Procedurálne modelovanie

Róbert Bohdal
bohdal@fmph.uniba.sk
flurry.dg.fmph.uniba.sk/webog/bohdal-vyucba

Modelovanie a teória simulácie

- Spojité ↔ Diskrétne systémy
- Deterministické ↔ Stochastické systémy

- Skutočný (reálny systém)

  ↓ **analýza systému**
- Abstraktný model

  ↓ **implementácia modelu**
- Počítačový model

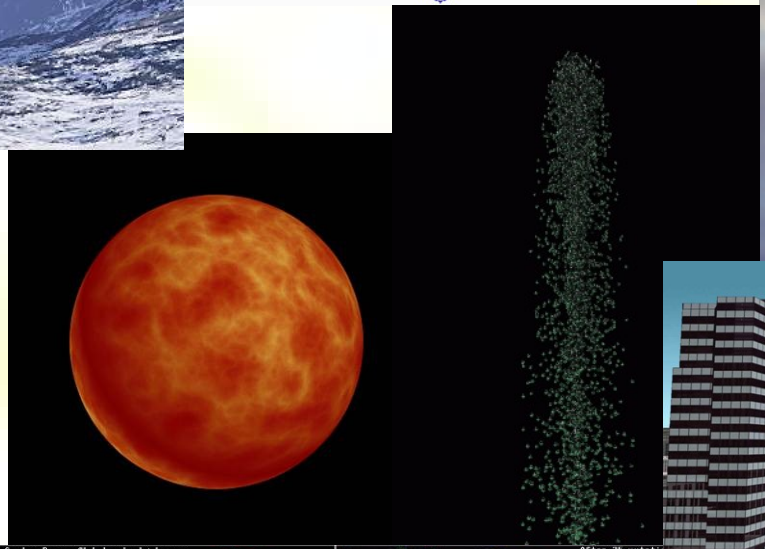
- Popisný jazyk:

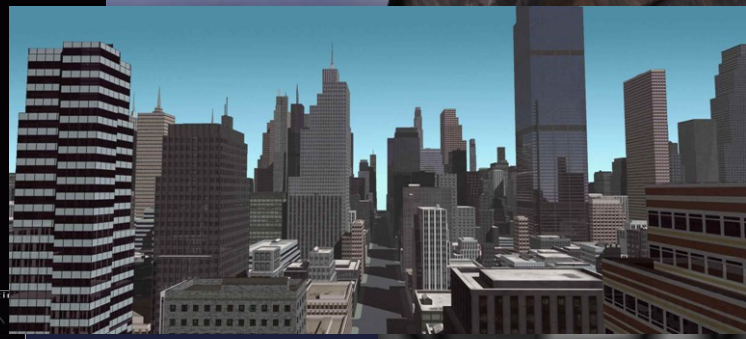
  slovný / schématický / matematický / vývoj. diagram / …

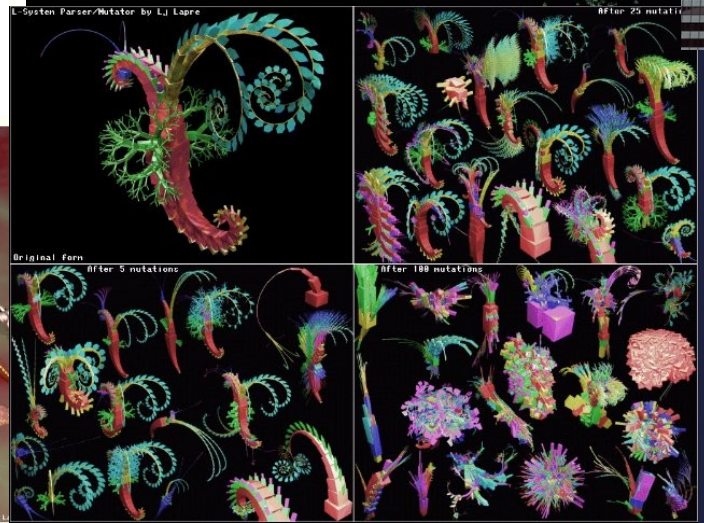
Procedurálne modelovanie

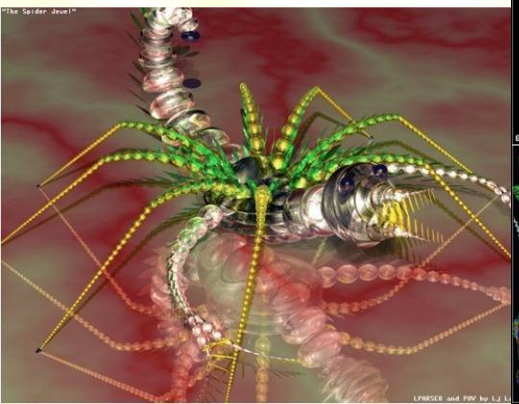
- Tvorba algoritmov pre modely
- Parametre algoritmov
- Zložitosť algoritmov
- Pamäťové a výpočtové nároky
- Simulácia modelov
- Implementácia/využitie výsledkov

