

Traffic Waves

New Mexico Adventures
in Supercomputing Challenge
Final Report
Team 043
Manzano High School

Team Members

- Adam Davie
- Kyle Fuerschbach

Teachers

- Stephen Schum
- Mr. Mathews

Table of Contents

- Introduction
- Project Description
- Research
- Mathematical Model
- Method Of Solution
- Code
- Results
- Conclusions
- Acknowledgements
- References

Introduction

This project involves a phenomenon called a traffic wave. A traffic wave occurs when a group of cars is slowed down or stopped as a result of a single car or other obstruction along a road. The cars will be temporarily slowed or stopped and then resume movement once again. This slowing and resuming occurs in a ripple effect through the rows of cars, which is why it is called a traffic wave.

The modeling of traffic waves is very important to a wide range of professional fields. Most importantly, traffic models are used because they are more practical than experimentation with actual vehicles. By applying traffic waves to computer code traffic modeling becomes much more efficient in many areas. Modeling traffic is also less expensive than a real life simulation. Not to mention, results are obtained in a significantly quicker, and there are no possibilities for disruptions to traffic flow.

Traffic models can be used by traffic engineers, construction engineers, the Highway Department, vehicle manufacturers, and many other professionals. Traffic engineers can determine how many cars a road needs to be capable of supporting as the population of an area increases. Construction engineers use traffic models for highway and road design, and to determine what kind of an impact construction, such as a lane closure, could have on traffic flow. Traffic models can also be used for bridge design to see how a bridge would hold up if a traffic jam occurred over a bridge. Vehicle manufacturers can also use traffic models to determine if certain vehicles will overheat if stationary in traffic for a long period of time.

Project Description

This project involves creating a program that will compute the positions of cars as they are affected by a traffic slow down, or traffic wave. There are several scenarios in which cars can be affected by a traffic wave. The slowdowns occur as a result of situations that can include object avoidance, construction, lane merging, and traffic lights. Obviously, modeling all of these situations would be nearly impossible considering how complex a traffic wave can be. We hope to focus on just one of these cases, most likely a scenario involving object avoidance on a highway.

For this situation, there are many possibilities for error in the results. For our project, we want to eliminate as many possibilities for error as possible. By doing this, the results that are outputted should be relatively close to that of real-world data. Of course, it's impossible to get exact data due to the fact that human individuality cannot be programmed. There may be people who want to drive very fast, and others that want to drive very slowly. The best way to estimate this human individuality is to create a rare occurrence of a vehicle that will drive at an extreme speed relative to the other cars. Once the data from the program is produced, it will be compared to that of real life data that comes from the Highway Department.

With the data we get from our program we will create a graphical output to better represent the concept of a traffic wave. The best way to simulate a traffic wave would be to build an applet in which the user would input the variable to understand how they affect the traffic flow. Unfortunately, an applet of this magnitude is very complex and extremely hard to create. We are going to make a simple graphical display that outputs the results from our program, and still gives an accurate representation of how the traffic wave works.

Research

Through our research we have found many resources to help us in this project. Traffic is a large field of study when it comes to modeling and testing. Everyone wants traffic to be eliminated, and by doing modeling and testing we come one step closer to achieving that goal. We found on the internet people who for a hobby study traffic waves and look for a cure to the problem. Also on the internet we have found numerous codes and applets that model traffic. One of the greatest sources of research for us was a website called traffic forum. This was a website dedicated to talking about traffic engineering, research, testing, and math. This website was a great spot for us to set the bar and figure out where we wanted to go, however, we did stray far away from anything that was proposed on the website. Mostly everything on their website was beyond my math and engineering skills. Another great source for inspiration for us was a java applet that models traffic flow. This applet was amazing. It showed real time traffic jams forming in many circumstances including changing lanes and traffic lights. This website showed us what kind of programming was possible in the field of traffic. Another website that was helpful to our mathematical model was a traffic competition website. This competition was for people to create computer program to model traffic. This website provided us with great ideas and inspiration for our mathematical model. Unfortunately, the code on that page did not work because I was really hoping to see what kind of code they made.

