Quantum Ensemble Statistics

Simulating Fermions and Bosons using Monte Carlo Methods

Team 6

Harsh Vardhan Singh
EP23BTECH11012

Jaideep Nirmal A J
EP23BTECH11013

Sarang Gund
EP23BTECH11018

EP3120 Statistical Mechanics
Indian Institute of Technology Hyderabad

Fall 2025

Problem Statement

Consider a system of quantum particles in a **3D cube**. We study its **canonical ensemble** at temperature T with N particles.

- Build microstates of this ensemble numerically by **Monte Carlo** sampling for both fermions and bosons.
- Obtain the **energy distribution** of particles and compare with theoretical predictions from a **grand canonical ensemble**.

Plan of Action

1. Assemble N particles in a canonical ensemble, filling the lowest energy states first (similar to constructing the ensemble at $T = 0\,\mathrm{K}$).
2. Use the **Metropolis algorithm** (with temperature-dependent acceptance criteria) to perform random walks in configuration space until equilibrium.
3. The previous steps construct the canonical ensemble at the given temperature T.
4. Continue random walks and record occupation numbers for each energy.
5. Average the recorded statistics to obtain:

$$f(E) = \frac{n(E)}{g(E)}$$

where $g(E)$ is the degeneracy of the energy level E.

Logical Flow

1. Assemble particles in lowest energy states ($T = 0\,\mathrm{K}$).
2. Perform random walks using Metropolis until equilibrium ($T = T_{sim}\,\mathrm{K}$).
3. Record and average energy occupation statistics while continuing to take walks.
4. Find the probability of occupation for every energy state available

Random Walk Algorithm

Algorithm 1: Random Walk

Procedure WALK(*Microstate x*):
    $i \leftarrow$ random integer in $[1, x.N]$;
    chosen_one $\leftarrow$ *particles*$[i]$;
    new_k $\leftarrow$ 3 random integers in [1, MAX_QUANTUM_NUMBER];
    if *x has a particle with quantum numbers new_k* **then**
        new_particle $\leftarrow$ PARTICLE(*new_k*);
        $E_{initial} \leftarrow$ *chosen_one*.E;
        $E_{final} \leftarrow$ *new_particle*.E;
        $\Delta E \leftarrow E_{final} - E_{initial}$;
        if $\Delta E < 0$ **then**
            remove *chosen_one* from ensemble;
            add *new_particle* to ensemble;
        else
            swap particles with probability $e^{-\Delta E / k_B T}$;
    else
        do nothing;

Metropolis Acceptance Criteria

When attempting a transition from E_1 to E_2:

$$\Delta E = E_2 - E_1$$

Acceptance probability:

$$P_{\text{accept}} = \begin{cases} 1, & \text{if } \Delta E \leq 0 \\ e^{-\Delta E/(k_B T)}, & \text{if } \Delta E > 0 \end{cases}$$

This ensures correct canonical sampling.

Comparing Results

Fermions:

$$f_{FD}(E) = \frac{1}{e^{(E-\mu)/(k_B T)} + 1}$$

Bosons:

$$f_{BE}(E) = \frac{1}{e^{(E-\mu)/(k_B T)} - 1}$$

Simulation averages $f(E)$ are compared with these theoretical curves.

Changes Made for Bosons

- Removed Pauli exclusion restrictions in `add_particle()`.
- Allowed multiple particles per state in `particles_hashmap`.
- No forbidden transitions; higher acceptance rate observed.

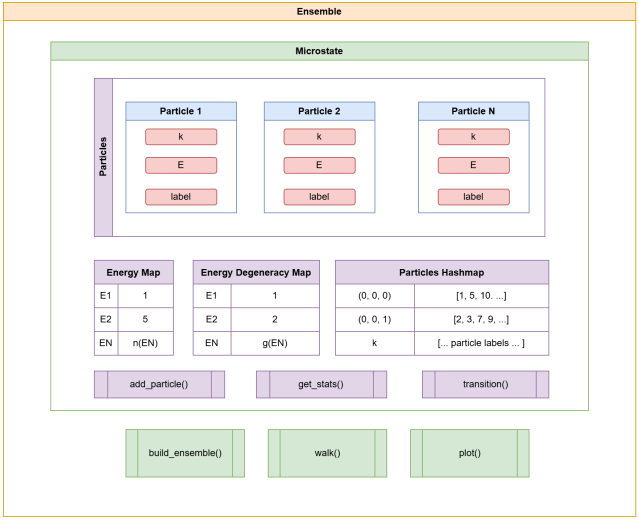
Structure

Object-Oriented Design for Clarity and Modularity:

- **Particle Class** – defines a single particle's state.
- **Microstate Class** – represents a system configuration.
- **Ensemble Class** – orchestrates simulation and statistics.

