Simulating a Water Droplet Falling Through Water Vapor Using Numerical Methods

Matthew N. Leger

Mount Allison University

Simulating a Water Droplet Falling Through Water Vapor Using Numerical Methods

# Introduction

The model being simulated describes the situation of a microscopic water droplet falling through a cloud of water vapor. The water droplet starts at rest, meaning it has no initial velocity nor acceleration. The radius of the water droplet is assumed to initially be some small value, denoted as a. As the water droplet falls through this cloud, it will be grazing other droplets of water. This will cause the surrounding droplets of water to join the initial droplet, making the volume of the original droplet grow. We assume that the water droplet joins with other droplets at a rate of pi(r^2)(v)(k). The r value in this rate of change is the radius of the droplet, the v value is the velocity of the droplet, and the k value is the rate of change of the droplet’s radius with respect to time, which is assumed to be constant. The goal of this simulation is to determine the position velocity, and acceleration of the droplet of water (as a function of time) as it falls through the cloud. Afterwards, we also calculate the rate in which the droplet grows.

# Theory Behind the Model

To determine the position of the water droplet, we must know the rate of change of position, that being, we must know the velocity of the water droplet. After consultation with Dr. Lovekin (November 1st, 2021), she revealed that the rate of change of position with respect to time is equivalent to the force of gravity. This can be represented by the equation dp/dt = dm/dt(v) + m(dv/dt), where dm/dt represents the rate of change of mass, m represents the current mass of the droplet, and dv/dt represents the acceleration of the droplet. Since we are not accounting for wind resistance, we can assume that the acceleration of the droplet is constant, so we assume that it is equal to 9.81 meters per second squared. As mentioned, the rate of change of the radius is constant, so in this simulation we assume it is equal to 1/16. Additionally, we are told that the radius of the droplet is some finite but small radius a. In this simulation, we assume that this small radius is equal to 0.00001. We now have three differential equations that all play a factor in the falling of the water droplet. Because of this, the numerical techniques used in the simulation is a technique used to solve a system of differential equations. To accomplish this, the technique uses the Fourth Order Runge-Kutta technique to solve each differential equation. After the differential equations have been solved, we will have found the position, the mass, and the radius of the droplet all represented as functions of time. To calculate these values, we must provide a limit of integration for the system of differential equations. We will use an interval of integration from zero to 10. Using the solutions we found for position, we can then plug these values back into our velocity formula to determine the velocity of the droplet as a function of time. Finally, since the acceleration was assumed to be constant, we can simply assume that the acceleration of the droplet will always be 9.81 meters per second squared regardless of time. To determine the rate of change of the size of the droplet, we need to assume that the droplet is spherical in shape. This is because the volume of a droplet of water (measured in milliliters) is equivalent to its weight in grams (Helmenstine, 2019). Since we have the values of position, mass, and radius at a given time, we can plug in our values into the rate of change of mass and then convert the values of this rate of change into volume, giving us our rate of change of volume for the water droplet.

# Description of the Algorithm

As mentioned, the simulation uses the Fourth Order Runge-Kutta technique to solve the system of differential equations. This algorithm has six inputs. Its first two inputs, a and b, denote the endpoints of integration for the differential equations. For example, if a differential equation is the change in a variable x with respect to time (measured in seconds), and the endpoints of integration are zero and 10, then that represents the value of x after 10 seconds. The next input of the algorithm is m. This represents the number of equations that this system solves. The input N is the number of mesh points we will use. Mesh points are time intervals, meaning they represent the value of a function after a certain number of time steps. For example, if one integrates a function from zero to 25 with respect to time, uses five mesh points, and each time step has size five seconds, then the values of the function will be calculated at time zero, time five, time 10, time 15, time 20, and time 25. The next algorithm input is alphas, which is a list of initial conditions. In our case, the initial conditions are that the velocity is initially zero, the rate of change of the radius is 1/16 (since dr/dt is constant), and the rate of change of mass is initially pi (4/3) (0.00001^3). For the initial condition regarding the rate of change of mass, we form our initial condition using the fact that the mass in grams is equivalent to the volume of the droplet in millimeters (Helmenstine, 2019). Since the volume of a sphere is (4/3)(pi)(r^3), and we assumed that our initial radius is 0.00001, our initial mass is pi (4/3) (0.00001^3). The final input of the algorithm is an array of functions. These functions will be the equations in which we want to solve. With the inputs covered, the first step of the algorithm is to calculate the size of our time step. This will allow us to determine how far of a gap there is between the mesh points. Next, we initialize five arrays: t, w, k1, k2, k3, and k4. The t array holds an array of time values. The w array is a two-dimensional array that holds the respective solutions to the equations in which the algorithm received. For example, if the algorithm received the functions in the order dp/dt, dm/dt, and dr/dt, then w will hold the solutions of these equations with the solutions of dp/dt first, followed by the solutions of dm/dt, and finishing with the solutions of dr/dt. We set the initial solutions to be the initial conditions of the equations. Next, for each equation, we calculate the values of k1, k2, k3, and k4. According to Fourth Order Runge Kutta (n.d.), the k values of the Fourth Order Runge-Kutta method represent the slopes at different intervals. The k1 value represents the slope at the beginning of the time step (Fourth Order Runge Kutta, n.d.). Using the slope of k1, we can calculate k2 as the slope at the halfway mark of the time step using the slope from k1 (Fourth Order Runge Kutta, n.d.). Likewise, if we start at the beginning of the time step and follow the slope of k2, we can find k3 (Fourth Order Runge Kutta, n.d.). Finally, if we use slope k3 to go through the entire time step, we can calculate the slope k4 (Fourth Order Runge Kutta, n.d.). Thus, the algorithm calculates k1, k2, k3, and k4 values for each differential equation. It then uses these k values to estimate the solution of the differential equation at a given time. After it has done this for each time step, the algorithm returns our time array t and our solution array w.

