A Numerical Approach to Calculating Population Spreading Speed

by

Angela Leo

A Thesis

submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science in

Applied Mathematics

May 2007

APPROVED:

Roger Lui, Major Advisor

Darko Volkov, Advisor

Abstract

A population density, $u_n(x)$, is recursively defined by the formula

$$u_{n+1}(x) = \int K(x-y)\Big(1 - g\big(u_n(y)\big)\Big) f\big(u_n(y)\big) dy + g\big(u_n(x)\big) f\big(u_n(x)\big).$$

Here, K is a probability density function, $g(u)$ represents the fraction of the population that does not migrate, and f is a monotonically decreasing function that behaves like the Beverton-Holt function. In this paper, I examine and modify the population genetics model found in [17] to include the case where a density-dependent fraction of the population does not migrate after the selection process.Using the expanded model, I developed a numerical application to simulate the spreading of a species and estimate the spreading speed of the population. The application is tested under various model conditions which include both density-dependent and density-independent dispersal rates. For the density-dependent case, I analyzed the fixed points of the model and their relationship to whether a given species will spread.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since the invasion of an exotic species can become a major threat to the biodiversity of an area, scientists are concerned with studying the migration and spread of a population. As a result, mathematicians are becoming increasingly involved with modeling the spreading speed of a species in a given habitat. An ability to predict the rate of spread, as well as the biological factors that can alter that rate would be very useful [15]. Most models thus far have considered a population with a density-independent dispersal rate. However, a large amount of empirical evidence suggests that the dispersal rate of a given species depends partially on the population density. Density-dependent dispersal rates have been observed in a range of species including spiders, insects, echinoderms, mammals, birds, and fish [5].

Using the model found in [17], a numerical application is created to simulate the spread of a species in a given habitat and calculate the spreading speed. The spreading speed of a population is a constant c^* such that

$$\lim_{n\to\infty} \max_{x\in[-nc,nc]} u_n(x) = 0 \qquad \forall c > c^*$$

and

$$\lim_{n\to\infty} \min_{x\in[-nc,nc]} u_n(x) = M \qquad \forall c < c^*.$$

where $M > 0$ is the carrying capacity of the population. The application is then tested using a number of different scenarios involving populations with density-independent disperal rates. The spreading speed estimates are then compared to results of the analytical formula for spreading speed found in [17]. Next, the model is modified to include a density-dependent dispersal rate. Thus, the fraction of the population that remained in an area after the selection process would now depend on the local population density.

The model considers a population living in a homogenous infinite habitat $\mathbf{R}^d$. The population density,

$$\begin{aligned} u_{n+1}(x) &= Q_g[u_n(x)] \\ &= \int K(x-y)\Big(1-g\big(u_n(y)\big)\Big)f\big(u_n(y)\big)dy + g\big(u_n(x)\big)f\big(u_n(x)\big) \end{aligned}$$

is in the $n+1$-generation at location x. The growth law, $f(u)$, is a monotonically increasing function, and $K(x)$ is a probability density function. If the dispersal rate is density-independent, $g(u)$ is a constant function that lies between 0 and 1. In our study, when the dispersal rate is density-dependent, the fraction of the population that does not migrate is chosen to be $g(u) = e^{-au}$, where $a > 0$.

For the density-independent case, the density-independent selection migration operator, Q_g, is order-preserving, i.e. if $u \leq v$, then $Q_g[u] \leq Q_g[v]$. In addition, [17] derives an analytical formula for the spreading speed, c^*, of a population with a Gaussian probability function, and for $c \geq c^*$, proves the existence of traveling wave solutions for Q_g. In order to expand the work done in [17], this paper shows that the density-dependent operator, Q_g, is also order-preserving, under certain conditions. The numerical application is used to estimate c^* for a population with a density-dependent dispersal rate. Also, the paper investigates the relationship between the model's fixed points and the chance that a species will spread.

Chapter 2

Mathematical Model

2.1 Model 1: The density-independent case

$Q_g[u]$ denotes the selection migration operator

$$Q_g[u] = \int K(x-y)\Big(1 - g\big(u(y)\big)\Big) f\big(u(y)\big)dy + g\big(u(x)\big) f\big(u(x)\big) \qquad (2.1)$$

where the growth law $f(u)$ is a concave increasing function defined on $[0, M]$ with $f(0) = 0$ and $f(M) = M$ where M is the carrying capacity of the popultaion. In the density-independent case, the fraction of the population that does not migrate, $g(u)$, is constant. Let $\mathbf{C}_M$ be the set of all bounded functions defined on $\mathbf{R}^d$ with values in $[0, M]$ and define $Q_g(0) = 0$ and $Q_g(M) = M$. Then, since $f(u)$ is increasing, Q_g is order-preserving, i.e. if $u, v \in \mathbf{C}_M$ and $u \leq v$, then $Q_g[u] \leq Q_g[v]$. Another important observation is that $Q_g[\alpha] > \alpha$ for $\alpha \in (0, M)$. This implies that 0 is an unstable fixed point and M is a stable fixed point for the operator Q_g since $Q_g^n[\alpha] \to M$ as $n \to \infty$. Therefore, with the initial population density $u_0(x) > 0$, a density-independent species will always grow to its carrying capacity and spread. However, when the dispersal rate is density-dependent, this is not the case as is discussed in Section 2.2.

Aronson and Weinberger define the spreading speed of a population, c^*, as the speed such that if one moves faster than c^*, the population density, u_n, is near 0 for large n. On the other hand, if one runs at a speed smaller than c^*, then u_n is near the carrying capacity M. He provides the analytical formula for spreading speed

$$c_g^*(\zeta) = \inf_{\mu>0} \frac{1}{\mu} \log[(1-g)r \int_{\mathbf{R^d}} e^{\mu x \zeta} K(x) dx + rg].$$

This formula is for a general kernel K and a function f as described later in this section. If $K(x) = (\frac{\sigma}{\pi})^{\frac{n}{2}} e^{-\sigma|x|^2}$, then

$$c_g^* = \inf_{\mu>0} \frac{1}{\mu} \log[r(1-g)e^{\mu^2/4\sigma} + rg], \tag{2.2}$$

where $r > 1$ is the intrinsic growth rate and g is constant $\in [0,1]$. This equation will be used in Section 4 in order to calculate the accuracy of the numerical program.