Method of Solution

Because this project was very complex, it was approached with an incrementing system to dissect the process little by little until the process was complete. We broke up the program into four cases in which we gradually approached the final code. These are the four cases:

Case 1

- There will be 1 lane and 4 cars
- Cars will be of constant length = 14 ft.
- Initial velocity will be constant for all cars ($v=c=60\text{mph}$)
- Initial spacing between cars will be 1 car length/10 mph. So spacing will be $14\text{ft.} \cdot (60\text{mph}/10\text{mph})=84\text{ft.}$
- Cars will continue at constant velocities.
- Positions of cars will be determined by $d=v(\Delta t)$ at 0.1s intervals

Case 2

- Still 1 lane and 4 cars
- Cars will be of constant length = 14 ft.
- Initial velocity will be constant for all cars ($v=c=60\text{mph}$)
- Initial spacing between cars will be 1 car length/10 mph. So spacing will be $14\text{ft.} \cdot (60\text{mph}/10\text{mph})=84\text{ft.}$
- An obstacle will be introduced, and the first car will decelerate at $a=-0.05\text{m/s}^2$
- The delay before decelerating will be $t=0.7\text{s}$ for each car
- Positions of cars will be determined by $d=d_0+v_0t+0.5at^2$

Case 3

- 2 lanes, 4 cars
- Cars will be of constant length = 14 ft.
- Initial velocity will be constant for all cars ($v=c=60\text{mph}$)
- Initial spacing between cars will be 1 car length/10 mph. So spacing will be $14\text{ft.} \cdot (60\text{mph}/10\text{mph})=84\text{ft.}$
- An obstacle will be introduced, and the first car will decelerate at $a=-0.05\text{m/s}^2$
- The delay before decelerating will be $t=0.7\text{s}$ for each car
- Cars will move over a lane to avoid slowing down
- Positions of cars will be determined by $d=d_0+v_0t+0.5gt^2$ @ 0.1s intervals

Case 4

- 3 lanes
- Multiple cars (20)
- Vary speeds to introduce vehicle interaction
- Positions of cars will be determined by $d=d_0+v_0t+0.5gt^2$ @ 0.1s intervals

Mathematical Model

The mathematical model for our project is fairly simple. The first and most important part of the model is the initial assumptions. Though we tried to eliminate as many assumptions as possible, it is impossible not to have any assumptions. In our situation we had to simplify many areas. For example, driver tendencies had to be simplified. We could not include everything that a driver may or may not do. For instance, drivers who ride the brakes could not be brought into the model. We had to assume that our driver drive the exact same and that means that for right now there is no reaction time. All are car drivers have a perfect reaction time. Also, we had to simplify the car situation. Some cars are faster and some cars are heavier. We did not include each and every car that can be seen on the roads, for that would be very complicated to the model and code. As for the model it follows basic physic formulas.

- $d=d_0+v_0t+1/2at^2$
- $v=v_0+gt$
- $d=vt$

These formulas are used throughout the model. As for the lead car we use the formula $d=vt$ because as of now the velocity is constant. The cars that follow the lead car, however, use acceleration because they have to slow down to keep a safe distance from the lead car. A certain distance is set in between cars. This distance follows the rule that one would learn in driver education. One should always try to keep a distance of ten feet per ten miles per hour. Since our code is in standard units that is approximately 3 meters per 4.4 m/s. Each car behind the first car has a specific speed it is trying to reach. However, if this speed conflicts with the lead car, the second car will have to slow down. These cars slow down according to distance it is from the car ahead. Right now the cars will slow down in increments of $-.05 \text{ m/s}^2$. The amount at which the car slows down depends on it distance to the car ahead. For example with a lead car going 26 m/s and a second car of 28 m/s, they try to stay spaced around 18 meters apart. If the cars are less than 4.5 meters apart it will slow down $-.50 \text{ m/s}^2$. If the cars are less than 9 meters apart it will slow down $-.20 \text{ m/s}^2$. If the cars are less than 18 meters apart it will slow down $-.05 \text{ m/s}^2$ and so on. This model works for some cases. The problem is that this model is too simplified. In later models the slow down of a car will work for all situations including if the lead car is going 25 m/s and the next car is going 50 m/s.

