On the Numeric Techniques for Problems in Physics

Presentation by Patryk Kwoczak

What Is The Objective?

Given some physical laws describing a system, how can we simulate the true evolution through time of the system?

- Physical Laws described by Differential Equations

How can we examine the accuracy of our results?

- What is ‘good enough’ and how do we do better?

What do we need to be mindful of?

- What problems arise?

My Background

- Applied Math Major interested in computational math
- Took a Numerical Methods course which described numeric ordinary differential methods
- Some experience with Python examining complex root-finding algorithms and their dynamical structure.

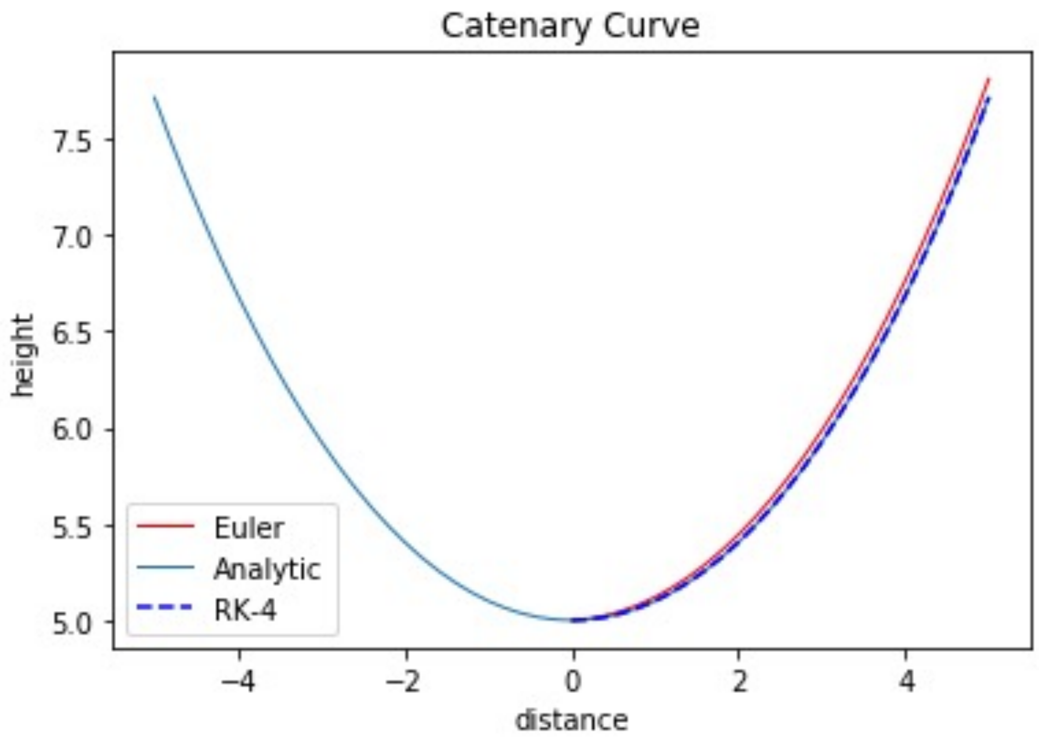
Learning Outcomes

- Learned to deal with how the methods are only as good as the limitations of the physical system they are modelling.
- Learned how to manipulate the mathematics to produce the right equations necessary.
- Learned about Symplectic Integrators which utilize the symmetry of the physical system to more accurately model it.

