NUMERICAL METHODS
Week-14
14.05.2013
Monte Carlo Simulations

Asst. Prof. Dr. Berk Canberk

Monte Carlo Simulations (MCS)

- **What** is Monte Carlo Simulations?
- **Why** do we use Monte Carlo Simulations?
- **How** do we implement Monte Carlo Simulations?

What is Monte Carlo Simulations?

- Any method which solves a problem by generating suitable random numbers and observing that fraction of the numbers obeying some property or properties.
- It is a computerized mathematical technique that allows people to account for risk in quantitative analysis and decision making.
- Repeating random sampling to compute the result of a computational algorithm, which is difficult to solve analytically.
- **"Chance" ! → RANDOM NUMBERS→ MONTE CARLO SIMULATIONS!**

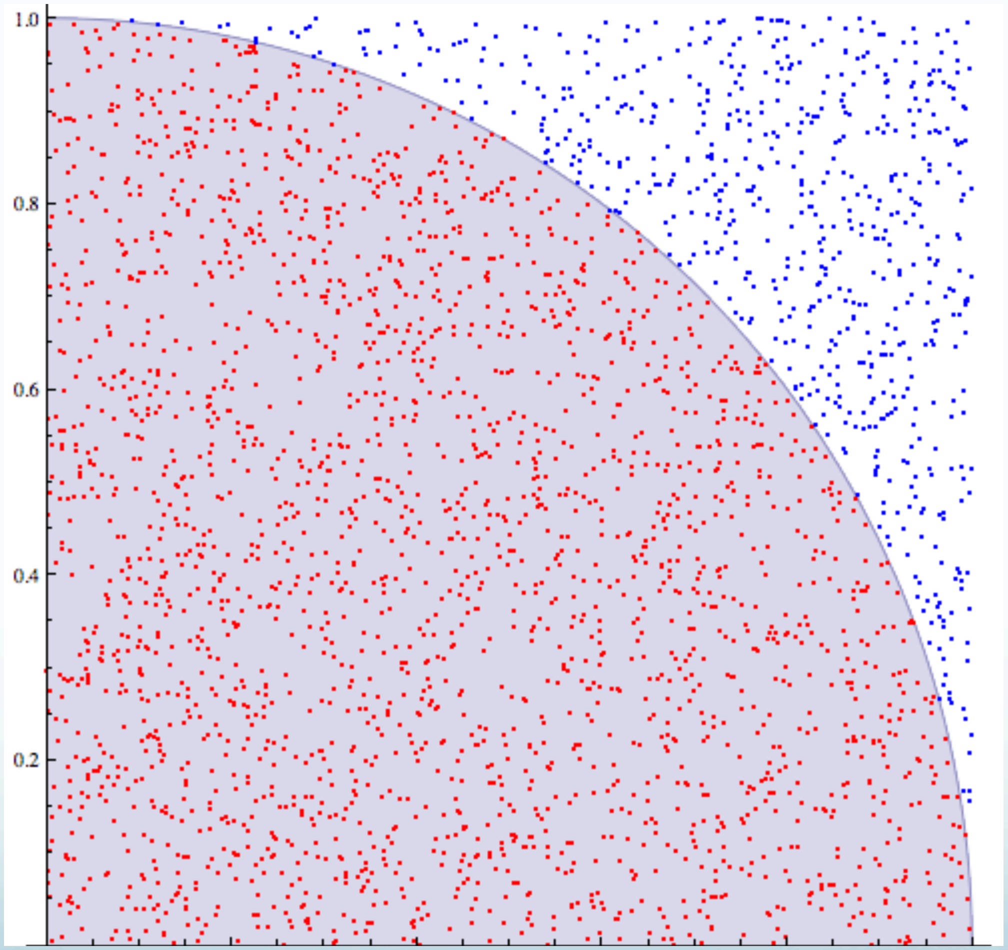
Why do we use MCS?

- The technique is used by professionals in such widely disparate fields as
  - finance,
  - project management,
  - energy, manufacturing, engineering,
  - insurance,
  - oil & gas,
  - transportation,
  - the environment.
- **Simulating systems with**
  - **random characteristics**
  - **Some degree of freedom**

How do we implement?