2.2 Model 2: The density-dependent case

Let $Q_g[u]$ denote the density-dependent selection migration operator with

$$\begin{aligned} Q_1[u] &= \int K(x-y)\Big(1-g\big(u(y)\big)\Big) f\big(u(y)\big) dy \\ Q_2[u] &= g\big(u(x)\big) f\big(u(x)\big) \\ Q_g[u] &= Q_1[u] + Q_2[u] \end{aligned}$$

where $g(u) = e^{-au}$ with $a > 0$, $K(x)$ is a probability function, and $f(u)$ is an increasing bounded function in $[0, M]$, where M is the carrying capacity of the population. Then, the following three lemmas hold true.

Lemma 2.2.1. *$Q_1[u]$ is order-preserving; i.e. if $u, v \in \mathbf{C}_M$ and $u \leq v$, then $Q_1[u] \leq Q_1[v]$.*

Proof. For any $u \leq v$, we have $f\big(u(y)\big) \leq f\big(v(y)\big)$ since f is an increasing function. Also, $(1 - e^{-au(y)}) \leq (1 - e^{-av(y)})$. Therefore, because $K(x-y)$ is positive and independent of u and v,

$$\begin{aligned} Q_1[u] &= \int K(x-y)\Big(1-g\big(u(y)\big)\Big) f\big(u(y)\big) dy \\ &\leq \int K(x-y)\Big(1-g\big(v(y)\big)\Big) f\big(v(y)\big) dy = Q_1[v]. \end{aligned}$$

Thus, $Q_1[u]$ is order-preserving. ☐

Lemma 2.2.2. *Assuming that $f(u)$ is a concave function with values on $[0, M]$, $Q_2[u]$ is order-preserving if and only if $f'(M) \geq aM$ with $a > 0$.*

Proof. ($\Rightarrow$) Assume $Q_2[u]$ is order-preserving. Then, for any $u \leq v$, $Q_2[u] \leq Q_2[v]$. Thus,

$$e^{-au} f(u) \leq e^{-av} f(v).$$

Let $h_1(u) = e^{-au} f(u)$. Since $Q_2[u]$ is order-preserving,

$$\begin{aligned} h_1'(u) &= e^{-au}\big(- af(u) + f'(u)\big) \\ &\geq 0. \end{aligned}$$

Due to the fact that f is increasing and concave, $-af(u)+f'(u)$ is decreasing on $[0, M]$. Thus, the sum $-af(u) + f'(u)$ attains its minimum at $u = M$. Knowing $e^{-au} \geq 0$ for all u, substitute $u = M$ to get

$$\begin{aligned} -af(M) + f'(M) &\geq 0 \\ f'(M) &\geq aM. \end{aligned}$$

($\Leftarrow$) Let $f'(M) \geq aM$. Then,

$$\begin{aligned} f'(M) &\geq af(M) \\ -af(M) + f'(M) &\geq 0. \end{aligned}$$

Because the sum of two decreasing functions is decreasing, we know $-af(u)+f'(u)$ decreases on $[0, M]$. Since the inequality holds at $u = M$, $-af(u) + f'(u) \geq 0$ for all u in $[0, M]$. Thus, for $u \leq v$, $Q_2[u] \leq Q_2[v]$, and $Q_2[u]$ is order-preserving. ☐

Lemma 2.2.3. *If $Q_1[u]$ and $Q_2[u]$ are order-preserving, then $(Q_1 + Q_2)[u]$ is also order preserving.*

Proof. Assume $Q_1[u]$ and $Q_2[u]$ are order-preserving. Then, for any $u \leq v$, $Q_1[u] \leq Q_1[v]$, and $Q_2[u] \leq Q_2[v]$. We then have

$$(Q_1 + Q_2)[u] = Q_1[u] + Q_2[u] \leq Q_1[v] + Q_2[v] = (Q_1 + Q_2)[v]$$

Therefore, $(Q_1 + Q_2)[u]$ is order-preserving. ☐

Lemma 2.2.4. *Let X, Y be two metric spaces and let f be a continuous function from $X \times Y$ into $\mathbf{R}$. Assume Y is compact and define*

$$F(x) = \sup_{y \in Y} f(x, y).$$

Then, F is continuous on X.

Proof. Fix x^* in X. Let x_n be a sequence in X converging to x^*. As Y is compact there is a y_n in Y such that $F(x_n) = f(x_n, y_n)$. Again by compactness, y_n has a subsequence y'_n converging to some y^* in Y.
As (x'_n, y'_n) converges to (x^*, y^*) in $X \times Y$ and by continuity of f, we infer that $f(x'_n, y'_n)$ converges to $f(x^*, y^*)$. It follows that

$$F(x'_n) = f(x'_n, y'_n) \to f(x^*, y^*) \leq F(x^*)$$

Arguing by contradiction assume that $F(x^*) > f(x^*, y^*)$. Set $\epsilon = F(x^*) - f(x^*, y^*)$ and let y_2^* in Y be such that $F(x^*) = f(x^*, y_2^*)$. By continuity of f, as x'_n converges to x^*, there is a p such that for $n > p$

$$f(x'_n, y_2^*) > f(x^*, y_2^*) - \epsilon/2.$$

But as $f(x'_n, y'_n) \geq f(x'_n, y_2^*)$ we find that $\lim f(x'_n, y'_n) \geq f(x^*, y_2^*) - \epsilon/2$, contradiction.
We showed that for any convergent sequence x_n in X of limit x^*, there is a subsequence x'_n such that $F(x'_n)$ converges to $F(x^*)$: this shows that F is continuous at x^*. $\square$

Remark 2.2.5. *If we only assume that f is continuous on $X \times Y$ and bounded then F may not be continuous as shown by setting $X = [0, 1]$, $Y = [1, \infty)$, $f(x, y) = 1 - x^y$.*

With the density-dependent case, since $g(u)$ is not constant, the model equation becomes more complex. Since the dispersal rate is now density-dependent, the fixed points of the system will be different. Define a fixed point of the system u_* such that $h(u_*) = u_*$. In this study, we assume the growth law $f(u)$ has the following properties:

1. $f \in C^1[0, M]$.
2. $f(0) = 0$ and $f(M) = M$.
3. $f(u) > u$ for $u \in (0, M)$.

4. $f'(u) \geq 0$, and $f'(0) = r > 1$.
5. $f(u) \leq ru$ for $u \in [0, M]$.

With $h(u) = (1 - e^{-au})f(u)$ where $a > 0$ the following lemma is valid.

