GPU Computing with CUDA
Lab 3 - FD with shared memory

Christopher Cooper
Boston University

August, 2011
UTFSM, Valparaíso, Chile

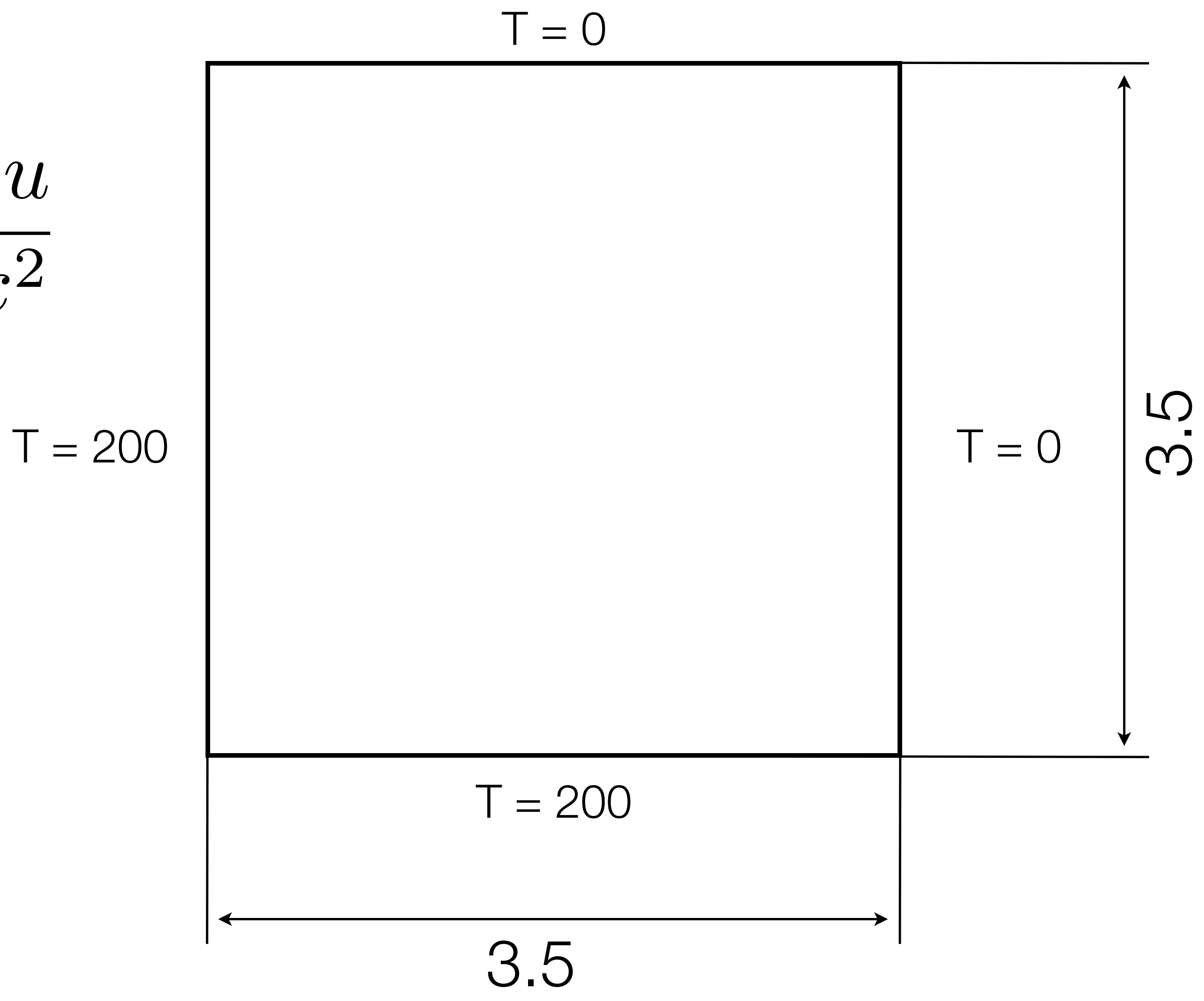
Objectives

- Implement a finite difference code in CUDA using shared memory
- Implement different approaches and measure timings

