Generovanie náhodných čísel

Kvalitné zdroje náhodných čísel sú dôležitou súčasťou výpočtov vo viacerých oblastiach ako napr.: modelovanie a simulácie, numerická analýza, rozhodovanie, počítačová grafika, kryptografia a dátová komunikácia. Keďže požiadavky na vlastnosti zdroja náhodných čísel sú v rôznych oblastiach rádovo odlišné, je ťažko vytvoriť nejaký univerzálny spôsob generovania náhodných čísel. Spôsobov ako získať náhodné čísla resp. postupnosť náhodných čísel je viacero. V zásade môžeme rozlíšiť nasledovné spôsoby [1]:

1) Hardwarové generátory (snímanie hodnôt nejakého fyzikálneho deja)
2) Tabuľky náhodných čísel (uložené napr. na CDROM)
3) Generovanie náhodných čísel určitým algoritmom z počiatočných hodnôt.

V súčasnosti sa na získanie náhodných čísel takmer výhradne používa spôsob 3, ktorý umožňuje rýchlu softwarovú implementáciu s malými nárokmi na pamäť. Oproti spôsobu 1 má výhodu opakovateľnosti a oproti spôsobu 2 výhodu kompaktnosti. Keďže získaná postupnosť čísel je v podstate deterministická, hovoríme o generovaní, resp. generátoroch pseudonáhodných čísel (GPČ). Nevýhodou je, že vlastnosti odpovedajúceho generátora je treba poznať, resp. starostlivo testovať, aby sme mohli posúdiť jeho vhodnosť resp. nevhodnosť pre danú oblasť použitia. Prehľad základných typov GPČ a ich vlastností je uvedený v častiach #.1 a #.2. Prehľad spôsobov ich testovania je uvedený v časti #.3.

Niekedy (napr. pre metódy typu Monte Carlo) je výhodné generovať výstupnú postupnosť čísel tak, aby čo najrovnomernejšie zapĺňala daný interval, a to aj za cenu, že deterministický spôsob jej generovania je očividný. Takéto postupnosti čísel sa nazývajú kvázináhodné, bližší popis je v časti #.4.

Získané postupnosti náhodných čísel majú zvyčajne rovnomerné rozdelenie (resp. kvázirovnomerné, lebo majú konečný počet hodnôt). Pomocou takýchto hodnôt už ľahko vypočítame hodnoty náhodnej premennej (NP) s daným rozdelením a na danom intervale hodnôt. Prehľad najčastejšie používaných metód je v časti #.5.

#.1 Generátory pseudonáhodných čísel (GPČ)

Ako sa vyvíjala výpočtová technika, vznikali v priebehu rokov viaceré vydarené aj menej vydarené metódy generovania pseudonáhodných čísel na počítači. Ako najvhodnejšie riešenie sa ukázali byť metódy využívajúce rekurenciu, t.j. vzťah, kde výstupné hodnoty sú generované na základe vzťahu

$$x_n = f(x_{n-1}, \ldots, x_{n-k}) \qquad 0 \le k \le n \qquad (\#.1\text{-}1)$$

Prvá metóda tohto typu bola metóda stredných rádov (autor J. von Neuman 1946), kde nasledovné číslo bolo tvorené strednými číslicami druhej mocniny čísla predchádzajúceho. Napr. začnime generovať postupnosť pomocou štvorciferného čísla 6100. Výsledok potom bude: 6100, 2100, 4100, 8100, 6100, ... (vidíme, že dĺžka cyklu je 5). Hoci je táto metóda schopná dať dobré výsledky a dlhú periódu vo všeobecnosti sa od nej kvôli početným nedostatkom už dávno upustilo.

Pri generovaní pseudonáhodných čísel pracujeme vo všeobecnosti na množine čísel $Z_m = \{0,1,2,...,m-1\}$, t.j. v matematike modulo m, kde pojem rovnosť nahrádza pojem kongruencia, definovaný nasledovne: Ak *m* je prirodzené číslo, potom hovoríme, že *a* a *b* sú kongruentné modulo *m* ak majú po delení číslom *m* ten istý zvyšok. Potom píšeme:

$$a \equiv b \pmod{m} \qquad (\#.1\text{-}2)$$

Postupnosť pseudonáhodných čísel u_n (pseudo)náhodnej premennej U s (kvázi)rovnomerným rozdelením zvyčajne generujeme v dvoch fázach:

1) rekurentný vzťah: $x_n = f(x_{n-1}, \ldots, x_{n-k})$ (#.1-3)
2) výstupná hodnota: $u_n = x_n / m, \quad u_n \in <0,1)$ (#.1-4)

Keďže je rozdelenie kvázirovnomerné, pre strednú hodnotu a rozptyl náhodnej premennej U platí:

$$E(U) = \sum_{j=0}^{m-1} u_j p_j = \sum_{j=0}^{m-1} \frac{j}{m}\frac{1}{m} = \frac{1}{2}\left(1 - \frac{1}{m}\right) \quad (\#.1\text{-}5)$$

$$D(U) = E(U^2) - (E(U))^2 = \frac{1}{12}\left(1 - \frac{1}{m^2}\right) \quad (\#.1\text{-}6)$$

Medzi základné vlastnosti, ktoré by mal dobrý generátor spĺňať patria predovšetkým:

1) *dlhá perióda* – v ideálnom prípade nekonečne dlhá, v súčasnosti napr. generátor Mersenne Twister [4] dosahuje 2^{19937}-*1*
2) *rovnomerné a s rastúcou dĺžkou sekvencie čoraz lepšie zaplnenie intervalu* (pokiaľ možno malo by platiť aj pre ľubovoľné subsekvencie)
3) *opakovatelnosť* schopnosť opakovane generovať tú istú sekvenciu pseudonáhodných čísel pomocou jednoducho špecifikovaných počiatočných podmienok
4) *čas generovania* – zanedbateľný oproti času operácií ktoré vytvorenú sekvenciu používajú
5) *požiadavky na pamäť a portabilita* – implementácia nenáročná na pamäť a vo vyššom programovacom jazyku
6) *dobré štatistické vlastnosti* – generátor musí úspešne zdolať súbor štatistických testov, teoretických (ak sú k dispozícii) aj empirických, ktoré je výhodné cielene voliť vzhľadom na oblasť použitia generátora.

V ďalšom texte je uvedený krátky prehľad základných typov GPČ.

#.1.1 Lineárne kongruenčné generátory (LCG)

Generátory tohoto typu zaviedol Lehmer v r.1949. Hodnoty sú generované pomocou rekurentného vzťahu:

$$x_n = (ax_{n-1} + c) \bmod m \quad (\#.1\text{-}7)$$

Sú jednoznačne určené štvoricou koeficientov m,a,c,x_0, pričom x_0 je počiatočná hodnota. Odpovedajúci generátor potom označujeme ako LCG(m,a,c,x_0). Pri $c = 0$, hovoríme o *multiplikatívnom* LCG, pri $c \neq 0$ o *zmiešanom* LCG. Koeficienty niektorých známych LCG sú uvedené v Tab. #.1. Maximálna perióda LCG je *m* a je dosiahnutá, keď:

a) *c* a *m* sú nesúdeliteľné
b) *a-1* je násobkom každého prvočiniteľa *m*
c) *a-1* je násobkom *4*, ak *m* je násobkom *4*

V istých špeciálnych prípadoch môžeme rovno vypočítať dĺžku periódy a zistiť niektoré dodatočné vlastnosti, viď Tab. #.2. Jednoduchým kritériom na posudzovanie náhodnosti LCG je koncept tzv. potencie *s* generátorov s max. Periódou, čo je najmenšie číslo, pre ktoré platí [1]:

$$(a-1)^s \equiv 0 \bmod m \qquad (\#.1\text{-}8)$$

ptičom za dostatočne dobrú je považovaná hodnota $s \geq 5$.

