Monte Carlo

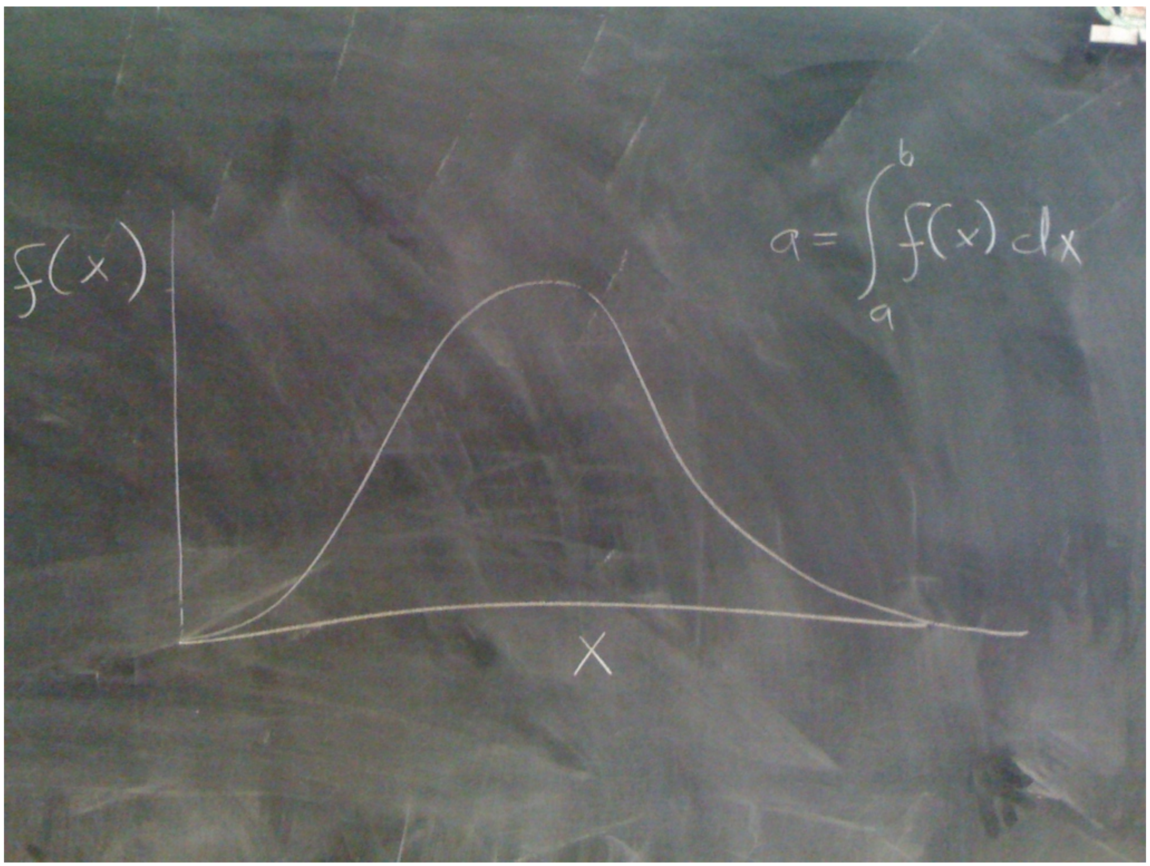
A. Monte Carlo (History and Overview)

B. A Monte Carlo method to calculate π

C. General Monte Carlo integration

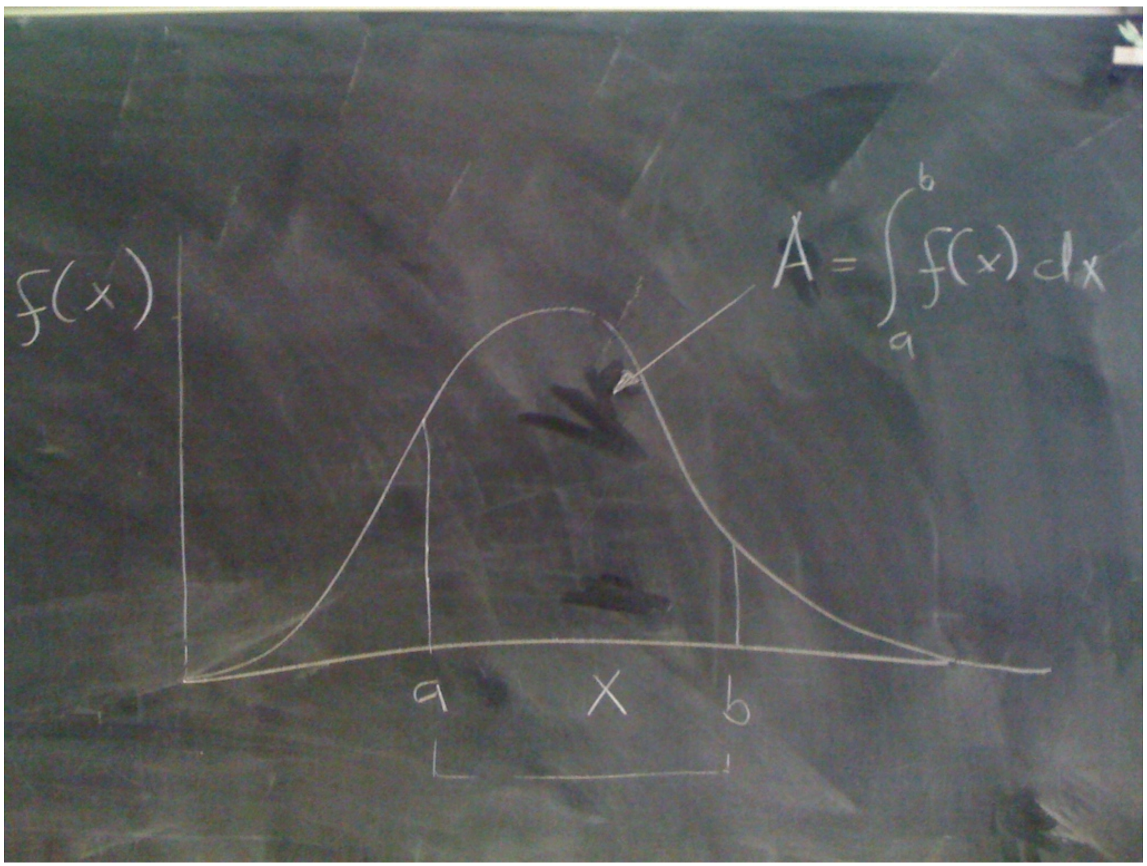
Numerical integration

$$\int_a^b f(x)dx$$



Numerical integration

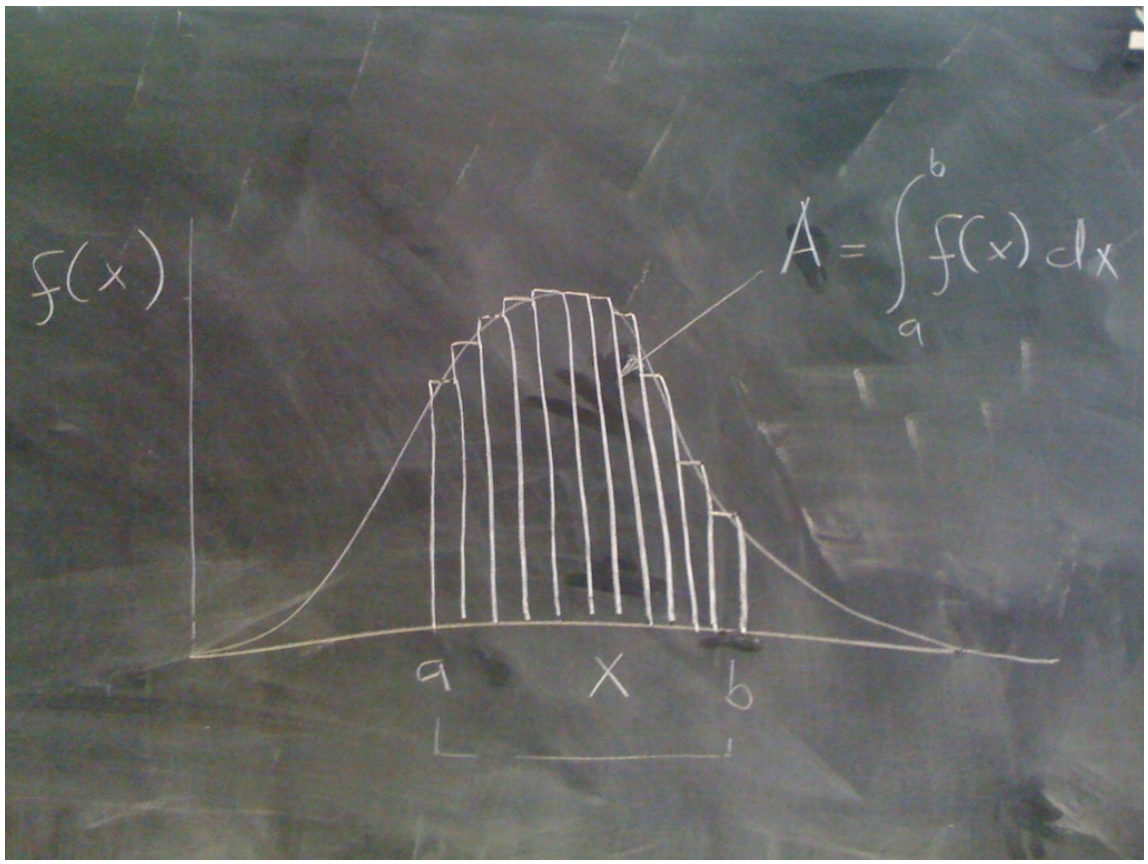
$$\int_a^b f(x)dx$$



Numerical integration

$$\int_a^b f(x)dx$$

We can approximate the area with thin bars that have all the same width, and so fill the area between a and b



