Numerical Methods on Traffic Flow Models

(Optimal Velocity Model)

Group Members

- 15BME0155
- 15BCL0398
- 15BCL0428
- 15BCL0395
- 15BIT0114

INTRODUCTION

- We apply the Runge-Kutta method to a mathematical model of traffic flow. This report sheds light on how the fourth-order Runge-Kutta method is implemented to solve the Optimal Velocity Model. We identify initial conditions and base cases to run simulations of the model. We consider one-car and two-car systems to validate the application of the fourth-order Runge-Kutta method and the Optimal Velocity Model. Our simulations accurately capture practical traffic scenarios. Numerical analysis is an important aspect of applied mathematics. Some equations can be solved using exact techniques, but other more complicated ones need to be solved using numerical estimation techniques.
- Both Runge Kutta and Euler’s methods are used to solve a differential equation with initial values by approximating points on the solution graph in a specific interval. A limitation of these numerical analysis techniques is the farther away from the initial value point the farther the approximation is from the actual value. Euler's method has a greater error than Runge-Kutta, but computations for Runge-Kutta are fairly complex. Euler's method has a greater error than the Runge Kutta method, but is useful as a stepping stone into more complex techniques.

ABSTRACT

- This report sheds light on how the fourth-order Runge-Kutta method is implemented to solve the Optimal Velocity Model (Kurata & Nagatani, 2003). We identify initial conditions and base cases to run simulations of the model. We consider one-car and two-car systems to validate the application of the fourth-order Runge-Kutta method and the Optimal Velocity Model. Our simulations accurately capture practical scenarios.

The Model

- We describe the movement of car i using the following equation (Kurata & Nagatani,2003):

$$\frac{d^2 x_i(t)}{dt^2} = a\left(V(\Delta x_i(t)) - \frac{dx_i(t)}{dt}\right).$$

- The position of car i at time t is xi(t). We dene xi(t) to be the distance between car i and the car in front of it at time t.

$$\Delta x_i(t) = x_{i+1}(t) - x_i(t).$$

referred to as the the headway of car i at time t.

- The constant a is the sensitivity parameter. It describe how sensitive the average driver is to the motion of the car in front of him. The sensitivity parameter is actually the inverse of the time lag between a front car changing speeds and the car behind it reacting and then adjusting to the front car.
- The velocity of the car i is given by $\frac{dx_i(t)}{dt}$
- The function V (xi(t)) is the optimal velocity of the car.

$$V(\Delta x_i(t)) = \frac{v_{max}}{2}[\tanh(\Delta x_i(t) - x_c) + \tanh(x_c)].$$

- When the optimal velocity in Equation above is greater than the current velocity of the car, the acceleration is positive.
- When the optimal velocity is equal to the current velocity then the accelerationis zero. Thus the car's speed stays constant.

- When the current velocity is greater than the optimal velocity the acceleration is negative and the car will slow down.
- The acceleration has a proportionality parameter a. Since the sensitivity parameter is the inverse of the time lag it takes for a driver to react to a car in front of it.
- When implemented the fourth-order Runge-Kutta method on the model, let the parameters be the following:

$$a=1$$

$$Vmax=4$$

$$xc = 4$$

- Runge-Kutta method was ran with a step size of .001 over the t interval [0; 6].

Implementation of Runge-Kutta on the Model

- The Optimal Velocity Model is a system of second-order differential equations. We need to change each second-order differential equation into two first-order differential equations.

$$\frac{dx_i(t)}{dt} = y_i(t) \text{ and } \frac{dy_i(t)}{dt} = a(V(x_{i+1}(t) - x_i(t)) - y_i(t)).$$

- Each car i has two differential equations associated with it. Thus when there are M cars there are 2M equations in our system.
- Issues with applying this:

→In this model, we require a starting position and velocity of each car in the system.

→The equation relies on an (M +1)st car which does not exist.

