OLD DOMINION UNIVERSITY

Monte Carlo method II

A. Godunov

1. Monte Carlo integration
2. Random walks
3. Stochastic search and optimization

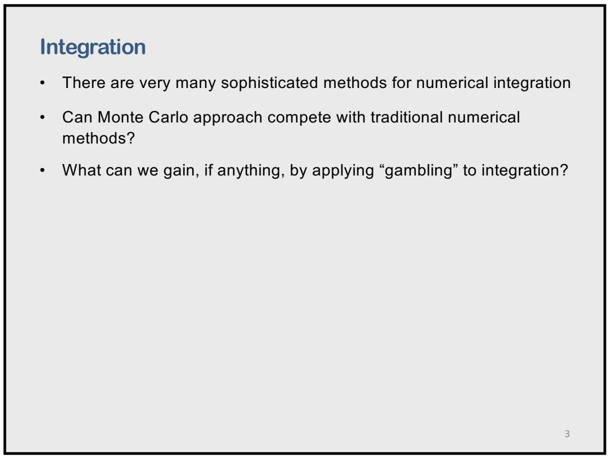
updated 4 February 2022

1

Part 1:

Monte Carlo integration

2



Integration

- There are very many sophisticated methods for numerical integration
- Can Monte Carlo approach compete with traditional numerical methods?
- What can we gain, if anything, by applying "gambling" to integration?

3

3

There is clearly a problem with nD integration

Example: Integration for a system with 12 electrons.

- $3 * 12 = 36$ dimensional integral
- If 64 points for each integration then $= 64^{36}$ points to evaluate
- For 1 Tera Flop computer = 10^{53} seconds
- That is … 3 times more then the age of the universe!

4

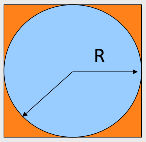
Two methods for MC integration

1. Monte Carlo Integration by "Stone Throwing" or "hit and miss" method.
2. Mean Value Integration (with many variations).

5

Calculating an area as hit and miss

Imagine that we need to evaluate the area of a circle (or any other shape)



Let the circle has a radius R

We draw a square box $R \times R$ that completely encloses the circle (thus the area of the box if R^2)

Then we set a counter n=0 and do a loop over N trials

1. We generate two random numbers x_i and y_i
2. If $x_i^2 + y_i^2 < R^2$ then the point is inside the circle and $n = n + 1$

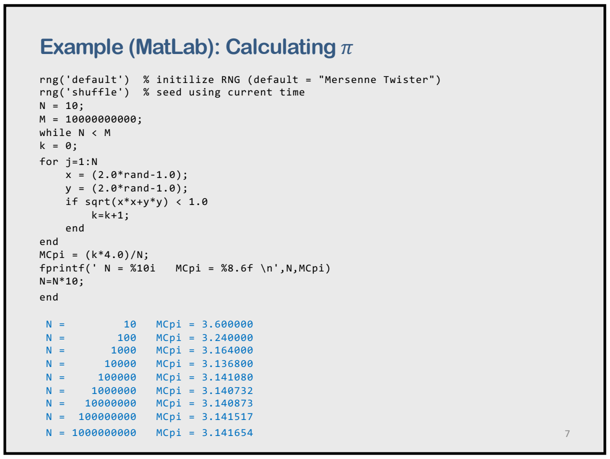
Since $A_{circle}/A\_(square) \sim n/N$ for large enough N we have

$$A_{circle} \approx \frac{n}{N} A_{square} \approx \frac{n}{N} 4R^2$$

Note: we can even find π this way

$$\pi \approx 4n/N$$

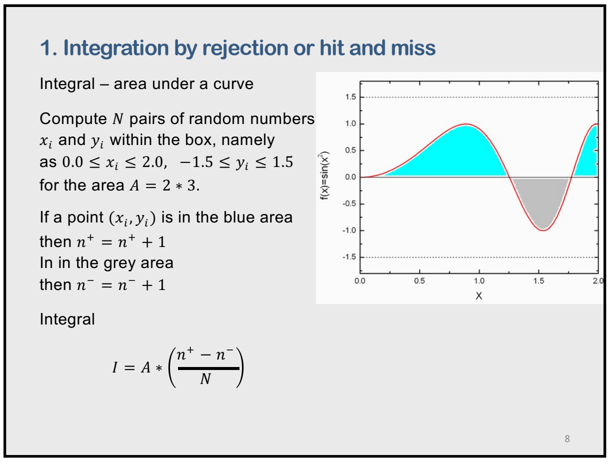
6



Example (MatLab): Calculating π

```
rng('default')  % initilize RNG (default = "Mersenne Twister")
rng('shuffle')  % seed using current time
N = 10;
M = 10000000000;
while N < M
k = 0;
for j=1:N
    x = (2.0*rand-1.0);
    y = (2.0*rand-1.0);
    if sqrt(x*x+y*y) < 1.0
        k=k+1;
    end
end
MCpi = (k*4.0)/N;
fprintf(' N = %10i   MCpi = %8.6f \n',N,MCpi)
N=N*10;
end
```

```
 N =         10   MCpi = 3.600000
 N =        100   MCpi = 3.240000
 N =       1000   MCpi = 3.164000
 N =      10000   MCpi = 3.136800
 N =     100000   MCpi = 3.141080
 N =    1000000   MCpi = 3.140732
 N =   10000000   MCpi = 3.140873
 N =  100000000   MCpi = 3.141517
 N = 1000000000   MCpi = 3.141654
```



1. Integration by rejection or hit and miss

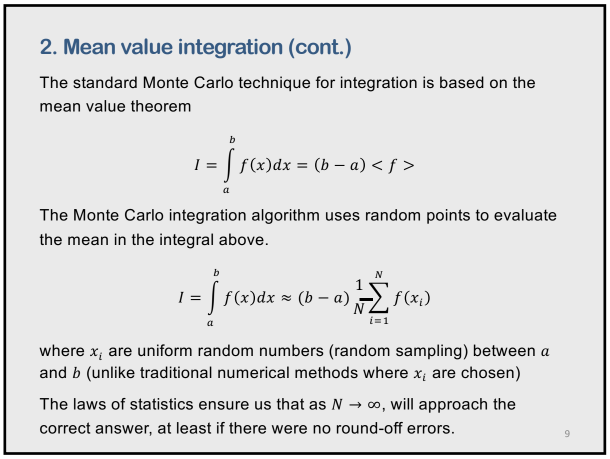
Integral – area under a curve

Compute N pairs of random numbers x_i and y_i within the box, namely as $0.0 \leq x_i \leq 2.0, \ \ -1.5 \leq y_i \leq 1.5$ for the area $A = 2 * 3$.

If a point (x_i, y_i) is in the blue area then $n^+ = n^+ + 1$
In in the grey area then $n^- = n^- + 1$

Integral

$$I = A * \left(\frac{n^+ - n^-}{N}\right)$$



2. Mean value integration (cont.)

The standard Monte Carlo technique for integration is based on the mean value theorem

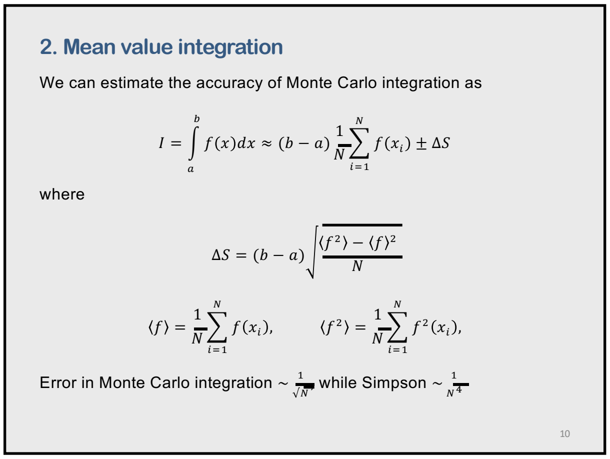
$$I = \int_a^b f(x)dx = (b - a) < f >$$

The Monte Carlo integration algorithm uses random points to evaluate the mean in the integral above.

$$I = \int_a^b f(x)dx \approx (b - a) \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

where x_i are uniform random numbers (random sampling) between a and b (unlike traditional numerical methods where x_i are chosen)

The laws of statistics ensure us that as $N \to \infty$, will approach the correct answer, at least if there were no round-off errors.



2. Mean value integration

We can estimate the accuracy of Monte Carlo integration as

$$I = \int_a^b f(x)dx \approx (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i) \pm \Delta S$$

where

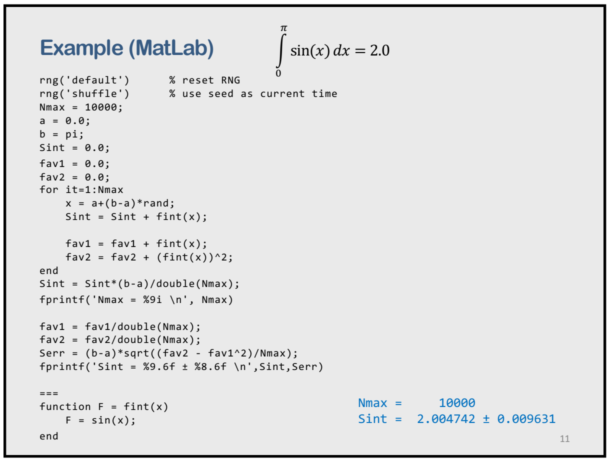
$$\Delta S = (b-a)\sqrt{\frac{\langle f^2\rangle - \langle f\rangle^2}{N}}$$

$$\langle f\rangle = \frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad \langle f^2\rangle = \frac{1}{N}\sum_{i=1}^{N} f^2(x_i),$$

Error in Monte Carlo integration ~ $\frac{1}{\sqrt{N}}$, while Simpson ~ $\frac{1}{N^4}$

10

10



Example (MatLab)