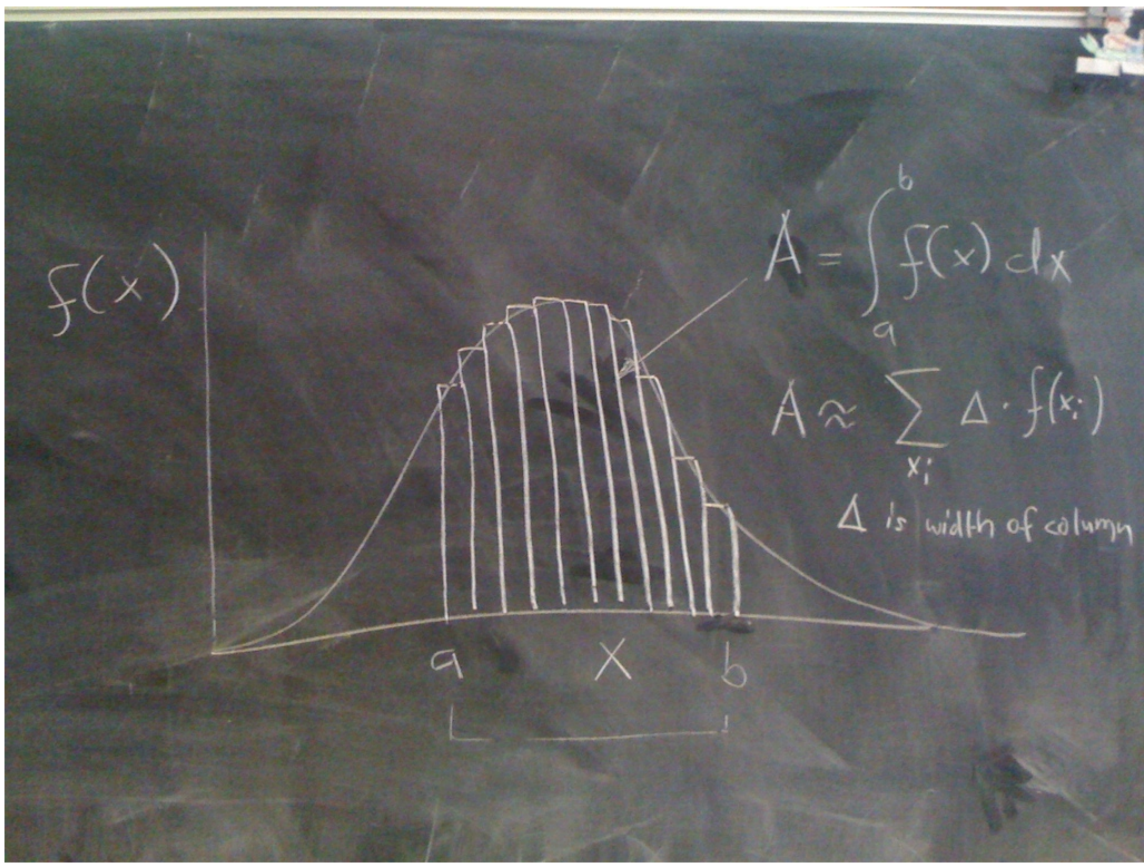
Numerical integration

$$\int_a^b f(x)dx$$

We can approximate the area with thin bars that have all the same width, and so fill the area between a and b

$$\sum_{a+\frac{\Delta}{2}}^{b-\frac{\Delta}{2}} \Delta f(x)$$

$$x_i = x_{i-1} + \Delta$$



JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION

Number 247 SEPTEMBER 1949 *Volume 44*

THE MONTE CARLO METHOD

Nicholas Metropolis and S. Ulam
Los Alamos Laboratory

We shall present here the motivation and a general description of a method dealing with a class of problems in mathematical physics. The method is, essentially, a statistical approach to the study of differential equations, or more generally, of integro-differential equations that occur in various branches of the natural sciences.

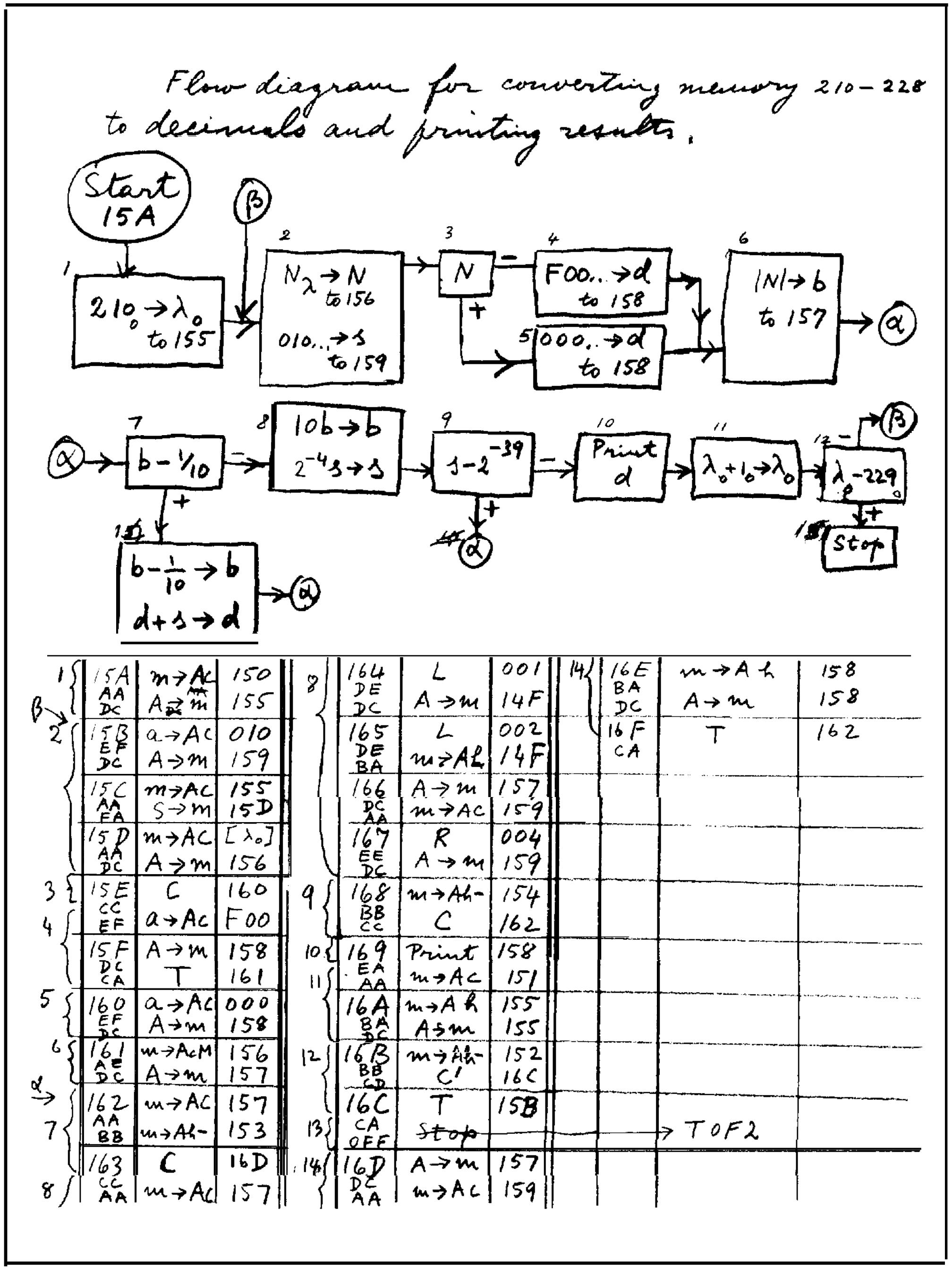


Fig. 3. A subprogram written by Fermi for converting data in memory from hexadecimal to decimal form and printing the results.

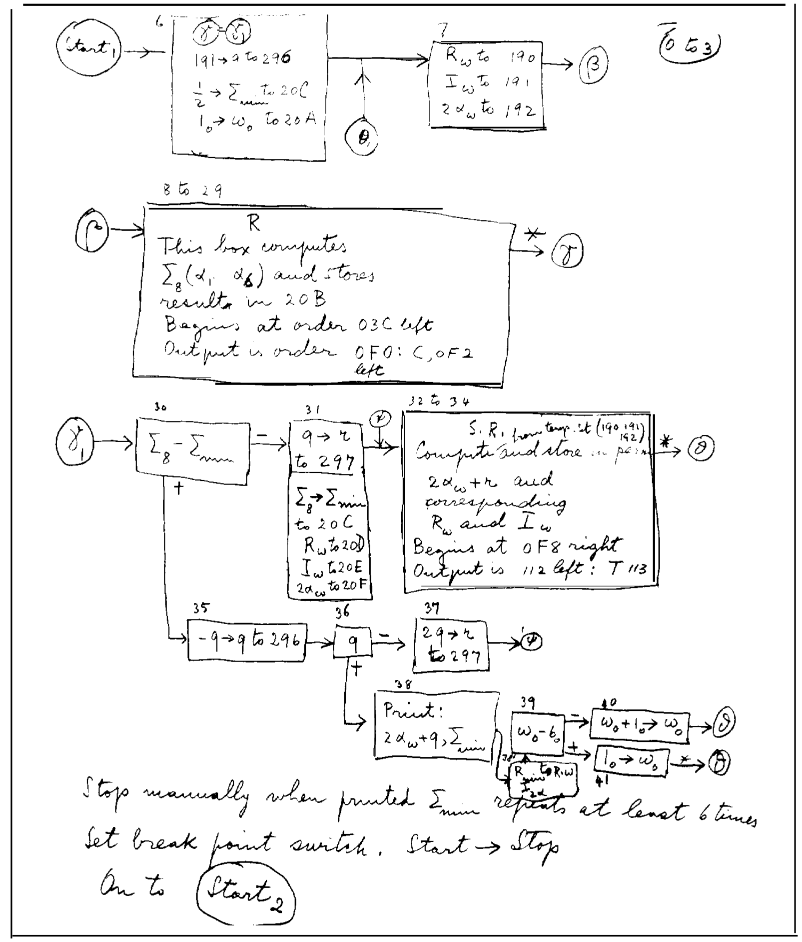


Fig. 4. A subprogram written by Fermi for calculating phase shifts by finding a minimum chi-squared in a fit to the data.

