Wind Tunnel Simulation

EduHPC'21 Peachy Assignment

1 Introduction

You are provided with a sequential code that simulates in a simplified way, how the air pressure spreads in a Wind Tunnel, with fixed obstacles and flying particles. The simulation represents a zenithal view of the tunnel, by a 2-dimensional cut of the tunnel along the main air-flow axis. The space is modelled as a set of discrete and evenly spaced points or positions. A bidimensional array is used to represent the value of air pressure at each point in a given time step. Another unidimensiona array stores the information of a collection of flying particles (they can be pushed and moved by the air flow), or fixed structures that cannot move. All particles have a mass and a resistance to the flow air, partially blocking and deviating it.

Fan inlet There is an air inlet at the start of the tunnel with a big fan. The values of air pressure introduced by the fan are represented in the row 0 of the array. After a given number of simulation steps the fan turning-phase is slightly changed recomputing the values of air pressure introduced, also adding some random noise. The fan phase is cyclically repeated.

Every time step, the values of air pressure in the array are updated with the old values in upper rows, propagating the air flow to the lower rows. On each iteration only some rows are updated. They represent wave-fronts, separated by a given number of rows.

Fixed and flying particles The information of a particle includes: Position coordinates, a two dimensional velocity vector, mass, and air flow resistance. All of them are numbers with decimals represented in fixed position. The particles let some of the air flow to advance depending on their resistance. The air flow that is blocked by the particle is deviated creating more pressure in the front of the particle. The flying particles are also pushed by the air when the pressure in their front is enough to move their mass. The updates of deviated pressure and flying particles movements are done after a given number of time steps (the same as the distance between propagation wave fronts).

End of simulation The simulation ends after a fixed number of iterations (input argument) or if the maximum variation of air pressure in any cell of the array is less than a given threshold. At the end, the program outputs a results line with the number of iterations done, the value of the maximum variability of an update in the last iteration, and the air pressure value of several chosen array positions in three rows: Near the inlet, in the middle of the tunnel, in the last row of the tunnel; or in the last row updated by the first wave front in case that the simulation stops before the air flow arrives at the last row.

