Program 17 / Chapter 20
Entropy and disorder

The relation between entropy and disorder is discussed in Section 20–7 of your textbook. The spontaneous mixing of dye in water or the spontaneous redistribution of energy between hot and cold objects can be understood in terms of increasing disorder, a process we model by the shuffling of an ordered deck of cards.

Theory To illustrate the concepts of entropy and disorder, rather than trying to simulate a complicated physical system, let us consider a simple mathematical model. Take a two dimensional lattice $N \times N$ sites; each site is either in a state of "up" or "down." A simple picture of such a system might be cards spread out on a table, a red card could be called "up" and a black card called "down." Initially, all the sites on the left side are "up" and all on the right side are "down." We now proceed to shuffle in the following way: Take two sites (or cards) at random and exchange their values (for cards, trade their locations). After many such shuffles, the initial order is destroyed and the system is said to be disordered.

We can be more quantitative about the degree of disorder by defining the quantity,

$$D = -\sum_{j=1}^{N} f_j \log(f_j)$$

where f_j is the fraction of sites in column j that are "up." This quantity, which we will call the disorder, is closely related to the definition of entropy in physical systems. Initially, the sites in each column are either all "up" or all "down" so $D = 0$.[†] As the shuffling proceeds, the disorder increases, approaching its maximum value,

$$D = -\sum_{j=1}^{N} \tfrac{1}{2} \log(\tfrac{1}{2}) = \frac{N}{2} \log(2)$$

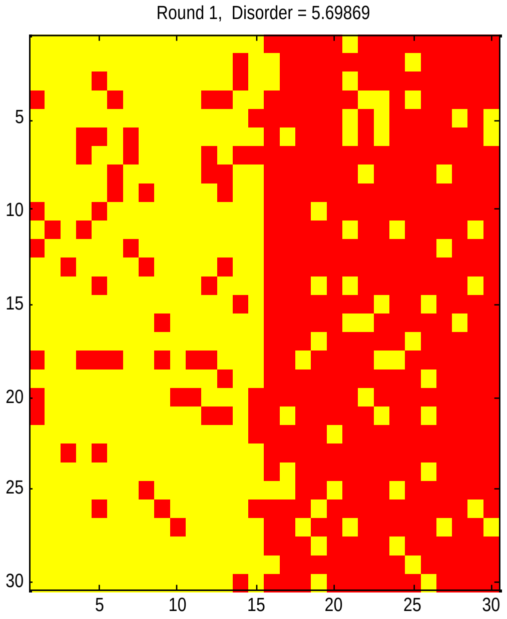
when in each column half the sites are "up" and half are "down."

Program The MATLAB program `shuffle`, which simulates a two dimensional lattice of sites being shuffled, is outlined below:

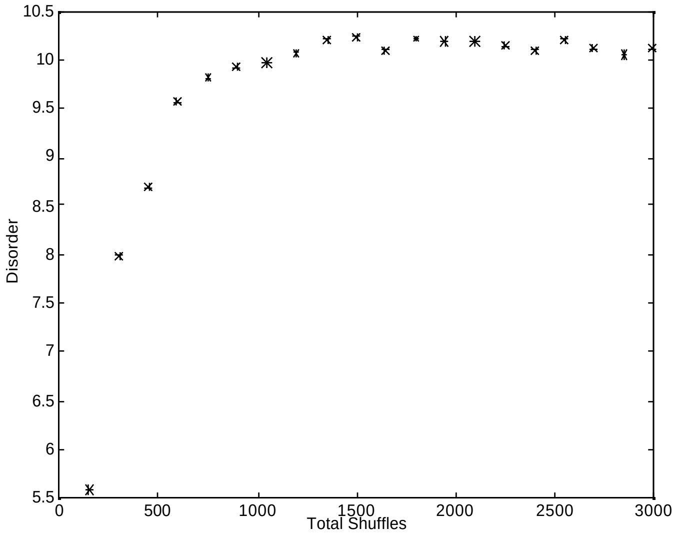
- Initialize variables (e.g., Number of sites, values at sites (0 or 1)).
- Prepare for plotting; plot the initial state.
- Loop over the desired number of rounds of shuffling.
  - Loop over the number of shuffles for this round.
    - Select two site at random.
    - Exchange the values at the two sites.
  - Sum over the columns to compute disorder.
    - Sum within a column to find fraction of sites that are up.
    - Compute disorder as sum of $f \log(f)$.
  - Plot a tableau showing the sites and display the disorder.
- Plot the disorder as a function total shuffles.

[†] Note that $\log(1) = 0$ and $x \log(x)$ goes to zero in the limit x going to zero.

The MATLAB function `rand` is used to generate the random numbers used in picking sites at random for shuffling; it returns real numbers uniformly drawn from the interval [0,1). Running the program with a lattice of 30 × 30 sites, after 150 shuffles the disorder has increased from zero to $D = 6.1$; the program produces the figure shown below.



A longer run with more shuffling (20 rounds, 150 shuffles per round), shows the evolution of the disorder towards its maximum value (see the figure below).