Lemma 2.2.6. *In the density-dependent case, $h(u)$ has the following properties:*

1. *$h(u)$ is increasing.*
2. *$h(u)$ is convex near 0.*
3. *$u = 0$ is a fixed point of $h(u)$.*
4. *Let f be the Beverton-Holt function, that is,*

$$f(u) = \frac{rMu}{M + (r-1)u}$$

There exists $\bar{a} > 0$ such that
- *(i) if $0 < a < \bar{a}$, h has one fixed point, $u = 0$.*
- *(ii) if $a = \bar{a}$, h has two fixed points, $u = 0$ and M_1 such that $0 < M_1 < M$.*
- *(iii) if $a > \bar{a}$, h has three fixed points, $u = 0$, M_1, and M_2 such that $0 < M_1, M_2 < M$.*

Proof. 1. 2. and 3. are clear. To show 4., we remark that

$$h(u) - u = \frac{rMu}{M + (r-1)u} h_2(u, a)$$

where

$$h_2(u, a) = \left(1 - \frac{1}{r}\right)\left(1 - \frac{u}{M}\right) - e^{-au}.$$

Define $H(a) = \max_{u \in [0,M]} h_2(u, a)$. H is continuous on $[0, \infty)$ due to the previous lemma.

It is clear that $H(0) = -\frac{1}{r} < 0$. Furthermore,

$$h_2(u, \frac{M}{2}) = \frac{1}{2}(1 - \frac{1}{r}) - e^{-a\frac{M}{2}}$$

is clearly positive if a is large enough, thus $H(a) > 0$ if a is large enough. By continuity of H, there exists $\bar{a}$ in $(0, \infty)$ such that $H(\bar{a}) = 0$.

We proceed to show that $\bar{a}$ is unique. Let a_1 and a_2 be such that $0 \le a_1 < a_2$. There exists u_1 in $[0, M]$ such that $H(a_1) = h_2(u_1, a_1)$. If $u_1 \ne 0$,

$$H(a_1) = h_2(u_1, a_1) < h_2(u_1, a_2) \le H(a_2).$$

If $u_1 = 0$, we can only conclude that $H(a_1) \le H(a_2)$, and $H(a_1) = -\frac{1}{r} < 0$. Thus if we denote $\tilde{a}$ the smallest a in $[0, \infty)$ such that $H(a) = 0$, H is strictly increasing on $[\tilde{a}, \infty)$, proving that $\bar{a}$ is unique.

If $0 < a < \bar{a}$, it is clear that $h_2(u, a) < 0$ for all u in $[0, M]$, thus proving h has one fixed point exactly, $u = 0$.

Assume $a = \bar{a}$. As $h_2(0, a) < 0$, $h_2(M, a) < 0$, and $\max_{u \in [0,M]} h_2(u, \bar{a}) = 0$, then if a point v in $(0, M)$ satisfies $h_2(v, \bar{a}) = 0$, we must have $\frac{\partial h_2}{\partial u}(v, \bar{a}) = 0$. As

$$\frac{\partial h_2(u,a)}{\partial u} = \left(1 - \frac{1}{r}\right)\left(\frac{-1}{M}\right) + ae^{-au}$$

and

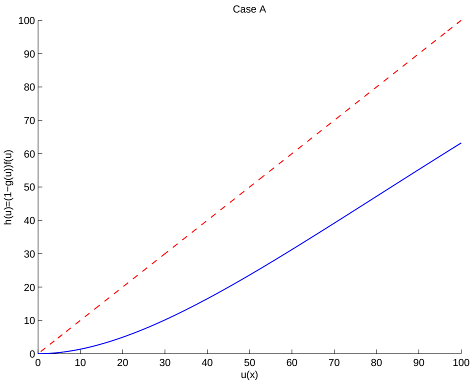
$$\frac{\partial^2 h_2(u,a)}{\partial u^2} = -a^2 e^{-au} < 0, \text{ for } a > 0,$$

by Rolle's theorem there is only one point M_1 in $(0, M)$ such that $h_2(M_1, \bar{a}) = 0$. h has the two fixed points 0 and M_1.

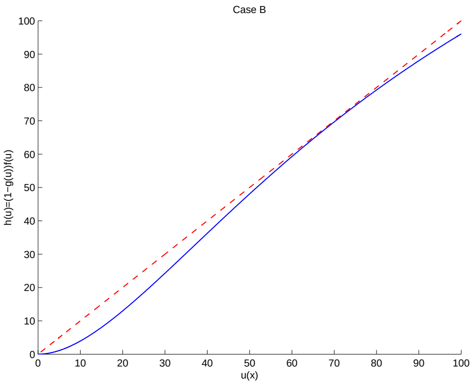
If $a > \bar{a}$, then as $h_2(0, a) < 0$, $h_2(M, a) < 0$, and $\max_{u \in [0,M]} h_2(u, a) > 0$ there are by continuity of h_2 two numbers M_1 and M_2 such that $0 < M_1 < M_2 < M$ and $h_2(M_1, a) = h_2(M_2, a) = 0$. There is no third point v in $(0, M)$ such that $h_2(v, a) = 0$: that would contradict Rolle's theorem as $\frac{\partial^2 h_2(u,a)}{\partial u^2} < 0$. We conclude that h has three fixed points, 0, M_1, and M_2.

Illustrations of the three cases can be seen in Figure 2.1. □

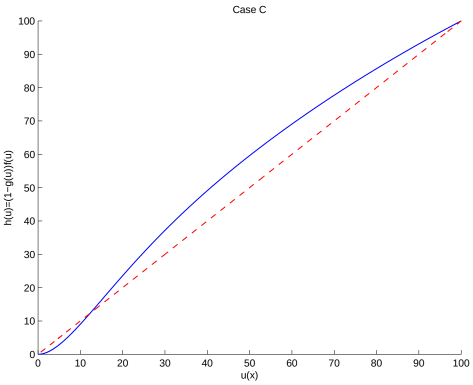
During Case A, the population density will approach 0 and the population will die out over time. If the conditions for Case C are satisfied, one of two things may occur. In order for a population with a density dependent dispersal rate to spread, the initial population density, $u_0(x)$ must be greater than or equal to M_1 for x on a sufficiently large interval. With such an initial population density, the population will grow towards its carrying capacity and spread. However, if $u_0(x) > M_1$ on a not large enough interval, the population will behave as in Case A and die out. Section 4.2 investigates these claims using the numerical program.



(a) Case A - 1 fixed point.



(b) Case B - 2 fixed points.



(c) Case C - 3 fixed points.

Figure 2.1: Illustrations of fixed points for $h(u)$.