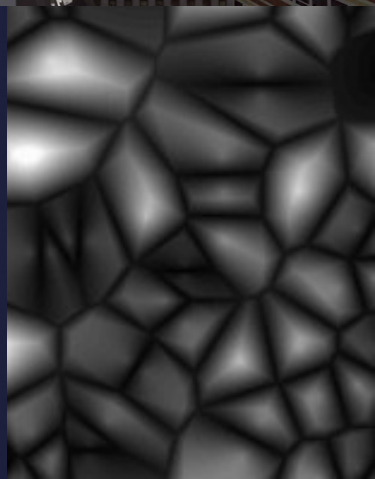
Príklady

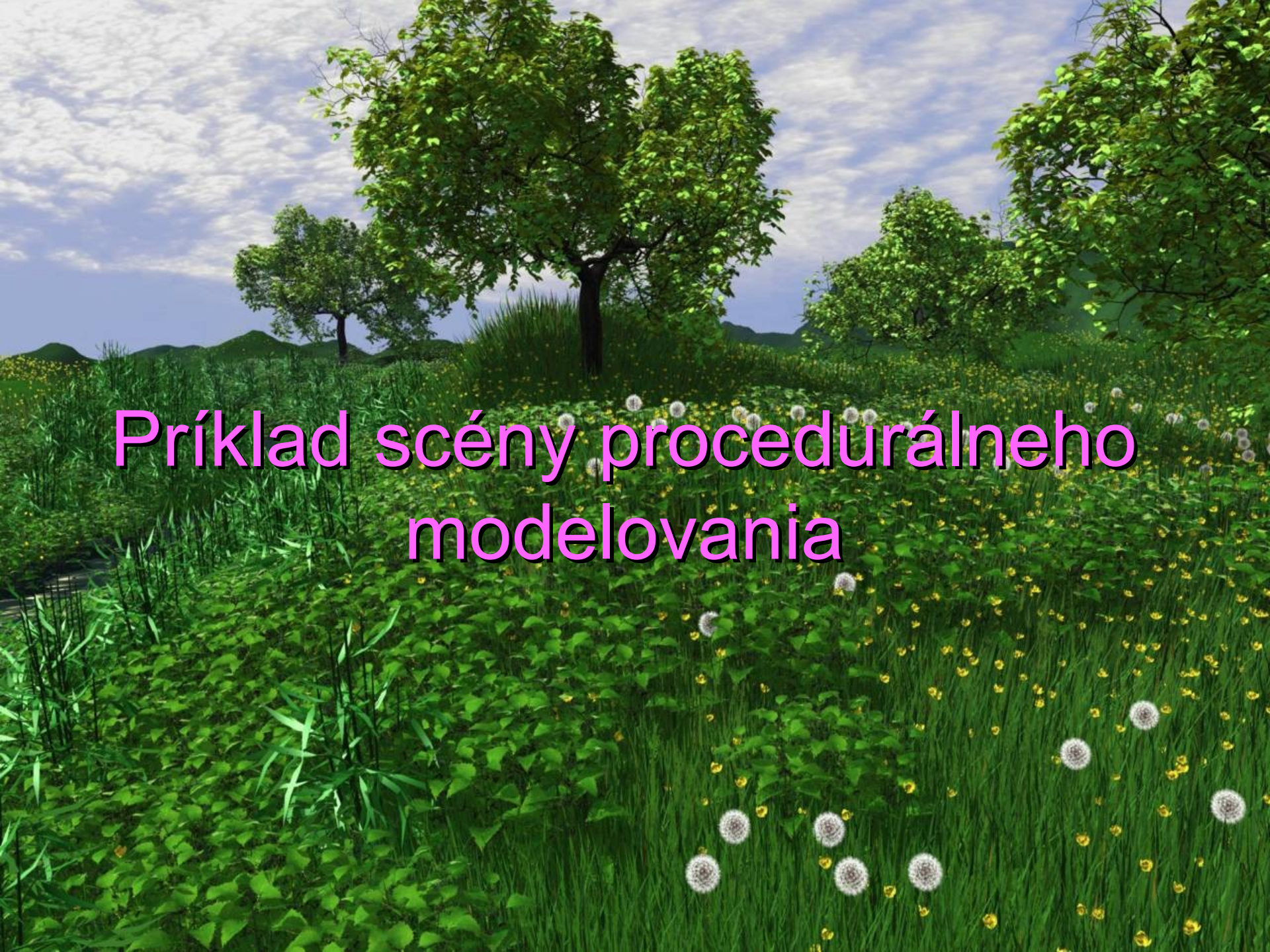





L-System Parser/Mutator by Lj Lapre
Original form
After 5 mutations
After 188 mutations






