Importance sampling algorithm

0. *M=Z=0; k=1*
1. Generate a state (**s**) with probability P(s)~exp(-E(s)/T).
2. *M_k=M(s); k=k+1*
3. If *k<k_{max}* go to 2
4. $$\langle M \rangle = \frac{1}{k_{\max}} \sum_{k=1}^{k_{\max}} M_k$$

How to generate **s** with probability ~exp(−E(s)/T) ?

- Metropolis algorithm

Monte Carlo: Metropolis algorithm

Figures removed for copyright reasons.

Nicolas Metropolis **Edward Teller** **Stanslaw Ulam**

Metropolis algorithm

0. *Initialize* $M=Z=0$; $k=1$, $s=s_0$
1. *Make an attempt*
   a) draw random $\boldsymbol{\delta}$ from uniform distribution $[-\Delta,+\Delta]$
   b) $\boldsymbol{\Delta}E=E(s_{k-1}+\boldsymbol{\delta})-E(s_{k-1})$
2. *Metropolis criteria*
   IF $\boldsymbol{\Delta}E\leq 0$ **THEN** accept $s_k=s_{k-1}+\boldsymbol{\delta}$
   ELSE $P=\exp(-\boldsymbol{\Delta}E/T)$
   draw r from uniform distribution [0,1]
   IF $r<P$ **THEN** accept $s_k=s_{k-1}+\boldsymbol{\delta}$
   ELSE reject $s_k=s_{k-1}$
   ENDIF
3. *Bookkeeping*
   $M_k=M(s_k)$; $k=k+1$
4. *If* $k<k_{max}$ *go to 1*
5. $$\langle M \rangle = \frac{1}{k_{\max}} \sum_{k=1}^{k_{\max}} M_k$$

Metropolis algorithm

0. *Initialize*
1. *Make an attempt* <--- MOVE SET
2. *Metropolis criteria* <--- ENERGY FUNCTION
3. *Bookkeeping*

Why does it work ?

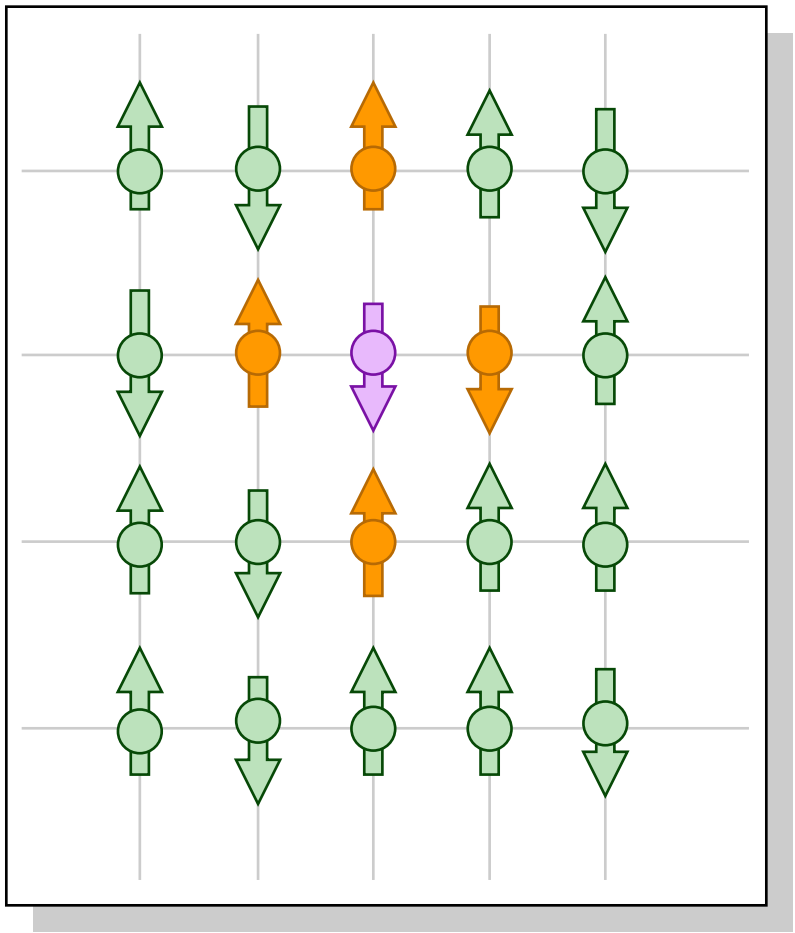
- Detailed balance
  - Equilibrium is reached if (and only if)

