Random-Variable Generators

Rationale

- Simulators need generators of random variable with specific distributions
- The actual procedure is:

1. We generate a sequence X of integer random numbers
2. Using X, we generate a sequence U of instances of a random variable uniform in [0,1]
3. Using U, we generate instances of the chosen random variable

Inverse-transform technique

- We want to generate instances of a random variable X with cumulative distribution $F(x)$
- The inverse-transform technique is based on the fact that $U = F(X)$ is a r.v. uniformly distributed in [0,1]
- Therefore, we can generate an instance of u uniform and then compute

$$X = F^{-1}(U)$$

Inverse-transform technique

Proof that U =F(X) is uniform in [0,1]:

- Let Y=g(X) be function of X monotone increasing (and therefore invertible), X=g^{-1}(Y)

$$F_Y(y) = P(Y \le y) = P(g^{-1}(Y) \le g^{-1}(y))$$
$$= P(X \le g^{-1}(y)) = F_X(g^{-1}(y))$$

- Select g(·) such that g(X)=F_X(X), or also Y=F_X(X), and 0≤y≤1,

$$F_Y(y) = F_X(g^{-1}(y)) = F_X(F_X^{-1}(y)) = y$$

Y is uniform in [0,1]!

Inverse-transform technique

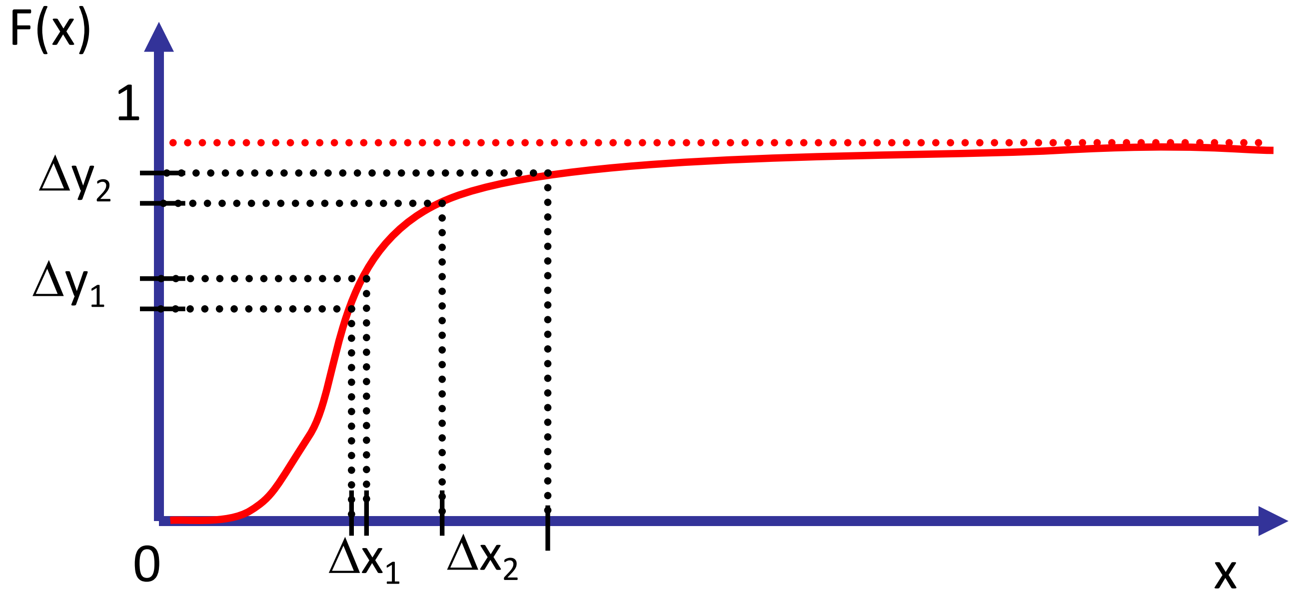
- In alternative, from $F_y(y)=y$, for $0<y<1$

$$f_y(y) = dF_y / dy = 1$$

hence Y is uniform in [0,1]

Generating Y uniform in [0,1]
and computing the inverse, $F^{-1}(Y)$,
we obtain X

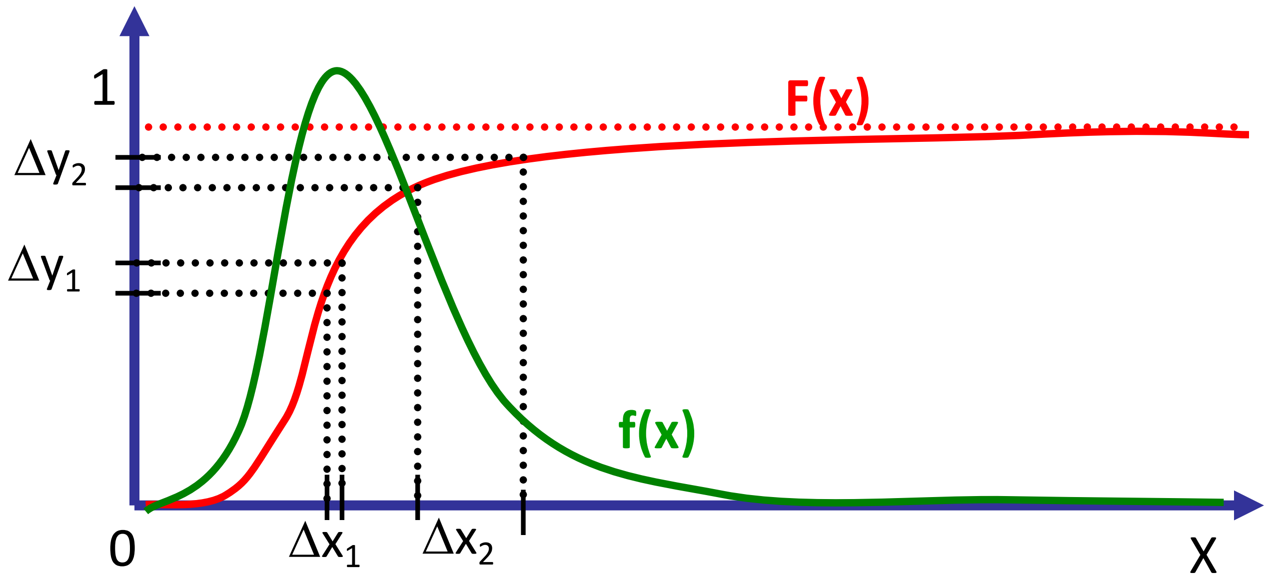
Inverse-transform technique



- If $\Delta y_1 = \Delta y_2$, the probability of generating an instance of X in Δx_1 or in Δx_2 is the same
- The density of samples in Δx_1 is larger than those in Δx_2, because Δx_1 is narrower

