) \mathcal{E}(\mathbf{R}) d\mathbf{R}$$

- Where:

$$\mathcal{W}(\mathbf{R}) = \frac{|\Phi(\mathbf{R})|^2}{\int |\Phi(\mathbf{R}')|^2 d\mathbf{R}'}, \qquad \mathcal{E}(\mathbf{R}) = \frac{1}{\Phi(\mathbf{R})} H \Phi(\mathbf{R})$$

Distribution function

Local energy of specific **R**

Variational minimization of many-body ground state

- We can write:

$$E[\alpha_i] = \frac{\int \Phi^\dagger(\mathbf{R}) H \Phi(\mathbf{R}) d\mathbf{R}}{\int |\Phi(\mathbf{R}')|^2 d\mathbf{R}'} \equiv \int \mathcal{W}(\mathbf{R}) \mathcal{E}(\mathbf{R}) d\mathbf{R}$$

- Where:

$$\mathcal{W}(\mathbf{R}) = \frac{|\Phi(\mathbf{R})|^2}{\int |\Phi(\mathbf{R}')|^2 d\mathbf{R}'}, \qquad \mathcal{E}(\mathbf{R}) = \frac{1}{\Phi(\mathbf{R})} H \Phi(\mathbf{R})$$

- If we know these, we can evaluate the expression via Monte Carlo
- Then vary α_i to minimize $E[\alpha_i]$

The trial wavefunction

- Common choice for trial wavefunctions:

$$\Phi(\mathbf{R}) = D(\mathbf{R})e^{-U(\mathbf{R})}$$

- $D(\mathbf{R})$ is a constant for bosons and a Slater determinant of single-particle orbitals for fermion systems
- $U(\mathbf{R})$ is ”Jastrow factor”:

$$U(\mathbf{R}) = \sum_{i=1}^{N} u_i(\mathbf{r}_i) + \sum_{i>j}^{N} u_2(\mathbf{r}_i, \mathbf{r}_j) + \dots$$

Comes from the interaction of particles with external potential

Comes from the interparticle interactions

The trial wavefunction

- Common choice for trial wavefunctions:

$$\Phi(\mathbf{R}) = D(\mathbf{R})e^{-U(\mathbf{R})}$$

- *D*(**R**) is a constant for bosons and a Slater determinant of single-particle orbitals for fermion systems
- *U*(**R**) is ”Jastrow factor”:

$$U(\mathbf{R}) = \sum_{i=1}^{N} u_i(\mathbf{r}_i) + \sum_{i>j}^{N} u_2(\mathbf{r}_i, \mathbf{r}_j) + \ldots$$

- The key to the method is to choose a trial wavefunction that contains the necessary physics

Procedure for variational QMC

- 1. Choose a basis of single-particle orbitals
- 2. For fermions, construct Slater determinant (by a linear combination of atomic orbitals, or a by single-particle method like Hartree-Fock or DFT)
- 3. Determine the interparticle interactions
- 4. Perform Metropolis steps, e.g., by altering particle positions $\mathbf{r}_i$
- 5. Use as the probability in the Markov chain: $\mathcal{W}(\mathbf{R}) = \dfrac{|\Phi(\mathbf{R})|^2}{\int |\Phi(\mathbf{R}')|^2 d\mathbf{R}'}$
- 6. Accumulate average energy via local energy: $E = \dfrac{1}{M}\displaystyle\sum_{m=1}^{M} \mathcal{E}(\mathbf{R}_m)$

Diffusion Monte Carlo with Green's functions

- Variational Monte Carlo limited by trial wave function. Can we go beyond this to find the exact ground state?
- Diffusion Monte Carlo: Treat the ground state of the Schrödinger equation as the stationary solution of a diffusion equation
  - We used this approach before for solving the Poisson equation
- Diffusion equation in this case is "imaginary time Schrödinger equation":

$$\frac{\partial \Phi(\mathbf{R}, t)}{\partial t} = -(H - E_c)\Phi(\mathbf{R}, t)$$

- E_c is adjustable energy offset

Green's function:

- At a later time, the wave function is:

$$\Phi(\mathbf{R}, t+\tau) = \int G(\mathbf{R}, \mathbf{R}'; \tau)\Phi(\mathbf{R}', t)d\mathbf{R}'$$