    Flux (s->t)=Flux(t->s)

Ising-Lenz model

Spin

$$s_i = \begin{cases} +1 \\ -1 \end{cases}$$



Energy

$$E(s) = -J \sum_{(i,j)} s_i \cdot s_j - H \sum_i s_i$$

Magnetization

$$M(s) = \sum_i s_i$$

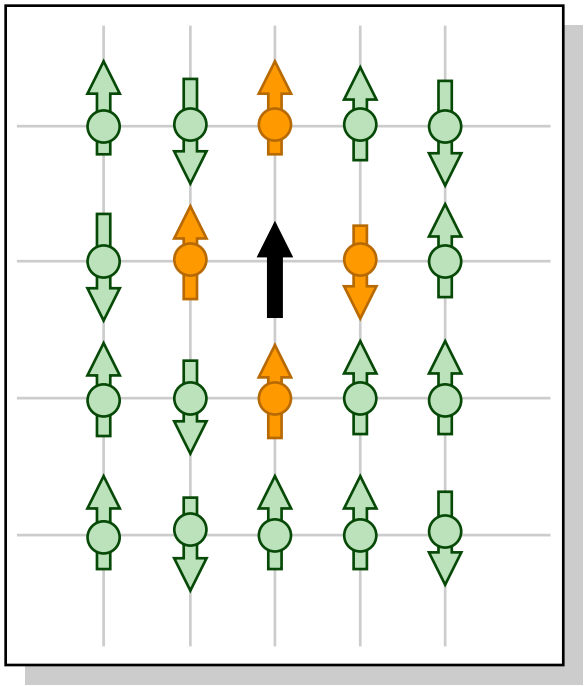
Figure by MIT OCW.

Metropolis algorithm

0. *Initialize*
1. *Make an attempt* <--- MOVE SET
2. *Metropolis criteria* <--- ENERGY FUNCTION
3. *Bookkeeping*

Metropolis algorithm

0. *Initialize*
1. *Make an attempt* <--- MOVE SET
2. *Metropolis criteria* <--- ENERGY FUNCTION
3. *Bookkeeping*



MOVE SET

$$E(s) = -J \sum_{(i,j)} s_i \cdot s_j - H \sum_i s_i$$

ENERGY FUNCTION

Figure by MIT OCW.

Metropolis algorithm

0. *Initialize* $M=Z=0$; $k=1$, $\mathbf{s}=\mathbf{s}_0$
1. *Make an attempt*
   a) Pick a spin at random and flip it : new configuration **s'**
   b) $\Delta E=E(\mathbf{s}')-E(\mathbf{s}_{k-1})$
2. *Metropolis criteria*
   IF $\Delta E \leq 0$ **THEN** accept $\mathbf{s}_k=\mathbf{s}'$
   ELSE $P=\exp(-\Delta E/T)$
   draw r from uniform distribution [0,1]
   IF $r<P$ **THEN** accept $\mathbf{s}_k=\mathbf{s}'$
   ELSE reject $\mathbf{s}_k=\mathbf{s}_{k-1}$
   ENDIF
3. *Bookkeeping*
   $M_k=M(\mathbf{s}_k)$; $k=k+1$
4. *If* $k<k_{max}$ *go to 1*
5. $$\langle M \rangle = \frac{1}{k_{max}} \sum_{k=1}^{k_{max}} M_k$$