N-tice generovaných hodnôt tvoria vo viacerých rozmeroch pravidelnú mriežkovú štruktúru, čo môže byť v niektorých prípadoch nevýhodné. Avšak aj napriek tomu sú LCG generátory vo všeobecnosti najviac používané. Voľba vhodných koeficientov c, x_0 a predovšetkým m a a nie je triviálna, avšak k dispozícii je viacero osvedčených generátorov, so zaručenými teoretickými vlastnosťami a experimentálne mnohokrát overených.

Názov, použitie LCG	***m***	***a***	***c***	X_0
RANDU –počítače IBM(1960)	2^{31}	65539	0	X_0
ANSI C – funkcia rand()	2^{31}	1103515245	12345	12345
Program DERIVE	2^{32}	3141592653	1	0
Jazyk SIMULA	2^{35}	5^{15}	0	1
ANSI C – funkcia drand48()	2^{48}	25214903917	11	0
Program MAPLE	10^{12}-11	427419669081	0	1
Program MATHEMATICA	a^{48}- a^8+1	$a = 2^{31}$	0	1

Tab.#.1 Všeobecne známe LCG a ich koeficienty

Prípad	max. perióda	Dosiahnutie max. periódy	Poznámka
$M=2^M,\ c > 0$	2^M	a mod *4 = 1* c je nepárne	najnižšie bity nemajú náhodný charakter
$M=2^M,\ c = 0$	2^{M-2}	a mod *8 = 3* alebo *5* x_0 je nepárne	najnižšie bity nemajú náhodný charakter
M – prvočíslo	*m –1*	$x_0 \neq 0$ alebo $c \neq 0$ a je primitívny element*) Z_m	najnižšie bity môžu mať náhodný charakter

*)prvok, ktorý je schopný svojimi mocninami generovať Z_m, t.j. $Z_m = \{a^0, a^1, a^2, ..., a^{m-1}\}$. Ak m je prvočíslo a c_i prvočinitele *m-1*, potom p je primitívny element Z_m, ak pre žiadny c_i neplatí $p^{(m-1)/c_i} \equiv 1$

Tab. #.2 Špeciálne prípady LCG a ich vlastnosti

#.1.2 Fibbonaciho generátory (LFG)

Rekurentný vzťah má tvar:

$$x_n = (x_{n-l} + x_{n-k}) \bmod m\ ;\ \ l > k > 0 \qquad (\#.1\text{-}9)$$

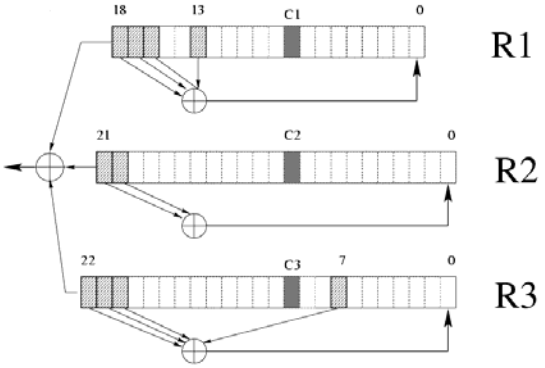
Väčšinou sa volí $m=2^M$, potom max. perióda môže dosiahnuť $(2^l - 1) \times 2^{M-1}$. Vhodná voľba je napr. l=55, k=24, M=31 [1]. Namiesto operácie + sa používajú aj operácie × (lepšie vlastnosti avšak štvrtinová max. perióda), XOR (horšie vlastnosti). Na výpočet potrebujeme posledných l hodnôt x_n. Z l počiatočných hodnôt $x_0, ..., x_{l-1}$ musí byť aspoň jedna nepárna. Vhodné generátory a metódy inicializácie sú uvedené napr. v [1].

#.1.3 Zložené rekurzívne generátory (MRG)

Rekurentný vzťah obsahuje lineárnu kombináciu v tvare:

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \bmod m \qquad (\#.1\text{-}10)$$

Max. perióda je $m^k - 1$. Ich špeciálnym prípadom sú LFSR (Linear Feedback Shift Register) generátory, kde $a_i = \{0,1\}$ a $m = 2$. Sú použité napr. v GSM algoritme A5/1 [Crypto1] (Obr. #.1).



Obr. #.1: Kombinácia troch LFSR generátorov R1, R2, R3 o veľkosti 18, 21 a 22 bitov použitá na generovanie pseudonáhodných bitov v kódovacom algoritme A5/1 pre GSM. V A5/1 je naviac v aktuálnom kroku taktovaný iba generátor, ktorého bit C súhlasí s majoritnou hodnotou bitov C.

#.1.4 Tausworthove generátory

Tausworthove generátory konštruujú výstupnú hodnotu z L hodnôt x_n pomocou

$$u_n = \sum_{j=1}^{L} x_{ns+j-1} m^{-j}, \qquad L \le s \qquad (\#.1.11)$$

pričom sú špeciálnym prípadom (#.1.11), keď m=2. Týmto prístupom je umožnené dosiahnuť viac ako m rôznych hodnôt na <0,1). Ak sú hodnoty x_n generované pomocou MRG s periódou ρ a $nsd(s, \rho) = 1$, potom aj postupnosť u_n generovaná pomocou (#.1.11) má periódu ρ.

#.1.5 Nelineárne generátory

Vo všeobecnosti majú lepšie autokorelačné vlastnosti ako lineárne generátory, lepšie spektrálne vlastnosti (výstupné hodnoty nemajú mriežkovú štruktúru, viď časť 2.3) a lepšie kryptografické vlastnosti. Ich nevýhoda je, že sú pomalšie. Najčastejšie sú používané inverzné kongruenčné generátory (1986 Eichenauer a Lehn) [7][1] a mocninové generátory bitov [1][9].

#.1.5.1 Inverzné kongruenčné generátory (ICG, EICG)

Nech m je prvočíslo a pre $x \in \mathbb{Z}_m$ nech $\bar{x} = 0$ ak $x = 0$ a $\bar{x} = x^{-1} = x^{m-2} \bmod m$ (analogické k Fermatovej vete: $x^m \equiv x \bmod m$), ak $m \neq 0$. Najviac používané sú dve formy generátorov - inverzné (ICG) a explicitne inverzné (EICG) kongruenčné generátory používajúce rekurentné vzťahy [7] [8]:

ICG: $x_n = (a\bar{x}_{n-1} + c) \bmod m \qquad (\#.1.12)$

EICG: $x_n = \overline{a(n + n_0) + c} \bmod m \qquad (\#.1.13)$

Maximálne dosiahnuteľná perióda je *m*. Inverzný prvok sa počíta inverzným Euklidovým algoritmom na nájdenie celočíselných riešení rovnice $\bar{x} \cdot x + k \cdot m = 1$. ICG, EICG majú podstatne lepšie autokorelačné vlastnosti ako LCG. Sú vhodné na použitie aj pre paralelné algoritmy.

#.1.5.2 Mocninové generátory bitov

Generátory tohoto typu sú vhodné predovšetkým na kryptografické účely [9]. Ako príklad si uveďme RSA generátor. Nech $m = p \cdot q$ je súčinom dvoch veľkých prvočísiel. Náhodne zvoľme b také, že $nsd(\phi(m), b) = 1$, kde $\phi(m) = (p-1)(q-1)$. Potom:

$$x_n = x_{n-1}^b \bmod m, \qquad (\#.1.14)$$

pričom $x_o \in \langle 1, m-1 \rangle$. Výstupom je najmenej významný bit x_n. Podobný je BBS (Blum Blum Shub generátor), kde je postupnosť výstupných bitov b_n generovaná pomocou:

$$x_n = x_{n-1}^2 \bmod m \qquad (\#.1.15)$$

$$b_n = x_n \cdot z \bmod 2 \qquad (\#.1.16)$$

,kde x_0 je nesúdeliteľné s *m*, *m* je tvorené súčinom dvoch veľkých prvočísiel, ktoré sa dajú vyjadriť v tvare $4k+3$ a $x_n \cdot z$ predstavuje skalárny súčin po bitoch s náhodnou bitovou maskou z. V [1] je dokázané, že pre náhodne zvolené z, *m*, x_0 uspeje generátor vo všetkých štatistických testoch, ktoré sú menej náročnejšie ako faktorizovanie čísla *m*, t.j. generované bity sú do tejto miery neodlíšitelné od postupnosti skutočne náhodných bitov.

