HASKELL SPACEFLIGHT WORKSHOP

Jonathan Merritt, Luke Clifton

YOW! LambdaJam 2019, May 13–15

———

Version timestamp: 2019-04-27T09:44:10+0000

Table of Contents

1

Introduction

In this workshop, we take an enthusiastic numerical approach to simulating spacecraft maneuvers. Our workshop examples appear in the published literature, yet we must begin by stating that there are often more elegant solutions to the problems we describe, allowing results to be found more economically and precisely. For these solutions, we must refer readers to a comprehensive textbook on astrodynamics (eg. [?]). We have chosen to take our somewhat simplified numerical approach for the following reasons:

1. It makes a 90 minute workshop possible, where an exploration of tailored methods would require weeks of work,

2. It allows us to frame several problems as initial value problems of ODE integration,

3. It introduces techniques that are widely applicable to dynamical systems outside of spaceflight, and

4. We believe it provides a good introduction, motivating more sophisticated approaches quite naturally as individuals explore further.

2

ODE Integration and Initial Value Problems

The models we use for a spacecraft depend upon a set of variables that represent its state at an instant in time. These state variables typically include:

- Position
- Velocity
- Mass

They may be scalar quantities or vectors, as appropriate to the problem.

Our simulations are all examples of "Initial Value Problems". In an initial value problem, we know the starting state of the spacecraft, and we have a set of ordinary differential equations (ODEs), which describe how its state evolves with time. We will integrate these ODEs to predict the state at future times. Using this approach, we can compute the time history of state variables that are critial to mission or maneuver planning. For example, we might find the trajectory of a spacecraft (its position as a function of time), and check whether it places the spacecraft in a desired orbit.

The motion of a spacecraft depends on multiple forces that might be acting on it. For example:

- Gravity
- Atmospheric drag
- Rocket thrust

Thrust from a rocket engine may be controlled (ie. its magnitude and direction may be commanded), and these control inputs can be modeled easily in our system. Testing the behavior of a control system, particularly under conditions of real-world variations, is a modern practical use of the methods we cover (eg. [? ?]).

2.1 Euler's Method

We can write a set of ODEs as:

$$\frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}} = f(t, \mathbf{x}) \tag{2.1}$$

Here, $\mathbf{x}$ is the state vector, t is time, and f is some function. In Euler's method, we approximate a step forward in time by multiplying the gradient by the size of the time step, h:

$$\mathbf{x}(t + h) \approx \mathbf{x}(t) + \dot{\mathbf{x}}\,h \tag{2.2}$$
$$\approx \mathbf{x}(t) + f\left(t, \mathbf{x}(t)\right)\,h \tag{2.3}$$

2.1.1 Radioactive Decay

We will begin implementing Euler's method, specialized to `Double`, using the process of radioactive decay as an example. Radioactive decay is a useful example because it is simple enough to solve analytically, thus providing a comparison with the numerical result, and it only involves a single state variable, which can be represented as a `Double`.

In radioactive decay, the rate of decay, $\dot{N}$, is proportional to the number of moles of radioactive particles that remain at any instant in time, N (N being the only state variable in this case):

$$\dot{N} = -\lambda N \tag{2.4}$$

where λ is called the decay constant. This equation can be solved by knowing in advance that an exponential function happens to fit exactly the expected equation:

$$N = N_0 \exp\left(-\lambda t\right) \tag{2.5}$$

So that:

$$\dot{N} = -\lambda\left(N_0 \exp\left(-\lambda t\right)\right) \tag{2.6}$$
$$= -\lambda N \tag{2.7}$$

as required. Conventionally, λ is specified in terms of the half-life of an isotope, $t_{\frac{1}{2}}$:

$$\text{at } t = t_{\frac{1}{2}}, \; N = \frac{N_0}{2} \tag{2.8}$$

thus:

$$\frac{N_0}{2} = N_0 \exp\left(-\lambda t_{\frac{1}{2}}\right) \tag{2.9}$$

$$\ln\left(\frac{1}{2}\right) = -\lambda t_{\frac{1}{2}} \tag{2.10}$$

$$\lambda = \frac{\ln 2}{t_{\frac{1}{2}}} \tag{2.11}$$

As an example, consider the isotope Plutonium-238 (^{238}Pu), which has been used in radioisotope thermoelectric generators (RTGs) for spacecraft (eg. Voyager 1 and 2). This isotope has a half-life of approximately $t_{\frac{1}{2}} = 2.77 \times 10^9$ s (87.7 years), so that $\lambda = 2.50 \times 10^{-10}$/s.