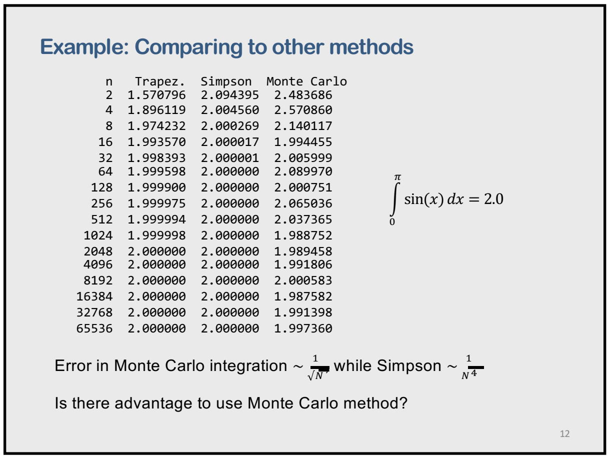
$$\int_0^\pi \sin(x)\,dx = 2.0$$

```
rng('default')      % reset RNG
rng('shuffle')      % use seed as current time
Nmax = 10000;
a = 0.0;
b = pi;
Sint = 0.0;
fav1 = 0.0;
fav2 = 0.0;
for it=1:Nmax
    x = a+(b-a)*rand;
    Sint = Sint + fint(x);

    fav1 = fav1 + fint(x);
    fav2 = fav2 + (fint(x))^2;
end
Sint = Sint*(b-a)/double(Nmax);
fprintf('Nmax = %9i \n', Nmax)

fav1 = fav1/double(Nmax);
fav2 = fav2/double(Nmax);
Serr = (b-a)*sqrt((fav2 - fav1^2)/Nmax);
fprintf('Sint = %9.6f ± %8.6f \n',Sint,Serr)

===
function F = fint(x)
    F = sin(x);
end
```

```
Nmax =     10000
Sint =  2.004742 ± 0.009631
```

11

11



Example: Comparing to other methods

```
    n   Trapez.  Simpson  Monte Carlo
    2  1.570796  2.094395  2.483686
    4  1.896119  2.004560  2.570860
    8  1.974232  2.000269  2.140117
   16  1.993570  2.000017  1.994455
   32  1.998393  2.000001  2.005999
   64  1.999598  2.000000  2.089970
  128  1.999900  2.000000  2.000751
  256  1.999975  2.000000  2.065036
  512  1.999994  2.000000  2.037365
 1024  1.999998  2.000000  1.988752
 2048  2.000000  2.000000  1.989458
 4096  2.000000  2.000000  1.991806
 8192  2.000000  2.000000  2.000583
16384  2.000000  2.000000  1.987582
32768  2.000000  2.000000  1.991398
65536  2.000000  2.000000  1.997360
```

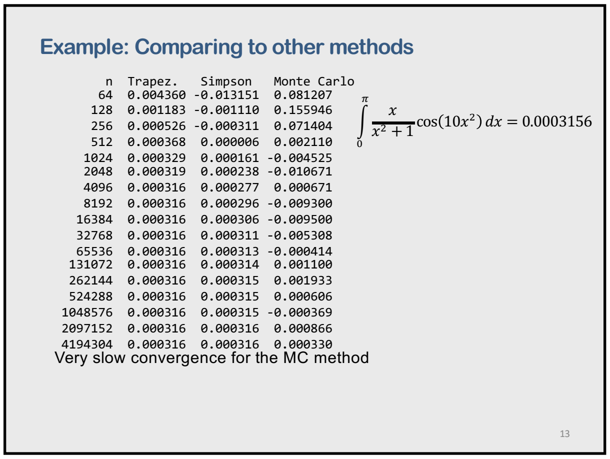
$$\int_0^\pi \sin(x)\,dx = 2.0$$

Error in Monte Carlo integration ~ $\frac{1}{\sqrt{N}}$, while Simpson ~ $\frac{1}{N^4}$

Is there advantage to use Monte Carlo method?

12

12



Example: Comparing to other methods

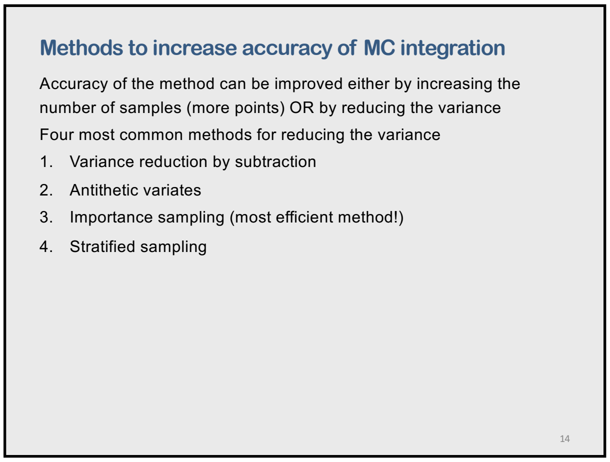
```
      n  Trapez.   Simpson   Monte Carlo
     64  0.004360 -0.013151  0.081207
    128  0.001183 -0.001110  0.155946
    256  0.000526 -0.000311  0.071404
    512  0.000368  0.000006  0.002110
   1024  0.000329  0.000161 -0.004525
   2048  0.000319  0.000238 -0.010671
   4096  0.000316  0.000277  0.000671
   8192  0.000316  0.000296 -0.009300
  16384  0.000316  0.000306 -0.009500
  32768  0.000316  0.000311 -0.005308
  65536  0.000316  0.000313 -0.000414
 131072  0.000316  0.000314  0.001100
 262144  0.000316  0.000315  0.001933
 524288  0.000316  0.000315  0.000606
1048576  0.000316  0.000315 -0.000369
2097152  0.000316  0.000316  0.000866
4194304  0.000316  0.000316  0.000330
```

$$\int_{0}^{\pi} \frac{x}{x^2+1}\cos(10x^2)\,dx = 0.0003156$$

Very slow convergence for the MC method

13

13



Methods to increase accuracy of MC integration

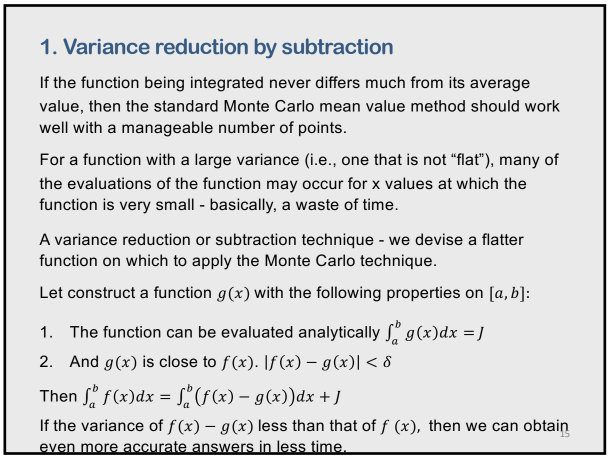
Accuracy of the method can be improved either by increasing the number of samples (more points) OR by reducing the variance

Four most common methods for reducing the variance

1. Variance reduction by subtraction
2. Antithetic variates
3. Importance sampling (most efficient method!)
4. Stratified sampling

14

14



1. Variance reduction by subtraction

If the function being integrated never differs much from its average value, then the standard Monte Carlo mean value method should work well with a manageable number of points.

For a function with a large variance (i.e., one that is not "flat"), many of the evaluations of the function may occur for x values at which the function is very small - basically, a waste of time.

A variance reduction or subtraction technique - we devise a flatter function on which to apply the Monte Carlo technique.

Let construct a function $g(x)$ with the following properties on $[a,b]$:

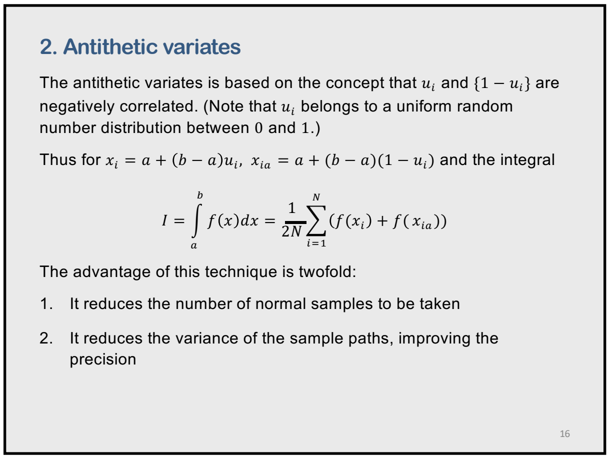
1. The function can be evaluated analytically $\int_a^b g(x)dx = J$
2. And $g(x)$ is close to $f(x)$. $|f(x) - g(x)| < \delta$

Then $\int_a^b f(x)dx = \int_a^b \big(f(x) - g(x)\big)dx + J$

If the variance of $f(x) - g(x)$ less than that of $f\ (x)$, then we can obtain even more accurate answers in less time.

15

15



2. Antithetic variates

The antithetic variates is based on the concept that u_i and $\{1 - u_i\}$ are negatively correlated. (Note that u_i belongs to a uniform random number distribution between 0 and 1.)

Thus for $x_i = a + (b - a)u_i,\ x_{ia} = a + (b - a)(1 - u_i)$ and the integral

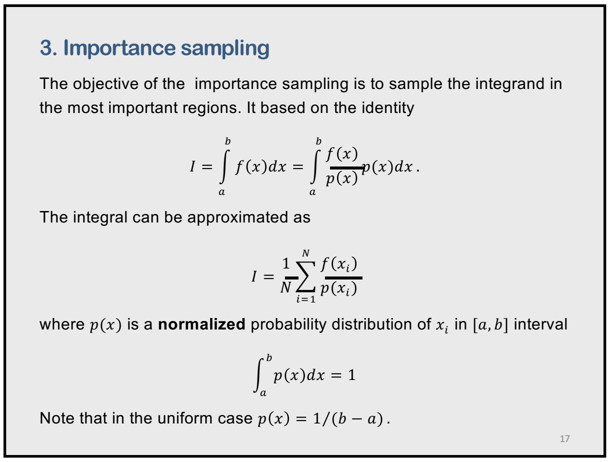
$$I = \int_a^b f(x)dx = \frac{1}{2N}\sum_{i=1}^{N}(f(x_i) + f(x_{ia}))$$

The advantage of this technique is twofold:

1. It reduces the number of normal samples to be taken
2. It reduces the variance of the sample paths, improving the precision

16

16



3. Importance sampling

The objective of the importance sampling is to sample the integrand in the most important regions. It based on the identity

$$I = \int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)}p(x)dx\,.$$

The integral can be approximated as

$$I = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

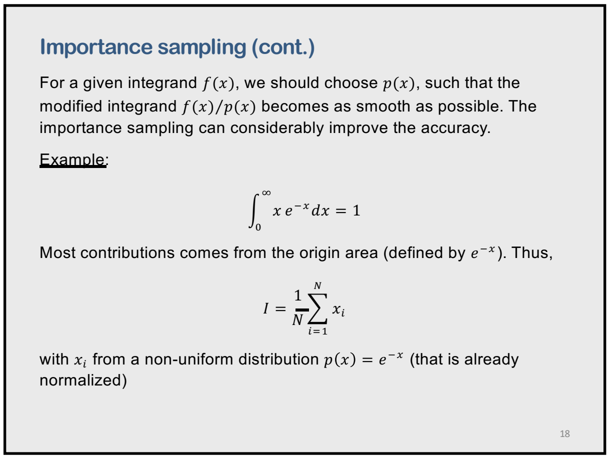
where $p(x)$ is a **normalized** probability distribution of x_i in $[a, b]$ interval

$$\int_a^b p(x)dx = 1$$

Note that in the uniform case $p(x) = 1/(b - a)$.