Techniques and Tools

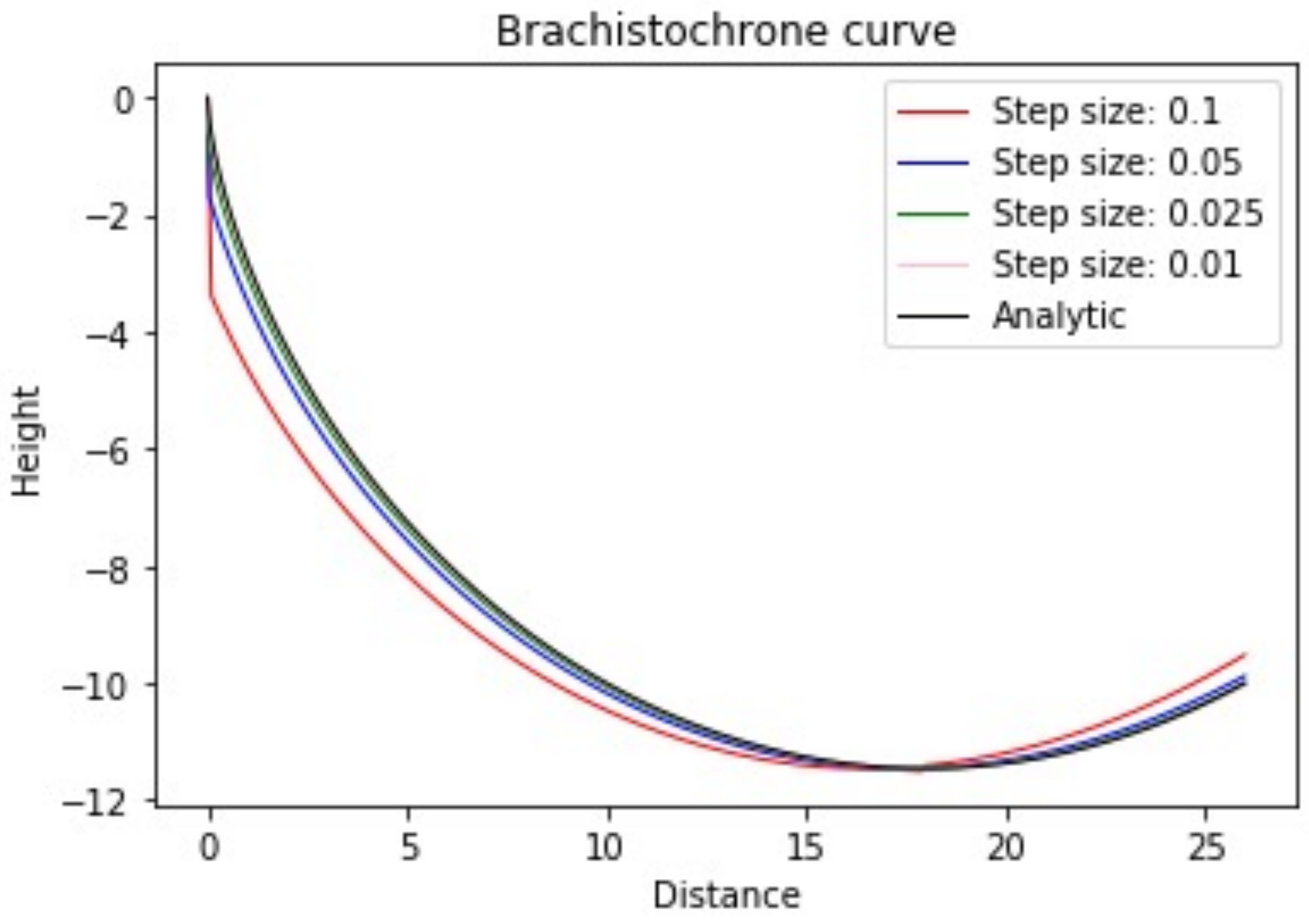
I took these tools and applied them to the physical situations they could be used in, such as the shape of a hanging rope, the path of fastest descent, and the Celestial Orbit of a planet.

- The Methods:
  - Euler's Method
  - Fourth-Order Runge-Kutta
  - Symplectic Euler
  - Verlet and Verlet-Velocity



Euler's Method

- $\left[\frac{dy}{dx}\right]^2 = \frac{k^2}{y} - 1$
- $\frac{dy}{dt} = f(y)$
- $y_{n+1} = y_n + f(y_n) * \Delta t$
- Problems:
  - Equation assumes y-axis increases downwards
  - Singularity at y=0
  - Different f for positive and negative slope
  - Must be real value



Runge-Kutta Fourth-Order (RK4)

