Numerical simulations of traveling wave solutions in a drift paradox inspired diffusive delay population model

Z. Jackiewicz a, b, H. Liu a, B. Li c, Y. Kuang a, * 

a Department of Mathematics, Arizona State University, Tempe, AZ 85287, USA 
b Department of Applied Mathematics, AGH University of Science and Technology, Kraków, Poland 
c Department of Mathematics, University of Louisville, Louisville, KY 40292, USA 

Received 5 September 2011; received in revised form 15 February 2012; accepted 5 June 2012 
Available online 27 June 2012 

Abstract 

We describe numerical algorithm for the simulation of traveling wave solutions in a newly formulated drift paradox inspired diffusive delay population model. We use method of lines to discretize the boundary value problem for the reaction-diffusion equations and we integrate in time the resulting system of delay differential equations using the embedded pair of continuous Runge–Kutta methods of order four and three. We advance the solution with the method of order four and the approximations of order three are used for local error estimation. Numerical results demonstrate the robustness, efficiency, and accuracy of our approach. Moreover, these numerical results confirm the recent theoretical results on the minimum traveling wave speed for this model. 
© 2012 IMACS. Published by Elsevier B.V. All rights reserved. 

Keywords: Reaction-diffusion equations; Traveling wave solutions; Drift paradox; Numerical simulations; Delay models 

1. Introduction 

Many organisms persist in their environments despite the presence of constant advection into often unfavorable habitats. Such intriguing phenomena is referred as the “drift paradox” in ecological literature. Examples include plants with windborn seeds, organisms in rivers and estuaries, and marine organisms with larval dispersal influenced by ocean currents [1]. A closer look at such species growth and disperse activities reveals that various specific biological and physical processes can contribute to the persistence of a given organism in such an environment [5,6]. 

There is a significant and growing interest in modeling population growths in a setting mixing advection with diffusion [2,3,10,17]. For example, the paper by Pachepsky et al. [16] studied such a population growth with additional assumptions that the reproduction occurs only in the stationary phase and the population can be divided into two interacting compartments: individuals residing on the benthos and individuals drifting in the flow. They proved that persistence of the population is guaranteed if at low population densities the local growth rate of the stationary component of the population exceeds the rate of entry of individuals into the drift. In [11], the authors incorporated a

* Corresponding author. 

E-mail addresses: jackiewi@math.la.asu.edu (Z. Jackiewicz), hliu@mathpost.asu.edu (H. Liu), bing.li@louisville.edu (B. Li), kuang@asu.edu (Y. Kuang). 

0378-4754/$36.00 © 2012 IMACS. Published by Elsevier B.V. All rights reserved. 
http://dx.doi.org/10.1016/j.matcom.2012.06.004
maturation delay $\tau$ in the model of Pachepsky et al. [16] which yielded the following two-dimensional single-species diffusive-delay population model

\begin{equation}
\frac{\partial n_d}{\partial t}(x, t) = \delta n_b(x, t) - \sigma n_d(x, t) - v \frac{\partial n_d}{\partial x}(x, t) + d \frac{\partial^2 n_d}{\partial x^2}(x, t),
\end{equation}

\begin{equation}
\frac{\partial n_b}{\partial t}(x, t) = r n_b(x, t - \tau) - \frac{r}{\kappa} n_b^2(x, t) - \delta n_b(x, t) + \sigma n_d(x, t),
\end{equation}

$x \in [a, b], t \geq 0$, with initial and boundary conditions

\begin{equation}
\begin{cases}
 n_d(x, 0) = n_d^0(x), & n_b(x, t) = n_b^0(x, t), & x \in [a, b], & t \in [-\tau, 0], \\
 n_d(a, t) = g_1(t), & n_d(b, t) = g_2(t), & t \geq 0.
\end{cases}
\end{equation}

Here, $n_b(x, t)$ is the population density of the benthos, $n_d(x, t)$ is the population density of the drift, the delay $\tau$ is the time taken from birth to maturity, $r$ is the rate of the benthic population at which individuals are born, $\kappa$ is the carrying capacity, $\delta$ is the per capita rate at which individuals in the benthic population enter the drift, $\sigma$ is the per capita rate at which the organism return to benthic population from drifting, $d$ is the diffusion coefficient, $v$ is the advection speed experienced by the organisms, and $n_d^0(x), n_b^0(x, t), g_1(t)$, and $g_2(t)$ are given functions. Inheriting the assumptions proposed in [16], the diffusion term $d$ for the drift compartment represents the effect of heterogeneous stream flows and random individual swimming.

