# Generování pseudonáhodných čísel

Literatura:


Knuth, D., E.: The Art of Computer Programming (third edition), Addison-Wesley 1998,

L´Ecuyer , P.: Random Number Generation (in Handbook of Simulation edited by Jerry Banks) John Wiley & Sons, inc. 1998,

Fishman, G., F.: Discrete-Event Simulation, Modeling, Programing and Analysis, Springer 2001,

L' Ecuyer, P.: Uniform Random Number Generators: a Review, Proceedings of the 1997 Winter Simulation Conference,

L' Ecuyer, P.: Uniform Random Number Generators, Proceedings of the 1998 Winter Simulation Conference,

Makoto Matsumoto, Takuji Nishimura: Mersene Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Generator, ACM Transactions on Modeling and Computer Simulations ( Special Issue on Uniform Random Number Generation), 1998

# Generování pseudonáhodných čísel

**náhodná posloupnost :** „ ….každý člen je nepředvídatelný“

požadované ideální vlastnosti:

- rovnoměrné (stejnoměrné) rozložení posloupnosti { $x_0$, $x_1$,……} v intervalu < 0,m >, případně < 0,1> , ( IID…independent and identically distributed ).
- platí: vektory $x_{n,k}$ = ( $x_n$, ……, $x_{n+k-1}$) jsou rovnoměrně rozloženy pro všechna **n** a **k** v k-dimenzionální hyperkrychli $[ 0, m ]^k$ , případně $[ 0, 1 ]^k$ ,

Možné zdroje:

**1) fyzikální zařízení** pro registraci fyzikálních procesů

- sledování radioaktivního rozpadu (n čitačů mod 2),
- šum elektronických prvků.

**2) tabulky náhodných čísel** - problém naplnění

**3) rekurentní algoritmy** : deterministický výpočet

definice (dle L’Ecuyer):

PRNG = ( S, $s_0$, T, X, G )… ……..pseudo-random generator

S……...množina vnitřních stavů,

$s_0$……..počátešní stav (seed),

T: S - > S…….přechodová funkce,

X…….. množina výstupních hodnot,

G.: S -> X…… výstupní funkce .

- existují i kombinace fyzikálních zařízení a rekurentních algoritmů: náhodná volba počátečního stavu $s_0$ , případně dalších parametrů PSRG .

# Vlastnosti rekurentního generování :

- 1. metoda z r. 1946: „metoda prostředních řádů 2. mocniny“ (autor: von Neumann ),
- rychlý výpočet, minimální nároky na paměť,
- nutné zvolit počáteční stav ( seed, sémě)
- reprodukovatelnost generování (hlavní výhoda),
- konečná délka slova (β bitů) => konečný počet zobrazitelných hodnot,
- deterministický algoritmus => konečná perioda :
  - perioda: nejmenší celé číslo p pro které platí, že $s_{n+p} = s_n$ pro všechna n, $s_n$…..vnitřní stav generátoru,
  - snaha o dosažení co největší periody , jednotlivé rekurentní metody se liší velikostí dosažitelné maximální periody pmax,
  - u každé metody je třeba znát všechny okolnosti, které jsou nutné pro dosažení maximální možné periody, tj. p = pmax; tyto okolnosti jsou výsledkem dlouhodobé a důkladné matematické analýzy,
- pro výpočet je nejčastěji použita celočíselná aritmetika ,
- pomocí výstupní funkce $G = s_i / (max+1)$, kde max je největší hodnota v generované posloupnosti; takto lze získat rovnoměrně rozložená reálná čísla $u_i$ z intervalu < 0, 1 ) , případně ( 0, 1 ):
- kvalita pseudonáhodných posloupností: :
  - chybí statistická nezávislost => rozložení IID (viz dříve) nelze rekurentní metodou dosáhnout,
  - nutné posuzovat dle „měkčích kriterií“.

# Kritérium pro měření kvality rozložení

$x_n$, $x_{n+1}$ , ……, $x_{n+k-1}$ ,…… …testovaná posloupnost s periodou P,
trunc $_v$ ($x_n$) ……hodnota vytvořená z v nejvýznamějších bitů $x_i$

- uvažujme posloupnost vektorů ( překrývají se k-tice ):
  $V_n$ = (trunc $_v$ ($x_n$), trunc $_v$ ($x_{n+1}$),…,trunc $_v$ ($x_{n+k-1}$) ) , n = 1,2,…, P
- posloupnost { $x_0$, $x_1$,……} je k distribuovaná s přesností na v bitů pokud splňuje následující skutečnosti:
  1. Každá z $2^{kv}$ možných nenulových hodnot vektoru $V_n$ se vyskytuje v periodě P ve stejném počtu.
  2. Hodnota k(v) je dána maximální délkou vektoru $V_n$ , pro níž ještě platí tvrzení z bodu 1, příčemž délka složek vektoru je v.

Geometrická interpretace:

předpoklady: délka generovaných čísel = β bitů, perioda = | S| = $2^e$ ,

1. Jednotlivé k-tice výstupních hodnot po této transformaci reprezentují body v k-rozměrné hyperkrychli [ 0, m] $^k$, celkový počet bodů = $2^e$ .
2. Každou osu hyperkrychle rozdělíme na $2^v$ stejných úseků; tím dostaneme $2^{kv}$ dílčích hyperkrychlí, jejichž všechny vrcholy jsou pokryty některým vektorem.
3. Při zvětšování hodnoty k vzrůstá počet vrcholů dílčích hyperkrychlí a od jisté hodnoty některé z jejich vrcholů nejsou pokryty.
4. optimální rozložení (zatím nedosaženo): k (v) = e / v, (tj. $2^{kv} = 2^e$ ) pak celá perioda právě pokrývá všechny vnitřní vrcholy (**asymptotically random, maximally distributed**)

# Typy transformačních funkcí T:

Příklady lineárních funkcí **:** :

a**) generátor typu MLG** (Multiple Rekursive Generator):

$x_{n+1} = ( a_0 x_n + a_1 x_{n-1} + \ldots\ldots\ldots a_{k-1} x_{n-k-1} ) \bmod m$ …formule řádu k

- vnitřní stav: $s_n = ( x_n , x_{n-1} , \ldots\ldots x_{n-k-1} )$,
- vhodné pro softvérové implementace,

b) **LFSR (Linear Feedback Shift Register), Tausworthe generator**

$x_{n+1} = ( a_0 x_n + a_1 x_{n-1} + \ldots\ldots\ldots a_{k-1} x_{n-k-1} ) \bmod 2$, $a_i$, $x_i \varepsilon \{0,1\}$,

- pro n-bitový register lze dosáhnout plné periody $p = 2^{n-1}$ ,
- vhodné pro hardvérové implementace.

Příklady použití nelineárních funkcí

a) přechodová funkce T je nelineární:

$x_{n+1} = ( a_0 x_n^2 + a_1 x_n + a_2) \bmod m$ , $s_n = ( x_n )$,

- **Quadratic Congruential Generator**

b) aplikace nelineární funkce G na výstup u generátoru s lineární přechodovou funkcí T ( např. typu MLG):

$x_n = ( x'_{n+1} + x'^{-1}_n ) \bmod m$ ; $x'_{n+1}$ a $x'_n$ jsou výstupy z MLG

- **Inversive Congruential Generator**

**Poznámka**: dále se omezíme na přehled nejvíce propracovaných a nejčastěji používaných lineárních metod určených pro softvérové implementace

# Smíšená kongruenční metoda

- známá od. r. 1948 ( Lehmer )
- **LCG….Linear Congruential Generator**

$$x_{n+1} = (a \cdot x_n + c) \bmod m$$

$x_0$…….startovací hodnota (sémě)

$x_0 \neq 0$, $x_0 < m$, $a \neq 0$, $a < m$, $c \neq 0$, $c < m$,

př.: a = 7, c = 7, m = 12

pro $x_0 = 3$ dostaneme: 3, 4, 11, 0, 7, 8, 3, 4,…..

pro $x_0 = 7$ dostaneme: 7, 8, 3, 4, 11, 0, 7 , 8,…...

př.: a = c = 1, m = 12

pro $x_0 = 3$ dostaneme: 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, 2, 3,….

plná perioda, ale není náhodná

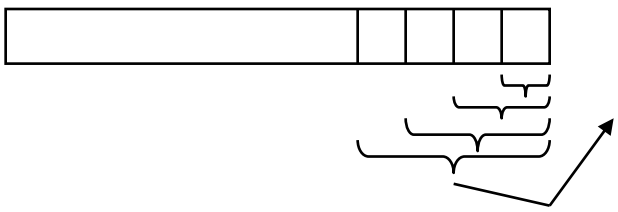
Volba parametrů :

**a) volba modulu m**… ( omezuje periodu shora ),

1) **m = $2^{\beta}$**, β…..počet informačních bitů

výhoda: snadný výpočet operace modulo,

nevýhoda: nejnižší bity výsledku jsou „málo náhodné“



*zase kongruenční posloupnosti s periodou 2, resp. 4, resp. 8, resp. 16, atd*

2) **m = $2^{\beta}$ + 1**, nebo **m = $2^{\beta}$ - 1**

– výše zmíněný jev nastává v daleko menší míře

3) **m = prvočíslo** …zmíněný jev nenastává,

# Smíšená kongruenční metoda

**b) volba parametrů a, c,**

Obecné podmínky plné periody:

1) c, m ……..navzájem nesoudělná čísla
2) ( a = 1 ) mod q, jestliže q je prvočinitelem modulu m
   tj. a -1 = k . q, k = celá část podílu a / q
3) ( a = 1 ) mod 4 , pokud 4 dělí m

Příklad.: m = 9, c = 7, => $x_{n+1}$ = ( a . $x_n$ + 7 ) mod 9

q = 3 => a - 1 = k . 3 => a = 1 + k . 3; ( k = 0, 1,….)

možné hodnoty parametru a: 1, 4, 7, 10, 13, 16, ….

volme $x_0$ = 2, pak:

pro a = 1: 2, 0, 7, 5, 3, 1, 8, 6, 4, 2, …..

pro a = 4: 2, 6, 4, 5, 0, 7, 8, 3, 1, 2,…..

pro a = 3: 2, 4, 1, 1, 1, …

Podmínky plné periody pro $m = 2^{\beta}$:

