

Численное решение систем уравнений с запаздыванием на видеопроцессорах

Дмитрий Глызин
glyzin@gmail.com
<http://ncycle.org>

Ярославский госуниверситет им. П.Г. Демидова

Таруса, 19 октября 2011 г.

Задача

Рассмотрим уравнение в частных производных с запаздыванием

$$\frac{\partial u(x, t)}{\partial t} = \lambda(1 - u(x, t - h))u(x, t) + d \frac{\partial^2 u(x, t)}{\partial x^2} \quad (1)$$

на отрезке $x \in [0, 1]$, при $t \geq 0$, с краевыми условиями Неймана и начальным условием

$$u(x, t) = \varphi(x, t), \quad 0 < x < 1, \quad -h \leq t \leq 0. \quad (2)$$

Уравнение (1) — уравнение Хатчинсона с диффузией, или уравнение Колмогорова-Фишера с запаздыванием.

Задача

Рассмотрим уравнение в частных производных с запаздыванием

$$\frac{\partial u(x, t)}{\partial t} = \lambda(1 - u(x, t - h))u(x, t) + d \frac{\partial^2 u(x, t)}{\partial x^2} \quad (1)$$

на отрезке $x \in [0, 1]$, при $t \geq 0$, с краевыми условиями Неймана и начальным условием

$$u(x, t) = \varphi(x, t), \quad 0 < x < 1, \quad -h \leq t \leq 0. \quad (2)$$

Уравнение (1) — уравнение Хатчинсона с диффузией, или уравнение Колмогорова-Фишера с запаздыванием.

Метод

Для численного решения задачи (1)-(2) воспользуемся следующей комбинацией методов:

- Дискретизация по пространственной переменной ("метод прямых")
- Сведение к неавтономной системе дифференциальных уравнений ("метод шагов")
- Применение явной схемы Рунге-Кутты с плотным выводом (метод Owren, Zennaro пятого порядка)

Метод

Для численного решения задачи (1)-(2) воспользуемся следующей комбинацией методов:

- Дискретизация по пространственной переменной ("метод прямых")
- Сведение к неавтономной системе дифференциальных уравнений ("метод шагов")
- Применение явной схемы Рунге-Кутты с плотным выводом (метод Owren, Zennaro пятого порядка)

Метод

Для численного решения задачи (1)-(2) воспользуемся следующей комбинацией методов:

- Дискретизация по пространственной переменной ("метод прямых")
- Сведение к неавтономной системе дифференциальных уравнений ("метод шагов")
- Применение явной схемы Рунге-Кутты с плотным выводом (метод Owren, Zennaro пятого порядка)

Метод

Для численного решения задачи (1)-(2) воспользуемся следующей комбинацией методов:

- Дискретизация по пространственной переменной ("метод прямых")
- Сведение к неавтономной системе дифференциальных уравнений ("метод шагов")
- Применение явной схемы Рунге-Кутты с плотным выводом (метод Owren, Zennaro пятого порядка)

Метод прямых

Разобьем отрезок $[0, 1]$ равномерной сеткой из N точек, обозначим $\Delta x = 1/(N - 1)$, заменим производную на конечную разность:

$$\frac{\partial^2 u(x, t)}{\partial x^2} \approx \frac{u(x + \Delta x, t) + u(x - \Delta x, t) - 2u(x, t)}{\Delta x^2} \quad (3)$$

Метод шагов

Зная решение $u_j(t) = \psi_j(t)$ на промежутке $[-h, nh]$, найдем его продолжение на $[nh, (n+1)h]$ как решение неавтономной системы

$$\dot{u}_j(t) = \lambda(1 - \psi_j(t-h))u(t) + \frac{d}{\Delta x}(u_{j-1}(t) - 2u_j(t) + u_{j+1}(t)) \quad (4)$$

Owren, Zennaro 5(4)

- Вложенная пара явных методов Рунге-Кутты
- 8 стадий
- Приближения порядка 5 (для продолжения) и 4 (для контроля шага)
- Плотный вывод порядка 5

Owren, Zennaro 5(4)

- Вложенная пара явных методов Рунге-Кутты
- 8 стадий
- Приближения порядка 5 (для продолжения) и 4 (для контроля шага)
- Плотный вывод порядка 5

Owren, Zennaro 5(4)