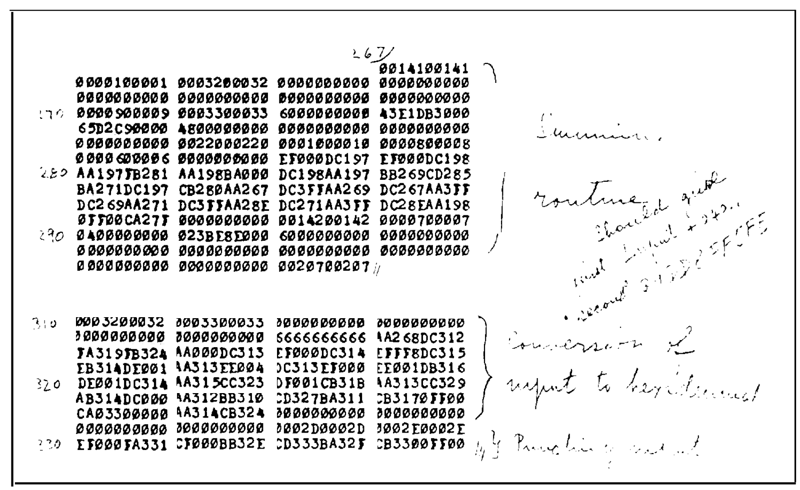
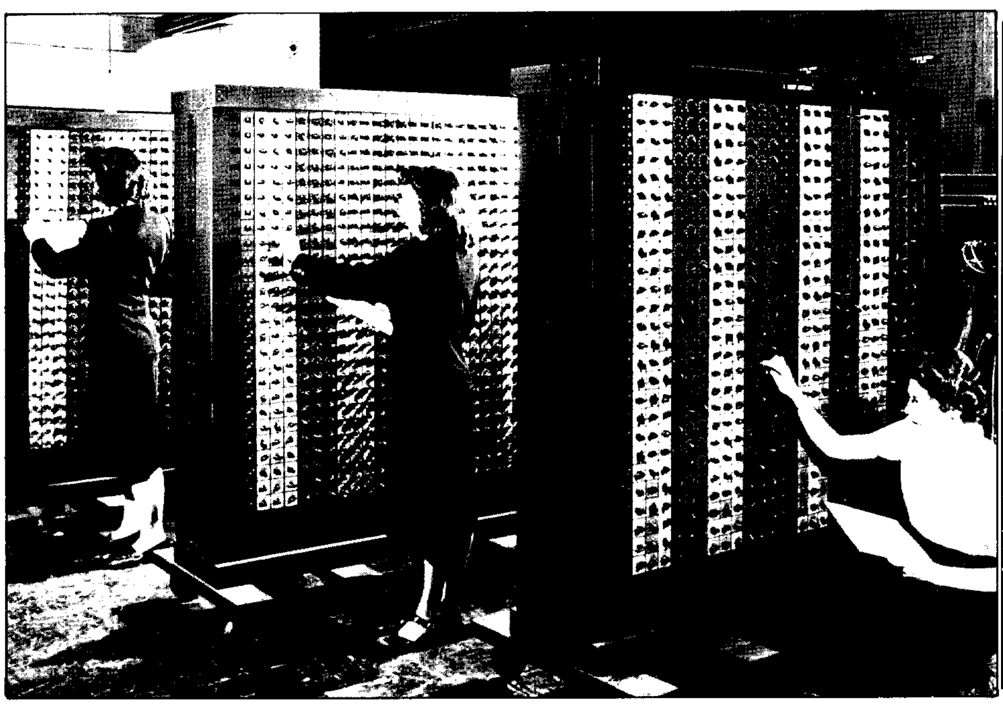
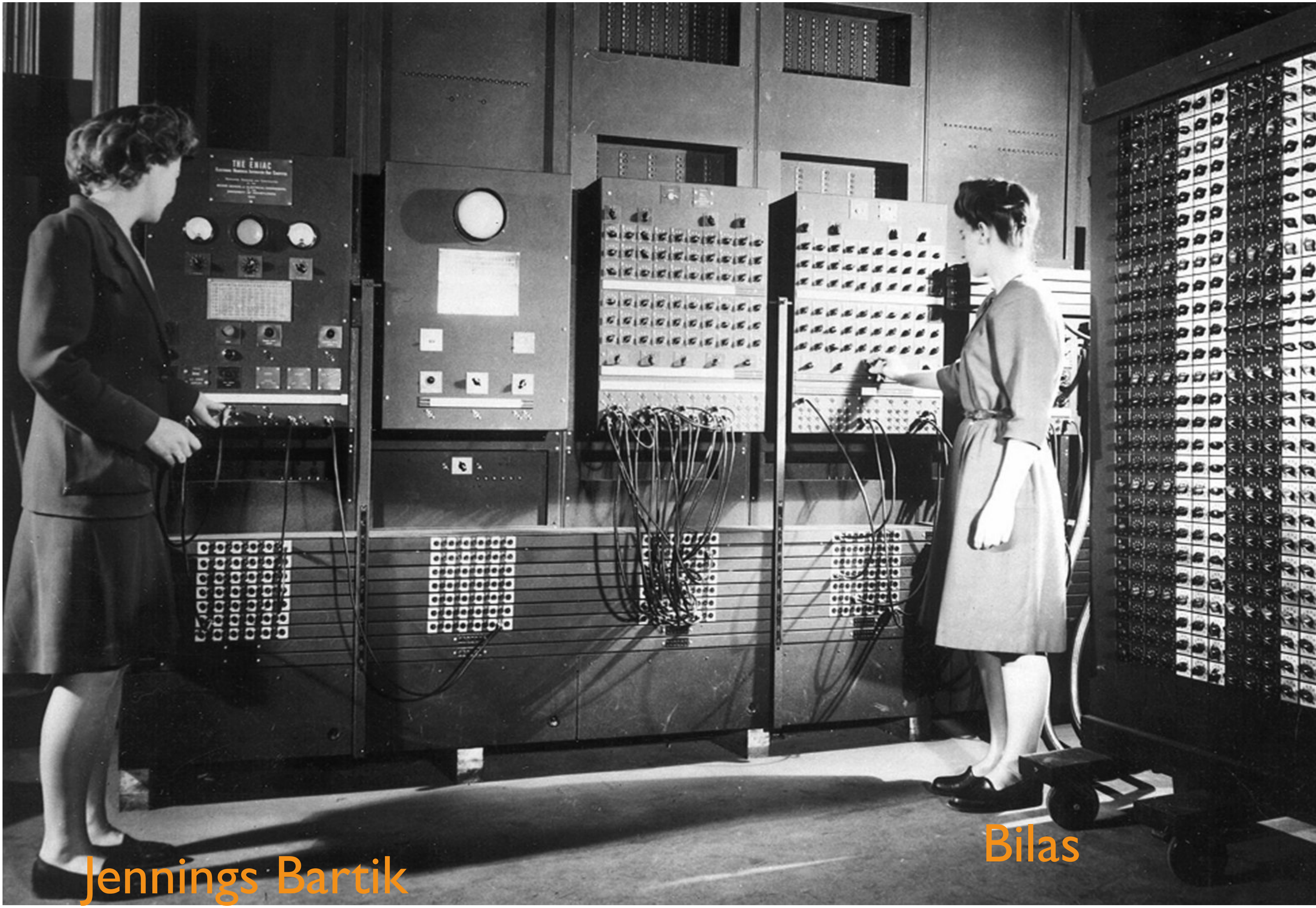


Fig. 5. A portion of the printout of the program containing the subprograms described in Figs. 3 and 4. The program is written in machine language in hexadecimal numbers.

Monte Carlo method is tied to first electronic computers and the development of the atomic bomb.

Stan Ulam, Enrico Fermi, John von Neumann, Nicolas Metropolis, Edward Teller, Marshall Rosenbluth
Augusta Harkanyi Teller, and Arianna Rosenbluth




THE ENIAC
Bilas
Jennings Bartik

Recipe

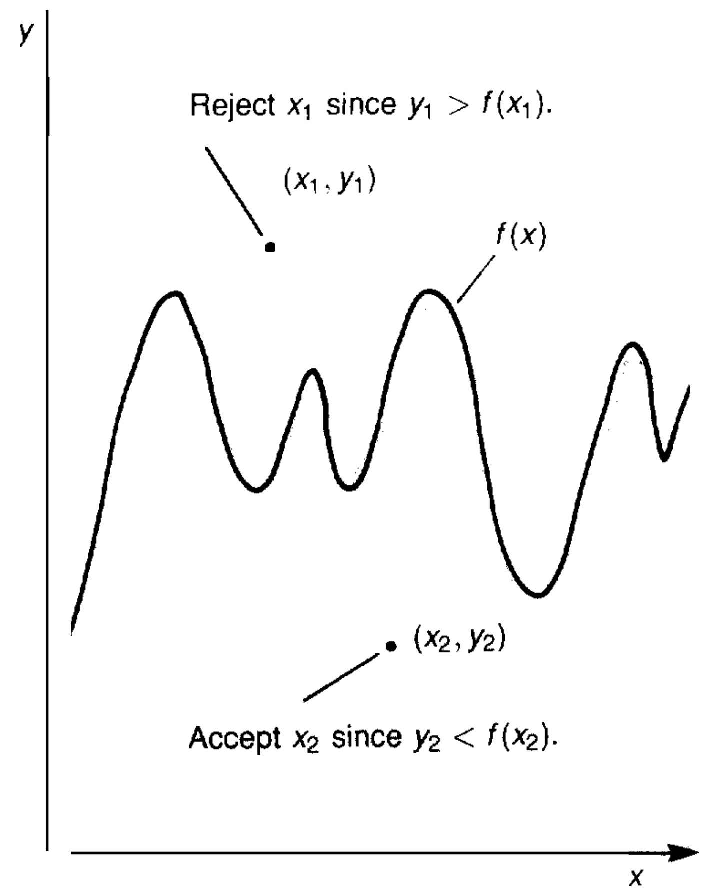
Do this many times:

- Draw a random number x using the distribution f(x) between a and b
- Save x

Create a histogram of all the x values

THE ACCEPTANCE-REJECTION METHOD

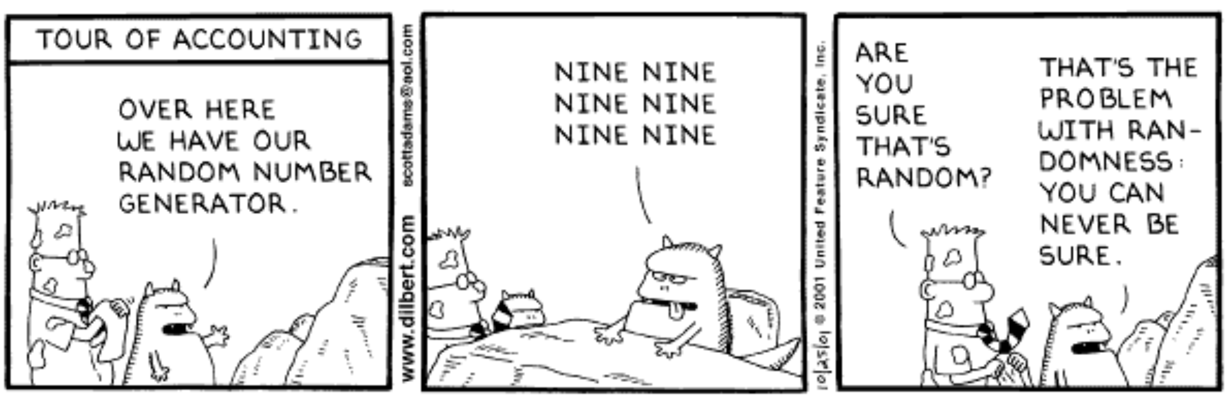
Fig. 4. If two independent sets of random numbers are used, one of which (x^i) extends uniformly over the range of the distribution function f and the other (y^i) extends over the domain of f, then an acceptance-rejection technique based on whether or not $y^i \leq f(x^i)$ will generate a distribution for (x^i) whose density is $f(x^i)\,dx^i$.



