

# 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:

$$\frac{d}{dt}y(t) + Cy(t) = 0$$
$$y(0) = y_0$$
(1)

Начальное загрязнение озера составляет  $10^7$  ед./м<sup>3</sup>, в то время как допустимый уровень составляет только  $5 \cdot 10^6$  ед. / м<sup>3</sup>. Концентрация загрязнителя снижается с поступлением пресной воды в озеро, причем известно, что скорость очищения озера составляет  $C = 0,06$  ед/неделя.

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

$$y(t) = y_0 \exp(-Ct).$$
(2)

Код на языке C++, который находит решения уравнение (1) методом Рунге-Кутты 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
```

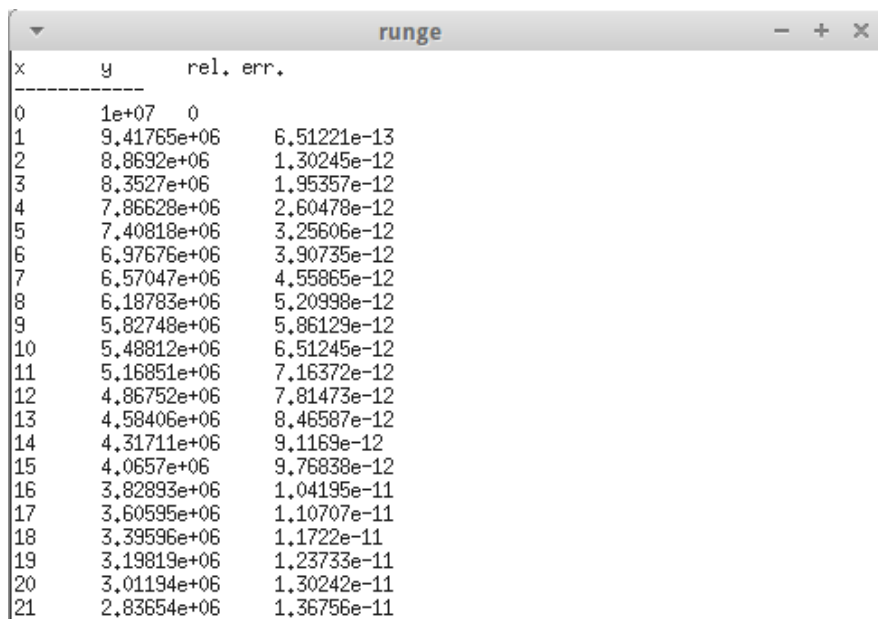
```

for (int i = 1; i < n; i++)
    y[i] = rk4(rate, dx, x0 + dx * (i - 1), y[i-1]);
printf("x\t y\t rel. 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 csv файл. Открыть файл в табличном редакторе и построить графики полученного и точного решения.
4. Определить момент времени, когда концентрация загрязнителя в озере снизится до  $5 \cdot 10^6$  ед. / м<sup>3</sup>



x	y	rel. err.
0	1e+07	0
1	9.41765e+06	6.51221e-13
2	8.8692e+06	1.30245e-12
3	8.3527e+06	1.95357e-12
4	7.86628e+06	2.60478e-12
5	7.40818e+06	3.25606e-12
6	6.97676e+06	3.90735e-12
7	6.57047e+06	4.55865e-12
8	6.18783e+06	5.20998e-12
9	5.82748e+06	5.86129e-12
10	5.48812e+06	6.51245e-12
11	5.16851e+06	7.16372e-12
12	4.86752e+06	7.81473e-12
13	4.58406e+06	8.46587e-12
14	4.31711e+06	9.1169e-12
15	4.0657e+06	9.76838e-12
16	3.82893e+06	1.04195e-11
17	3.60595e+06	1.10707e-11
18	3.39596e+06	1.1722e-11
19	3.19819e+06	1.23733e-11
20	3.01194e+06	1.30242e-11
21	2.83654e+06	1.36756e-11

Рис. 1 — Пример работы программы