Example Figure 1 shows the output of the program with an array of 39x43 cells, after 3 710 iterations. Fixed obstacles are added to represent the shape of a plane in the middle of the wind tunnel. Some flying particles have been moved by the air flow to the lateral boundaries, some others to the last row, and some others are floating behind the plane wings.

```
$ ./wind_seq 39 43 3710 0.8 2 33 1 0 0 5 3 0.3 3434 1242 965
8 17 0.9 9 16 0.9 9 17 0.9 9 18 0.9 10 16 0.9 10 17 0.9 10 18 0.9 11 16 0.9 11 17 0.9 11 18 0.9 12 16 0.9 12 17 0.9 12 18 0.9 13 16 0.9
13 17 0.9 13 18 0.9 14 7 0.9 14 8 0.9 14 9 0.9 14 10 0.9 14 11 0.9 14 12 0.9 14 13 0.9 14 14 0.9 14 15 0.9 14 16 0.9 14 17 0.9 14 18 0.9
14 19 0.9 14 20 0.9 14 21 0.9 14 22 0.9 14 23 0.9 14 24 0.9 14 25 0.9 14 26 0.9 14 27 0.9 15 6 0.9 15 7 0.9 15 8 0.9 15 9 0.9 15 10 0.9
15 11 0.9 15 12 0.9 15 13 0.9 15 14 0.9 15 15 0.9 15 16 0.9 15 17 0.9 15 18 0.9 15 19 0.9 15 20 0.9 15 21 0.9 15 22 0.9 15 23 0.9
15 24 0.9 15 25 0.9 15 26 0.9 15 27 0.9 15 28 0.9 16 5 0.9 16 6 0.9 16 7 0.9 16 8 0.9 16 9 0.9 16 10 0.9 16 11 0.9 16 12 0.9 16 13 0.9
16 14 0.9 16 15 0.9 16 16 0.9 16 17 0.9 16 18 0.9 16 19 0.9 16 20 0.9 16 21 0.9 16 22 0.9 16 23 0.9 16 24 0.9 16 25 0.9 16 26 0.9
16 27 0.9 16 28 0.9 16 29 0.9 17 5 0.9 17 6 0.9 17 7 0.9 17 8 0.9 17 9 0.9 17 10 0.9 17 11 0.9 17 12 0.9 17 13 0.9 17 14 0.9 17 15 0.9
17 16 0.9 17 17 0.9 17 18 0.9 17 19 0.9 17 20 0.9 17 21 0.9 17 22 0.9 17 23 0.9 17 24 0.9 17 25 0.9 17 26 0.9 17 27 0.9 17 28 0.9
17 29 0.9 18 6 0.9 18 7 0.9 18 8 0.9 18 9 0.9 18 10 0.9 18 11 0.9 18 12 0.9 18 13 0.9 18 14 0.9 18 15 0.9 18 16 0.9 18 17 0.9 18 18 0.9
18 19 0.9 18 20 0.9 18 21 0.9 18 22 0.9 18 23 0.9 18 24 0.9 18 25 0.9 18 26 0.9 18 27 0.9 18 28 0.9 19 16 0.9 19 17 0.9 19 18 0.9
20 16 0.9 20 17 0.9 20 18 0.9 21 16 0.9 21 17 0.9 21 18 0.9 22 16 0.9 22 17 0.9 22 18 0.9 23 16 0.9 23 17 0.9 23 18 0.9 24 16 0.9
24 17 0.9 24 18 0.9 25 16 0.9 25 17 0.9 25 18 0.9 26 16 0.9 26 17 0.9 26 18 0.9 27 16 0.9 27 17 0.9 27 18 0.9 28 17 0.9 29 17 0.9
30 17 0.9 31 16 0.9 31 17 0.9 31 18 0.9 32 15 0.9 32 16 0.9 32 17 0.9 32 18 0.9 32 19 0.9 33 15 0.9 33 16 0.9 33 17 0.9 33 18 0.9
33 19 0.9 34 17 0.9

Arguments, Rows: 39, Columns: 43, max_iter: 3710, threshold: 0.800000
Arguments, Inlet: 2, 33  Band of fixed particles: 0, 0, 0.000000  Band of moving particles: 5, 3, 0.300000
Arguments, Init Random Sequence: 58754,61152,57291
Particles: 215

Iteration: 3710, max_var: 10.597100
  +-----------------------------------------------------------------------------------------------------------------------------------------+
  |       7  7  7  8  7  8  8  8  7  8  8  8  8  8  8  8  8  8  9  9  9  9  9  *  9  9  *  9  *  *  *  *                                    |
  |    1  5  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  8  9  9  8  8  9  9  9  9  9  9  9  *  7  2                              |
  | +  2  5  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9  6  3  +                              |
  | 1  2  4  6  7  7  7  7  7  7  7  7  8  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  9  9  9  8  6  3  1  .              |
  | 1  2  4  6  7  7  7  7  7  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  7  5  3  1  .  .           |
  | 2  3  4  6  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  8  8  7  5  3  1  .  .  . [.]         |
->| 2  3  4  6  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  7  6  5  3  1  +  .  . [.][.][.]   |
  | 3  3  5  6  7  8  8  8  8  8  8  8  8  8  8  8  *  *  *  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  6  4  3  1  +  .  .  . [.] . |
  | 3  4  5  6  7  8  8  8  8  8  8  8  8  8  8  *  * [*] *  *  7  7  7  7  7  7  7  7  8  8  7  7  7  6  4  3  1  +  .  .  .  . [.] . |
  | 3  4  5  6  7  8  8  8  8  8  8  8  8  8  9  * [8][*][8] *  9  8  8  8  8  8  8  8  8  8  8  7  7  6  4  3  1  +  .  .  .  .  .  . |
  | 3  4  5  6  7  8  8  8  8  8  8  8  8  8  *  * [5][7][5] *  *  8  8  8  8  8  8  8  8  8  8  8  7  7  6  4  3  2  1  .  .  .  .  .  . |
