LOGISTICS $\longrightarrow$ MIDTERMS
- sign-up

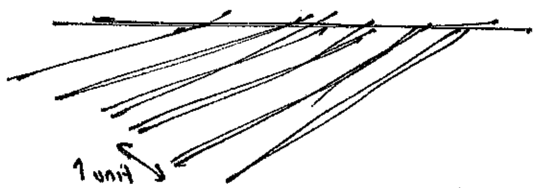
MON. MAY 14 class @ 3:30pm
Reading Room

homework - $\Theta(20)$ lines of code,
but it's a little subtle!

$$\frac{\partial}{\partial t} T(x,t) = D\left(\frac{\partial}{\partial x}\right)^2 T(x,t)$$

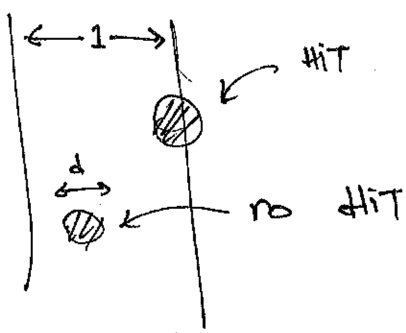
INDEX CARD: coin toss

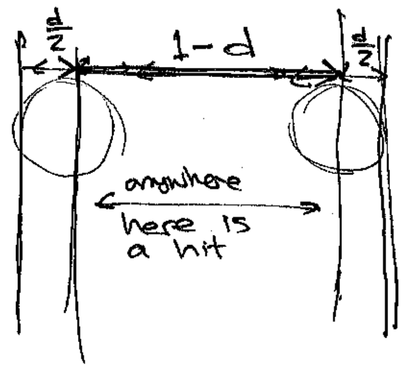
imagine floor w/ slats



toss a coin of
some size d

what's the prob.
that it hits a slat?





eg:

$$\text{So } p(\text{hit}) = \boxed{\frac{d}{1}}$$

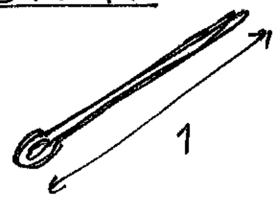
HARDER: Buffon's needle: 1 DIM

What is the PROBABILITY of a needle
crossing the slat?

$\rightarrow$ $\boxed{\text{DEMO}}$

ANALYTIC SOLUTION

take case :

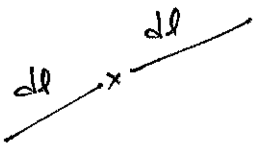


ie same length as slat separation

want P, probability of needle crossing slat

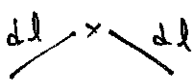
$$P = \frac{\#(\text{all config w/ needle crossing})}{\#(\text{all config})}$$

for small needle (smaller than slats – not worried about 2x crossing)



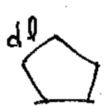
$$P = \underbrace{\overset{\text{PROB FOR UNIT LEN.}}{p}\,dl + p\,dl}_{\text{sum of separate probabilities}}$$

in fact, orientation doesn't matter



$$P = 2p\,dl$$

even closed shapes



$$P = 5p\,dl$$

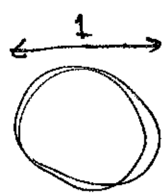
or bent shapes

$$\boxed{P = 2\pi r\, p}$$

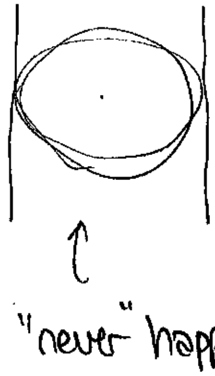
want this

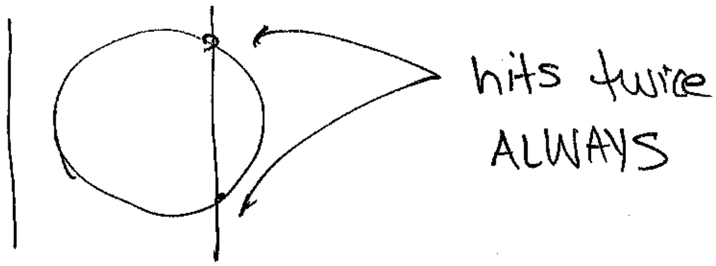
(r =

now consider case of



basically compass!





PROB OF UNIT NEEDLE X-ING ONCE

$$\underbrace{2\pi\left(\tfrac{1}{2}\right) p}_{\#\text{ crossings}} = 2$$

$$\Rightarrow \boxed{p = \frac{2}{\pi}}$$

for more, see HOW NOT TO BE WRONG, ELLENB.

