Knihovna pro generování pseudonáhodných čísel

Autor: bc. Pavel Novotný

1 Úvod

Při simulacích se často používají náhodné jevy či procesy, neboť některé části modelů jsou neurčité nebo je neumíme popsat jinak. Jedná se například o popisy příchodů (např. zákazníků) v systémech hromadné obsluhy, výskytu poruch nebo katastrof, určení doby obsluhy či doby životnosti nějakého zařízení. Především pro tvorbu simulačních modelů je tedy potřeba nástroj, který v průběhu simulace zajistí požadovanou náhodnost a to pokud možno rychle a přesně. Právě takovým nástrojem je zde dokumentovaná knihovna pro generování pseudonáhodných čísel nabízející tvůrci simulačního modelu na výběr z několika rozložení pravděpodobnosti výskytu žádané náhody.

Generátor pseudonáhodných čísel je program, jehož výstupem je deterministicky a efektivně určená posloupnost čísel taková, že je statisticky k nerozeznání od náhodné posloupnosti čísel [15]. Cílem této knihovny je vytvořit implementaci generátoru pseudonáhodných čísel, takže bude snadno použitelná v simulačních modelech nebo jiných náhodnost požadujících programech.

1.1 Zdroje faktů

Problematika generování náhodnosti je poměrně dobře popsána a to jak samotné generování pseudonáhodných čísel v rovnoměrném rozložení, tak také transformace z rovnoměrného rozložení do jiných žádaných rozložení.

Co se týče generování pseudonáhodných čísel, byly díky použitému generátoru s názvem Mersenne Twister hlavním zdrojem informací odborný článek *Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator* [8] a osobní stránka jeho tvůrce, profesora Makoto Matsumota [7].

Transformace mezi pravděpodobnostními rozloženími si vyžádaly více faktických zdrojů, ovšem tím stěžejním byla kniha *Numerical Recipes* [13], která k tématu poskytla obsáhlé, ba až vyčerpávající informace. V některých případech bylo nahlédnuto do odborných vědeckých článků, na něž bude ve vhodnou dobu upozorněno.

2 Rozbor tématu a použitých metod

Jak už mohlo vyplynout z úvodu, dá se knihovna dekomponovat na dva dílčí podproblémy. Jedním z nich je generování pseudonáhodných čísel v rovnoměrném rozdělení pravděpodobnosti, druhým pak transformace rovnoměrného rozdělení pravděpodobnosti na jiná rozdělení. Zmíněnou dekompozici bude sledovat i struktura této kapitoly.

2.1 Generování čísel v rovnoměrném rozdělení

V teorii modelování na simulací se normalizované rovnoměrné rozdělení bere jako základ pro generování dalších rozdělení. Dobrá knihovna pro generování pseudonáhodných čísel by tedy měla implementovat zvláště dobře právě generátor pro toto rozdělení, protože na něm závisí i kvalita čísel vygenerovaných pomocí jiných rozdělení pravděpodobnosti. Přitom *zvláště dobře* implementovaný (či navrhnutý) zde znamená rychlý, s co nejdelší periodou (počet čísel, po kterých se posloupnost začne opakovat) a se statisticky nezávisle generovanou posloupností.

Algoritmů pro generování pseudonáhodných čísel v rovnoměrném rozdělení je celá řada, avšak ne všechny mají *zvláště dobré* vlastnosti. V souvislosti se *zvláště dobrými* algoritmy se v poslední dekádě mluví především o algoritmu Mersenne Twister, známým též jako MT19937 [8], který nejenže má velmi velkou periodu $2^{19937}–1$, ale je také velmi rychlý a to i v porovnání s výpočetně jednoduchým lineárně kongruentním generátorem (oproti němu je však paměťově náročnější) [8]. Přes jistou nevýhodu, kterou je jeho nepoužitelnost pro kryptografické

účely (pokud získáme posloupnost určité délky, můžeme odvodit zbytek) [8], byl tento algoritmus vybrán, neboť knihovna najde své použití spíše při tvorbě simulačních modelů, kde zmíněný nedostatek nevadí.

Jako alternativa k algoritmu Mersenne Twister se nabízel algoritmus *Complimentary-multiply-with-carry*, od výzkumníka na poli generátorů pseudonáhodných čísel George Marsaglii (též autor tzv. Diehard testů). Jeho algoritmus má větší periodu (2^{131086}), je rychlejší, ale paměťově náročnější [6]. Přestože byl vybrán Mersenne Twister (pro lepší dohledatelnost a větší známost), knihovnu lze jednoduše přizpůsobit i pro CMWC algoritmus, o čemž je pojednáno dále.