Model (1.1) is a cooperative delayed reaction-diffusion system. The existence of analytic solution in (1.1) with appropriate initial and boundary conditions is ensured by the results in Martin and Smith [12]. It was established in [11] that the model (1.1) admits traveling wave solutions. In this paper we propose a numerical algorithm for the numerical simulations of these solutions. This method can be easily adapted to simulate other diffusive delay models such as those studied by Gourley and Kuang [8,9].

2. Discretization in space

Let be given an integer $N > 0$ and consider the uniform grid $x_i = a + ih$, $i = 0, 1, \ldots, N + 1$, where $(N + 1)h = b - a$. Putting $x = x_i$, $i = 1, 2, \ldots, N$, into (1.1) and discretizing the partial derivative $\frac{\partial n_d}{\partial x}$ at the points $(x_i, t)$ by the central differences

\[
\frac{\partial n_d}{\partial x}(x_i, t) \approx \frac{u_d(x_{i+1}, t) - u_d(x_{i-1}, t)}{2h},
\]

and the partial derivative $\frac{\partial^2 n_d}{\partial x^2}$ at the points $(x_i, t)$ by the finite differences of second order

\[
\frac{\partial^2 n_d}{\partial x^2}(x_i, t) \approx \frac{n_d(x_{i+1}, t) - 2u_d(x_i, t) + u_d(x_{i-1}, t)}{h^2},
\]

we obtain a system of the delay differential equations of the form

\begin{equation}
\begin{cases}
 n_{d,i}^{'}(t) = \delta n_{b,i}(t) - \sigma n_{d,i}(t) - v \frac{n_{d,i+1}(t) - n_{d,i-1}(t)}{2h} \\
+ d \frac{n_{d,i+1}(t) - 2u_d(x_i, t) + n_{d,i-1}(t)}{h^2}, \\
 n_{b,i}^{'}(t) = r n_{b,i}(t - \tau) - \frac{r}{\kappa} n_{b,i}^2(t) - \delta n_{b,i}(t) + \sigma n_{d,i}(t),
\end{cases}
\end{equation}

$i = 1, 2, \ldots, N$. Here, $n_{d,i}(t)$ are approximations to $n_d(x_i, t)$ and $n_{b,i}(t)$, $n_{b,i}(t - \tau)$ are approximations to $n_b(x_i, t)$, $n_b(x_i, t - \tau)$, respectively. For $i = 1$ and $i = N$ we have to incorporate into the above equations the boundary conditions from (1.2), i.e.,