With the Fourth Order Runge Kutta algorithm completed, we now have the position, the mass, and the radius of the water droplet all as functions of time. Since we know the formula for the rate of change of position (which is the velocity of the water droplet), we can then plug in our position values into dp/dt to determine the velocity of the droplet as a function of time. Since we assumed that the acceleration of the droplet is constant, no calculation is required. We thus have all the values needed to output the position, velocity, and acceleration of the droplet into a table.

To determine the rate of change of size, we once again use the fact that the mass of the droplet can be converted to the size of the droplet (Helmenstine, 2019). Using a similar technique as the one used to find the velocity of the droplet, we can input our solutions for mass into dm/dt to determine the rate of change of mass with respect to time. From there, it is a simple matter of converting the measurement of grams (since dm/dt is measured in grams per second) into milliliters.

# Verification of the Program

Initially, dr/dt was set to one. This was to test to see if the Fourth Order Runge Kutta function was working properly. Since dr/dt = 1 is a constant, the expected output over an interval of ten (with a time step of one) would be one, two, three, four, five, six, seven, eight, nine, and 10 respectively. This test succeeded, implying that the fourth order Runge-Kutta function works properly. Initially, the inputted equations yielded solutions equating roughly to 5.5 times 10^6 distance travelled in only 10 seconds. Intuitively, it seems highly unplausible for a water droplet to travel that far in the span of 10 seconds. After refactoring the equations, the numbers seemed much more plausible.

**Results**

Note that the position column represents total distance travelled at time t, not the altitude of the droplet at time t.

**Tabular Data of Position, Velocity, and Acceleration as Functions of Time**

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Position | Velocity | Acceleration |
| 0 | 0.00000 | 0.00000 | 9.81000 |
| 1 | 0.10457 | 0.39982 | 9.81000 |
| 2 | 1.47593 | 3.17051 | 9.81000 |
| 3 | 8.44703 | 12.62140 | 9.81000 |
| 4 | 30.79617 | 35.29872 | 9.81000 |
| 5 | 85.95579 | 79.96173 | 9.81000 |
| 6 | 201.22059 | 157.58081 | 9.81000 |
| 7 | 415.95961 | 281.33875 | 9.81000 |
| 8 | 783.82965 | 466.63158 | 9.81000 |
| 9 | 1,374.98929 | 731.06907 | 9.81000 |
| 10 | 2,278.31309 | 1,094.47496 | 9.81000 |

**Tabular Data of Volume as a Function of Time**

|  |  |
| --- | --- |
| Time | Volume |
| 0 | 0.00000 |
| 1 | 0.03010 |
| 2 | 0.12852 |
| 3 | 0.34447 |
| 4 | 0.72339 |
| 5 | 1.31046 |
| 6 | 2.15081 |
| 7 | 3.28956 |
| 8 | 4.77185 |
| 9 | 6.64282 |
| 10 | 8.94759 |

# Analysis of the Results

The tabulated data shows that in the span of 10 seconds, the water droplet travels over two kilometers as the droplet is falling through the cloud of water vapor. It reaches a top speed of over a kilometer per second. For the volume of the sphere, this represents that the droplet reaches its top rate of growth at the last time step, with that being almost 9 milliliters of water gained per second. A realization that can be made from this data is the relevance of wind resistance in the physical world. With wind resistance omitted (like in this simulation), the water droplet reaches surprisingly fast speeds. If the simulation were to be redone taking wind resistance into account, it would be reasonable to infer that the water droplet’s total distance travelled, velocity, acceleration, and rate of change of volume would decrease.

References

Fourth Order Runge Kutta. (n.d.). Retrieved from https://lpsa.swarthmore.edu/NumInt/NumIntFourth.html

Helmenstine, A., M., (2019). Calculating the Number of Atoms and Molecules in a Drop of Water. Retrieved from https://www.thoughtco.com/atoms-in-a-drop-of-water- 609425