#.2 Metódy zlepšenia vlastností GPČ

Vlastnosti už existujúceho generátora(generátorov), ktorého výstup sa nám nezdá dostatočne náhodný, môžeme zlepšiť napríklad pomocou nasledovných postupov [1] [6]:

a) Aritmetickým skladaním výstupov z viacerých generátorov.

$$z_n = (x_n + y_n) \bmod m, \qquad (\#.2.1)$$

kde x_n resp. y_n sú výstupy z pôvodných generátorov. Vhodné je voliť nesúdeliteľné veľkosti periód jednotlivých generátorov.

b) Premiešavaním (shuffling).
Touto metódou môžeme dodatočne zlepšiť vlastnosti existujúceho generátora, ktorý generuje vstupné hodnoty pre premiešavací algoritmus. Treba si však uvedomiť, že premiešavaním môžeme zmeniť iba poradie vstupných hodnôt, nie hodnoty samotné. Existujú viaceré verzie metódy, napríklad:

I) Prvých *k* vstupných hodnôt x_k uložíme do poľa. Potom z každej ďalšej vstupnej dvojice náhodných čísel pomocou prvého čísla náhodne vyberieme hodnotu z poľa, ktorá bude našim výstupom a druhé vložíme na uvoľnené miesto.

II) Prvých *k* hodnôt x_k uložíme do poľa a *(k+1)*-tu hodnotu do pomocnej premennej idx. Výstupné hodnoty potom generujeme v dvoch krokoch nasledovne:

1) pomocou premennej idx vyberieme hodnotu z poľa ktorá bude našim výstupom a na uvoľnené miesto vložíme novú vstupnú hodnotu
2) výstupnú hodnotu okopírujeme do premennej idx .

Takto sme oproti metóde I) schopní generovať z jednej vstupnej hodnoty jednu hodnotu výstupnú.

#.3 Testy GPČ

Cieľom testov GPČ je zistiť, či sa generované postupnosti správajú dostatočne náhodne. Štatistická teória nám poskytuje iba istú kvantitatívnu mieru pre náhodnosť. Existuje nekonečne veľa vlastností, ktoré môžeme testovať pričom v praxi sa používajú tie, ktoré sa ukázali byť najužitočnejšie. Avšak aj keď generátor uspeje v N testoch nemôžeme si byť istý, že v N+1 teste úplne nezlyhá. Inými slovami pre každý generátor existuje test, pri ktorom úplne zlyhá. S rastúcim počtom úspešne zdolaných testov však môžeme byť čoraz spokojnejší s náhodnosťou generovanej sekvencie a predpokladať že náhodná aj je (kým sa nedokáže opak). Testy môžeme v zásade rozdeliť na dva druhy: teoretické a empirické. Každému druhu sa budeme venovať zvlášť v nasledujúcich častiach. Doplnkovým testom môže byť aj vizuálny test, napr. náhodnosti generovaných bodov pri spektrálnych charakteristikách (viď. Obr.#.3.3), t.j. sledovanie či sa vytvárajú nejaké štruktúry a aké ...

#.3.1 Testovanie hypotéz

Teoretické aj empirické testy na testovanie náhodnosti používajú testovacie procedúry na testovanie hypotéz a to najmä χ^2 a KS test [1][6].

#.3.1.1 χ^2 (Chi-kvadrát) test.

Pri tomto teste testujeme hypotézu H_0: postupnosť pseudonáhodných čísel má dané rozdelenie. Z testovanej postupnosti použijeme ľubovoľných n za sebou nasledujúcich hodnôt $x_1, x_2, ..., x_n$, ktoré rozdelíme do k disjunktných tried. Počty v jednotlivých triedach označme z_j, *j=1,2,...k.* V praxi volíme n také aby v každej triede bolo $z_j > 5$. Ak označíme p_j = *P(hodnota je z j-tej triedy)*, očakávame, že do tejto triedy padne $n\,p_j$ hodnôt. Testovacie kritérium je založené na rozdiele očakávaných a skutočných počtov v triedach:

$$V = \sum_{j=1}^{k} \frac{(z_j - np_j)^2}{np_j} \qquad (\#.3.1)$$

Pre veľké n má V približne rozdelenie chi-kvadrát s k-1 stupňami voľnosti χ^2_{k-1}. Hodnotu V testujeme na zvolenej hladine testu α =*P(správnu* H_0 *zamietneme, t.j. nastane chyba I. druhu)* pomocou kritických hodnôt $\chi^2_{k-1}(\alpha)$. To sú hodnoty, ktoré náhodná veličina Y s rozdelením χ^2_{k-1} prekročí s pravdepodobnosťou α, t.j. $P(Y \geq \chi^2_{k-1}(\alpha)) = \alpha$. V praxi volíme α *=0.01-0.05*. Hodnoty $\chi^2_{k-1}(\alpha)$ sú pre dané *k-1* a α a uvedené v tabuľkách (napr. v [1][5][6]). V našom prípade budú proti H_0 svedčiť nielen príliš veľké hodnoty V (veľký rozdiel oproti očakávanému rozdeleniu), ale aj príliš malé hodnoty (nedostatočnej náhodnosť testovanej postupnosti). Hypotézu H_0 potom zamietneme na hladine α ak platí (#.3.2) alebo (#.3.3):

$$V \leq \chi^2_{k-1}(\alpha/2) \qquad (\#.3.2)$$

$$V \geq \chi^2_{k-1}(1-\alpha/2) \qquad (\#.3.3)$$

Pri overovaní rovnomerného rozdelenia postupnosti pseudonáhodných čísel sa používa *k=10-100* postupne pre rôzne úseky generovanej postupnosti.

Ak chceme zistiť či náhodná veličina Y má predpokladané rozdelenie so známou distribučnou funkciou $F(y) = P(Y \le y)$, môžeme uskutočniť ekvivalentný test, kde testujeme či náhodná veličina

$$Z = F(Y) \qquad (\#.3.4)$$

má rovnomerné rozdelenie.

Príklad 3.1: GPČ sme použili na simuláciu hádzania dvomi hracími kockami. Po 144 hodoch sme zistili nasledovné padnuté počty bodov:

Získaný počet bodov		2	3	4	5	6	7	8	9	10	11	12	Suma
Početnosť hodov (z_j)	GPČ A	4	10	10	13	20	18	18	11	13	14	13	144
	GPČ B	3	7	11	15	19	24	21	17	13	9	5	144

Podľa daných výsledkov otestujte oba generátory na hladine α=0,5.

Riešenie: V našom prípade je počet tried *k=11*. Pomocou počtu pravdepodobnosti si najprv si zistíme očakávané (resp. ideálne) hodnoty početností (pri počte hodov 144):

Trieda (*j*)	1	2	3	4	5	6	7	8	9	10	11
Získaný počet bodov	2	3	4	5	6	7	8	9	10	11	12
Pravdepodobnosť výskytu (p_j)	1/36	1/18	1/12	1/9	5/36	1/6	5/36	1/9	1/12	1/18	1/36
Odpovedajúca početnosť (np_j)	4	8	12	16	19	24	21	17	13	9	5

Z tabuliek dostaneme $\chi^2_{10}(0{,}025) = 3{,}25$ a $\chi^2_{10}(0{,}975) = 20{,}5$. Pomocou vzťahu (#.3.1) pre generátor A dostaneme $V_A = 29.242$, pre generátor B $V_B = 1{,}142$. Použitím (#.3.2) a (#.3.3) zistíme, že generátor A v teste uspel avšak $V_B \le \chi^2_{10}(0{,}025)$, takže na základe (#.3.3) musíme pre generátor B hypotézu, že generovaná postupnosť má očakávané rozdelenie, zamietnuť na hladine α=0,5.