Simulation with a Stopped Object and One Car

Throughout the simulation we use

$$x_2(t) = 20.$$

Thus, the system is

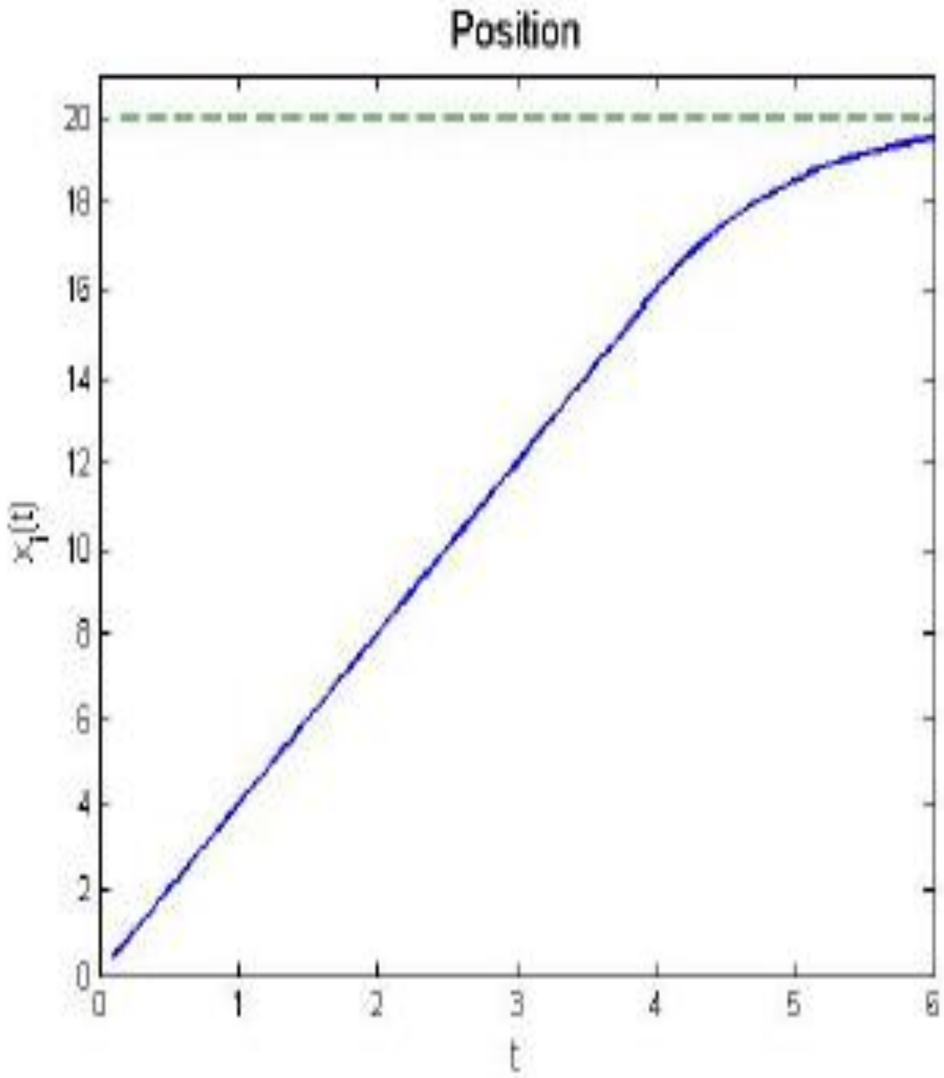
$$\frac{dx_1(t)}{dt} = y_1(t), \quad \frac{dy_1(t)}{dt} = a(V(20 - x_1(t)) - y_1(t)).$$