2.1.1 Princip fungování algoritmu Mersenne Twister

Název Mersenne Twister navrhl prof. Matsumotovi prof. Knuth při jejich setkání v Japonsku, kdy byl tento algoritmus ještě poměrně nový [9]. Tento název se vztahuje k tzv. Mersennovým prvočíslům, což jsou prvočísla, která jsou o jednotku menší než nějaká celočíselná mocnina dvou, tj. $M_p = 2^p - 1$. Mersenne Twister používá v pořadí 24. Mersennovo prvočílo 2^{19937}–1 jako svou periodu. Číslo 19937 se v algoritmu objevuje i jako délka (v bitech) tzv. semínka, které se v implementaci ukládá do 624 prvků dlouhého pole 32-bitových slov. Jednoduchým vynásobením čísel 624 a 32 lze zjistit, že 31 bitů z celého pole zůstane prázdných. Tyto prázdné bity se v poli přemisťují a jsou využity pro výpočet následující hodnoty semínka, k čemuž slouží lineární posuvný registr se zpětnou vazbou. Ten je za určitých podmínek schopen sám o sobě generovat pseudonáhodná čísla, přičemž, je-li jeho délka rovna exponentu některého z Mersennových prvočísel, získá maximální možnou periodu.

Algoritmus má dvě výpočetní fáze. První fáze je rekurentní změna určitého 19937-bitového stavu na stav následující. Ve druhé fázi se nový stav transformuje na několik 32-bitových čísel.

Teoreticky je změna stavu posuvem zpětnovazebného registru, ovšem prakticky se jedná o průchod již zmíněným 624-prvkovým polem 32-bitových slov. Mersenne Twister patří mezi parametrizované generátory a má 11 definovaných parametrů. Průchod polem je rozdělen na tři část, přičemž jsou to právě některé z parametrů, které určují, jak velké tyto části jsou. Při průchodu se bity v registru mění pomocí bitových operací AND, OR a XOR.

Následná extrakce 32-bitových slov je postupný výběr jednoho z $624 - 1$ prvků registru (poslední se zahazuje). Vybraný prvek se ještě tzv. temperuje, tj. vylepšuje se jeho statistické k-rozložení. Temperování má čtyři parametrizované kroky, při nichž se opět provádí bitové operace (AND a XOR) a posuny.

V roce 2006 byla uvedena nová verze generátoru, SIMD-oriented Fast Mersenne Twister (SFMT). Za určitých okolností je až dvakrát rychlejší než původní algoritmus, má lepší rovnoměrné rozdělení a nastavitelnou periodu. V knihovně pro generování pseudonáhodných čísel se ovšem nachází pouze verze původní.

2.1.2 Generátor založený na celulárním automatu

K Mersenne Twister byl záhy přidán nový generátor čísel v rovnoměrném rozdělení pravděpodobnosti, který je založen na 256-stavovém celulárním automatu [12]. Jedná se o generátor, který vyniká rychlostí i kvalitou generovaných čísel. Více informací, včetně Diehard testů a srovnání s jinými generátory, lze dohledat na webové stránce tohoto generátoru zde [12]. C++ implementace generátoru je součástí této knihovny.

2.2 Transformace rovnoměrného rozdělení na jiná rozdělení

Rozdělení, která knihovna podporuje, jsou následující: rovnoměrné, exponenciální, normální, Weibullovo, Poissonovo a gamma. Pro různá rozdělení byly použity různé metody transformace z výchozího rovnoměrného rozložení.

2.2.1 Metoda inverzní transformace

Inverzní transformace je způsob, jímž se dají náhodná čísla generovat přesně v daném rozdělení [14]. Metoda vychází z funkce inverzní k distribuční funkci požadovaného rozdělení, což však znemožňuje použitelnost

této metody pro rozdělení, jejichž distribuční funkce buď nemají inverzní funkci nebo je tato inverzní funkce nevyjádřitelná elementárními funkcemi [14].

V knihovně je metoda inverzní transformace použita pro rovnoměrné, exponenciální a Weibullovo rozdělení. Všechna rozdělení mají poměrně jednoduše vyjádřitelnou inverzní funkci k distribuční funkci. Podoba použitých inverzních funkcí daných rozdělení je uvedena dále.

2.2.2 Metoda Ratio-of-Uniforms

Pro některá rozdělení pravděpodobnosti existuje více způsobů transformace z rovnoměrného rozdělení, avšak ne všechny tyto způsoby jsou dostatečně efektivní. Kinderman a Monahan vytvořili kombinovanou transformační metodu, která je poměrně efektivní, přestože není implementačně složitá [3]. Metoda Ratio-of-Uniforms (česky snad „poměr rovnoměrných“) vychází z metody vylučovací, ze které přebírá princip nacházení takových dvou náhodných rovnoměrně rozdělených čísel, že leží uvnitř specifického dvourozměrného tvaru. Z těchto dvou čísel je pak náhodné číslo cílového rozdělení vytvořeno výpočtem poměru původních dvou [13]. Důkaz platnosti metody je podán v [3] i [13]. Metoda byla použita pro generování náhodných čísel v normálním a v Poissonově rozdělení.

