Министерство образования и науки РФ
ГОУ ВПО «Удмуртский государственный университет»

А. Сивков, Д. Сивков

ИСПОЛЬЗОВАНИЕ ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЕЛ НА МНОГОЯДЕРНЫХ КОМПЬЮТЕРАХ

Лабораторный практикум

Ижевск 2010

УДК 519.6
ББК 22.18
С34

Рекомендовано к изданию
учебно-методическим советом УдГУ

Сивков А. М., Сивков Д. А.

С34 Использование генераторов случайных чисел на многоядерных компьютерах: лабораторный практикум / Изд-во «Удмуртский университет», Ижевск: 2010, 34 с.

УДК 519.6
ББК 22.18

© А. Сивков, Д. Сивков, 2010
© ГОУВПО «Удмуртский государственный университет», 2010

Введение

Студентам естественнонаучных специальностей часто приходится ставить эксперименты, в том числе вычислительные. Как правило, эти эксперименты требуют использования значительных ресурсов. Так, например, для полного понимания динамики происходящих процессов бывает полезным демонстрировать развитие процесса в режиме реального времени или даже ускоренно.

Радикальное снижение стоимости многоядерных компьютеров и их доступность позволяют использовать их возможности при проведении лабораторного практикума.

Однако для того чтобы вычислительные ресурсы использовались эффективно, необходимо подготовить специальное программное обеспечение, ориентированное на применение таких систем, и расширить навыки студентов, которые будут его использовать.

Случайные события часто встречаются в натурных экспериментах и их влияние на результаты необходимо учитывать. Данное пособие призвано дать студентам необ-

ходимый лабораторный практикум по использованию генераторов случайных чисел.

Для проведения практикума рекомендуется использование многоядерных компьютеров, операционных систем семейства Linux, средств разработки GCC, библиотек GNU SL, Intel MKL.

Рассматриваемый лабораторный практикум является частью курса «Параллельные алгоритмы», читаемого на факультете Информационных технологий и вычислительной техники УдГУ.

Модель

Задача о горнолыжнике

Пусть горнолыжник совершает спуск из точки A в точку B на плоском склоне горы, расположенном под некоторым углом к горизонту. Требуется найти такую траекторию, спуск вдоль которой занимает наименьшее время.

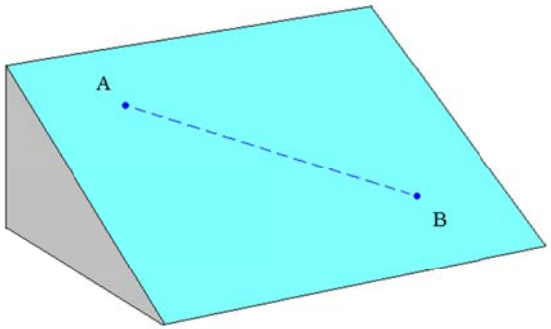

Рис.1. Положение точек старта и финиша на склоне

Обсуждение задачи

Заметим, что если бы скорость горнолыжника была постоянной и не зависела от вида траектории, то решением задачи была бы самая короткая из всех возможных траекторий: прямая, соединяющая точки A и B.

Однако различные направления на склоне не равноправны. Горнолыжник начинает спуск, имея нулевую (или очень небольшую) начальную скорость. Затем, вследствие действия силы тяжести, его скорость растет.

Быстрее всего скорость возрастает при спуске в направлении наибольшей крутизны склона.

Если точка B лежит, хотя и ниже точки A, но несколько в стороне, для горнолыжника может оказаться более выгодным, с точки зрения затраченного времени, двигаться не сразу в направлении к точке B, а по траектории, отличной от прямолинейной. Так возможно вначале набрать скорость, двигаясь круто вниз, и только потом повернуть к точке B.

Рассмотрим простейший случай, когда траектория состоит из двух отрезков прямых.