#.3.1.2 **Kolmogorov-Smirnov (KS) test.**

Pri tomto type testu [1] nedelíme hodnoty testovanej postupnosti do tried, ale pracujeme priamo s reálnymi číslami (resp. na počítači s konečným počtom reálnych čísel). T.j. test je vhodný na testovanie spojitých rozdelení. Testujeme zhodu empirickej distribučnej funkcie $F_n(x)$ s očakávanou distribučnou funkciou $F(x)$. Z n hodnôt $x_1, x_2, ..., x_n$ testovanej postupnosti vytvoríme distribučnú funkciu:

$$F_n(x) = (\text{počet hodnôt } x_1, x_2, ..., x_n, \text{ pre ktoré} < x) \text{ / n} \qquad (\#.3.5)$$

Na vykonanie testu potrebujeme 2 hodnoty K_n^+ a $K_n^=$:

$$K_n^+ = \sqrt{n}\max\left(F_n(x) - F(x)\right), \qquad x \in (-\infty, \infty) \qquad (\#.3.6)$$

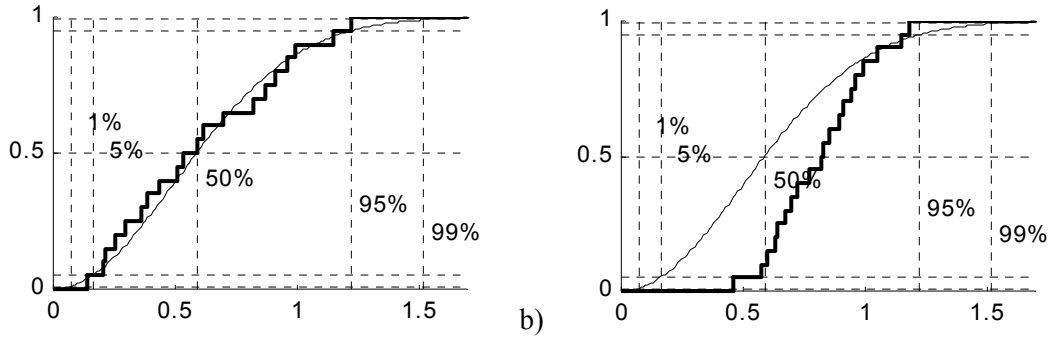
$$K_n^= = \sqrt{n}\max\left(F(x) - F_n(x)\right), \qquad x \in (-\infty, \infty) \qquad (\#.3.7)$$

Na danej hladine testu potom tieto hodnoty porovnávame s kritickými hodnotami z tabuliek, podobne ako tomu bolo pri χ^2 teste. Avšak kritické hodnoty (uvedené napr. v [1]) sú vypočítané pre konkrétne *n* a α a nie sú použité ako aproximácie (na rozdiel od $\chi^2_{k-1}(\alpha)$ v χ^2 teste).

KS test môžeme ľahko znovu aplikovať na výsledky KS testu, keďže distribučná funkcia *F(x)* rozdelenia hodnôt K_n^+ alebo K_n^- sa dá pre veľké *n* aproximovať pomocou:

$$F_\infty(x) = 1 - e^{-2x^2}, \qquad x \geq 0 \qquad (\#.3.8)$$

Takýmto spôsobom môžeme detekovať naraz lokálne aj globálne nenáhodné správanie sa. Napr. rozdelíme testované hodnoty na 20 skupín po 1000, zistíme hodnoty K_{1000}^+ a K_{1000}^- pre každú skupinu a z nich následne pomocou (#.3.8) K_{20}^+ a K_{20}^- (globálne správanie). Viď Obr.#.3.1.



Obr.#.3.1: Príklady empirických distribučných funkcií. Testujeme $F_n(x)$ získanú z 20 hodnôt K_{1000}^+ oproti očakávanej $F_\infty(x)$ zo vzťahu (#.3.8): a) je v norme b) sú evidentné nedostatky (nevyhovuje K_{20}^-), hoci každý dielčí výsledok K_{1000}^+ je v intervale 5-95%, t.j. vyhovel na hladine $\alpha = 0.1$.

#.3.2 **Empirické testy**

Sú to testy na overovanie náhodnosti vygenerovanej postupnosti pseudonáhodných čísel. T.j. testy aplikujeme na postupnosť $u_n = u_0, u_1, u_2, \ldots$ reálnych čísel, o ktorých predpokladnáme, že sú rovnomerne rozdelené na <0,1). V prípade, že testy boli navrhnuté ako celočíselné, použijeme pomocnú testovaciu postupnosť :

$$y_n = \lfloor d u_n \rfloor \qquad (\#.3.9)$$

, v ktorej sú hodnoty rovnomerne rozdelené na $Z_d = \{0,1,\ldots,d-1\}$. Známe batérie testov obsahujúce rozne druhy epmirických testov sú napr. DIEHARD [10] a NIST [11].

#.3.2.1 Test rovnomernosti rozdelenia(test frekvencie)

Základný predpoklad pri všetkých GPČ je, že výstupné hodnoty majú rovnomerné rozdelenie. Pri testovaní tohoto predpokladu sú v zásade dve možnosti:

a) použijeme KS test, kde $F_n(x) = x$ pre $0 \leq x \leq 1$, prípadne opakovaný KS test využitím vzťahu (#.3.9)
b) použijeme χ^2 test , pričom hodnoty transformujeme pomocou (#.3.8). Počet tried sa potom rovná d.

#.3.2.2 **Sériový test**

V sériovom teste overujeme správania sa N-tíc. Z n hodnôt testovanej postupnosti vyberáme postupne neprekrývajúce sa N-tice, ktoré považujeme za súradnice bodov v oblasti v N-rozmernom priestore. Transformujeme ich použitím (#.3.9) a aplikujeme χ^2 test. Počet tried je potom d^N, t.j. treba voliť minimálne $n > 5d^N$. V praxi sa najčastejšie takto testujú dvojice resp. trojice čísel.

#.3.2.3 Test maxima z t hodnôt

Označme $v_j = \max(u_{tj}, u_{tj+1}, ..., u_{tj+(t-1)})$. Potom môžeme postupovať nasledovne:

a) postupnosť v_j má mať rozdelenie s distribučnou funkciou $F(x) = x^t$, čo overíme KS testom
b) použijeme test rovnomernosti rozdelenia na postupnosť hodnôt v_j^t.

#.3.2.4 **Test intervalov medzi „narodeninami"**

Zvoľme m narodenín v roku ktorý ma n dní. Vytvorme zoznam dĺžok intervalov medzi narodeninami. Ak D je počet dĺžok, ktoré sa v zozname vyskytli viac ako raz, potom D má Poissonove rozdelenie so strednou hodnotou $\lambda = m^3/(4n)$, čo platí pre veľké n, napr. $n > 2^{18}$. V [10] je zvolené $n = 2^{24}$, $m = 2^9 \Rightarrow \lambda = 2$. Následne je na 500 vzoriek D aplikovaný chi-kvadrát test. Keďže n má 24 bitov a GPČ generujú čísla väčšinou 32 bitové môžeme z nich vyrobiť 9 postupností keď použijeme iba 1.) bity 1-24 2)bity2-25 ... 9)bity 9-32. Potom dostaneme 9 výsledkov chi-kvadrát testu, na ktoré môžeme aplikovať KS test.

