Introduction to computational programming

Chapter 3 Exercise

Direction fields and Trajectories

David M. Rosenberg
University of Chicago
Committee on Neurobiology

Version control information:
Last changed date: 2009-10-19 23:49:54 -0500 (Mon, 19 Oct 2009)
Last changes revision:144
Version: Revision 144
Last changed by: David M. Rosenberg

October 20, 2009

Part I

Tutorial

Overview

In this exercise we will generate *vector direction fields* as a means of exploring the large scale topology of an ODE. TODO: MORE INTRO HERE

Vector directional fields

The algorithm provided in your text (**??**, page **??**) is copied below. Both the verbatim textual description and the standard pseudocode representation are provided.

Pseudo code for plotting a direction field:

1. define a function $f(x,t)$ that represents the derivative of $x(t)$
2. choose a step length dx for the x direction and dt for the t direction, and vector length parameter h
3. choose a region in the x,t plane: x range (x_1, x_2) and t range (x_2, t_2)

4. set the number of grid points in the x direction $Nx = (x_2 - x_1)/dx$ and number of points in the t direction $Nt = (t_2 - t_1)/dt$
5. use the chosen step sizes evenly sample points from this region of the plane: do a loop for i from 0 to Nx and inside another loop for j from 0 to Nt:
   (a) let $x_i = x_1 + i * h$ and let $t_j = t_1 + j * h$
   (b) evaluate the function $f(x_i, t_j)$ and store it as the slope m
   (c) place a line segment with slope m at the point (t_j, x_i) to point $(t_j + h, x_i + h * m)$
   (d) increase i and j by 1

```
define function F(x: REAL, t: REAL)
var dx, dt : REAL                        // Step size
var h : REAL                             // Vector length
var x_min, x_max, t_min, t_max : REAL    // Plot limits
var x, t, m, u, v : REAL                 // temporary placeholders
var M, N : INT                           // Number of steps
init plot

M ← (x_max - x_min) / dx                // Number of
N ← (t_max - t_min) / dt                //      steps per axis

for i : INT in 0 to M
begin
    for j : INT in 0 to N
    begin
        x ← x_min + i * dx * h
        t ← y_min + j * dt * h
        m ← F(x, t)                     // Slope of field at (x, t)
        u ← x + h
        v ← y + h * m;
        draw [(x, t), (u, v)]            // Line segment
    end
end
```

Pseudocode listing 1 – Plotting direction fields

This is by far the largest algorithm presented in this course. Although it is *more important* to follow good coding *style* guidelines and organization when writing larger functions such as this, it is much harder to keep your code organized and readable. In writing the code for this exercise, try to recall these points.

Variable names Almost twenty different variable names are presented in the pseudocode outline above. Try to give your variables sensible names.

Comments Remember, the goal is to *express* your understanding of the abstract mathematical concepts. In-code comments also serve as the "post-it note" of the computing world. They are a great place to keep notes, questions, and references [1].

White space The algorithm above uses nested `for` loops to efficiently evaluate all points in the plotting range. As structure of an algorithm becomes more hierarchical (more heavily nested), indentation in

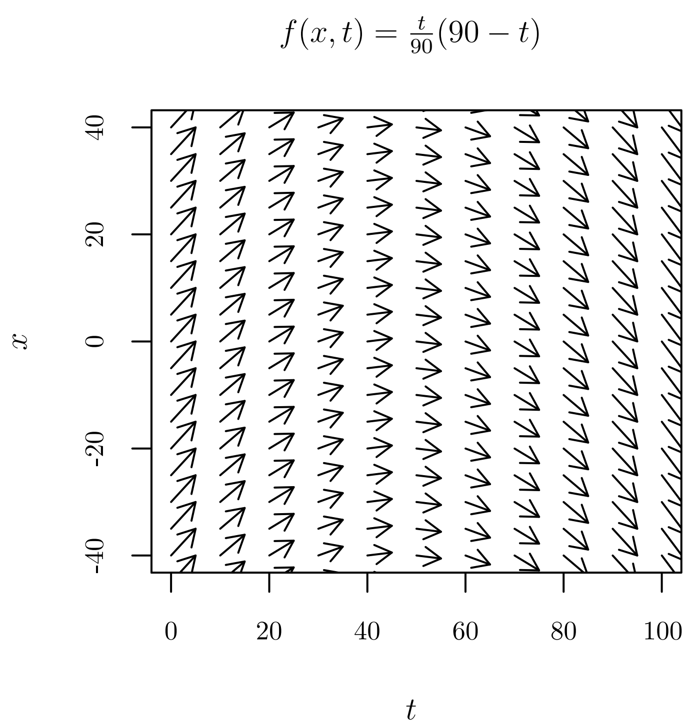
[1] TODO: Starting with commented out pseudocode.

particular helps to maintain readability.

