Heat Equation

Francesco Pasa
Enrico Panontin

Heat Equation

Our Code

Performances

OpenGL-CUDA
integration

Heat Equation

Francesco Pasa
Enrico Panontin

Technische Universität München
Physics Department
Parallelisation of Physics Calculations on GPUs with CUDA

16 Juni 2016

Heat Equation

Francesco Pasa
Enrico Panontin

Heat Equation
Our Code
Performances
OpenGL-CUDA
integration

Heat Equation

Our Code

Performances

OpenGL-CUDA integration

Heat Equation

Francesco Pasa
Enrico Panontin

Heat Equation

Our Code

Performances

OpenGL-CUDA
integration

Demonstration

We consider heat propagation in a medium (e.g. a solid) far away from any state transition and we define:

ϵ : internal energy density

c : specific heat

ρ : mass density

T : temperature

By virtue of the first thermodinamic principle
$(\Delta U = \Delta Q - L = \Delta Q)$:

$$d\left[\int_\Omega c\rho T \, \mathrm{d}V\right] = \left[-\int_{\partial\Omega} \vec{J} \cdot \vec{n} \, \mathrm{d}S\right] \mathrm{d}t \qquad (1)$$

Demonstration

Heat Equation

Francesco Pasa
Enrico Panontin

Heat Equation

Our Code

Performances

OpenGL-CUDA
integration

$$d\left[\int_\Omega c\rho T \,\mathrm{d}V\right] = \left[-\int_{\partial\Omega} \vec{J} \cdot \vec{n} \,\mathrm{d}S\right] \mathrm{d}t$$

Newton-Fourier Law, heat spreads and temperature becomes uniform throughout the medium:

$$\vec{J} = -k\vec{\nabla} T \qquad (2)$$

By substituting equation (2) in (1) and applying the *divergence theorem* one gets the **heat equation**:

$$\frac{\partial T}{\partial t} - \frac{k}{c\rho}\Delta T = 0 \qquad (3)$$

Heat Equation

Heat Equation

Francesco Pasa
Enrico Panontin

Heat Equation

Our Code

Performances

OpenGL-CUDA
integration

$$\frac{\partial T}{\partial t} - \frac{k}{c\rho}\Delta T = f(\vec{x}, t)$$

- First order in time
- Second order in space
- If we consider a heating system, we must add $f(\vec{x}, t)$

Numerical Solution

Heat Equation

Francesco Pasa
Enrico Panontin

Heat Equation

Our Code

Performances

OpenGL-CUDA
integration

$$\frac{\partial T}{\partial t} - \frac{k}{c\rho}\Delta T = f(\vec{x}, t)$$

Discretize space and time: foreward difference for time,
central difference for space:

$$T_{i,j}^{n+1} = T_{i,j}^n + \eta \left[T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n - 4T_{i,j}^n \right] \tag{4}$$

where n is the time index, i, j are x-index and y-index,
$\eta = \frac{k\Delta t}{c\rho(\Delta x)^2}$.

General Structure

Francesco Pasa
Enrico Panontin

kjh

Shared Memory

Random errors
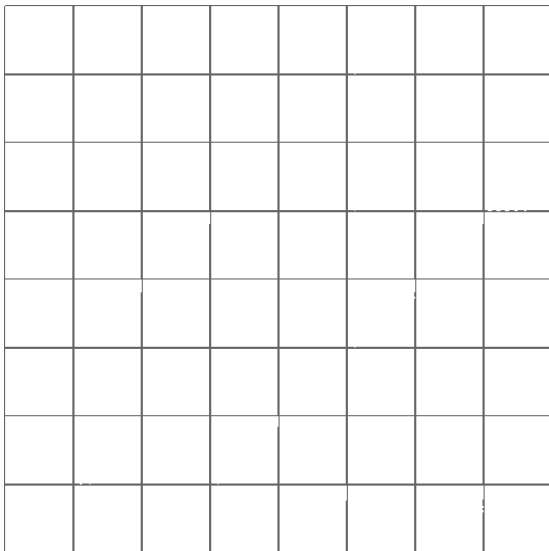
Heat Equation

Francesco Pasa
Enrico Panontin

Heat Equation

Our Code

Performances

OpenGL-CUDA
integration

Shared Memory

Synchronized threads
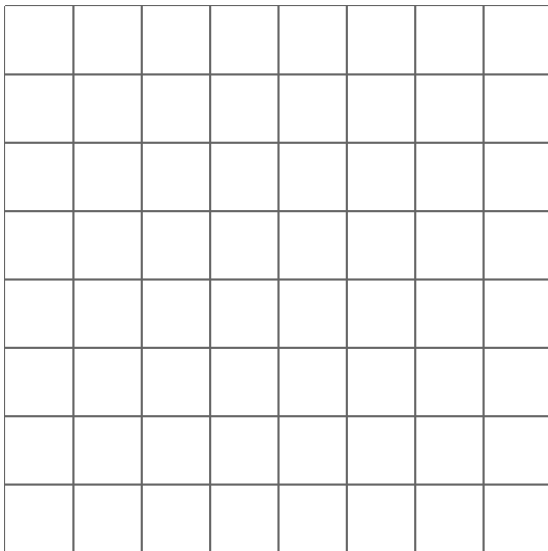
Performances
GPU

descrizione della gpu usata (numero blocchi e thread eccecc)

Performances

Execution time vs blocks number

Performances

Execution time vs loops number

Performances

Execution time GPU vs CPU

Performances
Different GPUs

Francesco Pasa
Enrico Panontin

Introduction