#.3.2.5 **Sériový korelačný test**

Pre dve postupnosti čísel o veľkosti n: $u_0, u_1, ..u_{n-1}$ a $v_0, v_1, ..v_{n-1}$ definujme korelačný koeficient nasledovne:

$$C = \frac{n\sum u_j v_j - \left(\sum u_j\right)\left(\sum v_j\right)}{\sqrt{\left(n\sum u_j^2 - \left(\sum u_j\right)^2\right)\left(n\sum v_j^2 - \left(\sum v_j\right)^2\right)}}, \qquad C \in \langle -1, +1 \rangle \qquad (\#.3.10)$$

Pričom

$$v_j = u_{(j+q) \bmod n}, \; 0 < q < n \qquad (\#.3.11)$$

C nazývame sériový korelačný koeficient ak $q=1$. Vtedy očakávame malú nenulovú hodnotu, pričom vyhovujúce je [1] $C \in (\mu_n - 2\sigma_n, \mu_n + 2\sigma_n)$, kde

$$\mu_n = 1/(1-n) \qquad \sigma_n^2 = \frac{n^2}{(n-1)^2(n-2)}, \quad n > 2 \qquad (\#.3.12)$$

Ak test opakujeme pre rôzne časti postupnosti, do uvedeného intervalu by malo spadnúť 95% hodnôt C. Zároveň očakávame, že s rastúcim q hodnota C klesá (môžeme taktiež testovať).

#.3.2.6 **Run test**

V tomto teste sa testujeme dĺžky monotónnych (rastúcich aj klesajúcich) sekvencií hodnôt v testovanej postupnosti. Získané dĺžky nie sú nezávislé (dlhá postupnosť býva nasledovaná krátkou a naopak), takže aby sme mohli použiť chi-kvadrát test, je nutné ich transformovať aby mali chi-kvadrát rozdeleniene. Transformačné vzťahy sú uvedené v [1].

#.3.2.7 Test minimálnej vzdialenosti

Na testovanú postupnosť sa pozeráme ako na súradnice bodov v rovine, pričom testujeme minimálnu vzdialenosť medi nimi. *N* krát opakuj: a) zvoľ *n* náhodných bodov na jednotkovom štvorci b)Nájdi minimálnu vzdialenosť *d* medzi 2 bodmi zo všetkých $(n^2 - n)/2$ párov. Ak body boli náhodne rozdelené, potom druhé mocniny získaných *N* hodnôt *d* májú exponenciálne rozdelenie so strednou hodnotou μ, t.j. $1 - e^{-d^2/\mu}$ má mať rovnomerné rozdelenie na <0,1), čo sa overí KS testom. Hodnoty použité v [10] sú použité *N*=100, *n*=8000, $\mu = 0{,}995$.

#.3.2.8 Test bitového prúdu

Na testovanú postupnosť sa pozeráme ako na prúd bitov *b1, b2, ... bN*, kde *N*=počet hodnôt postupnosti * bitová náročnosť jednej hodnoty. Z tohoto prúdu bitov vyberáme prekrývajúce sa slová o veľkosti 20 bitov (t.j. *1.slovo=b1,b2,...,b20, 2.slovo=b2,b3,...,b21, ...*) a počítame počet chýbajúcich slov. Z celkového počtu 2^{20} možných slov pri výbere 2^{21} slov zo vstupnejého prúdu bitov očakávame, že počty *J* chýbajúcich slov bude mať približne normálové rozdelenie s $\mu = 141909$ a $\sigma = 428$. Overiť to KS testom môžeme a) hneď b) *J* normujeme transformáciou (*J*-141909)/428) a následne prevedieme na rovnomerné rozdelenie na <0,1) pomocou (**#.**3.4) a testujeme až potom. V [10] je použitá možnosť b) pre 20 hodnôt *J*.

#.3.3 Teoretické testy

Teoretické testy nám poskytujú možnosť odhaliť nedostatky generátorov už pred ich empirickým testovaním. Zároveň umožňujú predpovedať a porozumieť ich správaniu sa za špecifických okolností. Žiaľ vzhľadom k ich komplikovanosti, prípadne obmedzenej platnosti je ich praktické použitie limitované. Pracuje sa s celou preriódou generátora, t.j. niektoré testy (napr. test na rovnomernosť rozdelenia) nemajú veľký zmysel, avšak pre iné testy je táto vlastnosť skôr výhodou. Napr. v [1] sú odvodené vzťahy na testovanie sériovej korelácie LCG generátorov. Z testov sa najčastejšie používajú spektrálny test a test diskrepancie[1]. Spektrálny test môžeme použiť iba na generátory s mriežkovou štruktúrou výstupných hodnôt. Test diskrepancie je všeobecnejší avšak vo viacerých rozmeroch príliš náročný na výpočet.

#.3.3.1 Spektrálny test

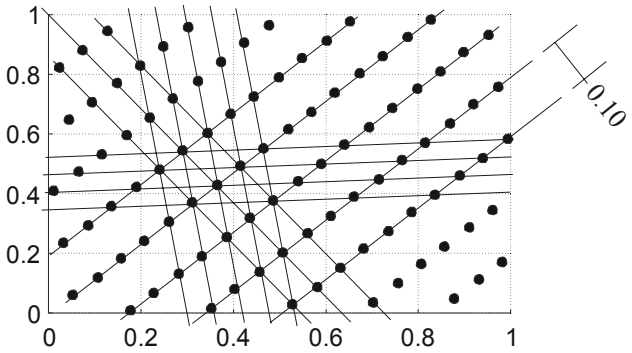
Je to veľmi dôležitý test na kontrolu LCG. Zatiaľ všetky LCG generátory o ktorých sa zistilo, že sú nevyhovujúce, neprešli ani týmto testom. Spektrálny test je úzko spätý so sériovým testom (časť **#.**3.2.2), avšak výsledky sú spoľahlivejšie, lebo sú rotačne invariantné vzhľadom na orientáciu mriežkovej štruktúry vstupných údajov. Označme u_n výstupy z LCG a vytvorme *N*-tice:

$$S_n^N = (u_n, u_{n+1}, \ldots, u_{n+N-1}) \qquad (\#.3.13)$$

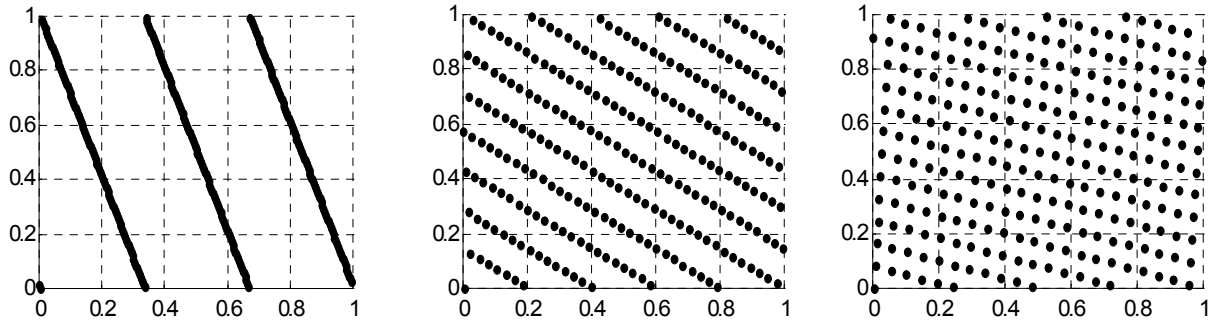
‚ktoré použijeme ako súradnice bodov v *N*-rozmernom priestore. Pri ich grafickom znázornení je evidentná mriežková štruktúra pri vypĺňaní *N*-rozmerného intervalu (viď Obr. #.3.2, Obr. #.3.3). Matematicky formulované: Body S_n^N ležia na množine ekvidištančných paralelných hyperrovín. Spektrálny test nám poskytuje exaktnú numerickú hodnotu ν_N vyplnenia *N*-rozmerného intervalu definovaním $1/\nu_N$ ako maximálnej vzdialenosti medzi hyperrovinami zo všetkých množín paralelných (*N*-1)-rozmerných hyperrovín prekrývajúcich všetky body $\{S_n^N\}$. T.j. v rovine(*N*=2) je $1/\nu_2$ maximálna vzdialenosť medzi susednými priamkami zo všetkých množín paralelných priamok prechádzajúcich všetkými bodmi $\{(u_n, u_{n+1})\}$. Situácia je znázornená na Obr #.3.2. Hodnoty ν_N môžeme normovať na $\nu_N^* \in$ <0,1> pomocou:

$$\nu_N^* = \nu_N / (\gamma_N^{1/2} m^{1/N}), \qquad (\#.3.14)$$

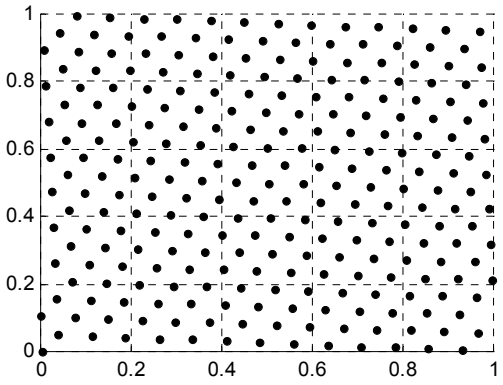
kde γ_N sú Hermitove konštanty[1][2]($\gamma_2 = (4/3)^{1/2}, \gamma_3 = 2^{1/2}, \ldots$).