Example

Consider the ODE given by $\dot{x} = \frac{x}{90}(90 - x)$ (example adapted from **?? ??**).

```
> exampleFun1 <- function(t) {
+   return(t * (1 - t/90))
+ }
> d_fun <- mDeriv(exampleFun1, 't');
> d_fun
> tRange <- c(-10, 100)
> dt <- 10
> xRange <- c(-50, 50)
> dx <- 5;
> h <- 5
> tSteps <- diff(range(tRange)) / dt
> xSteps <- diff(range(xRange)) / dx
> plot(0:100, -50:50 * 0.8, type='n',
+      main=paste("{\\large\\bf $f(x,t)",
+      " = \\frac{t}{90} (90 - t)$ }"),
+      xlab='{\\larger $t$ }',
+      ylab='{\\larger $x$}')
> for (ii in 0:(tSteps-1)) {
+   for (jj in 0:(xSteps-1)) {
+     x <- ii * dt
+     y <- jj * dx - 50
+     m <- d_fun(x)
+     u <- x + h
+     v <- y + h * m
+     arrows(x, y, u, v, code=2,
+       length=0.1)
+   }
+ }
```



1 New and revisited *R* commands

1. `range()`, `diff()`
   The `range()` function gives you both the minimum and maximum values of a vector. The `diff()` function takes as input a vector with n members and returns a vector with $n-1$ entries corresponding to the pairwise differences.

2. **Graphics primitives**

(a) `plot()` In the example above I "set up" the plot by plotting a pair of points using the `type='n'` parameter. This allows the plot to be "initialized" without actually drawing any points. Here the plot range is implicitly defined by the coordinates of these "dummy points."

(b) `arrows` *Appendix 1* provides a summary of the syntax for generating arrows and segments.

3. **Plot ranges and initial values**

(a) Identifying "interesting" plot ranges.

(b) Choosing initial values.

Part II

Exercise

1. Compare the following two models of population growth, where N represents the population size, and r and K are positive parameters.

$$\dot{N} = f(N) = rN(N/U - 1)(1 - N/K)$$

$$\dot{N} = g(N) = rN^2(1 - N/K)$$

(a) Find the equilibria of the two ODEs and analyze their stability analytically.

(b) Plot the functions $f(N)$ and $g(N)$ for values of $r = 10$ and $K = 200$ and describe for what values of N the population grows and for which it decays.

(c) What is the principal difference in the dynamics of the two population models? Give a biological interpretation of the intervals of growth and decay.

2. Suppose a bacterial culture in a lab environment grows with a rate r proportional to the population size N. Additionally, two groups of student experimenters are removing bacteria from the colony. The first group removes bacteria at a constant rate u and the second group removes bacteria as a sinusoidal function of time $v(t)$. (Maximal bacterial consumption at 12:00 noon, minimal consumption at 12:00 a.m.)

(a) Write down the differential equation model for describing this population.

(b) Find the fixed point(s) and analyze their stability analytically.

(c) Plot the direction field of this ODE using R using $r = 1$ and $d = 0.4$ and indicate where the fixed point(s) are and how the stability is reflected in the plot.

(d) Solve the ODE analytically, and plot three solution curves, starting at different initial conditions. Explain how the plot of the solution curves relates to the direction field you produced.

3. The rate of change of voltage across a membrane with two voltage-dependent ion channels can be described by the following equation:

$$C\dot{V} = -g_{Na}(V - V_{Na}) - g_K(V - V_K)$$

where C is the capacitance of the membrane, g_{Na} and g_K are the conductances of the sodium and potassium channels, and V_{Na} and V_K are the reversal voltages; all are positive constants.

Parameter	Sodium (Na^+)	Potassium K^+
Capacitance (C)	$0.15\ \mu F\ /\ cm^2$	
Conductance (g, mmho)	1	100
E_{rev} (mV)	58.1	-69.6

Characteristic electrical properties of the squid giant axon.

(a) Find the fixed points of the equation and analyze their stability analytically.

(b) Use R to plot the direction field of this model for the following typical scenario. Comment on whether it agrees with the fixed point stability analysis you performed.

(c) Find the general solution of Voltage as a function of time, and plot three solution curves, starting at different initial conditions. What does the model predict for the long-term behavior of the membrane potential? Did you need an analytic solution to come to this conclusion?

4. **Identifying algorithms.** Section 2.4.1 (**Plotting flow on the line**, page TEXT) describes another graphical method for the analysis of ODEs with one dependent variable.

(a) Translate the textual description from your text into a *pseudocode* algorithm that describes the production of plots similar to the one shown below. Your answer should be similar in structure **??**. Don't worry about precise syntax here.

(b) Implement your algorithm in *R* and use it to analyze logistic model

$$f(x) = \frac{1}{3}(1 - \frac{x}{5})x$$

Your plot should be similar in design to the above example.