  | 4  4  5  6  7  8  8  8  8  8  8  8  8  9  *  * [4][6][4] *  *  9  8  8  8  8  8  8  8  8  8  8  7  7  6  4  3  2  1  +  .  .  .  .  . |
  | 4  4  5  6  7  8  8  9  9  9  9  8  9  9  *  * [5][6][5] 9  *  9  8  8  8  8  8  8  8  8  8  8  7  7  6  4  3  2  1  +  .  .  . [.]|
  | 4  5  5  6  7  8  9  *  *  *  *  *  *  *  *  * [7][7][7] *  *  *  *  *  *  *  *  *  9  8  8  7  7  6  4  3  2  1  +  .  .  .  . |
->| 4  5  5  6  7  9  * [8][7][6][5][5][5][5][5][5][4][4][4][4][4][5][5][5][5][5][5][6][7][8] *  9  8  7  7  6  4  3  2  1  +  .  .  .  . |
  | 4  5  6  7  *  * [6][5][3][3][2][2][2][2][2][2][2][1][2][2][2][2][2][2][2][2][3][4][5][7] *  *  8  7  6  5  3  2  1  +  .  .  .  . |
  | 5  5  6  7  9 [2][2][2][1][1][1][1][1][+][+][+][+][+][+][+][+][+][1][1][1][1][1][2][3][3] *  8  7  6  5  3  2  1  +  +  .  .  . |
  | 5  5  6  7  7 [.][+][+][+][+][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][+][+][+][+][+] 8  8  7  6  5  3  2  1  1  +  .  .  . |
  | 5  5  6  7  5  2 [.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.][.] 2  6  8  7  6  5  4  2  1  1  +  .  .  . |
  | 5  5  6  6  5  2  +  .  .  .  .  .  .  .  .  . [.][.][.] .  .  .  .  .  .  .  .  .  .  + 2  6  7  7  6  5  4  2  1  1  +  .  .  . |
  | 5  5  6  6  4  2  +  .  .  . [.][.] .  .  .  . [.][.][.] .  . [.][.] .  .  .  .  .  . 1  3  5  7  7  6  5  4  2  1  1  +  .  .  . |
  | 5  5  5  5  4  2  1  .  .  . [.] .  .  .  . [.][.][.][.] .  . [.] . [.] . [.] .  .  1  3  5  6  7  6  5  4  2  2  1  +  .  .  . |
->| 5  5  5  5  4  2  1  .  .  . [.] . [.] .  . [.][.][.][.] . [.] .  . [.][.] .  .  + 1  3  5  6  6  6  5  4  3  2  1  +  .  .  . |
  | 5  5  5  5  4  2  1  +  .  . [.] . [.] .  .  . [.][.][.] .  .  .  .  .  .  .  .  + 1  3  5  6  6  6  5  4  3  2  1  +  .  .  . |
  | 5  5  5  4  4  2  1  +  .  .  .  .  .  .  .  . [.][.][.] .  .  .  .  .  .  .  .  + 1  3  4  5  6  5  5  4  3  2  1  +  +  .  . |
  | 5  5  5  4  3  2  1  +  .  .  .  .  .  .  .  . [.][.][.] .  .  .  .  .  .  .  .  + 1  3  4  5  6  5  5  4  3  2  1  +  +  .  . |
  | 5  5  5  4  3  2  1  +  .  .  .  .  .  .  .  . [.][.][.] .  .  .  .  .  .  .  .  1  2  3  4  5  5  5  4  4  3  2  1  +  +  .  . |
  | 5  5  5  4  3  2  1  +  .  .  .  .  .  .  .  . [.][.][.] .  .  .  .  .  .  .  + 1  2  3  4  5  5  5  4  3  3  2  1  1  +  .  . |
  | 5  5  5  4  3  2  1  1  +  .  .  .  .  .  .  .  . [.] .  .  .  .  .  .  .  .  + 1  2  3  4  5  5  5  4  3  3  2  1  1  +  .  . |
  | 5  5  5  4  3  2  1  1  +  .  .  .  .  .  .  .  . [.] .  .  .  .  .  .  .  .  + 1  2  3  4  5  5  5  4  3  3  2  1  1  +  .  . |
->| 5  5  4  4  3  2  1  1  +  .  .  .  .  .  .  .  . [.] .  .  .  .  .  .  .  .  + 1  2  3  4  4  5  5  4  3  3  2  1  1  +  .  . |
  | 5  5  4  4  3  2  1  1  +  .  .  .  .  .  .  .  . [.][.][.] .  .  .  .  .  .  .  + 1  2  3  4  4  5  5  4  3  3  2  1  1  +  +  . |
  | 5  5  4  4  3  2  2  1  +  .  .  .  .  .  .  .  . [.][.][.][.] .  .  .  .  .  .  + 1  2  3  4  4  5  4  4  3  3  2  1  1  +  +  . |
  | 5  5  4  4  3  2  2  1  +  .  .  .  .  .  .  .  . [.][.][.][.][.] .  .  .  .  .  + 1  2  3  4  4  4  4  4  3  3  2  1  1  +  +  . |
  | 5  5  4  4  3  2  2  1  +  .  .  .  .  .  .  .  .  . [.] .  .  .  .  .  .  .  .  + + 1  2  3  3  4  4  4  4  3  3  2  1  1  +  +  + |
  | 5  5  4  4  3  2  2  1  +  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  + 1  1  2  3  3  4  4  4  4  3  3  2  1  1  +  +  + |
  | 5  5  4  4  3  2  2  1  +  +  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  + 1  1  2  3  3  4  4  4  4  3  3  2  1  1  +  +  + |
  | 5  5  4  4  3  2  2  1  +  +  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  + 1  1  2  3  3  4  4  4  4  3  3  2  1  1  +  +  + |
->| 5  4  4  4  3  2  2  1  +  +  .  .  .  . [.][.] .  . [.] .  .  .  .  .  .  .  .  + 1  1  2  3  3  4  4  4  4  3  3  2  1  1  1  +  + |
  +-----------------------------------------------------------------------------------------------------------------------------------------+

Time: 0.557237
Result: 3710, 105971, 30276, 83356, 80400, 77902, 79165, 31418, 52557, 459, 294, 299, 6668, 40531, 51259, 15199, 94, 485, 24264, 37131
```

