HIGH PERFORMANCE COMPUTING

PROJECT STAGE 2

Name: Dylan Scotney
Student number: 18138211
Submission date: 29/04/2019

1. INTRODUCTION

Following on from Stage 1 of the project, we will now consider the parabolic equation,

$$u_t = \nabla \cdot (\sigma(x,y)\nabla)u \tag{1.01}$$

over the same spatial domain and boundary conditions as Stage 1 such that, $x \in (0,1)$ and $y \in (0,1)$ and $u(x,y,t) = 0$ on the boundaries for all $t \geq 0$. $\sigma(x,y)$ is the same random field used in Stage 1.

Stage 2 will develop an OpenCL implementation to solve this parabolic PDE using a forward difference scheme for the time discretisation, i.e.,

$$u_t(x,y) \approx \frac{u(x,y,t+\Delta t) - u(x,y,t)}{\Delta t} \tag{1.02}$$

The behaviour of the solver will be studied for several different initial conditions and time steps, Δt.

From Equation 1.02 it follows that an approximation the value of $u(x,y)$ at any given time can be calculated using,

$$u_{k,m+1} = u_{k,m} + \Delta t\big(Au_{k,m}\big) \tag{1.03}$$

Where, as explained in the Stage 1 report, k is the super index representing the spatial position of a grid point and A is the discrete version of the RHS operator in Equation 1.01. We have introduced index m to represent the discrete time such that $t = m\Delta t$.

2. IMPLEMENTATION IN OPENCL

We already know from stage one that the computation of $Au_{k,m}$ can be implemented in parallel and in order to do this we will be using a very similar OpenCL kernel as before. However, since work items in OpenCL have *no execution order guarantee*, it is not possible to compute all $u_{k,m}$ in parallel since each timestep is dependent on the previous. A simple algorithm for efficiently calculating the solution is as follows:

```
Set $m = 0$ and define initial condition $u_{m=0}$,
**While** ($m\Delta t < T$)
Pass $u_{k,m}$ to the OpenCl kernel and calculate $u_{k,m+1}$ $\forall k$ via Equation 1.03 in parallel
$m = m + 1$
**End while.**
```

Where T is the terminal time. Refer to the Jupyter notebook, HPCproject-Stage2.ipynb for more information.

3. BASIC SOLUTION FOR SINGLE POINT INITIAL CONDITION

A solution the the system was first found using an intial conditon of,

$$u(x, y, 0) = \begin{cases} 1, & x = 0.5, \quad y = 0.5 \\ 0, & elsewhere \end{cases}$$

Where $\sigma(x, y) = 1, \forall\, x, y$.

As t increased the solution diffused into a similar shape to that of the stage one solution. Once in this state the general shape of the solution remains the same as $t \to \infty$ but the amplitude tends to zero. See Figure 3.1. We will refer to the shape of the solution for large t as the relaxed state.

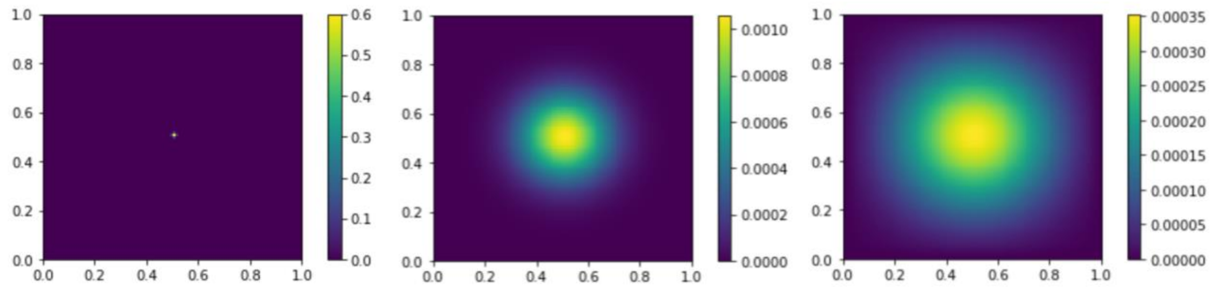

Figure 3.1. The solution at $t = 0, 0.15, 0.3$ using $\Delta t = 10^{-5}$ and $\sigma(x, y) = 1$.