This modular design allows extending between fermions and bosons easily.

Structure


Ensemble
Microstate
Particles
Particle 1
k
E
label
Particle 2
k
E
label
Particle N
k
E
label
Energy Map
E1 1
E2 5
EN n(EN)
Energy Degeneracy Map
E1 1
E2 2
EN g(EN)
Particles Hashmap
(0, 0, 0) [1, 5, 10. ...]
(0, 0, 1) [2, 3, 7, 9, ...]
k [... particle labels ...]
add_particle()
get_stats()
transition()
build_ensemble()
walk()
plot()

Particle

Attributes:

- Quantum numbers $\mathbf{k} = (l, m, n)$ define the state.
- Energy computed as:

$$E = \frac{h^2(l^2 + m^2 + n^2)}{8\,m\,L^2}$$

Methods:

- `update_energy()` – recalculates energy.
- `__eq__()` – checks if two particles share the same state (used for Pauli exclusion).

Microstate

Attributes:

- `particles_hashmap` – maps each **k** state to particle indices.
- `energy_map` – stores occupations per energy level.
- `E` – total system energy.

Methods:

- `add_particle()` – enforces Pauli exclusion.
- `transition()` – performs Monte Carlo move (Metropolis).
- `get_stats()` – returns occupation distribution.

Ensemble

Attributes:

- N – number of particles.
- `system` – instance of the Microstate class.

Methods:

- `build_ensemble()` – initializes at $T = 0\,\mathrm{K}$.
- `walk()` – performs Monte Carlo random walks and records statistics.

Driver Code

The driver initializes parameters:

- Number of particles N
- Temperature T
- Box length L

It then:

1. Builds the initial ensemble.
2. Calls the `walk()` function.
3. Outputs occupation distributions and logs.

Fermions: Initial Assembly

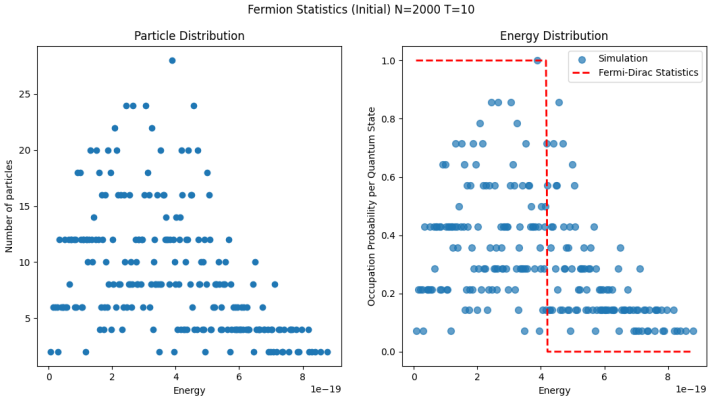


Figure: Filling of lowest states respecting Pauli exclusion

Fermions: Equilibrium (10K)