1) **c …..liché číslo**
2) **a = 1 + 4 . k**, k = 0, 1, 2, ……

Další doporučení pro volby a:

$a = z^r + 1$, z…..základ soustavy

výhoda: eliminace násobení (lze nahradit posuvem a přičtením )

$a \approx \sqrt{m}$ … výhodné z hlediska „kvality posloupnosti“
(viz. testování)

# Multiplikativní kongruenční metoda

- **MLCG ( Multiplicative LCG ),**

  $\mathbf{x_{n+1} = (a \,.\, x_n) \bmod m,}$ také : $x_n = (a^n \,.\, x_0) \bmod m$
- speciální případ MLG pro k = 1, $a = a_0$ ,
- nelze dosáhnout plné periody ( $x_n \neq 0$ ),

Podmínky pro dosažení maximální periody pmax:

- v obecné podobě složité ( viz Knuth, L´Ecuyer) ,
- případ $m = 2^{\beta}$: $pmax = 2^{\beta-2} = 1/4\, m$ ; pro $\beta \geq 4$
  - polovina lichých čísel z intervalu < 1, m-1 >,
  - volba $x_0$ : liché číslo
  - volba a: a = (3 nebo 5 ) mod8 ,

    tj. a = 3 + 8.k nebo a = 5 + 8. k, k = 0,1,2,..
  - perioda ν nejméně významných bitů = $2^{\nu-2}$
- případ m je mocnina prvočísla p :

  $\mathbf{m = p^{\beta}}$ : $pmax = p^{\beta-1}(p-1)$; pro $\beta = 1$ : pmax = m - 1
  - velmi často používaný generátor,
  - hledáme a tak, aby $x_n = (a^n \,.\, x_0) \bmod m = x_0$ pro n = m-1, tedy $a^{m-1} - 1 = 0 \pmod m$;
    toto splňují primitivní kořeny modulu m , jejich hodnoty pro zvolené m jsou publikovány,
  - případ m = 11: $(a^{10} - 1) \bmod 11 = 0$ platí pro a = 2, 6, 7, 8,
  - případ $m = 2^{31-1}$: existuje 534 600 000 primitivních kořenů, nejvhodnější: 742938285, 950706376, 1226874159, 62089911

  $\mathbf{m = p_1^{\beta 1} \ldots p_i^{\beta \iota}}$ :

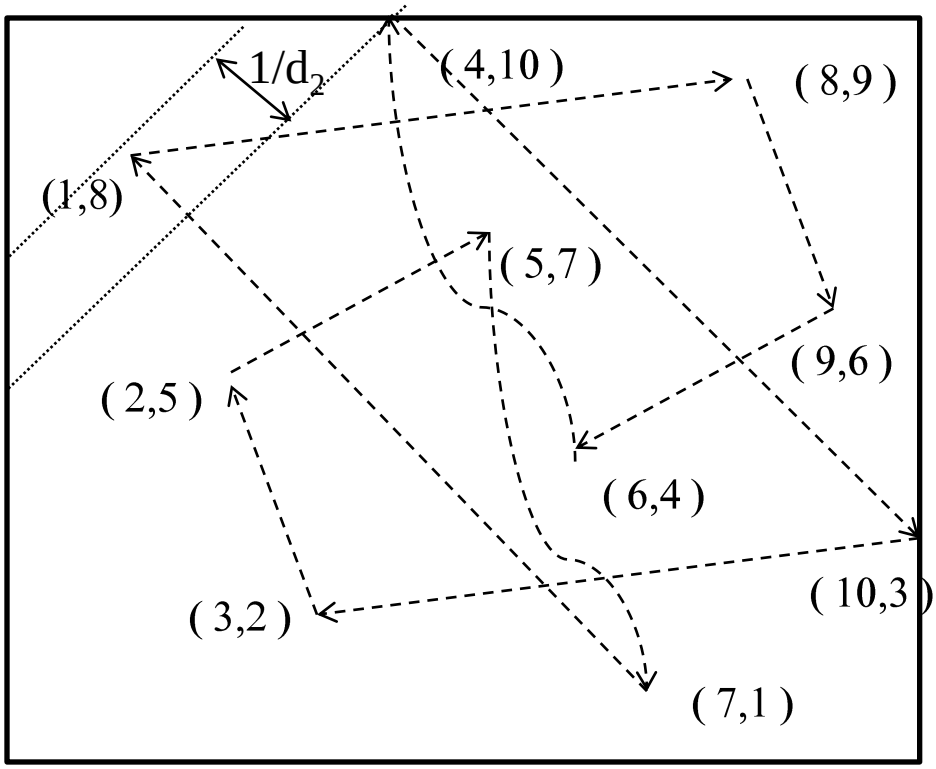
  $pmax = nsn(pmax(p_1^{\beta 1}), \ldots, pmax(p_i^{\beta \iota}))$

# Multiplikativní metoda – kvalita rozložení

Příklad: generátor MLCG: $m = 11$, $x_0 = 1$

$a = 8$ => perioda: 1, 8, 9, 6, 4, 10, 3, 2, 5, 7

Rozložení dvojic ( k=2):




všechny body leží na rovnoběžkách:

max. vzdálenost $= 1 / d_2$ => $d_2$ = dvoudimenziální přesnost generátoru,

$a = 2$ perioda: 1, 2 ,4, 8, 5, 10, 9, 7, 3, 6 , horší rozložení ($1/d_2$ je větší )

Spekrální test [Knuth] :

- kriterium kvality: vzdálenost $1/ d_2$, příp. počet rovnoběžek

  k = 2 ……10 obsazených bodů z 100,

  k = 3 ……10 obsazených bodů z 1 000,

  k = 4 ……10 obsazených bodů z 10 000, atd.

  k > 2 …. body leží v hyperrovinách, max. vzdálenost $= 1 / d^k$ ,

  $d_k$……k-dimenziální přesnost klesá s rostoucí hodnotou k

  ( $d_k \leq \sqrt[k]{m}$ )

# Kombinované generátory

- **CLCG ( combined MLCG ),**
- založeny na kombinaci hodnot několika nezávislých MLCG

Příklad: kombinace čtyř generátorů typu MLCG s prvočíselnými moduly ( viz. Liter.: Fishman)

$\mathbf{x_{j,n+1} = a_j \cdot x_{j,n} \pmod{m_j}}$ **……j-tý generátor**

$m_1 = 2^{31} - 1 = 2147483647$, $a_1 = 45991$,

– $pmax = m_1 - 1 \approx 2.14 \times 10^9$

$m_2 = 2^{31} - 105 = 2147483543$, $a_2 = 207707$ ,

– $pmax = m_2 - 1$

$m_3 = 2^{31} - 225 = 2147483423$, $a_3 = 138556$,

– $pmax = m_3 - 1$

$m_4 = 2^{31} - 325 = 2147483323$, $a_4 = 49689$,

– $pmax = m_4 - 1$

výsledná posloupnost:

$\mathbf{y_{n+1} = ( \delta_0 x_{0,n+1} + \delta_1 x_{1,n+1} + \delta_2 x_{2,n+1} + \delta_3 x_{3,n+1} ) \pmod{m_1}}$,
$\delta_j$ , $m_j$ ( j =1,2,3,4)….navzájem nesoudělná čísla

dosažitelná perioda :

$\text{lcm}[(m_1 - 1), (m_2 - 1), (m_3 - 1), (m_4 - 1)] \approx 2.13 \times 10^{37}$

# Kongruenční generátory s vyžitím přenosu

- **MWC (multiply with carry)**

  $x_{n+1} = ( a_1 x_n + a_2 x_{n-1} + \ldots\ldots\ldots a_k x_{n-k} + c_n ) \bmod m,$

  $c_{n+1} = ( a_1 x_n + a_2 x_{n-1} + \ldots\ldots\ldots a_k x_{n-k} ) \text{ div } m ,$

  - div …..celočíselné dělení,
  - počáteční stav $s_0 = ( x_n , x_{n-1} , \ldots\ldots, x_{n-k-1} , 1 )$,
  - lze dosáhnout velmi dlouhé periody: vlastnosti přibližně stejné jako generátor typu LCG s následujícími parametry:

    $x_{n+1} = a x_n \pmod M$, kde

    $M = a_1 m^1 + a_2 m^2 + \ldots\ldots\ldots\ldots + a_{k-} m^k - 1,$

    $a = 1 / m \pmod M$

- speciální případy MWC (autoří Marsaglia, Zaman 1991):
  - **AWC ( add with carry):**

    obecně: $x_{n+1} = ( x_{n-r} + x_{n-s} + c_n ) \bmod m, \quad a_r = 1, a_s = 1$

    $c_{n+1} = 1$ pokud $x_{n-r} + x_{n-s} + c_n \geq m$ ,

    $c_{n+1} = 0$, pokud $x_{n-r} + x_{n-s} + c_n < m$ ,
  - jednotlivé metody se liší indexy r, s,
  - <u>příklad:</u> r = 1, s = 2

    $x_{n+1} = ( x_{n-1} + x_{n-2} + c_n ) \bmod 2^{31},$

    přibližná délka periody: $2^{86}$

  - **SBC (subtract with carry):**
  - <u>příklad:</u> $x_{n+1} = ( x_{n-7} - x_{n-9} - c_n ) \bmod 2^{31-5},$

    přibližná délka periody: $2^{307}$

# Další varianty kombinovaných generátorů

Míchající generátory **(** vznikly v laboratořích firmy Boeing)

- snaha o odstranění závislosti po sobě jdoucích čísel generovaných jedním generátorem => posloupnost generovaná jedním generátorem je promíchávána pomocí druhého generátoru; v mnoha případech lze dosáhnout periody, která je nejmenším společným násobkem dílčích period,

Principy míchajících generátorů**:**

**a) použití indexů:**

- 1. generátor vygeneruje tabulku čítající k hodnot,
- výstup druhého generátoru je transformován na index směřující do zmíněné tabulky,
- výstupem míchajícího generátoru je hodnota z tabulky určená indexem; tato hodnota je nahrazena další hodnotou z 1. generátoru.