Heat transfer with FD

- Heat diffusion on square 2D flat plate with Dirichlet boundary conditions

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$



α = 0.645

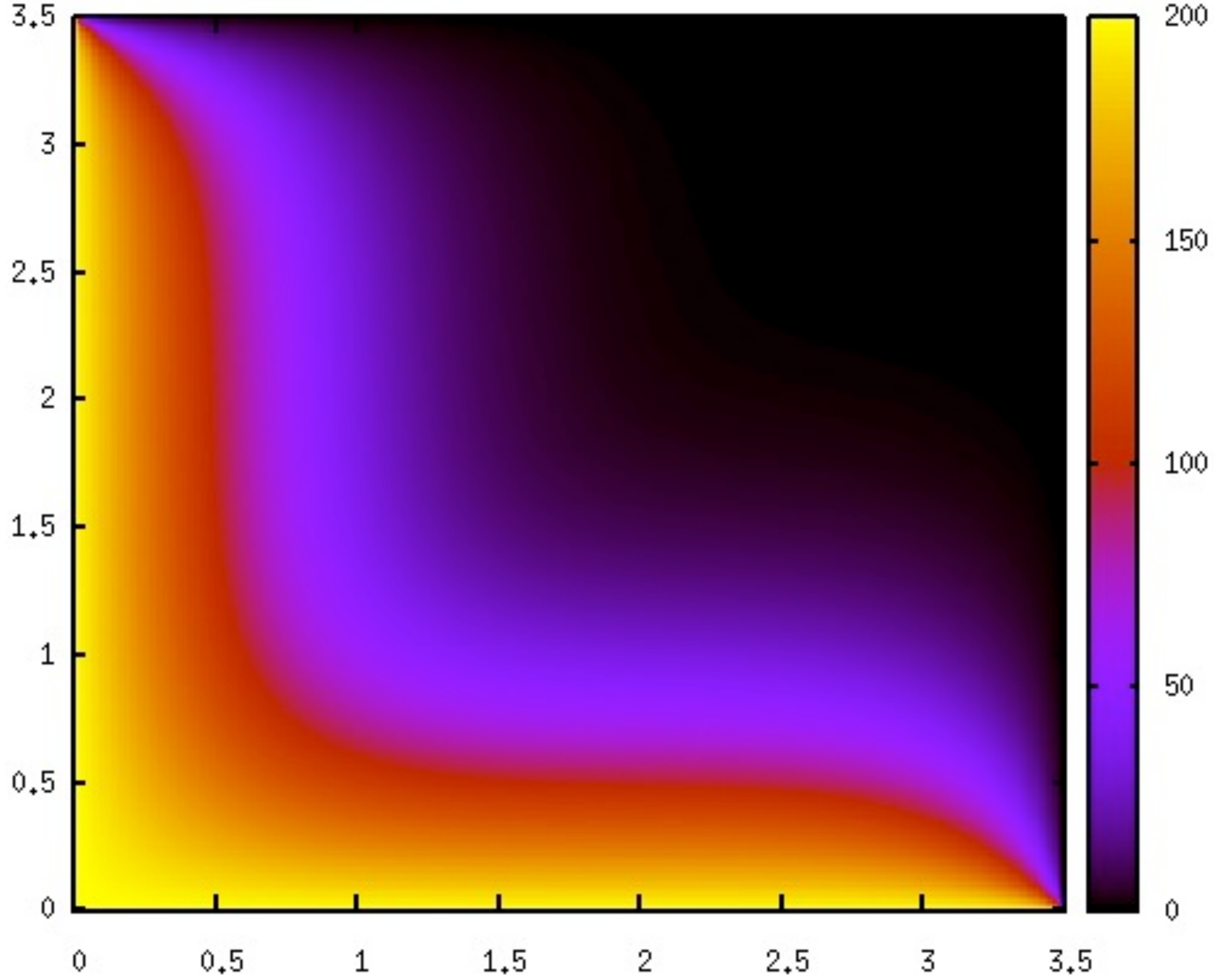
k = 1e-5

N = 128

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \frac{\alpha k}{h^2}(u_{i,j+1}^{n} + u_{i,j-1}^{n} + u_{i+1,j}^{n} + u_{i-1,j}^{n} - 4u_{i,j}^{n})$$

Heat transfer with FD

- Each thread will do one mesh point
- Experiment with other mesh sizes (N) that are not a multiple of block size



Heat transfer with FD

- Implementation 1
    - Load to shared memory

```
__shared__ float u_sh[BSZ][BSZ];
u_prev_sh[i][j] = u[I];
```

    - Compute internal points in shared
    - Compute boundaries of blocks with global memory
    - Problems: branching code and global memory access

Heat transfer with FD

- Implementation 2
  - Use halo nodes
  - All threads load to shared memory
  - Overlapping shared memory subdomains
  - Only internal threads compute
    - As subdomains overlap, we will cover the whole space
  - Problems: underutilization of threads in calculation

Heat transfer with FD

- Implementation 3
  - Make shared arrays bigger ([BSZ+2][BSZ+2])
  - BSZ+2xBSZ+2 subdomains overlap
  - Load data to shared in two passes
  - All threads do the computation
- Look at class 3 notes to refresh your memory!