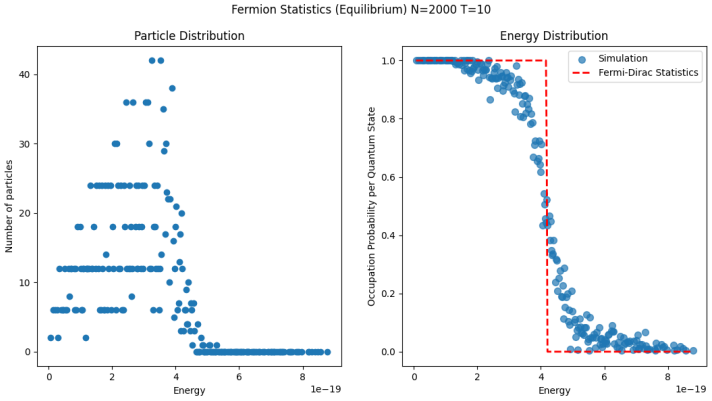


Figure: Occupation at equilibrium

Fermions: Statistics at 10 K

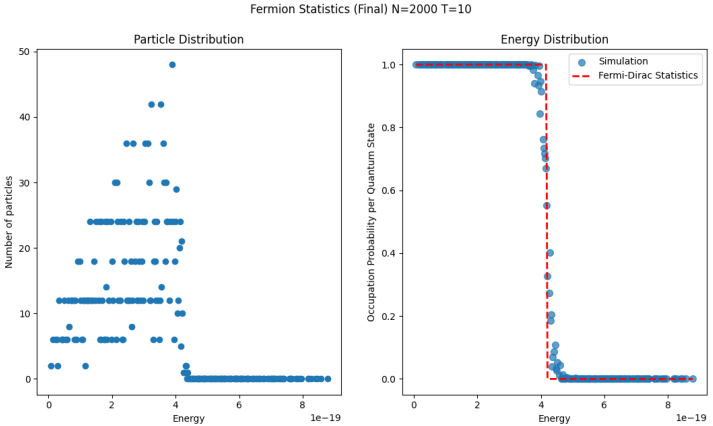


Figure: Final Statistics

Fermions: Metropolis Log

```
Equilibriating...
100%|| 50000/50000 [00:00<00:00, 62658.26it/s]

--- Simulation Summary ---
n_trials            : 50000
accepted            : 1529
rejected            : 28240
forbidden           : 20231
acceptance_rate     : 0.03
rejectance_rate     : 0.56
forbiddance_rate    : 0.4
Ef = 4.186843990036146e-19
Ensemble energy: 5.221397727999874e-16
1.5 N.Kb.T = 4.1419356000000007e-19
```

```
Exploring gamma space...
100%|| 150000/150000 [00:02<00:00, 58916.04it/s]

--- Simulation Summary ---
n_trials            : 150000
accepted            : 236
rejected            : 83350
forbidden           : 66414
acceptance_rate     : 0.0
rejectance_rate     : 0.56
forbiddance_rate    : 0.44
Ef = 4.186843990036146e-19
Ensemble energy: 5.183894119999871e-16
1.5 N.Kb.T = 4.1419356000000007e-19
```

Figure: Metropolis Algorithm Log

Fermions: Statistics at 300K

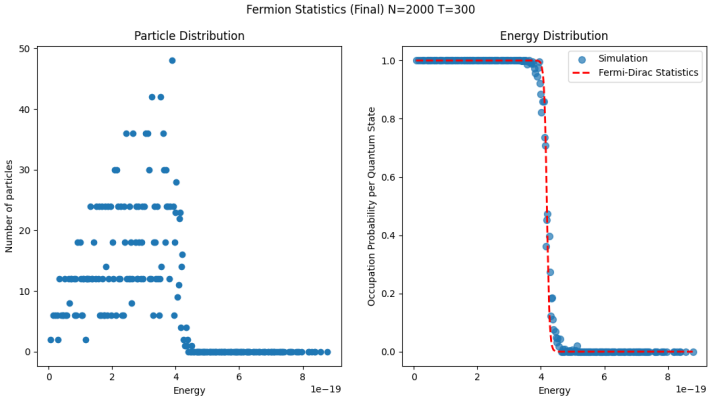


Figure: Fermion Statistics at 300K

Fermions: Metropolis Log

```
Equilibriating...
100%|| 50000/50000 [00:00<00:00, 62670.81it/s]

--- Simulation Summary ---
n_trials            : 50000
accepted            : 1710
rejected            : 28051
forbidden           : 20239
acceptance_rate     : 0.03
rejectance_rate     : 0.56
forbiddance_rate    : 0.4
Ef = 4.1909254363993487e-19
Ensemble energy: 5.2254142459998505e-16
1.5 N.Kb.T = 1.2425806800000001e-17
```

```
Exploring gamma space...
100%|| 150000/150000 [00:02<00:00, 61162.33it/s]

--- Simulation Summary ---
n_trials            : 150000
accepted            : 289
rejected            : 83554
forbidden           : 66157
acceptance_rate     : 0.0
rejectance_rate     : 0.56
forbiddance_rate    : 0.44
Ef = 4.1909254363993487e-19
Ensemble energy: 5.185466851999841e-16
1.5 N.Kb.T = 1.2425806800000001e-17
```

