GPU Computing with CUDA
Lab 7 - Heat Transfer with Crank Nicolson

Christopher Cooper
Boston University

August, 2011
UTFSM, Valparaíso, Chile

Objectives

- Implement an implicit heat transfer solver on the GPU
- Use Cusp and user defined kernels in the same code

wget http://www.bu.edu/pasi/files/2011/07/lab7_FD_2D_implicit.cu_.zip

Heat Transfer with Crank Nicolson

$$[A]\mathbf{u}^{n+1} = \mathbf{RHS}$$

$$[A] = I + \frac{\alpha k}{2h^2} \cdot [\text{Poisson}]$$

```
cusp::blas::axpy(M.values, M_aux, alpha*dt/(2*h*h));
M.values = M_aux;
cusp::add(I, M, A);
```

$$RHS_{i,j} = u^n_{i,j} + \frac{\alpha k}{2h^2}\left(u^n_{i,j-1} + u^n_{i,j+1} + u^n_{i-1,j} + u^n_{i+1,j} - 4u^n_{i,j}\right) - BC^{n+1}$$

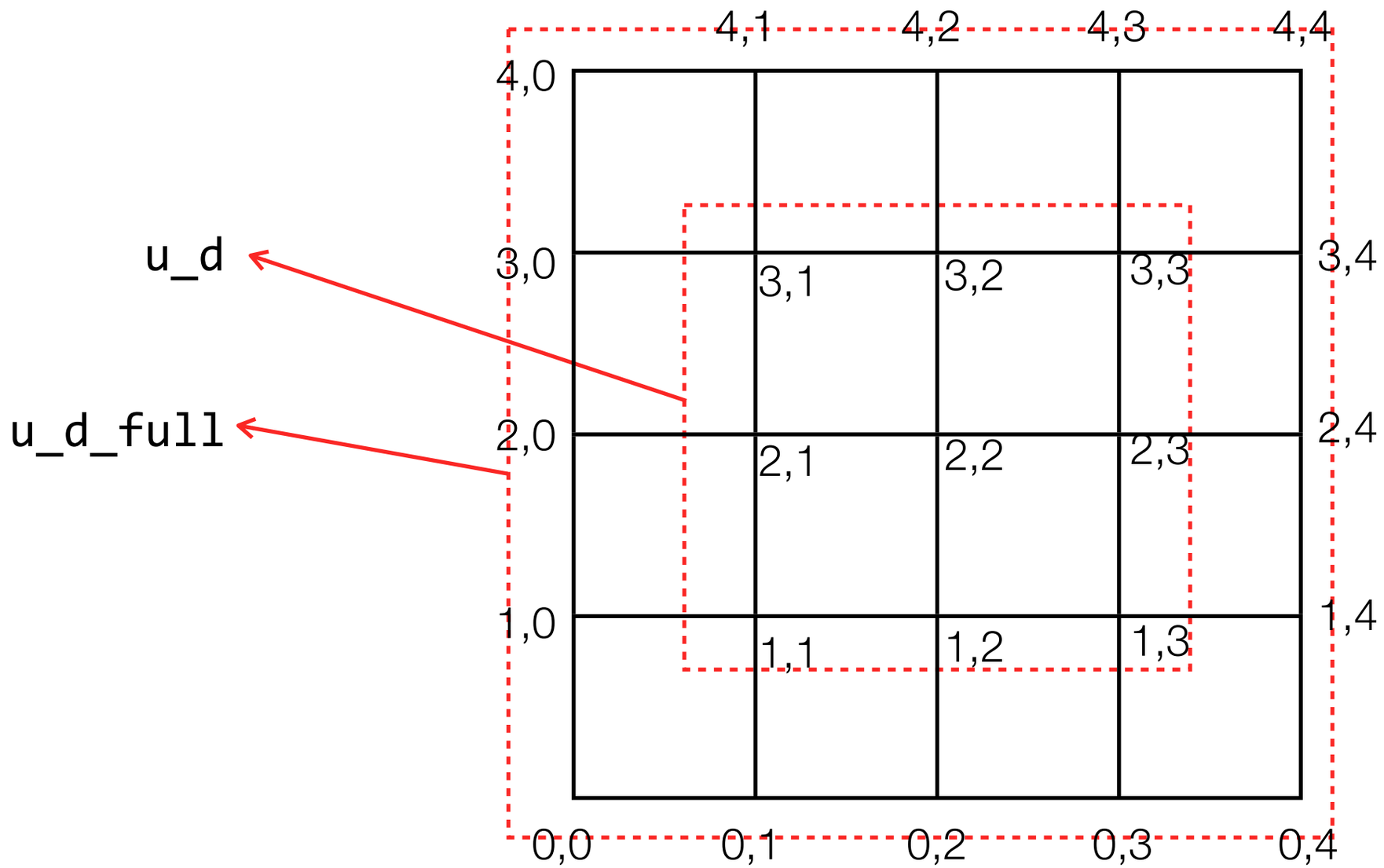
Stencil code from lab 2/3

$[A]$ size $(\text{N-2})^2$ x $(\text{N-2})^2$

RHS size $(\text{N-2})^2$

$\mathbf{u}^{n+1}$ size $(\text{N-2})^2$

Heat Transfer with Crank Nicolson



u_d ⇨ u_d_full

```
I_large = N*BSZ*by + N*(j+1) + BSZ*bx + i+1
I_small = (N-2)*BSZ*by + (N-2)*j + BSZ*bx + i
```