- Define the possible input domain
- Generate some RANDOM inputs using a probability distribution
- Compute and evaluate the input which fall into the underlying domain
- Repeat the process
- Finalize and aggregate the inputs.

Example: Computing Pi, π



1. Create a square “game board" and inscribe a circle in it (domain of infinitely many binary inputs, either a piece of rice lands on a space, or it doesn't, ignoring the boundary of the circle that is).

What is the ratio of the area of the circle and the corresponding square??

$$\frac{\text{Area of Circle}}{\text{Area of Square}} = \frac{\pi r^2}{l^2} = \frac{\pi \left(\frac{l}{2}\right)^2}{l^2} = \frac{\pi l^2}{4l^2} = \frac{\pi}{4}$$

2. Drop small pieces of rice onto this board uniformly at random. (generate inputs randomly).

3. Do this over and over again, say 10,000 times. (repeat).

4. Count the number of grains of rice that are in the circle. Assume that no piece of rice is on the boundary of the circle, and assume that we kept track of how many grains of rice we dropped. Then, the probability that a piece of rice landed in the circle is

$$p = \frac{\#\text{ of Grains of Rice in Circle}}{\#\text{ of Grains of Rice Dropped, Total}}$$

We know that the probability should be approximately π/4. Then,

$$p = \pi/4$$

But, if it were this simple, some mathematicians would be out of a job.

→The trick to getting a good approximation of is to use a huge number of grains of rice.

→If we were doing this experiment in real life, we would also probably want to make the square as large as possible.

Random Number Generation

- n random numbers x1,x2,…,xn in (0,1) in a 32-bit machine word-length

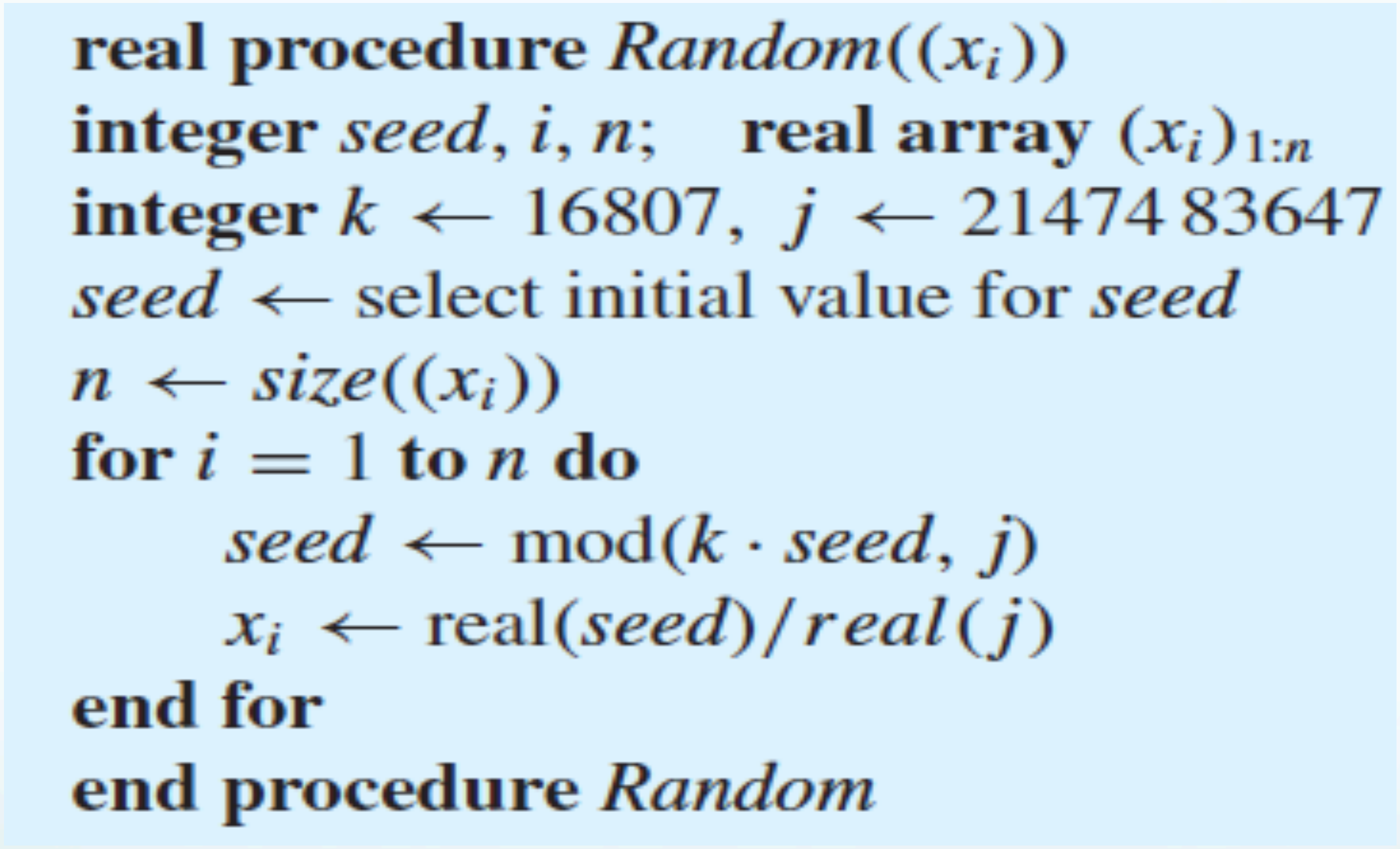
```
integer array $(\ell_i)_{0:n}$;    real array $(x_i)_{1:n}$
$\ell_0 \leftarrow$ any integer such that $1 < \ell_0 < 2^{31} - 1$
for $i = 1$ to $n$ do
    $\ell_i \leftarrow$ remainder when $7^5\ell_{i-1}$ is divided by $2^{31} - 1$
    $x_i \leftarrow \ell_i/(2^{31} - 1)$
end for
```