**b) použití rotací:**

- z každého čísla generovaného 1. generátorem se vyjme skupina r bitů; takto získaná hodnota se interpretuje jako číslo r bez znaménka,
- výstupem míchajícího generátoru je hodnota generovaná 2. generátorem, která je cyklicky rotována o r bitů doleva .

# Metoda Mersenne Twister

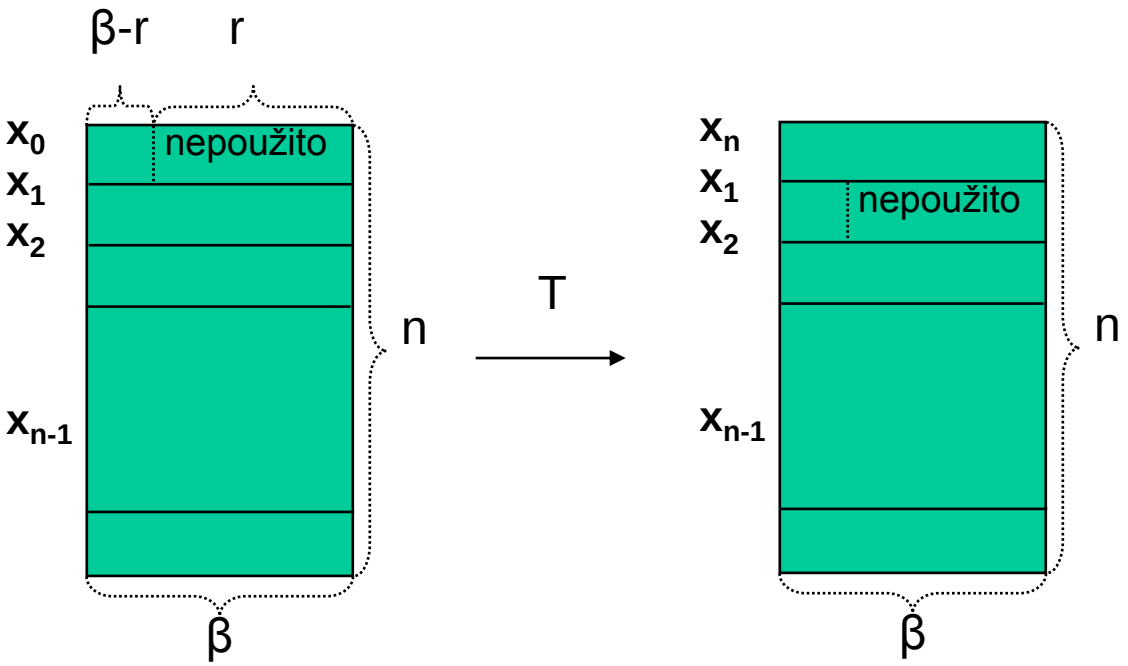
- autoři: M. Matsumoto, T. Nishimura (Keio University), 1998
- výstup: sekvence 32 bitových čísel typu integer,
- přechodová funkce T: lineární transformace tzv. neúplných polí

  $( x_0, x_1, x_2 \ldots\ldots x_{n-2}\ x_{n-1} ) \Rightarrow ( x_n, x_1, x_2 \ldots\ldots x_{n-2}\ x_{n-1} )$,

  $( x_n, x_1, x_2 \ldots\ldots x_{n-2}\ x_{n-1} ) \Rightarrow ( x_n, x_{n+1}, x_2 \ldots\ldots x_{n-2}\ x_{n-1} )$,

  ……

  sémě: $( x_0, x_1, x_2 \ldots\ldots x_{n-2}\ x_{n-1} )$….získat pomocí jiného generátoru




- dosažená perioda $p = 2^{n.\beta - r} - 1 = 2^{19937} - 1$ , n = 624, β =32, r =31
- produkuje velmi kvalitní posloupnost ( viz spektrální test):
  - např. s přesností 32 bitů je rovnoměrně distribuovaná v 623 dimensionálním prostoru,
  - s přesností na 6 bitů je rovnoměrně distribuovaná v 2492 dimensionálním prostoru,
- paměťové požadavky: 624 32 bitových slov

# Algoritmus metody Mersenne Twister

- generování $x_n$, resp. $x_{n+1}$, r esp.$x_{n+2}$,….dle následujících vzorců ( pro k = 0, resp. 1, resp. 2,… ):

1) $\mathbf{q} = (\mathbf{x_k}^{\mathbf{u}} \;\&\; \mathbf{x_{k+1}}^{\mathbf{l}})$, k = 0,1,2,……

$\mathbf{x_k}^{\mathbf{u}}$ ……β-r horních bitů čísla $x_k$

$\mathbf{x_{k+1}}^{\mathbf{l}}$ ……..r spodních bitů čísla $x_{k+1}$

& ……operátor zřetězení ,

**2)** $\mathbf{x_{k+n}} = (\mathbf{x_{k+m}} \oplus \mathbf{q}\,.\,\mathbf{A})$ m = **397,**

**(** např. pro k = 0: $\mathbf{x_n} = (\mathbf{x_m} \oplus \mathbf{q}\,.\,\mathbf{A})$ **)**

$$A = \begin{pmatrix} 0 & 1 & 0 & . & 0 \\ 0 & 0 & 1 & . & 0 \\ . & . & . & . & . \\ 0 & 0 & 0 & . & 1 \\ a_{\beta-1} & a_{\beta-2} & a_{\beta-3} & . & a_0 \end{pmatrix}$$

nechť $\mathbf{q} = (q_{31}, q_{30}, q_{29}, \ldots\ldots\ldots\ldots, q_0)$, pak

$\mathbf{q}\,.\,\mathbf{A} = (\mathbf{q} >> 1)$ je-li $q_0 = 0$,

$(\mathbf{q} >> 1) \oplus \mathbf{a}$ je-li $q_0 = 1$, kde

$\oplus$ …… součet modulo 2 (bitwise XOR)

$\mathbf{a} = (99\ 08\ B0\ DF)_{16}$

# Algoritmus metody Mersenne Twister

3) úprava výstupního slova ( pro zlepšení statistických vlastností generované sekvence):

$\mathbf{y} = \mathbf{x}_{k+n}$
$\mathbf{y} = \mathbf{y} \oplus (\mathbf{y} >> u)$
$\mathbf{y} = \mathbf{y} \oplus ((\mathbf{y} << s) \text{ AND } \mathbf{b})$
$\mathbf{y} = \mathbf{y} \oplus ((\mathbf{y} << t) \text{ AND } \mathbf{c})$
$\mathbf{y} = \mathbf{y} \oplus (\mathbf{y} >> l)$
**výstup y**

- hodnoty parametrů: u = 11 , s = 7, t = 15, l =18
  **b** = ( 9D 2C 56 80 )$_{16}$
  **c** = ( EF C6 00 00 )$_{16}$

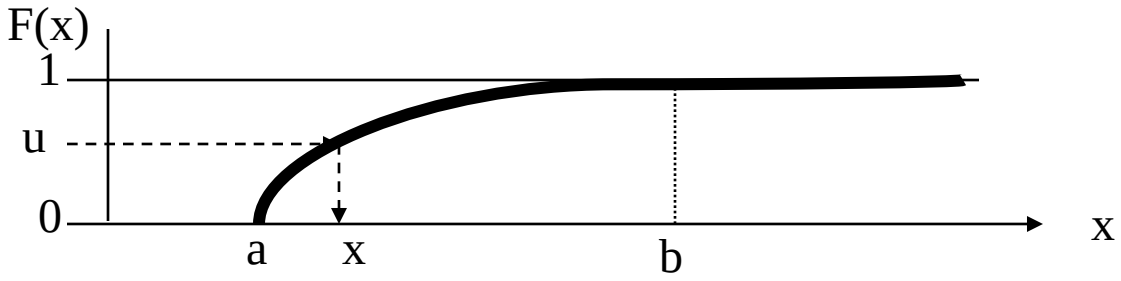
Poznámky:

- výše zmíněná varianta metody je označena jako MT19937 a representuje patrně nejdokonalejší existující generátor (nejdelší perioda a velmi dobré statistické vlastnosti),

- existují i jiné varianty metody Mersenne Twister ( MT11213A, MT11213B ) - mají jiné hodnoty parametrů

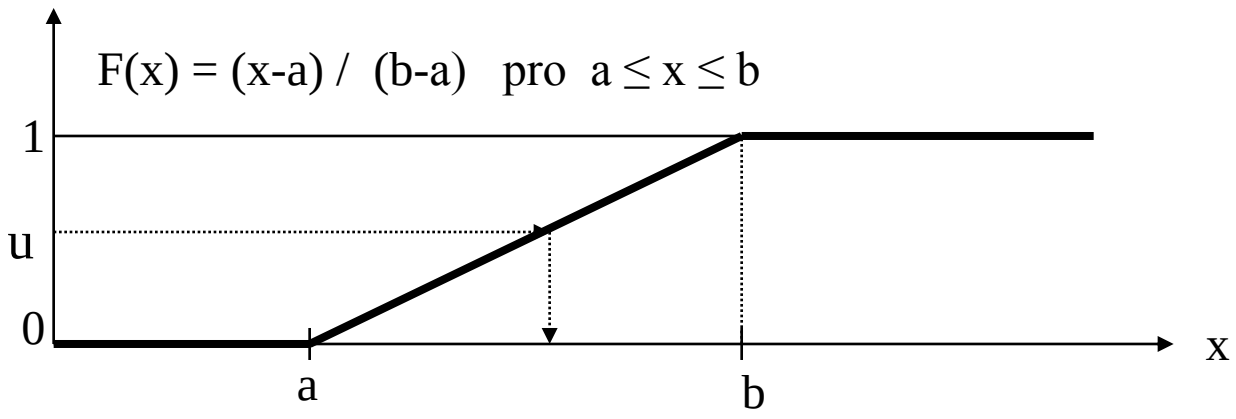
# Inversní metoda

- obecná metoda pro transformaci rozložení spojitých i diskrétních náhodných veličin,
- použitelná při znalosti inverzní funkce k distribuční funkci požadovaného rozdělení,
- princip:




F(x)…distribuční funkce požadovaného rozložení ,
F(x) = u, u ε U (0, 1) => **x = F $^{-1}$ (u),**

- generování čísel s rovnoměrným rozložením na interval (a,b):



