Mini-project 2: Traffic Jams

CX 4230, Spring 2024[1]

Due Mar 29 at 11:59 pm

[1] Last updated: `Thu Mar 7 18:25:24 EST 2024`.

In this mini-project, you will implement models of one-dimensional traffic flow on a freeway, based on models described in the reading,[2]

> H-J. Bungartz et al. (2014). *Modeling and simulation: an application-oriented approach*. Springer. [Online text via GT Library Proxy]

[2] Hans-Joachim Bungartz, Stefan Zimmer, Martin Buchholz, and Dirk Pflüger. *Modeling and simulation: an application-oriented introduction*. Springer, 2014

Here are the basic ground rules:

- You may work individually or in pairs. See section 2.
- You can implement your solution in any programming language.
- You will submit a written write-up (PDF file) *and* your source code. See section 3.
- The assignment is due on Sunday, March 31, at 11:59 pm. See section 3 for the late policy.

Context. The goal of this mini-project is to model the behavior of freeway traffic. Such behavior was measured for traffic flowing in one direction on a British highway and is depicted in fig. 1.

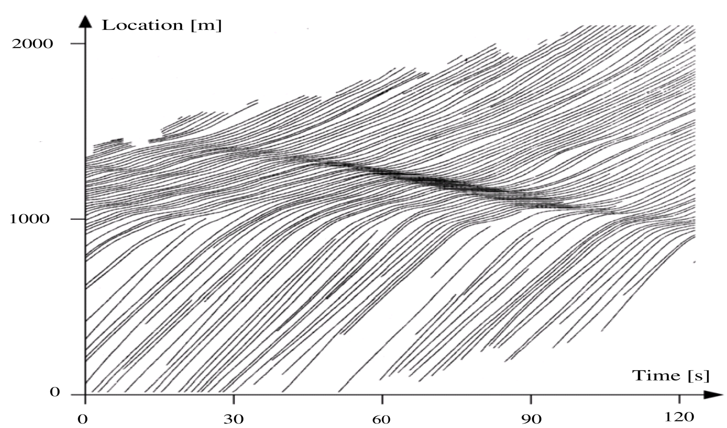

Figure 1: Traffic trajectories: Each line traces the location of a vehicle (y-axis) over time (x-axis). Regions of slower-moving traffic and jamming correspond with areas where there are higher densities of lines having nearly zero slopes.

Each line of fig. 1 represents the trajectory of a car, that is, its position (y-axis) over time (x-axis). The slope of a line at a given point in time represents a vehicle's instantaneous speed, with lower slopes meaning lower speeds. Additionally, higher densities of lines represent closely spaced cars; the combination of high density and small slopes corresponds to traffic jams. You can see such a jam forming in the region between the 1 km and 1.4 km, and it sharpens around time 30 s and persists, appearing to "move backwards." This type of behavior in the study of fluids is referred to as a *shock wave*.

1 *A CA-based model (50 points)*

Your task is to implement cellular automata (CA) models of traffic flow, replicating the experiments and results given in Chapter 8 of Bungartz et al. 2014.

1.1 *Task 1: A first CA model*

Implement the CA model described in Chapter 8.2 of Bungartz et al. 2014,[3] (Algorithm 8.1).

When your implementation is complete, do the following.

[3] You may need to read the background material in 8.1, too.

1. Verify your implementation, that is, explain how you know it is correct (i.e., relative to the specification, Algorithm 8.1).

2. Create plots like Figure 8.4 for occupancies of {10%, 25%, 50%, 80%}. Plot your results for several time steps to show the overall pattern. Describe what you observe; does it match the claims of Chapter 8.2?

1.2 *Task 2: Stochastic behavior*

Chapter 8.3 of Bungartz et al. 2014 suggests a small modification to the previous model, which randomizes the behavior of each car. Implement this change (Algorithm 8.2) and then do the following:

1. Try an initial impulse but with a "dally factor" of $p = 0.2$. How do these results compare to Task 1?

2. Repeat the experiments from Task 1 (random initial distribution at occupancies of {10%, 25%, 50%, 80%}) for $p \in \{0.1, 0.2, 0.5\}$. Plot the results in a similar fashion (i.e., like Task 1 and Figure 8.4). How do these results compare to the first CA model? How do they compare with Figures 8.5-8.7 of the reading?