Príklad scény procedurálneho
modelovania

Témy prednášok

- Náhodné a pseudonáhodné generátory
- Geometricko-procedurálne modelovanie
- Šumy a turbulencie
- Fraktály
- Modelovanie terénu a miest
- Reakčná difúzia
- Celulárne automaty
- Procedurálne textúry, procedurálne tieňovanie
- Časticové systémy
- Genetické, evolučné algoritmy a programovanie

Náhodné čísla

- Základ stochastických algoritmov/modelov/procesov
- Sú dôležitá súčasť výpočtov v:
  - modelovaní a simuláciách
  - počítačovej grafike
  - kryptografii a dátovej komunikácií
  - numerickej analýze
  - rozhodovaní
  - lotériach a výherných automatoch
  - ...

Generovanie náhodných čísel

- Skutočné náhodné generátory (hardvérové, TRNG)
- Pseudonáhodné generátory (softvérové, PRNG)
- Kvázináhodné generátory (softvérové, QRNG)
- **Hardvérové** generátory využívajú hodnoty niektorých veličín vo fyzikálnych dejoch:
  - elektronický šum
  - rozpad rádioaktívnych častíc (kvantová mechanika)
  - fotoelektrický jav (kvantové vlastnosti fotónov)
  - atmosferický šum (blesky, rádiové vlny)
- **Softvérové** generátory používajú nejaký algoritmus či rekurentný vzťah na výpočet postupnosti náhodných alebo kvázináhodných postupností čísel
- **Tabuľka** vygenerovaných náhodných čísel

Výhody a nevýhody SW/HW generátorov

- Softvérové generátory:
  - \+ jednoduchá realizácia ⇒ rýchly výpočet
  - \+ výhoda opakovateľnosti
  - \+ možnosť využitia 1. čísla = semienka
  - \+ možnosť generovania hodnôt podľa potreby/rozdelenia (rovnomerné, Gaussovo, kvázináhodné rozdelenie)
  - – determinizmus (možnosť predpovedania)
  - – konečný počet prvkov = po čase sa začnú čísla opakovať (konečná perióda)
- Hardvérové generátory:
  - \+ skutočne náhodné čísla
  - \+ nemožnosť predpovedania, neopakujú sa
  - – náročná realizácia ⇒ pomalý výpočet
  - – nutnosť prevedenia fyzikálnej veličiny na číselnú hodnotu

Náhodné čísla a postupnosti

- Generujú sa nielen čísla ale aj náhodné postupnosti objektov (písmená, pixely, obrázky či geometrické objekty,...)
- Riadenie algoritmu výpočtu generátora:
  - Dopredu dané semienko (seed) = 1. číslo z postupnosti
  - Funkcia náhodnosti generuje z daného náhodného čísla nové náhodné číslo, zvyčajne z intervalu $\langle 0,1)$ s normálnym/rovnomerným rozdelením (závisí od typu generátora)

História generátorov náhodných čísel

- Prehistorická: hádzanie mincou, kockou, …
- L. H. C. Tippet - anglický štatistik (1927):
  tabuľka so 40 000 náhodnými číslami zo sčítania ľudu
- RAND Corporation (1955):
  tabuľka s 1 000 000 náhodnými číslami
- Fyzikálny generátor náhodných čísel ERNIE (1957):
  - bol vytvorený pre britskú lotériu
  - zosilnenie impulzov výstupného elektrického prúdu prechádzajúceho neónovou trubicou
- 50' - tabuľka náhodných čísel v pamäti počítača

  ...
- ERNIE 5 (2019) využíva kvantový generátor náhodných čísel s využitím fotónov produkuje 10^6 čísel/hod

Vlastnosti generátorov

Dobrý generátor by mal spĺňať:

- dlhá perióda – v ideálnom prípade nekonečne dlhá, (generátor *Mersenne*), *Twister* dosahuje periódu $2^{19937}–1$
- rovnomerné a s rastúcou dĺžkou postupnosti čoraz lepšie zaplnenie intervalu
- opakovatelnosť = schopnosť opakovane generovať tú istú postupnosť pseudonáhodných čísel
- čas generovania je zanedbateľný oproti času operácií, ktoré vytvorenú sekvenciu používajú
- implementácia nenáročná na pamäť a vo vyššom programovacom jazyku
- generátor musí úspešne zdolať súbor štatistických testov

