

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Кафедра системного программирования

Фефелов Алексей Андреевич

Обработка потоков числовой информации и поиск предельных циклов Пуанкаре

Курсовая работа

Научный руководитель:
ст. преп. Я. А. Кириленко

Санкт-Петербург
2018

Содержание

Введение	3
1. Постановка задачи	4
2. Описание решения	5
3. Заключение	8
Список литературы	9

Введение

В 1900 году немецкий математик Давид Гильберт, выступая на II международном конгрессе математиков в Париже, представил математическому сообществу 23 проблемы, которые, по его мнению, должны были быть решены в течение 100 лет. Однако, на сегодняшний день решены лишь 16 проблем, ещё 2 являются некорректными математически, 2 не решены никак, а 3 — только для некоторых случаев.

Большой практический интерес для математиков составляет 16 проблема, а именно, частный случай её второй (дифференциальной) части, который заключается в нахождении максимального числа, а так же поиске расположения предельных циклов Пуанкаре системы дифференциальных уравнений вида

$$\begin{cases} \dot{x} = \alpha_1 x + \beta_1 y + ax_1^2 + b_1 xy + c_1 y^2 \\ \dot{y} = \alpha_2 x + \beta_2 y + ax_2^2 + b_2 xy + c_2 y^2 \end{cases}, \quad (1)$$

что равносильно системе

$$\begin{cases} \dot{x} = x^2 + xy + y \\ \dot{y} = ax^2 + bxy + cy^2 + \alpha x + \beta y \end{cases} \quad (2)$$

Этой проблемой уже занимались многие математики, в том числе А. Н. Колмагоров, В. И. Арнольд, Г. А. Леонов, Н. В. Кузнецов, И. Г. Бурова и др., однако решить эту проблему математически ещё пока никому не удалось.

Нам, при поддержке фонда Botan Investmens и НИВЦ МГУ, было предложено разработать программу для поиска местоположения предельных циклов данной системы и визуализации результатов.

По условиям гранта, программа для поиска местоположения и максимально возможного количества предельных циклов должна была быть разработана на языке программирования С и распараллелена с использованием интерфейса MPI¹, требований к второй (демонстрационной) программе предъявлено не было.

¹Message Passing Interface (MPI), <https://www.open-mpi.org>

1. Постановка задачи

- Разработать параллельную программу для автоматизированного поиска предельных циклов на языке программирования C с использованием интерфейса MPI для распараллеливания.
- Разработать программу с удобным пользовательским интерфейсом для демонстрации полученных результатов.
- Протестировать программу на параллельном кластере СПбГУ.
- Произвести запуск программы на суперкомпьютерном комплексе Ломоносов-1² и получить не менее $16 * 10^7$ результатов (под результатом понимается наличие/отсутствие предельных циклов в окрестности особых точек, и, по возможности, их расположение с точностью до 10^{-8} при заданных параметрах уравнения).
- Перед запуском программы на суперкомпьютере провести performance-тестирование при разных настройках компилятора и скомпилировать программу наиболее оптимальным образом.

²Ломоносов-1 — суперкомпьютер, установленный в Московском государственном университете, <https://www.msu.ru/lomonosov/science/computer.html>

2. Описание решения

После изучения работ [2],[1] было принято решение использовать метод Рунге-Кутты для нахождения численных решений заданной системы дифференциальных уравнений. Изначально планировалось использование различных алгоритмов распараллеливания, в том числе и распараллеливание по начальным данным, однако, впоследствии, выяснилось, что при таком подходе сильно падает производительность (это происходит из-за большого количества map-reduce фаз); поэтому было принято решение отказаться от такого подхода и использовать только распараллеливание по параметрам a, b, c, α и β . Этот подход оказался намного лучше, так как во-первых, мы полностью исключили map-reduce фазы (каждый процессор сохраняет результаты в свой файл по мере их получения), во-вторых, в силу огромного количества входных данных, в течение всего времени работы программы все процессоры суперкомпьютера (около 90 000) будут оставаться загруженными.

Для поиска предельных циклов был разработан следующий алгоритм:

- Задаем сетку в области параметров
- Для каждого набора параметров:
 - Находим особые точки системы ДУ
 - В окрестности каждой особой точки ищем циклы:
 - * Ставим n задач Коши
 - * Определяем "поведение" решений
 - * Если "поведение" разное — ищем начальные данные в окрестности цикла

Кроме того, нам было предложено использовать библиотеку MPFR³ для высокоточных вычислений, однако от этой идеи мы отказались по следующим причинам: во-первых, стандартные типы данных языка

³MPFR — библиотека для вычисления значения арифметических выражений с плавающей запятой с высокой точностью, <http://www.mpfr.org/>

программирования С обеспечивают необходимую точность; во-вторых, при использовании этой библиотеки резко падает производительность; в-третьих, код становится плохо читаемым и сопровождать его практически невозможно. Ниже приведен пример кода с использованием библиотеки MPFR (пример взят с официального сайта MPFR).

```
int main ()
{
    unsigned int i;
    mpfr_t s, t, u;

    mpfr_init2 (t, 200);
    mpfr_set_d (t, 1.0, MPFR_RNDD);
    mpfr_init2 (s, 200);
    mpfr_set_d (s, 1.0, MPFR_RNDD);
    mpfr_init2 (u, 200);

    for (i = 1; i <= 100; i++) {
        mpfr_mul_ui (t, t, i, MPFR_RNDU);
        mpfr_set_d (u, 1.0, MPFR_RNDD);
mpfr_div (u, u, t, MPFR_RNDD);
        mpfr_add (s, s, u, MPFR_RNDD);
    }
    printf ("Sum is ");
    mpfr_out_str (stdout, 10, 0, s, MPFR_RNDD);
    putchar ('\n');
    mpfr_clear (s);
    mpfr_clear (t);
    mpfr_clear (u);
    return 0;
}
```

Нетрудно догадаться, во что преобразуются длинные арифметические выражения...