Co se týče normálního (Gaussova) rozdělení, byla použita implementace algoritmu, který vytvořil a v článku *A Fast Normal Random Number Generator* publikoval Joseph L. Leva [4]. Ten výpočetně optimalizoval metodu Ratio-of-Uniforms tak, že minimalizoval potřebný počet výpočtů logaritmu, přičemž nabízí i srovnání s dalšími algoritmy [4]. Algoritmus od Leva je přesný a rychlý, takže při výběru metody nevzniklo žádné dilema. Alternativně by mohla být použita jedna z variant Box-Miller transformační metody.

Poissonovo rozdělení je jediné diskrétní rozdělení v dokumentované knihovně. Kniha [13] navrhuje a knihovna pro generování pseudonáhodných čísek implementuje dva způsoby generování čísel v Poissonově rozdělení, které se používají v závislosti na střední hodnotě rozdělení. První metodou je zmíněná Ratio-of-Uniforms s řadou výpočetních optimalizací, druhou pak metoda násobící (Multiplication Method) [2], kterou jako první popsal Donald Knuth ve svém díle *The Art of Computer Programming*, ale která je použitelná pouze pro nízké střední hodnoty λ (pro ně je výpočet efektivní). Nutno zmínit, že v článku [2] je nabízena řada dalších alternativních metod pro výpočet včetně jejich srovnání, z něhož vyplývá, že Knuthova metoda je výhodná jen pro střední hodnoty nepřesahující hodnotu 5 [2], tab. P]. Bohužel se zde nevyskytuje srovnání s metodou Ratio-of-Uniforms, avšak ta by dle [13] měla být rychlá i pro velmi velké střední hodnoty.

2.2.3 Vylučovací metoda

Vylepšená verze vylučovací metody, jak ji navrhli Marsaglia a Tsang v [5] je použita pro generování gamma rozdělení. Základní princip metody spočívá ve vygenerování dvojice náhodných čísel v rovnoměrném rozdělení a testování zda vyhovují funkci hustoty cílového rozdělení. Pro gamma rozdělení je metoda upravena, přičemž více podrobností je uvedeno dále.

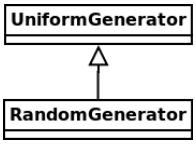
3 Koncepce a analýza

3.1 Návrh programu

Knihovna pro generování pseudonáhodných čísel využívá objektově orientovaný přístup ve svém návrhu (a posléze i v implementaci). Stanoveným cílem knihovny je dát uživateli možnost vytvořit jeden objekt, který umí generovat náhodnost všech rozdělení, a pokryje pro tyto účely potřeby daného lokálního prostoru, kde bude objekt moci až do svého zničení neustále poskytovat správná čísla. Uživateli však musí být umožněno vytvářet libovolný počet generátorů a to se stejnými i různými inicializačními hodnotami. Pokud bude uživatel knihovny chtít změnit základní generátor čísel v rovnoměrném rozdělení, musí být tato změna pro něj co nejméně náročná.

Ke splnění uvedených požadavků postačuje použití základních objektově orientovaných principů jako jednoduchá dědičnost a zapouzdření. Obrázek 1 zobrazuje, jak jednoduchá je struktura knihovny pro generování pseudonáhodných čísel. Dvě třídy ve vztahu rodič – potomek, kde rodič je libovolný generátor pseudonáhodných

čísel v rovnoměrném rozdělení a potomek generátor rozdělení z něho transformovaných. Použitá dědičnost je jednoduchá, použitá k účelu snadné záměny rodiče bez nutnosti změn v potomku a všude využívá volání metod časnou vazbou.



Obrázek 1: Struktura knihovny

3.2 Analýza rozdělení

Nyní budou podrobněji rozebrána jednotlivá rozdělení pravděpodobnosti. Popis každého z nich zahrnuje informace obecného rázu, způsob značení, jak se může objevit v literatuře nebo v abstraktním modelu a příklady využití rozdělení v oblasti modelování a simulací. K tomu jsou uvedeny konkrétní postupy výpočtů tak, jak je implementuje knihovna.

3.2.1 Rovnoměrné rozdělení

Rovnoměrné rozdělení je takové rozdělení, kdy náhodná veličina X nabývá se stejnou pravděpodobností, jakoukoliv hodnotu z intervalu $< a, b >$ [10]. Pokud jsou hodnoty $a = 0$ a $b = 1$, mluvíme o normované formě rovnoměrného rozdělení a takové rozdělení je základem pro generování dalších rozdělení. Rozdělení můžeme vyjádřit diskrétně i spojitě, avšak zde uvažujeme pouze vyjádření spojité.