17

17



Importance sampling (cont.)

For a given integrand $f(x)$, we should choose $p(x)$, such that the modified integrand $f(x)/p(x)$ becomes as smooth as possible. The importance sampling can considerably improve the accuracy.

Example:

$$\int_0^\infty x\,e^{-x}dx = 1$$

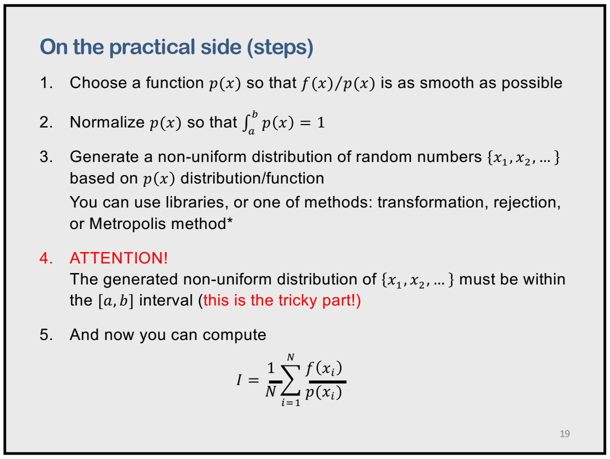
Most contributions comes from the origin area (defined by e^{-x}). Thus,

$$I = \frac{1}{N}\sum_{i=1}^{N}x_i$$

with x_i from a non-uniform distribution $p(x) = e^{-x}$ (that is already normalized)

18

18



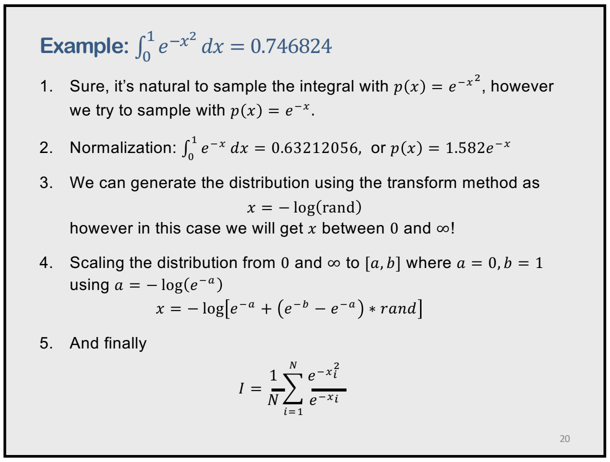
On the practical side (steps)

1. Choose a function $p(x)$ so that $f(x)/p(x)$ is as smooth as possible
2. Normalize $p(x)$ so that $\int_a^b p(x) = 1$
3. Generate a non-uniform distribution of random numbers $\{x_1, x_2, \dots\}$ based on $p(x)$ distribution/function
   You can use libraries, or one of methods: transformation, rejection, or Metropolis method*
4. ATTENTION!
   The generated non-uniform distribution of $\{x_1, x_2, \dots\}$ must be within the $[a, b]$ interval (this is the tricky part!)
5. And now you can compute

$$I = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

19

19



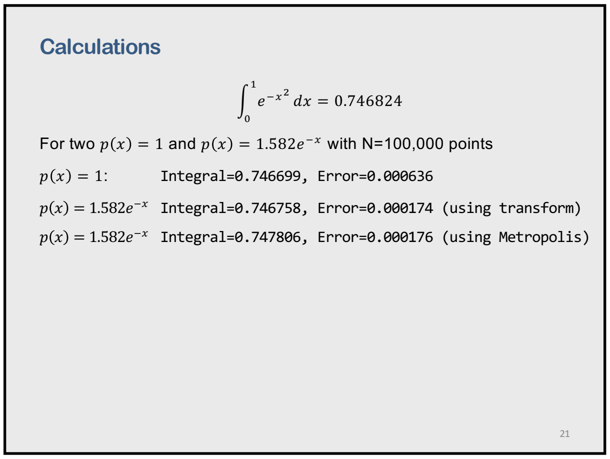
Example: $\int_0^1 e^{-x^2}\,dx = 0.746824$

1. Sure, it's natural to sample the integral with $p(x) = e^{-x^2}$, however we try to sample with $p(x) = e^{-x}$.
2. Normalization: $\int_0^1 e^{-x}\,dx = 0.63212056,$ or $p(x) = 1.582e^{-x}$
3. We can generate the distribution using the transform method as
   $$x = -\log(\mathrm{rand})$$
   however in this case we will get x between 0 and ∞!
4. Scaling the distribution from 0 and ∞ to $[a, b]$ where $a = 0, b = 1$ using $a = -\log(e^{-a})$
   $$x = -\log\left[e^{-a} + \left(e^{-b} - e^{-a}\right) * rand\right]$$
5. And finally

$$I = \frac{1}{N}\sum_{i=1}^{N}\frac{e^{-x_i^2}}{e^{-x_i}}$$

20

20



Calculations

$$\int_0^1 e^{-x^2}\,dx = 0.746824$$

For two $p(x) = 1$ and $p(x) = 1.582e^{-x}$ with N=100,000 points

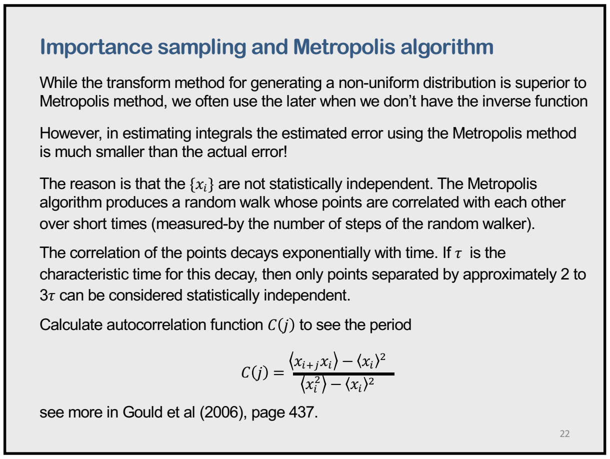
$p(x) = 1$: Integral=0.746699, Error=0.000636

$p(x) = 1.582e^{-x}$ Integral=0.746758, Error=0.000174 (using transform)

$p(x) = 1.582e^{-x}$ Integral=0.747806, Error=0.000176 (using Metropolis)

21

21



Importance sampling and Metropolis algorithm

While the transform method for generating a non-uniform distribution is superior to Metropolis method, we often use the later when we don't have the inverse function

However, in estimating integrals the estimated error using the Metropolis method is much smaller than the actual error!

The reason is that the $\{x_i\}$ are not statistically independent. The Metropolis algorithm produces a random walk whose points are correlated with each other over short times (measured-by the number of steps of the random walker).

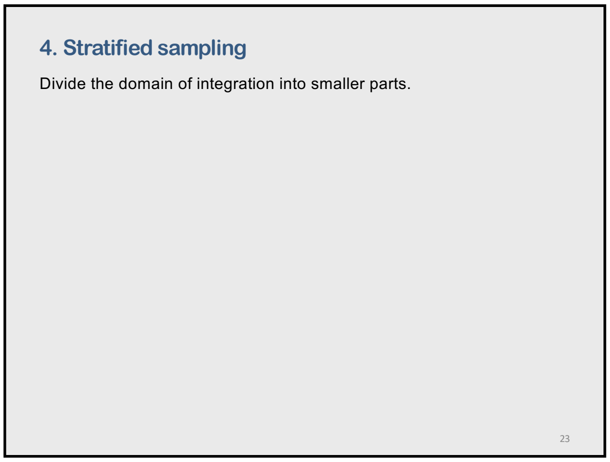
The correlation of the points decays exponentially with time. If τ is the characteristic time for this decay, then only points separated by approximately 2 to 3τ can be considered statistically independent.

Calculate autocorrelation function $C(j)$ to see the period

$$C(j) = \frac{\langle x_{i+j} x_i \rangle - \langle x_i \rangle^2}{\langle x_i^2 \rangle - \langle x_i \rangle^2}$$

see more in Gould et al (2006), page 437.

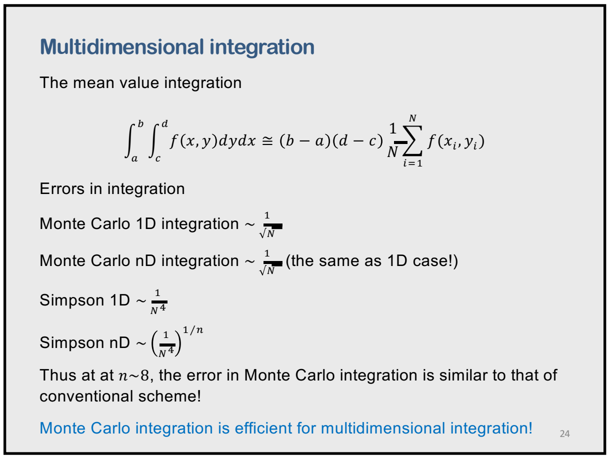
22



4. Stratified sampling

Divide the domain of integration into smaller parts.

23



Multidimensional integration

The mean value integration

$$\int_a^b \int_c^d f(x,y)dydx \cong (b-a)(d-c)\frac{1}{N}\sum_{i=1}^{N} f(x_i, y_i)$$

Errors in integration

Monte Carlo 1D integration ~ $\frac{1}{\sqrt{N}}$

Monte Carlo nD integration ~ $\frac{1}{\sqrt{N}}$ (the same as 1D case!)