- Where *G* is the Green's function, obeys the same equation of the wavefunctions (with a delta function initial condition):

$$\left[\frac{\partial}{\partial t} - (H - E_c)\right] G(\mathbf{R}, \mathbf{R}'; \tau) = \delta(\mathbf{R} - \mathbf{R}')\delta(\tau)$$

- Or:

$$G(\mathbf{R}, \mathbf{R}'; \tau) = \langle \mathbf{R} | \exp[-\tau(H - E_c)] | \mathbf{R}' \rangle$$

Projecting onto the ground state

- We now use the expansion in terms of eigenfunctions of *H,* Ψ_{i} with eigenvalues E_i:

$$\exp(-\tau H) = \sum_i |\Psi_i\rangle \exp(-\tau E_i)\langle\Psi_i|$$

- The the Green's function can be written:

$$G(\mathbf{R}, \mathbf{R}'; \tau) = \sum_i \Psi_i(\mathbf{R}) \exp[-\tau(E_i - E_c)]\Psi_i^*(\mathbf{R}')$$

- Choose an initial state, e.g., the trial wavefunction from variational QMC:

$$\Phi(\mathbf{R}, 0) = \Phi_{\text{init}}(\mathbf{R})$$

Projecting onto the ground state

- Now we take τ to infinity:

$$\lim_{\tau\to\infty}\langle\mathbf{R}|e^{-\tau(H-E_c)}|\Phi_{\text{init}}\rangle$$

$$= \lim_{\tau\to\infty}\int G(\mathbf{R},\mathbf{R}';\tau)\Phi_{\text{init}}(\mathbf{R}')d\mathbf{R}'$$

$$= \lim_{\tau\to\infty}\sum_i \Psi_i(\mathbf{R})e^{-\tau(E_i-E_c)}\langle\Psi_i|\Phi_{\text{init}}\rangle$$

$$= \lim_{\tau\to\infty}\left(\Psi_0(\mathbf{R})e^{-\tau(E_0-E_c)}\langle\Psi_0|\Phi_{\text{init}}\rangle + \sum_{i=1}\Psi_i(\mathbf{R})e^{-\tau(E_i-E_c)}\langle\Psi_i|\Phi_{\text{init}}\rangle\right)$$

- Adjusting E_c to E_0 suppresses the second term, giving us the ground state
  - As long as there is some overlap with our initial state

Wait, what is the Green's function?

$$G(\mathbf{R},\mathbf{R}';\tau) = \sum_i \Psi_i(\mathbf{R}) \exp[-\tau(E_i - E_c)]\Psi_i^*(\mathbf{R}')$$

- We don't know the eigenstates/eigenvalues *a priori*
- Let consider the the imaginary time Schrödinger equation with no potential

$$\frac{\partial \Phi(\mathbf{R},t)}{\partial t} = \frac{1}{2}\nabla^2\Phi(\mathbf{R},t)$$

- We have solved this problem before with a delta function initial condition! Just the linear diffusion equation.
  - At later times it is a Gaussian
  - In this case, we are in 3*N* dimensions

$$G_d(\mathbf{R},\mathbf{R}';\tau) = (2\pi\tau)^{-3N/2} \exp\left[-\frac{|\mathbf{R}-\mathbf{R}'|^2}{2\tau}\right]$$

Suzuki-Trotter decomposition and birth/death

- Can show that the effect of the potential on the Green's function is approximately (for small τ):

$$G(\mathbf{R},\mathbf{R}';\tau) \simeq (2\pi\tau)^{-3N/2} \exp\left[-\frac{|\mathbf{R}-\mathbf{R}'|^2}{2\tau}\right] \exp\left[-\tau\frac{V(\mathbf{R})+V(\mathbf{R}')-2E_c}{2}\right]$$