Chapter 3

Numerical Methods

In order to simulate the spread of a species and estimate the spreading speed, c^*, I develop a program in Matlab to model

$$\begin{aligned} u_{n+1}(x) &= Q_g[u_n] \\ &= \int K(x-y)\Big(1-g\big(u_n(y)\big)\Big)f\big(u_n(y)\big)dy + g\big(u_n(x)\big)f\big(u_n(x)\big) \end{aligned} \tag{3.1}$$

where

$$\begin{aligned} K(x) &= \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \\ f(u) &= \frac{rMu}{M+(r-1)u} \\ u(x) &\in [0, M]. \end{aligned}$$

For the density-independent case, $g(u) = g$ where g is a constant $\in [0, 1]$. The density-dependent case has $g(u) = e^{-au(x)}$ with $a > 0$ in our study. To define a given model, the following population specific parameters must be defined:

M - carrying capacity of the population

r - intrinsic growth rate, $(r > 1)$

g - fraction of the population that does not migrate[1]

---

[1] density-independent case

a - influential variable for $g(u)$[2], $a > 0$

μ - mean for $K(x)$

σ - standard deviation for $K(x)$

B - integer such that $K(x)$ is zero outside of $[-B, B]$, $B = 6$ for the algorithm

The step variable, dx, defines how the interval will be divided. Explorations into various values of dx showed that $dx = 1$ is the optimal choice for run time and error (See Table 3.1).

Table 3.1: Numerical computation of speed for different values of dx with $g = 0$, $r = 2$, and $B = 6$

dx	Numerical Speed	Error(percent)	Run time
0.1	1.1753563	0.18	15 minutes
1	1.1758002	0.14	5 minutes
2	1.1648973	1.06	2 minutes

3.1 Pseudocode

Below is the pseudocode to compute Equation 3.1 in Matlab. The initial population density is assumed to have non compact support. Let $\epsilon > 0$ be a small number.

- Let $u_0(x)$ be the initial population density. In the algorithm,

$$\begin{aligned} u_0(x) &= M \text{ for } x \leq 0 \\ &= Me^{-x^2} \text{ for } x > 0. \end{aligned}$$

- Find x^* such that $u_0(x^*) < \epsilon$.
- Let B be defined as above and set $[a_0, b_0] = [-4B, x^* + B]$.
- Assume a_n, b_n are defined and u_n is known.

---

[2] density-dependent case only

- Find the smallest a and the largest b such that f has the following properties:

$$\begin{aligned} f(u_n(a)) &\geq M - \epsilon \\ f(u_n(b) &\leq \epsilon. \end{aligned}$$

- Let $[a_{n+1}, b_{n+1}] = [a - B, b + B]$.
- For $x < a_n$, replace $u_n(x)$ by M, and for $x > b_n$, replace $u_n(x)$ by 0. Denote this new function as $u_n^*(x)$.
- Use convolution as described in Section 3.2 to evaluate the integral in Equation (3.1) and obtain the new population density $u_{n+1}(x)$.

If one wishes to model a population with compact support, a modification to the pseudocode is necessary. While searching for the new interval $[a_{n+1}, b_{n+1}]$, find the smallest a such that $f(u_n(a)) < \epsilon$.

3.2 Computing convolution

The convolution of two functions is defined as

$$\begin{aligned} K * h(x) &= \int_{-\infty}^{\infty} K(x-y)h(y)dy \\ &= \lim_{\delta x \to 0} \sum_{n=-\infty}^{\infty} K(x - y_n)h(y_n)\delta x \end{aligned}$$

where $y_n = n\delta x$.

Matlab's built-in function, *conv*, takes in two vectors h and K and returns a vector w where

$$w(k) = \sum_j h(j)K(k+1-j).$$

Since *conv* only computes the summation, one must multiple h and K by $\sqrt{\delta x}$ before calling *conv* to complete the riemann sum.

3.3 Calculating spreading speed

In the case of Fisher's equation, a well studied PDE having to do with population growth, the numerical equation for the spreading speed of a population, c^*, is

$$u_n^{-1}(\gamma) = nc^* + k \ln n + C$$

where C depends on the initial condition $u_0(x)$, [7]. We assume the equation holds for out model as well. Then, solving for c^*, we obtain

$$c^* = \frac{u_n^{-1}(\gamma)}{n} + \frac{k \ln(n)}{n} + \frac{C}{n} \tag{3.2}$$

as $n \to \infty$, where $\gamma \in (0, M)$ and k and C are constants. Originally, the constants k and C are assumed to be 0 so that Equation (3.2) is reduced to

$$c^* = \frac{u_n^{-1}(\gamma)}{n}. \tag{3.3}$$

Although we can use this formula to estimate c^*, the assumption that k and C are 0 may introduce error into the calculation. Equation (3.2) should therefore be modified so that a more accurate numerical formula for the spreading speed can be developed. Using Equation (3.2), we know that

$$\begin{array}{rcl} u_n^{-1}(\gamma) & = & nc^* + k\ln(n) + C \\ u_{2n}^{-1}(\gamma) & = & 2nc^* + k\ln(2n) + C \\ u_{4n}^{-1}(\gamma) & = & 4nc^* + k\ln(4n) + C. \end{array}$$

Subtracting $u_n^{-1}(\gamma)$ from $u_{2n}^{-1}(\gamma)$ leaves

$$u_{2n}^{-1}(\gamma) - u_n^{-1}(\gamma) = nc^* + k \ln 2$$

so that

$$c^* = \frac{u_{2n}^{-1}(\gamma) - u_n^{-1}(\gamma)}{n} = \frac{k \ln 2}{n}. \tag{3.4}$$

Now, as $n \to \infty$ the formula converges to the solution faster since the constant C has been eliminated and $\ln 2$ has replaced $\ln n$. Since we want to develop an accurate program, further modifications of Equation (3.2) are needed to eliminate k. Taking a linear combination of the above equations and solving for c^* yields

$$c^* = \frac{u_{4n}^{-1}(\gamma) - 2u_{2n}^{-1}(\gamma) + u_n^{-1}(\gamma)}{n}. \tag{3.5}$$

Because this equation does not contain k, it provides the most accurate estimation of the spreading speed of a population out of the three methods. Section 4 compares the accuracy of the three formulas with the analytical formula for spreading speed given in Equation (2.2).

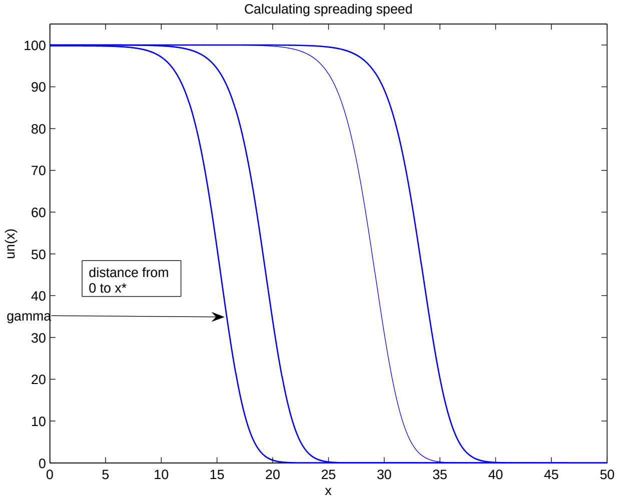


Figure 3.1: Illustration of γ and $u_n^{-1}(\gamma)$.

