Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

**High-Performance Computing Lab for CSE** **2024**

Student: CARLA JUDITH LOPEZ ZURITA Discussed with: FULL NAME

## Solution for Project 3

Due date: Monday 15 April 2024, 23:59 (midnight)

**HPC Lab for CSE 2024 — Submission Instructions**
**(Please, notice that following instructions are mandatory: submissions that don't comply with, won't be considered)**

- Assignments must be submitted to Moodle (i.e. in electronic format).
- Provide both executable package and sources (e.g. C/C++ files, Matlab). If you are using libraries, please add them in the file. Sources must be organized in directories called:
  *Project_number_lastname_firstname*
  and the file must be called:
  *project_number_lastname_firstname.zip*
  *project_number_lastname_firstname.pdf*
- The TAs will grade your project by reviewing your project write-up, and looking at the implementation you attempted, and benchmarking your code's performance.
- You are allowed to discuss all questions with anyone you like; however: (i) your submission must list anyone you discussed problems with and (ii) you must write up your submission independently.

In this report, we are solving Fisher's equation using the finite difference method. The equation is given by:

$$\frac{\partial s}{\partial t} = D\nabla^2 u + Rs(1-s) \tag{1}$$

where $s$ is the concentration of a species, $D$ is the diffusion coefficient, and $R$ is the reaction rate. The equation is solved on a 2D grid with Dirichlet boundary conditions,

$$s(x, y, t) = 0.1 \quad \text{for} \quad x = 0, x = 1, y = 0, y = 1 \tag{2}$$

and initial conditions is a circle of radius 1/8 at the lower left quadrant the domain.

# 1. Task: Implementing the linear algebra functions and the stencil operators [40 Points]

The first part of the project is to implement the linear algebra functions and the stencil operators. The linear algebra functions are implemented in the `linalg.cpp` file. As first try, I implementd all the functions using simple loops. I tried using iterators but since the Fields class is a coustom class, I decided to put that idea on hold until I have a better understanding of the code. On the other hand, the stencil operators are implemented in the `operators.cpp`. Writing the interior grid points was very simple since we already had the implementations for the boundary conditions.

Afterwards, I ran the code following the indications on the project description and I got the following results in the terminal

```
================================================================================
                    Welcome to mini-stencil!
version :: C++ Serial
mesh :: 128 * 128 dx = 0.00787402
time :: 100 time steps from 0 .. 0.005
iteration :: CG 300, Newton 50, tolerance 1e-06
================================================================================
--------------------------------------------------------------------------------
simulation took 0.203392 seconds
1513 conjugate gradient iterations, at rate of 7438.82 iters/second
300 newton iterations
--------------------------------------------------------------------------------
### 1, 128, 100, 1513, 300, 0.203392 ###
Goodbye!
```

Which are the expected results, even if a bit slower than the results included in the original document, but this is just one run and could be attributed to the randomness of run-time of the processes. Additionally, I got the following content for the `output.bin` file,

```
TIME: 0.005
DATA_FILE: output.bin
DATA_SIZE: 128 128 1
DATA_FORMAT: DOUBLE
VARIABLE: phi
DATA_ENDIAN: LITTLE
CENTERING: nodal
BRICK_ORIGIN: 0. 0. 0.
BRICK_SIZE: 1 1 1.0
```

which produced Fig. 1.

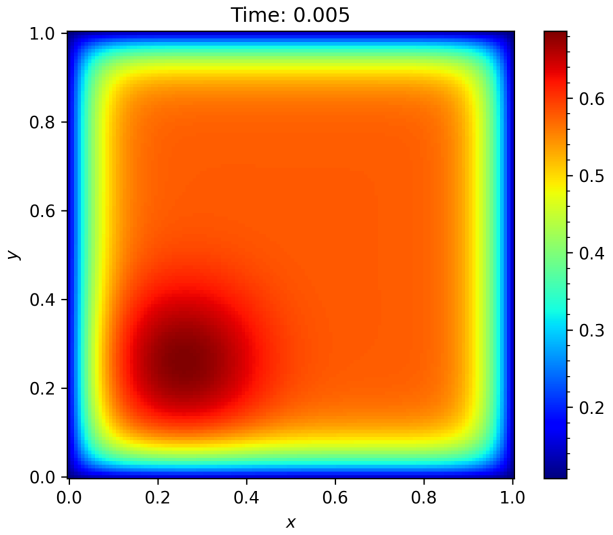


Figure 1: The population concentration at time t = 0.005.

## 2. Task: Adding OpenMP to the nonlinear PDE mini-app [60 Points]

This section is dedicated to parallelizing the code using OpenMP. The first step was to parallelize the linear algebra functions. I used the `omp parallel for` directive to parallelize the loops. The stencil operators were