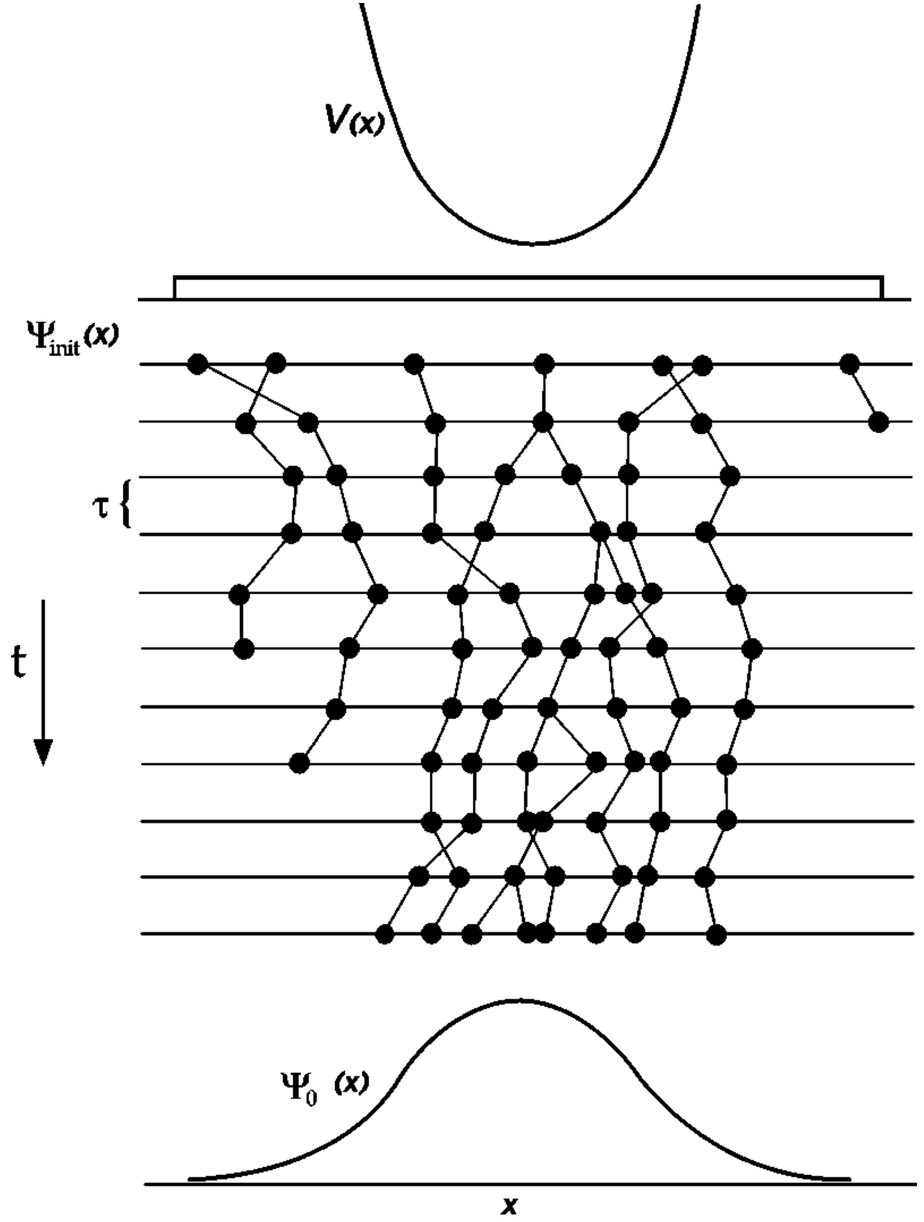
- Can treat the factor:

$$P = \exp\left[-\tau\frac{V(\mathbf{R})+V(\mathbf{R}')-2E_c}{2}\right]$$

- As a reweighting of the free-particle Green's function
- Can be used to kill random walkers that enter high-potential areas of Hilbert space (see next slide)

Markov chain for QMC

- To do the Metropolis algorithm, we first create an ensemble of independent configurations: "random walkers"
- Propagate based on the diffusion part of Green's function G_d
  - As we showed, will propagate to ground state over time
- Use P as a "branching" probability distribution to choose whether:
  - Walker is killed
  - Walker continues its propagation
  - Walker continues its propagation and an additional one is spawned



Rev. Mod. Phys. 73 (2001)