Random numbers

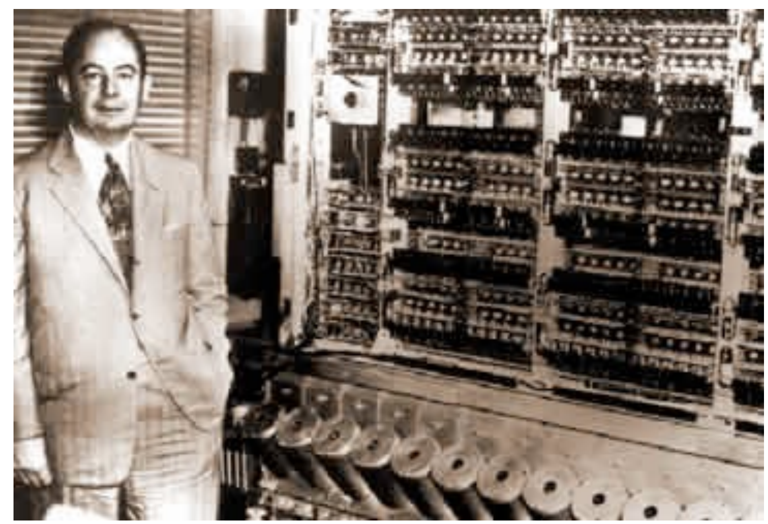
Pseudorandom numbers

John von Neumann proposed using the following method as one of the first random number generators. Suppose we want to create eight-digit numbers. Begin with an eight-digit number X_0, which we call the *seed*, and create the next integer in the sequence by removing the middle eight digits from X_0^2.



Physical random number generators

Lava-lamps
Geiger counter
Devices that combine
multiple events /dev/random



Pseudocode for von Neuman random number generator

```
initialize vector x
seed = input(a number with 8 digits)
for i in 1,n:
    x0 = seed * seed
    seed = take middle 8 digits of x0
    x[i] = seed
```

Example

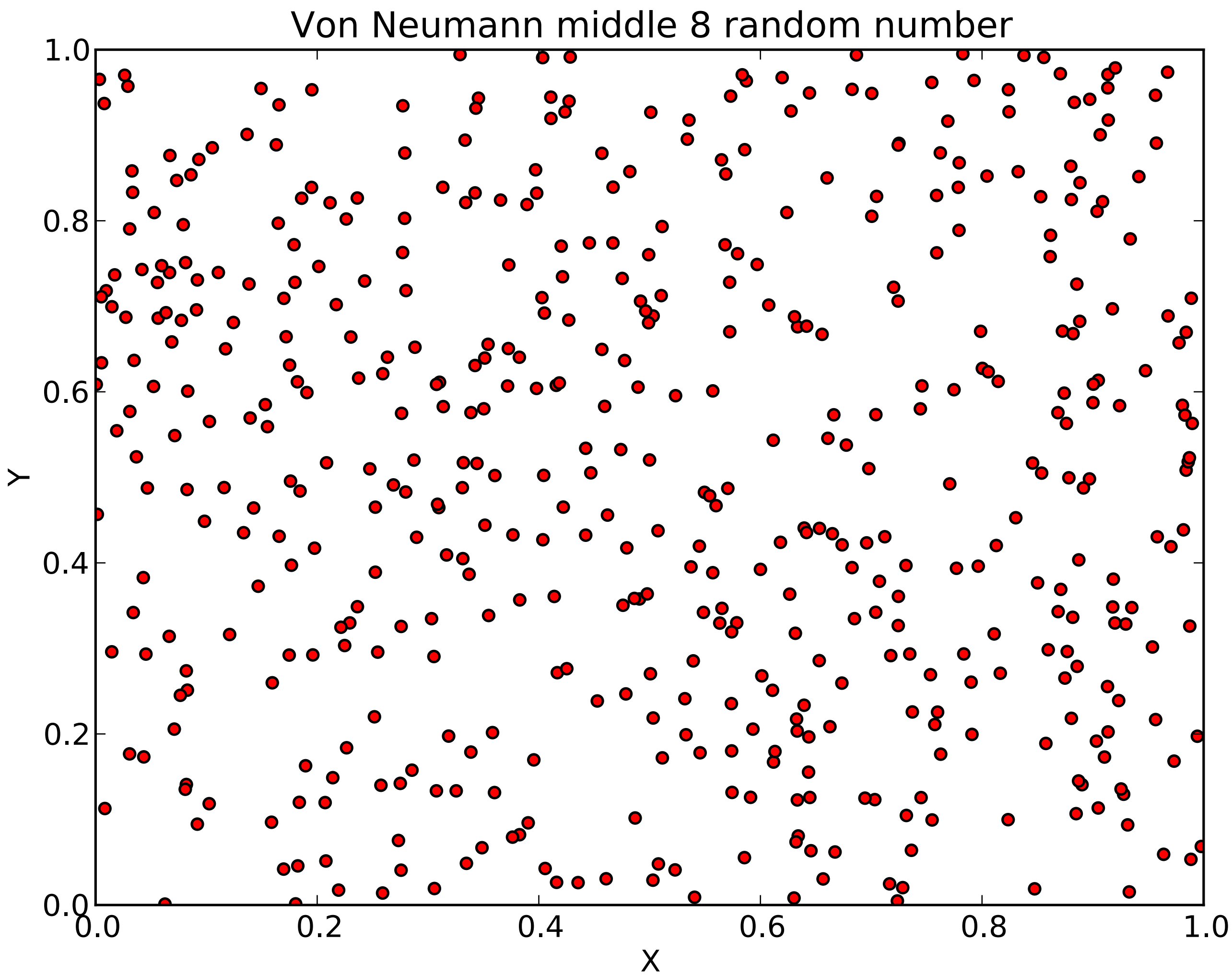
$$81989672$$

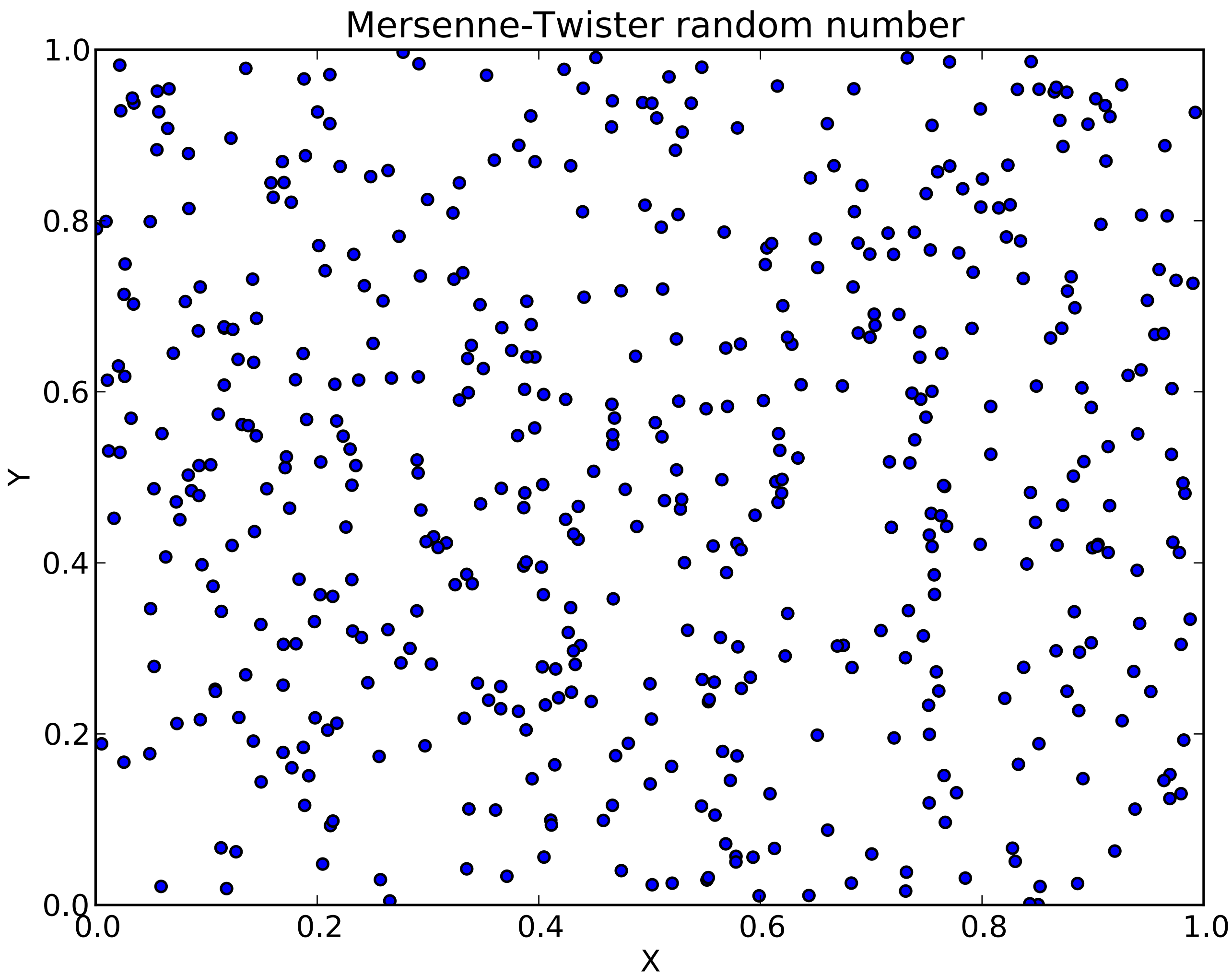
$$81989672 \times 81989672$$

$$6722\textcolor{red}{30631466}7584$$

$$30631466$$

$$30631466 \times 30631466$$


Von Neumann middle 8 random number
Y
X
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


