VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta Informačních Technologií

MODELOVÁNÍ A SIMULACE

2009/2010

Semestrální projekt

Implementace knihovny pro generování pseudonáhodných čísel

Autoři:
Kubiš Radim, xkubis03@stud.fit.vutbr.cz
Kunčar Petr, xkunca04@stud.fit.vutbr.cz

Obsah

1 Úvod **3**
1.1 Zadání 3
1.2 Analýza zadání 3

2 Implementovaná rozložení **4**
2.1 Rovnoměrné rozložení 4
2.2 Exponenciální rozložení 5
2.3 Normální rozložení 6
2.4 Rayleighovo rozložení 7
2.5 Obecné Paretovo rozložení 8
2.6 Poissonovo rozložení 9

3 Použité nástroje **10**

4 Publikační zdroje **10**

5 Popisy rovnic **10**

6 Závěr **11**

Úvod

1.1 Zadání

Naprogramujte knihovnu pro generován pseudonáhodných čísel. Knihovna musí obsahovat minimálně generátor čísel řídících se rovnoměrným rozložením, exponenciálním rozložením a normálním rozložením. Nad tyto tři vytvořte ještě generátory dalších třech existujících rozložení (zdroj: Wikipedia). Vašimi generátory vygenerujte pro každý typ rozložení statisticky hodnotný soubor dat (např. 1000 čísel). Ze souboru vygenerujte histogram a ten porovnejte (stačí vizuálně v grafu) s analytickým vyjádřením hustoty pravděpodobnosti. To znamená, že pro 6 rozložení to bude 6 grafů v dokumentaci. V dokumentaci popište metody generování čísel dle vašich zvolených rozložení.

1.2 Analýza zadání

Ze zadání vyplývá, že budeme generovat pseudonáhodná čísla a porovnávat jejich grafy s analytickým řešením, jestli naše generátory se dostatečně přiblížili k danému řešení, tudíž jestli jsou v praxi použitelná. Pseudonáhodné generování je takové generování, které jde zalgoritmizovat, čímž se liší od náhodného generování, kde výsledné hodnoty jsou zcela náhodné. V našem případě nelze udělat generátor náhodných čísel, neboť počítače jsou nedeterministické. Náhodný jev (proměnná) v takovém generování, který předem není znám, v našich generátorech tvoříme pomocí času a počtu taktů procesoru. Generátor tvoříme, tak že první vytvoříme primární generátor (generátor s rovnoměrným rozložením). Pak hodnoty primárního generátoru transformujeme na hodnoty s požadovaným rozložením. Pro transformaci na jiná rozložení používáme metody inverzní transformace (exponenciální rozložení) a vylučovací (zamítací) transformace (normálové rozložení). Testování našich generátorů jsme vždy prováděli na vzorku 1000 čísel.