Точку сопряжения отрезков будем называть узлом. Найти оптимальную траекторию из двух отрезков прямых – значит, найти положение узла, поскольку точки начала и конца траектории фиксированы. Один из способов поиска оптимальной траектории – метод перебора.

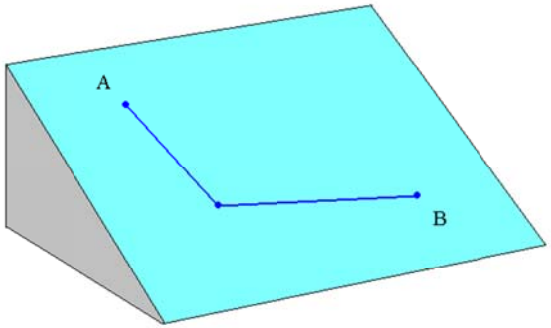

Рис.2. Одна из возможных траекторий спуска из точки *A* в точку *B*

В рассматриваемой задаче наиболее «быстрой» траекторией является гладкая кривая, а ломаная линия из двух отрезков прямых – всего лишь грубое приближение к ней. Тогда приближение к решению задачи можно искать в классе ломаных, состоящих из *N* отрезков прямых, и увеличивать *N* до тех пор, пока точность приближения не станет приемлемой.

Задача о горнолыжнике является типичным представителем целого семейства задач. Среди них, например, – задача о пикирующем бомбардировщике или задача о перелете космического корабля с одной планеты на другую.

Физическая модель

Будем полагать, что на каждом прямолинейном отрезке траектории на горнолыжника действуют силы:

1) тяжести $m\vec{g}$;
2) нормальной реакции опоры $\vec{N}$;
3) сухого трения $\vec{F}_{тр}$.

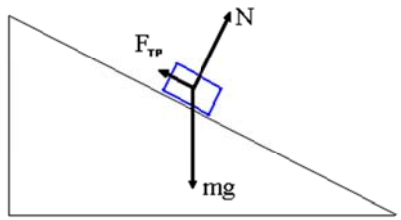

Рис. 3. Силы, действующие на горнолыжника

Под действием указанных сил происходит изменение скорости горнолыжника вдоль траектории. Вместе с тем, будем полагать, что в узле траектории поворот от одного направления движения к другому происходит мгновенно и без изменения быстроты движения.

Рассмотрим прямой отрезок траектории. Точка 1 – начало отрезка, точка 2 – его конец. Будем пользоваться прямоугольной декартовой системой координат. Ось Z направим вертикально вверх. Оси X и Y лежат в горизонтальной плоскости, причем ось Y направлена вдоль склона, а ось X – поперек.

Отрезок траектории задается координатами его начала - x_1, y_1, z_1 и координатами его конца - x_2, y_2, z_2. Пусть α - угол наклона этого отрезка. Длину отрезка траектории обозначим s, длину проекции этого отрезка на горизонтальную плоскость – d.

$$s = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}\ ,$$

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2}\ , \qquad (1)$$

$$\sin\alpha = -\frac{\Delta z}{s}\ ,$$

($\Delta z = z_1 - z_2 < 0$ при движении вниз по склону)

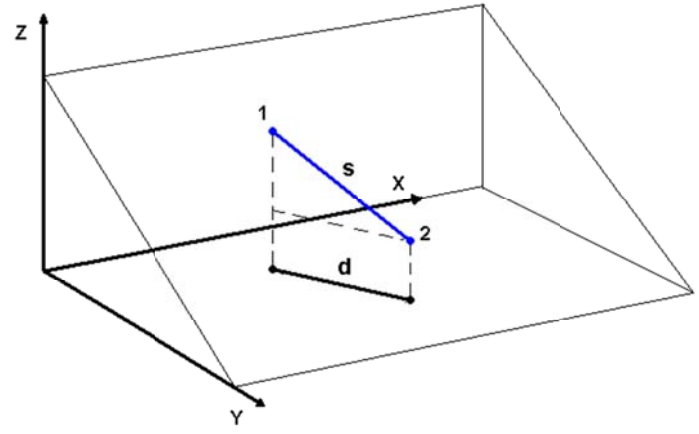

Рис. 4. Система координат в задаче о горнолыжнике

На прямолинейном отрезке траектории движение горнолыжника является равноускоренным.