- Вложенная пара явных методов Рунге-Кутты
- 8 стадий
- Приближения порядка 5 (для продолжения) и 4 (для контроля шага)
- Плотный вывод порядка 5

Owren, Zennaro 5(4)

- Вложенная пара явных методов Рунге-Кутты
- 8 стадий
- Приближения порядка 5 (для продолжения) и 4 (для контроля шага)
- Плотный вывод порядка 5

Owren, Zennaro 5(4)

- Вложенная пара явных методов Рунге-Кутты
- 8 стадий
- Приближения порядка 5 (для продолжения) и 4 (для контроля шага)
- Плотный вывод порядка 5

Параллелизм

- Методы Рунге-Кутты не обладают параллелизмом внутри метода: стадии нельзя сделать независимыми
- Нас интересует ускорение вычисления единственной траектории, поэтому параллелизм по траекториям также отсутствует
- Вычисление правых частей хорошо параллелизуется, поскольку связаны лишь соседние уравнения
- При разделении работы на несколько больших блоков (случай многоядерных центральных процессоров) потери на синхронизацию малы
- В силу особенностей графических процессоров для эффективного использования вычислительной мощности требуется доработка способа разбиения на блоки

Параллелизм

- Методы Рунге-Кутты не обладают параллелизмом внутри метода: стадии нельзя сделать независимыми
- Нас интересует ускорение вычисления единственной траектории, поэтому параллелизм по траекториям также отсутствует
- Вычисление правых частей хорошо параллелизуется, поскольку связаны лишь соседние уравнения
- При разделении работы на несколько больших блоков (случай многоядерных центральных процессоров) потери на синхронизацию малы
- В силу особенностей графических процессоров для эффективного использования вычислительной мощности требуется доработка способа разбиения на блоки

Параллелизм

- Методы Рунге-Кутты не обладают параллелизмом внутри метода: стадии нельзя сделать независимыми
- Нас интересует ускорение вычисления единственной траектории, поэтому параллелизм по траекториям также отсутствует
- Вычисление правых частей хорошо параллелизуется, поскольку связаны лишь соседние уравнения
- При разделении работы на несколько больших блоков (случай многоядерных центральных процессоров) потери на синхронизацию малы
- В силу особенностей графических процессоров для эффективного использования вычислительной мощности требуется доработка способа разбиения на блоки

Параллелизм

- Методы Рунге-Кутты не обладают параллелизмом внутри метода: стадии нельзя сделать независимыми
- Нас интересует ускорение вычисления единственной траектории, поэтому параллелизм по траекториям также отсутствует
- Вычисление правых частей хорошо параллелизуется, поскольку связаны лишь соседние уравнения
- При разделении работы на несколько больших блоков (случай многоядерных центральных процессоров) потери на синхронизацию малы
- В силу особенностей графических процессоров для эффективного использования вычислительной мощности требуется доработка способа разбиения на блоки

Параллелизм

- Методы Рунге-Кутты не обладают параллелизмом внутри метода: стадии нельзя сделать независимыми
- Нас интересует ускорение вычисления единственной траектории, поэтому параллелизм по траекториям также отсутствует
- Вычисление правых частей хорошо параллелизуется, поскольку связаны лишь соседние уравнения
- При разделении работы на несколько больших блоков (случай многоядерных центральных процессоров) потери на синхронизацию малы
- В силу особенностей графических процессоров для эффективного использования вычислительной мощности требуется доработка способа разбиения на блоки

Архитектура GPU

- **Массивный параллелизм: сотни вычислительных ядер и десятки тысяч нитей**
- SIMD/SIMT-архитектура
- Разбиение нитей на блоки, внутри которых между нитями возможен обмен через быструю общую память с эффективной синхронизацией
- Размер общей памяти крайне ограничен

Архитектура GPU

- Массивный параллелизм: сотни вычислительных ядер и десятки тысяч нитей
- SIMD/SIMT-архитектура
- Разбиение нитей на блоки, внутри которых между нитями возможен обмен через быструю общую память с эффективной синхронизацией
- Размер общей памяти крайне ограничен

Архитектура GPU

- Массивный параллелизм: сотни вычислительных ядер и десятки тысяч нитей
- SIMD/SIMT-архитектура
- Разбиение нитей на блоки, внутри которых между нитями возможен обмен через быструю общую память с эффективной синхронизацией
- Размер общей памяти крайне ограничен