Prvý algoritmus generovania náhodných čísel

- J. von Neumann (1946) – metóda stredných rádov
- Výber n číslic z prostriedku druhej mocniny daného čísla
- Príklad výberu 10 číslic:
  - 1. číslo = 5 772 156 649
  - mocnina = 33 317 792 380 594 909 201
  - výber = 7 923 805 949
- Nevýhody:
  - cyklickosť
  - číslica 0 z čísla nikdy nezmizne

Rekurentný vzťah

- Predstavuje najpoužívanejšie riešenie
- Výstupné hodnoty sú generované na základe vzťahu:

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

  – prvé hodnoty $x_1, ..., x_k$ sú dané, ostatné pre $n > k$ sa vypočítajú na základe rekurentného vzťahu s dopredu známou funkciou $f()$ (môže byť aj veľmi komplikovaná
- Metóda stredných rádov používa funkciu $f(x_n) = x_{n-1}^2$ a celočíslené delenie a modulo pre výber prostredných cifier

Lineárna kongruencia

- D.H. Lerner (1948)
- Využíva definíciu kongruencie:
  - a a b sú kongruentné modulo m, ak majú po delení číslom m ten istý zvyšok:

    $a \equiv b \pmod{m}$
- Postupnosť pseudonáhodných čísel y_n s rovnomerným rozdelením zvyčajne generujeme v dvoch fázach:

  1) rekurentný vzťah $x_n = f(x_{n-1}, ..., x_{n-k})$
  - 2) výstupná hodnota $y_n = x_n / m$, $y_n \in \langle 0,1)$

Lineárne kongruenčné generátory (LCG)

- Postupnosť je generovaná vzťahom:

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

- Požiadavky pre dlhú periódu:
  - začiatočná hodnota $x_0 \geq 0$
  - koeficient $a \geq 0$
  - prírastok $c \geq 0$ musí byť nepárne číslo
  - modulo $m \geq x_n$, $m > c$, čo najväčšie možné
  - c a m sú nesúdeliteľné

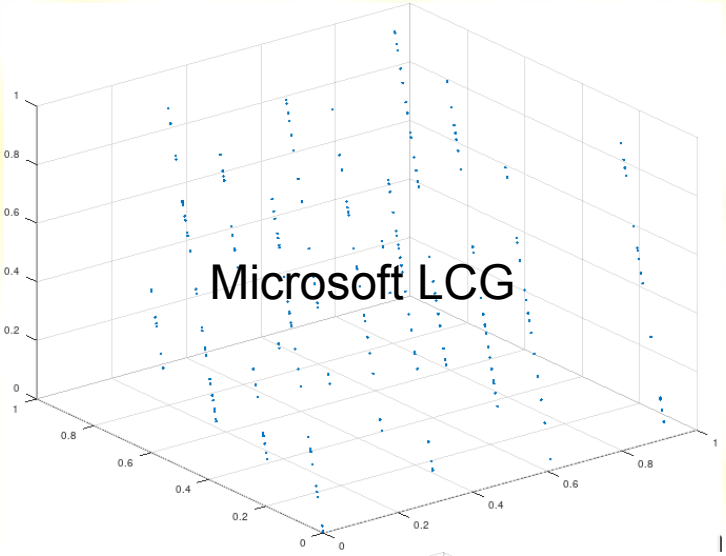
Známe generátory využívajúce lineárnu kongruenciu

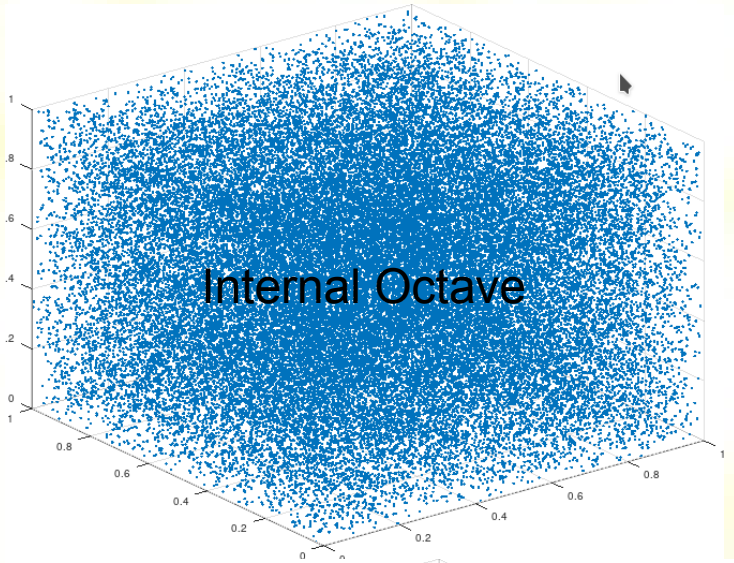
Názov, použitie LCG	m	a	c	X_0
RANDU –počítače IBM(1960)	2^{31}	65539	0	X_0
ANSI C – funkcia rand()	2^{31}	1103515245	12345	12345
Program DERIVE	2^{32}	3141592653	1	0
Jazyk SIMULA	2^{35}	5^{15}	0	1
ANSI C – funkcia drand48()	2^{48}	25214903917	11	0
Program MAPLE	10^{12}-11	427419669081	0	1
Program MATHEMATICA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