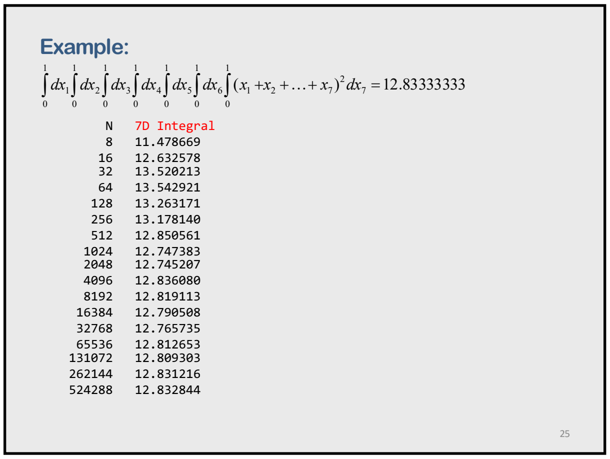
Simpson 1D ~ $\frac{1}{N^4}$

Simpson nD ~ $\left(\frac{1}{N^4}\right)^{1/n}$

Thus at at $n{\sim}8$, the error in Monte Carlo integration is similar to that of conventional scheme!

Monte Carlo integration is efficient for multidimensional integration!

24

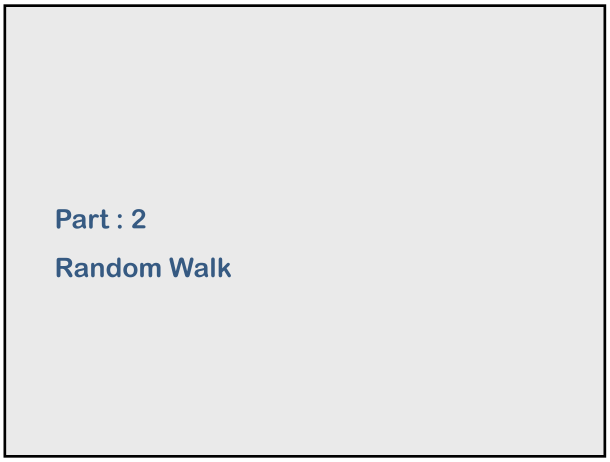


Example:

$$\int_0^1 dx_1 \int_0^1 dx_2 \int_0^1 dx_3 \int_0^1 dx_4 \int_0^1 dx_5 \int_0^1 dx_6 \int_0^1 (x_1 + x_2 + \ldots + x_7)^2 dx_7 = 12.83333333$$

```
     N   7D Integral
     8   11.478669
    16   12.632578
    32   13.520213
    64   13.542921
   128   13.263171
   256   13.178140
   512   12.850561
  1024   12.747383
  2048   12.745207
  4096   12.836080
  8192   12.819113
 16384   12.790508
 32768   12.765735
 65536   12.812653
131072   12.809303
262144   12.831216
524288   12.832844
```

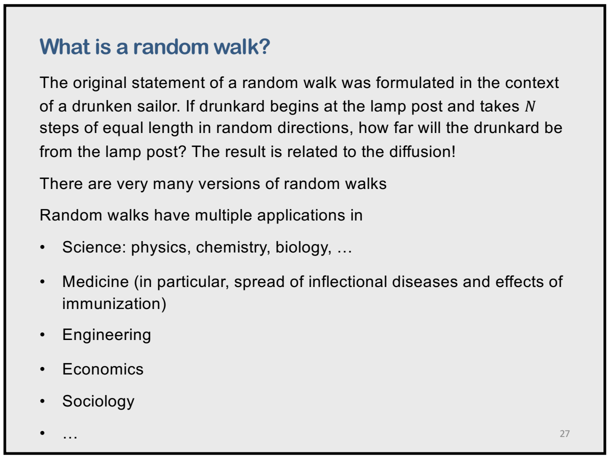
25



Part : 2

Random Walk

26



What is a random walk?

The original statement of a random walk was formulated in the context of a drunken sailor. If drunkard begins at the lamp post and takes N steps of equal length in random directions, how far will the drunkard be from the lamp post? The result is related to the diffusion!

There are very many versions of random walks

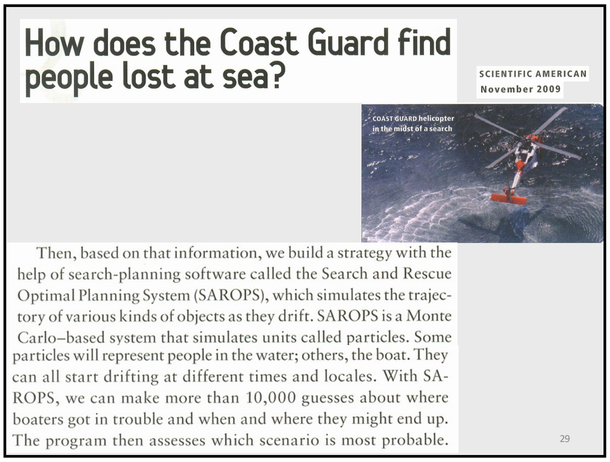
Random walks have multiple applications in

- Science: physics, chemistry, biology, …
- Medicine (in particular, spread of inflectional diseases and effects of immunization)
- Engineering
- Economics
- Sociology
- …

27

NATIONAL BESTSELLER
The Drunkard's Walk
How Randomness Rules Our Lives
Leonard Mlodinow
Elements of the Random Walk
An Introduction for Advanced Students and Researchers
Joseph Rudnick and George Gaspari
Graduate Texts in Mathematics
Frank Spitzer
Principles of Random Walk
Second Edition
Springer
DYNAMIC RANDOM WALKS
THEORY AND APPLICATIONS
Nadine Guillotin-Plantard
René Schott
Random Walk and the Heat Equation
Gregory F. Lawler
AMS
An Unbounded Experience in Random Walks with Applications
Michael F Shlesinger
World Scientific
First Steps in Random Walks
From Tools to Applications
J. Klafter and I. M. Sokolov
OXFORD
Random Walks in Biology
Howard C. Berg
New, expanded edition

28



How does the Coast Guard find people lost at sea?

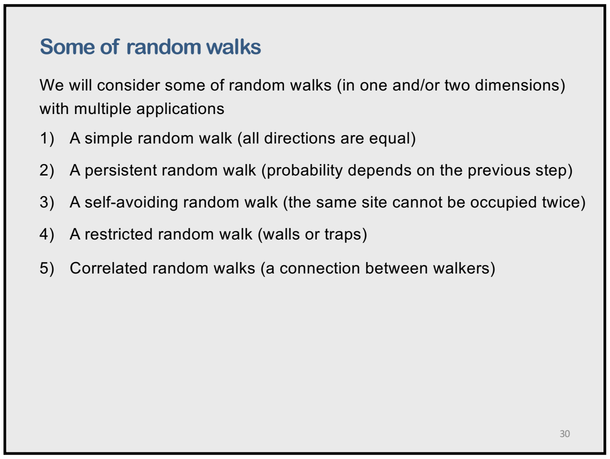
SCIENTIFIC AMERICAN
November 2009

COAST GUARD helicopter in the midst of a search

Then, based on that information, we build a strategy with the help of search-planning software called the Search and Rescue Optimal Planning System (SAROPS), which simulates the trajectory of various kinds of objects as they drift. SAROPS is a Monte Carlo–based system that simulates units called particles. Some particles will represent people in the water; others, the boat. They can all start drifting at different times and locales. With SAROPS, we can make more than 10,000 guesses about where boaters got in trouble and when and where they might end up. The program then assesses which scenario is most probable.

29

29



Some of random walks

We will consider some of random walks (in one and/or two dimensions) with multiple applications

1) A simple random walk (all directions are equal)
2) A persistent random walk (probability depends on the previous step)
3) A self-avoiding random walk (the same site cannot be occupied twice)
4) A restricted random walk (walls or traps)
5) Correlated random walks (a connection between walkers)

30

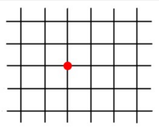
30

1) A Simple random walk

A simple random walk is a sequence of unit steps where each step is taken in the direction of one of the coordinate axis, and each possible direction has equal probability of being chosen.

In one dimension 1D random walk there are two possible directions (left and right)

In two dimensions 2D there are four possible directions, e.g., a single step starting at the point with integer coordinates (x, y) would be equally likely to move to any of one of the four neighbors $(x+1, y), (x-1, y), (x, y+1)$ or $(x, y-1)$.



31

1D Random simple walk

A particle (the walker) starts at the origin $(x = 0)$, and then steps (same length) are chosen randomly left or right with the same probability.

After N steps a position can be recorded as a function of N.

Evaluating the average distance form the starting after many trials would give (the result can easily be derived using that each step is random and it is independent from a previous step)

$$< x > \approx 0 \quad \text{and} \quad < x^2 > \sim N.$$

In many physical processes (such as the motion of a molecule in solution), the time between steps is approximately a constant, so that number of steps is roughly a proportional to time, then we can write

$$< x^2 > \sim Dt,$$

where the factor D is the diffusion constant.

32

1D Random simple walk and diffusion

The one-dimensional diffusion equation can be written as

$$\frac{\partial p(x,t)}{\partial t} = D\,\frac{\partial^2 p(x,t)}{\partial x^2}$$

where D is the self-diffusion coefficient, and $p(x,t)dx$ is the probability of a particle being in the interval between x and $x + dx$ at time t.

The solution gives

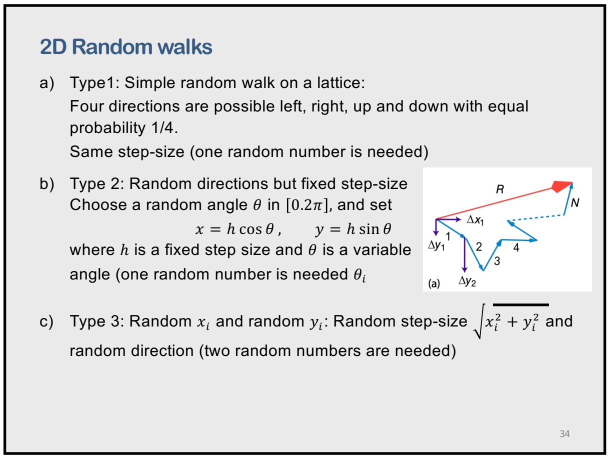
$$\langle x^2(t) \rangle = 2Dt$$

We can see that the random walk method gives the same time dependence.