Implementovaná rozložení

2.1 Rovnoměrné rozložení

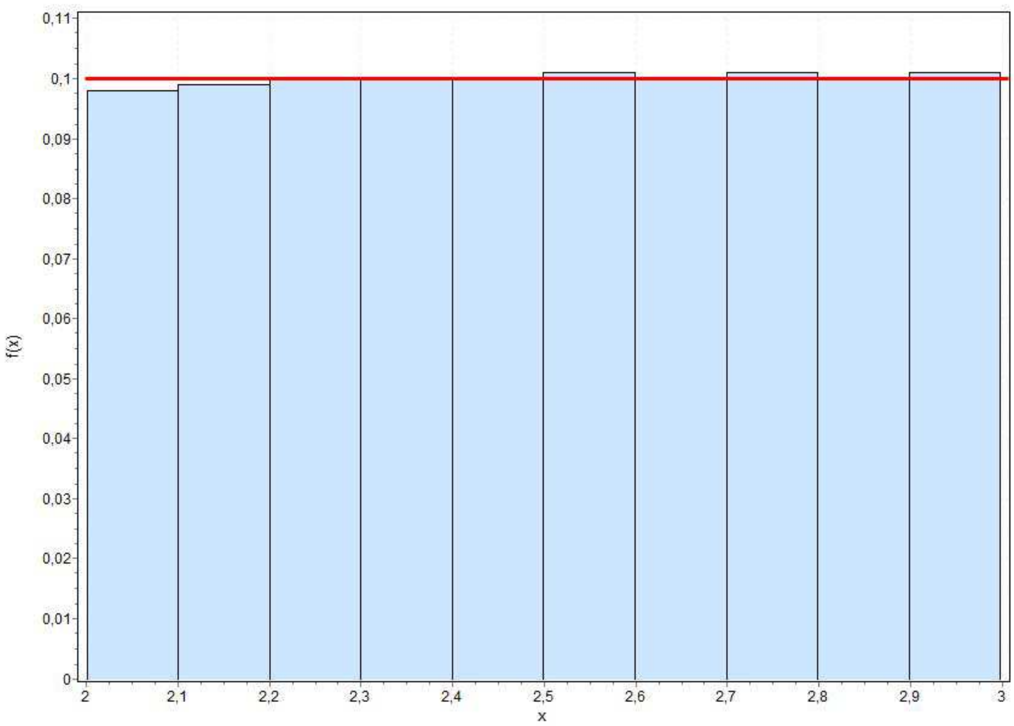
Pro rovnoměrné rozdělení jsme použili lineární kongruentní metodu generování pseudonáhodných čísel. Využili jsme na to vzorec:

$$x_{n+1} = (ax_n + b) \bmod m \quad \{1\}$$

kde a, b, m jsou konstanty. Problémem těchto generátorů je správně zvolit konstanty.
My jsme zvolili konstanty podle materiálů k předmětu IMS a to takto: a = 69069, b = 1, m = 232. Tento vzorec nám vygeneroval číslo v rozsahu od 0 do 1. Pro převod čísla do intervalu zvoleného uživatelem jsme použili jednoduchý převod:

$$od + (do\text{-}od) * vygenerovaná\ hodnota \quad \{2\}$$

Dále jsme vygenerovali příslušný histogram. Hodnoty vstupu byl interval od 2 do 3. Histogram jsme zakreslily do grafu a porovnali s vyjádřením hustoty pravděpodobnosti pro rovnoměrné rozložení.



Obr. a) Histogram vzorku vygenerovaných čísel
a analytické vyjádření hustoty rovnoměrného rozložení

Z grafu vyplývá, že náš algoritmus na vygenerování hodnot s rovnoměrným rozložením pracuje správně. Proto nejspíš zvolené hodnoty parametru byli správné.