Once a population density is found, the program must calculate the spreading speed of the population. For each $u_n(x)$, since u_n is decreasing, use cubic interpolation to choose x_n^* such that $u_n(x_n^*) = \gamma$ for some user defined $\gamma \in [0, M]$. Let $u_n^{-1}(\gamma)$ be uniquely defined as the distance from 0 to x_n^* (See Figure 3.1). Using Equation (3.5), the numerical speed of the population can now be found and is displayed in the command window of Matlab. Lastly, the population density $u_n(x)$ is plotted versus the location, x, for all n.

Chapter 4

Numerical Results

In order to determine the best numerical formula to use in calculating the spreading speed, data from all three formulas is examined at different numbers of iterations and can be seen in Table 4.1. All tests have $B = 6$and $dx = 1$. For the simulation, $g = \frac{2}{3}$, $r = 5$ and the actual spreading speed of the population is 1.36782334. Since Equation (3.5) has the lowest error, this formula will be used to estimate the spreading speed in the program.

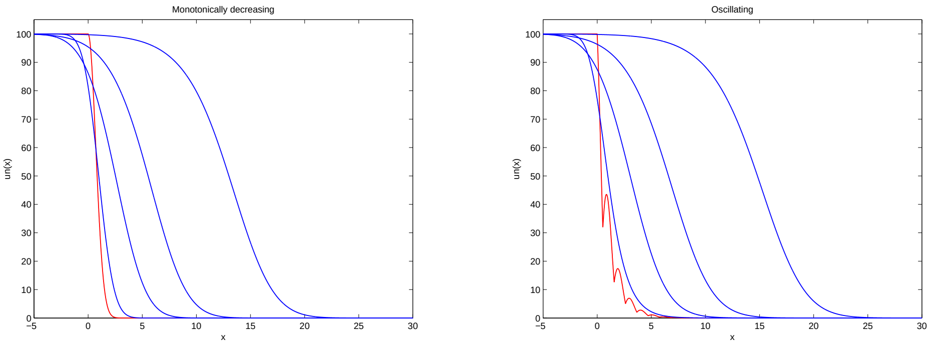
Table 4.1: Numerical computation of speed using different methods

Eq.	400 iter.	Error (%)	800 iter.	Error(%)	1200 iter.	Error(%)
(3.3)	1.3578948	0.72	1.3607321	0.52	1.3616819	0.45
(3.4)	1.3634974	0.32	1.3635694	0.31	1.363575	0.31
(3.5)	1.36351024	0.32	1.3635715	0.31	1.3635872	0.31

For the less complex, density-independent case, the accuracy of the results is determined using the analytical formula for c^* found in Equation (2.2) (see Section 4.1). Simulations are then run for the population with a density-dependent dispersal rate (see Section 4.2). In both cases, $B = 6$ and the step $dx = 1$.

4.1 Density-independent Case

In order to simulate the spread of a population, one must define the initial population density, $u_0(x)$. Three different types of initial population densities were tested. First, $u_0(x)$ was defined as a monotonically decreasing function with non-compact support. A second function that is not monotonically decreasing was also tested. Figure 4.1 shows the initial population density (red line) as well as the population densities at $n = 1, 5, 10, 20$ for each of the initial conditions.



(a) Monotonically decreasing.

(b) Oscillating.

Figure 4.1: Population spread where $u_0(x)$ has non-compact support.

To further test the robustness of the program, $u_0(x)$ was also defined as a function with compact support. From this example, one can see that even if $u_0(x) < M$, the population density $u_n(x)$ will increase to M as n grows. Figure 4.2 shows the initial conditions (red line) and the results of the simulation for $n = 1 - 5, 10, 20$. In this case, more values of n were shown to illustrate how $u_n(x)$ grows to M. Regardless of the initial conditions, the spreading speed of the given population is the same to a certain degree of accuracy.

In order to test the accuracy of the program, nine simulations with populations having different growth rates and migratory properties are performed. The intrinsic growth rate, r, and the fraction of the population that does not migrate, g, are the variables chosen to vary. The program estimates the numerical spreading speed for each population using Equation (3.5). The actual spreading speeds for the populations are then calculated with Equation (2.2). The results of these simulations and calculations as well as the relative error in each case can be seen in Table 4.2.

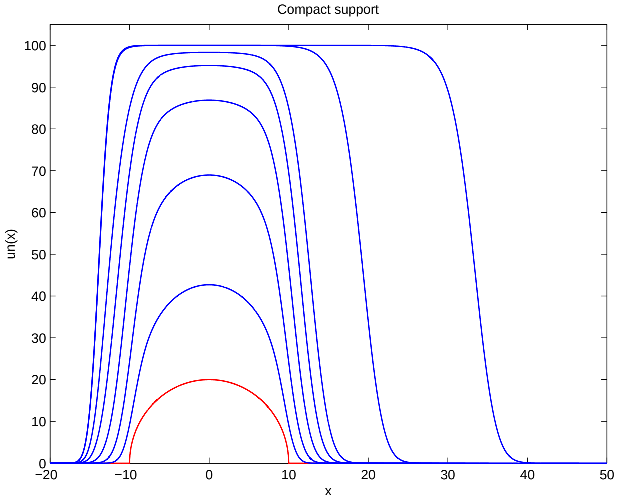


Figure 4.2: Population spread where $u_0(x)$ has compact support.

Table 4.2: Numerical computation of speed for the density-independent case

g	r	Actual Speed	Numerical Speed	Error(percent)
0	1.5	0.90051663	0.89995208	0.06
0	2	1.17741002	1.1758002	0.14
0	5	1.79412257	1.7910344	0.17
0.3333	1.5	0.76486977	0.7641128	0.10
0.3333	2	1.0200489	1.0180643	0.19
0.3333	5	1.62122706	1.6176532	0.22
0.6666	1.5	0.58870501	0.58768531	0.17
0.6666	2	0.81032342	0.80794862	0.29
0.6666	5	1.36782334	1.3635872	0.30

One can see that as the intrinsic growth rate, r, increases, the relative error increases. A similar situation is seen as the fraction of the population that does not migrate, g, increases. Analyzing these observations, one would expect the largest error to occur when g is close to 1 and r is large. In other words, as a population's migration declines and the growth rate expands, it is harder to numerically predict the spreading speed of the population.

