Generovanie náhodných čísel

Náhodné čísla sú dôležitá súčasť výpočtov v:

- modelovaní a simuláciách
- numerickej analýze
- rozhodovaní
- počítačovej grafike
- kryptografii
- dátovej komunikácií
- ...

Základné spôsoby získania postupnosti náhodných čísel:

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.

Súčasnosť ➔ používa sa najmä spôsob 3 - generovanie náhodných čísel algoritmom

Výhody

- rýchla SW implementácia s malými nárokmi na pamäť
- Oproti spôsobu 1 má výhodu opakovateľnosti
- oproti spôsobu 2 výhodu kompaktnosti

Vlastnosti

- získaná postupnosť čísel je v podstate deterministická ➔ , hovoríme o generovaní, resp. generátoroch ***pseudonáhodných*** čísel (GPČ)
- vlastnosti odpovedajúceho generátora je treba poznať, resp. starostlivo testovať, aby sme mohli posúdiť jeho vhodnosť resp. nevhodnosť pre danú oblasť použitia.
- Získané postupnosti náhodných čísel majú zvyčajne ***rovnomerné resp. kváziromerné (=diskrétne rovnomerné) rozdelenie***
- Je možné dosiahnuť iba ***konečnú periódu*** generovaných dát

Generátory pseudonáhodných čísel (GPČ)

Najpoužívanejšie riešenie - metódy využívajúce ***rekurenciu***, t.j. vzťah, kde výstupné hodnoty sú generované na základe vzťahu

$$x_n = f(x_{n-1}, \ldots, x_{n-k}) \qquad 0 \leq k \leq n$$

História:
Prvá metóda tohto typu - ***metóda stredných rádov*** (autor J. von Neuman 1946).
Nasledovné číslo bolo tvorené strednými číslicami druhej mocniny čísla predchádzajúceho.

***Príklad**:*

Generujteposupnosť pseudonáhodných čísiel pomocou metódy stredných rádov. Začnite z čísla 6100.

***Riešenie**:*

6100^2=37210000; 2100^2=4410000; 4100^2=16810000; 8100^2=65610000

T.j. výsledkom jepostupnosť (6100, 2100, 4100, 8100, 6100, ...).
Vidíme, že dĺžka cyklu je 5.

Pravidlá pri generovaní pseudonáhodných čísel:

- pracujeme vo všeobecnosti v ***matematike modulo m***, t.j. na množine čísel

$$Z_m = \{0,1,2,\ldots,m-1\}$$

- pojem rovnosť nahrádza pojem ***kongruencia***
  Ak m je prirodzené číslo, potom hovoríme:
  a a b sú ***kongruentné modulo m*** ak majú po delení číslom m ten istý zvyšok.
  Píšeme:

$$a \equiv b \pmod{m}$$

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

1) rekurentný vzťah: $x_n = f(x_{n-1}, \ldots, x_{n-k})$

2) výstupná hodnota: $u_n = x_n / m, \quad u_n \in <0,1)$

Základné vlastnosti, ktoré by mal dobrý generátor spĺňať:

1) ***dlhá perióda*** – v ideálnom prípade nekonečne dlhá, v súčasnosti napr. generátor Mersenne Twister dosahuje periódu $2^{19937}-1$

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.

Základné typy GPČ.

Lineárne kongruenčné generátory (LCG)

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

$$x_n = (ax_{n-1} + c) \bmod m$$

Označujeme ho ako: ***LCG(m,a,c,x_0).***

Platí:

- Pri $c = 0$, hovoríme o ***multiplikatívnom*** LCG
- pri $c \neq 0$ o ***zmiešanom*** LCG.