Mersenne-Twister random number
Y
X
0.0
0.2
0.4
0.6
0.8
1.0

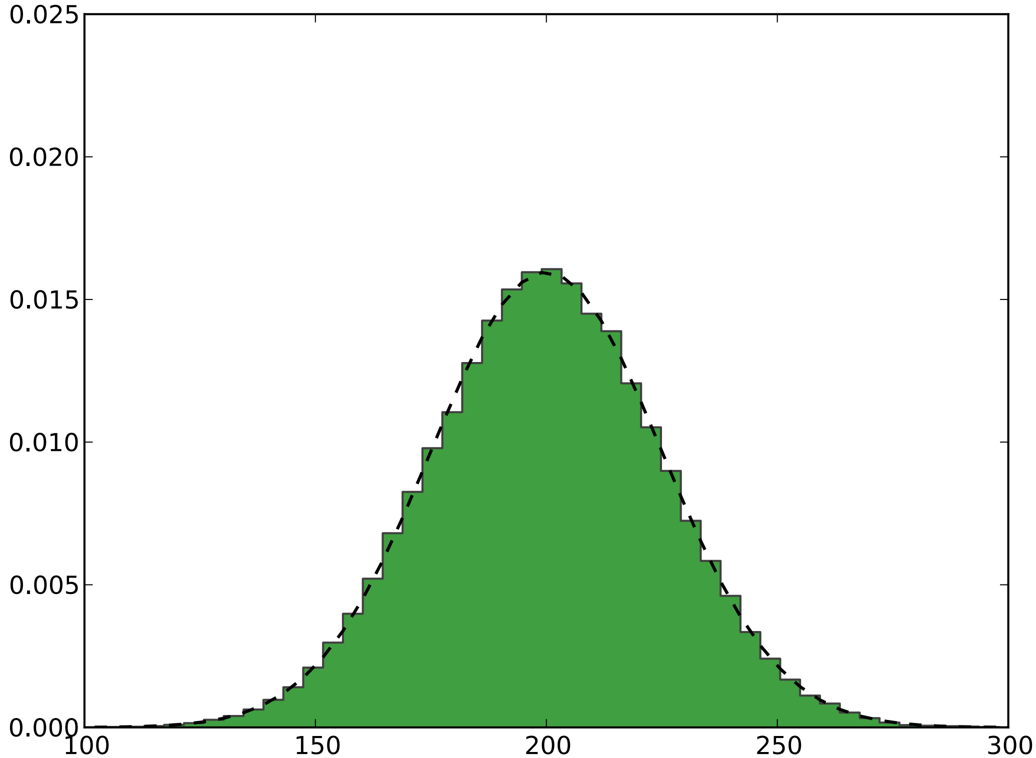
Monte Carlo

Recipe

Do this many times:

- Draw a random number x using the distribution f(x) between a and b
- Save x

Create a histogram of all the x values



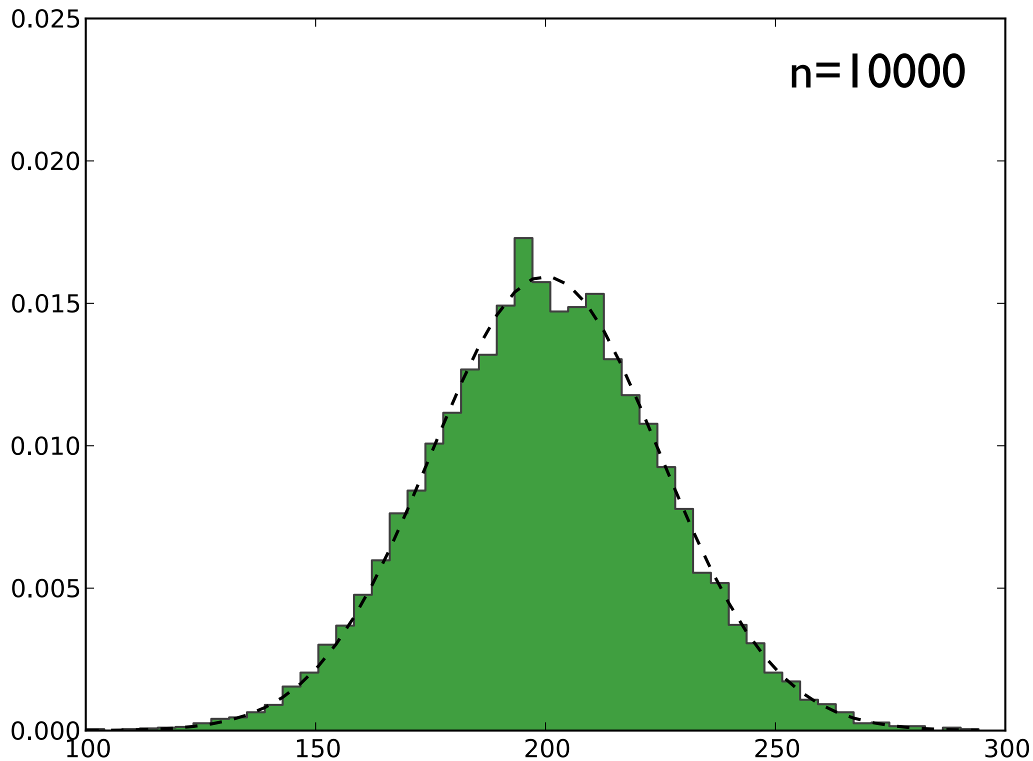
Monte Carlo

Recipe

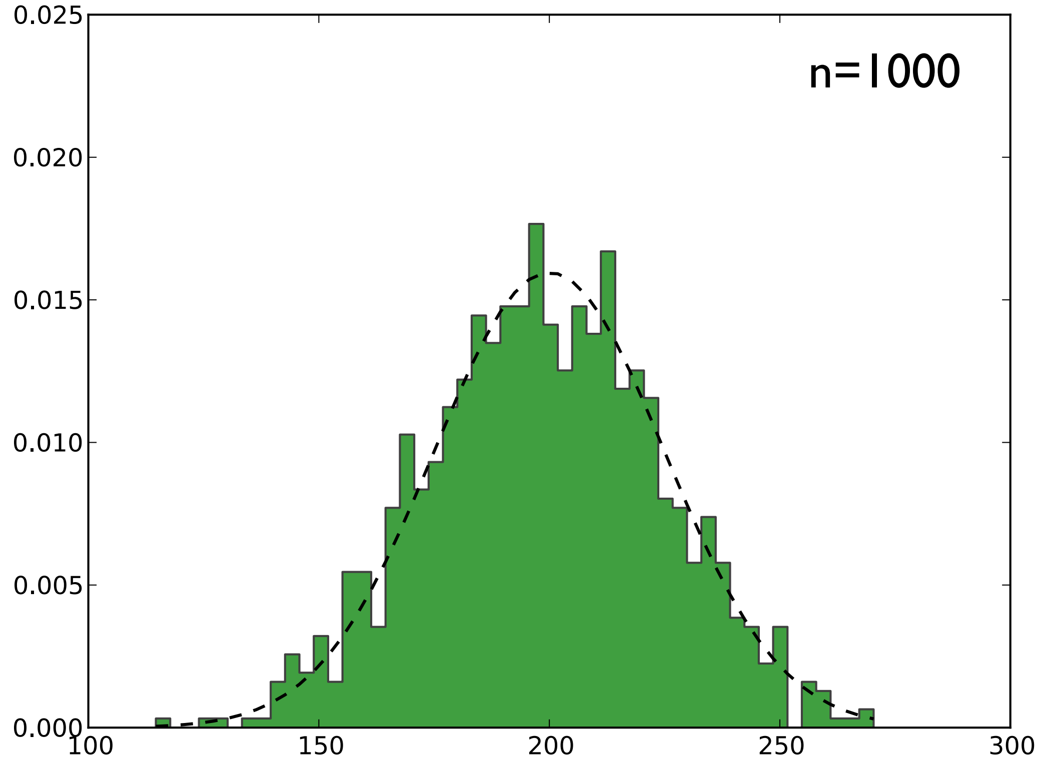
Do this many times:

- Draw a random number x using the distribution f(x) between a and b
- Save x

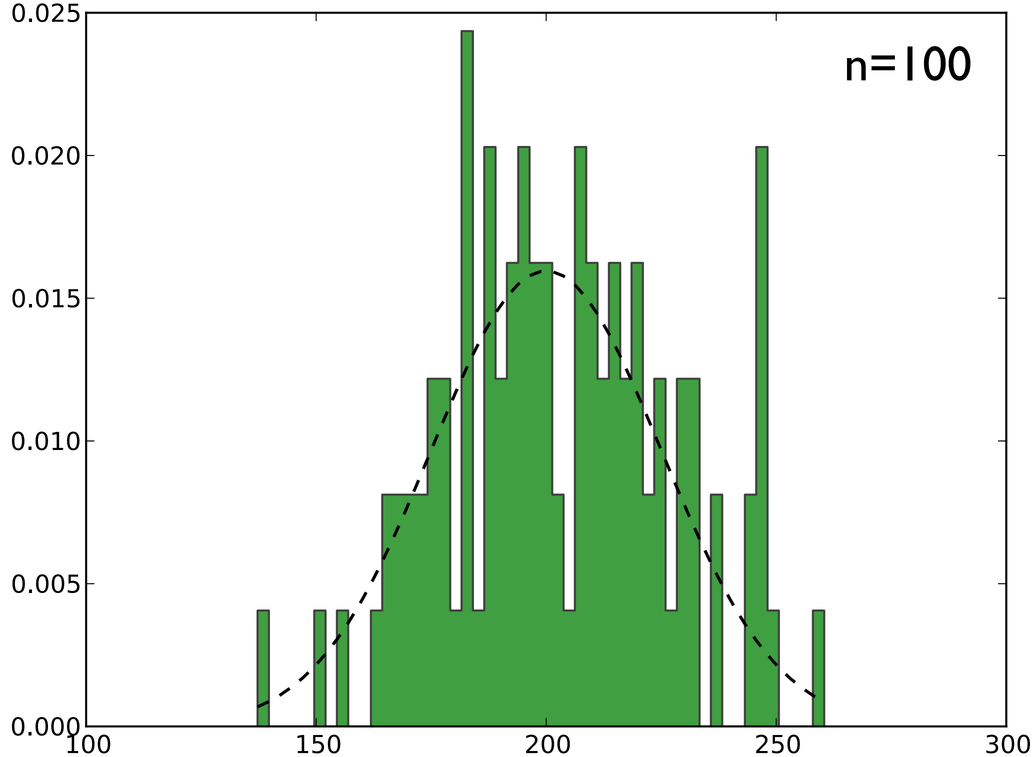
Create a histogram of all the x values



Monte Carlo



Monte Carlo



Monte Carlo Rules

- *Probability distribution functions (pdf's)* — the physical (or mathematical) system must be described by a set of pdf's.
- *Random number generator* — a source of random numbers uniformly distributed on the unit interval must be available.
- *Sampling rule* — a prescription for sampling from the specified pdf's, assuming the availability of random numbers on the unit interval, must be given.
- *Scoring (or tallying)* — the outcomes must be accumulated into overall tallies or scores for the quantities of interest.

