Introduction to computational programming

Chapter 5 Exercise

Bifurcations and second-order systems

David M. Rosenberg
University of Chicago
Committee on Neurobiology

Version control information:	
Last changed date:	2009-11-12 09:37:14 -0600 (Thu, 12 Nov 2009)
Last changes revision:	226
Version:	Revision 226
Last changed by:	David M. Rosenberg

November 12, 2009

Part I

Tutorial

Overview

1 Bifurcation diagrams

Pseudocode for plotting bifurcation diagrams

1. Choose a range of parameter values: $[R_1, R_2]$ and the grid size ΔR; define $NR = (R_2 - R_1)/\Delta R$
2. Choose an array of initial values X_0, the number of total iterations $Niter$, and the number of iterations to discard $Ndisc$
3. Obtain a function $sol(X_0, R, Niter)$ that will return a numerical solution for a dynamical system for a specified number of iterations $Niter$
4. Let $i = 1$ and repeat while $i \leq NR$:
   (a) let $R[i] = R_1 + (i-1) * \Delta R$
   (b) initialize array of stable values $Xstab$

(c) let $j = 1$ and repeat while $j \leq length(X_0)$
   i. call the function $sol(X_0[j], R[i], Niter)$, which returns an array X of size $Niter$
   ii. discard the first $Ndisc$ values of X, add the rest to array $Xstab$
   iii. increase j by 1
(d) let $j = 1$ and repeat while $j \leq length(Xstab)$
   i. plot the point $(R[i], Xstab[j])$ on a graph
   ii. increase j by 1
(e) increase i by 1

```
var Rmin, Rmax, dR: REAL
var NR, Niter, Ndisc : INT
var x0 : REAL ARRAY
define F(x0 : REAL, R : REAL, Niter: INT)

for INT i in 1 to NR
begin
    R[i] ← R1 + (i-1) * dR
    var xstab : REAL ARRAY
    for j in 1 to length(X0)
    begin
        Y ← F(x0[j], R[i], Niter)
        xstab ← join(xstab, Y[Ndisc+1..])
    end
    for j in 1 to length(xstab)
    begin
        draw (R[i], xstab[j])
    end
end
```

Pseudocode listing 1 – Bifurcation diagram

2 Second-order systems

An important technique for dealing with higher order differential equations is conversions to a first order system. For example, the differential equation

$$y''(t) - y(t) = 0$$

can be written as a system of first-order equations by introduction of the “dummy variable” z as follows.

$$z'(t) - z(t) = 0$$
$$z'(t) - y(t) = 0$$

3 Matrix differential equations

Part II

Exercises

1. Explore the dynamics of the discrete logistic model $X_{t+1} = rX_t(1 - X_t)$
   (a) Use your code from the chapter 1 lab to solve the logistic model for 500 iterations, given any initial value and any value of the parameter r.
   (b) Write a "wrapper" code that will call your code to solve the logistic model for a range of values of r between $r = 0$ and $r = 4$ (with a reasonable step size around 0.01)
   (c) For each value of r, discard the first 200 iterations, and plot the rest on a bifurcation diagram.
   (d) Pick a few representative values of r and produce plots of the solutions over time (after discarding the first 200 iterations) for qualitatively different types of behavior (single fixed point, two cycle, n-cycle, chaos).
2. Consider the following Leslie population model:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 0.3 & 0 \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \end{pmatrix}$$

   (a) Explain the meaning of the variables x and y, and what the numbers in the matrix represent.
   (b) Write code to propagate the population vector, starting with some initial value x_0 and y_0.
   (c) Starting at different initial conditions, describe what population distribution the model converges to.
   (d) Estimate the largest eigenvalue λ from the convergence behavior.
3. Modify the population model in the previous problem by adding a parameter a instead of the zero in the matrix
   (a) Explain the meaning of the parameter a.
   (b) Write code to explore the convergence behavior of the population model, as you vary a from near 0 to progressively larger values, and report the equilibrium population distribution and the largest eigenvalue λ.
   (c) Report the approximate value of a at which a qualitative change in the solution occurs (bifurcation point). Produce plots of representative solutions on both sides of the bifurcation value of a. What is the value of the largest eigenvalue λ at the bifurcation point?
4. *Enrichment problem. Extra credit only.* A linked list is a ordered data structure consisting of an arbitrary number of elements called *nodes*. Each *node* contains a data part and a *pointer* to the next element in the list. Consider the following example.

```
> dataPart <- c('D', 'a', 'v', 'i', 'd');
> pointPart <- c( 5, 1, -1, 3, 4);
> lList <- list(dataPart=dataPart, pointPart=pointPart);
```

This linked list structure is ordered alphabetically. The advantage of linked lists is the ease with which an element is added. For example, to add an element to the list while maintaining the alphabetical ordering:

```
> newElement <- 'b'
> lList$dataPart <- c(lList$dataPart, newElement);
> lList$pointPart[2] <- 6;
> lList$pointPart <- c(lList$pointPart, 6);
```

(a) Write a function which takes an (unsorted) vector of type numeric and returns a linked list containing those values.

(b) Write a function which takes a linked list and adds a single element to it (preserving ordering).

(c) Write a function which takes a linked list and removes a single element.