Označení: $R(a, b)$ nebo $Uniform(a, b)$.

V modelování a simulacích je toto rozdělení důležité zejména z hlediska generování náhodných čísel, která jsou dále transformována na jiná rozložení. Při modelování se použije například pro doby čekání nebo doby různých činností.

Rovnoměrné rozdělení poskytuje knihovna jako výchozí rozdělení, avšak v normalizované podobě. Pro zobecněnou formu je potřeba použít nějakou transformační metodu, jíž se nabízí metoda inverzní transformace. Distribuční funkce rovnoměrného rozdělení v intervalu $< a, b >$ je

$$F(x) = \frac{(x - a)}{(b - a)}$$

Vyjdeme-li ze vztahu

$$u = F(x), u \in Uniform(0, 1)$$

pak

$$u = \frac{(x - a)}{(b - a)}$$

a po jednoduché úpravě dostaneme

$$x = a + u * (b - a) \tag{1}$$

což je výsledná inverzní funkce

$$x = F^{-1}(u)$$

Jinými slovy máme-li náhodné číslo $u \in < 0, 1 >$, potom funkce 1 jej převede na náhodné číslo $x \in < a, b >$.

3.2.2 Normální rozdělení

Normální (též Gaussovo) rozdělení je nejdůležitějším spojitým rozdělením [10], slide 60]. Využívá se ve statistice (chyby měření) a při aproximaci mnoha jiných, spojitých i diskrétních, rozdělení. Množství náhodných veličin v různých odvětvích vědy a techniky má normální rozdělení. Graf hustoty pravděpodobnosti tohoto rozdělení nese vlastní název Gaussova křivka; ta je specifická tím, že je souměrná podle osy $x \equiv \mu$ a v bodě $x = \mu$ má maximum.

Jeho parametry jsou též jeho charakteristikami, přičemž střední hodnota $E(X) = \mu$ a rozptyl $D(X) = \sigma^2$.

Podobně jako u rovnoměrného rozdělení, pokud parametry nabývají hodnot $\mu = 0$ a $\sigma = 1$, mluvíme o normované (též standardizované) formě normálního rozdělení. [10]

Označení: $N(\mu, \sigma)$

Existuje velká řada jevů (např. jevy s vlivem většího počtu nezávislých faktorů), které odpovídají normálnímu rozdělení, takže se v simulacích jedná o poměrně častě využívané rozdělení. Také se může vyskytnout při použití metody Monte Carlo nebo při vyhodnocování výsledků simulací.

Pro výpočet normálního rozdělení je v knihovně použita transformační metoda Ratio-of-Uniforms, jejíž princip byl nastíněn v přecházející kapitole. Několikastránkový popis celého algoritmu je napsán v článku [4], kam odkazujeme zájemce o něj.

3.2.3 Exponenciální rozdělení

Exponenciální rozdělení je spojité rozdělení modelující situace, kdy opakovaně a nezávisle dochází k výskytu náhodné události a zároveň nenastane více těchto situací najednou [10]. Rozdělení má vhodné vlastnosti pro upotřebení v teorii spolehlivosti [17].

Střední hodnota $E(X) = \frac{1}{\lambda}$ rozptyl $D(X) = \frac{1}{\lambda^2}$.

Označení: $E(\lambda)$ nebo $Exp(\lambda)$ [10].

Jedná se o velmi významné rozdělení pro modelování a simulace. V systémech hromadné obsluhy je obvykle využité pro doby mezi příchody do front či pro doby čekání ve frontách. Používá se pro modelování doby čekání na výskyt nějakého jevu, takže dobře popisuje například dobu života zařízení, u kterého dochází k poruše ze zcela náhodných příčin (nikoliv z důsledků opotřebení) [17].

Protože existuje inverzní funkce k distribuční funkci exponenciálního rozdělení a je vyjádřitelná pomocí elementárních matematických funkcí, můžeme ji použít pro transformaci z rovnoměrného rozdělení. Distribuční funkce je

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{pro } x \geq 0 \\ 0 & \text{pro } x < 0 \end{cases}$$

Řekněme, že $u \in Uniform(0, 1)$ a $u = F(x)$. Potom po úpravě

$$-\lambda x = ln(1 - u)$$

$$x = -\frac{ln(1 - u)}{\lambda}$$

Můžeme však ušetřit jedno odečítání, protože

$$(1 - u) \in Uniform(0, 1)$$

stejně jako

$$u \in Uniform(0, 1)$$

takže výsledná funkce transformující normalizované rovnoměrné rozdělení na exponenciální je

$$x = -\frac{ln(u)}{\lambda}$$

POZN.: Knihovna pro generování pseudonáhodných čísel i tento dokument používá jako parametr exponenciálního rozdělení hodnotu λ, avšak někdy se exponenciální rozdělení definuje parametrem $\delta = \frac{1}{\lambda}$, tedy svou střední hodnotou.