\[
n_d,0(t) = g_1(t) \quad \text{and} \quad n_d,N+1(t) = g_2(t).
\]
This leads to
\[
\begin{align*}
    n_{d,1}'(t) &= \delta n_{b,1}(t) - \sigma n_{d,1}(t) - v \frac{n_{d,2}(t)}{2h} \\
    &+ d \frac{n_{d,2}(t) - 2 u_{d,1}(t)}{h^2} + v \frac{g_1(t)}{2h} + d \frac{g_1(t)}{h^2}, \\
    n_{b,1}'(t) &= r n_{b,1}(t - \tau) - \frac{r}{\kappa} n_{b,1}(t) - \delta n_{b,1}(t) + \sigma n_{d,1}(t),
\end{align*}
\]
(2.2)
and
\[
\begin{align*}
    n_{d,N}'(t) &= \delta n_{b,N}(t) - \sigma n_{d,N}(t) + v \frac{n_{d,N-1}(t)}{2h} \\
    &+ d \frac{-2 u_{d,N}(t) + n_{d,N-1}(t)}{h^2} - v \frac{g_2(t)}{2h} + d \frac{g_2(t)}{h^2}, \\
    n_{b,N}'(t) &= r n_{b,N}(t - \tau) - \frac{r}{\kappa} n_{b,N}(t) - \delta n_{b,N}(t) + \sigma n_{d,N}(t).
\end{align*}
\]
(2.3)
We will rewrite next the system (2.2), the system (2.1) corresponding to \(i = 2, 3, \ldots, N - 1\), and the system (2.3) in the vector form. Introducing the notation
\[
\begin{align*}
    n_d(t) &= [n_{d,1}(t) \cdots n_{d,N}(t)]^T, \quad n_b(t) = [n_{b,1}(t) \cdots n_{b,N}(t)]^T, \\
    G_1(t) &= [g_1(t) 0 \cdots 0 -g_2(t)]^T, \quad G_2(t) = [g_1(t) 0 \cdots 0 g_2(t)]^T,
\end{align*}
\]
\[
\begin{align*}
    S &= \begin{bmatrix}
        0 & -1 & 1 & 0 & -1 \\
        1 & 0 & -1 & 1 & 0 \\
        \vdots & \vdots & \vdots & \vdots & \vdots \\
        1 & 0 & -1 & 1 & 0
    \end{bmatrix}, \quad T = \begin{bmatrix}
        -2 & 1 & 1 & -2 & 1 \\
        1 & -2 & 1 & 1 & -2
    \end{bmatrix},
\end{align*}
\]
we obtain
\[
\begin{align*}
    n_d'(t) &= \left( \frac{v}{2h} S + \frac{d}{h^2} T - \sigma I \right) n_d(t) \\
    &+ \delta n_d(t) + \frac{v}{2h} G_1(t) + \frac{d}{h^2} G_2(t), \\
    n_b'(t) &= \sigma n_d(t) - \delta n_b(t) - \frac{r}{\kappa} n_b(t) + r n_b(t - \tau),
\end{align*}
(2.4)
t \geq 0. Here, \(n_d(t)^\circ\) stands for component wise exponentiation. This system has to be supplemented by appropriate initial conditions. It follows from (1.2) that these initial conditions take the form
\[
\begin{align*}
    n_d(0) &= n_d^0 := [n_d^0(x_1) \cdots n_d^0(x_N)]^T, \\
    n_b(t) &= n_b(t) := [n_b^0(x_1, t) \cdots n_b^0(x_N, t)]^T, \quad t \in [-\tau, 0].
\end{align*}
(2.5)
We reformulate next (2.4) and (2.5) as a single system of nonlinear delay differential equations. Put
\[
\begin{align*}
    \mathbf{n}(t) &= \begin{bmatrix}
        n_d(t) \\
        n_b(t)
    \end{bmatrix}, \quad \mathbf{n}^2(t) = \begin{bmatrix}
        n_d^2(t) \\
        n_b^2(t)
    \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix}
        \frac{v}{2h} S + \frac{d}{h^2} T - \sigma I & \delta I \\
        \sigma I & -\delta I
    \end{bmatrix}.
\end{align*}
\]
\[ A_2 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{r}{\kappa} \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 0 \\ 0 & r \end{bmatrix}, \quad G(t) = \begin{bmatrix} \frac{v}{2h}G_1(t) + \frac{d}{h^2}G_2(t) \\ 0 \end{bmatrix}. \]

Then the system (2.4) assumes the form
\[ n'(t) = A_1 n(t) - A_2 n^2(t) + A_3 n(t - \tau) + G(t), \quad t \geq 0, \]
and the initial condition can be written as
\[ n(t) = n^0(t) := \begin{bmatrix} n^0_d \\ n^0_0(t) \end{bmatrix}, \quad t \in [-\tau, 0]. \quad (2.7) \]

### 3. Discretization in Time

The initial value problem (2.6) and (2.7) is a special case of the delay differential system of equations
\[ \begin{cases} y'(t) = f(t, y(t), y(t - \tau)), & t \in [a, b], \\ y(t) = g(t), & t \in [a - \tau, a], \end{cases} \quad (3.1) \]
where \( f : \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m \) and \( g : [a - \tau, a] \to \mathbb{R}^m \) is a given initial function. To compute a numerical approximation to the solution \( y(t) \) of the problem (3.1) we consider the explicit continuous Runge–Kutta method of order \( p \) with \( s \) stages and the embedded discrete Runge–Kutta method of order \( q = p - 1 \) which is used for the estimation of the local discretization error of the method of order \( p \). On the nonuniform grid
\[ a = t_0 < t_1 < \cdots < t_N, \quad t_N \geq b, \]
the explicit continuous Runge–Kutta method takes the form
\[ \begin{cases} Y_i = y_h(t_n) + h_n \sum_{j=1}^{i-1} a_{ij} F_j, & i = 1, 2, \ldots, s, \\ F_i = f(t_n + c_i h_n, Y_i, y_h(t_n + c_i h_n - \tau)), & i = 1, 2, \ldots, s, \\ y_h(t_n + \theta h_n) = y_h(t_n) + h_n \sum_{j=1}^{s} b_j(\theta) F_j, \end{cases} \quad (3.2) \]

\( \theta \in [0, 1], n = 0, 1, \ldots, N - 1 \). Here, \( h_n = t_{n+1} - t_n \), \( Y_i \) are approximations (possibly of low order) to \( y(t_n + c_i h_n) \), \( i = 1, 2 \ldots, s \), and \( y_h(t_n + \theta h_n) \) is a continuous approximation of order \( p \) to \( y(t_n + \theta h_n) \). The method (3.2) is specified by the abscissa vector \( c = [c_1, \ldots, c_s]^T \), the coefficient matrix \( A = [a_{ij}]_{i,j=1}^{s} \), and the vector of continuous weights \( b(\theta) = [b_1(\theta), \ldots, b_s(\theta)]^T \).

