MA3227 Numerical Analysis II

Lecture 20: Trajectory of a Cannonball

Simon Etter

National University of Singapore

2019/2020

Trajectory of a Cannonball

Disclaimer

Like Lecture 19, the main purpose of this lecture is not to introduce any new material but rather to illustrate the application of the techniques that we developed so far to real-world problems.

Introduction

The last two lab sessions introduced the problem of finding $x : [0, T] \to \mathbb{R}^2$ such that

$$x(0) = 0, \quad \dot{x}(0) = v_0, \quad \ddot{x} = -D\,\|\dot{x}\|\,\dot{x} - g\,e_2. \qquad e_2 = \begin{pmatrix} 0 & 1 \end{pmatrix}^T$$

This equation models the trajectory of a cannonball subject to drag with drag coefficient $D > 0$ and gravity with gravitational constant $g > 0$.

Trajectory of a Cannonball

Conditional termination

In the lab sessions, we solved this ODE on a fixed time interval $[0, T]$. Now, we move to a more realistic setting where the goal is to compute the trajectory until the cannonball hits the ground.

Assuming a perfectly flat ground at $x_2 = 0$, the problem then becomes to find $x : [0, T] \to \mathbb{R}^2$ and $T > 0$ such that $x(t)$ satisfies the above ODE and additionally $x_2(T) = 0$.

This problem can be solved by combining ODE solvers with root finding algorithms as follows.

- Do the usual Runge-Kutta time-stepping as long as $x_2(t_k) > 0$.
- Once we find t_k such that $x_2(t_k) < 0$, go back to time t_{k-1} and use a root finder to determine Δt such that $x_2(t_{k-1} + \Delta t) = 0$.

Note that once we reach the root-finding part, we necessarily have a bracketing interval $[t_{k-1}, t_k]$ such that $x_2(t_{k-1}) > 0$ and $x_2(t_k) < 0$, so we can solve the root-finding problem reliably using the bisection method.

`example()` demonstrates how to run the above algorithm using the `DifferentialEquations` package.

Trajectory of a Cannonball

Shooting as far as possible

Let us next consider the following problem:

Given physical constants D, g and an initial velocity $\|v_0\|_2 = 1$, determine the angle θ in which to fire the cannon such that the cannonball flies as far as possible.

In mathematical terms, the aim is thus to find the θ maximising $x_1(T(\theta), \theta)$, where $x : [0, T] \to \mathbb{R}^2$ and $T > 0$ satisfy

$$x(0) = 0, \quad \dot{x}(0) = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}, \quad \ddot{x} = -D\,\|\dot{x}\|\,\dot{x} - g\,e_2, \quad x_2(T) = 0.$$

A necessary condition for θ to be optimal in the above sense is

$$\tfrac{d}{d\theta}\,x_1(T(\theta), \theta) = 0;$$

hence the optimisation problem can be solved using a root-finding algorithm once we have a procedure for computing the above derivative.

Trajectory of a Cannonball

Shooting as far as possible (continued)

Application of the chain rule yields

$$\frac{d}{d\theta}\, x_1(T(\theta), \theta) = \frac{\partial x_1}{\partial \theta}(T(\theta), \theta) + \dot{x}_1(T(\theta), \theta)\, \frac{\partial T}{\partial \theta}(\theta).$$

Our goal is therefore to determine $\frac{\partial x_1}{\partial \theta}(T)$ and $\frac{\partial T}{\partial \theta}$.

Note the difference between the total derivative $\frac{d}{d\theta}\, x_1(T(\theta), \theta)$ and partial derivative $\frac{\partial x_1}{\partial \theta}(T(\theta), \theta)$. The former says "differentiate everything with respect to θ", while the latter says "evaluate the derivative of x_1 with respect to θ at time $t = T$, but ignore the fact that T also depends on θ".

Regarding $\frac{\partial x_1}{\partial \theta}(T)$, we observe that if $\dot{y} = f(y)$, then we have

$$\frac{\partial}{\partial t}\frac{\partial y}{\partial \theta} = \frac{\partial}{\partial \theta}\dot{y} = \frac{\partial}{\partial \theta} f(y) = \nabla f(y)\, \frac{\partial y}{\partial \theta}.$$

This is a linear ODE for $\frac{\partial y}{\partial \theta}$ which we can combine with $\dot{y} = f(y)$ into a system of two coupled ODEs for y and $\frac{\partial y}{\partial \theta}$.

