Joel Chavali

Vistas In Advanced Computing

11 August 2017

Traffic Flow 1: Single Lane, Velocity Dependent Flow

**Abstract**:

**Equations**:

General Traffic Flow Equation:

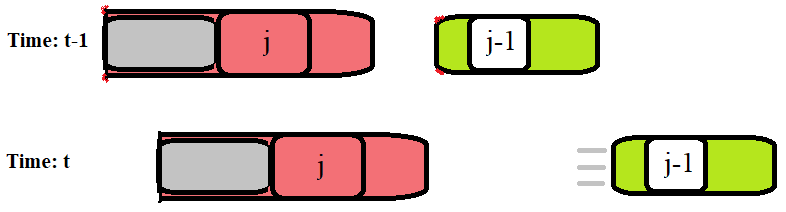
|  |  |
| --- | --- |
| Legend of Symbols | |
|  | Acceleration of the jth vehicle |
|  | Constant that user defines |
|  | Real Time Speed of Previous Car |
| (t) | Real Time Speed of Current Car |

In this model, the acceleration of the jth vehicle (current vehicle) is based on the difference in velocity of the jth and j-1 vehicle (current and immediately previous vehicle in the line of traffic). The model is designed to use real time velocities. As the difference in velocity increases, the acceleration follows suit to always try to match the vehicle in front of it. Since the equation does not take into consideration the position of the vehicle, crashes can occur if vehicles are near each other but vary greatly in velocity.

Optionally, the equation can be simplified to eliminate the term by altering .

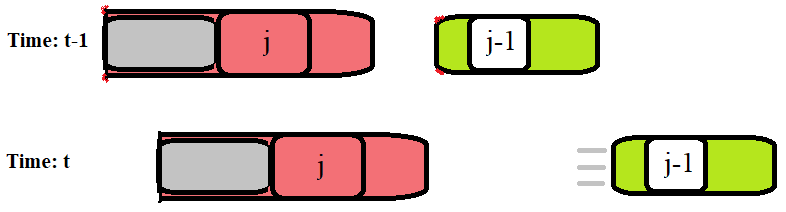
**Discretized Forms of Equations**:

Discretizing the equation requires two considerations. First the time and when things are happening as well as which vehicle in the line is having their velocity changed.



General Discretization:

However this has an error of order O(dt). Therefore, I used the 4th order Runge Kutta.

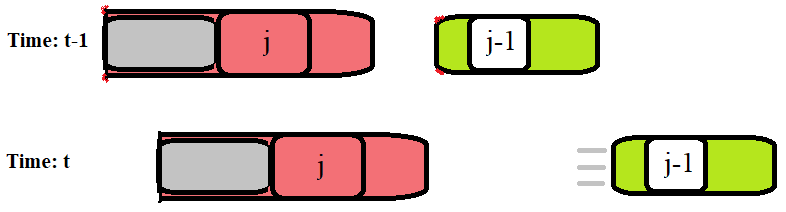


4th Order Runge-Kutta:

**Description of Numerical Methods with Pseudocode**:

The 4th Order Runge-Kutta is a versatile beast that uses a center weighted average of the slopes at various points along the discrete interval in which it is given. Using 4 instead of 1 approximation values allows it to get much closer to the actual solution.

This accuracy is important to the calculations as any error gets compounded later on. Additionally, all performance was measured in terms of displacement and not velocity. Since displacement relied on velocity, it was more important to have a higher accuracy on the front end of the calculations. Because displacement is inherently linked to velocity, the Runge-Kutta becomes slightly more complicated. However, it reduces to a simple concept of a three-legged race. In order to move on to the next step of the Runge-Kutta, both velocity AND displacement must be updated. The final pseudo code is below:



**Results**:

The results were an accurate representation of what should happen in a single lane traffic flow.

The model was subjected to 4 different situations testing the accuracy of the model as well as allowing the testing of several application scenarios that could occur on the roads today.

In all calculations only 10 vehicles consisted of the traffic flow. Other parameters are in tables next to graphs or within the captions of the graphs.

Situation 0: Steady State:

Steady Traffic Flow at a single velocity. Not very interesting but important to test the accuracy of the code and the graphing utilities.

GRAPH TABLE

Situation 1: Thirty Percent Velocity Increase

Steady Traffic Flow at a velocity but then the first car (which has no other vehicles in front of it) is instructed to accelerate until it has become thirty percent faster than its originally programmed speed.

GRAPH TABLE

Situations like this are more realistic to driving experiences as they reflect that the driving conditions rarely stay constant. Small velocity changes on straight tracks are common on every drive and every stretch of road. Situation 1 can happen in reality for a numerous amount of reasons, including a lane change, aggressive driving behavior, or drivers responding to a faster speed limit.