We also consider the discrete Runge–Kutta method of order \( q = p - 1 \) with \( s \) stages
\[ \hat{y}_{n+1} = y_h(t_n) + h_n \sum_{j=1}^{s} \hat{b}_j F_j, \quad (3.3) \]

\( n = 0, 1, \ldots, N - 1 \), and the weight vector \( \hat{b} = [\hat{b}_1, \ldots, \hat{b}_s]^T \), where \( y_h(t_n) \) and \( F_j \) are defined in (3.2). Then as demonstrated in [13–15] (see also [4]) the norm of the difference
\[ \text{est}(t_{n+1}) = ||\hat{y}_{n+1} - y_h(t_{n+1})|| \quad (3.4) \]
provides an estimate of the local discretization error of the continuous Runge–Kutta method at the point \( t_{n+1} \).
The continuous and discrete Runge–Kutta methods can be represented by the following table of their coefficients

<table>
<thead>
<tr>
<th>c</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_h$</td>
<td>$b(\theta)$</td>
</tr>
<tr>
<td>$\hat{y}_{n+1}$</td>
<td>$\hat{b}$</td>
</tr>
</tbody>
</table>

\[
\begin{array}{c|cccccc}
0 & \; & \; & \; & \; & \; & \; \\
\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\
37 & 3388 & 1369 & 1369 & 1369 & 1369 & 1369 \\
17 & 4913 & 4913 & 4913 & 4913 & 4913 & 4913 \\
13 & 366 & 366 & 366 & 366 & 366 & 366 \\
15 & 408375 & 1125 & 1125 & 1125 & 1125 & 1125 \\
1 & 101 & 101 & 101 & 101 & 101 & 101 \\
\end{array}
\]

<table>
<thead>
<tr>
<th>$y_h(t_n + \theta h_n)$</th>
<th>$b_1(\theta)$</th>
<th>$b_2(\theta)$</th>
<th>$b_3(\theta)$</th>
<th>$b_4(\theta)$</th>
<th>$b_5(\theta)$</th>
<th>$b_6(\theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{y}_{n+1}$</td>
<td>$\hat{b}_1$</td>
<td>$\hat{b}_2$</td>
<td>$\hat{b}_3$</td>
<td>$\hat{b}_4$</td>
<td>$\hat{b}_5$</td>
<td>$\hat{b}_6$</td>
</tr>
</tbody>
</table>

The coefficients of the optimal pair of continuous Runge–Kutta method of order $p = 4$ and discrete Runge–Kutta method of order $q = p - 1 = 3$, which is used in our numerical experiments discussed in the next section, are listed below [14]:

\[
\begin{align*}
    b_1(\theta) & = -\frac{866}{824} \cdot \frac{577}{252} \cdot \frac{1}{\theta^4} + \frac{1}{\theta^4} \cdot \frac{806}{618} \cdot \frac{901}{189} \cdot \frac{1}{\theta^3} - \frac{104}{37} \cdot \frac{217}{466} \cdot \frac{1}{\theta^2} + \theta, \\
    b_2(\theta) & = 0, \\
    b_3(\theta) & = -\frac{12}{5} \cdot \frac{308}{722} \cdot \frac{679}{320} \cdot \frac{1}{\theta^4} - \frac{2}{\theta^4} \cdot \frac{178}{380} \cdot \frac{079}{424} \cdot \frac{1}{\theta^3} + \frac{861}{230} \cdot \frac{101}{560} \cdot \frac{1}{\theta^2}, \\
    b_4(\theta) & = -\frac{7}{1} \cdot \frac{816}{1044} \cdot \frac{583}{640} \cdot \frac{1}{\theta^4} + \frac{6}{\theta^4} \cdot \frac{244}{5} \cdot \frac{423}{325} \cdot \frac{936}{936} \cdot \frac{1}{\theta^3} - \frac{63}{293} \cdot \frac{869}{440} \cdot \frac{1}{\theta^2}, \\
    b_5(\theta) & = -\frac{624}{217} \cdot \frac{375}{984} \cdot \frac{1}{\theta^4} + \frac{982}{190} \cdot \frac{125}{736} \cdot \frac{1}{\theta^3} - \frac{1}{762} \cdot \frac{522}{944} \cdot \frac{1}{\theta^2}, \\
    b_6(\theta) & = -\frac{296}{131} \cdot \frac{461}{131} \cdot \frac{1}{\theta^4} + \frac{165}{131} \cdot \frac{1}{\theta^3} - \frac{1}{\theta^2}.
\end{align*}
\]