3.2.4 Weibullovo rozdělení

Weibullovo rozdělení popisuje takovou náhodnou veličinu X, která vyjadřuje čekání na událost, jež se může může dostavit s šancí úměrnou mocninné funkci dosud pročekané doby. Používá se všude tam, kde nevyhovuje „rozdělení bez paměti“, tedy exponenciální [17] [11]. V praxi se jedná o zařízení, kde se projevuje mechanické opotřebení nebo únava materiálu [11]. Toto spojité rozložení má dva kladné parametry, které se nazývají měřítko a tvar (forma) [18] [10]. Pokud je tvar > 1, je charakterizováno zařízení, u kterého se pravděpodobnost poruchy zvyšuje, naopak pro tvar < 1 se pravděpodobnost poruchy snižuje. Je-li tvar = 1, jedná se o exponenciální rozdělení [18].

Označení: $W(tvar, mtko)$, $Wb(tvar, mtko)$ nebo $Weibull(tvar, mtko)$.

Při modelování spolehlivosti či selhání například výrobního zařízení se uplatní Weibullovo rozdělení před exponenciálním. Díky simulacím pak lze usnadnit rozhodování, zda se má modelované zařízení nahradit dříve než selže. Dle [18] se využívá i k prezentování výrobních a dodacích časů v průmyslu nebo k předpovědím počasí.

Distribuční funkce Weibullova a exponenciálního rozdělení si jsou dost podobné, neboť Weibullovo rozdělení je (jistým způsobem) zobecněné exponenciální rozdělení [13]. Podobné je i odvození inverzní funkce k distribuční funkci, která je definována jako

$$F(x) = 1\text{–}e^{-(\frac{x}{\beta})^{\alpha}},$$

kde β je měřítko a α je tvar.

Řekněme, že $u \in Uniform(0, 1)$ a $u = F(x)$, takže po úpravách získáme nejprve

$$\left(\frac{x}{\beta}\right)^{\alpha} = -ln(1-u)$$

a poté

$$x = \beta * ^{\frac{1}{\alpha}}.$$

Podobně jako bylo odstraněno odčítání v argumentu logaritmu u exponenciálního rozdělení, jde odstranit i zde.

3.2.5 Gamma rozdělení

Spojité rozdělení, které podobně jako rozdělení Weibullovo má dva parametry pojmenované tvar a měřítko. Odpovídá době čekání na n-tou událost, kde n je parametr tvar. Pro celočíselný tvar přechází na Erlangovo rozdělení [16], pro tvar = 1 se stává exponenciálním rozdělením [1]. V kombinaci s Poissonovým rozdělením tvoří negativní binomické rozdělení. Využití tohoto rozdělení najdeme (kromě modelování a simulací) ve statistice a meteorologii. [16]

Označení: $Gamma(k, \Pi)$ nebo $\Gamma(k, \Pi)$.

Využití v simulacích je možné pro modely života (umírání) [1] [16] nebo také tam, kde nachází uplatnění Erlangovo rozdělení, tedy modelování příchodů, dob čekání nebo u compartment models (česky snad „členěné modely“).

V [13] jsou uvedené dvě metody transformace a to pro pro různé hodnoty parametru k (tvar). Pro $k < 1$ je využito vztahu

$$y * u^{\frac{1}{k}} \sim \Gamma(k, 1), \text{kde } y \sim \Gamma(k+1, 1) \text{ a } u \sim Uniform(0, 1)[13].$$

Pro $k > 1$ je použita upravená verze vylučovací metody, kterou vytvořili Marsaglia a Tsang a která využívá Gaussovu křivku, takže její rychlost závisí na rychlosti počítání nejen rovnoměrného ale i normálního rozdělení. Jelikož je metoda poměrně složitá a její výklad by zabral několikero stran, uvádíme pouze informační zdroj, jímž je [5].

3.2.6 Poissonovo rozdělení

Poissonovo rozdělení je jediné diskrétní rozdělení v knihovně pro generování pseudonáhodných čísel. Úzce souvisí s exponenciálním rozdělením, které popisuje dobu mezi dvěma událostmi, zatímco Poissonovo počet výskytů události za určitou dobu [10]. Použijeme-li parametr rozdělení λ, který zároveň představuje střední hodnotu i rozptyl, můžeme tvrdit, že k výskytu události dochází průměrně jednou za $\frac{1}{\lambda}$ časových jednotek, tj. λ-krát za jednu časovou jednotku [10]. Pomocí Poissonova rozdělení jde za určitých podmínek aproximovat binomické rozdělení [10] a existuje také vztah pro převod na rozdělení exponenciální.

