Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики
Высшая школа прикладной математики и вычислительной физики

Параллельные вычислени

Лабораторная работа №1
«Решение задачи Коши для системы линейных неоднородных ОДУ методом Пикара»

Работу выполнила:
студентка II курса
магистратуры
Добрецова Е.В.
Группа:
5040102/40101
Преподаватель:
Козлов К.Н.

Санкт-Петербург
2025

Содержание

1. Постановка задачи и её формализация

Дана система линейных неоднородных обыкновенных дифференциальных уравнений с постоянными коэффициентами:

$$\frac{d\boldsymbol{y}}{dt} = A\boldsymbol{y} + \boldsymbol{f}(t), \tag{1}$$

и следующими начальными условиями:

$$y_i = 0, \quad i = \overline{1, n}. \tag{2}$$

Здесь $\boldsymbol{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \dots \\ y_n(t) \end{pmatrix}$, $\boldsymbol{f}(t) = \begin{pmatrix} f_1(t) \\ f_2(t) \\ \dots \\ f_n(t) \end{pmatrix}$, матрица A имеет размерность $n \times n$ и является нижнетреугольной ($a_{ij} = 0$ при $i > j$) с единичными ненулевыми элементами.

Необходимо решить задачу методом последовательных приближений Пикара с применением циклической схемы распределения уравнений по процессам для обеспечения равномерной загрузки процессоров, осуществляя программирование на языке C с применением технологии OpenMP. Также требуется провести анализ метода путём следующих исследований:

- зависимость времени выполнения от размера системы;
- зависимость времени выполнения от числа параллельных процессов/потоков;
- зависимость ускорения от числа параллельных процессов/потоков;
- зависимость эффективности параллелизации от числа параллельных процессов/потоков.

2. Алгоритм метода Пикара и условия его применимости

2.1. Описание метода

Метод Пикара для задачи (1) с начальными условиями (2) определяет последовательность приближений $\boldsymbol{y}^{(k)}(t)$ следующим образом:

$$\boldsymbol{y}^{(k+1)}(t) = \boldsymbol{y}_0 + \int_{t_0}^{t} \left(A\boldsymbol{y}^{(k)}(\tau) + \boldsymbol{f}(\tau)\right) d\tau, \quad k = 0, 1, 2, \dots$$

Процесс продолжается до выполнения критерия сходимости:

$$\left\|\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}\right\| < \varepsilon$$

Интеграл вычисляется численно по квадратурной формуле трапеций на каждом временном интервале $[t_m, t_{m+1}]$:

$$\boldsymbol{y}_{m+1}^{(k+1)} = \boldsymbol{y}_m^{(k+1)} + \frac{\Delta t}{2}\left[A\boldsymbol{y}_m^{(k)} + \boldsymbol{f}(t_m) + A\boldsymbol{y}_{m+1}^{(k)} + \boldsymbol{f}(t_{m+1)})\right],$$

где $\Delta t = t_{m+1} - t_m,\ \boldsymbol{y}_m^{(k)}$ - приближение в узле t_m на k-й итерации.

2.2. Параллелизация вычислений

На каждой итерации вычисления правых частей для разных компонент системы независимы, так как они используют значения предыдущего приближения $\boldsymbol{y}^{(k)}$, которые на этом этапе полностью известны. Поэтому в данной задаче применяется декомпозиция по данным (parallelism across space). При таком способе работы каждая компонентная функция $y_i(t)$ считается независимо, уравнения равномерно распределяются между потоками OpenMP. Используется циклическая схема распределения.
Каждый поток независимо выполняет:

1. вычисление своих компонент
$$A\boldsymbol{y}^{(k)} + \boldsymbol{f}(t),$$

2. применение квадратурной формулы трапеций для этих компонент.

Синхронизация необходима только между итерациями k и $k+1$, когда все потоки должны завершить обновление своих компонент.

2.3. Стандарт OpenMP

3. Предварительный анализ задачи

4. Проверка условий применимости метода

5. Тестовый пример с детальными расчётами для задачи малой размерности

6. Подготовка контрольных тестов для иллюстрации метода

7. Модульная структура программы

8. Численный анализ решения задачи

9. Выводы