This embedded pair was implemented in a variable stepsize environment with the estimates of the local discretization errors computed according to the formula (3.4). Following the approach in [7,19,18], described in the context of numerical solution of ordinary differential equations, the initial stepsize $h_0$ was computed from

\[
h_0 = \min\left\{ 0.01 \cdot (b - a), \frac{tol^{1/(p+1)}}{\|Ra, y(a), g(a - \tau)\|} \right\},
\]

$p = 4$. Then for $n = 0, 1, \ldots, N - 1$, the stepsize from $t_n$ to $t_{n+1} = t_n + h_n$ is accepted if

\[
est(t_{n+1}) \leq tol,
\]
where $tol$ is the accuracy tolerance specified by the user, and the new stepsize $h_{n+1}$ from $t_{n+1}$ to $t_{n+2}$ is computed from

$$h_{n+1} = \xi h_n \left( \frac{tol}{\text{est}(t_{n+1})} \right)^{1/(p+1)},$$

$p = 4$. Here, $0 < \xi \leq 1$ is a safety coefficient built into stepsize selection mechanism to avoid too many rejected steps. In our implementation of the code we have chosen $\xi = 0.8$. If $\text{est}(t_{n+1}) > tol$ the stepsize is rejected. In such cases the computations were repeated with a halved stepsize.

4. Numerical experiments

The algorithm described in the previous section was applied to the initial value problem (2.6) and (2.7) obtained by discretization of the boundary value problem (1.1) and (1.2) in space variable $x$ for $x \in [-200, 200]$ as described in Section 2. This problem was integrated for $t \in [0, 40]$. The parameters of the model (1.1) were chosen as $\tau = 1$, $r = 1$, $\kappa = 1$, $d = 1$, $\delta = 1.8$, $\sigma = 0.8$, and $\nu = 0.5$. The initial conditions in (1.2) were chosen as pulse functions located at $x = 0$, i.e.,

$$n_d(x, 0) = 2 \exp \left( \frac{-x^2}{1000} \right), \quad x \in [-200, 200],$$

$$n_b(x, 0) = \exp \left( \frac{-x^2}{1000} \right), \quad x \in [-200, 200], \quad t \in [-1, 0].$$

This choice translates to the appropriate initial functions $n_d^0$ and $n_b^0(t)$ appearing in (2.7). The boundary conditions in (1.2) were chosen as

$$n_d(-200, t) = n_d(200, t) = 0, \quad t \in [0, 40].$$

These conditions affect the definition of the forcing term $G(t)$ appearing in the Eq. (2.6).

The results of numerical experiments are presented in Figs. 1–5. In Figs. 1 and 2 we have plotted approximations to $n_d(x, t)$ and $n_b(x, t)$ versus $x \in [-200, 200]$ for $t = 0, 10, 20, 30, 40$. In Figs. 3 and 4 we have plotted approximations to $n_d(x, t)$ and $n_b(x, t)$ versus $x \in [-200, 200]$ and $t \in [0, 40]$. These plots correspond to accuracy tolerance $tol = 10^{-6}$. In Fig. 5 we have plotted the adaptive stepsize pattern $h$ versus $t$ for $tol = 10^{-3}$, $10^{-6}$, and $10^{-9}$. The rejected steps are marked by the symbol ‘x’. We can observe that after some initial stepsize adjustment the stepsize settles on fixed values. We have also listed in Table 1 some cost statistics and global errors. In this table $tol$ stands for the accuracy tolerance, $ns$ for the number of successful steps, $nr$ for the number of rejected step, $nfe$ is the number of evaluations of the right hand side of the delay differential equation (2.6), $time$ is the CPU time in seconds on Intel Core(TM) i7.
Fig. 2. Approximation to \( n_b(x, t) \) versus \( x \) for \( t = 0, 10, 20, 30, \) and 40.

