Week Eight PHY-480

Lewis

October 14, 2025

1. Strategies for finding a local minimum of an unconstrained objective function

Three main methods are used to find local minima in complex landscapes:

1. **Gradient Descent:** Iteratively updates position using

$$\vec{x}_{n+1} = \vec{x}_n - \eta \nabla f(\vec{x}_n),$$

where η is the step size.

2. **Newton's Method:** Uses curvature from the Hessian,

$$\vec{x}_{n+1} = \vec{x}_n - \mathbf{H}^{-1} \nabla f(\vec{x}_n),$$

for faster local convergence.

3. **Heuristic Methods:** Which are techniques like simulated annealing which we did in the previous class assignment, or random restarts help escape poor local minima.

2. Strategies for including constraints

Common ways to handle constraints include:

1. **Lagrange Multipliers:** Add constraints $g_i(\vec{x}) = 0$ to the objective:

$$\mathcal{L} = f(\vec{x}) + \sum_i \lambda_i g_i(\vec{x}).$$

2. **Penalty Functions:** Add terms that penalize constraint violations, e.g.

$$g(x) = 1 - 2x^2 + x^4,$$

leading to

$$H = \vec{x}^T \mathbf{M} \vec{x} + \lambda \sum_i g(x_i).$$

3. **Projection or Relaxation:** Project solutions back into the feasible region or relax discrete constraints to continuous ones.

3. Gradient Descent vs Newton's Method

- **Gradient Descent:** Uses first derivatives only; notably slower but simpler.
- **Newton's Method:** Uses second derivatives, but faster near minima and more costly.

4. Gradient Descent Outline for Continuous Ising Model

Modified from the Class 13 Ising model by allowing continuous spins $x_i \in [-1, 1]$ and adding a penalty term.

```
# Continuous Gradient Descent Ising Model

# 1. Parameters
N = 50
lam = 1.0
eta = 0.01
steps = 5000

# 2. Create symmetric random coupling matrix J
J = random normal (N x N)
J = (J + J.T) / 2

# 3. Initialize continuous spins
x = uniform random values in [-1, 1]

# 4. Define energy and gradient
H(x) = x.T @ J @ x + lam * (1 - 2x^2 + x^4)
grad(x) = 2 * J @ x + lam * (-4x + 4x^3)

# 5. Gradient descent loop
for k in range(steps):
    x = x - eta * grad(x)
    clip x to [-1, 1]
    record energy H(x)

# 6. Output results
plot energy vs iteration
print final energy and configuration
```

Replaces discrete spin flips with gradient-based updates and uses the penalty term to enforce $|x_i| \approx 1$.