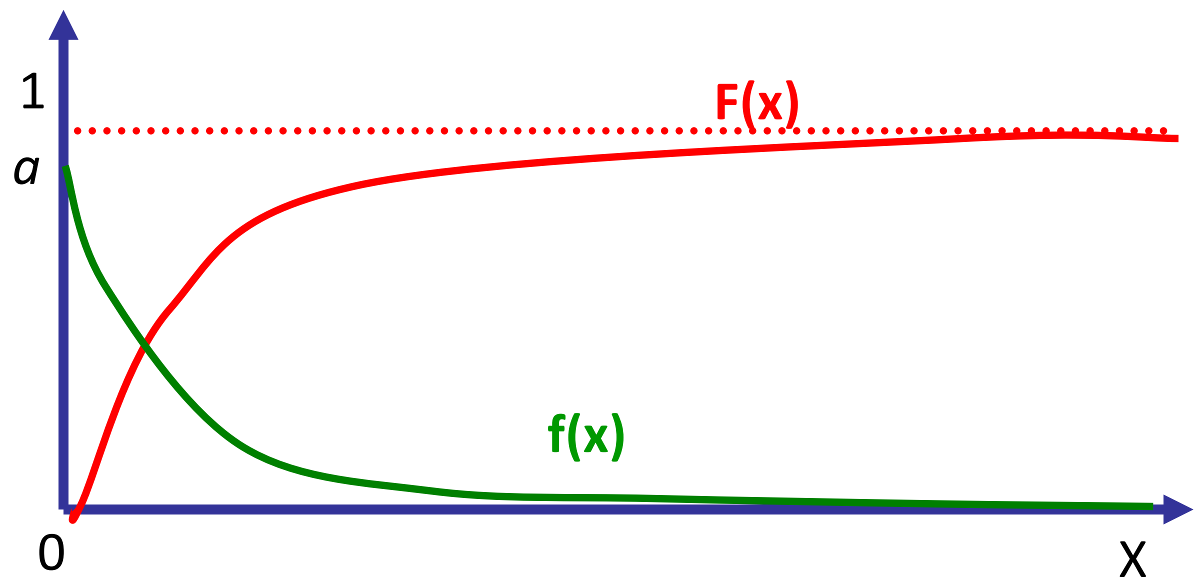
Inverse-transform technique

Intervals on the x axis are narrower if the derivative of F(x) is higher, i.e., f(x) is larger



Exponential

- We want to generate instances of a random variable with exponential distribution and rate a, x≥0,
  - $f(x) = a\, e^{-ax}$, $F(x) = 1 - e^{-ax}$



Exponential

- The inverse of F(x) is

$$y = 1 - e^{-ax}$$

$$e^{-ax} = 1 - y$$

$$-ax = \ln(1 - y)$$

$$x = -\frac{1}{a}\ln(1 - y)$$

Exponential

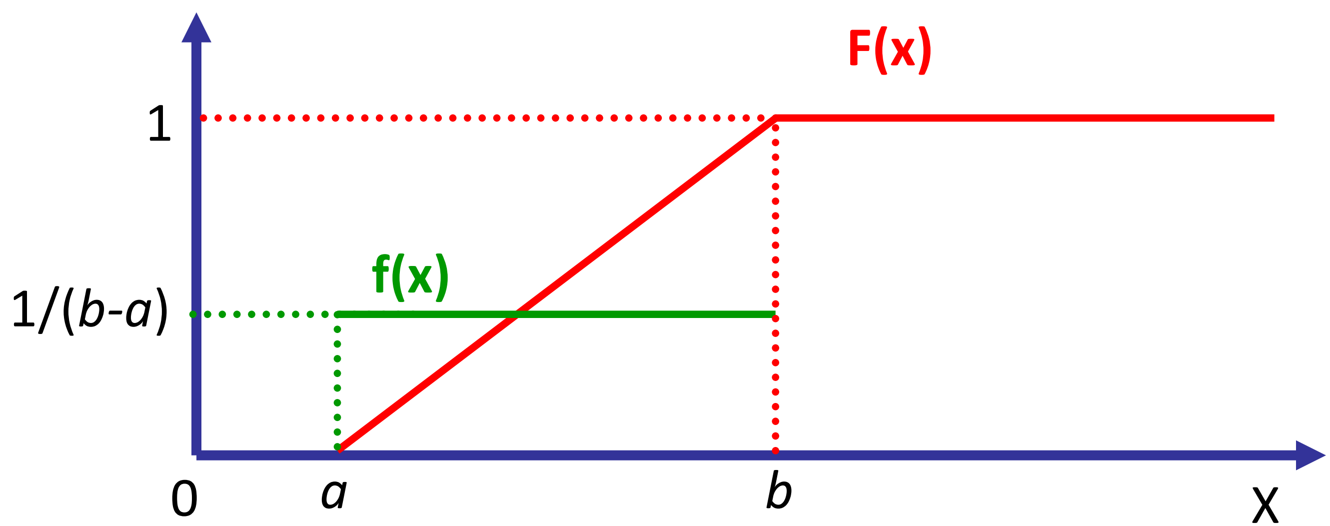
1. Generate u =U(0,1), uniform in (0,1)
2. Calculate $x = -1/a \ln(1-u)$
3. Return x

- Since both u and $1-u$ are uniformly distributed in (0,1), x can also be calculated as

$$x = -1/a \ln(u)$$

Uniform

- We want to generate instances of a uniform random variable with support [a,b]
  - $f(x) = 1/(b-a)$, $F(x) = (x-a)/(b-a)$, $a \leq x \leq b$



Uniform

- The inverse of F(x) is

$$y = \frac{(x-a)}{(b-a)}$$

$$(b-a)y = x - a$$

$$x = a + (b-a)y$$

Notice that the function goes from -inf to +inf but since y is in [0,1], x falls in support of the r.v. X

1. Generate u =U(0,1), uniform in (0,1)
2. Calculate $x = a + (b-a)\, u$
3. Return x

Pareto

- Pareto distribution (k=1)
  - $f(x) = ax^{-(a+1)}$, $F(x) = 1-x^{-a}$ $x \geq 1$
- Inverse:

$$y = 1 - x^{-a}$$

$$x^{-a} = 1 - y$$

$$x = \frac{1}{(1-y)^{1/a}}$$

Pareto

1. Generate u = U(0,1), uniform in (0,1)
2. Calculate $x = 1/u^{1/a}$
3. Return x

Weibull

- Weibull distribution
  - $F(x) = 1\text{-}e^{-(x/a)^b}$
- Inverse:

$$y = 1 - e^{-(x/a)^b}$$

$$e^{-(x/a)^b} = 1 - y$$

$$-\left(\frac{x}{a}\right)^b = \ln(1 - y)$$

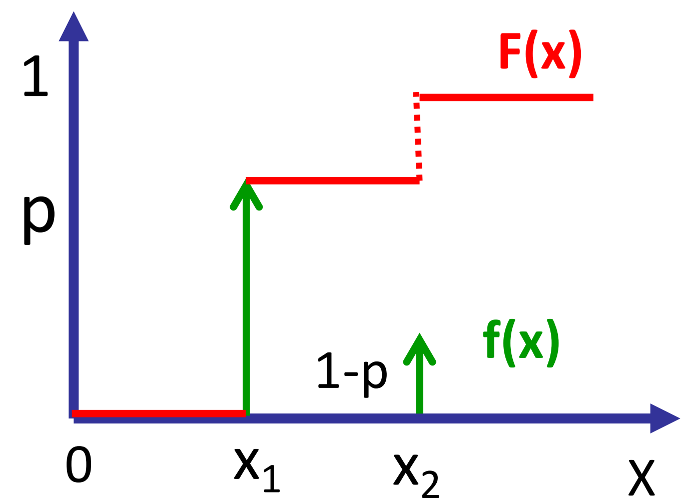
$$x = a[-\ln(1 - y)]^{1/b}$$

Weibull

1. Generate u=U(0,1), uniform in (0,1)
2. Calculate $x = a(-\ln(u))^{1/b}$
3. Return x

Bernoulli

- Bernoulli distribution
  - $P(X=x_1)=p$, $P(X=x_2)=1-p$ e $0<p<1$
  - It represents events with two possible outcomes (success/failure, true/false)
- Generation
  1. Generate u=U(0,1), uniform in (0,1)
  2. If $u \leq p$ return $x=x_1$
  3. otherwise return $x=x_2$



Geometric

- Geometric distribution
  - The probability of X is $P(X=n) = p\,(1-p)^n$, with $n=0,1,2,\ldots$ and $0<p<1$
  - The CDF is: $P(X\leq n) = 1-(1-p)^{n+1}$
- Inverse:

$$y = 1-(1-p)^{x+1}$$

$$(1-p)^{x+1} = 1-y$$

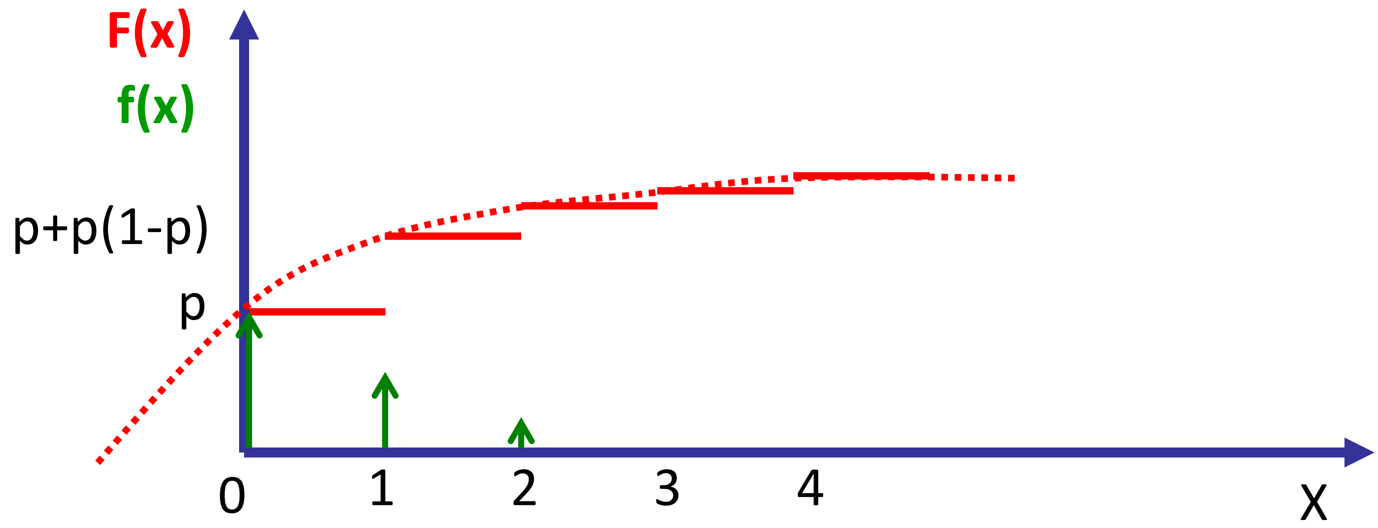
$$(x+1)\ln(1-p) = \ln(1-y)$$

$$x = \frac{\ln(1-y)}{\ln(1-p)} - 1$$

Extending the domain of the function to $\mathbb{R}$

Geometric

$$x = \frac{\ln(1-y)}{\ln(1-p)} - 1 \quad \Rightarrow \quad x = \left\lceil \frac{\ln(u)}{\ln(1-p)} - 1 \right\rceil$$



Geometric

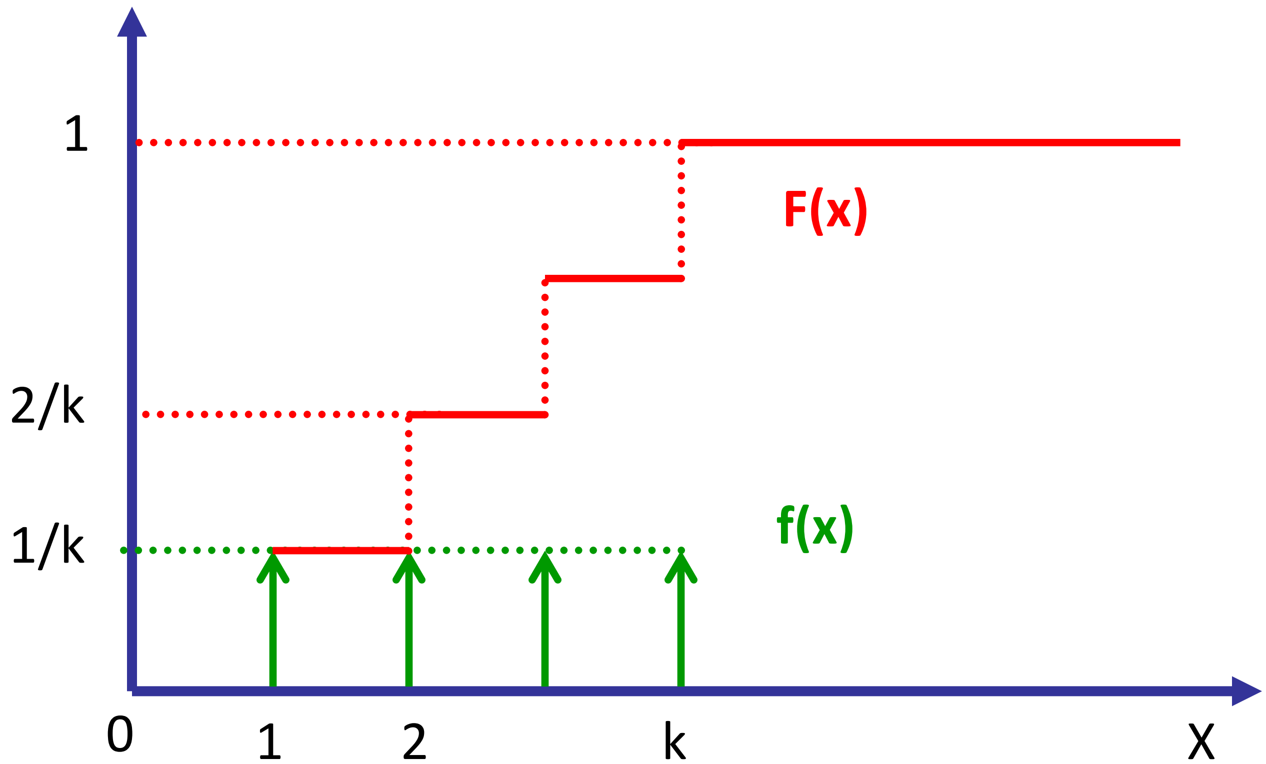
1. Generate u=U(0,1), uniform in (0,1)
2. Calculate x = ceil(ln(u)/ln(1-p) -1)
3. Return x

