

Steffan Wilkins - Assessment Document

by Steffan Wilkins

Submission date: 18-Nov-2022 08:59PM (UTC+0000)

Submission ID: 189498378

File name: 32671_Steffan_Wilkins_Steffan_Wilkins_-_Assessment_Document_467669_513447723.odt (51.56K)

Word count: 872

Character count: 4856

Games Architecture and Mechanics –

Assessment 1: Game Physics

Steffan Wilkins: 2101242

Part 1

Euler's numerical integration is a method to find out a point on the graph when given a different point on the graph. Usually, this is done when the line cannot be expressed in an equation. The way Euler's integration does this is by using the current value of y plus the small change in x multiplied by the current gradient. This is shown in the equation $Y_n = Y_{n-1} + \Delta X (dY/dX)$ [1]. The main advantage of Euler's method is that it is very quick to do, in cases where a computer needs to update in real-time, Euler's method is very crucial. The downside of using Euler's method is that it is less accurate than other methods where when the computer cannot calculate quickly enough the updated value is larger than expected, this is done by when the computer calculates there are fewer updates during its calculations Euler's method will increase the value more than it would realistically be making physical calculation more drastic than it should be. [2]

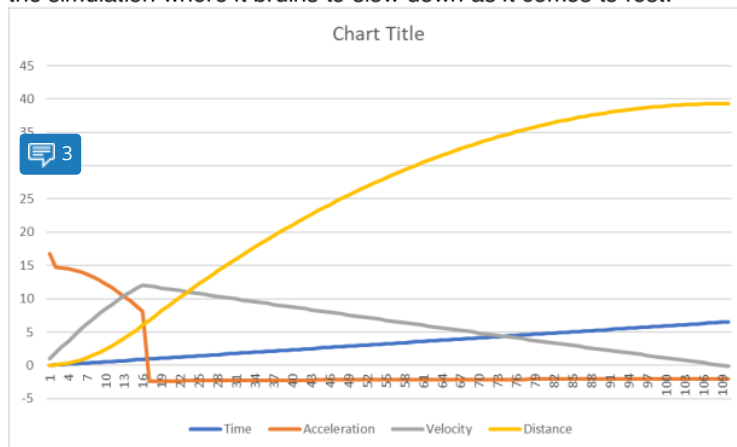
One different form of numerical integration used in gaming is the midpoint rule which you have to approximate the area "between the graph of $f(x)$ and the x -axis by summing the areas of rectangles with midpoints that are points on $f(x)$." [3] This method is used similarly to Euler's method where it is commonly used to calculate the movement of different objects, but unlike Euler's where it adds to the value, you're calculations with midpoint subtracts from the value but to a lesser degree. The advantage that the midpoint rule has over Euler's method is that the midpoint is much more accurate in its calculations. However, it is much less accurate when the speed changes rapidly.

Another numerical integration method is Verlet which is "used to calculate trajectories of particles in molecular dynamics simulations and computer graphics." [4] For Verlet integration to work, it needs the first two positions in the sequence, this is usually by using a different method to start the calculations which are its most significant disadvantage, it also is moderately accurate but for the purpose of calculating the movement of an object in a video game it should be accurate, enough although there are better options. The benefits of using Verlet over the other methods is that there is that even without velocity Verlet method can still find the next point in time, and Verlet is straightforward when compared to the other methods.

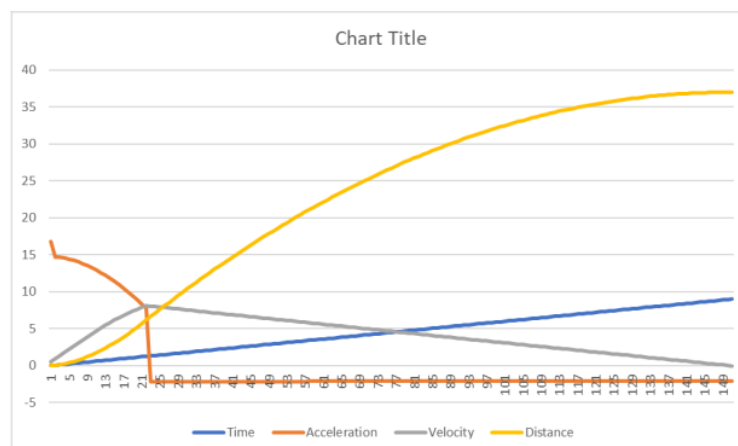
The final integration method commonly used in gaming is the RK4 method or Runge-Kutta, which works the same way as Euler's and midpoint methods but is the most complex where it uses the method of midpoint but expands on it making more calculations before moving the object making Runge-Kutta method the most precise out of these four. However, because of its precision, this method is much longer than the others, compared to Euler's method it is quadruple times as long at the same frequency [5], but because of its precision, it does not need to be updated as frequently as Euler's because of said precision.