Označení: $P(\lambda)$ nebo $Poisson(\lambda)$

Poissonovo rozdělení je pro modelování a simulace důležité podobně jako jeho exponenciální protějšek. Modeluje se s ním počet příchodů za jednotku času v systémech hromadné obsluhy nebo obecně počty jakýchkoliv jevů vyskytujících se v určitém časovém kvantu.

Stejně jako u normálního rozdělení je pro transformaci z rovnoměrného do Poissonova rozdělení použita metoda Ratio-of-Uniforms, která je tedy evidentně použitelná i pro diskrétní rozdělení. Trik převodu reálných hodnot, které jsou umístěné uvnitř metodou žádaného planárního útvaru, na hodnoty diskrétní spočívá v jednoduchém ořezání desetinné části [13]. Pro další detaily však musí zájemce prostudovat principy metody Ratio-of-Uniforms v knize [13], protože zde není dostatek prostoru tuto metodu vysvětlovat.

3.2.7 Paretovo rozdělení

Paretovo rozdělení pravděpodobnosti bylo původně použito k popisu alokace bohatství mezi lidmi, protože se na něm dá ukázat, že hrstka lidí vlastní většinu veškerého bohatství (Paretův princip). Rozdělení je hierarchizováno na čtyři typy, I až IV, přičemž typy I a II jsou speciální případy typu IV, který je dále zobecněn v tzv. Feller-Pareto rozdělení. Parametry rozdělení jsou posun l, měřítko σ a parametr proměnlivosti γ. Jedná se o spojité rozdělení pravděpodobnosti.

Označení: $Pareto(l, \sigma, \gamma)$ nebo $Pr(l, \sigma, \gamma)$.

Používá se k popisu mzdového rozdělení, výskytu chyb harddisků, velikosti souborů přenášených přes protokol TCP nebo ke statistickému vyjádření velikosti meteoritů. Pro generátor byla použita metoda inverzní transformace, ovšem hodnoty tohoto rozložení jsou vypočitatelné z hodnot exponenciálního rozložení, protože platí $Pr = x_m * e^Y$, kde x_m je minimum a Y je hodnota daná exponenciálním rozložením.

3.2.8 Rayleighovo rozdělení

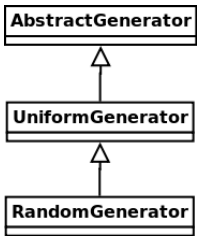
Rayleighovo rozdělení je speciálním případem Weibullova rozdělení, tudíž popisuje intenzitu poruch. Počítá s tím, že poruch přibývá s postupujícím časem (degradace věcí).

Označení: $Rayleigh(\sigma)$

Rozdělení se používá například při analýze spektra hustot rychlostí větru.

4 Architektura knihovny

Jak již bylo řečeno návrh knihovny využívá objektově orientované paradigma, které se přirozeně promítne také do implementace, pro níž byl vybrán jazyk C++, který toto paradigma podporuje. C++ je rychlý kompilovaný přenositelný jazyk, jenž navíc nabízí techniky generického programování, neboli v terminologii jazyka šablony. Struktura dědičnosti tříd tak, jak byla definována v návrhu je v implementaci rozšířena o čistě virtuální třídu *AbstractUniformGenerator*, která definuje rozhraní pro základní generátor náhodných čísel v rovnoměrném rozdělení, řekněme třeba *UniformGenerator*. Jeho potomkem je pak šablona třídy *RandomGenerator*, jejímž parametrem je právě onen generátor náhodnosti s rovnoměrným rozdělením, což uživateli umožňuje změnu základního generátoru při vytváření instance třídy *RandomGenerator* za jakýkoliv jiný generátor bez nutnosti zasahovat do kódu knihovny. Jedinou podmínkou je, že uživatelův *UniformGenerator* bude mít stejný protokol jako *AbstractUniformGenerator* tedy jinak řečeno, bude jeho potomkem. Dědičnosti tříd je znázorněna na obrázku 2.



Obrázek 2: Znázornění dědičnosti tříd

Knihovna nabízí dva generátory pseudonáhodných čísel s rovnoměrným rozdělením, Mersenne Twister a generátor založený na celulárním automatu. U Mersenne Twister se jedná o převzatou původní implementaci, kterou v jazyce C vytvořili a pod volnou licencí v [8] publikovali Matsumoto a Nishimura a kterou do jazyka C++ portoval Jasper Bedaux. V kódu bylo sice provedeno několik změn, ale jedná se spíše o změny formálního rázu za účelem přizpůsobení návrhu knihovny. Základní vlastnosti a srovnání algoritmu se v tomto dokumentu již objevily a jeho podrobný popis je podán tvůrci v článku [8]. Generátor založený na celulárním automatu, byl původně také implementován v jazyce C. Jeho port do jazyka C++ vytvořil autor knihovny.