Here, all l_i ’s are integers in the range

$1 < \mathbf{l_i} < 2^{31} - 1$.

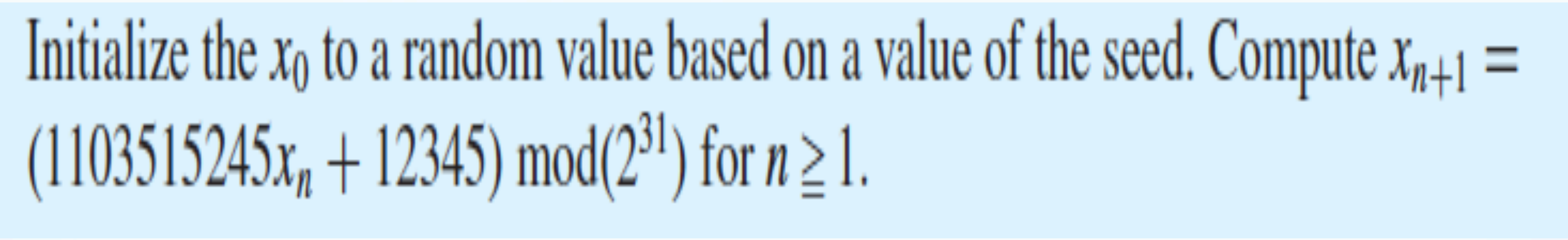
Seed: Initial integer l_0 within $(1, 2^{31} - 1)$

where $2^{31} - 1 = 2147483647$.



```
real procedure $Random((x_i))$
integer $seed, i, n$;    real array $(x_i)_{1:n}$
integer $k \leftarrow 16807,\ j \leftarrow 2147483647$
$seed \leftarrow$ select initial value for $seed$
$n \leftarrow size((x_i))$
for $i = 1$ to $n$ do
    $seed \leftarrow \text{mod}(k \cdot seed,\ j)$
    $x_i \leftarrow \text{real}(seed)/real(j)$
end for
end procedure $Random$
```

Random number in Unix (an example generation)



Initialize the x_0 to a random value based on a value of the seed. Compute $x_{n+1} = (1103515245x_n + 12345) \bmod(2^{31})$ for $n \geqq 1$.