4. IMPACT OF Δt

It is known that the forward-euler method is not unconditionally stable. Infact, for the case where $\sigma(x, y) = 1$ and $\Delta x = \Delta y$, it's easy to prove that the stability, Δt must satisfy[1]:

$$\Delta t < \frac{\Delta x^2}{4} \tag{4.01}$$

So intuitavely, a sufficiently small Δt is required for the solution to converge towards the relaxed state. Figure 4.1 illustrates the solution blowing up for $\Delta t > \frac{\Delta x^2}{4}$. Notice the value of the axis after just two timesteps.

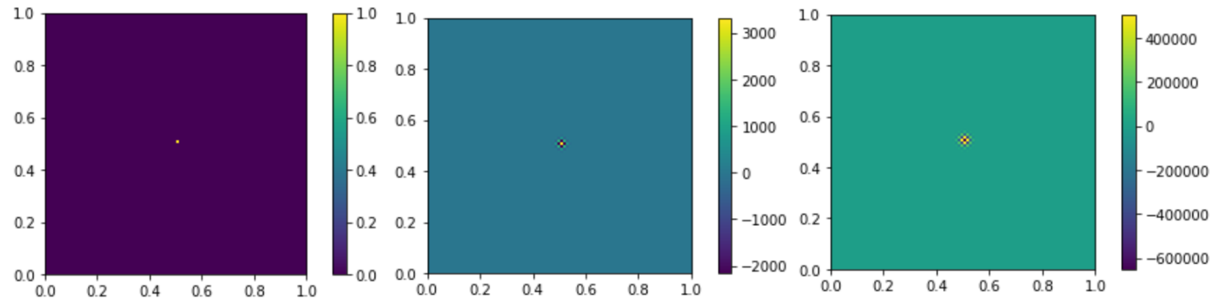

Figure 4.1. The solution at $t = 0, 10^{-4}, 2 \times 10^{-4}$ using $\Delta t = 10^{-4}$ and $\sigma(x, y) = 1$.

5. IMPACT OF VARIANCE IN σ(x,y)

Section 3 and 4 give results for a constant $\sigma(x,y) = 1$. This section will briefly discuss the impact of setting σ to be a random field such that $\sigma(x,y) = \exp\big(\mathbb{S}(x,y)\big)$ where $\mathbb{S}(x,y) \sim N(1,\sigma_s)$ and investigate behaviour of the solution depending on the size of σ_s.

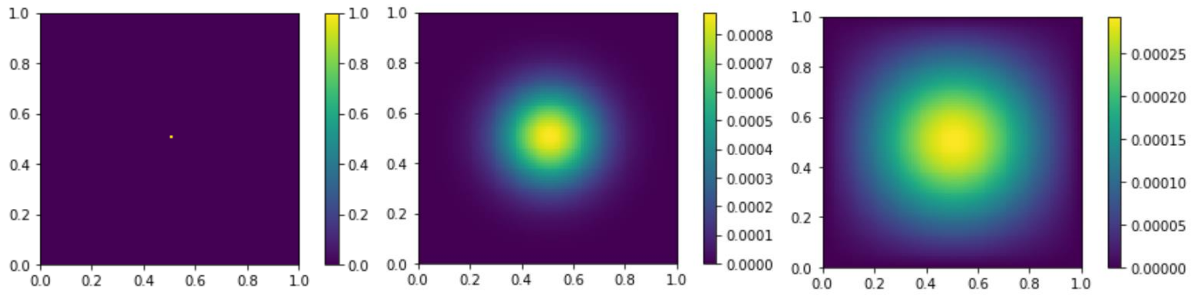

Figure 5.1. The solution at $t = 0, 0.0025, 0.005$ using $\Delta t = 10^{-6}$ and $\mathbb{S}(x,y) \sim N(1,0.01)$.