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*

Všeobecne známe LCG a ich koeficienty

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

Spektrálne charakteristiky LCG

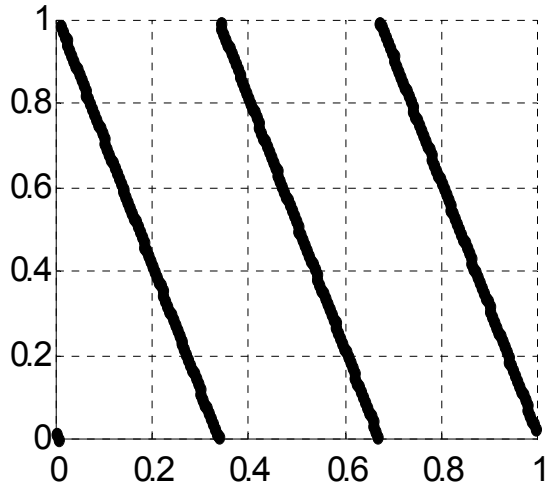
Označme u_n výstupy z LCG a vytvorme *N*-tice:

$$S_n^N = \left(u_n, u_{n+1}, \ldots, u_{n+N-1}\right)$$

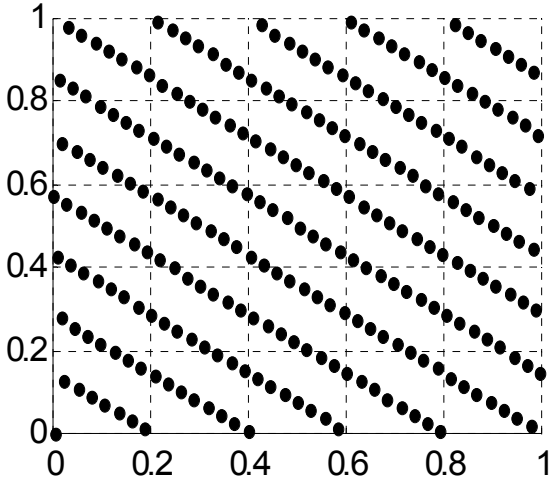
Použijeme ich ako súradnice bodov v *N*-rozmernom priestore.

Výsledok:

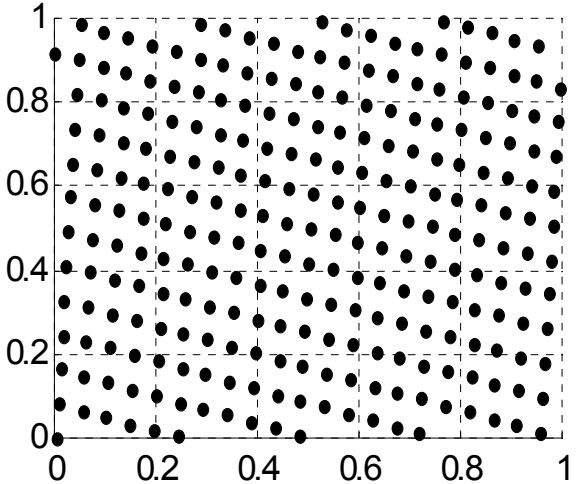
- *mriežková štruktúra* pri vypĺňaní *N*-rozmerného intervalu
- *t.j.* body S_n^N ležia na množine *ekvidištančných paralelných hyperrovín*.
- aj napriek tomu sú LCG generátory vo všeobecnosti najviac používané.



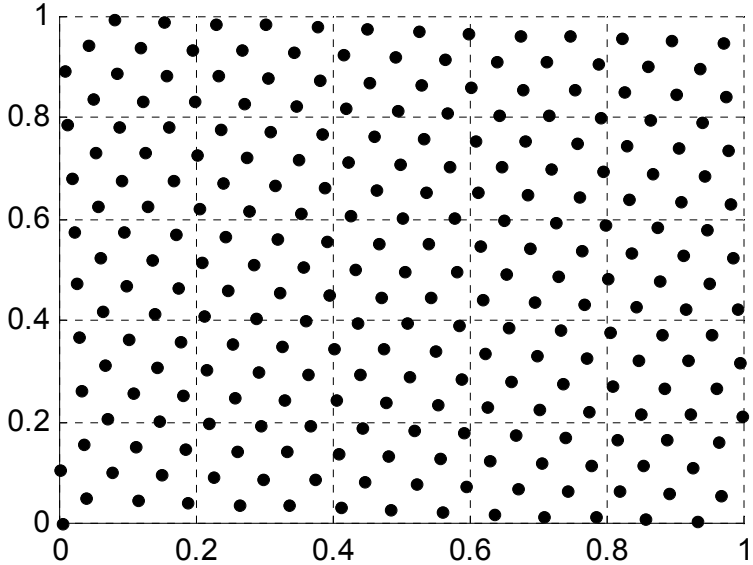
a)LCG(256,85,1,0)