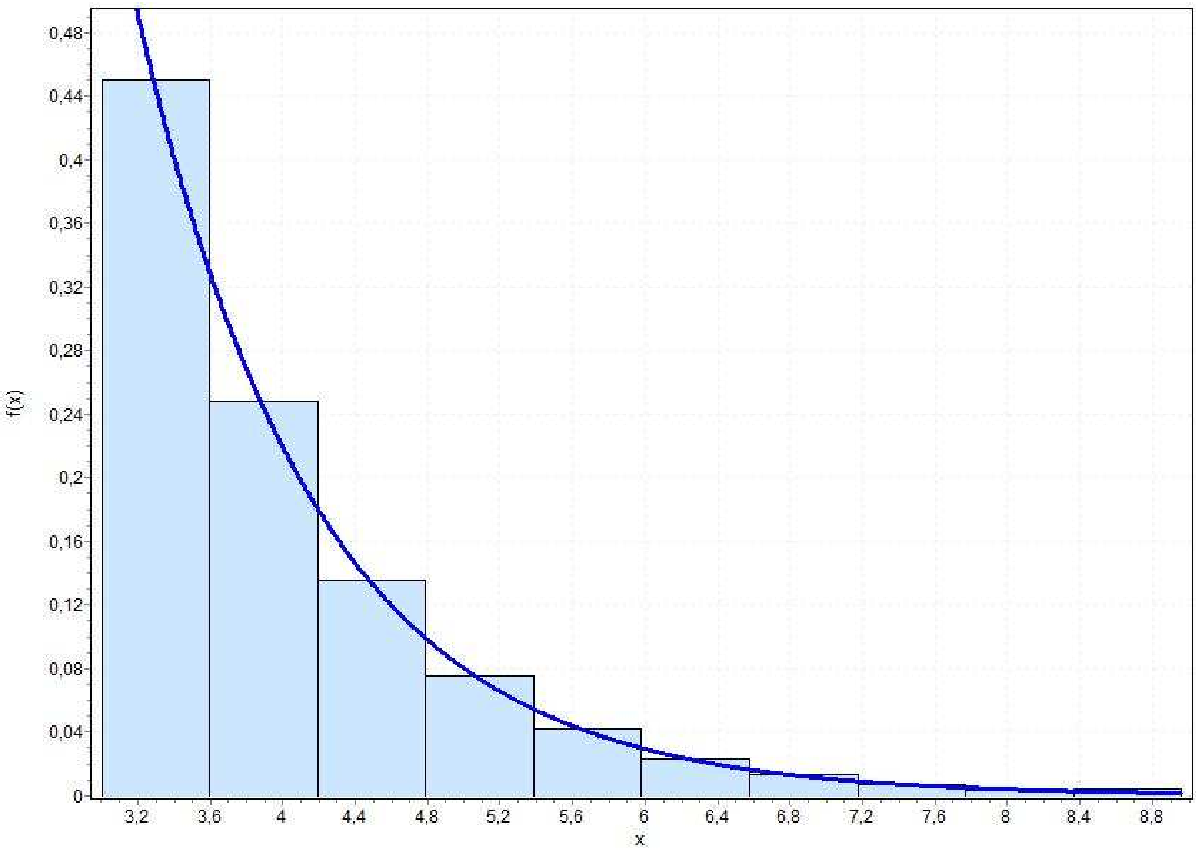
Funkce v programu: MyRandom – rozmezí 0-1; MyRandoms(od, do)

2.2 Exponenciální rozložení

Pro exponenciální rozložení pravděpodobnosti jsme použili metodu inverzní transformace. Kdy hodnotu vygenerovanou rovnoměrným generátorem transformujeme pomocí inverzní distribuční funkce, prohodí se obory hodnot, na vzniklou hodnotu distribuční funkce exponenciálního rozložení. Nevýhoda této metody je, že ne ke každému rozložení lze určit efektivně inverzní funkci. Postup je následující. Vygenerujeme náhodnou veličinu x s rovnoměrným rozdělením. Tuto veličinu pomocí vzorce:

$$y = \gamma - \lambda * \log(1-x) \quad \{3\}$$

transformujeme na veličinu s exponenciálním rozložením. γ a λ v daném vzorci jsou parametry. Pro náš případ jsme je zvolili následovně: $\lambda = 1$, $\gamma = 3$. Vygenerovaný histogram má tedy tuto podobu:



Obr. b) Histogram vzorku vygenerovaných čísel
a analytické vyjádření hustoty exponenciálního rozložení

Z grafu vyplývá, že vygenerované hodnoty vyšli s mírnými odchylkami a generování je závislé na vstupních parametrech