Formulating this coupled ODE for $\frac{\partial x_1}{\partial \theta}(T)$ in our cannonball example is quite cumbersome because it would require us to compute $\nabla f(y)$. Luckily, there are tricks how we can effectively have the computer generate $\nabla f(y)$ for us. I will return to this later.

Trajectory of a Cannonball

Shooting as far as possible (continued)

An expression for $\frac{\partial T}{\partial \theta}$ can be obtained by differentiating $x_2(T(\theta), \theta) = 0$ with respect to θ, which yields

$$0 = \tfrac{d}{d\theta}\, x_2(T(\theta), \theta) = \tfrac{\partial x_2}{\partial \theta}(T(\theta), \theta) + \dot{x}_2(T(\theta), \theta)\, \tfrac{\partial T}{\partial \theta}(\theta)$$

and therefore

$$\tfrac{\partial T}{\partial \theta} = -\tfrac{\partial x_2}{\partial \theta}(T) \,/\, \dot{x}_2(T).$$

$\frac{\partial x_2}{\partial \theta}(T)$ can be computed in the same way as $\frac{\partial x_1}{\partial \theta}(T)$ and in fact has to be computed anyway since it forms part of the system of ODEs for $\frac{\partial x_1}{\partial \theta}(T)$.

Putting everything together, we conclude that our aim is to find θ such that

$$\tfrac{d}{d\theta}\, x_1(T) = \tfrac{\partial x_1}{\partial \theta}(T) - \tfrac{\dot{x}_1(T)}{\dot{x}_2(T)}\, \tfrac{\partial x_2}{\partial \theta}(T) = 0.$$

Trajectory of a Cannonball

Discussion

The above shows that the "shooting as far as possible" problem is solved once we have an algorithm for computing $\frac{\partial x}{\partial \theta}(T)$.

We have further seen that $\frac{\partial x}{\partial \theta}(T)$ could in principle be computed by solving a system of coupled ODEs, but formulating these ODEs is quite cumbersome.

Tedious calculations by hand can be avoided by using dual numbers.
See next slide.

Trajectory of a Cannonball

Dual numbers (not examinable)

A dual number $x + y\varepsilon$ represents a point $x \in \mathbb{R}$ and an "infinitesimally small" perturbation $y \in \mathbb{R}$. The ε in $x + y\varepsilon$ is to be interpreted as a special constant similar to the imaginary unit $\iota = \sqrt{-1}$.

Dual numbers become very useful once we define

$$f(x + y\varepsilon) = f(x) + f'(x)\, y\varepsilon. \tag{1}$$

since this allows us to compute $f'(x)$ as the infinitesimal part of $f(x + \varepsilon)$.
Note that (1) mimics a first-order Taylor expansion.

As the simplest non-trivial example, consider

$$f\big(g(x + \varepsilon)\big) = f\big(g(x) + g'(x)\,\varepsilon\big) = f\big(g(x)\big) + f'\big(g(x)\big)\, g'(x)\,\varepsilon,$$

which is exactly the chain rule $\frac{d}{dx} f\big(g(x)\big) = f'\big(g(x)\big)\, g'(x)$.

Thus, dual numbers allow us to compute derivatives of an arbitrarily complicated piece of code simply by overloading a few basic operations like $+, -, \times, \div, \sqrt{\cdot}$, etc. such that these operations perform (1) when applied to dual numbers.

Trajectory of a Cannonball

Discussion

See `optimal_angle()` for code solving the “shoot as far as possible” problem using dual numbers, and `example2()` for a plot of the result.

Note that we have just solved a non-trivial, real-world problem with < 100 lines of code by combining three distinct Julia packages (`DifferentialEquations`, `Roots`, `ForwardDiff`).

This type of composability is frequently reported as one of the main strengths of Julia. Here is another example demonstrating how easy it is to combine the functionality of different Julia packages:

https://tutorials.juliadiffeq.org/html/type_handling/02-uncertainties.html.

Summary

- $\frac{d}{dt}\frac{\partial y}{\partial p} = \nabla f(y)\,\frac{\partial y}{\partial p}$ trick for computing derivatives of $y(t)$ with respect to a parameter p.