Natural Language Processing, Project Report

UID: 1690550

March 14, 2017

1 Initial readings, before any changes were made

After a clean make and running the code for the first time, we get the following results from gprof:

function	1
poisson	23.14
compute_tentative_velocity	1.10
compute_rhs	0.12
apply_boundary_conditions	0.02
set_timestep_interval	0.02
update_velocity	0.02

We can see that the poisson function is by far the most expensive one, and so we shall focus on it first.

2 Domain Decomposition

We have the option to decompose the region in 1 or 2 directions. After some trial and error, we find that attempting to split the region both horizontally and vertically is not very helpful. It creates a lot of problems for communication between processes later, and generally we do not get much improvement this way.

Instead, we split the region into vertical chunks, one for each process, as shown in Fig. 1. It is straightforward to calculate the number of chunks necessary. We can find the number of processes using:

```
int nprocs = 0
MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
```

3 Using OpenMP

First we observe that in the `simulation.c` there are two for loops iterating over the region: one is calculating the u values, the other is calculating the values of v.