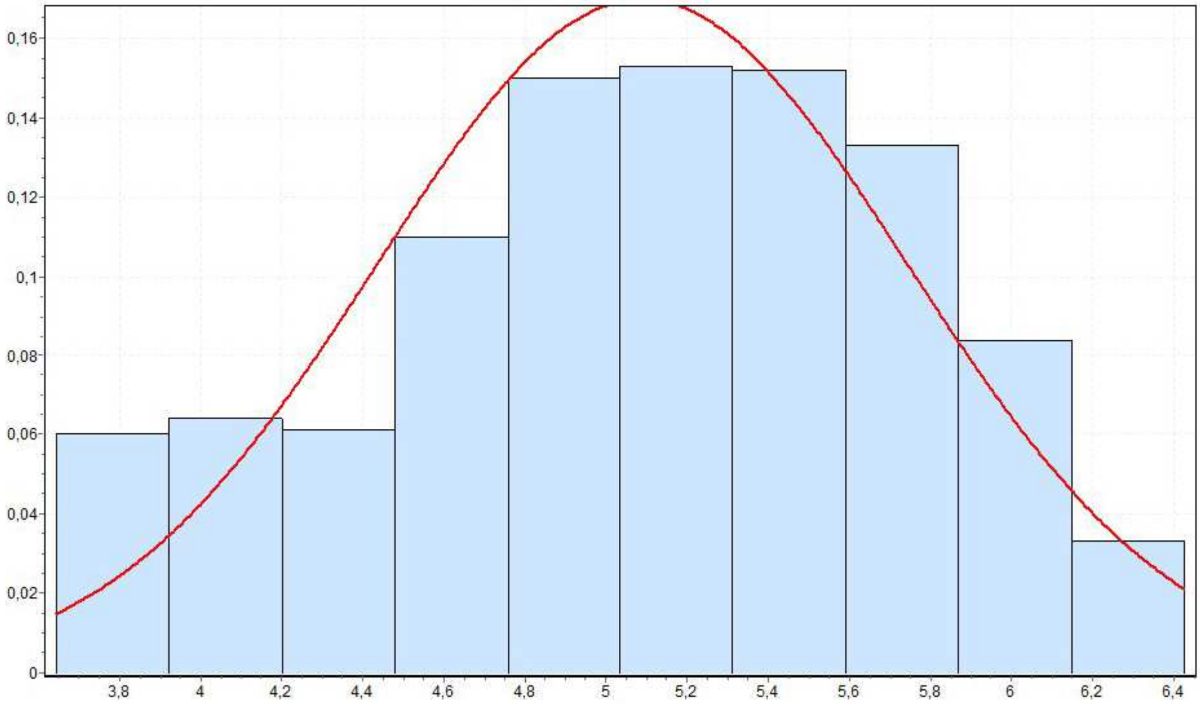
Funkce v programu: MyExpo(λ, γ)

2.3 Normální rozložení

Pro Normální rozložení pravděpodobnosti jsme použili vylučovací (zamítací) metodu. Tato metoda řeší problémy předchozí metody s nalezením inverzní distribuční funkce. Postupujeme, tak že danou plochu ohraničíme a náhodně vygenerované hodnoty s rovnoměrným rozložením porovnáváme, zdali spadají do funkce hustoty pravděpodobnosti. Pokud některá hodnota nespadá, vyloučím ji. Pro funkci hustoty pravděpodobnosti jsme použili vzorec z materiálů IMS:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \{4\}$$

kde μ je střední hodnota a σ je odchylka. Pro generování x je zvoleno generování s rovnoměrným rozložením s použitím pravidla 3 sigma. Toto pravidlo říká, že více jak 99% plochy, které vygenerovaná čísla zabírají je v rozsahu od μ-3σ do μ+3σ. Výsledný graf s porovnáním hustoty normálního rozložení:



Obr. c) Histogram vzorku vygenerovaných čísel
a analytické vyjádření hustoty normálníh rozložení

Z grafu vyplývá, že dané rozdělení neodpovídá úplně přesně analytickému řešení. Podařilo se nám vytvořit, zde jinou metodou generátor náhodných čísel s normálním rozložením. Daný generátor se řídil centrální limitní větou, je implementován (MyNormal2(střední hodnota, odchylka)) a odzkoušen. Pro graf jsme si vybrali méně zdařilí generátor schválně, protože jeho implementace odpovídá postupu implementace probírané na přednáškách předmětu IMS.

Funkce v programu: MyNormal(střední hodnota, odchylka); MyNormal2(střední hodnota, odchylka);