- For the geometric with probability $f(x)=p\,(1-p)^{x-k}$, with x=k,k+1,k+2,...

$$x = k + \left\lceil \frac{\ln(u)}{\ln(1-p)} - 1 \right\rceil$$

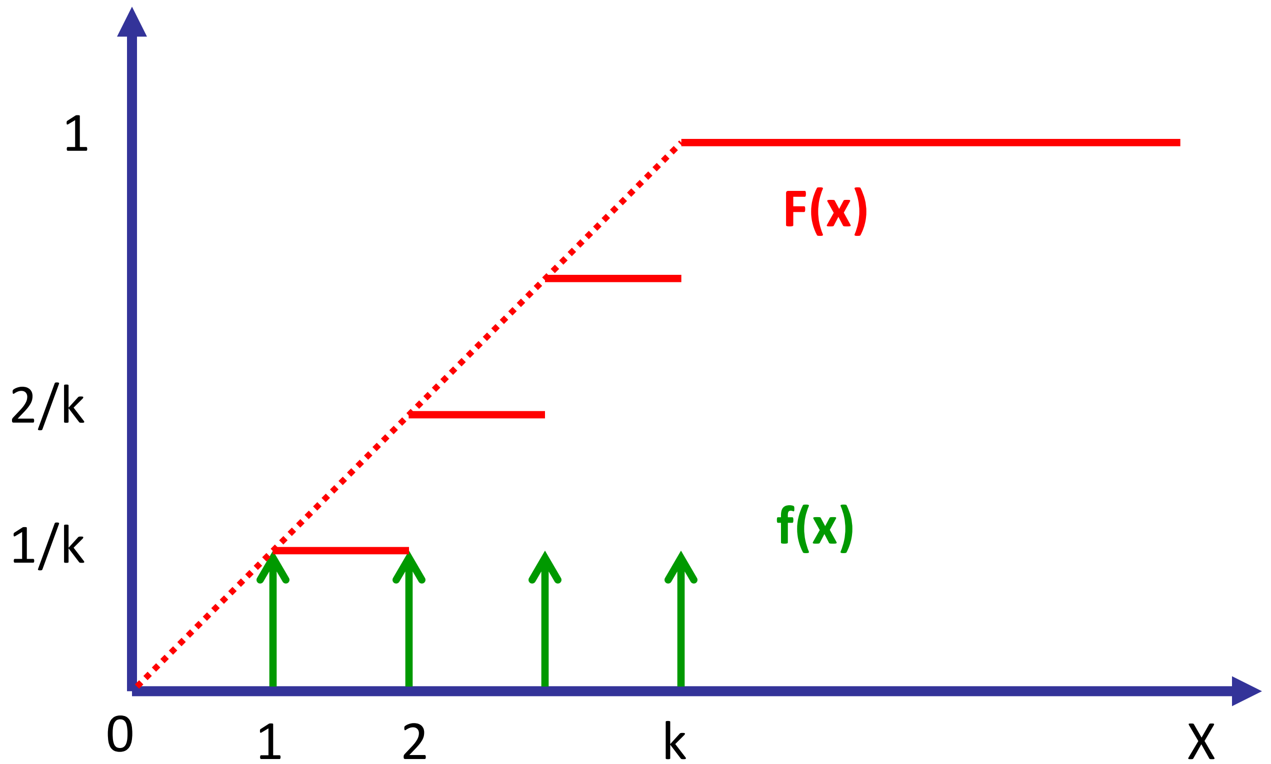
Discrete uniform distribution

- We want to generate an instance x of X, with x in {1,2,…,k} and p(X=x)=1/k



Discrete uniform distribution

- We have F(x)=*x*/k with *x* =1,2,...,k



Discrete uniform distribution

- The inverse is $x = yk$
- X can be calculated as $x = \lceil ku \rceil$
- Indeed, after generating u uniform in (0,1), we return x if

$$\frac{x-1}{k} < u \le \frac{x}{k}$$

that is

$$x - 1 < ku \le x$$

$$ku \le x < ku + 1$$

Empirical discrete distributions

- We want to generate instances of a random variable whose distribution is computed empirically through
  - Measurements
  - Approximations (e.g., when the inverse cannot be expressed in closed form)
- X holds the values $x_1, x_2, \ldots, x_k$ with probability $p_1, p_2, \ldots, p_k$
- We derive the cumulative
- We compute the inverse

Empirical discrete distributions

- The cumulative distribution is empirically derived by the measured or approximated values

x values	x_1	x_2	x_3	…	x_k
Probability	p_1	p_2	p_3	…	p_k
CDF, F(x)	0 $0 \leq x < x_1$	p_1 $x_1 \leq x < x_2$	p_1+p_2 $x_2 \leq x < x_3$	…	1 $x_k \leq x$

Empirical discrete distributions

- To generate instances of X, we can use the following procedure:

1. Generate u=U(0,1), uniform in (0,1)
2. Return *x*, according to which condition is satisfied by u in the following table

Returned value of x	x_1	x_2	…	X_k
Condition	$0<u\le F(x_1)$ $0<u\le p_1$	$F(x_1)<u\le F(x_2)$ $p_1<u\le p_1+p_2$	…	$F(x_{k-1})<u\le F(x_k)$ $p_1+\ldots+p_{k-1}<u\le 1$

Inverse-transform technique

- To apply it, we must be capable to derive the inverse of F(x)
- We just need to generate a single random number U(0,1)
- The computational cost depends on the computational complexity of the inverse function

Convolution method

- It is used when a random variable can be expressed as sum of other r.v. that can be easily generated

$$x = y_1 + y_2 + \ldots y_n$$

- We generate an instance for each y_i
- Adding all the y_i we obtain an instance of x

Convolution method

- Examples in which we can apply the convolution method:
  - Erlang-K: it is the sum of K random variables with exponential distribution
  - Binomial with parameters n and p : it is the sum of n Bernoulli distributed variables with success probability p
  - Chi-square with n degrees of freedom: it is the sum of the squares of n normal r.v. N(0,1)

Erlang-K distribution

- The Erlang-K with mean $1/m$ is the sum of K r.v. with exponential distribution with mean $1/(Km)$,
$$X = \sum_{i=1}^{K} X_i$$
- An instance x of X is obtained by summing instances x_i of X_i
$$x = \sum_{i=1}^{K} \left(-\frac{1}{Km} \ln(u_i) \right) = -\frac{1}{Km} \ln \left(\prod_{i=1}^{K} u_i \right)$$
- If K is large, it might be inefficient