SMAC, KRAUTH

statistical Mechanics in a nutshell

MANY particles, N_0

huge phase space $\leftrightarrow$ STATES

each possible config of positions, momenta, ... is a microstate

WE DON'T CARE ABOUT MICROSTATES, WE CARE ABOUT MACROSCOPIC PROPERTIES.

(time evolution mixes up these states
↳ system "explores" the landscope of states)

A RESULT: LOWER ENERGY STATE $\longrightarrow$ MORE PROBABLE

in fact, PROBABILITY of STATE i DEP. only on ENERGY & TEMP

$$P(E_i) = \frac{e^{-\beta E_i}}{Z} \qquad \beta = \frac{1}{k_B T}$$

$$Z = \sum_i e^{-\beta E_i}$$

PARTITION FUNCTION

then the expectation/average of a quantity X is simply

$$\langle X \rangle = \sum_i X_i P(E_i)$$

if $N_0 = 10^{23}$ & ea particle state has just two possibilities,

$2^{10^{23}}$ terms in sum! :(

MAYBE WE JUST MONTE CARLO OVER SOME STATES

$$\langle X \rangle \simeq \frac{\sum_i^N X_i \, P(E_i)}{\sum_i^N P(E_i)}$$

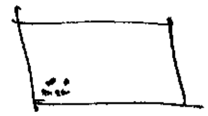
$\leftarrow$ normalization

$$\sum_i^{all} P(E_i) = 1$$

$$\Rightarrow \sum_i^N P(E_i) < 1$$

still a problem :

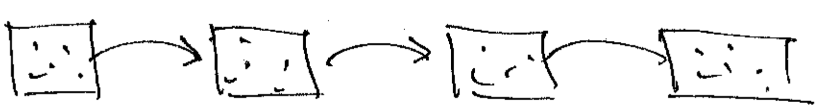
how to sample? random states will be high energy, $e^{-E_i/k_B T} \ll 1$.

like calc pressure by sampling 

so we have to sample intelligently.

proportionally to $P(E_i)$

TIME EVOL. SAMPLES "ALL" STATES BY SEQUENTIAL CHANGES

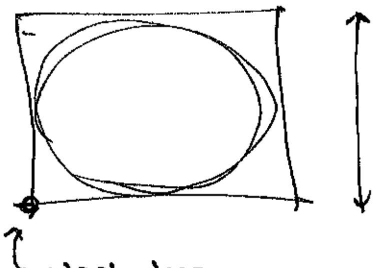


eg of MARKOV CHAIN.

PEBBLE/MARBLE GAME REDUX

↳ from SMAC (Krauth, ch. 1)

NOW IMAGINE:



$$\frac{N_{hit}}{N} = \frac{\pi r^2}{4 r^2}$$

better:

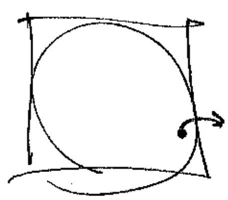
you have one marble to sample.

↳ James' solution: make a mark in the dirt, throw from there.

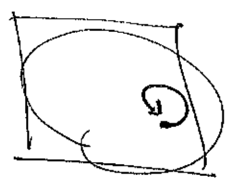
$\boxed{\text{What do we do if we throw the marble out of bounds?}}$

↑ access a "state" that is invalid

ANSWER:



counts as



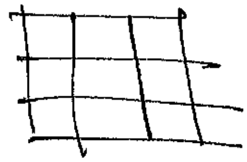
throw lands on same spot

Demo: gives correct answer ($\pi/4$)

Question: $\boxed{\text{Why?}}$ Answer: Metropolis algorithm.

Detailed Balance

simplified model:



marble can move in one unit per turn.

WANT: P() some config

we know answer: <u>should</u> be 1/9 for all

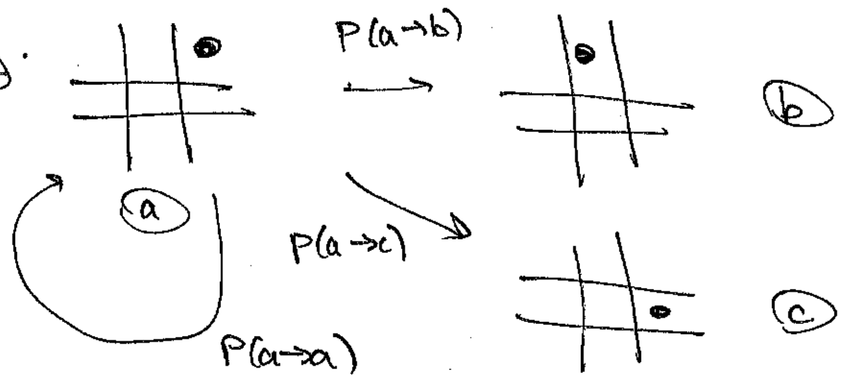
so we need an algorithm s.t moving this ball around will (for large N) lead it to be on each "state" roughly 1/9 of the time