Ising Demo

http://www.imedea.uib.es/~tessonec/applets/JIsing.php

Set-up of your simulations

1. Boundary conditions

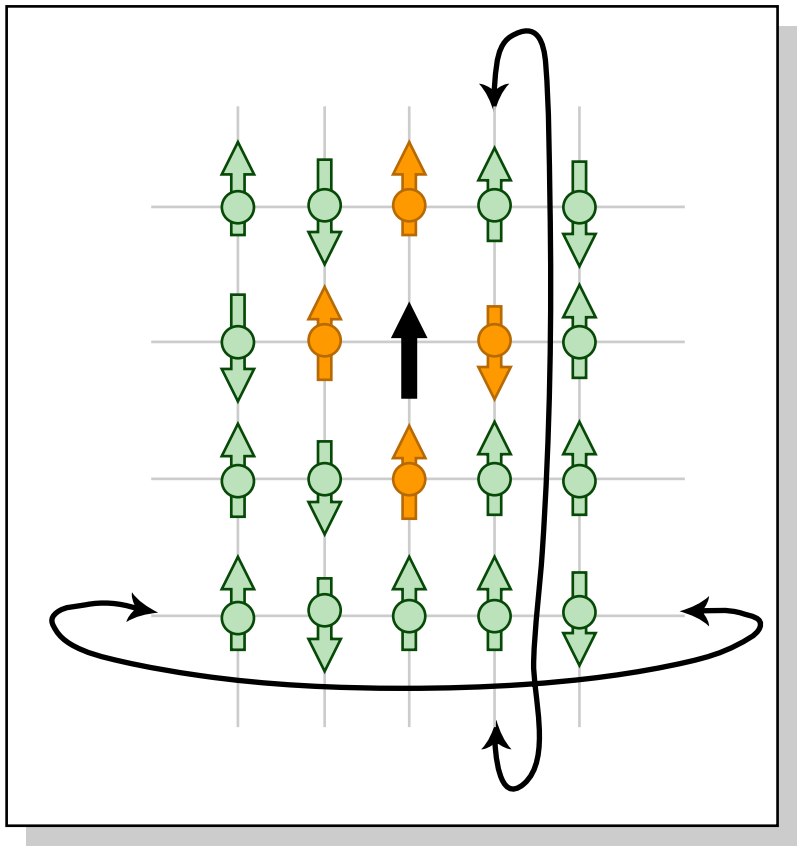
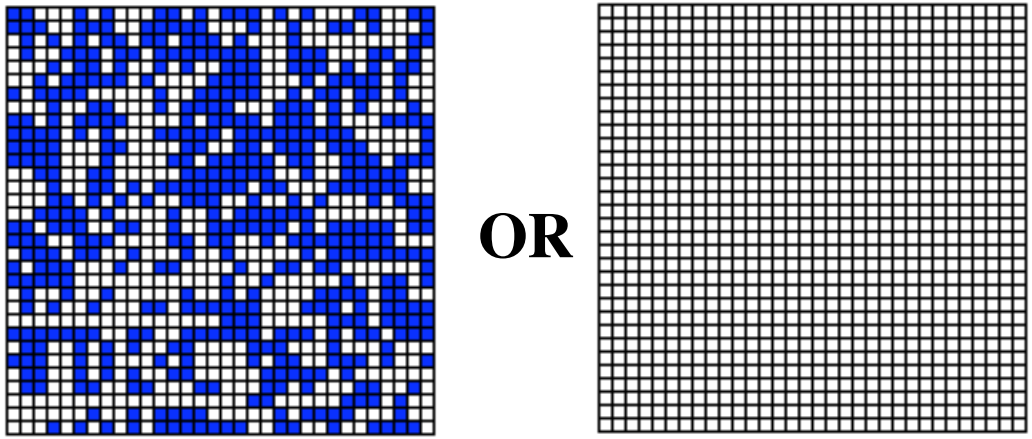


Figure by MIT OCW.

Set-up of your simulations

1. Boundary conditions

2. **Starting configuration**



Set-up of your simulations

1. Boundary conditions
2. Starting configuration
3. **Trajectory : s**(k), M(k), E(k)

Set-up of your simulations

1. Boundary conditions
2. Starting configuration
3. Trajectory : **s**(k), M(k), E(k)
4. **Sampling :** not every step

Set-up of your simulations

1. Boundary conditions
2. Starting configuration
3. Trajectory : **s**(k), M(k), E(k)
4. Sampling : not every step
5. **Equilibration :** the system you are about to enjoy is very hot
   ----> wait!