4.2 Density-dependent Case

Currently, no known analytical formula for the spreading speed in the density-dependent case has been derived. We assume the accuracy that our numerical methods demonstrated in the density-independent case will carry over to the dependent case. As a population's migration declines, g approaches 1, and the spreading speed of that population nears 0. On the other hand, as migration increases, g approaches 0, and the spreading speed of the population increases.

Table 4.3: Computation of speed as $a \to 0$ for the density-dependent case

a	Numerical Speed
10^{-16}	0.087140386
0.01	0.59398919
1	0.91579325

For the density-dependent case, let $g(u) = e^{-au}$. As a approaches 0, $g(u)$ becomes closer to 1. Therefore, the spreading speed, c^*, should be approaching 0. Simulations for $r = 2$ with decreasing values of a can be seen in Table 4.3. One can see that as a decreases, the numerical speed approaches 0.

Table 4.4: Computation of speed as $a \to \infty$ for the density-dependent case

a	Numerical Speed
10	1.101166
1000	1.1523489
10^{16}	1.1748878

Second, as a increases to ∞, $g(u)$ approaches 0. Thus, c^* should be approaching $c^*_{g=0}$, the spreading speed with complete migration. With $r = 2$,

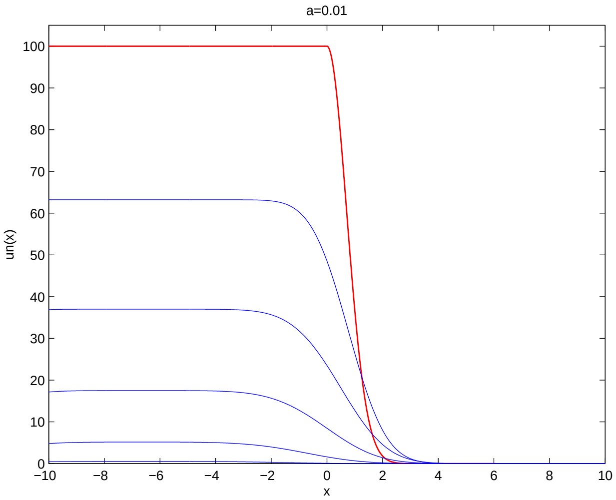


Figure 4.3: Example of Case A with $a = 0.01$ and $u_0(x)$ in red.

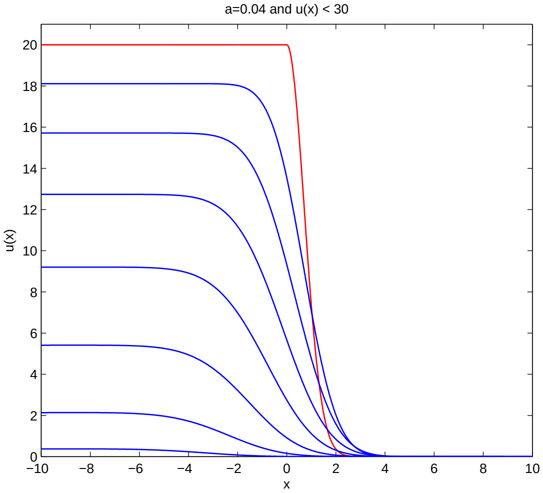


Figure 4.4: Example of Case C with $a = 0.04$ and $u_0(x) < M_1$ in red.

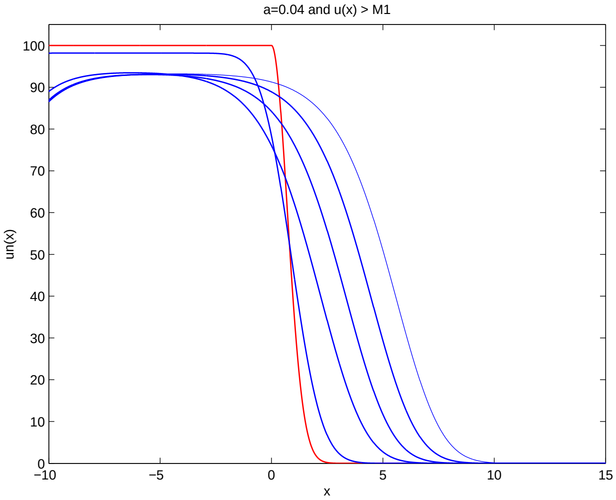


Figure 4.5: Example of Case C with $a = 0.04$ and $u_0(x) > M_1 \ \forall x \in [-20, 1]$ in red.

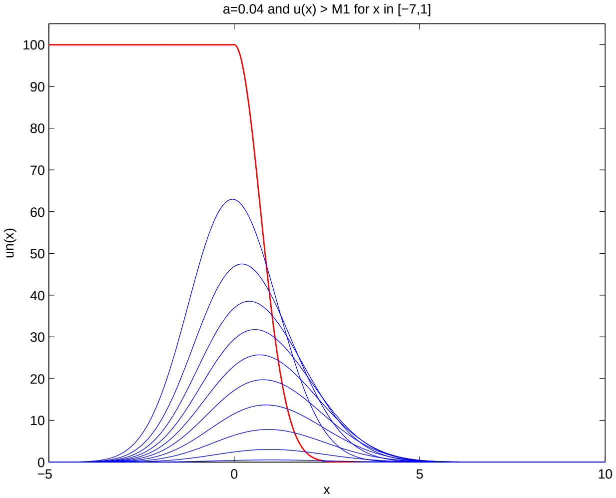


Figure 4.6: Example of Case C with $a = 0.04$ and $u_0(x) > M_1 \ \forall x \in [-7, 1]$ in red.

$c_0^* = 1.17741002$. Table 4.4 displays the spreading speed, c_a^*, as a increases. It is seen that c_a^* is approaching c_0^* as a grows.

Now that we have seen the program accurately predict the spreading speed in a density-dependent case, let us examine the fixed points of the density-dependent model. Section 2.2 defines $h(u) = (1 - e^{-au})f(u)$ with $a > 0$. Recall Case A is the scenario with $h(u) < u$ for all u and contains only one fixed point, $u = 0$. Case B includes the addition of a second fixed point, M_1, and Case C has a third fixed point, M_2 such that $M_1 < M_2 < M$. In Case C for $M_1 < u < M_2$, $h(u) > u$. It is believed that for Case A, the population will not spread. Instead, after some number of generations, the density of the population approaches zero. In Case C, in order for a population with a density dependent dispersal rate to spread, the initial population density, $u_0(x)$ must be greater than or equal to M_1 for x lying in a sufficiently large interval. If this condition is not satisfied, the population dies out as in Case A.