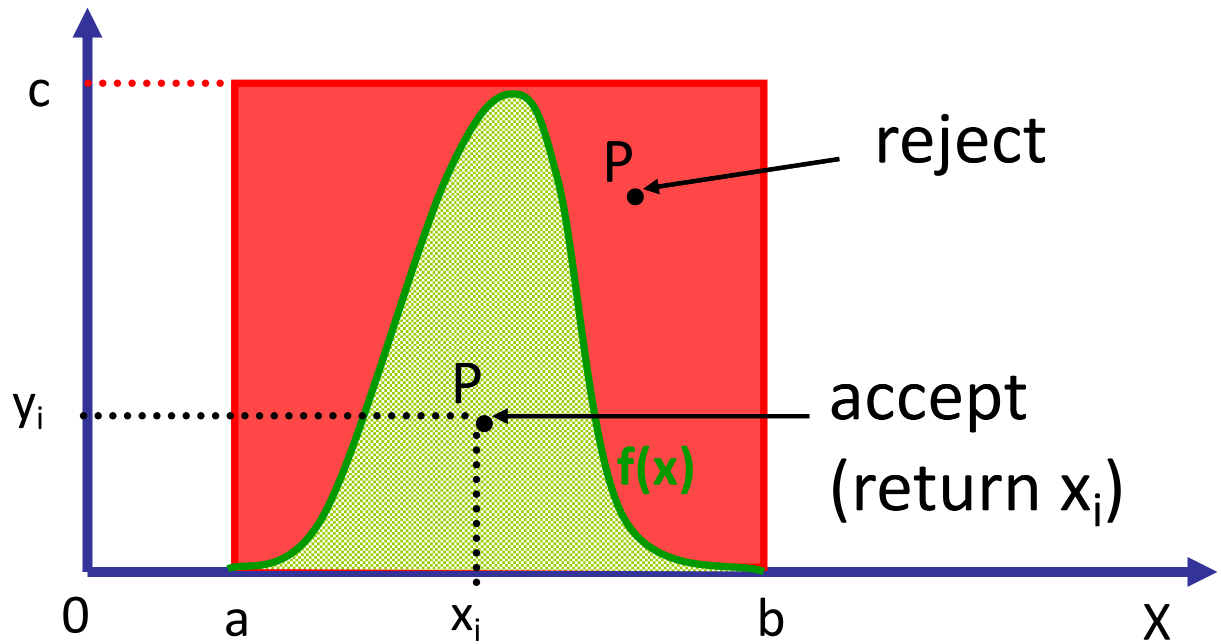
Acceptance/Rejection Technique

- The acceptance/rejection technique can be applied to random variables with continuous pdf f(x) defined over finite support [a,b]
- Being c the maximum value for f(x), we apply the following procedure:

1. Generate x_i =U(a,b), uniform in [a,b]
2. Generate y_i =U(0,c), uniform in [0,c]
3. If $y_i \leq f(x_i)$ return x_i, otherwise go back to step 1

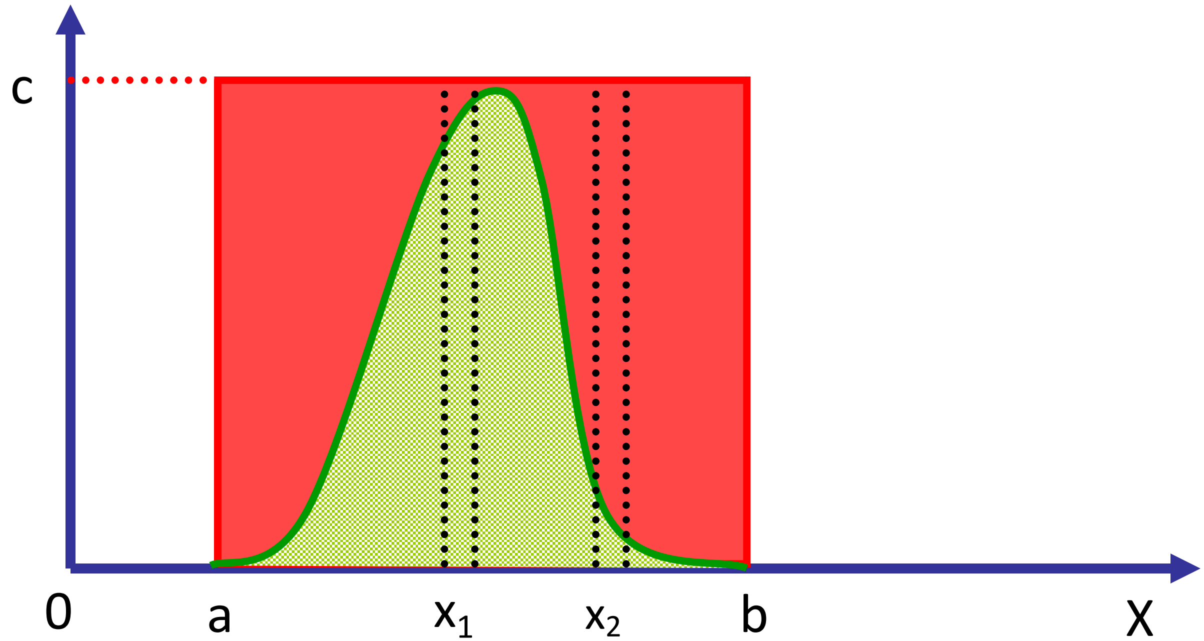
Acceptance/Rejection Technique

- Generating x_i =U(a,b) and y_i =U(0,c) means generating a random point P in the rectangle delimited by (a,b) on the x-axis and (0,c) on the y-axis