Importance sampling in QMC

- Note that P exponentially suppresses propagation into high-potential areas, and potential may vary quickly and significantly
- We can make this more efficient with importance sampling
- Construct a "probability-like" function:

$$F(\mathbf{R},t) = \Phi(\mathbf{R},t)\Psi(\mathbf{R})$$

- Where Ψ is a trial wave function, e.g., from variational QMC
- This satisfies the diffusion equation:

$$\frac{\partial F}{\partial t} = \frac{1}{2}\nabla^2 F - \nabla \cdot F\mathbf{U} + [E_c - \mathcal{E}(\mathbf{R})]F$$

- Where we have a "drift" velocity: $U = \nabla \ln \Psi(\mathbf{R})$
- And we see again the local energy: $\mathcal{E}(\mathbf{R}) = \dfrac{1}{\Psi(\mathbf{R})} H\Psi(\mathbf{R})$

Modified Green's function

- Can show that the new Green's function is:

$$G(\mathbf{R}, \mathbf{R}'; \tau) \simeq (2\pi\tau)^{-3N/2} \exp\left[-\frac{[\mathbf{R} - \mathbf{R}' - \tau\mathbf{U}(\mathbf{R}')]^2}{2\tau}\right]$$
$$\times \exp\left[-\tau \frac{\mathcal{E}(\mathbf{R}) + \mathcal{E}(\mathbf{R}') - 2E_c}{2}\right]$$

- The drift velocity pushes random walkers towards areas of high density of the trial wave function

- If the trial wavefunction is good, the local energy is approximately constant, so second term does not vary too rapidly

Sign problem and fixed node approximation

- We have a crucial issue not yet discussed: Probabilistic methods like MC assume that probability distributions are positive
- Because we require wavefunctions of fermions to be antisymmetric, they cannot be positive everywhere
  - Need to assign a sign to the walkers, may change as they move through configuration space
- This leads to the fermion sign problem: If we sample over many configurations, we will get approximately zero
  - Gives decaying signal to noise ratio rather than the other way around
- Fixed node approximation: Take the zeros of trial wavefunction to be fixed and prevent walkers from changing sign

Importance sampling and the fixed node approximation

- Recall the Green's function we got from importance sampling:

$$G(\mathbf{R},\mathbf{R}';\tau) \simeq (2\pi\tau)^{-3N/2}\exp\left[-\frac{[\mathbf{R}-\mathbf{R}'-\tau\mathbf{U}(\mathbf{R}')]^2}{2\tau}\right]$$

$$\times\exp\left[-\tau\frac{\mathcal{E}(\mathbf{R})+\mathcal{E}(\mathbf{R}')-2E_c}{2}\right]$$

- Drift velocity carries walkers away from nodal surface
- Local energy also diverges near the nodal surface
- So, this importance sampling helps enforce the fixed node approximation
    - Walkers can still traverse a node if the time step is too big

One more issue: Approximation for Green's function poor near nodes

- Our approximation for the Green's function is not good when the drift velocity and local energy become large
- Could take smaller time steps to make sure we are pushed away from nodes
- Alternative approach: One more accept/reject step:
  - Accept propagation with probability:

$$w(\mathbf{R}', \mathbf{R}, \tau) = \frac{\Psi(\mathbf{R}')^2 G(\mathbf{R}', \mathbf{R}; \tau)}{\Psi(\mathbf{R})^2 G(\mathbf{R}, \mathbf{R}'; \tau)}$$

- This actually improves the approximation to the Green's function by enforcing a key property of the exact Green's function: detailed balance

Procedure for diffusion QMC

- 1. Perform a variational Monte Carlo simulation to optimize variational parameters in trial wave function.
- 2. Use the wavefunction from step 1 to generate an initial ensemble of configurations
- 3. Update with drift term and random walk χ: $\mathbf{R}' = \mathbf{R} + \mathbf{U}\tau + \chi$
- 4. Reject any step that crosses a node.
- 5. Accept the move with probability:

$$w(\mathbf{R}', \mathbf{R}, \tau) = \frac{\Psi(\mathbf{R}')^2 G(\mathbf{R}', \mathbf{R}; \tau)}{\Psi(\mathbf{R})^2 G(\mathbf{R}, \mathbf{R}'; \tau)}$$