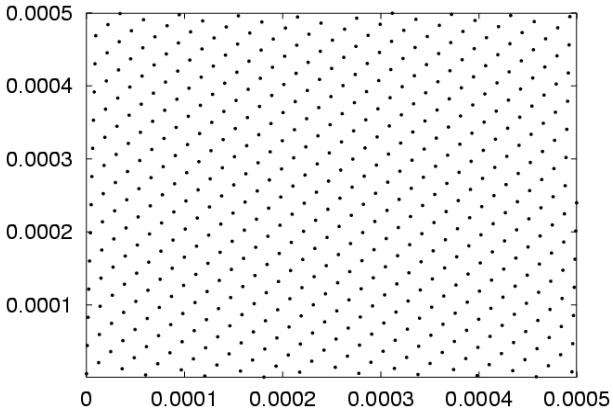
Obr. #.3.2 Príklad spektrálnych charakteristík GPČ (generátor LCG(97,17,0,1)), vynesené u_n voči u_{n-1} . Naznačené sú viaceré smery paralelných priamok a vzdialenosť medzi priamkami pre množinu s maximálnou vzdialenosťou.



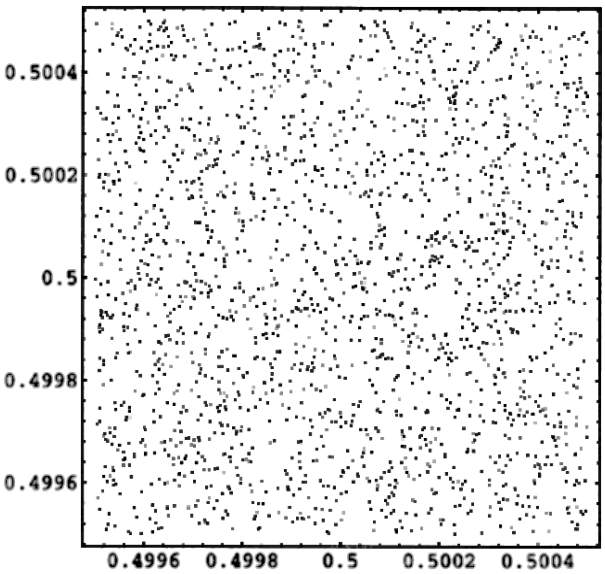
a)LCG(256,85,1,0) b)LCG(256,101,1,0) c)LCG(256,61,1,0)



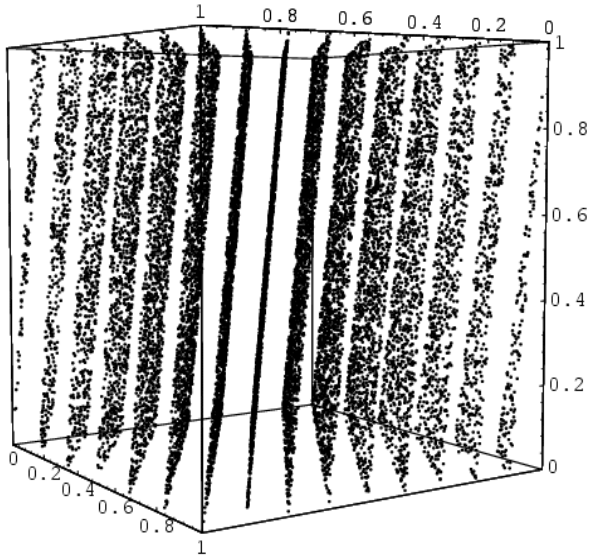
d) LCG(256,237,1,0)



e) LCG: ANSI C - funkcia `rand()`



f) ICG(2^{31}-1 , 1288490188 , 1 , 0)



g) LCG RANDU

Obr. #.3.3 Spektrálne charakteristiky GPČ – vynesené prekrývajúce sa *N*-tice a)–f) dvojice (u_n,u_{n-1}) g) trojice (u_n,u_{n-1},u_{n-2}). Pri LCG je patrná mriežková štruktúra pri ICG nie. Normovaná hodnota spektrálneho testu pre generátory a)–d) sa zlepšuje: a) 0,1839 b) 0,5003 c) 0,7357 d) 0,9196. V f) vidíme zlé vyplnenie priestoru pomocou 15 rovín pri generátore RANDU

#.4 Kvázináhodné postupnosti čísel

Kvázináhodnými postupnosťami nazývame také postupnosti, ktoré rovnomerne vypĺňajú daný interval, pričom medzi po sebe nasledujúcimi hodnotami môže existovať evidentná závislosť. T.j. napr. rekurentný vzťah:

$$x_n = (x_{n-1} + 1) \bmod 10 \qquad (\#.4.1)$$

generuje kvazináhodnú postupnosť. Prakticky sa však používajú iba také algoritmy, ktoré sa snažia o rovnomerné zapĺňanie intervalu už od začiatku postupnosti, t.j. mohli by sme hocikedy prestať a interval by bol danými hodnotami vyplnený rovnomerne. Pritom platí, že ľubovoľná sub-sekvencia zaplní interval približne rovnako rovnomerne ako iná rovnako dlhá sub-sekvencia. Jednoduchým

príkladom tohto druhu postupností sú Haltonove postupnosti. V jednom rozmere na intervale <0,1) sa *j*-ty člen Haltonovej postupnosti H_j generuje nasledovne:

1) zvoľme si prvočíslo *b*, ktoré bude predstavovať základ číselnej sústavy (napr. 2)
2) vyjadrime hodnotu *j* v číselnej sústave zo základom b. (napr. pre j=14, b=2 je výsledok 1100 pri základe 2)
3) hodnotu H_j dostaneme tak, že získané číslice napíšeme za desatinnú čiarku a v opačnom poradí (t.j. z príkladu dostaneme 0.0011 pri základe 2)

Keď chceme generovať n-tice v n-rozmernom priestore, treba použiť v jednotlivých rozmeroch ako základy rôzne prvočísla. Obyčajne sa zvykne používať prvých n prvočísiel. Dobrý prehľad ďalších používaných typov postupností kvázináhodných čísel je uvedený napr. v [3].

Príklad: Vygenerujte Haltonovu postupnosť na intervale $\langle 0,1)$ pre b=2 s periódou 8

Riešenie: Podľa krokov 2,3 postupne vypĺňame tabuľku:

j	0	1	2	3	4	5	6	7
$(j)_2$	000	001	010	011	100	101	110	111
$(H_j)_2$	0,000	0,100	0,010	0,110	0,001	0,101	0,011	0,111
H_j	0	0,5	0,25	0,75	0,125	0,625	0,375	0,875

Takže výsledná postupnosť je {0; 0,5; 0,25; 0,75; 0,125; 0,625; 0,375; 0,875 }.