Monte Carlo Examples

Nuclear reactor design
Quantum chromodynamics
Radiation cancer therapy
Traffic flow
Stellar evolution
Econometrics
Dow-Jones forecasting
Oil well exploration
VLSI design

Phylogeny inference
Population genetics inference

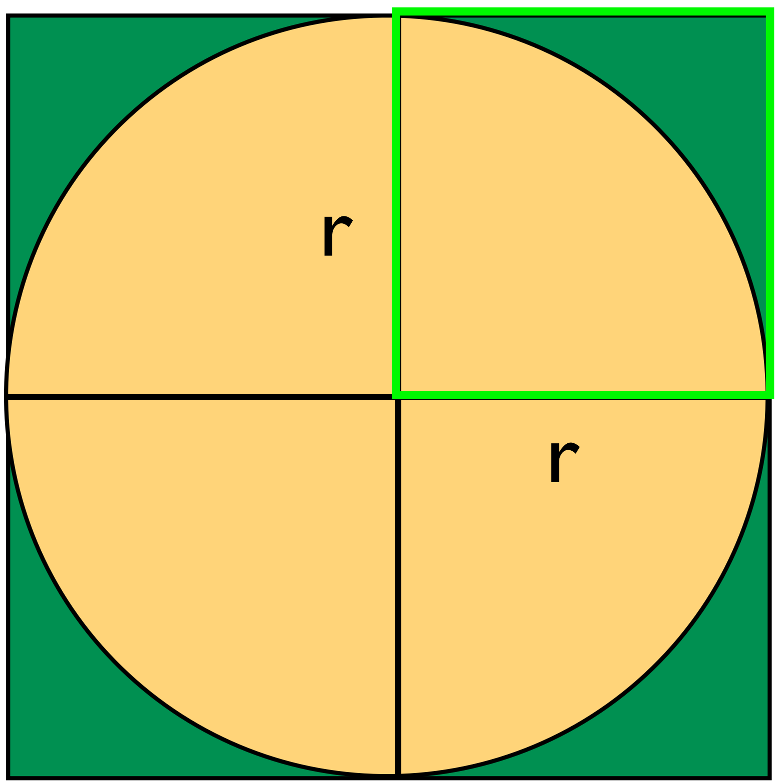
How to calculate π

?

How to calculate π

We know that the area of a circle is

$$\pi r^2$$



Looking only at the upper right corner
we can see a green square with side r
and we can calculate the area of the square as

$$A_s = r^2$$

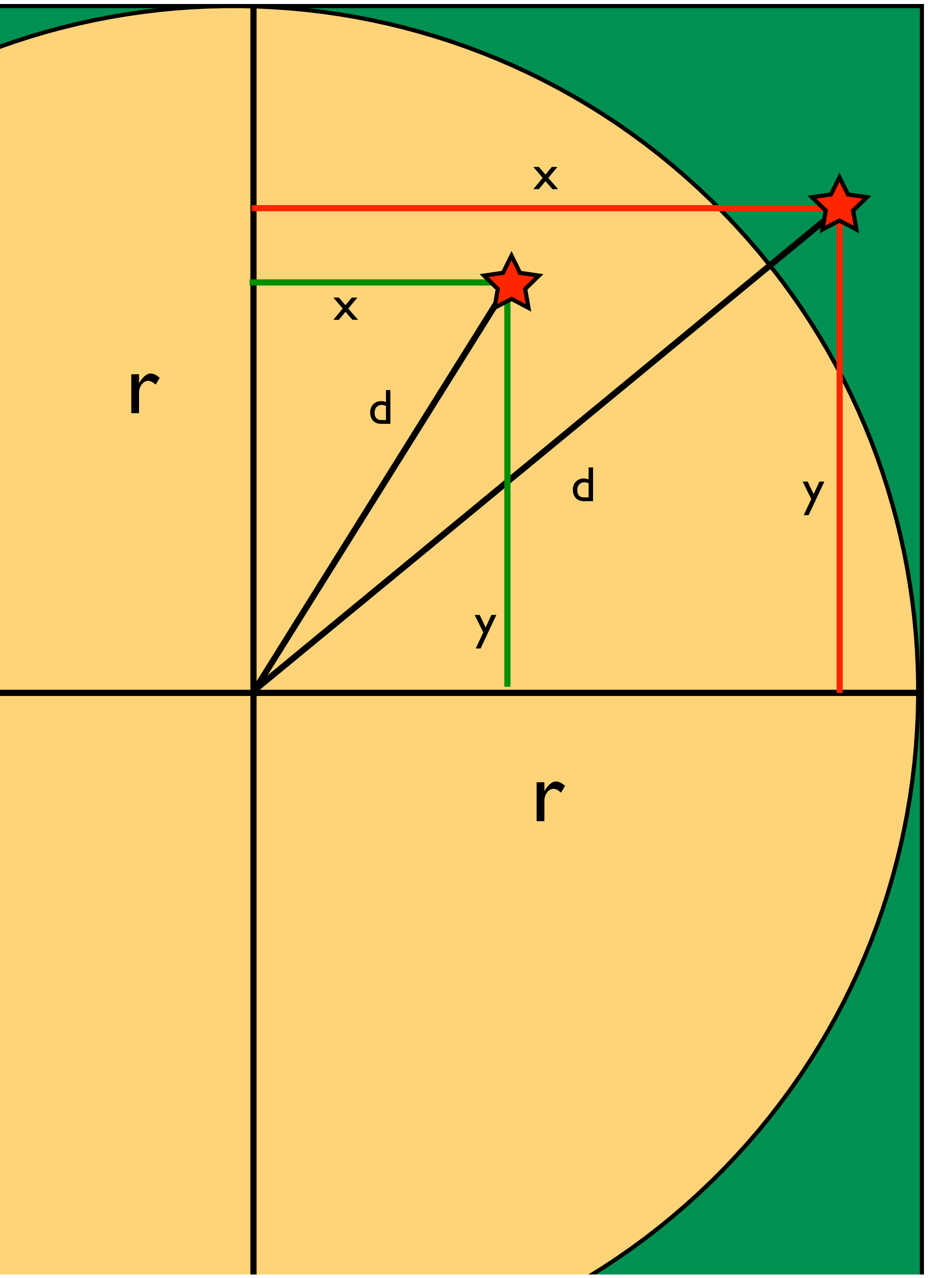
The quarter circle has the area

$$A_c = \frac{\pi}{4} r^2$$

So we can calculate the ratio of the two areas as

$$\frac{A_c}{A_s} = \frac{r^2}{\frac{\pi}{4} r^2} = \frac{\pi}{4}$$

How to calculate π



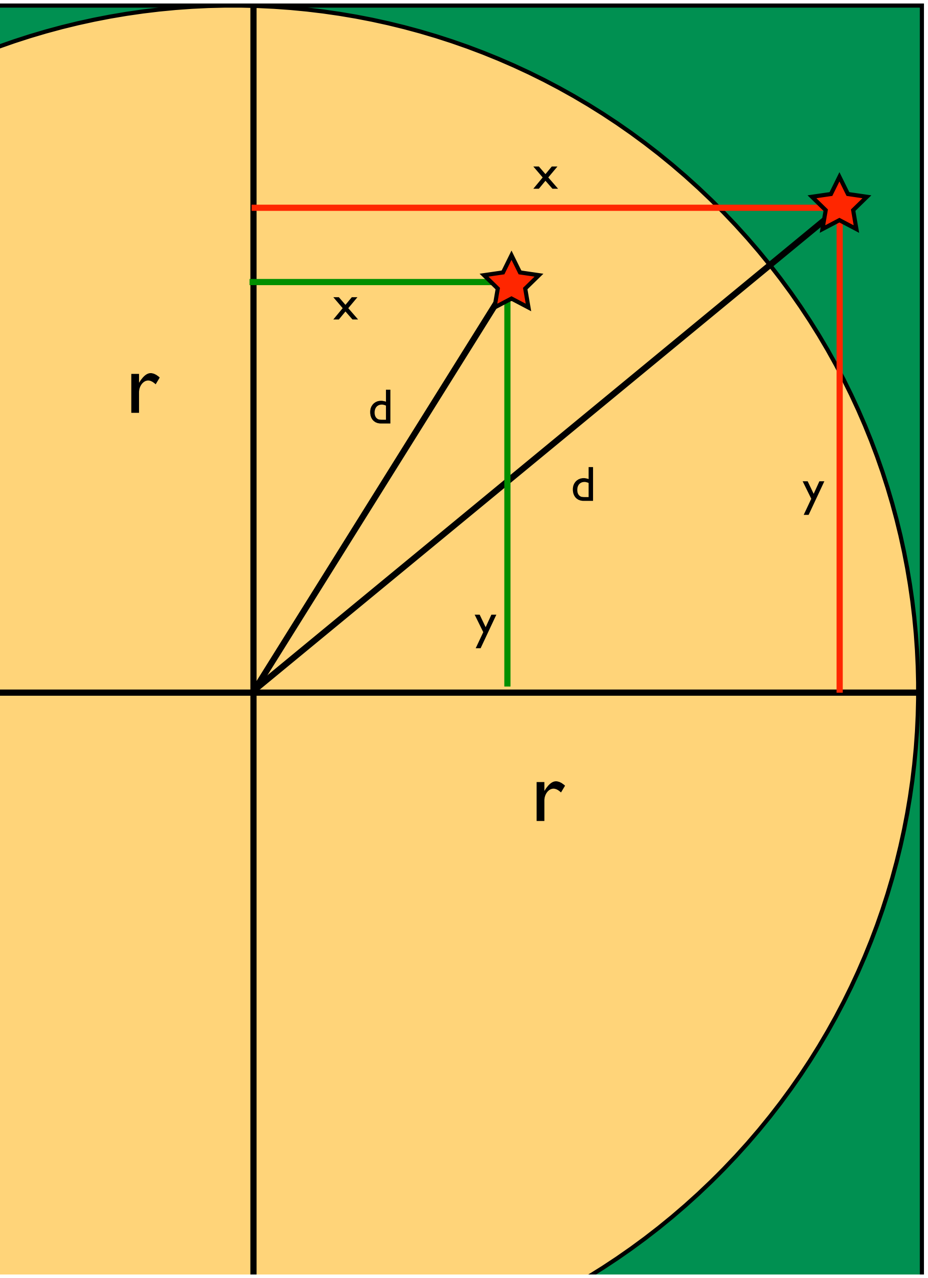
$$\frac{A_c}{A_s} = \frac{r^2}{\frac{\pi}{4}r^2} = \frac{\pi}{4}$$