In order to check this hypothesis, various populations are simulated using the numerical program. The program simulates a population with $g(u) = e^{-au}$ for a specific a value. For all figures, the initial population data is graphed in red and all other population densities are blue. The results of the simulation with $a = 0.01$ can be seen in Figure 4.3 where the population dies out after about 5 iterations. When $a = 0.04$, the fixed points of the system were found to be $M_1 = 40.40$ and $M_2 = 92.61$. With $u_0(x) < M_1 \forall x$, it takes about 7 iterations for the population to die out as can be seen in Figure 4.4. Choosing $u_0(x) > M_1$ for $x \in [-20, 1]$ allows the population to spread in Figure 4.5. However, if $u_0(x) > M_1$ only for $x \in [-7, 1]$, Figure 4.6 shows that the population density approaches 0 after about 10 iterations. A summary of the tests can been seen in Table 4.5. Many different a values were tested but all yielded results like those seen above. It was observed that if a is small, the population will die out, yet if a is large enough, the population will spread under the correct initial conditions.

Table 4.5: Investigations of the fixed points

a	$u_0(x)$	Case	Outcome	Figure
0.001	-	A	Population dies out	(4.3)
1	$\geq M_1 \forall x \in [-10, 1]$	C	Population spreads	(4.5)
1	$\leq M_1 \forall x$	C	Population dies out	(4.4)
1	$\geq M_1 \forall x \in [-7, 1]$	C	Population dies out	(4.6)

Chapter 5

Conclusions

The aim of this project was to study the population model found in [17] and then to develop an algorithm to simulate the spread of a species and estimate the spreading speed of the population. Once the program was developed, the application was used to predict the speed of populations with a density-independent dispersal rates. The numerical speed from the program was then compared to the analytical spreading speed found using Equation (2.2). After the accuracy of the application was estimated, the simulations were expanded to also include populations with a density-dependent dispersal rate. In this case, the fraction of the population that does not disperse now depends on the local population density, and as a result the fixed points of the model are altered. An investigation into the effects of the new fixed points was also performed.

For the density-independent case, the numerical program was used to simulate populations with various intrinsic growth rates and migratory habits. It was observed that as the population's migration declined or the growth rate expanded, it become harder to numerically estimate the spreading speed of the population. However, even with the aforementioned conditions, the program was able to predict the speed with less than a 1% relative error.

After, the dispersal rate was assumed to be density-dependent and new simulations were run. With $g(u) = e^{-au}$, as a was decreased the numerical spreading speed approached 0. This coincided with predictions since as a decreases, $g(u) \to 1$, and thus there is no migration. Alternatively, simulations for a specific growth rate with increasing values of a showed the numerical speed approaching the spreading speed in the density-independent case with full migration. Since the program produced expected results in the density-dependent case, the accuracy displayed in predicting the spreading speed for density-independent dispersal rates carries over to the density-dependent case.

Because the fixed points of the model change when the dispersal rate becomes density-dependent, this paper also studied the effects these points have on the spread of a given population. Although 0 remains a fixed point in the density-dependent case, there may be one or two more other fixed points. The numerical program illustrates that if a is too small, the population cannot spread and will eventually die out. Similar results were seen if the initial population density was not greater than the new fixed point or at least not on a sufficiently large enough interval. However, if a was chosen such that there exists three fixed points, 0, M_1, M_2, and $u_0(x) > M_1$ for $x \in [-L, L]$ with L sufficiently large, then the population spread.

Because of the overwhelming effect an invading species can have on an area, it is important to be able to predict the spread of the species in order to begin controlling the problem. Although this project worked to expand the ability to make these predictions, there is still much work to be done. A study to derive an analytical formula for the spreading speed in the density-dependent case would help to verify the accuracy of the numerical program. Since it has been shown that many species actually do have density-dependent dispersal rates, a program that accurately predicts population spread in the density-dependent case could be a useful way to simulate possible ways to reduce or stop the spread of a broader range of species. Furthermore, it is also recommended that effort be put into deriving an analytical formula for the fixed points of the model in order to more accurately describe the system.

Bibliography

[1] Al-Omari, J.; Gourley, S. (2002) Monotone travelling fronts in an age-structured reaction-diffusion model of a single species. *Journal of Mathematical Biology.* Vol. 45, 294-312.

[2] Andow, D.; Kareiva, P.; Levin, S.; Okubo, A. (1990) Spread of invading organisms. *Landscape Ecology.* Vol. 4, 177-188.

[3] Aronson, D.G.; Weinberger, H.F. (1975) Nonlinear diffusion in population genetics, combustion, and nerve propagation. *Partial Differential Equations and Related Topics, Lecture Notes in Mathematics.* Vol. 446.

[4] Caswell, H,; Neubert, M. (2000) Demography and Dispersal: Calculations and Sensitivity Analysis of Invasion Speed for Structured Populations. *Ecology.* Vol. 81, 1613-1628.

[5] Dytham, C.; Murrell, D.; Travis, J. (1999) The Evolution of Denisty-Dependent Dispersal. *Proceedings: Biological Sciences.* Vol. 266, 1837-1842.

[6] Engen, S.; Lande, R.; Saether, B. (1999) Finite Metapopulation Models with Density-Dependent Migration and Stochastic Local Dynamics. *Proceeding: Biological Sciences.* Vol. 266, 113-118.

[7] Fisher, R.A. (1937) The advance of advantageous genes. *Ann. of Eugenics.* Vol. 37, 355-369.

[8] Hastings, A.; et. al. (2005) The spatial spread of invasions: new developments in theory and evidence. *Ecology Letters.* Vol. 8, 91-101.

[9] Kerstre, H.; Levin, D. (1969) Density-Dependent Gene Dispersal in Liatris. *The American Naturalist.* Vol. 103, 61-74.

[10] Kot, M.; Lewis, M.; Neubert, M. (2000) Invasion Speeds in Fluctuationg Environments. *Proceedings: Biological Sciences*. Vol. 267, 1603-1610.

[11] Kot, M.; Neubert, M.; Wang, M. (2002) Integrodifference equations, Allee effects, and invasions. *Journal of Mathematical Biology*. Vol. 44, 150-168.

[12] Lewis, M.; Li, B.; Weinberger, H. (2002) Spreading speed and linear determinacy for two-species competition models. *Journal of Mathematical Biology*. Vol. 45, 219-233.

