CUDA Assignment #4

Решение краевых задач, системы линейных алгебраических уравнений

Задание

- Основное:
  - Решение 1D/2D задачи диффузии

- Дополнительное:
  - Решение СЛАУ с треугольной разряженной матрицей

Задача диффузии

- Уравнение теплопроводности:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\beta(u,x)\frac{\partial u}{\partial x}\right)$$

- $\beta(u,x)$ - коэффициент теплопроводности
  - Постоянный: $\beta(u,x) = \beta_0$ $\qquad \frac{\partial u}{\partial t} = \beta_0 \frac{\partial^2 u}{\partial x^2}$
  - Линейный: $\beta(u,x) = \beta(x) = x$ $\qquad \frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} + x\frac{\partial^2 u}{\partial x^2}$
  - Нелинейный: $\beta(u,x) = u$ $\qquad \frac{\partial u}{\partial t} = u\frac{\partial^2 u}{\partial x^2}$

Постановка 1D задачи

- Область определения – отрезок $[0, L]$
- Уравнение теплопроводности на $(0, L)$
- Начальные условия:

$$u(x,0) = u_{x0}(x)$$

- Граничные условия:

$$u(0,t) = u_{0t}(t)$$

$$u(L,t) = u_{Lt}(t)$$

Методы решения

- Метод конечных разностей

явные схемы

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\beta_0\left(\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{\Delta x^2}\right)$$

неявные схемы

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\beta_0\left(\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{\Delta x^2}\right)$$

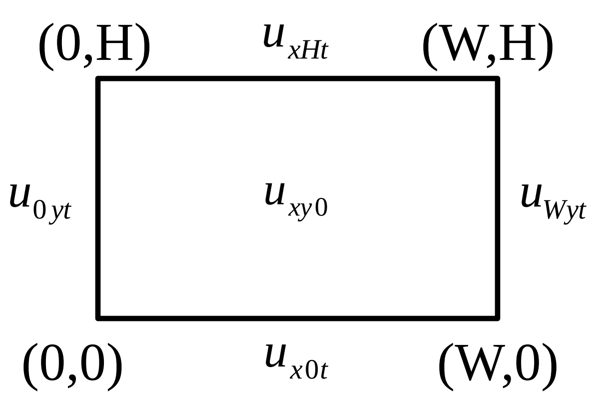
На каждом шаге необходимо решать трехдиагональную СЛАУ

Постановка 2D задачи

- Область определения: $[0,W]\text{x}[0,H]$
- Уравнение теплопроводности: $(0,W)\text{x}(0,H)$

$$\frac{\partial u}{\partial t} = \beta_0\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

- Начальные и граничные условия



$$u(x,y,0) = u_{xy0}(x,y)$$

$$u(0,y,t) = u_{0yt}(y,t)$$

$$u(W,y,t) = u_{Wyt}(y,t)$$

$$u(x,0,t) = u_{x0t}(x,t)$$

$$u(x,H,t) = u_{xHt}(x,t)$$

Метод по-координатного расщепления

- Временной шаг разбивается на два дробных

$$\frac{\partial u}{\partial t} = \beta_0\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial u}{\partial t} = \beta_0\left(\frac{\partial^2 u}{\partial x^2}\right)$$ для **каждого** фикс. y

$$\frac{\partial u}{\partial t} = \beta_0\left(\frac{\partial^2 u}{\partial y^2}\right)$$ для **каждого** фикс. x

Пример разностной аппроксимации

- Расщепление по X:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \beta_0 \left(\frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{\Delta x^2} \right)$$

- Независимые трехдиагональные СЛАУ для каждого j = 0 .. N_y
    - Легко параллелится

Трехдиагональные СЛАУ

- Прямые методы:
  - Метод прогонки
    - Оптимальный по числу операций, но последовательный
  - Метод редукции
    - Параллельный, но больше операций

Итерационные методы

- Якоби, Гаусса-Зейделя
- GMRES, CG
- Преобуславливатель (preconditioner)

Дополнительное задание

- Верхне-треугольная разряженная матрица NxN

x			x		x			x	
		x				x			x
			x			x		x	
			x		x			x	
				x			x		x
						x			
						x	x		x
								x	
								x	x
									x

- Число ненулевых элементов = NNZ

Формат хранения

- Compressed Sparse Row (CSR)
  - `Data[NNZ]` – массив ненулевых элементов
  - `Indices[NNZ]` – индексы столбцов для каждого ненулевого элемента
  - `Ptr[N+1]` – смещение в массиве данных для каждой строки
    - `Ptr[i+1]-Ptr[i]`: число ненулевых эл-тов в i-строке
    - `Ptr[N] = NNZ`: общее число ненулевых элементов

Пример матрицы

7			2	
	6	4		
			1	
			3	
				9

N=5

NNZ=7

Data	7	2	6	4	1	3	9
Indices	0	3	1	2	3	3	4
Ptr	0	2	4	5	6	7	

Задание

- Придумать эффективный алгоритм для реализации на CUDA/GPU

- Матрица задается в формате MatrixMarket (http://math.nist.gov/MatrixMarket/formats.html#MMformat)

Полезные ссылки

- Умножение разряженной матрицы на столбец: http://www.nvidia.com/object/nvidia_research_pub_001.html

Вопросы