The goal is now to estimate the ratio of the areas. We can devise an algorithm that draws random coordinates from the square and marks whether the coordinate fell into the circle or not. We can calculate the distance from the circle center using Pythagoras:

$$d = \sqrt{(x^2 + y^2)}$$

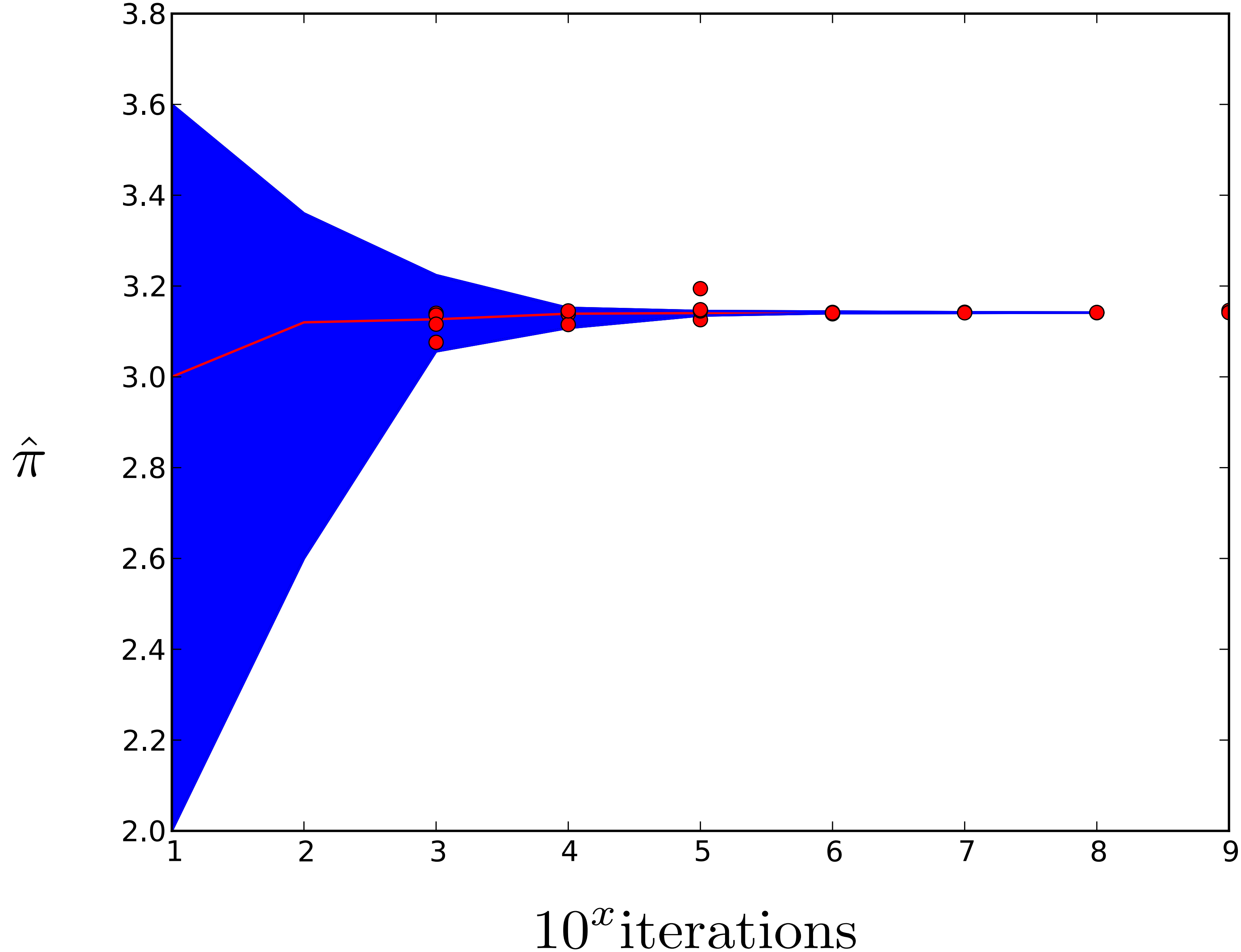
If d is smaller than r than we know the coordinate is in the circle otherwise only in the square. We can now create an algorithm for our program.

How to calculate π



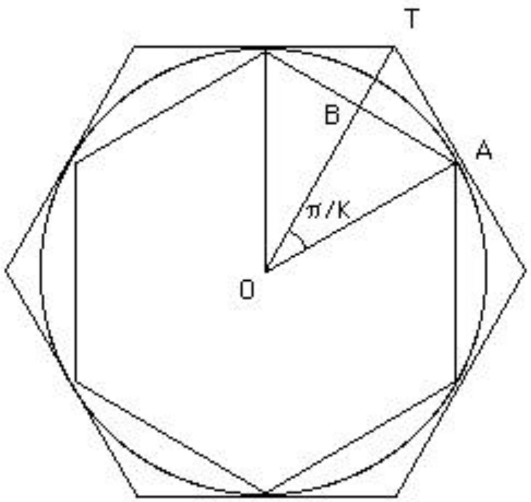

```
// Algorithm in pseudo code
// Do many times:
//    draw x, y coordinate
//    calculate d from center
//    check whether d < r:
//       True: add 1 to circle
//       False: do nothing
//    add 1 to square
//
// print pi: ratio circle/square * 4
```

Our Pi estimates



History of π

Archimedes (300 BC)
using 96-side polygons



$$223/71 < \pi < 22/7$$

3.140845070422535 3.142857142857142 8

Ptolemy	(c. 150 AD)	3.1416
Zu Chongzhi	(430-501 AD)	$^{355}/_{113}$
al-Khwarizmi	(c. 800)	3.1416
al-Kashi	(c. 1430)	14 places
Viète	(1540-1603)	9 places
Roomen	(1561-1615)	17 places
Van Ceulen	(c. 1600)	35 places

James Gregory
1638-1675

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots$$

History of π

1699: Sharp used Gregory's result to get 71 correct digits

1701: Machin used an improvement to get 100 digits and the following used his methods:

1719: de Lagny found 112 correct digits

1789: Vega got 126 places and in 1794 got 136

1841: Rutherford calculated 152 digits and in 1853 got 440

1873: Shanks calculated 707 places of which 527 were correct

Very soon after Shanks' calculation a curious statistical freak was noticed by De Morgan, who found that in the last of 707 digits there was a suspicious shortage of 7's. He mentions this in his Budget of Paradoxes of 1872 and a curiosity it remained until 1945 when Ferguson discovered that Shanks had made an error in the
528th place, after which all his digits were wrong. In 1949 a computer was used to calculate π to 2000 places. In this and all subsequent computer expansions the number of 7's does not differ significantly from its expectation, and indeed the sequence of digits has so far passed all statistical tests for randomness.

Buffon's needle experiment. If we have a uniform grid of parallel lines, unit distance apart and if we drop a needle of length $k < 1$ on the grid, the probability that the needle falls across a line is $2k/\pi$. Various people have tried to calculate π by throwing needles. The most remarkable result was that of Lazzerini (1901), who made 34080 tosses and got
$\pi = 355/113 = 3.1415929$
which, incidentally, is the value found by Zu Chongzhi. This outcome is suspiciously good, and the game is given away by the strange number 34080 of tosses. Kendall and Moran comment that a good value can be obtained by stopping the experiment at an optimal moment. If you set in advance how many throws there are to be then this is a very inaccurate way of computing π. Kendall and Moran comment that you would do better to cut out a large circle of wood and use a tape measure to find its circumference and diameter.

π

In the State of Indiana in 1897 the House of Representatives unanimously passed a Bill introducing a new mathematical truth:

Be it enacted by the General Assembly of the State of Indiana: It has been found that a circular area is to the square on a line equal to the quadrant of the circumference, as the area of an equilateral rectangle is to the square of one side. (Section I, House Bill No. 246, 1897)

The Senate of Indiana showed a little more sense and postponed indefinitely the adoption of the Act!

3.14159265358979323846264338327950288419716939937510582097494459230781640628620899
86280348253421170679821480865132823066470938446095505822317253594081284811174502841
02701938521105559644622948954930381964428810975665933446128475648233786783165271201
90914564856692346034861045432664821339360726024914127372458700660631558817488152
09209628292540917153643678925903600113305305488204665213841469519415116094330572703
65759591953092186117381932611793105118548074462379962749567351885752724891227938183
01194912983367336244065664308602139494639522473719070217986094370277053921717629
31767523846748184676694051320005681271452635608277857713427577896091736371787214684
40901224953430146549585371050792279689258923542019956112129021960864034418159813629
77477130996051870721134999999837297804995105973173281609631859502445945534690830
2642522308253344685035261931188171010003137838752886587533208381420617177669147303
5982534904287554687311595628638823537875937519577818577805321712268066130019278766
11195909216420198938095257201065485863278865936153381827968230301952035301852968995
77362259941389124972177528347913151557485724245415069595082953311686172785588907509
83817546374649393192550604009277016711390098488240128583616035637076601047101819
429555961989467678374494482553797747268471040475346462080466842590694912...