#.5 Tvorba náhodných premenných s nerovnomerným rozdelením

Existuje viacero spôsobov ako z hodnôt náhodných premenných s istým pôvodným rozdelením generovať hodnoty, ktoré majú rozdelenie želané. Väčšinou sa ako vstup berie rovnomerné rozdelenie, ktoré je priamym výstupom generátorov pseudonáhodných čísel.
Triviálne je generovať rovnomerné rozdelenie na ľubovoľnom intervale $< a,b)$ [6]:

$$uab_n = a + (b-a).u_n, \qquad u_n \in <0,1), \; uab_n \in <a,b) \qquad (\#.5.1)$$

Pri generovaní (a prípadnom testovaní) nerovnomerných rozdelení môžu byť užitočné aj nasledovné vlastnosti:

A) Ak X_1, X_2 sú nezávislé náhodné premenné s distribučnými funkciami $F_1(x)$, $F_2(x)$:

$\max(X_1, X_2)$ má rozdelenie $F_1(x)F_2(x)$ (#.5.2)
$\min(X_1, X_2)$ má rozdelenie $F_1(x) + F_2(x) - F_1(x).F_2(x)$ (#.5.3)

B) Funkcia náhodnej veličiny [5]. Nech X je náhodná premenná s funkciou hustoty pravdepodobnosti $f(x)$ a distribučnou funkciou $F(x)$. Pomocou monotónnej funkcie r vytvorme náhodnú premennú $Y = r(X)$ a označme $g(x)$ jej funkciou hustoty pravdepodobnosti. Potom:

$$g(x) = f\left[r^{-1}(x)\right] \cdot \left|\left(r^{-1}\right)'(x)\right| \qquad (\#.5.4)$$

#.5.1 **Generovanie náhodných premenných so spojitým rozdelením**

Pre generovanie náhodných premenných so spojitým rozdelením existujú viaceré všeobecné metódy, z ktorých sú najznámejšie metóda inverznej funkcie a vylučovacia metóda. Okrem nich existujú aj mmnohé špeciálne metódy, navrhnuté na jedno cieľové rozdelenie náhodnej premennej. Ako príklad uvádzame polárnu metódu na generovanie náhodných premenných s normálovým rozdelením. Ďalšie špeciálne metódy na generovanie normálových, Γ, β , F, χ^2,... rozdelení sú uvedené v [1], niektoré ďalšie všeobecné (napr. kompozičná metóda) a špeciálne metódy sú uvedené aj v [6][#GH].

#.5.1.1 **Metóda inverznej funkcie**

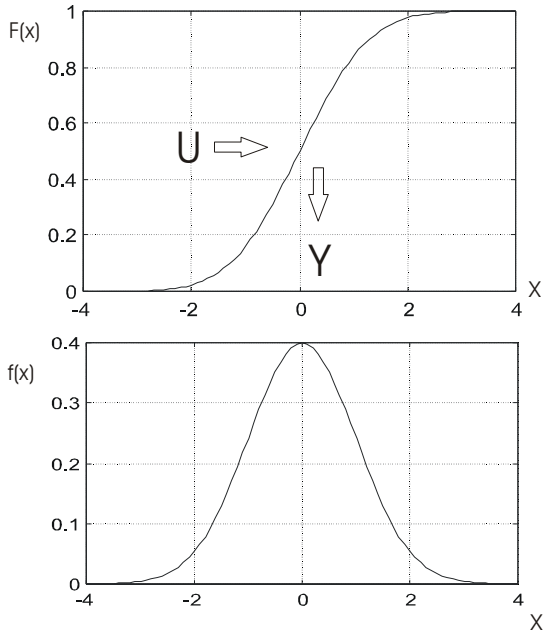
Ak spojitá náhodná premenná *Y* má distribučnú funkciu *F,* potom hodnoty *y* náhodnej premennej *Y* môžeme vytvoriť pomocou náhodnej premennej *U* s rovnomerným rozdelením na intervale <0,1) a hodnotami *u* použitím [1][6][#GH]:

$$Y = F^{-1}(U) \qquad (\#.5.4)$$

Dôkaz [6]: Distribučná funkcia náhodnej premennej *Y* je daná pravdepodobnosťou $P(Y < y)$:

$$P(Y < y) = P(F^{-1}(U) < y) = P(U < F(y)) = \int_0^{F(y)} dx = F(y)$$

teda náhodná premenná *Y* má distribučnú funkciu *F.* Situácia je ilustrovaná na Obr. **#.**5.1.



Obr. **#.**5.1: Metóda inverznej funkcie, zobrazené mapovanie náhodnej premennej *U* s rovnomerným rozdelením na náhodnú premennú *Y* s normálovým rozdelením *N*(0,1). Na spodnom obrázku je funkcia hustoty pravdepodobnosti výslednej náhodnej premennej

Príklad 5.1:Vytvorte spojitú náhodnú premennú *X* s exponenciálnym rozdelením pomocou metódy inverznej funkcie.

Riešenie: Funkcia hustoty pravdepodobnosti náhodnej premennej s exponenciálnym rozdelením je daná ako $f(x) = \lambda\, e^{-\lambda x}$ $(x > 0)$. Tomu odpovedá distribučná funkcia $F(x) = 1 - e^{-\lambda x}$. Inverziou

dostávame $F^{-1}(x) = -\ln(1-x)/\lambda$. Teda ak máme náhodnú premennú U s rovnomerným rozdelením, potom náhodná premenná $X = -\ln(1-U)/\lambda$ má exponenciálne rozdelenie s parametrom λ. Keď uvážime, že 1-U má taktiež rovnomené rozdelenie, môžeme výsledok zjednodušiť na $X = -\ln(U)/\lambda$.

#.5.1.2 **Vylučovacia metóda I.**

Nech spojitá náhodná premenná X *s hodnotami* $x \in (a,b)$ má funkciu hustoty pravdepodobnosti $f(x) \le d$ a distribučnú funkciu $F(x)$. Nech Y a Z sú náhodné premenné nasledovne vytvorené z dvoch nezávislých náhodných premenných s rovnomerným rozdelením U_y a U_z:

$$Y = a + (b-a)U_y \qquad (\#.5.5)$$

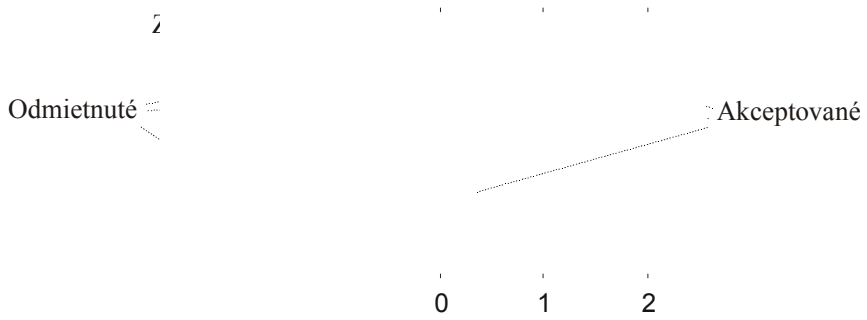
$$Z = dU_z \qquad (\#.5.6)$$

Potom hodnoty náhodnej premennej X sú výberom takých hodnôt Y, pre ktoré platí [6][#GH]:

$$Z \le f(Y) \qquad (\#.5.7)$$

Priemerný počet opakovaní na výber jednej hodnoty je:

$$d(b-a)/\big(F(b)-F(a)\big) \qquad (\#.5.8)$$



Obr. #.5.2: Príklad vylučovacia metódy I pre generovanie náhodnej premennej s normálovým rozdelením $N(0,1)$ na intervale $x \in (-2,2)$. Vygenerované sú prvé 4 hodnoty X.

#.5.1.3 **Vylučovacia metóda II**

Je zovšeobecnením metódy I. Môžeme ju použiť, keď chceme transformovať náhodnú premennú Y so známym rozdelením na náhodnú premennú X so želaným rozdelením. Nech spojité náhodné premenné X *a* Y majú funkcie hustoty pravdepodobnosti $f(x)$ a $g(y)$ a nech c je konštanta pre ktorú platí:

$$\frac{f(y)}{g(y)} \le c \qquad \text{pre všetky} \quad y \qquad (\#.5.9)$$

Potom hodnoty náhodnej premennej X môžeme vypočítať pomocou hodnôt Y a náhodnej premennej U s rovnomerným rozdelením v nasledovných krokoch [1][6]:

1) Generuj po jednej hodnote z Y s daným $g(y)$ a jednu hodnotu z U
2) Ak $U \le f(Y)/cg(Y)$, vyber aktuálnu hodnotu Y, t.j. $X = Y$
3) Opakuj od kroku 1

Priemerný počet opakovaní na výber jednej hodnoty z X je c, , t.j. pri výpočte treba voliť čo najmenšie c pre ktoré platí (#.5.9).