For this system, N = 30 so the maximum disorder is D = 15 log(2) = 10.4, which agrees with the maximum found in the figure above.

Exercises

1. (I) Run the `shuffle` program and print out the graph of disorder versus total shuffles for the following cases: (a) N = 10, 10 shuffles per round; (b) N = 20, 20 shuffles per round; (c) N = 20, 40 shuffles per round; (d) N = 30, 100 shuffles per round. In each case run the program for 30 rounds of shuffling.

2. (I) Modify the `shuffle` program so that it draws a horizontal line on the final plot marking the maximum possible disorder. Run the cases given in exercise 1 using your modified program.

3. (II) Modify the `shuffle` program so that it graphs the disorder versus total shuffles using a log-log scale (see the `loglog` MATLAB command). Run your program for N = 30, 100 rounds, 10 shuffles per round and print out your log-log graph.

4. (II) Suppose that initially we flip a coin at each site; with 50% probability we set the site to be “up” and with 50% probability set it to be “down.” What is the probability that a systems of 10 × 10 sites will have the initial state used by the program (i.e., all the sites on the left side are “up” and all on the right side are “down”)? Don’t write a program, work this out by pencil and paper.

5. (III) Modify the `shuffle` program so that shuffling only takes place between adjacent sites. That is, we shuffle by selecting a site at random and then exchanging its value with one of its four nearest neighbors (North, South, East or West). Use periodic boundary conditions so the top row is the South neighbor of the bottom row, the right column is the West neighbor of the left column, etc. Run your program and graph the disorder versus total shuffles for N = 30, 100 rounds, 100 shuffles per round. Note that this type of shuffling is a better model of how mixing and diffusion occur in physical systems.

6. (III) Modify the `shuffle` program so that shuffling occurs by entire rows or columns. At each shuffle, select two rows at random and exchange their sites, then select two columns at random and exchange their sites. Run your program and graph the disorder versus total shuffles for N = 30, 100 rounds, 1 shuffle per round.

Listing

`shuffle.m`

```
% shuffle - Simulation of two dimensional shuffling
clear; help shuffle;  % Clear memory; print header

%@ Initialize variables
NSide = input('Enter the number of sites on a side: ');
NSites = NSide*NSide;      % Total number of sites
% Assign half the sites to have value 1 ('up')
for i=1:NSide
  for j=1:(NSide/2)
    Site(i,j) = 1;
  end
end
% Assign the other half to have value 0 ('down')
```

```
for i=1:NSide
  for j=(NSide/2 + 1):NSide
    Site(i,j) = 0;
  end
end
NRounds = input('Enter the number of rounds: ');
NShuffle = input('Enter number of site shuffles per round: ');
epsilon = 1e-100;   % Small value used to avoid log(0) problems

%@ Prepare for plotting; plot the initial state
clf;  figure(gcf);           % Clear figure; bring window forward
image(Site+1);               % Plot values as an image (adding offset of 1)
axis('ij');  axis('square'); % Set up axis parameters
colormap(hot(3));            % Color map used by image
title('Initial state, Disorder = 0');
drawnow;

%@ Loop over the desired number of rounds of shuffling
for iRound=1:NRounds

  %@ Loop over the number of shuffles for this round
  for iShuffle=1:NShuffle
    %@ Select two site at random
    iSite1 = ceil(rand(1)*NSide);   % Index i of first site
       jSite1 = ceil(rand(1)*NSide);   % Index j of first site
    iSite2 = ceil(rand(1)*NSide);   % Index i of second site
       jSite2 = ceil(rand(1)*NSide);   % Index j of second site
       %@ Exchange the values at the two sites
       Copy1 = Site(iSite1,jSite1);
       Site(iSite1,jSite1) = Site(iSite2,jSite2);
       Site(iSite2,jSite2) = Copy1;
  end

  %@ Sum over the columns to compute disorder
  Disorder(iRound) = 0.0;
  for j=1:NSide
    TotalUp = 0.0;
       %@ Sum within a column to find fraction of sites that are up
       for i=1:NSide
         TotalUp = TotalUp + Site(i,j);  % Running tally of up sites
       end
       FractionUp = TotalUp/NSide;       % Fraction up in this column
       %@ Compute disorder as sum of f*log(f)
       %  Add epsilon in logarithm to avoid problems with log(0)
       Disorder(iRound) = Disorder(iRound) ...
                        - FractionUp*log(FractionUp + epsilon);
  end
  TotalShuffles(iRound) = iRound*NShuffle;  % Record total shuffles for plot

  %@ Plot a tableau showing the sites and display the disorder
  image(Site+1);   % Plot values as an image (adding offset of 1 for colors)
  axis('ij');  axis('square'); % Set up axis parameters
  title(sprintf('Round %g,  Disorder = %g',iRound,Disorder(iRound)));
  drawnow;
end

% Pause before continuing
text(NSide/3,NSide/2,'Strike any key to continue');
pause;

%@ Plot the disorder as a function total shuffles
clf; figure (gcf);   % Clear the figure; bring window forward
plot(TotalShuffles,Disorder,'*');
xlabel('Total Shuffles');   ylabel('Disorder');
```