Изменение кинетической энергии горнолыжника происходит за счет работы силы тяжести и работы силы трения,

$$\frac{mv_2^2}{2} - \frac{mv_1^2}{2} = -mg\Delta z - \mu mg \cdot \cos\alpha \cdot s\ , \qquad (2)$$

где m – масса, v – скорость горнолыжника, g – ускорение свободного падения, μ – коэффициент трения.

Если скорость горнолыжника в точке 1 известна, то его скорость в точке 2, с учетом соотношения (2), может быть найдена по формуле

$$v_2 = \sqrt{v_1^2 + 2g\left(z_1 - z_2 - \mu d\right)}, \qquad (3)$$

а время движения по отрезку траектории – по формуле

$$\Delta t = \frac{s}{(v_1 + v_2)/2}. \qquad (4)$$

Тогда общее время движения от точки A до точки B находится суммированием по всем отрезкам траектории.

Алгоритм

Алгоритм поиска траектории с наименьшим временем спуска методом Монте-Карло:

1. Рассчитывается время движения по случайной траектории. (Сюда входит выбор случайной траектории).
2. Если удалось достичь точки финиша, время спуска сравнивается с предыдущим наилучшим по времени результатом. (Точка финиша достижима не по любой траектории, так как сила трения может привести к преждевременной остановке лыжника).
3. Повторяются 1 и 2, пока не исследовано заданное количество траекторий.

Работа №1. Трансляция и запуск программ

Ознакомиться со средствами трансляции и запуска программ в ОС GNU/Linux:

1. Сохранить программу из Приложения А в файл sky_serial.c;
2. Командой

```
$ cc ski_serial.c -o ski_serial
```

подготовить исполняемый файл;

3. Запустить исполняемый файл ski_serial;
4. Замерить время исполнения файла ski_serial командой

```
$ time ski_serial
```

5. Внося изменения в программу ski_serial.c, для заданного числа отрезков из табл. 1 найти требуемое число траекторий, чтобы погрешность определения времени спуска была равна 10^{-6}.

6. Найти требуемое число отрезков и траекторий, чтобы общая погрешность нахождения траектории с минимальным временем спуска была равна 10^{-4}, найти времена спуска лыжника, времена исполнения программы, результаты занести в таблицу.

Таблица 1

Число отрезков	Время спуска	Число траек-торий	Погрешность	Время расчета
2				
2				
2				
2				
2				
3				
3				
3				
3				
3				
4				
4				
4				
4				
4				
4				
5				
5				
5				
5				
5				

Работа №2. Использование OpenMP

Ознакомиться со средствами создания параллельных программ OpenMP:

1. Определить число доступных вычислительных ядер процессора и информацию о них командой

```
$ cat /proc/cpuinfo
```

2. В программу Приложения А добавить директивы OpenMP, позволяющие использовать все ядра процессора одновременно.
3. Для данных таблицы 1 с погрешностью определения времени спуска 10^{-6} найти времена расчета.
4. Вычислить ускорение параллельного расчета

$$\text{ускорение} = \frac{\text{время расчета последовательного варианта}}{\text{время расчета параллельного варианта}}.$$

5. Результаты занести в табл. 2.