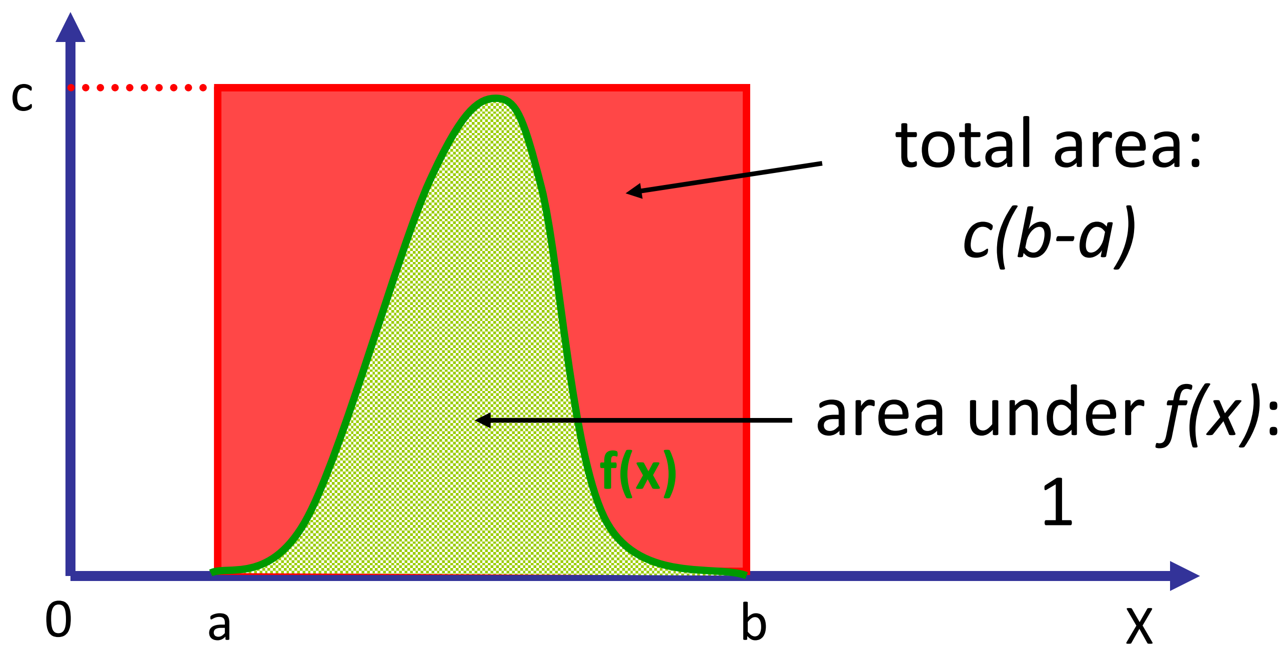
Acceptance/Rejection Technique

- The acceptance probability in an interval (x_1, x_1+dx) of the x-axis is proportional to $f(x_1)$



Acceptance/Rejection Technique

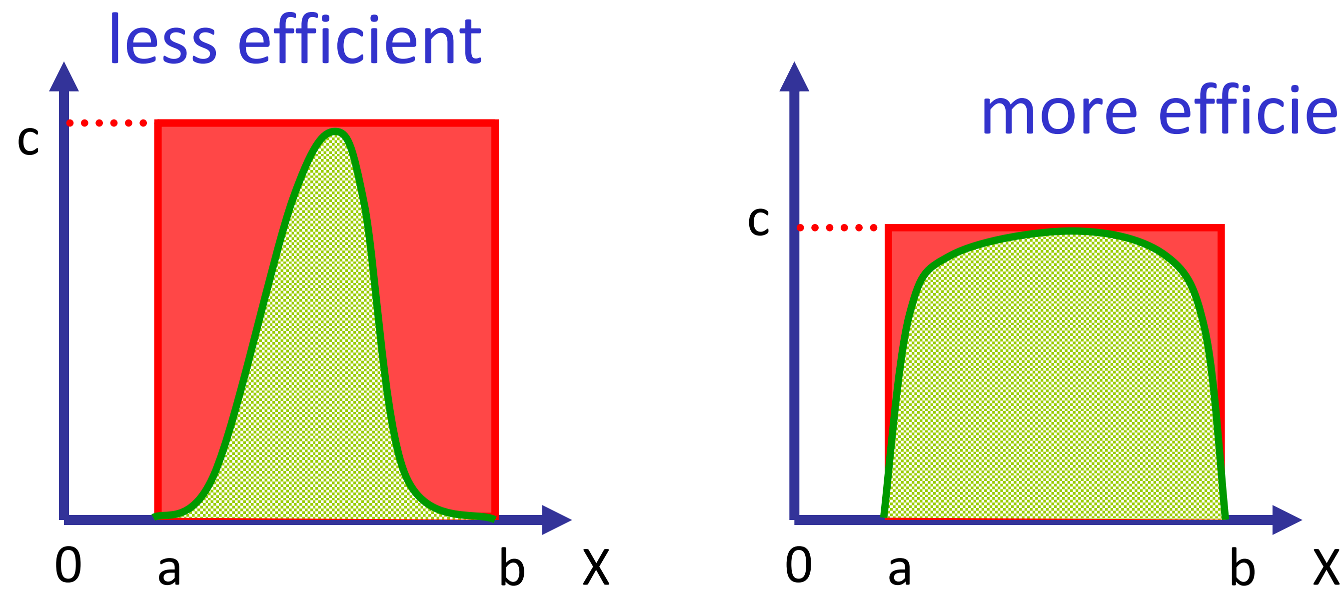
- The efficiency of the technique depends on the area of the rectangle delimitating *f(x)*, i.e., on how much the rectangle is a good approximation of *f(x)*



Acceptance/Rejection Technique

- The acceptance probability *P* for a random generated point is equal to the ratio between the area under *f(x)* and the total area:

$$P = 1/[c(b-a)]$$



Acceptance/Rejection Technique

- The average number of points generated to obtain an instance of *X* is

$$N = \sum_{i=1}^{\infty} i(1-P)^{i-1} P = \frac{1}{P}$$

- The technique can be applied to variables with infinite support only by approximation, truncating its support

Acceptance/Rejection Technique

- In general, if there exists a pdf $g(x)$ such that $kg(x) \geq f(x)$, we can obtain instances of X (with pdf $f(x)$) with the procedure:

1. Generate x_i from distribution $g(x)$
2. Generate $y_i = U(0, kg(x_i))$, uniform in $[0, kg(x_i)]$
3. If $y_i \leq f(x_i)$ return x_i, otherwise go back to step 1

- In this general form, acceptance probability P is now $P = 1/k$

Composition method

- This method is applied to random variables whose CDF can be expressed as weighted sum of other CDFs

$$F(x) = \sum_{i=1}^{n} p_i F_i(x), \quad \text{with} \quad p_i > 0, \sum_{i=1}^{n} p_i = 1$$

