Ising Model with Monte Carlo

▪

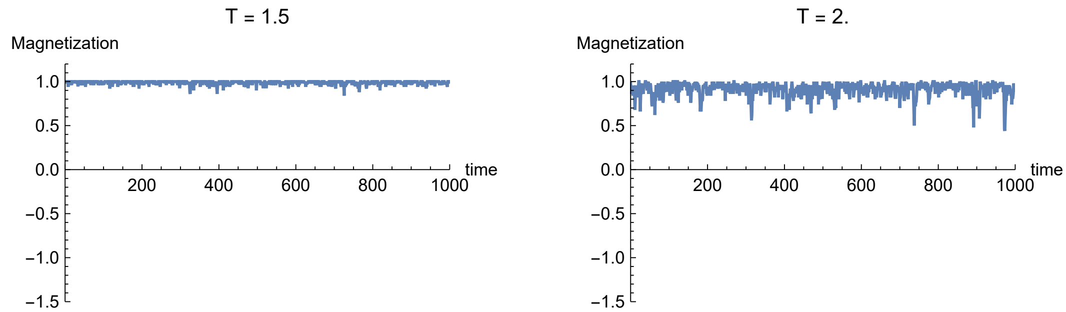
```
In[ ]:= dimension = 10; s = ConstantArray[1, {dimension + 2, dimension + 2}];
        (*2 is added to the dimension to account for the BCs*)
        s // MatrixForm ;

In[ ]:= magnetization = (dimension) ^2; (*bohr magneton * number*) (*total magnetization*)
        totalEnergy = -100 * 4 / 2; (*number * neighbors/ repititions*)
        EnergyPerSpin = totalEnergy / 100;

In[ ]:= Temp = Range[1, 5, 0.25] ; (*Temprature range*)
        LTemp = Length[Temp];

        (*Metrapolis*)
        times = 1000;(*run the metrapolis algorithm over all lattice spins 1000 times*)
        magnetizationAfterIteration = ConstantArray[0, times];
        EnergyAfterIteration = magnetizationAfterIteration;
        magnetizationAverage = ConstantArray[0, LTemp];
        EnergyAverage = magnetizationAverage;
        Mag = magnetizationAverage;

        Do[(*over Temp: nt*)
         t = Temp⟦nt⟧;

         Do[(*over the Metrapolis*)
          Do[(*over spins*)
           sumMag = 0;
           sumEng = 0;
           Do[
            Einitial = - s⟦i, j⟧ * (s⟦i + 1, j⟧ + s⟦i, j + 1⟧ + s⟦i - 1, j⟧ + s⟦i, j - 1⟧);
            Efinal = - (-1) * s⟦i, j⟧ * (s⟦i + 1, j⟧ + s⟦i, j + 1⟧ + s⟦i - 1, j⟧ + s⟦i, j - 1⟧);
            Eflip = Efinal - Einitial;

            If[Eflip ≤ 0,
             s⟦i, j⟧ = -s⟦i, j⟧;
             magnetization = magnetization + 2 * s⟦i, j⟧;
```

```
      totalEnergy = totalEnergy + Eflip;
      EnergyPerSpin = totalEnergy / 100
     ];

     If[Eflip > 0,
      r = RandomReal[];
      If[Exp[-Eflip / t] > r,
       s⟦i, j⟧ = -s⟦i, j⟧;
       magnetization = magnetization + 2 * s⟦i, j⟧;
       totalEnergy = totalEnergy + Eflip;
       EnergyPerSpin = totalEnergy / 100
      ];
     ];
     (*Periodic Boundary Conditions*)
     If[i == 2, s⟦dimension + 2, j⟧ = s⟦i, j⟧];
     If[j == 2, s⟦i, dimension + 2⟧ = s⟦i, j⟧];
     If[i == dimension + 1, s⟦1, j⟧ = s⟦i, j⟧];
     If[j == dimension + 1, s⟦i, 1⟧ = s⟦i, j⟧];

     , {j, 2, dimension + 1 }]
    , {i, 2, dimension + 1}
   ];
   magnetizationAfterIteration⟦nm⟧ = magnetization / dimension^2;
   EnergyAfterIteration⟦nm⟧ = EnergyPerSpin * dimension^2;
   sumMag += magnetization;
   sumEng += EnergyPerSpin;
   , {nm, times}
  ];
  Mag⟦nt⟧ = magnetizationAfterIteration;
  magnetizationAverage⟦nt⟧ = Mean[magnetizationAfterIteration];
  EnergyAverage⟦nt⟧ = Mean[EnergyAfterIteration] / dimension^2;
  , {nt, LTemp}]
```

