

SIMULATING INTEGRODIFFERENCE EQUATIONS WITH MATLAB  
TRIESTE, ITALY  
MARCH 2000

Michael Neubert  
Biology Department, MS 34  
Woods Hole Oceanographic Institution  
Woods Hole, MA 02543-1049, USA  
*mneubert@whoi.edu*

**Exercise 1: Critical Habitat Size** <sup>1</sup>

As provided, the Matlab program “ide.m” simulates the scalar integrodifference equation

$$n_{t+1}(x) = \int_{-L/2}^{L/2} k(x-y) f(n_t(y)) dy, \quad (1)$$

with the Laplace kernel

$$k(x) = \frac{\alpha}{2} e^{-\alpha|x|}, \quad (2)$$

and the Beverton-Holt growth function

$$f(n) = \frac{rn}{1 + (r-1)n}. \quad (3)$$

The parameters are initially set as  $r = 1.5$ ,  $\alpha = 1$ , and  $L = 100$ . Explore the effect the effect of decreasing habitat size by reducing  $L$ .

**Exercise 2: Period-Doubling Bifurcations**

Change the growth function in your program to the Ricker curve:

$$f(n) = ne^{-r(1-n)}. \quad (4)$$

Starting from the same parameter set as in the previous exercise, explore the consequences of increasing density dependence by increasing  $r$ . (Try  $r = 1.8, 2.2, 2.6, \dots$ . You may need to change the graphics parameter `ymax` in order to fit the solution in the axes.) The program starts with the population with a small range at the center of the habitat. What happens if the population is initially uniformly distributed over the interval. (You can change the extent of the initial condition with the constant “`irange`”.)

**Exercise 3: Allee Effects**

Now try the growth function

$$f(n) = n + n(n - 0.25)(1 - n), \quad (5)$$

which has an Allee effect. How does your solution depend upon the initial condition? In particular, explore the consequences of decreasing the population’s initial range (`irange`) while keeping the initial density over that range at the carrying capacity (`idens=1`).

---

<sup>1</sup>Kot, M. and W. M. Schaffer. 1986. Discrete-time growth-dispersal models. *Mathematical Biosciences* 80:109-136.

#### Exercise 4: Pattern Formation <sup>2</sup>

Modify your program by introducing a predator:

$$n_{t+1}(x) = \int_{-L/2}^{L/2} k_n(x-y) f(n_t(y), p_t(y)) dy, \quad (6)$$

$$p_{t+1}(x) = \int_{-L/2}^{L/2} k_p(x-y) g(n_t(y), p_t(y)) dy, \quad (7)$$

with

$$f(n, p) = ne^{r(1-n-p)}, \quad (8)$$

$$g(n, p) = cnp. \quad (9)$$

Use the dispersal kernels

$$k_n(x) = \frac{\alpha^2}{2} |x| e^{-\alpha|x|} \quad (10)$$

$$k_p(x) = \frac{\beta}{2} e^{-\beta|x|}. \quad (11)$$

On an infinite domain, this system would have a spatially uniform equilibrium at  $n = n^* = 1/c$ ,  $p = p^* = 1 - 1/c$ , that would be stable for  $1 < c < 2$  and  $0 < r < 4c/(3 - c)$ . Try using this as your initial condition, with the parameter values  $L = 100$ ,  $r = 2.5$ ,  $c = 1.2$ ,  $\alpha = 13$ , and  $\beta = 5$ . What happens as you increase the predator's average dispersal distance (by decreasing  $\beta$ )? Next try the parameters  $r = 0.9$ ,  $c = 1.75$ ,  $\alpha = 13$ , and  $\beta = 10$ . What happens as you increase the average dispersal distance of the prey (by decreasing  $\alpha$ )?

---

<sup>2</sup>Neubert, M. G., M. Kot and M. Lewis. 1995. Dispersal and pattern formation in a discrete-time predator-prey model. *Theoretical Population Biology* 48:7-43.

```

% ide.m
% An integrodifference equation simulator
% Author: Mike Neubert
% 2/25/2000

% BIOLOGICAL PARAMETERS
% r = growth parameter
% alpha = dispersal parameter

    r = 1.5;
    alpha = 1.0;

% PHYSICAL PARAMETERS
% iterations = number of iterations to perform
% L = length of the domain

    iterations = 100;
    L = 100;
    radius = L/2;

% NUMERICAL PARAMETERS
% nodes = number of computational nodes in domain (should be 2^m + 1)
% x = vector of node locations
% x2 = vector of node locations for extended domain
% dx = internode distance
% n = species density
% lowval = numerical truncation

nodes = (2^11)+1;
x = linspace(-radius,radius,nodes);
x2 = linspace(-L,L,2*nodes-1);
dx = L/(nodes-1);
n = zeros(1,length(x));
lowval = 1e-15;

% k = dispersal kernel

k = (alpha/2)*exp(-alpha*abs(x2));

% GRAPHICS PARAMETERS
% xmin, xmax, ymin, ymax: graphics scales

xmin = -radius; xmax = radius;
ymin = 0; ymax = 1.5;
grlim = [xmin xmax ymin ymax];

% SET THE INITIAL CONDITIONS
% irange = initial range radius
% idens = initial density

irange = 5;
idens = 1;
temp = find(abs(x)<irange);
n(temp) = idens*ones(size(n(temp)));

```

```

% PLOT THE INITIAL CONDITION

plot(x,n);
axis(grlim);
xlabel('x');
ylabel('density');
title('t = 0');
drawnow;
disp('Strike any key to continue...');
pause

%DO THE SIMULATION ITERATIONS

n1 = zeros(1,length(x2)+length(x)-1);

for j = 1:iterations

    n1 = zeros(size(n1));

    % First we apply the Beverton-Holt Growth
    % for the sedentary stage. (The operator .* multiplies
    % vectors element-by-element.)

    f = r*n./(1 + (r-1)*n);

    % Then we convolve f with k for the dispersal stage
    n1 = fft_conv(k,f);
    n = dx*n1(nodes:length(x2));
    n(1) = n(1)/2; n(nodes) = n(nodes)/2;
    v = find(n < lowval);
    n(v) = zeros(size(n(v)));

    % Plot the result
    plot(x,n);
    title(['t = ',num2str(j)]);
    xlabel('x');
    ylabel('density');
    axis(grlim);
    drawnow;

end

```