

# Обработка потоков числовой информации и поиск предельных циклов Пуанкаре

Алексей Фефелов  
группа 244

научный руководитель Я. А. Кириленко

22.05.2018г

# 16 проблема Гильберта

Количество и расположение предельных циклов дифференциального уравнения (системы) на плоскости

## Частный случай

$$\dot{x} = ax_1^2 + b_1xy + c_1y^2 + \alpha_1x + \beta_1y$$

$$\dot{y} = ax_2^2 + b_2xy + c_2y^2 + \alpha_2x + \beta_2y$$

что равносильно системе

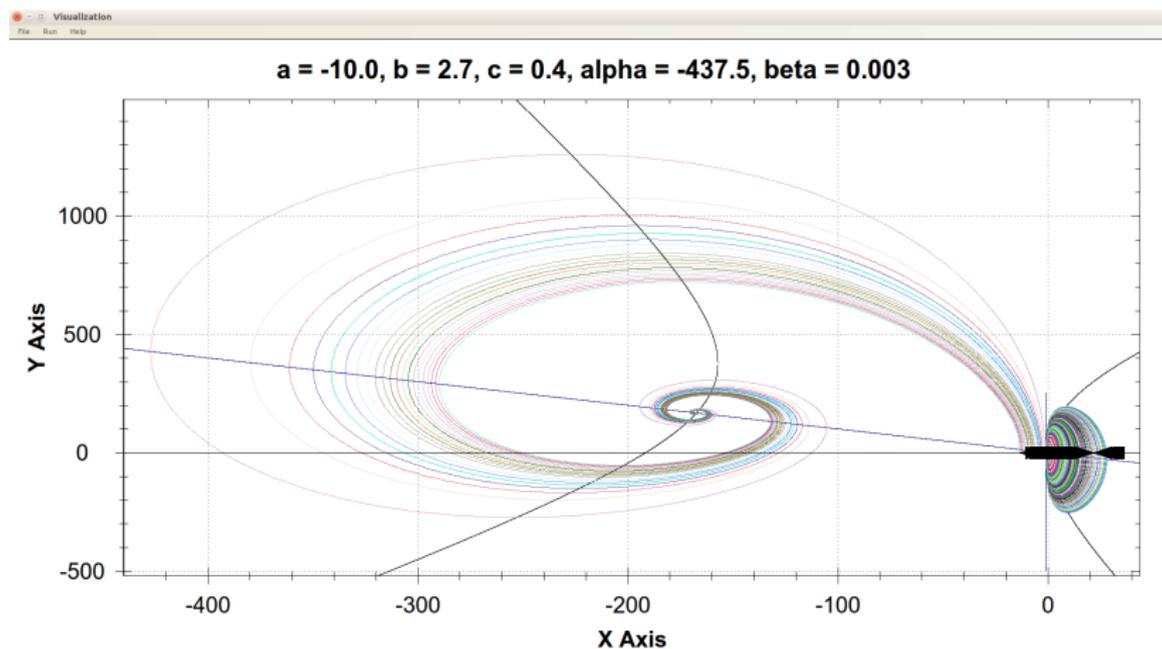
$$\dot{x} = x^2 + xy + y$$

$$\dot{y} = ax^2 + bxy + cy^2 + \alpha x + \beta y$$

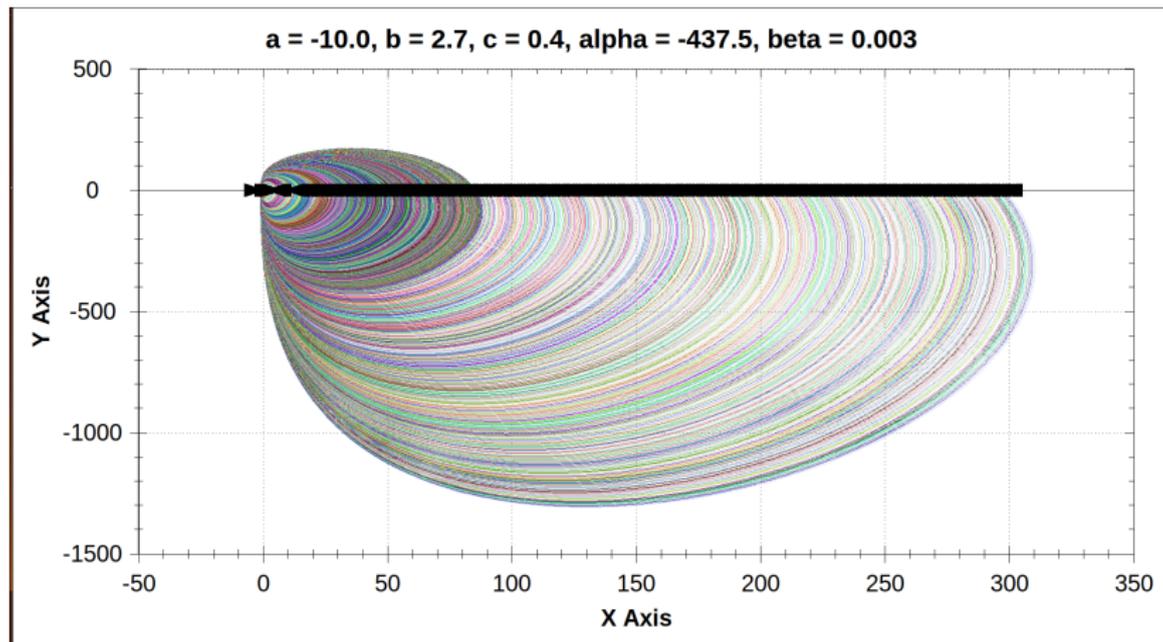
# Постановка задачи

- ▶ Разработать параллельную программу для поиска предельных циклов
- ▶ Разработать программу для демонстрации полученных результатов
- ▶ Протестировать программу на параллельном кластере СПбГУ
- ▶ Произвести запуск программы на суперкомпьютерном комплексе Ломоносов-1
- ▶ Перед запуском программы на суперкомпьютере провести performance-тестирование при разных настройках компилятора

