Lattice Gauge Theories and Lattice QCD

Unlike the QED fine structure constant \( \alpha \), the QCD coupling parameter \( \alpha_s(Q) \) depends strongly on the energy (inverse distance) scale at which the strong interaction takes place. Perturbation theory works only at high energies \( Q \gg 1 \text{ GeV} \). Lattice QCD is the most powerful and widely used approximation at lower energy scales, see A. Bazavov et al. Rev. Mod. Phys. 82, 1349-1417 (2010).
Left: Scale dependence of $\alpha_s$ Figure 9.4 PDG QCD Review. Right: Phases of QCD from USQCD The baryon chemical potential is the Fermi energy $E_F \sim n_B^{3/2}$ where $n_B$ is the number density.

**U(1) Lattice Gauge Theory: Quantumelectrodynamics on a Lattice**

Lattice QCD is a little too complicated to study in this course because it involves the Lie group SU(3) and numerical simulations are not practical on personal computers.

The essential concepts are illustrated by lattice QED, which involves the simpler Circle Group U(1).

**Covariant Formulation of Electrodynamics**

The Lagrangian density and Action for classical electrodynamics are discussed in Wikipedia Covariant formulation of classical electromagnetism

$$\mathcal{L}_{CED} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} - \frac{1}{e} J_\mu A^\mu, \quad S = \int d^4x \mathcal{L}_{ED},$$

where the electromagnetic field is expressed in terms of the 4-vector potential $A^\mu = (\Phi, A)$
\[ F_{\mu \nu} = \frac{\partial A_\nu}{\partial x^\mu} - \frac{\partial A_\mu}{\partial x^\nu} = \begin{pmatrix} 0 & E_x & E_y & E_z \\ -E_x & 0 & -B_z & B_y \\ -E_y & B_z & 0 & -B_x \\ -E_z & -B_y & B_x & 0 \end{pmatrix} \]

and \( J_\mu(x) = (c \rho, \textbf{J}) \) is the electromagnetic 4-vector current density. The action is invariant under local gauge transformations

\[ A_\mu(x) \rightarrow A_\mu(x) - \partial_\mu \chi(x) , \]

where \( \chi(x) \) is an arbitrary function of space and time, provided that the electromagnetic current is conserved

\[ \partial_\mu J^\mu(x) = 0 , \]

which implies on integrating by parts

\[
\int d^4 x \ A_\mu J^\mu \rightarrow \int d^4 x \ A_\mu J^\mu + \int d^4 x \ (\partial_\mu \chi) J^\mu \\
= \int d^4 x \ A_\mu J^\mu - \int d^4 x \ \chi \partial_\mu J^\mu = \int d^4 x \ A_\mu J^\mu .
\]

Relativistic electrons obey the Dirac equation

\[ [\gamma^\mu (i \partial_\mu - eA_\mu) - m] \psi(x) = 0 , \]

where \( e = -|e| \) is the electron charge, \( \psi \) is a 4-component electron wavefunction, and \( \gamma^\mu \) are \( 4 \times 4 \) Dirac Gamma matrices. The Dirac equation can be derived from the Lagrange density

\[ \mathcal{L}_{\text{Dirac}} = \bar{\psi}(x) \left[ \gamma^\mu (i \partial_\mu - eA_\mu) - m \right] \psi(x) , \quad \bar{\psi} = \psi^\dagger \gamma^0 . \]
The electromagnetic current for electrons in QED is

\[ J^\mu = e \bar{\psi}(x) \gamma^\mu \psi(x) , \]

and the Lagrange density for QED is

\[ \mathcal{L}_{\text{QED}} = \bar{\psi}(x) \left[ \gamma^\mu (i\partial_\mu - eA_\mu) - m \right] \psi(x) - \frac{1}{4} F_{\mu\nu} F^{\mu\nu} . \]

Photons and electrons are quanta of the \( A_\mu(x) \) and \( \psi(x) \) quantum fields, respectively.

### Local U(1) Gauge Invariance of QED

The 4-vector field \( A_\mu \) has four components, but real photons have only two independent polarization states.

The time component \( A_0 \) does not have a canonical momentum associated with it because

\[ F_{00} = 0 , \quad \frac{\partial \mathcal{L}}{\partial A_0} = 0 , \]

and does not have quanta associated with it when the field is quantized.