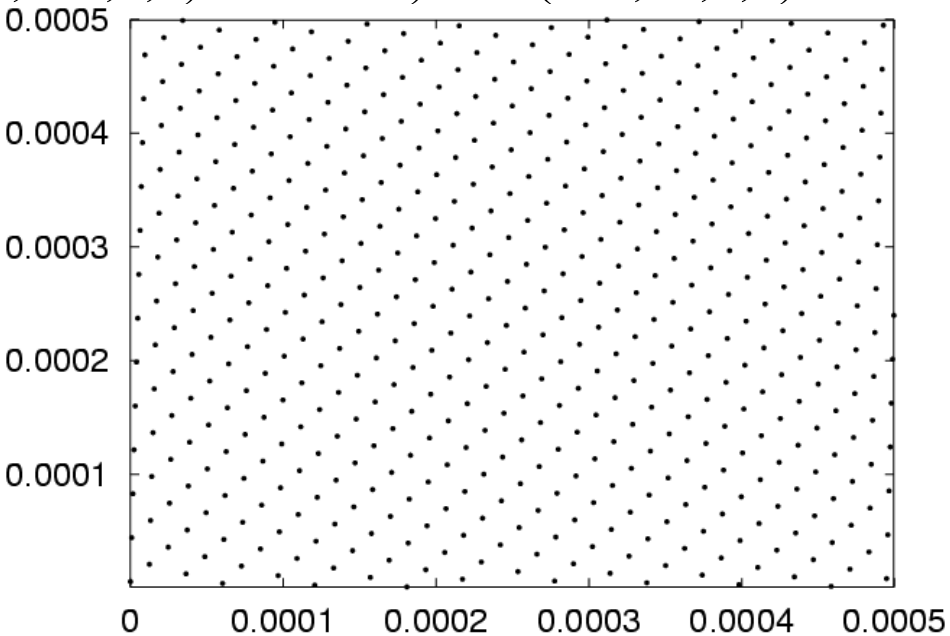
b)LCG(256,101,1,0)