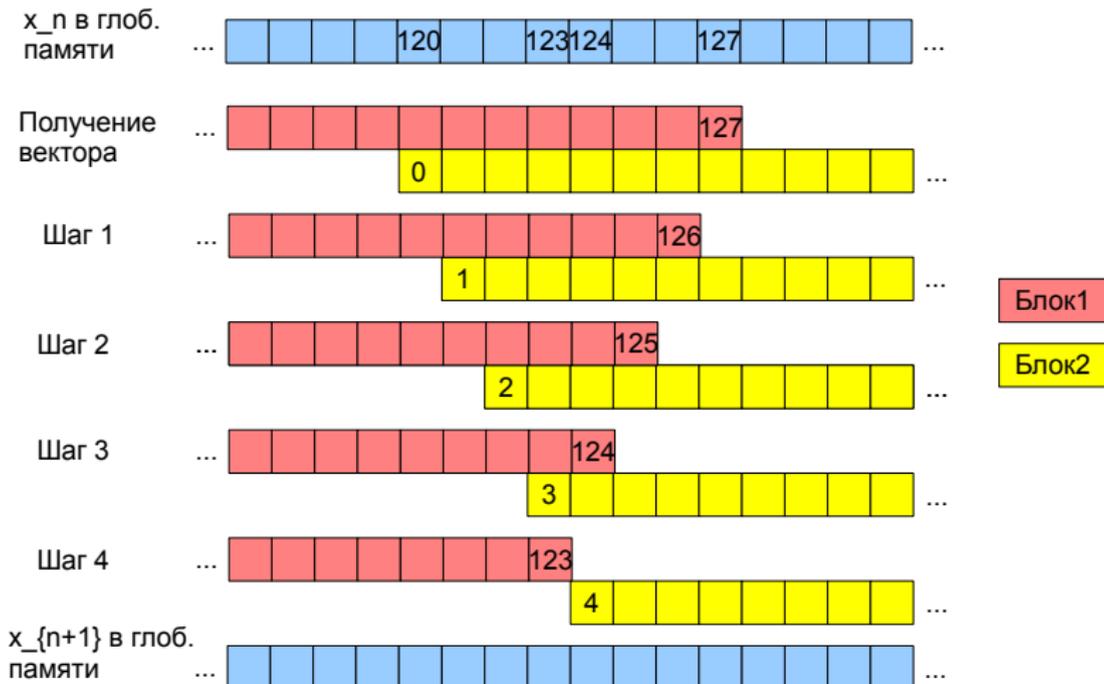
Архитектура GPU

- Массивный параллелизм: сотни вычислительных ядер и десятки тысяч нитей
- SIMD/SIMT-архитектура
- Разбиение нитей на блоки, внутри которых между нитями возможен обмен через быструю общую память с эффективной синхронизацией
- Размер общей памяти крайне ограничен

Разбиение на блоки

- Каждой координате системы сопоставим отдельную нить
- Разбиение на блоки проведем так, чтобы блок мог вычислить все 8 стадий метода, не используя глобальную (медленную) память для хранения промежуточных результатов

Реализация метода



Результаты

- Для сравнения быстродействия использовались процессор Intel I7 950 и видеокарта NVIDIA GTS450
- Тестовая задача: (1)-(2) с параметрами $\lambda = 2$ и $d = 0.01$, дискретизация по $N = 1024$ точкам
- Ускорение версии для видеокарты в сравнении с одноядерной версией для процессора составило 9.6 раз

Результаты

- Для сравнения быстродействия использовались процессор Intel I7 950 и видеокарта NVIDIA GTS450
- Тестовая задача: (1)-(2) с параметрами $\lambda = 2$ и $d = 0.01$, дискретизация по $N = 1024$ точкам
- Ускорение версии для видеокарты в сравнении с одноядерной версией для процессора составило 9.6 раз

Результаты

- Для сравнения быстродействия использовались процессор Intel I7 950 и видеокарта NVIDIA GTS450
- Тестовая задача: (1)-(2) с параметрами $\lambda = 2$ и $d = 0.01$, дискретизация по $N = 1024$ точкам
- Ускорение версии для видеокарты в сравнении с одноядерной версией для процессора составило 9.6 раз