1.3 *(**Teams only**) Task 3: Fundamental diagrams*

Read about *fundamental diagrams* in Chapter 8.3.3 and summarize them in your own words. Then, conduct simulation experiments suggested in Chapter 8.3.3 and try to reconstruct Figure 8.9 (top and bottom left) from your results.

1.4 Task 4: Two lane traffic

Suppose we wish to extend the stochastic one-lane CA model to *two* lanes of traffic moving in the same direction with the following behaviors in mind:

- **Differences in maximum speed by lane.** Denote the lanes by a "left" and a "right." Suppose that the maximum possible speed in the left lane can be *higher* than the maximum speed in the right lane. Denote the maximum speed in the left lane as v_L and in the right lane as v_R, where $v_L \geq v_R$. Both maximum speeds will need to be given as input parameters to the simulation.
- **Acceleration.** Assume that the process of accelerating is the same as the one-lane model: a driver in a given lane always tries first to accelerating with a delay up to the maximum speed of their lane.
- **Lane inertia.** Assume that a driver prefers to stay in whatever lane they are in and continuing accelerating toward the maximum speed for that lane. Once they reach that maximum speed, they stay there as long as they can do so.
- **Cautious lane changes.** If the driver sees they must decelerate to stay in their lane, they first check whether they can drive *at their current speed* in the adjacent lane *without* causing an accident. That is, there must be enough space in the adjacent lane so that a) they can switch to the lane and move forward by the number of steps equal to their current speed without hitting the next car that would be ahead of them, and b) the closest car that would be behind them can *both* accelerate *and* move forward. This case is subtle: other cars can *also* change lanes! Therefore, assume that a driver is *cautious*: when determining whether it is safe to move, the driver assumes that the closest cars behind it in *both* lanes *may* accelerate *or* change lanes, and that the closest cars ahead of it *may* decelerate *or* change lanes. The driver only changes lanes if there is *no* possibility of a collision.
- **Deceleration.** If a driver cannot accelerate and cannot safely change lanes, then the driver decelerates by the same process as the one-lane model.

Your tasks are as follows.

1. Formalize these rules into an algorithm (similar to Algorithms 8.1 and 8.2 of the reading).
2. Show your rules as both an algorithm along with an explanation in plain English.

3. Then implement your algorithm.
4. Explain how you have tested and verified your implementation.
5. Conduct experiments to see the effects.
6. Summarize your observations and assess whether the results corroborate or contradict what you might expect to see "IRL" (in real life).

2 *Teaming*

You may work individually or in pairs. Teams of two have additional requirements for the assignment, as noted above.

To "declare" your team, do the following:

- Visit the Canvas "People" page for Mini-project 2 ("MP2").[4]
- If you are working individually: Pick an unused team number and add yourself.
- If you are working in a team of two: Both of you should pick an unused team number and add yourself to the same unused team number.

[4] Link: `https://gatech.instructure.com/courses/372568/groups#tab-43871`

3 *How to Submit*

Create a **private** git code repository at `github.gatech.edu` (*not* `github.com`!) For teams, only one person needs to create the repository. Place both your code and a PDF report summarizing your results in this repo. You'll submit the URL to this repository on the page for this assignment in Canvas. In the repository itself, create a `README.md` file that tells us a) your individual or team number and member(s) of the team; b) which file is your PDF report; and c) what the organization of your code is (so we can inspect and evaluate it).

Important! Since your repository will be private, please be sure to add *everyone* from the teaching staff to your repo: the two TAs, Rahul Komatineni (`rkomatineni6`) and Youjie Zhang (`yzhang3988`), and Prof. Vuduc (`rvuduc3`). Otherwise, we will not be able to see and grade your submission and you will get a zero.

Late submissions. You get an automatic extension through the weekend: to submit without penalty, submit by Sunday, March 31, at 11:59 pm. There is a 10% penalty if you submit up to one day late (Monday, April 1), a 25% penalty if you submit two days late (Tuesday, April 2), and 50% penalty if you submit three days late (Wednesday, April 3). No assignments will be accepted after that time.

References

Hans-Joachim Bungartz, Stefan Zimmer, Martin Buchholz, and Dirk Pflüger. *Modeling and simulation: an application-oriented introduction*. Springer, 2014.