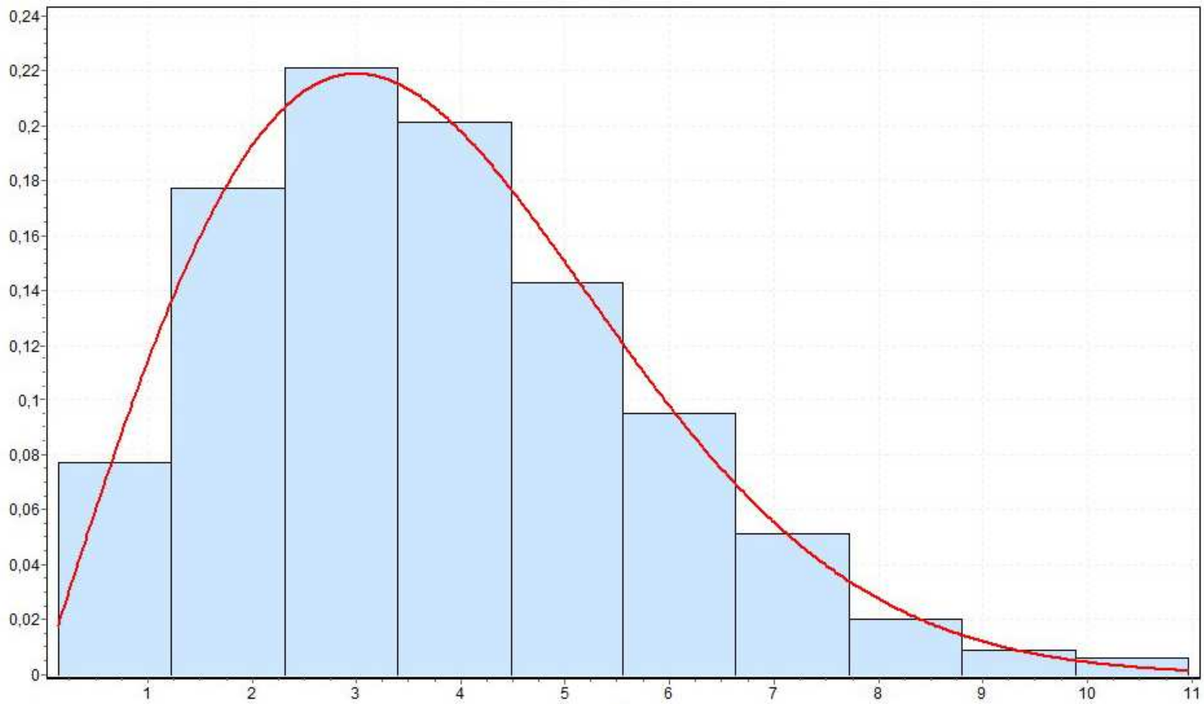
2.4 Rayleighovo rozložení

Rayleighovo rozložení je spojité rozložení pravděpodobnosti. Pro generování náhodných čísel s tímto rozložením se vychází z rovnoměrného rozložení pravděpodobnosti v intervalu (0, 1). Pokud chceme získat číslo X, které by se vygenerovalo s pravděpodobností podle Rayleighova rozložení, použijeme vzorec

$$X = \sigma\sqrt{-2\ln\left(1-U\right)} \quad \{5.\}$$

kde U je náhodné číslo vygenerované z rovnoměrného rozložení a parametr σ je modul Rayleighova rozložení. Modul je bod, ve kterém dosahuje rozložení svého maxima.

Graf:



Obr. d) Histogram vzorku vygenerovaných čísel
a analytické vyjádření hustoty Rayleighova rozložení

Zdroje:

http://en.wikipedia.org/wiki/Rayleigh_distribution

Funkce v programu: myRayleigh(sigma)