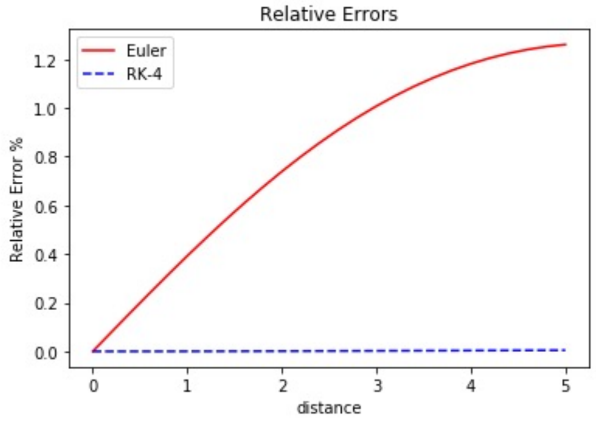
- given $\frac{dy}{dt} = f(t, y)$,

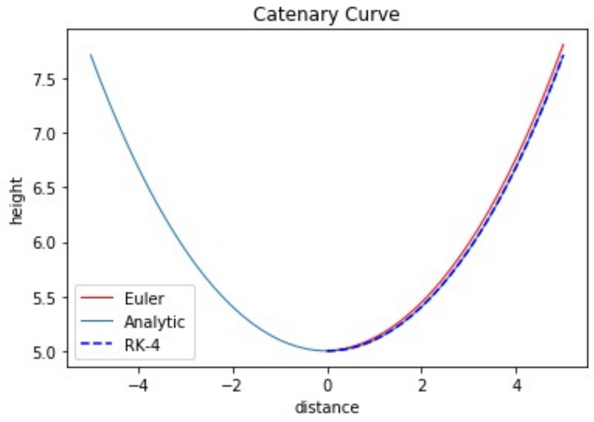
$$k_1 = f(t_n, y_n);\ k_2 = f\left(t_n + \frac{\Delta t}{2}, y_n + \Delta t * \frac{k_1}{2}\right);$$

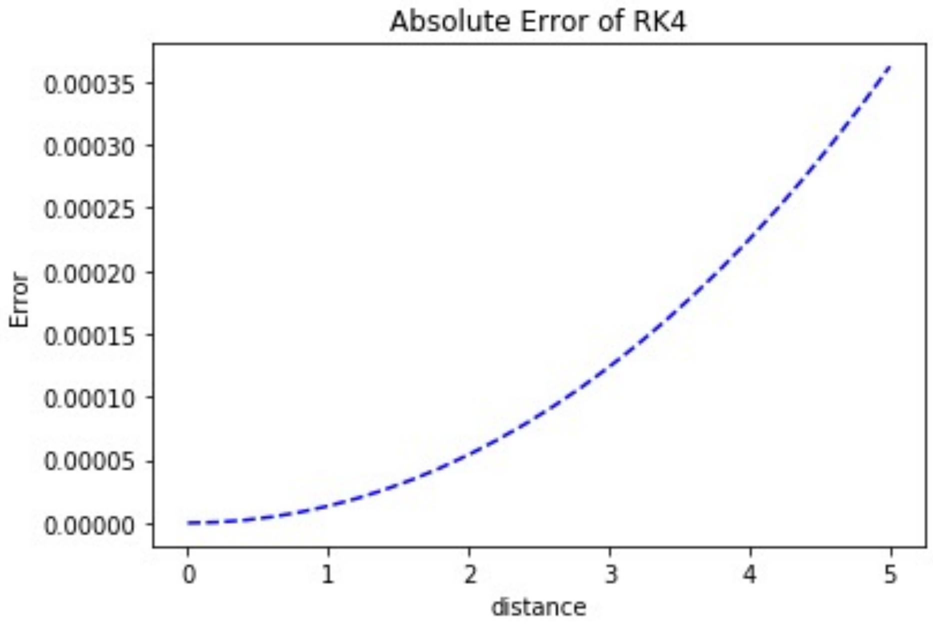
$$k_3 = f\left(t_n + \frac{\Delta t}{2}, y_n + \Delta t * \frac{k_2}{2}\right);\ k_3 = f(t_{n+1}, y_n + \Delta t * k_3)$$

$$y_{n+1} = y_n + \frac{\Delta t}{6} * (k_1 + 2k_2 + 2k_3 + k_4)$$

- Problem: Discrete values are obtained







Symplectic Integrators

Good for certain systems

- Must conserve a quantity independent of time
- Follow Hamiltonian
  - H = Total Energy
- Given by Equations of Motions
  - $\frac{d\boldsymbol{p}}{dt} = -\frac{\partial H}{\partial \boldsymbol{q}}, \frac{d\boldsymbol{q}}{dt} = +\frac{\partial H}{\partial \boldsymbol{p}}.$