The introduced random field increases the instability and slows convergence of the system so that a smaller Δt and less iterations are required for the system to approach the relax state. Comparing Figure 5.1 to 3.1 we see that the system with the random field converges to the relaxed state in at least ~ 5000 iterations/timesteps where as for $\sigma(x,y) = 1$ the system required more like $\sim 30{,}000$.

6. IMPACT OF INITIAL CONDITIONS

For sufficiently small Δt, the system will mostly converge to the relaxed state regardless of the initial condition. Since the OpenCL framework forces the boundary conditions (see accompanying code), the initial condition does not even need to naturall satisfy the boundary conditions. Intuitatively, if the initial condition has negative values rather than positive, the system will converge to a negative form of the relaxed state. Figures 6.1 – 6.4 exhibit this behaviour.

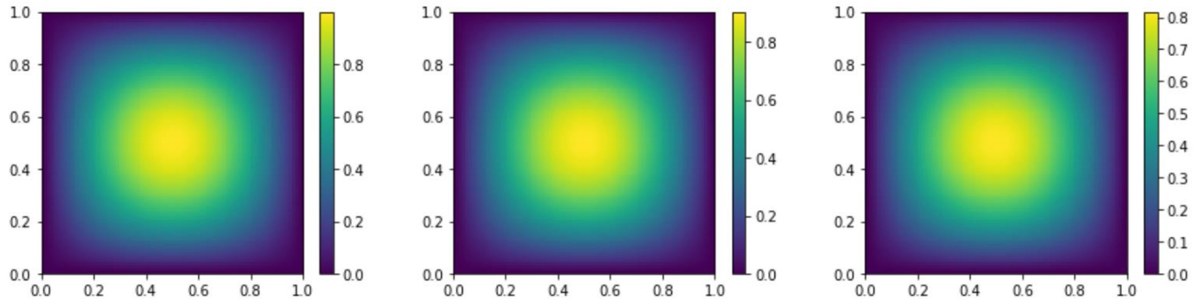

Figure 6.1. The progression of the solution initial condition $u(x,y,0) = \sin(\pi x)\sin(\pi y)$.

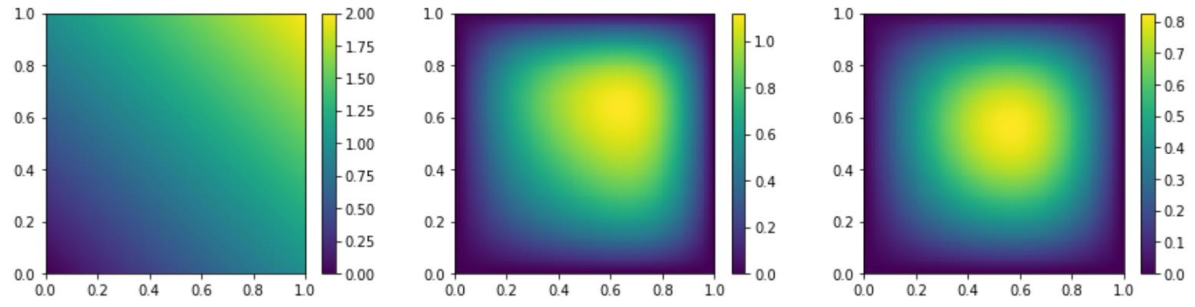

Figure 6.2. The progression of the solution initial condition $u(x,y,0) = x + y$.