Таблица 2

Число отрезков	Число траекторий	Время последовательного расчета	Время параллельного расчета	Ускорение
2				
3				
4				
5				

Работа №3. Использование GNU SL

Ознакомиться со средствами библиотеки GNU Scientific Library[1]:

1. Создать файл gsl_uniform_example.c с указанным примером:

```
#include <stdio.h>
#include <gsl/gsl_rng.h>

int main (void)
{
  const gsl_rng_type * T;
  gsl_rng * r;

  int i, n = 10;

  gsl_rng_env_setup();

  T = gsl_rng_default;
```

[1] GNU Scientific Library v1.13.

```
  r = gsl_rng_alloc (T);

  for (i = 0; i < n; i++)
    {
      double u = gsl_rng_uniform (r);
      printf ("%.5f\n", u);
    }

  gsl_rng_free (r);

  return 0;
}
```
[2]

2. Откомпилировать файл gsl_uniform_example.c командой

```
$ cc gsl_uniform_example.c -lgsl
-lgslcblas -lm -o gsl_uniform_example
```

[2] URL: http://www.gnu.org/software/gsl/manual/html_node/Random-Number-Generator-Examples.html

3. Запустить исполняемый файл gsl_uniform_example для проверки использования библиотеки GNU SL;
4. Добавить в программу Приложения А директивы OpenMP и функции генерации случайных чисел библиотеки GNU SL;
5. Для данных таблицы 1 с погрешностью определения времени спуска 10^{-6} найти времена расчета.
6. Вычислить ускорение расчета

$$\text{ускорение} = \frac{\text{время расчета последовательного варианта}}{\text{время расчета параллельного } GSL \text{ варианта}}.$$

7. Результаты занести в табл. 3.

Таблица 3

Число отрезков	Число траекторий	Время последовательного расчета	Время GSL расчета	Ускорение
2				
3				
4				
5				

Работа №4. Использование Intel MKL

Ознакомиться со средствами библиотеки Intel MKL[3]:

1. Создать файл mkl_uniform_example.c с указанным примером:

```
#include <stdio.h>
#include "omp.h"
#include "mkl.h"

#define NPROC 2

int main( void )
{
  Double drn[10];
  VSLStreamStatePtr stream[NPROC];
  int i;

  for(i=0; i<NPROC; i++)
```

[3] Intel® Math Kernel Library 10.3.

```
    {
      vslNewStream( &stream[i],
VSL_BRNG_MCG31, 1);
      vslLeapfrogStream( stream[i], i,
NPROC );
    }

#pragma omp parallel private(drn, i)
  {
      vdRngUniform(
VSL_METHOD_DUNIFORM_STD ,
stream[omp_get_thread_num()], 10, drn,
0.0, 1.0 );
      for (i=0; i<10; i++)
          printf("%.5f\n", drn[i]);
  }

  for(i=0; i<NPROC; i++)
      vslDeleteStream( &stream[i]);

  return 0;
}
```


2. Откомпилировать файл mkl_uniform_example.c командами:

```
$ source /opt/intel/mkl/bin/mklvars.sh
ia32
$ cc -O2 -g -fopenmp -lmkl_blas95 -
lmkl_intel -lmkl_gnu_thread -lmkl_core
-liomp5 -lpthread -lm
mkl_uniform_example.c -o
mkl_uniform_example
```

3. Запустить исполняемый файл mkl_uniform_example для проверки использования библиотеки Intel MKL;
4. Добавить в программу Приложения А директивы OpenMP и функции генерации случайных чисел библиотеки Intel MKL;
5. Для данных таблицы 1 с погрешностью определения времени спуска 10^{-6} найти времена расчета.
6. Вычислить ускорение расчета

$$\text{ускорение} = \frac{\text{время расчета последовательного варианта}}{\text{время расчета параллельного } MKL \text{ варианта}}.$$

7. Результаты занести в табл. 4.

Табл. 4.