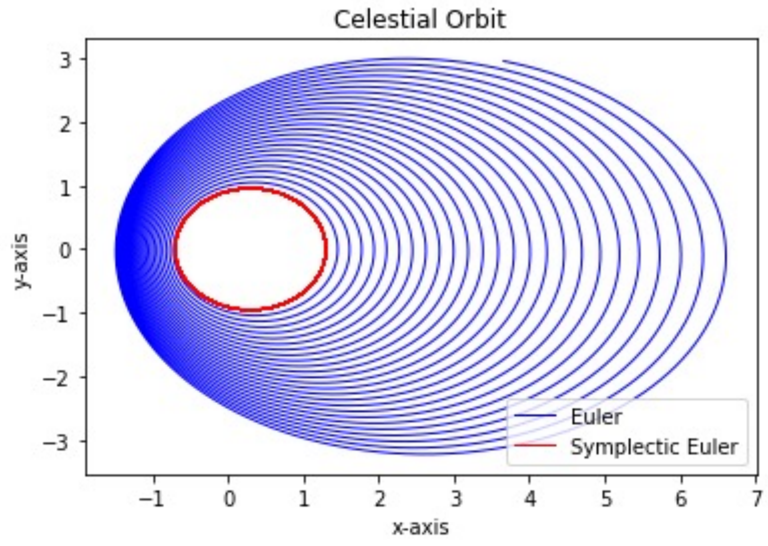
The Methods

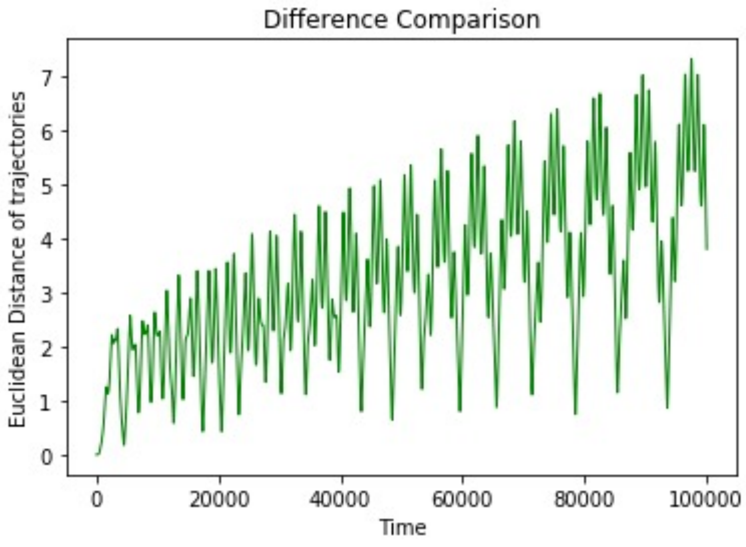
- Symplectic Euler
  - $y_{i+1} = y_i + \Delta t * v_i$
  - $v_{i+1} = v_i + \Delta t * f(y_{i+1})$
- Verlet-Velocity
  - Gives explicit velocity
  - $y(t + \Delta t) = y(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2$

