

# IMS

## Příklady na semestrálku

Přebráno z Fitušky ( fórum a wiki)

2. února 2014

### 1 Programování na papír

#### 1.1 Monte Carlo (nick)

Napište v jazyku C funkci `double MCIntegral()` pro výpočet N-násobného určitého integrálu funkce `double f(double x[], unsigned N)` (podle všech  $x_i$ ,  $i$  náleží  $0, \dots, N-1$ ) metodou Monte Carlo (integrál musí být N-násobný, jinak 0 bodů) s přesností přibližně 1%. Meze integrálu jsou uloženy v globálním poli struktury `Range` s položkami `.min` a `.max`. Předpokládejte, že neznáte rozsah funkčních hodnot funkce `f`. Máte k dispozici pouze funkci `double Random(void)` vracející pseudonáhodná čísla v rozsahu  $(0,1)$ , `f` a žádnou jinou

```
1 double MCIntegral() {
2     // chyba = 1/sqrt(N) -> 1% = 1/sqrt(10000)
3     unsigned pokusu = 10000;
4     double suma = 0.0;
5     for(unsigned i=0; i<pokusu; i++){
6         double x[N];
7         for(unsigned ii=0; ii<N; ii++){
8             x[ii] = Range[ii].min
9                 + Random() * (Range[ii].max - Range[ii].min);
10        }
11        suma += f(x,N);
12    }
13    double refPlocha = 1.0;
14    for(unsigned ii=0; ii<N; ii++)
15        refPlocha *= Range[ii].max - Range[ii].min;
16    return refPlocha * suma/pokusu;
17 }
```

## 1.2 Kongruentní generátor pseudonáhodných čísel (nick)

V ISO C napište funkci `double Random(void)` ve které bude váš vlastní kód kongruentního generátoru s rozsahem  $\langle 0, 1 \rangle$ , konstanty `A`, `B`, `M` máte k dispozici. Máte zadánu funkci hustoty pravděpodobnosti  $f(x)$  (nakreslete ilustrační obrázek).

$$f(x) = \frac{x}{2} \text{ pro } x \text{ na intervalu } \langle 0, 2 \rangle \quad (1)$$

$$f(x) = 0 \text{ jinak} \quad (2)$$

- Napište funkci `double RandGenInv(void)`, která generuje číslo se zadaným rozložením s využitím metody inverzní transformace (jinak -5 bodů).
- Napište funkci `double RandGenReject(void)`, která generuje číslo se zadaným rozložením s využitím metody vylučovací (jinak -5 bodů).
- Vypočtěte pravděpodobnost, že náhodná proměnná  $X$  s hustotou pravděpodobnosti  $f(x)$  zadanou výše bude mít hodnotu 0.5.
- Vyjádřete distribuční funkci obecného spojitého rozložení  $F(x)$  pomocí její funkce hustoty pravděpodobnosti  $f(x)$ .

```
1 double Random() {
2     static unsigned long ix = 0;
3     ix = A*ix+B; // <0,ULONG_MAX>
4     return ix / ((double)ULONG_MAX + 1); // <0,1>
5 }
```

Řešení

- a) Rovnice

$$F(x) = \int \frac{x}{2} dx = \frac{x^2}{4} \quad (3)$$

$$F^{-1}(x) = \sqrt{4x} = 2\sqrt{x} \quad (4)$$

Program

```
1 double RandGenInv() {
2     return sqrt(Random()) * 2.0;
3 }
```

- b) Program

```

1 double f(double x) { // funkce hustoty
2     if (0.0 <= x && x < 2.0)
3         return x / 2.0;
4     else
5         return 0.0;
6 }
7 double RandGenReject() {
8     double x,y;
9     do {
10         x = Random() * 2;
11         y = Random() * 1;
12     } while (y > f(x)); // trefili jsme se nad funkci
        hustoty -> znovu
13     return x;
14 }

```

c)

$$\int_{0.5}^{0.5} f(x)dx = 0$$

Poznámka: Pravděpodobnost jednoho konkrétního čísla z nekonečna (na spojitém intervalu je nekonečné množství reálných čísel) je vždy nulová. Aby byla nenulová, musí to být hodnota z intervalu (pak se nebudou meze integrálu rovnat).

d)

$$F(x) = \int_{-\infty}^x f(t)dt$$

Poznámka:  $t$  je pomocná proměnná zavedená pro odlišení  $x$ , které je parametrem funkce  $f(x)$  a podle kterého se derivuje, od  $x$ , které je vstupním parametrem funkce  $F(x)$ . Nesouvisí s žádnou proměnnou mimo tento výraz.