Príklad 5.2: Vytvorte náhodnú premennú s normálovým X rozdelením N(0,1) pomocou exponenciálneho rozdelenia použitím vylučovacej metódy II.

Riešenie: $N(0,1)$ má absolutnu hodnotu funkciu hustoty prevdepodobnosti:

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \qquad 0 < x < \infty$$

Zvoľme:

$$g(x) = e^{-x}, \qquad 0 < x < \infty$$

Hľadaním maxima podielu týchto dvoch funkcií dostaneme

$$c = \sqrt{2e/\pi} \approx 1.32$$

Hodnoty náhodnej premennej X s $N(0,1)$ vytvoríme v nasledovných krokoch:

1) generujme hodnotu z Y s exponenciálnym rozdelením s $\lambda = 1$ a hodnotu U s rovnomerným rozdelením na (0,1).
2) Ak $-\log U \geq (Y-1)^2/2$, vyber $|X|$, t.j. $|X| = Y$, ináč opakuj od kroku 1)
3) S rovnakou pravdepodobnosťou vyber $-X$ alebo X. Opakuj od kroku 1)

Poznamka: ak chceme generovať náhodnú premennú Z s rozdelením $N(\mu, \sigma^2)$, potom je treba výsledok transformovať pomocou $Z = \mu + \sigma X$

#.5.1.4 **Polárna metóda**

Je príkladom špeciálnej metódy, pomocou ktorej generujeme dve nezávislé náhodné premenné X_1, X_2 s normálovým rozdelením. Postup je nasledovný[1]:

1) Vygeneruj po jednej hodnote z dvoch nezávislých náhodných premenných U_1, U_2 s rovnomerným rozdelením
2) Vytvor premenné $V_1 = 2U_1 - 1$, $V_2 = 2U_2 - 1$, ktoré majú rovnomerne rozdelené na <-1, 1)
3) Ak pre $S = V_1^2 + V_2^2$ platí $S \geq 1$, opakuj postup znovu od bodu 1
4) Vygeneruj výstupné hodnoty náhodných premenných s normálovým rozdelením

$$X_1 = V_1\sqrt{(-2\ln S)/S} \qquad X_2 = V_2\sqrt{(-2\ln S)/S}$$

a opakuj postup od bodu 1

#.5.4 **Generovanie vybraných diskrétnych rozdelení:**

V tejto časti uvádzame niektoré dôležité diskrétne rozdelenia a príklady generovania odpovedajúcich náhodných premenných použitím spojitých náhodných premenných U s rovnomerným rozdelením [1]:

a) Geometrické rozdelenie - má náhodná premenná G, ktorá predstavuje počet nezávislých pokusov medzi výskytmi nejakej udalosti, ktorá sa vyskytuje z pravdepodobnosťou p. T.j. hodnoty G sú rovné n s pravdepodobnosťou $(1-p)^{n-1}p$. Môžeme ich generovať pomocou:

$$G = \lceil \ln U / \ln(1-p) \rceil \qquad (\#.5.10)$$

, kde operátor $\lceil \ \rceil$ znamená najvyššie bližšie celé číslo.

b) Binomické rozdelenie (*t,p*) – má *N*, počet výskytov udalosti, ktorá môže nastať s pravdepodobnosťou p v každom z t nezávislých pokusov, ktoré sú k dispozícii. Potom: $P(N=n)=\binom{t}{n}p^n(1-p)^{t-n}$. T.j generujeme po jednej hodnote z *t* generátorov $U_1,U_2,...,U_t$ a počítame koľko z nich má hodnotu menšiu ako *p*. Algoritmus je efektívny pre malé *t*.

c) Poissonove rozdelenie so strednou hodnotou μ . Medzi Poissonovým a exponenciálnym rozdelením je podobný vzťah ako medzi binomickým a geometrickým rozdelením: Reprezentuje počet výskytov udalosti za jednotkový čas. Udalosť sa môže vyskytnúť hocikedy. Hodnoty náhodnej premennej *N* s týmto rozdelením môžeme generovať nasledovne: Sčítavame po jednej hodnote náhodných premenných $X_1, X_2,...$ s exponenciálnym rozdelením s so strednou hodnotou $1/\mu$ a zisťujeme koľko ich treba na splnenie podmienky $X_1 + X_2 + ...X_m \geq 1$. Ak uvážime výsledok príkladu 5.1 môžeme podmienku napísať v tvare $U_1 \cdot U_2 \cdot ... \cdot U_m \leq e^{-\mu}$. Následne vygenerujeme hodnotu *N*=*m*-1 a postup opakujeme. Algoritmus je efektívny pre malé μ.

Príklady:

Príklad 5.3: Vytvorte prvých 5 hodnôt náhodnej premennej s funkciou hustoty pravdepodobnosti $f(x)=3x$ ak $x \in \langle a,b \rangle$, ináč $f(x)=0$. Vhodne zvoľte $\langle a,b \rangle$, aby $F(\infty)=1$. Použite generátor LCG(509,108,0,1)

Príklad 5.4: Vytvorte prvých 5 hodnôt náhodnej premennej s exponenciálnym rozdelením použitím vylučovacej metódy I. Použite generátor LCG(97,17,0,1)

Príklad 5.5: Zjedodušte riešenie prikladu 5.2 ak vieme (viď Príklad 5.1), že –logU generuje hodnoty exponenciálneho rozdelenia s $\lambda = 1$.

Príklad 5.6: Pomocou metódy inverznej funkcie ukážte, že náhodné premenné X, Y
$X = \max(U_1,U_2)$, $Y = \sqrt{U}$. Majú ekvivalentné rozdelenia.

Príklad 5.7.: Zistite funkciu hustoty prevdepodobnosti $g(y)$ náhodnej premennej Y

$$Y = y(X) = (X-a)/b, \ b \neq 0$$

kde, X má hustotu pravdepodobnosti $f(x)=1$ *ak* $0<x<1$, ináč $f(x)=0$. Použite vedomosti o funkcii náhodnej premennej.

[1] D. E. Knuth: The art of computer programming, Addison Wesley Longman, 1998

[2] P. L'Ecuyer, “Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure”, Mathematics of Computation, 68, 225 (1999), 249--260.
[3] Bratley P., Fox, B.L. 1988, ACM Transactions on Mathematical Software, vol. 14, pp.88-100.
[4] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", ACM Trans. on Modeling and Computer Simulation Vol. 8, No. 1, Januray pp.3-30 1998
[5] J. Hátle, J. Likeš, Základy počtu pravdepodobnosti a matematické statistiky, SNTL, 1972
[6] Š.Neuschl, Modelovanie a simulácie, Alfa, 1988
[7] J. Eichenauer and J. Lehn. A nonlinear congruential pseudorandom number generator. *Statist. Hefte*, 37:315-326, 1986.

[8] Hellekalek, P.: Inversive pseudorandom number generators: concepts, results, and links**.** Proceedings of the 1995 Winter Simulation Conference, pp. 255--262. , 1995

[9] Menezes,A., Oorschot, P., Vanstone, S, Handbook of Applied Cryptography, CRC Press 1996

[10] G. Marsaglia. Diehard. ftp://stat.fsu.edu/pub/diehard.

[11] Soto, J.: "Statistical Testing of Random Number Generators," Proceedings of the 22nd National Information Systems Security Conference, 10/99.

[12] Shamir, A., Biryukov, A., Real Time Cryptanalysis of A5/1 on a PC, Fast Software Encryption Workshop 2000, April 10-12, 2000, New York City.

[#GH] Gross – Harris, …