Code

```
//Author: Kyle Fuerschbach, Adam Davie whilemovingdiffdeltawg.cpp 04/07/04
/* This program is a traffic simulation that models traffic buildup. This
simulation consists of one lane and two cars. The lead car has a set speed
and the following car has a set speed that it tries to achieve but must
remain at a safe distance from the first car. This distance is 3 meters
per 4.4 m/s.*
```

```
#include <iostream.h>
#include <iomanip.h>
#include <string.h>
```

```
#include <math.h>
```

```
int main()
```

```
{  
    float carA[200];  
    float carB[200];  
    float t[200];  
    float deltat(.1);  
    float diff(0),apart(0);  
    float accel[200];  
    int iA(0), i(0);  
    int maxnum;  
    float tmax(0),tBstart(0),tCstart(0);  
    float vAi(0),vBi(0);  
    float vA[200],vB[200];  
    cout<<"Enter the velocity of car A in m/s."<<endl;  
    cin>>vAi;  
    cout<<"Enter the velocity of car B in m/s."<<endl;  
    cin>>vBi;  
    apart=(vBi/4.4)*3;  
    cout<<"Car B will try to stay "<<apart<<" m from Car A."<<endl;  
    cout<<"<<endl;  
    cout<<"Enter the amount of time to model the situation."<<endl;  
    cin>>tmax;  
    maxnum = int(tmax * 10);  
    for(iA=0; iA<=maxnum; iA++)  
    {  
        carA[iA]=0;  
        carB[iA]=0;  
        vA[iA]=0;  
        vB[iA]=0;  
        t[iA]=0;  
        accel[iA]=0;  
    }  
    for(iA=0; iA<=maxnum; iA++)  
    {  
        t[iA] = float(iA/10.0);  
        vA[iA]=vAi;  
        carA[iA]= t[iA]*vA[iA];  
        if(carA[iA]<14.3)  
        {  
            carB[iA]=0;  
        }  
        else  
        {  
            if((carA[iA]-carB[iA-1])<=apart)  
            {  
                accel[iA]=-0.05;  
                vB[iA]=vBi+(accel[iA]*t[iA]);  
                carB[iA]= carB[iA-1]+(deltat*vB[iA-1])+(0.5*accel[iA]*pow(deltat,2));  
            }  
            if((carA[iA]-carB[iA-1])<=(apart/2))  
            {  
                accel[iA]=-0.20;  
                vB[iA]=vBi+(accel[iA]*t[iA]);
```

```

        carB[iA]= carB[iA-1]+(deltat*vB[iA-1])+(0.5*accel[iA]*pow(deltat,2));
    }
    if((carA[iA]-carB[iA-1])<=(apart/4))
    {
        accel[iA]=- .50;
        vB[iA]=vBi+(accel[iA]*t[iA]);
        carB[iA]= carB[iA-1]+(deltat*vB[iA-1])+(0.5*accel[iA]*pow(deltat,2));
    }
    if((carA[iA]-carB[iA-1])>=apart && (carA[iA]-carB[iA])<=(2*apart))
    {
        accel[iA]=- .025;
        vB[iA]=vBi+(accel[iA]*t[iA]);
        carB[iA]= carB[iA-1]+(deltat*vB[iA-1])+(0.5*accel[iA]*pow(deltat,2));
    }
    if((carA[iA]-carB[iA-1])>=(2*apart))
    {
        accel[iA]=.00;
        vB[iA]=vBi+(accel[iA]*t[iA]);
        carB[iA]= carB[iA-1]+(deltat*vB[iA-1])+(0.5*accel[iA]*pow(deltat,2));
    }
}
cout<<"Time"<<setw(15)<<"D of Car A"<<setw(12)
<<"V of Car A"<<setw(12)<<"D of Car B"
<<setw(12)<<"V of Car B"<<setw(12)<<"Accel of B"<<endl;
for(i=0; i<=maxnum; i++)
{
    cout.precision(2);
    cout.setf(ios::showpoint|ios::fixed);
    cout<<t[i]<<setw(12)<<carA[i]<<setw(12)<<vA[i]<<setw(12)<<carB[i]<<setw(12)
    <<vB[i]<<setw(12)<<accel[i]<<endl;
}
return 0;
}