Problem 1: Euler integration specialized to `Double`.

In the file `ODE.hs`,
- implement `eulerStepDouble`, which takes a single step of Euler integration,
- implement `integrateEulerDouble`, which takes multiple steps.

We can observe how Euler integration behaves on the exponential decay problem with `plotEulerDouble` in the module `Examples.ODEExamples`.

Exponential Decay - Analytic vs Euler

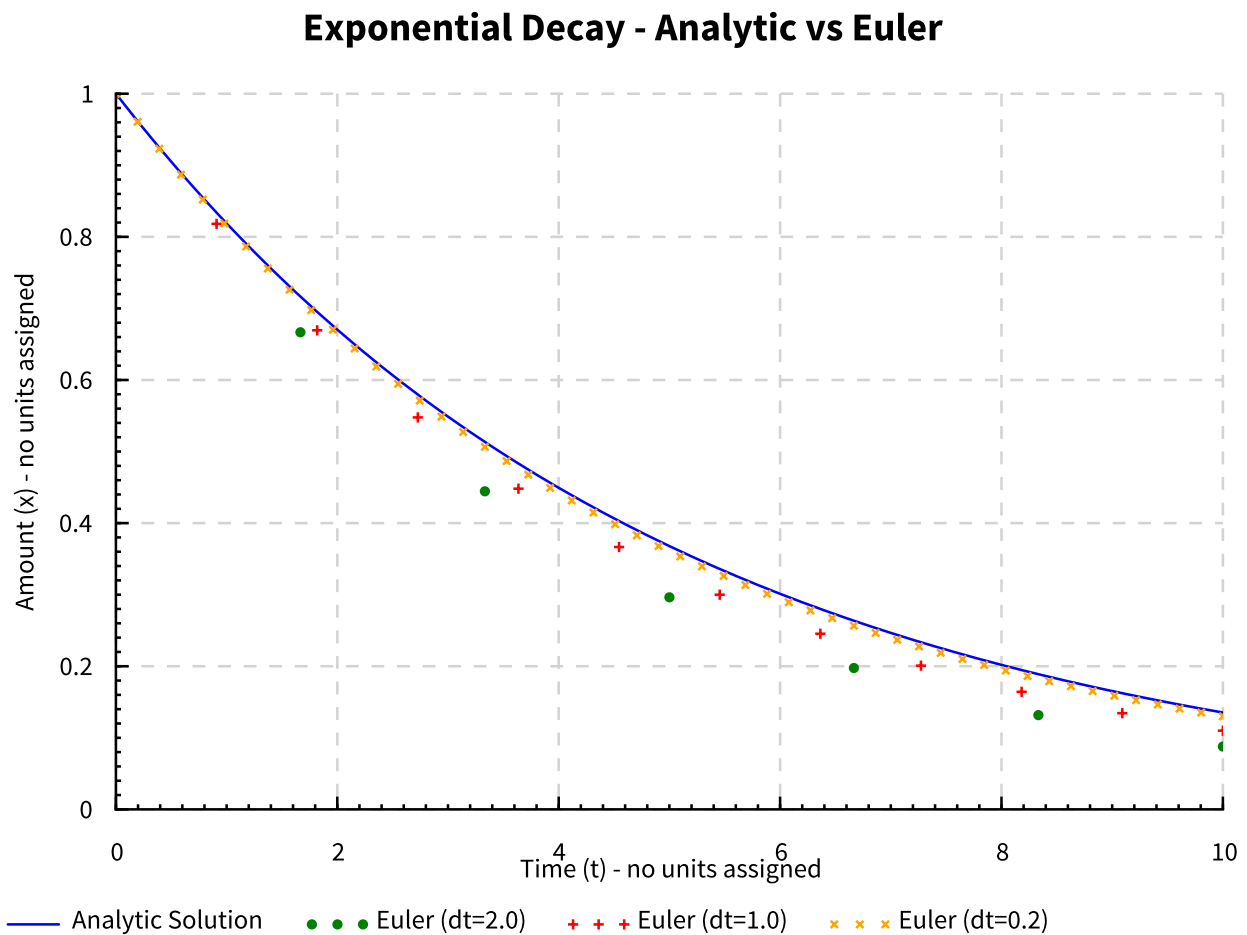**Figure 2.1:** Comparison of Euler integration with the analytic result for exponential decay of ^{238}Pu.

[type=symbols,style=long]