Traffic flow: numerical models

by E. Alec Johnson, March 17, 2009

I study computational *continuum mechanics*. Continuum mechanics is the study of how continuous stuff moves/flows. Computation is the use of computer algorithms to simulate physical systems.

I want to show you how to write down an equation that you can punch into a computer to model the flow of a substance.

1 Traffic flow

The example that I will use is the movement of cars along a highway. We will start with almost the simplest system we can imagine: All the cars are traveling on a one-lane, one-way road. Each driver sets his speed based on the distance to the car in front of him. There is a minimum distance (bumper to bumper) and a maximum speed (the defacto speed limit). For simplicity, we assume that cars are moving in the positive x direction (i.e. from left to right).

2 Discrete Cars

We want to write down a mathematical equation that describes how these cars move. The most basic way is to model the motion of each car.

- Let t represent **time** and
- let $x_i(t)$ represent the **position** of the i-th car as a function of time.
- Let $u(D)$ represent the **velocity** of a driver as a function of the distance D to the car in front of him. **Problem:** Draw a sketch of a reasonable-looking function $u(D)$ if the speed limit is 100 kilometers per hour and the typical car is 3 meters long.
- Let $D_i = x_{i+1} - x_i$ represent the distance of the of the i-th car to the car in front of it.
- Let $u_i(t)$ represent the velocity of driver i. Then $u_i(t) = u(x_{i+1}(t) - x_i(t))$

Recall that (average) velocity is change in position over change in time. So:

$$\Delta x_i \simeq \Delta t\, u_i.$$

How can you use this to write down a formula that computes an approximate value for $x_i(t + \Delta t)$ based on $x_i(t)$?

3 Continuum car density

Now let's take a continuum mechanics approach. Rather than modeling individual cars, we'll pan out and look at the road from the viewpoint of a helicopter. There are way too many cars to try to model each car. Instead we view the cars as a substance.

- Let $p(x,t)$ = density of cars, i.e. the number (or "amount") of cars per unit distance.
- Let $v(x,t)$ = velocity of cars.
- Assume that the velocity of the cars at each point is determined by the density of cars: $v(x,t) = V(p(x,t))$.

This function $V(p)$ is what determines how the cars will move.

Suppose that V is a linear function of p. Assume that $V(p_{\max}) = 0$ and $V(0) = V_{\max}$ (hopefully the speed limit).

1. Graph V as a function of p. Does the graph agree with your intuition? (What would you expect to happen to the velocity as the density increases from 0 to a "maximal traffic jam" $p_{\max}$?)
2. What is a formula for $V(p)$?

3.1 Discretization

Computers are finite and discrete, and continuum models are not, so to compute solutions for continuum models we have come up with discrete approximations. We will discretize our model by chopping up space into discrete intervals called *cells*, and partitioning time into discrete *time steps*. Our model will try to keep track of the amount of cars in each cell and how the amount of cars in each cell changes from one time

step to the next. We define the *flux* across an interface to be the number of cars that cross the interface in one time step. The basic idea is to try to estimate the flux of cars that flow across each cell boundary.

- Let Δx = width of each cell,
- let Δt = length of each time step,
- let x_j = center of j-th cell,
- let t^n = time of n-th time step,
- let P_j^n = number of cars in j-th cell at time step n.

Our computer algorithm will attempt to keep track of how many cars are in each cell. The basic idea is that **the number of cars in a cell at the end of a time step equals the number of cars in the cell at the beginning of the time step plus the flux of cars into the cell minus the flux of cars out of the cell.** So we can write down a formula for the number of cars in each cell at the end of a time step based on the number of cars in the cell at the beginning of the time step if we can just approximate the rate at which the cars flow across each cell boundary.

Let f represent the number of cars per unit time that are flowing from left to right past position x at a given time t.

- What is a formula for f in terms of the density p and the velocity V of the cars?

Let $F_{j+\frac{1}{2}}$ denote the number of cars that flow across the boundary between cell j and cell $j+1$ in one time step.

- What is a formula for P_j^{n+1} in terms of P_j^n, $F_{j+\frac{1}{2}}$, and $F_{j-\frac{1}{2}}$?

So if we are given the amount of cars in each cell at the beginning of a time step and the amount of cars that flow across each cell boundary we can say how many cars are in each cell at the end of the time step.

Now comes the most challenging part. We need a way to estimate the flux of cars across each cell boundary during a time step.

Here's a simple way that works if the car density does not exceed $p_{\max}/2$.

1. Pretend that the density of cars in each cell is uniformly equal to the average density $p_j := \frac{P_j^n}{\Delta x}$.
2. Compute the velocity of the cars in each cell using $v_j = V(p_j)$.
3. Since cars are moving to the right, compute the flux at interface $j + \frac{1}{2}$ using the overall density and velocity of cars in cell j. What formula does this give you for $F_{j+\frac{1}{2}}$?
4. Put everything together to get a formula for P_j^{n+1} in terms of P_j^n and P_{j-1}^n.

If you implement this method it will work as long as you choose p values that stay below $p_{\max}/2$; otherwise when the car density starts to form a shock you will begin to see oscillations in the solution that blow up and destroy it.

The problem with this recipe is that even though the *cars* are moving to the right, there is a *density wave* that might be moving to the left. In that case, since we are really keeping track of the density wave, we need to get our information from the *upwind* direction. Otherwise the solver is looking in the wrong direction for its update information and it cannot possibly give the right solution in the long run.

So for each cell we need to look to see which way the density wave is moving and then get the flux from the appropriate direction. To get a formula for the speed of the density wave would involve a discussion of the partial differential equations formulation of our model. It turns out that the velocity of the density wave is $\frac{df}{dp}$, the derivative of the flux with respect to density.

However, there is another viewpoint we can take. One way to stabilize our method is to make sure that it does not augment local extrema.[1] To obtain such a method, if $\frac{df}{dp}$ is positive we take the flux from the left, and if $\frac{df}{dp}$ is negative we take the flux from the right. This basically ensures that density does not pile up at a local max or leak from a local min. To obtain a value for $\frac{df}{dp}$ at an interface, you need an estimate of the value of p at the cell boundary, which you can get from averaging the p values of the cells on either side of the boundary. By a careful study of cases you can convince yourself that this method does not augment local extrema.

[1] The technical literature talks about methods being *Total Variation Diminishing* (TVD), which is basically the same condition, although slightly less strict.