# Демонстрационная программа

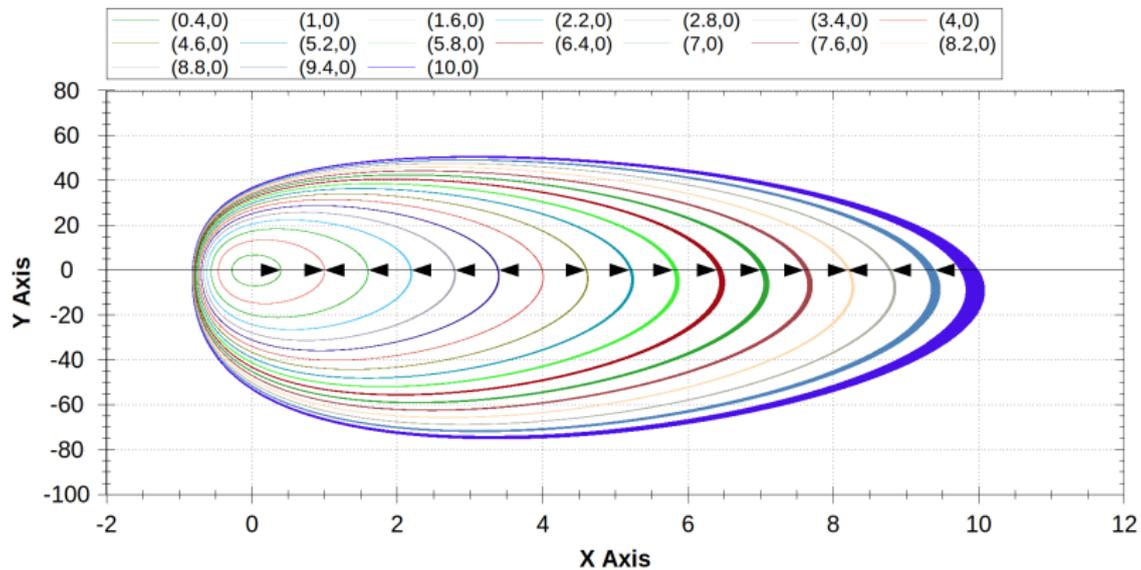


# Фазовый портрет (1)



# Фазовый портрет (2)

**$a = -10.0$ ,  $b = 2.7$ ,  $c = 0.4$ ,  $\alpha = -437.5$ ,  $\beta = 0.003$**



# Основные идеи

- ▶ Метод Рунге-Кутты для нахождения численных решений
- ▶ Интерфейс MPI
- ▶ Распараллеливание по данным

# Метод Рунге-Кутта

- ▶ Позволяет находить численное решение задачи Коши
- ▶ Последовательное вычисление точек
- ▶ Зная значение решения З.К. в момент времени  $t_0$ , находим значение решения в момент времени  $t_0 + h$

# Интерфейс MPI (Message Passing Interface)

- ▶ Позволяет обмениваться информацией между процессами
- ▶ Поддерживается суперкомпьютером Ломоносов-1
- ▶ Есть реализация на C

# Идея решения

- ▶ 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

# Алгоритм распараллеливания (1)

- ▶ Распараллеливание по параметрам  $a, b, c, \alpha, \beta$
- ▶ Параллельное решение разных задач Коши

## Плюсы

- ▶ Все процессоры загружены

## Минусы

- ▶ Много map-reduce фаз
- ▶ Как следствие, низкая производительность

## Алгоритм распараллеливания (2)

- ▶ Распараллеливание только по параметрам  $a, b, c, \alpha, \beta$

### Плюсы

- ▶ Все процессоры по прежнему загружены
- ▶ Нет map-reduce фаз
- ▶ Более высокая производительность

### Минусы

- ▶ Необходим большой набор исходных данных

# Почему мы отказались от библиотек для высокоточных вычислений

- ▶ При использовании метода Рунге-Кутты погрешности не критичны
- ▶ Резко падает производительность
- ▶ Код становится плохо читаемым

# Сравнение производительности, время указано в секундах

Таблица: + -flt0, -march=native

	O0	O1	O2	O3	O <sub>s</sub>
Мат. ожидание	78.0	40.2	31.5	30.5	31.2
Дисперсия	0.2	0.1	0.1	0.1	0.1

# Результаты

- ▶ Разработана параллельная программа для поиска предельных циклов
- ▶ Разработана программа для демонстрации полученных результатов
- ▶ Программа протестирована на параллельном кластере СПбГУ
- ▶ Запуск программы на суперкомпьютерном комплексе Ломоносов-1 запланирован на конец мая – начало июня
- ▶ Написан скрипт для performance-тестирования