In addition, the action is invariant under local gauge transformations

\[ A_\mu(x) \rightarrow A_\mu(x) - \partial_\mu \chi(x) , \quad \psi(x) \rightarrow e^{ie\chi(x)} \psi(x) , \]

which rotate the phase of the electron field. This local gauge freedom can be used to eliminate one of the spatial components of the field \( A_\mu \) from the equations of motion. In the quantum theory this gauged component corresponds to a longitudinal polarization. Real photons only have transverse polarization.
Local gauge invariance also ensures that the mass of the photon is zero and is not affected by quantum fluctuations or higher order perturbations.

In the Standard Model, the strong and weak interactions are also mediated by massless gauge fields like the photon. Gauge invariance required in order to define a consistent (renormalizable) perturbation theory in these models. It is therefore very important to preserve local gauge invariance when these theories are discretized for numerical simulation.

Path Integral Quantization and Euclidean Space

We saw that the harmonic oscillator can be quantized using Feynman’s path integral formulation of quantum mechanics. Analytically continuing the path integral to imaginary time $\tau = -it$ gives the partition function of a classical oscillator ensemble at finite temperature $k_B T = 1/\beta = \hbar/\tau$, which can be simulated using Monte Carlo methods such as the Metropolis algorithm.

Continuing to imaginary time introduces various factors of $i$ and minus signs relative to the conventions of perturbative QED. To simplify the Euclidean formulas and make them consistent with lattice gauge theory references, use natural units $\hbar = c = k_B = 1$ and define a positive Euclidean space metric and Euclidean Dirac matrices:

$$ g_{\mu\nu} = \text{diag}(1, -1, -1, -1) \rightarrow \delta_{\mu\nu}, \quad i\gamma^\mu \partial_\mu = i\gamma^0 \frac{\partial}{\partial t} - i\gamma \cdot \nabla \rightarrow -\gamma_E^0 \frac{\partial}{\partial \tau} - \gamma_E \cdot \nabla = -\gamma_E^\mu \partial_\mu, $$
which defines Dirac matrices in Euclidean space

\[ \gamma^\mu_E \gamma^\nu_E + \gamma^\nu_E \gamma^\mu_E = \delta^{\mu\nu}. \]

The Euclidean space action

\[ S = \int d^4x \mathcal{L}_{\text{QED}} \rightarrow i \int d^4xE \mathcal{L}_{\text{EQED}}, \quad \mathcal{L}_{\text{EQED}} = \frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \bar{\psi} \left[ \gamma^\mu_E (\nabla - ieA_\mu) + m \right] \psi. \]

The subscript \( E \) is not generally used in the literature and we will drop it, keeping in mind that Euclidean Dirac matrices and solutions of the Dirac equation are not the same in Minkowski and Euclidean space.

**Gauge Invariant Plaquette Action**

The basic concept introduced by Wilson *Phys. Rev. D* 10, 2445-2459 (1974) was to discretize the action on a 4-D hypercubic lattice in Euclidean space

\[ x_\mu \rightarrow an = a(n_\tau, n_x, n_y, n_z), \quad n_\mu = 0, \pm 1, \pm 2, \ldots \]

with lattice spacing \( a \). Wilson’s discretization breaks translation, rotation and Lorentz symmetry, but preserves exact gauge invariance for any lattice spacing.

If \( A_\mu(n), \psi(n) \) represent the discretized photon and electron fields at lattice point \( n \) and the 4-gradient is discretized using a symmetric difference

\[ \partial_\mu \psi(x) \rightarrow \frac{\psi(n + \hat{\mu}) - \psi(n - \hat{\mu})}{2a} \]
the Dirac action for the electrons is discretized

\[ \int d^4x \bar{\psi}(x) \gamma^\mu \nabla_\mu \psi(x) \rightarrow \frac{a^3}{2} \sum_n \sum_\mu \left( \bar{\psi}(n) \gamma^\mu \psi(n + \hat{\mu}) - \bar{\psi}(n + \hat{\mu}) \gamma^\mu \psi(n) \right). \]

This discretization couples the electron field at neighboring sites and will break local gauge invariance if the interaction term is discretized

\[ \int d^4x \bar{\psi}(x) \gamma^\mu A_\mu \psi(x) \rightarrow a^4 \sum_n \sum_\mu \bar{\psi}(n) \gamma^\mu A_\mu(n) \psi(n). \]

Wilson modified the representation of the photon field on the lattice to preserve gauge invariance by replacing the real variable \(-\infty < A_\mu(x) < \infty\) with a periodic angular variable

\[ \theta_\mu(n) = -\theta_{-\mu}(n + \hat{\mu}), \quad 0 < \theta < 2\pi \]

associated with the directed link \(n \to n + \hat{\mu}\) as illustrated in the following figure:

![Directed Plaquette and Its Angular Variables](image)
The kinetic and interaction terms are discretized together using
\[
\frac{a^3}{2} \sum_n \sum_{\mu} \left[ \bar{\psi}(n) \gamma^\mu \psi(n + \hat{\mu}) e^{i \theta_\mu(n)} - \bar{\psi}(n + \hat{\mu}) \gamma^\mu \psi(n) e^{-i \theta_\mu(n)} \right].
\]

This action will be invariant local phase rotation of the electron field if the angular variable transforms simultaneously as follows:
\[
\psi(n) \rightarrow e^{i \chi(n)} \psi(n), \quad \bar{\psi}(n) \rightarrow e^{-i \chi(n)} \bar{\psi}(n), \quad \theta_\mu(n) \rightarrow \theta_\mu(n) - e[\chi(n + \hat{\mu}) - \chi(n)].
\]

To discretize the locally gauge invariant field strength tensor $F_{\mu\nu}(x)$ Wilson defined the quantity
\[
f_{\mu\nu}(n) = \theta_\mu(n) + \theta_\nu(n + \hat{\mu}) + \theta_{-\mu}(n + \hat{\mu} + \hat{\nu}) + \theta_{-\nu}(n + \hat{\nu})
= \theta_\mu(n) + \theta_\nu(n + \hat{\mu}) - \theta_\mu(n + \hat{\nu}) - \theta_\nu(n)
\]
associated with the elementary oriented plaquette (unit square) formed by the closed loop of links
\[
n \rightarrow n + \hat{\mu} \rightarrow n + \hat{\mu} + \hat{\nu} \rightarrow n + \hat{\nu} \rightarrow n.
\]

The action for the photon field is discretized using
\[
\frac{1}{4} \int d^4 x \ F_{\mu\nu} F^{\mu\nu} \rightarrow \sum_{n\mu\nu} [1 - \cos f_{\mu\nu}(n)].
\]

The mass term is locally gauge invariant with the simple discretization
\[
\int d^4 x \bar{\psi}(x) m \psi(x) \rightarrow ma^4 \sum_n \bar{\psi}(n) \psi(n).
\]
In the continuum limit \( a \to 0 \) the discretized action is recovered by scaling the lattice gauge fields

\[
\theta_\mu(n) = a A_\mu(x), \quad f_{\mu\nu}(n) = a^2 F_{\mu\nu}(x).
\]

**U(1) Lattice Gauge Theory in 4-D**

The article by M. Creutz, L. Jacobs, C. Rebbi, "Monte Carlo study of Abelian lattice gauge theories", *Phys. Rev. D20*, 1915-1922 (1979) describes simulations of lattice QED that illustrate methods used for the more complex and computationally demanding problem of lattice QCD.

Maxwell’s equations for electromagnetic waves in vacuum can be derived from the Lagrangian density

\[
\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu}, \quad F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu.
\]

The corresponding quantum field theory defined by the Feynman path integral \( \mathcal{Z} \) and action \( S \)

\[
\mathcal{Z} = \int \mathcal{D}[A] e^{iS[A]/\hbar}, \quad S = \int d^4x \mathcal{L}.
\]

describes non-interacting massless spin-1 photons with two transverse polarization degrees of freedom.

**Thermal Cycles**

In a thermal cycle the system is prepared in the ordered state at low temperature (large \( \beta \)) and then gradually heated to high temperature (\( \beta \to 0 \)) and then cooled down again. A first order phase transition
will appear as a **Hysteresis loop** in the internal energy, analogous to lagging of the magnetic induction when a ferromagnetic material is cycled in an external magnetic field.

For $N \leq 4$ the $A_N$ models exhibit a single first-order phase transition, while for $N \geq 5$ they observe two transitions of higher order. As $N$ increases, one of these transitions moves toward zero temperature, whereas the other other remains at finite temperature and survives in the limit $N \to \infty$.

**FIG. 1.** Thermal cycles on the models $Z_2$, $Z_3$, $Z_4$, $Z_5$, $Z_6$, and $Z_8$. 

For $N \leq 4$ the $A_N$ models exhibit a single first-order phase transition, while for $N \geq 5$ they observe two transitions of higher order. As $N$ increases, one of these transitions moves toward zero temperature, whereas the other other remains at finite temperature and survives in the limit $N \to \infty$.
Wilson Loops

FIG. 8. Square Wilson loops of sides one and two for the group $Z_2$.