```

Results

The results that we have so far are for a case in which there is one lane and two cars in that lane. The cars are inputted at user specified velocities and when the second car reaches the first car, it decelerates. The car tries to maintain its velocity, but when it gets within a certain distance range from the car in front of it, it decelerates. Below are some outputted results from the code. Gravity was not a factor in these results.

For a situation where the velocity of Car A is 26 m/s and the velocity of Car B is 28m/s and Car B will try to stay 19.09 meters from Car A. This will be modeled for 7 seconds.

Time	D of Car A	V of Car A	D of Car B	V of Car B	Accel of B
0.00	0.00	26.00	0.00	0.00	0.00
0.10	2.60	26.00	0.00	0.00	0.00
0.20	5.20	26.00	0.00	0.00	0.00
0.30	7.80	26.00	0.00	0.00	0.00

0.40	10.40	26.00	0.00	0.00	0.00
0.50	13.00	26.00	0.00	0.00	0.00
0.60	15.60	26.00	0.00	27.97	-0.05
0.70	18.20	26.00	2.80	27.97	-0.05
0.80	20.80	26.00	5.59	27.96	-0.05
0.90	23.40	26.00	8.39	27.95	-0.05
1.00	26.00	26.00	11.18	27.95	-0.05
1.10	28.60	26.00	13.98	27.94	-0.05
1.20	31.20	26.00	16.77	27.94	-0.05
1.30	33.80	26.00	19.57	27.93	-0.05
1.40	36.40	26.00	22.36	27.93	-0.05
1.50	39.00	26.00	25.15	27.92	-0.05
1.60	41.60	26.00	27.94	27.92	-0.05
1.70	44.20	26.00	30.74	27.92	-0.05
1.80	46.80	26.00	33.53	27.91	-0.05
1.90	49.40	26.00	36.32	27.91	-0.05
2.00	52.00	26.00	39.11	27.90	-0.05
2.10	54.60	26.00	41.90	27.90	-0.05
2.20	57.20	26.00	44.69	27.89	-0.05
2.30	59.80	26.00	47.48	27.89	-0.05
2.40	62.40	26.00	50.26	27.88	-0.05
2.50	65.00	26.00	53.05	27.88	-0.05
2.60	67.60	26.00	55.84	27.87	-0.05
2.70	70.20	26.00	58.63	27.86	-0.05
2.80	72.80	26.00	61.41	27.86	-0.05
2.90	75.40	26.00	64.20	27.85	-0.05
3.00	78.00	26.00	66.98	27.85	-0.05
3.10	80.60	26.00	69.77	27.84	-0.05
3.20	83.20	26.00	72.55	27.84	-0.05
3.30	85.80	26.00	75.34	27.83	-0.05
3.40	88.40	26.00	78.12	27.83	-0.05
3.50	91.00	26.00	80.90	27.83	-0.05
3.60	93.60	26.00	83.68	27.82	-0.05
3.70	96.20	26.00	86.47	27.82	-0.05
3.80	98.80	26.00	89.25	27.81	-0.05
3.90	101.40	26.00	92.03	27.81	-0.05
4.00	104.00	26.00	94.81	27.80	-0.05
4.10	106.60	26.00	97.59	27.80	-0.05
4.20	109.20	26.00	100.37	27.79	-0.05
4.30	111.80	26.00	103.15	27.78	-0.05
4.40	114.40	26.00	105.92	27.78	-0.05
4.50	117.00	26.00	108.70	27.77	-0.05
4.60	119.60	26.00	111.48	27.77	-0.05
4.70	122.20	26.00	114.26	27.76	-0.05
4.80	124.80	26.00	117.03	27.76	-0.05
4.90	127.40	26.00	119.81	27.75	-0.05