В качестве языка для реализации демонстрационной программы был выбран C#, так как в нем есть библиотека ZedGraph⁴, которая позволяет легко создать интерактивный график (рис. 1).

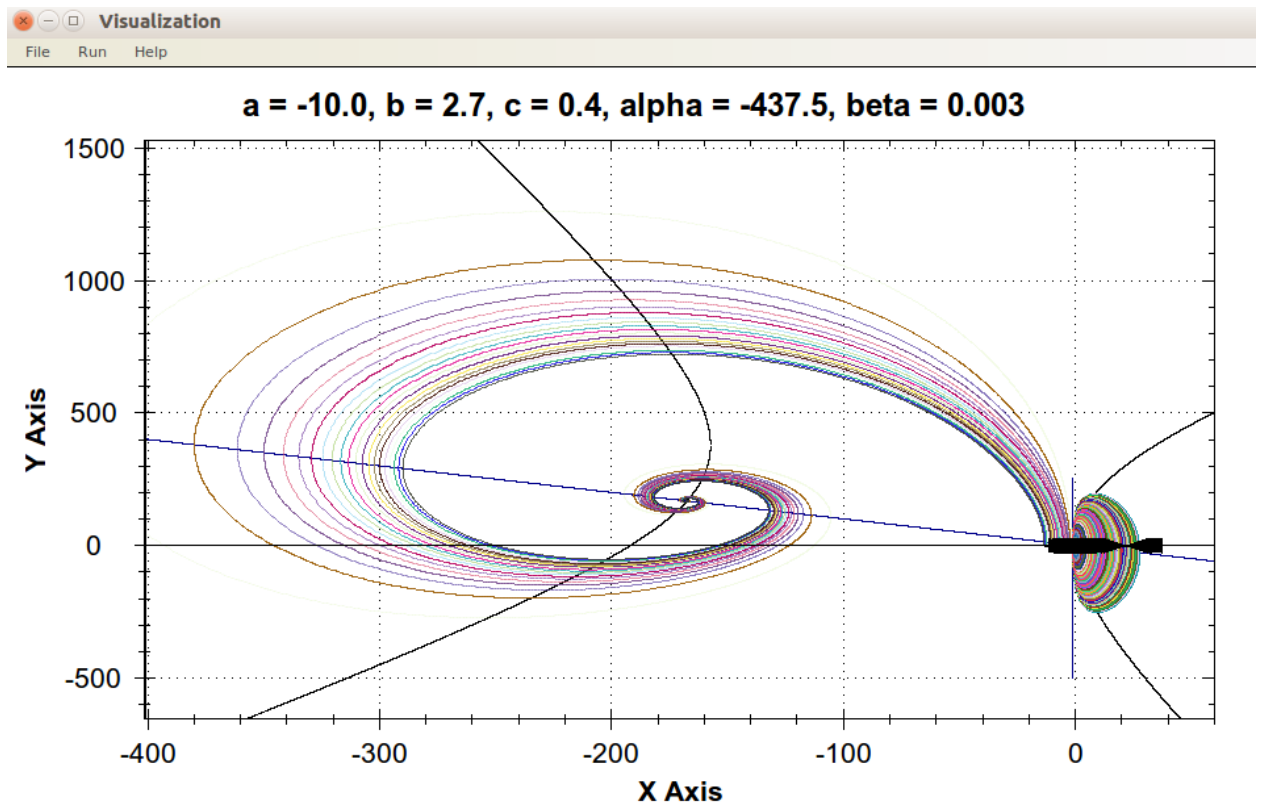


Рис. 1

Программа была протестированна и отлажена на параллельном кластере с использованием 2, 3 и 4 компьютеров. Катастрофических погрешностей, описанных в статье [3] не наблюдалось. На тестовом наборе параметров, который был взят из этой же статьи, метод Рунге-Кутты показал тот же результат, что и метод Гира с автоматическим выбором шага, что опровергает полученные в данной статье результаты.

Кроме того, в связи с лимитированным доступом к ресурсам суперкомпьютера, была проведена подготовка к performance-тестированию и написан sh-скрипт, который выполняет 20 запусков с разными настройками компилятора и сохраняет результаты в файл.

⁴ZedGraph — <https://sourceforge.net/projects/zedgraph/>

3. Заключение

- Разработана параллельная программа для поиска предельных циклов на языке программирования C с использованием интерфейса MPI для распараллеливания.
- Разработана программа для демонстрации некоторых полученных результатов.
- Параллельная программа протестирована на параллельном кластере СПбГУ.
- Ожидается получение доступа к ресурсам СК Ломоносов-1 и запуск программы в июне с.г.
- Написан sh-скрипт для performance-тестирования на СК Ломоносов-1.

Список литературы

- [1] G.A. Leonov. Four Limit Cycles in Quadratic Two-Dimensional Systems with a Perturbed First-Order Weak Focus.
- [2] Kuznetsov N.V. Kuznetsova O.A. Leonov G.A. Visualization of four normal size limit cycles in two-dimensional polynomial quadratic system.
- [3] Кузнецов Н.В. Бурова И.Г. Александров К.Д. Визуализация четырех предельных циклов двумерных квадратичных систем в пространстве параметров.