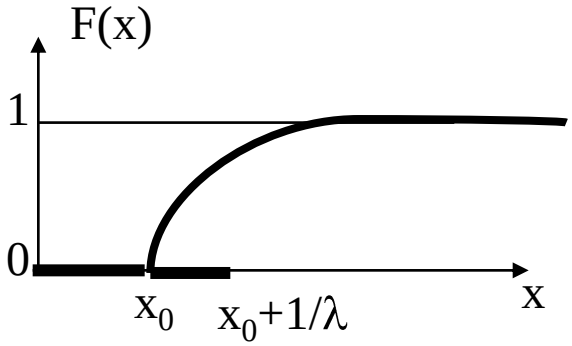

F (x ) = u => ( x - a ) / ( b-a ) = u => **x = a + u ( b - a )**
E(X) = ( a + b) / 2 ; D ( X) = ( b - a)$^2$ / 12

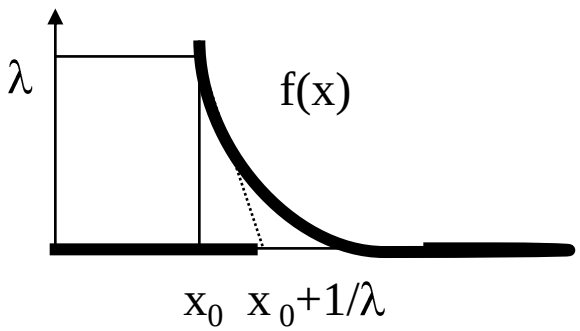
# Inversní metoda

- aplikace inverzní metody na exponenciální rozložení (s posunutím o $x_0$)

$$F(x) = 1 - e^{-\lambda(x-x_0)}; \quad f(x) = \lambda \cdot e^{-\lambda(x-x_0)}; x \geq x_0$$

$$E(x) = x_0 + \frac{1}{\lambda}; \; D(x) = \frac{1}{\lambda^2}$$




$$u = F(x) = 1 - e^{-\lambda(x-x_0)} \Rightarrow -\lambda(x - x_0) = \ln(1-u) \Rightarrow$$

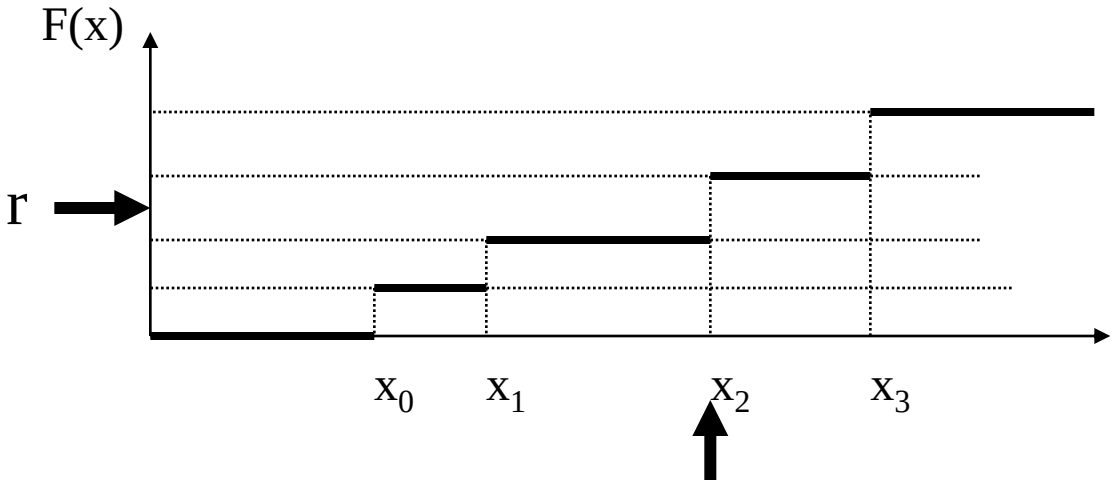
$$x = x_0 - \frac{1}{\lambda}\ln(1-u), \; u \in U(0,1) \Rightarrow (1-u) \in U(0,1) \Rightarrow$$

$$\underline{x = x_0 - \frac{1}{\lambda}\ln u}$$

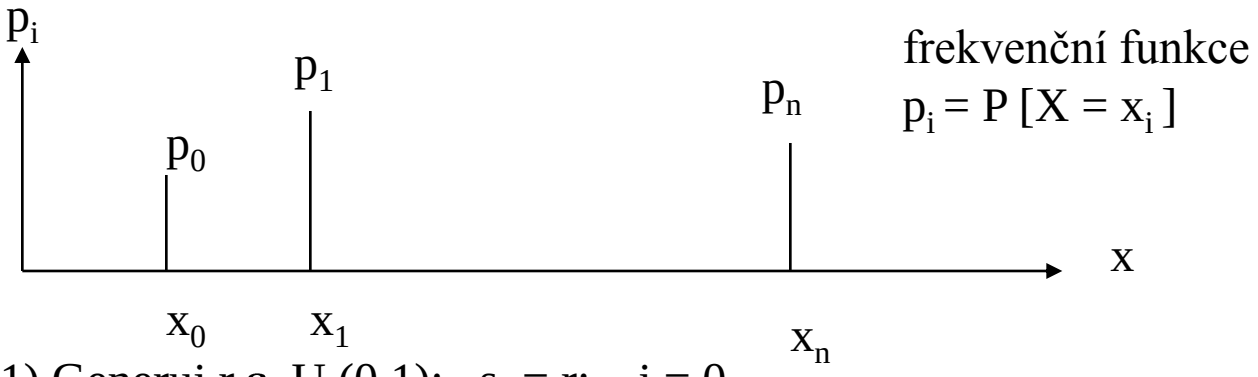
- normované exponenciální rozložení E (0, 1 ): případ $x_0 = 0, \lambda = 1$
- pro x´ ε E ( 0, 1 ): transformace $\underline{x' = -\ln u}$,
- pro x ε E ( 0, 1/λ ) : transformace $\underline{x = (1/\lambda)\, x'}$,
- pro x ε E ( $x_0$, 1/λ ) : transformace $\underline{x = x_0 + (1/\lambda)\, x'}$

# Inversní metoda

- aplikace pro diskrétní náhodné veličiny:,
- analogie inversní metody pro spojité náhodné




Algoritmus vycházející z frekvenční funkce:




1) Generuj r ε U (0,1); s = r; i = 0.

2) Polož s = s - $p_i$.

3) Pokud s > 0 , polož i = i + 1 a jdi na bod 2, jinak jdi na bod 4).

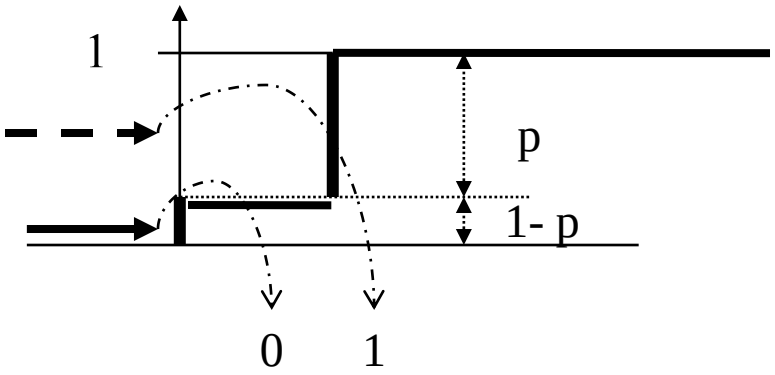
4) Jako výsledek vezmi hodnotu $x_i$.

$$\underline{P[X = x_i]} = P(\sum_{k=0}^{i-1} p_k < r < \sum_{k-0}^{i} p_k) = \sum_{k=0}^{i} p_k - \sum_{k=0}^{i-1} p_k = \underline{p_i}$$

# Inversní metoda

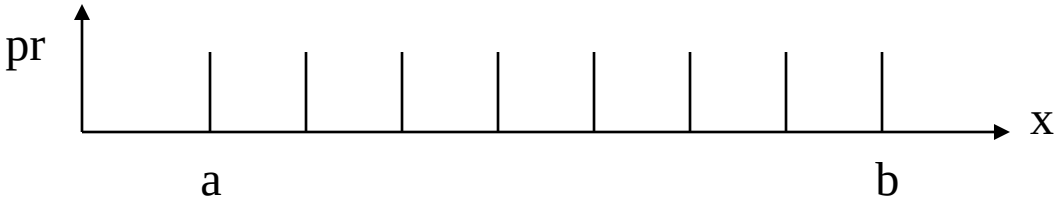
- aplikace na diskrétní **Bernoulliovo rozdělení** (nula -jedničkové, alternativní)

$$X = \begin{cases} 1 \ldots\ldots \text{s pravděpodobností } p \\ 0 \ldots\ldots \text{s pravděpodobností } 1 - p \end{cases}$$




$E(X) = p \; ; \quad D(X) = p\,(1 - p)$

- transformace rovnoměrného diskrétního rozložení na interval a, b:

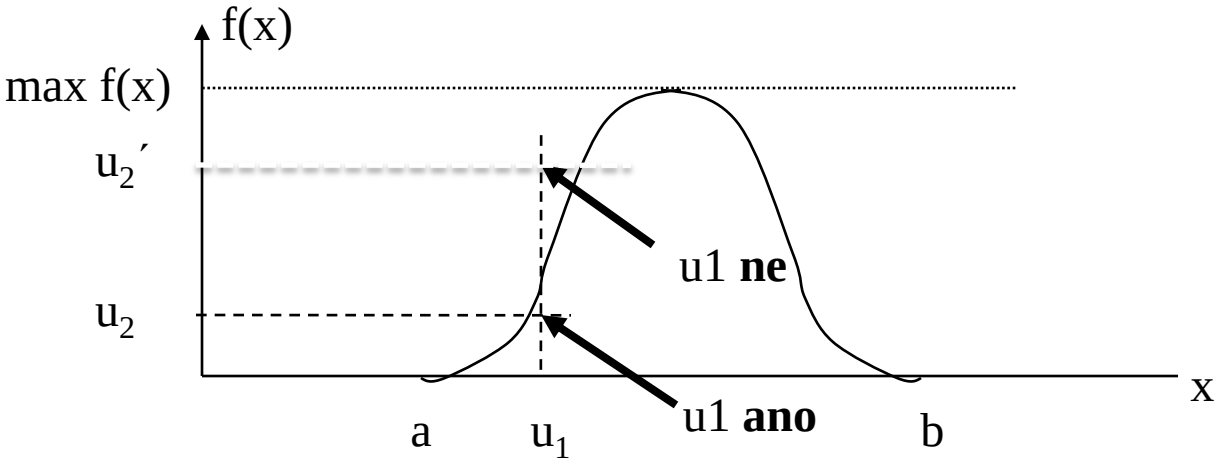



  - platí: $(b - a + 1).\, pr = 1 \Rightarrow pr = 1 / (b - a + 1)$
  - algoritmus:
    1) Generuj $u \in U(0,1)$.
    2) Polož $x = a + \lfloor (b - a + 1) \cdot u \rfloor$ (…celá část).