2.5 Obecné Paretovo rozložení

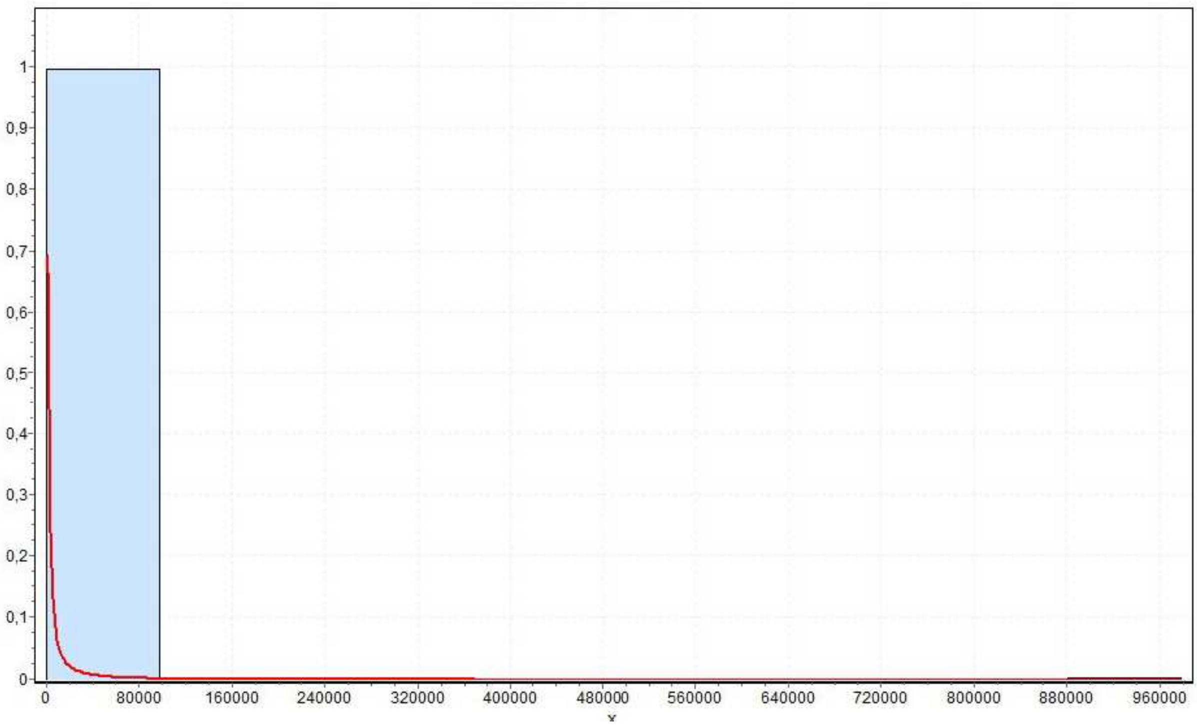
Obecné Paretovo rozložení pravděpodobnosti je spojité. Funkce

$$X = \mu + \frac{\sigma(U^{-\xi} - 1)}{\xi} \quad \{6.\}$$

pro generování náhodných čísel s tímto rozložením má tři parametry:

- µ – dolní hranice generovaných čísel
- σ – měřítko
- k – ovlivňuje tvar funkce

Graf:



Obr. e) Histogram vzorku vygenerovaných čísel
a analytické vyjádření hustoty obecného Paretova rozložení

Zdroje:

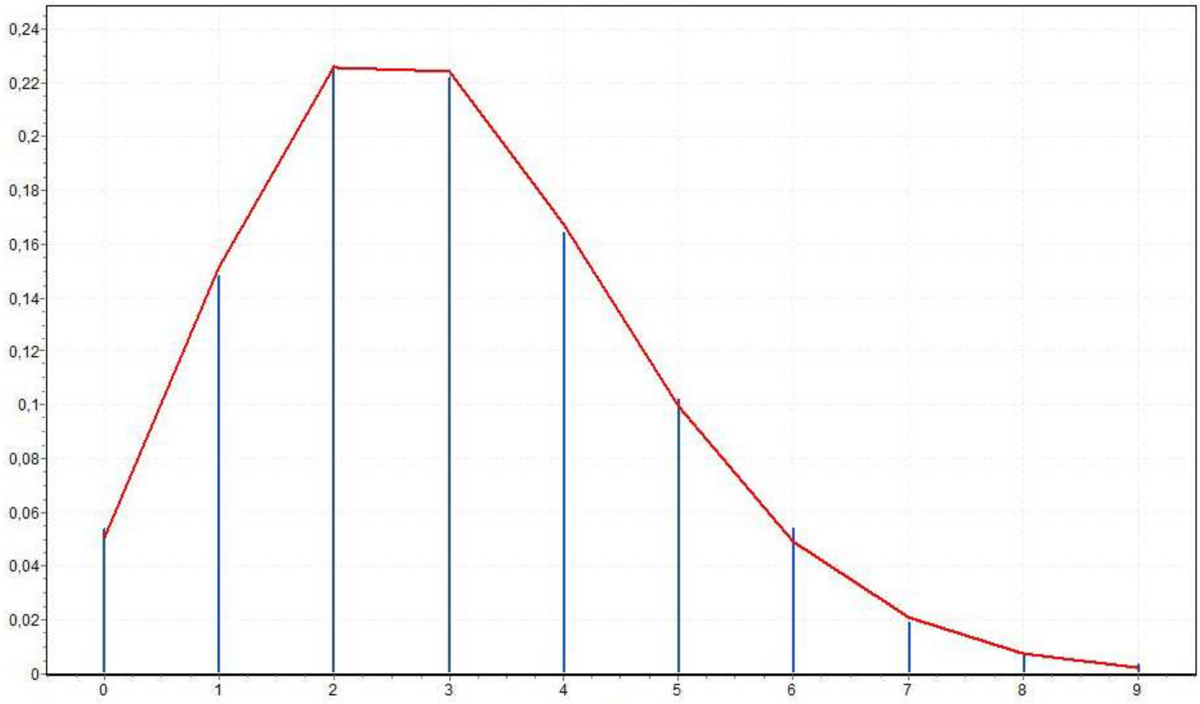
http://en.wikipedia.org/wiki/Pareto_distribution