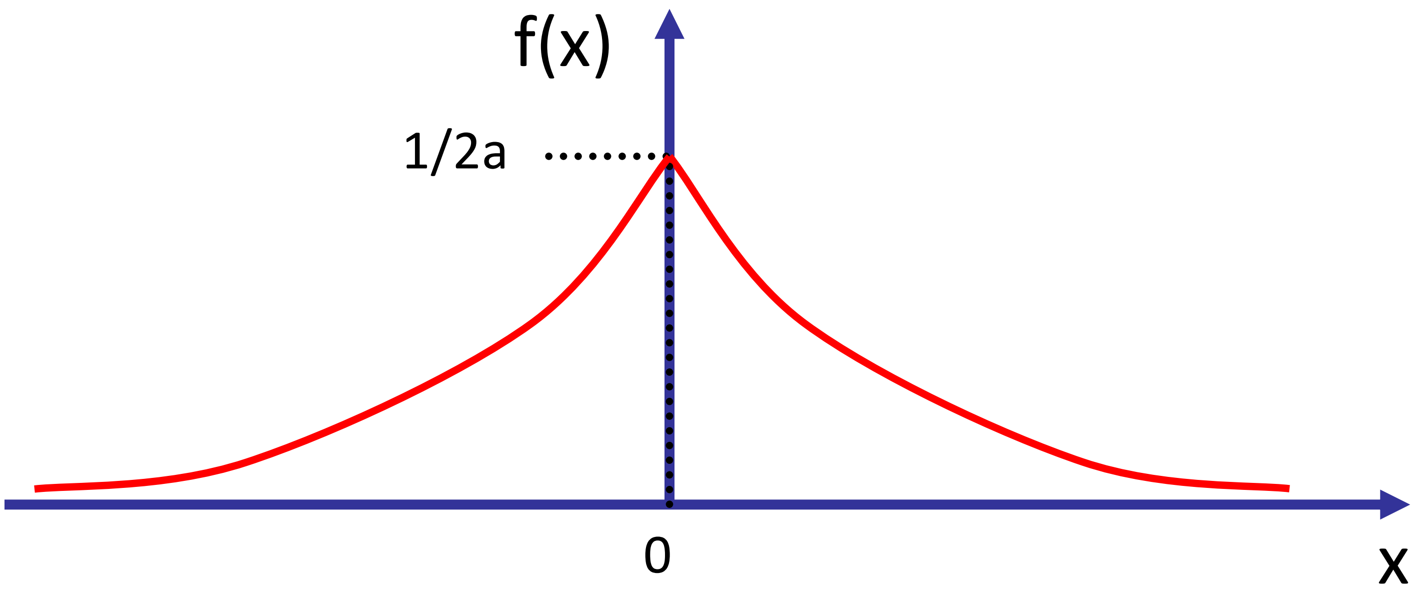
1. Generate an instance of the r.v. *I*, such that $P(I=i)=p_i$
2. Generate an instance of the r.v. with CDF $F_i(x)$

- The method can be also applied to the pdf $f(x)$

Laplace distribution

- We want to generate *X* with pdf

$$f(x) = \frac{1}{2a} e^{-|x|/a}$$



Laplace distribution

- The Laplace distribution is the composition of two exponential r.v., with probability ½ it is positive and with the same probability it is negative

1. Generate $u_1=U(0,1)$, and $u_2=U(0,1)$
2. If $u_1<1/2$ return $x = a\ ln(u_2)$

   otherwise return $x = -\ a\ ln(u_2)$

- Notice: this distribution can be generated more efficiently with the inverse-transform technique

Other methods

- There are a few other methods applied to some special distributions
- These *ad hoc* methods are usually based on mathematical proprieties that are satisfied by some random variables

Normal distribution

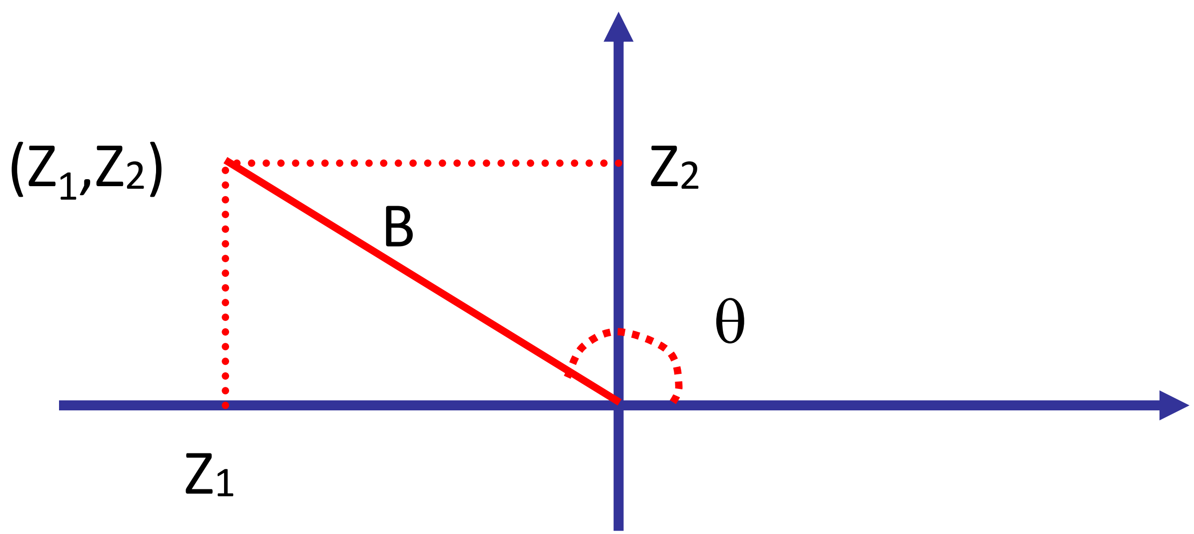
- We want to generate X with normal (Gaussian) distribution, mean 0 and variance 1, N(0,1),

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

We cannot apply the inverse-transform technique because the inverse cannot be written in closed form

Normal distribution

- We use a few properties of the normal distribution
- Let’s consider 2 normal independent r.v. considered as Cartesian coordinates of a point in a plane



Normal distribution

- Expressing the point in polar coordinates, we have
  - $Z_1 = B \cos \theta$
  - $Z_2 = B \sin \theta$
  - $B^2 = Z_1^2 + Z_2^2$
- We generate separately B and θ, then we calculate Z_1 and Z_2

Normal distribution

- The variable B^2 is distributed according to a chi-square with 2 degrees of freedom, i.e. it is an exponential with mean 2
- B^2 is generated with the inverse- transform
  1. Generate $u=U(0,1)$
  2. Calculate $B = sqrt(-2ln(u))$
  3. Return B

Normal distribution

- The angle θ is distributed uniformly in $(0,2\pi)$ and it is independent from B
- θ is also generated with the inverse-transform
  1. Generate u=U(0,1)
  2. Calculate $\theta = 2\pi u$
  3. Return θ

Normal distribution

The procedure is hence the following one:

1. Generate u=U(0,1)
2. Generate v=U(0,1)
3. Calculate B=sqrt(-2ln(u))
4. Calculate $\theta = 2\pi$ v
5. Calculate $Z_1 = B \cos(\theta)$
6. Calculate $Z_2 = B \sin(\theta)$

Normal distribution

- The procedure generates pairs of instances of a normal r.v. at the cost of generating two instances of uniform r.v. and some algebraic operations
- Given an instance Z of a normal r.v. $N(0,1)$, instances of a r.v. X normal with mean μ and variance σ^2 are obtained from

$$X = \mu + \sigma Z$$

Normal distribution

- The previously proposed method might not be efficient (logarithms and trigonometric functions are complex)
- If an approximation is sufficient, we can use the Central Limit Theorem to generate a normal r.v.

$$N(\mu,\sigma) \approx \mu + \sigma \frac{\left(\sum_{i=1}^{n} u_i\right) - n/2}{(n/12)^{1/2}}$$