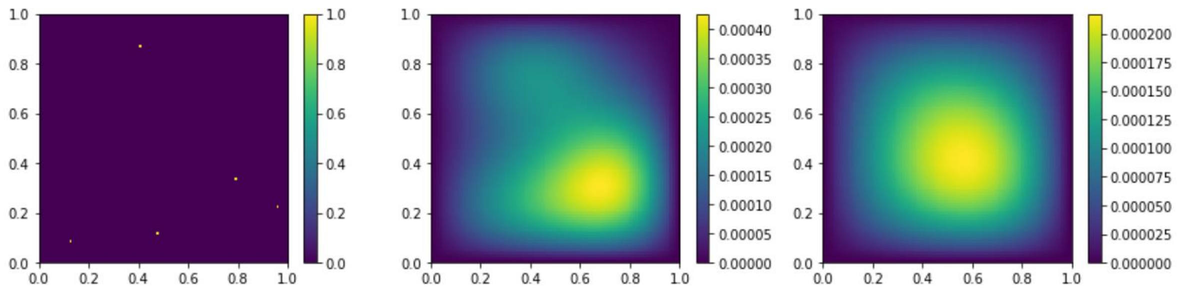
Figure 6.3. The progression of the solution initial condition $u(x, y, 0) = 1$ at 5 random points.

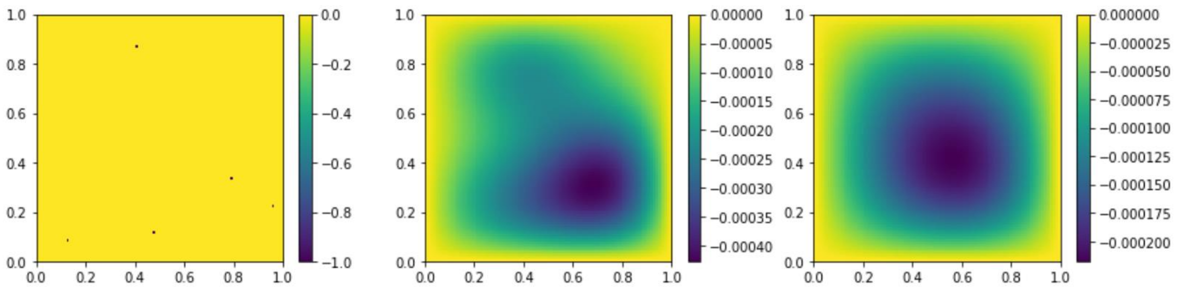
Figure 6.4. The progression of the solution initial condition $u(x, y, 0) = -1$ at 5 random points.

Figures 6.1 – 6.4 consider only intial conditions where u values are not of opposite sign. Some interesting behaviour can be shown if we choose intial conditions which change sign across the xy space. See Figure 6.5. It describes a situation of random points of opposing values diffusing, the diffusion of the system almost appears to momentarily reach a steady symmetrical state until a side with greater weight eventually overcomes the other and the system diffuses back to the previous steady states of Figures 6.1 – 6.4.

The behaviour of Figure 6.5 prompts the question of what happens if the initial condition changes sign, but is symmetrical across the considered space. Figures 6.6 and 6.7 explore this. It turns out that if an intial input is symmetrical, the system cannot break symmetry as it's ammplitude slowly decays

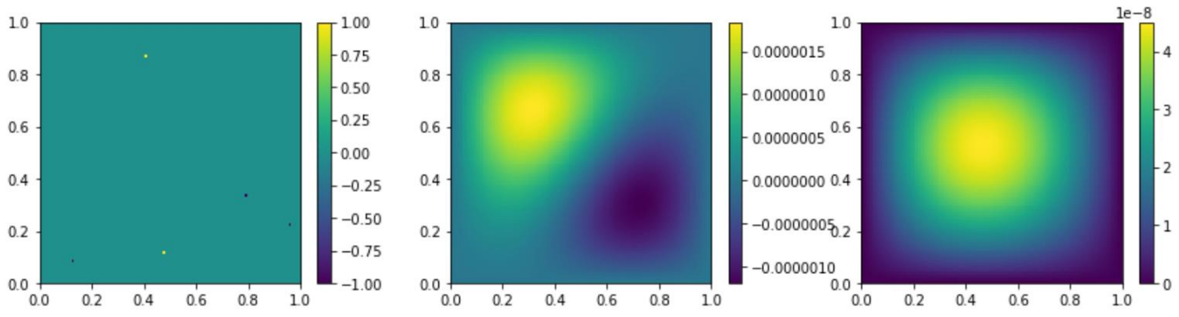
Figure 6.5. The progression of the solution initial condition $u(x, y, 0) = \pm1$ at 5 random points.