Using Monte Carlo to approximate an integral

- Suppose we want to evaluate $\int_a^b f(x)\,dx$
- If $f(x) \geq 0$ for $a \leq x \leq b$ then we know that this integral represents the area under the curve $y = f(x)$ and above the $x-$axis.
- Standard deterministic numerical integration rules approximate this integral by evaluating the integrand $f(x)$ at a set number of points and multiplying by appropriate weights.
  - For example, the midpoint rule is

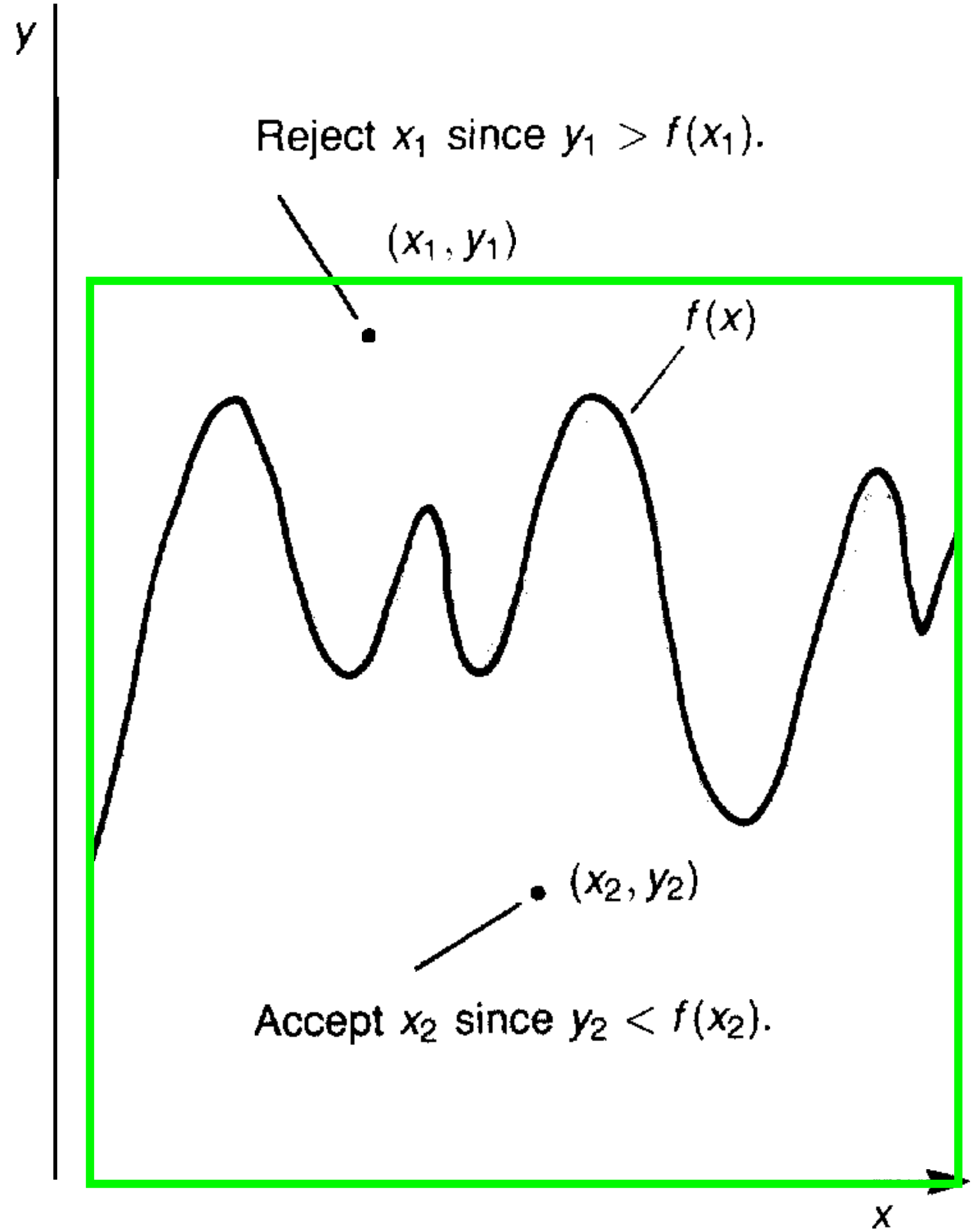
$$\int_a^b f(x)\,dx \approx f\left(\frac{a+b}{2}\right)(b-a)$$

  - Simpson's rule is

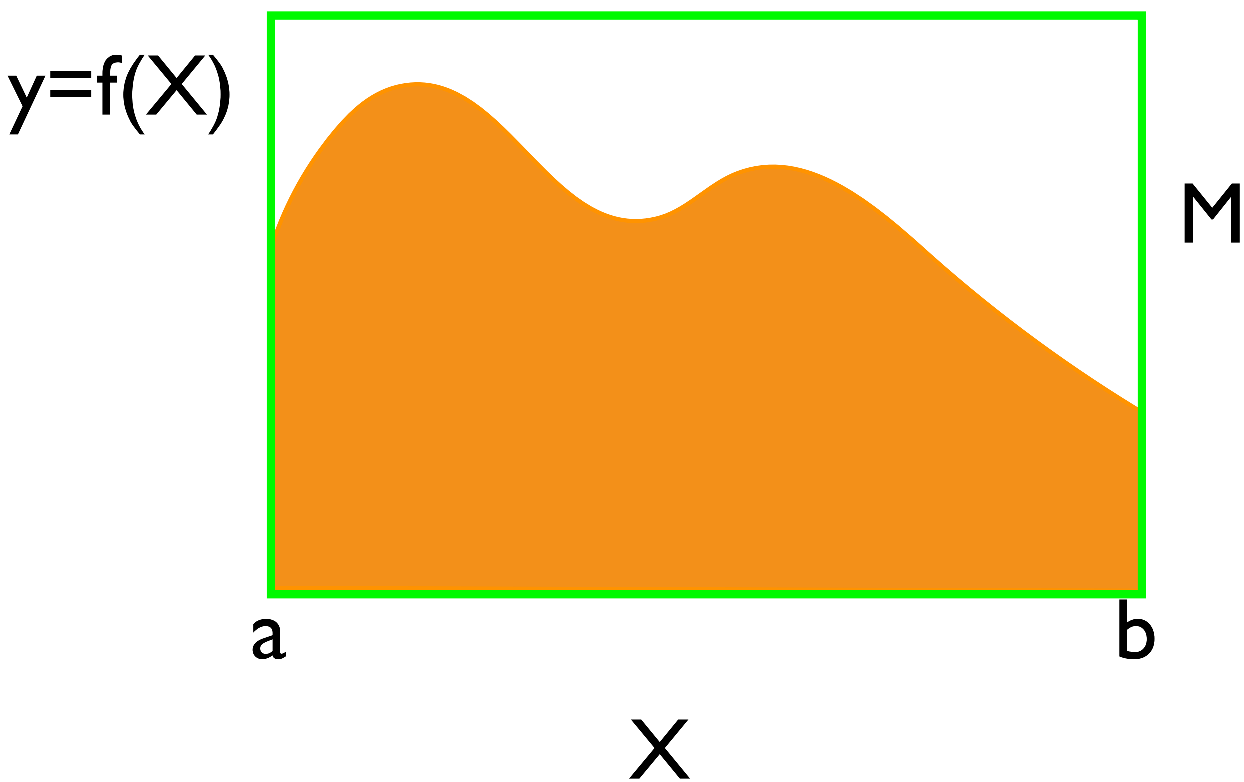
$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\Big[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\Big]$$

THE ACCEPTANCE-REJECTION METHOD

Fig. 4. If two independent sets of random numbers are used, one of which (x^i) extends uniformly over the range of the distribution function f and the other (y^i) extends over the domain of f, then an acceptance-rejection technique based on whether or not $y^i \leq f(x^i)$ will generate a distribution for (x^i) whose density is $f(x^i)\,dx^i$.



Acceptance-Rejection method



	Area
Bounding box	$A_b = M\,(b\text{-}a)$
Area under curve	$A_b \dfrac{\#inside}{\#bounding\ box}$

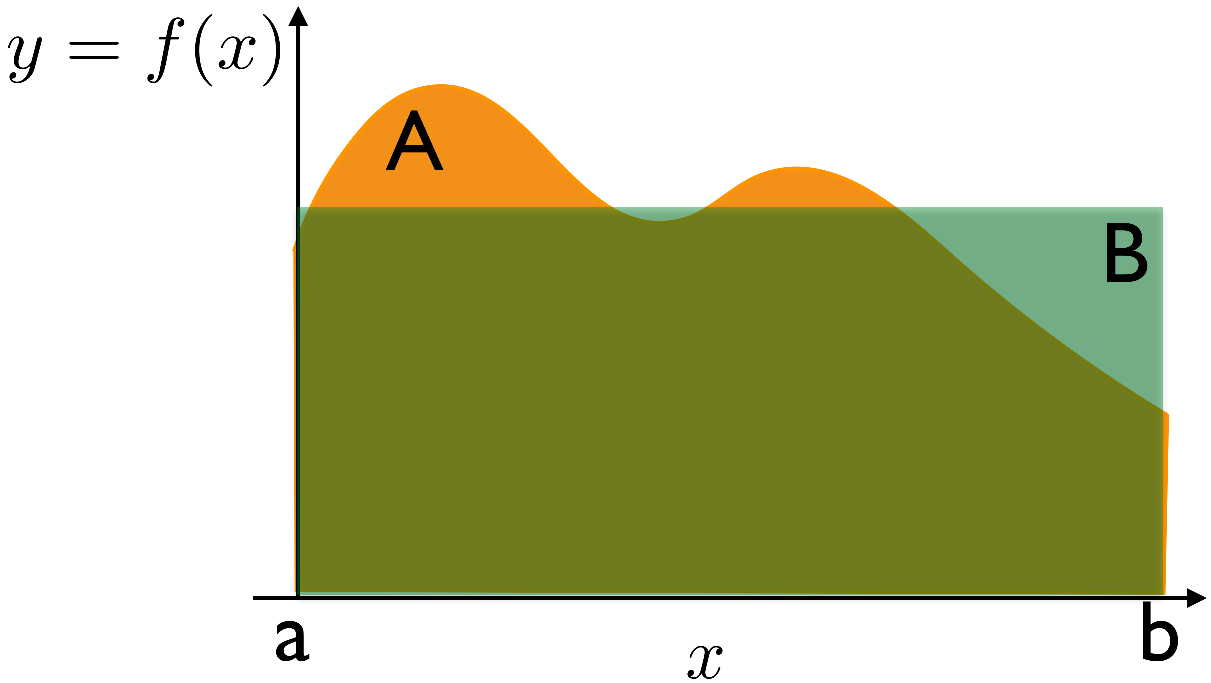
We need to be able to calculate f(x) for any possible x with the range a and b.

We draw 2 random values, one for x and one for y. then evaluate f(x), if $y < f(x)$ then we count this as #inside. The #bounding box is the total number of draws.

Problem with the Acceptance-Rejection method

- How to choose the bounding box? It need to be big enough to contain the whole function. But if it is too big then we draw often random numbers above the function. If the bounding box is much larger than the area under the curve then we need many draws (or steps) to get a good accuracy.
- We need to draw two random numbers and 'discard' the draws that are above the function.

Usual Monte Carlo Integration



$$A = \int_a^b f(x)dx$$

$$B = (b-a)f(c)$$

There must be an $f(c)$ that satisfies A = B.

Find $f(c)$ and we are done!

$$f(c) = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

where x_i are drawn uniformly between a and b

Monte Carlo Integration Algorithm

- Draw many x_i *between the boundaries a and b*
- *Calculate the average* $\overline{x}$ *of the collected* x_i
- *Calculate the area as* $(b-a)\ \overline{x}$

Evaluation of Monte Carlo error

We will discuss how to calculate error at the end of the semester when we again talk about Monte Carlo.