příklad: a = 1, b = 10: x = 1 + celá část ( 10 . u)

# Vylučovací metoda

- **též zamítací metoda: ( von Neumann ),**
- **princip:**




Algoritmus:

1) Generuj dvojici rovnoměrně rozložených čísel $u_1$ , $u_2$.: $u_1 \varepsilon$ U (a,b) a $u_2 \varepsilon$ ( 0, max (f(x)).
2) Pokud $u_2 > f(u_1)$, ignoruj obě čísla a pokračuj od body 1.
3) Pokud $u_2 \leq f(u_1)$, pak použij číslo $u_1$; číslo $u_2$ ignoruj.

příklad:

$u_1, u_2$ : $u_2 < f(u_1)$ => $u_1$, ~~$u_2$~~ ……výstup $u_1$

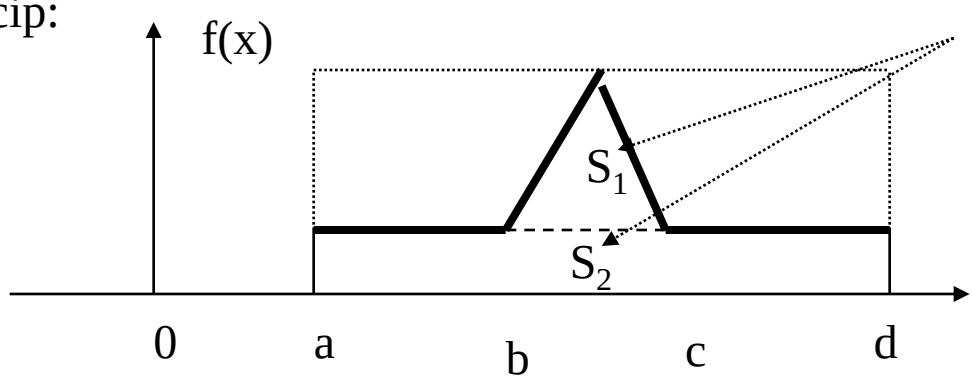
$u_1, u_2'$ : $u_2' < f(u_2)$ => ~~$u_1$~~, ~~$u_2'$~~ …..žádný výstup

Použití: např. Gaussovo (normální ) rozložení

- nelze použít inversní metoda
- uvažovaný obor hodnot normovaného rozložení :
  - pro malé serie: $(-3\sigma , +3\sigma)$
  - pro velké série $(-5\sigma , +5\sigma)$

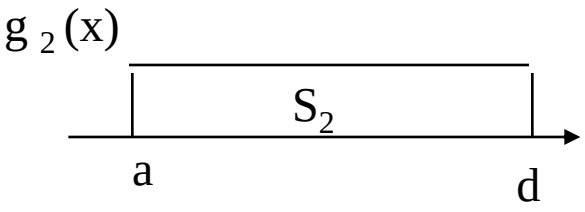
# Dekompoziční metoda

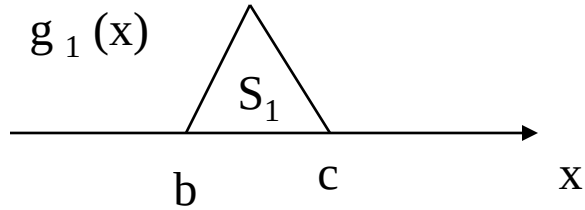
- aplikace pro spojité náhodné veličiny,
- rozkládá požadovanou hustotu pravděpodobnosti na dílčí hustoty $f_i$ ,
- princip:




plochy vyznačených oblastí

  - f(x).. hustota pravděpodobnosti => $S_1 + S_2 = S = 1$ => $S_1 / S + S_2 / S = 1$ => dekomposice: $f(x) = g_1(x) + g_2(x)$




  - pro i = 1, 2 platí: $\int_{-\infty}^{+\infty} g_i(x)dx = S_i \Rightarrow \int_{-\infty}^{+\infty} \frac{1}{S_i} g_i(x)dx = 1$
  - $f_1(x) = 1 / S_1 \; g_1(x)$……. hustota pravděpodobnosti
  - $f_2(x) = 1 / S_2 \; g_2(x)$……..hustota pravděpodobnosti
  - označíme: k…..počet dílčích hustot $f_i(x)$

<u>Postup</u>:

1) Generuj hodnotu i $\epsilon$ < 1,……,k > dle frekvenční funkce
   S = < $S_1, S_2, \ldots\ldots S_k$ > , kde $S_1 + S_2 + \ldots\ldots + S_k = 1$,
2) S použitím hustoty $f_i(x)$ generuj hodnotu x ( dle některé z předešlých metod) v intervalu na kterém je hustota definována.

# Speciální transformační metody

Polární metoda pro generování čísel s Gaussovým rozložením

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \sigma > 0, x \in (-\infty, +\infty)$$

$$F(x) = \frac{1}{2\pi\sigma} \int_{-\infty}^{x} e^{-\frac{(\xi-\mu)^2}{2\sigma^2}} d\xi$$

$$X' = \frac{X-\mu}{\sigma} \Rightarrow E(X') = 0, D(X) = \sigma^2 = 1$$

$$f(x') = \frac{1}{\sqrt{2\pi}} e - \frac{x^2}{2}; F(x') = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{\xi}{2}} d\xi$$

Postup generování:

1) Generuj $p_1$, $t_1 \varepsilon$ U (0,1)
2) Vypočti $u_1$, $v_1$ dle vztahů: $u_1 = 2\,p_1 - 1$, $v_1 = 2\,t_1 - 1$,
   ( $u_1$, $v_1$ $\varepsilon$ U (-1, +1)).
3) Vypočti $s_1 = u_1^2 + v_1^2$.
4) Pokud s $\geq 1$ pokračuj od bodu 1 , jinak přejdi na bod 5.
5) Vypočti

$$x_1 = u_1 \sqrt{\frac{-2\ln s_1}{s_1}}$$

$$y_1 = v \sqrt{\frac{-2\ln s_1}{s_1}}$$

Poznámka: $x_1$, $y_1$ ……dvojice hodnot náhodné veličiny s normálním rozložením ( důkaz: viz Knuth D.E.: The Art of Computer programming)

# Speciální transformační metody

Metoda pro generování čísel s Poissonovým rozložením:

- frekvenční funkce: $f(n) = \frac{e^{-\lambda}.\lambda^n}{n!}; n \geq 0$
- algoritmus generování: založen na základě vztahu k rozložení exponenciálnímu E (1/λ): 1/ λ...průměrná hodnota intervalu mezi nezávislými událostmi

Postup generování:

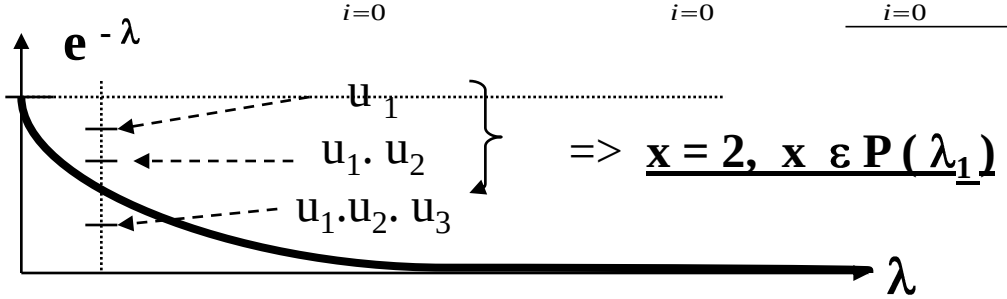
1) Polož i = 0, s = 0.
2) Generuj $v_i$, $v_i$ ε E (1 / λ ).
3) Vypočti $s = s + v_i$. Pokud s < 1 polož i = i + 1 a pokračuj od bodu 2.
4) Výsledek x = i.

Modifikace algoritmu pro přímé použití čísel $u_j$ ε U (0,1):

1) Polož i = 0, s =1.
2) Generuj $u_i$, $u_i$ ε U ( 0,1 ).
3) Vypočti $s = s.\ u_i$. Pokud $s \geq e^{-\lambda}$ polož i = i + 1 a pokračuj od bodu 2.
4) Výsledek x = i.