### 1.3 Algoritmy řízení simulace

#### 1.3.1 Řízení diskrétní simulace – Next-event (skeWer)

```

1 Inicializace_kalendare
2 while(kalendar_neprazdny) {
3     vyjmi_prvni_zaznam_z_kalendare
4     if (aktivacni_doba_zaznamu > doba_konce_simulace) break;
5     cas = aktivacni_doba_zaznamu; //hlavne neprehodte tyto
        dva radky
6     proved_udalost //jinak...
7 }
8 cas = doba_konce_simulace

```

### 1.3.2 Řízení spojitě simulace (Royce)

```
1 inicializace poc. stavu a promennych a casu;
2 while (cas < koncovy_cas) {
3     Print_results();
4     Update_integrators();
5     if (cas + krok > koncovy_cas)
6         krok = koncovy_cas - cas;
7     euler();
8     cas += krok;
9 }
```

### 1.3.3 Řízení kombinované simulace (skeWer)

```
1 inicializace poc. stavu a promennych a casu;
2 while (cas < koncovy_cas) {
3     vyber_zaznam_z_kalendare();
4     while ( (cas + krok < koncovy_cas) && (cas + krok <
        aktivacni_cas_pristi_udalosti) ) { //"neprekroceni"
        udalosti / konce
5     krok_numericke_metody();
6     cas += krok;
7     }
8     //v tomhle miste jsme bud kousek pred udalosti (kousek <=
        normalni_krok), nebo kousek pred koncem simulace
9     //cili pred provedenim udalosti / koncem simulace musime
        udelat jeste jeden krok, ale jeho velikost urcime
10    //podle toho, jestli budeme provadet udalost nebo konciti
11    if (aktivacni_cas_pristi_udalosti > koncovy_cas) { //prvni
        udalost lezi mimo simulaci -> kaslem na ni
12        krok = koncovy_cas - cas;
13        krok_numericke_metody();
14        cas += krok; //nebo: cas = koncovy_cas
15        konec_simulace();
16    }
17    else {
18        krok = aktivacni_cas_pristi_udalosti - cas;
19        krok_numericke_metody();
20        cas += krok; //nebo: cas = aktivacni_cas_pristi_udalosti
21        proved_udalost();
22        krok = normalni_delka_kroku;
23    }
24 }
```

## 2 Markovské procesy

### 2.1 D/D/x, kalendáře

#### 2.1.1 D/D/x bez poruchy (wiki)

D/D/2 popsat stav simulace, kalendář, pořadí událostí a frontu v čase 6.5s, začíná se v 0s, generuje se každou 1s a zpracování trvá 3s.

Časový diagram:

t	0	1	2	3	4	5	6	7	8
Z0	P0	P0	P0	P2	P2	P2	P4	P4	P4
Z1		P1	P1	P1	P3	P3	P3		
Q			P2	P3	P4	P4	P5		
						P5	P6		

Seznam událostí:

0. vygenerování P0, P0 obsazuje Z0
1. vygenerování P1, P1 obsazuje Z1
2. vygenerování P2, P2 jde do fronty
3. konec obsluhy P0, P2 obsazuje Z0, vygenerování P3, P3 jde do fronty
4. konec obsluhy P1, P3 obsazuje Z1, vygenerování P4, P4 jde do fronty
5. vygenerování P5, P5 jde do fronty
6. konec obsluhy P2, P4 obsazuje Z0, vygenerování P6, P6 jde do fronty

---

(dále jsou již jenom naplánované události)

7. konec obsluhy P3, vygenerování P7
- 8.
9. konec obsluhy P4

Fronta:

→	P5
	P6

Obsah kalendáře	čas	priorita	událost
	7	0	P3→Release
	7	0	Generator→Activate
	9	0	P4→Release

### 2.1.2 D/D/x s poruchou (mesma)

D/D/3, fronta LIFO, příchody každou 1s, obsluha trvá 3s. V čase 2s přijde porucha do Z0 a trvá 2s. Vyjádřit v čase 4.5s.

Časový diagram:

t	0	1	2	3	4	5	6
Z0	P0	P0	x	x	P0		
Z1		P1	P1	P1	P3	P3	P3
Z2			P2	P2	P2		
Q				P3	P4		

Seznam událostí:

- vygenerování P0, P0 obsazuje Z0
- vygenerování P1, P1 obsazuje Z1
- porucha obsazuje Z0, vygenerování P2, P2 obsazuje Z2
- vygenerování P3, P3 jde do fronty
- konec poruchy Z0, konec obsluhy P1, P3 obsazuje Z1, vygenerování P4, P4 jde do fronty

---

(dále jsou již jenom naplánované události)

- konec obsluhy P2, konec obsluhy P0, vygenerování P5
- 
- konec obsluhy P3

Fronta: 