Figure 1: Example of the program output in debug mode for the indicated argument values.

The following symbols are used to represent the state of each control point:

`[ ]`: One or more particles/obstacles in the position.

·: Air pressure in the range $[0, 0.5)$.

`+`: Air pressure in the range $[0.5, 1.0)$.

`n`: A number n represents a pressure between $[n, n + 1)$

`*`: Air pressure equal or grater than 10.

2 Sequential code description

Program arguments The program arguments describe the simulation scenario:

1. `rows, columns:` Number of array positions
2. `max_iter:` Maximum number of simulation iterations to execute.
3. `threshold:` Minimum variability of the propagation on any updated cell. If no update is above this threshold the propagation has achieve a very stable situation and the simulation ends.
4. `inlet_pos:` Position of the first column of the inlet.
5. `inlet_size:` Number of columns of the inlet.
6. *Description of a band of random fixed particles:*
   - First row of the band
   - Number of rows
   - Density: Approximate ratio of particles per array position. Between 0.0 and 1.0.
7. *Description of a band of random flying particles:*
   - First row of the band
   - Number of rows
   - Density: Approximate ratio of particles per array position. Between 0.0 and 1.0.
8. `short_rnd1, 2 and 3:` Three unsigned short numbers that are used as seeds to initialize random generators to initialize the particles in random bands, and during the simulation for the inlet noise
9. Optional: A list of non-random fixed particles. Each one described by three values: Position row and column, and resistance between 0.0 and 1.0.

Students are encouraged to create their own scenarios, to test different situations and problems. Several examples are provided along with the code.

Results The program shows at the end of the simulation the following results. The execution time of the simulation (without initialization times) and a list of values that is used to verify the correctness of the simulation.

Debug mode If the program is compiled with the flag -DDEBUG (option included in the provided makefile) a piece of code is activated to print with ASCII art a representation of the simulation state at the end of each iteration (as in the previous figure). It can help to detect errors, simulation deviations, or simply to show its evolution to obtain a better idea of how it works. The arguments and starting data are also shown to have a record of the arguments used to create the scenario.

3 Project goal

Use the parallel programming model proposed by the teacher to parallelize this program without changing the algorithm. Optimize the code and obtain the best possible performance. Always check that the results are correct and equivalent to the sequential execution with the same input arguments.

A note about random numbers and parallelism The generation of a pseudo-random sequence of numbers with the classical C library functions is inherently sequential. Our program uses library functions that compute the next random value using a small input array that stores the state of a random sequence after each call. Take care to avoid changing the order in which random numbers from a given sequence are obtained, or the simulation results will change and the program will be non correct.

Code modifications allowed Students can modify the sequential code provided as long as they observe the following restrictions:

- Exploit parallelism using only the parallel programming model proposed.
- The argument processing section, array memory allocations, time measurement points, and output of results, should not be changed. The section of code that the students should target and modify is clearly identified in the main function. This section is found between the points where the time measurement is started and ended. Functions defined in the program that are called from the target section of the code can also be modified, substituted or eliminated.
- Any change in the algorithm or data structures must be discussed with the teachers in advance, in order to avoid modifications that significantly alter the parallelism exploitation with the proposed parallel programming model, which is the purpose of the assignment.

To measure execution times, compile the program with the maximum optimization level of the compiler (for example gcc -O3). We want to focus on program changes that do not interfere, and even facilitate, compiler optimizations.