# Heat equation in CUDA

Eric Goubault

March 26, 2023

## 1 Introduction

The objective of this project is to compute (approximate) solutions of the heat equation on the 2D plane, which is expressed by the following partial differential equation:

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}$$

where $U$ is the heat function defined at a given time $t$, and at coordinates $x$ and $y$ in the 2D plane.

In order to find approximate solutions of this heat equation, we are going to discretize the value of $U$ at coordinates $x_i = i\Delta$ and $y_j = j\Delta$, and at times $t_n = n\Gamma$ for some time and space steps $\Gamma$ and $\Delta$.

Noting $U^n_{i,j} = U(x_i, y_j, t_n)$, and discretizing the partial derivatives at first order, we can write:

$$\frac{\partial U}{\partial t}^n_{i,j} \sim \frac{U^{n+1}_{i,j} - U^n_{i,j}}{\Gamma}$$

and:

$$\frac{\partial^2 U}{\partial x^2}^n_{i,j} + \frac{\partial^2 U}{\partial y^2}^n_{i,j} \sim \frac{U^n_{i+1,j} + U^n_{i,j+1} + U^n_{i-1,j} + U^n_{i,j-1} - 4U^n_{i,j}}{\Delta^2}$$

Finally, we get the *explicit scheme*:

$$U^{n+1}_{i,j} = (1 - 4\lambda)U^n_{i,j} + \lambda(U^n_{i+1,j} + U^n_{i,j+1} + U^n_{i-1,j} + U^n_{i,j-1})$$

where $\lambda = \frac{\Gamma}{\Delta^2}$, that we suppose is positive, less than 0.5 (to ensure stability of the numerical scheme)

## 2 Steps of the project

We will suppose we are propagating the heat using the numerical scheme above, from time $t = 0$ where the heat is initialized, for instance, as a start, with one heat point, and seval ones, or several "hot" regions for later tests.

- The first step is to program a sequential version of the numerical scheme, in C, with a visualization of the solutions found (you are free to use any method for graphical visualisation, a simple one would be to use gnuplot, through files written in C)
- The second step is to test the program and make sure that the solutions found are indeed valid solutions (you can have a look at e.g. the "heat equation" wikipedia page)
- The third step is to parallelize the numerical scheme, so that it can fit in a GPU card
- The fourth step is to test this implementation and measure its performances, according to different choices of parameters, with respect to the sequential version

An optional step is to consider, after this, the implicit numerical scheme which is obtained by writing:

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n}}{\Gamma} = \frac{U_{i+1,j}^{n+1} + U_{i,j+1}^{n+1} + U_{i-1,j}^{n+1} + U_{i,j-1}^{n+1} - 4U_{i,j}^{n+1}}{\Delta^2}$$

and solve this system of equation to find $U^{n+1}$ from $U^n$ (using for instance a Gauss-Seidl iterative method). For this optional step, you should describe this numerical scheme and how to parallelize it on a GPU. Ideally, also, write and test the corresponding CUDA program.