Situation 2: Thirty Percent Velocity Decrease

Exactly the same as Situation 1, but now the reverse response is necessary. Steady Traffic Flow at a velocity but then the first car (which has no other vehicles in front of it) is decelerates until it has become thirty percent slower than its originally programmed speed.

GRAPH TABLE

Situation 2 can happen in reality for a numerous amount of reasons, including a lane change, defensive driving behavior, or drivers noticing law enforcement a couple thousand feet down the road.

Situation 3: From Rest to Forty Two Meters/Second with constant Acceleration

All vehicles start at rest with an initial separation distance. Then the first car accelerates at maximum acceleration until it hits the maximum velocity it can go. Once met the problem turns into a steady state problem.

GRAPH TABLE

In real life, this has applications at traffic lights, stop and go traffic, and highway on ramps.

Situation 4: Deceleration From an Initial Speed to a Complete Stop using Full Braking Power

All vehicles start with an initial velocity and an initial separation distance. The first car proceeds to use full braking power to decelerate all the way to rest. Once at rest, the car stays at rest. In the perspective of traffic models, this situation helps to determine the minimum safe following distance assuming that the reaction time is 0. Although this is never the case, it’s a good reference for the traffic models to start considering the safety of drivers and other passengers.

GRAPH TABLE

On the road, this type of behavior is usually exhibited only in emergency conditions: Responding to a crash, a pedestrian on the road, an unexpected obstruction, but it can also occur in different scenarios as well such as racing.

Additionally, some parameters were varied to observe their effects on traffic flow. Since human judgement is complex and irregular or inconsistent, the application of this structured variance, unmodified, to such a highly irregular system would not be very helpful, so, this project identifies parameters that could be improved with the development of automated traffic.

First, initial separation distances were changed to find the most optimal separation distance.

GRAPH

The optimal separation distance is 38 meters with a lambda of 0.9. This was found by varying the separation distances and holding everything else constant. Situation 4 was used in determining this distance as a crash would easily indicate a failure in the system. However, this minimum separation distance is impossible to safely achieve in reality as there is a time delay (and therefore distance cost) between the operator (whether it be human or computer) sensing an obstruction and the mechanical application of the brake. As a result, all “optimized” driving separation distances were padded to 50 meters of separation.

Lambda was also varied to observe the results.

GRAPH

The higher the lambda, the faster each vehicle’s reaction was to the change in speed of its forward neighbor. This means a much safer road as vehicles would react much faster to each other’s changes and maintain a safe following distance. However, this also came with a unique consequence. Higher lambda’s mean faster reactions but that means, each vehicle would end up maintaining their initial following distance much better. This ends up being a problem because vehicles stopping will always maintain their distance and never closes the gap on high density, high use roads.

Application:

As stated before, the main application of these findings is to automated driving developments. In a fully automated world in which each vehicle has reliable sensors and precise data about the vehicles around it, models like this can find the maximum load a single, one-way road can handle while maintaining an exceptional margin of safety for the passengers and vehicles on it.

Primary Limitation:

However, this model is not without its limitations. As stated earlier, since the model is solely based off of difference in speed to change acceleration, the model cannot save space (and therefore increase vehicle load on a single road) between slower or stopped vehicles. A major improvement on this model is the Traffic Flow II and the Traffic Flow III models that my colleagues have made.

Model Specific Suggestions:

There were many goals that I had for the project that were in progress or were scrapped at the time this report was submitted. So, some suggested research and development for this code would be to implement a single click or reduced click input creation, application runtime, and visualization. Currently the input creation and application runtime exists and is functional but the visualization is not. The application is designed to have vehicles with different parameters. For instance, all the vehicles do not necessarily have to have the same lambda but for development time constraints, at the submission of this report, they all do. Parameters subject to change is lambda, vehicle length, initial separation distances, and initial velocity. Maximum acceleration and maximum velocity can also be added with just a few more lines of code. My main personal goal was to see the what the minimum following distance should be of each car with a variety of different vehicles on the road like trucks, buses, motorcycles, and cars each with different parameters but I simply ran out of time. Basically, the whole point of this final paragraph is to say, there is more work to be done and I’m excited that I won’t be stopping yet.

**Technical Specifications of Computer Used**:

Processor: Intel® Core™ i5 CPU M 560 @ 2.67GHz

Installed RAM: 4.00 GB

Local Storage: 297 GB

System Type: 64-bit operating system, x64-based processor

Windows Edition: Windows 10 Pro Insider Preview

Windows Version: 1703

OS Build: 16251.0