Poznámka: platí $v_j = -1/\lambda \ln u_j \Rightarrow -\sum_{i=0}^{x} \ln u_i \leq \lambda < -\sum_{i=0}^{x+1} \ln u_i \Rightarrow$

$$-\ln \prod_{i=0}^{x} u_i \leq \lambda < -\ln \prod_{i=0}^{x+1} u_i \Rightarrow \prod_{i=0}^{x} u_i \geq e^{-\lambda} > \prod_{i=0}^{x+1} u_i$$




=> **x = 2, x ε P ( $\lambda_1$ )**

$\lambda_1$.... hodnota konkrétního parametru

# Testování generátorů náhodných čísel

**účel**: prověřit, zda generovanou posloupnost lze považovat za náhodný výběr ze souboru, který má určité pravděpodobnostní rozložení

**Teoretické testy:**

- vychází z matematického rozboru,
- poskytujé hlubší pohled.
- příklad teoretického testu: sériový korelační test

Sériový korelační test:

- pomocí korelačního koeficientu C měří vztah mezi překrývajícími se dvojicemi čísel uvnitř posloupnosti délky n:

$$C = \frac{n(x_0x_1 + x_1x_2 + ....x_{n-2}x_{n-1} + x_{n-1}x_0) - (x_0 + x_1 + ...x_{n-1})^2}{n(x_0{}^2 + x_1{}^2 + ...x_{n-1}{}^2) - (x_0 + x_1 + .....x_{n-1})^2}$$

  -1 <= C <= + 1 , snaha o C -> 0

- liter. (Knuth ) : praktický postup pro výpočet C generátoru typu LCG s parametry m, a, c :

- $$C = \frac{1}{a}\left(1 - 6\frac{c}{m} + 6\left(\frac{c}{m}\right)^2\right) .....odhad\ s\ chybou < a/m)$$

  pro a -> 0: C -> ∞, ale odhad chyby ≤ a / m => doporučení:

$$a \cong \sqrt{m}, \qquad C \cong \frac{2}{\sqrt{m}}$$

$$\frac{c}{m} \cong \frac{1}{2} - \frac{1}{6}\sqrt{3}$$

**Empirické testy**:

- založeny na výběru určité testovací statistiky (naměřené na výstupu testovaného generátoru) a na jejím srovnání se statistikou, kterou bychom získali aplikací na skutečný na náhodný generátor.

# Spektrální test

- existuje v různých podobách,
- spojuje charakter teoretických i experimentálních testů,
- společný rys: sleduje výskyt překrývajících se dvojic, trojic, …, k-tic, které pak interpretuje jako body v dvojrozměrném, trojrozměrném, ……, k-rozměrném prostoru a vyhodnotí kvalitu rovnoměrnosti jejich rozložení ,
- kriterium měření rovnoměrnosti ( dle L' Ecuyer): **extrémní k-dimenzionální odchylka (discrepancy) testované posloupnosti N vektorů:**

$$D_N^{(k)} = \max_{\forall r \in R'} \left| \frac{V(r)}{V} - \frac{I(r)}{N} \right|$$

r…...podhyperkrychle jejíž strany jsou v každé ose vymezeny úsečkami $< \alpha \; \beta >$, kde $0 \leq \alpha < \beta \leq m$, m….maximální generovaná hodnota ,

R´ …množina všech možných podhyperkrychlí v hyperkrychli $< 0, m >^k$ ,

V…..objem hyperkrychle $< 0, m>^k$ ,

N……celkový počet vygenerovaných bodů testované posloupnosti,

I (r)…počet bodů ležících v podhyperkrychli r,

$V(r) = \Pi_{i=1}^{i=k} ( \beta i - \alpha_i )$……objem podhyperkrychle r.

- v případě **generátorů pseudonáhodných čísel** se tato hodnota zvětšuje pro rostoucí hodnotu k =:> zmenšování přesnosti,
- v případě **generátoru náhodných** čísel zůstává rozložení nezávislé na hodnotě k (viz. definice IID).

# Test dobré shody ( $\chi^2$ test)

Obecný postup:

1) získání kontrolovaného souboru hodnot
   - generování kontrolní serie n - hodnot
   - rozdělení vygenerovaných hodnot do k - disjunktních kategorií
   - nalezení počtu hodnot v jednotlivých kategoriích

2) získání srovnávacího (kontrolního) souboru hodnot

   možné zdroje:
   - distribuční funkce F (x) :
     - pro odhad počtu hodnot v kategorii definované intervalem (r, s> platí: $n \,.\, p_s = n \,.\, (F(s) - F(r))$
       - n…celkový počet hodnot testované serie,
       - $p_s$…pravděpodobnost hodnoty z intervalu (r, s>
     - požadavek: $n\, p_s > 5$

   - hustota pravděpodobnosti f(x):

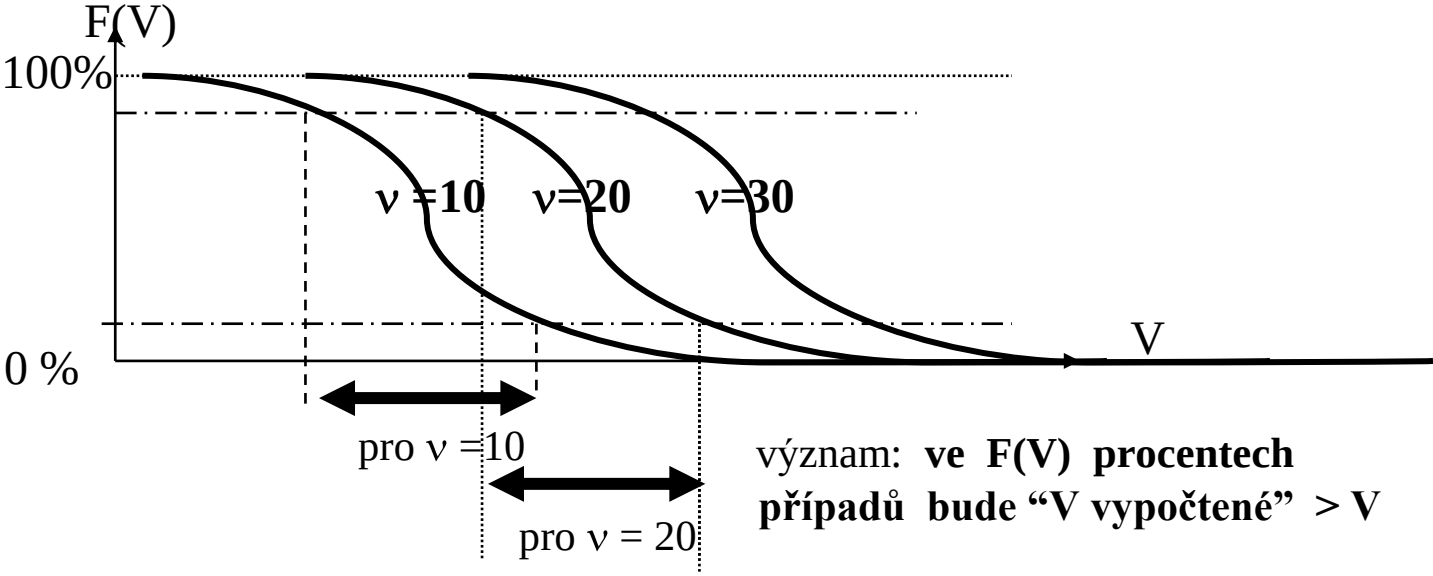
$$n.p_s = n.\int_r^s f(x)dx$$

   - frekvenční funkce p(x): $n \,.\, p_s = n \,.\, (\underbrace{p_k + p_{k+1} + \ldots\ldots + p_{k+q}}_{\text{požadované pravděpodobnosti jednotlivých hodnot}})$

# Test dobré shody ( $\chi^2$ test)

3) srovnání obou souborů
   - vyhodnocení výrazu $$V = \chi^2 = \sum_{1 \le s \le k} \frac{(Y_s - n \cdot p_s)^2}{n \cdot p_s}$$
     - $Y_s$…..počet hodnot v kategorii s (kontrolovaný soubor)
     - n . $p_s$ ….teoretický počet hodnot z kontrolního souboru

4) zhodnocení odchylky obou souborů
   - V…hodnota náhodné veličiny s rozložením $\chi^2$, která má $\nu$ stupňů volnosti, kde $\nu = k - 1$ ( k = počet kategorií )
   - zhodnocení na základě tabulky pro $\chi^2$ rozložen




Poznámky :

- test dobré shody je vhodný pro spojité i diskrétní náhodné veličiny
- doporučení pro volbu n: dosaáhnout hodnoty n . $p_s \geq 5$ pro každou kategorii

# Test dobré shody ( $\chi^2$ test) – příklad použití

Tabulky pro test dobré shody:

| $\nu$ | 99% | 95% | 75% | 50% | 25% | 5% | 1% |
|---|---|---|---|---|---|---|---|
| 1 | ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 10 | 2.55 | 3.94 | 6.74 | 9.34 | 12.55 | 18.3 | 23.31 |
| ... | ... | ... | ... | ... | ... | ... | ... |

Příklad: házení dvou kostek - sledujeme hodnotu součtu s pro n = 144 hodů

**výsledky:**

| $s$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $n \cdot p_s$ | 4 | 8 | 12 | 16 | 20 | 24 | 20 | 16 | 12 | 8 | 4 |
| $Y_s$ | 2 | 4 | 10 | 12 | 22 | 29 | 21 | 15 | 14 | 9 | 6 |

skutečné počty v kategorii s

teoretické počty v kategorii s: např.: 144 . 1/6.1/6 = 4,
144 . 2/6.1/6 = 8, atd.

$$V = \frac{(2-4)^2}{4} + \frac{(4-8)^2}{8} + \ldots\ldots\ldots\ldots + \frac{(6-4)^2}{4} = 7\frac{7}{48}$$

počet kategorií: k =11 => ν = k - 1 = 10

z tabulky vyplývá: 6.74 < 7 . 1458 < 9.34

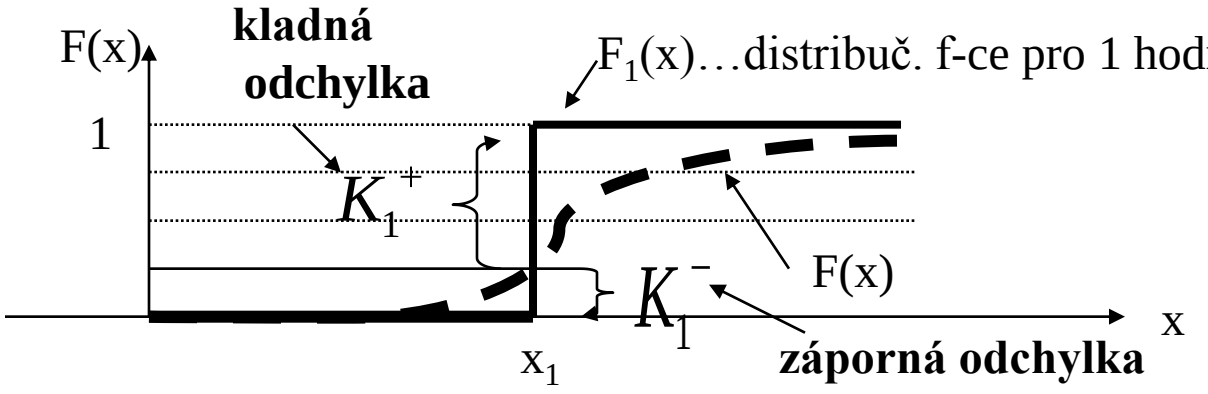
75% 50%

„dostatečně náhodný výběr“

# Kolmogorov - Smirnovův test

Princip:

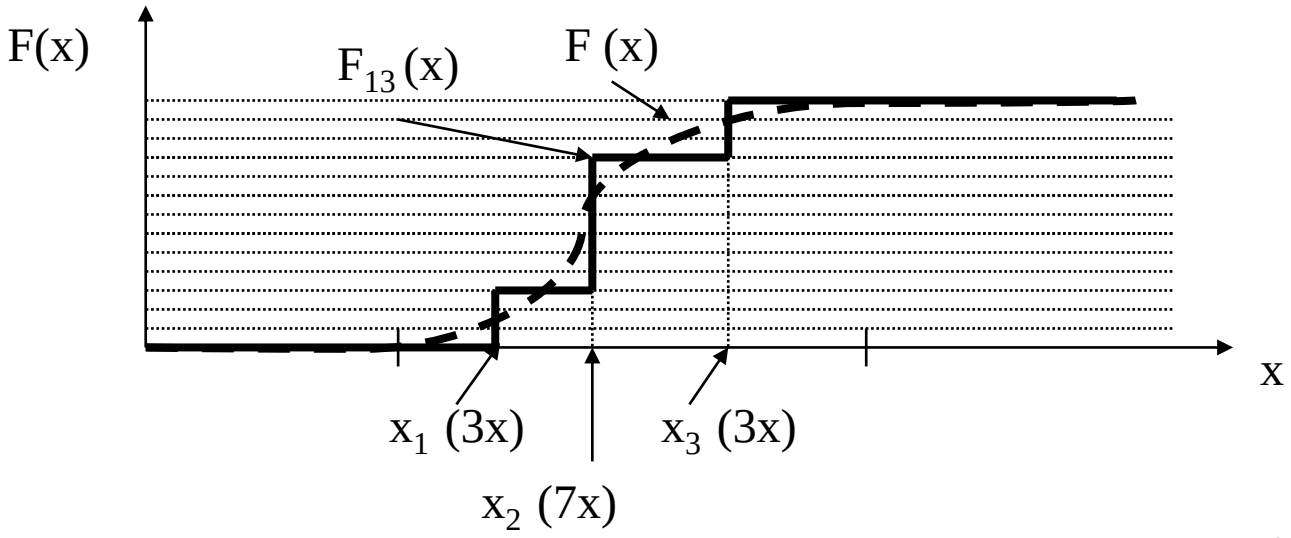
**porovnání F(x) a $F_n(x)$** (tj. teoretické a empirické distribuční funkce)




Empirická distrib. funkce pro n hodnot:

$$F_n(x) = (\text{počet hodnot } x_i \leq x) \ / \ n$$

Příklad: n=13



# Kolmogorov - Smirnovův test

**měření odlišností teoretické a distribuční funkce:**

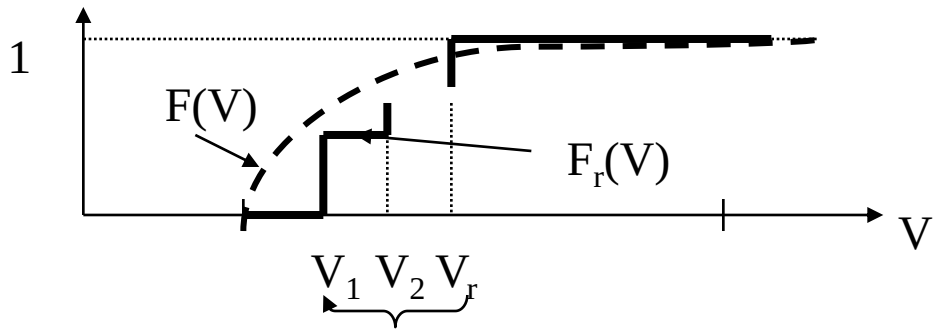
$$K_n^+ = \sqrt{n} \cdot \max_{-\infty<x<+\infty} (F_n(x) - F(x)), K_n^- = \sqrt{n} \cdot \max_{-\infty<x<+\infty} (F(x) - F_n(x))$$

**rozložení hodnot $K_n^+$, $K_n^-$ je tabelováno pro různá n (celkový počet testovaných hodnot)**

| | $p = 99\%$ | $p = 95\%$ | $p = 75\%$ | $p = 50\%$ | $p = 25\%$ | $p = 5\%$ | $p = 1\%$ |
|---|---|---|---|---|---|---|---|
| $n = 1$ | 0.01 | 0.05 | 0.25 | 0.5 | 0.75 | 0.25 | 0.99 |
| .... | | | | | | | |
| $n = 10$ | 0.02912 | 0.1147 | 0.3297 | 0.5426 | 0.7845 | 1.1658 | 1.4444 |
| ... | | | | | | | |
| $n = 30$ | 0.04354 | 0.1351 | 0.3509 | 0.5605 | 0.8036 | 1.1916 | 1.4801 |

Poznámky:

- KS test pouze pro spojité náhodné veličiny
- oba testy kombinovat:
  - provedeme r aplikací testu $\chi^2$ : získáme r hodnot: $V_1, \ldots V_r$
  - z hodnot $V_1, V_2, \ldots V_r$ sestrojíme empirickou distribuční funkci $F_r(V)$
  - funkci $F_r(V)$ porovnáme pomocí K- S testu s požadovanou distribuční funkcí F (V) ( je tabelována)




„větší“ hodnoty (než očekáváno)

# Další empirické testy

- zaměřeny na určitou vlastnost posloupnosti ze které sledujeme n hodnot,
- některé testy jsou aplikovány na reálnou posloupnost z intervalu < 0,1), některé na celočíselnou posloupnost z intervalu < 0, m-1);
- z praktických důvodů se v případě celých čísel můžeme omezit na určitý počet nejvýznamějších bitů, řekněme q bitů => test pak pracuje pouze s hodnotami z intervalu $< 0 , d-1 >$, kde $d = 2^q$ ,

Frekvenční test

- testuje rovnoměrnost rozložení hodnot v daném intervalu, možnosti:
  a) aplikace **K - S testu** - ideální distribuční f-ce:
     F(x) = x na daném intervalu
  b) aplikace **$\chi^2$ testu** – počet kategorií k = d:
     pravděpodobnost každé kategorie $p_k = 1 / d$.

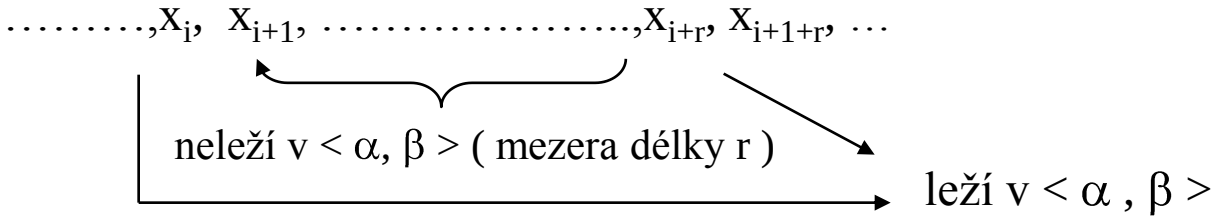
Sériový test:

- aplikace na celočíselnou posloupnost $y_0, y_1, \ldots\ldots, y_n$
- v dané posloupnosti sleduje nepřekrývající se dvojice čísel intervalu $< 0 , d-1 >$, $d= 2^q$ ,
- pro každou dvojici čísel r,s ( $0 \leq r,s < d$ ) čítáme počet případů kdy $(y_{2i}, y_{2i+1}) = ( r,s )$
- aplikace **$\chi^2$ testu** - počet kategorií $k = d^2$ => $p_k = 1 / (d^2)$

# Další empirické testy

Test mezer ( gap test )

- v dané podobě je určen pro rovnoměrně rozložená reálná čísla z intervalu $< 0,1 >$, kdy platí : $P [ 0 \leq \alpha \leq x \leq \beta \leq 1] = \beta - \alpha = p$
- v testované posloupnosti testujeme mezery ( délky úseků, které neleží v podintervalu $< \alpha , \beta >$ )




- zjišťujeme počty mezer o délkách 0, 1, 2, 3, 4, 5, ….., t -1, t a více,
- počet kategorií: $k = t + 1$,
- $p_k$ = pravděpodobnosti mezery délky k:

$p_0 = p$,

$p_1 = p ( 1 - p)$ ,

……..

$p_{t-1} = p ( 1 - p)^{t-1}$,

$p_t = p ( 1 - p)^t$

# Další empirické testy

Poker test

- aplikace na celočíselnou posloupnost, čísla z intervalu < 0, d-1 >
- test po sobě následujících pětic ( obecně k - tic ) čísel, který zjišťuje počet r různých čísel vyskytujících se v dané pětici:
  příklad: k = 5 ( ukázky možných pětic: 13579 => r = 5, 25578 => r = 4 , 11223 => r = 3, 55577 => r = 2, 88888 => r = 1 )

  d……celkový počet různých symbolů (0, 1, 2,……..d -1)

  d (d -1)……( d -r +1)…celkový počet r-tic neobsahující stejné symboly ( variace r-té třídy )

  $d^k$…….celkový počet všech možných k-tic

  $\left\langle \frac{k}{r} \right\rangle$……možný počet rozšíření r různých číslic na k-tici $\left\langle \frac{5}{4} \right\rangle = 10$
  (Stirlingova čísla 2. druhu ….jsou tabelována )

  příklad: počet pětic sestavených rozšířením čtveřice 1234:

  | | |
  |---|---|
  | 1234 **1** | 123 **2** 4 |
  | 1234 **2** | 123 **3** 4 |
  | 1234 **3** | 12 **1** 34 |
  | 1234 **4** | 12 **2** 34 |
  | 123 **1** 4 | 1 **1** 234 |

- **aplikace $\chi^2$ testu:** počet kategorií: k = r
  pravděpodobnost, že v k - tici je právě r různých číslic

$$p_r = \frac{d(d-1)(d-2).....d-r+1)}{d^k} \left\langle \frac{k}{r} \right\rangle$$

# FIPS PUB 140-1

- od r. 1994 : **Federal Information Processing Standard:** americká norma na testování souborů náhodných čísel; zkoumá posloupnosti délky 20 000 bitů a definuje tyto základní testy:

- **test četnosti jedniček:** definuje rozmezí pro dovolenou hodnotu celkového počtu jedniček generované posloupnosti:
  9654 < celkový počet jedniček < 10346.