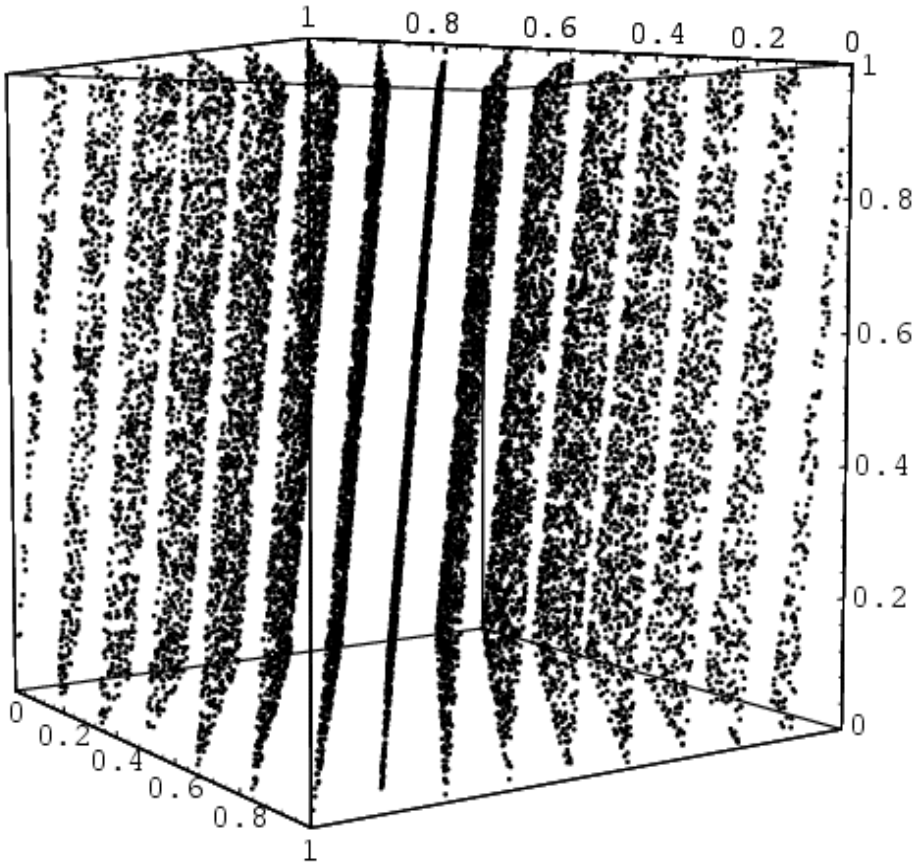
c)LCG(256,61,1,0)



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



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



Zlé vyplnenie priestoru pomocou 15 rovín pri LCG generátore RANDU

Fibbonaciho generátory (LFG)

Rekurentný vzťah má tvar:

$$x_n = (x_{n-l} + x_{n-k}) \bmod m \; ; \quad l > k > 0$$

- Väčšinou sa volí $m=2^M$ ➔ max. perióda môže dosiahnuť $(2^l - 1) \times 2^{M-1}$
- Vhodná voľba je napr. l=55, k=24, M=31
- Namiesto operácie + sa používajú aj operácie
  - "×" lepšie vlastnosti avšak štvrtinová max. perióda
  - XOR horšie vlastnosti, jednoduchý výpočet
- Na výpočet potrebujeme posledných l hodnôt x_n.
- Z l počiatočných hodnôt $x_0, \ldots, x_{l-1}$ musí byť aspoň jedna nepárna.

Zložené rekurzívne generátory (MRG)

Rekurentný vzťah:

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \bmod m$$

Max. perióda je $m^k - 1$.

Špeciálny prípad ➔ LFSR (Linear Feedback Shift Register) generátory, kde $a_i = \{0,1\}$ a $m = 2$.
Sú použité napr. v GSM algoritme A5/1.

Nelineárne generátory

Výhody

- lepšie autokorelačné vlastnosti ako lineárne generátory
- lepšie spektrálne vlastnosti (výstupné hodnoty nemajú mriežkovú štruktúru)
- lepšie kryptografické vlastnosti

Nevýhody

- pomalšie

Inverzné kongruenčné generátory (ICG, EICG)

Nech m je prvočíslo a pre $x \in Z_m$ 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 $m \neq 0$. Potom platí

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

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

, kde ICG označuje inverzné a EICG explicitne inverzné kongruenčné generátory.

Vlastnosti

- 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 pre paralelné algoritmy.

Mocninové generátory bitov

Generátory tohoto typu sú vhodné predovšetkým na kryptografické účely.

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:

$$x_n = x_{n-1}^b \bmod m$$

pričom $x_o \in \langle 1, m-1 \rangle$. Výstupom je najmenej významný bit x_n.

BBS (Blum Blum Shub) generátor

Postupnosť výstupných bitov b_n generovaná pomocou

$$x_n = x_{n-1}^2 \bmod m \qquad b_n = x_n \cdot z \bmod 2$$

kde

- x_0 je nesúdeliteľné s m
- m je tvorené súčinom dvoch veľkých prvočísiel, ktoré sa dajú vyjadriť v tvare $4k+3$
- $x_n \cdot z$ predstavuje skalárny súčin po bitoch s náhodnou bitovou maskou z

Vlastnosti:

- 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.