Fig. 3. Approximation to \( n_d(x, t) \) versus \( x \) for \( t \).

Fig. 4. Approximation to \( n_b(x, t) \) versus \( x \) and \( t \).

Fig. 5. Stepsize \( h \) versus \( t \) for \( tol = 10^{-3}, 10^{-6} \) and \( 10^{-9} \). Rejected steps are marked by the symbol ‘×’.
3.47 Ghz processor, and $e_d$ and $e_b$ are the global errors in $n_d$ and $n_b$ at the end point of integration. The integration was continued until $l_{n+1} > b$ and the approximations at the end point $t = b$ were computed by continuous extensions

$$y_b(b) = y_h(t_n) + h_n \sum_{j=1}^{6} b_j(\theta) F_j,$$

with $\theta = (b - t_k)/(t_{n+1} - t_k)$. The reference solution was computed using our algorithm with a stringent tolerance $tol = 10^{-12}$. We can observe quite good proportionality between the global errors at the end point of integration and the accuracy tolerance.

### 5. Discussion

Our numerical results not only confirmed the existence of a traveling wave solution for model (1.1) which has been rigorously established [11], they also provide clear pictures of the shape and stability of these wave solutions. In addition to our tailor-made and highly effective discretization techniques with high-order time-stepping methods and adjustable time steps, there is an alternative way to approximate solutions of model (1.1). We can view the discrete delay as a limiting case for some appropriately selected distributed delay formulations. In other words, like in Gourley and Kuang [9], we can approximate the single discrete delay term with a distributed delay and then use suitable numerical techniques, such as the first author’s method of Hermite quadrature, to compute related integrals. In the following, we provide some additional details on this approach.

A standard argument on population with age structure and diffusion (see Section 4 of [9]) assumes

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial a} = D(a) \frac{\partial^2 u}{\partial x^2} - d(a) u$$

where $u(x, a, t)$ is defined as the density of population at time $t$, at age $a$ and at location $x$, and $D(a)$, $d(a)$ denote age-dependent diffusion rate and death rate. Let $r$ be the maturation age. Then

$$w(x, t) = \int_{r}^{\infty} u(x, a, t) da$$

is the total matured population density. One can show that (again see Section 4 of [9]) $w(x, t)$ satisfies

$$\frac{\partial w}{\partial t} = D_m \frac{\partial^2 w}{\partial x^2} - d_m w + \frac{\exp(-\int_{0}^{t} d_l(a) da)}{\sqrt{4\pi \alpha}} \int_{-\infty}^{\infty} b(w(y, t - r)) e^{-(x-y)^2/4\alpha} dy,$$

for $t \geq r$. Here $D_m$, $d_m$ are diffusion and death rates of the matured, $D_l$, $d_l$ denote diffusion and death rates of the immature,

$$\alpha = \int_{0}^{r} D_l(a) da \quad \text{and} \quad b(w(x, t)) = u(x, 0, t).$$

Using this result, considering the benthic population in our model, we can modify the equation for $n_b$. In our original model, we did not assume death rate for the benthic population, so for convenience let $\mu$ be the homogeneous age-independent death rate for benthic population. We also assume similarly that the diffusion rate $D_l$ is age independent, and the birth rate is location independent. Therefore our benthic density equation is reformulated as

$$\frac{\partial n_b}{\partial t} = \sigma n_d - \partial n_b - \frac{r}{\kappa} n_b^2 + \frac{\exp(-\mu \tau)}{\sqrt{4\pi D_l \tau}} \int_{-\infty}^{\infty} n_b(y, t - \tau) e^{-(x-y)^2/4D_l} dy.$$
where $D$ is the constant age-independent diffusion rate for the benthic population, and the other parameters are the same as previously defined. Notice also that the diffusion of the benthic compartment is implicitly assumed and in fact expressed by the parameter $D$. Intuitively, the smaller $\mu$, $D$ are, i.e., the smaller the death rate and diffusion rate for benthic population are, the closer this model is to the original model (1.1). Our preliminary implementation of this approximation method indeed produced solutions that converge to the solution of model (1.1) as we let both $\mu$ and $D$ tend to zero. However, the discretization techniques with high-order time-stepping methods and adjustable time steps presented here is superior in terms accuracy, robustness and user-friendliness.

Acknowledgements

We thank the reviewers for their helpful suggestions/comments. This work of YK and HL is supported in part by NSF DMS-0436341 and DMS-0920744. BL is partially supported by NSF DMS-0616445.

References