Figure: Metropolis Algorithm Log

Fermions: Statistics at 3000K

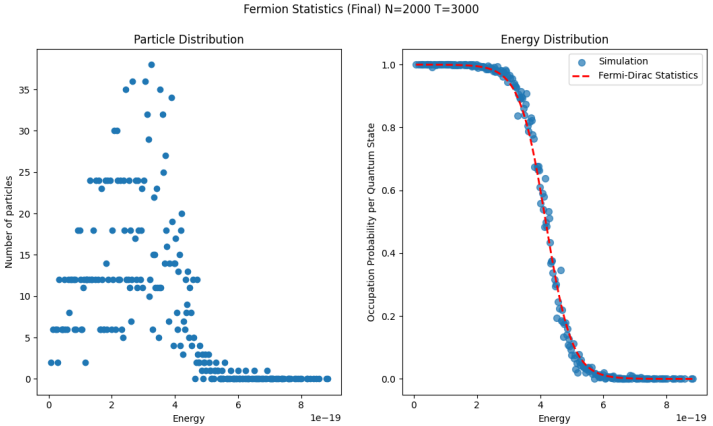


Figure: Fermion Statistics at 3000K

Fermions: Metropolis Log (3000K)

```
Equilibriating...
100%|██████████| 50000/50000 [00:01<00:00, 25174.32it/s]

--- Simulation Summary ---
n_trials            : 50000
accepted            : 3318
rejected            : 27570
forbidden           : 19112
acceptance_rate     : 0.07
rejectance_rate     : 0.55
forbiddance_rate    : 0.38
Ef = 4.1772812584776744e-19
Ensemble energy: 5.338941293999908e-16
1.5 N.Kb.T = 1.24258068e-16
```

```
Exploring gamma space...
100%|██████████| 150000/150000 [00:03<00:00, 41868.97it/s]

--- Simulation Summary ---
n_trials            : 150000
accepted            : 6059
rejected            : 83655
forbidden           : 60286
acceptance_rate     : 0.04
rejectance_rate     : 0.56
forbiddance_rate    : 0.4
Ef = 4.1772812584776744e-19
Ensemble energy: 5.335844219999929e-16
1.5 N.Kb.T = 1.24258068e-16
```

Figure: Metropolis Algorithm Log

Bosons: Initial Assembly

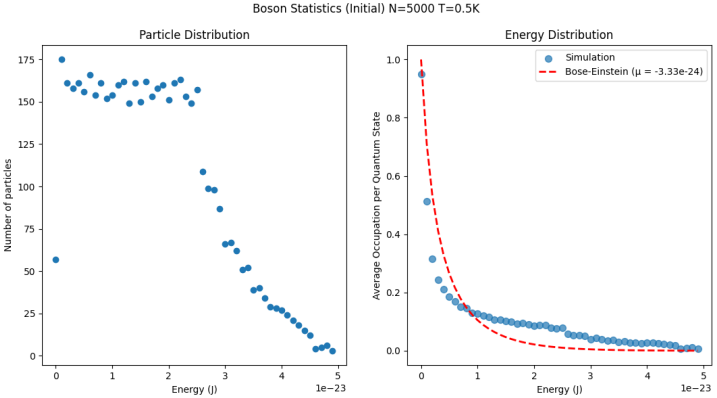


Figure: Filling of Bosons

Bosons: Equilibrium

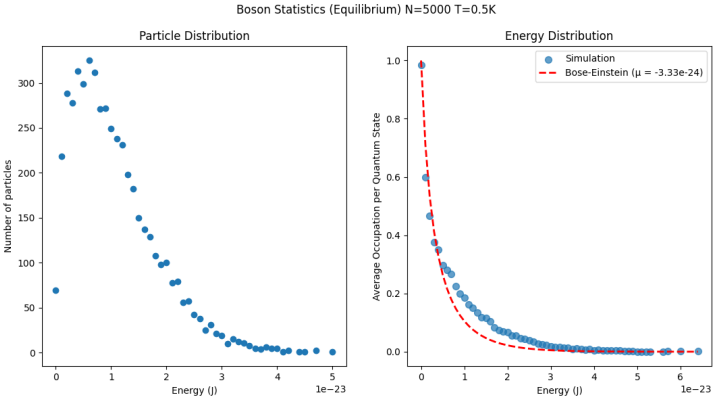


Figure: Energy clusters toward low levels after equilibration

Bosons: Statistics

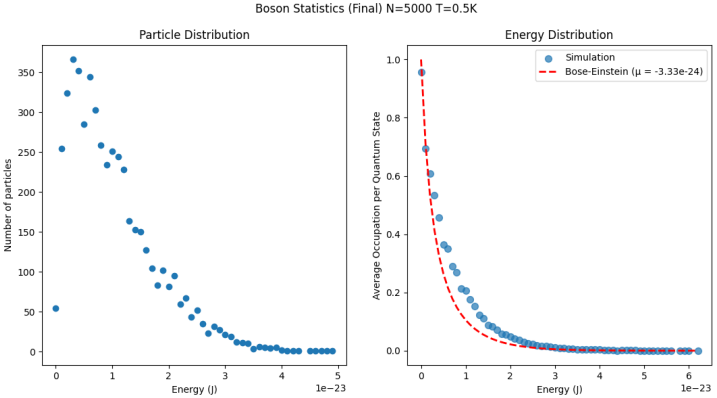


Figure: Averaged Bose–Einstein distribution from Monte Carlo simulation

Bosons: Metropolis Log

```
Equilibriating...
100%|████████████████████████████████████████████████████████████| 50000/50000 [00:00<00:00, 57188.87it/s]

--- Simulation Summary ---
n_trials            : 50000
accepted            : 17708
rejected            : 32292
acceptance_rate     : 0.35
rejectance_rate     : 0.65
Computed chemical potential μ = -3.330764e-24 J
```

```
Exploring the gamma space...
100%|████████████████████████████████████████████████████████████| 150000/150000 [00:02<00:00, 51374.15it/s]

--- Simulation Summary ---
n_trials            : 150000
accepted            : 45555
rejected            : 104445
acceptance_rate     : 0.3
rejectance_rate     : 0.7
```