Число отрезков	Число траекторий	Время последовательного расчета	Время MKL расчета	Ускорение
2				
3				
4				
5				

Список литературы

1. Перышкин А. В., Гнутик Е. М. Физика. 9 кл.: Учебник для общеобразоват. учеб. заведений. – 2-е изд., испр. и доп. – М.:Дрофа, 2000. – 256 с.
2. Савельев И. В. Курс физики: учебник для студентов втузов: в 3 т. – М.: Наука;.Гл.ред.физ.-мат.лит., 1989. – .Т.1 : Механика. Молекулярная физика. – 350 с.
3. Хемминг Р.В. Численные методы для научных работников и инженеров. – М., 1972. – 400 с.
4. Язык C/C++
   URL: http://www.codenet.ru/progr/cpp/
5. OpenMP Blaise Barney, Lawrence Livermore National Laboratory
   URL: https://computing.llnl.gov/tutorials/openMP/
6. GCC 4.5.1 manual
   URL: http://gcc.gnu.org/onlinedocs/gcc-4.5.1/gcc/
7. GNU Scientific Library – Reference Manual
   URL: http://www.gnu.org/software/gsl/manual/html_node/

8. Intel® Math Kernel Library for Linux* OS User's Guide
   URL: http://software.intel.com/sites/products/documentation/hpc/mkl/userguides/mkl_userguide_lnx.pdf
9. Vector Statistical Library Notes
   URL: http://software.intel.com/sites/products/documentation/hpc/mkl/vsl3/vslnotesv6.pdf

Приложение А

```
/* ski_serial.c  Найти время спуска из
одной точки в другую на плоском склоне
*/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

/* Координаты точки старта */
#define     XS   0.0
#define     YS   0.0

/* Координаты точки финиша */
#define XF  100.0
#define     YF   100.0

/* Параметры склона */
#define TGFI     -0.2 /* Тангенс угла
наклона плоскости */
```



```
#define     MJU  0.0  /* Коэффициент
трения */

/* Начальные параметры */
#define     V0   0.0
#define     G    9.8

/* Параметры траектории */
#define     N    2    /* Число отрезков
траектории */
#define     JMAX 100000    /* Число ис-
пытаний серии */
#define     NSER 10   /* Число серий */

/* Задать вид поверхности */
double z(double x, double y){
  return TGFI*y;
}

int main( void ){
```

```
  double
  dx,dy,dz,d,s,v1,v12,v22,v2,t;

  printf("tgfi=%8.6g mju=%8.6g\n\n",
TGFI, MJU);

  /* Прямолинейная траектория */

  dx=XF-XS;  dy=YF-YS; dz=z(XF, YF)-
z(XS, YS);
  d=sqrt(dx*dx+dy*dy);
  s=sqrt(dx*dx+dy*dy+dz*dz);
  v1=V0; v12=V0*V0;
  v22=v12-2.0*G*(dz+MJU*d);

  if (v22<0) {
    printf("Точка финиша недостижи-
ма\n"); return 0;
  }

  v2=sqrt(v22);
  t=2*s/(v1+v2);
```



```
  /* Подготовка вывода */

  printf("Движение по прямой AB:  xf=
%8.6g   yf= %8.6g  vf= %8.6g  t=
%8.6g\n\n",XF,YF,v2,t);

  printf("Движение по траектории из %d
отрезков\n\n",N);
  printf("   k     tmin       vf
x1       y1      x2       y2      x3
y3\n");

  return 0;
}
```

Содержание

Введение 3

Модель 5

Задача о горнолыжнике 5

Обсуждение задачи 6

Физическая модель 8

Алгоритм 12

Работа №1. Трансляция и запуск программ 13

Работа №2. Использование OpenMP 16

Работа №3. Использование GNU SL 18

Работа №4. Использование Intel MKL 22

Список литературы 27

Приложение А 29

Учебное издание

Сивков Анатолий Михайлович

Сивков Дмитрий Анатольевич

ИСПОЛЬЗОВАНИЕ ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЕЛ НА МНОГОЯДЕРНЫХ КОМПЬЮТЕРАХ

Лабораторный практикум

Авторская редакция

Подписано в печать 15.12.10. Формат 60x84$^{1/16}$.

Печать офсетная. Усл. печ. л. 1,98. Уч.-изд. л. 0,3.

Тираж 25 экз. Заказ №

426034, Ижевск, Университетская, 1, корп. 4.

Издательство «Удмуртский университет»