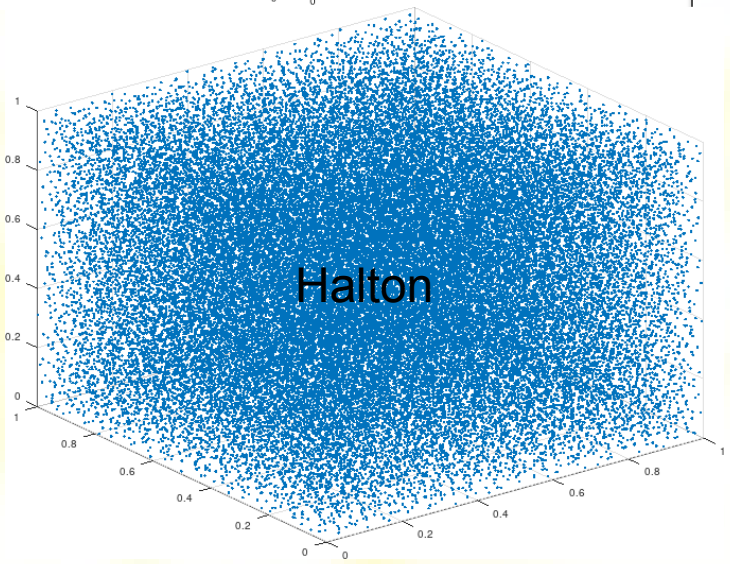
Spektrálne charakteristiky LCG

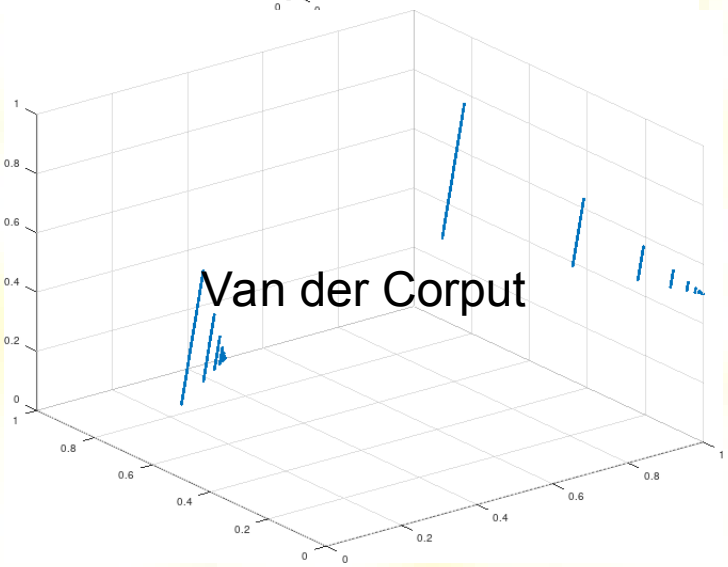
- Majme $\{x_i\}$ výstup z LCG a vytvorme N-tice:
  $S_n = [x_n, x_{n+1}, ..., x_{n+N-1}]$
- Použijeme ich ako súradnice bodov v N-rozmernom priestore a výsledkom je mriežková štruktúra pri vypĺňaní N-rozmerného intervalu, t.j. body S_n ležia na množine ekvidištančných paralelných hyperrovín
- Napriek tomu sú LCG generátory vo všeobecnosti najviac používané.

Spektrálne charakteristiky niektorých generátorov









Fibbonaciho generátory

- Používajú postupnosť $x_n = (x_{n-l} + x_{n-k}) \bmod m,\ l > k > 0$
- Často sa volí $m = 2^M$
- Vhodná voľba pre parametre je napr. $l = 65,\ k = 71,\ M = 32$
- Namiesto operácie + sa používajú aj operácie:

  „×“ – má lepšie vlastnosti, avšak kratšiu periódu
- „XOR“ – má horšie vlastnosti, avšak jednoduchší výpočet
- Pre výpočet potrebujeme posledných l hodnôt x_n
- Z l počiatočných hodnôt x_0, ... , x_{l-1} by mala byť aspoň jedna nepárna
- Generátor je veľmi citlivý na vstupné parametre l, k, M