Example

Consider the problem of generating 1000 random points **uniformly distributed inside the ellipse**

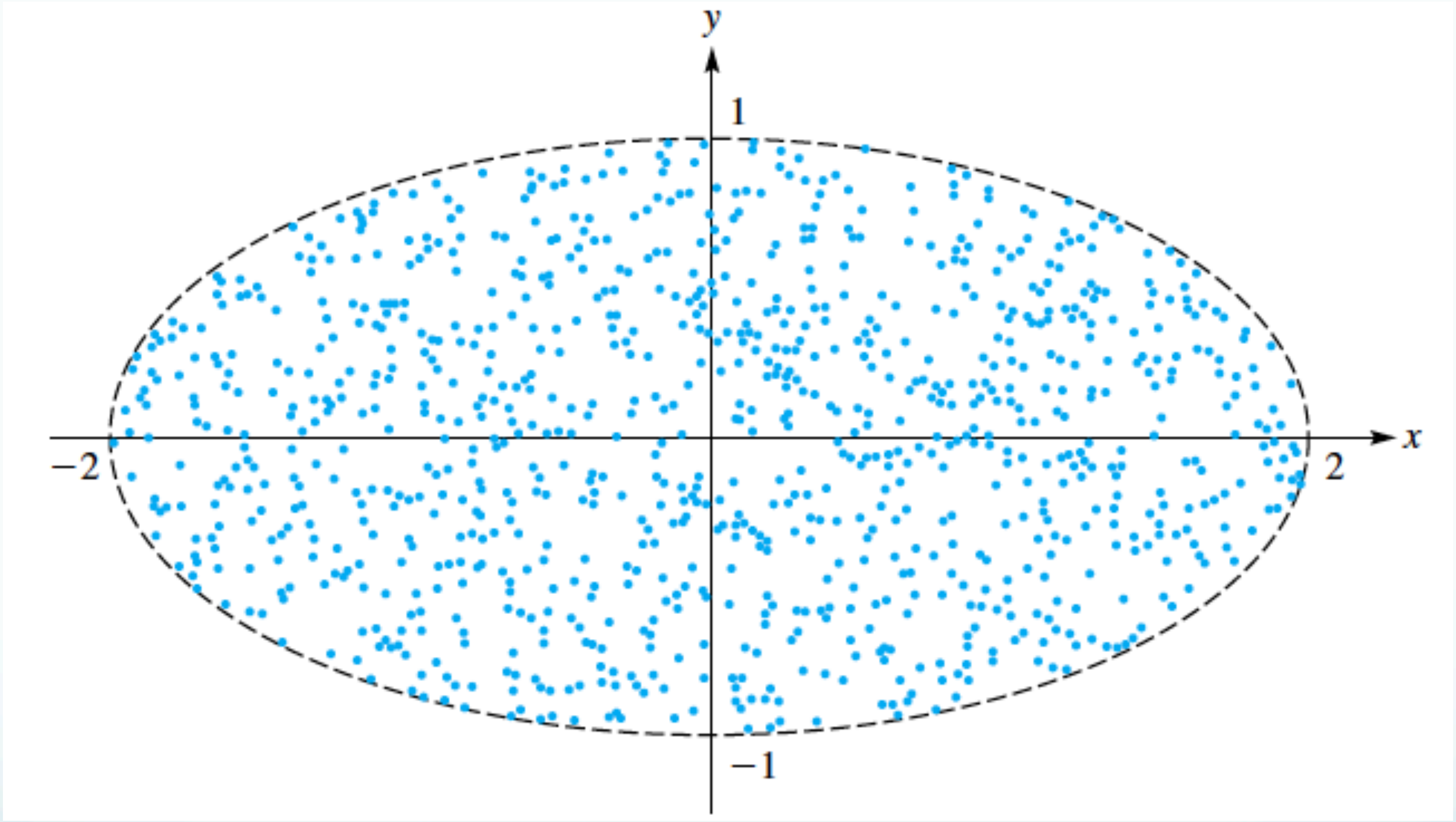
$$\mathbf{x}^2 + 4\mathbf{y}^2 = 4.$$

→One way to do so is to generate random points in the rectangle

$-2 \le \mathbf{x} \le 2,$

$-1 \le \mathbf{y} \le 1,$

and discard those that do not lie in the ellipse.