Transformace z rovnoměrného rozdělení do všech šesti rozdělení pravděpodobnosti, která knihovna podporuje, jsou implementovány v šabloně třídy *RandomGenerator* jako její veřejné metody. Použité algoritmy jsou jak původní tak i převzaté a již byly detailněji rozebrány v přecházející části tohoto dokumentu, kde jsou uvedeny postupy odvození (je-li použita metoda inverzní transformace) nebo alespoň zdroje z nichž byly algoritmy čerpány. Z hlediska implementace však jistě stojí za zmínku mechanizmus uložení některých výsledků mezivýpočtů, který je použit pro urychlení generování Poissonova rozdělení za předpokladu, že je aspoň dvakrát za sebou použit stejný parametr. Jinými slovy, je zefektivněno časté po sobě jdoucí generování Poissonova rozdělení se stejným parametrem.

Implementace knihovny je kompletně celá umístěna v hlavičkových souborech a to jednak, protože je to nutnou podmínkou při použití šablon, a jednak z pragmatických důvodů proto, aby se uživatel nemusel starat o kompilaci knihovny a mohl ji jednoduše přiložit direktivou include. Ke knihovně je přiložena implementace funkce logaritmus gamma funkce, jež je použita v některých algoritmech pro urychlení výpočtu faktoriálu [13]. Tato funkce je sice součástí chystaného standardu C++0x jazyka C++, který je na některých platformách již nyní k dispozici ve standardní knihovně funkcí (ač má být oficiálně vydán na až jaře roku 2011) avšak zatím by bez ní knihovna nebyla úplná. V budoucnu, až se standard C++0x rozšíří, jí bude možno z knihovny odstranit.

5 Testování

Pro testování byly použity platformy Unix a MS Windows. Otestované kompilátory jsou následující:

- gcc version 4.4.3, 64-bitové PC
- gcc version 4.4.4, 32-bitové PC
- MingW-32

Knihovna je bez problémů přenositelná. Kromě standardní knihovny funkcí nejsou použity žádné jiné knihovny ani moduly jazyka.

5.1 Testování správnosti generovaných rozdělení

Důležitým milníkem ve vývoji knihovny bylo otestování správnosti generování implementovaných pravděpodobnostních rozdělení, respektive správnosti transformačních metod. Výstupem testovacího programu je posloupnost čísel v určitém pravděpodobnostním rozdělení a tu je potřeba ověřit.

Existuje sice řada testů, které matematicky dokáží zda daná posloupnost čísel náleží určitému rozdělení pravděpodobnosti, pro testování knihovny byl však vybrán méně exaktní test. Byly vytvořeny skripty pro program gnuplot, což je program určený pro počítačové kreslení grafů, které do jednoho svého výstupu, jímž je rastrový obrázek, vykreslí jeden histogram náhodné posloupnosti 100 000 hodnot a k tomu odpovídající analytické vyjádření funkce hustoty pravděpodobnosti daného rozdělení. Každé rozdělení je takto testováno několikrát, vždy s jinými parametry. Histogram posloupnosti a křivka funkce hustoty pravděpodobnosti spolu musí vždy přesně korelovat. U některých rozdělení odhalila testovací metoda v průběhu testování chyby, které, jak se později ukázalo, byly implementačního rázu; například opačné znaménko u výpočtu v algoritmu transformační metody normálního rozdělení. Chyba se projevila tak, že histogram vykazoval výchylky (ne na celém svém rozsahu) od analytického vyjádření hustoty pravděpodobnosti.

Všechna rozdělení úspěšně prošla řadou testů s různými parametry.

Ke knihovně jsou přiloženy také testovací skripty pro program gnuplot, které se spouští pomocí testovacího programu (je třeba mít nainstalovaný program gnuplot), takže je možno vyzkoušet a nahlédnout na výsledky tří vybraných posloupností pro každé rozložení (každá posloupnost má samozřejmě jiné přednastavené parametry).

Pozn.: Grafický výstup nebude fungovat na platformě MS Windows.

6 Závěr

Byla vytvořena knihovna pro generování pseudonáhodných čísel implementující šest pravděpodobnostních rozdělení pro potřeby modelování a simulací, která jsou generována transformací z volitelného generátoru rovnoměrně rozložené náhodnosti. Knihovna se uplatní při psaní simulačních modelů v jazyce C++, jejichž cílem je efektivní a přesná simulace. Použití knihovny je tak jednoduché jako vložení hlavičkového souboru a vytvoření instance třídy, přičemž má uživatel možnost použít vlastní generátor rovnoměrného rozdělení, pokud nechce použít výchozí, byť kvalitní.

