# Технологии параллельных систем и распределенных вычислений

# Лабораторная работа №7.

Рассмотрим задачу нахождения концентрации загрязняющего вещества в озере через определенного времени. Задача нахождения концентрации загрязнителя моделируется с помощью обыкновенного дифференциального уравнения вида:

(1)

Начальное загрязнение озера составляет 107 ед./м3, в то время как допустимый уровень составляет только 5∙106 ед. / м3. Концентрация загрязнителя снижается с поступлением пресной воды в озеро, причем известно, что скорость очищения озера составляет *С*= 0,06 ед/неделя.

В работе предлагается решить это уравнение численно с помощью метода Рунге-Кутта 4-го порядка. Так же это уравнение имеет аналитическое решение:

. (2)

Код на языке С++, который находит решения уравнение (1) методом Рунге-Кутта приведен в листинге ниже. Пример выполнения программы приведен на рис. 1.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define Y0 1e7

#define C0 0.06

double rk4(double(\*f)(double,double),double dx,double x,double y)

{

double k1 = dx \* f(x, y),

k2 = dx \* f(x + dx / 2, y + k1 / 2),

k3 = dx \* f(x + dx / 2, y + k2 / 2),

k4 = dx \* f(x + dx, y + k3);

return y + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;

}

double rate(double x, double y)

{

return -C0\*y; // правая часть уравнения вида y'=f(x,y)

}

int main(void)

{

double x0 = 0; // Начальная точка решения

double x1 = 52; // Конечная точка решения (один год)

double dx = 1e-4; // Шаг

int n = 1 + (x1 — x0)/dx; // Число шагов

double \*y = new double[n];

y[0] = Y0; // Начальное условие

for (int i = 1; i < n; i++)

y[i] = rk4(rate, dx, x0 + dx \* (i - 1), y[i-1]);

printf("x\ty\trel. err.\n------------\n");

for (int i = 0; i < n; i += 10)

{

double x = x0 + dx \* i;

double y2 = Y0\*exp(-C0\*x);

// Точное решение и оценка ошибки

printf("%g\t%g\t%g\n", x, y[i], y[i]/y2 - 1);

}

delete []y;

return 0;

}

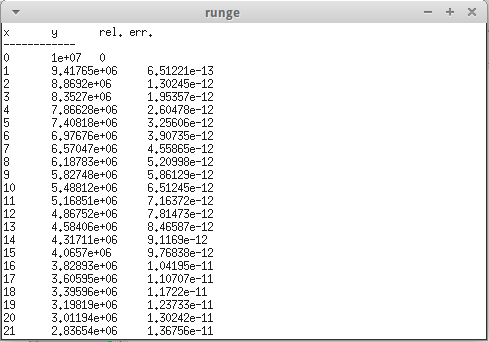
**Задание**:

1. Выполнить распараллеливание кода программы с помощью технологии OpenMP;

2. Оценить выигрыш по времени выполнения программы от ее распараллеливания;

3. Сохранить полученное решение в сsv файл. Открыть файл в табличном редакторе и построить графики полученного и точного решения.

4. Определить момент времени, когда концентрация загрязнителя в озере снизится до 5∙106ед. / м3

Рис. 1 — Пример работы программы