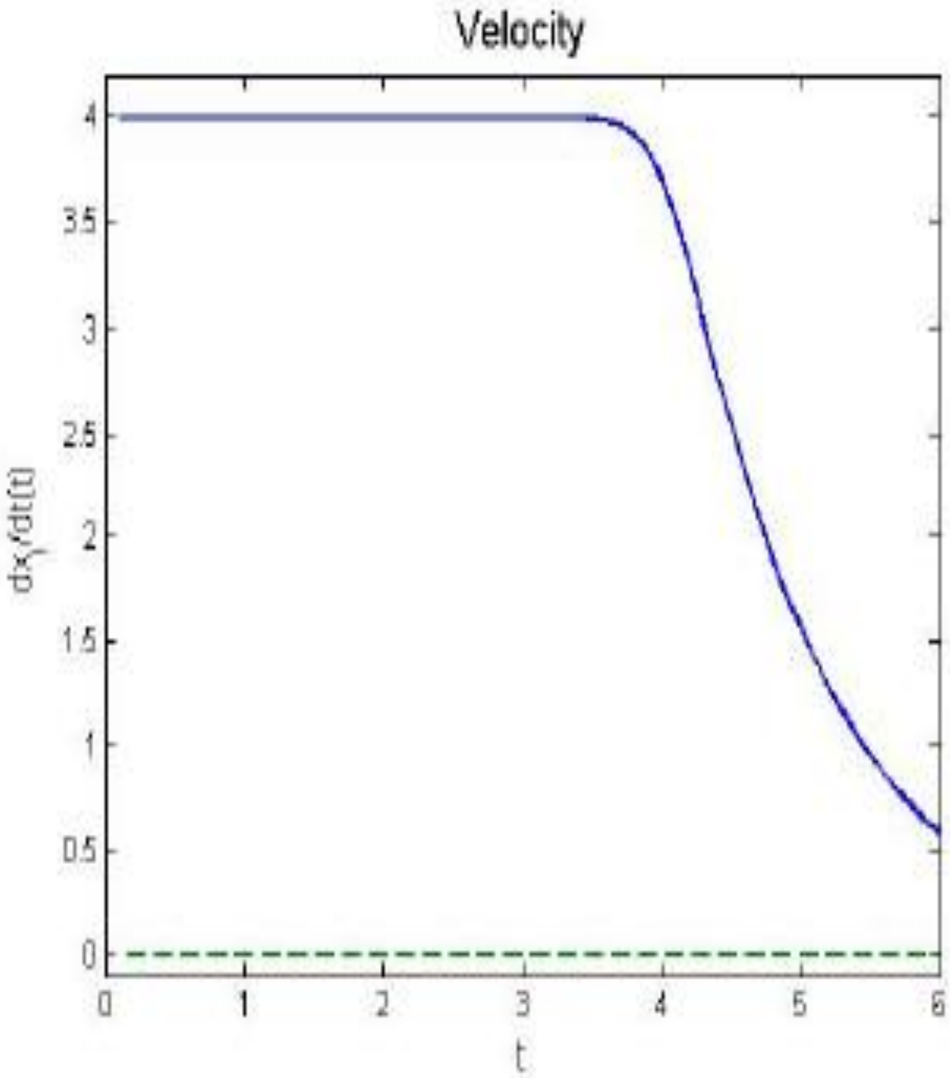
The car begins at position

$$x_1(0) = 0.$$

The velocity of the car is initially set to be the optimal velocity as follows:

$$\left.\frac{dx_1}{dt}\right|_{t=0} = V(\Delta x_1(0)),$$


Position
$x_1(t)$
t
0 2 4 6 8 10 12 14 16 18 20
0 1 2 3 4 5 6

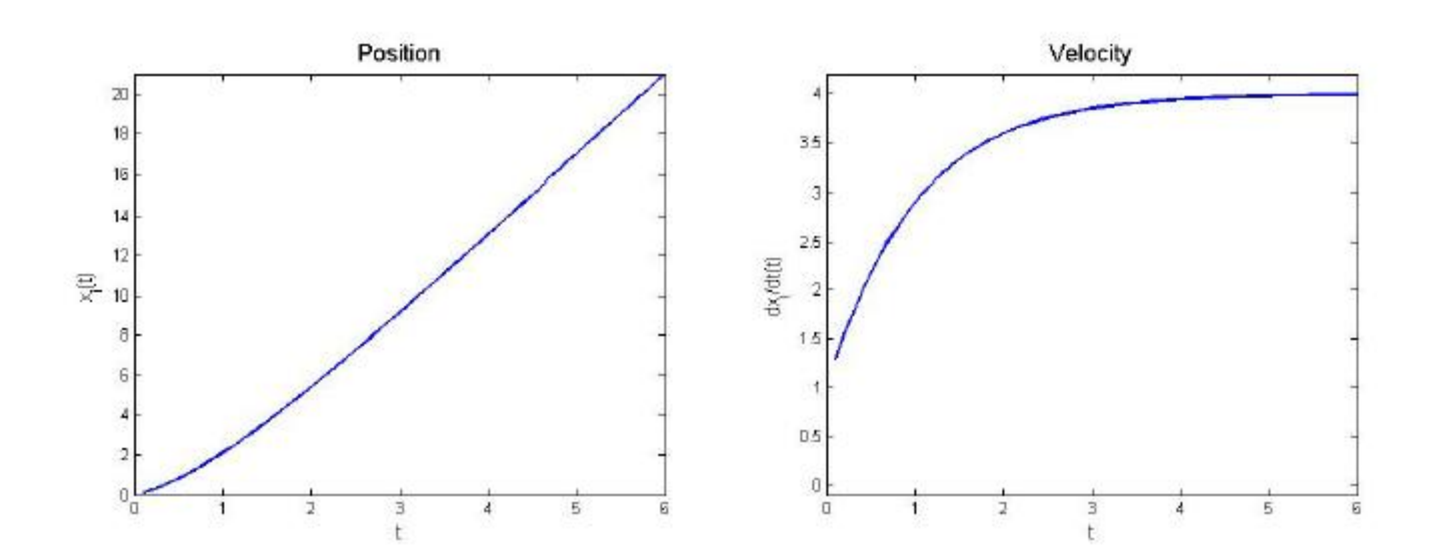

Velocity
$dx_1/dt(t)$
t
0 0.5 1 1.5 2 2.5 3 3.5 4
0 1 2 3 4 5 6

Simulation with a Clear Road in Front of the Car

- Just as in the stopped car simulation we think of x2(t) as an object stopped at the far away position of 100. Throughout the simulation we use x2(t) = 100. Thus, the system is

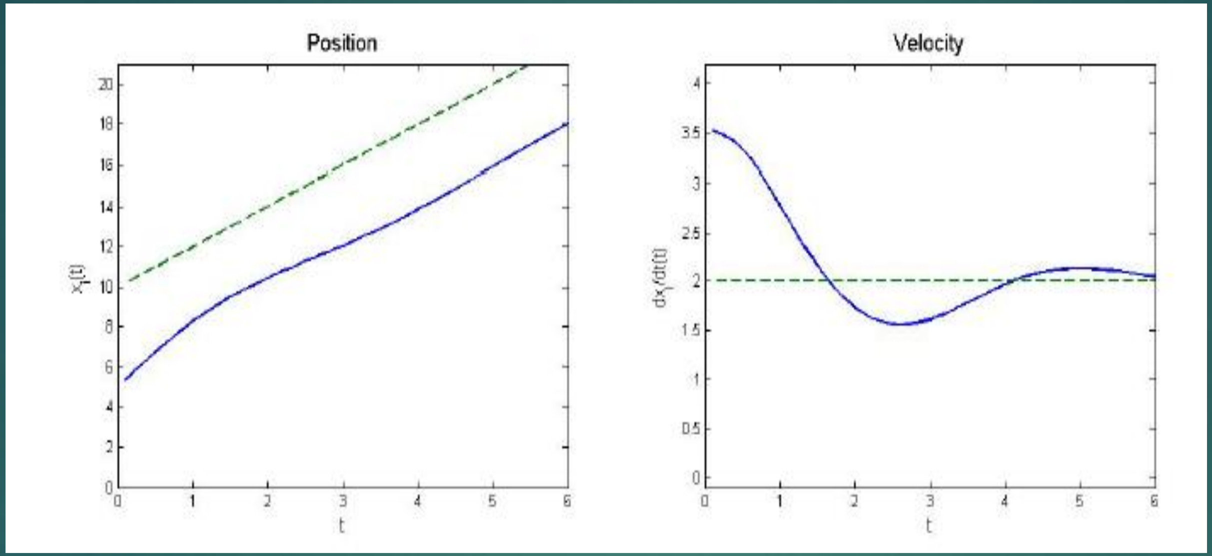
$$\frac{dx_1(t)}{dt} = y_1(t), \quad \frac{dy_1(t)}{dt} = a(V(100 - x_1(t)) - y_1(t)).$$