While the diffusion equation can be solved numerically (e.g. Crank-Nicholson method), it can be very challenging to treat complicated boundary conditions.

Formulating the diffusion problem as a random walk is straightforward to incorporate various boundary conditions.

33



2D Random walks

a) Type1: Simple random walk on a lattice:
Four directions are possible left, right, up and down with equal probability 1/4.
Same step-size (one random number is needed)

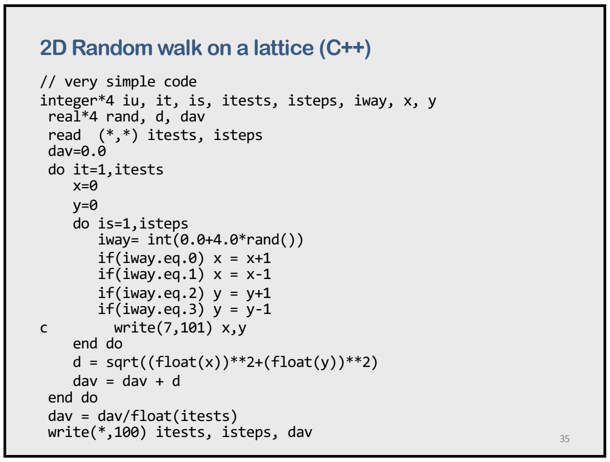
b) Type 2: Random directions but fixed step-size
Choose a random angle θ in $[0.2\pi]$, and set

$$x = h\cos\theta\,, \qquad y = h\sin\theta$$

where h is a fixed step size and θ is a variable angle (one random number is needed θ_i

c) Type 3: Random x_i and random y_i: Random step-size $\sqrt{x_i^2 + y_i^2}$ and random direction (two random numbers are needed)

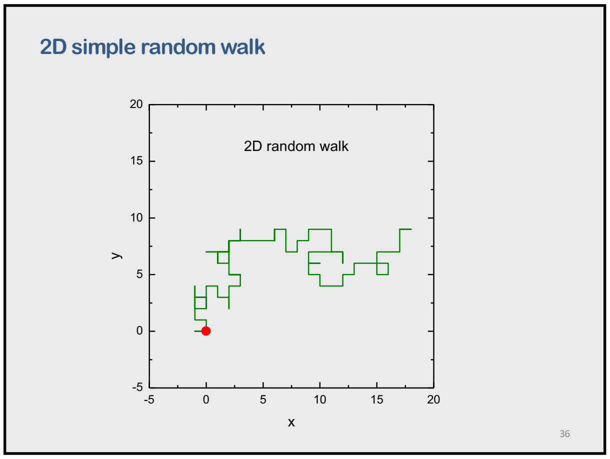
34



2D Random walk on a lattice (C++)

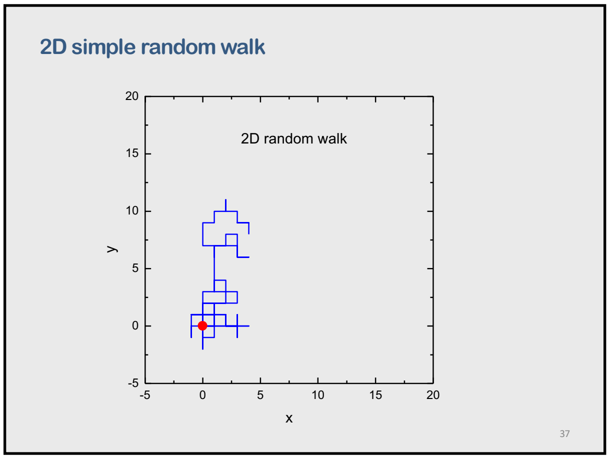
```
// very simple code
integer*4 iu, it, is, itests, isteps, iway, x, y
 real*4 rand, d, dav
 read  (*,*) itests, isteps
 dav=0.0
 do it=1,itests
    x=0
    y=0
    do is=1,isteps
       iway= int(0.0+4.0*rand())
       if(iway.eq.0) x = x+1
       if(iway.eq.1) x = x-1
       if(iway.eq.2) y = y+1
       if(iway.eq.3) y = y-1
c         write(7,101) x,y
    end do
    d = sqrt((float(x))**2+(float(y))**2)
    dav = dav + d
 end do
 dav = dav/float(itests)
 write(*,100) itests, isteps, dav
```

35



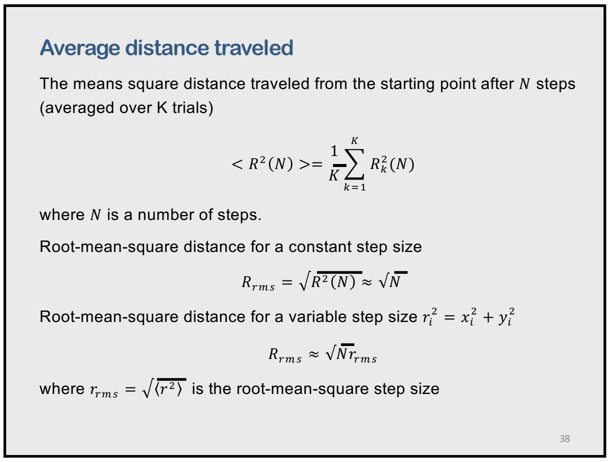
2D simple random walk

36



2D simple random walk

37



Average distance traveled

The means square distance traveled from the starting point after N steps (averaged over K trials)

$$< R^2(N) >= \frac{1}{K}\sum_{k=1}^{K} R_k^2(N)$$

where N is a number of steps.

Root-mean-square distance for a constant step size

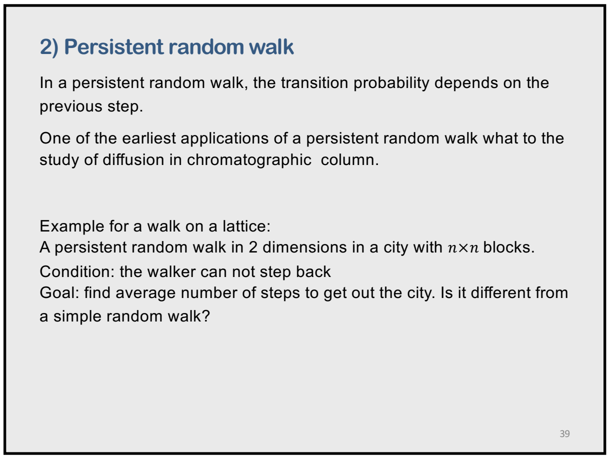
$$R_{rms} = \sqrt{\overline{R^2(N)}} \approx \sqrt{N}$$

Root-mean-square distance for a variable step size $r_i^2 = x_i^2 + y_i^2$

$$R_{rms} \approx \sqrt{N} r_{rms}$$

where $r_{rms} = \sqrt{\langle r^2 \rangle}$ is the root-mean-square step size

38



2) Persistent random walk

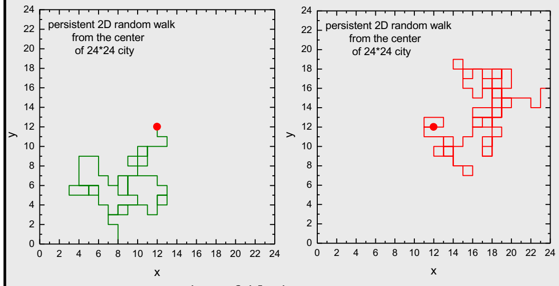
In a persistent random walk, the transition probability depends on the previous step.

One of the earliest applications of a persistent random walk what to the study of diffusion in chromatographic column.

Example for a walk on a lattice:
A persistent random walk in 2 dimensions in a city with $n \times n$ blocks.
Condition: the walker can not step back
Goal: find average number of steps to get out the city. Is it different from a simple random walk?

39

example



```
Average number of blocks to go to
leave the city with 24*24 blocks
from the center:     92 blocks
from a random point: 47 blocks
```

40

40

3) Self avoiding random walk

Example: using random walk for studying protein growth

Note: A protein is a large biological molecule made up of molecular chains (the residues of amino acids). These chains are formed from monomers, that is, molecules that bind chemically with other molecules.

Random walk is perfectly suited for modelling protein grows.

However, the walk is restricted such that the only positions available after each step are the three neighboring sites (if random walk on a lattice), with the already-occupied sites excluded

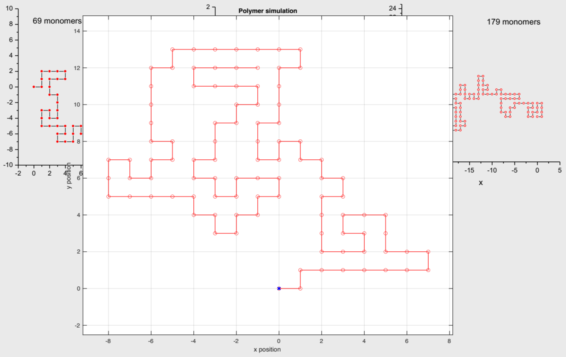
This is why this technique is known as a self-avoiding random walk.

Attention: the walk stops when there are no empty neighboring sites available.

41

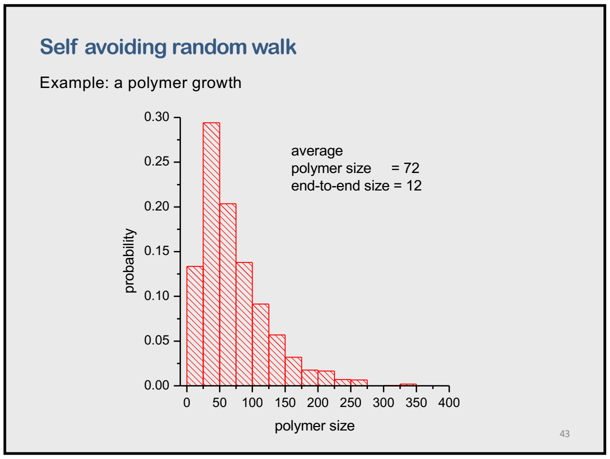
41

Examples for self avoiding RW on lattice



42

42

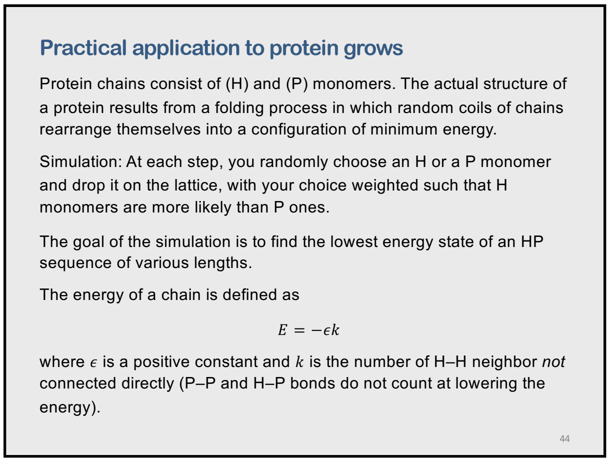


Self avoiding random walk

Example: a polymer growth

43

43



Practical application to protein grows

Protein chains consist of (H) and (P) monomers. The actual structure of a protein results from a folding process in which random coils of chains rearrange themselves into a configuration of minimum energy.