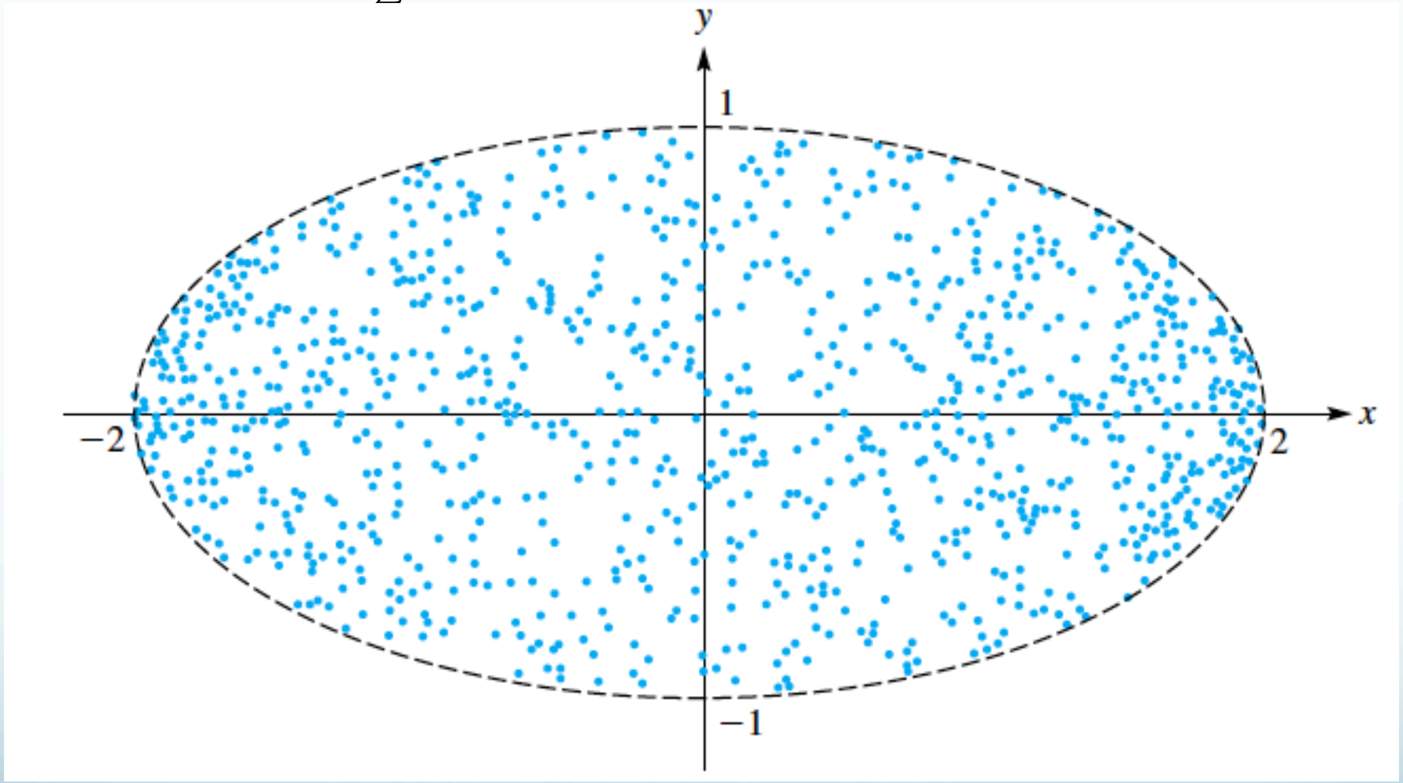


Uniformly Distributed random points in ellipse $\mathbf{x}^2 + 4\mathbf{y}^2 = 4$.