[13] Lewis, M.; Li, B.; Weinberger, H. (2005) Spreading speeds as slowest wave speeds for cooperative systems. *Mathematical Biosciences*. Vol. 196, 82-98.

[14] Lewis, M.; Veit, R. (1996) Dispersal, Populations Grwoth, and the Allee Effect: Dynamics of the House Finch Invasion of Eastern North America. *The American Naturalist*. Vol. 148, 255-274.

[15] Li, B.; Petrovskii, S. (2003) An exactly solvable model of population dynamics with density-dependent migrations and the Allee effect. *Mathematical Biosciences*. Vol. 186, 79-91.

[16] Liang, X.; Zhao, X.-Q. (2007) Asymptotic Speeds of Spread and Traveling Waves for Monotone Semiflows with Applications. *Communications on Pure and Applied Mathematics*. Vol. LX, 0001-0040.

[17] Lui, R.; Volkov, D. (2006) Spreading Speed and Traveling Wave Solutions of a Partially Sedentary Population. Unknown.

[18] Weinberger, H. (2002) On spreading speeds and traveling waves for growth and migration models in a periodic habitat. *Journal of Mathematical Biology*. Vol. 45, 511-548.

[19] Xiao, D.; Zhao, X. (2006) The Asymptotic Speed of Spread and Travelling Waves for a Vector Disease Model. *Journal of Dynamics and Differential Equations*. Vol. 18, 1001-1019.

Appendix A

Matlab M-files

```
function thesis_numerics = thesis_numerics(numit, testpt);

%                       VARIABLES

global mu % Mean
global sigma % Standard deviation
global r % Intrinsic growth rate, must be greater than one
global M % Carrying capacity of population
global gamma % Value between 0 and M where horizontal distance is measured
global dx % Delta x
global a % Right endpoint of interval
global x % Interval
global lx % Length of the vector x
global B % K is defined on [-B, B]
global ga % Part of gconv

g = (0);
mu = 0;
sigma = 1;
r = 1.5;
M = 100;
gamma = testpt;
ga =0.04;

a = -24; % initial beginning of interval
```

```
b = 24; % initial end of interval
dx = 1; % delta x
x = a:dx:b; %initial set of points
lx = length(x);

B = 6; % bound of K s.t. integral of K from -B to B is 1
epsilon = 10e-20; % small number vey close to zero

%                         CREATE INITIAL DATA                     %
%                 K is the probability density function           %
%                 u is the population density function            %

for i=1:lx
    unext(i)= uinitial(x(i)); % u values on [a, b]
end

for i=1:(2*B/dx+1)
    K(i) = Kconv(-B+(i-1)*dx); % K values on [-B, B]
end

hold on;

%For each iteration:
%   1. Find an interval [a,b] s.t f(u(x))<=epsilon for all x>=b+B.
%   2. Calculate f using the u values on the new interval [a, b].
%   3. Convolve (1-g)*f with K.
%   4. Add your convolution and g*f to find your new u values.
%   4. Plot u versus x and calculate spreading speed

for j=1:numit

%                         Non-compact support
    k=0;
    while (fconv(unext(lx-k))<=epsilon) % find newb
        newb=lx-k;
        k=k+1;
        if(k==lx)
            newb=lx;
```

```
            break;
        end
    end

    b = a+newb*dx+B; % new right endpoint
    x=a:dx:b; % new interval x
    lx=length(x); % length of x

%finds u on new interval [a,b]
        for i=1:lx
            if(j==1)
                unext(i)= uinitial(x(i)); % initial u values on [a, b]
            elseif (i) > length(unext)
                unext(i) = 0; % Assume u=0 for all x>=b
            end
        end
% End non-compact support

%                             Compact support

n=0;
while(unext(lx-n)<=epsilon)
    ansB=lx-n;
    n=n+1;
end

b=a+ansB*dx+B;
x=a:dx:b;
lx=length(x);
tempu=unext;

% finds u on new interval [a,b]
     for i=1:lx
         if(j==1)
             unext(i)= uinitial(x(i)); % initial u values on [a, b]
         elseif (i) > length(unext)
             unext(i) = 0; % Assume u=0 for all x>=b
         end
```

```
     end
% End compact support

% finds f and g values on [a,b]
    for i=1:lx
        f1(i) = fconv(unext(i)); % f values on [a, b]
        %g1(i) = gconv(f1(i)); % g values on [a, b]
    end

% finds f and g values on [a+B, b-B] so that convolution is on [a,b]
    for i=1:(lx-2*(B/dx))
        f(i) = fconv(unext(i+B/dx)); % f values on [a+B,b-B]
        %g(i) = gconv(f(i));
    end

% find convolution of (1-g)*f with K
    first = sqrt(dx)*(1-g)*(f(1:lx-2*(B/dx))); % on [a+B,b-B]
    second = K*sqrt(dx); % on [-B, B]
    ansconv = conv(first, second);% on [a, b]
    unext = ansconv + g*f1(1:lx); % on [a, b]

    plot(x,unext(1:lx));
    xdist(j)= distance(unext);
end

xlabel('x');
ylabel('un(x)');
title('Population density');

n=1:1:length(xdist);
limit=xdist./n

hold off;
figure;
plot(limit);
title('Horizontal distance divided by iteration number');
xlabel('iteration number');
ylabel('distance/iteration number');
```

```
limit=limit';
save filename.out limit -ASCII;

function u = uinitial(x);

global M

if (x<=0)
    u=M;
else
     u=M*exp(-x^2);  % monotone decreasing
     u=M*exp(-7*x/8)*(abs(cos(3*x)) + 1)*.5; % oscillating
end

function K = Kconv(x);

global mu
global sigma

tmp1 = ((x-mu)^2)/(2*sigma^2);
tmp2 = sqrt(2*pi*sigma^2);
K = (exp(-tmp1))/tmp2;

function f = fconv(u);

global r
global M

f=(r*M*u)/(M+(r-1)*u);

function gconv = gconv(fu)

global ga

gconv = exp(-ga*fu);

function dist = distance(unext);
```

```
global gamma
global dx
global a
global lx

for i=1:lx
    if(unext(i)>=gamma)&&(unext(i+1)<=gamma)
        point1=i;
        point2=i+1;
        break
    end
end

x0=a+(i-2)*dx;
x1=a+(i-1)*dx;
x2=a+(i)*dx;
x3=a+(i+1)*dx;

dist=interp1([unext(point1-1), unext(point1), unext(point2), unext(point2+1)],
         ...[x0, x1, x2, x3] ,gamma,'cubic');
```