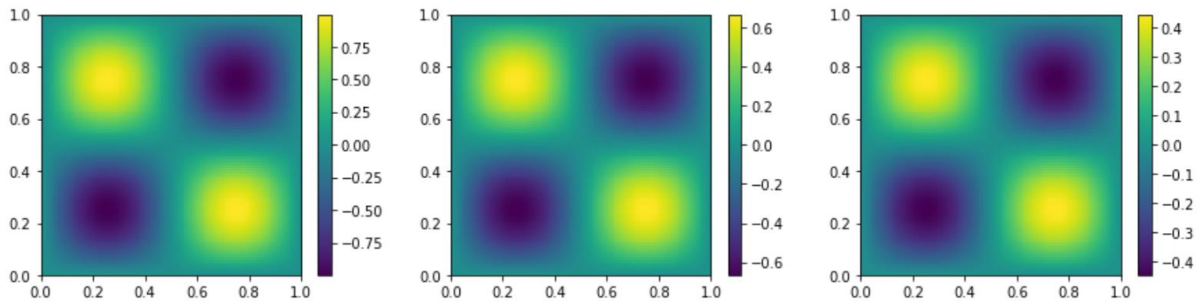
Figure 6.6. The progression of the solution initial condition $u(x, y, 0) = \sin(2\pi x) \sin(2\pi y)$.

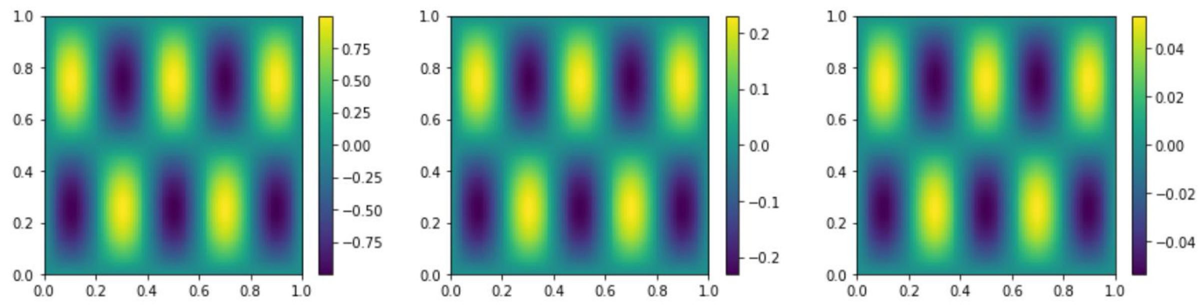

Figure 6.7. The progression of the solution initial condition $u(x, y, 0) = \sin(5\pi x)\sin(2\pi y)$.

to zero. One can easily convice themselves why this behaviour occurs by simply considering the nature in which each timestep updates. Of course, if the system was simulated for a large amount of timesteps, it's reasonable to expect the system will eventually break symmetry (and hence converge to a single peak state like initially explored). This is not natural behaviour but rather a consequence of the computers rounding error as the magnitude of the system approaches machine epsilon.

[1] Smith. G. D. 2019. *Numerical solution to differential equations: finite difference method*. 3rd ed. Oxford: Clarendon 1985. Print. Oxford Applied Mathematics and Computing Science Ser.