5.00	130.00	26.00	122.58	27.75	-0.05
5.10	132.60	26.00	125.36	27.75	-0.05
5.20	135.20	26.00	128.13	27.74	-0.05
5.30	137.80	26.00	130.91	27.74	-0.05
5.40	140.40	26.00	133.68	26.92	-0.20
5.50	143.00	26.00	136.37	26.90	-0.20
5.60	145.60	26.00	139.06	26.88	-0.20
5.70	148.20	26.00	141.75	26.86	-0.20
5.80	150.80	26.00	144.43	26.84	-0.20
5.90	153.40	26.00	147.11	26.82	-0.20
6.00	156.00	26.00	149.79	26.80	-0.20
6.10	158.60	26.00	152.47	26.78	-0.20
6.20	161.20	26.00	155.15	26.76	-0.20
6.30	163.80	26.00	157.83	26.74	-0.20
6.40	166.40	26.00	160.50	26.72	-0.20
6.50	169.00	26.00	163.17	26.70	-0.20
6.60	171.60	26.00	165.84	26.68	-0.20
6.70	174.20	26.00	168.51	26.66	-0.20
6.80	176.80	26.00	171.17	26.64	-0.20
6.90	179.40	26.00	173.83	26.62	-0.20
7.00	182.00	26.00	176.49	26.60	-0.20

Conclusions

The results that we got from the program seem to be accurate with results that we have checked online. From the accuracy, we can conclude that our code is doing what we intended it to do. From this point, there is still much improvement that can be made to make the program more complex and also more accurate. Over the next few weeks we are going to be adding much to our project, and the program will more complete. We are still going to add to the code to reach our goals. We want to make the best program that we can possibly make. As far as improvements, there are many ideas that we are going to concentrate. Our program is still very basic in terms of simulating a traffic wave. We want to add complexity to our program so that it can output results that are similar to those of real world data. First, we are going to make the simulation include more lanes as well as many more cars, as it would be in the real world. Also, we want to better incorporate acceleration into the program and make the code work for all situations including those that are probable and highly improbable.

Acknowledgements

Mr. Schum - A special thanks to our teacher who makes the AiSC a possibility at our school. He has been the base for our enjoyment in this computing challenge.

Mr. Mathews - A student teacher who came halfway through the year, and was a big help in assisting us with our project.

References

- [TrafficForum](http://www.trafficforum.org)
www.trafficforum.org
- [Traffic Modeling](http://www.berkshireplanning.org/3/2/6/)
http://www.berkshireplanning.org/3/2/6/
- [Traffic Waves](http://www.amasci.com/amateur/traffic/traffic1.html)
http://www.amasci.com/amateur/traffic/traffic1.html
- [The Roles of Traffic Models](http://ops.fhwa.dot.gov/Travel/Traffic_Analysis_Tools/roles_of_transportation_models.htm)
http://ops.fhwa.dot.gov/Travel/Traffic_Analysis_Tools/roles_of_transportation_models.htm
- [Traffic Competition](http://www.math.toronto.edu/mathnet/plain/carcompet/model.html)
http://www.math.toronto.edu/mathnet/plain/carcompet/model.html
- [Mathematical Traffic Model](http://www.math.toronto.edu/mathnet/carcompet/model.html)
http://www.math.toronto.edu/mathnet/carcompet/model.html
- [Traffic Microsimulation](http://www.mtreiber.de/MicroApplet)
http://www.mtreiber.de/MicroApplet