Part 2

In my first model, I used Euler's method of integration to gather the velocity and the distance travelled at each time step. Here is the graph of my result of this simulation. As shown in the graph as time increases, the Acceleration starts at its highest point as it decreases until the acceleration is at a minus slowing the spring-powered model car. This is also shown in the velocity as at the start of the test the car speeds up reaching its maximum speed between 10 and 15 metres per second, after that the velocity gradually decreases at a constant rate until it makes a complete stop. Finally, the distance the car moves increases at the beginning of the simulation where it begins to slow down as it comes to rest.



For my second simulation of the spring-powered model car, I used the Verlet method of integration to gather my velocity and distance this time. When viewing this graph it's easy to see the similarity between both methods of numerical integration, however, the main two differences are that all of the values for the Verlet chart are less drastic than the Euler's method, such as the Distance has a slightly shorter, acceleration becomes zero at time step 21 unlike Euler's where it's at close to time step 16 other main difference is that for the Spring car to stop it only takes Euler's method about 6 seconds whereas Verlet takes around 9 seconds to finish.



Reference

- [1] "How to do Euler's Method?", *CalcWorkshop*, Dec. 31, 2019. [Online]. Available: <https://calcworkshop.com/first-order-differential-equations/eulers-method-table/#chapter> [Accessed Nov. 15, 2022]
- [2] Calvin, "Euler's Method – All you Need to Know About it!", *TechPatio*, Aug. 10, 2021. [Online]. Available: <https://techpatio.com/2021/articles/eulers-method-all-you-need-to-know-about-it> [Accessed Nov. 15, 2022]
- [3] "Numerical integration - Midpoint, Trapezoid, Simon's rule", *LibreTexts*, Jul. 25, 2021. [Online]. Available: [https://math.libretexts.org/Courses/Mount_Royal_University/MATH_2200%3A_Calculus_for_Scientists_II/2%3A_Techniques_of_Integration/2.5%3A_Numerical_Integration_-_Midpoint%2C_Trapezoid%2C_Simpson's_rule#:~:text=1%3A%20The%20midpoint%20rule%20approximates,points%20on%20f\(x\).](https://math.libretexts.org/Courses/Mount_Royal_University/MATH_2200%3A_Calculus_for_Scientists_II/2%3A_Techniques_of_Integration/2.5%3A_Numerical_Integration_-_Midpoint%2C_Trapezoid%2C_Simpson's_rule#:~:text=1%3A%20The%20midpoint%20rule%20approximates,points%20on%20f(x).) [Accessed Nov. 15, 2022.]
- [4] A. Hazra, "Making a Verlet Physics Engine in JavaScript", *Better Programming*, May. 16, 2019. [Online]. Available: <https://betterprogramming.pub/making-a-verlet-physics-engine-in-javascript-1dff066d7bc5> [Accessed Nov. 15, 2022.]
- [5] J. Rodriguez, "Math for Game Developers - Runge-Kutta Order 4", *Youtube*, Aug. 4, 2016. [Online] Available: <https://www.youtube.com/watch?v=hGCP6l2WisM> [Accessed Nov. 15, 2022.]

Steffan Wilkins - Assessment Document

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

69/100

PAGE 1



Comment 1

..but is applied to problems where the integral is evaluated at a given point.



Comment 2

reference please. Clarify.

PAGE 2



Comment 3

units please

PAGE 3

GRADING FORM: SMU GENERIC_GJ

STEFFAN WILKINS

68.50

RESEARCH-INFORMED LITERATURE



Try to use fewer online refs. You could have used Bishop and van Verth, for example or located the original Verlet paper. Your review of methods covers essential areas - some further detail would help here.

6.5

KNOWLEDGE AND UNDERSTANDING



You demonstrate a good understanding of the requisite coding and SE practice skills. Units were specified in your code -well done.

13.5

A more detailed description of Eulers' and Verlet would help -eg show equations and algorithm. Which kind of Verlet - why choose it?

ANALYSIS



Some essential observations and conclusions made - good. Try to explain why the curves were shaped as they were. Label axes/key with units used.

13.5

You report ends suddenly try to summarise and draw conclusions at the end of your report.

PRACTICAL APPLICATION AND DEP.



Code generated produces good results that have been graphed in your report. Well done. Good commenting. Try combining both programs into one with a console menu.

32

Some functions, eg "SumForces" poorly named.
Overall a good attempt.

SKILLS FOR PROFESSIONAL PRAC.



Try to avoid too many electronic references as mentioned earlier.

3

Report well structured. Good use of graphs - try to include a final conclusions section. Placing source code in appendix would help.