- 6. Create a new ensemble of walkers using branching probability *P*
- 7. Measure local energy
- 8. Update E_c by averaging local energy over configurations **R** and **R'**

Some comments on QMC

- Quantum Monte Carlo is often the standard for accuracy for numerical calculations of solids and molecules

- It is at the basis of many other methods in condensed-matter physics
  - I.e., density-functional theory approximations rely on QMC of homogeneous electron gas
  - Solvers for embedding methods such as dynamical mean-field theory use ”continuous time” QMC

- The key to an efficient accurate scheme is how to deal with the sign problem

Today's lecture:
QMC and Machine learning

- Variational and Diffusion Quantum Monte Carlo
- Intro to Machine learning

Machine learning

- Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data (Wikipedia)

- ML is a huge subject and is being applied in a wide range of scientific fields
    - Supervised learning: We know the “output” for some set of input data, want to know the output for the rest
    - Unsupervised: Take a set of inputs and find some structure
    - …

- We will focus our discussion: Supervised learning with neural networks

Patter recognition with computers

- Classic problem: Identify pictures of dogs versus cats
  - Easy for human, difficult for computer

Neural networks

- Neural networks attempt to mimic the action of neurons in a brain

- Good for problems where we have an incomplete or unsophisticated physical model, but a lot of data
  - Create a nonlinear fitting routine with free parameters
  - Train the network on data with known input and output to set the parameters
  - Trained network can be used on new inputs to predict outcome

- Help with pattern recognition, which is difficult for computers (often easy for humans)
  - Classic problem, identifying pictures of cats versus dogs

- Some uses:
  - Character / image recognition
  - AI for games
  - Classification of data
  - Finance

A simple linear model

- Represent input data as a vector *x*
- Represent output data as a vector *z*

- Simplest "model" that relates *x* and *z* is an unknown matrix **A**:

$$z = \mathbf{A}x$$

- This is just the same linear problem we have solved many times, but usually for *x* with a known **A**

- How can we get the values for **A**? If we have enough input/output data, we can figure it out

Solving for our linear model

- Say we have following data of input-output pairs:

$$x_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad z_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad z_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$x_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad z_3 = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

- We want to find **A** such that:

$$z_1 = \mathbf{A}x_1, \quad z_2 = \mathbf{A}x_2, \quad z_3 = \mathbf{A}x_3,$$

Solving our linear model

- We write: $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

- Take the first two pairs: $\mathbf{A}x_1 = \begin{pmatrix} a \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$$\mathbf{A}x_2 = \begin{pmatrix} b \\ d \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

- Now **A** is fully specified: $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$

- But we can't fulfill the last condition:

$$\mathbf{A}x_3 = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \neq \begin{pmatrix} 4 \\ 1 \end{pmatrix} = z_3$$

Nonlinear models

- We saw with the previous example:
  - We can "train" a model using known inputs and outputs
  - A linear model is too "definite," which is too restrictive
- Let's run our linear model through a nonlinear function g(*x*):

$$g(x) = \begin{pmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_n) \end{pmatrix}$$

- To get: $z = g(\mathbf{A}x)$

Nonlinear models

- Consider the simple nonlinear function: $g(p) = p^2$

- Solving the nonlinear equation with our inputs:

$$z_1 = g(\mathbf{A}x_1), \quad z_2 = g(\mathbf{A}x_2), \quad z_3 = g(\mathbf{A}x_3),$$

- Gives four valid solutions:

$$\mathbf{A}_1 = \begin{pmatrix} -1 & -1 \\ 0 & -1 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} -1 & -1 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{A}_3 = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \quad \mathbf{A}_4 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

- Nonlinear models give much greater flexibility for describing data
- Tradeoff is that they are harder to solve

After class tasks

- Homework 5 due November 19

- Readings:
  - QMC:
    - Pang Secs. 10.5, 10.6
    - https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.73.33
    - https://journals.aps.org/prb/abstract/10.1103/PhysRevB.16.3081
  - *Computational Methods for Physics*, Joel Franklin, Chapter 14
  - *Make Your Own Neural Network*, Tariq Rashid
  - http://playground.tensorflow.org