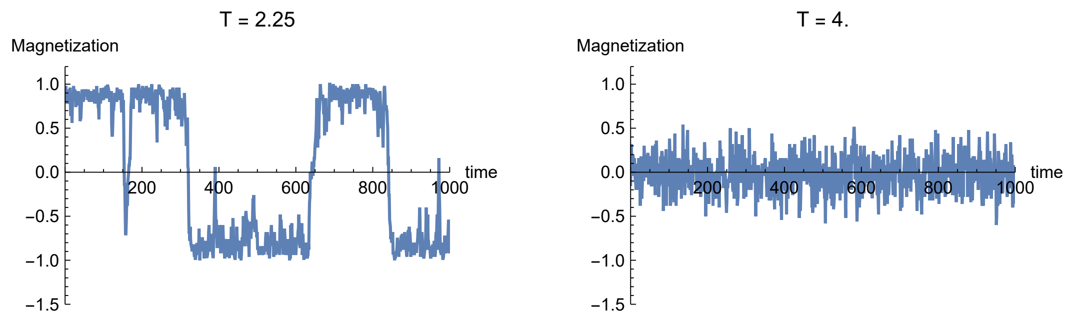
In[•]:=
```
selectedTemps = {1.5, 2.0, 2.25, 4.0};

plots = Table[0, {Length[selectedTemps]}];
counter = 1;
For[i = 1, i ≤ Length[selectedTemps], i++,
 index = Position[Temp, selectedTemps⟦i⟧]⟦1, 1⟧;

 plots⟦i⟧ = ListPlot[Mag⟦index⟧, AxesLabel → {"time", "Magnetization"},
   PlotRange → {{0, 1000}, {-1.5, 1.2}}, DataRange → {0, times},
   Joined → True, PlotLabel → "T = " <> ToString[Temp⟦index⟧]];
]
```

In[∘]:=
```
(*Display the plots*)
GraphicsRow[{plots⟦1⟧, plots⟦2⟧}]
GraphicsRow[{plots⟦3⟧, plots⟦4⟧}]
```

Out[∘]=



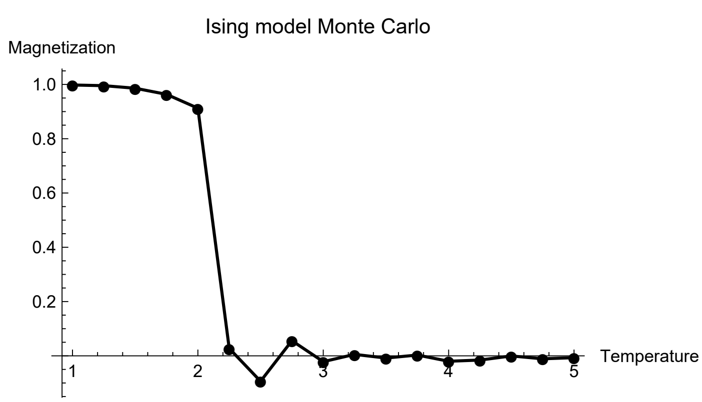
Out[∘]=



for T=1 the magnetization is close to saturation with minor fluctuations.
T=2 The average value of M decreases.
T=2.25 near the critical point, the system is extremely sensitive to small perturbations.
T=4 (The critical Temprature in our case) the fluctuations decrease in magnitude and are centered around M = 0.

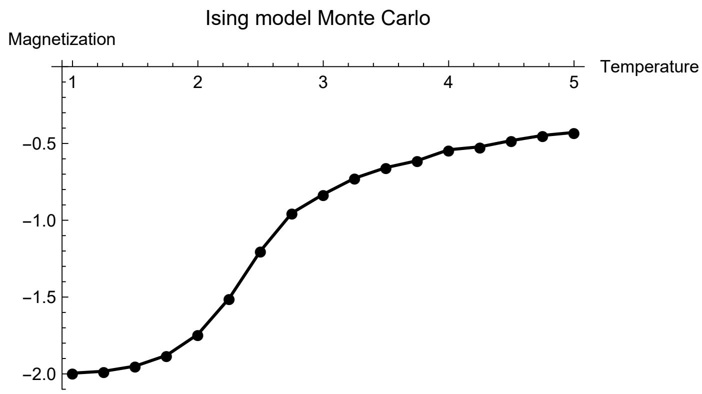
In[∘]:=
```
Show[ListPlot[Transpose[{Temp, magnetizationAverage}],
  Joined → True, PlotStyle → {Black}, PlotMarkers → "●",
  AxesLabel → {"Temperature", "Magnetization"}, PlotLabel → "Ising model Monte Carlo"]]
```

Out[∘]=



In[∘]:=
```
Show[ListPlot[Transpose[{Temp, EnergyAverage}],
  Joined → True, PlotStyle → {Black}, PlotMarkers → "●",
  AxesLabel → {"Temperature", "Magnetization"}, PlotLabel → "Ising model Monte Carlo"]]
```

Out[∘]=



We Note that at low tempratures we expect a ferromagnetic state with the lowest energy corresponding to $\frac{\langle E \rangle}{N} = -2$ down configuration as expected.