Simulation: At each step, you randomly choose an H or a P monomer and drop it on the lattice, with your choice weighted such that H monomers are more likely than P ones.

The goal of the simulation is to find the lowest energy state of an HP sequence of various lengths.

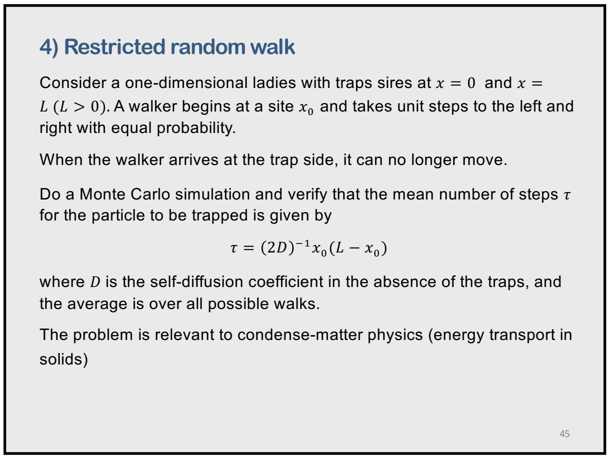
The energy of a chain is defined as

$$E = -\epsilon k$$

where ϵ is a positive constant and k is the number of H–H neighbor *not* connected directly (P–P and H–P bonds do not count at lowering the energy).

44

44



4) Restricted random walk

Consider a one-dimensional ladies with traps sires at $x = 0$ and $x = L\ (L > 0)$. A walker begins at a site x_0 and takes unit steps to the left and right with equal probability.

When the walker arrives at the trap side, it can no longer move.

Do a Monte Carlo simulation and verify that the mean number of steps τ for the particle to be trapped is given by

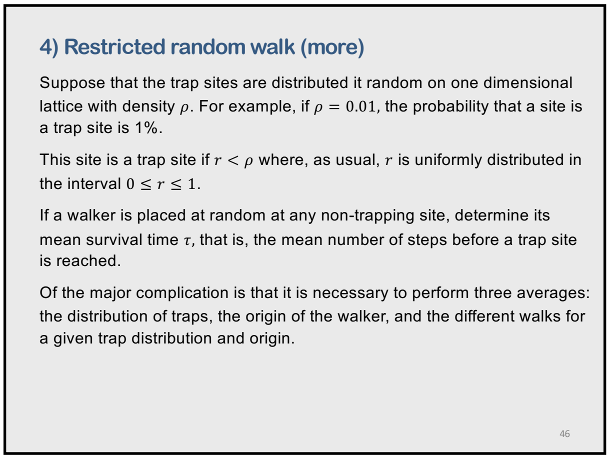
$$\tau = (2D)^{-1} x_0 (L - x_0)$$

where D is the self-diffusion coefficient in the absence of the traps, and the average is over all possible walks.

The problem is relevant to condense-matter physics (energy transport in solids)

45

45



4) Restricted random walk (more)

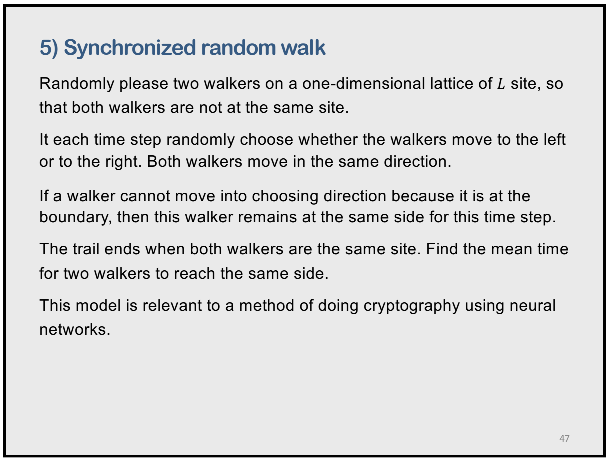
Suppose that the trap sites are distributed it random on one dimensional lattice with density ρ. For example, if $\rho = 0.01$, the probability that a site is a trap site is 1%.

This site is a trap site if $r < \rho$ where, as usual, r is uniformly distributed in the interval $0 \leq r \leq 1$.

If a walker is placed at random at any non-trapping site, determine its mean survival time τ, that is, the mean number of steps before a trap site is reached.

Of the major complication is that it is necessary to perform three averages: the distribution of traps, the origin of the walker, and the different walks for a given trap distribution and origin.

46



5) Synchronized random walk

Randomly please two walkers on a one-dimensional lattice of L site, so that both walkers are not at the same site.

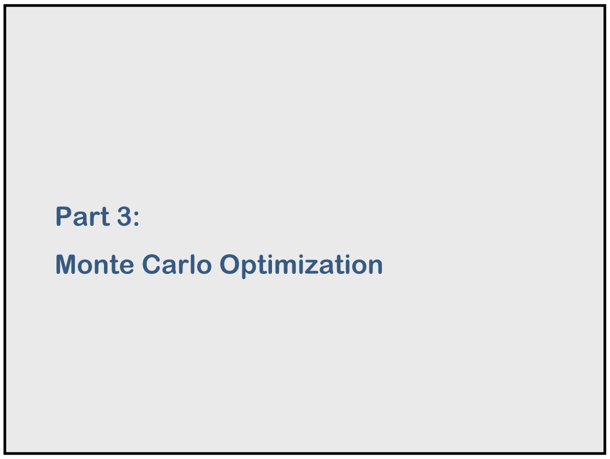
It each time step randomly choose whether the walkers move to the left or to the right. Both walkers move in the same direction.

If a walker cannot move into choosing direction because it is at the boundary, then this walker remains at the same side for this time step.

The trail ends when both walkers are the same site. Find the mean time for two walkers to reach the same side.

This model is relevant to a method of doing cryptography using neural networks.

47



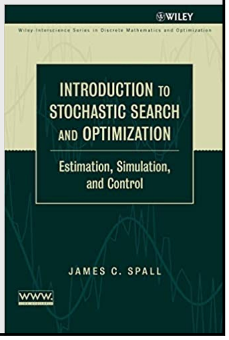
Part 3:

Monte Carlo Optimization

48

Areas

- Stochastic optimization
  or the problem of local minima.
- Swarm intelligence
  or the ant colony optimization
- Genetic algorithms
  Use Darwinian evolution of a gene pool to find the fittest genes
- Simulated annealing
- and many more …


WILEY
Wiley-Interscience Series in Discrete Mathematics and Optimization
INTRODUCTION TO STOCHASTIC SEARCH AND OPTIMIZATION
Estimation, Simulation, and Control
JAMES C. SPALL
WWW.

49

Example of Problems Using Stochastic Search and Optimization

- Minimize the costs of shipping from production facilities to warehouses
- Maximize the probability of detecting an incoming warhead (vs. decoy) in a missile defense system
- Place sensors in manner to maximize useful information
- Determine the times to administer a sequence of drugs for maximum therapeutic effect
- Find the best red-yellow-green signal timings in an urban traffic network
- Determine the best schedule for use of laboratory facilities to serve an organization's overall interests

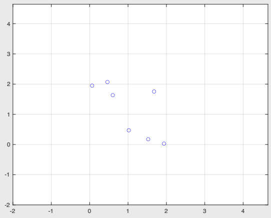
50

50

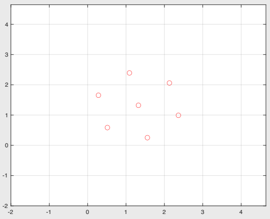
Example 1

Find a configuration of 7 particles (interacting by Lenard-Jones potential) that has the lowest energy

Initial (random) configuration.

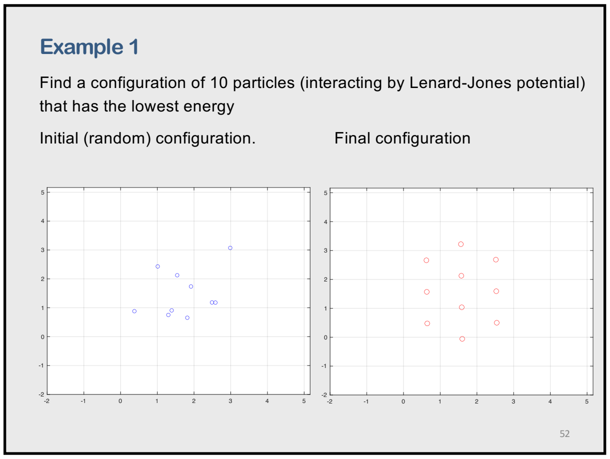


Final configuration



51

51



Example 1

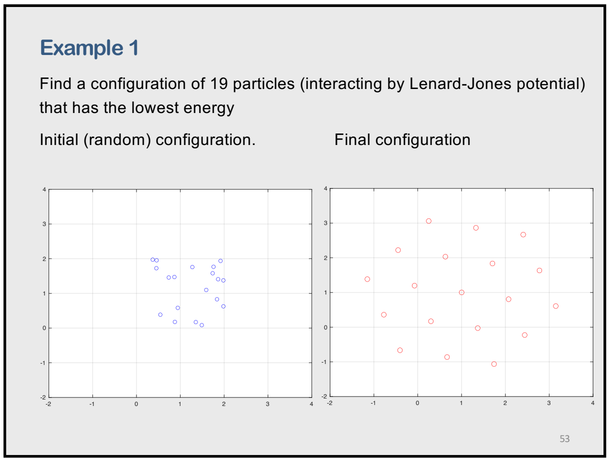
Find a configuration of 10 particles (interacting by Lenard-Jones potential) that has the lowest energy

Initial (random) configuration.