- Generally, *n=12* is used

Poisson distribution

- We want to generate instances of a r.v. *X* with Poisson distribution and mean *a*,

$$p(n) = P(X = n) = \frac{e^{-a} a^n}{n!}, \quad n = 0,1,2,\cdots$$

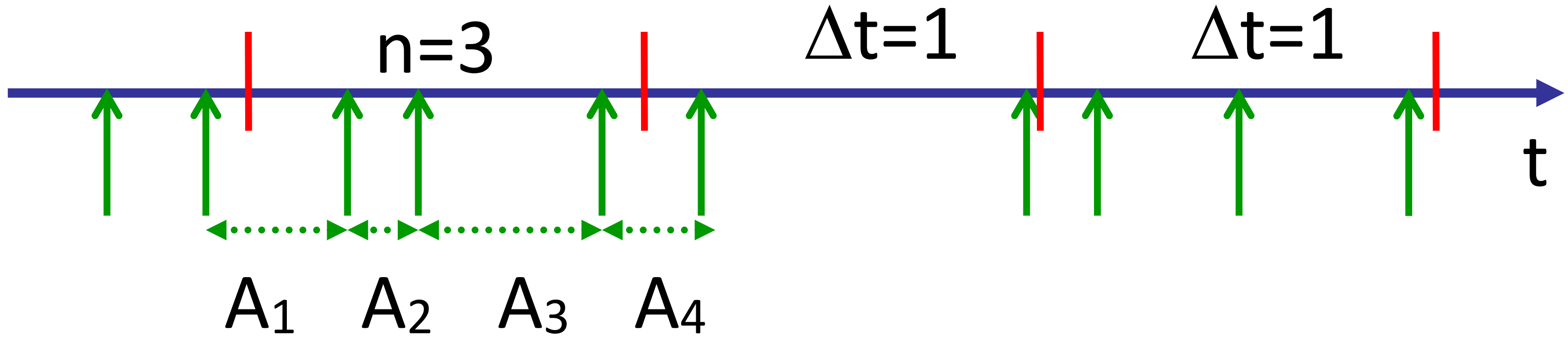
- *X* represents the number of arrivals of a Poisson process in the time unit, when *a* is the average number of arrivals in the time unit

Poisson distribution

- $X=n$ if and only if the following relation holds

$$A_1+A_2+\cdots+A_n \leq 1 < A_1+A_2+\cdots+A_n+A_{n+1}$$

where A_i is the i-th interarrival time



Poisson distribution

- Interarrival times A_i are distributed according to an exponential distribution with mean *1/a*
- Hence, we can generate an instance of *X* generating interarrival times and calculating how many arrivals we have in the time unit

Poisson distribution

$$A_1 + \cdots + A_n \le 1 < A_1 + \cdots + A_n + A_{n+1}$$

$$\sum_{i=1}^{n} -\frac{1}{a}\ln(u_i) \le 1 < \sum_{i=1}^{n+1} -\frac{1}{a}\ln(u_i)$$

$$\sum_{i=1}^{n} \ln(u_i) \ge -a > \sum_{i=1}^{n+1} \ln(u_i)$$

$$\ln\left(\prod_{i=1}^{n} u_i\right) \ge -a > \ln\left(\prod_{i=1}^{n+1} u_i\right)$$

$$\prod_{i=1}^{n} u_i \ge e^{-a} > \prod_{i=1}^{n+1} u_i$$

Poisson distribution

We have the following procedure:

1. Initialization: $n=0$, $q=1$
2. Generate $u_{n+1}=U(0,1)$
3. $q=qu_{n+1}$
4. If $q < e^{-a}$

   return n,

   otherwise

   $n=n+1$ and go back to step 2

Poisson distribution

- The average number of instances of a uniform r.v. needed to generate one instance of *X* is

$$N=\sum_{n=0}^{\infty}(n+1)p(n)=\sum_{n=0}^{\infty}(n+1)\frac{e^{-a}a^{n}}{n!}=$$

$$=\sum_{n=0}^{\infty}n\frac{e^{-a}a^{n}}{n!}+\sum_{n=0}^{\infty}\frac{e^{-a}a^{n}}{n!}=\sum_{n=1}^{\infty}\frac{e^{-a}a^{n}}{(n-1)!}+1=$$

$$=a\sum_{n=0}^{\infty}\frac{e^{-a}a^{n}}{n!}+1=a+1$$

Binomial distribution

- We want to generate instances of a r.v. X with binomial (p,n) distribution

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0,1,2,\cdots,n$$

- X represents the number of successes among n independent Bernoulli experiments with success probability p

Binomial distribution

Generation method 1 (convolution)

1. Generate n instances u_i di U(0,1)
2. Count the number x of variables u_i that are smaller than p
3. Return x

Binomial distribution

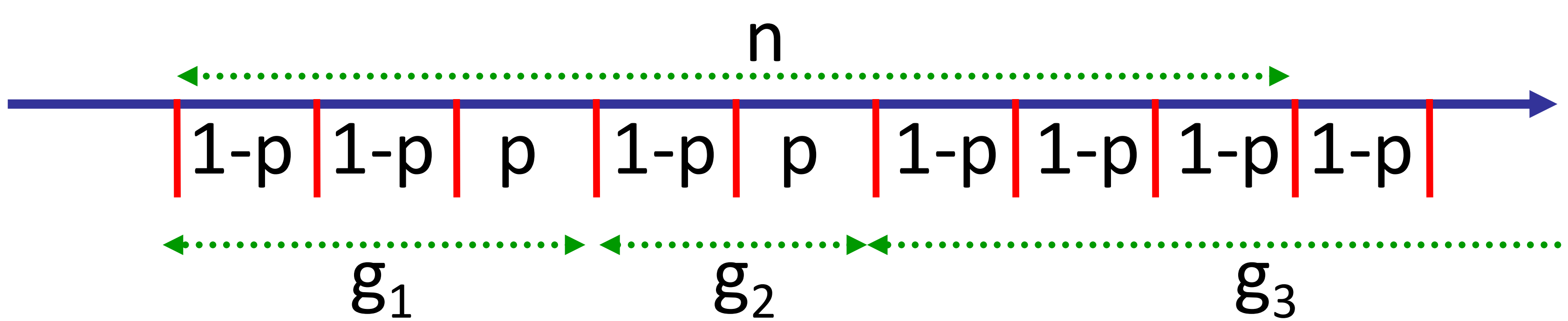
Generation method 2 (inverse-transform)

1. Calculate F(x) and store its values in vector A
2. Generate u=U(0,1)
3. Lookup x such that $A[x] \leq u < A[x+1]$
4. Return x

Binomial distribution

Generation method 3

- It is based on the observation that a geometric with parameter *(1-p)* assuming the value *i+1* corresponds to *i* failed Bernoulli experiments



$g_1+g_2<n$ and $g_1+g_2+g_3>n$ **->** return 2

Binomial distribution

Generation method 3

1. Initialization: $m=1$, $q=0$
2. Generate $u_m=U(0,1)$
3. Generate the geometric $g_m=\text{ceil}(\ln(u_m)/\ln(1-p))$
4. Calculate $q=q+g_m$
5. If $q > n$

    return $m-1$,

   otherwise

    $m=m+1$ and go back to step 2

Wrap-up

- Good generation of instances of random variables are fundamental for simulation
  - Must be efficient
  - Accurate
- Some (a few) general methods
  - Inverse-transform
  - Acceptance-rejection
- and several specific methods