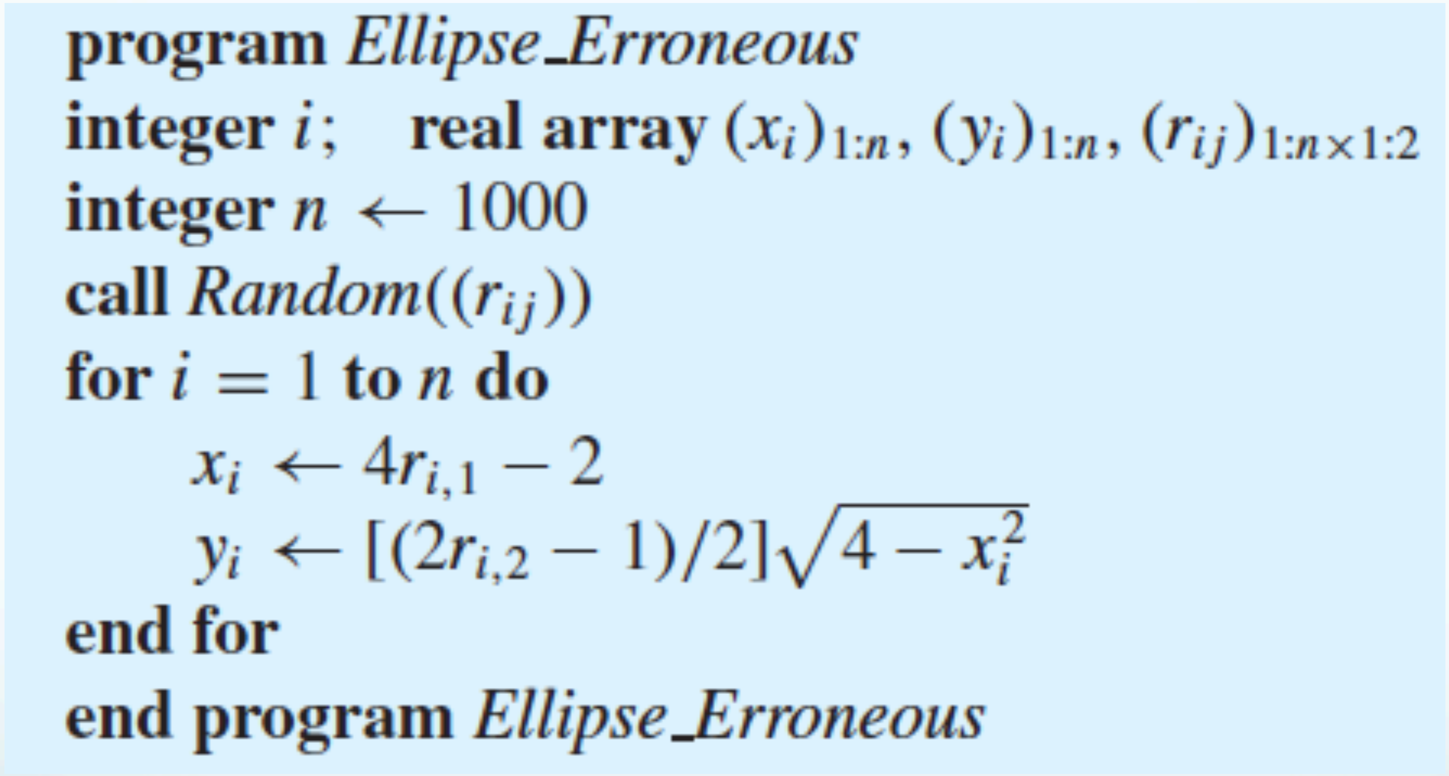
```
program Ellipse
integer $i, j$;   real $u, v$;   real array $(x_i)_{1:n}, (y_i)_{1:n}, (r_{ij})_{1:npts \times 1:2}$
integer $n \leftarrow 1000$, $npts \leftarrow 2000$
call $Random((r_{ij}))$
$j \leftarrow 1$
for $i = 1$ to $npts$ do
    $u \leftarrow 4r_{i,1} - 2$
    $v \leftarrow 2r_{i,2} - 1$
    if $u^2 + 4v^2 \leqq 4$ then
        $x_j \leftarrow u$
        $y_j \leftarrow v$
        $j \leftarrow j + 1$
    if $j = n$ then exit loop $i$
    end if
end for
end program Ellipse
```

Eclipse: Non-uniformly distributed case..