- The car begins at position x1(0) = 0:
- The velocity of the car is initially set to be the slow velocity of 1.

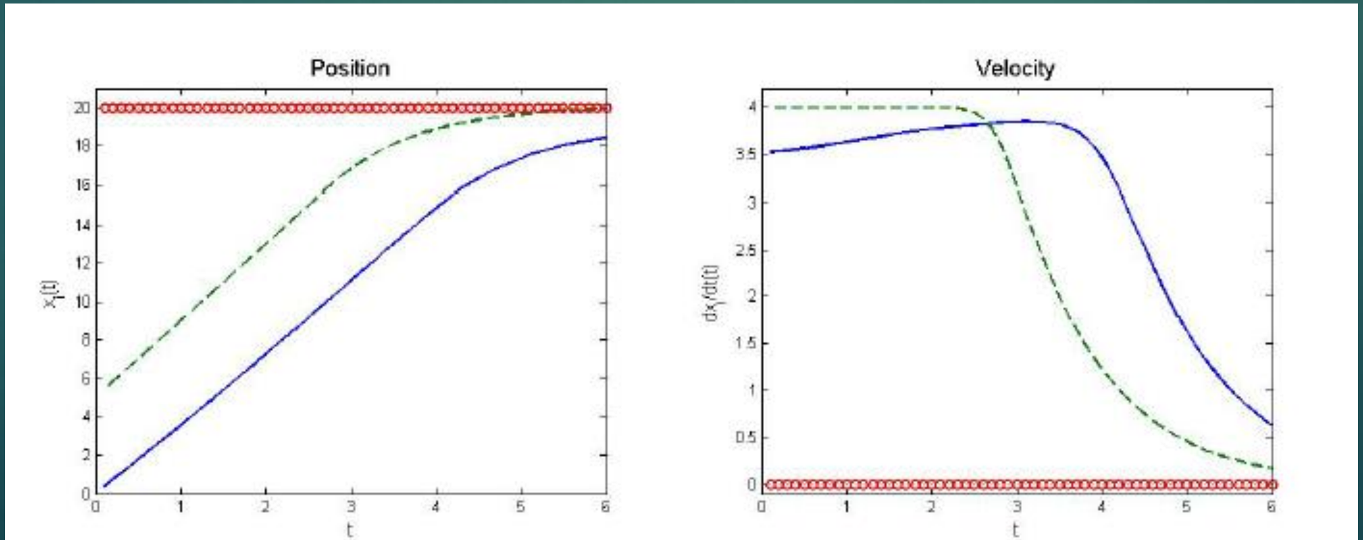


What other simulations can be done

- Simulation with an Object Traveling with a Slow constant Speed:



- Simulation with a Stopped Object and Two Cars:



Conclusion:

- Application of this model is in transportation engineering.
- This simple two car or one car system can be extended to a multi car system as well with 2N equations for a N car system.
- This model can be extended to real life multi lane traffic models as well.
- The given system can be extended to a real time automated traffic control system, where each simulation discussed will be interconnected with each other.
- For example, in highway the traffic is less and headway distance is more. Therefore the maximum velocity for vehicle can be more. But the same is not in highly crowded places like in the center of the city.
- With the help of this model we can reduce many accidents.