Násobné rekurzívne generátory

- Používajú rekurentný vzťah:
  - $x_n = (a_1 x_{n-1} + \ldots + a_k x_{n-k} + b) \bmod m$, kde $k, m > 0, b \geq 0$
- Z dôvodov efektívnosti sa často navrhuje použiť iba dva nenulové koeficienty a_i a $b = 0$
- Špeciálny prípad sú tzv. *Linear Feedback Shift Register* generátory, kde $a_i = \{0, 1\}$ a $m = 2$

Nelineárne generátory

- Vo všeobecnosti majú lepšie autokorelačné vlastnosti ako lineárne generátory
- Lepšie spektrálne vlastnosti (výstupné hodnoty nemajú mriežkovú štruktúru)
- Lepšie kryptografické vlastnosti
- Nevýhodou je, že sú pomalšie
- Najčastejšie sú používané inverzné kongruenčné generátory

Inverzné kongruenčné generátory

- Nech je dané prvočíslo m *a celé číslo* x, potom $x^{m-1} \equiv 1 \bmod m \Rightarrow x^{m-2} \equiv x \bmod m$. Označme inverzný prvok $x^{-1} = x^{m-2} \bmod m$ a nulový prvok $0^{-1} = 0$
- Rekurentný vzťah pre výpočet nasledujúceho náhodného čísla:

$$x_n = \left(a x_{n-1}^{-1} + c\right) \bmod m$$

alebo

$$x_n = \left(a\left(n + n_0\right) + c\right)^{-1} \bmod m$$

- Inverzný prvok sa počíta inverzným Euklidovým algoritmom na nájdenie celočíselných riešení rovnice

$$x^{-1} \cdot x + k \cdot m = 1$$

- Majú lepšie autokorelačné vlastnosti ako LCG

Mocninové generátory bitov

- Vhodné najmä na kryptografické účely
- Najznámejší je tzv. RSA (Rivest–Shamir–Adleman) generátor
- Používa sa súčin m dopredu daných veľkých prvočísel p, q $m = p \cdot q$ a náhodne zvolené kladné celé číslo b, ktoré nemá spoločného deliteľa s $(p - 1)(q - 1)$
- Generátor je určený rekurentným vzťahom:

$$x_n = x_{n-1}^{b} \bmod m, \text{ kde } x_0 \in \langle 1, m-1 \rangle$$

- Výstupom je najmenej významný bit $b_n = x_n \bmod 2$

Blum Blum Shub generátor bitov

- Podobný ako RSA
- Generátor je určený rekurentným vzťahom:

  $x_n = x_{n-1}^2 \bmod m$, kde $x_0 \in <1, m-1>$ je nesúdeliteľné s $m = p \cdot q$, pričom p a q je možné vyjadriť v tvare $4k + 3$ $(\equiv 3 \bmod 4)$
- Výstupom je paritný bit alebo niekoľko (napr. 1) najmenej významných bitov
- Pre výsledný bit je možné použiť aj vzťah $b_n = x_n \cdot z \bmod 2$, kde $x_n \cdot z$ predstavuje skalárny súčin po bitoch s náhodnou bitovou maskou z
- x_0 by malo byť celé číslo, ktoré je prvočíslo k m
- Generované bity sú takmer neodlíšitelné od postupnosti skutočne náhodných bitov

Metódy zlepšenia vlastností generátorov

- Vlastnosti už existujúcich generátorov, môžeme zlepšiť:
  - **aritmetickým skladaním** (kombinovaním) výstupov z viacerých generátorov, napr:

    $z_n = (x_n + y_n) \bmod m$, pričom je vhodné zvoliť nesúdeliteľné veľkosti periód jednotlivých generátorov
  - **premiešavaním** (*shuffling*) jednej alebo viacerých postupností náhodných čísel