Final configuration

52

52



Example 1

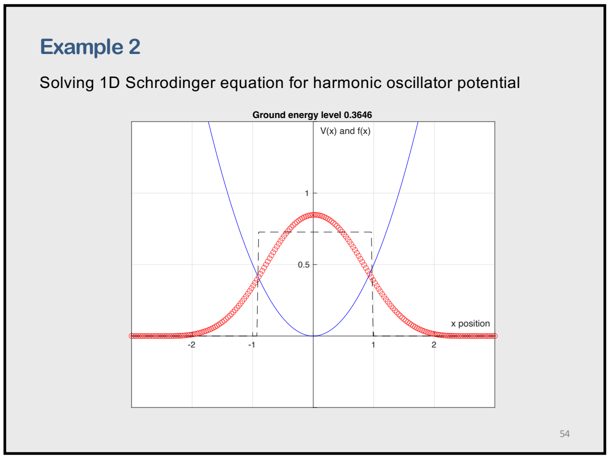
Find a configuration of 19 particles (interacting by Lenard-Jones potential) that has the lowest energy

Initial (random) configuration.

Final configuration

53

53

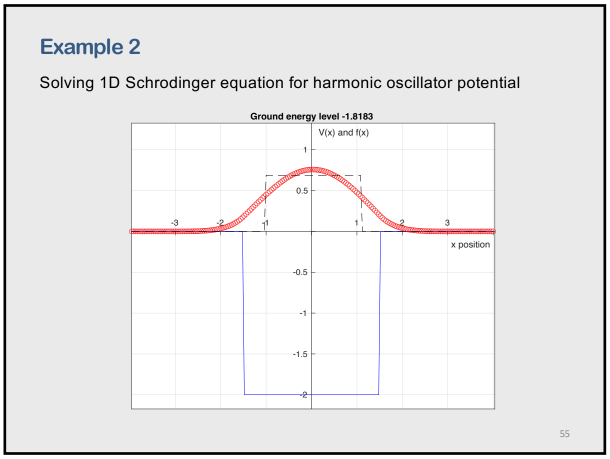


Example 2

Solving 1D Schrodinger equation for harmonic oscillator potential

54

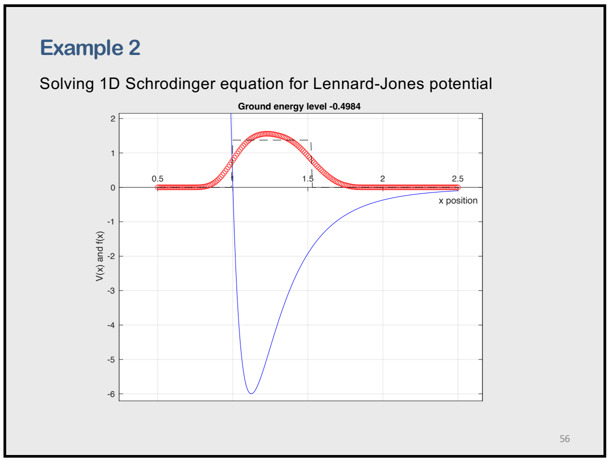
54



Example 2

Solving 1D Schrodinger equation for harmonic oscillator potential

55



Example 2

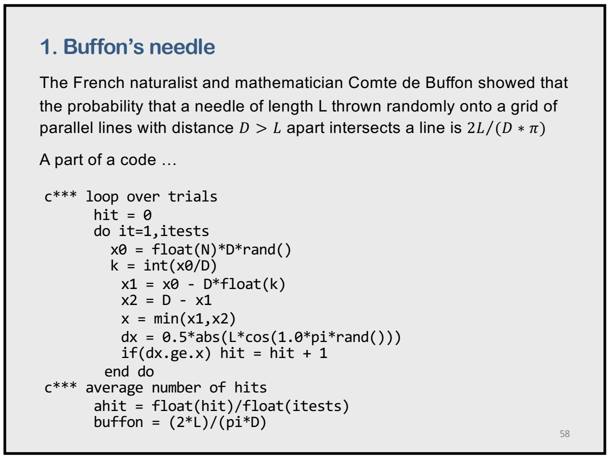
Solving 1D Schrodinger equation for Lennard-Jones potential

56

Part : 4

Problems for a curious student

57



1. Buffon’s needle

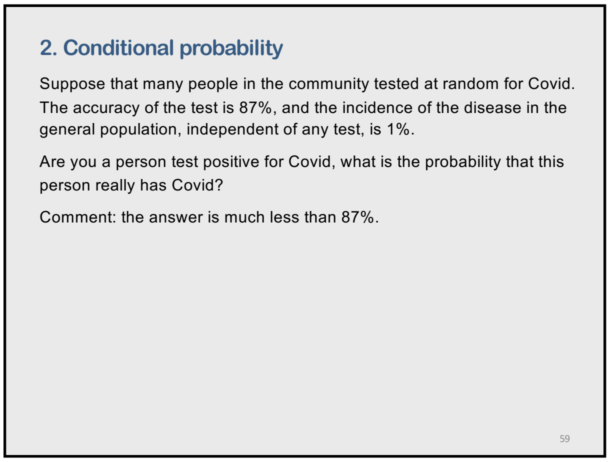
The French naturalist and mathematician Comte de Buffon showed that the probability that a needle of length L thrown randomly onto a grid of parallel lines with distance $D > L$ apart intersects a line is $2L/(D * \pi)$

A part of a code …

```
c*** loop over trials
      hit = 0
      do it=1,itests
        x0 = float(N)*D*rand()
        k = int(x0/D)
         x1 = x0 - D*float(k)
         x2 = D - x1
         x = min(x1,x2)
         dx = 0.5*abs(L*cos(1.0*pi*rand()))
         if(dx.ge.x) hit = hit + 1
       end do
c*** average number of hits
      ahit = float(hit)/float(itests)
      buffon = (2*L)/(pi*D)
```

58

58



2. Conditional probability

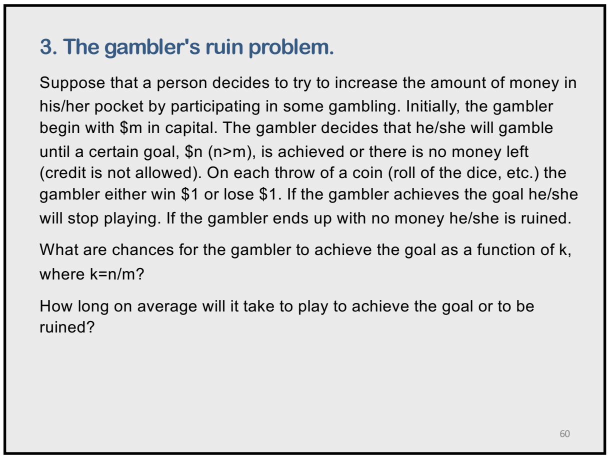
Suppose that many people in the community tested at random for Covid. The accuracy of the test is 87%, and the incidence of the disease in the general population, independent of any test, is 1%.

Are you a person test positive for Covid, what is the probability that this person really has Covid?

Comment: the answer is much less than 87%.

59

59



3. The gambler's ruin problem.

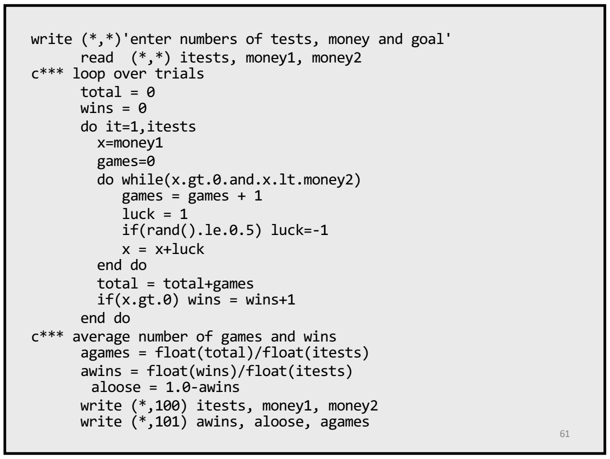
Suppose that a person decides to try to increase the amount of money in his/her pocket by participating in some gambling. Initially, the gambler begin with $m in capital. The gambler decides that he/she will gamble until a certain goal, $n (n>m), is achieved or there is no money left (credit is not allowed). On each throw of a coin (roll of the dice, etc.) the gambler either win $1 or lose $1. If the gambler achieves the goal he/she will stop playing. If the gambler ends up with no money he/she is ruined.

What are chances for the gambler to achieve the goal as a function of k, where k=n/m?

How long on average will it take to play to achieve the goal or to be ruined?