#include <cmath>
#include <cstdlib>
#include <fstream>
#include <iostream>
#include <string>
using namespace std;

#include "../tools/random.hpp"

const double pi = 4 * atan(1.0);

const int D = 4; // number of dimensions
// number of sites in t, x, y, z directions
const int N0 = 20, N1 = 8, N2 = 8, N3 = 8;
const int N[D] = { N0, N1, N2, N3 };
int spin[N0][N1][N2][N3][D];

enum Initial_state {


init_identity,       // set all U = 1
init_random         // set each U randomly
};

void initialize_spins(int Z_N, Initial_state state)
{
    for (int i0 = 0; i0 < N[0]; ++i0) for (int i1 = 0; i1 < N[1]; ++i1)
        for (int i2 = 0; i2 < N[2]; ++i2) for (int i3 = 0; i3 < N[3]; ++i3)
            for (int d = 0; d < D; ++d)
            {
                if (state == init_random)
                    spin[i0][i1][i2][i3][d] = int(uniform_dist() * Z_N);
                else
                    spin[i0][i1][i2][i3][d] = 0;
            }
}
Average Energy Per Plaquette

The energy is

$$E = \sum_{\text{plaquettes}} \left[ 1 - \cos(\theta_\mu(n) + \theta_\nu(n + \hat{\mu}) - \theta_\mu(n + \hat{\nu}) - \theta_\nu(n)) \right].$$

abelian.cpp

double average_energy(int Z_N)
{
    double e_av = 0;
    int n[D];
    for (n[0] = 0; n[0] < N[0]; ++n[0])
        for (n[1] = 0; n[1] < N[1]; ++n[1])
                for (n[3] = 0; n[3] < N[3]; ++n[3])
                {
                    for (int mu = 0; mu < D - 1; ++mu)
                        for (int nu = mu + 1; nu < D; ++nu)
                        {
                            int n_mu[D], n_nu[D];
                            for (int d = 0; d < D; ++d)
                                n_mu[d] = n_nu[d] = n[d];
                            n_mu[mu] = (n_mu[mu] + 1) % N[mu];
                            n_nu[nu] = (n_nu[nu] + 1) % N[nu];
                            int spin_sum = spin[n[0]][n[1]][n[2]][n[3]][mu]
+ spin[n_mu[0]][n_mu[1]][n_mu[2]][n_mu[3]][nu]
- spin[n_nu[0]][n_nu[1]][n_nu[2]][n_nu[3]][mu]
- spin[n[0]][n[1]][n[2]][n[3]][nu];

    e_av += 1 - cos(2 * pi * spin_sum / double(Z_N));
}
}

Monte Carlo Update using the Metropolis Algorithm

A sweep of the lattice visits every spin in some fixed order. The spin associated with the pair of lattice sites \( n, n + \hat{\mu} \) is updated using the basic Metropolis algorithm.

- A trial step resets \( U_\mu(n) \) to a randomly chosen element of \( Z_N \)

\[
Z_N = \left\{ e^{2\pi ik/N}, \quad k = 0, 1, \ldots, N-1 \right\}
\]

- The change \( \Delta E \) in energy of the system is computed and the change is accepted if

\[
e^{-\beta \Delta E} > r
\]