- **poker test:** zkoumá 5000 čtyřbitových úseků generované ho souboru a tyto interpretuje jako hodnoty z intervalu <0, 15 >; tyto hodnoty jsou základem pro výpočet testovací hodnoty
  $X = 16 / 5000 * [ f(0)^2 + f(1)^2 + f(2)^2 + f(3)^2 + \ldots + f(15)^2 ]$
  pro kterou test definuje povolené rozmezí $1.03 < X < 57.4$

.

- **test úseků stejných znaků:** definuje rozmezí pro celkové počty bloků a mezer délek 1, 2, 3, 4, 5, 6 a více; termínem blok, resp. mezera rozumíme úsek souboru, který obsahuje samé 1 resp. 0.

| délka bloku resp.mezery | povolené rozmezí |
|---|---|
| 1 | 2267 - 2733; |
| 2 | 1079 – 142; |
| 3 | 502 – 748; |
| 4 | 223 – 402; |
| 5 | 90 - 223; |
| 6 a vice | 90 – 223; |

- **test nejdelšího runu:** zkoumá délky nejdelších bloků a mezer; v případě výskytu runu (tj. bloku nebo mezery) o délce přesahující hodnotu 34 pak generovaný soubor je nevyhovující.

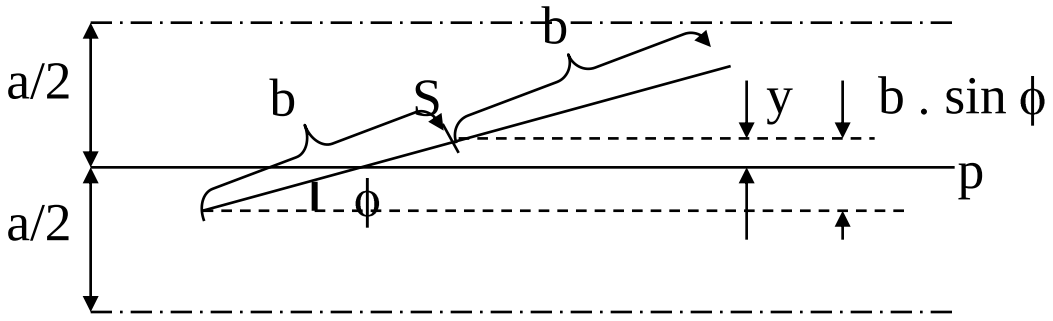
# Jiná použití náhodných čísel

**Metoda Monte Carlo:**

- název pochází od vědců pracujících v USA na vývoji atomové bomby,
- převádí úlohu na stochastický proces, tento simuluje na počítači a statisticky vyhodnotí výsledky ,
- simulační modely SHO: pouze jedna z aplikací metody Monte Carlo
- použití pro výpočty nejrůznějšího typu.

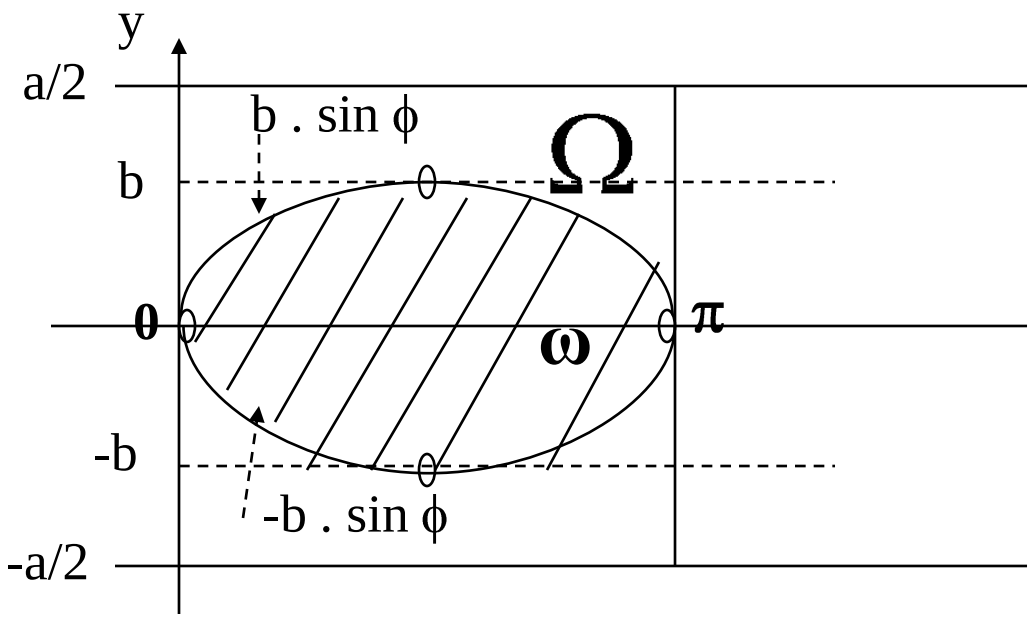
Buffonova úloha: výpočet čísla $\pi$:

- házení jehly na nekonečnou rovinu s rovnoběžkami
- známe: délka jehly = 2b, vzdálenost rovnoběžek = a
- pro každý hod sledujeme zda jehla protne některou z rovnoběžek
- pravděpodobnost protnutí zjistíme dvojím způsobem:
  - vypočteme analyticky (ve zjištěném vzorci se vyskytuje $\pi$)
  - zjistíme experimentálně ( tj..simulací)
  - z rovnosti obou výrazů explicitně vyjádříme číslo $\pi$
- jedna ze vzniklých situací:



# Jiná použití náhodných čísel

Buffonova úloha : pokračování




- místo házení jehly - generování náhodných bodů z oblasti Ω
- poloha jehly - representována dvojicí Φ, y
- protože zřejmě platí: $$|\omega| = 2\int_0^{\pi} b.\sin\varphi d\varphi = 4b$$

$$|\Omega| = a.\pi$$

- pravděpodobnost protnutí rovnoběžky jehlou:

$$= \frac{4.b}{\pi.a} \cong \frac{n}{N} \Rightarrow \pi \cong \frac{4.b}{a} \cdot \frac{N}{n}$$

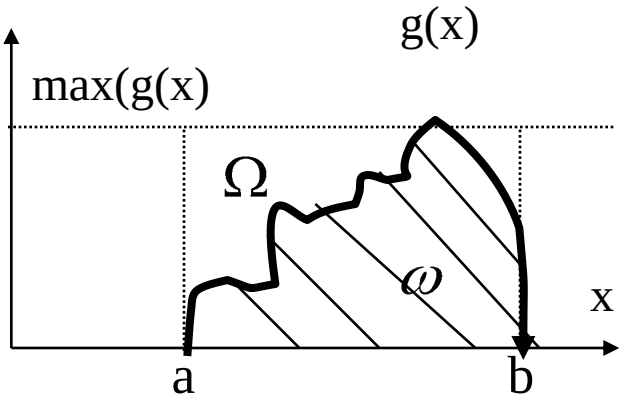
- potřeba: velký počet bodů , znalost b, znalost a, počet úspěšných pokusů (n) a celkový počet pokusů (N)

# Jiná použití náhodných čísel

Výpočet určitých integrálů (i vícerozměrných):

- přístupy založené na generování náhodných bodů z oblasti Ω :

$$I = \int_a^b g(x)dx$$




Metoda odhadu pravděpodobnosti s jakou generovaný bod padne do oblasti ω:

nechť: | ω |….plocha oblasti ω , | Ω | …. plocha oblasti Ω,

N……celkový počet bodů z oblasti Ω ,

n…....počet bodů z oblasti ω,

pak: $$\frac{|\omega|}{|\Omega|} \cong \frac{n}{N} \Rightarrow I = |\omega| = \frac{n}{N}|\Omega|$$

Odhad střední hodnoty uměle vytvořené náhodné veličiny**:**

- nechť X je náh. veličina ( s libovolnou hustotou pravděpodobnosti f (x) ) nabývající hodnot $x_i$ z intervalu a, b
- vytvoříme jinou náhodnou veličinu Q tak , aby E(Q) = I: ,

její hodnoty::

$$q_i = \frac{g(x_i)}{f(x_i)}$$

$$E(Q) = \int_a^b Q(x) \cdot f(x)dx = \int_a^b g(x)dx = I$$

$$\Rightarrow I = E(Q) \cong \frac{1}{N}\sum_{i=1}^{N} q_i$$

# Jiná použití náhodných čísel

Dirichletova úloha:

- je dána čtvercová síť ( tabulka s n * n vnitřními poli ), ve které známe hodnoty v krajních polích
- je třeba najít hodnoty ve všech vnitřních polích tak, aby tyto byly střední hodnotou z hodnot všech (tj. čtyř) polí sousedních

$$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})$$

- zmíněné hodnoty lze nalézt řešením soustavy (n - 2) * (n - 2) algebraických rovnic
- na takovouto takovouto soustavu vede řešení určitého typu parciálních diferenciálních rovnic metodou sítí
- použití metody Monte Carlo spočívá v simulaci velkého počtu „náhodných procházek“ z určitého vnitřního pole do krajních polí se známými hodnotami. Při určování pohybu v průběhu těchto procházek mají všechny čtyři směry stejnou pravděpodobnost. Výslednou hodnotu pro vnitřní výchozí pole lze určit jako aritmetický průměr z cílových hodnot (uložených v krajních polích tabulky) zmíněných procházek. Opakováním takovýchto náhodných procházek z ostatních vnitřních polí lze určit i jejich hodnoty.

Poznámka: na Dirichletovu úlohu lze pomocí diskretizace transformovat řešení některých typů parciálních diferenciálních rovnic