

## **Pseudo Random Numbers**

---

Random Numbers  
Pseudo Random Numbers  
Statistical tests

### **Random numbers**

---

- ◆ Real random numbers are hard to obtain
  - ◆ cosmic radiation
  - ◆ atmospheric noise
  - ◆ used mainly for one-time encryption keys
  
- ◆ Pseudo random numbers
  - ◆ generating random numbers algorithmically is a sin!
    - ◆ not random at all
    - ◆ completely deterministic
  - ◆ look nearly random however when algorithm is not known
  - ◆ maybe good enough for our purposes
  
- ◆ Never trust pseudo random numbers however!

## Linear congruential generators

---

- ◆ are of the simple form  $x_{n+1}=f(x_n)$ , with  $f$  usually a linear function
- ◆ A good choice is the GGL generator

$$x_{n+1} = (ax_n + c) \bmod m$$

with  $a = 16807$ ,  $c = 0$ ,  $m = 2^{31}-1$ ,  $x_0=667790$

- ◆ quality depends sensitively on  $a,c,m$  and the seed
- ◆ Periodicity is a problem with such 32-bit generators
  - ◆ The sequence repeats identically after  $2^{31}-1$  iterations
  - ◆ With 500 million numbers per second that is just 4 seconds!
  - ◆ Nowadays such 32-bit generators should not be used!

## Lagged Fibonacci generators

---

- ◆ 
$$x_n = x_{n-p} \otimes x_{n-q} \bmod m$$
- ◆ Good choices for 64-bit floating point numbers ( $m=1$ )
  - ◆ (55,24,+)
  - ◆ (607,273,+)
  - ◆ (2281,1252,+)
  - ◆ (9689,5502,+)
  - ◆ (44497,23463,+)
- ◆ Seed blocks usually generated by linear congruential
- ◆ Has very long periods since large block of seeds
- ◆ no data dependencies for  $\min(p,q)$  iterations
  - ◆ can be vectorized on vector CPUs
  - ◆ can be pipelined on scalar CPUs

## Are these numbers really random?

---

- ◆ No!
- ◆ Are they random enough?
  - ◆ Maybe?
- ◆ How can we test?
  - ◆ Statistical tests for distribution
  - ◆ Statistical tests for short time correlations
  - ◆ Statistical tests for long time correlations
  - ◆ ...
- ◆ Are these tests enough?
  - ◆ No! Your calculation could depend in a subtle way on hidden correlations!
- ◆ What is the ultimate test?
  - ◆ Run your simulation with various random number generators and compare the results

## Easiest: graphical

---

- ◆ Before discussing statistical tests there is a simple first tool:
  - ◆ Create random pairs  $(x,y)$  and plot them
  - ◆ Create random triples  $(x,y,z)$  and plot them
- ◆ Can you see correlations?
- ◆ A Mathematica Notebook for these plots is on the web page of this course

## Non-uniform random numbers

---

- ◆ we found ways to generate pseudo random numbers  $u$  in the interval  $[0,1[$
- ◆ How do we get other uniform distributions?
  - ◆ uniform in  $[a,b[$ :  $a+(b-a)u$
- ◆ Other distributions:
  - ◆ inversion of integrated distribution
  - ◆ acceptance-rejection method

## Non-uniform distributions

---

- ◆ How can we get a random number  $x$  distributed with  $f(x)$  in the interval  $[a,b[$  from a uniform random number  $u$ ?
- ◆ Look at probabilities:

$$P[x < y] = \int_a^y f(t)dt =: F(y) \equiv P[u < F(y)]$$

$$\Rightarrow x = F^{-1}(u)$$

- ◆ This method is feasible if the integral can be inverted easily
  - ◆ exponential distribution  $f(x)=\lambda \exp(-\lambda x)$
  - ◆ can be obtained from uniform by  $x=-1/\lambda \ln(1-u)$

## Normally distributed numbers

---

- ◆ The normal distribution

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2)$$

- ◆ can be easily integrated in 2 dimensions!
- ◆ We can obtain two normally distributed numbers from two uniform ones (Box-Muller method)

$$n_1 = \sqrt{-2 \ln(1 - u_1)} \sin u_2$$

$$n_2 = \sqrt{-2 \ln(1 - u_1)} \cos u_2$$

## Uniform random numbers on $N$ -sphere

---

- ◆ random points  $\mathbf{s}$  on the surface of an  $N$ -sphere
- ◆ using acceptance-rejection
  - ◆ get uniform random vector  $\mathbf{x}$  with each component in  $[-1, 1[$
  - ◆ if norm is greater than one choose new one
  - ◆ normalize length to one
- ◆ using Box-Muller
  - ◆ start with uniform random vector  $\mathbf{x}$
  - ◆ use Box-Muller to get normally distributed vector  $\mathbf{n}$
  - ◆ normalize the length to one
- ◆ first method better only for very small  $N$

## Acceptance-rejectance method

---

- ◆ If the integral of the distribution function  $f$  cannot be inverted easily
- ◆ Look for a simpler distribution  $h$  that bounds  $f$ :

$$f(x) < \lambda h(x)$$

- ◆ Repeat
  - ◆ Choose one  $h$ -distributed number  $x$
  - ◆ Choose a uniform number  $u$
- ◆ Until  $u < f(x)/\lambda h(x)$
- ◆ Needs a good guess  $h$
- ◆ Where that is not possible numerical inversion of integral might be faster!