Autorem implementace knihovny pro generování pseudonáhodných čísel, s výjimkou generátoru založeného na celulárních automatech, je Pavel Novotný. Implementace byla učiněna na základě metod zmíněných v této dokumentaci a s pomocí pseudokódů obsažených v knize Numerical Recipes (2. a 3. vydání, autoři: William H. Press, Saul A. Teukolsky, William T. Vetterling a Brian P. Flannery). Implementace algoritmu Mersenne Twister byla inspirována existující implementací od Jasper Bedaux tak, že jeho kód byl optimalizován a přizpůsoben návrhu knihovny. Testování pravděpodobnostních rozložení bylo provedeno vizuálně vykreslením hustoty příslušného rozdělení a histogramu vygenerovaných čísel. Korelace obou grafů byla považována za úspěch v testu.

Návrh a architektura knihovny včetně popisu použitých metod a postupů jsou zdokumentovány, aby pomohly pochopit, jakým způsobem knihovna pracuje, jak může být použita či případně upravena. Dokumentace knihovny si nekladla za cíl dopodrobna vyložit všechny detaily a vysvětlit všechny algoritmy. Mnoho věcí zůstalo skryto, ale vždy bylo ukázáno, kde je možno je najít. To nejdůležitější, jímž je nastínění celkové problematiky, kterou tvorba knihovny pro generování pseudonáhodných čísel představuje, je v dokumentaci obsaženo.

Bibliografie

[1] The Gamma Distribution. 2006, [online].
URL <http://www.weibull.com/LifeDataWeb/the_gamma_distribution.htm>

[2] Ahrens, J. H.; Dieter, U.: Computer Methods for Sampling from Gamma, Beta, Poisson and Binomial Distributions*. ročník 12, Září 1973: s. 223 – 246.

[3] Kinderman, A. J.; Monahan, J. F.: Computer Generation of Random Variables Using the Ration of Uniform Deviates. *ACM Transactions on Mathematical Software*, ročník 3, č. 3, Září 1977: s. 257 – 260.

[4] Leva, J. L.: A Fast Normal Random Number Generator. *ACM Transactions on Mathematical Software*, ročník 18, č. 4, Prosinec 1992: s. 449 – 453.

[5] Marsaglia, G.; Tsang, W.-W.: A Simple Method for Generating Gamma Variables. ročník 26, č. 3, Září 2000: s. 363 – 372.

[6] Marsaglia, G: Random Number Generators. *Journal Of Modern Applied Statistical Methods*, ročník 2, č. 1, Květen 2003: s. 2 – 13.

[7] Matsumoto, M.: Mersenne Twister Home Page. [online].
URL <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>

[8] Matsumoto, M.; Nishimura, T.: Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Transactions on Modeling and Computer Simulation*, ročník 8, č. 1, Leden 1998: s. 3 – 30.

[9] Matsumoto, M.; Nishimura, T.; M., S.; aj.: Pseudorandom Number Generation: Impossibility and Compromise. *Journal of Universal Computer Science*, ročník 12, č. 6, Červen 2006: s. 672 – 690.

[10] Novák, M.: *Numerická matematika a pravděpodobnost (slajdy)*. VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, Září 2009.

[11] Otipka, P.; Šmajstrla, V.: *Základní typy rozdělení pravděpodobnosti spojité náhodné veličiny. Pravděpodobnost a statistika*. 2003, [online].
URL <http://homen.vsb.cz/~oti73/cdpast1/KAP05/PRAV5.HTM>

[12] Pasqualoni, A.: Random number generation using a 256-state cellular automaton. [online].
URL <http://home.southernct.edu/~pasqualonia1/ca/report.html>

[13] Press W. H. and Teukolsky, S. A. and Vetterling, W. T. and Flannery B. P.: *Numerical Recipes, The Art of Scientific Computing*, ročník 3. Cambridge University Press, 2007, ISBN 0-521-88068-8.

[14] Rábová, Z. and Češka, M. and Zendulka, J. and Peringer, P. and Janoušek, V.: *Modelování a simulace (skripta)*. VUT v Brně, Fakulta informačních technologií, Leden 2005.

[15] Wikipedia: Generátor pseudonáhodných čísel. Listopad 2009, [online].
URL <http://bit.ly/fMno3p>

[16] Wikipedia: Gamma distribution. Prosinec 2010, [online].
URL <http://en.wikipedia.org/wiki/Gamma_distribution>

[17] Řezanková, H.; Marek, L.; Vrabec, M.: *Exponenciální rozdělení. Interaktivní učebnice statistiky*. 2001, [online].
URL <http://iastat.vse.cz/Exponenc.htm>

[18] Ťoupal, T.: *Weibullovo rozdělení, dvourozměrné normální rozdělení*. Říjen 2009, materiál KMA/PSB. Západočeská univerzita.