    $v(t + \Delta t) = v(t) + \frac{a(t) + a(t + \Delta t)}{2}\Delta t$

Symplectic vs. Euler

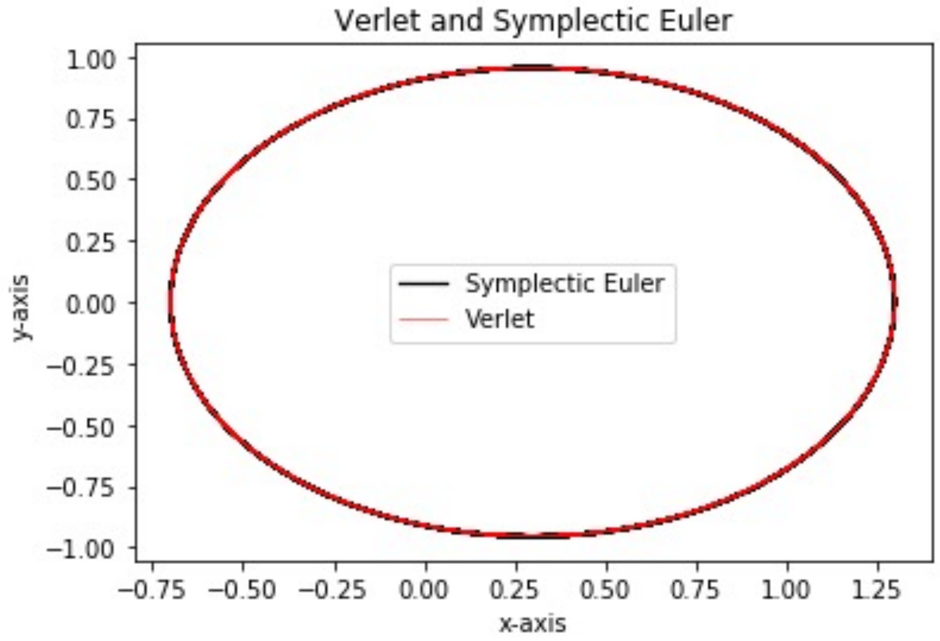
- Symplectic solves an approximate Hamiltonian Exactly
  - Preserves area of trajectory (Doesn't escape)
- Standard Euler solves exact system approximately
  - Error builds up and escapes correct orbit

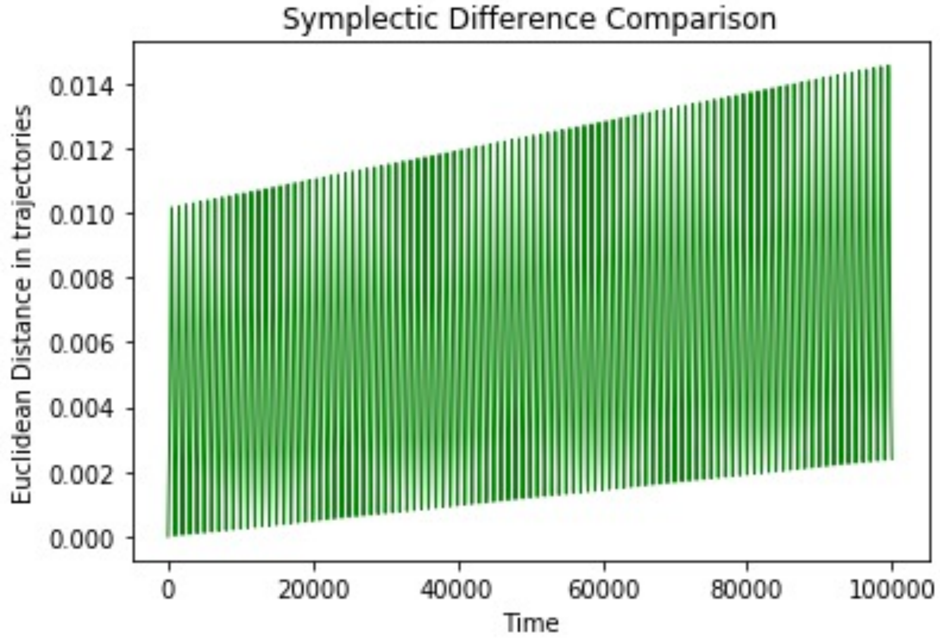




Symplectic Euler vs Varlet

Both Very similar





References

- 1: “Brachistochrone Problem.” *From Wolfram MathWorld*, mathworld.wolfram.com/BrachistochroneProblem.html.
- 2: “Catenary.” *From Wolfram MathWorld*, mathworld.wolfram.com/Catenary.html.
- 4: Chai, Patrick, and Evan Anzalone. “Numerical Integration Techniques in Orbital Mechanics Applications.” *Research Gate*, doi:10.13140/RG.2.1.1474.2243.
- 4:Rosenthal, Mike, “Numeric Integration of Orbits – A Crash Course” presentation, https://people.ucsc.edu/~mmrosent/talks_notes/num_int_pres.pdf

Thank you ☺