Verzie premiešavania

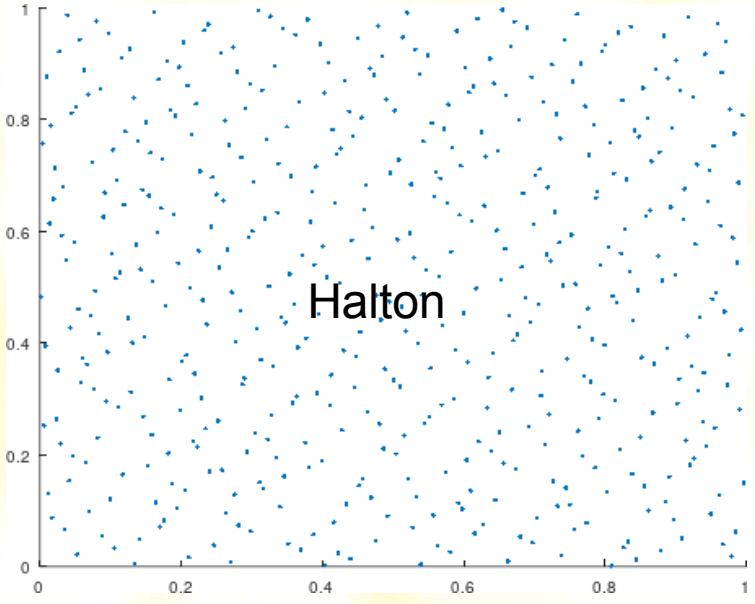
- Prvých k vstupných hodnôt x_k uložíme do poľa. Potom z každej ďalšej vstupnej dvojice náhodných čísel pomocou prvého čísla náhodne vyberieme hodnotu z poľa, ktorá bude našim výstupom, druhé číslo vložíme na uvoľnené miesto
- Prvých k hodnôt x_k uložíme do poľa a $(k+1)$-tu hodnotu do pomocnej premennej 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 (z jednej vstupnej hodnoty dostaneme jednu hodnotu výstupnú)

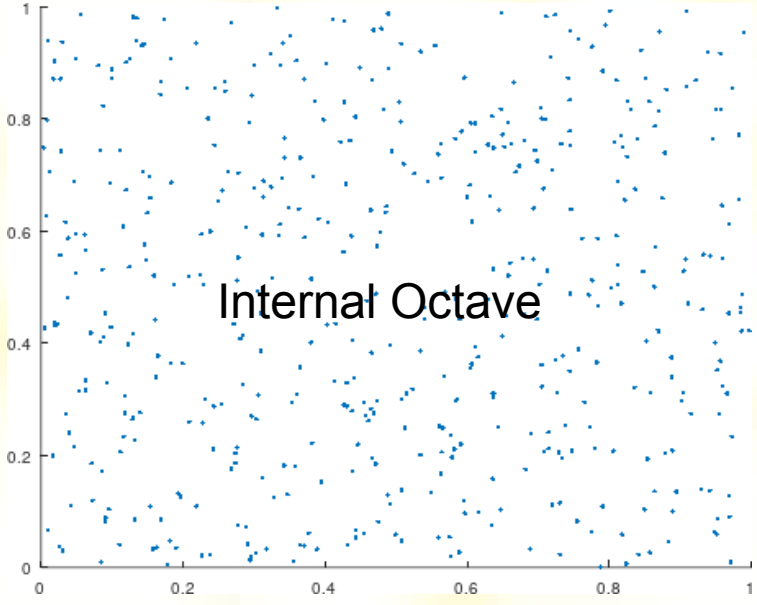
Príklad premiešavania

- D. MacLaren, G. Marsagliu (1965)
- Dané dva LCG pre $\{x_i\}$ a $\{y_i\}$
- Vytvoríme pole V[0],V[1], …, V[$k-1$], $k \sim 100$ náhodných čísel
- Algoritmus:
  1) vygenerujú sa dve nasledujúce x, y hodnoty z $\{x_i\}$, $\{y_i\}$
  2) vypočíta sa index $j = k \cdot \mathrm{y} / m$, pričom m je modulo z postupnosti $\{y_n\}$
  3) výstupné náhodné číslo je V[j], potom prehoď V[j] s x_n
  4) opakuj od kroku 1)

Kvázináhodné postupnosti čísel

- Postupnosti, ktoré rovnomerne vypĺňajú daný interval
- Používajú iba také algoritmy, ktoré rovnomerne zapĺňajú interval už od začiatku postupnosti, t.j. interval je v každej iterácii vyplnený danými hodnotami rovnomerne
- Platí, že ľubovoľná podsekvencia zaplní interval približne rovnako rovnomerne ako iná rovnako dlhá podsekvencia
- Príkladom sú Haltonove Van der Corputove postupnosti





Haltonove postupnosti čísel

- Generuje reálne čísla v intervale $\langle 0,1)$
- Postupnosť je konštruovaná podľa deterministickej metódy, ktorá ako základ používa prvočísla:
  1) zvolí sa prvočíselný základ sústavy b (napr. 2)
  2) pre k-ty člen sa prevedie hodnota k do sústavy o základe b (napr. pre k=10, b=2 je výsledok 1010)
  3) získané číslo (o základe b) sa zapíše za desatinnú čiarku v opačnom poradí a prevedie sa na reálne číslo (napr. 1010 → 0.0101 → 0.3125)
- Pre generovnie N-tice treba použiť v jednotlivých súradniciach ako základy iné prvočísla (napr. 2, 3, 5)