Metódy zlepšenia vlastností GPČ

- Vlastnosti už existujúcich generátorov, môžeme zlepšiť a nevhodným algoritmom resp. voľbou konštnánt aj zhoršiť

a) ***Aritmetickým skladaním*** výstupov z viacerých generátorov.

$$z_n = (x_n + y_n) \bmod m$$

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 premiešavania, 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 .

Výhoda***: Oproti metóde I) sme schopní generovať z jednej vstupnej hodnoty jednu hodnotu výstupnú.***

Kvázináhodné postupnosti čísel

➔ také postupnosti, ktoré rovnomerne vypĺňajú daný interval, pričom medzi po sebe nasledujúcimi hodnotami môže existovať evidentná závislosť.

Príklad:

$$x_n = (x_{n-1} + 1) \bmod 10$$

- Prakticky sa 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.
- platí, že ľubovoľná sub-sekvencia zaplní interval približne rovnako rovnomerne ako iná rovnako dlhá sub-sekvencia

Príklad:

➔ Haltonove postupnosti

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.

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 }.

Testy GPČ

Cieľom testov GPČ ➔ zistiť, či sa generované postupnosti správajú dostatočne náhodne.

- Existuje nekonečne veľa vlastností, ktoré môžeme testovať
- v praxi sa používajú tie, ktoré sa ukázali byť najužitočnejšie.
- aj keď generátor uspeje v *N* testoch nemôžeme si byť istý, že v *N+1* teste úplne nezlyhá. (t.j. 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.

Základné druhy testov

- ***Teoretické***
- ***Empirické***

Teoretické aj empirické testy na testovanie náhodnosti používajú testovacie procedúry na testovanie hypotéz a to najmä

- χ^2 test
- KS (Kolmogorov - Smirnov) test

Empirické testy

- testy na overovanie náhodnosti ***vygenerovanej*** postupnosti pseudonáhodných čísel

- testy aplikujeme na postupnosť $u_n = u_0, u_1, u_2, \ldots$ reálnych čísel, o ktorých predpokladnáme, že sú rovnomerne rozdelené na intervale <0,1)

- V prípade, že testy boli navrhnuté ako celočíselné, použijeme pomocnú testovaciu postupnosť :

$$y_n = \lfloor d u_n \rfloor$$

,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 a NIST.

Test maxima z t hodnôt

Označme $v_j = \max(u_{tj}, u_{tj+1}, \ldots, u_{tj+(t-1)})$. 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.

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 párov. Ak body boli náhodne rozdelené, potom druhé mocniny získaných *N* hodnôt *d* majú exponenciálne rozdelenie so strednou hodnotou μ má mať rovnomerné rozdelenie na <0,1), čo sa overí KS testom.

Teoretické testy

- poskytujú možnosť odhaliť nedostatky generátorov už pred ich empirickým testovaním.
- umožňujú predpovedať a porozumieť ich správaniu sa za špecifických okolností.
  - Pracuje sa s celou periódou generátora ➔ niektoré testy (napr. test na rovnomernosť rozdelenia) nemajú veľký zmysel

Najčastejšie používajú

- spektrálny test - použiteľný iba na generátory s mriežkovou štruktúrou výstupných hodnôt.
- test diskrepancie - všeobecnejší avšak vo viacerých rozmeroch príliš náročný na výpočet