<u>Funkce v programu</u>: myPareto(my, sigma, ksi)

2.6 Poissonovo rozložení

Jediné implementované diskrétní rozložení pravděpodobnosti. Opět vychází z rovnoměrného rozložení v intervalu (0, 1). Náhodná čísla s tímto rozložením lze generovat funkcí, která má jako jediný parametr λ, tedy bod maxima Poissonova rozložení. Algoritmus pro generování těchto čísel lze jen těžko zapsat matematicky.

Graf:



Obr. f) Histogram vzorku vygenerovaných čísel
a analytické vyjádření hustoty Poissonova rozložení

Zdroje:

http://en.wikipedia.org/wiki/Poisson_distribution
http://everything2.com/title/Generating+random+numbers+with+a+Poisson+distribution

<u>Funkce v programu</u>: myPoisson(lambda)

3. Použité nástroje

Programovací jazyk C

Microsoft Excel – pro kreslení histogramu

EasyFit – program pro zpracování dat

4. publikační zdroje

Modelování a simulace náhodných jevů (Jaroslav Pelikán)

Odkaz: http://fast10.vsb.cz/science/konf-2002-04-10/prednasky/22-Pelikan.pdf

Bakalářská práce (Radek Hetnerovič)

Odkaz: https://stag-ws.zcu.cz/ws/services/rest/kvalifikacniprace/downloadPraceContent?adipIdno=31643

Generátor náhodných čísel v jazyce C (Martin Pergel)

Slajdy z přednášek k předmětu IMS

5. popisy rovnic

1. pro kongruentní metodu generování
2. převod hodnoty do daného intervalu
3. pro generování s exponenciální rozložením
4. pro generování s normální rozložením
5. pro generování s Rayleighovo rozložením
6. pro generování s obecné Paretovo rozložením

5. Závěr

Vypracování projektu bylo zajímavé, zejména porovnávání vygenerovaných hodnot s analytickým řešením. Narazili jsme i na zajímavý problém, kdy jsme se pokoušeli naprogramovat obdélníkové rozložení pravděpodobnosti, tak s naším generátorem pro rovnoměrné rozložení, který zde byl použit, graf funkce vypadal nepřijatelně. Po použití standardního generátoru v programovacím jazyku C, graf vypadal přijatelně. Byť náš generátor pro rovnoměrné rozložení podle grafů, kterých jsme generovali nespočetně mnoho pro různé argumenty a konstanty, vychází, že je implementován správně.

Zde by se tedy dalo uvažovat o tom, že generátor v standardní knihovně C generuje lépe náhodná čísla, ovšem, dle dokumentu „Generátor náhodných čísel v jazyce C“ od Martina Pergela jsou tyto generátory opticky špatně. Bylo to dokázáno na testech DIEHARD.