where \( r \) is a uniform random number in \( [0, 1) \).
Because the trial step changes the value of a single spin, the change in energy can be computed locally. In $D = 4$ dimensions bond in the $\mu$ direction belongs to six plaquettes, two each in the 3 orthogonal directions.

```cpp
bool one_metropolis_step(int Z_N, double w_B[], int n[], int mu) {
    bool accepted = false;
    int trial_spin = int(uniform_dist() * Z_N);
    double w = 1;
    for (int nu = 0; nu < D; ++nu) {
        if (nu == mu)
            continue;
        int n_mu[D], n_nu[D];
        // plaquette in n + nu direction
        for (int d = 0; d < D; ++d)
            n_mu[d] = n_nu[d] = n[d];
        n_mu[mu] = (n_mu[mu] + 1) % N[mu];
        n_nu[nu] = (n_nu[nu] + 1) % N[nu];
        int spin_sum = trial_spin
            + spin[n_mu[0]][n_mu[1]][n_mu[2]][n_mu[3]][nu]
            - spin[n_nu[0]][n_nu[1]][n_nu[2]][n_nu[3]][mu];
    }
    //...
- spin[n[0]][n[1]][n[2]][n[3]][nu];
spin_sum = (spin_sum + 2 * Z_N) % Z_N;
w *= w_B[spin_sum];
spin_sum = spin[n[0]][n[1]][n[2]][n[3]][mu]
    + spin[n_mu[0]][n_mu[1]][n_mu[2]][n_mu[3]][nu]
    - spin[n_nu[0]][n_nu[1]][n_nu[2]][n_nu[3]][mu]
    - spin[n[0]][n[1]][n[2]][n[3]][nu];
spin_sum = (spin_sum + 2 * Z_N) % Z_N;
w /= w_B[spin_sum];

// plaquette in n - nu direction
int n_mn[D];  // n + mu - nu
for (int d = 0; d < D; ++d)
    n_mn[d] = n_nu[d] = n[d];
n_mn[mu] = (n_mn[mu] + 1) % N[mu];
n_mn[nu] = (n_mn[nu] - 1 + N[nu]) % N[nu];
n_nu[nu] = (n_nu[nu] - 1 + N[nu]) % N[nu];
spin_sum = - trial_spin
    - spin[n_nu[0]][n_nu[1]][n_nu[2]][n_nu[3]][nu]
    + spin[n_nu[0]][n_nu[1]][n_nu[2]][n_nu[3]][mu]
    + spin[n_mn[0]][n_mn[1]][n_mn[2]][n_mn[3]][nu];
spin_sum = (spin_sum + 2 * Z_N) % Z_N;
w *= w_B[spin_sum];
spin_sum = - spin[n[0]][n[1]][n[2]][n[3]][mu]
        - spin[n_nu[0]][n_nu[1]][n_nu[2]][n_nu[3]][nu]
        + spin[n_nu[0]][n_nu[1]][n_nu[2]][n_nu[3]][mu]
        + spin[n_mn[0]][n_mn[1]][n_mn[2]][n_mn[3]][nu];
spin_sum = (spin_sum + 2 * Z_N) % Z_N;
w /= w_B[spin_sum];

// Metropolis test
if (w > 1 || w > uniform_dist()) {
    spin[n[0]][n[1]][n[2]][n[3]][mu] = trial_spin;
    accepted = true;
}
return accepted;

void sweep_update_lattice(int Z_N, double beta)
{
    // compute exponential Boltzmann weights
    double w_B[Z_N];
    for (int j = 0; j < Z_N; ++j)
\[ w_B[j] = \exp(-\beta \times (1 - \cos(2 \times \pi \times j / \text{double}(Z_N)))); \]

int accepted_steps = 0;
int n[D];
for (n[0] = 0; n[0] < N[0]; ++n[0]) for (n[1] = 0; n[1] < N[1]; ++n[1])
{
    for (int mu = 0; mu < D; ++mu)
        if (one_metropolis_step(Z_N, w_B, n, mu))
            ++accepted_steps;
}

void thermal_cycle(int Z_N, double beta_0, int iterations)
{
    initialize_spins(Z_N, init_identity); // cold start
    cout << " Z_" << Z_N << " Model Thermal Cycle\n" << " Iter\t\t\tE\n" << ' ' << 0 << 't' << beta_0 << 't' << average_energy(Z_N) << '\n';
    ofstream file("abelian.data");
for (int iter = 0; iter < iterations; ++iter) {

    double beta = beta_0;
    if (iter < iterations / 2)
        beta *= (iterations / 2 - 1 - iter);
    else beta *= (iter + 1 - iterations / 2);
    beta /= iterations / 2;

    for (int i = 0; i < 1; ++i)
        sweep_update_lattice(Z_N, beta);

    if (((iter+1) % 16 == 0) {
        double e_av = average_energy(Z_N);
        cout << ' ' << iter + 1 << '	' << beta << '	' << e_av << '
';
        file << beta << '	' << e_av << '
';
    }
    }
    cout << " Output in file abelian.data" << endl;
}

int main()
{ 
    cout << " Monte Carlo Simulation of 4-D Abelian Gauge Models\n"
    << " ----------------------------------------------\n"
    << " N_t = " << N[0] << " , N_x = " << N[1] << " , N_y = " << N[2]
    << endl;

    int Z_N = 8;
    double beta_0 = 3.5; // starting value for thermal cycle
    int iterations = 7000;
    thermal_cycle(Z_N, beta_0, iterations);
}