<u>WHAT WE NEED</u>:

rule for how to "throw the marble"

↳ transition probabilities between states

<u>eg.</u>



<u>RULES</u>: cons. of PROB: $p(a\to a) + p(a\to b) + p(a\to c) = 1$

"flow" : $P(a) = P(a)\, p(a\to a) + P(b)\, p(b\to a) + P(c)\, \underbrace{p(c\to a)}_{\text{ASSUME TIME-INDEP!}}$

then :

$$P(a)\,[\underbrace{1 - p(a\to a)}] = P(b)\,p(b\to a) + P(c)\,p(c\to a)$$

$$= p(a\to b) + p(a\to c)$$

$$\Rightarrow \boxed{P(a)\,p(a\to b)} + \boxed{P(a)\,p(a\to c)}$$

$$= \boxed{P(b)\,p(b\to a)} + \boxed{P(c)\,p(c\to a)}$$

one solution (DETAILED BALANCE) :
these boxes are separately equal
→ HOLDS TRUE EVEN IF NOT IN CORNER.

so in the bulk :
WE KNOW the "answer" is $P(\text{any state}) = 1/9$
↳ then we need all $p(x\to y)$ equal to $1/4$
$y \neq x$

that means that corner sites w/ only two neighbors need to have the
$p(\text{corner} \to \text{else}) \equiv 1/4$
ie. half of the time you don't leave
↳ $p(\text{corner} \to \text{itself}) = 1/2$

Lecture 10 Buffon Needle

May 10, 2018

1 Lec 10: Buffon's Needle

```
In [19]: import numpy as np
         import random
         from math import pi, cos

         a = 1 # length of needle (in units of floor crack spacing)
         N = 1000

         Nhits = 0
         for i in range(N):
             x_center = random.uniform(0,1)
             theta = random.uniform(0,2*pi)
             if (x_center + a/2*cos(theta) > 1) or (x_center - a/2*cos(theta) > 1):
                 Nhits += 1
             elif (x_center + a/2*cos(theta) < 0) or (x_center - a/2*cos(theta) < 0):
                 Nhits += 1

         print(Nhits/N)
         print(2/pi)
```

```
0.625
0.6366197723675814
```

Correct answer: $\langle N_{\text{hit}} \rangle = \frac{a}{b}\frac{2}{\pi}$
How would we do it if we didn't know what π is?

```
In [47]: from math import sqrt

         a = 1
         N = 10000

         Nhits = 0
         for i in range(N):
             x = random.uniform(0,1) # only focus on upper right quadrant
             y = random.uniform(0,1)
         #     print(x)
```

```
#       print(y)
        while x**2 + y**2 > 1:
            # print("bad value, remove from statistics") ## do you see why?
            x = random.uniform(0,1)
            y = random.uniform(0,1)
        x_center = random.uniform(.5,1)
        halfacos = (a/2)*x/sqrt(x**2+y**2)
        if x_center + halfacos > 1:
            Nhits += 1

    print(Nhits/N)
    print(2/pi)
```

```
0.6284
0.6366197723675814
```

1.1 Markov Chain Monte Carlo for the marble game

```
In [116]: N = 500000
          side = 100
          rad2 = (side/2)**2
          x = 0
          y = 0
          Nhit = 0

          for i in range(N):
              dx = random.uniform(-1,1)
              dy = random.uniform(-1,1)
              if abs(x+dx) < side/2 and abs(y+dy) < side/2:
                  x += dx
                  y += dy
              if x**2 + y**2 < rad2:
                  Nhit += 1

          print(Nhit/N)
```

```
0.797454
```

```
In [120]: # THIS IS THE WRONG WAY
          N = 500000
          side = 100
          rad2 = (side/2)**2
          x = 0
          y = 0
          Nhit = 0

          for i in range(N):
```

```
        dx = random.uniform(-1,1)
        dy = random.uniform(-1,1)
        while abs(x+dx) > side/2 and abs(y+dy) > side/2:
            dx = random.uniform(-1,1)
            dy = random.uniform(-1,1)
        x += dx
        y += dy
        if x**2 + y**2 < rad2:
            Nhit += 1

print(Nhit/N)
```

```
0.119566
```

```
In [119]: # Correct answer
          print(pi/4)
```

```
0.7853981633974483
```