Figure: Metropolis Algorithm Log

Bosons: Statistics at 1K

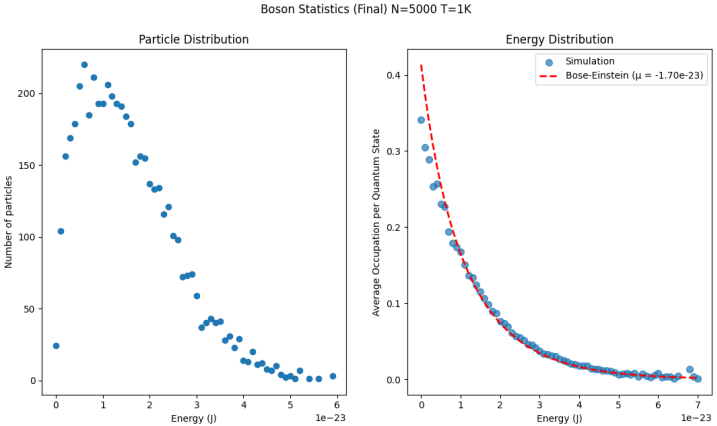


Figure: Boson Statistics at 1K

Bosons: Metropolis Log

```
Equilibriating...
100%|| 50000/50000 [00:00<00:00, 52436.29it/s]

--- Simulation Summary ---
n_trials            : 50000
accepted            : 27858
rejected            : 22142
acceptance_rate     : 0.56
rejectance_rate     : 0.44
Computed chemical potential μ = -1.696727e-23 J
```

```
Exploring the gamma space...
100%|| 150000/150000 [00:02<00:00, 50348.06it/s]

--- Simulation Summary ---
n_trials            : 150000
accepted            : 83167
rejected            : 66833
acceptance_rate     : 0.55
rejectance_rate     : 0.45
```

Figure: Metropolis Algorithm Log

Bosons: Statistics at 10K

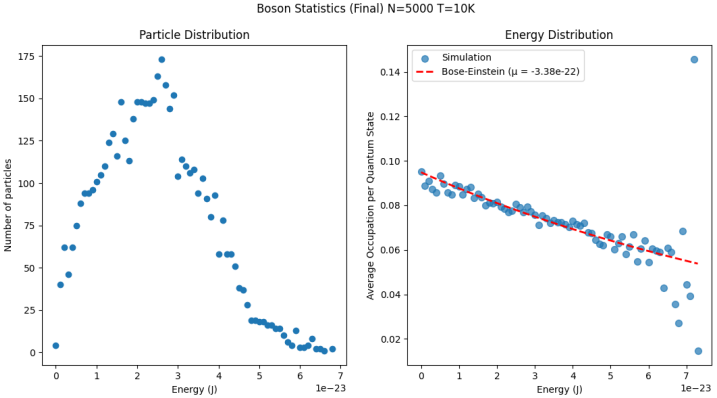


Figure: Boson Statistics at 10K

Bosons: Metropolis Log

```
Equilibriating...
100%|| 50000/50000 [00:01<00:00, 47638.21it/s]

--- Simulation Summary ---
n_trials            : 50000
accepted            : 47191
rejected            : 2809
acceptance_rate     : 0.94
rejectance_rate     : 0.06
Computed chemical potential μ = -3.377262e-22 J
```

```
Exploring the gamma space...
100%|| 150000/150000 [00:03<00:00, 49103.66it/s]

--- Simulation Summary ---
n_trials            : 150000
accepted            : 142222
rejected            : 7778
acceptance_rate     : 0.95
rejectance_rate     : 0.05
```

Figure: Metropolis Algorithm Log

Salient features of our code

- Highly modular implementation using OOPS
- Optimized code using hashmaps for
  1. Faster checking of existence of particle
  2. Accumulating energy records during walks
- We successfully simulated (at least) 2000 particles and took 1,50,000 random walks in < 5 seconds
- $f(E)$ matches very close to theoretical results

Thank you!