Spektrálny test

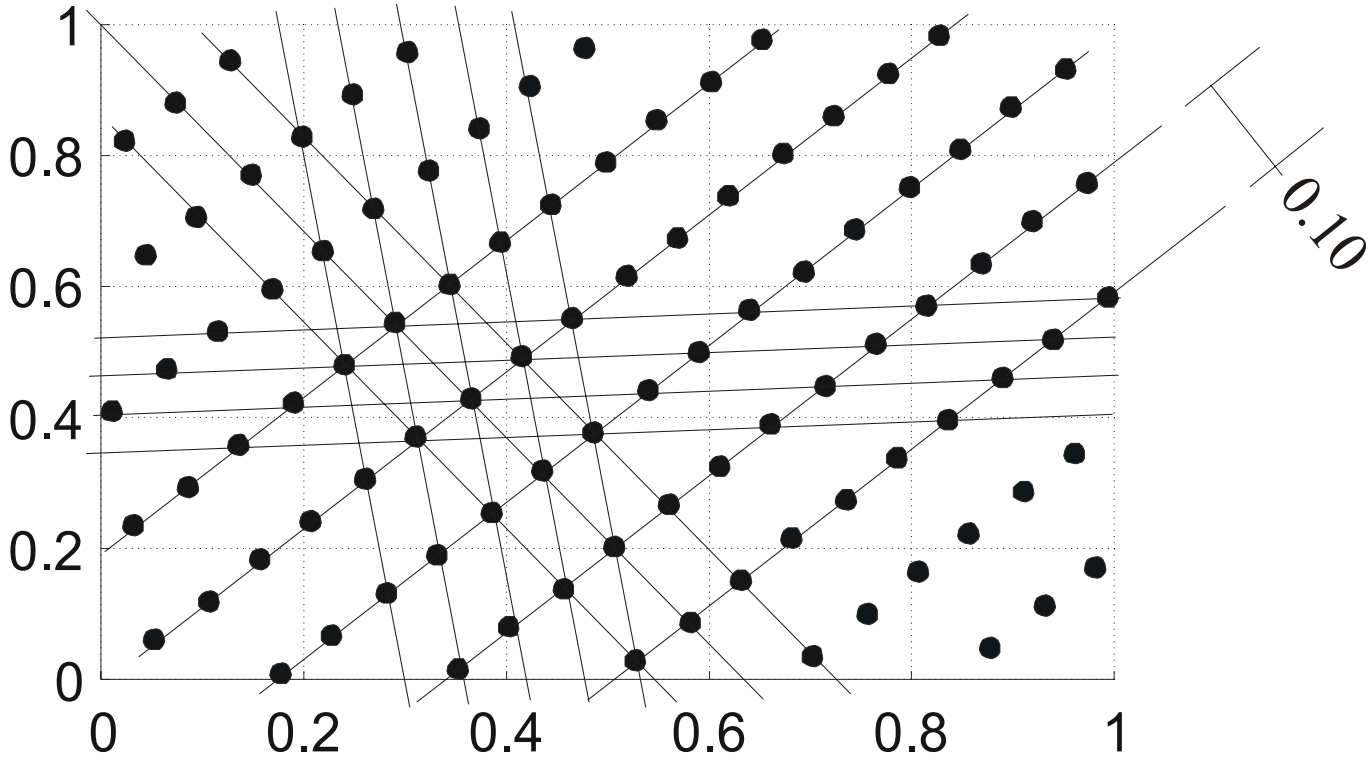
- veľmi dôležitý test na kontrolu LCG
- Zatiaľ všetky LCG generátory o ktorých sa zistilo, že sú nevyhovujúce, neprešli ani týmto testom.
- Spektrálny test je úzko spätý so *sériovým testom* avšak výsledky sú spoľahlivejšie, lebo sú rotačne invariantné vzhľadom na orientáciu mriežkovej štruktúry vstupných údajov.

Označme u_n výstupy z LCG a vytvorme *N*-tice:

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

Použijeme ich ako súradnice bodov v *N*-rozmernom priestore.

Je evidentná *mriežková štruktúra* pri vypĺňaní *N*-rozmerného intervalu
➔ Body S_n^N ležia na množine *ekvidištančných paralelných hyperrovín*.



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.