→	P4
---	----

Obsah kalendáře	čas	priorita	událost
	5	0	P2→Release
	5	0	P0→Release
	5	0	Generator→Activate
	7	0	P3→Release

### 3 Teorie (*wiki*)

**Abstraktní model** zjednodušený popis zkoumaného systému

**Buňka (cell)** základní element, může být v jednom z konečného počtu stavů (0/1)

**Celulární automat** je souhrnné označení pro určitý typ fyzikálního modelu reálné situace, slouží k časové i prostorové diskrétní ideální modelaci fyzikálních systémů, kde hodnoty veličin nabývají pouze diskrétních hodnot

**Chování systému** každému časovému průběhu vstupní veličiny přiřazuje časový průběh výstupních veličin, je dáno vzájemnými interakcemi mezi prvky systému a mezi systémem a jeho okolím

**Diskrétní simulace** simulace s diskrétními prvky a časem. Experimentování se simulačním modelem.

**Markovova vlastnost** následující stav procesu závisí jen na aktuálním stavu (ne na minulosti)

**Markovský proces** náhodný diskrétní proces se spojitým časem, který splňuje markovovu vlastnost

**Markovův řetězec** náhodný diskrétní proces s diskrétním časem, který splňuje markovovu vlastnost, ekvivalentem je konečný automat s pravděpodobnostmi přechodů

**Model** napodobenina systému jiným systémem

**Modelování** proces vytváření abstraktního modelu

**Modelový čas** časová osa modelu, může být reálný, zrychlený, zpomalený

**Monte Carlo** metoda používaná k výpočtu určitých integrálů, určená pro simulaci systémů, které jsou založeny na experimentování se stochastickými modely s využitím generátorů náhodných čísel

**Okolí (neighbourhood)** několik typů, liší se počtem okolních buněk, se kterými se pracuje

**Pole buněk (lattice)**  $n$ -rozměrné, obvykle 1D nebo 2D, rovnoměrné dělení prostoru, může být konečné nebo nekonečné

**Pravidla (rules)** funkce stavu buňky a jejího okolí definují nový stav buňky v čase  $s(t + 1) = f(s(t), N_s(t))$

**Petriho síť** graf popisující stavy a přechody mezi stavy, umožňuje znázornění paralelismu a nedeterminismu

**Priorita obsluhy** proces s vyšší prioritou obsluhy než současný může vyhodit současný proces a sám obsadit zařízení

**Priorita procesu** pro řazení ve vnější frontě u zařízení a v kalendáři (primárne podľa času a sekundárne podľa priority)

**Proces** posloupnost událostí

**Reálný čas** probíhá v něm skutečný děj v reálném systému

**Řád metody numerické integrace** stupeň polynomu  $N$  (Taylorův rozvoj) definuje tzv. řád metody, tímto polynomem aproximujeme na základě aktuální hodnoty

**Rychlá smyčka** tvoří cyklus v orientovaném grafu závislostí funkčních bloků.

**SHO** systémy obsluhující zařízení, která poskytují obsluhu transakcím, součástí SHO jsou transakce, fronty, obslužné linky

**Simulace** experimentování s modelem, získávání nových znalostí

**Simulační model** abstraktní model zapsaný kódem

**Spojité systém** všechny prvky systému mají spojitě chování (nemění své parametry skokově)

**Statistiky** vytížení zařízení, délky front, doby čekání ve frontách, využití kapacity skladů, celková doba transakce strávené v systému, maximum, minimum, střední hodnoty, směrodatná odchylka, rozptyl, ...

**Stavová podmínka** diskretní podmínka ve spojitě simulaci, která může nějakým způsobem ovlivnit model (jiné rovnice pro výpočet kapaliny a páry), je citlivá na hranu, takže se vykonává pouze při změně, na základě její změny vyvolám stavovou událost

**Stavová událost** akce, která je podmíněna změnou pravdivostní hodnoty stavové podmínky (např.: otočení hodnoty integrátoru), diskretní skoková změna systému

**Strojový čas** čas CPU spotřebovaný na výpočet programu (závisí na složitosti simulačního modelu, nesouvisí přímo s modelovým časem)

**Systém** soubor elementárních částí, které mají mezi sebou vazby

**Třídy statistik** stat, tstat (bere v potaz čas), histogram, mají společné operace s.Clear() (inicializace), s.Output() (tisk), s(x) (záznam hodnoty x)

**Událost** změna stavu diskretního systému, jednorázová, nepřerušitelná

**Validace modelu** ověřování platnosti modelu a proces, kdy se snažíme dokázat, že pracujeme s modelem adekvátním modelovanému systému

**Verifikace modelu** dokazování izomorfnosti (rovnosti) abstraktního modelu s modelem simulačním, jinak řečeno ověřování korespondence abstraktního modelu se simulačním modelem