Rovnomerné rozdelenie na intervale

- Máme daný interval $\langle a,b)$

1) Generujeme reálne čísla z intervalu $\langle 0,1)$
2) Použijeme jednoduchý transformačný vzťah:

$$y_n = a + (b - a) \cdot x_n$$

Testy postupností náhodných čísel

- Cieľom testov je zistiť, či sa generované postupnosti správajú dostatočne náhodne
- Môžeme ich rozdeliť na **teoretické** a **empirické**
- Používajú známe štatistické testy, napr. χ^2 test či Kolmogorov – Smirnov test, spektrálny test, test diskrepancie, test na autokoreláciu
- Existuje veľa vlastností, ktoré môžeme testovať, avšak postupom času sa niektoré ukázali ako užitočné (napr. test rovnomerného rozdelenia)

Chi-kvadrát test

- Testujeme hypotézu či postupnosť pseudonáhodných čísel $\{x_i\}$ má rovnomerné rozdelenie
- Algoritmus:
  1) Vyberieme ľubovoľných n za sebou nasledujúcich čísel $x_1, x_2, \ldots, x_n$, ktoré rozdelíme do k disjunktných tried
  2) Počty prvkov v jednotlivých triedach označíme z_j
  3) Označme $p_j = \mathrm{P}(\text{náhodné číslo je z } j\text{-tej triedy})$
  4) Využijeme rozdiel reálnych počtov prvkov z_j a očakávaných $n \cdot p_j$ počtov prvkov v triedach:

$$V = \sum_{j=1}^{k} \frac{\left(z_j - n\,p_j\right)^2}{n\,p_j}$$

Chi-kvadrát test

- Hodnotu V testujeme na zvolenej hladine testu $\alpha \in <0.01, 0.05>$ pomocou kritických hodnôt $\chi^2_{k-1}(\alpha)$
- Pseudonáhodné čísla **nemajú** rovnomerné rozdelenie, ak platí:

$$V \leq \chi^2_{k-1}(\alpha/2) \text{ alebo } V \geq \chi^2_{k-1}(1-\alpha/2)$$

- Pri overovaní rovnomerného rozdelenia postupnosti pseudonáhodných čísel sa používa $k = 10\text{-}100$, postupne pre rôzne úseky generovanej postupnosti

Kolmogorov-Smirnov test

- Pri tomto type testu nerozdeľujeme hodnoty testovanej postupnosti $\{x_i\}$ do tried, ale testujeme zhodu (rozdiel) empirickej distribučnej funkcie $F_n(x)$ s očakávanou (teoretickou) distribučnou funkciou $F(x)$
- Algoritmus:
  1) Z hodnôt $\{x_i\}$ vytvoríme distribučnú funkciu:
     $F_n(x)$ = (počet prvkov $\{x_i\}$, pre ktoré $x_i < x$) / n
  2) Vypočítame dve čísla:
     $$K_n^+ = \sqrt{n} \max_{1 \le i \le n} \left\{ \frac{i}{n} - F_n\left(x_{(i)}\right) \right\} \quad K_n^- = \sqrt{n} \max_{1 \le i \le n} \left\{ F_n\left(x_{(i)}\right) - \frac{i-1}{n} \right\}$$
     kde $\{x_{(i)}\}$ je usporiadaná postupnosť z prvkov $\{x_i\}$
  3) Nájdeme $K_n = \max\left\{ K_n^-, K_n^+ \right\}$
  4) Hodnotu K_n testujeme na zvolenej hladine testu $\alpha \in \langle 0.01, 0.05 \rangle$ pomocou kritických hodnôt $\mathrm{KS}_n(\alpha)$

Empirické testy

- Existujú testy pre náhodné čísla z intervalu $\langle 0,1)$ alebo pre celočíselné postupnosti (*DIEHARD*, *NIST*)
- V prípade celočíselných testov môžeme desatinné čísla transformovať pre konkrétnu max. hodnotu m:

$$y_n = \lfloor m\, x_n \rfloor$$

- **Sériový** test – z $\{x_i\}$ vytvoríme N-tice, ktoré tvoria body S_n v N-rozmernom priestore a aplikujeme na ne χ^2 test
- Test **minimálnej vzdialenosti** – N krát sa testujú minimálne vzdialenosti d^2 medzi n náhodne vybranými bodmi S_n. Vo výsledku sa otestuje sa či N hodnôt d^2 má rovnomerné rozdelenie

Empirické testy

- Test **bitového prúdu** – na testovanú postupnosť sa pozeráme ako na prúd bitov, z ktorého vyberáme prekrývajúce sa slová o veľkosti 20 bitov. Počítame počty chýbajúcich slov a otestujeme či tieto počty majú normálne rozdelenie
- Existuje množstvo ďalších empirických testov – *Poker test*, *Coupon Collector's test*, *Permutation test*, *Run test*, *Maximum of t test*, ...