- Forcing $y \le \frac{1}{2}\sqrt{4 - x^2}$

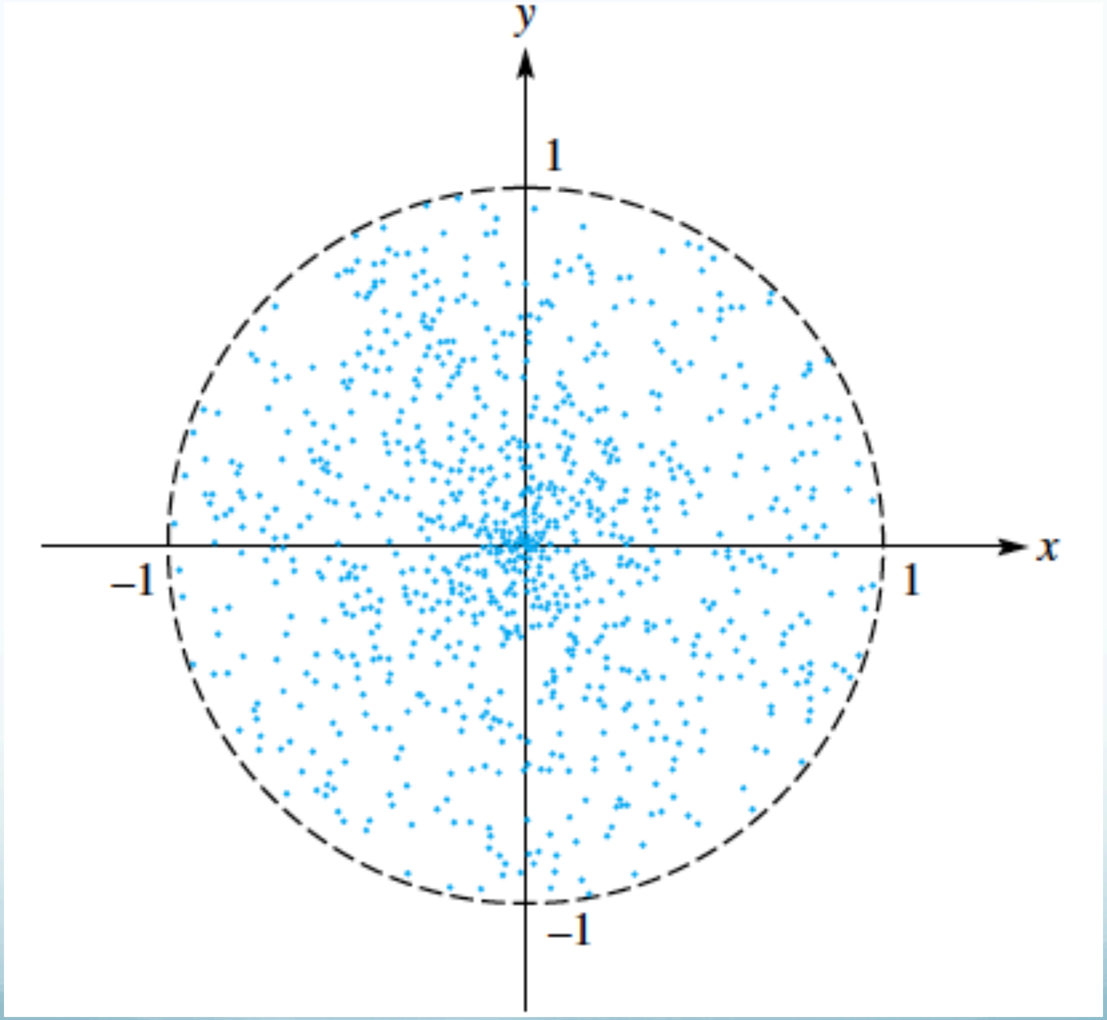


Uniformly Distributed random points in ellipse $\mathbf{x}^2 + 4\mathbf{y}^2 = 4$.



```
program $Ellipse\_Erroneous$
integer $i$;    real array $(x_i)_{1:n}, (y_i)_{1:n}, (r_{ij})_{1:n \times 1:2}$
integer $n \leftarrow 1000$
call $Random((r_{ij}))$
for $i = 1$ to $n$ do
    $x_i \leftarrow 4r_{i,1} - 2$
    $y_i \leftarrow [(2r_{i,2} - 1)/2]\sqrt{4 - x_i^2}$
end for
end program $Ellipse\_Erroneous$
```

Non-uniformly distributed random points in a circle



```
program Circle_Erroneous
integer $i$;    real array $(x_i)_{1:n}$, $(y_i)_{1:n}$, $(r_{ij})_{1:n \times 1:2}$
integer $n \leftarrow 1000$
call Random($(r_{ij})$)
for $i = 1$ to $n$ do
    $x_i \leftarrow r_{i,1} \cos(2\pi r_{i,2})$
    $y_i \leftarrow r_{i,1} \sin(2\pi r_{i,2})$
end for
end program Circle_Erroneous
```