60

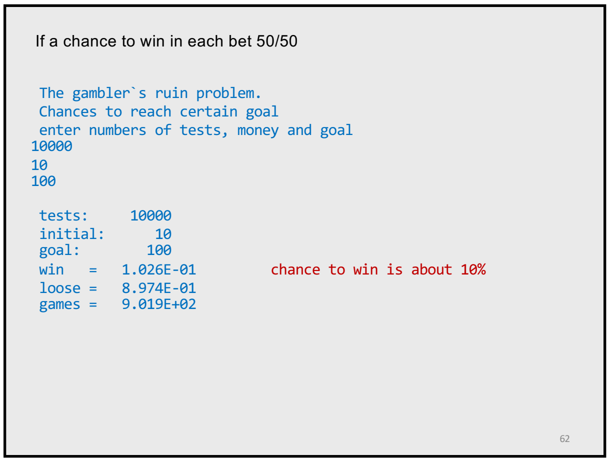
60



```
write (*,*)'enter numbers of tests, money and goal'
      read  (*,*) itests, money1, money2
c*** loop over trials
      total = 0
      wins = 0
      do it=1,itests
        x=money1
        games=0
        do while(x.gt.0.and.x.lt.money2)
           games = games + 1
           luck = 1
           if(rand().le.0.5) luck=-1
           x = x+luck
        end do
        total = total+games
        if(x.gt.0) wins = wins+1
      end do
c*** average number of games and wins
      agames = float(total)/float(itests)
      awins = float(wins)/float(itests)
       aloose = 1.0-awins
      write (*,100) itests, money1, money2
      write (*,101) awins, aloose, agames
```

61

61

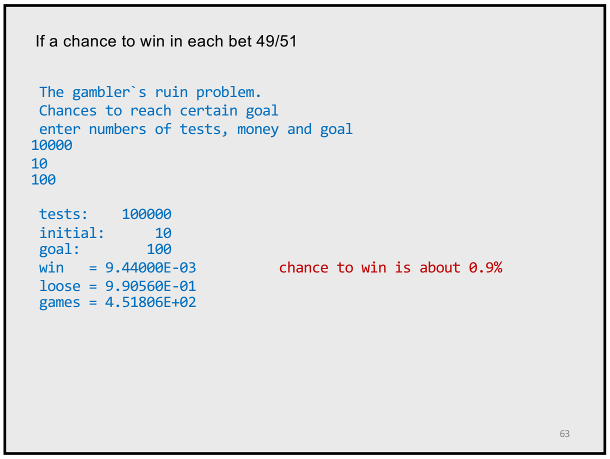


If a chance to win in each bet 50/50

```
 The gambler`s ruin problem.
 Chances to reach certain goal
 enter numbers of tests, money and goal
10000
10
100

 tests:     10000
 initial:      10
 goal:        100
 win   =   1.026E-01        chance to win is about 10%
 loose =   8.974E-01
 games =   9.019E+02
```

62

62

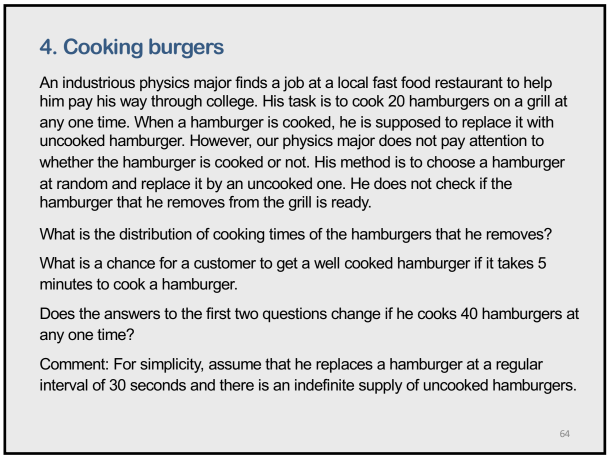


If a chance to win in each bet 49/51

```
 The gambler`s ruin problem.
 Chances to reach certain goal
 enter numbers of tests, money and goal
10000
10
100

 tests:    100000
 initial:      10
 goal:        100
 win   = 9.44000E-03         chance to win is about 0.9%
 loose = 9.90560E-01
 games = 4.51806E+02
```

63

63



4. Cooking burgers

An industrious physics major finds a job at a local fast food restaurant to help him pay his way through college. His task is to cook 20 hamburgers on a grill at any one time. When a hamburger is cooked, he is supposed to replace it with uncooked hamburger. However, our physics major does not pay attention to whether the hamburger is cooked or not. His method is to choose a hamburger at random and replace it by an uncooked one. He does not check if the hamburger that he removes from the grill is ready.

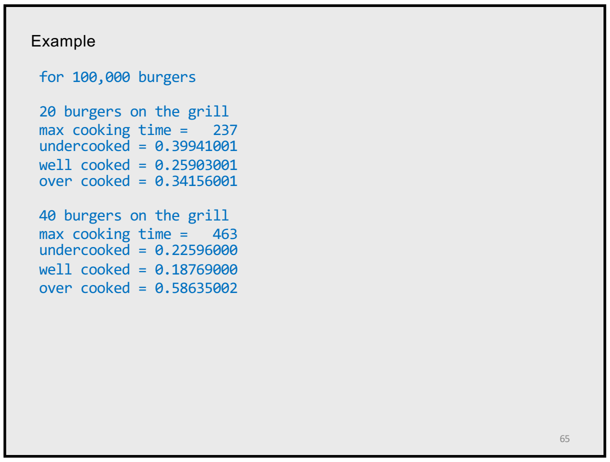
What is the distribution of cooking times of the hamburgers that he removes?

What is a chance for a customer to get a well cooked hamburger if it takes 5 minutes to cook a hamburger.

Does the answers to the first two questions change if he cooks 40 hamburgers at any one time?

Comment: For simplicity, assume that he replaces a hamburger at a regular interval of 30 seconds and there is an indefinite supply of uncooked hamburgers.

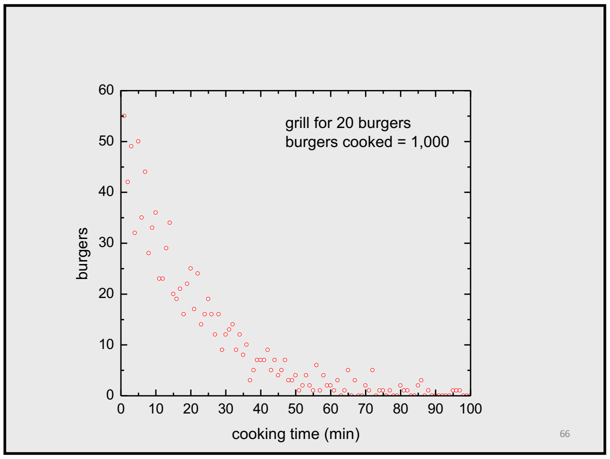
64



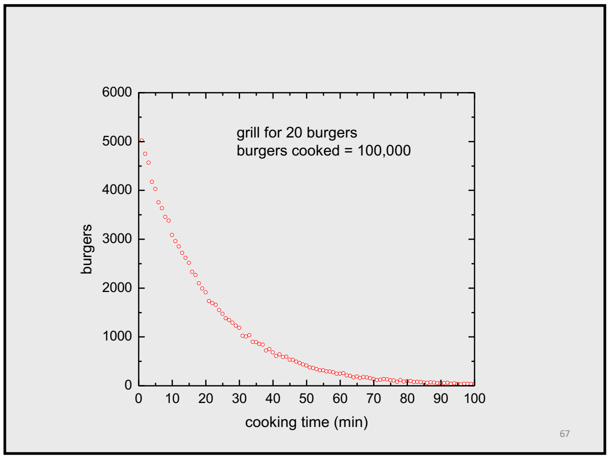
Example

```
for 100,000 burgers

20 burgers on the grill
max cooking time =    237
undercooked = 0.39941001
well cooked = 0.25903001
over cooked = 0.34156001

40 burgers on the grill
max cooking time =    463
undercooked = 0.22596000
well cooked = 0.18769000
over cooked = 0.58635002
```

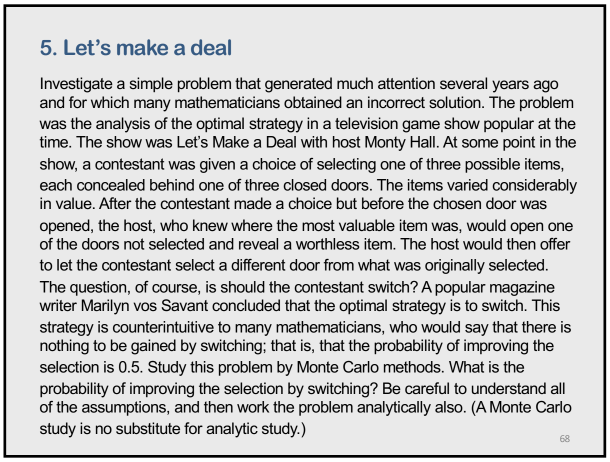
65



66



67

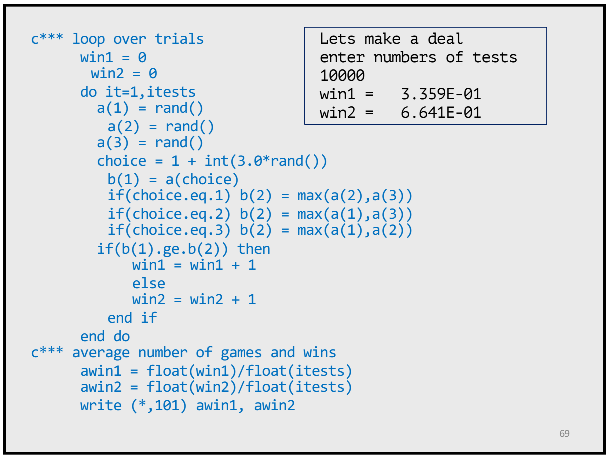


5. Let's make a deal

Investigate a simple problem that generated much attention several years ago and for which many mathematicians obtained an incorrect solution. The problem was the analysis of the optimal strategy in a television game show popular at the time. The show was Let's Make a Deal with host Monty Hall. At some point in the show, a contestant was given a choice of selecting one of three possible items, each concealed behind one of three closed doors. The items varied considerably in value. After the contestant made a choice but before the chosen door was opened, the host, who knew where the most valuable item was, would open one of the doors not selected and reveal a worthless item. The host would then offer to let the contestant select a different door from what was originally selected. The question, of course, is should the contestant switch? A popular magazine writer Marilyn vos Savant concluded that the optimal strategy is to switch. This strategy is counterintuitive to many mathematicians, who would say that there is nothing to be gained by switching; that is, that the probability of improving the selection is 0.5. Study this problem by Monte Carlo methods. What is the probability of improving the selection by switching? Be careful to understand all of the assumptions, and then work the problem analytically also. (A Monte Carlo study is no substitute for analytic study.)

68

68



```
c*** loop over trials
     win1 = 0
      win2 = 0
     do it=1,itests
       a(1) = rand()
        a(2) = rand()
       a(3) = rand()
       choice = 1 + int(3.0*rand())
        b(1) = a(choice)
        if(choice.eq.1) b(2) = max(a(2),a(3))
        if(choice.eq.2) b(2) = max(a(1),a(3))
        if(choice.eq.3) b(2) = max(a(1),a(2))
       if(b(1).ge.b(2)) then
           win1 = win1 + 1
           else
           win2 = win2 + 1
        end if
     end do
c*** average number of games and wins
     awin1 = float(win1)/float(itests)
     awin2 = float(win2)/float(itests)
     write (*,101) awin1, awin2
```

```
Lets make a deal
enter numbers of tests
10000
win1 =   3.359E-01
win2 =   6.641E-01
```

69

69